Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Створення інтерактивного вебсайту «TasteCrafters»

Виконав: студент IV курсу, групи СНз-41
спеціальності 122 Комп’ютерні науки
(шифр і назва спеціальності)

(підпис) Ваврик М. М.
(прізвище та ініціали)

Керівник (підпис) Гром’як Р. С.
(прізвище та ініціали)

Нормоконтроль (підпис) Шимчук Г.В.
(прізвище та ініціали)

Завідувач кафедри (підпис) Боднарчук І.О.
(прізвище та ініціали)

Рецензент (підпис) Тимощук Д.І.
(прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«_ __ _» _червня_ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Ваврику Максиму Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Створення інтерактивного вебсайту «TasteCrafters»

Керівник роботи Гром'як Роман Сильвестрович, к.ф.-м.н, доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*14* » *травня* 2026 року № *4/9-238*

2. Термін подання студентом завершеної роботи 17 червня 2026р.

3. Вихідні дані до роботи Інтернет джерела про технології розробки сайту Flavor Alchemy засобами HTML5, CSS3, PHP, JavaScript та MySQL.

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. Розділ 1. Теоретичне обґрунтування та визначення вимог до створення вебсайту. 1.1 Дослідження предметної області. 1.2 Огляд і порівняння існуючих вебрішень. 1.3 Формування функціональних і нефункціональних вимог до вебсайту. 1.4 Визначення основних категорій користувачів та особливостей їхньої взаємодії із системою. 1.5 Опис сценаріїв використання платформи для обміну кулінарними рецептами. 1.6 Аналіз архітектурних і технологічних підходів до створення вебсайтів. 1.7 Обґрунтування вибору технологічного стеку та засобів розроблення. 1.8 Вибір програмного середовища для реалізації вебсайту. Розділ 2. Проєктування структури та технічної основи вебсайту. 2.1 Побудова логічної та технічної архітектури вебсайту. 2.2 Розроблення ієрархічної схеми та навігаційної моделі вебсайту. 2.3 Створення логічної та фізичної моделі бази даних вебсайту. 2.4 Опис користувацьких сценаріїв і модульної організації системи. 2.5 Побудова діаграми класів вебсайту. 2.6 Формування файлової структури та ієрархії каталогів проєкту. Розділ 3. Перевірка якості, валідація та тестування вебсайту. 3.1 Обґрунтування вибору хостингового середовища для розміщення вебсайту. 3.2 Проведення валідації та тестування вебсайту. 3.3 Перевірка функціональної роботи та супровід вебсайту в умовах експлуатації. Розділ 4. Безпека життєдіяльності та охорона праці. Висновки. Перелік джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Титулка. 2. Актуальність роботи. 3. Мета і завдання роботи. 4. Практичне значення одержаних результатів. 5. TasteCrafters. 6. Порівняльний аналіз існуючих рішень. 7. Вимоги до вебсайту «TasteCrafters». 8. Архітектура вебсайту «TasteCrafters». 9. Розробка бази даних вебсайту «TasteCrafters». 10. Концептуальна модель БД вебсайту «TasteCrafters» 11. Діаграма станів публікації рецепту на вебсайті «TasteCrafters» Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик Олег Ярославович, к.т.н., доцент кафедри МТ		

7. Дата видачі завдання 26 січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	27.01.2026-16.02.2026	*Виконано*
2.	Опрацювання джерел по темі кваліфікаційної роботи	17.02.2026-10.05.2026	*Виконано*
3.	Виконання дослідження щодо розробки сайту	11.05.2026-17.05.2026	*Виконано*
4.	Оформлення розділу “Теоретичне обґрунтування та	18.05.2026-24.05.2026	*Виконано*
	визначення вимог до створення вебсайту”		
5.	Оформлення розділу “Проєктування структури та	25.05.2026-31.05.2026	*Виконано*
	технічної основи вебсайту”		
6.	Оформлення розділу “Перевірка якості, валідація та	01.06.2026-08.06.2026	*Виконано*
	тестування вебсайту”		
7.	Виконання завдання до підрозділу “Безпека життєдіяльності”	01.06.2026-07.06.2026	*Виконано*
8.	Виконання завдання до підрозділу “Основи охорони		*Виконано*
	праці”	07.06.2026-8.06.2026	
9.	Оформлення кваліфікаційної роботи	08.06.2026	*Виконано*
10.	Нормоконтроль	9.06.2026	*Виконано*
11.	Перевірка на плагіат	10.06.2026	*Виконано*
12.	Попередній захист кваліфікаційної роботи	11.06.2026	*Виконано*
13.	Захист кваліфікаційної роботи	18.06.2026	

Студент _____ Ваврик М. М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Гром’як Р. С.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Створення інтерактивного вебсайту «TasteCrafters» // Кваліфікаційна робота освітнього рівня “Бакалавр” // Ваврик Максим Михайлович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедри комп’ютерних наук, група СНз-41 // Тернопіль, 2026 // С. – 84, рис. – 32, табл. – 7, слайди – 15, додат. – 8, бібліогр. – 43.

Ключові слова: кулінарія, обмін рецептами, веб-портал, адаптивний дизайн, кросплатформність, React, Node.js, PostgreSQL, тестування, CI/CD-деплоймент

Кваліфікаційна робота присвячена розробленню та впровадженню вебсайту для обміну кулінарними рецептами. Основна ідея проєкту полягає у створенні зручної онлайн-платформи, адаптованої до мобільних пристроїв, де користувачі можуть публікувати власні рецепти, переглядати матеріали інших авторів, залишати коментарі, оцінювати страви та зберігати вподобані рецепти в особистій добірці. Такий ресурс орієнтований на широку аудиторію: від початківців, які лише навчаються готувати, до досвідчених кулінарів, зацікавлених у пошуку нових ідей.

У першому розділі розглянуто предметну область і проаналізовано сучасні тенденції розвитку кулінарних онлайн-сервісів та фуд-блогінгу. Також досліджено наявні вебрішення, визначено їхні переваги й недоліки. На основі проведеного аналізу сформовано функціональні та нефункціональні вимоги до майбутнього вебсайту, описано основні сценарії взаємодії користувачів із системою та визначено ключові ролі: гість, зареєстрований користувач, автор рецепта й адміністратор. Окрему увагу приділено вибору технологічного стеку, зокрема React, Node.js і PostgreSQL, які забезпечують гнучкість розробки, масштабованість і надійну роботу з даними.

Другий розділ присвячено проєктуванню вебсайту. У ньому описано клієнт-серверну архітектуру, логічну й фізичну структуру бази даних, а також основні сутності системи: користувачів, рецепти, інгредієнти, коментарі та оцінки. Наведено діаграми, що відображають взаємозв'язки між компонентами системи, принципи роботи фронтенду й бекенду, а також логіку обміну даними. Додатково розглянуто структуру каталогів проєкту, маршрутизацію запитів і підхід до керування станом клієнтської частини.

У третьому розділі проведено перевірку працездатності розробленого вебсайту. Здійснено валідацію HTML- і CSS-коду, тестування API, перевірку React-компонентів, а також ручне тестування в різних браузерах і на екранах різного розміру. Це дало змогу підтвердити адаптивність, кросбраузерність і стабільність основних функцій платформи. Також обґрунтовано вибір хостингу, описано процес розгортання проєкту та наведено порядок використання ключових сторінок: від реєстрації користувача до створення й публікації рецепта.

Четвертий розділ охоплює питання безпеки життєдіяльності та охорони праці. У ньому подано порядок надання першої допомоги в разі ураження електричним струмом, а також рекомендації щодо безпечної організації робочого місця за комп'ютером. Розглянуто вимоги до ергономіки, освітлення, режиму праці й відпочинку, пожежної безпеки та зменшення шкідливого впливу під час тривалої роботи з комп'ютерною технікою.

ANNOTATION

Development of the «TasteCrafters» Interactive Website // Qualification work of the educational, level «Bachelor» // Vavryk Maksym Mykhailovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNz-41 // Ternopil, 2026 // P. – 84, fig. – 32, tabl. – 7, slides – 15, annexes. – 8, references – 43.

Keywords: cuisine, recipe sharing, web portal, responsive design, cross-platform compatibility, React, Node.js, PostgreSQL, testing, CI/CD deployment

The qualification work is devoted to the development and implementation of a website for sharing culinary recipes. The main idea of the project is to create a convenient online platform adapted for mobile devices, where users can publish their own recipes, view materials created by other authors, leave comments, rate dishes, and save favorite recipes to a personal collection. This resource is intended for a broad audience: from beginners who are only learning to cook to experienced culinary enthusiasts looking for new ideas.

The first chapter examines the subject area and analyzes current trends in the development of online culinary services and food blogging. Existing web solutions are also reviewed, with their advantages and disadvantages identified. Based on the conducted analysis, functional and non-functional requirements for the future website are formulated, the main scenarios of user interaction with the system are described, and the key roles are defined: guest, registered user, recipe author, and administrator. Particular attention is paid to the choice of the technology stack, namely React, Node.js, and PostgreSQL, which provide development flexibility, scalability, and reliable data processing.

The second chapter is devoted to website design. It describes the client-server architecture, the logical and physical structure of the database, as well as the main system entities: users, recipes, ingredients, comments, and ratings. The chapter presents diagrams that illustrate the relationships between system components, the

principles of front-end and back-end operation, and the logic of data exchange. In addition, the project directory structure, request routing, and the approach to managing the state of the client side are considered.

The third chapter presents the verification of the developed website's functionality. HTML and CSS code validation, API testing, React component testing, and manual testing in different browsers and on screens of various sizes were carried out. This made it possible to confirm the adaptability, cross-browser compatibility, and stability of the platform's main functions. The choice of hosting is also justified, the deployment process is described, and the procedure for using the key pages is presented: from user registration to creating and publishing a recipe.

