

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Веб-застосунок для торгівлі виробами із шкіри

Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр КвРІПЗ.190130.01.05.ПЗ

Виконав студент IV курсу, група ІПЗ-19-1


підпис

Н. В. Данькевич
Ініціали, прізвище

Керівник канд. техн. наук, доцент


підпис

Ю. В. Форкун.
Ініціали, прізвище

Нормоконтролер канд. техн. наук, доцент


Підпис

І. В. Гурман
Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення


Підпис

Л. П. Бедратюк
Ініціали, прізвище

7 червня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри 

Л. П. Бедратюк

06 03 2023 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ)

Данькевичу Назарію Вікторовичу

Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи) Веб-застосунок для торгівлі виробами із шкіри

Керівник проєкту (роботи) Форкун Юрій Вікторович, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проєкту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проєкту (роботи) Матеріали переддипломної практики, Методичні вказівки до виконання кваліфікаційної роботи





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

Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація системи та тестування програмної системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проєкту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|---|--|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Гурман І.В., доцент кафедри ПЗ | 5.06.2023  | 6.06.2023  |
| Антиплагіат | Гурман І.В., доцент кафедри ПЗ | 6.06.2023  | 6.06.2023  |

7. Дата видачі завдання « 05 » лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

| Назва етапів (розділів) дипломного проєкту (роботи) | Строк виконання етапів проєкту (роботи) | Примітка |
|---|---|----------|
| 1 Ознайомлення з тематикою дипломного проєктування (ДП), визначення та узгодження індивідуальних тем ДП | 05.02 – 14.02.2023 | |
| 2 Дослідження предметної області, в якій планується використання програмного засобу (ПЗ), визначення задач та вимог, розробка технічного завдання | 15.02 – 28.02.2023 | |
| 3 Проєктування програмного забезпечення | 01.03 – 30.03.2023 | |
| 4 Програмна реалізація | 01.04 – 20.04.2023 | |
| 5 Тестування програмного забезпечення | 21.04 – 30.04.2023 | |
| 6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів | 01.05 – 25.05.2023 | |
| 7 Попередній захист ДП | Травень 2023 (згідно графіка) | |
| 8 Перевірка ДП на плагіат, нормконтроль, отримання відгуків та рецензій. Брошування (зшиття) пояснювальної записки | 26.05 – 30.05.2023 | |
| 9 Підготовка до захисту та захист ДП | з 01.06.2023 | |

Студент


 Підпис


 Ініціали, прізвище

Керівник проєкту (роботи)


 Підпис


 Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Веб-застосунок для торгівлі виробами із шкіри»

Автор проекту: Данькевич Назарій Вікторович

Керівник проекту: Форкун Юрій Вікторович

Пояснювальна записка: 97 с., 11 рис., 3 табл., 5 дод., 44 джерела

Графічна частина: 15 презентаційних слайдів

Ключові слова: BR, CSS, CMF, CMS, DAO, FT, PT, ST, UIT, UT.

Мета кваліфікаційної роботи: розробка архітектури, компонентів веб застосунку створення інтернет-магазину на базі платформи WordPress для продажу товарів та послуг в інтернеті.

Практичне значення полягає у створенні веб- сайту інтернет-магазину, завдання якого полягає у обробці у застосунку продажу виробів зі шкіри, обробки даних користувачі та продажів виробів, .використання засобів комунікації з клієнтами. Розроблений інтернет-магазин має дозволити власникам в загальному значно розширити кількість клієнтів, збільшити продажі товарів і послуг та дасть можливість скоротити час всього процесу торгівлі - від надання виробу до його продажу, поліпшити управління веб-магазином, знизити витрати на оренду та покупку торгових площі та наймання великої кількості обслуговуючого персоналу.

Основна перевага даної системи полягає у зручності користування та простоті користування нею.

05.06.2023



ЗМІСТ

| | |
|--|----|
| ПЕРЕЛІК СКОРОЧЕНЬ | 7 |
| ВСТУП | 8 |
| 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ЗА ТЕМОЮ ТА ПОСТАНОВКА ЗАДАЧІ | |
| 1.1 Цілі та завдання створення веб-застосунку виробів зі шкіри..... | 10 |
| 1.2 Поняття веб-магазину | 11 |
| 1.3 Аналіз предметної області..... | |
| 1.4 Аналіз відомих проектів | 5 |
| 1.5 Постановка задачі..... | 8 |
| 1.6 Висновки першого розділу..... | 9 |
| 2 Аналіз засобів проектування та розробка архітектури застосунку | 20 |
| 2.1 Цілі і завдання розвитку інтернет-магазину | 20 |
| 2.2 Обґрунтування вибору системи управління контентом | 25 |
| 2.3 Опис системи управління контентом WORDPRESS..... | 17 |
| 2.4 Опис модуля-плагіну WOOCOMMERCE | 27 |
| 2.5 Розробка архітектури застосунку | 27 |
| 2.6 Висновки другого розділу..... | 32 |
| 3. Програмна реалізація застосунку | 33 |
| 3.1 Проектування та розробка даних для веб-застосунку..... | 33 |
| 3.2 Створення та заповнення бази даних | 37 |
| 3.3 Розробка програмних модулів веб-застосунку | 42 |
| 3.4 Інструкція користувача | 52 |
| 3.5 Тестування веб-застосунку та аналіз результатів..... | 54 |

| | | | | | | | | |
|------------------|-------------|------------------------|---------------|--------------|--|----------------------|-------------|----------------|
| | | | | | <i>КвРІПЗ.190130.01.05.ПЗ</i> | | | |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | <i>Веб-застосунок для торгівлі виробами із шкіри</i> | <i>Літ.</i> | <i>Арк.</i> | <i>Аркушів</i> |
| <i>Виконав</i> | | <i>Данькевич Н. В.</i> | | <i>06.06</i> | | | 4 | 56 |
| <i>Керівник</i> | | <i>Форкун Ю. В.</i> | | <i>06.06</i> | | <i>ХНУ, ІПЗ-19-1</i> | | |
| <i>Реценз.</i> | | | | | | | | |
| <i>Н. Контр</i> | | <i>Гурман І. В.</i> | | <i>6.06</i> | | | | |
| <i>Затвердив</i> | | <i>Бедратюк Л.П.</i> | | <i>7.06</i> | | | | |

| | |
|-------------------------------------|----|
| 3.6 Висновки третього розділу | 56 |
| ВИСНОВКИ | 57 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ | 59 |
| ДОДАТКИ | 63 |
| ДОДАТОК А | 63 |
| ДОДАТОК Б..... | 69 |
| ДОДАТОК В..... | 70 |
| ДОДАТОК Г..... | 79 |
| ГРАФІЧНА ЧАСТИНА | 93 |

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|------------------------------|------|
| | | | | | <i>КвРПЗ.190130.01.05.ПЗ</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 6 |

ПЕРЕЛІК СКОРОЧЕНЬ

| | |
|-------|------------------------------|
| CMF - | Content Management Framework |
| DAO - | Data access object |
| FT | Functionality testing |
| UIT | UI testing |
| UT | Usability testing |
| PT | Performance testing |
| ST | Security testing |

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 8 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

ВСТУП

В умовах сьогодення людина постійно зайнята різними справами та практично не має часу та зусиль, на відвідування різних магазинів, а також тривалі і виснажливі подорожі. Одним з варіантів є купівля необхідного в інтернет-магазинах де можна обрати необхідну йому річ. Майже усі сучасні інтернет-магазини, дозволяють підібрати певний товар навіть з доставкою, так щоб клієнт мав право оцінити річ та вибрати саме таку яку йому потрібно. Зараз у світі існує велика кількість різноманітних інтернет-магазинів, як для певного роду товарів, так і загального спрямування: Ebay, AliExpress, Розетка, Алло тощо. Величезний вибір інтернет-магазинів, вузького спрямування, такі як: електротехніки, господарські магазини, магазини одягу, спортивні магазини, для залучення клієнтів організовують різноманітні акції, проводять різні рекламні та заохочувальні кампанії. При успішній рекламній кампанії та маркетингу вони мають прибуток кращий ніж звичайні магазини.

В даний час розробка сайтів є досить розповсюдженим та поширеним видом діяльності де програмуванням та розробкою сайту займається значна кількість людей. Найбільшим досягненням в технологіях розробки різнотипних сайтів є розробка сайту на основі систем управління контентом. Засоби управління контентом широко використовуються в інтернет-просторі при проектуванні веб-сайтів різних типів та різного рівня їх складності.

Основними відмінними характеристиками CMS-систем, які в той же час є їх перевагами, є можливість проектування сайтів, на практиці будь-якого рівня складності без використання знань в мовах розмітки HTML, CSS та інших веб-орієнтованих мов програмування. Крім того є можливість створювати, видаляти, редагувати та формувати певний контент практично легко, інтуїтивно, швидко, що досить спрощує завдання проектування та адміністрування самого сайту.

Особливо популярними CMS, на сьогодні, для створення інтернет-магазину є OpenCart, Wordpress, PrestaShop.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 9 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Загалом ці CMS являють собою систему управління контентом з відкритим вихідним кодом, написана на мові PHP, з використанням сервера баз даних MySQL, який працює під ліцензією GNU GPL версії 2 та вище. Сфера їх застосування: від різноманітних блогів до досить складних веб сайтів, інтернет-ресурсів та інтернет-магазинів. Вбудована система різних тем та плагінів, поряд з побудованою архітектурою дозволяє розробляти проекти різноманітної широкої функціональної складності.

Таким чином, стає актуальною задачею для нашого завдання є розробка інтернет-магазину виробів із шкіри.

Метою нашого завдання є розробка інтернет-магазину на базі платформи Wordpress для торгівлі виробами із шкіри в інтернеті. Для досягнення мети проекту необхідно виконати такі задачі:

- дослідити предметну область поставленого завдання та визначити особливості та специфіку процесу побудови веб-сайту інтернет-магазину;
- провести аналіз відомих існуючих рішень предметної області;
- виокремити функціональні задачі предметної області;
- здійснити аналіз засобів розробки, інструментів та плагінів, які мають бути використані при для створення інтернет-магазину;
- спроектувати та розробити архітектуру застосунку;
- спроектувати та розробити функціональний сучасний інтерфейс веб-застосунку;
- здійснити програмну реалізацію веб-застосунку;
- провести тестування готового розробленого програмного застосунку.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 10 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ЗА ТЕМОЮ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Цілі та завдання створення веб-застосунку виробів зі шкіри

Метою даної кваліфікаційної випускної роботи є створення веб-магазину на базі платформи WordPress для продажу виробів зі шкіри в Веб.

Загальними завданнями дипломного проекту для реалізації нашої мети є:

- аналіз та опис предметної області;
- постановка завдання на розробку веб-магазину;
- технічне завдання для вирішення проектування та розробки програмного застосунку веб-магазину;
- підбір програмних та технічних засобів для реалізації створення веб-магазину;
- подання та опис архітектури веб-магазину;
- програмна реалізація веб-магазину.

Практичне значення даної кваліфікаційної випускної роботи полягає в розробці веб-магазину, який після установки та розміщення його на хостингу, і в подальшому також заповнення його товарами та послугами почав би повноцінно виконувати поставлені перед ним мету, функції та завдання.

До таких особливостей можна віднести:

- можливість переглядати, створювати та редагувати категорії виробів;
- можливість редагування та обслуговування замовлення;
- право вибору видів та способів оплати товарів та послуг;
- право вибору видів, можливостей та способів доставки;
- наявність панелі управління для адміністратора веб-магазину.

Основними способами виконання завдання є дослідження та аналіз аналогічних проектів теми, вивчення та розробка інтернет-магазину з подальшою його програмною реалізацією на сучасних веб-технологіях.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 11 |

1.2 Поняття веб-магазину

Веб-магазин - це веб-сайт, який продає товари або послуги, отримує та обробляє заявки та замовлення на покупку різноманітних товарів та послуг, представляє користувачеві вибір типу для оплати, вид покупки замовлення, а також виставляє рахунок на оплату.

Веб-магазин переслідує наступні цілі:

- збільшення кількості клієнтів магазину та збільшення ринку;
- зниження трудовитрат продавців та менеджерів з продажу;
- знизити витрати на поставку ключових функцій продаж;
- зменшення кількості помилок та недоліків у звітності та збільшення швидкості її складання.

Проектований веб-магазин має забезпечити виконання всіх звичних для подібних проектів цілей та функцій, зокрема:

- можливість авторизації/реєстрації користувачів;
- можливість робити замовлення будь-якої кількості товарів та послуг;
- мати наявність категорій усіх товарів та послуг з пошуком за певними критеріями;
- кошик для покупок, куди будуть додаватись товари та послуги обрані користувачами;
- особистий кабінет користувача;
- реєстрація та управління кейсами, товарами та поставками для адміністрації сайту;
- особистий кабінет користувача з можливістю керувати своїм профілем та замовленнями;
- можливість мати зворотній зв'язок з допомогою коментарів та повідомлень про товар та про роботу інтернет-магазину загалом.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 12 |

1.3 Аналіз предметної області

З моменту створення Всесвітньої павутини пройшла велика кількість років. З плином часу багато що змінилося, зокрема збільшився технічний її потенціал, значно збільшилася інтернет-аудиторія, вирішилися поставлені завдання та мета, для вирішення яких мережа інтернет була створена. Інтернет виріз з існування тільки для наукових цілях та військових потреб. Зараз Інтернет це велике сховище інформації, де кожна людина, відповідно до своїх потреб, зможе знайти необхідний документ, фото, музику, відео, прозу та лірику, книги та інше. Також за допомогою мережі можна щоб придбати все необхідне людині – різноманітні товари, продукти харчування, програмне забезпечення, ігри, комп'ютерного обладнання та іншої техніки. В даний час більшість керівників і власників приватного бізнесу бачать, що необхідно розвивати своє невелике інтернет-офіси організації. Загалом все частіше з'являються різноманітні комерційні інтернет-майданчики, в яких основною метою яких є продаж товарів та послуг за допомогою інтернету.

На даний момент можна сказати, що конкуренція таких інтернет-магазинів та майданчиків з продажем через мережу інтернет є доволі складною, як зазвичай в звичайній торгівлі.

Зараз постає питання, що являє собою інтернет-магазин, як він створюється, розвиваються та функціонує в чому їх переваги та недоліки. Сам інтернет-магазин - це загалом сайт, який має свій каталог з певними категоріями товарів, описом та зображенням до них. Основна відмінність від звичаного онлайн-каталогу полягає в тому, що будь-який товар в інтернет-магазині можна не тільки переглянути, але і зробити його замовлення, не покидаючи свого місця та не відриватись від своїх справ на подорож звичайного магазину.

Зараз існують різні види та типи інтернет-магазинів:

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 13 |

- магазини, які продають товари та послуги певного вибраного асортименту;

- магазини, що продають товари різних типів та категорій;

- магазини великих торгових мереж, які є інтернет площадками для інших інтернет-магазинів.,

Проте всі інтернет-магазини світу мають певний визначений набір компонентів, зокрема:

- загальний спільний каталог з категоріями, де відображаються всі товари магазину. Загалом типи каталогів бувають різними - деревоподібними, мережевими, складними списками в його категоріях.

- процедура реєстрації користувача створює для кожного нового клієнта особисту «корзину», через яку можливість додавати обраний товар, або обрану послугу в неї, а потім оплачувати. У міру проходження по вітках каталогу також буде відстежуватись і побажання клієнта, на основі чого йому в подальшому можна будуть пропонувати не тільки товари з колекції цього магазину, а й архітектуру видачі супутніх товарів з цього каталогу. Для прикладу, якщо клієнт вказав телефон в своєму замовленні то система запропонує ненав'язливо йому переглянути аксесуари до нього, або ж тієї ж пральної машинки яка продається, запропонувати почитати коментарі тих, хто її купив раніше. Інтернет-магазин може надіслати повідомлення про те, що люди, які замовляли цей товар, також зацікавлені супутніми товарами та послугами до них -, при покупці мікрохвильової печі пропонується підібрати посуд для неї.

Система розрахунків за товари та послуги побудована так, що: клієнту дається на вибір використовувати різноманітні способи та види оплати товарів та послуг. Це може бути безготівковий розрахунок, оплата банківською картою, оплата криптовалютою чи оплата звичайною готівкою після доставки кур'єром або при отриманні товару на пошті.

Система доставки товарів і послуг в інтернет-магазинах таж пропонує великий вибір методів доставки, від відправки по e-mail, наприклад, програмне

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 14 |

забезпечення або ліцензійний ключ, доставка кур'єром компанії або іншою кур'єрською службою доставки та звичайною поштою.

Однак, незважаючи на такі універсальні можливості, інтернет-магазини все ж мають відмінність один від одного. Власник будь-якого інтернет-магазину прагне зробити інтернет-сторінку максимально комфортною для клієнта. Зазвичай, удосконалюючи способи пошуку в різних каталогах в іншій і системі. Також, як і в звичному простому магазині, в інтернет-магазині є можливість для влаштувань знижок і розпродажів. Головною відмінністю звичайного магазину від інтернет-магазину є можливість людині витратити менше коштів, а також можливість зробити покупку, не відволікаючись від важливих справ та не виходити для відвідування його з дому або робочого місця. Як бачимо завдяки здійсненню покупка в інтернет-магазині є досить людям вигідною.

Для повноцінної роботи, проектування розробки та створення веб-магазину не потрібно орендувати чи купляти нерухомість під магазин, а також робити ремонт, обслуговувати його, оформляти його, наймати обслуговуючий персонал, таких як продавці, касири, прибиральниці тощо. Цим самим ми завищуємо початкові витрати та а відповідно і собівартість самих товарів і послуг. В інтернет-магазині, навіть з урахуванням доставки товарі вартість його буде дешевшою, оскільки клієнт не буде оплачувати певну частину вартості товару, за яку продавець магазину буде намагається компенсувати свої витрати щомісяця обслуговування магазину і всього його персоналу.

Загалом інтернет-магазин має такі переваги перед звичайним магазином:

- дозволяє просту орієнтацію в каталогах та категоріях і прискорює сам процес пошуку необхідного товару або послуги в напрямках вартості, назва тощо;
- вибір товару або послуги, здійснити порівняння його характеристик з іншими, вартість, зовнішній вигляд та інші атрибути в порівнянні з іншими ;
- дозволяє переглядати інформацію про наявні знижки, акції, сезонні розпродажі та подібні заходи.
- дізнатися точну ціну замовлення разом з доставкою;

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 15 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

- обирати товар або послугу додаючи їх в кошик, здійснювати замовлення он-лайн, замовляти доставку товару додому, або інше місце;
- дозволяє підтримувати зв'язок через контакти між покупцем і продавцем, зокрема:
 - переглядати історію своїх попередніх замовлень;
 - переглядати інформацію про свої поточні замовлення.

Інтернет-аудиторія будь-якого інтернет-магазину не обмежується певним містом або районом де знаходиться магазин. Має свої стандарти обслуговування, обсяг інтернет-аудиторії не обмежений певною країною.

Незважаючи на чітко видимі і деякі явні технічні труднощі, створити інтернет-магазин простіше і економніше ніж звичайний. Також інтернет-магазин безсумнівно є відмінним доповненням до офф-лайн реклами. Головна перевага в тому, що покупцями є ті, у кого з певних причин немає часу та можливості відвідати звичайний магазин, і ті хто дізнався про привабливість придбання товару, не встаючи з місця зазвичай оцінить всі його переваги. .

1.4 Аналіз відомих проектів

Початковий аналіз нам показав, що типовою та необхідною інформацією при створенні інтернет-магазину, окрім програмної частини є власна архітектура інтернет-магазину, принцип роботи його модулів , оформлення дизайну та каталогу. Головна так важливо дізнатися інформацію про наявні каталоги товарів, їх місцезнаходження, а також новини, пов'язані з товарами постачальники та фірми, які пов'язані з цим товаром. Наявності різних онлайн-сервісів, як онлайн-оплата та замовлення доставки, посилання на контактну інформацію, інформацію про саму компанію та способи її діяльності. Крім усього перерахованого в інтернет-магазинах відображаються матеріали та публікації і новини, які пов'язані з

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 16 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

проданими товарами і послугами.

Розглянемо на прикладі одну із альтернатив - Інтернет-мазин Ебай (Рисунок 2.1). Цей інтернет-магазин працює з продажу товарів і послуг в мережі. В цьому інтернет-магазині продають різні товари люди та компанії з усього світу, але основну його частину становлять жителі США.

