

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

 проф. Приходько С.Б.

«__» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: Розробка вебсервісу для магазину кави «Kavortra»

Виконав: студент групи 4151

 Руденко А.М.

(підпис, ПІБ)

Керівник роботи:

доцент, к.т.н., доцент

(посада, науковий ступень вчене звання)

 Каіров В.О.

(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»



«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

доц. Макарова Л.М.

(підпис)

«___» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту Руденко Альберту Миколайовичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебсервісу для магазину кави «Kavorra»

Керівник роботи Каіров Володимир Олексійович

Затверджені наказом ректора №153 від 08.04.2025 р.

2. Термін подання роботи: 02.06.2025 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проєкт програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів: 1 Титульний слайд, 2 Мета, завдання та об'єкт роботи, 3 Існуючі рішення, 4 Функціональні вимоги до вебсервісу для магазину кави «Kavorra»», 5 Діаграма варіантів використання вебсервісу для магазину кави «Kavorra»», 6 Архітектура вебсервісу для магазину кави «Kavorra»», 7 Діаграми діяльності вебсервісу для магазину кави «Kavorra»», 8 Інформаційна модель вебсервісу для магазину кави «Kavorra»», 9 Діаграма класів вебсервісу для магазину кави «Kavorra»», 10 Діаграми послідовності вебсервісу для магазину кави «Kavorra»», 11 Логічна модель даних вебсервісу для магазину кави «Kavorra»», 12 Результати розробки вебсервісу для магазину кави «Kavorra»», 13 Висновки, 14 Заключний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 08.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	31.03.2025	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	07.04.2025	ПП*
3.	Аналіз вимог до ПЗ	14.04.2025	ПП*
4.	Розробка архітектури ПЗ	21.04.2025	
5.	Розробка проєкту ПЗ	05.05.2025	
6.	Реалізація та тестування ПЗ	12.05.2025	
7.	Підготовка розділу Результати розробки ПЗ	16.05.2025	
8.	Підготовка розділу з охорони праці	19.05.2025	ПП*
9.	Підготовка розділу ВИСНОВКИ	23.05.2025	
10.	Оформлення списку використаних джерел та додатків	26.05.2025	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	02.06.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12.	Підготовка презентації та доповіді	06.06.2025	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	09.06.2025	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	16.06.2025	

* - за результатами переддипломної практики (ПП), яке було з 05.02.2024 по 24.03.2024 р.

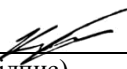
Студент


(підпис)

Руденко А.М.

(ПІБ)

Керівник роботи


(підпис)

Каіров В.О.

(ПІБ)

АНОТАЦІЯ

Дана кваліфікаційна робота присвячена вирішенню практичної задачі інженерії програмного забезпечення, а саме розробці вебсервісу для магазину кави «Kavogga». Задача характеризується комплексністю, що полягає в наявності взаємопов'язаних складових, в необхідності оптимізації ресурсів, в різноманітності вимог, та невизначеністю умов, яка виникає через нечіткі або недостатньо визначені умови задачі, змінність вимог користувача та динамічність середовища.

Об'єктом кваліфікаційної роботи є процес розробки вебсервісу для магазину кави «Kavogga».

Робота містить аналіз практичної задачі інженерії програмного забезпечення, аналіз існуючих аналогів, постановку задачі на розробку вебсервісу для магазину кави «Kavogga», аналіз вимог, розробку архітектури, проєкт програмного забезпечення, результати розробки, розділ з охорони праці та додатки.

Робота виконана на 112 сторінках друкованого тексту, містить 20 рисунків, 29 таблиць, 5 додатків та список використаних джерел з 7 найменувань.

ABSTRACT

This qualification work is dedicated to solving a practical problem of software engineering, namely the development of a web service for the coffee shop "Kavorra". The task is characterized by complexity, which consists in the presence of interconnected components, the need to optimize resources, the diversity of requirements, and the uncertainty of conditions, which arises due to vague or insufficiently defined task conditions, variability of user requirements and dynamism of the environment.

The object of the qualification work is the process of developing a web service for the coffee shop "Kavorra".

The work contains an analysis of a practical software engineering problem, an analysis of existing analogues, a task statement for developing a web service for the Kavorra coffee shop, requirements analysis, architecture development, software project, development results, a section on occupational safety and appendices

The work is completed on 112 pages of printed text, contains 20 figures, 29 tables, 5 appendices and a list of used sources from 7 names.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA».....	9
1.1 Аналіз практичної задачі інженерії програмного забезпечення для розробки вебсервісу для магазину кави «Kavorra»	9
1.2 Опис існуючих аналогів та обґрунтування необхідності розробки власного вебсервісу для магазину кави «Kavorra».....	10
1.3 Постановка задачі на розробку вебсервісу для магазину кави «Kavorra»...	15
2 РОЗРОБКА ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA».....	18
2.1 Аналіз вимог до вебсервісу для магазину кави «Kavorra».....	18
2.1.1 Розробка моделі варіантів використання вебсервісу для магазину кави «Kavorra»	18
2.1.2 Специфікації прецедентів вебсервісу магазину кави «Kavorra».....	22
2.2 Розробка архітектури вебсервісу для магазину кави «Kavorra»	34
2.2.1 Діаграма компонентів вебсервісу магазину кави «Kavorra»	36
2.2.2 Діаграма розгортання вебсервісу магазину кави «Kavorra»	38
2.3 Проєкт вебсервісу для магазину кави «Kavorra».....	39
2.3.1 Ескізний проєкт вебсервісу магазину кави «Kavorra»	39
2.3.2 Технічний проєкт вебсервісу магазину кави «Kavorra»	48
2.3.3 Робочий проєкт вебсервісу магазину кави «Kavorra»	63
3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»	71
4 ОХОРОНА ПРАЦІ	73
4.1 Основні положення охорони праці в Україні	73
4.2 Аналіз шкідливих та небезпечних факторів при роботі з комп'ютером.....	74
4.3 Заходи щодо зменшення впливу шкідливих та небезпечних факторів під час роботи за комп'ютером	75

ВИСНОВКИ	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	78
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»	79
ДОДАТОК Б – ОПИС ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»	85
ДОДАТОК В – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA».....	89
ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»	93
ДОДАТОК Д – ТЕКСТ ПРОГРАМИ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA».....	101

ВСТУП

Бурхливий розвиток сучасних технологій зумовив формування та розвиток цифрової економіки. У цей процес залучена і торгівля, яка переходить на новий рівень: спостерігаються збільшення частки електронної комерції у загальному обсязі торгівлі, зміна співвідношення реклами online та offline, підвищення продуктивності за рахунок автоматизації роботи. Електронна комерція охоплює широкий спектр діяльності, зокрема: інтернет-магазини, вебсервіси, електронні аукціони, майданчики для розміщення оголошень і таке інше.

Кваліфікаційна робота присвячена вирішенню практичної задачі інженерії програмного забезпечення, а саме розробці вебсервісу для магазину кави «Kavorra», який планується відкрити найближчим часом.

Розробка власного вебсервісу має ряд переваг, а саме: унікальний дизайн, тільки необхідний функціонал, забезпечення безпеки, збір і аналіз даних клієнтів, аналіз відгуків, швидке внесення змін, гнучкість ціноутворення.

Метою даної кваліфікаційної роботи є розв'язання практичної задачі інженерії програмного забезпечення, а саме розробка якісного, зручного вебсервісу для магазину кави «Kavorra».

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- виконати аналіз практичної задачі інженерії програмного забезпечення;
- виконати аналіз існуючих аналогів та обґрунтування необхідності розробки власного вебсервісу для магазину кави «Kavorra»;
- сформулювати вимоги до розроблюваного продукту та розробити технічне завдання на розробку вебсервісу для магазину кави «Kavorra»;
- виконати аналіз вимог;
- розробити архітектуру програмного забезпечення;
- розробити ескізний, технічний та робочий проєкти;
- розробити документацію до програмного продукту.

Об'єктом кваліфікаційної роботи є процес розробки вебсервісу для магазину кави «Kavorra».

У результаті ми повинні отримати вебсервіс для магазину кави «Kavorra», який буде зручним у використанні та сприятиме розширенню бізнесу за рахунок представництва в мережі Інтернет.

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»

1.1 Аналіз практичної задачі інженерії програмного забезпечення для розробки вебсервісу для магазину кави «Kavorra»

Кваліфікаційна робота присвячена вирішенню практичної задачі інженерії програмного забезпечення, а саме розробці вебсервісу для магазину кави «Kavorra», який планується відкрити найближчим часом.

Дана задача характеризується комплексністю, що полягає в наявності взаємопов'язаних складових, в необхідності оптимізації ресурсів, в різноманітності вимог, та невизначеністю умов, яка виникає через нечіткі або недостатньо визначені умови задачі, змінність вимог користувача та динамічність середовища.

Вебсервіси дозволяють вирішувати наступні задачі:

- Розширення аудиторії та географії клієнтів за рахунок правильно побудованої структури вебсервісу, яка підвищує його видимість в пошукових системах.
- Оптимізація витрат. Це досягається за рахунок збільшення прибутковості при поєднанні онлайн та офлайн продажів.
- Зручна презентація всіх товарів в асортименті з використанням правильно побудованого та повного каталогу, що дозволить продемонструвати переваги товару, спростити вибір та зменшити навантаження на менеджерів.
- Забезпечення додаткових конкурентних переваг для бізнесу, які полягають в доступі до більших аудиторій, виходу на нові ринки.
- Можливість пропонувати супутні додаткові послуги та товари.

1.2 Опис існуючих аналогів та обґрунтування необхідності розробки власного вебсервісу для магазину кави «Kavorra»

Перед розробкою вебсервісу для магазину кави «Kavorra» буде здійснено огляд існуючих аналогів. Такий аналіз допоможе сформувати уявлення про необхідний функціонал, визначити сильні сторони конкурентів, типові недоліки та особливості реалізації подібних сервісів.

Розглянемо деякі вебсервіси з продажу кави:

Вебсервіс COFFE UA [1]

Головна сторінка вебсервісу наведена на рисунку 1.1.

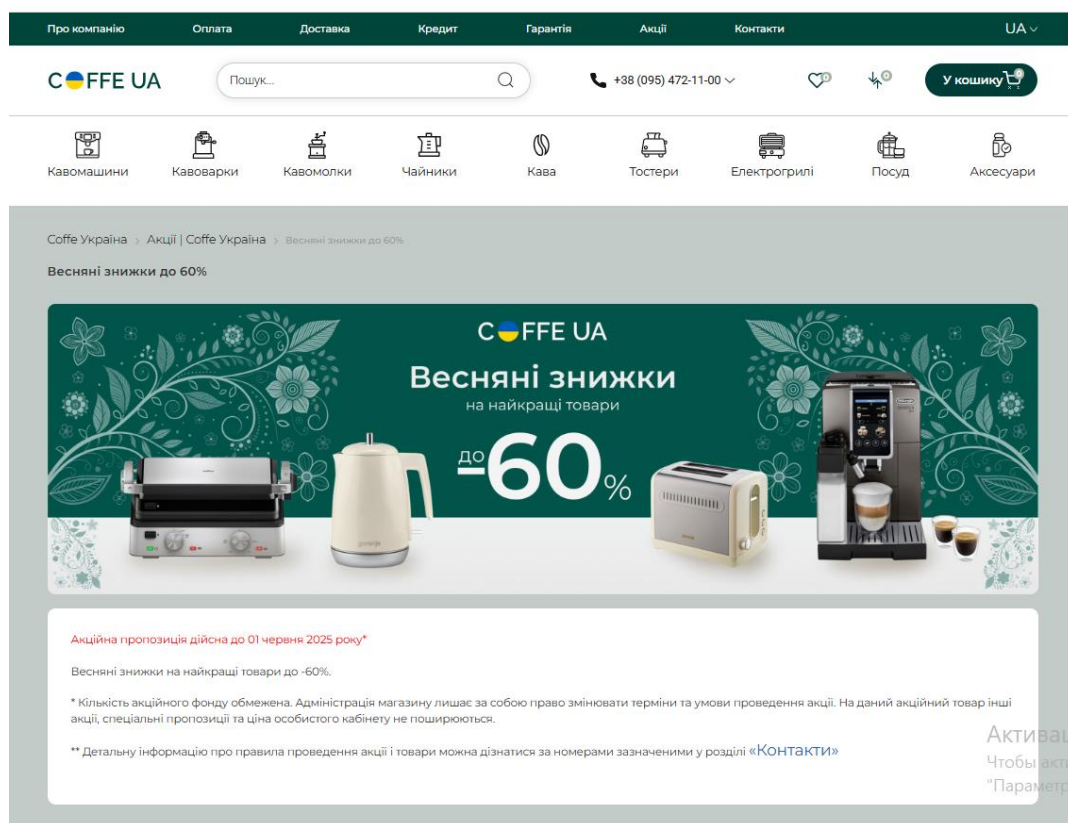


Рисунок 1.1 – Головна сторінка вебсервісу COFFE UA

Функції сайту:

- каталог продукції з поділом за типами кави;
- сортування та фільтрація товарів за брендом, типом, ступенем обсмаження;
- вибір типу оплати;

- акції та система знижок;
- контакти та інформація про компанію.

Плюси:

- наявний вибір кави;
- зрозумілий інтерфейс;
- акції та знижки – часто оновлюються пропозиції;
- можливість онлайн-оплати;
- фільтрація товарів – економить час підбору потрібної кави.

Мінуси:

- дещо застарілий дизайн;
- слабкий зворотній зв'язок (тільки контактні телефони та email);
- мало інформації про каву.

Вебсервіс Coffee Store [2]

Головна сторінка вебсервісу наведена на рисунку 1.2.

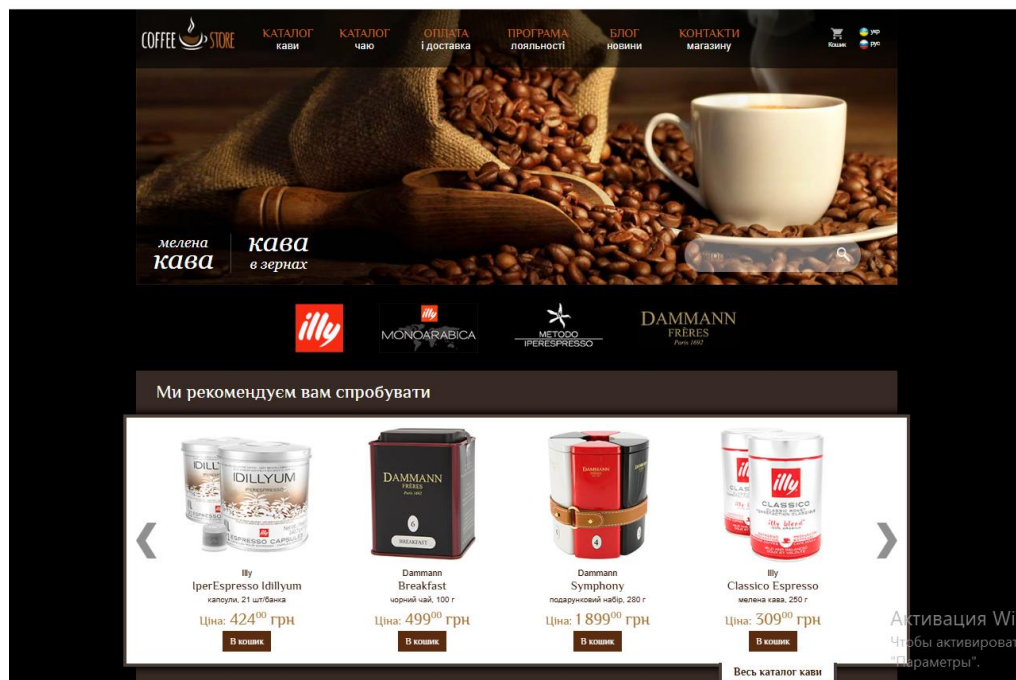


Рисунок 1.2 – Головна сторінка вебсервісу Coffee Store

Функціональні можливості:

- сортування та фільтрація;
- вибір типу оплати;

- програма лояльності;
- блог з порадами щодо вибору кави та методів приготування;
- контактна форма.

Плюси:

- зручна навігація – зрозуміле меню та логічна структура сайту;
- широкий асортимент;
- сучасний дизайн, приємне оформлення;
- блог та поради – додаткова цінність для клієнтів;
- можливість надіслати повідомлення.

Мінуси:

- часом повільне завантаження;
- опис деяких товарів короткий;
- обмежена інформація про магазин.

Вебсервіс *IDEALIST COFFEE CO* [3]

Головна сторінка вебсервісу наведена на рисунку 1.3.

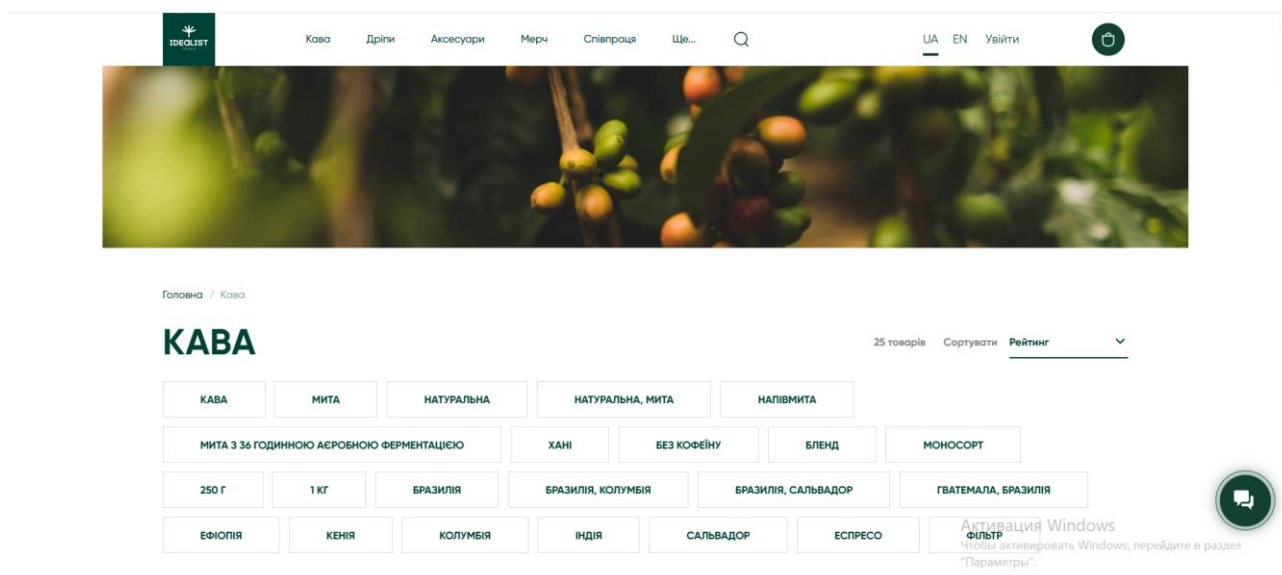


Рисунок 1.3 – Головна сторінка вебсервісу *IDEALIST COFFEE CO*

Функціональні можливості:

- повноцінний каталог продукції з фільтрами;
- використання промокодів;

- онлайн-оплата;
- інформаційні сторінки про походження кави, відповідальність та цінності бренду;
- різні форми доставки.

Плюси:

- сучасний стильний дизайн з фокусом на користувацький досвід;
- підписка на каву – рідкісна опція для українських магазинів;
- зрозуміла подача інформації про продукти;
- висока швидкість роботи.

Мінуси:

- обмежений асортимент – переважно кава власного обсмаження, без сторонніх брендів чи обладнання;
- слабкий зворотній зв'язок;
- вища цінова категорія – не для всіх сегментів покупців.

Аналіз існуючих аналогів дозволив виокремити як найкращі їх практики так і типові проблеми, а також визначитися з функціоналом вебсервісу для магазину кави «Kavorra».

Функціонал користувача сайту та адміністратора сайту (користувач):

– Вебсервіс повинен надавати можливість переглядати всі товари каталогу, які мають відображатися на головній сторінці. Для кожного товару мають бути вказані: зображення товару, назва товару, категорія, вага та ціна.

– Користувач повинен мати можливість здійснювати вибір певної категорії товарів з подальшим відображенням на головній сторінці лише відповідних товарів.

– Також необхідно реалізувати перегляд детальної інформації про товар, яка має включати зображення товару, назву товару, виробника, країну виробника, походження, категорію, сорт, помел, вагу, обсмаження, упаковку, ціну, опис.

– Програмне забезпечення повинно реалізовувати два види пошуку товару в каталозі: пошук по назві та розширений пошук. Вхідними даними для розширеного пошуку може бути довільна комбінація характеристик товару.

– При перегляді каталогу товарів чи детальної інформації про товар повинна надаватися можливість додати товар у кошик.

– Для роботи з кошиком необхідно реалізувати перегляд вмісту кошика з зазначенням для кожного товару номеру по порядку, інформації про товар, вартості одиниці, кількості, загальної суми. Також повинна бути підрахована загальна сума всіх товарів у кошику.

– Програмне забезпечення повинно надавати можливість редагувати вміст кошика, видаляючи певні товари, або змінюючи кількість певного виду товарів.

– Користувачу повинна надаватися можливість зареєструватися з зазначенням логіну, паролю, П.І.Б., телефону, e-mail та домашньої адреси. Після введення логіна необхідно перевірити його унікальність у базі даних магазину і, якщо він не зайнятий, зберегти інформацію про користувача.

– Також необхідно реалізувати функцію авторизації користувача з введенням логіну та паролю, які повинні перевірятися на відповідність.

– При роботі з кошиком у користувача повинна бути можливість оформити замовлення, при якому необхідно вказати П.І.Б., номер телефону, e-mail та домашню адресу. Усі відомості про користувача разом із переліком замовлених товарів мають фіксуватись у базі даних магазину.

– У програмному забезпеченні, що розробляється, необхідно передбачити можливість для користувача зв'язатися з адміністратором через електронну пошту, вказавши при цьому свої П.І.Б, e-mail та текст повідомлення.

– Користувач повинен мати можливість ознайомитися з інформацією про магазин.

– Програмне забезпечення має надавати детальну інформацію про способи доставки та оплати.

Для адміністратора сайту мають бути реалізовані розширені можливості:

– Редагування списку замовлень, яке включає перегляд списку замовлень, що зберігаються в базі даних, та видалення виконаних замовлень, які зберігаються в базі даних.

– Додавання, оновлення, видалення товарів.

– Додавання, оновлення, видалення категорій.

На сьогоднішньому ринку представлено чимало готових рішень для вебсервісів, які не вимагають значних витрат часу на проєктування та програмування, дозволяючи швидко запустити вебсайт. Більшість із них вирізняються широким функціоналом і сучасним дизайном, що робить їх доволі привабливими. Водночас, такі рішення мають певні обмеження.

Серед мінусів використання готових рішень варто зазначити залежність від стороннього сервісу, надлишковий функціонал, обмежену гнучкість, низьку швидкість, обмежену оптимізацію, складне масштабування, обмежену безпеку. Значною вадою готових рішень є висока вартість, що робить їх малодоступними для малих підприємств. Тому було вирішено розробити власне програмне забезпечення, яке буде значно дешевшим і сприятиме ефективному старту та розвитку бізнесу.

1.3 Постановка задачі на розробку вебсервісу для магазину кави «Kavorra»

Виконавши аналіз практичної задачі інженерії програмного забезпечення та існуючих аналогів, сформулюємо наступну постановку задачі: потрібно розробити вебсервіс для магазину кави «Kavorra», який буде задовольняти нижче сформульованим вимогам.

Вимоги до складу виконуваних функцій

Програмне забезпечення повинно надавати наступні функціональні можливості клієнту сайту та адміністратору сайту:

- Переглянути інформацію про магазин.
- Переглянути інформацію про способи оплати та доставки.
- Зворотній зв'язок.
- Переглянути каталог.
- Вибрати категорію та переглянути товари цієї категорії.
- Знайти товар по назві.
- Розширений пошук товару.
- Переглянути інформацію про товар.

- Додати товар до кошику.
- Переглянути вміст кошика.
- Редагувати вміст кошика.
- Оформити замовлення.
- Реєстрація користувача.
- Авторизація користувача.