The fourth chapter covers issues of life safety and occupational health. It presents the procedure for providing first aid in the event of electric shock, as well as recommendations for the safe organization of a computer workstation. The chapter considers requirements for ergonomics, lighting, work and rest schedules, fire safety, and the reduction of harmful effects during prolonged work with computer equipment.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CSS (англ. Cascading Style Sheets) – каскадні таблиці стилів.

HTML (англ. HyperText Markup Language) – мова розмітки гіпертекстових документів.

ID (англ. Identity Document) – унікальний номер, який використовується для ідентифікації.

JS (англ. JavaScript) – мова програмування для створення інтерактивності та веб-сайтах.

MySQL (англ. My Structured Query Language) – вільна система керування реляційними базами даних.

PHP (англ. Hypertext Preprocessor, Personal Hone Page) – гіпертекстовий препроцесор, скриптова мова програмування для генерування веб-сторінок на стороні веб-сервера.

W3C (англ. World Wide Web Consortium) – організація, яка створила стандарти веб-розробки й надає онлайн валідатор для перевірки правильності коду.

БД – база даних.

ПК – персональний комп’ютер.

ЗМІСТ

ВСТУП

Актуальність теми. У сучасних умовах інтенсивної цифровізації повсякденного життя кулінарна діяльність дедалі більше виходить за межі суто побутової практики та набуває ознак відкритого інформаційно-комунікаційного середовища. Приготування їжі, пошук рецептів, обмін кулінарним досвідом і формування харчових уподобань сьогодні значною мірою відбуваються за допомогою цифрових сервісів, соціальних мереж і спеціалізованих вебплатформ. Такі ресурси не лише спрощують доступ до великої кількості рецептів, а й сприяють популяризації здорового харчування, розвитку фуд-блогінгу, підтримці локальних гастрономічних традицій та об’єднанню користувачів за спільними інтересами.

Особливої актуальності набувають онлайн-платформи, які забезпечують не пасивне споживання контенту, а активну участь користувачів у його створенні, обговоренні та оцінюванні. Сучасний користувач очікує від кулінарного вебсайту не лише наявності бази рецептів, а й зручного пошуку, персоналізованих добірок, можливості зберігати улюблені матеріали, залишати коментарі, оцінювати страви та взаємодіяти з іншими учасниками спільноти. Водночас значна частина наявних сервісів має низку обмежень: недостатній рівень персоналізації, перевантажений або застарілий інтерфейс, слабко реалізовані соціальні функції, недостатню адаптивність для мобільних пристроїв, а також технічні труднощі, пов’язані з масштабуванням, безпекою та підтримкою продуктивності.

У зв’язку з цим розроблення сучасного вебсайту для обміну кулінарними рецептами є актуальним завданням як з практичного, так і з науково-технічного погляду. Такий ресурс має поєднувати функціональність кулінарного каталогу, соціальної платформи та персонального інструмента для планування харчування. Крім того, важливим є використання сучасного технологічного стеку, який забезпечує швидкодію, надійне зберігання даних, можливість подальшого розширення функціоналу та коректну роботу на різних типах пристроїв.

Мета і задачі дослідження. Метою кваліфікаційної роботи є проєктування, розроблення, впровадження та експериментальна перевірка вебсайту для обміну кулінарними рецептами, який забезпечує зручну публікацію й пошук рецептів, підтримує соціальну взаємодію між користувачами та відповідає сучасним вимогам до безпеки, масштабованості, продуктивності й ергономіки користувацького інтерфейсу.

Для досягнення поставленої мети в межах роботи передбачено виконання таких завдань:

- Провести аналіз предметної області, зокрема дослідити сучасні тенденції розвитку кулінарних онлайн-платформ, фуд-блогінгу та цифрових сервісів для обміну рецептами, а також визначити основні потреби потенційних користувачів.
- Проаналізувати функціональні можливості конкурентних вебсервісів, виявити їхні переваги й недоліки, а також сформувати перелік вимог до майбутньої системи з урахуванням очікувань цільової аудиторії.
- Визначити функціональні вимоги до вебсайту, зокрема реєстрацію та авторизацію користувачів, створення й редагування рецептів, пошук, фільтрацію, коментування, оцінювання та додавання матеріалів до вибраного.
- Сформувати нефункціональні вимоги до системи, включаючи продуктивність, безпечне зберігання даних, захист користувацької інформації, адаптивність інтерфейсу, зручність використання та можливість подальшого масштабування.
- Обґрунтувати вибір технологічного стеку для реалізації вебсайту, зокрема використання React для клієнтської частини, Node.js для серверної логіки та PostgreSQL для організації реляційної бази даних.
- Спроєктувати архітектуру програмної системи, розробити багаторівневу клієнт-серверну модель, визначити основні модулі вебсайту та описати логіку їхньої взаємодії.
- Побудувати логічну та фізичну структуру бази даних, визначити ключові сутності, зв’язки між ними, атрибути таблиць і принципи збереження інформації про користувачів, рецепти, коментарі, оцінки та вибрані матеріали.

- Підготувати необхідні UML-діаграми, зокрема діаграми класів, послідовностей, варіантів використання та розміщення, які відображають структуру, поведінку й особливості функціонування розроблюваної системи.
- Реалізувати програмний прототип вебплатформи з підтримкою основних функцій: реєстрації користувачів, авторизації, створення рецептів, перегляду кулінарного контенту, пошуку, фільтрації та соціальної взаємодії.
- Провести перевірку працездатності розробленого вебсайту за допомогою автоматизованого й ручного тестування, включаючи юніт-тести, інтеграційне тестування, перевірку кросбраузерності, адаптивності інтерфейсу та коректності взаємодії з базою даних.

Виконання зазначених завдань дає змогу комплексно вирішити проблему створення сучасного вебсайту для обміну кулінарними рецептами, оцінити ефективність обраних технологічних рішень і підтвердити доцільність використання запропонованої архітектури для подібних інформаційних систем.

Практичне значення одержаних результатів. Практична цінність результатів кваліфікаційної роботи полягає у створенні функціонального вебсайту, який може бути використаний як готове програмне рішення або як основа для подальшого розвитку кулінарної онлайн-платформи. Розроблений ресурс орієнтований на різні категорії користувачів: домашніх кулінарів, професійних кухарів, фуд-блогерів, прихильників здорового харчування, а також користувачів, які шукають зручний інструмент для систематизації власних рецептів і планування щоденного меню.

Платформа надає можливість швидко створювати, редагувати, публікувати та впорядковувати рецепти, супроводжувати їх описами, категоріями, інгредієнтами й додатковою інформацією. Завдяки цьому користувачі отримують інтуїтивний інструмент для збереження власного кулінарного досвіду та поширення його серед інших учасників спільноти. Наявність функцій коментування, оцінювання та додавання рецептів до вибраного сприяє активнішій взаємодії між користувачами, підвищує якість контенту та допомагає швидше знаходити найбільш релевантні й корисні рецепти.

Окреме практичне значення має те, що вебсайт може сприяти популяризації здорового харчування та збереженню локальних гастрономічних традицій. Користувачі можуть публікувати як сучасні рецепти, так і страви регіональної кухні, передаючи кулінарний досвід ширшій аудиторії. Крім того, система може бути корисною для формування тематичних добірок рецептів, створення кулінарних блогів, просування авторського контенту та організації спільнот за харчовими вподобаннями.

З технічного погляду розроблений програмний продукт демонструє практичне застосування сучасного JavaScript-стеку в реальному вебпроєкті. Використання React, Node.js і PostgreSQL дозволяє забезпечити гнучку структуру системи, швидку взаємодію клієнтської та серверної частин, надійне зберігання даних і можливість подальшого розширення функціоналу. Клієнт-серверна архітектура, модульний підхід до розроблення, використання тестування та можливість налаштування CI/CD-процесів роблять систему придатною не лише для практичного впровадження, а й для використання як навчального прикладу в межах дисциплін, пов’язаних із веброзробкою, базами даних, тестуванням програмного забезпечення та DevOps-практиками.

Результати роботи можуть бути адаптовані для різних бізнес-моделей і сценаріїв використання. Зокрема, розроблене рішення може стати основою для нішевих кулінарних блогів, корпоративних порталів харчових брендів, сервісів із планування раціону, маркетплейсів локальних виробників харчових продуктів або мобільних додатків, пов’язаних із кулінарією та здоровим способом життя. Завдяки відкритій і зрозумілій структурі програмного коду систему можна модифікувати, доповнювати новими модулями та інтегрувати з іншими сервісами.

Таким чином, одержані результати мають прикладну, освітню та потенційну комерційну цінність. Розроблений вебсайт може використовуватися як самостійний цифровий продукт, як основа для стартапу у сфері food-tech або як база для подальших досліджень, пов’язаних із персоналізованими рекомендаціями, соціальними мережами харчових уподобань, аналізом поведінки користувачів і автоматизованим формуванням кулінарного контенту.

РОЗДІЛ 1. ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ ТА ВИЗНАЧЕННЯ ВИМОГ ДО СТВОРЕННЯ ВЕБСАЙТУ

1.1 Дослідження предметної області

Кулінарні онлайн-спільноти вже тривалий час залишаються помітним і динамічним напрямом розвитку інтернет-середовища. Інтерес до якісного контенту про приготування їжі зростає разом із популярністю здорового харчування, локальних продуктів і бажанням користувачів робити щоденний раціон різноманітнішим. Сучасні сервіси з рецептами поступово набули ознак соціальних платформ: користувачі можуть оцінювати страви, залишати коментарі, зберігати улюблені рецепти та публікувати власні кулінарні напрацювання. Водночас значна частина таких ресурсів орієнтована переважно на англомовну аудиторію або конкретну кухню. Україномовних універсальних платформ, де рецепти зручно структуровані, а інтерфейс повноцінно адаптований до мобільних пристроїв, досі недостатньо.

