

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ ІМЕНІ ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ

Пояснювальна записка

до дипломної роботи

бакалавр

(освітньо-кваліфікаційний рівень)

на тему «Розробка веб-додатку для торгової організації»

Виконала: студентка 4 курсу, групи ІТз-151
напряму підготовки 6.040302 „Інформатика”

_____ Андрєєва А.В.
(підпис)

Керівник,
професор, д.т.н. _____ Марченко Д.М.
(підпис)

Рецензент,
доцент, к.т.н. _____ Фесенко Т.М.
(підпис)

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ

ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет інформаційних технологій та електроніки

Кафедра програмування та математики

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 6.040302 „Інформатика”

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМ,

д.т.н., доцент

_____Лифар В.О.

«___»_____2019 р.

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
АНДРЕЄВОЇ АНАСТАСІЇ ВІКТОРІВНИ

1. Тема роботи Розробка веб-додатку для торгової організації.
керівник роботи Марченко Д.М.

затверджені наказом вищого навчального закладу від “23” квітня 2019 року № 68/14.04

2. Строк подання студентом роботи 7 травня 2019 р.

3. Вихідні дані до роботи

Об'єктом даної роботи є розробка веб-додатка для торгової організації.

3.1 Літературні джерела:

Гутманс Е. РНР 5. Професійне програмування : Гутманс Е. — М:
2006. — 1418 с.

Давид Скляр. РНР. Рецепты программирования. 3-е изд.: Скляр Д.
— М: 2017. — 369 с.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Вступ

4.2 Аналітичний огляд:

Аналіз предметної області.

- Аналіз проектування веб-ресурсу.
- 4.3 Области використання сайтів створених по різних технологіям.
- 4.4 Розробка програмного забезпечення
- 4.5 Тестування веб-сайту
- 4.6 Висновок
- 4.7 Перелік використаних джерел
5. Перелік графічного матеріалу немає
6. Дата видачі завдання 4 лютого 2019 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Одержання завдання на виконання роботи	04.02.19	
2	Укладання і погодження з керівником план етапів виконання роботи	20.02.19	
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	1.03.19	
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	10.03.19	
3	Проектування інфологічної моделі задачі що реалізується.	01.04.19	
5	Укладання та тестування програмного продукту	10.05.19	
6	Укладання, оформлення та погодження пояснювальної записки з керівником	15.05.19	
7	Укладання, оформлення та погодження з консультантом розділу «Охорона праці»	20.05.19	
7	Здача готової пояснювальної записки на кафедрі	13.06.19	
8	Укладання доповіді і презентації	14.06.19	

Студент

_____ (підпис)

Керівник роботи

_____ (підпис)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ
дипломної роботи студентки гр. ІТз-151 Андрєєвої А.В.

Науковий керівник:

Професор, д.т.н. _____

Марченко Д.М.

Оцінка наукового керівника: _____

Рецензент: Фесенко Т.М., доцент каф. ПМ СНУ ім. В. Даля

ПІБ, місто роботи, посада

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК

Зав. Кафедри ПМ

д.т.н., доцент

Лифар В.О.

(підпис)

РЕФЕРАТ

Текст – 63., рис. – 21, табл. – 5, додатків – 2, літературних джерел – 24

Метою роботи – є розробка веб-додатку для торгової організації, який допоможе організації вийти на новий рівень бізнесу.

За допомогою веб-технологій такі як PHP – для розробки серверної частини сайту, HTML 5 – в якості базової розмітки та JavaScript – для оформлення вебресурсу і його адаптивності на різних пристроях, було розроблено веб-додаток. На підставі виконаної роботи, можна зробити висновок про те, що поставлені до даного проекту вимоги виконані.

САЙТ. СТВОРЕННЯ САЙТУ. ТЕСТУВАННЯ САЙТУ. ЗАМОВНИК.
ЗАМОВЛЕННЯ

ЗМІСТ

ВСТУП.....	7
1 АНАЛІТИЧНИЙ ОГЛЯД	9
1.1 Аналіз предметної області	9
1.2 Аналіз етапів проектування веб-ресурсу	9
1.3 Аналіз методів розв'язання задачі	13
1.4 Аналіз існуючих варіантів розв'язання досліджуваної задачі.....	14
1.5 Постановка задач розробки.....	16
1.6 Техніко-економічне обґрунтування доцільності розробки	17
1.7 Оцінка конкурентоспроможності інноваційного рішення.....	19
2 ОБЛАСТІ ВИКОРИСТАННЯ САЙТІВ СТВОРЕНИХ ПО РІЗНИМ ТЕХНОЛОГІЯМ	22
2.1 Особистий сайт.....	24
2.2 Корпоративні сайти та іміджеві портали	26
2.3 Односторінкові сайти	27
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	29
3.1 Варіантний аналіз і обґрунтування вибору мови програмування.....	29
3.2 Варіантний аналіз і обґрунтування вибору середовища розробки	32
3.3 Розробка модулів програми	36
4 ТЕСТУВАННЯ ВЕБ-САЙТУ	45
4.1 Опис методик, що використовуються при тестуванні веб-сайту.....	45
4.2 Тестування програми	46
ВИСНОВКИ.....	52
СПИСОК ЛІТЕРАТУРИ.....	53
ДОДАТОК А.....	55
ДОДАТОК Б	60

ВСТУП

Актуальність роботи.

Використання веб-технологій помітно зростає в усіх сферах людського життя, зокрема це стосується сфери освіти. Однією з головних причин посиленої уваги педагогів до проблеми упровадження вебтехнологій є зручність та простота використання наявних інструментів для пошуку, створення та використання веб-ресурсів (ВР).

Використовуючи ВР, можна суттєво підвищити ефективність навчального процесу, активізувати навчально-пізнавальну та самостійну діяльність учнів. Впровадження веб-технологій у навчальний процес та підготовка учителя до їх використання розглядається у ряді праць українських та закордонних авторів. Ці питання є предметом серйозного обговорення на науково-практичних та науково-методичних конференціях, а також на сторінках періодичних видань.

Візуалізація навчальної інформації у різних своїх формах дозволяє вирішити цілий ряд педагогічних завдань: забезпечення інтенсифікації навчання, активізації навчальної та пізнавальної діяльності, формування і розвиток критичного і візуального мислення, зорового сприйняття, образного представлення знань і навчальних дій, передачі знань і розпізнавання образів, підвищення візуальної грамотності та візуальної культури.

Зв'язок роботи з науковими програмами, планами, темами.

Головна задача – покращення візуального сприйняття інформації учнями молодших класів.

Для досягнення мети розв'язано такі задачі:

- проведено аналіз типів веб-ресурсів та методів і засобів їх створення;
- розроблено інтерфейс веб-ресурсу;
- розроблено структурний елемент інтерфейсу веб-ресурсу;
- розроблено інформаційне наповнення веб-ресурсу.

Об'єктом дослідження є процес покращення візуального сприйняття інформації. Предметом дослідження роботи є методи і засоби покращення процесу візуального сприйняття інформації. Наукова новизна магістерської кваліфікаційної роботи полягає у тому, що зазнав подальшого розвитку засіб покращення візуального сприйняття інформації спрямований на учнів молодших класів, з застосуванням сучасних веб-технологій; уточнено поняття «веб-ресурс» та подано його класифікацію.

Практичне значення одержаних результатів полягає у наступному:

- розроблено та наведено класифікацію веб-ресурсів;
- розроблено веб-ресурс для покращення візуального сприйняття інформації учнями молодших класів.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Аналіз предметної області

Веб-ресурси – це електронні ресурси, які розміщені у “веб-просторі” локальної чи глобальної мережі у вигляді різних форматів (текстового, графічного, архівного, аудіо та відеоформатів та ін.). ВР можна класифікувати за різними критеріями(рис 1.1).

Використання ВР ефективно впливає на процес навчання та засвоєння навчального матеріалу, сприяє формуванню навичок самостійної діяльності, дає змогу організувати індивідуальну та групову роботу учнів [2].

1.2 Аналіз етапів проектування веб-ресурсу

Розробка веб-ресурсу – це процес створення веб-сторінок або сайтів. Веб-сторінки створюються з використанням HTML, CSS і JavaScript і т.д. Сторінки можуть містити простий текст і графіку, нагадуючи собою статичний документ. Сторінки також можуть бути інтерактивними або відображати інформацію, що змінюється. Створювати інтерактивні сторінки складніше, але вони дозволяють створювати веб-сайти з більш наповненим вмістом. Сьогодні більшість сторінок інтерактивні і надають сучасні інтерактивні послуги, такі як корзини Internet-магазинів, динамічна візуалізація та навіть складні соціальні мережі [3].

Створенню конкретного веб-ресурсу передують детальний комплексний аналіз, визначаючий критерії, яким він має відповідати.

Процес створення включає шість основних етапів:

- визначення цілей та задач;
- розробка структури;
- розробка дизайн-макетів;
- HTML-верстка;
- програмування та контроль якості;
- запуск та оптимізація.

Кожен з перелічених етапів самодостатній, що дозволяє обирати схему роботи і виконавця для кожного з них окремо. Розглянемо кожний етап більш детально.

На етапі проектування формуються бізнес-цілі проекту, що створюється, визначаються вимоги, яким він повинен відповідати, розробляється загальна концепція. Під час роботи на цьому етапі уточнюються вимоги замовника, формується технічне завдання, виконується аналіз цільової аудиторії.

Для більш глибокого аналізу можна запросити у замовника відповідні матеріали: брошури, щорічні звіти, зразки продукції, інші супутні дані - все, що допоможе скласти уявлення про те, хто і з якою метою буде відвідувати сайт, які завдання будуть виконуватися на сайті.

Важливо з'ясувати технічні можливості майбутньої основної користувацької аудиторії - пропускну спроможність каналів зв'язку, які використовуються Internet-браузери і т.д.

Для знаходження цільової аудиторії доцільно увійти в роль кваліфікованого користувача на аналогічно створюваних Internet-ресурсах. Це допоможе виявити нові креативні концепції для того, щоб сайт був більш конкурентоспроможним і «не загубився» серед безлічі інших.

Коли цілі визначені, приступають до складання розширеного плану проекту, що відображає скільки часу, грошей та інших коштів знадобиться для виконання робіт на кожному з наступних етапів.

Такий план часто містить інформацію про бюджет проекту, графік робіт (з відповідним розподілом ролей між веб-розробниками), технічну документацію, а також розділ «деталей і уточнень», де обумовлені конкретні аспекти можливих спірних питань. У цей розділ також включають пропозиції по готових розробках і шаблонах.

Наступний етап – розробка структури. Включає в себе зміст сторінки в якій буде розміщено веб-ресурс, а також – інформаційну стратегію, яка визначає, як організована подача інформації, щоб майбутні користувачі могли легко та зручно використовувати надані ресурси.

Головною задачею цього етапу розробки є створення карти, яка відображає зв'язок сторінок та їх найбільш значущі функціональні можливості. Її подають у вигляді блок-схеми, на якій кожна сторінка відображається окремим прямокутником, зв'язки між ними означають переходи між сторінками.

Дизайн-макет – це графічне, наглядне зображення елементів сайту. Дизайн-макет повністю втілює візуальну концепцію сайту. Його розробка виконується в одній з графічних програм (у переважній більшості випадків - в Adobe Photoshop).

У процесі розробки дизайнер керується письмовою угодою (брифом) на створення дизайн-макету, який заповнюється замовником і містить побажання до дизайну: тип, кольорова гама, наявність тих чи інших графічних елементів, тощо. На цій стадії створюються всі елементи веб-дизайну відповідно до стилю подачі інформації та загальної концепції.

Головним при дизайні сайту є вміння розробити графічні об'єкти, які б швидко завантажувалися і добре виглядали, незалежно від використовуваного Internet-браузера.

Часто вдаються до використання готових дизайн-шаблонів, які широко представлені в мережі Internet або є вбудованими в різні графічні редактори, такі як, Microsoft FrontPage або Adobe Photoshop. За допомогою подібних шаблонів сайт створюється за максимально стислий час. Однак слід зазначити, що у такого рішення є ряд істотних недоліків, головний з яких - повторюваність і не унікальність дизайну.

Шаблон є оболонкою з мінімальною кількістю інтерактивних елементів і корисних модулів. Тому при виборі шаблону варто звертати увагу не тільки на дизайн, але і на функціональність. Важливим елементом дизайну є графіка, яку умовно розділяють на три категорії:

- ілюстративна графіка – пояснювальні зображення, схеми та графіки, фото;
- функціональна графіка – кнопки навігації, лічильники та інші елементи управління;
- декоративна графіка – естетичні елементи дизайну, такі як

фоновий малюнок, заголовки, баннери, рамки.

Така класифікація передбачає чітке розмежування форматів і передбачає використання певних категорій по максимуму (наприклад, без використання функціональної графіки навігація буде не зручною в той час як без декоративного оформлення сайт буде більш швидко завантажуватися і буде більш простим для сприйняття, не буде перевантажений графікою).

