

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри

\_\_\_\_\_ Скарга-Бандурова І.С.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА**

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

НА ТЕМУ:

**Система он-лайн замовлення їжі**

---

---

---

Освітньо-кваліфікаційний рівень “бакалавр”  
Напрямок підготовки 6.050102 – “Комп’ютерна інженерія”

Керівник проекту:

\_\_\_\_\_

(підпис)

Щербакова М.Є.

\_\_\_\_\_

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_

(підпис)

Критська Я.О.

\_\_\_\_\_

(ініціали, прізвище)

Здобувач вищої освіти:

\_\_\_\_\_

(підпис)

Михайлова А.О.

\_\_\_\_\_

(ініціали, прізвище)

Група:

КІ-14ад

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітньо-кваліфікаційний рівень бакалавр  
Напрямок підготовки 6.050102 – “Комп'ютерна інженерія”  
(шифр і назва)  
Спеціальність \_\_\_\_\_  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_  
І.С. Скарга-Бандурова  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

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

Михайлової Аліси Олександрівни  
(прізвище, ім'я, по батькові)

1. Тема роботи Система он-лайн замовлення їжі

керівник проекту  
(роботи) Щербакова М.Є., к.т.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 4 05 2018 р. № 117/48

2. Термін подання студентом роботи 11.06.2018

3. Вихідні дані до  
роботи Матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області та постановка задачі; вибір засобів для розробки; розробка системи он-лайн замовлення їжі; охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Блок-схеми алгоритмів, електронні плакати

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	ст.викл. Критська Я.О.		

## 7. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Ознайомлення з предметною галуззю	03.05.18 – 6.05.18	
2	Аналіз існуючих аналогів	7.05.18 – 8.05.18	
3	Вибір засобів для розробки	9.05.18 – 17.05.18	
4	Розробка системи	18.05.18 – 03.06.18	
5	Розробка розділу «Охорона праці та безпека в надзвичайних ситуаціях»	04.06.18 – 07.06.18	
6	Оформлення пояснювальної записки	08.06.18 – 11.06.18	

**Здобувач вищої освіти** \_\_\_\_\_

( підпис )

**Михайлова А.О.** \_\_\_\_\_

(прізвище та ініціали)

**Керівник** \_\_\_\_\_

( підпис )

**Щербакова М.Є.** \_\_\_\_\_

(прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра: 88 с., 38 рис., 4 табл., 25 бібліографічних джерел посилань, 2 додатки.

Об'єкт розробки: система он-лайн замовлення їжі.

Мета роботи: аналіз ринку foodtech та визначення методики створення сервісу замовлення їжі, а також подальша розробка сервісу.

В проекті виконано:

1. Проведено аналіз стану ринку foodtech та функціоналу існуючих сервісів
2. Проведено аналіз аспектів та методів розробки агрегатора доставки їжі;
3. Спроектовано та розроблено систему он-лайн замовлення їжі.
4. Зроблено аналіз умов праці та приведено рекомендації щодо організації робочого місця.

Отримані наступні результати: придатна до використання система он-лайн замовлення їжі.

Практичне значення, галузь застосування роботи: система може бути використана у галузі продажу та реклами.

**Ключові слова: СИСТЕМА ОН-ЛАЙН ЗАМОВЛЕННЯ, ІНТЕРНЕТ, FOODTECH, WEB-ТЕХНОЛОГІЇ, FRAMEWORK**

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєвєродонецьк, 93400.

## ЗМІСТ

<b>ВСТУП</b> .....	<b>7</b>
<b>1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ</b> .....	<b>8</b>
1.1 Галузь foodtech .....	8
1.1.1 Походження та сфера діяльності .....	8
1.1.2 Агрегатори служб доставки їжі .....	9
1.1.3 Стислий аналіз існуючих сервісів .....	10
1.2 Обґрунтування необхідності розробки сервісу .....	11
1.3 Технічне завдання на розробку .....	12
1.3.1 Постановка завдання на розробку програмного забезпечення .....	12
Висновки до першого розділу .....	13
<b>2 АНАЛІЗ ЗАСОБІВ РОЗРОБКИ СЕРВІСУ</b> .....	<b>14</b>
2.1 Мова розмітки HTML .....	14
2.2 Мова розмітки CSS .....	15
2.3 Мова сценаріїв JavaScript .....	15
2.3.1 Бібліотека jQuery .....	16
2.4 Скриптова мова PHP .....	17
2.5 Сервер баз даних MySQL .....	18
2.6 Програмна платформа Framework .....	19
2.6.1 Фреймворк CodeIgniter .....	20
2.6.2 Фреймворк Bootstrap .....	22
2.7 Інтегровані середовища розробки .....	24
2.7.1 Інтегроване середовище розробки Cloud9 .....	25
Висновки до другого розділу .....	27
<b>3 ПРОЕКТУВАННЯ ТА РОЗРОБКА СЕРВІСУ</b> .....	<b>28</b>
3.1 Проектування структури сервісу .....	28
3.2 Розробка структури сервісу на основі фреймворку CodeIgniter .....	33
3.3 Розробка дизайну .....	34
3.4 Сторінка реєстрації .....	37

3.5 Авторизація та особистий кабінет .....	38
3.6 Категорії і доставки .....	42
3.7 Кошик .....	44
3.8 Оформлення замовлення .....	45
3.9 Оцінки та коментарі.....	47
Висновки до третього розділу .....	48
<b>4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....</b>	<b>50</b>
4.1 Аналіз стану умов праці .....	50
4.1.1 Вимоги до приміщення .....	50
4.1.2 Вимоги до організації місця праці .....	51
4.2 Виробнича санітарія .....	51
4.2.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу.....	52
4.2.2 Пожежна безпека .....	53
4.2.3 Електробезпека .....	55
4.3 Гігієнічні вимоги до параметрів виробничого середовища.....	55
4.3.1 Мікроклімат .....	55
4.3.2 Освітлення .....	56
4.4 Вентилювання.....	58
4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій.....	58
4.5.1 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі .....	59
Висновки до четвертого розділу .....	62
<b>ВИСНОВКИ.....</b>	<b>63</b>
<b>ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>65</b>
<b>ДОДАТОК А.....</b>	<b>67</b>
<b>ДОДАТОК Б .....</b>	<b>82</b>

## ВСТУП

Головний ресурс, який затребуваний у сучасному світі – час. Він цінний для людини будь-якого роду діяльності. У зв'язку з цим створюються сервіси, що дозволяють людям економити цей ресурс.

Подібні сервіси нерозривно пов'язані з Інтернетом. Розвиток Інтернету набуває все більших масштабів, що призводить до появи інновацій у різних сферах людського життя, які покликані зробити життя якомога більш комфортним.

Одна із сфер, в якій з'являються подібні інновації – споживання їжі. Ринок продовольства разом з ринком доставки їжі освоюють нові технології, до яких входить і інтернет. Таким чином з'явився ринок foodtech.

Однією з галузей даного ринку є агрегатори доставки їжі – сервіси, що поєднують у собі різноманітні доставки і онлайн-ресторани. Загальний обсяг світового ринку foodtech-сервісів, зокрема – сервісів замовлення їжі онлайн, становить \$ 324 млрд щорічно.

Мета дипломного проекту: аналіз ринку foodtech та визначення методики створення системи замовлення їжі, а також подальша її розробка.

Об'єкт дослідження: система он-лайн замовлення їжі.

Предмет дослідження: методика створення та подальшої підтримки системи он-лайн замовлення їжі.

Відповідно до мети дипломного проекту було сформульовано такі завдання:

- проаналізувати стан ринку foodtech;
- проаналізувати функціонал існуючих сервісів;
- проаналізувати аспекти та методи розробки агрегатора доставки їжі;
- розробити систему он-лайн замовлення їжі.

Інформаційна база проектування ґрунтується на ресурсах мережі Інтернет, а також на відповідних методичних та навчальних матеріалах.

# 1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Галузь foodtech

### 1.1.1 Походження та сфера діяльності

Foodtech є цілком новим видом бізнесу, що походить від колаборації між технологіями та харчовою промисловістю. Офіційного визначення цього явища ще не існує через його новизну. Основною метою є пришвидшення процесу замовлення та отримання продуктів харчування на найбільш комфортних для клієнта умовах[12].

На даний час існує п'ять основних напрямків розвитку foodtech:

а) Агрегатори служб доставки їжі

Компанії даної категорії працюють з ресторанами, які мають свої власні служби доставки. Агрегуючи пропозиції з різних ресторанів, сервіси надають користувачеві широкий вибір і можливості зручного пошуку та порівняння різних пропозицій. Щоб потрапити на платформу і почати отримувати замовлення, ресторани, як правило, платять фіксований відсоток з кожного підтверженого замовлення.

б) Сервіси доставки їжі з ресторанів

Компанії цієї категорії доставляють їжу з ресторанів менш ніж за 45 хвилин. Дозволяють не тільки швидко доставити замовлення, але й представляють доступ до ресторанів, з яких не можна отримати замовлення іншим чином. Це найпоширеніший спосіб замовлення обіду серед офісних працівників.

в) Онлайн-сервіси для бронювання місць у ресторанах

На сайті або в мобільному додатку можна обрати ресторан за набором критеріїв (кухня, місце розташування, ціна та інші) і забронювати місце на заданий час і кількість гостей. Сервіс, як правило, інтегрований з системою управління рестораном і автоматично передає інформацію про замовлення.



Компанія заробляє фіксовану суму з кожного бронювання або відсоток з його рахунку в ресторані.

г) Доставка наборів продуктів за передплатою

Щотижня споживачі отримують рецепти страв і підібраний набір інгредієнтів, щоб їх приготувати. Сервіс дозволяє приготувати вдома їжу ресторанної якості і заощадити час клієнта.

д) Доставка продуктів із супермаркетів

Компанії агрегують весь асортимент магазинів на своєму сайті. Замовлення покупця доставляється протягом 1-3 годин. Ціна зазвичай трохи вища за середньоринкову[10].

### **1.1.2 Агрегатори служб доставки їжі**

Агрегатор служб доставки їжі – це сервіс, який є посередником між клієнтами і закладами громадського харчування. З покупців на сервісі не береться комісія, також клієнти можуть отримати знижки та інші вигідні пропозиції. Купуючи їжу, використовуючи такий сервіс, можна[11]:

- швидко оформити замовлення;
- знайти потрібну страву;
- знайти ресторан з необхідними умовами;
- знайти страву з найвигіднішою ціною.

На сьогоднішній день агрегатори працюють у десятках міст, тому, ставши користувачем одного з сервісів, клієнт може робити замовлення навіть знаходячись у відрядженнях та на відпочинку.

Також, агрегатори – це майданчики, на яких можна побачити рейтинг закладу та залишити коментар. Складають такі рейтинги власне користувачі, отже, оцінка є об'єктивною і чесною.

Фінансування агрегаторів служб доставки виконується за рахунок ресторанів та доставок, розміщених на сервісі[11].

### 1.1.3 Стислий аналіз існуючих сервісів

Основним ареалом використання сервісів онлайн-замовлення їжі є країни Європи та США, тому більша частина існуючих на сьогоднішній день сервісів не охоплюють країни СНГ та, зокрема, Україну. Далі наведений стислий аналіз найвідоміших служб доставки.

а) Eda.ua

Країна: Україна

Перевагами сервісу є: єдина бонусна система, щоденні акції та знижки, оплата кредитною картою, обмін балів на призи, рейтинг доставок.

Недоліками сервісу є: обмежена кількість міст, що знаходяться в зоні доставки, необхідність реєстрації.

Платформа: веб-сайт та мобільний додаток.

б) ZakaZaka

Країна: Росія

Перевагами сервісу є: накопичувальна система балів для безкоштовних замовлень, можливість оплати картою, рейтинг доставок.

Недоліками сервісу є: складна структура сайту, недоступність для українських користувачів.

Платформа: веб-сайт та мобільний додаток.

в) Delivery Club

Країна: Росія

Перевагами сервісу є: додаткові можливості і бали для зареєстрованих користувачів, накопичувальна система балів для безкоштовних замовлень і блог, де публікуються замітки про нові можливості та пропозиції, і інша корисна і цікава інформація, можливість оплати картою.

Недоліками сервісу є: складна структура сайту недоступність для українських користувачів.

Платформа: веб-сайт та мобільний додаток.

г) Just Eat

Країна: Великобританія, але охоплює 13 країн Європи, Азії, Америки та Океанії.

Перевагами сервісу є: блог, де публікуються замітки про нові можливості та пропозиції, та інша корисна інформація, можливість оплати картою.

Недоліками сервісу є: сторінки сайту досить довго підвантажуються, складна структура сайту, недоступність для українських користувачів.

Платформа: веб-сайт та мобільний додаток.

д) HungryHouse

Країна: Великобританія

Перевагами сервісу є: можливість оплати картою, швидкодія сайту.

Недоліками сервісу є: відсутність зображень страв у меню, недоступність для українських користувачів.

Платформа: веб-сайт.

е) Foodpanda

Країна: Німеччина, але охоплює 43 країни Європи, Азії, Океанії і Америки.

Перевагами сервісу є: можливість оплати картою, можливість замовлення вегетаріанської їжі.

Недоліками сервісу є: відсутність зображень страв у меню, сторінки сайту досить довго підвантажуються, недоступність для українських користувачів.

Платформа: веб-сайт.

## 1.2 Обґрунтування необхідності розробки сервісу

У зв'язку з пришвидшенням ритму людського життя, поява сервісів, які допомагають економити час стає помітною тенденцією. Необхідність подібних сервісів особливо відчують на собі жителі великих міст, а жителі країн Європи все частіше відмовляються від домашнього приготування страв.

Ще одним важливим моментом є орієнтованість не тільки на клієнтів, а й на власників доставок та ресторанів. Таким аспектом є реклама.

Реклама в сучасному світі відіграє провідну роль. Без засобів просування товару мало хто знав би навіть про найпопулярніші в наш час бренди. Як і інші сектори індустрії, рекламна діяльність робить свій внесок у розвиток економіки, забезпечує ефективне просування нових товарів і супутніх в їх експлуатації послуг.

Завдяки цьому, агрегатори доставки їжі необхідні також і власникам доставок. Система може працювати, як стартовий майданчик для нових доставок і додатковий засіб просування для існуючих.

Виходячи з цього, розробка надає можливості одночасно користувачам та власникам доставок.

Для користувачів:

- можливість зручного пошуку доставок;
- можливість економити час за рахунок багатьох факторів;
- можливість залишати та переглядати відгуки;
- рейтингова система.

Для власників:

- додаткова реклама;
- можливість обійтись без власного сайту.

### **1.3 Технічне завдання на розробку**

#### **1.3.1 Постановка завдання на розробку програмного забезпечення**

Необхідно розробити сервіс, який матиме наступні функції:

- можливість оформлення замовлення їжі з будь-якої з представлених доставок;
- реєстрація;
- пошук доставки за категоріями (такими як: кухня, місцеположення, час роботи, час поставки, популярність та багато інших);
- можливість залишати оцінки та коментарі;
- можливість зв'язатися зі службою підтримки.

Основна ідея готового сервісу полягає у максимальній зручності його використання для користувача будь-якого рівня. Важливими є інтерфейс зрозумілий на інтуїтивному рівні та максимально логічна побудова. Даний сервіс покликаний замінити собою безліч розрізаних сайтів і тим самим скоротити час, витрачений на пошук і замовлення.

Виходячи з цього, можна сказати, що головними завданнями проектування є:

- зручність;
- орієнтація на широке коло користувачів;
- лаконічність в поєднанні з інформативністю;
- можливість швидкого пошуку доставок за категоріями;
- легка система оформлення замовлень;
- можливість залишати і переглядати відгуки на доставки.

### **Висновки до першого розділу**

Мережа Інтернет проникає у всі сфери людського життя. Кількість сервісів, які допомагають користувачам економити час, зростає з кожним роком. Також вони нерозривно пов'язані з рекламою, яка є важливим засобом просування.

Ринок foodtech розвивається завдяки новим проектам та отримує значне фінансування. Агрегатори доставки їжі з'являються у багатьох країнах світу і є перспективною галуззю ринку.

На території України ринок не майже не має представників, що підтверджує актуальність розробки.

## 2 АНАЛІЗ ЗАСОБІВ РОЗРОБКИ СЕРВІСУ

### 2.1 Мова розмітки HTML

HTML (Hypertext Markup Language. З англ. - «мова гіпертекстової розмітки») – додаток SGML (Standard Generalized Markup Language [Стандартна узагальнена мова розмітки]), відповідає міжнародному стандарту ISO 8879; вважається стандартною мовою публікації в World Wide Web.

HTML була створена, як мова для обміну документацією (технічною та науковою), вона може бути застосована людьми, які не є фахівцями в області веб-розробок. HTML від самого початку справлялась з проблемою складності SGML, завдяки визначенню обмеженого набору семантичних та структурних елементів, що використовуються для створення простих, але правильно оформлених документів. Спрощення структури документа не є єдиним нововведенням, в HTML, крім того, є підтримка гіпертексту. Мультимедійні та інші можливості було додано пізніше. На початковому етапі HTML була задумана, як засіб форматування та структурування документів, не прив'язуючи їх до засобів відтворення. Текст з розміткою HTML повинен був без структурних та стилістичних спотворень відтворюватися на обладнанні з різною технологічною оснащеністю [6].

Текстові документи, які містять у собі код на мові HTML (такі документи мають розширення «html» та «htm»), обробляються спеціалізованими додатками, що відображають документ в його кінцевому вигляді. Необхідна наявність у телефоні HTML-браузера. З його використанням веб-браузер дає змогу переглядати html-сторінки в мережі Інтернет. Телефон з підтримкою WAP дає змогу переглядати тільки попередньо оптимізовані для мобільного телефону сторінки формату WML або XHTML[6].

## 2.2 Мова розмітки CSS

CSS (англ. Cascading Style Sheets - каскадні таблиці стилів) – спеціалізована технологія для опису зовнішнього вигляду документа, що оформлений мовою розмітки.

Зазвичай використовується як засіб оформлення веб-сторінок у форматі HTML і XHTML, тим не менш, може застосовуватися з іншими видами документів у форматі XML, включаючи SVG і XUL[6].

Каскадні таблиці стилів використовуються для верстки веб-сторінок, щоб задати кольори, шрифти та розташування будь-яких інших аспектів представлення веб-документа. Початковою метою розробки мови CSS було розділення вмісту, що написаний на HTML (або іншій мові розмітки) і оформлення документа, що написане на CSS. Цей розподіл допомагає збільшити доступність документа, надати більшу гнучкість та можливість управління його поданням, зменшити складність та повторюваність в структурному вмісті. CSS дозволяє використовувати один і той самий документ у різних стилях та методах виведення, таких як екранне уявлення, друк, читання голосом (спеціалізованим голосовим браузером), або при виведенні пристроями, які використовують шрифт Брайля.

## 2.3 Мова сценаріїв JavaScript

JavaScript (JS) – об'єктно-орієнтована, динамічна мова програмування. Є реалізацією стандарту ECMAScript. Зазвичай використовується для створення сценаріїв веб-сторінок. Це надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, асинхронно обмінюватися даними з сервером, керувати браузером, змінювати структурний та зовнішній вигляд веб-сторінок.

JavaScript класифікується як прототипна (підвид об'єктно-орієнтованої), скриптова мова програмування з динамічною типізацією. JavaScript, крім того, підтримує імперативну та частково функціональну парадигми програмування.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-додатків (ReactJS, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних додатків (Electron, NW.js);
- мобільних додатків (React Native, Cordova);
- всередині PDF-документів, тощо.

Не дивлячись на деяку схожість назв, мови Java та JavaScript є різними мовами, які мають різну семантику, хоча у них і наявні схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов успадкований від мови C, але дизайн та семантика JavaScript є результатом впливу мов Self та Scheme[5].

### 2.3.1 Бібліотека jQuery

jQuery ([dʒeɪ kwɪəri]) – бібліотека JavaScript, що базується на взаємодії між JavaScript та HTML. Бібліотека jQuery надає можливість без проблем отримувати доступ до різних елементів DOM, а також звертатися до вмісту та атрибутів елементів DOM, керуючи ними. Що важливо, бібліотека jQuery надає комфортний для користувача API для роботи з AJAX.

