

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»
Директор інституту(декан факультету)
Андрій ФОРСЮК
(ім'я та прізвище)

«10» червня 2024р.

«До захисту допущено»
Завідувач кафедри
Сергій ГРИБКОВ
(ім'я та прізвище)

«10» червня 2024р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності 122 «Комп'ютерні науки»

(код та назва спеціальності)

освітньо-професійної програми Комп'ютерні науки

на тему: Розроблення веб орієнтованої системи для підтримки роботи закладу громадського харчування

Виконав: здобувач 4 курсу, групи КН-4-2

Тур Артем Володимирович

(прізвище, ім'я, по батькові повністю)

(підпис)

Керівник Грама Михайло Петрович

(прізвище, ім'я, по батькові повністю)

(підпис)

Консультанти

(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

(підпис)

Рецензент

(ім'я та прізвище)

(підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) незарядженої допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач _____

(підпис)

Київ - 2024р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь бакалавр

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інформаційних технологій, штучного інтелекту і кібербезпеки

Сергій ГРИБКОВ

“ 15 ” квітня 2024 року

ЗАВДАННЯ**НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА**

Тура Артема Володимировича

(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення веб орієнтованої системи для підтримки роботи закладу громадського харчування»

керівник роботи Грама Михайло Петрович ст. вик., PhD

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 15 квітня 2024 року № 279-кс

2. Строк подання здобувачем роботи 03.06.2024 р.

3. Вихідні дані до роботи

Дані про ресторани, дані про страви

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

1) Системний аналіз об'єкту дослідження

2) Проектування бази даних

3) Проектування інформаційної системи

4) Охорона праці та техніка безпеки

5. Перелік графічного матеріалу

1) База даних

2) Інтерфейс веб-орієнтованої системи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1.	Грама М.П. , ст. вик. , д. ф	15.04.2024	15.04.2024
2.	Грама М.П. , ст. вик. , д. ф	15.04.2024	15.04.2024
3.	Грама М.П. , ст. вик. , д. ф	15.04.2024	15.04.2024
4.	Грама М.П. , ст. вик. , д. ф	15.04.2024	15.04.2024

7. Дата видачі завдання 15 квітня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної області та постановка завдання на проектування	01.04.24 – 10.04.24	Виконано
2	Проектування бази даних	11.04.24 – 16.04.24	Виконано
3	Створення інформаційної системи	16.04.24 – 07.05.24	Виконано
4	Оформлення пояснювальної записки	07.05.24 – 16.05.24	Виконано
5	Оформлення презентації	16.05.24 – 22.05.24	Виконано

Здобувач

(підпис)

Тур А. В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Грама М. П.

(прізвище та ініціали)

АННОТАЦІЯ

Дипломна робота фокусується на розробці альтернативної веб-орієнтованої системи для мережі ресторанів української кухні “Пузата хата”. Задача полягає у створенні інноваційного і зручного інструменту, спрямованого на покращення взаємодії з клієнтами та підвищення їхнього задоволення від візитів до закладів мережі. Ця система має забезпечити не лише більшу функціональність для користувачів, а й оптимізувати процеси обслуговування. Таким чином, мета цієї розробки полягає в створенні зручної та ефективної системи, спрямованої на максимальне задоволення потреб та очікувань клієнтів мережі “Пузата Хата”.

У даній роботі був здійснений аналіз вимог мережі та визначені необхідні функціональні вимоги до системи. Створена база даних, а також розроблено інтерфейс користувача та функціональні можливості системи.

У результаті проекту розроблена альтернативна веб-орієнтована система для мережі ресторанів “Пузата хата”, що включатиме додаткові функціональності для задоволення потреб клієнтів. Ця нова система повинна покращити взаємодію між клієнтами та ресторанами, забезпечуючи їм зручний спосіб замовлення страв та інших послуг.

Дипломна робота складається із 104 сторінок, 10 таблиць, 48 рисунків, 2 додатків та 30 літературних джерел.

Ключові слова: ВЕБ-ОРІЄНТОВАНА СИСТЕМА, БАЗА ДАНИХ, ІНТРЕФЕЙС КОРИСТУВАЧА, РЕСТОРАН.

ABSTRACT

This thesis focuses on the development of an alternative web-based system for the Puzata Khata chain of Ukrainian restaurants. The task is to create an innovative and user-friendly tool aimed at improving customer interaction and increasing their satisfaction from visits to the chain's establishments. This system should not only provide greater functionality for users but also optimize service processes. Thus, the goal of this development is to create a convenient and efficient system aimed at maximizing the satisfaction of the needs and expectations of the Puzata Hata chain's customers.

In this work, the system requirements were analyzed and all functional requirements for the system were defined. The database was created, and the user interface and functionality of the system were developed.

The project will result in the development of an alternative web-based system for the Puzata Khata restaurant chain, which will include additional functionality to meet customer needs. This new system should improve the interaction between customers and restaurants, providing them with a convenient way to order food and services.

The thesis consists of 104 pages, 10 tables, 48 figures, 2 appendices and 30 references.

Keywords: WEB-BASED SYSTEM, DATABASE, USER INTERFACE, RESTAURANT.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ МЕРЕЖІ РЕСТОРАНІВ “ПУЗАТА ХАТА”	8
1.1. Загальна характеристика про мережу ресторанів “Пузата хата”	8
1.2. Організаційна структура мережі ресторанів “Пузата Хата”, роль і взаємодія підрозділів	9
1.3. Аналіз нинішнього стану комп’ютеризації мережі ресторанів “Пузата Хата”	13
1.4. Розроблення функціональної моделі та аналіз існуючих бізнеспроцесів. ...	15
1.5. Виявлені проблеми	18
1.6. Задачі автоматизації	19
1.7. Огляд існуючих рішень для розв’язання виявлених проблем	20
1.8. Порівняльна характеристика систем	22
1.9. Обґрунтування доцільності розробки веб-орієнтованої системи для мережі ресторанів “Пузата хата”	23
1.10. Концептуальна модель	24
1.11. Розрахунок економічного ефекту від впровадження системи	25
1.12. Підсумок розділу	33
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ	34
РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ	47
3.1. Інформаційне забезпечення системи	47
3.2. Алгоритмізація та реалізація комплексу задач автоматизації	50
3.3. Інструкція користувача	76
3.4. Технічне та системне забезпечення розробки	87
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	96
ВИСНОВОК	100
БІБЛІОГРАФІЧНИЙ СПИСОК.....	101
ДОДАТКИ	105
Додаток А. Інтерфейс веб-орієнтованої системи	105
Додаток Б. Код програми.....	108

ВСТУП

В сучасному світі, де швидкість, зручність і доступність є ключовими складовими успіху, розробка веб-орієнтованої системи для підтримки роботи закладу громадського харчування стає необхідністю, що визначає конкурентоспроможність. Ця система має потенціал не лише оптимізувати процеси внутрішнього управління закладом, а й істотно збільшити його привабливість для клієнтів.

Запровадження веб-системи для підтримки роботи закладу громадського харчування перетворює звичайні способи ведення бізнесу на більш ефективні та сучасні. Це забезпечує швидку обробку замовлень, а також покращує взаємодію з клієнтами через онлайн-сервіси та зручну систему замовлення.

Найбільш значущий вплив веб-орієнтованої системи відчувається як для бізнесу, так і для його користувачів. Для закладу громадського харчування це означає збільшення продуктивності, зниження витрат на управління та рекламу, а також підвищення рівня задоволеності клієнтів через зручність та швидкість обслуговування. Для користувачів це відкриває доступ до широкого асортименту послуг та страв за допомогою різних пристроїв, забезпечуючи зручність та доступність у будь-який час і в будь-якому місці. Клієнти матимуть змогу здійснювати замовлення онлайн, матимуть опцію доставки їжі кур'єром, що забезпечить їм зручність та швидкість обслуговування. Вони також зможуть забронювати столик онлайн, писати відгуки про ресторани, переглядати актуальну інформацію про ресторани, матимуть можливість почати кар'єру у ресторанному бізнесі заповнивши анкету.

Таким чином, розробка веб-орієнтованої системи для підтримки роботи закладу громадського харчування стає ключовим етапом у вдосконаленні сучасного бізнесу, підвищенні його конкурентоспроможності та задоволеності клієнтів.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ МЕРЕЖІ РЕСТОРАНІВ “ПУЗАТА ХАТА”

1.1. Загальна характеристика про мережу ресторанів “Пузата хата”

Мережа ресторанів “Пузата Хата” відкрила свої двері першим відвідувачам у 2003 році. Тоді це був перший ресторан на першому поверсі будинку на вулиці Басейній, 1/2, що на Бессарабці в центрі Києва.

Смачна українська кухня та швидке обслуговування були високо оцінені споживачами. І невдовзі про “Пузату Хату” дізнаються навіть за межами Києва. Зазвичай заклади розташовуються біля станцій метро або у великих торгових центрах. Наразі мережа нараховує більше тридцяти закладів швидкого харчування.

Сьогодні “Пузата Хата” є найбільшою мережею ресторанів сучасної домашньої кухні в Україні. 39 ресторанів працюють у Києві, Дніпрі, Львові, Одесі, Запоріжжі, Харкові та Луцьку.

У 2021 році “Пузата Хата” відсвяткувала своє повноліття – 18 років з дня відкриття першого ресторану – і залишається однією з небагатьох мереж-довгожителів у нашій країні.

Слідкуючи за трендами у ресторанному бізнесі, “Пузата Хата” удосконалювалася від самого початку свого заснування. Сьогодні “Пузата Хата” - це мережа сучасних ресторанів з якісним та швидким обслуговуванням, в яких можна скуштувати смачні страви української кухні.

Товариство з обмеженою відповідальністю “Пузата хата” належить В'ячеславу Костянтинівському і його братові Олександрю.

“Пузата хата” у Львові займає приміщення колишньої кав'ярні “Ренесанс”, у якому на зустрічі у вересні 1918 року українські офіцери утворили Центральний військовий комітет.

27 лютого 2010 року оголосили про процедуру банкрутства ТОВ «Пузата хата». 9 березня 2011 року Господарський суд Києва офіційно визнав товариство банкрутом [1].

Попри позови кредиторів “Пузата хата” продовжила свою діяльність і навіть спромоглась збільшити прибутки.

Меню “Пузата Хата” включає широкий вибір страв, які відображають культурну спадщину та регіональні особливості української кухні. Сюди входять борщ, вареники, голубці, ковбаси, м'ясні та рибні страви, салати, супи, десерти та інші страви, які популярні серед українців.

Одним із основних принципів “Пузатої Хати” є доступність. Вони пропонують широкий вибір страв за доступні ціни, що робить їх популярними серед різних шарів населення, включаючи студентів, туристів та місцевих мешканців.

Ця мережа відома також своєю уважністю до якості продуктів і сервісу. Вони активно використовують місцеві інгредієнти та стежать за високими стандартами безпеки та якості продуктів.

1.2. Організаційна структура мережі ресторанів “Пузата Хата”, роль і взаємодія підрозділів

1.2.1. Загальна схема організаційної структури

Відділ маркетингу ресторану відіграє надзвичайно важливу роль у привертанні клієнтів, створенні позитивного іміджу та підвищенні обігу. Одним з найефективніших інструментів, який використовується маркетологами ресторанів, є веб-система, та рекламні компанії.

Для отримання необхідної інформації для подальшого написання кваліфікаційної роботи, необхідно аналізувати організаційну структуру ресторану та провести аналіз працівників.

Згідно з дослідженням структури підприємства, було розроблено модель його організації (рис.1.1).

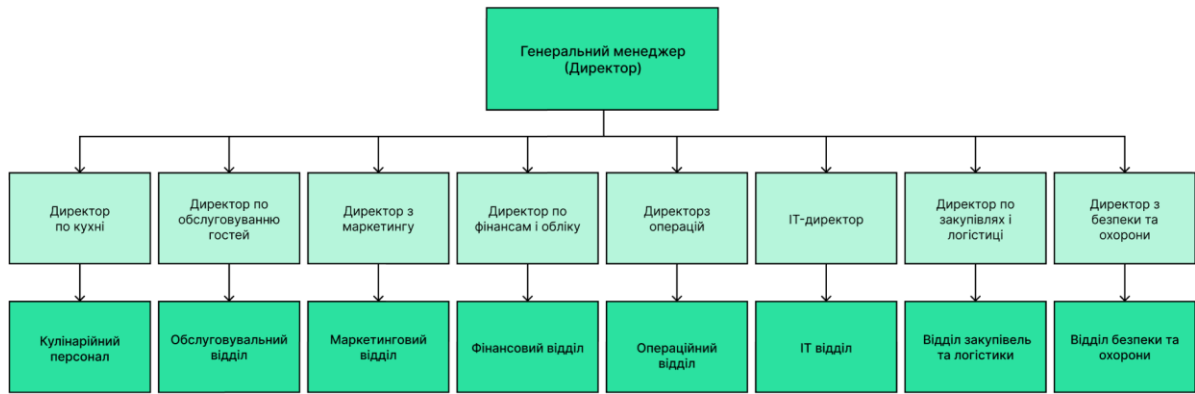


Рисунок 1.1 - Організаційно-функціональна схема ресторану

Ресторан має досить складну організаційну структуру з декількома директорами, кожен з яких відповідає за певний аспект діяльності. Нижче представлена детальна розбивка структури ресторану:

- **директор по кухні** - він керує кулінарним персоналом, включаючи шеф-кухарів, кухарів, помічників кухарів та інших працівників, які відповідають за приготування страв;
- **директор по обслуговуванню гостей** - він керує обслуговувальним відділом, включаючи офіціантів, барменів, хостес та інший персонал, який взаємодіє з гостями;
- **директор з маркетингу** - він відповідає за рекламні кампанії, акції, публічні відносини та стратегії просування ресторану на ринку;
- **директор по фінансам і обліку** - його обов'язки включають фінансове планування, облік, бюджетування та звітність;
- **директор операцій** - він відповідає за забезпечення плавного функціонування ресторану, включаючи управління робочими процесами та оптимізацію ефективності;
- **it-директор** - його відділ займається управлінням інформаційними технологіями, включаючи підтримку комп'ютерних систем, програмного забезпечення та мереж;

- **директор по закупівлях і логістиці** - він відповідає за закупівлю і доставку необхідних продуктів та матеріалів для ресторану, а також за логістичні аспекти діяльності;
- **директор з безпеки та охорони** - його відділ відповідає за забезпечення безпеки персоналу та гостей, включаючи планування екстрених ситуацій та заходи безпеки.

Ця структура дозволяє ресторану ефективно керувати різними аспектами діяльності та забезпечує плавний робочий процес.

Відділ який нам потрібний - відділ маркетингу. Отже, відділ маркетингу у ресторані відповідає за розробку та впровадження стратегій просування бізнесу, залучення нових клієнтів, збереження і залучення популярності ресторану, а також підтримку і розвиток іміджу закладу.

Відділ маркетингу відповідає за розробку рекламних кампаній та рекламних матеріалів, управління соціальними медіа та веб-системою ресторану для залучення уваги і комунікації з клієнтами.

Для більш детального дослідження відділу маркетингу, розглянемо його структуру у наступному підрозділі (рис. 1.2).

1.2.1. Структура відділу маркетингу

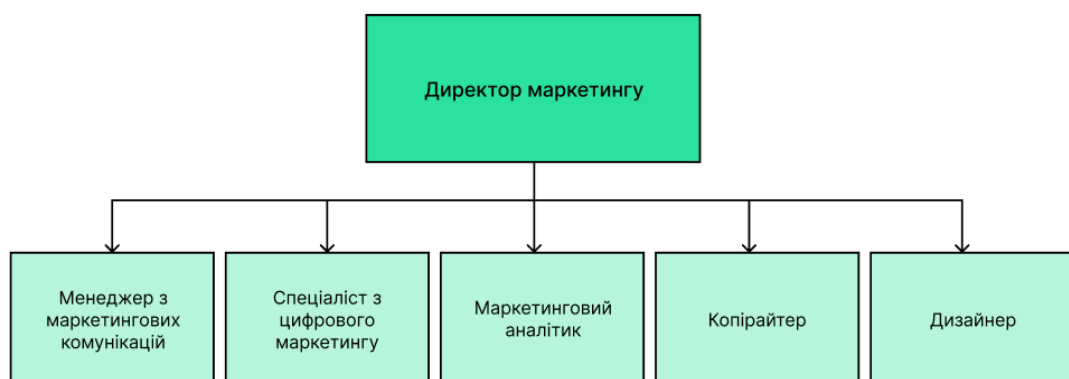


Рисунок 1.2 - Організаційна структура маркетингового відділу

Відділ маркетингу ресторану включає в себе різноманітні функції, спрямовані на просування закладу, залучення клієнтів та підвищення його популярності. Ось деякі з основних функцій:

- **маркетингові стратегії** - розробка стратегій маркетингу, включаючи аналіз цільової аудиторії, конкурентів та ринкових тенденцій;
- **продаж та реклама** - планування та впровадження рекламних кампаній, акцій та спеціальних пропозицій для привертання нових клієнтів та збереження існуючих;
- **брендування** - створення та підтримка бренду ресторану через логотипи, дизайн меню, обслуговування та інші аспекти;
- **онлайн присутність** - розробка та управління веб-системою ресторану;
- **соціальні медіа** - управління профілями ресторану в соціальних мережах, включаючи публікації, відповіді на коментарі та спілкування з клієнтами;
- **зв'язки з громадськістю (PR)** - організація подій, акцій, тематичних вечірок та інших заходів для привертання уваги ЗМІ та клієнтів;
- **аналіз даних** - збір, аналіз та інтерпретація даних про відвідуваність, відгуки клієнтів, популярність страв тощо для вдосконалення маркетингових стратегій;
- **креативний маркетинг** - виняткові та нестандартні способи просування, які вирізняють ресторан серед конкурентів;
- **зворотній зв'язок** - збір та аналіз відгуків клієнтів для постійного вдосконалення сервісу та маркетингових стратегій.

Ці функції можуть бути розділені між різними співробітниками відділу маркетингу в залежності від розміру та потреб ресторану.

Таблиця 1.1. Функціональні зв'язки із відділом маркетингу

Підрозділ	Одержання	Надання
Кухня	Запит на фотографії страв для реклами, та веб-системи	Представлення фотографій страв для рекламних матеріалів
Обслуговуючий персонал	Звіти про задоволеність клієнтів	Інформація щодо основних претензій клієнтів
Фінансовий відділ	Бюджет для маркетингових заходів	Звіти про витрати та результати маркетингових компаній
Адміністративний відділ	Отримання документації та інф. для підтримки маркетингових проєктів	Надання адміністративної підтримки для рекламних ініціатив та подій

Після того, як ми коротко розглянули структуру підприємства та його підрозділи, а також взаємозв'язки між ними, можемо приступити до аналізу стану комп'ютеризації.

1.3. Аналіз нинішнього стану комп'ютеризації мережі ресторанів “Пузата Хата”

Аналізуючи отриману інформацію, варто звернути увагу на роль технічного та програмного забезпечення в роботі.

Застосування веб-систем в сучасному бізнесі, включаючи ресторанний сегмент, вже стало стандартом через їхню велику ефективність у забезпеченні зручності для клієнтів і оптимізації бізнес-процесів. У випадку мережі ресторанів “Пузата Хата” веб-система може відігравати ключову роль у наступних аспектах:

- **онлайн-замовлення їжі** - веб-система дозволяє клієнтам замовляти страви онлайн, що значно полегшує їм процес вибору, забезпечує зручність та швидкість обслуговування;

- **інформаційна платформа** - система може містити інформацію про меню, акції, режим роботи, контактні дані та іншу корисну інформацію для клієнтів;
- **зворотний зв'язок** - веб-система може мати форми зворотного зв'язку, через які клієнти можуть залишати відгуки, пропозиції чи скарги, що дозволяє ресторану збирати фідбек та поліпшувати якість обслуговування;
- **маркетинг і реклама** - шляхом розміщення акцій, знижок та інших промо-подій у веб-система, ресторан може привертати нових клієнтів та підтримувати існуючу базу клієнтів;
- **адміністрування і управління** - веб-система також може використовуватися для внутрішнього управління та аналітики, наприклад, для відстеження обсягів продажів, управління запасами або планування виробничих процесів.

Усі ці аспекти дозволяють мережі ресторанів “Пузата Хата” покращити обслуговування своїх клієнтів, оптимізувати внутрішні процеси та підвищити конкурентоспроможність на ринку [2].

Хоча система комп'ютеризації мережі ресторанів “Пузата Хата” має значний потенціал і вже включає в себе деякі корисні функції, проте є очевидні недоліки, які варто вирішити. Відсутність таких функцій, як бронювання столика, відгуки клієнтів, підбірка найпопулярніших страв, обмежує можливості ресторанної мережі в покращенні обслуговування та залученні нових клієнтів.

Для оптимізації роботи і поліпшення обслуговування рекомендується впровадження додаткових функцій, які забезпечать зручність для клієнтів та оптимізують внутрішні бізнес-процеси. Наприклад, додавання онлайн-бронювання столиків, можливості залишати відгуки та пропозиції, а також функції доставки їжі кур'єром.

З урахуванням широкого спектру можливостей веб-системи, вона може стати не лише інструментом для замовлення їжі, але й інформаційною платформою для клієнтів та інструментом для внутрішнього управління ресторанною мережею. Впровадження таких інновацій дозволить мережі “Пузата Хата” підвищити рівень обслуговування, оптимізувати робочі процеси та збільшити конкурентоспроможність на ринку.

В додаток до відсутності деяких ключових функцій, варто зазначити, що деякі процеси виконуються не ефективно через недоліки у веб-системі. Наприклад, неефективна мапа ресторанів може ускладнювати для клієнтів пошук потрібного закладу. Відсутність можливості прокласти оптимальний маршрут до обраного ресторану зменшує зручність використання системи для користувачів, особливо для тих, хто шукає швидке та зручне рішення.

Такі несправності у функціоналі можуть призвести до збоїв в обслуговуванні та негативно вплинути на досвід користувача. Оптимізація картографічного інтерфейсу та введення можливості побудови оптимального маршруту до обраного ресторану можуть значно поліпшити ефективність та зручність використання веб-системи для клієнтів.

1.4. Розроблення функціональної моделі та аналіз існуючих бізнеспроцесів.

Розроблення функціональної моделі та аналіз існуючих бізнеспроцесів є важливим кроком в оптимізації діяльності підприємства. Це дозволяє зрозуміти, як саме працюють різні аспекти бізнесу, ідентифікувати потенційні проблеми та можливості для покращень. Шляхом аналізу існуючих процесів можна виявити ефективні та неефективні елементи, що дає змогу зосередитися на вдосконаленні та оптимізації бізнес-операцій. Функціональна модель надає візуальне відображення цих процесів, що полегшує їх розуміння та сприяє виявленню можливостей для розвитку та інновацій.

Можемо бачити схему процесу створення рекламної компанії (рис.1.3).

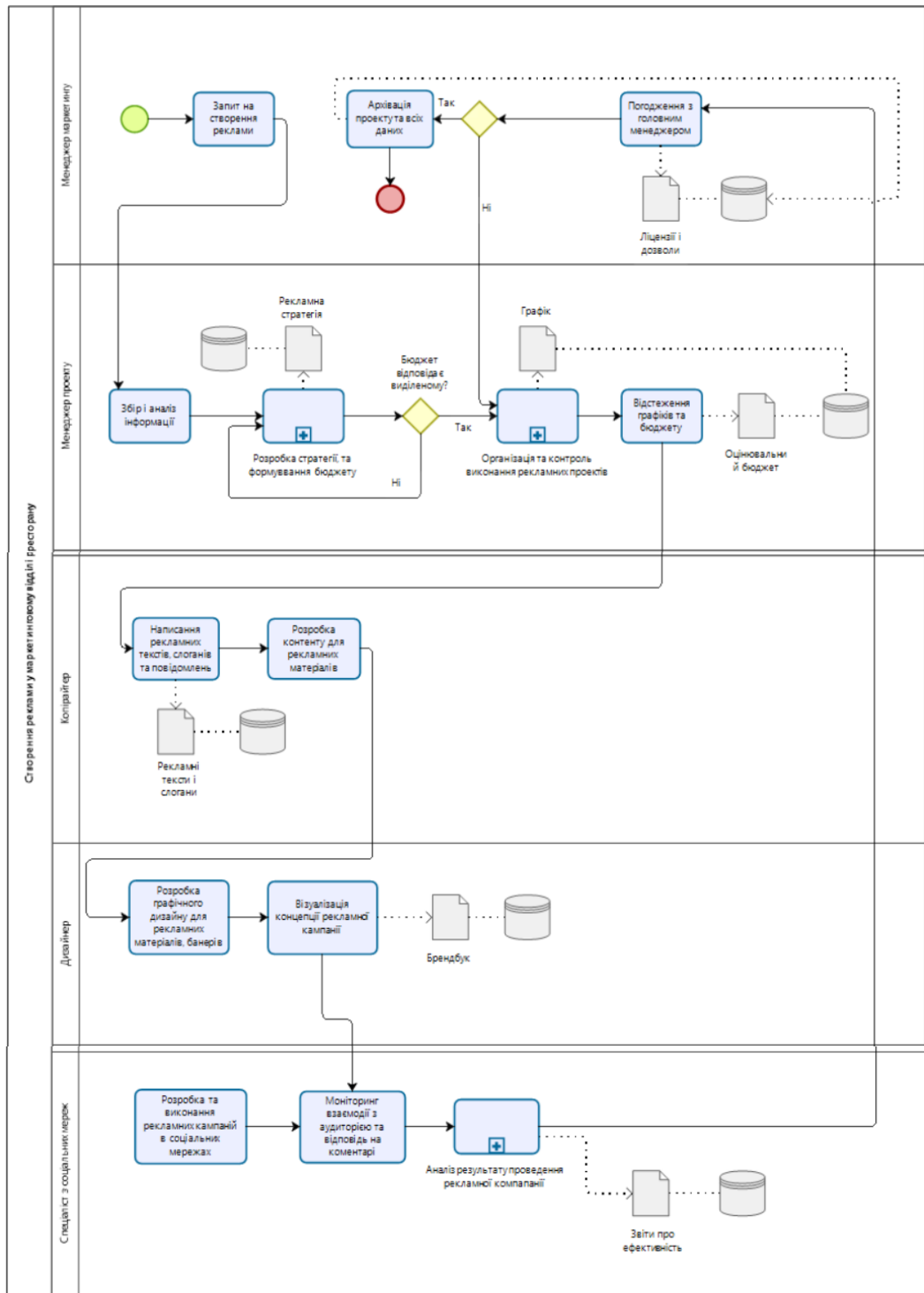


Рисунок 1.3 - Процес створення рекламної компанії у відділі маркетингу

Процес “створення рекламної компанії в маркетинговому відділі”, як видно з рисунку, наведеного вище, складається із наступних підпроцесів:

- розробка стратегії, та формування бюджету (рис.1.4);
- організація та контроль виконання рекламних проектів (рис.1.5);
- аналіз результату проведення рекламної компанії (рис.1.6).



Рисунок 1.4 - Підпроцес “Розробка стратегії, та формування бюджету”



Рисунок 1.5 - Підпроцес “Організація та контроль виконання рекламних проектів”



Рисунок 1.6 - Підпроцес “Аналіз результату проведення рекламної компанії”

1.5. Виявлені проблеми

При дослідженні веб-системи ресторану “Пузата хата” було помічено недоліки, які можуть негативно вплинути на користувачів та загальний досвід використання:

- **відсутність можливості бронювання столиків онлайн** - це обмежує зручність користувачів і може втратити клієнтів, які віддають перевагу швидкому та простому процесу бронювання через Інтернет. Багато людей шукають ресторани, де можна забронювати столик заздалегідь, і відсутність такої можливості може відлякати або розчарувати потенційних клієнтів;
- **відсутність можливості залишити відгук або переглянути відгуки прямо у веб-системі** - це призводить до втрати можливості для ресторану контролювати свій репутаційний образ та взаємодіяти з клієнтами. Відгуки є важливим елементом в будь-якому бізнесі, оскільки вони дозволяють відобразити задоволеність або незадоволеність клієнтів і вносити відповідні зміни для покращення сервісу;
- **відсутність E-mail розсилки** - це ускладнює збір та управління базою клієнтів, а також втрачає можливість для ресторану спілкуватися зі своїми клієнтами, повідомляти їх про акції, новини та спеціальні пропозиції;
- **незручна мапа розташування ресторанів** - незручність полягає в тому, що її можна побачити тільки коли вперше заходиш у систему, а потім вже її ніяк не знайти у системі, також немає можливості прокласти маршрут до потрібного ресторану;
- **відсутність підбірки найпопулярніших страв** - підбірка найпопулярніших страв може бути прекрасним інструментом для привернення уваги користувачів, особливо для тих, хто відвідує веб-систему вперше. Показуючи найкращі та найбільш популярні страви,

ресторан може зацікавити клієнтів і спонукати їх спробувати щось нове або популярне;

- **відсутність доставки їжі кур'єром** - відсутність можливості доставки їжі кур'єром у системі - це прямий шлях до втрати клієнтів та негативного впливу на репутацію бізнесу. Клієнти очікують зручності та швидкості. Без цієї послуги заклад втрачає конкурентну перевагу і ризикує тим, що клієнти перейдуть до конкурентів, які надають такий сервіс. Крім того, відсутність доставки кур'єром обмежує клієнтів, які мешкають або працюють далеко від закладу;
- **немає можливості подати заявуку для початку кар'єри у мережі** - відсутність можливості подачі заявки може призвести до того, що багато обіцяючих талантів просто не звертають увагу на цю мережу або шукають інші альтернативи. Це може призвести до втрати потенційних працівників та обмежити розвиток мережі. Додавання функціоналу подачі заявок може значно поліпшити взаємодію з потенційними кандидатами і забезпечити стабільний потік талантів для мережі.

