

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
**Факультет електроніки та інформаційних технологій**  
**Кафедра інформаційних технологій**

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня бакалавр**

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Вебдодаток сервісу замовлень доставки з ресторанів

Здобувача групи ІТ-03-2 Глущенко Іван Вікторович

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_ (підпис)

Іван ГЛУЩЕНКО

Керівник

к.т.н., доцент, Анна НЕНЯ

\_\_\_\_\_ (підпис)

**Суми – 2024**

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра інформаційних технологій  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

В.о. зав. кафедри ІТ

\_\_\_\_\_ Ващенко С.М.

«\_\_» \_\_\_\_\_ 2024 р.

## **З А В Д А Н Н Я**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

*Глущенко Іван Вікторович*

**1 Тема роботи** Вебдодаток сервісу замовлень доставки з ресторанів

**керівник роботи** Неня Анна Вікторівна, к.т.н., доцент, \_\_\_\_\_

затверджені наказом по університету від «07» 05 2024 р. №0482-VI

**2 Строк подання студентом роботи** « 26 » травня 2024 р.

**3 Вхідні дані до роботи** технічне завдання

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** \_\_\_\_\_

аналіз предметної області, моделювання вебдодатку, розробка вебдодатку сервісу замовлень доставки з ресторанів

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** \_\_\_\_\_

Актуальність, постановка задачі, аналіз сайтів-аналогів, порівняльна таблиця аналогів, функціональні вимоги, діаграма бізнес-процесів в нотації IDEF0, концептуальний рівень, декомпозиція діаграми, діаграма варіантів використання, модель бази даних, засоби реалізації, демонстрація результату, висновки

## 6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання      8 березня 2024

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної області	11.04.24 – 15.04.24	
2	Розробка технічного завдання	16.04.24 – 17.04.24	
3	Планування робіт проекту	18.04.24 – 21.04.24	
4	Огляд останніх досліджень	22.04.24 – 25.04.24	
5	Аналіз існуючих вебдодатків	26.04.24 – 30.04.24	
6	Постановка задачі	01.05.24 – 02.05.24	
7	Вибір засобів реалізації	03.05.24 – 10.05.24	
8	Структурно-функціональне моделювання	11.05.24 – 14.05.24	
9	Розробка вебдодатку	15.05.24 – 29.05.24	
10	Оформлення документації	30.05.24 – 02.06.24	

Студент

\_\_\_\_\_

(підпис)

Глущенко І.В.

Керівник роботи

\_\_\_\_\_

(підпис)

к.т.н., доц. Неня А.В.

## АНОТАЦІЯ

**Записка:** 99 сторінок, 58 рисунків, 4 додатки, 24 використаних джерел та літератури.

**Обґрунтування актуальності теми роботи:** В сучасному світі зростає популярність замовлення їжі з ресторанів через мобільні додатки та вебплатформи. Це обумовлено збільшенням кількості людей, які використовують ці технології для зручності та швидкості отримання їжі. Розробка вебдодатка сервісу замовлень доставки з ресторанів стане важливим інструментом для полегшення процесу замовлення та доставки їжі, що дозволить підвищити конкурентоспроможність сервісу та рівень задоволення клієнтів.

**Мета роботи:** розробка вебдодатка сервісу замовлень доставки з ресторанів для забезпечення зручності та ефективності процесу замовлення їжі.

**Методи дослідження:** аналіз та впровадження сучасних вебтехнологій, вивчення та впровадження кращих практик управління процесом замовлення та доставки їжі.

**Результат:** Розроблений вебдодаток призначений для замовлення та доставки їжі з ресторанів, що забезпечує зручність та ефективність процесу. Ця система автоматизує завдання, полегшує взаємодію між клієнтами та ресторанами, та забезпечує високий рівень задоволення клієнтів. Використання цього вебдодатка призведе до підвищення продуктивності ресторанів, кращого контролю над процесом доставки та покращення рівня задоволення клієнтів.

**Ключові слова:** вебдодаток, замовлення їжі, доставка, ресторани, ефективність, зручність, технології.

## ЗМІСТ

ВСТУП .....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	8
1.1 Огляд останніх досліджень та публікацій.....	8
1.2 Переваги та недоліки вебдодатків для замовлення доставки їжі.....	10
1.3 Аналіз програмних продуктів – аналогів .....	14
1.4 Обґрунтування актуальності розробки.....	21
1.5 Аналіз існуючих технологій вирішення задачі .....	22
1.6 Постановка завдання.....	25
2 ПРОЕКТУВАННЯ WEB-ПЛАТФОРМИ .....	26
2.1 Структурно-функціональне моделювання .....	26
2.2 Моделювання варіантів використання .....	28
2.3 Проектування моделі бази даних .....	30
3. РЕАЛІЗАЦІЯ ВЕБДОДАТКУ СЕРВІСУ ЗАМОВЛЕНЬ ДОСТАВКИ З РЕСТОРАНІВ .....	31
3.1 Реалізація бази даних.....	31
3.2 Реалізація серверної та клієнтської частини .....	35
3.3 Настанови з використання .....	39
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	52
ДОДАТОК А.....	56
ДОДАТОК Б .....	61
ДОДАТОК В.....	69
ДОДАТОК Г .....	84

## ВСТУП

**Актуальність.** Сервіси ресторанної доставки є важливим інструментом підтримки конкурентоспроможності в гастрономічній індустрії. Це популярний спосіб залучити нових клієнтів і розширити свою аудиторію. Використання веб-додатків для організації процесу замовлення доставки їжі має вирішальне значення для того, щоб залишатися на передовій у цій галузі.

Метою є створення вебдодатку, який дозволить користувачам замовляти їжу з різних ресторанів. Даний вебдодаток надасть зручний інструмент для оформлення замовлення доставки в сфері харчової індустрії та забезпечить безпрограшну ситуацію для всіх залучених сторін. Платформа принесе користь як ресторанам, розширивши їх клієнтську базу, так і користувачам, забезпечивши швидке та ефективне замовлення їжі.

**Мета.** Розробка вебдодатку для замовлення доставки їжі та відслідковування статусу доставки, що надасть можливість користувачам замовляти їжу з ресторанів за вказаною адресою та отримувати доставку до дому або офісу.

Декомпозиція мети на задачі наступна:

- аналіз останніх тенденцій та ринку послуг з доставки їжі;
- аналіз аналогів вебдодатків для замовлення доставки їжі та визначення ключових критеріїв;
- розробка функціоналу вебдодатку для замовлення доставки їжі, включаючи систему оформлення замовлення, перегляду наявних послуг а також вибір способу доставки та оплати;
- визначення технологій та системи управління бази даних для розробки вебдодатку.

**Об'єкт дослідження.** Процес замовлення та доставки їжі з ресторанів за допомогою вебдодатку.

**Предмет дослідження:** Вебдодаток для замовлення доставки їжі з ресторанів.

**Практична цінність:** Розробка даного вебдодатку для замовлення їжі з ресторанів надасть зручний підхід до підтримки реалізації операції доставки їжі з ресторанів за вказаною адресою. Використання сучасних технологій реалізації дозволять створити зручний та функціональний додаток, який задовольнить потреби користувачів та менеджерів.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень та публікацій

Інтернет і технологічний прогрес мають великий вплив на діяльність ресторанів. Оскільки більшість людей проводять значну частину свого дня в Інтернеті, це створює величезний ринковий потенціал для ресторанів. Насправді, попит на онлайн-замовлення в ресторанах продовжує зростати серед споживачів ресторанних послуг. Власники та менеджери ресторанів постійно шукають рішення, які дозволять клієнтам розміщувати замовлення онлайн і швидко отримувати їжу. Системи онлайн-замовлень і доставки поступово стають нормою, оскільки все більше ресторанів адаптуються до цієї бізнес-моделі.

Онлайн-замовлення та доставка в ресторанній індустрії розвиваються завдяки готовності онлайн-ринку. Дослідження, проведене Pew Research Center, показало, що близько 26% дорослих американців проводять цілий день онлайн, а 43% – кілька разів протягом дня [1].

Така поведінка споживачів продовжує впливати на те, як працює бізнес у ресторанній індустрії. Зокрема, онлайн-замовлення та доставка поступово стають нормою серед ресторанів.

Система онлайн-замовлень і доставки має вирішальне значення для сучасного ресторану. Вона сприяє збільшенню обсягів продажів і підвищенню рівня задоволеності клієнтів. Якщо поглянути на статистику, то ця можливість все ще залишається великою і привабливою. Близько 60% клієнтів замовляють їжу онлайн щонайменше раз на тиждень, а ще 31% - щонайменше двічі. З них близько 34% витрачають щонайменше \$50, замовляючи їжу онлайн. Це чітко вказує на те, скільки ресторан може заробити, продаючи їжу онлайн [2].

За допомогою ефективного програмного забезпечення для замовлень ресторан може продавати їжу безпосередньо клієнтам без залучення

зовнішнього порталу. За такого підходу клієнт робить замовлення безпосередньо з сайту. Він переглядає меню і вибирає позиції, які хотів би замовити. Потім клієнт оплачує свій вибір, і їжа доставляється в потрібне йому місце, або ж він може забрати її самостійно.

Онлайн-замовлення та доставка їжі стали можливими завдяки додаткам підтримки діяльності ресторанів. Нещодавнє поширення додатків зробило життєво важливим для ресторанів йти в ногу з часом. Сучасні тенденції показують, що додатки для онлайн-замовлення та доставки їжі незабаром стануть невід'ємним елементом управління успішним рестораном. Ресторану без таких додатків буде важко конкурувати.

Система онлайн-замовлення та доставки забезпечує повноцінний канал продажів для ресторану. Це означає, що ресторан може використовувати її як інструмент для отримання більшого прибутку та кращої організації роботи ресторану. Це також дозволяє власникам ресторанів заощаджувати на витратах на робочу силу та площі ресторану, необхідних для обслуговування таких клієнтів. При правильному впровадженні система може допомогти ресторану обійти своїх конкурентів, ставши ближчим до клієнтів [3].

Система онлайн-замовлень є ідеальною альтернативою стороннім агрегаторам або маркетплейсам. Харчовий маркетплейс – це централізований хаб, де кілька ресторанів та закладів харчування розміщують свої продукти для замовлення клієнтами. Коли клієнт відвідує платформу, він знайомиться з переліченими ресторанами та кухнями. Потім він може замовити їжу безпосередньо в ресторан, виходячи з того, що він хоче. Це вигідно для клієнтів, оскільки вони можуть досліджувати страви з різних ресторанів в одному місці замість того, щоб встановлювати кілька додатків для кожного ресторану. Однак маркетплейс призводить до збільшення витрат з боку ресторану через комісійні та інші збори [4].

Система онлайн-замовлень дозволяє обмежити витрати на комісійні, пов'язані з маркетплейсами та платіжними компаніями. Завдяки простоті використання та ефективності доставки їжі, система ефективна для залучення

нових клієнтів та завоювання лояльності існуючих. З огляду на те, що 70% клієнтів вважають за краще замовляти їжу безпосередньо в ресторані, впровадження системи є вірним способом залучення більшої кількості клієнтів. При правильній інтеграції в POS-систему ресторану, онлайн-замовлення також спрощує процес прийому замовлень. Воно усуває помилки, пов'язані з людським фактором, і довгий час доставки, дозволяючи клієнтам миттєво розміщувати замовлення, а замовлення потрапляє на кухню за лічені секунди [5].

Системи онлайн-замовлень і доставки забезпечують чіткий канал продажів для ресторанів. Клієнти переглядають пункти меню, обирають, що замовити, оформлюють замовлення та здійснюють оплату через додаток або веб-сайт ресторану. Отримавши замовлення, ресторан готує їжу і доставляє її за вказаною клієнтом адресою. Це допомагає ресторану охопити більшу частку ринку та обслуговувати клієнтів, не вимагаючи від них приходу до закладу.

## **1.2 Переваги та недоліки вебдодатків для замовлення доставки їжі**

Аналіз предметної області надає чітке розуміння про значний попит користувачів щодо додатків для замовлення доставки їжі. Так звана діджиталізація дистрибуції продуктів харчування дозволяє транспортувати їжу швидше і з меншою кількістю людських контактів. Враховуючи вищезазначені потреби користувачів які потребують автоматизації для підвищення взаємодії користувачів з рестораном, необхідно проаналізувати недоліки та переваги додатків даного типу [6].

Перелік переваг та недоліків вебдодатків для замовлення доставки їжі представлено у таблиці 1.1.

Таблиця 1.1 – Переваги та недоліки вебдодатків для замовлення доставки їжі

Переваги	Недоліки
<p>Запуск вебдодатку онлайн-замовлення їжі додає бізнесу гнучкості, що в кінцевому підсумку збільшує продажі та прибутки.</p>	<p>Одним з основних недоліків вебдодатків онлайн-замовлення їжі є ціна. Коли їжа замовляється для більш ніж однієї людини, її вартість зазвичай дорівнює вартості вечері в хорошому ресторані щовечора. Для людей з обмеженим бюджетом на харчування системи онлайн-замовлення їжі часто є занадто дорогими.</p>
<p>Однією з важливих переваг систем замовлення їжі - користь для здоров'я. Оскільки їжа запланована, легко визначити точну кількість калорій, спожитих в кожному прийомі їжі. Багато систем замовлення їжі зберігають меню для користі для здоров'я та схуднення, що може бути дуже корисним для людей, які намагаються схуднути і перейти на здорову дієту.</p>	<p>Більшість вебдодатків замовлення їжі мають обмежену кількість страв. Меню змінюється кожні кілька тижнів або місяців, але якщо ви користуєтеся системою довше, ніж кілька місяців, пункти меню будуть повертатися знову і знову.</p>

## Продовження таблиці 1.1

Переваги	Недоліки
<p>Щоб знову відкритися, підприємствам харчової промисловості доведеться організувати роботу відповідно до норм охорони здоров'я та безпеки, встановлених індійським урядом. Власники повинні дотримуватися соціальної дистанції, використовувати безконтактні способи замовлення та оплати, а також забезпечувати регулярне прибирання поверхонь.</p>	<p>Більшість вебдодатків замовлення їжі надають заморожену їжу. Зазвичай їх легко готувати, але зазвичай це займає більше години, оскільки їжа заморожена. Щоб уникнути тривалого приготування, ви можете дістати їжу з морозильної камери напередодні.</p>
<p>Однією з переваг системи онлайн-замовлення їжі для клієнтів є те, що вона гарантує точність цін і зменшує ймовірність помилки, коли приходить час оплачувати рахунок. Це має низку переваг для вашого бізнесу: низькі шанси на неправильне нарахування, менше часу на помилки при замовленні та надання задовільного сервісу клієнтам.</p>	<p>Однією з проблем вебдодатків замовлення їжі є те, що якість поданої їжі часто гірша, ніж у ресторані. Часто їжу доводиться доставляти на великі відстані, і з часом дорогоцінні вітаміни можуть бути втрачені. Крім того, їжа з замовлень часто подається в пластиковій упаковці, яка може бути не дуже привабливою для очей у порівнянні з їжею, акуратно покладеною на тарілку в ресторані.</p>

## Продовження таблиці 1.1

Переваги	Недоліки
Якщо спосіб оплати та меню не викликає жодних проблем, то постійні клієнти будуть рекомендувати сервіс своїм друзям і розповідатимуть про додаток у соціальних мережах, а також ділитися інформацією про нього.	Через великі відстані замовлення, їжа може бути холодною, коли її нарешті доставлять додому, і доведеться розігрівати її або їсти холодною. Це особливо актуально, якщо замовлення виконується в екстрених випадках: вулиці часто переповнені, і замовник може затриматися в заторі.
За допомогою системи онлайн-замовлень у ресторані можна надавати персоналізовані пропозиції постійним клієнтам, запитувати відгуки, щоб підвищити свій рейтинг, і отримувати зворотній зв'язок щодо діяльності вебдодатку.	У деяких ресторанах є також хороша обставина, яку користувач втрачає, якщо замовляє доставку їжу додому. Наприклад, якщо користувач проведе вечір у хорошому китайському ресторані, то буде відчуватися атмосфера перебування в Китаї, тому що обстановка і вся атмосфера відповідають китайському способу життя. В разі замовлення їжі додому, то можна втратити атмосферу перебування.

*Джерело: розроблено автором*

Аналіз наведених недоліків і переваг впровадження вебдодатків в роботу ресторанів для реалізації підтримки замовлень доставки вказує на актуальність та популярність сервісу.

### 1.3 Аналіз програмних продуктів – аналогів

Для аналізу сервісів було обрано наступні вебдодатки: Mister.AM, IKURA, Glovo.

Mister.AM співпрацює з неймовірним розмаїттям ресторанів, від національних мереж до сімейних кафе, щоб задовольнити майже всі смакові вподобання користувачів – і все це за запитом [7].

За допомогою Mister.AM користувач може шукати ресторани за кухнею або місцем розташування, швидко замовляти їжу з меню і відстежувати своє замовлення в вебдодатку – все це без зайвих операцій і додаткових витрат (окремі ресторани можуть встановлювати мінімальні замовлення та інші збори). Користувачі Mister.AM можуть замовляти доставку або забрати їжу з собою, а також скористатися спеціальними пропозиціями, доступними лише в рамках сервісу. А у вікні з інструкціями щодо доставки в додатку користувач може вказати водію, де залишити їжу, якщо є бажання, щоб доставка була безконтактною.

На рисунках 1.1 – 1.3 представлено вигляд головної сторінки, сторінки обраного ресторану та сторінку оформлення замовлення вебдодатку Mister.AM

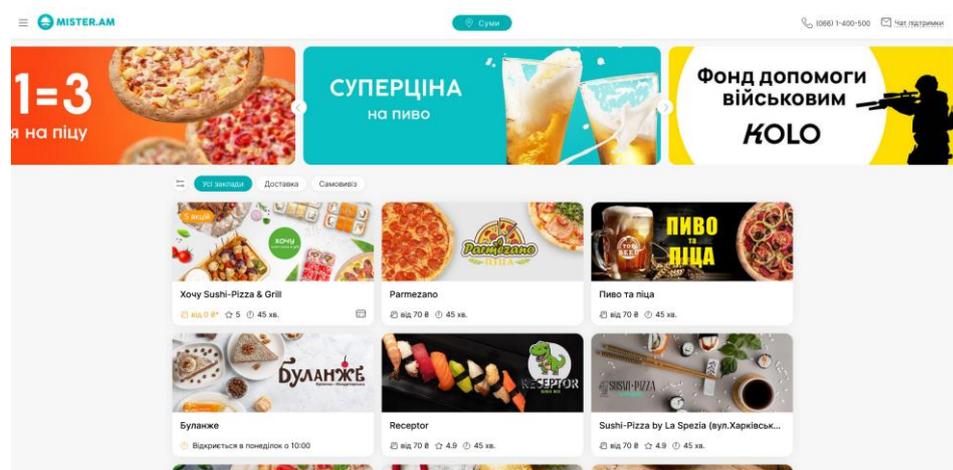


Рисунок 1.1 – Вигляд головної сторінки вебдодатку Mister.AM

Джерело: [7]

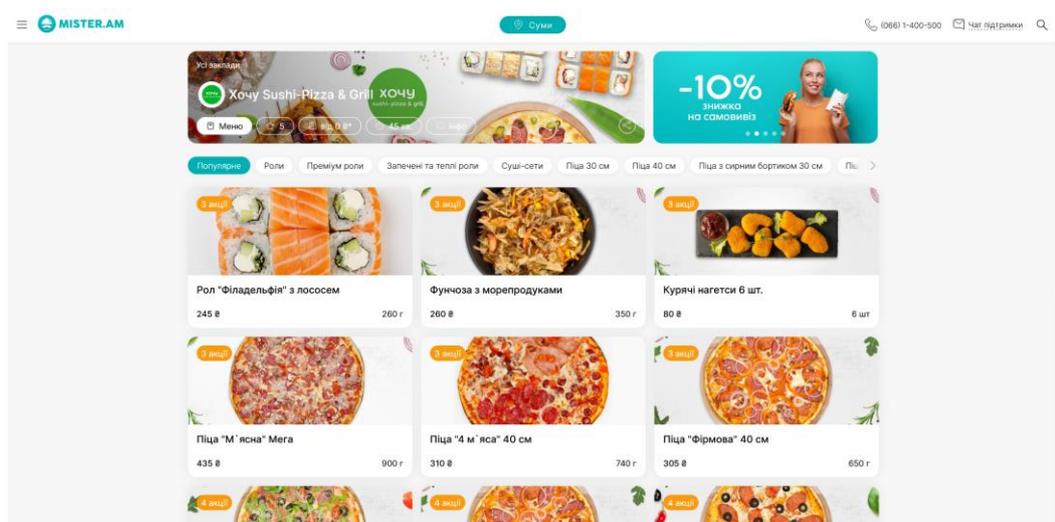


Рисунок 1.2 – Вигляд сторінки обраного ресторану

Джерело: [7]

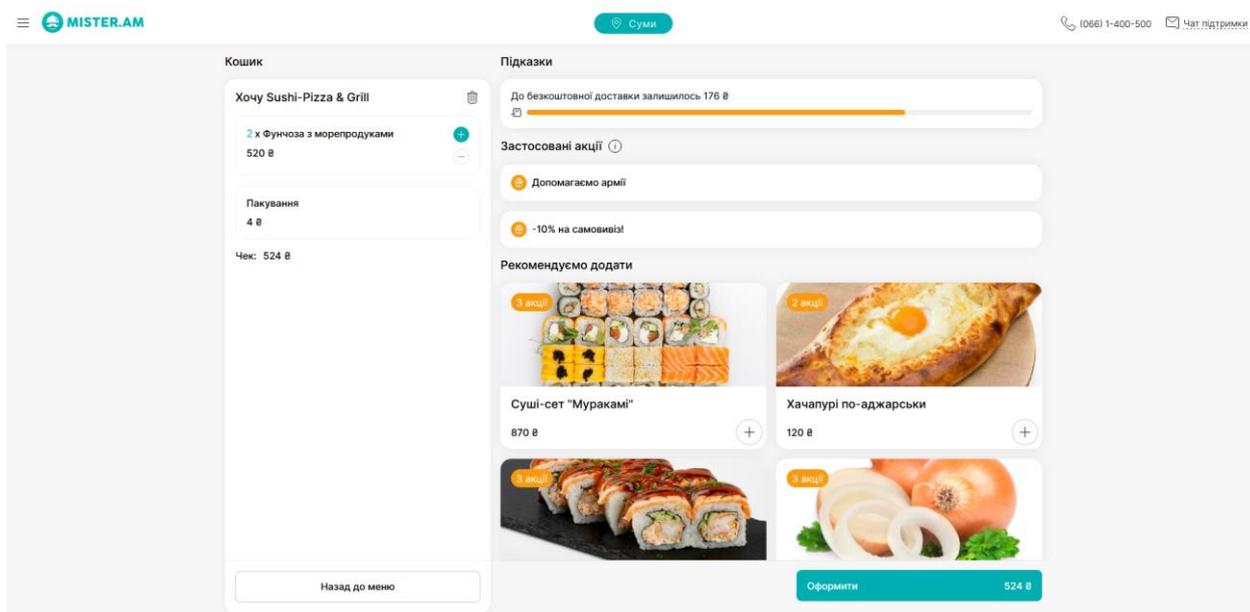


Рисунок 1.3 – Вигляд сторінки оформлення замовлення

Джерело: [7]

IKURA працює у 1000 містах України. Пропонуючи широкий вибір страв від суші до фаст-фуду, IKURA дозволяє легко робити замовлення

(включаючи швидке повторне замовлення улюблених страв), з можливістю відстежувати доставку, забирати їжу або планувати доставку на найбільш зручний час [8].