Основні можливості[1]:

- движок крос-браузерних CSS-селекторів Sizzle, що виділився в окремий проект;
- перехід по дереву DOM, включаючи підтримку XPath, як плагіна;
- події;
- візуальні ефекти;
- AJAX-додатки;
- JavaScript-плагіни.



## 2.4 Скриптова мова PHP

PHP (англ. PHP: Hypertext Preprocessor – PHP: гіпертекстовий препроцесор), початкова назва: Personal Home Page Tools – мова програмування, що була розроблена для створення та запуску HTML-сторінок зі сторони веб-сервера. PHP називають однією з найпоширеніших мов, що використовуються для веб-розробок. PHP підтримується переважно більшістю хостинг-провайдерів. PHP - проект відкритого програмного забезпечення[1].

PHP – це мова програмування, яка була спеціально спроектована та розроблена для написання веб-сценаріїв, що виконуються на Web-сервері.

Абревіатура PHP розшифровується як "Hypertext Preprocessor (Препроцесор Гіпертексту)". Синтаксис цієї мови бере свій початок від C, Java та Perl. Мова проста у вивченні та використанні. Сильною стороною PHP є надання веб-розробникам можливість швидкого створення веб-сторінок, які генеруються динамічно.

Важлива перевага мови PHP перед такими мовами, як Perl і C, полягає у можливості створення HTML документів у які впроваджені команди PHP.

Однією з головних відмінностей PHP від будь-якого іншого коду, що виконується зі сторони клієнта (JavaScript та інші) є те, що PHP-скрипти виконуються зі сторони сервера. Є можливість налаштування серверу таким чином, що HTML-файли обробляються процесором PHP, так, що клієнт не бачить наочно, чи отримує він звичайний HTML-файл чи результат виконання PHP-скрипта.

Використовуючи PHP, можна створювати якісні веб-додатки за дуже короткі терміни, отримуючи продукти, які легко модифікуються і підтримуються протягом часу.

PHP доволі простий для освоєння, але в той же час здатний задовольнити запити навіть професійних програмістів.

Мова PHP постійно вдосконалюється, і їй, напевно, забезпечено у наступні десятиліття довге панування серед мов web-програмування[3].

## 2.5 Сервер баз даних MySQL

MySQL – це популярний сервер баз даних, який використовується в різних додатках. SQL означає мову структурованих запитів - (S)tructured (Q)uery (L)anguage, який MySQL використовує для комунікації з іншими програмами. Понад те, MySQL має свої власні розширені функції SQL для того щоб забезпечити користувачам додатковий функціонал. Що важливо, розробники розробляють додатковий функціонал на замовлення користувачів. Дякуючи подібним замовленням, у ранніх версіях з'явився механізм реплікації[16].

Спільнотою розробників MySQL створені різні відгалуження коду, такі як Drizzle, OurDelta, Percona Server і MariaDB. Всі ці відгалуження вже існували на момент поглинання компанії Sun корпорацією Oracle.

Мова SQL (Structured Query Language - структурована мова запитів) являє собою стандартну високорівневу мову опису даних і маніпулювання ними в системах управління базами даних (СКБД), побудованих на основі реляційної моделі даних[16].

Мова SQL була розроблена фірмою IBM наприкінці 70-х років, а перший міжнародний стандарт мови був прийнятий у 1989 р, а новий (більш повний) – у 1992 р. У даний час всі виробники реляційних СУБД підтримують з різним ступенем відповідності стандарт SQL92.

SQL є мовою структурованих запитів для доступу до баз даних і їх управління.

SQL може:

- виконувати запити до бази даних;
- отримувати дані з бази даних;
- додати записи в базу даних;
- оновити записи в базі даних;
- записи з бази даних;
- створювати нові бази даних;
- створювати нові таблиці в базі даних;

- створювати збережені процедури в базі даних;
- налаштувати дозволу для таблиць.

Прийняті змінні:

- table\_name - ім'я таблиці;
- column\_name - колонки (поля);
- value – значення;
- pattern – шаблон;
- database\_name – ім'я бази даних у БД[17].

## 2.6 Програмна платформа Framework

Фреймворк (іноді фреймворк; англiцизм, неологiзм вiд framework – остов, каркас, структура) – програмна платформа, що визначає структуру програмних систем; програмне забезпечення, що робить легшою розробку рiзних компонентiв великих програмних проектiв.

Прийнято використовувати термiн «каркас». Можна також говорити про каркасний пiдхiд, як про особливий пiдхiд до побудови програм, де будь-яка конфiгурацiя програми будується з двох частин:

- Постiйна частина – каркас, не змiнний вiд конфiгурацiї до конфiгурацiї i несе в собi гнiзда, в яких розмiщується друга, змiнна частина;
- Змiннi модулi (або точки розширення).

«Фреймворк» вiдрiзняється вiд поняття бiблiотеки тим, що бiблiотека може бути використана в програмному продуктi просто як набiр пiдпрограм близької функцiональностi, не впливаючи на архiтектуру програмного продукту i не накладаючи на неї нiяких обмежень. У той час як «фреймворк» диктує правила побудови архiтектури додатку, задаючи на початковому етапi розробки «каркас», який потрiбно буде розширювати i змiнювати, вiдповiдно до зазначених вимог.

Також, на вiдмiну вiд бiблiотеки, яка об'єднує в собi набiр близькою функцiональностi, – «фреймворк» може мiстити в собi велику кiлькiсть рiзних за тематикою бiблiотек[14].

Іншою ключовою відмінністю «фреймворка» від бібліотеки може бути інверсія управління: призначений для користувача код викликає функції бібліотеки (або класи) і отримує управління після виклику. У «фреймворку» призначений для користувача код може реалізовувати конкретну поведінку, вбудовується у більш загальний – «абстрактний» код фреймворка. При цьому «фреймворк» викликає функції (класи) для користувача коду.

Одна з головних переваг при використанні «каркасних» додатків – «стандартність» структури програми. «Каркаси» стали популярні з появою графічних інтерфейсів користувача, які мали тенденцію до реалізації стандартної структури для додатків. З їх використанням стало набагато простіше створювати засоби для автоматичного створення графічних інтерфейсів, так як структура внутрішньої реалізації коду програми стала відома заздалегідь. Для забезпечення каркаса, зазвичай використовуються техніки об'єктно-орієнтованого програмування (наприклад, частини програми можуть успадковуватися від базових класів фреймворка).

Одним з перших комерційних фреймворків був MacApp, написаний Apple для «Macintosh». Спочатку створений за допомогою розширеної (об'єктно-орієнтованої) версії мови «Object Pascal», згодом він був переписаний на «C++».

### **2.6.1 Фреймворк CodeIgniter**

CodeIgniter – популярний MVC фреймворк з відкритим вихідним кодом, написаний на мові програмування PHP, для розробки повноцінних веб-систем і додатків. Розроблено компанією EllisLab (англ.) Рос., А також Ріком Еллісом (Rick Ellis) і Полом Бурдик (Paul Burdick)[14].

Стара версія CodeIgniter'a (CodeIgniter 2.x), як і більш ранні версії, поширюються під пропріетарної ліцензією в стилі Apache / BSD, проте поточна гілка CodeIgniter 3 переліцензувана під MIT.

Перший публічний реліз фреймворку стався 28 лютого 2006 року. З виходом версії 2.0 28-го січня 2011 року CodeIgniter розділювався на дві гілки:

- CodeIgniter Core – гілка, яка робить акцент на стабільності, в зв'язку з чим розвивається повільно. Ця версія фреймворку є ядром популярної CMS ExpressionEngine 2.0.
- CodeIgniter Reactor – гілка, яку розвиває співтовариство. Ця гілка розвивається швидко і містить найновіші можливості.

CodeIgniter відрізняє простота, яка досягається завдяки наступним факторам:

- Якісна і повна документація з прикладами, а також велика спільнота і Wiki.
- Безліч відео-уроків, які можна знайти як на офіційному сайті, так і на сторонніх ресурсах.
- Фреймворк дає свободу програмісту, не створюючи будь-яких структурних обмежень і конвенцій.
- Програмісту не потрібно вчитися користуватися генераторами коду з командного рядка.
- CodeIgniter працює практично на будь-якому хостинговому плані, який має підтримку PHP версії 5.1 і вище.
- CodeIgniter вважається одним з найшвидших і не вимогливих до ресурсів фреймворків.

Можливості:

- Підтримка баз даних MySQL, PostgreSQL, MSSQL, SQLite, Oracle.
- Підтримка псевдо-ActiveRecord, який здебільшого повторює синтаксис мови SQL.
- Легко розширювана система за рахунок можливості використання сторонніх і самописних бібліотек, а також доповнення або перевизначення існуючих.
- Підтримка як сегментованих ЧПУ, так і звичайних URL-ів з передачею параметрів.
- Фреймворк містить в собі безліч необхідних бібліотек, які створюють функціональність для роботи з файлами, надсилання електронних листів, валідації форм, підтримки сесій, роботи з зображеннями і так далі.

- Володіє можливістю кешування на стороні сервера SQL-запитів і генерує html-сторінки. З версії 2.0 для кешування можуть використовуватися XCache або APC.
- У 2011 році з'явився менеджер пакетів під назвою Sparks, який дозволяє легко встановити сторонні бібліотеки за допомогою командного рядка.
- У 2011 році з'явилося доповнення, яке робить можливим підтримку міграцій.
- Підтримка модульності (HMVC) за допомогою доповнень.
- За замовчуванням CodeIgniter не використовує шаблонізатор, але є можливість використовувати його власний або один з альтернативних шаблонізаторів (Smarty, TinyButStrong)[9].

### **2.6.2 Клієнтський фреймворк Bootstrap**

Bootstrap – це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері. Репозиторій з даним фреймворком є одним з найбільш популярних на GitHub. Серед інших, його використовують NASA і MSNBC.

Bootstrap (початкова назва — Twitter Blueprint) був розроблений Марком Отто та Джейкобом Торнтоном як фреймворк для забезпечення однаковості внутрішніх інструментів Twitter. До появи Bootstrap у розробці інтерфейсу застосовувалися різні бібліотеки, що призводило до появи суперечностей та ускладнювало супровід[2].

Через кілька місяців до розробки долучилося багато розробників компанії Twitter. Проект було перейменовано з Twitter Blueprint на Bootstrap. Реліз із відкритим кодом вийшов 19 серпня 2011 року. Нині проект підтримується невеликою групою розробників на чолі з Марком Отто та Джейкобом Торнтоном, а також широкою спільнотою прихильників.

Bootstrap сумісний з останніми версіями браузерів Google Chrome, Firefox, Internet Explorer, Opera і Safari (деякі з цих браузерів підтримуються не на всіх платформах).

Структура Bootstrap – модульна, вона здебільшого складається з наборів таблиць стилів LESS. Ці таблиці реалізують різні компоненти набору інструментів. Розробник у будь-який час може самостійно налаштувати файли Bootstrap, обираючи ті чи інші компоненти проекту, який знаходиться у розробці.

Основними інструментами фреймворку Bootstrap є:

- Сповіщення (alert) – класи, що використовуються для оформлення підказок та діалогових вікон.
- Шаблони (template) — шаблони сторінок (стандартні чи адаптивні).
- Типографіка (typography) — визначенні та опис класів для шрифтів, (наприклад, шрифт для коду та цитат).
- Форми (form) – засоби для оформлення форм та подій.
- Мультимедіа (media) – засоби управління, що використовуються для роботи з зображеннями та відео.
- Сітки (grid) — колонки, що готові до використання.
- Таблиці (table) – засоби для оформлення таблиць, що забезпечують сортування.
- Навігація (nav, navbar) – класи, що використовуються для оформлення вкладок, меню, сторінок та панелей навігації.
- Іконочний шрифт (icon font) – набір іконок у вигляді шрифту.

Фреймворк містить функціональні компоненти, які побудовані на основі мови JS з подальшим використанням jQuery. Вони містять наступні плагіни:

- Transitions – плавні зміни (використовується для налаштування останніх компонентів фреймворку).
- Modal – модальні вікна, як спливні, так і вбудовані в сторінку.
- Scrollspy – плагін, що автоматично міняє активний пункт в меню.
- Tab – вкладки (використовується для стилізованої навігації).

- Tooltip – спливні підказки, текстові елементи, що з'являються поряд із вказаним об'єктом після наведення курсору.
- Alert – інформаційні повідомлення (створюються класом `.alert` з можливістю закриття).
- Button – плагін для керуваннями станами кнопок.
- Collapse – згортання блочних елементів.
- Affix – плагін, що «приліплює» меню до одного з країв екрану при прокрутці сторінки[7].
- Carousel – мультимедійна галерея зображень.
- Dropdown – випадні списки, побудовані без тегу `select`.
- Popover – аналог спливних підказок.

## 2.7 Інтегровані середовища розробки

Інтегроване середовище розробки (ICP, англ. *Integrated development environment* або англ. *IDE*) – комплексне програмне рішення для розробки програмного забезпечення. Зазвичай, складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм. Більшість сучасних середовищ розробки мають можливість автодоповнення коду.

Деякі середовища розробки містять компілятор, інтерпретатор або ж обидва (наприклад *NetBeans* та *Eclipse*), інші не містять жодного з них (*SharpDevelop* та *Lazarus*). Деякі інтегровані середовища розробки містять систему керування версіями або інструменти для полегшення розробки графічного інтерфейсу користувача (GUI) (*XCode*, *Embarcadero Delphi*). Багато сучасних IDE містять інспектор класів, інспектор об'єктів, схему ієрархії класів для полегшення об'єктно-орієнтованої розробки програмного забезпечення.

Інтегровані середовища розробки створені для того, щоб максимізувати продуктивність програміста, надавши йому пов'язані інструменти розробки зі схожими інтерфейсами як одну програму, в якій відбуватиметься весь процес розробки й яка надає необхідні функції для модифікації, компілювання,



розгортання та налагодження програмного забезпечення. Протилежним до цього є підхід до розробки ПЗ, під час якого використовуються окремі інструменти, так як `vi`, `GCC` або `make`.

Одним із завдань IDE є зменшення часу, необхідного на конфігурацію різноманітних інструментів розробки, натомість пропонуючи той самий набір, як єдине ціле. Це може збільшити продуктивність розробника, у випадку, коли навчання тому, як працює інтегроване середовище розробки є швидшим, ніж освоєння всіх інструментів зокрема. Крім того, більша інтеграція між вбудованими інструментами потенційно може сприяти додатковому збільшенню продуктивності. Наприклад, синтаксичний аналіз коду може відбуватися безпосередньо під час його редагування, тим самим виявляючи помилки ще до трансляції коду.

Деякі інтегровані середовища розробки призначені для використання певної мови програмування (або декількох споріднених мов), надаючи набір можливостей, які більш підходять до парадигми програмування відповідної мови. Такими IDE є, наприклад `PhpStorm`, `Xcode`, `Hojo` та `Delphi`.

З іншої сторони, існує чимало більш універсальних IDE, які є багатомовними, наприклад `Eclipse`, `ActiveState Komodo`, `IntelliJ IDEA`, `MyEclipse`, `Oracle JDeveloper`, `NetBeans`, `Codenvy` and `Microsoft Visual Studio`.

Переважає більшість нинішніх IDE мають графічний інтерфейс користувача. До появи систем, які підтримують вікна, таких як `Microsoft Windows` та `X Window System (X11)`, широко використовувалися консольні інтегровані середовища розробки, такі як `Turbo Pascal`. Особливою їх прикметою було широке використання функціональних клавіш та поєднань клавіш для запуску команд або макросів, які часто використовувалися.

### **2.7.1 Інтегроване середовище розробки Cloud9**

`Cloud9 IDE` — це відкрите (з версії 3.0) онлайнове інтегроване середовище розробки. Підтримує сотні мов програмування, включно з `PHP`, `Ruby`, `Perl`, `Python`,

JavaScript з Node.js та Go. Дозволяє розробникам швидко починати розробку зі створеними робочими середовищами, працювати разом і мати живе прев'ю проекту, а також підтримку тестування в різноманітних браузерах.

Майже повністю програму написано мовою JavaScript з використанням Node.js для серверної частини. Використовується редактор Ace. На липень 2014-го редактор використовує контейнери Docker для своїх робочих просторів, а розміщується на Google Compute Engine[15].

Особливості:

- вбудований термінал з прм та базовими командами Unix;
- заповнення коду для фрагментів та ідентифікаторів;
- кілька курсорів для одночасного редагування;
- аналіз мови в режимі реального часу для JavaScript;
- паперизація, кронштейн та кодування відповідності символів;
- номери рядків, попередження про помилки;
- відладчик;
- інтерактивне керування файлами;
- теми;
- налаштовувані ключі, включаючи пресетів для Vim, Emacs та Sublime Text;
- вбудований редактор зображень;
- переформатування коду через JSBeautify та CSSLint;
- можливість перетягувати файли у ваш проект;
- підтримка наступних сховищ коду:
  - 1) GitHub
  - 2) Bitbucket
  - 3) Mercurial-сховища
  - 4) Git-сховища
  - 5) FTP-сервери
- підтримка розгортання:
  - 1) Heroku
  - 2) Joyent

- 3) Microsoft Azure
- 4) Google App Engine
- 5) SFTP/FTP

- підтримка публічних та приватних проектів;
- підтримка плагіна;
- підсвічування синтаксису для таких мов: C #, C / C ++, Clojure, CoffeeScript, ColdFusion, CSS, Apache Groovy, Java, JavaScript, LaTeX, Lua, Markdown, OCaml, PHP, Perl, PowerShell, Python, Ruby, Scala, SCSS, SQL, Textile, X (HTML), XML[8].

### **Висновки до другого розділу**

Розробка онлайн-системи замовлення їжі потребує вибору програмних засобів, за допомогою яких вона буде реалізована.

Для розробки використовуються наступні технології:

- HTML
- CSS
- JavaScript
- jQuery
- PHP
- CodeIgniter
- Bootstrap
- Cloud9

Використання цих технологій робить розробку максимально простою, а кінцевий результат зручним у використанні.

У розділі були вирішені завдання по вибору інструментів розробки та проаналізовані аспекти їх структури та використання.

## 3 ПРОЕКТУВАННЯ ТА РОЗРОБКА СЕРВІСУ

### 3.1 Проектування структури сервісу

Спираючись на поставлені задачі, було вирішено розробити наступні розділи та підрозділи:

- «Головна сторінка»

Сторінка містить інформацію про найкращі пропозиції тижня, кількість відвідувачів сайту, посилання на інші розділи (рис. 3.2).

- «Про нас»

Сторінка містить контактні дані сервісу та поля для відправки питань, які можна задати власникам сервісу (рис. 3.3).

- «Реєстрація»

Вікно містить перелік полів, необхідних для реєстрації нового користувача (рис. 3.4).

- «Авторизація»

Вікно містить поля для входу в систему (рис. 3.4).

- «Особистий кабінет»

Сторінка дає можливість зміни паролю та редагування профілю (рис. 3.5).

- «Кошик»

Сторінка містить інформацію про поточне замовлення та має функцію його оформлення (рис. 3.6).

- «Категорії»

Сторінка містить категорії пошуку доставок (рис. 3.7).

- «Доставки»

Сторінка є особистою сторінкою кожної доставки (рис. 3.8).

Після узагальнення всіх попередніх даних, була створена структурна схема сервісу, яка представлена на рис. 3.1.