Усі ці недоліки можуть призвести до того, що користувачі нададуть перевагу іншим ресторанам або сервісам, які пропонують кращий функціонал та зручний досвід користування.

1.6. Задачі автоматизації

Отже, проаналізувавши веб-системи ресторану “Пузата хата”, ми отримуємо відомості щодо вимог майбутнього проекту на кваліфікаційну роботу. Аби задовільнити вимоги, майбутній проект на кваліфікаційну роботу має давати користувачу здійснювати реалізацію наступних задач:

- функція бронювання столика;
- відгуки, та можливість їх додавати;
- підбірка найпопулярніших страв;
- створити більш продуману, та зручнішу мапа ресторанів;

- додати E-mail розсилку;
- додати можливість заповнення анкети для потенційних працівників;
- функція доставки їжі кур'єром.

Слід зазначити, що інтерфейс веб-системи повинен бути зрозумілим і легким у використанні, щоб спростити роботу персоналу і забезпечити зручне замовлення для клієнтів.

1.7. Огляд існуючих рішень для розв'язання виявлених проблем

На ринку існує багато веб-орієнтованих систем для підтримки роботи закладів громадського харчування, розглянемо найбільш відомі з цих систем, ознайомимося з їхніми особливостями та функціональними можливостями.

1.7.1. Веб-орієнтована система закладу “Салатерія”

Головною метою даної систем є надання клієнтам інформації про ресторани мережі, їхні меню, послуги, акції та умови обслуговування [3].

Можливості:

- замовлення їжі онлайн;
- акції та спеціальні пропозиції;
- контактна інформація;
- відгуки;
- мапа ресторанів;
- e-mail розсилка;
- можливість доставки;
- можливість почати кар'єру у закладі;
- адаптивна версія.

1.7.2. Веб-орієнтована система закладу “Макдональдз”

Ця система спрямована на надання клієнтам можливості легко замовляти їжу онлайн, отримувати інформацію про акції та пропозиції, дізнаватися про робочий час та розташування ресторанів, а також на отримання зручного зв'язку зі службою підтримки. В цілому, головною метою є полегшення

процесу замовлення та взаємодії з клієнтами, забезпечення їхнього задоволення та збільшення лояльності до бренду Макдональдз [4].

Можливості:

- замовлення їжі онлайн;
- акції та спеціальні пропозиції;
- контактна інформація;
- мапа ресторанів;
- e-mail розсилка;
- можливість доставки;
- можливість почати кар'єру у закладі;
- адаптивна версія.

1.7.3. Веб-орієнтована система закладу “КФС”

Головною метою системи є забезпечити зручний та ефективний спосіб замовлення їжі та послуг у ресторанах КФС через інтернет, щоб забезпечити клієнтам швидкий доступ до меню, збільшити зручність обслуговування та покращити загальний досвід від відвідування закладу [5].

Можливості:

- Замовлення їжі онлайн;
- Акції та спеціальні пропозиції;
- Контактна інформація;
- Мапа ресторанів;
- E-mail розсилка;
- Можливість доставки;
- Можливість почати кар'єру у закладі;
- Адаптивна версія.

1.8. Порівняльна характеристика систем

Отже, після уважного аналізу трьох систем ми можемо створити загальну таблицю, щоб зробити інформацію більш зрозумілою та простою для сприйняття. Для вказання наявності або відсутності відповідних функцій використовується символ “+”, якщо функція присутня, або “-”, якщо відсутня.

Таблиця 1.2. Порівняння систем аналогів “Салатерія”, “Макдональдз” та “КФС”

Функції	Салатерія	Макдональдз	КФС
Замовлення їжі онлайн	+	+	+
Акції та спеціальні пропозиції	+	+	+
Контактна інформація	+	+	+
Перегляд, додавання відгуків	+	-	-
Мапа ресторанів	+	+	+
E-mail розсилка	+	+	+
Можливість доставки кур'єром	+	+	+
Можливість почати кар'єру у закладі	+	+	+
Замовлення їжі онлайн	+	+	+
Можливість бронювання столиків онлайн	-	-	-
Адаптивна версія	+	+	+

1.9. Обґрунтування доцільності розробки веб-орієнтованої системи для мережі ресторанів “Пузата хата”

Розробка веб-орієнтованої системи для мережі ресторанів “Пузата хата” має великий потенціал для поліпшення якості обслуговування та збільшення конкурентоспроможності компанії. З урахуванням сучасних тенденцій у сфері харчової індустрії, впровадження такої системи дозволить забезпечити клієнтам зручність та швидкість обслуговування, збільшить їхню задоволеність та лояльність до бренду. Такий крок також сприятиме розвитку технологічного іміджу компанії та розширить її аудиторію через доступність онлайн-сервісів.

Також розробка такої веб-орієнтованої системи для мережі ресторанів “Пузата хата” стане важливим кроком у вдосконаленні їхнього бізнесу з ряду причин:

- сучасні споживачі все більше віддають перевагу зручності та швидкості обслуговування. Тому запровадження онлайн-системи бронювання столиків та замовлення страв дозволить клієнтам здійснювати ці операції у будь-який зручний для них час, що підвищить їхню задоволеність та відчуття комфорту;
- впровадження можливості залишати відгуки та переглядати їх збільшить впевненість клієнтів у якості обслуговування та страв, що пропонуються. Тому це сприятиме підвищенню довіри до бренду та збільшенню його привабливості для нових клієнтів;
- введення Email-розсилки дозволить ресторану ефективно спілкуватися з клієнтами, що відкриє нові можливості для реклами акцій, знижок та інших спеціальних пропозицій;
- покращена мапа розташування ресторанів спростить процес знаходження закладу для клієнтів, що також позитивно вплине на їхню задоволеність від обслуговування;

- можливість доставки їжі кур'єром розширить аудиторію ресторану та забезпечить зручність для клієнтів, які не можуть відвідати заклад особисто;
- можливість подати заявку для початку кар'єри у мережі приверне талановитих працівників та сприятиме розвитку кадрового потенціалу компанії.

Отже, розробка веб-орієнтованої системи для мережі ресторанів “Пузата хата” буде доцільною, оскільки вона дозволить покращити обслуговування клієнтів, збільшити їхню задоволеність та зміцнити позиції бренду на ринку.

1.10. Концептуальна модель

Концептуальна модель нашої веб-орієнтованої системи для мережі ресторану спрямована на задоволення потреб клієнтів. В роботу буде впроваджено онлайн сервіс, що дозволить користувачам швидко та зручно закривати свої потреби.

Перелік функцій:

- *перегляд інформації про ресторани* - система дозволить користувачам переглядати інформацію про кожен ресторан, включаючи розташування, контактні дані;
- *прокладання маршруту до ресторану* - система дозволить користувачам прокладати маршрут до обраного ресторану за допомогою інтеграції з картографічними сервісами Google Maps. Користувачі зможуть отримати детальні інструкції щодо проїзду, включаючи варіанти громадського транспорту або пішохідного маршруту;
- *відгуки* - система також включатиме функціонал для відгуків і оцінок, що допоможе новим клієнтам робити обґрунтований вибір, а ресторану - покращувати якість обслуговування на основі зворотного зв'язку;

- *бронювання столиків* - система дозволить користувачам вибрати бажану дату та час для резервування, також користувач може обрати кількість гостей;
- *меню* - система дозволить користувачам переглядати актуальне меню, розділене на категорії. Також система дозволить відображати детальну інформацію про кожну страву, включаючи опис інгредієнтів, калорійність. Це надасть клієнтам повну інформацію для прийняття рішення. У користувачів буде можливість бачити актуальні ціни на всі страви в меню. Це забезпечить прозорість і дозволить клієнтам планувати свої витрати;
- *панель управління* - адміністрація ресторану отримає доступ до розширеної панелі управління, де зможе відстежувати замовлення в реальному часі, керувати меню, цінами і акціями, та відгуками;
- *здійснювати замовлення їжі з доставкою або на виніс* - система дозволить користувачам оформлювати замовлення з доставкою додому. Можна буде обрати адресу доставки та бажаний час доставки. Система також дозволить користувачам оформлювати замовлення на виніс, обираючи зручний час для отримання.

Загалом, наша система буде спрямована на те, щоб забезпечити максимальну зручність для клієнтів і ефективність для персоналу ресторану, підвищуючи якість обслуговування та залученість клієнтів.

1.11. Розрахунок економічного ефекту від впровадження системи Визначення загальних даних веб-системи

Оцінка економічного ефекту від впровадження системи є надзвичайно важливою, оскільки вона базується на техніко-економічному обґрунтуванні розробки автоматизованої системи.

Визначення розміру оплати праці

Джерелами прибутку від створення веб-орієнтованої системи для ресторану можуть бути такі фактори:

- замовлення онлайн;
- резервації онлайн;
- маркетинг і реклама;
- аналітика.

Визначаємо ознаку – ефективність управління онлайн-замовленнями та резерваціями.

Рівень новизни розроблюваних завдань — "В" — включає використання типових проектних рішень з їх модифікацією, а також розробку проектів, що мають аналогічні рішення.

Група складності алгоритму — 3.

Узагальнені дані вхідної та вихідної інформації для сайту за видами вхідної та вихідної інформації таблиці 1.3.

Таблиця 1.3. Узагальнені дані для вхідної та вихідної інформації

Вид інформації	Позначення	К-сть наборів даних
Змінна інформація	ЗІ	m=6
Нормативно – довідкова інформація	НДІ	n=4
Банк(база) даних	БД	p=1
Обробка в режимі реального часу	РЧ	Так
Забезпечення телекомунікаційної обробки даних і управління віддаленими об'єктами	ТОУ	Ні

Таблиця 1.4. Визначення витрат часу для створення сайту ресторану

Вид системи	Стадія розробки сайту	
	Ескізний проект	Технічне завдання
	В	В
Функція онлайн замовлення; Функція онлайн бронювання; Функція додавання відгуків; Функція заповнення анкети;	$T_1=73$	$T_2=48$

Визначимо витрати часу на стадіях «технічний проект», «робочий проект» і «впровадження».

Вхідними даними для визначення є:

- кількість форм вхідної інформації 6;
- кількість форм вихідної інформації 4;
- базове значення витрат часу для стадії «Технічний проект» $T_{B3}=90$;
- базове значення витрат часу для стадії «Робочий проект» $T_{B4}=150$;
- базове значення витрат часу для стадії «Впровадження» $T_{B5}=55$.

Базове значення витрат часу T_B коригується за допомогою поправочних коефіцієнтів для всіх стадій розробки автоматизованої системи.

Таблиця 1.5. Коефіцієнти k_1, k_2, k_3 з видом використаної інформації

Стадія розробки проекту	Вид обробки	Ступінь ноизни
		В
Технічний проект	РЧ	1.26

Продовження таблиці 1.5. Коефіцієнти k_1, k_2, k_3 з видом використаної інформації

Робочий проект	РЧ	1.32
Впровадження	РЧ	1.21

Визначення витрат часу для стадії «Технічний проект»(ТЗ)

Щоб обчислити час, витрачений на етап "технічного проектування", можна використати наступний математичний вираз (1.1 – 1.4):

$$T_3 = T_{БЗ} * k_{\Pi} * k_o, \quad (1.1)$$

$$k_{\Pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p}, \quad (1.2)$$

Таблиця 1.6. Коефіцієнти k_1, k_2, k_3 для стадії «Технічний проект»

Вид використаної інформації	Ступінь новизни
	В
k_1 , (ЗІ)	1.0
k_2 , (НДІ)	0.72
k_3 , (БД)	2.08

$$k_{\Pi} = \frac{(1*6+0.72*4+2.08*1)}{(6+4+1)} = \frac{10.96}{11} = 0.996, \quad (1.3)$$

$$T_3 = 90 * 0.996 * 1.26 = 1126, \quad (1.4)$$

Визначення витрат часу на стадії «Робочий проект»

Для визначення часу на етапі «Робочий проект», слід скористатися формулою (1.5 – 1.6) для розрахунку k_{Π} .

$$k_{\Pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p}, \quad (1.5)$$

Таблиця 1.7. Коефіцієнти k_1, k_2, k_3 для стадії «Робочий проект»

Вид використаної інформації	Ступінь новизни
	В
$k_1, (ЗІ)$	1.3
$k_2, (НДІ)$	0.40
$k_3, (БД)$	2.08

$$k_{\Pi} = \frac{(1.3 \cdot 6 + 0.57 \cdot 4 + 0.40 \cdot 1)}{(6 + 4 + 1)} = 10.48/11 = 0.95, \quad (1.6)$$

$$T_4 = T_{Б4} * k_{\Pi} * k_o * k_c, \quad (1.7)$$

Щоб знайти k_c для формули (1.8) потрібно визначити складність контролю вхідної та вихідної інформації.

Тобто $k_c = 1.3$

$$T_4 = 150 * 0.953 * 1.32 * 1.3 = 196, \quad (1.8)$$

Визначення витрат часу на стадії «Впровадження» (1.9 – 1.10)

$$k_{\Pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p}, \quad (1.9)$$

$$T_5 = T_{Б5} * k_{\Pi} * k_o * k_c = T_5 = 55 * 0.953 * 1.21 * 1.3 = 80, \quad (1.10)$$

Отже, сумарні затрати людської праці становлять (1.11)

$$T_{\Sigma} = T_1 + T_2 + T_3 + T_4 + T_5 = 73 + 48 + 112 + 196 + 80 = 509, \quad (1.11)$$

Визначимо чисельність виконавців Ч (1.12)

$$\Phi = \frac{T_{\Sigma}}{\phi}, \quad (1.12)$$

Для розробки проекту знадобилося 530 робочих годин, при цьому робочий день тривав 7 годин. Отже, на розробку проекту було виділено Φ (1.13) днів:

$$\Phi = 530/7 = 75 \text{ днів}, \quad (1.13)$$

Для курсової роботи $\Phi = 75$ днів. Тоді визначаймо кількість місяців із розрахунку 25 робочих днів.

Кількість місяців на розробку, M (1.14):

$$M = \Phi/25 = 75/25 = 3 \text{ місяці}, \quad (1.14)$$

Отже, для виконання такого проекту потрібно така чисельність виконавців $Ч$, яка обраховується за формулою (1.15):

$$Ч = 509/75 = 6 \text{ виконавців}, \quad (1.15)$$

Прийmemo розмір заробітної плати розробників - 25000 грн, тоді загальна сума заробітних плат розробників складає (1.16):

$$V'_1 = Ч * M * ЗП = 5 * 3 * 25000 = 450000 \text{ грн}, \quad (1.16)$$

Розрахунок річного фонду часу роботи ПК

Дійсний річний фонд часу роботи комп'ютера у годинах дорівнює кількості робочих годин оператора протягом року, за винятком часу, витраченого на технічне обслуговування та ремонт комп'ютера. (в середньому 5 год/міс + 6 роб.днів/рік) (1.17).

$$T_{ПК} = 2000 - (6 * 8 + 5 * 12) = 1892 \text{ год.}, \quad (1.17)$$

Оскільки в процесі виконання проекту здобувач витрачає в середньому 450 годин машинного часу, то обсяг фонду часу ПК можна визначити за формулою (1.18):

$$T'_{ПК} = 1892 * (450/2000) = 425.7 \text{ год.}, \quad (1.18)$$

Поточні витрати на експлуатацію V

Балансована вартість ПК, де $Ц_P$ - ринкова вартість ПК, орієнтовно складає 40000 грн, $k_{УН}$ - коефіцієнт (1.19), що враховує витрати на установку ПК. $k_{УН} = 0,12$

$$Ц_{ПК} = Ц_P * (1 + k_{УН}) = 40000 * (1 + 0,12) = 44800 \text{ грн}, \quad (1.19)$$

Амортизаційні відрахування використання ПК, $З_{АМ}$, обчислюються за формулою (1.20)

$$Z_{AM} = \frac{C_{ПК}}{H_A} = 44800/5 = 8960 \text{ грн}, \quad (1.20)$$

Витрати на електроенергію ($Z_{ЕЛ}$), споживану ПК, обчислюються за формулою (1.21)

$$Z_{ЕЛ} = P_{ПК} * T_{ПК} * C_{ЕЛ} * A, \quad (1.21)$$

де потужність ПК, $P_{ПК} = 0.5$ кВт; фонд корисного часу роботи ПК, $T_{ПК} = 435.16$ год, вартість 1 кВт електроенергії для підприємств, $C_{ЕЛ} = 1,86$ грн/кВт, коефіцієнт інтенсивного використання ПК, $A = 0.9$ (1.22).

$$Z_{ЕЛ} = 0,5 * 435.16 * 1.68 * 0.9 = 329 \text{ грн}, \quad (1.22)$$

Витрати на поточний ремонт і технічне обслуговування ПК (Z_P) визначаються як 6% від балансової вартості ПК, $C_{ПК}$, за формулою (1.23):

$$Z_P = C_{ПК} * 0.06 = 44800 * 0.06 = 2688 \text{ грн}, \quad (1.23)$$

Непрямі витрати, пов'язані з експлуатацією ПК, визначаються як 5% від балансової вартості ПК $C_{ПК}$, за формулою (1.24):

$$Z_{МАТ} = C_{ПК} * 0.05 = 44800 * 0.05 = 2240 \text{ грн}, \quad (1.24)$$

Поточні витрати на експлуатацію V'' розраховуємо за формулою (1.25)

$$V''_1 = Z_{ОП} + Z_{AM} + Z_{ЕЛ} + Z_P + Z_{МАТ} \quad (1.25)$$

Середньою заробітною платою обслуговуючого персоналу $\epsilon - 10000$

Тож, поточні витрати на експлуатацію, V''_1 , грн, (1.26) складають:

$$V''_1 = 10000 + 8960 + 329 + 2688 + 2240 = 24\ 217 \text{ грн}, \quad (1.26)$$

А, загальні витрати на розробку програмного забезпечення комп'ютерної системи складуть (1.27):

$$V_1 = V'_1 + V''_1 = 450000 + 24\ 217 = 474\ 217 \text{ грн}, \quad (1.27)$$

Розрахунок витрат на придбання і установку ПК (1.28)

$$V_2 = C_{ПК} = 0 \text{ грн}, \quad (1.28)$$

Розрахунок витрат на підготовку приміщення і навчання персоналу

Витрати на підготовку приміщення $V_3 = 0$, так як приміщення є в наявності.

Витрати на навчання персоналу V_4 .

В середньому навчання персоналу триватиме один місяць, тому можна вважати, що $V_4 = 4500$ грн;

Вартість створення та запровадження системи розраховується згідно із наступною формулою (1.29):

$$V_{\Sigma} = V_1 + V_2 + V_3 + V_4 = 474217 + 0 + 0 + 4500 = 478717 \text{ грн} \quad (1.29)$$

Оскільки норма амортизаційних втрат для комп'ютерних систем $HA = 5$, то для обрахування (1.30) річного економічного ефекту слід брати до розгляду величину:

$$V_p = \frac{V_{\Sigma}}{HA} = \frac{478717}{5} = 95743 \text{ грн}, \quad (1.30)$$

Термін окупності розробки визначається за формулою (1.31):

$$T_{ок} = \frac{1}{K_{ЕФ}}, \quad (1.31)$$

де коефіцієнт економічної ефективності $K_{ЕФ} = \frac{P_p}{V_p}$, (1.32) де річний прибуток P_p від впровадження системи буде досягнуто за рахунок збільшення кількості працюючих і як результат – зменшення часу на перевірку знань учнів, і оцінюється приблизно в 40 000 грн на рік.

$$K_{ЕФ} = \frac{40\,000}{95743} = 0.4, \quad (1.32)$$

Оцінемо термін окупності веб-орієнтованої системи. Цей період визначає момент, коли витрати на розробку та впровадження системи будуть компенсовані прибутком або іншими вигодами для компанії. Важливо враховувати не тільки прямі витрати на розробку ПЗ, але й тривалі перспективи покращення та підвищення ефективності, які сприятимуть сталому успіху мережі ресторанів.

Отже, термін окупності ІС складатиме (1.33):

$$T_{ок} = \frac{1}{0.4} = 2.5 \text{ років}, \quad (1.33)$$

1.12. Підсумок розділу

На основі проведеного аналізу можна зробити висновок, що наявна веб-система мережі ресторанів “Пузата Хата” не є досконалою. Вона має певні недоліки, які негативно впливають на користувацький досвід та ефективність роботи системи в цілому. Зокрема, існують проблеми з функціональністю, яка обмежує можливості користувачів в отриманні повноцінного обслуговування.

Відсутність можливості бронювання столиків унеможливорює заздалегідь планувати візит до ресторану, що може викликати незручності та втрату клієнтів. Також відсутність системи відгуків позбавляє клієнтів можливості ділитися своїми враженнями та впливати на репутацію ресторанів у мережі. Недоліками також є не зручна мапа розташування ресторанів, що ускладнює їхнє знаходження та обирання для відвідування. Додавання функції доставки їжі кур'єром може привести до розширення аудиторії та збільшення обсягів продажів. У цілому, вдосконалення веб-системи мережі “Пузата Хата” є необхідним кроком для підвищення конкурентоспроможності та задоволеності клієнтів.

Тому для виправлення ситуації рекомендується розробити альтернативну веб-систему, яка б робила більший акцент на функціональності для користувачів. Вона повинна бути інтуїтивно зрозумілою, та зручною у використанні.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ

2.1. Загальні положення.

2.1.1. Найменування системи: «Розробка веб орієнтованої системи для підтримки роботи закладу громадського харчування»

2.1.2. Результати робіт зі створення системи оформлюються згідно з вимогами ДСТУ на відповідні етапи розробки. Порядок оформлення і передачі результатів у даному випадку визначається змістом і календарним планом виконання розробки.

2.1.3. У випадку необхідності на наступних стадіях робіт по створенню системи окремі положення можуть уточнюватися і розвиватися.

2.2. Призначення і цілі створення системи.

2.2.1. Призначення системи.

Система призначена для полегшення та оптимізації управління різними аспектами діяльності закладу. Ця система надає можливість здійснювати онлайн-замовлення для клієнтів, резервувати столики онлайн, можливість залишати відгуки та оцінювати якість обслуговування та страв, що допомагає покращувати роботу закладу, можливість легко знаходити заклади завдяки мапі ресторанів.

2.2.2. Цілі створення системи.

Основною метою створення системи полягає в покращенні ефективності та якості обслуговування закладу шляхом впровадження нових можливостей. Ця система спрямована на полегшення та оптимізацію управління різними аспектами діяльності, включаючи обробку замовлень, резервування столиків, збір відгуків і оцінок клієнтів і т.д. Головною метою є створення зручного і привабливого середовища для клієнтів, а також забезпечення зручності для управлінців закладу у веденні бізнесу та взаємодії з клієнтами.

2.3. Характеристика об'єкта автоматизації.

2.3.1. Короткі відомості про об'єкт автоматизації.

Об'єктом автоматизації є веб орієнтована система закладу мережі ресторанів “Пузата хата”

2.4. Вимоги до системи

2.4.1. Вимоги до системи в цілому.

2.4.1.1. Вимоги до структури і функціонування системи.

2.4.1.1.1. Система повинна мати привабливий сучасний дизайн, включати меню та фотографії страв, можливість онлайн-резервації та замовлення, інформацію про ресторан, акції, соціальні медіа, та можливості залишати відгуки.

2.4.1.1.2. Система повинна дозволяти користувачам виконувати наступні операції:

- функція бронювання столика;
- вибір дати та часу бронювання;
- онлайн-замовлення їжі;
- додавання страв у кошик та їх замовлення;
- вибір методу доставки або самовивозу;
- введення адреси для доставки і зручного часу отримання їжі;
- оновлення меню з фотографіями та описами страв та напоїв;
- відображення акцій, знижок та спеціальних пропозицій;
- інформація про розташування ресторану, години роботи та контактні дані;
- можливість адміністраторам ресторану керувати меню, акціями;
- відстеження замовлень та оплат;
- безпека та захист даних;
- додатково мобільна версія;
- можливість переглядати відгуки та їх додавання;
- можливість почати кар'єру відправивши заявку.

2.4.1.1.3. Структура і склад веб-орієнтованої системи

За функціональним складом система повинна складатися із:

- підсистема «замовлення онлайн»;
- підсистема «резервація онлайн».

2.4.1.1.4. Вимоги до підсистема «замовлення онлайн»

- зручний інтерфейс кошика;
- додавання страв до кошика з можливістю коригування;
- вибір часу доставки;
- забезпечення конфіденційності даних.

2.4.1.1.5. Вимоги до підсистема «резервація онлайн»

- календар для вибору дати та часу;
- вибір кількості гостей;
- захист конфіденційності.

2.4.1.1.6. Вимоги до клієнтської частини:

- зручне користувацьке інтерфейс - система повинна мати інтуїтивно зрозумілий і легкий у використанні інтерфейс для користувачів будь-якого рівня обізнаності;
- мобільна сумісність - система повинна бути адаптивною до різних розмірів екранів і бути доступною на смартфонах та планшетах.
- меню та ціни - система повинна містити актуальне меню ресторану, включаючи ціни та описи страв;
- замовлення онлайн - клієнти повинні мати можливість зробити замовлення онлайн, вибравши страви та напої з меню та здійснити оплату через систему;
- резервації - система повинна надавати можливість клієнтам робити резервації столів обираючи кількість гостей, дату та час;
- години роботи та контакти - інформація про години роботи ресторану, адресу, номер телефону повинна бути доступною в системі;

- акції та спеціальні пропозиції - клієнти повинні мати можливість дізнаватися про акції, через систему;
- фотографії - високоякісні фотографії, щоб показати атмосферу ресторану та представити страви в найкращому світлі;
- адміністративний доступ - забезпечте можливість адміністраторам ресторану оновлювати контент системи, додавати нові страви та акції;
- мапа ресторанів - сторінка на якій користувачі можуть переглядати ресторани, додаткову інформації про кожен ресторан, такої як години роботи, контактна інформація;
- відгуки - функціонал для додавання та перегляду відгуків про ресторани.

2.4.1.1.7. Вимоги до серверної частини:

- висока продуктивність - серверна частина повинна забезпечувати швидкий доступ до системи, навіть при великому навантаженні та пікових навантаженнях в часи популярних подій або свят;
- безпека даних - забезпечити надійний рівень захисту для всієї інформації, що передається та зберігається на сервері, включаючи особисті дані користувачів та інформацію про замовлення;
- масштабованість - система повинна бути готова до масштабування, щоб в разі потреби забезпечувати додаткові ресурси та функціонал без значних затримок;
- підтримка різних пристроїв - забезпечити адаптивність і підтримку для різних типів пристроїв, включаючи комп'ютери, та мобільні телефони;
- швидкість завантаження - важливо забезпечити максимально можливу швидкість завантаження сторінок та ресурсів для користувачів;
- моніторинг та аналітика - забезпечити засоби моніторингу серверної частини та аналізу її продуктивності та безпеки;

- відповідність законодавству - дотримуватися всіх відповідних законодавчих вимог, зокрема щодо захисту даних користувачів та авторських прав;
- можливість роботи в різних браузерах - забезпечити сумісність з різними веб-браузерами для забезпечення зручності користувачів.

2.4.1.1.8. Вимоги до модуля обміну інформацією з користувачами системи.

- форма замовлення - модуль повинен мати зручну форму для клієнтів, де вони можуть замовити їжу та напої онлайн. Форма має містити поля для вибору страв, кількості порцій, адреси доставки і контактної інформації;
- зворотній зв'язок - модуль повинен мати функцію зворотнього зв'язку;
- політика конфіденційності - має бути встановлена чітка політика конфіденційності, яка забезпечує захист особистих даних клієнтів.

2.4.1.1.9. Вимоги до кількості користувачів та обслуговуючого персоналу
До системи повинні бути підключені:

- адміністратор системи - це може бути одна або кілька осіб, які керують контентом та налаштовують систему;
- оператори підтримки - кількість операторів залежить від обсягу замовлень та запитів користувачів;
- персонал для обслуговування в ресторані.

Кількість користувачів системи - без обмежень, веб-система повинна бути доступним для всіх, хто відвідує його.

2.4.1.1.10. Вимоги до захисту інформації

Система повинна містити комплексні засоби захисту інформації (КСЗІ), що пройшли державну атестацію. Система повинна бути сертифікованою щодо класу захисту інформації АС-3 для систем, що обробляють інформацію з обмеженим доступом, яка не містить державну таємницю.

Спеціалізоване програмне забезпечення системи повинно отримати експертний висновок ДСТЗІ на відповідність вимогам щодо захисту інформації, яка не містить державну таємницю.

