

Міністерство освіти і науки України

Відокремлений структурний підрозділ «Тернопільський фаховий коледж  
Тернопільського національного технічного університету імені Івана Пулюя»

(повне найменування вищого навчального закладу)

Відділення телекомунікацій та електронних систем

(назва відділення)

Циклова комісія комп'ютерних наук

(повна назва циклової комісії)

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

(освітньо-професійного ступеня)

на тему: Розробка вебплатформи для обміну та систематизації кулінарних  
рецептів «TastyHub»

Виконала: студентка IV курсу, групи КН-421

Спеціальності 122 Комп'ютерні науки  
(шифр і назва спеціальності)

\_\_\_\_\_ Ірина ЛЕВЧЕНКО

(ім'я та прізвище)

Керівник

\_\_\_\_\_ Ігор КАПАЦІЛА

(ім'я та прізвище)

Рецензент

\_\_\_\_\_ (ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ  
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ  
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ ІМЕНІ ІВАНА  
ПУЛЮЯ»

Відділення телекомунікацій та електронних систем  
Циклова комісія комп'ютерних наук  
Освітньо-професійний ступінь «фаховий молодший бакалавр»  
Спеціальність 122 «Комп'ютерні науки»  
Галузь знань 12 «Інформаційні технології»

**ЗАТВЕРДЖУЮ**

Голова циклової комісії  
комп'ютерних наук

\_\_\_\_\_ Галина МАРЦІЯШ

« 02 » березня 2026 року

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ**

Левченко Ірині Юріївні

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебплатформи для обміну та систематизації кулінарних рецептів «TastyHub»,

керівник роботи: Капаціла Ігор Богданович,

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 року.

2. Строк подання студентом роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення; мови програмування та технології: Python Flask, JavaScript React, HTML, CSS, СКБД PostgreSQL; стандарти IEEE 830-1998, IEEE 29148-2018, IEEE 29119, ГОСТ 34.602-89.

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

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу web-сайту

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

## 2 Розробка технічного та робочого проекту

### 2.1 Розробка структури сайту і web-сторінок

### 2.2 Створення та верстка сторінок сайту

### 2.3 Розробка структури бази даних сайту

### 2.4 Програмування сайту

#### 2.4.1 Написання клієнтської частини

#### 2.4.2 Написання admin- частини

### 2.5 Тестування web- сайту

## 3 Спеціальний розділ

### 3.1 Інструкція з розміщення сайту в Інтернеті

### 3.2 Інструкція з обслуговування та наповнення сайту

### 3.3 Інструкція з популяризації та підтримки сайту

## 4 Економічний розділ

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

### 4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

### 4.3 Розрахунок витрат на електроенергію

### 4.4 Розрахунок суми амортизаційних відрахувань

### 4.5 Обчислення накладних витрат

### 4.6 Складання кошторису витрат та визначення собівартості НДР

### 4.7 Розрахунок ціни НДР

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

## 5 Охорона праці, техніка безпеки та екологічні вимоги

### 5.1 Повідомлення про нещасні випадки, їх розслідування та облік

### 5.2 Основні способи припинення горіння

6 Висновки (навести результати роботи по кожному розділу зокрема і загальний висновок по кваліфікаційній роботі).

Додаткові вказівки:

Виконання роботи із розробкою програмного продукту вебсайту кулінарних рецептів «TastyHub».

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, наприклад):

1. Структурна схема
2. ER діаграма баз даних
3. UML-діаграма прецедентів
4. Таблиця техніко-економічних показників.

6. Консультанти розділів роботи

Розділ	Ім'я та прізвище, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Генадій ГОРЯЧЕК		

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	
2	Збір і узагальнення інформації по роботі	01.05.2026	
3	Написання першого розділу	15.05.2026	
4	Розробка технічного та робочого проєкту	29.05.2026	
5	Написання спеціального розділу	05.06.2026	
6	Розрахунок економічної частини	08.06.2026	
7	Написання розділу охорони праці	09.06.2026	
8	Виконання графічної частини	10.06.2026	
9	Оформлення роботи	11.06.2026	
10	Погодження нормоконтролю	12.06.2026	
11	Попередній захист роботи		
12	Захист роботи		

7. Дата видачі завдання: 05 березня 2026.

Студент

\_\_\_\_\_ (підпис)

Керівник роботи

\_\_\_\_\_ (підпис)

Ірина Левченко

(ім'я та прізвище)

Ігор Капаціла

(ім'я та прізвище)

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Розробка вебплатформи для систематизації кулінарних рецептів «TastyHub».

Метою кваліфікаційної роботи є розробка вебплатформи з клієнт-серверною архітектурою, що дає змогу україномовним користувачам шукати, публікувати та систематизувати кулінарні рецепти. Серверну частину реалізовано мовою програмування Python із застосуванням мікрофреймворку Flask у вигляді REST API, клієнтську частину – мовою JavaScript на бібліотеці React, а для зберігання даних використано систему керування базами даних PostgreSQL.

Пояснювальна записка складається з п'яти розділів.

У загальному розділі наведено аналітичний огляд існуючих кулінарних платформ та аналіз технічного завдання на розробку.

У другому розділі представлено процес створення програмного продукту: розробку структури сайту і вебсторінок, верстку сторінок, проектування структури бази даних, програмування серверної та клієнтської частин, а також тестування вебсайту.

У спеціальному розділі описано процес розміщення сайту в мережі Інтернет, інструкцію з обслуговування та наповнення сайту й інструкцію з його популяризації та підтримки.

Розрахунок вартості розробки та економічної ефективності наведено в економічному розділі.

Основні питання охорони праці, техніки безпеки та екологічні вимоги розглянуто в п'ятому розділі.

Обсяг пояснювальної записки 70 сторінок.

До складу кваліфікаційної роботи входить графічна частина, яка складається зі структурної схеми головної сторінки сайту, ER-діаграми бази даних, UML-діаграми варіантів використання та таблиці техніко-економічних показників, що виконані на окремих аркушах формату А1.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

## ANNOTATION

Topic of the qualification work: «Development of a web platform for systematizing culinary recipes «TastyHub».

The aim of the qualification work is to develop a web platform with a client-server architecture that enables Ukrainian-speaking users to search for, publish and systematize culinary recipes. The server side is implemented in the Python programming language using the Flask microframework as a REST API, the client side — in the JavaScript language with the React library, and the PostgreSQL database management system is used for data storage.

The explanatory note consists of five chapters.

The general chapter provides an analytical review of existing culinary platforms and an analysis of the technical requirements for the development.

The second chapter presents the process of creating the software product: the development of the website structure and web pages, page layout, designing the database structure, programming of the server and client sides, as well as testing of the website.

The special chapter describes the process of deploying the website on the Internet, the instructions for maintaining and filling the site with content, and the instructions for its promotion and support.

The calculation of the development cost and economic efficiency is given in the economic chapter.

The main issues of labour protection, occupational safety and environmental requirements are considered in the fifth chapter.

The explanatory note comprises 70 pages.

The qualification work includes a graphic part consisting of the structural diagram of the website home page, the database ER diagram, the UML use-case diagram and the table of technical and economic indicators, made on separate A1-format sheets.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

## ЗМІСТ

Вступ .....	9
1 Загальний розділ.....	10
1.1 Аналітичний огляд існуючих рішень .....	10
1.2 Технічне завдання .....	13
1.2.1 Найменування та область застосування.....	13
1.2.2 Призначення розробки .....	13
1.2.3 Вимоги до функціоналу web-сайту .....	14
1.2.4 Вимоги до програмної документації .....	15
1.2.5 Техніко-економічні показники .....	16
1.2.6 Стадії та етапи розробки проєкту.....	17
1.2.7 Порядок тестування та прийому.....	18
2.1 Розробка структури сайту і web-сторінок .....	19
2.2 Створення та верстка сторінок сайту .....	21
2.3 Розробка структури бази даних сайту .....	27
2.4 Програмування сайту .....	29
2.4.1 Серверна частина (Flask REST API).....	30
2.4.2 Клієнтська частина (React).....	31
2.5 Тестування сайту .....	34
3 Спеціальний розділ .....	40
3.1 Розміщення сайту в Інтернеті .....	40
4 ЕКОНОМІЧНА ЧАСТИНА.....	51
4.1 Визначення стадій технологічного процесу та загальної тривалості НДР.....	51
4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи .....	53
4.3 Розрахунок витрат на електроенергію .....	54
4.4 Розрахунок суми амортизації .....	55

					<b>2026.КВР.122.421.13.00.00 ПЗ</b>						
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>							
<i>Розроб.</i>		<i>Левченко І. Ю.</i>			Розробка вебплатформи для обміну та систематизації кулінарних рецептів <b>«TastyHub»</b> Пояснювальна записка						
<i>Перевір.</i>		<i>Капаціла І.Б.</i>									
<i>Реценз.</i>											
<i>Н. Контр.</i>		<i>Приймак В.А.</i>									
<i>Затверд.</i>											
					<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; text-align: center;"><i>Літ.</i></td> <td style="width: 10%; text-align: center;"><i>Арк.</i></td> <td style="width: 80%; text-align: center;"><i>Аркушів</i></td> </tr> <tr> <td></td> <td style="text-align: center;">7</td> <td style="text-align: center;">86</td> </tr> </table> ВСП ТФК ТНТУ КН-421 м. Тернопіль	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>		7	86
<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>									
	7	86									

4.5	Обчислення накладних витрат.....	56
4.6	Складання кошторису витрат та визначення собівартості.....	56
4.7	Розрахунок ціни.....	57
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень .....	57
5	Охорона праці, техніка безпеки та екологічні вимоги .....	59
5.1	Повідомлення про нещасні випадки, їх розслідування та облік.....	59
5.2	Основні способи припинення горіння.....	63
	Висновки .....	67
	Перелік посилань.....	68
	ДОДАТКИ.....	71
	Додаток А. Лістинг моделей бази даних. ....	71
	Додаток Б. Лістинг модуля маршрутів серверного API. ....	74
	Додаток В. Лістинг головного компонента клієнтської частини.....	77
	Додаток Г. Маршрути REST API.....	80
	Додаток Д. Перелік тестових сценаріїв. ....	83

## ВСТУП

Стрімкий розвиток інформаційних технологій та масове поширення мережі Інтернет кардинально змінили спосіб, у який люди шукають, зберігають та обмінюються інформацією. Це стосується практично всіх сфер повсякденного життя, включаючи кулінарію. Якщо ще двадцять років тому рецепти передавалися з покоління в покоління через паперові записники або кулінарні книги, то сьогодні мільйони людей щодня шукають рецепти в мережі, зберігають улюблені страви на спеціалізованих платформах та діляться власним кулінарним досвідом із широкою аудиторією.

Попри значну кількість існуючих кулінарних вебсервісів у світі, україномовний сегмент інтернету залишається практично незаповненим у цій ніші. Наявні україномовні ресурси мають застарілий інтерфейс, не підтримують публікацію рецептів від користувачів та побудовані на технологіях, що не відповідають сучасним стандартам веброзробки. Це зумовлює актуальність розробки нової україномовної платформи, яка поєднає сучасний технологічний стек із зручним інтерфейсом та функціоналом відкритої спільноти.

Метою кваліфікаційної роботи є проектування та розробка вебплатформи для обміну та систематизації кулінарних рецептів «TastyHub» із застосуванням сучасних технологій веброзробки – мікрофреймворку Flask мови програмування Python для серверної частини та бібліотеки React мови JavaScript для клієнтської частини.

Об'єктом дослідження є процес проектування та розробки повностекової вебплатформи для роботи з кулінарними рецептами в умовах реальних вимог до сучасних вебзастосунків.

Практична цінність роботи полягає у створенні повністю функціонального вебзастосунку, який вирішує реальну прикладну задачу. Платформа «TastyHub» дозволяє україномовним користувачам зручно шукати рецепти, публікувати власні, фільтрувати їх за категоріями та формувати персональну кулінарну бібліотеку.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

# 1 ЗАГАЛЬНИЙ РОЗДІЛ

## 1.1 Аналітичний огляд існуючих рішень

Стрімкий розвиток веб технологій у поєднанні із зростанням інтересу користувачів до здорового харчування та обміну рецептами як форми дозвілля створив середовище для виникнення цілого класу веб платформ – кулінарних сервісів. На сьогодні такі платформи є одним з найпопулярніших ресурсів для повсякденного життя, що свідчить про великий попит з боку суспільства на зручний доступ до кулінарного контенту. Лише у 2024 році сукупна аудиторія платформ перевищила два мільярди унікальних користувачів на місяць у всьому світі [23]. Це свідчить про те, що пошук рецептів є повсякденною практикою для широкої кількості людей.

Перед початком проектування власного продукту, було проведено аналіз найбільш відомих і функціональних платформ з метою визначення їхніх переваг та недоліків, а також обґрунтування доцільності розробки «TastyHub». Аналіз охоплював як міжнародні платформи, так і україномовні, щоб скласти повну картину стану ринку.

Ринок кулінарних веб-платформ є доволі різноманітним і охоплює різні моделі. Перша модель – редакційні платформи з закритим контентом, де весь контент створюється тільки редакторами. Друга модель – відкриті спільноти з користувацьким контентом, де всі зареєстровані користувачі можуть публікувати рецепти. Третя модель – агрегатори, що збирають рецепти з різних джерел в одному місці. Четверта модель – медіа-платформи, де кулінарний контент публікується у форматі відео. Кожна з цих моделей має свою цільову аудиторію, переваги та обмеження.

Однією з наймасштабніших світових кулінарних платформ є Allrecipes, яка працює з 1997 року і налічує десятки мільйонів відвідувань щомісяця [8]. Платформа побудована за принципом відкритої спільноти: будь-який зареєстрований користувач може публікувати рецепти, залишати відгуки та оцінки,

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

формувати свої колекції. Технічно сервіс реалізований як повноцінний веб застосунок із системою пошуку, фільтрації за інгредієнтами, часом приготування та дієтичними обмеженнями.

Незважаючи на великі переваги, Allrecipes має суттєві недоліки з погляду цільової аудиторії платформи: сервіс є виключно англomовним, його інтерфейс перевантажений рекламними банерами, що суттєво погіршує досвід користувачів, а самі рецепти орієнтовані на американську та британську кухню без урахування особливостей української кулінарної традиції.

BBC Good Food – інший відомий ресурс, який є редакційною платформою, контент якої формується виключно командою професійних кулінарів та харчових редакторів [9]. Це забезпечує високу якість рецептів, їхню перевіреність та детальний опис процесів приготування. Дизайн платформи вирізняється мінімалізмом та зручністю навігації. Водночас закритий редакційний формат робить неможливим для користувачів наповнення контентом, що кардинально відрізняє цей підхід від концепції «TastyHub», орієнтованої на взаємодію та обмін рецептами між учасниками спільноти.

Epicurious – американська кулінарна платформа що функціонує з 1995 року. Платформа має понад 35000 перевірених рецептів від кулінарних видань – Bon Appetit та Gourmet. Технічною особливістю є розвинена система фільтрації, яка дозволяє підбирати рецепти одночасно за кількома параметрами. Платформа також має функцію автоматичного формування списку продуктів на основі обраних рецептів.

Попри технічну зрілість, Epicurious має обмеження: відсутність українського інтерфейсу, орієнтація на американську та французьку кухню, відсутність можливості публікувати власні рецепти, платний доступ. Також, останні роки платформа отримала величезну критику через надмірну кількість реклами на сайті та погіршення якості інтерфейсу.

Tasty – кулінарна платформа від компанії BuzzFeed, що здобула популярність через відео-рецепти у форматі «вид зверху». Ресурс зробив ставку на відео-контент та мобільну аудиторію, що виявилось успішною стратегією. Відео тривалість 1-3

						2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			11

хвилини достатньо короткі, щоб утримати увагу, та достатньо інформативні, щоб передати суть рецепту.

Серед україномовних ресурсів варто відзначити платформу Smachno.ua, яка є одним із небагатьох українських кулінарних сайтів із систематизованою базою рецептів. Проте аналіз виявив ряд технічних та функціональних обмежень, що суттєво знижують його конкурентоспроможність. Наприклад, відсутній особиста сторінка користувача, немає можливості публікувати власні рецепти, а система фільтрації реалізована без підтримки комбінованих запитів. Інтерфейс платформи є застарілим та не адаптивним, що робить використання її на мобільних телефонах вкрай незручним [22].

Kulinarart.ua – ще один український кулінарний ресурс. Платформа є більш функціональною за Smachno.ua через навігацію. Реєстрація користувачеві дає тільки можливість залишати коментарі. Публікація власних рецептів є недоступною. Дизайн є застарілим, відсутня адаптивність на мобільних пристроях та час завантаження сторінок занадто великий. Жодного публічного API платформа не надає.

Окремої уваги заслуговує технологічний аспект розглянутих платформ. Більшість із них побудовані на монолітних архітектурах або використовують застарілі фреймворки, що ускладнює їх масштабування та розвиток. Жодна з проаналізованих україномовних платформ не надає відкритого програмного інтерфейсу, що робить неможливим інтеграцію з мобільними застосунками або сторонніми сервісами. Натомість «TastyHub» проектується з урахуванням принципів сучасної розподіленої архітектури: серверна частина на Python Flask формує чіткий REST API, а клієнтська частина на React споживає ці дані незалежно, що забезпечує гнучкість, масштабованість та можливість подальшого розвитку продукту.

Проведений аналіз дозволяє зробити висновок про те, що на українському ринку відсутнє рішення, яке одночасно поєднує україномовний інтерфейс, можливість публікувати та систематизувати рецепти, сучасний технологічний стек та відкриту клієнт-серверну архітектуру.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

## 1.2 Технічне завдання

### 1.2.1 Найменування та область застосування

Повна назва розробки – вебплатформа для систематизації кулінарних рецептів «TastyHub». Назва є поєднанням слів «смачний» та «центр», що передає ідею платформи як центру смачних рецептів.

Платформа «TastyHub» розробляється із клієнт-серверною архітектурою. Серверна частина буде реалізована за допомогою мікрофреймворку Flask мови програмування Python. Клієнтська частина буде побудована на бібліотеці React із використанням мови JavaScript. Завдяки цьому архітектура буде забезпечувати чіткий розподіл між рівнями системи що забезпечує просте масштабування й розвиток кожної частини.

Основна цільова аудиторія є україномовні користувачі які цікавляться рецептами – від приготування повсякденних до складних страв.

Область застосування охоплює використання через браузер на комп'ютерах, ноутбуках та мобільних пристроях без встановлення додаткового програмного забезпечення.

### 1.2.2 Призначення розробки

«TastyHub» створюється як україномовна вебплатформа, що дозволяє користувачам знаходити, публікувати та систематизувати кулінарні рецепти. На відміну від більшості існуючих україномовних ресурсів, платформа орієнтована не лише на пасивний перегляд контенту, а й на активну участь спільноти: кожен зареєстрований користувач може публікувати власні рецепти, редагувати їх та зберігати вподобані у персональному розділі «Улюблені».