Рисунок 3.1. – Структурна схема сервісу

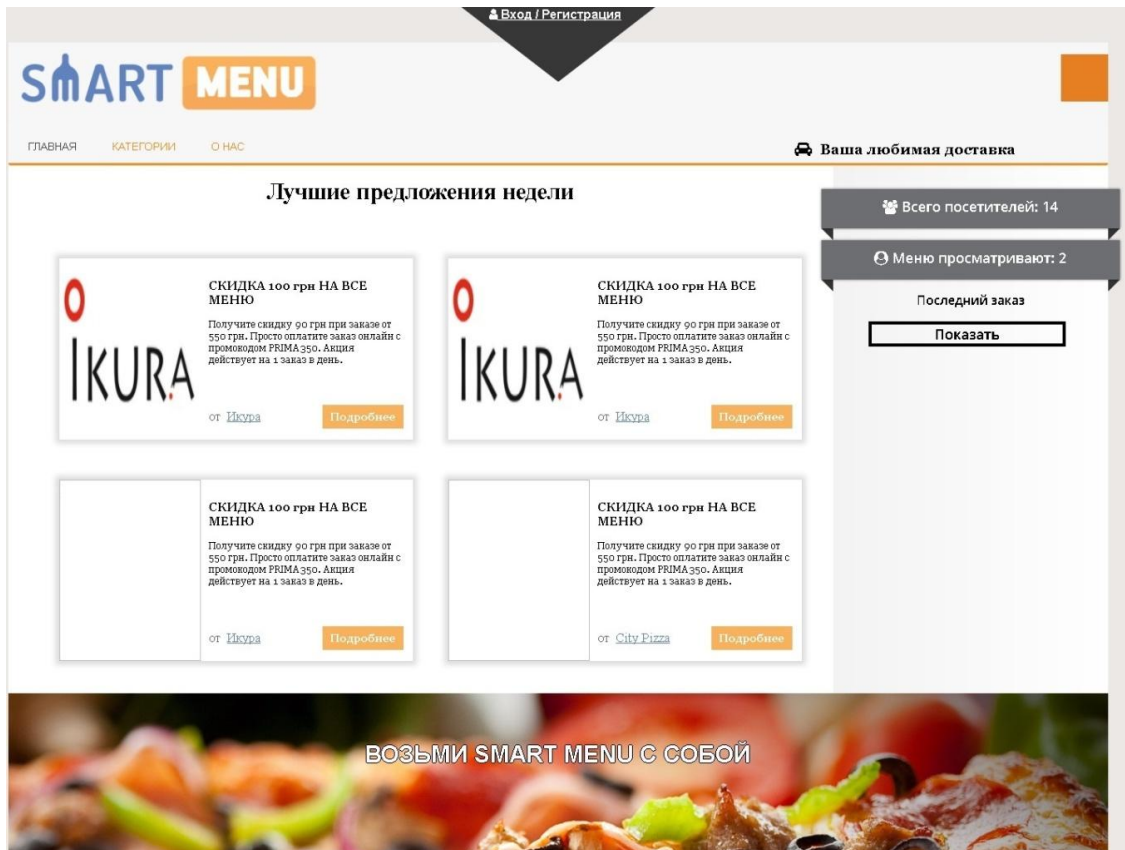


Рисунок 3.2 – Головна сторінка

**SMART MENU**

ГЛАВНАЯ КАТЕГОРИИ О НАС КОРЗИНА ЛИЧНЫЙ КАБИНЕТ ВЫХОД

🚗 **Ваша любимая доставка**

Ваше имя: \*

Ваш e-mail: \*

Тема:

Текст сообщения: \*

\* Обязательное поле

**Address**

Smart Menu  
10 10th Street NE United States  
Atlanta, GA 30309

Если вы хотите задать вопрос, внести предложение либо высказать претензии по качеству оказания услуг, заполните форму слева.

Контактный номер:  
**0800/ 567 345**

Время работы:  
**24/7**

КОМПАНИЯ

- Партнерство для ресторанов
- Пользовательское соглашение
- Условия проведения акций
- Контакты

ПОЛЕЗНЫЕ ССЫЛКИ

- Доставка пиццы
- Доставка суши
- Доставка еды

МЫ В СОЦ СЕТЯХ

- FACEBOOK
- INSTAGRAM
- TWITTER
- GOOGLE

Рисунок 3.3 – Про нас

**Регистрация**

Имя

Логин

Мобильный телефон

Электронная почта

Пароль

Согласен с [пользовательским соглашением](#)

**Личный кабинет**

Электронная почта или логин

Пароль

[Забыли пароль?](#)

✖ [Скрыть](#)

👤 [Вход / Регистрация](#)

Рисунок 3.4 – Реєстрація та авторизація

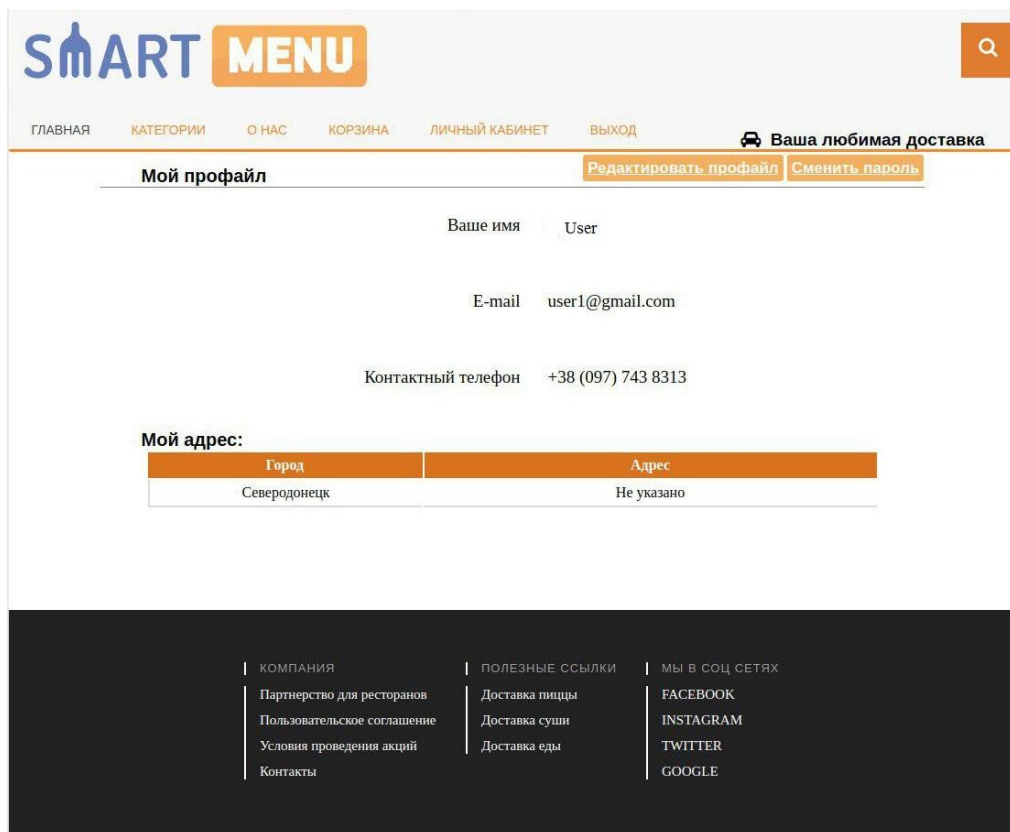


Рисунок 3.5 – Особистий кабінет

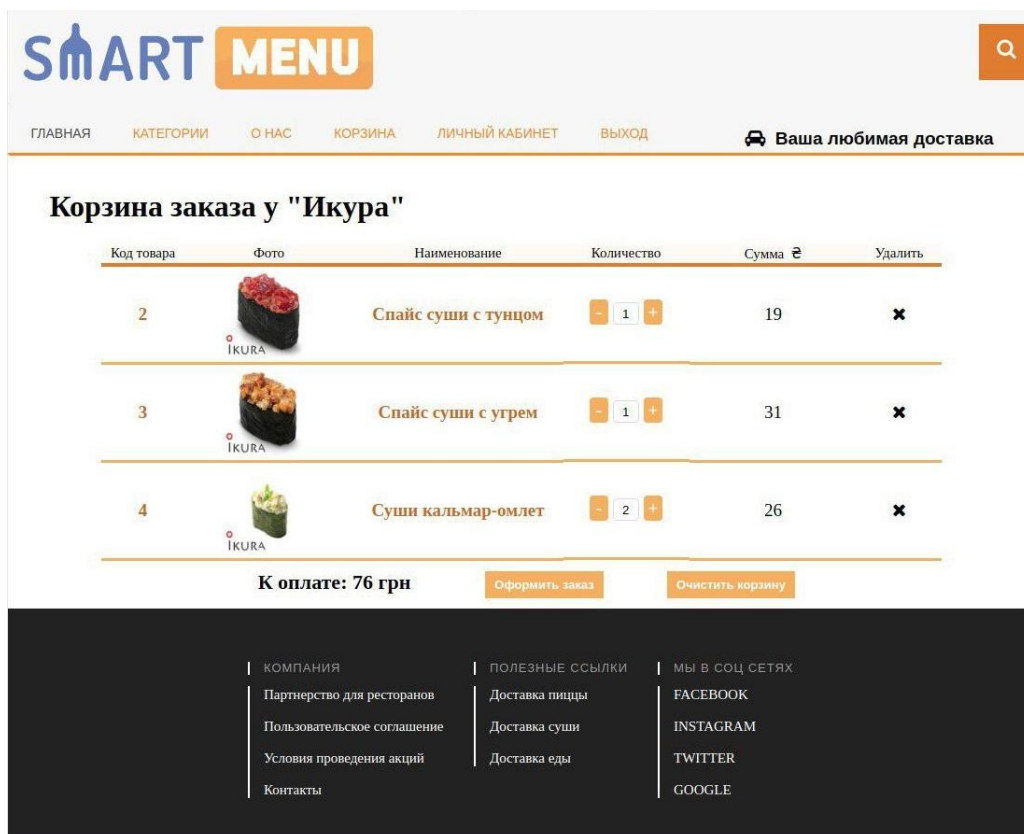


Рисунок 3.6 – Кошик

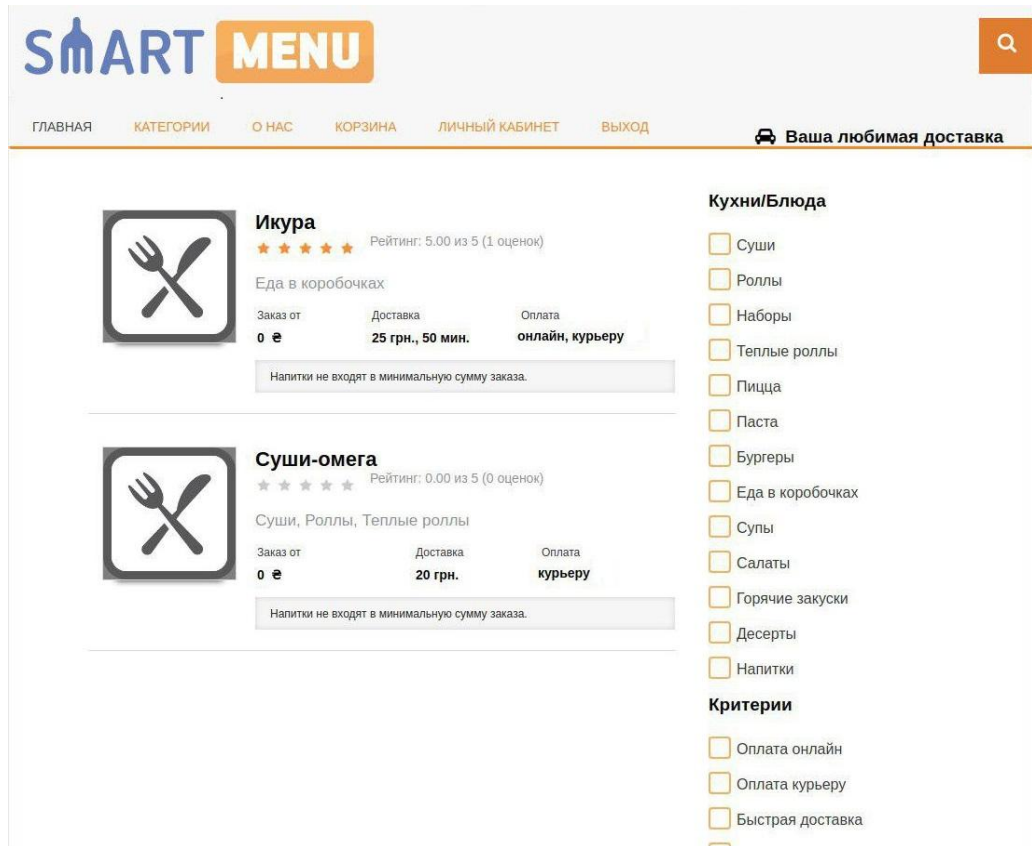


Рисунок 3.7 – Категорії

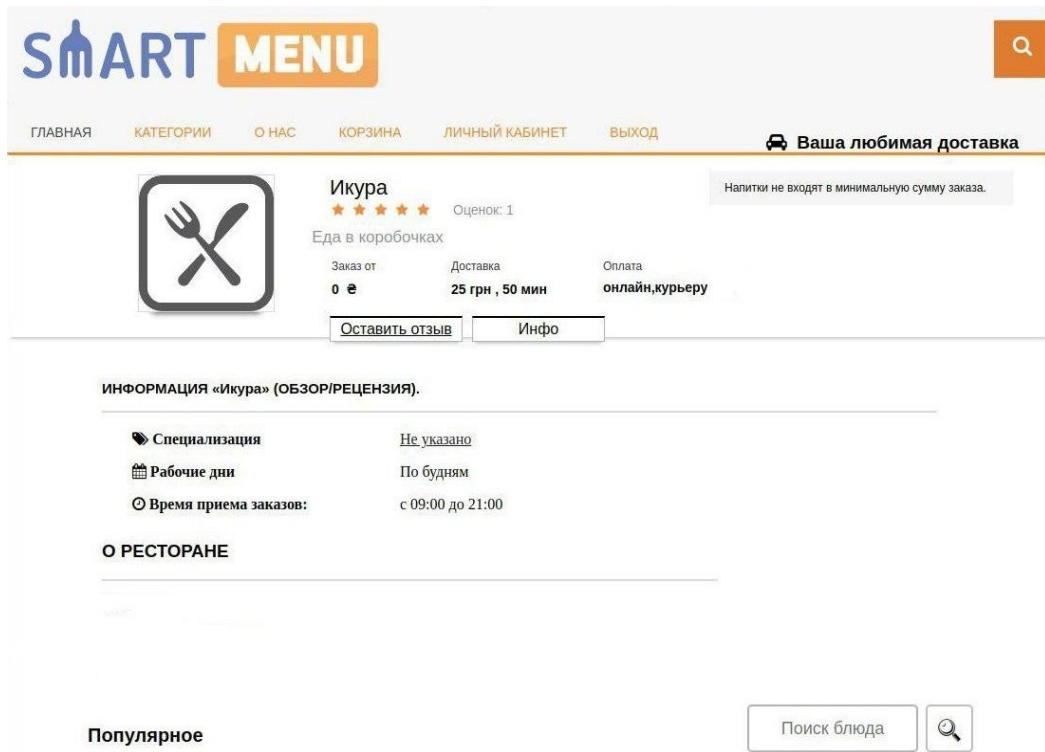


Рисунок 3.8 – Доставки



### 3.2 Розробка структури сервісу на основі фреймворку CodeIgniter

Для кращої структуризації та більш зручної розробки сервісу був використаний фреймворк CodeIgniter.

Структура web-додатку, створеного на основі фреймворку, відрізняється від усіх інших і має риси, що притаманні кожному окремому фреймворку. Це називають «каркасом».

CodeIgniter не є винятком і має свою структурні особливості. У якості основного архітектурного шаблону використовується MVC (model-view-controller або модель-вид-контролер).

Сайт, побудований відповідно до цієї архітектури складається з компонентів трьох видів:

- Контролер

Являє собою центр управління. Він отримує запити від браузера і обробляє їх.

- Модель

Використовується для роботи з даними (для читання, запису, поновлення даних у БД і т.д.)

- Вид

Вид формує html сторінку, яка відправляється браузеру.

Зв'язки між цими компонентами показані на рис. 3.9.

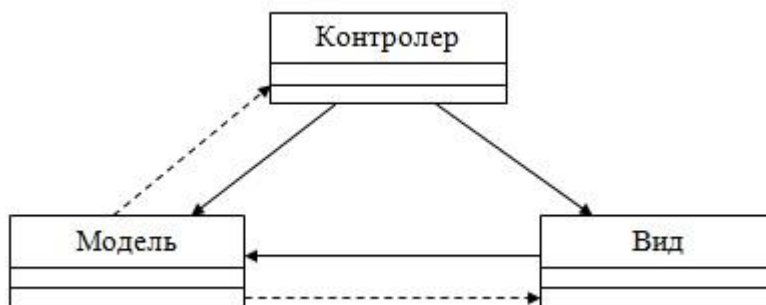


Рисунок 3.9 – Зв'язки між компонентами шаблону MVC

Структурний «каркас» також має свої особливості. Структуру файлів показано на рис. 3.10.

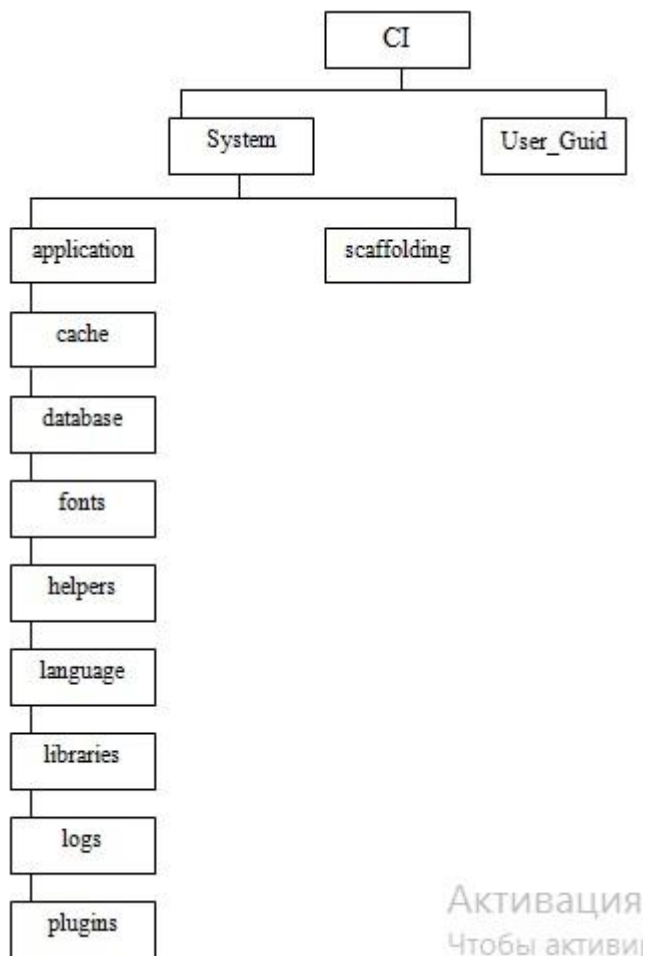


Рисунок 3.10 – Структурний «каркас» CodeIgniter

### 3.3 Розробка дизайну

Розробка дизайну веб-сайту потребує багато часу та є доволі трудомісткою задачею.

Це зумовлено тим, що людина, яка вперше заходить на сайт, у першу чергу оцінює не його наповнення, а зовнішній вигляд.

Саме тому у процесі розробки необхідно врахувати усі важливі чинники. Серед них можна виділити такі:

- розташування елементів управління;
- кольорова гамма;

- розташування важливої для клієнта інформації;
- структура, зрозуміла на інтуїтивному рівні;

Головним завданням при розробці ресурсу, який просуває свою продукцію чи продукцію інших виробників, є зацікавлення клієнта з перших хвилин перебування на сайті. Користуючись сервісом, людина не повинна відчувати дискомфорт та стикатися з незрозумілістю та нелогічністю. У кожний момент часу клієнт має відчувати, що знаходиться у межах ресурсу та може знайти будь-яку потрібну йому інформацію. Навігаційні та інформаційні елементи повинні бути помітними, а їх пошук повинен займати якомога менше часу.

Кольорова гамма має бути максимально спокійною для того, щоб довге знаходження на сайті не викликало неприємних зорових відчуттів та роздратування. Сервіс, який пов'язаний з продажем продуктів харчування, може містити яскраві кольори у поєднанні з зображеннями їжі для того, щоб привертати увагу та викликати апетит.

