

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій  
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

другий (магістерський)  
(рівень вищої освіти)

Розроблення інформаційної системи для оптимізації  
бізнес-процесів закладу харчування  
(тема)

Виконав:  
студент 2 курсу, групи АУТПм-22-2

Веснянка В. О.

Спеціальності 151 Автоматизація та  
комп'ютерно-інтегровані технології

Тип програми Освітньо-професійна

Освітня програма Автоматизоване  
управління технологічними процесами

Керівник доц. Демська Н. П.

Допускається до захисту  
Зав. кафедри КІТАР

\_\_\_\_\_  
(підпис)

Невлюдов І. Ш.  
(прізвище, ініціали)

2024р.

# ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет \_\_\_\_\_ АКТ \_\_\_\_\_  
Кафедра \_\_\_\_\_ КІТАР \_\_\_\_\_  
Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 151 Автоматизація та комп'ютерно-інтегровані технології \_\_\_\_\_  
Тип програми \_\_\_\_\_ Освітньо-професійна \_\_\_\_\_  
Освітня програма Автоматизоване управління технологічними процесами \_\_\_\_\_  
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР \_\_\_\_\_  
(підпис)

« \_ » \_\_\_\_\_ 20\_ р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Веснянці Владиславу Олександровичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інформаційної системи для оптимізації бізнес-процесів закладу харчування

Затверджена наказом по університету від 03.11.2023 р. № 11286 Ст \_\_\_\_\_

2. Термін подання студентом роботи до екзаменаційної комісії 10.01.2024 р.

3. Вихідні дані до роботи Об'єкт дослідження – бізнес процеси закладів харчування. Предмет дослідження – розроблена інформаційна система, що складається з комплексу програмних засобів.

3.1 Операційна система Windows;

3.2 Середовище розробки Microsoft Visual Studio;

3.3 Середовище розробки Visual Studio Code;

3.4 Система управління базами даних SQL Server Management Studio;

3.5 Мова програмування C#, JavaScript, React;

4. Перелік питань, що потрібно опрацювати в роботі Вступ. Визначення мети, об'єкту та предмету дослідження. Аналіз існуючого стану бізнес процесів у закладх харчування. Опис проблематики створення системи оптимізації бізнес процесів закладу харчування. Дослідження існуючих програмних продуктів у сфері оптимізації бізнес процесів. Опис засобів розробки. Постановка задачі. Формування технічного завдання. Системні вимоги. Функціональні вимоги. Вибір мов програмування. Вибір середовища розробки бази даних. Опис структури інформаційної системи. Тестування розробленої системи. Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Графічний матеріал у вигляді презентації – 12 арк. ф. А 4

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Вступ. Визначення мети, об'єкту і предмету дослідження	5.11-15.11 2023	викон.
2	Проведення аналізу існуючого стану бізнес процесів у закладах харчування	15.11-19.11 2023	викон.
3	Опис проблематики створення системи оптимізації бізнес процесів закладу харчування	19.11-25.11 2023	викон.
4	Дослідження існуючих програмних продуктів у сфері оптимізації бізнес процесів	25.11-01.12 2023	викон.
5	Постановка завданчі проєкту	01.12-05.12 2023	викон.
6	Розробка системних та функціональних вимог до програмного засобу	05.12-15.12 2023	викон.
7	Розробка програмного засобу	15.12-29.12 2023	викон.
8	Тестування розробленого програмного засобу	29.12-31.12 2023	викон.
9	Розглядання питань, пов'язаних з охороною праці	01.01-02.01 2024	викон.
10	Оформлення пояснювальних записки	02.01-07.01 2024	викон.
11	Подання роботи на перевірку Інтернет-сервісом Unichesk	14.01-16.01 2024	викон.
12	Подання роботи на рецензію	16.01-20.01 2024	викон.
13	Подання роботи на підпис зав. кафедри	20.01-22.01 2024	викон.
14	Подання кваліфікаційної роботи в ЕК	23.01-24.01 2024	викон.

Дата видачі завдання 01.09.2023 р.

Студент \_\_\_\_\_ Веснянка В. О.

(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

доц. Демська Н. П.

(посада, прізвище, ініціали)

*Я, як студент ХНУРЕ (Веснянка Владислав Олександрович), розумію і підтримую політику закладу із академічної доброчесності. Я не надав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.*

Дата

04.01.2024 р.



Веснянка В.О.

## РЕФЕРАТ

Пояснювальна записка: 88 с., 3 табл., 58 рис., 2 додатки, 37 джерел.

ІНТЕРФЕЙС, ІНФОРМАЦІЙНА СИСТЕМА, ПРОГРАМНИЙ ЗАСІБ,  
БІЗНЕС-ПРОЦЕСИ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ВЕБ САЙТ.

Об'єкт дослідження – бізнес процеси закладів харчування.

Предмет дослідження – програмний застосунок для оптимізації бізнес-процесів закладу харчування.

Мета роботи – створення інформаційної системи для збільшення ефективності управління бізнес-процесами закладу харчування для підвищення якості та швидкості надання обслуговування, оптимізації управління закладом харчування за рахунок автоматизації процесів.

У першому розділі проаналізовано існуючий стан бізнес процесів у закладах харчування, було описано проблематику створенн системи оптимізації бізнес процесів, а також визначено сферу застосування та проведено дослідження існуючих аналогів.

В другому розділі проведено опис засобів розробки та поставлена задача на проєкт, що дозволить чітко дотримуватися визначенії етапів розробки інформаційної системи. Визначено основні системні та функціональні вимоги до проєкту. Та головне, обрано мови програмування для фронтенд та бекенд частин, а також середовище розробки бази даних.

У завершальному третьому розділу було розроблено інтерфейс інформаційної системи, а також програмний код бекенд частини та проведено тестування розробленої системи.

У результаті розроблена інформаційна система з оптимізації бізнес процесів закладу харчування у вигляді веб-сайту, яка оптимізована для будь-яких пристроїв з різними розмірами дисплею.

## ABSTRACT

Explanatory note: 85 pages, 1 tables, 56 figures, 2 appendices, 37 sources.

INTERFACE, INFORMATION SYSTEM, SOFTWARE, BUSINESS PROCESSES, SOFTWARE, WEBSITE.

The object of research is the business processes of catering establishments.

The subject of research is a developed information system consisting of a complex of software tools.

The purpose of the work is to create an information system to increase the efficiency of management of business processes of a catering establishment to improve the quality and speed of service provision, to optimize the management of a catering establishment due to the automation of processes.

In the first chapter, the existing state of business processes in food establishments was analyzed, the problems of creating a business process optimization system were described, as well as the scope of application was determined and a study of existing analogues was carried out.

In the second section, a description of the development tools is provided and a task is set for the project, which will allow clearly adhering to the defined stages of the development of the information system. The main system and functional requirements for the project have been determined. And most importantly, the programming languages for the front-end and back-end parts, as well as the database development environment, have been chosen.

In the final third section, the interface of the information system was developed, as well as the program code of the backend part, and the developed system was tested.

As a result, an information system for optimizing the business processes of a catering establishment was developed in the form of a website, which is optimized for any devices with different display sizes.

## ЗМІСТ

Перелік скорочень.....	7
Вступ.....	8
1 Аналіз існуючого стану бізнес-процесів у закладах харчування.....	10
1.1 Автоматизація бізнес-процесів у закладі харчування.....	11
1.2 Опис проблематики створення системи оптимізації бізнес-процесів закладу харчування.....	14
1.3 Визначення сфери застосування розроблювальної системи.....	15
1.4 Дослідження існуючих програмних продуктів у сфері оптимізації бізнес-процесів для закладу харчування.....	16
1.5 Висновки до розділу.....	18
2 Аналіз технічного завдання та постановка задач дослідження.....	19
2.1 Опис засобів розробки та постановка задачі.....	19
2.2 Формування тіхчного завдання до додатку, що розроблюється.....	22
2.2.1 Системні вимоги.....	22
2.2.2 Функціональні вимоги.....	23
2.3 Вибір засобів реалізації.....	30
2.3.1 Вибір мов програмування.....	31
2.3.2 Вибір середовища розробки та СУБД.....	34
2.4 Висновки до розділу.....	35
3 Розробка та тестування додатку.....	37
3.1 Опис структури інформаційної системи.....	37
3.2 Розробка інтерфейсу програмного засобу.....	38
3.3 Розробка програмного коду застосунку.....	40
3.4 Розроблений програмний додаток.....	50
3.5 Тестування розробленої системи.....	62
3.6 Експериментальні дослідження.....	69
3.7 Охорона праці.....	71

	6
3.8 Висновки до розділу.....	74
Висновки.....	72
Перелік джерел посилання.....	73
Додаток А Апробація наукових результатів.....	77
Додаток Б Демонстраційний матеріал.....	85

## ПЕРЕЛІК СКОРОЧЕНЬ

- ВРМ – управління бізнес-процесами;
- БП – бізнес-процеси;
- ПЗ – програмний засіб;
- БД – база даних;
- СУБД – система управління базою даних;
- ООП – об’єктно орієнтоване програмування;
- ІК – інтерфейс користувача;
- SSMS – SQL Server Management Studio;
- VS Code – Visual Studio Code;
- API – Application Programming Interface;
- ORM – Object-Relational Mapping;
- DTO – Data Transfer Object.

## ВСТУП

Сьогодні підприємства громадського харчування мають важливе значення для життя суспільства. Автоматизація роботи закладів громадського харчування є важливим кроком, який має на меті покращити управління та забезпечити суворий контроль за всіма операційними процесами закладу. Розробка інформаційної системи полягає в тому, щоб створити ефективний і функціональний інструмент для покращення управління всіма аспектами діяльності закладу харчування.

Ключові цілі проекту включають автоматизацію рутинних операцій, оптимізацію контролю за запасами, покращення обробки замовлень і підвищення ефективності управління персоналом. Одним із найефективніших інструментів забезпечення ефективності діяльності підприємств є оптимізація бізнес-процесів, яка допомагає збільшити прибуток і збільшити продуктивність, одночасно знижуючи витрати, а також адаптуючи їх до потреб клієнтів і споживачів. Методи оптимізації, які дозволять встановити баланс між задоволенням клієнтів і конкурентними силами ринку з одного боку, і показниками ефективності бізнес-процесів з іншого, необхідні для максимізації якості бізнес-процесів на підприємствах України.

Мета системи також включає в себе покращення якості обслуговування клієнтів, шляхом поліпшення способів комунікації, забезпечення зручних сервісів замовлення, а також швидкого та точного обліку замовлень. Важливою складовою мети є створення зручних аналітичних інструментів для аналізу різних характеристик закладу.

Таким чином метою роботи є створення інформаційної системи для збільшення ефективності управління бізнес-процесами закладу харчування для підвищення якості та швидкості надання обслуговування, оптимізації управління закладом харчування за рахунок автоматизації процесів.

Об'єктом дослідження є бізнес процеси закладів харчування.

Предметом дослідження є розроблена інформаційна система, що складається з комплексу програмних засобів.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести аналіз існуючого стану бізнес процесів у закладах харчування;
- провести аналіз існуючих систем для автоматизації бізнес-процесів у закладах харчування, визначити їх переваги та недоліки;
- проаналізувати технічне завдання;
- визначити функціональні можливості розроблювальної системи;
- провести аналіз та обрати мови програмування для розробки інформаційної системи;
- проаналізувати та обрати СУБД;
- розробити систему згідно поставленої мети.

Кваліфікаційна робота виконана згідно [1-4]. Результати роботи опубліковані в [5].

## 1 АНАЛІЗ ІСНУЮЧОГО СТАНУ БІЗНЕС-ПРОЦЕСІВ У ЗАКЛАДАХ ХАРЧУВАННЯ

Аналіз поточного стану бізнес-процесів (БП) у закладах харчування може виявити проблеми, які можуть вплинути на ефективне обслуговування клієнтів і управління закладом. Заклади харчування часто стикаються з проблемами, пов'язаними з великою кількістю ручної роботи. Це включає бухгалтерію, оформлення замовлень, облік товарів, відсутність ефективної системи керування персоналом тощо. Заклади харчування не автоматизують достатньо процесів, що викликає більший час для оформлення замовлень і виконання інших завдань, а також відсутність зручної аналітики для контролю. Ресторани є складною системою з багатьма різними бізнес-процесами, які необхідні для ефективної та високоякісної роботи. Щоб покращити свою роботу, власники ресторанів повинні мати можливість відстежувати та аналізувати кожен етап своєї діяльності [5].

Аналіз закладів харчування, який включає такі елементи, як аналіз продажів, фінансів і персоналу, дозволяє удосконалити управління та виявити слабкі сторони, що призводить до кращого обслуговування та функціонування закладу.

Ці рішення можливі завдяки системи оптимізації бізнес-процесів. На теперішній час можна виділити наступні основні бізнес процеси у закладах харчування [5]:

- управління складом та постачанням;
- управління меню;
- готування страв;
- обслуговування клієнтів;
- фінансове управління;
- управління персоналом;
- моніторинг;

– звітність.

Найчастіше автоматизацію бізнес-процесів виконують для процесів пов'язаних з фінансами, продажами, управління персоналом, управління складом. Тому розроблювальна програма повинна допомогти вирішувати ці питання [5].

Отже, для вирішення питання оптимізації деяких перерахованих бізнес-процесів необхідно ввести систему автоматизації.

### 1.1 Автоматизація бізнес-процесів у закладі харчування

Використання процесного підходу в управлінні підприємством передбачає більш глибоке розуміння поняття бізнес-процесів. Поняття – бізнес-процес, що хоча і не отримало доки нормативного тлумачення, є таким, що найчастіше вживається. У широкому розумінні під бізнес-процесом розуміють структуровану послідовність дій по виконанню відповідного виду діяльності на всіх етапах життєвого циклу предмета діяльності [5].

Бізнес-процес – це сукупність різних видів діяльності, в межах якої «на вході» використовується один або декілька ресурсів, а в результаті цієї діяльності «на виході» створюється продукт, що має цінність для споживача. Деякі автори розглядають бізнес-процес як множину внутрішніх кроків (видів) діяльності по створенню продукції, необхідної клієнту, вартість, довговічність, сервіс та якість якої задовольняє споживача [6, 7].

Оптимізація бізнес-процесів – це практика підвищення ефективності організації шляхом вдосконалення процесів. Це частина дисципліни управління бізнес-процесами (BPM). Оптимізовані процеси ведуть до оптимізації бізнес-цілей [8].

На сьогоднішній день автоматизація БП виступає невіддільною складовою середнього та великого бізнесу. Автоматизація бізнес-процесів представляє собою заміну рутинної мануальної праці співробітників набором

алгоритмів та сервісів, які використовуються для досягнення важливих операційних та фінансових цілей шляхом впровадження спеціальних систем для управління підприємством. Автоматизація є важливою частиною керування бізнес-процесами тому, що дозволяє оптимізувати та координувати їх [5, 9].

За допомогою делегування трудомістких задач інформаційній системі оптимізації бізнес-процесів дозволить досягти ефективності (рис. 1.1).

Впровадження комплексу автоматизації відкриває для працівників ресторану всіх рівнів багато нових можливостей, які змінюють загальний стиль роботи закладу. Автоматизація бізнес-процесів є життєво важливою для підвищення продуктивності та швидкості роботи організації. Автоматизація полегшує аналіз можливостей і переваг основних бізнес-процесів, які виконуються всередині організації, і дозволяє впроваджувати та оптимізувати відповідні технології.



Рисунок 1.1 – Переваги автоматизації бізнес-процесів підприємства [10,11]

На думку науковців, автоматизована система управління являє собою цілісний програмно апаратний комплекс, який дає змогу задовольнити як поточні, так і стратегічні потреби підприємства в опрацюванні даних. Вимога автоматизації всіх цих процесів виникає перш за все через необхідність врахування великої кількості деталей; неможливість постійного наочного контролю виконання кожного технологічного процесу керівниками; необхідність швидкого прийняття рішень і високої мобільності людських кадрів [12].

Ресторанські підприємства часто стикаються з проблемами, які перешкоджають їм ефективно працювати. Тим не менш, впровадження автоматизованих управлінських систем може допомогти вирішити проблеми, пов'язані з витратами. цв

Автоматизацію бізнес процесів закладу харчування можна розділити на декілька груп, а саме:

- адміністратор;
- офіціант;
- повар.

Кожна група має свої бізнес-процеси, які необхідно оптимізувати задля покращення якості роботи закладу в цілому. Для адміністратора існують бізнес-процеси, зокрема, це все, що пов'язане з аналітикою або керування закладу: аналітика різноманітних даних, можливість швидкої зміни параметрів таблиць пов'язаних з фінансами, зокрема це меню, склад, персонал [5].

Для офіціантів важливою частиною оптимізації бізнес-процесів є автоматизація оформлення замовлень, це дозволить зменшити час на обслуговування клієнтів а також зведе до мінімуму можливість людської помилки при оформленні замовлень.

Блок повара надає можливість слідкувати за вхідними замовленнями у реальному часі, а також відстежувати стадію приготування страв, що значно полегшує роботу персоналу [5].

Таким чином, застосування інформаційних технологій у ресторанному бізнесі дозволяє гостям ресторану значно скоротити час, який їм потрібен, дозволяє ефективно відстежувати роботу підприємства, аналізувати отримані дані та своєчасно реагувати на зміни в соціально-економічному середовищі та конкурентному середовищі ресторанного бізнесу. Інформаційні технології значно підвищують економічну ефективність підприємств ресторанного господарства.

## 1.2 Опис проблематики створення системи оптимізації бізнес-процесів закладу харчування

Через велику кількість ресторанів і кафе у світі існує сильна конкуренція. Це підвищує очікування клієнтів щодо якості та швидкості обслуговування. Багато компаній все ще використовують традиційні методи управління, але це може призвести до неефективного керівництва. Основною проблемою є впровадження АСУ для часткового або повного вдосконалення існуючих бізнес-процесів організації.

Рисунок 1.2 надає пояснення причин і наслідків неефективного стилю керування рестораном.

Перехід від традиційного стилю керування до використання новітніх технологій і інформаційних систем викликає певні труднощі. Можно припустити, що цими причинами є обмежений бюджет і ризики, пов'язані зі зміною методів управління компанією.

З іншого боку, впровадження таких технологій гарантує покращення якості обслуговування, збільшення прибутку та зниження витрат. Крім того, системи моніторингу дозволяють робити правильні висновки з роботи, що покращить роботу підприємства в цілому та його конкурентоспроможність.



Рисунок 1.2 – Проблеми, що призводять до неефективної діяльності ресторану [13]

Розроблювальний застосунок повинен вирішувати основні проблеми, з якими стикаються заклади харчування, які керуються традиційними методами управління, щоб вони могли легко перейти на новітні методи управління, коли розробляють інформаційну систему, спрямовану на оптимізацію бізнес-процесів закладу [5].

### 1.3 Визначення сфери застосування розроблювальної системи

Сфера застосування інформаційної системи для оптимізації закладів харчування доволі велика так, як подібний програмний засіб (ПЗ) підходить

для багатьох видів закладу харчування. Розроблювальна система повинна працювати в усіх видах закладів харчування, які виконують функції оформлення замовлення та подачу їжі у зал за допомогою офіціантів та/або бару. Завдяки тому, що розроблювальна система бути представлена у вигляді веб-застосунку, вона підходить будь-якому підприємцю у якого є гаджет з виходом у мережу Інтернет. Розроблювальний веб-застосунок буде орієнтований на персонал закладу та власника.

#### 1.4 Дослідження існуючих програмних продуктів у сфері оптимізації бізнес-процесів для закладів харчування

Для того, щоб розробити ефективну систему для оптимізації бізнес-процесів закладу харчування необхідно провести дослідження існуючих програмних продуктів. Варто зазначити, що більшість існуючих ПЗ розроблені або мають в собі інтеграції застосунків іноземного виробництва, тому виникає необхідність розробляти власні системи для оптимізації бізнес-процесів.

В рамках цього дослідження було відібрано для аналізу чотирьох аналоги з метою з'ясування їх можливостей, недоліків та переваг (табл. 1.1).

Таблиця 1.1 – Порівняння аналогічних системи

Система	Переваги	Недоліки
Waiterio	– простий у використанні; – широкі можливості налаштування.	– відсутність деяких додаткових функцій; – не завжди стабільна робота; – відсутність аналітики; – підходить тільки для обробки замовлень; – поганий інтерфейс для замовлень на винесення.