2.4.1.2. Вимоги до безпеки.

Для забезпечення безпеки при експлуатації, налагодженні, монтажі, обслуговуванні і ремонті технічних засобів системи потрібно дотримуватись вимог ДСТУ: ДСТУ 2293:2014, ДСТУ EN ISO 7010:2019, ДСТУ 12.0.230:2008, ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011; по доступним рівням освітленості, вібраційних і шумових навантажень слід дотримуватися вимог відповідно ДСТУ Б А.3.2-15:2011, ДСТУ EN 14253:2018, ДСТУ 2867-94.

2.4.1.3. Вимоги з ергономіки та технічної естетики.

Загальні ергономічні і естетичні вимоги до системи повинні відповідати держстандартам ДСТУ 8604:2015, ДСТУ 7298:2013. Освітленість робочого місця повинна відповідати ДСТУ EN 12464-1:2016, ДБН В.2.5-28:2018.

Засоби відображення повинні розміщуватися таким чином, щоб кут спостереження екрану складав не більше, ніж 45 градусів, мінімальна відстань спостереження екрану — 0,3 м, рекомендована — 0,5 м.

При розробленні ПЗ слід створити зручний інтерфейс для запобігання втомлюваності користувача.

2.4.1.4. Вимоги по експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи.

2.4.1.4.1. Види обслуговування системи визначаються у відповідності з ДСТУ EN 13306:2019. Загальні вимоги по експлуатації, технічному обслуговуванню і ремонту повинні відповідати ДСТУ 3576-97.

2.4.1.4.2. Для розміщення технічних засобів системи необхідні площі, визначені в ДБН В.2.2-9:2018. При цьому слід дотримуватися вимог, зазначених в експлуатаційній документації. Напруга живлення технічних засобів системи 220/380 В змінного струму, частотою (50 ± 1) Гц. Допустиме відхилення напруги від +10 до -15%, тривалість перерв у живленні не повинна перевищувати 0,001 с.

2.4.1.4.3. Кількість, кваліфікація і режими роботи обслуговуючого персоналу повинні відповідати рекомендаціям, зазначеним в технічних умовах і інструкціях з експлуатації окремих ТЗ.

2.4.1.4.4. Склад, розміщення і умови зберігання компонентів технічних засобів системи визначається рекомендаціями, зазначеними в експлуатаційній документації на ці елементи.

2.4.1.4.5. Регламент обслуговування повинен відповідати їх рівню і умовам роботи, щоб у випадку відмови системи забезпечити роботу в аварійному режимі.

2.4.1.5. Вимоги по захисту від впливу зовнішніх діянь.

2.4.1.5.1. Електрична складова електромагнітного поля завад в приміщеннях не повинна перевищувати $0,3 \text{ В/м}^2$ в діапазоні частот від 0,15 до 300 МГц. Для захисту від впливу електромагнітних полів та індустриальних завад слід передбачити різноманітні екрани та фільтри.

2.4.1.5.2. Засоби, які виключають вплив шкідливих факторів на функціонування комплексу технічних засобів, повинні бути запроектовані згідно з ДБН В.2.2-9-2009. Обчислювальні засоби по стійкості до зовнішніх впливів повинні відповідати ДСТУ 2506-94.

2.4.1.6. Вимоги до патентної чистоти.

При створенні даної системи патентні дослідження не проводяться.

2.4.1.7. Вимоги по стандартизації і уніфікації.

У системі кодування інформації необхідно проводити за світовим класифікатором і стандартом.

2.4.2. Вимоги до функцій.

2.4.2.1. Перелік функцій із зазначенням вхідної та вихідної інформації наведено в таблиці 1.

Функції мають забезпечити раціональну організацію роботи користувача на основі безперервної технології: заповнення БД, довідників, формування різноманітних звітів і виконання інших функцій, визначених чинним документом.

При цьому пріоритетом є зручність введення та використання інформації користувачем за рахунок формування підказок і меню на екрані монітора.

Таблиця 2.1. Перелік функцій, вхідної та вихідної інформації

№	Найменування функції	Вхідна інформація	Вихідна інформація
1	Бронювання столика	Дані користувача: ім'я, прізвище, дата, час, кількість місць	Підтвердження успішного бронювання
2	Заповнення анкети для кар'єри	Дані користувача: ім'я, прізвище, резюме, посада	Заповнена анкета
3	Додавання страви в кошик	Вибір страви, кількість порцій	Оновлений список страв у кошику, загальна вартість
4	Оформлення замовлення	Адреса, ресторан/самовивіз, оплата	Підтвердження замовлення
5	Адміністрування страв	Інформація про страви	Оновлене меню
6	Перегляд даних про ресторани	Локація користувача, вибраний ресторан	Інформація про ресторан, маршрут до ресторану
7	Додавання відгуків	Рейтинг, коментарі	Підтвердження отримання відгуку

2.4.3. Вимоги до видів забезпечення.

2.4.3.1. У вимогах до математичного забезпечення (МЗ) система не вимагає спеціального математичного забезпечення для реалізації покладених на неї функцій. Достатньо можливостей обраної СУБД.

2.4.3.2. Вимоги до інформаційного забезпечення (ІЗ).

2.4.3.2.1. Інформаційне забезпечення системи повинно містити дані, достатні для виконання всіх покладених на систему функцій. ІЗ повинно гарантувати раціональну організацію зберігання інформації та доступу до неї.

Заповнення БД інформацією покладається на замовника за методиками і формами, створеними розробниками системи.

Склад, структура і спосіб організації інформації представляються у логічній моделі БД і можуть уточнюватись на етапі технічного проектування.

2.4.3.2.2. Слід передбачити захист даних від руйнування при аваріях і порушеннях у енергоживленні системи — використання резервних копій БД.

2.4.3.3. Вимоги до лінгвістичного забезпечення (ЛЗ).

2.4.3.3.1. Для розробки програмних засобів, які реалізують виконання функцій і забезпечують сервіс користувачів повинні використовуватися мови високого рівня, які забезпечують створення структурних програм, а також мова обраної СУБД для здійснення доступу та маніпулювання даними.

2.4.3.3.2. Організація діалогу користувача до системи має будуватися на наборах меню і підказок, орієнтованих на виконання користувачем функцій. Запити користувача до системи повинні задаватись переважно природною мовою.

2.4.3.4. Вимоги до програмного забезпечення (ПЗ).

2.4.3.4.1. Загальносистемне ПЗ має забезпечувати надійне і якісне виконання функціональних завдань системи. До загальносистемного ПЗ належить:

- операційна система (далі ОС) - Windows;
- система управління БД (далі СУБД) – MySQL.

Веб-розробка та дизайн:

- html/css - для створення структури та стилю веб-системи;
- Javascript - для інтерактивності та додавання функціональності;
- фреймворки - використання популярних фреймворків для стилізації tailwind;

- графічні програми - для створення графіки, візуальних елементів – Figma;

2.4.3.4.2. Загальні вимоги до системного ПЗ можна сформулювати так:

- мінімальні вимоги до ресурсів технічних засобів (ТЗ);
- максимальна швидкодія;
- повне задоволення потреб функціональних завдань системи.

2.4.3.4.3. Вимоги до ОС:

- мінімальне використання ресурсів комп'ютера для власних потреб, передусім оперативної і дискової пам'яті;
- максимальна швидкодія при управлінні зовнішніми пристроями;
- ос сервера - Windows, ОС клієнта - Windows.

2.4.3.4.4. Вимоги до СУБД:

- максимальне задоволення потреб функціональних задач;
- надійність;
- ефективне управління потрібного обсягу і структури;
- швидкість виконання запитів користувачів;
- мінімальні вимоги до ТЗ.

2.4.3.5. Вимоги до технічного забезпечення.

2.4.3.5.1. Технічні засоби системи (табл. 2) повинні забезпечувати виконання функцій, перерахованих в таблиці 1.

2.4.3.5.2. Засоби обчислювальної техніки повинні забезпечувати обмін інформації в об'ємах, приведених в п. 4.3.2.

Таблиця 2.2. Вимоги до технічного забезпечення системи

№ п/п	Основні характеристики комп'ютера
Технічне забезпечення для сервера	
1	Asus Intel(R) Core(TM) i5-8300H CPU 2.30GHz 2.30 GHz
Технічне забезпечення для клієнта	
1	Athlon QL-65 Dual Core 2,1 GHz; RAM: 2048 Mb; HDD: 250 Gb;

Продовженн таблиці 2.2. Вимоги до технічного забезпечення системи

2	Монітор 15"
3	Миша USB
4	Клавіатура USB

2.4.3.6. Вимоги до метрологічного забезпечення.

Система не має вимірювальних каналів, вимірювального обладнання і приладів, тому вимоги до даного виду забезпечення не висуваються.

2.4.3.7. Вимоги до організаційного забезпечення.

2.4.3.7.1. Організаційне забезпечення системи розробляється в відповідності з вимогами державного стандарту по АСУП.

2.4.3.7.2. При впровадженні системи не передбачається збільшення штатної чисельності підприємства. Територіальне розміщення робочих місць, на яких буде встановлена система, визначається підприємством.

2.4.3.7.3. До функціонування системи висуваються наступні вимоги:

- наказом директора визначається список співробітників, які мають доступ до системи;
- контроль і прийняття рішень при аварійних ситуаціях при експлуатації системи здійснює відповідальний за систему.

2.5. Склад і зміст робіт по створенню системи.

2.5.1. Стадії створення системи і терміни виконання робіт наведені в таблиці 3.

Таблиця 2.3. Найменування робіт при створенні системи

№ п/п	Найменування робіт	Строки виконання робіт
1	Затвердження теми роботи	19.01.2024
2	Огляд існуючої методології	15.02.2024
3	Розробка теоретичної основи	30.03.2024

Продовження таблиці 2.3. Найменування робіт при створенні системи

4	Програмна реалізація	02.05.2024
5	Оформлення документації	29.05.2025

2.6. Порядок контролю і приймання системи.

2.6.1. Види, склад, об'єми і методи випробувань системи та її складових частин.

У процесі розробки системи виконуються наступні види випробувань:

- тестування елементів системи;
- тестування системи у цілому;
- дослідна експлуатація.

Крім того, при впровадженні комплексної системи захисту інформації, проводяться додаткові випробування, передбачені інструкціями ДСТЗІ.

2.6.2. Загальні вимоги до приймання робіт за стадіями

Після закінчення відповідного етапу робіт формується комплект документації.

Завершення етапу фіксується відповідним протоколом між Виконавцем і Замовником.

2.7. Вимоги до складу і змісту робіт із підготовки до впровадження системи.

Для введення в дію замовник виконує ряд робіт із підготовки об'єкта:

- проводить укомплектування технічних засобів;
- організовує навчання користувачів системи роботі на ПК і вивчення інструкції з її експлуатації;
- проводить дослідну експлуатацію і вводить систему в дію.

2.8. Вимоги до документації.

2.8.1. На систему розробляється комплекс документації у складі: технічне завдання та технічний проект.

2.8.2. Документація на систему розробляється у відповідності з вимогами Державних стандартів серії 19 «Єдина система програмної документації» та серії 24 «Єдина система стандартів автоматизованих систем управління».

2.9. Джерела розробки.

2.9.1. При розробленні технічного завдання на систему використано наступні документи:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;
- ДСТУ 3973–2000 Система розроблення та поставлення продукції на виробництво;
- ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом.

РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1. Інформаційне забезпечення системи

Веб-орієнтована система для підтримки закладу громадського харчування була розроблена із застосуванням наступних програмних інструментів:

- WordPress;
- phpMyAdmin;
- figma;
- мови HTML, CSS, JavaScript;
- фреймворк для стилізації tailwind;
- препроцесор scss;
- середовище розробки PhpStorm.

Для створення основної структури та функціональності системи було використано засоби WordPress. Він надав швидкий та ефективний спосіб створення та управління вмістом, роблячи додавання, редагування та видалення контенту легкими та простими діями [6].

Для збереження різноманітної інформації, такої як дані користувачів, відгуків, ми використали базу даних MySQL. Для керування цією базою даних ми використовували інструмент phpMyAdmin, який забезпечує зручний веб-інтерфейс для роботи з MySQL, дозволяючи нам легко створювати таблиці, виконувати запити та здійснювати інші операції з базою даних безпосередньо через браузер [7].

Дизайн інтерфейсу (Додаток А) був розроблений за допомогою інструменту Figma з причини його широкого застосування у сучасному дизайні веб-інтерфейсів. Figma відома своєю потужною функціональністю та інтуїтивним інтерфейсом, що робить її найбільш популярним інструментом для дизайну в IT-індустрії [8].

Завдяки цим інструментам досягнута ціль ефективно управляти веб-системою, забезпечивши зручний інтерфейс для користувачів.

Структура включає в себе наступні елементи:

1. Таблиця *wp_commentmeta* - це одна з таблиць у базі даних WordPress, яка використовується для зберігання додаткової мета-інформації про коментарі, що залишають користувачі у вашій системі.
2. Таблиця *wp_db7_forms* - це таблиця, яка використовується плагіном Contact Form 7 для WordPress. Вона зберігає інформацію про створені форми зв'язку, такі як назва форми, код шорткоду, дату створення та іншу мета-інформацію, необхідну для правильного відображення та функціонування форм на сторінках вашої системи. Кожен запис у цій таблиці відповідає окремій формі, яка була створена або налаштована в адмін панелі WordPress за допомогою плагіну Contact Form 7.
3. Таблиця *wp_links* у WordPress використовується для зберігання посилань на сайтові ресурси. Ця таблиця містить інформацію про посилання на інші сайти, які додані або використовуються на вашому сайті WordPress.
4. Таблиця *wp_options* у системі WordPress - це одна з основних таблиць у базі даних, де зберігаються різноманітні налаштування, параметри і дані, необхідні для роботи вашої системи на WordPress.
5. Таблиця *wp_postmeta* - це одна з ключових таблиць у базі даних WordPress. Вона використовується для зберігання додаткової мета-інформації про записи (пости) у вашій системі.
6. Таблиця *wp_termmeta* в WordPress використовується для зберігання метаданих (додаткових даних) термінів, які можуть бути прив'язані до таких об'єктів, як терміни таксономій (такі як категорії, мітки тощо).
7. Таблиця *wp_terms* - це одна з основних таблиць у базі даних WordPress, яка використовується для зберігання інформації про терміни (або теги).
8. Таблиця *wp_term_relationships* - це одна з ключових таблиць у базі даних WordPress, яка відповідає за відношення між записами (post) і термінами (term).

3.2. Алгоритмізація та реалізація комплексу задач автоматизації

Користувач може взаємодіяти з системою, використовуючи переходи, доступні через навігаційне меню. Коли користувач заходить в систему він бачить головну сторінку, на якій зможе обрати подальші дії (рис.3.2).

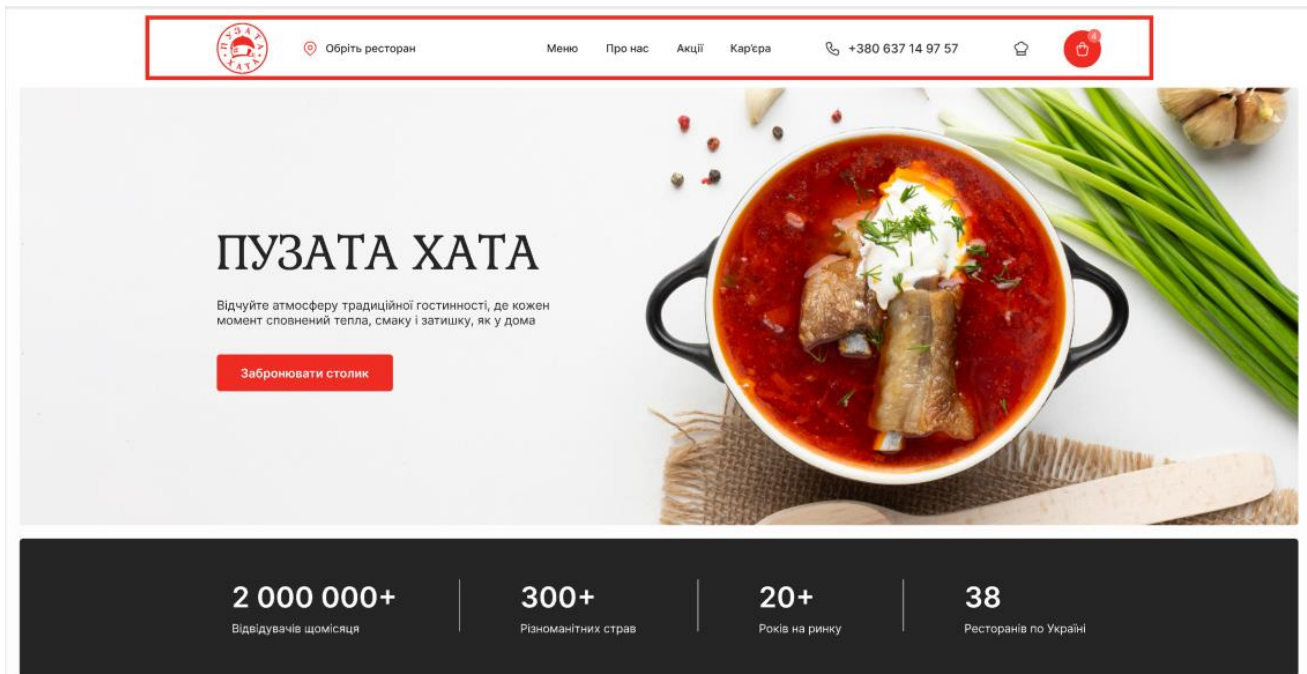


Рисунок 3.2 - Головне меню з навігацією

Файл програми з назвою `header.php` відповідає за встановлення адрес сторінок. У ньому знаходяться всі URL-адреси, необхідні для доступу до різних сторінок. Нижче представлений програмний код цього файлу:

```
<?php wp_nav_menu(array(
    'theme_location' => 'primary',
    'container' => false,
    'menu_class' => false,
    'items_wrap' => '%3$s',
    'walker' => new Custom_Walker_Nav_Menu()
));?>
```

Розробка веб-орієнтованої системи включала використання HTML, CSS і JavaScript з різними цілями:

- `html` (HyperText Markup Language) - стандартизована мова розмітки документів для перегляду вебсторінок у браузері. Він використовувався для створення структури системи. За допомогою

HTML було створено основні елементи сторінки, такі як заголовки, параграфи, списки. HTML допоміг організувати та відобразити вміст системи [9];

- *css* (Cascading Style Sheets) - спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. CSS було використано для стилізації системи. Він дозволив задавати вигляд і відображення елементів HTML, таких як кольори, шрифти, розміри, відступи та інші властивості [10];
- *javascript* - динамічна, об'єктно-орієнтована прототипна мова програмування. Javascript було використано для додавання інтерактивності до системи. Це означає, що за допомогою JavaScript було створено різноманітні функції і ефекти, такі як анімація, переходи між сторінками, валідація форм та взаємодія з користувачем через різні події, такі як кліки мишею або введення тексту [11];

Також використовувалися деякі плагіни WordPress для розширення його функціональності:

- *carbon fields* для створення користувацьких полів - *carbon fields*, бібліотека для легкого створення кастомних (мета) полів в адміністративній панелі WordPress. Вона дозволяє розробникам тем пов'язувати мета-інформацію з різними об'єктами на сайті WordPress (такими як пости, терміни таксономії, віджети і так далі [12];
- *contact form 7* для створення форм і прийому заявок - *contact form 7*, популярний плагін WordPress, за допомогою якого було додано до системи форми зворотного зв'язку. Contact Form 7 має можливість обробляти численні контактні форми, дозволяючи легко налаштовувати як саму форму, так і вміст листа [13].

У веб-орієнтованій системі реалізовано функцію бронювання столику, користувач має заповнити відповідну форму (рис.3.3), аби вона з'явилася, потрібно натиснути на кнопку - *Забронювати столик* на головному екрані. Цю

форму було створено за допомогою плагіна Contact form 7, і виведено її на сторінку за допомогою шорткодів, також було створено валідацію для перевірки введених користувачем даних на відповідність встановленим критеріям:

Код шорткодів для форми бронювання столика:

```
<?= do_shortcode(['contact-form-7 id="07fbdbc" title="Бронювання столика"]) ?>
```

Рисунок 3.3 - Форма бронювання столику у звичайному стані, та при валідації

Код форми бронювання столику:

```
<div class="form-inputs">
  <div class="form-field">
    <label for="book-username">Ваше ім'я<span
class="required">*</span></label>
    <div class="input-container">
      [text* name id:book-username placeholder "Наприклад, Артем"]
      <span class="input-error">Заповніть це поле, будь ласка</span>
    </div>
  </div>
```

```

    <div class="form-field">
      <label          for="book-phone">Ваш          номер          телефону<span
class="required">*</span></label>
      <div class="input-container">
        [tel* phone id:book-phone class:phone placeholder "+380"]
        <span class="input-error">Заповніть це поле, будь ласка</span>
      </div>
    </div>
    <div class="form-field">
      <label          for="book-time">Оберіть          час<span
class="required">*</span></label>
      <div class="input-container">
        <input required type="time" id="book-time">
        <span class="input-error">Заповніть це поле, будь ласка</span>
      </div>
    </div>
    <div class="form-field">
      <label          for="book-date">Оберіть          дату<span
class="required">*</span></label>
      <div class="input-container">
        [date* date id:book-date placeholder "Дата"]
        <span class="input-error">Заповніть це поле, будь ласка</span>
      </div>
    </div>
    <div class="book-form__quantity">
      <span>Кількість гостей</span>
      <div class="book-quantity js-change-quantity"><button type="button"
class="btn js-q-minus icon-btn"><i class="i-minus"></i></button>[number*
persons_count id:book-persons placeholder "1"]<button type="button" class="btn
js-q-plus icon-btn"><i class="i-plus"></i></button>
      </div>
    </div>
    [submit class:btn class:primary class:lg class:w-full class:mt-0
"Забронювати столик"]
  </div>

```

Код валідації форми бронювання столику буде розміщений у (Додаток Б). Аналогічним чином валідація розроблена для інших форм.

Користувач може залишати відгуки за допомогою відповідної форми (рис. 3.4), для цього потрібно натиснути кнопку - *Написати відгук*, після цього з'являється спеціальна форма де користувач, може залишити свій відгук. Цю

форму також було створено за допомогою плагіна Contact form 7, і виведено її на сторінку за допомогою шорткодів:

Код шорткодів для форми написання відгуку:

```
<?= do_shortcode('[contact-form-7 id="8a0bd03" title="Форма відгуків"]')
?>
```

Рисунок 3.4 - Форма написання відгуку у звичайному стані, та при валідації

Код форми написання відгуку:

```
<div class="form-inputs">
  <div class="form-field">
    <label for="review-your-name">Ваше ім'я<span
class="required">*</span></label>
    <div class="input-container">
      [text* name id:review-your-name placeholder "Наприклад, Артем"]
      <span class="input-error">Заповніть це поле, будь ласка</span>
    </div>
  </div>
  <div class="form-field">
    <label for="review-your-text">Ваш відгук<span
class="required">*</span></label>
    <div class="input-container">
      [textarea* your-review id:review-your-text rows:10 placeholder "Напишіть
щось хороше про нас :)"]
      <span class="input-error">Заповніть це поле, будь ласка</span>
    </div>
  </div>
</div>
```

```

<div class="rating-container">
<div class="text-center my-15"><span>Моя оцінка</span></div>
<div class="rating">
<input type="radio" name="rating" value="5" id="rating-5" required>
<label for="rating-5">
<i class="i-star"></i>
</label>
<input type="radio" name="rating" value="4" id="rating-4" required>
<label for="rating-4">
<i class="i-star"></i>
</label>
<input type="radio" name="rating" value="3" id="rating-3" required>
<label for="rating-3">
<i class="i-star"></i>
</label>
<input type="radio" name="rating" value="2" id="rating-2" required>
<label for="rating-2">
<i class="i-star"></i>
</label>
<input type="radio" name="rating" value="1" id="rating-1" required>
<label for="rating-1">
<i class="i-star"></i>
</label>
</div>
</div>
[submit class:btn class:primary class:lg class:w-full class:mt-0 "Надіслати"]

```

Веб-орієнтована система передбачає можливість щоб користувач зміг залишити заявку, при бажанні прикріпити резюме, для того щоб розпочати кар'єру у мережі ресторанів. Для цього йому потрібно заповнити спеціальну анкету (Додаток А), та відпрайти її. Тут використовується плагін wordpress contact form 7 для створення форм і прийому заявок.

Код форми заповнення анкети буде наведено у додатку Б

Якщо користувач, хоче обрати конкретне місто, та ресторан він повинен в меню навігації натиснути на - *Обрати ресторан* (рис. 3.5).

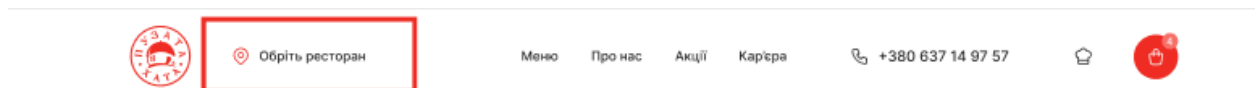
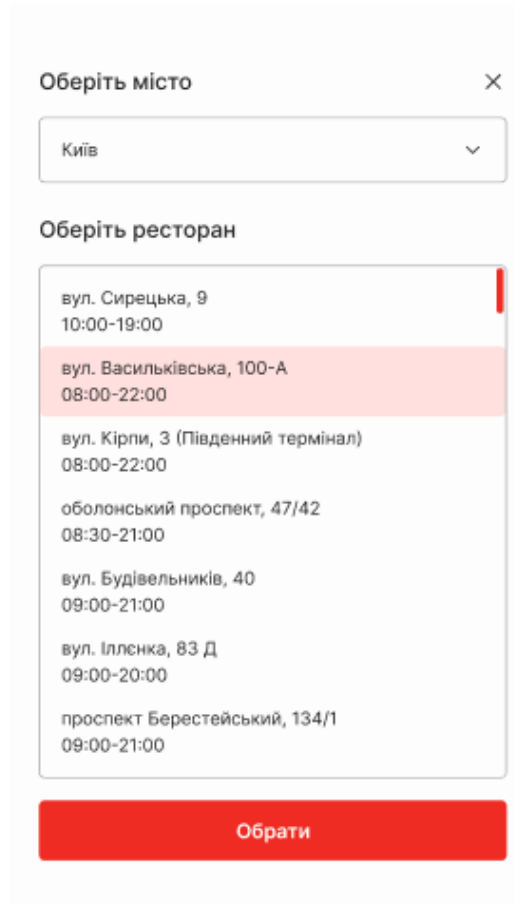


Рисунок 3.5 - Навігаційне меню

Після чого користувач зможе обрати бажане місто та ресторн з випадуючого списку (рис. 3.6):



The screenshot shows a mobile application interface with two main sections. The first section is titled "Оберіть місто" (Select city) and contains a dropdown menu with "Київ" (Kyiv) selected. The second section is titled "Оберіть ресторан" (Select restaurant) and contains a list of restaurant options, each with an address and operating hours. The second option, "вул. Васильківська, 100-А" (Vasylkivska St, 100-A) with hours "08:00-22:00", is highlighted in red. At the bottom of the interface is a red button labeled "Обрати" (Select).

Address	Hours
вул. Сирецька, 9	10:00-19:00
вул. Васильківська, 100-А	08:00-22:00
вул. Кірпи, 3 (Південний термінал)	08:00-22:00
оболонський проспект, 47/42	08:30-21:00
вул. Будівельників, 40	09:00-21:00
вул. Ілленка, 83 Д	09:00-20:00
проспект Берестейський, 134/1	09:00-21:00

Рисунок 3.6 - Форма обрання бажаного міста та ресторану

Фрагмент коду функції для вибору міста та ресторану буде представлений у програмному лістингу (Додаток Б).

Веб-орієнтована система передбачає можливість переглядати ресторани, та дані про них, натиснувши на цю кнопку (рис. 3.7) у навігаційному меню користувач потрапляє на сторінку з рестоорнанами:



Рисунок 3.7 - Кнопка переходу на сторінку ресторанів

Користувачеві відкриється сторінка (Додаток А) де можна обирати потрібне йому місто, та переглядати інформацію про відповідні йому ресторани. Користувач зможе побачити інформацію про конкретний ресторан, наприклад: чи відчинений він, години роботи, адресу.

Години роботи відображаються у режимі реального часу, тобто, якщо зараз час відповідний, то користувач побачить, що ресторан відкритий, якщо час не відповідатиме вказаному у ресторані (рис. 3.8), то користувач побачить, що він зачинений.