Важливим елементом є логотип. Він повинен добре запам'ятовуватися та відображати суть та ідею проекту. Розробка логотипу є одним з перших етапів розробки дизайну всього сервісу. Його зображення є неодмінним атрибутом дизайну та займає важливе місце на сайті. Логотип розробленого сервісу зображений на рис. 3.11.



Рисунок 3.11 – Логотип сервісу

Новим кроком у напрямку створення дизайну стала концепція чуйного дизайну, коли сайт сам вміє належним чином підлаштовуватися під розмір екрану користувача. Робиться це шляхом підвантаження різних стилів оформлення CSS при перегляді сайту на екранах різних пристроїв. Сам зміст і кольорове

оформлення сайту при цьому не змінюється, а змінюється лише спосіб угруповання інформаційних блоків. Такий дизайн вимагає більше зусиль від розробника, але завдяки фреймворку Bootstrap дана задача стає простішою.

Структура та приклад дизайну представлені на рис. 3.12 - 3.13.

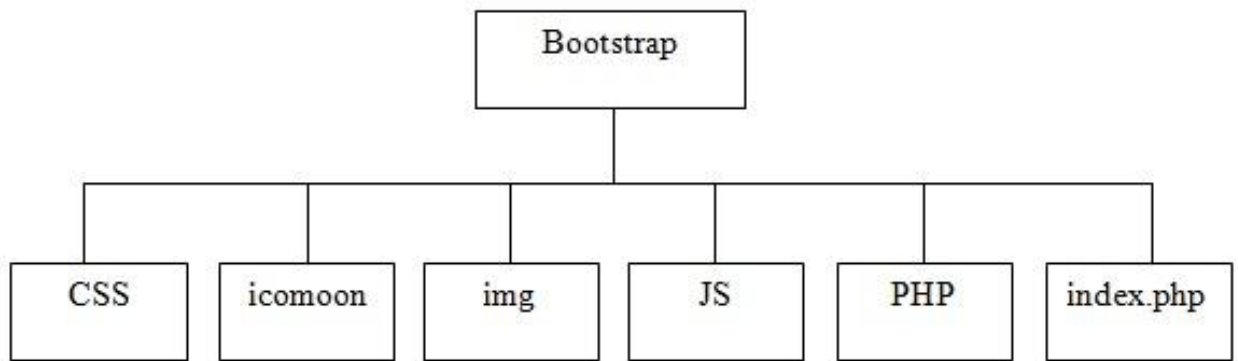


Рисунок 3.12 – Структура дизайну

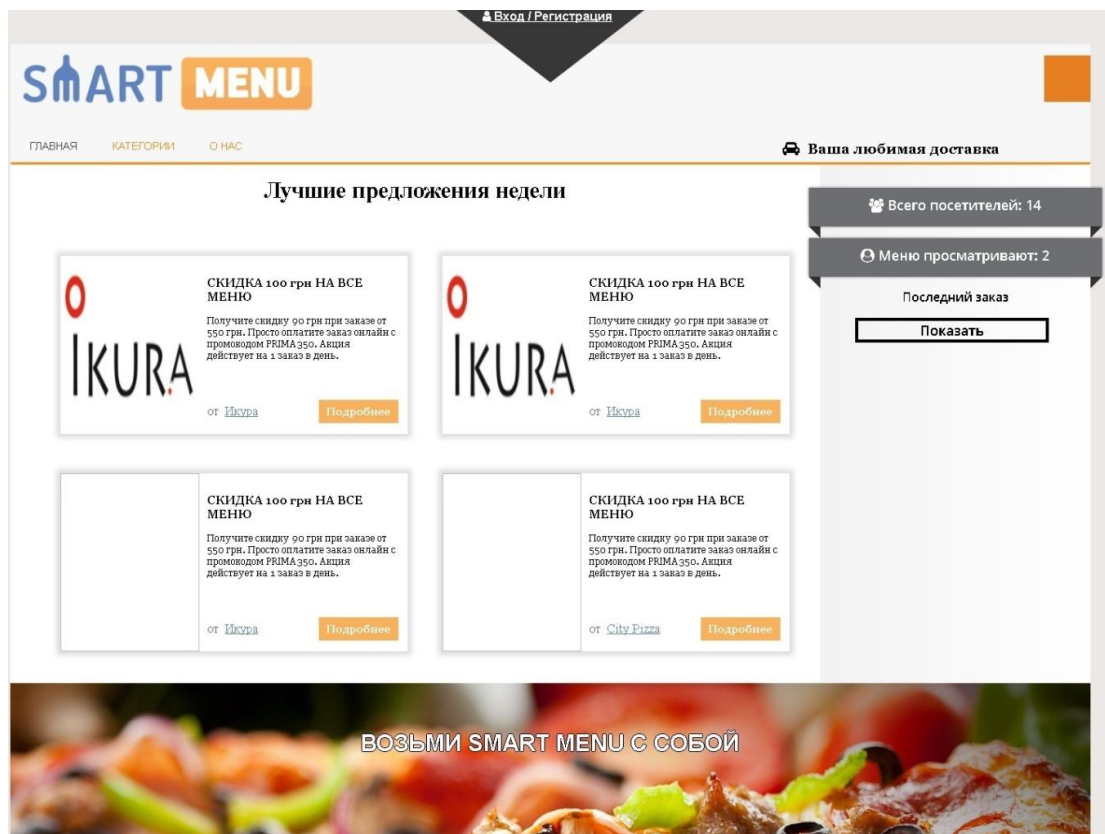


Рисунок 3.13 – Приклад дизайну

### 3.4 Сторінка реєстрації

Реєстрація на сайті – це дії, спрямовані на створення особистого облікового запису на web-ресурсі, з метою отримання доступу до його повного функціоналу (будь то розширені можливості або доступ до сторінок інтернет-ресурсів, переглядати які можливо лише після реєстрації).

Тобто, реєстрація – це спосіб входу (або отримання можливості входу) на певний web-ресурс. У більшості випадків цей процес є обов'язковим, особливо це стосується соціальних мереж, де можливості користувачів без реєстрації обмежені.

Реєстрація на сервісі онлайн-доставки їжі необхідна для власників доставок та клієнтів сервісу, щоб отримати доступ до повного функціоналу.

Сторінка реєстрації містить поля для обов'язкового заповнення. Коректність введених даних перевіряється за допомогою алгоритмів перевірки за заданими параметрами:

- логін (\$login)

Має містити не менше трьох символів. Проводиться перевірка на співпадання з існуючими логінами.

- email (\$email)

Не має містити сторонні символи. Проводиться перевірка на співпадання з зареєстрованими поштовими скриньками.

- ім'я та прізвище (\$name)

Має містити не менше трьох символів.

- пароль (\$pwd)

Має містити не менше шести символів.

- номер телефону (\$phone)

Має містити не менше п'яти символів та складатися тільки з чисел.

Блок-схему алгоритму реєстрації на сервісі представлено на рис. 3.14.

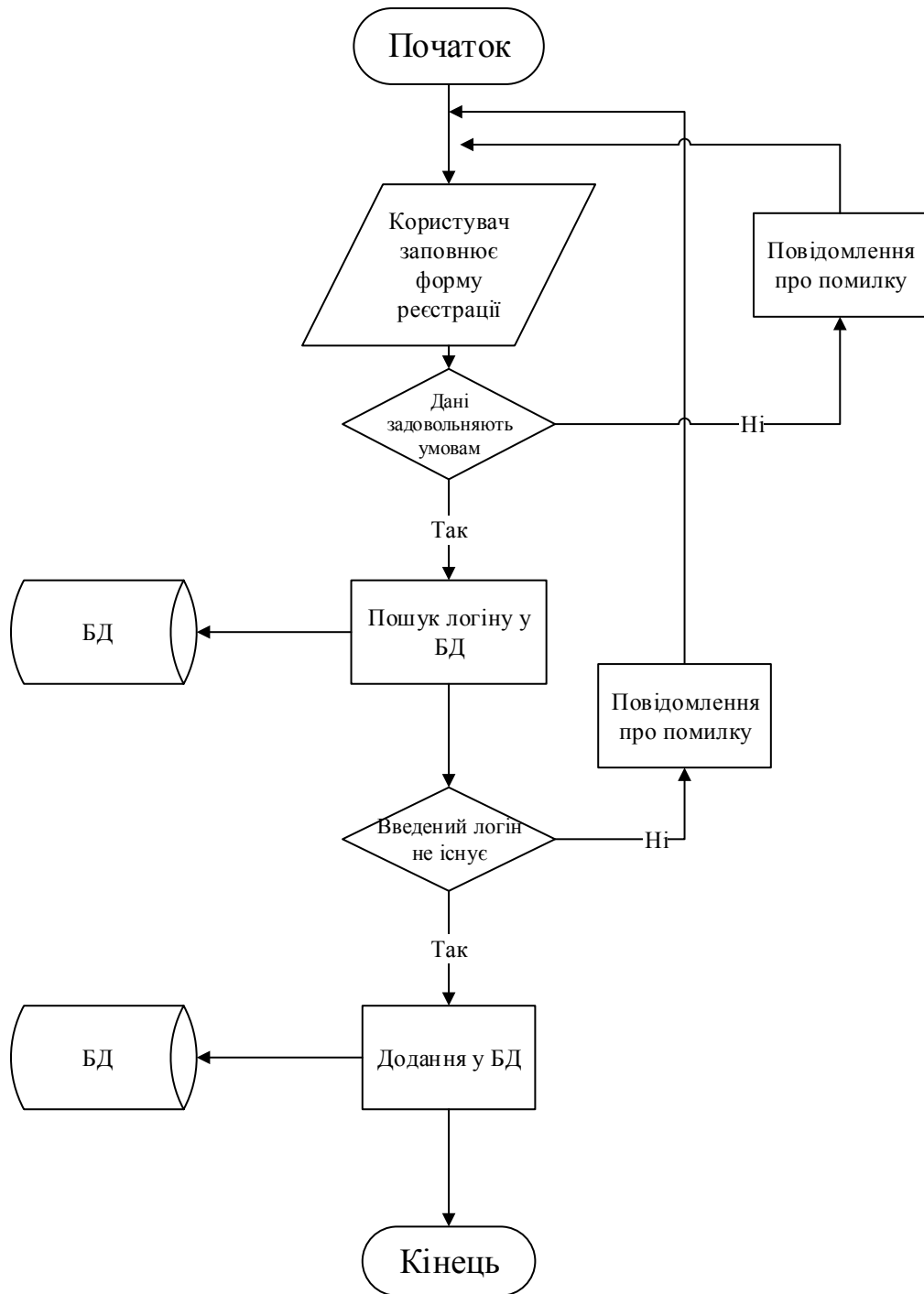


Рисунок 3.14 – Блок-схема алгоритму реєстрації

### 3.5 Авторизація та особистий кабінет

Авторизація – перевірка справжності пред'явленого користувачем ідентифікатора. Авторизація потрібна при доступі до таких інтернет-сервісів як:

- електронна пошта;
- веб-форум;
- соціальні мережі;
- інтернет банкінг;
- платіжні системи;
- корпоративні сайти;
- інтернет магазини.

Суть цього процесу полягає в обмеженні доступу до акаунту користувача і збереженні його особистих даних.

Особистий кабінет – власний простір користувача на сервісі. Після реєстрації та авторизація на сайті користувач має можливість змінити внесені дані саме у особистому кабінеті.

Зовнішній вигляд вікна авторизації зображений на рис. 3.15.

Личный кабинет

Электронная почта или логин

user1

Пароль

.....

Войти

[Забыли пароль?](#)

Рисунок 3.15 – Вікно авторизації

Блок-схема алгоритму авторизації представлена на рис. 3.16.

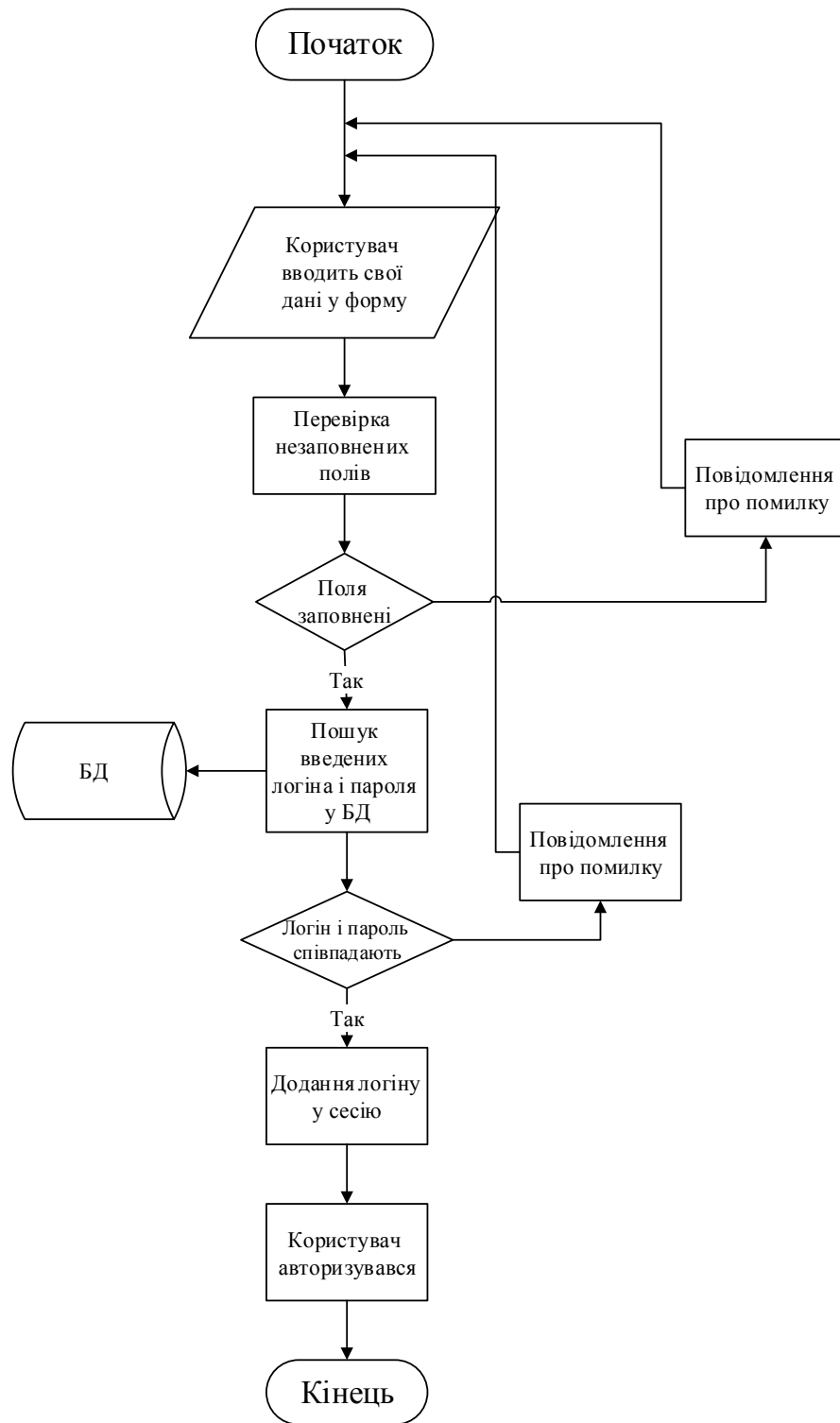


Рисунок 3.16. – Блок-схема алгоритму аутентифікації

Вікно зміни налаштувань зображено на рис. 3.17.



## Редактировать профиль Профиль

Имя	Артём
Фамилия	Борисовский
E-mail	user1@gmail.com
Контактный телефон	+38 (097) 743 8313
Дата рождения	06/06/2018
Город	Северодонецк
Адрес	Не указано

Сохранить

Рисунок 3.17 – Вікно зміни налаштувань

Блок-схему алгоритму зміни налаштувань зображено на рис. 3.18.

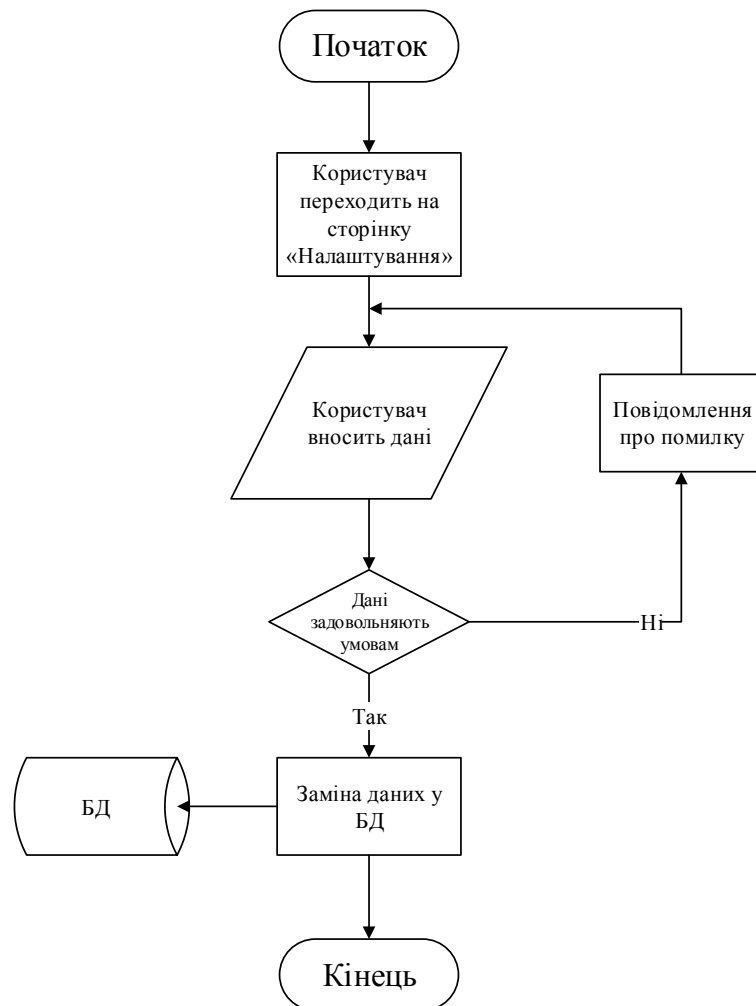


Рисунок 3.18 – Блок-схема алгоритму зміни налаштувань

### 3.6 Категорії і доставки

Основним та головним контентом агрегатору доставки їжі є доставки, які на ньому знаходяться. Доставка розділені за категоріями, завдяки яким можна здійснити пошук.

Пошук здійснюється за наступними критеріями:

- кухня/страва;
- форма оплати;
- рейтинг;
- час доставки.

Приклад форми вибору категорій зображена на рис. 3.19.



The image shows a vertical list of food categories under the heading "Кухни/Блюда". Each category is preceded by a small, empty square checkbox. The categories listed are: Суши, Роллы, Наборы, Теплые роллы, Пицца, Паста, Бургеры, Еда в коробочках, Супы, Салаты, Горячие закуски, Десерты, and Напитки.

Категорія
<input type="checkbox"/> Суши
<input type="checkbox"/> Роллы
<input type="checkbox"/> Наборы
<input type="checkbox"/> Теплые роллы
<input type="checkbox"/> Пицца
<input type="checkbox"/> Паста
<input type="checkbox"/> Бургеры
<input type="checkbox"/> Еда в коробочках
<input type="checkbox"/> Супы
<input type="checkbox"/> Салаты
<input type="checkbox"/> Горячие закуски
<input type="checkbox"/> Десерты
<input type="checkbox"/> Напитки

Рисунок 3.19 – Форма вибору категорій

На сервісі розміщені доставки, їх коротке описання, меню та умови замовлення. Кожна з них має власну сторінку на сервісі.

Розміщення доставки на сервісі виконується після узгодження між сервісом та її власником. Алгоритм розміщення зображено на рис. 3.20.

