

Державний торговельно-економічний університет
Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Розробка Web-додатку з підтримки діяльності
продуктового магазину»**

Студентки 4 курсу, 9 групи,
Спеціальності
122 «Комп'ютерні науки»

Мельник Єлизавети
Миколаївни

підпис студента

Науковий керівник
кандидат фізико-математичних
наук, доцент

Самойленко Анна
Тимофіївна

підпис керівника

Гарант освітньої програми,
кандидат технічних наук,
доцент

Демідов Павло
Георгійович

підпис керівника

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра комп'ютерних наук та інформаційних систем

Спеціальність 122 «Комп'ютерні науки»

Затверджую

Зав. кафедри _____ Пурський О.І.

«12» грудня 2022р.

Завдання

на випускн у кваліфікаційну роботу студентці

Мельник Єлизавети Миколаївни

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи

«Розробка Web-додатку з підтримки діяльності
продуктового магазину»

Затверджена наказом ректора від «09» грудня 2022р. №3332

2. Строк здачі студентом закінченої роботи 30 травня 2023 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: розробка Web-додатку з підтримки діяльності
продуктового магазину

Об'єкт дослідження: процеси підтримки діяльності магазинів за
допомогою Web-додатків

Предмет дослідження: програмний додаток для підтримки діяльності
продуктового магазину

4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Самойленко А.Т.	15.12.2022 р.	15.12.2022 р.
2	Самойленко А.Т.	15.12.2022 р.	15.12.2022 р.
3	Самойленко А.Т.	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБ-СИСТЕМ

1.1 Поняття веб-додатків, їх переваги

1.2 Загальна характеристика предметної області

1.3 Сучасний стан проектування інформаційних систем

РОЗДІЛ 2. ОРГАНІЗАЦІЯ РОЗРОБКИ WEB-ДОДАТКУ

2.1 Аналіз існуючих вітчизняних додатків

2.2 Специфікація вимог до системи

2.3 Моделювання використання Web-додатку

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Розробка архітектури Web-додатку

3.2 Програмна реалізація

3.3 Використання програмного додатку

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДОДАТОК

ДОДАТКИ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	04.10.2022	04.10.2022
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	15.12.2022	15.12.2022
3	<i>Вступ</i>	03.02.2023	03.02.2023
4	<i>РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБ-СИСТЕМ</i>	28.02.2023	28.02.2023
5	<i>РОЗДІЛ 2. ОРГАНІЗАЦІЯ РОЗРОБКИ WEB-ДОДАТКУ</i>	06.04.2023	06.04.2023
6	<i>РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ</i>	12.05.2023	12.05.2023
7	<i>Висновки</i>	15.05.2023	15.05.2023
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	30.05.2023	30.05.2023
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	31.05.2023 -01.06.2023	31.05.2023 -01.06.2023
10	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	02.06.2023	02.06.2023
11	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	05.06.2023	05.06.2023
12	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2022 р.

Керівник випускної кваліфікаційної роботи

Самойленко А.Т.

(прізвище, ініціали, підпис)

Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

Завдання прийняв студент-дипломник

Мельник Є.М.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

У випускній кваліфікаційній роботі здійснено комплексну розробку Web-додатку для підтримки діяльності продуктового магазину з метою підвищення ефективності роботи магазину. Теоретично обґрунтовано основні поняття веб-додатків та їх переваги, проаналізований сучасний стан проектування інформаційних систем. Проведено аналіз існуючих вітчизняних додатків і виходячи з цього запропоновано концепцію створення додатку. Також розроблено специфікацію вимог до системи і розроблено архітектуру майбутнього додатку. На основі цього створено Web-додаток з підтримки діяльності продуктового магазину.

Ключові слова: Web-додаток, розробка, вимоги, Java, HTML, CSS, JavaScript.

Abstract

In the graduation qualification work, a comprehensive development of a Web application was carried out to support the operation of a grocery store with a method of increasing the efficiency of the store. The main concepts of web applications and their advantages are theoretically substantiated, the current state of information systems design is analyzed. An analysis of existing domestic applications was carried out and, based on this, the concept of creating an application was proposed. The system requirements specification was also developed and the architecture of the future application was developed. Based on this, a Web application was created to support the activities of a grocery store.

Keywords: Web application, development, requirements, Java, HTML, CSS, JavaScript.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБ-СИСТЕМ	11
1.1 Поняття веб-додатків, їх переваги.....	11
1.2 Загальна характеристика предметної області.....	14
1.3 Сучасний стан проектування інформаційних систем.....	15
РОЗДІЛ 2. ОРГАНІЗАЦІЯ РОЗРОБКИ WEB-ДОДАТКУ	19
2.1 Аналіз існуючих вітчизняних додатків.....	19
2.2 Специфікація вимог до системи.....	26
2.3 Моделювання використання Web-додатку.....	27
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ	31
3.1 Розробка архітектури Web-додатку.....	31
3.2 Програмна реалізація.....	35
3.3 Використання програмного додатку.....	48
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53

ВСТУП

Забезпечення ефективної роботи Web-додатку вимагає не тільки розробки, але й моніторингу поточного стану додатків, з метою актуалізації продукту і визначення для майбутніх перспектив розвитку. Проведення аналізу існуючих додатків є однією з важливих цілей, задля створення актуальної картини вигляду додатку. В нинішній час Web-додаток є обличчям компанії, який показує її успішність. Характерною особливістю розробки Web-додатків є стрімка зміна тенденцій, тому необхідно приділити особливу увагу проведенню аналізу сучасного стану технологій та головних тенденцій їх розвитку. Задля цього необхідно визначити саме сучасний стан проектування інформаційних систем. Головним завданням додатку є представлення інформації про компанію, продукти та послуги, які вона надає.

Однією з найважливіших науково-технічних проблем на даний час є проблема неймовірно стрімкого зростання технологій і кількості додатків, які створюються щодня. Через це є актуальною проблема конкурентоспроможності додатків, адже необхідно заохочувати нових клієнтів, саме тому постає питання яким повинен бути додаток, задля максимально ефективного просування компанії. Безпосередньо виходячи з цього необхідно створити додаток, який буде поєднанням сучасних інформаційних, програмних, медійних і візуальних засобів, що пов'язані між собою логічно. Таким чином, постає необхідність розробки методів і засобів для реалізації програмного додатку, з метою створення сучасного та актуального Web-додатку, усе це й зумовило **актуальність** обраної теми дослідження, її мету і завдання.

Мета і завдання дослідження. Метою даного дослідження є розробити додаток, який буде зручним і зовнішньо привабливим для потенційних клієнтів з урахуванням усіх особливостей продуктового магазину. Для досягнення поставленої мети необхідно було вирішити наступні **завдання**:

- Провести аналіз предметної області
- Створити інтерфейс додатку, який буде відображати усю необхідну інформацію
- Розробити базу даних, в якій буде зберігатися вся інформація щодо замовлень
- Розмістити базу даних на локальному сервері чи сервері хостинга та підключити її до програми
- Провести тестування додатку

Об'єкт дослідження: процеси підтримки діяльності магазинів за допомогою Web-додатків.

Предмет дослідження: засоби розробки програмного додатку для підтримки діяльності продуктового магазину.

Методи дослідження: Теоретичною основою дослідження є загальнонауковий аналітичний метод, а також системний підхід і праці провідних учених з проблем дослідження та оцінка рівня розвитку сучасного стану проектування інформаційних систем. Для практичного вирішення поставлених задач використовувалися такі методи:

- Загальнонауковий аналітичний метод (розділ 1);
- Методи моделювання інформаційних систем та процесів взаємодії між ними для підвищення ефективності використання;
- Методи моделювання діаграм варіантів використання задля співвідношення потреб і їх виконання;
- Методи програмування, для створення Web- додатку для підтримки діяльності продуктового магазину.

Практичне значення. Отриманні результати, можуть бути використані для створення, сучасного, ефективного та конкурентоспроможного Web-додатку для підтримки діяльності продуктового магазину. Програмна

реалізація додатку, надає можливість використання проаналізованих методів, моделей та створеної архітектури додатку.

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається зі вступу, трьох розділів, висновків та списку використаних джерел із 22 найменувань і містить 41 сторінку основного тексту, 31 рисунок і 1 таблицю.



РОЗДІЛ 1.

ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБ-СИСТЕМ

1.1 Поняття веб-додатків, їх переваги

Веб-додатки - це програми, які працюють в мережі Інтернет, а не на локальному комп'ютері користувача. Вони доступні з будь-якого пристрою з підключенням до Інтернету та можуть бути запуснені в будь-якому веб-браузері. Веб-додатки виконують різні завдання, від онлайн-ігор до електронної пошти та відеозв'язку [1].

Переваги веб-додатків полягають в їх доступності та простоті використання. Користувачам не потрібно завантажувати та встановлювати окремі програми, оскільки веб-додатки працюють просто у веб-браузері [2]. Вони можуть бути доступні з будь-якого пристрою та операційної системи, що дозволяє користувачам працювати зі своїми даними з будь-якого місця та в будь-який час.

Також веб-додатки можуть бути легко оновлені та модифіковані, оскільки їх функціональність зберігається на сервері, а не на користувачевій локальній машині. Це дозволяє розробникам веб-додатків швидко внести зміни та відповісти на нові вимоги користувачів.

Існує також можливість інтеграції веб-додатків з іншими системами та сервісами. Наприклад, веб-додатки можуть бути пов'язані зі соціальними мережами, електронними поштовими сервісами або системами управління проектами, що забезпечує більш ефективну роботу та зменшує час, необхідний для переміщення між різними системами.

Окрім того, веб-додатки мають багато переваг в порівнянні зі стандартними десктоп-додатками. Один з найважливіших факторів - це те,

що веб-додатки можуть бути запущені на будь-якій операційній системі, оскільки вони працюють у веб-браузері. Це дає користувачам вибір, що є важливим в сучасному світі, де люди використовують різні операційні системи і пристрої [2].

Іншою перевагою веб-додатків є те, що їх легко оновлювати, тому користувачі можуть отримувати нові функції та виправлення помилок без необхідності завантаження та встановлення нової версії додатку. Це також полегшує роботу розробників, оскільки вони можуть швидко виправляти помилки та оновлювати свій продукт без необхідності випуску патчів або оновлень для кожного користувача окремо [1].

Крім того, веб-додатки зазвичай використовують механізм авторизації, що дозволяє користувачам захищати свої дані від несанкціонованого доступу. Це особливо важливо для бізнес-додатків, які зберігають конфіденційну інформацію про компанію та її клієнтів.

Взагалі, веб-додатки стають все більш популярними через свою доступність, гнучкість та простоту використання. Це не означає, що вони є універсальним рішенням для всіх випадків, проте вони дійсно важливі для багатьох.

З іншого боку, веб-додатки не вимагають встановлення на кожному пристрої, що дає можливість отримати доступ до програми з будь-якого пристрою з доступом до Інтернету. Це забезпечує зручність та доступність користувачам, зокрема, тим, хто часто подорожує або працює з різних пристроїв.

Крім того, веб-додатки можуть бути оновлені централізовано, що дозволяє розробникам виправляти помилки та додавати нові функції без необхідності встановлювати оновлення на кожному пристрої окремо. Це робить веб-додатки більш гнучкими та ефективними у порівнянні зі звичайними десктопними програмами.

Крім того, веб-додатки дозволяють користувачам зберігати дані в хмарі, що забезпечує їх доступність з будь-якого пристрою. Наприклад, електронна пошта, онлайн-банкінг та інші веб-додатки можуть зберігати дані у безпечному місці та забезпечувати доступ до них з будь-якої точки світу з Інтернетом.

Веб-додатки також можуть бути інтегровані з іншими системами, що дає можливість автоматизувати бізнес-процеси та полегшувати роботу співробітників. Наприклад, веб-додаток для управління відділом продажів може бути інтегрований з системою управління клієнтською базою даних, що дозволяє автоматично відстежувати нові замовлення та зв'язок з клієнтами [2].

Отже, веб-додатки мають багато переваг порівняно зі звичайними десктопними програмами, зокрема:

- **Доступність.** Веб-додатки можуть бути доступні з будь-якого комп'ютера або мобільного пристрою з доступом до Інтернету. Вам не потрібно встановлювати програмне забезпечення на кожен комп'ютер, що використовується, що дозволяє економити час і кошти.
- **Оновлення.** Оновлення веб-додатків можуть бути проведені централізовано на веб-сервері, що забезпечує просте та швидке оновлення для всіх користувачів.
- **Масштабованість.** Веб-додатки легко масштабуються. Якщо ви маєте більше користувачів, просто додайте ще один веб-сервер або збільшіть потужність існуючого. Це значно простіше, ніж масштабування традиційних настільних додатків.
- **Безпека.** Веб-додатки можуть бути більш безпечними, ніж традиційні настільні додатки. Завдяки тому, що всі дані зберігаються на серверах, що знаходяться в безпечному місці, користувачам не потрібно бути занадто обережними щодо захисту своїх даних.

- Сумісність. Веб-додатки зазвичай мають більшу сумісність.

1.2 Загальна характеристика предметної області

Web-додаток для продуктового магазину може забезпечувати підтримку діяльності магазину у ряді аспектів, включаючи управління продуктами та запасами, замовленнями та оплатою, взаємодією з клієнтами та аналізом продажів.

Для розробки такого Web-додатку можуть бути використані різні системи, такі як:

- Система управління контентом (CMS): вона може дозволити додавати, редагувати та видаляти продукти, керувати запасами та оновлювати ціни.
- Система управління замовленнями: вона може допомогти у прийманні та обробці замовлень, відстеженні доставки та керуванні поверненнями.
- Система оплати: вона може забезпечити безпечну оплату онлайн-замовлень.
- Система аналітики: вона може дозволити аналізувати продажі та тренди в споживанні, що допоможе в управлінні запасами та маркетингових стратегіях.
- Система взаємодії з клієнтами: вона може забезпечити зручний спосіб взаємодії клієнтів з магазином, включаючи зворотний зв'язок, онлайн-чат та інше.

Основною метою розробки такого додатку є автоматизація процесів замовлення та продажу товарів в інтернет-магазині, забезпечення зручного

та швидкого вибору товарів, оплати та доставки, а також збільшення ефективності управління продуктовим магазином в цілому [3].