Основними учасниками такої системи є автори рецептів, звичайні читачі, досвідчені кулінари та модератори. Авторам важливо мати простий механізм додавання матеріалів, можливість завантажувати фото чи відео, а також швидко формувати зручну для друку версію рецепта. Читачі очікують зрозумілої навігації, пошуку за інгредієнтами, часом приготування, калорійністю і наявності реальних відгуків. Досвідчені користувачі звертають увагу на теги для складніших кулінарних технік, наприклад ферментації або су-від, а також на можливість створювати власні добірки. Модераторам потрібні засоби перевірки текстів, виявлення дублікатів і контролю за дотриманням авторських прав.

Технологічні вимоги до вебсайту для обміну рецептами формуються під впливом кількох актуальних тенденцій. Насамперед зростання мобільного трафіку робить адаптивний дизайн обов’язковою умовою. Крім того, популярність коротких відео потребує підтримки мультимедійного контенту та інтеграції з хмарними сховищами. Також дедалі важливішими стають персоналізовані рекомендації: замість звичайного сортування за датою

користувачі очікують добірок, сформованих відповідно до їхніх переглядів, уподобань і харчових обмежень. Однак така функціональність потребує уважного ставлення до безпеки персональних даних і відповідності вимогам GDPR та українського законодавства.

Аналіз конкурентних продуктів дозволяє виокремити дві типові проблеми. Перша полягає в поверхневій локалізації: українська версія часто обмежується перекладом інтерфейсу без адаптації кулінарних мір, назв продуктів і локальних харчових звичок. Друга проблема — слабка соціальна взаємодія. Коментарі, чати або механізми обговорення нерідко реалізовані формально й не сприяють накопиченню практичного досвіду спільноти. Тому новий вебсайт має поєднувати гнучке керування контентом, якісну українську локалізацію та інструменти, що заохочують користувачів обмінюватися порадами, уточненнями й результатами приготування.

Отже, ринок вебсервісів для поширення кулінарних рецептів залишається відкритим для рішень, які поєднують адаптивність, мультимедійну подачу, соціальні функції та якісну локалізацію. Така платформа може бути корисною для широкої аудиторії — від студентів і молодих сімей до професійних кухарів — і водночас створити основу для подальшого розвитку персоналізованих гастрономічних рекомендацій.

1.2 Огляд і порівняння існуючих вебрішень

Світові кулінарні платформи давно вийшли за межі звичайних сайтів із рецептами й перетворилися на соціальні медіа-майданчики. Одним із найвідоміших прикладів є Allrecipes, де мільйони оцінок і відгуків формують своєрідний «колективний смак» аудиторії. На основі активності користувачів сервіс ранжує рецепти, показує популярні страви та підказує авторам, які матеріали варто доповнити або переосмислити. Платформа також мотивує користувачів додавати власні фото готових страв і формує тематичні добірки відповідно до сезонних харчових трендів.

Food52 демонструє інший підхід: він поєднує рецептурний контент із магазином посуду та товарів для дому. Регулярні конкурси, редакційні добірки й промоція авторських рецептів допомагають підтримувати інтерес спільноти та водночас монетизувати трафік. Мобільно орієнтований напрям добре представлений сервісом Tasty від BuzzFeed, який пропонує велику кількість відеорецептів, покроковий режим приготування та додаткові функції для зміни інгредієнтів чи калорійності. Останні оновлення також включають AI-асистента Botatouille та інтеграцію з онлайн-кошиком Walmart.

Персоналізація є ключовою перевагою Yummly. Сервіс використовує алгоритми машинного навчання для аналізу вподобань, фільтрації алергенів і формування списків покупок. Завдяки цьому користувач отримує більш релевантні рекомендації, а платформа підвищує рівень утримання аудиторії.

Український сегмент представлений переважно медійними та авторськими проєктами. Smachno.ua пропонує тематичні добірки до свят і меню вихідного дня, однак навігація там більше відповідає редакційній логіці, ніж індивідуальним запитам користувачів. Klopotenko.com спирається на особистий бренд шеф-кухаря та має нестандартні рубрики, проте не пропонує розвиненої системи взаємних оцінок, рецензій і стандартизованих кулінарних мір. Cookpad підтримує українську локалізацію і дає змогу публікувати власні рецепти з фото, однак пошук за складниками може працювати неточно через неоднозначність назв інгредієнтів і відсутність єдиного словника мір.

Порівняння наявних сервісів показує кілька спільних недоліків: обмежену багатомовність, недостатню підтримку інтерактивних форматів, слабку гейміфікацію та нерівномірну увагу до авторських прав на зображення й відео. Отже, актуальним залишається створення вебсайту для обміну рецептами, який поєднає українську локалізацію, рекомендаційні алгоритми, змістовну соціальну взаємодію та безпечну роботу з персональними даними.

Платформа «Пательня» має приємну кольорову гаму та достатньо зрозумілу навігаційну структуру. Головна сторінка відкривається добіркою популярних страв, а картки рецептів містять лічильники вподобань і переглядів, що стимулює користувача переглядати більше матеріалів. На сайті реалізовано

пошук за ключовими словами й базові фільтри за типом кухні, складністю та тривалістю приготування. Авторизовані користувачі можуть додавати власні рецепти через просту форму з полями для інгредієнтів, опису кроків і порад.

Водночас ресурс має низку недоліків, які помітно впливають на зручність користування. На окремих сторінках рецепти подані без фотографій (див. рис. 1.1), що є суттєвим мінусом, адже візуальна складова для кулінарного контенту має першочергове значення [1]. Ще одна проблема — подвійне головне меню: частина навігації розташована у шапці сайту, а частина — у футері. Через це користувачам смартфонів доводиться додатково прокручувати сторінку, щоб перейти до потрібного розділу. Така організація уповільнює взаємодію і може дезорієнтувати нових відвідувачів [1].

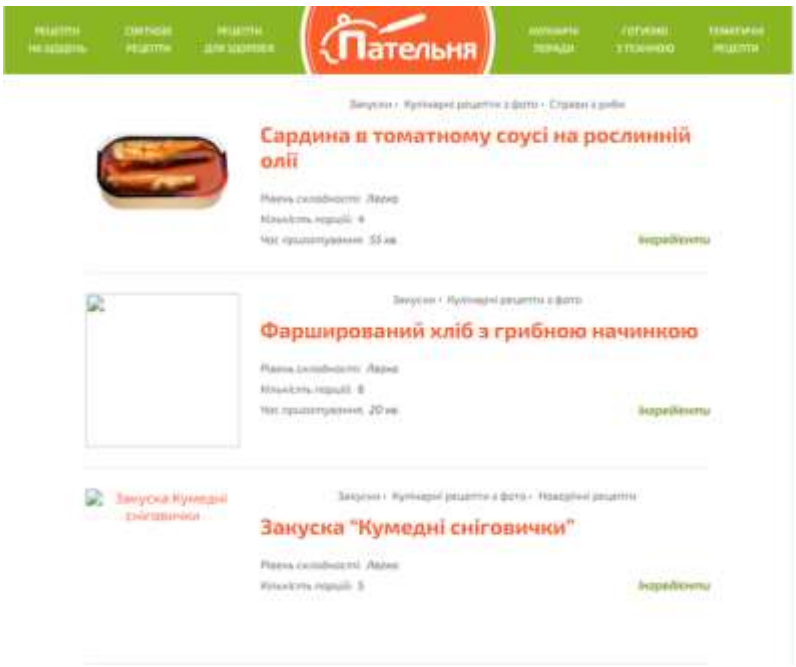

Рисунок 1.1 – Приклад рецепта без доданого зображення

До слабких місць також належить дрібний шрифт у списках інгредієнтів, що ускладнює читання на екранах із високою щільністю пікселів. Крім того, не всі зображення мають альтернативний текст, через що знижується рівень доступності сайту. Технічно помітно, що сторінки без медіафайлів завантажуються швидше, однак відсутність ілюстрацій погіршує сприйняття рецепта. Тому доцільним було б упровадити оптимізовану систему обов’язкового додавання зображень, яка не шкодила б швидкодії. Загалом

«Пательня» справляє позитивне перше враження, але потребує доопрацювання в частині презентації контенту та мобільної навігації.

«Кукорама» вирізняється світлою кольоровою палітрою, зрозумілим розміщенням основних елементів і швидким доступом до розділів «Рецепти», «Блоги» та «Обговорення». Після реєстрації користувач отримує інструменти для створення власних записів: можна додавати інгредієнти, кроки приготування, фотографії та зберігати страви у персональні добірки. Наявність функції «улюблених» є значною перевагою, оскільки користувачеві не потрібно повторно шукати рецепти, які вже його зацікавили.

Водночас широка функціональність призводить до перевантаження сторінок. Поруч із рецептом часто розміщуються рекламні блоки, списки популярних матеріалів і зовнішні віджети, які не завжди пов’язані з конкретною стравою. Через це сторінки завантажуються повільніше, особливо в мобільних мережах. На рисунку 1.2 видно, що поряд із рецептурною частиною виведено значний обсяг другорядної інформації, яка відволікає користувача від основного змісту [2].