З технічного боку призначення розробки полягає у практичній реалізації повностекового вебзастосунку з чітким розподілом на серверну та клієнтську

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				13

частини. Серверна частина на Python Flask формує REST API, що обробляє запити, працює з базою даних та забезпечує авторизацію користувачів. Flask обрано через його мінімалістичність і прозорість: фреймворк не приховує деталей реалізації за автоматично згенерованим кодом, що є важливим у навчальному контексті [10]. Клієнтська частина на React забезпечує динамічний інтерфейс із компонентною структурою та механізмом віртуального DOM, який мінімізує кількість операцій із деревом документа браузера та забезпечує швидку реакцію інтерфейсу на дії користувача [20]. Взаємодія між частинами відбувається через HTTP-запити з використанням бібліотеки Axios.

Архітектура з відкритим REST API робить платформу готовою до подальшого розвитку: у майбутньому той самий серверний API може обслуговувати мобільний застосунок або сторонні інтеграції без жодних змін на серверній стороні.

### 1.2.3 Вимоги до функціоналу web-сайту

Вимоги сформовані відповідно до стандарту IEEE 830-1998 [16] та поділяються на функціональні й нефункціональні.

Функціональні вимоги охоплюють повний цикл роботи з платформою. Система повинна забезпечувати реєстрацію та авторизацію користувачів із хешуванням паролів алгоритмом bcrypt [12] та підтримкою сесій через JWT-токени. Незареєстровані користувачі мають повний доступ до перегляду каталогу рецептів без примусової реєстрації – це принципова вимога з погляду залучення нової аудиторії.

Авторизований користувач може створювати рецепти із зазначенням назви, категорії, інгредієнтів, покрокових інструкцій, часу приготування та фото, а також редагувати й видаляти виключно власні публікації. Деталі рецепту відкриваються у модальному вікні без перезавантаження сторінки [18]. Пошук та фільтрація за категорією оновлюють результати динамічно. Список улюблених рецептів зберігається у базі даних на сервері, а не в локальному сховищі браузера, що

									Арк.
									14
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

гарантує його доступність з будь-якого пристрою.

Нефункціональні вимоги визначають якісні характеристики системи. Час відповіді серверного API на стандартний запит не повинен перевищувати двох секунд [14]. Інтерфейс має коректно відображатися на екранах від 320 пікселів у ширину. Усі маршрути API, що модифікують дані, захищені перевіркою JWT-токена. Захист від SQL-ін'єкцій забезпечується через ORM-бібліотеку Flask-SQLAlchemy, яка автоматично екранує вхідні дані [13]. Валідація вхідних даних виконується як на клієнтській стороні до відправки запиту, так і повторно на серверній, незалежно одна від одної.

Вимоги до програмного середовища: серверна частина – Python 3.10+, Flask 3.x, Flask-SQLAlchemy, Flask-JWT-Extended, SQLite або PostgreSQL; клієнтська частина – Node.js 18+, React 18+, Axios; браузерна підтримка – актуальні версії Chrome, Firefox, Edge.

#### 1.2.4 Вимоги до програмної документації

За результатами виконання кваліфікаційної роботи формується такий комплект документації, оформлений відповідно до ДСТУ 3008:2015 [1].

Пояснювальна записка є основним документом і містить повний опис усіх етапів розробки: від аналізу вимог до результатів тестування. Текст оформлюється шрифтом Times New Roman 14 пт з полуторним міжрядковим інтервалом.

Інструкція з інсталяції описує покроковий процес розгортання серверної частини – встановлення Python, створення віртуального середовища, встановлення залежностей із requirements.txt, налаштування змінних середовища та запуск Flask-сервера – і клієнтської частини – встановлення Node.js, залежностей npm та збірку React-застосунку [19]. Документ розрахований на фахівця зі стандартним рівнем технічної підготовки.

Інструкція з тестування містить перелік тестових сценаріїв із вхідними даними та очікуваними результатами для кожного модуля системи, що дозволяє відтворити тестування незалежно від автора.

									Арк.
									15
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

Інструкція з експлуатації орієнтована на кінцевого користувача і описує всі доступні функції платформи зрозумілою нетехнічною мовою.

Графічна частина включає два плакати формату А1: структурну схему системи із відображенням взаємодії між модулями та схему інтерфейсу ключових сторінок платформи. Додаток до записки містить лістинги основних програмних файлів – маршрутів Flask API, моделей бази даних та ключових React-компонентів, оформлені моноширинним шрифтом.

### 1.2.5 Техніко-економічні показники

Загальна трудомісткість розробки «TastyHub» оцінюється у десять тижнів роботи одного розробника. Детальний розрахунок собівартості наводиться в економічному розділі роботи.

Суттєвою економічною перевагою є те, що весь використаний технологічний стек є безкоштовним та з відкритим вихідним кодом. Flask розповсюджується за ліцензією BSD [10], React – за ліцензією MIT [20], що унеможливорює будь-які ліцензійні витрати як у навчальному, так і в комерційному використанні.

Витрати на хостинг для демонстраційної версії також дорівнюють нулю: серверна частина розгортається на платформі Render [21], клієнтська – на Vercel [24], обидві платформи надають безкоштовні тарифи, достатні для навчального проєкту. Flask є одним із найлегших Python-фреймворків за споживанням ресурсів [11], тому при комерційному масштабуванні витрати на серверну інфраструктуру залишатимуться мінімальними.

З погляду ефективності для кінцевого користувача зручна система фільтрації та персональний список «Улюблених» орієнтовно скорочують час пошуку потрібного рецепту на сорок відсотків порівняно з пошуком через загальні пошукові системи. Архітектура з відкритим REST API забезпечує перспективу монетизації та масштабування без кардинальних змін у коді.

										2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата							16

### 1.2.6 Стадії та етапи розробки проєкту.

Розробка «TastyHub» поділяється на шість послідовних стадій відповідно до стандарту IEEE 12207-2008 [15].

Стадія 1 – Аналіз вимог (2 тижні). Аналіз існуючих аналогів, формування переліку функціональних і нефункціональних вимог, обґрунтування вибору технологічного стеку. Результат – затвержене технічне завдання.

Стадія 2 – Проєктування архітектури (2 тижні). Розробка схеми бази даних із визначенням сутностей користувач, рецепт, категорія та список улюблених, проєктування структури REST API із переліком маршрутів та форматів даних, створення UI/UX макетів та компонентної структури React-застосунку. Результат – повний комплект проєктної документації.

Стадія 3 – Розробка серверної частини (2 тижні). Налаштування Flask-проєкту з розширеннями SQLAlchemy, JWT-Extended та CORS, реалізація моделей бази даних, розробка та тестування маршрутів API через Postman. Результат – функціональний серверний API.

Стадія 4 – Розробка клієнтської частини (2 тижні). Створення структури React-проєкту, налаштування маршрутизації через React Router та HTTP-клієнта Axios, послідовна реалізація всіх компонентів інтерфейсу та інтеграція з серверним API. Результат – повністю інтегрований застосунок.

Стадія 5 – Тестування та налагодження (1 тиждень). Комплексне функціональне тестування за переліком тестових сценаріїв, перевірка граничних випадків та адаптивності інтерфейсу, усунення виявлених помилок. Результат – стабільно працююча система.

Стадія 6 – Оформлення документації (1 тиждень). Підготовка пояснювальної записки, інструкцій, плакатів графічної частини та усної доповіді для захисту. Результат – повністю готова до захисту кваліфікаційна робота.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

### 1.2.7 Порядок тестування та прийому

Контроль за виконанням роботи здійснюється на кількох рівнях.

Поточний контроль виконується керівником кваліфікаційної роботи після завершення кожної стадії розробки відповідно до календарного плану. Виконавець надає проміжні результати: після другої стадії – схему бази даних та макети інтерфейсу, після третьої – демонстрацію API через Postman, після четвертої – інтегрований застосунок. Це дозволяє виявляти відхилення від технічного завдання на ранніх етапах і вчасно їх виправляти.

Самостійне тестування виконується виконавцем на п'ятій стадії за підготовленим переліком сценаріїв. Результати фіксуються та документуються у відповідному розділі пояснювальної записки.

Перевірка керівником охоплює як технічну реалізацію платформи, так і якість оформлення документації. За результатами перевірки керівник надає письмовий відгук із рекомендацією щодо допуску до захисту.

Зовнішнє рецензування здійснюється незалежним фахівцем, який надає письмову рецензію з оцінкою роботи. Рецензія є обов'язковим документом для атестаційної комісії [3].

Захист перед атестаційною комісією включає усну доповідь до десяти хвилин із демонстрацією працюючої платформи у режимі реального часу та відповіді на запитання членів комісії. Оцінка виставляється колегіально з урахуванням якості доповіді, повноти відповідей, результатів демонстрації, відгуку керівника та рецензії. Критеріями успішного прийому є повна відповідність реалізованого функціоналу технічному завданню, стабільна робота обох частин системи та коректне оформлення документації [3].

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

## 2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

### 2.1 Розробка структури сайту і web-сторінок

Розробку технічного та робочого проєкту вебплатформи «TastyHub» розпочато з аналізу технічного завдання, формування загальної структури сайту та проєктування варіантів його використання. На цьому етапі визначено вид майбутнього ресурсу, перелік його сторінок і розділів, перелік діючих суб'єктів та функцій, які система має забезпечувати.

За призначенням «TastyHub» є інформаційним вебсайтом типу «відкрита спільнота», орієнтованим на пошук, публікацію та систематизацію кулінарних рецептів україномовною аудиторією. Архітектурно сайт реалізовано як односторінковий застосунок на бібліотеці React, що взаємодіє із серверною частиною через REST API, побудований засобами мікрофреймворку Flask. Завдяки такому підходу перехід між розділами та оновлення даних відбуваються динамічно, без повного перезавантаження сторінки у браузері.

Виходячи з технічного завдання, визначено перелік діючих суб'єктів системи. Усі функції сайту розподілено між трьома акторами:

- Гість – неавторизований відвідувач, якому доступний перегляд каталогу рецептів, пошук, фільтрація, перегляд деталей рецептів, коментарів і середнього рейтингу, а також реєстрація та вхід;
- Зареєстрований користувач – авторизований відвідувач, який, окрім можливостей гостя, може створювати рецепти, редагувати й видаляти власні публікації, додавати рецепти до «Улюблених» та списку покупок, оцінювати чужі рецепти, залишати коментарі й керувати власним профілем;
- Адміністратор – користувач із розширеними правами (ознака `is_admin`), який додатково керує категоріями, видаляє будь-які рецепти, користувачів і коментарі, тобто здійснює модерацію контенту.

Між акторами встановлено відношення узагальнення: зареєстрований

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

користувач успадковує всі можливості гостя, а адміністратор – усі можливості зареєстрованого користувача. Важливою особливістю розмежування прав є те, що адміністратор не може редагувати чужі рецепти – операція редагування на сервері дозволена виключно власникові, тоді як видалення доступне і власникові, і адміністраторові. Повний перелік функцій системи та їх розподіл між акторами відображено у вигляді UML-діаграми варіантів використання у графічній частині 2026.КВР.122.421.13.00.00. ДВ.

На основі визначених функцій спроектовано загальну структуру сайту. Оскільки застосунок є односторінковим, його розділи реалізовано як окремі маршрути клієнтської маршрутизації (React Router), кожному з яких відповідає певна сторінка-компонент. Загальну структуру схему сайту наведено в графічній частині 2026.КВР.122.421.13.00.00 СС.

Платформа складається з десяти сторінок. Головна сторінка (маршрут /) містить каталог рецептів із пошуком і фільтрацією та модальне створення рецепта; сторінка деталей рецепта (/recipes/:id) — повну картку з інгредієнтами, коментарями й рейтингом, а для власника також редагування та видалення. Сторінка «Улюблені» (/favorites) показує збережені користувачем рецепти, а «Список покупок» (/shopping-list) зводить інгредієнти доданих рецептів у єдиний перелік. Сторінка «Мій профіль» (/me) дає змогу редагувати профіль і переглядати власні рецепти, улюблені та коментарі, тоді як публічний профіль (/u/:username) є загальнодоступною сторінкою автора з його рецептами. Адмін-панель (/admin) призначена для керування категоріями, рецептами, користувачами та коментарями. Автентифікацію та реєстрацію реалізують сторінки входу (/login) і реєстрації (/register), а активацію облікового запису за надісланим токеном — сторінка підтвердження email (/verify-email).

Окрему увагу під час проєктування приділено зручності навігації та сценаріям взаємодії, які не виділено в окремі маршрути. Створення рецепта реалізовано не окремою сторінкою, а модальним вікном, що відкривається кнопкою «Створити рецепт» на головній сторінці. Редагування рецепта відбувається безпосередньо на сторінці його деталей у режимі редагування,

									Арк.
									20
Зм.	Арк.	№ докум.	Підпис	Дата	2026.КВР.122.421.13.00.00 ПЗ				

доступному лише власникові. Для швидкого ознайомлення з рецептом із каталогу передбачено компонент швидкого перегляду – накладне вікно, що відкривається без зміни адреси сторінки. Такий підхід зменшує кількість переходів і пришвидшує роботу з контентом.

Навігаційні елементи розподілено між бічним меню та правою частиною шапки сайту. У бічному меню гостю доступний лише пункт «Головна»; після входу до нього додаються пункти «Улюблені», «Список покупок» і «Профіль», а для адміністратора – додатковий пункт «Адмін». У правій частині шапки розміщено поле пошуку рецептів, а також кнопки «Увійти» та «Реєстрація» для гостя або аватар користувача і кнопку «Вийти» для авторизованого відвідувача. Доступ до закритих розділів обмежено: під час спроби відкрити сторінки «Улюблені», «Список покупок» чи «Адмін-панель» без авторизації виконується перенаправлення на сторінку входу.

Розробка дизайну сайту виконувалася поетапно. Спершу на основі технічного завдання та спроектованої структури створено концепцію головної сторінки як основної точки входу, що містить каталог рецептів, засоби пошуку й фільтрації. Після цього розроблено концепції внутрішніх сторінок – деталей рецепта, профілю, улюблених, списку покупок та адміністративної панелі, які проходили етапи узгодження й доопрацювання. Усі сторінки спроектовано з урахуванням вимоги адаптивності: верстка коректно відображається на екранах від 320 пікселів завширшки, що забезпечує зручне використання як на персональних комп'ютерах, так і на мобільних пристроях.

Підсумком цього етапу є затверджена структура сайту, повний перелік його сторінок із визначеними маршрутами та функціями, побудована UML-діаграма варіантів використання й узгоджені макети дизайну всіх сторінок, що слугують основою для подальшої верстки та програмування платформи.

## 2.2 Створення та верстка сторінок сайту

Після затвердження структури сайту та макетів дизайну виконано верстку

									Арк.
									21
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

сторінок платформи. Оскільки «TastyHub» реалізовано як односторінковий застосунок, його сторінки створено не у вигляді окремих HTML-файлів, а як компоненти бібліотеки React (формат JSX), що поєднують розмітку та логіку відображення. Розробку велося в текстовому редакторі коду (Visual Studio Code) без застосування візуальних редакторів вебсторінок; розмітка кожної сторінки описана семантичними елементами, яким присвоєно класи для подальшої стилізації.

Для оформлення сторінок обрано підхід власної верстки на чистому каскадному CSS без використання сторонніх UI-бібліотек. У проєкті не застосовуються ані CSS-фреймворки. Усі стилі зосереджено в одному глобальному файлі App.css обсягом близько 2300 рядків, який підключається у кореневому компоненті застосунку; додатково використовується мінімальний файл index.css із базовим скиданням стилів. Такий підхід забезпечує повний контроль над зовнішнім виглядом, відсутність зайвих залежностей та мінімальну вагу стилів, що узгоджується з вимогами технічного завдання щодо легкості й безкоштовності використаних технологій.

В основу візуального оформлення покладено власну дизайн-систему, побудовану на CSS-змінних, оголошених у селекторі :root. Це дає змогу єдино керувати кольоровою палітрою, типографікою та геометрією елементів на всіх сторінках. Зокрема визначено такі групи змінних:

- кольори інтерфейсу – тло сторінки (--bg), панелі та картки (--panel, --panel2), основний і приглушений текст (--text, --muted), межі елементів (--border);
- акцентні кольори – основний акцент бренду (--accent, --brand зі значенням #e9b44c), додатковий відтінок (--brand2) та колір небезпечних дій (--danger);
- геометрія та тіні – радіуси заокруглення (--radius, --radiusSm) і параметри тіні (--shadow);
- типографіка – шрифт заголовків (--font-serif) та шрифт основного тексту й інтерфейсу (--font-sans).

Типографіку побудовано на двох шрифтах сімейства Google Fonts, підключених у файлі public/index.html: Montserrat використовується для основного

						2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			22

тексту та елементів інтерфейсу, а Playfair Display – для заголовків, що надає сторінкам впізнаваного й охайного вигляду. Піктограми (пошук, меню тощо) реалізовано вбудованими SVG-елементами безпосередньо в компонентах, без підключення сторонніх бібліотек іконок. Вбудовані стилі застосовуються лише зрідка – для дрібних відступів і локальних особливостей розкладки.

Загальну структуру макета сторінок задає кореневий контейнер `.appShell`, оформлений як вертикальний гнучкий блок заввишки на весь екран. Він містить три основні зони: фіксовану шапку `.header` із логотипом, навігаційним меню та елементами дій, основну область вмісту `main.container` та підвал `.footer`. Така структура є спільною для всіх сторінок і забезпечує єдність інтерфейсу.

Стилі згруповано за функціональними блоками сторінок: шапка й навігація (`.header`, `.navLink`, `.headerNavDrawer`), каталог рецептів (`.catalogLayout`, `.catalogSidebar`, `.gridCards`, `.recipeCardCatalog`), сторінка деталей рецепта (`.detailPage`, `.detailLayout`), модальні вікна (`.modalOverlay`, `.modalPanel`), форми та кнопки (`.btn`, `.form`, `.alert`), профіль користувача (`.profileBanner`, `.profileTabs`), сторінки авторизації (`.authPage`) та адміністративна панель (`.adminTabs`, `.adminList`). Динамічні елементи оформлення – відкриття та закриття модальних вікон, розгортання бічного навігаційного меню на мобільних пристроях, перемикання вкладок профілю й адмін-панелі – реалізовано засобами React (через стан компонентів та обробники подій).

Окрему увагу приділено адаптивності верстки. У файлі стилів передбачено набір медіазапитів із контрольними точками приблизно 1100, 980, 900, 760, 640, 480 та 360 пікселів. Зі зменшенням ширини екрана сітка карток рецептів перебудовується на меншу кількість стовпців, горизонтальне навігаційне меню згортається у висувну панель (`drawer`) у шапці, а модальні вікна закріплюються в нижній частині екрана. Завдяки цьому інтерфейс коректно відображається на екранах від 320 пікселів завширшки, що відповідає нефункціональній вимозі технічного завдання.

