

**Міністерство освіти і науки України**  
**Національний університет «Запорізька політехніка»**

кафедра програмних засобів

**Самостійна робота**  
з дисципліни «Вебтехнології та вебдизайн»

Виконав:

ст.групи КНТ-113сп

Артем БЕДСЬКИХ

Прийняв:

ст.викладач

Сергій ЛЕОЩЕНКО

2023

## РЕФЕРАТ

КЛЮЧОВІ СЛОВА: React, Webpack, JavaScript, HTML, SCSS, Broadcast-Channel, JSON-Server, Вебтехнології, Вебдизайн, Сайт замовлення піци.

Об'єкт дослідження – Розробка веб-сайту для замовлення піци.

Предмет дослідження – Використання технологій React, Webpack, JavaScript, HTML, SCSS, Broadcast-Channel та JSON-Server у створенні веб-застосунку для замовлення піци.

Метою роботи є розробка та аналіз функціональності веб-сайту для зручного та ефективного замовлення піци в онлайн режимі.

Матеріали, методи та технічні засоби, що використовувалися під час роботи: Мова програмування JavaScript, фреймворк React, інструмент сборки Webpack, мови розмітки HTML та стилізації SCSS [8], технологія Broadcast-Channel для взаємодії між вкладками, та JSON-Server для емуляції серверної частини.

Результати. Розроблено функціональний веб-сайт для замовлення піци, який використовує сучасні технології та забезпечує зручність користувачів у виборі та замовленні піци через Інтернет.

Висновки. Використання технологій React, Webpack, та інших зазначених інструментів значно полегшує розробку веб-застосунків, а також сприяє їхній продуктивності та зручності для кінцевих користувачів. Розроблений веб-сайт може бути використаний у галузі ресторанного бізнесу для оптимізації процесу прийому та обробки замовлень на доставку або самовивіз піци.

Галузь використання – Розроблений веб-сайт може бути використаний у сфері ресторанного бізнесу для покращення сервісу замовлення та доставки піци.

## ЗМІСТ

	С.
Перелік скорочень та умовних позначок .....	4
Вступ.....	5
1 Аналіз предметної області .....	6
1.1 Опис теми (предметної області).....	6
1.2 Постановка завдання .....	7
1.2.1 Межі системи.....	7
1.2.2 Функціональність системи .....	7
1.2.3 Вимоги до інтерфейсу.....	8
1.2.4 Вимоги до продуктивності .....	8
1.2.5 Вимоги до безпеки .....	9
1.2.6 Експлуатаційні вимоги .....	9
1.2.7 Не функціональні вимоги до продукту .....	10
1.3 Висновки за розділом 1 .....	11
2 Матеріали і методи .....	12
2.1 Опис засобів розробки.....	12
2.1.1 Вибір мови програмування .....	12
2.1.2 Вибір середовища розробки.....	13
2.2 Структурна схема розробки.....	14
2.3 Висновки за розділом 2 .....	15
3 Основні рішення щодо реалізації компонентів системи .....	16
3.1 Проєктування дизайну застосунку.....	16
3.2 Розгортання серверу .....	20
3.3 Висновки за розділом 3 .....	21
4 Експлуатація, тестування та експериментальне дослідження програми.....	22
4.1 Призначення й умови застосування програми .....	22
4.2 Методика та результати тестування .....	22
4.2.1 Чек-лист тестування.....	22
4.2.2 Тестування за сценарієм .....	23

Висновки .....	24
Перелік джерел посилання .....	25

## **ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК**

ІТ – інформаційні технології;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

MVC – model-view-controller;

UML – Unified Modeling Language;

CRA – Create React App;

CMS - система управління вмістом;

VS Code - Visual Studio Code.

## ВСТУП

Актуальність використання веб-технологій у сучасному світі набуває все більшого значення, особливо в галузі комерції та обслуговування. Запровадження онлайн-сервісів і веб-додатків є необхідним етапом для вдосконалення бізнес-процесів та поліпшення взаємодії з клієнтами. У рамках дисципліни "Вебтехнології та вебдизайн" проведено самостійне дослідження та розробку веб-сайту, спрямованого на оптимізацію процесу замовлення піци в ресторані.

Об'єктом дослідження стала розробка веб-застосунку для замовлення піци, а предметом дослідження – використання сучасних технологій, зокрема React, Webpack, JavaScript, HTML [7], SCSS, Broadcast-Channel та JSON-Server, у створенні функціонального та зручного інструменту для клієнтів ресторану [1].

Метою роботи є створення ефективного веб-сайту, що дозволить клієнтам легко та зручно здійснювати замовлення піци через Інтернет, зменшуючи час та зусилля, необхідні для цього процесу.

Матеріали, методи та технічні засоби, що використовувалися включають в себе мову програмування JavaScript, фреймворк React [2] для створення користувацького інтерфейсу, Webpack для забезпечення оптимальної збірки та розгортання проекту [4], а також Broadcast-Channel [9] та JSON-Server для забезпечення сприятливої взаємодії між вкладками та емуляції серверної частини веб-застосунку.

Результати включають розробку функціонального та естетичного веб-сайту для замовлення піци, що використовує передові технології для полегшення та удосконалення процесу замовлення їжі.

У продовженні роботи буде викладено аналіз предметної області, обґрунтування вибору технічних засобів, опис реалізації компонентів системи та етапи експлуатації, тестування та експериментального дослідження розробленого веб-застосунку.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Опис теми (предметної області)

В сучасному суспільстві, що швидко розвивається, використання веб-технологій у галузі обслуговування та комерції стає невід'ємною частиною конкурентоспроможного бізнесу. Створення веб-сайтів для замовлення їжі, зокрема піци, набуває все більшого попиту серед споживачів, які цінують зручність та швидкість отримання послуг.