HTML-верстка макета є наступним кроком після розробки сайту. Верстка - це перетворення створених дизайнером графічних макетів сторінок в HTML-код, який би відображався в Internet-браузері в точній відповідності до вихідного макету. Складність верстки залежить від складності дизайну. Основними завданнями при верстці є:

- коректність відображення сторінок сайту при різних розширеннях екрану;
- кросбраузерність – однакове відображення сторінок сайту в найбільш популярних браузерах - Internet Explorer, Mozilla Firefox, Opera, Chrome.

Програмування - це практична реалізація проекту, інтеграція напрацювань за окремими напрямками. Іншими словами, це процес побудови функціональних інструментів для наповнення і обробки даних. Програмування визначає наскільки стабільним і захищеним буде функціонування сайту. Вибір платформи, технологій і грамотного підходу до програмування відіграє істотну роль.

На даному етапі важливо визначитися з підходом до створення веб-ресурсу: чи буде він статичним або динамічним. Створення веб-ресурсу, як і будь-якого іншого програмного продукту, зіштовхується з проблемою постійної зміни даних та файлів.

Контроль за змінами, що вносяться в проект, допомагають забезпечити системи управління версіями (Version Control System - VCS), які зберігають попередні версії вихідних файлів проекту, відстежують вироблені в файлах зміни, забезпечують спільну командну роботу над проектом. До найбільш популярних на поточний момент VCS відносяться : SVN, GIT, Microsoft VSS. Використання

системи контролю версій піднімає загальний рівень якості розробки.

По завершенні етапу активного програмування починається етап тестування коректності функціонування створеного веб-ресурсу: перевірки на наявність граматичних помилок, пропущених картинок, непрацюючих посилань і т.д., а також перевірки функціонування сайту в різних веб-браузерах

Розробка веб-ресурсу - це комплексний багатокроковий процес, що вимагає знання безлічі різних технологій і мов програмування, вміння працювати з базами даних, використовувати безліч інструментальних засобів і програмних пакетів.

Підсумовуючи все вищесказане, слід при створенні веб-ресурсу пройти всі перелічені етапи та визначити задачі, параметри та цільову аудиторію.

1.3 Аналіз методів розв'язання задачі

Всі методи розробки веб-ресурсів можна умовно розділити на 2 основні групи. Перша група методів розробки – це методи авторського написання сайтів на одній або декількох мовах веб-програмування. При цьому робота може здійснюватися як у простих (текстових), так і візуальних редакторах HTML та CSS. Останні дозволяють створювати сайти в режимі WYSIWYG -

«Що Бачу Те й Одержую».

У випадку статичного сайту цілком достатнім для авторського написання буде використання «зв'язування» HTML і CSS, з можливим включенням Javascript. Для створення ж динамічного сайту не обійтися без серверних скриптів, таких як PHP, ASP.NET і т.д.

Створювати й редагувати вручну файли .php можна навіть у звичайному «Блокноті», що поставляється з ОС Windows. Для роботи з ASP.NET потрібно додатково встановити програмний продукт Microsoft Visual Studio.

При використанні авторських методів розробки веб-ресурсу дизайн сайту (графічне оформлення) також створюється вручну. Для цих цілей застосовуються будь-які графічні редактори за бажанням. Вручну можна відредагувати й уже готові шаблони дизайну, як платні так і безкоштовні.

Друга група методів розробки веб-ресурсів містить у собі методи

автоматизованого створення сайтів: за допомогою спеціальних конструкторів сайтів або ж систем керування контентом (CMS).

Конструктори сайтів - це, як правило, онлайн-системи, що дозволяють із готового типового набору модулів і компонентів «зібрати» сайт і відразу ж розмістити його в веб. Одні з найбільш популярних конструкторів сайтів - це системи ucoz.ru, narod.ru, sites.google.com.

Популярні блог-сервіси, такі як LiveJournal або LiveInternet, також є своєрідними конструкторами сайтів-блогів. Однак можливості цих конструкторів досить обмежені.

Методи розробки сайтів з використанням CMS - одні з найбільш популярних на сьогоднішній день. CMS, виражаючись умовно, являє собою певну готову візуальну й програмну оболонку, що користувач може заповнити необхідним контентом, а також за своїм бажанням змінити й настроїти.

Автоматизовані методи розробки сайтів передбачають поділ структури сайту на «дизайн» і «контент». Тоді легко можна змінювати контент, не зачіпаючи дизайну сайту або його програмного коду. При ручному створенні сайту поділу структури сайту на дві окремі «галузі» - дизайн і вміст - не відбувається.

Методи авторської розробки веб-ресурсів досить складні, адже вони вимагають значних пізнань в області веб-програмування або дизайну сайтів. Однак вони мають незаперечну перевагу: створюючи сайт вручну, завжди можна одержати саме те, що хочеш..

Створення сайтів на основі безкоштовних онлайн-конструкторів зручне для вебмайстрів-початківців, що бажають «випробувати свої сили». Переважно цей метод підходить для створення невеликих простих сайтів, наприклад, сайтів-візиток.

1.4 Аналіз існуючих варіантів розв'язання досліджуваної задачі.

Проведемо аналіз трьох найпопулярніших аналогів: 5plys, Teacher, Myclass.

Розглянуто веб-ресурс 5plys [4]. Дизайн сайту виконаний в світло синіх кольорах з темно-синім меню, що в свою чергу не повністю фокусує увагу користувача на інформації. Блоки з новинами розроблено у сірому кольорі. На головній сторінці користувачеві доступні для перегляду останні новини та вступна стаття. Меню складається з таких розділів: Головна, Задачі, Тести, Гороскоп, Корисні поради, Мій телефон, DVD, Форум.

Розглянуто веб-ресурс Teacher [5]. Сайт розроблено на самописній системі керування контентом. Дизайн сайту виконаний в блакитних кольорах. Меню складається з таких пунктів: Головна, Інформація про сайт, Презентації та відео матеріали, Вчитель вчителю, Вчитель учням, Різне, Форум, Зворотній зв'язок. На головній сторінці відображаються останні новини та наведений наочний перелік дипломів і сертифікатів сайту.

Розглянуто веб-ресурс MyClass [6]. Сайт розроблено на самописній системі керування контентом. Дизайн сайту виконано у вигляді сірих блоків з об'ємною тінню на нижній стороні блоку. Меню складається з таких пунктів: Головна, Літопис класу, Для моїх учнів, Для мами і тата, Кабінет вчителя, Освітні сайти, Книга для гостей. На головній сторінці є слайдер з 12 різними слайдами, які періодично змінюються.

Таблиця 1.1 – Порівняльна характеристика веб-ресурсів аналогів

Критерії	5plys	Teacher	Myclass
Форма авторизації	1	1	1
Анімований слайдер	0	0	1
Пошук	1	1	1
Багатомовність	0	0	0
Кросплатформеність	1	0	1
Оновлюваність контенту	0	1	1
Рейтингова система Користувачів	0	1	1

Сумарний коефіцієнт:	3	4	6
----------------------	---	---	---

1.5 Постановка задач розробки

Проаналізувавши предметну область та виконавши порівняльний аналіз аналогів, потрібно сформулювати задачі та детально розглянути завдання, вирішення яких представляє магістерська кваліфікаційна робота:

- проведення аналізу типів веб-ресурсів та методів і засобів їх створення;
- розробка інтерфейсу веб-ресурсу;
- розробка структурного елемента інтерфейсу веб-ресурсу;
- розробка інформаційного наповнення веб-ресурсу
- апробація та реалізація веб-ресурсу.

Перша представлена задача докладно розглянута вище. Рішення задачі надає загальну інформацію, визначає сильні та слабкі сторони існуючих рішень. Визначає актуальність рішення задачі новим шляхом, оскільки існуючі реалізації не є ідеальними. Представляються орієнтири для подальших етапів, формулюються вимоги до розробника. Тому вирішення цієї задачі стоїть на першому місці.

Наступна задача передбачає розробку на основі виділених даних оптимального інтерфейсу веб-ресурсу з точки зору його інноваційності та корисності на фоні інших продуктів. Вона має на меті поділити завдання на під-завдання, визначити модулі та зв'язки між ними, і на основі цього створити структуру інтерфейсу зручного для багатофункціонального використання в житті, його оптимального рішення.

Визначивши під-задачі, у третьому визначеному завданні, створюється структура всього веб-ресурсу.

Останній етап рішення завдання – тестування і реалізація результату, його впровадження в життя.

1.6 Техніко-економічне обґрунтування доцільності розробки

Аналіз ринку – це комплекс дій, спрямованих на вивчення всіх факторів, умов, ситуацій, які впливають на стан і розвиток ринку, на зміни його обсягів, структури, масштабів.

Аналіз ринку проводиться з метою одержання об'єктивної інформації про реальну ситуацію на ринку, виявлення та оцінки сили впливу основних чинників. У ході аналізу виявляють обсяги, структуру, широту і якість попиту, пропозиції за певний час та тенденції розвитку ринку, визначають ступінь збалансованості попиту і пропозиції, встановлюють відхилення між цими категоріями. Особливе значення надається аналізу цін як основному факторові, що формує попит населення. Оцінюють розмір і структуру незадоволеного попиту, ступінь масовості попиту (як відношення кількості споживачів товарів (послуг) до загальної чисельності ймовірних споживачів). Аналіз ринку дає оцінку обсягів і структури формування попиту, характеристику конкуруючих фірм, їх місце та перспективи на даному товарному чи регіональному ринку. Оцінюються сильні і слабкі місця конкурентів для розробки стратегії і тактики конкурентної боротьби з метою посилення своїх позицій на ринках [7].

Потенційними споживачами даної розробки виступають загальноосвітні навчальні заклади.

Аналіз попиту на нову продукцію – один з найважливіших напрямів діяльності інноваційних компаній. Комерційним компаніям немає сенсу вкладати свої кошти в науково-дослідні і дослідно-конструкторські роботи (ННДКР), якщо кінцевий результат себе не окупить [8]. Тому необхідним є розрахунок ємності ринку. Для цього необхідно спрогнозувати наступні дані:

- Середня кількість споживачів, які використовують товари аналогічні розроблюваному ($P=17337$) [9].
 - Середній відсоток споживачів, які зацікавлені придбанням інноваційного продукту ($P_P=30\%$);
 - Середній термін заміни інноваційного продукту ($T=3$ роки);

- Середній відсоток споживачів, що захочуть придбати розроблюваний товар повторно ($C_{\Pi}=20\%$).

Визначившись із вихідними даними, можна встановити:

- потребу в інноваційному продукті
- оптимістичний прогноз попиту на інноваційне рішення
- песимістичний прогноз попиту на інноваційне рішення
- реалістичний прогноз попиту на інноваційне рішення

Під якістю розуміють сукупність властивостей і характеристик продукції (послуг), що дають можливість задовольнити відповідні потреби. Міра придатності товару задовольнити певну потребу споживача оцінюється за допомогою спеціальних показників якості – кількісних характеристик однієї або кількох властивостей продукції за конкретних умов її створення або використання.

Оцінка рівня якості інноваційного рішення проводиться з метою порівняльного аналізу і визначення найбільш ефективного в технічному відношенні варіанта інженерного рішення.

Рівень якості – це кількісна характеристика міри придатності певного виду продукції для задоволення конкретного попиту на неї при порівнянні з відповідними базовими показниками за фіксованих умов споживання.

Під час оцінки якості продукції доцільно визначити абсолютний і відносний її рівні.

Одночасно визначають відносний рівень якості окремих видів продукції, що виробляється (проекується), порівнюючи її показники з абсолютними показниками якості найліпших вітчизняних та зарубіжних аналогів (товарів-конкурентів) (табл. 1.2). Для оцінки значень параметрів використовується 5-бальна шкала.

Таблиця 1.2 – Основні параметри інноваційного рішення та товару-конкурента

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
Функціональність	4	5	1,25	0,3
Вимоги до оперативної пам'яті	4	5	1,25	0,1
Стабільність роботи	3	4	1,33	0,1
Супровідна документація	3	3	1	0,03
Відповідність стандартам	4	5	1,25	0,03
Зручність роботи	4	5	1,25	0,25
Сумісність	5	5	1	0,05
Надійність	5	4	0,8	0,07
Захищеність	4	4	1	0,07

Крім того, в якості товару-конкурента необхідно прийняти вироби з найкращими показниками, які представлені на ринку подібної продукції, або запроектовані вироби по яким є відповідна інформація.

1.7 Оцінка конкурентоспроможності інноваційного рішення

У найширшому розумінні конкурентоспроможність товару – це можливість його успішного продажу на певному ринку і в певний проміжок часу. Водночас конкурентоспроможною можна вважати лише однорідну продукцію з технічними параметрами і техніко-економічними показниками,

що ідентичні аналогічним показникам уже проданого товару. Для того, щоб високоякісний товар був одночасно і конкурентоспроможним, він має відповідати критеріям оцінки споживачів конкретного ринку в конкретний час.

