

Галицький фаховий коледж імені В'ячеслава Чорновола  
відділення комп'ютерних технологій  
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач відділення  
комп'ютерних технологій  
Наталія СТЕФУРАК/\_\_\_\_\_/\_\_\_\_\_  
підпис  
«\_\_» \_\_\_\_\_ 2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА  
до дипломного проєкту  
освітньо-кваліфікаційного рівня «молодший спеціаліст»  
зі спеціальності 122 «Комп'ютерні науки»  
на тему: «Web-додаток для організації автоматизації процесів ресторану»

Студент групи КН-41

Дуніковський С.Б

\_\_\_\_\_  
(підпис)

Керівник проєкту

Кузик В.М

\_\_\_\_\_  
(підпис)

Консультанти:

з техніко-економічного  
обґрунтування

Меленчук Л.І.

\_\_\_\_\_  
(підпис)

нормоконтролер

Кульчинська Н.З.

\_\_\_\_\_  
(підпис)

Тернопіль – 2023

Галицький фаховий коледж імені В'ячеслава Чорновола  
відділення комп'ютерних технологій  
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ  
Завідувач відділення  
комп'ютерних технологій  
Наталія СТЕФУРАК/\_\_\_\_\_/  
підпис  
«\_\_» \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**

на дипломне проєктування  
на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»  
студенту Дуніковському Стефану Богдановичу

---

(прізвище, ім'я та по-батькові студента)

1. Тема проєкту «Web-додаток для організації автоматизації процесів ресторану»  
затверджена наказом по коледжу від “16” грудня 2022 р., №1193-н
2. Термін здачі студентом завершеного проєкту “\_\_” \_\_\_\_\_ 2023 р.
3. Вихідні дані до проєкту \_\_\_\_\_  
\_\_\_\_\_
4. Перелік питань, які повинні бути розроблені в проєкті:
  - а) основна частина \_\_\_\_\_  
\_\_\_\_\_
  - б) техніко-економічне обґрунтування \_\_\_\_\_  
\_\_\_\_\_
5. Перелік графічного матеріалу \_\_\_\_\_  
\_\_\_\_\_
6. Консультанти проєкту:

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко-економічного обґрунтування	_____ (вчена ступень, звання П.І.Б. консультанта)		

## КАЛЕНДАРНИЙ ПЛАН

### дипломного проєктування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проєктування	14.11.22	16.12.22
2.	Огляд типових рішень та написання відповідного розділу ПЗ	16.12.22	27.01.23
3.	Дослідження технологій реалізації та написання відповідного розділу ПЗ	27.01.23	14.02.23
4.	Розробка функціональних вимог до проєкту та робота над структурою програмного продукту. Написання відповідного розділу ПЗ	17.02.23	2.03.23
5.	Встановлення на налаштування середовища реалізації та написання відповідного розділу ПЗ	2.03.23	16.03.23
6.	Проектування програмного засобу (функціоналу, інтерфейсу, бази даних продукту) та написання відповідного розділу ПЗ	16.03.23	17.04.23
7.	Реалізація та налаштування програмного засобу та написання відповідного розділу ПЗ	17.04.23	4.05.23
8.	Доопрацювання модулів	5.05.23	18.05.23
9.	Тестування на налагодження програмного продукту та написання відповідного розділу ПЗ	18.05.23	19.06.23
10.	Опрацювання економічного розділу дипломного проєкту та оформлення спеціального розділу	19.05.23	5.06.23
11.	Робота над оформленням пояснювальної записки	8.06.23	18.06.23
12.	Попередній захист дипломного проєкту, доопрацювання	15.06.23	22.06.23
13.	Підготовка до захисту дипломного проєкту	15.06.23	25.06.23
14.	Захист дипломного проєкту	26.06.23	26.06.23

7. Дата видачі завдання “ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

Керівник \_\_\_\_\_ /

Завдання прийняв до виконання \_\_\_\_\_ /

## Реферат

Web-додаток для організації автоматизації процесів ресторану. Дипломний проєкт. Дуніковський Стефан Богданович. Галицький фаховий коледж імені В'ячеслава Чорновола, відділення комп'ютерних технологій. Спеціальність 122 «Комп'ютерні науки». ГК, 2023. Сторінок – 79, рисунків – 36, додатків – 3.

Об'єкт дослідження – методи та засоби автоматизації бізнес-процесів, інструменти для розробки та впровадження інформаційних систем, клієнт-серверна архітектура, методи та інструменти розробки адаптивних веб-інтерфейсів для взаємодії з базою даних.

Метою проєкту є реалізація інформаційної системи, яка забезпечить автоматизацію процесів харчування приватної фірми та надасть можливість для подальшого розвитку автоматизації та удосконалення процесів, які базуються на результатах попередніх.

Система повинна бути реалізована у вигляді бази даних, яка зберігатиме записи, буде подавати інформацію у структурованому вигляді та адаптивного інтерфейсу для взаємодії з нею, який забезпечить високу доступність системи.

Інформаційна система повинна забезпечити цілісність та доступність даних та коректно виконувати запроєктовані функції, які забезпечують автоматизацію основних бізнес-процесів.

Для реалізації даної системи було використано інструменти, які надає мова програмування JS та СУБД Mongo.

Результатом розробки стала завершена система, яка виконує всі запроєктовані функції та готова до впровадження.

ІНФОРМАЦІЙНА СИСТЕМА, АВТОМАТИЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ, РНР, АРАСНЕ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, БАЗИ ДАНИ.

## Abstract

Web-application for organizing the automation of restaurant processes. Diploma project. Dunikovskyi Stefan Bogdanovych. Halych Vocational College named after Vyacheslav Chornovol, department of computer technologies. Specialty 122 "Computer Science". GK, 2023. Pages – 93, figures – 36, appendices – 3.

The object of research is methods and tools for automating business processes, tools for developing and implementing information systems, client-server architecture, methods and tools for developing adaptive web interfaces for interacting with a database.

The goal of the project is the implementation of an information system that will ensure the automation of the catering processes of a private company and provide an opportunity for further development of automation and improvement of processes based on the results of previous ones.

The system should be implemented in the form of a database that will store records, provide information in a structured form and an adaptive interface for interacting with it, which will ensure high availability of the system.

The information system must ensure the integrity and availability of data and correctly perform the designed functions that ensure the automation of the main business processes.

The tools provided by the JS programming language and the Mongo DBMS were used to implement this system.

The result of the development was a completed system that performs all the designed functions and is ready for implementation.

INFORMATION SYSTEM, AUTOMATION OF BUSINESS PROCESSES, PHP, APACHE, CLIENT-SERVER ARCHITECTURE, DATA BASES.

## ЗМІСТ

Вступ .....	7
1 Аналіз існуючих рішень та постановка завдання .....	9
1.1 Дослідження об'єкту інформатизації.....	9
1.2 Аналіз існуючих рішень.....	10
1.3 Обґрунтування доцільності створення системи .....	19
1.4 Постановка завдання .....	20
1.5 Дослідження технологій реалізації .....	22
2 Проєктування інформаційної системи.....	24
2.1 Поняття та важливість проєктування користувацького інтерфейсу .	24
2.2 Формалізація вимог .....	26
2.3 Побудова дерева функцій та проєктування структури системи .....	28
2.4 Проєктування бази даних.....	30
2.5 Проєктування користувацького інтерфейсу .....	33
3 Реалізація та тестування інформаційної системи .....	35
3.1 Опис технологій та засобів реалізації.....	35
3.2 Реалізація основних функцій інформаційної системи .....	38
3.3 Реалізація користувацького інтерфейсу системи .....	39
4 Техніко-економічне обґрунтування .....	48
4.1 Аналіз ринку.....	48
4.3 Обґрунтування необхідності розробки.....	49
Висновки.....	50
Перелік джерел посилання.....	51
Додатки .....	52

					ДП.КН 23.505.08.000 ПЗ				
Зм.	Арк.	№ докум.	Підпис	Дат					
Розроб.		Дуніковський С.Б.			Web-додаток для організації автоматизації процесів ресторану	Літ.		Арк.	Аркуші
Перев.		Кузик В.М.						5	101
Рецензет.		Чубей О.О.				ГФК.ВКТ.ЦКІКД КН-41			
Н. Контр.		Кульчинська Н.З.							
Зав. від.		Стефурак Н.А.							

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІС – Інформаційна система.

СУБД – Система управління базами даних.

MS – Microsoft.

ОС – Операційна система.

ЗП – збережувальна процедура.

HTML – HyperText Markup Language

SASS – Syntactically Awesome Stylesheets.

UI – user interface.

API – Application Programing Interface.

ІТ – Інформаційні технології.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дат		

## ВСТУП

У сучасному світі, де технології займають все більш важливе місце в нашому житті, ресторанний бізнес також не може залишатися осторонь цих тенденцій. З кожним роком, ресторани все більше стикаються з конкуренцією та вимогами клієнтів, що змушує їх удосконалювати свої бізнес-процеси та пропонувати нові технологічні рішення, які допомагатимуть ефективніше управляти бізнесом та поліпшувати якість обслуговування клієнтів.

Одним з головних викликів для ресторанів є оптимізація процесів та забезпечення якості обслуговування. Сьогоднішні клієнти мають високі вимоги до швидкості та якості обслуговування, тому ресторани повинні дбайливо підходити до цього питання. І одним з ключових рішень для досягнення цієї мети є використання технологій та автоматизації.

Метою даного дипломного проєкту є розробка web-додатку, який допоможе організувати та автоматизувати процеси ресторану. Додаток буде допомагати управляти замовленнями, резервуванням столиків, управлінням меню та складом, а також забезпечувати ефективну комунікацію між співробітниками ресторану. Розроблений додаток буде включати різноманітні функції, які допоможуть ресторану ефективніше функціонувати та підвищити рівень обслуговування.

Для досягнення поставленої мети, будуть використані сучасні технології програмування та бази даних, такі як HTML, SASS, JavaScript, ReactJs, NextJs, NodeJs та MongoDB.

При розробці додатку буде звернена особлива увага на його зручність та користувацьку досвід. Важливо, щоб користувачі змогли швидко та легко зрозуміти функціонал додатку та його можливості.

Також буде надана можливість додавати нові функції та розширювати функціональність додатку, щоб він міг відповідати змінним потребам ресторану.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дат		



Одним з основних викликів під час розробки додатку буде забезпечення безпеки та захисту особистої інформації клієнтів та працівників ресторану. Для цього будуть застосовуватися сучасні методи та алгоритми шифрування, щоб забезпечити захист даних.

Отже, розробка web-додатку для організації автоматизації процесів ресторану є актуальною та важливою задачею, яка дозволить покращити ефективність та якість обслуговування клієнтів, а також зробити ресторанний бізнес більш конкурентоспроможним. Розробка цього додатку буде виконана з використанням сучасних технологій та методів програмування, з урахуванням потреб та вимог користувачів.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дат		

# 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ПОСТАНОВКА ЗАВДАННЯ

## 1.1 Дослідження об'єкту інформатизації

Дослідження об'єкту інформатизації є важливим етапом в процесі розробки будь-якої інформаційної системи. У даному дипломному проєкті об'єктом інформатизації є процеси ресторану, що передбачає автоматизацію різних аспектів роботи, від прийому замовлень до обліку запасів та фінансових показників.

Для дослідження об'єкту інформатизації було проведено аналіз існуючих рішень та сайтів, які займаються схожими завданнями. Серед сайтів, що були проаналізовані, можна виділити «Choiceqr», «Oddmenu» та «Evrasia»

Результатом дослідження об'єкту інформатизації є постановка завдань для подальшого розвитку власного веб-додатку для автоматизації процесів ресторану. Зокрема, це передбачає створення системи замовлень, обліку запасів, контролю фінансових показників та багато іншого.

Крім того, ресторанний бізнес є взаємодією зі зовнішнім середовищем, що включає в себе постачальників, клієнтів та конкурентів. Для ефективної роботи ресторану необхідно вести контроль над всіма аспектами, пов'язаними з цими групами людей. Web-додаток може допомогти відстежувати статистику продажів, замовлень, запасів продуктів та інших важливих показників, що дозволить вчасно реагувати на потреби ринку та збільшувати ефективність роботи.

Метою дипломного проєкту є розробка web-додатку для автоматизації процесів ресторану, що дозволить значно полегшити роботу персоналу, збільшити ефективність роботи та зменшити витрати. Для досягнення цієї мети будуть використані сучасні технології та інструменти розробки програмного забезпечення, такі як React, Node.js, MongoDB тощо.

Кожен з цих сайтів має свої переваги та недоліки, але загалом вони допомагають клієнтам замовляти їжу онлайн, а ресторанам забезпечують автоматизацію процесів роботи.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

## 1.2 Аналіз існуючих рішень

В сучасному світі ресторанний бізнес стає все більш конкурентним, і для успішного функціонування необхідно мати ефективні інструменти автоматизації процесів. Відтак, виникає потреба у розробці веб-додатку, що допоможе відслідковувати та керувати різними процесами в ресторані.

Першим веб-додатком, який був проаналізований є «Choiceqr». Це веб-додаток, який надає послуги автоматизації процесів ресторану. Головна сторінка сайту пропонує користувачам вибір між різними опціями послуг, такими як QR-меню, онлайн-бронювання та замовлення на винос. Крім того, ви можете переглянути інформацію про компанію та її послуги, а також зв'язатися з командою підтримки.

Дизайн сайту доволі сучасний та затишний (рисунок 1.1). Колірна палітра містить приємні, теплі відтінки, які підходять для ресторанного бізнесу. Фотографії страв та інтер'єру додають настрою та створюють затишну атмосферу. Інтерфейс доволі інтуїтивно зрозумілий та легкий у використанні.

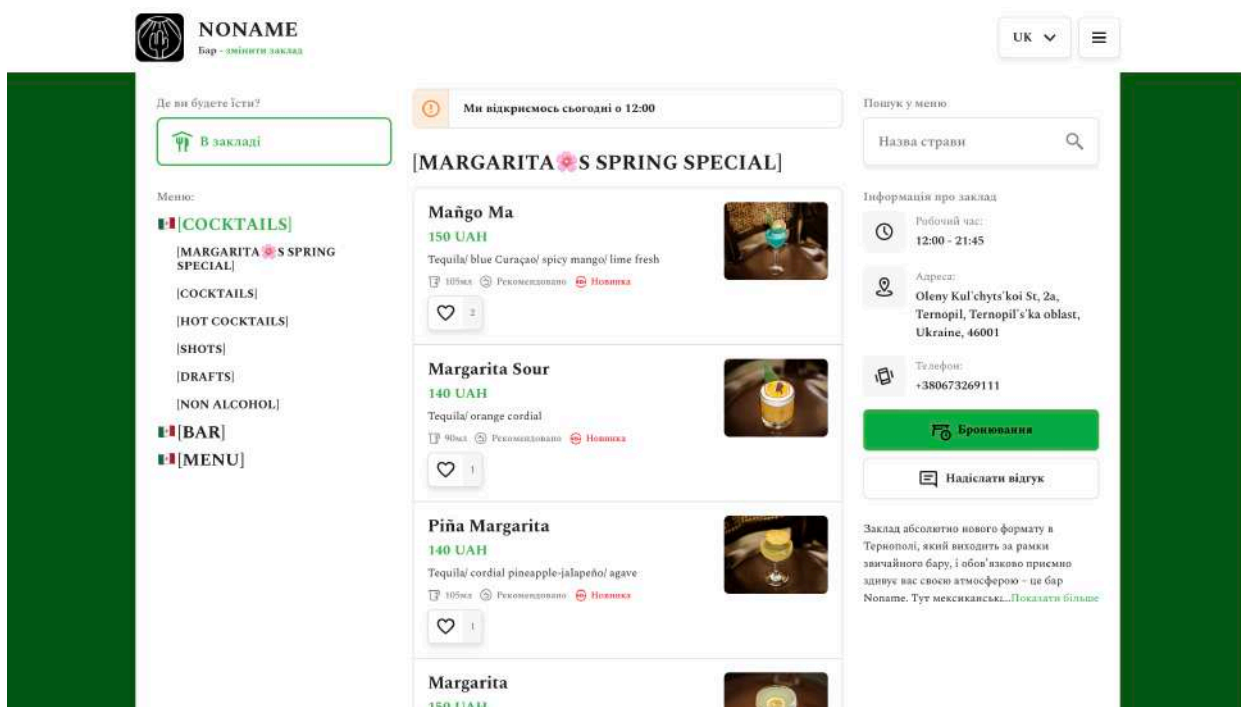


Рисунок 1.1 – Головна сторінка Choiceqr

Проте, є деякі недоліки. Наприклад, на головній сторінці може бути більше інформації про послуги компанії та їх переваги. Крім того, деякі

					ЛП.КН 23.505.08.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		10

розділи, такі як "Про нас" та "Контакти", можуть бути краще розташовані на головній сторінці замість того, щоб бути доступними тільки через гамбургер-меню (рисунок 1.2).