Тема даної роботи – "Розробка веб-сайту для замовлення піци" – є актуальною та важливою в контексті сучасних тенденцій у галузі харчового сервісу. Розуміння потреб та вимог користувачів, а також використання передових веб-технологій для покращення якості обслуговування, стає стратегічно важливим завданням для підприємств громадського харчування.

Веб-сайт для замовлення піци має на меті забезпечити користувачам можливість швидко та зручно обирати страви, визначати їхні характеристики та здійснювати онлайн-замовлення. Це дозволяє не лише оптимізувати роботу ресторану чи кафе, але і підвищує задоволеність клієнтів, які отримують доступ до широкого асортименту страв та можливість вибору за кілька кліків миші.

Такий веб-сайт може враховувати різні дієтичні обмеження, пропонувати акції та знижки, забезпечуючи користувачам не лише смачні, але і доступні страви. Важливою частиною розробки є також забезпечення зручності використання веб-інтерфейсу та можливості замовлення з різних пристроїв, що розширює аудиторію користувачів.

Таким чином, обґрунтування та розробка веб-сайту для замовлення піци є важливим завданням, спрямованим на підвищення конкурентоспроможності підприємств громадського харчування та задоволення зростаючих очікувань споживачів.

## **1.2 Постановка завдання**

Опис завдання. В рамках цього проекту ставиться завдання розробити веб-сайт для замовлення піци, який відповідає сучасним вимогам та технічним стандартам. Основна мета полягає в створенні зручного та ефективного інтерфейсу для користувачів, які мають бажання замовити піцу онлайн.

### **1.2.1 Межі системи**

Визначення меж системи в контексті веб-сайту для замовлення піци включає в себе обговорення функціонального та технічного обмеження проекту. Функціональні межі визначають обсяг операцій та сервісів, які пропонує система. В даному випадку це може охоплювати вибір та замовлення різних видів піц, сортування піц за деякими критеріями та здійснення додавання замовлення до кошику.

Технічні межі, з іншого боку, обмежують характеристики системи, такі як продуктивність, безпека та підтримка різних платформ. Наприклад, система повинна забезпечувати ефективну обробку та відображення великої кількості продуктів та замовлень одночасно.

### **1.2.2 Функціональність системи**

Функціональні вимоги до системи:

- перегляд всіх позицій;
- фільтр піц за їх видом;
- сортування за критеріями;
- реалізація кошика;
- взаємодія між вкладками.



### **1.2.3 Вимоги до інтерфейсу**

Важливою складовою розробки веб-сайту для замовлення піци є належно визначені та виконані вимоги до інтерфейсу. Цей розділ ставить перед собою завдання визначити, як користувачі будуть взаємодіяти з системою, щоб здійснювати замовлення максимально зручно та ефективно.

- Інтерактивний інтерфейс: Забезпечити інтуїтивно зрозумілий та легкий у використанні інтерфейс, що дозволяє користувачам легко зорієнтуватися та взаємодіяти з різними функціями сайту.

- Меню та категорії: Створити чітке та структуроване меню, яке дозволяє користувачам швидко переглядати різні види піц, їхні складові та інші додаткові опції.

- Адаптація до різних пристроїв: Забезпечити адаптивний дизайн для оптимального відображення на різних пристроях, зокрема на комп'ютерах, планшетах і смартфонах.

- Привабливий та сучасний дизайн: Створити естетично приємний та сучасний дизайн, який відповідає тематиці ресторану та сприяє позитивному враженню від взаємодії з сайтом.

### **1.2.4 Вимоги до продуктивності**

У цьому розділі визначаються основні параметри та очікувані рівні продуктивності системи для оптимального користувацького досвіду та ефективності в роботі.

- Швидкодія: Встановити вимоги до часу відповіді системи на користувацькі запити, забезпечуючи при цьому швидку інтерактивність веб-інтерфейсу. Мінімізувати час завантаження сторінок та обробки даних.

- Оптимізація ресурсів: Забезпечити ефективне використання серверних та клієнтських ресурсів. Це включає в себе оптимізацію роботи бази

даних, ефективне управління зображеннями та інші ресурсозберігаючі практики.

– Масштабованість: Розробити систему так, щоб вона могла ефективно масштабуватися для обробки збільшеної кількості користувачів та замовлень.

### **1.2.5 Вимоги до безпеки**

Вимоги до надійності програмного продукту наступні:

- програма повинна не давати можливість користувачеві вводити дані в базу даних;
- програма повинна конкретно зберігати інформацію;
- програма повинна давати можливість правильно відображати інформацію на декількох вкладках;
- програма повинна давати можливість видаляти позиції замовлення;
- програма повинна надати користувачу обрати за яким критерієм робити сортування піц.

### **1.2.6 Експлуатаційні вимоги**

Даний розділ визначає умови ефективного та стабільного використання системи, забезпечуючи коректну роботу та зручний досвід для кінцевого користувача.

– Оптимізація продуктивності: Враховувати оптимізацію продуктивності системи, забезпечуючи швидку відповідь на запити користувачів та мінімальний час завантаження сторінок.

– Сумісність із різними браузерами: Забезпечити сумісність системи із різними веб-браузерами, зокрема Chrome, Firefox, Safari та Edge [1]. Використовувати стандартизовані технології та методи для забезпечення коректного відображення та функціонування на різних платформах.

- Регулярні оновлення та покращення: Передбачити систему регулярних оновлень для вдосконалення функціональності та усунення можливих помилок. Забезпечити користувачам доступ до оновлень та повідомлення про нововведення.

Цей підрозділ визначає ключові аспекти експлуатації системи та забезпечує її ефективну та стабільну роботу.

### **1.2.7 Не функціональні вимоги до продукту**

Даний підрозділ визначає нефункціональні вимоги, які визначають параметри та властивості системи, не пов'язані з конкретними функціональними можливостями, але впливають на загальну ефективність та якість продукту.

- Продуктивність: Забезпечити високу продуктивність системи, зокрема, швидкість відгуку на користувацькі запити та оптимальний час завантаження сторінок. Мінімізувати час відповіді сервера та оптимізувати роботу бази даних для ефективного використання ресурсів.