Вебдодаток включає "Yum Score", який оцінює якість їжі та те, наскільки добре ресторан співпрацює з IKURA, щоб швидко доставити їжу - корисна функція, якщо користувач хоче, щоб ваша їжа була доставлена свіжою та гарячою. IKURA також дозволяє замовляти алкоголь і міксери з місцевих винних і лікєро-горілчаних магазинів, де це дозволено місцевим законодавством. Як і в інших найкращих службах доставки їжі, поле з інструкціями щодо доставки в IKURA дозволяє вам вказати, де користувач хоче, щоб його замовлення доставили для безконтактної доставки.

На рисунках 1.4 – 1.6 представлено вигляд головної сторінки, сторінки обраного ресторану та сторінку оформлення замовлення вебдодатку IKURA.

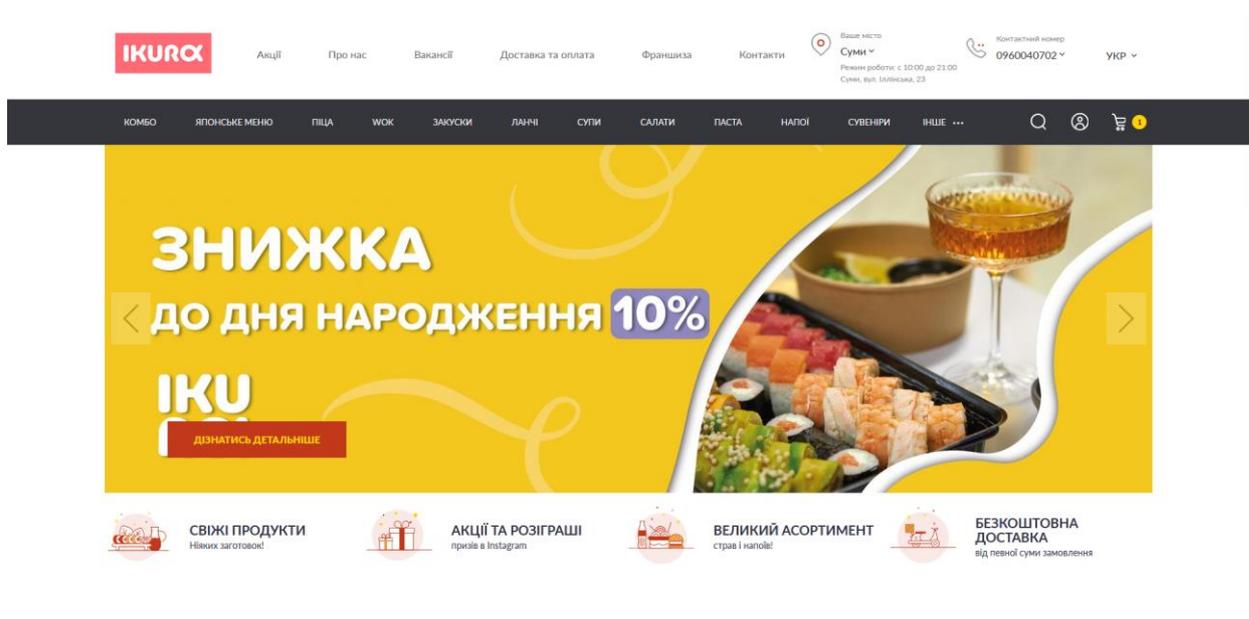


Рисунок 1.4 – Вигляд головної сторінки вебдодатку IKURA

Джерело: [8]

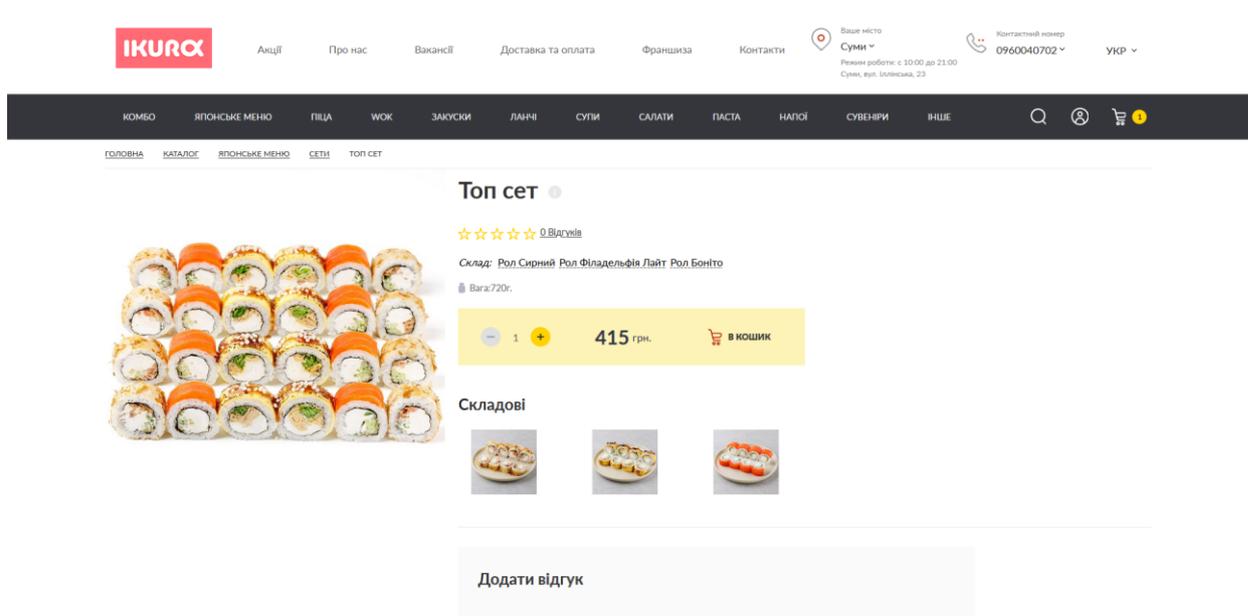


Рисунок 1.5 – Вигляд сторінки обраної страви

Джерело: [8]

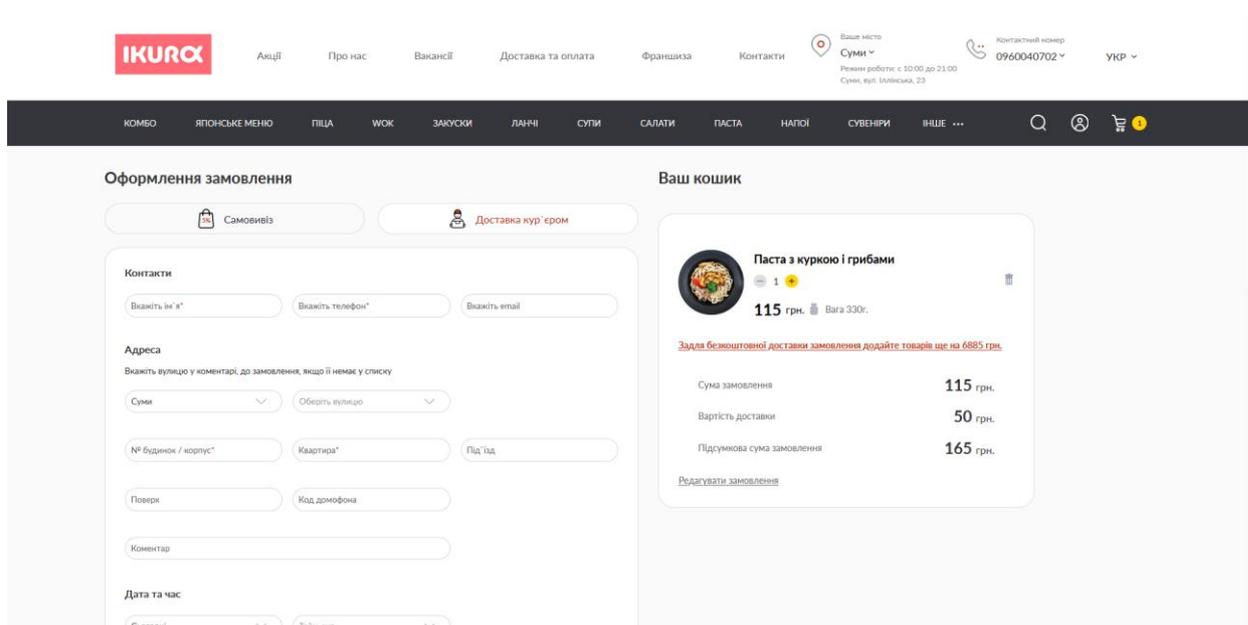


Рисунок 1.6 – Вигляд сторінки оформлення замовлення

Джерело: [8]

Експансія Glovo на ринок додатків для доставки їжі - це стратегічний крок, який використовує досвід компанії у сфері технологій "на вимогу", отриманий завдяки додатку для спільного використання поїздок, для розвитку

індустрії доставки їжі. Користувачі Glovo можуть легко знайти свій улюблений ресторан або відкрити для себе нові заклади поблизу [9].

Користувачі можуть впевнено розміщувати замовлення на негайну або заплановану доставку та відстежувати його в режимі реального часу, подібно до замовлення поїздки в Glovo. Оплату можна здійснити кредитною картою або кредитами Glovo. Додаток також дозволяє давати кур'єру чайові та залишати інструкції для безконтактної доставки. Крім того, з кур'єром можна зв'язатися через додаток для отримання будь-яких подальших інструкцій або запитань.

Глобальне охоплення Glovo є значною перевагою, що дає можливість замовляти їжу на вимогу по всій Україні.

На рисунках 1.7 – 1.9 представлено вигляд головної сторінки, сторінки обраного ресторану та сторінку авторизації вебдодатку Glovo.

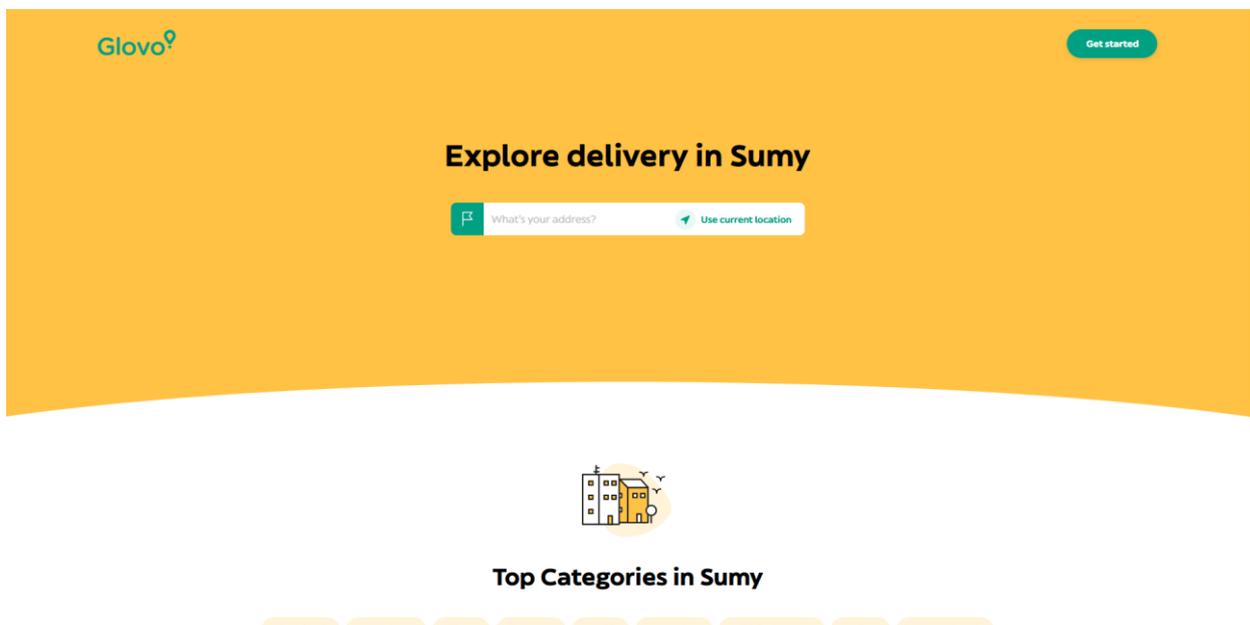


Рисунок 1.7 – Вигляд головної сторінки вебдодатку Glovo

*Джерело: [9]*

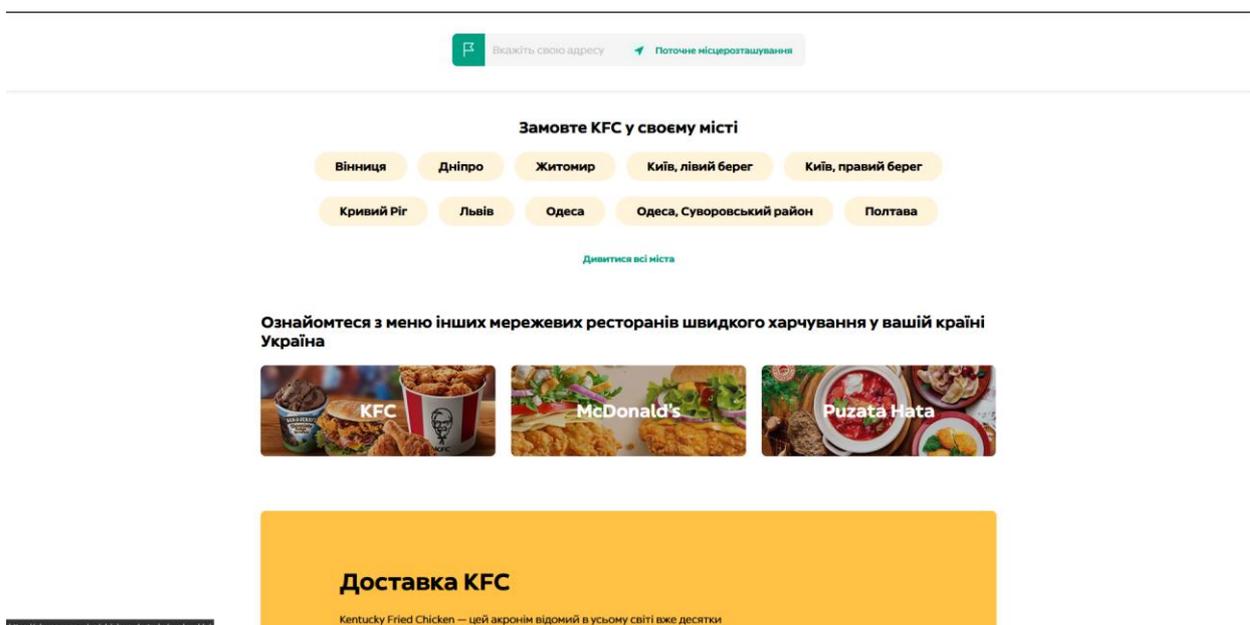


Рисунок 1.8 – Вигляд сторінки обраного ресторану вебдодатку Glovo  
Джерело: [9]

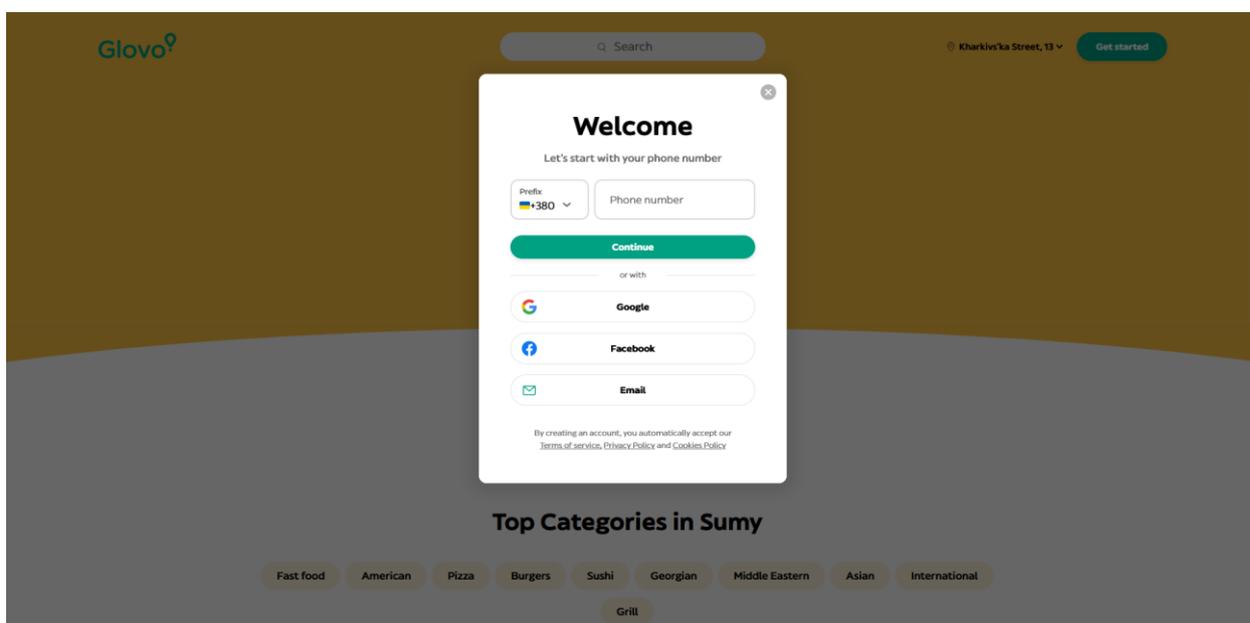


Рисунок 1.9 – Вигляд сторінки авторизації вебдодатку Glovo  
Джерело: [9]

Оцінюючи служби доставки їжі, важливо враховувати їхнє охоплення у місцевості та здатність доставляти їжу з ресторанів, яким користувачі надають

перевагу. Також необхідно звернути увагу про будь-які додаткові збори, хоча їх може бути важко знайти, оскільки вони часто визначаються рестораном.