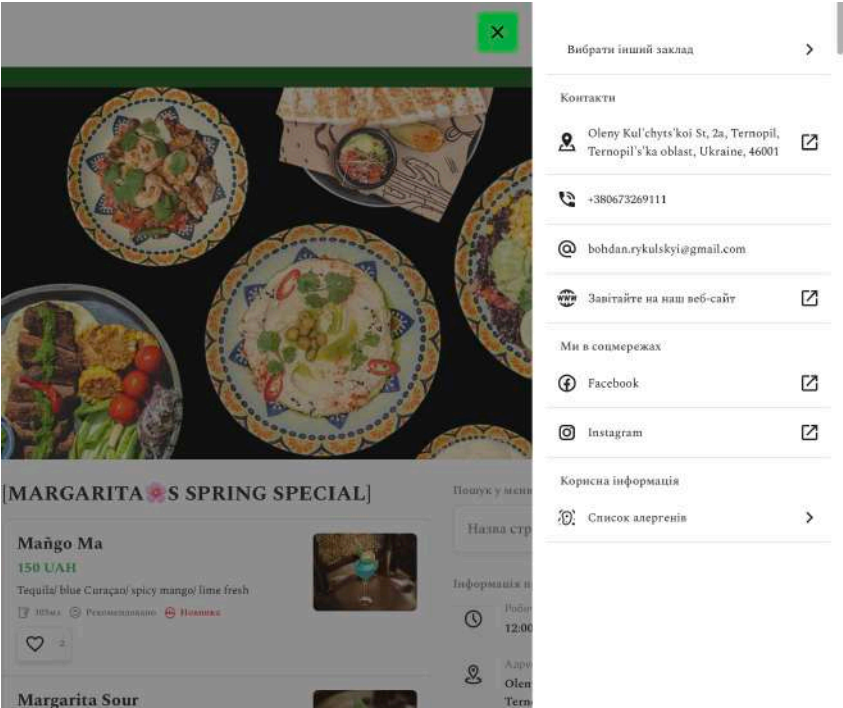


Рисунок 1.2 – Гамбургер-меню Choiseqr

Сайт [poname.choiseqr.com](http://poname.choiseqr.com) пропонує користувачам багато корисних функцій для автоматизації процесів ресторану. Наприклад, можна створити QR-код для меню, який клієнти можуть сканувати за допомогою свого смартфона та переглядати страви та ціни. Крім того, є можливість бронювання столиків онлайн (рисунок 1.3).

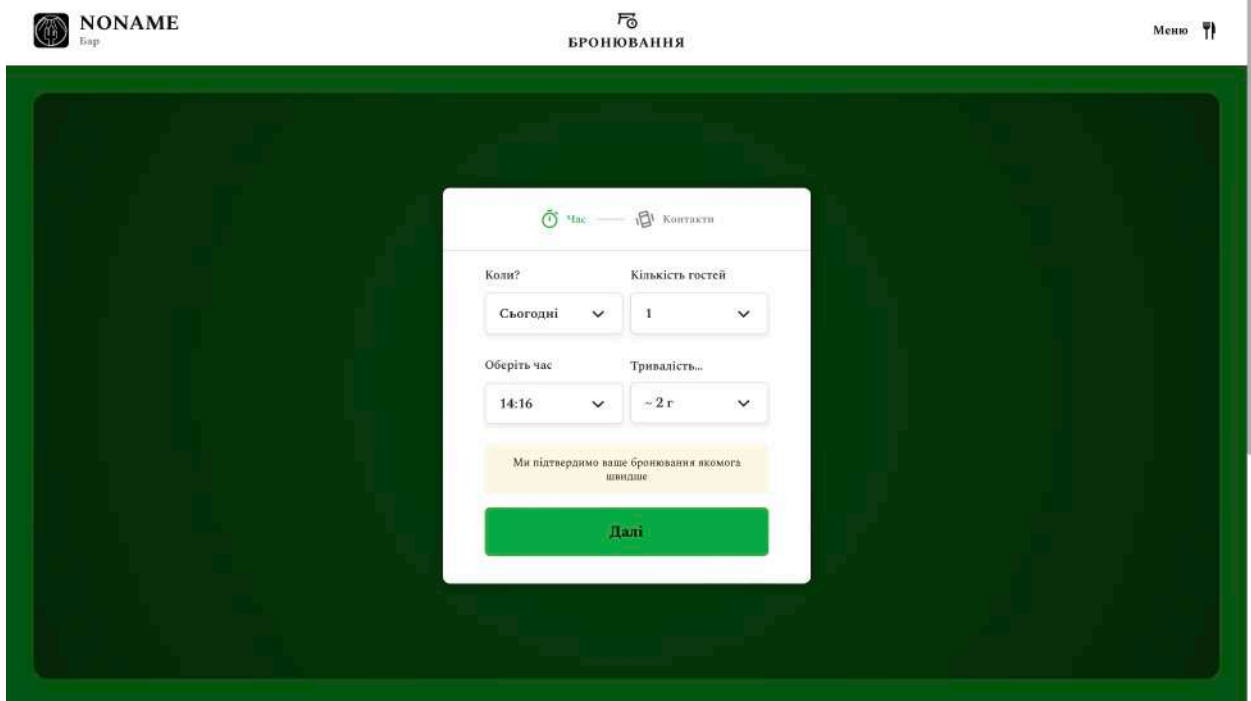


Рисунок 1.3 – Бронювання столів в Choiseqr

Інтерфейс дуже простий та легкий у використанні. Користувач може швидко та легко знайти потрібну інформацію та взаємодіяти з сайтом.

Проте, є деякі проблеми з функціональністю. Наприклад, деякі кнопки можуть бути неочевидними для користувача, або функціонал може бути недостатнім для деяких користувачів. Крім того, сайт може бути повільним для завантаження, особливо при використанні на мобільному пристрої.

З огляду на те, що сайт [noname.choiseqr.com](https://noname.choiseqr.com) пропонує послуги з обробки та зберігання даних користувачів, безпека та захист даних є дуже важливими аспектами.

Сайт має SSL-шифрування, що забезпечує захист передачі даних між браузером та сервером. Крім того, сайт використовує різні заходи безпеки, такі як двофакторна автентифікація та регулярні аудити безпеки.

Проте, є деякі питання щодо захисту даних користувачів. Наприклад, не зрозуміло, як відбувається зберігання та обробка даних користувачів. Крім того, можливо, не всі дані захищені від несанкціонованого доступу.

Загалом, сайт [noname.choiseqr.com](https://noname.choiseqr.com) має декілька плюсів, таких як привабливий дизайн та багатий функціонал. Проте, є деякі недоліки, такі як

					ЛП.КН 23.505.08.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		12

недостатня інформація на головній сторінці та можливість повільного завантаження. Окрім того, є деякі питання щодо безпеки та захисту даних користувачів.

Другим веб-додатком є «Le Rock» на платформі Oddmenu.com займає естетичний та чіткий дизайн. Головний екран містить фотографії страв та декору ресторану, які сприймаються як привабливі та збалансовані. Однак, можна зазначити, що багато відкритого простору на головній сторінці, що може створити відчуття нестачі інформації для користувача в десктопному режимі (рисунок 1.4).

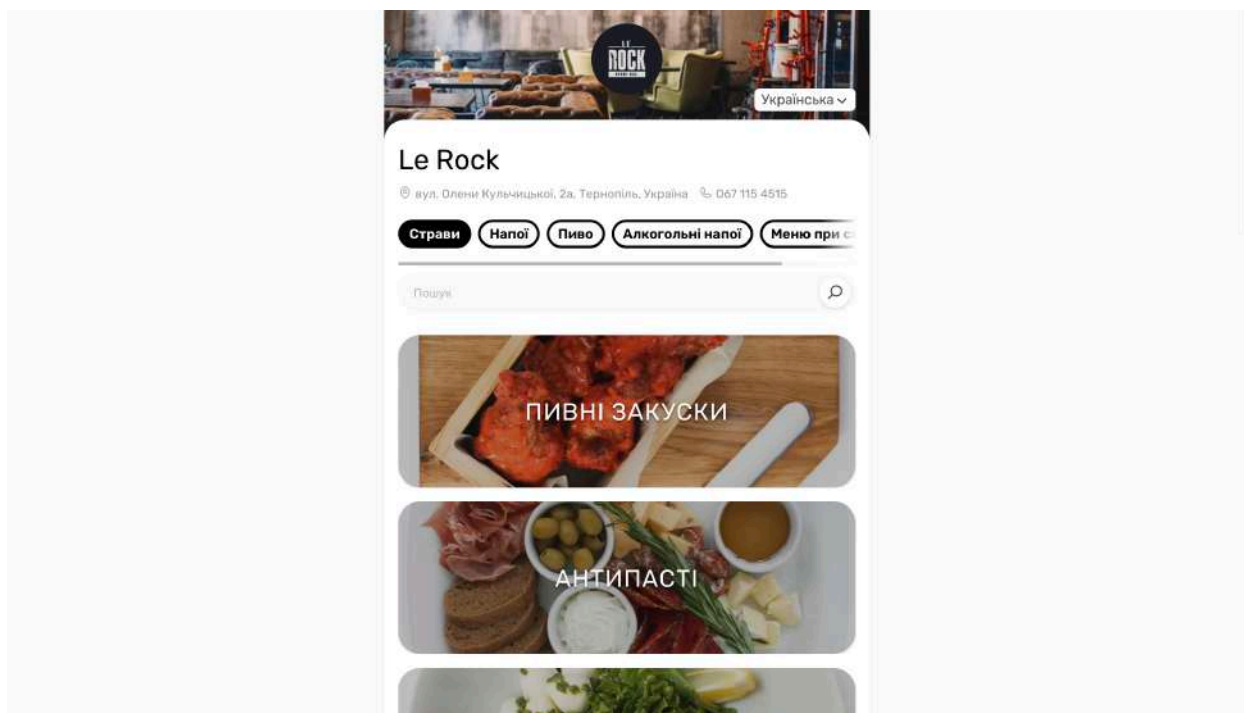


Рисунок 1.4 – Головний екран «Le Rock»

Меню розташовано вгорі сторінки та містить вибір мов, таких як англійська та українська, що є плюсом для веб-додатку, який має міжнародну аудиторію. Всі розділи меню чітко розміщені та навігація є зручною. Проте, не досить очевидною може виявитися функція пошуку страв, яка може бути корисною для швидкого доступу до інформації про певну страву.

Основною перевагою пратформи є велика кількість фотографій страв, що можуть привернути увагу та збудити апетит (рисунок 1.5).

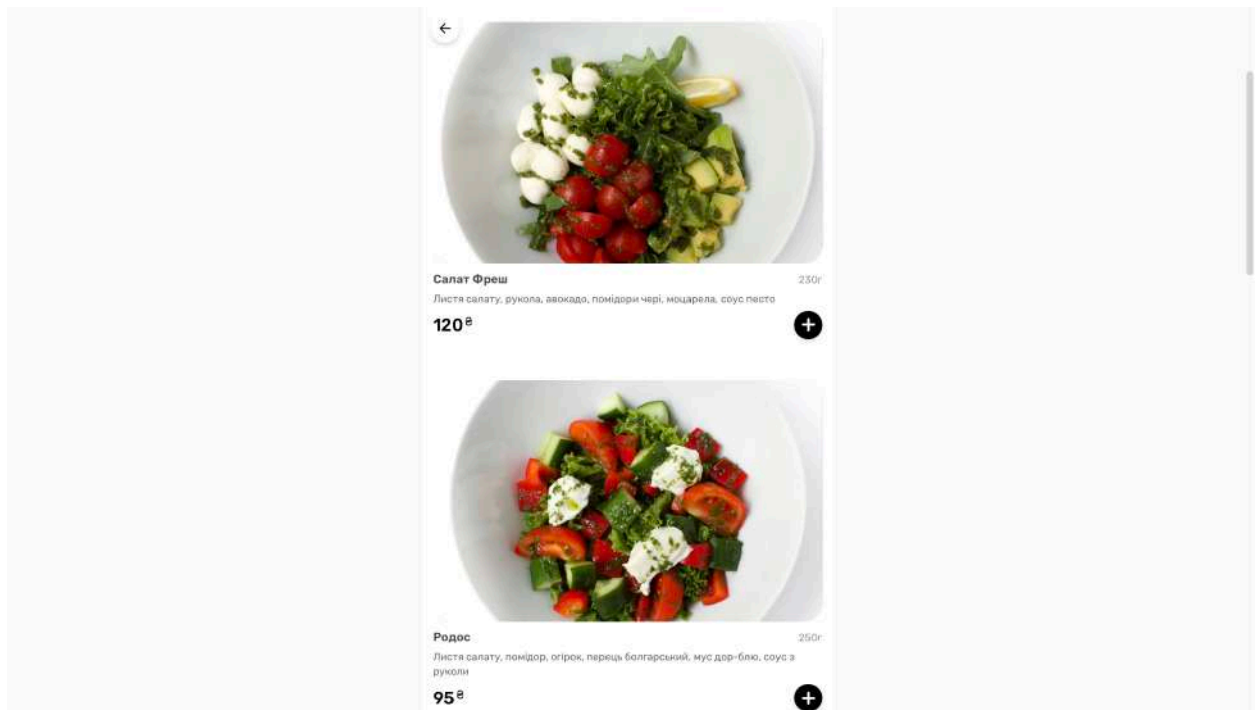


Рисунок 1.5 – Страви в «Le Rock»

Описи страв є досить короткими, але інформативними. Проте, у деяких страв може відсутній склад або інформація про особливості приготування, що може призвести до питань у відвідувачів ресторану.

Додатковою перевагою є можливість перегляду фотографій декору ресторану та атмосфери, що може допомогти в прийнятті рішення про відвідування ресторану.

Контактна інформація розміщена вгорі сторінки та включає адресу, номер телефону та електронну пошту. Проте, не знайдено інформації про робочий час ресторану, що може бути важливим для користувачів, які планують відвідати ресторан.

В цілому, на платформа має зручну навігацію, привабливий дизайн та достатньо інформації про ресторан. Однак, деякі недоліки, такі як відсутність інформації про робочий час та підтвердження замовлення, можуть створити певні незручності для користувачів. Крім того, відсутність рейтингу та відгуків може створювати певну невизначеність для користувачів, які хочуть дізнатися більше про якість обслуговування та їжі.



На основі аналізу можна зробити деякі рекомендації для поліпшення веб-сайту ресторану "Le Rock" на платформі Oddmenu.com. Наприклад, варто додати інформацію про робочий час ресторану, яка допоможе користувачам планувати свій час та уникнути незручностей. Крім того, можна додати функціонал підтвердження замовлення та скасування, що полегшить процес бронювання столиків та збільшить задоволеність користувачів.

Третім веб-додатком є «Evrasia». Головна сторінка зроблена в світлих тонах і містить банер зі зображенням страв та інформацію про ресторан. Крім того, вона містить меню з можливістю вибору категорій страв. Дизайн сайту є простим і зрозумілим для користувача. На головній сторінці є також можливість перейти до кількох розділів: меню, акції, доставка, контакти. При натисканні на кожен з цих розділів користувач може отримати детальнішу інформацію про ресторан.

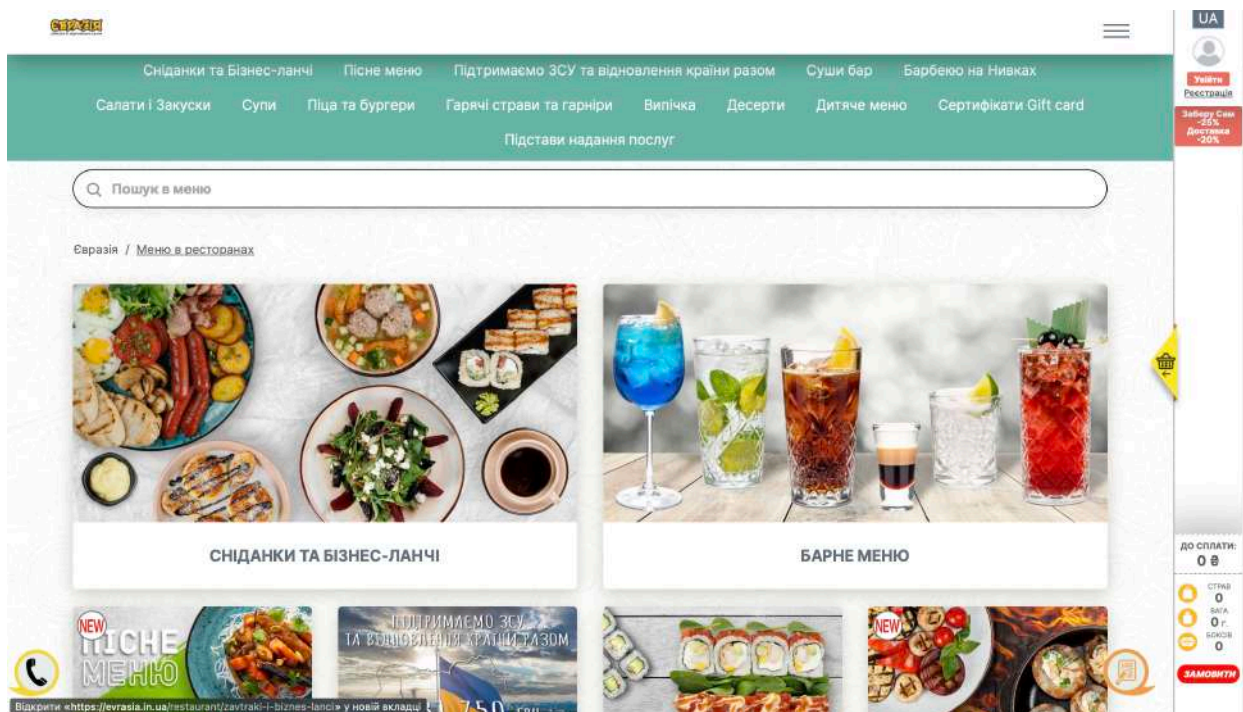


Рисунок 1.6 – Головна сторінка «Evrasia»

Сайт містить детальну інформацію про страви та їх склад. В меню є можливість вибору страви за категорією, типом страви, складом, а також можливість сортування за ціною. Кожна страва має власну сторінку зі



зображенням, описом та вартістю. Крім того, на сайті є розділ з акціями та можливістю замовлення доставки (рисунки 1.7).