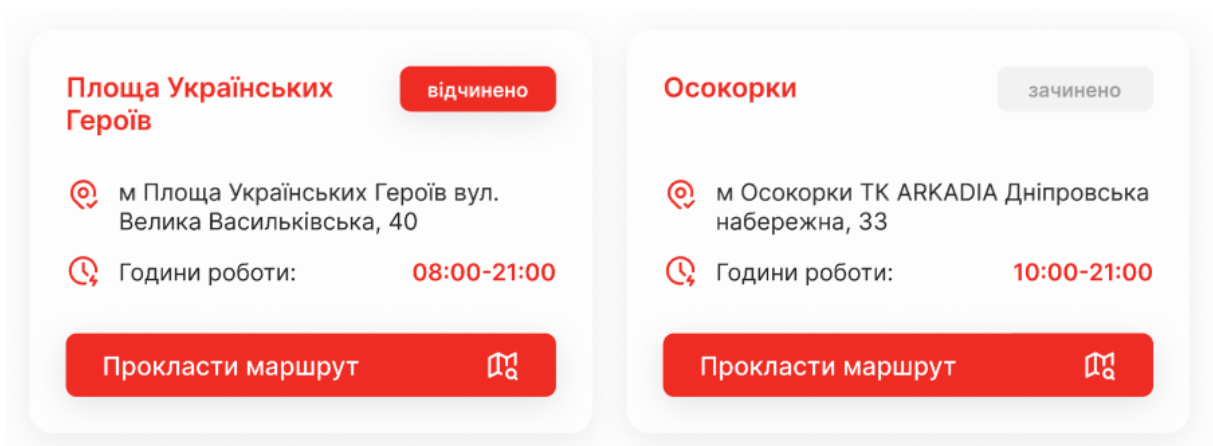


Рисунок 3.8 - Статус годин робіт ресторанів

Реалізації функції відображення статусу ресторану у режимі реального часу:

Код формування масиву ресторанів:

```
function get_restaurants($city): array
{
    $cities = carbon_get_theme_option('cities');
    $restaurants_array = [];
    foreach ($cities as $city_data) {
        if ($city_data['name'] == $city) {
            foreach ($city_data['restaurants'] as $restaurant) {
                $restaurants_array[] = array(
                    'name' => $restaurant['name'],
                    'address' => $restaurant['address'],
                    'address_link' => $restaurant['address_link'],
                    'opens_at' => date('H:i', strtotime($restaurant['opens_at'])),
                    'closes_at' => date('H:i', strtotime($restaurant['closes_at'])),
                    'is_opened' => is_rest_opened($restaurant),
                );
            }
        }
    }
}
```



```

        </a>
    </div>
</li>
<?php endforeach; ?>
<?php endif; ?>
</ul>
<?php endforeach; ?>

```

Код для перевірки чи відкритий ресторан у поточний момент часу:

```

function is_rest_opened($rest): bool
{
    $now = current_time('H:i');

    $opens_at = date('H:i', strtotime($rest['opens_at']));
    $closes_at = date('H:i', strtotime($rest['closes_at']));
    if ($now >= $opens_at && $now <= $closes_at) {
        return true;
    }
    return false;
}

```

Також у користувача є змога прокласти маршрут до обраного ресторану, для цього потрібно натиснути на кнопку - *Прокласти маршрут* і користувача перекине у Google Maps (рис.3.9) де йому автоматично прокладе маршрут від місця його знаходження до ресторану, але для цього має бути увімкнена функція геоданих.

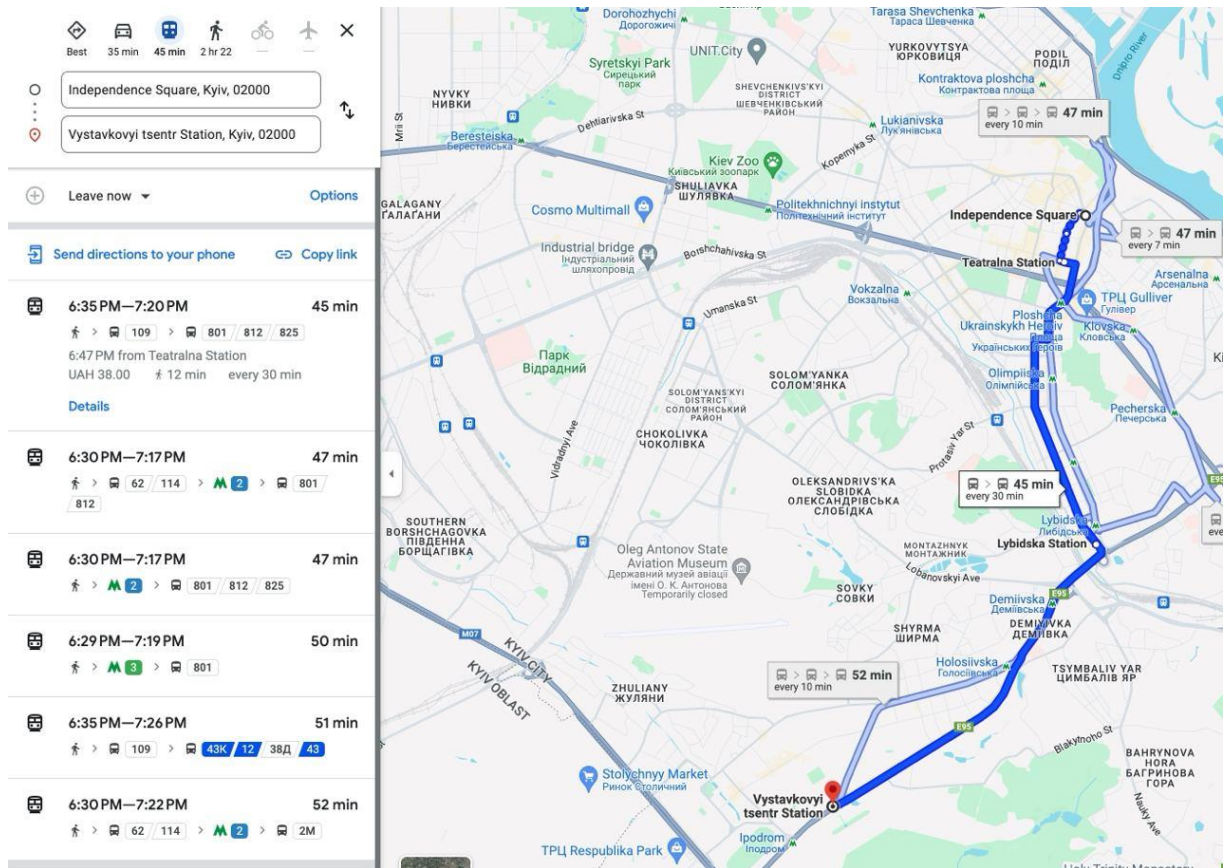


Рисунок 3.9 - Прокладений маршрут до ресторану

Код реалізації функції прокладання маршруту:

```

Container::make('theme_options', __('Ресторани', 'dream-theme'))
->set_icon('dashicons-admin-home')
->add_fields(array(
Field::make('complex', 'cities', __('Міста', 'dream-theme'))
->set_layout('tabbed-horizontal')
->add_fields(array(
Field::make('text', 'name', __('Місто', 'dream-theme')),
Field::make('complex', 'restaurants', __('Ресторани', 'dream-theme'))
->add_fields(array(
Field::make('text', 'name', __('Назва ресторану', 'dream-theme')),
Field::make('text', 'address', __('Адреса ресторану', 'dream-theme'))-
>set_width(50),
Field::make('text', 'address_link', __('Посилання на адресу (прокласти
маршрут)', 'dream-theme'))->set_width(50),
Field::make('separator', 'hours_separator', __('Години роботи', 'cis')),
Field::make('time', 'opens_at', __('Відчиняється о', 'crb'))->set_width(50),
Field::make('time', 'closes_at', __('Закривається о', 'crb'))->set_width(50),
))
->set_header_template( '<% if (name) { %><%- name %> <% } %>' ),

```

```

))
->set_header_template( '<% if (name) { %><%- name %> <% } %>' ),
));

```

Для того, щоб користувач зміг зробити замовлення, він має перейти в режимі навігації у пункт – *Меню* (рис. 3.10).

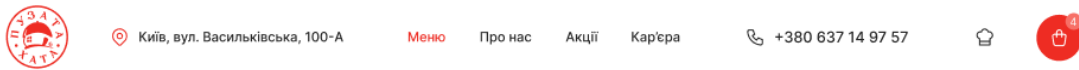


Рисунок 3.10 - Навігаційне меню, пункт меню

Користувачеві відкривається сторінка, де він може побачить актуальні акції, категорії страв, та самі страви (рис. 3.11):

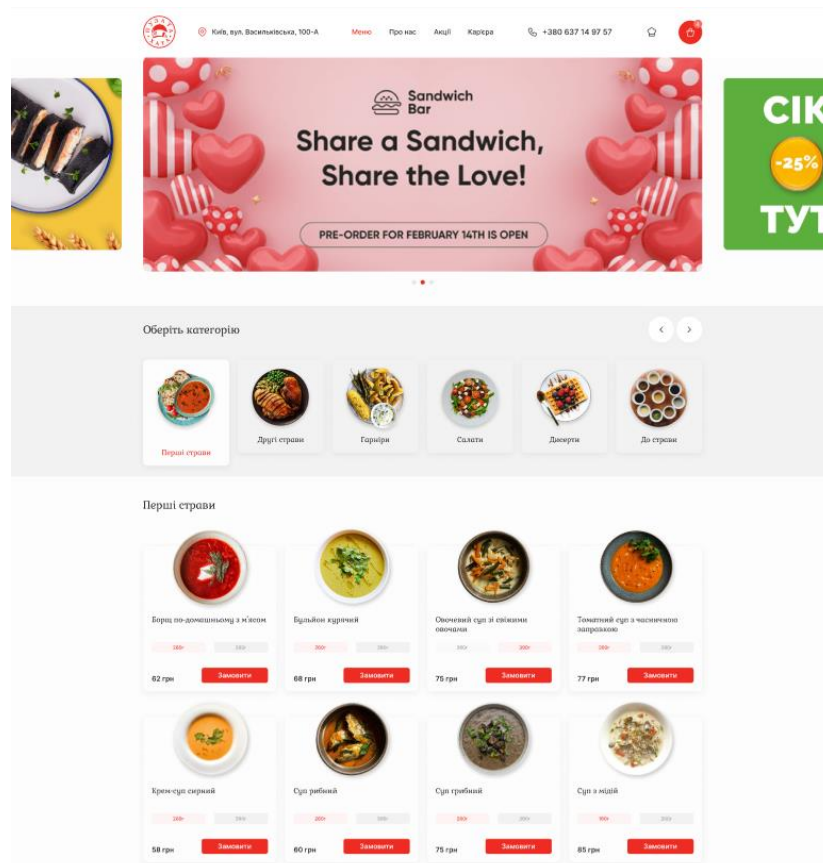


Рисунок 3.11 - Категорії, та страви до них

Для того аби додати обрану страву до кошика потрібно натиснути на кнопку - *Замовити*. Також користувач може обрати більшу або меншу порцію.

Код реалізацій вибору порцій:

```
const productContainers = document.querySelectorAll('.js-product-container');
productContainers.forEach((container) => {
  const variantsButtons = container.querySelectorAll('.js-product-variants-input');
  variantsButtons.forEach((button) => {
    button.addEventListener('change', () => {
      const weightElement = container.querySelector('.js-weight-value');
      const priceElement = container.querySelector('.js-price-value');
      if(weightElement) {
        weightElement.textContent = button.value;
      }
      if(priceElement) {
        priceElement.textContent = button.getAttribute('data-variant-price');
      }
    });
  });
});
```

Також користувач може побачити рекламні банери (рис.3.12), натиснувши на один із них йому буде представлено акційні страви певної категорії. Для створення слайдерів використовується бібліотека Swiper.js [14].

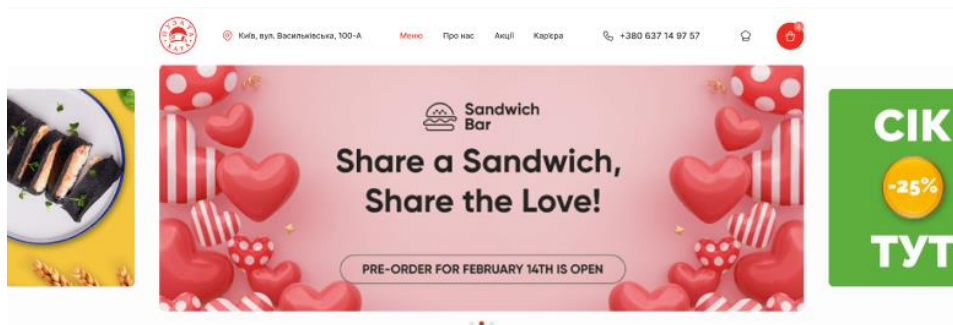


Рисунок 3.12 - Рекламні банери

Код для створення банерів наведено у програмному лістингу (Додатку Б).

Код ініціалізації та налаштування головних слайдерів:

```
const openModalBtns = document.querySelectorAll('[data-target-modal]');
const modalWindows = document.querySelectorAll('.modal');
openModalBtns.forEach((btn) => {
  btn.addEventListener('click', (e) => {
    e.preventDefault();
    let userData = JSON.parse(localStorage.getItem('userData'));
```

```

const isRestaurantRequired = btn.getAttribute('data-restaurant-required')
=== 'true';
const targetModal = btn.getAttribute('data-target-modal');
if (isRestaurantRequired && !userData.restaurant) {
  chooseRestaurantModal.classList.add('active');
  userData.initiator = targetModal;
  localStorage.setItem('userData', JSON.stringify(userData));
  return;
}
const modalWindow = document.querySelector(`[data-
modal="${targetModal}"]`);
openModal(modalWindow);
});
});
modalWindows.forEach((modal) => {
  modal.addEventListener('click', (e) => {
    e.stopPropagation();
    closeModal(e);
  });
});
document.addEventListener('click', (e) => {
  closeModal(e);
});

```

При натисканні на одну з категорій, користувачу виводяться страви обраної категорії (рис.3.13):

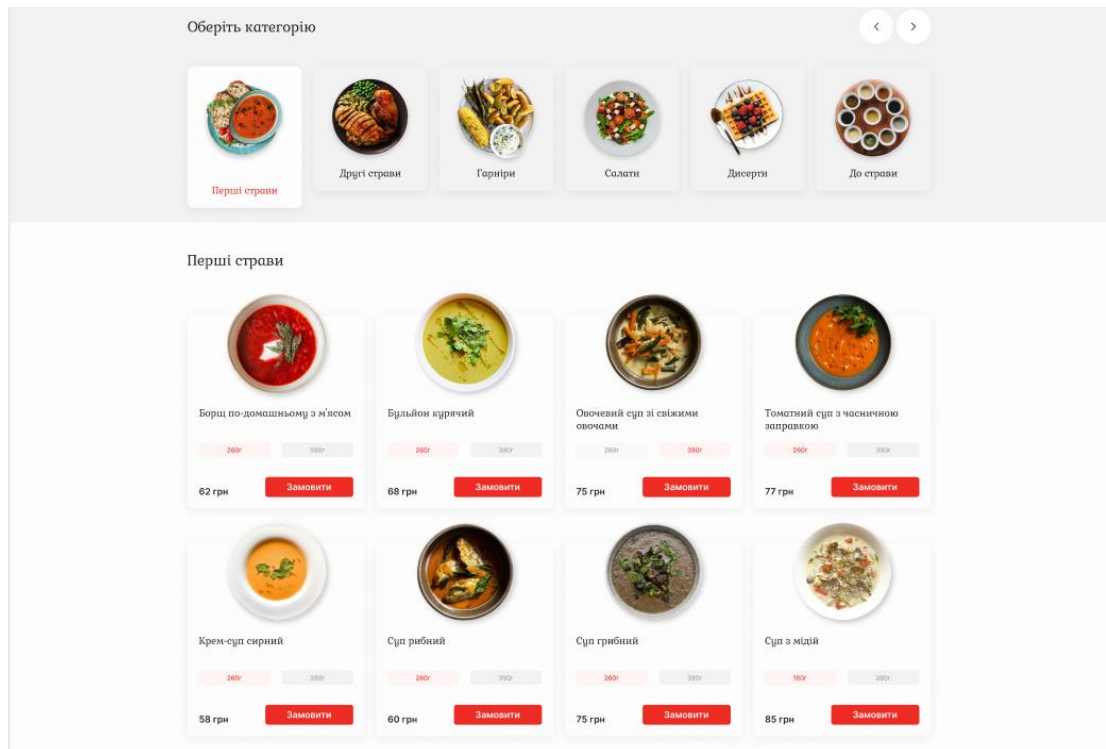


Рисунок 3.13 - Вивід страв до обраної категорії

Код виведення певної категорії страв:

```
<?php
$current_category = get_queried_object();
$current_category_id = $current_category->term_id ?: null;
?>
<section class="category-products">
  <div class="container">
    <div class="section-heading">
      <div class="section-title">
        <h2>Обирайте наші найсмачніші страви</h2>
      </div>
    </div>
    <ul class="category-products__list products-list">
      <?php $products = get_products($current_category_id); ?>
      <?php if(!empty($products)) : ?>
        <?php foreach ($products as $product) : ?>
          <?php render_product_element($product, 'li') ?>
        <?php endforeach; ?>
      <?php endif ?>
    </ul>
  </div>
</section>
```

</section>

Код формування масиву страв:

```
function get_products($category_id = null, $products_ids = null): array
{
    $args = array(
        'post_type' => 'product',
        'posts_per_page' => -1,
    );
    if($products_ids !== null) {
        $args['post__in'] = $products_ids;
    }
    if ($category_id !== null) {
        $args['tax_query'] = array(
            array(
                'taxonomy' => 'product_category',
                'field' => 'term_id',
                'terms' => $category_id,
            )
        );
    }
    $products = get_posts($args);
    $products_array = array();

    foreach ($products as $product) {
        $products_array[] = array(
            'id' => $product->ID,
            'name' => $product->post_title,
            'link' => get_permalink($product->ID),
            'image' => get_the_post_thumbnail_url($product->ID),
            'weights' => carbon_get_post_meta($product->ID, 'product_weights'),
            'ingredients' => carbon_get_post_meta($product->ID,
'product_ingredients'),
            'kcal' => carbon_get_post_meta($product->ID, 'product_kcal'),
            'protein' => carbon_get_post_meta($product->ID, 'product_protein'),
            'fat' => carbon_get_post_meta($product->ID, 'product_fat'),
            'carbohydrate' => carbon_get_post_meta($product->ID,
'product_carbohydrate'),
        );
    }

    return $products_array;
}
```

Картка страви виглядає таким чином (рис. 3.14):

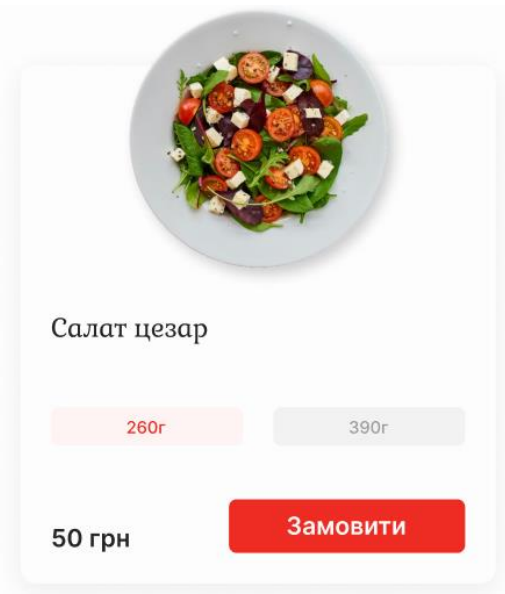


Рисунок 3.14 - Картка страви

Код виводу картки страви:

```
function render_product_element($product, $tag = 'div'): void
{ ?>
    <?php if($tag === 'li') : ?>
        <li class="product-card js-product-container">
    <?php else : ?>
        <div class="product-card js-product-container">
    <?php endif; ?>
        <div class="product-card__body solid-card">
            <div class="product-card__heading">
                <a href="<?= $product['link'] ?>" class="product-
card__image">
                    ">
                </a>
                <div class="product-card__title">
                    <h3><a href="<?= $product['link'] ?>"><?=
$product['name'] ?></a></h3>
                </div>
            </div>
            <?php if(count($product['weights']) > 1) : ?>
                <div class="product-card__variant product-variants js-product-
variants">
                    <?php $random_string = substr(md5(rand()), 0, 7); ?>
                    <?php foreach ($product['weights'] as $key => $variant) : ?>
```

```

        <label for="var-<?= $random_string ?>-<?= $key ?>"
class="btn xs">
            <input <?= $key === 0 ? 'checked' : ''; ?> type="radio"
name="variant-<?= $random_string ?>" id="var-<?= $random_string ?>-<?=
$key ?>" value="<?= $variant['weight'] ?>" data-variant-price="<?=
$variant['price'] ?>">
                <span><?= $variant['weight'] ?>г</span>
            </label>
        <?php endforeach; ?>
    </div>
<?php endif; ?>
<div class="product-card__footer">
    <div class="product-card__price">
        <?php if(count($product['weights']) === 1) : ?>
            <span class="weight sm"><span class="js-weight-
value"><?= $product['weights'][0]['weight'] ?></span>г</span>
            <?php endif; ?>
            <span class="price"><span class="js-price-value"><?=
$product['weights'][0]['price'] ?></span> грн</span>
        </div>
        <button class="btn primary sm">Замовити</button>
    </div>
</div>
<?php if($tag === 'li') : ?>
    </li>
<?php else : ?>
</div>
<?php endif; ?>
<?php }

```

Код реалізації функції додавання страв у кошик:

```

const products = document.querySelectorAll('.js-product-container');
const cartContent = document.querySelector('.js-cart-content')
products.forEach(product => {
    const addToCartBtn = product.querySelector('.js-add-to-cart-btn');
    addToCartBtn?.addEventListener('click', e => {
        e.preventDefault();
        const productId = product.dataset.productId;
        const quantityInput = product.querySelector('input[type="number"]');
        const quantity = quantityInput ? parseInt(quantityInput.value) : 1;

```

```

    const chosenVariant = product.querySelector('.js-product-variants
input:checked').value;
    const variant = chosenVariant ? chosenVariant :
product.querySelector('.js-weight-value').textContent.trim();
    const price = product.querySelector('.js-price-value').textContent.trim();
    updateCart(productId, quantity, variant, price);
    updateElementText(addToCartBtn, 'В кошику');
  });
})
function updateCart(productId, quantity, variant, price) {
  const cart = JSON.parse(localStorage.getItem('cart')) || [];
  const existingProduct = cart.find(item => item.productId === productId
&& item.variant === variant);
  if (existingProduct) {
    existingProduct.quantity += quantity;
  } else {
    cart.push({ productId, quantity, variant, price });
  }
  localStorage.setItem('cart', JSON.stringify(cart));
}

```

Після того, як страви обрані користувач може їх переглянути натиснувши кнопку кошику у навігації (рис. 3.15).



Рисунок 3.15 - Іконка кошику

Перед користувачем відкривається кошик з його обраними стравами, які він зможе переглянути, відредагувати, видалити (рис. 3.16).

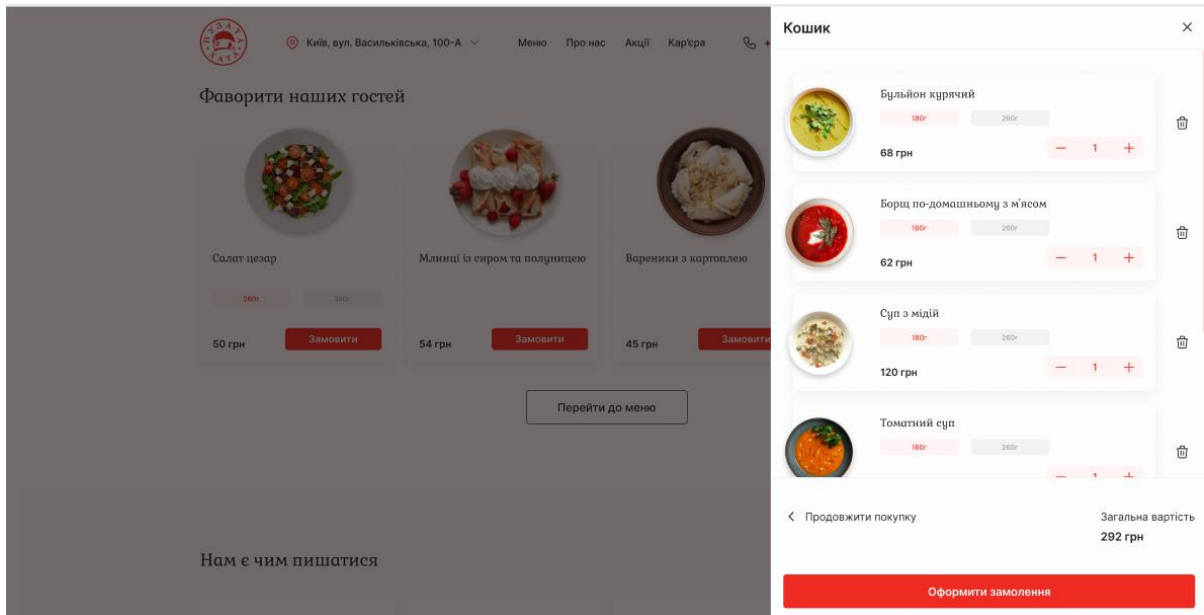


Рисунок 3.16 - Кошик заповнений

Код реалізації видалення страви з кошику:

```
function removeProduct(button) {
    const productContainer = button.closest('.js-product-container');
    const productId = productContainer.dataset.productId;
    const variant = productContainer.querySelector('.js-weight-value').textContent.trim();
    const cart = JSON.parse(localStorage.getItem('cart')) || [];
    const updatedCart = cart.filter(item => !(item.productId === productId
    && item.variant === variant));
    const cartTotal = cartContent.querySelector('.js-cart-total');
    const total = updatedCart.reduce((acc, item) => acc + item.quantity *
    item.price, 0);
    cartTotal.textContent = total;
    localStorage.setItem('cart', JSON.stringify(updatedCart));
    productContainer.remove();
    if (updatedCart.length === 0) {
```

```

document.querySelector('.js-cart-content').innerHTML = emptyCartHtml;
}
}

```

Після натискання кнопки - *Оформити замовлення*, користувач потрапляє на сторінку оформлення замовлення (рис. 3.17).

Рисунок 3.17 - Оформлення замовлення

На цій сторінці користувач повинен оформити замовлення заповнивши спеціальну форму своїми даними при цьому обравши метод отримання замовлення, кур'єром, чи самовивіз, потім обравши зручний час отримання, та спосіб оплати.

Код форми оформлення:

js:

```

function sendOrderToTelegram(e) {
    e.preventDefault();
    const form = document.querySelector('#checkout-form');
    const formData = new FormData(form);
    const userData = JSON.parse(localStorage.getItem('userData'));

```

```
const cart = JSON.parse(localStorage.getItem('cart'));
const city = userData.city;
const restaurant = userData.restaurant;
const total = cart.reduce((acc, item) => acc + item.quantity * item.price, 0);
formData.append('city', city);
formData.append('restaurant', restaurant);
formData.append('total', total);
formData.append('cart', JSON.stringify(cart));
fetch('/wp-admin/admin-ajax.php?action=send_order_to_telegram', {
  method: 'POST',
  body: formData
})
.then(response => response.text())
.then(html => {
  form.querySelector('[type="submit"]').disabled = true;
  localStorage.removeItem('cart');
  if(formData.get('payment_method').includes('БАНКІВСЬКОЮ')) {
    window.location.href = '/success?online_order=1';
  } else {
    window.location.href = '/success';
  }
})
.catch(error => {
  console.error('Error sending order to telegram: ', error);
});
}
```

```

php:
    add_action('wp_ajax_send_order_to_telegram',
'send_order_to_telegram_ajax_handler');
    add_action('wp_ajax_nopriv_send_order_to_telegram',
'send_order_to_telegram_ajax_handler');
    function send_order_to_telegram_ajax_handler() {
        if (!isset($_POST['city']) || !isset($_POST['restaurant'])
|| !isset($_POST['total']) || !isset($_POST['cart'])) {
            wp_send_json_error('Invalid order data', 400);
        }
        $city = sanitize_text_field($_POST['city']);
        $restaurant = sanitize_text_field($_POST['restaurant']);
        $total = sanitize_text_field($_POST['total']);
        $cart = json_decode(stripslashes($_POST['cart']), true);
        function is_pickup(): bool
        {
            return !str_contains($_POST['shipping_method'], 'Самовивіз');
        }
        if (empty($cart) || !is_array($cart)) {
            wp_send_json_error('Invalid cart data', 400);
        }
        $message = "<b>Нове замовлення!</b>\n\n";
        $message .= "Місто: $city\n";
        if(is_pickup()) {
            $message .= "Ресторан: $restaurant\n";
        }
        $message .= "Загальна вартість: $total грн\n\n";
        $message .= "Спосіб доставки: " .
sanitize_text_field($_POST['shipping_method']) . "\n";

```

```

$message .= "Спосіб оплати: " .
sanitize_text_field($_POST['payment_method']) . "\n\n";
$message .= "Час отримання: " . sanitize_text_field($_POST['time']) . "\n";
$message .= "\n<b>Контактна інформація:</b>\n";
$message .= "Ім'я: " . sanitize_text_field($_POST['firstname']) . ' ' .
sanitize_text_field($_POST['lastname']) . "\n";
$message .= "Телефон: " . sanitize_text_field($_POST['phone']) . "\n";
if (is_pickup()) {
    $message .= "\n<b>Адреса доставки:</b>\n";
    $message .= "Вулиця: " . sanitize_text_field($_POST['street']) . "\n";
    $message .= "Будинок: " . sanitize_text_field($_POST['house']) . "\n";
    $message .= "Квартира: " . sanitize_text_field($_POST['flat']) . "\n";
}
$message .= "\n<b>Кошик:</b>\n";
foreach ($cart as $item) {
    $product = get_product($item['productId']);
    if (isset($product)) {
        $message .= "- {$product['name']} " . ($product['id'] != 56 ? '(' ' ' .
$item['variant'] . 'r)') : ") . " x{$item['quantity']}\n";
    } else {
        $message .= "Назва: Невідомо\n";
    }
}
$telegram_bot_token = carbon_get_theme_option('telegram_bot_token');
$telegram_chat_id = carbon_get_theme_option('telegram_chat_id');

$get_query = array(
    "chat_id" => $telegram_chat_id,
    "text" => $message,
    "parse_mode" => "html"

```

```

);
$ch = curl_init("https://api.telegram.org/bot" . $telegram_bot_token .
"/sendMessage?" . http_build_query($get_query));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HEADER, false);
$resultQuery = curl_exec($ch);
curl_close($ch);
if ($resultQuery === false) {
    wp_send_json_error('Failed to send order to Telegram', 500);
}
wp_send_json_success('Order sent to telegram');
}

```

Після заповнення своїх даних користувач повинен натиснути на чекбокс, для згоди на обробку персональних даних, після цього натиснути на кнопку - *Підтвердити оплату*. Користувач потрапляє на сторінку подяки (рис. 3.18) де повинен просканувати Qr-код для переказу коштів за замовлення.