Дані для розрахунку загального показника конкурентоспроможності

інноваційного рішення занесено до таблиці 1.3. Для оцінки значень параметрів використовується 5-бальна шкала.

Таблиця 1.3 – Нормативні, технічні та економічні параметри інноваційного рішення і товару-конкурента

Показник	Варіанти		Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)	
Технічні показники:			
Стабільність роботи	3	4	0,07
Сумісність	5	5	0,06
Надійність	5	4	0,06
Споживацькі показники:			
Аудіо ресурси	3	5	0,07
Відео ресурси	2	5	0,07
Використання зображень	3	5	0,08
Наявність авторизації	2	4	0,05
Наявність календаря	1	4	0,05
Економічні показники			
Ціна	750	600	0,05
Експлуатаційні витрати	12000		0,03

Загальний показник конкурентоспроможності інноваційного рішення (К) з урахуванням вищезазначених груп показників можна визначити за формулою:

$$K = \frac{I_{m.n.}}{I_{e.n.}}, \quad (1)$$

де $I_{m.n.}$ – індекс технічних параметрів (відносний рівень якості

інноваційного рішення); $I_{e.n.}$ – індекс економічних параметрів.

Індекс економічних параметрів визначається за формулою:

$$I_{e.n.} = \frac{\sum_{i=1}^n P_{Hei}}{\sum_{i=1}^n P_{Bei}}, \quad (2)$$

де P_{Hei} , P_{Bei} – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів.

2 ОБЛАСТІ ВИКОРИСТАННЯ САЙТІВ СТВОРЕНИХ ПО РІЗНИМ ТЕХНОЛОГІЯМ

Не можна однозначно сказати що існує ідеальна технологія для створення сайтів, які призначені для різних цілей. Існує дуже багато різновидів, як з технологічної сторони так і з цілей для яких вони були створені. Порівнювати інтернет-магазин із особистим блогом або сайт-візитка із корпоративним немає можливості, так як вони повністю відрізняються своєю специфікацією та призначенням. Саме тому для кожного типу потрібно використовувати своєрідні методи розробки, в залежності від типу, якщо це інтернет магазин – основною характеристикою має бути швидкість обміну інформацією та стійкість до великих навантажень, особистий блог – зручність в користуванні та красивий дизайн.

Існує дана класифікація: Доступність ресурсу:

- Відкритий веб ресурс - всі розділи сайту повністю доступні для відвідувачів
- Напіввідкритий веб ресурс - розділи сайту доступні після реєстрації відвідувача
- Закритий веб ресурс - як правило, службовий сайт доступ до якого має обмежене число користувачів

Розміщення ресурсу:

- У мережі Інтернет - загальнодоступні
- У локальній мережі - локальні

сайти Технологія відображення

ресурсу:

- Статичні сайти - складаються з Html файлів і передаються користувачеві у вигляді, в якому зберігаються на сервері

- Динамічні сайти - як правило, працюють на основі системи управління сайтом (CMS). Вміст генерується, за допомогою різних скриптів використовуючи базу даних

Розглянемо детальніше типи сайтів:

1. Сайт-візитка

Саме той набір інформації, поширення якої в Інтернеті, безсумнівно, корисно як починаючим, так і акулам бізнесу. Це своєрідний довідник про фірму, який містить всі необхідні контактні дані та інформацію про діяльність компанії. Це повинна бути коротка, чітка і найважливіша інформація про Вас, Вашу сферу діяльності, інтереси. Не варто перевантажувати Ваш сайт-візитку інформацією, яка можливо не знадобиться потенційному клієнтові або партнерам.

Сайт-візитка – представництво компанії в мережі Інтернет. Надає загальну інформацію про власника (компанія, підприємець і тд). Як правило, складається з 5-7 сторінок, при цьому максимально описує вид діяльності. Головні розділи - про компанію, прайс-листи, контактні дані, схема проїзду.

2. Бізнес-сайт

Бізнес сайт -це каталог продукції компанії (товари або послуги). У каталозі присутній докладний опис товару, публікуються сертифікати й нагороди, відгуки клієнтів і звичайно технічні дані. Сайт даної специфікації незамінний для будь- якої компанії.

3. Промо-сайт

Промо сайт - веб сайт конкретної торгової марки (бренду). Створюється з метою реклами нового продукту або бренду в цілому. Розділи: акції, вікторини, голосування, ігри і тд.

4. Інтернет-магазин

Інтернет - магазин це каталог товарів з можливістю онлайн замовлення. Клієнт має можливість оплати товару, за допомогою різних сервісів використовуючи при цьому пластикову карту, рахунок або Інтернет гроші. Розділи: про магазин, допомога, каталог, кошик, сторінка замовлення,

КОНТАКТИ.

5. Корпоративний сайт

Корпоративний сайт - містить повну інформацію про компанію (послуги, товари і тд.). Як правило, має складну структуру і має розширену функціональність. Може бути відкритий для відвідувача або призначений для внутрішнього використання (зв'язок, документообіг і тд.).

2.1 Особистий сайт

Для даного типу сайтів можна використовувати будь-яку технологію, так як це проста сторінка із певним текстовим або будь-яким іншим наповненням, яка повинна виконувати ознайомчу роль. Найчастіше це сайт-блог, в якому користувач описує про себе або про компанію чи установу, на рисунках 2.1, 2.2, 2.3, 2.4 актуальні приклади сайт-візиток. Для таких цілей існує багато не складних технологій таких як Google Site. На даний момент з'явилося дуже багато аналогів, які дозволяють недосвідченому користувачу створити так звану сайт-візитку.

Наприклад:

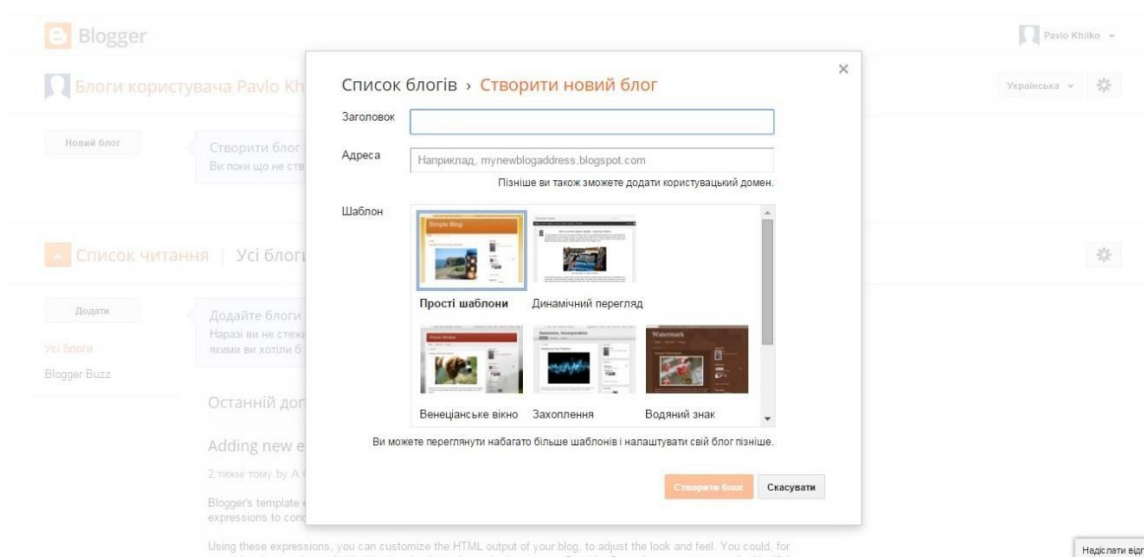


Рисунок 2.1 Сайт-візитка

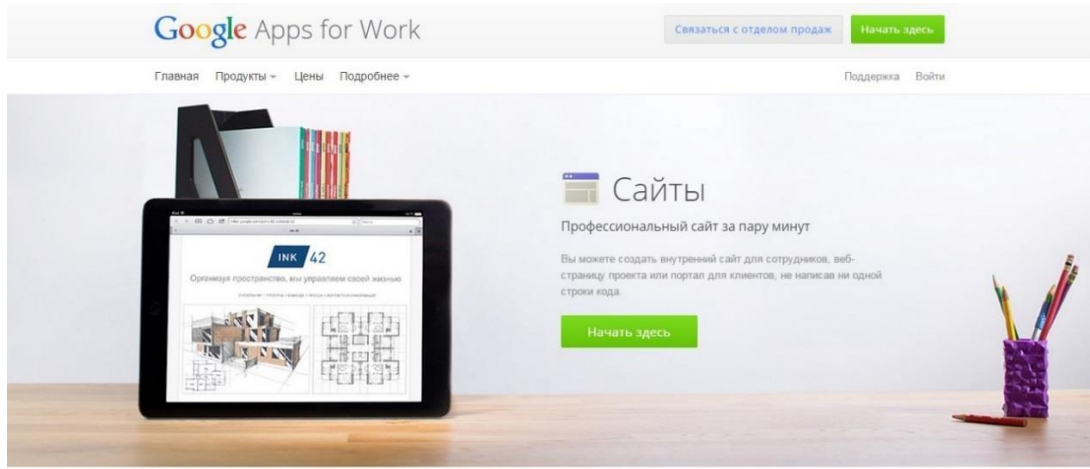


Рисунок 2.2 Сайт-визитка Google Site

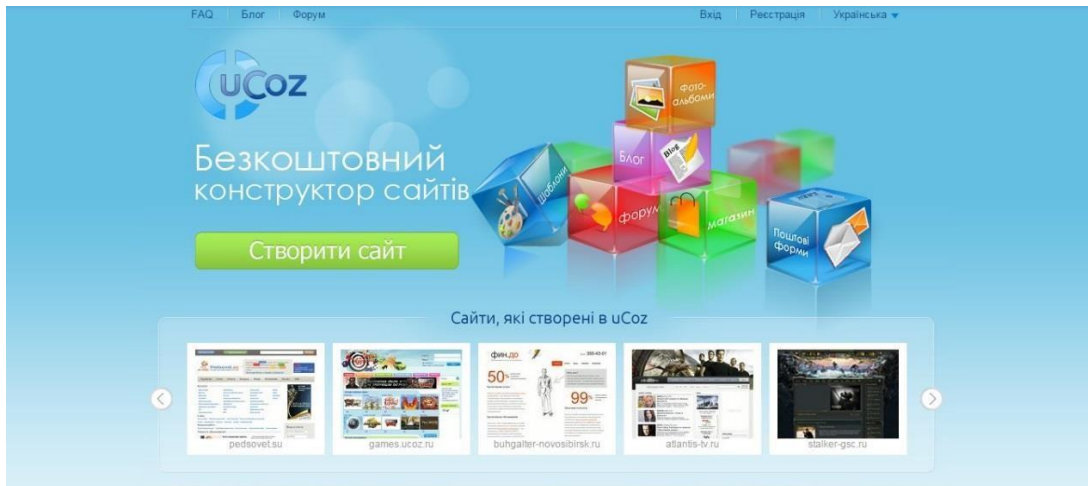


Рисунок 2.3 Сайт-визитка

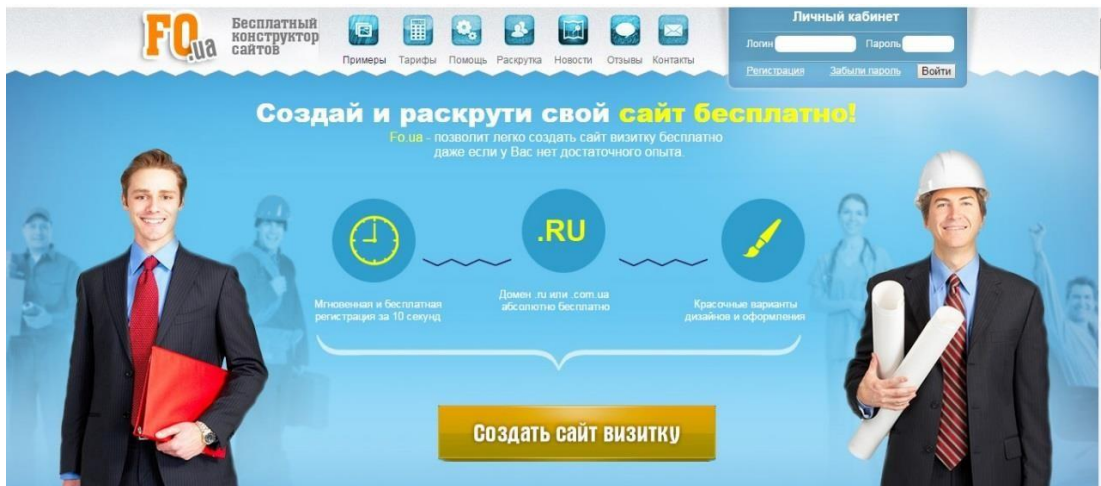


Рисунок 2.4 Сайт-визитка

Результат – невеликий сайт з 3-4 переходами, і вичерпною інформацією.

Ці методи прості в створенні і подальшому використанні.