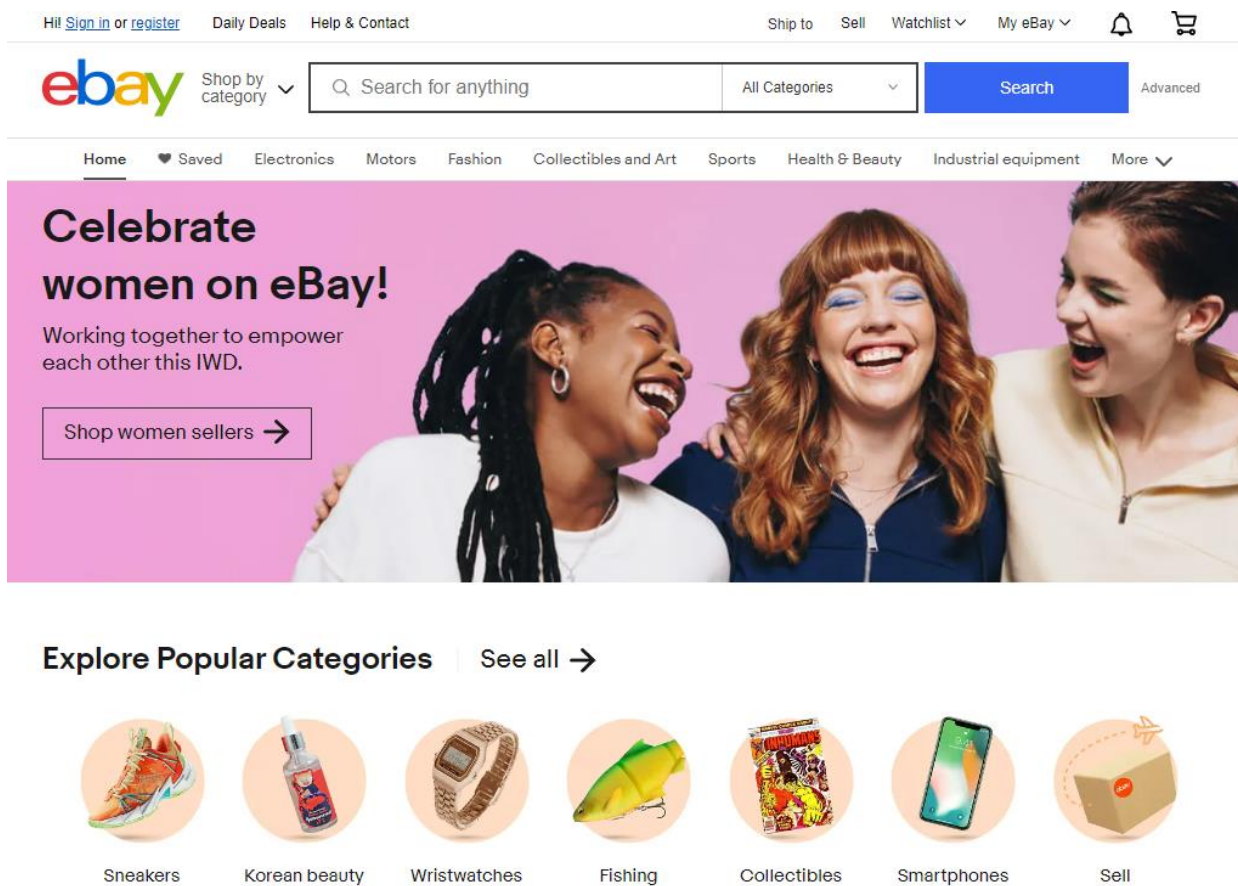


Рисунок 1.1 – Інтернет-магазин Ебай

У цьому інтернет-магазині виділяють наступні розділи:

- основний розділ містить нові і рекомендовані товари, перехід різні на категорії товарів наприклад: техніка, електроніка, мода, краса і здоров'я. Також є присутнім посилання для інших ресурсів, в тому числі для онлайн-сервісів);
- крім того, на сайті є інформація про сам сайт, контактні дані, різні способи оплати та вся необхідна для користувача магазину додаткова інформація, яка необхідна покупцеві;

| | | | | | | | | | |
|------|------|----------|--------|------|--|--|--|--|------|
| | | | | | | | | | Арк. |
| | | | | | | | | | 17 |
| Змн. | Арк. | № докум. | Підпис | Дата | | | | | |

КвРІПЗ. 180121.19.20

– в кожному каталозі є свої підкатегорії для зручності клієнтів, як в навігації так і пошуку необхідних товарів;

– посилання на авторизацію, розробників, карта сайту і так далі знаходяться в нижньому колонтитулі веб-сторінки. Даний інтернет-магазин був розроблений у 1995 році, в Сан-Хосе, Каліфорнія програмістом П'єром Омідьяром. Він створив інтернет-аукціон під назвою auctionWeb як частину його персонального веб-сайту.

Наступний розглянутий нами для аналізу веб магазин є великий і масштабний всесвітньо відомий інтернет-магазин - Aliexpress (Рисунок 1.2.)

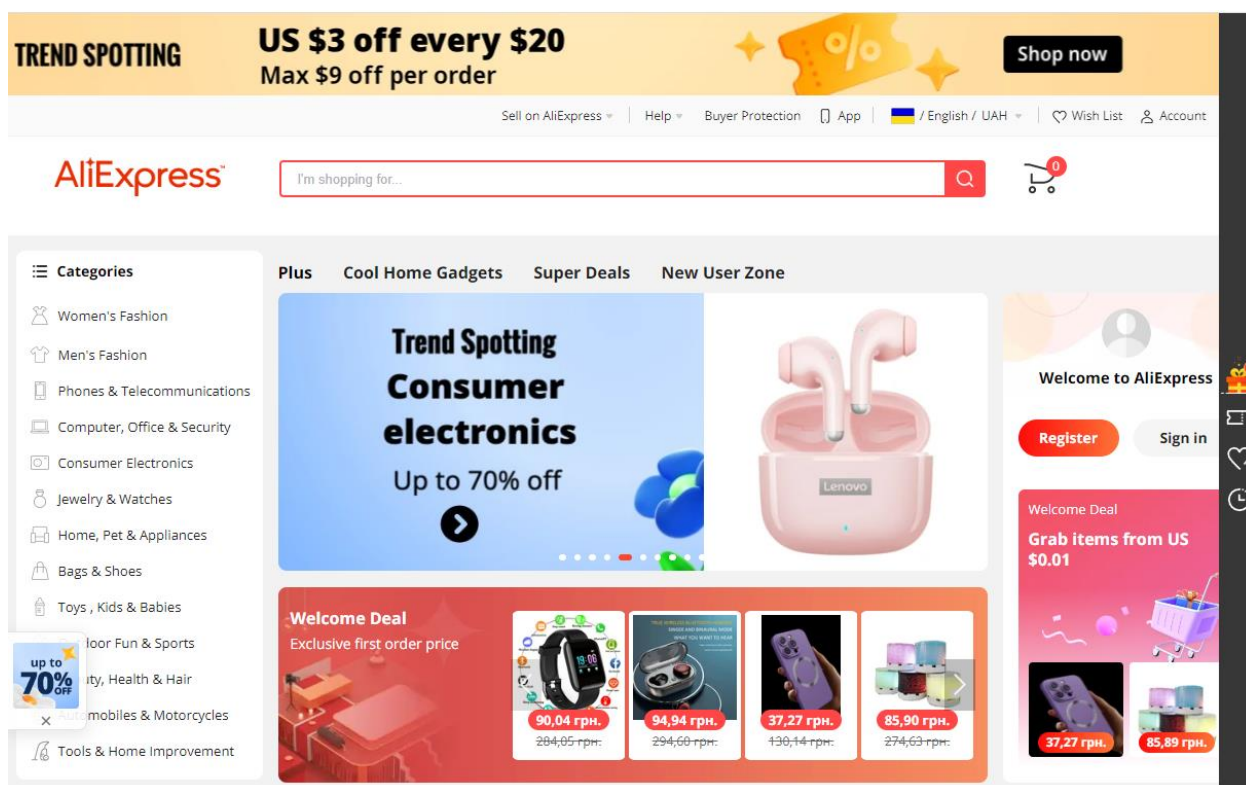


Рисунок 1.2 - Інтернет-магазин Aliexpress

Aliexpress є одним із найбільших інтернет-магазинів усього світу, який знаходиться в Китаї. Основне призначення даного інтернет-магазину, це продаж різних товарів та послуг. Розглянемо архітектуру та основні складові і елементи інтернет-магазину

– на головній сторінці магазину розміщене основне меню, куди входять різні тематичні та видові категорії та підкатегорії, а також рекомендовані та популярні різноманітні товари та послуги (Рисунок 1.2);

| | | | | | | | | | |
|------|------|----------|--------|------|--|--|--|--|------|
| | | | | | | | | | Арк. |
| | | | | | | | | | 18 |
| Змн. | Арк. | № докум. | Підпис | Дата | | | | | |

КвРІПЗ. 180121.19.20

– на нижньому колонтитулі головної сторінки є рекомендовані матеріали, навчання покупкам в інтернет, контактна інформація та інформція для зворотного зв'язку тощо;

– пункт меню для пошуку та вибору товарів, навігаційне меню для різних категорій товарів, при натисненні на яке клієнт може легко знайти потрібну йому категорію товару та сам товар чи послугу;

– також на головній сторінці міститься форма для реєстрації нових клієнтів, яким після авторизації буде доступний їх особистий кабінет та кошик з обраними товарами, та історія покупок якщо такі є.

Розробниками цього інтернет-магазину є засновник та власник магазину Джек Ма, який його засновав як Alibaba Group в 1999 році. Сьогодні ця компанія має власну систему електронних платежів з назвою Alipay, інтернет-магазини Taobao.com та магазин AliExpress.

1.5 Постановка задачі

Метою кваліфікаційної випускної роботи є розробка веб-застосунку для торгівлі виробами із шкіри.

Реалізація даної мети передбачає виконання ряду задач з дослідження, проектування та розробки застосунків:

- аналіз актуальності теми роботи;
- дослідження аналогів даного інтернет-магазину для виокремлення необхідних функцій сайту;
- обрання мови програмування;
- обрання засобів проектування;
- обрання засобів розробки сайту;
- обрання засобів обробки даних;

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 19 |

- проектування та розробка сайту;
- авторизацію користувачів у системі;
- адміністрування веб-сайту;
- пошукову система;
- перегляд та редагування даних.
- тестування та аналіз результатів;
- загальні висновки.

Веб-застосунок, що розробляється має бути простим у користування зі зручним та сучасним дизайном та інтерфейсом.

1.6 Висновки першого розділу

У даному розділі визначено мета і завдання створення веб-застосунку виробів зі шкіри, визначено поняття веб-магазину та його цілі, переваги використання інтернет-магазину. У розділі проведено аналіз предметної області проаналізовано подібні проекти. Здійснено постановку задачі та описано технічне завдання.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 20 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

2. Аналіз засобів проектування та розробка архітектури застосунку

2.1 Аналіз існуючих систем управління контентом розробки сайтів

Розроблюваний інтернет-магазин виробів зі шкіри, по-перше, це багатосторінковий веб-сайт з динамічними сторінками, які мають один дизайн, та зроблені за одним шаблоном оформлення і схожий інтерфейс. Нестандартні модальні сторінки сайту можуть бути виконані за іншими стандартами. Таким чином, для нашого проекту потрібна буде CMS-платформа, що дозволить розробити різні шаблони сторінок сайту автоматично та самостійно .

Розглянемо наступні найбільш поширені, актуальні і популярні системи сьогодення управління контентом сайту.

Joomla безкоштовна CMS для створення різних блогів, інтернет-магазинів, та інформаційних порталів. для написання програмного коду використовуються мови програмування PHP і JavaScript, СКБД MySQL або інші звичайні реляційні СКБД (Рисунок 2.1). Joomla вважається безкоштовним програмним забезпеченням яке розповсюджується за ліцензією GNU GPL. Дана система інтенсивно розвивається, має якісну документацію та службу підтримки різними мовами. Система управління контентом сайту Joomla добру розширюваність та відкриту систему. Вона має величезний вибір готових шаблонів до створення дизайну сайтів та інших рішень. Легкість, комфорт та зручність адміністративної панелі ситеми дає можливість легко створювати веб-сторінки, блоки, каталоги та наповнювати їх змістом, контентом, дає змогу організувати меню сайту і налаштовувати підключення різних сторонніх модулів.

Основні переваги цієї системи:

- вільна безкоштовна ліцензія;
- постійне оновлення системи;
- велика підтримка спільнот;
- добре документована на різних мовами;
- легкість в налаштуванні системи;

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 21 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

- можливість редагувати власні теми;
- можливість розробки та модифікації плагінів;
- можливість додавання метаданих для всіх веб-сторінок при розробці будь-якого типу сайту;
- загальний функціонал включає в себе управління правами користувача та можливість його переналаштування;
- публікація вмісту веб-сторінки різними мовами без застосування розширень та додаткових плагінів.



Рисунок 2.1 – Система управління контентом Joomla

Основні недоліки системи управління контентом Joomla:

- досить значне повільне завантаження веб-сторінки, яка містить велику кількість розширень і модулів;
- складність адміністрування та модернізації системи з досить великою кількістю розширень та модулів системи;
- відносно невелика кількість безкоштовних тем та розширень дає обмеження на додавання списку додаткових функцій.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 22 |

Наступна система WordPress систему управління контентом сайту з відкритим вихідним кодом. WordPress також використовує мову програмування PHP для розробки. Він випущений за ліцензією GNU GPL (Рисунок 2.5). Дана систему управління контентом сайту спочатку створювалася як система для створення інтернет-блогів, але потім багатфункціональні можливості значно поліпшились і це дозволило цій CMS отримати інструменти для розробки новинних сайтів, інтернет-магазинів та інших веб-сайтів довільної складності. WordPress дозволяє збільшити функціональність сайту за допомогою різних готових модулів, а також розширень які можна створити самостійно.

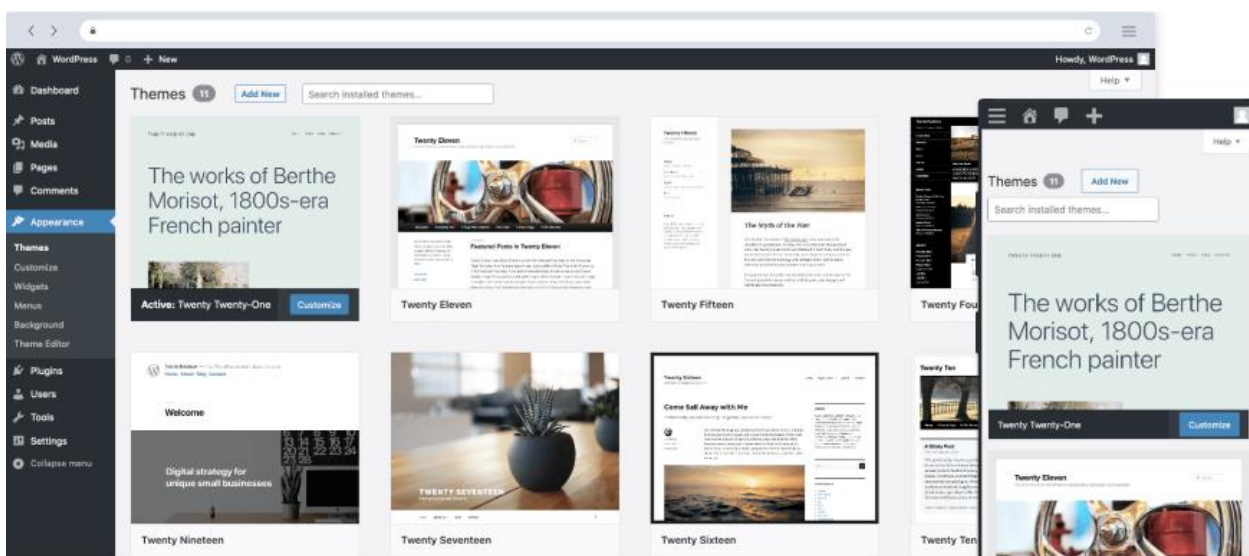


Рисунок 2.5 – Система управління контентом WordPress

Основні переваги систему управління контентом сайту WordPress:

- вільна ліцензія з постійними оновленнями;
- Добре документована та має службу супроводу на різних мовах;
- простота налаштування системи;
- досить зручний візуальний редактор;
- можливість створювати свої власні теми;
- доступна велика кількість безкоштовних тем, моділів та плагінів;
- можливість розробки нових та редагування наданих розширень, модулів та плагінів;

- система управління правами користувачів.

Основні недоліки системи Drupal:

- вузький набір основних функцій системи;
- велика частина спеціальних розширень і плагінів не переведена на різні мови.

Система управління контентом Drupal – CMS яка використовує мову програмування PHP для створення сайтів і використовує базу даних MySQL для обробки і зберігання даних (Рисунок 2.6). Система себе апрокомендувала як широкий, гнучкий та постійно вдосконалюємий функціонал.. Drupal захищений GPL та є вільним програмним забезпеченням. Його розробці якого сприяють програмісти міжнародного співтовариства шанувальників.

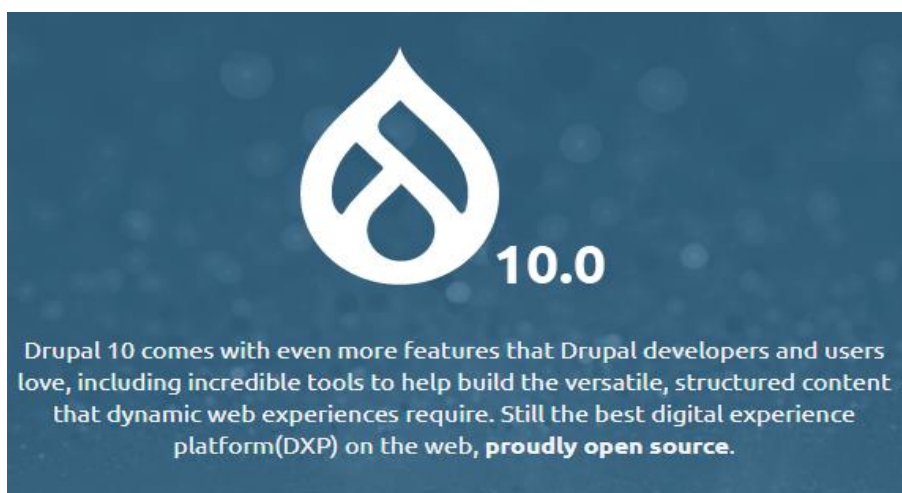


Рисунок 2.6 – Система управління контентом Drupal

Основні переваги системи:

- кросплатформність;
- канонічні URL-адреси є у ній по замовчуванню.
- велика кількість стандартних функцій для розробки різних типів сайту;
- досить гнучке управління привілеями користувачів.

Основні недоліки системи:

- труднощі вивчення для недосвідчених;

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 24 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

- значні недоліки в області кешування БД та сторінок;
- мала кількість якісних безкоштовних тем;
- для редагування метаданих потрібно стороннє розширення SEO.

2.2 Обґрунтування вибору системи управління контентом

На основі основних переваг аналізованої системи управління контентом , згаданої в попередньому пункті, а також із застосуванням дослідницької інформації, була створена порівняльна таблиця наявних систем управління контентом з використанням основного методу. Всі критерії отримали відповідні оцінки за бальною шкалою від 1 до 10, де 10 - найкраща оцінка, а 1 - найгірша оцінка за цим кожним критерієм .

Таблиця 1.2. - Порівняльний аналіз систем управління контентом

| Критерій | Joomla | Wordpress | Drupal |
|--|-----------|-----------|-----------|
| Ціна | Вільне ПЗ | Вільне ПЗ | Вільне ПЗ |
| Безпека | 7/10 | 7/10 | 9/10 |
| Розширення | 8/10 | 10/10 | 9/10 |
| Безкоштовні теми | 9/10 | 9/10 | 6/10 |
| Швидкість створення простого сайту | 7/10 | 10/10 | 5/10 |
| Швидкість створення багатофункціонального сайту | 4/10 | 9/10 | 6/10 |
| Вимоги до технічної сторони хостингу для модулів | 6/10 | 9/10 | 5/10 |

На підставі вищенаведеного порівняльного аналізу систем управління

контентом по ряду критеріїв можемо зробити висновок про те, що система WordPress є найбільш оптимальним засобом розробки інтернет-магазину. Ця CMS виграла за 5 критеріями з 7. Велика кількість різних розширень, плагінів і тем дасть можливість їй найкращим чином вирішити завдання розробки веб-магазину з можливістю створення, редагування та публікації товарів, включаючи авторизацію користувачів, та впровадження способів оплати і доставки, також новин в темах товарів і послуг, наявних в магазині. Впровадження персонального шаблону, який засновано на принципах використання безкоштовних шаблонів, дозволяє створити нам найбільш сучасний дизайн та інтерфейс для застосунку.

2.3 Опис системи управління контентом WORDPRESS

WordPress - платформи систем управління контентом, існуючі функції якої можна змінити за допомогою плагінів практично до невпізнання. У величезному списку плагінів є і ті, які призначені. Існує думка, що використання системи WordPress призначено для розробки інтернет-магазину, це питання досить спірне та має свої плюси і недоліки.

Незважаючи на стереотипи, що WORDPRESS призначено для сайтів блочного типу на цій платформі можна розробити практично всі типи сайтів, для різних потреб.

У проектах з розробки інтернет-магазинів найвідомішим рішенням є поєднання WordPress та плагіну Woocommerce.

Модуль Woocommerce безкоштовний і розробник має можливість протестувати цей плагін на звичайних темах WordPress.

Що відноситься до установки і настройки модуля Woocommerce то для нього є досить інформації та підручників. З інформації про плюси і недоліки реалізації інтернет-магазину з використанням модуля Woocommerce то така інформація відсутня. Розглянемо основні переваги даної CMS.