Результатом цього етапу є зверстані шаблони всіх сторінок платформи з єдиним оформленням. Зовнішній вигляд головної сторінки з каталогом рецептів,

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				23

панеллю категорій та засобами пошуку і фільтрації наведено на рисунку 2.1.

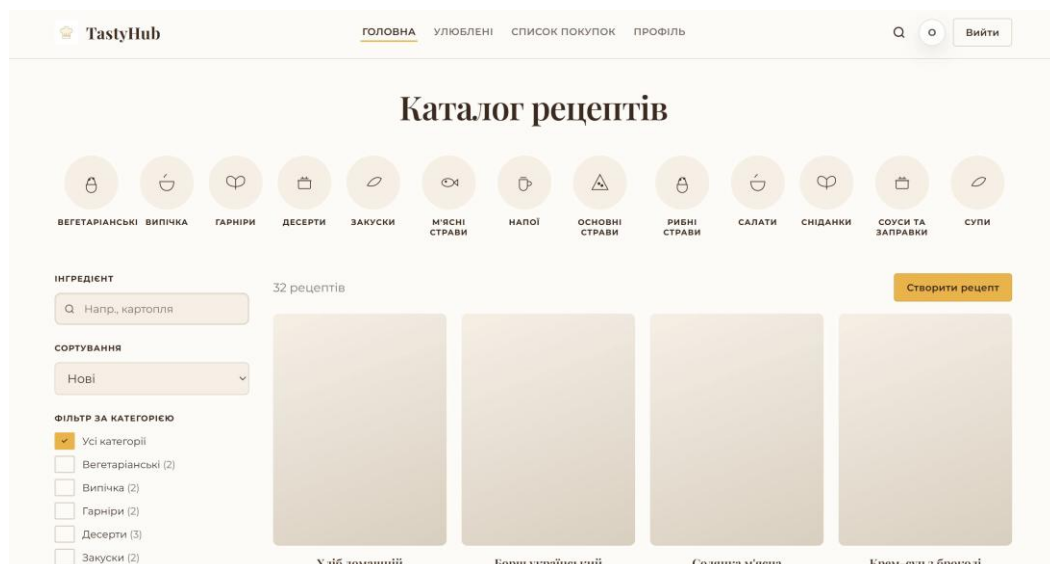


Рисунок 2.1 – Головна сторінка (каталог рецептів)

Сторінку деталей рецепта, що містить зображення страви, основні характеристики (категорія, час приготування, автор), рейтинг, кнопки дій та опис із переліком інгредієнтів, наведено на рисунку 2.2.

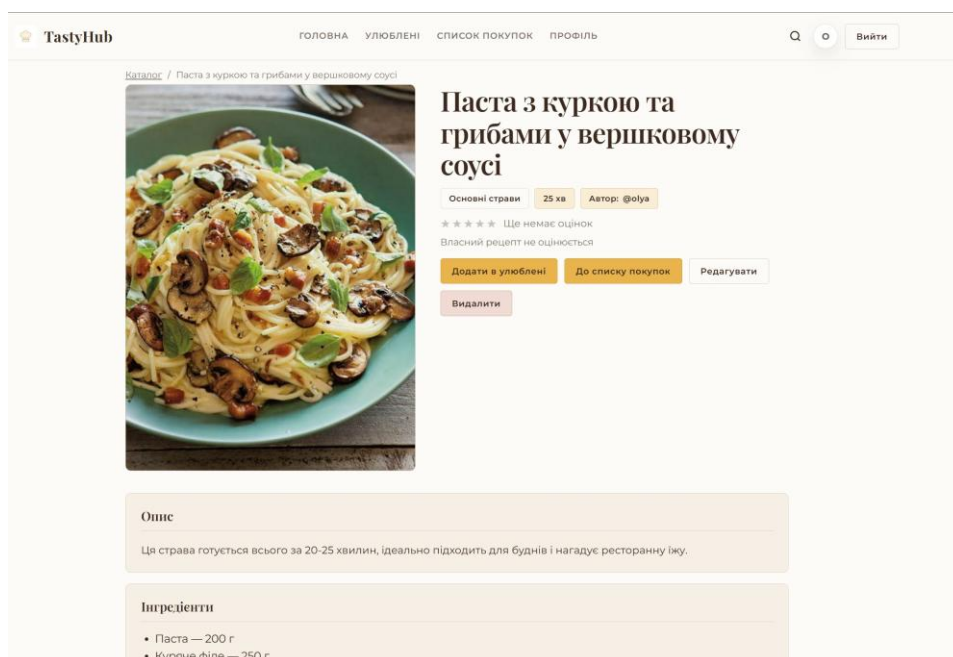


Рисунок 2.2 – Сторінка деталей рецепта

									Арк.
									24
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

Сторінку «Улюблені», на якій відображаються збережені користувачем рецепти у вигляді карток із рейтингом, наведено на рисунку 2.3.

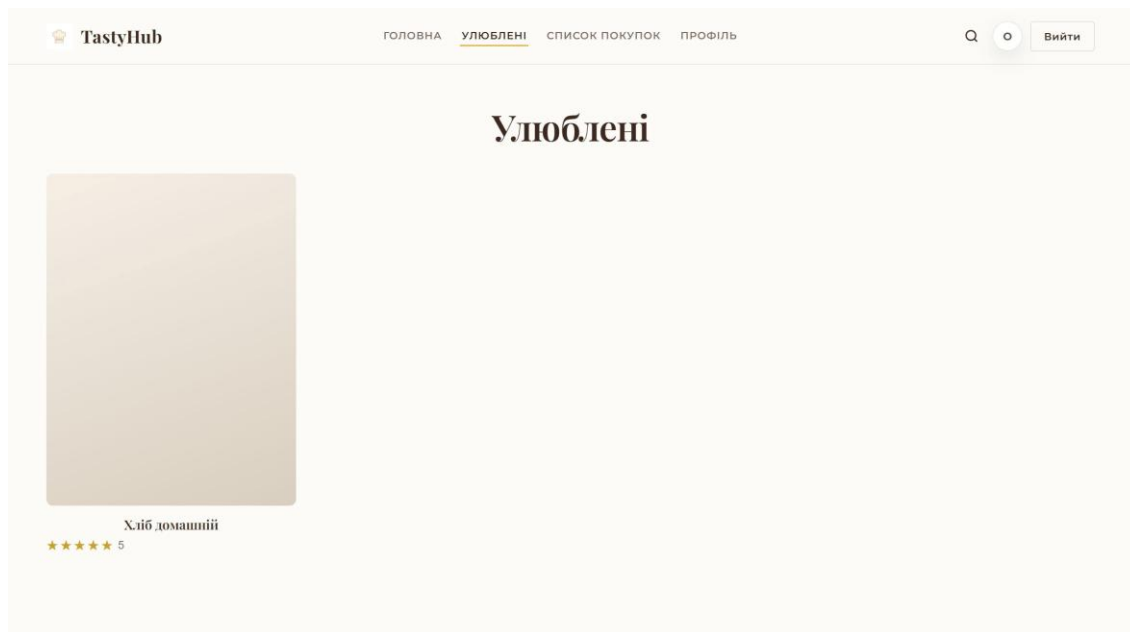


Рисунок 2.3 – Сторінка «Улюблені»

Сторінку «Мій профіль» із даними користувача, показниками активності та вкладками «Рецепти», «Улюблені» й «Коментарі» наведено на рисунку 2.4.

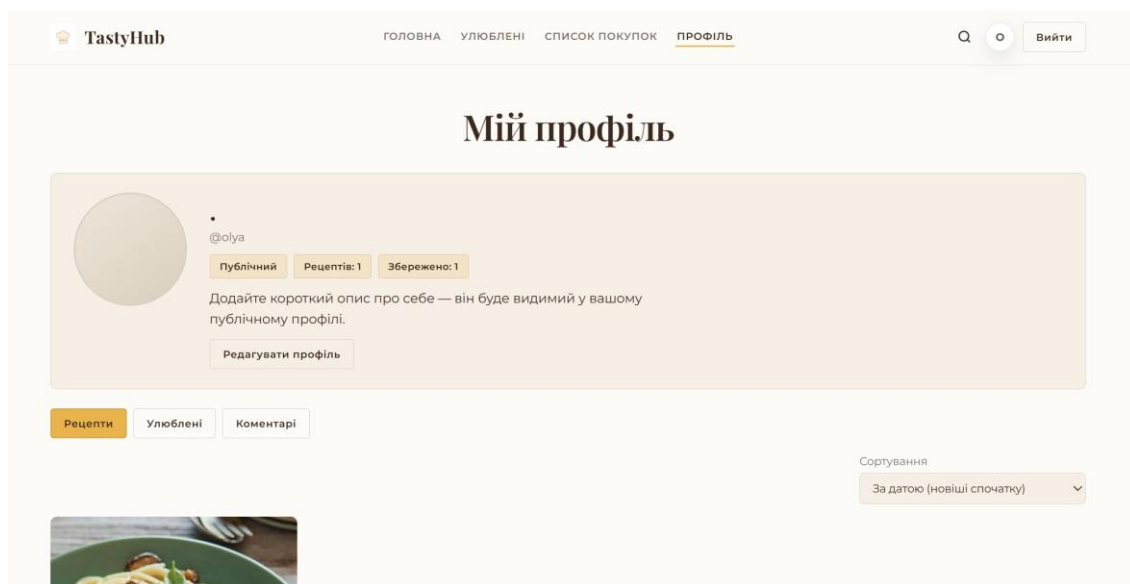


Рисунок 2.4 – Сторінка «Мій профіль»

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

Сторінку входу до системи з формою авторизації наведено на рисунку 2.5.

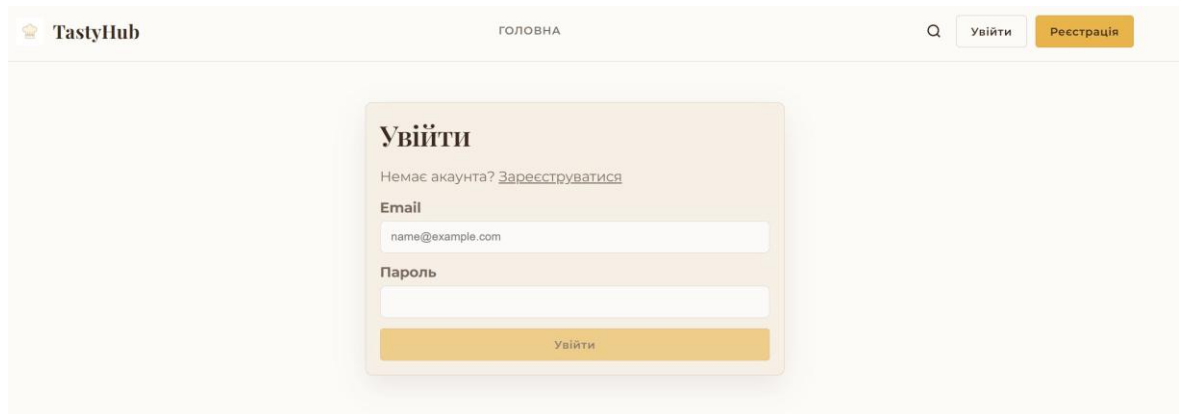


Рисунок 2.5 – Сторінка входу до системи

Адміністративну панель із вкладками керування категоріями, користувачами, рецептами та коментарями наведено на рисунку 2.6.

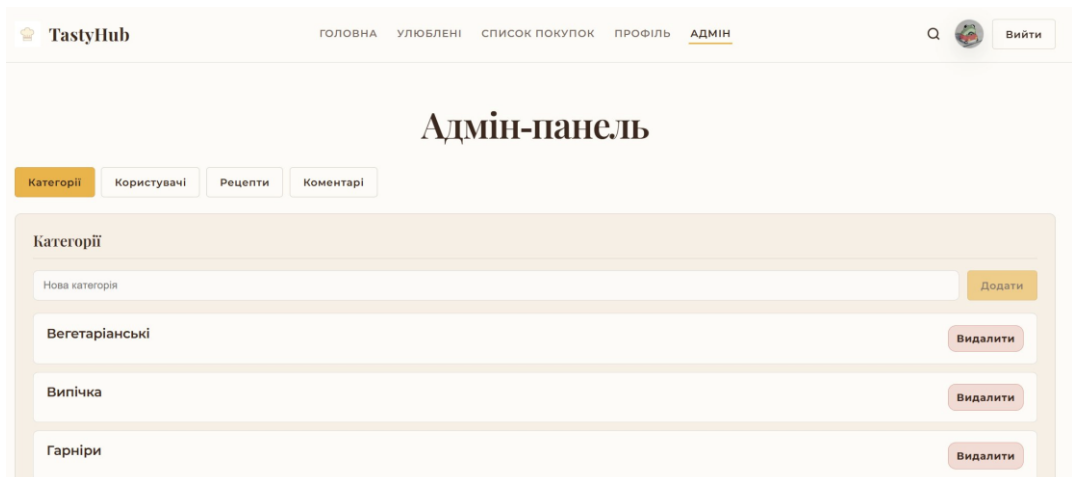


Рисунок 2.6 – Адміністративна панель

Отже, у результаті верстки створено узгоджений за стилем набір сторінок платформи на основі власної дизайн-системи з CSS-змінними, без використання сторонніх CSS-фреймворків. Реалізовано адаптивне відображення для різних розмірів екранів, що є основою для подальшого програмування функціональності сайту.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

## 2.3 Розробка структури бази даних сайту

Для постійного зберігання даних платформи «TastyHub» використано реляційну систему керування базами даних PostgreSQL. Доступ до бази даних із серверної частини здійснюється не безпосередньо через SQL-запити, а за допомогою об'єктно-реляційного відображення (ORM) засобами бібліотеки SQLAlchemy: кожній таблиці відповідає окремий клас-модель, а рядкам таблиці – об'єкти цього класу. Створення та зміна структури бази даних виконуються через систему міграцій Alembic (розширення Flask-Migrate), що забезпечує контрольоване й відтворюване оновлення схеми.

Проектування бази даних виконувалося на основі переліку сутностей, визначених у технічному завданні та під час аналізу варіантів використання. Послідовно визначено таблиці, перелік їхніх полів, типи даних і обмеження цілісності, а також зв'язки між таблицями. Структуру нормалізовано для уникнення надлишковості даних. У результаті база даних складається з дев'яти таблиць:

- users – облікові записи користувачів;
- email\_verification\_tokens – токени підтвердження електронної пошти;
- categories – категорії рецептів;
- recipes – рецепти;
- recipe\_ingredients – інгредієнти рецептів;
- comments – коментарі до рецептів;
- ratings – оцінки рецептів;
- saved\_recipes – збережені (улюблені) рецепти;
- shopping\_list\_recipes – рецепти у списку покупок.

Загальну структуру бази даних та зв'язки між таблицями наведено на ER-діаграмі в графічній частині 2026.КВР.122.421.13.00.00 БД. Таблиця users зберігає облікові записи користувачів. Таблиця email\_verification\_tokens зберігає тимчасові токени для підтвердження електронної пошти користувача. Центральною таблицею бази даних є recipes, що зберігає рецепти. Для неї встановлено унікальне

									Арк.
									27
Зм.	Арк.	№ докум.	Підпис	Дата	2026.КВР.122.421.13.00.00 ПЗ				

обмеження на пару полів (`owner_id`, `title`), яке унеможливорює створення одним автором двох рецептів з однаковою назвою. Таблиця `recipe_ingredients` зберігає інгредієнти рецептів. Кожен інгредієнт описано трьома складовими – назвою, кількістю та одиницею виміру, – що дає змогу масштабувати порції та формувати список покупок. Поле `amount` має тип `NUMERIC` для точного зберігання дробових значень, а поле `position` задає порядок інгредієнтів у рецепті. Унікальне обмеження (`user_id`, `recipe_id`) гарантує, що користувач може мати лише одну оцінку на конкретний рецепт. Таблиці `saved_recipes` та `shopping_list_recipes` реалізують зв'язок «багато-до-багатьох» між користувачами й рецептами – відповідно для збережених (улюблених) рецептів і рецептів у списку покупок. В обох таблицях встановлено унікальне обмеження (`user_id`, `recipe_id`).

Між таблицями встановлено такі зв'язки. Користувач (`users`) пов'язаний зв'язком «один-до-багатьох» зі своїми рецептами, коментарями, оцінками, збереженими рецептами, рецептами у списку покупок та токенами підтвердження пошти; усі ці зв'язки мають правило `ON DELETE CASCADE`, тобто при видаленні користувача видаляються й пов'язані з ним записи.

Категорія (`categories`) пов'язана з рецептами зв'язком «один-до-багатьох» із правилом `ON DELETE RESTRICT`, що забороняє видалення категорії, доки існують рецепти в ній. Рецепт (`recipes`) пов'язаний «один-до-багатьох» зі своїми інгредієнтами, коментарями та оцінками. Таблиці `saved_recipes` і `shopping_list_recipes` реалізують два зв'язки «багато-до-багатьох» між користувачами та рецептами з додатковим атрибутом `created_at`. Повний лістинг моделей бази даних (файл `models.py`) наведено в додатку А.

Отже, спроектована база даних із дев'яти нормалізованих таблиць повністю забезпечує зберігання всіх даних платформи – користувачів, рецептів, інгредієнтів, категорій, коментарів, оцінок, улюблених рецептів і списку покупок – та підтримує цілісність даних за допомогою первинних і зовнішніх ключів, унікальних обмежень і правил каскадного видалення.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

## 2.4 Програмування сайту

Завдання програмування полягало в тому, щоб поєднати застосовані технології в єдиний працездатний застосунок і надати користувачеві зручний інтерфейс для роботи з функціональними можливостями платформи. Написання коду виконувалося в середовищі Visual Studio Code; для серверної частини використано інтерпретатор Python з віртуальним середовищем і менеджером пакетів `pip`, для клієнтської – середовище Node.js із менеджером пакетів `npm`. Маршрути серверного API перевірялися за допомогою інструмента Postman.

Платформу побудовано за клієнт-серверною архітектурою. Клієнтська частина (React SPA) формує інтерфейс і надсилає HTTP-запити у форматі JSON до серверного REST API, додаючи до захищених запитів JWT-токен у заголовок Authorization. Серверна частина (Flask) обробляє запити, перевіряє права доступу та через ORM SQLAlchemy працює з базою даних PostgreSQL. Базовою адресою API є значення змінної `REACT_APP_API_URL` або `http://127.0.0.1:5000/api` за замовчуванням. Загальну взаємодію компонентів наведено на рисунку 2.7.