2.2 Корпоративні сайти та іміджеві портали

Самий кращий варіант для серйозної компанії, яка багато що хоче розповісти своїм клієнтам, що як вже існують, так і потенційним. Це має бути сайт, відповідний фірмовому стилю компанії і орієнтований саме на ті послуги, які компанія надає. Важливі умови ефективної роботи - оновлення сайту разом з розвитком компанії і простота навігації. Основна функція корпоративного сайту - зробити так, щоб клієнт зміг отримати через інтернет, не виходячи з будинку, ті ж самі послуги, неначе він приїхав в офіс компанії. Це стосується як інформації (текст, фотографії або живе спілкування з менеджером), так і процесу оформлення замовлення, виставляння рахунку, визначення способу доставки.

Корпоративний портал — це, в загальному випадку, веб-інтерфейс для доступу співробітника до корпоративних даних і додатків. Часто корпоративний портал сприймається, як синонім інтранету. Альтернативна точка зору полягає в тому, що корпоративний портал — це лише видима для користувача частина інтранету.

З розвитком веб-технологій призначення і можливості корпоративних порталів зазнали ряд змін. Нижче наведені основні групи функціональних можливостей корпоративних порталів, які з'явилися в ході цього розвитку.

Первісним призначенням корпоративних порталів є функції внутрішнього сайту організації:

- Публікація новин та інших матеріалів для співробітників
- Створення бази файлів і документів
- Форум для внутрішнього спілкування

Багато програмних продуктів для створення внутрішніх порталів досі обмежуються даним функціоналом. Основною відмінністю таких порталів від публічних сайтів є система управління правами доступу, яка забезпечує безпеку комерційної інформації.

По вище приведеній інформації, можна зробити висновки, які основні

вимоги та цілі лежать перед корпоративними сайтами та іміджевими порталами. На основі цього найкраще використовувати так званий «ручний метод», а саме написання на певних мовах програмування з використанням framework, наприклад таких як:

- Ruby on Rails
- Python
- Perl
- PHP
- Javascript
- Angularjs

Саме дані технології надають високу швидкість роботи, захищеність та стійкість до значних навантажень.

2.3 Односторінкові сайти

Одно сторінковий додаток (SPA), є веб-додаток або веб-сайт, який поміщається на одній веб-сторінці з метою забезпечення більш однорідного наповнення сторінки, що спрощує користування. У Single page application, або одночасно завантажується весь необхідний код - HTML, JavaScript, CSS і – який витягується з однієї завантаженої сторінки, або відповідні ресурси динамічно завантажується і додається до сторінки, це необхідно, як правило, у відповідь на дії користувача. Сторінка не оновлюється в будь-який момент в процесі роботи, сучасні веб-технології (такі, як ті, які включені в HTML5) можуть забезпечити сприйняття окремих логічних сторінок в додатку. Взаємодія з однієї програми сторінки часто включає в себе динамічну зв'язок з веб-сервером за лаштунками.

SPA повністю завантажує основну частину сторінки, а потім за вимогою користувача здійснюється подальше довантаження додаткових сервісів. Щоб уникнути надмірного завантаження невикористовуваних функцій, SPA часто поступово завантажує більше можливостей, оскільки вони стають обов'язковим, або невеликі фрагменти сторінки, або повні модулі екрану.

Наочними прикладами зображеними на рисунках 2.5, 2.6 є Single page application:



Рисунок 2.5 Сайт-візитка



Рисунок 2.6 Сайт-візитка

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Варіантний аналіз і обґрунтування вибору мови програмування

Для розробки сайту доцільно використати скрипкову серверну мову програмування. До найпопулярніших із них можна віднести PHP, Ruby та Python.

PHP – мова програмування, яка виконується на стороні веб-сервера, спроектована Расмусом Лердорфом як інструмент створення динамічних та інтерактивних веб-сайтів. Ця мова виявилася досить гнучкою і потужною, тому отримала велику популярність і використовується в проектах будь-якого масштабу: від простого блогу до найбільших веб-додатків в Інтернеті.

Переваги PHP:

- є вільним програмним забезпеченням, поширюваним під особливою ліцензією (PHP license);
- легка в освоєнні на всіх етапах;
- підтримується великою спільнотою користувачів і розробників;
- має розвинену підтримку баз даних;
- є величезна кількість бібліотек і розширень мови;
- може використовуватися в ізольованому середовищі;
- є досить повною заміною середовища Active Server Pages від Microsoft;
- може бути розгорнута майже на будь-якому сервері;
- є портативною під велику кількість апаратних платформ і операційних систем [18].

Недоліки PHP:

- не підходить для створення десктопних додатків або системних компонентів;
- має слабкі засоби для роботи з винятками;
- глобальні параметри конфігурації впливають на базовий синтаксис мови, що ускладнює налаштування сервера і розгортання додатків;
- об'єкти передаються за значенням, що бентежить багатьох програмістів, які звикли до передачі об'єктів по посиланню, як це робиться в більшості інших мов;
- веб-додатки, написані на PHP, часто мають проблеми з безпекою.

Згідно статистичних даних, на 1 січня 2013 року PHP був встановлений на більш ніж 240 мільйонах веб-сайтів (39% від усього обсягу досліджених) або на 2,1 мільйонах веб-серверів [19].

Ruby – динамічна об'єктно-орієнтована мова програмування, розроблена Юкіхіро Матсумото. Ruby був створений під впливом таких мов, як Perl, Eiffel і Smalltalk. Вона характеризується динамічною типізацією і автоматичним управлінням пам'яттю. Мова Ruby використовується у веб-розробці в складі відкритого веб-фреймворку Rails, частіше званого Ruby on Rails (RoR).

Переваги Ruby:

- відкрита розробка;
- працює на багатьох платформах;
- відноситься до мов програмування надвисокого рівня (VHLL), тобто володіє високим рівнем абстракції і предметним підходом у реалізації алгоритмів;
- реалізує концептуально чисту об'єктно-орієнтовану парадигму;
- легко інтегрує в свої програми високопродуктивні сервери баз даних (DB2, MySQL, Oracle і Sybase);
- простий і чистий синтаксис значно полегшує програмістам перші кроки в навчанні цієї мови;
- можливості мови можна розширити за допомогою бібліотек,

написаних на C або Ruby;

- додаткові можливості для забезпечення безпеки;
- вбудований відладчик.

Недоліки Ruby:

- навчання мови вище початкового рівня може виявитися непростим;
- інформаційних ресурсів, присвячених Ruby, явно недостатньо;
- Ruby менш продуктивна порівняно з багатьма іншими мовами, вживаними в веб-розробці;
- Ruby відносно повільно розробляється і розвивається.

Python широко застосовується як інтерпретована мова для скриптів різного призначення (хоча існують і транслятори мови Python). Як і Ruby, Python має на меті наблизити синтаксис реальної програми, написаної на ній, до опису задачі псевдокоду, що дозволяє програмісту зменшити обсяг програми. Ідея створення мови виникла наприкінці 1980-х і була реалізована Гвідо ван Россумом [19].

Переваги Python:

- відкрита розробка;
- досить проста у вивченні, особливо на початковому етапі;
- має велике співтовариство, позитивно налаштоване по відношенню до новачків;
- безліч корисних бібліотек і розширень мови можна легко використовувати в своїх проектах завдяки гранично уніфікованому механізму імпорту та програмним інтерфейсами;
- механізми модульності добре продумані і можуть бути легко використані;
- абсолютно все в Python є об'єктами в сенсі ООП, але при цьому об'єктний підхід не нав'язується програмісту.

Недоліки Python:

- не надто вдала підтримка багатопоточності;
- на Python створено не так вже багато якісних програмних проектів у порівнянні з іншими універсальними мовами програмування;

- відсутність комерційної підтримки засобів розробки (хоча ця ситуація з часом змінюється);
- обмеженість засобів для роботи з базами даних;
- бенчмарки показують меншу продуктивність Python [20]. Загальні особливості мов порівнюються у таблиці 3.1.

Таблиця 3.1 – Порівняння мов програмування

Характеристика	PHP	Ruby	Python
Рік створення	1995	1995	1991
Тип виконання	інтерпретатор компілюючого типу	інтерпретований	інтерпретований, компільований в байт-код Java
Система типізації	не жорстка, динамічна, неявна	жорстка, динамічна, явна	жорстка, динамічна, неявна
Підтримка ООП	+	+	+
Стандарти	PSR-0/1/2	ISO/IEC 30170	PEP 8
Відладчик	+ (з версії 5.6)	+	+
Парадигми програмування:	мульти-парадигмальний	мульти-парадигмальний	мульти-парадигмальний

Отже, враховуючи перераховані особливості та переваги, для розробки було обрано мову програмування PHP.

3.2 Варіантний аналіз і обґрунтування вибору середовища розробки

Для розробки додатків мовою програмування PHP на сьогодні створено велику кількість інтегрованих середовищ розробки (IDE) та легких редакторів. Різниця між ними – в тому, що IDE завантажує весь проект цілком, тому може надавати автодоповнення за функціями всього проекту, зручну навігацію по його файлам і т.п. Легкі редактори – редагують конкретний файл (або декілька) і знати не знають про зв'язки між ними. Деякі IDE можна використовувати як легкі

редактори, але зазвичай IDE складніше, важче і працюють повільніше. Розглянемо деякі популярні з них, а саме PHP Storm, Subline Text 2 та NotePad++ [21].

JetBrains PhpStorm – комерційне крос-платформне інтегроване середовище розробки для PHP. Розробляється компанією JetBrains на основі платформи IntelliJ IDEA.

PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок в кодї і автоматизованими засобами рефакторінга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3, 5.4, 5.5 і 5.6 (сучасні і традиційні проекти), включаючи генератори, співпрограми, простір імен, замикання, типажі і синтаксис коротких масивів. Мається повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плагінів, розроблених для платформи IntelliJ, або написавши власні плагіни.

Sublime Text – багатоплатформовий текстовий редактор. Підтримує плагіни на мові програмування Python. Інструментарій програми зручний для редагування вихідних текстів програм.

Sublime Text підтримує велику кількість мов програмування і має можливість підсвічування синтаксису для C, C ++, C #, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, Ruby, SQL, TCL і XML. На додаток до цих мов програмування, які включені в початковий пакет, користувачі мають можливість завантажувати додатки для підтримки додаткових мов.

Можливості Sublime Text:

- автозбереження;
- міні-карта – попередній перегляд повного вихідного коду;
- можливість вибору кількох розділів коду;

- редагування в декількох панелях;
- вбудована підтримка 27 мов програмування;
- пошук і заміна, засновані на регулярних виразах;
- підтримка макросів і плагінів на мові програмування Python;
- користувальницькі гарячі клавіші.

Notepad ++ являє собою вільний текстовий редактор з відкритим вихідним кодом для Windows з підсвічуванням синтаксису великої кількості мов програмування та розмітки. Підтримує відкриття більше 100 форматів. Базується на компоненті Scintilla, написаний на C ++ з використанням STL, а також Windows API і розповсюджується під ліцензією GNU General Public License. Базова функціональність програми може бути розширена як за рахунок плагінів, так і сторонніх модулів, таких як компілятори і препроцесори [22].

Серед найважливіших опцій Notepad ++ можна виділити опцію підсвічування тексту і можливість згортання блоків, згідно синтаксису мови програмування. Користувач може самостійно визначити синтаксис мови програмування. Є можливість налаштувати режим підсвічування. Доступно виділення кольором директив та операторів мови програмування.

Notepad ++ забезпечує можливість одночасного перегляду та редагування декількох документів. Також Ви можете переглядати і редагувати в двох вікнах відображення одного і того ж документа в різних місцях. Зміна документа в одному вікні перегляду буде автоматично переміщена в друге вікно перегляду.

Можливості Notepad ++:

- підсвічування синтаксису;
- згортання коду;
- автодоповнення і автоматичне закриття дужок і тегів (якщо активовано);
- карта документа;

- перевизначення будь-яких гарячих клавіш;
- підтримка великої кількості кодувань;
- паралельне редагування документів;
- порівняння документів;
- робота з файлами на FTP сервері;
- менеджер проектів.

Порівняння вищеописаних середовищ розробки наведено у таблиці 3.2.

Таблиця 3.2 – Порівняння середовищ розробки

Характеристика	PhpStorm	SublineText	Notepad++
Ліцензія	пропрієтарна	пропрієтарна	GNU GPL 2
Підтримка ОС:			
- Windows	+	+	+
- Linux	+	+	-
- Mac OS X	+	+	-
Відлагоджувач	+	-	-
Механізм автодоповнення	+	+	+
Мова реалізації	Java	C++ i Python	C++
Характеристика	PhpStorm	SublineText	Notepad++
Статичний аналіз коду	+	+	+

Враховуючи особливості, наведені вище, а також наявність великої кількості інструментів для швидкої розробки додатків на мові PHP для

розробки найкраще було б обрати PhpStorm. Але, так як дане програмне забезпечення не є безкоштовним, то вибір впав на SublineText.