Крім того, необхідно реалізувати розширені можливості для адміністратора сайту:

- Редагувати категорію.
- Редагувати товар.
- Редагувати список замовлень.

Вимоги до організації вхідних та вихідних даних

Вхідною інформацією є дані, які вводяться вручну (в залежності від типу поля введення, дані можуть бути: числові цілі, числові десяткові, текстові), або обираються із представлених списків (для уникнення можливих помилок, у випадних списках представлені готові варіанти вибору), або отримуються з бази даних MySQL.

Вихідною інформацією є результати операцій, що виведені у екранні форми, повідомлення та сформовані звіти.

Вхідними даними є наступні:

- дані для реєстрації користувача: логін, пароль, П.І.Б., телефон, e-mail, домашня адреса;
- дані для авторизації користувача: логін, пароль.
- дані про товари каталогу: зображення товару, назва товару, виробник, країна виробник, походження, категорія, сорт, помел, вага, обсмаження, упаковка, ціна, опис;
- дані про категорію каталогу: назва категорії;
- дані про вміст кошика: ідентифікаційний номер товару у каталозі, обрана кількість для кожного товару;
- дані про замовлення: П.І.Б., домашня адреса, номер телефону, адреса

електронної пошти та дані про вміст кошика.

Вимоги до інформаційної та програмної сумісності

Системні програмні засоби повинні задовольняти наступним вимогам:

- операційна система Windows 2000 та вище;
- веб-сервер Apache;
- підтримка PHP;
- СКБД MySQL;
- вихід у мережу Internet.

Вимоги до складу та параметрів технічних засобів

- процесор 1 ГГц і вище;
- оперативна пам'ять 512 МБ і більше;
- дискового простору 5 ГБ і більше;
- дисплей;
- миша.

Повністю постановку задачі на розробку вебсервісу для магазину кави «Kavorra» сформульовано в технічному завданні, яке представлено в додатку А.

2 РОЗРОБКА ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»

З урахуванням результатів аналізу практичної задачі інженерії програмного забезпечення та визначених вимог до майбутнього програмного забезпечення, розпочинається етап розробки та реалізації вебсервісу для магазину кави «Kavorra».

При розробці застосовується об'єктно-орієнтований підхід і використовується уніфікована мова моделювання UML для візуального представлення структури та поведінки програмного забезпечення [4].

2.1 Аналіз вимог до вебсервісу для магазину кави «Kavorra»

У процесі аналізу вимог до вебсервісу для магазину кави «Kavorra» було ідентифіковано основних акторів і сценарії їхньої взаємодії із системою, побудовано діаграму варіантів використання, а також підготовлено відповідні специфікації для кожного з них.

2.1.1 Розробка моделі варіантів використання вебсервісу для магазину кави «Kavorra»

UML-діаграма варіантів використання є графічним інструментом, що відображає можливі сценарії взаємодії користувачів із програмним забезпеченням.

Процес побудови діаграми варіантів використання зазвичай включає: ідентифікацію та опис акторів, визначення та опис варіантів використання (прецедентів), створення самої діаграми, а також розробка специфікацій варіантів використання [4].

В результаті аналізу практичної задачі розробки вебсервісу для магазину кави «Kavorra» були виявлені наступні актори: Користувач сайту та Адміністратор сайту. Стислий опис акторів представлено в таблиці 2.1.

Таблиця 2.1 – Стислий опис акторів вебсервісу для магазину кави

Актор	Стислий опис
Користувач сайту	Особа, яка відвідує вебсервіс магазину з метою перегляду товарів та здійснення покупок.
Адміністратор сайту	Особа, що, крім стандартних можливостей користувача сайту, може змінювати інформацію про товари й категорії та керувати замовленнями.

Функціональні вимоги до розроблюваного програмного забезпечення моделюються як варіанти використання (use cases). Стислий опис виявлених варіантів використання наведено в таблиці 2.2.

Таблиця 2.2 – Стислий опис варіантів використання вебсервісу для магазину кави

Актор	Назва	Формулювання
1	2	3
Адміністратор сайту	Редагувати категорію	Завдяки цьому варіанту використання Адміністратор сайту може редагувати інформацію про категорії.
Адміністратор сайту	Редагувати товар	Завдяки цьому варіанту використання Адміністратор сайту може редагувати інформацію про товари.
Адміністратор сайту	Редагувати список замовлень	Завдяки цьому варіанту використання Адміністратор сайту може редагувати список замовлень клієнтів.
Користувач сайту, Адміністратор сайту	Авторизація	Завдяки цьому варіанту використання актори можуть пройти аутентифікацію.
Користувач сайту, Адміністратор сайту	Реєстрація	Завдяки цьому варіанту використання актори можуть зареєструватися, якщо вони не зареєстровані.

Продовження таблиці 2.2

1	2	3
Користувач сайту, Адміністратор сайту	Переглянути інформацію про магазин	Завдяки цьому варіанту використання актори можуть отримувати довідкову інформацію про магазин.
Користувач сайту, Адміністратор сайту	Переглянути інформацію про способи оплати і доставки	Завдяки цьому варіанту використання актори можуть отримувати довідкову інформацію про способи оплати і доставки.
Користувач сайту, Адміністратор сайту	Зворотній зв'язок	Завдяки цьому варіанту використання актори можуть написати листа адміністрації сайту.
Користувач сайту, Адміністратор сайту	Переглянути каталог	Завдяки цьому варіанту використання актори можуть переглядати інформацію з каталогу.
Користувач сайту, Адміністратор сайту	Вибрати категорію	Завдяки цьому варіанту використання актори можуть переглядати товари окремої категорії.
Користувач сайту, Адміністратор сайту	Знайти товар	Завдяки цьому варіанту використання актори можуть здійснювати пошук товарів в каталозі.
Користувач сайту, Адміністратор сайту	Переглянути інформацію про товар	Завдяки цьому варіанту використання актори можуть переглядати інформацію про окремий товар.
Користувач сайту, Адміністратор сайту	Додати товар до кошика	Завдяки цьому варіанту використання актори можуть додавати товар до кошика.
Користувач сайту, Адміністратор сайту	Переглянути вміст кошика	Завдяки цьому варіанту використання актори можуть переглядати інформацію в кошику.
Користувач сайту, Адміністратор сайту	Редагувати вміст кошика	Завдяки цьому варіанту використання актори можуть редагувати вміст кошика.

Кінець таблиці 2.2

1	2	3
Користувач сайту, Адміністратор сайту	Оформити замовлення	Завдяки цьому варіанту використання актори можуть оформити замовлення на купівлю обраних товарів.

На рисунку 2.1 представлена візуалізація варіантів використання, акторів та зв'язків між ними вебсервісу для магазину кави «Kavorra».

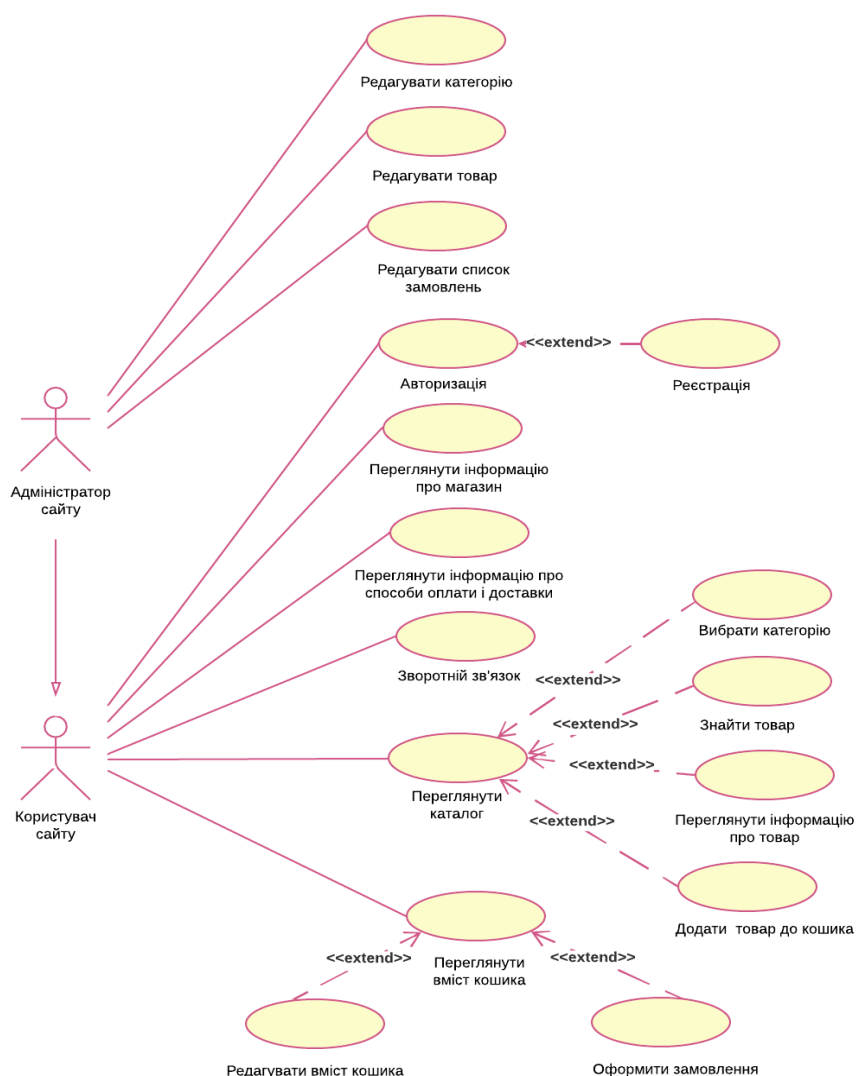


Рисунок 2.1 – Діаграма варіантів використання вебсервісу для магазину кави «Kavorra»

2.1.2 Специфікації прецедентів вебсервісу магазину кави «Kavonra»

Специфікація варіанта використання містить такі основні складові: "Найменування варіанта використання", "Опис варіанта використання", "Дійові особи варіанта використання", "Передумови", "Базовий потік", "Альтернативні потоки", "Постумови". Специфікації варіантів використання розроблюваного програмного забезпечення наведено в таблицях 2.3-2.17 .

Таблиця 2.3 – Специфікація варіанта використання "Редагувати список замовлень"

Складова	Опис
Найменування варіанта використання	Редагувати список замовлень.
Опис варіанта використання	Завдяки цьому варіанту використання Адміністратор може редагувати список замовлень клієнтів, а саме переглядати список замовлень та вилучати вже опрацьовані.
Дійові особи варіанта використання	Адміністратор сайту.
Передумови	Програма запущена. Для ідентифікації Адміністратора сайту потрібна авторизація.
Базовий потік	1. Адміністратор сайту переходить на сторінку редагування замовлень. 2. З'являється список всіх отриманих магазином замовлень з інформацією про кожне замовлення, замовлені товари та кнопкою «Вилучити» для кожного замовлення. 4. Адміністратор сайту натискає на кнопку «Вилучити». 5. Виконується вилучення замовлення з бази даних, виводиться повідомлення «Замовлення було вилучене!» та оновлений список замовлень.
Альтернативні потоки	Не визначені.
Постумови	Не визначені.

Таблиця 2.4 – Специфікація варіанта використання "Редагувати категорію"

Складова	Опис
Найменування варіанта використання	Редагувати категорію.
Опис варіанта використання	Завдяки цьому варіанту використання Адміністратор сайту може редагувати інформацію про категорії, тобто додавати і вилучати категорії товарів, а також змінювати їх назву.
Дійові особи варіанта використання	Адміністратор сайту.
Передумови	Програма запущена. Для ідентифікації Адміністратора сайту потрібна авторизація.
Базовий потік	1. Адміністратор сайту переходить на сторінку редагування категорій. 2. З'являється форма для додавання нової категорії з кнопкою «Додати категорію» та список вже існуючих категорій з кнопками «Зберегти», «Вилучити». 3. Адміністратор сайту вводить назву категорії та натискає на кнопку «Додати категорію». 4. Здійснюється перевірка поля на заповненість та унікальність і, якщо назва не порожня, то до бази даних додається нова категорія з відповідною назвою і виводиться повідомлення "Була створена категорія '...!'“ у випадку оригінальної назви, а в випадку існування назви категорії виводиться повідомлення «Категорія з такою назвою вже існує!».
Альтернативні потоки	1. Якщо Адміністратор сайту змінив в списку назву категорії та натиснув кнопку «Зберегти», то здійснюється перевірка поля на заповненість. Якщо назва не порожня, то до бази даних вносяться відповідні зміни та виводиться повідомлення «Категорія '...' була перейменована в '...!'». 2. Якщо Адміністратор сайту натиснув кнопку «Вилучити» в списку для якоїсь категорії, то виконується її вилучення з бази даних та виводиться повідомлення «Категорія '...' була вилучена!». 3. Якщо при додаванні або редагуванні Адміністратор сайту намагається зберегти категорію з порожньою назвою, то буде виведено повідомлення «Ви не ввели назву категорії!».
Постумови	Не визначені.

Таблиця 2.5 – Специфікація варіанта використання "Реєстрація"

Складова	Опис
Найменування варіанта використання	Реєстрація.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть зареєструватися, якщо вони не зареєстровані.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Програма запущена.
Базовий потік	1. Користувач заходить до сторінки реєстрації. 2. З'являється форма для введення особистої інформації (логін, пароль, П.І.Б., телефон, електронна адреса, домашня адреса) та кнопка «Надіслати дані». 3. Користувач вводить дані та натискає кнопку «Надіслати дані». 4. Відбувається перевірка даних та, у разі успішності, їх занесення до БД з виведенням повідомлення про успішну реєстрацію.
Альтернативні потоки	1. Якщо користувач заповнив не всі поля, то виводиться повідомлення «Ви заповнили не всі поля!» та форма для реєстрації; 2. Якщо користувач ввів некоректну електронну адресу, то виводиться повідомлення «E-mail не коректний!» та форма для реєстрації; 3. Якщо користувач ввів логін, який вже зайнятий іншим користувачем, то виводиться повідомлення «Користувач з таким логіном уже є!» та форма для реєстрації.
Постумови	Не визначені.

Таблиця 2.6 – Специфікація варіанта використання "Авторизація"

Складова	Опис
1	2
Найменування варіанта використання	Авторизація.

Кінець таблиці 2.6

1	2
Опис варіанта використання	Завдяки цьому варіанту використання актори (користувачі) можуть пройти аутентифікацію.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Програма запущена. Користувач має бути зареєстрований.
Базовий потік	1. Користувач заходить до сторінки авторизації. 2. З'являється форма з полями для введення логіну та паролю та кнопкою «Надіслати дані». 3. Користувач вводить логін та пароль та натискає кнопку. 4. Відбувається ідентифікація користувача та виводиться повідомлення про успішну авторизацію.
Альтернативні потоки	1. Якщо пара логін-пароль не знайдена, виводиться повідомлення про помилку та рекомендація повторити введення або пройти реєстрацію у разі її відсутності.
Постумови	Не визначені.

Таблиця 2.7 – Специфікація варіанта використання "Редагувати товар"

Складова	Опис
1	2
Найменування варіанта використання	Редагувати товар.
Опис варіанта використання	Завдяки цьому варіанту використання Адміністратор сайту може редагувати інформацію про товари, тобто додавати і вилучати окремі товари, а також змінювати їх назву та інші параметри.
Дійові особи варіанта використання	Адміністратор сайту.

Кінець таблиці 2.7

1	2
Передумови	Програма запущена. Для ідентифікації Адміністратора сайту потрібна авторизація.
Базовий потік	1. Адміністратор сайту переходить на сторінку редагування товарів. 2. З'являється список вже існуючих товарів з кнопками «Редагувати», «Вилучити» та кнопка «Додати товар». 3. Адміністратор сайту натискає на кнопку «Додати товар». 4. З'являється форма для додавання нового товару з кнопкою «Зберегти товар». 5. Адміністратор сайту вводить інформацію про новий товар та натискає на кнопку «Зберегти товар». 6. Здійснюється перевірка даних і, якщо дані коректні, то до бази даних додається новий товар і виводиться повідомлення «Був створений товар з 'id =...!'».
Альтернативні потоки	1. Якщо при додаванні або зміні існуючого товару Адміністратор сайту залишив порожні поля, то виводиться повідомлення «Ви заповнили не всі поля!». 2. Якщо при виборі зображення товару був обраний файл, формат якого не підтримується, буде виведено повідомлення «Формат зображення не підтримується!» та обране зображення за замовченням. 3. Якщо зображення товару не було обране, то буде обране зображення за замовченням. 4. При натисканні кнопки «Редагувати» для будь-якого товару з'являється та ж сама форма, що і для додавання товару, але на відміну від останньої, при редагуванні в полях форми відразу вписані дані редагованого товару. Після того як користувач ввів дані та натиснув кнопку «Зберегти товар», проводиться перевірка даних. Якщо дані коректні, до бази даних додаються відповідні зміни та виводиться повідомлення «Був змінений товар з id =...!». 5. При натисканні кнопки «Вилучити» для будь-якого товару виконується його вилучення з бази даних та виводиться повідомлення «Був вилучений товар з id =...!».
Постумови	Не визначені.

Таблиця 2.8 – Специфікація варіанта використання "Переглянути інформацію про магазин"

Складова	Опис
Найменування варіанта використання	Переглянути інформацію про магазин.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть переглянути інформацію про магазин.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Програма запущена.
Базовий потік	1. Користувач переходить до сторінки «Про нас». 2. В результаті цих дій на екран виводиться довідкова інформація про магазин.
Альтернативні потоки	Не визначені.
Постумови	Не визначені.

Таблиця 2.9 – Специфікація варіанта використання "Переглянути інформацію про способи оплати і доставки"

Складова	Опис
Найменування варіанта використання	Переглянути інформацію про способи оплати і доставки.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть отримати інформацію про способи оплати і доставки.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Програма запущена.
Базовий потік	1. Користувач переходить до сторінки «Способи оплати і доставки». 2. В результаті цих дій на екран виводиться інформація про способи оплати і доставки в магазині.
Альтернативні потоки	Не визначені.
Постумови	Не визначені.

Таблиця 2.10 – Специфікація варіанта використання "Зворотній зв'язок"

Складова	Опис
Найменування варіанта використання	Зворотній зв'язок.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть написати листа адміністрації сайту.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Програма запущена.
Базовий потік	1. Користувач заходить до сторінки зворотнього зв'язку 2. З'являється форма, яка містить поля вводу особистої інформації (П.І.Б., електронна адреса), поле для вводу повідомлення та кнопку «Надіслати дані». 3. Користувач вводить дані та натискає кнопку «Надіслати дані». 4. Відбувається перевірка даних на коректність. Якщо перевірка пройшла успішно, виконується відправка введеного повідомлення на email адміністрації сайту, та копії цього повідомлення на email відправника. Також виводиться повідомлення «Ваше повідомлення надіслане».
Альтернативні потоки	1. Якщо відправник не заповнив поле з повідомленням, виводиться повідомлення «Введіть текст повідомлення» та форма зворотнього зв'язку. 2. Якщо відправник ввів некоректний email, виводиться повідомлення «E-mail не коректний!» та форма зворотнього зв'язку.
Постумови	Не визначені.

Таблиця 2.11 – Специфікація варіанта використання "Переглянути каталог"

Складова	Опис
1	2
Найменування варіанта використання	Переглянути каталог.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть переглядати інформацію з каталогу. Даний варіант використання може бути розширений за допомогою варіантів використання: Вибрати категорію, Знайти товар, Переглянути інформацію про товар, Додати товар до кошика.

Кінець таблиці 2.11

1	2
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Програма запущена.
Базовий потік	1. Користувач заходить до головної сторінки вебсервісу магазину. 2. В центральній частині сторінки відображаються всі товари каталогу з кнопкою «В кошик» для кожного товару. Також передбачена посторінкова навігація.
Альтернативні потоки	1. При відсутності товару на складі для нього, замість кнопки "В кошик", з'явиться текст «Немає на складі».
Постумови	Не визначені.

Таблиця 2.12 – Специфікація варіанта використання "Вибрати категорію"

Складова	Опис
Найменування варіанта використання	Вибрати категорію.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть переглядати товари окремої категорії.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Програма запущена.
Базовий потік	1. Користувач вибирає окрему категорію у меню категорій, яке знаходиться на кожній сторінці сайту. 2. В центральній частині сторінки відображаються всі товари категорії з кнопками «В кошик» та «Докладніше» для кожного товару. Також передбачена посторінкова навігація.
Альтернативні потоки	1. Якщо обрана категорія порожня, буде виведено повідомлення: «В цій категорії немає товарів». 2. При відсутності товару на складі для нього, замість кнопки «В кошик», з'явиться текст «Немає на складі».
Постумови	Не визначені.

Таблиця 2.13 – Специфікація варіанта використання "Знайти товар"

Складова	Опис
Найменування варіанта використання	Знайти товар.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть здійснювати пошук товарів в каталозі.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Програма запущена.
Базовий потік	1. Користувач натискає на кнопку «Розширений пошук», яка знаходиться на кожній сторінці сайту. 2. З'являється сторінка, яка містить форму для введення параметрів пошуку та кнопки «Знайти» та «Очистити форму». 3. Користувач вводить значення параметрів пошуку, жоден з яких не є обов'язковим, і натискає кнопку «Знайти». 4. На тій же самій сторінці виводяться результати пошуку з кнопкою «В кошик» для кожного товару та можливістю переглянути по назві товару детальну інформацію про товар.
Альтернативні потоки	1. Користувач вводить фразу в поле швидкого пошуку, котре є на кожній сторінці сайту, та натискає кнопку «Знайти». При швидкому пошуку в назві товару шукається довільне слово з введеного користувачем запиту. Відповідь виводиться на сторінці розширеного пошуку, перехід до якої здійснюється автоматично. 2. Якщо користувач натиснув кнопку «Знайти», не ввівши жодного параметру, або по запиту нічого не знайшлося, буде виведено повідомлення «По вашому запиту нічого не знайдено». 3. При відсутності товару на складі для нього, замість кнопки «В кошик», з'явиться текст «Немає на складі».
Постумови	Не визначені.

Таблиця 2.14 – Специфікація варіанта використання "Оформити замовлення"

Складова	Опис
Найменування варіанта використання	Оформити замовлення.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть оформити замовлення на купівлю обраних товарів.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	1. В кошику повинен бути товар. 2. Користувач повинен знаходитись на сторінці відображення вмісту кошика.
Базовий потік	1. Користувач натискає кнопку «Оформити замовлення», яка знаходиться на сторінці перегляду кошика. 2. Відкривається сторінка з формою для вводу особистої інформації користувача (П.І.Б., телефон, електронна адреса, домашня адреса) та кнопкою «Надіслати дані». 3. Користувач вводить свої дані в форму та натискає кнопку. 4. Відбувається перевірка даних на коректність. Якщо перевірка пройшла успішно, в базу даних додається інформація про замовлення, яка складається з П.І.Б., телефону, електронної адреси, домашньої адреси користувача, дати замовлення, загальної суми замовлення та інформації про обрані товари (ідентифікатор товару у базі даних, його кількість та ціна на момент замовлення). Також прибирається форма замовлення, очищується кошик та виводиться повідомлення «Ваше замовлення № ... прийняте!».
Альтернативні потоки	1. Якщо користувач попередньо авторизувався, в полях вводу відображаються його дані, взяті з бази даних магазину. Користувачу лишається лише підтвердити їх, натиснувши кнопку «Надіслати дані». 2. Якщо користувач заповнив не всі поля, виводиться повідомлення «Ви заповнили не всі поля!» та форма для повторного вводу. 3. Якщо користувач ввів некоректний email, виводиться повідомлення «E-mail не коректний!» та форма для повторного вводу.
Постумови	Не визначені.