Продовження табл. 1.1

Система	Переваги	Недоліки
Парус-Ресторан	<ul style="list-style-type: none"> <li>– великий функціонал;</li> <li>– інтеграція з обліковою системою.</li> </ul>	<ul style="list-style-type: none"> <li>– складний та не зрозумілий інтерфейс;</li> <li>– довгий термін вивчення;</li> <li>– висока вартість підтримки;</li> <li>– відсутність веб форми або мобільного додатку.</li> </ul>
LIBRO	<ul style="list-style-type: none"> <li>– багатий функціонал;</li> <li>– доступні ціни.</li> </ul>	<ul style="list-style-type: none"> <li>– обмежені можливості;</li> <li>– складність вивчення у користуванні;</li> <li>– незрозумілий інтерфейс;</li> <li>– відсутність підтримки;</li> <li>– обмежена кількість користувачів.</li> </ul>
Microinvest	<ul style="list-style-type: none"> <li>– доступні ціни;</li> <li>– висока продуктивність.</li> </ul>	<ul style="list-style-type: none"> <li>– складність вивчення;</li> <li>– відсутність звітності та аналітики;</li> <li>– відсутність веб чи мобільного додатку.</li> </ul>

Дослідження існуючих програмних продуктів для автоматизації керування закладами харчування показало, що кожна система мала як переваги, так і недоліки. Більшість з цих систем пропонують багато можливостей для оптимізації бізнес-процесів, але вони мають деякі недоліки, такі як важка інструкція користувача, незрозумілий інтерфейс і високі витрати на підтримку. Крім того, можна зробити висновок, що для ефективного оптимізації всіх бізнес-процесів ресторану необхідно використовувати кілька ПЗ, спеціально призначених для цього. Це, у свою чергу, призведе до збільшення витрат ресторану [5].

Враховуючи ці обмеження, було вирішено розробити нової інформаційної системи, яка б усунула деякі недоліки та перейняла б переваги існуючих програмних продуктів, та надала б закладам харчування зручний та інтуїтивно зрозумілий інтерфейс для зручного користування з мінімальним часом навчання. Розроблювальна система буде спрямована на оптимізацію процесів замовлення, обліку, управління персоналом, а також важливої частини аналітики та ABC-аналізу, тощо. Можна зазначити, що саме у

жодної з порівняних систем не було присутнього АВС-аналізу, який відіграє важливу роль у оптимізації роботи закладу харчування.

На відміну від основної маси подібних застосунків, розроблювальна система буде представлена у вигляді веб-застосунку, а не мобільної чи комп'ютерної програми, що збільшить можливість користування нею за допомогою будь-якого пристрою, де є вихід до мережі Інтернет, а також зменшить витрати для закладів харчування [5].

### 1.5 Висновки до розділу

У цьому розділі було проаналізовано поточний стан бізнес-процесів у закладах харчування, і було виявлено, що традиційні підходи до керування підприємствами мають бути кинуті, а системи автоматизації управління повинні бути введені, щоб заклади харчування працювали якнайкраще.

Як наслідок того, що автоматизація підприємства ресторанного господарства значно спрощує процес роботи та допомагає звести до мінімуму збитки та зайві витрати, стало очевидним, що вона є необхідною і необхідною. Автоматизація зменшує тимчасові витрати на пошук і аналіз інформації, підвищує якість сервісу та престиж закладу, забезпечує швидку обробку замовлень і обслуговування відвідувачів, покращує узгодженість і контроль за роботою персоналу та забезпечує облік руху матеріальних цінностей і товарів.

А також, провели аналіз існуючих аналогів, виділили їх переваги та недоліки, які будуть враховані під час написання наступних розділів. Враховуючи інформацію, отриману з проведеного дослідження було вирішено розробляти інформаційну систему у вигляді веб-застосунку. Це дозволить зменшити витрати на розробку всієї системи загалом, а також витрати для підприємств для обслуговування такої системи та збільшить можливість використання системи на будь-яких пристроях.

## 2 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

### 2.1 Опис засобів розробки та постановка задачі

Метою атестаційної роботи є розробка інформаційних системи оптимізації бізнес-процесів закладу харчування у вигляді веб-застосунку, яка повинна автоматизувати роботу закладів харчування, збільшити ефективність управління, та якості обслуговування клієнтів, а також покращити внутрішні бізнес-потоки. Веб-застосунок, що розроблюється призначений для управління персоналом, аналітики та звітності за роботою закладу в цілому, створення та редагування меню, оформлення замовлення, тощо. Основне завдання розробити інтуїтивно зрозумілий та простий у використанні інтерфейс, що не потребує часу на навчання користування та виконати автоматизацію основних важливих бізнес-процесів для будь-якого закладу харчування – оформлення замовлення та різних видів аналітики та звітів для своєчасного прийняття правильних рішень з метою збільшення ефективності роботи закладу [5, 14].

Для реалізації поставлених цілей необхідно виконати наступні завдання:

- проаналізувати предметну область;
- провести аналіз стану існуючих бізнес-процесів у закладах харчування та виявити ті, які необхідно буде оптимізувати;
- провести дослідження аналогів систем;
- виявити переваги та недоліки аналогів та прийняти їх до уваги;
- визначити вимоги для інтерфейсу, враховуючи дослідження аналогів;
- проаналізувати мови програмування для розробки веб-застосунків та визначити необхідну;
- розробити базу даних для інформаційної системи;

- розробити інформаційну систему оптимізації бізнес-процесів закладу харчування, враховуючи вище перераховане;
- провести тестування розробленої системи.

Для виконання поставлених цілей необхідно розробити веб-сайт, що відповідає темі роботи. Реалізацію веб-сайту можна поділити етапі що зображені на рисунку 2.1.



Рисунок 2.1 – Етапи розробки веб-сайту [14]

У рамках цих етапів необхідно чітко визначити цілі та основні цілі розробки веб-сайту. Після цього необхідно створити технічне завдання. На наступному етапі необхідно розробити дизайн, який відповідає поставленим завданням. Після завершення цих етапів необхідно розробити код front-end і back-end, щоб веб-сайт міг працювати. Після того, як код і верстка сайту були написані та завершені, необхідно завершити наповнення контентом і тестування. На останньому етапі необхідно підтримувати та розвивати продукт, який було розроблено [14].

У рамках даного атестаційного проекту було здійснено вибір двох інтегрованих середовищ розробки (IDE) – Visual Studio Code та Visual Studio, як основних інструментів для написання та рефакторингу коду. Вибір цих середовищ був зумовлений їх широкими можливостями, ефективністю та

гнучкістю в різних аспектах розробки програмного забезпечення.

У контексті розробки back-end частини для веб-сайту на базі ASP.NET Core з використанням Entity Framework, було вирішено застосувати інтегроване середовище розробки (IDE) Visual Studio (рис. 2.2). Цей вибір зумовлений комплексною підтримкою середовища необхідними інструментами та бібліотеками для ефективної реалізації API для веб-сайту.

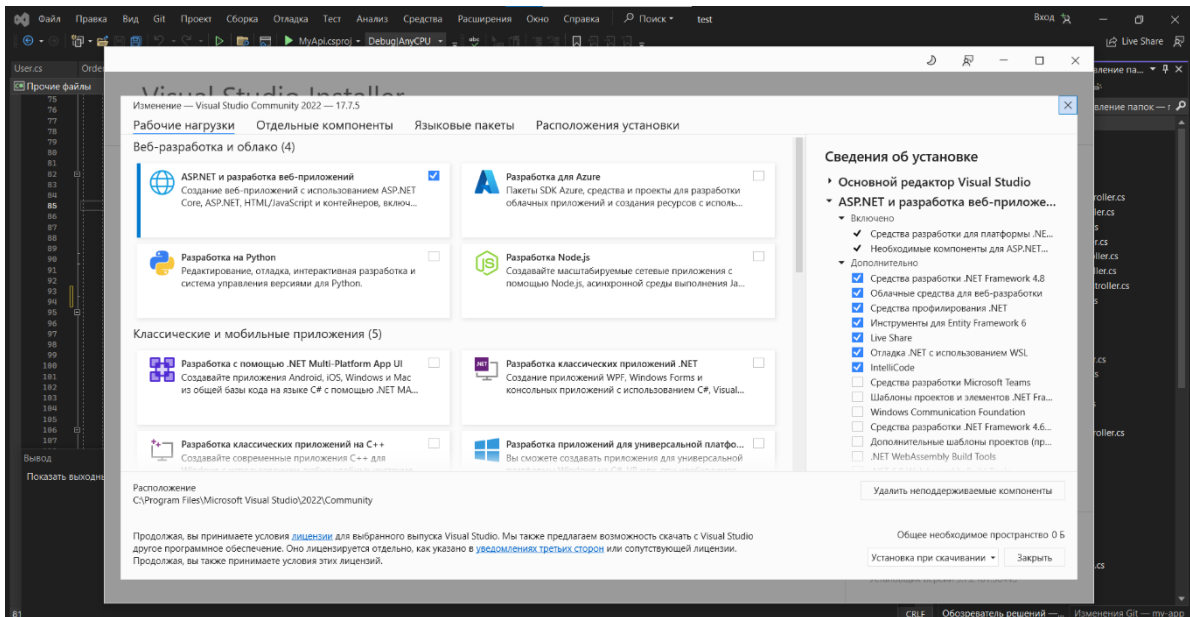


Рисунок 2.2 – Вікно інсталяції модулів Visual Studio

Для розробки вказаного проєкту були встановлені наступні компоненти та бібліотеки:

- SDK.NET Core. Комплект інструментів для розробки та запуску програм на основі.NET Core, включаючи компілятори, бібліотеки та рантайм.
- ASP.NET Core Framework. Потужний, модульний фреймворк для створення веб-додатків і API, оптимізований для хмарних рішень.
- основа Entity Framework орієнтований на об'єкти фреймворк доступу до даних дозволяє розробникам працювати з базою даних за допомогою об'єктів.NET, що забезпечує високу абстракцію та ефективність.
- управління пакетами NuGet, інструменти для розробки баз даних; система управління пакетами для.NET, яка дозволяє легко інтегрувати та

керувати зовнішніми бібліотеками та залежностями в проєкті;

– можна інтегруватися з багатьма системами управління базами даних (СУБД), включаючи SQL Server, для виконання операцій, таких як міграція баз даних, управління схемами та інші.

Вирішено використовувати Visual Studio Code (VS Code) як основне середовище розробки для створення фронтенду веб-сайту. Основними перевагами VS Code є його гнучкість, легкість і розширюваність, які є важливими для сучасної фронтенд-розробки.

Для ефективної роботи над фронтенд частиною проєкту були встановлені та налаштовані необхідні компоненти та розширення VS Code. Ці компоненти будуть визначені під час вибору мови програмування.

## 2.2 Формування технічного завдання до розроблюваного додатку

Для якісного виконання роботи необхідно встановити системні та функціональні вимоги до розроблювальної інформаційної системи оптимізації бізнес-процесі закладу харчування.

### 2.2.1 Системні вимоги

Система, що розроблюється для ефективної роботи повинна бути оптимізована для усіх користувачів, для безперервної та якісної роботи. Для виконання поставлених задач було вирішено розроблювати систему у вигляді веб-застосунку, виходячи з цього можна поставити наступні вимоги:

- серверна частина інформаційної системи замовлення їжі з ресторану повинна бути виконана на основі бази даних для зберігання інформації;
- реалізований веб-застосунок повинен бути доступним на будь-якому пристрої, що має вихід до мережі Інтернет;
- система повинна мати добре реалізовану систему безпеки, та мати надійне зберігання інформації про користувачів, а також унеможливити потрапляння до веб-застосунку неавторизованих користувачів;

– інтерфейс всіх сторінок повинен бути простим для вивчення за допомогою керівництва користувача, а також інтуїтивно зрозумілим для швидкого навчання користування системою.

### 2.2.2 Функціональні вимоги

У сучасному світі автоматизація бізнес-процесів є життєво важливою для підвищення ефективності та конкурентоспроможності підприємств, зокрема в галузі громадського харчування. Розробка інформаційної системи для оптимізації бізнес-процесів у ресторані вимагає визначення функціональних вимог, які відображають основні функції та цілі системи.

Для того, щоб визначити функціональні вимоги було розроблено функціональна модель системи з застосуванням стандарту IDEF0.

IDEF0 – нотація графічного моделювання, яка використовується для створення функціональної моделі, відображає структуру і функції системи, а також потоки інформації і матеріальних об'єктів, які зв'язують ці функції [15]. Загальний вигляд діаграми представлено на рис. 2.3.

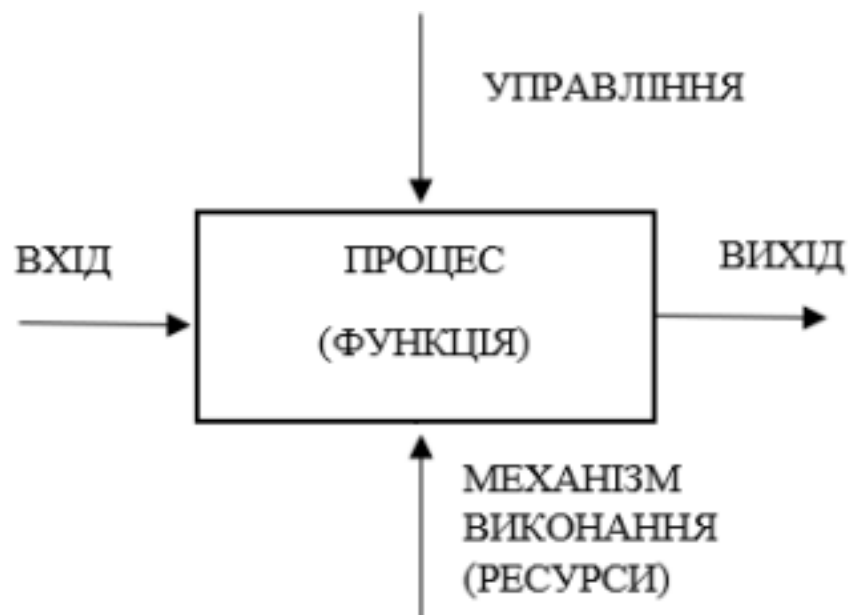


Рисунок 2.3 – Загальний вигляд контекстної діаграми IDEF0

На діаграмі показані основні компоненти функційної моделі:

- вхід представляє дані або інформацію, які входять у процес ззовні та впливають на його хід;
- процес або функція є центральним блоком, який описує дії або перетворення, які відбуваються в системі. Він обробляє вхідні дані для отримання результатів.;
- вихід – це дані або результати, які створює процес і передаються зовні або наступним етапам системи;
- управління – це вказівки або обмеження, які керують процесом. Це може бути механізмом виконання, який охоплює ресурси або механізми, які використовуються для виконання процесу.

Кожен окремий елемент моделі IDEF0 може бути предметом подальшого аналізу на окремій діаграмі. У межах цієї методології кожен блок трактується як самостійна одиниця з визначеними характеристиками, і процес розділення цієї одиниці на її складові елементи (блоки та потоки, які формують діаграму) відомий як процес декомпозиції [16]

Декомпозиційний аналіз контекстної діаграми, який показує ієрархічну структуру взаємопов'язаних діаграм системи, показано на рис. 2.4. У цьому зображенні показано структурований розклад системи на більш деталізовані частини, що дозволяє краще зрозуміти окремі функціональні елементи та те, як вони взаємодіють у загальній архітектурі. На верхньому рівні контекстна діаграма надає загальний огляд системи, який включає зовнішні суб'єкти або акторів, а також те, як вони взаємодіють з основними процесами системи.

В подальшому процесі декомпозиції кожен компонент цієї основної діаграми представлений у вигляді окремих, більш детальних діаграм, які підкреслюють внутрішні процеси та механізми, які забезпечують функціонування системи. Завдяки цьому методу можна провести ретельний аналіз і моделювання системи на різних рівнях абстракції, що сприяє кращому проектуванню та оптимізації системних процесів.

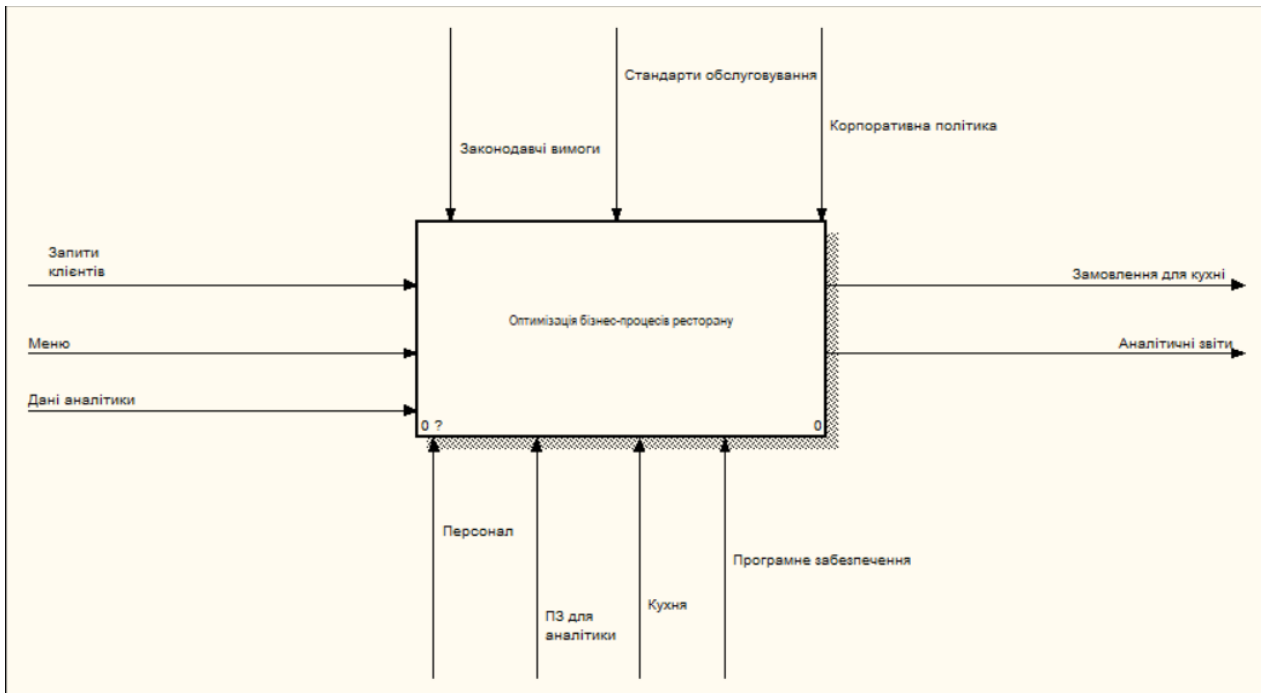


Рисунок 2.4 – Контекстна діаграма системи

На рисунку 2.5 представлена декомпозицію контекстної діаграми, яка представляє ієрархічну модель окремих діаграм системи.

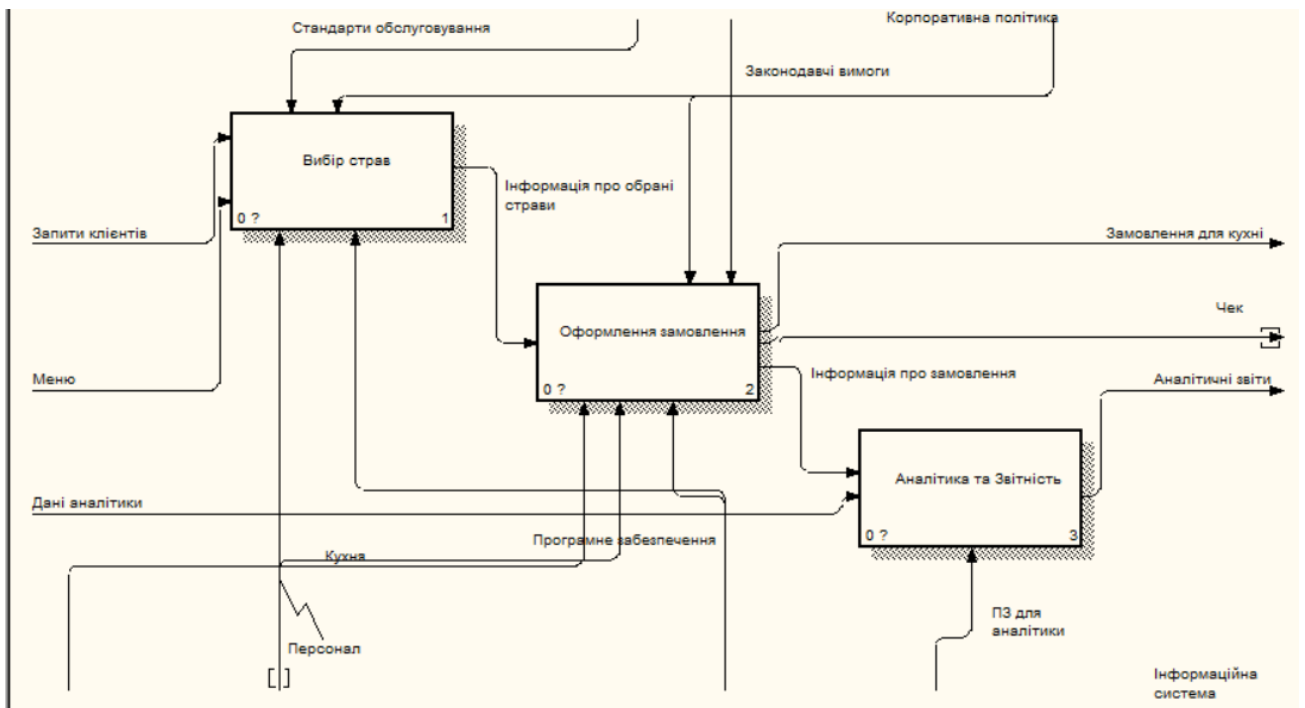


Рисунок 2.5 – Діаграма декомпозиції

Рисунок 2.6 відображає декомпозицію функції «Вибір страв» системи.