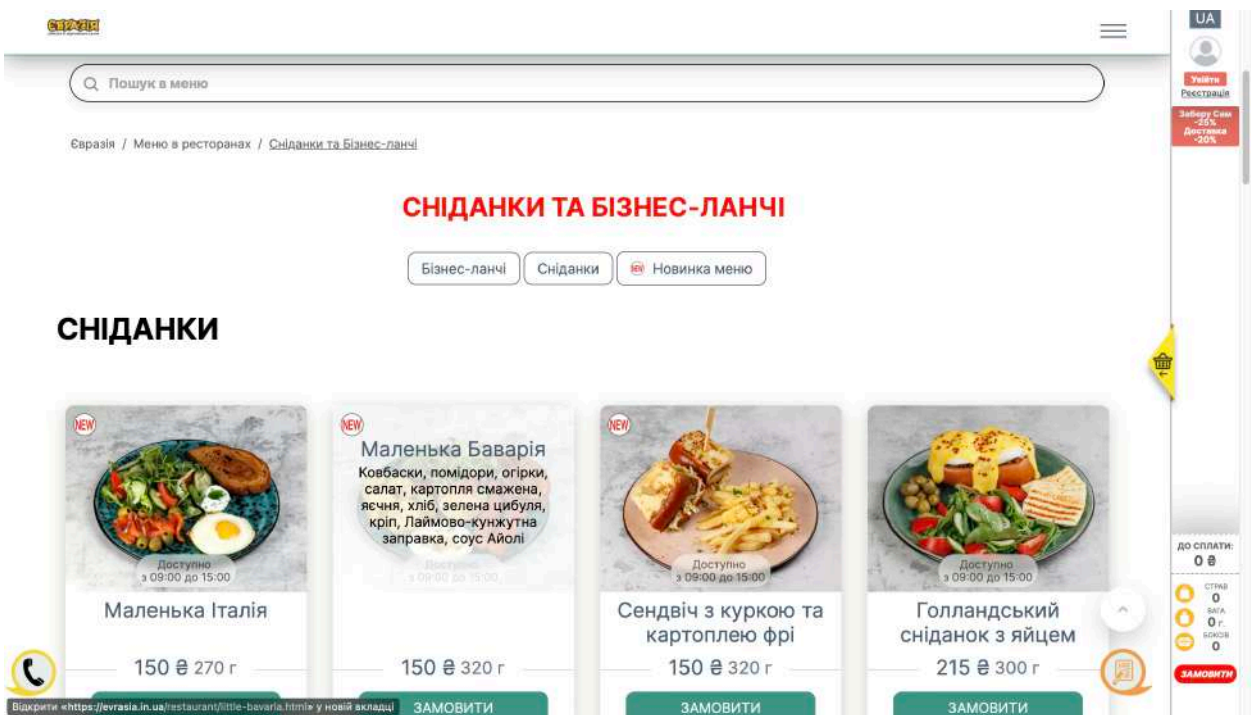


Рисунок 1.7 – Розділ з акціями та доставкою

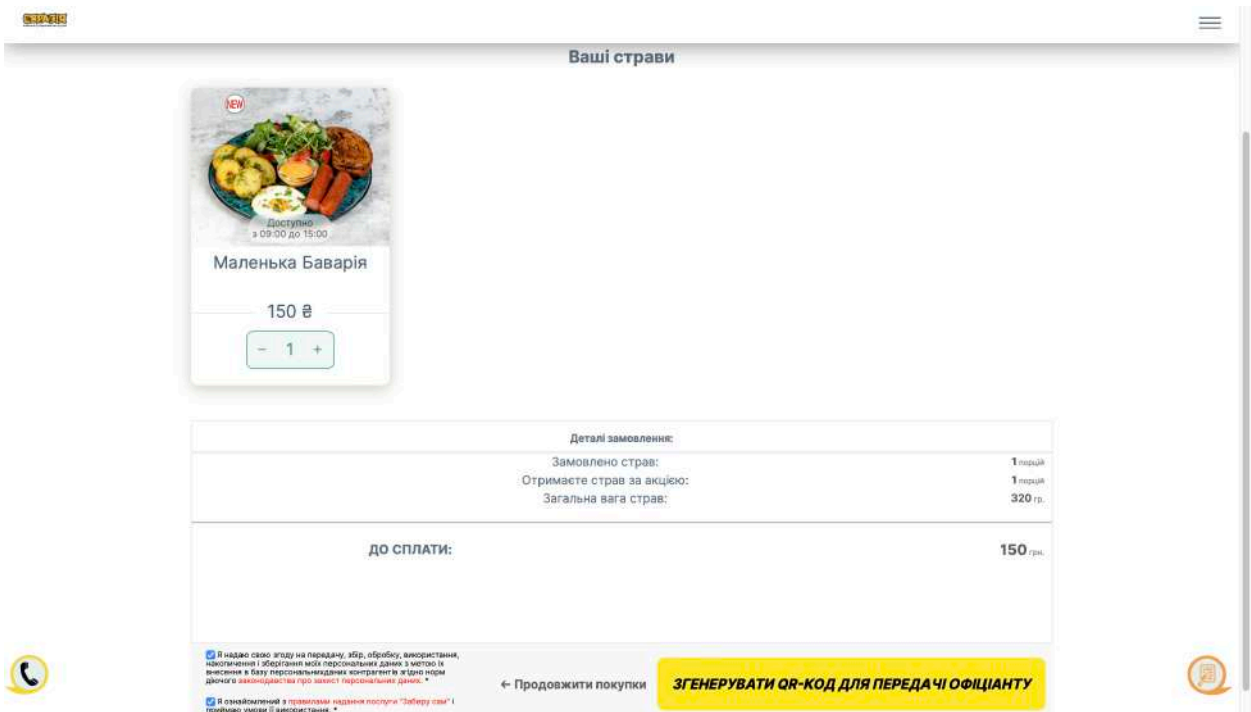


Рисунок 1.8 – Вкладка прийнятого замовлення

Сайт має просту і зрозумілу навігацію, яка дозволяє користувачеві легко знайти необхідну інформацію. Крім того, сайт має швидкий час завантаження

сторінок, що дозволяє користувачеві швидко переглянути інформацію. Сайт має можливість замовлення доставки з вказанням адреси та вартості доставки.

До переваг сайту можна віднести: простий та зрозумілий дизайн, детальний опис страв та їх склад, можливість замовлення доставки та просту навігацію. Однак, на сайті недостатньо інформації про відгуки клієнтів та рейтинг ресторану, а також відсутність можливості онлайн-бронювання столів. Крім того, на сайті немає розділу "Про нас", де можна було б дізнатися більше про історію ресторану та його концепцію. Також варто зазначити, що на деяких сторінках сайту зображення страв можуть бути недостатньо якісні, що може впливати на враження користувача.

Також варто зазначити, що на сайті Evrasia досить зручна система пошуку страв за назвою (рисунк 1.9). Користувач може швидко знайти потрібну страву та дізнатися про її склад, ціну та фотографію. Також на сайті присутня можливість додати страву до кошика та зробити замовлення онлайн, що також є досить зручною функцією.

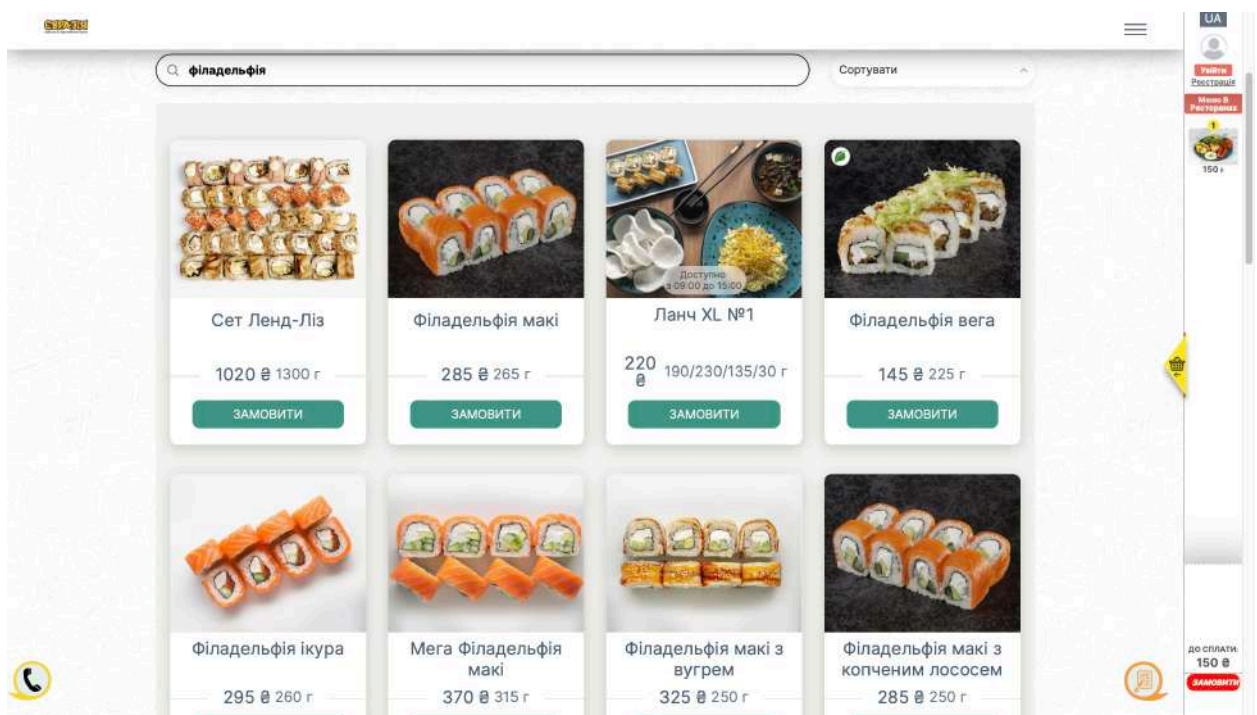


Рисунок 1.9 – Пошук страви

У розділі "Контакти" користувач може знайти адресу та години роботи ресторану, а також зв'язатися з ним за допомогою форми зворотного зв'язку

					ЛП.КН 23.505.08.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		17

(рисунок 1.10). Крім того, на сайті присутня інформація про додаткові послуги, які надає ресторан, такі як організація банкетів, кетеринг та інші.

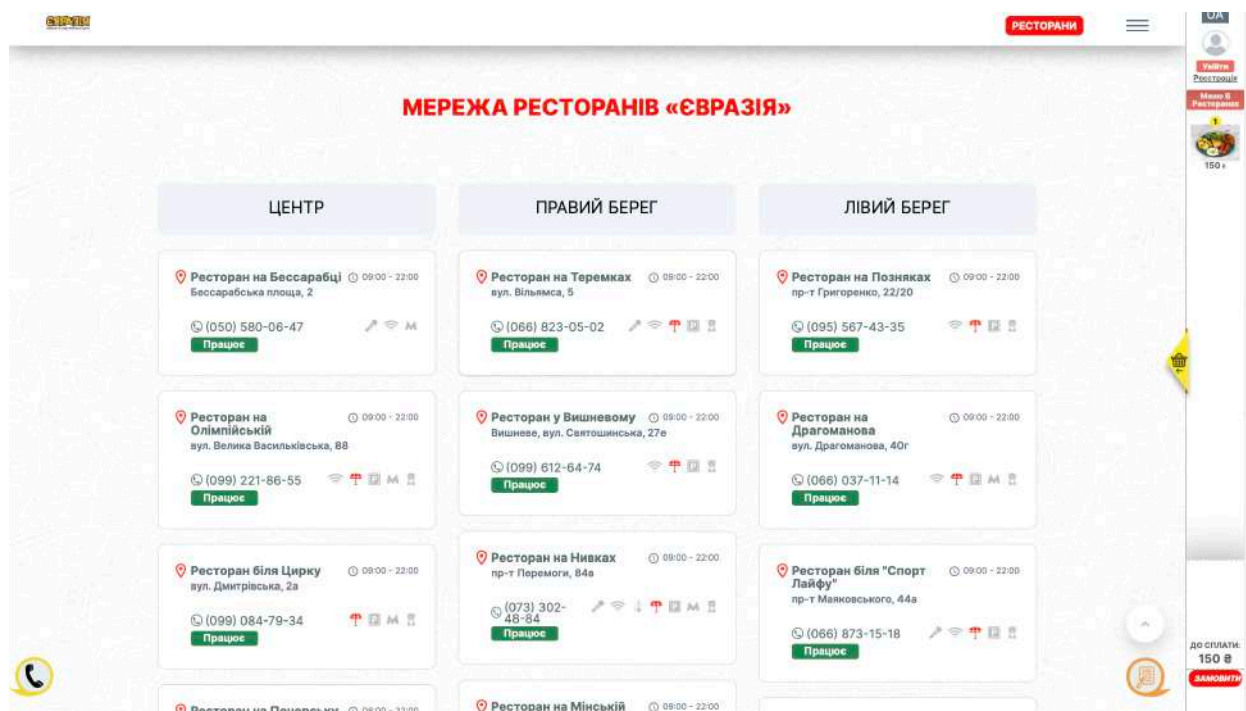


Рисунок 1.10 – Адреса та години роботи мережі «Evrasia»

Узагалі, сайт «Evrasia» має багато плюсів, але також має певні недоліки, які можна покращити. Для розробки web-додатка для організації автоматизації процесів ресторану можна використати деякі ідеї та рішення, які вже існують на сайті. Наприклад, можна створити зручне меню з можливістю сортування та фільтрації страв за категоріями, типами страви та складом. Також варто додати відгуки клієнтів та рейтинг ресторану. Для покращення враження користувача варто звернути увагу на якість фотографій та зробити дизайн сайту ще більш привабливим та зручним для користувача.

Загалом, сайт «Evrasia» має зручну навігацію та досить простий та зрозумілий інтерфейс. Однак варто звернути увагу на те, що на деяких сторінках сайту можуть бути перевантажені тексти та недостатньо простору для дихання, що може впливати на зручність користування сайтом. Також варто звернути увагу на адаптивність сайту та його доволі повільну швидкість завантаження.

### 1.3 Обґрунтування доцільності створення системи

В сучасному світі інтернет став невід’ємною складовою життя людей. Він надає можливості для швидкого та ефективного спілкування, здійснення покупок, збору інформації та багато іншого. У свою чергу, ресторанна галузь займає важливе місце в галузі харчування та відпочинку населення. Завдяки зручному місцезнаходженню, смачній їжі та приємній атмосфері, ресторани стають улюбленим місцем відпочинку для людей різного віку та соціального статусу.

Проте в сучасному світі, де час – найцінніший ресурс, ресторани повинні не лише надавати якісні послуги, але й забезпечувати їх швидку та ефективну доставку. Саме тому необхідно створити Web-додаток, що дозволить організувати автоматизацію процесів ресторану. Це не лише полегшить роботу персоналу, але й забезпечить клієнтам зручний та швидкий сервіс.

Створення Web-додатку для організації автоматизації процесів ресторану має безліч переваг. По-перше, це дозволить ресторану збільшити кількість замовлень та знизити витрати на обслуговування клієнтів. По-друге, система автоматизації дозволить покращити облік та контроль за складом продуктів, оптимізувати роботу кухні та зменшити кількість помилок при замовленні.

Доцільність створення Web-додатку для організації автоматизації процесів ресторану стає ще більш очевидною, якщо порівнямо його з традиційною системою обслуговування в ресторані. Традиційна система, яка базується на паперових замовленнях та їх обробці вручну, часто викликає затримки та незручності для клієнтів та персоналу. Більше того, така система може призвести до помилок в замовленнях та збільшення часу обробки замовлень, що в свою чергу може знизити задоволення клієнтів та призвести до втрати прибутку.

Завдяки Web-додатку, клієнти зможуть швидко та зручно робити замовлення, переглядати меню та отримувати повідомлення про статус їхнього замовлення. А персонал зможе оптимізувати свою роботу, спрощуючи процеси

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дат		

приготування та обробки замовлень, отримувати повідомлення про нові замовлення та керувати запасами продуктів.