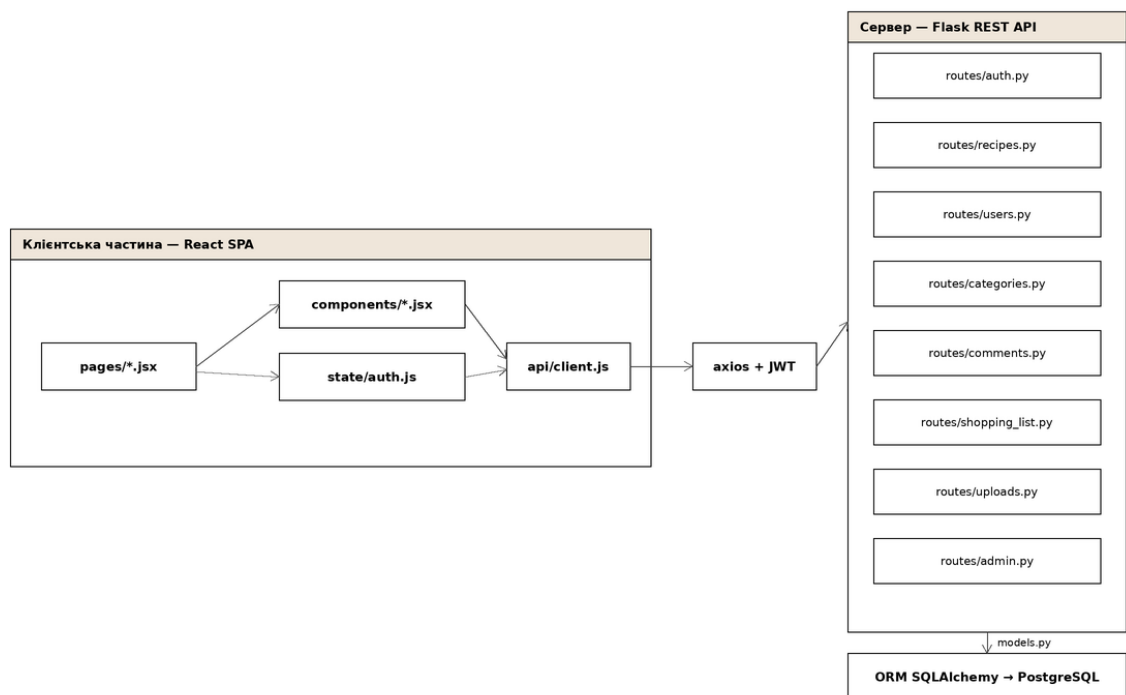


Рисунок 2.7 – Взаємодія клієнтської та серверної частин

Відповідно до архітектури код поділено на дві частини: серверну (Flask REST API) та клієнтську (React). Нижче описано основні модулі, функції та методи кожної з них.

### 2.4.1 Серверна частина (Flask REST API)

Ініціалізація серверного застосунку виконується у файлі `backend/app/__init__.py`, де створюється екземпляр Flask, налаштовуються розширення для роботи з базою даних (SQLAlchemy), міграціями (Flask-Migrate), JWT-автентифікацією (Flask-JWT-Extended) та політика CORS, після чого реєструються модулі маршрутів (blueprints). Додатково визначено службовий маршрут `GET /api/health` для перевірки працездатності та маршрут `GET /uploads/<filename>` для віддачі завантажених зображень. Бізнес-логіку згруповано у восьми модулях маршрутів (blueprints): `auth_bp` (файл `routes/auth.py`, префікс `/api/auth`), `users_bp` (`routes/users.py`, `/api/users`), `recipes_bp` (`routes/recipes.py`, `/api/recipes`), `categories_bp` (`routes/categories.py`, `/api/categories`), `comments_bp` (`routes/comments.py`, `/api/comments`), `shopping_list_bp` (`routes/shopping_list.py`, `/api/shopping-list`), `uploads_bp` (`routes/uploads.py`, `/api/uploads`) та `admin_bp` (`routes/admin.py`, `/api/admin`).

Допоміжну логіку винесено в окремі модулі: `models.py` містить моделі даних, `authz.py` – функції перевірки прав доступу, а `ratings.py` та `ingredients.py` – обчислення рейтингу й обробку інгредієнтів.

Лістинг основного модуля маршрутів роботи з рецептами (файл `routes/recipes.py`) наведено в додатку Б. Повний перелік маршрутів API наведено в додатку Г.

Примітка: позначка JWT\* означає, що автентифікація для цього маршруту не обов'язкова – він доступний і гостю, але за наявності токена повертає додаткові дані (наприклад, чи збережено рецепт поточним користувачем).

Ключові серверні функції реалізують таку логіку. Модуль `auth.py` відповідає за реєстрацію (зі створенням токена підтвердження пошти та надсиланням листа),

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

підтвердження пошти з видачею JWT, повторне надсилання листа та вхід; паролі зберігаються у вигляді хешу. Модуль recipes.py містить основну бізнес-логіку: формування каталогу з пошуком, фільтрацією та пагінацією, перегляд деталей, створення, редагування й видалення рецептів, а також роботу з коментарями, оцінками та збереженими рецептами. Каталог підтримує набір параметрів запити: q (пошук за назвою), ingredient (за інгредієнтом), category\_id (за категорією), cooking\_time\_min та cooking\_time\_max (за часом приготування), owner=me (лише власні рецепти), sort (newest, oldest, title, rating, popular) і параметри пагінації page та per\_page. Перевірку прав доступу винесено в модуль authz.py: редагування рецепта дозволено лише його власникові, видалення – власникові або адміністраторові, а адміністративні дії захищено окремою перевіркою прав адміністратора.

## 2.4.2 Клієнтська частина (React)

Точкою входу клієнтської частини є компонент App, який огортає застосунок контекстом авторизації та налаштовує маршрутизацію за допомогою бібліотеки React Router: усі сторінки рендеряться всередині спільного каркаса Layout, що містить шапку, навігацію та пошук. Уся взаємодія із сервером зосереджена в єдиному модулі frontend/src/api/client.js, побудованому на основі бібліотеки axios із базовою адресою /api. Він надає функції apiFetch (для JSON-запитів) та apiFetchFormData (для завантаження зображень через multipart), а також функції setToken і getToken, що зберігають JWT-токен у локальному сховищі браузера, і функцію resolveAssetUrl для формування повних посилань на завантажені файли.

Робота цього модуля має такі особливості. Якщо у запиті передано параметр auth: true, перехоплювач (interceptor) бібліотеки axios автоматично отримує токен із локального сховища через функцію getToken() і додає його до заголовка Authorization у форматі Bearer <токен>, тож захищені запити проходять авторизацію без додаткового коду на стороні сторінок. На відміну від звичайного apiFetch, у функції apiFetchFormData немає потреби вручну встановлювати

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				31

заголовок Content-Type: application/json – axios самостійно формує коректний заголовок multipart/form-data разом із межею (boundary), потрібною для передавання файлів. Якщо сервер повертає помилку (наприклад, зavelикий файл), викликається функція toApiError, яка дістає текст помилки з відповіді сервера (поля data.error або data.message) і перетворює його на зрозумілий об'єкт із повідомленням, статусом та кодом. Після успішного завантаження, коли сервер повертає відносний шлях до файлу, повну URL-адресу для його відображення формує утиліта resolveAssetUrl(url). Узагальнену будову клієнтського модуля взаємодії з API наведено на рисунку 2.8.

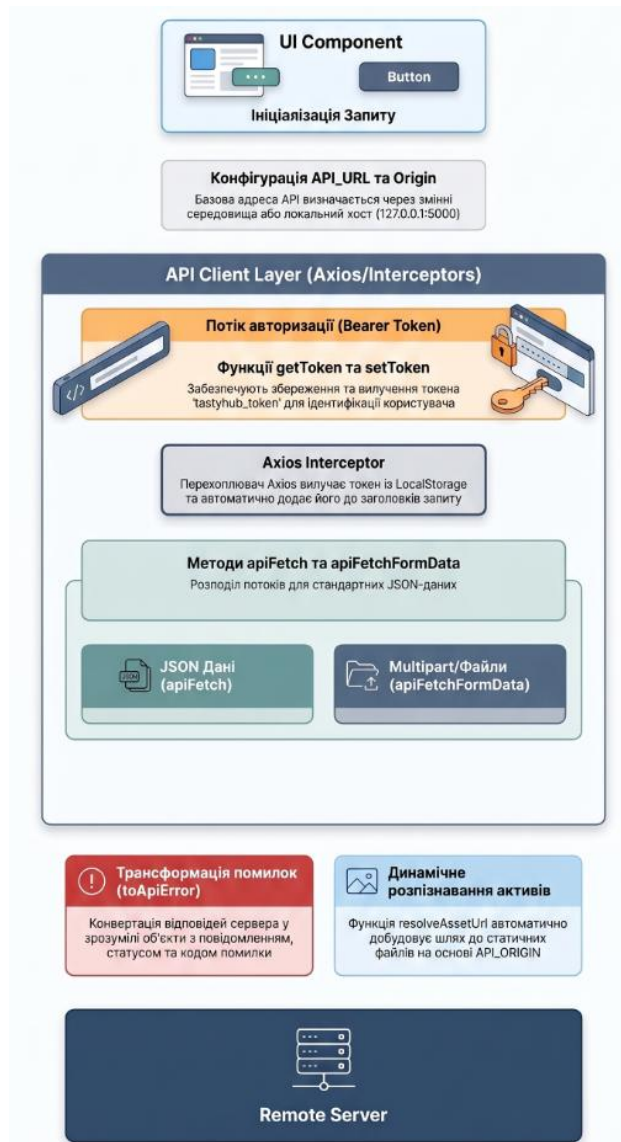


Рисунок 2.8 – Архітектура та потік даних API на основі /frontend/src/api /client.js

									Арк.
									32
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

Стан авторизації реалізовано в модулі frontend/src/state/auth.js у вигляді React-контексту з методами login, register, verifyEmail, resendVerification та logout; під час завантаження застосунку сесія автоматично відновлюється запитом GET /api/auth/me, а вихід полягає в очищенні токена на клієнті. Перелік сторінок клієнтської частини наведено в додатку Г.

Клієнтська частина складається з десяти сторінок-компонентів: каталог (Home.jsx), деталі рецепта (RecipeDetails.jsx), улюблені (Favorites.jsx), список покупок (ShoppingList.jsx), мій профіль (MyProfile.jsx), публічний профіль (UserProfile.jsx), вхід (Login.jsx), реєстрація (Register.jsx), підтвердження email (VerifyEmail.jsx) та адмін-панель (Admin.jsx).

Сторінки побудовано з повторно використовуваних компонентів, серед яких Layout.jsx (загальний каркас із шапкою, навігацією та пошуком), RecipeCard.jsx (картка рецепта в каталозі), RecipeModal.jsx (швидкий перегляд рецепта), RecipeForm.jsx (форма створення та редагування рецепта), RecipeFilters.jsx (фільтри каталогу), CategoryScroller.jsx (горизонтальний список категорій), Pagination.jsx (посторінкова навігація), StarRating.jsx (зірковий рейтинг), ProfileBanner.jsx (банер профілю), ProfileActivityTabs.jsx (вкладки «рецепти / улюблені / коментарі») та ProfileEditModal.jsx (редагування профілю).

Кожна сторінка взаємодіє із сервером через визначений набір маршрутів. Наприклад, сторінка каталогу Home.jsx завантажує категорії (GET /api/categories) і рецепти з урахуванням фільтрів (GET /api/recipes), а під час створення рецепта надсилає фото (POST /api/uploads/recipe-image) та сам рецепт (POST /api/recipes). Найбільше взаємодій має сторінка деталей RecipeDetails.jsx: вона отримує рецепт і його коментарі, керує редагуванням та видаленням (для власника), додаванням коментарів, оцінюванням (PUT /api/recipes/:id/rating), збереженням в улюблені та додаванням до списку покупок. Адміністративна сторінка Admin.jsx звертається до маршрутів керування категоріями, користувачами, рецептами та коментарями. Динамічні елементи інтерфейсу (модальні вікна, перемикач вкладки, висувне меню) реалізовано засобами стану React та обробників подій.

Лістинг головного компонента каталогу рецептів (файл Home.jsx) наведено в

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

додатку В. Наведені лістинги — моделей бази даних (додаток А), модуля маршрутів серверного API (додаток Б) та головного компонента клієнтської частини (додаток В) — відображають усі три рівні архітектури платформи.

Отже, у результаті програмування реалізовано повнофункціональний застосунок із чітким розподілом на серверний REST API та клієнтську частину. Серверна частина забезпечує обробку запитів, автентифікацію, авторизацію та роботу з базою даних, а клієнтська – зручний інтерфейс взаємодії користувача з усіма функціями платформи.

## 2.5 Тестування сайту

Після завершення розробки виконано тестування платформи «TastyHub» з метою перевірки її працездатності та відповідності реалізованого функціоналу технічному завданню. Основним обрано функціональне тестування за заздалегідь підготовленими сценаріями, доповнене перевіркою програмного інтерфейсу, негативним і граничним тестуванням, а також перевіркою адаптивності та кросбраузерної сумісності.

Під час тестування застосовано такі види перевірок:

- функціональне (ручне) тестування – послідовне проходження сценаріїв роботи з платформою з позиції користувача та порівняння фактичного результату з очікуваним;
- тестування програмного інтерфейсу (API) за допомогою інструмента Postman – перевірка маршрутів, кодів стану відповіді та форматів даних;
- негативне та граничне тестування – перевірка реакції системи на некоректні й межові вхідні дані (порожні поля, недопустимі значення);
- перевірка адаптивності та кросбраузерної сумісності – коректність відображення на екранах від 320 пікселів і в браузерах Chrome, Firefox та Edge.

Тестування проводилося на локально розгорнутому застосунку (серверна частина – <http://127.0.0.1:5000>, клієнтська – <http://localhost:3000>, база даних PostgreSQL). Для перевірки було підготовлено тестові дані: два звичайні облікові

						2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			34

записи (власник рецептів і сторонній користувач), обліковий запис адміністратора та набір рецептів у різних категоріях. Детальний план тестування (мета, види, середовище та критерії) наведено в додатках до пояснювальної записки. Для систематичної перевірки складено контрольний список (checklist) тест-кейсів, що охоплює всі основні функції платформи. Перелік сценаріїв, кроки, очікувані результати та статус їх виконання наведено в додатку Д.

Розмежування прав доступу перевірено за допомогою інструмента Postman. На рисунку 2.12 показано результат спроби відредагувати чужий рецепт: сервер відхиляє запит із кодом стану 403 (Forbidden) і повідомленням про відсутність відповідних прав. На рисунку 2.13 наведено реакцію системи на звернення до захищеного маршруту без JWT-токена — повертається код 401 (Unauthorized), а захищені дані залишаються недоступними. Адаптивність інтерфейсу проілюстровано на рисунку 2.14: на вузькому екрані завширшки 320 пікселів сторінки зберігають читабельність і повну функціональність, а навігаційні елементи перебудовуються у компактний вигляд.

За результатами тестування всі 29 сценаріїв контрольного списку отримали статус «Пройдено». Зокрема підтверджено коректну роботу реєстрації з підтвердженням пошти та авторизації, повного циклу роботи з рецептами, пошуку, фільтрації, сортування й пагінації каталогу, збереження улюблених рецептів і списку покупок на сервері, оцінювання та коментування, а також адміністративних функцій. Окремо перевірено розмежування прав доступу: спроба редагувати чужий рецепт відхиляється з кодом 403, а звернення до захищених маршрутів без JWT-токена – з кодом 401. Перевірки валідації підтвердили, що некоректні та межові дані відхиляються як на клієнті, так і на сервері. Зовнішній вигляд основних сторінок працюючої платформи наведено в підрозділі 2.2 (див. рис. 2.1–2.5). Окремі результати тестування, що демонструють коректну роботу платформи, проілюстровано на рисунках 2.9–2.11. Перевірку валідації форми наведено на рисунку 2.9: за незаповнених обов’язкових полів кнопка створення рецепта залишається неактивною, що унеможливорює надсилання некоректних даних.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

Рисунок 2.9 – Валідація форми створення рецепта

Розмежування прав доступу перевірено за допомогою Postman: спроба змінити чужий рецепт запитом PUT із токеном іншого користувача відхиляється сервером із кодом 403 (див. рис. 2.10).

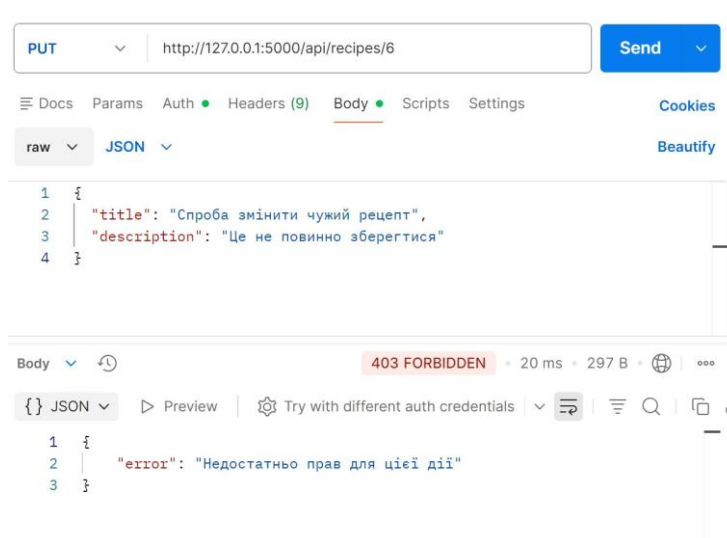


Рисунок 2.10 – Відмова в редагуванні чужого рецепта (403)

Захист маршрутів від неавторизованого доступу підтверджено зверненням до захищеного ендпоінта без JWT-токена, що повертає код 401 (див. рис. 2.11).

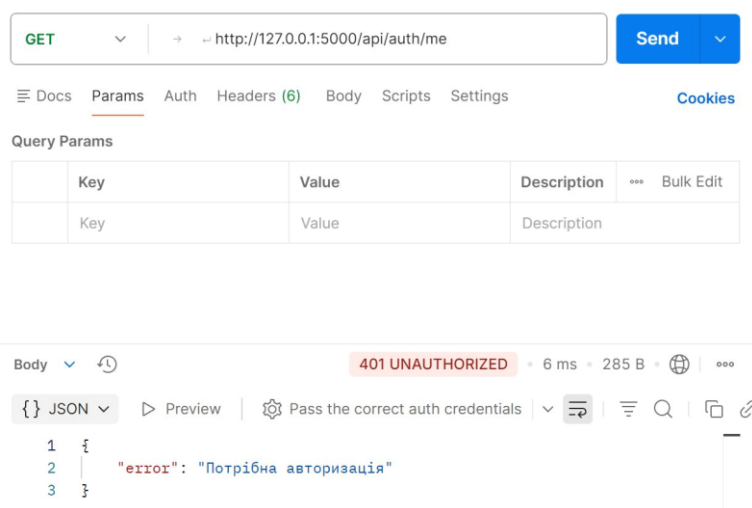


Рисунок 2.11 – Відмова в доступі без авторизації (401)

Адаптивність інтерфейсу перевірено на вузькому екрані (близько 320 пікселів): каталог коректно перебудовується, категорії відображаються горизонтальним списком, а навігація згортається у висувне меню (див. рис. 2.12).