Перевагами використання Web-додатку для продуктового магазину є:

- Зручність та доступність - покупці можуть здійснювати замовлення та оплату товарів в будь-який зручний для них час та з будь-якої точки світу.
- Швидкість та ефективність - автоматизація процесів замовлення та оплати дозволяє забезпечити швидке та ефективне обслуговування покупців, зменшити час на обробку замовлень та покращити якість обслуговування.
- Масштабованість та гнучкість - Web-додаток може бути з легкістю збільшений та модифікований в залежності від потреб магазину та ринку [4].
- Зменшення витрат - автоматизація процесів дозволяє зменшити витрати

Сучасний стан проблеми полягає в тому, що продуктові магазини все більше переходять на онлайн-платформи, щоб забезпечити зручний та швидкий доступ до продуктів. Проте, цей переходовий процес не є простим, і вимагає багато ресурсів та зусиль для розробки та підтримки ефективних веб-додатків. Основні виклики, з якими стикаються продуктові магазини при розробці Web-додатків, полягають у забезпеченні безпеки та конфіденційності даних клієнтів.

1.3 Сучасний стан проектування інформаційних систем

Проектування інформаційних систем є складним та багатоплановим процесом, який постійно розвивається та покращується. Сучасний стан проектування інформаційних систем характеризується використанням

новітніх технологій, підходів та методів, які дозволяють ефективно створювати інформаційні системи з високим рівнем функціональності та якості [5].

Одним з основних тенденцій у сучасному проектуванні інформаційних систем є використання хмарних технологій. Хмарні технології дозволяють розміщувати додатки та дані в інтернеті, що забезпечує більш високий рівень доступності, масштабованості та безпеки[6]. За даними Forbes, в 2023 році прогнозується зростання глобального ринку хмарних технологій на 28% щорічно, що свідчить про великий попит на ці технології.

Ще одним важливим напрямом у сучасному проектуванні інформаційних систем є використання штучного інтелекту. Штучний інтелект дозволяє здійснювати аналіз та обробку великих обсягів даних, автоматизувати бізнес-процеси та забезпечувати підвищення ефективності діяльності [7]. За даними MarketsandMarkets, до 2025 року прогнозується зростання ринку штучного інтелекту до 190 мільярдів доларів, що свідчить про значний інтерес до цієї технології.

Ще одним важливим напрямом у сучасному проектуванні інформаційних систем є використання блокчейн технологій. Блокчейн дозволяє зберігати дані в безпечному та доступному для всіх вигляді. Він забезпечує цілісність даних та недоступність для зміни, що робить його ідеальним вибором для зберігання фінансових, медичних, логістичних та інших даних. Дані у блокчейні зберігаються в блоках, які містять у собі інформацію про попередні блоки. Таким чином, будь-яка зміна в одному з блоків призведе до зміни хеш-функції попереднього блоку та всіх наступних блоків [8].

Одна з переваг блокчейну полягає в тому, що він забезпечує високий рівень безпеки та надійності. Блокчейн заснований на криптографії, що забезпечує конфіденційність та інтегритет даних. Кожен блок містить у

собі унікальний хеш, що підтверджує його цілісність. Також блокчейн забезпечує високу доступність даних, оскільки вони зберігаються на різних комп'ютерах, що дозволяє уникнути втрати даних в разі збоїв в системі.

Іншою перевагою блокчейну є його прозорість та відкритість. Кожен блок містить інформацію про всі попередні блоки, що забезпечує прозорість в системі. Крім того, блокчейн відкритий для всіх користувачів, що дозволяє забезпечити довіру між учасниками системи [8].

Ще однією перевагою блокчейну є його ефективність та швидкість. Блокчейн може зберігати великі обсяги даних та обробляти їх швидко та ефективно. Це робить його ідеальним вибором для систем з великою кількістю транзакцій.

Також однією з ключових тенденцій сучасного проектування інформаційних систем є використання методів Agile та DevOps. Agile є методологією розробки програмного забезпечення, що передбачає ітераційний та інкрементальний підхід до проектування. Такий підхід дозволяє прискорити процес розробки та зменшити час між виходом нових версій продукту.

DevOps - це філософія розробки програмного забезпечення, яка поєднує методи Agile з принципами автоматизації інфраструктури, що дозволяє забезпечити швидкий та безперервний процес розробки, тестування та впровадження [8].

Нарешті, важливим аспектом проектування інформаційних систем є їх безпека. У зв'язку зі зростаючими загрозами кібербезпеці, проектування безпечних інформаційних систем стає все більш важливим завданням. Сьогодні багато компаній використовують інструменти, такі як інтелектуальні системи виявлення загроз, шифрування даних та механізми аутентифікації, щоб забезпечити безпеку своїх інформаційних систем.

Узагалі, сучасний стан проектування інформаційних систем постійно змінюється та розвивається. Кожен день на ринку з'являються нові технології, інструменти та методи, які дозволяють створювати більш ефективні та безпечні інформаційні системи. Однак, необхідно пам'ятати, що проектування інформаційних систем - це складний процес, який вимагає від експертів багато знань та досвіду, а також уважної уваги до потреб користувачів та забезпечення безпеки даних [5].



РОЗДІЛ 2.

ОРГАНІЗАЦІЯ РОЗРОБКИ WEB-ДОДАТКУ

2.1 Аналіз існуючих вітчизняних додатків

На сьогоднішній день існує безліч Web-додатків для замовлення продуктових товарів в мережі інтернет. Для аналізу мною був обраний додаток «Zakaz.ua» [10]. Zakaz.ua - це веб-додаток для онлайн замовлення продуктів та доставки на дім. З точки зору створення веб-додатку, цей додаток є прикладом добре побудованої та ефективної платформи. На його прикладі можна розглянути основні задачі додатку, такі як дизайн, функціональність, безпека, підтримка клієнтів.

Головна сторінка додатку зображена на рисунку 2.1

Дизайн та користувацький досвід: веб-додаток має чистий та простий дизайн, що забезпечує легку навігацію та швидкий доступ до інформації. Зліва меню з необхідними категоріями товарів. Меню не перевантажено, немає зайвої інформації, яка могла б відволікати від переліку головних категорій. Центральна частина головної сторінки містить опцію зміни мови, підтримки, інформацію про партнерів.

Користувач легко обирає потрібний магазин і товар з випадаючого списку.

Зображено на рисунку 2.2

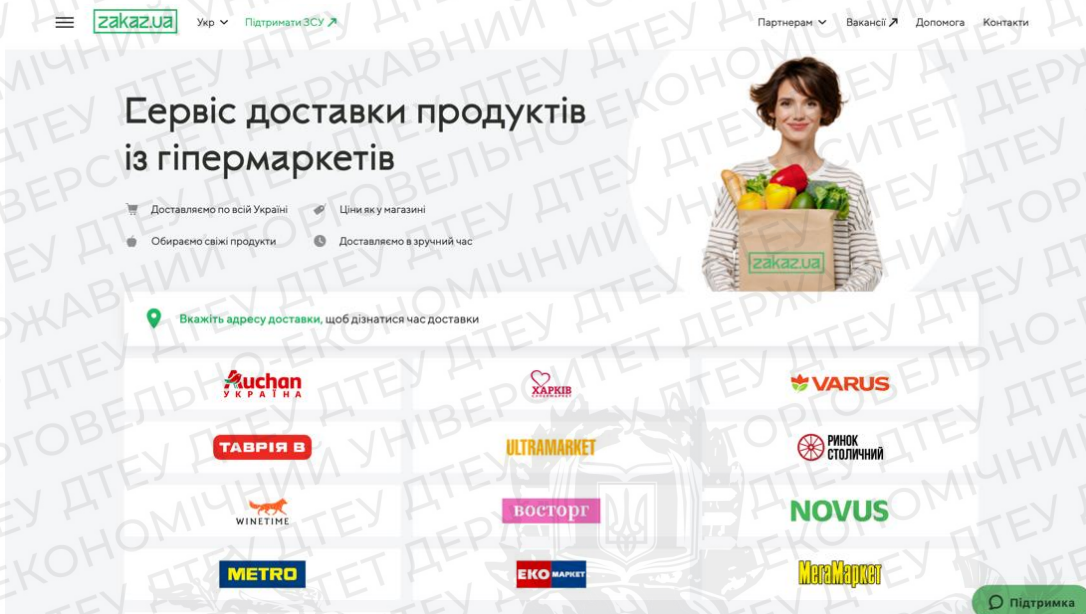


Рис. 2.1 – Головна сторінка Web-додатку «Zakaz.ua»

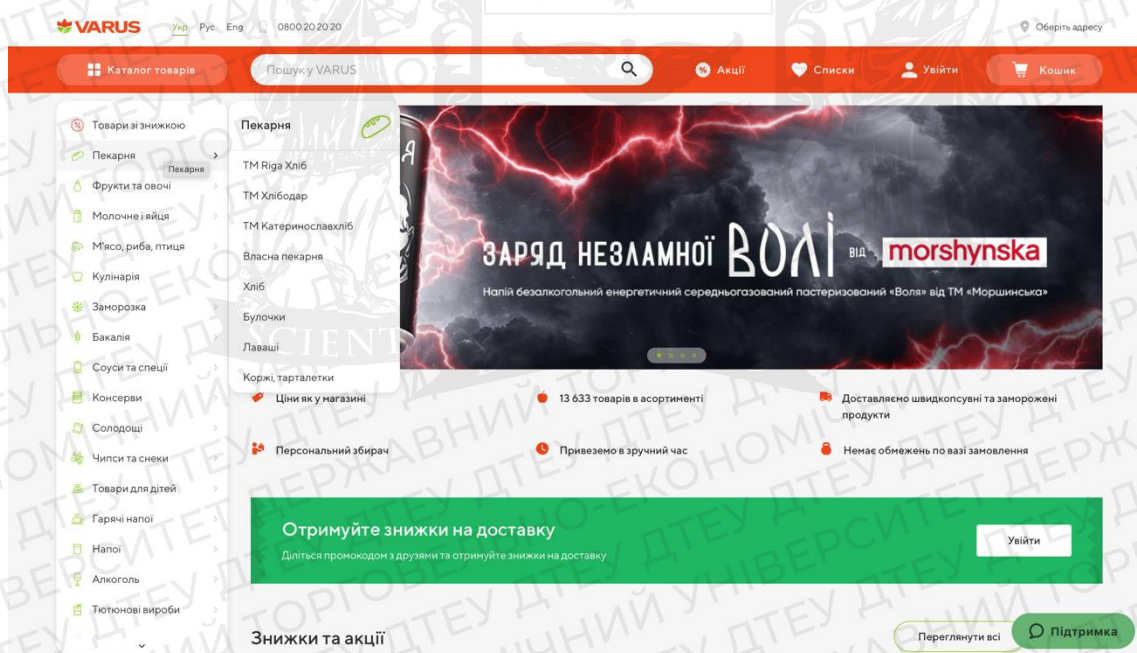


Рис. 2.2 – Функції пошуку товари, та обрання певної категорії.

Також на сайті немає футера, як такого, він був перенесений вниз, завдяки чому займає менше простору. Zakaz.ua має різноманітні канали підтримки клієнтів, такі як онлайн-чат, електронна пошта та телефонна лінія підтримки. Це зображено на рисунку 2.3-2.4

