

Міністерство освіти і науки України  
«Прикарпатський національний університет імені Василя Стефаника»  
Кафедра комп'ютерної інженерії та електроніки  
(повна назва кафедри)

Білянський Андрій Романович  
Andrii Bilianskyi

УДК 004.93

Спеціальність 123 «Комп'ютерна інженерія»  
(шифр та назва спеціальності)

Кваліфікаційна робота  
на здобуття освітньо-кваліфікаційного рівня магістр  
(бакалавр, спеціаліст, магістр)

Реалізація інформаційної системи для пошуку та перегляду  
закладів харчування на основі фреймворку Next.js

Implementation of an information system for searching and  
reviewing food establishments based on the Next.js framework

Науковий керівник:  
к. т. н., доцент Грига В. М.

Рецензент:  
к. ф.-м. н., професор Никируй Л. І.

Івано-Франківськ  
2024



## АНОТАЦІЯ

Магістерська магістерська кваліфікаційна робота присвячена проектуванню та реалізації інформаційної системи для пошуку та перегляду закладів харчування на основі фреймворку Next.js.

В роботі розроблено систему для пошуку та перегляду закладів харчування на основі фреймворку Next.js, яка дозволяє переглядати заклади харчування по містах, фільтрувати заклади за певними критеріями, переглядати детальну інформацію про заклад: фотографії, опис, контактну інформацію, графік роботи, меню, спеціальні пропозиції та відгуки.

Пояснювальна записка до магістерської кваліфікаційної роботи «Реалізація інформаційної системи для пошуку та перегляду закладів харчування на основі фреймворку Next.js» містить: 68 ст., 67 рис., 22 джерела.

## ABSTRACT

The master's thesis is devoted to the design and implementation of the information system for searching and browsing food establishments based on the Next.js framework.

The work develops a system for searching and browsing food establishments based on the Next.js framework, which allows you to browse food establishments by city, filter establishments by certain criteria, view detailed information about the establishment: photos, description, contact information, opening hours, menu, special offers and reviews.

The explanatory note to the master's thesis " Implementation of an information system for searching and reviewing food establishments based on the Next.js framework" contains: 68 p., 67 fig., 22 sources.

					123.КІ(М)-24.03			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		Білянський А. Р.			Анотація	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевірів</i>		Грига В.М					3	3
<i>Н. Контр.</i>								
<i>Затвердив</i>								

Міністерство освіти і науки України  
«Прикарпатський національний університет імені Василя Стефаника»  
Фізико-технічний факультет  
Кафедра «Комп'ютерної інженерії та електроніки»

Пояснювальна записка  
до кваліфікаційної роботи на тему:

Реалізація інформаційної системи для пошуку та перегляду  
закладів харчування на основі фреймворку Next.js

									Арк.А
									4
Зм.З	Арк.А	№ докум.№	Підпис	Дата	123.КІ(М)-24.03				

## ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1. Аналіз характеристики предметної області	11
1.2. Аналіз існуючих рішень	12
1.3. Висновки до розділу 1	14
РОЗДІЛ 2. ВИБІР ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ	15
2.1. Вибір архітектури проекту	15
2.2. Вибір засобів для реалізації клієнтської частини	15
2.3. Вибір засобів для реалізації серверної частини	17
2.4. Вибір програмного забезпечення для розробки інформаційної системи	18
2.5. Висновки до розділу 2	19
РОЗДІЛ 3. АНАЛІЗ ВИМОГ ТА РЕАЛІЗАЦІЯ	20
3.1. Розроблення вимог до системи та її компонентів	20
3.2. Розроблення функціональної, інфологічної схеми системи та опис структури проекту	24
3.2.1. Розроблення функціональної схеми системи	24
3.2.2. Розроблення інфологічної схеми системи	24
3.2.3. Опис структури проекту	32
3.3. Висновки до розділу 3	33
РОЗДІЛ 4. ОПИС ІНТЕРФЕЙСУ КОРИСТУВАЧА	35
4.1. Опис роботи інформаційної системи для користувача із роллю «клієнт»	35
4.1.1. Сторінка вибору міста	35
4.1.2. Список закладів у вибраному місті	36
4.1.3. Перегляд інформації про заклад	37
4.1.4. Перегляд меню закладу	37
4.1.5. Відгуки про заклад	38
4.1.6. Сервіс кешбеку	39
4.1.7. Сторінка зі спеціальними пропозиціями закладів	40
4.1.8. Пошук закладів та страв	40

						Арк.А
						5
Зм.З	Арк.А	№ докум.№	Підпис	Дата	123.КІ(М)-24.03	

4.1.9. Пошук збережених закладів та страв	41
4.1.10. Карта закладів	42
4.1.11. Автентифікація	42
4.1.12. Керування обліковим записом	43
4.1.13. Додавання нового закладу	45
4.2. Опис роботи інформаційної системи для користувача із роллю «ресторан»	47
4.2.1. Автентифікація	47
4.2.2. Сторінка про заклад	49
4.2.3. Сторінка зі стравами	51
4.3. Висновки до розділу 4	55
ВИСНОВКИ	56
СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ	57
ДОДАТОК А	59
ДОДАТОК Б	61
ДОДАТОК В	63

									Арк.А
									6
Зм.З	Арк.А	№ докум.№	Підпис	Дата	123.КІ(М)-24.03				

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

1. ПЗ – програмне забезпечення;
2. UML – Unified Modeling Language (уніфікована мова моделювання);
3. БД – база даних;
4. API – прикладний програмний інтерфейс;
5. Десктопна версія – версія ПЗ для комп'ютерів;
6. Мобільна версія – версія ПЗ для мобільних пристроїв;
7. Бібліотека – збірка об'єктів чи підпрограм для вирішення близьких за тематикою задач;
8. Фреймворк – інфраструктура програмних рішень, що полегшує розробку складних систем;
9. СКБД – система управління базами даних;
10. ІС – інформаційна система.

									Арк.А
									7
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

123.КІ(М)-24.03

## ВСТУП

Ідея створення інформаційної системи для закладів харчування виникла в січні 2022 року під час пандемії COVID-19. У ті часи було популярне використання QR-кодів для перегляду меню. Найчастіше посилання з QR-коду перенаправляло користувача на файл в форматі PDF, але ним було незручно користуватися на мобільних телефонах через розміри файлу: весь час треба було приближувати і прокручувати меню, щоб щось прочитати. Після того виникла ідея створити онлайн-меню, яке буде адаптованим для телефонів, і його можна буде легко переглядати. Проектована система повинна мати назву страви, опис, ціни, фотографії, тобто мати всю необхідну інформацію, щоб гість міг мати уявлення про те, що він буде замовляти. Цей сервіс отримав назву In-Phone [1], оскільки дозволяє виконувати всі дії з телефону (взаємодія із застосунком зі сторони гостя та закладу). Після запуску цієї інформаційної системи було прийнято рішення розширити функціонал, зробивши окрему інформаційну систему, призначену для пошуку та перегляду закладів харчування.

В роботі розроблено систему для пошуку та перегляду закладів харчування на основі фреймворку Next.js, яка дозволяє переглядати заклади харчування по містах, фільтрувати заклади за певними критеріями, переглядати детальну інформацію про заклад: фотографії, опис, контактну інформацію, графік роботи, меню, спеціальні пропозиції та відгуки.

Після проведених інтерв'ю із власниками закладів харчування, було зроблено висновок, що існуючі на той момент інструменти вимагають значних вкладень часу та зусиль з обох сторін, щоб їх об'єднати, а люди витрачають в середньому 30 хвилин на те, щоб вибрати куди піти, враховуючи побажання всіх сторін: меню та ціни, інтер'єр та відгуки.

EatWу – це платформа, яка допоможе знайти заклад на будь-який смак та дасть відповідь на питання “Куди піти?” у кілька кліків, заощадити гроші та запросити нового гостя для бізнесу [2].

						123.КІ(М)-24.03	Арк.А
							8
Зм.З	Арк.А	№ докум.№	Підпис	Дата			



Для гостей це:

- Економія часу на пошуки “Куди піти?”
- Знайти саме те, що вони хочуть
- Отримувати кешбек з кожного відвідування
- І витратити кешбек в інших закладах

Для бізнесу – це новий канал залучення гостей, вартість якого нижча за прибуток від них. EatWу це:

- Платформа з цільовою аудиторією закладів
- Місце, куди інтегровані всі інструменти для HoReCa (онлайн-меню, доставка та інше)
- Відстежування гостей
- Зменшення витрат на SMM до 40%

Для реалізації інформаційної системи було вибрано фреймворк Next.js – це вебфреймворк з відкритим вихідним кодом, створений компанією Vercel, що дозволяє розробляти веб-додатки на основі React із рендерингом на стороні сервера та генерацією статичних веб-сайтів [3]. Також використано API сервісу QR-меню In-Phone, який розроблений за допомогою стеку MERN (MongoDB, Express, React, Node.js).

Розроблена система може бути застосована для пошуку закладів харчування за певними критеріями (місто, тип закладу, тип кухні, розташування закладу, наявність певних страв чи їх комбінацій і т.д.). Перевагами розробленої системи є простота користування, швидкість роботи. До недоліків можна віднести важкість підтримки актуальних даних.

									Арк.А
									9
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

**Мета роботи:** основною метою цієї магістерської роботи є розроблення інформаційної системи для пошуку та перегляду закладів харчування.

**Основні завдання, які необхідно вирішити:**

- Технічна реалізація. Система має бути простою, за потреби має бути легко масштабованою. Щодо функціоналу, інформаційна система має мати такі можливості: авторизація користувачів, робота із набором страв/товарів, робота із інформацією про заклад, оброблення та адміністрування даних.

**Об'єкт дослідження:** інформаційна система для пошуку та перегляду закладів харчування.

**Предмет дослідження:** є серверна та клієнтська версія інформаційної системи для пошуку та перегляду закладів харчування.

									Арк.А
									10
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

123.КІ(М)-24.03

## РОЗДІЛ 1. ХАРАКТЕРИСТИКА ПРЕДМЕТНОЇ ОБЛАСТІ

Інформаційна система для пошуку та перегляду закладів харчування використовується для полегшення процесу пошуку, вибору та оцінки ресторанів, кафе, барів та інших місць громадського харчування для кінцевих користувачів, таких як відвідувачі, туристи або місцеві жителі. Вона також може слугувати корисним інструментом для власників закладів, які прагнуть популяризувати свій бізнес, підвищити лояльність клієнтів та отримати зворотний зв'язок.

### 1.1. Аналіз характеристики предметної області

Характеристика предметної області:

- Категорії закладів:
  - Види закладів: ресторани, кафе, бари, їдальні, заклади швидкого харчування, паби.
  - Спеціалізація кухні: українська, італійська, японська, китайська, фаст-фуд, вегетаріанська, веганська та інші.
  - Доступність різних видів харчування: сніданки, обіди, вечері.
- Користувацький досвід (UX):
  - Пошук: за типом кухні, розташуванням, рейтингом, ціною категорією, часом роботи та додатковими послугами.
  - Рейтинги та відгуки: можливість переглядати відгуки, оцінки, рекомендації інших користувачів.
  - Візуалізація даних: карти з розташуванням, фото інтер'єру та страв.
- Функціональність для адміністраторів:
  - Модерація відгуків, управління рейтингами.
  - Інтеграція з соціальними мережами, можливість публікувати спеціальні пропозиції, події, акції.
- Технологічні аспекти:
  - Інтеграція з картографічними сервісами для відображення місцезнаходження закладів.

						123.КІ(М)-24.03	Арк.А
							11
Зм.З	Арк.А	№ докум.№	Підпис	Дата			

- Адаптивність інтерфейсу для мобільних пристроїв.
- Використання баз даних для зберігання інформації про заклади, відгуки та оцінки користувачів.
- Можливість розробки додаткових функцій, наприклад, онлайн-резервації столиків, замовлення їжі на винос або доставки.
- Цільова аудиторія:
  - Кінцеві користувачі (місцеві жителі, туристи, інші відвідувачі), які шукають місце для прийому їжі чи дозвілля.
  - Власники та менеджери закладів, які прагнуть залучити більше клієнтів і підтримувати якість обслуговування на високому рівні.

## 1.2. Аналіз існуючих рішень

Конкуруючі готові рішення можна поділити на три категорії:

- Сервіси доставки. Наприклад, Glovo [4] (рис. 1.1) та Bolt Food мають функціонал пошуку закладів, систему рейтингу, рекомендації страв та пошук по стравах.

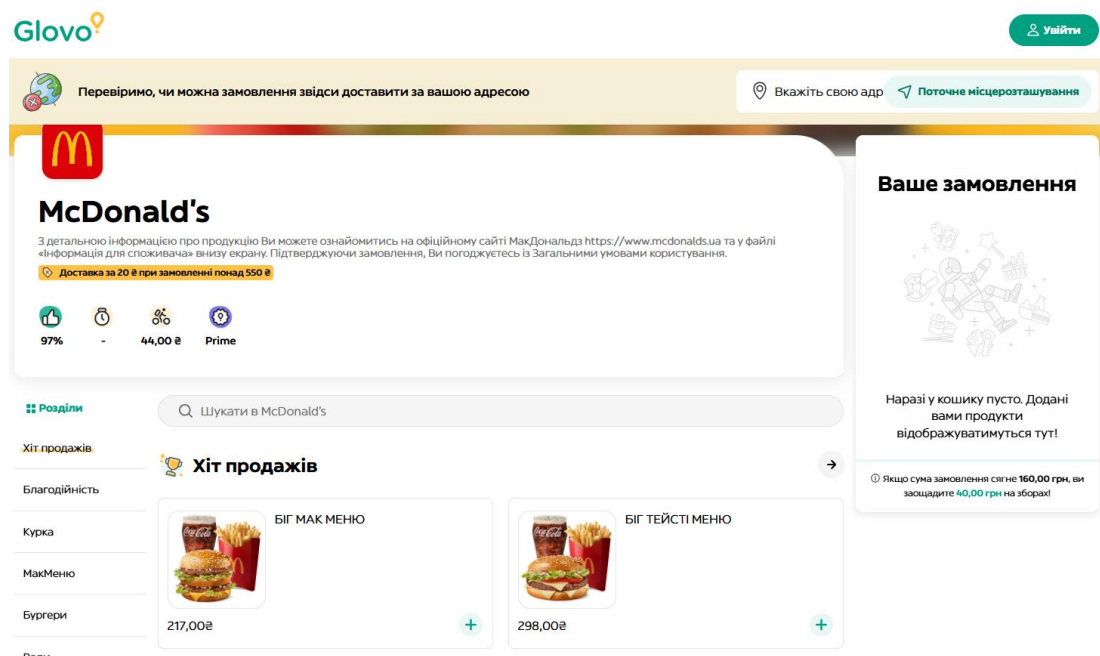


Рис. 1.1. Користувальницький інтерфейс сервісу Glovo

- Сервіси пошуку. Наприклад, Google Maps [5] (рис. 1.2) і TripAdvisor мають функціонал пошуку закладів, систему рейтингу, рекомендації страв.

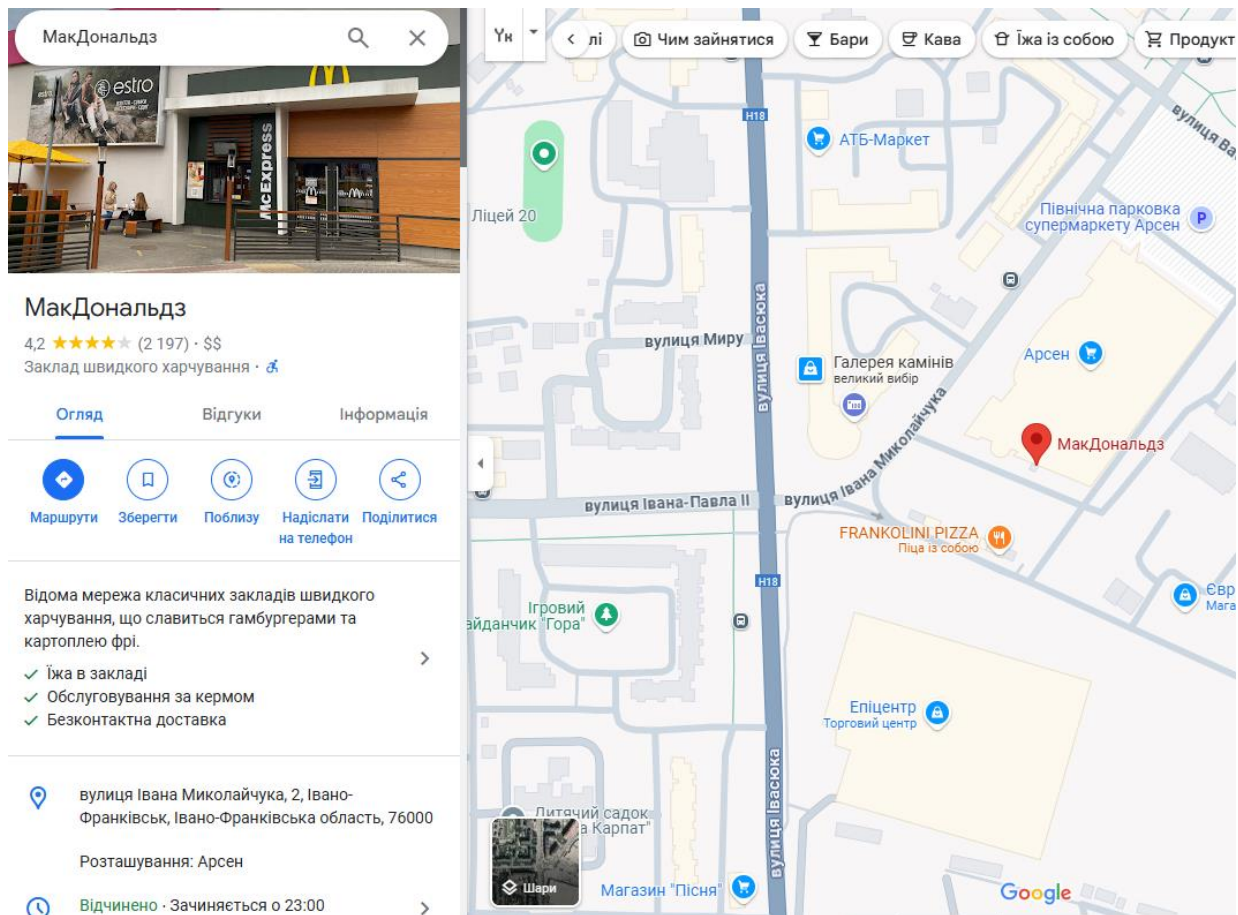


Рис. 1.1. Користувальницький інтерфейс сервісу Google Maps

- Соцмережі. Наприклад, Instagram [6] (рис. 1.3) та Tiktok мають функціонал рекомендації закладів та функціонал поширення інформації про якісь події.