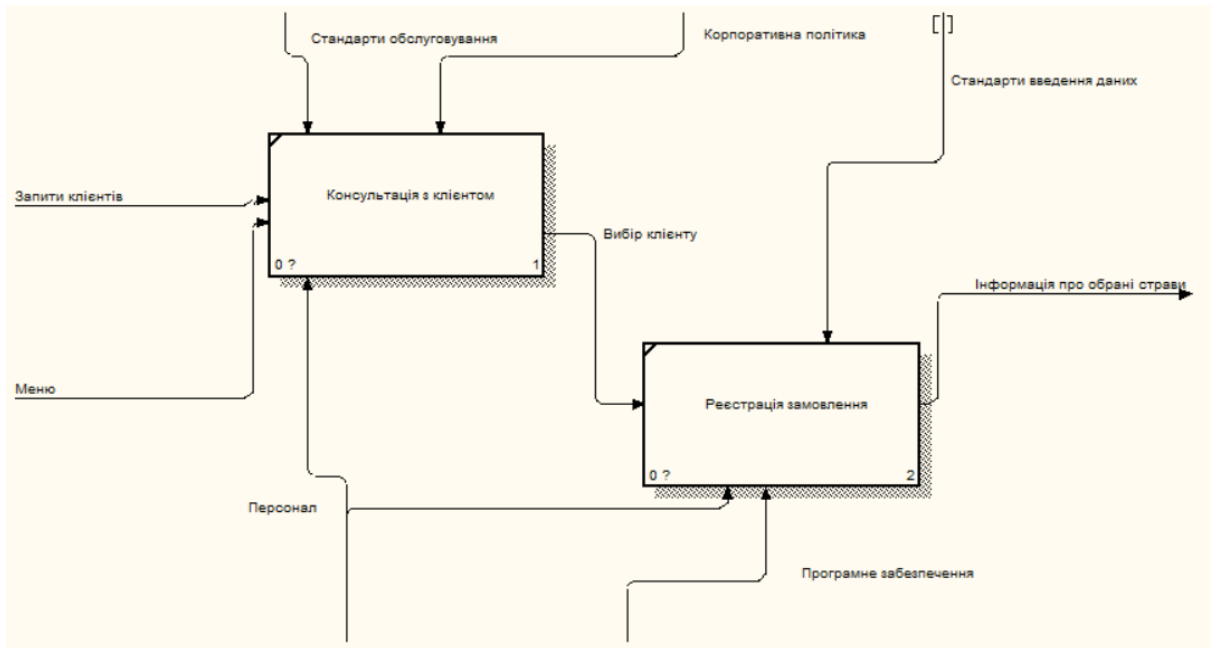


Рисунок 2.6 – Діаграма декомпозиції функції «Вибір страв» системи

Рисунок 2.7 відображає декомпозицію функції «Оформлення замовлення» системи.

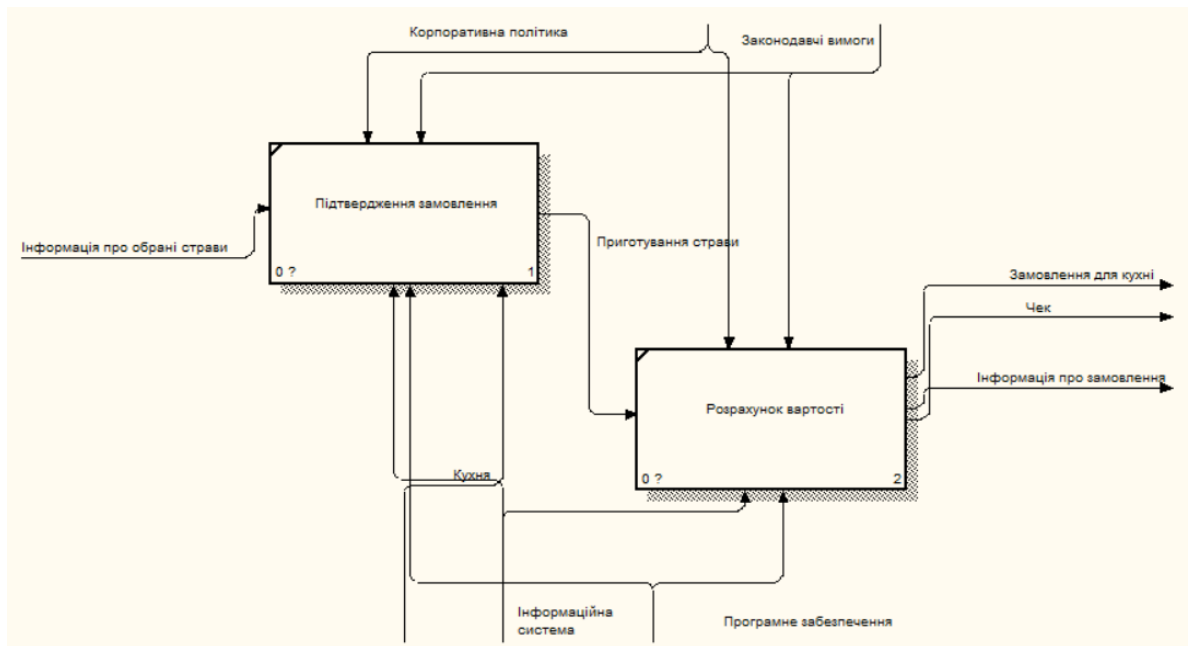


Рисунок 2.7 – Діаграма декомпозиція функції «Оформлення замовлення» системи

Рисунок 2.8 відображає декомпозицію функції «Аналітика та звітність» системи.

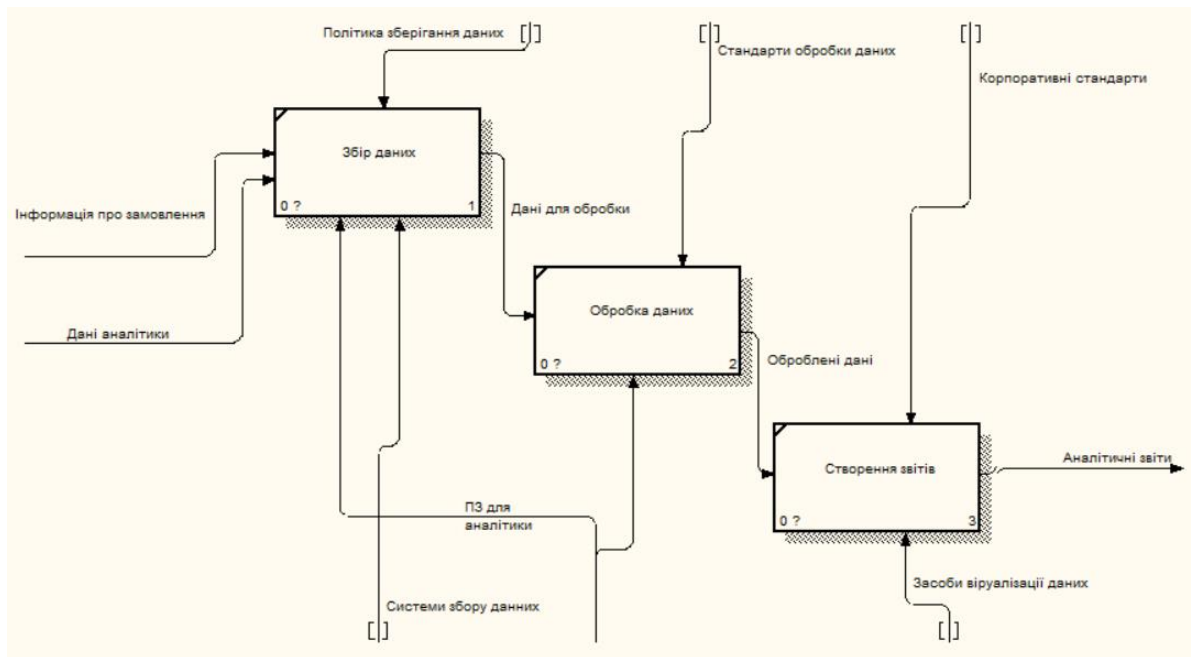


Рисунок 2.8 – Діаграма декомпозиція функції «Аналітика та звітність» системи

Таким чином, виходячи з розробленої IDEF0 діаграми можна визначити ключові функціональні вимоги до розроблювальної системи:

- автоматизація процесу прийому та обробки замовлень, яка спрощує взаємодію між клієнтами та персоналом закладу;
- управління запасами продуктів, що включає в себе перегляд та додавання нових продуктів на склад, а також моніторинг стану придатності;
- моніторинг та аналіз продажів та фінансових показників;
- управління персоналом;
- генерація аналітики та звітів для аналізу та оцінки ефективності бізнес-процесів, що дозволить швидко та ефективно приймати обґрунтовані рішення.

Ці функціональні вимоги служать основою для розробки високоефективної системи, яка відповідатиме актуальним потребам закладу харчування та сприятиме його стабільному розвитку у динамічному бізнес-

середовищі. Також важливою функцією є наявність ABC-аналізу для моніторингу стану.

ABC-аналіз – метод, за допомогою якого визначають розподіл конкретного параметра між окремими елементами певної множини. Вважається, що відносно невелика кількість товарів займає велику частку у загальній вартості усіх товарів. В основі методу ABC лежить так зване правило Паретто, згідно з яким сукупність будь-яких об'єктів можна поділити на дві неоднакові частини у співвідношенні 80/20. Метод ABC, який використовується у логістиці запасів, передбачає поділ на три частини. Методика проведення ABC-аналізу для управління матеріальними запасами має на меті нормування і контроль стану товарно-матеріальних цінностей, розбитих на три підмножини А, В і С [17].

Щодо кожної окремої групи, то підприємству слід дотримуватися наступних рекомендацій щодо управління ними [18]:

- для групи А необхідно створювати значний страховий запас за даною складовою асортименту. Вона потребує детального планування, постійного обліку та контролю;
- об'єкти групи В у меншій мірі важливі для підприємства і вимагають звичайного контролю та налагодженого обліку. Для них застосовуються ті ж заходи, що і для категорії А, але вони здійснюються рідше і з великими прийнятими допусками;
- до замовлення товарної групи С необхідно підходити з обережністю та організовувати заходи зі стимулювання збуту.

Також, важливою функцією є прогнозування попиту, це можливо зробити за допомогою простого експоненціального згладжування. За допомогою прогнозування попиту страв, за допомогою інформації про кількість проданих страв у останній відомий попит та попередній прогноз можна визначити наступний попит.

Цей метод прогнозування найбільш широко використовується з усіх методів прогнозування. Це вимагає невеликих обчислень. Цей метод

використовується, коли шаблон даних є приблизно горизонтальний (тобто немає ні циклічної зміни, ні виражена тенденція в історичних даних) [19].

Формально проста експонента рівняння згладжування набуває вигляду:

$$S_t = \alpha y_{t-1} + (1 - \alpha) \times S_{t-1},$$

де  $y_{t-1}$  – останній відомий попит на страву;

$S_{t-1}$  – попередній прогноз;

$\alpha$  – коефіцієнт, що характеризує величину згладжування,  $0 < \alpha < 1$ .

### 2.2.3 Діаграма варіантів використання системи

Діаграма варіантів використання — це візуальний інструмент, який використовується для опису поведінки системи шляхом ілюстрації того, як користувачі взаємодіють з нею. Цей тип діаграм є частиною Уніфікованої мови моделювання (UML), яка є стандартним способом візуалізації дизайну системи [20].

Ця діаграма варіантів використання надає візуальне представлення взаємодії системи із зовнішнім світом. Він описує різні сценарії, у яких використовується система, включаючи завдання, служби та функції, які вона виконує.

А також, служить основною формою визначення вимог до системи чи програмного забезпечення, зосереджуючись на тому, «що» повинна робити система, а не на тому, «як» вона має це робити. Це розрізнення має вирішальне значення, оскільки дозволяє чітко повідомити заплановану функціональність системи без приписування технічних деталей її реалізації [21].

Рисунок 2.9 ілюструє діаграму варіантів використання, яка забезпечує графічне відображення потенційних функціональних можливостей системи, що знаходиться у стадії розробки.



Рисунок 2.9 – Діаграма варіантів використання

### 2.3 Вибір засобів реалізації

Перед початком розробки інформаційної системи найважливішим є правильний вибір засобів реалізацій та інструментів, що визначають, як шляхи для реалізації проєкту, так і напряду впливають на його гнучкість, ефективність, легкість, тощо. Які саме обрати засоби реалізації визначається факторами, що впливають на розробку, такі як специфіка проєкту, доступні ресурси, майбутній розвиток та підтримка, а також, головне, це потреби цільового користувача, які визначають функціональні вимоги.

Важливим пунктом під час розробки інформаційної системи є чітке розмежування проєкту на бекенд та фронтенд компоненти, що сильно впливає на гнучкість роботи та оптимізації робочого процесу.

Frontend – це публічна частина веб-застосунків, з якою користувач може взаємодіяти і контактувати безпосередньо. У frontend входить відображення функціональних завдань, інтерфейсу користувача, що виконуються на стороні клієнта, а також обробка призначених для користувача запитів. Основна задача frontend полягає візуальному представленні даних, забезпечення користувацького інтерфейсу, а також зв'язок з backend [22].

Backend, або серверна частина, включає в себе логіку обробки даних, їх зберігання та управління. Розробка бекенду фокусується на створенні та підтримці баз даних, серверної логіки, API для взаємодії з фронтендом, та забезпеченні безпеки інформації [23].

Отже, поділ розробки веб-застосунку на frontend та backend забезпечить більшу гнучкість в процесі розробки, так як дозволить працювати та оновлювати їх незалежно один від одного, що призведе до підвищення продуктивності та якості роботи. Необхідно обрати мови програмування для frontend та backend компонентів.

### 2.3.1 Вибір мов програмування

Вдалих вибір мов програмування для розробки інформаційної системи є ключовим рішенням, котре напряду впливає на легкість розробки та обслуговування, архітектуру проєкту, ефективність роботи, а також можливість масштабування проєкту надалі. Яку саме мову програмування обрати залежить від поставленої мети та задач системи, вдалих вибір мови для фронтенд та бекенд частини залежить від ряду факторів, таких як вимоги функціональних можливостей, специфіка проєкту, наявність ком'юніті підтримки та наявний досвід програмування у розробників.

При виборі мови для backend частини необхідно враховувати наступні

фактори:

- продуктивність і масштабування;
- безпека;
- бібліотеки та фреймворки;
- управління базами даних;
- інтеграція з іншими системами.

З іншого боку, для frontend важливі наступні характеристики:

- висока продуктивність;
- масштабованість;
- безпека;
- зручність використання;
- екосистема та спільнота;
- адаптивність і швидкість.

У сучасному світі, де технології та мови програмування постійно розвиваються, існує значна різноманітність мов для розробки як фронтенд, так і бекенд частини застосунків.

Згідно з опитування, яке провели спільнота програмістів DOU демонструє рейтинг мов програмування (рис. 2.10 – 2.11).

Проаналізувавши опитування, що провели спільнота програмістів DOU [24] було вирішено обрати для роботи з backend частиною – C# так як це одна з найпопулярніших та швидко розвиваючих мов, що має велику кількість документації та при навчанні в університеті був досвід праці саме з цією мовою, а для frontend – JavaScript, бо це найпопулярніша мова для роботи з фронтендом, а також має досить велике ком'юніті та легкий вхід.

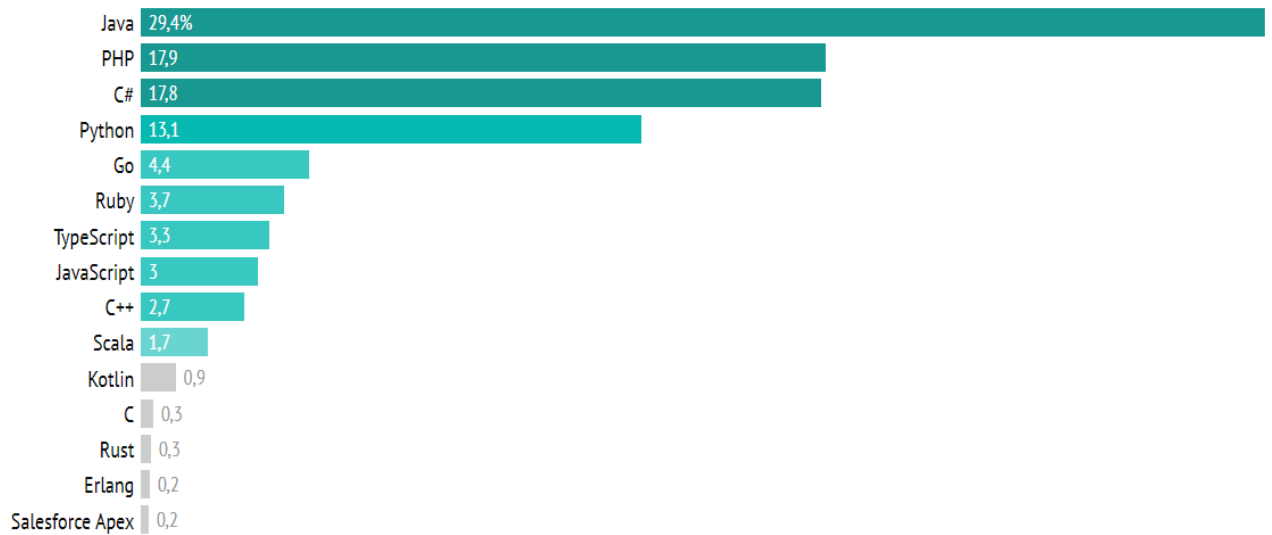


Рисунок 2.10 – Рейтинг мов програмування backend



Рисунок 2.11 – Рейтинг мов програмування frontend

Також, згідно їх дослідження, найпопулярніших фреймворків для JavaScript (рис. 2.12) було обрано для роботи – React.

C# – це сучасна об'єктно-орієнтована мова програмування загального призначення, розроблена компанією Microsoft. Вона була розроблена, щоб бути простою, ефективною та легкою у вивченні. C# широко використовується для розробки десктопних і веб-додатків, ігор та мобільних додатків. Деякі з переваг використання C# для ООП включають безумовно легкість у вивченні C#, бо він має простий синтаксис, який легко вивчити особливо для розробників з досвідом роботи з Java або C++. Надійність C# – це сильно типізована мова, яка забезпечує безпеку типів, запобігаючи поширеним помилкам, таким як виключення за нульовим посиланням. Ефективність C# компілюється в машинний код, що робить її швидшою та ефективнішою, ніж інтерпретовані мови, такі як Python. Інтеграція з фреймворком .NET C# легко інтегрується з фреймворком .NET, надаючи розробникам широкий спектр бібліотек та інструментів для створення програмних додатків [25].

JavaScript – мультипарадигмова мова програмування. Це універсальна мова, що дозволяє використовувати кілька парадигм та програмувати у різних стилях. Наприклад, об'єктноорієнтований стиль, де саме об'єкти, а не функції, є основним будівельним блоком, або процедурний стиль програмування, як в мові C, де функції є набором інструкцій, який може бути викликано з іншого місця в програмі.

React (старі назви: React.js, ReactJS) — відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту вебсторінки, з якими стикаються в розробці односторінкових застосунків. Розробляється Meta (раніше Facebook) і спільнотою індивідуальних розробників [26].

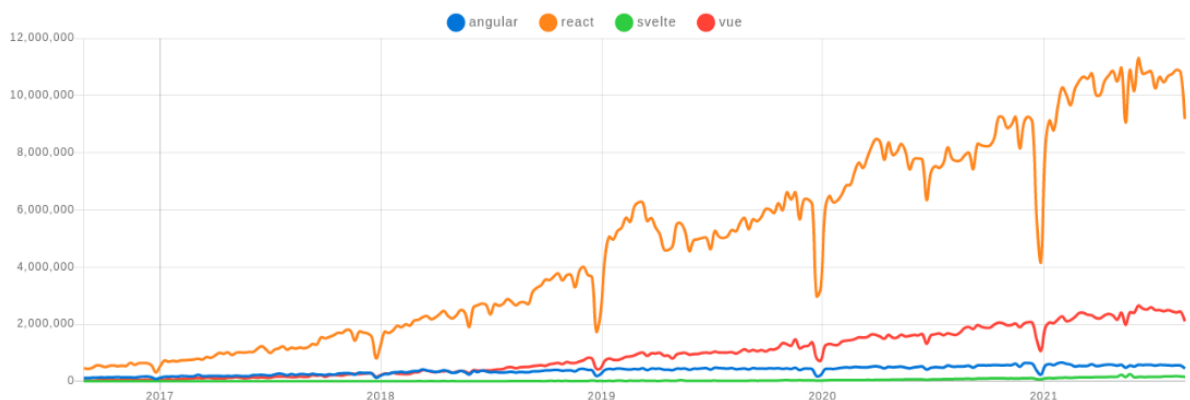


Рисунок 2.12 – Рейтинг фреймворків JavaScript

Розробка бекенду застосунку передбачає створення API, який слугує мостом для взаємодії між бекендом і фронтендом.