- Складність інтерфейсу: Забезпечити зручний та інтуїтивно зрозумілий інтерфейс для користувачів, мінімізуючи кількість кроків для виконання конкретних завдань.

- Надійність та стабільність: Забезпечити високу стабільність та надійність системи, мінімізуючи вірогідність виникнення помилок та аварій.

- Сумісність: Забезпечити сумісність системи із різними пристроями та платформами, включаючи роботу на різних веб-браузерах та операційних системах.

Цей підрозділ визначає параметри, які визначають загальну якість та ефективність продукту з точки зору його нефункціональних характеристик.

### **1.3 Висновки за розділом 1**

У підсумку аналізу предметної області і постановки завдання, можна зробити кілька ключових висновків. Перш за все, розробка веб-застосунку для замовлення піци вимагає ретельного вивчення потреб користувачів і врахування специфіки галузі.

Межі системи чітко визначені, враховуючи основні функції та вимоги до інтерфейсу. Функціональність системи орієнтована на зручне та ефективне замовлення піци, а також надання інформації про асортимент.

Вимоги до продуктивності та безпеки визначені з урахуванням потреб бізнесу та побажань користувачів. Експлуатаційні вимоги забезпечують надійність та ефективність роботи системи в умовах реального використання.

Нефункціональні вимоги до продукту визначають його якість та зручність використання. Розглянуті аспекти дозволяють визначити базові параметри, які будуть слугувати основою для подальшої розробки.

В цілому, аналіз предметної області і формулювання завдань визначили ключові аспекти, які слід враховувати при розробці веб-застосунку для замовлення піци, а також визначили основні напрямки роботи на наступних етапах розробки.

## 2 МАТЕРІАЛИ І МЕТОДИ

### 2.1 Опис засобів розробки

#### 2.1.1 Вибір мови програмування

Для реалізації клієнтської частини застосунку використовувався JavaScript, зокрема був обраний фреймворк React. Використання React дозволило створити динамічний та високопродуктивний інтерфейс, спростивши розробку компонентів та їх взаємодію [2].

Проведемо порівняння мов програмування для реалізації вебзастосунків, результати порівняння представимо в табличній формі (табл. 2.1). У якості критеріїв будемо використовувати наступні:

- наявність надбудов та комплексів для спрощення розробки – загальна оцінка кількості фреймворків або систем керування вмістом (CMS) для спрощення розробки;
- підтримка технології серед серверів – підтримка технології розроблених застосунків серед серверів для простоти розгортання та підтримки;
- можливості масштабованості застосунків – оцінка подальшої простоти масштабованості застосунків;
- кросплатформність розробки та застосунків – кросплатформність розроблених застосунків та її рівень.

Таблиця 2.1 – Порівняння мов програмування

Критерій	JavaScript (Мова #1)	Python (Мова #2)
Наявність надбудов та комплексів для спрощення розробки	Так	Так
Підтримка технології серед серверів	Так	Так
Можливості масштабованості застосунків	Задовільні	Високі

Кросплатформність розробки та застосунків	Так	Так
---	-----	-----

З таблиці видно, що обидві мови програмування, JavaScript [5] та Python, мають свої переваги та відповідають критеріям для розробки вебзастосунків. JavaScript є популярним вибором для клієнтської сторони завдань, завдяки широкому спектру фреймворків та бібліотек, які спрощують розробку веб-інтерфейсів. З іншого боку, Python відзначається високою масштабованістю на серверній стороні та має велику кількість інструментів для швидкої розробки.

### 2.1.2 Вибір середовища розробки

Одним із ключових етапів у розробці вебзастосунків є вибір відповідного середовища розробки. У даному випадку, для створення веб-сайту для замовлення піци, використовувалось середовище Visual Studio Code (VS Code). Це інтегроване середовище розробки здатне забезпечити ефективну підтримку мови програмування JavaScript та фреймворку React.

Основні переваги використання VS Code включають:

- Легкість використання: VS Code є легким та інтуїтивно зрозумілим середовищем, що спрощує розробку та полегшує освоєння для новачків.
- Підтримка JavaScript та React: VS Code забезпечує потужний набір інструментів для розробки на мові JavaScript і фреймворку React, включаючи підсвічування синтаксису, автодоповнення та інші корисні функції.
- Розширюваність: Середовище розробки відкрите для розширень, що дозволяє додавати необхідні плагіни для поліпшення функціоналу та відповідності конкретним потребам проекту.
- Інтеграція з Git: VS Code має вбудовану інтеграцію з системою керування версіями Git, що робить спільну роботу над проектом більш зручною та ефективною [3].

Обраний інструментарій відображає виважений вибір для забезпечення продуктивності та зручності у розробці вебзастосунків з використанням React та JavaScript.

## 2.2 Структурна схема розробки

Загальна схема системи за обраною темою на рис. 2.1.

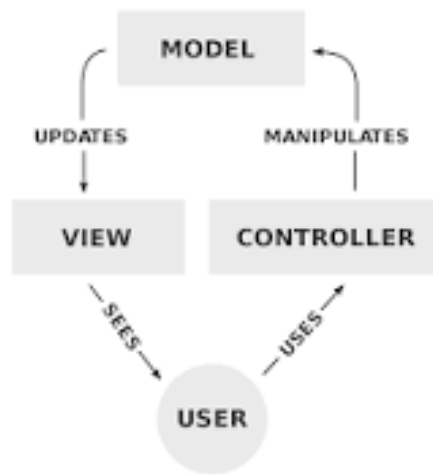


Рисунок 2.1 – Загальна схема системи за обраною темою

Архітектура вебзастосунку базується на моделі Модель-вид-контролер (MVC) – рис. 2.2.



Рисунок 2.2 – Архітектура вебзастосунку

### **2.3 Висновки за розділом 2**

На підставі аналізу засобів розробки, вибраних для створення вебзастосунку замовлення піци, можна зробити декілька ключових висновків. Обрана мова програмування та фреймворк відповідають вимогам проекту і враховують його особливості.