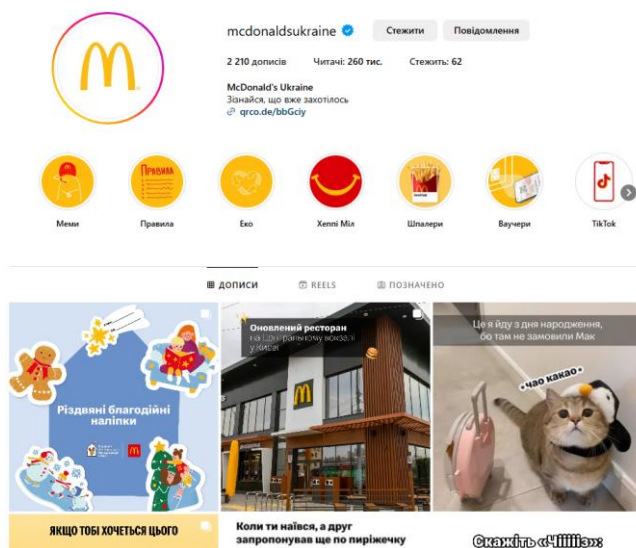






Рис. 1.3. Користувальницький інтерфейс сервісу Instagram

Зм.3	Арк.А	№ докум.№	Підпис	Дата

Але жоден із цих сервісів не надає цей увесь функціонал одночасно.

Щодо економічної складової, то для закладу EatWу буде ефективнішим за інші сервіси. Згідно проведеного дослідження ринку, заклад платить сервісам доставки приблизно 33% від чеку замовлення, а на соцмережі заклади харчування витрачають більше 20% доходу. EatWу надає весь функціонал за 15% від чеку замовлення, а також у майбутньому приведе ще раз гостя до закладу для того, щоб він витратив кешбек (рис. 1.4)

<b>Конкуренти</b>				
<b>Список закладів</b>	✓	✓	✓	✗
<b>Рейтинг</b>	✓	✓	✓	✗
<b>Рекомендації</b>	✓	✓	✓	✓
<b>Пошук по стравах</b>	✓	✓	✗	✗
<b>Події</b>	✓	✗	✗	✓
<b>Кешбек</b>	✓	✗	✗	✗
<b>Витрати закладу</b>	<b>15%</b>	<b>33%</b>		<b>22%</b>

Від чеку гостя Від доходу

Рис. 1.4. Порівняльна таблиця існуючих рішень

### 1.3. Висновки до розділу 1

У першому розділі описано функціонал розроблюваної системи, а також проведено аналіз існуючих рішень і визначено їх переваги та недоліки у порівнянні з сервісом EatWу.

## РОЗДІЛ 2. ВИБІР ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ

### 2.1. Вибір архітектури проекту

Для реалізації інформаційної системи було обрано клієнт-серверну архітектуру, яка складається з двох компонентів: клієнта та сервера.

Клієнт – це пристрій, що належить користувачу, і який відправляє запити до сервера, щоб отримати інформацію або виконати певні дії.

Сервер – це пристрій, який обслуговує запити клієнтів, надає доступ користувачам до ресурсів, зберігає інформацію в базах даних.

У такій системі клієнт відправляє запит на сервер, де він оброблюється, а результат відправляється назад до клієнта. [7]

Переваги клієнт-серверної архітектури:

- Розподілена обробка даних та функцій між клієнтом та сервером забезпечує більшу ефективність та масштабованість системи.
- Централізоване управління та контроль над доступом до даних та ресурсів.
- Зменшується навантаження на клієнтські пристрої.
- Підвищується рівня безпеки за рахунок централізованого контролю доступу та автентифікації користувачів. [8]

### 2.2. Вибір засобів для реалізації клієнтської частини

Під час виконання завдання було використано такі технології в клієнтській частині:

- Next.js – це вебфреймворк з відкритим вихідним кодом, створений компанією Vercel, що дозволяє розробляти веб-додатки на основі React із рендерингом на стороні сервера та генерацією статичних веб-сайтів. [3]
- TypeScript – це мова програмування, розроблена компанією Microsoft, яка надає можливості розширення функціоналу JavaScript для розробки веб-застосунків. Оскільки TypeScript базується на

						123.KI(M)-24.03	Арк.А
							15
Зм.З	Арк.А	№ докум.№	Підпис	Дата			

JavaScript, він є зворотно сумісним і після компіляції можна використовувати програми в будь-якому браузері або на серверній платформі Node.js. [9]

- JavaScript – це динамічна мова програмування, що базується на об'єктно-орієнтованій прототипній парадигмі та реалізована відповідно до стандарту ECMAScript. JS використовується для створення скриптів на веб-сторінках, які дають можливість користувачам взаємодіяти з вмістом веб-сторінки, керувати браузером, асинхронно взаємодіяти з сервером, змінювати структуру та вигляд веб-сторінки, та інші функції. [10]
- React – це JavaScript бібліотека з відкритим кодом, яка призначена для створення інтерфейсів користувача. Вона призначена для вирішення проблем часткового оновлення вмісту веб-сторінок, з якими стикаються розробники односторінкових застосунків. Розробляється компанією Meta, а також групою індивідуальних розробників. [11]

React дозволяє зробити процес створення інтерактивних інтерфейсів значно простішим. Для цього достатньо описати, як виглядають різні частини інтерфейсу в кожному стані додатку, і React автоматично оновлює та візуалізовує тільки необхідні компоненти, коли дані змінюються.

React пропонує декларативний підхід до створення інтерфейсів, що означає, що код містить опис того, що має бути виконано, а не як це робити. Це робить код більш передбачуваним та дозволяє легше налагоджувати програму. [12]

- Redux – це JavaScript бібліотека, яка дозволяє ефективно керувати станом програми. Зазвичай вона використовується в поєднанні з React або Angular для створення інтерфейсів користувача. [13]
- SCSS (SASS) – це скриптова метамова, яка перетворюється в каскадні таблиці стилів (CSS). SCSS розширює стандартний CSS, дозволяючи використовувати змінні, вкладені селектори, функції та

						Арк.А
					123.КІ(М)-24.03	16
Зм.З	Арк.А	№ докум.№	Підпис	Дата		



багато іншого. SCSS має схожий синтаксис з CSS, але дозволяє програмувати стилі ефективніше та зручніше. Перед тим, як використовувати CSS, код SCSS потрібно скомпілювати в CSS. [14]

- Chart.js – це бібліотека JavaScript для створення діаграм на основі HTML. Це одна з найпростіших бібліотек візуалізації для JavaScript, яка містить такі вбудовані типи діаграм: точкова діаграма, лінійна діаграма, гістограма, секторна діаграма, кругла діаграма, бульбашкова діаграма, діаграма з областями, радарна діаграма та змішана діаграма. [15]

### 2.3. Вибір засобів для реалізації серверної частини

Технології, які використані під час реалізації серверної частини:

- Next.js
- Node.js – відкрита платформа для високопродуктивного виконання мережевих застосунків, що була розроблена на мові JavaScript. Перевагою Node.js є те, що вона дозволяє виконувати JavaScript-скрипти на сервері, що раніше було неможливо. Це перетворило JavaScript з мови, яка використовувалась виключно на стороні клієнта, на мову, яку можна використовувати для написання мережевих застосунків та програм на сервері. [16]
- TypeScript.
- Express.js – це безкоштовний відкритий каркас розробки серверних веб-застосунків для Node.js, який використовується для створення веб-застосунків та API. Він є одним з найпопулярніших каркасів для розробки серверної частини Node.js і вважається стандартом для створення веб-застосунків з використанням Node.js. [17]
- MongoDB – це СКБД з відкритим вихідним кодом, яка зберігає дані в форматі документів і не вимагає фіксованої схеми таблиць. Вона займає проміжне місце між системами зберігання даних у форматі

ключ/значення, що дозволяють швидко масштабувати, і реляційними СКБД, які зручні для складних запитів.

MongoDB має наступні можливості:

- просте та потужне сховище даних у форматі JSON;
- масштабованість і стійкість до відмов;
- гнучка мова для формування запитів;
- динамічні запити;
- профілювання запитів;
- швидкі оновлення;
- журналювання операцій зміни даних в базі даних;
- підтримка методів Map/Reduce. [18]

#### **2.4. Вибір програмного забезпечення для розробки інформаційної системи**

Програмне забезпечення, яке використовувалося для реалізації інформаційної системи:

- Visual Studio Code – інструмент для створення, редагування та налагодження веб-застосунків та програм для хмарних платформ, який надається безкоштовно і працює на операційних системах Windows, Linux та OS X. Цей засіб розроблено компанією Microsoft. [19]
- Postman – це платформа, яка допомагає створювати та використовувати API, спрощуючи кожен етап життєвого циклу цих інтерфейсів. Також, Postman використовується для тестування API. [20]
- MongoDB Compass – це безкоштовний інтерактивний інструмент для роботи з базами даних MongoDB, що дозволяє виконувати запити, оптимізувати та аналізувати дані. [21]

									Арк.А
									18
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

## 2.5. Висновки до розділу 2

У другому розділі проведено вибір засобів для реалізації інформаційної системи для пошуку та перегляду закладів харчування, а саме: описано вибраний архітектурний шаблон (це клієнт-серверна архітектура. Її перевагами є те, що за рахунок розподілення виконання коду на сервері і клієнтському пристрої, вона допомагає зменшити навантаження на пристрій клієнта, збільшити безпеку (наприклад, з'єднання з БД відбувається на сервері) та підвищити продуктивність); описано технології, які використані для розробки клієнтської (Next.js, TypeScript, JavaScript, React, Redux, SCSS, Chart.js) і серверної (Next.js, Node.js, TypeScript, Express.js, MongoDB) частини проекту, а також ПЗ, яке використовувалося для реалізації проекту (Visual Studio Code, Postman, MongoDB Compass).

									Арк.А
									19
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

### РОЗДІЛ 3. АНАЛІЗ ВИМОГ ТА РЕАЛІЗАЦІЯ

Цей розділ містить вимоги до роботи, функціональну та інфологічну схеми системи, бізнес-логіку інформаційної системи.

#### 3.1. Розроблення вимог до системи та її компонентів

Тут наведено мінімальні вимоги до інформаційної системи, які стосуються створення та обробки замовлень:

- Авторизація користувачів.

Можливість реєстрації та авторизації користувачів з такими ролями: «ресторан», «працівник», «клієнт», «адміністратор».

- Робота із набором страв/товарів.

Користувач із роллю «ресторан» може додати страву/товар, змінити статус (чи є в наявності), змінити дані (назва, опис, вартість), видалити страву або товар.

- Оброблення та адміністрування даних.

Система зберігає введену користувачем інформацію в базі даних.

Користувач із роллю «ресторан» або «працівник» має мати можливість прочитати, змінити або видалити інформацію про страви/товари, а також різну інформацію про заклад.

Користувач із роллю «адміністратор» може переглянути інформацію про користувачів із роллю «ресторан» (кількість замовлень, кількість страв, рахунки, графіки та іншу інформацію).

- Звітність.

Система містить такі розділи статистики:

- Рахунки. При переході на сторінку про рахунок, система відображає детальну інформацію про нього (сума, дата, замовлення) та дозволяє експортувати рахунок в формат PDF.
- Доставка. Той самий функціонал, що в рахунках, тільки ще додається інформація про місце, дату і тип доставки.
- Входи на меню. Інформація про входи на онлайн-меню закладу.

									Арк.А
									20
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

- Входи на доставку. Інформація про входи на онлайн меню з можливістю доставки.
- Відгуки. Список відгуків, які написані гостями закладу.

Функції інформаційної системи поділені за ролями користувачів. Існує чотири ролі користувачів:

- Клієнт – це гість закладу, який переглядає список закладів через інформаційну систему, переходить за посиланням або сканує QR-код.

Для клієнта система дозволяє:

- шукати заклади харчування за певними критеріями;
- переглядати інформацію про заклад;
- переглядати онлайн-меню;
- залишати відгуки про заклад.

- Ресторан – це обліковий запис закладу, який має доступ до всього функціоналу ІС, який стосується ресторану.

Для ресторану система дозволяє:

- додати страву;
- змінити статус страви;
- змінити інформацію про страву;
- видалити страву;
- переглянути інформацію про страву;
- переглянути рахунки;
- змінити інформацію про заклад, яка буде відображатися в онлайн-меню;
- аналізувати роботу закладу в режимі реального часу за допомогою контрольної панелі;
- переглядати статистику;
- керувати персоналом (реєструвати працівників, створювати та редагувати ролі працівників).

									Арк.А
									21
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

- Працівник – це обліковий запис працівника закладу (наприклад, адміністратор закладу, офіціант, бармен). Працівник має доступ до функціоналу, який визначений його дозволами. Дозволи для працівників містяться в інформації про ролі працівників (тобто ресторан створює нові ролі з певними дозволами і призначає їх працівникам).
- Адміністратор – це обліковий запис, у якому відображено всю інформацію про всі заклади. Доступ до нього має тільки розробник інформаційної системи.

Для адміністратора система дозволяє:

- реєструвати нові заклади;
- переглядати всю інформацію про заклад, працівників закладу: статистика відвідувань закладу, список замовлень, інформацію про страви.

									Арк.А
									22
Зм.З	Арк.А	№ докум.№	Підпис	Дата	123.КІ(М)-24.03				

UML-діаграма прецедентів складається з чотирьох акторів (адміністратор, ресторан, працівник і клієнт) та двох платформ для дії (сайт та адміністративна панель) (рис. 3.1).

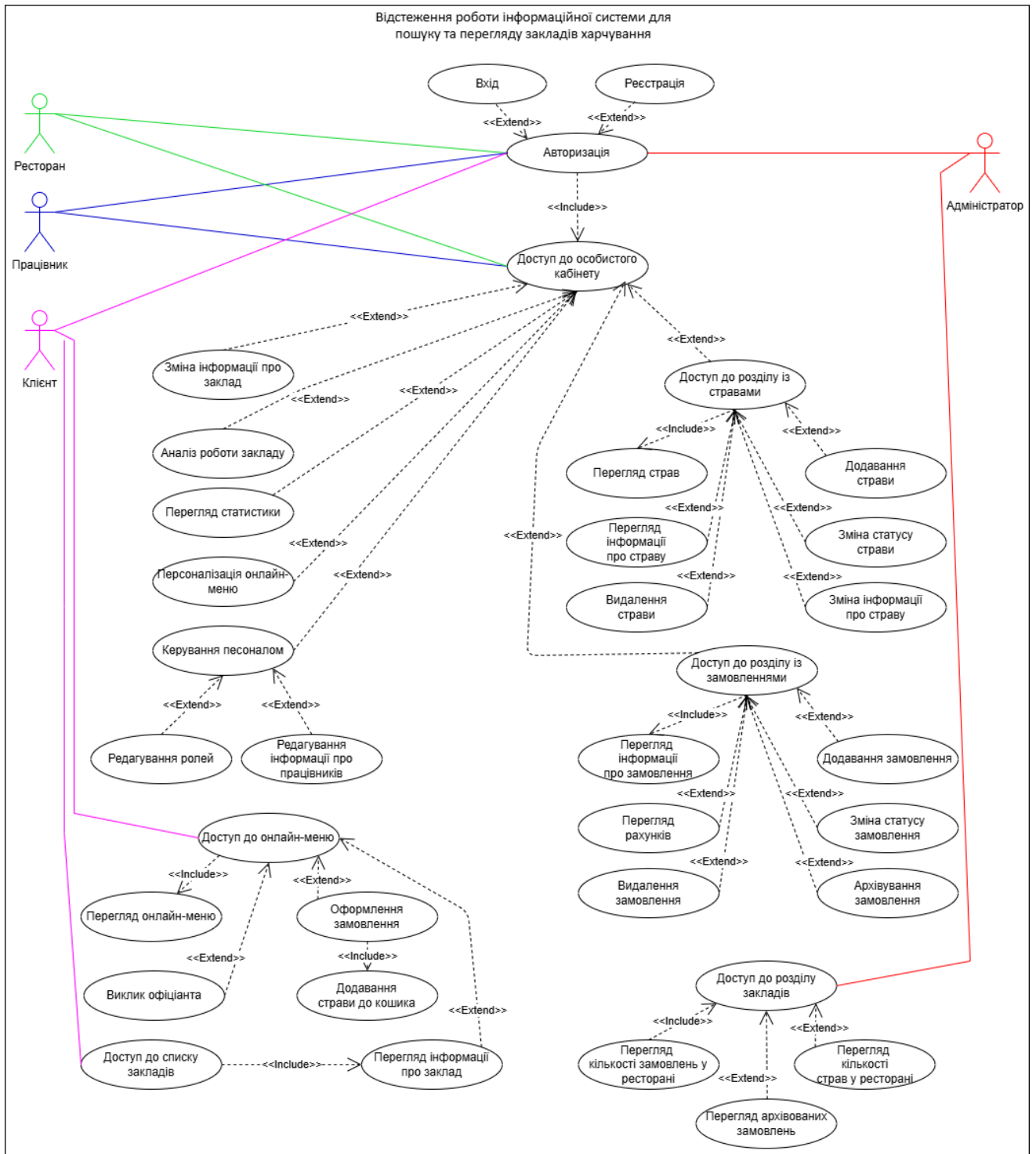


Рис. 3.1. UML-діаграма прецедентів

## 3.2. Розроблення функціональної, інфологічної схеми системи та опис структури проекту

### 3.2.1. Розроблення функціональної схеми системи

Інформаційна система для закладу харчування реалізована за допомогою клієнт-серверної архітектури. На пристрої клієнта в браузері за допомогою фреймворку Next.js запущено веб-додаток, який зв'язується із сервером по API. Сервер працює на платформі Node.js, на якій за допомогою фреймворку Next.js працює програма серверу. Сервер з'єднаний з базою даних MongoDB (рис. 3.2).