Інтерфейс прикладного програмування (API) – це спосіб взаємодії двох чи більше комп'ютерних програм одна з одною. Це різновид програмного інтерфейсу, який пропонує послуги іншим частинам програмного забезпечення [27].

### 2.3.2 Вибір середовища розробки та СУБД

У процесі розробки веб-застосунків вибір середовища розробки та

СУБД є критичними рішеннями, які впливають на продуктивність, ефективність робочого процесу, а також на легкість подальшого масштабування та підтримки проєкту. Вибір SQL Server Management Studio як основного інструменту для роботи з базами даних зумовлений його високою функціональністю та зручністю в управлінні даними.

SQL Server Management Studio – це високопродуктивне та потужне інтегроване середовище, розроблене Microsoft для роботи з базами даних, спеціально зорієнтоване на обслуговування Microsoft SQL Server. Це корисний інструмент для адміністраторів баз даних, розробників та аналітиків, які працюють з SQL Server [28].

Нижче наведено перелік основних функцій SQL Server Management Studio, які надають переваги та можливості:

- інтуїтивно зрозумілий інтерфейс;
- універсальність;
- підтримка скриптів;
- безпека;
- потужні інструменти для адміністрування;
- інтеграція з іншими інструментами.

Завдяки проведеному аналізу недоліків та переваг, а також функціональним можливостям для розробки бази даних було обрано SSMS, що підходить для реалізації визначеної мети проєкту. Рішення обрати саме це середовище для роботи з базами даних зумовлене інтегрованим набором функцій, завдяки яким підвищується ефективність розробки системи, а також оптимізація та швидкодія бази даних. SSMS має необхідні інструменти для моніторингу, адміністрування, безпеки даних, також має високу функціональність та зручне управління даними, через це дане середовище є оптимальним вибором для розробки надійних та ефективних баз даних.

## 2.4 Висновки до розділу

У цьому розділі було проведено аналіз технічного завдання та поставлена задача на роботу з проектом, було вирішено, що роботу з розробки інформаційної системи необхідно розділити на сім етапів, що включають в себе: підготовку до роботи, проектування, розробка дизайну, верстку, програмування сайту, наповнення контентом та тестування, а також додаткові роботи. Було вирішено, що є необхідність розділяти проєкт на фронтенд та бекенд для зручнішої та роботи, що в свою чергу дало можливість визначитись з засобами реалізації для цих компонентів, а також для проектування та розробки бази даних.

Завдяки розроблених IDEF0 діаграм було визначено головні функціональні вимоги. Розроблена діаграма варіантів використання системи, що було розроблена також у цьому розділі, надала візуальне представлення взаємодії користувачів з системою.

## 3 РОЗРОБКА ТА ТЕСТУВАННЯ ДОДАТКУ

### 3.1 Опис структури інформаційної системи

Структура інформаційної системи, що розроблюється повинна базуватись на клієнт-серверній архітектурі, вона одна з найпоширеніших та найефективніших архітектур для веб-сайтів. Клієнтом вважається локальний комп'ютер віртуального споживача, який виконує запити на сервер, а сервером віртуальна або персональна електронна машина з розподілюваною пам'яттю, обробленням даних, комунікаційними засобами та засобами управління периферійним обладнанням, що обробляє запити клієнтів, надає доступ до певних системних ресурсів, зберігає дані та взаємодіє з базами даних [28].

Основні переваги клієнт-серверної архітектури:

- відповідальність розділена між пристроями користувача (клієнт) та сервером;
- сервери та клієнти можуть бути різними за ОС та типом, ця архітектура дозволяє легку взаємодію між різними пристроями та системами;
- централізоване управління даними;
- безпека.

Використання саме цієї архітектури (клієнт-сервер) було обрано, виходячи з переваг, що описані вище. Ця архітектура обслуговує велику кількість користувачів і запитів, забезпечуючи ефективне розподілення ресурсів і високу доступність. Крім того, це дозволяє гнучке оновлення та розвиток системи, що необхідно для сучасних динамічних і розширюваних веб-додатків.

Параметри, наведені нижче, реалізуються на стороні сервера:

- зберігання;
- передача;

- резервне копіювання та захист даних;
- обробка запитів клієнтів;
- надання відповідей клієнтам.

Параметри, що наведено нижче, реалізуються на стороні клієнта:

- формування запиту до сервера;
- відправлення запиту на сервер;
- отримання відповідей на відправлені запити.

Рисунок 3.1 демонструє структуру клієнт-серверної архітектури.



Рисунок 3.1 – Структура клієнт-серверної архітектури

### 3.2 Розробка інтерфейсу програмного засобу

Проектування та розробка інтерфейсу користувача важливий етап під час створення інформаційної системи для повноцінної роботи веб сайту, так як інтерфейс являє собою проміжну ланку сервера та клієнта, що дозволяє виконувати всі функціональні можливості системи, для цього необхідно, щоб користувацький інтерфейс був ефективним, привабливим для ока та найважливіше інтуїтивно зрозумілим користувачу [29].

Основною метою розробки ІК є забезпечення високого рівня зручності користування, зниження когнітивного навантаження на користувача та оптимізація процесу взаємодії з програмним засобом. Задачами є:

- створення логічної та послідовної структури навігації;
- розробка чіткого та зрозумілого дизайну елементів управління;
- адаптованість до мобільних пристроїв.

Основними принципами, які використовувалися під час розробки

інтерфейсу, були такі:

- єдність стилістики та повторюваність елементів управління сприяє швидкому звиканню користувачів до програми;
- мінімалізм дизайну та відсутність надлишкової інформації допомагають уникнути перевантаження користувачів;
- інтерфейс повинен бути простим для користувача без додаткових інструкцій або навчання;
- інтерфейс повинен бути адаптований до різних робочих процесів і потреб.

На рисунку 3.2 представлено зразки часто використовуваних модульних сіток для створення веб-сторінок інформаційної системи [30].

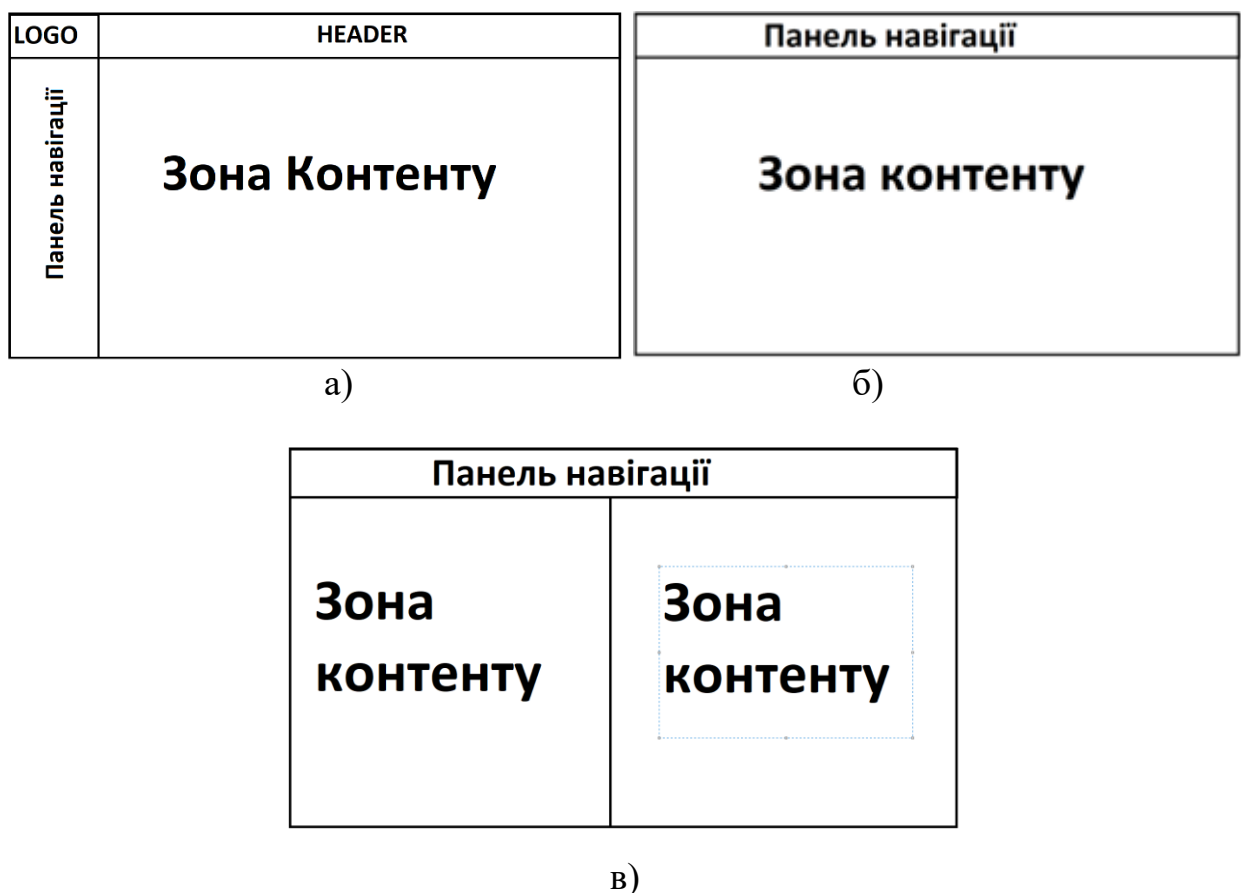


Рисунок 3.2 – Часто використовувані зразки модульних сіток

З макетів видно, що основною відмінністю сторінок є положення навігаційної панелі, залежно від кількості необхідного для відображення

контенту панель навігації змінює своє положення, для сторінок, де необхідно виводити більше інформації – вона повинна знаходитись зверху та займати менше корисного простору.

Розробка інтерфейсу проводилась у Visual Studio Code за допомогою JavaScript та фреймворку React та інших бібліотек що представлені на рисунку 3.3.

```
{
  "name": "my-app",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@date-io/date-fns": "^2.17.0",
    "@emotion/react": "^11.11.1",
    "@emotion/styled": "^11.11.0",
    "@mui/icons-material": "^5.14.9",
    "@mui/lab": "^5.0.0-alpha.146",
    "@mui/material": "^5.14.10",
    "@mui/x-date-pickers": "^6.18.1",
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "axios": "^1.5.1",
    "chart.js": "^4.4.0",
    "cors": "^2.8.5",
    "dotenv": "^16.3.1",
    "formik": "^2.4.2",
    "leaflet": "^1.9.4",
    "ml-matrix": "^6.10.5",
    "numeric": "^1.2.6",
    "openai": "^3.0.0",
    "react": "^18.2.0",
    "react-chartjs-2": "^5.2.0",
    "react-date-range": "^1.4.0",
    "react-datepicker": "^4.18.0",
    "react-dom": "^18.2.0",
    "react-leaflet": "^4.2.1",
    "react-router-dom": "^6.14.2",
    "react-scripts": "5.0.1",
    "recharts": "^2.7.2",
    "sass": "^1.64.1",
    "smallest-enclosing-circle": "^1.0.2",
    "web-vitals": "^2.1.4"
  },
}
```

Рисунок 3.3 – Встановлені бібліотеки для розробки інтерфейсу

Завершальним етапом є впровадження інтерфейсу в загальну структуру інформаційної системи та його інтеграція з бекенд-компонентами, що забезпечують функціонування бізнес-процесів закладу харчування.

### 3.3 Розробка програмного коду додатку

Реалізація програмного коду застосунку виконувалась з використанням мови програмування C#, вона є об'єктно-орієнтованою та опирається на високопродуктивний, модульний, відкритий фреймворк ASP.NET Core, що служить для розробки сучасних інтернет-орієнтованих застосунків та веб

сторінок.

Інтеграція з базою даних та забезпечення взаємодії з API виконується за допомогою Entity Framework Core, що є розширюваним, легким та крос-платформним ORM рішенням. Даний фреймворк надає можливість розробникам, взаємодіяти з базою даних за допомогою високорівневих об'єктних моделей, спрощуючи функції читання, створення, видалення та оновлення та забезпечувати абстракцію від складнощів SQL синтаксису.

У поєднанні з Entity Framework Core використання ASP.NET Core полегшує створення масштабованих веб-додатків і сприяє оптимізації процесу розробки завдяки вбудованим механізмам автентифікації, авторизації та обробки запитів, а також покращує взаємодію з базою даних та API. Завдяки використанню цих сучасних технологій розширюваність, безпеку та високий рівень управління якістю програмного коду гарантуються.

Перед початком процесу розробки програмного коду інформаційної системи, важливою частиною, на яку необхідно звернути увагу, є проектування та розробка бази даних. Даний етап є основоположним та обов'язковим, так як оптимізована та правильно структурована база даних є основою для ефективного управління даними та результативної взаємодії сервера з інтерфейсом користувача. Правильне проектування бази даних полягає у визначенні сутностей та атрибутів, а також взаємозв'язків, котрі забезпечують надійність, цілісність та швидкодію роботи розроблювальної інформаційної системи. Створення реляційної бази даних є важливим етапом, який потребує детального аналізу необхідних бізнес процесів, та функціональних вимог користувачів для того, щоб гарантувати високий рівень зберігання даних, оптимізації та швидкодії запитів [31].

Розробка БД проводилась за допомогою програми SSMS, були визначені необхідні таблиці, стовбці та зв'язки з іншими таблицями для коректної роботи, ER-діаграма представлена на рис. 3.4.

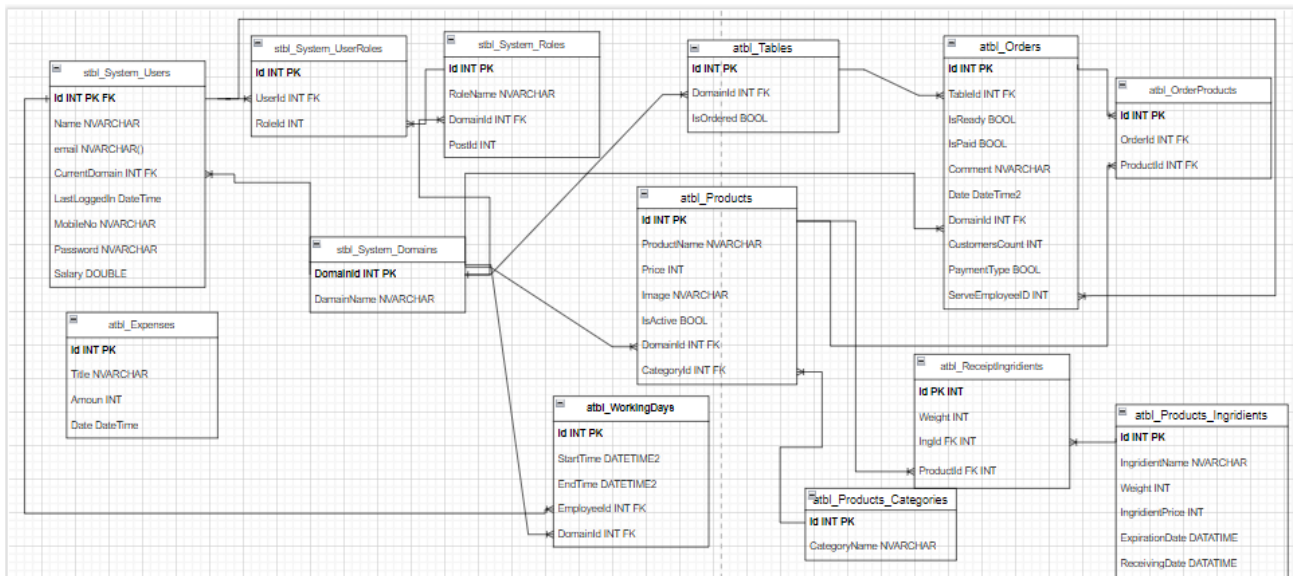


Рисунок 3.4 – ER-діаграма бази даних

В ході розробки бази даних були створені наступні таблиці:

- System\_Users – таблиця призначена для зберігання інформації про користувачів;
- System\_UserRoles – проміжна таблиця, призначена для зберігання id користувачів та id ролей, які у них є;
- System\_Domains – таблиця для зберігання назв домейнів для можливості використовувати базу даних для декількох сфер;
- System\_Roles – таблиця, що зберігає інформацію про існуючі ролі у системі;
- Tables – таблиця, що зберігає інформацію про кількість столів та їх статус;
- Orders – таблиця, що зберігає інформацію про існуючі замовлення у системі;
- OrderProducts – проміжна таблиця, призначена для зберігання інформації про замовлені страви конкретного замовлення;
- Products – таблиця, що зберігає інформацію про доступні продукти у системі;
- ReceiptIngredients – проміжна таблиця, призначена для створення рецептів для конкретного продукту;

- Products\_Ingredients – таблиця, що зберігає інформацію про наявні інгредієнти у системі;
- Products\_Categories – таблиця, що зберігає інформацію про існуючі категорії у системі;
- WorkingDays – таблиця, призначена для запису про початок та кінець робочого дня для конкретних працівників;
- Expenses – таблиця, призначена для запису витрат.

Всі ці таблиці повинні організувати коректну роботу інформаційної системи для оптимізації бізнес процесів закладу харчування. Наступним кроком необхідно підключити створену БД у SSMS до програмного коду backend частини у Microsoft Visual Studio.

Для підключення бази даних до програмного коду C# необхідно виконати наступний код у SSMS для отримання строки підключення у SQL, код представлено на рисунку 3.5.

```

select
    'data source=' + @@servername +
    ';initial catalog=' + db_name() +
    case type_desc
        when 'WINDOWS_LOGIN'
            then ';trusted_connection=true'
        else
            ';user id=' + suser_name() + ';password=<<YourPassword>>'
    end
    as ConnectionString
from sys.server_principals
where name = suser_name()

```

Рисунок 3.5 – Код отримання строки підключення

В результаті виконання запиту отримаємо строку підключення бази даних, яку припишемо в налаштуванні файлу appsettings.json проєкту з програмним кодом, підключення зображено на рис. 3.6.

```

{
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "data source=DESKTOP-5L1AK1L\\SERVER;initial catalog=RestourantDB;trusted_connection=true;TrustServerCertificate=True"
  }
}

```

Рисунок 3.6 – Підключення БД у програмному коді

Для можливості працювати з таблицями у підключений базі даних, необхідно оголосити кожен з них окремо, як представлено на рис. 3.7.

```
// Data/RestaurantContext.cs
using Microsoft.EntityFrameworkCore;
using MyApi.Models;

namespace MyApi.Data
{
    public class RestaurantContext : DbContext
    {
        public RestaurantContext(DbContextOptions<RestaurantContext> options) : base(options)
        {
        }
        public DbSet<Product> Products { get; set; }
        public DbSet<Ingredient> Ingredients { get; set; }
        public DbSet<ProductReceipt> ProductReceipts { get; set; }
        public DbSet<Category> ProductsCategories { get; set; }
        public DbSet<Table> Tables { get; set; }
        public DbSet<Order> Orders { get; set; }
        public DbSet<OrderProduct> OrderProducts { get; set; }
        public DbSet<Expense> Expences { get; set; }
        public DbSet<User> Users { get; set; }
        public DbSet<Role> Roles { get; set; }
        public DbSet<UserRoles> UserRoles { get; set; }
        public DbSet<Post> Posts { get; set; }
        public DbSet<Domain> Domains { get; set; }
        public DbSet<WorkingDay> WorkingDays { get; set; }
    }
}
```

Рисунок 3.7 – Оголошення таблиць БД

Після цих дій, базу даних можна вважати підключеною до програмного коду, що дозволить створювати необхідні API для комунікації з frontend частиною.

Отже, головною частиною написання програмного коду є розробка API для взаємодії з інтерфейсом та передачі даних з бази даних до нього. Для цього необхідно створити моделі, що представляють собою вигляд таблиць з бази даних, перелік створених моделей представлено на рис. 3.8.