Таблиця 2.15 – Специфікація варіанта використання "Додати товар до кошика"

Складова	Опис
Найменування варіанта використання	Додати товар до кошика.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть додавати товар до кошика.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Користувач повинен знаходитись на сторінці з товарами (сторінці каталогу, сторінці з інформацією про товар або сторінці результатів пошуку).
Базовий потік	1. Користувач натискає кнопку «В кошик» для будь-якого товару. 2. До кошика додається одна одиниця обраного товару. 3. Автоматично оновлюється вміст маленького кошика, який відображається на кожній сторінці сайту та служить своєрідним індикатором, за яким користувач може орієнтуватися в кількості відкладених товарів та загальній сумі замовлення.
Альтернативні потоки	1. Якщо кошик ще порожній, тобто не був ініціалізований, то створення кошика відбувається автоматично.
Постумови	Не визначені.

Таблиця 2.16 – Специфікація варіанта використання "Переглянути вміст кошика"

Складова	Опис
1	2
Найменування варіанта використання	Переглянути вміст кошика.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть переглядати інформацію в кошику.

Кінець таблиці 2.16

1	2
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	Програма запущена.
Базовий потік	1. Користувач натискає на посилання «Перейти до кошику», яке міститься на будь-якій сторінці сайту. 2. Відкривається окрема сторінка, яка містить інформацію про вміст кошика в формі таблиці з кнопкою «Вилучити» для кожного товару. Кількість кожного товару відображена у полі для вводу та може бути змінена. Під таблицею із вмістом кошика є кнопки «Перерахувати» та «Очистити кошик». Окремо вказана загальна сума всього замовлення та кнопка «Оформити замовлення».
Альтернативні потоки	1. У разі пустого кошика буде виведено повідомлення «В вашому кошику немає товарів!».
Постумови	Не визначені.

Таблиця 2.17 – Специфікація варіанта використання "Редагувати вміст кошика"

Складова	Опис
1	2
Найменування варіанта використання	Редагувати вміст кошика.
Опис варіанта використання	Завдяки цьому варіанту використання користувачі можуть редагувати вміст кошика, а саме видаляти або змінювати кількість для кожного товару в кошику.
Дійові особи варіанта використання	Користувач сайту, Адміністратор сайту.
Передумови	В кошику повинні бути товари.

Кінець таблиці 2.17

1	2
Базовий потік	1. Користувач, що знаходиться на сторінці перегляду кошика, змінює кількість товару в спеціальному полі або відмічає товар для вилучення та натискає кнопку «Перерахувати». 2. Змінюється кількість товарів в кошику та перераховуються суми для кожного товару та для замовлення взагалі.
Альтернативні потоки	1. Якщо користувач натискає кнопку «Очистити кошик», то всі товари видаляються та виводиться повідомлення «В вашому кошику немає товарів!». 2. Якщо при зміні кількості товару користувач ввів число, яке не є цілим, то за кількість буде взято найближче менше ціле; 3. Якщо при зміні кількості товару користувач ввів від'ємне число, то кількість буде вважатися рівною нулю, а товар буде видалений з кошика.
Постумови	Не визначені.

2.2 Розробка архітектури вебсервісу для магазину кави «Kavorra»

Згідно з вимогами до вебсервісу для магазину кави «Kavorra», було обрано архітектурний стиль, який найбільш відповідає поставленим задачам, – MVC (Model-View-Controller).

Основними перевагами MVC є:

– Чітке розділення відповідальності

Model (Модель) відповідає за бізнес-логіку та роботу з даними (наприклад, управління товарами, замовленнями, користувачами). Вона незалежна від інтерфейсу користувача.

View (Представлення) відповідає за відображення даних користувачу. Воно не містить бізнес-логіки.

Controller (Контролер) виконує роль посередника між Моделлю та Представленням. Він приймає запити від користувача, обробляє їх, звертається до Моделі для отримання або оновлення даних, а після цього обирає відповідне Представлення для виведення результатів користувачеві.

В програмному забезпеченні є багато різних сутностей (товари, категорії товарів, кошик, замовлення). MVC дозволяє чітко розмежувати логіку управління цими сутностями від їх відображення, що спрощує розробку та підтримку.

– Полегшення підтримки та масштабованості

Завдяки чіткому розділенню, зміни в одному компоненті (наприклад, зміна дизайну інтерфейсу у View) зазвичай не вимагають значних змін в інших компонентах (Model або Controller). Це значно спрощує подальшу підтримку та еволюцію додатку.

Вебсервіси постійно розвиваються: додаються нові функції, змінюється дизайн, інтегруються нові способи оплати та доставки. MVC дозволяє легше вносити ці зміни без ризику порушити вже працюючі частини.

– Зручність тестування

Кожен компонент MVC можна тестувати ізольовано. Бізнес-логіка в Моделі може бути перевірена без необхідності візуалізації через View, а логіка Контролера може бути протестована незалежно від View.

Критично важливо забезпечити надійну роботу всіх функцій вебсервісу магазину (кошик, замовлення, реєстрація). Можливість легкого модульного тестування в MVC дозволяє виявити та виправити помилки на ранніх етапах розробки, забезпечуючи високу якість продукту.

– Повторне використання коду

Одна Модель може бути використана з декількома Представленнями (наприклад, веб-інтерфейс, мобільний додаток), що зменшує дублювання коду.

Якщо в майбутньому знадобиться створити мобільний додаток для партнерів, одна і та ж бізнес-логіка (Модель) може бути використана повторно, що економить час та ресурси.

– Зручність для веб-розробки

MVC добре узгоджується з циклом запит-відповідь у веб-додатках. Контролер отримує запит, Модель обробляє дані, а Представлення формує відповідь.

Вебсервіс є типовим веб-додатком, тому MVC є природним вибором, який дозволяє ефективно керувати взаємодією між клієнтом (браузером) і сервером.

Таким чином, архітектура MVC є оптимальним вибором для розробки вебсервісу для магазину кави «Kavorra» завдяки своїй здатності забезпечити чітке розділення відповідальності, легкість підтримки, масштабованість, та ефективне тестування.

2.2.1 Діаграма компонентів вебсервісу магазину кави «Kavorra»

Діаграма компонентів (Component Diagram) в UML призначена для моделювання фізичної структури програмної системи.

Основним призначенням діаграми компонентів являється [4]:

- відображення структури системи;
- демонстрація залежностей та зв'язків;
- управління складністю (розбивка системи на логічні та фізичні частини);
- планування розгортання;
- розуміння повторного використання;
- визначення меж системи;
- комунікація між командами.

Діаграму компонентів вебсервісу для магазину кави «Kavorra» представлено на рисунку 2.2.

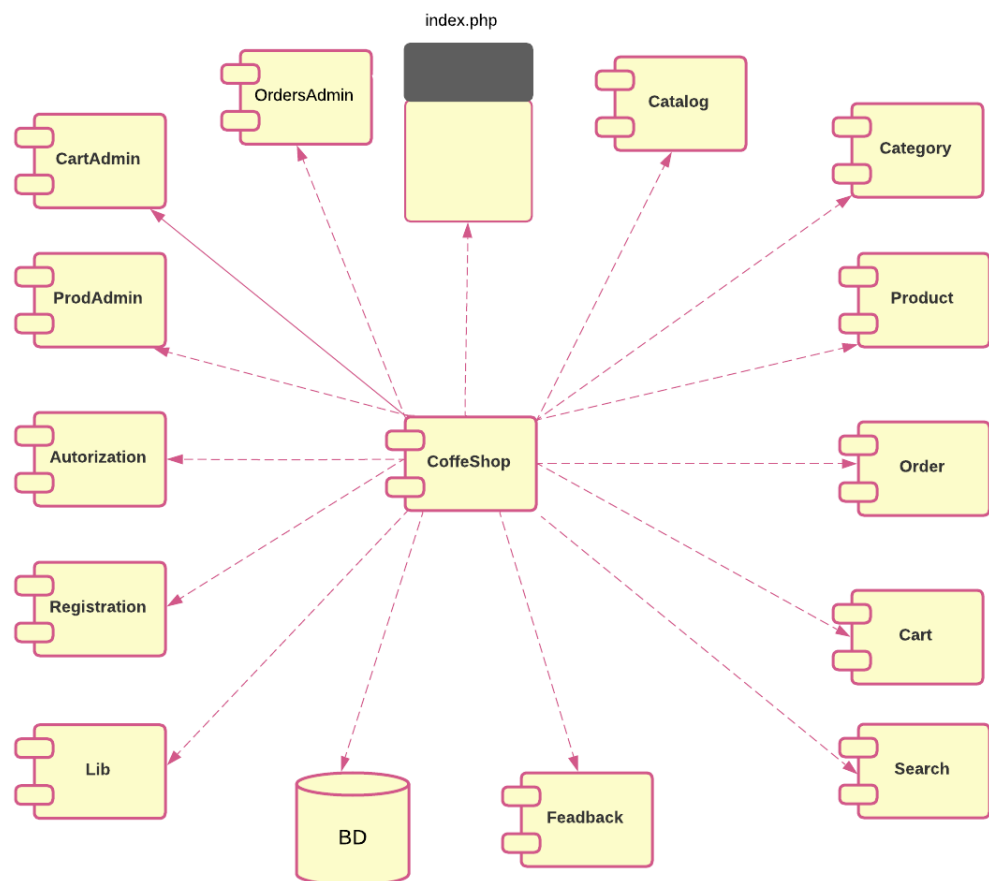


Рисунок 2.2 – Діаграма компонентів вебсервісу для магазину кави «Kavorra»

Діаграма компонентів для вебсервісу магазину показує такі основні компоненти:

- компонент, що надає інформацію про товари;
- компонент для управління кошиком користувача;
- компонент для обробки та керування замовленнями;
- компонент для управління користувачами та їх доступом;
- компонент, що зберігає дані (база даних).

2.2.2 Діаграма розгортання вебсервісу магазину кави «Kavorra»

Діаграма розгортання (Deployment Diagram) призначена для моделювання фізичного розміщення програмних компонентів на вузлах апаратної інфраструктури. Простіше кажучи, вона показує, де і як програмне забезпечення працює у реальному фізичному середовищі.

На рисунку 2.3 представлено діаграму розгортання вебсервісу для магазину кави «Kavorra».

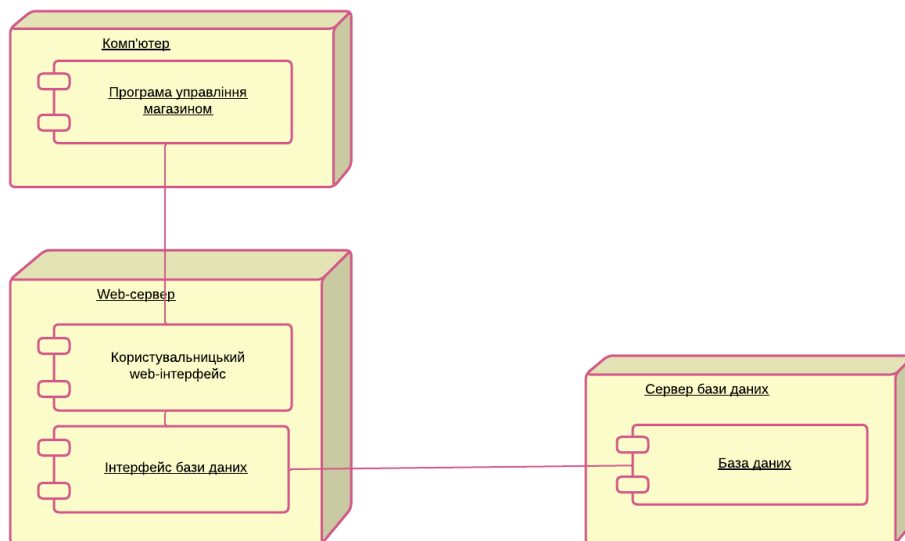


Рисунок 2.3 – Діаграма розгортання програмного забезпечення вебсервісу для магазину кави «Kavorra»

Діаграми розгортання використовуються для:

- візуалізації фізичної архітектури;
- демонстрації розгортання компонентів;
- визначення мережових взаємодій;
- комунікації між командам.

2.3 Проєкт вебсервісу для магазину кави «Kavorra»

В підрозділі по проєктуванню програмного забезпечення представлені ескізний, технічний та робочий проєкти.

2.3.1 Ескізний проєкт вебсервісу магазину кави «Kavorra»

На етапі ескізного проєктування були побудовані діаграми діяльності для основних прецедентів. Ці діаграми стали основою для розробки проєкту користувацького інтерфейсу, а також для створення концептуальної моделі даних програмної системи.

Створення діаграм діяльності для ключових варіантів використання вебсервісу магазину кави «Kavorra»

Діаграми діяльності (Activity Diagrams) – це графічні представлення потоку робіт, бізнес-процесів або послідовності дій у програмній системі. Вони наочно демонструють, як певні дії виконуються, які умови впливають на їхній хід і як процеси можуть розгалужуватися чи виконуватися паралельно. Ці діаграми є важливим інструментом для візуалізації динамічної поведінки системи, допомагаючи зрозуміти логіку виконання та взаємодію її складових.

Основними елементами діаграм діяльності є: початковий вузол (Initial Node), кінцевий вузол діяльності (Activity Final Node), дія (Action/Activity), потік керування (Control Flow), Вузол рішення (Decision Node), вузол злиття (Merge Node), вузол розгалуження / Розділення (Fork Node), Вузол об'єднання / Синхронізації (Join Node), Доріжки (Swimlanes). Ці елементи дозволяють детально моделювати навіть найскладніші процеси, роблячи їх зрозумілими та легкими для аналізу [4].

На рисунках 2.4 – 2.6 зображено діаграми діяльності для варіантів використання «Вибрати категорію», «Редагувати список замовлень» та «Переглянути каталог» відповідно.



Рисунок 2.4 – Діаграма діяльності для варіанту використання «Вибрати категорію»

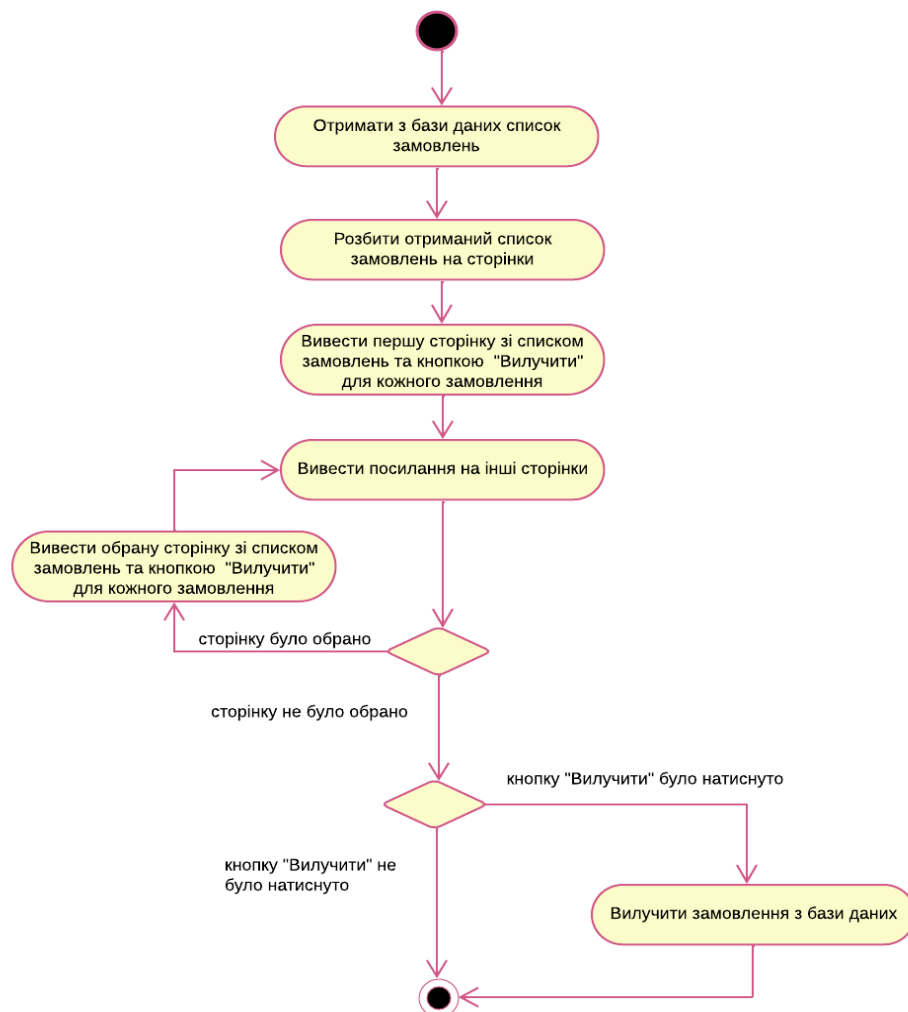


Рисунок 2.5 – Діаграма діяльності для прецеденту «Редагувати список замовлень»

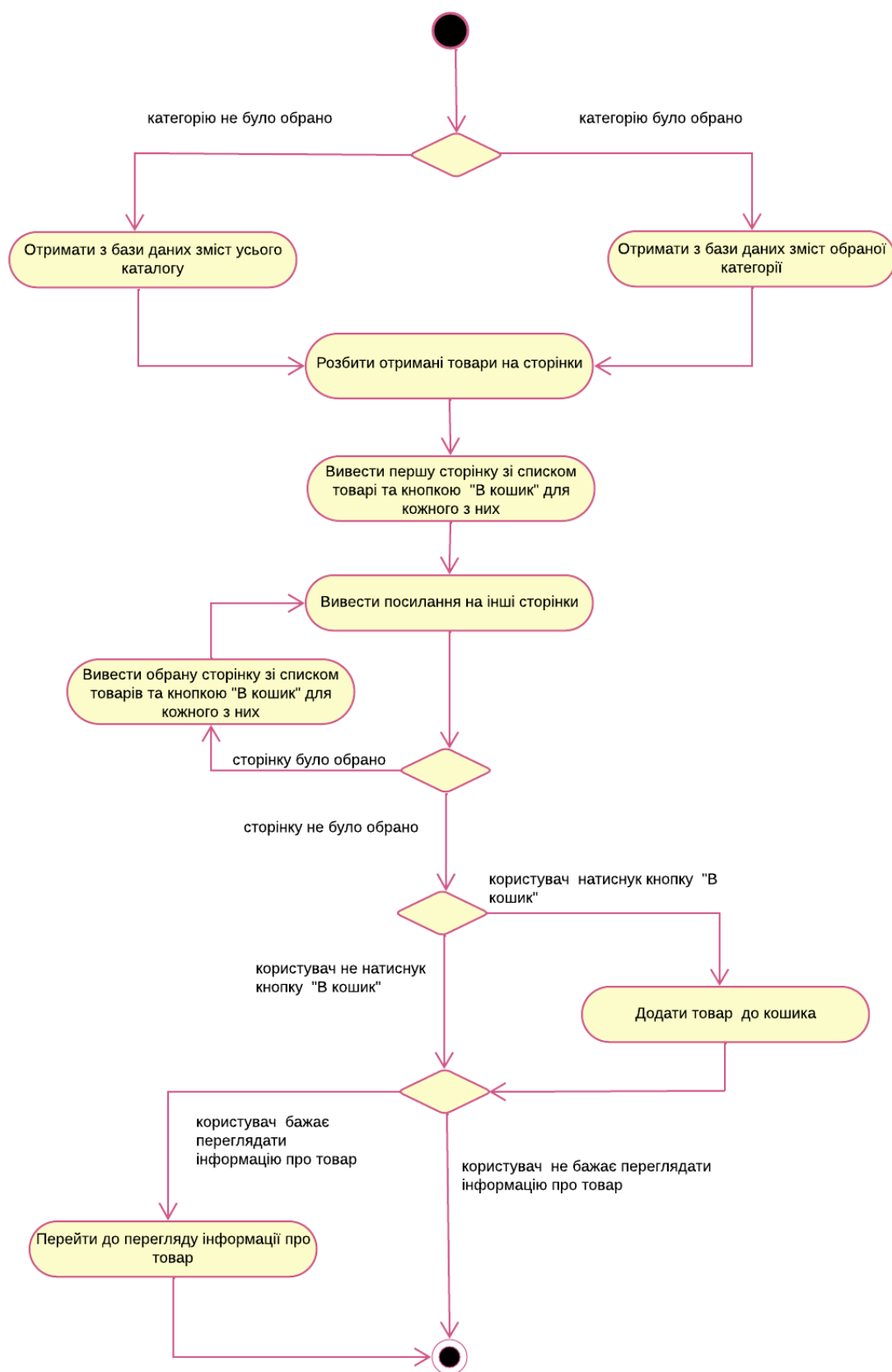


Рисунок 2.6 – Діаграма діяльності для прецеденту «Переглянути каталог»

Концептуальна модель даних вебсервісу магазину кави «Kavorra»

Концептуальна модель даних – це представлення структури даних, яке описує сутності, атрибути та зв'язки між ними без урахування реалізації (СУБД).

В результаті проведення аналізу практичної задачі інженерії вебсервісу для магазину кави «Kavorra» була побудована концептуальна модель даних у вигляді діаграми класів, яка наведена на рисунку 2.7.

Вона містить наступні класи-сутності User (Користувач), Order (Замовлення), Product (Товар), Category (Категорія товарів), Cart (Кошик).

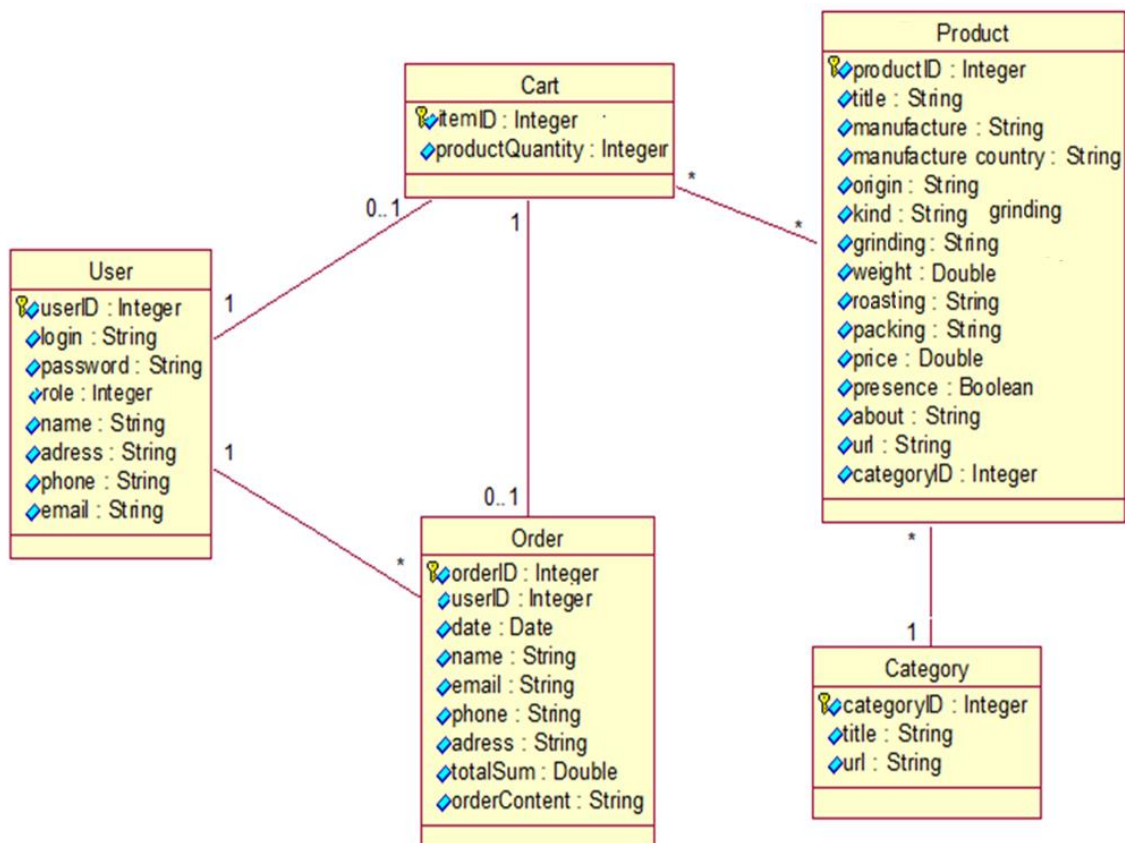


Рисунок 2.7 – Концептуальна модель даних вебсервісу магазину кави «Kavorra»

Основне призначення концептуальної моделі [5]:

– Формалізація предметної області

Модель дозволяє описати, які об'єкти існують у системі, які між ними зв'язки, і які характеристики вони мають.