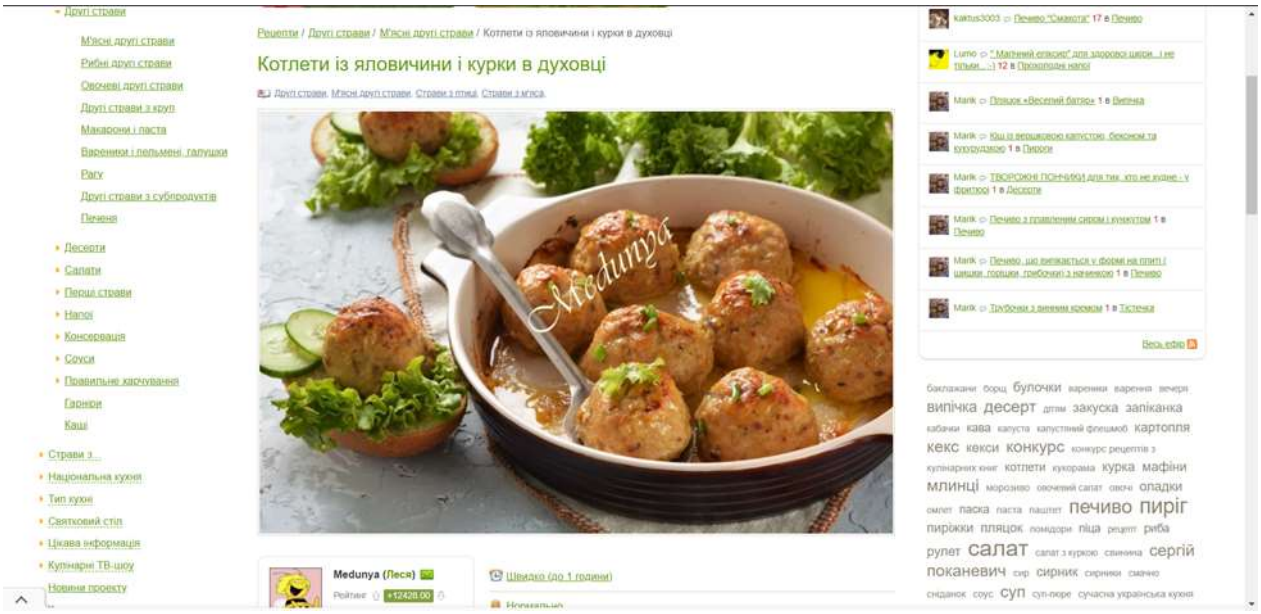

Рисунок 1.2 – Перегляд рецепта з перевантаженим інформаційним наповненням

Також помітні проблеми з типографікою. На одному екрані можуть поєднуватися шрифти різного розміру й блідо-сірі відтінки, що погіршує

читабельність, особливо для людей із порушеннями зору [2]. У таблицях інгредієнтів іноді використовується дрібний курсив, який додатково ускладнює сприйняття тексту. Покращити ситуацію могла б єдина типографічна система, збільшений контраст і зменшення кількості другорядних блоків на сторінках рецептів.

Порівняльний аналіз українських кулінарних платформ «Пательня» та «Кукорама» дає змогу точніше визначити недоліки чинних рішень і сформувати вимоги до нового вебсайту для обміну рецептами. Обидва ресурси мають власну логіку взаємодії з користувачами, однак стикаються зі схожими проблемами: необхідністю збалансувати візуальну привабливість, швидкодію, доступність і соціальну активність.

«Пательня» орієнтується на мінімалістичний інтерфейс і компактну подачу контенту. Користувач бачить картки страв без зайвого тексту, тому головний акцент зроблено на зображеннях. Якщо фото відсутнє, сторінка виглядає неповною, а рецепт втрачає частину привабливості. На мобільних пристроях цей недолік посилюється розірваною навігацією між верхнім меню та футером. Через це навіть прості переходи потребують зайвих дій, що негативно впливає на зручність користування.

«Кукорама», навпаки, пропонує багато додаткової інформації на одній сторінці. Поруч із рецептом виводяться коментарі, популярні публікації, банери та інші блоки. Такий підхід збільшує функціональність, але водночас перевантажує інтерфейс і сповільнює завантаження. На слабших мобільних пристроях це може стати суттєвим бар’єром для користувача. До того ж нечітка типографіка та низький контраст знижують комфорт читання.

Якщо порівнювати сильні сторони, «Пательня» виграє завдяки простоті, швидкому пошуку й чистому інтерфейсу. «Кукорама» натомість має ширший соціальний функціонал: блоги, коментарі, обговорення та персональні добірки. Водночас слабкі місця кожної платформи фактично відображають переваги іншої. Там, де «Пательні» бракує соціальної насиченості й медіаконтенту, «Кукорама» страждає від надмірної кількості елементів і недостатньої візуальної впорядкованості.

Отже, оптимальне рішення має поєднувати легку адаптивну верстку, якісні фото- та відеоматеріали для рецептів, зрозумілу навігацію, швидке завантаження, уніфіковану типографіку та продуману систему соціальної взаємодії. Соціальні функції варто реалізувати так, щоб вони доповнювали основний контент, а не відволікали від нього. Саме поєднання найкращих практик обох платформ може стати основою для сучасного кулінарного вебсайту.

1.3 Формування функціональних і нефункціональних вимог до вебсайту

Вимоги до вебсайту для обміну кулінарними рецептами формуються на основі аналізу предметної області, конкурентних рішень і потреб цільової аудиторії. Насамперед необхідно визначити функціональні можливості, які забезпечують головні сценарії роботи. Автор повинен мати змогу швидко опублікувати рецепт із фотографіями, покроковим описом і списком інгредієнтів. Читач, у свою чергу, має легко знаходити, фільтрувати, оцінювати й зберігати потрібні страви. Тому обов’язковими є пошук за ключовими словами, тегами, кухнями світу, харчовою цінністю та часом приготування. Також необхідно реалізувати коментарі, оцінки, особисті колекції та механізми повторного залучення користувачів.

Нефункціональні вимоги визначають якість користувацького досвіду. Інтерфейс має коректно працювати на екранах від 320 px до 4K, а дизайн повинен бути повністю адаптивним. Час першого візуального відгуку не має перевищувати трьох секунд за умови мобільного з’єднання 4G. Типографіка повинна відповідати щонайменше рівню AA рекомендацій WCAG 2.1, усі медіафайли мають супроводжуватися альтернативним текстом, а кольоровий контраст — залишатися комфортним для користувачів із порушеннями зору. Зображення доцільно оптимізувати у форматі WebP із використанням lazy load, а відео — передавати через HLS-потоки з автоматичним вибором якості.

Архітектурні вимоги передбачають розділення клієнтської та серверної частин, використання REST або GraphQL API, а також можливість горизонтального масштабування бекенду. Дані варто зберігати у реляційній СКБД, доповненій об’єктним CDN-сховищем для великих медіафайлів. Модульна структура репозиторію спростить підтримку CI/CD-конвеєра та дозволить розгортати оновлення без тривалого простою сервісу.

Безпека є окремим важливим блоком вимог. Аутентифікація має підтримувати як класичний вхід через e-mail і пароль, так і соціальний логін. Усі запити повинні передаватися через HTTPS, а критичні дані — зберігатися у зашифрованому вигляді. На рівні бізнес-логіки необхідно передбачити захист від XSS, CSRF і SQL-ін’єкцій. Також потрібна система резервного копіювання з можливістю відновлення даних не пізніше ніж через шість годин після збою. Окремо варто реалізувати видалення персональних даних за запитом користувача, що відповідає вимогам GDPR та українського законодавства.

Важливою частиною є локалізація і керування контентом. Сайт має підтримувати багатомовний інтерфейс, переклад кулінарних мір і можливість публікувати рецепти різними мовами. Для модераторів потрібні інструменти перевірки правопису, пошуку дублікатів, обробки скарг і підтвердження фотографій. Для мотивації авторів доцільно впровадити систему бейджів, рейтингів і можливість експорту рецептів у PDF або друкований формат.

Надійність платформи також має бути визначена кількісно. Планова доступність сервісу повинна становити не менше 99,5 % на рік, а середній час виправлення критичних помилок — не перевищувати 24 годин. Моніторинг продуктивності, журналювання подій і аналітика поведінки користувачів мають інтегруватися вже на ранніх етапах розробки. Це дозволить швидко реагувати на пікові навантаження й удосконалювати рекомендаційні механізми на основі реальних даних.

1.4 Визначення основних категорій користувачів та особливостей їхньої взаємодії із системою

Для коректного проєктування майбутньої системи необхідно визначити основних учасників, які взаємодіятимуть із вебсайтом. Такий аналіз дає змогу окреслити функції кожної ролі, передбачити обмеження доступу, типові переходи між ролями та можливі напрями розвитку функціоналу. Для платформи обміну кулінарними рецептами доцільно виділити чотири базові групи користувачів.

Незареєстрований відвідувач — це користувач, який уперше переходить на сайт або не має облікового запису. Він може переглядати головну сторінку, відкривати рецепти, користуватися базовим пошуком і знайомитися з можливостями ресурсу. Водночас такий відвідувач не може залишати коментарі, оцінювати рецепти чи зберігати їх у власні добірки.

Зареєстрований користувач після створення акаунта отримує ширший набір можливостей. Йому доступні персональний профіль, збереження улюблених рецептів, налаштування фільтрів, коментування й оцінювання матеріалів. За умови активності та дотримання правил спільноти такий користувач може перейти до ролі автора.

Автор контенту — це зареєстрований користувач, який публікує власні рецепти. Для нього важливими є зручне завантаження фото, автозаповнення інгредієнтів, перевірка унікальності тексту, попередній перегляд і статистика переглядів. Автори взаємодіють між собою через коментарі, відповіді, добірки та тематичні активності, що сприяє зростанню бази рецептів.

Адміністратор має найширші права в системі. Він керує категоріями, призначає модераторські ролі, перевіряє дотримання правил, контролює авторські права, опрацьовує скарги та відповідає за резервне копіювання і відновлення роботи ресурсу після збоїв.

На діаграмі ролей (див. рис. 1.3) показано, як змінюються права користувача: від незареєстрованого відвідувача до зареєстрованого користувача,

автора, модератора або адміністратора. Така схема демонструє зв’язок між рівнем довіри до користувача та обсягом доступних функцій.