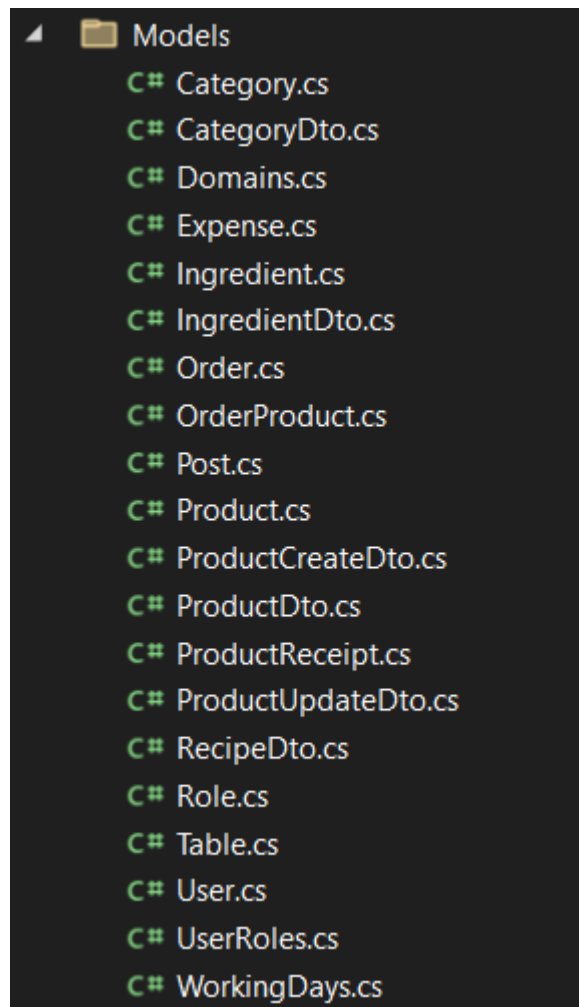


Рисунок 3.8 – Перелік створених моделей

Для більш зручної роботи, необхідно створити DTO класи для відображення окремих полів з БД, це необхідно для відокремлення внутрішньої структури даних від тих, що передаються між клієнтом та сервером, це дозволить використовувати окремі поля, не змінюючи інші. DTO – об'єкт, що використовується для передачі даних з одного місця програми в інше. Найчастіше для передачі даних з бекенд рівня на рівень UI [31].

Загальний вигляд створеної моделі даних зображено на рис. 3.9.

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MyApi.Models
{
    [Table("atbl_Products_Ingridients")]
    public class Ingredient
    {
        [Key]
        public int Id { get; set; }
        public string IngridientName { get; set; }
        public int Weight { get; set; }
        public string Price { get; set; }
        public DateTime ExpirationDate { get; set; }
        public DateTime ReceivingDate { get; set; }
        public virtual ICollection<ProductReceipt> Receipts { get; set; }
    }
}

```

Рисунок 3.9 – Загальний вигляд створеної моделі

Розробка API полягає у створенні контролерів класу, який обробляє HTTP запити та відповіді, для цього необхідно використовувати ASP.NET CORE.

ASP .NET Core є крос-платформовим, високопродуктивним фреймворком з відкритим кодом, випущений компанією Microsoft в 2016 році. Він призначений для створення сучасних хмарних додатків, таких як веб-додатки та служби, додатки Інтернету речей (IoT) та API [31].

До переваг цього фреймворку можна віднести:

- можливість працювати на будь-якій операційній системі;
- opensource-фреймворк;
- підтримка багатьох сервісів;
- велика спільнота;
- оптимізація високого рівня.

У контексті розробки API в ASP.NET Core, контролер – це клас, який наслідується від класу ControllerBase і обробляє HTTP запити та відповіді. Контролери активуються та знищуються для кожного запиту [32].

Чотири типи запитів до БД використовуються під час розробки API веб-сайту:

- GET – використовує атрибут [HttpGet] для обробки запиту GET;
- POST – використовує атрибут [HttpPost];
- PUT – використовує атрибут [HttpPut] для обробки запиту PUT;
- DELETE – використовує атрибут [HttpDelete] для обробки запиту DELETE.

DELETE.

Загальний вигляд запитів продемонстровано на рис. 3.10.

```
[HttpGet("{id}")]
public ActionResult<Product> GetProduct(int id)
{
    var product = _context.Products.Find(id);
    if (product == null)
    {
        return NotFound();
    }
    return Ok(product);
}

[HttpPost]
public ActionResult<Product> CreateProduct(Product product)
{
    _context.Products.Add(product);
    _context.SaveChanges();

    return CreatedAtAction(nameof(GetProduct), new { id = product.Id }, product);
}

[HttpPut("{id}")]
public IActionResult UpdateProduct(int id, Product productToUpdate)
{
    if (id != productToUpdate.Id)
    {
        return BadRequest();
    }

    _context.Entry(productToUpdate).State = EntityState.Modified;
    try
    {
        _context.SaveChanges();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!_context.Products.Any(e => e.Id == id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }

    return NoContent();
}

[HttpDelete("{id}")]
public IActionResult DeleteProduct(int id)
{
    var product = _context.Products.Find(id);
    if (product == null)
    {
        return NotFound();
    }

    _context.Products.Remove(product);
    _context.SaveChanges();

    return NoContent();
}
```

Рисунок 3.10 – Загальний вигляд запитів GET, POST, PUT, DELETE

За допомогою цих запитів виконується основний функціонал для відображення інформації на веб-сайті, оновлення даних у БД, а також видалення та додавання інформації. Для перевірки коректної роботи розробленого API було прийнято рішення використовувати Swagger.

Swagger – це набір інструментів для проектування, будівництва, документування та використання RESTful веб-сервісів. Він включає автоматичне документування, генерацію клієнтського коду та перевірку API. Swagger дозволяє розробникам описувати структуру своїх API так, що машини можуть її читати, що є ключем до автоматизації і використання веб-сервісів [33].

Swagger часто використовується разом з інструментом Swagger UI, який генерує інтерактивну документацію для веб-сервісів. Ця документація дозволяє розробникам та іншим зацікавленим сторонам переглядати та тестувати API [34]. На рис. 3.11 представлено код підключення Swagger до проєкту.