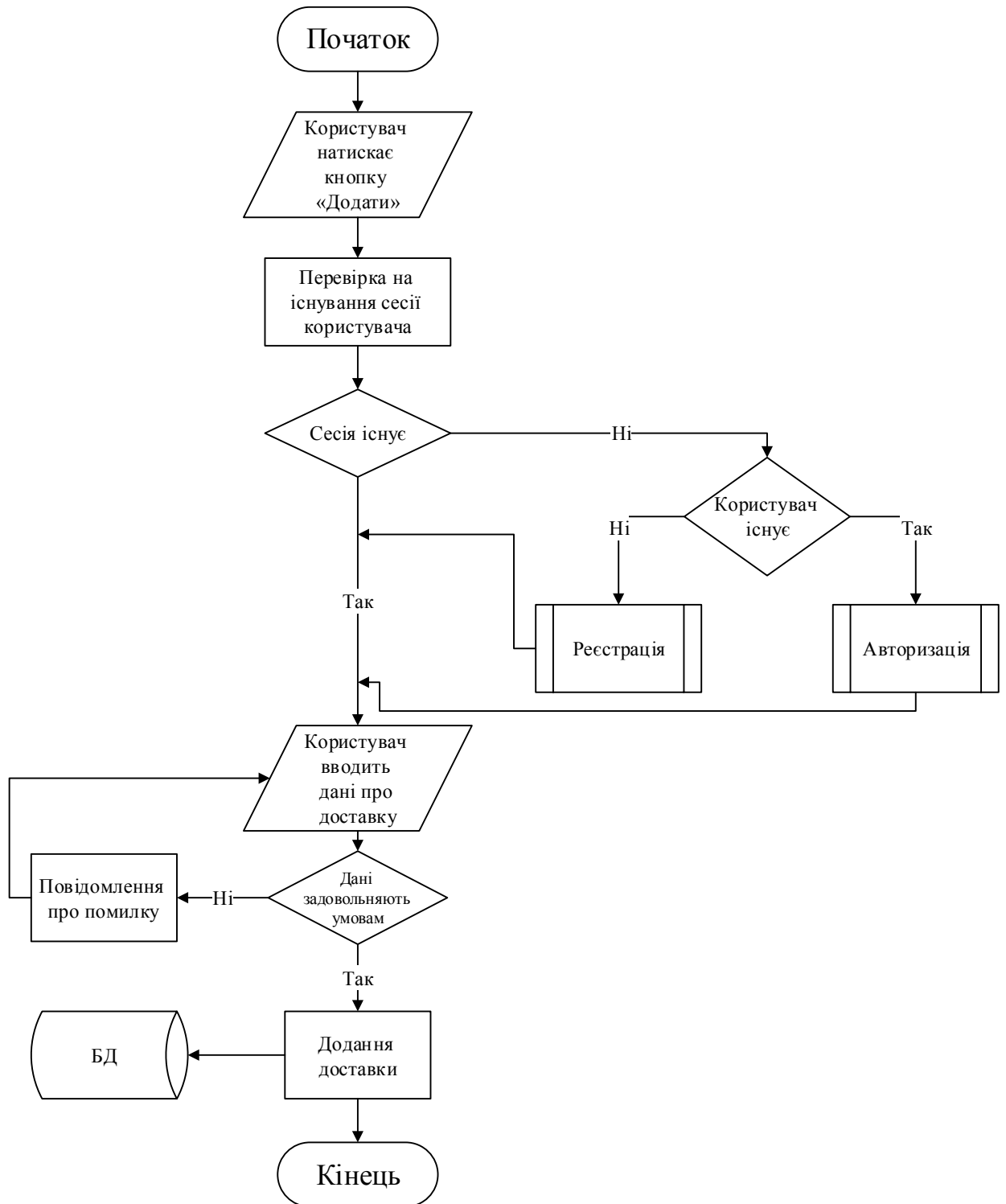


Рисунок 3.20 – Алгоритм розміщення доставки

### 3.7 Кошик

Кошик – елемент інтерфейсу будь-якого сервісу, пов'язаного з продажами. За допомогою кошика здійснюється формування замовлення, його наочна демонстрація та оформлення. Кошик є невід'ємною частиною багатьох сервісів.

Необхідність кошика для користувача базується на наступних чинниках:

- структуризація покупок;
- зручне корегування списку покупок;
- підрахунок підсумкової ціни;
- зручне оформлення замовлення.

Врахування цих чинників призводить до залучення нових клієнтів і збільшення середнього розміру замовлення на кілька пунктів. Саме тому цей елемент інтерфейсу важливий для створення успішного сервісу.

Кошик повинен виконувати наступні функції:

- додавання товару;
- видалення товару;
- підрахунок остаточної суми;
- формування замовлення.

Крім того, кількість кожного товару можна регулювати за допомогою кошика. Дані про зміну кількості товарів вносяться у базу даних, після чого кінцевий рахунок також змінюється.

Кошик для сервісу був створений за допомогою мови PHP та її бібліотек.

Блок-схему алгоритму роботи кошику представлено на рис. 3.21.

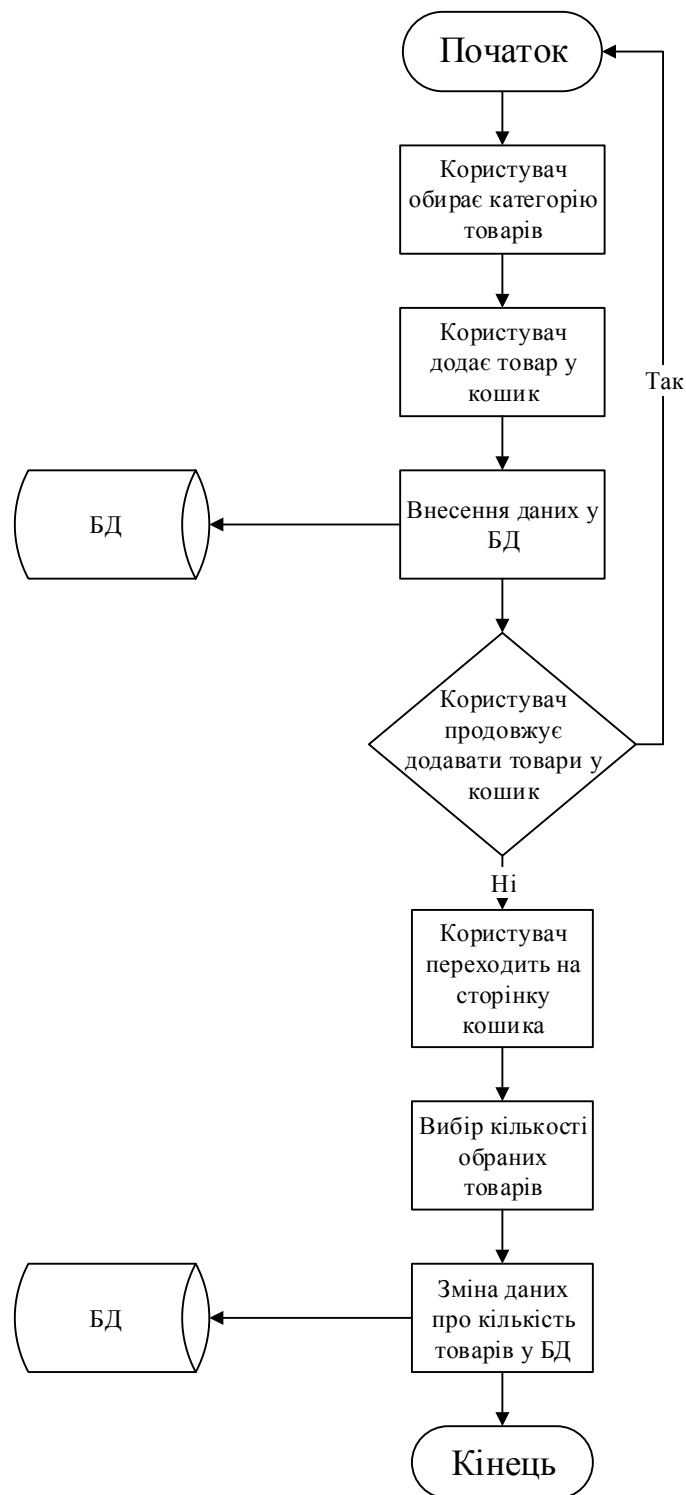


Рисунок 3.21 – Блок-схема алгоритму роботи кошику

### 3.8 Оформлення замовлення

Результатом роботи всієї системи агрегатора доставки їжі стає формування та оформлення замовлення.

Оформлення замовлення – процес, що завершує використання сервісу, пов’язаного з продажами та є його кінцевою метою.

Для оформлення замовлення необхідне існування поточної сесії користувача, бо оформлення замовлення можливе тільки після реєстрації. Якщо сесія не існує, то користувачу пропонується авторизуватися або зареєструватися на сервісі. Після цього замовлення відправляється на обробку.

Алгоритм роботи функції оформлення замовлення представлено на рис. 3.22.

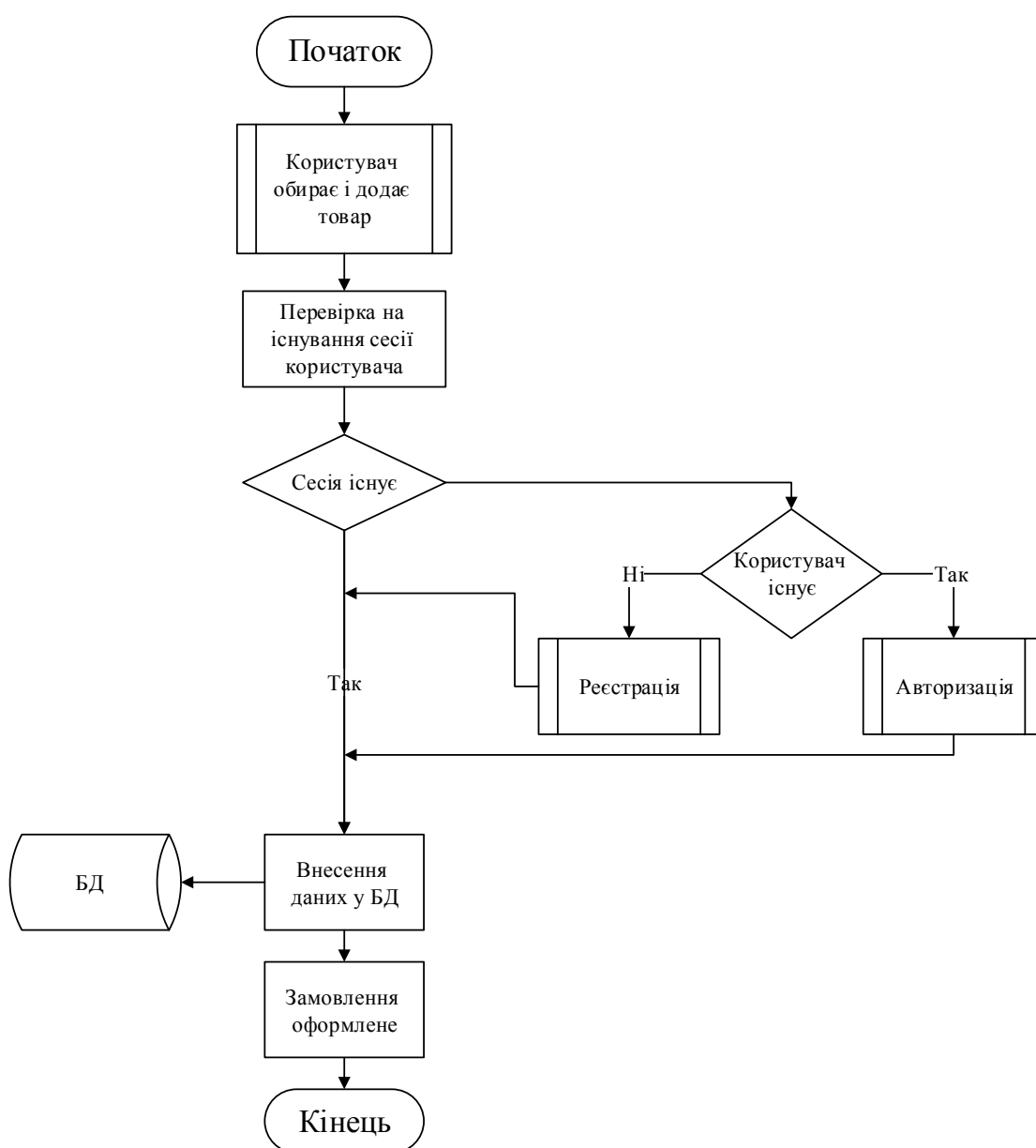


Рисунок 3.22 – Алгоритм роботи функції оформлення замовлення

### 3.9 Оцінки та коментарі

Критеріями оцінювання послуг, що представлені розміщеними на сервісі доставками, є коментарі та оцінки.

Відгуки та оцінки на сайті – важливе джерело інформації для клієнтів і власників доставок. Клієнти можуть обирати місця для замовлення і страви, спираючись на оцінки людей, які вже отримали свої замовлення. Власники ж, отримуючи негативні відгуки, можуть виправити наявні недоліки. Залишати оцінки і коментарі можуть тільки зареєстровані користувачі.

Форма додавання оцінки зображена на рис. 3.23.

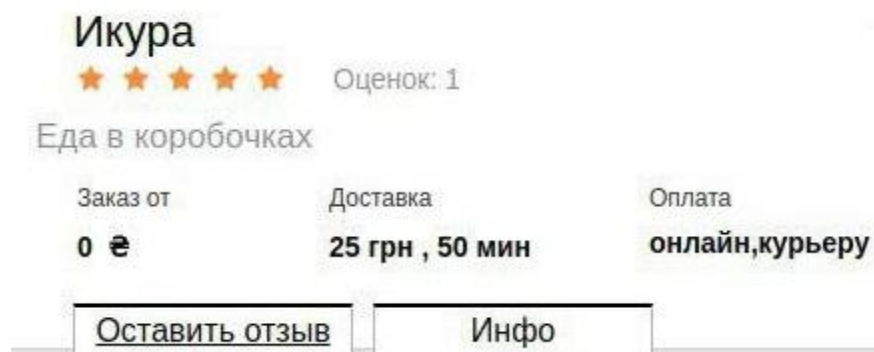


Рисунок 3.23 – Форма додавання оцінки

Форма додавання коментарів зображена на рис. 3.24.

**ОСТАВИТЬ ОТЗЫВ**

Выберите тип отзыва:

Комментарий:

Рисунок 3.24 – Вікно додавання коментарів

Алгоритм додавання коментарів представлено на рис. 3.25.

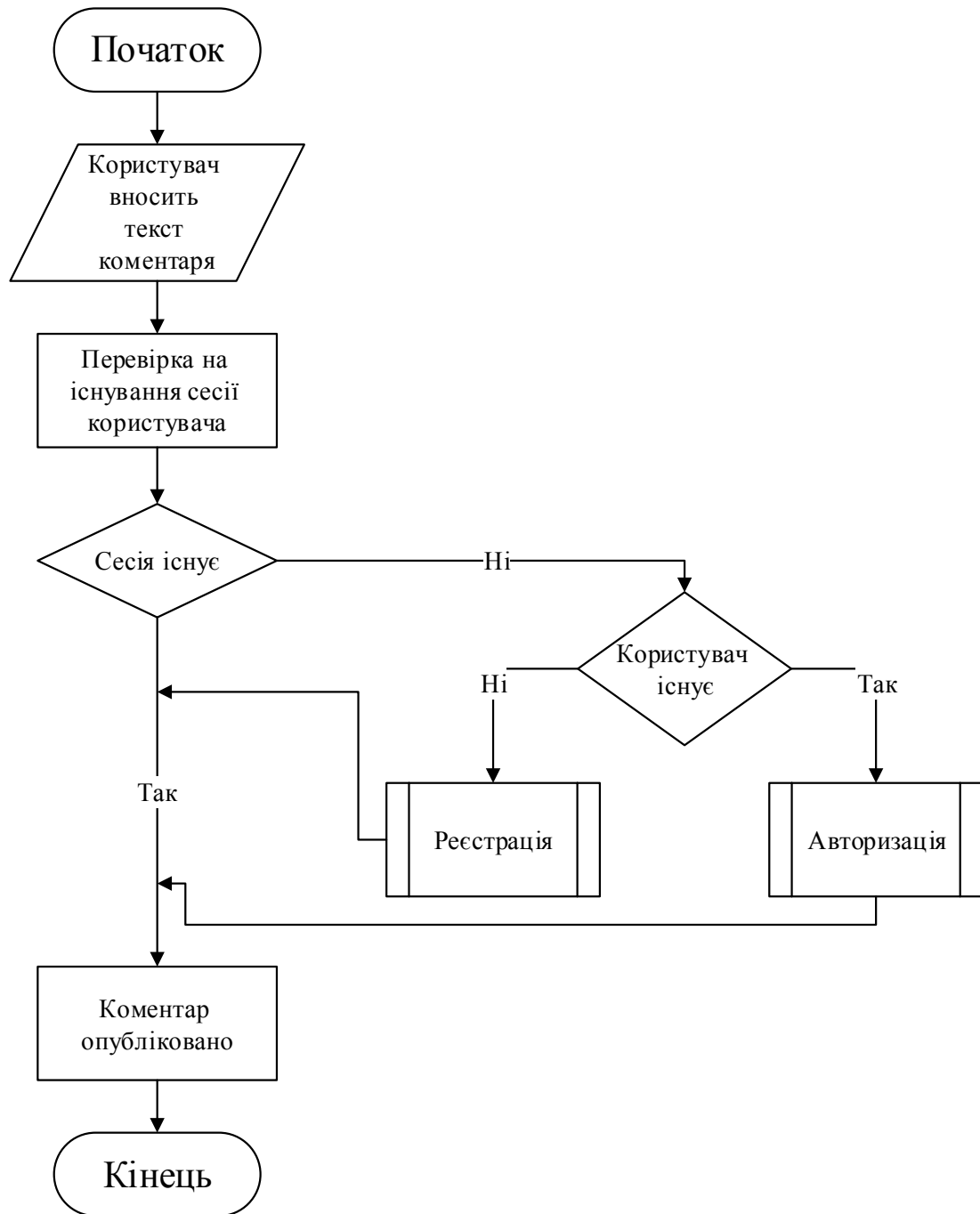


Рисунок 3.25 – Алгоритм додавання коментарів

### Висновки до третього розділу

У розділі описано проектування та програмна реалізація онлайн-сервісу замовлення їжі. Для пояснення процесу реалізації приведені блок-схеми алгоритмів роботи основних програмних функцій та зображення інтерфейсу та сторінок.



У розділі вирішуються наступні питання:

- проектування структури сервісу;
- розробка структури сервісу;
- розробка дизайну;
- розробка вікна реєстрації;
- розробка сторінок авторизації та особистого кабінету;
- розробка сторінок категорій та доставки;
- розробка кошику;
- розробка сторінки оформлення замовлення;
- розробка вікна коментарів.

У результаті всіх етапів проектування та розробки був розроблений сервіс замовлення їжі, який відповідає поставленим задачам. Код розробки представлений у додатку А.

## 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної роботи бакалавра було створення системи он-лайн замовлення їжі, і як результат було створено онлайн - систему, яка виконує всі зазначені вимоги. Так як в процесі проектування використовувався персональний комп'ютер, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде розроблятися та використовуватися розроблена система.

### 4.1 Аналіз стану умов праці

#### 4.1.1 Вимоги до приміщення

Таблиця 4.1 - Розміри приміщення.

Найменування	Значення
Довжина, м	4,4
Ширина, м	2,8
Висота, м	2,5
Площа, м <sup>2</sup>	12,32
Об'єм, м <sup>3</sup>	30,8

Згідно з ДСН 3.3.6.042-99 [18] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для забезпечення потрібного рівного освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для

дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

#### 4.1.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за ДСанПіН 3.3.2.007-98 [19] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне Значення	Нормативне Значення
Висота робочої поверхні, мм	700	680 ÷ 800
Висота простору для ніг, мм	650	не менше 600
Ширина простору для ніг, мм	540	не менше 500
Глибина простору для ніг, мм	660	не менше 650
Висота поверхні сидіння, мм	420	400 ÷ 500
Ширина сидіння, мм	410	не менше 400
Глибина сидіння, мм	420	не менше 400
Висота поверхні спинки, мм	500	не менше 300
Ширина опорної поверхні спинки, мм	400	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	750	700 ÷ 800

#### 4.2 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання

необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

#### 4.2.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-1.28-10 [22], яке встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга  $U=+220\text{В} \pm 5\%$ ;
- робочий струм  $I=2\text{А}$ ;
- споживана потужність  $P=350\text{ Вт}$ .

