

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ПРОГРАМУВАННЯ

До захисту допускається:  
завідувач кафедри \_\_\_\_\_ ІТП  
\_\_\_\_\_ Лифар В.О.  
« \_\_\_\_\_ » \_\_\_\_\_ 2023 р.

**ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА**  
**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
НА ТЕМУ:

Веб-сайт онлайн-маркетплейсу

---

---

---

Освітній ступінь – «бакалавр»

Спеціальність \_\_\_\_\_ 121 «Інженерія програмного забезпечення»  
(шифр і назва спеціальності)

Керівник проекту:

\_\_\_\_\_ Лифар В.О.  
(підпис) (ініціали, прізвище)

Здобувач вищої освіти

\_\_\_\_\_ Попов Д.В.  
(підпис) (ініціали, прізвище)

Київ 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Інформаційних технологій та програмування

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

завідувач кафедри ІТП

Лифар В.О.

«    » 2023 р.

**З А В Д А Н Н Я**  
**НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Попов Денис Вадимович  
(прізвище, ім'я, по батькові)

1. Тема роботи: Веб-сайт онлайн-маркетплейсу

Керівник проекту (роботи) Лифар Володимир Олексійович, д.т.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 26.04.2023 № 240/15.15-ОД

3. Вихідні дані до роботи матеріали переддипломної практики.

---

---

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): а) аналіз предметної області та аналогів;

б) вибір та обґрунтування програмних засобів розробки;

в) розробка функціонального додатку;

г) висновки.

---

---

5. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

6. Дата видачі завдання 24.03.2023

Керівник

\_\_\_\_\_  
(підпис)

Завдання прийняв до виконання

\_\_\_\_\_  
(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Одержання завдання на виконання роботи	24.03.2023-25.03.2023	
2	Аналіз існуючих платформ	27.03.2023-28.03.2023	
3	Узагальнення даних літературних джерел	10.04.2023-11.04.2023	
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху. Розробка базової частини інтерфейсу	19.04.2023-25.04.2023	
5	Укладання та тестування фінальної версії програмного продукту	27.04.2023-05.05.2023	
6	Укладання, оформлення та погодження пояснювальної записки з керівником	28.05.2023-30.05.2023	
7	Здача готової пояснювальної записки на кафедрі	28.05.2023-30.05.2023	
8	Укладання доповіді і презентації	31.05.2023-05.06.2023	

**Здобувач вищої освіти**

\_\_\_\_\_ ( підпис )

**Попов Д.В.**

\_\_\_\_\_ (ініціали, прізвище)

**Керівник**

\_\_\_\_\_ ( підпис )

**Лифар В.О.**

\_\_\_\_\_ (ініціали, прізвище)

## РЕФЕРАТ

Робота містить: 58 сторінок основного тексту, 68 сторінок додатків, 53 рисунки, 6 таблиць, 28 використаних джерел.

Метою випускної кваліфікаційної роботи є вивчення створення веб-сайту онлайн-маркетплейсу, методів реалізації, оптимальних рішень, в плані програмування і підбору продуктивних інструментів.

Було проаналізовано багато аналогів, їх принцип роботи, перелік наданих можливостей і кращі рішення взаємодії з користувачами.

В результаті виконаної роботи, було реалізовано веб-додаток, що дозволяє отримувати різноманітні послуги у 4 актуальних сферах: додаткової освіти, дозвілля, практико-орієнтованих проектах та сфери зайнятості – для студентів; для викладачів і роботодавців: презентувати корисні матеріали, заходи для студентів та отримувати зворотній зв'язок.

Веб-додаток може бути легко масштабованим для подальшої розробки даного проекту.

Запрограмований додаток реалізує задуманий функціонал. Документи, що додаються, допоможуть цільовій аудиторії легко і радісно користуватися готовим продуктом.

## ЗМІСТ

ВСТУП .....	7
1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1 Актуальність розроблення веб-додатка маркетплейса .....	12
1.2. Огляд цільової аудиторії та актуальних для неї послуг і товарів .....	15
1.3 Огляд і порівняння наявних пропозицій .....	22
1.4 Висновок за розділом 1 .....	30
2. РОЗРОБКА МОДЕЛІ ДОДАТКА.....	31
2.1 Постановка завдання.....	31
2.2 Вибір інструментарію програмування .....	46
2.3 Функціональна схема додатка .....	48
2.4 Опис схеми даних .....	52
2.5 Висновок із розділу 2.....	55
3. РЕАЛІЗАЦІЯ РІШЕННЯ.....	56
3.1. Опис алгоритмів роботи програми.....	56
3.2 Демонстрація функціонування програми.....	59
3.3 Опис застосування програмного продукту .....	62
3.4. Список документації до додатка. ....	63
3.5 Можливості подальшого розвитку рішення.....	66
3.6 Висновок із розділу 3.....	67
ВИСНОВОК.....	68
ДЖЕРЕЛА .....	69

## ВСТУП

Усе більше сучасних людей вважають за краще робити покупки в інтернеті, і це не дивно. Онлайн-шопінг надає безліч переваг, таких як зручність, доступність, широкий вибір товарів і конкурентні ціни. Однак, існують і деякі недоліки, такі як відсутність можливості подивитися товар в живу, недостатня інформація про продукт і ризик потрапити на шахраїв.

У цій ситуації на допомогу приходять сайти-маркетплейси. Вони надають можливість купувати товари та послуги у різних продавців в одному місці. По суті, це віртуальний торговий майданчик, де магазини і приватні продавці можуть продавати свої товари. Як правило, на сайті-маркетплейсі представлено товари різних категорій, починаючи від електроніки та одягу і закінчуючи послугами з ремонту та обслуговування автомобілів.

Сайти-маркетплейси є новим трендом в онлайн-шопінгу і стають дедалі популярнішими серед споживачів. Вони надають широкий вибір товарів, конкурентні ціни та зручний сервіс покупцям, а також додаткову можливість продавцям продавати свої товари.

Один із популярних і затребуваних ресурсів нині - маркетплейси освітніх послуг, які забезпечують можливість просування та надання навчальних курсів у сфері освіти. У контексті сучасних навчальних закладів і приватних викладачів вибір правильного майданчика або навіть декількох майданчиків для поширення і просування їхніх курсів стає все більш важливим. Для цих цілей можуть бути використані різноманітні онлайн-механізми, призначені для ефективної взаємодії з цільовими споживачами освітніх послуг.

Онлайн маркетплейс освітнього закладу можна розглядати як агрегатор не лише освітніх товарів і послуг, а й інших актуальних пропозицій для студентів і викладачів закладу.

Дослідження вільного часу студентської молоді показують, що загалом учнівська молодь орієнтована насамперед на розважальне дозвілля, пізнавальна і творча спрямованість вільного проведення часу менш поширена серед студентства. Водночас має місце прагнення молоді до активного позадомашнього розвивального дозвілля, спрямованого на інтелектуальне та фізичне самовдосконалення.

**Мета дипломної роботи:** створити зручний і надійний онлайн-маркетплейс для студентів і викладачів освітніх закладів, зацікавлених роботодавців, що дасть змогу:

- для студентів: отримувати різноманітні послуги у 4 актуальних сферах: додаткової освіти, дозвілля, практико-орієнтованих проектах та сфери зайнятості;

- для викладачів і роботодавців: презентувати корисні матеріали та заходи для студентів, та отримувати зворотній зв'язок.

Якщо з'єднати разом перелічені елементи, а також нові інформаційні технології, плюси маркетплейсів можна отримати чудові результати:

- розширення професійного горизонту молоді за рахунок додаткової освіти;

- урізноманітнення дозвілля та включення до нього заходів актуальної тематики для молоді та інтерактивних форм проведення;

- здобуття студентами нових, зокрема, і соціальних навичок за рахунок участі в організації та проведенні проектів;

- врахування потреб роботодавців;

- зменшення ізольованості від ринку праці;

- формування необхідних якостей фахівців нового типу;



- пропозиція додаткової роботи молодим людям, набуття досвіду, нових компетенцій за рахунок неповної зайнятості.

**Об'єкт дослідження** - сервіс для студентів і викладачів освітнього закладу (ВНЗ).

**Предмет роботи** - процес розроблення web-сервісу, що об'єднує додаткову освіту, дозвілля, проектну діяльність і додаткову роботу.

**Методи дослідження:** реінжиніринг бізнес-процесів роботи маркетплейсів, методи структурного та об'єктно-орієнтованого аналізу і проектування.

Для досягнення поставленої мети необхідно сформулювати та розв'язати такі завдання [17].

1. Провести аналіз наявних рішень.
2. Розробити функціональні вимоги до сервісу.
3. Розробити архітектуру web-додатка.
4. Обрати технології для реалізації
5. Виконати роботу зі створення онлайн-маркетплейса.
6. Написати документацію.
7. Протестувати web-додаток.
8. Продумати, як можна використовувати в освітньому закладі це рішення.

Наукова новизна роботи полягає в розробці такого сервісу, що відповідає основним потребам цільової аудиторії.

Проект розробляється на основі узагальнення наявних теоретичних [11-14] і практичних матеріалів - психолого-педагогічних засад дистанційного

навчання [6-8], форм дозвілля, яким надається перевага [9, 10], форми співробітництва між університетом і бізнесом [15, 16].

Практична значущість роботи полягає в опануванні сучасних бізнес-процесів роботи маркетплейсів, набутті компетенцій web-програмування, вивченні інтересів цільової аудиторії, створенні сучасного сервісу з різноманітними функціями та з'єднанні їх у єдиним агрегатом.

Мета, сформульовані завдання, предмет та об'єкт дослідження визначили структуру та зміст цієї роботи.

Ця робота складається зі вступу, трьох розділів, висновку, бібліографічного списку та додатків.

У вступі обґрунтовано актуальність теми роботи, визначено її мету, на підставі якої сформульовано об'єкт, предмет і завдання дослідження, викладено практичну значущість отриманих результатів.

У першому розділі досліджуються наявні сайти, мобільні, desktop додатки для особистого, корпоративного та навчального спілкування. Проводиться їхній аналіз, плюси та мінуси використання цих рішень. Тут же наводиться огляд цільової аудиторії та актуальних для неї послуг і товарів.

Другий розділ присвячено архітектурі застосунку та опису методів і моделей, необхідних для розв'язання задачі. Обираються інструменти програмування маркетплейса з урахуванням його призначення.

У третьому розділі - реалізація додатка і використані інструменти розробки. А також опис застосування рішення і можливості його розвитку надалі.

У додатках містяться:

1. Документація до web-дodatка.

2. Код програми.

У висновку підведені підсумки виконаної роботи.

# 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Актуальність розроблення веб-додатка маркетплейса

Маркетплейс (торговельний майданчик) надає можливість продавцям розміщувати свої товари і послуги з метою продажу через глобальну мережу Інтернет. Водночас покупці можуть вільно переглядати доступні варіанти, порівнюючи та обираючи найбільш оптимальні продукти або послуги на підставі різних параметрів, включно з ціною, характеристиками та іншими релевантними факторами [1].

Самі маркетплейси нічого не продають, їхні завдання [1,2]:

- Забезпечувати просування товарів.
- Залучати трафік.
- Робити сайт і пошук товарів зручними.
- Збільшувати продажі селерів (постачальників товарів).

Сьогодні маркетплейси є одними з найпопулярніших засобів купівлі та продажу товарів і послуг в Інтернеті. З розвитком Інтернету та електронної комерції з'явилося безліч маркетплейсів, таких як Amazon, Etsy, AliExpress, eBay і багато інших. Вони пропонують широкий вибір товарів і послуг, а також надають зручну систему пошуку та фільтрації. Історія розвитку маркетплейсів показує, що вони стали невід'ємною частиною сучасної електронної комерції і продовжують розвиватися, пропонуючи нові можливості для продавців і покупців.

Основні переваги маркетплейсів [3]:

- велика аудиторія;

- просування товарів;
- оптимізація бізнес-процесів;
- швидкий зворотній зв'язок;
- зручний інтерфейс.

Згідно з доповіддю, представленою компанією Mirakl, маркетплейси є основним рушієм зростання у сфері онлайн-торгівлі. У 2021 році 42% усіх замовлень в Інтернеті було здійснено через ці торгові майданчики. Цікаво, що чим більше часу користувачі проводили в онлайн-покупках, тим частіше вони зверталися до маркетплейсів для своїх покупок. Понад 70% опитаних учасників зазначили, що маркетплейси представляють найбільш зручний спосіб здійснення покупок [1].

Переваги маркетплейсів успішно застосовують сучасні освітні маркетплейси: Edutoria, SkillBox, Coursera, Stepik, Skype.

За прогнозами очікується, що до 2025 року обсяг міжнародного ринку EdTech (технологій в освіті) збільшиться і становитиме від 404 до 434 мільярдів доларів США. Крім того, за підсумками 2030 року прогнозується зростання частки міжнародного ринку онлайн-освіти до 5,6%, що відповідає 585 мільярдам доларів США. Детальна інформація представлена на рисунку 1, наданому в анонімному джерелі [18].

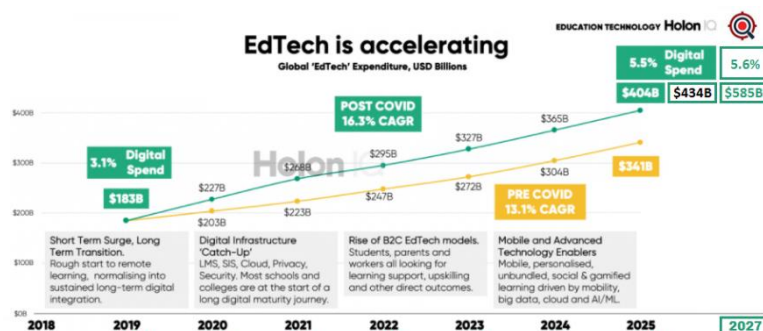


Рисунок 1 - "Освітні технології", HolonIQ, 2022

Сьогодні маркетплейси є не тільки платформою для продажу товарів і послуг, а й засобом обміну досвідом і знаннями між продавцями та покупцями. Деякі маркетплейси створюють спільноти, де користувачі можуть ділитися своїм досвідом, задавати питання та отримувати відповіді від інших учасників.

Висока увага приділяється важливості підготовки висококваліфікованих кадрів і розвитку людських ресурсів, оскільки ці завдання є одними з пріоритетних для забезпечення сталого економічного розвитку країни. Успішне розв'язання цих завдань є невід'ємною умовою для забезпечення стійкості та процвітання.

Вивчення практики європейських країн у сфері співробітництва закладів вищої освіти та працедавців свідчить, що зазвичай співпраця відбувається за кількома напрямками [14]:

1) Максимально раннє залучення і знайомство студентів з виробничим процесом або бізнес-процесом компанії.

2) Участь студентів у роботі проектної (дослідницької) групи на підприємстві як стажера-спостерігача.

3) Участь представників працедавця в прийомі випускних кваліфікаційних робіт і випускних іспитів.

4) Співпраця в науково-дослідницькій та дослідно-конструкторській діяльності компаній. Це R&D (research and development), як це прийнято іменувати на Заході. Тут відкривається величезне поле для співпраці між закладами вищої освіти та бізнесом, оскільки це об'єднання і ресурсної бази, і наукового потенціалу, і кадрового резерву.

5) Створення консалтингового бізнесу, який частково перетинається з R&D, але в більшості випадків має дещо інший напрямок.

6) Створення ендаумент-фондів, завдяки наявності яких заклади вищої освіти можуть фінансувати й підтримувати інноваційні дослідження, найкращих педагогів, науковців, студентів, створювати умови для розвитку інфраструктурної бази університету тощо, що зрештою приносить користь усім учасникам цього процесу - від випускників до бізнесу та держави.

7) Створення на базі університетів технопарків та малих інноваційних підприємств.

Проект, що описується в даній роботі, забезпечує поєднання технології маркетплейсів, освітнього змісту та один із варіантів вирішення завдань професійної освіти.

З усього вище сказаного випливає, що дана розробка має практичну цінність.

## **1.2. Огляд цільової аудиторії та актуальних для неї послуг і товарів**

Цільова аудиторія розробленого онлайн маркетплейса включає в себе 3 групи:

- студенти;
- викладачі;
- роботодавці.

### **Портрет молодої людини, яка навчається**

Сучасні студенти здебільшого приділяють особливу увагу цінностям, пов'язаним із міжособистісними стосунками, роботою та сім'єю.

Згідно з психологічним портретом студента, що ґрунтується на теорії В. Штрауса та Н. Хоува, вони є представниками покоління Z, як показано на рисунку 2. Це підтверджує особливості їхнього світогляду та характеру, що

може впливати на їхні навчальні вподобання та взаємодію соціальним середовищем.



Рисунок 2 - Студенти - представники покоління Z

Для сучасних молодих людей характерні [24]:

- "кліпове" мислення, коли будь-яка інформація засвоюється швидко, але маленькими порціями;
- слабо розвинені комунікативні здібності порівняно з іншими поколіннями, що пояснюється активним віртуальним життям;
- гіперактивність;
- незалежність думок;
- схильність до постійного саморозвитку та здобуття нових знань.

Більшість представників молодого покоління не відчуває пристрасті до тривалих академічних текстів. Однак багато хто з них виявляє інтерес до наукових і технологічних галузей, таких як біомедицина, робототехніка, програмування та екологія. Переважна більшість воліє займатися престижними діяльностями, що не вимагають інтенсивної фізичної праці. У



зumerів можна спостерігати виражене прагнення до досягнення легкого успіху, що відбивається в їхніх цінностях і вподобаннях [11, 24].

Під час залучення студентів до дистанційного навчання як цільової аудиторії необхідно враховувати, що молодь, яка навчається, може діяти в рамках обмеженого бюджету, не просто утримувати їхню увагу і формувати прихильність до певного бренду.

### **Вподобані форми дозвілля студентської молоді [9, 10].**

Різноманітно організоване дозвілля дає змогу молодим людям як збагатити світ внутрішніх переживань, так і отримати новий досвід, спрямований на подальший розвиток особистості. Результати досліджень бажаних форм дозвілля в молодіжному середовищі показали, що найбільш затребуваними є нові практики. Таке "позадомашнє" дозвілля включає в себе розважальні, спортивні, екстремальні, культурні, ігрові компоненти. Розважальні альтернативи дають змогу задовольнити максимальну кількість потреб (матеріальних, духовних і соціальних) сучасної молоді, проте вектор у них зміщений на розвагу.

Молодь, що навчається, сприйнятлива до нового, вирізняється активністю. Самоціннісна характеристика дозвілля пов'язана з тим, що, окрім навчання (роботи), у сучасної людини є бажання не тільки розслабитися (відволіктися), а й розвивати свої творчі можливості та фізичний потенціал.

### **Портрет викладача закладу вищої освіти**

Викладач ВНЗ - це особистість, яка у своїй професійній діяльності повинна володіти комплексом різноманітних універсальних якостей. Він має виявляти навички організатора, вміння виступати перед аудиторією, аналітичну здатність, розуміння психології студентів. Він повинен володіти суворою логікою ведення педагогічного процесу та розвитку, бути майстром

усного та писемного мовлення, а також мати високу компетентність у своїй галузі та ерудицію в інших сферах знань [20].

Від викладача потрібні не тільки природні здібності, а бажано й талант, величезні розумові, фізичні, часові та емоційно-вольові зусилля [21].

Особливу важливість має комунікативна функція, яку виконує викладач під час своєї діяльності, оскільки саме навички комунікації мають найбільший вплив на формування образу викладача вишу. Якісна взаємодія та вміння ефективно спілкуватися з учнями відіграють першорядну роль у створенні позитивного враження про педагога в університеті.

Суттєвого значення у здійсненні педагогічної роботи в університеті набуває орієнтація особистості викладача вищого навчального закладу. Саме вона є ключовим чинником, що визначає вибір стосунків та активності викладача. У структурі особистості викладача особливу значущість має професійно-педагогічна спрямованість, яка визначає його дії та вчинки [20].

Однією з найважливіших якостей є професійна компетентність викладача, яка виступає умовою становлення та розвитку його педагогічної майстерності і становить зміст педагогічної культури [22].

Важливо зазначити, що наявна висока вимогливість до викладача вищого навчального закладу та якості його роботи є основним чинником, який стимулює прагнення до самовдосконалення. Ефективне виконання професійних обов'язків викладача буде забезпечено, якщо він зуміє усвідомлено використовувати свій потенціал і застосовувати його відповідним чином [20].

Дослідження взаємодії викладачів і студентів в інформаційному освітньому середовищі ВНЗ засвідчили, що ефективність цієї взаємодії залежить не тільки від уміння та можливостей викладачів використовувати

електронний освітній процес, а й від здатності до адаптації та бажання навчатися у студентів [20, 21].

### **Експерти склали портрет ідеального роботодавця**

Під час опитування, проведеного експертами серед співробітників різних компаній, було виявлено, яким роботодавцем вони вважають ідеального. Не дивно, що співробітники пріоритетно оцінюють керівників, здатних забезпечити стабільність і можливості для особистого розвитку. Результати опитування показали, що ідеальним керівником вважається той, хто може надати такі переваги [24]:

75% опитаних вказали на високу заробітну плату;

63% відзначили хороший соціальний пакет;

51% вважають стабільність у компанії важливим фактором;

45% пріоритетно оцінюють можливості для кар'єрного зростання;

29% виділили дружні стосунки в колективі;

25% важливими факторами є можливості для професійного розвитку.

Особливо для молодих людей віком від 18 до 24 років важливим є професійне зростання на робочому місці, а також пропоновані роботодавцем навчальні програми для співробітників. Фактори, такі як розмір компанії, розвинена корпоративна культура, статус роботи і рівень повноважень, мають найменший вплив під час оцінки роботодавця співробітниками.

### **Система взаємодії ВНЗ та роботодавців у підготовці майбутніх фахівців**

Протягом часу накопичилася значна кількість способів взаємодії між вишами та підприємствами. Більшість вищих навчальних закладів мають великий список партнерських підприємств, а деякі компанії вже активно

шукають шляхи спільної роботи з вищими навчальними закладами, з метою скорочення розриву між теорією і практикою [13].

Можливо виділити такі цінності взаємодії (співпраці) для ключових суб'єктів (таб. 1) [16, 17].

Таблиця 1 - Деякі цінності взаємодії вишу і компанії.

Для студентів	Для ВНЗ	Для роботодавця
1) отримання практичного досвіду; 2) розширення кругозору; 3) розширення кола спілкування та можливостей; 4) участь у реальних проєктах; 5) підвищення мотивації до опанування знань і навчального процесу; 6) формування додаткових компетенцій; тощо.	1) підвищення якості освіти; 2) практико-орієнтованість навчального процесу; 3) підвищення якості випускників; 4) отримання додаткового фінансування; 5) формування додаткової мотивації у студентів до освітнього процесу та його якісного освоєння; 6) верифікація практикою тих знань, які даються студенту в процесі навчання; тощо.	1) можливість селективного багатоступеневого і довгострокового відбору випускників, 2) підвищення продуктивності праці у роботодавця за рахунок залучення найкращих кадрів; 3) доступ до науково-методичної та дослідницької інформації та методичної агрегованої бази ВНЗ; 4) апробація новацій і нововведень, у тому числі в порядку розробки симуляторів і студентських проєктних груп; 5) формування бренд-лояльності як майбутніх працівників, так і майбутніх "споживачів" і контрагентів;

		б) зниження трансакційних витрат на комунікацію з майбутніми здобувачами, контрагентами, партнерами тощо.
--	--	---

Відповідно до поставлених завдань на систему взаємодії ВНЗ і роботодавця можуть бути реалізовані через такі заходи:

- організацію та проведення авторських навчальних занять;
- організацію та проведення мастер-класів і групових консультацій зі студентами;
- розроблення нових навчальних програм і навчально-методичних матеріалів;
- проведення заходів з професійної орієнтації студентів;
- залучення студентів до виконання завдань за конкретними проектами компанії.

Отже, у центрі розгляду цільової аудиторії - студенти, саме на них спрямований освітній контент створюваного маркетплейсу, практична спрямованість заходів, презентованих на сайті. Для викладачів маркетплейс - це майданчик для підвищення ефективності навчання, зворотного зв'язку, поліпшення комунікацій зі студентами та роботодавцями, отримання учнями практичних навичок, розширення кругозору студентів. Для роботодавців сайт маркетплейс - інструмент взаємодії зі студентами, можливість розташовувати корисні матеріали, навчальні курси, просувати спільні проекти вишу та підприємства/компанії, пропонувати вакансії та обирати претендентів.

### 1.3 Огляд і порівняння наявних пропозицій

Маркетплейс - це торговельний майданчик або онлайн-платформа, на якій здійснюються купівля та продаж товарів та послуг [4, 5].