Для більш детального аналізу було створено таблицю порівняння додатків – аналогів результати якої представлено в таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика аналогів вебдодатків

Параметр /Сервіс	Mister.AM	IKURA	Glovo	Food Delivery
Географічне покриття	Більше 20 міст по всій Україні	Більше 10 міст по всій Україні	Доступ по всьому світу	По всій Україні
Асортимент	Ресторани, кафе, фаст-фуди	Японська кухня	Ресторани, кафе, продуктові магазини	Ресторани, кафе, фаст-фуди
Інтерфейс користувача	Інтуїтивно зрозумілий	Інтуїтивно зрозумілий з акцентом на японську кухню	Інтуїтивно зрозумілий	Інтуїтивно зрозумілий
Час доставки	Індивідуально, в залежності від потреб клієнта	Залежить від ресторану та часу замовлення	В залежності від ресторану та типу доставки	В залежності від ресторану та типу доставки
Вартість доставки	В залежності від відстані	В залежності від спеціальних страв	В залежності від відстані та типу замовлення	В залежності від відстані та типу замовлення
Платіжні методи	Онлайн-оплата, готівка	Готівка, онлайн-оплата	Готівка, онлайн-оплата	Готівка, онлайн-оплата

Продовження таблиці 1.2 - Порівняльна характеристика аналогів вебдодатків

Параметр /Сервіс	Mister.AM	IKURA	Glovo	Food Delivery
Особливості	Акції та знижки	Акції та накопичувальна система	Великий вибір категорій товарів та послуг,	Акції та знижки

*Джерело: розроблено автором*

#### 1.4 Обґрунтування актуальності розробки

Надання клієнтам можливості замовляти їжу онлайн через вебдодаток має кілька переваг. Для власника ресторану це важливий крок до розширення цільової аудиторії та підвищення впізнаваності закладу. Такий підхід демонструє впевнений і дипломатичний підхід до обслуговування клієнтів, що є важливим для будь-якого успішного ресторанного бізнесу. Для власника ресторану це важливий крок до розширення цільової аудиторії та підвищення впізнаваності вашого закладу. Пропонуючи онлайн-замовлення, клієнти оцінять зручність можливості швидко і легко робити замовлення в будь-який час, зменшуючи при цьому ймовірність непорозумінь при оформленні замовлень.

Система онлайн-замовлення їжі може принести велику користь ресторанам. Налаштувати систему онлайн-замовлення їжі дуже просто. Просто натиснувши кілька кнопок, можна встановити свою персоналізовану платформу і відчувати всі переваги, описані вище, що дасть змогу досягти успіху на ринку.

## 1.5 Аналіз існуючих технологій вирішення задачі

Вибір відповідної технології має вирішальне значення для розробки успішного цифрового програмного продукту. Це пов'язано з тим, що створення продукту передбачає більше, ніж просто розробку візуально привабливого користувацького інтерфейсу (UI) та зручного для користувача досвіду (UX). Це також передбачає розробку стабільного, безпечного та підтримуваного продукту, який не лише приваблює клієнтів, але й дозволяє масштабувати бізнес. Правильна технологія може допомогти в досягненні цих цілей [10].

Технологічний стек – це комбінація програмних інструментів і мов програмування, які використовуються для розробки веб- або мобільних додатків. Ці додатки складаються з фронтенду та бекенду, які є клієнтською та серверною частинами відповідно.

Інтерфейс, який також називають клієнтською частиною, – це частина програми, з якою користувачі безпосередньо взаємодіють. У випадку вебдодатків ця взаємодія відбувається у веб-браузері і забезпечується різноманітними інструментами програмування. Клієнтські вебдодатки зазвичай створюються з використанням JavaScript, HTML і CSS.

Нижче наведено огляд компонентів, які складають стек технологій інтерфейсу.

Для розробки вебдодатку доставки їжі необхідно використовувати останню версію HTML, HTML5, яка включає нові елементи та атрибути для більш простого та ефективного створення вебдодатків. Головною перевагою HTML5 є підтримка аудіо та відео, чого не було в попередніх версіях HTML.

Також необхідно використовувати CSS3, останню версію каскадних таблиць стилів, у поєднанні з HTML5 для опису зовнішнього вигляду та форматування HTML-документа.

CSS3 – це потужний інструмент, який підтримує адаптивний дизайн, дозволяючи елементам сайту плавно адаптуватися до різних розмірів екрану. CSS3 розділений на окремі модулі, що розширює його функціональність і спрощує використання. Крім того, CSS3 забезпечує чудову підтримку анімації та 3D-трансформацій.

JavaScript (JS) – це основна технологія для створення інтерфейсів вебдодатків. Він широко використовується для створення динамічних та інтерактивних веб-сторінок, дозволяючи створювати як прості, так і складні веб-анімації, що значно покращують користувацький досвід. JavaScript широко використовується у небраузерних середовищах, таких як веб-сервери та бази даних [11].

PHP – це потужна та універсальна мова програмування з відкритим вихідним кодом, яка є ідеальним вибором для розробки вебдодатків завдяки зручному синтаксису який подібний до синтаксису C, Java та Perl, що полегшує швидке навчання для розробників, знайомих з цими мовами. Крім того, її вихідний код знаходиться у вільному доступі для спільноти, що демонструє її прихильність до прозорості та співпраці. Розробники можуть впевнено модифікувати код відповідно до своїх специфічних вимог і робити свій внесок у розвиток спільноти, ділячись своїм кодом.

PHP є чудовим вибором для створення динамічних веб-сторінок, оскільки його можна легко вбудувати в HTML-код.

Це дозволяє розробникам додавати динамічний контент на веб-сторінки, роблячи сайт більш цікавим та інтерактивним для користувачів. Крім того, PHP має високу масштабованість, що означає, що вона може обробляти великий обсяг веб-трафіку без шкоди для своєї продуктивності. При виборі мови програмування для веб-розробки, PHP є дипломатичним вибором, який балансує між функціональністю та продуктивністю [12].

Дана мова має велику спільноту розробників, які роблять свій внесок у розвиток мови, фреймворків та інструментів. Спільнота також надає підтримку та допомогу розробникам, які стикаються з проблемами при

створенні PHP-додатків. Мова PHP - це надійний і перспективний вибір для створення вебдодатків. Велика спільнота розробників постійно розвиває та вдосконалює мову, демонструючи її надійність та авторитет у цій галузі.

У сфері веб-розробки база даних – це організована та структурована колекція записів даних або файлів. Базу даних можна використовувати для зберігання та управління різними типами інформації, такими як транзакції з продажу, каталоги товарів, інвентаризації та профілі клієнтів, особливо в контексті розвитку електронної комерції.

PostgreSQL є чудовим вибором для розробки фінансового програмного забезпечення, виробництва, досліджень та наукових проектів завдяки своїм винятковим аналітичним можливостям та потужному SQL-рушію, який дозволяє з легкістю обробляти великі обсяги даних.

MySQL – це високопродуктивна і масштабована система управління базами даних, що робить її чудовим вибором для веб-розробки, особливо для додатків, які значною мірою покладаються на багаторядні транзакції, таких як типові додатки для доставки їжі. Вона залишається чудовим вибором для широкого спектру додатків.

MongoDB, з іншого боку, пропонує численні можливості завдяки своїй моделі даних на основі документів. MongoDB з її специфічними геопросторовими функціями є чудовим варіантом для обчислення відстаней та отримання геопросторової інформації про клієнтів. Це також цінне доповнення до технологічного стеку для електронної комерції, івент та ігрових додатків [14].

Для розробки вебдодатку сервісу замовлень доставки з ресторанів в якості основної СУБД було обрано MySQL через розширений функціонал з PHP.

## 1.6 Постановка завдання

Мета цього проекту – розробка вебдодатку сервісу замовлень доставки з ресторанів, який полегшує процес замовлення доставки їжі, виступаючи посередником між ресторанами та клієнтами. Вебдодаток дозволить користувачам впевнено обирати страви з асортименту ресторанів, та оформлювати замовлення. Ресторани зможуть впевнено приймати замовлення від клієнтів, вести з ними діалог і отримувати відгуки про їжу та обслуговування засобами вебдодатку.

Для успішної реалізації поставленого завдання необхідно виконати наступні етапи:

- проаналізувати потреби сучасного ринку в вебдодатку для замовлення доставки їжі, які підкреслюють сучасні тренди споживання;
- дослідити особливості роботи сервісів доставки їжі та проаналізувати потреби користувачів;
- виконати аналіз аналогів вебдодатків та виявити їх сильні та слабкі сторони, які б можна було покращити;
- виконати структурно-функціональне моделювання для наочного представлення процесів взаємодії;
- виконати розробку компонентів вебдодатку на основі потреб клієнтів;
- виконати тестування вебдодатку та перевірити працездатність додатку

## 2 ПРОЕКТУВАННЯ WEB-ПЛАТФОРМИ

### 2.1 Структурно-функціональне моделювання

IDEF0 – це метод, який виник у ВПС США в 1970-х роках як спосіб документування та комунікації складних систем. Він базується на мові функціонального моделювання SADT (Structured Analysis and Design Technique), яка використовує рамки та стрілки для опису діяльності та потоків даних системи.

IDEF0 працює шляхом декомпозиції системи на чотири рівні абстракції: контекст, декомпозиція, деталізація та реалізація. Кожен рівень складається з діаграм, які показують функції та взаємодію системи або її компонентів. На діаграмах використовуються чотири типи елементів: функції, входи, виходи та елементи керування. Функції представлені блоками, які мають назву та номер. Входи – це дані або ресурси, які надходять до функції ззовні. Виходи - це дані або результати, які виходять з функції назовні. Елементи керування – це умови або обмеження, які впливають на те, як функція працює. Входи, виходи та елементи керування зображені стрілками, які з'єднуються з боковими сторонами клітинок. Стрілки мають підписи, які описують їхній зміст або джерела [15].

На рисунку 2.1 представлено контекстну діаграму в нотації IDEF0 для вебдодатку сервісу замовлень доставки з ресторанів.

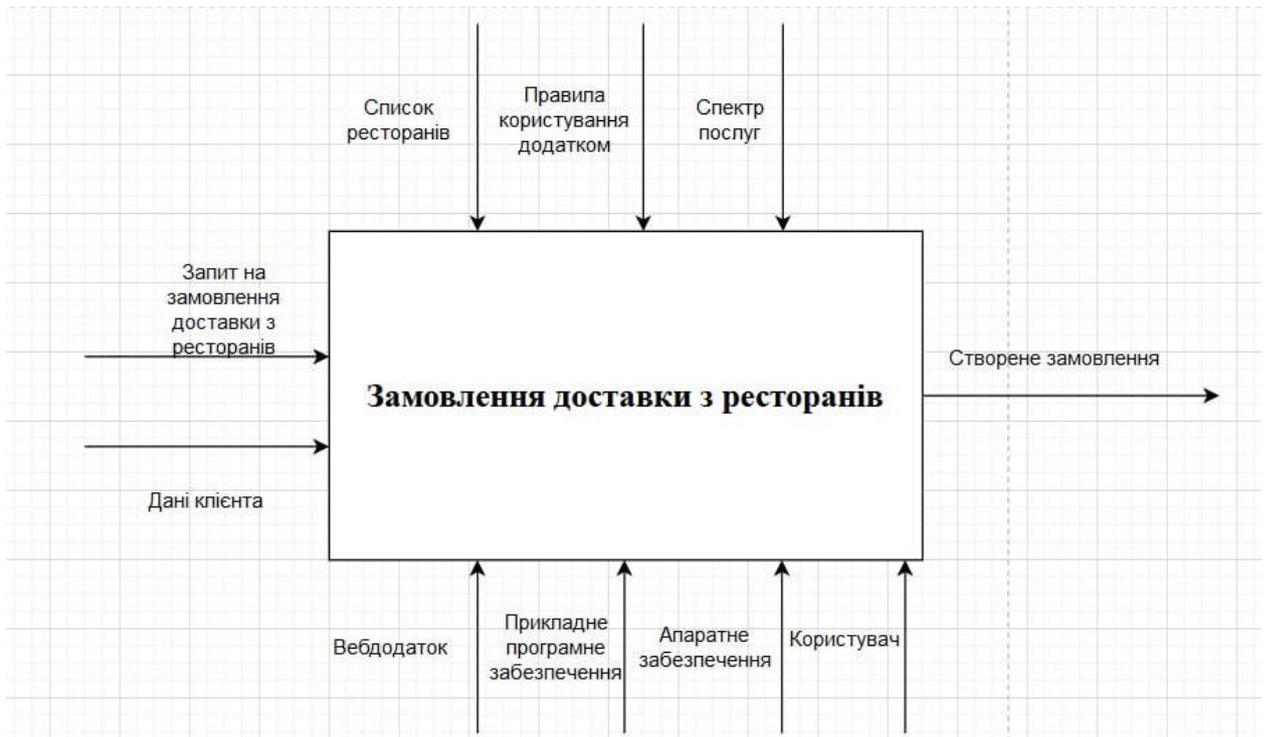


Рисунок 2.1 – Контекстна діаграма в нотації IDEF0 для вебдодатку сервісу замовлень доставки з ресторанів.

*Джерело: розроблено автором*

Функціональна декомпозиція відповідає різним функціональним взаємозв'язкам, як було розроблено початкову складну бізнес-функцію. Вона в основному фокусується на тому, як розробляється загальна функціональність та її взаємодія між різними компонентами.

Великі або складні функціональні можливості легше зрозуміти, якщо розбити їх на частини за допомогою функціональної декомпозиції [16].

На рисунку 2.2 представлено декомпозицію IDEF0-діаграми для вебдодатку сервісу замовлень доставки з ресторанів.

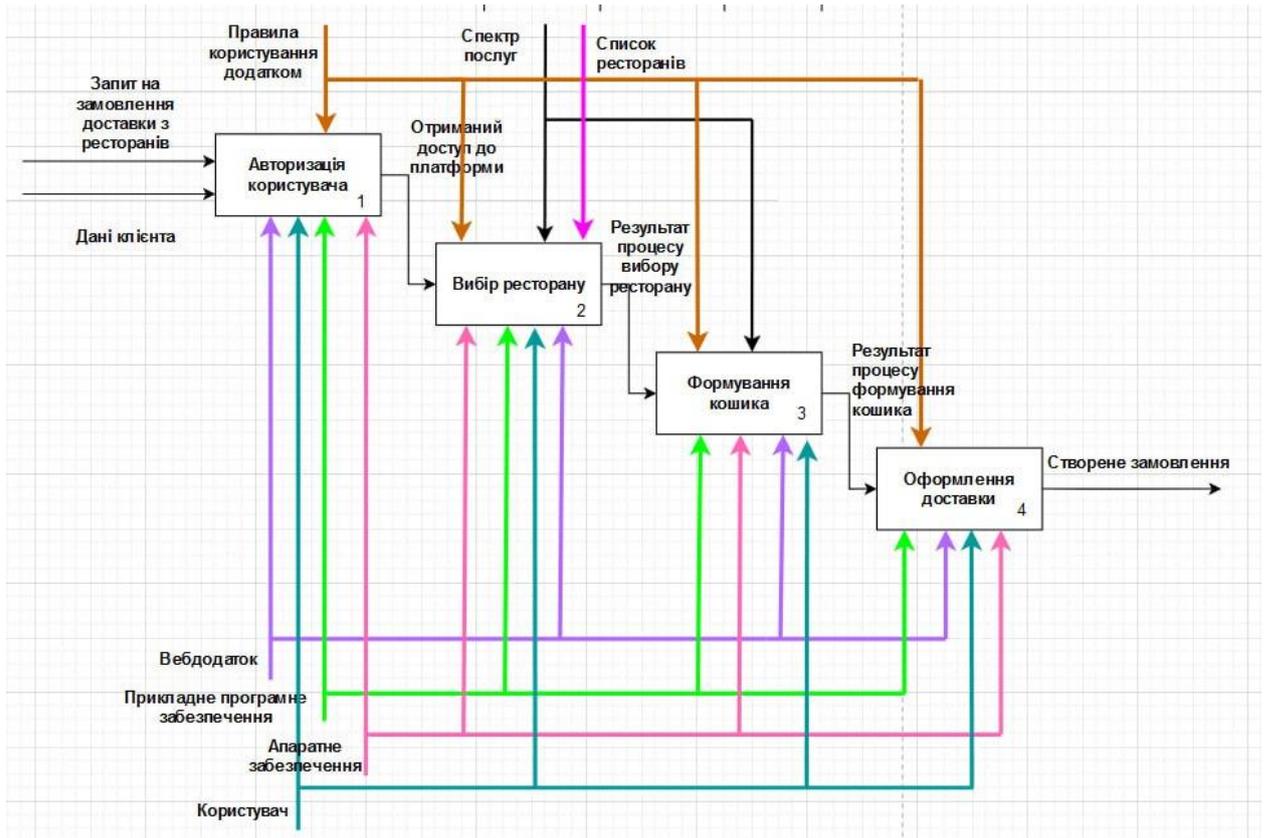


Рисунок 2.2 - декомпозиція IDEF0-діаграми для вебдодатку сервісу замовлень доставки з ресторанів

*Джерело: розроблено автором*

## 2.2 Моделювання варіантів використання

Діаграма варіантів використання складається з системи, пов'язаних з нею варіантів використання та акторів і пов'язує їх один з одним, щоб візуалізувати: що описується? (система), хто використовує систему? (актори) і чого хочуть досягти актори? (варіанти використання), таким чином, варіанти використання допомагають забезпечити розробку правильної системи, фіксуючи вимоги з точки зору користувача.

Варіанти використання визначають взаємодію між зовнішніми акторами і системою для досягнення певних цілей. Діаграма варіантів використання містить три основні компоненти.

Актори – це, як правило, особи, які взаємодіють з системою відповідно до їхніх ролей. Актором може бути людина або інша зовнішня система.

Варіант використання описує, як актори використовують систему для досягнення певної мети. Варіанти використання, як правило, ініціюються користувачем для досягнення цілей, що описують дії та варіанти, пов'язані з досягненням мети[17].

Взаємовідносини – відносини між акторами та варіантами використання.

Діаграма варіантів використання вебдодатку сервісу замовлень доставки з ресторанів представлена на рисунку 2.3.

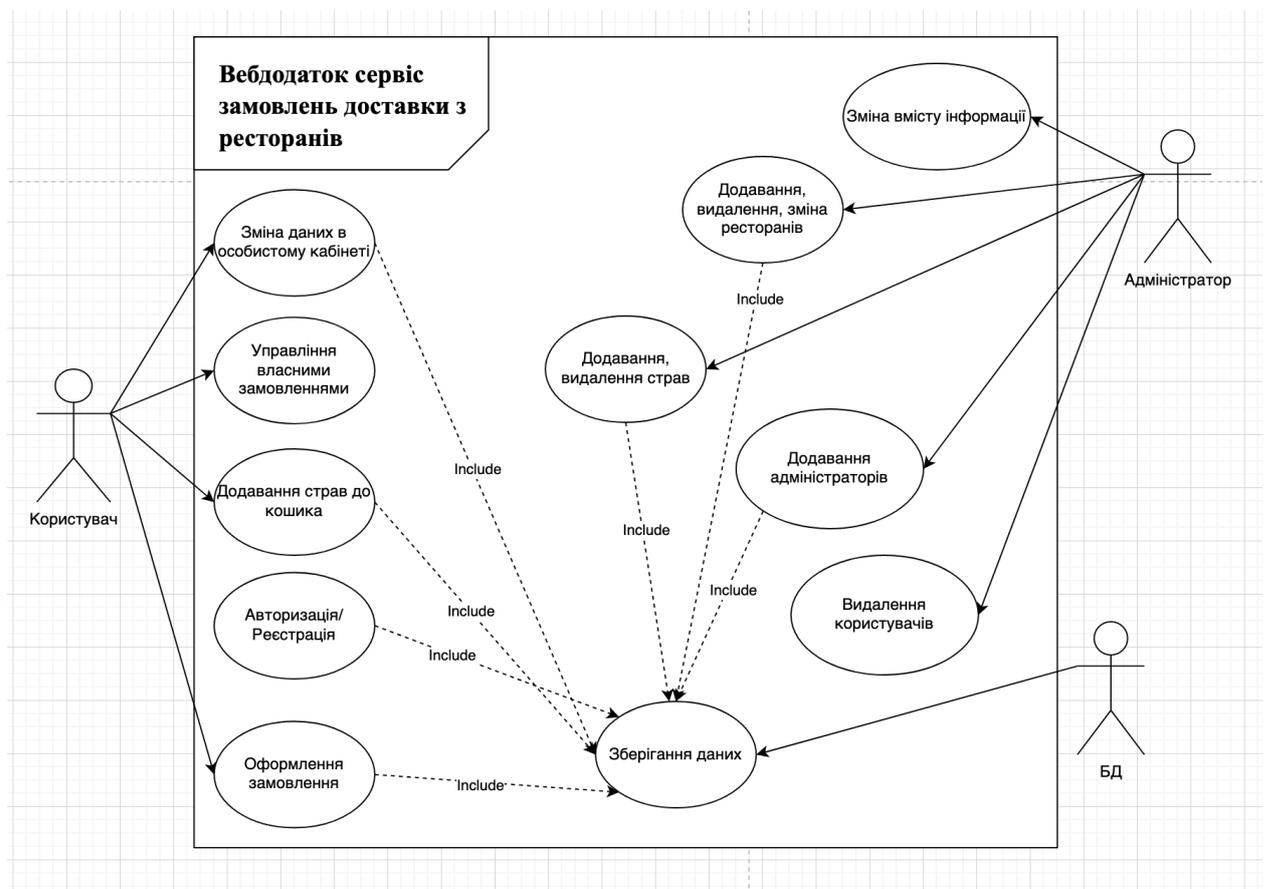


Рисунок 2.3 - Діаграма варіантів використання вебдодатку сервісу замовлень доставки з ресторанів

Джерело: розроблено автором

## 2.3 Проектування моделі бази даних

Як провідна транзакційна СУБД, MySQL має забезпечувати повну підтримку атомарних, послідовних, ізольованих, довговічних і багатоверсійних транзакцій, а також необмежене блокування на рівні рядків. Вибір MySQL – це надійне рішення для забезпечення цілісності даних з унікальними функціями, які дозволяють миттєво ідентифікувати ситуації завдяки посилянню на цілісність посилань, що забезпечується сервером [19].