Отже, створення Web-додатку для організації автоматизації процесів ресторану має низку переваг, які дозволять забезпечити клієнтів якісним та швидким сервісом, покращити роботу персоналу та збільшити прибуток ресторану. Важливо зазначити, що така система може бути легко масштабована, що дозволить застосовувати її в різних типах ресторанів та кафе.

Крім того, створення Web-додатку для організації автоматизації процесів ресторану є актуальною та перспективною темою для дослідження та розробки в галузі інформаційних технологій та сервісів. Результати розробки можуть бути корисними для більшості ресторанів та кафе, що мають потребу в автоматизації процесів обслуговування клієнтів та оптимізації роботи персоналу. Крім того, така робота може сприяти вдосконаленню вже існуючих систем та створенню нових рішень для покращення обслуговування в сфері громадського харчування.

Таким чином, можна зробити висновок, що створення Web-додатку для організації автоматизації процесів ресторану є доцільним та перспективним кроком, який дозволить забезпечити більш якісне та швидке обслуговування клієнтів, оптимізувати роботу персоналу та збільшити прибуток ресторану. При реалізації проекту необхідно враховувати особливості роботи ресторану, потреби клієнтів та персоналу, а також забезпечувати надійну та безпечну роботу системи.

#### 1.4 Постановка завдання

Метою даного дипломного проєкту є розробка Web-додатку для автоматизації процесів ресторану з метою полегшення роботи персоналу та забезпечення більш швидкого та якісного обслуговування клієнтів.

Для досягнення цієї мети було поставлено наступні завдання:

- 1) Розробити структуру та функціонал Web-додатку для організації процесів ресторану.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дат		

2) Розробити інтерфейс користувача Web-додатку з урахуванням потреб клієнтів та персоналу.

3) Розробити базу даних, яка забезпечить зберігання та обробку інформації про замовлення, меню, склад продуктів, персонал та інші дані, необхідні для роботи ресторану.

4) Реалізувати функціонал додатку, який дозволить клієнтам здійснювати замовлення онлайн, отримувати інформацію про меню та акції ресторану.

5) Реалізувати функціонал додатку, який дозволить персоналу здійснювати облік замовлень, станів замовлень, складу продуктів, роботи персоналу та інші функції, необхідні для ефективної роботи ресторану.

6) Забезпечити безпеку та надійність додатку, а також захист від несанкціонованого доступу до даних.

7) Провести тестування додатку на ефективність та зручність використання.

Результатом роботи буде розроблений та протестований Web-додаток, який забезпечить автоматизацію процесів ресторану та підвищення якості обслуговування клієнтів та ефективності роботи персоналу.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дат		

## 1.5 Дослідження технологій реалізації

Для розробки Web-додатка для організації автоматизації процесів ресторану, необхідно обрати підходящі технології. У цьому розділі дослідимо технології реалізації, які можуть бути використані для розробки додатку.

Першими на увазі є мови програмування. Для Web-розробки найпопулярнішими мовами програмування є JavaScript, Python та PHP. JavaScript є основною мовою для фронтенд-розробки, тобто для розробки користувацького інтерфейсу. Python та PHP є популярними мовами для бекенд-розробки, тобто для розробки серверної логіки.

Крім того, необхідно вибрати фреймворки та бібліотеки для розробки додатку. Фреймворки та бібліотеки забезпечують розробникам готові рішення для розробки додатків. Деякі з популярних фреймворків для Web-розробки включають React, Angular та Vue.js.

Нарешті, необхідно вибрати базу даних для зберігання даних. Для Web-розробки найпопулярнішими базами даних є MySQL, PostgreSQL та MongoDB. Кожна база даних має свої переваги та недоліки, тому вибір бази даних залежить від конкретних потреб проекту.

JavaScript є однією з найбільш популярних мов програмування для Web-розробки. JavaScript використовується для розробки фронтенду, а також для розробки серверної логіки з використанням Node.js. Однією з переваг JavaScript є велика кількість бібліотек та фреймворків, які забезпечують розробникам готові рішення для створення додатків.

React є одним з найпопулярніших фреймворків для фронтенд-розробки на базі JavaScript. React забезпечує розробникам можливість створювати перевикористовувані компоненти, що дозволяє розробляти додатки більш ефективно та швидко, також надає можливість розробляти додатки за допомогою TypeScript.

MySQL є однією з найпопулярніших реляційних баз даних для Web-розробки. MySQL має високу продуктивність та надійність, що робить його популярним вибором для багатьох проектів. PostgreSQL є ще однією

					ЛП.КН 23.505.08.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		22

реляційною базою даних, яка надає багато можливостей для складних проектів. MongoDB є нереляційною базою даних, яка надає можливість зберігати структуровані та неструктуровані дані.

Іншою популярною мовою програмування для Web-розробки є Python. Python використовується для бекенд-розробки, а також для наукових та математичних розрахунків.

Node.js є популярним серед розробників для створення серверних додатків. Node.js дозволяє розробникам використовувати JavaScript для розробки бекенду, що полегшує процес розробки та підтримки додатків. Express.js є популярним фреймворком для створення додатків на базі Node.js. Express.js надає розробникам більш прості та легкі рішення для створення серверних додатків.

Для розробки інтерфейсів користувача (UI) можна використовувати різні інструменти та бібліотеки. Bootstrap є одним з найпопулярніших інструментів для створення UI. Bootstrap надає розробникам готові рішення для стилізації та розміщення елементів на сторінці. Material Design є ще однією популярною бібліотекою для створення UI. Material Design надає розробникам можливість створювати елементи з високою якістю дизайну.

У розробці Web-додатків важливо також враховувати питання безпеки. Захист від хакерських атак та зловмисного програмного забезпечення є ключовим завданням для будь-якого Web-додатка. Для забезпечення безпеки додатка можна використовувати різні технології та інструменти, наприклад, SSL-шифрування, валідацію введених даних, мережеві фільтри та багато іншого.

Також важливо розглянути питання масштабованості додатка. При збільшенні кількості користувачів або обсягу.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дат		



## 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Поняття та важливість проектування користувацького інтерфейсу

Проектування користувацького інтерфейсу передбачає глибоке розуміння потреб користувачів, яке досягається за допомогою досліджень, упорядкування інформації, візуального дизайну та багатьох інших методів. Головна мета проектування користувацького інтерфейсу – задоволення потреб користувачів. Під час цього процесу користувачі стають центром уваги, а проектування та розробка проводяться в рамках ітеративного циклу досліджень, проектування та оцінки.

Більшість користувачів віддають перевагу програмному забезпеченню з привабливим зовнішнім виглядом та зручним інтуїтивним інтерфейсом. Для досягнення цього необхідно поєднати дизайн досвіду користувача (UX design) та дизайн інтерфейсу користувача (UI design). Відмінний User Interface дизайн залучить увагу користувача до програми, а хороший досвід роботи змусить його повернутися до використання продукту знову і знову.

Розробка користувацького інтерфейсу (UI) – це процес поліпшення вигляду та взаємодії веб- або мобільних додатків з користувачами. У користувацький інтерфейс входять текстові блоки, кнопки, текстові поля, форми, зображення та інші візуальні елементи, які користувач бачить та з якими взаємодіє під час використання програми. Користувацький інтерфейс також містить макет екранів, переходів між сторінками, анімацію та кожен окрему взаємодію з користувачем. Основна мета розробки користувацького інтерфейсу полягає у створенні зручного та привабливого середовища для користувача.

UX дизайн – це процес поліпшення загального користувацького досвіду взаємодії з веб-сайтом або додатком, який має на меті задоволення потреб споживачів та надання їм додаткової цінності. UX дизайн включає в себе такі елементи як інтерактивний дизайн, інформаційну архітектуру, візуальний дизайн, юзабіліті та взаємодію між продуктом та користувачем. Ці елементи

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дат		

разом створюють зручний та задоволений користувачів досвід взаємодії з продуктом.

При використанні UX підходу в розробці продукту, передбачається наявність циклу, що складається з кількох етапів. Починається він з дослідження, яке передбачає проведення початкового аналізу користувачів з метою з'ясування їх потреб, очікувань, цілей, обмежень та інших аспектів, які необхідні для розуміння цільової аудиторії. Наступним етапом є дизайн, на якому базується статистика, отримана під час дослідження користувачів, для генерації ідей та створення початкового прототипу продукту. Після цього, проводиться етап оцінки, який включає збір відгуків користувачів на ранніх етапах розробки, щоб покращити продукт та забезпечити максимально комфортний досвід взаємодії. В результаті, цикл повторюється до тих пір, поки не буде досягнута оптимальна якість продукту та задоволення користувачів.

Ключові моменти UX дизайну це:

- інформаційна архітектура;
- дизайн взаємодії;
- юзабіліті;
- вайфрейми;
- візуальний дизайн.

В результаті цього проекту була створена інформаційна система, яка дозволяє автоматизувати деякі функції, пов'язані з додаванням нових працівників, замовленням їжі, бронюванням столиків та створенням меню та звітів кухарем. Систему будуть використовувати працівники фірми, кухар та адміністратор.

Інформаційна архітектура (Information Architecture) має на меті створення оптимальної інформаційної структури додатку або веб-сайту, щоб задовольнити бізнес-стратегії та забезпечити користувачам зручну навігацію. Це включає використання різноманітних методів для створення ефективного та зручного меню верхнього рівня та інших елементів навігації.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дат		

Дизайн взаємодії (Interaction Design) охоплює створення концептуального дизайну, який забезпечує зручну та ефективну взаємодію між користувачами та продуктом або додатком. Це включає використання різних елементів, таких як колір, шрифт, графіка, зображення та інші, щоб забезпечити максимальну зручність та задоволення від взаємодії з продуктом.

Юзабіліті означає, наскільки зручним є використання продукту для користувача. Вона вирішує, чи можуть користувачі знайти потрібну інформацію при першому відвідуванні програми або веб-сайту, а також наскільки зручною є програма / веб-сайт для навігації та використання інших функцій.

Вайрфрейм (wireframe) – це прототип програми, який використовується для перевірки функціональності, вигляду та зручності використання програми перед її запуском. Це економічний спосіб перевірити, чи відповідає додаток поставленим цілям.

Візуальний дизайн програми або веб-сайту повинен відображати бренд компанії. Це включає вибір текстових шрифтів, кольорової палітри, зображень та іконок, а також визначення впливу зовнішнього вигляду програми на її взаємодію з користувачами.

## 2.2 Формалізація вимог

В результаті цього проекту була створена інформаційна система, яка дозволяє автоматизувати деякі функції, пов'язані з додаванням нових працівників, замовленням їжі, бронюванням столиків та створенням меню та звітів кухарем. Систему будуть використовувати працівники фірми, кухар та адміністратор.

На рисунку 2.1 зображено бізнес-процеси, які будуть автоматизовані.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дат		

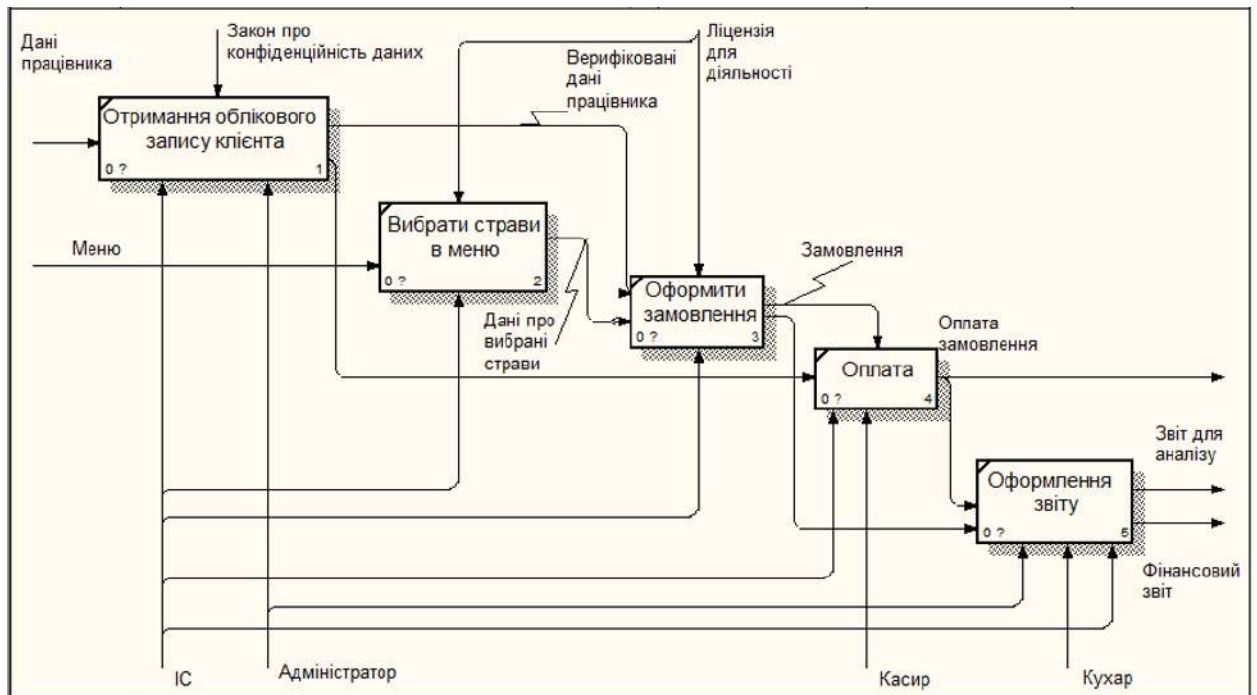


Рисунок 2.1 – Діаграма оформлення замовлення

На рисунку 2.2 було зображено основні елементи узагальненої умовно замкненої моделі.

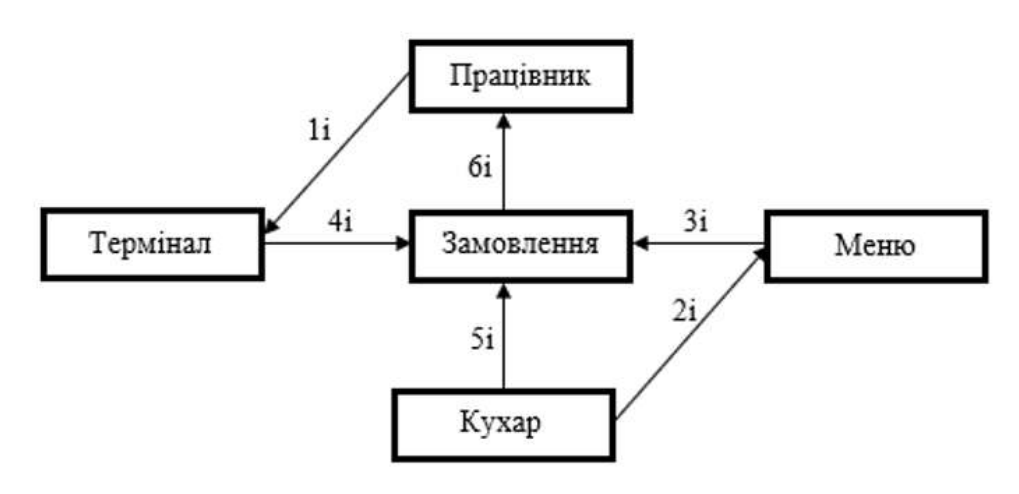


Рисунок 2.2 – Узагальнена умовно замкнута модель інформаційної системи

Значення інформаційних управляючих зв'язків даної схеми (рис. 2.1):

- 1i – внесення персональних даних клієнта;
- 2i – створення меню на певний день;
- 3i – відображення меню у формі замовлення;
- 4i – здійснення замовлення працівником;

Змн.	Арк.	№ докум.	Підпис	Дат

5і – виконання замовлення;

6і – видача виконаного замовлення.

### 2.3 Побудова дерева функцій та проектування структури системи

Робота системи базується на швидкій обробці та оновленні інформації, а також на безпомилковому обслуговуванні клієнтів. Для забезпечення ефективності та стабільності роботи системи, адміністратор повинен ретельно вести облік всіх процесів, які відбуваються в ній. Це дасть змогу вчасно виявляти проблеми, які виникають, та швидко їх усувати.

Окрім того, для оптимізації роботи системи необхідно проводити аналіз числових показників, що дозволить виявити тенденції та прогнозувати можливі проблеми. Для цього можна використовувати спеціальні інструменти та програмні засоби, які допоможуть зібрати та обробити необхідні дані.

Крім того, важливо пам'ятати про потребу в постійному вдосконаленні та оновленні системи. Для цього можна використовувати нові технології та розробки, а також слід розглядати пропозиції та відгуки від користувачів системи, що дозволить покращити якість обслуговування та забезпечити задоволення клієнтів.