Маркетплейс можна порівняти з великим ринком, де на одному прилавку можна знайти одні й ті самі продукти від різних брендів і марок. На маркетплейсах покупці шукають потрібні для них товари і послуги та обирають найкращі варіанти за важливими для них параметрами, наприклад, виробник, характеристики, якість, ціна тощо. Важливо зазначити, що самі маркетплейси здебільшого є лише сполучною ланкою між виробником і споживачем.

Існують дві суттєво різні моделі продажів в онлайн-сфері: інтернет-магазин і маркетплейс. Головна відмінність полягає в тому, що інтернет-магазин прагне просувати свій власний бізнес, і всі його зусилля спрямовані на досягнення цієї мети. З іншого боку, маркетплейс не переслідує інтереси конкретної компанії або бренду, його основний пріоритет полягає у збільшенні обсягу продажів на своїй платформі. Маркетплейс прагне надати максимальну кількість можливостей для продажу товарів і послуг, залучаючи якомога більше продавців і покупців [27].

Маркетплейс, що найчастіше виступає посередником між продавцем і покупцем, зазвичай бере на себе різноманітні зобов'язання та функції, пов'язані з обслуговуванням веб-сайту та мобільного додатку магазину. Вона інвестує значні ресурси, як час, так і гроші, у залучення потенційних клієнтів. Крім того, пропонується безпечний процес купівлі та оплати, а також організація доставки та зберігання товарів на власному складі. Платформа також займається вирішенням різноманітних питань, пов'язаних з обробкою замовлень, поверненнями та обмінами товарів, забезпечуючи зручність і надійність для всіх сторін [24].

Однією з головних привабливих особливостей маркетплейса для продавця є його зручний інструмент продажів, який забезпечує приплив клієнтів і сприяє розширенню їхньої бази. З іншого боку, для покупця маркетплейс представляє цінність у можливості придбати все необхідне в одному місці, що дає змогу заощадити значну кількість часу та зусиль. Завдяки широкому асортименту товарів і зручній платформі, маркетплейс забезпечує зручність і впевненість покупцям під час здійснення покупок.

Маркетплейси розрізняються за декількома критеріями:

### **Тип товару / послуги.**

Наприклад, освітні послуги надає Coursera, фінансові послуги - Axilog, товари - Amazon, інвестиції - Kickstarter. В таблиці 2 представлені деякі маркетплейси та їхні товари.

Таблиця 2 - Товари різних маркетплейсів.

<b>Маркетплейс</b>	<b>Категорії товарів</b>
Etsy (США)	HANDMADE товари, інструменти та матеріали для творчості, одяг, аксесуари
Allegro (Польща)	Електроніка, дім, сад та одяг
Amazon (США)	Товари всіх категорій

### **Бізнес-модель.**

Хто є продавцем, а хто покупцем? Тут можна запропонувати класичні категорії: B2B - бізнес продає бізнесу (китайський онлайн-майданчик Alibaba); B2C - бізнес продає звичайним людям (великий польський маркетплейс Allegro); C2C - товари та послуги пропонують не підприємці (наприклад, на Etsy.com продавцями виступають звичайні люди, які виготовляють товари на продаж своїми руками).

### **Тематика.**

На сьогодні існують два типи маркетплейсів, які розрізняються за своїм асортиментом товарів і цільовою аудиторією. Перший тип - мультитоварні майданчики, як-от Amazon US, які пропонують широкий спектр товарів повсякденного попиту, включно з прикрасами, одягом, побутовою хімією, книжками та продуктами харчування. Ці маркетплейси націлені на обслуговування масової аудиторії і прагнуть задовольнити різноманітні потреби покупців.

Другий тип - нішеві маркетплейси, наприклад, ManoMano в Європі, які спеціалізуються на конкретних категоріях товарів, у даному випадку - товари для дому. Такі майданчики пропонують спеціалізований асортимент продукції, який відповідає конкретним потребам і перевагам цільової аудиторії. Вони орієнтовані на вузьку нішу ринку і прагнуть надати спеціалізований вибір товарів і послуг для своїх клієнтів [37].

### **Спосіб комунікації з покупцем:**

Існують дві моделі маркетплейсів, які пропонують різні способи взаємодії між продавцями і покупцями. Перша модель - O2O (online to offline), де клієнти знаходять товари або послуги на маркетплейсі в онлайн-режимі, але отримують їх у фізичному пункті видачі. Прикладом такого підходу є Amazon, де покупці можуть замовити товари онлайн і забрати їх у найближчому пункті видачі.

Друга модель - онлайн-торгівля, де весь процес здійснення покупки відбувається виключно в онлайн-режимі. Прикладом такого маркетплейсу є Envato Tits+, який пропонує споживачам широкий вибір творчих продуктів і послуг. Покупці можуть опанувати різноманітні креативні навички та придбати відповідні ресурси прямо на платформі без необхідності фізичного відвідування магазину чи пункту видачі.

### **Монетизація.**



У маркетплейса є кілька способів монетизації своєї діяльності та генерації доходу. Він може отримувати прибуток від різних джерел, як-от трафік, ліди (заявки), дії користувачів, відсоток з продажів товарів і послуг, доставка, реклама та надання додаткових послуг.

Маркетплейсу приносить дохід залучення великої кількості користувачів, які відвідують і використовують його платформу. Крім того, маркетплейс може вигідно продавати ліди, тобто інформацію про зацікавлених клієнтів, яка може бути передана постачальникам товарів або послуг.

Додаткове джерело доходу для маркетплейса - це комісійні з продажів, які він стягує з кожної успішної угоди між продавцем і покупцем. Крім того, маркетплейс може пропонувати додаткові послуги, як-от реклама або поліпшені можливості продажу, за які також стягується плата.

Ще один спосіб монетизації - це співпраця з доставкою, де маркетплейс може отримувати комісію з кожної доставки товарів або послуг.

Загалом, маркетплейс має кілька джерел доходу, що дає йому змогу ефективно функціонувати та надавати цінні послуги як для продавців, так і для покупців.

Серед популярних міжнародних маркетплейсів: Amazon, Ebay, Etsy, Aliexpress, Asos тощо.

### **Amazon** (рис. 3)

Один із найбільших міжнародних маркетплейсів, який існує вже понад 30 років. Щоденна аудиторія в розмірі 28 млн. осіб сприяє високому товарообігу.

Питання, як продавати на Амазон, може турбувати підприємців через відсутність перекладу сайту українською мовою. Проте цей недолік компенсується широким охопленням аудиторії, продуманою логістикою і надійністю платформи - правила використання майданчика прозорі, а виплати продавцеві надходять вчасно.

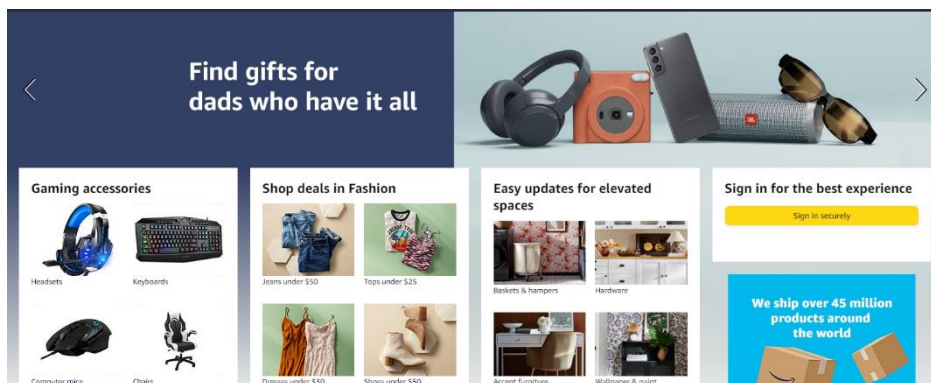


Рисунок 3 - Маркетплейс Amazon

Торгівля на маркетплейсі не вимагає великих інвестицій. Амазон бере на себе зберігання товарів і логістику. Вийти на міжнародний ринок можуть навіть невеликі компанії: для цього не потрібно витратитися на оренду офісу або складу.

Механізми просування, що працюють на цьому майданчику, не вимагають додаткових витрат на рекламу, а наявність великої кількості доступних категорій товарів дає змогу торгувати навіть нестандартною продукцією.

### **ЕВау (рис. 4)**

ЕВау - другий за розміром аудиторії маркетплейс за кордоном, який від початку працював як онлайн-аукціон. Зараз більшість товарів мають фіксовану вартість, але покупці все ще можуть пропонувати свою ціну.

Офіційний сайт маркетплейса заявляє про 135 млн активних покупців, що дає підприємцям доступ до широкої аудиторії по всьому світу. Крім того, компанія пропонує продавцям-початківцям можливість навчання в eBay Academy.



Рисунок 4 - Маркетплейс ЕВау

Реєстрація бізнес-акаунта на цьому маркетплейсі безкоштовна, як і розміщення оголошень. Продавець платить комісію тільки в разі продажу товарів. Проте для підприємців-початківців є обмеження: не більше 10 товарів щомісяця на суму не більше 500 доларів.

За 2022 рік обсяги продажів на великих маркетплейсах зросли у 2,7 рази. Є всі шанси, що у 2023 році ця тенденція збережеться.

Обсяг social commerce (або соціальної комерції) зростає з року в рік. За дослідженням Shopify, стріми, короткі відео з розпакуваннями та оглядами товарів дедалі сильніше впливають на рішення про покупку і допомагають залучати нову аудиторію.

Маркетплейси продовжать переймати механіки соціальних медіа та нарощувати аудиторію. Щоб використовувати це на свою користь, варто вже зараз розвиватися в напрямку залучального і брендового контенту:

створювати на майданчиках свої блоги, знімати відео з товарами і налагоджувати зв'язки з інфлюенсерами.

### **Найцікавіші міжнародні освітні платформи**

У міру розвитку онлайн-технологій особливо зростаючу затребуваність показують послуги з дистанційного навчання.

Нині за допомогою онлайн-курсів кожна людина може отримати можливість віртуального навчання в престижних університетах і ділових школах, прослуховувати лекції та проходити майстер-класи в найуспішніших професіоналів своєї галузі з різних країн. Крім того, вона також має можливість покращити свої навички в англійській мові, що є додатковою перевагою. Таким чином, онлайн-курси надають широкий доступ до освіти та професійного розвитку, даючи змогу людям розширити свої знання та навички незалежно від місця проживання та географічних обмежень.

Онлайн-освіта - перспективна галузь, у якій з'являються все нові платформи.

#### **Coursera (рис. 5)**

«Coursera робить найкращу у світі освіту доступною кожному, пропонуючи онлайн-курси від провідних університетів та організацій».

Курси на Coursera найбільше нагадують академічну освіту.

Те, що пропонує Coursera - це унікальний продукт і нестандартний контент. Наприклад, до деяких курсів, крім звичайних лекцій від професорів, додаються також записані півгодинні інтерв'ю з авторами тих книжок, що були в списку рекомендованої до курсу літератури.

Курси на Coursera проходити легше тим, у кого проблеми із самодисципліною. Наявність чітких часових рамок дисциплінує і стимулює.

Coursera - це можливість отримати якісну університетську онлайн-освіту.

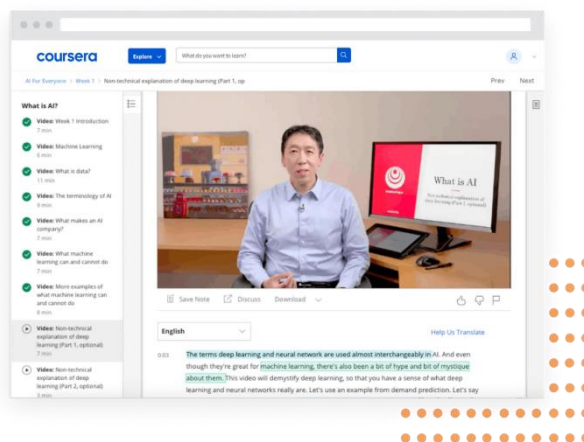


Рисунок 5 - Маркетплейс освітніх послуг Coursera. Відгуки про курси

## Udemy

«Будьте у всеозброєнні. Від програмування до фотографії - пройдіть поглиблений онлайн-курс і отримайте навички для вирішення будь-яких завдань».

Udemy являє собою глобальний майданчик для навчання і навчання онлайн, де студенти опановують нові навички та досягають своїх цілей навчання з великою бібліотекою з більш ніж 55000 курсів, які викладають досвідчені фахівці.

## SkillShare

«Заняття ведуть практикуючі фахівці експерти. Для вашої кар'єри. Для вашого захоплення».

Skillshare - навчальна онлайн-спільнота, де кожен може відкривати нове, отримувати знання, або навіть викладати.

Понад 2 млн студентів.

Понад 17 тисяч уроків.

Будь-який навчальний заклад може зайнятися просуванням своїх курсів (або себе загалом) за допомогою відразу декількох ефективних онлайн-інструментів. У ряді випадків у найоперативніші терміни збільшити прибуток установи за рахунок припливу нових клієнтів.

## **1.4 Висновок за розділом 1**

Маркетплейс (торговельний майданчик) - це онлайн-платформа для продажу та купівлі товарів і послуг через Інтернет.

Онлайн-маркетплейси освітніх послуг існують у світі досить давно. Є майданчики, де пропозиції виставляють спеціалізовані навчальні заклади, а є ті, що створені для фізичних осіб, які надають освітні послуги в приватному порядку.

Існує спеціальна категорія маркетплейсів освітніх послуг - платформи, на яких той чи інший навчальний заклад або розробник продукту, використовуюваного в навчальному процесі, можуть, безплатно розмістивши ті чи інші корисні матеріали, просунути свій бренд, поліпшити його сприйняття в середовищі цільових клієнтів.

Цільовою аудиторією розроблюваного освітнього маркетплейса є: студенти, викладачі та роботодавці.

Освітній контент призначений для студентів. Викладачі та роботодавці можуть розміщувати картки послуг для студентів (онлайн курси, запрошення на участь у проектах, описи виробничої практики, вакансії та інше).

Слід враховувати якості цільової аудиторії, їхні функції та особливості взаємодії одна з одною.

## **2. РОЗРОБКА МОДЕЛІ ДОДАТКА**

### **2.1 Постановка завдання**

#### **Найменування системи**

«ОДІН» - зручний і надійний онлайн-маркетплейс для просування освітніх послуг, послуг з організації дозвілля, проектної діяльності студентів, взаємозв'язку учнів з викладачами, роботодавцями вищого професійного закладу.

#### **Цілі створення та призначення системи:**

Організація взаємодії студентів, викладачів і роботодавців для розвитку практичних знань і навичок випускників, формування залученості студентів у процес навчання.

#### **Цільова аудиторія:**

- студенти;
- викладачі;
- роботодавці.

#### **Завдання роботи:**

- запрограмувати онлайн-маркетплейс у вигляді сайту;
- написати супровідні документи;
- продумати, як можна використовувати в освітньому закладі це рішення.

#### **Очікувані результати:**

- розширення професійного горизонту молоді за рахунок додаткової освіти;

- урізноманітнення дозвілля та включення до нього заходів актуальної тематики для молоді та інтерактивних форм проведення;
- здобуття студентами нових, зокрема, і соціальних навичок за рахунок участі в організації та проведенні проектів;
- врахування потреб роботодавців;
- зменшення ізоляваності від ринку праці;
- формування необхідних якостей фахівців нового типу;
- пропозиція додаткової роботи молодим людям, набуття досвіду, нових компетенцій за рахунок неповної зайнятості.

Назва проекту походить від імені верховного бога - **Одін**, який у скандинавській міфології науками відав і опікувався навчанням.

**Функціями** маркетплейса можуть бути [20]:

- залучення клієнтів на сайт;
- просування товарів і послуг;
- зручний інтерфейс;
- комфортна схема оформлення замовлення;
- вітрина;
- швидкий пошук товару і послуги в списку, організація фільтрів;
- формування Кошика покупок;
- доступ до Особистого кабінету, збереження історії;
- 4 розділи товарів: освіта, дозвілля, інсайт, найм;
- організація зворотного зв'язку і виставлення оцінок товарам і послугам;
- оплата товару/послуги; отримання доступу до матеріалів.



**Обмеження:** рекомендується використовувати будь-які браузері, крім Internet Explorer.

Додаток призначений для використання переваг сучасних онлайн-технологій з можливістю відкрити web-додаток з будь-якого пристрою.

### **Рамки підходу до змісту сайту**

Система взаємодії роботодавців і представників сфери освіти є важливою умовою якості професійної підготовки фахівців. Здебільшого роботодавці задоволені теоретичною підготовкою випускників ВНЗ, але більшість компаній відзначають, що не вистачає практичного досвіду.

Причину цього вони вбачають у тому, що часто компанії та підприємства, які працюють в умовах ринкової економіки, раніше за виші освоюють новітні технології.

Професор Тодд Дейві виокремлює такі основні напрями (завдання) розвитку співробітництва вишів і роботодавців (СУБ) за ключовими учасниками цієї співпраці (див. табл. 3).

Таблиця 3 - Напрями співпраці ВНЗ і роботодавців.

<b>ВНЗ</b>	<b>Вчені</b>	<b>Бізнес</b>	<b>Суспільство</b>
1. Створити «стратегії реалізації» (наприклад, СУБ у рамках академічного оцінювання).	1. Шукати можливості та запрошувати бізнес до партнерства в дослідницьких проектах. 2. Налагоджувати контакти з	1. Налагоджувати контакти в академічних колах. 2. Прагнути краще зрозуміти мотивацію науковців і університетів.	1. Підвищити фінансування для усунення бар'єрів для співпраці університетів і бізнесу. 2. Підтримувати програми, що

2. Скоротити бюрократію у ВНЗ.	бізнесом, наприклад, через	3 Шукати партнерів за проектами або	фінансують дослідження
3. Більше уваги приділяти розвитку відносин (особливо довіри) між науковцями та бізнесом, оскільки це сприяє СУБ.	студентські стажування. 3.Взаємодіяти з бізнесом у процесі розробки навчальних програм	надавати фінансування для співпраці з ВНЗ. 4.Підтримувати контакти з тими, хто працював в академічних колах, щоб мати зв'язок з науковцями	
4. Заохочувати СУБ (грошово і не грошово)			

Існують певні функції, які виконують усі західні моделі дозвілля, включаючи фізіологічну, психологічну, пізнавальну, естетичну, комунікативну та компенсаторно-творчу.

Соціальна спрямованість може бути притаманна навіть суто розважальним заходам. Дозволяючи людям розважитися й активно провести час, такі заходи сприяють відновленню їхніх духовних і фізичних сил, що має велике значення як для окремої людини, так і для суспільства загалом.

Соціалізаційний процес індивіда присутній навіть у винятково розважальних заходах. Наприклад, беручи участь в іграх чи атракціонах, що потребують спритності, сміливості та інших аналогічних навичок, людина

відчуває себе рівною іншим, немовби вона стає частиною кращих або виявляє солідарність із широкою аудиторією співвітчизників.

Пріоритетні завдання організації позааудиторної діяльності студентів:

- 1) Студент як член правової держави. Громадянськість.
- 2) Студент як потенційний професійно-компетентний фахівець (здебільшого в галузі управління або політики).
- 3) Студент як член суспільства, навички життя в соціумі, роботи з колективом.
- 4) Студент як член багатонаціонального суспільства.
- 5) Студент як потенційний керівний працівник.
- 6) Виховання в студентах духовності, співчуття.
- 7) Студент як майбутній учений.
- 8) Естетичне виховання студентів, реалізація їхнього творчого потенціалу.
- 9) Підтримка студента, адаптація до університетського життя.
- 10) Організація дозвілля студентів
- 11) Організація побуту студента

Формами організації дозвілля можуть бути:

- 1) Спортивні клуби, команди, організація змагань, створення умов для спорту, відпочинку (секції, тренажерні та спортивні зали тощо).
- 2) Музичні (оркестр, курси з освоєння муз. інструментів, класи вокалу, хори).

3) Театр (трупі, класи акторської майстерності, запрошення на сцену студентського театру відомих труп).

4) Художні (живопис, малюнок, скульптура, прикладне мистецтво, виставки).

5) Організація "пасивного" відпочинку студентів - бари, ресторани, кафе, організація тематичних вечірок.

6) Туристичне бюро.

Як видно організація дозвілля лише частина пріоритетних завдань організації позааудиторної діяльності студентів. Можна виокремити такі форми реалізації позааудиторних заходів:

- служби юридичної допомоги студентам (які консультують з питань їхніх прав);
- вузькоспеціалізовані курси в рамках навчального плану (наприклад, для майбутніх спортивних журналістів - "історія футболу");
- дискусійні клуби (навички політичної боротьби, вчать відстоювати й аргументувати свою точку зору);
- публіцистичні студентські спільноти;
- бюро працевлаштування;
- "Школи лідерства";
- професійно-орієнтовані тренінги та інші.

### **Зміст матеріалів сайту**

Назву створюваного сайту «ОДІН» можна розглядати і як аббревіатуру, яка розшифровується як: освіта, дозвілля, інсайт, найм.

Опис змістовних розділів онлайн-маркетплейсу, цілепокладання для студентів та можливі варіанти матеріалів міститься в таблиці 4. При цьому авторами матеріалів можуть стати: студенти, викладачі, роботодавці (та їхні представники). Заходи можуть бути як очними, так і дистанційними, як платними, так і безкоштовними. Такий підхід до створення матеріалів для маркетплейсу слугує для залучення до викладання фахівців, які можуть не мати наукового ступеня та вченого звання, а також стажу науково-педагогічної роботи, але мають достатній практичний досвід за напрямом професійної діяльності, що відповідає профілю вишу.

Таблиця 4 - Змістовні розділи онлайн-маркетплейса.

<b>Розділ</b>	<b>Мета для студента</b>	<b>Варіанти продуктів і послуг</b>
Освіта	Отримувати додаткові знання та вміння і навички	Навчальні заходи в інтерактивній формі проведення, наприклад: <ul style="list-style-type: none"> <li>- курси додаткової освіти;</li> <li>- консультації (студенти студентам);</li> <li>- цікаві вебінари;</li> <li>- зустрічі з відомими людьми;</li> <li>- ділові ігри;</li> <li>- конкурси.</li> </ul>
Дозвілля	Духовно, фізично розвиватися та повноцінно відпочивати	Розважально-розвивальні заходи, наприклад: <ul style="list-style-type: none"> <li>- соціальний театр;</li> <li>- виставки;</li> <li>- фестивалі;</li> <li>- шоу-кейси;</li> <li>- квести;</li> <li>- дебати;</li> <li>- спортивні заходи.</li> </ul>
Інсайд	Проявляти інтелект, творчість, підприємництво	Особливі великі проекти та практичні корисні поради, наприклад: <ul style="list-style-type: none"> <li>- практико-орієнтовані проекти (соціальні, наукові, навчальні) за участю роботодавців;</li> <li>- хакатони;</li> </ul>

		<ul style="list-style-type: none"> <li>- конкурс бізнес-ідей;</li> <li>- лайфхаки;</li> <li>- чек-листи;</li> <li>- кейси-завдання.</li> </ul>
Найм	Отримувати нові компетенції	<p>Пропозиції роботи від компаній, приватних осіб, навчальних закладів. Рекомендації для успішної кар'єри, наприклад:</p> <ul style="list-style-type: none"> <li>- корисні поради щодо написання резюме, побудови кар'єри, поведінки на співбесіді тощо;</li> <li>- співбесіди з роботодавцем;</li> <li>- волонтерська робота;</li> <li>- вакансії з неповною зайнятістю, віддаленою роботою.</li> </ul>

Способів взаємодії ВНЗ і підприємств чимало. У даній роботі пропонується створити такий майданчик для співпраці студентів, викладачів ВНЗ з одного боку, роботодавців та їхніх представників з іншого боку, який за допомогою нових інформаційних технологій, маркетингових підходів, інтерактивних форм проведення заходів може оживити такий взаємозв'язок.

На вітрині сайту можуть відображатися такі картки послуг як організація та проведення:

- авторських навчальних занять і курсів;
- майстер-класів та групових консультацій зі студентами;
- заходів з професійної орієнтації студентів;
- мозкових штурмів, хакатонів та інших заходів у рамках конкретних проектів компаній.

Підприємство і вищий навчальний заклад є двома сторонами освітнього процесу. Тому саме від ефективності зворотного зв'язку між ними залежить

ступінь відповідності якості підготовки спеціалістів побажанням роботодавця, а отже, і працевлаштування випускників вишу [22,23].

Результативність такого взаємообміну забезпечить:

- підвищення ступеня прояву показників професійної діяльності;
- швидку адаптацію до мінливих умов роботи;
- підвищення комунікативної компетентності;
- знайомство з методами пошуку роботи, правилами поведінки в діловому середовищі;
- покращення навичок публічної презентації себе та результатів своєї праці;
- формування вмінь і навичок роботи в команді.