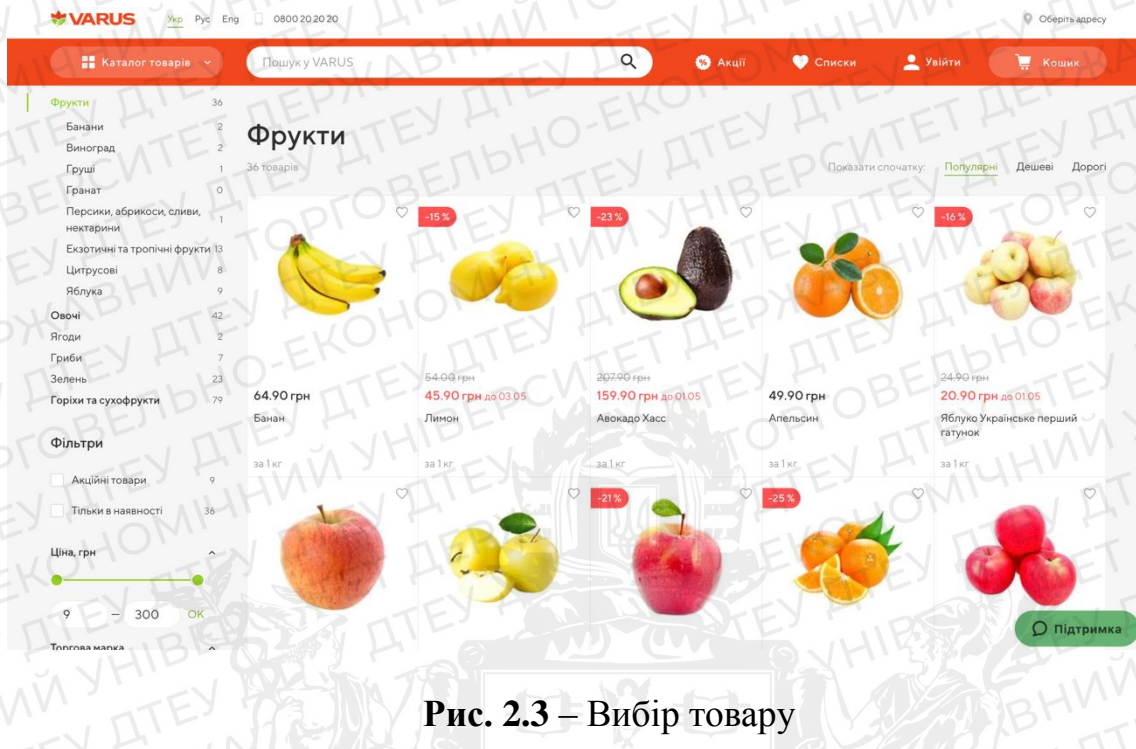


Рис. 2.3 – Вибір товару

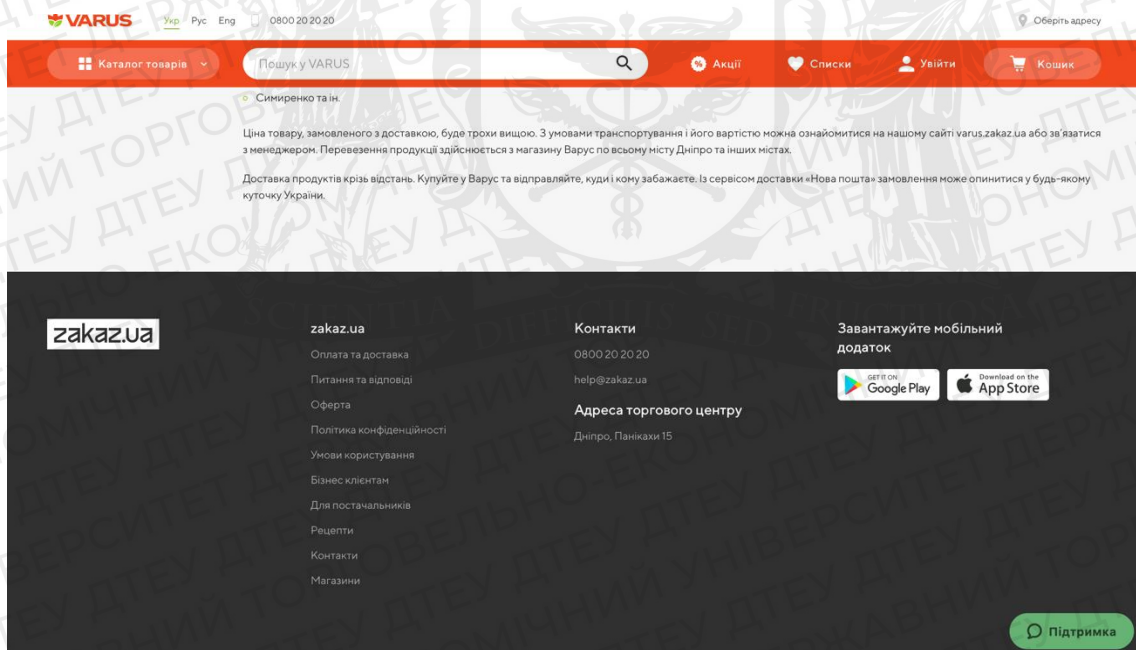


Рис. 2.4 – футер додатку та служби підтримки

Web-додаток має всі необхідні функції для замовлення продуктів та їх доставки, включаючи реєстрацію користувача, пошук продуктів, додавання їх до кошика, вибір способу оплати та доставки. Додаток «Zakaz.ua» працює досить швидко та стабільно, завдяки чому клієнти

можуть зробити замовлення за декілька хвилин. Зображено на рисунку 2.5-2.6

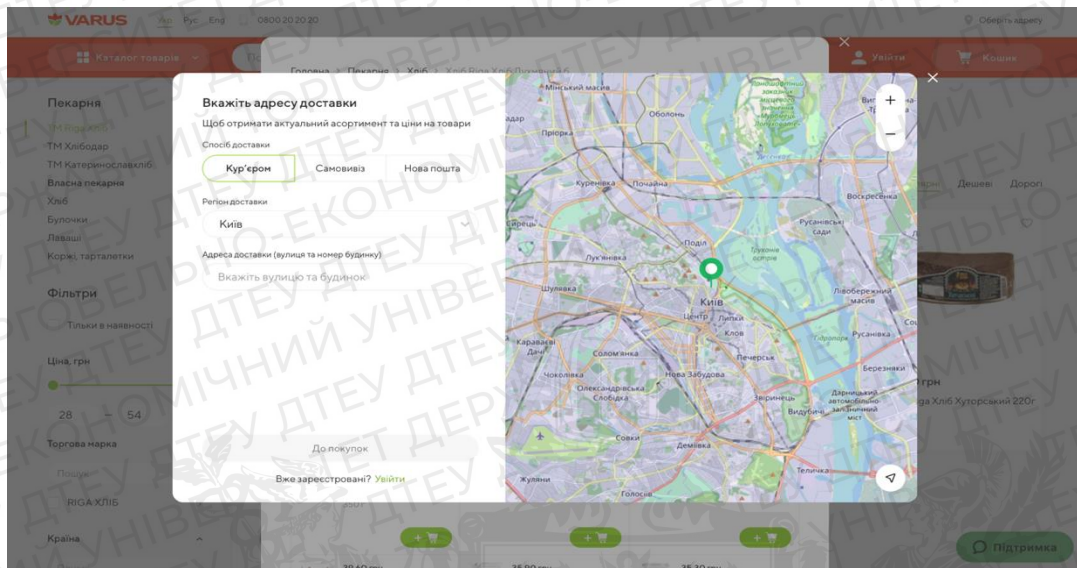


Рис. 2.5 – Приклад доставки

Веб-додаток має надійний механізм захисту даних користувачів та платежів, що дозволяє клієнтам здійснювати покупки безпечно та з конфіденційністю. Також додаток «Zakaz.ua» має систему аналітики, що дозволяє аналізувати продажі, поведінку користувачів та інші метрики, що допомагає оптимізувати роботу веб-додатку та зробити його максимально зручним для користувачів. Зображено на рисунку 2.6

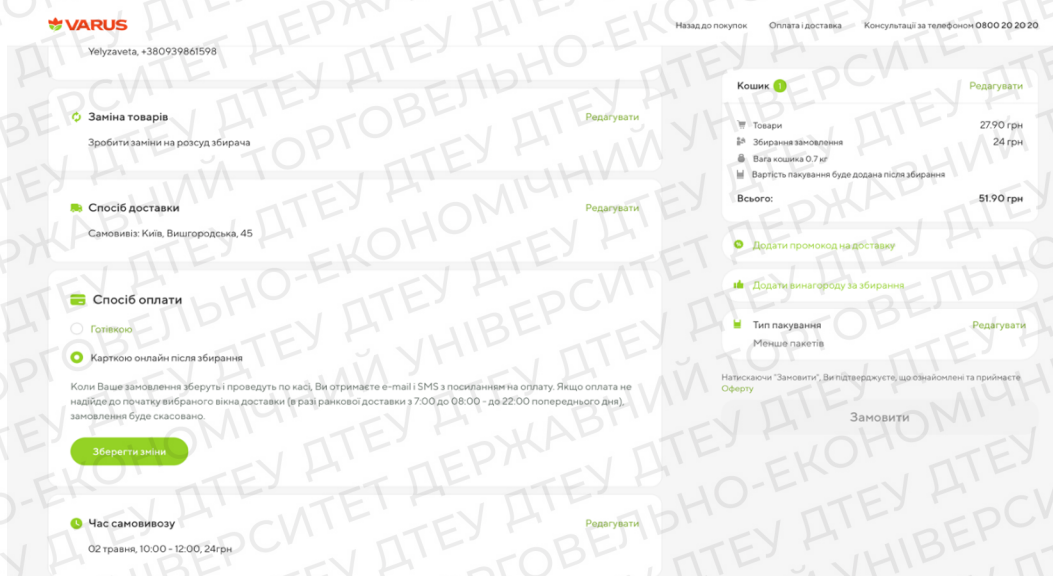


Рис. 2.6 – Приклад оформлення замовлення

Наступний приклад це додаток «АТБ» [11].

«АТБ» - це веб-додаток для онлайн замовлення продуктів та доставки на дім в магазинах мережі «АТБ». Ось деякі аспекти, які можуть бути виділені при аналізі з точки зору створення веб-додатку:

Веб-додаток має простий та зрозумілий дизайн, що дозволяє клієнтам легко орієнтуватись та знайти потрібні продукти. Однак, дизайн може виглядати дещо застарілим та не настільки привабливим для клієнтів.

Головна сторінка додатку зображена на рисунку 2.7

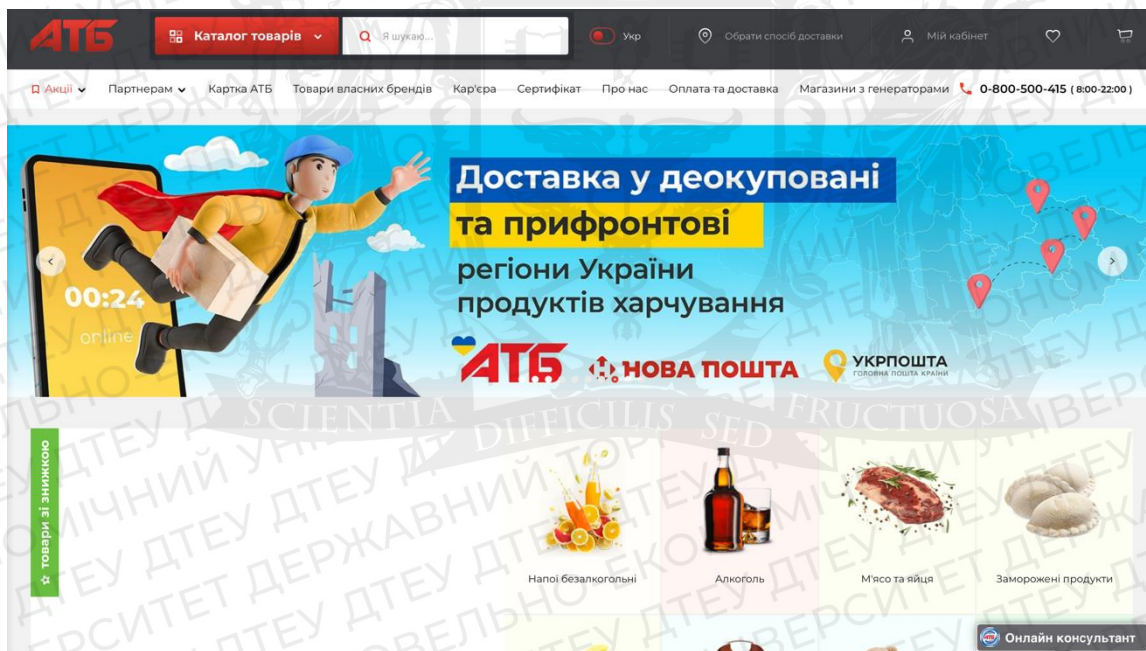


Рис. 2.7 – Головна сторінка Web-додатку «АТБ»

Веб-додаток має всі необхідні функції для замовлення продуктів та доставки, а також можливість зберігати списки покупок та переглядати історію замовлень. Проте, на відміну від деяких інших веб-додатків, «АТБ» не має можливості оплати замовлення готівкою, тому користувачам потрібно оплачувати замовлення лише онлайн при оформленні. Зображено на рисунку 2.8-2.9

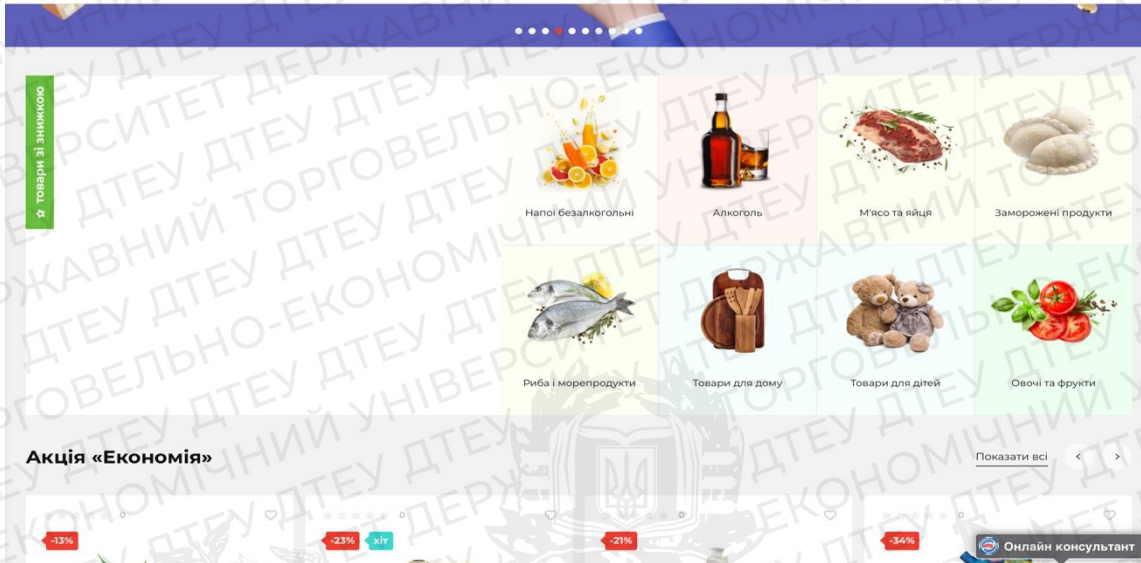


Рис. 2.8 – Функції пошуку товари, та обрання певної категорії.

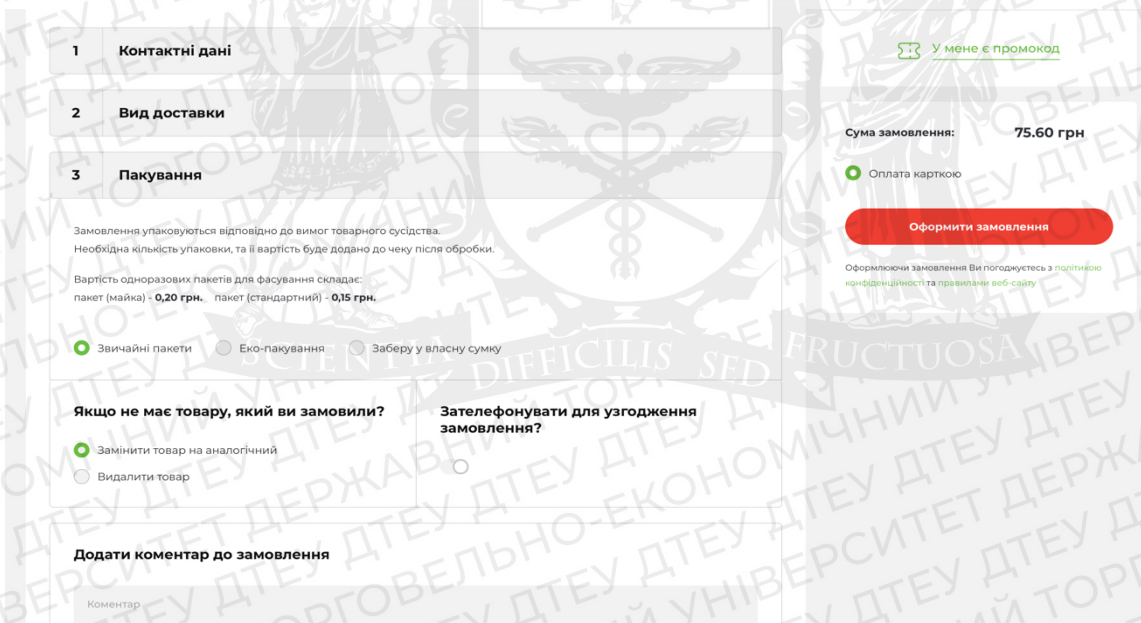


Рис. 2.9 – Приклад оформлення замовлення

«АТБ» працює достатньо швидко, однак на деяких сторінках може з'являтися затримки, особливо при виборі продуктів та оформленні замовлення. Веб-додаток має механізми захисту даних користувачів та платежів, що дозволяє клієнтам здійснювати покупки безпечно та з конфіденційністю.