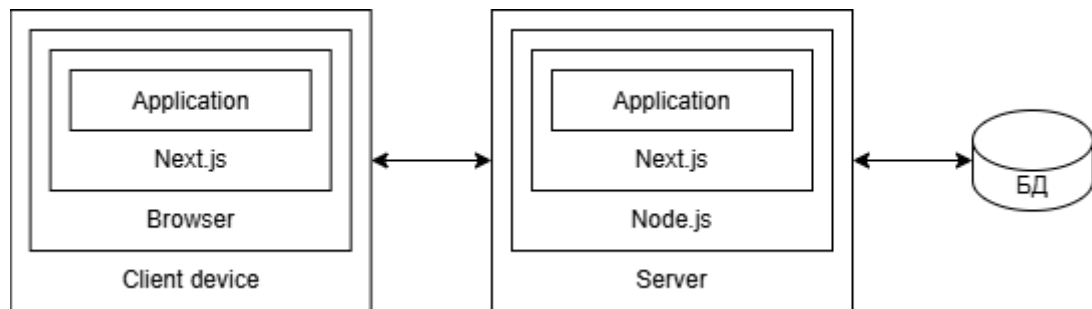


Рис. 3.2. Функціональна схема системи

Приклад роботи інформаційної системи під час авторизації користувача:

- Користувач вводить електронну адресу та пароль на сторінці входу.
- Після натискання на кнопку «Увійти» клієнт робить запит до сервера по API.
- Сервер перевіряє чи існує такий користувач. Якщо існує, то отримує інформацію про користувача і відправляє назад. Інакше, сервер повертає помилку.
- Після успішного запиту, клієнт зберігає в локальній пам'яті інформацію про користувача: це необхідно для того, щоб дані не видалялися при перезавантаженні сторінки. І далі надає доступ до облікового запису.

### 3.2.2. Розроблення інфологічної схеми системи

Для цієї інформаційної системи обрано базу даних MongoDB. Оскільки MongoDB є нереляційною базою даних, то у ній немає зв'язків. Замість терміну



«таблиця» в MongoDB використовується термін «схема», і «документ» замість «рядок».

MongoDB – це нереляційна система баз даних, яка пропонує покращену гнучкість і горизонтальну масштабованість, порівняно із реляційними базами даних (наприклад, MySQL) [22]. Підходить для збереження великих обсягів даних.

Для зв'язування даних вибрано ключ restaurantID. Цей ключ є у всіх схемах, які пов'язані із закладом, тобто всі дані закладу зв'язуються за допомогою ключа restaurantID.

База даних містить 19 схем:

- admins – інформація про користувачів з роллю «адміністратор»;
- archives – інформація про архівовані замовлення;
- bills – інформація про рахунки;
- clientlogins – інформація про входи на онлайн-меню закладу;
- clients – інформація про гостей закладу, які створили замовлення, і воно ще не завершено;
- deliveries – інформація про рахунки доставки;
- errors – інформація про помилки;
- feedbackrestaurants – інформація про відгуки в закладах;
- feedbacks – інформація про заявки, залишені щодо відгуку про сервіс EatWy;
- items – інформація про страви;
- logs – інформація про логи;
- notifications – інформація про сповіщення;
- orders – інформація про замовлення;
- restaurantlogins – інформація про входи закладів в обліковий запис;
- restarurants – інформація про заклади;
- roles – інформація про ролі працівників закладу;
- tabs – інформація про групи і підгрупи страв;

									Арк.А
									25
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

123.КІ(М)-24.03

- workers – інформація про працівників закладу.

Опис деяких схем бази даних (рис. 3.3-3.6):

1. Ресторани (Restaurants). Містить інформацію про заклад. Складається з 33 полей.

```

const restaurantSchema = new mongoose.Schema({
  username: { type: String },
  password: { type: String },
  token: { type: String },
  createdAt: { type: String },
  role: { type: String },
  isBlocked: { type: Boolean },
  mainLanguage: { type: String },
  restaurantInformation: { type: Object },
  restaurantLanguages: { type: Array },
  tables: { type: Array },
  priceFormat: { type: String },
  onlyMenu: { type: Boolean },
  canChangeOnlyMenuStatus: { type: Boolean },
  waiterOnlyMenuStatus: { type: Boolean },
  deliveryStatus: { type: Boolean },
  institutionType: { type: String },
  restaurantCustomization: { type: Object },
  enabledGames: { type: Boolean },
  telegramChatID: { type: String },
  sendNotificationsInTelegram: { type: Boolean },
  sendFeedbackNotificationsInTelegram: { type: Boolean },
  reviewLink: { type: String },
  showReviewLink: { type: Boolean },
  paidFeatures: { type: Array },
  personalData: { type: Object },
  tariffInfo: { type: Object },
  payUsageInfo: { type: Object },
  deliveryInfo: { type: Object },
  hasMadeOrdersColumn: { type: Boolean },
  seoTags: { type: Array },
  apiLogin: { type: String },
});

  _id: ObjectId('637a1439d14f2de107109904')
  username: "restaurant_demo"
  password: "$2b$10$Um2bddQRPvZJGt/Ff9E.veq
  token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXV
  createdAt: "2022-11-20T11:49:13.663Z"
  role: "restaurant"
  isBlocked: false
  mainLanguage: "uk"
  > restaurantLanguages: Array
  > tables: Array
  _v: 0
  > restaurantInformation: Object
  priceFormat: "zero"
  canChangeOnlyMenuStatus: true
  onlyMenu: false
  waiterOnlyMenuStatus: true
  institutionType: "restaurant"
  > restaurantCustomization: Object
  enabledGames: true
  sendNotificationsInTelegram: false
  telegramChatID: ""
  reviewLink: ""
  showReviewLink: true
  > paidFeatures: Array
  deliveryStatus: true
  > confirmedContracts: Array
  mainEmail: "email@gmail.com"
  > personalData: Object
  > tariffInfo: Object
  > payUsageInfo: Object
  sendFeedbackNotificationsInTelegram: true
  > deliveryInfo: Object
  hasMadeOrdersColumn: true
  > seoTags: Array

```

а

б

Рис. 3.3. Схема в БД, а – опис схеми Restaurants, б – запис у схемі Restaurants

Ця схема містить такі поля:

- `_id` (ідентифікатор, тип: `ObjectId`, обов'язкове поле);
- `username` (ім'я користувача, тип: рядок);
- `password` (пароль, тип: рядок);
- `token` (це токен авторизації, тип: рядок);

- `createdDate` (дата реєстрації, тип: рядок);
- `role` (роль користувача, тип: рядок);
- `isBlocked` (чи заклад заблокований; тип: логічний);
- `mainLanguage` (основна мова закладу, тип: рядок);
- `restaurantInformation` (інформація про заклад, тип: об'єкт);
- `restaurantLanguages` (масив доступних мов закладу, тип: масив);
- `tables` (список столів в закладі, тип: масив об'єктів);
- `priceFormat` (формат відображення цін, тип: рядок);
- `onlyMenu` (чи вимкнено замовлення до столу, тип: логічний);
- `canChangeOnlyMenuStatus` (чи може заклад вмикати замовлення до столу, тип: логічний);
- `waiterOnlyMenuStatus` (чи вимкнено виклик офіціанта в закладі, тип: логічний);
- `deliveryStatus` (чи ввімкнена доставка, тип: логічний);
- `institutionType` (тип закладу, тип: рядок);
- `restaurantCustomization` (налаштування персоналізації закладу, тип: об'єкт);
- `enabledGames` (чи ввімкнені ігри в кошику, тип: логічний);
- `telegramChatID` (ID чату в Telegram, тип: рядок);
- `sendNotificationsInTelegram` (чи відправляти сповіщення в Telegram, тип: логічний);
- `sendFeedbackNotificationsInTelegram` (чи відправляти сповіщення з відгуками в Telegram, тип: логічний);
- `reviewLink` (посилання на сторінку відгуків про заклад, тип: рядок);
- `showReviewLink` (чи відображати посилання на сторінку відгуків, тип: логічний);
- `paidFeatures` (список доступних платних функцій для закладу, тип: масив);
- `personalData` (персональні дані закладу, тип: об'єкт);

										Арк.А
										27
Зм.З	Арк.А	№ докум.№	Підпис	Дата						

123.КІ(М)-24.03

- tariffInfo (інформація про тарифний план, тип: об'єкт);
  - payUsageInfo (інформація про оплату використання, тип: об'єкт);
  - deliveryInfo (інформація про доставку, тип: об'єкт);
  - hasMadeOrdersColumn (чи є колонка "приготовані замовлення" на сторінці із замовленнями, тип: логічний);
  - seoTags (теги для SEO, тип: масив);
  - apiLogin (логін для доступу до API інших POS-систем, тип: рядок).
2. Страви (Items). Містить інформацію про страви. Складається з 24 полей.

```
const itemSchema = new mongoose.Schema({
  _id: { type: mongoose.Schema.Types.ObjectId },
  index: { type: Number },
  title: { type: Object },
  description: { type: Object },
  itemParameter: { type: Object },
  transliteratedTitle: { type: String },
  priceInfo: { type: Array },
  restaurantID: { type: String },
  inStock: { type: Boolean },
  isArchived: { type: Boolean },
  createdAt: { type: String },
  lastUpdateDate: { type: String },
  itemImages: { type: Array<string> },
  itemGroup: { type: String },
  itemSubGroup: { type: String },
  offeredItems: { type: Array },
  imageType: { type: String },
  isNewItem: { type: Boolean },
  notForDelivery: { type: Boolean },
  likesCount: { type: Number },
  allergens: { type: Array },
  allergensManual: { type: Array },
  isPoster: { type: Boolean },
  showPosterTitle: { type: Boolean },
});

const Items = mongoose.model('item', itemSchema);
```

Рис. 3.4. Опис схеми Items

```
_id: ObjectId('63bd7f47ca55793312e89975')
index: 36
title: Object
  uk: "Тирамісу"
  en: "Tiramisu"
description: Object
  uk: "Повітряний крем, просочене кавою печиво з легким посмаком коньяку чи р..."
  en: "Airy cream, coffee-soaked cookies with a slight taste of cognac or rum..."
itemParameter: Object
  uk: "🍰"
  transliteratedTitle: "tiramisu"
priceInfo: Array
  0: Object
    count: "120"
    units: "п"
    price: "82"
    parameter: Object
      uk: ""
      id: "983f900c-419c-4b05-9ee1-dbd2f87d0c6"
      restaurantID: "63bbf1089505251e0c530302"
    inStock: true
    isArchived: false
    createdAt: "2023-01-10T15:07:51.554Z"
    lastUpdateDate: "2023-01-10T15:07:51.554Z"
itemImages: Array
  0: "https://www.cookingclassy.com/wp-content/uploads/2022/08/tiramisu-17.j..."
itemGroup: "63bd4c58b4cae5b2403a6d70"
itemSubGroup: "1b2eb38c-697a-4605-8400-cb922f2f4d8f"
offeredItems: Array
```

Рис. 3.5. Запис у схемі Items

									Арк.А
									28
Зм.3	Арк.А	№ докум.№	Підпис	Дата					

Ця схема містить такі поля:

- `_id` (ідентифікатор, тип: `ObjectId`, обов'язкове поле);
- `index` (індекс страви, тип: число);
- `title` (назва страви, тип: об'єкт);
- `description` (опис страви, тип: об'єкт);
- `itemParameter` (параметри страви, тип: об'єкт);
- `transliteratedTitle` (транслітерована назва страви, тип: рядок);
- `priceInfo` (інформація про ціну страви, тип: масив);
- `restaurantID` (ідентифікатор закладу, тип: рядок);
- `inStock` (наявність страви, тип: логічний);
- `isArchived` (чи заархівована страва, тип: логічний);
- `createdDate` (дата створення запису про страву, тип: рядок);
- `lastUpdateDate` (дата останнього оновлення запису про страву, тип: рядок);
- `itemImages` (зображення страви, тип: масив рядків);
- `itemGroup` (група страви, тип: рядок);
- `itemSubGroup` (підгрупа страви, тип: рядок);
- `offeredItems` (інформація про страви, які пропонуються з цією стравою, тип: масив);
- `imageType` (тип зображення страви, тип: рядок);
- `isNewItem` (чи є страва новою, тип: логічний);
- `notForDelivery` (чи страва недоступна на доставку, тип: логічний);
- `likesCount` (кількість лайків страви, тип: число);
- `allergens` (інформація про алергени, тип: масив);
- `allergensManual` (вручну вказана інформація про алергени, тип: масив);
- `isPoster` (чи є це постер, тип: логічний);
- `showPosterTitle` (чи показувати назву постера, тип: логічний).

									Арк.А
									29
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

123.КІ(М)-24.03

3. Групи страв (Tabs). Містить інформацію про групи і підгрупи страв. Складається з 10 полей.

```

const tabsSchema = new mongoose.Schema({
  _id: { type: mongoose.Schema.Types.ObjectId },
  index: { type: Number },
  tabtitle: { type: Object },
  transliteratedTitle: { type: String },
  restaurantID: { type: String },
  subtabs: { type: Object },
  toShow: { type: Boolean },
  createdAt: { type: String },
  notForDelivery: { type: Boolean },
  pos: { type: Object },
});

const Tabs = mongoose.model('tabs', tabsSchema);

```

```

  _id: ObjectId('636a191e735179caca2046b4')
  tabtitle: Object
    uk: "Їжа"
    restaurantID: "6369307fdd42b7d4e743c9bc"
  subtabs: Array
    0: Object
      typeDishes: Object
      itemsID: Array
      image: "https://in-phone.com/images/6369307fdd42b7d4e743c9bc/0677ae10-808c-42c3-9278-6ef37958a908"
      id: "0677ae10-808c-42c3-9278-6ef37958a908"
      toShow: true
      transliteratedTitle: "zakuski-do-piva"
      index: 0
      i: 0
    1: Object
    2: Object
    3: Object
    4: Object
    5: Object
    6: Object
  __v: 0
  toShow: true
  transliteratedTitle: "yizha"
  index: 0

```

а

б

Рис. 3.6. Схеми в БД, а – опис схеми Tabs, б – запис у схемі Tabs

Ця схема містить такі поля:

- `_id` (ідентифікатор, тип: `ObjectId`, обов'язкове поле);
- `index` (індекс групи, тип: число);
- `tabtitle` (назва групи, тип: об'єкт);
- `transliteratedTitle` (транслітерована назва меню, тип: рядок);
- `restaurantID` (ідентифікатор закладу, тип: рядок);
- `subtabs` (підгрупа, тип: масив);
- `toShow` (чи показувати, тип: логічний);
- `createdAt` (дата створення, тип: рядок);
- `notForDelivery` (чи недоступне для доставки, тип: логічний);
- `pos` (інформація про групу, яка імпортована з іншої POS-системи, тип: об'єкт).

У нереляційних базах даних немає таких зв'язків, які є у реляційних БД, отже треба зв'язувати дані за якимись ключами. Всі схеми, які стосуються закладу

харчування, з'єднані за допомогою ключа restaurantID (\_id в схемі Restaurants). Так само з'єднані інші дані. Наприклад, замовлення має ідентифікатор страви (itemID - \_id), а рахунок має масив замовлень (ordersID - \_id). Працівник має поле subroleID, яке зв'язане з полем \_id у схемі Roles.

Інфологічна схема зображена на рис. 3.9.