3.3 Розробка модулів програми

Розроблюваний сайт має модульну структуру, тому опишемо детально розробку та принципи роботи кожного із модулів.

Модуль «База товарів». Використовується для зберігання та обліку виробів майстерні. Головна сторінка показана на рисунку 3.1.

<input type="checkbox"/>	Изображение	Категория	Производитель	Название товара	Модель	Цена на сайте	Количество	Статус	Действие
<input type="checkbox"/>		Кольца		jwr111-001	jwr111-001	4320.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Кольца		jwr111-002	jwr111-002	4200.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Кольца		jwr111-003	jwr111-003	3240.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Кольца		jwr111-004	jwr111-004	2760.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Кольца		jwr111-005	jwr111-005	4440.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Кольца		jwr111-006	jwr111-006	3600.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Кольца		jwr111-007	jwr111-007	3396.0000	1	Включено	[Изменить]

Рисунок 3.1 – Головна сторінка бази товарів

Модуль надає такий функціонал:

- додавання нових записів;
- копіювання записів;
- видалення записів;
- редагування записів.

Модель даних представлена у вигляді класу ModelCatalogProduct, через який здійснюється основні дії з базою даних. Всі маніпуляції здійснюються за допомогою контролера ControllerCatalogCategory. Далі наведено приклад функції контролера, що здійснює додавання нового товару.

```

public function insert() {
    $this->language->load('catalog/category');
    $this->document->setTitle($this->language->get('heading_title'));
    $this->load->model('catalog/category');
        if (($this->request->server['REQUEST_METHOD'] ==
            'POST') &&
    $this->validateForm()) {
        $this->model_catalog_category->addCategory($this->request->post);
        $this->session->data['success'] = $this->language->get('text_success');

        $url = "";
        if (isset($this->request->get['page'])) {
            $url .= '&page=' . $this->request->get['page'];
        }
        $this->redirect($this->url->link('catalog/category', 'token=' .
            $this->session->data['token'] . $url, 'SSL'));
        }
    $this->getForm();
}

```

При додаванні товару з'являється форма, показана на рисунку 3.2.


* Код товара:	<input type="text"/>	Необходима доставка:	<input checked="" type="radio"/> Да <input type="radio"/> Нет
Показать / Скрыть		SEO URL: Должно быть уникальным на всю систему.	<input type="text"/>
Расположение:	<input type="text"/>	Изображение товара:	 Обзор Очистить
Цена:	<input type="text"/>	Дата поступления:	<input type="text" value="2015-04-03"/>
Налог:	<input type="text" value="-- Не выбрано --"/>	Размеры (длина x ширина x высота):	<input type="text"/> <input type="text"/> <input type="text"/>
Количество:	<input type="text" value="1"/>		
Минимальное количество: Указание минимального количества в заказе	<input type="text" value="1"/>		
Вычитать со склада:	<input type="text" value="Да"/>		

Рисунок 3.2 – Додавання товару

Модуль фільтрації даних призначений для швидкого сортування товарів. Представлений у вигляді панелі у верхній частині (рисунок 3.3).

<input type="checkbox"/>	Изображение	Категория	Производитель	Название товара ▲	Модель	Цена на сайте	Количество	Статус
		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Рисунок 3.3 – Панель фільтрації даних

Фільтрація здійснюється за допомогою функції `filter()` контролера `ControllerCatalogCategory`. Далі наведена функція `filter()`.

```
public function filter() {
    $json = array();
    if (isset($this->request->get['filter_name'])) {
        $filter_name = $this->request->get['filter_name'];
    } else {
        $filter_name = null;
    }
    if (isset($this->request->get['filter_category_id'])) {
        $filter_category_id = $this->request->get['filter_category_id'];
    } else {
        $filter_category_id = null;
    }
}
```

Для роботи модуля формування ціни існує клас `ControllerPricePrice` в якому задаються основні параметри сторінки. Нижче наведено код класу `ControllerPricePrice`.

```
class ControllerPricePrice extends Controller
{
    public function index() {
        $this->language->load('price/price');
        $this->document->setTitle($this->language->get('heading_title'));
        $this->data['breadcrumbs'] = array();
        $this->data['breadcrumbs'][] = array(
            'text' => $this->language->get('text_home'),
```

```

        'href'      => $this->url->link('common/home', 'token=' . $this-
        >session-
>data['token'], 'SSL'),
        'separator' => false
    );
    $this->data['breadcrumbs'][] = array(
        'text'      => $this->language->get('heading_title'),
        'href'      => $this->url->link('price/price', 'token=' . $this-
        >session-
>data['token'], 'SSL'),
        'separator' => ' :: '
    );
    $this->template = 'price/price.tpl';
    $this-
        >children =
        array(
            'common/he
            ader',
            'common/fo
            oter'
        );
    $this->response->setOutput($this->render());
    }
}

```

Далі формується масив, що передається у представлення для фільтрування.

```

$data = array(
    'filter_name' => $filter_name,
    'filter_category_id' =>
    $filter_category_id,
    'filter_manufacturer_id' =>
    $filter_manufacturer_id, 'filter_model' =>
    $filter_model,
    'filter_price' =>
    $filter_price,
    'filter_quantity' =>

```

```

$filter_quantity,
'filter_status' =>
$filter_status, 'sort'
=> $sort,
'order' => $order,
'start' => ($page - 1) * $this->config-
>get('config_admin_limit'), 'limit' => $this->config-
>get('config_admin_limit')
);

```

Модуль адміністрування. Призначений для здійснення керування сайтом. Для входу необхідно ввести логін та пароль (рисунок 3.4). В адміністративному модулі можна здійснювати роботу зі всіма іншими через панель управління.

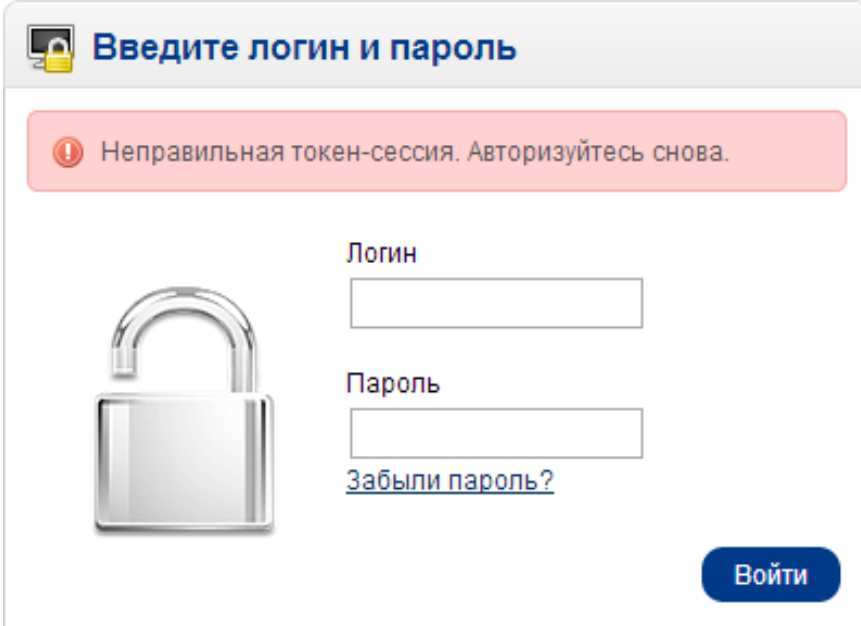


Рисунок 3.4 – Фома входу

Для входу в адміністративну частину використовується контролер `ControllerCommonLogin`. Вся робота побудована на принципах сесії користувача. Для перевірки введених даних використовується функція `validate()`.

```
protected function validate() {
```



```

if (isset($this->request-
>post['username'])
    && isset($this->request-
>post['password'])
    && !$this->user->login($this->request-
>post['username'],
    $this->request->post['password'])) {
    $this->error['warning'] = $this->language->get('error_login');
}
if (!$this->error) {
    return true;
} else {
    return false;
}
}

```

Модуль рекомендованих товарів. Призначений для виводу рекомендованих товарів (рисунок 3.5). За допомогою даного модуля задається розмір виведення зображення та місце.

Рекомендуемые товары



6960 грн.



4320 грн.



1200 грн.



1440 грн.



12720 грн.



7680 грн.

© 2013-2014 Интернет-магазин JWR
Всі права захищені.
[Создание сайтов Design Orbita](#)

Телефоны:
(068) 704-96-94 (093) 589-85-22
Эл. почта: info@jwr.com.ua

Мы в социальных сетях



Рисунок 3.5 – Рекомендовані товари

Для виведення рекомендованих товарів спочатку формується масив `$products_similar`. Далі відбувається його парсинг та занесення у представлення.

```
<?php if ($products_similar) { ?>
```

```

<div id="tab-similar" class="tab-content">
  <div class="box-product">
    <?php foreach ($products_similar as $product) { ?>
      <div>
        <?php if ($product['thumb']) { ?>
          <div class="image"><a href="<?php echo $product['href'];
            ?>">"
            /></a></div>
          <?php } ?>
          <?php if ($product['price']) { ?>
            <div class="price">
              <?php if (!$product['special']) { ?>
                <?php echo $product['price']; ?>
              <?php } else { ?>
                <span class="price-old"><?php echo $product['price']; ?></span>
                <span class="price-new"><?php echo $product['special']; ?></span>
              <?php } ?>
            </div>
          <?php } ?>
          <?php if ($product['rating']) { ?>
            <div class="rating">"
            /></div>
          <?php } ?>
        </div>
      <?php } ?>
    </div>
  <?php } ?>

```

Модуль новінок. Призначений для відображення останніх доданих товарів (рисунок 3.6).

Новинки

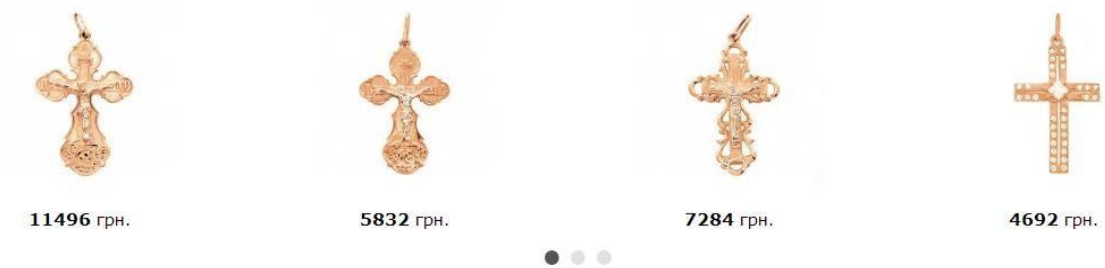


Рисунок 3.6 – Виведення новинок

Виведення новинок формується за допомогою моделі ModelCatalogProduct.

Для отримання новинок снує функція getLatestProducts(), опис якої наведено далі.

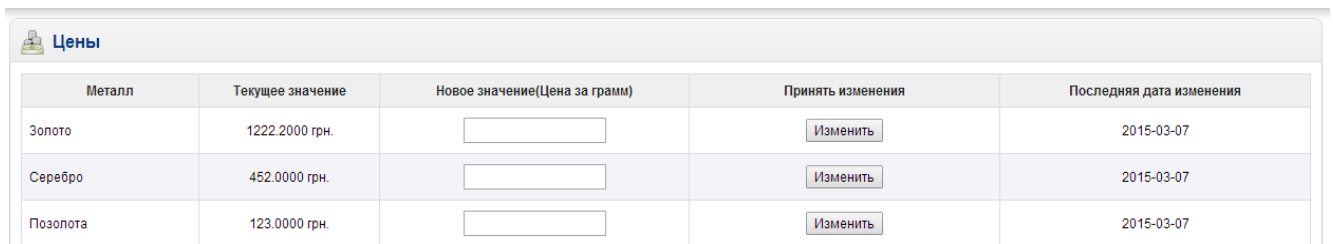
```
public function
getLatestProducts($limit) { if
($this->customer->isLoggedIn()) {
$customer_group_id = $this->customer->getCustomerId();
} else {
$customer_group_id = $this->config->get('config_customer_group_id');
}
$product_data = $this->cache->get('product.latest.' . (int)$this-
>config-
>get('config_language_id') . '.' . (int)$this->config-
>get('config_store_id') . '.' .
$customer_group_id . '.' .
(int)$limit); if (!$product_data)
{
$query = $this->db->query("SELECT p.product_id FROM " .
DB_PREFIX . "product p LEFT JOIN " . DB_PREFIX .
"product_to_store p2s ON AND p.date_available <= " . $this->NOW
. " AND p2s.store_id = " . (int)$this-
>config->get('config_store_id') . " ORDER BY p.date_added DESC
LIMIT " . (int)$limit);
```

```

foreach ($query->rows as $result) {
    $product_data[$result['product_id']] = $this->
    >getProduct($result['product_id']);
}

```

Модуль формування ціни. Призначений для задання цін на всі три типи металів товарів (золото, срібло, позолота), що зображено на рисунку 3.7. Даний модуль використовує технологію Ajax для фонового занесення даних до БД. Також присутній валідатор введення даних за допомогою регулярних виразів, який видає повідомлення, якщо дані введено в невірному форматі.