Головні переваги розробки інтернет-магазину на системі WordPress, з

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 26 |

плагіном Woocommerce полягають у наступному:

– простота і зручність монтажу, яка плягає всього лише у завантаженні та активації самого плагіну Woocommerce. , найменше це потрібно вставити в шаблон певну кількість рядків, які використовуються для активації підтримки модуля Woocommerce. Після таких дій інтернет-магазин активується та дасть можливість його налаштувати;

– у систем управління контентом WordPress є дизайнерські теми, які підтримують інтеграцію модуля-плагіна Woocommerce та досить велика кількість, безкоштовних і платних варіантів, вартість яких варіюється в межах 60-90 доларів;

– SEO оптимізація так само, як і в CMS WordPress, з цим пунктом у Woocommerce немає труднощів. Карта сайту, метатеги для будь-якої сторінки сайту індивідуально доцільність включення аналітики - все стає можливим;

– досить легкий і зручний процес додавання, редагування нових товарів та управління категоріями каталогу товарів;

– Використання каталогу категорій товарів, зображення товару і інших частин інтернет-магазину за допомогою Woocommerce є доступні та зрозумілі для більшості;

– якісний переклад багатомовних інтерфейсів для плагіна Woocommerce має, який вбудовані в систему мови;

– можливість розширення існуючих функцій плагіна Woocommerce для якого існує величезна кількість платних, а також безкоштовних доповнень.

– також плагін є вагомим аргументом для користувачів, особливо якщо користувач не зробив вибір на користь певної теми інтернет-магазину і немає досвіду просування.

Виходячи з вищесказаного, можемо стверджувати, що CMS WordPress разом з плагіном Woocommerce є оптимальним вибором для розробки довільного цільового інтернет-магазину.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 27 |

2.4 Опис плагіна WOOCOMMERCE

WooCommerce - це модуль-плагін, який використовується для роботи в системі управління контентом WordPress. В даний час даний модуль виступає одним з кращих рішень для програмістів, яким необхідно розробляти веб-сайти різного роду складності. Кількість завантажень модуля Woocommerce давно перевищує мільйони завантажень. WooCommerce має хорошу гнучкість та простоту у використанні. Практично усі хто ним хоче користуватись може легко його налаштувати через легкість і простоту установки.. Ще одна його перевага в великій кількості додаткових плагінів та модулів.

Велика частина інтернет-магазинів у світі працюють з підтримкою цього плагіна Woocommerce.

WooCommerce є безперечним на ринку подібних рішень, що підтверджує світовий рейтинг.

2.5 Розробка архітектури застосунку

Архітектура програмного забезпечення (англ. software architecture) - спосіб структурування програмної або обчислювальної системи, абстракція елементів системи на певній фазі її роботи. Система може складатись з кількох рівнів абстракції, кожен з яких може мати окрему архітектуру.

Дослідження архітектури програмного забезпечення намагається визначити як найкраще розбити систему на частини, як ці частини визначають та взаємодіють одна з одною, як між ними передається інформація, як ці частини розвиваються поодиночі і як все вищеописане найкраще записати використовуючи формальну чи неформальну нотацію.

Більша частина процесів по проектуванню програмних систем має на увазі

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 28 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

використання досвіду реалізації схожих проектів. Складно уявити систему, для реалізації якої не можна було б застосувати вже готові рішення або досвід, отриманий при їхньому створенні. Архітектурний стиль можна охарактеризувати як подібність у підходах до реалізації поставлених завдань, обумовлене досвідом. Він визначає перелік компонентів системи, способи й умови їхньої взаємодії. На жаль, всупереч безлічі спроб, не існує стандартних мов опису архітектур.

Архітектурні стилі поділяються на п'ять груп, що видно на рисунку 2.7:

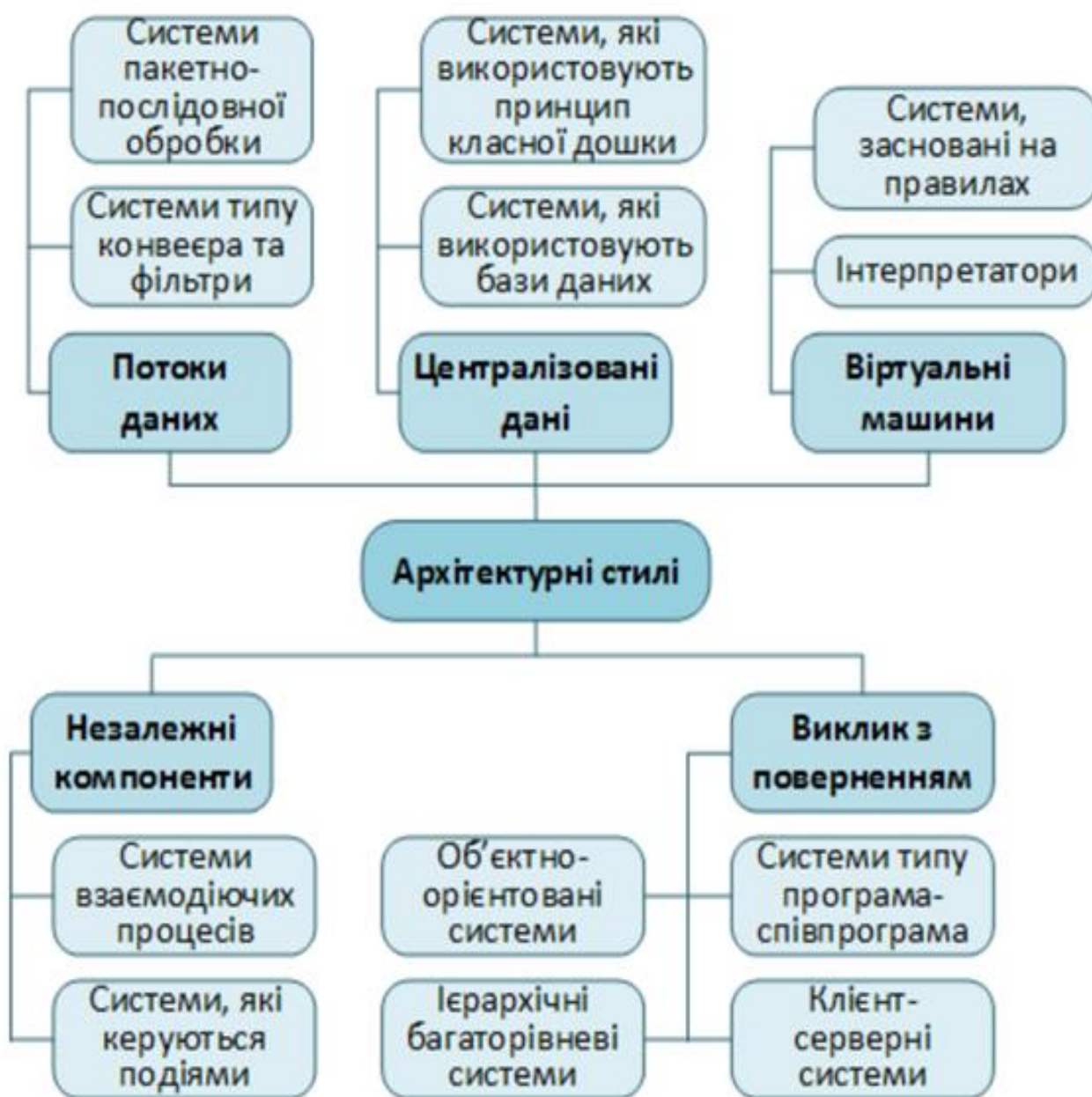


Рисунок 2.7 – Архітектурні стилі

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

Опишемо ці групи:

- потоки даних (Data Flow Systems);
- виклик з поверненням (Call-and-Return Systems);
- незалежні компоненти (Independent Component Systems);
- централізовані дані (Data-Centric Systems);
- віртуальні машини (Virtual machines).

Системи потоків даних, у свою чергу, поділяються на:

- системи пакетно-послідовної обробки (Batch Sequential Systems);
- системи типу конвеєри й фільтри (Pipe and Filter Architecture).

Оскільки ми обрали тип архітектури – клієнт-серверна архітектура, перейдемо до опису модулів проєктованої системи та подальшої їх декомпозиції. Головними структурними елементами у даній архітектурі є сервер та клієнт. Під сервером, з точки зору програмного проєктування, мається на увазі база даних та програмне забезпечення для роботи з нею, яке окрім цього буде проводити необхідні розрахунки та являтиметься посередником між базою даних та клієнтською частиною програми. У свою чергу, клієнт повинен містити інтерфейс для роботи з користувачем (в нашому випадку адміністратором (для адміністрування роботи магазину) і користувачем (для перегляду товарів і здійснення замовлення) та приймати дані від нього, проводити деяку валідацію та відправляти на сервер для подальшої обробки. Клієнтська частина програми має обмінюватись з серверною повідомленнями у одному з загальноприйнятих форматів: JSON, XML. Загальна модель архітектури подана на рисунку 2.8.

Перейдемо до виділення та опису окремих модулів системи. Для більшої наглядності створимо діаграми варіантів використання, що видно на рисунку 2.9, 2.10, 2.11, 2.12 та діаграми діяльності, яка зображена у додатку Б. На них можна побачити функції гостя, зареєстрованого користувача і адміністратора та процес зміни станів їх діяльності, що зображено на рисунку 6. Виділимо головні модулі системи. За роботу з замовленнями відповідатиме модуль замовлень. Для роботи з заповненням інформацією – модуль заповнення інформації і так далі.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 30 |

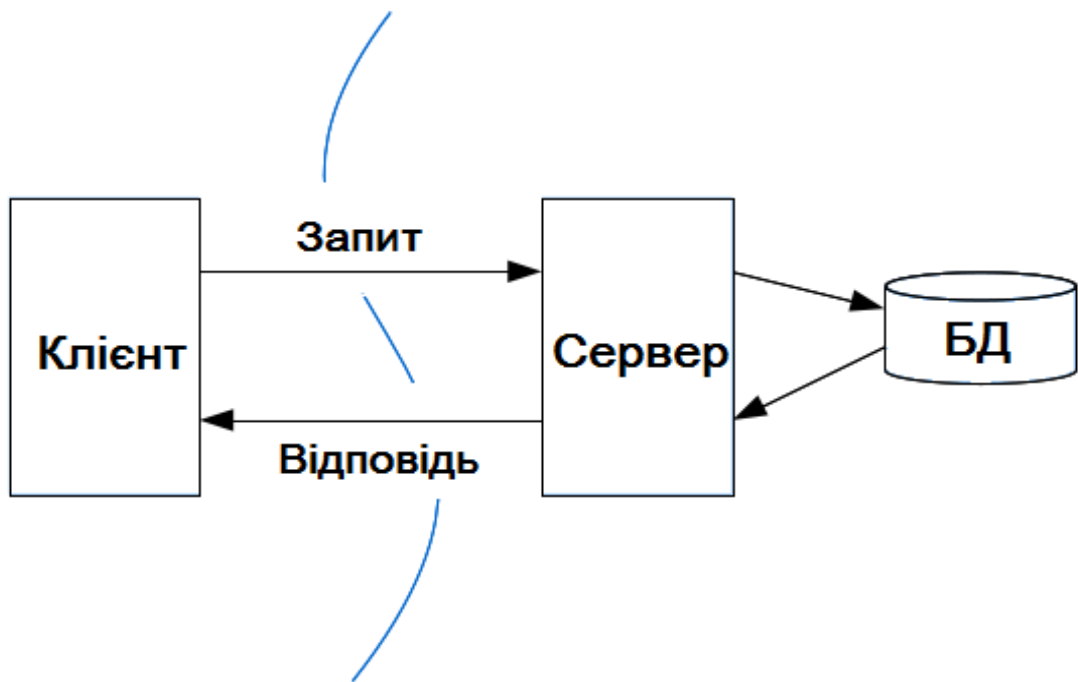


Рисунок 2.8 – Клієнт-серверна архітектура

Отже, перелік основних модулів проектованої системи:

- модуль для роботи з товарами;
- модуль для роботи з цінами;
- модуль для роботи з покупцями;
- модуль для роботи з замовленнями
- модуль оплати.

Проведемо декомпозицію модулів в рамках клієнт-серверної RESTархітектури. Серверна частина (рівень) програми буде складатись з наступних пакетів:

- model - entity);
- DAO - data access object;
- controller.

Пакет model відповідає за збереження тих чи інших моделей та бізнес-логіки пов'язаної з ними. DAO відповідає за роботу з базою даних, збереження тих чи інших моделей в базі та отримання з неї. Пакет controller має обробляти запити, які надходять від клієнтів та відповідати на них.

Клієнтська частина програми відповідає класичному шаблону проектування MVC та міститиме наступні пакети:

- model – модель даних;
- view – інтерфейс користувача;
- controller - модуль керування .
-

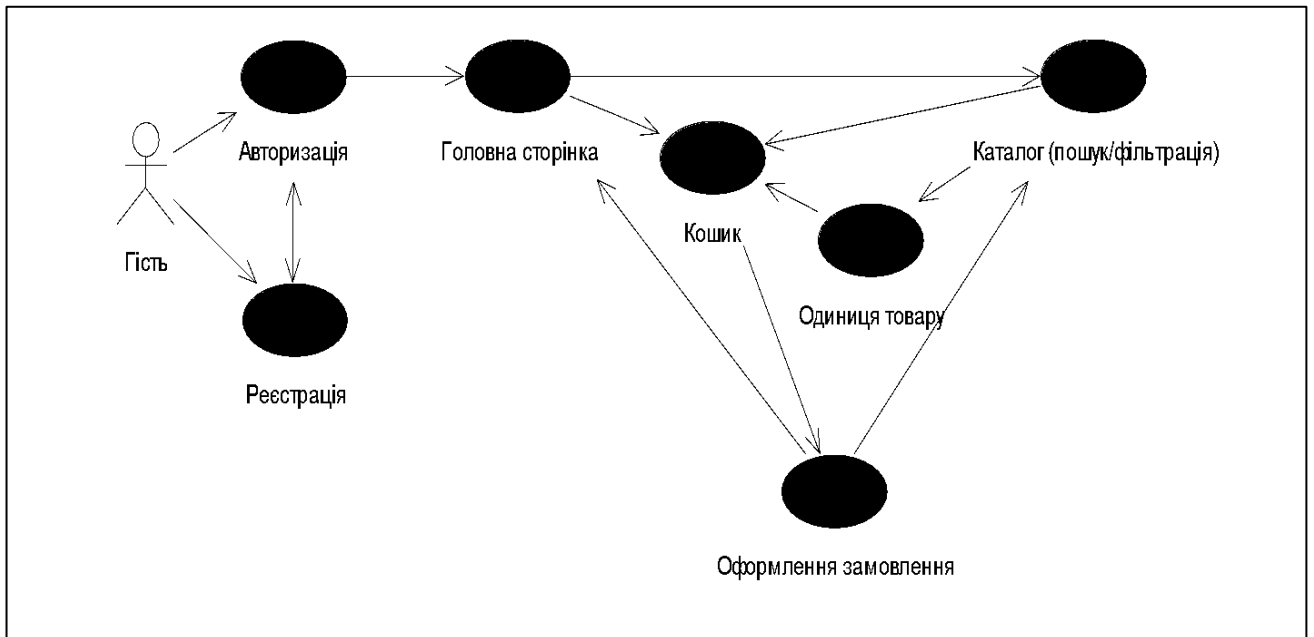


Рисунок 2.9 – Use case діаграма для гостя

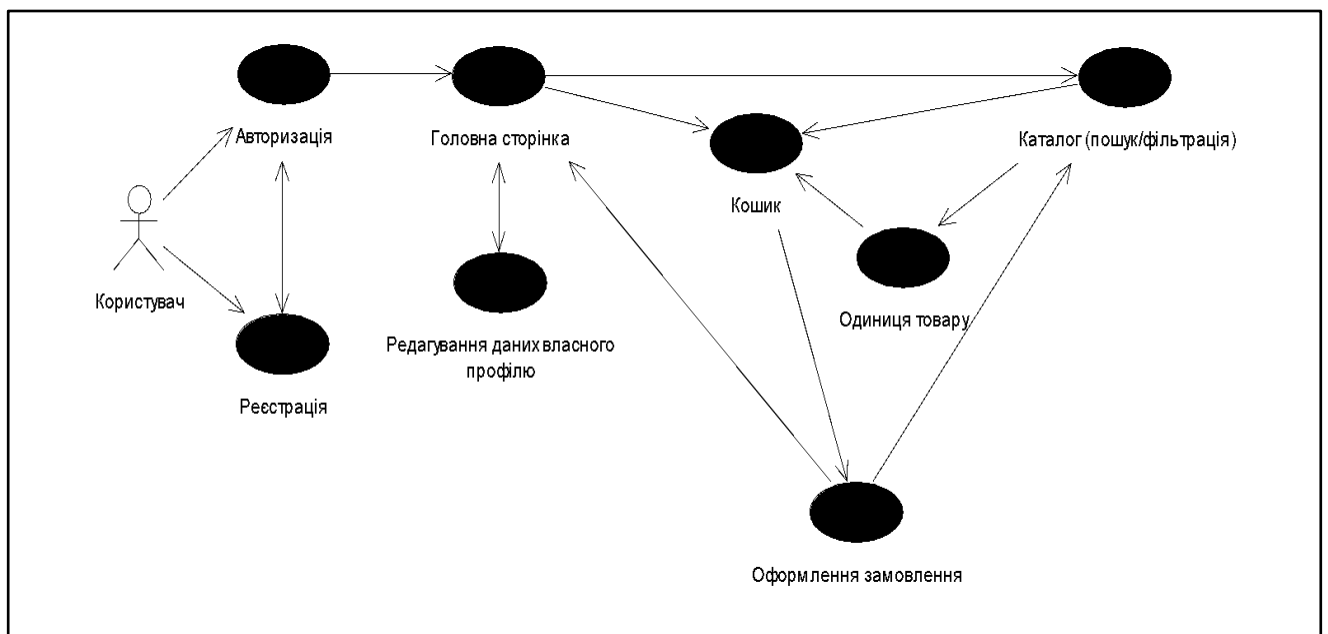


Рисунок 2.10 – Use case діаграма для користувача

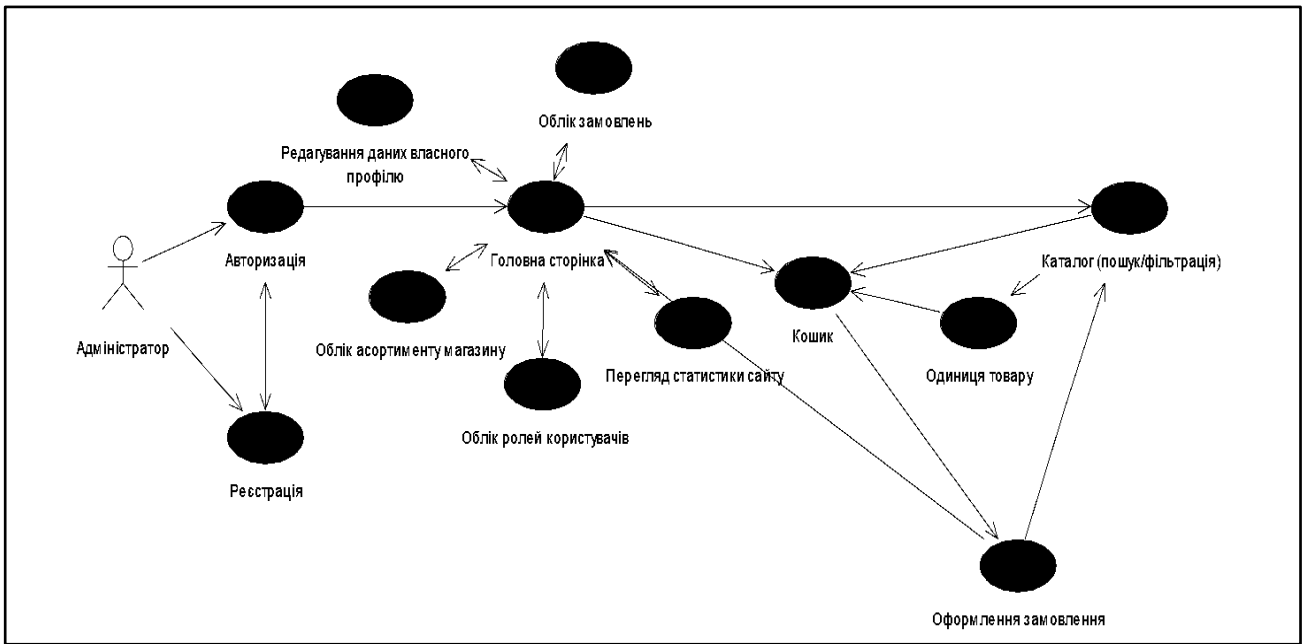


Рисунок 2.11 – Use case діаграма для адміністратора

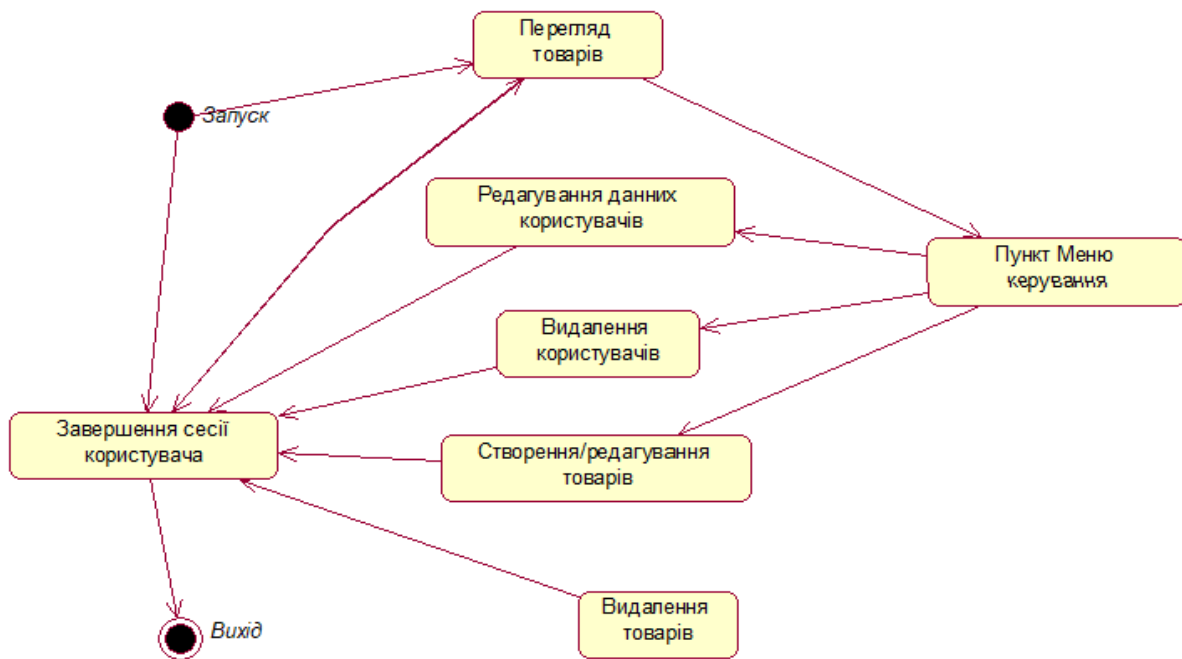


Рисунок 2.12 – Діаграма станів

У пакеті model мають зберігатись моделі та деяка додаткова логіка застосунку, яку недоцільно обробляти на сервері. Пакет view відповідає за відображення моделей користувачеві та взаємодії з ним. У пакеті controller будуть зберігатись класи, які використовуючи дані отримані за допомогою controller-а з сервера,

будуть наповнювати об'єкти з пакету model та надавати методи для роботи з ними, також міститимуть методи для відправки даних на сервер.