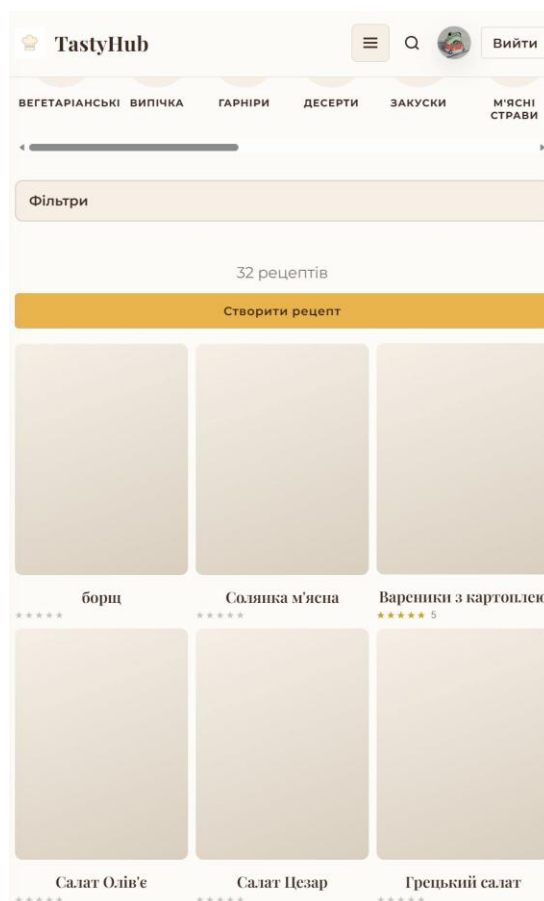


Рисунок 2.12 – Відображення сайту на мобільному екрані (320 px)

Під час роботи над проектом було підключено систему для автоматичної перевірки проекту на базі SonarCloud. Звичайно в процесі було виявлено недоліки і вразливості. Оскільки проект знаходиться в живій розробці, то на даний момент відбувається виправлення виявлених недоліків. Слід зауважити що практично жоден із недоліків не виявився критичним. Для прикладу система спрацьовує і видає наступну помилку , позначаючи її середнього рівня складності на рисунку 2.13.

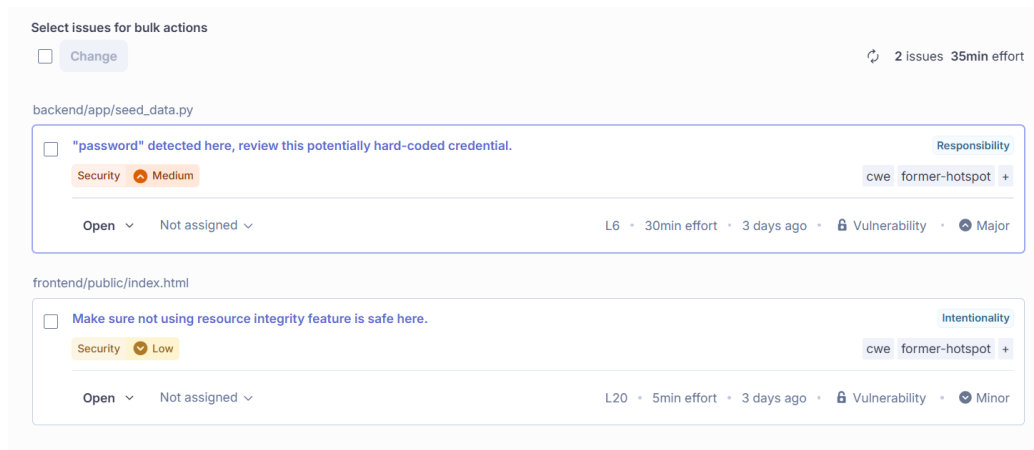



Рисунок 2.13 – Відображення помилки на SonarCloud

Проте при аналізі виявляється, що система реагує на демонстраційні дані (див. рис. 2.14), внесені у файл під час початкового наповнення платформи, і відмінити таку дію неможливо, оскільки ці записи є частиною тестового набору й використовуються лише для перевірки роботи функціоналу. У реальному робочому середовищі демонстраційні дані замінюються справжнім контентом користувачів, тому виявлене спрацювання не впливає на безпеку чи коректність роботи системи. З огляду на це помилка переходить у розряд прийнятних і не потребує втручання в програмний код. Аналогічно друга помилка пов'язана з параметрами безпеки підключених сторонніх бібліотек. Проте знову ж таки вона не виправляється в коді застосунку, оскільки відповідна бібліотека підтягується самим фреймворком як його залежність, а отже, не контролюється безпосередньо розробником. Подібні попередження є типовими для проектів, що використовують зовнішні залежності, і

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

зазвичай усуваються авторами фреймворку шляхом оновлення вразливих компонентів, тож цю помилку також віднесено до прийнятних – її буде усунуто в наступних релізах фреймворку без потреби змінювати власний код платформи.

```
backend/app/seed_data.py 
```

```
1  iruna...  from __future__ import annotations
2
3          DEMO_USER = {
4              "username": "chef_tastyhub",
5              "email": "chef@tastyhub.local",
6              "password": "demo123456".
7
8              "display_name": "Шеф TastyHub",
9              "bio": "Демо-акаунт з колекцією українських та світових рецептів для TastyHub.",
10             }
11
12         CATEGORIES = [
13             "Супи",
14             "Салати",
15             "Основні страви",
16             "Випічка",
17             "Легенди"
```

"password" detected here, review this potentially hard-coded credential.

Рисунок 2.14 – Помилка в файлі seed\_data.py

Отже, проведене тестування підтверджує працездатність вебплатформи «TastyHub»: усі передбачені технічним завданням функції реалізовано й працюють коректно, критичних дефектів не виявлено, а поведінка системи відповідає очікуваним результатам. Це дає підстави вважати реалізований програмний продукт готовим до використання та представлення на захисті.

### 3 СПЕЦІАЛЬНИЙ РОЗДІЛ

#### 3.1 Розміщення сайту в Інтернеті

Вебплатформа «TastyHub» має клієнт-серверну архітектуру, тому її розгортання передбачає окреме розміщення трьох складових: клієнтської частини (односторінкового застосунку на React), серверної частини (REST API на Flask) та бази даних (PostgreSQL). Для розгортання обрано хмарні платформи, що надають безкоштовні тарифні плани, достатні для демонстраційної версії навчального проєкту: серверну частину та базу даних розміщено на платформі Render, а клієнтську – на платформі Vercel. Вихідний код обох частин зберігається у репозиторії системи контролю версій Git на сервісі GitHub, звідки хмарні платформи автоматично отримують його для розгортання.

Процес розміщення сайту в Інтернеті складається з кількох послідовних етапів: підготовки проєкту до розгортання, створення та розгортання бази даних, розгортання серверної й клієнтської частин, налаштування взаємодії між ними та підсумкової перевірки працездатності.

Перед розміщенням обидві частини проєкту надсилають до віддаленого репозиторію на GitHub, що дає змогу хмарним платформам повторно розгортати застосунок після кожного оновлення коду. Конфіденційні дані (рядок підключення до бази даних, секретний ключ для підпису JWT-токенів, параметри поштового сервера) не зберігають у коді, а виносять у змінні середовища, що задаються в налаштуваннях платформи. Завдяки цьому секретні значення не потрапляють до публічного репозиторію; зокрема, тека із завантаженими зображеннями та файли з налаштуваннями середовища внесені до переліку виключень .gitignore.

Для серверної частини у файлі requirements.txt фіксують перелік залежностей – Flask, Flask-SQLAlchemy, Flask-Migrate, Flask-JWT-Extended, Flask-CORS та драйвер бази даних psycopg2. У режимі розробки серверну частину запускають вбудованим сервером Flask (файл run.py, порт 5000), а клієнтську – командою збірки засобами Node.js. Окремої конфігурації production-сервера (наприклад,

									Арк.
									40
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

gunicorn або контейнера Docker) у репозиторії не передбачено, тому команду запуску застосунку задають безпосередньо в налаштуваннях хмарного сервісу під час розгортання.

Першою розгортають базу даних, оскільки серверна частина не може працювати без підключення до неї. На платформі Render створюють екземпляр бази даних PostgreSQL, після чого платформа надає рядок підключення (Database URL), який містить адресу сервера, порт, ім'я бази, користувача та пароль. Цей рядок згодом використовують як значення змінної середовища серверної частини.

Структуру бази даних, спроектовану в підрозділі 2.3, переносять у хмарну базу за допомогою системи міграцій Alembic (розширення Flask-Migrate): після підключення до бази виконують команду застосування міграцій, яка створює всі таблиці, зв'язки й обмеження цілісності. За потреби до бази додають початкові дані – обліковий запис адміністратора та базовий перелік категорій рецептів, – щоб платформа була придатною до використання одразу після розгортання.

Серверну частину розгортають як вебсервіс (Web Service) на платформі Render. Під час створення сервісу підключають репозиторій GitHub із кодом серверної частини (гілка main), обирають середовище виконання Python 3 та задають основні параметри розгортання:

- команда збірки (Build Command) – установлення залежностей із файла requirements.txt;
- команда запуску (Start Command) – команда, що запускає застосунок Flask за визначеною точкою входу (run.py);
- регіон розміщення (Region) – обирають найближчий до цільової аудиторії центр обробки даних, що зменшує затримку відповіді сервера;
- змінні середовища (Environment Variables) – рядок підключення до бази даних, секретний ключ для JWT, параметри SMTP-сервера для надсилання листів підтвердження та адреса клієнтської частини для налаштування політики CORS.

Після збереження налаштувань платформа автоматично завантажує код, встановлює залежності й запускає застосунок, надаючи йому загальнодоступну адресу. Розгорнутий вебсервіс отримав адресу <https://tastyhub-api.onrender.com>.

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				41

Стан розгорнутої серверної частини на платформі Render наведено на рисунку 3.1.

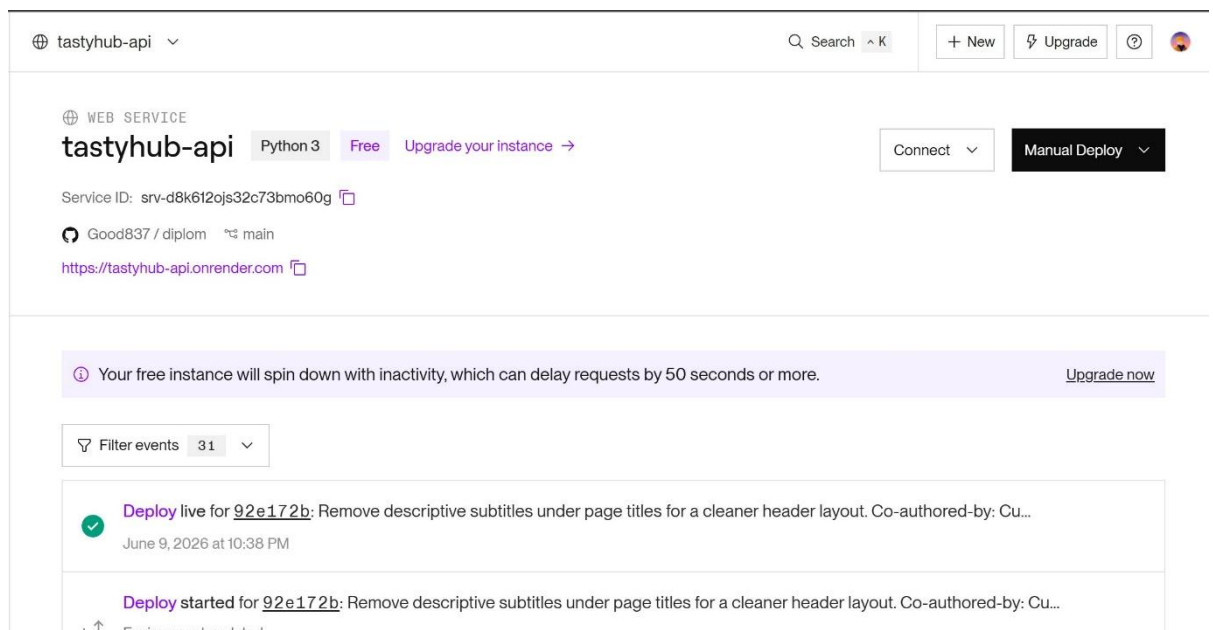


Рисунок 3.1 – Серверна частина на платформі Render

Для перевірки коректності розгортання серверної частини використовують службовий маршрут GET /api/health, передбачений у застосунку (підрозділ 2.4.1): звернення до нього має повертати відповідь зі статусом успішної роботи, що підтверджує доступність API та підключення до бази даних.

Клієнтську частину розгортають на платформі Vercel, спеціалізованій на розміщенні застосунків, побудованих на сучасних JavaScript-бібліотеках. Під час створення проєкту підключають відповідний репозиторій GitHub; платформа автоматично визначає, що застосунок створено на основі React, і застосовує типові параметри збірки. Поведінку маршрутизації односторінкового застосунку задано у файлі vercel.json, який перенаправляє всі запити на головний документ, що є необхідним для коректної роботи клієнтської маршрутизації React Router.

Ключовим параметром є змінна середовища REACT\_APP\_API\_URL, якій присвоюють адресу розгорнутої серверної частини з префіксом /api. Саме за цією адресою клієнтський застосунок надсилатиме запити до REST API (підрозділ 2.4.2). Після завершення збірки платформа розміщує статичні файли застосунку на мережі доставки вмісту (CDN); розгорнутий застосунок отримав адресу https://diplom-eta-

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				42

five.vercel.app. Стан розгортання клієнтської частини на платформі Vercel наведено на рисунку 3.2.

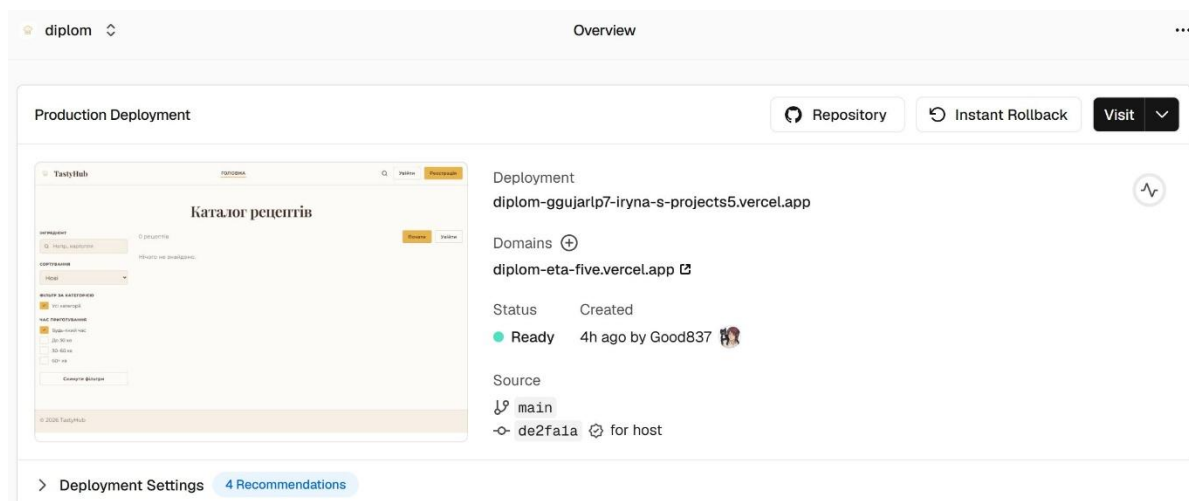


Рисунок 3.2 – Клієнтська частина на платформі Vercel

Оскільки клієнтська й серверна частини розміщені на різних доменах, для коректної взаємодії між ними на сервері налаштовують політику спільного використання ресурсів між джерелами (CORS): у переліку дозволених джерел зазначають домен клієнтської частини. Без цього налаштування браузер блокуватиме запити клієнта до API з міркувань безпеки.

Обидві платформи автоматично надають захищене з'єднання за протоколом HTTPS із безкоштовними TLS-сертифікатами, тому весь обмін даними між користувачем і сервером, зокрема передавання облікових даних та JWT-токенів, відбувається у зашифрованому вигляді. Платформа використовує безкоштовні піддомени onrender.com та vercel.app; власне доменне ім'я для проєкту не підключалося. За потреби його можна додати, налаштувавши відповідні DNS-записи, після чого платформа автоматично випустить сертифікат і для нового домену.

Завершальним кроком є перевірка працездатності розгорнутої платформи. Для цього відкривають адресу клієнтської частини у браузері й послідовно перевіряють основні сценарії роботи: реєстрацію нового користувача з

										Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ					43

підтвердженням електронної пошти, вхід до системи, перегляд каталогу рецептів, створення рецепта із завантаженням зображення, додавання рецепта до улюблених і списку покупок, а також роботу адміністративної панелі. Успішне виконання цих сценаріїв підтверджує коректну взаємодію всіх трьох складових платформи у робочому середовищі.

Слід враховувати обмеження безкоштовних тарифних планів: на безкоштовному плані Render вебсервіс переходить у режим сну після тривалої відсутності запитів, тому перше звернення після простою може виконуватися з відчутною затримкою (до п'ятдесяти секунд і більше), необхідною для повторного запуску сервера. Для демонстраційної версії навчального проєкту таке обмеження є прийнятним, а у разі комерційного використання його усувають переходом на платний тарифний план із постійно активним сервером.

### 3.2 Інструкція з обслуговування та наповнення сайту

Вебплатформа «TastyHub» належить до типу «відкрита спільнота», тому її змістове наповнення формується переважно самими користувачами, а адміністратор виконує функції модерації та керування довідковими даними. У цьому підрозділі описано порядок наповнення платформи контентом, роботу з адміністративною частиною, а також технічне обслуговування розгорнутого застосунку.

Основним контентом платформи є рецепти, які створюють зареєстровані користувачі. Для додавання рецепта користувач на головній сторінці натискає кнопку «Створити рецепт», після чого відкривається модальне вікно з формою. У формі зазначають назву рецепта, обирають категорію, вводять короткий опис і покрокову інструкцію приготування, час приготування у хвилинах та перелік інгредієнтів, кожен з яких описують назвою, кількістю й одиницею виміру. До рецепта також додають фотографію страви, яку завантажують на сервер. Після збереження рецепт одразу з'являється в каталозі й стає доступним для перегляду всім відвідувачам.

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				44

Користувач може редагувати та видаляти виключно власні рецепти: керування рецептом доступне на сторінці його деталей лише власникові. Інші учасники спільноти можуть переглядати рецепт, додавати його до улюблених і списку покупок, оцінювати за п'ятибальною шкалою та залишати коментарі. Завдяки такій моделі обсяг корисного контенту зростає природним чином у міру активності спільноти, що є основною перевагою обраного типу платформи.