- Єдина точка розуміння для розробників та замовника
Концептуальна модель зрозуміла як технічним фахівцям, так і бізнес-користувачам – вона служить основою для узгодження вимог.
- Підготовка до логічного та фізичного проектування бази даних
Концептуальна модель – це проміжний етап між описом процесів та створенням структури таблиць у СУБД.
- Уникнення помилок на ранніх етапах розробки
Завдяки моделюванню можна заздалегідь виявити невідповідності або неповноту вимог.

Розробка проєкту користувальницького інтерфейсу

Користувацький інтерфейс – це засіб взаємодії відвідувача з вебсервісом магазину, який реалізується за допомогою набору програмних компонентів, що забезпечують його функціонування. Він охоплює всі візуальні елементи, з якими користувач стикається при вході на сайт: тексти, зображення, кнопки, панелі навігації, меню тощо.

Інтерфейс не лише представляє товарний асортимент, а й демонструє весь спектр функціональних можливостей, доступних потенційному покупцеві, та шляхи їх використання. Саме через інтерфейс здійснюється навігація, вибір товарів, оформлення замовлень і взаємодія з сервісними функціями.

На рисунках 2.8-2.13 представлений проєкт користувальницького інтерфейсу.

Для забезпечення зручної навігації на кожній сторінці магазину міститься:

- логотип та назва магазину;
головне меню сайту, яке складається з пунктів: «Каталог», «Оплата і доставка», «Про нас», «Зворотній зв'язок», «Реєстрація», «Авторизація» (для авторизованого користувача замість пункту «Авторизація» з'являється ім'я користувача та рядом посилання «Вийти»);
- список категорії кави у вигляді окремого меню;

- вміст маленького кошика (загальна кількість і сума товарів у кошику, посилання на сторінку роботи з кошиком «Перейти до кошику»);
- поле з кнопкою для швидкого пошуку товару та посилання на сторінку розширеного пошуку.

При швидкому пошуку в назві товару шукається довільне зі слів з введеного користувачем запиту. Відповідь виводиться на сторінці розширеного пошуку, перехід до якої здійснюється автоматично.

На рисунку 2.8 представлено сторінку перегляду каталогу, яка є головною сторінкою сайту. Ця сторінка реалізує функції варіанту використання «Переглянути каталог». В центральній частині сторінки відображаються всі товари каталогу або лише товари окремої категорії, обраної в меню категорій.

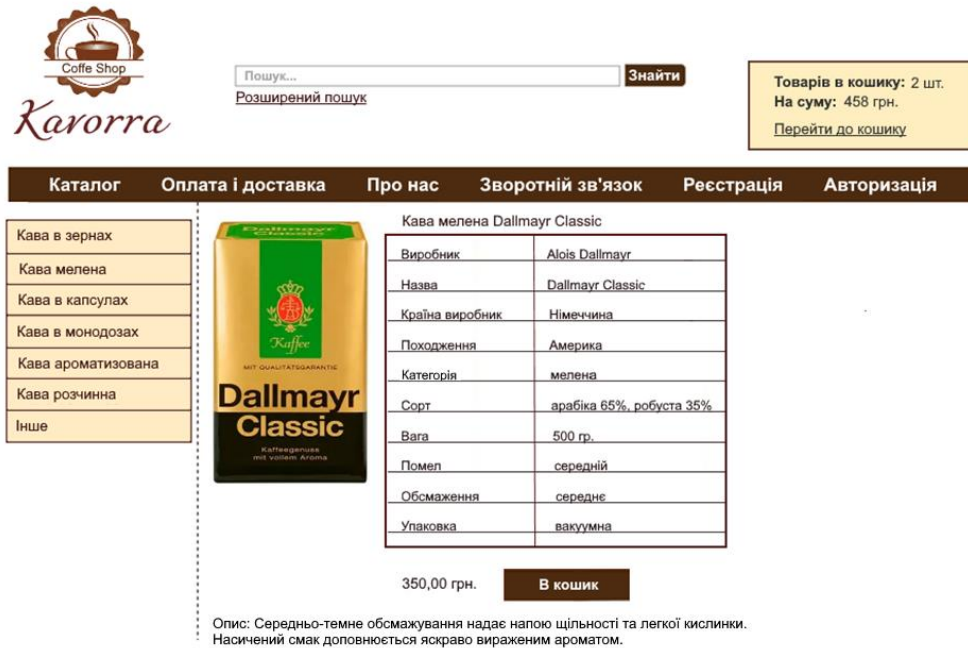
Для кожного товару вказується зображення товару, виробник, категорія, назва, вага, ціна, кнопка «В кошик» та посилання «Докладніше», яке допомагає перейти на сторінку з інформацією про цей товар. Також передбачена зручна посторінкова навігація.



Рисунок 2.8 – Сторінка перегляду каталогу вебсервісу магазину кави «Kavorra»

На рисунку 2.9 представлено сторінку перегляду детальної інформації про товар, яка містить інформацію про окремий товар. Ця сторінка реалізує функції варіанту використання «Переглянути інформацію про товар».

Для товару вказується зображення товару, назва, виробник, країна виробник, походження, категорія, сорт, помел, вага, обсмаження, упаковка, ціна, опис та кнопка «В кошик».



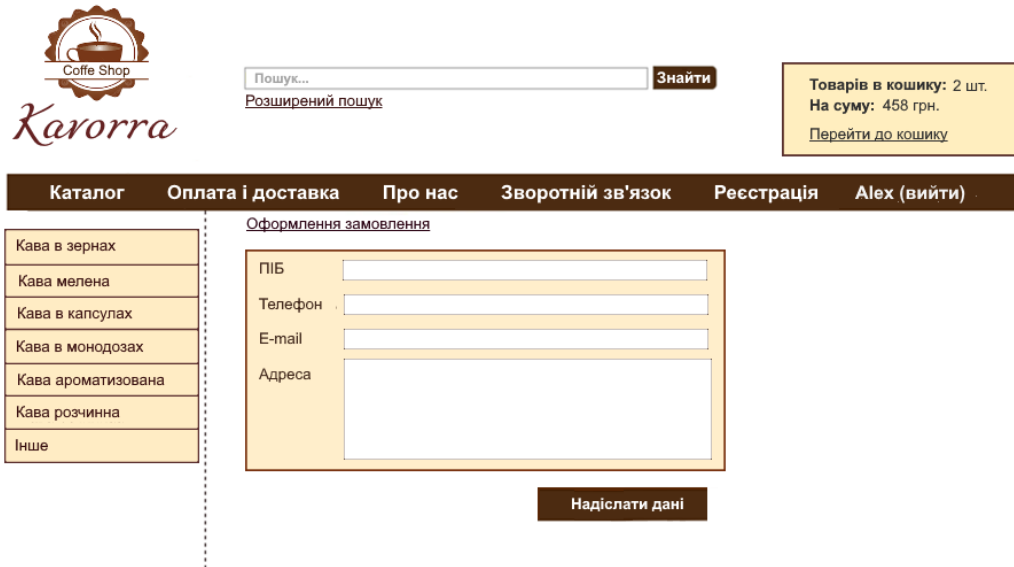
The screenshot shows the Kavorra website interface. At the top left is the Kavorra logo with a coffee cup icon. To its right is a search bar with the text "Пошук..." and a "Знайти" button. Below the search bar is a link "Розширений пошук". On the top right, a yellow box displays "Товарів в кошику: 2 шт. На суму: 458 грн." and a link "Перейти до кошику". Below the header is a navigation bar with links: "Каталог", "Оплата і доставка", "Про нас", "Зворотній зв'язок", "Реєстрація", and "Авторизація". On the left side, there is a vertical menu with categories: "Кава в зернах", "Кава мелена", "Кава в капсулах", "Кава в монодозах", "Кава ароматизована", "Кава розчинна", and "Інше". The main content area displays the product "Кава мелена Dallmayr Classic". To the left of the product name is an image of the coffee packaging. To the right is a table with product details:

Виробник	Alois Dallmayr
Назва	Dallmayr Classic
Країна виробник	Німеччина
Походження	Америка
Категорія	мелена
Сорт	арабіка 65%, робуста 35%
Вага	500 гр.
Помел	середній
Обсмаження	середнє
Упаковка	вакуумна

Below the table, the price "350,00 грн." is shown next to a "В кошик" button. At the bottom, there is a description: "Опис: Середньо-темне обсмажування надає напою щільності та легкої кислинки. Насичений смак доповнюється яскраво вираженим ароматом."

Рисунок 2.9 – Сторінка перегляду детальної інформації про товар вебсервісу магазину кави «Kavorra»

На рисунку 2.10 представлено сторінку оформлення замовлення. Ця сторінка реалізує функції варіанту використання «Оформити замовлення». Вона містить форму для вводу особистої інформації користувача (П.І.Б., телефон, електронна адреса, домашня адреса) та кнопку «Надіслати дані». Якщо користувач попередньо авторизувався, в полях вводу відображаються його дані, взяті з бази даних магазину. Користувачу лишається лише підтвердити їх, натиснувши кнопку «Надіслати дані».



Каворра Coffe Shop

Пошук... Знайти

Розширений пошук

Товарів в кошику: 2 шт.
На суму: 458 грн.
[Перейти до кошику](#)

Каталог Оплата і доставка Про нас Зворотній зв'язок Реєстрація Alex (вийти)

Оформлення замовлення

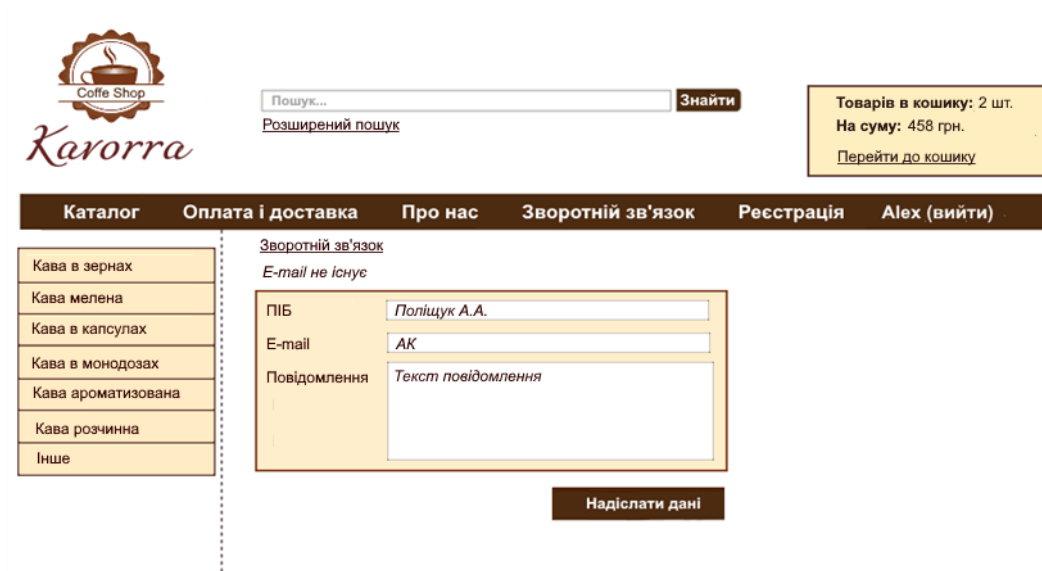
Кава в зернах
Кава мелена
Кава в капсулах
Кава в монодозах
Кава ароматизована
Кава розчинна
Інше

PIB
Телефон
E-mail
Адреса

Надіслати дані

Рисунок 2.10 – Сторінка оформлення замовлення вебсервісу магазину кави «Kavorra»

На рисунку 2.11 представлено сторінку зворотнього зв'язку, за допомогою якої користувач має змогу написати листа адміністрації сайту.



Каворра Coffe Shop

Пошук... Знайти

Розширений пошук

Товарів в кошику: 2 шт.
На суму: 458 грн.
[Перейти до кошику](#)

Каталог Оплата і доставка Про нас Зворотній зв'язок Реєстрація Alex (вийти)

Зворотній зв'язок
E-mail не існує

Кава в зернах
Кава мелена
Кава в капсулах
Кава в монодозах
Кава ароматизована
Кава розчинна
Інше

PIB: Поліщук А.А.
E-mail: АК
Повідомлення: Текст повідомлення

Надіслати дані

Рисунок 2.11 – Сторінка зворотнього зв'язку вебсервісу магазину кави «Kavorra»

Ця сторінка реалізує функції варіанта використання «Зворотній зв'язок». Вона містить форму з полями вводу особистої інформації користувача (П.І.Б.,

електрона адреса), полем для вводу повідомлення та кнопкою «Надіслати дані». При вводі некоректних даних буде з'являтися повідомлення про помилку.

На рисунку 2.12 представлено сторінку реєстрації. Ця сторінка реалізує функцію варіанту використання «Реєстрація». Вона містить форму для вводу особистої інформації користувача (логін, пароль, П.І.Б., телефон, електрона адреса, домашня адреса) та кнопку «Надіслати дані».

The screenshot displays the registration page of the Kavorra website. At the top left is the Kavorra logo with a coffee cup icon. Next to it is a search bar with the text 'Пошук...' and a 'Знайти' button. To the right, a box shows the shopping cart status: 'Товарів в кошику: 2 шт. На суму: 458 грн.' and a link 'Перейти до кошику'. Below the header is a navigation bar with links: 'Каталог', 'Оплата і доставка', 'Про нас', 'Зворотній зв'язок', 'Реєстрація', and 'Авторизація'. The main content area is titled 'Реєстрація' and includes a warning: 'Ви заповнили не всі поля!'. The registration form contains the following fields: 'Логін' (filled with 'Alex'), 'Пароль', 'ПІБ' (filled with 'Поліщук А.А.'), 'Телефон' (filled with '093-41-37-371'), 'E-mail', and 'Адреса'. A 'Надіслати дані' button is located at the bottom right of the form. On the left side, there is a sidebar with a list of coffee products: 'Кава в зернах', 'Кава мелена', 'Кава в капсулах', 'Кава в монодозах', 'Кава ароматизована', 'Кава розчинна', and 'Інше'.

Рисунок 2.12 – Сторінка реєстрації вебсервісу магазину кави «Kavorra»

Для користувача, який авторизувався на правах адміністратора на всіх сторінках додатково відображається панель адміністрування. Вона має наступні пункти: «Перегляд магазину», «Замовлення», «Товари», «Категорії», «Вийти».

Якщо обраний пункт «Перегляд магазину», то адміністратор може переглядати всі сторінки та користуватися всіма функціями доступними для звичайного користувача. Єдина різниця у наявності панелі адміністрування у верхній частині екрану.

На рисунку 2.13 представлено панель адміністрування на прикладі сторінки перегляду каталогу.

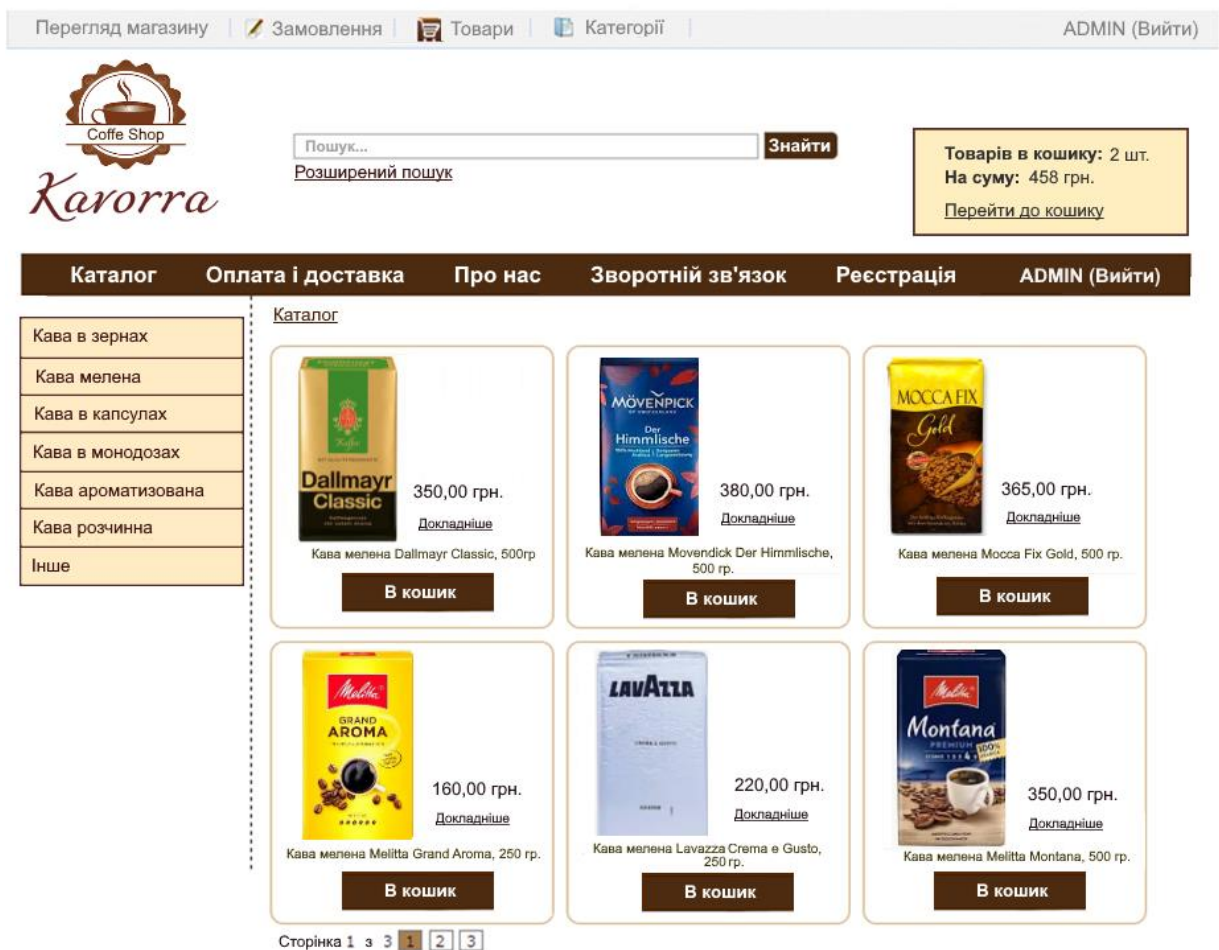


Рисунок 2.13 – Сторінка перегляду каталогу з панеллю адміністрування у верхній частині

Якщо обраний пункт «Категорії», то адміністратор має змогу додавати і видаляти окремі категорії каталогу, а також змінювати їх назву.

При виборі пункту «Товари» адміністратор має змогу додавати і вилучати окремі товари до каталогу, а також змінювати їх назву та інші параметри.

Також адміністратор має змогу працювати зі списком замовлень. За допомогою відповідної сторінки адміністратор може переглядати список замовлень та вилучати вже опрацьовані.

2.3.2 Технічний проєкт вебсервісу магазину кави «Kavorra»

У результаті аналізу ескізного проєкту був розроблений технічний проєкт даного ПЗ, що включає статичну та динамічну моделі системи. Статична модель

реалізована у вигляді діаграми класів, яка описує основні класи, їхні атрибути, методи та взаємозв'язки між класами. Динамічна модель представлена діаграмами послідовності, які відображають сценарії взаємодії об'єктів у процесі виконання функцій системи. Окрім того, на цьому етапі було сформовано логічну модель даних, що деталізує структуру концептуальної моделі даних та враховує обрану модель даних [4].

Статична модель вебсервісу магазину кави «Kavorra»

На основі розроблених моделей варіантів використання, концептуальної моделі даних та проєкту користувацького інтерфейсу було створено статичну модель програмного забезпечення. Її візуалізація у вигляді діаграми класів представлена на рисунку 2.14.

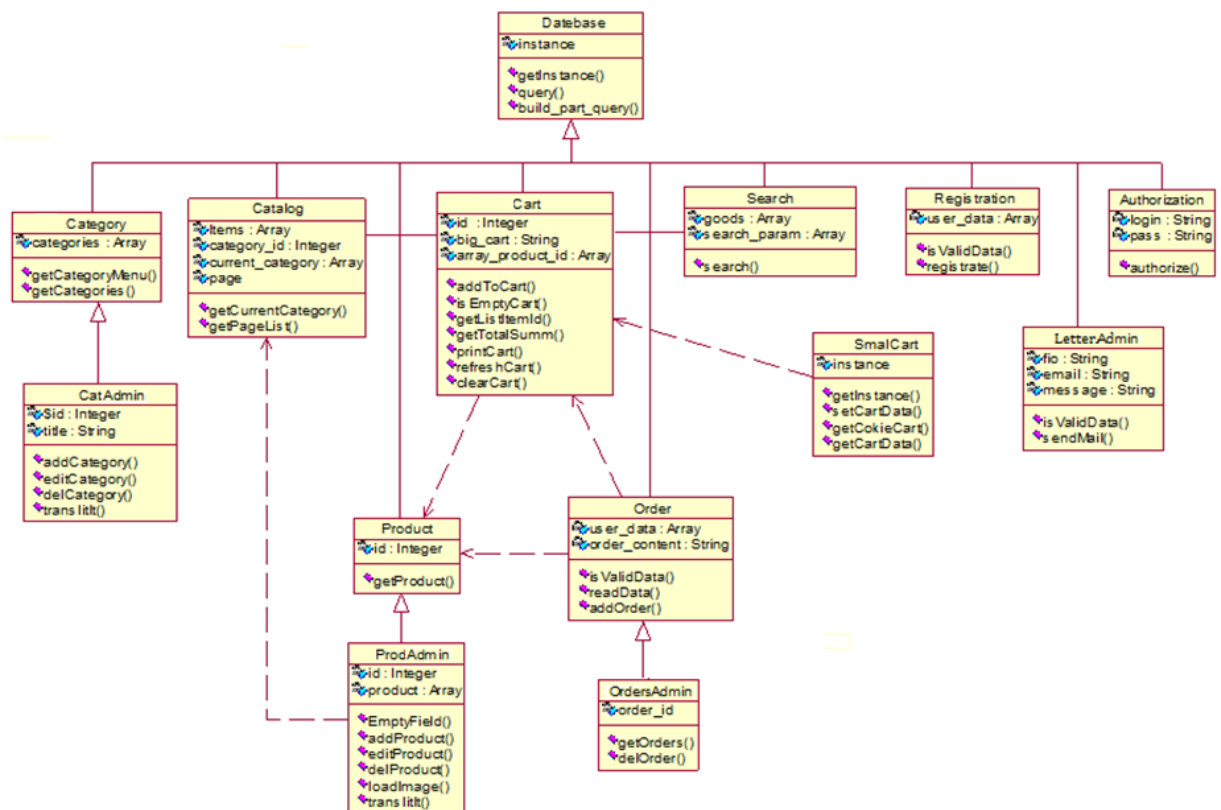


Рисунок 2.14 – Статична модель вебсервісу магазину кави «Kavorra»

Специфікації класів вебсервісу магазину кави «Kavorra»

Специфікація класу «LetterAdmin»

Ім'я: LetterAdmin

Відповідальність: даний клас відповідає за реалізацію зворотнього зв'язку.

Атрибути:

fio – П.І.Б. користувача;

email – електронна адреса користувача;

message – текст повідомлення.

Операції:

isValidData(fio, email, message) – цей метод спрацьовує при виконанні методу sendMail(). У результаті запуску даного методу виконується перевірка даних на валідність (коректність email, заповненість полів).

sendMail() – цей метод використовується для реалізації зворотнього зв'язку. У результаті запуску даного методу виконується перевірка даних на валідність (isValidData()) та створюється відправка на email листа адміністратору сайту та копії користувачу сайту.

Специфікація класу «Registration»

Ім'я: Registration

Відповідальність: даний клас відповідає за реєстрацію користувача.

Атрибути:

user_data – масив з даними користувача.

Операції:

isValidData(user_data) – цей метод спрацьовує при отриманні даних від користувача. У результаті запуску даного методу виконується перевірка даних на валідність (коректність email, наповненість полів).

registrate(user_data) – цей метод спрацьовує при реєстрації користувача. У результаті запуску даного методу виконується перевірка даних на валідність (метод isValidData()) та додавання запису з даними про користувача до бази даних.