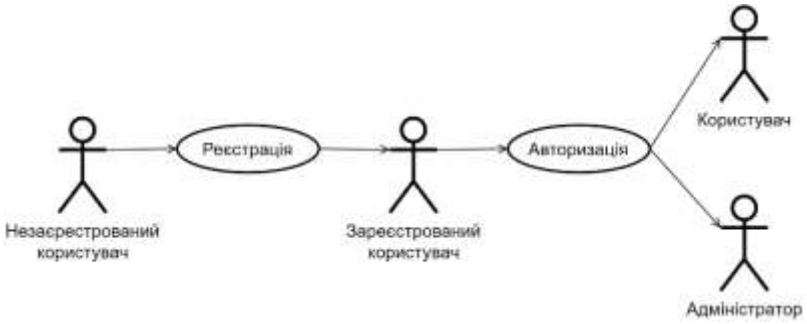

Рисунок 1.1 – Діаграма ролей користувачів вебсайту

Другий графічний матеріал (див. рис. 1.4) деталізує сторінки й функції, доступні кожній ролі. Відвідувач без акаунта бачить головну сторінку, каталог рецептів, інформаційні розділи та форми входу або реєстрації. Зареєстрований користувач додатково отримує особистий кабінет і функції взаємодії з контентом. Автор має доступ до конструктора публікацій і статистики власних матеріалів. Адміністратор, своєю чергою, працює з усіма модулями платформи, включно з керуванням користувачами та глобальними налаштуваннями.

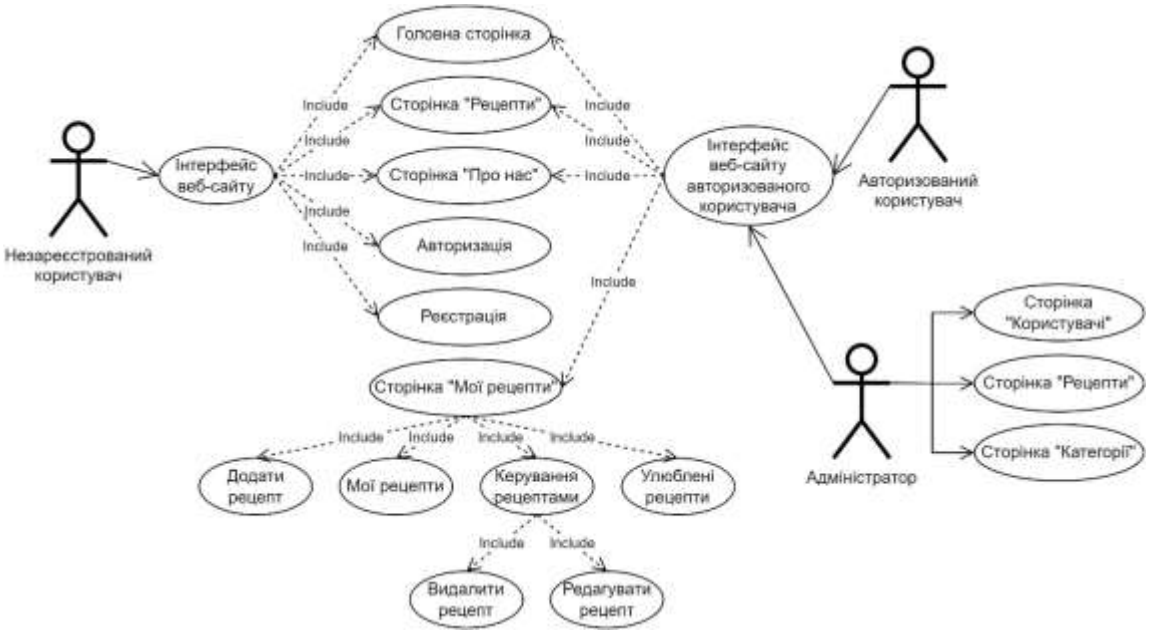

Рисунок 1.4 – Діаграма сценаріїв використання вебсайту

Отже, чотири описані ролі формують базову модель взаємодії в кулінарній платформі. Чітке розмежування прав спрощує проєктування системи доступу, підвищує безпеку даних і дозволяє поступово відкривати користувачам нові можливості відповідно до їхньої активності.

1.5 Опис сценаріїв використання платформи для обміну кулінарними рецептами

Щоб зрозуміти практичну логіку роботи платформи, доцільно описати типові сценарії взаємодії користувачів із вебсайтом. Кожен сценарій відображає конкретну ситуацію: чому людина відкриває ресурс, які дії виконує і який результат отримує.

Перший сценарій стосується незареєстрованого відвідувача, який шукає, наприклад, «пасту з креветками без глютену». Він переходить на головну сторінку й бачить спрощений інтерфейс із великим рядком пошуку та популярними категоріями. Після введення запиту система пропонує підказки, а потім формує перелік страв із фільтрами за часом приготування та дієтичними позначками. Пройшовши шлях «Головна → Пошук → Перегляд рецепта», користувач може або завершити перегляд, або зареєструватися, щоб зберегти рецепт.

Інший сценарій описує роботу автора. Користувач натискає «Додати рецепт», вводить назву, опис, інгредієнти та кроки приготування. Автодоповнення допомагає обирати стандартизовані одиниці вимірювання, а після завантаження фото автор може переглянути майбутню сторінку рецепта. Система пропонує хештеги, перевіряє текст на можливе дублювання й автоматично розраховує орієнтовну харчову цінність. Після публікації рецепт надходить на модерацію, а в профілі автора з’являється відповідний статус.

Зареєстрований користувач взаємодіє з уже опублікованим рецептом через стрічку рекомендацій. Він може поставити оцінку, залишити коментар, запропонувати альтернативний інгредієнт і додати страву до власної колекції. Пізніше, відкривши збережену добірку, користувач може перейти в режим

покрокового приготування з великими кнопками для зручності роботи зі смартфона. Після приготування він завантажує фото власного результату, що доповнює галерею рецепта та підвищує його активність.

Окремий сценарій пов’язаний із роботою адміністратора. Отримавши сповіщення про нову публікацію, він перевіряє текст, оригінальність фото та відповідність правилам платформи. За потреби адміністратор додає рецепт до тематичної добірки або повертає матеріал автору на доопрацювання. Якщо система виявляє підозрілу активність у коментарях, адміністратор може приховати повідомлення, заблокувати обліковий запис або розпочати додаткову перевірку.

Ще один корисний сценарій — створення тижневого меню. Користувач додає кілька страв до плану, після чого система автоматично формує список покупок, об’єднує однакові інгредієнти та підраховує їхню сумарну кількість. Якщо в профілі зазначені алергії чи харчові обмеження, сайт пропонує безпечні замінники або виключає небажані продукти. Результат можна експортувати у PDF або надіслати на електронну пошту.

Розгорнутий перелік сценаріїв подано у таблиці-реєстрі варіантів використання (див. додаток А). У ній узагальнено основних акторів, їхні дії, повноваження й очікувані результати. Такий формат полегшує управління вимогами, допомагає пріоритезувати завдання у беклозі та швидко вносити зміни у разі появи нових функцій або потреб користувачів [5].

1.6 Аналіз архітектурних і технологічних підходів до створення вебсайтів

Під час вибору підходу до створення вебсайту для обміну рецептами необхідно оцінити швидкість запуску, масштабованість, безпеку та гнучкість подальшого розвитку. Одним із найпростіших варіантів є використання готових CMS, наприклад WordPress із плагінами для recipe-card. Таке рішення дозволяє швидко створити мінімально життєздатний продукт і скористатися великою кількістю тем та розширень. Однак CMS часто містить надлишковий код,

складніше оптимізується під нестандартні поля, наприклад калорійність або алергени, і потребує ретельного захисту від типових вебзагроз.

Low-code та no-code платформи, зокрема Bubble, Webflow або Wix, ще більше спрощують запуск. Вони дають змогу створити базову версію сайту без глибоких технічних знань і швидко перевірити ринковий попит. Проте головним недоліком є залежність від провайдера. Перенесення бекенду або розширення функціональності часто потребує повного переписування, а складні рекомендаційні механізми можуть обмежуватися тарифами й технічними можливостями платформи.

Класичні MVC-фреймворки, такі як Django, Ruby on Rails або Laravel, забезпечують зрозумілу структуру проєкту, ORM для роботи з базою даних і готові модулі автентифікації. Вони добре підходять для швидкого розроблення CRUD-функціоналу, але за великого навантаження потребують додаткових архітектурних рішень для масштабування. Крім того, серверна генерація сторінок без SPA-механізмів може знижувати інтерактивність, оскільки частина дій супроводжується перезавантаженням сторінки.

JAMstack із генерацією статичних сторінок у Gatsby або Next.js і винесенням динаміки у serverless-функції забезпечує високу швидкість завантаження через CDN. Такий підхід зручний для сайтів із переважно статичним контентом, однак при великій кількості рецептів і частих оновленнях виникають складнощі з build-процесами, кешуванням і підтримкою актуальності даних.

SPA на React, Vue або Svelte у поєднанні з REST чи GraphQL API дає змогу створити інтерактивний інтерфейс із миттєвими реакціями на оцінки, коментарі та фільтрацію. Такий підхід добре підходить для мобільної версії та подальшого розвитку у форматі PWA. Водночас він потребує окремої роботи над фронтендом і бекендом, а питання SEO слід вирішувати через серверний рендеринг або пререндеринг.

Мікросервісна архітектура з Docker і Kubernetes забезпечує високу масштабованість, відмовостійкість і можливість розробляти різні сервіси незалежно. Проте разом із перевагами зростає складність інфраструктури,

з’являється потреба у DevOps-підтримці, моніторингу, кешуванні та контролі мережевих викликів.

З огляду на те, що кулінарний портал має щодня наповнюватися користувацьким контентом, а інтерфейс повинен працювати без зайвих перезавантажень, найбільш збалансованим є підхід SPA з окремим API. На старті достатньо React-фронтенду із серверним рендерингом через Next.js, бекенду на PHP або Node.js і реляційної бази даних MySQL чи PostgreSQL. Надалі окремі підсистеми, наприклад автентифікацію, рекомендації або обробку медіафайлів, можна винести у мікросервіси. Додавання Redis, S3-сумісного сховища, Kafka чи Spark забезпечить стабільність при зростанні аудиторії та збільшенні навантаження [6].