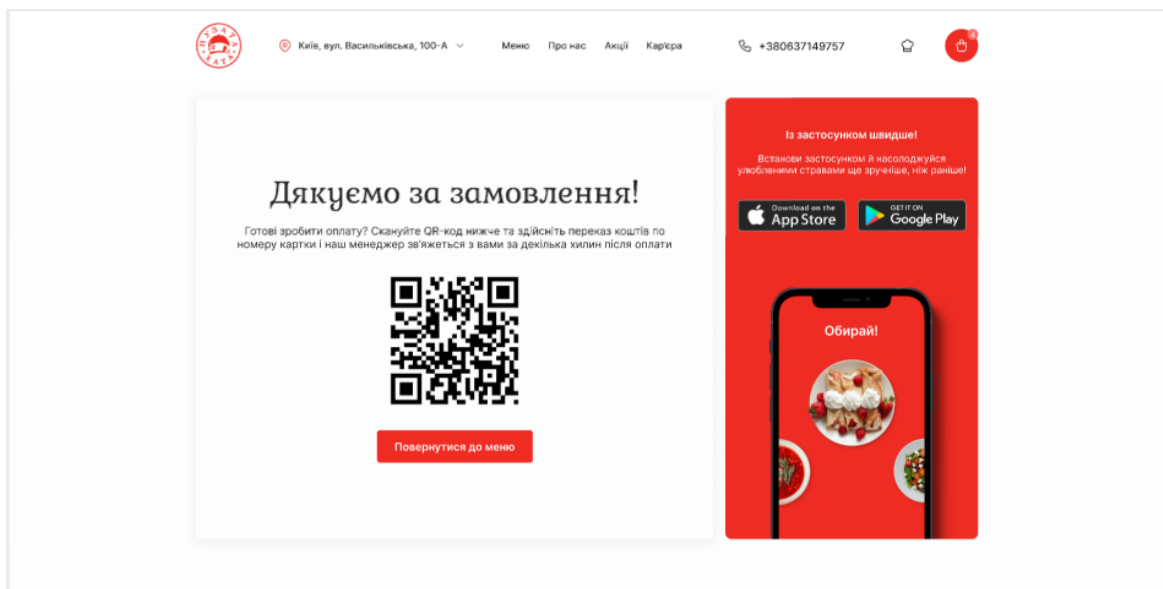


Рисунок 3.18 - Сторінка подяки

Додатково до бази даних було створено телеграм-бота для прийому замовлень, бронювань та відгуків. Цей бот надає можливість адміністраторам

та менеджерам зручно та швидко обробляти запити клієнтів в режимі реального часу. Використання телеграм-бота дозволяє оптимізувати робочий процес, зменшити кількість помилок при обробці інформації та забезпечити високий рівень обслуговування завдяки оперативному реагуванню на потреби клієнтів.

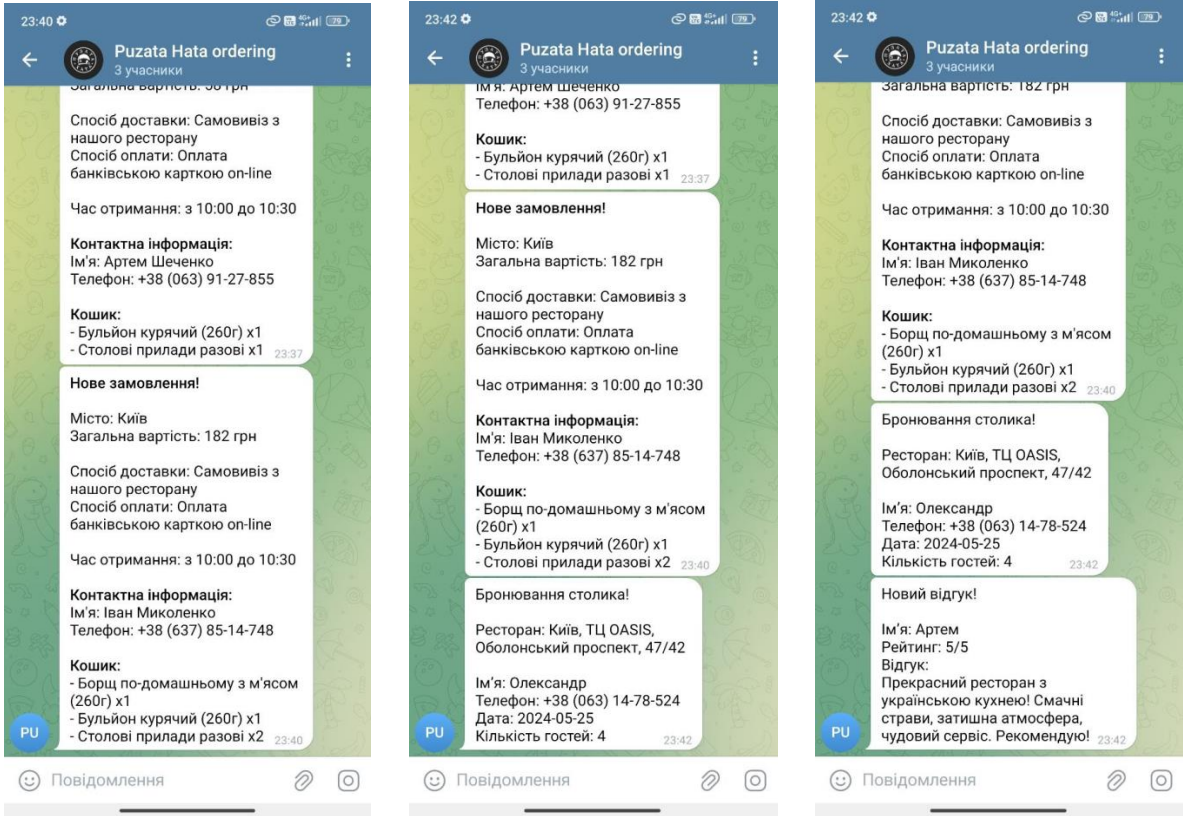


Рисунок 3.19 – Сповіднення про нове замовлення, нове бронювання, новий відгук

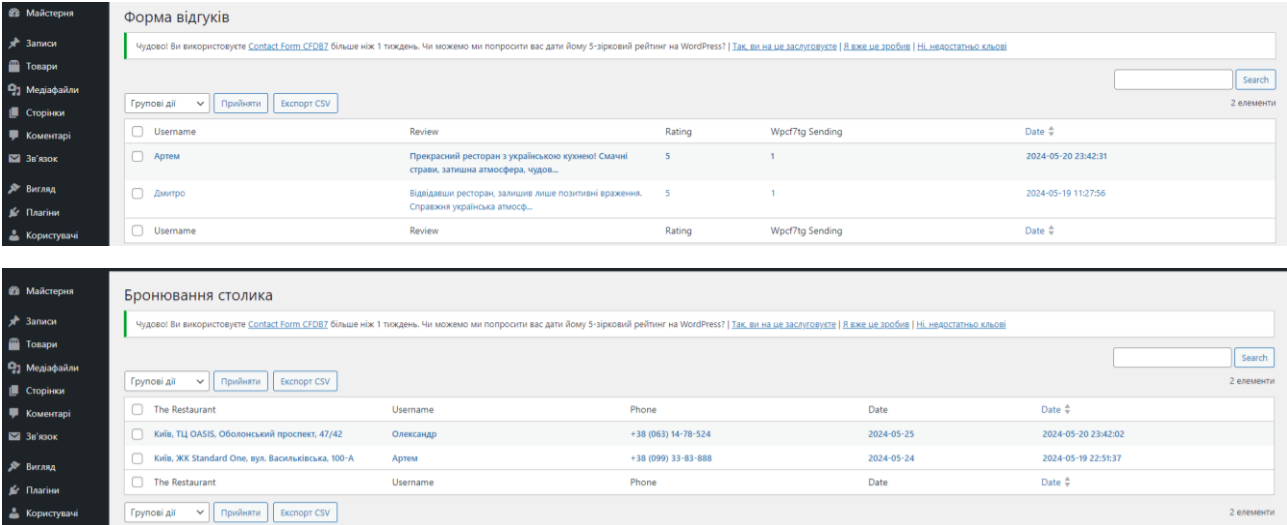


Рисунок 3.20 – Заявки у бази даних

3.3. Інструкція користувача

При вході в систему перед користувачем з'являється головне меню (рис.3.21). У користувача системи є можливість забронювати столик, для цього потрібно клікнути на кнопку - *Забронювати столик*.

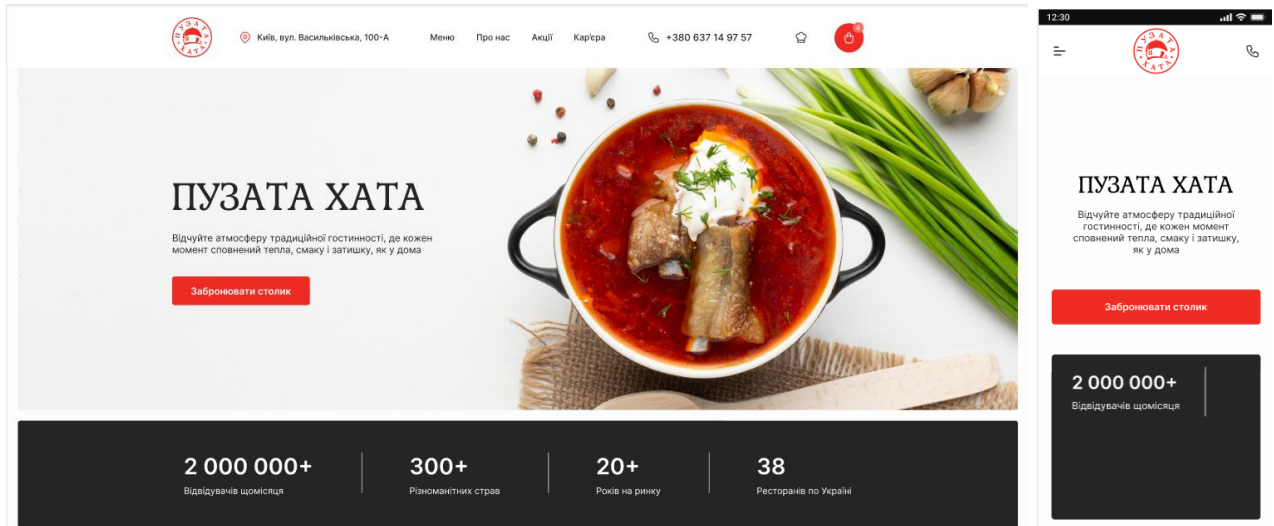


Рисунок 3.21 - Головна сторінка

Після цього користувач побачить форму бронювання (рис.3.22) на якій зможе залишити свої контактні дані, та обрати бажаний час, дату, кількість гостей, та натиснути кнопку - *Забронювати столик*.

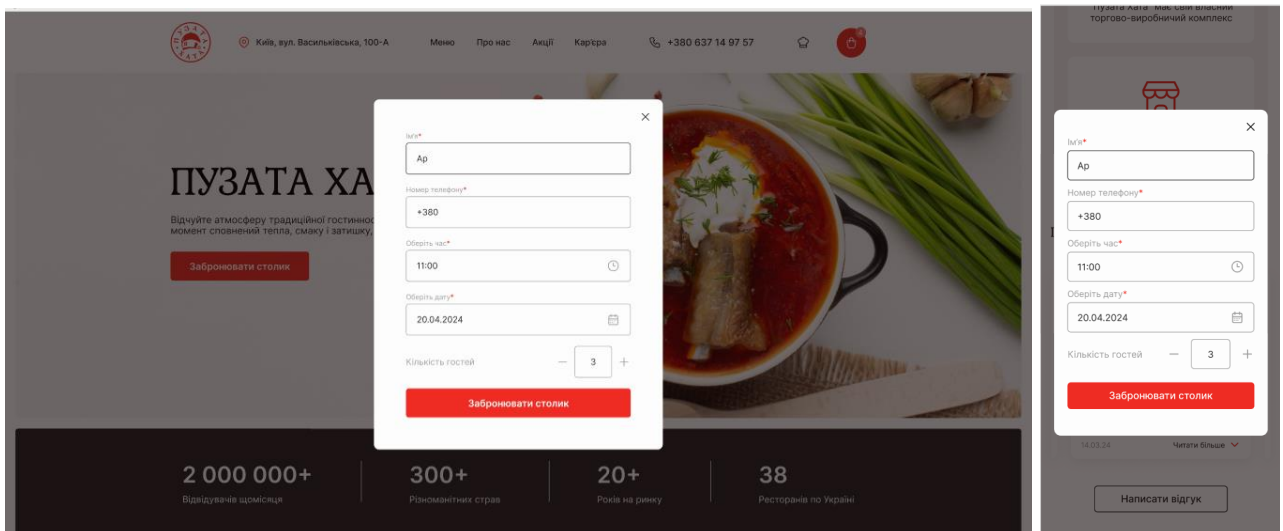


Рисунок 3.22 - Форма бронювання

Після цих дій користувач побачить підтвердження бронювання (рис.3.23).

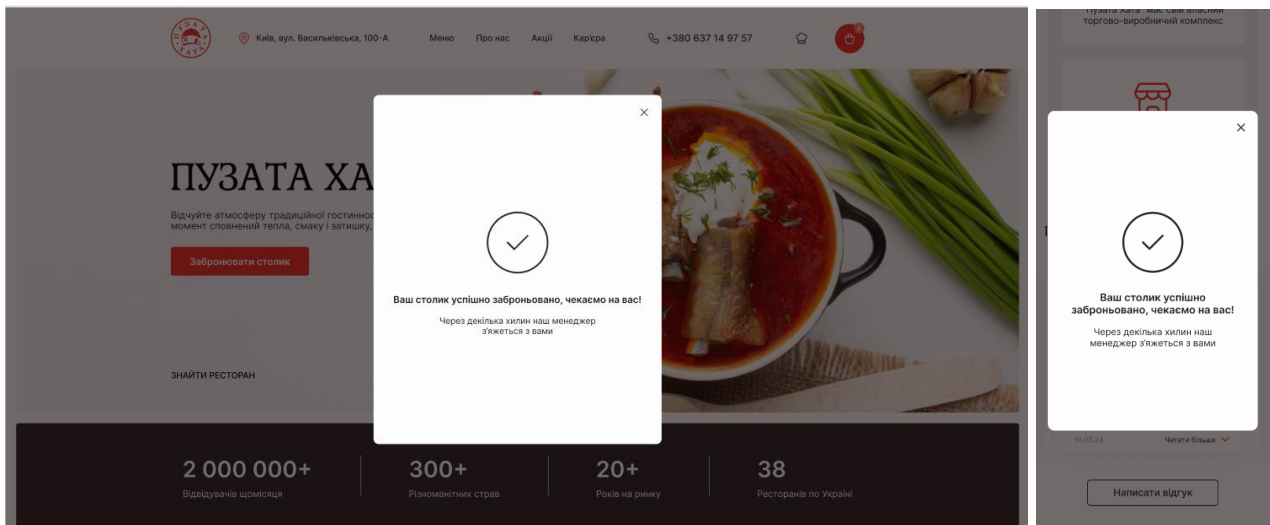


Рисунок 3.23 - Форма підтвердження бронювання

Після бронювання столику клієнту слід очікувати дзвінок менеджера ресторану для узгодження бронювання.

Для переходу між сторінками користувач може скористатися навігацією на початку головної сторінки (рис.3.24).

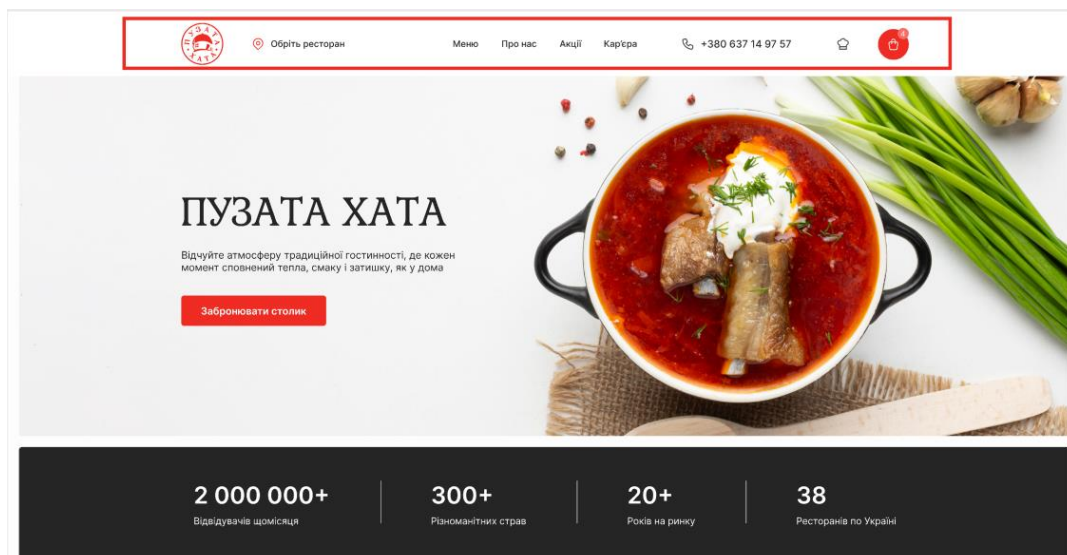


Рисунок 3.24 - Пункти навігації

Користувач може переглядати страви, які є фаворитами гостей (рис. 3.25), ця можливість зручна тим, що допомагає користувачам швидко знайти найпопулярніші та найбільш рекомендовані страви. Це економить час на вибір і

підвищує задоволення від користування системою, оскільки користувачі можуть бути впевнені у високій якості цих страв.

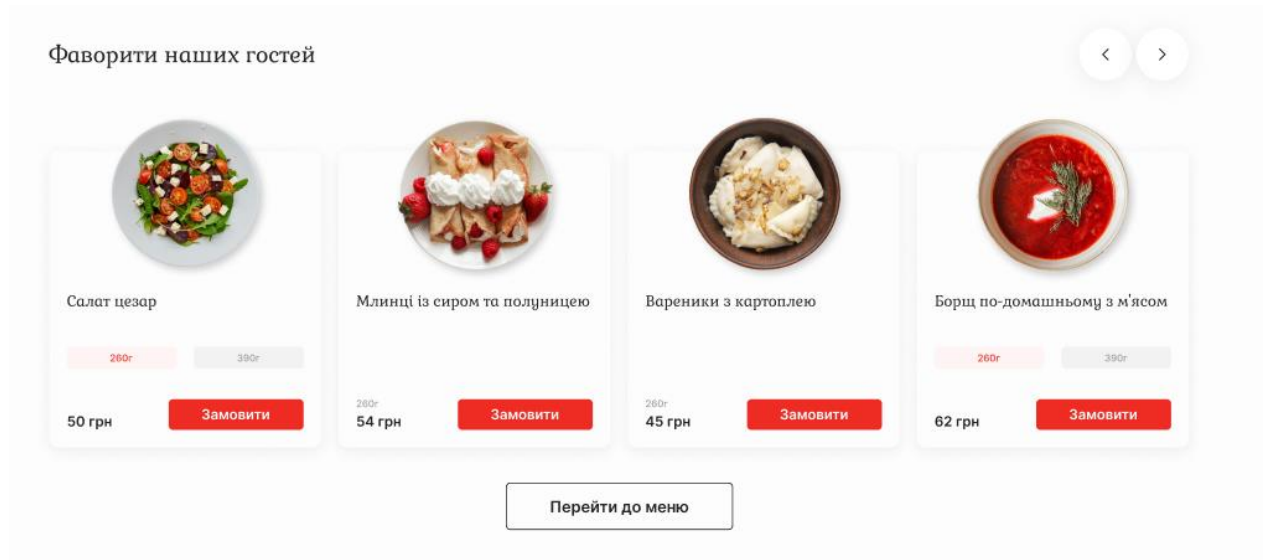


Рисунок 3.25 - Фаворити гостей

Для того, щоб обрати потрібний ресторан користувач має натиснути - *Обрати ресторан* (рис. 3.26).

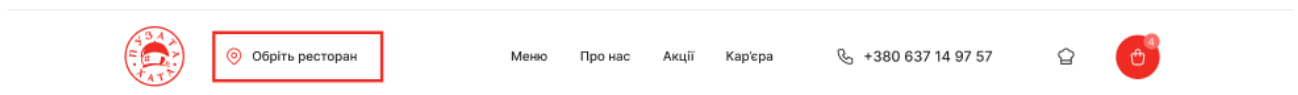


Рисунок 3.26 - Пункт меню – обрати ресторан

Після цього користувачу відкривається форма для вибору міста, та ресторану (рис.3.27). Потрібно обрати бажане місто та ресторан, та натиснути кнопку - *Обрати*.

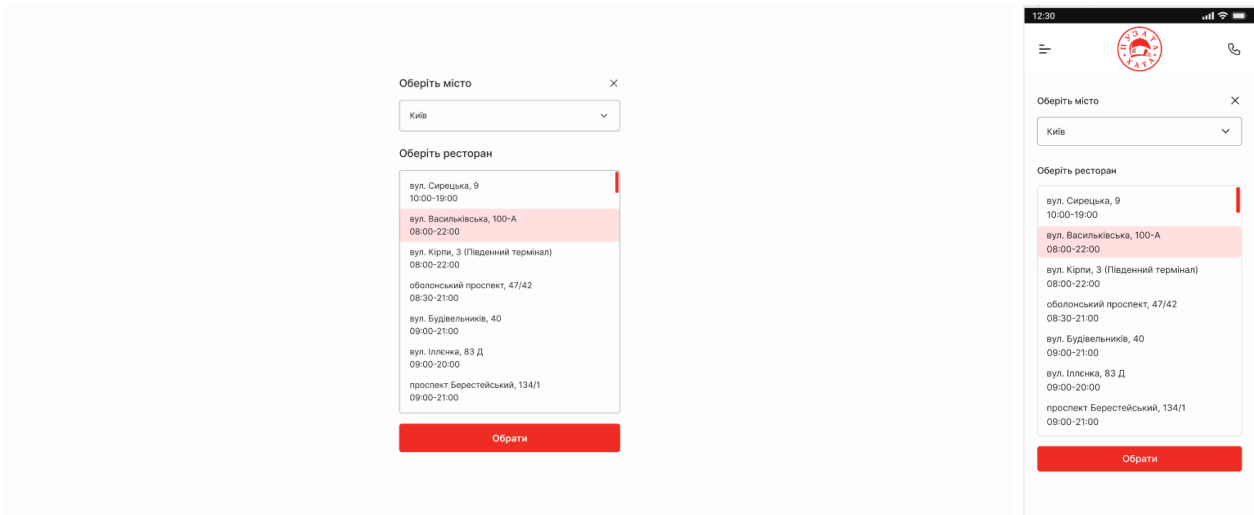


Рисунок 3.27 - Сторінка для обрання міста та ресторану

Якщо користувач не обрав ресторан, але при цьому хоче забронювати столик, то його перекидає на сторінку (рис.3.27) для обрання ресторану для подальшого бронювання.

Коли користувач натискає на пункт *Меню*, він буде перенаправлений на сторінку з категоріями та вибором страв (рис.3.28). На цій сторінці буде розташовано список доступних категорій страв, таких як перші страви, другі страви, напої тощо. Користувач може обирати категорію, щоб побачити перелік страв, що належать до цієї категорії.

Кожна категорія буде представлена у вигляді кнопки, на які користувач може клікнути, щоб переглянути страви. Після вибору категорії користувачу буде показано перелік страв, які відповідають цій категорії. Кожна страву буде представлена з назвою, ціною, грамівкою, зображенням. Для кожної страви буде доступна кнопка - *Замовити*, за допомогою якої користувач може додати обрані страви до свого кошика для подальшого оформлення замовлення.

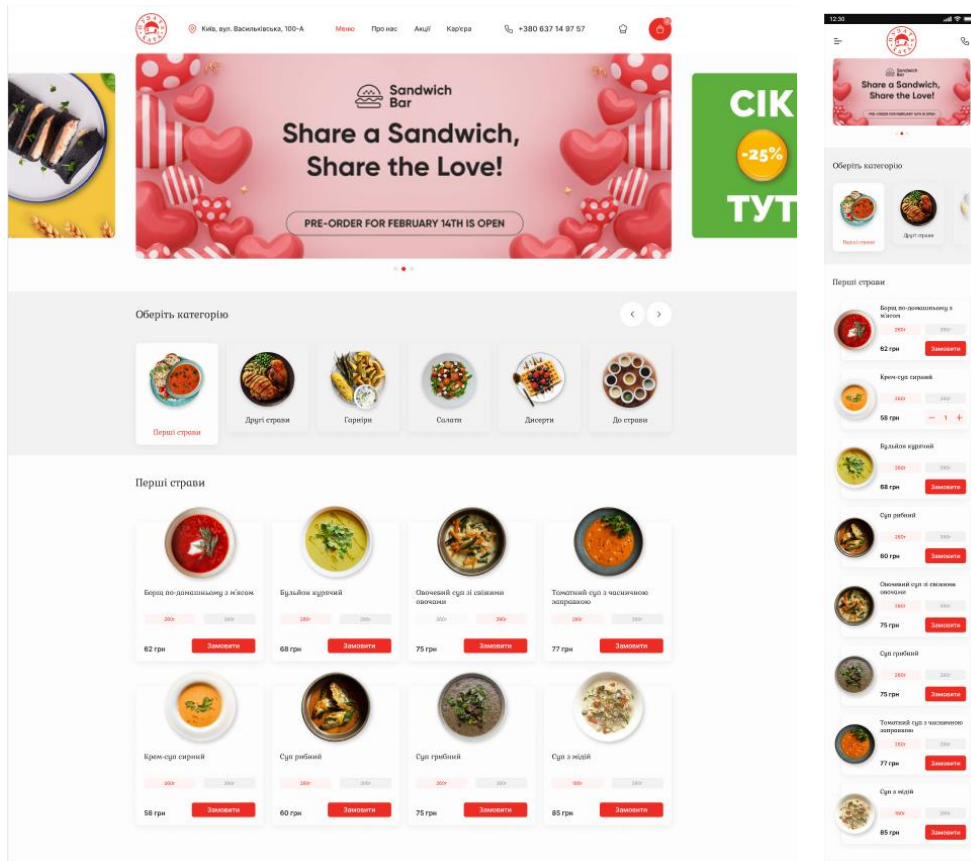


Рисунок 3.28 - Сторінка меню

Крім того, буде доступна додаткова опція, клікнувши на потрібну страву користувача перекидає на сторінку (рис.3.29), де детально описано страву (наприклад, інгредієнти, енергетична цінність).