Металл	Текущее значение	Новое значение(Цена за грамм)	Принять изменения	Последняя дата изменения
Золото	1222.2000 грн.	<input type="text"/>	<input type="button" value="Изменить"/>	2015-03-07
Серебро	452.0000 грн.	<input type="text"/>	<input type="button" value="Изменить"/>	2015-03-07
Позолота	123.0000 грн.	<input type="text"/>	<input type="button" value="Изменить"/>	2015-03-07

Рисунок 3.7 – Задання ціни

Далі наведено програмний код задання ціни разом з валідацією введених даних.

В ході виконання третього розділу було здійснено пошук оптимальної мови програмування для виконання поставленого завдання. В результаті було обрано мову програмування PHP, так як вона відповідає основним вимогам краще ніж інші мови програмування для сфери малого і середнього бізнесу.

Для обраної мови програмування необхідно було вибрати середовище розробки. Спочатку вибір впав на інтегроване середовище PhpStorm, але так, як середовище не є безкоштовним було обрано легкий редактор Sublime Text, який містить різноманітний функціонал для роботи з різними мовами програмування.

Також розроблено основні модулі програми та описано їх функціонал з наведеним кодом та рисунками.

4 ТЕСТУВАННЯ ВЕБ-САЙТУ

4.1 Опис методик, що використовуються при тестуванні веб-сайту

Тестування, як завершальний етап розробки веб-сайту, відіграє життєво важливу роль у процесі створення якісного програмного забезпечення. Чим складніше сайт, тим більше часу потрібно на його перевірку і налагодження.

Починається все з підготовчих робіт – тестувальник вивчає отриману документацію (аналізує функціонал по технічному завданню, вивчає кінцеві макети сайту і складає план тесту для подальшого тестування) [23].

Функціональне тестування – найбільш тривалий етап перевірки ресурсу. Суть цього процесу полягає у перевірці всього описаного функціоналу:

- перевірки роботи всіх обов'язкових функцій сайту;
- тестування працездатності користувальницьких форм на сайті (наприклад, зворотній зв'язок, додавання коментаря в блог);
- перевірки роботи пошуку;
- перевірки гіперпосилань, пошук неробочих посилань;
- перевірки завантаження файлів на сервер;
- перевірки працездатності лічильників, встановлених на сторінках сайту;
- перегляд на відповідність вмісту сторінок сайту вихідного контенту, наданого замовником.

Тестування верстки – при перевірці верстки першим ділом необхідно перевірити розташування елементів, відповідність їх позицій наданим макетів, а також оптимізацію зображень і графіки. Далі здійснюється перевірка валідності коду. У процесі верстки важливо дотримуватися коректної ієрархії об'єктів, і важливо упевнитися в її валідності за фактом завершення робіт. Браузери, незважаючи на явно невірний код, в будь-якому випадку постараються відобразити веб-сторінку. Але оскільки не існує єдиного регламенту про те, як же має бути показаний «кривий» документ, кожен браузер намагається зробити це

по-своєму. А це в свою чергу призводить до того, що один і той же документ може виглядати по-різному в різних браузерах. виправлення явних промахів і систематизація коду призводить, як правило, до стабільного результату. Завершивши перевірку на валідність, фахівець приступає до перевірки на кроссбраузерність, тобто перевіряє працездатність сайту в різних браузерах, а так само при різних параметрах настройки екрану [24].

Usability тестування – проводиться для оцінки зручності продукту у використанні, засноване на залученні користувачів в якості тестувальників і аналіз отриманих результатів.

Тестування безпеки – на даній стадії тестування перевіряється чи немає у користувачів доступу до службових чи закритих сторінок, а також проводиться перевірка захисту всіх критично важливих сторінок (наприклад, розділу адміністрування сайту) від зовнішнього впливу.

4.2 Тестування програми

Для розроблюваного сайту було проведене повноцінне тестування: перевірка посилань, перевірка відповідності текстів заданій темі, перевірка відповідності сайту індивідуальному завданню, перевірка на кроссбраузерність, перевірка на зрозумілість інтерфейсу та коректність ведення даних.

Перевірка інформаційного наповнення показала відповідність текстів заданій тематиці, відсутність орфографічних, стилістичних та пунктуаційних помилок.

Під час перевірки на юзабіліті, виявлено, що сайт є зручним для використання, має просте та інтуїтивно зрозуміле меню, приємну кольорову гаму та нескладну логічну структуру.

Виконано перевірку на кроссбраузерність у такі браузерах: Internet Explorer, Mozilla Firefox, Chrome. При перевірці виявлено, що веб-сайт однаково відображається у всіх браузерах.

Тестування головної сторінки у браузері Google Chrome зображено на рисунку 4.1.

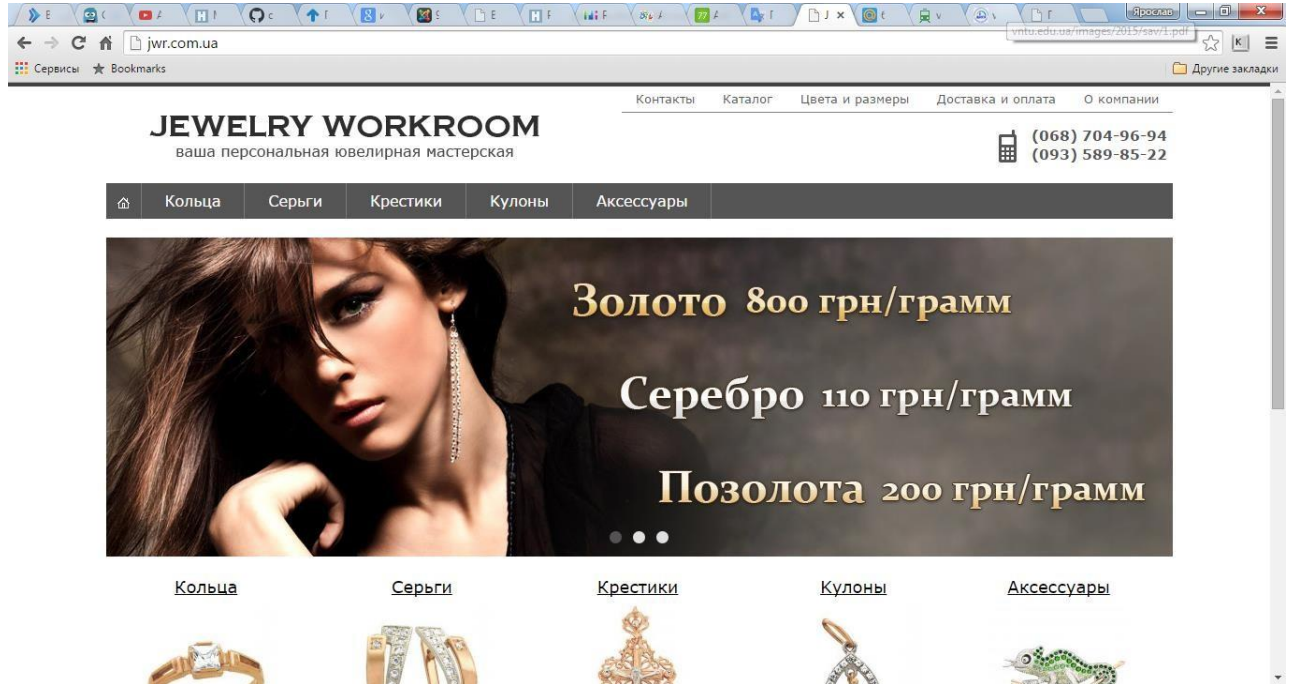


Рисунок 4.1 – Тестування головної сторінки в браузері Google Chrome

Аналогічно проведемо тестування в браузері Mozilla Firefox (рисунок 4.2). Ніяких помилок і некоректностей у відображенні не виявлено.

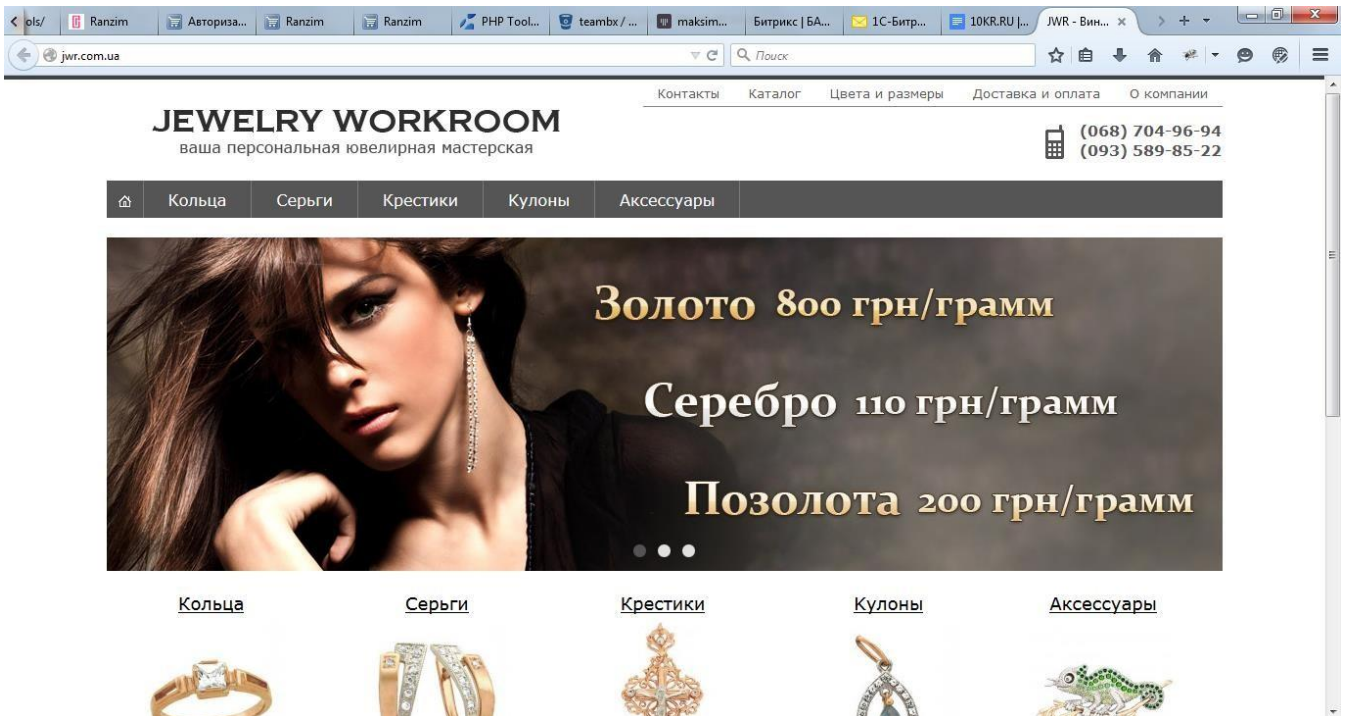


Рисунок 4.2 – Тестування головної сторінки в браузері Mozilla Firefox

Проведено тестування в браузері Internet Explorer (рисунок 4.3). Під час тестування жодних помилок в роботі сайту не було виявлено.