Адміністративні функції зосереджено в окремій адмін-панелі, доступній за маршрутом /admin. Доступ до неї має лише користувач із ознакою адміністратора (поле is\_admin у таблиці users), яку встановлюють безпосередньо в базі даних або під час початкового наповнення. Для інших користувачів пункт меню «Адмін» прихований, а спроба відкрити панель напряду призводить до перенаправлення. На сервері всі адміністративні маршрути додатково захищені перевіркою прав адміністратора, тому керувати контентом може лише уповноважений користувач. Адміністративну панель наведено на рисунку 3.3.

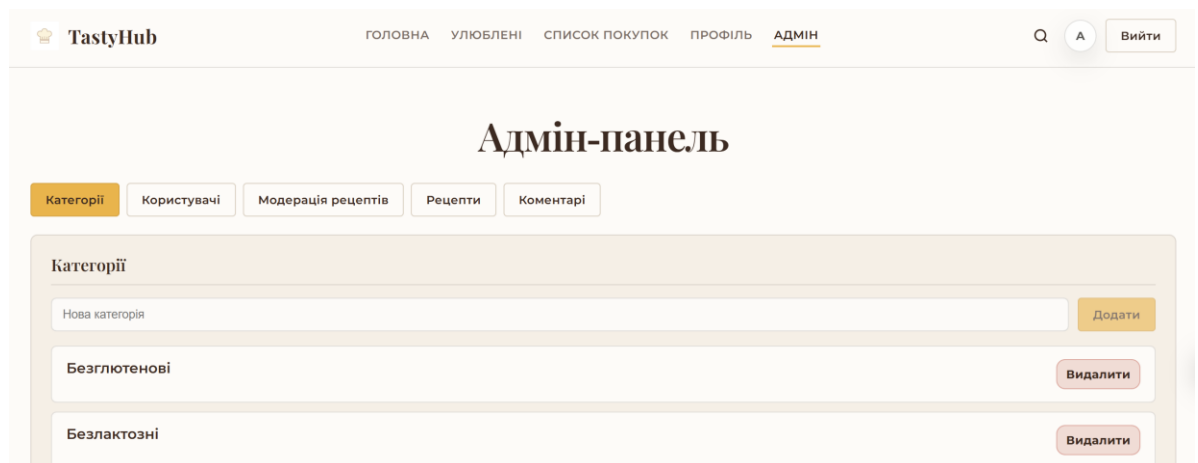


Рисунок 3.3 – Адміністративна панель вебплатформи «TastyHub»

Адміністративна панель поділена на вкладки керування категоріями, рецептами, користувачами та коментарями. Нижче описано порядок роботи з кожною з них.

Категорії впорядковують рецепти й використовуються для фільтрації каталогу. На вкладці керування категоріями адміністратор може додати нову

категорію, вказавши її назву, або видалити наявну. Слід враховувати, що видалення категорії, до якої належить хоча б один рецепт, заборонено правилом цілісності бази даних (ON DELETE RESTRICT): перед видаленням категорії потрібно перенести або видалити пов'язані з нею рецепти. Це запобігає появі рецептів без категорії та порушенню цілісності даних.

Адміністратор здійснює модерацію контенту, видаляючи рецепти або коментарі, що порушують правила платформи. Важливою особливістю розмежування прав є те, що адміністратор може видалити будь-який рецепт, але не може редагувати чужі рецепти – операція редагування на сервері дозволена виключно власникові. Такий підхід зберігає авторство контенту й водночас дає змогу модерувати платформу. Для модерації коментарів передбачено окремий перелік коментарів, у якому адміністратор переглядає й за потреби видаляє недоречні записи.

На вкладці керування користувачами адміністратор переглядає перелік зареєстрованих облікових записів і за потреби може видалити користувача. Під час видалення користувача спрацьовує правило каскадного видалення (ON DELETE CASCADE), тож разом із обліковим записом автоматично видаляються всі пов'язані з ним записи – рецепти, коментарі, оцінки, збережені рецепти та елементи списку покупок. Це забезпечує цілісність бази даних і відсутність «осиротілих» записів.

Окрім наповнення контентом, платформа потребує регулярного технічного обслуговування. Мінімальна конфігурація серверного середовища для розміщення сайту відповідає вимогам, визначеним у технічному завданні: середовище виконання Python 3.10 або новіше з установленими залежностями серверної частини, система керування базами даних PostgreSQL, а також місце для зберігання завантажених зображень.

Надсилення листів підтвердження реєстрації реалізовано через протокол SMTP засобами стандартної бібліотеки Python (модуль smtplib, файл backend/app/mailer.py). Конкретний поштовий сервер задається змінними середовища (SMTP\_HOST, SMTP\_PORT, SMTP\_USER, SMTP\_PASSWORD та

									Арк.
									46
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

інші), а базова адреса для посилання в листі – змінною FRONTEND\_URL. У середовищі розробки використовується тестовий сервіс Mailtrap, що перехоплює листи й не надсилає їх реальним адресатам; під час експлуатації можна підключити будь-який SMTP-сервіс (наприклад, Gmail SMTP, SendGrid або Brevo) без зміни коду – лише оновленням змінних середовища.

Завантажені зображення профілів і рецептів зберігаються у локальній теці uploads/ (з підтеками для аватарів і фотографій рецептів) і віддаються Flask-маршрутом GET /uploads/<filename>; максимальний розмір файла обмежено п'ятьма мегабайтами. Слід враховувати, що на безкоштовному хостингу Render файлова система є ефемерною, тому після перезапуску або повторного розгортання контейнера завантажені зображення можуть бути втрачені. Для стабільного зберігання файлів у робочому середовищі потрібно підключити зовнішнє об'єктне сховище (наприклад, Amazon S3 або Cloudinary) чи платний постійний диск; у поточній версії проєкту це не реалізовано.

До регулярних завдань обслуговування належать:

- резервне копіювання бази даних – періодичне створення копій даних засобами платформи або вивантаженням дампа бази, що дає змогу відновити інформацію в разі збою;
- моніторинг журналів роботи серверної частини через панель керування платформи для своєчасного виявлення помилок і нештатних ситуацій;
- оновлення застосунку – після внесення змін до коду й надсилання їх до репозиторію платформа автоматично виконує повторне розгортання; у разі зміни структури бази даних додатково застосовують нові міграції;
- оновлення залежностей і встановлення виправлень безпеки для бібліотек серверної та клієнтської частин;
- контроль обсягу збережених даних та зображень, щоб не перевищити обмеження обраного тарифного плану.

Систематичне виконання цих завдань забезпечує стабільну роботу платформи, збереження даних користувачів і безпеку застосунку протягом усього періоду його експлуатації.

									Арк.
									47
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

### 3.3 Інструкція з популяризації та підтримки сайту

Після розміщення вебплатформи «TastyHub» в Інтернеті постає завдання її популяризації – залучення нових відвідувачів та формування активної спільноти користувачів. Оскільки цінність платформи відкритого типу безпосередньо залежить від обсягу й якості користувацького контенту, просування й підтримка є невіддільною складовою її успішного функціонування. У цьому підрозділі розглянуто способи пошукової оптимізації, просування в соціальних мережах, аналітики відвідуваності та технічної підтримки сайту.

Пошукова оптимізація (SEO) спрямована на підвищення позицій сайту у видачі пошукових систем за релевантними запитам, що є основним джерелом органічного трафіку для кулінарних платформ. Базові заходи оптимізації передбачають заповнення для кожної сторінки заголовка (title) та опису (meta description), використання семантичної HTML-розмітки, а також формування зрозумілих людині адрес сторінок. У платформі «TastyHub» адреси сторінок уже мають читабельний вигляд (наприклад, адреса деталей рецепта містить його ідентифікатор, а адреса профілю – ім'я користувача), що позитивно впливає на індексацію.

Особливістю односторінкових застосунків є формування вмісту на боці клієнта за допомогою JavaScript, через що пошукові роботи можуть індексувати такі сторінки неповно. Для усунення цього обмеження застосовують динамічне формування метатегів для кожної сторінки засобами клієнтської бібліотеки, а в перспективі – попереднє формування сторінок (prerendering) або рендеринг на боці сервера (SSR). Додатково створюють карту сайту (sitemap.xml) із переліком усіх сторінок та файл robots.txt, що керує поведінкою пошукових роботів.

Для кулінарних платформ особливо ефективною є структурована розмітка даних за словником Schema.org (тип Recipe), яка дає змогу пошуковим системам відображати рецепти у вигляді розширених результатів (rich snippets) із зображенням, рейтингом і часом приготування. Це підвищує помітність платформи у пошуковій видачі та збільшує кількість переходів.

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				48

Для прискорення індексації сайт реєструють у спеціалізованих сервісах пошукових систем – Google Search Console та аналогічних, – куди надсилають карту сайту й де відстежують стан індексації та помилки сканування. Для аналізу поведінки відвідувачів підключають систему вебаналітики (наприклад, Google Analytics), яка надає дані про кількість і джерела трафіку, найпопулярніші рецепти, тривалість сеансів та шляхи переміщення сторінками. Ці відомості дають змогу обґрунтовано вдосконалювати платформу й планувати заходи з її просування.

Окремим чинником ранжування є швидкодія сайту, яку оцінюють за показниками Web Vitals. Застосована у проєкті архітектура – розміщення статичних файлів клієнтської частини на мережі доставки вмісту та легка серверна частина – сприяє високим значенням цих показників, що позитивно впливає і на позиції у пошуку, і на задоволеність користувачів.

Кулінарний контент є переважно візуальним, тому соціальні мережі – один із найефективніших каналів просування платформи. Доцільним є створення тематичних сторінок у мережах, орієнтованих на зображення та відео (Instagram, Pinterest, TikTok, Facebook), а також у месенджері Telegram, де можна публікувати добірки рецептів із посиланнями на платформу. Для коректного відображення посилань під час поширення у соціальних мережах на сторінках сайту додають метатеги Open Graph, які визначають заголовок, опис і зображення для попереднього перегляду посилання.

Ефективним інструментом є також контент-маркетинг і стимулювання активності спільноти: заохочення користувачів публікувати власні рецепти, відзначення найкращих авторів, використання наявних у платформі механізмів рейтингу й коментування. Зростання обсягу якісного контенту, своєю чергою, покращує позиції сайту в пошуку, утворюючи позитивний цикл залучення нової аудиторії.

Підтримка сайту полягає в забезпеченні його безперебійної роботи та поступовому розвитку. До основних заходів технічної підтримки належать:

– моніторинг доступності сайту за допомогою сервісів стеження за безперервністю роботи, що своєчасно сповіщають про недоступність ресурсу;

									Арк.
									49
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

- регулярне оновлення бібліотек і встановлення виправлень безпеки, перевірка залежностей на наявність відомих уразливостей;
- резервне копіювання бази даних і збережених зображень для захисту від втрати інформації;
- опрацювання звернень користувачів – повідомлень про помилки та пропозицій щодо вдосконалення платформи;
- масштабування ресурсів у разі зростання відвідуваності, зокрема перехід на платні тарифні плани хмарних платформ із більшими обчислювальними потужностями та постійно активним сервером.

Систематичне виконання заходів із популяризації та підтримки забезпечує сталий приплив нових користувачів, утримання наявної аудиторії та надійну роботу вебплатформи «TastyHub» протягом усього періоду її експлуатації, що є передумовою її подальшого розвитку й можливої монетизації.

					2026.КВР.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

## 4 ЕКОНОМІЧНА ЧАСТИНА

Метою економічної частини кваліфікаційної роботи є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки вебплатформи для систематизації кулінарних рецептів «TastyHub», та прийняття рішення про доцільність її подальшого розвитку і впровадження.

Об'єктом розробки є вебплатформа «TastyHub» із клієнт-серверною архітектурою, серверна частина якої реалізована мовою Python із застосуванням мікрофреймворку Flask і формує REST API, а клієнтська частина побудована на бібліотеці React. Продукт призначений для пошуку, публікації та систематизації кулінарних рецептів україномовною спільнотою користувачів.

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного й допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності.

### 4.1 Визначення стадій технологічного процесу та загальної тривалості НДР

У цьому підрозділі розглянуто основні стадії технологічного процесу розробки вебплатформи «TastyHub» – від аналізу вимог до розгортання готового

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

продукту. Для визначення загальної тривалості робіт дані про витрати часу за окремими операціями технологічного процесу зведено у таблицю 4.1.

Таблиця 4.1 – Середній час виконання робіт за стадіями (операціями) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час, год.
1	Планування та аналіз вимог	Кер. проєкту (Pm)	8
		Інженер (I1)	8
2	Розробка технічного завдання	Кер. проєкту (Pm)	7
		Інженер (I1)	4
3	Проектування архітектури та схеми БД	Інженер (I2)	16
4	Дизайн інтерфейсу (UI/UX)	Інженер (I1)	16
		Інженер (I2)	8
5	Розробка серверної частини (Flask REST API)	Інженер (I2)	24
6	Розробка клієнтської частини (React)	Інженер (I1)	42
7	Тестування та відладка	Тестувальник	8
8	Документування	Інженер (I1)	2
9	Розгортання та підтримка (Render/Vercel)	Інженер (I2)	8
10	Управління проєктом	Кер. проєкту (Pm)	24
<b>Разом</b>			<b>175</b>

Сумарний час виконання операцій технологічного процесу становить 175 годин, що не перевищує граничного обсягу робіт за календарним планом.

## 4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проведено аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки вебплатформи «TastyHub». Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та діяльності підприємства.

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн}} = T_c \cdot K_r \quad (4.1)$$

де  $T_c$  – тарифна ставка, грн. (приймаємо для керівника проєкту (Pm) – 450 грн./год., інженера (I2) – 272 грн./год., інженера (I1) – 113 грн./год., тестувальника – 100 грн./год.);  $K_r$  – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

керівника проєкту (Pm):  $Z_{\text{осн}} = 39 \cdot 450 = 17\,550$  грн.;

інженера (I2):  $Z_{\text{осн}} = 56 \cdot 272 = 15\,232$  грн.;

інженера (I1):  $Z_{\text{осн}} = 72 \cdot 113 = 8\,136$  грн.;

тестувальника:  $Z_{\text{осн}} = 8 \cdot 100 = 800$  грн.

Сумарна основна заробітна плата становить:

$$Z_{\text{осн}} = 17\,550 + 15\,232 + 8\,136 + 800 = 41\,718 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати:

$$Z_{\text{дод}} = Z_{\text{осн}} \cdot K_{\text{допл}} \quad (4.2)$$

де  $K_{\text{допл}}$  – коефіцієнт додаткових виплат працівникам (приймаємо 0,1).

Отже, додаткова заробітна плата за категоріями працівників становить:

керівника проєкту (Pm):  $Z_{\text{дод}} = 17\,550 \cdot 0,1 = 1\,755$  грн.;

інженера (I2):  $Z_{\text{дод}} = 15\,232 \cdot 0,1 = 1\,523$  грн.;

інженера (I1):  $Z_{\text{дод}} = 8\,136 \cdot 0,1 = 814$  грн.;

тестувальника:  $Z_{\text{дод}} = 800 \cdot 0,1 = 80$  грн.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

Загальна додаткова заробітна плата становить:

$$З_{\text{дод}} = 1\,755 + 1\,523 + 814 + 80 = 4\,172 \text{ грн.}$$

Звідси загальні витрати на оплату праці визначаються за формулою:

$$V_{\text{о.п}} = Z_{\text{осн}} + Z_{\text{дод}} \quad (4.3)$$

$$V_{\text{о.п}} = 41\,718 + 4\,172 = 45\,890 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22 %) визначається за формулою:

$$V_{\text{ЄСВ}} = V_{\text{о.п}} \cdot 0,22 \quad (4.4)$$

$$V_{\text{ЄСВ}} = 45\,890 \cdot 0,22 = 10\,096 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№	Категорія працівників	Тарифна ставка, грн.	К-сть годин	Основна ЗП, грн.	Додат. ЗП, грн.	ЄСВ, грн.	Всього, грн.
1	Кер. проєкту (Pm)	450	39	17 550	1 755	4 247	23 552
2	Інженер (I2)	272	56	15 232	1 523	3 686	20 441
3	Інженер (I1)	113	72	8 136	814	1 969	10 919
4	Тестувальник	100	8	800	80	194	1 074
<b>Разом</b>			<b>175</b>	<b>41 718</b>	<b>4 172</b>	<b>10 096</b>	<b>55 986</b>

Отже, загальні витрати на оплату праці з урахуванням ЄСВ становлять 55 986 грн.

#### 4.3 Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію одиниці

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				54

обладнання визначаються за формулою:

$$Z_b = W \cdot T \cdot S \quad (4.5)$$

де  $W$  – необхідна потужність, кВт;  $T$  – кількість годин роботи обладнання;  $S$  – вартість кіловат-години електроенергії (приймаємо 15,94 грн.).

У системі розробки використовується один персональний комп'ютер. Час роботи обладнання приймаємо рівним сумарній трудомісткості робіт (згідно з табл. 4.1), а споживану потужність комп'ютера – 0,82 кВт/год.

$$Z_{ек} = 0,82 \cdot 175 \cdot 15,94 = 2\,287 \text{ грн.}$$

Витрати на електроенергію становлять 2 287 грн.

#### 4.4 Розрахунок суми амортизації

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації. Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів, мінімально допустимий строк корисного використання яких становить 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = БВ \cdot H_A \cdot T / T_{міс} \quad (4.6)$$

де  $A$  – амортизаційні відрахування за період, грн.;  $БВ$  – балансова вартість основних фондів, грн.;  $H_A$  – місячна норма амортизації (4 %);  $T$  – час використання обладнання, год.;  $T_{міс}$  – місячний фонд робочого часу (150 год.).

Оскільки для розробки та тестування використовується один ПК вартістю 50 000 грн., сума амортизаційних відрахувань становитиме:

$$A = 50\,000 \cdot 0,04 \cdot 175 / 150 = 2\,333 \text{ грн.}$$

Сума амортизаційних відрахувань становить 2 333 грн.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

#### 4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства та створенням необхідних умов праці. Залежно від організаційно-правової форми діяльності господарюючого суб'єкта накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = V_{o.n} \cdot (0,2 \dots 0,6) \quad (4.7)$$

де  $H_B$  – накладні витрати (приймаємо коефіцієнт 0,4).

$$H_B = 45\,890 \cdot 0,4 = 18\,356 \text{ грн.}$$

Накладні витрати становлять 18 356 грн.

#### 4.6 Складання кошторису витрат та визначення собівартості

Для складання кошторису витрат та визначення собівартості результати проведених вище розрахунків зведемо у таблицю 4.3.