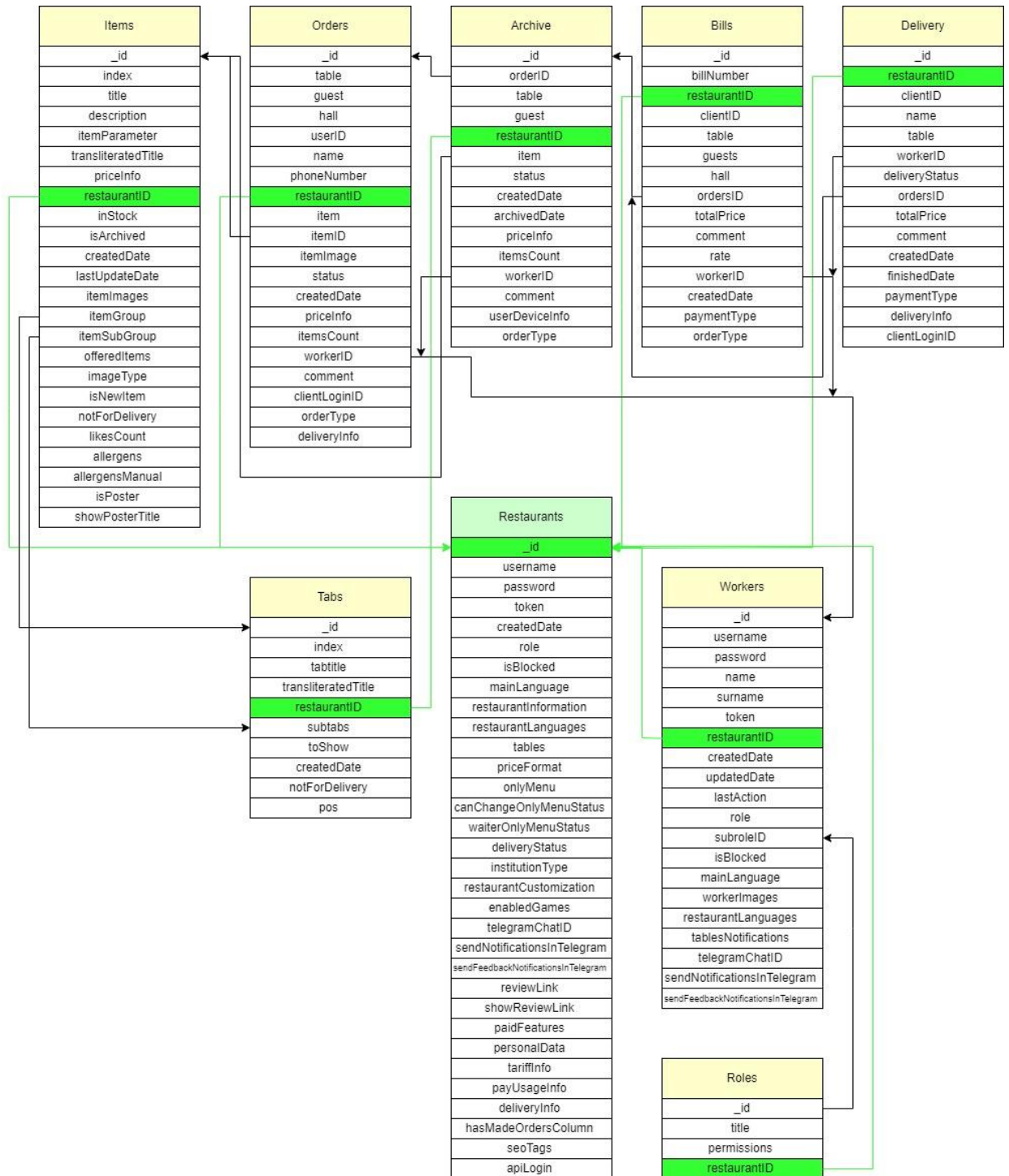


Рис. 3.7. Інфологічна схема

### 3.2.3. Опис структури проекту

Проект містить такі елементи (рис. 3.8):

- `node_modules` – папка, в якій зберігаються бібліотеки, фреймворки, які необхідні для створення проекту. Вони завантажуються за допомогою менеджера пакетів `npm`;
- `public` – папка, в якій розташовані статичні графічні матеріали;
- `components` – папка, в якій розташовані React компоненти, які можна багаторазово використовувати;
- `context` – папка, в якій розташовані описи структури даних, які використовуються під час виконання застосунку;
- `db > controllers` – папка, в якій розташовані методи доступу до бази даних;
- `db > interfaces` – папка, в якій розташовані описи моделей (інтерфейси);
- `db > models` – папка, в якій розташовані описи моделей для генерації БД;
- `db > interfaces` – папка, в якій розташовані описи типів даних;
- `languages` – папка, в якій розташовані файли з перекладами тексту;
- `pages` – папка, в якій розташовані описи шляхів (наприклад, `/user`, `/user/workers`) і дані, які будуть відображатися;
- `shared` – папка, в якій розташовані різні функції;
- `styles` – папка, в якій розташовані файли, які описують глобальні стилі (CSS-файли);
- `package.json` – файл, у якому прописані залежності. При запуску команди `npm install`, відбувається інсталяція бібліотек, які записані у цьому файлі;
- `tsconfig.json` – файл конфігурації TypeScript.

									Арк.А
									32
Зм.З	Арк.А	№ докум.№	Підпис	Дата					



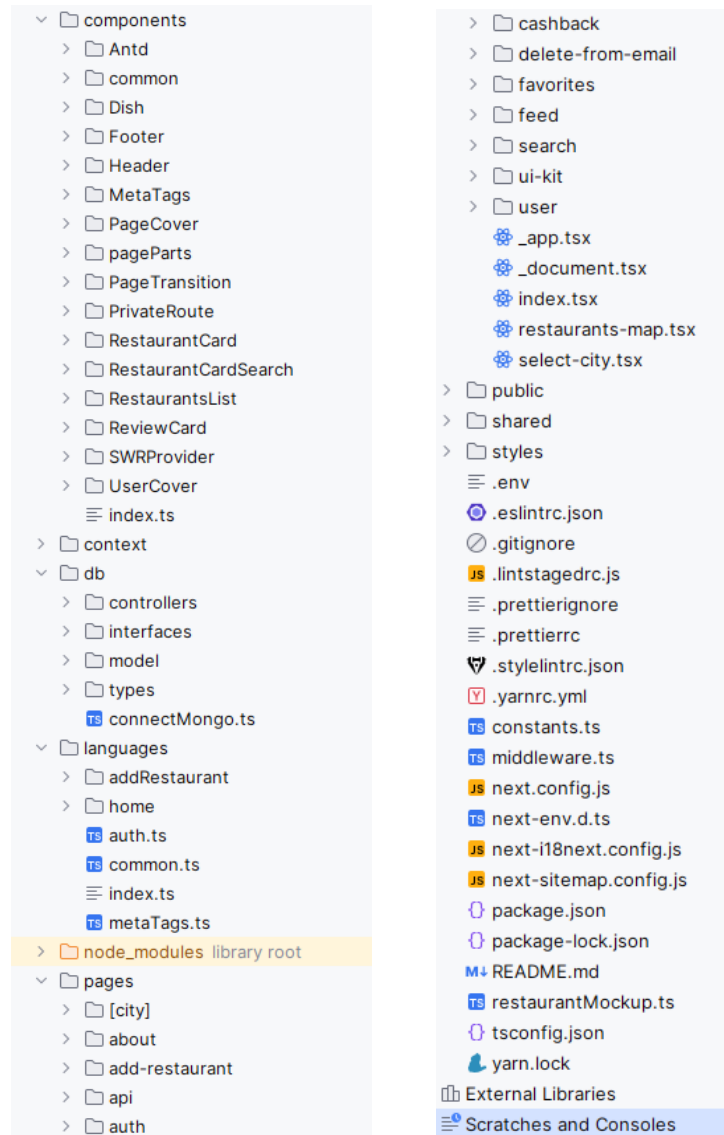


Рис. 3.10. Структура проекту

### 3.3. Висновки до розділу 3

У третьому розділі описано вимоги до системи та її компонентів: авторизація користувачів, робота із набором страв/товарів, робота із інформацією про заклад, оброблення та адміністрування даних.

Побудовано UML-діаграму прецедентів і описано ролі користувачів:

- клієнт – обліковий запис гостя закладу (коли користувач переглядає список закладів і переходить за посиланням або сканує QR-код). Він може переглядати інформацію про заклад харчування, шукати заклади за певними критеріями, переглядати онлайн-меню;

- ресторан – обліковий запис закладу, може виконувати всі дії, які пов’язані із закладом;
- працівник – обліковий запис працівника закладу. Користувач із роллю «ресторан» створює ролі (за замовчуванням є «адміністратор» (має всі дозволи, що і «ресторан») і «офіціант» (має обмежений доступ до ІС (наприклад, не може переглядати і редагувати статистику)) або редагує існуючі, а потім реєструє новий обліковий запис працівника;
- адміністратор – має доступ до інформації про всі заклади, а також може створювати нові заклади і керувати ними.

Також розроблено функціональну, інфологічну схеми системи та описано структуру проекту.

Для зберігання даних обрано базу даних MongoDB – нереляційна БД, яка підходить для збереження великих обсягів даних і обрано ключ, за допомогою якого буде відбуватися доступ до всіх даних – це restaurantID – поле, яке є у всіх схемах БД. Загалом, база даних містить 19 схем: admins, archives, bills, clientlogins, clients, deliveries, errors, feedbackrestaurants, feedbacks, items, logs, notifications, orders, restaurantlogins, restarurants, roles, tabs, workers.

Проектування архітектури сервісу відбувалося таким чином, щоб його можна було легко масштабувати. Це було досягнуто завдяки використанню клієнт-серверної архітектури, у якій сервер відповідає за обробку запитів та формування результатів, а клієнт створює запити та відображає користувачеві отриманий результат. Клієнтська та серверна частини реалізовані таким чином, щоб можна було легко перевикористовувати компоненти (в клієнтській частині це елементи інтерфейсу, які часто використовуються) та різні методи (наприклад, на сервері є методи, які формують запити до БД. Це спрощує з’єднання до БД, оскільки треба просто передати необхідні параметри до методу, а не кожного разу вручну описувати запити). У майбутньому це спростить додавання нового функціоналу до інформаційної системи.

									Арк.А
									34
Зм.З	Арк.А	№ докум.№	Підпис	Дата	123.КІ(М)-24.03				

## РОЗДІЛ 4. ОПИС ІНТЕРФЕЙСУ КОРИСТУВАЧА

Для тестування клієнтської сторони інформаційної системи використано ручне тестування. Для тестування серверної сторони інформаційної системи використано додаток Postman, за допомогою якого можна робити запити до API сервера.

Під час створення інтерфейсу користувача використано мову програмування TypeScript з фреймворком Next.js. Для розроблення дизайну використано каскадні таблиці стилів CSS з препроцесором SCSS.

Опис прикладу роботи інформаційної системи відображено на двох типах ролей: «клієнт» і «ресторан».

### 4.1. Опис роботи інформаційної системи для користувача із роллю «клієнт»

#### 4.1.1. Сторінка вибору міста

Після переходу за посиланням <https://eatwy.com.ua/> користувач потрапить на сторінку вибору міста (рис. 4.1).

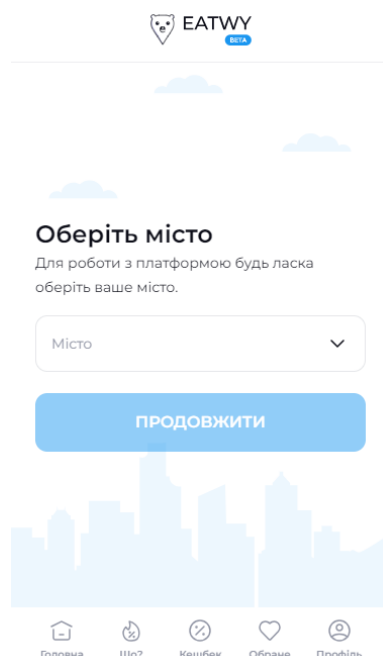


Рис. 4.1. Сторінка вибору міста.

									Арк.А
									35
Зм.3	Арк.А	№ докум.№	Підпис	Дата					

#### 4.1.2. Список закладів у вибраному місті

Після вибору міста, користувач бачить заклади та інформацію про них (рис. 4.2)

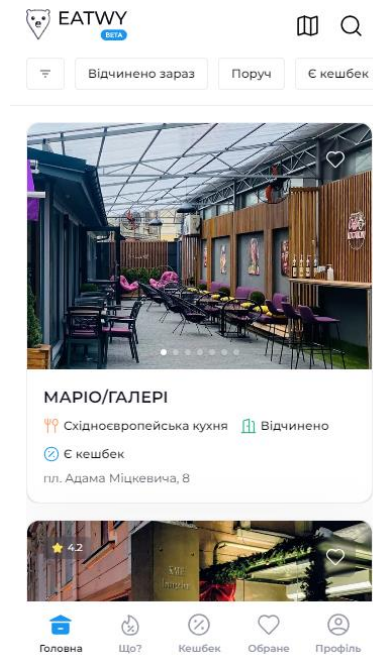


Рис. 4.2. Список закладів у вибраному місті.

Користувач також має можливість відфільтрувати заклади за заданими критеріями (рис. 4.3).

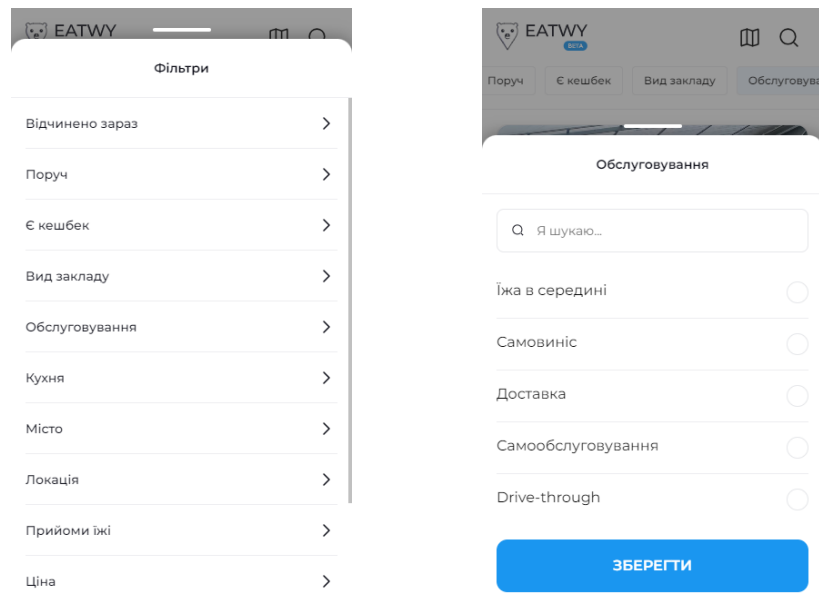


Рис. 4.3. Фільтрація закладів, а – параметри фільтрів, б – фільтри по типу обслуговування

Зм.3	Арк.А	№ докум.№	Підпис	Дата

### 4.1.3. Перегляд інформації про заклад

Після натискання на фотографію чи назву закладу, користувач потрапить на сторінку з детальною інформацією. Тут можна побачити фотографії, назву, опис, тип кухні, відстань до закладу, контактну інформацію, графік роботи, меню, спеціальні пропозиції та відгуки користувачів (рис. 4.4).

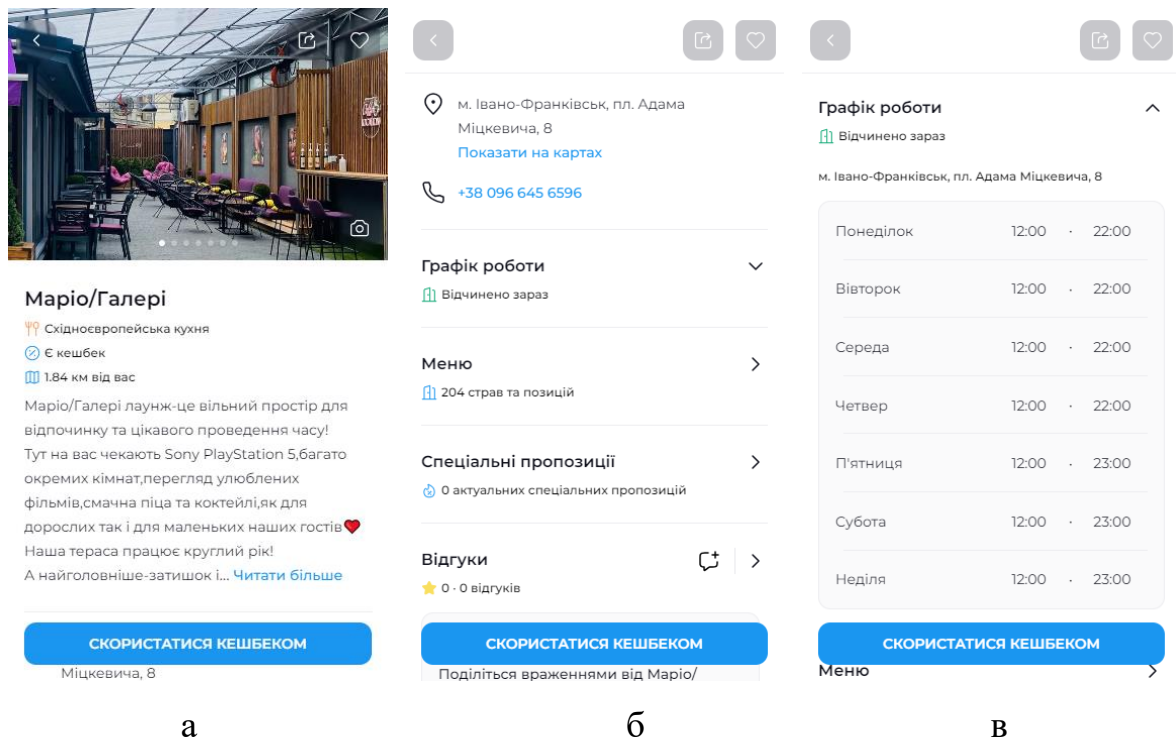


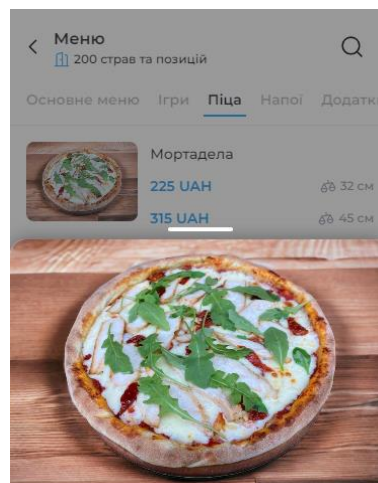
Рис. 4.4. Перегляд детальної інформації про заклад, а – фото та опис закладу, б – контактна інформація, в – графік роботи закладу

### 4.1.4. Перегляд меню закладу

Натиснувши на розділ «меню», відкриється сторінка з меню закладу. Там можна переглядати страви по групах, дивитися детальну інформацію про страви, використовувати пошук (рис. 4.5).