«АТБ» має різні канали підтримки клієнтів, такі як онлайн-чат та телефонна лінія підтримки, проте може бути не досить швидкою в обробці запитів клієнтів. Зображено на рисунку 2.10. Також «АТБ» має гарну систему аналітики, що дозволяє дізнаватися вподобання користувачів.

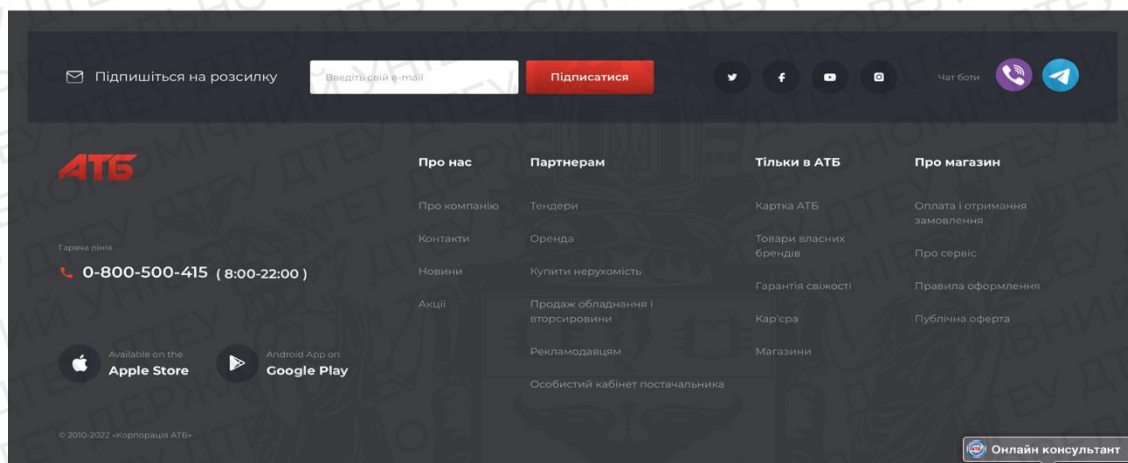


Рис. 2.10 – Служби підтримки

Після аналізу головних елементів сайтів-аналогів створимо порівняльну таблицю, в якій відобразити основні переваги та недоліки розглянутих сайтів. Результати аналізу зображені в таблиці 1.1

Таблиця 2.1 – Порівняльний аналіз характеристик веб-сайтів

Характеристика	«Zakaz.ua»	«АТБ»
Сучасний дизайн	+	-
Навігація сайтом	+	+
Оплата карткою та готівкою	+	-
Захист даних	+	+
Швидка підтримка	+	-
Швидка робота сайту	+	-

Проведений аналіз таблиці порівняння дає нам можливість виділити основні переваги та недоліки розглянутих сайтів. За допомогою даної таблиці ми використаємо корисний функціонал сайту та побудуємо свій додаток.

2.2 Специфікація вимог до системи

Метою випускної кваліфікаційної роботи є створення веб-додатку продуктового магазину. Додаток дозволить замовляти товари додому і як наслідок це збільшить кількість покупців. Інтерфейс веб-додатку необхідно розробити в діловому стилі. Також він має бути сучасним, для зацікавлення нових клієнтів. І водночас він повинен бути простий, аби кожен з легкістю міг розібратися.

Проаналізувавши основні переваги і недоліків сайтів-аналогів та виконавши аналіз предметної області, були визначені головні вимоги до створення додатку:

- Створити кошик та можливість додавати до неї замовлення
- Виконати розподіл інформації по блокам задля легкої навігації
- Створити можливість оплати декількома способами

Для досягнення поставлених цілей та вимог потрібно вирішити наступні задачі, які дозволять виконати проєкт:

- Провести аналіз предметної області
- Створити інтерфейс, який буде відображати лише необхідну інформацію
- Провести тестування
- Скласти необхідні документи щодо проєкту

Мови, що будуть використовуватися для розробки: HTML, CSS, JAVA, JAVA SCRIPT. Ці мови програмування є універсальними для створення веб-додатків. Вони дозволяють створити інтерфейс та функціонал.

2.3 Моделювання використання Web-додатку

Моделювання використання Web-додатку продуктового магазину можна провести за допомогою визначення основних функцій, які має виконувати такий додаток, та процесів, які повинні відбуватися під час його використання.

Основними функціями можуть бути:

- Перегляд товарів: Користувач може переглянути список товарів, які доступні в магазині. Для кожного товару може бути вказана його назва, ціна та інші характеристики.
- Додавання товару в кошик: Користувач може додати товар в кошик, щоб пізніше придбати його.
- Оформлення замовлення: Користувач може оформити замовлення на товари, які знаходяться в його кошику. Для цього йому потрібно буде вказати свої контактні дані та адресу доставки.
- Оплата товарів: Користувач може оплатити своє замовлення, використовуючи один з доступних методів оплати, наприклад, кредитною карткою.

Виконати ці задачі можна за допомогою Unified Modeling Language (UML) - це мова моделювання, яка дозволяє розробляти, відображати та документувати програмні системи. UML є стандартом, прийнятим Об'єднанням Міжнародних стандартів зі створення програмного забезпечення (OMG) використовується в багатьох галузях, пов'язаних з розробкою програмного забезпечення. UML, є способом візуального

представлення архітектури, проектування та реалізації складних програмних систем. Коли йде процес написання коду, у програмі є тисячі рядків, і важко відстежити взаємозв'язки та ієрархії в програмній системі. Діаграмам UML поділяють цю програмну систему на компоненти та підкомпоненти [17].

Також UML дозволяє створювати графічні моделі, які відображають різні аспекти програмної системи, такі як її структура, поведінка, взаємодія з користувачами та інші. Моделі, створені з використанням UML, допомагають розробникам та іншим учасникам проекту краще зрозуміти систему, її компоненти та взаємодію між ними.

Існує декілька варіантів діаграм UML, і кожна призначена з для певної мети, незважаючи на те, вона розробляється до впровадження чи потім (як одна з частин документації). Також кожна діаграма має свої власні правила та синтаксис, що дозволяє відображати інформацію в зручному форматі[18].

Є два найбільш розповсюджених типи діаграм, вони включають в себе всі інші категорії, - це поведінкова діаграма UML та структурна діаграма UML. Згідно з назвою діаграм, деякі з них аналізують і зображують структуру процесу або системи, в той час, як інші діаграми характеризують поведінку системи та її компонентів[17].

Діаграма випадків використання відноситься до поведінкових діаграм. Модель використання описує функціональні вимоги системи з точки зору випадків використання. Це модель передбачуваної функціональності системи (випадки використання) та її середовища (актори). Приклади використання дозволяють співвідносити те, що потрібно від системи, з тим, як система задовольняє ці потреби[17].

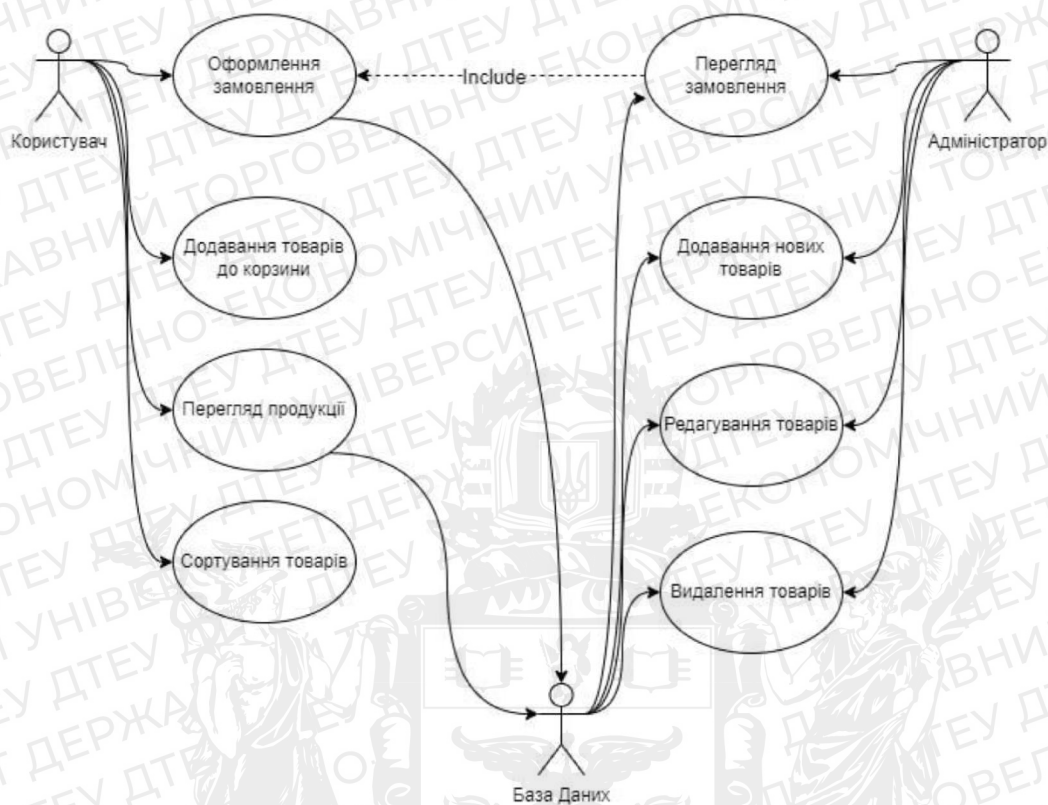


Рис. 2.11 – Діаграма варіантів використання

Також використовується IDEF1 (Integration Definition for Information Modeling) - це стандарт, який визначає методологію моделювання інформаційних систем та процесів взаємодії між ними. Методологія IDEF1 включає в себе опис інформаційних потоків та їх взаємодії, опис потреб користувачів в інформації, опис інформаційних ресурсів, які використовуються в системі [12].

IDEF1 також включає в себе методики аналізу та проектування баз даних, що дозволяють визначити структуру та зміст даних, які використовуються в системі. Використання методології IDEF1 дозволяє покращити процеси взаємодії між користувачами та системами, підвищити ефективність використання інформаційних ресурсів та знизити витрати на розробку та підтримку систем[16].

При побудові інформаційної моделі особа, що проектує завжди працює з двома головними областями, перша це реальний світ, або

сукупність фізичних і інтелектуальних об’єктів і всі властивості цих об’єктів і залежності між ними. Друга це інформаційна область, вона включає в себе інформаційні відображені об’єктів першої області і їх властивостей [11]. По суті, інформаційне відображення не є об’єктом реального світу, проте його зміна, як правило, є наслідком деякої зміни відповідного йому об’єкта реально світу. Декомпозиція оформлення замовлення представлена на рисунку 2.12.

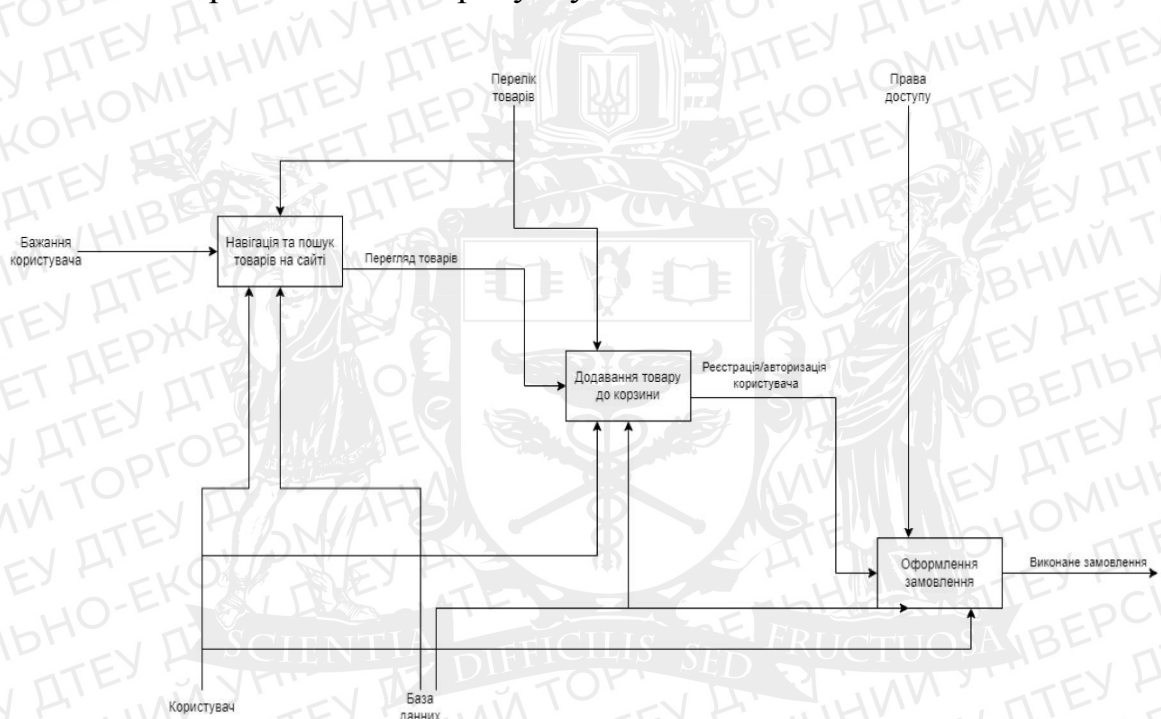


Рис. 2.12 – Діаграма декомпозиції оформлення замовлення (IDEF1)

РОЗДІЛ 3.