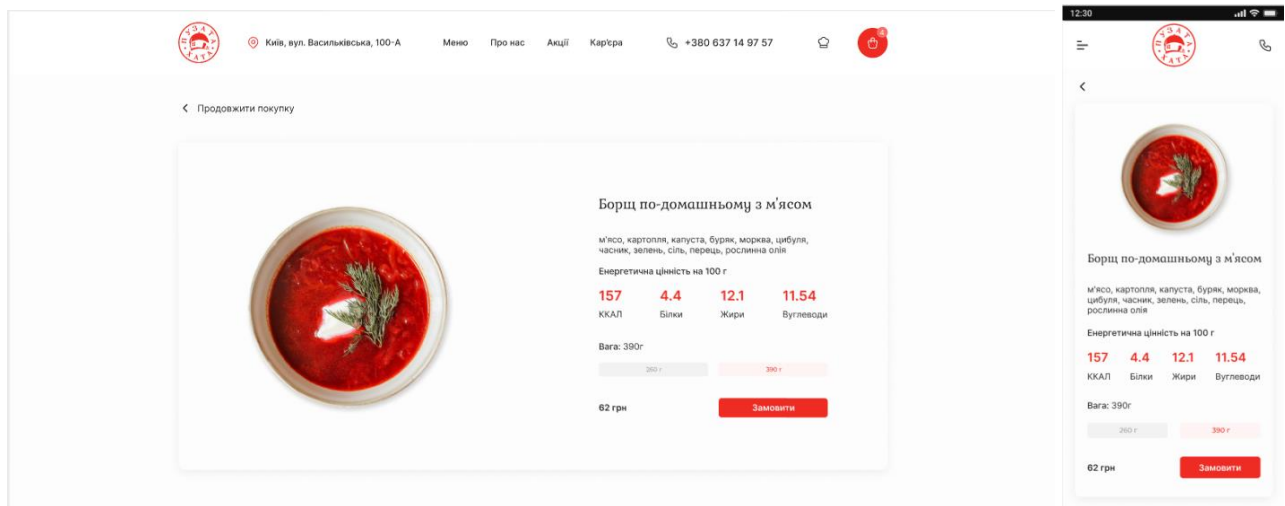


Рисунок 3.29 - Сторінка для перегляду складу страви

Після того, як користувач визначився, що буде замовляти, потрібно перейти у кошик (рис.3.30) де можна переглянути замовлення і після натиснути на кнопку - *Оформити замовлення*.

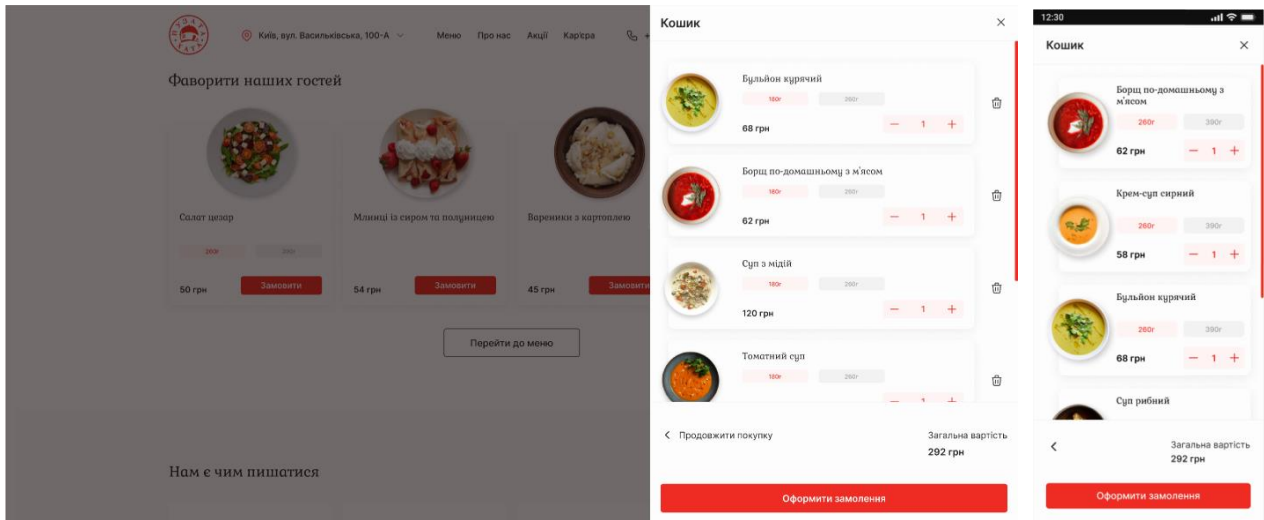


Рисунок 3.30 - Кошик заповнений

Якщо користувач не обирає ніяких страв, але натиснув на кошик він побачить, що кошик порожній (рис. 3.31).

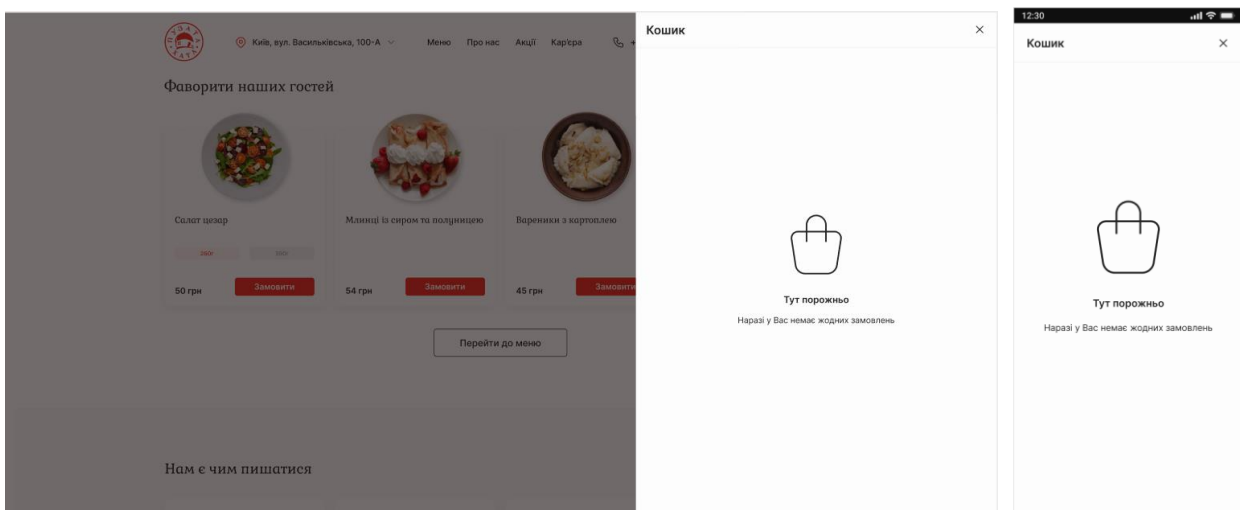


Рисунок 3.31 - Кошик порожній

Потім користувач потрапляє на сторінку, де зможе оформити замовлення вказавши свої дані, спосіб доставки та оплати (рис. 3.32).

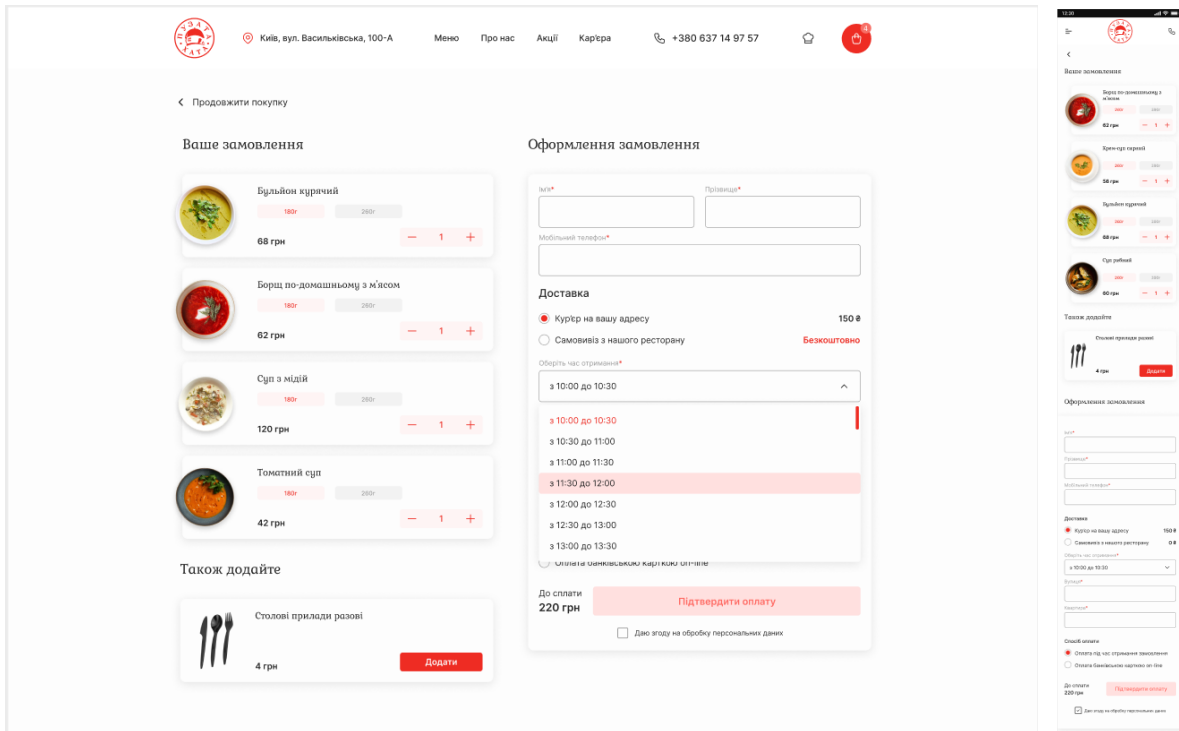


Рисунок 3.32 - Сторінка оформлення замовлення

Коли користувач закінчить оформлення він зможе натиснути кнопку - *Підтвердити оплату* і його перекине на сторінку подяки де він зможе просканувати Qr – код, щоб скопіювати реквізити для подальшої оплати. Після оплати клієнту слід очікувати дзвінок менеджера ресторану для узгодження замовлення (рис. 3.33).

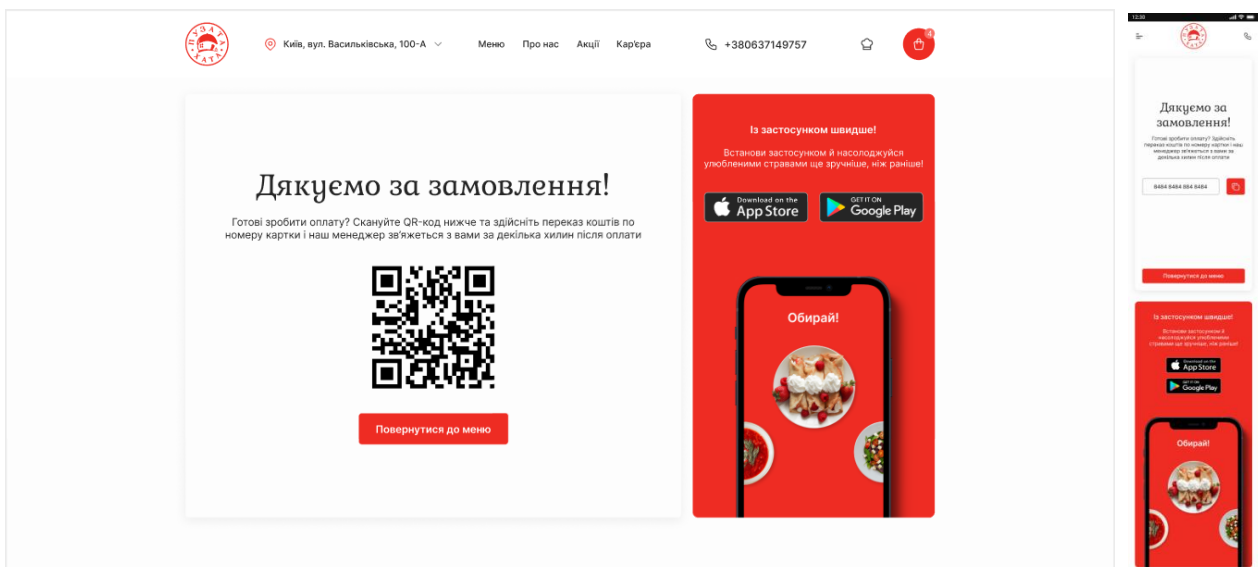


Рисунок 3.33 - Сторінка подяки за замовлення

Коли користувач натискає на пункт *Про нас*, він буде перенаправлений на сторінку де побачить інформацію про цінності ресторану, та його переваги. Також користувач нижче побачить відгуки інших користувачів, у нього буде можливість також залишити свій відгук (рис. 3.34).

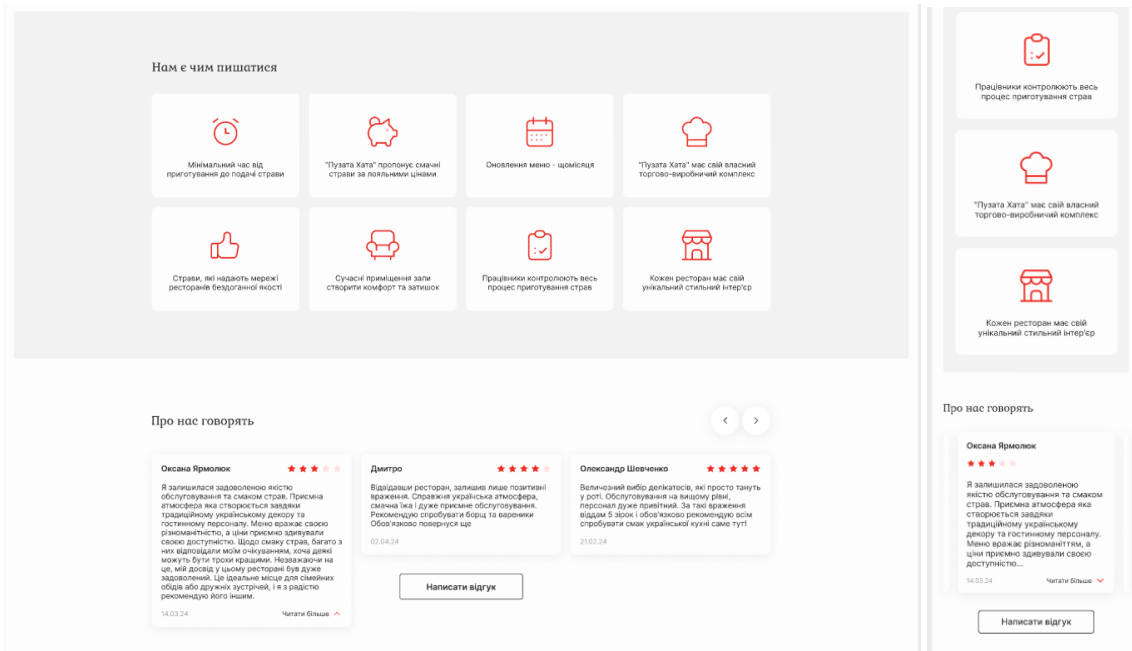


Рисунок 3.34 - Сторінка про ресторан, та відгуки

Для того щоб написати відгук потрібно натиснути на кнопку - *Написати відгук*. Відкривається форма (рис.3.35) для написання відгуку, а також можливість поставити оцінку, обраши певну кількість зірочок.

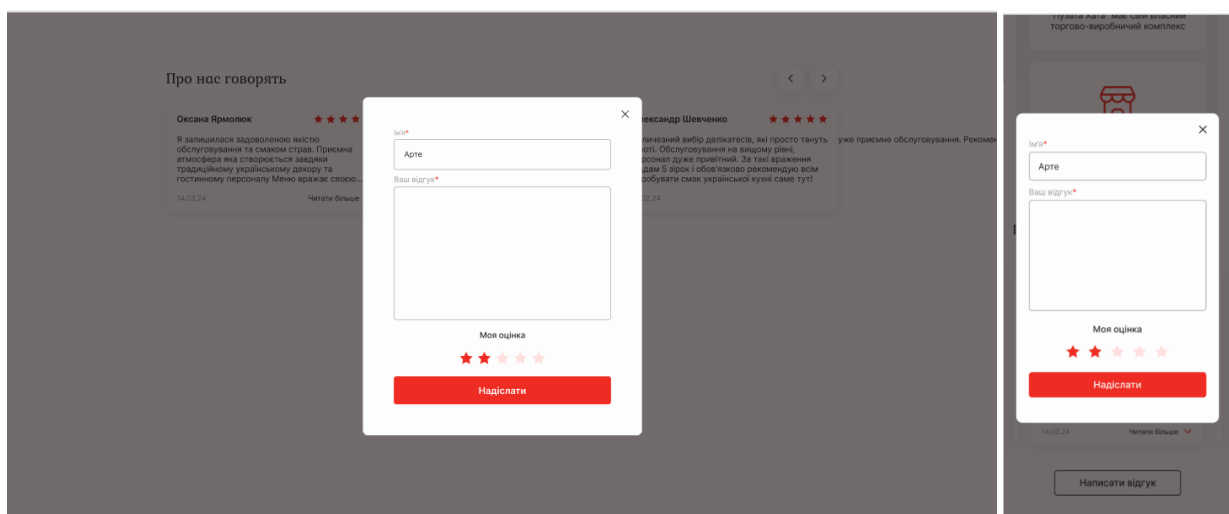


Рисунок 3.35 - Форма для написання відгуку

Після того, як користувач написав відгук, потрібно натиснути на кнопку - *Надіслати* (рис. 3.36). Після цього користувач побачить подяку за відгук.

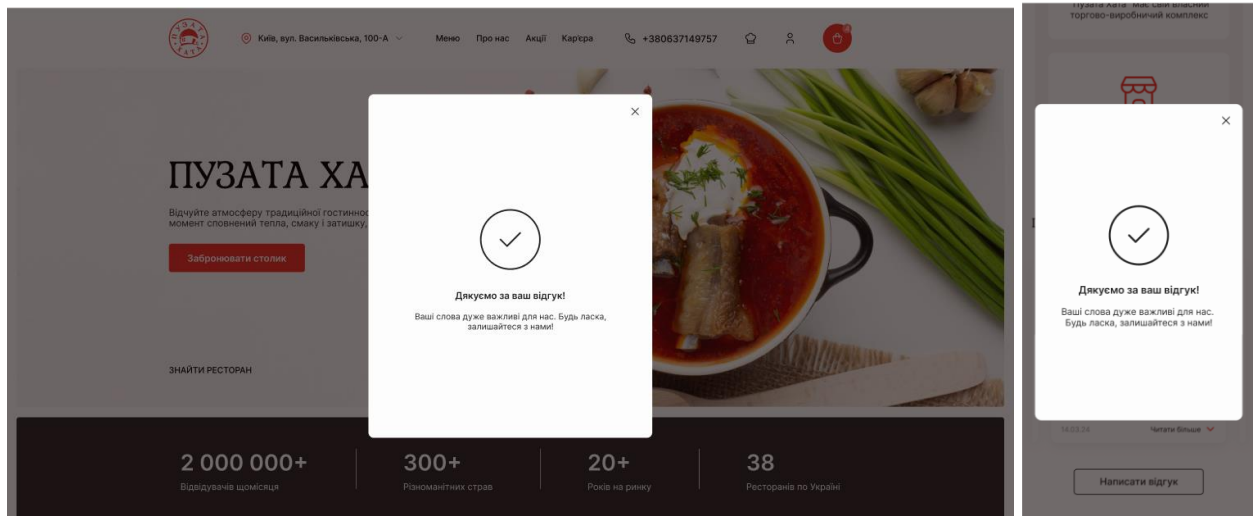


Рисунок 3.36 - Подяка за відгук

Коли користувач натискає на пункт *Акції*, він буде перенаправлений на сторінку де побачить страви які на даний момент мають знижку.

Якщо користувач натискає на пункт *Кар'єра*, він буде перенаправлений на сторінку де зможе заповнити спеціальну анкету, за допомогою якої людина яка бажає працювати у ресторані може потрапити у команду, якщо адміністрації закладу підіодитиме даний кандидат. Для заповнення анкети потрібно ввести свої дані, по можливості прикріпити резюме, обрати бажану посаду та місто і натиснути на кнопку відправити (рис. 3.37).

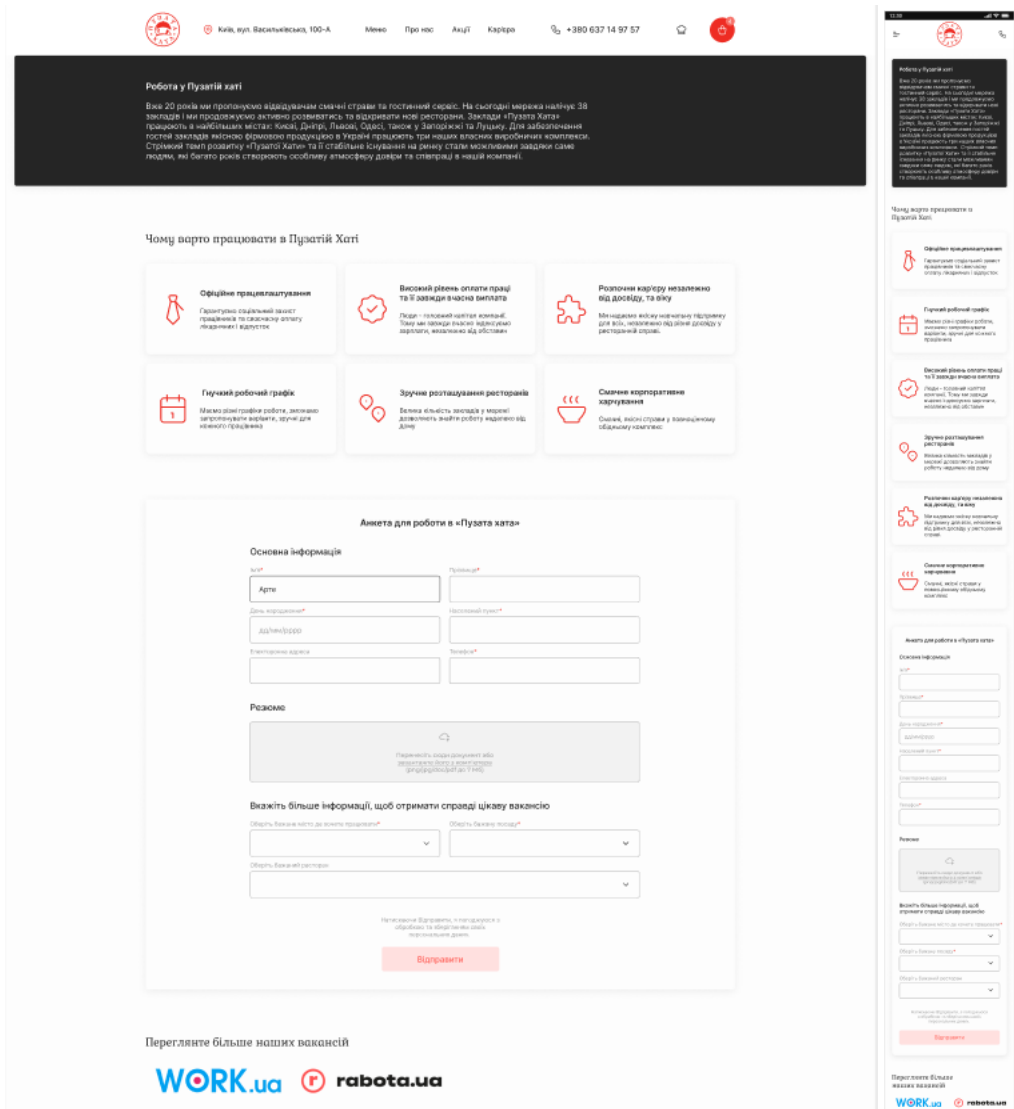


Рисунок 3.37 - Сторінка для заповнення анкети

Після цих дій з кандидатом зв'яжуться адміністрація для подальших співбесід.

Коли користувач натискає на кнопку (рис. 3.38) показану нижче, він буде перенаправлений на сторінку де побачить інформацію про ресторани по кожному з міст де знаходиться дана мережа ресторанів.

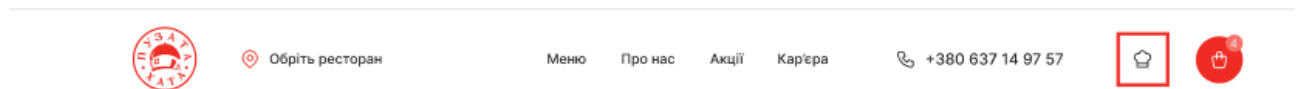


Рисунок 3.38 - Пункт меню мапа ресторанів

Користувач побачить інформацію про заклади (наприклад, години роботи, адресу, відчинено, чи не відчинено). Також користувач зможе

прокладати маршрут до обраного ресторану натиснувши на кнопку - *Прокласти маршрут* (рис. 3.39).

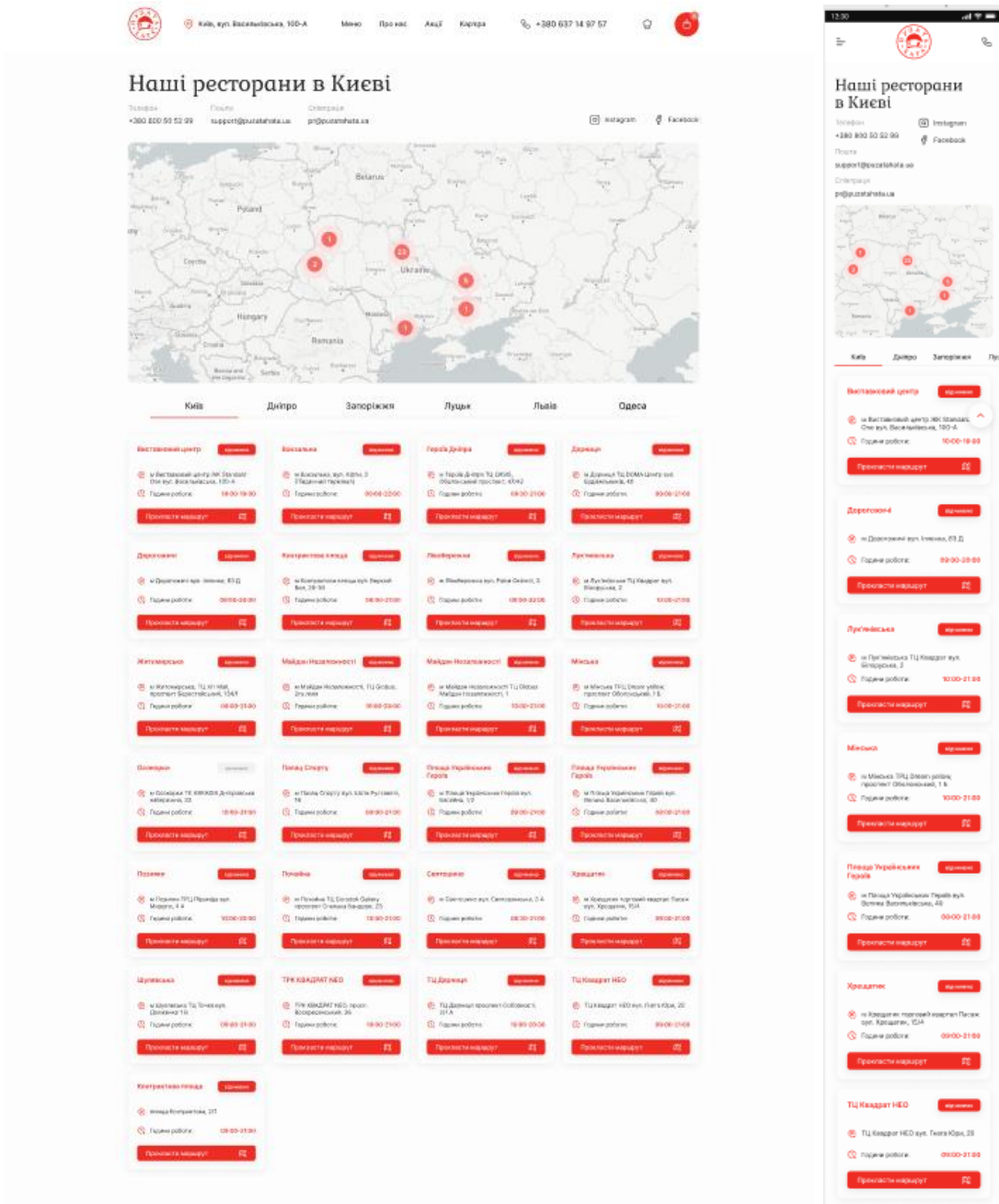


Рисунок 3.39 - Сторінка мапа ресторанів

3.4. Технічне та системне забезпечення розробки

3.4.1. Обґрунтування вибору технічних засобів

Вибір технічного забезпечення для сервера та користувача обґрунтовується наступним чином:

Для сервера:

- *Процесор:*

Для веб-орієнтованої системи ресторану, де передбачається велика кількість користувачів, які одночасно звертаються до системи, потрібен потужний процесор, який може швидко обробляти запити і відповідати на них. Наприклад, процесори з лінійки Intel Core (назва лінійки процесорів Intel) або AMD Ryzen Threadripper (назва лінійки процесорів компанії AMD) можуть бути чудовими виборами. Вони мають багатоядерну архітектуру, яка дозволяє обробляти багато запитів одночасно [15], [16];

- *Пам'ять:*

Пам'ять сервера також є критичною для продуктивності. Чим більше пам'яті, тим більше запитів можна обробляти одночасно без збоїв. Наприклад, для веб-орієнтованої системи ресторану, що включає у себе зображення страв, меню, систему бронювання тощо, 16-32 ГБ оперативної пам'яті можуть бути досить;

- *Сховище даних (диск):*

Для збереження великої кількості даних, таких як замовлення, інформація про клієнтів, меню тощо, потрібен надійний і швидкий диск. SSD диски зазвичай є кращим вибором для серверів, оскільки вони пропонують швидший доступ до даних порівняно з HDD дисками;

- *Мережеві можливості:*

Важливо також враховувати можливості мережі сервера. Наприклад, гігабітні мережеві інтерфейси дозволяють швидко обмінюватися даними між сервером і користувачами;

- *Безпека:*

Забезпечення безпеки є ще одним ключовим аспектом. Потрібно враховувати можливості захисту від DDoS-атак, встановлення файрволів, шифрування даних тощо;

- *Операційна система:*

Вибір операційної системи для користування системою має вагомим значення з точки зору його ефективності, надійності та безпеки. Windows може бути оптимальним вибором для сервера системи, оскільки вона забезпечує широку підтримку технічних засобів, програмного забезпечення, має велику компетенцію персоналу та спрощує інтеграцію з існуючою інфраструктурою Microsoft.