На рисунку 2.4 представлено схему логічної моделі бази даних вебдодатку сервісу замовлень доставки з ресторанів яка була реалізована в додатку phpMyAdmin.

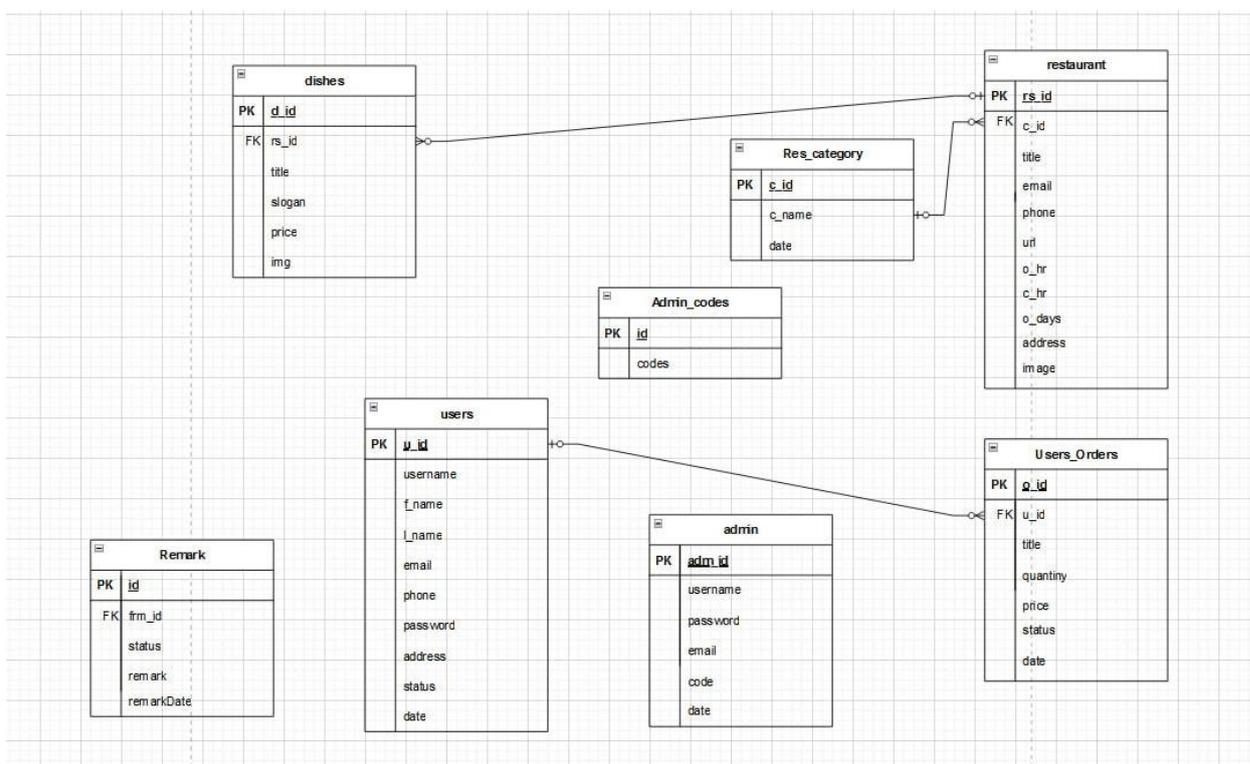


Рисунок 2.4 - Схему логічної моделі бази даних вебдодатку сервісу замовлень доставки з ресторанів

Джерело: розроблено автором

## 3. РЕАЛІЗАЦІЯ ВЕБДОДАТКУ СЕРВІСУ ЗАМОВЛЕНЬ ДОСТАВКИ З РЕСТОРАНІВ

### 3.1 Реалізація бази даних

Для реалізації бази даних було використано безкоштовний вбудований інструмент в локальний сервер OpenServer – phpMyAdmin.

На рисунках 3.1 – 3.8 представлено вигляд реалізованих таблиць з атрибутами для вебдодатку сервісу замовлень доставки з ресторанів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	<b>adm_id</b>			Ні	Немає		AUTO_INCREMENT	<a href="#">Більше</a>
<input type="checkbox"/>	2	<b>username</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	3	<b>password</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	4	<b>email</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	5	<b>code</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	6	<b>date</b>	timestamp	on update CURRENT_TIMESTAMP	Ні	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	<a href="#">Більше</a>

Рисунок 3.1 – Вигляд таблиці «admin»

Джерело: розроблено автором

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	<b>id</b>			Ні	Немає		AUTO_INCREMENT	<a href="#">Більше</a>
<input type="checkbox"/>	2	<b>codes</b>	varchar(6)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>

Рисунок 3.2 – Вигляд таблиці «admin\_codes»

Джерело: розроблено автором

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	<b>d_id</b>			Ні	Немає		AUTO_INCREMENT	<a href="#">Більше</a>
<input type="checkbox"/>	2	<b>rs_id</b>			Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	3	<b>title</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	4	<b>slogan</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	5	<b>price</b>	decimal(10,2)		Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	6	<b>img</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>

Рисунок 3.3 – Вигляд таблиці «dishes»

Джерело: розроблено автором

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	<b>id</b>			Ні	Немає		AUTO_INCREMENT	<a href="#">Більше ▾</a>
<input type="checkbox"/>	2	<b>frm_id</b>			Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	3	<b>status</b>	varchar(255)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	4	<b>remark</b>	mediumtext	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	5	<b>remarkDate</b>	timestamp		Ні	CURRENT_TIMESTAMP		DEFAULT_GENERATED	<a href="#">Більше ▾</a>

Рисунок 3.4 – Вигляд таблиці «remark»

*Джерело: розроблено автором*

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	<b>rs_id</b>			Ні	Немає		AUTO_INCREMENT	<a href="#">Більше ▾</a>
<input type="checkbox"/>	2	<b>c_id</b>			Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	3	<b>title</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	4	<b>email</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	5	<b>phone</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	6	<b>url</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	7	<b>o_hr</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	8	<b>c_hr</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	9	<b>o_days</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	10	<b>address</b>	text	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	11	<b>image</b>	text	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	12	<b>date</b>	timestamp		Ні	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	<a href="#">Більше ▾</a>

Рисунок 3.5 – Вигляд таблиці «restaurant»

*Джерело: розроблено автором*

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	<b>c_id</b>			Ні	Немає		AUTO_INCREMENT	<a href="#">Більше ▾</a>
<input type="checkbox"/>	2	<b>c_name</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше ▾</a>
<input type="checkbox"/>	3	<b>date</b>	timestamp		Ні	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	<a href="#">Більше ▾</a>

Рисунок 3.6 – Вигляд таблиці «res\_category»

*Джерело: розроблено автором*

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	<b>u_id</b>			Ні	Немає		AUTO_INCREMENT	<a href="#">Більше</a>
<input type="checkbox"/>	2	<b>username</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	3	<b>f_name</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	4	<b>l_name</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	5	<b>email</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	6	<b>phone</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	7	<b>password</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	8	<b>address</b>	text	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	9	<b>status</b>	int		Ні	1			<a href="#">Більше</a>
<input type="checkbox"/>	10	<b>date</b>	timestamp		Ні	CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP	DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	<a href="#">Більше</a>

Рисунок 3.7 – Вигляд таблиці «users»

*Джерело: розроблено автором*

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	<b>o_id</b>			Ні	Немає		AUTO_INCREMENT	<a href="#">Більше</a>
<input type="checkbox"/>	2	<b>u_id</b>			Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	3	<b>title</b>	varchar(222)	latin1_swedish_ci	Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	4	<b>quantity</b>	int		Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	5	<b>price</b>	decimal(10,2)		Ні	Немає			<a href="#">Більше</a>
<input type="checkbox"/>	6	<b>status</b>	varchar(222)	latin1_swedish_ci	Так	NULL			<a href="#">Більше</a>
<input type="checkbox"/>	7	<b>date</b>	timestamp		Ні	CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP	DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	<a href="#">Більше</a>

Рисунок 3.8 – Вигляд таблиці «users\_orders»

*Джерело: розроблено автором*

Для взаємодії вебдодатку з базою даних було створено файл connect.php, який створює з'єднання з MySQL та перевіряє його успішність.

На рисунку 3.9 представлено вигляд реалізації коду connection.php.



## 3.2 Реалізація серверної та клієнтської частини

Для початку було реалізовано логіку авторизації та реєстрації користувача. При відправці заповненої форми, PHP-скрипт проводить валідацію на момент наявності введених користувачем даних в базі даних.

Якщо дані були введені коректно, то користувач перенаправляється на головну сторінку (index.php). В іншому випадку, користувач отримає повідомлення про помилку.

На рисунку 3.11 представлено процес реалізації логіки авторизації.

```

21 </head>
22
23 <body>
24 <?php
25
26 include("connection/connect.php"); //INCLUDE CONNECTION
27 error_reporting(0); // Hide unuseful index errors
28 session_start(); // Temp sessions
29 if(isset($_POST['submit'])) // If button is submit
30 {
31     $username = $_POST['username']; //fetch records from login form
32     $password = $_POST['password'];
33
34     if(!empty($_POST['submit'])) // If records were not empty
35     {
36         $loginquery = "SELECT * FROM users WHERE username='$username' && password='".md5($password)."'"; //selecting matching records
37         $result=mysql_query($db, $loginquery); //executing
38         $row=mysql_fetch_array($result);
39
40         if(is_array($row)) // if matching records in the array & if everything is right
41         {
42             $_SESSION["user_id"] = $row["u_id"]; // put user id into temp session
43             header("refresh:1;url=index.php"); // redirect to index.php page
44         }
45         else
46         {
47             $message = "Invalid Username or Password!"; // throw error
48         }
49     }
50 }
51
52 }
53 ?>
54
55 <!-- Form Action -->
56 <!-- Input Field -->
57 <!-- Button Field -->
58 <!-- Page Title -->
59 <div class="page-title">
60 <h1>Login Form</h1>
61 </div>
  
```

Рисунок 3.11 – Процес реалізації логіки авторизації

*Джерело: розроблено автором*

За схожою логікою було створено логіку для реєстрації користувача з валідацією даних. На рисунку 3.12 представлено процес реалізації бізнес-логіки для сторінки реєстрації. В додатку В представлені повні лістинги кодів login.php та register.php.

```

connect.php
online_restapi
index.php
registration.php
all_menu.php

Project
  delete_menu.php
  delete_orders.php
  delete_stores.php
  delete_users.php
  index.php
  logout.php
  order_update.php
  update_category.php
  update_menu.php
  update_restraint.php
  update_users.php
  userprofile.php
  view_order.php
  connection
  css
  fonts
  images
  js
  scss
  SQL
  checkout.php
  delete_orders.php
  dishes.php
  index.php
  login.php
  logout.php
  product-action.php
  registration.php
  restaurants.php
  your_orders.php

8  if(isset($_POST['submit'])) //if submit btn is pressed
9  {
10     if(empty($_POST['firstname']) || //fetching and find if its empty
11        empty($_POST['lastname']))
12        empty($_POST['email']) ||
13        empty($_POST['phone']) ||
14        empty($_POST['password']) ||
15        empty($_POST['cpassword']) ||
16        empty($_POST['cpassword']))
17     {
18        $message = "All fields must be Required!";
19     }
20     else
21     {
22        //checking username & email if already present
23        $check_username= mysql_query($db, "SELECT username FROM users where username = '".$_POST['username']."' ");
24        $check_email = mysql_query($db, "SELECT email FROM users where email = '".$_POST['email']."' ");
25
26
27
28        if($_POST['password'] != $_POST['cpassword']){ //matching passwords
29            $message = "Password not match!";
30        }
31        elseif(strlen($_POST['password']) < 6) //col password length
32        {
33            $message = "Password Must be >=6";
34        }
35        elseif(strlen($_POST['phone']) < 10) //col phone length
36        {
37            $message = "Invalid phone number!";
38        }
39
40        elseif (!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) // Validate email address
41        {
42            $message = "Invalid email address please type a valid email!";
43        }
44        elseif(mysql_num_rows($check_username) > 0) //check username
45        {
46            $message = 'username Already exists!';
47        }
48    }
49 }

```

Рисунок 3.12 – Процес реалізації бізнес-логіки для сторінки реєстрації

*Джерело: розроблено автором*

Також для кожної сторінки було надано стилізацію за допомогою CSS та Boostrap. На рисунку 3.13 представлено частину реалізації стилів для вебдодатку сервісу доставки їжі з ресторанів.

```

connect.php
online_restapi
index.php
login.css
all_menu.php

Project
  delete_menu.php
  delete_orders.php
  delete_stores.php
  delete_users.php
  index.php
  logout.php
  order_update.php
  update_category.php
  update_menu.php
  update_restraint.php
  update_users.php
  userprofile.php
  view_order.php
  connection
  css
  fonts
  images
  js
  scss
  SQL
  checkout.php
  delete_orders.php
  dishes.php
  index.php
  login.php
  logout.php
  product-action.php
  registration.php
  restaurants.php
  your_orders.php

1 body {
2     background: #e6e6fa;
3     color: #000000;
4     font-family: 'RobotoDraft', 'Roboto', sans-serif;
5     font-size: 16px;
6     -webkit-font-smoothing: antialiased;
7     text-decoration: none;
8 }
9
10 // Font Family
11 .open-title {
12     padding: 5px 0;
13     text-align: center;
14     letter-spacing: 2px;
15 }
16 .open-title h1 {
17     margin: 0 0 20px;
18     font-size: 18px;
19     font-weight: bold;
20 }
21 .open-title span {
22     font-size: 12px;
23 }
24 .open-title span .fa {
25     color: #f39c12;
26 }
27 .open-title span a {
28     color: #f39c12;
29     font-weight: bold;
30     text-decoration: none;
31 }
32
33 // Form Wrapper
34 .form-wrapper {
35     position: relative;
36     background: #ffffff;
37     min-width: 100px;
38     width: 100%;
39     border-top: 1px solid #f39c12;
40     box-shadow: 0 0 1px rgba(0, 0, 0, 0.1);
41     margin: 0 auto;
42 }

```

Рисунок 3.13 – Результат реалізації стилів для вебдодатку

*Джерело: розроблено автором*

Для відображення ресторанів з яких можна замовити доставку їжі було створено файл `restaurants.php`.

Вебсторінка містить базову структуру HTML, включаючи мета-теги, посилання на зовнішні файли CSS і файли JavaScript. PHP-скрипти починаються з підключення до бази даних і містять файли для управління сеансом. У верхній частині сторінки знаходиться панель навігації з посиланнями на головну сторінку, розділ ресторанів та варіанти входу і реєстрації.

На сторінці використовуються різні бібліотеки JavaScript для таких функцій, як повзунки, фільтрація за ізотопами та навігація.

На рисунку 3.14 представлено вигляд процесу реалізації коду `restaurants.php`.

```

45 <li class="nav-item"><a href="your_orders.php" class="nav-link active">your orders</li>;
46 <li class="nav-item"><a href="logout.php" class="nav-link active">logout</li>;
47 </li>
48 </ul>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>

```

Рисунок 3.14 – Процес реалізації коду `restaurants.php`

*Джерело: розроблено автором*

Для відображення замовлень користувача було створено сторінку `your_orders.php`. Процес реалізації представлено на рисунку 3.15. Повний лістинг коду `your_orders.php` представлено в додатку Г.



*Джерело: розроблено автором*

### 3.3 Настанови з використання

Для початку користування вебдодатком користувач має можливість виконати операцію авторизації, щоб отримати доступ до функцій замовлення доставки та вибору ресторану.

Якщо користувач не має облікового запису в вебдодатку, то йому необхідно виконати операцію реєстрації. На рисунках 3.17 – 3.18 представлено вигляд сторінок авторизації та реєстрації.

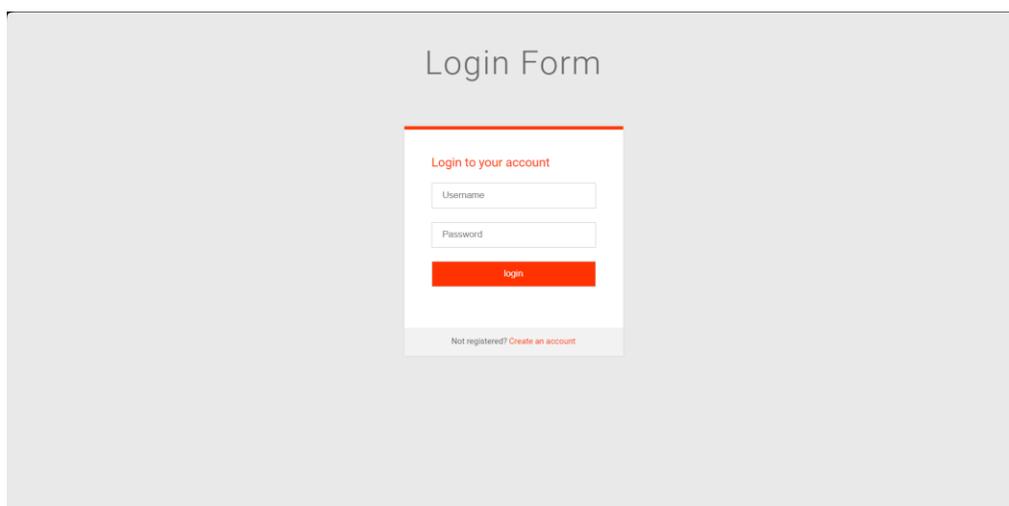
The image shows a login form titled "Login Form" centered on a light gray background. The form itself is a white rectangle with a thin red border. At the top of the form, it says "Login to your account" in red. Below this are two input fields: "Username" and "Password". Under the password field is a red "login" button. At the bottom of the form, there is a link that says "Not registered? Create an account" in red.

Рисунок 3.17 – Вигляд сторінки авторизації

*Джерело: розроблено автором*

Рисунок 3.18 – Видгляд сторінки реєстрації

*Джерело: розроблено автором*

Після того, як користувач успішно виконав авторизацію чи реєстрацію, він потрапляє на головну сторінку вебдодатку. На рисунках 3.19 – 3.22 представлено вигляд головної сторінки вебдодатку.