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Розробка архітектури Web-додатку

Архітектура web-додатку з підтримки діяльності продуктового магазину створена за сучасним та популярним принципом, в якій загальну структуру розділено на дві частини:

- Взаємодія з клієнтом (front-end) – в цій частині клієнту відображаються web- елементи та інформація з бази даних.
- Взаємодія з сервером (back-end) – в даній частині розробляється основний функціонал проекту (логіка роботи web-додатку, спілкування за допомогою API з клієнтською частиною, CRUD).

Web-додаток був розроблений згідно з технічним завданням.

Додаток має базові функції інтернет магазину та відповідає сучасним критеріям. Цей додаток є конкурентно здатним з іншими магазинами, що зосереджуються на доставці їжі та/або продуктів.

Візуальне відображення архітектури зображено на рисунку 3.1

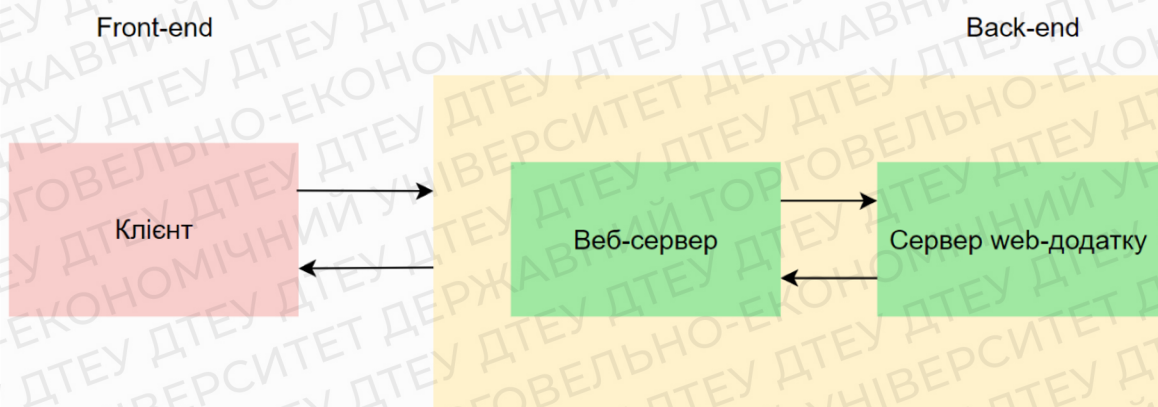


Рис. 3.1- Архітектура web-додатку

Головна ціль веб-додатку – це надання користувачеві можливості зроби запит на доставку бажаного продукту до дому. Як і було зазначено в технічному завданні сайт був розроблений використовуючи наступні мов програмування: JAVA, JAVASCRIPT, HTML та CSS.

Інтерфейс має сучасний дизайн та є інтуїтивно зрозумілим для будь-якого користувача який вміє користуватися персональним комп'ютером. Основні елементи, що відповідають за навігацію представлені у вигляді кнопок, що відповідають за певну дію. У процесі реалізації були сторінні такі розділи:

- Головна сторінка – сторінка-слайдер, на якій зображено товари, які пропонує магазин і зверху знаходиться навігаційна панель сайту.
- Навігаційна панель сайту – яка містить усі головні сторінки сайту і дає змогу легко знаходити потрібну інформацію на сайті.
- Сторінка меню – вона має список, що розкривається, де представлені товари розподілені за розділами такі, як («Продукти», «Напої» та «Кулінарія», кожна товар має кнопку купити і також є кошик, де вказується загальна сума замовлення.
- Сторінка підтримки – де наведена основна інформація про магазин та способи зв'язку, і відповідь на часті запитання.
- Сторінка оформлення замовлення – на цій сторінці клієнт вказує таку інформацію, як ім'я, номер телефону, електронна пошта, адреса доставки, обирає спосіб оплати і оформлює замовлення.

На рисунку 3.2 зображено макет сторінка меню веб-додатку

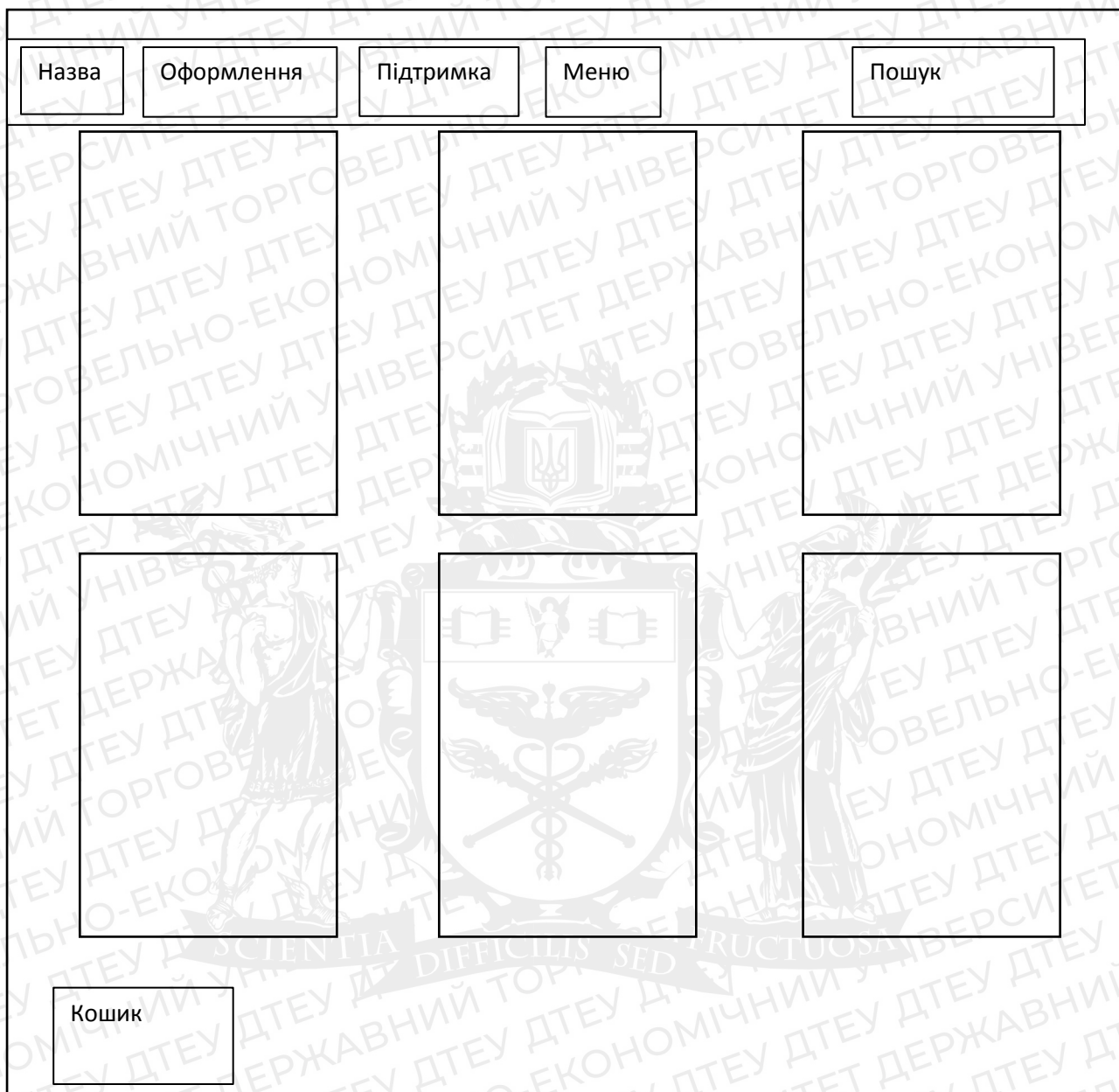
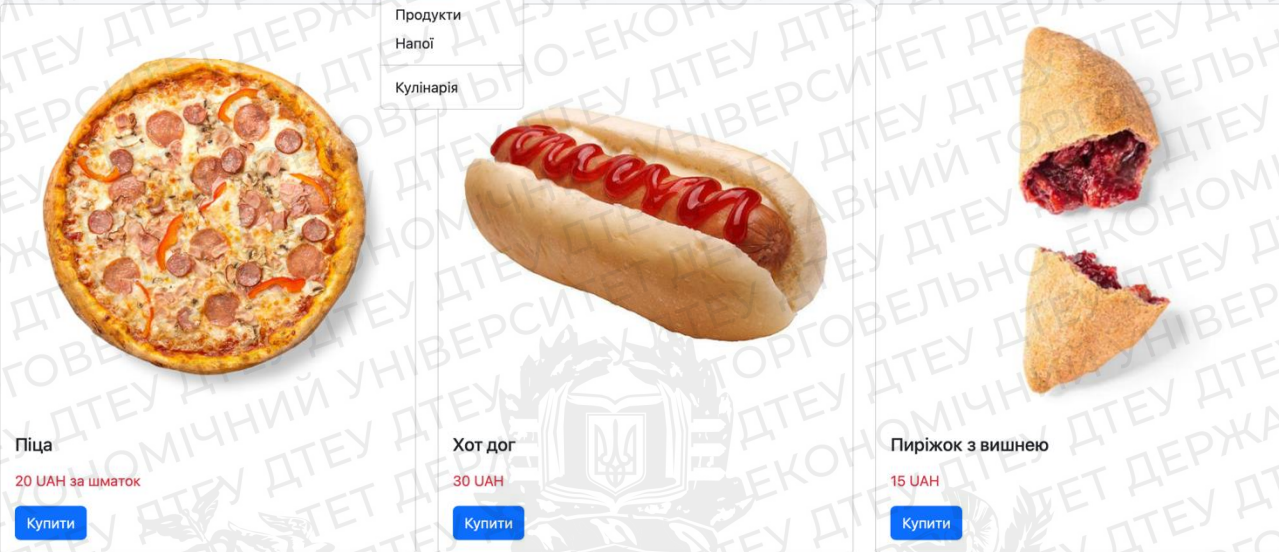


Рис. 3.2 – Макет сторінки меню Web-додатку

Ссилаючись на макет, що зображений на рисунку 3.2, а також технічне завдання була розроблена сторінка меню. На цій сторінці користувач може знайти бажаний товар, відфільтрувати товар за категоріями, додати його до кошика. Також користувач має можливість з сторінки-меню побачити кошик товарів та потрапити на сторінку з його замовленнями.

Сторінка меню проілюстрована на рисунку 3.3-3.4

Продукти
Напої
Кулінарія

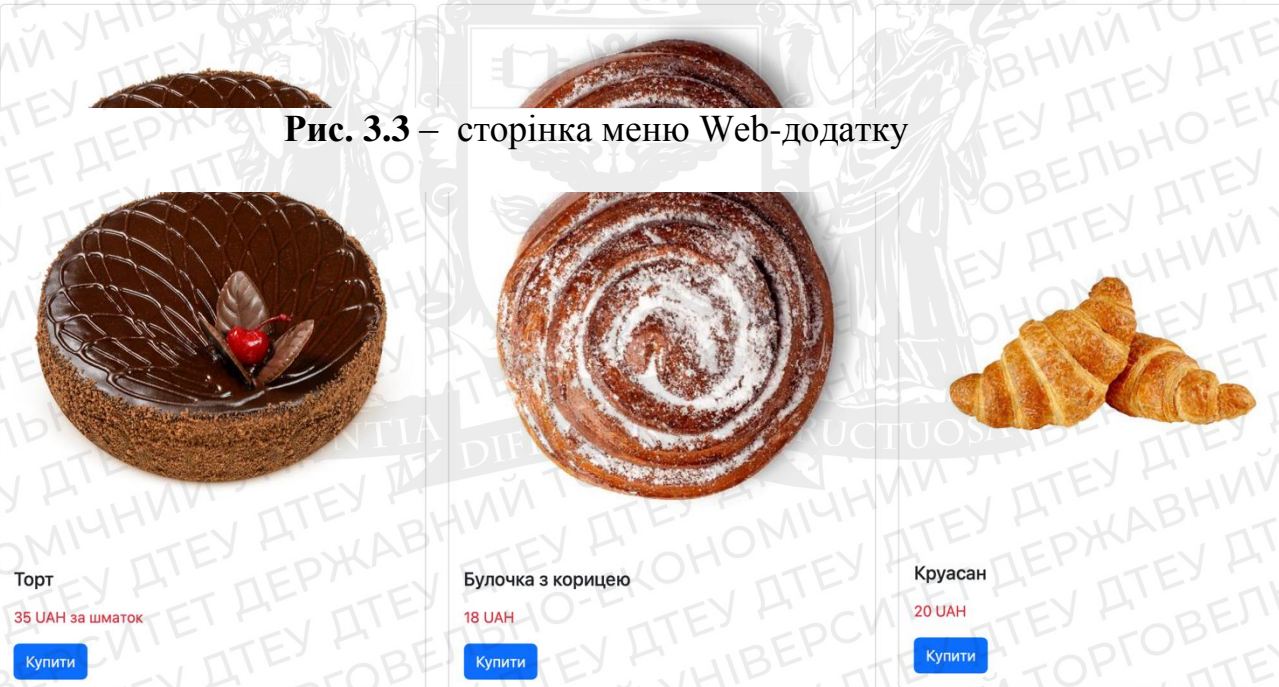


Піца
20 UAH за шматок
Купити

Хот дог
30 UAH
Купити

Пиріжок з вишнею
15 UAH
Купити

Рис. 3.3 – сторінка меню Web-додатку



Торт
35 UAH за шматок
Купити

Булочка з корицею
18 UAH
Купити

Круасан
20 UAH
Купити

Кошик

- Хот дог - 3 x 30 UAH
- Пиріжок з вишнею - 1 x 15 UAH
- Піца - 2 x 20 UAH
- Булочка з корицею - 1 x 18 UAH

Загалом: 163 UAH

Оформити замовлення

Рис. 3.4 – сторінка меню Web-додатку

3.2 Програмна реалізація

Для програмної реалізації додатку з підтримки діяльності продуктового магазину включає в себе використання різних технологій та мов користування. Мною використовувалися такі як:

- Front-end технології: Для розробки користувацького інтерфейсу та клієнтської частини додатку використовуються HTML, CSS та JavaScript. HTML відповідає за структуру сторінок, CSS - за оформлення та стилізацію елементів, а JavaScript - за логіку та взаємодію з користувачем [15].
- Back-end технології: Для обробки бізнес-логіки, взаємодії з базою даних та надання даних клієнту використовуються різні мови програмування та фреймворки, для створення додатку використовувалася Java.