а

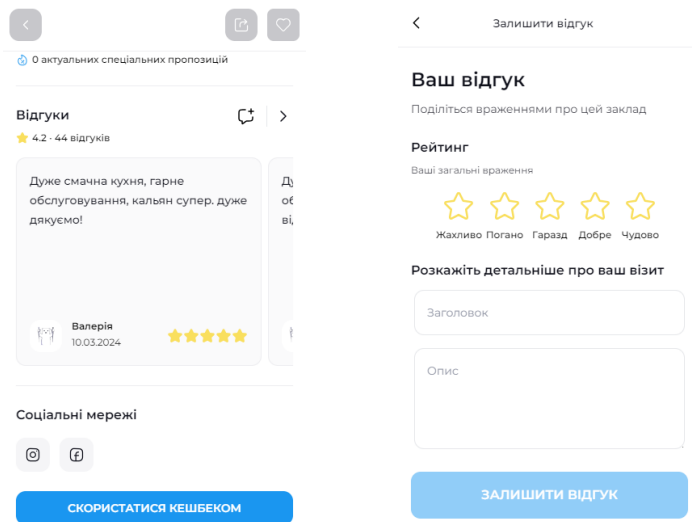


б

Рис. 4.5. Перегляд меню закладу, а – список страв, б – перегляд детальної інформації про страву

#### 4.1.5. Відгуки про заклад

Сервіс має систему відгуків (рис. 4.6). Вони допомагають закладам харчування швидко виявляти проблеми та покращувати якість обслуговування, орієнтуючись на побажання клієнтів. Вони також сприяють залученню нових відвідувачів, оскільки позитивні відгуки підвищують довіру до закладу. Після того, як гість залишить відгук, адміністратори закладу мають можливість їх опрацьовувати в панелі керування.



а

б

Рис. 4.6. Відгуки, а – перегляд відгуків закладу, б – створення нового відгуку

#### 4.1.6. Сервіс кешбеку

Гість закладу може пред'явити офіціанту свій QR-код (рис. 4.7), який той сканує і фіксує суму чеку. Після цього гостю нараховується певний відсоток, а система отримує комісію. Аналогічно, гість може розрахуватися за допомогою бонусів.

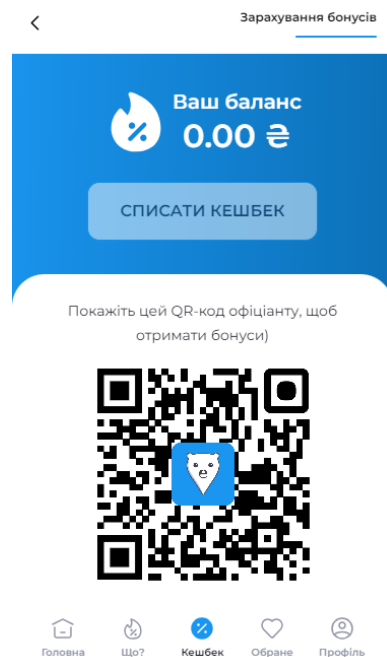


Рис. 4.7. Система кешбеку

						123.КІ(М)-24.03	Арк.А
							39
Зм.3	Арк.А	№ докум.№	Підпис	Дата			

#### 4.1.7. Сторінка зі спеціальними пропозиціями закладів

Користувачам не потрібно стежити за всіма ресторанами в Instagram, щоб бути в курсі спеціальних пропозицій, знижок та заходів. Усі актуальні оновлення зібрані на одній сторінці, щоб вони завжди були під рукою тоді, коли це потрібно, а не випадково під час перегляду стрічки (рис. 4.8).

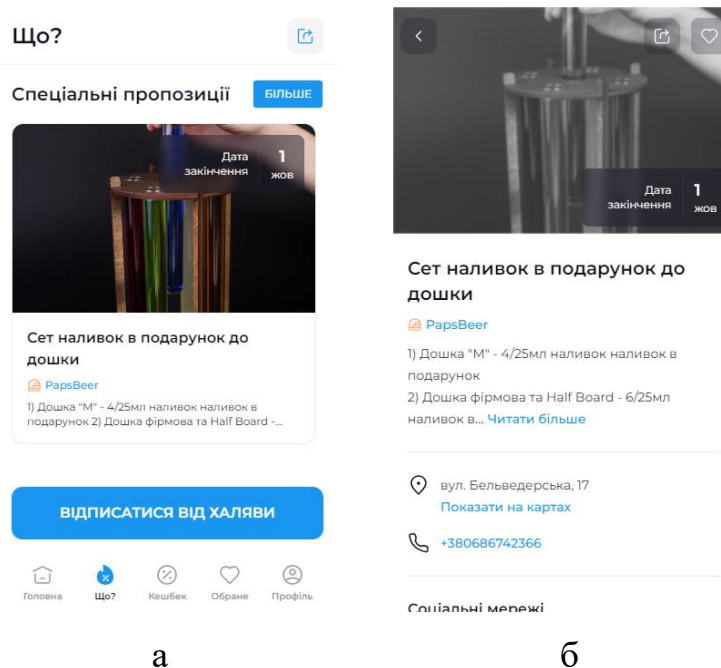
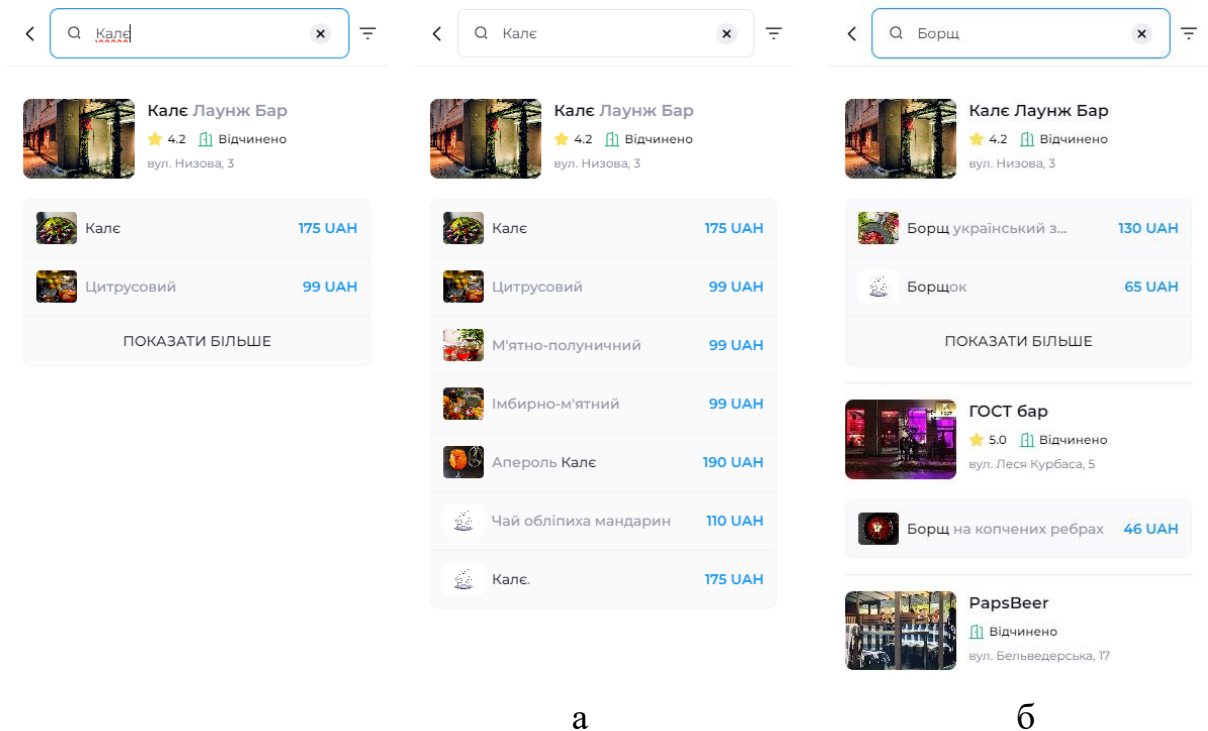


Рис. 4.8. Сторінка зі спеціальними пропозиціями закладів, а – список спеціальних пропозицій зі всіх закладів, б – сторінка з детальною інформацією

#### 4.1.8. Пошук закладів та страв

Користувач може почати вводити назву закладу або страви, і система відобразить заклади, де ця страву доступна (рис. 4.9). Також є розширений пошук, де користувач може вказати кілька страв, і система знайде заклади, які пропонують цю комбінацію.

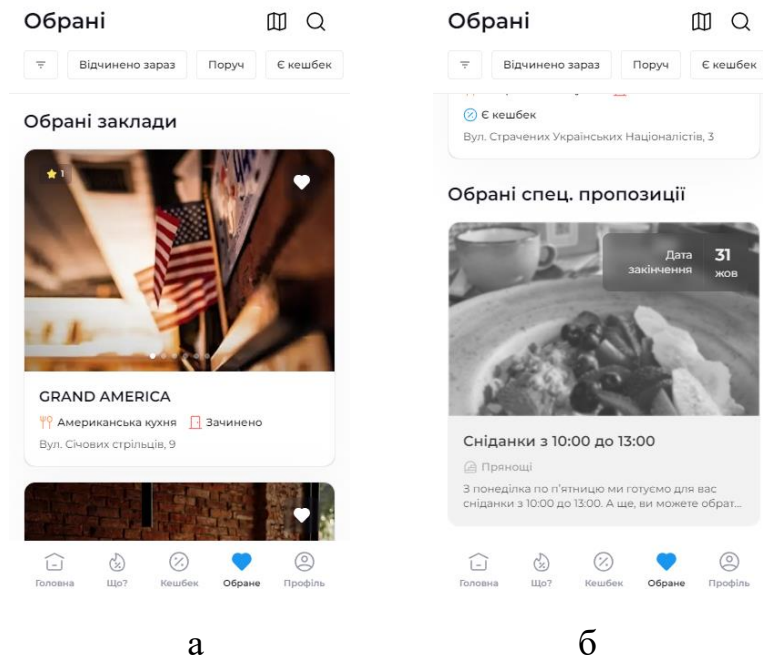




а б  
Рис. 4.9. Сторінка пошуку, а – пошук закладів, б – пошук страв

#### 4.1.9. Пошук збережених закладів та страв

Список обраних закладів та спеціальних пропозицій, збережених користувачем (рис. 4.10).



а б  
Рис. 4.10. Сторінка обраного, а – обрані заклади, б – обрані спец. пропозиції

#### 4.1.10. Карта закладів

На карті відображаються всі заклади: звичайні позначені синім кольором, обрані – червоним, а ті, де є активні спеціальні пропозиції – зеленим. Користувач може побачити назву закладу, перевірити, чи він зараз відчинений, дізнатися відстань до нього та прокласти маршрут (рис. 4.11).

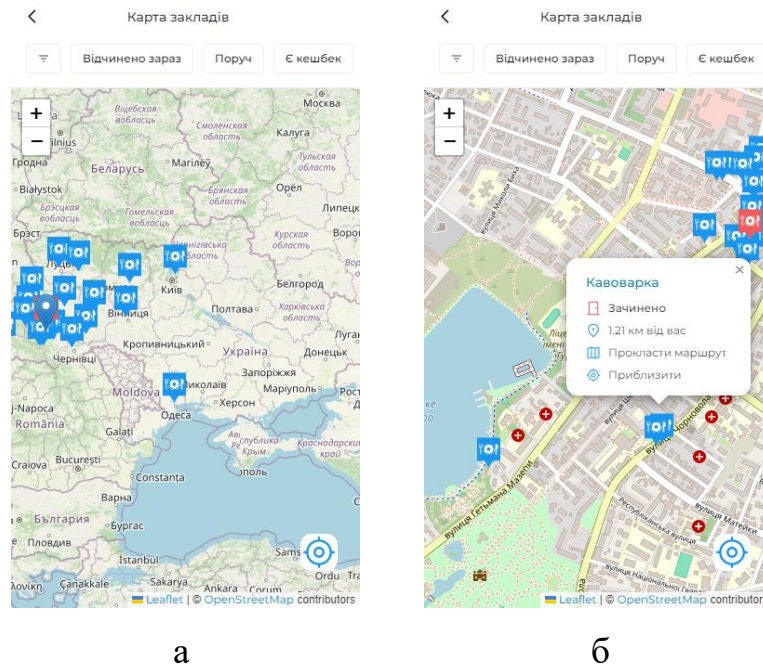


Рис. 4.11. Карта закладів, а – перегляд закладів на карті, б – перегляд короткої інформації про заклад

#### 4.1.11. Автентифікація

Для реєстрації користувач вводить свою електронну пошту та пароль, які згодом використовує для входу в систему (рис. 4.12). Це забезпечує зручний доступ до облікового запису та персоналізованих функцій платформи.

Після реєстрації користувач має підтвердити обліковий запис за допомогою листа, надісланого на електронну пошту. Це необхідний крок, який гарантує, що адреса електронної пошти належить користувачу та активує всі функції платформи. Якщо користувач не підтвердить свою електронну пошту, він не зможе отримати доступ до всіх функцій платформи, включаючи персоналізовані

сервіси та спеціальні пропозиції. Підтвердження пошти є обов'язковим для повноцінного використання системи.

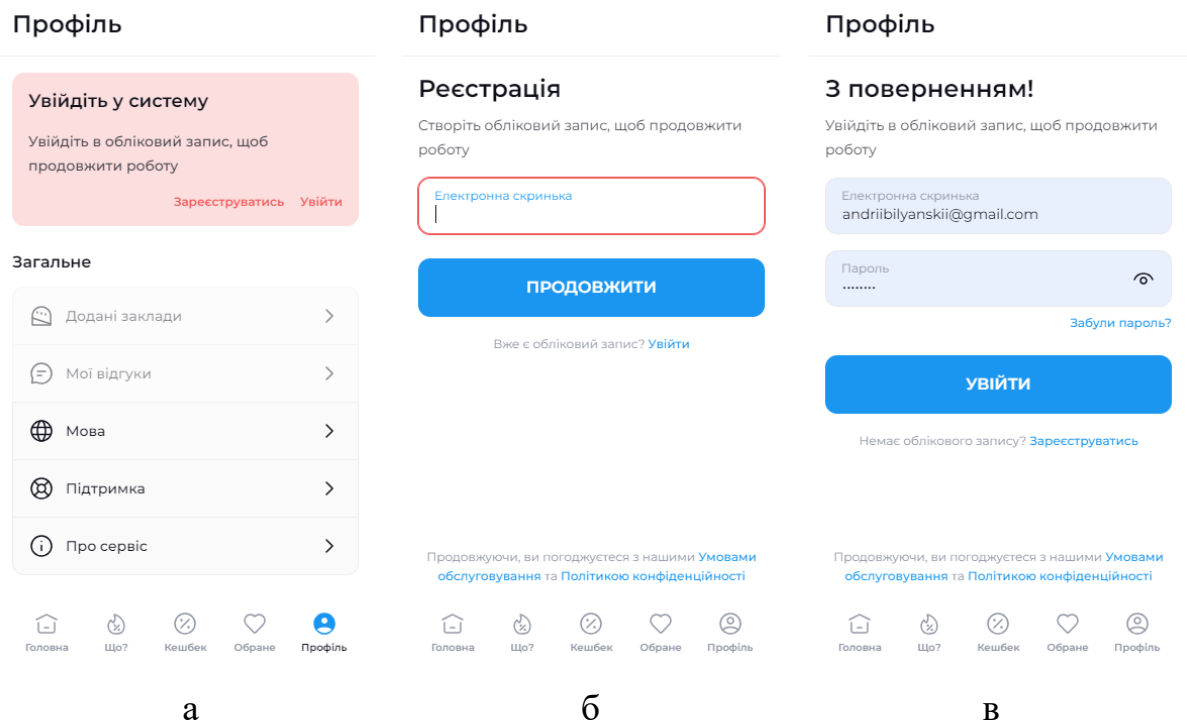


Рис. 4.12. Авторизація, а – сторінка профілю, б – реєстрація, в – вхід в обліковий запис

#### 4.1.12. Керування обліковим записом

Сторінка редагування облікового запису дозволяє користувачеві змінювати особисту інформацію (рис. 4.13), міняти пароль (рис. 4.14). Користувач може переглядати список закладів, які він додав у систему, а також переглядати написані відгуки (рис. 4.15).