На сайті маркетплейса розміщуються картки пропонованих товарів і послуг. Після покупки відкривається доступ до самих матеріалів (файлів, посилань, документів, квитків тощо). Рекомендовані заходи (семінари, відповіді на актуальні питання, співбесіди тощо) можна здійснювати в очній та дистанційній формах. Дистанційні технології мають багато оптимальних властивостей, при цьому звільняючи учня від жорсткого регламенту навчального процесу.

Одна з важливих функцій маркетплейса - залучення клієнтів.

### **Як залучити клієнтів у маркетплейс?**

Для цього використовується контент, розміщення статей у блогах, форумах, написання гостьових постів і використання реферальних посилань. Необхідно створювати навколо свого маркетплейсу інформаційну активність, формувати бренд, щоб зацікавити аудиторію і легше залучити продавців [5, 29].

## **Оформлення картки товару на маркетплейсах**

Завдання, які вирішуються карткою товару:

- привернення уваги аудиторії;
- посилення бажання покупця володіти продуктом;
- спонукати до цільової дії - покупки;
- відповіді на запитання і зняття заперечень.

Якість оформлення картки товару - один із найбільших чинників, що впливають на просування продукту на маркетплейсах. Для досягнення підвищення конверсії та обсягу продажів потрібна постійна оптимізація картки товару, включно з проведенням тестування фотографій, внесенням змін до описів товарів і роботою з характеристиками. Важливо пам'ятати, що якщо картка товару перебуватиме в топі пошуку, то на неї часто заглядатимуть і купуватимуть товари, що згодом ще більше просуватиме товар угору.

Найбільш важливим фактором є відповідність картки товару запиту користувача. Для досягнення цього варто ретельно опрацювати семантичне ядро і включити в нього відповідні ключові слова. Також необхідно надати всю доступну інформацію про товар. Не слід обмежуватися лише характеристиками – треба запропонувати покупцеві різні способи використання продукту, його детальний опис, однак треба пам'ятати, що опис має бути лаконічним і зрозумілим для користувача.

Унікальні та високоякісні фотографії також сприяють підвищенню видимості картки товару. Крім зображень, можна також додати відеоролик, що напевно приверне увагу користувачів, даючи їм змогу розглянути товар з різних ракурсів.



Завдяки переліченим вище підходам створений web-додаток «ОДН» є майданчиком для розв'язання питань розширення форм взаємозв'язку роботодавців та освітнього закладу через застосування технологій маркетплейсів, дистанційного навчання, просування освітніх продуктів і послуг, створення матеріалів усіма учасниками процесу (студентами, викладачами, роботодавцями).

### **Питання безпеки**

Для розв'язання питань безпеки створеного продукту виділено такі аспекти: безпека пароля (метод хешування), використання протоколу https, парольна аутентифікація.

### **Вимоги до додатка**

#### ***Вимоги до функціональних характеристик***

Додаток має забезпечувати можливість виконання перелічених нижче функцій:

- 1) Реєстрація покупця, продавця (постачальника освітніх товарів і послуг).
- 2) Авторизація покупця, продавця та адміністратора.
- 3) Організація вітрини товарів і послуг за розділами (навчання, дозвілля, інсайт, найм).
- 4) Швидкий пошук і фільтрація.
- 5) Ведення Особистого кабінету.
- 6) Перегляд і редагування Кошика.
- 7) Відправлення та перегляд відгуків, оцінка якості товару/послуги.
- 8) Розміщення нового матеріалу продавцем.

9) Перевірка матеріалу на відповідність вимогам.

10) Перегляд доходу продавцем.

11) Функціонування панелі адміністратора (робота зі списками користувачів, перевірка, підтвердження та розміщення на сайті нового матеріалу).

### ***Вимоги до інтерфейсу***

Мають бути реалізовані такі вимоги:

- інтерфейс системи має бути інтуїтивно зрозумілим;
- система має бути однаково зручною і для просунутого, і для користувача-початківця ПК;
- кольорова гама, шрифти та ін. мають забезпечувати зручність читання, наочність інформації для користувачів.

### ***Вимога до програмного забезпечення***

Організація цієї мережі як Web-сайту значною мірою забезпечить популярність і адаптивність ресурсу до різних пристроїв і операційних систем.

Опис екранів програми за макетами [33]

На початковому етапі треба визначитися, що має бачити користувач під час роботи з web-додатком.

Макет у дизайнерському сенсі - це наочне представлення дизайну майбутнього застосунку. На цьому етапі макет має бути максимально наближеним до реальності, щоб можна було зрозуміти, який вигляд має додаток і як він працює.

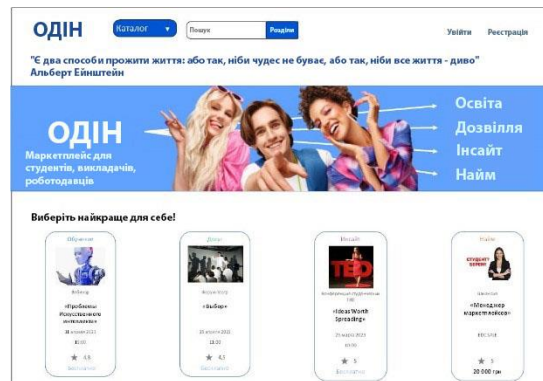


Рисунок 6 - Макет головної сторінки

Макет головної сторінки сайту, яка вітає споживача і занурює в тематику (рис. 6).

Далі представлено проекти декількох сторінок Особистого кабінету покупця, тобто споживача послуг (рис. 7).



Рисунок 7 - Макети сторінок Особистого кабінету покупця

Далі представлено проекти кількох сторінок Особистого кабінету продавця, тобто того, хто надає послуги (рис. 8).

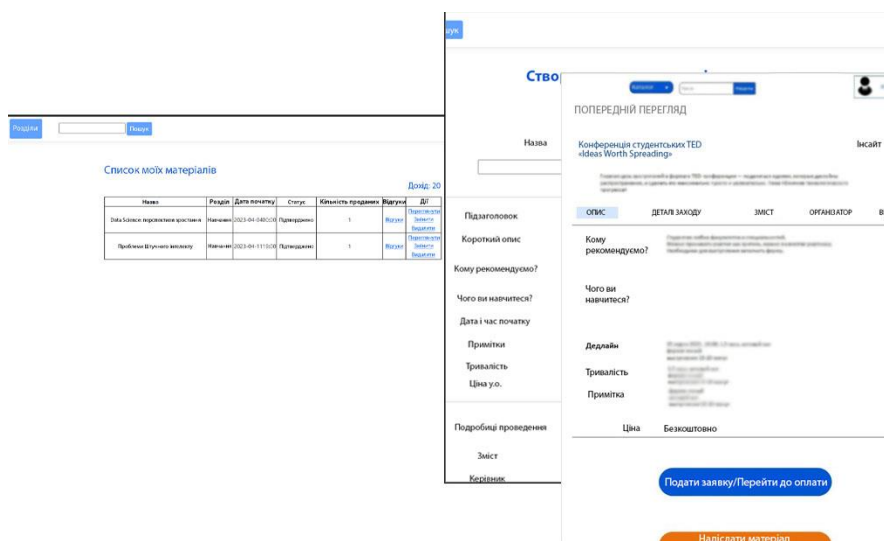


Рисунок 8 - Макети сторінок Особистого кабінету продавця

На рисунку 9 можна побачити макет сторінки пошуку та фільтрації матеріалів на вітрині послуг і товарів.

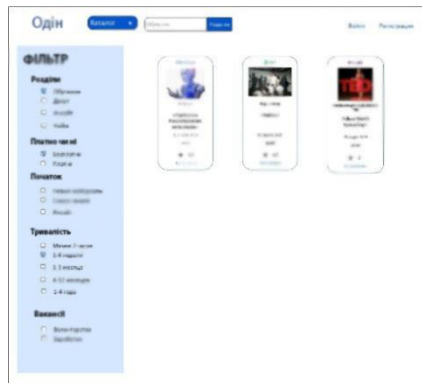


Рисунок 9 - Макет сторінки пошуку та фільтрації

Підсумовуючи, слід зазначити, що маркетплейс - це сучасна форма презентації, просування товарів і послуг, у тому числі й освітніх. Застосування переваг маркетплейсів для презентації, привернення уваги споживачів двох сторін (ВНЗ та бізнесу) для онлайн-взаємодії цих сторін.

### **Склад робіт зі створення системи:**

#### 1) Проектування додатка.

- a) Опрацювання структури (де і що буде знаходитися).
- b) Підбір колірних рішень для інтерфейсу.
- c) Формування вимог до розроблюваної системи.
- d) Вибір інструментів реалізації проекту.
- e) Опис структури даних.

#### 2) Реалізація рішення.

- a) Опис алгоритмів роботи.
- b) Програмування модулів.
- c) Пробні випробування. Виправлення помилок (на кожному етапі реалізації окремих модулів).
- d) Написання супровідних документів.

#### 3) Формування планів і перспектив на майбутнє.

## 2.2 Вибір інструментарію програмування

Для розробки ПЗ необхідно вибрати мову програмування, що підтримує методологію об'єктно-орієнтованого програмування. Завдяки особливостям ООП спроектоване ПЗ матиме гнучку архітектуру, що легко розширюється. Як основний засіб розроблення ПЗ було обрано мову C#, оскільки ця мова підтримує всі можливості ООП і має у своєму арсеналі безліч зручних засобів для роботи зі сховищами даних.

Як основний інструмент розробки буде використовуватися Microsoft Visual Studio [26], оскільки це середовище розробки володіє всіма засобами для максимально зручної розробки на мові C# [27].

Далі належить спроектувати базу даних. Як СУБД було обрано Microsoft SQL. Ця СУБД має всі можливості роботи з реляційними базами даних і більше того, завдяки технології Entity Framework, має відмінну сумісність із мовою програмування C#. Завдяки чому можливо створити гнучкий додаток, що працює з базою даних.

Для клієнтського додатка буде спроектовано графічний інтерфейс, що дає змогу користувачеві легко орієнтуватися у функціях програми. Інтерфейс реалізовано за допомогою шаблонів ASP.NET, HTML, CSS, Java Script [28, 29].

ASP.NET MVC - надає можливість створення клієнт-серверного додатка; має вбудовану підтримку шаблону проектування MVC, що дає змогу розмежувати частини додатка для підвищення зручності розробки; використовує мову програмування C#, що дає змогу писати ефективний і безпечний код [27].

Далі коротко будуть описані всі названі вище інструменти.

**Об'єктно-орієнтоване програмування. C# [27]**

Майже все сучасне програмування побудовано на принципах об'єктно-орієнтованого програмування. Застосування ООП дає змогу розробляти програмне забезпечення підвищеної складності за рахунок поліпшення його технологічності (кращих механізмів поділу даних, збільшення повторюваності кодів, використання стандартизованих інтерфейсів користувача тощо). Застосування ООП має свої плюси та мінуси (табл. 5).

Таблиця 5 - Плюси та мінуси об'єктно-орієнтованого програмування

Плюси	Мінуси
Легко читається	Споживає великий обсяг пам'яті
Швидко пишеться	Знижує продуктивність
Простіше реалізувати великий функціонал	Парадигма ООП складніша за функціональне програмування
Оптимальний код	

C# - потужна мова об'єктно-орієнтованого програмування.

### **ASP.NET - платформа для створення веб-додатків і служб .NET і C#**

ASP.NET - це веб-платформа, що надає всі необхідні служби для створення серверних веб-додатків корпоративного класу. ASP.NET створено на основі платформи .NET Framework, тому всі функції .NET Framework доступні для додатків ASP.NET [28].

#### **Переваги:**

- компільований код;
- висока швидкодія як наслідок компільованого коду;
- великий набір компонентів і продуману схему їхньої взаємодії;
- сучасна мова C#;
- бібліотека (framework);

- чудова документація.

## **Entity Framework Core**

Entity Framework (EF) Core - це проста, кросплатформенна і розширювана версія популярної технології доступу до даних Entity Framework з відкритим вихідним кодом.

EF Core може використовуватися як об'єктно-реляційний модуль зіставлення (O/RM), який:

- дозволяє розробникам .NET працювати з базою даних за допомогою об'єктів .NET;

- усуває необхідність у більшій частині коду для доступу до даних, який зазвичай доводиться писати.

Entity Framework Core - зручний інструмент для управління базами даних.

Після всіх етапів проектування буде реалізовано надійне ПЗ.

## **2.3 Функціональна схема додатка**

На стадії проектування програмного продукту необхідно створити інфологічну модель - модель, представлену безліччю понять і зв'язків між ними, що визначають бізнес-процеси та смислову структуру предметної області, що розглядається, або її конкретного об'єкта.

Під моделлю слід розуміти опис системи (текстовий і графічний), що демонструє загальну структуру та функціональність застосунку.

**Нульовий рівень** являє собою контекстну діаграму, яка описує найзагальніший опис системи та її взаємодії із зовнішнім середовищем.

Схему нульового рівня інфологічної моделі наведено на рисунку 10.



### Модель загального опису системи та її взаємодії з користувачем



Рисунок 10 - Схема нульового рівня інфологічної моделі

На діаграмі відображено вхідні та вихідні дані додатка. Взаємодія системи можлива в режимі двох типів користувачів (цільової аудиторії додатка): покупця (користувача освітніх товарів і послуг) і продавця (постачальника освітніх товарів і послуг). Студенти вишу найчастіше є споживачами, але можуть бути й постачальниками (тобто організаторами якогось заходу), викладачі та роботодавці найчастіше є постачальниками товарів і послуг.

На вхід системи із зовнішнього середовища надходять такі дані:

- у режимі реєстрації: особисті дані користувача;
- у режимі споживання: вибір товару або послуги, оцінка якості наданих матеріалів;
- у режимі особистого профілю: редагування даних профілю, кошика, списку обраних товарів і послуг;
- у режимі постачальника: створення карток товарів і послуг, їх редагування, відстеження кількості виборів товарів, доходу.

На виході відображаються каталог (вітрина), знайдений за пошуком і фільтром товар (послуга), інформація з Особистого кабінету.

Перший рівень являє собою декомпозицію контекстної діаграми, відображає внутрішню структуру програми. На цьому рівні показано взаємодію окремих модулів програми один з одним і зі сховищем даних



Рисунок 11 - Схема першого рівня інфологічної моделі

програми. Структура першого рівня інфологічної моделі відображена на рисунку 11.

Перший рівень функціонально розділений на такі модулі.

Розділ "**Реєстрація**" - модуль, що реалізує реєстрацію користувача в системі.

Розділ "**Авторизація**" надає можливість користувачеві увійти в систему (введення email і пароля).

Розділ "**Особистий кабінет**" дає змогу користувачеві керувати своїми даними, особистою інформацією. Різні користувачі мають різні можливості в особистому кабінеті. Покупець в особистому кабінеті бачить список усіх придбаних товарів і послуг, кошик, які може редагувати. Продавець в особистому кабінеті може створювати новий матеріал (інформацію про продукт або послугу), може переглядати, редагувати і видаляти матеріали про свої товари і послуги. Адміністратор працює зі списками користувачів,

списками товарів і послуг, а також підтверджує опублікування на сайті нового матеріалу.

**Сховище** - база даних додатка.

Після декомпозиції контекстної діаграми проводиться декомпозиція кожного великого фрагмента системи на більш дрібні.

Головне завдання маркетплейса «ОДІН» - презентація товарів і послуг освітнього характеру. Деталізація цього завдання складається з кроків:

1. Зареєструватися:

а. ввести дані для реєстрації;

2. авторизуватися:

а. ввести вхідні дані;

3. Робота покупця:

а. обрати товар;

б. придбати його;

с. обраний товар або послуга з'являється в кошику, а потім у списку товарів покупця;

д. переглянути та редагувати кошик;

е. оцінити якість придбання.

4. Робота продавця:

а. створення нового матеріалу, його перегляд;

б. перегляд, редагування, видалення своїх матеріалів зі списку.

5. Діяльність адміністратора:

а. робота зі списками користувачів;

б. дії зі списками товарів і послуг;

с. опублікування перевірених нових матеріалів.

## 2.4 Опис схеми даних

Виходячи з розробленої структури, виникає необхідність у зберіганні даних про користувачів, надісланих ними повідомлень, діалогів і групових чатів для функціонування розроблених модулів програми.

Створено базу даних із такою схемою, представленою на рисунку 12.

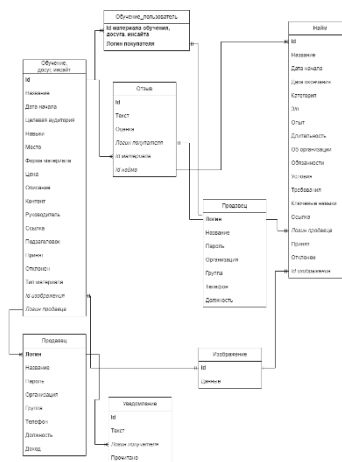


Рисунок 12 - Схема даних DB

Схема даних "DB" являє собою відносини між таблицями, у яких зберігається інформація, необхідна для функціонування розроблюваного додатка. У цій схемі визначено таблиці, описані нижче.

На рисунках 12-20 представлено таблиці бази DB: "Таблиця "Навчання, дозвілля, інсайт", "Навчання\_користувач", "Робота", "Повідомлення", "Відгук", "Продавець", "Користувач-покупець", "Зображення". Ключові поля позначені "ключиками".

Имя	Тип данных	Допустимы значения NULL	По умолчанию
Имя	Тип данных	Допустимы значения NULL	По умолчанию
id	int		

Рисунок 14 – Таблица «Уведомление»

Имя	Тип данных	Допустимы значения NULL
Id	int	<input type="checkbox"/>
EducationName	nvarchar(MAX)	<input type="checkbox"/>
StartDate	nvarchar(MAX)	<input type="checkbox"/>
TargetAudience	nvarchar(MAX)	<input type="checkbox"/>
Skills	nvarchar(MAX)	<input type="checkbox"/>
Place	nvarchar(MAX)	<input type="checkbox"/>
EducationForm	int	<input type="checkbox"/>
Price	int	<input type="checkbox"/>
Description	nvarchar(MAX)	<input type="checkbox"/>
Content	nvarchar(MAX)	<input type="checkbox"/>
Supervisor	nvarchar(MAX)	<input type="checkbox"/>
Link	nvarchar(MAX)	<input type="checkbox"/>
Subtitle	nvarchar(MAX)	<input type="checkbox"/>
IsAccepted	bit	<input type="checkbox"/>
IsDeclined	bit	<input type="checkbox"/>
EducationType	int	<input type="checkbox"/>
ImageId	int	<input checked="" type="checkbox"/>
SellerLogin	nvarchar(450)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Рисунок 15 - Таблица "Навчання, дозвілля, інсайт"

Имя	Тип данных	Допустимы значения NULL
Id	int	<input type="checkbox"/>
JobName	nvarchar(MAX)	<input type="checkbox"/>
StartDate	nvarchar(MAX)	<input type="checkbox"/>
EndDate	nvarchar(MAX)	<input type="checkbox"/>
Category	nvarchar(MAX)	<input type="checkbox"/>
Salary	int	<input type="checkbox"/>
Experience	nvarchar(MAX)	<input type="checkbox"/>
Duration	int	<input type="checkbox"/>
OrganizationDescription	nvarchar(MAX)	<input type="checkbox"/>
Obligations	nvarchar(MAX)	<input type="checkbox"/>
Conditions	nvarchar(MAX)	<input type="checkbox"/>
Requirements	nvarchar(MAX)	<input type="checkbox"/>
KeySkills	nvarchar(MAX)	<input type="checkbox"/>
Link	nvarchar(MAX)	<input type="checkbox"/>
OwnerLogin	nvarchar(450)	<input checked="" type="checkbox"/>
IsAccepted	bit	<input type="checkbox"/>
IsDeclined	bit	<input type="checkbox"/>
ImageId	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Рисунок 16 - Таблица "Робота"

Имя	Тип данных	Допустимы значения NULL	По умолчанию
Id	int	<input type="checkbox"/>	
Text	nvarchar(MAX)	<input type="checkbox"/>	
Mark	int	<input type="checkbox"/>	
UserLogin	nvarchar(450)	<input checked="" type="checkbox"/>	
EducationId	int	<input checked="" type="checkbox"/>	
JobId	int	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рисунок 17 - Таблица "Відгук"

Имя	Тип данных	Допустимы значения NULL	По умолчанию
Login	nvarchar(450)	<input type="checkbox"/>	
UserName	nvarchar(MAX)	<input type="checkbox"/>	
Password	nvarchar(MAX)	<input type="checkbox"/>	
Organization	nvarchar(MAX)	<input type="checkbox"/>	
Group	nvarchar(MAX)	<input type="checkbox"/>	
Phone	nvarchar(MAX)	<input type="checkbox"/>	
Position	nvarchar(MAX)	<input type="checkbox"/>	
Income	int	<input type="checkbox"/>	
		<input type="checkbox"/>	

Рисунок 18 - Таблица "Продавець"

Имя	Тип данных	Допустимы значения NULL	По умолчанию
Login	nvarchar(450)	<input type="checkbox"/>	
UserName	nvarchar(MAX)	<input type="checkbox"/>	
Password	nvarchar(MAX)	<input type="checkbox"/>	
Organization	nvarchar(MAX)	<input type="checkbox"/>	
Group	nvarchar(MAX)	<input type="checkbox"/>	
Phone	nvarchar(MAX)	<input type="checkbox"/>	
Position	nvarchar(MAX)	<input type="checkbox"/>	

Рисунок 19 - Таблица "Користувач-покупець"

Имя	Тип данных	Допустимы значения NULL	По умолчанию
Id	int	<input type="checkbox"/>	
ImageData	varbinary(MAX)	<input type="checkbox"/>	

Рисунок 20 - Таблица "Зображення"

Усі товари розділені на чотири розділи: навчання, дозвілля, інсайт, найм. Описи товарів трьох перших розділів, крім товарів розділу "найм" організовані однаково, тому вони містяться в одній таблиці "Навчання, дозвілля, інсайт". Інформація про товари розділу "найм" зберігається в таблиці "Робота".

До каталогу товарів і послуг маркетплейсу «ОДІН» включено основні дані зі стислою інформацією про освітні продукти сайту, зокрема (для привабливості) і зображення, що зберігаються в таблиці з однойменною назвою.

Для ідентифікації користувача використовується його ID. Створюються 2 таблиці користувачів: покупці та продавці. Кожному користувачеві відповідають його особисті дані. Таблиця "Навчання\_користувач" зберігає інформацію про товари, обрані споживачем. Таблиця "Відгуки" містить текст повідомлення покупця та оцінку (поле Double). Таблиця "Повідомлення" необхідна, щоб продавець знав, чи схвалений його матеріал адміністратором.

Таким чином, розроблена схема даних забезпечує необхідну функціональність реалізованої програми.

## 2.5 Висновок із розділу 2

Підбивши підсумки, перед початком програмування програми необхідно чітко сформулювати постановку задачі з погляду бажаного результату, вибрати інструментарій.

Далі, використовуючи інфологічну модель, можливо наочно відобразити структуру розробленого додатка, функціональні модулі та їхню роботу.

Спроектовано БД і схему взаємодії таблиць.

Цьому і було присвячено Розділ 2.

Тепер, коли зрозуміло, яким чином працюватиме застосунок, який він може мати вигляд, можливо переходити до реалізації проекту.

### 3. РЕАЛІЗАЦІЯ РІШЕННЯ

#### 3.1. Опис алгоритмів роботи програми

##### Алгоритми додатка

Алгоритм - набір інструкцій, що описують порядок дій виконавця для досягнення результату розв'язання задачі за кінцеве число дій.

Блок-схему узагальненого алгоритму роботи програми подано на рисунку 21.

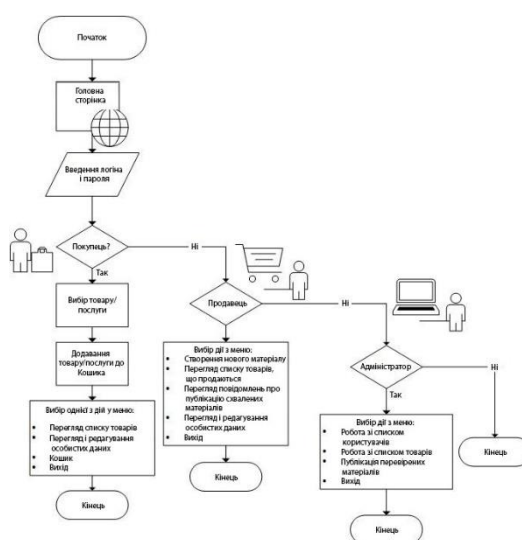


Рисунок 21 - Блок-схема додатка

Після запуску веб-додатка відображається головна сторінка сайту. Споживач вводить свій логін (електронну пошту) і пароль, після цього завантажується доступна для кожного виду користувача інформація і доступні різні види функцій. У цього маркетплейса 3 види користувачів: покупець (споживач товарів і послуг), продавець і роботодавець (постачальники товарів і послуг).