На рисунку 2.3 зображено дерево функцій системи.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дат		

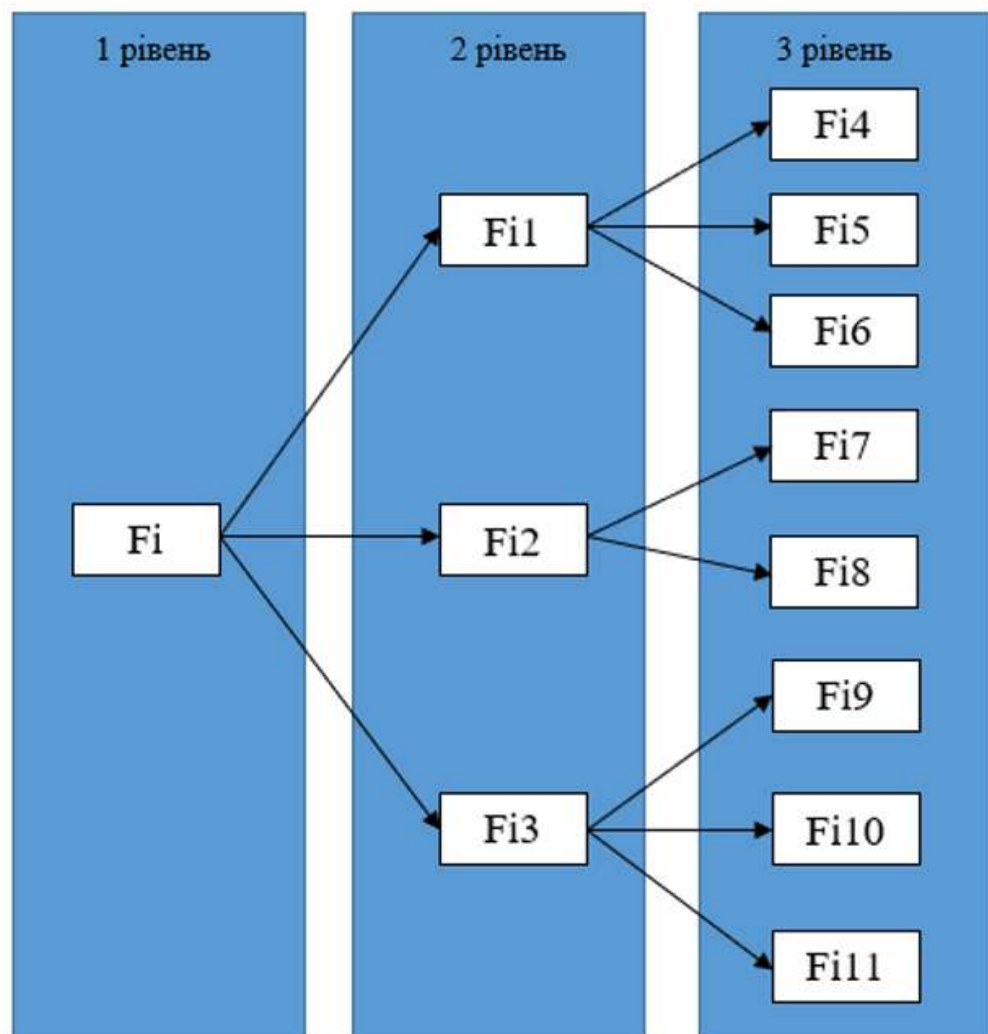


Рисунок 2.3 – Дерево функцій інформаційної системи

Fi1 – облік замовлень:

Fi4 – облік надходжень;

Fi5 – облік наявних;

Fi6 – облік виконаних.

Fi2 – інформація про доходи:

Fi7 – вартість усіх виконаних замовлень;

Fi8 – доходи за певний період.

Fi3 – облік клієнтів:

Fi9 – нові працівники (не зареєстровані);

Fi10 – офіційні працівники (зареєстровані);

Fi11 – отримані працівниками замовлення.

Змн.	Арк.	№ докум.	Підпис	Дат

Після визначення функціоналу та аналізу спроектовано структуру інформаційної системи, яка зображена на рисунку 2.4.



Рисунок 2.4 – Схема компонентів системи

## 2.4 Проектування бази даних

Під час аналізу предметної області було виявлено наступні інформаційні сутності: «Авторизація», «Працівник», «Замовлення», «Меню», «Звіт», «Керівник» та «Касир». Для подальшого проектування було проведено додатковий аналіз, щоб встановити зв'язки між цими сутностями.

Під час аналізу було виявлено, що між сутностями «Авторизація» та «Працівник» існує зв'язок «багато до одного». Це означає, що один працівник може авторизуватись в системі декілька разів для подальшої роботи.

В компанії потрібен касир для ефективного обробки замовлень працівників. Між сутностями «Касир» та «Замовлення» існує зв'язок «один до

багатьох», оскільки один касир може обслуговувати багато замовлень. У поточній системі передбачено тільки одного касира, проте у майбутньому, зі зростанням компанії та збільшенням кількості працівників, може знадобитися ще один касир для більш швидкої обробки замовлень.

Також було встановлено зв'язок «один до багатьох» між сутностями «Працівник» та «Замовлення» через проміжну сутність «Меню». Для доступу до меню та здійснення замовлення потрібний неідентифікований зв'язок.

Однією з важливих функцій запланованої інформаційної системи є генерація звітів, тому було включено сутність "Звіт", яка має зв'язок типу "один до багатьох" з сутністю "Замовлення". Щодо обліку харчування в компанії, було створено зв'язок "багато до одного" між сутностями "Звіт" та "Керівник". Для контролю над всіма операціями в системі необхідний аналіз та виправлення всіх потрібних недоліків.

Наступним етапом проектування є створення логічної моделі реляційної бази даних.

Таблиця 2.1 menus

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	ObjectId	-	-	-	-	P
category	String	-	-	-	-	-
description	String	-	-	-	-	-
image	String	-	-	-	-	-
like	Int32	-	-	-	-	-
name	String	-	-	-	-	-
percent_alcohol	Double	-	-	-	-	-
price	Int32	-	-	-	-	-
type	String	-	-	-	-	-
weight	Int32	-	-	-	0	-



Таблиця 2.2 Orders

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	ObjectId	-	-	-	-	Р
createdAt	Date	-	-	-	-	-
order	Array	-	-	-	-	-
id	String	-	-	-	-	-
quantity	Int32	-	-	-	-	-
payment	String	-	-	-	-	-
status	Int32	-	-	-	0	-
tableId	Int32	-	-	-	-	-
totalPrice	Int32	-	-	-	-	-
updatedAt	Date	-	-	-	-	-

Наступним етапом проектування є створення логічної моделі для столів.

Таблиця 2.3 tables

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	ObjectId	-	-	-	-	Р
tableNumber	Int32	-	-	-	-	-
url	String	-	-	-	-	-

Таблиця 2.4 users

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	ObjectId	-	-	-	-	Р
email	String	-	-	-	-	-
name	String	-	-	-	-	-
password	String	-	-	-	-	-
role	String	-	-	-	-	-

## 2.5 Проектування користувацького інтерфейсу

Потрібно розділити функціонал на декілька ролей, які зображено на рисунку 2.5.



Рисунок 2.5 – Діаграма прецедентів Шеф кухаря

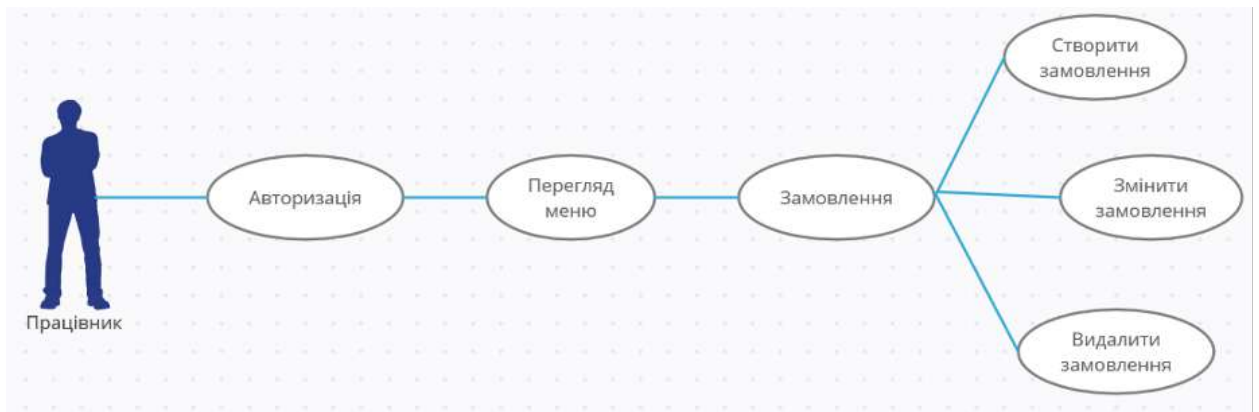


Рисунок 2.6 – Діаграма прецедентів Авторизованого користувача

Функції кухарів:

- авторизація;
- створення меню;
- створення звіту.

Функції зареєстрованого працівника:

- авторизація;
- перегляд меню;
- здійснення замовлення;
- керуванням замовлення.

Змн.	Арк.	№ докум.	Підпис	Дат

## 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 3.1 Опис технологій та засобів реалізації

Для виконання всі поставлених завдань було вирішено використати СУБД MongoDB. За допомогою цього веб додатка користувач має можливість адмініструвати базу даних. Ця база даних має певні недоліки, а саме:

- Великі обсяги пам'яті. MongoDB використовує більше пам'яті, ніж реляційні бази даних, оскільки документ-орієнтований підхід вимагає зберігання більшої кількості метаданих.

- Немає транзакцій. Mongo не підтримує транзакції, що може бути недостатньою умовою для критичних додатків, які потребують гарантованого цілісності даних.

- Обмежена підтримка JOIN. MongoDB не підтримує JOIN-запити на кількох колекціях, що може ускладнювати вибірку та обробку даних в деяких випадках.

Але компенсує їх великою кількістю переваг, а саме:

- Гнучкість. Mongo дозволяє додавати та видаляти поля в документах, не змінюючи схему бази даних, що дає можливість додавати нові дані без перепроєктування бази даних.

- Швидкість. MongoDB дозволяє швидше зберігати та вибирати дані в порівнянні з реляційними базами даних, оскільки вона не використовує SQL та таблиці, а зберігає дані в документах формату JSON.

- Географічна реплікація. Mongo дозволяє реплікувати дані на кілька серверів, розташованих у різних географічних регіонах, що дозволяє зменшити час відгуку та підвищити доступність системи.

За вимогами проекту, мова програмування JavaScript є найбільш оптимальним варіантом для розробки. Це забезпечить можливість реєстрації, авторизації та створення користувацького інтерфейсу для взаємодії з базою даних.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дат		

Однією з основних вимог до системи є її максимальна доступність, тому було прийнято рішення розробити інтерфейс, який зможе адаптуватися під будь-який пристрій користувача. Для досягнення цієї мети ми обрали фреймворк Next.js. Його основними перевагами є швидкість розробки та можливість автоматичної оптимізації веб-сторінок для кращої продуктивності.

Крім того, Next.js дозволяє легко взаємодіяти з базою даних та іншими додатками. Він має багатий набір готових компонентів та модулів, що зробить розробку більш ефективною та простою.

Незважаючи на всі переваги Next.js, варто розглянути його недоліки. Наприклад, при великому обсязі даних на сторінці може знижуватися продуктивність, що може вплинути на користувацький досвід. Також, використання фреймворку може змусити розробника дотримуватися певних правил та обмежень, що може вплинути на творчу свободу.

У загальному, обираючи фреймворк для розробки веб-додатків, потрібно ретельно зважити всі його переваги та недоліки, а також зважити на конкретні потреби проекту. Тільки так можна створити ефективну та надійну систему, яка зможе задовольнити потреби користувачів та власників проекту.

В якості IDE було обрано VSCode і завантажено з офіційного сайту, показаного на рисунку 3.1.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дат		

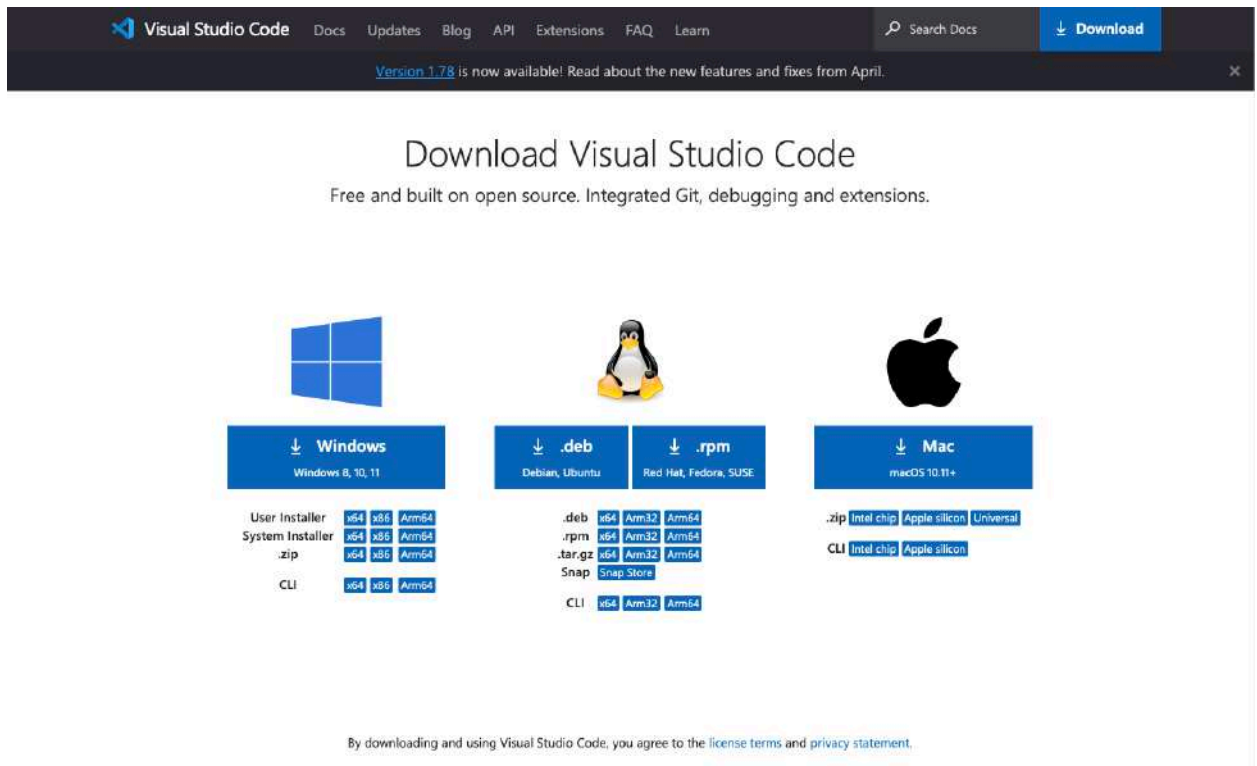


Рисунок 3.1 – Офіційний сайт завантаження VSCode

Далі було завантажено менеджер пакетів з офіційного сайту NodeJs, як на рисунку 3.2.



Рисунок 3.2 – Офіційний сайт NodeJs

Далі був ініціалізований проект на NextJs (рис. 3.3).

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дат		

```

stefan — -zsh — 86x29
Last login: Tue May  9 16:36:27 on ttys004
stefan@Stefans-MacBook-Air ~ % node -v
v18.16.0
stefan@Stefans-MacBook-Air ~ % npx create-next-app@latest

```

Рисунок 3.3 – Ініціалізація проекту н NextJs

### 3.2 Реалізація основних функцій інформаційної системи

Відповідно до фізичної моделі було розроблено запити на створення таблиць (лістинг A1 в додатку A), унікальних полів, індексів та ключів (лістинг A2 в додатку A), а також зберезувальні процедури, які дозволяють додавати, змінювати та видаляти дані з певних таблиць шляхом виконання додаткових маніпуляцій з вхідними параметрами.

Необхідні змінні екрануються, передається прототип запити та значення параметрів на сервер.

Для таблиці «Керівник» було розроблено три зберезувальні процедури (див. лістинг A3 в додатку A):

`create_lead` – приймає 3 параметри: ім'я керівника, електронну пошту та пароль. Після цього перевіряється наявність керівника, у якого хоча б одне з полів співпадає з новим записом. Якщо збіг знайдено, то запис не буде додано.

Якщо ж збігів не виявлено, то буде підготовлено запит до сервера, передано його разом з екранованими параметрами і виконано запит.

update\_lead – приймає 4 параметри: ідентифікатор запису, ім'я користувача, пароль та електронну пошту. Після цього перевіряється можливість зміни даних керівника на нові. Шукаються збіги з іншими записами. Якщо збіг знайдено, то запис не буде оновлено, в іншому випадку на сервер передається запит разом з екранованими параметрами. Після цього параметри підставляються в запит і виконується оновлення запису.

delete\_lead – приймає 1 параметр: ідентифікатор запису і виконує запит на видалення запису з таблиці «Керівник» з відповідним ідентифікатором.

Для сутності «Меню» було створено три зберезувальні процедури: create\_menu, update\_menu та delete\_menu.

create\_menu приймає 8 параметрів: ідентифікатор страви, категорію, фотографію, кількість вподобань, ціну, вагу та тип страви.

update\_menu приймає ті ж параметри, що і create\_menu, а також ідентифікатор запису, який необхідно змінити. Далі виконується перевірка можливості внесення змін, тобто порівнюються значення атрибутів наявних записів із значеннями, які передаються у процедуру оновлення інформації. Якщо збігів не знайдено, то запис з ідентифікатором, переданим у зберезувальну процедуру, оновлюється.

delete\_menu приймає 1 параметр – ідентифікатор запису. Оскільки вхідний параметр приводиться до цілочисельного значення, екранування не виконується, через відсутність потреби. Перед виконанням запиту на видалення виконується перевірка наявності запису з переданим ідентифікатором, що додано з метою подальшого нарощування функціоналу.