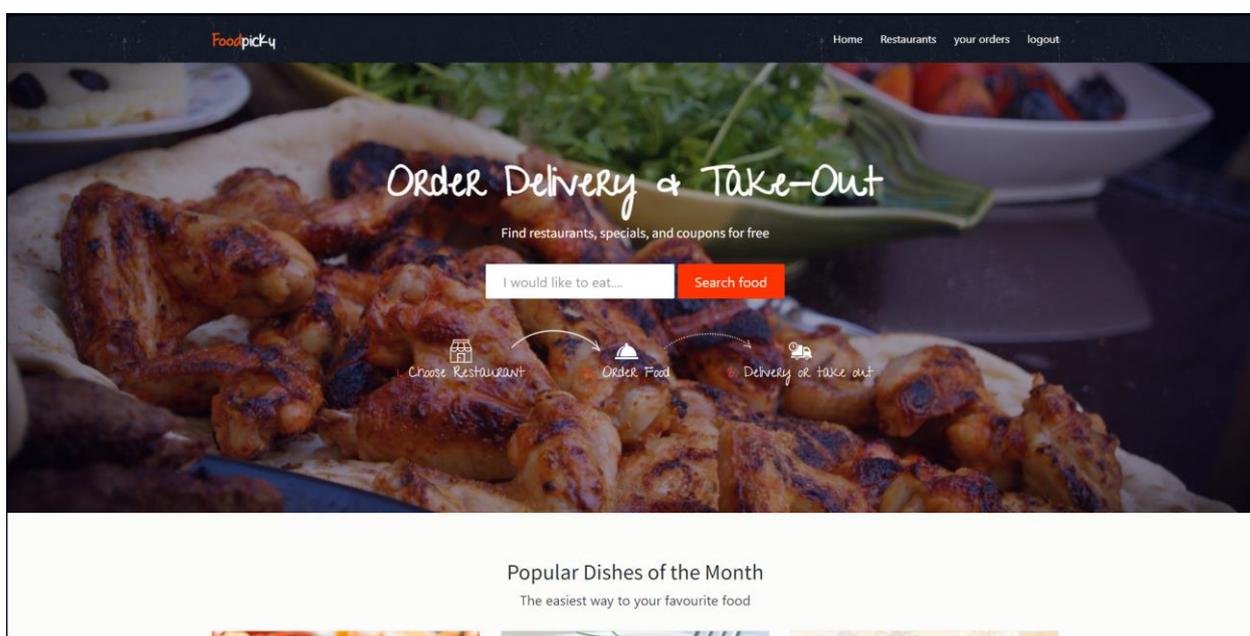


Рисунок 3.19 – Головна сторінка вебдодатку - частина 1

*Джерело: розроблено автором*

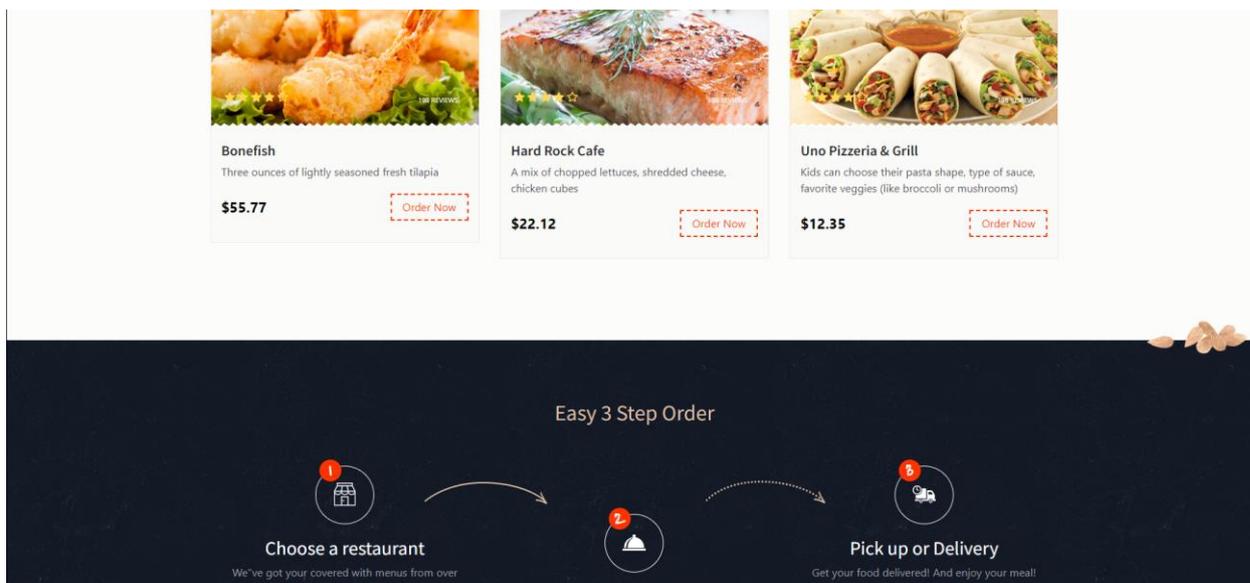


Рисунок 3.20 – Головна сторінка вебдодатку - частина 2

Джерело: розроблено автором

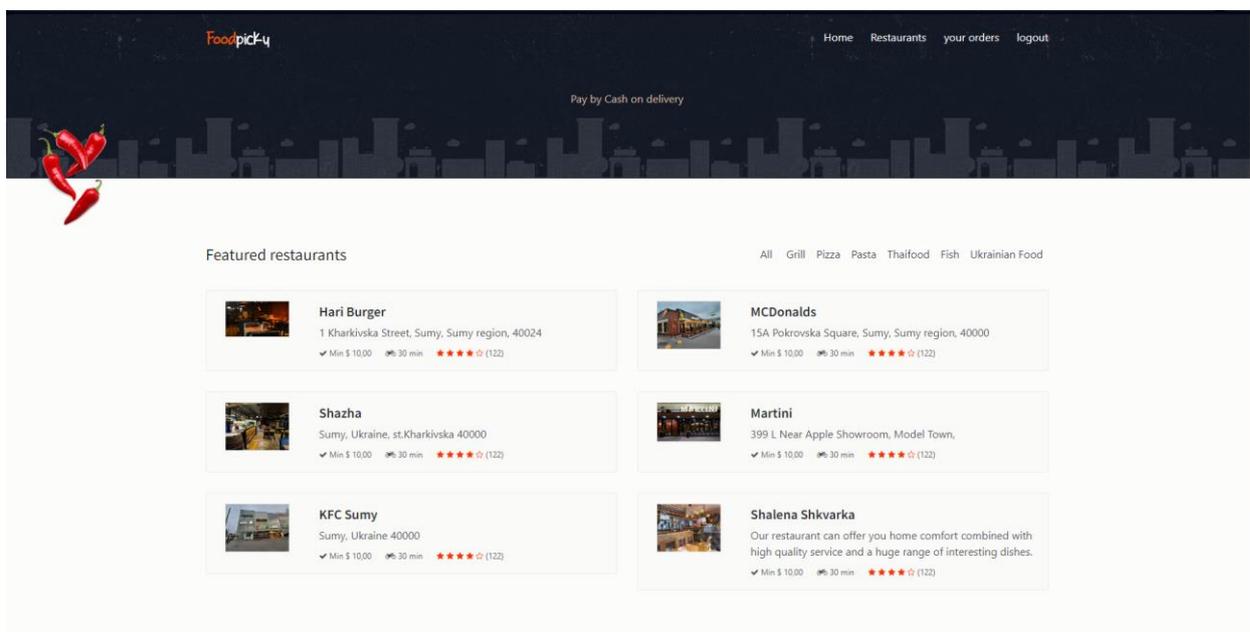


Рисунок 3.21 – Головна сторінка вебдодатку - частина 3

Джерело: розроблено автором

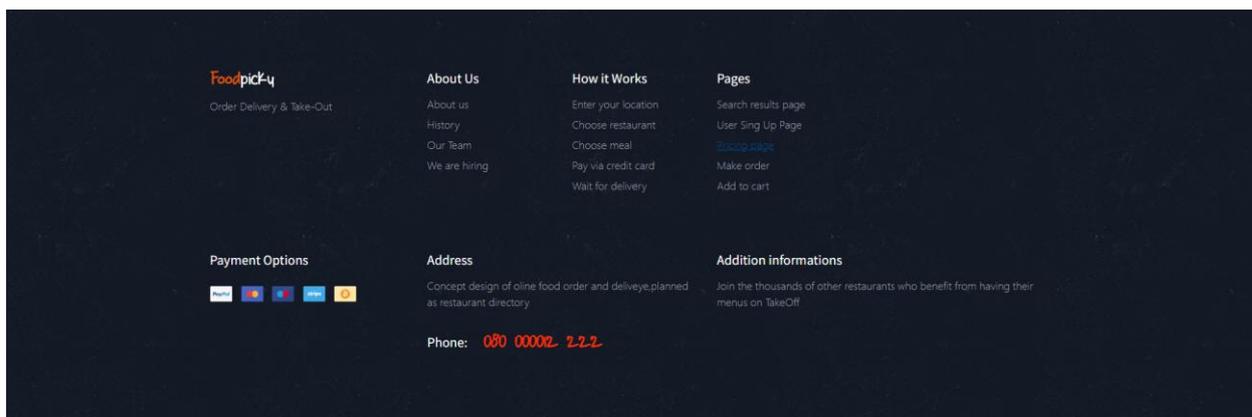


Рисунок 3.22 – Головна сторінка вебдодатку - частина 4

*Джерело: розроблено автором*

Для більш зручного та швидкого пошуку на головній сторінці присутня функція фільтрації. На рисунку 3.23 представлено результат фільтрації ресторанів за типом страв.

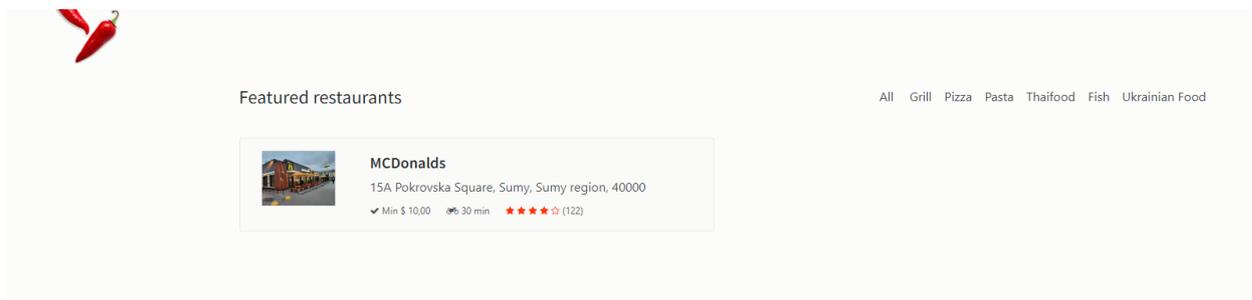


Рисунок 3.23 – Результат фільтрації ресторанів

*Джерело: розроблено автором*

Кожний ресторан має сторінку з інформацією про діяльність та меню страв. За допомогою такого підходу до представлення інформації користувач може детально обрати ресторан який йому до вподоби та замовити доставку.

На рисунках 3.24 – 3.25 представлено вигляд сторінку з наявними ресторанами та сторінку обраного ресторану.

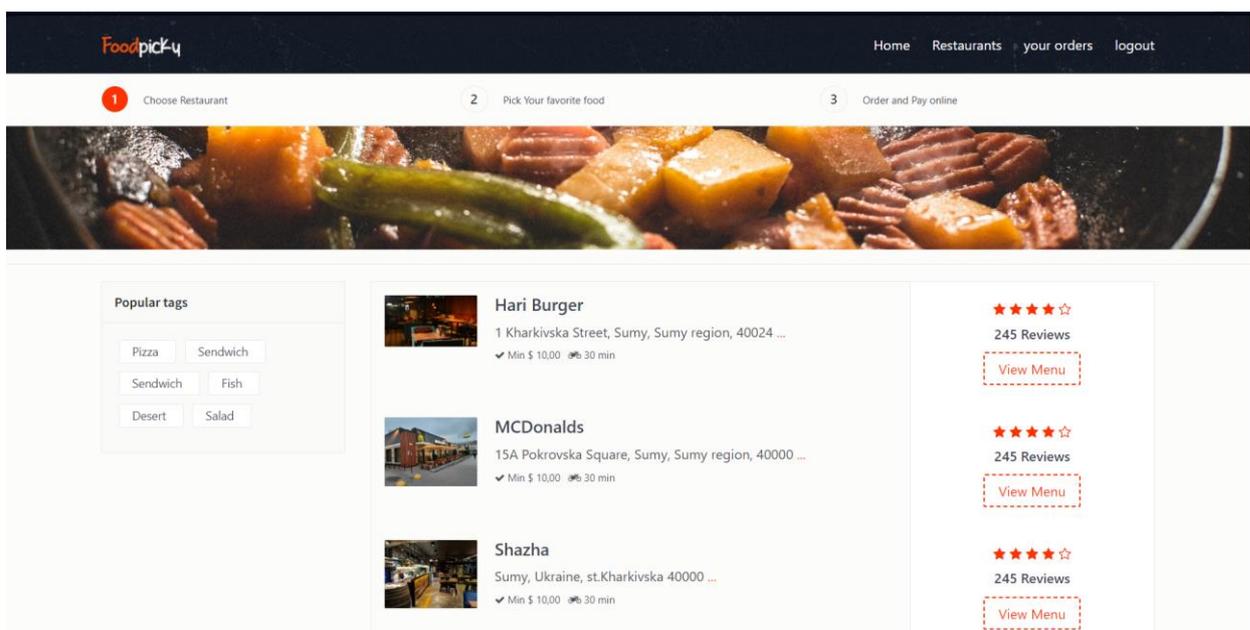


Рисунок 3.24 – Сторінка з наявними ресторанами

*Джерело: розроблено автором*

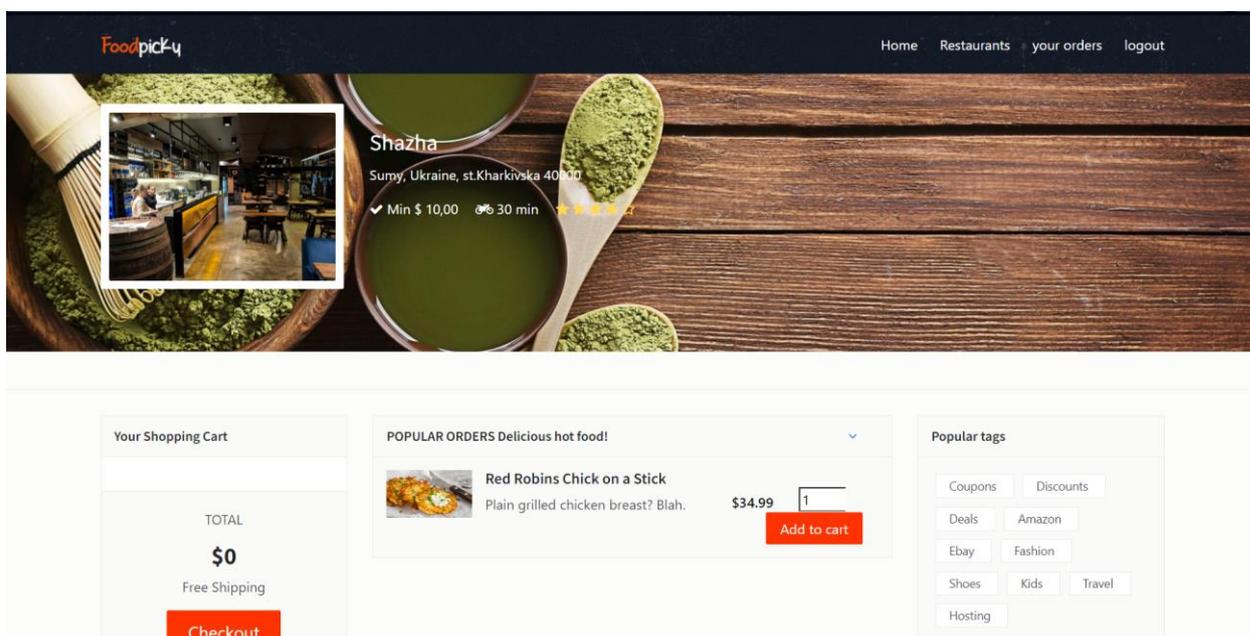


Рисунок 3.25 – Сторінка обраного ресторану

*Джерело: розроблено автором*

Під час вибору їжі, користувач може формувати кошик та отримувати інформацію про суму замовлення. На рисунку 3.26 представлено вигляд кошика замовлень.

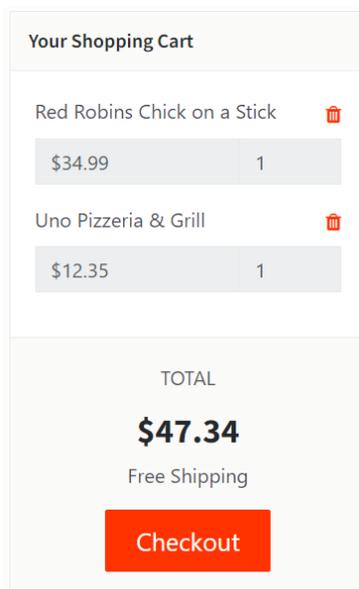


Рисунок 3.26 – Вигляд кошика замовлень

*Джерело: розроблено автором*

На сторінці оформлення замовлення користувач може обрати варіант оплати доставки (рис. 3.27).

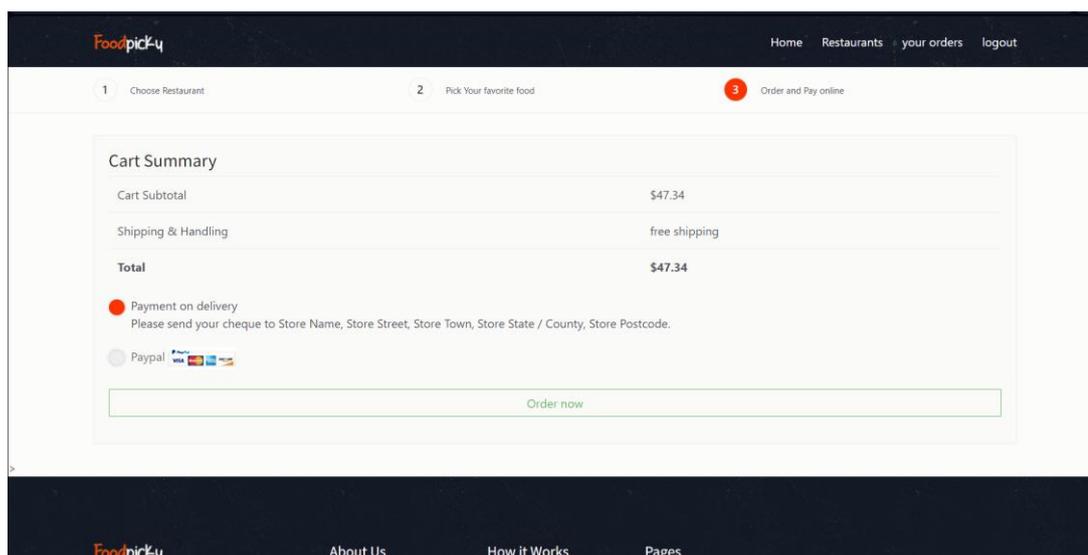


Рисунок 3.27 – Сторінка оформлення замовлення

*Джерело: розроблено автором*

Після того, як замовлення сформовано, користувач може перевірити власне замовлення на сторінці особистих замовлень та виконати операцію відмови від доставки за потреби. На рисунку 3.28 представлено вигляд сторінки особистих замовлень.

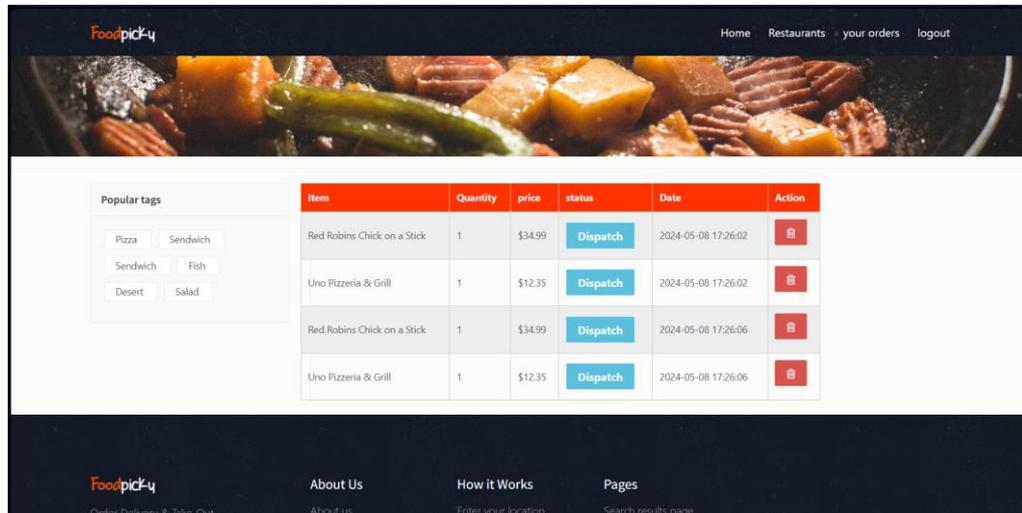


Рисунок 3.28 – Вигляд сторінки особистих замовлень

*Джерело: розроблено автором*

Якщо користувач має права адміністратора, то йому доступна до використання панель адміністратора. Дана панель дозволяє виконувати CRUD операції з контентом вебдодатку.

Для авторизації користувач повинен ввести унікальні дані в форму. При успішному процесі, користувач автоматично буде переадресований на головну сторінку адмін панелі. На рисунку 3.29 представлено вигляд головної сторінки адміністративної панелі.

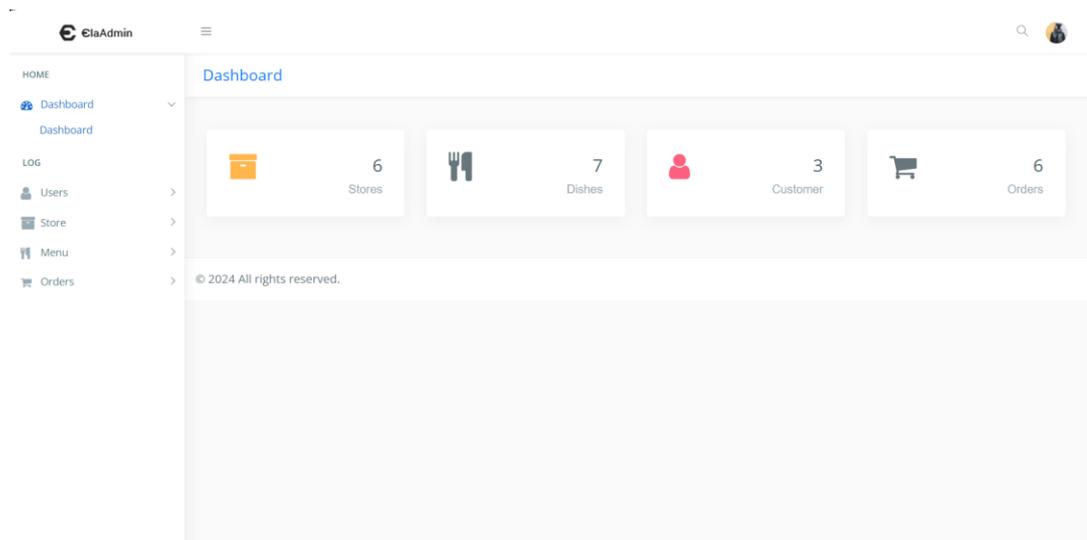


Рисунок 3.29 – Головна сторінка адмін панелі

*Джерело: розроблено автором*

Адміністратор може переглянути інформацію про користувачів, які мають облікові записи в вебдодатку, а також додати нових користувачів. На рисунках 3.30 – 3.31 представлено вигляд інформації про наявних користувачів та форма додавання нових користувачів.