2.6 Висновки другого розділу

У даному розділі здійснено аналіз засобів проектування та розробка архітектури застосунку. Зокрема проведено аналіз існуючих систем управління контентом розробки сайтів, здійснено обґрунтування вибору системи управління контентом, **зроблено** опис обраної системи управління контентом WORDPRESS, опис плагіна WOOCOMMERCE, здійснено та розробку архітектури застосунку з представленням UML-діаграм.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 34 |

3 Програмна реалізація застосунку

3.1 Проектування та розробка даних для веб-застосунку

Після проведення нормалізації бази даних ми можемо переходити до побудови інформаційно-логічної моделі бази даних. Дана БД призначена для окресленої моделі предметної області, для якої розробляється наша база даних. На основі побудованої моделі буде проектуватись концептуальна модель нашої бази даних, а також фізична ER-діаграма. Вони використовуватимуться для демонстрації структури бази даних. Для побудови інформаційно-логічної моделі насамперед необхідно мати наявний список сутностей, список їх атрибутів, а також встановлені зв'язки між сутностями. Далі нам на початку необхідно встановити зв'язки між усіма таблицями бази даних.

Для сутності «Накладна» встановимо такі наступні атрибути, як Код накладної», «Код клієнта», «Код співробітника». Крім того, у накладній також мусить міститися така додаткова контактна інформація співробітника: ім'я, прізвище, по-батькові, номер пошти, встановлена ціна, спосіб оплати, передплата та дата замовлення. Атрибути в цьому відношенні мають містити відповідно: дані про саму накладну: ім'я, прізвище, по-батькові замовника, номер накладної пошти, за яким відстежується користувачем його замовлення, дата зроблення замовлення, передплата (якщо така є), яка відраховується пізніше від загальної суми ціни товару. Атрибут «Код накладної» повинен тут має бути унікальним. Він, відповідно, не має повторюватись, оскільки для кожного замовлення клієнта та його підтвердження має оформлятися накладна. Також у нашій базі даних є сутність «Замовлення», що містить такі атрибути: «Код замовлення», «код накладної», «код знижки», «код товару». Для цієї сутності ми додаємо атрибути: «кількість товарів у замовленні». Отже можемо бачити зв'язок між клієнтом та накладною, яка виражається в тому, що для кожного клієнта нашого магазину виписується накладна, в якій може міститися багато різних замовлень з

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 35 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

різноманітними товарами.

Таким чином зв'язок між сутностями «Накладна» та «Замовлення» відповідає типу зв'язку «один-до-багатьох». За аналогією, Користуючись таким визначеним способом визначаємо зв'язки між усіма іншими таблицями бази даних.

Побудована інфологічна модель бази даних представлена в рисунку 1 додатку А.

Після виконаних дій переходимо до занесення визначених сутностей та атрибутів в систему керування базою даних. Нами, як зазначалось, було обрано СКБД MySQL. В цій системі для встановлення типам атрибутам таблиць, будемо використовувати наступні типи даних :

- integer (int) – для задання цілих чисел;
- text – текстове поле змінної довжини;
- date – тип визначення задання дати;
- varchar – тип, який використовується для задання полів таблиці, у яких довжина постійна і встановлюється при створенні таблиць;

Опишемо спроектовані раніше таблиці та дані, які вони мають зберігати в базі даних.

Таблиця «Товар»: “Код товару”, “Код виду товару”, , “Розмір”, “Колір”, “Матеріал”, “Ціна”;

Таблиця «Вид товару»: “Код виду”, “Назва моделі”, “Виробник”,

Таблиця «Замовлення»: “Код замовлення”, “Код товару”, “Код накладної”, “Кількість товару”, “Код знижки для товару”;

Таблиця «Знижка»: “Код знижки для товару”, “Знижка”, “Термін дії для знижки”;

Таблиця «Накладна товару»: “Код накладної”, “Код клієнта”, “Номер відстеження товару”, “Загальна ціна товару”, “Код співробітника”, “Дата замовлення товару”, “Спосіб оплати товару”, “Передоплата”;

Таблиця «Клієнт»: “Код клієнта”, “Ім'я клієнта”, “Прізвище клієнта”, “По-батькові”, “Адреса”, “Телефон”, “Відділення пошти”;

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 36 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Таблиця «Співробітник»: “Код працівника”, “Ім’я”, “Прізвище”, “По-
батькові”, “Адреса”, “Номер телефону”, “Банківські реквізити”, “e-mail”;

Далі задамо типи даних для полів в цих таблицях.

Таблиця «Товар»:

- поле “Код товару” - тип даних Int;
- поле “Код виду товару” - тип даних Int;
- поле “Колір” - тип даних Text;
- поле “Розмір” - тип даних Varchar;
- поле “Матеріал” - тип даних Text;
- поле “Ціна” - тип даних Int.

Таблиця «Вид товару»:

- поле “Код виду товару” - тип даних Int;
- поле “Фірма моделі” - тип даних Text;
- поле “Назва моделі” - тип даних Text.

Таблиця «Замовлення»:

- поле “Код замовлення” - тип даних Int;
- поле “Код накладної” - тип даних Int;
- поле “Код товару” - тип даних Int;
- поле “Кількість” - тип даних Int;
- поле “Код знижки” - тип даних Int.

Таблиця «Знижка»:

- поле “Код знижки” - тип даних Int;
- поле “Знижка” - тип даних Int;
- поле “Термін дії для знижки” - тип даних Date.

Таблиця «Накладна»:

- поле “Код накладної” - тип даних Int;
- поле “Код клієнта” - тип даних Int;
- поле “Номер відстеження НП” - тип даних Int;
- поле “Загальна ціна” - тип даних Int;

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 37 |

- поле “Код співробітника” має тип даних Int;
- поле “Дата замовлення” - тип даних Date;
- поле “Спосіб оплати” - тип даних Text;
- поле “Передплата” - тип даних Int.

Таблиця «Клієнт»:

- поле “Код клієнта” - тип даних Int;
- поле “Ім’я” - тип даних Text;
- поле “Прізвище” - тип даних Text;
- поле “По-батькові” - тип даних Text;
- поле “Адреса” - тип даних Text;
- поле “Телефон” - тип даних Varchar;
- поле “Відділення пошти” - тип даних Int.

Таблиця «Співробітник»:

- поле “Код співробітник” - тип даних Int;
- поле “Ім’я” - тип даних Text;
- поле “Прізвище” - тип даних Text;
- поле “По-батькові” - тип даних Text;
- поле “Адреса” - тип даних Text;
- поле “Паспортні дані” - тип даних Text;
- поле “Телефон” - тип даних Varchar;
- поле “Банківські реквізити” - тип даних Varchar;
- поле “e-mail” - тип даних Text.

3.2 Створення та заповнення бази даних

Створювати базу даних будемо за допомогою PhpMyAdmin, де створим спроектовані сутності, з їх полями, типами та розмірністю даних.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 38 |

Наведемо структуру таблиці «client» з усіма встановленими типами та характеристиками - «Клієнт» (Рисунок 3.1).

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию |
|----------------------------|-------------|-------------|---------------------|----------|------|--------------|
| <input type="checkbox"/> 1 | code_client | int(11) | | | Нет | Нет |
| <input type="checkbox"/> 2 | first_name | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 3 | last_name | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 4 | middle_name | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 5 | address | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 6 | phone | varchar(20) | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 7 | vidd_np | int(5) | | | Нет | Нет |

Рисунок 3.1 – Створення таблиці «Клієнт»

SQL код створення таблиць наведено на прикладі створення таблиці «product» - «Товар».

```
CREATE TABLE `product` (`code_product` `INT(8)` NOT NULL,
`ID_code_product` `INT(8)` NOT NULL, `color` `TEXT` CHARACTER SET cp1251
NOT NULL, `material` `TEXT` CHARACTER SET cp1251 NOTNULL , `size`
`VARCHAR(5)` CHARACTER SET cp1251 NOTNULL, `price` `(10)` NOT NULL
) ENGINE = InnoDB;
```

Далі встановлюємо первинний ключ відношення для індексації та пошук, а також для зв'язку з іншими таблицями бази даних. :

```
ALTER TABLE `product`
ADD PRIMARY KEY (`code_product`).
```

Аналогічно створюємо інші таблиці нашої бази даних, створюємо сутності, поля та первинні ключі.

На рисунку 3.2 наведено структуру таблиці «Товар» з усіма встановленими типами та характеристиками.

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию |
|----------------------------|-----------------|------------|---------------------|----------|------|--------------|
| <input type="checkbox"/> 1 | code_product | int(11) | | | Нет | Нет |
| <input type="checkbox"/> 2 | ID_code_product | int(10) | | | Нет | Нет |
| <input type="checkbox"/> 3 | color | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 4 | material | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 5 | size | varchar(5) | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 6 | price | int(10) | | | Нет | Нет |

Рисунок 3.2 – Таблица базы данных «Товар»

На рисунку 3.3 наведено структуру таблиці «Знижка».

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию |
|----------------------------|-------------------|---------|-----------|----------|------|--------------|
| <input type="checkbox"/> 1 | code_discount | int(11) | | | Нет | Нет |
| <input type="checkbox"/> 2 | discount | int(10) | | | Нет | Нет |
| <input type="checkbox"/> 3 | validity_discount | date | | | Нет | Нет |

Рисунок 3.3 – наведено структуру таблиці «Знижка»

На рисунку 3.4 показано структуру таблиці «Накладна».

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию |
|----------------------------|----------------|-------------|---------------------|----------|------|--------------|
| <input type="checkbox"/> 1 | code_note | int(11) | | | Нет | Нет |
| <input type="checkbox"/> 2 | code_client | int(10) | | | Нет | Нет |
| <input type="checkbox"/> 3 | numb_np | varchar(30) | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 4 | total_price | int(10) | | | Нет | Нет |
| <input type="checkbox"/> 5 | code_worker | int(10) | | | Нет | Нет |
| <input type="checkbox"/> 6 | payment_method | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 7 | date_order | date | | | Нет | Нет |
| <input type="checkbox"/> 8 | prepay | int(10) | | | Нет | Нет |

Рисунок 3.4 – Таблица базы данных «Накладна»

На рисунку 3.5 наведено структуру таблиці «Замовлення».

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию |
|----------------------------|---------------|---------|-----------|----------|------|--------------|
| <input type="checkbox"/> 1 | code_order | int(11) | | | Нет | Нет |
| <input type="checkbox"/> 2 | code_note | int(10) | | | Нет | Нет |
| <input type="checkbox"/> 3 | code_product | int(10) | | | Нет | Нет |
| <input type="checkbox"/> 4 | number | int(10) | | | Нет | Нет |
| <input type="checkbox"/> 5 | code_discount | int(10) | | | Нет | Нет |

Рисунок 3.5 – Таблица базы данных «Замовлення».

На рисунку 3.6 наведено структуру таблиці «Код виду товару».

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию |
|----------------------------|-----------------|---------|---------------------|----------|------|--------------|
| <input type="checkbox"/> 1 | ID_type_product | int(11) | | | Нет | Нет |
| <input type="checkbox"/> 2 | firm_model | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 3 | name_model | text | cp1251_ukrainian_ci | | Нет | Нет |

Рисунок 3.6 – Таблица базы данных «Код виду товару»

На рисунку 3.7 наведено структуру таблиці «Працівник».

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию |
|-----------------------------|-------------|-------------|---------------------|----------|------|--------------|
| <input type="checkbox"/> 1 | code_worker | int(11) | | | Нет | Нет |
| <input type="checkbox"/> 2 | posada | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 3 | first_name | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 4 | last_name | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 5 | middle_name | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 6 | address | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 7 | passport | text | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 8 | phone | varchar(20) | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 9 | bank_rekv | varchar(50) | cp1251_ukrainian_ci | | Нет | Нет |
| <input type="checkbox"/> 10 | email | text | cp1251_ukrainian_ci | | Нет | Нет |

Рисунок 3.7 – Таблица базы данных «Працівник»

Після створення таблиць зв'язки між таблицями бази даних: (SQL наведений код у додатку В)

“Код товару” із таблиці «Товар» пов'язаний зв'язком «один до багатьох» із полем “Код товару” таблиці «Замовлення».

“Код виду товару” із таблиці «Вид товару» пов'язаний зв'язком «один до багатьох» із полем “Код виду товару” таблиці «Товар».

“Код знижки” із таблиці «Знижка» пов'язаний зв'язком «один до багатьох» із полем “Код знижки” таблиці «Замовлення».

“Код накладної” із таблиці «Накладна» пов'язаний зв'язком «один до багатьох» із полем “Код накладної” таблиці «Замовлення».

“Код працівника” із таблиці «Працівник» пов'язаний зв'язком «один до багатьох» із полем “Код працівника” таблиці «Накладна».

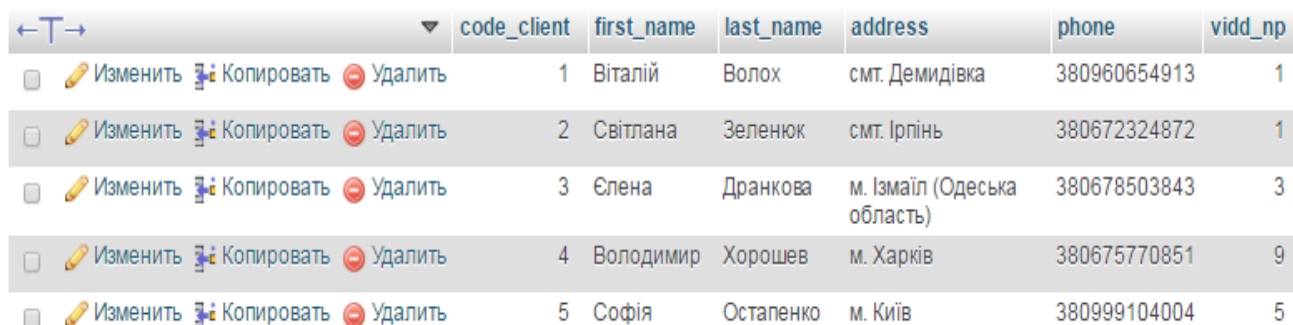
“Код клієнта” із таблиці «Клієнт» пов'язаний зв'язком «один до багатьох» із полем “Код клієнта” таблиці «Накладна».

Для заповнення таблиці бази даних скоритуємось SQL-кодом. На базі таблицю «Клієнт» продемонструємо цей код :

```
INSERT INTO `client` (`code_client`, `first_name`, `last_name`, `address`,  
`phones`, `vid_np`) VALUES ('8', 'Іван', 'Петровенко', 'м. Ужгород',  
'+80754775852', '1');
```

```
INSERT INTO `client` (`code_client`, `first_name`, `last_name`, `address`,  
`phones`, `vid_np`) VALUES ('9', 'Марія', 'Чурхай', 'м. Обухів', '380549112011',  
'5');
```

Таблиця «Клієнт» після заповнення наведена на рисунку 3.8.



| | code_client | first_name | last_name | address | phone | vid_np |
|--|-------------|------------|-----------|-----------------------------|--------------|--------|
| <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> | 1 | Віталій | Волох | смт. Демидівка | 380960654913 | 1 |
| <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> | 2 | Світлана | Зеленюк | смт. Ірпінь | 380672324872 | 1 |
| <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> | 3 | Єлена | Дранкова | м. Ізмаїл (Одеська область) | 380678503843 | 3 |
| <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> | 4 | Володимир | Хорошев | м. Харків | 380675770851 | 9 |
| <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> | 5 | Софія | Остапенко | м. Київ | 380999104004 | 5 |

Рисунок 3.8 – Таблиця “Клієнт”, після заповнення

Аналогічно заповнюються інші таблиці бази даних. Після створення всіх таблиць ми можемо переходити до створення застосунку та розробки його інтерфейсу.

Фізична модель бази даних наведена у додатку Б Рисунок 1

3.3 Розробка програмних модулів веб-застосунку

Логіка роботи нашого застосунку складається з двох частин, де одна частина розроблена для звичайного користувача, а друга для адміністрування веб-магазину.

В адміністративній частині застосунку доступні функції, які недоступні звичайному користувачеві, наприклад, додавання позиції товару, внесення змін а також видалення.

Одним з найважливіших для користувача функціоналом виступає зручне, доступне та зрозуміле в користуванні меню веб сайту, його інтерфейс з використанням, щоб кожен користувач сайту міг досить легко та швидко орієнтуватися в веб-магазині, пошук та вибору необхідних йому товарів.

Загалом головне меню, та, відповідно розроблений функціонал веб-застосунку має включати наступні розділи:

1. Пункт меню одягу та взуття із шкіри має містити такі категорії:

- куртки
- штани;
- головні убори;
- костюми;
- взуття;
- кросівки;

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 43 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

- ремені;
- туфлі;
- сумки;
- барсетки.

2. Пункт меню «Авторизація/Реєстрація» - вікно входу в систему;

3. Пункт меню «Про магазин» – звичайна веб-сторінка з текстом, таблицями та рисунками;

4. Пункт «Особистий кабінет» з інформацію про користувача, замовлені, куплені товари тощо;

5. Елемент «Кошик» - для замовлених товарів, та його обробки;

6. Пункт меню «Доставка та проплата товар» – інформаційна сторінка користувача про здійснення ним покупки.

Головною перевагою модульності системи є обробка використання низхідного процесу певного ступеня складності - від спільного та абстрактного, а також концептуального модуля нашої системи до більш конкретного в програмній системі. Зміст модуля полягає у використанні в : системі різних суб'єктів, об'єктів або сутностей, які взаємодіють між собою в застосунку. Взаємодією між модулями можуть виступати, як прямі зв'язки та і реакції інших модулів системи. На взаємодію модулів системи, та їх роботу загалом впливає реакція користувача застосунку, інших застосунків і також технічних пристроїв, а також вплив застосунку з деякої іншої системи обрав розробник. Варіанти використання модулів системи, у свою чергу вимагають опису наданих послуг системою, які надані їй користувачам. Можна стверджувати, що кожен такий випадок, який використовується системою - визначає деяку послідовність певних дій, які має виконувати застосунок після заданих користувачем умов.

Характеристика роботи модулів береться з діаграми варіантів використання та показана у таблиці 3.1. В даній таблиці наведено найбільш великі та використовувані модулі, опис характеристики менших модулів веб-застосунку аналогічний до наведених в таблиці

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 44 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Таблиця 3.1 - Характеристика модулів застосунку

| № | Назва модуля | Функції модуля |
|---|---|--|
| 1 | Авторизація | Функція обробки входу в веб-застосунок |
| 2 | Перегляд каталогу та його окремих позицій | Функція обробки дій користувача з позиціями сайту |
| 3 | Пошук позицій | Функція пошуку потрібного виробу зі шкіри серед всіх представлених позицій |
| 4 | Додавання знайденого виробу до кошика | Виконання внутрішніх функцій обробки товарів бази даних та оформлення замовлення |
| 5 | Зміна та оновлення інформації про вироби | Адміністративна функція можливості оновлення інформації про вироби, і можливість додавання і видалення позицій з бази даних магазину |
| 6 | Перегляд інформації про вироби замовлень | Функція відображення графічної та текстової інформації про вироби зі шкіри звичайним користувачем та адміністратором. має можливість переглянути замовлення покупців |
| 7 | Перегляд замовлень | Функція, яка надає можливість переглянути замовлення як користувачам так і адміністратору |
| 8 | Оформлення замовлень | Функція обробки даних у «кошику», здійснення фінансових розрахунків та оформлення документів для продажу/відправки товару |

Надзвичайно важливим механізмом не тільки нашого, а й будь-якого веб-

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 45 |

магазину - це функція, яка дозволяє користувачам магазину переглядати запропоновані вироби. В незалежності від того, шукає відвідувач певний виріб, або здійснює перегляд всіх виробів – це механізм повинен чітко працювати і бути для користувача процес простим в використанні та якомога зручним.