Важливим аспектом є те, що обрана архітектура системи (зокрема, модель Модель-вид-контролер) надає чітку структуру для подальшого розвитку. Це сприяє полегшенню розробки та обслуговування проекту, забезпечуючи високий рівень модульності та розширюваності.

Окрім того, порівняльний аналіз мов програмування дозволяє обрати оптимальний інструментарій для подальшої роботи над проектом. Враховуючи критерії, такі як наявність надбудов, підтримка технології серед серверів, можливості масштабованості та кросплатформність, можна зробити обґрунтований вибір, що сприятиме успішному розвитку проекту в майбутньому.



### 3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

#### 3.1 Проектування дизайну застосунку

Проектування дизайну веб-сайту для замовлення піци є ключовим етапом у створенні зручного та привабливого інтерфейсу для користувачів. Для візуалізації та узгодження замірів розроблялися 3 вайрфрейми, 2 мокапи та 1 прототип для головної сторінки.

Вайрфрейми зображені на рис. 3.1, 3.2 та 3.3.

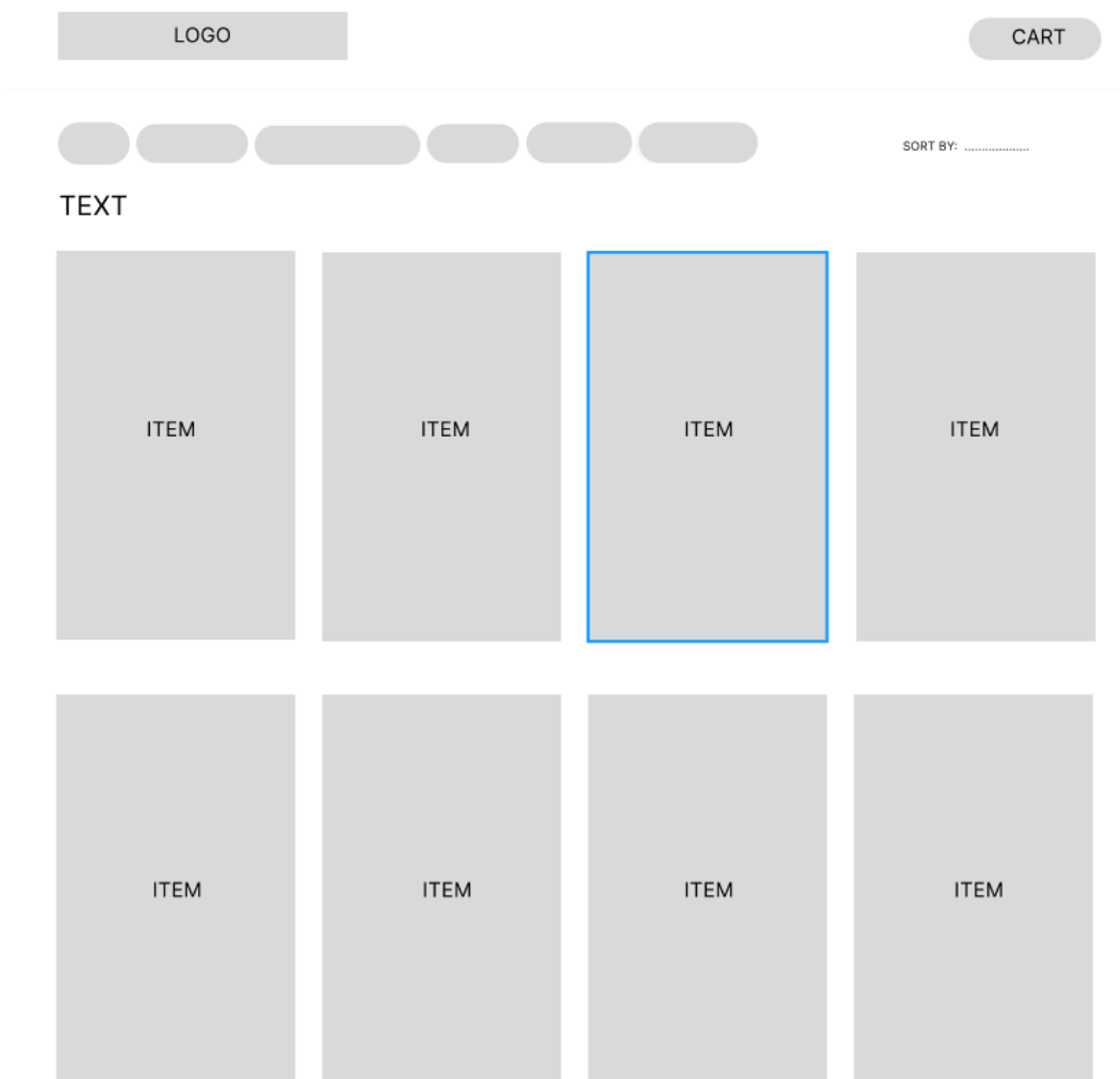


Рисунок 3.1 – Вайрфрейм головної сторінки

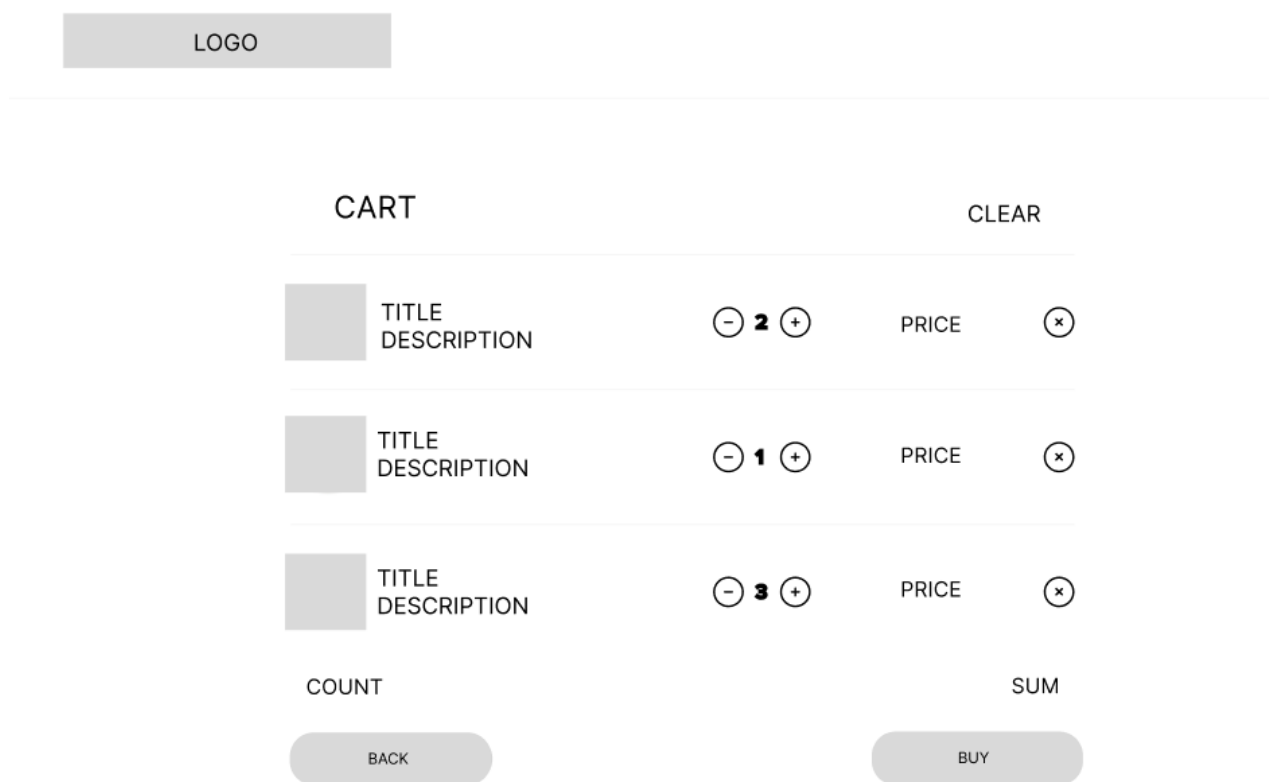


Рисунок 3.2 – Вайрфрейм сторінки з заповненим кошиком

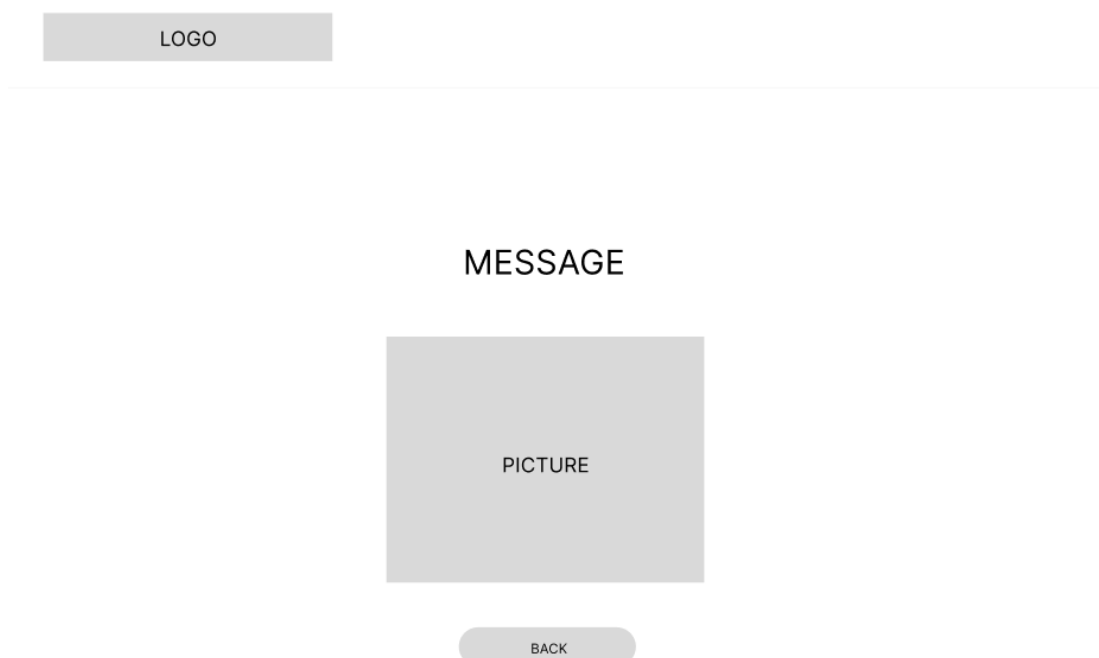


Рисунок 3.3 – Вайрфрейм сторінки з порожнім кошиком

Мокапи зображені на рис. 3.4 та 3.5.