Усім видам користувачів доступна можливість редагування своїх особистих даних.

Для покупця (ним найчастіше є студент вишу) відкриті такі дії:



- вибір товару або послуги;
- пошук і фільтрація - для швидкого знаходження потрібного в каталозі;
- перегляд і редагування кошика;
- перегляд історії покупок ("список моїх покупок");
- вихід із профілю.

Продавець (ним найчастіше є викладач ВНЗ і роботодавець, студент може виконувати роль організатора і розмістити про нього інформацію на сайті) обрати з меню:

- створення нового матеріалу;
- список товарів, що продаються (їх перегляд, редагування, видалення);
- повідомлення про публікацію схвалених адміністратором матеріалів;
- перегляд і редагування особистих даних;
- вихід із профілю.

Адміністратор платформи виконує таку роботу:

- робота зі списком користувачів;
- робота зі списком товарів;
- публікація перевірених матеріалів;
- перегляд і редагування особистих даних;
- вихід із профілю.

Приклад реалізації алгоритмічних дій покупця на сайті та "доставка" товару або послуги йому наведено на рисунку 22.

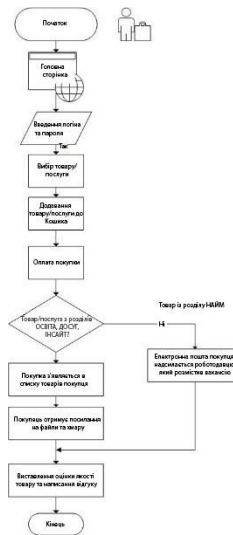


Рисунок 22 - Блок-схема алгоритму поведінки покупця та реакції додатка на його дії

Покупець обирає товар/послугу, додає його до Кошика, оплачує покупку. У результаті покупка з'являється в списку матеріалів користувача. Покупець освітнього контенту (товари та послуги розділів "Навчання", "Дозвілля", "Інсайт" отримує і посилання на файли. У хмарі можуть перебувати: інструкції, тексти, відео, посилання на реєстрацію події тощо. Усе це стосується обраного користувачем навчального курсу, заходу, семінару тощо.

Якщо ж покупець звернув увагу на вакансію (розділ "Найм"), то електронну пошту покупця надсилають роботодавцю, який зробив оголошення (розмістив картку товару на маркетплейсі).

Покупець може проголосувати за найкращий продукт і залишити зворотний зв'язок.

На рисунку 23 знаходиться блок-схема, яка описує дії продавця, а також, як на це реагує додаток.



Рисунок 23 - Блок-схема алгоритму поведінки продавця та реакції додатка на його дії

Після авторизації продавець має можливість створити новий матеріал (картку товару/послуги та короткий опис), попередньо переглядати отримане та відправити на перевірку адміністратору. Якщо адміністратор схвалює матеріал, він його публікує і відправляє повідомлення автору (продавцю). Матеріал з'являється у списку товарів продавця, де автор може змінити, видалити, переглянути картки товарів.

У цьому розділі описано тільки частину алгоритмів у загальному вигляді, щоб було яскравіше видно бізнес-процеси роботи маркетплейса «ОДІН».

### 3.2 Демонстрація функціонування програми

Спочатку користувач побачить головну сторінку маркетплейса «ОДІН» (рис. 24).

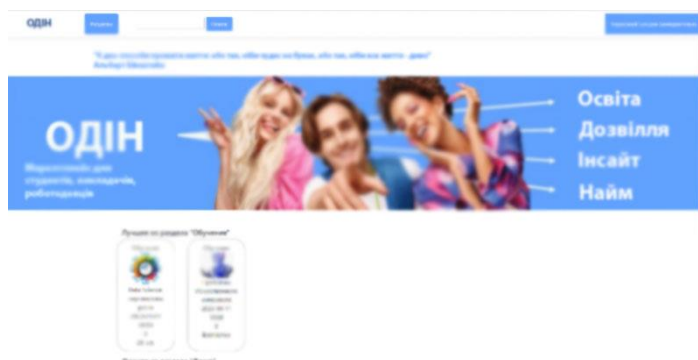


Рисунок 24 - Головна сторінка

Після авторизації покупець обирає товар або послугу, додає покупку в кошик (рис. 25).



Рисунок 25 - Кошик покупця

Почувець оплачує товари, їх можна знайти потім у пункті меню "Список моїх покупок" (рис. 26).



Рисунок 26 – Список покупок користувача (історія покупок)

Якщо товар належить до одного з розділів "Навчання", "Дозвілля", "Інсайт", то після поміщення до Кошика споживач отримує посилання на хмару. Якщо покупка - вакансія з розділу "Найм", то електронну адресу покупця направляють роботодавцю в Особистий кабінет. На рисунку 27 зображено вміст хмари для форум-театру "Вибір".

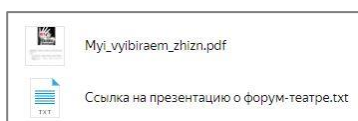


Рисунок 27 - Вміст хмари для форум-театру, представлення якого розміщено на маркетплейсі «ОДІН»

Після авторизації продавець у своєму Особистому кабінеті може створити новий матеріал: короткий опис товару для вітрини та картку товару (повний опис усіх характеристик) (рис. 28).

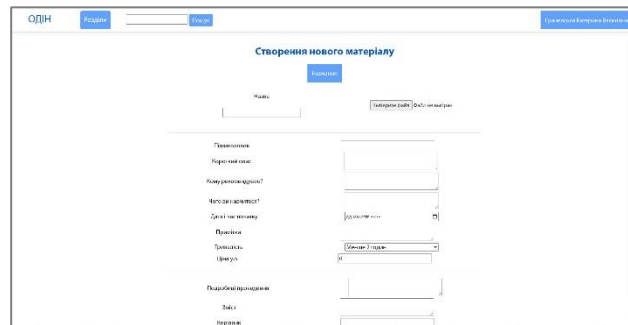


Рисунок 28 - Створення нового матеріалу продавцем

Після перегляду готового опису продукту автоматично надсилається заявка адміністратору на перевірку.

Будь-який споживач (покупець, продавець) у пункті меню "Профіль" відкриває сторінку перегляду та редагування даних споживача (рис. 29).

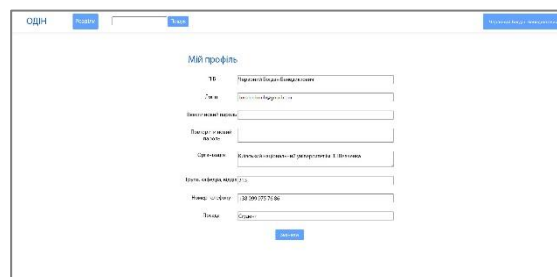


Рисунок 29 - Сторінка "Профіль" для покупця

На рисунку 30 представлена сторінка з повідомленнями продавця про те, що його матеріал прийнято та опубліковано.



Рисунок 30 - Сторінка "Повідомлення"

Фрагмент панелі адміністратора з можливими діями представлено на рисунку 31. На ньому зображено таблицю з даними про матеріали.

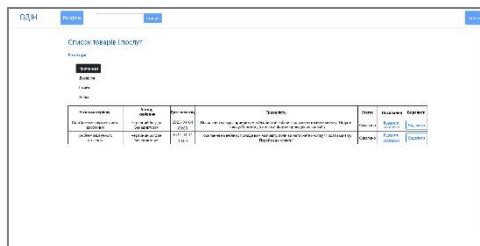


Рисунок 31 – Панель адміністратора с таблицею матеріалів сайту

Демонстрація сторінок веб-додатка «ОДІН» показує, що всі задумані функції реалізовано, інтерфейс простий і зрозумілий.

### 3.3 Опис застосування програмного продукту

Представлений web-додаток «ОДІН», є зручним і надійним онлайн-маркетплейсом. Він спеціалізується на просуванні різноманітних освітніх послуг, послуг з організації дозвілля, а також підтримці проектної діяльності студентів і встановленні зв'язків між учнями, викладачами та роботодавцями у вищій професійній освіті.

Необхідно забезпечити заповнення вебсайту змістом, який є цікавим, актуальним і перспективним для користувачів, виходячи з цілей і завдань конкретного навчального закладу.

Важливо зазначити, що успіх роботи онлайн-маркетплейса визначається ефективними маркетинговими стратегіями і методами просування.

Для привернення уваги аудиторії використовують різноманітні стратегії. Важливо приділяти увагу якості оформлення карток товарів.

Для досягнення підвищеної конверсії та збільшення обсягу продажів необхідно постійно проводити оптимізацію картки товару. Це охоплює тестування і поліпшення фотографій, редагування опису товарів, а також ретельну роботу з деталями і характеристиками.

Необхідно розуміти, що створений маркетплейс - це тільки інструмент. Щоб досягти поставлених цілей, потрібно створити систему, в якій буде продумано такі елементи, як: людські ресурси, фінансування, утримання, просування, акценти взаємодії вищу та бізнесу тощо.

Додаток «ОДІН» доцільніше застосувати не для одного навчального закладу, а для кількох. Бо ця обставина підвищить конкурентоспроможність наданих товарів і послуг, дасть змогу обмінюватися досвідом, отримувати більший зворотний зв'язок.

### **3.4. Список документації до додатка.**

Досвід показує, що як супровідні документи можуть бути:

1. інструкція із застосування для покупця.
2. Керівництво продавця (постачальника товарів і послуг).
3. Керівництво адміністратора.

Короткий опис кожного документа розташовано нижче.

#### **Інструкція із застосування для покупця**

Покупцем (користувачем) веб-маркетплейсу «ОДІН» є студент вищої професійної освіти, який є споживачем контенту цього сайту.

Інтерфейс онлайн-маркетплейса «ОДІН» інтуїтивно зрозумілий. Але для тих, кому хочеться заощадити свій час на освоєння цього додатка, і призначена ця інструкція. У ній представлені вікна, що відкриваються, і дані пояснення до них.

Крім того, в цій інструкції описано особливості роботи маркетплейса «ОДІН».

Зміст:

1. загальна інформація;
2. реєстрація;
3. авторизація;
4. пошук і фільтрація товару;
5. особистий кабінет покупця;
  - a. список товарів покупця;
  - b. кошик;
  - c. профіль;
  - d. вихід;
6. схема роботи покупця з маркетплейсом. Основні алгоритми.

### **Посібник продавця (постачальника товарів і послуг).**

Продавцем веб-маркетплейсу «ОДІН» є викладач вищої професійної освіти, роботодавець, які є постачальниками контенту цього сайту.

Студент вишу також може зареєструватися на сайті як продавець (але логіни покупця і продавця мають відрізнятися!). У такій ситуації студент виступає як організатор будь-якого заходу.



У цьому посібнику представлено основні можливості продавця, а також особливості роботи маркетплейса «ОДІН».

Зміст:

1. загальна інформація;
2. авторизація;
3. особистий кабінет продавця;
  - a. створення нового матеріалу;
  - b. список товарів продавця. Перегляд, редагування, видалення товарів продавця;
  - c. повідомлення;
  - d. профіль;
  - e. вихід;
4. схема роботи продавця з маркетплейсом. Основні алгоритми.

### **Посібник адміністратора.**

Адміністрування сайту - це комплекс заходів, пов'язаних із працездатністю та ефективністю веб-проекту.

Адміністратор сайту має бути єдиним відповідальним за ведення всіх заходів, пов'язаних із роботою сайту в Інтернеті. Так само адміністратор сайту може очолювати команду профільних фахівців.

У цьому посібнику описано особливості роботи адміністратора маркетплейса «ОДІН».

Зміст:

1. загальна інформація;
2. реєстрація;
3. авторизація;

4. панель адміністратора;

- a. робота зі списком користувачів сайту;
- b. робота зі списком товарів маркетплейса;
- c. кошик.

Детальний опис документів можна знайти в Додатку А.

Супровідні документи полегшать роботу користувачів сайту «ОДІН».

### **3.5 Можливості подальшого розвитку рішення**

Маркетплейс можна порівняти з величезним торговим майданом, де на кожному стенді представлені однакові товари від різних брендів і марок. На маркетплейсах покупці активно шукають необхідні товари та послуги, відбираючи найкращі варіанти відповідно до їхніх важливих критеріїв, як-от виробник, характеристики, якість, ціна та інші. Важливо підкреслити, що маркетплейси самі по собі є всього лише посередниками між виробниками і споживачами, забезпечуючи зручну платформу для взаємодії [1].

Існує значна відмінність між інтернет-магазином і маркетплейсом, оскільки це дві абсолютно різні моделі продажів. Головна відмінність полягає в тому, що інтернет-магазин переслідує цілі та інтереси власного бізнесу, сфокусовані на його власному розвитку. Водночас, маркетплейс не переслідує мети розвитку певної компанії чи бренду, його головне завдання полягає у збільшенні обсягу продажів на самій платформі.

Слід зауважити, що спочатку презентоване рішення, онлайн-маркетплейс «ОДІН», рекомендується використати в одному навчальному закладі (ВНЗ). Відпрацювати не лише цей онлайн-інструмент, а й організаційні, технічні та інші питання.

Для повнішого застосування переваг маркетплейсів перед інтернет-магазинами найкраще сайт «ОДІН» експлуатувати кільком освітнім установам

(вишам), створивши простір взаємодії, обміну досвідом, спільними інтересами, збагативши один одного.

Безперервні зміни - запорука руху вперед.

З технічного погляду, необхідно розвивати рішення: додати аналітику, розширити можливості фільтра, посилити права адміністратора функцією зміни розділів.

Передбачити ще кілька ролей: менеджерів, методистів, які б забезпечували контроль за контентом, реалізацію задуманих проєктів. Для цього треба передбачити й особисті кабінети менеджерів.

Варто продумати й додати функції безпеки.

### **3.6 Висновок із розділу 3**

У цьому розділі були розглянуті важливі аспекти реалізації задуманого додатка: основні алгоритми, способи застосування маркетплейса.

У тексті цієї роботи було запропоновано й шляхи розвитку застосунку, збільшуючи його функціонал.

## ВИСНОВОК

У першому розділі було проаналізовано вже наявні аналоги маркетплейсів, зокрема й освітніх.

Крім того, застосування онлайн-маркетплейса для розв'язання проблем професійної освіти, поліпшення взаємодії вишів і бізнесу може бути надзвичайно ефективним.

У другому розділі було створено моделі системи. Описано постановку задачі, інструменти для програмування, намальовано інформаційно-функціональні моделі застосунку. Це базова підготовка до програмування.

У третьому розділі розказано, як було реалізовано задумане мовою C# із застосуванням HTML, CSS, Java Script. Для роботи СУБД було обрано Microsoft SQL і технологію Entity Framework. Для розгортання веб-сервера було використано програмне забезпечення IIS, ASP.NET.

Створено супровідні документи: інструкцію із застосування, керівництва для трьох видів користувачів.

Запрограмований додаток реалізує задуманий функціонал. Документи, що додаються, допоможуть цільовій аудиторії легко і радісно користуватися готовим продуктом.

## ДЖЕРЕЛА

1. Mikitani Hiroshi , Marketplace 3.0: Rewriting the Rules of Borderless Business / Hiroshi Mikitani - New York, 2014 - 272 с.
2. Брайар К., «Стратегія Amazon. Інструменти безкомпромісної роботи на вражаючий результат» / К. Брайар, Б. Карр - "Ексмо", 2022р. - 352с.
3. Кларк Д., «Alibaba. Історія світового сходження від першої особи». / Д. Кларк - "Форс", 2017 р. - 240с.
4. Паркер Д. «Революція платформ. Як мережеві ринки змінюють економіку - і як змусити їх працювати на вас». / Паркер Д., Альстайн М., Чаударі П. – «W. W. Norton & Company», 2016 р. – 304с.
5. Хіз Д., «Зроблено, щоб прилипати. Чому одні ідеї виживають, а інші вмирають?» / Д. Хіз,, Ч. Хіз. – «Манн», 2007 р. – 256с.
6. Дірксен Д., «Мистецтво навчати. Як зробити будь-яке навчання ненудним і ефективним». / Д. Дірксен – «Міф», 2016 р. – 312с.
7. Уейншенк С., «100 головних принципів дизайну. Як утримати увагу?». / С. Уейншенк, - «New Raiders», 2012 р. - 272с.
8. К. Вербах, «Залучай і володарюй. Ігрове мислення на службі бізнесу». / К. Вербах, Д. Хантер, - «Манн», 2014 – 224с.
9. Гріненко О.Д. «Теоретико-методологічні засади вивчення вільного часу молоді // Проблеми сучасної науки і освіти.». / О.Д. Гріненко, - 2014 р. №5 (23). – С.112-117
10. Huang C. «The relationships among leisure participation, leisure satisfaction, and life satisfaction of college students in Taiwan // Journal of Exercise Science and Fitness.» / C. Huang, B. Carleton, - 2003 y. -132 p.
11. Davey T. «Education and Culture Study on the Cooperation Between HEIs and Public and Private Organisations in Europe.». Content presentation. / T. Davey, Th. Baaken, V. Galan Muros, A. Meerman

12. Europe 2020: «A European strategy for smart, sustainable and inclusive growth» - Brussels, Belgium: «European Commission», 2010 y. – 32с.
13. McAvoy J. «Global Education Outlook» / J. McAvoy, K. Pickering. – «Macmillan Education» 2010 y. – 92 p.
14. Лаптон Е. «Графічний дизайн. Нові основи» / Е. Лаптон, Дж. Коул – «ArtHuss», 2020р. - .262с.
15. «Оцінювання ефективності роботи вчителів: Нові тенденції, відображені в заявках на участь у першому етапі конкурсу "Гонка до вершин" на рівні штатів» / Співробітники навчальних частин, 2011 р. – 20 с.
16. Кенеді Д. «Написання та застосування результатів навчання: практичний посібник. Реалізація Болонського процесу у вашому виші. Використання результатів навчання та компетенцій // Болонський процес: Результати навчання та компетентній підхід» / Д. Кенеді Е. Хайленд, Н. Райан, - 2019 р. –502с.
17. Блюменфельд П. К. «Мотивація проектного навчання: Підтримувати діяльність, підтримувати навчання» // П. К. Блюменфельд, Е. Солоуей, Р. В. Маркс, - «Educational Psychologist», 1991 р. – 398с.
18. Тенгвалл М. «Бізнес-цінність відносин з розробниками: Як і чому технічні спільноти є ключем до вашого успіху» / М. Тенгвалл, 2018 р. – 260с.
19. Роберт А. «Після роботи: пошук оптимального способу проведення дозвілля». / А. Роберт – «Detselig Enterprises», 2002 р. – 184с.
20. «Дозвіллів освіта серед студентів» / Європейський збірник наукових праць електронна сторінка:  
<https://www.europeanproceedings.com/article/10.15405/epsbs.2017.07.03.6>  
6.
21. Круг С., «Не змушуй мене думати: Підхід до юзабіліті на основі здорового глузду» / С. Круг, - 3-е вид., «New Riders» 2013 р. – 216с.

22. Офіційна сторінка Visual Studio: <https://visualstudio.microsoft.com>
23. Офіційна документація C#: <https://docs.microsoft.com/en-us/dotnet/csharp>.
24. Фрімен А. «ASP.NET Core MVC» / А. Фрімен, - «Діалектика», 2018 р. - 1088с.
25. Мейер Е. «CSS повний довідник» / Е. Мейер, Е. Вейл. – «Естелла», 2019 р. – 816с.
26. Дакетт Дж. «JavaScript і jQuery: розробка інтерактивних сайтів» / Дж. Дакетт. – 2017 р. – 641с.
27. Ніксон Р., «Вивчаємо PHP, MySQL та Javascript; з jQuery, CSS та HTML5». / Р. Ніксон, – 5-е вид. «O'Reilly Media», 2018 р. – 832с.
28. Роббінс Дж., «Вивчаємо веб-дизайн: Посібник для початківців з HTML, CSS, Javascript та веб-графіки» / Дж. Роббінс - «O'Reilly Media», 2018 р. – 808с.

## Додаток А. Документація до web-додатку

Досвід показує, що як супровідні документи можуть бути:

1. інструкція із застосування для покупця.
2. Керівництво продавця (постачальника товарів і послуг).
3. Керівництво адміністратора.

### **Інструкція із застосування для покупця**

Покупцем (користувачем) веб-маркетплейса «ОДІН» є студент вищої професійної освіти, який є споживачем контенту цього сайту.

Інтерфейс онлайн-маркетплейса «ОДІН» інтуїтивно зрозумілий. Але для тих, кому хочеться заощадити свій час на освоєння цього додатка, і призначена ця інструкція. У ній представлено вікна, що відкриваються, і дано пояснення до них.

Крім того, в цій інструкції описані особливості роботи маркетплейса «ОДІН».

### Зміст

1. загальна інформація;
2. реєстрація;
3. авторизація;
4. пошук і фільтрація товару;
5. особистий кабінет покупця;
  - a. список товарів покупця;
  - b. кошик;
  - c. профіль;
  - d. вихід;
6. схема роботи покупця з маркетплейсом. Основні алгоритми.



## Загальна інформація

«ОДІН» - зручний і надійний онлайн-маркетплейс для просування освітніх послуг, послуг з організації дозвілля, проектної діяльності студентів, взаємозв'язку учнів з викладачами, роботодавцями вищого професійного закладу. Головна сторінка сайту - на рисунку 32.

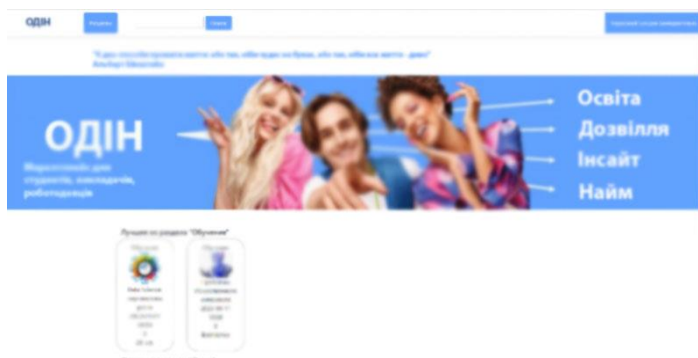


Рисунок 32 - Головна сторінка

### Цілі створення та призначення системи:

організація взаємодії студентів, викладачів і роботодавців для розвитку практичних знань і навичок випускників, формування залученості студентів у процес навчання.

### Цільова аудиторія:

- студенти;
- викладачі;
- роботодавці.

Нижче наведено приклади продуктів, інформація про які розміщена на сайті. На рисунку 33 розміщена коротка інформація про товари/послуги маркетплейсу. 4 розділи: навчання, дозвілля, інсайт, найм. 4 продукти: вебінар, форум-театр, конференція студентських TED, вакансія. Продукти можуть бути як платними, так і безкоштовними.



Рисунок 33 - На вітрині маркетплейса може бути розміщена інформація про 4 продукти

Покупець шукає потрібний товар, додає в кошик, оплачує. Маркетплейс "доставляє" покупку споживачеві: це може бути посилання на вебінар, квиток на захід тощо.

### ***Реєстрація***

Спочатку треба зареєструватися. Для цього є сторінка реєстрації. На рисунку 34 знаходяться поля, які потрібно заповнити.

Рисунок 34 - Користувач почав свою реєстрацію

### ***Авторизація***

Для того, щоб увійти в додаток, споживач вводить свої логін і пароль.

**ВАЖЛИВО!** Студент може виступати, як і в ролі покупця, так і в ролі продавця (постачальника послуги). Для цього треба буде зареєструватися 2 рази у двох ролях з різними логінами.

Для пошуку товару покупець може скористатися інструментами "пошук" і "фільтр" (рис. 35).



Рисунок 35 - Схема роботи покупця з маркетплейсом при виборі товару - вакансії

Фільтр пропонує різні параметри відбору. Наприклад, параметр "Скоро початок" означає, що до початку події менше 1 тижня.

### ***Особистий кабінет покупця***

В Особистому кабінеті покупець може вибрати будь-який пункт меню: (Список товарів покупця, Кошик, Профіль, Вихід) і виконати будь-яку відповідну дію.

На рисунку 36 зображено таблицю тих товарів користувача, які були придбані (історія покупок). У таблиці "Список куплених товарів" знаходяться 2 товари з розділу "Навчання", доступні дві дії: перегляд і перехід за посиланням.



Рисунок 36 – Список товарів користувача

За допомогою посилання маркетплейс "доставляє" товар до споживача. Перший продукт - курс "Data Science: перспективи зростання". Посилання в стовпчику "Файли" - це посилання на хмару, де зберігаються файли курсу.