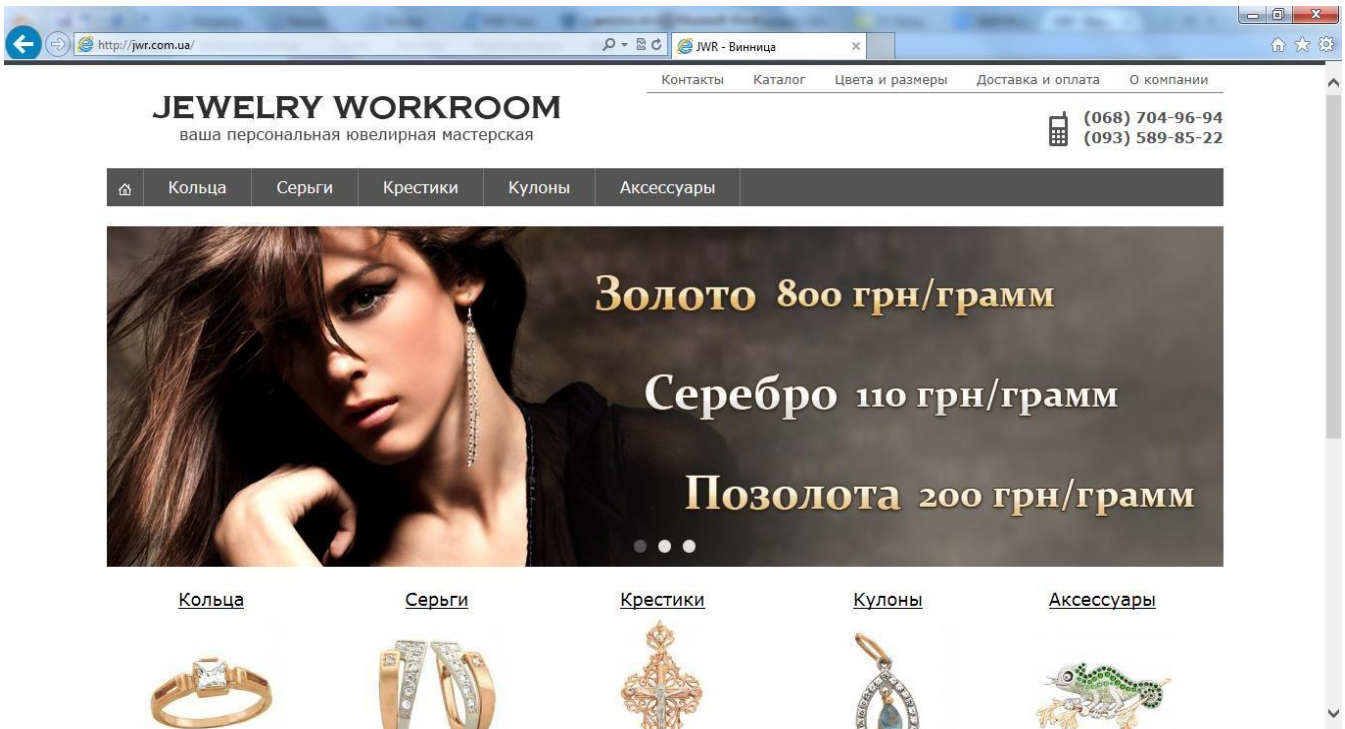


Рисунок 4.3 – Тестування головної сторінки в браузері Internet Explorer

Перевіримо прямий доступ користувача до службової частини. Для захисту адміністративної частини існує розподілений доступ. Щоб увійти до адміністративної частини відкриється форма на якій вводимо логін та пароль. Якщо дані введено не вірно, то буде виведено відповідне повідомлення, що зображено на рисунку 4.4.

Рисунок 4.4 – Форма розподілено доступу

Перевіримо можливість додавання нового виробу до бази даних. Для цього перейдемо до відповідного модуля і додамо новий товар з заданими властивостями (рисунок 4.5).


* Код товара:	<input type="text" value="jwr120-120"/>	Вычитать со склада:	<input type="text" value="Да"/>
Показать / Скрыть		Отсутствие на складе:	<input type="text" value="В наличии"/>
Расположение:	<input type="text"/>	Статус, показываемый, когда товара нет на складе	
Цена:	<input type="text" value="1200"/>	Необходима доставка:	<input checked="" type="radio"/> Да <input type="radio"/> Нет
Налог:	<input type="text" value="--- Не выбрано ---"/>	SEO URL:	<input type="text"/>
Количество:	<input type="text" value="1"/>	Должно быть уникальным на всю систему.	
Минимальное количество:	<input type="text" value="1"/>	Изображение товара:	
Указание минимального количества в заказе			

Рисунок 4.5 – Додавання нового товару

Після додавання нового товару він з'являється у таблиці товарів (рисунок 4.6).








<input type="checkbox"/>		Серьги		jwr120-088	jwr120-088	4800.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Серьги		jwr120-089	jwr120-089	5640.0000	1	Включено	[Изменить]
<input type="checkbox"/>				jwr120-120	jwr120-120	1200.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Крестики		jwr131-001	jwr131-001	6960.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Крестики		jwr131-002	jwr131-002	4320.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Крестики		jwr131-003	jwr131-003	1200.0000	1	Включено	[Изменить]
<input type="checkbox"/>		Крестики		jwr131-004	jwr131-004	1440.0000	1	Включено	[Изменить]

Рисунок 4.6 – Новий запис у базі товарів

Для перевірки роботи фільтрів виберемо категорію «Хрестики», що показано на рисунку 4.7. Після цього відбувається фільтрація і виводиться вироби відповідної категорії.

Товары									
Изображение	Категория	Производитель	Название товара ▲	Модель	Цена на сайте	Количество	Статус	Действие	
	Крестики ▼		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Сброс
	Крестики		jwr131-001	jwr131-001	6960.0000	1	Включено	[Изменить]	
	Крестики		jwr131-002	jwr131-002	4320.0000	1	Включено	[Изменить]	
	Крестики		jwr131-003	jwr131-003	1200.0000	1	Включено	[Изменить]	
	Крестики		jwr131-004	jwr131-004	1440.0000	1	Включено	[Изменить]	
	Крестики		jwr131-005	jwr131-005	12720.0000	1	Включено	[Изменить]	
	Крестики		jwr131-006	jwr131-006	7680.0000	1	Включено	[Изменить]	
	Крестики		jwr131-007	jwr131-007	10800.0000	1	Включено	[Изменить]	

Рисунок 4.7 – Фільтрація даних

При встановленні ціни на товари відбувається контроль введення даних за допомогою регулярних виразів. Для введення ціни забороняється вводити будь-які символи крім цифр та крапки (рисунок 4.8).

Цены			
Металл	Текущее значение	Новое значение(Цена за грамм)	Принять изменения
Золото	1222.2000 грн.	<input type="text" value="прар"/>	Изменить
Серебро	452.0000 грн.	<input type="text" value="апр"/>	Изменить
Позолота	123.0000 грн.	<input type="text" value="апрапр"/>	Изменить

Неверный формат числа!
Можно использовать только цифры
Данные не добавлены. Не верный формат

Рисунок 4.8 – Валідація введення ціни

Під час перевірки гіперпосилань не виявлено ніяких помилок у роботі гіперпосилань. Всі гіперпосилання ведуть на потрібні сторінки. Перехід між сторінками здійснюється з будь-якої сторінки сайту. Перевірка горизонтального та

вертикального меню показала коректність роботи гіперпосилань. Ще одним способом переходу з будь-якої сторінки на головну є натискання на логотип.

Тестування сайтів призначено для того, щоб зробити процес відвідування сайтів приємніше – отже, тестування має велике значення. Воно являє собою поєднання людського управління і можливостей комп'ютера. Тестування дозволить вам створити якісний і оптимізований сайт, який обов'язково стане популярним і визнаним веб-ресурсом.

Описано основні методики для тестування сайту та проведено повноцінне тестування сайту інтернет майстерні ювелірних виробів «Jewerly WorkRoom». Під час проведення тестування веб-сайту виявлено, що сайт відображається однаково в найбільш популярних браузерях, таких як Internet Explorer, Mozilla Firefox, Chrome. Перевірка сайту на інтуїтивність та зручність інтерфейсу показала, що сайт є зручним для користувача, має зручну навігаційну панель та інтуїтивно зрозумілу логічну структуру.

Після проведених тестувань можна зробити висновок, що загалом сайт працює коректно та відповідає поставленим до нього вимогам.

ВИСНОВКИ

Під час виконання бакалаврської роботи було створено дизайн та розроблено комерційний інтернет-магазин "Майстерня ювелірних виробів".

Проаналізовано сучасні технології у створенні веб-сайтів. Найпопулярнішою тенденцією останніх років є мінімалістичний дизайн, який виключає нагромадження об'ємних об'єктів.

Проаналізовано існуючі засоби для розробки веб-сайтів: порівняння серверних мов програмування PHP, Ruby, Python, різноманітних середовищ розробки та легких редакторів. Розроблено мережну структуру веб-сайту. Основні розділи веб-сайту: Контакты, Каталог, Цвета и размеры, Доставка и оплата, О компании.

Розроблено дизайн веб-сайту згідно з основними принципами мінімалістичного дизайну. Створено меню сайту. Розроблено горизонтальне меню каталогу та головне верхнє горизонтальне меню. На детальній сторінці виводиться форма замовлення виробу.

Протестовано веб-сайт у найпопулярніших браузерях та була проведена перевірка валідації даних. В ході тестування виявлено, що сайт працює коректно.

Для готового програмного продукту було проведене повноцінне тестування: перевірка гіперпосилань, перевірка відповідності текстів заданій темі, перевірка відповідності сайту індивідуальному завданню, перевірка на кросбраузерність та юзабіліті.

Отже, в ході виконання бакалаврської роботи було продемонстровано практичні навички створення та розробки веб-сайту.

СПИСОК ЛІТЕРАТУРИ

1. Орлов Л. А. Як створити електронний магазин в Інтернет / Л. А. Орлов. – М.: БУК-ПРЕС, 2006. – 384 с.
2. Калужский М.Л. Электронная коммерция / М. Л. Калужский. – М.: Экономика, 2014. – 328 с.
3. Старовойтова Т. Ф. Электронный бизнес и коммерция / Т. Ф. Старовойтова. – К.: ТетраСистемс, 2009. – 144 с.
4. Нельзина О. Г. Информационные системы для электронной коммерции. Учебное пособие / О. Г. Нельзина. – СПб.: Феникс, 2008. – 265 с.
5. Веб-сайт «Золотий вік» [Електронний ресурс] // Режим доступу : <http://www.zolotoyvek.ua/ru/> – Назва з екрану.
6. Веб-сайт «GoldMaster» [Електронний ресурс] // Режим доступу : <http://goldmaster.in.ua/> – Назва з екрану
7. Веб-сайт «Мальва» [Електронний ресурс] // Режим доступу : <http://www.malva.vn.ua/> – Назва з екрану.
8. Технологии создания сайтов [Електронний ресурс] // Режим доступу : http://www.dbest.ru/information/advice/site_technologies/ – Назва з екрану.
9. Лабберс П. HTML5 для профессионалов. Мощные инструменты для разработки современных веб-приложений / П. Лабберс. – М.: Вильямс, 2011. – 272 с.
10. Ши Д. Философия CSS дизайна / Д. Ши. – СПб.: НТ Пресс, 2015. – 312 с.
11. Зандстра М. А. Объекты, шаблоны и методики программирования, 3-е издание / М. А. Зандстра. – СПб.: Издательский дом «Питер», 2011. – 560 с.
12. Буч Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Якобсон. – М.: ДМК Пресс, 2007. – 496 с.
13. Кэмпбел М. Строим Web-сайты. Дизайн. HTML. CSS / М. Кэмпбел.

– М.: Триумф, 2006. – 480 с.

14. Джамса К. Эффективный самоучитель по креативному Web-дизайну / К. Джамса. – СПб.: ДиаСофтЮп, 2005. – 672 с.

15. 5 причин использовать минималистический дизайн [Электронный ресурс]

// Режим доступа : <http://www.imagecms.net/blog/obzory/minimalisticheskii-dizain-saita>

– Назва з екрану.

16. Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Бен Фрейн – СПб.: Питер, 2014. – 304 с.

17. Раскин Д. Интерфейс: Новые направления в проектировании компьютерных систем / Д. Раскин. – М.: Символ-Плюс, 2004. – 272 с.

18. Когзолл Д. PHP 5. Полное руководство / Д. Когзолл. – М.: Диалектика, 2006. – 752 с.

19. PHP, Ruby, Python – краткая характеристика языков программирования [Электронный ресурс] // Режим доступа : http://www.internet-technologies.ru/articles/article_1991.html – Назва з екрану

20. Саммерфилд М. Программирование на Python 3 / М. Саммерфилд. – СПб.: Символ-Плюс, 2009. – 608 с.

21. Инструменты и библиотеки для разработчика приложений PHP [Электронный ресурс] : // Режим доступа : <http://kb.mista.ru/article.php?id=670> – Назва з екрану

22. Сравнение сред разработки PHP [Электронный ресурс] : // Режим доступа : <https://toster.ru/q/8531> – Назва з екрану.

23. Уиттакер Д. А, Арбон Д. В. Как тестируют в гугл / Д. А Уиттакер, Д. В. Арбон. – СПб.: Питер, 2014. – 385 с.

24. Стотлемайер Д. Тестирование Web-приложений / Д. Стотлемайер – М.: Кудиц-образ, 2003. – 240 с.