Специфікація класу «Authorization»

Ім'я: Authorization

Відповідальність: даний клас відповідає за авторизацію користувача.

Атрибути:

login – логін користувача;

pass – пароль користувача.

Операції:

authorize(login, pass) – цей метод спрацьовує при авторизації користувача.

У результаті запуску даного методу виконується пошук відповідної пари логін-пароль у базі даних. В разі успіху виконується ідентифікація користувача. Для користувача, який авторизувався як адміністратор сайту, надається доступ до панелі адміністрування.

Специфікація класу «Database»

Ім'я: Database

Відповідальність: даний клас відповідає за захист програмного забезпечення від SQL-ін'єкцій (один з поширених способів злому сайтів і програм, що працюють з базами даних, заснований на впровадженні в запит довільного SQL-коду).

Атрибути:

instance – екземпляр об'єкту класу Database (захищається від створення другого екземпляру за допомогою патерну Singleton).

Операції:

getInstance() – цей метод спрацьовує при спробі створити екземпляр об'єкту класу Database. У результаті запуску даного методу повертається вже існуючий екземпляр об'єкту класу Database, в разі його відсутності, створюється новий екземпляр об'єкту.

query(query) – цей метод спрацьовує при створенні запиту до бази даних. В якості параметрів виступає текст запиту (query) та параметри, що підставляються до запиту (не обов'язково). У результаті запуску даного методу виконується перевірка на кількість отриманих параметрів. Якщо прийшло більше ніж один параметр (крім тексту запиту прийшли параметри для підстановки), параметри для підстановки обробляються (екрануються спец.символи) та підставляються у текст запиту з урахуванням очікуваних типів даних. Далі виконується запит до бази даних. В разі отримання лише одного параметру (текст запиту) запит виконується відразу.

`build_part_query(array, devide = ',')` – цей метод використовується для створення частини тексту складного SQL-запиту. В якості параметрів виступає асоціативний масив параметрів запиту та розділювач (за замовченням кома). У результаті запуску даного методу виконується обробка (екрануються спец. символи) отриманих параметрів та створюється частина тексту SQL-запиту.

Специфікація класу «Category»

Ім'я: Category

Відповідальність: даний клас відповідає за роботу з категоріями каталогу.

Атрибути:

`categories` – многовимірний масив, який містить інформацію про всі категорії.

Операції:

`getCategories()` – цей метод спрацьовує при визові методу `getCategoryMenu()`. У результаті запуску даного методу створюється многовимірний масив `categories`, який містить інформацію про всі категорії.

`getCategoryMenu(categories)` – цей метод спрацьовує при виведенні списку категорій на екран у вигляді меню. У результаті запуску даного методу створюється список категорій, придатний до використання в меню (назви категорій обрамляються html-тегами).

Специфікація класу «CatAdmin»

Ім'я: CatAdmin

Відповідальність: даний клас відповідає за роботу адміністратору з категоріями

Атрибути:

`id` – ідентифікатор категорії;

`title` – стара назва категорії.

Операції:

`addCategory(title)` – цей метод спрацьовує при додаванні нової категорії. В якості параметрів виступає назва категорії. У результаті запуску даного методу за допомогою методу `translitIt()` класу `Category` отримується `url` (представлення

в адресній строчці). Враховуючи отриману назву та url створюється нова категорія в базі даних.

`editCategory(id, title)` – цей метод спрацьовує при зміні назви вже існуючої категорії. В якості параметрів виступають ідентифікатор категорії та її назва. У результаті запуску даного методу за допомогою методу `translitIt()` класу `Category` отримується url для нової назви. Враховуючи отриману інформацію вносяться зміни до опису категорії в базі даних.

`delCategory(id, title)` – цей метод спрацьовує при видаленні категорії. В якості параметрів виступає ідентифікатор категорії та її назва. У результаті запуску даного методу виконується вилучення категорії з бази даних.

`translitIt(title)` – цей метод спрацьовує при обчисленні url категорії (представлення в адресній строчці). В якості параметру виступає назва категорії. У результаті запуску даного методу виконується транслітерація назви категорії.

Специфікація класу «Catalog»

Ім'я: `Catalog`

Відповідальність: даний клас відповідає за відображення товарів каталогу.

Атрибути:

`items` – многовимірний масив для зберігання інформації про товари;

`category_id` – ідентифікатор поточної категорії;

`current_category` – масив для зберігання інформації про поточну категорію;

`page` – поточна сторінка каталогу.

Операції:

`getCurrentCategory(category_id)` – цей метод спрацьовує при обиранні окремої категорії під час перегляду каталогу. У результаті запуску даного методу до масиву `current_category` записується вся інформація про поточну категорію.

`getPageList(current_category, page=1, step=6)` – цей метод спрацьовує при виведенні списку товарів на екран. В якості параметрів виступають інформація

про поточну категорію, номер сторінки та бажана кількість товарів на одній сторінці. У результаті запуску даного методу список товарів поточної категорії розбивається на сторінки. До масиву `items` записується інформація про вміст товарів, які потрапили на обрану сторінку.

Специфікація класу «Cart»

Ім'я: `Cart`

Відповідальність: даний клас відповідає за роботу з кошиком.

Атрибути:

`id` – ідентифікатор товару у базі даних;

`big_cart` – частина html-коду для представлення кошику;

`array_product_id` – многовимірний масив, ключами якого є `id` товарів, а значеннями кількість цих товарів.

Операції:

`addToCart(id)` – цей метод спрацьовує при додаванні товару до кошику. У результаті запуску даного методу до кошику додається одна одиниця обраного товару.

`isEmptyCart()` – цей метод спрацьовує при визові методу `printCart()`. У результаті запуску даного методу виконується перевірка кошику на заповненість та повертається відповідь `true` чи `false`.

`getListItemId()` – цей метод спрацьовує при визові методів `getTotalSumm()` та `printCart()`. У результаті запуску даного методу отримується список `id` всіх товарів з кошику.

`getTotalSumm()` – цей метод спрацьовує при визові методу `printCart()`. У результаті запуску даного методу отримується сума всіх товарів з кошику.

`printCart()` – цей метод спрацьовує при виведенні вмісту кошику на екран. У результаті запуску даного методу виконується перевірка кошику на наповненість (метод `isEmptyCart()`), отримується список `id` всіх товарів з кошику (метод `getListItemId()`), для отримання інформації про кожний товар визивається метод `getProduct()` класу `Product` для кожного отриманного `id`, для отримання загальної суми всіх товарів з кошику визивається метод

getTotalSumm (). Використовуючи отриманні дані формується частина html-коду для представлення кошику та записується в big_cart.

refreshCart(array_product_id) – цей метод спрацьовує при зміні вмісту кошику. У результаті запуску даного методу виконується відновлення інформації про вміст кошику.

clearCart() – цей метод спрацьовує при натисканні на кнопку «Очистити корзину» або при вдалому оформленні замовлення. У результаті запуску даного методу виконується видалення всіх товарів з кошику.

Специфікація класу «SmalCart»

Ім'я: SmalCart

Відповідальність: даний клас відповідає за відображення маленького кошику.

Атрибути:

instance – екземпляр об'єкту класу SmalCart (захищається від створення іншого екземпляру за допомогою патерну Singleton).

Операції:

getInstance() – цей метод спрацьовує при спробі створити екземпляр об'єкту класу SmalCart. У результаті запуску даного методу повертається вже існуючий екземпляр об'єкту класу SmalCart, в разі його відсутності створюється новий екземпляр об'єкту.

setCartData() – цей метод спрацьовує при зміні вмісту кошику. У результаті запуску даного методу виконується серіалізація та запис інформації про вміст кошику в cookie.

getCokieCart() – цей метод спрацьовує при виконанні методу getCartData(). У результаті запуску даного методу виконується десеріалізація та отримання інформації про вміст кошику з cookie.

getCartData() – цей метод спрацьовує при оновленні будь-якої сторінки. У результаті запуску даного методу отримується інформація про id та кількість кожного товару кошику (метод getCokieCart()), отримується інформація про ціну товарів з бази даних та обраховується загальна сума замовлення. Отримана інформація виводиться на екран у вигляді «маленького кошику».

Специфікація класу «Product»

Ім'я: Product

Відповідальність: даний клас відповідає за отримання інформації про окремий товар.

Атрибути:

id – ідентифікаційний номер продукту в базі даних.

Операції:

getProduct(id) – цей метод спрацьовує при переході на сторінку з інформацією про окремий товар. У результаті запуску даного методу отримується інформація з бази даних про обраний продукт та виводиться на екран.

Специфікація класу «ProdAdmin»

Ім'я: ProdAdmin

Відповідальність: даний клас відповідає за редагування адміністратором товарів з каталогу.

Атрибути:

id – ідентифікаційний номер продукту в базі даних;

product – масив для зберігання отриманих від користувача параметрів продукту.

Операції:

emptyField(product) – цей метод спрацьовує при додаванні нового або редагуванні вже існуючого товару. У результаті запуску даного методу виконується перевірка масиву product на пусті значення.

addProduct(product) – цей метод спрацьовує при додаванні нового товару до каталогу. У результаті запуску даного методу отримуються атрибути товару від користувача, перевіряються на наповненість (метод emptyField()), зберігається зображення обкладинки (метод loadImage()), з назви товару отримується його url (метод translitIt ()). З урахуванням всіх отриманих параметрів, додається інформація про товар до бази даних.

editProduct(product, id) – цей метод спрацьовує при редагуванні вже існуючого товару з каталогу. У результаті запуску даного методу отримуються

атрибути товару від користувача, перевіряються на наповненість (метод `emptyField()`), зберігається зображення обкладинки (метод `loadImage()`), з назви товару отримується його url (метод `translitIt()`). З урахуванням всіх отриманих параметрів, вносяться зміни до інформації про товар в базі даних.

`delProduct(id)` – цей метод спрацьовує при видаленні товару з каталогу. У результаті запуску даного методу виконується вилучення обраного товару з бази даних.

`loadImage()` – цей метод спрацьовує при додаванні нового або редагуванні вже існуючого товару. У результаті запуску даного методу виконується перевірка (розширення, розмір) файлу та його збереження до папки `uploads`.

`translitIt(title)` – цей метод спрацьовує при додаванні нового або редагуванні вже існуючого товару, використовується для отримання url товару (представлення в адресному рядку). В якості параметру виступає назва товару. У результаті запуску даного методу виконується транслітерація назви товару.

Специфікація класу «Order»

Ім'я: `Order`

Відповідальність: даний клас відповідає за оформлення замовлення.

Атрибути:

`user_data` – асоціативний масив даних про замовника;

`order_content` – строка для збереження серіалізованого вмісту кошику.

Операції:

`isValidData(user_data)` – цей метод спрацьовує при виконанні методу `addOrder()`. У результаті запуску даного методу виконується перевірка даних, введених користувачем, на валідність (коректність email, заповнення всіх полів).

`readData()` – цей метод спрацьовує при виконанні методу `addOrder()`. У результаті запуску даного методу виконується перевірка проходження авторизації. Для авторизованих користувачів виконується отримання інформації про замовника з бази даних. Отримана інформація виводиться на екран для підтвердження користувачем.

`addOrder()` – цей метод спрацьовує при оформленні замовлення. У результаті запуску даного методу виконується метод `readData()`, перевіряється

валідність отриманих користувачем даних (метод `isValidData()`), одержується інформація про вміст кошику та суму замовлення (метод `getTotalSumm()` класу `Cart`). Після цього дані про вміст кошику серіалізуються в строку, вся отримана інформація (дані про замовника, серіалізований вміст кошику, загальна сума) додається до бази даних.

Специфікація класу «OrdersAdmin»

Ім'я: `OrdersAdmin`

Відповідальність: даний клас відповідає за роботу адміністратору з замовленнями.

Атрибути:

`order_id` – ідентифікаційний номер замовлення в базі даних.

Операції:

`getOrders()` – цей метод спрацьовує при виведенні списку отриманих магазином замовлень на екран. У результаті запуску даного методу виконується отримання інформації про замовлення з бази даних, десеріалізація вмісту кошику. Для кожного продукту з кошику отримується інформація з бази даних (метод `getProduct()` класу `Product`).

`delOrder(order_id)` – цей метод спрацьовує при видаленні замовлення з загального списку. У результаті запуску даного методу виконується вилучення обраного замовлення з бази даних.

Специфікація класу «Search»

Ім'я: `Search`

Відповідальність: даний клас відповідає за здійснення пошуку товарів з каталогу.

Атрибути:

`search_param` – асоціативний масив з параметрами для пошуку;

`goods` – багатовимірний масив зі знайденими товарами.

Операції:

`search(search_param)` – цей метод спрацьовує при здійсненні пошуку товарів з каталогу. У результаті запуску даного методу з отриманих від користувача параметрів формується запит до бази даних. В разі знаходження

товару, який задовольняє всім вимогам, створюється масив з отриманими товарами goods.

Динамічна модель вебсервісу магазину кави «Kavorra»

Динамічна модель вебсервісу магазину кави розроблена у вигляді діаграм послідовності.

Діаграма послідовності показує, як об'єкти взаємодіють між собою в певному сценарії. Вона відображає всі важливі події, що відбуваються з об'єктами (наприклад, коли вони з'являються чи зникають), а також повідомлення, які вони надсилають один одному, щоб керувати цією взаємодією. [4].

На рисунку 2.15 представлено діаграму послідовності для варіанту використання «Переглянути каталог».

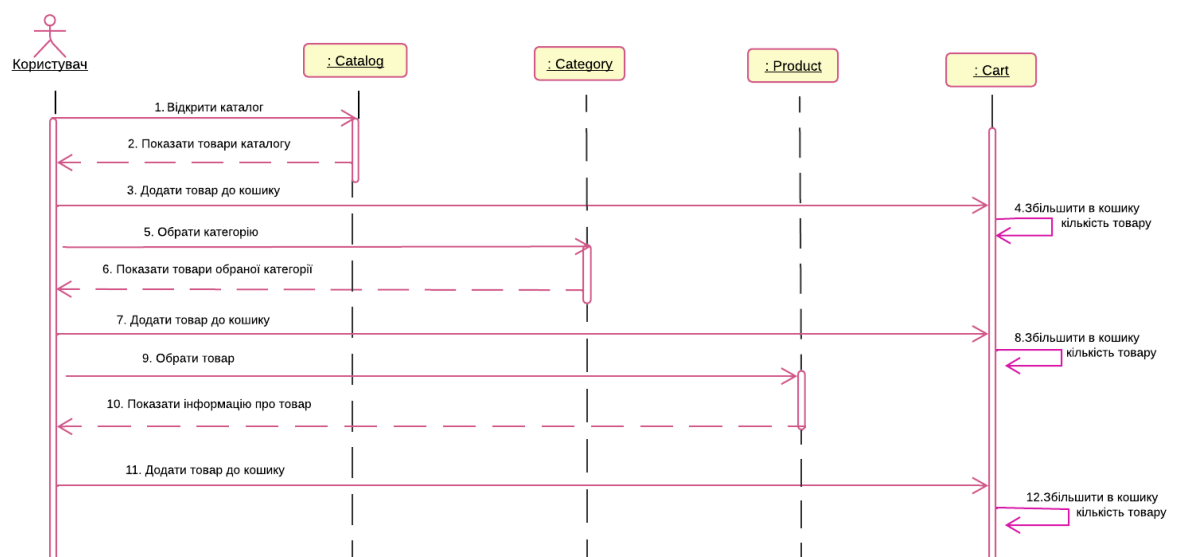


Рисунок 2.15 – Діаграма послідовності для варіанту використання «Переглянути каталог»

На рисунку 2.16 представлено діаграму послідовності для варіанту використання «Редагувати категорію».

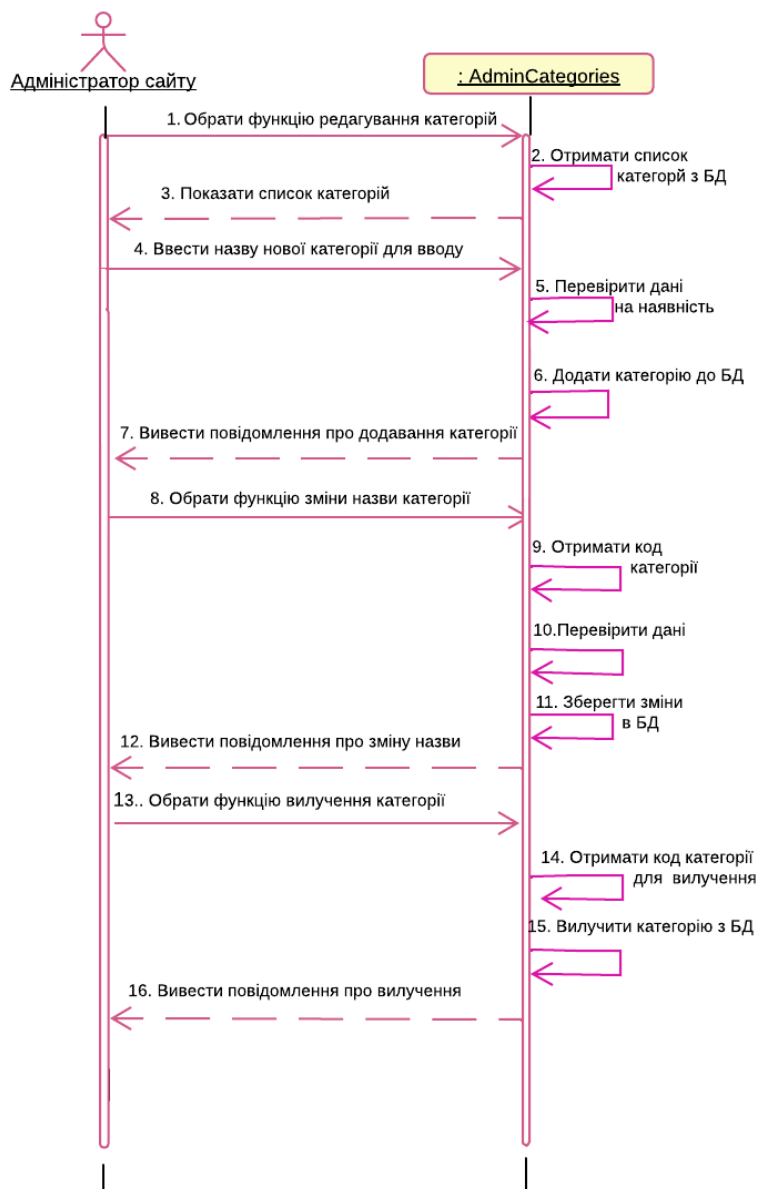


Рисунок 2.16 – Діаграма послідовності для варіанту використання «Редагувати категорію»

Розробка логічної моделі даних вебсервісу для магазину кави «Kavorra»

У процесі побудови логічної моделі даних ключову роль у формуванні структури даних відіграє поняття функціональної залежності (ФЗ). Визначення функціональних залежностей і формування мінімального їх набору дає змогу створити оптимальну структуру даних, яка забезпечує ефективне зберігання й обробку інформації. Це досягається за допомогою еквівалентних перетворень схем відношень у рамках реляційної моделі. Такий процес має назву нормалізації. Основним завданням логічного проєктування є деталізація

концептуальної моделі даних з урахуванням обраної моделі даних – реляційної, ієрархічної, мережевої тощо [5].

Під час побудови логічної моделі були визначені всі типи атрибутів, для кожного відношення були визначені ключі, для кожного атрибуту було визначено обов'язковість заповнення. Таблиці логічної моделі даних представлені в таблицях 2.18 – 2.23.

Таблиця 2.18 – Таблиця «Product»

Назва атрибуту	Тип	Ключ
productID	Integer	PK
title	String	
manufacture	String	
manufacture country	String	
origin	String	
kind	String	
grinding	String	
weight	Double	
roasting	String	
packing	String	
price	Double	
presence	Boolean	
about	String	
url	String	
categoryID	Integer	FK

Таблиця 2.19 – Таблиця «Category»

Назва атрибуту	Тип	Ключ
categoryID	Integer	PK
title	String	
url	String	

Таблиця 2.20 – Таблиця «User»

Назва атрибуту	Тип	Ключ
userID	Integer	ПК
login	String	
password	String	
role	Integer	
name	String	
email	String	
phone	Integer	
adress	String	

Таблиця 2.21 – Таблиця «Order»

Назва атрибуту	Тип	Ключ
orderID	Integer	ПК
userID	Integer	FK
date	Date	
name	String	
email	String	
phone	Integer	
adress	String	
totalSum	Неціле	
orderContent	String	

Таблиця 2.22 – Таблиця «CartProduct»

Назва атрибуту	Тип	Ключ
itemID	Integer	ПК
productID	String	ПК
productQuantity	Integer	

Таблиця 2.23 – Таблиця «Cart»

Назва атрибуту	Тип	Ключ
itemID	Integer	ПК

Створену логічну модель даних вебсервісу магазину кави «Kavorra» наведено на рисунку 2.17.

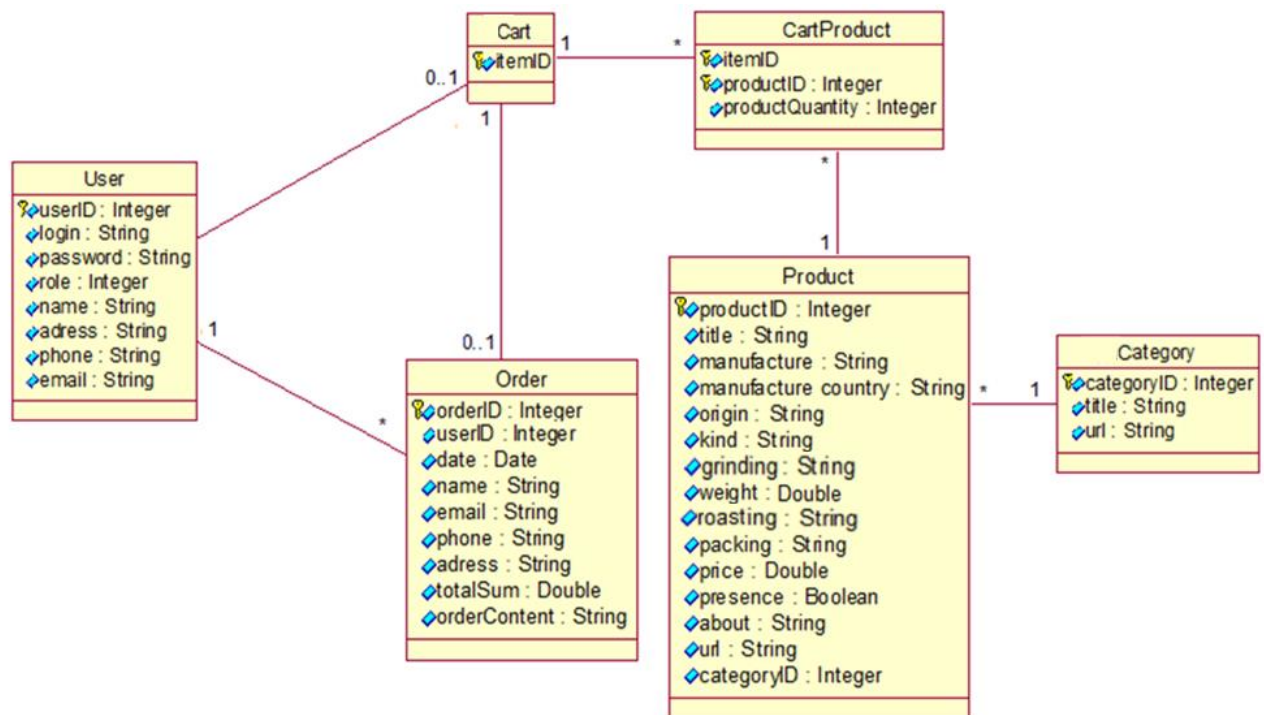


Рисунок 2.17 – Логічна модель даних вебсервісу магазину кави «Kavorra»

Логічна модель даних знаходиться у третій нормальній формі.