Якщо в списку є товар із розділу "Найм", то в стовпчику "Файл" посилання немає. Наприклад, при замовленні товару "вакансія", роботодавцю,



який розмістив вакансію, прийде електронна пошта, за допомогою якої роботодавець зможе поспілкуватися з охочим знайти роботу.

Рисунок 37 - Кошик покупця

На рисунку 37 представлена сторінка з Кошиком. Оскільки товар з ім'ям "Вибір" - це безкоштовний захід, то до оплати представлено - 0 у.о.

Профіль відображає особисті дані покупця, які можна редагувати (рис. 38).

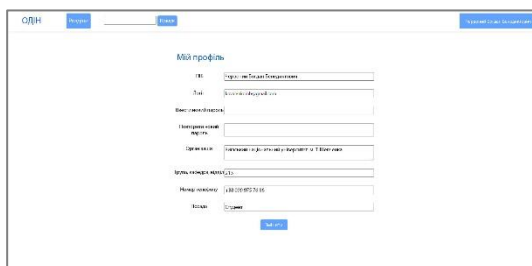


Рисунок 38 - Профіль покупця

### ***Схема роботи покупця з маркетплейсом. Основні алгоритми***

На рисунку 39 треба розглянути схему взаємодії всіх учасників процесу (покупець, маркетплейс, продавець). Усі товари та послуги, представлені на сайті, доставляються онлайн, віддалено. А самі товари та послуги (курси,



вистави тощо) можуть проходити і в очному, і в дистанційному, і в заочному тощо форматах.

Рисунок 39 - Схема роботи покупця з маркетплейсом

Покупець робить замовлення. Після додавання товару/послуги до Кошика, користувач оплачує, якщо треба, замовник отримує посилання на файли.

Якщо товар належить до розділу "Найм", наприклад, це вакансія, то продавець отримує електронну пошту і по ній зв'язується з претендентом на посаду (рис. 40).

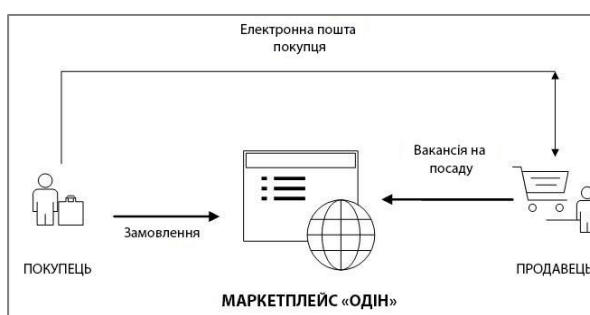


Рисунок 40 - Схема роботи покупця з маркетплейсом при виборі товару - вакансії

### Посібник продавця (постачальника товарів і послуг)

Продавцем веб-маркетплейсу «ОДІН» є викладач вищої професійної освіти, роботодавець, які є постачальниками контенту цього сайту.

Студент вишу також може зареєструватися на сайті як продавець (**але логіни покупця і продавця мають відрізнятися!**). У такій ситуації студент виступає як організатор будь-якого заходу.

У цьому посібнику представлено основні можливості продавця, а також особливості роботи маркетплейса «ОДІН».

Зміст

1. загальна інформація;
2. реєстрація;
3. авторизація;
4. особистий кабінет продавця;
  - a. створення нового матеріалу;
  - b. список товарів продавця. Перегляд, редагування, видалення товарів продавця;
  - c. повідомлення;
  - d. профіль;
  - e. вихід.

### ***Загальна інформація***

«ОДІН» - зручний і надійний онлайн-маркетплейс для просування освітніх послуг, послуг з організації дозвілля, проектної діяльності студентів, взаємозв'язку учнів з викладачами, роботодавцями вищого професійного закладу. Головна сторінка сайту - на рисунку 41.

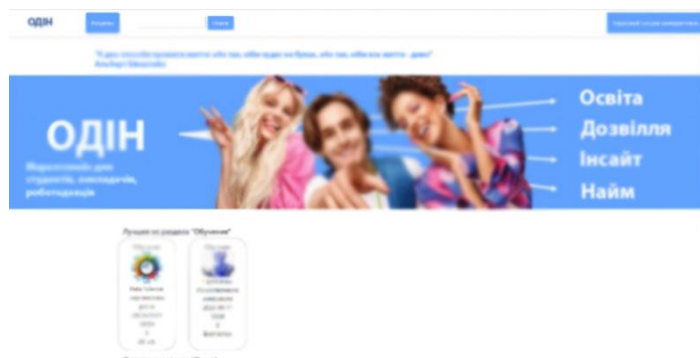


Рисунок 41 - Головна сторінка

**Цілі створення та призначення системи:**

організація взаємодії студентів, викладачів і роботодавців для розвитку практичних знань і навичок випускників, формування залученості студентів у процес навчання.

### Цільова аудиторія:

- студенти;
- викладачі;
- роботодавці.

Нижче наведено приклади продуктів, інформація про які розміщена на сайті. На рисунку 40 розміщена коротка інформація про товари/послуги маркетплейсу. 4 розділи: навчання, дозвілля, інсайт, найм. 4 продукти: вебінар, форум-театр, конференція студентських TED, вакансія. Продукти можуть бути як платними, так і безкоштовними.



Рисунок 42 - На вітрині маркетплейса може бути розміщена інформація про 4 продукти

Покупець шукає потрібний товар, додає в кошик, оплачує. Маркетплейс "доставляє" покупку споживачеві: це може бути посилання на вебінар, квиток на захід тощо.

Продавець презентує покупця про товар, розміщуючи на сайті коротку і повну інформацію про товар/послугу. Наприклад, картку товару для заходу "Конференція студентських TED" можна знайти на рисунку 43 (це тільки частина інформації).



Рисунок 43 - Картка товару (частина інформації)

### ***Реєстрація***

Для того, щоб увійти в додаток, споживач вводить свої логін і пароль.

**ВАЖЛИВО!** Студент вишу може виступати, як і в ролі покупця, так і в ролі продавця (постачальника послуги). Для цього треба буде зареєструватися 2 рази у двох ролях з різними логінами.

### ***Особистий кабінет продавця***

В Особистому кабінеті покупець може вибрати будь-який пункт меню: (Створення нового матеріалу, Список товарів продавця, Повідомлення, Профіль, Вихід) і виконати будь-яку відповідну дію.



Рисунок 44 відображає сторінку створення нового матеріалу. Матеріал містить у собі всю інформацію про товар/послугу: невелику частину видно в каталозі товарів/послуг (рис. 42) і на окремій сторінці (рис. 43).



Рисунок 44 - Сторінка створення нового матеріалу

Усі товари/послуги розділені на 4 розділи: навчання, дозвілля, інсайт, найм. Для трьох розділів (навчання, дозвілля, інсайт) створено одну форму для заповнення (рис. 44). Для розділу "Найм" - інша форма.

У таблиці 6 містяться варіанти продуктів і послуг.

Таблиця 6 - Приклади товарів і послуг за кожним розділом

Розділ	Мета для студента	Варіанти продуктів і послуг
Освіта	Отримувати додаткові знання та вміння і навички	Навчальні заходи в інтерактивній формі проведення, наприклад: <ul style="list-style-type: none"> <li>- курси додаткової освіти;</li> <li>- консультації (студенти студентам);</li> <li>- цікаві вебінари;</li> <li>- зустрічі з відомими людьми;</li> <li>- ділові ігри;</li> <li>- конкурси.</li> </ul>
Дозвілля	Духовно, фізично розвиватися та повноцінно відпочивати	Розважально-розвивальні заходи, наприклад: <ul style="list-style-type: none"> <li>- соціальний театр;</li> <li>- виставки;</li> <li>- фестивалі;</li> </ul>

		<ul style="list-style-type: none"> <li>- шоу-кейси;</li> <li>- квести;</li> <li>- дебати;</li> <li>- спортивні заходи.</li> </ul>
Інсайт	Проявляти інтелект, творчість, підприємництво	<p>Особливі великі проекти та практичні корисні поради, наприклад:</p> <ul style="list-style-type: none"> <li>- практико-орієнтовані проекти (соціальні, наукові, навчальні) за участю роботодавців;</li> <li>- хакатони;</li> <li>- конкурс бізнес-ідей;</li> <li>- лайфхаки;</li> <li>- чек-листи;</li> <li>- кейси-завдання.</li> </ul>
Найм	Отримувати нові компетенції	<p>Пропозиції роботи від компаній, приватних осіб, навчальних закладів.</p> <p>Рекомендації для успішної кар'єри, наприклад:</p> <ul style="list-style-type: none"> <li>- корисні поради щодо написання резюме, побудови кар'єри, поведінки на співбесіді тощо;</li> <li>- співбесіди з роботодавцем;</li> <li>- волонтерська робота;</li> <li>- вакансії з неповною зайнятістю, віддаленою роботою.</li> </ul>

Продавець створює новий матеріал, зберігає його, і він автоматично надсилається адміністратору на узгодження.

На рисунку 45 зображено таблицю тих товарів користувача, які створені продавцем. У таблиці "Список моїх товарів" знаходяться 2 товари з розділу "Навчання", обидва товари вже підтверджені і розміщені адміністратором на сайті, доступні три дії: переглянути, змінити, видалити. На цій сторінці можна натиснути на гіперпосилання і подивитися відгуки.



Рисунок 45 - Список товарів продавця

За допомогою посилання маркетплейс "доставляє" товар до споживача. Перший продукт - курс "Data Science: перспективи зростання". Доставка онлайн - це посилання на хмару, де зберігаються файли курсу.

Якщо в списку знаходиться товар із розділу "Найм", то доставки немає. Наприклад, при замовленні товару "вакансія", роботодавцю, який розмістив вакансію, прийде електронна пошта, за допомогою якої роботодавець зможе поспілкуватися з охочим знайти роботу. На рисунку 46 зображено відгуки на вакансію.



Рисунок 46 - Відгуки на вакансію

Профіль відображає особисті дані продавця, які можна редагувати (рис. 47).

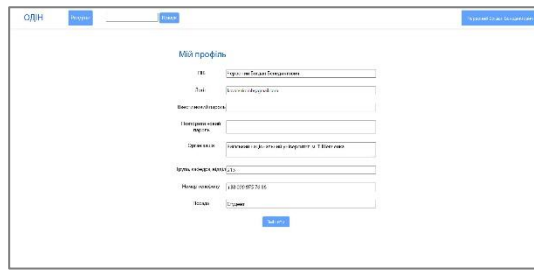


Рисунок 47 - Профіль покупця

### ***Схема роботи продавця з маркетплейсом. Основні алгоритми***

На рисунку 48 треба розглянути схему взаємодії всіх учасників процесу (покупець, маркетплейс, продавець). Усі товари та послуги, представлені на сайті, доставляються онлайн, віддалено. А самі товари та послуги (курси, вистави тощо) можуть проходити і в очному, і в дистанційному, і в заочному тощо форматах.



Рисунок 48 - Схема взаємодії

Покупець робить замовлення. Після додавання товару/послуги до Кошика, користувач оплачує, якщо треба, замовлення і отримує посилання на файли.

Якщо товар належить до розділу "Найм", наприклад, це вакансія, то продавець отримує електронну пошту і по ній зв'язується з претендентом на посаду (рис. 49).

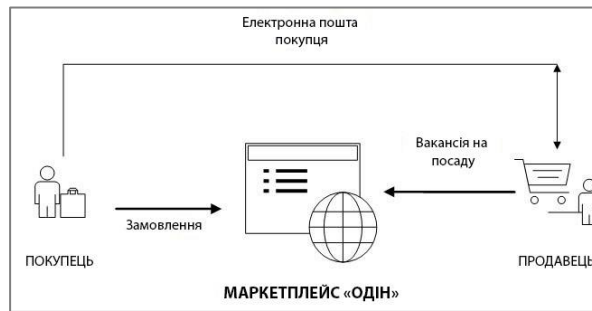


Рисунок 49 - Схема роботи покупця з маркетплейсом при виборі товару  
- вакансії

Схема взаємодії продавця й адміністратора розміщена на рисунку 50.



Рисунок 50 - Схема взаємодії продавця та адміністратора

Збережений продавцем матеріал автоматично направляється адміністратору на перевірку.

### Керівництво адміністратора

Адміністрування сайту - це комплекс заходів, пов'язаних із працездатністю та ефективністю веб-проекту.

Адміністратор сайту має бути єдиним відповідальним за ведення всіх заходів, пов'язаних з роботою сайту в Інтернеті. Так само адміністратор сайту може очолювати команду профільних фахівців.

У цьому посібнику описано особливості роботи адміністратора маркетплейса «ОДІН».

## Зміст

1. загальна інформація;
2. авторизація;
3. панель адміністратора;
  - a. робота зі списком користувачів сайту;
  - b. робота зі списком товарів маркетплейса;
  - c. кошик;
  - d. вихід;
4. схема роботи продавця з маркетплейсом. Основні алгоритми.

## *Загальна інформація*

«ОДІН» - зручний і надійний онлайн-маркетплейс для просування освітніх послуг, послуг з організації дозвілля, проектної діяльності студентів, взаємозв'язку учнів з викладачами, роботодавцями вищого професійного закладу. Головна сторінка сайту - на рисунку 51.

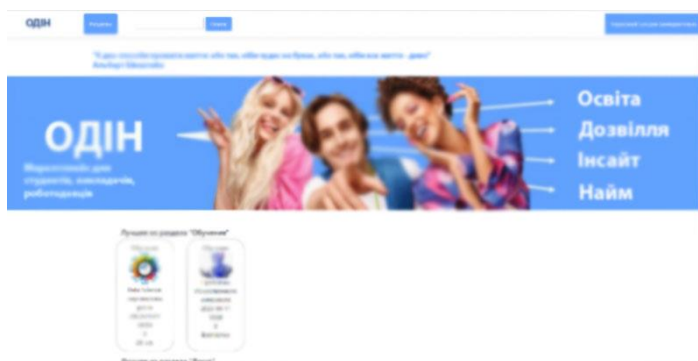


Рисунок 51 - Головна сторінка

## **Цілі створення та призначення системи:**

організація взаємодії студентів, викладачів і роботодавців для розвитку практичних знань і навичок випускників, формування залученості студентів у процес навчання.

### Цільова аудиторія:

- студенти;
- викладачі;
- роботодавці.

Нижче наведено приклади продуктів, інформація про які розміщена на сайті. На рисунку 52 розміщена коротка інформація про товари/послуги маркетплейсу. 4 розділи: навчання, дозвілля, інсайт, найм. 4 продукти: вебінар, форум-театр, конференція студентських TED, вакансія.



форум-театр, конференція студентських TED, вакансія. Продукти можуть бути як платними, так і безкоштовними.

Рисунок 52 - На вітрині маркетплейса може бути розміщена інформація про 4 продукти

Покупець шукає потрібний товар, додає в кошик, оплачує. Маркетплейс "доставляє" покупку споживачеві: це може бути посилання на вебінар, квиток на захід тощо.

Продавець презентує покупця про товар, розміщуючи на сайті коротку і повну інформацію про товар/послугу. Наприклад, картку товару для заходу "Конференція студентських TED" можна знайти на рисунку 53 (це тільки частина інформації).



Рисунок 53 - Картка товару (частина інформації)

Збережений продавцем матеріал автоматично направляється адміністратору на перевірку.

Схема взаємодії продавця й адміністратора розміщена на рисунку 54.

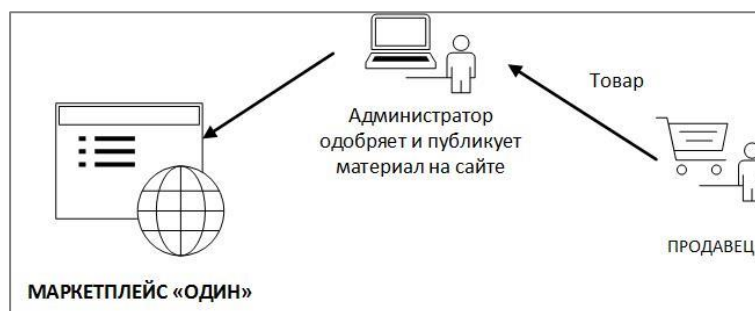


Рисунок 54 - Схема взаємодії продавця та адміністратора

### **Авторизація**

Логін адміністратора - Admin

Пароль адміністратора - 100

**Панель адміністратора** надає такі можливості:



- a. Робота зі списком користувачів сайту.
- b. Робота зі списком товарів маркетплейса.
- c. Вихід.

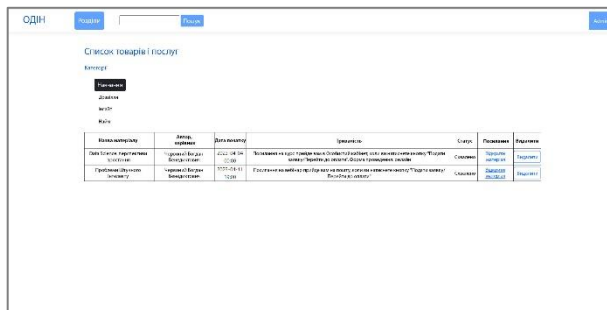


Рисунок 55 - Сторінка «Список товарів і послуг»

На рисунку 55 розміщена таблиця, в якій пропонуються функції, що дають змогу переглянути матеріал за товаром/послугою і видалити його.

## Додаток Б. Код програми

Лістинг файлу ListController.cs:

```
using MarketPlace.Extra;
using MarketPlace.Models;
using MarketPlace.ViewModels;
using Microsoft.AspNetCore.Mvc;
using System.Globalization;

namespace MarketPlace.Controllers
{
    /// <summary>
    /// Фільтр "Початок"
    /// </summary>
    public enum Start
    {
        StartSoon,
        Insight
    }

    /// <summary>
    /// Фільтр "Платний чи ні"
    /// </summary>
    public enum Payment
    {
        Free,
        Paid
    }

    /// <summary>
    /// Фільтр "Вакансія"
    /// </summary>
    public enum VacancyType
    {
        Volunteer,
        Salary
    }

    /// <summary>
    /// Контролер каталогу товарів
    /// </summary>
    public class ListController : Controller
    {
        /// <summary>
        /// Контекст БД
        /// </summary>
        private readonly ApplicationContext context;

        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="context">Контекст БД</param>
        public ListController(ApplicationContext context)
        {
            this.context = context;
        }

        /// <summary>
        /// Виклик List/Index
    }
}
```

```

/// </summary>
/// <returns> Відповідь сервера</returns>
public IActionResult Index()
{
    var viewModel = new ListViewModel();
    var educations = context.GetEducations();
    foreach (var education in educations)
    {
        if (!education.IsAccepted)
            continue;

        var reviews = context.GetReviewsOfEducationByEducation(education);
        double mark = 0.0;
        if (reviews != null && reviews.Any())
            mark = reviews.Average(x => x.Mark);

        if (education.Image == null)
            continue;

        string imageBase64Data =
Convert.ToBase64String(education.Image.ImageData);

        ItemViewModel itemViewModel = new()
        {
            ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
            ItemName = education.EducationName,
            Company = education.Seller.UserName,
            EducationType = education.EducationType,
            StartDate = education.StartDate.Replace("T", "\n"),
            Mark = mark,
            Price = education.Price
        };
        viewModel.Items.Add(itemViewModel);
    }

    var jobs = context.GetJobs();
    foreach (var job in jobs)
    {
        if (!job.IsAccepted)
            continue;

        var reviews = context.GetReviewsOfJobByJob(job);
        double mark = 0.0;
        if (reviews != null && reviews.Any())
            mark = reviews.Average(x => x.Mark);

        if (job.Image == null)
            continue;

        string imageBase64Data =
Convert.ToBase64String(job.Image.ImageData);

        ItemViewModel itemViewModel = new()
        {
            ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
            ItemName = job.JobName,
            Company = job.Owner.UserName,
            EducationType = Models.EducationType.Job,
            Mark = mark,

```

```

        Price = job.Salary
    };
    viewModel.Items.Add(itemViewModel);
}

return View(viewModel);
}

/// <summary>
/// Провести пошук товарів за назвою
/// </summary>
/// <param name="searchString"> Рядок, який може міститися в назві
матеріалу </param>
/// <returns> Відповідь сервера </returns>
public IActionResult Search(string searchString)
{
    var viewModel = new ListViewModel();

    var educations = context.GetEducations();
    var filteredEducations = educations.Where(education =>
education.EducationName.Contains(searchString));
    var itEducation = filteredEducations.GetEnumerator();
    while (itEducation.MoveNext())
    {
        var education = itEducation.Current;
        if (!education.IsAccepted)
            continue;

        var reviews = context.GetReviewsOfEducationByEducation(education) =
double mark = 0.0;
        if (reviews != null && reviews.Any())
            mark = reviews.Average(x => x.Mark);

        if (education.Image == null)
            continue;

        string imageBase64Data =
Convert.ToBase64String(education.Image.ImageData);

        ItemViewModel itemViewModel = new()
        {
            ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
            ItemName = education.EducationName,
            Company = education.Seller.UserName,
            EducationType = education.EducationType,
            StartDate = education.StartDate.Replace("T", "\n"),
            Mark = mark,
            Price = education.Price
        };
        viewModel.Items.Add(itemViewModel);
    }

    var jobs = context.GetJobs();
    var filteredJobs = jobs.Where(j =>
j.JobName.Contains(searchString));
    var itJob = filteredJobs.GetEnumerator();
    while (itJob.MoveNext())
    {
        var job = itJob.Current;
        if (!job.IsAccepted)

```

```

        continue;

        var reviews = context.GetReviewsOfJobByJob(job);
        double mark = 0.0;
        if (reviews != null && reviews.Any())
            mark = reviews.Average(x => x.Mark);

        if (job.Image == null)
            continue;

        string imageBase64Data =
Convert.ToBase64String(job.Image.ImageData);

        ItemViewModel itemViewModel = new()
        {
            ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
            ItemName = job.JobName,
            Company = job.Owner.UserName,
            EducationType = Models.EducationType.Job,
            Mark = mark,
            Price = job.Salary
        };
        viewModel.Items.Add(itemViewModel);
    }

    return View("Index", viewModel);
}

/// <summary>
/// Виклик List/Filter, який застосовує фільтри до товарів
/// </summary>
/// <param name="education_type"> Розділ</param>
/// <param name="free"> Платно чи ні</param>
/// <param name="new_mat"> Початок</param>
/// <param name="duration"> Тривалість</param>
/// <param name="volunteering">Оплата</param>
/// <returns> Відповідь сервера</returns>
public IActionResult Filter(EducationType[] education_type, Payment[]
free
    , Start[] new_mat
    , EducationDuration[] duration, VacancyType[] volunteering)
{
    var viewModel = new ListViewModel();
    viewModel.Types = education_type;
    viewModel.Start = new_mat;
    viewModel.IsFree = free;
    viewModel.Duration = duration;
    viewModel.IsVolunteering = volunteering;

    var educations = context.GetEducations();
    foreach (var education in educations)
    {
        if (!education.IsAccepted)
            continue;

        if (education_type.Any() &&
!education_type.Contains(education.EducationType)) &&
            continue;

        if (free.Any() && free.Length == 1)
        {

```

```

        if (free.Contains(Payment.Free) && education.Price > 0
            || free.Contains(Payment.Paid) && education.Price == 0)
            continue;
    }

    if (new_mat.Any() && !CheckStart(education, new_mat))
        continue;

    if (duration.Any() && !duration.Contains(education.Duration))
        continue;

    if (education.Image == null)
        continue;

    string imageBase64Data =
Convert.ToBase64String(education.Image.ImageData);
    var reviews =
context.GetReviewsOfEducationByEducation(education);
    double mark = 0.0;
    if (reviews != null && reviews.Any())
        mark = reviews.Average(x => x.Mark);

    viewModel.Items.Add(new ItemViewModel()
    {
        ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
        ItemName = education.EducationName,
        Company = education.Seller.UserName,
        EducationType = education.EducationType,
        StartDate = education.StartDate.Replace("T", "\n"),
        Mark = mark,
        Price = education.Price
    });
}

    if (!education_type.Any() ||
education_type.Contains(EducationType.Job))
    {
        var jobs = context.GetJobs();
        foreach (var job in jobs)
        {
            if (!job.IsAccepted)
                continue;

            if (volunteering.Any() && volunteering.Length == 1)
            {
                if (volunteering.Contains(VacancyType.Volunteer) &&
job.Salary > 0
                || volunteering.Contains(VacancyType.Salary) &&
job.Salary == 0)
                    continue;
            }

            var reviews = context.GetReviewsOfJobByJob(job);
            double mark = 0.0;
            if (reviews != null && reviews.Any())
                mark = reviews.Average(x => x.Mark);

            if (job.Image == null)
                continue;
        }
    }
}