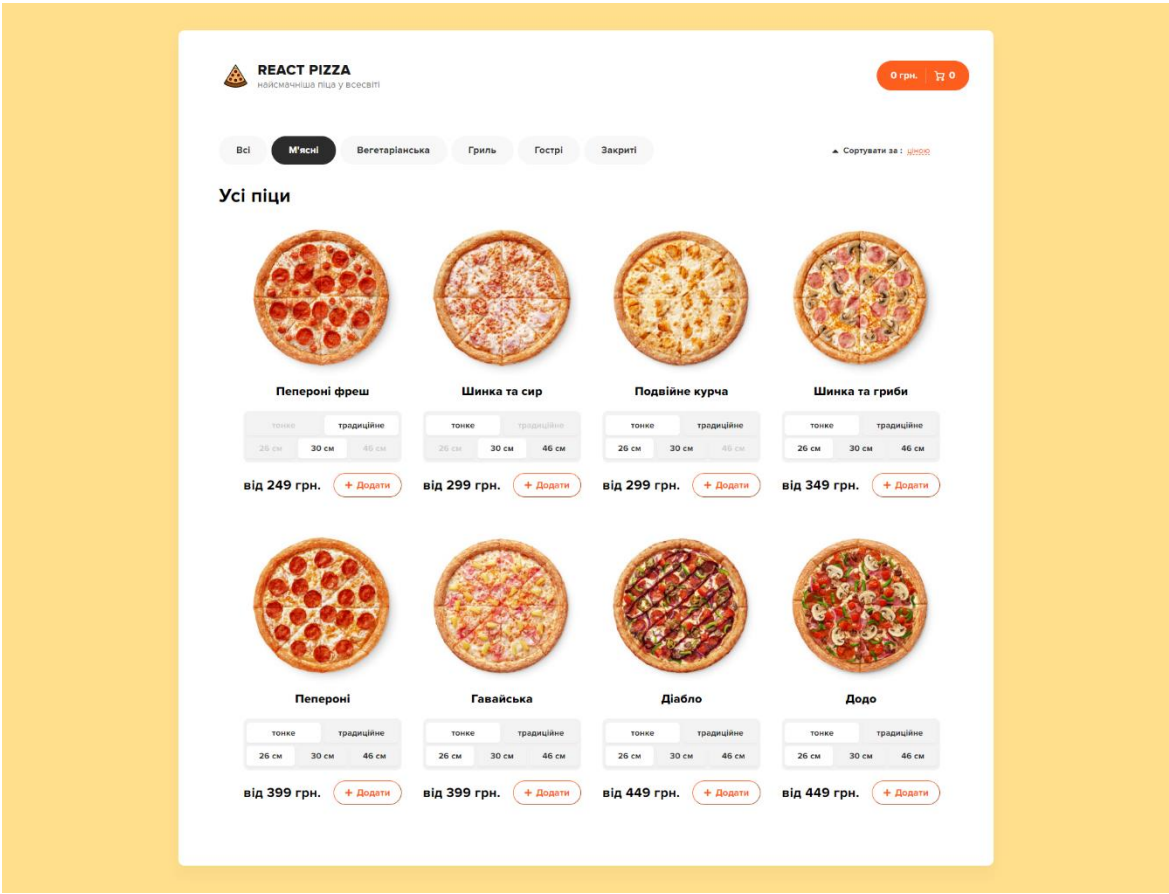


Рисунок 3.4 – Мокап головної сторінки

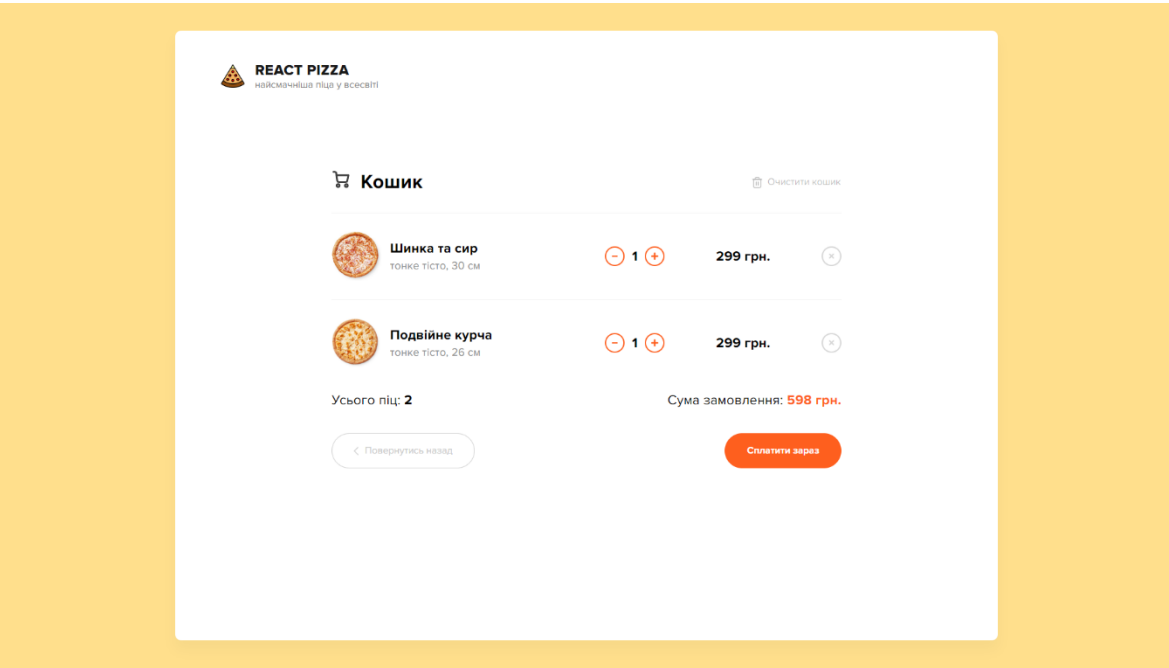


Рисунок 3.5 – Мокап сторінки з заповненим кошиком

Прототип для головної сторінки зображено на рис. 3.6.