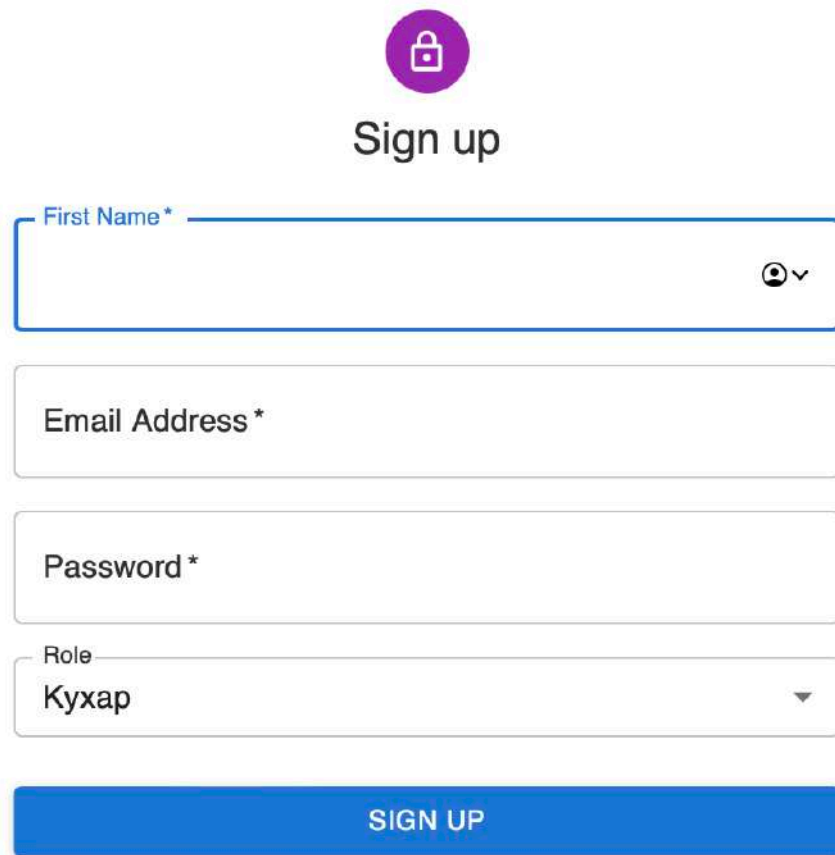
### 3.3 Реалізація користувацького інтерфейсу системи

Було впроваджено систему реєстрації для обмеження прав доступу користувачів, яка вимагає мінімального обсягу особистої інформації від

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дат		



користувача. Процес реєстрації включає заповнення форми (лістинг Б1, додаток Б), що складається з набору полів (показано на рис. 3.4).



The image shows a 'Sign up' form. At the top is a purple circular icon with a white padlock. Below it is the text 'Sign up'. The form consists of four input fields: 'First Name \*' (with a clear button), 'Email Address \*', 'Password \*', and 'Role' (a dropdown menu with 'Кухар' selected). At the bottom is a blue button labeled 'SIGN UP'.

Рисунок 3.4 – Форма реєстрації

Після успішної реєстрації, користувач повинен ввести свою електронну пошту та пароль, які були вказані при реєстрації, у форму авторизації (лістинг Б2, додаток Б) (рис. 3.5). Після цього система здійснить запит до бази даних для перевірки наявності відповідного облікового запису.



## Sign in

Email Address \*

admin@gmail.com

Password \*

.....

SIGN IN

Рисунок 3.5 – Форма авторизації

Після успішної авторизації, користувач отримує доступ до різних функцій системи, які включають роботу з адміністраторами, кухарами, барменами, меню, активними столами, керування станом замовлення та підсумковими даними, що обчислюються на основі таблиць бази даних.

Вкладка для роботи з керівниками містить 3 функції (лістинг Б3, додаток Б): «Додати», «Змінити» і «Видалити».

Функція "Додати" дозволяє користувачу додавати новий персонал. Для цього потрібно заповнити форму (лістинг Б4, додаток Б), яка зображена на рисунку 3.4, і надіслати дані, натиснувши кнопку «Sign Up».

Після надсилання даних, новий користувач отримує певні привілеї в програмі та з'являється в списку користувачів, як показано на рисунку 3.6.

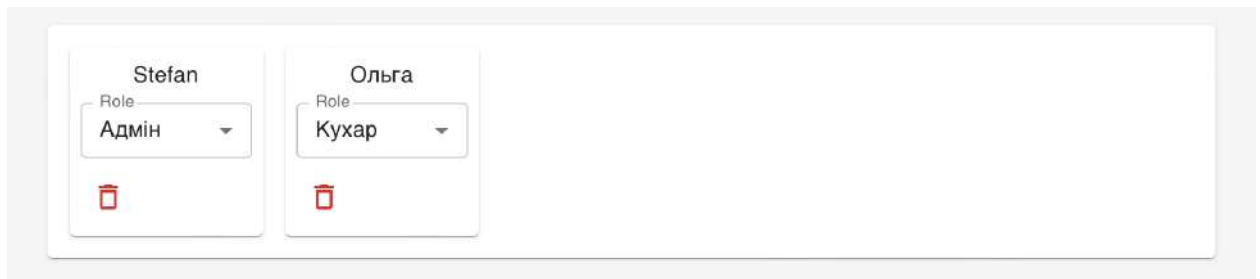


Рисунок 3.6 – Таблиця персоналу

Також є можливість «Змінити» роль користувача в системі або видалити його (рис. 3.7).

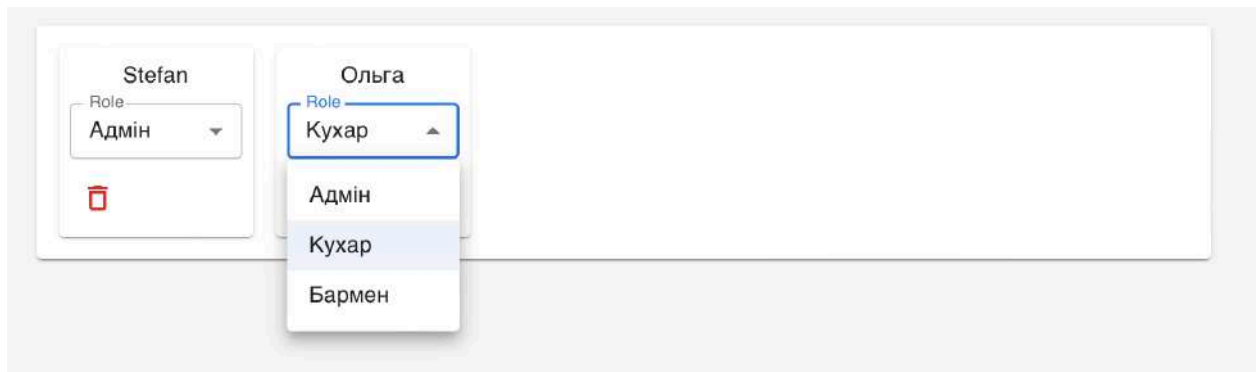


Рисунок 3.6 – Кнопки зміни і видалення користувачів

Вкладки для роботи з меню вимагає права доступу адміністратора.

Вкладка для роботи з меню вимагає права доступу адміністратора та містить 3 функції (лістинг Б5, додаток Б): «Додати», «Змінити» і «Видалити».

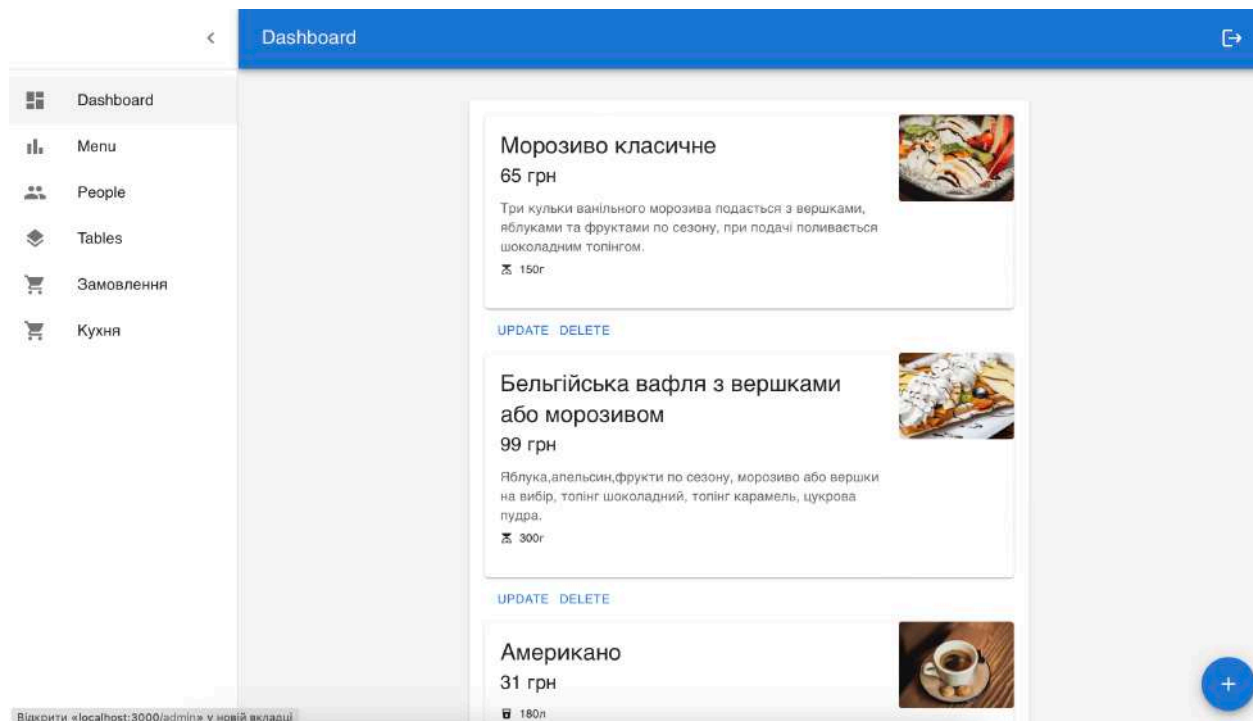


Рисунок 3.7 –Список страв у меню для адміністраторів

Вкладка "Столи" включає можливість генерації QR-кодів для кожного столу з метою отримання ідентифікованого замовлення. Вона включає чотири функції (лістинг Б6, додаток Б): "Скопіювати", "Створити", "Роздрукувати" і "Видалити".



Рисунок 3.8 – Список столів

Після сканування QR-коду відкривається веб-сайт з меню для замовлення страв (рис 3.9). На цьому сайті можна вибрати бажану кількість страв та додати їх у кошик. Також, можна виразити свої уподобання, натиснувши на кнопку "Вподобайка" чи на подібний елемент інтерфейсу.

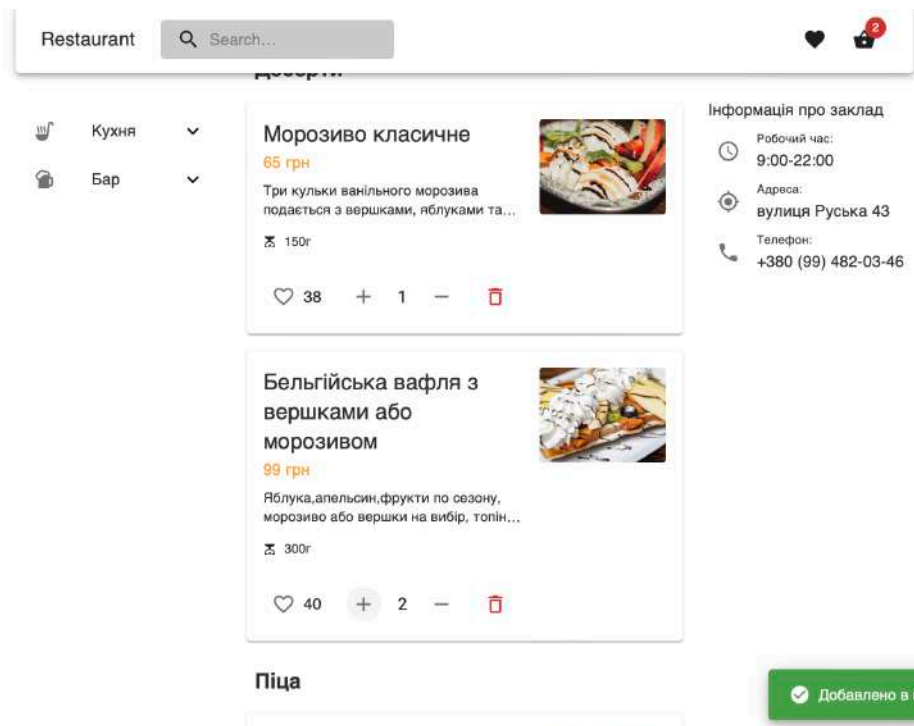


Рисунок 3.9 – Меню для замовлення

В розділі "Корзина" ви зможете переглянути загальну суму вашої покупки, яка включає вартість усіх вибраних вами страв. Також, тут ви знайдете різні варіанти оплати для вашого замовлення (рис 3.10).

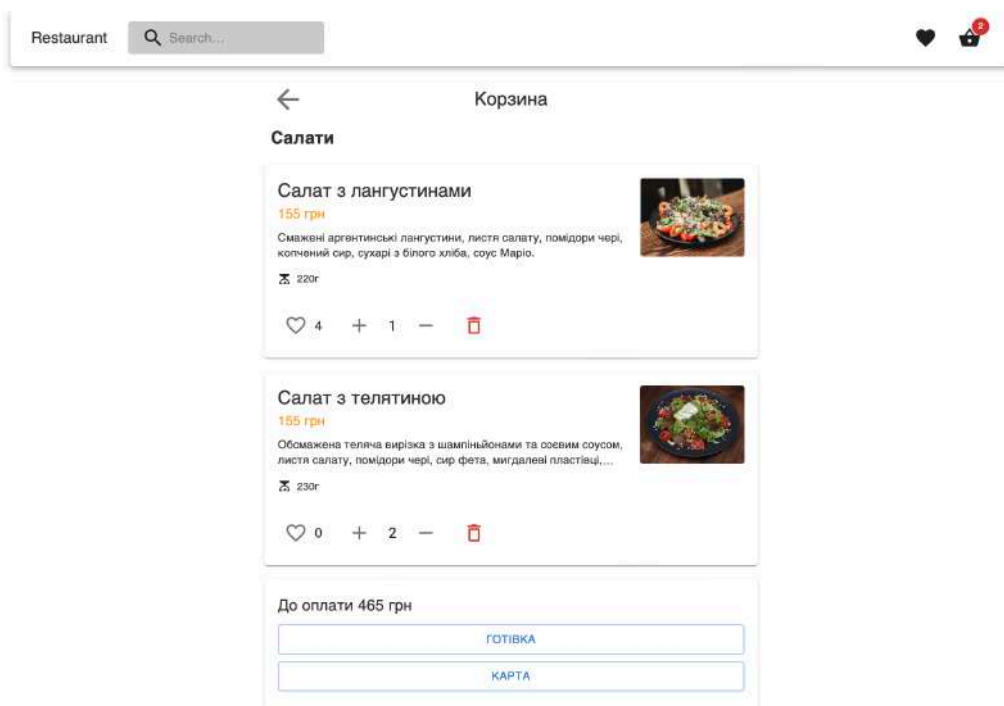


Рисунок 3.10 – Корзина

					ЛП.КН 23.505.08.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		45

Після вибору способу оплати, дані замовлення автоматично передаються до адміністративної панелі, де кухарі та інші працівники можуть переглянути деталі замовлення. Це дозволяє кухарям побачити список страв, що були замовлені, а також кількість кожної страви, які були вказані клієнтом (рис 3.11).

Стіл	Страви	Стан	Контроль
9	1 - Морозиво класичне 2 - Бельгійська вафля з вершками або морозивом		<input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/>
9	1 - Морозиво класичне		<input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/>
2	11 - Морозиво класичне 4 - Фітнес		<input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/>
2	1 - Бельгійська вафля з вершками або морозивом 1 - Салюмі 3 - Американо 3 - Файні Наливки		<input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/>
2	15 - Морозиво класичне		<input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/>

Рисунок 3.11 – Панель для контролю станом замовлення

Також доступний альтернативний вигляд для звітності замовлень, який можна переглянути на рисунку 3.12. Цей вигляд надає більш комплексну інформацію про замовлення, що були здійснені.