## Профіль



### Загальне

- Додані заклади
- Мої відгуки
- Мова
- Підтримка
- Про сервіс



а

## Редагування профілю



### Загальне

- Персональні дані
- Зміна паролю

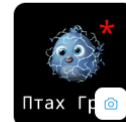
### Інше

- Вихід
- Видалити профіль



б

## Редагування профілю



### Інформація про вас

#### Інформація про вас

Ім'я та прізвище  
Андрій

+380 (66) 858 61 84

Електронна скринька  
andriibilyanski@gmail.com

ЗБЕРЕГТИ ЗМІНИ



в

Рис. 4.13. Профіль, а – сторінка профілю, б – налаштування профілю, в – редагування інформації профілю

## Редагування профілю

### Зміна паролю

#### Зміна паролю

Поточний пароль

Новий пароль

Повторіть новий пароль

ЗБЕРЕГТИ ЗМІНИ



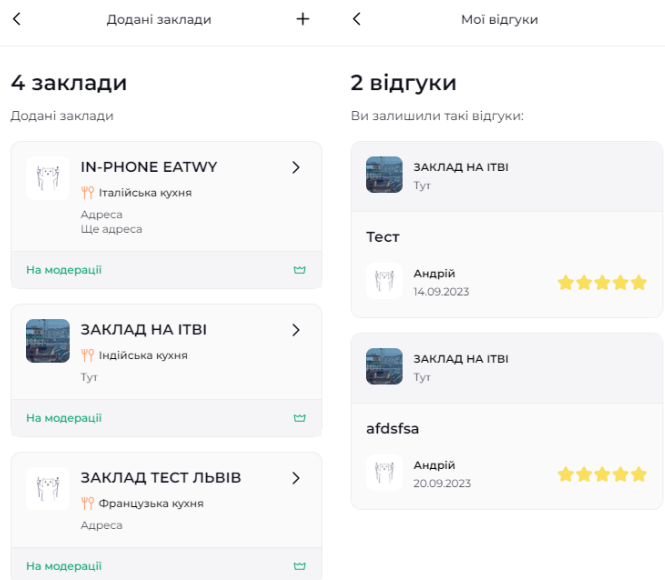
Рис. 4.14. Сторінка зміни паролю

Зм.З	Арк.А	№ докум.№	Підпис	Дата

123.КІ(М)-24.03

Арк.А

44



а

б

Рис. 4.15. Профіль користувача, а – сторінка із закладами, які додані користувачем, б – написані користувачем відгуки

#### 4.1.13. Додавання нового закладу

Користувач має можливість додати свій заклад, визначаючи, чи він є відвідувачем, чи власником закладу. Потім він вводить загальну інформацію про заклад, включаючи назву, адресу, контактні дані та опис (рис. 4.16).

Рис. 4.16. Додавання нового закладу

Далі додає фото закладу, графік роботи, та меню (рис. 4.17).

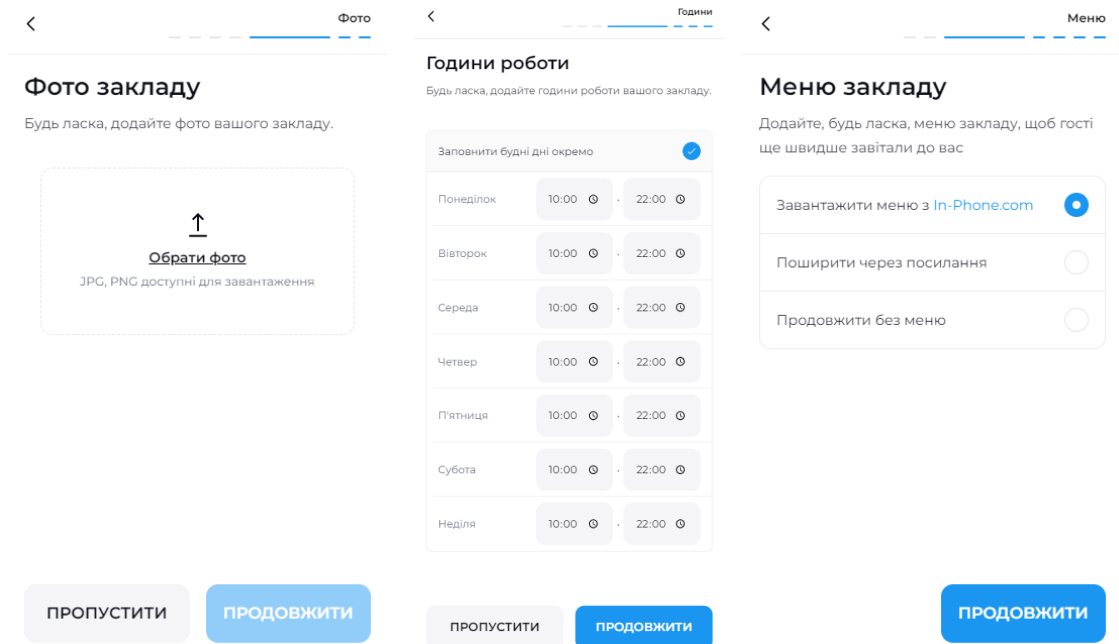


Рис. 4.17. Додавання нового закладу

Далі треба обрати типологію (тип закладу, тип кухні, місцезнаходження) та ввести персональні дані для того, щоб система могла зв'язатися та уточнити інформацію (рис. 4.18-4.19).

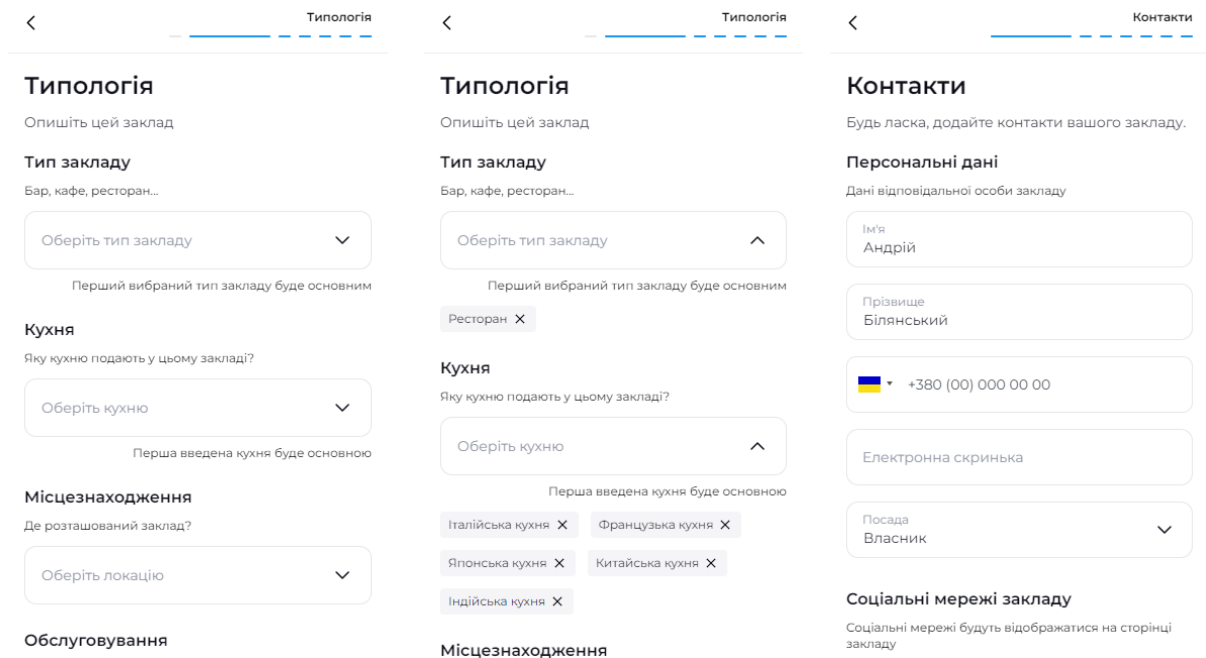


Рис. 4.18. Додавання нового закладу

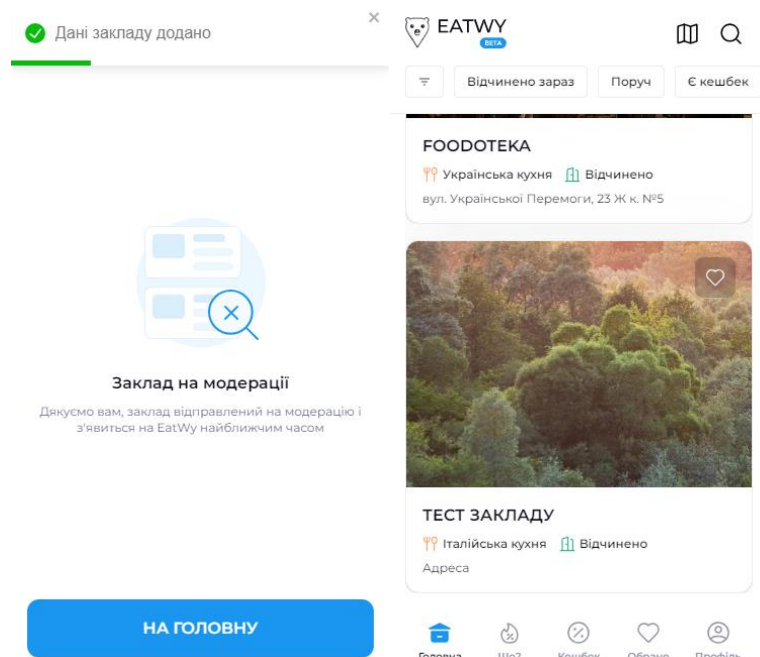


Рис. 4.19. Додавання нового закладу

## 4.2. Опис роботи інформаційної системи для користувача із роллю «ресторан»

### 4.2.1. Автентифікація

Для входу в обліковий запис користувача треба перейти по посиланню <https://in-phone.com/login>. Також можна перейти по посиланню <https://in-phone.com/> і натиснути кнопку «Увійти», яка розміщена у верхньому правому куті (рис. 4.20).

Зм.3	Арк.А	№ докум.№	Підпис	Дата

123.КІ(М)-24.03

Арк.А

47

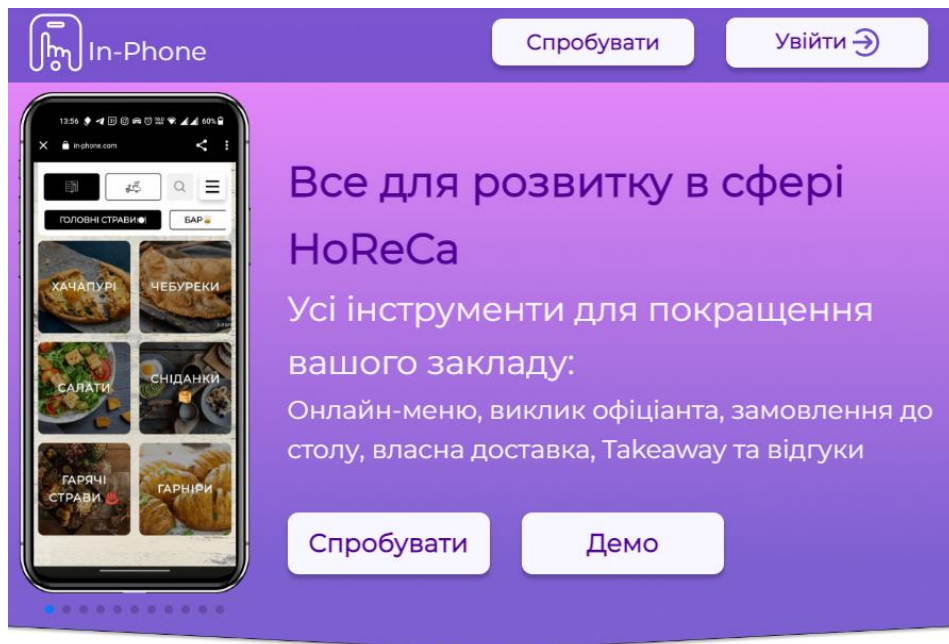


Рис. 4.20. Головна сторінка сервісу In-Phone

Після цього користувач потрапить на сторінку автентифікації (рис 4.21).

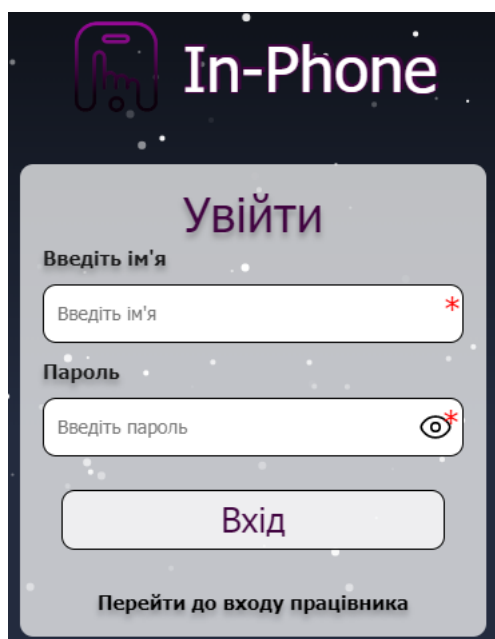


Рис. 4.21. Сторінка автентифікації

Зм.3	Арк.А	№ докум.№	Підпис	Дата



## 4.2.2. Сторінка про заклад

Після успішного входу користувач потрапить на головну сторінку з інформацією про заклад. На цій сторінці можна міняти логотип закладу, назву, опис, адресу, номери телефонів, посилання на соціальні мережі (рис. 4.22-4.25).

**Ваш Заклад**

**Загальна інформація**  
Додайте більш детальну інформацію про ваш заклад

UK: Назва закладу  
Ваш Заклад

EN: Restaurant name  
Vash Zaklad

UK: Загальна інформація  
Французький ресторан у місті Ментона, який має три зірки Michelin.  
Mirazur працює у будівлі 1930-х поруч із морем, горою та садом.

EN: General information  
A French restaurant in the city of Menton that has three Michelin stars.  
Mirazur operates in a 1930s building next to the sea, a mountain and a garden.

UK: Особливість закладу

EN: Restaurant's feature

**Локації**

Місто  
Івано-Франківськ (Івано-Франківський район, Івано-Франківська область)

UK: Адреса  
Івано-Франківськ

EN: Restaurant address  
The city of Menton

Посилання з карт

Координати закладу

НАЛАШТУВАННЯ ГОДИН РОБОТИ

ВИДАЛИТИ

ДОДАТИ

**Номери телефонів**

+380 (67) 646 43 41

UK: Опис телефону

EN: Phone description

ВИДАЛИТИ

ДОДАТИ

Логін: VashZaklad  
ID закладу: 639d97151a8a373c693047c6

ВІДМІНИТИ

ЗБЕРЕГТИ

Рис. 4.22. Сторінка інформації про заклад



Головна

Замовлення

Доставка

Страви

Акції

Кешбек

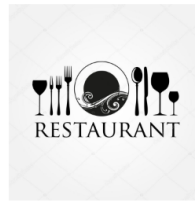
Аналітика

Налаштування

### Ваш Заклад

#### Логотип закладу

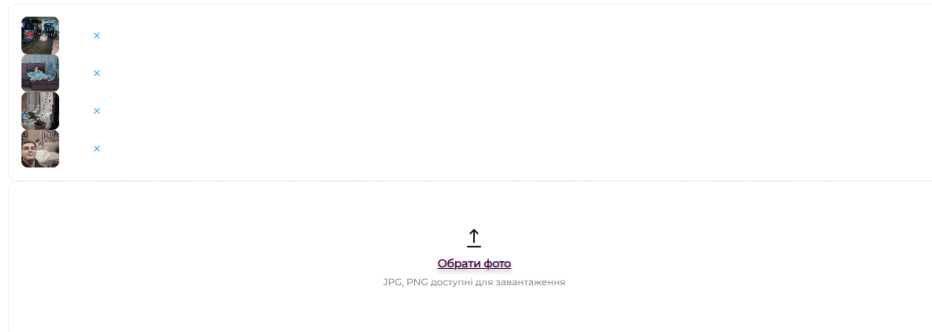
Будь ласка додайте логотип вашого закладу.



[https://in-phone.com/images/639d97151a8a373c693047c6/user/depositphotos\\_77833254-](https://in-phone.com/images/639d97151a8a373c693047c6/user/depositphotos_77833254-)

#### Фото закладу

Будь ласка додайте фото вашого закладу.



Логін: [VashZaklad](#)

ID закладу: 639d97151a8a373c693047c6

ВІДМІНИТИ

ЗБЕРЕГТИ

Рис. 4.23. Сторінка інформації про заклад



Головна

Замовлення

Доставка

Страви

Акції

Кешбек

Аналітика

Налаштування

### Ваш Заклад

#### Меню закладу

Будь ласка, завантажте меню вашого закладу.

Завантажити меню з In-Phone.com

Поширити через посилання

Логін: [VashZaklad](#)

ID закладу: 639d97151a8a373c693047c6

ВІДМІНИТИ

ЗБЕРЕГТИ


Рис. 4.24. Сторінка інформації про заклад








Зм.3	Арк.А	№ докум.№	Підпис	Дата

123.КІ(М)-24.03

Арк.А

50



-  Головна
-  Замовлення
-  Доставка
-  Страви
-  Акції
-  Кешбек
-  Аналітика

**Ваш Заклад**

**Типологія**  
Опишіть ваш заклад

**Вид закладу**  
Вид закладу

Оберіть вид закладу

Перший вибраний тип закладу буде основним

Ресторан X
Кафе X
Бар&Паб X
Кав'ярня (посадка) X
Кав'ярня (на винос) X

**Кухня**  
Кухня

Оберіть кухню

Перша введена кухня буде основною

Італійська кухня X
Французька кухня X
Японська кухня X

**Місцезнаходження**  
Місцезнаходження

Оберіть локацію

**Обслуговування**  
Обслуговування


Оберіть тип обслуговування

Інка в середині X
Доставка X

**Паяння**  
Паяння

Оберіть пору паяння

Сніданки X
Обіди X
Вечеря X

Логін: VashZaklad 

ID закладу: 639d97151a8a373c693047c6

ВІДМІНИТИ

ЗБЕРЕГТИ

ЗАКЛАД

ФОТО

МЕНЮ

**ТИПОЛОГІЯ**

КОНТАКТИ


 Налаштування

Рис. 4.25. Сторінка інформації про заклад

### 4.2.3. Сторінка зі стравами

Заклад може додавати страви до свого меню, які потім відобразатимуться в інформаційній системі (рис. 4.26, 4.27).