```
builder.Services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new() { Title = "My API", Version = "v1" });
});
app.UseSwagger();
app.UseSwaggerUI(c =>
{
    c.SwaggerEndpoint("/swagger/v1/swagger.json", "My API V1");
});
```

Рисунок 3.11 – Підключення Swagger

За допомогою підключеного Swagger до проєкту зменшилась кількість помилок у розробці коду, а також час на перевірку правильності написання програмного коду, що полегшує розробку backend частини інформаційної системи та збільшує ефективність та якість роботи. Вигляд Swagger та результат його роботи представлено на рис. 3.12 – 3.13.

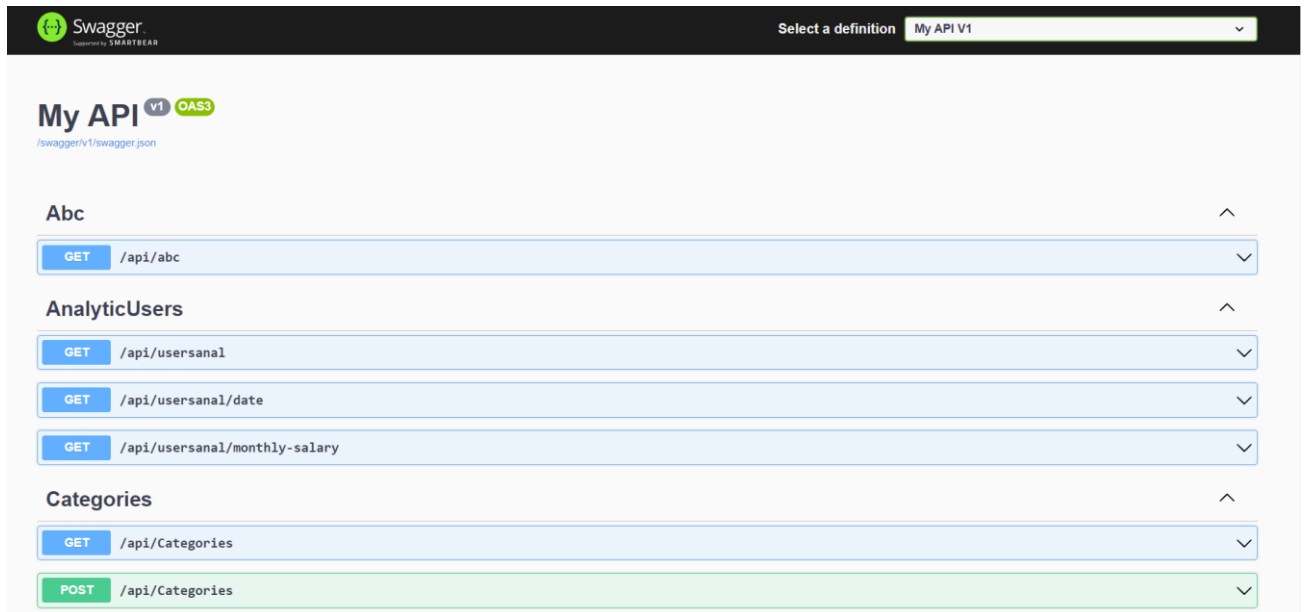


Рисунок 3.12 – Вигляд вікна Swagger UI

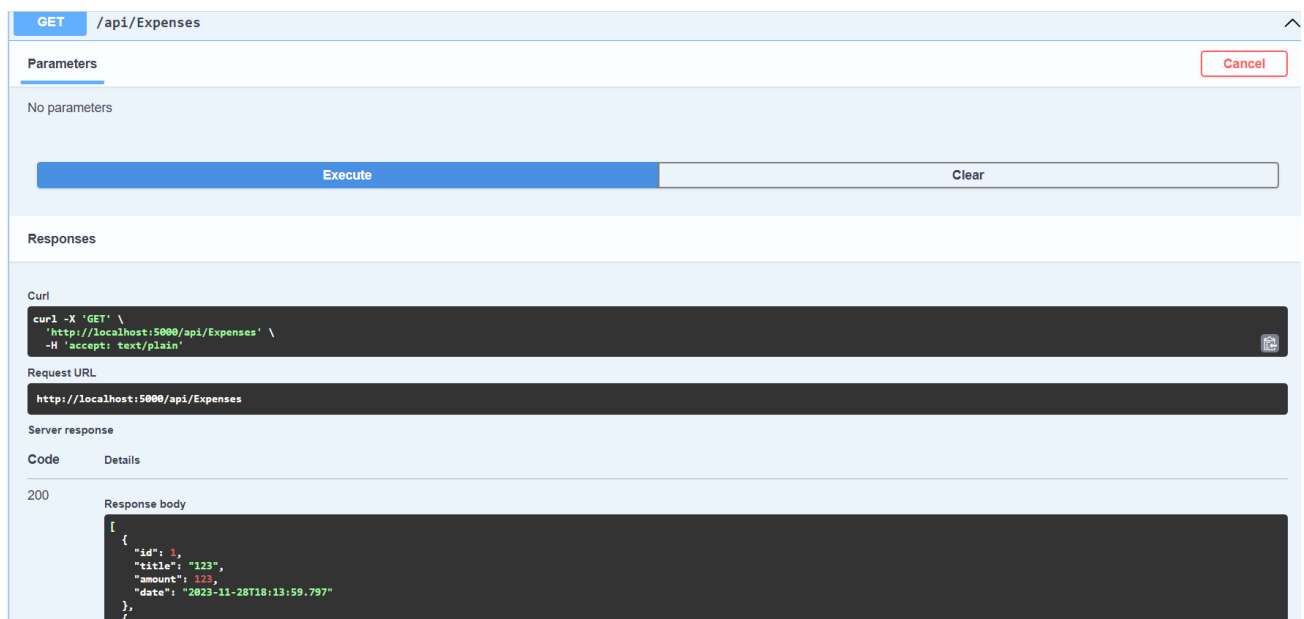


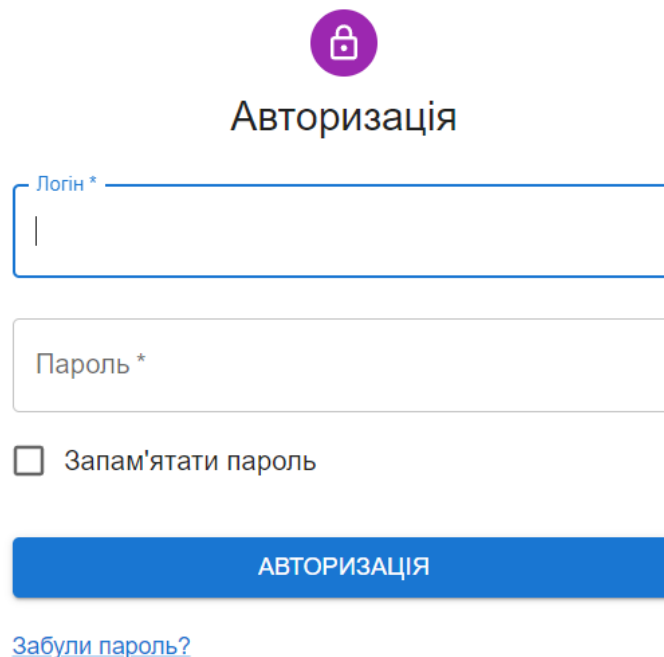
Рисунок 3.13 – Результат роботи запиту GET

Як видно з рисунку 3.12, у разі успішного виконання запиту GET до БД, отримуємо посилання на API та результат виконання запиту. Отримане посилання дозволяє використовувати цей запит у frontend частині проєкту для роботи з базою даних, а завдяки отриманій відповіді є можливість перевірити правильність написання коду контролера, та побачити інформацію, що він виводить.

### 3.4 Розроблений програмний додаток

Після виконання поставлених задач та мети проєкту на виході було розроблено інформаційну систему оптимізації бізнес процесів закладу харчування. Веб застосунок складається з трьох основних компонентів: сторінка аналітики та звітності, сторінка офіціантів та сторінка кухарів [35].

Першою сторінкою, що зустрічає користувача є сторінка авторизації, після успішної авторизації, в залежності від ролі користувача проходить переадресування на необхідну сторінку. Сторінка авторизації зображена на рис. 3.14.



Логін \*

Пароль \*

Запам'ятати пароль

АВТОРИЗАЦІЯ

[Забули пароль?](#)

Рисунок 3.14 – Сторінка авторизації

Після успішної авторизації, як адміністратор проходить авторизація до сторінок з управлінням, аналітикою та звітністю. Головна сторінка блоку адміністратору продемонстрована на рисунку 3.15, що показує інформацію про витрати, прибуток, кількість замовлень, а також середній чек на теперішній час, а також має форму для перегляду та запису витрат.

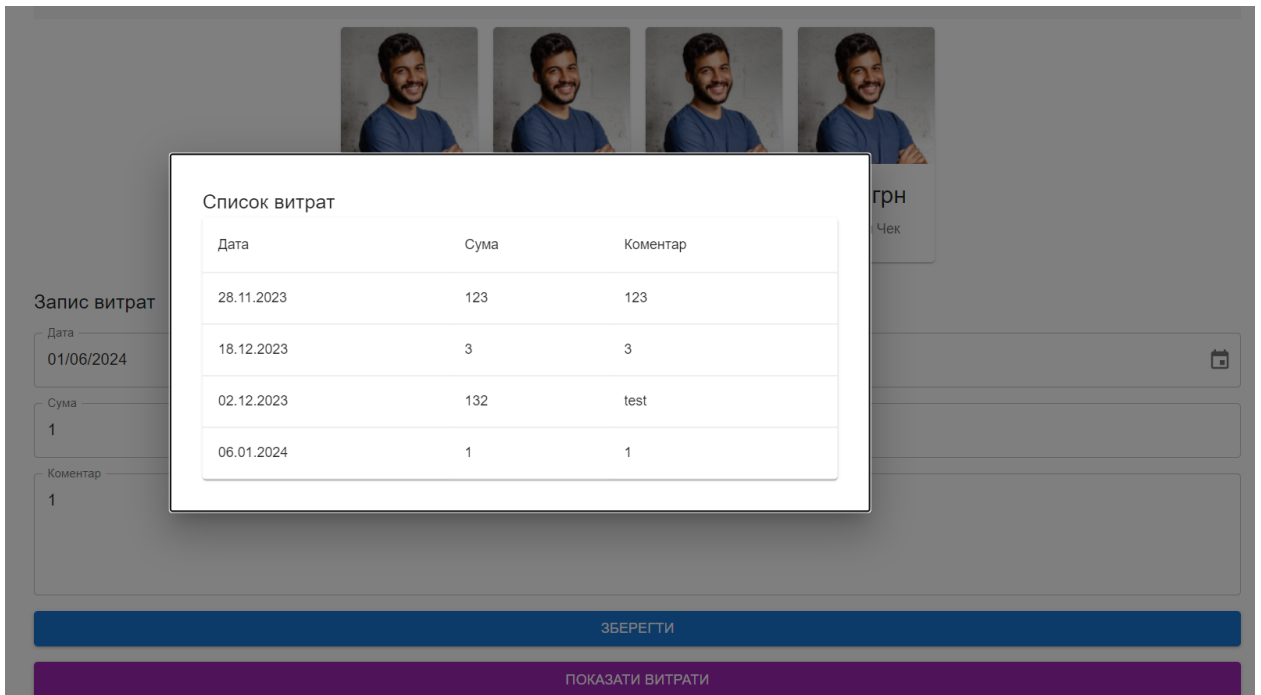


Рисунок 3.15 – Головна сторінка блоку адміністратору

Сторінка з меню дозволяє переглядати та редагувати меню, або додавати нові страви та рецепти та категорії страв. Вигляд сторінки меню показано на рис. 3.16.

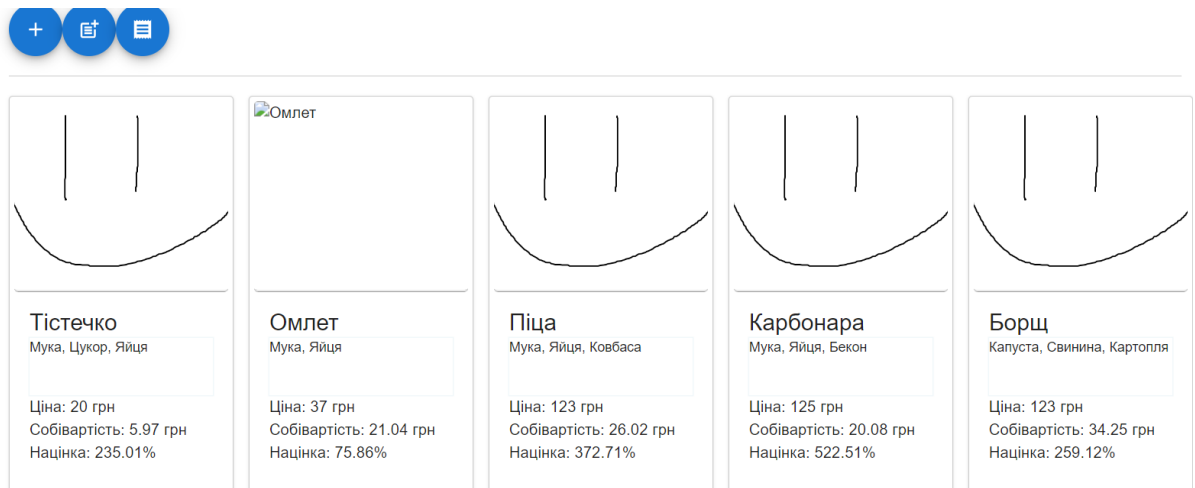


Рисунок 3.16 – Вигляд сторінки меню


Форми для редагування страв, додавання категорій та нових страв у меню, а також форма редагування існуючої страви продемонстровано на рис. 3.17-3.20.

### Редагування страви

Name  
Тістечко

Price  
20

Категорія  
Desert



Знижка

[ЗБЕРЕГТИ](#) [ВИДАЛИТИ](#) [ЗАКРИТИ](#)

Рисунок 3.17 – Форма Редагування існуючої страви

### Створення рецепту

Дія  
Add Recipe

Оберіть страву

Select Ingredients

Weight (g)  
0

[ДОДАТИ ІНГРІДІЄНТ](#)

[СТВОРИТИ РЕЦЕПТ](#)

Рисунок 3.18 – Форма створення нового рецепту

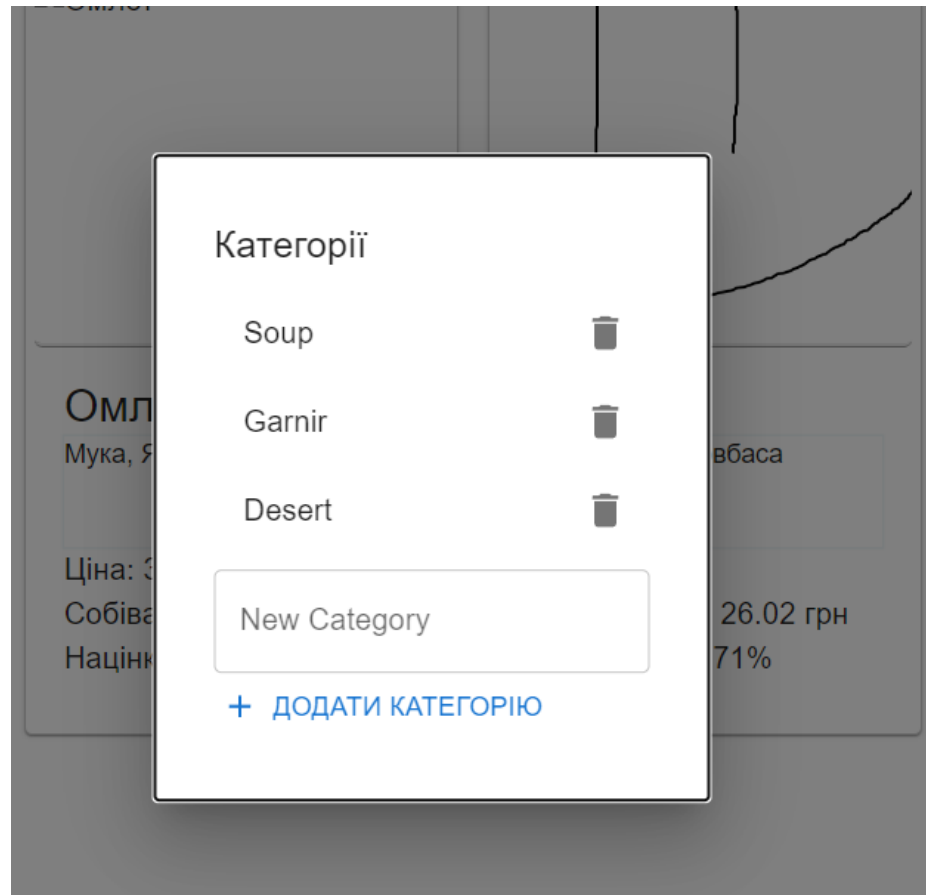


Рисунок 3.19 – Форма додавання нових категорій

## Додати нову страву

Назва страви

Ціна

Вибрати файл Файл не вибрано

Категорія ▼

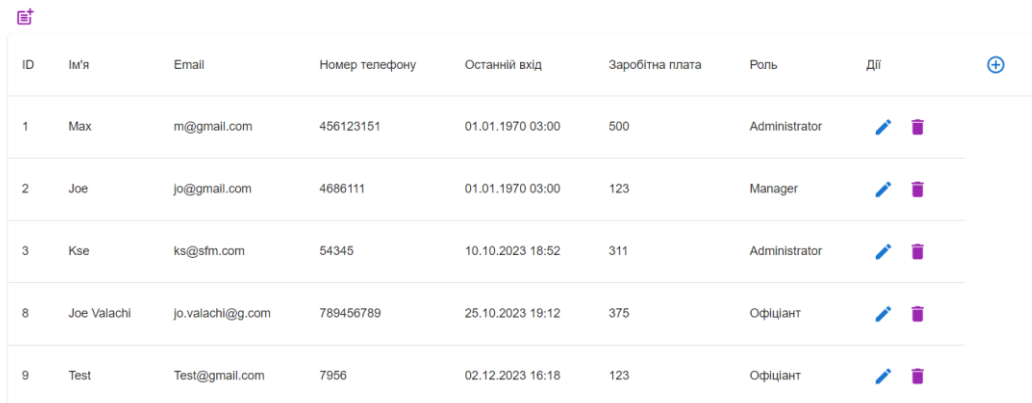
**ДОДАТИ СТРАВУ**

Рисунок 3.20 – Форма додавання нової страви у меню

За допомогою цих форм проводиться повне керування меню, що дозволяє повністю редагувати страви, додавати нові з новими рецептами, а також фільтрувати страви за категоріями.

Наступна сторінка відповідає за користувачів, що зареєстровано в системі, рисунок 3.21 відображає її вигляд.

### Список користувачів



ID	Ім'я	Email	Номер телефону	Останній вхід	Заробітна плата	Роль	Дії
1	Max	m@gmail.com	456123151	01.01.1970 03:00	500	Administrator	
2	Joe	jo@gmail.com	4686111	01.01.1970 03:00	123	Manager	
3	Kse	ks@sfm.com	54345	10.10.2023 18:52	311	Administrator	
8	Joe Valachi	jo.valachi@g.com	789456789	25.10.2023 19:12	375	Офіціант	
9	Test	Test@gmail.com	7956	02.12.2023 16:18	123	Офіціант	

Рисунок 3.21 – Вигляд сторінки з користувачами

На цій сторінці адміністратор має можливість переглядати інформацію про користувачей, додавати нових, а також редагувати інформацію про існуючих. Також, на цій сторінці є форма для упраління ролями, що дозволяє створювати нові ролі та видаляти старі, та присвоювати ролям звання адміністратора, вигляд форми управління ролями показано на рис. 3.22.

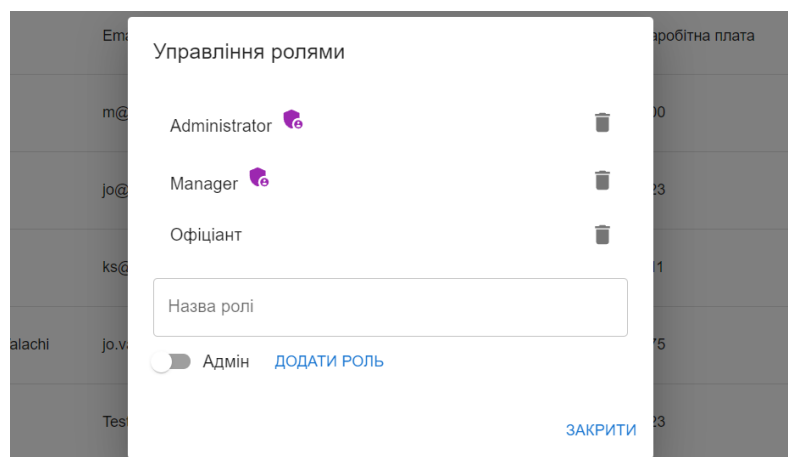


Рисунок 3.22 – Форма управління ролями

Форма редагування користувача надає адміністратору змінювати інформацію про них, та присвоювати їм ролі, а також вносити корективи, за необхідністю, що стосуються заробітної плати. Рисунок 3.23 демонструє форму редагування користувача.

Редагування користувача

Ім'я  
Max

Email  
m@gmail.com

Номер телефону  
456123151

Зарплата  
500

Роль  
Administrator

ЗБЕРЕГТИ ЗАКРИТИ

Рисунок 3.23 – Форма редагування користувача

Також в розробленій інформаційній системі є сторінка для моніторингу та управління товарами на складі, що виводить всі існуючі товари та інформацію про них, а також дає можливість редагувати та додавати нові. Вигляд сторінки продемонстровано на рис. 3.24.

Склад

+ ДОДАТИ ІНГРЕДІЄНТ

Назва	Вага	Ціна	Срок придатності	Дата	Дії
Мука	1234	\$123	31-1-2024	16-11-2022	
Цукор	500	\$3	6-2-2024	15-1-2023	
Яйця	111	\$123	19-11-2023	19-11-2023	
Ковбаса	1000	\$223	2-1-2024	25-12-2023	
Капуста	1000	\$24	2-1-2024	25-12-2023	
Картопля	1000	\$34	2-1-2024	25-12-2023	
Свинина	1000	\$167	2-1-2024	25-12-2023	

Рисунок 3.24 – Вигляд сторінки складу

Вигляд форм додавання та редагування представлено на рис. 3.25.

а)

### Додати Інгредієнт

Срок придатності

[ЗБЕРЕГТИ](#)   [ЗАКРИТИ](#)

---

б)

### Редагування Інгредієнта

Срок придатності   Дата

[ЗБЕРЕГТИ](#)   [ЗАКРИТИ](#)

Рисунок 3.25 – Вигляд форм додавання та редагування інгредієнтів

Наступна сторінка виводить інформацію про всі замовлення, що були зроблені у закладі харчування, рисунок 3.26.

Список Замовлень					
ID	Офіціант	Дата	Сума	Тип Платежу	Дії
3	Test	2023-11-23T00:00:00	\$1021.00	Готівка	<a href="#">ДЕТАЛЬНІШЕ</a>
4	Kse	2023-11-21T18:33:21.986	\$57.00	Картка	<a href="#">ДЕТАЛЬНІШЕ</a>
47	Joe Valachi		3.00	Картка	<a href="#">ДЕТАЛЬНІШЕ</a>
63	Test		3.00	Картка	<a href="#">ДЕТАЛЬНІШЕ</a>
86	Мах		8.00	Готівка	<a href="#">ДЕТАЛЬНІШЕ</a>
87	Мах	2023-11-22T19:24:41.5918404	\$123.00	Готівка	<a href="#">ДЕТАЛЬНІШЕ</a>

**Деталі замовлення #3**  
**Номер столу:** 2  
**Страви:** Тістечко-4, Омлет-2, Піца-3,  
 Карбонара-3, Борщ

[ЗАКРИТИ](#)

Рисунок 3.26 – Вигляд сторінки зі списком замовлень

Ця сторінка виводить інформацію про офіціанта, що приймав замовлення, дату та час, загальну суму, а також тип платежу. Кнопка «Детальніше» відкриває модальне вікно, що надає інформацію про номер столу, та замовлені страви.

Також для адміністратора доступні сторінки з аналітикою, першою сторінкою є аналітика продажів, що представлена на рисунку 3.27

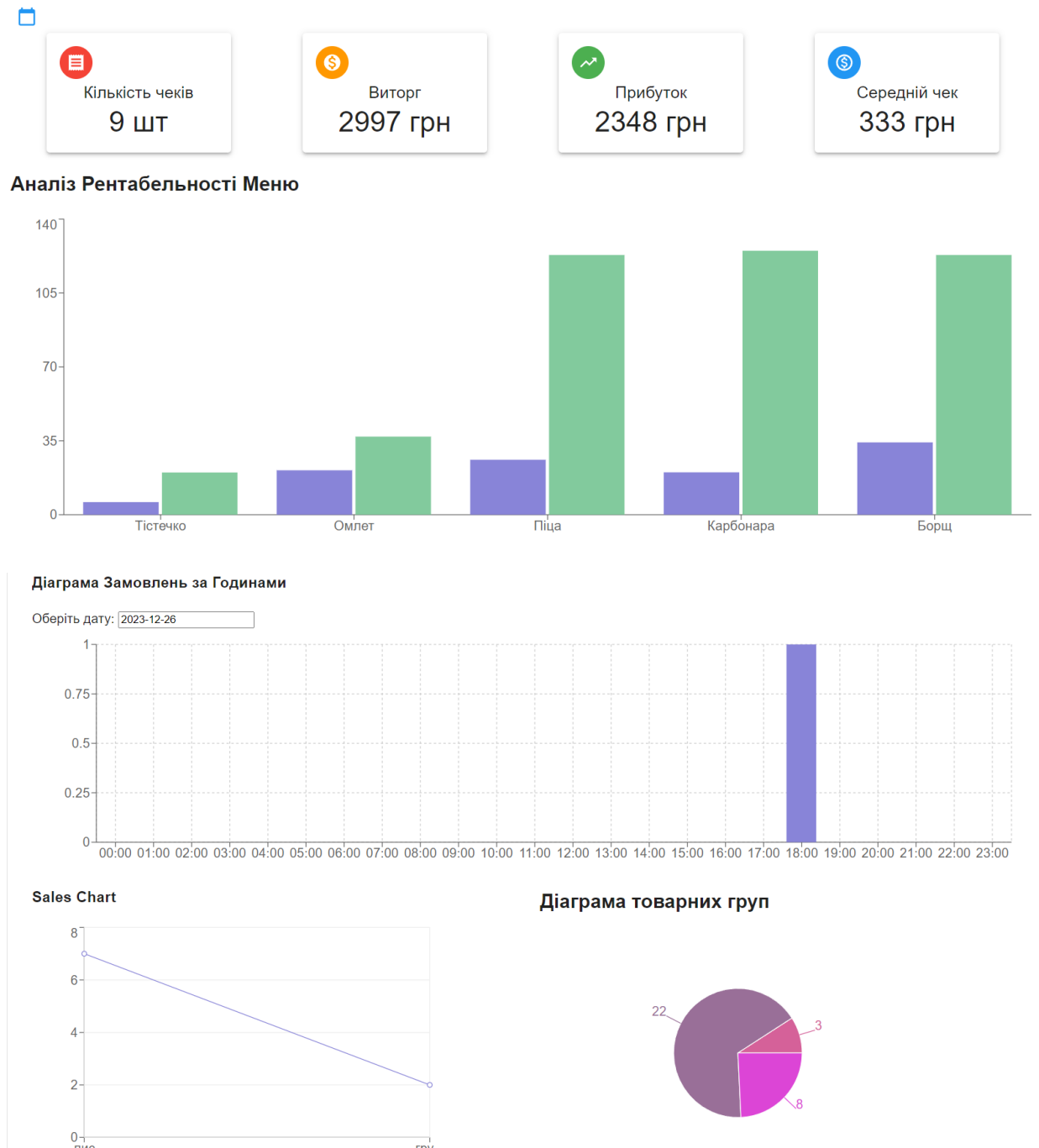


Рисунок 3.27 Вигляд сторінки аналітики продажів

Завдяки цій сторінці адміністратор має можливість переглядати інформацію пов'язану із продажами, та завдяки вбудованому календарю обирати період дати за який потрібно переглянути інформацію.

Також інформаційна система має можливість переглядати статистику працівників, що продемонстровано на рис. 3.28.

Таблиця працівників

ЦЕЙ ДЕНЬ ЦЕЙ МІСЯЦЬ ЦЕЙ РІК

Ім'я	Кількість замовлень	Кількість відпрацьованих годин	ЗП
Max	4	0.20	100.00 грн
Joe	0	2.06	253.97 грн
Kse	1	0.00	0.00 грн
Joe Valachi	2	6.01	2251.95 грн
Test	2	12.12	1491.29 грн

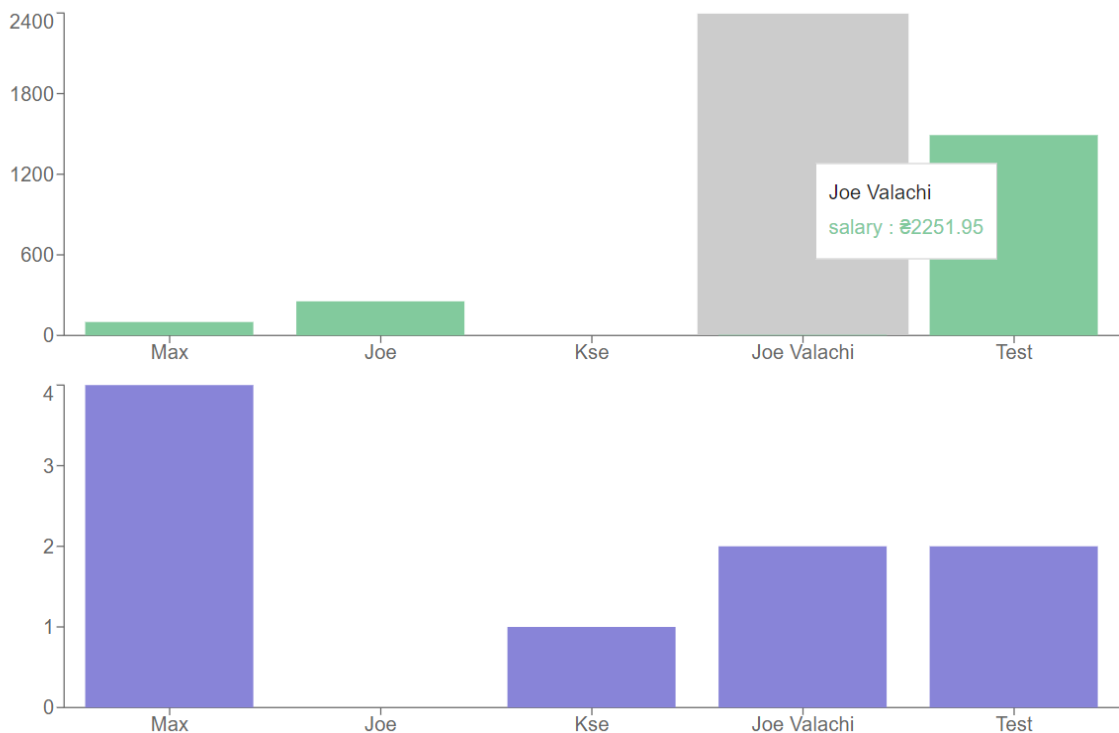
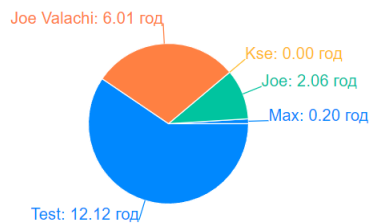


Рисунок 3.28 – Вигляд сторінки аналітики працівників

Ця сторінка також має можливість обирати період часу за який користувач має бажання отримати інформацію про працівників, а також декілька графічних діаграм для зручного користування сторінкою. Вона відображає таблицю про роботу працівників для формування заробітної плати, а також виводить інформацію про відпрацьовані години, заробтні плати та кількість оброблених замовлень за будь-який проміжок часу.

Сторінка аналітики фінансів призначена для отримання повної звітності про фінанси за проміжками часу, рис. 3.29.

### Звіт по фінансах

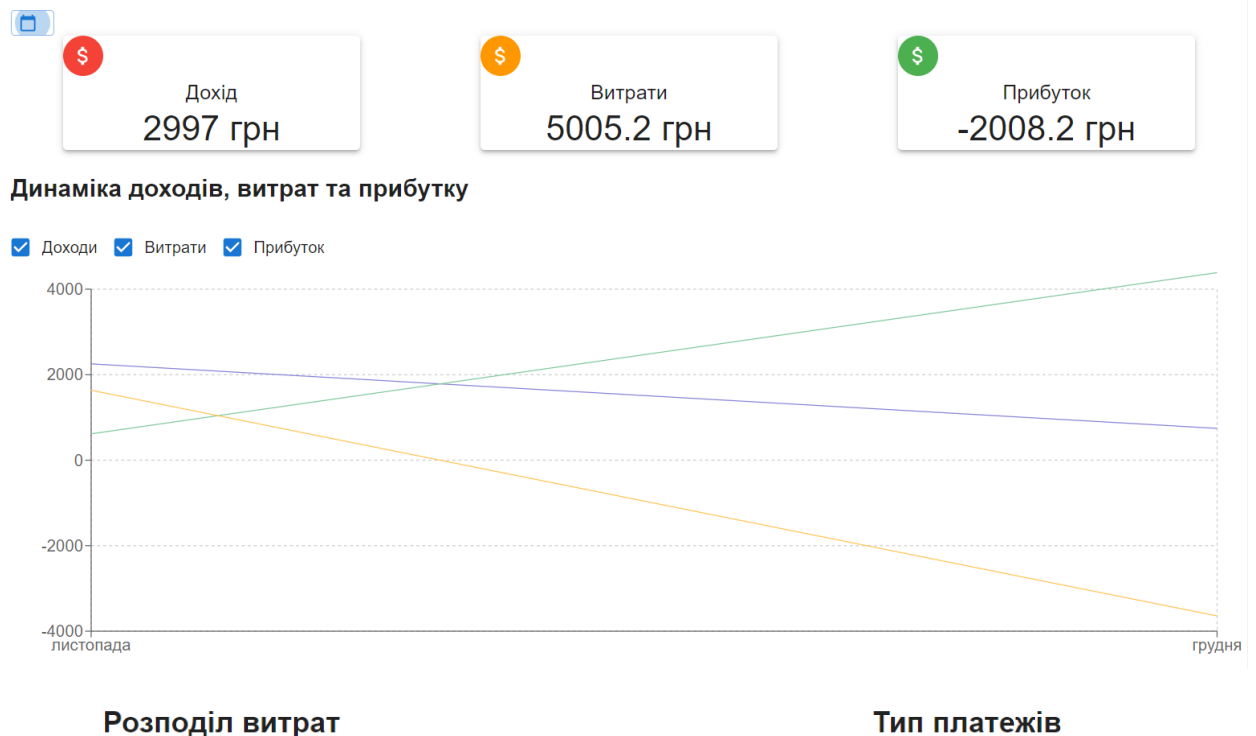


Рисунок 3.29 – Вигляд сторінки аналітики фінансів

На цій сторінці виводиться інформацію про повний дохід, витрати та прибуток, включаючи витрати, записані самостійно користувачем на головній сторінці, а також враховуючи податки. Для кращого розуміння є графік, який демонструє динаміку доходів, витрат та прибутку за місяцями. Дві кругові діаграми відображають інформацію про розподіл витрат та типи платежів у відсотках. На наступній сторінці виконується ABC аналіз (рис. 3.30).

#### ABC Аналіз Товарів

Назва товару	Прибуток	% прибутку	Виторг	% виторгу	Кількість продажів	% продажів	Група ABC (Прибуток)	Група ABC (Виторг)	Група ABC (Кількість продажів)
Піца	872.79	37.17%	1107.00	36.94%	9	27.27%	A	A	A
Карбонара	1049.20	44.68%	1250.00	41.71%	10	30.30%	A	A	A
Борщ	266.25	11.34%	369.00	12.31%	3	9.09%	B	B	B
Тістечко	112.25	4.78%	160.00	5.34%	8	24.24%	C	C	A
Омлет	47.87	2.04%	111.00	3.70%	3	9.09%	C	C	C

Рисунок 3.30 – Вигляд сторінки з ABC аналізом товарів

На цій сторінці автоматично виконується аналіз товарів, після кожного перезавантаження сторінки за трьома групами (прибуток, виторг, кількість продажів), це призначено для кращого розуміння товарів, що потребують корегування.

Наступна сторінка демонструє інформацію про загальні податки, а також графік за місяцями, рис. 3.31.

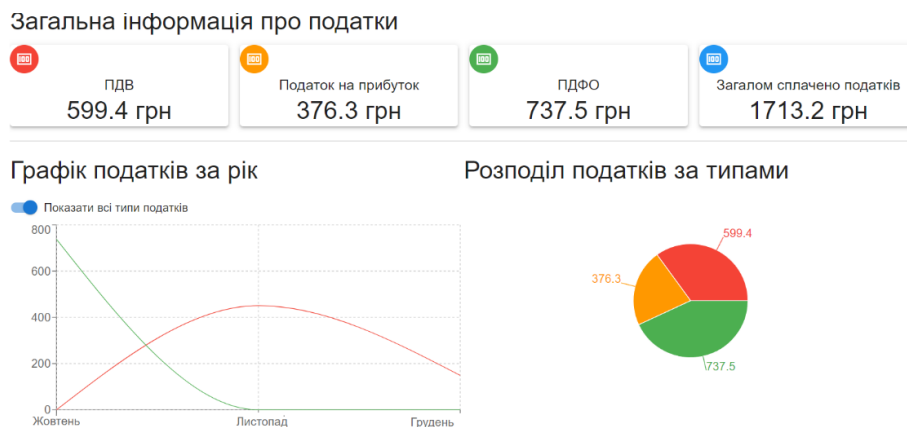


Рисунок 3.31 – Сторінка аналітики податків

Після авторизації, як офіціант, користувач потрапляє на сторінку зі списком столів (рис. 3.32).



Рисунок 3.32 – Сторінка зі списком столів

Червоний колір столу показує те, що стіл вільний, а зелений – на столі вже є активне замовлення. Якщо перейти на стіл на якому є активне замовлення, виведиться інформація про це замовлення, як показано на рис. 3.31.

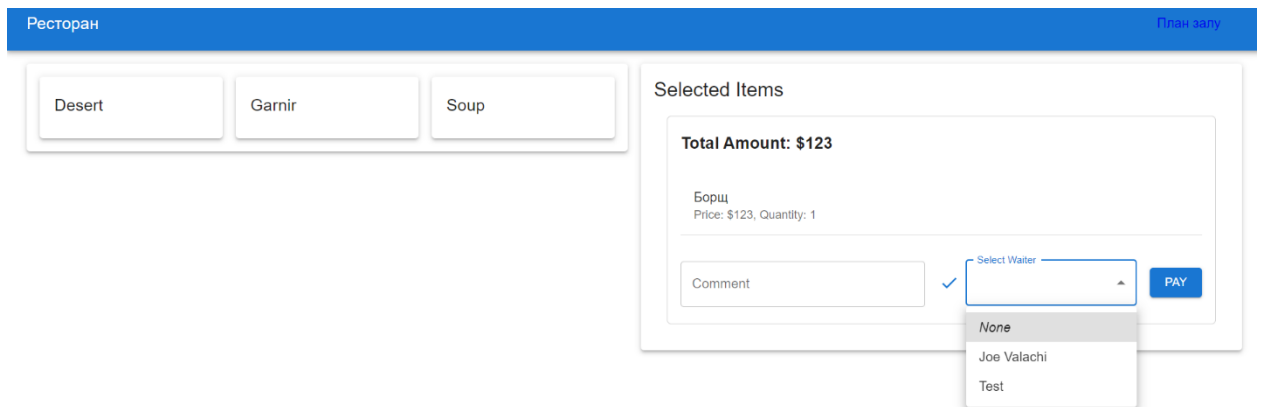


Рисунок 3.33 – Вигляд сторінки оформлення замовлення

Якщо ж перейти на сторінку де немає замовлення, його можна буде розпочати і тоді стіл стане зеленим. Також тут є можливість повернутись до плану залу зі списками столів, або завершити замовлення та перейти до оплати.

У офіціантів є можливість переходити на сторінку з активними замовленнями, як показано на рис. 3.34.