```

```

        string imageBase64Data =
Convert.ToBase64String(job.Image.ImageData);
        viewModel.Items.Add(new ItemViewModel()
        {
            ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
            ItemName = job.JobName,
            Company = job.Owner.UserName,
            EducationType = EducationType.Job,
            Mark = mark,
            Price = job.Salary
        });
    }
}

return View("Index", viewModel);
}

/// <summary>
/// Перевірити, що товар підходить до зазначеного фільтру
/// </summary>
/// <param name="education">Товар</param>
/// <param name="newMat">Фільтр</param>
/// <returns>Ознака успішної фільтрації</returns>
private bool CheckStart(Education education, Start[] newMat)
{
    DateTime dateTime = DateTime.ParseExact(education.StartDate,
"yyyy-MM-ddThh:mm", CultureInfo.InvariantCulture);
    DateTime now = DateTime.Now;
    var diff = now - dateTime;
    foreach (var item in newMat)
    {
        switch (item)
        {
            case Start.StartSoon:
                return -7 <= diff.Days && diff.Days < 0;
            case Start.Insight:
                return education.EducationType ==
EducationType.Insight;
            default:
                return false;
        }
    }
    return false;
}
}
}
}

```

Лістинг файлу LaunchSettings.json:

```

{
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:34420",
      "sslPort": 44326
    }
  },
  "profiles": {
    "MarketPlace": {

```

```

        "commandName": "Project",
        "dotnetRunMessages": true,
        "launchBrowser": true,
        "applicationUrl": "https://localhost:7240;http://localhost:5150",
        "environmentVariables": {
            "ASPNETCORE_ENVIRONMENT": "Development"
        }
    },
    "IIS Express": {
        "commandName": "IISExpress",
        "launchBrowser": true,
        "environmentVariables": {
            "ASPNETCORE_ENVIRONMENT": "Development"
        }
    }
}
}
}

```

Лістинг файлу Create.css:

```

/* Dropdown Button */
.dropbtn {
    background-color: #61a1ff;
    color: white;
    padding: 16px;
    font-size: 16px;
    border: none;
}

/* The container <div> - needed to position the dropdown content */
.dropdown {
    position: relative;
    display: inline-block;
}

/* Dropdown Content (Hidden by Default) */
.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f1f1f1;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
}

/* Links inside the dropdown */
.dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
}

.dropdown-content button {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {

```



```

        background-color: #ddd;
    }

    .dropdown-content button:hover {
        background-color: #ddd;
    }

    /* Show the dropdown menu on hover */
    .dropdown:hover .dropdown-content {
        display: block;
    }

    /* Change the background color of the dropdown button when the dropdown content
    is shown */
    .dropdown:hover .dropbtn {
        background-color: #0056d2;
        color: white;
    }

    .preview {
        width: 20vw;
        height: 40vh;
        position: absolute;
        left: 40vw;
        top: 10vh;
        display: none;
        -ms-flex-direction: column;
        -webkit-flex-direction: column;
        flex-direction: column;
        -webkit-justify-content: space-around;
        justify-content: space-around;
        -webkit-align-items: center;
        align-items: center;

        background-color: #aec8ff;
    }

    .preview-form {
        display: flex;
        flex-direction: column;
        justify-content: space-around;
        align-items: center;
        background-color: white;
        box-shadow: 0 0 7px 0;
        border-radius: 15px;
    }

    .preview-form img {
        width: 50%;
    }

```

Лістинг файлу Education.css:

```

.star-rating {
    border:solid 1px #ccc;
    display:flex;
    flex-direction: row-reverse;
    font-size:1.5em;
    justify-content:space-around;
    padding:0 .2em;
    text-align:center;
    width:5em;

```

```

}

.star-rating input {
  display:none;
}

.star-rating label {
  color:#ccc;
  cursor:pointer;
}

.star-rating :checked ~ label {
  color:#f90;
}

.star-rating label:hover,
.star-rating label:hover ~ label {
  color:#fc0;
}

.review {
  box-shadow: 0 0 5px 0;
  border-radius: 25px;
  width: 23%;
  display: flex;
  flex-direction: column;
  justify-content: space-around;
  align-items: center;
  padding: 1%;
  margin-right: 1%;
}

```

Лістинг файлу Home.css:

```

.home_section {
  grid-template-columns: repeat(4, 24.2%);
  -webkit-align-items: center;
  align-items: center;
  width: 100%;
  gap: 1%;
}

```

Лістинг файлу Product.css:

```

/* Style the tab */
.tab {
  overflow: hidden;
  border: 1px solid #ccc;
  background-color: #f1f1f1;
}

/* Style the buttons that are used to open the tab content */
.tab button {
  background-color: inherit;
  float: left;
  border: none;
  outline: none;
  cursor: pointer;
  padding: 14px 16px;
  transition: 0.3s;
}

```

```

    /* Change background color of buttons on hover */
.tab button:hover {
    background-color: #ddd;
}

/* Create an active/current tablink class */
.tab button.active {
    background-color: #ccc;
}

/* Style the tab content */
.tabcontent {
    display: none;
    padding: 6px 12px;
}

```

Лістинг файлу Search\_layout.css:

```

.search_section {
    display: flex;
    -webkit-justify-content: space-evenly;
    justify-content: space-evenly;
    -webkit-align-items: center;
    align-items: center;

    width: 100%;
    height: fit-content;
}

.popup {
    box-shadow: 0 0 2px 0px;
    border-radius: 30px;
    -webkit-align-items: center;
    align-items: center;

    width: 12%;

    margin-right: 3%;
}

.popup:hover {
    cursor: pointer;
    -moz-box-shadow: 0 0 5px 0px;
    -webkit-box-shadow: 0 0 5px 0px;
    box-shadow: 0 0 5px 0px;
}

```

Лістинг файлу Site.css:

```

html {
    font-size: 14px;
}

@media (min-width: 768px) {
    html {
        font-size: 16px;
    }
}

html {
    position: relative;
    min-height: 100%;
}

```

```

}

body {
  margin: 0;
  width: 100vw;
  height: 100vh;

  overflow-x: hidden;
}

main {
  width: 100%;
  height: 100%;
}

th, td {
  border: 1px solid #000000;
}

li {
  list-style-type: none;
}

.button-common {
  background-color: #61a1ff;
  color: white;
}

.button-common:hover {
  background-color: #0056d2;
  color: white;
}

/* Style The Dropdown Button */
.dropbtn {
  background-color: #61a1ff;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
  cursor: pointer;
}

/* The container <div> - needed to position the dropdown content */
.dropdown {
  position: relative;
  display: inline-block;
}

/* Dropdown Content (Hidden by Default) */
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

/* Links inside the dropdown */
.dropdown-content a {
  color: black;

```

```

padding: 12px 16px;
text-decoration: none;
display: block;
}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {
background-color: #f1f1f1
}

/* Show the dropdown menu on hover */
.dropdown:hover .dropdown-content {
display: block;
}

/* Change the background color of the dropdown button when the dropdown content
is shown */
.dropdown:hover .dropbtn {
background-color: #0056d2;
color: white;
}

.quote {
color: #61a1ff;
font-size: 1.2rem;
}

```

Лістинг файлу Admin.js:

```

var sideMenuBtn = null;
var chosenProductName = "";

document.addEventListener('DOMContentLoaded', _ev => {
sideMenuBtn = document.getElementById('side_menu');
if (sideMenuBtn !== null) {
sideMenuBtn.addEventListener('mouseup', onSideMenuClicked);
document.addEventListener('mouseup', e => {
sideMenuBtn.style.display = 'none';
chosenProductName = "";
});
}
});

function onSideMenuClicked(e, obj) {
if (obj === null)
return;
let box = obj.getBoundingClientRect();
let box1 = sideMenuBtn.getBoundingClientRect();

sideMenuBtn.style.display = 'flex';
sideMenuBtn.style.left = `${box.right}px`;
sideMenuBtn.style.top = `${box.bottom - box1.height / 2}px`;
chosenProductName = obj.name;

e.stopPropagation();
}

async function onUpdateProductClicked() {
document.location.href = `/Create?name=${chosenProductName}`;
}

async function onRemoveProductClicked() {

```

```

    let response = await fetch('/Admin/Remove', {
      method: 'DELETE',
      headers: {
        'Content-Type': 'text/plain'
      },
      body: chosenProductName
    });

    if (!response.ok) {
      alert(await response.text());
      return;
    }

    document.location.reload();
  }

  async function onApplyBtnClicked(formName, path) {
    let form = document.forms[formName];
    let i = 0;
    let userData = [];
    while (form[i + 1] !== undefined) {
      if (!form[i].checked) {
        i++;
        continue;
      }

      let name = form[i].name;
      let value = parseInt(form[i].value);
      userData.push({
        'name': name,
        'value': value
      });
      i++;
    }
    userData = JSON.stringify(userData);

    let response = await fetch(path, {
      method: 'DELETE',
      headers: {
        'Content-Type': 'text/plain'
      },
      body: userData
    });

    if (!response.ok) {
      alert(await response.text());
      return;
    }

    document.location.reload();
  }
}

```

Лістинг файлу Auth.js:

```

import { sha256 } from './crypt.js'

document.addEventListener('DOMContentLoaded', ev => {
  let btn = document.getElementById('sign_in_btn');
  if (btn !== null)
    btn.onclick = onSignInPressed;
  btn = document.getElementById('change_pwd_btn');
}

```

```

    if (btn !== null)
        btn.onclick = onChangePasswrPressed;
});

async function onSignInPressed(event) {
    let form = document.forms.auth;
    const userData = {
        'login': form.login.value,
        'passwr': await sha256(form.passwr.value)
    };
    const obj = {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json;charset=utf-8'
        },
        body: JSON.stringify(userData)
    };
    let response = await fetch("/Auth/SignIn", obj);

    if (!response.ok) {
        alert(await response.text());
        return;
    }

    let text = await response.json();

    sessionStorage.setItem("cookie", JSON.stringify({
        'login': userData.login,
        'passwr': userData.passwr,
        'role': text['type'],
        'name': text['name']
    }));

    document.location = "/Home/Index";
}

async function onChangePasswrPressed(event) {
    let form = document.forms.change_pwd;
    const obj = {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json;charset=utf-8'
        },
        body: JSON.stringify({
            'login': form.login.value,
            'passwr': await sha256(form.passwr.value),
            'passwrRepeat': await sha256(form.passwrRepeat.value)
        })
    };
    fetch('/Auth/ChangePwd', obj)
        .then(async response => {
            if (response.ok)
                document.location = '/Auth';
            else {
                throw await response.text();
            }
        })
        .catch(reason => alert(reason));
}

```

Лістинг файлу Basket.js:

```
document.addEventListener('DOMContentLoaded', _ev => {
  let itemsDiv = document.getElementById('items');
  let summaryItemsDiv = document.getElementById('summary_items');
  let summary = 0;

  let basket = JSON.parse(sessionStorage.getItem('basket'));
  basket.forEach(item => {
    let tr = document.createElement('tr');

    for (const [key, value] of Object.entries(item)) {
      if (key === 'login')
        continue;

      let td = document.createElement('td');
      td.classList.add('text-center');
      td.innerText = value;
      tr.appendChild(td);
    }

    tr.lastChild.textContent = tr.lastChild.textContent.replace('T', ' ');

    itemsDiv.appendChild(tr);

    let div = document.createElement('div');
    div.classList.add('d-flex', 'justify-content-between');
    let span = document.createElement('span');
    span.innerText = item['name'];
    div.appendChild(span);

    span = document.createElement('span');
    span.innerText = item['price'];
    div.appendChild(span);

    summaryItemsDiv.appendChild(div);

    summary += item['price'];
  });

  let summarySpan = document.getElementById('summary');
  summarySpan.innerText = summary;
});
```

Лістинг файлу Create.js:

```
var dropdownBtn = null;
var preview = null;

var save_education_btn = null;

var save_job_btn = null;
var submit_job_btn = null;

var submit_btn = null;

document.addEventListener('DOMContentLoaded', e => {
  dropdownBtn = document.getElementById('dropdown_btn');
  preview = document.getElementById('preview');
  submit_btn = document.getElementById('submit');

  save_education_btn = document.getElementById('save_education_btn');
```



```

save_job_btn = document.getElementById('save_job_btn');

const cookie = JSON.parse(sessionStorage.getItem('cookie'));

let items = document.getElementsByName('login');
items.forEach(item => item.value = cookie['login']);
});

document.addEventListener('mouseup', e => {
  if (preview === null)
    return;
  let box = preview.getBoundingClientRect();
  if ((box.x <= e.clientX && e.clientX <= box.x + box.width)
    && (box.y <= e.clientY && e.clientY <= box.y + box.height))
    return;

  preview.style.display = 'none';
  if (save_education_btn !== null) {
    save_education_btn.style.display = 'flex';
  }

  if (save_job_btn !== null) {
    save_job_btn.style.display = 'flex';
  }
});

async function onSaveBtnClickedForEducation(path) {
  if (preview.style.display === 'none') {
    preview.style.display = 'flex';
    return;
  }
  let form = document.forms.educationForm;
  if (form === null)
    return;

  const productData = {
    'name': form.name.value,
    'startDate': form.startDate.value + ":00",
    'targetAudience': form.targetAudience.value,
    'skills': form.skills.value,
    'notes': form.notes.value,
    'type': parseInt(form.type.value),
    'price': parseInt(form.price.value),
    'description': form.description.value,
    'content': form.content.value,
    'supervisor': form.supervisor.value,
    'subtitle': form.subtitle.value,
    'login': JSON.parse(sessionStorage.getItem('cookie'))['login']
  };

  let response = await fetch(path, {
    method: "POST",
    headers: {
      'Content-Type': 'application/json;charset=utf-8'
    },
    body: JSON.stringify(productData)
  });

  if (!response.ok) {
    alert(await response.text());
    return;
  }
}

```

```

    }

    const cookie = JSON.parse(sessionStorage.getItem('cookie'));
    document.location = "/OrgCabinet/ProductsList?login=" + cookie['login'];
}

function onSaveEducationBtnClicked(event) {
    let form = document.forms.educationForm;
    if (!form.checkValidity()) {
        form.reportValidity();
        return;
    }
    event.preventDefault();
    document.getElementById('aside_type').innerText = dropdownBtn.innerText;
    if (form.picture.files.length !== 0)
        document.getElementById('aside_img').src =
URL.createObjectURL(form.picture.files[0]);
    document.getElementById('aside_subtitle').innerText = form.subtitle.value;
    document.getElementById('aside_name').innerText = form.name.value;
    document.getElementById('aside_date').innerHTML =
form.startDate.value.replace('T', "<br />");
    document.getElementById('aside_price').innerText = form.price.value in [
'0', ''] ? "Бесплатно" : form.price.value;

    preview.style.display = 'flex';
    submit_btn.onclick = e => {
        document.forms.educationForm.submit();
    }
}

function onSaveJobBtnClicked() {
    let form = document.forms.jobForm;
    if (!form.checkValidity()) {
        form.reportValidity();
        return;
    }
    event.preventDefault();
    document.getElementById('aside_type').innerText = form.type.value;
    if (form.picture.files.length !== 0)
        document.getElementById('aside_img').src =
URL.createObjectURL(form.picture.files[0]);
    document.getElementById('aside_subtitle').innerText = "Вакансия";
    document.getElementById('aside_name').innerText = form.name.value;
    document.getElementById('aside_date').innerHTML =
form.organization.value.replace('T', "<br />");
    document.getElementById('aside_price').innerText = form.salary.value == ''
? "Волонтерство" : form.salary.value;

    preview.style.display = 'flex';
}

async function onSaveBtnClickedForJob(path) {
    let form = document.forms.jobForm;
    if (form === null)
        return;

    const productData = {
        'name': form.name.value,
        'startDate': form.startDate.value + ":00",
        'endDate': form.endDate.value + ":00",
        'category': form.category.value,
        'salary': parseInt(form.salary.value),

```

```

        'skills': form.skills.value,
        'duration': parseInt(form.duration.value),
        'descr': form.descr.value,
        'obligations': form.obligations.value,
        'conditions': form.conditions.value,
        'requirements': form.requirements.value,
        'login': JSON.parse(sessionStorage.getItem('cookie'))['login']
    };

    let response = await fetch(path, {
        method: "POST",
        headers: {
            'Content-Type': 'application/json;charset=utf-8'
        },
        body: JSON.stringify(productData)
    });

    if (!response.ok) {
        alert(await response.text());
        return;
    }

    const cookie = JSON.parse(sessionStorage.getItem('cookie'));
    document.location = "/OrgCabinet/ProductsList?login=" + cookie['login'];
}

function onEducationBtnClicked(ev) {
    let form = document.forms.educationForm;
    form.style.display = 'flex';
    form = document.forms.jobForm;
    if (form !== undefined)
        form.style.display = 'none';

    dropdownBtn.innerText = ev.target.innerText;
    document.getElementById('mat_type').value =
`$${types.indexOf(ev.target.innerText)}`;

    const cookie = JSON.parse(sessionStorage.getItem('cookie'));
    let items = document.getElementsByName('login');
    items.forEach(item => item.value = cookie['login']);

    submit_btn.onclick = e => {
        document.forms.educationForm.submit();
    }
}

const types = ['Навчання', "Дозвілля", "Інсайт"];

function onJobBtnClicked() {
    let form = document.forms.jobForm;
    form.style.display = 'flex';
    form = document.forms.educationForm;
    if (form !== undefined)
        form.style.display = 'none';

    dropdownBtn.innerText = "Найм";

    const cookie = JSON.parse(sessionStorage.getItem('cookie'));
    let items = document.getElementsByName('login');
    items.forEach(item => item.value = cookie['login']);

    submit_btn.onclick = e => {

```

```

        document.forms.jobForm.submit();
    }
}

```

Лістинг файлу Crypt.js:

```

export async function sha256(message) {
    // encode as UTF-8
    const msgBuffer = new TextEncoder().encode(message);

    // hash the message
    const hashBuffer = await crypto.subtle.digest('SHA-256', msgBuffer);

    // convert ArrayBuffer to Array
    const hashArray = Array.from(new Uint8Array(hashBuffer));

    // convert bytes to hex string
    const hashHex = hashArray.map(b => b.toString(16).padStart(2,
'0')).join('');
    return hashHex;
}

```

Лістинг файлу List.js:

```

Org_cabinet

import { sha256 } from './crypt.js'

var sideMenuBtn = null;

document.addEventListener('DOMContentLoaded', _ev => {
    sideMenuBtn = document.getElementById('side_menu');
    if (sideMenuBtn !== null) {
        sideMenuBtn.addEventListener('mouseup', onSideMenuClicked);
        document.addEventListener('mouseup', e => {
            sideMenuBtn.style.display = 'none';
        });
    }

    let cookie = JSON.parse(sessionStorage.getItem('cookie'));

    let btn = document.getElementById('change_data_btn');
    if (btn !== null)
        btn.onclick = onRegistrateBtnClicked;

    btn = document.getElementById('create_new_product');
    if (btn != null)
        btn.onclick = onAddNewProductBtnClicked;
});

async function onRegistrateBtnClicked() {
    let cookie = JSON.parse(sessionStorage.getItem('cookie'));

    const form = document.forms.frm;
    const userData = {
        "name": form.name.value,
        "login": form.login.value,
        "passwrд": form.passwrд.value.length == 0 ? "" : await
sha256(form.passwrд.value),
        "passwrдRepeat": form.passwrдRepeat.value.length == 0 ? "" : await
sha256(form.passwrдRepeat.value),
        "org": form.org.value,

```

```

        "group": form.group.value,
        "phone": form.phone.value,
        "position": form.position.value,
        "accountType": cookie['role'] == 'Student' ? 0 : 1
    };

    let response = await fetch("/OrgCabinet/UpdateData", {
        method: "POST",
        headers: {
            'Content-Type': 'application/json;charset=utf-8'
        },
        body: JSON.stringify(userData)
    });

    if (!response.ok) {
        alert(await response.text());
        return;
    }

    document.location.reload();
}

function onAddNewProductBtnClicked(e) {
    document.location = "/Create?type=0";
}

function onSideMenuClicked(e) {
    let x = e.screenX;
    let y = e.screenY;

    sideMenuBtn.style.display = 'flex';
    sideMenuBtn.style.x = `${x}`;
    sideMenuBtn.style.y = `${y}`;

    e.stopPropagation();
}

```

Лістинг файлу Org.cabinet\_handlers.js:

```

var sideMenuBtn = null;
var chosenProductName = "";

document.addEventListener('DOMContentLoaded', _ev => {
    sideMenuBtn = document.getElementById('side_menu');
    if (sideMenuBtn !== null) {
        sideMenuBtn.addEventListener('mouseup', onSideMenuClicked);
        document.addEventListener('mouseup', e => {
            sideMenuBtn.style.display = 'none';
            chosenProductName = "";
        });
    }
});

function onSideMenuClicked(e, obj) {
    if (obj === null)
        return;
    let box = obj.getBoundingClientRect();
    let box1 = sideMenuBtn.getBoundingClientRect();

    sideMenuBtn.style.display = 'flex';
    sideMenuBtn.style.left = `${box.right}px`;
    sideMenuBtn.style.top = `${box.bottom - box1.height / 2}px`;
}

```

```

        chosenProductName = obj.name;

        e.stopPropagation();
    }

    async function onUpdateProductClicked() {
        document.location.href = `/Create?name=${chosenProductName}`;
    }

    async function onRemoveProductClicked() {
        let response = await fetch('/OrgCabinet/Remove', {
            method: 'DELETE',
            headers: {
                'Content-Type': 'text/plain'
            },
            body: chosenProductName
        });

        if (!response.ok) {
            alert(await response.text());
            return;
        }

        document.location.reload();
    }

    async function onRemoveNotificationsClicked() {
        const cookie = JSON.parse(sessionStorage.getItem('cookie'));
        let response = await fetch(`/OrgCabinet/RemoveNotifications?login=${cookie['login']}`, {
            method: "DELETE"
        });

        if (!response.ok) {
            alert(await response.text());
            return;
        }

        document.location = `/OrgCabinet/Notifications?login=${cookie['login']}`;
    }

```

Лістинг файлу Pay.js:

```

async function onPayBtnClicked() {
    const basket = sessionStorage.getItem('basket');
    if (basket.length === 2)
        return;

    let response = await fetch('/Pay/Pay', {
        method: "POST",
        headers: {
            'Content-Type': 'application/json;charset=utf-8'
        },
        body: basket
    });

    if (!response.ok) {
        alert(await response.text());
        return;
    }

    let parsed = JSON.parse(basket);

```

```

    sessionStorage.setItem('basket', '[]');
    document.location = `/Home/Index?login=${parsed[0]['login']}`;
}

```

Лістинг файлу Product.js:

```

async function onPayPressed(productName, login, type, startDate, price) {
    let basket = JSON.parse(sessionStorage.getItem('basket'));
    basket.push({
        'name': productName,
        'login': login,
        'type': type,
        'price': parseInt(price),
        'startDate': startDate
    });
    sessionStorage.setItem('basket', JSON.stringify(basket));
}

async function onOpinionBtnClicked(productName, login) {
    const opinionTextarea = document.getElementById('opinion_textarea');
    const form = document.forms.ratio;
    let response = await fetch('/Product/AddOpinion', {
        method: "POST",
        headers: {

        },
        body: JSON.stringify({
            'login': login,
            'name': productName,
            'opinion': opinionTextarea.value,
            'mark': getCheched(form)
        })
    });

    if (!response.ok) {
        alert(await response.text());
        return;
    }

    document.location.reload();
}

function getCheched(form) {
    for (let i = 0; form[i] != undefined; i++) {
        if (form[i].checked)
            return 5 - i;
    }

    return 0;
}

function openTab(evt, cityName) {
    // Declare all variables
    let i, tabcontent, tablinks;

    // Get all elements with class="tabcontent" and hide them
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }

    // Get all elements with class="tablinks" and remove the class "active"

```

```

tablinks = document.getElementsByClassName("tablinks");
for (i = 0; i < tablinks.length; i++) {
    tablinks[i].className = tablinks[i].className.replace(" active", "");
}

// Show the current tab, and add an "active" class to the button that opened
the tab
document.getElementById(cityName).style.display = "block";
evt.currentTarget.className += " active";
}

```

Лістинг файлу Reg.js:

```

import { sha256 } from './crypt.js'

document.addEventListener('DOMContentLoaded', e => {
    let btns = document.getElementsByTagName('button');
    btns.item(btns.length - 1).onclick = onRegistrateBtnClicked;
});

async function onRegistrateBtnClicked() {
    const form = document.forms.frm;
    const userData = {
        "name": form.name.value,
        "login": form.login.value,
        "passwd": await sha256(form.passwd.value),
        "passwdRepeat": await sha256(form.passwdRepeat.value),
        "org": form.org.value,
        "group": form.group.value,
        "phone": form.phone.value,
        "position": form.position.value,
        "accountType": parseInt(form.accountType.value)
    };

    let response = await fetch("Reg/Registrate", {
        method: "POST",
        headers: {
            'Content-Type': 'application/json;charset=utf-8'
        },
        body: JSON.stringify(userData)
    });

    if (!response.ok) {
        alert(await response.text());
        return;
    }

    document.location = "/Auth";
}