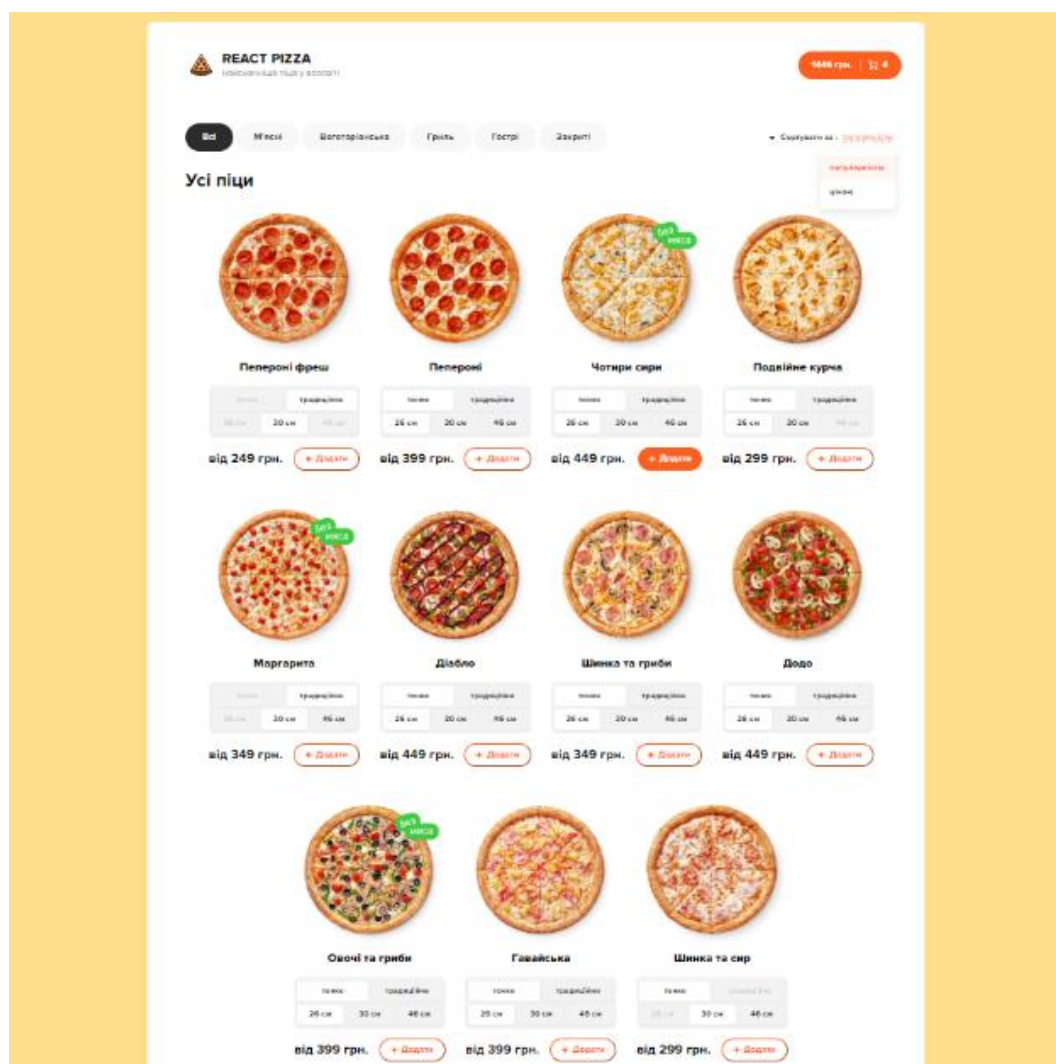


Рисунок 3.6 – Прототип головної сторінки

Прийняті рішення:

- Мініمالізм інтерфейсу: Зосередження на основних функціях та інтуїтивно зрозумілому навігаційному процесі.
- Кольорова палітра: Використання приємних та апетитних кольорів, що асоціюються з піцею.
- Адаптивний дизайн: Забезпечення оптимального відображення на різних пристроях для комфортного використання.

Процес проєктування спрямований на створення привабливого та зручного середовища для користувачів, щоб зробити процес замовлення піци максимально приємним та ефективним.

### 3.2 Розгортання серверу

Під час розгортання серверної частини веб-застосунку використовуючи Create React App (CRA), важливо дотримуватися певних кроків та процедур для забезпечення успішного розгортання та оптимальної продуктивності [10].

- Створення застосунку: Використовуючи команду `npm create-react-app my-app`, де `my-app` - це назва проєкту, створюється основний каркас застосунку. CRA автоматично налаштовує всі необхідні інструменти для розробки та розгортання.

- Інтеграція з `npm`: CRA надає вбудовані скрипти для розробки (`npm start`), збірки (`npm run build`), та інші, що спрощує використання. Інтеграція з `npm` дозволяє легко керувати різними етапами життєвого циклу застосунку.

- Автоматична конфігурація інструментів: CRA автоматично конфігурує Webpack [4], Babel та інші інструменти, щоб розробник міг фокусуватися на написанні коду, не втрачаючи час на вручну налаштовування конфігурацій.

- Структура проєкту: CRA створює чітку структуру проєкту, розташовуючи ресурси у відповідних папках, що полегшує організацію та розширення застосунку.

- Скрипти для розгортання: В `package.json` можна знайти різні скрипти для розгортання, такі як `start` (відкрийте `http://localhost:3000/`, щоб побачити свій додаток) для розробки та гарячого перезавантаження, `build` для створення оптимізованої версії для продакшену [6].

- Розгортання на вибраний сервер: Після створення оптимізованої версії застосунку за допомогою `npm run build`, можна вибрати сервер для

розгортання. Популярні сервіси, такі як Vercel, Netlify або GitHub Pages, надають зручні інтерфейси для розгортання застосунків.

### **3.3 Висновки за розділом 3**

У даному розділі проведено детальний аналіз та опис рішень, пов'язаних з проєктуванням компонентів системи. Основні етапи включали в себе розробку дизайну застосунку, розгортання серверної частини за допомогою Create React App (CRA), а також прийняття важливих архітектурних рішень.

На етапі проєктування дизайну застосунку використовувались 3 вайрфрейми, 2 мокапи та 1 прототип для головної сторінки. Ці інструменти допомогли візуалізувати структуру та зовнішній вигляд веб-застосунку, що сприяло кращому розумінню його функціоналу та вигляду.

Розгортання серверу за допомогою Create React App (CRA) надало простий та швидкий спосіб створення основи для веб-застосунку, враховуючи оптимізацію конфігурацій інструментів. Зручність використання прм-скриптів сприяє швидкому розгортанню та розвитку застосунку.