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна Оцінка	Нормативні Документи
1	2	3	4
<b>Фізичні:</b>			
підвищена або знижена вологість повітря	-//-	3	[18]
підвищений рівень напруги електричної мережі, замикання	-//-	3	[20] [21]

якої може відбутися через тіло людини			
<b>Психофізіологічні:</b>			
1	2	3	4
нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- формулювання теми; - пошук інформацію про предметну область; - пошук інформації про наявні аналоги; - проектування структур та алгоритмів; - виконання роботи; - оформлення записки.	4	[21] [22]
фізичні (статичне - сидіння)	порушення умов організації робочого часу (безперервна робота)	2	[19] [22]

Робочі місця в обов'язковому порядку повинні відповідати вимогам до санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, що затверджені постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [19].

#### 4.2.2 Пожежна безпека

Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80...100°C). При проходженні електричного струму по провідниках і деталей виділяється тепло, що в умовах їх високої щільності може привести до перегріву, і може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання між струмоведучими частинами обладнання (шини, електроди).

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу.

В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

- 1) поліамід - матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420 °С;
- 2) полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С;
- 3) склотекстоліт ДЦ - матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- 4) пластикат кабельний №489 - матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1;
- 5) деревина - будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255 °С, температура самозаймання 399 °С.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходитися пожежонебезпечні речовини і матеріали відповідно до НАПБ Б.03.002-2007 [23] відносяться до пожежонебезпечної зони класу П-Па. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та важкозаймисті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, присвоюється II ступень вогнестійкості.

Причинами можливого загоряння і пожежі можуть бути:

- 1) несправність електроустановки;
- 2) конструктивні недоліки устаткування;
- 3) коротке замикання в електричних мережах;
- 4) запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол [24].

### **4.2.3 Електробезпека**

Виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

### 4.3 Гігієнічні вимоги до параметрів виробничого середовища

#### 4.3.1 Мікроклімат

Мікроклімат робочих приміщень - це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт 1а. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають ДСН 3.3.6.042-99 [18] і наведені в табл. 4.4:

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

Період Року	Категорія Робіт	Температура С <sup>0</sup>	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	Легка-1а	22-24	40-60	0,1
Тепла	Легка-1а	23-25	40-60	0,1

#### 4.3.2 Освітлення

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше -1/8, в побутових - 1/10:

$$S_b = \left(\frac{1}{5} / \frac{1}{10}\right) * S_n \quad (4.1)$$

де  $S_b$  – площа віконних прорізів, м<sup>2</sup>;

$S_n$  – площа підлоги, м<sup>2</sup>.

$$S_n = a \cdot b = 4,4 \cdot 2,8 = 12,32 \text{ м}^2,$$

$$S = 1/10 \cdot 25 = 1,232 \text{ м}^2.$$

Приймаємо 1 вікно площею  $S=1,6 \text{ м}^2$ .



Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 4,4 м, ширина 2,8 м, світильниками ЛПО2П, оснащеними лампою типа ЛБ (одна - 80 Вт) з світловим потоком 5400 лм. Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників  $n$  виробляється по формулі (4.2):

$$n = \frac{E * S * Z * K}{F * U * M} \quad (4.2)$$

де  $E$  - нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  - освітлювана площа,  $m^2$ ;  $S = 12,32 m^2$ ;

$Z$  - поправочний коефіцієнт світильника ( $Z = 1,15$  для ламп розжарювання та ДРЛ;  $Z = 1,1$  для люмінесцентних ламп) приймаємо рівним 1,1;

$K$  - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

$U$  - коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п.

- 0,575  $M$  - число люмінесцентних ламп в світильнику - 1;

$F$  - світловий потік лампи - 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 * 12,32 * 1,15 * 1,5}{5400 * 0,575 * 1} \approx 2,0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, оснащених лампами типа ЛБ (одна - 80 Вт) зі світловим потоком 5400 лм.

#### 4.4 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при  $V$  приміщення  $> 40 \text{ м}^3$  на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП. Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

#### 4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала  $2/3$  нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціювання повітря:
  - а) якщо об'єм приміщення  $20 \text{ м}^3$ , то потрібно подати не менш як  $30 \text{ м}^3$  /год повітря;
  - б) якщо об'єм приміщення у межах від  $20$  до  $40 \text{ м}^3$ , то потрібно подати не менш як  $20 \text{ м}^3$ /год повітря;
  - в) якщо об'єм приміщення становить понад  $40 \text{ м}^3$ , допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;
- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;

#### **4.5.1 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).**

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [25], приміщення в якому проводяться всі роботи відносяться до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контуру заземлення повинен мати не більше 4 Ом.

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів  $R_{шт.з.}$ :

$$R_{шт.з.} = \frac{R_0 * R_{пр.з.}}{R_{пр.з.} - R_0} \quad (4.3)$$

де  $R_{пр.з.}$  - опір природних заземлювачів;

$R_0$  - допустимий опір заземлення.

Якщо природні заземлювачі відсутні, то  $R_{шт.з.} = R_0$ .

Підставивши числові значення у формулу (А.3), отримуємо:

$$R_{ум.з} = \frac{4 * 40}{40 - 4} \approx 40 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту  $\rho$ , Ом·м. Приблизне значення питомого опору глини приймаємо  $\rho=40$  Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту,  $R_{розр}$ , Ом·м, визначається відповідно для вертикальних заземлювачів  $R_{розр.в}$ , і горизонтальних  $R_{розр.г}$ , Ом·м за формулою:

$$P_{розр.} = \Psi * \rho \quad (4.4)$$

де  $\Psi$  - коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів

$R_{розр.в}=1,7$  і горизонтальних  $R_{розр.г}=5,5$  Ом·м

$$P_{розр.в} = 1,7 * 40 = 68 \text{ Ом/м}$$

$$P_{розр.г} = 5,5 * 40 = 220 \text{ Ом/м}$$

4) Розраховується опір розтікання струму вертикального заземлювача  $R_B$ , Ом, за (4.5).

$$R_B = \frac{P_{розр.в}}{2 * \pi * l_B} * \left( \ln \frac{2 * l_B}{d_{ст}} + \frac{1}{2} * \ln \frac{4 * t + l_B}{4 * t - l_B} \right) \quad (4.5)$$

де  $l_B$  - довжина вертикального заземлювача (для труб - 2–3 м;  $l_B=3$  м);

$d_{ст}$  - діаметр стержня (для труб - 0,03–0,05 м;  $d_{ст}=0,05$  м);

$t$  - відстань від поверхні землі до середини заземлювача, яка визначається за ф. (4.6):

$$t = h_B + \frac{l_B}{2} \quad (4.6)$$

де  $h_B$  - глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м}$$

$$R_B = \frac{68}{2 * \pi * 3} * \left( \ln \frac{2 * 3}{0,05} + \frac{1}{2} * \ln \frac{4 * 2,3 + 3}{4 * 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів  $n$  штук, без урахування коефіцієнта використання  $\eta_B$ :

$$n = \frac{2 * R_B}{R_0} = \frac{2 * 18,5}{4} = 9,25 \quad (4.7)$$

6) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання  $n_B$ , шт:

$$n_B = \frac{2 * R_B}{R_0 * \eta_B} = \frac{2 * 18,5}{4 * 0,57} = 16,2 \approx 16 \quad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м:

$$l_c = 1,05 * L_B * (n_B - 1) \quad (4.9)$$

де  $L_B$  - відстань між вертикальними заземлювачами, (прийняти за  $L_B = 3$ м);  
 $n_B$  - необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 * 3 * (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки)  $R_G$ , Ом:

$$R_G = \frac{P_{розр,г}}{2 * \pi * l_c} * \ln \frac{2 * l_c^2}{d_{cm} * h_G} \quad (4.10)$$

де  $d_{cm}$  - еквівалентний діаметр смуги шириною  $b$ ,  $d_{cm} = 0,95b$ ,  $b = 0,15$  м;

$h_G$  - глибина закладання горизонтальних заземлювачів (0,5 м);

$l_c$  - довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м

$$R_G = \frac{220}{2 * \pi * 48} * \ln \frac{2 * 48^2}{0,95 * 0,15 * 0,5} = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача  $\eta_c$ , відповідно до необхідної кількості вертикальних заземлювачів  $n_B$ . Коефіцієнт використання з'єднувальної смуги  $\eta_c = 0,3$  (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{заг} = \frac{R_B * R_G}{R_B * \eta_c + R_G * n_B * \eta_B} \quad (4.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова:  $R_{заг}$  Ом, а саме:

$$R_{\text{заг}} = \frac{18,5 * 8,1}{18,5 * 0,3 + 8,1 * 16 * 0,57} = 1,9 \leq R_0$$

### **Висновки до четвертого розділу**

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

## ВИСНОВКИ

Мережа Інтернет проникає у всі сфери людського життя, що призводить до появи нових галузей економіки, ринкових відносин, маркетингу та багатьох інших. Вплив новітніх технологій стає невід'ємною частиною загального прогресу.

Кількість сервісів, які допомагають користувачам економити час, зростає з кожним роком. Також, вони нерозривно пов'язані з рекламою, яка є важливим засобом просування.

Ринок foodtech розвивається завдяки новим проектам та отримує значне фінансування. Агрегатори доставки їжі з'являються у багатьох країнах світу і є перспективною галуззю ринку.

Метою виконання дипломної роботи була розробка системи он-лайн замовлення їжі. Для розробки подібного сервісу необхідні спеціалізовані технології, які були описані у роботі.

Для вирішення цієї задачі було проаналізовано та використано такі технології, як:

- HTML
- CSS
- JavaScript
- jQuery
- PHP
- MySQL
- CodeIgniter
- Bootstrap
- Cloud9

За допомогою даних технологій було розроблено сервіс он-лайн замовлення їжі, повністю готовий до застосування, були виконані всі заплановані завдання та вимоги.

Були поставлені та вирішені наступні завдання:

- 1) Проектування структури сервісу.
- 2) Розробка структури сервісу.
- 3) Розробка дизайну.
- 4) Розробка вікна реєстрації.
- 5) Розробка сторінок авторизації та особистого кабінету.
- 6) Розробка сторінок категорій та доставки.
- 7) Розробка кошику.
- 8) Розробка сторінки оформлення замовлення.
- 9) Розробка вікна коментарів.

У результаті проведеної роботи були проаналізовані умови праці, шкідливих та небезпечних чинників, з якими стикається робітник. Були визначені параметри і характеристики приміщення для роботи над проектом та описані заходи, які необхідні, щоб використане приміщення відповідало необхідним нормам і було комфортним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки.

Були наведені розміри приміщення, розраховано кількість ламп та захисне заземлення, а також - наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.



## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Джейсон Ленгсторф. PHP и jQuery для профессионалов = Pro PHP and jQuery. — М.: «Вильямс», 2011. — 352 с.
2. Jake Spurlock. Bootstrap. Responsive Web-Development. — O'Reilly, 2013. — 128 с.
3. Костарев А. Ф. PHP 5. — СПб.: «БХВ-Петербург», 2011. — 1104 с.
4. Дженнифер Н. Роббинс. HTML5, CSS3 и JavaScript. Исчерпывающее руководство = Learning Web Design, 4th Edition. — М.: Эксмо, 2014. — 528 с.
5. Дэвид Флэнаган. 15.4.6. Независимые от языка DOM-интерфейсы // JavaScript. Подробное руководство = JavaScript. The Definite Guide / Перевод А. Киселева. — 5-е изд. — СПб.: «Символ-Плюс», 2012. — 332 с.
6. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание = HTML, XHTML, and CSS Bible, 5th Edition. — М.: «Диалектика», 2013. — 656 с.
7. David Cochran. Twitter Bootstrap Web Development How-To. — Пакт, 2012. — 332 с.
8. Горбунов-Посадов М. М. Расширяемые программы. — М.: Полиптих, 2011. — 336 с.
9. Code Igniter [Электронный ресурс] // URL: [https://www.codeigniter.com/user\\_guide/](https://www.codeigniter.com/user_guide/) (дата звернения: 10.05.2018).
10. DMU [Электронный ресурс] // URL: <http://digital-me-up.com/2016/11/27/foodtech/> (дата звернения: 10.05.2018).
11. Entrepreneur [Электронный ресурс] // URL: <https://www.entrepreneur.com/article/285544> (дата звернения: 11.05.2018).
12. Proov [Электронный ресурс] // URL: <https://proov.io/blog/foodtech/> (дата звернения: 12.05.2018).
13. Web-технологии [Электронный ресурс] // URL: <https://htmlweb.ru/java/jquery.php> (дата звернения: 12.05.2018).

14. VR-онлайн [Електронний ресурс] // URL: <http://www.vr-online.ru/content/codeigniter-obzor-frejmworka-dlja-php-635> (дата звернення: 16.05.2018).
15. Wikipedia [Електронний ресурс] // URL: <https://uk.wikipedia.org/wiki/Cloud9> (дата звернення: 16.05.2018).
16. habr [Електронний ресурс] / Представлення (VIEW) в MySQL - Режим доступу: [www. URL: https://habr.com/post/47031/](http://www.habr.com/post/47031/) (дата звернення: 16.05.2018).
17. Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL. Библия программиста, 2-е издание = PHP 6 and MySQL 6 Bible. — М.: «Диалектика», 2011. — 218 с.
18. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. Міністерство охорони здоров'я України (МОЗ). Постанова № 42 від 01.12.1999
19. ДСанПІН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Міністерство охорони здоров'я України (МОЗ).
20. ГОСТ 12.1.030-81 ССБТ. Електробезпека. Захисне заземлення. Занулення.
21. ГОСТ 13109-97. Норми якості електричної енергії в системах електропостачання загального призначення.
22. НПАОП 0.00-1.28-10. Про погодження матеріалів правил охорони праці під час експлуатації електронно-обчислювальних машин.
23. НАПБ Б.03.002-2007. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою.
24. ГОСТ 12.1.044-89. Система стандартів безпеки праці. Вогнестійкість. Номенклатура показників і методи їх визначення (ІСО 4589-84).
25. НПАОП 40.1-1.01-97. Правила безпечної експлуатації електроустановок.

## ДОДАТОК А

### Лістинг програми

#### Файл index.php

```

<?php
define('ENVIRONMENT', isset($_SERVER['CI_ENV']) ? $_SERVER['CI_ENV'] :
'development');
    switch (ENVIRONMENT)
    {
        case 'development':
            error_reporting(-1);
            ini_set('display_errors', 1);
            break;
        case 'testing':
        case 'production':
            ini_set('display_errors', 0);
            if (version_compare(PHP_VERSION, '5.3', '>='))
            {
                error_reporting(E_ALL & ~E_NOTICE & ~E_DEPRECATED & ~E_STRICT &
~E_USER_NOTICE & ~E_USER_DEPRECATED);
            }
            else
            {
                error_reporting(E_ALL & ~E_NOTICE & ~E_STRICT & ~E_USER_NOTICE);
            }
            break;
        default:
            header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
            echo 'The application environment is not set correctly.';
            exit(1); // EXIT_ERROR
    }

    $system_path = 'system';
    $application_folder = 'application';
    $view_folder = '';
    if (defined('STDIN'))
    {
        chdir(dirname(__FILE__));
    }
    if (($temp = realpath($system_path)) !== FALSE)
    {
        $system_path = $temp.DIRECTORY_SEPARATOR;
    }
    else
    {
        // Ensure there's a trailing slash
        $system_path = rtrim(
rtrim($system_path, '/\\'),
'/\\',
DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
).DIRECTORY_SEPARATOR;
    }
    // Is the system path correct?
    if (! is_dir($system_path))
    {
        header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
        echo 'Your system folder path does not appear to be set correctly.
Please open the following file and correct this: '.pathinfo(__FILE__,
PATHINFO_BASENAME);
        exit(3); // EXIT_CONFIG
    }

```

```

    }
// The name of THIS file
define('SELF', pathinfo(__FILE__, PATHINFO_BASENAME));
// Path to the system directory
define('BASEPATH', $system_path);
// Path to the front controller (this file) directory
define('FCPATH', dirname(__FILE__).DIRECTORY_SEPARATOR);
// Name of the "system" directory
define('SYSDIR', basename(BASEPATH));
// The path to the "application" directory
    if (is_dir($application_folder))
    {
        if (($_temp = realpath($application_folder)) !== FALSE)
        {
            $application_folder = $_temp;
        }
        else
        {
            $application_folder = strstr(
                rtrim($application_folder, '\\'),
                '\\',
DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
            );
        }
    }
    elseif (is_dir(BASEPATH.$application_folder.DIRECTORY_SEPARATOR))
    {
        $application_folder = BASEPATH.strstr(
            trim($application_folder, '\\'),
            '\\',
DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
        );
    }
    else
    {
header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
echo 'Your application folder path does not appear to be set correctly.      Please
open the following file and correct this: '.SELF;
exit(3); // EXIT_CONFIG
    }
define('APPPATH', $application_folder.DIRECTORY_SEPARATOR);
// The path to the "views" directory
    if ( ! isset($view_folder[0]) &&
        is_dir(APPPATH.'views'.DIRECTORY_SEPARATOR))
    {
        $view_folder = APPPATH.'views';
    }
    elseif (is_dir($view_folder))
    {
        if (($_temp = realpath($view_folder)) !== FALSE)
        {
            $view_folder = $_temp;
        }
        else
        {
            $view_folder = strstr(
                rtrim($view_folder, '\\'),
                '\\',
DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
            );
        }
    }
    elseif (is_dir(APPPATH.$view_folder.DIRECTORY_SEPARATOR))

```

```

{
    $view_folder = APPPATH.strtr(
        trim($view_folder, '\\'),
        '\\',
        DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
    );
}
else
{
header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
    echo 'Your view folder path does not appear to be set correctly. Please open
the following file and correct this: '.SELF;
exit(3); // EXIT_CONFIG
}
define('VIEWPATH', $view_folder.DIRECTORY_SEPARATOR);
require_once BASEPATH.'core/CodeIgniter.php';

```

## **Файл Pages.php**

```

<?php
class Pages extends CI_Controller {
    public function view($page = 'home') {
        $this->load->library('session');
        $this->load->model('online_model');

        if (!file_exists(APPPATH.'views/pages/'.$page.'.php')) {
            show_404();
        }

        $data = array();
        $header = array();
        $data['title'] = $header['title'] = ucfirst($page);
        $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
        $data['in_menu'] = $this->online_model->get_in_menu();
        $data['users'] = $this->online_model->get_users_count();

        if ($data['user'] != false)
            $data['user']['deliveries'] = isset($this->session->token) ?
$this->auth_model->get_user_deliveries($this->session->token) : false;

        $data['title'] = $header['title'] = 'Smart Menu - Ваша любимая
доставка';
        $this->load->view('templates/header', $data);
        $this->load->view('pages/'.$page, $data);
        $this->load->view('templates/footer');
    }
}

```

## **Файл Register.php**

```

<?php
class Register extends CI_Controller {
    public function index() {
        $this->load->library('session');
        $data['title'] = 'Регистрация';
        $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;