```

Лістинг файлу site.js:

```

document.addEventListener('DOMContentLoaded', async event => {
    let basket = sessionStorage.getItem('basket');
    if (basket === null)
        sessionStorage.setItem("basket", JSON.stringify([]));
    let cookie = sessionStorage.getItem("cookie");
    if (cookie === null)
        return;

    let enterBtn = document.getElementById('enter_btn');
    enterBtn.style.visibility = 'hidden';
}

```



```

let userCabinetDiv = document.getElementById('user_cabinet_div');
userCabinetDiv.style.visibility = 'visible';

let cookieObj = JSON.parse(cookie);

setList(cookieObj);

let userCabinetExit = document.getElementById('user_cabinet_exit');
userCabinetExit.onclick = e => {
    sessionStorage.removeItem('cookie');
    document.location.reload();
};
});

function setList(cookie) {
    let dropdownContent = document.getElementById('dropdown_content');
    if (dropdownContent === null)
        return;

    let dropbtn = document.getElementById('dropbtn');
    if (dropbtn === null)
        return;

    dropbtn.innerText = cookie['name'];

    let role = cookie['role'];
    if (role === 'Student') {
        let a = document.createElement('a');
        a.href = `/Basket/Index?login=${cookie['login']}`;
        a.innerText = " Кошик";
        dropdownContent.insertAdjacentElement('afterbegin', a);

        a = document.createElement('a');
        a.href = `/UserCabinet/Cabinet?login=${cookie['login']}`;
        a.innerText = " Профіль";
        dropdownContent.insertAdjacentElement('afterbegin', a);

        a = document.createElement('a');
        a.href = `/UserCabinet/ProductsList?login=${cookie['login']}`;
        a.innerText = " Список куплених матеріалів";
        dropdownContent.insertAdjacentElement('afterbegin', a);
    }
    else if (role === 'Organizator') {
        let a = document.createElement('a');
        a.href = `/Create/Index?login=${cookie['login']}`;
        a.innerText = " Створити новий матеріал";
        dropdownContent.insertAdjacentElement('afterbegin', a);

        a = document.createElement('a');
        a.href = `/OrgCabinet/Notifications?login=${cookie['login']}`;
        a.innerText = " Відгуки";
        dropdownContent.insertAdjacentElement('afterbegin', a);

        a = document.createElement('a');
        a.href = `/OrgCabinet/Cabinet?login=${cookie['login']}`;
        a.innerText = " Профіль";
        dropdownContent.insertAdjacentElement('afterbegin', a);

        a = document.createElement('a');
        a.href = `/OrgCabinet/ProductsList?login=${cookie['login']}`;
        a.innerText = " Список матеріалів";
    }
}

```

```

        dropdownContent.insertAdjacentElement('afterbegin', a);
    }
    else if (role === 'Admin') {
        let a = document.createElement('a');
        a.href = `/Admin/UsersList?login=${cookie['login']}`;
        a.innerText = " Список користувачів";
        dropdownContent.insertAdjacentElement('afterbegin', a);

        a = document.createElement('a');
        a.href = `/Admin/AccDecl?login=${cookie['login']}`;
        a.innerText = " Список товарів і послуг для підтвердження/відхилення";
        dropdownContent.insertAdjacentElement('afterbegin', a);

        a = document.createElement('a');
        a.href = `/Admin/ProductsList?login=${cookie['login']}`;
        a.innerText = " Список товарів і послуг";
        dropdownContent.insertAdjacentElement('afterbegin', a);
    }
    else {
        return;
    }
}

```

Лістинг файлу Site\_handlers.js:

```

function onItemClick(name) {
    let url = `/Product?name=${name}`;

    const cookie = sessionStorage.getItem('cookie');
    if (cookie !== undefined && cookie !== null) {
        url = `${url}&login=${JSON.parse(cookie)['login']}`;
    }

    document.location = url;
}

```

Лістинг файлу User\_cabinet.js:

```

import { sha256 } from './crypt.js'

document.addEventListener('DOMContentLoaded', ev => {
    let cookie = JSON.parse(sessionStorage.getItem('cookie'));

    let btn = document.getElementById('change_data_btn');
    if (btn !== null)
        btn.onclick = onRegistrateBtnClicked;
});

async function onRegistrateBtnClicked() {
    let cookie = JSON.parse(sessionStorage.getItem('cookie'));

    const form = document.forms.frm;
    const userData = {
        "name": form.name.value,
        "login": form.login.value,
        "passwrд": form.passwrд.value.length == 0 ? "" : await
sha256(form.passwrд.value),
        "passwrдRepeat": form.passwrдRepeat.value.length == 0 ? "" : await
sha256(form.passwrдRepeat.value),
        "org": form.org.value,

```

```

        "group": form.group.value,
        "phone": form.phone.value,
        "position": form.position.value,
        "accountType": cookie['role'] == 'Student' ? 0 : 1
    };

    let response = await fetch("/UserCabinet/UpdateData", {
        method: "POST",
        headers: {
            'Content-Type': 'application/json;charset=utf-8'
        },
        body: JSON.stringify(userData)
    });

    if (!response.ok) {
        alert(await response.text());
        return;
    }

    document.location.reload();
}

```

Лістинг файлу AdminController.cs:

```

using Marketplace.Extra;
using Marketplace.Models;
using Marketplace.ViewModels;
using Microsoft.AspNetCore.Mvc;
using System.Collections;
using System.Text.Json;

namespace Marketplace.Controllers
{
    /// <summary>
    /// Контролер адмін-панелі
    /// </summary>
    public class AdminController : Controller
    {
        /// <summary>
        /// Контекст БД
        /// </summary>
        private ApplicationDbContext context;

        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="context">Контекст БД</param>
        public AdminController(ApplicationDbContext context)
        {
            this.context = context;
        }

        /// <summary>
        /// Виклик Admin/Index
        /// </summary>
        /// <returns>Відповідь сервера</returns>
        public IActionResult Index()
        {
            return View();
        }

        /// <summary>

```

```

/// Виклик Admin/Aside
/// </summary>
/// <returns> Відповідь сервера</returns>
public IActionResult Aside()
{
    return PartialView();
}

/// <summary>
/// Виклик Admin/ProductsList
/// </summary>
/// <param name="type"> Тип матеріалу, який необхідно
відобразити</param>
/// <returns> Відповідь сервера</returns>
public IActionResult ProductsList(EducationType type)
{
    var items = new ListViewModel();
    items.Type = type;

    if (type == EducationType.Job)
    {
        var jobs = context.GetJobs();
        foreach (var job in jobs)
        {
            items.Items.Add(new()
            {
                ItemName = job.JobName,
                Company = job.Owner.UserName,
                Category = job.Employment,
                Status = job.IsAccepted ? "Схвалено" : (job.IsDeclined
? " Відхилено" : " Очікує на перевірку"),
                Organization = job.Organization
            });
        }
    }
    else
    {
        var products = context.GetEducations();
        foreach (var product in products)
        {
            if (product.EducationType != type)
                continue;

            items.Items.Add(new()
            {
                ItemName = product.EducationName,
                Company = product.Seller.UserName,
                StartDate = product.StartDate,
                Status = product.IsAccepted ? "Одобрено" :
(product.IsDeclined ? "Отклонено" : "Ожидает проверку"),
                Duration = product.Notes
            });
        }
    }

    return View(items);
}

/// <summary>
/// Виклик Admin/AccDecl
/// </summary>

```

```

        /// <param name="type"> Тип матеріалу, який необхідно
відобразити</param>
        /// <returns> Відповідь сервера</returns>
        public IActionResult AccDecl(EducationType type)
        {
            var products = context.GetEducationsNotAccepted();
            var items = new ListViewModel();
            items.Type = type;
            if (type == EducationType.Job)
            {
                var jobs = context.GetJobsNotAccepted();
                foreach (var job in jobs)
                {
                    if (job.IsDeclined)
                        continue;

                    items.Items.Add(new()
                    {
                        ItemName = job.JobName,
                        Company = job.Owner.UserName,
                        Category = job.Employment,
                        Status = job.IsAccepted ? " Схвалено" : (job.IsDeclined
? " Відхилено" : " Очікує на перевірку"),
                        Organization = job.Organization
                    });
                }
            }
            else
            {
                foreach (var product in products)
                {
                    if (product.IsDeclined)
                        continue;

                    items.Items.Add(new()
                    {
                        ItemName = product.EducationName,
                        Company = product.Seller.UserName,
                        StartDate = product.StartDate,
                        Status = product.IsAccepted ? " Схвалено" :
(product.IsDeclined ? " Відхилено" : " Очікує на перевірку"),
                        Duration = product.Notes
                    });
                }
            }

            return View(items);
        }

        /// <summary>
        /// Видалити матеріал
        /// </summary>
        /// <param name="name"> Назва матеріалу</param>
        /// <param name="type"> Тип матеріалу</param>
        /// <returns> Відповідь сервера</returns>
        public IActionResult Remove(string name, EducationType type)
        {
            var product = context.GetEducationByName(name);
            if (product != null)
            {
                if (!context.RemoveEntity(product))
                    return BadRequest("Wrong data");
            }
        }
    }
}

```

```

    }

    var job = context.GetJobByName(name);
    if (job != null)
    {
        if (!context.RemoveEntity(job))
            return BadRequest("Wrong data");
    }

    return RedirectToAction("PoductList", "Admin", type);
}

/// <summary>
/// Схвалити/відхилити навчання, дозвілля або інсайт
/// </summary>
/// <returns> Відповідь сервера</returns>
public async Task<IActionResult> AcceptDeclineEducation()
{
    string json = await new
StreamReader(Request.Body).ReadToEndAsync();
    var items = JsonSerializer.Deserialize<List<AccDeclRecord>>(json);
    if (items == null)
        return BadRequest("Wrong data");

    List<Education> educations = new();
    foreach (var item in items)
    {
        if (item == null)
            continue;

        var itEducation = context.GetEducationByName(item.name);
        if (itEducation != null)
        {
            itEducation.IsAccepted = item.value == 1;
            itEducation.IsDeclined = item.value == 0;
            educations.Add(itEducation);
            continue;
        }
    }

    if (context.UpdateEntities(educations))
    {
        return Ok();
    }

    return BadRequest("Wrong data");
}

/// <summary>
/// Схвалити/відхилити найм
/// </summary>
/// <returns> Відповідь сервера</returns>
public async Task<IActionResult> AcceptDeclineJob()
{
    string json = await new
StreamReader(Request.Body).ReadToEndAsync();
    var items = JsonSerializer.Deserialize<List<AccDeclRecord>>(json);
    if (items == null)
        return BadRequest("Wrong data");

    List<Job> jobs = new();
    foreach (var item in items)

```

```

    {
        if (item == null)
            continue;

        var itJob = context.GetJobByName(item.name);
        if (itJob != null)
        {
            itJob.IsAccepted = item.value == 1;
            itJob.IsDeclined = item.value == 0;
            jobs.Add(itJob);
            continue;
        }
    }

    if (context.UpdateEntities(jobs))
    {
        return Ok();
    }

    return BadRequest("Wrong data");
}

/// <summary>
/// Виклик Admin/UsersList
/// </summary>
/// <param name="isUser"> Відображаємо список покупців</param>
/// <returns> Відповідь сервера</returns>
public IActionResult UsersList(bool isUser)
{
    var viewModel = new AdminUserViewModel();
    viewModel.IsUser = isUser;
    if (viewModel.IsUser)
    {
        var users = context.GetUsers();
        foreach (var user in users)
        {
            viewModel.Users.Add(new(user));
        }
    }
    else
    {
        var users = context.GetSellers();
        foreach (var user in users)
        {
            viewModel.Users.Add(new(user));
        }
    }
    return View(viewModel);
}

/// <summary>
/// Видалити користувача
/// </summary>
/// <param name="login"> Логін користувача</param>
/// <param name="isCustomer"> Ознака того, що ми видаляємо
покупця</param>
/// <returns></returns>
public IActionResult RemoveUser(string login, bool isCustomer)
{
    if (isCustomer)
    {
        var user = context.GetUserByLogin(login);
    }
}

```

```

        if (user == null)
            return RedirectToAction("UsersList", "Admin", isCustomer);
        context.RemoveEntity(user);
    }
    else
    {
        var job = context.GetJobByName(login);
        if (job == null)
            return RedirectToAction("UsersList", "Admin", isCustomer);
        context.RemoveEntity(job);
    }
    return RedirectToAction("UsersList", "Admin", isCustomer);
}
}
}
}
}

```

Лістинг файлу AuthController.cs:

```

using MarketPlace.Extra;
using MarketPlace.Models;
using Microsoft.AspNetCore.Mvc;
using System.Text.Json;

namespace MarketPlace.Controllers
{
    /// <summary>
    /// Контролер аутентифікації
    /// </summary>
    public class AuthController : Controller
    {
        /// <summary>
        /// Контекст БД
        /// </summary>
        private ApplicationDbContext context;

        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="context">Контекст БД</param>
        public AuthController(ApplicationContext context)
        {
            this.context = context;
        }

        /// <summary>
        /// Виклик Auth/Index
        /// </summary>
        /// <returns>Відповідь сервера</returns>
        public IActionResult Index()
        {
            return View();
        }

        /// <summary>
        /// Виклик Auth/SignIn
        /// </summary>
        /// <returns>Відповідь сервера</returns>
        public async Task<IActionResult> SignIn()
        {
            string json = await new
StreamReader(Request.Body).ReadToEndAsync();

```



```

        SignInRecord? record =
JsonSerializer.Deserialize<SignInRecord>(json);
        if (record == null)
            return NotFound("Невірний логін або пароль");

        UserType type = UserType.Count;
        if (!context.CheckUser(record.login, record.passwrд, out type, out
string name))
            return NotFound("Невірний логін або пароль");

        json = JsonSerializer.Serialize(new Dictionary<string, string>()
        {
            ["type"] = type.ToString(),
            ["name"] = name
        });

        return Ok(json);
    }

    /// <summary>
    /// Виклик Auth/ResetPwd
    /// </summary>
    /// <returns> Відповідь сервера</returns>
    public IActionResult ResetPwd()
    {
        return View();
    }

    /// <summary>
    /// Виклик Auth/ChangePwd
    /// </summary>
    /// <returns> Відповідь сервера</returns>
    public async Task<IActionResult> ChangePwd()
    {
        string json = await new
StreamReader(Request.Body).ReadToEndAsync();
        var record = JsonSerializer.Deserialize<ChangePwdRecord>(json);
        if (record == null)
            return NotFound("Неправильні логін або пароль або ви зробили
помилку при повторенні пароля");

        if (!record.passwrд.Equals(record.passwrдRepeat))
            return NotFound("Неправильні логін або пароль або ви зробили
помилку при повторенні пароля");

        UserType type = UserType.Count;
        if (!context.UpdatePasswrд(record.login, record.passwrд))
            return NotFound("Неправильні логін або пароль або ви зробили
помилку при повторенні пароля");

        return Ok(type.ToString());
    }
}
}
}

```

Лістинг файлу BusketController.cs:

```

using Microsoft.AspNetCore.Mvc;

namespace MarketPlace.Controllers
{
    /// <summary>

```

```

    /// Контролер кошика
    /// </summary>
    public class BasketController : Controller
    {
        /// <summary>
        /// Виклик Basket/Index
        /// </summary>
        /// <returns> Відповідь сервера</returns>
        public IActionResult Index()
        {
            return View();
        }
    }
}

```

Лістинг файлу CreateController.cs:

```

using MarketPlace.Extra;
using MarketPlace.Models;
using MarketPlace.ViewModels;
using Microsoft.AspNetCore.Mvc;
using System.Text.Json;

namespace MarketPlace.Controllers
{
    /// <summary>
    /// Контролер сторінки створення матеріалу
    /// </summary>
    public class CreateController : Controller
    {
        /// <summary>
        /// Контекст БД
        /// </summary>
        private ApplicationContext context;

        /// <summary>
        /// ID редагованого матеріалу
        /// </summary>
        private static int idOfChanged = -1;

        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="context">Контекст БД</param>
        public CreateController(ApplicationContext context)
        {
            this.context = context;
        }

        /// <summary>
        /// Виклик Create/Index
        /// </summary>
        /// <param name="name"> Назва матеріалу</param>
        /// <returns> Відповідь сервера</returns>
        public IActionResult Index(string name)
        {
            image = null;
            var education = context.GetEducationByName(name);
            var viewModel = new CreateViewModel();
            string imageBase64Data;
            if (education != null)
            {

```

```

        viewModel.Education = new EducationVM(education);
        idOfChanged = education.Id;

        imageBase64Data
Convert.ToBase64String(education.Image.ImageData);

        viewModel.Education.ImageURL
string.Format("data:image/jpg;base64,{0}", imageBase64Data);

        return View(viewModel);
    }

    var job = context.GetJobByName(name);
    if (job == null)
        return View(viewModel);

    idOfChanged = job.Id;
    viewModel.Job = new JobVM(job);

    imageBase64Data = Convert.ToBase64String(job.Image.ImageData);

    viewModel.Job.ImageURL
string.Format("data:image/jpg;base64,{0}", imageBase64Data);

    return View(viewModel);
}

/// <summary>
/// Зберегти найм
/// </summary>
/// <param name="record"> Інформація щодо найму</param>
/// <returns> Відповідь сервера</returns>
public IActionResult SaveJob([FromForm]JobRecord record)
{
    Seller? owner = context.GetSellerByLogin(record.login);
    if (owner == null)
        return BadRequest("Wrong data");

    Job job = new(record);
    job.Owner = owner;
    job.Link = "";

    if (idOfChanged == -1)
    {
        job.Image = image;
        if (!context.AddNewEntity(job))
            return BadRequest("Wrong data");
    }
    else
    {
        job.Id = idOfChanged;
        if (!context.UpdateEntity(job))
            return BadRequest("Wrong data");
    }

    idOfChanged = -1;
    return RedirectToAction("Index", "Home");
}

/// <summary>
/// Оновити роботу
/// </summary>

```

```

    /// <returns> Відповідь сервера</returns>
    public async Task<IActionResult> UpdateJob()
    {
        string json = await new
StreamReader(Request.Body).ReadToEndAsync();
        var record = JsonSerializer.Deserialize<JobRecord>(json);
        if (record == null)
            return BadRequest("Wrong data");

        Seller? owner = context.GetSellerByLogin(record.login);
        if (owner == null)
            return BadRequest("Wrong data");

        Job job = new(record);
        job.Owner = owner;

        if (!context.UpdateEntity(job))
            return BadRequest("Wrong data");

        return Ok();
    }

    /// <summary>
    /// Картинка поточного матеріалу
    /// </summary>
    private static Image? image = null;

    /// <summary>
    /// Зберегти навчання, дозвілля або інсайт
    /// </summary>
    /// <param name="record"> Інформація про навчання, дозвілля або інсайт
</param>
    /// <returns> Відповідь сервера</returns>
    public IActionResult SaveEducation([FromForm]EducationRecord record)
    {
        Seller? owner = context.GetSellerByLogin(record.login);
        if (owner == null)
            return BadRequest("Wrong data");

        Education education = new(record);
        if (record.link == null)
            education.Link = "";
        education.Seller = owner;

        if (idOfChanged == -1)
        {
            education.Image = image;
            if (!context.AddNewEntity(education))
                return BadRequest("Wrong data");
        }
        else
        {
            education.Id = idOfChanged;
            if (!context.UpdateEntity(education))
                return BadRequest("Wrong data");
        }

        idOfChanged = -1;
        return RedirectToAction("Index", "Home");
    }

    /// <summary>

```

```

/// Оновити навчання, дозвілля або інсайт
/// </summary>
/// <param name="record"> Інформація про навчання, дозвілля або інсайт
</param>
/// <returns> Відповідь сервера</returns>
record)
public IActionResult UpdateEducation([FromForm] EducationRecord
{
    Seller? owner = context.GetSellerByLogin(record.login);
    if (owner == null)
        return BadRequest("Wrong data");

    Education education = new(record);
    education.Id = idOfChanged;
    education.Seller = owner;

    if (!context.UpdateEntity(education))
        return BadRequest("Wrong data");

    return Ok();
}

/// <summary>
/// Відкрити вікно попереднього перегляду навчання, дозвілля або
інсайту
/// </summary>
/// <param name="record"> Інформація про навчання, дозвілля або інсайт
</param>
/// <returns> Відповідь сервера</returns>
public IActionResult Education([FromForm] EducationRecord record)
{
    Seller? owner = context.GetSellerByLogin(record.login);
    if (owner == null)
        return BadRequest("Wrong data");

    Education education = new(record);
    education.Seller = owner;

    if (record.picture != null)
    {
        image = new();
        {
            MemoryStream memoryStream = new MemoryStream();
            record.picture.CopyTo(memoryStream);
            image.ImageData = memoryStream.ToArray();

            memoryStream.Close();
            memoryStream.Dispose();
        }
    }

    var viewModel = new CreateViewModel();

    viewModel.Education = new EducationVM(education);
    return View(viewModel);
}

/// <summary>
/// Відкрити вікно попереднього перегляду найму
/// </summary>
/// <param name="record"> Інформація про найм</param>
/// <returns> Відповідь сервера</returns>

```

```

public IActionResult Job([FromForm] JobRecord record)
{
    Seller? owner = context.GetSellerByLogin(record.login);
    if (owner == null)
        return BadRequest("Wrong data");

    Job job = new(record);
    job.Owner = owner;

    image = new();
    {
        MemoryStream memoryStream = new MemoryStream();
        record.picture.CopyTo(memoryStream);
        image.ImageData = memoryStream.ToArray();

        memoryStream.Close();
        memoryStream.Dispose();
    }

    var viewModel = new CreateViewModel();

    viewModel.Job = new JobVM(job);

    return View(viewModel);
}
}
}

```

Лістинг файлу HomeController.cs:

```

using Marketplace.Extra;
using Marketplace.ViewModels;
using Microsoft.AspNetCore.Mvc;
using System.Diagnostics;

namespace Marketplace.Controllers
{
    /// <summary>
    /// Контролер головної сторінки
    /// </summary>
    public class HomeController : Controller
    {
        /// <summary>
        /// Контекст БД
        /// </summary>
        private readonly ApplicationContext context;

        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="context">Контекст БД</param>
        public HomeController(ApplicationContext context)
        {
            this.context = context;
        }

        /// <summary>
        /// Виклик Home/Index
        /// </summary>
        /// <returns> Відповідь сервера</returns>
        public IActionResult Index()
        {

```

```

var products = context.GetEducations();
ItemsListViewModel itemsListViewModel = new();
foreach (var product in products)
{
    if (!product.IsAccepted)
        continue;

    var reviews = context.GetReviewsOfEducationByEducation(product);
    double mark = 0.0;
    if (reviews != null && reviews.Any())
        mark = reviews.Average(x => x.Mark);

    string imageBase64Data =
Convert.ToBase64String(product.Image.ImageData);

    ItemViewModel itemViewModel = new()
    {
        ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
        ItemName = product.EducationName,
        Company = product.Seller.UserName,
        EducationType = product.EducationType,
        StartDate = product.StartDate.Replace("T", "\n"),
        Mark = mark,
        Price = product.Price
    };

    switch (product.EducationType)
    {
        case Models.EducationType.Education:
            itemsListViewModel.Educations.Add(itemViewModel);
            break;
        case Models.EducationType.Leisure:
            itemsListViewModel.Leisures.Add(itemViewModel);
            break;
        case Models.EducationType.Insight:
            itemsListViewModel.Insights.Add(itemViewModel);
            break;
        default:
            break;
    }
}

var jobs = context.GetJobs();
foreach (var job in jobs)
{
    if (!job.IsAccepted)
        continue;

    var reviews = context.GetReviewsOfJobByJob(job);
    double mark = 0.0;
    if (reviews != null && reviews.Any())
        mark = reviews.Average(x => x.Mark);

    string imageBase64Data =
Convert.ToBase64String(job.Image.ImageData);

    ItemViewModel itemViewModel = new()
    {
        ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),

```

```

        ItemName = job.JobName,
        Company = job.Owner.UserName,
        EducationType = Models.EducationType.Job,
        Mark = mark,
        Price = job.Salary
    };

    itemsListViewModel.Jobs.Add(itemViewModel);
}

return View(itemsListViewModel);
}

[HttpPost]
public IActionResult Index(string searchString)
{
    return View();
}
}
}