Саме ці мови використовувалися через те, що HTML є основною мовою розмітки для створення веб-сторінок [19]. Використання HTML для створення веб-додатку з підтримки діяльності продуктового магазину має кілька переваг:

- Структура сторінки: HTML дозволяє структурувати веб-сторінку за допомогою різних елементів, таких як заголовки, абзаци, списки, таблиці тощо. Це дозволяє логічно організувати контент на сторінці підтримки, забезпечуючи зрозумілість та легкість навігації для користувачів.
- Форми зв'язку: HTML надає можливість створювати форми, за допомогою яких користувачі можуть зв'язатися зі службою підтримки. Форми дозволяють збирати відвідувачів інформацію, таку як ім'я, контактні дані та повідомлення, що допомагає у вирішенні їх питань або проблем.

- Розмітка FAQ: HTML дозволяє створювати структуровані списки та вкладені елементи, які можна використовувати для розмітки часто задаваних питань (FAQ). Це дозволяє чітко виділити запитання та їх відповіді, спрощуючи сприйняття інформації користувачами.
- Інтеграція з CSS та JavaScript: HTML дозволяє використовувати CSS для оформлення та стилізації веб-сторінок, забезпечуючи привабливий та зручний дизайн для користувачів. Також, за допомогою JavaScript, можна реалізувати додаткові функціональні можливості, такі як валідація форм, динамічне оновлення контенту тощо.

Загалом, використання HTML для створення веб-додатку з підтримки продуктового магазину дозволяє створити структуровану та зручну веб-додаток, який надає користувачам доступ до необхідної інформації та можливість зв'язатися зі службою підтримки.

Тоді як Java є мовою програмування, яка широко використовується для розробки веб-додатків. Використання Java для створення веб-додатку з підтримки діяльності продуктового магазину має кілька переваг:

- Масштабованість: Java є мовою, яка добре підходить для великих та складних веб-додатків. Вона має потужну підтримку для масштабування, що дозволяє легко розширювати функціональність додатку і обробляти великі обсяги даних[21].
- Безпека: Java відома своєю високою рівнем безпеки. Вона має вбудовану систему безпеки, яка дозволяє захистити веб-додаток від різних типів атак, таких як внедрення коду, перехоплення сесії тощо. Це особливо важливо для продуктивних магазинів, які операцій з фінансовою інформацією користувачів.
- Багатопотоковість: Java має вбудовану підтримку багатопотоковості, що дозволяє одночасно обробляти багато запитів веб-додатку. Це

особливо корисно для продуктових магазинів, які мають велику кількість користувачів та одночасних запитів.

- **Наявність багатих фреймворків:** У Java є широкий вибір веб-фреймворків, таких як Spring, JavaServer Faces (JSF), Play Framework тощо. Ці фреймворки надають зручні інструменти для розробки веб-додатків, дозволяючи швидко реалізувати функціональність, роботу з базами даних, безпеку та інші аспекти веб-програмування[21].
- **Переносимість:** Java є переносимою мовою програмування, що означає, що веб-додатки, написані на Java, можуть працювати на різних платформах і операційних системах без необхідності переписування коду.

Загалом, використання Java для створення веб-додатку з підтримкою діяльності продуктового магазину забезпечує надійність, безпеку та масштабованість додатку, що є важливими факторами у такому бізнесі.

Перейдемо до JavaScript є однією з основних мов програмування, яка використовується для створення інтерактивних веб-додатків.

Використання JavaScript для створення веб-додатку з підтримкою діяльності продуктового магазину має кілька переваг[20]:

- **Взаємодія з користувачем:** JavaScript дозволяє створювати динамічні та інтерактивні веб-сторінки, що дає змогу користувачам взаємодіяти з продуктовим магазином, додавати товари до кошика, виконувати пошук, фільтрацію, сортування тощо без перезавантаження сторінки.
- **Валідація та перевірка даних:** JavaScript може бути використаний для валідації форм, які користувачі використовують для оформлення замовлень. Він може перевіряти правильність введених даних, таких як ім'я, адреса електронної пошти, телефонний номер тощо, перед відправленням їх на сервер[20].

- Кошик покупок: JavaScript може бути використаний для реалізації функціональності кошика покупок. Він дозволяє додавати товари до кошика, змінювати їх кількість, обраховувати загальну вартість замовлення, видаляти товари з кошика та здійснювати розрахунок.
- Асинхронні запити: JavaScript дозволяє виконувати асинхронні запити до сервера без перезавантаження сторінки. Це особливо корисно для отримання даних з бази даних, наприклад, для відображення списку товарів, оновлення цін, перевірки наявності тощо.
- Взаємодія з API: JavaScript може взаємодіяти з різними API, такими як платіжні шлюзи, системи доставки, соціальні мережі тощо. Це дозволяє реалізувати оплату замовлення, відстеження стану доставки, обмін даними з іншими платформами та сервісами.
- Анімація та візуалізація: JavaScript надає можливості для створення анімацій, слайдерів, графіків та інших візуальних ефектів, що можуть покращити користувацький досвід і зробити продуктової магазин більш привабливим.

Загалом, використання JavaScript дозволяє створити багатофункціональний та інтерактивний веб-додаток з підтримкою діяльності продуктового магазину, що забезпечує зручний та ефективний спосіб покупки продуктів онлайн.

Наостанок CSS (Cascading Style Sheets) використовується для стилізації і оформлення веб-сторінок. Використання CSS для створення веб-додатку з підтримкою діяльності продуктового магазину має кілька переваг[22]:

- Дизайн та вигляд: CSS дозволяє задавати зовнішній вигляд елементів веб-сторінки, таких як кольори, шрифти, розміри, межі, фони тощо. Завдяки CSS ви можете створити привабливий та

професійний вигляд продуктового магазину, що сприяє приверненню уваги користувачів та покращує їх враження.

- Респонсивний дизайн: CSS дозволяє створювати респонсивний дизайн, що адаптується до різних розмірів екранів та пристроїв. Це особливо важливо для продуктових магазинів, оскільки користувачі можуть відвідувати їх з різних пристроїв, включаючи комп'ютери, планшети та смартфони. CSS дозволяє створювати адаптивний дизайн, що забезпечує зручну та приємну навігацію для користувачів незалежно від їхнього пристрою[22].
 - Макет та розташування: CSS дозволяє контролювати макет та розташування елементів на сторінці. Ви можете використовувати CSS для створення колонок, сіток, ґридів, а також для позиціонування елементів на сторінці. Це дозволяє організувати вміст продуктового магазину зручним та логічним способом.
 - Анімація та переходи: CSS надає можливості для створення анімацій, переходів та інших візуальних ефектів. Ви можете застосовувати анімацію до елементів, таких як кнопки, слайдери, зображення, що додає живості та динаміки до продуктового магазину.
 - Структурований код: Використання CSS дозволяє відокремити оформлення від HTML-коду. Це дозволяє створювати більш структурований та підтримуваний код. Ви можете використовувати CSS-класи та ідентифікатори, щоб застосовувати стилі до багатьох елементів одночасно та легко змінювати їхні стилі в одному місці.
- Загалом, використання CSS для створення веб-додатку з підтримкою діяльності продуктового магазину дозволяє створити привабливий та функціональний дизайн, забезпечити адаптивність для різних пристроїв, контролювати розташування елементів та додавати візуальні ефекти, що покращують користувацький досвід та допомагають створити

професійний вигляд продуктового магазину. Саме тому для створення додатку використовувалися усі ці мови.

Головна сторінка, дає можливість переглянути картинки-слайдери, які ілюструють тематику магазину. Також на сторінці присутня навігаційна панель, яка дає змогу перейти на інші сторінки додатку, це і є головною задачею цієї сторінки. Головна сторінка створена за допомогою мови програмування «HTML». За зовнішній вигляд відповідає CSS файл, що прив'язаний до цієї сторінки.

Описані вище елементи можна переглянути на рисунку 3.5

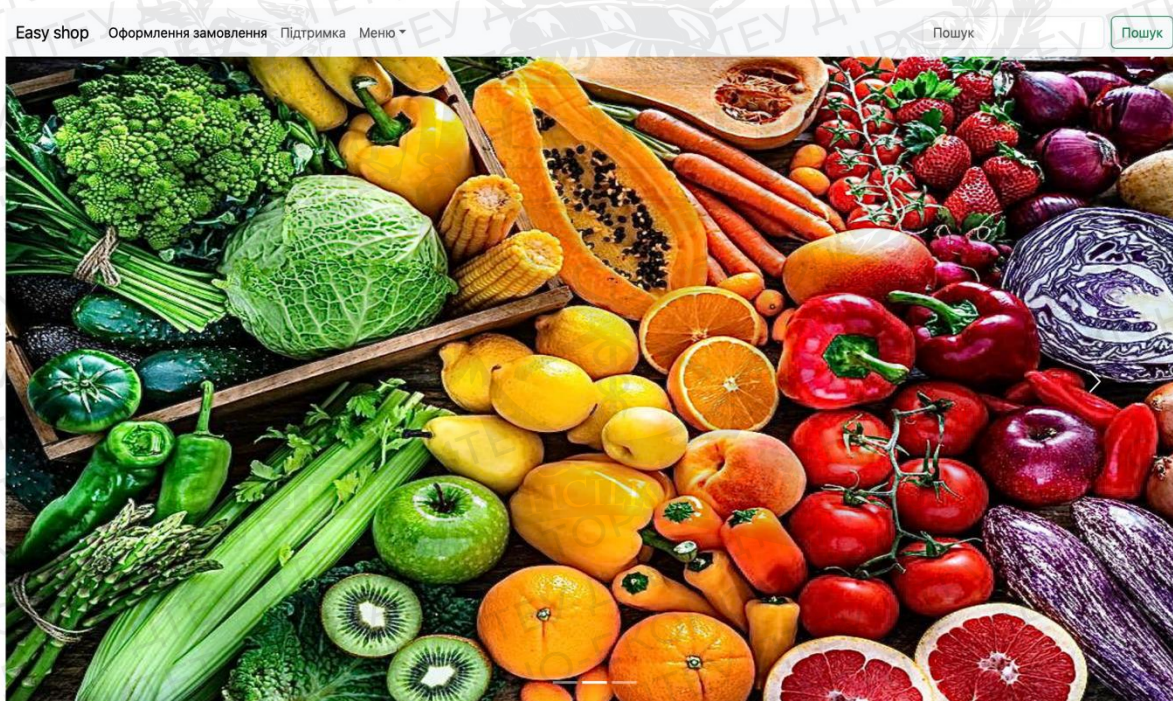


Рис. 3.5 – Головна сторінка Web-додатку

Навігаційне меню, за допомогою якого користувач може переміщатися по сайту (рис 3.6) було створено за допомогою мови HTML, та файлу CSS, що прив'язаний до нього і відповідає за зовнішній вигляд. Меню закріплено вгорі екрану і розташоване на усіх сторінках додатку. Меню досить просте і зрозуміле, для будь якого користувача, але при цьому виконує увесь потрібний функціонал. Також кнопка меню, містить в собі список, що розкривається, (рис 3.7) що зручно і функціонально водночас,

при цьому візуально не захаращуємо навігаційну панель. Кожна кнопка панелі веде користувача на бажану сторінку.

Дизайн навігаційної панелі є лаконічним і сучасним, виконаний у світлих тонах. Розташування навігаційної панелі було вирішено зробити у лівому блоці сторінки та зафіксувати на усіх сторінках. Це дозволить користувачу просто та інтуїтивно його використовувати. Дизайн обраний лаконічний і сучасний, з переважним використанням світлих кольорів. Кожен пункт навігаційної панелі містить посилання на відповідну сторінку.



Рис. 3.6 – Навігаційна панель



Рис. 3.7 – Навігаційна панель (список, що розкривається)

На сторінці меню, користувач може обрати товар за відповідною категорією. Головним функціоналом цієї сторінки є додавання товарів до кошика, це можна зробити за клікнувши по кнопці «купити» та виведення загальної ціни товару. За допомогою HTML був створений порядок усіх цих елементів на сторінці. Бекенд логіка була створена за допомогою «JavaScript», за зовнішній вигляд відповідає файл CSS, що прикріплений до цієї сторінки.

Розташування товарів обране по 3 товари в ряд, аби користувач зміг добре розглянути товар. Дизайн виконаний у переважно світлих кольорах, але ціна спеціально виділена червоним кольором, аби покупець одразу бачив її. Кнопки купити та оформити замовлення виділені синім кольором

і заокругленої форми, задля їх візуального виділення, також при наведенні на кнопки з'являється тінь.