В нашому випадку достатньо зробити сортування виробів за категоріями. Це наведено у нижче представленому програмному коді.

```
public class CategoryActiv extends AppCompatActivity
{
private Button btn_vidimin;
// розділ чол
private ImageView vstntshirt,vshirt,vshood,vsweater;
private ImageView vpants,vsweatpants,vsportsute, viacket,vcoat,vshose;
////розділ жін

private ImageView wstshirt,whirt,wshood,wwearer; private ImageView
wpants,wsweatpants,wsportsute,wumpsuit; private ImageView wskirt,wdres,wiacket,wshose;
@Override
protected void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState); setContentView(R.layout.activity_category_actvity); init;
//розділ чол
wstshirt.setOnClickListener(new View.OnClickListener) {
@Override
public void onClick(View view)
{
Intent intent = new Intent (packageContext: CategoryActvity.this, FindPoCategory.class);
intent.putExtra name: "category", value: "mtshirt"); startActivity(intent);
}
});
wshirt.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view)
{
Intent intent = new Intent( packageContext: CategoryActvity.this,
FindPoCategory.class); intent.putExtra( names: "categorys", values: "wshirt");
startActivity(intent);
}
}
```

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 46 |

```
});  
}
```

Далі, для написання програмного коду рівня логіки застосунку, нам на початку необхідно проаналізувати усі можливі варіанти реалізації на дану рівні програмування.

Потрібно врахувати, що наші SQL-запити в програмному коді необхідно певним чином передавати MySQL для їх обробки. При цьому слід врахувати, що є два різних основних підходи для роботи з такими з'єднаннями з СКБД.

Перший підхід полягає у послідовність таких, які необхідно враховувати при кожному зверненні до СКБД:

- при необхідності виконати запит до СКБД потрібно відкрити з'єднання з базою даних;
- далі виконується запит SQL чи збережена процедура в процесі відкритого з'єднання і таким же чином отримуємо результат виконання цього запиту..
- після виконання запиту з'єднання з базою даних відразу закривається.

Основним недоліком такого підходу є досить великі витрати процесорного часу, передачі надлишкових запитів до СКБД при частому відкритті та закритті таких з'єднань.

В нашому застосунку ми використаємо другий підхід який має наступний алгоритм:

- разі необхідності звернення до СКБД для виконання потрібного запиту відкривається з'єднання з базою даних,.
 - з допомогою відкритого з'єднання здійснюються всі необхідні запити, збережені процедури, та тригери, при цьому не закриваючи з'єднання.
- Також на цьому етапі відбувається обробка виключених ситуацій, які виникли;

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | КвРІПЗ. 180121.19.20 | Арк. |
| | | | | | | 47 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

– при завершенні клієнтом роботи закривається з'єднання з СКБД та базою даних відповідно. У разі використання такого підходу всі операції з СКБД та базою даних при виконанні одного клієнтського запиту, який активізується, при маніпуляціях користувача на веб-сторінці нашого застосунку, відбувається через відкрите з'єднання з базою даних, тобто не відбувається відкриття та закриття з'єднання для кожної окремої операції. Такий підхід дає можливість використання цих постійних з'єднань для підвищення ефективності роботи застосунку.

Створення модулів нашого застосунку за таким принципом в програмному коді продемонструємо на модулях обробки даних користувачів та модулі перегляду та видалення товарів.

Для запобігання несанкціонованому доступу до функцій користувача-адміністратора, покажемо процес на створенні форми введення ідентифікатора та пароля, при цьому зробимо доступ до функцій адміністратора сайту лише тим користувачам, які мають певний ідентифікатор та пароль. Фрагмент коду цього модуля входу користувача наведений нижче.

```
private void ValidateUser(final String cmp, final String pwd)
{
    if (checkBoxRememberMe.isChecked)
    {
        Paper.tovr(). write (PrevSaveInf.UserAfumberKeys, cmp) ;
        Paper. tovr ().writeCPrevSaveInf.UsertxtPassKeys, pwd);
    }
    final DatabaseReference RootRef;
    RootRef = MySQLDatabase.getInstance() .getReference();
    RootRef.addListenerForSingleValueEvent(new ValueEventListener()
    {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot)
        {
            if (dataSnapshot.child(parentDbName).child(cmp).exists())
            {
```

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 48 |

```

        UseruserData =
dataSnapshot.child(parentDbName).child(phone).getValue(User.class);
if (usersData.getCmd().equals(phone)) {
if (usersData.getPassword().equals(password))
{
    if(parentDbName.equals("Users"))
    {
        loadingBar.dismiss();
        Toast.makeText(context LoginLeafActivity.this, text "Вдалий вхід",
        Toast.LENGTH_SHORT) .show();
        Intent homeIntent = new Intent( packageContext LoginLeafActivity.this,
        StorageActivity.class); Paper.book().write(PrevSaveInf.txtphonekey, cmd);
        startActivity(homeIntent);
    }
        else if(parentDbName.equals("Admins"))
        {
            loadingBar.dismiss();
            Toast.makeTextC context LoginLeafActivity.this, text "Вдало",
            LENGTH_SHORT) .show();
            Intent homeIntent = new Intent( packageContext:
            LoginLeafActivity.this, AdminPageActyvity.class); startActivity(homeIntent);
        }
    }
}
};

```

Фрагмент коду модуля перегляду, редагування та видалення виробів наведено нижче.

```

MySQLRecyclerAdapter<Product, ProductViewHolder> adapter =
    new MySQLRecyclerAdapter<~>(options)
    {
        «Override
        protected void onBindViewHolderCponNull ProductViewHolder holder, int
        position, @NonNull Products model)

```

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 49 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

```

{
holder.txtProductId.setText(model.getPid());
holder.txtProductName.setText(model.getPname());
holder.txtProductSize.setText(model.getPsize());
holder.txtProductPriceP.setText(model.getPPriseP());

Picasso.get().load(model.getImageQ).into(holder.imageView);
holder.itemView.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View p) {
CharSequence options [] = new CharSequence[]
{
"Редагування",
"Вставка",
"Видалення"
}
AlertDialog.Builder buil = new AlertDialog.Builder( context:
ProsmotrSpisanies.this); buil.setTitle("Виберіть варіант:");
buil.setItems(options, new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int i) { if (i == 0)
Intent intent = new Intent( packageContext ProsmotrSpisanies.this,
AddTovarActivity.class );
intent.putExtra( name: "Pid", model.getRidO): startActivity(intent);
}
if (i == 1)
{
ProductListRef.child(model.getRid())
removeValue().addOnCompleteListener(new OnCom tainer<Void>() {
@Override
public void onComplete((3NonNull Task<Void> task) ■{ if

```

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 50 |

```

(task.isSuccessfuK)H
Toast.makeText(. context: ProsmotrSpisanies.this, text "Видалення
виробу", Toast.LENGTH_SHORT) .showi();
}
}
};

```

3.3 Інструкція користувача

Щодо налаштування нашого веб-магазину, то розроблений нами веб-застосунок для торгівлі виробами із шкіри містить такі головні сторінки:

- index.shtml – головна сторінка сайту;
- catalog.shtml – сторінка для роботи з модулями по опрацювання каталогу та роботи з базою даних;
- modelsn.shtml – робота з модулями ситеми;

У папці css знаходяться основні таблиці стилів (CSS) усі зображення для роботи з сайтом знаходяться в папці img. Також у папці css розташовані CSS обраного шаблону.

У папці js знаходяться плагіни JavaScript. У проект вглючено плагіни: main.js, bootstrap.min.js, jquery.nicescrolljs.js, jquery.min.js, sly.min.js mixitup.js, owl.carousel.js тощо.

При заходженні користувача на сайт інтернет-магазину виробів із шкіри потрапляє на головну сторінку, яка зображена на рисунку 3.9. Як видно з рисунка сайт зроблено в мінімалістичному оформленні з контрастним дизайном на білому фоні для більш чіткого зображення виробів та концентрації на ньому уваги. На сайті присутні зліва меню «Магазин» з якого розкривається меню, для роботи з магазином, каталогами, виробами, інформацією про магазин тощо. З правого боку присутні динамічні гіперпосилання для зручності зареєстрованого клієнта.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 51 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Безкоштовна доставка всього!

Вироби із шкіри

Магазин

Мій рахунок

Перевірити



Ваша покупка змінює життя

Рисунок 3.9 - Головна сторінка сайту

Для пошуку необхідного виробу користувач задає пошук необхідного виробу за назвою, але найчастіше користується каталогом, який розбитий на категорії та підкатегорії відповідно за типами шкіряних виробів де можна обрати необхідні вироби (Рисунок 3.10).

При виборі конкретного виробу користувач бачить перед собою сторінку, яка зображена на Рисунку 3.11. На ній додається мінімум інформації для зосередження на ціні та покупці товару, а точніше додавання вибраного шкіряного виробу до кошика для подальшого його замовлення разом із іншими виробами, якщо покупець купує декілька виробів. Для більш детального ознайомлення з товаром користувач може перейти, клацнувши на фото, на наступну сторінку з детальною інформацією про товар, де також може оформити замовлення.

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

КвРІПЗ. 180121.19.20

Арк.

52

Безкоштовна доставка всього!

Вироби із шкіри

Каталог

Мій рахунок

Перевірити



Рюкзак



Сумка



Куртка



Барсетка



Ваша покупка змінює життя
Рисунок 3.10 - Вибір з виробів із каталогу

Безкоштовна доставка всього!

Вироби із шкіри

Сумки жіночі

Мій рахунок

Перевірити



Сумка жіноча
чорна

Ціна: 4999 грн

КУПИТИ

Ваша покупка змінює життя
Рисунок 3.11 - Вибір товару

| | | | | |
|------|------|----------|--------|------|
| | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата |


КвРІПЗ. 180121.19.20

Арк.

53



Крім вибору та оформлення заказів, для клієнтів в магазині передбачена можливість зворотнього зв'язку, як для побажання щодо асортиментів товарів та їх якості, так для клієнтів, які вже купили певний вибір і можуть звертатись до працівників магазину щодо купленого ними вибору (Рисунок 3.12). На цій формі зворотнього зв'язку клієнти можуть вносити свої відгуки, побажання, претензії щодо виробу, а також задавати питання щодо догляду за ними, ремонту тощо. Доцільно відмітити, що на формі присутні елементи які дозволяють не тільки висловити свої запитання чи пропозиції, а й розміщенні елементи, з допомогою яких клієнт може обрати та уточнити форми та способи контактування з ним, уточнити своє ім'я та прізвище, у випадку реєстрації його під псевдонімами. Окремо клієнт встановлює тему, для швидшої обробки професіоналами магазину та відповідними спеціалістами.

Безкоштовна доставка всього!



Каталог
Мій рахунок
Перевірити

Оновити / Форма контактів

Заповнити форму

Відправити

Ваша покупка змінює життя

Рисунок 3.12 – Форма відправки повідомлень (зворотній зв'язок)

| | | | | | | |
|------|------|----------|--------|------|----------------------|------|
| | | | | | КвРІПЗ. 180121.19.20 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 54 |

3.3 Тестування веб-застосунку та аналіз результатів

В інженерії розробки та проектування програмних застосунків на даний час виділяються такі основні види тестування веб-застосунків:

- Functionality testing (FT) - функціональне тестування програмних застосунків;
- UI testing (UIT) - тестування інтерфейсу користувача програмних застосунків;
- Usability testing (UT) - тестування зручності користування програмних застосунків;
- Performance testing (PT) - тестування продуктивності програмних застосунків;
- Security testing (ST) - тестування безпеки програмних застосунків.

Функціональне тестування визначається як тип тестування програмних застосунків, яке перевіряє зокрема, як працюють функції застосунку працює у відповідності до специфікації вимог. Даний тип тестування відноситься до типу «чорного ящика», тобто даний тип тестування не вимагає від тестувальника знання структури програмного забезпечення. Воно перевіряє різні аспекти, описані в документі специфікації вимог і специфікації функцій.

Функціональне тестування для тестування веб-застосунків нашого типу включає в себе:

- тестування коректності роботи гіперпосилань для виявлення некоректних посилань;
- тестування HTML-коду сторінок та шаблонів стилів для полегшення сканування сайту різноманітними пошуковими системами;
- тестування модальних користувацьких форм для виявлення некоректності заповнення полів та заповнення обов'язкових полів;

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 55 |

– Workflow перевірка – тестування робочих бізнес-процесів та тестування різних типів сценаріїв;

Тестування інтерфейсу користувача здійснюється, як процес тестування інтерфейсу застосунку т включає в себе тестування елементів управління:

- загального інтерфейсу
- меню сайту;
- гіперпосилань;
- кнопок,
- модальних вікон;
- сумісність з інтернет браузерями;
- виляд на різних типах пристроїв.

За допомогою тестування зручності користування перевіряються різні методи людино-машинної взаємодії людини. Основні пункти такого тестування полягають у тестуванні

- дизайну;
- навігації.
- простоти та зрозумілості користування.
- задоволеності від користування самим користувачем.

Тестування продуктивності здійснюється для перевірки прдуктивності роботи, перевірки навантаження на сервер, тествання пропускнуої здатності в мережі тощо. Також продиться так зване стрес-тестування при максимальних навантаження на веб-застосунок, зметою виявлення слабких технічних сторін сервера та роботи програмного коду.

Тестування безпеки, як тип тестування веб-застосунків перевіряє на вплив ризиків, вразливостей та загроз, які можуть привести до фінансових втрат та витоку конференційної інформації. Метою цього тестування є виявлення загроз та попередити їх. Це тестування дає змогу виявити можливі загрози у безпеці в дає рекомендації розробникам для усунення виявлених проблем у роботі веб-застосунку за допомогою введення додаткових елементів захисту.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 56 |

Під час проведення нами тестування помилок та збоїв в роботі веб-застосунку для торгівлі виробами із шкіри виявлено не було.

3.4 Висновок до третього розділу

В даному розділі було зроблено проектування та розробка даних для веб-застосунку, створено за здійснено початкове заповнення бази даних. В розділі проаналізовано і спроектовано основні категорії для меню сайту, спроектовано необхідні модулі реєстрація та авторизації користувачів у системі, модулі пошуку необхідного виробу, додавання та видалення товарів адміністратором. Здійснено механізм замовлення виробу та додавання цього виробу до кошика та оформлення замовлення.

Наведено процес розробки програмних модулів веб-застосунку з демонстрацією та поясненням реалізації програмного коду. Наведена інструкція користувача а прикладах роботи веб-застосунку.

Загалом показано, що програмний застосунок для веб-застосунку для торгівлі виробами із шкіри був розроблений так, щоб можна було забезпечити максимальну зручність та простоту у користуванні.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 57 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

ВИСНОВКИ

У ході виконання кваліфікаційної випускної роботи нами було визначено основні аспекти реалізації

Зокрема першому розділі даному розділі визначено цілі і завдання створення веб-застосунку виробів зі шкіри, визначено поняття веб-магазину та його цілі, переваги використання інтернет-магазину. У розділі проведено аналіз предметної області проаналізовано подібні проекти. Здійснено постановку задачі та описано технічне завдання.

У другому розділі здійснено аналіз засобів проектування та розробка архітектури застосунку. Зокрема проведено аналіз існуючих систем управління контентом розробки сайтів, здійснено обґрунтування вибору системи управління контентом, **зроблено** опис обраної системи управління контентом WORDPRESS, опис плагіна WOOCOMMERCE, здійснено та розробку архітектури застосунку з представленням UML-діаграм.

В третьому розділі було зроблено проектування та розробка даних для веб-застосунку, створено за здійснено початкове заповнення бази даних. В розділі проаналізовано і спроектовано основні категорії для меню сайту, спроектовано необхідні модулі реєстрація та авторизації користувачів у системі, модулі пошуку необхідного виробу, додавання та видалення товарів адміністратором. Здійснено механізм замовлення виробу та додавання цього виробу до кошика та оформлення замовлення.

Наведено процес розробки програмних модулів веб-застосунку з демонстрацією та поясненням реалізації програмного коду. Наведена інструкція користувача а прикладах роботи веб-застосунку.

У розділі показано, що програмний застосунок для веб-застосунку для торгівлі виробами із шкіри був розроблений так, щоб можна було забезпечити максимальну зручність та простоту у користуванні.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 58 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

Загалом можна сказати, що розроблений нами інтернет-магазин з дозволить власникам в загальному значно розширити кількість клієнтів, збільшити продажі товарів і послуг та дасть можливість скоротити час всього процесу торгівлі - від надання виробу до його продажу, поліпшити управління веб-магазином, знизити витрати на оренду та покупку торгових площі та наймання великої кількості обслуговуючого персоналу.

Розроблений нами веб-застосунок підтримує великі можливості щодо вдосконалення та покращення, зокрема планується додавання до веб-застосунку чат-ботів, підтримки взаємодії з соціальними мережами та зворотнього зв'язку з ними, що дуже є досить важливим в умовах сучасного розвитку суспільства та інтернет-технологій.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 59 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дипломний проект : методичні вказівки щодо його виконання для студентів спеціальності 121 «Інженерія програмного забезпечення» / Л. П. Бедратюк, Г.І. Радельчук, Ю. В. Форкун, О. М. Яшина. – Хмельницький : ХНУ, 2020. –
2. Brad Williams, David Damstra, Hal Stern. "Professional WordPress: Design and Development". Wiley, 2021. 552 p.
3. Brad Williams, Jeff Starr, Lisa Sabin-Wilson. "WordPress for Beginners 2023
4. A Visual Step-by-Step Guide to Mastering WordPress". O'Reilly Media, 2023. 576 стор.
5. Robbert Ravensbergen. "Building E-commerce Solutions with WooCommerce: Customize and extend WooCommerce for your online store". Packt Publishing, 2020. 470 p.
6. Lisa Sabin-Wilson. "WordPress All-in-One For Dummies". For Dummies, 2022. 912 p.
7. Patrick Rauland, Raúl P. García. "WooCommerce Cookbook". O'Reilly Media, 2020. 384 p.
8. Yannick Lefebvre. "WordPress Plugin Development Cookbook". Packt Publishing, 2019. 358 p.
9. Stephen Burge. "WooCommerce Explained: Your Step-by-Step Guide to Building an E-commerce Store with WooCommerce". OStraining, 2021. 352 p.
10. Stephanie Leary. "WordPress for Web Developers: An Introduction for Web Professionals". O'Reilly Media, 2021. 260 p.
11. David Trounce. "WooCommerce Made Easy: A Step-by-Step Guide to Setting Up an Online Store". CreateSpace Independent Publishing Platform, 2019. 224 p.
12. Sufyan bin Uzayr. "Mastering WooCommerce: Build a Professional Online Store with WordPress and WooCommerce". Packt Publishing, 2020. 320 p.
13. Karol Krol. WordPress 5 Complete: Build beautiful and feature-rich websites from scratch. Packt Publishing, 2018. 616 p.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІПЗ. 180121.19.20</i> | Арк. |
| | | | | | | 60 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

14. Brad Williams, David Damstra, Hal Stern. "Professional WordPress: Design and Development". Wiley, 2021. 552 p.
15. Eric Schwartzman. "WooCommerce Tutorial: Build an E-commerce Website". Independently published, 2022. 152 p.
16. Vladimir Prelovac. WordPress Plugin Development: Beginner's Guide. Packt Publishing, 2019. 208 p.
17. Paul Thewlis. WordPress for Business Bloggers. Apress, 2022. 342 p.
18. Florian Kiersch. WooCommerce Explained: Master WooCommerce Plugin: A Step-by-Step Tutorial. Independently published, 2020. 208 p.
19. Patrick Rauland. WooCommerce Explained: Your Step-by-Step Guide to WooCommerce". WooExperts Press, 2021. 346 p.
20. Lisa Sabin-Wilson. "WooCommerce For Dummies. For Dummies, 2019. 416 p.
21. Brad Williams, David Damstra, Hal Stern. Professional WordPress: Design and Development. Wiley, 2021. 552 p.
22. Christian Schmullius - Software Architecture Handbook. Independently published, 2020, 210 p.
23. Titus Winters, Tom Manshreck, Hyrum Wright - Software Engineering at Google: Lessons Learned from Programming Over Time. O'Reilly Media, 2020, 752 p.
24. Lee Atchison - Architecting for Scale: High Availability for Your Growing Applications. O'Reilly Media, 2020, 284 p.
25. Andrew Hunt, David Thomas - The Pragmatic Programmer: Your Journey to Mastery. Addison-Wesley Professional, 2019, 352 p.
26. Joseph Ingeno - Software Architecture Handbook. Independently published, 2020, 132 p.
27. Gregor Hohpe, Bobby Woolf - Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional, 2023, 736 p.
28. Christoph Fehling, Frank Leymann, Ralph Retter, Walter Schupeck - Cloud Computing Patterns: Design Patterns for Cloud Services. Springer, 2021, 528 p.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|-------------|
| | | | | | <i>КвРППЗ. 180121.19.20</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | <i>61</i> |

29. Chris Richardson - Microservices Patterns: With examples in Java. Manning Publications, 2020, 488 p.
30. Steve 'ardave' Albin - Software Architecture for Developers. Independently published, 2020, 116 p.
31. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom - "Database Systems: The Complete Book". Pearson, 2020, 1440 p.
32. Raghuram Ramakrishnan, Johannes Gehrke - "Database Management Systems". McGraw-Hill Education, 2020, 1360 p.
33. Anthony Molinaro - "SQL Cookbook: Query Solutions and Techniques for Database Developers". O'Reilly Media, 2019, 576 p.
34. Martin Kleppmann - "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems". O'Reilly Media, 2017, 614 p.
35. Pramod J. Sadalage, Martin Fowler - "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence". Addison-Wesley Professional, 2018, 192 p.
36. Alex Petrov. Database Internals: A Deep Dive into How Distributed Data Systems Work. O'Reilly Media, 2020, 402 p.
37. S. K. Singh. Database Systems: Concepts, Design and Applications. Pearson, 2020, 824 p.
38. Anthony DeBarros. Practical SQL: A Beginner's Guide to Storytelling with Data. No Starch Press, 2018, 504 p.
39. Seyed M. M. Learning MySQL: Get a Handle on Your Data. O'Reilly Media, 2020, 472 p.
40. Nathan Marz, James Warren. Data Management at Scale: Lessons Learned from Building Large Distributed Systems. O'Reilly Media, 2020, 200 p.
41. Shannon Bradshaw, Kristina Chodorow. MongoDB: The Definitive Guide: Powerful and Scalable Data Storage. O'Reilly Media, 2019, 448 p.
42. Viktor Mayer-Schönberger, Kenneth Cukier .Big Data: A Revolution That Will Transform How We Live, Work, and Think. Eamon Dolan/Houghton Mifflin Harcourt, 2019, 256 p.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|------|
| | | | | | <i>КвРІІЗ. 180121.19.20</i> | Арк. |
| | | | | | | 62 |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

43. Michael Kifer, Arthur Bernstein, Philip M. Lewis, Leonidas Fegaras .Database Systems: An Application-Oriented Approach. Pearson, 2021, 1000 p.