Обладнання для користувача:

- *Ноутбук/Комп'ютер:*

Для веб-орієнтованої системи ресторану, як правило, потрібен комп'ютер або ноутбук, який може забезпечити стабільну та швидку роботу, особливо при обробці великої кількості даних, графіки та інших мультимедійних елементів. Наприклад, для таких завдань оптимальним вибором будуть процесори Intel Core i5 або AMD Ryzen 5. Їхні можливості забезпечують ефективний рівень продуктивності для користувача веб-орієнтованої системи, здатну ефективно опрацьовувати великі обсяги даних та мультимедійний контент.

- *Браузер:*

Користувач має можливість обирати браузер, який найбільше відповідає його потребам та вимогам. Проте, вибираючи браузер, слід враховувати його здатність підтримувати сучасні веб-сайти та веб-додатки. Надійний браузер повинен бути безпечним, швидким та сумісним із різними платформами. З цього погляду, Google Chrome може бути одним із найбільш оптимальних варіантів, оскільки він відповідає вищезазначеним критеріям і є популярним

серед багатьох користувачів вебу. Однак користувачі також можуть розглянути інші варіанти залежно від їх власних уподобань та потреб.

- *З'єднання з мережею Інтернет:*

Швидке й стабільне підключення дозволить користувачеві без затримок ознайомитися з меню, зробити замовлення та вирішити питання про резервацію столика. Крім того, воно забезпечить безперервний доступ до актуальної інформації про акції, знижки, що робить відвідування ще більш приємним та зручним.

- *Мобільний пристрій:*

Вибір технічного забезпечення для користувача всієї системи ґрунтується на його адаптивності, що дозволяє зручно переглядати контент з будь-якого пристрою. Використання адаптивної системи та кросс-браузерної сумісності забезпечує максимальний комфорт користувача незалежно від пристрою, що використовується.

- *Операційна система:*

Вибір операційної системи для користувача може бути важливим фактором, який визначає його зручність, продуктивність та задоволення від користування комп'ютером. У цьому контексті вибір наприклад між Windows 10 та MacOS може бути обґрунтований різними факторами, які включають уподобання, потреби та особливості користувача.

3.4.2. Розрахунок та визначення топології комп'ютерної мережі

Для того щоб користувачі могли безперешкодно користуватися веб-системою, вона повинен бути розміщений на надійній інфраструктурі, яка забезпечує постійну доступність, швидкість завантаження та безпеку даних. Сучасний підхід до розміщення веб-систем передбачає використання хмарних сервісів, які пропонують гнучкі та масштабовані рішення для різних потреб. Хмарний хостинг дозволяє власникам веб-ресурсів швидко адаптуватися до змін у навантаженні, забезпечуючи безперебійну роботу системи незалежно від кількості відвідувачів. Такий підхід гарантує високий рівень продуктивності та

безпеки, що є критично важливим для успішного функціонування будь-якого інтернет-ресурсу.

Розміщення веб-систем в хмарному сервісі - це процес використання обчислювальних ресурсів, що надаються через Інтернет провайдерами хмарних послуг, для розміщення та обслуговування веб-систем. Хмарний хостинг дозволяє користувачам орендувати віртуальні сервери або використовувати платформи як послугу (PaaS) для розгортання своїх веб-додатків, забезпечуючи високу доступність, масштабованість та гнучкість.

Переваги хмарного хостингу:

- *масштабованість* - легке збільшення або зменшення ресурсів відповідно до потреб;
- *висока доступність* - більшість хмарних провайдерів гарантують високу доступність завдяки розподілу даних по декількох серверах;
- *гнучкість* - можливість вибору та зміни конфігурацій без переривання роботи;
- *безпека* - хмарні провайдери часто пропонують комплексні рішення з безпеки, включаючи резервне копіювання, захист від DDoS-атак тощо;
- *економічність* - оплата за використані ресурси, що дозволяє зменшити витрати на апаратне забезпечення та його обслуговування.

В Україні є кілька провайдерів, що пропонують хмарний хостинг:

1. *GigaCloud* - один з провідних хмарних провайдерів України, що пропонує широкий спектр послуг, включаючи IaaS (інфраструктура як послуга), *PaaS та SaaS* (програмне забезпечення як послуга) [17].
2. *De Novo* - український національний провайдер хмарних сервісів IaaS, PaaS та ЦОД, великий оператор хмарних сервісів, що надає послуги для корпоративних клієнтів, зокрема державних установ та великих компаній [18].
3. *UA-Cloud* - український національний сервіс хмарних послуг, з багаторічним досвідом роботи у сфері хостингу. Є платформою для

розміщення веб-сайтів та додатків, яка пропонує послуги IaaS, PaaS та SaaS з високим рівнем безпеки та доступності [19].

4. *Cityhost* - український хостинг-провайдер, що надає послуги хмарного хостингу, VPS та оренди серверів, забезпечуючи гнучкі тарифи та надійну технічну підтримку [20].
5. *MiroHost* - компанія з більш ніж 20-річним досвідом, що надає хмарні рішення для різних потреб бізнесу, включаючи віртуальні сервери та хостинг веб-сайтів [21].
6. *HostPro* - пропонує хмарний хостинг, віртуальні сервери та різноманітні послуги для розробників і підприємств з високим рівнем технічної підтримки [22].

3.4.3. Обґрунтування вибору ОС та протоколу обміну даними

Для використання даної системи у мережі ресторанів обрано операційної системи Windows, оскільки багато програм та сервісів, які використовуються в галузі ресторанного бізнесу, оптимізовані саме під цю ОС.

Щодо сервера, було обрано Apache HTTP. По-перше, Apache є одним з найпопулярніших та надійних веб-серверів у світі. Він має велику спільноту користувачів та розробників, що забезпечує активний розвиток, підтримку та безпеку. По-друге, Apache HTTP Server безкоштовний та з відкритим вихідним кодом, що робить його економічно вигідним варіантом для бізнесу. Також, Apache має гнучку конфігурацію та широкі можливості розширення, що дозволяє легко налаштувати сервер під конкретні потреби ресторанної системи.

Щодо засобу передачі даних, що використовується у веб-орієнтованій системі підтримки закладу громадського харчування, рекомендується використовувати протокол HTTP (Hypertext Transfer Protocol). HTTP обрано через його широке застосування, надійність та сумісність з веб-технологіями, що робить його ідеальним вибором для взаємодії між веб-сервером та клієнтськими пристроями, які отримують доступ до системи.

3.4.4. Адміністрування системою

На сторінці входу адміністратору потрібно авторизуватися. Після успішної автентифікації адміністратор буде перенаправлений на адміністративну панель, де він матиме доступ до всіх функцій для управління системою (рис.3.40).

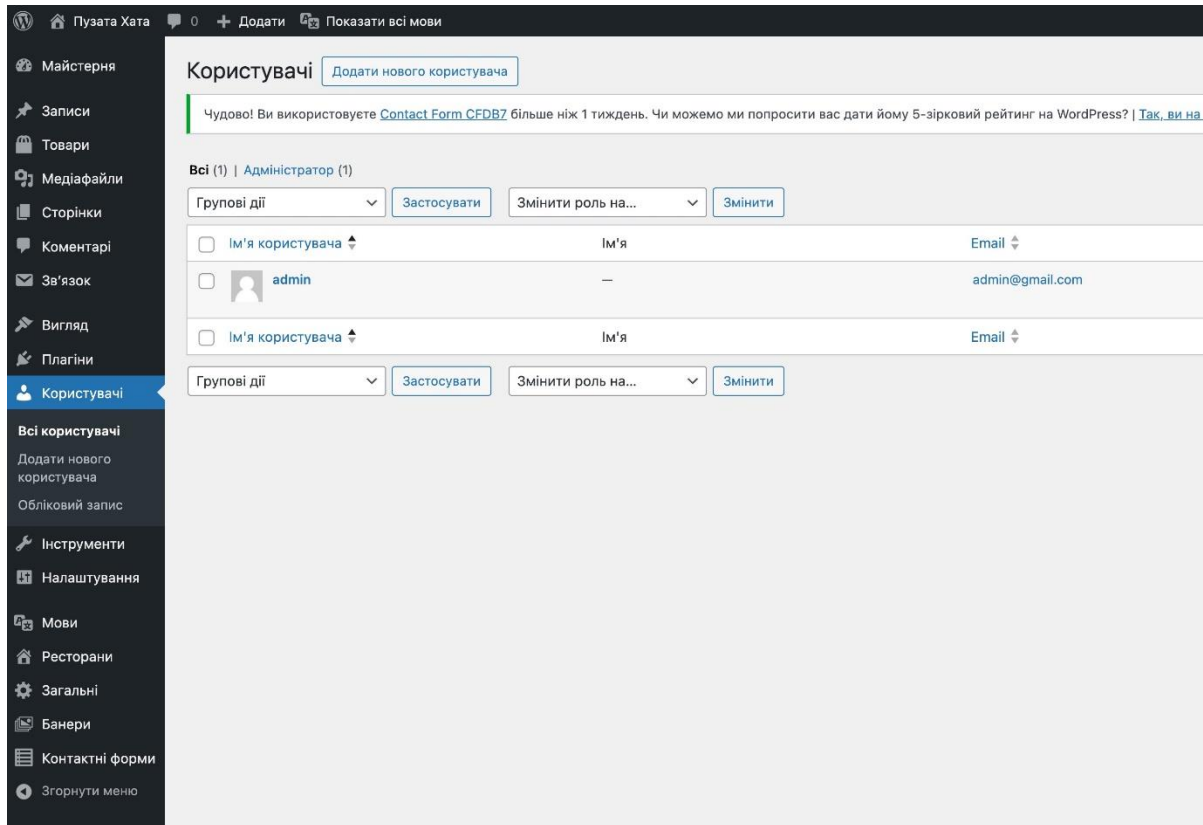


Рисунок 3.40 - Панель адміністратора

Адміністратор має такі можливості:

- *управління замовленнями* - адміністратор може переглядати, змінювати та видаляти замовлення, які надійшли через систему. Це включає в себе відстеження статусу замовлення;
- *управління відгуками користувачів* - адміністратор може переглядати відгуки користувачів проресторан. Він також може видаляти неприйнятні або спамні відгуки;
- *створення та управління акційними банерами* - адміністратор може додавати та редагувати акційні банери у системі, які привертають увагу користувачів та рекламують акції та знижки;

- *бронювання столиків* - адміністратор може переглядати та керувати всіма бронюваннями столиків;
- *створення нових записів страв* - адміністратор може додавати інформацію про нові страви та напої до системи. Це може включати назву страви, опис, складники, категорію (наприклад, закуски, основні страви, десерти) та інші важливі атрибути.

Вхід до на сторінку адміністратора здійснюється шляхом введення посилання <http://localhost/admin> .

Адміністратор може надавати ролі іншим працівникам (рис.3.41).

Додати нового користувача

Чудово! Ви використовуєте [Contact Form CFDB7](#) більше ніж 1 тиждень. Чи можемо ми попросити вас дати йому 5-зірк

Створити обліковий запис нового користувача і додати його до цього сайту.

Ім'я користувача (обов'язково)

Email (обов'язково)

Ім'я

Прізвище

Сайт

Мова

Пароль

Сильний

Надіслати сповіщення Send the new user an email about their account

Роль

- Учасник
- Помічник
- Автор
- Редактор
- Адміністратор

Рисунок 3.41 - Надання ролей

3.4.5. Заходи захисту від несанкціонованого доступу до системи

Для захисту системи від несанкціонованого доступу впроваджено механізм авторизації з одним рівнем доступу, це адміністратор.

Користувачам без авторизації не дозволено скористатися жодною з функцій. Додавання нових користувачів може проводити лише адміністратор. Також тільки адміністратор має повноваження розподіляти права доступу (рис. 3.42).

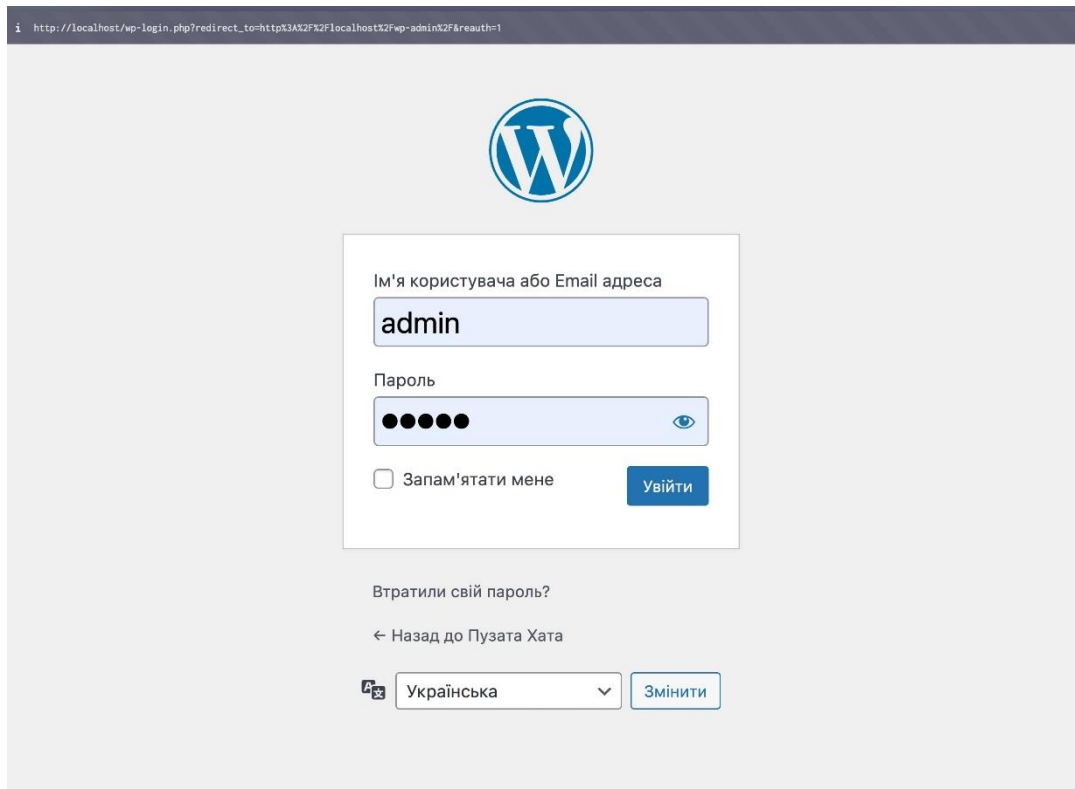


Рисунок 3.42 - Форма авторизації

Якщо користувач ввів неправильний пароль або ім'я користувача, він отримає відповідне сповіщення (рис.3.43).

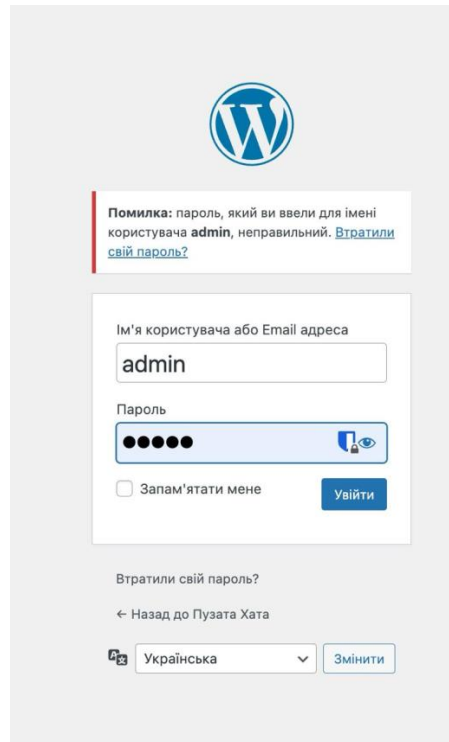


Рисунок 3.43 - Сповіщення про неправильно введені дані

Хешування (рис.3.44) використовується для передачі та збереження паролю, що робить його неможливим для розшифрування та зламу.

ID	user_login	user_pass	user_nicename	user
1	admin	\$P\$BY0RI2WmjdpjyknM33DGHJjzqTaCX.	admin	adm

Рисунок 3.44 - Хешований пароль користувача

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці є фундаментальним аспектом будь-якої діяльності, що має на меті забезпечення безпеки та здоров'я працівників на робочому місці. Вона не лише визначається законодавством, але і є моральною та етичною відповідальністю керівництва підприємства.

Важливість охорони праці полягає в тому, що вона дозволяє забезпечити безпеку, здоров'я та добробут працівників. Ретельно розроблені та дотримувані норми охорони праці допомагають уникнути травматичних ситуацій, а також сприяють підвищенню продуктивності та ефективності роботи.

Мають бути враховані наступні ключові аспекти охорони праці:

- *навчання та інструктаж* - перед початком роботи з системою працівники повинні отримати належне навчання з її використання та безпеки. Працівники мають бути забезпечені інструкціями з використання системи та процедур безпеки в разі виникнення проблем;
- *доступ до інформації* - повинна бути забезпечена доступність інформації про безпеку та процедури дій у випадку аварійного вимкнення системи або інших проблем;
- *ергономіка робочого місця* - робоче місце працівників, які взаємодіють з системою, має бути спроектовано з урахуванням правильної ергономіки. Це включає належне розміщення моніторів, клавіатур, мишей та інших пристроїв;
- *правила безпеки щодо роботи з електронними пристроями та дротами* - працівники мають бути навчені правилам безпеки при використанні електронних пристроїв, а також техніці безпечної роботи з електричними дротами і кабелями, щоб уникнути потенційних травм або аварій;

- *захист даних* - персонал повинен дотримуватися найвищих стандартів захисту даних, щоб запобігти можливим порушенням конфіденційності даних клієнтів або фінансових даних ресторану;
- *регулярні перевірки системи* - повинні регулярно проводитися перевірки технічного стану системи, щоб впевнитися, що вона працює належним чином і не створює загроз для даних;
- *обмеження доступу* - важливо, щоб доступ до системи мали лише необхідним працівники з відповідними рівнями дозволу для кожного користувача відповідно до їх ролі в ресторані;
- *регулярні перерви* - протягом робочого дня повинні бути запроваджені регулярні перерви для відпочинку очей, рук і тіла, що дозволить працівникам розслабитися і зменшити напругу.

Шляхом впровадження цих заходів можна уникнути травм, підвищити продуктивність працівників, а також зберегти репутацію компанії та підвищити задоволеність персоналу. Таким чином, охорона праці є не лише правовим обов'язком, але й ключовим фактором успішної та ефективної діяльності підприємства.

Стосовно аспектів інтерфейсу системи охорони праці, важливо звернути увагу на наступне:

- *простота та зрозумілість інтерфейсу* - інтерфейс системи повинен бути простим та зрозумілим для всіх користувачів. Це дозволить працівникам швидко орієнтуватися у системі та виконувати необхідні дії без зайвих зусиль;
- *інтуїтивність навігації* - навігація в системі повинна бути логічною та інтуїтивно зрозумілою. Важливо, щоб користувачам було легко користуватися системою. Це допоможе їм швидше знаходити потрібні функції і уникати помилок;
- *відповідність потребам користувачів* - важливо, щоб інтерфейс системи відповідав потребам користувачів і враховував їхні вимоги та

вподобання. Наприклад, якщо система використовується працівниками різних вікових категорій або з різним рівнем технічної підготовки, інтерфейс повинен бути придатний для використання для всіх цих користувачів, забезпечуючи зручність та ефективність роботи;

- *підтримка різних пристроїв* - у зв'язку з тим, що користувачі можуть мати доступ до системи через різні пристрої (наприклад, комп'ютери, планшети, мобільні телефони), інтерфейс повинен бути адаптований до різних типів пристроїв та їхніх розмірів екранів.

Ефективна організація та належне обладнання робочих місць працівників відіграють ключову роль у забезпеченні успішної та продуктивної роботи всього закладу. Перш за все, це сприяє оптимізації робочих процесів, що дозволяє швидше та ефективніше обслуговувати клієнтів. Добре організовані місця праці сприяють збереженню часу та зменшенню стресу для персоналу, що в свою чергу позитивно позначається на загальній якості обслуговування.

Розглянемо докладніше, які аспекти охорони праці слід враховувати при використанні веб-орієнтованої систем у ресторанному бізнесі, зосереджуючись на організації та обладнанні робочих місць:

- *зручність робочих місць* - забезпечення оптимальної організації робочого простору для зменшення фізичного напруження та уникнення травм. Регулювання стільців, столів та комп'ютерного обладнання належним чином для підтримки здорової постави;
- *освітлення* - достатнє природне та штучне освітлення в робочих приміщеннях, щоб уникнути напруги очей та покращити продуктивність. Забезпечення можливості регулювання яскравості освітлення відповідно до потреб кожного працівника;
- *вентиляція та якість повітря* - забезпечення належної вентиляції для запобігання перегріванню та перенасиченню повітря в приміщенні. Регулярна перевірка та обслуговування систем вентиляції для забезпечення їх ефективності;

- *комп'ютери та монітори* - працівники мають бути забезпеченими сучасними комп'ютерами з потужними процесорами та достатньою кількістю оперативної пам'яті для ефективної роботи з веб-орієнтованою системою. Монітори повинні мати високу роздільну здатність та можливість регулювання яскравості для запобігання напругам очей;
- *клавіатури та миші* - працівники мають бути забезпеченими зручними для використання та якісного вигляду клавіатурами та мишами, щоб зменшити ризик напруг та травм, пов'язаних з довготривалим використанням;
- *інтернет-підключення* - має бути стабільне та швидке інтернет-підключення для безперебійної роботи з веб-орієнтованою системою та забезпечення зручного замовлення та обробки замовлень;

ВИСНОВОК

У процесі розробки було створено альтернативну веб-орієнтовану систему для мережі ресторанів “Пузата хата”, були успішно досягнуті основні цілі проекту. Система відповідає всім вимогам, визначеним на початковому етапі, включаючи функціональність, продуктивність та безпеку. В кваліфікаційній роботі вдалося створити інтуїтивно зрозумілий та зручний для користувачів інтерфейс, що забезпечує ефективне виконання послуг.

Завдяки ретельному плануванню та впровадженню новітніх технологій, система демонструє високу ефективність у виконанні основних функцій, значно полегшуючи робочі процеси користувачів. Проведене тестування підтвердило відповідність усіх ключових вимог щодо функціональності та зручності використання, забезпечуючи позитивний досвід для кінцевих користувачів.

Отже, у бакалаврській роботі було створено веб-орієнтовану систему, призначену для підтримки діяльності закладу громадського харчування.

Крім того, було розширено функціональні можливості наявної веб-системи, включаючи онлайн-замовлення та резервування столиків, що сприяє зручності та швидкості обслуговування клієнтів. Також було вдосконалено систему зворотного зв'язку, щоб отримувати і враховувати відгуки від клієнтів для подальшого вдосконалення сервісу.

Таким чином, у цій бакалаврській роботі не лише було створено веб-систему для громадського харчування, а й було вдосконалено існуючі функції, сприяючи подальшому розвитку та успіху мережі ресторанів “Пузата хата”.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Про компанію. Пузата хата. *Портал топ-менеджерів оптової та роздрібною торгівлі.* URL: https://trademaster.ua/company/puzata-hata/page/o_kompanii/0 (дата звернення: 17.05.2024).
2. Розділ “Про нас”. URL: <https://www.puzatahata.ua/kyiv/kyiv-puzata-khata-yrx6v> (date of access: 17.05.2024).
3. Salateira Київ. Меню, замовлення і доставка. Salateira. *Salateira.* URL: <https://salateiramenue.com.ua/kyiv/salateira> (дата звернення: 17.05.2024).
4. МакДональдз Україна: Бургери, Хрустка Фрі та МакФлурі. *McDonald's Deutschland.* URL: <https://www.mcdonalds.com/ua/uk-ua.html> (дата звернення: 17.05.2024).
5. KFC Ukraine | Home. *KFC Ukraine | Home.* URL: <https://www.kfc-ukraine.com/> (date of access: 17.05.2024).
6. Figma: The Collaborative Interface Design Tool. *Figma.* URL: <https://www.figma.com/> (date of access: 17.05.2024).
7. Інструмент створення блогів, платформа для публікацій і CMS - WordPress.org Україна. *WordPress.org Україна.* URL: <https://uk.wordpress.org/> (дата звернення: 17.05.2024).
8. phpMyAdmin. *phpMyAdmin.* URL: <https://www.phpmyadmin.net/> (date of access: 17.05.2024).
9. Учасники проєктів Вікімедіа. HTML – Вікіпедія. *Вікіпедія.* URL: <https://uk.wikipedia.org/wiki/HTML> (дата звернення: 18.05.2024).
10. Учасники проєктів Вікімедіа. CSS – Вікіпедія. *Вікіпедія.* URL: <https://uk.wikipedia.org/wiki/CSS> (дата звернення: 18.05.2024).
11. Учасники проєктів Вікімедіа. JavaScript – Вікіпедія. *Вікіпедія.* URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення: 18.05.2024).
12. Home - Carbon Fields. *Carbon Fields.* URL: <https://carbonfields.net/> (date of access: 18.05.2024).

13. Contact Form 7. *WordPress.org*. URL: <https://wordpress.org/plugins/contact-form-7/> (date of access: 18.05.2024).
14. Swiper - The Most Modern Mobile Touch Slider. *Swiper*. URL: <https://swiperjs.com/> (date of access: 18.05.2024).
15. Учасники проєктів Вікімедіа. Intel Core – Вікіпедія. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Intel_Core#:~:text=Intel%20Core%20\(укр.,покоління%20ще%20з%201993%20року](https://uk.wikipedia.org/wiki/Intel_Core#:~:text=Intel%20Core%20(укр.,покоління%20ще%20з%201993%20року) (дата звернення: 19.05.2024).
16. Учасники проєктів Вікімедіа. Ryzen – Вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Ryzen> (дата звернення: 19.05.2024).
17. GigaCloud: Хмарні Технології та Хмарний Сервіс для Бізнесу. GigaCloud: Хмарні Технології та Хмарний Сервіс для Бізнесу. URL: https://gigacloud.ua/?utm_source=googleads&utm_medium=cpc&utm_campaign=brand_ua&utm_content=140197731052&utm_term=gigacloud&utm_device=c&utm_gad_source=1&utm_gclid=Cj0KCQjwxqayBhDFARIsAANWRnSk1BBAyKE_-TE5p9J5MmKkJvq7oCdK2_nrRkQ2KgvogfKZFRDFiRkaAsrjEALw_wcB (дата звернення: 19.05.2024).
18. VMware Sovereign Cloud IaaS провайдер. Провайдер хмарних сервісів та технологій IaaS, PaaS, ЦОД в Києві та Україні | De Novo. URL: https://denovo.ua/vmware-sovereign-cloud?utm_gad_source=1&utm_gclid=Cj0KCQjwxqayBhDFARIsAANWRnQziibrOFzxEOy7-ga82R6-EPs0Azsl2TnJMAVz_mK9nsfWghTN0RcaAmvBEALw_wcB (дата звернення: 19.05.2024).
19. Послуги хостингу і оренди серверів – UA CLOUD | Київ. UA Cloud. URL: <https://uacloud.com.ua/> (дата звернення: 19.05.2024).
20. CityHost.ua. Хостинг в Україні від Cityhost – найкращий хостинг сайту. URL: https://cityhost.ua/uk/?partner=brand&utm_gad_source=1&utm_gclid=Cj0KCQjwxqayBhDFARIsAANWRnS1Yc5w8x2j09zVCY7XDCjdOSSrZdAr2jBrR4rYnL2WmMpm9Cg6tq4aAlZfEALw_wcB (date of access: 19.05.2024).