Найдоцільнішим способом реалізації платформи обміну кулінарними рецептами є розробка з нуля, коли клієнтська й серверна частини створюються відповідно до конкретних вимог проєкту. Такий підхід дозволяє уникнути обмежень готових CMS і конструкторів, одразу закласти адаптивну односторінкову архітектуру, продуману систему ролей і гнучку логіку взаємодії користувачів із контентом.

Розробка з нуля дає можливість самостійно обрати технології, які відповідають очікуваному навантаженню та сценаріям використання. Для фронтенду доцільно використати React із серверним рендерингом через Next.js, а для бекенду — PHP або Node.js із REST чи GraphQL API. Така комбінація дозволяє надалі інтегрувати рекомендаційні механізми, покрокові відео, push-сповіщення, офлайн-режим PWA та інші функції без жорстких обмежень сторонньої платформи.

Окрема база даних у MySQL або PostgreSQL дає змогу коректно змоделювати рецепти, інгредієнти, харчову цінність, теги, коментарі та оцінки. Використання Docker-контейнерів і CI/CD-конвеєра забезпечить контроль версій, передбачуване розгортання та можливість перенесення проєкту між хмарними середовищами. Безпека також легше контролюється в кастомному рішенні: можна налаштувати HTTPS, шифрування критичних даних, аудит коду та регулярне оновлення залежностей. У підсумку така стратегія створює

технічну основу, яка здатна розвиватися разом з аудиторією і не накопичувати зайвий технічний борг [7].

На рисунку 1.5 подано узагальнену схему життєвого циклу вебсайту, яка охоплює всі ключові етапи — від формування вимог до подальшої підтримки ресурсу.

Рисунок 1.5 – Схема життєвого циклу розроблення та супроводу вебсайту

Першим етапом є визначення мети, цільової аудиторії та переліку основних функцій платформи. Після цього команда переходить до проєктування логіки роботи, структури даних, інтерфейсів і сценаріїв взаємодії. Далі відбувається безпосередня розробка, під час якої концепція перетворюється на робочий програмний продукт.

Наступний етап — тестування. Воно охоплює перевірку функціональності, безпеки, адаптивності, продуктивності та зручності користування. Після виправлення виявлених помилок сайт розгортається у виробничому середовищі, налаштовуються домен, хостинг, SSL-сертифікат, аналітика та базові засоби моніторингу. Завершальна фаза — експлуатація і супровід. Вона передбачає регулярне оновлення функцій, встановлення патчів безпеки, обробку звернень користувачів і розвиток платформи відповідно до нових потреб спільноти [8].

1.7 Обґрунтування вибору технологічного стеку та засобів розроблення

Для створення вебсайту обміну кулінарними рецептами доцільно використовувати стек, що базується на відкритих стандартах і добре підтримується більшістю пристроїв та хостинг-платформ. Основними складовими такого рішення є HTML5, CSS3, JavaScript, PHP і реляційна СУБД MySQL.

HTML5 обрано як базу для структурування сторінок. Семантичні теги <article>, <section>, <nav> і <figure> дозволяють логічно описувати рецепти, блоки інгредієнтів, навігацію та ілюстрації. Це покращує доступність, спрощує індексацію пошуковими системами й робить структуру сторінки зрозумілішою для скринрідерів. Крім того, HTML5 підтримує вбудовані медіатеги <video> та <audio>, що дає можливість додавати покрокові відео без сторонніх плагінів.

CSS3 відповідає за оформлення та адаптивність. Медіазапити дозволяють реалізувати Mobile First-підхід і забезпечити коректне відображення на смартфонах, планшетах і великих моніторах. Flexbox і CSS Grid зручні для побудови сітки карток рецептів, а анімації та переходи допомагають зробити інтерфейс плавнішим без значного навантаження на JavaScript. CSS-змінні також спрощують створення світлої та темної теми.

JavaScript забезпечує інтерактивність на клієнтському боці. Сучасний синтаксис ES6+, модулі, async/await і Fetch API дозволяють реалізувати динамічний пошук, фільтрацію, обробку форм і оновлення частини сторінки без перезавантаження. Підтримка Service Worker відкриває можливість створення PWA, завдяки чому популярні рецепти й медіафайли можуть бути доступними навіть за нестабільного інтернет-з’єднання.

Для серверної частини доцільно використати PHP. Ця мова має зрілу екосистему, низький поріг входу й широку підтримку на більшості веб-хостингів. Майже кожен провайдер пропонує LAMP або LEMP-середовище, що знижує витрати на інфраструктуру та спрощує розгортання. Сучасні версії PHP забезпечують достатню продуктивність для вебпроєктів середнього масштабу, а

фреймворки Laravel, Symfony і CodeIgniter надають готові засоби для маршрутизації, автентифікації, ORM і захисту від CSRF, XSS та SQL-ін’єкцій.

PHP добре інтегрується з фронтендом через JSON API і може використовувати серверний рендеринг шаблонів для сторінок, важливих із погляду SEO. У поєднанні з MySQL він дозволяє ефективно працювати з рецептами, користувачами, інгредієнтами, коментарями й оцінками. Такий вибір виправданий доступністю хостингів, стабільністю екосистеми, великою кількістю готових бібліотек і можливістю швидко запустити MVP із перспективою подальшого розвитку [10].

MySQL обрано для зберігання структурованих даних. Реляційна модель добре відповідає сутностям «Користувач», «Рецепт», «Інгредієнт», «Коментар» та «Оцінка», між якими існують зв’язки «один-до-багатьох» і «багато-до-багатьох». Індекси, повнотекстовий пошук InnoDB і механізми реплікації дозволяють оптимізувати читання даних у періоди високого навантаження, наприклад перед святами. Популярність MySQL гарантує доступність документації, хостингових рішень, інструментів міграції та стратегій резервного копіювання.

Отже, стек HTML5, CSS3, JavaScript, PHP і MySQL забезпечує баланс між простотою розробки, доступністю інфраструктури, продуктивністю та можливістю поступового масштабування. Він підходить для запуску кулінарної платформи й подальшого розширення її функціональності відповідно до потреб користувачів [11].

1.8 Вибір програмного середовища для реалізації вебсайту

У таблиці 1.1 подано порівняння популярних середовищ розробки, які можуть використовуватися для роботи зі стеком HTML5 / CSS3 / JavaScript / PHP / MySQL

Таблиця 1.1 – Порівняльна характеристика середовищ веброзробки

Характеристика	**Visual Studio Code**	**WebStorm**	**Atom**	**Sublime Text**
Ціна	Безкоштовний	Платний	Безкоштовний	Платний
Мови програмування	Багатомовна	HTML, CSS, JS	Багатомовна	Багатомовна
Розширення / плагіни	Наявні	Наявні	Наявні	Наявні
Налагодження	Вбудоване	Вбудоване	Використання плагінів	Використання плагінів
Системні вимоги	Невисокі	Невисокі	Середні	Невисокі
Спільнота	Велика	Менша	Велика	Менша

Під час вибору IDE для створення й супроводу кулінарного вебсайту доцільно враховувати три основні критерії: функціональність «з коробки», доступність розширень і загальну вартість використання.

Visual Studio Code є одним із найпопулярніших редакторів серед веброзробників. Він підтримує велику кількість розширень для JavaScript, PHP, SQL, систем контролю версій, відлагодження й DevOps-інструментів. Вбудований термінал, інтеграція з Git і GitHub, а також безкоштовна ліцензія роблять VS Code зручним рішенням для командної роботи та навчальних проєктів.

WebStorm має глибоку інтеграцію з JavaScript-екосистемою. Він пропонує якісні підказки для React і Vue, інструменти рефакторингу, аналіз продуктивності та перевірку пакетів. Для складних фронтенд-проєктів це потужне середовище, однак у комерційному використанні потрібно враховувати витрати на ліцензію.

Atom більше не є актуальним вибором, оскільки його підтримку було припинено. Відсутність оновлень і патчів безпеки робить цей редактор непридатним для сучасного публічного вебпроєкту. Sublime Text залишається швидким і легким редактором, підтримує багато мов і зручний для невеликих

правок. Проте він має меншу екосистему порівняно з VS Code, а повноцінна версія потребує придбання ліцензії.

Для проєкту, який передбачає швидкий старт і можливе розширення команди, найкращим основним середовищем є Visual Studio Code. Він безкоштовний, регулярно оновлюється, має великий маркетплейс плагінів і не створює ліцензійних обмежень. WebStorm може використовуватися як додатковий професійний інструмент для складної роботи з JavaScript, Sublime Text — як легкий редактор для швидких змін, а Atom доцільно виключити з технічного стеку через припинення підтримки [12].

1.9 Висновок до першого розділу

У першому розділі розглянуто особливості кулінарного сегмента інтернету та визначено потребу в україномовній платформі з адаптивним дизайном, мультимедійною подачею, соціальними функціями та зручними механізмами пошуку. Аналіз глобальних і вітчизняних рішень показав, що наявні сервіси часто мають недостатню локалізацію, перевантаженість другорядним контентом або обмежену соціальну взаємодію. Порівняння платформ «Пательня» і «Кукорама» дозволило визначити основні напрями вдосконалення: якісні зображення й відео, оптимізовану мобільну навігацію, збалансовану структуру сторінок, читабельну типографіку та гейміфіковані механізми активності.

На основі проведеного аналізу сформовано функціональні й нефункціональні вимоги до майбутнього вебсайту. Основну увагу приділено безпеці персональних даних, доступності, багатомовності, швидкодії та можливості масштабування. Також визначено ключові ролі користувачів — незареєстрований відвідувач, зареєстрований користувач, автор контенту й адміністратор — та описано типові сценарії їхньої взаємодії із системою.

Оцінка архітектурних підходів показала, що найбільш доцільним є створення платформи з нуля на основі SPA з окремим API. Такий підхід дає змогу забезпечити гнучкість, інтерактивність і можливість подальшого розвитку. Для реалізації обрано стек HTML5, CSS3, JavaScript, PHP і MySQL, який поєднує

доступність, стабільність, продуктивність і широку підтримку на хостинг-платформах.