44. Baron Schwartz, Peter Zaitsev, Vadim Tkachenko .High Performance MySQL: Optimization, Backups, and Replication. O'Reilly Media, 2018, 828 p.

| | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|-----------------------------|-------------|
| | | | | | <i>КвРІІЗ. 180121.19.20</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 63 |

ДОДАТОК А
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Робота виконується в рамках проекту розробки веб-застосунок для торгівлі виробами із шкіри. Технічне завдання розроблено у відповідності до стандарту ДСТУ 7363:2013.

1 Підстава для розробки

Підставою для розробки є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри інженерії програмного забезпечення. Найменування розробки: Програмний модуль запису на прийом до лікаря приватної клініки.

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням додатку є клієнтська та серверна частина для веб-магазину, та серверна частина з можливістю редагування вмісту сайту.

2.2 Експлуатаційне призначення

Програма повинна експлуатуватися на будь-яких пристроях, на яких є браузер. Кінцевим користувачем додатку може виступати будь-яка особа.

3 Вимоги до програми

3.1 Вимоги до функціональних характеристик

Для звичайного користувача:

- перегляд інформаційних блоків про компанію;
- перегляд виробів за шкіри;
- покупка виробів за шкіри;
- можливість здійснення зворотнього зв'язку з магазином
- окремі сторінки виробів зі шкіри;

Для адміністратора:

- авторизація;
- створення ролей адміністраторів з відповідними доступами до редагування вмісту сайту;
- можливість створення та редагування блоків про вироби;
- можливість додавання виробів;

- можливість завантаження фото/відео на веб-сайті;
- можливість створення та редагування окремих сторінок виробу з необхідною інформацією;
- можливість оновлювати дані про вироби;
- можливість створювати контент;
- можливість опрацювання усіх вхідних онлайн-звернень з веб-сайту;
- система обліку усіх вхідних заявок;
- можливість редагування усього контенту сайту.

3.2 Вимоги до надійності

Веб-додаток повинен забезпечувати такі вимоги до надійності:

- обробляти невірні дії користувача і попереджати його про можливі наслідки;
- можливість самостійно відновлюватись у разі збою;
- можливість резервного копіювання бази даних.

3.3 Умови експлуатації та вимоги до технічних засобів

Веб-додаток повинен працювати на всіх пристроях, які мають браузер та стабільний доступ до мережі «Інтернет»: смартфонах, планшетах та комп'ютерах. Браузер може бути будь-яким: Edge, Safari, GoogleChrome, Opera, MozillaFirefox, тощо.

Для роботи платформи без збоїв, потрібно, щоб пристрій, на якому вона запускається задовольняв наступні мінімальні вимоги:

- 100 Гб внутрішньої пам'яті;
- 2 Гб оперативної пам'яті;
- 2-ядерний процесор;
- доступ до мережі «Інтернет» зі швидкістю мінімум 10 Мбіт/с.

3.4 Вимоги до інформаційної та програмної сумісності

Для розробки веб-застосунку було використаний фреймворк WordPress, фронтенд JavaScript фреймворк Vue.js, бекенд PHP, та база даних MySQL.

3.5 Спеціальні вимоги

Програма повинна мати зручний та зрозумілий зовнішній інтерфейс користувача.

4 Вимоги до програмної документації

У момент здачі проекту замовнику надається наступний набір документів:

- текст програми;
- опис програми;
- технічне завдання;
- керівництво користувача.

5 Стадії та етапи розробки

Стадії та етапи розробки веб-застосунку для автоматизації клієнто-орієнтованих бізнес-процесів веб-магазину А.1.

Таблиця А.1 – Стадії та етапи розробки проекту

| Стадія розробки | Етапи робіт | Зміст робіт |
|--|--|---|
| 1 | 2 | 3 |
| Технічне завдання 02.01.23 – 31.01.23 | Обґрунтування необхідності розробки програми | Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання |
| Ескізний проект 01.02.23 – 26.02.23 | Розробка ескізного проекту | Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури |

Кінець таблиці А.1

| 1 | 2 | 3 |
|---|---|--|
| Технічний проект 29.02.23 – 19.03.23 | Розробка технічного проекту | Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми |
| Робочий проект 20.03.23 – 15.04.23 | Розробка програмного забезпечення | Реалізація програмного забезпечення; відлагодження; проведення попереднього тестування |
| Розробка програмної документації 16.04.23 – 22.04.23 | Розробка документації до програмного забезпечення | Розробка необхідної документації, передбаченої технічним завданням |
| Тестування | Проведення | Розробка методики тестування; |

| | | |
|---|--|---|
| системи 23.04.23 – 30.04.23 | тестування програмного забезпечення | проведення основних тестів; коректування програмного забезпечення |
| Впровадження | Підготовка і передача програми | Підготовка і розгортання програмного забезпечення |
| Розробка програмної документації 16.04.23 – 22.04.23 | Розробка документації до програмного забезпечення | Розробка необхідної документації, передбаченої технічним завданням |

6 Порядок контролю та приймання

Контроль здійснюється кінцевими користувачами системи, підключеними на етапі тестування додатку.

ДОДАТОК Б
(обов'язковий)
ДІАГРАМИ

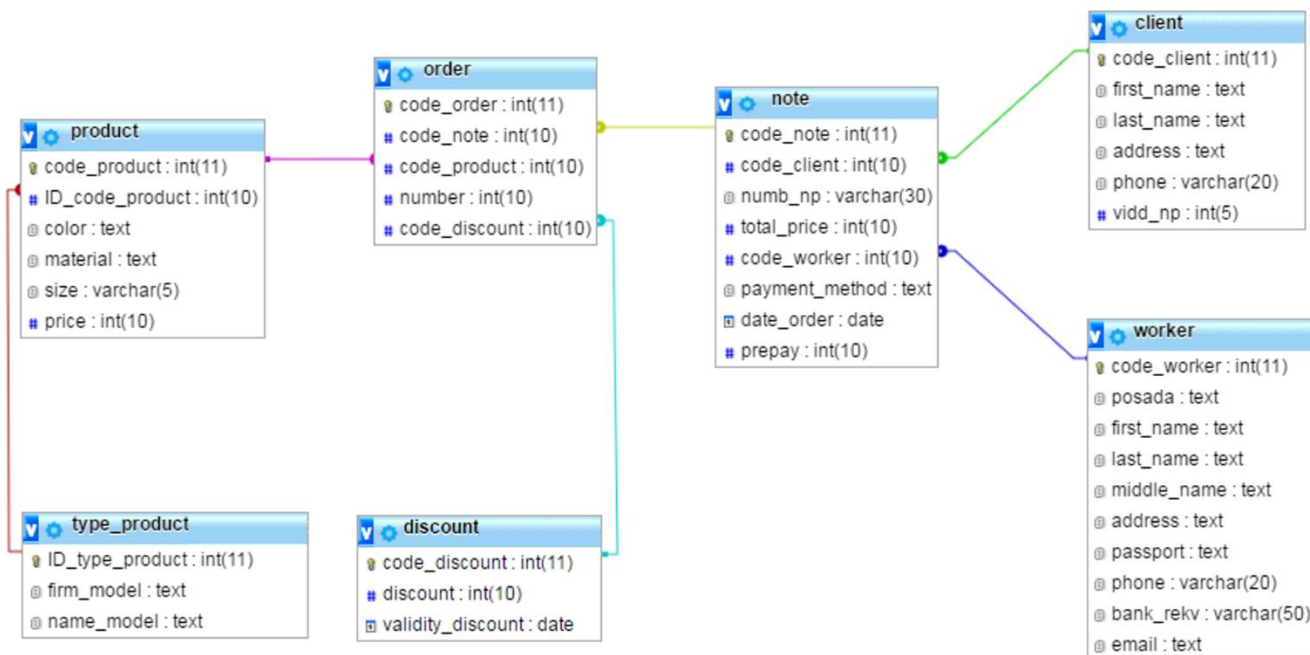


Рисунок 1. - Даталогіна модель

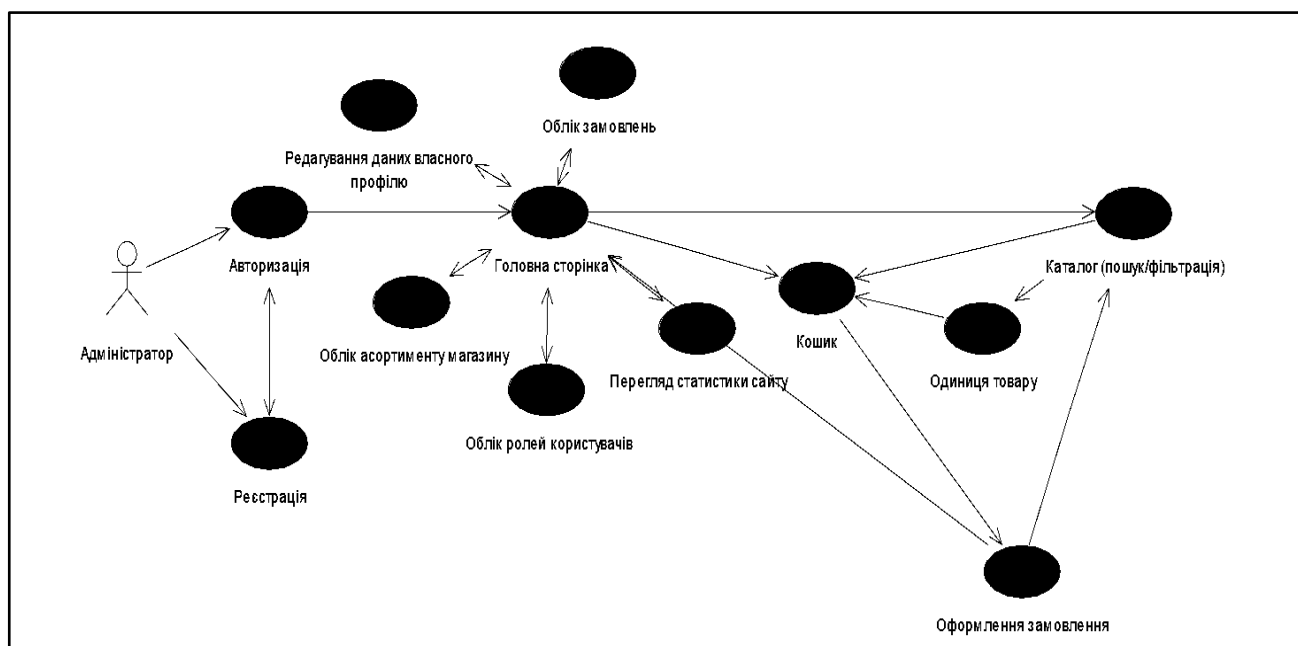


Рисунок 2 – Use case діаграма для адміністратора

ДОДАТОК В
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Кваліфікаційна робота

«Веб-додаток для торгівлі виробами із шкіри»

Виконав: студент IV курсу, групи ІПЗ-19-1 Данькевич Н.В.

Керівник: канд. тех. наук, доцент Форкун Ю.В.

Вступ

У сучасному світі, де у людини практично постійно не вистачає часу, а іноді і фізичних зусиль, на відвідування різних магазинів, а також тривалі і виснажливі подорожі по ним. Одним з варіантів є купівля необхідного в інтернет-магазинах де можна обрати необхідну річ. Майже усі сучасні інтернет-магазини, дозволяють підібрати певний товар для доставки, так щоб клієнт мав право оцінити річ та вибрати саме таку яку йому потрібно. Зараз у світі існує велика кількість різноманітних інтернет-магазинів, як для певного роду товарів, так і загального спрямування: Ebay, AliExpress, Розетка, Алло тощо. Величезний вибір інтернет-магазинів, вузького спрямування, такі як: електротехніки, господарські магазини, магазини одягу, спортивні магазини, для залучення клієнтів організують різноманітні акції, проводять різні рекламні та заохочувальні кампанії. При успішній рекламній кампанії та маркетингу вони мають прибуток кращий ніж звичайні магазини.

Таким чином, стає актуальною задачею для нашого завдання є розробка інтернет-магазину виробів із шкіри.

Мета розробки

Метою нашого завдання є розробка інтернет-магазину на базі платформи Wordpress для торгівлі виробами із шкіри в інтернеті.

Завданнями дипломного проекту для реалізації нашої мети є:

- 1) Аналіз і опис предметної області;
- 2) Постановка завдання на створення інтернет-магазину;
- 3) Розробка методів і способів вирішення завдання для створення інтернет-магазину;
- 4) Підбір програмних засобів для реалізації інтернет-магазину;
- 5) Розробка архітектури реалізованого додатку;
- 6) Програмна реалізація інтернет-магазину.

Практичне значення даного дипломного проекту полягає в розробці Інтернет-магазину, який після установки і розміщення його на хостингу, а також заповнення його товарами і послугами почне повноцінно виконувати поставлені перед ним цілі, функції і завдання.

До таких особливостей можна віднести:

- 1) можливість створювати, переглядати і редагувати категорії товарів і послуг;
- 2) можливість редагування, обслуговування замовлення;
- 3) право вибору видів і способів оплати товарів і послуг;
- 4) право вибору видів і способів доставки;
- 5) наявність панелі управління і адміністратора інтернет-магазину.

4

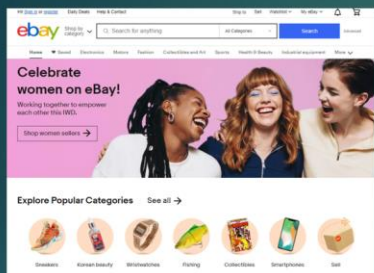
Інтернет-магазин - це веб-сайт, який рекламує товари або послуги, отримує і обробляє заявки-замовлення на покупку тих чи інших товарів і послуг, представляє користувачеві вибір типу і опції оплати, вид покупки замовлення і виставляє рахунок на оплату.

Інтернет-магазин переслідує наступні цілі:

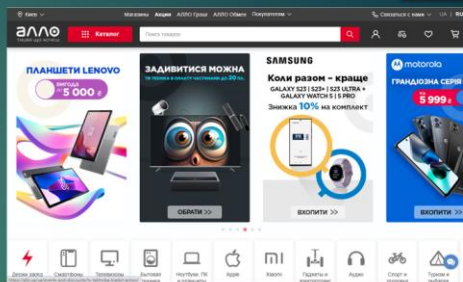
- збільшення ринку і збільшення кількості клієнтів;
- зниження трудовитрат менеджерів з продажу;
- знизити витрати на поставку ключових дій (продажі);
- зменшення кількості помилок у звітності та збільшення швидкості її складання;
- можливість реєстрації користувачів;
- особистий кабінет зареєстрованих користувачів
- можливість створення замовлення з будь-яких кількостей товарів і послуг;
- наявність багаторівневих категорій товарів і послуг з простим і зручним пошуком за заданими критеріями;
- кошик для покупок, куди будуть додані товари та послуги, обрані клієнтами;
- реєстрація та управління кейсами, товарами та послугами адміністрацією;
- можливість створювати коментарі та відгуки про роботу магазину та товар.

Аналіз аналогічних проєктів

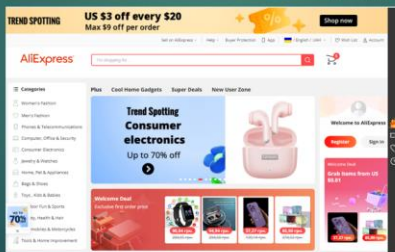
7



ebay



Allo



AliExpress

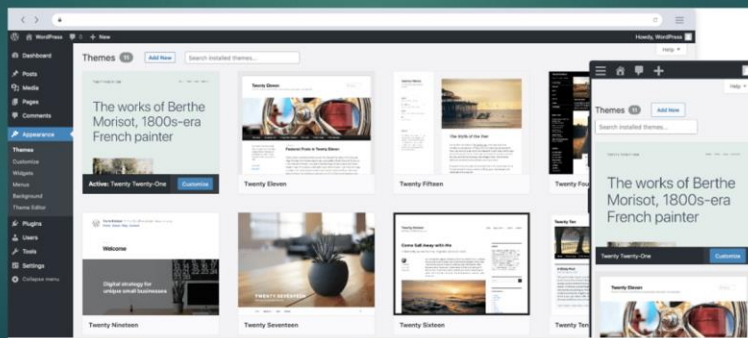
Аналіз платформ CMS



Joomla!