2.3.3 Робочий проєкт вебсервісу магазину кави «Kavorra»

На стадії робочого проєкту рішення, прийняті на попередніх етапах, перетворюються у форму, доступну користувачеві. Вибираються засоби розробки програмного забезпечення. Складаються програми, здатні розв'язати задачу.

На етапі робочого проєктування рішення, ухвалені на попередніх стадіях, набувають форми, доступної користувачеві. Визначаються інструменти та мови розробки, які будуть застосовані для реалізації програмного забезпечення. Розробляються програми, призначені для розв'язання поставленої задачі.

Крім того, на цьому етапі проводяться тестування та випробування програмного продукту для перевірки його працездатності та відповідності вимогам.

Обґрунтування вибору засобів розробки вебсервісу магазину кави «Kavorra».

Для розробки вебсервісу магазину кави «Kavorra» використовувалися наступні засоби розробки:

– Мова PHP

Основними перевагами PHP є: простота інтеграції з HTML – PHP-скрипти легко впроваджуються безпосередньо у HTML-розмітку; підтримка великої кількості СКБД – таких як MySQL, PostgreSQL, Oracle; платформна незалежність – PHP працює на більшості операційних систем (Windows, Linux, macOS); широка спільнота та безліч готових рішень – фреймворки, CMS, бібліотеки, плагіни.

Сьогодні PHP активно використовується для побудови різноманітних вебрішень: від простих особистих сайтів до великих корпоративних порталів, CRM, ERP, інтернет-магазинів (наприклад, OpenCart, Magento) і навіть популярних CMS (наприклад, WordPress, Joomla, Drupal).

– СКБД MySQL

Однією з найпопулярніших сучасних систем керування базами даних є MySQL, яка виступає як альтернатива платним комерційним рішенням. Вона широко застосовується, в першу чергу, для розробки динамічних веб-сторінок, оскільки має підтримку з боку різноманітних мов програмування, зокрема і PHP. MySQL відповідає ключовим вимогам, що висуваються до систем керування базами даних під час створення веб-сайтів, оскільки вирізняється надійністю, високою продуктивністю, простотою встановлення та зручністю у використанні. Завдяки своїй доступності, швидкодії та рівню безпеки, MySQL забезпечує ефективну взаємодію з базами даних через Інтернет.

– HTML

HTML відповідає за структуру вебсторінки. За її допомогою задаються елементи сторінки, такі як заголовки, абзаци, таблиці, зображення, форми тощо. HTML формує логічний каркас контенту, визначаючи, які елементи будуть присутні на сторінці та в якому порядку.

– CSS

CSS – це мова стилів, яка призначена для візуального оформлення вебсторінок. Вона задає зовнішній вигляд HTML-елементів, зокрема кольори, шрифти, розміри, відступи, розташування на сторінці та анімаційні ефекти. Основою роботи CSS є принцип каскадності: стилі можуть наслідуватися від батьківських елементів, а також перекриватися, що забезпечує гнучке та точне налаштування дизайну сайту.

Реалізація та тестування вебсервісу магазину кави «Kavonra»

З метою перевірки відповідності програмного забезпечення встановленим вимогам та супровідній документації було проведено процедуру тестування.

Тестування – це процес запуску програмного коду з метою виявлення потенційних помилок. Головна мета тестування полягає не стільки в демонстрації коректної роботи програми, скільки в пошуку дефектів. Вихідним припущенням є те, що будь-яке програмне забезпечення, особливо складне, може містити помилки. Завдання тестування – виявити якомога більше таких помилок до введення системи в експлуатацію.

У процесі тестування найбільша увага приділяється створенню якісних та ефективних тестів. Це пов'язано з тим, що повне тестування складного програмного забезпечення є практично недосяжним, а отже, жоден набір тестів не може гарантувати абсолютну відсутність помилок. Тому основним завданням під час вибору тестів є зменшення цього обмеження шляхом охоплення якомога ширшого кола можливих сценаріїв роботи системи.

Існують дві основні стратегії тестування програмного забезпечення:

- Функціональне тестування, яке зосереджується на перевірці вхідних та вихідних даних без урахування внутрішньої структури коду.

- Структурне тестування, що передбачає аналіз логіки виконання та внутрішніх алгоритмів програми.

До основних методів функціонального тестування належать:

- еквівалентне розбиття,
- аналіз граничних значень,
- типові помилки,
- побудова функціональних діаграм.

Серед методів структурного тестування виділяють:

- покриття операторів,
- покриття рішень,
- покриття умов,
- комбінації рішень/умов,
- комбінаторне тестування умов.

У межах цієї кваліфікаційної роботи було застосовано метод еквівалентного розбиття, що дозволило ефективно перевірити коректність функціонування програмних компонентів. Усі модулі були повністю протестовані і в ході перевірки помилок не виявлено. Через великий обсяг програмного коду та враховуючи складність програмних модулів підсистеми прийнято рішення помістити в роботу тільки незначну частину тестів.

Розглянемо тестування функцій «Редагувати вміст кошика», «Редагувати категорію», «Переглянути каталог».

У таблиці 2.24 наведено класи еквівалентності вхідних даних для функції «Редагувати вміст кошика».

Таблиця 2.24 – Правильні та неправильні класи еквівалентності вхідних даних функції «Редагувати вміст кошика»

Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
Дані про кількість одиниць окремого товару	1. Введено ціле додатне число	2. Введено число, менше за нуль 3. Введено дробове число 4. В кошику не лишилось товарів

У таблиці 2.25 наведено класи еквівалентності вхідних даних для функції «Редагувати категорію».

Таблиця 2.25 – Правильні та неправильні класи еквівалентності вхідних даних функції «Редагувати категорію»

Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
Назва категорії	1. Назва введена коректно	2. Назва порожня 3. Така назва вже існує

У таблиці 2.26 наведено класи еквівалентності вхідних даних для функції «Переглянути каталог».

Таблиця 2.26 – Правильні та неправильні класи еквівалентності вхідних даних функції «Переглянути каталог»

Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
Дані про обрану категорію	1. Категорія обрана та непорожня	2. Категорія не обрана 3. Категорія порожня

Тести для класів еквівалентності вхідних даних розглянутих функцій представлені в таблицях 2.27, 2.28, 2.29.

Таблиця 2.27 – Тести для класів еквівалентності з таблиці 2.24

№	Вхідні дані	Результат
1	Введено ціле додатне число	Кількість дорівнює вхідним даним
2	Введено число, менше за нуль	Кількість буде рівною нулю, а товар буде видалений з кошика
3	Введено дробове число	Кількість дорівнює найближчому меншому цілому числу
4	В корзині не лишилось товарів	Повідомлення «В вашому кошику немає товарів»

Таблиця 2.28 – Тести для класів еквівалентності з таблиці 2.25

№	Вхідні дані	Результат
1	Назва введена коректно	Повідомлення «Була створена категорія»
2	Назва порожня	Повідомлення «Ви не ввели назву категорії»
3	Така назва вже існує	Повідомлення «Категорія з такою назвою вже існує»

Таблиця 2.29 – Тести для класів еквівалентності з таблиці 2.26

№	Вхідні дані	Результат
1	Категорія обрана та непорожня	Інформація про товари з обраної категорії
2	Категорія не обрана	Інформація про товари всього каталогу
3	Категорія порожня	Повідомлення «В цій категорії немає товарів»

Текст програмного забезпечення вебсервісу магазину кави «Kavonra» представлений в додатку Д пояснювальної записки.

Випробування вебсервісу магазину кави «Kavonra»

Випробування ПЗ є завершальною фазою розробки та одночасно однією з ключових складових життєвого циклу ПЗ. На цьому етапі здійснюється перевірка досягнутих характеристик якості програмного продукту. Основна мета випробувань полягає в експериментальному встановленні фактичних

параметрів роботи ПЗ, а також у визначенні рівня відповідності його функціональності вимогам, визначеним у технічному завданні.

Програма та методика випробувань розміщена в Додатку В. Згідно документу, наведеному в цьому додатку, проводимо випробування розробленого ПЗ:

Функція «Редагувати вміст кошика»

Обрали пункт «Каталог» в головному меню. Отримали очікуваний результат: з'явився список товарів. Для двох товарів натиснули кнопку «В кошик». Отримали очікуваний результат: до кошика додалися обрані товари. Натиснули на посилання «Перейти до кошику», яке знаходиться в верхній частині сторінки в якості частини інтерфейсу маленького кошику. В результаті цих дій відкрилася окрема сторінка, яка містила інформацію про вміст кошика в формі таблиці. Для кожного товару була кнопка «Видалити». Також під таблицею зі вмістом кошика були кнопки «Перерахувати» та «Очистити кошик».

Далі для одного товару збільшили кількість на одиницю, а для другого поставили відмітку «Видалити», потім натиснули кнопку «Перерахувати». Отримали очікуваний результат: видалений товар зник з кошика, а кількість для іншого збільшилася на одиницю. Також змінилася сума для цього товару та загальна сума всього кошика.

Далі натиснули кнопку «Очистити кошик». Отримали очікуваний результат: всі товари видалилися з кошика та було виведено повідомлення «В вашому кошику немає товарів».

При випробуванні функції «Редагувати вміст кошика» проблем не виникло.

– Функція «Реєстрація»

Обрали пункт «Реєстрація» в головному меню. Отримали очікуваний результат: з'явилася форма з полями для вводу логіну, паролю, П.І.Б., телефону, електронної адреси, домашньої адреси, а також кнопкою «Надіслати». Ввели правильні дані в форму (логін, пароль, П.І.Б., телефон, електронна адреса, домашня адреса) та натиснули кнопку. Отримали очікуваний результат: з'явилося повідомлення «Реєстрація пройшла успішно!». При вводі логіну, який вже існує в базі, отримали повідомлення «Користувач з таким логіном уже

є!» та форму для реєстрації. При вводі не всіх даних отримали повідомлення «Ви заповнили не всі поля!» та форму для реєстрації. При вводі некоректного email отримали повідомлення «E-mail не коректний!» та форму для реєстрації.

При випробуванні функції «Реєстрація» проблем не виникло.

– Функція «Переглянути каталог»

Обрали пункт «Каталог» в головному меню. Отримали очікуваний результат: вивівся список усіх товарів каталогу, Для кожного товару вказується зображення товару, виробник, категорія, назва, вага, ціна та кнопки «В кошик» та «Докладніше». Також передбачена посторінкова навігація. Для товарів, яких немає на складі, замість кнопки «В кошик», з'явиться текст «Немає на складі».

Далі в меню категорій обрали категорію «Кава мелена». Отримали очікуваний результат: вивівся список товарів обраної категорії. Для кожного було вказано зображення товару, виробник, категорія, назва, вага, ціна та кнопки «В кошик» та «Докладніше».

Натиснули кнопку «Докладніше» для окремого товару на сторінці. Отримали очікуваний результат: відкрилася сторінка з описом обраного товару. Для товару було вказано зображення товару, назва, виробник, країна виробник, походження, категорія, сорт, помел, вага, обсмаження, упаковка, ціна, опис та кнопка «В кошик».

При випробуванні функції «Переглянути каталог» проблем не виникло.

Випробування програмного забезпечення продемонструвало відсутність помилок і відповідність всіх реалізованих функцій вимогам технічного завдання, яке наведено в додатку А.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»

Результатом кваліфікаційної роботи є розроблене програмне забезпечення, яке реалізує весь основний функціонал типового вебсервісу магазину.

Відповідно до вимог до даного програмного забезпечення, що були сформульовані в технічному завданні, яке приведене в додатку А, розроблене ПЗ виконує наступні функції для клієнта сайту та адміністратора сайту:

- Переглянути інформацію про магазин.
- Переглянути інформацію про способи оплати та доставки.
- Зворотній зв'язок.
- Переглянути каталог.
- Вибрати категорію та переглянути товари цієї категорії.
- Знайти товар по назві.
- Розширений пошук товару.
- Переглянути інформацію про товар.
- Додати товар до кошика.
- Переглянути вміст кошика.
- Редагувати вміст кошика.
- Оформити замовлення.
- Реєстрація користувача.
- Авторизація користувача.

Крім того, реалізовані розширені можливості для адміністратора сайту:

- Редагувати категорію.
- Редагувати товар.
- Редагувати список замовлень.

Програмне забезпечення повністю відповідає вимогам до організації вхідних та вихідних даних, вимогам до надійності та іншим вимогам, зазначеним в технічному завданні.

Текст програми наведено у додатку Д.

При розробці ПЗ підготовлено відповідний програмної документації:

- технічне завдання, яке представлено в додатку А,
- опис програмного забезпечення, наведений в додатку Б,
- програма та методика випробувань, розміщена в додатку В,
- інструкція користувача, яка наведена в додатку Г.

У процесі реалізації проєкту було використано мову програмування PHP, яка є однією з найпоширеніших у веб-розробці. Вона входить до «стандартного» набору для створення веб-сайтів (Linux, Apache, MySQL, PHP (Python або Perl)).

Як систему керування базами даних було обрано MySQL, яка є реляційною, вільно розповсюджуваною для використання та повністю відповідає технічним і функціональним вимогам до сучасних СКБД. Її застосування дозволяє ефективно організувати зберігання, обробку та доступ до структурованих даних у межах програмного продукту.

Створений програмний продукт було протестовано на відповідність технічному завданню відповідно до програми та методики випробовування програмного забезпечення. У ході тестування програмне забезпечення показало свою повну працездатність, а також високий рівень надійності.

Для роботи вебсервісу для магазину кави «Kavogga» рекомендується наступний склад технічних засобів з мінімальними параметрами:

- процесор 1 ГГц;
- оперативна пам'ять 512МБ;
- вільного місця на диску 5 ГБ;
- дисплей;
- миша.

Для коректної роботи програмного забезпечення системні програмні засоби повинні задовольняти наступним вимогам:

- операційна система Windows 2000 та вище;
- веб-сервер Apache;
- підтримка PHP;
- СКБД MySQL;
- вихід у мережу Internet.

4 ОХОРОНА ПРАЦІ

4.1 Основні положення охорони праці в Україні

Охорона праці в Україні — це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів і засобів, спрямована на збереження життя, здоров'я та працездатності працівника під час виконання ним трудових обов'язків [6].

Основні положення охорони праці визначаються низкою нормативно-правових актів, серед яких ключовими є [6]:

- Закон України «Про охорону праці» – базовий нормативний документ, що регулює питання охорони праці, обов'язки роботодавців і працівників, організацію безпечних умов праці.

- Кодекс законів про працю України (КЗпП) – містить розділи, присвячені правам працівників на безпечні умови праці, гарантії, відшкодування шкоди в разі нещасного випадку.

- Закон України «Про охорону здоров'я» – регулює правові засади охорони здоров'я громадян, зокрема в контексті умов праці.

- Державні санітарні норми та правила, зокрема: ДСанПіН 3.3.2.007-98 – «Гігієнічні вимоги до відеодисплейних терміналів і персональних електронно-обчислювальних машин».

- Типове положення про службу охорони праці: визначає організаційні засади функціонування служб охорони праці на підприємствах.

Ключовими засадами державної політики в галузі охорони праці є:

- Визнання життя та здоров'я працівника найвищою цінністю;
- Повна відповідальність роботодавця за забезпечення безпечних умов праці;
- Надання соціальних гарантій і підтримки постраждалим у разі нещасних випадків на виробництві;
- Комплексний підхід до вирішення питань охорони праці;

– Запровадження економічних механізмів у систему управління охороною праці.

Дотримання цих принципів, правильна організація умов праці має ключове значення для продуктивності працівника, його безпеки, психоемоційного стану та довготривалого здоров'я. Також це дозволяє уникнути ризиків пов'язаних з професійними захворюваннями, запобігти травматизму та забезпечити відповідність нормативно-правовій базі України.

4.2 Аналіз шкідливих та небезпечних факторів при роботі з комп'ютером

Робота з комп'ютером передбачає вплив ряду шкідливих та небезпечних факторів, які, за умов їх недооцінки або ігнорування, можуть спричинити серйозні наслідки для здоров'я працівників. Розглянемо найважливіші з них [7]:

– Фізіологічні фактори:

- Статичне навантаження – довготривале сидіння за робочим столом сприяє порушенню кровообігу, що з часом призводить до розвитку остеохондрозу, варикозного розширення вен, порушення постави.

- Напруження зору – тривале фокусування на екрані монітора викликає синдром комп'ютерного зору (зниження гостроти зору, сухість очей, подразнення).

- Вплив електромагнітного випромінювання – сучасні екрани мають низький рівень випромінювання, однак при тривалому контакті можуть виникати головні болі та підвищена втомлюваність.

- Шумові подразники – вентилятори ПК, принтери, системи охолодження можуть створювати фоновий шум, що сприяє перевтомі та зниженню концентрації.

– Психоемоційні фактори:

- Монотонність роботи – рутинне виконання однотипних операцій знижує мотивацію, викликає втому та емоційне вигорання.

- Високий темп роботи, дедлайни – можуть спричинити стрес, тривожність, безсоння.

– Візуально-ергономічні фактори:

- Неправильне розташування обладнання – порушення ергономіки призводить до зайвого навантаження на шию, спину, зап'ястя.
- Недостатнє освітлення або надмірне природне світло – викликає напругу очей та знижує ефективність праці.

4.3 Заходи щодо зменшення впливу шкідливих та небезпечних факторів під час роботи за комп'ютером

Щоб мінімізувати негативний вплив зазначених факторів, необхідно впроваджувати комплекс заходів технічного, організаційного та санітарно-гігієнічного характеру:

– Організація робочого місця:

- Робоче місце повинно бути оснащено ергономічним кріслом із регульованою спинкою та підлокітниками.
- Висота столу та монітора має відповідати стандартам: верхній край екрана – на рівні очей користувача, відстань до екрана – 50-70 см.
- Клавіатура повинна бути розміщена на рівні ліктів, для зап'ясть рекомендовано використовувати м'які підкладки.
- Стілець має бути з підлокітниками, спинкою та регульованою висотою.
- Під ногами бажано мати підставку для зменшення навантаження на хребет.

– Дотримання режиму праці та відпочинку:

- Після кожної години роботи за ПК рекомендується 5-10 хвилинна перерва.
 - Через кожні 2 години – перерва не менше 15 хвилин.
 - Варто використовувати вправи для очей (наприклад, переведення погляду з ближніх предметів на дальні) та легкі фізичні розминки.
- Освітлення та мікроклімат:

- Приміщення повинно мати комбіноване освітлення: природне та штучне, з коефіцієнтом освітленості не менше 300 лк.

- Температурний режим у приміщенні має становити від 20 до 24°C, відносна вологість – 40-60%, швидкість руху повітря – не більше 0,1 м/с.

- Необхідно забезпечити регулярне провітрювання кімнати або використання систем кондиціонування.

– Технічні заходи:

- Використання антивібраційних килимків для миші.

- Встановлення фільтрів синього світла або активація режиму "нічного світла" в операційній системі.

- Регулярне обслуговування комп'ютерної техніки для зниження шуму.

- Використання пристроїв захисного відключення.

- Заземлення.

- Регулярна перевірка технічного стану обладнання.

– Інформаційна та профілактична робота:

- Проведення інструктажів з техніки безпеки.

- Ознайомлення працівників з основами ергономіки та правилами гігієни праці.

- Медичні огляди працівників згідно з вимогами законодавства.

Забезпечення належних умов праці при роботі з комп'ютером є важливою складовою ефективності трудового процесу. Це включає дотримання ергономічних вимог, норм мікроклімату, електробезпеки, а також організацію режиму праці та відпочинку. Впровадження сучасних стандартів охорони праці, дотримання гігієнічних норм та рекомендацій дозволяє мінімізувати шкідливий вплив факторів, покращити самопочуття працівників та підвищити загальну результативність діяльності підприємства.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розв'язано практичну задачу програмної інженерії, що характеризується комплексністю та невизначеністю умов, а саме – розроблений вебсервіс для магазину кави «Kavotta».

Для досягнення поставленої мети було вирішено наступні задачі:

- виконано аналіз практичної задачі інженерії програмного забезпечення;
- виконано аналіз існуючих аналогів та обґрунтування необхідності розробки власного вебсервісу для магазину кави «Kavotta»;
- сформовано вимоги до розроблюваного продукту та розроблено технічне завдання на створення вебсервісу для магазину кави «Kavotta»;
- виконано аналіз вимог;
- розроблено архітектуру програмного забезпечення;
- розроблено ескізний, технічний та робочий проекти вебсервісу для магазину кави «Kavotta»;
- розроблено документацію до програмного продукту.

У результаті отримано вебсервіс для магазину кави «Kavotta», який є зручним у використанні та сприятиме розширенню бізнесу за рахунок представництва в мережі Інтернет.

.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. COFFE UA. URL:
<https://coffe.ua/?srsltid=AfmBOop9y5Cdn6CUwaoLopwEEgenMePPmInwx60ozMgVdgiY3F9aXPhp> (дата звернення – 16.05.25)
2. COFFEE STORE. URL: <https://coffee-store.com.ua/> (дата звернення – 16.05.25)
3. IDEALIST COFFEE CO. URL: <https://idealistcoffee.com/ua/coffee.html> (дата звернення – 16.05.25)
4. Neustadt Ila, Arlow Jim. UML 2 And The Unified Process: Practical Object-Oriented Analysis And Design. Editorial: Addison-Wesley Professional, 2005. – 624 p.
5. Створення та обробка баз даних. Навчальний посібник для студентів технічних спеціальностей вищих навчальних закладів / Л.С. Глоба, М.Ю. Терновой, Р.Л. Новогрудська, О.С. Штогриня. Київ: КПІ, 2013. – 477 с.
6. Про охорону праці: Закон України від 21.11.2002 р. № 229-IV. URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> (дата звернення 10.05.2025)
7. Шкідливі та небезпечні виробничі чинники при роботі з комп'ютерною технікою. URL: <https://studfile.net/preview/5211197/page:3/> (дата звернення 10.05.2025).

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»

Вступ

Повна назва розроблюваного проєкту: «Розробка вебсервісу для магазину кави «Kavorra»».

Галузь застосування: програмне забезпечення планується використовувати як представництво магазину кави, який планується відкрити найближчим часом.

1 Підстави для розробки

Підставою для розробки програмного забезпечення є завдання на кваліфікаційну роботу на здобуття ступеня вищої освіти «бакалавр».

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням ПЗ є перегляд інформації про товари, оформлення замовлення на товари, зворотний зв'язок покупців з магазином, ведення каталогу товарів та керування замовленнями.

2.2 Експлуатаційне призначення

Програмне забезпечення буде використовуватися як сайт для представництва магазину кави, який планується відкрити найближчим часом, в мережі Інтернет для розширення можливостей як клієнтів так і співробітників підприємства.

3 Вимоги до програмного забезпечення

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

– Програмне забезпечення повинно надавати наступні функціональні

можливості клієнту сайту та адміністратору сайту:

- Переглянути інформацію про магазин.
- Переглянути інформацію про способи оплати та доставки.
- Зворотній зв'язок.
- Переглянути каталог.
- Вибрати категорію та переглянути товари цієї категорії.
- Знайти товар по назві.
- Розширений пошук товару.
- Переглянути інформацію про товар.
- Додати товар до кошику.
- Переглянути вміст кошика.
- Редагувати вміст кошика.
- Оформити замовлення.
- Реєстрація користувача.
- Авторизація користувача.
- Крім того, необхідно реалізувати розширені можливості для

адміністратора сайту:

- Редагувати категорію.
- Редагувати товар.
- Редагувати список замовлень.

3.1.2 Вимоги до організації вхідних та вихідних даних

Вхідною інформацією є дані, які вводяться вручну (в залежності від типу поля введення, дані можуть бути: числові цілі, числові десяткові, текстові), або обираються із представлених списків (для уникнення можливих помилок, у випадних списках представлені готові варіанти вибору), або отримуються з бази даних MySQL.

Вихідною інформацією є результати операцій, що виведені у екранні форми, повідомлення та сформовані звіти.

Вхідними даними є наступні:

- дані для реєстрації користувача: логін, пароль, П.І.Б., телефон, e-mail,

домашня адреса;

- дані для авторизації користувача: логін, пароль.
- дані про товари каталогу: зображення товару, назва товару, виробник, країна виробник, походження, категорія, сорт, помел, вага, обсмаження, упаковка, ціна, опис;
- дані про категорію каталогу: назва категорії;
- дані про вміст кошика: ідентифікаційний номер товару у каталозі, обрана кількість для кожного товару;
- дані про замовлення: П.І.Б., домашня адреса, номер телефону, адреса електронної пошти та дані про вміст кошика.