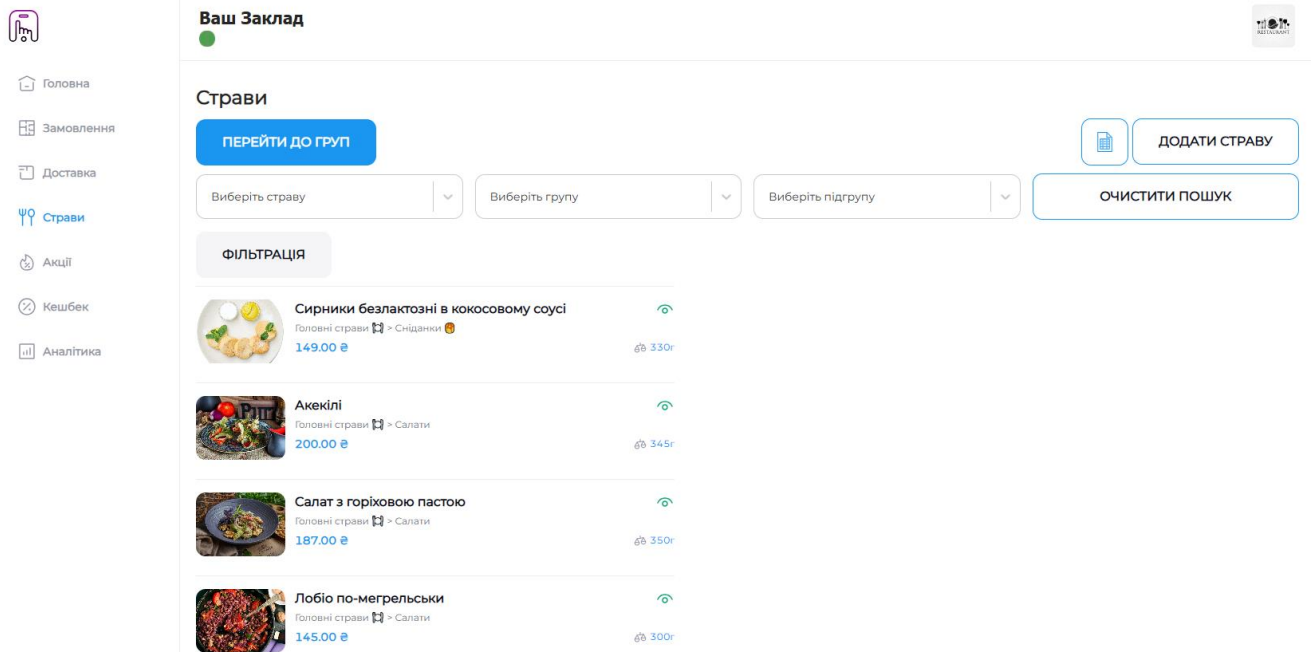


Рис. 4.26. Сторінка зі стравами

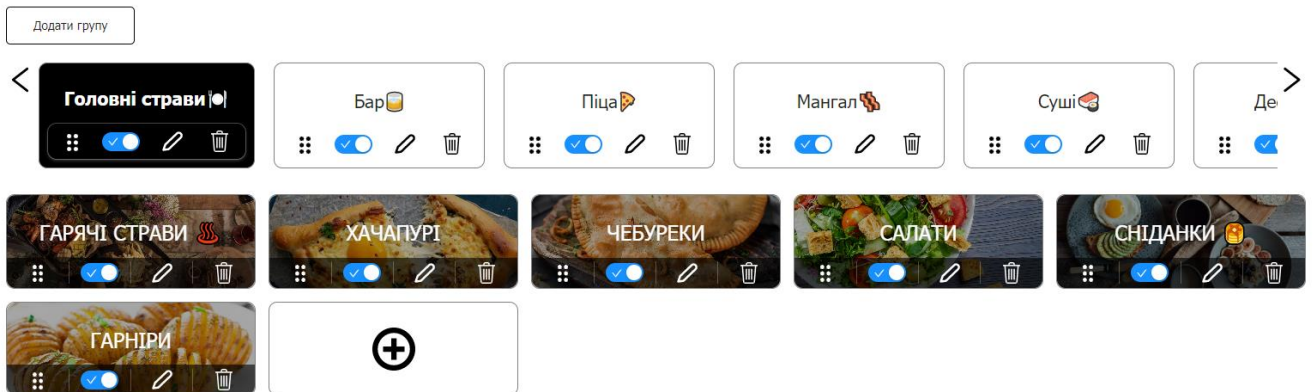


Рис. 4.27. Перегляд груп і підгруп

На сторінці з групами можна створити групи, підгрупи, переглянути страви по групах (рис. 4.28).

## Страви

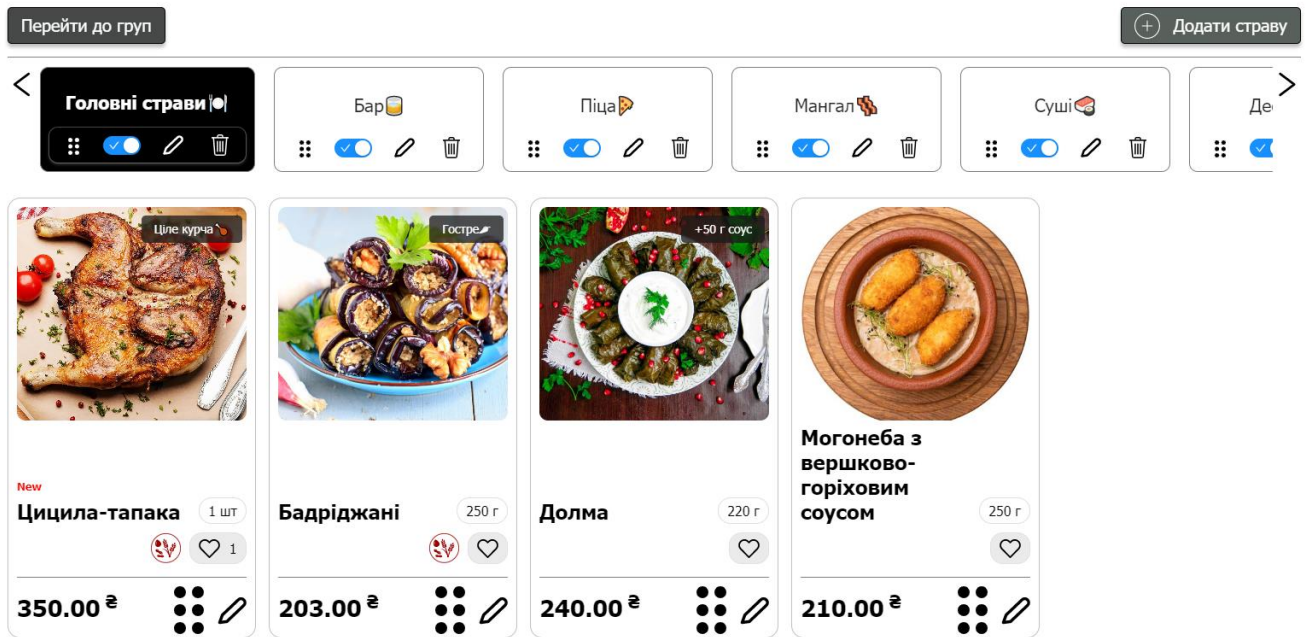


Рис. 4.28. Перегляд страв по підгрупах

Якщо натиснути кнопку «Додати страву», відкриється сторінка з редагуванням страви (рис. 4.29).

Рис. 4.29. Додавання страви

Тут можна ввести назву страви, опис, параметр, вибрати групу і підгрупу, позначити чи страва є новинкою і чи недоступна на доставку. Також можна завантажити фотографію або ввести посилання на неї.

Зм.3	Арк.А	№ докум.№	Підпис	Дата

123.КІ(М)-24.03

Арк.А  
53

Далі треба ввести ціну (можна додавати кілька цін). Також можна вибрати алергени, які містяться у цій страві та обрати рекомендації (рис. 4.30).

The screenshot shows a form titled "Ціни" (Prices) with several input fields and a list of allergens. The "Ціни" section includes:
 

- Input for quantity: "Введіть кількість" (350)
- Unit dropdown: "г"
- Input for price: "Введіть ціну" (187)
- Input for packaging price: "Введіть ціну упаковки"
- Language/parameter input: "UK: Введіть параметр" and "EN: Enter parameter"
- Input for old price: "Введіть стару ціну"
- Weight unit toggle: "На вагу" with a radio button.

 Below this is the "Алергени" (Allergens) section, which lists various allergens with corresponding icons and toggle switches:
 

- Злакові вироби, що містять глютен
- Молюски та ракоподібні
- Яйця
- Риба
- Арахіс
- Соя
- Молочні продукти
- Горіхи
- Гострий перець
- Мед
- Цитрусові
- Шоколад
- Часник

 At the bottom of the allergen list is a button: "Додати новий алерген (вести вручну)".

Рис. 4.30. Додавання страви

Заклад може додавати спеціальні пропозиції (рис. 4.31).

The screenshot shows a page titled "Ваш Заклад" (Your Establishment) with a sub-section "Спеціальні пропозиції" (Special Offers). The page includes:
 

- A navigation menu on the left with items like "Головна", "Замовлення", "Доставка", "Страви", "Акції", "Кешбек", and "Аналітика".
- A "ДОДАТИ АКЦІЮ" (ADD OFFER) button in the top right.
- Two featured offers:
  - "Тест" (Test) by "фірфі" with a date of "31 жовт." and a link to "https://in-phone.com/feed".
  - "«ФОЛІАНТ» ДРУКАРНЯ" (FOLIANT PRINTING) by "asd" with a date of "31 жовт." and a link to "https://in-phone.com/feed".

Рис. 4.31. Сторінка зі спеціальними пропозиціями

### 4.3. Висновки до розділу 4

У четвертому розділі описано інструкцію для користування інформаційною системою на прикладі двох типах ролей: «клієнт» і «ресторан».

Для «клієнта» описано процес перегляду вибору міста, перегляду та пошуку закладів харчування, перегляду онлайн-меню, створення відгуків, використання сервісу кешбеку, авторизація та керування даними облікового запису.

Для «ресторану» описано процес автентифікації, редагування інформації про заклад, керування стравами, групами та підгрупами страв і налаштування закладу.

									Арк.А
									55
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

## ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи розроблено інформаційну систему для пошуку та перегляду закладів харчування на основі фреймворку Next.js. Досягнуто початкової мети – розроблено інформаційну систему, яка дає користувачеві можливість зареєструватися та авторизуватися, переглядати заклади харчування в обраному місті, шукати заклади харчування за певними критеріями, переглядати онлайн-меню закладу, залишати відгуки.

У роботі проведено аналіз сучасних інформаційних систем за даною тематикою, аргументовано актуальність теми дослідження.

Під час виконання завдання здійснено функціональне та інфологічне проектування, описано логіку функціонування програмних частин системи.

У порівнянні з існуючими системами, EatWу надає зручність користування, об'єднує весь функціонал інших систем в одному сервісі.

Станом на листопад 2024 року, на платформі зареєстровано 91 користувач, та щодня на інформаційну систему входять 100-500 користувачів.

Отже, в результаті магістерської кваліфікаційної роботи є інформаційна система для пошуку та перегляду закладів харчування EatWу, яка надає функціонал онлайн-меню та POS-системи. Аналізуючи статистику використання, а саме – входи на сервіс (станом на середину листопада 2024 року – це більше 8 унікальних користувачів, які зайшли на платформу, середня кількість переглядів за добу 100-500 переглядів) можна зробити висновок, що система є корисною і потрібною.

									Арк.А
									56
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

123.КІ(М)-24.03



## СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. In-Phone. Все для розвитку в сфері HoReCa. [Електронний ресурс] – Режим доступу до ресурсу <https://in-phone.com/> (дата звернення: 29.10.2024)
2. Eatwy. Eat where you are. [Електронний ресурс] – Режим доступу до ресурсу <https://eatwy.com.ua/> (дата звернення: 29.10.2024)
3. Next.js by Vercel - The React Framework for the Web. [Електронний ресурс] – Режим доступу до ресурсу <https://nextjs.org/> (дата звернення: 29.10.2024)
4. Доставка з McDonald's у м. Івано-Франківськ | Замовляйте онлайн з Glovo. [Електронний ресурс] – Режим доступу до ресурсу <https://glovoapp.com/ua/uk/ivano-frankivsk/mcdonald-s-ivf> (дата звернення: 29.10.2024)
5. МакДональдз – Карти Google. [Електронний ресурс] – Режим доступу до ресурсу <https://maps.app.goo.gl/KcKNwyNLhEh9m2h48> (дата звернення: 29.10.2024)
6. McDonald's Ukraine (@mcdonaldsukraine) • Світлини та відео в Instagram. [Електронний ресурс] – Режим доступу до ресурсу <https://www.instagram.com/mcdonaldsukraine/> (дата звернення: 29.10.2024)
7. КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА. [Електронний ресурс] – Режим доступу до ресурсу <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/> (дата звернення: 29.10.2024)
8. Архітектура клієнт-сервер. [Електронний ресурс] – Режим доступу до ресурсу <http://inter.ptngu.com/kompyuterni-merezhi/arhitektura-kliiyent-server> (дата звернення: 29.10.2024)
9. TypeScript. [Електронний ресурс] – Режим доступу до ресурсу <https://uk.wikipedia.org/wiki/TypeScript> (дата звернення: 29.10.2024)
10. JavaScript. [Електронний ресурс] – Режим доступу до ресурсу <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення: 29.10.2024)
11. React. [Електронний ресурс] – Режим доступу до ресурсу <https://uk.wikipedia.org/wiki/React> (дата звернення: 29.10.2024)

									Арк.А
									57
Зм.З	Арк.А	№ докум.№	Підпис	Дата				123.КІ(М)-24.03	

12. React – A JavaScript library for building user interfaces. [Електронний ресурс] – Режим доступу до ресурсу <https://uk.reactjs.org/> (дата звернення: 29.10.2024)
13. Redux. [Електронний ресурс] – Режим доступу до ресурсу <https://uk.wikipedia.org/wiki/Redux> (дата звернення: 29.10.2024)
14. SASS. [Електронний ресурс] – Режим доступу до ресурсу <https://uk.wikipedia.org/wiki/Sass> (дата звернення: 29.10.2024)
15. Chart.js. [Електронний ресурс] – Режим доступу до ресурсу [https://www.w3schools.com/ai/ai\\_chartjs.asp](https://www.w3schools.com/ai/ai_chartjs.asp) (дата звернення: 29.10.2024)
16. Node.js. [Електронний ресурс] – Режим доступу до ресурсу <https://uk.wikipedia.org/wiki/Node.js> (дата звернення: 29.10.2024)
17. Express.js. [Електронний ресурс] – Режим доступу до ресурсу <https://uk.wikipedia.org/wiki/Express.js> (дата звернення: 29.10.2024)
18. MongoDB. [Електронний ресурс] – Режим доступу до ресурсу <https://uk.wikipedia.org/wiki/MongoDB> (дата звернення: 29.10.2024)
19. Visual Studio Code. [Електронний ресурс] – Режим доступу до ресурсу [https://uk.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://uk.wikipedia.org/wiki/Visual_Studio_Code) (дата звернення: 29.10.2024)
20. Postman. [Електронний ресурс] – Режим доступу до ресурсу <https://www.postman.com/> (дата звернення: 29.10.2024)
21. Compass. The GUI for MongoDB. [Електронний ресурс] – Режим доступу до ресурсу <https://www.mongodb.com/products/compass> (дата звернення: 29.10.2024)
22. MongoDB vs. MySQL Differences. [Електронний ресурс] – Режим доступу до ресурсу <https://www.mongodb.com/compare/mongodb-mysql> (дата звернення: 29.10.2024)

						Арк. А
						58
Зм.З	Арк.А	№ докум.№	Підпис	Дата		

## ДОДАТОК А

### Лістинг коду компонента авторизації

```
import { NextPage } from 'next';
import { useEffect, useState } from 'react';
import { signIn } from 'next-auth/react';
import { useRouter } from 'next/router';
import { AuthLayout, Input, Link, Text } from '@components';
import { errorMessage, getI18N, getUserDeviceInfo, notificationToastFunc,
  successMessage, useUserContext } from '@shared';
import { useAuthSession, useLanguage } from '@shared/hooks';

export async function getServerSideProps({ locale }) {
  return {
    props: {...(await getI18N(locale))},
  };
}

const Login: NextPage = () => {
  const { language } = useLanguage();
  const { user } = useUserContext();

  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const router = useRouter();
  const { isAuthenticated, deleteSession } = useAuthSession();

  useEffect(() => {
    if (isAuthenticated) {
      deleteSession();
    }
  }, [deleteSession, isAuthenticated]);

  return (
    <AuthLayout
      metaTags={{
        title: language('metaTags:auth.login'),
        description: language('metaTags:description.login'),
      }}
      header={'info'}
      signInText={{
        title: language('auth:welcome'),
        p2: language('auth:singInLoginText'),
      }}
      form={{
        onSubmit: async (e) => {
          const result = await signIn('credentials', {
            redirect: false,
            email,
            password,
            userRole: 'clientEatwy',
            udi: getUserDeviceInfo(false, true),
          });
          if (result?.ok) {
            return { ...successMessage, result };
          } else {

```

									Арк.А
									59
Зм.3	Арк.А	№ докум.№	Підпис	Дата				123.KI(M)-24.03	

```

        notificationToastFunc({
            language: 'uk',
            type: 'error',
            text: 'auth error',
        });

        return errorMessage({ error: 'auth error', status: 500
});
    }
},
onOk: (e) => {
    router.replace('/user');
},
inputs: (
    <>
        <Input
            placeholder={language('auth:email')}
            value={email}
            type={'email'}
            onChange={setEmail}
            required
            name={'email'}
        />

        <Input
            placeholder={language('auth:password')}
            value={password}
            onChange={setPassword}
            type='password'
            pattern={'password'}
            required
        />
    </>
),
underInputs: (
    <Text.S1 type={'medium'}>
        <Link to={'/auth/forgot-password'}>
            {language('auth:forgotPassword')}
        </Link>
    </Text.S1>
),
button: { text: language('auth:loginButton') },
}}
underForm={{
    text: language('auth:dontHaveAccount'),
    link: {
        to: '/auth/register',
        text: language('auth:register'),
    },
}},
/>
);
};
export default Login;