Стіл	Страви	Стан	Оплата	Ціна
9	1 - Морозиво класичне 2 - Бельгійська вафля з вершками або морозивом	Очікується		263 грн
9	1 - Морозиво класичне	Готово		65 грн
2	11 - Морозиво класичне 4 - Фітнес	Оплачено		1095 грн
2	1 - Бельгійська вафля з вершками або морозивом 1 - Салюмі 3 - Американо 3 - Файні Наливки	Оплачено		722 грн
2	15 - Морозиво класичне	Оплачено		975 грн

Рисунок 3.11 – Панель для звітності замовлень

На вкладці «Статистика» також доступна статистика, яка надає корисну інформацію про фінансові показники ресторану. Ця статистика включає розрахунок загальної суми, яку заробив ресторан протягом певного періоду. Вона дає вам змогу відстежувати фінансову продуктивність ресторану та оцінювати його дохідність. Також, сайт також надає графік, який відображає обсяг продажу товарів за кожен місяць. Цей графік надає вам візуальне уявлення про тенденції у продажах та дозволяє вам аналізувати, як змінюється попит на різні страви та категорії товарів з часом (рис 3.12).

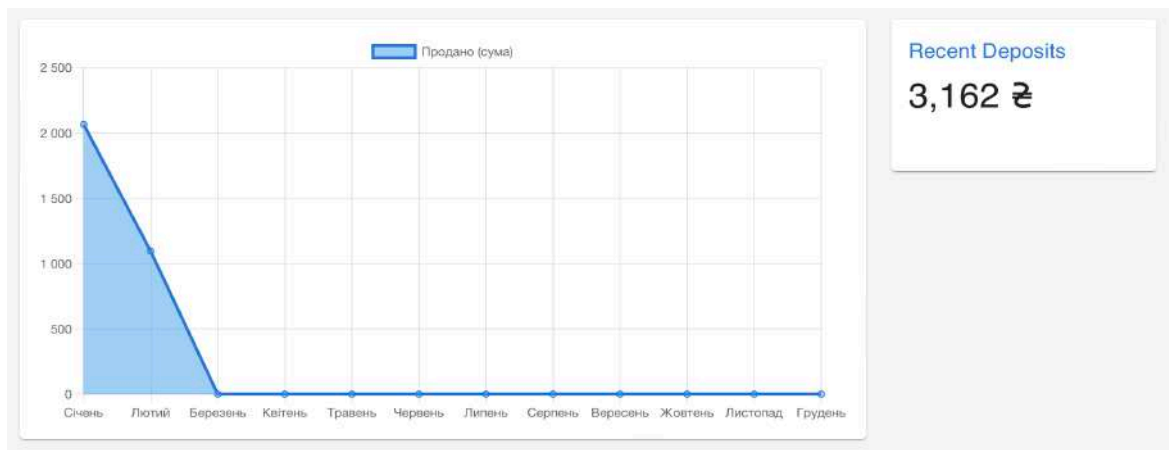


Рисунок 3.12 – Статистика

Для забезпечення належного відображення всіх елементів на екранах різних розмірів було застосовано медіа-запити. Медіа-запити – це техніка веб-розробки, яка дозволяє адаптувати стилі та вигляд веб-сторінки до різних пристроїв та розмірів екранів.



## 4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

### 4.1 Аналіз ринку

У сучасній динамічній інформаційній сфері, де постійно з'являються нові технології та рішення, конкуренція в створенні програм і систем дійсно є високою. Компанії постійно змагаються за увагу та вподобання користувачів, шукаючи способи відрізнитися від конкурентів та надати унікальний функціонал.

Аналіз ринку є важливим етапом в розробці будь-якого проекту. Він дозволяє визначити потреби та очікування користувачів, ідентифікувати конкурентів і їхні переваги, а також визначити актуальні теми і тенденції для створення продукту, який буде відповідати потребам ринку.

Одним з недоліків багатьох систем і програм є обмежений пробний період і необхідність оплати для подальшого використання. Проте, це залежить від бізнес-моделі та стратегії розробників. Деякі системи пропонують безкоштовний базовий функціонал або безстроковий пробний період, а інші пропонують платні пакети з розширеними можливостями.

У програмній частині системи використовуються програмні засоби, які забезпечують зберігання та зміну даних, а також взаємодію користувача з системою. Дизайн та функціонал системи розроблені з метою забезпечити легкість освоєння та зручну роботу для користувачів. Така система може бути цікавою для різних підприємств, які шукають автоматизацію процесів організації харчування. Важливим аспектом є можливість користуватись системою з будь-якої точки світу, якщо є доступ до Інтернету. Це дозволяє забезпечити зручність та гнучкість в роботі з системою незалежно від місця знаходження.

У життєвому циклі продуктів важливо мати потенціал для майбутнього розвитку. Якщо продукт не оновлюється та його функціонал не відповідає зростаючим потребам користувачів, він може втратити актуальність і стати застарілим. Розробники повинні постійно вдосконалювати свій продукт,

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дат		

враховуючи зміни на ринку та потреби користувачів, щоб забезпечити його успішну експлуатацію в майбутньому.

#### 4.2 Розрахунок витрат на проєктування інформаційної системи

В таблиці Г1 додатку Г представлені загальні витрати фінансової частини, пов'язані з розробкою системи. Заробітна плата працівників, які брали участь у розробці проєкту, розраховувалась на основі витраченого робочого часу та кількості днів. У процесі розробки інформаційної системи брали участь керівник ДП, який відповідав за контроль всіх працівників та просування роботи, тестувальник, який після завершення розробки проводив тестування функціональності системи, всіх функцій та процедур, фронтенд-розробник та бекенд-розробник, які відповідали за розробку інтерфейсу користувача та програмно-апаратного забезпечення, відповідно. Розрахунки заробітної плати були здійснені згідно з таблицею Г2 додатку Г і представлені у таблицях Г3-Г7 додатку Г.

#### 4.3 Обґунтування необхідності розробки

Інформаційна система для організації харчування в приватному ІТ-підприємстві є спеціалізованим рішенням, яке спрямоване на полегшення роботи персоналу кафетерію та автоматизацію багатьох процесів, що раніше вимагали людського втручання. Основна мета розробки цієї системи – покращення ефективності роботи підприємства та зниження витрат.

Шляхом автоматизації процесів харчування, система спрощує та прискорює взаємодію між персоналом кафетерію та працівниками. Наприклад, система може включати в себе замовлення страв онлайн, автоматичний розрахунок суми замовлення та вибір варіанту оплати. Це дозволяє зменшити час, який потрібен на обробку замовлень та операції з готівкою. Автоматизація процесів харчування також забезпечує можливість зберігати та аналізувати дані про замовлення, продажі та інші важливі параметри.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дат		

## ВИСНОВКИ

Розробка інформаційної системи «Web-додаток для організації автоматизації процесів ресторану» є актуальною і вдалою, оскільки надає можливість автоматизувати процеси роботи працівників на кухні під час приготування страв.

Під час проектування системи були створені діаграми бізнес-процесів, дерево функцій та схема компонентів системи. Проведений детальний аналіз аналогічних програм допоміг у розробці дипломного проєкту.

При виборі засобів для розробки системи було використано мову програмування JavaScript. У базі даних зберігаються інформація про працівників, створені меню та замовлення, а також перелік страв для вибору. Інформаційна система надає користувачам можливість отримати доступ до особистого кабінету, а кухар може створювати меню для цих замовлень та додавати нові страви за потреби та переглядати звіти.

Цей проєкт містить набір інструментів, які дозволяють виконувати всі необхідні функції, а також має потенціал для розширення функціоналу у майбутньому. Наприклад, можливість доступу за допомогою чіп-карти, списування коштів з банківської карти та безконтактна оплата.

Отже, в результаті була розроблена повноцінна інформаційна система, яка успішно виконує всі поставлені завдання і функції, а також має потенціал для подальшого розширення та розвитку.

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дат		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. VScode – редактор коду. *Wikipedia*: вебсайт. URL: [https://uk.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://uk.wikipedia.org/wiki/Visual_Studio_Code) (дата звернення : 10.09.2022).
2. Розробка вебсайті на Next.js. *Brander*: вебсайт. URL: <https://brander.ua/technologies/nextjs> (дата звернення : 08.10.2022).
3. ChoiceQR меню – сервіс онлайн меню. *Choice*: вебсайт. URL: <https://choiceqr.com/uk/> (дата звернення 15.01.2023).
4. QR меню – електронне QR меню. *Oddmenu*: вебсайт. URL: <https://oddmenu.com/uk> (дата звернення : 18.01.2023).
5. Євразія – меню в ресторані. *Evrasia*: вебсайт. URL: <https://evrasia.in.ua/menu> (дата звернення : 25.01.2023).
6. Creately – Ideate, Plan amp Run Projects on a Smart Visual Canvas Creately. Creately: вебсайт. URL: <https://creately.com/> (дата звернення 23.02.2023).
7. Дуніковський С.Б. *Дослідження можливостей компонентного підходу при розробці веб-сайтів* : Збірник наукових тез. Тернопіль, 2023. С.103-104

## ДОДАТКИ

### Додаток А

#### Програмний код серверної частини

##### Лістинг А1 – Запити на створення таблиць

```
import { Schema, model, models } from "mongoose";

const MenuSchema = new Schema({
  name: String,
  category: String,
  description: String,
  image: String,
  like: Number,
  persent_alcho: Number,
  price: Number,
  weigth: Number,
  type: String,
});

const MenuModel = models.Menu || model("Menu", MenuSchema);
export default MenuModel;
```

##### Лістинг А2 – Запити на створення користувача

```
import { Schema, model, models } from "mongoose";
import bcrypt from "bcryptjs";

const userSchema = new Schema({
  name: String,
  email: String,
  password: String,
  role: {
    type: String,
    default: "cooker",
  },
});
```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дат		

```

    },
  });
  userSchema.pre("save", async function (next) {
    if (!this.isModified("password")) {
      next();
    }
    this.password = await bcrypt.hash(this.password, 5);
  });

  const UserModel = models.Users || model("Users", userSchema);
  export default UserModel;

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дат		

### Лістинг А3 – Запити на створення керівника

```
import { Schema, model, models } from "mongoose";

const OrderSchema = new Schema(
  {
    tableId: Number,
    order: Array,
    payment: String,
    status: Number,
    totalPrice: Number,
  },
  { versionKey: false, timestamps: true }
);

const OrderModel = models.Orders || model("Orders",
OrderSchema);

export default OrderModel;
```

### Лістинг А4 – Запити на створення доступу столів

```
import { Schema, model, models } from "mongoose";

const TablesSchema = new Schema(
  {
    tableNumber: { unique: true, type: Number },
    url: String,
  },
  { versionKey: false }
);

const TablesModel = models.Tables || model("Tables",
TablesSchema);

export default TablesModel;
```

### Лістинг А5- Замовлення

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дат		

```

import dbConnect from "@/lib/mongodb";
import OrderModel from "@/models/Order";

dbConnect();

const handler = async (req, res) => {
  try {
    const body = req.body;

    if (req.method === "POST") {
      const results = await OrderModel.create(body);
      res.status(201).json(results);
    }
  } catch (error) {
    console.log(error);
  }
};

export default handler;

import mongoose from "mongoose";
mongoose.set("strictQuery", false);
const MONGODB_URI = process.env.MONGODB_URI;

if (!MONGODB_URI) {
  throw new Error(
    "Please define the MONGODB_URI environment variable inside .env.local"
  );
}

let cached = global.mongoose;

if (!cached) {
  cached = global.mongoose = { conn: null, promise: null };
}

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дат		



```

async function dbConnect() {
  if (cached.conn) {
    return cached.conn;
  }

  if (!cached.promise) {
    const opts = {
      useUrlParser: true,
      useUnifiedTopology: true,
    };

    cached.promise = mongoose.connect(MONGODB_URI,
    opts).then((mongoose) => {
      console.log("mong con");

      return mongoose;
    });
  }
  cached.conn = await cached.promise;

  return cached.conn;
}

export default dbConnect;

import dbConnect from "@/lib/mongodb";
import UserModel from "@/models/Users";

const handle = async (req, res) => {
  dbConnect();
  try {
    const { id } = req.query;
    if (req.method === "PUT") {
      const body = req.body;
      const results = await UserModel.findByIdAndUpdate(id,
body);

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        res.status(201).json(results);
    } else if (req.method === "DELETE") {
        const results = await UserModel.deleteOne({ _id: id });
        res.status(201).json(results);
    }
} catch (error) {
    console.log(error);
}
};

export default handle;

import dbConnect from "@/lib/mongodb";
import UserModel from "@/models/Users";

const register = async (req, res) => {
    if (req.method === "POST") {
        dbConnect();

        const user = await UserModel.create(req.body);
        res.status(201).json({ user });
    }
};

export default register;

```

## Додаток Б

### Функціональні компоненти

#### Лістинг Б1- Компонент форми реєстрації

```
import * as React from "react";
import Avatar from "@mui/material/Avatar";

import TextField from "@mui/material/TextField";

import Grid from "@mui/material/Grid";
import Box from "@mui/material/Box";
import LockOutlinedIcon from "@mui/icons-material/LockOutlined";
import Typography from "@mui/material/Typography";
import Container from "@mui/material/Container";
import EmptyLayout from "@components/Layout/EmptyLayout";
import axios from "axios";
import { useRouter } from "next/router";
import { FormControl, InputLabel, MenuItem, Select } from "@mui/material";
import { useState } from "react";
import LoadingButton from "@mui/lab/LoadingButton";

const SignUp = () => {
  const router = useRouter();
  const [role, setRole] = useState("cooker");
  const [loading, setLoading] = useState(false);
  const handleSubmit = async (event) => {
    event.preventDefault();
    const formData = new FormData(event.currentTarget);
    setLoading(true);
    try {
      const { data } = await axios.post("/api/register", {
```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        name: formData.get("name"),
        email: formData.get("email").toLowerCase(),
        password: formData.get("password"),
        role: formData.get("role"),
    });
    setLoading(false);
    router.back();
} catch (error) {
    setLoading(false);
    console.log("=====");
    console.log(error);
    console.log("=====");
}
};

return (
    <Container component="main" maxWidth="xs">
        <Box
            sx={{
                marginTop: 8,
                display: "flex",
                flexDirection: "column",
                alignItems: "center",
            }}
        >
            <Avatar sx={{ m: 1, bgcolor: "secondary.main" }}>
                <LockOutlinedIcon />
            </Avatar>
            <Typography component="h1" variant="h5">
                Sign up
            </Typography>
            <Box
                component="form"
                noValidate
                onSubmit={handleSubmit} sx={{ mt: 3 }}>
                <Grid container spacing={2}>
                    <Grid item xs={12}>
                        <TextField

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        autoComplete="given-name"
        name="name"
        required
        fullWidth
        id="name"
        label="First Name"
        autoFocus
    />
</Grid>

<Grid item xs={12}>
    <TextField
        required
        fullWidth
        id="email"
        label="Email Address"
        name="email"
        autoComplete="email"
        type={"email"}
    />
</Grid>
<Grid item xs={12}>
    <TextField
        required
        fullWidth
        name="password"
        label="Password"
        type="password"
        id="password"
        autoComplete="new-password"
    />
</Grid>
<Grid item xs={12}>
    <FormControl size="small" fullWidth>
        <InputLabel id="demo-select-
small">Role</InputLabel>

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        <Select
            labelId="demo-select-small"
            id="demo-select-small"
            value={role}
            label="Age"
            name="role"
            onChange={ (event) =>
setRole(event.target.value) }
        >
            <MenuItem value={"admin"}>Адмін</MenuItem>
            <MenuItem value={"cooker"}>Кухар</MenuItem>
            <MenuItem
value={"barman"}>Бармен</MenuItem>
        </Select>
    </FormControl>
</Grid>
</Grid>
<LoadingButton
    type="submit"
    sx={{ mt: 3, mb: 2 }}
    fullWidth
    loading={loading}
    loadingIndicator="Loading..."
    variant="contained"
    >
        <span>Sign Up</span>
    </LoadingButton>
</Box>
</Box>
</Container>
);
};
SignUp.Layout = EmptyLayout;
export default SignUp;

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дат		

## Лістинг Б2- Компонент форми авторизації

```
import EmptyLayout from "@components/Layout/EmptyLayout";
import Avatar from "@mui/material/Avatar";
import Button from "@mui/material/Button";
import TextField from "@mui/material/TextField";
import Box from "@mui/material/Box";
import LockOutlinedIcon from "@mui/icons-material/LockOutlined";

import Typography from "@mui/material/Typography";
import Container from "@mui/material/Container";
import { signIn } from "next-auth/react";
import Router from "next/router";
import { useState } from "react";

const Login = () => {

  const [error, setError] = useState();
  const handleSubmit = async (event) => {
    event.preventDefault();
    const data = new FormData(event.currentTarget);

    const { ok, error } = await signIn("credentials", {
      email: data.get("email"),
      password: data.get("password"),
      redirect: false,
    });

    if (ok) Router.replace("/admin");
    if (error) setError(error);
  };

  return (
    <Container component="main" maxWidth="xs">
      <Box
        sx={{
```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        marginTop: 8,
        display: "flex",
        flexDirection: "column",
        alignItems: "center",
      }}
    >
    <Avatar sx={{ m: 1, bgcolor: "secondary.main" }}>
      <LockOutlinedIcon />
    </Avatar>
    <Typography component="h1" variant="h5">
      Sign in
    </Typography>
    <Box      component="form"      onSubmit={handleSubmit}
noValidate sx={{ mt: 1 }}>
      <TextField
        margin="normal"
        required
        fullWidth
        id="email"
        label="Email Address"
        name="email"
        autoComplete="email"
        autoFocus
      />
      <TextField
        margin="normal"
        required
        fullWidth
        name="password"
        label="Password"
        type="password"
        id="password"
        autoComplete="current-password"
      />
      <Typography color="error">{error}</Typography>
      <Button