3.2 Вимоги до надійності

3.2.1 Вимоги до забезпечення надійності функціонування програмного забезпечення

Програмне забезпечення має функціонувати неперервно на персональних обчислювальних системах. У випадку виникнення помилок у роботі комп'ютера, відновлення нормального функціонування повинно відбуватися після перезавантаження програмного продукту.

Для забезпечення надійності та цілісності інформації, слід використовувати системи керування базами даних (СКБД), які відповідають за збереження цілісності транзакцій та захист інформації.

3.2.2 Час відновлення після відмови

Час відновлення після відмови, викликаній перебоями в електроживленні, порушенням зв'язку або нефатальним збоєм операційної системи на ПК чи сервері, не повинен перевищувати 10 хвилин за умови дотримання експлуатаційних характеристик апаратного та програмного забезпечення. У випадку відмови, спричиненої технічною несправністю або критичним збоєм операційної системи, відновлення має відбуватися в межах часу, необхідного для ремонту або заміни технічних засобів та повторного встановлення програмного забезпечення.

3.2.3 Вимоги до контролю вхідної та вихідної інформації

У випадку введення користувачем неправильних даних, програмне забезпечення повинно повідомити про помилку і надати можливість її виправити.

3.3 Вимоги до умов експлуатації

3.3.1 Кліматичні умови експлуатації

Умови експлуатації програми збігаються з умовами експлуатації технічних засобів, на яких буде завантажена та використовуватись програма для подальшого використання.

3.3.2 Вимоги до чисельності та кваліфікації користувачів

Вимоги до підготовки користувачів: базові навички роботи з комп'ютером.

3.4 Вимоги до складу і параметрів технічних засобів

Для використання програми потрібна така конфігурація технічних засобів із мінімально допустимими характеристиками:

- процесор 1 ГГц і вище;
- оперативна пам'ять 512МБ і більше;
- вільного місця на диску 5 ГБ і більше;
- дисплей;
- миша.

3.5 Вимоги до інформаційної та програмної сумісності

Системні програмні засоби повинні задовольняти наступним вимогам:

- операційна система Windows 10 та вище;
- веб-сервер Apache;
- підтримка PHP;
- СКБД MySQL;

- вихід у мережу Internet.

3.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

3.7 Вимоги до транспортування та зберігання

Можливе переміщення та зберігання резервних копій програмного забезпечення на будь-яких носіях інформації, при дотриманні конкретних кліматичних умов:

- температура навколишнього середовища, °C – від +5 до +35;
- атмосферний тиск, 101,3 кПа;
- відносна вологість повітря при 25 °C – 30%.

4 Вимоги до програмної документації

Програмна документація повинна містити:

- технічне завдання;
- опис прогами;
- програму та методику випробувань;
- інструкцію користувача;
- тексти програмних модулів.

5 Техніко-економічні показники

У цій кваліфікаційній роботі не проводився розрахунок техніко-економічних показників.

6 Стадії та етапи розробки

Стадії та етапи розробки забезпечення вебсервісу магазину кави «Kavorra» представлені в таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки програмного забезпечення

Стадії розробки	Етапи розробки	Дата початку	Дата завершення
1 Технічне завдання	1.1 Обґрунтування необхідності розробки програми	10.02.2025	13.02.2025
	1.2 Розробка технічного завдання	14.02.2025	20.02.2025
	1.3 Затвердження технічного завдання	21.02.2025	25.02.2025
2 Ескізний проєкт	2.1 Розробка ескізного проєкту	03.03.2025	10.03.2025
	2.2 Затвердження ескізного проєкту	11.03.2025	12.03.2025
3 Технічний проєкт	3.1 Розробка технічного проєкту	13.03.2025	20.03.2025
	3.2 Затвердження технічного проєкту	21.03.2025	24.03.2025
4 Робочий проєкт	4.1 Розробка програми	26.03.2025	15.04.2025
	4.2 Розробка програмної документації	16.04.2025	21.04.2025
	4.3 Випробування програми	22.04.2025	06.05.2025

7 Порядок контролю та приймання

Для здійснення контролю та приймання необхідно надати опис програми разом із програмою та методикою випробувань. У разі, якщо під час випробувань буде виявлено недоліки, виконавець зобов'язаний усунути помилки та недоліки у термін, що не перевищує двох тижнів з моменту проведення випробування.

ДОДАТОК Б – ОПИС ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»

1 Загальні відомості

Найменування: Вебсервіс для магазину кави «Kavorra».

Позначення: «Kavorra».

Для реалізації проєкту використовувалась мова програмування PHP, яка входить в «стандартний» набір для створення веб-сайтів (Linux, Apache, MySQL, PHP (Python або Perl)).

У якості системи управління базами даних було обрано MySQL. Ця СКБД є безкоштовною та відповідає всім нормам та вимогам сучасних реляційних СКБД.

Для коректної роботи програмного забезпечення системні програмні засоби повинні задовольняти наступним вимогам:

- операційна система Windows 10 та вище;
- веб-сервер Apache;
- підтримка PHP;
- СКБД MySQL;
- вихід у мережу Internet.

2 Функціональне призначення розробки

Функціональним призначенням ПЗ є перегляд інформації про товари, оформлення замовлення на товари, зворотний зв'язок покупців з магазином, ведення каталогу товарів та керування замовленнями.

Програмне забезпечення надає наступні функціональні можливості користувачу сайту та адміністратору сайту:

- Переглянути інформацію про магазин.
- Переглянути інформацію про способи оплати та доставки.
- Зворотній зв'язок.
- Переглянути каталог.
- Вибрати категорію та переглянути товари цієї категорії.

- Знайти товар по назві.
- Розширений пошук товару.
- Переглянути інформацію про товар.
- Додати товар до кошику.
- Переглянути вміст кошика.
- Редагувати вміст кошика.
- Оформити замовлення.
- Реєстрація користувача.
- Авторизація користувача.

Крім того, реалізовані розширені можливості для адміністратора сайту:

- Редагувати категорію.
- Редагувати товар.
- Редагувати список замовлень.

3 Опис логічної структури

Програмне забезпечення розроблено за шаблоном проєктування MVC (Model-View-Controller), тому воно розділено на 3 основні складові:

- Model (Модель) відповідає за бізнес-логіку та роботу з даними (наприклад, управління товарами, замовленнями, користувачами). Вона незалежна від інтерфейсу користувача.

- View (Представлення) відповідає за відображення даних користувачу. Воно не містить бізнес-логіки.

- Controller (Контролер) виступає посередником між Моделлю та Представленням. Він отримує запити від користувача, обробляє їх, взаємодіє з Моделлю для отримання або зміни даних, а потім вибирає відповідне Представлення для відображення результату.

В програмному забезпеченні є багато різних сутностей (товари, категорії товарів, кошик, замовлення). MVC дозволяє чітко розмежувати логіку управління цими сутностями від їх відображення, що спрощує розробку та підтримку.

Модель даних програмного забезпечення.

Об'єкти бази даних представлені у вигляді таблиць. Кожна таблиця складається з рядків та стовбців. Стовбці є атрибутами, а рядки – записами (або кортежами). Ключем є атрибут, значення якого однозначно визначає запис в таблиці. Кожна сутність таблиці має ключове поле, що представляє собою первинний ключ і складається з одного, або декількох атрибутів. Програмні дані зберігаються у нормалізованих таблицях бази даних.

4 Технічні засоби

Для роботи з програмою необхідний наступний склад технічних засобів з мінімальними параметрами:

- процесор 1 ГГц і вище;
- оперативна пам'ять 512МБ і більше;
- дискового простору 5 ГБ і більше;
- дисплей;
- миша.

5 Виклик та завантаження

Для запуску програми необхідно буде перейти на сайт вебсервісу магазину «Kavorra». Програмне забезпечення повинно розташовуватися на інтернет-сервері, база даних вебсервісу має зберігатися окремо на сервері баз даних.

6 Вхідні дані

- дані для реєстрації користувача: логін, пароль, П.І.Б., телефон, e-mail, домашня адреса;
- дані для авторизації користувача: логін, пароль.
- дані про товари каталогу: зображення товару, назва товару, виробник, країна виробник, походження, категорія, сорт, помел, вага, обсмаження, упаковка, ціна, опис;
- дані про категорію каталогу: назва категорії;
- дані про вміст кошика: ідентифікаційний номер товару у каталозі,

обрана кількість для кожного товару;

– дані про замовлення: П.І.Б., домашня адреса, номер телефону, адреса електронної пошти та дані про вміст кошика.

7 Вихідні дані

Вихідною інформацією є результати операцій, що виведені у екранні форми, повідомлення та сформовані звіти.

ДОДАТОК В – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»

1 Об'єкт випробовування

Об'єктом для випробувань є вебсервіс для магазину кави «Kavorra».

Сфера застосування: для працівників магазину кави «Kavorra» з метою поширення інформації про магазин, здійснення віддаленої демонстрації товару для його просування, збільшення обсягу продажів, пошуку нових клієнтів та для клієнтів з метою отримання інформації про товари і магазин та купівлі товарів з каталогу.

Позначення випробуваної програми: «Kavorra».

2 Мета випробовування

Метою випробувань є перевірка того, наскільки програмне забезпечення відповідає вимогам, зазначеним у технічному завданні, наведеному в додатку А.

3 Вимоги до програми

При проведенні випробувань функціональні характеристики програми підлягають перевірці на відповідність вимогам, викладеним у пункті «Вимоги до функціональних характеристик» технічного завдання.

3 Вимоги до програми

У процесі випробувань здійснюється перевірка функціональних можливостей програми на відповідність вимогам до функціональних характеристик технічного завдання.

4 Вимоги до програмної документації

Документація до програмного засобу повинна містити наступні документи:

- технічне завдання;
- опис програми;
- програму та методику випробувань;
- інструкцію користувача;
- текст програми.

5 Засоби та порядок випробувань

Для проведення випробувань необхідні наступні системні програмні засоби:

- операційна система Windows 10 та вище;
- веб-сервер Apache;
- підтримка PHP;
- СКБД MySQL;
- вихід у мережу Internet.

Для проведення випробувань необхідний наступний склад технічних засобів з мінімальними параметрами:

- процесор 1 ГГц і вище;
- оперативна пам'ять 512МБ і більше;
- дискового простору 5 ГБ і більше;
- дисплей;
- миша.

Порядок проведення випробувань

Випробування програмного забезпечення проводиться в такій послідовності:

- Спочатку виконуються всі інструкції для кожної функції.
- Потім аналізуються отримані результати, щоб перевірити, чи відповідає програмний продукт визначеним вимогам та системній документації.
- Якщо виявляються помилки, складається їх перелік і обговорюються терміни виправлення з розробником. Після цього замовник

повторно тестує систему в повному обсязі, за необхідності використовуючи нові тести.

6 Методи випробувань

Для проведення випробувань розробленого програмного забезпечення було обрано метод "чорної скриньки", що передбачає перевірку функціональності програми без необхідності аналізу її внутрішнього коду.

Всі функції програми будуть проходити випробовування. Через великий обсяг в роботі наводиться випробування тільки декількох функцій.

Складемо програму випробувань:

– Функція «Редагувати вміст кошика»

Обрати пункт «Каталог» в головному меню. Повинен з'явитися список товарів. Для кількох товарів натиснути кнопку «В кошик». В кошику мають з'явитися обрані товари. Натиснути на посилання «Перейти до кошику», яке знаходиться в верхній частині сторінки в якості частини інтерфейсу маленького кошику. Повинна відкритися окрема сторінка з інформацією про вміст кошика в формі таблиці. Для кожного товару має бути кнопка «Видалити». Також під таблицею зі вмістом кошика мають бути кнопки «Перерахувати» та «Очистити кошик».

Далі для одного товару збільшити/зменшити кількість, а для іншого поставити відмітку «Видалити», потім натиснути кнопку «Перерахувати». В результаті видалений товар має зникнути з кошика, а кількість для іншого змінитися. Також змінитися повинна сума для цього товару та загальна сума всього кошика.

Далі натиснути кнопку «Очистити кошик». Всі товари повинні видалилися з кошика та має з'явитися повідомлення «В вашому кошику немає товарів».

– Функція «Реєстрація»

Обрати пункт «Реєстрація» в головному меню. Повинна з'явитися форма з полями для вводу логіну, паролю, П.І.Б., телефону, електронної адреси, домашньої адреси, а також кнопкою «Надіслати дані». Якщо ввести коректні дані в форму (логін, пароль, П.І.Б., телефон, електронна адреса, домашня адреса)

та натиснути кнопку, то повинно з'явитися повідомлення «Реєстрація пройшла успішно». При вводі логіну, який вже існує в базі, повинно з'явитися повідомлення «Користувач з таким логіном уже є» та форма для реєстрації. При вводі не всіх даних маємо отримати повідомлення «Ви заповнили не всі поля». При вводі некоректного email повинно з'явитися повідомлення «E-mail не коректний».

– Функція «Авторизація»

Обрати пункт «Авторизація» в головному меню. Повинна з'явитися форма з полями для вводу логіну та паролю, а також кнопкою «Надіслати дані». Якщо ввести некоректні дані в форму та натиснути кнопку, то повинно з'явитися повідомлення «Дані введені не вірно».

Якщо ввести коректні дані в форму та натиснути кнопку, то повинно з'явитися повідомлення «Авторизація пройшла успішно» та в головному меню пункт «Авторизація» має змінитися на «введений логін [вийти]».

– Функція «Переглянути каталог»

Обрати пункт «Каталог» в головному меню. Повинен з'явитися список всіх товарів каталогу з наступною інформацією для кожного: зображення товару, виробник, категорія, назва, вага, ціна та кнопки «В кошик» та «Докладніше». Для товарів, яких немає на складі, замість кнопки «В кошик» має з'явитися текст «Немає на складі».

Якщо в меню категорій обрати певну категорію, то має з'явитися список товарів обраної категорії з наступною інформацією для кожного: зображення товару, виробник, категорія, назва, вага, ціна та кнопки «В кошик» та «Докладніше». Для товарів, яких немає на складі, замість кнопки «В кошик» має з'явитися текст «Немає на складі».

Якщо натиснути на посилання «Докладніше» для окремого товару на сторінці, то має відкритися сторінка з описом обраного товару, в якому вказано зображення товару, назва, виробник, країна виробник, походження, категорія, сорт, помел, вага, обсмаження, упаковка, ціна, опис та кнопка «В кошик».

ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВИ «KAVORRA»

1 Призначення програми

Функціональним призначенням ПЗ є перегляд інформації про товари, оформлення замовлення на товари, зворотний зв'язок покупців з магазином, ведення каталогу товарів та керування замовленнями.

Експлуатаційне призначення програмного забезпечення: Програмне забезпечення буде використовуватися як сайт для представництва магазину кави, який планується відкрити найближчим часом, в мережі Інтернет для розширення можливостей як клієнтів так і співробітників підприємства.

Програмне забезпечення надає наступні функціональні можливості користувачу сайту та адміністратору сайту:

- Переглянути інформацію про магазин.
- Переглянути інформацію про способи оплати та доставки.
- Зворотній зв'язок.
- Переглянути каталог.
- Вибрати категорію та переглянути товари цієї категорії.
- Знайти товар по назві.
- Розширений пошук товару.
- Переглянути інформацію про товар.
- Додати товар до кошика.
- Переглянути вміст кошика.
- Редагувати вміст кошика.
- Оформити замовлення.
- Реєстрація користувача.
- Авторизація користувача.

Крім того, програмне забезпечення надає розширені можливості для адміністратора сайту:

- Редагувати категорію.

- Редагувати товар.
- Редагувати список замовлень.

2 Умови виконання програми

Вимоги до складу програмних засобів

Для проведення випробувань необхідні наступні системні програмні засоби:

- операційна система Windows 10 та вище;
- веб-сервер Apache;
- підтримка PHP;
- СКБД MySQL;
- вихід у мережу Internet.

Вимоги до складу технічних засобів

Для проведення випробувань необхідний наступний склад технічних засобів з мінімальними параметрами:

Для роботи з програмою

- процесор 1 ГГц і вище;
- оперативна пам'ять 512МБ і більше;
- дискового простору 5 ГБ і більше;
- дисплей;
- миша.

Для вмілого користування програмним забезпеченням користувачі повинні володіти базовими навичками роботи з комп'ютером.

3 Виконання програми

Для запуску програми необхідно буде перейти на сайт вебсервісу магазину «Kavorra». Програмне забезпечення повинно розташовуватися на інтернет-сервері, база даних магазину зберігається окремо на сервері баз даних.

Для забезпечення зручної навігації на кожній сторінці магазину міститься:

- логотип та назва магазину;

– головне меню сайту, яке складається з пунктів: «Каталог», «Оплата і доставка», «Про нас», «Зворотній зв'язок», «Реєстрація», «Авторизація» (для авторизованого користувача замість пункту «Авторизація» з'являється ім'я користувача та рядом посилання «Вийти»);

– список категорій кави у вигляді окремого меню;

– вміст маленького кошику (загальна кількість і сума товарів у кошику, посилання на сторінку роботи з кошиком «Перейти до кошику»);

– поле з кнопкою для швидкого пошуку товару та посилання на сторінку розширеного пошуку.

При швидкому пошуку в назві товару шукається довільне зі слів з введеного користувачем запиту. Відповідь виводиться на сторінці розширеного пошуку, перехід до якої здійснюється автоматично.

На рисунку Г.1 представлено сторінку перегляду каталогу, яка є головною сторінкою сайту. Ця сторінка реалізує функції варіанту використання «Переглянути каталог». В центральній частині сторінки відображаються всі товари каталогу або лише товари окремої категорії, обраної в меню категорій.

Для кожного товару вказується зображення товару, виробник, категорія, назва, вага, ціна, кнопка «В кошик» та посилання «Докладніше», яке допомагає перейти на сторінку з інформацією про цей товар. Також передбачена зручна посторінкова навігація.

На рисунку Г.2 представлено сторінку перегляду детальної інформації про товар, яка містить інформацію про окремий товар. Ця сторінка реалізує функції варіанту використання «Переглянути інформацію про товар».

Для товару вказується зображення товару, назва, виробник, країна виробник, походження, категорія, сорт, помел, вага, обсмаження, упаковка, ціна, опис та кнопка «В кошик».



Рисунок Г.1 – Сторінка перегляду каталогу вебсервісу магазину кави «Kavorra»

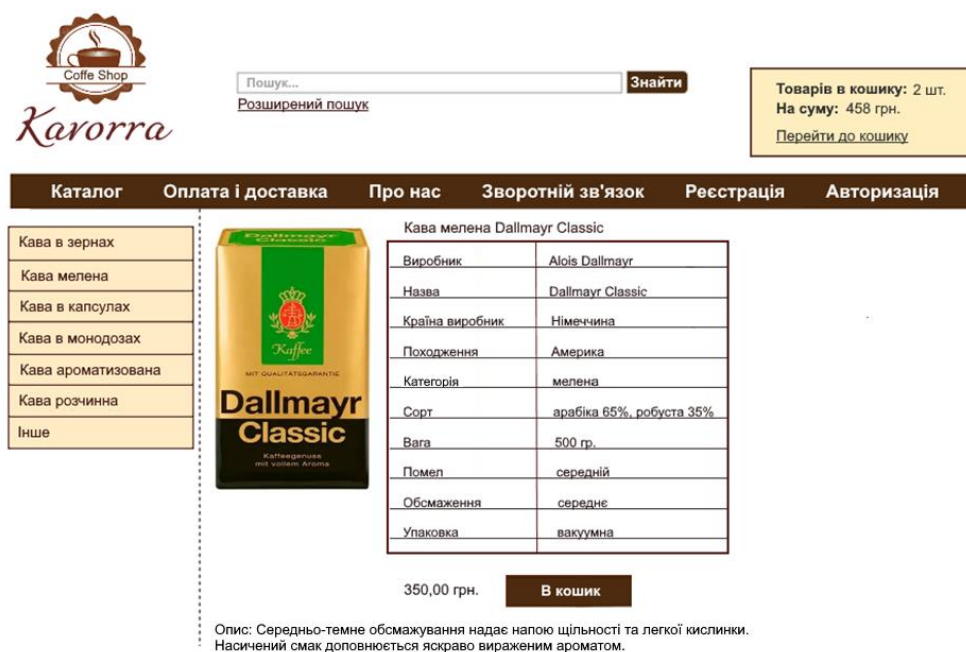


Рисунок Г.2 – Сторінка перегляду детальної інформації про товар вебсервісу магазину кави «Kavorra»

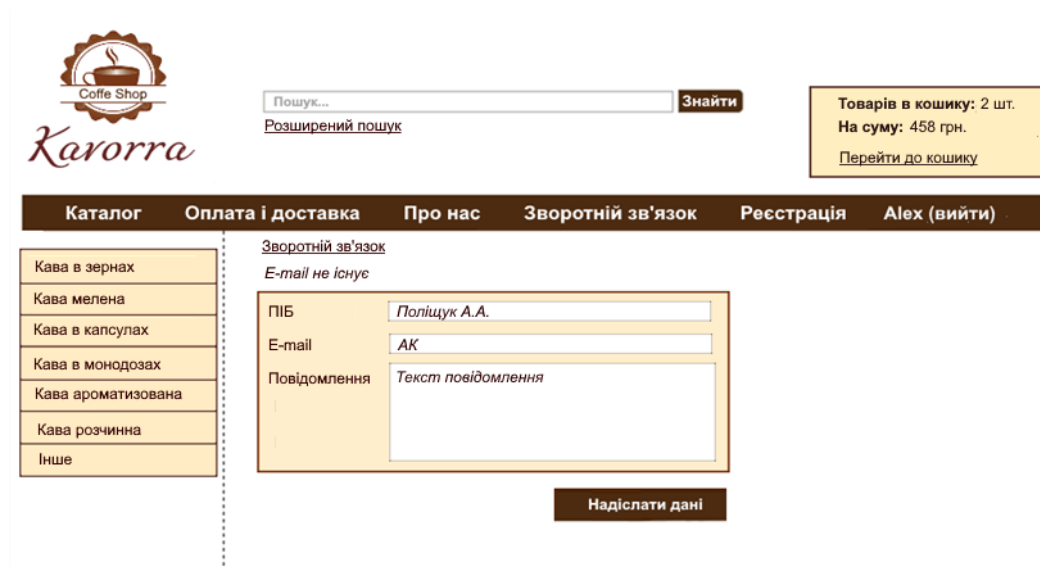
На рисунку Г.3 представлено сторінку оформлення замовлення. Ця сторінка реалізує функції варіанту використання «Оформити замовлення». Вона містить форму для вводу особистої інформації користувача (П.І.Б., телефон, електронна адреса, домашня адреса) та кнопку «Надіслати дані». Якщо користувач попередньо авторизувався, в полях вводу відображаються його дані, взяті з бази даних магазину. Користувачу лишається лише підтвердити їх, натиснувши кнопку «Надіслати дані».

The screenshot shows the 'Kavorra' website interface. At the top left is the 'Kavorra' logo with a coffee cup icon. To its right is a search bar with the text 'Пошук...' and a 'Знайти' button. Further right is a yellow box containing the text: 'Товарів в кошику: 2 шт.', 'На суму: 458 грн.', and a link 'Перейти до кошику'. Below the header is a dark navigation bar with white text links: 'Каталог', 'Оплата і доставка', 'Про нас', 'Зворотній зв'язок', 'Реєстрація', and 'Alex (вийти)'. The main content area has a title 'Оформлення замовлення'. On the left is a sidebar menu with a list of coffee products: 'Кава в зернах', 'Кава мелена', 'Кава в капсулах', 'Кава в монодозах', 'Кава ароматизована', 'Кава розчинна', and 'Інше'. The central part contains a form with fields for 'ПІБ', 'Телефон', 'E-mail', and 'Адреса'. Below the form is a 'Надіслати дані' button.