```

```

$data['err'] = array();
if ($data['user'] !== false) {
    header("Location: ".base_url());
}
if (empty($_REQUEST)) {
    echo json_encode(array('status' => 'err', 'error' => 'No
parameters received. '));
    return;
    $this->load->view('templates/header', $data);
    $this->load->view('register/index', $data);
    $this->load->view('templates/footer');
    return;
}
$json = array();
$error = false;
$login = (string) $_REQUEST['login'];
$name = (string) $_REQUEST['name'];
$surname = 'Не указано'; // (string) $_REQUEST['surname'];
$pwd = (string) $_REQUEST['pwd'];
$pwd_confirm = $pwd; // (string) $_REQUEST['pwd-confirm'];
$email = (string) $_REQUEST['email'];
$city = 'Не указано'; // (string) $_REQUEST['city'];
$phone = (string) $_REQUEST['phone'];
if (strlen($login) < 3)
    $data['err']['login'] = "Логин слишком короток!";
if ($this->auth_model->is_user_exists($login))
    $data['err']['login'] = "Пользователь с таким логином уже
существует!";
if ($this->auth_model->is_email_exists($email))
    $data['err']['email'] = "Данный почтовый ящик уже используется!";
if (!preg_match('/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/ ',
$email))
    $data['err']['email'] = "Неверный формат почтового ящика!";
if (strlen($name) < 2)
    $data['err']['name'] = "Неверное имя!";
if (strlen($surname) < 2)
    $data['err']['surname'] = "Неверная фамилия!";
if (strlen($pwd) < 6)
    $data['err']['pwd'] = "Пароль слишком короток! Минимальная длина -
6";
if ($pwd !== $pwd_confirm)
    $data['err']['pwd-confirm'] = "Пароли не совпадают!";
if (strlen($city) < 3)
    $data['err']['city'] = "Некорректное название города!";
if (strlen($phone) < 7)
    $data['err']['phone'] = "Некорректный номер телефона!";
if (sizeof($data['err']) > 0) {
    echo json_encode(array('status' => 'err', 'errors' =>
$data['err']));
    return;
    $data['login'] = $login;
    $data['name'] = $name;
    $data['surname'] = $surname;
    $data['email'] = $email;
    $data['city'] = $city;
    $data['phone'] = $phone;
    $this->load->view('templates/header', $data);
    $this->load->view('register/index', $data);
    $this->load->view('templates/footer');
    return;
}

```

```

    }
    $query = $this->db->query("INSERT INTO users SET login=?, name=?,
surname=? ,pwd=? ,city=? , email=? , phone=?",
        array($login, $name, $surname, $pwd, $city, $email,
$phone));
    if ($this->db->affected_rows() === FALSE) {
        echo json_encode(array('status' => 'err', 'error' => 'DB error!
Last query: '.$this->db->last_query()));
        return;
        echo 'DB ERROR!';
        echo 'Last query: '.$this->db->last_query();
    } else {
        $this->session->token = $this->auth_model->sign_in($login, $pwd);
        echo json_encode(array('status' => 'OK'));
        return;
        header("Location: ".$base_url());
    }
}
}
}

```

## Файл Auth.php

```

<?php
class Auth extends CI_Controller {
    public function sign_in() {
        if (!isset($_POST['login_email']) or !isset($_POST['pwd'])) {
            header("Location: ".$base_url());
        }
        $this->load->model('auth_model');
        $this->load->library('session');
        $login_email = (string)$_POST['login_email'];
        $pwd = (string)$_POST['pwd'];
        $token = $this->auth_model->sign_in($login_email, $pwd);
        if ($token) {
            $this->session->set_userdata(
                array(
                    'token' => (string)$token
                )
            );
            echo json_encode(array('status' => 'OK'));
            return;
            header("Location: ".$base_url());
        }
        echo json_encode(array('status' => 'err', 'error' =>
"SIGN_IN_ERROR!"));
        return;
        echo "SIGN_IN_ERROR!";
    }
    public function sign_out() {
        $this->load->model('auth_model');
        $this->load->library('session');
        if (!isset($this->session->token)) {
            echo "ERR_EXIT";
        }
        $result = $this->auth_model->sign_out($this->session->token);
        if ($result) {
            $this->session->unset_userdata('token');
            header("Location: ".$base_url());
        }
    }
}

```

```

        echo "ERR_EXIT. NO SUCH TOKEN!";
    }
}

```

## **Файл Settings.php**

```

<?php
class Settings extends CI_Controller {
    public function index() {
        $this->load->library('session');
        $this->load->model('settings_model');
        $header = array();
        $data = array();
        $header['title'] = 'Настройки аккаунта';
        $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
        $header['user'] = $data['user'];
        if ($header['user'] != false)
            $header['user']['deliveries'] = isset($this->session->token)
? $this->auth_model->get_user_deliveries($this->session->token) : false;

        $this->load->view('templates/header', $header);
        $this->load->view('settings/index', $data);
        $this->load->view('templates/footer');
    }
    public function edit() {
        $this->load->library('session');
        $this->load->model('settings_model');
        $header = array();
        $data = array();
        $header['title'] = 'Редактирование аккаунта';
        $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
        $header['user'] = $data['user'];
        if ($header['user'] != false)
            $header['user']['deliveries'] = isset($this->session->token)
? $this->auth_model->get_user_deliveries($this->session->token) : false;
        if ($data['user'] != false and !empty($_POST) and
isset($_POST['name'], $_POST['phone'],
$_POST['city'], $_POST['address'], $_POST['surname'])) {
            $name = $_POST['name'];
            $surname = $_POST['surname'];
            $phone = $_POST['phone'];
            $city = $_POST['city'];
            $address = $_POST['address'];
            $data['err'] = array();
            if (strlen($name) < 2)
                $data['err']['name'] = "Неверное имя!";
            if (strlen($surname) < 2)
                $data['err']['surname'] = "Неверная фамилия!";
            if (strlen($city) < 3)
                $data['err']['city'] = "Некорректное название города!";
            if (strlen($phone) < 7)
                $data['err']['phone'] = "Некорректный номер телефона!";
            if (sizeof($data['err']) > 0) {
                $data['user']['name'] = $name;
                $data['user']['surname'] = $surname;
                $data['user']['email'] = $email;
                $data['user']['city'] = $city;
                $data['user']['phone'] = $phone;
            }
        }
    }
}

```



```

        $data['user']['address'] = $address;
        $this->load->view('templates/header', $data);
        $this->load->view('settings/edit', $data);
        $this->load->view('templates/footer');
        return;
    }
    $update_data = array(
        'name' => $name,
        'surname' => $surname,
        'city' => $city,
        'phone' => $phone,
        'address' => $address
    );
    $result = $this->settings_model->update_user($data['user']['id'],
$update_data);
    if ($result == true) {
        $data['user']['name'] = $name;
        $data['user']['surname'] = $surname;
        $data['user']['city'] = $city;
        $data['user']['phone'] = $phone;
        $data['user']['address'] = $address;
    }
}
$this->load->view('templates/header', $header);
$this->load->view('settings/edit', $data);
$this->load->view('templates/footer');
}
public function change() {
    $this->load->library('session');
    $this->load->model('settings_model');
    $header = array();
    $data = array();
    $header['title'] = 'Смена пароля';
    $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
    $header['user'] = $data['user'];
    if ($header['user'] != false)
        $header['user']['deliveries'] = isset($this->session->token)
? $this->auth_model->get_user_deliveries($this->session->token) : false;
    $this->load->view('templates/header', $header);
    $this->load->view('settings/change', $data);
    $this->load->view('templates/footer');
}
}
}

```

## **Файл Deliveries.php**

```

<?php
class Deliveries extends CI_Controller {
public function index() {
    $this->load->model('search_model');
    $this->load->library('session');
    $header = array();
    $data = array();
    $chosen_opts = array();
    $chosen_specials = array();
    $chosen_cls = 0;
    $offset = 0;
    $limit = 6;
}
}

```

```

if (isset($_GET['p'])) {
    $offset = (int)$_GET['p']-1;
}
if (isset($_GET['l'])) {
    $limit = (int)$_GET['l'];
}
if (isset($_GET['n']) && empty($_GET['n'])) {
    unset($_GET['n']);
}
$search_opts = $this->search_model->get_search_opts();
foreach ($search_opts as $opt) {
    $id = $opt['id'];

    if (isset($_GET['t'].$id)) {
        $chosen_opts[] = $id;
    }
}
$specials = array(
    'c-o', 'c-c', 'c-f',
    'c-n', 'c-hr', 'c-hp', 'c-a'
);
foreach ($specials as $special) {
    if (isset($_GET[$special])) {
        $chosen_specials[] = $special;
    }
}
if (isset($_GET['cls']) && !empty($_GET['cls']))
    $chosen_cls = (int)$_GET['cls'];

$delivery_name = isset($_GET['n']) && !empty($_GET['n']) ?
                    $_GET['n']
                    :
                    false;

if (!true) {
    echo '<pre>';
    echo 'Chosen opts:<br>';
    print_r($chosen_opts);
    echo '<br>Chosen specials:<br>';
    print_r($chosen_specials);
    echo '<br>Chosen class: '.$chosen_cls."<br>";
    echo 'Food name: '.$food_name."<br><br>";
    echo '$_GET: ';
    print_r($_GET);
    echo '</pre>';
}
if (isset($_GET['p']))
    unset($_GET['p']);
$search_str = '';
if (!empty($_GET)) {
    foreach ($_GET as $key => $value) {
        $search_str .= "$key=$value&";
    }
}
$search_str = substr($search_str, 0, -1);
}
if (!true) {
    echo '<pre>';
    echo $search_str;
    echo '</pre>';
}
$form_opts = isset($_GET) ? $_GET : false;

```

```

foreach ($chosen_specials as $special) {
    $form_opts[$special] = 1;
}

    $deliveries_count = $this->delivery_model->
>get_deliveries_count($chosen_opts, $chosen_specials, $chosen_cls,
$delivery_name);
    $max_pages = (int) ($deliveries_count/$limit + 1);
    $data['title'] = $header['title'] = 'Катеропии';
    $data['deliveries'] = $this->delivery_model->get_deliveries($limit,
$offset, $chosen_opts, $chosen_specials, $chosen_cls, $delivery_name);
    $data['search_opts'] = $this->search_model->get_search_opts();
    $data['user'] = isset($this->session->token) ? $this->auth_model->
>get_user($this->session->token) : false;
    $data['form_opts'] = $form_opts;
    $header['user'] = $data['user'];
    if ($header['user'] != false)
        $header['user']['deliveries'] = isset($this->session->token)
? $this->auth_model->get_user_deliveries($this->session->token) : false;
    $data['html_pagination'] = '<ul class="pagination">';
    $data['html_pagination'] .= ($offset > 0) ? '<li class="pagination-
item--wide first"><a class="pagination-link--wide first"
href="'.base_url().'.deliveries?'.$search_str.'"><i class="fa fa-angle-
left"></i><i class="fa fa-angle-left"></i></a></li>':'';
    $data['html_pagination'] .= ($offset > 0) ? '<li><a
class="pagination-link"
href="'.base_url().'.deliveries?p='.$offset.'&'.$search_str.'"><i class="fa fa-
angle-left"></i> Назад</a></li>':'';
    $data['html_pagination'] .= ($offset-1 > 0) ? '<li
class="pagination-item"><a class="pagination-link"
href="'.base_url().'.deliveries?p='.$(offset-1).'&'.$search_str.'">'.($offset-
1).'</a></li>':'';
    $data['html_pagination'] .= ($offset > 0) ? '<li class="pagination-
item"><a class="pagination-link"
href="'.base_url().'.deliveries?p='.$(offset).'&'.$search_str.'">'.($offset).'<
/a></li>':'';
    $data['html_pagination'] .= '<li class="pagination-item is-
active"><a class="pagination-link"
href="'.base_url().'.deliveries?p='.$(offset+1).'&'.$search_str.'">'.($offset+1
).'</a></li>';
    $data['html_pagination'] .= ($offset+1 < $max_pages) ? '<li
class="pagination-item"><a class="pagination-link"
href="'.base_url().'.deliveries?p='.$(offset+2).'&'.$search_str.'">'.($offset+2
).'</a></li>':'';
    $data['html_pagination'] .= ($offset+2 < $max_pages) ? '<li
class="pagination-item"><a class="pagination-link"
href="'.base_url().'.deliveries?p='.$(offset+3).'&'.$search_str.'">'.($offset+3
).'</a></li>':'';
    $data['html_pagination'] .= ($offset < $max_pages) ? '<li
class="pagination-item"><a class="pagination-link"
href="'.base_url().'.deliveries?p='.$(offset+2).'&'.$search_str.'">Вперед <i
class="fa fa-angle-right"></i></a></li>':'';
    $data['html_pagination'] .= ($offset+1 < $max_pages) ? '<li
class="pagination-item--wide last"><a class="pagination-link--wide last"
href="'.base_url().'.deliveries?p='.$max_pages.'&'.$search_str.'"><i class="fa
fa-angle-right"></i><i class="fa fa-angle-right"></i></a></li>':'';
    $data['html_pagination'] .= '</ul>';

    // $data['html_pagination'] = '<ul class="pagination">

```

```

        //      <li class="pagination-item--wide first"> <a
class="pagination-link--wide first" href="#"><i class="fa fa-angle-left" aria-
hidden="true"></i>&nbsp;&nbsp;&nbsp;Previous</a> </li>
        //      <li class="pagination-item"> <a class="pagination-link"
href="#">1</a> </li>
        //      <li class="pagination-item"> <a class="pagination-link"
href="#">2</a> </li>
        //      <li class="pagination-item is-active"> <a
class="pagination-link" href="#">3</a> </li>
        //      <li class="pagination-item"> <a class="pagination-link"
href="#">4</a> </li>
        //      <li class="pagination-item"> <a class="pagination-link"
href="#">5</a> </li>
        //      <li class="pagination-item--wide last"> <a
class="pagination-link--wide last" href="#">Next&nbsp;&nbsp;&nbsp;<i class="fa fa-
angle-right" aria-hidden="true"></i></a> </li>
        //      </ul>';
        $data['html_pagination'] = '';
        $this->load->view('templates/header', $header);
        $this->load->view('deliveries/index', $data);
        $this->load->view('templates/footer');
    }
    public function view($slug = NULL) {
        $this->load->library('session');
        $header = array();
        $data = array();
        if (is_numeric($slug)) {
            $data['delivery'] = $this->delivery_model-
>get_delivery((int)$slug);
        } else if (!empty($slug)) {
            $data['delivery'] = $this->delivery_model-
>get_delivery_by_slug((string)$slug);
        }
        if (empty($data['delivery'])) {
            show_404();
        }
        $header['title'] = $data['delivery']['name'];
        $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
        $data['foods'] = $this->delivery_model-
>get_delivery_foods((int)$data['delivery']['id']);
        $data['comments'] = $this->delivery_model-
>get_delivery_comments((int)$data['delivery']['id']);
        $data['categories'] = $this->delivery_model-
>get_delivery_categories((int)$data['delivery']['id']);
        $data['foods_categorised'] = $this->delivery_model-
>get_categorised_foods((int)$data['delivery']['id']);
        $header['user'] = $data['user'];
        if (!empty($_POST) && isset($_POST['msg']) &&
isset($_POST['r']) && $data['user'] != false) {
            $comment = (string)$_POST['msg'];
            $rating = (string)$_POST['r'];
            $result = $this->delivery_model-
>send_comment($data['user']['id'], $data['delivery']['id'], $comment,
$rating);
            if ($result) {
                $data['comment_result'] = '<h2 class="text-center"><text
class="text-success">Комментарий успешно добавлен!</text></h2>';
            }
        }
    }
}

```

```

        $data['foods'] = $this->delivery_model-
>get_delivery_foods((int)$data['delivery']['id']);
        $data['comments'] = $this->delivery_model-
>get_delivery_comments((int)$data['delivery']['id']);

        if (is_numeric($slug)) {
            $data['delivery'] = $this->delivery_model-
>get_delivery((int)$slug);
        } else if (!empty($slug)) {
            $data['delivery'] = $this->delivery_model-
>get_delivery_by_slug((string)$slug);
        }
    }
}
$this->load->view('templates/header', $header);
$this->load->view('deliveries/view', $data);
$this->load->view('templates/footer');
}
public function add() {
    $this->load->library('session');
    $data = array();
    $header['title'] = 'Добавление доставки';
    $data['title'] = 'Добавление доставки';
    $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
    $this->load->view('templates/header', $data);
    $json = array();
    $error = false;
    if (isset($_REQUEST['submit']) and $_REQUEST['submit']=='Добавить') {
        $name = (string) $_REQUEST['name'];
        $email = (string) $_REQUEST['email'];
        $phone = (string) $_REQUEST['phone'];
        $address = (string) $_REQUEST['address'];
        $data['name'] = $name;
        $data['email'] = $email;
        $data['address'] = $address;
        $data['phone'] = $phone;
        if (strlen($name) < 3)
            $data['err']['name'] = "Название слишком короткое!";
        if (!preg_match('/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/',$email))
            $data['err']['email'] = "Неверный формат почтового ящика!";
        if (strlen($address) < 3)
            $data['err']['address'] = "Некорректное название города!";
        if (strlen($phone) < 7)
            $data['err']['phone'] = "Некорректный номер телефона!";
        if (isset($data['err']) and sizeof($data['err']) > 0) {
            $this->load->view('add/fail', $data);
        }
        else {
            $query = $this->db->query("INSERT INTO delivery_queries SET
name='$name', email='$email', phone='$phone', address='$address'");
            $this->load->view('add/success', $data);
        }
    }
    else {
        $this->load->view('add/delivery', $data);
    }
    $this->load->view('templates/footer');
}
}

```

## Файл Ajax.php

```

<?php
class Ajax extends CI_Controller {
    public function delivery_search() {
        $this->load->model('search_model');
        $chosen_opts = array();
        $chosen_specials = array();
        $search_opts = $this->search_model->get_search_opts();
        foreach ($search_opts as $opt) {
            $id = $opt['id'];

            if (isset($_GET['delivery-type-'].$id)) {
                $chosen_opts[] = $id;
            }
        }
        $specials = array(
            'chk-online', 'chk-courier', 'chk-fast',
            'chk-new', 'chk-high-rate', 'chk-high-pop'
        );
        foreach ($specials as $special) {
            if (isset($_GET[$special])) {
                $chosen_specials[] = $special;
            }
        }
        $delivery_name = isset($_GET['delivery-name-search']) &&
!empty($_GET['delivery-name-search']) ?
            $_GET['delivery-name-search']
            :
            false;

        echo '<pre>';
        echo 'Chosen opts:<br>';
        print_r($chosen_opts);
        echo 'Chosen specials:<br>';
        print_r($chosen_specials);
        echo 'Delivery name: '.$delivery_name;
        echo '</pre>';
    }
}

```

## Файл Food.php

```

<?php
class Foods extends CI_Controller {
    public function index() {
        $this->load->model('search_model');
        $this->load->model('food_model');
        $this->load->library('session');
        $header = array();
        $data = array();
        $chosen_opts = array();
        $chosen_specials = array();
        $offset = 0;
        $limit = 10;
        if (isset($_GET['p'])) {
            $offset = (int)$_GET['p']-1;
        }
        if (isset($_GET['l'])) {
            $limit = (int)$_GET['l'];
        }
    }
}

```

```

if (isset($_GET['fn']) && empty($_GET['fn'])) {
    unset($_GET['fn']);
}
$search_opts = $this->search_model->get_search_opts();
foreach ($search_opts as $opt) {
    $id = $opt['id'];

    if (isset($_GET['t'].$id)) {
        $chosen_opts[] = $id;
    }
}
$specials = array(
    'c-o', 'c-c', 'c-f',
    'c-n', 'c-hr', 'c-hp'
);
foreach ($specials as $special) {
    if (isset($_GET[$special])) {
        $chosen_specials[] = $special;
    }
}

$food_name = isset($_GET['fn']) && !empty($_GET['fn']) ?
    $_GET['fn']
    :
    false;

if (!true) {
    echo '<pre>';
    echo 'Chosen opts:<br>';
    print_r($chosen_opts);
    echo 'Chosen specials:<br>';
    print_r($chosen_specials);
    echo 'Food name: '.$food_name;
    echo '$_GET: ';
    print_r($_GET);
    echo '</pre>';
}
if (isset($_GET['p']))
    unset($_GET['p']);
$search_str = '';
if (!empty($_GET)) {
    foreach ($_GET as $key => $value) {
$search_str .= "$key=$value&";
    }
    $search_str = substr($search_str, 0, -1);
}
if (!true) {
    echo '<pre>';
    echo $search_str;
    echo '</pre>';
}

$food_count = $this->food_model->get_foods_count($chosen_opts,
$searched_specials, $food_name);
$max_pages = (int) ($food_count/$limit + 1);
$header['title'] = 'Еда';
$data['foods'] = $this->food_model->get_foods($limit, $offset,
true, true, true, $chosen_opts, $chosen_specials, $food_name);
$data['search_opts'] = $this->search_model->get_search_opts();
$data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
$data['form_opts'] = isset($_GET) ? $_GET : false;