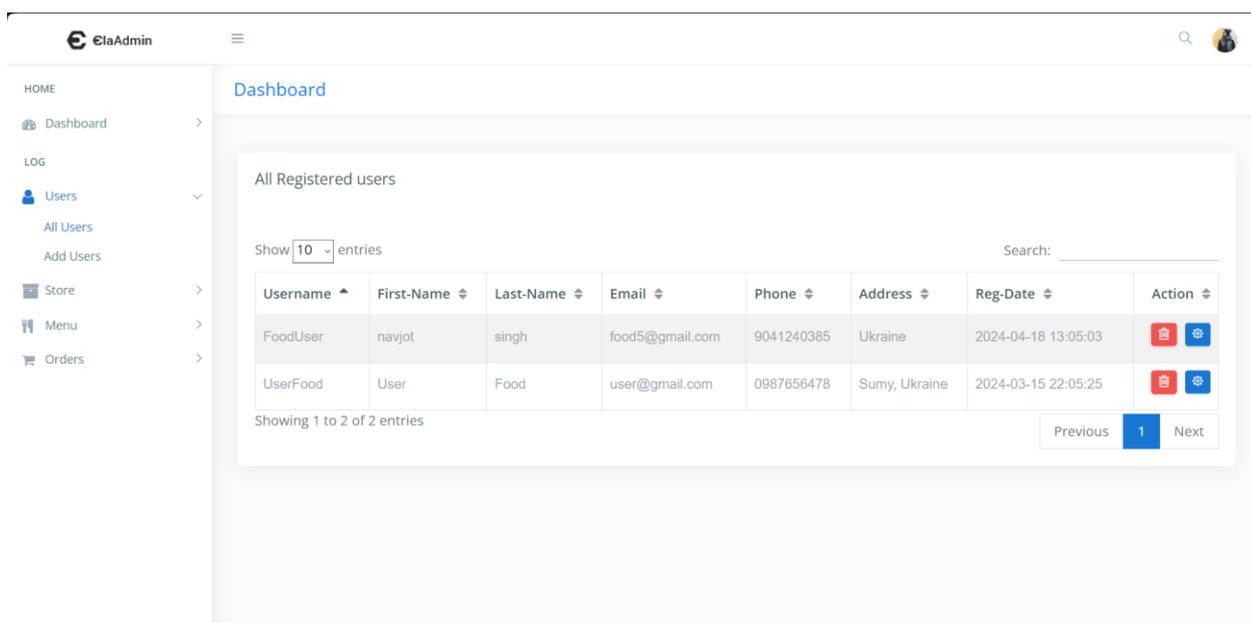
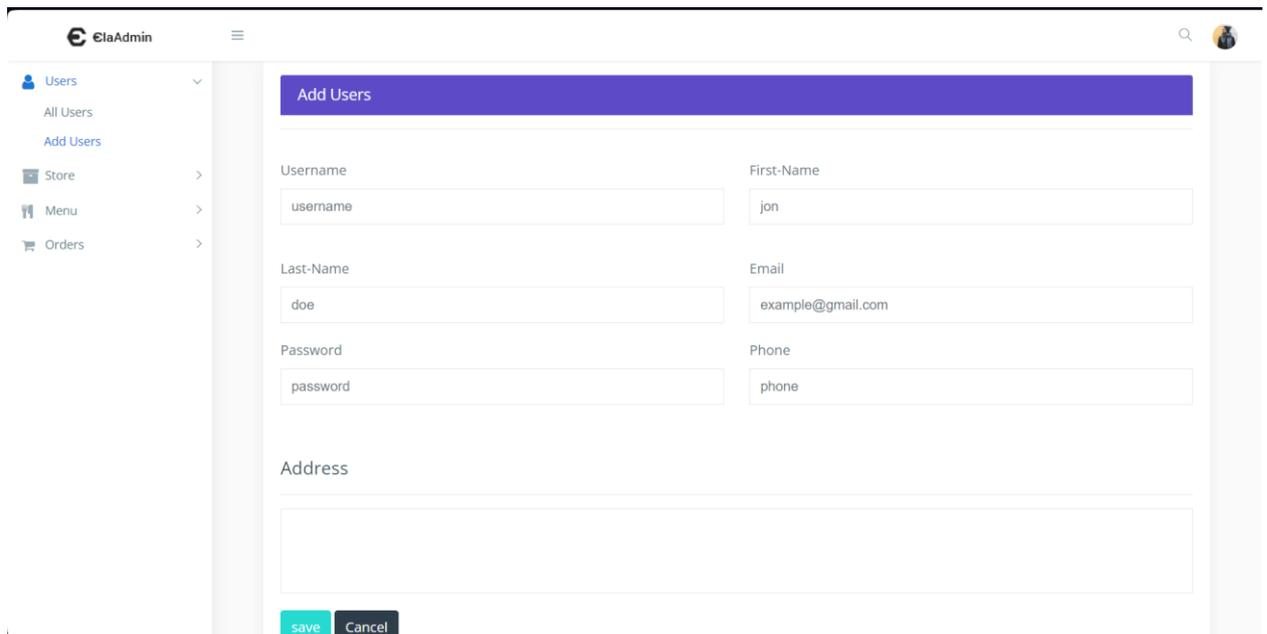


Рисунок 3.30 – Вигляд інформації про наявних користувачів

*Джерело: розроблено автором*



The screenshot shows the 'Add Users' form in the ElaAdmin dashboard. The form is titled 'Add Users' and is located in the main content area. On the left side, there is a sidebar menu with the following items: 'Users' (selected), 'All Users', 'Add Users', 'Store', 'Menu', and 'Orders'. The form itself contains the following fields:

Username	First-Name
username	jon
Last-Name	Email
doe	example@gmail.com
Password	Phone
password	phone
Address	

At the bottom of the form, there are two buttons: 'save' and 'Cancel'.

Рисунок 3.31 – Вигляд форми додавання нових користувачів

*Джерело: розроблено автором*

Також адміністратор може перевірити інформацію про наявні ресторани, а також додати категорію та ресторан за потреби. На рисунках 3.32 – 3.34 представлено вигляд сторінки про наявні ресторани, сторінку для додавання категорії та сторінку для додавання ресторану.

Export data to Copy, CSV, Excel, PDF & Print

Copy CSV Excel PDF Print

Search:

Cat	Store-Name	Email	Phone	Url	Open Hrs	Close Hrs	Open Days	Address	Store-Image	Date	Action
grill	MCDonalds	mcdon@gmail.com	0987456321	mcdon.com	8am	4pm	24hr-x7	15A Pokrovska Square, Sumy, Sumy region, 40000		2024-05-08 17:03:32	
pasta	Marlini	martin@gmail.com	3454345654	martin.com	8am	4pm	mon-thu	399 L Near Apple Showroom, Model Town,		2018-04-18 16:57:19	
pizza	Shazha	shazha@gmail.com	0956437894	shazha.com	7am	5pm	mon-thu	Sumy, Ukraine, st.Kharkivska 40000		2024-05-08 17:01:22	
pizza	Hari Burger	HariBurger@gmail.com	0984567342	HariBurger.com	9am	5pm	mon-thu	1 Kharkivska Street, Sumy, Sumy region, 40024		2024-05-08 17:05:04	
Ukrainian Food	Shalena Shkvarka	shalenashkvarka@gmail.com	0987656478	shalenashkvarka.com	6am	6pm	24hr-x7	Our restaurant can offer you home		2024-05-07	

Рисунок 3.32 – Сторінка з наявними ресторанами

*Джерело: розроблено автором*

Dashboard

Add Restaurant Category

Category

Category Name

save cancel

Listed Categories

ID#	Category Name	Date	Action
11	Ukrainian Food	2024-03-16 17:02:48	
9	FastFood	2024-05-08 17:53:46	
7	ChinaFood	2024-05-08 17:54:02	
6	Pizza	2024-05-08 17:54:18	
5	Grill	2024-05-08 17:54:26	

Рисунок 3.33 – Форма додавання категорії

*Джерело: розроблено автором*

The screenshot shows the 'Add Restaurant' form in the E!Admin dashboard. The form is titled 'Add Restaurant' and contains the following fields:

- Restaurant Name: John doe
- Business E-mail: example@gmail.com
- Phone: 1-(555)-555-5555
- website URL: http://example.com
- Open Hours: --Select your Hours--
- Close Hours: --Select your Hours--
- Open Days: --Select your Days--
- Image: Вибрати файл (File not selected)
- Select Category: --Select Category--
- Store Address: (empty field)

Рисунок 3.34 – Форма додавання ресторану

*Джерело: розроблено автором*

Також для наповнення інформації про ресторан доступна функція додавання меню для кожного ресторану. При потребі, адміністратор може переглянути доступні страви.

На рисунках 3.35 – 3.36 представлено інформацію про страви в наявності та форму для додавання нових страв.

The screenshot shows the 'All Menu data' page in the E!Admin dashboard. The page displays a table of menu items with the following columns: Restaurant, Dish-Name, Slogan, Price, Image, and Action. The table contains 7 entries.

Restaurant	Dish-Name	Slogan	Price	Image	Action
Hari Burger	Hard Rock Cafe	A mix of chopped lettuces, shredded cheese, chicken cubes	\$22.12		
Hari Burger	Bonefish	Three ounces of lightly seasoned fresh tilapia	\$55.77		
KFC Sunny	Houlihans Mini Cheeseburger	Creekstone Farms, where no antibiotics or growth hormones are used	\$22.55		
Martini	Lyfe Kitchens Tofu Taco	This chain, known for a wide selection of vegetarian and vegan choices	\$11.99		
McDonalds	Uno Pizzeria & Grill	Kids can choose their pasta shape, type of sauce, favorite veggies (like broccoli or mushrooms)	\$12.35		
Shalena Shkvarka	jdjmo	great taste great whatever	\$17.99		
Shazha	Red Robins Chick on a Stick	Plain grilled chicken breast? Blah.	\$34.99		

Showing 1 to 7 of 7 entries

Рисунок 3.35 – Сторінка з наявними стравами

*Джерело: розроблено автором*

The screenshot shows a web form titled "Add Menu to Restaurant" within a dashboard. The form contains the following elements:

- Dish Name:** A text input field containing "Mozzarella".
- About:** A text input field containing "slogan".
- price:** A text input field containing "\$".
- Image:** A file upload button labeled "Вибрати файл" (Choose file) with the text "Файл не вибрано" (File not selected).
- Select Category:** A dropdown menu currently showing "--Select Restaurant--".
- Buttons:** "save" (green) and "Cancel" (black) buttons at the bottom left.

Рисунок 3.36 – Форма для додавання страви

*Джерело: розроблено автором*

Для перевірки замовлень клієнтів доступна сторінка яка надає детальну інформацію про вподобання клієнтів з вмістом інформації про страви. На рисунку 3.37 представлено вигляд сторінки замовлень клієнтів.

The screenshot shows a table titled "All user Orders" with the following data:

Username	Title	Quantity	price	Address	status	Reg-Date	Action
FoodUser	jkimno	5	\$17.99	Ukraine	Delivered	2024-04-18 22:51:50	[Red] [Blue]
FoodUser	Red Robins Chick on a Stick	2	\$34.99	Ukraine	Dispatch	2024-04-18 22:52:34	[Red] [Blue]
UserFood	Red Robins Chick on a Stick	1	\$34.99	Sumy, Ukraine	Dispatch	2024-05-08 17:26:02	[Red] [Blue]
UserFood	Uno Pizzeria & Grill	1	\$12.35	Sumy, Ukraine	Dispatch	2024-05-08 17:26:02	[Red] [Blue]
UserFood	Red Robins Chick on a Stick	1	\$34.99	Sumy, Ukraine	Dispatch	2024-05-08 17:26:06	[Red] [Blue]
UserFood	Uno Pizzeria & Grill	1	\$12.35	Sumy, Ukraine	Dispatch	2024-05-08 17:26:06	[Red] [Blue]

Additional details from the screenshot: "Showing 1 to 6 of 6 entries", "Search:" field, and pagination buttons "Previous", "1", "Next".

Рисунок 3.37 – Сторінка замовлень клієнтів

*Джерело: розроблено автором*

## ВИСНОВКИ

В рамках виконання кваліфікаційної роботи бакалавра було проведено комплексне дослідження попиту на вебдодаток сервісу замовлень доставки з ресторанів. Крім того, були враховані сучасні тенденції споживання продуктів харчування. Дослідження включало аналіз пристроїв та платформ, які зазвичай використовуються клієнтами для замовлення їжі, а також їхні вподобання та очікування щодо функціональності вебдодатків.

У дослідженні розглядалися різні служби доставки їжі, включаючи їхні системи замовлення, взаємодію між клієнтами та ресторанами, а також обробку та виконання замовлень. Крім того, було проаналізовано аналоги вебдодатків, щоб визначити їхні переваги та недоліки. Даний аналіз допоміг отримати уявлення про те, як покращити користувацький досвід вебдодатку.

Після аналізу було виконано структурно – функціональне моделювання та моделювання варіантів використання, що дало змогу візуалізувати підпроцеси додатку та взаємодію між ними. Були розроблені інтерфейси для замовлення їжі, системи комунікації з представниками ресторану, зручні інструменти для оновлення та управління асортиментом страв.

Після виконання всіх вище описаних процесів, вебдодаток було протестовано, щоб переконатися в його функціональності та відповідності вимогам проекту. Було проведено кілька тестів, включаючи тести взаємодії, безпеки та продуктивності, щоб гарантувати якість та надійність додатку перед його впровадженням.

Впровадження вебдодатку сервісу замовлення доставки з ресторанів полегшить взаємодію між клієнтами та ресторанами, та забезпечить високий рівень задоволення клієнтів. Використання цього вебдодатка призведе до підвищення продуктивності ресторанів, кращого контролю над процесом доставки та покращення рівня задоволення клієнтів.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Дослідження Pew Research Center «About three-in-ten U.S. adults say they are ‘almost constantly’ online» Автор: Andrew Perrin, Sara Atske [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pewresearch.org/short-reads/2021/03/26/about-three-in-ten-u-s-adults-say-they-are-almost-constantly-online/> (дата звернення 24.03.2024).
2. Importance of Online Food Ordering System for Restaurants [Електронний ресурс] – Режим доступу до ресурсу: <https://www.fullestop.com/blog/importance-online-food-ordering-system-restaurants> (дата звернення 24.03.2024).
3. Top Reasons to Implement Online Food Delivery System Your Restaurant [Електронний ресурс] – Режим доступу до ресурсу: <https://www.foodiv.com/importance-of-online-food-delivery-system/> (дата звернення 24.03.2024).
4. The Importance of Online Food Ordering and Delivery for Restaurants [Електронний ресурс] – Режим доступу до ресурсу: <https://lavu.com/the-importance-of-online-ordering-and-delivery-for-restaurants/> (дата звернення 24.03.2024).
5. Reasons Your Restaurant Should Offer Online Ordering and Delivery [Електронний ресурс] – Режим доступу до ресурсу: <https://www.superorder.com/blog/reasons-for-online-ordering-delivery> (дата звернення 24.03.2024).
6. Advantages and Disadvantages of the Online Food Ordering System Автор: Bhaval Patel [Електронний ресурс] – Режим доступу до ресурсу: <https://www.spaceotechnologies.com/blog/online-food-ordering-system-advantages-disadvantages/> (дата звернення 24.03.2024).

7. Онлайн додаток для замовлення доставки їжі Mister.AM [Електронний ресурс] – Режим доступу до ресурсу: <https://misteram.com.ua/?city=sumy> (дата звернення 24.03.2024).
8. Онлайн додаток для замовлення доставки їжі IKURA [Електронний ресурс] – Режим доступу до ресурсу: <https://ikura.ua/sumy/uk> (дата звернення 24.03.2024).
9. Онлайн додаток для замовлення доставки їжі Glovo [Електронний ресурс] – Режим доступу до ресурсу: <https://glovoapp.com/ua/uk/sumi/> (дата звернення 24.03.2024).
10. Technology Stack for Web Application Development Автор: Gleb B. [Електронний ресурс] – Режим доступу до ресурсу: <https://rubygarage.org/blog/technology-stack-for-web-development> (дата звернення 24.03.2024).
11. Choosing a tech stack for the full-cycle web application development in 2023 Автор: Daria Bulatovych [Електронний ресурс] – Режим доступу до ресурсу: <https://yalantis.com/blog/tech-stack-for-web-app-development/> (дата звернення 24.03.2024).
12. PHP vs Java: The Ideal Programming Language in 2024 Автор: Ritwik Verma [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bacancytechnology.com/blog/php-vs-java> (дата звернення 24.03.2024).
13. The Best Tech Stack for Web App Development in 2024 [Електронний ресурс] – Режим доступу до ресурсу: <https://5ly.co/blog/best-web-app-tech-stack/> (дата звернення 24.03.2024).
14. Top 12 Best Databases for Web Application Development in 2024 [Електронний ресурс] – Режим доступу до ресурсу: <https://positiwise.com/blog/best-databases-for-web-application-development> (дата звернення 24.03.2024).
15. How can you use IDEF0 to model a system in Systems Design? [Електронний ресурс] – Режим доступу до ресурсу:

- <https://www.linkedin.com/advice/0/how-can-you-use-idef0-model-system-systems-design-9iiff> (дата звернення 24.03.2024).
16. Functional Decomposition Introduction [Електронний ресурс] – Режим доступу до ресурсу: [https://www.tutorialspoint.com/software\\_testing\\_dictionary/functional\\_decomposition.htm](https://www.tutorialspoint.com/software_testing_dictionary/functional_decomposition.htm) (дата звернення 24.03.2024).
17. Use Case Diagram Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://online.visual-paradigm.com/diagrams/tutorials/use-case-diagram-tutorial/> (дата звернення 24.03.2024).
18. Benefits of using MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://subscription.packtpub.com/book/data/9781788397186/1/ch01lv1sec13/benefits-of-using-mysql> (дата звернення 24.03.2024).
19. 8 Major Advantages of Using MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.datamation.com/storage/8-major-advantages-of-using-mysql/> (дата звернення 24.03.2024).
20. SMART project management: definitions and examples MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.indeed.com/career-advice/career-development/smart-project-management> (дата звернення 24.03.2024).
21. Work Breakdown Structure (WBS) [Електронний ресурс] – Режим доступу до ресурсу: [https://www.projectmanagement.com/wikis/397395/work-breakdown-structure--wbs-#\\_=#\\_](https://www.projectmanagement.com/wikis/397395/work-breakdown-structure--wbs-#_=#_) (дата звернення 24.03.2024).
22. Organizational Breakdown Structure (OBS) in Project Management [Електронний ресурс] – Режим доступу до ресурсу: <https://pmiuk.co.uk/unlocking-project-success-the-power-of-organizational-breakdown-structure-obs/> (дата звернення 24.03.2024).
23. Benefits of Gantt Chart Creation [Електронний ресурс] – Режим доступу до ресурсу: <https://teamboard.cloud/benefits-of-gantt-chart/> (дата звернення 24.03.2024).

24. What is risk management and why is it important? [Электронный ресурс] –  
Режим доступа до ресурсу:  
[https://www.techtarget.com/searchsecurity/definition/What-is-risk-  
management-and-why-is-it-important](https://www.techtarget.com/searchsecurity/definition/What-is-risk-management-and-why-is-it-important) (дата звернення 24.03.2024).

# ДОДАТОК А

## Технічне завдання на створення програмного продукту

«Вебдодаток сервісу замовлень доставки з ресторанів»

ПОГОДЖЕНО

Старший викладач кафедри  
інформаційних технологій

\_\_\_\_\_ доц. Неня А.В.

Студент групи ІТ-03-2

\_\_\_\_\_ Глущенко І.В.

2024

# **1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ ВЕБДОДАТКУ СЕРВІСУ ЗАМОВЛЕНЬ ДОСТАВКИ З РЕСТОРАНІВ**

## **1.1 Призначення вебдодатку сервісу замовлень доставки з ресторанів**

Вебдодаток призначений спростити процес замовлення доставки їжі з ресторанів, пропонуючи зручний та ефективний засіб комунікації між клієнтами та ресторанами.

## **1.2 Мета створення вебдодатку сервісу замовлень доставки з ресторанів**

Метою проекту є розробка надійного та зручного вебдодатку сервісу замовлень доставки з ресторанів для замовлення їжі з ресторанів, який забезпечить швидку та зручну доставку. Вебдодаток підвищить зручність та задовольнить потреби клієнтів у швидкій та якісній доставці їжі.

## **1.3 Цільова аудиторія**

Цільова аудиторія вебдодатку – клієнти, які цінують зручність і швидкість у замовленні та доставці їжі, а також менеджери ресторанів, які прагнуть покращити свою присутність в Інтернеті та забезпечити відмінний сервіс доставки для своїх клієнтів.