Ресторан							РОБОЧИЙ ДЕНЬ	План залу	Замовлення
Номер замовлення	Номер столу	Ім'я офіціанта	Час створення замовлення	Сума	Готовність	Переглянути замовлення			
2002	6	Joe Valachi	18:15:04	621	Ready	<a href="#">ПЕРЕЙТИ</a>			
2003	3		18:31:30	123	Not Ready	<a href="#">ПЕРЕЙТИ</a>			
3003	4		18:13:24	0	Not Ready	<a href="#">ПЕРЕЙТИ</a>			

Рисунок 3.34 – Сторінка з активними замовленнями

На цій сторінці є можливість переглянути замовлені страви, натиснувши на номер столу, або перейти до сторінки з замовленнями при натисканні на кнопку «ПЕРЕЙТИ». За допомогою всіх цих сторінок офіціанти виконують свої завдання, а вся інформація про всі замовлення відразу ж відправляється до аналітики, котра оновлюється.

Після авторизації, як кухар, користувач потрапляє на сторінку, що приведена на рис. 3.35.

Список замовлень			Деталі замовлення #2003			
№	Статус	Час	Страва	Інгредієнти	Кількість	Стан
2003	Не готово	18:31:30	Борщ	Капуста - 100g, Свинина - 150g, Картопля - 200g	1	<input type="checkbox"/>
3003	Не готово	18:13:24				

**ГОТОВО**
**ГОТУЄТЬСЯ**
**ВИДАЛИТИ**

Рисунок 3.35 – Сторінка кухаря

На цій сторінці кухар має можливість обрати замовлення для початку роботи з ним, відмічати приготовані страви, та після приготування всіх страв у замовленні, при натисканні на кнопку «ГОТОВО» замовлення видаляється з його сторінки.

### 3.5 Тестування розробленої системи

Тема роботи «Розробка інформаційної системи для оптимізації бізнес-процесів закладу харчування», тому предметом тестування є розроблена інформаційна система.

Тестування розробленої системи полягає у:

- оцінка функціональних вимог;
- перевірка коректності реалізації елементів взаємодії з інтерфейсом (кнопки, чекбокси, скролбари, тощо);
- перевірка інтерфейсу розробленої інформаційної системи на коректність відображення на різних розмірах екрану;
- тестування швидкості завантаження сторінок.

Ці процеси тестування забезпечують, що розроблена інформаційна система виконує необхідні дії, зазначені в меті проєкту, для злагодженої роботи всіх компонентів один з одним, згідно з поставленими задачами.

У першому етапі тестування необхідно перевірити, що всі функції виконуються, а саме функції пов'язані з базою даних, тобто дані коректно виводяться/оновлюються/додаються/видаляються. Перевірка відбувалось за допомогою людського ресурсу, тобто натисканням на функціональні кнопки та переревірки виконання результату. Тестування функцій системи пройшло успішно.

Другий етап тестування полягає у перевірці всіх елементів інтерфейсу, за допомогою яких користувач має взаємодіяти з системою. На цьому етапі важливо впевнитися, що всі посилання працюють, кожна кнопка має свою функцію, яка виконується при натисканні, кожна сторінка має скролбар, що дозволяє прогартувати сторінку. В результаті перевірки, було визначено, що всі елементи для взаємодії з інтерфейсом працюють коректно, окрім декількох кнопок, що не несли в собі ніякої мети – їх було видалено, а також на одній сторінці не було додано посилання на попередню сторінку, це також було виправлено, шляхом додавання кнопки, що повертає на попередню сторінку.

Наступний етап тестування полягає у перевірці розробленої системи на різних пристроях, з різними розмірами екрану. Тестування проводилось за допомогою вмонтованого у веб-браузер Google Chrome імітації мобільних пристроїв. Результати експерименту продемонстровано на рис. 3.36 – 3.41.

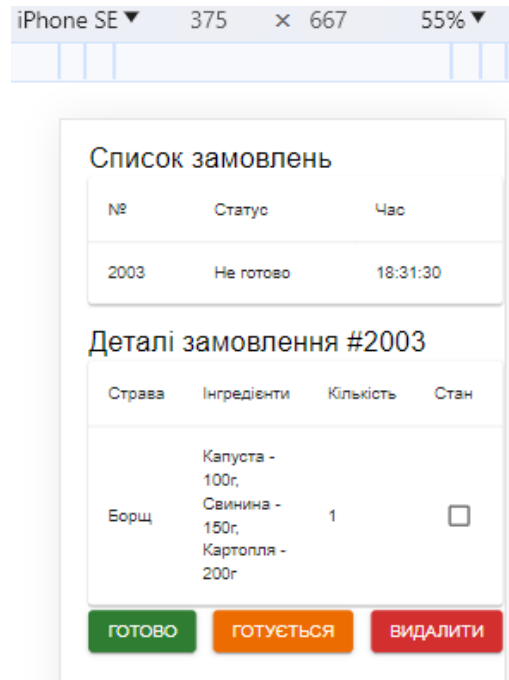


Рисунок 3.36 – Вигляд вікна повара на мобільному приладі

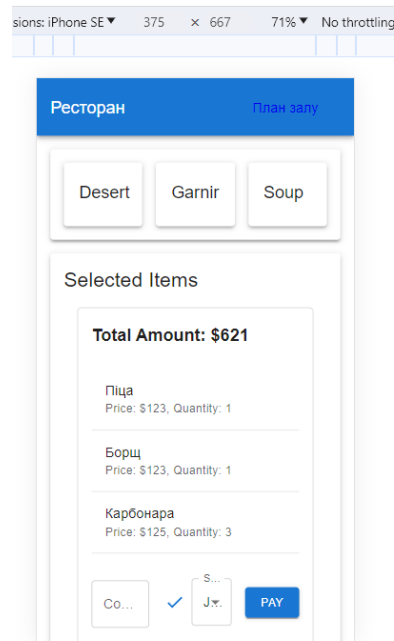


Рисунок 3.37 – Вигляд вікна оформлення замовлення на мобільному приладі

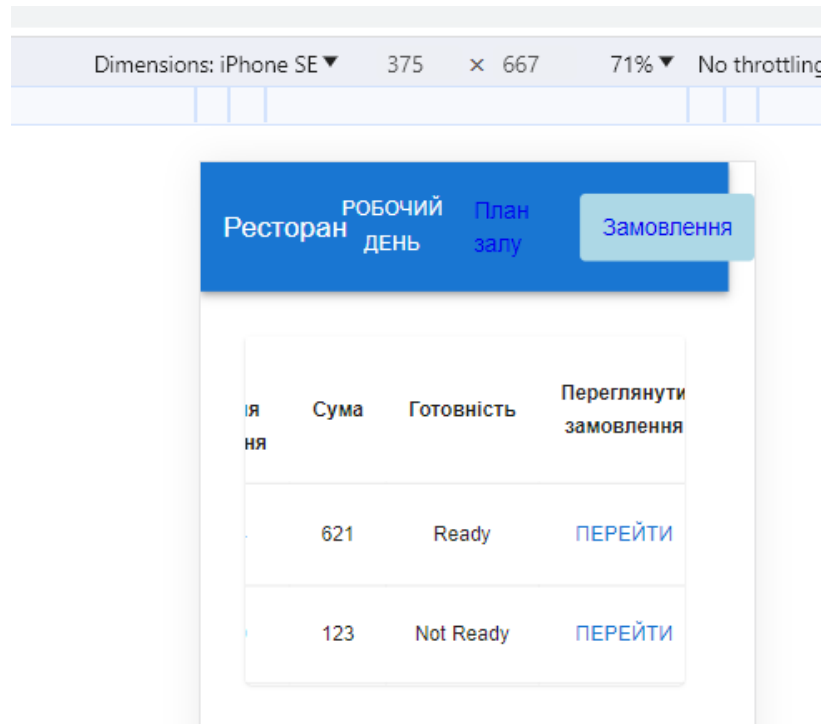


Рисунок 3.38 – Вигляд вікна відкритих замовлень на мобільному приладі



Рисунок 3.39 – Вигляд вікна списку столів на мобільному приладі

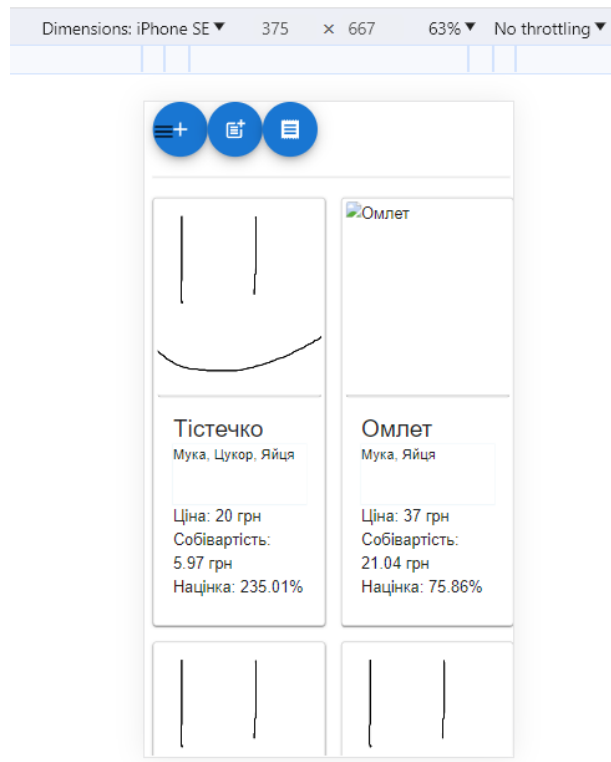


Рисунок 3.40 – Вигляд вікна меню на мобільному приладі

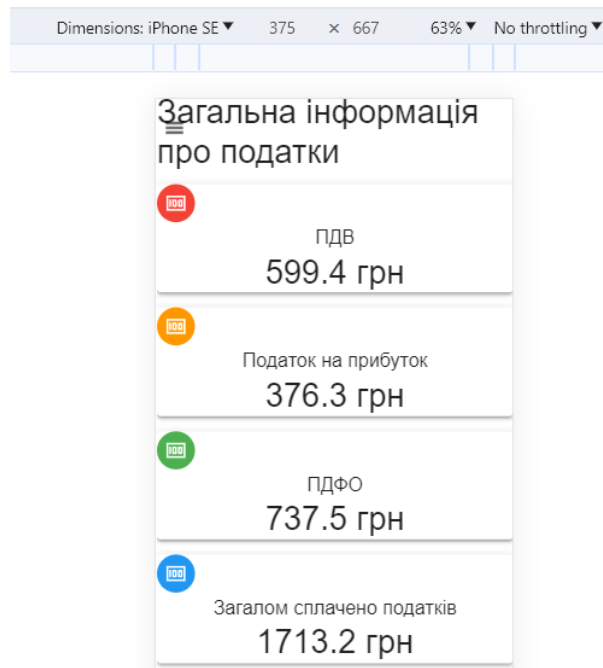


Рисунок 3.41 – Вигляд вікна аналітики на мобільному приладі

Під час проведення цього тестування було виявлено, що не всі сторінки адаптовані до різних розмірів екрану, це було негайно виправлено, бо адаптованість веб застосунку інформаційної системи для оптимізації бізнес

процесів закладу харчування є важливою частиною для повноцінної роботи з розробленою системою.

Останнім етапом є тестування швидкості завантаження сторінки. Важливо зауважити, що нормою швидкості завантаження сторінки вважається 1 – 3 секунди. В цьому етапі тестування проводилось також за допомогою Google Chrome. Результати оцінки завантаження сторінок представлено на рис. 3.42.

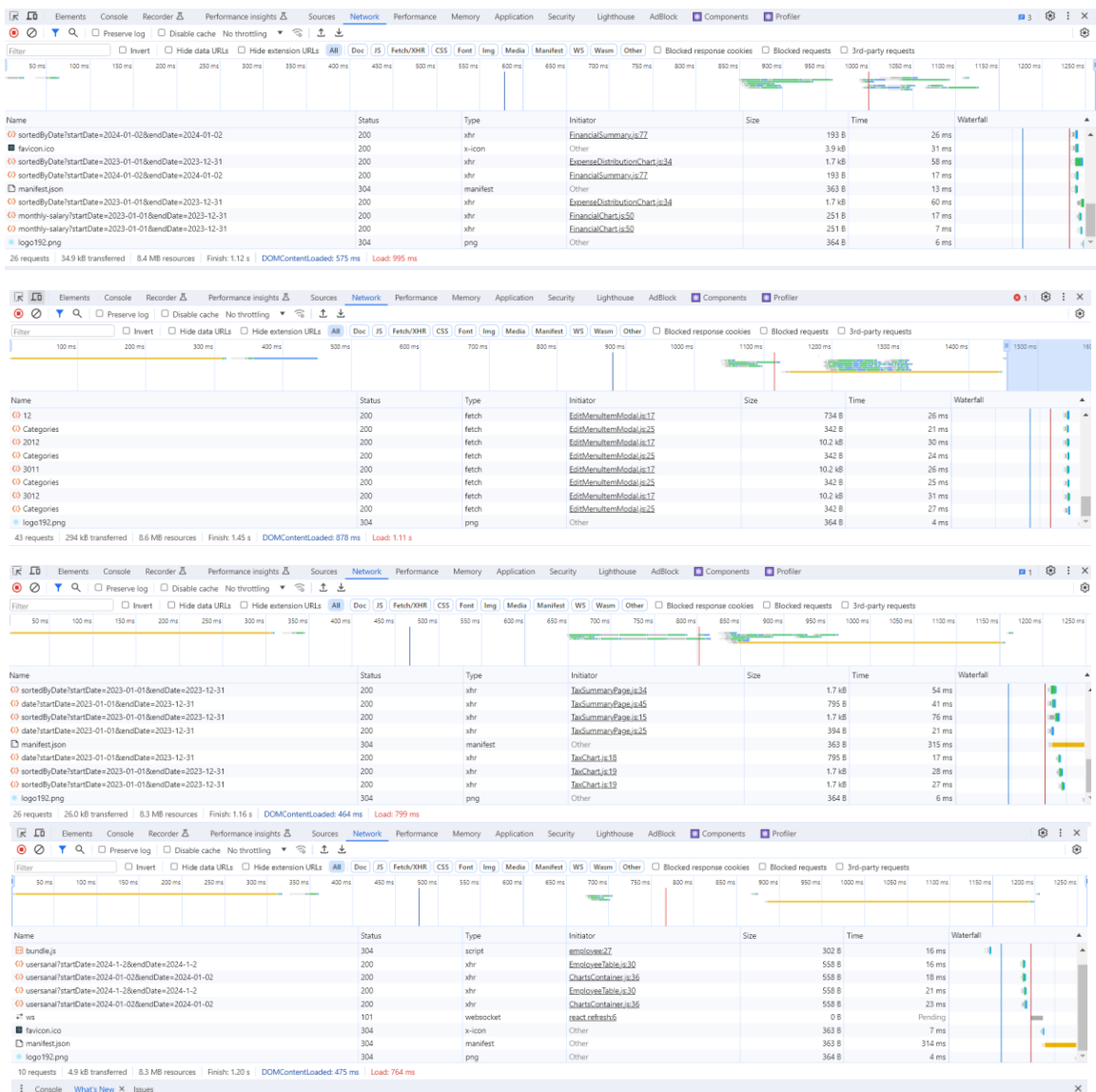


Рисунок 3.42 – Час завантаження сторінок

Тестування розробленої системи дозволило виявити невеликі недоліки та вчасно прийняти міри по їх усуненню.

### 3.6 Експериментальні дослідження

Для визначення показників ефективності роботи закладу, було проведено експеримент порівняння часу обслуговування та кількість людських помилок при роботі традиційними методами та з впровадженою інформаційною системою. Дані при роботі традиційними методами наведені у таблиці 3.1.

Таблиця 3.1 – Середній час обробки замовлення та кількість помилок при роботі традиційними методами

Показник	ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
Середній час обробки замовлення	29	35	32	36	45	48	39
Кількість помилок (на 100 замовлень)	12	13	10	12	15	18	16

На рисунку 3.43 показано гістограму розподілу часу обробки замовлень та кількості помилок при роботі традиційними методами за днями тижня.

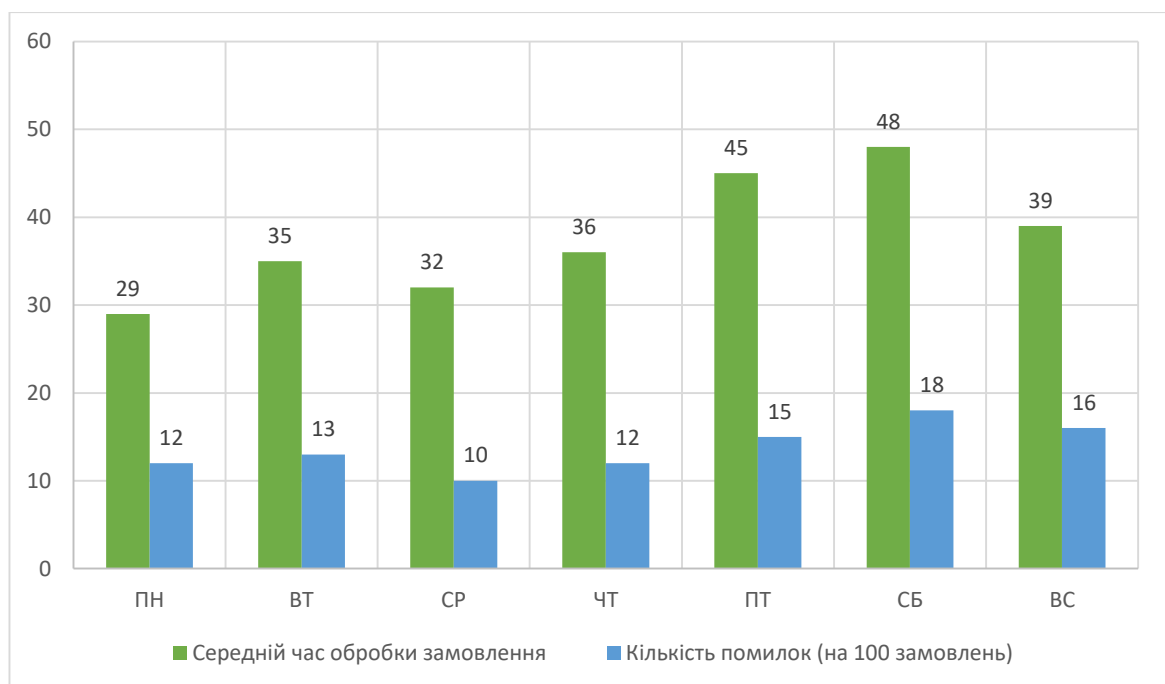


Рисунок 3.43 – Гістограма розподілу часу обробки замовлень та кількості помилок при роботі традиційними методами за днями тижня

Видно, що пікове навантаження на заклад харчування приходить на п'ятницю, суботу та неділю.

Дані роботи закладу харчування з впровадженою інформаційною системою наведені у таблиці 3.2.

Таблиця 3.2 – Середній час обробки замовлення та кількість помилок при роботі з впровадженою інформаційною системою

Показник	ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
Середній час обробки замовлення	14	18	15	19	22	23	25
Кількість помилок (на 100 замовлень)	3	5	4	5	6	6	5

На рисунку 3.44 показано гістограму розподілу часу обробки замовлень та кількості помилок при роботі з впровадженою інформаційною системою за днями тижня.

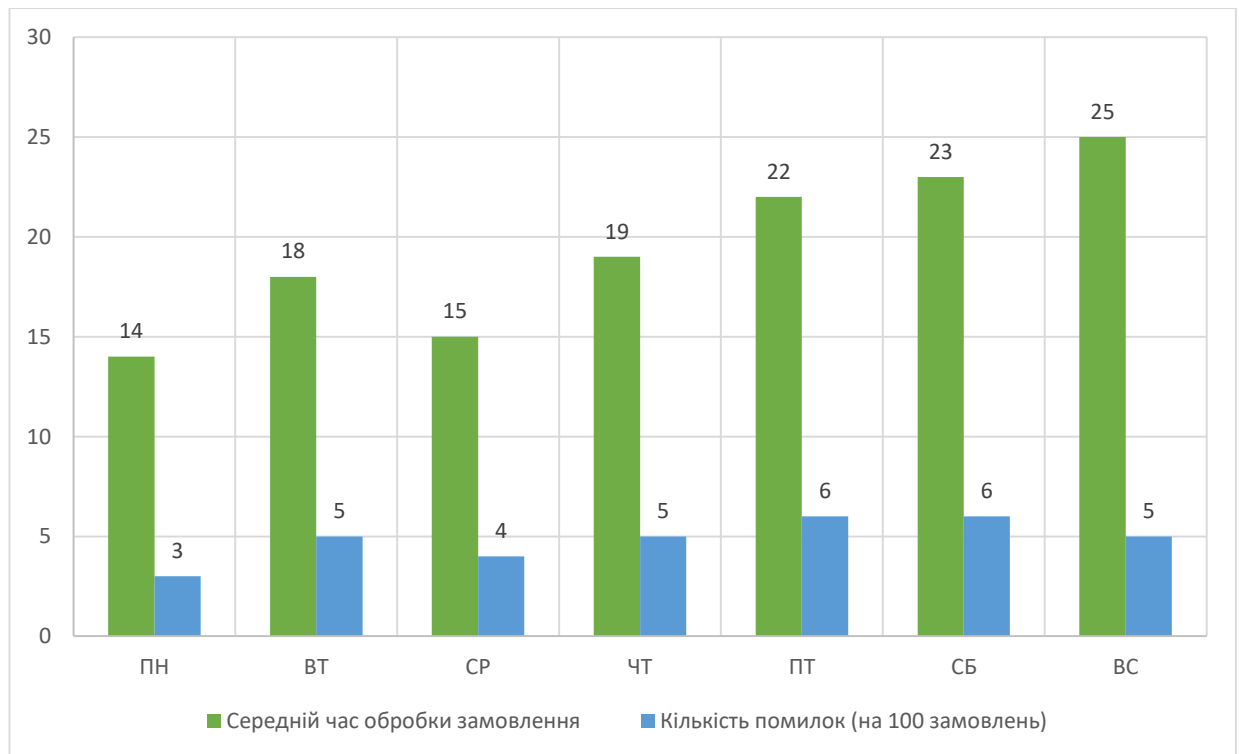


Рисунок 3.44 – Гістограма розподілу часу обробки замовлень та кількості помилок при роботі з впровадженою інформаційною за днями тижня

Як бачимо з наведених діаграм час обробки замовлень, а також кількість помилок значно зменшилась.

Формула розрахунку середнього часу обробки замовлення за тиждень:

$$\overline{t_{зам}} = \frac{\sum_{ВС} t_{зам}}{ПН}, \quad (3.1)$$

де  $t_{зам}$  – середній час обробки замовлення за день тижня;

$\overline{t_{зам}}$  – середній час обробки замовлення за тиждень.

Формула розрахунку середньої кількості помилок за тиждень:

$$\overline{\delta} = \frac{\sum_{ВС} \delta_{зам}}{ПН}, \quad (3.2)$$

де  $\delta$  – кількість помилок у замовленнях за день тижня;

$\overline{\delta}$  – кількість помилок у замовленнях за тиждень.

Отже, згідно (3.1) час обробки замовлень зменшився на 34.2%. Кількість помилок через людський фактор згідно (3.2) також зменшилась на 50.29%.

Виходячи з результатів експерименту можна зробити висновок, що завдяки впровадженню інформаційної системи загальна якість та ефективність роботи закладу збільшилась, через зменшення часу обробки замовлень та кількості помилок через людський фактор.

### 3.7 Охорона праці

Робота велася у приміщені, що відповідає рекомендованим стандартам площі та освітлення, а саме якісними освітлювальними пристроями для

темної пори доби, а також вікна для роботи у світлу пору доби. Вікно розташоване ліворуч від робочого місця, що дозволяє працювати не під прямими променями сонця. Робоче місце складається з столу, якісного офісного крісла, ноутбук, підставка під ноутбук для забезпечення правильної висоти монітору, що становить 400 – 700 міліметрів від очей.

Для того, щоб перебування на робочому місці було комфортним для роботи та відповідало стандартам охорони праці, де кут лівого суглоба повинен бути в межах  $70^{\circ}$  –  $90^{\circ}$ , кут згибу зап'ясть –  $20^{\circ}$ , а нахил голови  $15^{\circ}$  –  $20^{\circ}$ . Це забезпечується завдяки регулюванням робочого місця, а саме крісла за допомогою раймагів для регулювання висоти сидіння, та регулювання кута нахилу спинки, що в свою чергу змінює висоту підлокітників.

Розглянемо вимоги безпеки перед початком роботи з ПК [36]:

- увімкнути систему кондиціонування в приміщенні;
- перевірити надійність встановлення апаратури на робочому столі:
- повернути монітор так, щоб було зручно дивитися на екран - під прямим кутом (а не збоку) і трохи зверху вниз, при цьому екран має бути трохи нахиленим, нижній його край ближче до оператора;
- перевірити загальний стан апаратури, перевірити справність електропроводки, з'єднувальних шнурів, штепсельних вилок, розеток, заземлення захисного екрана;
- відрегулювати освітленість робочого місця;
- відрегулювати та зафіксувати висоту крісла, зручний для користувача нахил його спинки;
- приєднати до системного блоку необхідну апаратуру. Усі кабелі, що з'єднують системний блок з іншими пристроями, слід вставляти та виймати при вимкненому комп'ютері;
- ввімкнути апаратуру комп'ютера вимикачами на корпусах в послідовності:
- монітор, системний блок, принтер (якщо передбачається

друкування);

- відрегулювати яскравість свічення монітора, мінімальний розмір світної точки, фокусування, контрастність. Не слід робити зображення надто яскравим, щоб не втомлювати очей.

Рекомендовано:

- яскравість свічення екрана - не менше 100Kг/М2;
- відношення яскравості монітора до яскравості оточуючих його поверхонь в робочій зоні – не більше 3:1;

- мінімальний розмір точки свічення не більше 0,4 мм для монохромного монітора і не менше 0,6 мм для кольорового, контрастність зображення знаку – не менше 0,8;

Вимоги безпеки під час виконання роботи за ПК:

- необхідно стійко розташовувати клавіатуру на робочому столі, не опускати її хитання. Під час роботи на клавіатурі сидіти прямо, не напружуватися;

- для забезпечення несприятливого впливу на користувача пристроїв типу миша належить забезпечувати вільну велику поверхню столу для переміщення миші і зручного упору ліктьового суглоба;

- не дозволяються сторонні розмови, подразнюючі шуми;