Таблиця 4.3 – Кошторис витрат на розробку вебплатформи «TastyHub»

№	Зміст витрат	Сума, грн.	% до загальної суми
1	Витрати на оплату праці (з ЄСВ)	55 986	71
2	Витрати на електроенергію	2 287	3
3	Амортизаційні відрахування	2 333	3
4	Накладні витрати	18 356	23
<b>Собівартість</b>		<b>78 962</b>	<b>100</b>

Собівартість ( $C_B$ ) НДР розраховуємо за формулою:

$$C_B = V_{o.n} + V_{ЄСВ} + Z_{ек} + A + H_B \quad (4.8)$$

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

$$C_B = 45\,890 + 10\,096 + 2\,287 + 2\,333 + 18\,356 = 78\,962 \text{ грн.}$$

Отже, собівартість розробки дорівнює 78 962 грн.

#### 4.7 Розрахунок ціни

Розрахунок ціни науково-дослідної роботи враховує різноманітні фактори, такі як рівень рентабельності, собівартість та податкова ставка. Ціну робіт можна

визначити за формулою:

$$Ц = C_B \cdot (1 + P_{рен}) \cdot (1 + ПДВ) \quad (4.9)$$

де  $C_B$  – собівартість;  $P_{рен}$  – рівень рентабельності (приймаємо 0,3); ПДВ – ставка податку на додану вартість (0,2).

$$Ц = 78\,962 \cdot (1 + 0,3) \cdot (1 + 0,2) = 123\,181 \text{ грн.}$$

Розрахункова ціна розробки становить 123 181 грн.

#### 4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене й повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу. Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

Очікуваний річний грошовий потік визначається як різниця між ціною та собівартістю розробки:

$$ГП = 123\,181 - 78\,962 = 44\,219 \text{ грн.}$$

Чиста теперішня вартість розраховується за формулою:

$$ЧТВ = -C_B + \sum ГП / (1 + i)^t \quad (4.10)$$

де  $C_B$  – собівартість розробки; ГП – грошовий потік за t-ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (приймаємо 0,1).

$$ЧТВ = -78\,962 + 44\,219/1,1 + 44\,219/1,21 + 44\,219/1,331 = 31\,028 \text{ грн.}$$

					2026.КВР.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

Оскільки  $ЧТВ \geq 0$ , проєкт може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{ок} = T_{пв} + НВ / ГП_p \quad (4.11)$$

де  $T_{пв}$  – період до повного відшкодування витрат, років;  $НВ$  – невідшкодовані витрати на початок року, грн.;  $ГП_p$  – грошовий потік за рік, грн.

$$T_{ок} = 2 + 2\,218 / 44\,219 = 2,1 \text{ р.}$$

Усі отримані дані зведемо у таблицю 4.4.

Таблиця 4.4 – Техніко-економічні показники

№ п/п	Показник	Значення
1	Собівартість, грн.	78 962
2	Плановий прибуток (грошовий потік), грн.	44 219
3	Ціна, грн.	123 181
4	Чиста теперішня вартість, грн.	31 028
5	Термін окупності, рік	2,1

Додатне значення чистої теперішньої вартості та термін окупності близько 2,1 року свідчать про фінансову ефективність проєкту та його здатність повернути вкладені кошти у прийнятний строк. Ознайомившись зі всіма показниками таблиці техніко-економічних показників, яка наведена у графічній частині 2026.КВР.122.421.13.00.00 ТБ, можна зробити висновок, що розробка вебплатформи «TastyHub» є доцільною з економічної точки зору.

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.КВР.122.421.13.00.00 ПЗ				58

## **5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ**

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Дотримання вимог охорони праці та пожежної безпеки є обов'язковою умовою організації будь-якого виробничого процесу, зокрема й процесу створення програмного забезпечення.

Розробка вебплатформи «TastyHub» виконується на комп'ютеризованому робочому місці із застосуванням персонального комп'ютера, монітора та периферійного обладнання, що працює під електричною напругою. Праця розробника пов'язана з тривалим перебуванням за відеотерміналом, дією електромагнітних полів, підвищеним зоровим і нервово-емоційним навантаженням, а також з потенційною небезпекою ураження електричним струмом і виникнення пожежі. Це зумовлює потребу дотримання вимог охорони праці й техніки безпеки як на етапі розробки, так і під час подальшої експлуатації програмного продукту. У межах цього розділу розглянуто два питання: порядок повідомлення про нещасні випадки, їх розслідування та облік, а також основні способи припинення горіння.

### **5.1 Повідомлення про нещасні випадки, їх розслідування та облік**

Нещасний випадок на виробництві – це обмежена в часі подія або раптовий вплив на працівника небезпечного виробничого фактора чи середовища, що сталися під час виконання трудових обов'язків і внаслідок яких заподіяно шкоду здоров'ю працівника або настала його смерть. Поряд із поняттям нещасного випадку розрізняють також гостре професійне захворювання (отруєння), що виникає внаслідок короткочасного впливу шкідливих чинників, та аварію – небезпечну подію техногенного характеру. Порядок повідомлення, розслідування

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

та обліку таких подій регламентується Законом України «Про охорону праці» [7] та Порядком розслідування та обліку нещасних випадків, професійних захворювань та аварій на виробництві, затвердженим постановою Кабінету Міністрів України від 17 квітня 2019 року № 337 (зі змінами) [5].

Нещасні випадки класифікують за кількома ознаками. За зв'язком з виробництвом їх поділяють на пов'язані з виробництвом, що беруться на облік і оформлюються актом за формою Н-1/П, та не пов'язані з виробництвом, що оформлюються актом за формою Н-1/НП. За ступенем тяжкості наслідків розрізняють випадки з легкими наслідками, з тяжкими наслідками та зі смертельними наслідками. Окремо виділяють груповий нещасний випадок, який стався одночасно з двома й більше працівниками незалежно від ступеня тяжкості отриманих ними травм. Правильна класифікація події визначає вид розслідування та перелік документів, що складаються за його результатами.

Про кожний нещасний випадок потерпілий або працівник, який його виявив, чи будь-який інший очевидець повинен негайно повідомити безпосереднього керівника робіт або іншу уповноважену особу підприємства та вжити заходів для надання потерпілому домедичної допомоги. Своєчасне повідомлення є першою і ключовою умовою належного розслідування, оскільки дає змогу зберегти обстановку на місці події та зафіксувати її обставини.

Роботодавець, одержавши повідомлення про нещасний випадок, зобов'язаний:

- організувати надання потерпілому домедичної допомоги, а в разі потреби – виклик екстреної медичної допомоги та доставлення його до закладу охорони здоров'я;
- повідомити про подію територіальний орган Державної служби з питань праці, робочий орган Фонду соціального страхування, первинну організацію профспілки, а в разі потреби – інші органи, визначені Порядком;
- зберегти до прибуття комісії обстановку на робочому місці й стан обладнання у такому вигляді, в якому вони були на момент події, якщо це не загрожує життю та здоров'ю інших працівників і не призведе до аварії;

									Арк.
									60
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

- за потреби припинити виконання робіт, що становлять небезпеку для життя і здоров'я працівників;
- не пізніше наступного робочого дня утворити своїм наказом комісію з розслідування нещасного випадку.

Комісія з розслідування утворюється у складі не менше трьох осіб. До її складу, як правило, входять керівник служби охорони праці підприємства, представник роботодавця, представник первинної організації профспілки, а в окремих випадках – представник робочого органу Фонду соціального страхування. До роботи комісії може залучатися сам потерпілий або уповноважена ним особа. Водночас до складу комісії не включають керівника робіт, який безпосередньо відповідав за стан охорони праці на ділянці, де стався нещасний випадок, що забезпечує об'єктивність розслідування.

Під час розслідування комісія проводить засідання, обстежує місце події та складає відповідні протоколи, виготовляє ескіз і за можливості фотографії місця нещасного випадку, опитує потерпілого та свідків, за потреби призначає необхідні експертизи, лабораторні дослідження й випробування. На підставі зібраних матеріалів комісія встановлює обставини та причини нещасного випадку, визначає, чи пов'язаний він з виробництвом, осіб, які допустили порушення вимог охорони праці, та розробляє заходи щодо запобігання подібним подіям. Звичайне розслідування нещасного випадку, що не підлягає спеціальному розслідуванню, проводиться протягом п'яти робочих днів з дня утворення комісії. За потреби проведення лабораторних досліджень, експертиз або випробувань цей строк може бути обґрунтовано продовжено [5].

За результатами розслідування комісія складає акт за формою Н-1 у необхідній кількості примірників: у разі визнання випадку пов'язаним з виробництвом – за формою Н-1/П, а якщо випадок не пов'язаний з виробництвом – за формою Н-1/НП. Акт затверджується роботодавцем у встановлений строк та надсилається потерпілому, а також органам і організаціям, визначеним Порядком. Затверджений акт є основним документом, що підтверджує факт і обставини нещасного випадку та є підставою для відшкодування шкоди потерпілому.

					2026.КВР.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

Спеціальному розслідуванню підлягають нещасні випадки зі смертельними наслідками, групові нещасні випадки за участю двох і більше осіб, випадки з тяжкими наслідками, а також випадки зникнення працівника під час виконання трудових обов'язків та гострі групові професійні захворювання (отруєння). Спеціальне розслідування призначається органом Державної служби з питань праці, проводиться спеціально утвореною комісією за участю представників відповідних органів і триває протягом п'ятнадцяти робочих днів з можливістю продовження для отримання висновків експертиз та інших матеріалів [5].

Облік нещасних випадків полягає у їх реєстрації у відповідному журналі підприємства та зберіганні матеріалів розслідування. Примірник акта за формою Н-1 разом з матеріалами розслідування підлягає зберіганню на підприємстві протягом сорока п'яти років. По закінченні періоду тимчасової непрацездатності потерпілого роботодавець складає та надсилає повідомлення про наслідки нещасного випадку за формою Н-2 організаціям і особам, яким надсилався акт. На основі облікових даних формується державна статистична звітність про стан виробничого травматизму.

Аналіз причин нещасних випадків є важливою складовою профілактичної роботи. Причини виробничого травматизму умовно поділяють на технічні (несправність обладнання, недосконалість технологічних процесів, відсутність або несправність захисних засобів), організаційні (порушення правил експлуатації, недоліки в навчанні та інструктажах, неналежний нагляд за виконанням робіт) та психофізіологічні (втома, неуважність, невідповідність психофізіологічних якостей працівника характеру роботи). Виявлення переважних причин дає змогу цілеспрямовано розробляти заходи щодо їх усунення.

Застосовується топографічний метод (нанесення місць травмування на план підприємства) та монографічний метод (детальне вивчення окремого випадку чи робочого місця). Систематичний аналіз цих показників і методів дає змогу виявляти найбільш небезпечні ділянки та операції й планувати заходи з охорони праці [4].

Стосовно умов розробки вебплатформи «TastyHub» імовірність важких

									Арк.
									62
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				

нешасних випадків є невисокою, проте робота за відеотерміналом потребує дотримання вимог щодо організації робочого місця, режимів праці та відпочинку, а також правил електробезпеки, оскільки основним джерелом небезпеки є електрообладнання. Своєчасне повідомлення про будь-яку небезпечну подію, її коректне розслідування та облік залишаються обов'язковими незалежно від характеру виробництва.

## 5.2 Основні способи припинення горіння

Горіння – це складна фізико-хімічна реакція окиснення, що супроводжується інтенсивним виділенням тепла та світла. Залежно від співвідношення горючої речовини й окисника розрізняють повне горіння (за достатньої кількості кисню) та неповне горіння (за його нестачі, з утворенням токсичних продуктів). До характерних форм процесу належать спалах, займання, самозаймання та вибух. Неконтрольоване горіння поза спеціальним вогнищем, що завдає матеріальної шкоди та створює загрозу життю людей, називають пожежею.

Для виникнення й підтримання горіння необхідна одночасна наявність трьох умов: горючої речовини, окисника (як правило, кисню повітря) та джерела запалювання, що забезпечує необхідну для початку реакції енергію. Сукупність цих умов утворює так званий «трикутник горіння», а з урахуванням ланцюгового характеру реакції – «тетраedr горіння». Припинення горіння досягається усуненням однієї або кількох із цих умов чи припиненням ланцюгової реакції окиснення [4].

Виходячи з цього, розрізняють такі основні способи припинення горіння:

- охолодження зони горіння або речовин, що горять, нижче температури їх займання – досягається подаванням води чи водних розчинів, які поглинають значну кількість тепла й знижують температуру пального;
- ізоляція осередку горіння від доступу окисника – перекриття надходження повітря до зони горіння за допомогою повітряно-механічної піни, негорючих покривал, шару порошку, піску або землі;

					2026.КВР.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

- розбавлення (зниження концентрації) реагуючих речовин – подавання у зону горіння інертних газів (вуглекислого газу, азоту, водяної пари), що зменшує вміст кисню до рівня, за якого горіння стає неможливим;
- хімічне гальмування реакції горіння – введення речовин-інгібіторів (вогнегасних порошків, галогеновуглеводнів), які гальмують ланцюгову реакцію окиснення на молекулярному рівні;
- механічний зрив полум'я – дія на зону горіння сильного струменя води чи газу або ударної (вибухової) хвилі.

Перелічені способи реалізуються через застосування вогнегасних речовин, кожна з яких діє переважно за одним або кількома механізмами. Вода є найпоширенішою вогнегасною речовиною завдяки високій теплоємності та доступності; вона діє переважно охолоджувально й може подаватися компактним або розпиленним струменем. Повітряно-механічна піна ізолює поверхню пального від кисню та частково охолоджує її. Інертні гази й водяна пара знижують концентрацію кисню в зоні горіння. Вогнегасні порошки та галогеновмісні сполуки гальмують хімічну реакцію горіння. Аерозольні склади поєднують інгібувальну й розбавлювальну дію. Вибір конкретної речовини залежить від властивостей пального та класу пожежі.

Залежно від виду горючих речовин і матеріалів пожежі поділяють на класи, що визначає допустимі засоби гасіння:

- клас А – горіння твердих речовин переважно органічного походження (деревина, папір, тканини);
- клас В – горіння рідких речовин (нафтопродукти, розчинники, мастила);
- клас С – горіння газоподібних речовин (природний газ, водень, пропан);
- клас D – горіння металів та їх сплавів (магній, алюміній);
- клас Е – горіння електроустановок, що перебувають під напругою;
- клас F – горіння жирів та олій, зокрема на кухонному обладнанні.

Така класифікація дає змогу правильно добирати вогнегасні речовини,

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

оскільки застосування невідповідного засобу може бути неефективним або навіть небезпечним [2].

До первинних засобів пожежогасіння належать вогнегасники, пожежні крани, ящики з піском, протипожежні покривала та інший інвентар. Найпоширенішими є водяні та водопінні вогнегасники, що застосовуються переважно для пожеж класів А та В; порошкові вогнегасники, які є універсальними й придатні для більшості класів пожеж, у тому числі електроустановок; та вуглекислотні вогнегасники, що ефективні для гасіння електрообладнання й цінних матеріалів, оскільки не залишають слідів. Кожен вогнегасник має маркування із зазначенням типу, класів пожеж, для яких він призначений, і порядку приведення в дію, з яким користувач повинен бути ознайомлений заздалегідь.

Для захисту приміщень із підвищеною пожежною небезпекою застосовують також стаціонарні системи пожежогасіння – спринклерні та дренчерні водяні установки, газові й порошкові установки, що спрацьовують автоматично, а також системи пожежної сигналізації та оповіщення, які своєчасно виявляють загоряння й попереджають людей. У серверних приміщеннях і центрах обробки даних перевагу надають автоматичним газовим установкам, що не пошкоджують електронне обладнання.

Оскільки розробка та експлуатація вебплатформи «TastyHub» пов'язані з використанням персональних комп'ютерів і серверного обладнання, що перебуває під електричною напругою, особливу увагу слід приділити пожежам електроустановок (клас пожежі Е). Застосування води та піни для гасіння такого обладнання є небезпечним через їх електропровідність і ризик ураження струмом. Тому для приміщень з обчислювальною технікою рекомендовано використовувати вуглекислотні та порошкові вогнегасники, які не проводять електричний струм і не пошкоджують електронні компоненти. Питання пожежної безпеки регламентуються Правилами пожежної безпеки в Україні [6] та відповідними державними стандартами.

Запобігання пожежам забезпечується комплексом організаційних і технічних заходів: проведенням протипожежних інструктажів, дотриманням правил

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				65

експлуатації електрообладнання, своєчасним технічним обслуговуванням техніки, заборонаю використання несправних електроприладів, підтриманням вільними шляхів евакуації та укомплектуванням приміщень первинними засобами пожежогасіння. Дотримання цих заходів істотно знижує ймовірність виникнення й поширення пожежі на робочому місці розробника.

Отже, дотримання встановленого порядку повідомлення, розслідування й обліку нещасних випадків, а також знання основних способів припинення горіння та правильний добір засобів пожежогасіння забезпечують безпечні умови праці під час розробки й експлуатації вебплатформи «TastyHub» та зменшують ризик виробничого травматизму й пожеж.

					2026.КВР.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи розроблено вебплатформу для систематизації кулінарних рецептів «TastyHub». Усі вимоги технічного завдання реалізовано в повному обсязі. Проведений огляд існуючих кулінарних сервісів засвідчив, що на українському ринку відсутнє рішення, яке одночасно поєднує україномовний інтерфейс, можливість публікації користувацького контенту, сучасний технологічний стек та відкриту клієнт-серверну архітектуру.

Спроектовано та реалізовано клієнт-серверну архітектуру: серверна частина мовою Python із мікрофреймворком Flask формує REST API, клієнтська частина на бібліотеці React незалежно споживає ці дані, а зберігання інформації забезпечує база даних PostgreSQL. Реалізовано реєстрацію та авторизацію з хешуванням паролів і JWT-токенами, повний цикл роботи з рецептами, пошук і фільтрацію, список улюблених, коментарі, оцінювання, список покупок та адміністративну панель для модерації контенту.

Працездатність системи підтверджено функціональним тестуванням. Платформу розгорнуто у хмарному середовищі: серверну частину – на платформі Render, клієнтську – на Vercel, що забезпечило її доступність у мережі Інтернет.

За результатами економічних розрахунків собівартість розробки становить 78 962 грн, орієнтовна ціна продукту – 123 181 грн, а термін окупності – близько 2,1 року, що свідчить про економічну доцільність проекту. У розділі охорони праці розглянуто порядок повідомлення, розслідування та обліку нещасних випадків і основні способи припинення горіння, що забезпечують безпечні умови праці.

Усі етапи проектування, розробки, тестування та оформлення роботи виконано самостійно. Напрямами подальшого розвитку є створення мобільного застосунку на основі наявного REST API, упровадження рендерингу на боці сервера для пошукової оптимізації, підключення зовнішнього об'єктного сховища для зберігання зображень та розширення соціальних можливостей спільноти.