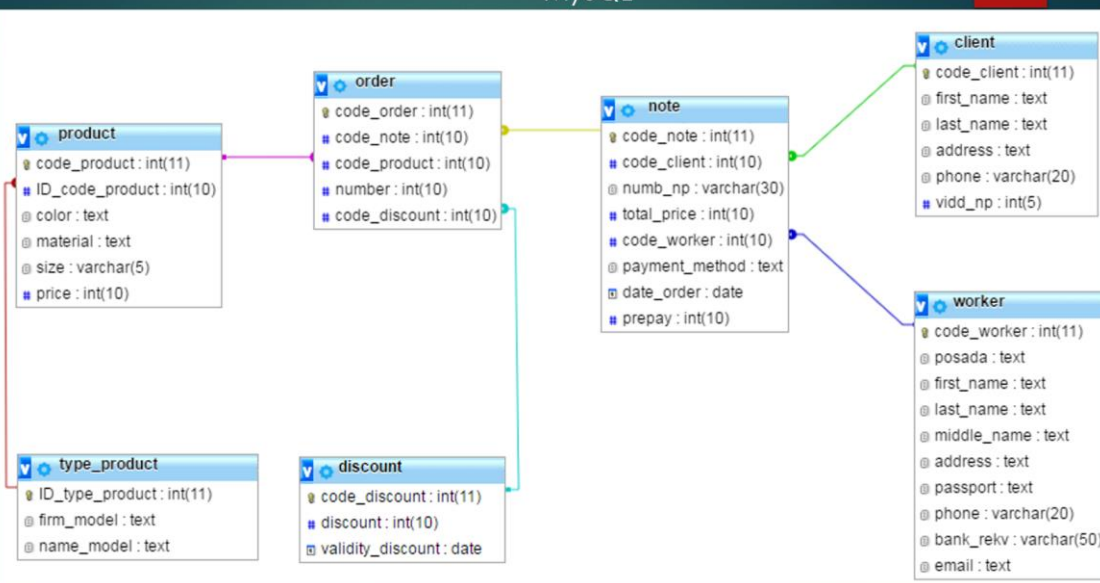
Drupal



WordPress

Моделювання баз даних MySQL

9



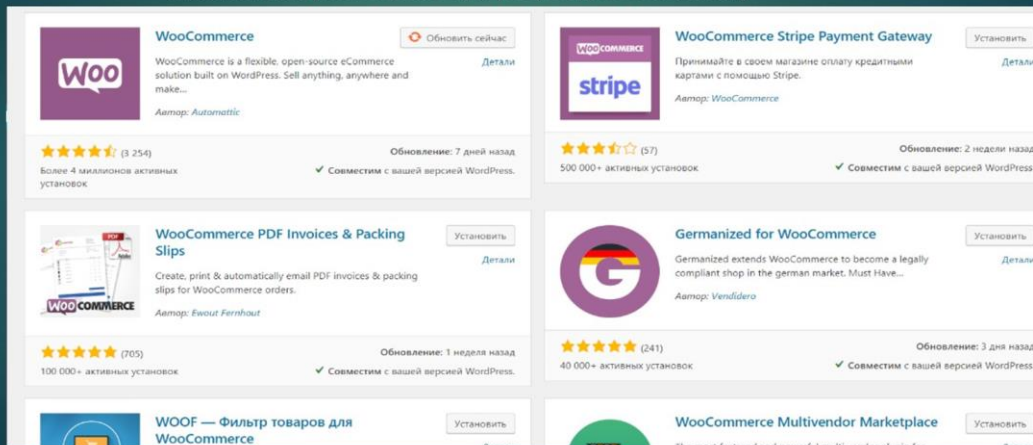
Характеристика модулів застосунок

10

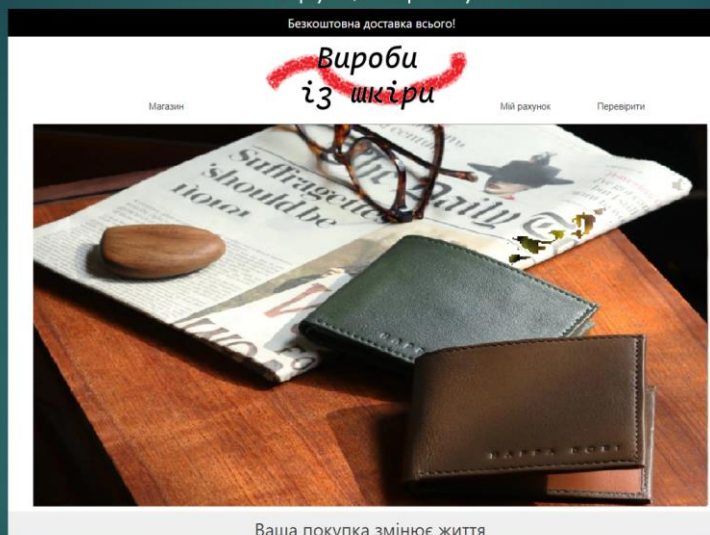
| № | Назва модуля | Функції модуля |
|---|---|--|
| 1 | Авторизація | Функція обробки входу в веб-застосунок |
| 2 | Перегляд каталогу та його окремих позицій | Функція обробки дій користувача з позиціями сайту |
| 3 | Пошук позицій | Функція пошуку потрібного виробу зі шкіри серед всіх представлених позицій |
| 4 | Додавання знайденого виробу до кошика | Виконання внутрішніх функцій обробки товарів бази даних та оформлення замовлення |
| 5 | Зміна та оновлення інформації про вироби | Адміністративна функція можливості оновлення інформації про вироби, і можливість додавання і видалення позицій з бази даних магазину |
| 6 | Перегляд інформації про вироби замовлень | Функція відображення графічної та текстової інформації про вироби зі шкіри звичайним користувачем та адміністратором. має можливість переглянути замовлення покупців |
| 7 | Перегляд замовлень | Функція, яка надає можливість переглянути замовлення як користувачам так і адміністратору |
| 8 | Оформлення замовлень | Функція обробки даних у «кошику», здійснення фінансових розрахунків та оформлення документів для продажу/відправки товару |

Плагін WooCommerce

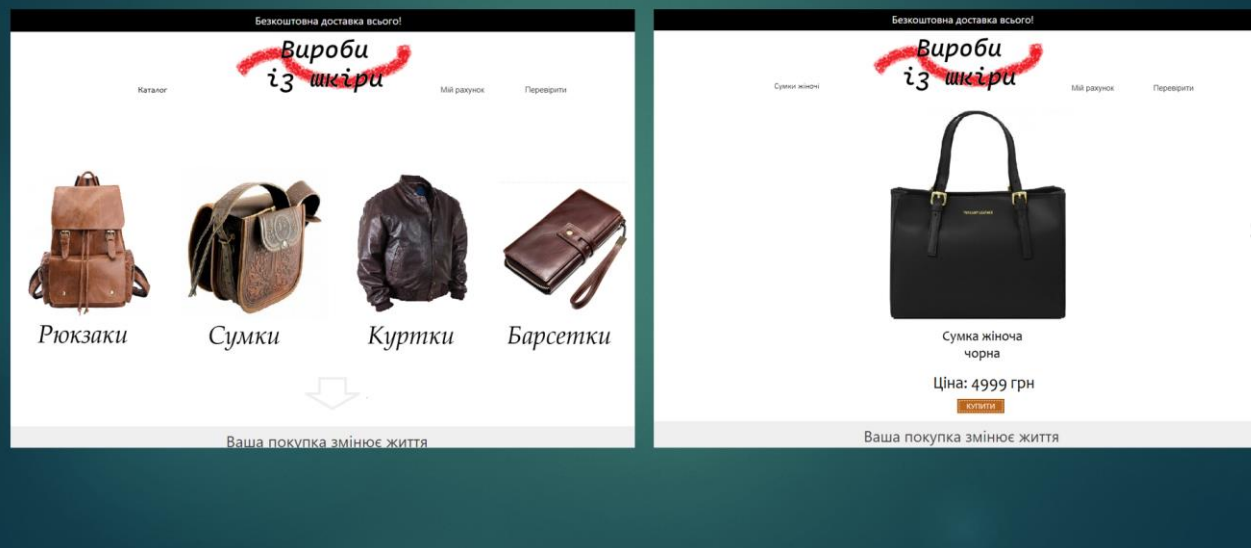
Після успішної установки WordPress на хостинг і проходження авторизації на панель управління CMS нам потрібно спочатку встановити плагін WooCommerce який ми використовуємо при розробці веб-магазину



Інструкція користувача



Інструкція користувача



Інструкція користувача

Безкоштовна доставка всього!

Каталог **Вироби із шкіри** Мій рахунок Перевірити

Створити / Форма контакту

Заповнити форму

Введіть ім'я *

Введіть номер телефону *

Телефон

Електронна пошта

Відправити

Ваша покупка змінює життя

Висновки

У ході виконання кваліфікаційної випускної роботи нами було визначено основні аспекти реалізації. Зокрема першому розділі даному розділі визначено цілі і завдання створення веб-застосунку виробів зі шкіри, визначено поняття веб-магазину та його цілі, переваги використання інтернет-магазину. У розділі проведено аналіз предметної області проаналізовано подібні проекти. Здійснено постановку задачі та описано технічне завдання.

У другому розділі здійснено аналіз засобів проектування та розробка архітектури застосунку. Зокрема проведено аналіз існуючих систем управління контентом розробки сайтів, здійснено обґрунтування вибору системи управління контентом, зроблено опис обраної системи управління контентом WORDPRESS, опис плагіна WOOCOMMERCE, здійснено та розробку архітектури застосунку з представленням UML-діаграм. В третьому розділі було зроблено проектування та розробка даних для веб-застосунку, створено за здійснено початкове заповнення бази даних. В розділі проаналізовано і спроектовано основні категорії для меню сайту, спроектовано необхідні модулі реєстрація та авторизації користувачів у системі, модулі пошуку необхідного виробу, додавання та видалення товарів адміністратором. Здійснено механізм замовлення виробу та додавання цього виробу до кошика та оформлення замовлення.

Наведено процес розробки програмних модулів веб-застосунку з демонстрацією та поясненням реалізації програмного коду. Наведена інструкція користувача а прикладах роботи веб-застосунку.

У розділі показано, що програмний застосунок для веб-застосунку для торгівлі виробами із шкіри був розроблений так, щоб можна було забезпечити максимальну зручність та простоту у користуванні.

Загалом можна сказати, що розроблений нами інтернет-магазин з дозволить власникам в загальному значно розширити кількість клієнтів, збільшити продажі товарів і послуг та дасть можливість скоротити час всього процесу торгівлі від надання виробу до його продажу, поліпшити управління веб-магазином, знизити витрати на оренду та покупку торгових площі та наймання великої кількості обслуговуючого персоналу.

Розроблений нами веб-застосунок підтримує великі можливості щодо вдосконалення та покращення, зокрема планується додавання до веб-застосунку чат-ботів, підтримки взаємодії з соціальними мережами та зворотнього зв'язку з ними, що дуже є досить важливим в умовах сучасного розвитку суспільства та інтернет-технологій.

Дякую за увагу!

ДОДАТОК Г
(обов'язковий)

КОД (ЛІСТИНГ) ПРОГРАМИ

```

public class StartLeafActivity extends AppCompatActivity {
    Button btnSingIn, btnRegister; private ProgressDialog
    loadStopsBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_start_leaf); btnSingIn =
        findViewById(R.id.buttonSingIn); btnRegister =
        findViewById(R.id.buttonRegister); loadStopsBar = new
        ProgressDialog(this); Paper.init(this);

        btnSingIn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent logsintent = new Intent(StartLeafActivity.this, LoginLeafActivity.class);
                startActivity(logsintent);
            }
        });
        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent logsintent = new Intent(StartLeafActivity.this, RegisterLeafActivity.class);
                startActivity(logsintent);
            }
        });
        String UserPhoneKey = Paper.book().read(PrevSaveInf.UserNumberKeys); String
        UserPasswordKey = Paper.book().read(PrevSaveInf.UsertxtPassKeys);

        if(UserPhoneKey != "" && UserPasswordKey != ""){ if(!TextUtils.isEmpty(UserPhoneKey) &&
        !TextUtils.isEmpty(UserPasswordKey) ){
            ValidateUser(UserPhoneKey,UserPasswordKey);

            loadStopsBar.setTitle("Вхід"); loadStopsBar.setMessage("Будь-ласка
            зачекайте..."); loadStopsBar.setCanceledOnTouchOutside(false);
            loadStopsBar.show();
        }
    }
}

private void ValidateUser(final String usphone, final String password) { final
    DatabaseReference RootLogsRef;
    RootLogsRef = FirebaseDatabase.getInstance().getReference();

    RootLogsRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if(dataSnapshot.child("Users").child(usphone).exists())
            {
                User usersData = dataSnapshot.child("Users").child(usphone).getValue(User.class);

                if(usersData.getPhone().equals(usphone))

```

```

        { if(usersData.getPassword().equals(password))
          {
            loadStopsBar.dismiss();
            Toast.makeText(StartLeafActivity.this, "Успішний вхід!",
Toast.LENGTH_SHORT).show();

                Intent homeIntent = new Intent(StartLeafActivity.this, StorageActivity.class);
                Paper.book().write(PrevSaveInf.txtphoneKey, usphone);

                startActivity(homeIntent);
            }
            else {
                loadStopsBar.dismiss
                ();
            }
        }
    }
    else {
        loadStopsBar.dismiss
        ();
        Toast.makeText(StartLeafActivity.this, "Помилка при введенні даних
спробуйте знову", Toast.LENGTH_SHORT).show();
        Paper.book().destroy();
        Intent registerIntent = new Intent(StartLeafActivity.this, StartLeafActivity.class);
        startActivity(registerIntent);
    }
}
@Override
public void onCancelled(@NonNull DatabaseError databaseError) { }
});
}
} public class RegisterLeafActivity extends AppCompatActivity {

private ImageView exit;

private Button registerBtn;

private EditText usernameInput, phoneInput, passwordInput, emailInput; private
ProgressDialog loadingBar;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState); setContentView(R.layout.activity_reg);

    exit = findViewById(R.id.exit); usernameInput = (EditText)
findViewById(R.id.NameFiled); phoneInput = (EditText)
findViewById(R.id.PhoneFiled); passwordInput = (EditText)
findViewById(R.id.passFiled); emailInput = (EditText)
findViewById(R.id.EmailFiled);

    exit.setOnClickListener(new View.OnClickListener() {
        @Override

```

```

        public void onClick(View view) {
            Intent intent = new Intent(RegisterLeafActivity.this, StartLeafActivity.class);
            startActivity(intent);
        }
    });

    loadingBar = new ProgressDialog(this); registerBtn =
    (Button) findViewById(R.id.knRegister);

    registerBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { NewAccountCreate();
        }
    });
}
private void NewAccountCreate() {
    String peoplename = usernameInput.getText().toString();
    String peoplephone = phoneInput.getText().toString();
    String peoplepassword = passwordInput.getText().toString(); String
    peopleemail = emailInput.getText().toString();

    if (TextUtils.isEmpty(peoplename)) {
        Toast.makeText(this, "Введіть ім'я", Toast.LENGTH_SHORT).show(); } else if
    (TextUtils.isEmpty(peopleemail)) {
        Toast.makeText(this, "Введіть email", Toast.LENGTH_SHORT).show();
    } else if (TextUtils.isEmpty(peoplephone)) {
        Toast.makeText(this, "Введіть номер телефону", Toast.LENGTH_SHORT).show();
    } else if (peoplepassword.length() < 6) {
        Toast.makeText(RegisterLeafActivity.this, "Пароль менше 7 символів",
    Toast.LENGTH_SHORT).show();
        return;
    } else if (peoplepassword.length() > 15) {
        Toast.makeText(RegisterLeafActivity.this, "Пароль більше 14 символів",
    Toast.LENGTH_SHORT).show();
        return;
    } else { loadingBar.setTitle("Створення облікового
        запису"); loadingBar.setMessage("Будь ласка
        зачекайте...");
        loadingBar.setCanceledOnTouchOutside(false);
        loadingBar.show();

        Validate(peoplename, peoplephone, peopleemail, peoplepassword);
    }
}

private void Validate(final String username, final String phone, final String email, final String
password) { final DatabaseReference
    ValidatessUS;
    ValidatessUS = FirebaseDatabase.getInstance().getReference();

    ValidatessUS.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override

```

```

public void onDataChange(@NonNull DataSnapshot dataSnapshot) { if
    (!(dataSnapshot.child("Users").child(phone).exists())) {
        HashMap<String, Object> userDataMap = new HashMap<>();
        userDataMap.put("phone", phone); userDataMap.put("name",
        username); userDataMap.put("email", email);
        userDataMap.put("password", password);

        ValidatessUS.child("Users").child(phone).updateChildren(userDataMap)
        .addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) { if
                (task.isSuccessful()) { loadingBar.dismiss();
                    Toast.makeText(RegisterLeafActivity.this, "Реєстрація пройшла
успішно.", Toast.LENGTH_SHORT).show();

                    Intent loginIntent = new Intent(RegisterLeafActivity.this,
StartLeafActivity.class); startActivity(loginIntent);
                } else {
                    loadingBar.dismiss();
                    Toast.makeText(RegisterLeafActivity.this, "Помилка.",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    } else {
        loadingBar.dismiss();
        Toast.makeText(RegisterLeafActivity.this, phone + "вже зареєстрований",
Toast.LENGTH_LONG).show();

        Intent loginIntent = new Intent(RegisterLeafActivity.this, LoginLeafActivity.class);
        startActivity(loginIntent);
    }
}

@Override
public void onCancelled(@NonNull DatabaseError databaseError) {

}
});
}
}

```

```

public class LoginLeafActivity extends AppCompatActivity {
    private ImageView exit; private EditText passInput,
    phoneInput; private ProgressDialog loadingBar; private Button
    buttonSingIn; private String parentDbName = "Users"; private
    CheckBox checkBoxRememberMe; private TextView
    AdminLink, NotAdminLink;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login); buttonSingIn =

```

```

(Button) findViewById(R.id.buttonSingIn2);

phoneInput = findViewById(R.id.PhoneFiled);
passInput = findViewById(R.id.passFiled); exit
= findViewById(R.id.exit);
AdminLink = (TextView)findViewById(R.id.Admin); NotAdminLink =
(TextView)findViewById(R.id.NoAdmin);

checkBoxRememberMe = (CheckBox) findViewById(R.id.login_checkbox);
Paper.init(this); loadingBar = new

ProgressDialog(this);

exit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(LoginLeafActivity.this, StartLeafActivity.class); startActivity(intent);
    }
});

buttonSingIn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { loginUser();
    }
});

    AdminLink.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
    AdminLink.setVisibility(View.INVISIBLE); NotAdminLink.setVisibility(View.VISIBLE);

    buttonSingIn.setText("Вхід для продавця");
    parentDbName = "Admins";
    }
});
NotAdminLink.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
    AdminLink.setVisibility(View.VISIBLE); NotAdminLink.setVisibility(View.INVISIBLE);
    buttonSingIn.setText("Вхід");
    parentDbName = "Users";
    }
});
}
private void loginUser() {
    String phone = phoneInput.getText().toString(); String
    password = passInput.getText().toString();

    if (TextUtils.isEmpty(phone)) {
        Toast.makeText(this, "Введіть номер телефону", Toast.LENGTH_SHORT).show();
    } else if (password.length() < 6) {
        Toast.makeText(LoginLeafActivity.this, "Пароль повинен бути не менше 7

```

```

символів", Toast.LENGTH_SHORT).show()); return;
    } else if (password.length() > 15) {
        Toast.makeText(LoginLeafActivity.this, "Пароль повинен бути не більше 14
символів", Toast.LENGTH_SHORT).show()); return;
    } else { loadingBar.setTitle("Вхід");
        loadingBar.setMessage("Будьласка зачекайте...");
        loadingBar.setCanceledOnTouchOutside(false);
        loadingBar.show();

        ValidateUser(phone, password);
    }
}
private void ValidateUser(final String phone, final String password) {

    if (checkBoxRememberMe.isChecked()) {
        Paper.book().write(PrevSaveInf.UserNumberKeys, phone);
        Paper.book().write(PrevSaveInf.UsertxtPassKeys, password);
    }

    final DatabaseReference RootRef;
    RootRef = FirebaseDatabase.getInstance().getReference();

    RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) { if
            (dataSnapshot.child(parentDbName).child(phone).exists()) {
                User usersData =
dataSnapshot.child(parentDbName).child(phone).getValue(User.class);

                if (usersData.getPhone().equals(phone)) {

                    if (usersData.getPassword().equals(password)) {
                        if(parentDbName.equals("Users")){ loadingBar.dismiss();
                            Toast.makeText(LoginLeafActivity.this, "Успішний вхід!",
Toast.LENGTH_SHORT).show();
                            Intent homeIntent = new Intent(LoginLeafActivity.this,
StorageActivity.class);
                            Paper.book().write(PrevSaveInf.txtphoneKey, phone);

                            startActivity(homeIntent);
                        } else if(parentDbName.equals("Admins")){ loadingBar.dismiss();
                            Toast.makeText(LoginLeafActivity.this, "Успішний вхід!",
Toast.LENGTH_SHORT).show();

                            Intent homeIntent = new Intent(LoginLeafActivity.this,
AdminPageActyvity.class);
                            startActivity(homeIntent);

                        }

                    } else {
                        loadingBar.dismiss();
                        Toast.makeText(LoginLeafActivity.this, "Невірний пароль",
Toast.LENGTH_SHORT).show();

```

```

        }
    }
} else {
    loadingBar.dismiss();
    Toast.makeText(LoginLeafActivity.this, "Акаунт з номером " + phone + " не
ичує", Toast.LENGTH_SHORT).show();

    Intent registerIntent = new Intent(LoginLeafActivity.this,
RegisterLeafActivity.class);
    startActivity(registerIntent);
}
}

@Override
public void onCancelled(@NonNull DatabaseError databaseError) {

}
});
}
} public class StorageActivity extends AppCompatActivity { private
ImageView exit, cart,myaccount,history,category,setting; private
DatabaseReference ProductRef; private RecyclerView recyclerView;

RecyclerView.LayoutManager layoutManager;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState); setContentView(R.layout.activity_storage);

    cart = findViewById(R.id.Cart);
    exit = findViewById(R.id.exit);

    myaccount = findViewById(R.id.MyAccount);
    category = findViewById(R.id.Category);
    history = findViewById(R.id.History); setting
= findViewById(R.id.Setting);

    ProductRef = FirebaseDatabase.getInstance().getReference().child("Products");

    recyclerView = findViewById(R.id.recycler_menu);
    recyclerView.setHasFixedSize(true); layoutManager =
new LinearLayoutManager(this);
    recyclerView.setLayoutManager(layoutManager);

    exit.setOnClickListener(new View.OnClickListener() {
        @Override public void
onClick(View view) {

            Paper.book().destroy();

            Intent intent = new Intent(StorageActivity.this, StartLeafActivity.class); startActivity(intent);
        }
    }
}

```

```

});

cart.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(StorageActivity.this, CartActivity.class); startActivity(intent);
    }
});

myaccount.setOnClickListener(new View.OnClickListener() { @Override
    public void onClick(View view) {
        Intent intent = new Intent(StorageActivity.this, MyAccount.class); startActivity(intent);
    }
});

category.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(StorageActivity.this, CategoryActivity.class); startActivity(intent);
    }
});

history.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(StorageActivity.this, History.class); startActivity(intent);
    }
});

setting.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(StorageActivity.this, SettingActivity.class); startActivity(intent);
    }
});
}

protected void onStart() {
    super.onStart();

    FirebaseRecyclerOptions<Products> options = new
        FirebaseRecyclerOptions.Builder<Products>()
            .setQuery(ProductRef, Products.class)
            .build();

    FirebaseRecyclerAdapter<Products, ProductViewHolder> adapter = new
        FirebaseRecyclerAdapter<Products, ProductViewHolder>(options) {
        @Override
        protected void onBindViewHolder(@NonNull ProductViewHolder holder, int
position, @NonNull Products model)
            { holder.txtProductId.setText(model.getPid());
              holder.txtProductName.setText(model.get
Pname());
              holder.txtProductSize.setText(model.getPsi