## **2 ВИМОГИ ДО ВЕБДОДАТКУ СЕРВІСУ ЗАМОВЛЕНЬ ДОСТАВКИ З РЕСТОРАНІВ**

### **2.1 Вимоги до структури й функціонування вебдодатку сервісу замовлень доставки з ресторанів**

Реалізований вебдодаток дозволить користувачам легко реєструватися та авторизуватися, шукати ресторани та доступні страви, розміщувати замовлення, здійснювати платежі та відстежувати статус своїх замовлень. Дані функції забезпечать безперебійну та ефективну роботу додатку для користувачів.

### **2.2 Вимоги до персоналу**

Обслуговуючий персонал платформи повинен володіти різноманітними навичками, включаючи обслуговування клієнтів та підтримку користувачів, щоб забезпечити безперебійну роботу платформи.

### **2.3 Вимоги до стилістичного оформлення вебдодатку сервісу замовлень доставки з ресторанів**

Інтерфейс вебдодатку сервісу замовлень доставки з ресторанів повинен бути розроблений таким чином, щоб бути зрозумілим і привабливим для користувачів, з легким доступом до всіх основних функцій та інтуїтивно зрозумілим навігаційним меню.

## **2.4 Вимоги до розмежування доступу**

Вебдодаток сервісу замовлень доставки з ресторанів повинен надавати диференційований доступ до своїх функцій залежно від ролі користувача, будь то клієнт, чи адміністратор.

## **2.5 Наповнення сайту (контент)**

Вебдодаток сервісу замовлень доставки з ресторанів міститиме вичерпну інформацію про ресторани, меню, акції, відгуки клієнтів та процес доставки. Крім того, він міститиме цінні поради щодо вибору страв та оптимального часу для оформлення замовлення.

## **2.6 Вимоги до системи управління контентом вебдодатку сервісу замовлень доставки з ресторанів**

CMS повинна бути зручною для користувача і дозволяти адміністраторам ефективно управляти інформацією про ресторан, такою як меню, новини.

## **2.7 Функціональні можливості розділів**

Функціональні можливості розділів включають в себе персоналізовані сторінки з детальною інформацією про заклад, його меню, рейтинг та відгуки клієнтів. Також система надає можливість перегляду меню ресторану, додавання страв до кошика та замовлення доставки. Також передбачена інтеграція платіжних систем для здійснення оплати та можливість відстеження статусу замовлення в реальному часі.

## **2.8 Вимоги до програмної платформи та коду**

Вебдодаток повинен бути розроблений з використанням сучасних технологій, таких як HTML5, CSS3, JavaScript та PHP для створення інтерактивного та реактивного інтерфейсу. Використання бази даних MySQL забезпечить стабільну та безпечну роботу платформи.

## ДОДАТОК Б

### Планування робіт

Мета цього проекту - розробка вебдодатку сервісу замовлень доставки з ресторанів, який полегшує процес замовлення доставки їжі, виступаючи посередником між ресторанами та клієнтами. Вебдодаток дозволить користувачам впевнено обирати страви з асортименту ресторанів, оформлювати замовлення та спілкуватися безпосередньо з представниками ресторанів для уточнення деталей або внесення змін до замовлень. Ресторани зможуть впевнено приймати замовлення від клієнтів, вести з ними діалог і отримувати відгуки про їжу та обслуговування.

Для успішної реалізації поставленого завдання необхідно виконати наступні етапи:

- проаналізувати потреби сучасного ринку в вебдодатку для замовлення доставки їжі, які підкреслюють сучасні тренди споживання;
- дослідити особливості роботи сервісів доставки їжі та проаналізувати потреби користувачів;
- виконати аналіз аналогів вебдодатків та виявити їх сильні та слабкі сторони, які б можна було покращити;
- виконати розробку компонентів вебдодатку на основі потреб клієнтів;
- виконати структурно-функціональне моделювання для наочного представлення процесів взаємодії;
- виконати тестування вебдодатку та перевірити працездатність додатку.

## Деталізація мети проекту методом SMART

SMART управління проектами - це спосіб управління командою, який базується на досягненні цілей на основі SMART процесу. SMART-процес - це набір критеріїв, розроблених для того, щоб зробити цілі досяжними, а також забезпечити чіткий і очевидний прогрес впродовж проекту [20].

Результати наведені у таблиці Б.1

Таблиця Б.1 – Формалізація мети за технологією SMART

Конкретизація	Розробка вебдодатку для сервісу замовлень доставки з ресторанів
Вимірюваний	Створення додатку допоможе зменшити час (на 5 днів) та матеріальні ресурси, які ресторани витрачають на організацію доставки страв.
Досяжний	Проект реалізовується на основі великого досвіду веб-розробки та відповідає вимогам, визначеним у технічному завданні.
Значимість	Розроблений додаток дозволить ресторанам залучати більше клієнтів завдяки зручному процесу замовлення та доставки страв.
Обмежений в часі	Проект має бути завершений до встановленого терміну (червень 2024 року), що відповідає поставленому графіку розробки.

*Джерело: розроблено автором*

## **Work Breakdown Structure (WBS) – Ієрархічна структура робіт**

Структура розбиття робіт (WBS) - орієнтована на результат ієрархічна декомпозиція робіт, які має виконати команда проекту. WBS візуально визначає обсяг робіт на керовані частини, які команда проекту може зрозуміти, оскільки кожен рівень WBS забезпечує подальше визначення та деталізацію.

Команда проекту створює WBS шляхом визначення основних функціональних результатів і поділу цих результатів на менші системи та підрезультати. Ці підрезультати декомпозуються далі, поки не будуть визначені конкретні робочі пакети, необхідні для створення підрезультату.

Кожен пакет робіт повинен бути взаємовиключним (наприклад, будь-яка частина роботи може бути визначена тільки один раз в одному пакеті робіт), а сума всіх пакетів робіт повинна покривати 100% обсягу проекту (наприклад, якщо вона не включена в WBS, то вона не входить в обсяг проекту).

Обсяг робіт - це основа проекту, на якій розраховуються роботи, ресурси, вартість та інші області знань. Мета WBS - полегшити управління проектом в осмислений спосіб [21].

На рисунку Б.1 представлено WBS проекту щодо розробки web – додатку для сервісу замовлень доставки з ресторанів.

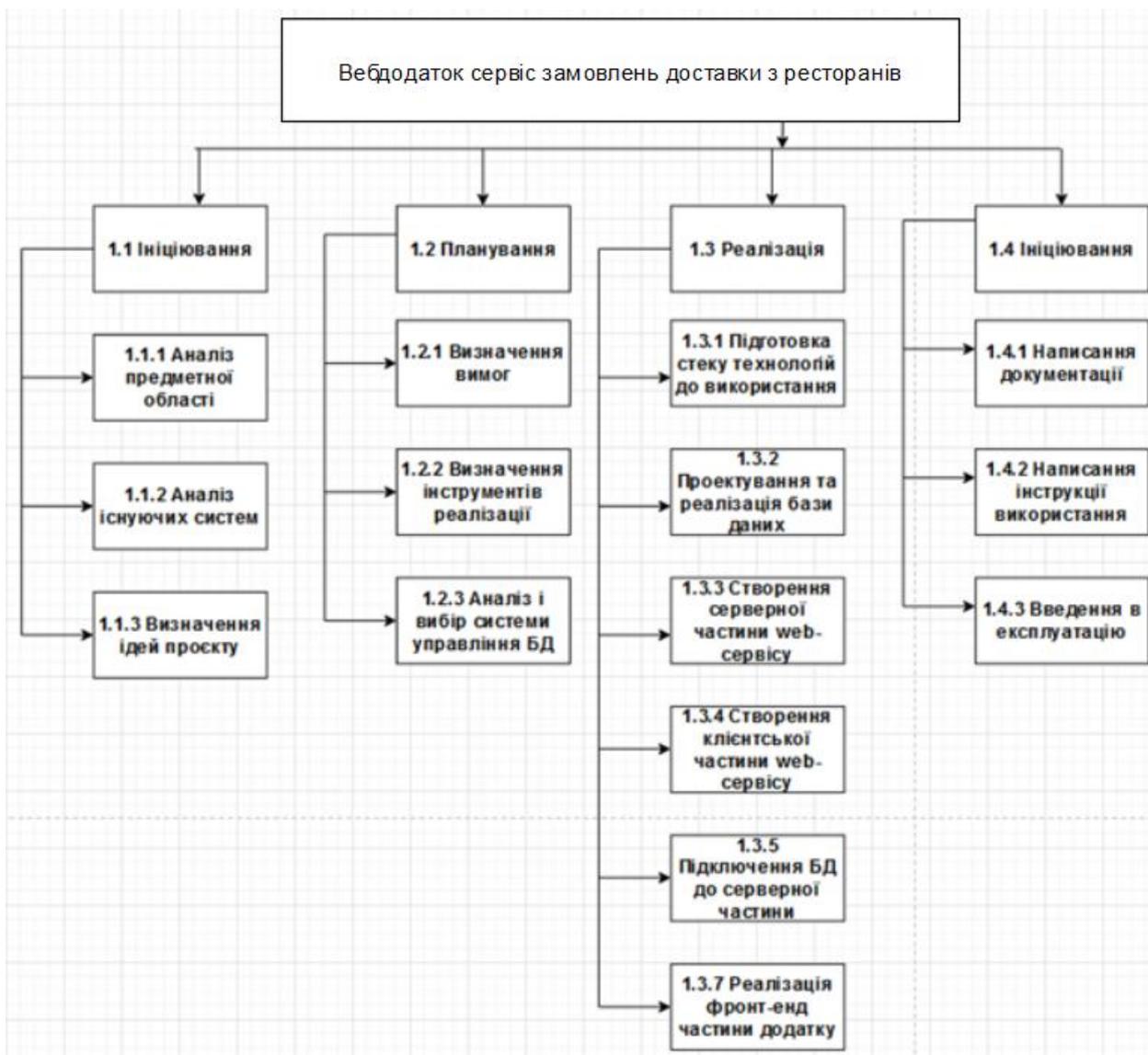


Рисунок Б.1 – WBS-структура робіт проекту

*Джерело: розроблено автором*

### Організаційна структура робіт (OBS)

OBS відіграє важливу роль в управлінні проектами, сприяючи чіткій комунікації, ефективному розподілу ресурсів, а також ефективному контролю та моніторингу проєктів. Вона дозволяє менеджерам проєктів впорядкувати робочі процеси, визначити вузькі місця та оптимізувати використання ресурсів. Завдяки чітко визначеній OBS проєктні команди можуть працювати злагоджено для досягнення успіху проєкту.

Організаційна структура проекту (OBS) - це потужний інструмент, який покращує комунікацію, координацію та контроль в управлінні проектами. Надаючи чіткі рамки для організації та визначення структури організації, OBS дозволяє проектним командам працювати злагоджено для досягнення успіху проекту [22].

На рисунку Б.2 представлено OBS-структуру робіт проекту

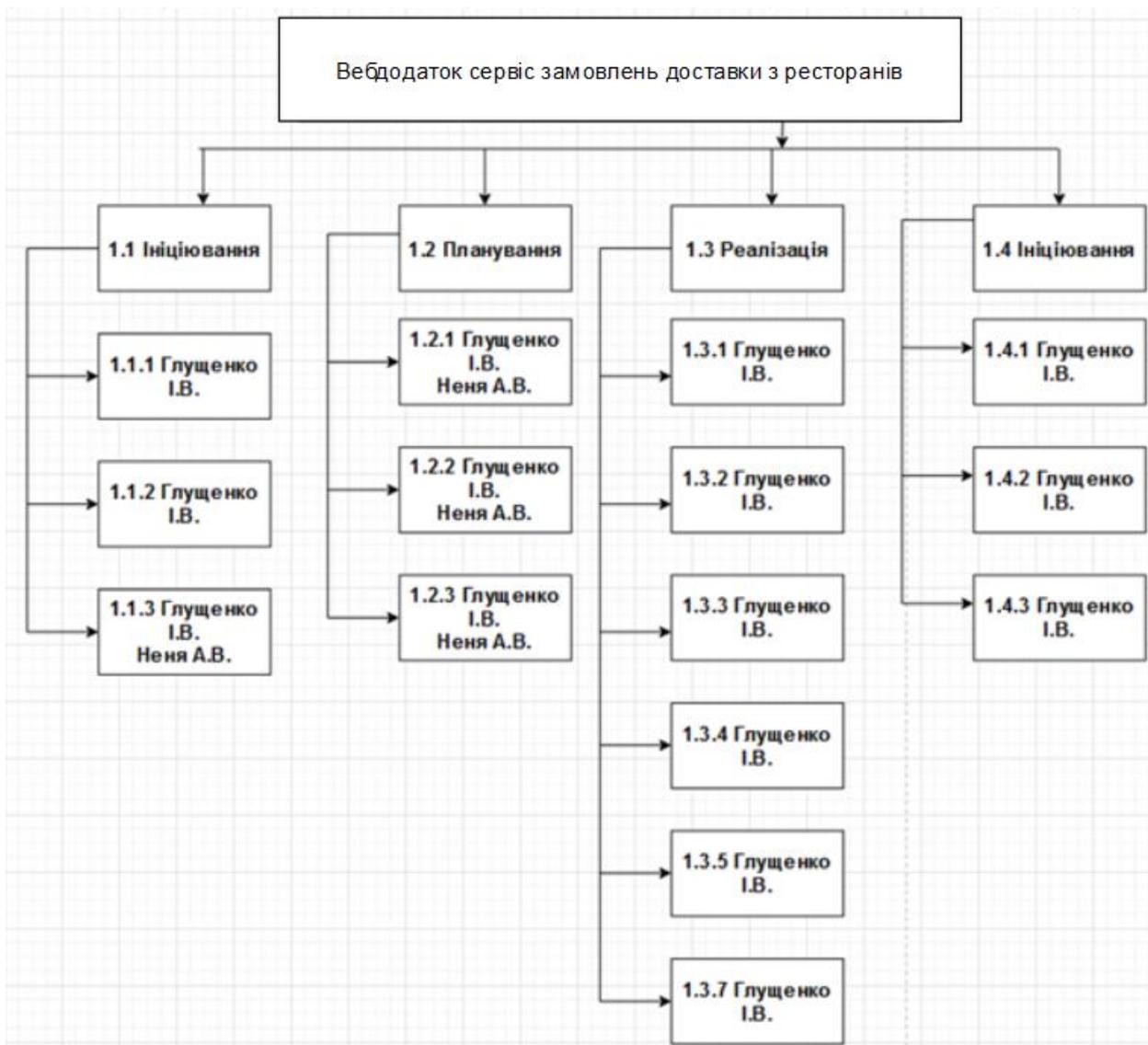


Рисунок Б.2 – OBS-структура робіт проекту

*Джерело: розроблено автором*

## Діаграма Ганта

Діаграми Ганта є дуже важливими для управління проектами та обговорення. Вони дозволяють керівникам проектів ефективно співпрацювати з членами своєї команди. Керівникам команд легко призначати та імпровізувати завдання за допомогою діаграм Ганта. На додаток до інших переваг діаграми Ганта, вони слугують для надання повної картини залежностей, етапів та часових рамок [23].

Календарний план проекту представлено на рисунках Б.3 – Б.4.

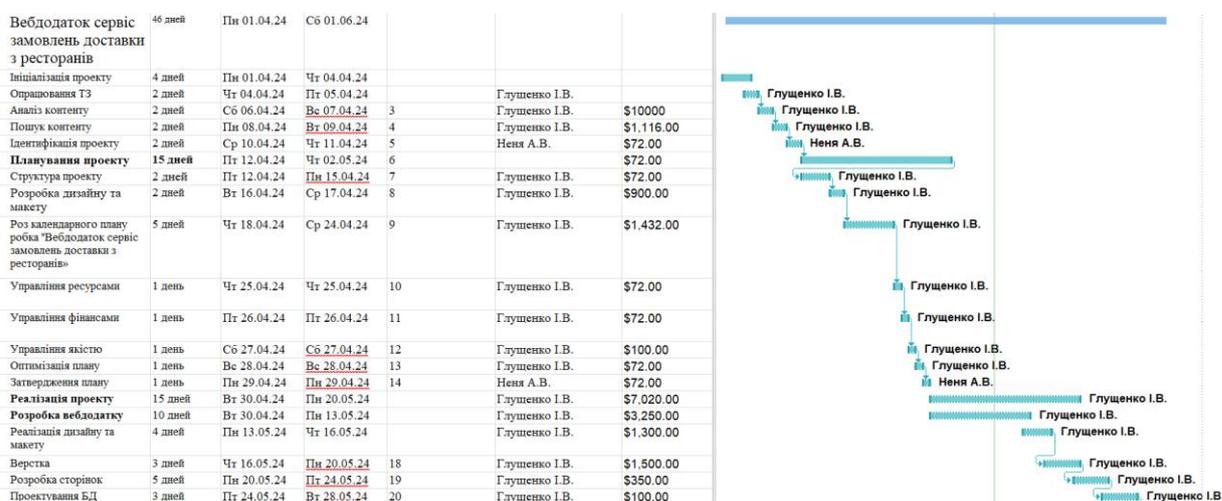


Рисунок Б.3 – Діаграма Ганта частина 1

Джерело: розроблено автором



Рисунок Б.4 – Діаграма Ганта частина 2

Джерело: розроблено автором

## Управління ризиками

Управління ризиками - це процес виявлення, оцінки та контролю загроз для капіталу, прибутків та діяльності організації. Ці ризики виникають з різних джерел, включаючи фінансову невизначеність, юридичні зобов'язання, технологічні проблеми, помилки в стратегічному управлінні, нещасні випадки та стихійні лиха.

Успішна програма управління ризиками допомагає організації враховувати весь спектр ризиків, з якими вона стикається. Ризик-менеджмент також вивчає взаємозв'язок між різними типами бізнес-ризиків і каскадний вплив, який вони можуть мати на стратегічні цілі організації.

Управління ризиками, мабуть, ніколи не було таким важливим, як зараз. Ризики, з якими стикаються сучасні організації, стали більш складними через швидкі темпи глобалізації. Постійно з'являються нові ризики, часто пов'язані та породжені повсюдним використанням цифрових технологій. Зміна клімату була названа експертами з ризиків "мультиплікатором загроз".

Нещодавній зовнішній ризик, який спочатку проявився у багатьох компаніях як проблема ланцюга постачання - пандемія COVID-19 - швидко перетворився на загрозу для існування, впливаючи на здоров'я та безпеку працівників, засоби ведення бізнесу, здатність взаємодіяти з клієнтами та корпоративну репутацію.

Бізнес швидко пристосувався до загроз, які несе з собою пандемія. Але, рухаючись вперед, вони стикаються з новими ризиками, серед яких постійне питання про те, як і чи варто повертати співробітників в офіс, що можна зробити, щоб зробити ланцюги поставок менш вразливими, інфляція, а також бізнес та економічні наслідки війни в Україні [24].