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

## ПЕРЕЛІК ПОСИЛАНЬ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ : ДП «УкрНДНЦ», 2016. 26 с.
2. ДСТУ EN 2:2014. Класифікація пожеж. Київ : ДП «УкрНДНЦ», 2014. 6 с.
3. Марціяш Г.Я., Слободян Р.О. Методичні вказівки до виконання кваліфікаційної роботи для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки». Тернопіль : ВСП «ТФК ТНТУ ім. І.Пулюя», 2026. 48 с.
4. Охорона праці та цивільний захист : підручник / О. Г. Левченко, О. І. Полукаров, В. В. Зацарний та ін. ; за ред. О. Г. Левченко. Київ : КПІ ім. Ігоря Сікорського, 2019. 420 с.
5. Про затвердження Порядку розслідування та обліку нещасних випадків, професійних захворювань та аварій на виробництві : постанова Кабінету Міністрів України від 17.04.2019 № 337. URL: <https://zakon.rada.gov.ua/laws/show/337-2019-п> (дата звернення: 20.05.2026).
6. Про затвердження Правил пожежної безпеки в Україні : наказ Міністерства внутрішніх справ України від 30.12.2014 № 1417. URL: <https://zakon.rada.gov.ua/laws/show/z0252-15> (дата звернення: 20.05.2026).
7. Про охорону праці : Закон України від 14.10.1992 № 2694-ХІІ. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 20.05.2026).
8. Allrecipes : вебсайт. URL: <https://www.allrecipes.com> (дата звернення: 10.05.2026).
9. BBC Good Food : вебсайт. URL: <https://www.bbcgoodfood.com> (дата звернення: 10.05.2026).
10. Flask Documentation. Deploying to Production : вебсайт. URL: <https://flask.palletsprojects.com/en/3.0.x/deploying> (дата звернення: 11.05.2026).
11. Flask Documentation. Foreword : вебсайт. URL: <https://flask.palletsprojects.com/en/3.0.x/foreword> (дата звернення: 11.05.2026).

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

12. Flask-Bcrypt Documentation : вебсайт. URL: <https://flask-bcrypt.readthedocs.io/en/1.0.1> (дата звернення: 13.05.2026).
13. Flask-SQLAlchemy Documentation : вебсайт. URL: <https://flask-sqlalchemy.palletsprojects.com/en/3.1.x> (дата звернення: 13.05.2026).
14. Google Developers. Web Vitals : вебсайт. URL: <https://web.dev/vitals> (дата звернення: 13.05.2026).
15. IEEE 12207-2008. Systems and Software Engineering – Software Life Cycle Processes : вебсайт. URL: <https://ieeexplore.ieee.org/document/4475906> (дата звернення: 15.05.2026).
16. IEEE 830-1998. IEEE Recommended Practice for Software Requirements Specifications : вебсайт. URL: <https://ieeexplore.ieee.org/document/720574> (дата звернення: 12.05.2026).
17. MDN Web Docs. Browser compatibility : вебсайт. URL: [https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5/HTML5\\_element\\_list](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5/HTML5_element_list) (дата звернення: 11.05.2026).
18. Nielsen J. 10 Usability Heuristics for User Interface Design. Nielsen Norman Group : вебсайт. URL: <https://www.nngroup.com/articles/ten-usability-heuristics> (дата звернення: 13.05.2026).
19. Postman Documentation. Testing APIs : вебсайт. URL: <https://learning.postman.com/docs/writing-scripts/test-scripts> (дата звернення: 14.05.2026).
20. React Documentation. Main Concepts : вебсайт. URL: <https://react.dev/learn> (дата звернення: 12.05.2026).
21. Render Documentation. Free Plan Overview : вебсайт. URL: <https://render.com/docs/free> (дата звернення: 14.05.2026).
22. Smachno.ua : вебсайт. URL: <https://smachno.ua> (дата звернення: 10.05.2026).
23. Statista. Share of mobile traffic in food and recipe websites worldwide 2024 : вебсайт. URL: <https://www.statista.com> (дата звернення: 12.05.2026).

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

24. Vercel Documentation. Deploying React Applications : вебсайт. URL:  
<https://vercel.com/docs/frameworks/react> (дата звернення: 14.05.2026).

					2026.КВР.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

## ДОДАТКИ

### Додаток А. Лістинг моделей бази даних.

```
class User(db.Model):
    __tablename__ = "users"

    id: Mapped[int] = mapped_column(Integer, primary_key=True)
    username: Mapped[str] = mapped_column(String(50), unique=True,
index=True, nullable=False)
    // ... display_name, email, password_hash, bio, avatar_url ...
    is_admin: Mapped[bool] = mapped_column(Boolean, nullable=False,
server_default="false")
    is_private: Mapped[bool] = mapped_column(Boolean, nullable=False,
server_default="false")
    email_verified: Mapped[bool] = mapped_column(Boolean, nullable=False,
server_default="false")

    recipes: Mapped[list["Recipe"]] = relationship(back_populates="owner",
cascade="all, delete-orphan")
    comments: Mapped[list["Comment"]] =
relationship(back_populates="author", cascade="all, delete-orphan")
    ratings: Mapped[list["Rating"]] = relationship(back_populates="user",
cascade="all, delete-orphan")
    shopping_list_entries: Mapped[list["ShoppingListRecipe"]] =
relationship(
    back_populates="user", cascade="all, delete-orphan"
)
class Recipe(db.Model):
    __tablename__ = "recipes"
    __table_args__ = (UniqueConstraint("owner_id", "title",
name="uq_recipe_owner_title"),)

    id: Mapped[int] = mapped_column(Integer, primary_key=True)
```

									Арк.
Зм.	Арк.	№ докум.	Підпис	Дата	2026.KBP.122.421.13.00.00 ПЗ				71

```

    title: Mapped[str] = mapped_column(String(200), nullable=False,
index=True)
    // ... description, instructions, cooking_time, image_url ...

    owner_id: Mapped[int] = mapped_column(ForeignKey("users.id",
ondelete="CASCADE"), nullable=False, index=True)
    category_id: Mapped[int] = mapped_column(
        ForeignKey("categories.id", ondelete="RESTRICT"), nullable=False,
index=True
    )

    owner: Mapped[User] = relationship(back_populates="recipes")
    category: Mapped[Category] = relationship(back_populates="recipes")
    ingredients: Mapped[list["RecipeIngredient"]] = relationship(
        back_populates="recipe", cascade="all", delete-orphan",
order_by="RecipeIngredient.position"
    )
    comments: Mapped[list["Comment"]] =
relationship(back_populates="recipe", cascade="all, delete-orphan")
    ratings: Mapped[list["Rating"]] = relationship(back_populates="recipe",
cascade="all, delete-orphan")
class Rating(db.Model):
    __tablename__ = "ratings"
    __table_args__ = (UniqueConstraint("user_id", "recipe_id",
name="uq_rating_user_recipe"),)
    // ... value, user_id FK CASCADE, recipe_id FK CASCADE ...

class SavedRecipe(db.Model):
    __tablename__ = "saved_recipes"
    __table_args__ = (UniqueConstraint("user_id", "recipe_id",
name="uq_saved_user_recipe"),)
    // ... user_id, recipe_id, created_at ...

class ShoppingListRecipe(db.Model):

```

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

```
__tablename__ = "shopping_list_recipes"
__table_args__ = (UniqueConstraint("user_id", "recipe_id",
name="uq_shopping_user_recipe"),)
// ... user_id, recipe_id, created_at ...
```

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

## Додаток Б. Лістинг модуля маршрутів серверного API.

```
@recipes_bp.get("")
def list_recipes():
    q = (request.args.get("q") or "").strip()
    ingredient = (request.args.get("ingredient") or "").strip()
    category_id = _int_query("category_id", None)
    cooking_time_max = _int_query("cooking_time_max", None)
    cooking_time_min = _int_query("cooking_time_min", None)
    sort = (request.args.get("sort") or "newest").strip().lower()

    query = Recipe.query
    if q:
        query = query.filter(Recipe.title.ilike(f"%{q}%"))
    if ingredient:
        query = query.join(RecipeIngredient).filter(RecipeIngredient.name.ilike(f"%{ingredient}%")).distinct()
    if category_id is not None:
        query = query.filter(Recipe.category_id == category_id)
    // ... фільтри за часом, сортування за рейтингом/популярністю ...

    pagination, page, per_page = _paginate(query)
    items = recipes_to_dicts_with_ratings(pagination.items,
current_user_id=current_user_id)
    return jsonify({"items": items, "page": page, "per_page": per_page,
"total": pagination.total, "pages": pagination.pages})
@recipes_bp.post("")
@jwt_required()
def create_recipe():
    data = request.get_json(silent=True) or {}
    required = ["title", "description", "ingredients", "instructions",
"cooking_time", "category_id"]
```

```

// ... валідація полів ...

owner_id = int(get_jwt_identity())
recipe = Recipe(
    title=title,
    description=description,
instructions=instructions,
    cooking_time=cooking_time, image_url=image_url,
    owner_id=owner_id, category_id=category_id,
)
_apply_ingredients(recipe, ingredient_items)
db.session.add(recipe)
db.session.commit()
return jsonify({"message": "Рецепт успішно створено", "recipe":
recipe_to_dict_with_ratings(recipe, ...)}), 201
def _require_owner(recipe: Recipe, user_id: int) -> None:
    if recipe.owner_id != user_id:
        raise APIError("Недостатньо прав для цієї дії", 403)

@recipes_bp.put("/<int:recipe_id>")
@jwt_required()
def update_recipe(recipe_id: int):
    // ... _require_owner(recipe, user_id) ...
    // ... оновлення полів і інгредієнтів ...
    db.session.commit()
    return jsonify({"message": "Рецепт успішно оновлено", "recipe":
...})

@recipes_bp.delete("/<int:recipe_id>")
@jwt_required()
def delete_recipe(recipe_id: int):
    current_user = get_current_user()
    if not can_manage_recipe(current_user=current_user,
recipe_owner_id=recipe.owner_id):
        raise APIError("Недостатньо прав для цієї дії", 403)

```

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

```

    db.session.delete(recipe)
    db.session.commit()
    return jsonify({"message": "Рецепт успішно видалено"})
@recipes_bp.put("/<int:recipe_id>/rating")
@jwt_required()
def upsert_rating(recipe_id: int):
    user_id = int(get_jwt_identity())
    if int(recipe.owner_id) == user_id:
        raise APIError("Не можна оцінювати власний рецепт", 409)

    value = _parse_rating_value(request.get_json(silent=True) or {})
    rating = Rating.query.filter_by(user_id=user_id,
recipe_id=recipe_id).first()
    if rating:
        rating.value = value
    else:
        rating = Rating(value=value, user_id=user_id,
recipe_id=recipe_id)
        db.session.add(rating)
    db.session.commit()
    return jsonify({"message": "Оцінку збережено", "recipe":
recipe_to_dict_with_ratings(recipe, ...)})

```

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		76

## Додаток В. Лістинг головного компонента клієнтської частини.

```
function readFiltersFromParams(searchParams) {
  return {
    q: searchParams.get('q') || '',
    categoryId: searchParams.get('category_id') || '',
    timePreset: searchParams.get('time') || '',
    ingredient: searchParams.get('ingredient') || '',
    sort: searchParams.get('sort') || 'newest',
    page: /* ... */,
  };
}

function buildRecipesQuery({ q, categoryId, cookingTimeMin,
  cookingTimeMax, ingredient, sort, page }) {
  const params = new URLSearchParams();
  if (q.trim()) params.set('q', q.trim());
  if (categoryId) params.set('category_id', categoryId);
  // ... cooking_time_min/max, ingredient, sort, page, per_page ...
  return `?${params.toString}`;
}

useEffect(() => {
  async function loadCategories() {
    const cats = await apiFetch('/categories');
    setCategories(cats.items || []);
  }
  loadCategories();
}, []);

const recipesPath = useMemo(
  () => `/recipes${buildRecipesQuery({ q, categoryId, /* ... */ sort,
  page })}` ,
  [q, categoryId, timeQuery, ingredient, sort, page]
```

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		77

```

);

const fetchRecipes = useCallback(async () => {
  setLoading(true);
  try {
    const list = await apiFetch(recipesPath);
    setRecipes(list.items || []);
    setListMeta({ total: list.total ?? 0, pages: list.pages ?? 1 });
  } catch (e) {
    setError(e.message || 'Не вдалося завантажити рецепти');
  } finally {
    setLoading(false);
  }
}, [recipesPath]);

async function createRecipe(payload) {
  setBusy(true);
  try {
    let nextPayload = payload;
    if (payload.image_file) {
      const fd = new FormData();
      fd.append('file', payload.image_file);
      const uploaded = await apiFetchFormData('/uploads/recipe-image', { auth: true, formData: fd });
      nextPayload = { ...payload, image_url: uploaded.image_url };
    }
    const data = await apiFetch('/recipes', { method: 'POST', auth: true, body: nextPayload });
    setCreating(false);
    fetchRecipes();
  } catch (e) {
    setError(e.message || 'Не вдалося створити рецепт');
  } finally {
    setBusy(false);
  }
}

```

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		78

```

    }
    return (
      <div>
        <header className="catalogPageHeader">
          <h1 className="catalogPageTitle fontSerif">Каталог
рецептів</h1>
        </header>

        <CategoryScroller categories={categories}
categoryId={categoryId} onSelect={updateCategoryId} />

        <div className="catalogLayout">
          <RecipeFilters /* категорія, час, інгредієнт, сортування */ />
          <div className="catalogMain">
            <div className="catalogToolbar">
              {auth.isAuthenticated ? (
                <button onClick={() => setCreating(true)}>Створити
рецепт</button>
              ) : ( /* Увійти / Почати */ )}
            </div>
            <div className="gridCards">
              {recipes.map((r) => (
                <RecipeCard key={r.id} recipe={r}
onOpenDetails={setModalRecipeId} variant="catalog" />
              ))}
            </div>
            <Pagination page={page} pages={listMeta.pages}
onPageChange={updatePage} />
          </div>
        </div>
        { /* модалка RecipeForm для створення */ }
      </div>
    );
  }
}

```

					2026.KBP.122.421.13.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

## Додаток Г. Маршрути REST API.

Метод	Шлях	Доступ	Призначення
Автентифікація (auth)			
POST	/api/auth/register	–	Реєстрація та лист підтвердження
POST	/api/auth/verify-email	–	Підтвердження пошти, видача JWT
POST	/api/auth/resend-verification	–	Повторний лист підтвердження
POST	/api/auth/login	–	Вхід, видача JWT
GET	/api/auth/me	JWT	Поточний користувач
Користувачі (users)			
GET	/api/users/me	JWT	Свій профіль
PUT	/api/users/me	JWT	Оновлення профілю
GET	/api/users/me/saved	JWT	Улюблені рецепти
GET	/api/users/me/comments	JWT	Свої коментарі
GET	/api/users/:username	JWT*	Публічний профіль
GET	/api/users	JWT admin +	Список користувачів
DELETE	/api/users/:id	JWT admin +	Видалення користувача
Категорії (categories)			
GET	/api/categories	–	Список категорій
POST	/api/categories	JWT admin +	Створення категорії



Продовження таблиці

1	2	3	4
Завантаження файлів (uploads)			
POST	/api/uploads/avatar	JWT	Завантаження аватара
POST	/api/uploads/recipe-image	JWT	Завантаження фото рецепта
Адміністрування (admin)			
GET	/api/admin/comments	JWT admin	Коментарі для модерації

## Додаток Д. Перелік тестових сценаріїв.

№	Сценарій	Кроки та вхідні дані	Очікуваний результат	Статус
1	2	3	4	5
<b>Авторизація та доступ</b>				
1	Реєстрація користувача	Заповнити форму реєстрації коректними даними	Акаунт створено, надіслано лист підтвердження	Пройдено
2	Підтвердження email	Перейти за токеном із листа	Акаунт активовано, видано JWT, виконано вхід	Пройдено
3	Вхід із правильними даними	Ввести email і пароль	Успішний вхід, доступні розділи користувача	Пройдено
4	Вхід із неправильним паролем	Ввести хибний пароль	Відмова з повідомленням, вхід не виконано	Пройдено
5	Доступ без авторизації	Відкрити /favorites без входу	Перенаправлення на сторінку входу	Пройдено
<b>Рецепти</b>				
6	Створення рецепта з фото	Заповнити форму, додати інгредієнти та фото	Рецепт створено й відображається в каталозі	Пройдено
7	Редагування власного рецепта	Змінити дані свого рецепта	Зміни збережено й відображено	Пройдено
8	Видалення власного рецепта	Видалити свій рецепт	Рецепт зникає з каталогу	Пройдено

Продовження таблиці

1	2	3	4	5
9	Заборона редагування чужого	Іншим користувачем відкрити чужий рецепт; PUT напряму	Кнопки немає; запит повертає 403	Пройдено
10	Дубль назви рецепта	Створити рецепт із наявною назвою цього автора	Відмова через унікальність (owner_id, title)	Пройдено

Каталог, пошук і фільтрація

11	Пошук за назвою	Ввести запит у поле пошуку	Показано лише відповідні рецепти	Пройдено
12	Фільтр за категорією	Обрати категорію	Каталог оновлюється	Пройдено
13	Фільтр за інгредієнтом і часом	Задати інгредієнт і діапазон часу	Результати відповідають умовам	Пройдено
14	Сортування	Обрати варіант сортування	Порядок рецептів змінюється	Пройдено
15	Пагінація	Перейти на наступну сторінку	Завантажуються інші рецепти	Пройдено

Улюблені та список покупок

16	Додавання в улюблені	Натиснути «Додати в улюблені»	Рецепт з'являється на /favorites	Пройдено
17	Збереження на сервері	Додати в улюблені, перезайти / інший браузер	Список збережено незалежно від пристрою	Пройдено

Продовження таблиці

1	2	3	4	5
18	Список покупок	Додати рецепти; відкрити /shopping-list; очистити	Інгредієнти агрегуються; очищення працює	Пройдено
Рейтинг і коментарі				
19	Оцінювання чужого рецепта	Поставити оцінку 1–5	Середній рейтинг оновлюється	Пройдено
20	Заборона оцінки власного	Відкрити власний рецепт	Оцінювання недоступне	Пройдено
21	Коментарі	Додати, відредагувати, видалити свій коментар	Усі дії виконуються коректно	Пройдено
Адміністрування				
22	Керування категоріями	Додати й видалити категорію	Зміни застосовуються	Пройдено
23	Модерація контенту	Видалити чужий рецепт і коментар; редагувати чужий	Видалення працює; редагування недоступне	Пройдено
24	Видалення користувача	Адміном видалити користувача	Користувач і пов'язані записи видаляються	Пройдено
Валідація та безпека				
25	Порожні обов'язкові поля	Надіслати форму без назви	Відмова з повідомленням (клієнт і сервер)	Пройдено
26	Граничні значення	Від'ємний час; оцінка поза 1–5	Сервер відхиляє запит (400)	Пройдено

Продовження таблиці

1	2	3	4	5
27	Доступ без токена	Викликати захищений ендпоінт без JWT	Сервер повертає 401	Пройдено
Адаптивність і сумісність				
28	Мобільний вигляд	Відкрити на ширині 320 px	Верстка коректна, меню працює	Пройдено
29	Кросбраузерність	Відкрити в Chrome, Firefox, Edge	Відображення й робота однакові	Пройдено