```

Лістинг файлу ListController.cs:

```

using Marketplace.Extra;
using Marketplace.Models;
using Marketplace.ViewModels;
using Microsoft.AspNetCore.Mvc;
using System.Globalization;

namespace Marketplace.Controllers
{
    /// <summary>
    /// Фільтр "Початок"
    /// </summary>
    public enum Start
    {
        StartSoon,
        Insight
    }

    /// <summary>
    /// Фільтр "Платний чи ні"
    /// </summary>
    public enum Payment
    {
        Free,
        Paid
    }

    /// <summary>
    /// Фільтр "Вакансія"
    /// </summary>
    public enum VacancyType
    {
        Volunteer,
        Salary
    }

    /// <summary>
    /// Контролер каталогу товарів
    /// </summary>
    public class ListController : Controller

```



```

{
    /// <summary>
    /// Контекст БД
    /// </summary>
    private readonly ApplicationContext context;

    /// <summary>
    /// Конструктор
    /// </summary>
    /// <param name="context">Контекст БД</param>
    public ListController(ApplicationContext context)
    {
        this.context = context;
    }

    /// <summary>
    /// Виклик List/Index
    /// </summary>
    /// <returns>Відповідь сервера</returns>
    public IActionResult Index()
    {
        var viewModel = new ListViewModel();
        var educations = context.GetEducations();
        foreach (var education in educations)
        {
            if (!education.IsAccepted)
                continue;

            var reviews = context.GetReviewsOfEducationByEducation(education);
            double mark = 0.0;
            if (reviews != null && reviews.Any())
                mark = reviews.Average(x => x.Mark);

            if (education.Image == null)
                continue;

            string imageBase64Data = Convert.ToBase64String(education.Image.ImageData);

            ItemViewModel itemViewModel = new()
            {
                ImageURL = string.Format("data:image/jpg;base64,{0}",
                    imageBase64Data),
                ItemName = education.EducationName,
                Company = education.Seller.UserName,
                EducationType = education.EducationType,
                StartDate = education.StartDate.Replace("T", "\n"),
                Mark = mark,
                Price = education.Price
            };
            viewModel.Items.Add(itemViewModel);
        }

        var jobs = context.GetJobs();
        foreach (var job in jobs)
        {
            if (!job.IsAccepted)
                continue;

            var reviews = context.GetReviewsOfJobByJob(job);
            double mark = 0.0;

```

```

        if (reviews != null && reviews.Any())
            mark = reviews.Average(x => x.Mark);

        if (job.Image == null)
            continue;

        string imageBase64Data =
Convert.ToBase64String(job.Image.ImageData);

        ItemViewModel itemViewModel = new()
        {
            ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
            ItemName = job.JobName,
            Company = job.Owner.UserName,
            EducationType = Models.EducationType.Job,
            Mark = mark,
            Price = job.Salary
        };
        viewModel.Items.Add(itemViewModel);
    }

    return View(viewModel);
}

/// <summary>
/// Провести пошук товарів за назвою
/// </summary>
/// <param name="searchString"> Рядок, який може міститися в назві
матеріалу </param>
/// <returns> Відповідь сервера</returns>
public IActionResult Search(string searchString)
{
    var viewModel = new ListViewModel();

    var educations = context.GetEducations();
    var filteredEducations = educations.Where(education =>
education.EducationName.Contains(searchString));
    var itEducation = filteredEducations.GetEnumerator();
    while (itEducation.MoveNext())
    {
        var education = itEducation.Current;
        if (!education.IsAccepted)
            continue;

        var reviews =
context.GetReviewsOfEducationByEducation(education);
        double mark = 0.0;
        if (reviews != null && reviews.Any())
            mark = reviews.Average(x => x.Mark);

        if (education.Image == null)
            continue;

        string imageBase64Data =
Convert.ToBase64String(education.Image.ImageData);

        ItemViewModel itemViewModel = new()
        {
            ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
            ItemName = education.EducationName,

```

```

        Company = education.Seller.UserName,
        EducationType = education.EducationType,
        StartDate = education.StartDate.Replace("T", "\n"),
        Mark = mark,
        Price = education.Price
    };
    viewModel.Items.Add(itemViewModel);
}

var jobs = context.GetJobs();
var filteredJobs = jobs.Where(j =>
j.JobName.Contains(searchString));
var itJob = filteredJobs.GetEnumerator();
while (itJob.MoveNext())
{
    var job = itJob.Current;
    if (!job.IsAccepted)
        continue;

    var reviews = context.GetReviewsOfJobByJob(job);
    double mark = 0.0;
    if (reviews != null && reviews.Any())
        mark = reviews.Average(x => x.Mark);

    if (job.Image == null)
        continue;

    string imageBase64Data =
Convert.ToBase64String(job.Image.ImageData);

    ItemViewModel itemViewModel = new()
    {
        ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
        ItemName = job.JobName,
        Company = job.Owner.UserName,
        EducationType = Models.EducationType.Job,
        Mark = mark,
        Price = job.Salary
    };
    viewModel.Items.Add(itemViewModel);
}

return View("Index", viewModel);
}

/// <summary>
/// Виклик List/Filter, який застосовує фільтри до товарів
/// </summary>
/// <param name="education_type"> Розділ </param>
/// <param name="free"> Платно чи ні</param>
/// <param name="new_mat"> Початок</param>
/// <param name="duration"> Тривалість</param>
/// <param name="volunteering"> Оплата</param>
/// <returns> Відповідь сервера</returns>
public IActionResult Filter(EducationType[] education_type, Payment[]
free
, Start[] new_mat
, EducationDuration[] duration, VacancyType[] volunteering)
{
    var viewModel = new ListViewModel();
    viewModel.Types = education_type;

```

```

viewModel.Start = new_mat;
viewModel.IsFree = free;
viewModel.Duration = duration;
viewModel.IsVolunteering = volunteering;

var educations = context.GetEducations();
foreach (var education in educations)
{
    if (!education.IsAccepted)
        continue;

    if (education_type.Any()
!education_type.Contains(education.EducationType))
        continue;

    if (free.Any() && free.Length == 1)
    {
        if (free.Contains(Payment.Free) && education.Price > 0
            || free.Contains(Payment.Paid) && education.Price == 0)
            continue;
    }

    if (new_mat.Any() && !CheckStart(education, new_mat))
        continue;

    if (duration.Any() && !duration.Contains(education.Duration))
        continue;

    if (education.Image == null)
        continue;

    string imageBase64Data =
Convert.ToBase64String(education.Image.ImageData);
    var reviews =
context.GetReviewsOfEducationByEducation(education);
    double mark = 0.0;
    if (reviews != null && reviews.Any())
        mark = reviews.Average(x => x.Mark);

    viewModel.Items.Add(new ItemViewModel()
    {
        ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
        ItemName = education.EducationName,
        Company = education.Seller.UserName,
        EducationType = education.EducationType,
        StartDate = education.StartDate.Replace("T", "\n"),
        Mark = mark,
        Price = education.Price
    });
}

if (!education_type.Any()
education_type.Contains(EducationType.Job))
{
    var jobs = context.GetJobs();
    foreach (var job in jobs)
    {
        if (!job.IsAccepted)
            continue;

        if (volunteering.Any() && volunteering.Length == 1)

```

```

        {
            if (volunteering.Contains(VacancyType.Volunteer) &&
                || volunteering.Contains(VacancyType.Salary) &&
                continue;
        }

        var reviews = context.GetReviewsOfJobByJob(job);
        double mark = 0.0;
        if (reviews != null && reviews.Any())
            mark = reviews.Average(x => x.Mark);

        if (job.Image == null)
            continue;

        string imageBase64Data =
Convert.ToBase64String(job.Image.ImageData);
        viewModel.Items.Add(new ItemViewModel()
        {
            ImageURL = string.Format("data:image/jpg;base64,{0}",
imageBase64Data),
            ItemName = job.JobName,
            Company = job.Owner.UserName,
            EducationType = EducationType.Job,
            Mark = mark,
            Price = job.Salary
        });
    }
}

return View("Index", viewModel);
}

/// <summary>
/// Перевірити, що товар підходить до зазначеного фільтру
/// </summary>
/// <param name="education">Товар</param>
/// <param name="newMat">Фільтр </param>
/// <returns>Ознака успішної фільтрації</returns>
private bool CheckStart(Education education, Start[] newMat)
{
    DateTime dateTime = DateTime.ParseExact(education.StartDate,
"yyyy-MM-ddThh:mm", CultureInfo.InvariantCulture);
    DateTime now = DateTime.Now;
    var diff = now - dateTime;
    foreach (var item in newMat)
    {
        switch (item)
        {
            case Start.StartSoon:
                return -7 <= diff.Days && diff.Days < 0;
            case Start.Insight:
                return education.EducationType ==
EducationType.Insight;
            default:
                return false;
        }
    }

    return false;
}
}

```

```
}  
}
```

Лістинг файлу OrgCabinetController.cs:

```
using MarketPlace.Extra;  
using MarketPlace.Models;  
using MarketPlace.ViewModels;  
using Microsoft.AspNetCore.Mvc;  
using System.Text.Json;  
  
namespace MarketPlace.Controllers  
{  
    /// <summary>  
    /// Контролер вікна кабінету продавця  
    /// </summary>  
    public class OrgCabinetController : Controller  
    {  
        /// <summary>  
        /// Контекст БД  
        /// </summary>  
        private ApplicationContext context;  
  
        /// <summary>  
        /// Конструктор  
        /// </summary>  
        /// <param name="context">Контекст БД</param>  
        public OrgCabinetController(ApplicationContext context)  
        {  
            this.context = context;  
        }  
  
        /// <summary>  
        /// Виклик OrgCabinet/Index  
        /// </summary>  
        /// <returns> Відповідь сервера</returns>  
        public IActionResult Index()  
        {  
            return View();  
        }  
  
        /// <summary>  
        /// Виклик OrgCabinet/Aside  
        /// </summary>  
        /// <returns> Відповідь сервера</returns>  
        public IActionResult Aside()  
        {  
            return PartialView();  
        }  
  
        /// <summary>  
        /// Виклик OrgCabinet/Cabinet  
        /// </summary>  
        /// <param name="login"> Логін того, хто викликає</param>  
        /// <returns> Відповідь сервера</returns>  
        public IActionResult Cabinet(string login)  
        {  
            var user = context.GetSellerByLogin(login);  
            if (user == null)  
                return View(new UserViewModel());  
  
            var userModel = new UserViewModel(user);  
        }  
    }  
}
```

```

        return View(userModel);
    }

    /// <summary>
    /// Виклик OrgCabinet/ProductsList
    /// </summary>
    /// <param name="login"> Логін того, хто викликає</param>
    /// <returns> Відповідь сервера</returns>
    public IActionResult ProductsList(string login)
    {
        context.GetEducationsByLogin(login, out List<Education>?
educations);

        var items = new ListViewModel();
        items.Login = login;
        foreach (var education in educations)
        {
            items.Items.Add(new()
            {
                ItemName = education.EducationName,
                Company = education.Seller.UserName,
                Status = education.IsAccepted ? " Підтверджено" :
(education.IsDeclined ? " Відхилено" : " У листі очікування"),
                StartDate = education.StartDate.Replace("T", ""),
                SoldCount = education.Users.Count,
                Price = education.Price,
                EducationType = education.EducationType
            });
        }

        context.GetJobsByLogin(login, out List<Job>? jobs);
        foreach (var job in jobs)
        {
            items.Items.Add(new()
            {
                ItemName = job.JobName,
                Company = job.Owner.UserName,
                Status = job.IsAccepted ? " Підтверджено" : (job.IsDeclined
? "Відхилено" : " У листі очікування"),
                EducationType = EducationType.Job
            });
        }

        items.Income = educations.Count > 0 ?
educations.First().Seller.Income : 0;

        return View(items);
    }

    /// <summary>
    /// Виклик OrgCabinet/UpdateData
    /// </summary>
    /// <returns> Відповідь сервера</returns>
    public async Task<IActionResult> UpdateData()
    {
        string json = await new
StreamReader(Request.Body).ReadToEndAsync();
        var record = JsonSerializer.Deserialize<RegRecord>(json);
        if (record == null)
            return NotFound("Wrong data");

        User user = new()

```

```

    {
        UserName = record.name,
        Login = record.login,
        Organization = record.org,
        Group = record.group,
        Phone = record.phone,
        Position = record.position
    };

    if (record.passwrд != null
        && record.passwrдRepeat != null
        && record.passwrд.Length != 0
        && record.passwrдRepeat.Length != 0
        && record.passwrд.Equals(record.passwrдRepeat))
    {
        user.Password = record.passwrд;
    }

    if (!context.UpdateEntity(user))
        return NotFound("Wrong data");

    return Ok();
}

/// <summary>
/// Виклик OrgCabinet/Remove
/// </summary>
/// <returns> Відповідь сервера</returns>
public async Task<IActionResult> Remove()
{
    string name = await new
StreamReader(Request.Body).ReadToEndAsync();

    var product = context.GetEducationByName(name);
    if (product == null)
        return BadRequest("Wrong data");

    if (!context.RemoveEntity(product))
        return BadRequest("Wrong data");

    return Ok();
}

/// <summary>
/// Виклик OrgCabinet/Notifications
/// </summary>
/// <param name="login"> Логін того, хто викликає</param>
/// <returns> Відповідь сервера</returns>
[HttpGet]
public IActionResult Notifications(string login)
{
    var notifications = context.GetNotificationsForSeller(login);
    var viewModel = new NotificationsViewModel();
    foreach (var notification in notifications)
        viewModel.Notifications.Add(new
NotificationViewModel(notification));

    return View(viewModel);
}

/// <summary>
/// Виклик OrgCabinet/RemoveNotifications

```



```

    /// </summary>
    /// <param name="login"> Логін того, хто викликає</param>
    /// <returns> Відповідь сервера</returns>
    [HttpDelete]
    public IActionResult RemoveNotifications(string login)
    {
        if (!context.RemoveNotifications(login))
            return BadRequest("Wrong data");

        return Ok();
    }
}
}

```

Лістинг файлу PayController.cs:

```

using MarketPlace.Extra;
using Microsoft.AspNetCore.Mvc;
using System.Text.Json;

namespace MarketPlace.Controllers
{
    /// <summary>
    /// Контролер вікна оплати
    /// </summary>
    public class PayController : Controller
    {
        /// <summary>
        /// Контекст БД
        /// </summary>
        private readonly ApplicationContext context;

        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="context">Контекст БД</param>
        public PayController(ApplicationContext context)
        {
            this.context = context;
        }

        /// <summary>
        /// Виклик Pay/Index
        /// </summary>
        /// <returns> Відповідь сервера</returns>
        public IActionResult Index()
        {
            return View();
        }

        /// <summary>
        /// Виклик Pay/Pay
        /// </summary>
        /// <returns> Відповідь сервера</returns>
        public async Task<IActionResult> Pay()
        {
            string json = await new
            StreamReader(Request.Body).ReadToEndAsync();
            var records = JsonSerializer.Deserialize<List<PayRecord>>(json);
            if (records == null)
                return NotFound("Wrong data");
        }
    }
}

```

```

        foreach (var record in records)
        {
            var user = context.GetUserByLogin(record.login);
            if (user == null)
                return BadRequest("Wrong data");

            var education = context.GetEducationByName(record.name);
            if (education == null)
            {
                var job = context.GetJobByName(record.name);
                if (job == null)
                    return BadRequest("Wrong data");

                return Ok();
            }

            education.Users.Add(user);
            education.Seller.Income += education.Price;
            if (!context.UpdateEntity(education))
                return BadRequest("Wrong data");

            return Ok();
        }

        return BadRequest("Wrong data");
    }
}

```

Лістинг файлу ProductController.cs:

```

using Marketplace.Extra;
using Marketplace.Models;
using Marketplace.ViewModels;
using Microsoft.AspNetCore.Mvc;
using System.Text.Json;

namespace Marketplace.Controllers
{
    /// <summary>
    /// Контролер вікна показу інформації про продукт: навчання, дозвілля,
інсайт, найм
    /// </summary>
    public class ProductController : Controller
    {
        /// <summary>
        /// Контекст БД
        /// </summary>
        private readonly ApplicationContext context;

        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="context">Контекст БД</param>
        public ProductController(ApplicationContext context)
        {
            this.context = context;
        }

        /// <summary>
        /// Виклик Product/Index
        /// </summary>

```

```

        /// <param name="name"> Назва продукту(навчання, дозвілля або
інсайту)</param>
        /// <param name="login"> Логін того, хто викликав</param>
        /// <param name="isReviews"> Показувати відгуки</param>
        /// <returns> Відповідь сервера</returns>
        public IActionResult Index(string name, string? login, bool? isReviews)
        {
            var education = context.GetEducationByName(name);
            if (education != null)
            {
                var viewModel = new EducationViewModel(education);
                var reviews = context.GetReviewsOfEducationByEducation(education);
                foreach (var review in reviews)
                    viewModel.Reviews.Add(new(review));

                if (reviews.Any())
                    viewModel.Mark = (double)reviews.Sum(review =>
review.Mark) / reviews.Count;

                if (login != null)
                    viewModel.IsBought = context.IsBoughtByUser(education,
login);

                viewModel.Login = login;

                if (isReviews != null)
                    ViewBag.IsReviews = true;

                return View("Education", viewModel);
            }

            var job = context.GetJobByName(name);
            if (job != null)
            {
                var viewModel = new JobViewModel(job);
                var reviews = context.GetReviewsOfJobByJob(job);
                foreach (var review in reviews)
                    viewModel.Reviews.Add(new(review));

                if (reviews.Any())
                    viewModel.Mark = (double)reviews.Sum(review =>
review.Mark) / reviews.Count;
                viewModel.Login = login;

                return View("Job", viewModel);
            }

            return NotFound();
        }

        /// <summary>
        /// Виклик Product/AddOpinion
        /// </summary>
        /// <returns> Відповідь сервера</returns>
        public async Task<ActionResult> AddOpinion()
        {
            string json = await new
StreamReader(Request.Body).ReadToEndAsync();
            var record = JsonSerializer.Deserialize<AddOpinionRecord>(json);
            if (record == null)
                return BadRequest("Wrong data");
        }
    }
}

```

```

var user = context.GetUserByLogin(record.login);
if (user == null)
    return BadRequest("Wrong data");

Review review = new()
{
    Text = record.opinion,
    Mark = record.mark,
    User = user
};

var education = context.GetEducationByName(record.name);
if (education == null)
{
    var job = context.GetJobByName(record.name);
    review.Job = job;
    if (job == null)
        return BadRequest("Wrong data");

    return Ok();
}

review.Education = education;
if (!context.AddNewEntity(review))
    return BadRequest("Wrong data");
return Ok();
}

/// <summary>
/// Виклик Product/Reply
/// </summary>
/// <param name="name"> Назва продукту(найму)</param>
/// <param name="login"> Логін того, хто викликав</param>
/// <returns></returns>
public IActionResult Reply(string name, string login)
{
    var job = context.GetJobByName(name);
    if (job == null)
        return BadRequest("Wrong name of job");

    var user = context.GetUserByLogin(login);
    if (user == null)
        return BadRequest("Wrong login");

    var notification = new Notification();
    notification.UserName = user.UserName;
    notification.Login = login;
    notification.JobName = job.JobName;
    notification.ReceiverLogin = job.Owner.Login;
    if (!context.AddNewEntity(notification))
        return BadRequest("Wrong Data");

    return RedirectToAction("Index", "Product", new Dictionary<string,
string>()
    {
        ["name"] = name,
        ["login"] = login
    });
}
}
}
}

```

Лістинг файлу RegController.cs:

```
using MarketPlace.Extra;
using MarketPlace.Models;
using Microsoft.AspNetCore.Mvc;
using System.Text.Json;

namespace MarketPlace.Controllers
{
    /// <summary>
    /// Контролер реєстрації
    /// </summary>
    public class RegController : Controller
    {
        /// <summary>
        /// Контекст БД
        /// </summary>
        private ApplicationDbContext context;

        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="context">Контекст БД</param>
        public RegController(ApplicationDbContext context)
        {
            this.context = context;
        }

        /// <summary>
        /// Виклик Reg/Index
        /// </summary>
        /// <returns> Відповідь сервера</returns>
        public IActionResult Index()
        {
            return View();
        }

        /// <summary>
        /// Виклик Reg/Registrate
        /// </summary>
        /// <returns> Відповідь сервера</returns>
        [HttpPost]
        public async Task<IActionResult> Registrate()
        {
            // Отримуємо тіло запиту
            Stream stream = Request.Body;
            string json = await new StreamReader(stream).ReadToEndAsync(); //
            Считываем тело
            var regRecord = JsonSerializer.Deserialize<RegRecord>(json);
            if (regRecord == null)
            {
                return BadRequest("Couldn't manage the data. Please, check
            it");
            }

            if (regRecord.accountType == UserType.Student)
            {
                User user = new()
                {
                    UserName = regRecord.name,
                    Login = regRecord.login,
                    Password = regRecord.passwrд,
                }
            }
        }
    }
}
```

```

        Organization = regRecord.org,
        Group = regRecord.group,
        Phone = regRecord.phone,
        Position = regRecord.position
    };

    if (context.AddNewEntity(user))
        return Ok();
    else
        return BadRequest("The process failed due to the incorrect
input data");
    }

    Seller seller = new()
    {
        UserName = regRecord.name,
        Login = regRecord.login,
        Password = regRecord.passwrд,
        Organization = regRecord.org,
        Group = regRecord.group,
        Phone = regRecord.phone,
        Position = regRecord.position
    };

    if (context.AddNewUser(seller))
        return Ok();
    else
        return BadRequest("The process failed due to the incorrect
input data");
    }
}
}
}

```

Лістинг файлу UserCabinetController.cs:

```

using Marketplace.Extra;
using Marketplace.Models;
using Marketplace.ViewModels;
using Microsoft.AspNetCore.Mvc;
using System.Text.Json;

namespace Marketplace.Controllers
{
    /// <summary>
    /// Контролер кабінету користувача
    /// </summary>
    public class UserCabinetController : Controller
    {
        /// <summary>
        /// Контекст БД
        /// </summary>
        private readonly ApplicationContext context;

        /// <summary>
        /// Конструктор
        /// </summary>
        /// <param name="context">Контекст БД</param>
        public UserCabinetController(ApplicationContext context)
        {
            this.context = context;
        }
    }
}

```

```

/// <summary>
/// Виклик UserCabinet/Index
/// </summary>
/// <returns> Відповідь сервера</returns>
public IActionResult Index()
{
    return View();
}

/// <summary>
/// Виклик UserCabinet/Aside
/// </summary>
/// <returns> Відповідь сервера</returns>
public IActionResult Aside()
{
    return PartialView();
}

/// <summary>
/// Виклик UserCabinet/Cabinet
/// </summary>
/// <param name="login"> Логін того, хто викликає</param>
/// <returns> Відповідь сервера</returns>
[HttpGet]
public IActionResult Cabinet(string login)
{
    var user = context.GetUserByLogin(login);
    if (user == null)
        return View(new UserModel());

    var userModel = new UserModel(user);

    return View(userModel);
}

/// <summary>
/// Виклик UserCabinet/ProductsList
/// </summary>
/// <param name="login"> Логін того, хто викликає</param>
/// <returns> Відповідь сервера</returns>
[HttpGet]
public IActionResult ProductsList(string login)
{
    var user = context.GetUserByLogin(login);
    if (user == null)
        return View(new ListViewModel());

    var viewModel = new ListViewModel();
    foreach (var item in user.Educations)
    {
        viewModel.Items.Add(new ItemViewModel()
        {
            ItemName = item.EducationName,
            Company = item.Seller.UserName,
            StartDate = item.StartDate.Replace("T", " "),
            ImageURL = item.Link
        });
    }
    viewModel.Login = login;
    return View(viewModel);
}

```

```

/// <summary>
/// Виклик UserCabinet/UpdateData
/// </summary>
/// <returns> Відповідь сервера</returns>
[HttpPost]
public async Task<IActionResult> UpdateData()
{
    string json = await new
StreamReader(Request.Body).ReadToEndAsync();
    var record = JsonSerializer.Deserialize<RegRecord>(json);
    if (record == null)
        return NotFound("Wrong data");

    User user = new()
    {
        UserName = record.name,
        Login = record.login,
        Organization = record.org,
        Group = record.group,
        Phone = record.phone,
        Position = record.position
    };

    if (record.passwrд != null
        && record.passwrдRepeat != null
        && record.passwrд.Length != 0
        && record.passwrдRepeat.Length != 0
        && record.passwrд.Equals(record.passwrдRepeat))
    {
        user.Password = record.passwrд;
    }

    if (!context.UpdateEntity(user))
        return NotFound("Wrong data");

    return Ok();
}
}
}

```