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дат		



```

        type="submit"
        fullWidth
        variant="contained"
        sx={{ mt: 3, mb: 2 }}
      >
        Sign In
      </Button>
    </Box>
  </Box>
</Container>
);
};

```

```

export default Login;
Login.Layout = EmptyLayout;

```

### Лістинг Б3 – Функціонал «користувачів»

```

import { Box, Container, Fab, Grid, Paper } from
"@mui/material";

import PeopleSettings from
"@/components/Admin/PeopleSettings";

import AdminLayout from "@/components/Layout/AdminLayout";
import dbConnect from "@/lib/mongodb";
import AddIcon from "@mui/icons-material/Add";
import UserModel from "@/models/Users";
import Link from "next/link";
import { useState } from "react";

const People = ({ users }) => {
  const [data, setData] = useState(users);

  return (
    <>
      <Box
        component="main"
        sx={{
          backgroundColor: (theme) =>

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        theme.palette.mode === "light"
        ? theme.palette.grey[100]
        : theme.palette.grey[900],
    flexGrow: 1,
    height: "100vh",
    overflow: "auto",
  }}
>
<Container maxWidth="lg" sx={{ mt: 4, mb: 4 }}>
  <Grid
    container
    spacing={3}
    sx={{ display: "flex", justifyContent: "center"
  }}
  >
    <Grid item xs={12} md={8} lg={9}>
      <Paper
        sx={{
          p: 2,
          display: "flex",
          flexDirection: "column",
        }}
      >
        <Box sx={{ display: "flex", gap: 2, flexWrap:
"wrap" }}>
          {data.map((user) => {
            return (
              <PeopleSettings
                key={user._id}
                user={user}
                setData={setData}
              />
            );
          })}
        </Box>
      </Paper>
    </Grid>
  </Grid>
</Container>

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        </Grid>
    </Grid>
</Container>
</Box>
<Link href="/signup">
    <Fab
        color="primary"
        aria-label="add"
        sx={{ position: "absolute", bottom: 20, right: 20
    }}
    >
        <AddIcon />
    </Fab>
</Link>
</>
);
};

export default People;
People.Layout = AdminLayout;

export async function getServerSideProps(ctx) {
    try {
        await dbConnect();

        const results = await UserModel.find();

        return {
            props: {
                users: JSON.parse(JSON.stringify(results)),
            },
        };
    } catch (error) {
        console.log(error);
        return { notFound: true };
    }
}

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дат		

```

    }

import {
    Box,
    Typography,
    InputLabel,
    MenuItem,
    FormControl,
    IconButton,
    Paper,
    Button,
} from "@mui/material";

import Select from "@mui/material/Select";
import { useEffect, useState } from "react";

import DeleteOutlineIcon from "@mui/icons-
material/DeleteOutline";
import axios from "axios";

const PeopleSettings = ({ user, setData }) => {
    const [role, setRole] = useState(user.role);
    const [showSub, setShowSub] = useState(false);
    useEffect(() => {
        if (role !== user.role) {
            setShowSub(true);
        } else {
            setShowSub(false);
        }
    }, [user.role, role]);
    const handleChange = (event) => {
        setRole(event.target.value);
    };
    const handleClick = async () => {
        const { data } = await
        axios.put(`/api/role?id=${user.id}`, { role });
    };

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		67

```

        setShowSub(false);
    };
    const handleClickDel = async () => {
        try {
            setData((prev) => prev.filter((e) => e._id !==
user._id));
            const { data } = await
axios.delete(`/api/role?id=${user._id}`);
        } catch (error) {
            console.log(error);
        }
    };
    return (
        <>
        <Paper
            sx={{
                p: 1,
                width: 140,
                display: "flex",
                flexDirection: "column",
            }}
        >
            <Typography
textAlign="center">{user.name}</Typography>
            <FormControl sx={{ my: 1 }} size="small" fullWidth>
                <InputLabel id="demo-select-
small">Role</InputLabel>
                <Select
                    labelId="demo-select-small"
                    id="demo-select-small"
                    value={role}
                    label="Age"
                    onChange={handleChange}
                >
                    <MenuItem value={"admin"}>Адмін</MenuItem>
                    <MenuItem value={"cooker"}>Кухар</MenuItem>

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        <MenuItem value={"barman"}>Бармен</MenuItem>
      </Select>
    </FormControl>
    <Box display="flex">
      <IconButton color="error" onClick={handleClickDel}>
        <DeleteOutlineIcon></DeleteOutlineIcon>
      </IconButton>

      {showSub                                     &&                                     <Button
onClick={handleClick}>Submit</Button>}

    </Box>
  </Paper>
</>

);
};

export default PeopleSettings;

import {

  Divider,
  IconButton,
  List,
  Toolbar,
  Typography,
} from "@mui/material";
import MuiAppBar from "@mui/material/AppBar";
import MuiDrawer from "@mui/material/Drawer";

import MenuIcon from "@mui/icons-material/Menu";
import ChevronLeftIcon from "@mui/icons-material/ChevronLeft";

import LogoutIcon from "@mui/icons-material/Logout";

import { styled, createTheme, ThemeProvider } from
"@mui/material/styles";

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дат		

```

import CssBaseline from "@mui/material/CssBaseline";

import Box from "@mui/material/Box";

import { MainListItems } from "@components/Admin/ListItem";
import { useState } from "react";
import { SessionProvider } from "next-auth/react";
import { signOut } from "next-auth/react";

const drawerWidth = 240;

const AppBar = styled(MuiAppBar, {
  shouldForwardProp: (prop) => prop !== "open",
})(({ theme, open }) => ({
  zIndex: theme.zIndex.drawer + 1,
  transition: theme.transitions.create(["width", "margin"], {
    easing: theme.transitions.easing.sharp,
    duration: theme.transitions.duration.leavingScreen,
  }),
  ...(open && {
    marginLeft: drawerWidth,
    width: `calc(100% - ${drawerWidth}px)`,
    transition: theme.transitions.create(["width", "margin"],
{
    easing: theme.transitions.easing.sharp,
    duration: theme.transitions.duration.enteringScreen,
  }),
  }),
}));

const Drawer = styled(MuiDrawer, {
  shouldForwardProp: (prop) => prop !== "open",
})(({ theme, open }) => ({
  "& .MuiDrawer-paper": {
    position: "relative",
    whiteSpace: "nowrap",
    width: drawerWidth,

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		70

```

        transition: theme.transitions.create("width", {
          easing: theme.transitions.easing.sharp,
          duration: theme.transitions.duration.enteringScreen,
        })),
        boxSizing: "border-box",
        ...(!open && {
          overflowX: "hidden",
          transition: theme.transitions.create("width", {
            easing: theme.transitions.easing.sharp,
            duration: theme.transitions.duration.leavingScreen,
          })),
        width: theme.spacing(7),
        [theme.breakpoints.up("sm")]: {
          width: theme.spacing(9),
        },
      })),
    },
  ));
const mdTheme = createTheme();

const AdminLayout = ({ children }) => {
  const [open, setOpen] = useState(false);

  const toggleDrawer = () => {
    setOpen(!open);
  };

  return (
    <>
      <SessionProvider>
        <ThemeProvider theme={mdTheme}>
          <Box sx={{ display: "flex" }}>
            <CssBaseline />
            <AppBar position="absolute" open={open}>
              <Toolbar
                sx={{

```



```

pr: "24px", // keep right padding when
drawer closed

    }}
  >
    <IconButton
      edge="start"
      color="inherit"
      aria-label="open drawer"
      onClick={toggleDrawer}
      sx={{
        marginRight: "36px",
        ...(open && { display: "none" }),
      }}
    >
      <MenuIcon />
    </IconButton>
    <Typography
      component="h1"
      variant="h6"
      color="inherit"
      noWrap
      sx={{ flexGrow: 1 }}
    >
      Dashboard
    </Typography>
    <IconButton color="inherit" onClick={() =>
signOut()}>

      <LogoutIcon />
    </IconButton>
  </Toolbar>
</AppBar>
<Drawer variant="permanent" open={open}>
  <Toolbar
    sx={{
      display: "flex",
      alignItems: "center",

```

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        justifyContent: "flex-end",
        px: [1],
      }}
    >
    <IconButton onClick={toggleDrawer}>
      <ChevronLeftIcon />
    </IconButton>
  </Toolbar>
  <Divider />
  <List component="nav">
    <MainListItems></MainListItems>
  </List>
</Drawer>
<Box
  component="main"
  sx={{
    backgroundColor: (theme) =>
      theme.palette.mode === "light"
        ? theme.palette.grey[100]
        : theme.palette.grey[900],
    flexGrow: 1,
    height: "100vh",
    overflow: "auto",
  }}
>
  <Toolbar />
  {children}
</Box>
</Box>
</ThemeProvider>
</SessionProvider>
</>
);
};

```

```
export default AdminLayout;
```

					ДП.КН 23.505.08.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дат		

					ЛП.КН 23.505.08.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		74

# Додаток Г

## Таблиці до економічного розділу

Таблиця Г1 – Кошторис витрат на проектування

Найменування статей витрат	Сума, грн	Обґрунтування
1. Зарплата проектувальників.	113600	Сумарна вартість оплати праці усіх фахівців до створення проекту.
2. Відрахування на соціальні потреби.	18902	Сума соціальних відрахувань із коштів виділених на оплату праці (становить 22% від суми заробітних плат).
3. Контрагентські роботи і послуги.	0	Проект не потребує контрагентських робіт та послуг.
4. Витрати на відрядження.	0	Витрати на відрядження не передбачені через відсутність їх потреби.
5. Інші прямі витрати.	500	Вартість домену та хостингу.
6. Усього прямих витрат.	133002	Прямі витрати проекту.
7. Накладні витрати.	39900,6	Витрати на забезпечення проведення робіт, амортизація.
8. Планові накопичення.	34580,52	Витрати на розвиток проекту.
9. Усього, кошторисна вартість проекту.	207483,12	Сума прямих, накладних витрати і планових накопичень.
10. Податок на додану вартість.	41496,624	Розраховується як 20% від кошторисної вартості.
11. Загалом, договірна ціна розробки Зп.	248979,744	Витрати замовника на проект.

Таблиця Г2 – Розрахунок заробітної плати проектувальників

Посада виконавця	Ок лад	Відрахув ання	Кількість		Сум а
	гр н/міс	грн/міс	ол.	мі сяців	з/п, грн.
Керівни к ДП	90 00	1245		4	360 00
Back-end розробн ик	45 500	5700		1	455 00
Front- end розробн ик	18 500	4500		1	185 00
Тестувал ьник	13 600	13000		1	136 00
Загальна сума зарплати:					113 600

Таблиця Г3 – Розрахунок податків на доходи фізичних осіб

Посада	Обчислення згідно чинного законодавства в сфері податкування та середньої ринкової заробітної плати
Керівник ДП	$9000 * 18\% = 1620$ грн
Back-end розробник	$45500 * 18\% = 8190$ грн
Front-end розробник	$18500 * 18\% = 3330$ грн
Тестуваль ник	$13600 * 18\% = 2448$ грн

Таблиця Г4 – Розрахунок військового збору

Керівник ДП	$9000 * 1.5\% = 135$ грн
----------------	--------------------------

Back-end розробник	$45500 * 1.5\% = 682,5$ грн
Front-end розробник	$18500 * 1.5\% = 277,5$ грн
Тестуваль ник	$13600 * 1.5\% = 204$ грн

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дат		

Таблиця Г5 – Розрахунок єдиного соціального внеску

Посада	Обчислення згідно чинного законодавства в сфері податкування та середньої ринкової заробітної плати
Керівник ДП	$9000 * 22\% = 1980$ грн
Back-end розробник	$45500 * 22\% = 9900$ грн
Front-end розробник	$18500 * 22\% = 4070$ грн
Тестуваль ник	$13600 * 22\% = 2992$ грн

Таблиця Г6 – Розрахунок вартості утримання

Посада	Обчислення згідно чинного законодавства в сфері податкування та середньої ринкової заробітної плати
Керівник ДП	1755 (1620+135)
Back-end розробник	8872,5 (8190+682,5)
Front-end розробник	3607,5 (3330+277,5)
Тестуваль ник	2652 (2448+204)

Таблиця Г7 – Розрахунок суми виплати працівнику

Посада	Обчислення згідно чинного законодавства в сфері податкування та середньої ринкової заробітної плати
Керівник	7245 (9000грн-1620грн-135грн)

ДП	
Back-end розробник	36627,5 (45500грн-8190грн-682,5грн)
Front-end розробник	14892,5 (18500грн-3330грн-277,5грн)
Тестуваль ник	10948 (13600грн-2448грн-204грн)

					ЛП.КН 23.505.08.000 ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дат		



**ВІДГУК**  
на дипломний проєкт  
студента відділення комп'ютерних технологій  
Галицького фахового коледжу імені В'ячеслава Чорновола

студента IV курсу групи КН-41  
Дуніковського Стефана Богдановича \_\_\_\_\_  
(прізвище та ініціали)

Спеціальність 122 «Комп'ютерні науки»

Керівник ДП \_\_ Кузик Василь Миколайович \_\_\_\_\_

Тема: «Web-додаток для організації автоматизації процесів ресторану»

1. Загальна характеристика студента під час написання ДП: \_\_ студент проявив себе як фахівець, яка володіє засобами проектування та створення відповідної інформаційної системи. Хоча на ринку програмних продуктів існує велика кількість аналогічного програмного забезпечення, автор достатньо переконливо аргументує необхідність створення своєї системи, акцентуючи увагу на сучасних засобах розробки та авторській архітектурі системи на основі розробленої функціональної моделі.

2. Практична або теоретична цінність опрацьованих питань: у роботі продемонстрована практична значущість. При реалізації дипломного проєкту обґрунтовано вибрані технології та програмні засоби реалізації інформаційної системи. Пояснювальна записка написана чітко та ясно. Автор мотивує своє рішення щодо архітектури, демонструє високу програмістську кваліфікацію. З даною темою студентка виступила на Днях науки у Галицькому фаховому коледжі імені В.Чорновола та опублікувала статтю в віснику наукових тез вказаного закладу.

3. Недоліки роботи: \_\_\_\_\_ суттєвих недоліків не виявлено \_\_\_\_\_

4. Загальний висновок \_\_ дипломний проєкт виконаний згідно вимог і заслуговує на позитивну оцінку \_\_\_\_\_

Керівник дипломного проєкту

  
\_\_\_\_\_

Кузик Василь Миколайович, викладач ЦК Інформатики та комп'ютерних дисциплін

23 червня 2023 р.

---

РЕЦЕНЗІЯ  
на дипломний проєкт  
відділення комп'ютерних технологій  
Галицького фахового коледжу імені В'ячеслава Чорновола  
студента IV курсу групи КН-41

Дуниковського Стефана  
(прізвище та ініціали)

Спеціальність 122 „Комп'ютерні науки”

Обсяг дипломного проєкту: 50 стор.

Кількість аркушів креслень     арк.

Кількість сторінок записки: 79 стор.

Тема: Web-застосунок для організації автоматизації процесів ресторану

1. Актуальність теми: *одним з головних викликів для ресторанного бізнесу є оптимізація процесів та забезпечення якості обслуговування. Сьогоднішні клієнти мають високі вимоги до швидкості та якості обслуговування, тому ресторани повинні дбайливо підходити до цього питання. І одним з ключових рішень для досягнення цієї мети є використання інформаційних технологій.*

2. Практична або теоретична цінність опрацьованих питань: *робота спроектована під замовлення, насичена ілюстративним матеріалом (рисунками і таблицями), що в повній мірі дозволяє відобразити як теоретичну так і практичну частину аналізу предметної області.*

3. Недоліки роботи: *перелік джерел посилання є досить невеликим як для дипломного проєкту.*

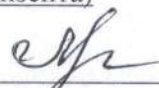
4. Загальний висновок: *дипломний проєкт розроблений економічно грамотно, тему розкрито. Дотримані усі вимоги на достатньому рівні для присвоєння відповідної фахової кваліфікації.*

Рецензент

Чубей О.О.

(прізвище та ініціали рецензента)

«22» 06 2023р.

  
(підпис)

Ім'я користувача:  
Василь Кузик

Дата перевірки:  
15.06.2023 15:38:58 EEST

Дата звіту:  
15.06.2023 15:43:35 EEST

ID перевірки:  
1015615360

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100012366

Назва документа: Дуніковський

Кількість сторінок: 46 Кількість слів: 6115 Кількість символів: 46232 Розмір файлу: 54.74 MB ID файлу: 1015262833

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

8.14%  
Схожість

Найбільша схожість: 3.79% з джерелом з Бібліотеки (ID файлу: 1011434117)

1.7% Джерела з Інтернету

43

Сторінка 48

6.74% Джерела з Бібліотеки

36

Сторінка 48

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%  
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

11

Підозріле форматування

11  
сторінок