Зображення сторінки меню наведено на рисунках 3.8-3.10

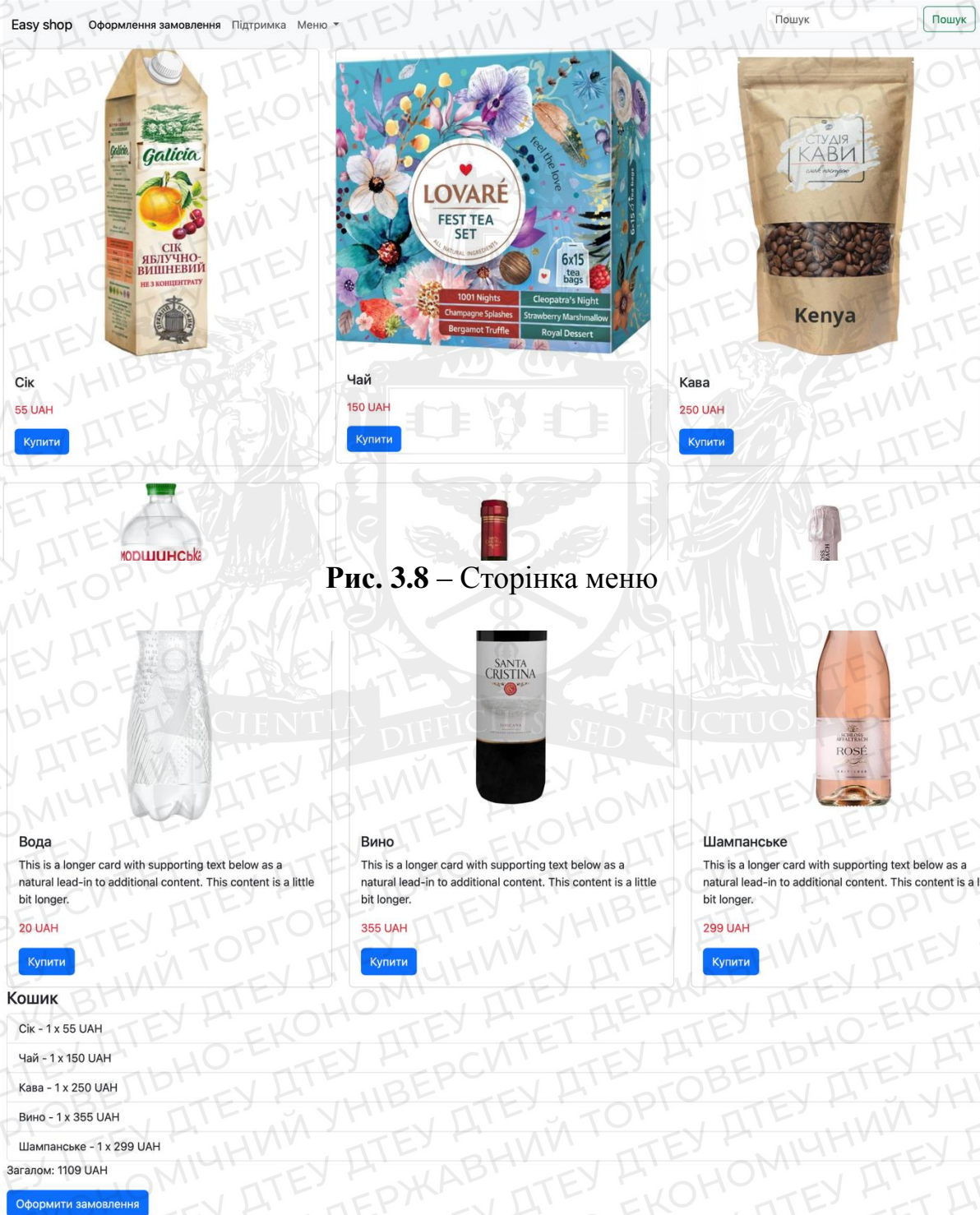


Рис. 3.8 – Сторінка меню

Рис. 3.9 – Сторінка меню з кошиком



Рис. 3.10 – Змінення кнопки купити

Кошик за його реалізацію відповідає мова програмування JavaScript і містить функції для відображення списку товарів у ньому, розрахунку загальної вартості товарів та додавання товарів до кошика. За зовнішній вигляд відповідає CSS файл. Код спочатку вибирає відповідні елементи з HTML-документа, включаючи сам кошик, список товарів, загальну вартість товарів у кошику та кнопку «оформити замовлення». Також, якщо кошик порожній і ви натискаєте «оформити замовлення», то виводиться вікно з надписом «Кошик порожній» (рис 3.11), якщо товар є у кошику, то виводиться напис «Для продовження замовлення натиснуть закрити!» (рис. 3.12) і виводиться форма для оформлення замовлення.



Рис. 3.11 – вікно з надписом «Кошик порожній»

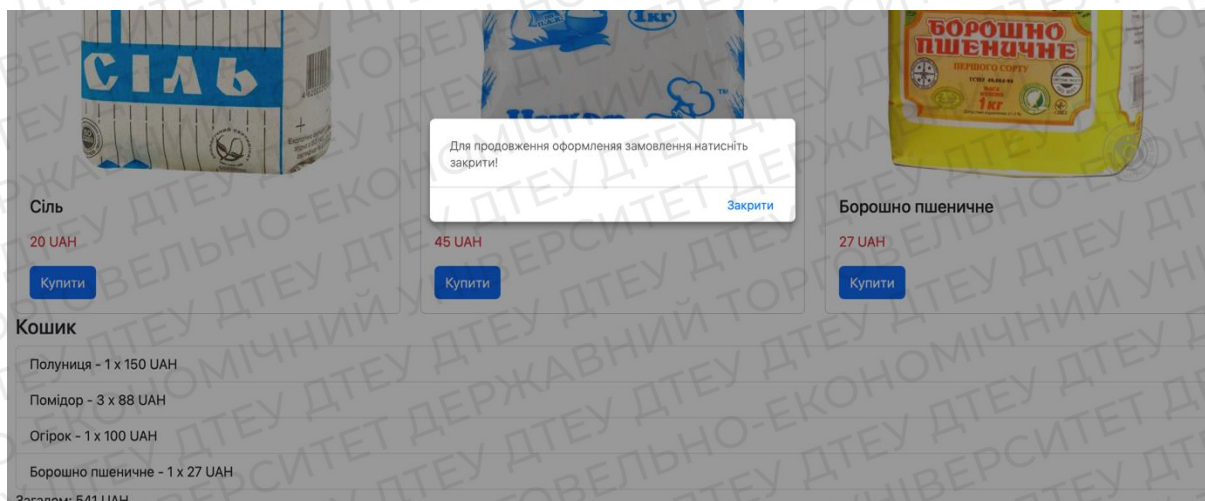


Рис. 3.12 – напис «Для продовження замовлення натиснуть закрити!»

Кошик

Сіль - 1 x 20 UAH

Помідор - 2 x 88 UAH

Огірок - 1 x 100 UAH

Полуниця - 1 x 150 UAH

Цукор - 1 x 45 UAH

Борошно пшеничне - 1 x 27 UAH

Загалом: 518 UAH

[Оформити замовлення](#)

Рис. 3.13 – Кошик

Сторінка оформлення замовлення представляє собою код написаний в HTML, який є формою замовлення, яка містить поля для введення даних про замовника (ім'я та прізвище, номер телефону, адресу доставки, вибір способу оплати і кнопку для відправлення замовлення). Для кожного поля встановлено атрибути «required», що забезпечують обов'язковість заповнення усіх полів перед відправленням форми (рис 3.14).

Для вибору способу оплати (рис 3.15) використовується випадючих список, у якому можна вибрати один з двох варіантів оплати: кредитною картою або готівкою при отриманні. Кнопка оформити замовлення має

тип «submit», що дозволяє відправляти дані форми на сервер при її натисканні. Сторінка оформлення замовлення виконана у світлих тонах.

Ім'я та прізвище:

Телефон:

Заповніть це поле

Email:



Адреса доставки:

Франка 5

Спосіб оплати:

Готівкою при отриманні



Рис. 3.14 – Обов'язковість заповнення усіх полів

Ім'я та прізвище: **Єлизавета**

Телефон: **0939764856**

Email:

Адреса доставки: **Франка 5**

Спосіб оплати Кредитною картою
 Готівкою при отриманні

Оформити замовлення

Рис. 3.15 – Варіанти оплати

Easy shop [Оформлення замовлення](#) [Підтримка](#) [Меню](#) [Disabled](#)

Заповніть це поле

Ім'я та прізвище:

Телефон:

Email:

Адреса доставки:

Спосіб оплати: Кредитною картою

Рис. 3.16 – Сторінка оформлення замовлення

Сторінка підтримки створена за допомогою HTML коду, та має посилання на CSS файл, що відповідає за зовнішній вигляд сторінки.

Сторінка складається з двох секцій. У першій секції міститься контактна інформація для підтримки клієнтів, така як номер телефону, електронна пошта та чат. У другій секції міститься список найчастіших питань (FAQ) та відповіді на них, що стосуються оплати, часу доставки і повернення товару. Також код містить розділи заголовку та підвалу з назвою магазину та інформацією про авторські права (рис. 3.17). Дизайн виконаний у світлих тонах.

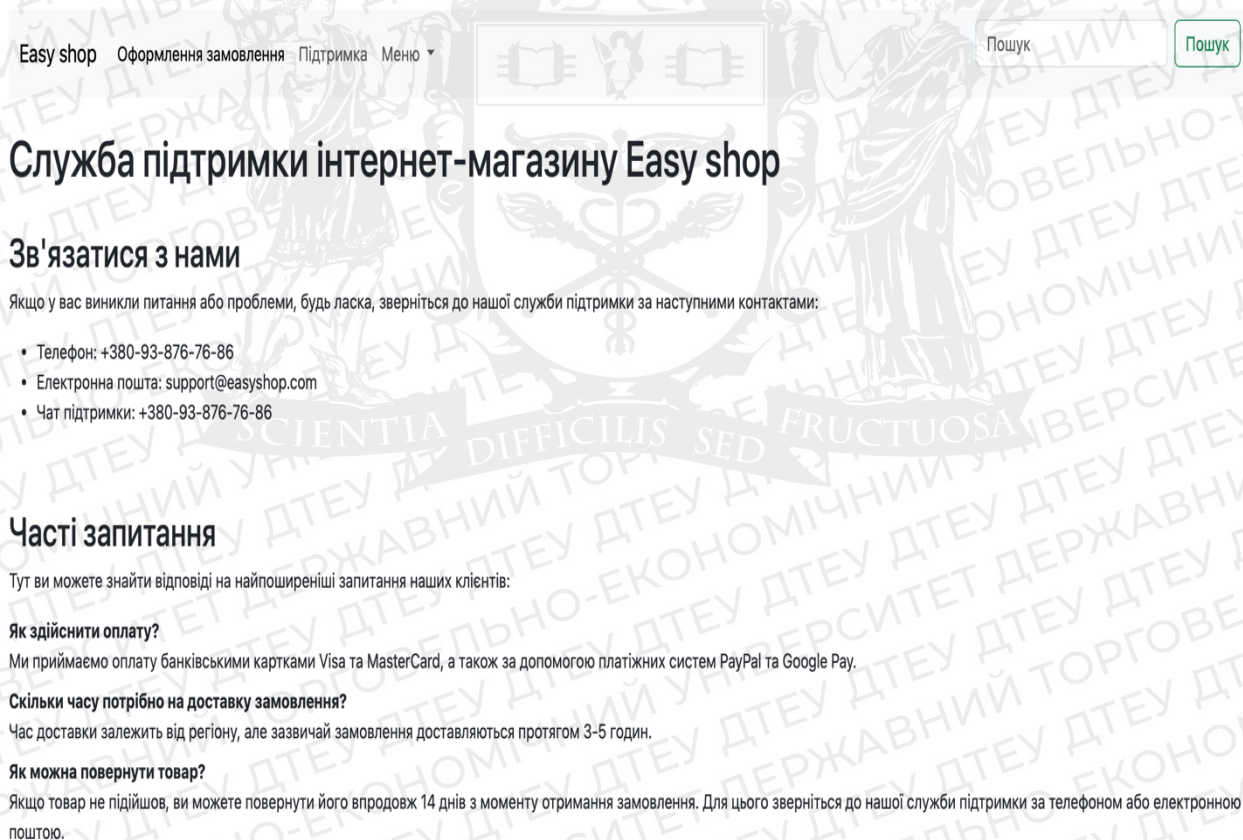


Рис. 3.17 – Сторінка підтримки

3.3 Використання програмного додатку

Головною метою створення web-додатку це можливість замовлення товару онлайн. З моменту коли користувач зайшов у додаток, до того моменту, коли він побачить сторінку оформлення замовлення. Користувач заходить на головну сторінку та бачить перед собою навігаційну панель, де він може обрати перелік товарів, додати їх до кошика, оформити замовлення і зв'язатися з службою підтримки. Ілюстрація зображена на рисунку 3.18.

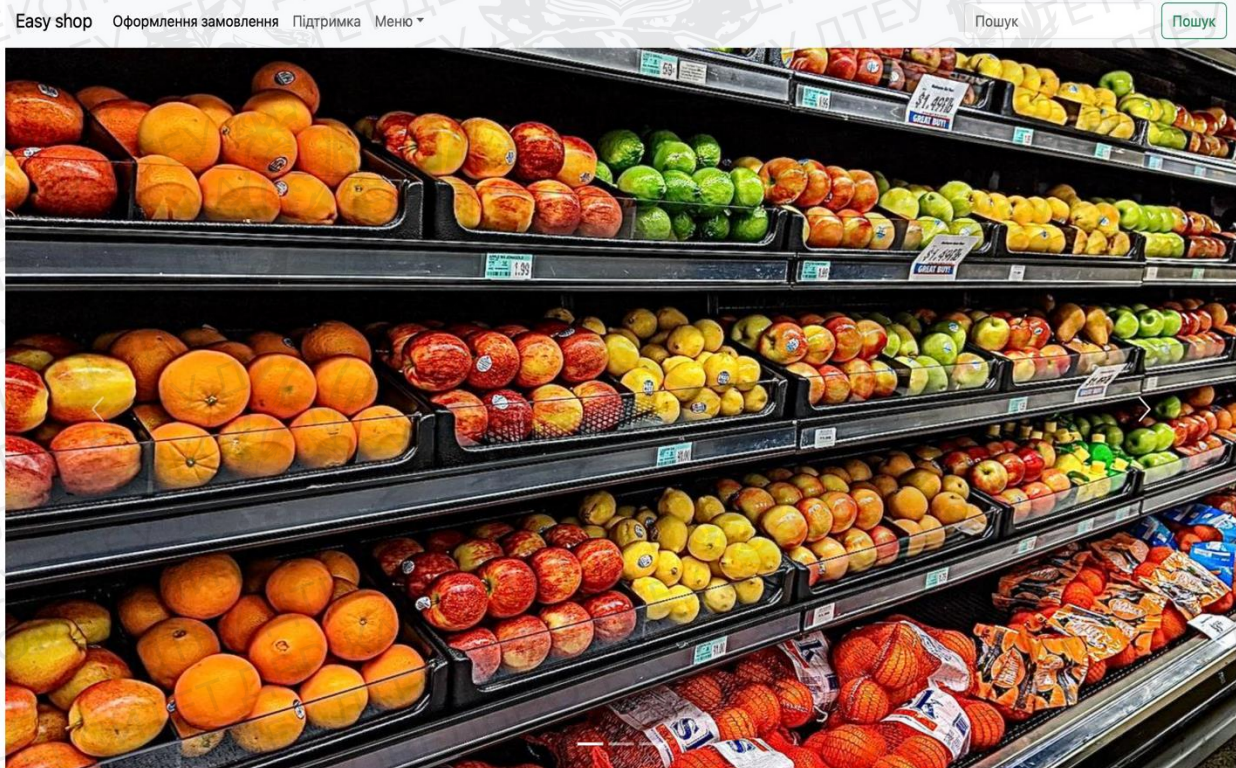


Рис. 3.18 – Головна сторінка

Потім користувач може обрати, яка сама категорія товарів його цікавить і додати їх до кошика. В кошику записується кількість товарів та їх загальна ціна зображено на рисунку 3.19.