Порівняння середовищ розробки засвідчило перевагу Visual Studio Code як основного інструмента завдяки безкоштовній ліцензії, великій кількості розширень і зручній інтеграції з Git. WebStorm може використовуватися як додаткове середовище для складної фронтенд-розробки. Сукупність проведених досліджень і прийнятих рішень формує методологічну основу для подальшого створення вебсайту, здатного задовольнити потреби широкої аудиторії та забезпечити стабільний розвиток кулінарної онлайн-спільноти.

РОЗДІЛ 2. ПРОЄКТУВАННЯ СТРУКТУРИ ТА ТЕХНІЧНОЇ ОСНОВИ ВЕБСАЙТУ

2.1 Побудова логічної та технічної архітектури вебсайту

Для розроблення вебсайту обміну кулінарними рецептами доцільно використати трирівневу архітектуру, яка передбачає поділ системи на клієнтський рівень, рівень прикладної логіки та рівень даних. Такий підхід дає змогу чітко розмежувати функції окремих компонентів, спростити супровід системи й забезпечити можливість її подальшого масштабування.

Клієнтський рівень відповідає за взаємодію користувача з вебсайтом. На цьому рівні працює браузер або мобільна PWA-оболонка, яка отримує HTML-розмітку, таблиці стилів і JavaScript-сценарії. Саме тут відображаються сторінки, картки рецептів, форми публікації, фільтри пошуку та інші елементи інтерфейсу. Також клієнтський рівень може підтримувати локальне кешування статичних ресурсів і часткову роботу в офлайн-режимі за допомогою Service Worker.

Рівень прикладної логіки забезпечує виконання основних бізнес-процесів вебсайту. До нього належать API-контролери, модулі авторизації, валідатори, серіалізатори відповідей, механізми кешування та компоненти, що відповідають за рекомендації. На цьому рівні перевіряються права доступу, обробляються запити на створення й редагування рецептів, формується рейтинг страв, здійснюється обробка зображень і виконується підготовка даних для передавання клієнту.

Рівень даних призначений для зберігання всієї інформації, необхідної для роботи платформи. Основою цього рівня є реляційна база даних MySQL або PostgreSQL, у якій зберігаються відомості про користувачів, рецепти, категорії, інгредієнти, коментарі, оцінки та збережені рецепти. За потреби для великих медіафайлів може використовуватися окреме об’єктне сховище. Доступ до бази даних здійснюється через ORM або SQL-запити, що дозволяє отримувати, змінювати й передавати інформацію у структурованому вигляді.

Трирівневий підхід дає змогу незалежно розвивати інтерфейс, серверну логіку та підсистему зберігання даних. Наприклад, редизайн клієнтської частини не потребує повної зміни бази даних, а заміна СУБД або оптимізація серверної логіки не впливає безпосередньо на зовнішній вигляд вебсайту. Крім того, така структура підвищує безпеку, полегшує тестування та дозволяє масштабувати окремі частини системи залежно від навантаження.

Трирівнева архітектура, подана на рисунку 2.1, відображає логічну організацію вебсайту як сукупність взаємопов’язаних, але функціонально відокремлених шарів.

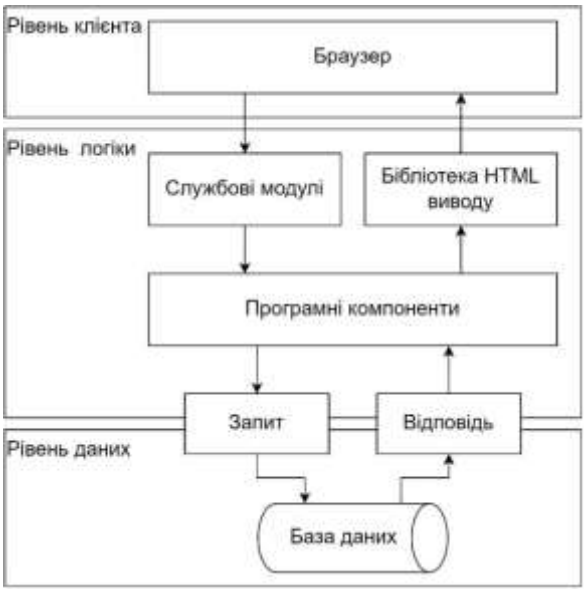

Рисунок 2.1 – Схема трирівневої архітектури вебсайту

На презентаційному рівні браузер користувача надсилає HTTP-запити, отримує відповіді від сервера та відображає їх у вигляді сторінок або динамічних елементів. Тут працюють HTML, CSS і JavaScript, а також виконуються AJAX- або Fetch-запити для оновлення окремих частин сторінки без повного перезавантаження.

Прикладний рівень опрацьовує запити користувача, перевіряє введені дані, застосовує правила безпеки, звертається до бази даних і формує відповідь. Наприклад, коли користувач відкриває рецепт, система перевіряє доступ, отримує потрібний запис із бази, додає коментарі та передає клієнту готову відповідь у вигляді HTML або JSON.

Рівень даних забезпечує надійне зберігання інформації та підтримує її цілісність. У базі даних розміщуються сутності «Користувач», «Рецепт», «Категорія», «Коментар», «Оцінка» та інші об’єкти, необхідні для роботи платформи. Індекси й зовнішні ключі пришвидшують пошук і запобігають появі некоректних або «сирітських» записів.

Отже, обрана архітектура створює технічну основу для стабільної, безпечної та зручної в підтримці системи

2.2 Розроблення ієрархічної схеми та навігаційної моделі вебсайту

Навігаційна структура вебсайту визначає розміщення основних сторінок і логіку переходів між ними. Вона має бути зрозумілою як для користувачів, так і для розробників, оскільки саме від неї залежить зручність користування платформою та подальший супровід проєкту [14]. Ієрархічну схему вебсайту наведено на рисунку 2.2.

Рисунок 2.2 – Структурна модель вебсайту

У центрі структури розташована головна сторінка, яка є початковою точкою входу для більшості відвідувачів. Від неї користувач може перейти до ключових розділів вебсайту: каталогу рецептів, особистого кабінету, інформаційної сторінки, сторінок реєстрації та авторизації, а також адміністративної частини.

Розділ «Рецепти» є відкритим каталогом, у якому користувачі можуть переглядати опубліковані страви, використовувати пошук, фільтри та сортування. Цей блок доступний як зареєстрованим користувачам, так і гостям сайту.

Розділ «Мої рецепти» призначений для авторизованих користувачів. Він об’єднує сторінки додавання нового рецепта, перегляду власних публікацій, керування рецептами та доступу до збережених страв. Тут користувач може створювати новий контент, редагувати вже опубліковані матеріали, видаляти записи або переглядати рецепти, додані до вибраного.

Сторінка «Про нас» містить загальну інформацію про платформу, її призначення, контактні дані та політику конфіденційності. Вона виконує інформаційну функцію та допомагає користувачам краще зрозуміти ідею сервісу.

Сторінка «Реєстрація» забезпечує створення нового облікового запису. Користувач вводить необхідні дані або, за наявності такої можливості, використовує швидкий вхід через соціальні мережі. Сторінка «Авторизація» призначена для входу зареєстрованих користувачів і може містити механізм відновлення пароля.

Окремий блок становить адміністративна частина, доступна лише користувачам із відповідними правами. Вона містить інструменти для керування обліковими записами, категоріями та всіма рецептами. Адміністратор може переглядати користувачів, змінювати їхні ролі, блокувати порушників, створювати й редагувати категорії, а також модерувати опублікований контент.

Запропонована структура забезпечує чітке розмежування прав доступу. Гості можуть переглядати загальний контент, зареєстровані користувачі отримують інструменти для створення й збереження рецептів, а адміністратори мають окремий функціонал для контролю над системою.

2.3 Створення логічної та фізичної моделі бази даних вебсайту

Під час створення вебсайту важливо правильно організувати зберігання взаємопов’язаних даних. Платформа працює з інформацією про користувачів, рецепти, категорії, коментарі та збережені матеріали, тому база даних має бути структурованою, продуктивною та придатною до розширення.

На етапі концептуального моделювання було враховано три основні принципи: нормалізацію даних, мінімізацію дублювання та прозорість зв’язків між сутностями. Це дозволяє уникнути надлишкового зберігання однакової інформації, спростити оновлення записів і забезпечити коректну роботу фільтрів, пошуку та аналітичних вибірок.

У системі передбачено шість основних об’єктів даних:

- Users — таблиця користувачів, у якій зберігаються облікові записи, ролі, налаштування профілю та дата реєстрації.
- Recipes — центральна таблиця рецептів, що містить назву страви, опис, інгредієнти, кроки приготування, час, калорійність, зображення та посилання на автора.
- Saved_recipes — службова таблиця для збережених рецептів, яка фіксує, які страви користувач додав до власної колекції.
- Comments — таблиця коментарів, що забезпечує обговорення рецептів і зв’язує кожен коментар із конкретним користувачем та рецептом.
- Categories — довідник категорій, який містить перелік кулінарних рубрик.
- Recipe_categories — проміжна таблиця, що реалізує зв’язок «багато-до-багатьох» між рецептами та категоріями.

Такий поділ дає змогу зберігати дані без зайвого дублювання. Наприклад, назва категорії не повторюється в кожному рецепті, а зберігається окремо в таблиці categories. Якщо потрібно додати нову рубрику або змінити її назву, це можна зробити без редагування таблиці рецептів.

У таблиці users первинний ключ id використовується для ідентифікації кожного користувача. Поле admin або аналогічний атрибут ролі дозволяє

відрізняти звичайних користувачів від адміністраторів. Для паролів передбачено зберігання хешованих значень, що відповідає вимогам безпеки.