В ході проєктування та розгортання варто було узгодити ряд ключових архітектурних рішень, таких як вибір інструментів для розробки, структура проєкту та розгортання. Використання популярних та вже налаштованих інструментів значно полегшило цей процес.

Загалом, результати розділу вказують на успішне виконання етапів проєктування та підготовки до розгортання веб-застосунку. Використання сучасних інструментів дозволило швидко перейти від ідеї до готового застосунку, забезпечуючи при цьому його ефективність та стабільність.

## **4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ**

### **4.1 Призначення й умови застосування програми**

Призначення розробленої веб-програми полягає в створенні зручного та ефективного інструмента для замовлення піци онлайн.

Розроблена система підтримує:

- перегляд всіх позицій;
- фільтрацію піц за їх видом;
- сортування за критеріями;
- зручну реалізацію кошика;
- взаємодію між декількома вкладками в браузері.

Розроблена програма "Сайт для замовлення піци" призначена для зручного та ефективного здійснення онлайн-замовлень піци. Її застосування передбачає відповідні умови, спрямовані на забезпечення задоволення потреб користувачів та ефективне функціонування системи.

### **4.2 Методика та результати тестування**

#### **4.2.1 Чек-лист тестування**

Для забезпечення високої якості та надійності розробленого веб-сайту для замовлення піци, був розроблений детальний чек-лист тестування. Цей чек-лист включає в себе різноманітні тести, спрямовані на перевірку функціональності, безпеки та зручності використання застосунку.

#### **1. Вибір та замовлення піци:**

- Тестування коректності вибору піци з меню.
- Впевненість у правильному розміщенні та оформленні замовлення.

#### **2. Операції з кошиком:**

- Перевірка можливості додавання та видалення товарів у кошику.
- Розрахунок правильної суми замовлення.

### 3. Інтерфейс:

- Оцінка інтуїтивно зрозумілого та зручного дизайну інтерфейсу.
- Перевірка адаптивності сайту для різних пристроїв.

### 4. Тестування на різних браузерах:

- Перевірка сумісності веб-сайту з різними веб-браузерами (Chrome, Firefox, Safari, Edge).

Результати тестування фіксуються та аналізуються для подальшого виправлення виявлених помилок та поліпшення загальної ефективності системи.

#### **4.2.2 Тестування за сценарієм**

Опис сценарію тестування:

1. Користувач відкриває додаток.
2. Користувач сортує позиції за допомогою можливих параметрів.
3. Користувач обирає піцу з різних доступних варіантів та додає її до кошика.
4. Після вибору піци користувач переходить до кошика для перевірки та оформлення замовлення.

Результати тестування за обраним сценарієм:

Тестування сценарію підтвердило, що веб-сайт для замовлення піци працює із заданою функціональністю без суттєвих помилок. Кожен крок сценарію виконується коректно, і користувач отримує необхідну інформацію про стан свого замовлення. Всі етапи відбуваються ефективно та інтуїтивно зрозуміло для користувача.

Отже, розроблений веб-сайт успішно пройшов тестування за визначеним сценарієм, і його функціонал може бути впроваджений для реального використання з метою замовлення піци.



## ВИСНОВКИ

В ході виконання самостійного завдання з розробки веб-застосунку для замовлення піци, використовуючи технології React, Webpack, JavaScript, HTML, SCSS, broadcast-channel та json-server, було проведено аналіз предметної області, визначено мету та об'єкт дослідження.

Аналіз предметної області дозволив чітко сформулювати завдання, визначивши межі системи, функціональні вимоги, вимоги до інтерфейсу, продуктивності, безпеки, а також експлуатаційні та нефункціональні вимоги. Зазначено, що система розроблялася з урахуванням моделі MVC.

У розділі "Матеріали і методи" було розглянуто засоби розробки, зокрема вибір мови програмування (JavaScript), середовища розробки (VS Code), та визначено структурну схему розробки.

Основні рішення щодо реалізації компонентів системи включали проектування дизайну застосунку з використанням 3 вайрфреймів, 2 мокапів, та 1 прототипу для головної сторінки. Також проведено розгортання серверу з використанням Create React App.

У розділі "Експлуатація, тестування та експериментальне дослідження програми" описано призначення та умови застосування програми, розроблено чек-лист та проведено тестування за сценарієм.

Загальні висновки підсумовують успішне виконання завдання та відповідність розробленого веб-застосунку визначеним критеріям та вимогам. Розроблене рішення може бути успішно використане для замовлення піци, забезпечуючи зручний та ефективний інтерфейс для користувачів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Велика база знань від Mozilla для веб-розробників [Electronic resource]. – Access mode: <https://developer.mozilla.org/en-US/docs/Web>.
2. Офіційна документація з React. [Electronic resource]. – Access mode: <https://legacy.reactjs.org/docs/getting-started.html>.
3. Офіційний сайт VS Code. [Electronic resource]. – Access mode: <https://code.visualstudio.com/>.
4. Документація Webpack. [Electronic resource]. – Access mode: <https://webpack.js.org/guides/getting-started/>.
5. Руководство з JavaScript. [Electronic resource]. – Access mode: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>.
6. Офіційний сайт JSON. [Electronic resource]. – Access mode: <https://www.json.org/json-en.html>.
7. Документація з HTML. [Electronic resource]. – Access mode: <https://developer.mozilla.org/en-US/docs/Web/HTML>.
8. Документація Sass (мови, яка компілюється в CSS). [Electronic resource]. – Access mode: <https://sass-lang.com/documentation/>.
9. Документація API каналу мовлення. [Electronic resource]. – Access mode: [https://developer.mozilla.org/en-US/docs/Web/API/Broadcast\\_Channel\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Broadcast_Channel_API).
10. Документація Create React App. [Electronic resource]. – Access mode: <https://github.com/facebook/create-react-app#readme>.