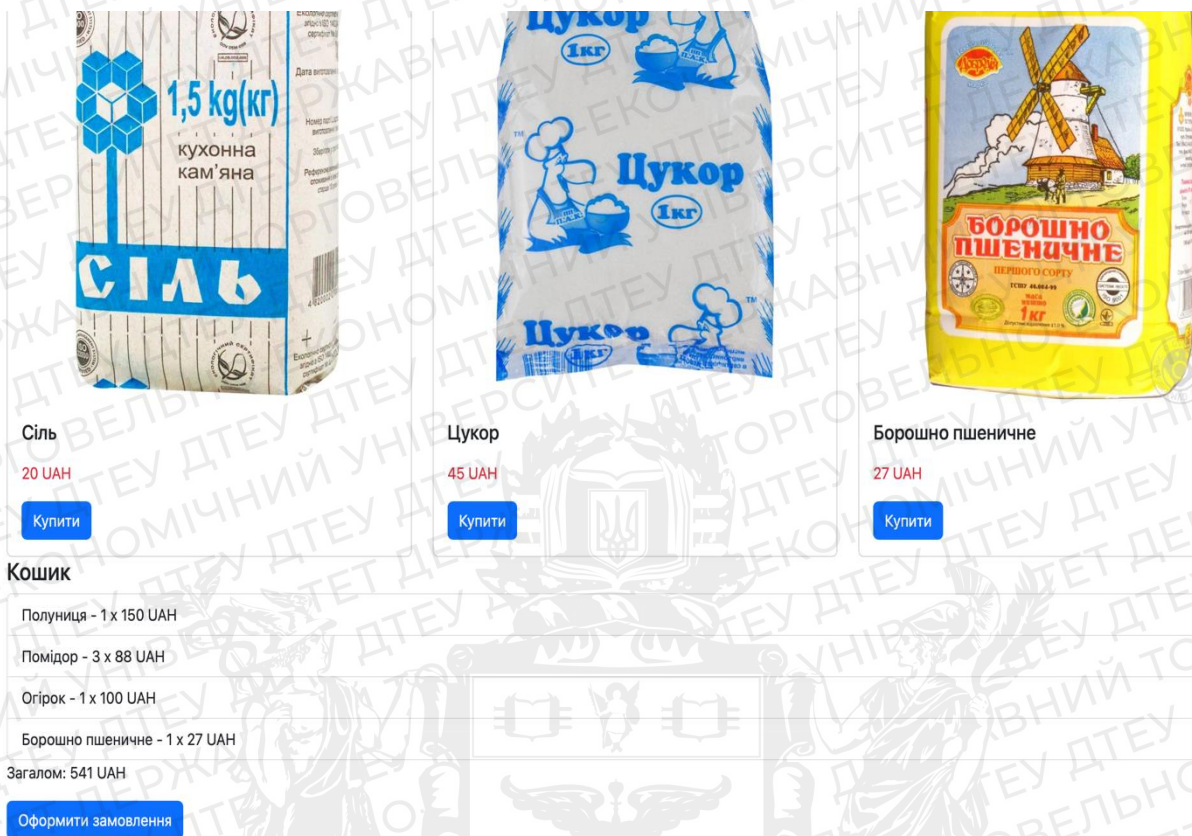


Рис. 3.19 – Сторінка меню та кошик

Далі користувач натискає «Оформити замовлення» і переходить на сторінку підтвердження замовлення (рис 3.20).

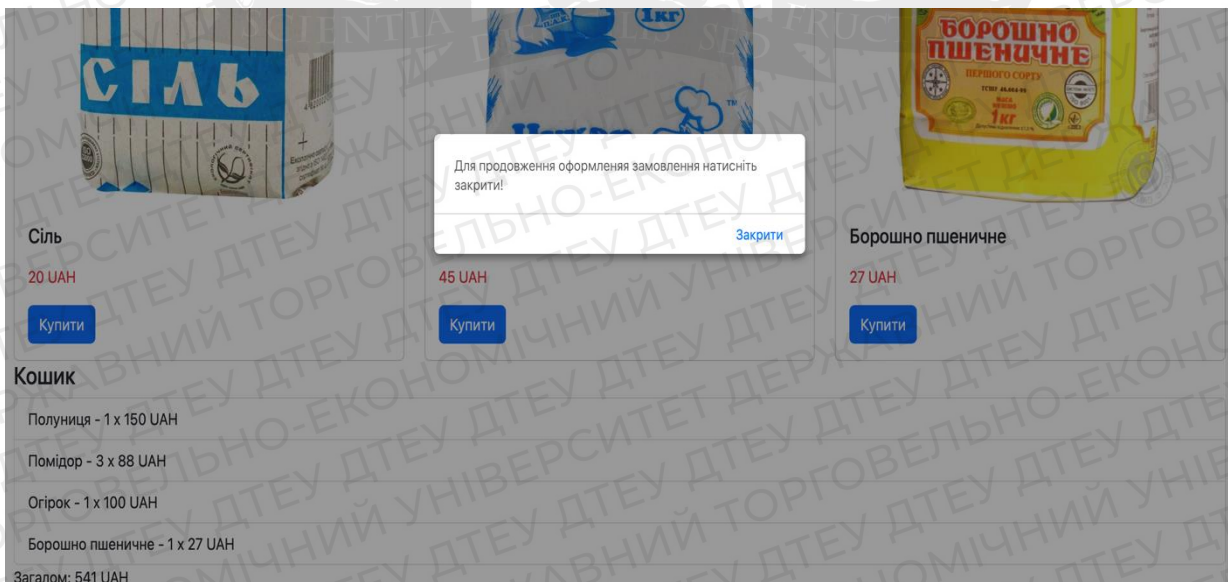


Рис. 3.20 – Оформлення замовлення

На цій сторінці користувач вводить усі необхідні данні для доставки товарів, а саме своє ім'я та прізвище, номер телефону, електронну пошту,

адресу для доставки та обирає спосіб оплати, після цього він натискає на кнопку «Оформити замовлення», якщо якісь данні заповнені не правильно або зовсім не заповнені, натиснути на кнопку не вийде поки все не буде заповнено правильно і лише після цього замовлення вдасться оформити зображено на рисунку 3.21.

Ім'я та прізвище: **Єлизавета**

Телефон: **0939764856**
Заповніть це поле

Email:

Адреса доставки: **Франка 5**

Спосіб оплати: **Готівкою при отриманні**

Оформити замовлення

Рис. 3.21 – Підтвердження оформлення замовлення

Далі користувач просто чекає на своє замовлення, або ж якщо в нього є якісь питання, то він може зайти на сторінку підтримки і подивитися відповіді на запитання, які найчастіше задають, якщо ж він не

знайшов там відповіді на своє запитання, то він може зв'язатися з магазином за допомогою декількох способів (рис 3.22).



Рис. 3.22 – Сторінку підтримки

Функціонал сайту досить простий та інтуїтивно зрозумілий, причому сайт вміщує в собі усі необхідні функції для його гармонічної і якісної роботи.

ВИСНОВКИ

В результаті виконання випускної кваліфікаційної роботи розроблено Web-додаток з підтримки діяльності продуктового магазину. У процесі роботи було проведено аналіз предметної області та визначений сучасний стан проектування інформаційних систем. Проведений аналіз вітчизняних аналогів, що допомогло визначити основні вимоги та характеристики для створення додатку.

Сформульовано мету та завдання для розробки проекту «Web-додаток з підтримки діяльності продуктового магазину». Визначені шляхи та засоби для розробки Web-додатку. Також було здійснено моделювання використання додатку, побудовані необхідні діаграми.

Програмна реалізація роботи здійснюється в останній частині роботи. Для цього було розроблена архітектура додатку з використанням макетів. Наведена програмна реалізація додатку та його використання з малюнками, детальним описом виконаної роботи та поясненнями.

Отже, результатом випускної кваліфікаційної роботи є Web-додаток, який відповідає всім поставленим вимогам і є конкурентоспроможним, додаток був створений для полегшення споживачами користування послугами, а саме замовлення продуктів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Web Application" - Wikipedia [Електронний ресурс]/ Режим доступу: (https://en.wikipedia.org/wiki/Web_application)
2. "What Are Web Applications?" - Lifewire [Електронний ресурс]/ Режим доступу: <https://www.lifewire.com/what-are-web-applications-2483330>
3. Mozilla Developer Network [Електронний ресурс]/ Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/Apps>
4. Microsoft Developer Network [Електронний ресурс]/ Режим доступу: <https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/build-a-single-page-application-spa-with-aspnet-web-api-and-angularjs>
5. TechCrunch [Електронний ресурс]/ Режим доступу: <https://techcrunch.com>
6. [Hoffer, J. A., George, J. F., & Valacich, J. S.] / Modern systems analysis and design. Pearson, 2020 [Електронний ресурс]/ Режим доступу: <http://www.mim.ac.mw/books/Modern%20Systems%20Analysis%20And%20Design.pdf>/ 44-78 p.
7. [Laudon, K. C., & Laudon, J. P.] / Management information systems: managing the digital firm, 2016. [Електронний ресурс]/ Режим доступу: https://repository.dinus.ac.id/docs/ajar/Kenneth_C.Laudon,Jane_P_.Laudon_-_Management_Information_System_13th_Edition_.pdf/ 32-55 p.
8. Kshetri, N./Blockchain's roles in meeting key supply chain management objectives, 2018.
9. Торгівельна площадка zakaz.ua: [Електронний ресурс]/ Режим доступу: <https://zakaz.ua/uk/>
10. Торгівельна площадка atbmarket: [Електронний ресурс]/ Режим доступу: <https://www.atbmarket.com/>
11. Глава 4. Діаграма варіантів використання (use case diagram): [Електронний ресурс]/ Режим доступу: <http://mmsa.kpi.ua/sites/default/files/disciplines/%D0%A0%D0%BE%D0%B7>

%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D1%96%20%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%B%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC/didkovska_m_v_testing_lecture_3.pdf

12. Інформаційне моделювання Idefi і моделювання даних в бізнес-процесах (Ideflx): [Електронний ресурс]/ Режим доступу:

<https://www.maxzosim.com/idef/>

13. Що таке інформаційні технології: [Електронний ресурс]/ Режим доступу: <http://apereps.kpi.ua/shcho-take-informatsiini-technologii>

14. Рекомендація: Діаграма послідовності: [Електронний ресурс]/ Режим доступу: <https://studfile.net/preview/9445425/page:116/> 116 с.

15. Infographic: HTML vs CSS?: [Електронний ресурс]/ Режим доступу: <https://www.codingdojo.com/blog/html-vs-css>

16. ОМПС. Діаграми діяльності: [Електронний ресурс]/ Режим доступу: https://studopedia.su/16_5481_omps-diagrami-diyalnosti.html/ 13 с.

17. UML для бізнес-моделювання: для чого потрібні діаграми процесів: [Електронний ресурс]/ Режим доступу:

<https://evergreens.com.ua/ua/articles/uml-diagrams.html>

18. What is UML? Everything You Need to Know About Unified Modeling Language: [Електронний ресурс]/ Режим доступу:

<https://www.gliffy.com/blog/what-is-uml-everything-you-need-to-know-about-unified-modeling-language#:~:text=Unified%20Modeling%20Language%2C%20or%20UML,rules%20specifically%20for%20drawing%20diagrams.>

19. HTML For Beginners The Easy Way: Start Learning HTML & CSS Today: [Електронний ресурс]/ Режим доступу:

<https://html.com/#ixzz81sSjbb00>

20. Why Is JavaScript So Popular?: [Електронний ресурс]/ Режим доступу:

<https://www.codepwr.com/why-is-javascript->

sopopular/#:~:text=Why%20is%20JavaScript%20so%20ubiquitous,necessary%20before%20executing%20the%20language.

21. What is Java?: [Електронний ресурс]/ Режим доступу:

[https://aws.amazon.com/what-](https://aws.amazon.com/what-is/java/#:~:text=%D0%AF%D0%B7%D1%8B%D0%BA%20Java%20%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D0%B5%D1%82%D1%81%D1%8F%20%D0%B4%D0%BB%D1%8F%20%D0%BC%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC%D0%BE%D0%B2,%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B2%20%D1%80%D0%B5%D0%B6%D0%B8%D0%BC%D0%B5%20%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%B8.)

[is/java/#:~:text=%D0%AF%D0%B7%D1%8B%D0%BA%20Java%20%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D0%B5%D1%82%D1%81%D1%8F%20%D0%B4%D0%BB%D1%8F%20%D0%BC%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC%D0%BE%D0%B2,%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B2%20%D1%80%D0%B5%D0%B6%D0%B8%D0%BC%D0%B5%20%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%B8.](https://aws.amazon.com/what-is/java/#:~:text=%D0%AF%D0%B7%D1%8B%D0%BA%20Java%20%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D0%B5%D1%82%D1%81%D1%8F%20%D0%B4%D0%BB%D1%8F%20%D0%BC%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC%D0%BE%D0%B2,%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B2%20%D1%80%D0%B5%D0%B6%D0%B8%D0%BC%D0%B5%20%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%B8.)

22. Css: [Електронний ресурс]/ Режим доступу:

[https://www.atinternet.com/en/glossary/css/#:~:text=CSS%20is%20the%20acronym%20of,are%20equally%20called%20CSS%20files%20\(](https://www.atinternet.com/en/glossary/css/#:~:text=CSS%20is%20the%20acronym%20of,are%20equally%20called%20CSS%20files%20()