- періодично при вимкненому комп'ютері прибирати ледь змоченою ильним розчином бавовняною ганчіркою порох з поверхонь апаратури. Екран ВДТ та захисний екран протирають ганчіркою, змоченою у спирті. Не дозволяється використовувати рідинні або аерозольні засоби чищення поверхонь комп'ютера.

Заборонено:

- самостійно ремонтувати апаратуру. Ремонт апаратури здійснюється спеціалістами з технічного обслуговування комп'ютера, 1 раз на півроку повинні відкривати процесор і вилучати пирососом пил і бруд, що накопичилися;

- класти будь-яку предмети на апаратуру комп'ютера;

– закривати будь-чим вентиляційні отвори апаратури, що може призвести до її перегрівання і виходу з ладу.

### 3.8 Висновки до розділу

У розділі було розроблено користувацький інтерфейс системи. Для розробки інтерфейсу були сформовані основні вимоги, а також створено основні види макетів для сторінок. Для того, щоб він був, згідно поставленої мети, інтуїтивно зрозумілий, було вирішено розробити його у мінімалістичному дизайні та приємних кольорах. Інтерфейс кожної сторінки було адаптовано для різних розмірів дисплеїв. В результаті інтерфейс було розроблено та додано до проекту.

Програмний код бекенд частини було виконано згідно всім поставленим задачам до інформаційної системи, що були сформовані у ході виконання другого розділу та за допомогою розроблених API, розроблений інтерфейс має доступ до БД.

Також, в цьому розділі провели тестування розробленої інформаційної системи, що полягало у тестуванні функціональних можливостей проекту, а також тестування інтерфейсу на відповідність та було досліджено роботу застосунку на екранах з різними розмірами дисплею. В результаті виконання цього розділу було отримано розроблену інформаційну систему з оптимізації бізнес процесів закладу харчування.

## ВИСНОВКИ

Метою кваліфікаційної роботи було розробити інформаційну системи з оптимізації бізнес процесів закладів харчування, для цього необхідно поставити задачі та виконати їх.

У першому розділі було досліджено аналіз існуючого стану бізнес процесів у закладах харчування, у ході виконання цього розділу стали зрозумілий стан автоматизації бізнес процесів на тепершній час, а також визначено основні переваги оптимізації. Досліджено проблеми, що призводять до неефективної роботи закладів харчування, вирішення яких дозволить оптимізувати бізнес процеси. Також в цьому розділі було визначено сферу застосування та проведено дослідження існуючих аналогів.

В другому розділі було описано та обрано засоби розробки інформаційної системи, а також поставлено задачу для виконання, що дозволило структурувати роботу з проектом та оптимізувати її. Для виконання поставленої мети проекту було сформовано системні та функціональні вимоги, які відображають основні задачі та цілі проекту. Окрім цього визначено мови програмування та середовище розробки БД.

Третій розділ є завершальним і саме в цьому розділі проводилась розробка інтерфейсу програмного засобу та розробка програмоного коду застосунку, а також було створено базу даних для зберігання інформації. В результаті виконання цього розділу було розроблено працюючу інформаційну систему з оптимізації бізнес процесів закладу харчування, та протестовано її.

Таким чином, можна сказати, що всі поставлені цілі та задачі було виконано, розроблена система дозволяє оптимізувати роботу закладів харчування, що дозволить зробити їх більш ефективним в роботі та управлінні, а також дозволить швидко приймати обґрунтовані рішення та добре впливає на автоматизацію закладів харчування, що значно зменшить час прийому та обробки замовлень.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: ДП “УкрНДНЦ”, 2016. – 30 с.
2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп’ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп’ютерно-інтегровані технологічні процеси і виробництва», «Комп’ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2021. 55 с.
3. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс] : Наказ ХНУРЕ від 06 травня 2021 р. № 143. – Режим доступу: [https://nure.ua/wpcontent/uploads/Main\\_Docs\\_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi -universitetu.pdf](https://nure.ua/wpcontent/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi -universitetu.pdf)
4. Невлюдов І.Ш. Основи наукових досліджень: Навч. посібник / І.Ш. Невлюдов, Ю.М. Олександров, А.О. Андрусевич, О.О. Чала. Кривий Ріг: Криворізький коледж НАУ, 2019. 396 с
5. Веснянка В.О. Розроблення інформаційної системи для оптимізації бізнес-процесів закладу харчування //«Автоматизація та приладобудування» ADED-2023, Випуск 2, С. 121-125
6. Шушакова, І., Свистун, А. (2021). Інформаційні технології в управлінні підприємствами ресторанного господарства. Економіка та суспільство, (25).
7. Камушков О., Жилко О., Незвещук-Когут Т. Технології автоматизації системи управління у закладах ресторанного господарства. Економіка та суспільство. № 43 2022. DOI: <https://doi.org/10.32782/2524->

0072/2022-43-48.

8. Ultimate Guide on Business Process Optimization For 2024 URL: <https://kissflow.com/workflow/bpm/business-process-optimization/> (дата звернення: 28.09.2023)

9. Автоматизація бізнес процесів URL:<https://snov.io/glossary/ua/businessprocess-automation-ua/> (дата звернення: 28.09.2023)

10. Lawton, G., & Tucci, L. (2022). Business Process Automation. TechTarget. 2022. URL: <https://www.techtarget.com/searchcio/definition/businessprocess-automation> [in English] (дата звернення: 29.09.2023)

11. Shcho take avtomatyzatsiia biznes-protseviv? [What is business processautomation?] (2022). URL: <https://from-ua.info/shcho-take-avtomatyzatsiia-biznesprotseviv/>(дата звернення: 30.09.2023)

12. Сидорук А. В. Автоматизація системи управління закладами ресторанного господарства / А. В. Сидорук, Г. А. Омеляненко, І. В. Середа // Східна Європа: економіка, бізнес та управління. 2021. № 32. С. 76-82.

13. Дубук, В. І., Хар, К. В. (2022). Розробка автоматизованої системи управління замовленнями послуг харчування. *Міжнародна наукова інтернет-конференція "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення* (випуск 65). (м. Тернопіль, 8-9 лютого 2022 р.). С.6-10

14. Етапи розробки сайту від ідеї до реалізації (2019). URL: [https://www.zhitomir.info/news\\_188373.html](https://www.zhitomir.info/news_188373.html) (дата звернення: 28.11.2023)

15. Тітомир, Л. А. (2021) Інноваційні напрямки розвитку закладів готельного господарства. *Food Industry Economics / Ekonomika Harčovoї Promislovosti*, Vol 13, Issue 1, p. 62

16. Декомпозиція IDEF0 Веб сайт. URL: <http://ukr.vipreshebnik.ru/entsiklopediya/50-d/1792-dekompozitsiya.html> (дата звернення: 08.06.2022)

17. Логістика [Електронний ресурс] : електронний навчальний посібник / О. В. Безсмертна, О. О. Мороз, Т. М. Білоконь, І. В. Шварц. Вінниця : ВНТУ, 2018. 171 с.

18. Тузенко, О., Балалаєва, О., & Лапшун, О. (2023). Моделювання і реінжиніринг бізнес-процесів закладів громадського харчування з використанням методики комплексної оцінки систем та теорії масового обслуговування. *Наука та виробництво*, (25), 64-75.

19. Комар, М. (2020). Інформаційна технологія інтелектуальної обробки та аналізу великих даних. *Вісник Хмельницького національного університету*, №5, 2020 (289). С. 125-130

20. Етапи розробки сайту від ідеї до реалізації. URL:<https://www.geeksforgeeks.org/use-case-diagram/> (дата звернення: 29.11.2023)

21. What is Use Case Diagram? URL:<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/#:~:text=A%20use%20case%20diagram%20is,examples%20of%20use%20case%20diagram> (дата звернення: 30.11.2023)

22. Гризун Л., Міщенко Є. Огляд інструментів frontend розробки // Матеріали Міжнародної науково-практичної конференції “Інформаційні технології та системи”: тези доповідей, 14-15 квітня 2022 р. Х.: ХНЕУ імені Семена Кузнеця. С. 28.

23. A Guide to Backend Databases URL: <https://www.couchbase.com/blog/a-guide-to-backend-databases/> (дата звернення: 4.12.2023)

24. Рейтинг мов програмування 2022. URL: <https://dou.ua/lenta/articles/language-rating-2022/> (дата звернення: 15.12.2023).

25. Чорний Г. Вивчення об’єктно-орієнтованих мов програмування // Матеріали наук.-практ. конференції студ., аспір. і молодих вчених, 27-28 березня 2023 р. Дніпро : ПДАБА, 2023. С. 565-567

26. Жеребцов О.; Іжиков А. Використання функціонального програмування у JavaScript та фреймворках. In: *2022 International Conference on Innovative Solutions in Software Engineering (ICISSE)*. 2022. p. 162

27. Гнатюк, С. С. Що таке API? Вступ до інтерфейсів прикладного програмування. Інформаційні технології та моделювання систем : *збірник праць учасників Всеукраїнської науково-практичної конференції здобувачів*

*вищої освіти і молодих вчених*, 30 березня 2023 р. Житомир : Поліський національний університет, 2023.С. 71

28. Горбачов, Є. І.; Константинова, Л. В. Огляд SQL Server Management Studio для роботи з БД // Матеріали VII Міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”: тези доповідей, 1 листопада 2023 р. Кропивницький: ЦНТУ, 2023. С. 60.

29. Фомішина О. Модульне проектування мережевих журнальних видань як спосіб підвищення якості сприймання контенту // Український інформаційний простір. 2018. №. 1. С. 141-148.

30. Demska A., Yevsieiev V., Kolesnykova T., Tkachenko V. Methods and means of evaluation usability of human-machine interface // Innovations in Publishing, Printing and Multimedia Technologies: International Scientific Practical Conference (Kaunas, 17th-18th of April, 2019). P. 40-46.

31. Про проблему DTO та шляхи її вирішення. URL: <https://dou.ua/forums/topic/43912/> (дата звернення: 16.12.2023)

32. Кравченко С. В. Аналіз сучасного фреймворка ASP.NET CORE для WEB-додатків // «Automation and Development of Electronic Devices» ADED-2023: Kharkiv: Kind of Kharkiv National University of Radio Electronics [electronic edition], 2023. – Part 1. P. 151-155.

33. Create web APIs with ASP.NET Core. URL: <https://learn.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-8.0> (дата звернення: 17.12.2023)

34. About Swagger. URL: <https://swagger.io/about/> (дата звернення: 18.12.2023)

35. ASP.NET Core web API documentation with Swagger / OpenAPI. URL: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger?view=aspnetcore-8.0> (дата звернення: 20.12.2023)

36. Герасимова, З. З.; Біленко, В. І.; Кирєєв, О. М. Техніка безпеки при роботі з персональним комп’ютером // Розвиток системи управління закладом освіти : матеріали Всеукраїнської науково-практичної конференції; Х. : ХНПУ імені Г. С. Сковороди, 2021, С. 49