```

						123.KI(M)-24.03	Арк.А
							60
Зм.3	Арк.А	№ докум.№	Підпис	Дата			

## ДОДАТОК Б

### Лістинг коду компонента авторизації

```
import NextAuth from 'next-auth';
import CredentialsProvider from 'next-auth/providers/credentials';

import connectMongo from '@db/connectMongo';
import { UserController } from '@db/controllers';
import { IClientEatwy, IRestaurant } from '@db/interfaces';

import { comparePassword, cookies } from '@shared';

async function authorize(credentials: any) {
  connectMongo();

  let user: IRestaurant | IClientEatwy;

  switch (credentials?.userRole) {
    case 'restaurant':
      user = await
UserController.restaurants.get.byUserName_login(credentials.username);
      break;
    case 'clientEatwy':
      user = await
UserController.clientsEatwy.get.byEmail_login(credentials.email);
      break;
    case 'clientEatwyRegistered':
      user = await
UserController.clientsEatwy.get.byEmail_login(credentials.email);
      break;
    default:
      user = {} as IClientEatwy;
  }

  if (
    credentials?.userRole === 'clientEatwyRegistered' ||
    (user && comparePassword(credentials.password, user.password))
  ) {
    if (credentials?.udi && user?._id) {
      await UserController.clientsEatwy.update.saveLogin(user?._id,
credentials.udi);
    }

    return user;
  }

  return null;
}

export default NextAuth({
  session: {
    strategy: 'jwt',
  },
  callbacks: {
    async jwt({ token, user, trigger, session }) {
      if (trigger === 'update') {
        return { ...token, ...session.user };
      }
    }
  }
});
```

```
    }  
    return { ...token, ...user };  
  },  
  async session({ session, token }) {  
    return { ...session, user: { ...token, ...session?.user } };  
  },  
},  
providers: [  
  CredentialsProvider({  
    authorize,  
  } as any),  
],  
secret: process.env.TOKEN_KEY,  
});
```

									Арк.А
									62
Зм.З	Арк.А	№ докум.№	Підпис	Дата				123.КІ(М)-24.03	

## ДОДАТОК В

### Лістинг коду компонента авторизації

```
const result = await signIn('credentials', {
  redirect: false,
  email,
  password,
  userRole: 'clientEatwy',
  udi: getUserDeviceInfo(false, true),
});

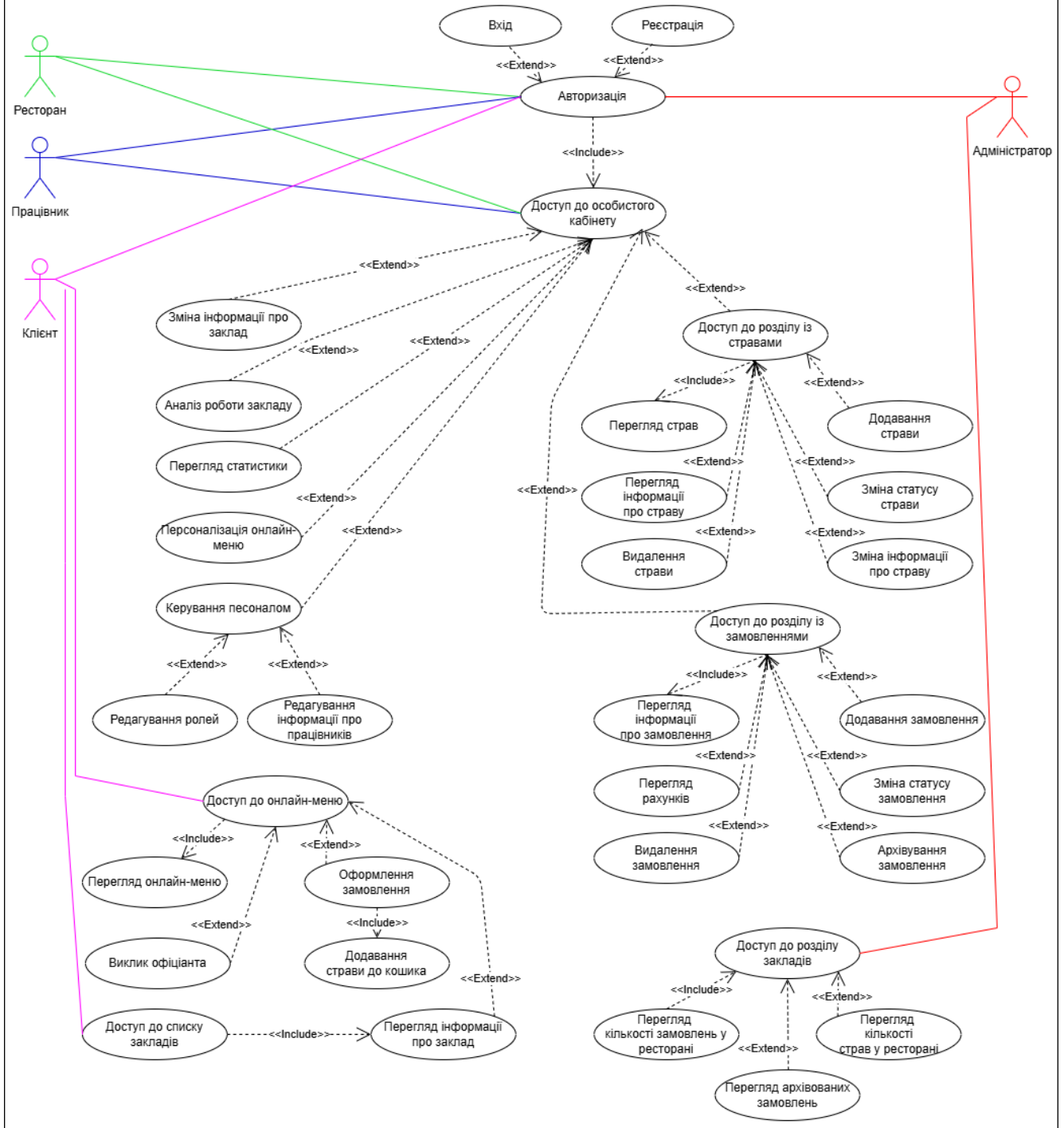
if (result?.ok) {
  return { ...successMessage, result };
} else {
  notificationToastFunc({
    language: 'uk',
    type: 'error',
    text: 'auth error',
  });

  return errorMessage({ error: 'auth error', status: 500 });
}
```

									Арк.А
									63
Зм.З	Арк.А	№ докум.№	Підпис	Дата					

123.КІ(М)-24.03

Відстеження роботи інформаційної системи для пошуку та перегляду закладів харчування



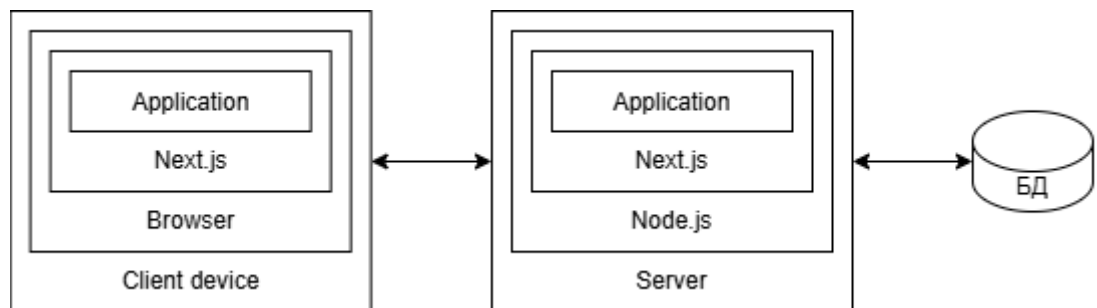
123.KI(M)-24.03

Змн.	Арк.	№ докум.	Підпис	Дата
Розробив		Білянський А. Р.		
Перевірив		Грига В.М		
Н. Контр.				
Затвердив				

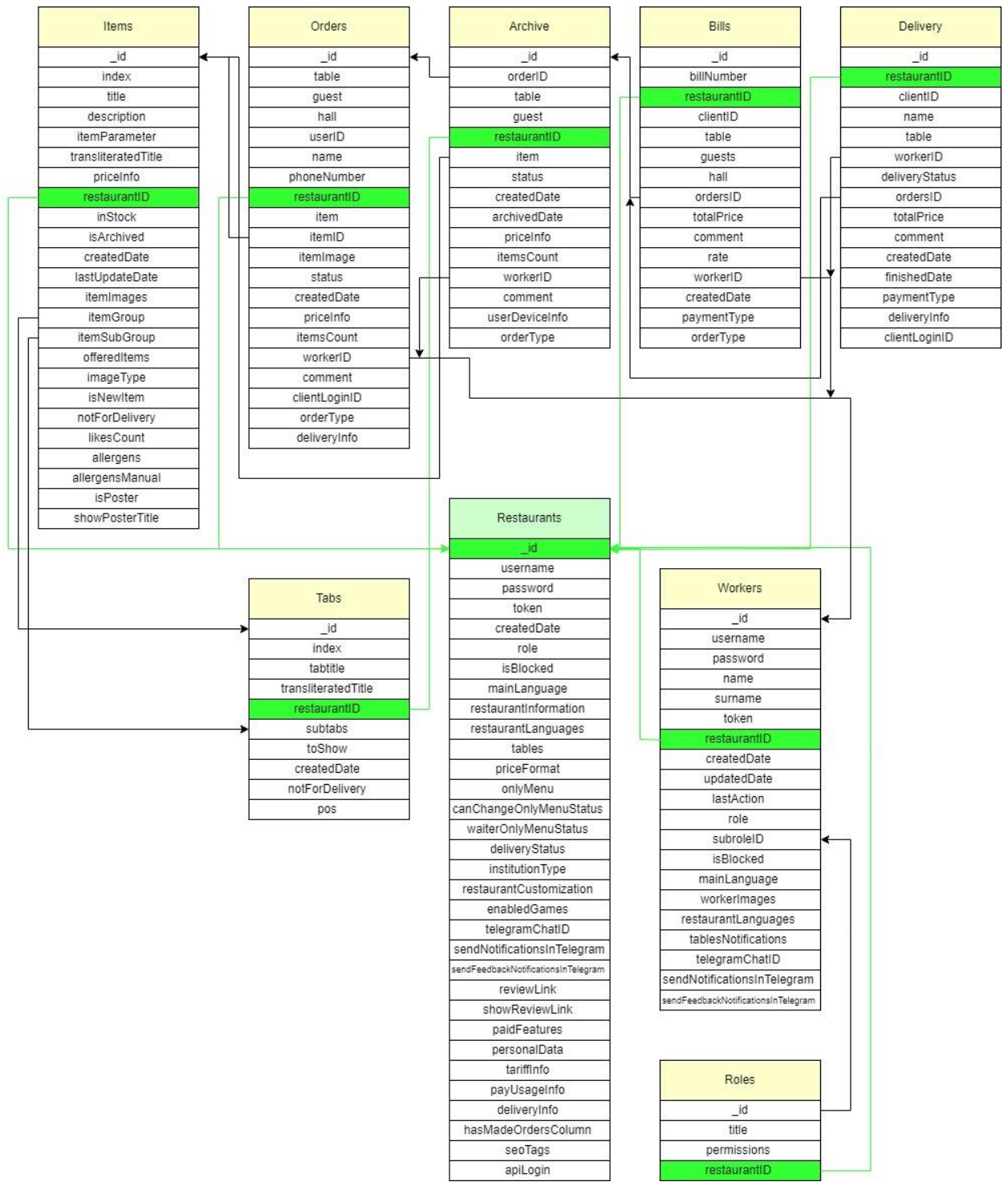
UML-діаграма  
прецедентів

Лім.	Арк.	Аркуші
	1	





					123.КІ(М)-24.03			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розробив		Білянський А. Р.			Функціональна схема СИСТЕМИ	<i>Літ.</i>	<i>Арк.</i>	<i>Аркуші</i>
Перевірив		Грига В.М					2	
Н. Контр.								
Затвердив								



					123.KI(M)-24.03			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розробив		Білянський А. Р.			Інфологічна схема бази даних	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушіє</i>
Перевірів		Грига В.М					3	
Н. Контр.								
Затвердив								

- ▼ components
  - > Antd
  - > common
  - > Dish
  - > Footer
  - > Header
  - > MetaTags
  - > PageCover
  - > pageParts
  - > PageTransition
  - > PrivateRoute
  - > RestaurantCard
  - > RestaurantCardSearch
  - > RestaurantsList
  - > ReviewCard
  - > SWRProvider
  - > UserCover
  - ≡ index.ts
- > context
- ▼ db
  - > controllers
  - > interfaces
  - > model
  - > types
  - TS connectMongo.ts
- ▼ languages
  - > addRestaurant
  - > home
  - TS auth.ts
  - TS common.ts
  - ≡ index.ts
  - TS metaTags.ts
- > node\_modules library root
- ▼ pages
  - > [city]
  - > about
  - > add-restaurant
  - > api
  - > auth

- > cashback
- > delete-from-email
- > favorites
- > feed
- > search
- > ui-kit
- > user
  - 🌀 \_app.tsx
  - 🌀 \_document.tsx
  - 🌀 index.tsx
  - 🌀 restaurants-map.tsx
  - 🌀 select-city.tsx
- > public
- > shared
- > styles
  - ≡ .env
  - 🌀 .eslintrc.json
  - 🌀 .gitignore
  - JS .lintstagedrc.js
  - ≡ .prettierrc
  - ≡ .prettierrc
  - 🌀 .stylelintrc.json
  - Y .yarnrc.yml
  - TS constants.ts
  - TS middleware.ts
  - JS next.config.js
  - TS next-env.d.ts
  - JS next-i18next.config.js
  - JS next-sitemap.config.js
  - 📁 package.json
  - 📁 package-lock.json
  - M+ README.md
  - TS restaurantMockup.ts
  - 📁 tsconfig.json
  - 👤 yarn.lock
- 📁 External Libraries
- 📄 Scratches and Consoles

					123.KI(M)-24.03		
Змн.	Арк.	№ докум.	Підпис	Дата	Файлова структура проекту		
Розробив	Білянський А. Р.						
Перевірив	Грига В.М						
Н. Контр.							
Затвердив							
					Лім.	Арк.	Аркушіє
						4	



**МАРІО/ГАЛЕРІ**

Східноєвропейська кухня | Відчинено

Є кешбек

пл. Адама Міцкевича, 8



Головна | Що? | Кешбек | Обране | Профіль



м. Івано-Франківськ, пл. Адама Міцкевича, 8

[Показати на карті](#)

+38 096 645 6596

**Графік роботи**

Відчинено зараз

**Меню**

204 страв та позицій

**Спеціальні пропозиції**

0 актуальних спеціальних пропозицій

**Відгуки**

0 · 0 відгуків

**СКОРИСТАТИСЯ КЕШБЕКОМ**

Поділіться враженнями від Маріо/

пл. Адама Міцкевича, 8



**КАЛЕ ЛАУНЖ БАР**

Італійська кухня | Відчинено

Є кешбек

вул. Низова, 3



Головна | Що? | Кешбек | Обране | Профіль

**Меню**

200 страв та позицій

**Основне меню**

Ігри | Піца | Напої | Додатк.

салати | закуски до пива | Гарячі страви



**Цезар**

160 UAH

300 г

Листя салату, черрі, куряче філе, яйця перепелині, пармезан, соус цезар, грінки



**Салат з куркою та грибами**

160 UAH

300 г

Листя салату, перець болгарський, шамп'йони, куряче філе, синя цибуля, соус, кунжут



**Салат з тунцем**

160 UAH

300 г

Салату, рукола, огірок, черрі, оливки, тунець, синя цибуля

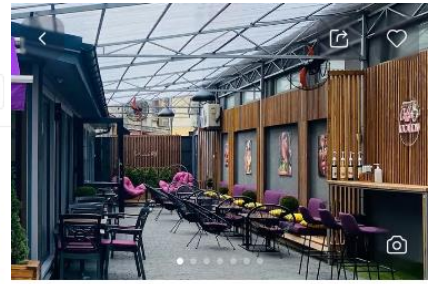


**Грецький**

150 UAH

300 г

Виставимо фото



**Маріо/Галері**

Східноєвропейська кухня

Є кешбек

1.84 км від вас

Маріо/Галері лаунж-це вільний простір для відпочинку та цікавого проведення часу!

Тут на вас чекають Sony PlayStation 5, багато окремих кімнат, перегляд улюблених фільмів, смачна піца та коктейлі, як для дорослих так і для маленьких наших гостей! Наша тераса працює круглий рік!

А найголовніше-затишок і... [Читати більше](#)

**СКОРИСТАТИСЯ КЕШБЕКОМ**

Міцкевича, 8

**Меню**

200 страв та позицій

**Основне меню**

Ігри | Піца | Напої | Додатк.



**Мортадела**

225 UAH

32 см

315 UAH

45 см



**Мортадела**

225 UAH

32 см

315 UAH

45 см

Томатна основа, сир моцарела, куряче філе, черрі, сир фета, рукола, часникова олія

					123.KI(M)-24.03			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розробив		Білянський А. Р.			Користувальницький інтерфейс	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
Перевірив		Грига В.М					5	
Н. Контр.								
Затвердив								