21. MiroHost. MiroHost - найбільший хостинг-провайдер в Україні. URL: <https://mirohost.net/ua> (date of access: 19.05.2024).
22. Хостинг сайтів [2024] від Hostpro. Купити hosting для сайту в Україні. Hostpro. URL: https://hostpro.ua/ua/?utm_source=google&utm_medium=src&utm_id=13476600457&utm_campaign=Search_Брендовые&utm_source=1&utm_gclid=Cj0KCQjwxqayBhDFARIsAANWRnTAua0E4swwcoA07Q3jJGZwh99aCoJVLd5qqHGvJz1Snw0umcqRkjwaAuJqEALw_wcB (дата звернення: 19.05.2024).
23. Знати CSS – недостатньо: що таке SCSS та як з ним працювати. Розбір синтаксису. Highload.today - медіа для розробників. URL: <https://highload.today/uk/scss/> (дата звернення: 19.05.2024).
24. JetBrains. PhpStorm: The PHP IDE by JetBrains. *JetBrains*. URL: <https://www.jetbrains.com/phpstorm/> (date of access: 17.05.2024).
25. Process Modeling Software - Bizagi Modeler | Bizagi. *Low-Code Platform for Process Automation and Enterprise Apps* / Bizagi. URL: <https://www.bizagi.com/en/platform/modeler> (date of access: 17.05.2024).
26. Управління ІТ проектами [Електронний ресурс]: лабораторний практикум для студентів напряму підготовки 6.050101 "Комп'ютерні науки" денної та заочної форм навч. / уклад. О. А. Хлобистова, М. В. Гладка. - К. : НУХТ, 2013. – 108 с.. URL: <http://library.nuft.edu.ua/ebook/file/51.07A.pdf>.
27. Управління ІТ проектами [Електронний ресурс]: методичні рекомендації до виконання курсової роботи для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навч. / уклад. М. В. Гладка, О. А. Хлобистова. – К. : НУХТ, 2014.– 91 с.. URL: <http://library.nuft.edu.ua/ebook/file/51.13.pdf>.
28. Управління ІТ проектами [Електронний ресурс]: методичні рекомендації до виконання курсової роботи для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навч. / уклад. М. В.

Гладка, О. А. Хлобистова. – К. : НУХТ, 2014.– 91 с.. URL:
<http://library.nuft.edu.ua/ebook/file/51.13.pdf>.

29.Проектування інформаційних систем [Електронний ресурс]: конспект лекцій для студентів освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» денної та заочної форм навчання. Уклад.: О. М. М'якшило, О. В. Харкянен: НУХТ, 2018. – 47 с.

30.Проектування інформаційних систем. [Електронний ресурс]: лабораторний практикум для студ. освітнього ступеню "бакалавр" спец. 122 “Комп'ютерні науки ” денної і заочної форм навчання. Частина 1 / Уклад.: О.М. М'якшило, О.В. Харкянен – К.: НУХТ, 2017 – 33 с.

ДОДАТКИ

Додаток А. Інтерфейс веб-орієнтованої системи

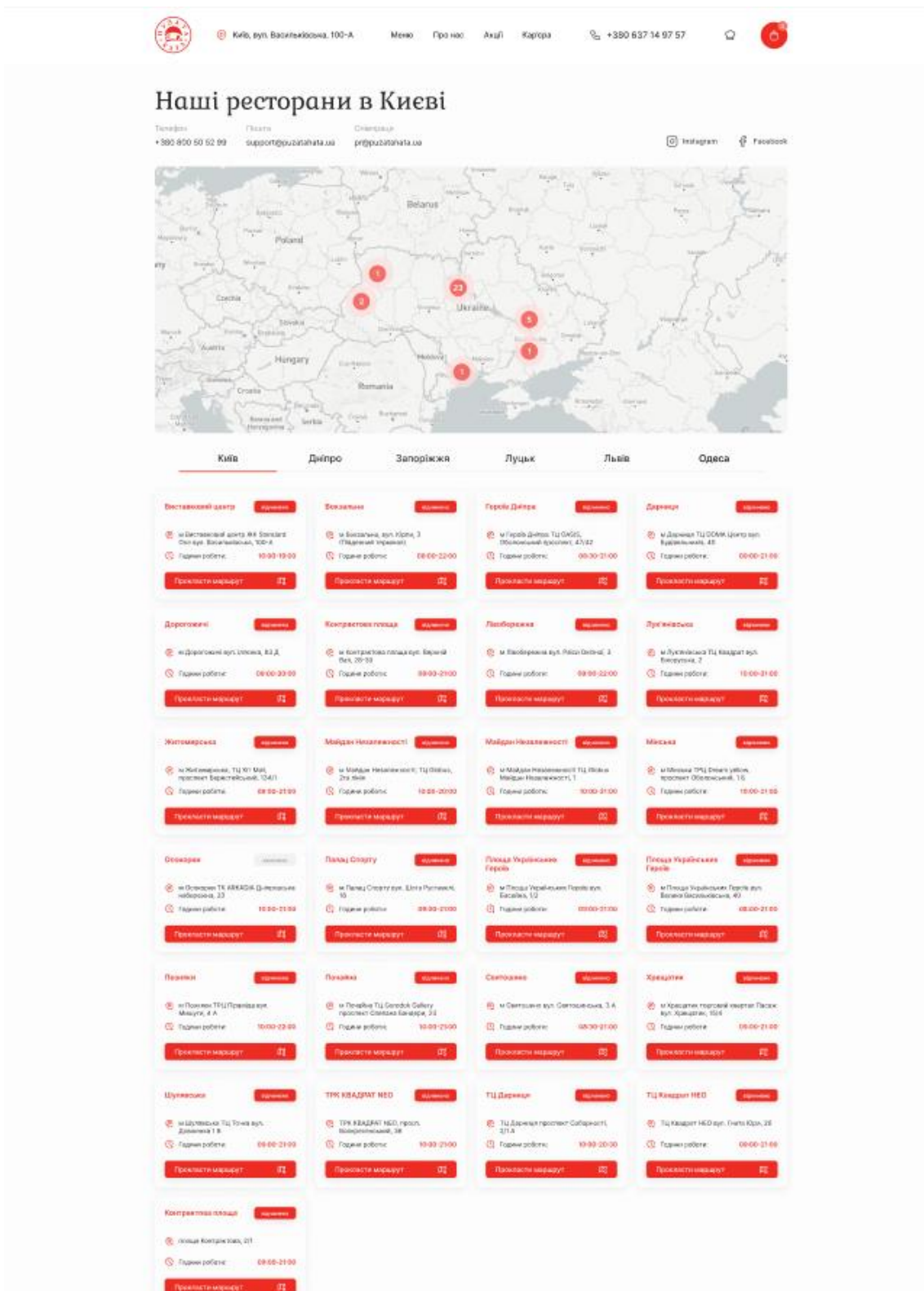



Рисунок А.1 - Сторінка ресторанів


Київ, вул. Васильківська, 100-А
Меню
Промас
Асїї
Кар'єра
+380 637 14 97 57

Робота у Пузатій хаті

Вже 20 років ми пропонуємо відвідувачам смачні страви та гостинний сервіс. На сьогодні мережа налічує 38 закладів і ми продовжуємо активно розвиватись та відкривати нові ресторани. Заклади «Пузата Хата» працюють в найбільших містах: Київ, Дніпро, Львів, Одеса, також у Запоріжжі та Луцьку. Для забезпечення гостей закладами високої фізичної продукцією в Україні працюють три наших власних виробничих комплексів. Стрімкий темп розвитку «Пузаті Хати» та її стабільне існування на ринку стали можливими завдяки саме людям, які багато років створюють особливу атмосферу довіри та співпраці в нашій компанії.

Чому варто працювати в Пузатій Хаті

Офіційне працевлаштування

Гарантуємо соціальний захист працівника та своєчасну оплату зарплати і відпуск.

Високий рівень оплати праці та її завжди вчасна виплата

Люди - головний капітал компанії. Тому ми завжди вчасно відслідковуємо зарплату, незалежно від обставин.

Розпочни кар'єру незалежно від досвіду, та віку

Ми надіємо масу навчальної підтримки для всіх, незалежно від рівня досвіду у ресторанній справі.

Гнучкий робочий графік

Мало різні графіки роботи, напевно запропонуємо вакансії, зручні для кожного працівника.

Зручне розташування ресторанів

Велика кількість закладів у мережі дозволяє знайти роботу недалеко від дому.

Смачне корпоративне харчування

Смачні, якісні страви у власному об'єднаному комплексі.

Анкета для роботи в «Пузата хата»

Основна інформація

<input type="text" value="Арте"/>	<input type="text" value="Павлово"/>
<input type="text" value="ДД/ММ/РРРР"/>	<input type="text" value="Київський район"/>
<input type="text" value=""/>	<input type="text" value=""/>

Резюме

Перегляньте нашу вакансію або зв'яжіться з нами за електронною поштою hr@puzatkhat.com до 7 НОД.

Вкажіть більше інформації, щоб отримати справді цікаву вакансію

<input type="text" value="Оберть бажане місце для життя працівника"/>	<input type="text" value="Оберть бажану посаду"/>
<input type="text" value="Оберть бажаний ресторан"/>	<input type="text" value=""/>

Написавши відповідь, ви погоджуєтесь з обробкою та використанням своїх персональних даних.

[Відправити](#)

Перегляньте більше наших вакансій






Рисунок А.2 - Сторінка для заповнення анкети

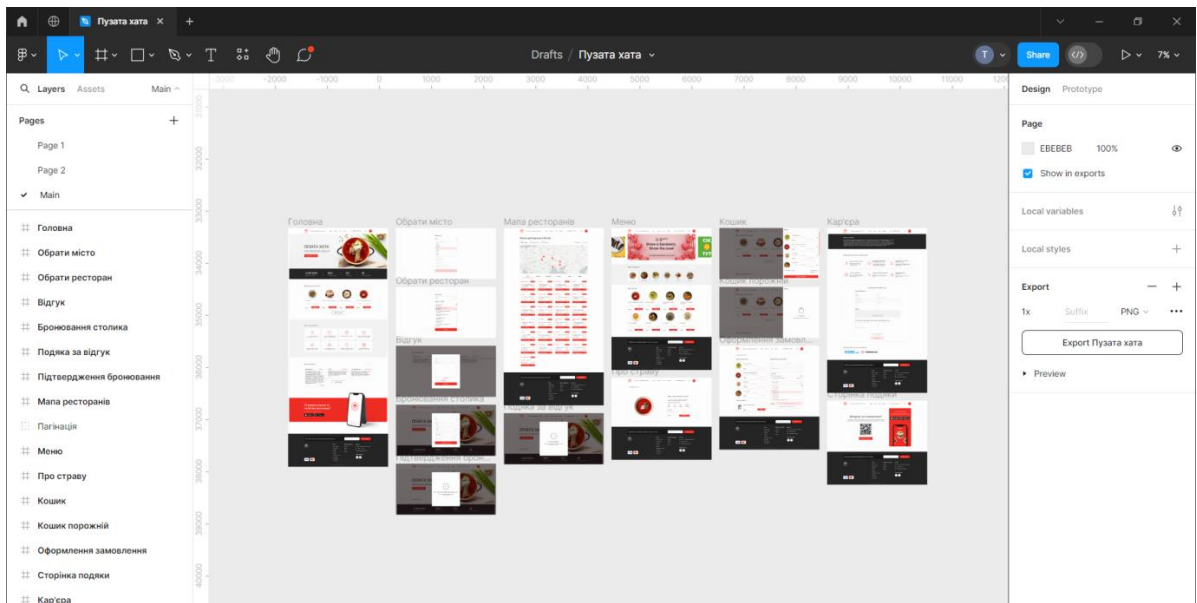


Рисунок А.3 - Дизайн макету у Figma, версія для моніторів

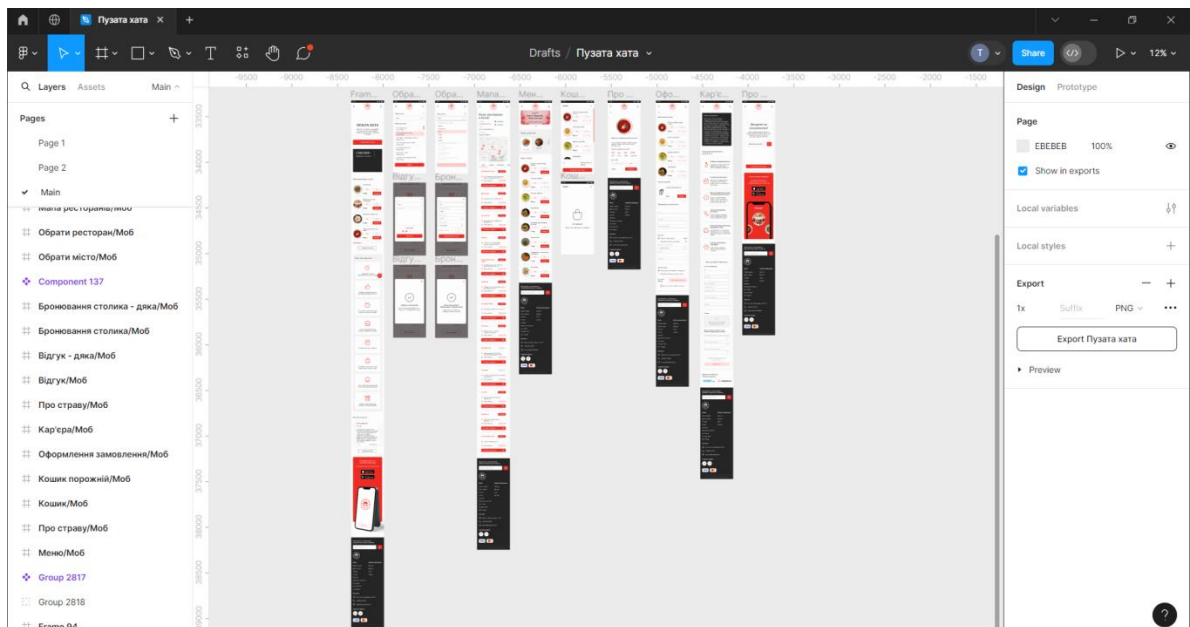


Рисунок А.4 - Дизайн макету у Figma, версія мобільна

Додаток Б. Код програми

Код валідації форм:

main.js:

```
// Form validation -----  
  
function validateForm() {  
  function validateTextField(input) {  
    if (input.value.length > 1) {  
      input.classList.remove('invalid')  
      return true;  
    } else {  
      input.classList.add('invalid')  
      return false;  
    }  
  }  
}  
  
function validateEmail(input) {  
  const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
  if (regex.test(input.value.trim())) {  
    input.classList.remove('invalid')  
    return true;  
  } else {  
    input.classList.add('invalid')  
    return false;  
  }  
}  
  
function validatePhone(input) {  
  let regex = new RegExp(`^\\+38 \\(\\d{3}\\) \\d{2}-\\d{2}-\\d{3}$`);  
  
  if (regex.test(input.value)) {  
    input.classList.remove('invalid')  
    return true;  
  } else {  
    input.classList.add('invalid')  
    return false;  
  }  
}  
  
function validatePassword(input) {  
  const regex = /^(?=.*\\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[^\\w\\d\\s]).{8,}$/;
```

```

    if (regex.test(input.value)) {
      input.classList.remove('invalid')
      return true;
    } else {
      input.classList.add('invalid')
      return false;
    }
  }
}

```

```

function validateConfirmPassword(form, input) {
  const password = form.querySelector('.password-input');
  if (input.value === password.value) {
    input.classList.remove('invalid')
    return true;
  } else {
    input.classList.add('invalid')
    return false;
  }
}

```

```

function validateIfEmpty(input) {
  if (input.value.length) {
    input.classList.remove('invalid')
    return true;
  } else {
    input.classList.add('invalid')
    return false;
  }
}

```

```

const forms = document.querySelectorAll('.validate-form');

```

```

forms.forEach((form) => {
  if (form.tagName === 'DIV') {
    form = form.querySelector('form');
  }
  const inputs = form.querySelectorAll('input[required], input[aria-
required="true"], textarea[aria-required="true"], textarea[required]');
  const selects = form.querySelectorAll('select[required]');
  const checkboxes = form.querySelectorAll('input[type="checkbox"]');
  const
      someCheckboxes
form.querySelectorAll('input[type="checkbox"].some');
  const
      requiredCheckboxes
form.querySelectorAll('input[type="checkbox"][required]');

```

```

const submitBtn = form.querySelector('[type="submit"]');

function validateTextInput(input) {
  if (input.classList.contains('password-input')) {
    return validatePassword(input);
  } else if (input.classList.contains('confirm-password-input')) {
    return validateConfirmPassword(form, input);
  } else if (input.type === 'text') {
    return validateTextField(input);
  } else if (input.type === 'email') {
    return validateEmail(input);
  } else if (input.type === 'tel') {
    return validatePhone(input);
  } else {
    return validateIfEmpty(input)
  }
}

function validateCheckboxes() {
  const someCheckboxInvalid = someCheckboxes.length &&
  ![...someCheckboxes].some(checkbox => checkbox.checked);
  const reqCheckboxInvalid = requiredCheckboxes.length &&
  ![...requiredCheckboxes].every(checkbox => checkbox.checked);

  function makeCheckboxInvalid(checkboxes) {
    checkboxes.forEach(checkbox => {
      const checkmark = checkbox.closest('.checkbox-
container').querySelector('.checkmark');
      if(checkmark) {
        checkmark.classList.add('invalid')
      }
    });
  }

  function makeCheckboxValid(checkboxes) {
    checkboxes.forEach(checkbox => {
      const checkmark = checkbox.closest('.checkbox-
container').querySelector('.checkmark');
      if(checkmark) {
        checkmark.classList.remove('invalid')
      }
    });
  }

  makeCheckboxValid(someCheckboxes)

```

```

makeCheckboxValid(requiredCheckboxes)

if (someCheckboxInvalid) {
  makeCheckboxInvalid(someCheckboxes)
  return false;
}

if (reqCheckboxInvalid) {
  makeCheckboxInvalid(requiredCheckboxes)
  return false;
}

return true;
}

submitBtn.addEventListener('click', e => {
  console.log('test')
  e.stopPropagation();
  const validateSelects = [...selects].every(select => select.value.length);

  inputs.forEach((input) => {
    validateTextInput(input);

    input.addEventListener('input', () => {
      validateTextInput(input);
    })
  });

  validateCheckboxes()

  if ([...inputs].some(input => input.value.length > 3)) {
    const allInputsValid = ([...inputs].every(validateTextInput) &&
validateCheckboxes() && validateSelects);
    inputs.forEach(input => {
      validateTextInput(input);
    })
    if(!allInputsValid) {
      e.preventDefault();
    }
  }
}, true);

checkboxes.forEach(checkbox => {
  checkbox.addEventListener('change', () => {

```

```

        validateCheckboxes();
    });
});
});

document.addEventListener('wpcf7mailsent', function( event ) {
    setTimeout( () => {
        window.location = '/success/';
    }, 2000 );
}, false );
}

```

Код функції для вибору міста та ресторану:

```

const chooseRestaurantModal = document.querySelector('#choose-restaurant-modal');
const chooseRestaurantBtn = document.querySelector('#choose-restaurant-btn');
const openChooseRestaurantModal = document.querySelectorAll('.open-choose-restaurant-btn');
const currentRestaurantElement = document.querySelector('#current-restaurant span');
const closeModalBtn = chooseRestaurantModal.querySelector('.close-btn');
const currentCityElement = document.querySelectorAll('.current-city-paste');
let userData = JSON.parse(localStorage.getItem('userData'));

function isRestaurantSelected() {
    return userData && userData.restaurant;
}

if(!userData || !userData.modalClosed) {
    chooseRestaurantModal.classList.add('active');
    userData = {
        city: "",
        restaurant: "",
        modalClosed: false,
    };
} else {
    updateCurrentRestaurant()
    chooseRestaurantModal.classList.remove('active');
}
closeModalBtn.addEventListener('click', e => {
    chooseRestaurantModal.classList.remove('active');
    userData.modalClosed = true;
}

```

```

    updateUserData(userData);
  })
  openChooseRestaurantModal.forEach(btn => {
    btn.addEventListener('click', e => {
      chooseRestaurantModal.classList.add('active');
    })
  })
  chooseRestaurantBtn.addEventListener('click', e => {
    let userData = JSON.parse(localStorage.getItem('userData'));
    const citySelect = document.querySelector('[name="city"]');
    const resSelect = document.querySelector('[name="restaurant"]');
    if (citySelect.value === '0') {
      alert('Оберіть місто');
      return;
    } else if (resSelect.value === '0') {
      alert('Оберіть ресторан');
      return;
    }
    userData.city = citySelect?.value;
    userData.restaurant = resSelect.value;
    updateUserData(userData);
    updateCurrentRestaurant()
    chooseRestaurantModal.classList.remove('active');
    if(userData.initiator) {
      const modalWindow = document.querySelector(`[data-modal="${userData.initiator}"]`);
      openModal(modalWindow);
      userData.initiator = "";
      updateUserData(userData);
    }
  })

function updateUserData(data) {
  localStorage.setItem('userData', JSON.stringify(data));
}

function updateCurrentRestaurant() {
  const userData = JSON.parse(localStorage.getItem('userData'));
  if(userData.restaurant) {
    const label = userData.restaurant.length > 30 ? userData.restaurant.slice(0, 30) + '...' : userData.restaurant;
    currentRestaurantElement.textContent = label;
    currentRestaurantElement.title = userData.restaurant;
  }
}

```

```

    }
  }
  currentCityElement.forEach(el => {
    el.textContent = userData.city;
  })
  document.querySelectorAll('[data-tab-city]').forEach(el => {
    if(el.dataset.tabCity === userData.city) {
      el.classList.add('active');
    }
  })
  // fetch restaurants by city
  const citySelect = document.getElementById('city');
  const restaurantSelect = document.getElementById('restaurant');

  citySelect.addEventListener('change', function () {
    const selectedCity = this.value;
    fetchRestaurants(selectedCity);
  });
  function fetchRestaurants(city) {
    restaurantSelect.disabled = true;
    if (city === '0') {
      restaurantSelect.innerHTML = '<option value="0">Ресторан</option>';
      return;
    }
    fetch(`/wp-admin/admin-ajax.php?action=get_restaurants&city=${city}`)
      .then(response => response.json())
      .then(data => {
        restaurantSelect.innerHTML = '<option value="0">Ресторан</option>';
        data.data.restaurants.forEach(restaurant => {
          const option = document.createElement('option');
          const { name, address } = restaurant;
          // const label = ${name} (${address});
          const label = ${address};
          option.value = label;
          option.textContent = label;
          restaurantSelect.appendChild(option);
        });
        restaurantSelect.disabled = false;
      })
      .catch(error => {
        console.error('Error fetching restaurants:', error);
        restaurantSelect.disabled = false;
      });
  }
}

```

Код html даної функції:

```
<div class="solid-modal" id="choose-restaurant-modal">
  <div class="solid-modal__body">
    <div class="choose-restaurant-form">
      <button class="close-btn icon-btn">
        <i class="i-close"></i>
      </button>
      <div class="choose-restaurant-form__field">
        <label for="city">Оберіть місто</label>
        <?php $cities = get_cities(); ?>
        <select name="city" id="city">
          <option value="0">Місто</option>
          <?php foreach ($cities as $city) : ?>
            <option value="<?= $city ?>"><?= $city ?></option>
          <?php endforeach; ?>
        </select>
      </div>
      <div class="choose-restaurant-form__field">
        <label for="restaurant">Оберіть ресторан</label>
        <select name="restaurant" id="restaurant" disabled>
          <option value="0">Ресторан</option>
        </select>
      </div>
      <button class="btn primary w-full lg" id="choose-restaurant-
btn">Обрати</button>
    </div>
  </div>
</div>
```

Код для створення рекламних банерів:

```
const bannersSliders = document.querySelectorAll(".banners-slider");

if (bannersSliders.length) {
  bannersSliders.forEach((slider) => {
    const prev = slider.querySelector(".prev");
    const next = slider.querySelector(".next");

    const bannersSlider = new Swiper(slider, {
      spaceBetween: 0,
      speed: 700,
      grabCursor: true,
      keyboard: {
        enabled: true,
```

```

    },

    navigation: {
      prevEl: prev,
      nextEl: next,
    },
    pagination: {
      el: slider.querySelector('.swiper-pagination'),
      clickable: true
    },
    centeredSlides: true,
    slidesPerView: 1,
    initialSlide: 1
  });
});
}
const commonSlidersSections = document.querySelectorAll(".common-slider-section");
if (commonSlidersSections.length) {
  commonSlidersSections.forEach((sliderContainer) => {
    const slider = sliderContainer.querySelector(".common-slider");
    const prev = sliderContainer.querySelector(".prev");
    const next = sliderContainer.querySelector(".next");
    const initialSlide = slider?.getAttribute("data-initial-slide") || 0;
    const slidesPerView = slider?.getAttribute("data-slides-count") || 4;
    const isHalfSlide = slider?.getAttribute("data-half-slide") || true;
    const initForMobile = slider?.getAttribute("data-slider-init") === 'for-mobile' ||
false;
    const mobileSlidesPerView = slider?.getAttribute("data-mobile-slides-count") ||
1.2;

    if(initForMobile && window.screen.width > 992) return;
    const commonSlider = new Swiper(slider, {
      spaceBetween: 20,
      speed: 700,
      grabCursor: true,
      keyboard: {
        enabled: true,
      },
      navigation: {
        prevEl: prev,
        nextEl: next,
      },
    },

```

```

breakpoints: {
  1200: {
    slidesPerView,
    initialSlide,
  },
  992: {
    slidesPerView: 3.1,
    initialSlide,
  },
  768: {
    slidesPerView: 2.3,
    initialSlide: 0,
  },
  479: {
    slidesPerView: 1.5,
  },
  0: {
    slidesPerView: mobileSlidesPerView,
  },
},
});
});
}

```

Код заповнення анкети:

```

<div class="form-inputs checkout-form__inputs">
  <div class="form-title">Основна інформація</div>
  <div class="form-cols-inputs">
    <div class="form-field">
      <label for="firstname">Ваше ім'я<span class="required">*</span></label>
      <div class="input-container">
        [text* firstname id:firstname placeholder "Наприклад, Артем"]
        <span class="input-error">Заповніть це поле, будь ласка</span>
      </div>
    </div>
    <div class="form-field">
      <label
        for="lastname">Ваше
        прізвище<span
class="required">*</span></label>
      <div class="input-container">
        [text* lastname id:lastname placeholder "Наприклад, Богдаренко"]
        <span class="input-error">Заповніть це поле, будь ласка</span>
      </div>
    </div>
  </div>
</div>

```

```

</div>
<div class="form-cols-inputs">
  <div class="form-field">
    <label for="bday">День народження<span
class="required">*</span></label>
    <div class="input-container">
      [text* bday id:bday placeholder "дд/мм/рррр"]
      <span class="input-error">Заповніть це поле, будь ласка</span>
    </div>
  </div>
  <div class="form-field">
    <label for="town">Населений пункт<span
class="required">*</span></label>
    <div class="input-container">
      [text* town id:town placeholder "Наприклад, Київ"]
      <span class="input-error">Заповніть це поле, будь ласка</span>
    </div>
  </div>
</div>
<div class="form-cols-inputs">
  <div class="form-field">
    <label for="phone">Телефон<span class="required">*</span></label>
    <div class="input-container">
      [tel* phone id:phone class:phone placeholder "+380"]
      <span class="input-error">Заповніть коректно це поле, будь
ласка</span>
    </div>
  </div>
  <div class="form-field">
    <label for="email">Електронна адреса<span
class="required">*</span></label>
    <div class="input-container">
      [email* email id:email placeholder "Наприклад, example@gmail.com"]
      <span class="input-error">Заповніть це поле, будь ласка</span>
    </div>
  </div>
</div>
<div class="form-title">Резюме</div>
<div class="form-field">
  <label for="file-upload" class="dropzone">
    [file file-upload id:file-upload limit:7mb filetypes:png|jpg|doc|pdf]
    <span class="dropzone-content">
      <i class="i-cloud dropzone-content__icon"></i>
      <span class="dropzone-content__text">

```

```

        Перенесіть сюди документ або <span class="underlined">завантажте
його з комп'ютера</span>
        (png/jpg/doc/pdf до 7 Мб)
    </span>
</span>
</label>
</div>
<div class="form-title">Вкажіть більше інформації, щоб отримати справді цікаву
вакансію</div>
<div class="form-cols-inputs">
    <div class="form-field">
        <label for="city">Оберіть бажане місто де хочете працювати<span
class="required">*</span></label>
        <div class="input-container">
            [select* city id:city "Місто" "Київ" "Рівне"]
            <span class="input-error">Заповніть коректно це поле, будь
ласка</span>
        </div>
    </div>
    <div class="form-field">
        <label for="role">Оберіть бажану посаду<span
class="required">*</span></label>
        <div class="input-container">
            [select* role id:role "Посада" "Кухар" "Програміст"]
            <span class="input-error">Заповніть коректно це поле, будь
ласка</span>
        </div>
    </div>
</div>
<div class="form-field">
    <label for="rest">Оберіть бажаний ресторан<span
class="required">*</span></label>
    <div class="input-container">
        [select* restaurant id:rest "ресторан 1" "ресторан 2"]
        <span class="input-error">Заповніть коректно це поле, будь ласка</span>
    </div>
</div>
</div>
<div class="form-disclaimer">
    <span>Натискаючи Відправити, я погоджуюся з обробкою та зберіганням
своїх персональних даних.</span>
    [submit class:btn class:primary class:lg "Відправити"]
</div>
[telegram]

```