```

```

ze());
holder.txtProductPriceP.setText(model.get
PPriseP());
Picasso.get().load(model.getImage()).into(holder.imageView);

holder.itemView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent Intent = new
Intent(StorageActivity.this,ProductDetailsActivity.class); Intent.putExtra("Rid",
        model.getRid());
        startActivity(Intent);
    }
});
}
@NonNull
@Override
public ProductViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType)
{
    View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.product_items_layout, parent, false);
    ProductViewHolder holder = new ProductViewHolder(view); return
    holder;
}
};
recyclerView.setAdapter(adapter);
adapter.startListening();
}
}

```

```

package com.dd_1;
import androidx.appcompat.app.AppCompatActivity;

```

```

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import
android.widget.ImageView;

```

```

public class AdminPageActvity extends AppCompatActivity { private
    ImageView exit;
    private Button addtovbtn,poiskspisaniebtn,prosmotrorders;

    @Override
    protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin_page);

        addtovbtn=(Button)findViewById(R.id.AddTov);
        poiskspisaniebtn=(Button)findViewById(R.id.SPT);
        prosmotrorders=(Button)findViewById(R.id.AdminOrders); exit =
        findViewById(R.id.exit);
        exit.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View view) {
            Intent intent = new Intent(AdminPageActivity.this, StartLeafActivity.class); startActivity(intent);
        }
    });
    addtovbtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(AdminPageActivity.this, AddTovarActivity.class);
            intent.putExtra("Pid", "");
            startActivity(intent);
        }
    });
    poiskspisaniembtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(AdminPageActivity.this, ProsmotrSpisaniyes.class);
            startActivity(intent);
        }
    });
    prosmotrorders.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(AdminPageActivity.this, ProsmotrOrdersAdmin.class);
            startActivity(intent);
        }
    });
}
}

```

```
package com.dd_1;
```

```

import
android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle; import
android.view.LayoutInflater; import
android.view.View; import
android.view.ViewGroup; import
android.widget.Button; import
android.widget.ImageView; import
android.widget.Toast;

```

```

import androidx.annotation.NonNull; import
androidx.appcompat.app.AlertDialog; import
androidx.appcompat.app.AppCompatActivity; import
androidx.recyclerview.widget.LinearLayoutManager; import
androidx.recyclerview.widget.RecyclerView;

```

```

import com.dd_1.Models.Products; import
com.dd_1.ViewHolder.ProductViewHolder; import
com.firebase.ui.database.FirebaseRecyclerAdapter; import
com.firebase.ui.database.FirebaseRecyclerOptions; import

```

```

com.google.android.gms.tasks.OnCompleteListener; import
com.google.android.gms.tasks.Task; import
com.google.firebase.database.DatabaseReference; import
com.google.firebase.database.FirebaseDatabase; import
com.squareup.picasso.Picasso; public class ProsmotrSpisanies
extends AppCompatActivity {

    private DatabaseReference ProductRef;
    private RecyclerView recyclerView;
    RecyclerView.LayoutManager layoutManager; private

    ImageView exit;

    @Override protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_prosmotr_spisanies);

        ProductRef = FirebaseDatabase.getInstance().getReference().child("Products");

        recyclerView = findViewById(R.id.recycler_menu);
        recyclerView.setHasFixedSize(true); layoutManager =
        new LinearLayoutManager(this);
        recyclerView.setLayoutManager(layoutManager);
        exit = findViewById(R.id.exit);
        exit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent intent = new Intent(ProsmotrSpisanies.this, AdminPageActyvity.class);
                startActivity(intent);
            }
        });
    }

    @Override
    protected void onStart()
    {
        super.onStart();
        final DatabaseReference ProductListRef =
        FirebaseDatabase.getInstance().getReference().child("Products");

        FirebaseRecyclerOptions<Products> options = new
        FirebaseRecyclerOptions.Builder<Products>()
            .setQuery(ProductRef, Products.class)
            .build();

```

```

        FirebaseRecyclerAdapter<Products, ProductViewHolder> adapter = new
        FirebaseRecyclerAdapter<Products, ProductViewHolder>(options) {
            @Override protected void onBindViewHolder(@NonNull ProductViewHolder holder,
            int
position, @NonNull Products model)
            { holder.txtProductId.setText(model.getPid());
              holder.txtProductName.setText(model.getPname());
              holder.txtProductSize.setText(model.getPsize());
              holder.txtProductPriceP.setText(model.getPPriseP());

              Picasso.get().load(model.getImage()).into(holder.imageView);
              holder.itemView.setOnClickListener(new View.OnClickListener() {
                  @Override
                  public void onClick(View v) {
                      CharSequence options [] = new CharSequence[] {

                          "Редагувати",
                          "Видалити"
                      };
                      AlertDialog.Builder buil = new
AlertDialog.Builder(ProsmotrSpisanies.this); buil.setTitle("Опції:");

                      buil.setItems(options, new DialogInterface.OnClickListener() {
                          @Override
                          public void onClick(DialogInterface dialog, int i) { if(i ==
0){
                              Intent intent = new Intent(ProsmotrSpisanies.this,
AddTovarActivity.class ); intent.putExtra("Pid", model.getRid());
                              startActivity(intent);
                          }
                          if (i == 1) {
                              ProductListRef.child(model.getRid())
                                  .removeValue().addOnCompleteListener(new
OnCompleteListener<Void>() {
                                  @Override public void onComplete(@NonNull
Task<Void> task) {

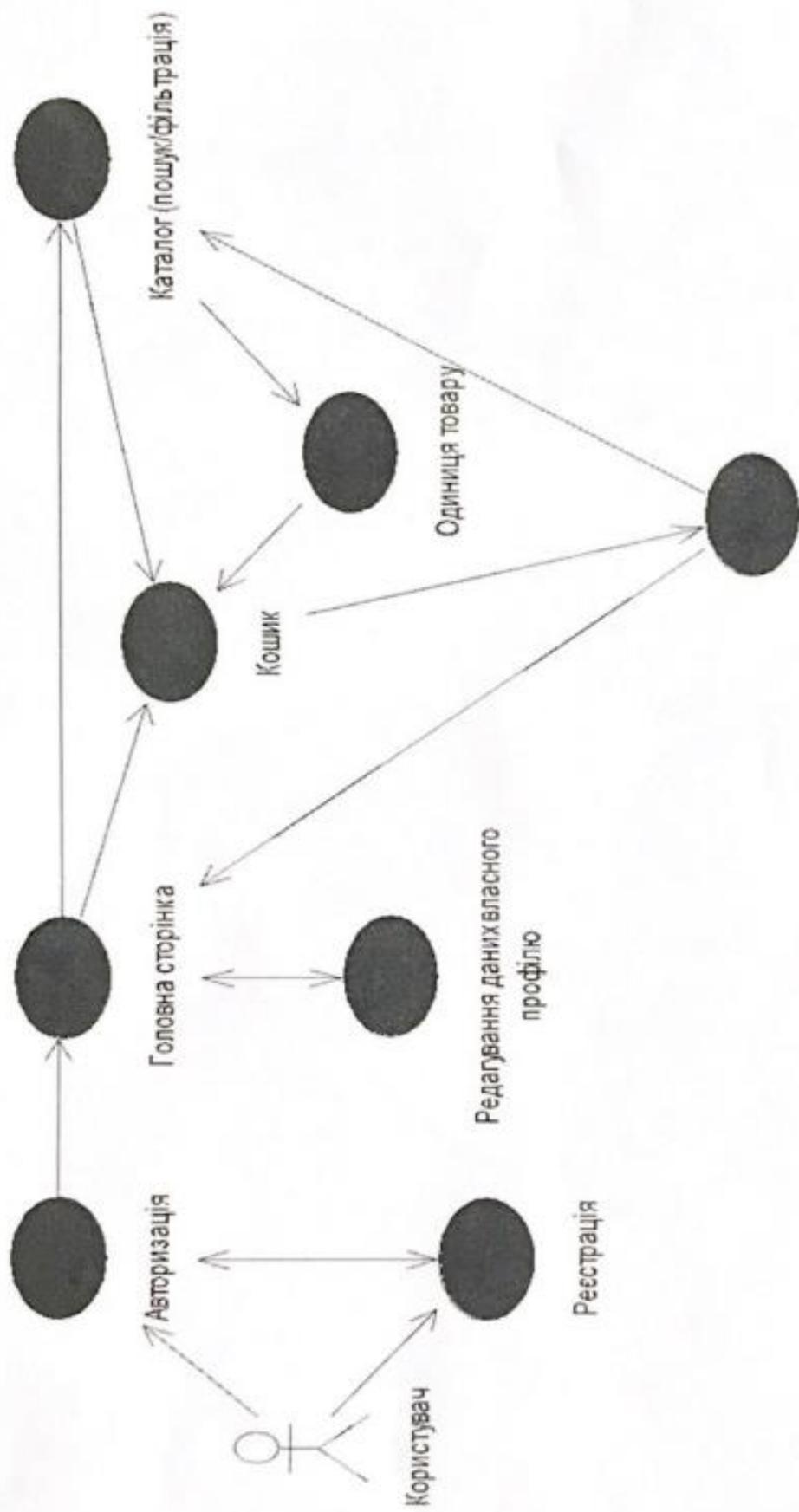
                                      if (task.isSuccessful()){
                                          Toast.makeText(ProsmotrSpisanies.this, "Видалення
позиції", Toast.LENGTH_SHORT).show();

                                          Intent intent = new Intent(ProsmotrSpisanies.this,
ProsmotrSpisanies.class);
                                          startActivity(intent);
                                      }
                                  }
                              });
                          }
                      }
                  }
            }
    }

```

```
        });
        build()
        how()
        ;
    }
    });
}
@NonNull @Override public ProductViewHolder onCreateViewHolder(@NonNull
viewType) ViewGroup parent, int viewType)
{
    View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.product_items_layout, parent, false);
    ProductViewHolder holder = new ProductViewHolder(view); return
    holder;
}
};
recyclerView.setAdapter(adapter);
adapter.startListening();
}
```

ГРАФІЧНА ЧАСТИНА



Оформлення замовлення

Рисунок 2 – Діаграма використання для користувачів

| | | | | |
|---|--|-----------|-----------------|----------------|
| КвРІПЗ.190130.01.05.Е8 | | Ділова | Місяць | Місяць |
| Веб-застосунок для торгівлі виробами із шкіри | | Розробник | Прийом | Місяць |
| | | Варіант | Версія | Версія |
| | | Місяць | Версія | 3 |
| | | Н.Коліс | Г.Коліс | ХРНУ ІІІЗ-19-1 |
| | | Дан. виф. | Сейчасовий П.П. | 7.000 |

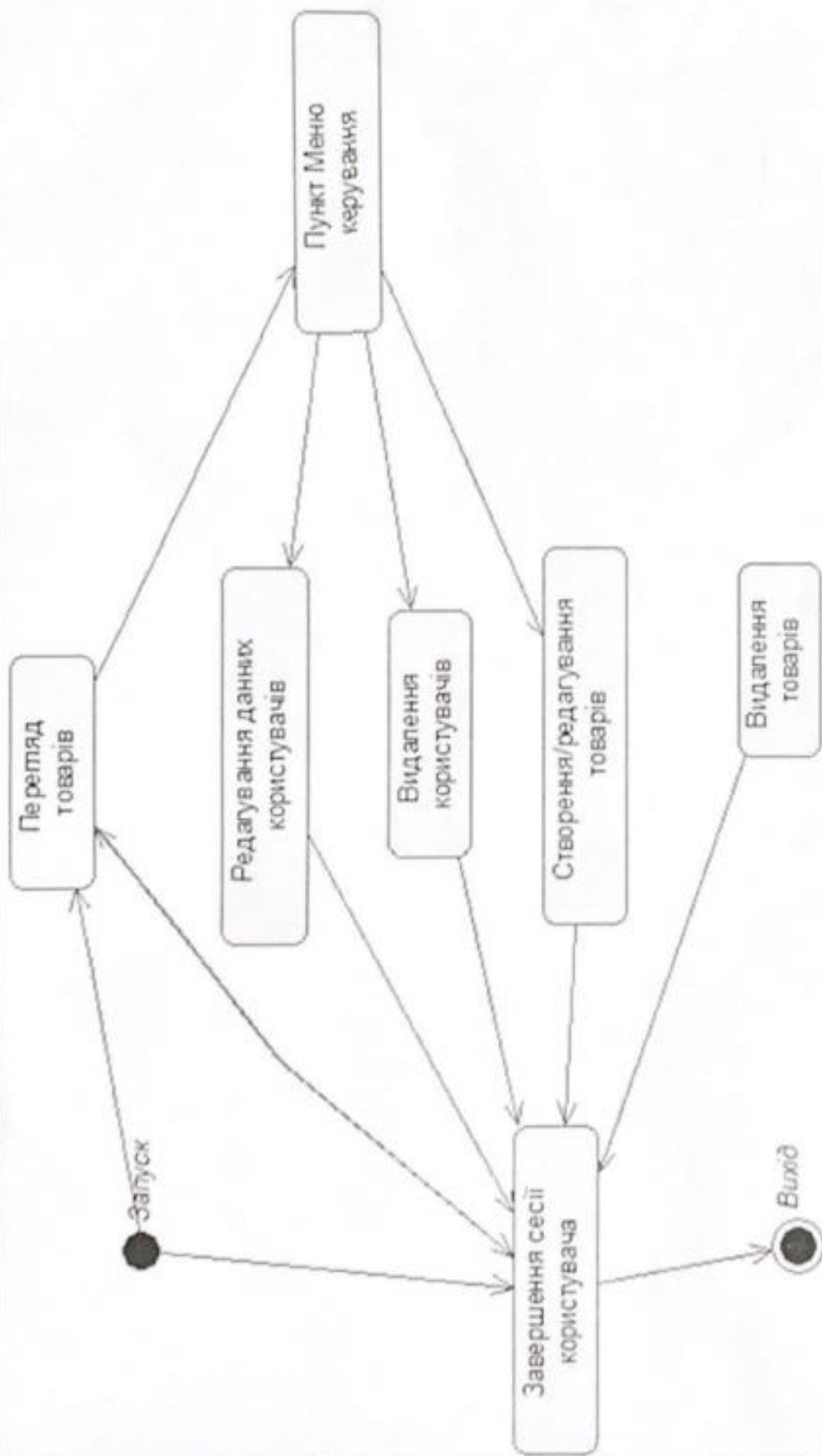


Рисунок 3 – Діаграма станів

| | | | | |
|---|--------------|--------|----------|--------|
| КерІПЗ.190130.01.05.Е8 | | Ліпень | Місяць | Місяць |
| Веб-застосунок для торгівлі виробами із шкіри | | Ліпень | Листопад | 3 |
| Зав. проєктом | Михайло М. | Ліпень | Листопад | 3 |
| Програмувальник | Давид М. | Ліпень | Листопад | 3 |
| Тестувальник | Світлана Ю. | Ліпень | Листопад | 3 |
| Відповідальний | Володимир П. | Ліпень | Листопад | 3 |
| Інженер | Гурман І. | Ліпень | Листопад | 3 |
| Зав. офіс | Світлана П. | Ліпень | Листопад | 3 |
| ХНУ ІІІЗ-19-1 | | | | |

Завідувачу кафедри інженерії програмного
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Данькевич Н. В.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІПЗ-19-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/або Anti-Plagiarism) і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

05.06.2023

дата



підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 4.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 16%**

| | | | | |
|--|----------|---------|-----------------------------|----------|
| ID: 114964 Назва: БКР Веб-застосунок для торгівлі виробами із шкіри Додано в БД: 2023-06-06 Автора: Данькевич Н.В. Керівники: Форкун Ю.В. к.т.н. доц Консультанти: Опоненти: | Документ | | Сумарний збіг по Базі Даних | |
| | Символи | Лексеми | Символи | Лексеми |
| | 58126 | 523 | 5951 (10%) | 66 (13%) |

Джерело плагіату

| ID | Опис | Наявність плагіату в документі | |
|----|------|--------------------------------|---------|
| | | Символи | Лексеми |
| | | | |

Ім'я користувача:
Кафедра ІПЗ

ID перевірки:
1015463525

Дата перевірки:
06.06.2023 16:08:18 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
06.06.2023 16:13:05 EEST

ID користувача:
100005589

Назва документа: Данькевич КВР Плагіат

Кількість сторінок: 60 Кількість слів: 9135 Кількість символів: 72911 Розмір файлу: 3.51 MB ID файлу: 1015122575

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

10.6% Схожість

Найбільша схожість: 1.43% з джерелом з Бібліотеки (ID файлу: 1015061585)

9.22% Джерела з Інтернету

716

Сторінка 62

3.47% Джерела з Бібліотеки

76

Сторінка 66

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

28

Підозріле форматування

11
сторінок

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «Бакалавр»

Дипломник Данькевич Назарій Вікторович

Тема Веб-застосунок для торгівлі виробами із шкіри

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 ; кількість сторінок записки 97

1. Короткий зміст пояснювальної записки та прийнятих рішень У процесі виконання кваліфікаційної роботи була проведено детальне дослідження предметної області, включаючи вивчення всіх функціональних і нефункціональних вимог. Було ретельно проаналізовано існуючі альтернативні рішення на ринку, їх переваги та недоліки, що підтверджує актуальність розробки нового програмного забезпечення. В процесі вивчення були враховані різні інструменти для реалізації запропонованих рішень, що дозволило успішно розробити програмне забезпечення. Також було проведено тестування розробленого програмного застосунку, що підтверджено його коректну роботу та готовність до впровадження.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступній частині було показано, що обрана тема є актуальною, а також визначено мету та завдання дипломного проектування. Перший розділ присвячений детальному аналізу предметної області, в якому було розглянуто наявні рішення та встановлені функціональні та нефункціональні вимоги до розроблюваного програмного забезпечення. У другому розділі проведений аналіз сучасних архітектур, в якому були розглянуті переваги та недоліки, і визначено, що система буде базуватися на монолітній архітектурі та моделі клієнт-сервер. Третій розділ був присвячений підготовці всіх залежностей для програмного коду і практичній розробці програмних модулів з описом їх особливостей, що призвело до створення програмного продукту. У цьому розділі також проведено функційне тестування системи згідно з функціональними вимогами, що підтвердило правильну роботу застосунку.

4. Позитивні сторони роботи Застосунок має приємний та сучасний дизайн в стилі мінімалізму та в спокійних тонах. Елементи інтерфейсу працюють плавно та мають зрозумілу та просту структуру

5. Негативні сторони роботи Необхідність додати чат-бота в застосунок.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки, оскільки матеріал пояснювальної записки демонструє чітку структуру, логічну послідовність, зрозумілий та доступний виклад. Це дозволяє чітко сприйняти зміст роботи, який гармонійно вписується в контекст тематики проектування. Графічний матеріал, наданий у роботі, надає можливість візуально ознайомитись з деталями проектування системи.

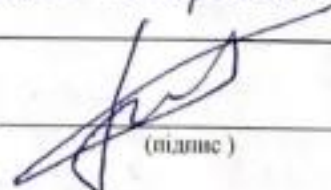
8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «задовільно».

РЕЦЕНЗЕНТ: Олександр Вал. Комісаренко керівник
на інформаційних системах Бюро КІС

“ 05 ” 06

2023 р.


(підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів подібності щодо роботи, продуктованими програмно-технічним засобом (ами) перевірки текстів на плагіат:

Назва: «Веб-застосунок для торгівлі виробами із шкіри»

Автор: Данькевич Назарій Вікторович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Форкун Юрій Вікторович, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|--|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |
| 5 | Інше: | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 10,6% і адресується до титульної сторінки, відомості документів, змісту, заголовків пунктів, літературних джерел та звіту з переддипломної практики, який містить у собі частину кваліфікаційної роботи, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 7.06.23

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи





Леонід БЕДРАТЮК

Леонід БЕДРАТЮК

Юрій ФОРКУН

Завідувачу кафедри інженерії програмного
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Данькевич Н. В.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІІЗ-19-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/або Anti-Plagiarism) і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

05.06.2023

дата



підпис

Завідувачу кафедри
інженерії програмного забезпечення
проф. Бедратюку Л. П.

студента групи ІПЗ-19-1

Рачкевич Н.В.
Прізвище, ініціали

ЗАЯВА

Прошу закріпити за мною тему кваліфікаційної роботи освітнього ступеня
«бакалавр» за спеціальністю 121 «Інженерія програмного забезпечення»:

Веб-застосунок для торгівлі виробами із
шкіри

(керівник роботи – Форкун Ю.В.)
Прізвище, ім'я, по батькові

05.06.2023
Дата


Підпис студента

ДЕКЛАРАЦІЯ УЧАСНИКА ОСВІТНЬОГО ПРОЦЕСУ

щодо дотримання академічної доброчесності

Цією декларацією я, Данієл Вікторівич Назарій
Прізвище, ім'я, по батькові

студент ІV курсу факультету інформаційних технологій, кафедри інженерії програмного забезпечення

здобувач вищої освіти (шифр та назва спеціалізації, курс, академічна група) / науково-педагогічний працівник (назва кафедри)

назва факультету

підтверджую, що ознайомився (- лась) з Положенням про систему забезпечення академічної доброчесності у Хмельницькому національному університеті та Кодексом академічної доброчесності і **зобов'язуюсь** дотримуватися їх вимог під час освітнього процесу, проведення наукової діяльності, виконання організаційно-адміністративних функцій тощо.

Усвідомлюю, що у разі порушення мною принципів академічної доброчесності нестиму відповідальність перед академічною спільнотою ХНУ згідно з нормами, визначеними Положенням про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, законодавства України

« 05 » лютого 2023 р.


Підпис