Рисунок Г.3 – Сторінка оформлення замовлення вебсервісу магазину кави «Kavorra»

На рисунку Г.4 представлено сторінку зворотнього зв'язку, за допомогою якої користувач має змогу написати листа адміністрації сайту.

Ця сторінка реалізує функції варіанту використання «Зворотній зв'язок». Вона містить форму з полями вводу особистої інформації користувача (П.І.Б., електронна адреса), полем для вводу повідомлення та кнопкою «Надіслати дані». При вводі некоректних даних буде з'являтися повідомлення про помилку.



Kavorra Coffe Shop

Пошук... **Знайти**

[Розширений пошук](#)

Товарів в кошику: 2 шт.
На суму: 458 грн.
[Перейти до кошику](#)

Каталог Оплата і доставка Про нас Зворотній зв'язок Реєстрація Alex (вийти)

Зворотній зв'язок
E-mail не існує

ПІБ

E-mail

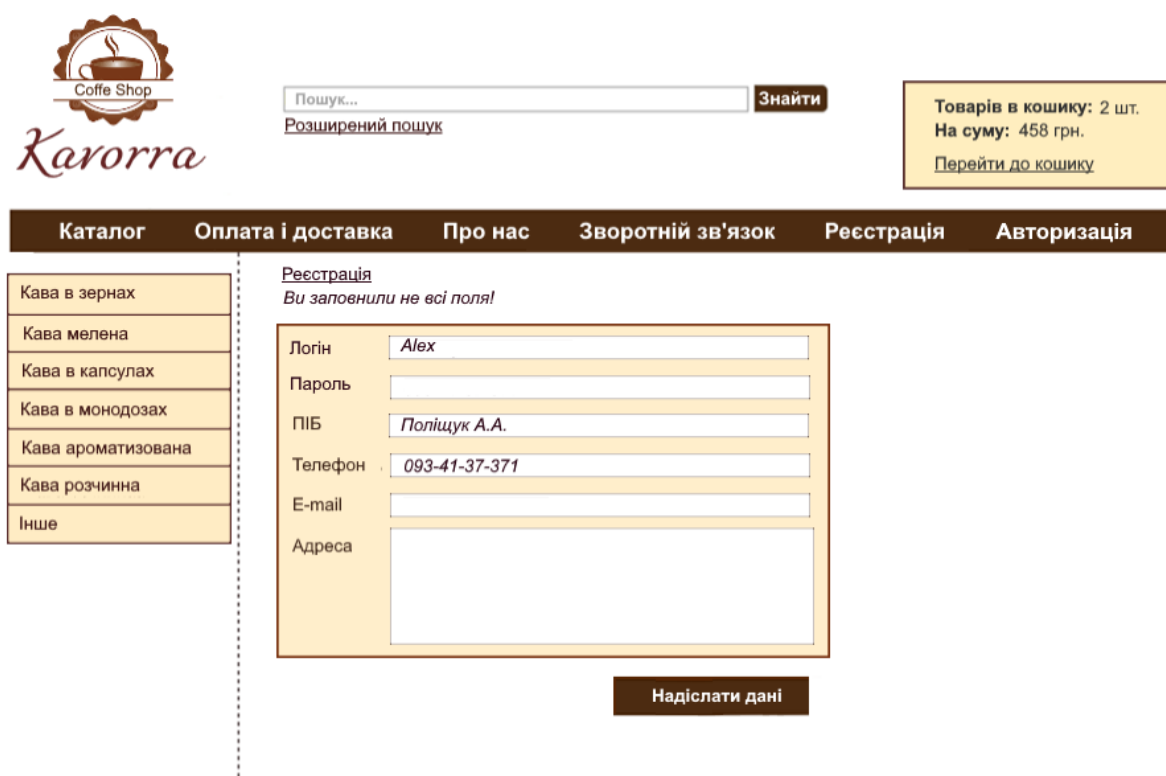
Повідомлення

Надіслати дані

Кава в зернах
Кава мелена
Кава в капсулах
Кава в монодозах
Кава ароматизована
Кава розчинна
Інше

Рисунок Г.4 – Сторінка зворотнього зв'язку вебсервісу магазину кави «Kavorra»

На рисунку Г.5 представлено сторінку реєстрації. Ця сторінка реалізує функцію варіанту використання «Реєстрація». Вона містить форму для вводу особистої інформації користувача (логін, пароль, П.І.Б., телефон, електронна адреса, домашня адреса) та кнопку «Надіслати дані».



Kavorra Coffe Shop

Пошук... **Знайти**

[Розширений пошук](#)

Товарів в кошику: 2 шт.
На суму: 458 грн.
[Перейти до кошику](#)

Каталог Оплата і доставка Про нас Зворотній зв'язок Реєстрація Авторизація

Реєстрація
Ви заповнили не всі поля!

Логін

Пароль

ПІБ

Телефон

E-mail

Адреса

Надіслати дані

Кава в зернах
Кава мелена
Кава в капсулах
Кава в монодозах
Кава ароматизована
Кава розчинна
Інше

Рисунок Г.5 – Сторінка реєстрації вебсервісу магазину кави «Kavorra»

Для користувача, який авторизувався на правах адміністратора на всіх сторінках додатково відображається панель адміністрування. Вона має наступні пункти: «Перегляд магазину», «Замовлення», «Товари», «Категорії», «Вийти».

Якщо обраний пункт «Перегляд магазину», то адміністратор може переглядати всі сторінки та користуватися всіма функціями, доступними для звичайного користувача. Єдина різниця у наявності панелі адміністрування у верхній частині екрану.

На рисунку Г.6 представлено панель адміністрування на прикладі сторінки перегляду каталогу.

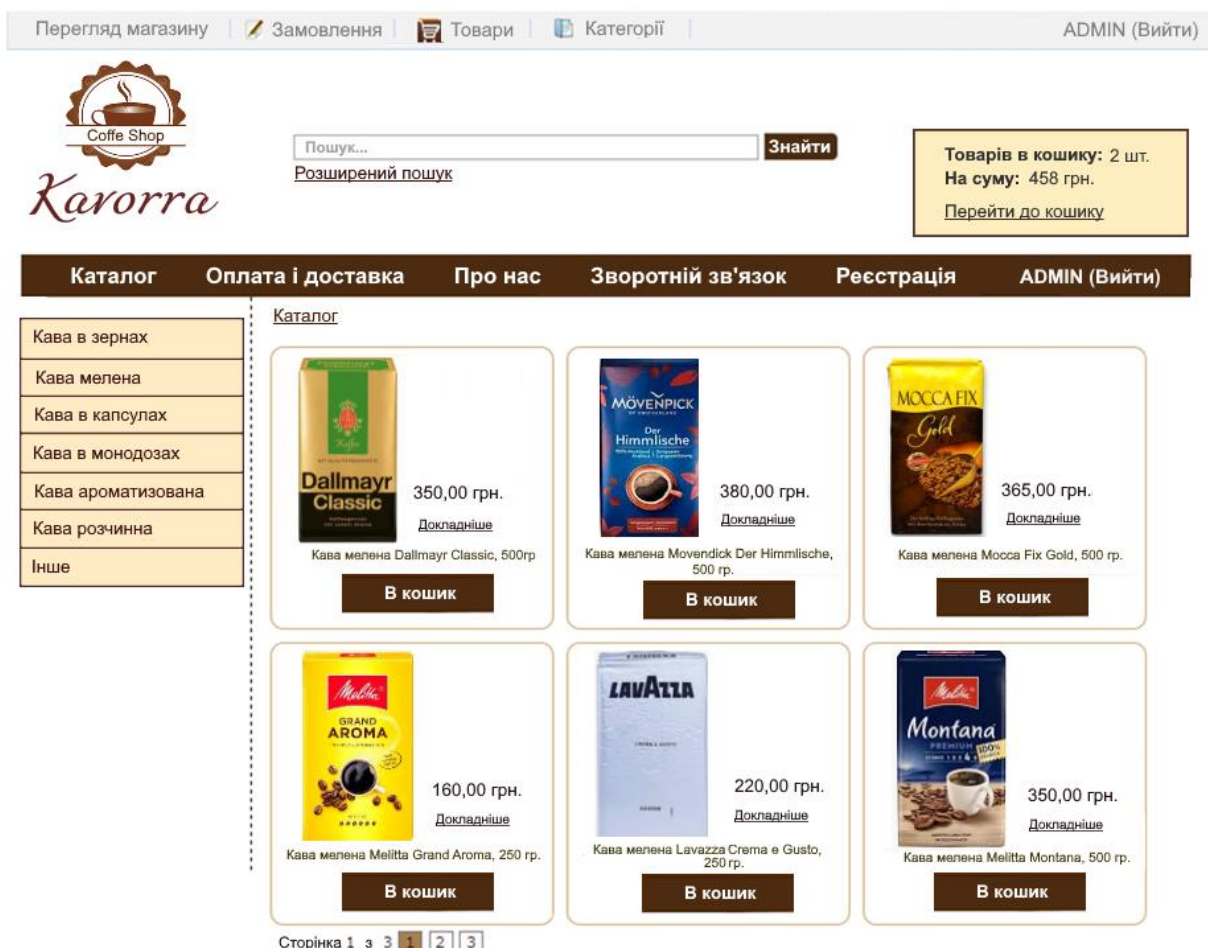


Рисунок Г.6 – Сторінка перегляду каталогу з панеллю адміністрування у верхній частині

Якщо обраний пункт «Категорії», то адміністратор має змогу додавати і видаляти окремі категорії каталогу, а також змінювати їх назву.

Аналогічно реалізована функція редагування товарів каталогу, за допомогою якої адміністратор має змогу додавати і вилучати окремі товари до каталогу, а також змінювати їх назву та інші параметри.

Також адміністратор має змогу працювати зі списком замовлень. За допомогою відповідної сторінки адміністратор може переглядати список замовлень та вилучати вже опрацьовані.

ДОДАТОК Д – ТЕКСТ ПРОГРАМИ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ КАВІ «KAVORRA»

```

    authorization.php
<?
class Application_Models_Authorization extends DataBase
{
    function authorize($login,$pass)
    {
        $sql = parent::query("SELECT * FROM `user` WHERE login='%s' and
pass='%s'", $login,$pass);
        if( mysql_num_rows($sql)!=0){ //якщо є співпадіння
            $row=mysql_fetch_assoc($sql);
            $_SESSION["Auth"]=true;
            $_SESSION["User"]=$login;
            $_SESSION["role"]=$row["role"];
            return true;
        }
        else{
            $_SESSION["Auth"]=false;
            return false;
        }
    }
}

class Application_Controllers_Authorization extends BaseController
{
    function __construct(){
        $this->dislpay_form = true; // показувати форму введення даних
        if(isset($_REQUEST["send"])){ // якщо прийшли дані з форми
            $authorization = new Application_Models_Authorization;

            $login = $_POST['login'];
            $pass = $_POST['pass'];

            if($authorization->authorize($login,$pass)){ //якщо повернуло true
                header('Location: /authorization?thanks=1');
                exit;
            }
        }
    }
}

```

```

    }
    else
        $this->message="Дані введені не вірно!";
    }
    if($_GET['thanks']){
        $this->message="Авторизація пройшла успішно!";
        $this->dislpay_form = false;
    }
    if($_GET['out']=="1"){
        $_SESSION["Auth"]=false;
        $_SESSION["User"]="";
        $_SESSION["role"]="";
        $this->dislpay_form = true;
    }
}
}
}

```

registration.php

```

<?
class Application_Models_Registration extends DataBase
{
    function isValidData($array_data){
        foreach($array_data as $name => $value){
            if(!trim($value))
                $result['error'] = "Ви заповнили не всі поля!";
            else
                $result["$name"] = $value;
        }
        if(!$result['error']){ //Якщо всі поля заповнені перевіряємо email
            if(!preg_match("/^[A-Za-z0-9._-]+@[A-Za-z0-9._-]+([A-Za-z0-9_-][A-Za-z0-9_]+)$/",$array_data['email'])){
                $result['error'] = "E-mail не коректний!";
            }
            if($res = parent::query("SELECT login FROM user WHERE
login='%s'", $array_data['login'])){
                if(mysql_num_rows($res) != 0)
                    $result['error'] = "Користувач з таким логіном уже є!";
            }
        }
        return $result;
    }
}

```

```

function registrate($array){
    if(parent::build_query("INSERT INTO `user` SET ",$array))
        return true;
    else
        return false;
}
}

class Application_Controllers_Registration extends BaseController
{
    function __construct(){
        $this->dislpay_form = true; // показувати форму введення даних
        if(isset($_REQUEST["send"])){ // якщо прийшли дані з форми
            $registration = new Application_Models_Registration;

            $array_data['login'] = $_POST['login'];
            $array_data['pass'] = $_POST['pass'];
            $array_data['name'] = $_POST['name'];
            $array_data['phone'] = $_POST['phone'];
            $array_data['email'] = $_POST['email'];
            $array_data['adress'] = $_POST['adress'];

            $result=$registration->isValidData($array_data);
            if($result['error'])
                $this->error = $result['error'];
            else{
                $this->result = $result;

                if($registration->registrate($result)){
                    header('Location: /registration?thanks=1');
                    exit;
                }
                else
                    $this->error = "Не вдалося зареєструватися!";
            }
        }

        if($_GET['thanks']){
            $this->message="Реєстрація пройшла успішно!";
            $this->dislpay_form = false;
        }
    }
}

```



```

    }
}
}

```

application/ controllers/ search.php

```

<?
class Application_Controllers_Search extends BaseController
{
    function construct(){
        if(isset($_REQUEST['to_search'])){ // якщо натиснута кнопка пошуку
            $model = new Application_Models_Search;
            $data = $_REQUEST;
            $searchgoods = $model->search($data);
            if($searchgoods)
                $this->searchgoods = $searchgoods; //якщо щось знайшли
робимо доступним для view
            else
                $this->message = "<h1>По вашому запиту нічого не
знайдено</h1>!";
        }
        $allcategories=Category::getInstance()->categories;
        $this->allcategories = $allcategories;
    }
}

```

application/ controllers/ about.php

```

<?
//контролер сторінки "Про нас"
class Application_Controllers_About extends BaseController
{
    function __construct() {}
}

```

application/ controllers/ cart.php

```

<?
class Application_Controllers_Cart extends BaseController
{
    function __construct(){
        $model=new Application_Models_Cart;
        if($_REQUEST['refresh']){ // якщо користувач змінив дані в корзині
            $list_Item_Id=$_REQUEST;

```

```

/* знаходимо в масиві помітки на зміну кількості та видалення */
foreach($list_Item_Id as $Item_Id => $new_count){
    $id="";
    if(substr($Item_Id, 0, 5)=="item_") {
        $id=substr($Item_Id, 5);
        $count=$new_count;
    }
    elseif(substr($Item_Id, 0, 4)=="del_"){
        $id=substr($Item_Id, 4);
        $count=0;
    }
    if($id){
        $array_product_id[$id]=(int)$count;
    }
}
$model->refreshCart($array_product_id); // передаємо в модель дані
для оновлення корзини
    SmalCart::getInstance()->setCartData();// перераховуємо маленьку
корзину
    header('Location: /cart');
    exit;
}
if($_REQUEST['clear']){ // якщо користувач хоче очистити корзину
    $model->clearCart(); // передаємо в модель дані для оновлення
корзини
    SmalCart::getInstance()->setCartData();// перераховуємо маленьку
корзину
    header('Location: /cart');
    exit;
}
$big_cart=$model->printCart(); //виводимо список позицій до
замовлення()
    $this->big_cart=$big_cart; //в представленні він буде доступний через
змінну $big_cart
    $this->empty_cart=$model->isEmptyCart();
}
}

```

application/ controllers/letteradmin.php

<?

class Application_Controllers_LetterAdmin extends BaseController

```

{
    function __construct(){
        $this->dislpay_form = true; // показувати форму введення даних
        if(isset($_REQUEST["send"])){ // якщо прийшли дані з форми
            $letteradmin= new Application_Models_ LetterAdmin;
            //створюємо модель відправки повідомлення
            $error=$letteradmin->isValidData($_REQUEST); //перевіряємо на
коректність введення
            if($error)$this->error=$error;
            else{
                $ letteradmin ->sendMail();
                header('Location: / letteradmin?thanks=1');
                exit;
            }
        }

        if(isset($_REQUEST["thanks"])){ //формуємо повідомлення
            $this->message="Ваше повідомлення надіслане!";
            $this->dislpay_form = false;
        }
    }
}

```

```

application/ controllers/ order.php
<?
class Application_Controllers_Order extends BaseController
{
    function __construct()
    {
        $model = new Application_Models_Order;    //створюємо модель
замовлення

        if($_SESSION["Auth"]=true){
            $login=$_SESSION["User"];
            $this->user_data=$model->readData($login);
            $this->message="Підтвердіть правильність ваших даних!";
        }

        $this->dislpay_form = true; // показувати форму введення даних

        if($_REQUEST["send"]){ // якщо прийшли дані з форми

```

```

        $error=$model->isValidData($_REQUEST); // перевіряємо на
коректність введення
        if($error)$this->error=$error;
        else{
            //якщо помилок немає, то додаємо замовлення до БД
            $order_id=$model->addOrder();
            $smalCart::getInstance()->setCartData();// перераховуємо
            маленьку корзину
            header('Location: /order?thanks='.$order_id);
            exit;
        }
    }
}

```

```

        if(isset($_REQUEST["thanks"])){ //формуємо повідомлення
            $this->message="Ваше замовлення <strong>№
".$_REQUEST["thanks"]."</strong> прийняте!";
            $this->display_form = false;
        }
    }
}

```

application/ controllers/ product.php

```
<?
```

//контролер обробляє дані каталогу

```
class Application_Controllers_Product extends BaseController
```

```
{
```

```
    function __construct(){
```

```
        $model=new Application_Models_Product;
```

```
        $product = $model->getProduct($_REQUEST['id']);
```

```
        $this->product=$product;
```

```
    }
```

```
}
```

application/ controllers/ registration.php

```
<?
```

```
class Application_Controllers_Registration extends BaseController
```

```
{
```

```
    function __construct(){
```

```
        $this->display_form = true; // показувати форму введення даних
```

```
        if(isset($_REQUEST["send"])){ // якщо прийшли дані з форми
```

```
            $registration = new Application_Models_Registration;
```

```

$array_data['login'] = $_POST['login'];
$array_data['pass'] = $_POST['pass'];
$array_data['name'] = $_POST['name'];
$array_data['phone'] = $_POST['phone'];
$array_data['email'] = $_POST['email'];
$array_data['adress'] = $_POST['adress'];

$result=$registration->isValidData($array_data);
if($result['error'])
    $this->error = $result['error'];
else{
    $this->result = $result;

    if($registration->registrate($result)){
        header('Location: /registration?thanks=1');
        exit;
    }
    else
        $this->error = "Не вдалося зареєструватися!";
    }
}

if($_GET['thanks']){
    $this->message="Реєстрація пройшла успішно!";
    $this->display_form = false;
}
}
}

```

application/ controllers/ search.php

<?

class Application_Controllers_Search extends BaseController

{

function __construct(){

if(isset(\$_REQUEST['to_search'])){ // якщо натиснута кнопка пошуку

\$model = new Application_Models_Search; //створюємо модель

\$data = \$_REQUEST;

\$searchgoods = \$model->search(\$data);

if(\$searchgoods)

```

        $this->searchgoods = $searchgoods; //якщо щось знайшли
робимо доступним для view
    else
        $this->message = "<h1>По вашому запиту нічого не
знайдено</h1>!";
    }
    $allcategories=Category::getInstance()->categories;
    $this->allcategories = $allcategories;
}
}

```

category.php

```

<?
//Реалізований патерн Singleton
class Category extends DataBase
{
    public $categories;

    protected static $instance; //

    //отримуємо таблицю категорій з БД, записуємо в масив, для подальшої
роботи
    public function __construct() {
        $result = parent::query("SELECT * FROM `category`");
        if(mysql_num_rows($result)){
            while ($row = mysql_fetch_assoc($result)){
                $this->categories[]=$row;
            }
        }
    }

    public static function getInstance(){
        if (!is_object(self::$instance)) self::$instance = new self;
        return self::$instance;
    }

    public function addCategory($title){
        $array['title'] = $title;
        $array['url']=$this->translitIt($title);
        if(parent::build_query("INSERT INTO category SET
", $array)){
            $message = "Була створена категорія '$title'!";

```

```

    }
    else{
        $message = "Не вдалося створити категорію '$title!'";
    }
    return $message;
}

public function editCategory($id,$old_title,$new_title){
    $array['title'] = $new_title;
    $array['url']=$this->translitIt($new_title);
    if(parent::query("UPDATE category SET
".parent::build_part_query($array)." WHERE id = %d", $id)){
        $message = "Категорія '$old_title' була перейменована в
'$new_title!'";
    }
    else{
        $message = "Не вдалося перейменувати категорію '$old_title!'";
    }
    return $message;
}

public function delCategory($id, $title){
    if(parent::query("DELETE FROM category WHERE id = %d",$id)){
        $message = "Категорія '$title' була вилучена";
    }
    else{
        $message = "Не вдалося вилучити категорію '$title!'";
    }
    return $message;
}

//
public function getCategoryMenu(){
    foreach($this->categories as $category){
        $print.='<li><a href="/'.$category['url'].'">'.$category['title'].'</a></li>';
    }
    return $print;
}

public function translitIt($str){
    $str = array(
        "А"=>"a", "Б"=>"b", "В"=>"v", "Г"=>"g",
        "Д"=>"d", "Е"=>"e", "Ж"=>"j", "З"=>"z", "И"=>"i",

```

```

"Й"=>"y","К"=>"k","Л"=>"l","М"=>"m","Н"=>"n",
"О"=>"o","П"=>"p","Р"=>"r","С"=>"s","Т"=>"t",
"У"=>"u","Ф"=>"f","Х"=>"h","Ц"=>"ts","Ч"=>"ch",
"Ш"=>"sh","Щ"=>"sch","Ъ"=>"","Ы"=>"yi","Ь"=>"" ,
"Э"=>"e","Ю"=>"yu","Я"=>"ya","а"=>"a","б"=>"b",
"в"=>"v","г"=>"g","д"=>"d","е"=>"e","ж"=>"j",
"з"=>"z","и"=>"i","й"=>"y","к"=>"k","л"=>"l",
"м"=>"m","н"=>"n","о"=>"o","п"=>"p","р"=>"r",
"с"=>"s","т"=>"t","у"=>"u","ф"=>"f","х"=>"h",
"ц"=>"ts","ч"=>"ch","ш"=>"sh","щ"=>"sch","ъ"=>"y",
"ы"=>"yi","ь"=>"","э"=>"e","ю"=>"yu","я"=>"ya",
" "=>"_","."=>"" , "/"=>"_","1"=>"1","2"=>"2","3"=>"3",
"4"=>"4","5"=>"5","6"=>"6","7"=>"7","8"=>"8",
"9"=>"9","0"=>"0"
);
return strpos($str,$str);
}

```

application/ views/ order.php

<h1>Оформления заовления</h1>

```

<?if($error){ echo $error;}}?><br/>
<?if($display_form){?>
<form action="" method="post" class="registration">
    <table border="1" class="table_form">
        <tr>
            <th>ПИБ:</th>
            <td>
                <input type="text" name="name"
value="<?=( $user_data['name']) ? $user_data['name'] : $_REQUEST['name']?>"/>
            </td>
        </tr>
        <tr>
            <th>Телефон:</th>
            <td>
                <input type="text" name="phone"
value="<?=( $user_data['phone']) ? $user_data['phone'] : $_REQUEST['phone']?>"/>
            </td>
        </tr>
        <tr>
            <th>E-mail:</th>
            <td>

```



```

        <input type="text" name="email"
value="<?=( $user_data['email']) ? $user_data['email'] : $_REQUEST['email']?>"/>
        </td>
    </tr>
    <tr>
        <th>Адреса:</th>
        <td>
            <textarea name="adress"><?=( $user_data['adress']) ?
$user_data['adress'] : $_REQUEST['adress']?></textarea>
        </td>
    </tr>
</table>
<br/>
<input type="submit" class="table_submit" style="width: auto;" name="send"
value="Надіслати дані">
</form>
<?
}else
    echo $message;
?>

```