Таблиця А.2. Ймовірність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	Недостатня відповідність законодавству щодо захисту даних	2	4
2	Перенавантаження додатку у пікові часи	3	3
3	Порушення авторських прав на програмне забезпечення	3	5
4	Технічний збій в роботі БД	2	3
5	Витік особистої інформації користувачів про замовлення	4	5
6	Зловживання привілеями персоналу	2	3

*Джерело: розроблено автором*

Таблиця А.3 – Матриця впливу

Вірогідність виникнення	Матриця впливу				
5			3	5	
4		4			
3		6	2		
2				1	
1					
Ступінь впливу	1	2	3	4	5

*Джерело: розроблено автором*

## ДОДАТОК В

## Лістинг коду login.php

```
<!DOCTYPE html>
<html lang="en" >

<head>
  <meta charset="UTF-8">
  <title>login</title>

  <link                                rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/meyer-
reset/2.0/reset.min.css">

  <link                                rel='stylesheet'                prefetch'
href='https://fonts.googleapis.com/css?family=Roboto:400,100,300
,500,700,900|RobotoDraft:400,100,300,500,700,900'>
<link                                rel='stylesheet'                prefetch'
href='https://maxcdn.bootstrapcdn.com/font-
awesome/4.3.0/css/font-awesome.min.css'>

  <link rel="stylesheet" href="css/login.css">

  <style type="text/css">
  #buttn{
    color:#fff;
    background-color: #ff3300;
  }
</style>

</head>

<body>
<?php
include("connection/connect.php"); //INCLUDE CONNECTION
error_reporting(0); // hide undefine index errors
```

```

session_start(); // temp sessions
if(isset($_POST['submit'])) // if button is submit
{
    $username = $_POST['username']; //fetch records from login
form
    $password = $_POST['password'];

    if(!empty($_POST["submit"])) // if records were not empty
    {
        $loginquery ="SELECT * FROM users WHERE username='$username'
&& password='".md5($password)."'"; //selecting matching records
        $result=mysqli_query($db, $loginquery); //executing
        $row=mysqli_fetch_array($result);

                                if(is_array($row)) // if matching
records in the array & if everything is right
                                {
                                    $_SESSION["user_id"] =
$row['u_id']; // put user id into temp session

header("refresh:1;url=index.php"); // redirect to index.php page
                                }
                                else
                                {
                                    $message = "Invalid
Username or Password!"; // throw error
                                }
                            }

}

?>

<div class="pen-title">
    <h1>Login Form</h1>

```

```

</div>
<!-- Form Module-->
<div class="module form-module">
  <div class="toggle">

  </div>
  <div class="form">
    <h2>Login to your account</h2>
    <span style="color:red;"><?php echo $message; ?></span>
    <span style="color:green;"><?php echo $success; ?></span>
    <form action="" method="post">
      <input          type="text"          placeholder="Username"
name="username"/>
      <input          type="password"      placeholder="Password"
name="password"/>
      <input type="submit" id="buttn" name="submit" value="login"
/>
    </form>
  </div>

  <div class="cta">Not registered?<a href="registration.php"
style="color:#f30;"> Create an account</a></div>
</div>
  <script
src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.m
in.js'></script>

</body>

</html>

```

### Лістинг коду registration.php

```

<!DOCTYPE html>
<html lang="en">
<?php

```

```

session_start();
error_reporting(0);
include("connection/connect.php");
if(isset($_POST['submit'] ))
{
    if(empty($_POST['firstname']) ||
        empty($_POST['lastname']) ||
        empty($_POST['email']) ||
        empty($_POST['phone']) ||
        empty($_POST['password']) ||
        empty($_POST['cpassword']) ||
        empty($_POST['cpassword']))
    {
        $message = "All fields must be Required!";
    }
    else
    {

        $check_username= mysqli_query($db, "SELECT username FROM
users where username = '". $_POST['username']."' ");
        $check_email = mysqli_query($db, "SELECT email FROM users
where email = '". $_POST['email']."' ");

        if($_POST['password'] != $_POST['cpassword']){ //matching
passwords
            $message = "Password not match";
        }
        elseif(strlen($_POST['password']) < 6) //cal password length
        {
            $message = "Password Must be >=6";
        }
        elseif(strlen($_POST['phone']) < 10) //cal phone length

```

```

    {
        $message = "invalid phone number!";
    }

    elseif (!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL))
// Validate email address
    {
        $message = "Invalid email address please type a valid
email!";
    }
    elseif(mysqli_num_rows($check_username) > 0) //check
username
    {
        $message = 'username Already exists!';
    }
    elseif(mysqli_num_rows($check_email) > 0) //check email
    {
        $message = 'Email Already exists!';
    }
    else{

        //inserting values into db
        $mysql = "INSERT INTO
users(username,f_name,l_name,email,phone,password,address)
VALUES('".$_POST['username']."','".$_POST['firstname']."','".$_P
OST['lastname']."','".$_POST['email']."','".$_POST['phone']."','
".md5($_POST['password']).','".$_POST['address']."'");
        mysqli_query($db, $mysql);
        $success = "Account Created successfully! <p>You will be
redirected in <span id='counter'>5</span> second(s).</p>

<script type='text/javascript'>

function countdown() {

```

```

var i = document.getElementById('counter');

if (parseInt(i.innerHTML)<=0) {

    location.href = 'login.php';

}

i.innerHTML = parseInt(i.innerHTML)-1;

}

setInterval(function(){ countdown(); },1000);

</script>";

    header("refresh:5;url=login.php"); // redirected once
inserted success
}
}

}

?>

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <!-- The above 3 meta tags *must* come first in the head; any
other head content must come *after* these tags -->
    <meta name="description" content="">
    <meta name="author" content="">

```

```

<link rel="icon" href="#">
<title>Starter Template for Bootstrap</title>
<!-- Bootstrap core CSS -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<link href="css/font-awesome.min.css" rel="stylesheet">
<link href="css/animation.min.css" rel="stylesheet">
<link href="css/animate.css" rel="stylesheet">
<!-- Custom styles for this template -->
<link href="css/style.css" rel="stylesheet"> </head>
<body>

    <!--header starts-->
    <header id="header" class="header-scroll top-header
headrom">

        <!-- .navbar -->
        <nav class="navbar navbar-dark">
            <div class="container">
                <button class="navbar-toggler hidden-lg-up"
type="button" data-toggle="collapse" data-
target="#mainNavbarCollapse">&#9776;</button>
                <a class="navbar-brand" href="index.php">  </a>
                <div class="collapse navbar-toggleable-md
float-lg-right" id="mainNavbarCollapse">
                    <ul class="nav navbar-nav">
                        <li class="nav-item"> <a
class="nav-link active" href="index.php">Home <span class="sr-
only">(current)</span></a> </li>
                        <li class="nav-item"> <a class="nav-
link active" href="restaurants.php">Restaurants <span class="sr-
only"></span></a> </li>

                    <?php
                        if(empty($_SESSION["user_id"]))
                            {

```



```

</span>

</a></li>

</ul>
</div>
</div>
<section class="contact-page inner-page">
  <div class="container">
    <div class="row">
      <!-- REGISTER -->
      <div class="col-md-8">
        <div class="widget">
          <div class="widget-body">
            <form                                action=""
method="post">
              <div class="row">
                <div class="form-group
col-sm-12">
                  <label
for="exampleInputEmail1">User-Name</label>
                  <input                class="form-
control" type="text" name="username" id="example-text-input"
placeholder="UserName">
                </div>
                <div class="form-group col-
sm-6">
                  <label
for="exampleInputEmail1">First Name</label>
                  <input                class="form-
control" type="text" name="firstname" id="example-text-input"
placeholder="First Name">
                </div>

```

```

        <div class="form-group col-
sm-6">
            <label
for="exampleInputEmail1">Last Name</label>
            <input class="form-
control" type="text" name="lastname" id="example-text-input-2"
placeholder="Last Name">
        </div>
        <div class="form-group col-
sm-6">
            <label
for="exampleInputEmail1">Email address</label>
            <input type="text"
class="form-control" name="email" id="exampleInputEmail1" aria-
describedby="emailHelp" placeholder="Enter email"> <small
id="emailHelp" class="form-text text-muted">We'll never share your
email with anyone else.</small>
        </div>
        <div class="form-group col-
sm-6">
            <label
for="exampleInputEmail1">Phone number</label>
            <input class="form-
control" type="text" name="phone" id="example-tel-input-3"
placeholder="Phone"> <small class="form-text text-muted">We'll
never share your email with anyone else.</small>
        </div>
        <div class="form-group col-
sm-6">
            <label
for="exampleInputPassword1">Password</label>
            <input type="password"
class="form-control" name="password" id="exampleInputPassword1"
placeholder="Password">
        </div>

```

```

        <div class="form-group col-
sm-6">
            <label
for="exampleInputPassword1">Repeat password</label>
            <input type="password"
class="form-control" name="cpassword" id="exampleInputPassword2"
placeholder="Password">
        </div>
        <div class="form-
group col-sm-12">
            <label
for="exampleTextarea">Delivery Address</label>
            <textarea class="form-
control" id="exampleTextarea" name="address"
rows="3"></textarea>
        </div>
    </div>
    <div class="row">
        <div class="col-sm-4">
            <p> <input type="submit"
value="Register" name="submit" class="btn theme-btn"> </p>
        </div>
    </div>
</form>
</div>
<!-- end: Widget -->
</div>
<!-- /REGISTER -->
</div>
<!-- WHY? -->
<div class="col-md-4">

```

```

        <h4>Registration is fast, easy, and
free.</h4>

        <hr>

        <p></p>

        <!-- end:panel -->

        <!-- end:Panel -->
        <h4 class="m-t-20">Contact Customer
Support</h4>

        <p> If you're looking for more help or
have a question to ask, please </p>
        <p> <a href="contact.html" class="btn
theme-btn m-t-15">contact us</a> </p>
        </div>
        <!-- /WHY? -->
    </div>
</div>
</section>

<!-- start: FOOTER -->
<footer class="footer">
    <div class="container">
        <!-- top footer statrs -->
        <div class="row top-footer">
            <div class="col-xs-12 col-sm-3 footer-logo-
block color-gray">
                <a href="#">  </a> <span>Order Delivery & Take-
Out </span>
            </div>
            <div class="col-xs-12 col-sm-2 about color-
gray">

```

```

<h5>About Us</h5>
<ul>
  <li><a href="#">About us</a> </li>
  <li><a href="#">History</a> </li>
  <li><a href="#">Our Team</a> </li>
  <li><a href="#">We are hiring</a>
</li>

  </ul>
</div>
<div class="col-xs-12 col-sm-2 how-it-works-
links color-gray">
  <h5>How it Works</h5>
  <ul>
    <li><a href="#">Enter your
location</a> </li>
    <li><a href="#">Choose restaurant</a>
</li>
    <li><a href="#">Choose meal</a> </li>
    <li><a href="#">Pay via credit
card</a> </li>
    <li><a href="#">Wait for delivery</a>
</li>

  </ul>
</div>
<div class="col-xs-12 col-sm-2 pages color-
gray">
  <h5>Pages</h5>
  <ul>
    <li><a href="#">Search results
page</a> </li>
    <li><a href="#">User Sing Up Page</a>
</li>
    <li><a href="#">Pricing page</a> </li>
    <li><a href="#">Make order</a> </li>
    <li><a href="#">Add to cart</a> </li>

```

```

        </ul>
    </div>

</div>

<!-- top footer ends -->
<!-- bottom footer starts -->
<div class="row bottom-footer">
    <div class="container">
        <div class="row">
            <div class="col-xs-12 col-sm-3
payment-options color-gray">
                <h5>Payment Options</h5>
                <ul>
                    <li>
                        <a href="#">  </a>
                    </li>
                    <li>
                        <a href="#">  </a>
                    </li>
                    <li>
                        <a href="#">  </a>
                    </li>
                    <li>
                        <a href="#">  </a>
                    </li>
                    <li>
                        <a href="#">  </a>
                    </li>
                </ul>
            </div>
        </div>
    </div>

```

```

        <div class="col-xs-12 col-sm-4 address
color-gray">
            <h5>Address</h5>
            <p>Concept design of oline food
order and deliveye,planned as restaurant directory</p>
            <h5>Phone:                <a
href="tel:+080000012222">080 000012 222</a></h5>
        </div>
        <div class="col-xs-12 col-sm-5
additional-info color-gray">
            <h5>Addition informations</h5>
            <p>Join the thousands of other
restaurants who benefit from having their menus on TakeOff</p>
        </div>
    </div>
</div>
</div>
</div>
<!-- bottom footer ends -->
</div>
</footer>
<!-- end:Footer -->
</div>
<!-- end:page wrapper -->

<script src="js/jquery.min.js"></script>
<script src="js/tether.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/ansition.min.js"></script>
<script src="js/bootstrap-slider.min.js"></script>
<script src="js/jquery.isotope.min.js"></script>
<script src="js/headroom.js"></script>
<script src="js/foodpicky.min.js"></script>
</body>
</html>

```

## ДОДАТОК Г

## Лістинг коду your\_orders.php

```
<!DOCTYPE html>
<html lang="en">
<?php
include("connection/connect.php");
error_reporting(0);
session_start();

if(empty($_SESSION['user_id'])) //if usser is not login
redirected baack to login page
{
    header('location:login.php');
}
else
{
?>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="#">
    <title>Starter Template for Bootstrap</title>
    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/font-awesome.min.css" rel="stylesheet">
    <link href="css/ansition.min.css" rel="stylesheet">
    <link href="css/animate.css" rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/style.css" rel="stylesheet">
<style type="text/css" rel="stylesheet">
```

```
.indent-small {
  margin-left: 5px;
}
.form-group.internal {
  margin-bottom: 0;
}
.dialog-panel {
  margin: 10px;
}
.datepicker-dropdown {
  z-index: 200 !important;
}
.panel-body {
  background: #e5e5e5;
  /* Old browsers */
  background: -moz-radial-gradient(center, ellipse cover,
#e5e5e5 0%, #ffffff 100%);
  /* FF3.6+ */
  background: -webkit-gradient(radial, center center, 0px,
center center, 100%, color-stop(0%, #e5e5e5), color-stop(100%,
#ffffff));
  /* Chrome,Safari4+ */
  background: -webkit-radial-gradient(center, ellipse cover,
#e5e5e5 0%, #ffffff 100%);
  /* Chrome10+,Safari5.1+ */
  background: -o-radial-gradient(center, ellipse cover, #e5e5e5
0%, #ffffff 100%);
  /* Opera 12+ */
  background: -ms-radial-gradient(center, ellipse cover, #e5e5e5
0%, #ffffff 100%);
  /* IE10+ */
  background: radial-gradient(ellipse at center, #e5e5e5 0%,
#ffffff 100%);
  /* W3C */
```

```
    filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr='#e5e5e
5', endColorstr='#ffffff', GradientType=1);
    /* IE6-9 fallback on horizontal gradient */
    font: 600 15px "Open Sans", Arial, sans-serif;
}
label.control-label {
    font-weight: 600;
    color: #777;
}

table {
    width: 750px;
    border-collapse: collapse;
    margin: auto;

}

/* Zebra striping */
tr:nth-of-type(odd) {
    background: #eee;
}

th {
    background: #ff3300;
    color: white;
    font-weight: bold;
}

td, th {
    padding: 10px;
    border: 1px solid #ccc;
    text-align: left;
```

```
font-size: 14px;

}

@media
only screen and (max-width: 760px),
(min-device-width: 768px) and (max-device-width: 1024px) {

    table {
        width: 100%;
    }

    /* Force table to not be like tables anymore */
    table, thead, tbody, th, td, tr {
        display: block;
    }

    /* Hide table headers (but not display: none;, for
accessibility) */
    thead tr {
        position: absolute;
        top: -9999px;
        left: -9999px;
    }

    tr { border: 1px solid #ccc; }

    td {
        /* Behave like a "row" */
        border: none;
        border-bottom: 1px solid #eee;
        position: relative;
        padding-left: 50%;
    }
}
```

```

td:before {
    /* Now like a table header */
    position: absolute;
    /* Top/left values mimic padding */
    top: 6px;
    left: 6px;
    width: 45%;
    padding-right: 10px;
    white-space: nowrap;
    /* Label the data */
    content: attr(data-column);

    color: #000;
    font-weight: bold;
}
}

</style>

</head>

<body>

    <!--header starts-->
    <header id="header" class="header-scroll top-header
headrom">
        <!-- .navbar -->
        <nav class="navbar navbar-dark">
            <div class="container">
                <button class="navbar-toggler hidden-lg-up"
type="button" data-toggle="collapse" data-
target="#mainNavbarCollapse">&#9776;</button>

```

```

        <a class="navbar-brand" href="index.html">
 </a>

        <div class="collapse navbar-toggleable-md
float-lg-right" id="mainNavbarCollapse">
            <ul class="nav navbar-nav">
                <li class="nav-item"> <a class="nav-
link active" href="index.php">Home <span class="sr-
only">(current)</span></a> </li>
                <li class="nav-item"> <a class="nav-
link active" href="restaurants.php">Restaurants <span class="sr-
only"></span></a> </li>

                <?php
                    if(empty($_SESSION["user_id"]))
                    {
                        echo '<li class="nav-
item"><a href="login.php" class="nav-link active">login</a>
</li>
                        <li class="nav-item"><a
href="registration.php" class="nav-link active">signup</a>
</li>';
                    }
                    else
                    {

                        echo '<li
class="nav-item"><a href="your_orders.php" class="nav-link
active">your orders</a> </li>';
                        echo '<li
class="nav-item"><a href="logout.php" class="nav-link
active">logout</a> </li>';
                    }

```



```

<!-- /widget heading -->
<div class="widget-heading">
  <h3 class="widget-title
text-dark">
    Popular tags
</h3>
  <div class="clearfix"></div>
</div>
<div class="widget-body">
  <ul class="tags">
    <li> <a href="#"
class="tag">
      Pizza
    </a> </li>
    <li> <a href="#"
class="tag">
      Sandwich
    </a> </li>
    <li> <a href="#"
class="tag">
      Sandwich
    </a> </li>
    <li> <a href="#"
class="tag">
      Fish
    </a> </li>
    <li> <a href="#"
class="tag">
      Desert
    </a> </li>
    <li> <a href="#"
class="tag">
      Salad
    </a> </li>
  </ul>

```

```

        </div>
    </div>
    <!-- end:Widget -->
</div>
<div class="col-xs-12 col-sm-7 col-md-7
">
    <div class="bg-gray restaurant-
entry">
        <div class="row">
            <table >
            <thead>
            <tr>
                <th>Item</th>
                <th>Quantity</th>
                <th>price</th>
                <th>status</th>
                <th>Date</th>
                <th>Action</th>
            </tr>
            </thead>
            <tbody>
                <?php
                // displaying current session user
login orders
                $query_res=
mysqli_query($db,"select * from users_orders where
u_id='". $_SESSION['user_id']."'");
                if(!mysqli_num_rows($query_res) > 0 )

```

```

    {

        echo '<td colspan="6"><center>You have No orders Placed
yet. </center></td>';

    }

else

{

while($row=mysqli_fetch_array($query_res))

{

    ?>

    <tr>

    <td data-column="Item"> <?php echo $row['title']; ?></td>

    <td data-column="Quantity"> <?php echo $row['quantity'];
?></td>

    <td data-column="price">$<?php echo $row['price']; ?></td>

    <td data-column="status">

    <?php

        $status=$row['status'];

        if($status==" " or $status=="NULL")

        {

```

```

?>

        <button type="button" class="btn
btn-info" style="font-weight:bold;">Dispatch</button>

<?php

        }

        if($status=="in process")

        { ?>

                <button type="button"
class="btn btn-warning"><span class="fa fa-cog fa-spin" aria-
hidden="true" ></span>On a Way!</button>

<?php

        }

        if($status=="closed")

        {

?>

                <button type="button" class="btn
btn-success" ><span class="fa fa-check-circle" aria-
hidden="true">Delivered</button>

<?php

        }

```

```

?>

<?php

if($status=="rejected")

    {

?>

        <button type="button" class="btn
btn-danger"> <i class="fa fa-close"></i>cancelled</button>

<?php

    }

?>

</td>

<td data-column="Date"> <?php echo $row['date']; ?></td>

<td data-column="Action"> <a
href="delete_orders.php?order_del=<?php echo $row['o_id'];?>"
onclick="return confirm('Are you sure you want to cancel your
order?');" class="btn btn-danger btn-flat btn-addon btn-xs m-b-
10"><i class="fa fa-trash-o" style="font-size:16px"></i></a>

```

```

        </div>
    </div>
</div>
</section>

<!-- start: FOOTER -->
<footer class="footer">
    <div class="container">
        <!-- top footer statrs -->
        <div class="row top-footer">
            <div class="col-xs-12 col-sm-3 footer-
logo-block color-gray">
                <a href="#">  </a> <span>Order Delivery
&amp; Take-Out </span> </div>
                <div class="col-xs-12 col-sm-2 about
color-gray">
                    <h5>About Us</h5>
                    <ul>
                        <li><a href="#">About us</a>
</li>
                        <li><a href="#">History</a>
</li>
                        <li><a href="#">Our Team</a>
</li>
                        <li><a href="#">We are
hiring</a> </li>
                    </ul>
                </div>
            <div class="col-xs-12 col-sm-2 how-it-
works-links color-gray">
                <h5>How it Works</h5>
                <ul>
                    <li><a href="#">Enter your
location</a> </li>

```

```

                                <li><a href="#">Choose
restaurant</a> </li>
                                <li><a href="#">Choose meal</a>
</li>
                                <li><a href="#">Pay via credit
card</a> </li>
                                <li><a href="#">Wait for
delivery</a> </li>
                                </ul>
</div>
<div class="col-xs-12 col-sm-2 pages
color-gray">
                                <h5>Pages</h5>
                                <ul>
                                <li><a href="#">Search results
page</a> </li>
                                <li><a href="#">User Sing Up
Page</a> </li>
                                <li><a href="#">Pricing page</a>
</li>
                                <li><a href="#">Make order</a>
</li>
                                <li><a href="#">Add to cart</a>
</li>
                                </ul>
</div>
</div>
<!-- top footer ends -->
<!-- bottom footer statrs -->
<div class="row bottom-footer">
    <div class="container">
        <div class="row">
            <div class="col-xs-12 col-sm-3
payment-options color-gray">

```

```

        <h5>Payment Options</h5>
        <ul>
            <li>
                <a href="#">  </a>
            </li>
            <li>
                <a href="#">  </a>
            </li>
            <li>
                <a href="#">  </a>
            </li>
            <li>
                <a href="#">  </a>
            </li>
            <li>
                <a href="#">  </a>
            </li>
        </ul>
    </div>
    <div class="col-xs-12 col-sm-4
address color-gray">
        <h5>Address</h5>
        <p>Concept design of oline
food order and deliveye,planned as restaurant directory</p>
        <h5>Phone: <a
href="tel:+080000012222">080 000012 222</a></h5> </div>
    <div class="col-xs-12 col-sm-5
additional-info color-gray">
        <h5>Addition
informations</h5>

```

```
        </div>
    </div>
</div>
</div>
<!-- bottom footer ends -->
</div>
</footer>
<!-- end:Footer -->
</div>

<script src="js/jquery.min.js"></script>
<script src="js/tether.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/animation.min.js"></script>
<script src="js/bootstrap-slider.min.js"></script>
<script src="js/jquery.isotope.min.js"></script>
<script src="js/headroom.js"></script>
<script src="js/foodpicky.min.js"></script>
</body>

</html>
<?php
}
?>
```