ДОДАТОК А

Лістинг головної сторінки index.php

```

    <?php
// Version

define('VERSION', '1.5.5.1.1');

// Configuration
if
    (file_exists('config
    .php')) {
    require_once('config.
    php');
}

// Install
if (!defined('DIR_APPLICATION')) {
    header('Location:
    install/index.php'); exit;
}

// VirtualQMOD
require_once('./vqmod/vqmod.
php'); VQMod::bootup();

// VQMODDED Startup
require_once(VQMod::modCheck(DIR_SYSTEM .
'startup.php'));

// Application Classes
require_once(VQMod::modCheck(DIR_SYSTEM .
'library/customer.php'));
require_once(VQMod::modCheck(DIR_SYSTEM .
'library/affiliate.php'));
require_once(VQMod::modCheck(DIR_SYSTEM .
'library/currency.php'));
require_once(VQMod::modCheck(DIR_SYSTEM .
'library/tax.php'));
require_once(VQMod::modCheck(DIR_SYSTEM .
'library/weight.php'));
require_once(VQMod::modCheck(DIR_SYSTEM .
'library/length.php'));
require_once(VQMod::modCheck(DIR_SYSTEM .
'library/cart.php'));
require_once(VQMod::modCheck(DIR_SYSTEM .
'library/ocstore.php'));

// Registry
$registry = new Registry();

// Loader
$loader = new Loader($registry);
$registry->set('load', $loader);

// Config
$config = new Config();
$registry->set('config', $config);

// Database
$db = new DB(DB_DRIVER, DB_HOSTNAME, DB_USERNAME, DB_PASSWORD, DB_DATABASE);
$registry->set('db', $db);

```

```

// Store
if (isset($_SERVER['HTTPS']) && (($_SERVER['HTTPS'] == 'on') || ($_SERVER['HTTPS'] ==
'1'))))
{
    $store_query = $db->query("SELECT * FROM " . DB_PREFIX . "store WHERE
REPLACE(`ssl`, 'www.', '') = '' . $db->escape('https://' . str_replace('www.', '',
$_SERVER['HTTP_HOST'])).
rtrim(dirname($_SERVER['PHP_SELF']), './\') . '/') . """);
} else {
    $store_query = $db->query("SELECT * FROM " . DB_PREFIX . "store WHERE
REPLACE(`url`, 'www.', '') = '' . $db->escape('http://' . str_replace('www.', '',
$_SERVER['HTTP_HOST'])).
rtrim(dirname($_SERVER['PHP_SELF']), './\') . '/') . """);
}

if ($store_query->num_rows) {
    $config->set('config_store_id', $store_query->row['store_id']);
} else {
    $config->set('config_store_id', 0);
}

// Settings
$query = $db->query("SELECT * FROM " . DB_PREFIX . "setting WHERE store_id = '0' OR
store_id
= '' . (int)$config->get('config_store_id') . "" ORDER BY store_id ASC");

foreach ($query->rows as
    $setting) { if
        (!$setting['serialize
        d']) {
            $config->set($setting['key'], $setting['value']);
        } else {
            $config->set($setting['key'], unserialize($setting['value']));
        }
    }

if (!$store_query->num_rows) {
    $config->set('config_url', HTTP_SERVER);
    $config->set('config_ssl', HTTPS_SERVER);
}

// Url
$url = new Url($config->get('config_url'), $config->get('config_secure') ? $config-
>get('config_ssl') : $config->get('config_url'));
$registry->set('url', $url);

// Log
$log = new Log($config->get('config_error_filename'));
$registry->set('log', $log);

function error_handler($errno, $errstr, $errfile,
    $errline) { global $log, $config;

    switch ($errno) {
        case E_NOTICE:
        case E_USER_NOTICE:
            $error =
                'Notice';
            break;
        case E_WARNING:
        case E_USER_WARNING:
            $error =
                'Warning';
    }
}

```



```

        break;
    case E_ERROR:
    case E_USER_ERROR:
        $error =
            'Fatal Error';
        break;
    default:
        $error =
            'Unknown';
        break;
    }

    if ($config->get('config_error_display')) {
        echo '<b>' . $error . '</b>: ' . $errstr . ' in <b>' . $errfile . '</b>'
        on line
        <b>' . $errline . '</b>';
    }

    if ($config->get('config_error_log')) {
        $log->write('PHP ' . $error . ': ' . $errstr . ' in ' . $errfile . ' on
        line '
        . $errline);
    }

    return true;
}

// Error Handler
set_error_handler('error_handler');

// Request
$request = new Request();
$registry->set('request', $request);

// Response
$response = new Response();
$response->addHeader('Content-Type: text/html; charset=utf-8');
$response->setCompression($config->get('config_compression'));
$registry->set('response', $response);

// Cache
$cache = new Cache();
$registry->set('cache', $cache);

// Session
$session = new Session();
$registry->set('session', $session);

// Language Detection
$languages = array();

$query = $db->query("SELECT * FROM `". DB_PREFIX . "language` WHERE status =
'1'"); foreach ($query->rows as $result) {
    $languages[$result['code']] = $result;
}

$detect = '';

if (isset($request->server['HTTP_ACCEPT_LANGUAGE']) && $request-
>server['HTTP_ACCEPT_LANGUAGE']) {
    $browser_languages = explode(',', $request->server['HTTP_ACCEPT_LANGUAGE']);
}

```

```

        foreach ($browser_languages as
            $browser_language) { foreach
            ($languages as $key => $value) {
                if ($value['status']) {
                    $locale = explode(',', $value['locale']);

                    if (in_array($browser_language, $locale)) {
                        $detect = $key;
                    }
                }
            }
        }

        if (isset($session->data['language']) && array_key_exists($session->
            >data['language'],
            $languages) && $languages[$session->data['language']]['status']) {
            $code = $session->data['language'];
        } elseif (isset($request->cookie['language']) && array_key_exists($request->
            >cookie['language'], $languages) && $languages[$request->
            >cookie['language']]['status']) {
            $code = $request->cookie['language'];

        } elseif ($detect) {
            $code = $detect;
        } else {
            $code = $config->get('config_language');
        }

        if (!isset($session->data['language']) || $session->data['language'] != $code) {
            $session->data['language'] = $code;
        }

        if (!isset($request->cookie['language']) || $request->cookie['language'] !=
            $code) { setcookie('language', $code, time() + 60 * 60 * 24 * 30,
            '/', $request->
            >server['HTTP_HOST']);
        }

        $config->set('config_language_id', $languages[$code]['language_id']);
        $config->set('config_language', $languages[$code]['code']);

        // Language
        $language = new Language($languages[$code]['directory']);
        $language->load($languages[$code]['filename']);
        $registry->set('language', $language);

        // Document
        $registry->set('document', new Document());

        // Customer
        $registry->set('customer', new Customer($registry));

        // Affiliate
        $registry->set('affiliate', new Affiliate($registry));

        if (isset($request->get['tracking'])) {
            setcookie('tracking', $request->get['tracking'], time() + 3600 * 24 * 1000,
                '/');
        }

        // Currency
        $registry->set('currency', new Currency($registry));

```

```

// Tax
$registry->set('tax', new Tax($registry));

// Weight
$registry->set('weight', new Weight($registry));

// Length
$registry->set('length', new Length($registry));

// Cart
$registry->set('cart', new Cart($registry));

// ocStore features
$registry->set('ocstore', new ocStore($registry));
// Encryption
$registry->set('encryption', new Encryption($config->get('config_encryption')));

// Front Controller
$controller = new Front($registry);
// SEO URL's
if (!$seo_type = $config->get('config_seo_url_type')) {
    $seo_type = 'seo_url';
}
$controller->addPreAction(new Action('common/' . $seo_type));

// Maintenance Mode
$controller->addPreAction(new Action('common/maintenance'));

// Router
if (isset($request->get['route'])) {
    $action = new Action($request->get['route']);
} else {
    $action = new Action('common/home');
}

// Dispatch
$controller->dispatch($action, new Action('error/not_found'));

// Output
$response->output();
?>
    sql += "));";
    SQLQuery2->SQL-
    >Text = sql;
    SQLQuery2-
    >Execute();
    RefreshSQL();
}
//-----

```

ДОДАТОК Б

Лістинг каскадної таблиці стилів style.css

```

body {
font: 13px verdana, tahoma, sans-serif; color: #000;
line-height: 18px;
}
a {
color: #5a5a5a; font-size: 13px;
}
a:hover {
text-decoration: none;
}
.top_nav li { float: right;
}
.top_nav a { color: #5a5a5a;
margin: 0px 15px; line-height: 27px; text-decoration: none;
}
.header_bg {
background-color: #fff;
}
.header .fourcol { padding-top: 0px; height: 75px;
}
.search {
text-align: center;
}
.search div {
display: inline-block; margin-top: 8px;
}
.search input[type="text"] { width: 200px;
float: left;
border: 1px solid #d0d0d0; height: 27px;
border-right: none; text-indent: 5px; outline: none;
}
.search input[type="button"] { width: 60px;
font-size: 14px; color: #fff; height: 31px; border: none;
background-color: #4c7dbb; cursor: pointer;
}
.tel {
margin-top: 14px;
background: url(..img/tel_bg.png) 0px 0px no-repeat; padding-left: 35px;

line-height: 20px; display: inline-block; float: right;
color: #555;
}
.tel p {
font-weight: bold; margin-bottom: 0px; font-size: 15px;
}
.tel a {
margin-right: 56px;
}
.nav {
height: 38px; background-color: #555;

```

```

}
.nav li {
float: left;
}
.nav li a {
display: inline-block; line-height: 38px; color: #fff;
text-decoration: none;
border-right: 1px solid #7a7a7a; padding: 0 25px;
font-size: 16px;
}
@media handheld, only screen and (max-width: 1080px) {
.nav li a { padding: 0 10px; transition: 0.45s;
}
}
@media handheld, only screen and (max-width: 1020px) {
.nav li a { padding: 0 7px; transition: 0.45s;
}
}
.nav li:first-child a { padding: 0 12px;
}
.nav li a:hover {
background-color: #0D3A73; transition: 0.65s;
}
.active_nav a {
background-color: #0D3A73;
}
.header_bg {
/* box-shadow: 2px 2px 2px 0px #ccc;*/
}
.no_decor {
text-decoration: none;
}
.header_bg {
margin-bottom: -3px;
}
.title {
display: block;
/* font: bold 15px tahoma, sans-serif;*/

font-size: 18px; color: #000;
line-height: 20px;
margin: 5px 0px 10px 0px !important;
}
.sticker {
/* background: url(..img/sticker.png) 0 0 no-repeat;*/ display: inline-block;
/* width: 84px;*/ height: 26px;
/* line-height: 36px;*/ font-size: 15px; padding-left: 25px; padding-right: 35px; color:
#000 !important; text-decoration: none; overflow: hidden;
}
.sticker span, .tovar_price span { font-weight: bold;
}
.prod_name { display: block; text-align: center;
/* margin-bottom: 10px; margin-top: 10px;*/ height: 30px;
}

.category_home_name { display: block;
}

.prod_col {

```

```

text-align: center;
display: inline-block!important; float: none!important;
vertical-align: top!important; margin-right: 1.00%!important;
}
.prod_col a img {
/* box-shadow: 1px 1px 1px 0 #ccc;*/ margin-bottom: 5px;
}
.slider {
height: 23px;
/* background: url(..img/slide_bg.png) 0 0 repeat-x;*/ margin-top: 10px;
text-align: center; padding-top: 5px;
border-bottom: 1px solid #ccc;
}
.slider a {
display: inline-block!important; width: 13px!important;
height: 13px!important;
background: url(..img/link.png) 0 0 no-repeat!important; margin-right: 10px!important;
}
.slider a:last-child {
margin-right: 0px!important;
}
.slider .active_link {

background: url(..img/active_link.png) 0 0 no-repeat;
}
.slide_category {
margin: 20px 0px 0px 0px;
/* border-top: 1px solid #ccc;*/ padding-top: 10px;
}
.slide_category .slider{
border-bottom: none !important;
}
.slide_category .jcarousel-wrapper{ margin-bottom: 0px ;
}
.slide_category .threecol { width: 23.9%!important;
}
.ninecol .category_col, .ninecol .prod_col { width: 32.25%!important;
}
.category_col, .category_col a { text-align: left;
}
    .category_col a, .prod_col a, .left_menu a {
        /*color: #1c6399;*/
    }
.category_col a { margin-bottom: 0; line-height: 20px;
}
.category_col a img { margin-bottom: 10px;
}
.category_col a:last-child { margin-bottom: 25px;
}
.main_text { color: #555555; font-size: 13px;
}
.footer_bg {
/* border-bottom:4px solid #424242; */

font-size: 13px;
padding: 10px 0px 0px 0px; height: 90px;
/* box-shadow: 2px 2px 2px 0px #ccc;*/ line-height: 18px;
margin-top: 10px;
}

```

```

.create { height: 30px;
text-align: right; padding-top: 20px;
}
    .social {
        float: right;
        padding-right: 40px;
    }
    .social p {
        margin-bottom: 5px;
    }
    .social a {
        display: inline-block;

margin-right: 10px;
}
    .adress_line { color: #8f8f8f; height: 25px;
margin-top: 7px;
}
    .adress_line a { font-size: 12px;
}
.content {
    background-color: #fff;

padding: 20px 20px
    0px 0px;
    margin-
    bottom:
    10px;
font-size: 14px !important;
}
    .left_menu a { display: block; font-size: 14px; line-height: 26px;
}
    .cat_name {
font-size: 14px!important; line-height: 26px!important;
}
    .msfeatureupdater{ display: none!important;
}
    .products .prod_col { margin-bottom: 20px;
}
.pagination {
text-align: center; margin-top: 25px;
}
.pagination li {
display: inline-block;
}
    .pagination li a { font-weight: bold; font-size: 14px;
display: inline-block; margin-right: 3px; width: 18px;
height: 18px;
}
    .pagination li:first-child a, .pagination li:last-child a { font-weight: normal;
width: auto!important;
}
    .no_active a { color: #b5b5b5;
text-decoration: none;
}
}

```