```

```

$header['user'] = $data['user'];
if ($header['user'] != false)
    $header['user']['deliveries'] = isset($this->session->token)
? $this->auth_model->get_user_deliveries($this->session->token) : false;
    $data['html_pagination'] = '<ul class="pagination">';
    $data['html_pagination'] .= ($offset > 0) ? '<li><a
href="'.base_url().'foods?'.$search_str.'"><i class="fa fa-angle-left"></i><i
class="fa fa-angle-left"></i></a></li>':'';
    $data['html_pagination'] .= ($offset > 0) ? '<li><a
href="'.base_url().'foods?p='.$offset.'&'.$search_str.'"><i class="fa fa-
angle-left"></i> Hazan</a></li>':'';
    $data['html_pagination'] .= ($offset-1 > 0) ? '<li><a
href="'.base_url().'foods?p='.$(offset-1).'&'.$search_str.'">'.(offset-
1).'</a></li>':'';
    $data['html_pagination'] .= ($offset > 0) ? '<li><a
href="'.base_url().'foods?p='.$(offset).'&'.$search_str.'">'.(offset).'</a></
li>':'';
    $data['html_pagination'] .= '<li class="active"><a
href="'.base_url().'foods?p='.$(offset+1).'&'.$search_str.'">'.(offset+1).'</
a></li>';
    $data['html_pagination'] .= ($offset+1 < $max_pages) ? '<li><a
href="'.base_url().'foods?p='.$(offset+2).'&'.$search_str.'">'.(offset+2).'</
a></li>':'';
    $data['html_pagination'] .= ($offset+2 < $max_pages) ? '<li><a
href="'.base_url().'foods?p='.$(offset+3).'&'.$search_str.'">'.(offset+3).'</
a></li>':'';
    $data['html_pagination'] .= ($offset < $max_pages) ? '<li><a
href="'.base_url().'foods?p='.$(offset+2).'&'.$search_str.'">Вперед <i
class="fa fa-angle-right"></i></a></li>':'';
    $data['html_pagination'] .= ($offset+1 < $max_pages) ? '<li><a
href="'.base_url().'foods?p='.$max_pages.'&'.$search_str.'"><i class="fa fa-
angle-right"></i><i class="fa fa-angle-right"></i></a></li>':'';
    $data['html_pagination'] .= '</ul>';
    $this->load->view('templates/header', $header);
    $this->load->view('foods/index', $data);
    $this->load->view('templates/footer');
}
public function view($id) {
    $this->load->library('session');
    $this->load->model('food_model');
    $header = array();
    $data['food'] = $this->food_model->get_food((int)$id);
    if (empty($data['food'])) {
        show_404();
    }
    $data['title'] = $data['food']['name'];
    $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
    $data['comments'] = $this->food_model-
>get_food_comments((int)$data['food']['id']);
    $header['user'] = $data['user'];
    $this->load->view('templates/header', $header);
    $this->load->view('foods/view', $data);
    $this->load->view('templates/footer');
}
}

```



**Файл Cart.php**

```

<?php
class Cart extends CI_Controller {
    public function index() {
        $this->load->model('cart_model');
        $this->load->library('session');
        $data = array();

        $header['title'] = 'Корзина';
        $data['title'] = 'Корзина';
        $data['foods'] = isset($_COOKIE['cart']) ? $this->cart_model-
>get_cart_foods($_COOKIE['cart']) : array();
        $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
        $data['delivery'] = isset($_COOKIE['cart']) ? $this->cart_model-
>get_cart_delivery($_COOKIE['cart']) : array();
        $data['total_price'] = 0;
        foreach ($data['foods'] as $food) {
            $data['total_price'] += $food['price'] * $food['amount'];
        }
        $this->load->view('templates/header', $data);
        $this->load->view('cart/index', $data);
        $this->load->view('templates/footer');
    }
    public function success() {
        $this->load->model('cart_model');
        $this->load->library('session');
        $data = array();
        $data['title'] = 'Заказ успешен';
        $data['foods'] = isset($_COOKIE['cart']) ? $this->cart_model-
>get_cart_foods($_COOKIE['cart']) : array();
        $data['user'] = isset($this->session->token) ? $this->auth_model-
>get_user($this->session->token) : false;
        $data['delivery'] = isset($_COOKIE['cart']) ? $this->cart_model-
>get_cart_delivery($_COOKIE['cart']) : array();
        $this->load->view('templates/header', $data);
        $this->load->view('cart/success', $data);
        $this->load->view('templates/footer');
    }
}

```

## ДОДАТОК Б

### Електронна презентація

Східноукраїнський національний університет ім. В.  
Дала  
Кафедра комп'ютерних технологій та електроніки  
Напрямок підготовки: 6.050102 "Комп'ютерна  
інженерія"

## Дипломний проект бакалавра на тему: "Система он-лайн замовлення їжі"

Студент: Михайлова Аліса Олександрівна  
Керівник проекту: Щербакова Марина Євгенівна

Северодонецьк – 2018 рік

Рисунок Б.1 – Слайд «Титульний слайд»

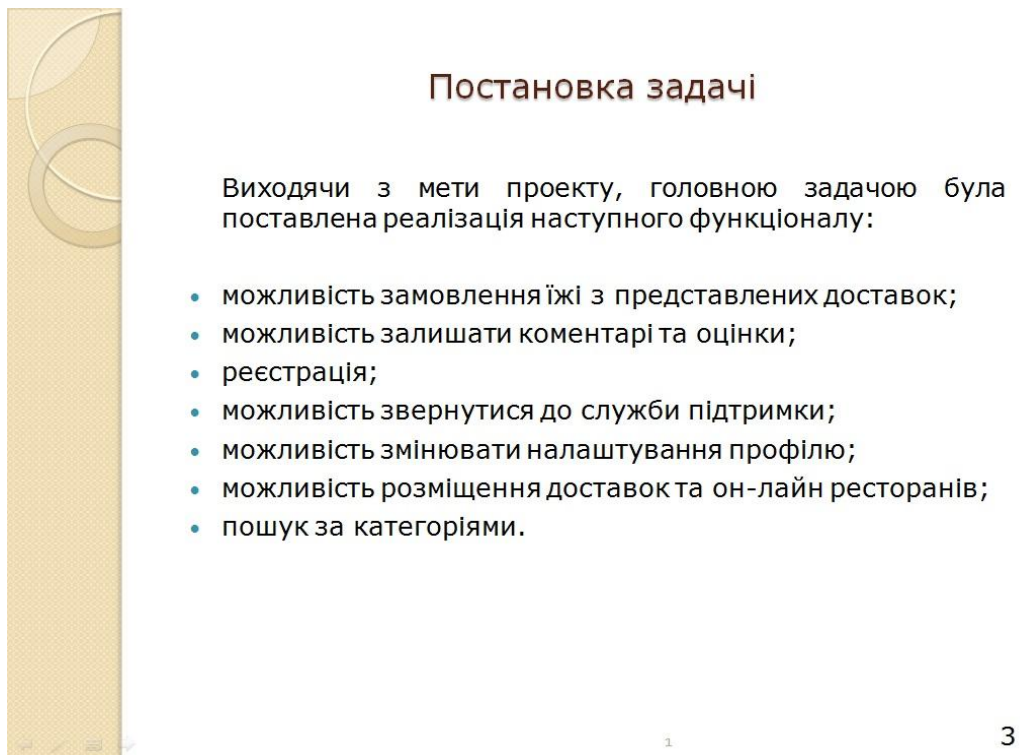
### Актуальність і мета проекту

**Актуальність.** Мережа Інтернет проникає у всі сфери людського життя. Кількість сервісів, які допомагають користувачам економити час, зростає з кожним роком. Також, вони нерозривно пов'язані з рекламою, яка є важливим засобом просування. Одна із сфер, в якій з'являються подібні інновації – споживання їжі. Це спричинило появу ринку foodtech.

Однією з галузей даного ринку є агрегатори доставки їжі – сервіси, що поєднують у собі різноманітні доставки і онлайн-ресторани. Загальний обсяг світового ринку foodtech-сервісів, зокрема – сервісів замовлення їжі онлайн, становить \$ 324 млрд щорічно.

**Мета проекту:** аналіз ринку foodtech та визначення методики створення сервісу замовлення їжі, а також подальша розробка сервісу.

Рисунок Б.2 – Слайд «Актуальність і мета проекту»



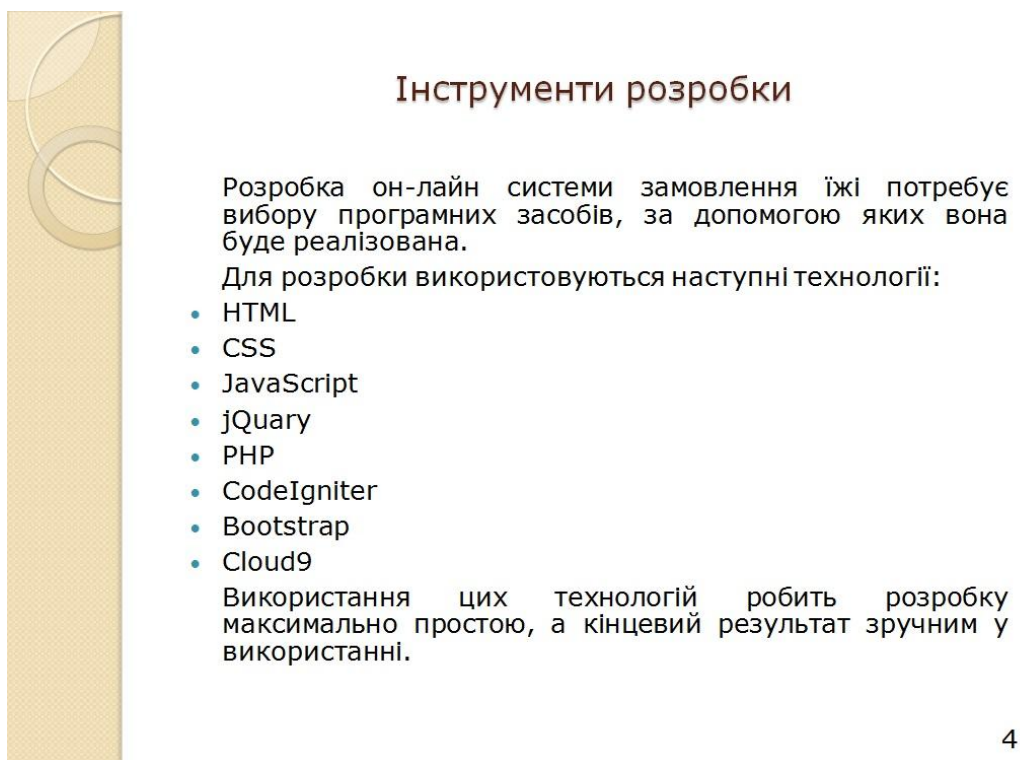
## Постановка задачі

Виходячи з мети проекту, головною задачею була поставлена реалізація наступного функціоналу:

- можливість замовлення їжі з представлених доставок;
- можливість залишати коментарі та оцінки;
- реєстрація;
- можливість звернутися до служби підтримки;
- можливість змінювати налаштування профілю;
- можливість розміщення доставок та он-лайн ресторанів;
- пошук за категоріями.

1 3

Рисунок Б.3 – Слайд «Постановка задачі»



## Інструменти розробки

Розробка он-лайн системи замовлення їжі потребує вибору програмних засобів, за допомогою яких вона буде реалізована.

Для розробки використовуються наступні технології:

- HTML
- CSS
- JavaScript
- jQuery
- PHP
- CodeIgniter
- Bootstrap
- Cloud9

Використання цих технологій робить розробку максимально простою, а кінцевий результат зручним у використанні.

4

Рисунок Б.4 – Слайд «Інструменти розробки»

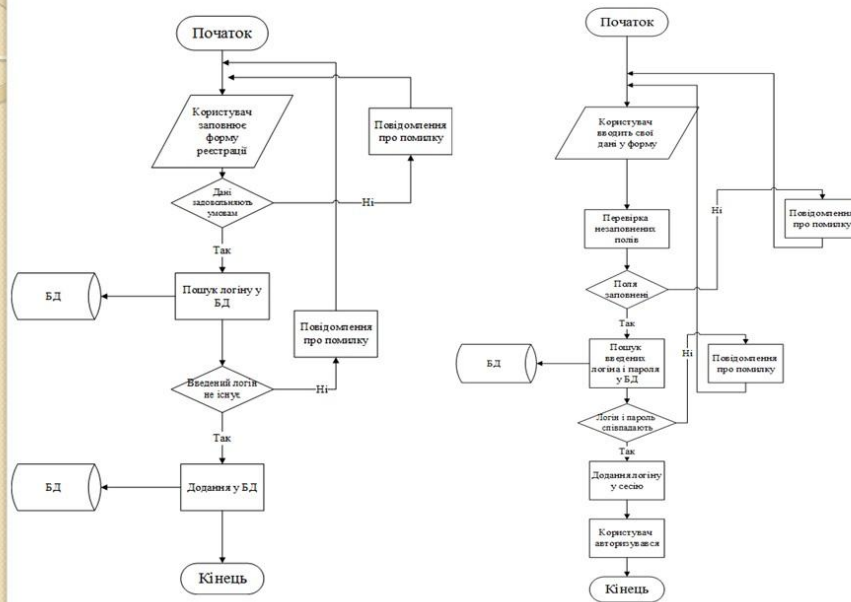
### Структура системи



5

Рисунок Б.5 – Слайд «Структура системи»

### Алгоритми реєстрації та авторизації



6

Рисунок Б.6 – Слайд «Алгоритми реєстрації та авторизації»

### Алгоритми додавання доставок та коментарів

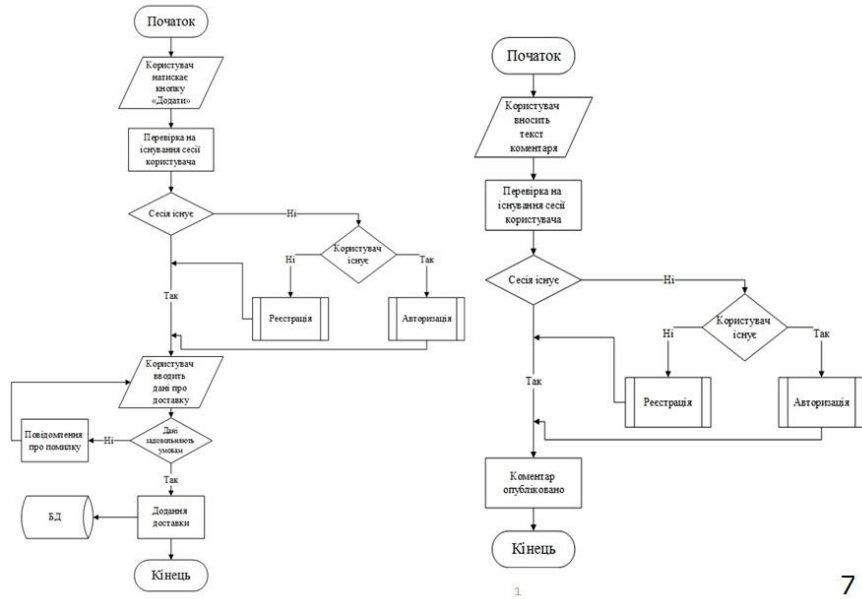


Рисунок Б.7 – Слайд «Алгоритми додавання доставок та коментарів»

### Алгоритми роботи кошика та оформлення замовлення

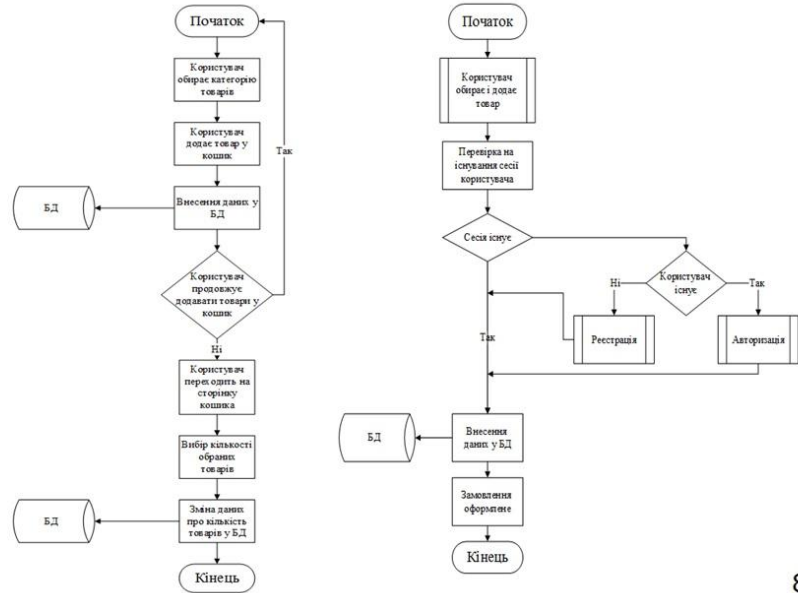


Рисунок Б.8 – Слайд «Алгоритми роботи кошика та оформлення замовлення»

## Вікно реєстрації та авторизації

1

9

Рисунок Б.9 – Слайд «Вікно реєстрації та авторизації»

## Вікно додавання коментарів

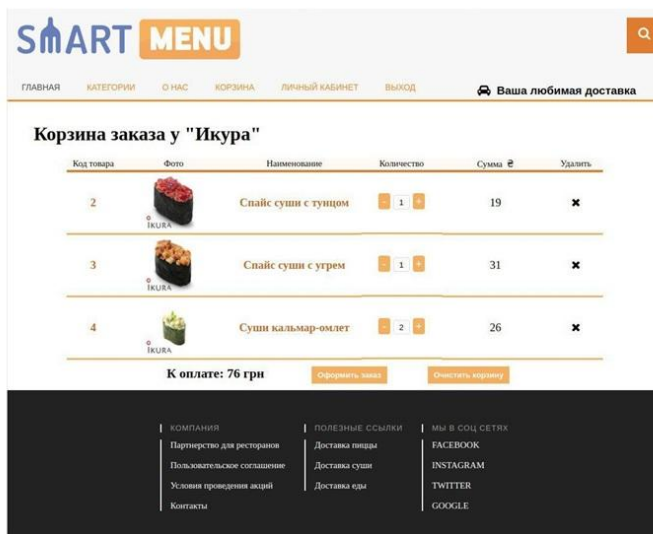
1

10

Рисунок Б.10 – Слайд «Вікно додавання коментарів»



## Сторінка "Кошик"



1

11

Рисунок Б.11 – Слайд «Сторінка «Кошик»»

## Результати роботи

У результаті виконання дипломного проекту були виконані наступні задачі:

- проектування структури сервісу;
- розробка структури сервісу;
- розробка дизайну;
- розробка вікна реєстрації;
- розробка сторінок авторизації та особистого кабінету;
- розробка сторінок категорій та доставки;
- розробка кошику;
- розробка сторінки оформлення замовлення;
- розробка вікна коментарів.

Кінцевим результатом роботи стала придатна до використання система он-лайн замовлення їжі.

1

12

Рисунок Б.12 – Слайд «Результати роботи»

## Висновки

- Мета роботи, поставлена на початковому етапі, була досягнута за допомогою обраних та проаналізованих програмних інструментів. Було розроблено справно працюючий сервіс для замовлення їжі он-лайн.
- Обрана галузь розробок є перспективною та прибутковою, тож у розробки є шанси стати затребуваною.

1

13

Рисунок Б.13 – Слайд «Висновки»