Таблиця 1.1 – Структура таблиці користувачів users

Назва поля	Тип даних	Призначення
id	int (255)	ID користувача
admin	tinyint (1)	Роль користувача
login	varchar (20)	Ім’я користувача
email	varchar (50)	Електрона пошта користувача
password	varchar (255)	Пароль користувача
create_date	datetime	Дата реєстрації користувача

Таблиця recipes є основним сховищем кулінарного контенту. Кожен запис у ній описує окрему страву й містить як базові характеристики, так і додаткові метадані. До цієї таблиці звертаються модулі пошуку, сторінки перегляду рецептів, система коментарів і рекомендаційні механізми.

Таблиця 2.2 – Структура таблиці рецептів recipes

Назва поля	Тип даних	Призначення
id	int (255)	ID рецепту
user_id	int (255)	ID користувача
title	varchar (255)	Назва рецепту
ingredients	text	Інгредієнти рецепту
preparation	text	Інструкція приготування
image	varchar (255)	Назва та формат зображення
published	tinyint (1)	Статус публікації
create_date	datetime	Дата додавання рецепту

Таблиця saved_recipes виконує функцію закладок. Вона зберігає зв’язок між користувачем і рецептом, який він додав до вибраного. Завдяки цьому можна

швидко сформувати персональну добірку, а також визначити популярність окремих страв. Дата створення запису дозволяє сортувати збережені рецепти за часом додавання.

Таблиця 2.3 – Структура таблиці збережених рецептів saved_recipes

Назва поля	Тип даних	Призначення
id	int (255)	ID збереженого рецепту
user_id	int (255)	ID користувача
recipe_id	int (255)	ID рецепту
create_date	datetime	Дата збереження рецепту

Таблиця comments використовується для збереження відгуків, порад і уточнень під рецептами. Кожен коментар пов’язаний із конкретним рецептом і автором. Це дає змогу відображати обговорення на сторінці страви, модерувати повідомлення та аналізувати активність користувачів.

Таблиця 2.4 – Структура таблиці коментарів comments

Назва поля	Тип даних	Призначення
1	2	3
id	int (255)	ID коментаря
recipe_id	int (255)	ID рецепту
user_id	int (255)	ID користувача
comment	text	Текст коментаря
create_date	datetime	Дата додавання коментаря

Таблиця recipe_categories є проміжною ланкою між рецептами та рубриками. Вона потрібна для реалізації ситуації, коли один рецепт може належати до кількох категорій, а одна категорія містить багато рецептів. Наприклад, одна страва може одночасно входити до розділів «Сніданки», «Вегетаріанські страви» та «Швидкі рецепти».

Таблиця 2.5 – Структура таблиці зв’язків recipe_categories

Назва поля	Тип даних	Призначення
id	int (255)	ID рецепт-категорії
recipe_id	int (255)	ID рецепту
categoru_id	int (255)	ID категорії
create_date	timestamp	Дата додавання категорії до рецепту

Таблиця categories містить перелік тематичних рубрик. Вона забезпечує єдину систему класифікації рецептів і запобігає появі дубльованих або випадково написаних по-різному категорій. За потреби цю структуру можна розширити, додавши опис категорії, іконку або порядок відображення.

Таблиця 2.6 – Структура таблиці категорій categories

Назва поля	Тип даних	Призначення
id	int (10)	ID категорії
name	varchar (50)	Назва категорії

Концептуальну модель бази даних вебсайту подано на рисунку 2.3.

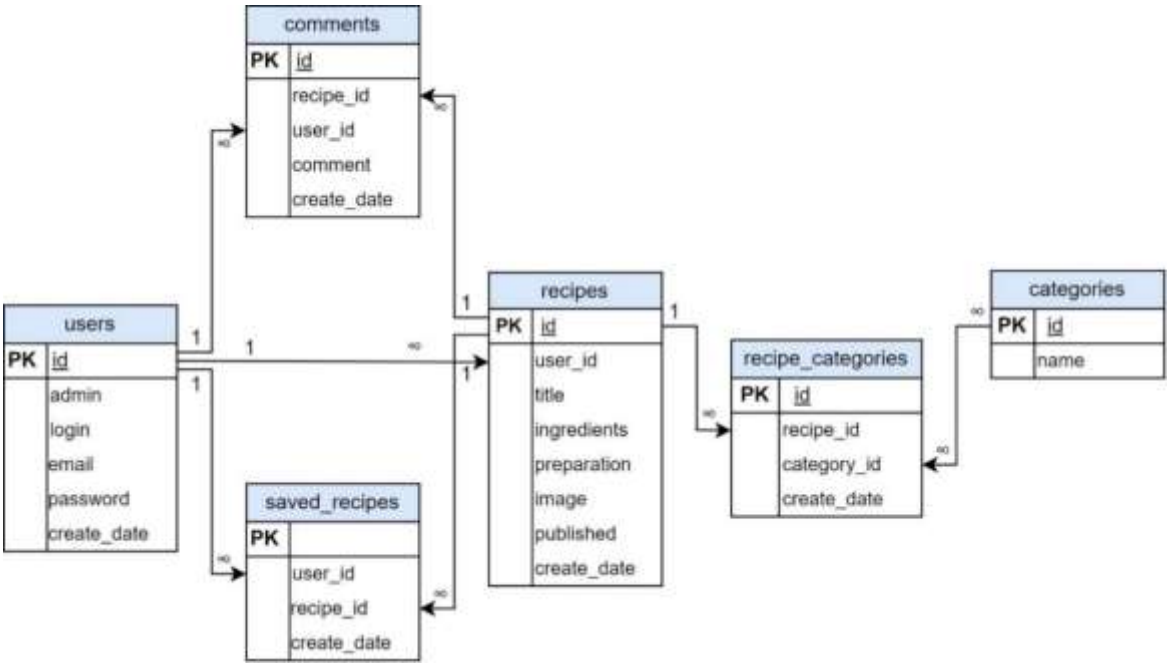

Рисунок 2.3 – Концептуальна модель бази даних вебсайту

У моделі таблиця users є однією з ключових, оскільки вона пов’язана з рецептами, коментарями та збереженими матеріалами. Один користувач може створити багато рецептів, залишити багато коментарів і додати до вибраного необмежену кількість страв. Усі ці зв’язки реалізуються через зовнішні ключі.

Таблиця recipes також має кілька важливих зв’язків. Один рецепт може мати багато коментарів і багато записів у таблиці saved_recipes. Крім того, через таблицю recipe_categories він може бути пов’язаний із кількома категоріями.

Наявність зовнішніх ключів гарантує референційну цілісність даних. Система не дозволить створити коментар до неіснуючого рецепта або зберегти до вибраного страву, якої немає в базі. Каскадні обмеження допомагають автоматично очищати залежні записи в разі видалення основного об’єкта.

Отже, розроблена модель бази даних є достатньо гнучкою для подальшого розширення, підтримує логічну цілісність інформації та забезпечує ефективну роботу вебсайту навіть за збільшення кількості користувачів і контенту.

2.4 Опис користувацьких сценаріїв і модульної організації системи

Поведінкова модель вебсайту описує, як система реагує на дії користувача: відкриття сторінки, пошук рецепта, реєстрацію, додавання страви, коментування або модерування контенту. Для цього вебсайт поділено на окремі модулі, кожен із яких виконує власну функцію та взаємодіє з іншими через контролери й спільні службові файли.

До навігаційного блоку належать основні сторінки публічної частини сайту. Файл index.php виконує роль головної сторінки, підтягує актуальні або популярні рецепти та забезпечує перехід до ключових розділів. Сторінка about_us.php містить інформацію про сервіс, його призначення та контакти. Файл search.php обробляє пошукові запити, передає параметри фільтрації до відповідного контролера й повертає результати користувачу.

Модуль автентифікації забезпечує реєстрацію, вхід і вихід із системи. Файл register.php відповідає за створення нового облікового запису, перевірку введених даних і збереження користувача в базі. Файл login.php виконує

перевірку облікових даних, а logout.php завершує сесію та перенаправляє користувача на головну сторінку.

Блок керування рецептами містить сторінки, пов’язані зі створенням, переглядом і редагуванням кулінарного контенту. Файл all_recipes.php відображає загальний каталог опублікованих страв. Сторінка add_recipe.php дає змогу додати новий рецепт із назвою, описом, інгредієнтами, інструкцією та зображенням. Файл edit_recipe.php використовується для редагування власних публікацій, а my_recipes.php і manage_recipes.php дозволяють автору переглядати й адмініструвати свої записи. Сторінка favorite_recipes.php показує рецепти, додані користувачем до вибраного, а single_recipe.php відображає повну інформацію про конкретну страву, коментарі та схожі рецепти.

Адміністративна зона призначена для керування всією платформою. Вона містить інструменти для роботи з користувачами, категоріями та рецептами. Файли add_category.php і edit_category.php дозволяють створювати й редагувати рубрики. Сторінка index_users.php використовується для перегляду користувачів, зміни ролей і блокування облікових записів. Файли published_recipes.php та unpublished_recipes.php допомагають адміністратору контролювати опубліковані й неопубліковані матеріали.

Окрему роль відіграють файли доступу до даних. connect.php відповідає за підключення до бази даних, а db.php містить універсальні методи для виконання операцій вибірки, додавання, оновлення та видалення. Такий підхід дозволяє не дублювати код підключення в кожному модулі й зменшує ризик помилок під час роботи з базою.

Шаблонні файли, зокрема header.php, footer.php і comments.php, забезпечують єдиний вигляд сторінок і повторне використання типових елементів інтерфейсу. Наприклад, шапка й підвал сайту підключаються на різних сторінках, що полегшує редагування дизайну.

Контролери логіки відповідають за обробку запитів. recipes_controller.php керує діями з рецептами, users_controller.php — реєстрацією й авторизацією, comments_controller.php — додаванням і перевіркою коментарів. Адміністративні контролери обробляють дії з користувачами, категоріями та

рецептами в бек-офісі. Завдяки цьому сторінки представлення не перевантажуються бізнес-логікою, а лише передають дані відповідним модулям.

Логіку взаємодії модулів адміністративної частини подано на рисунку 2.4.

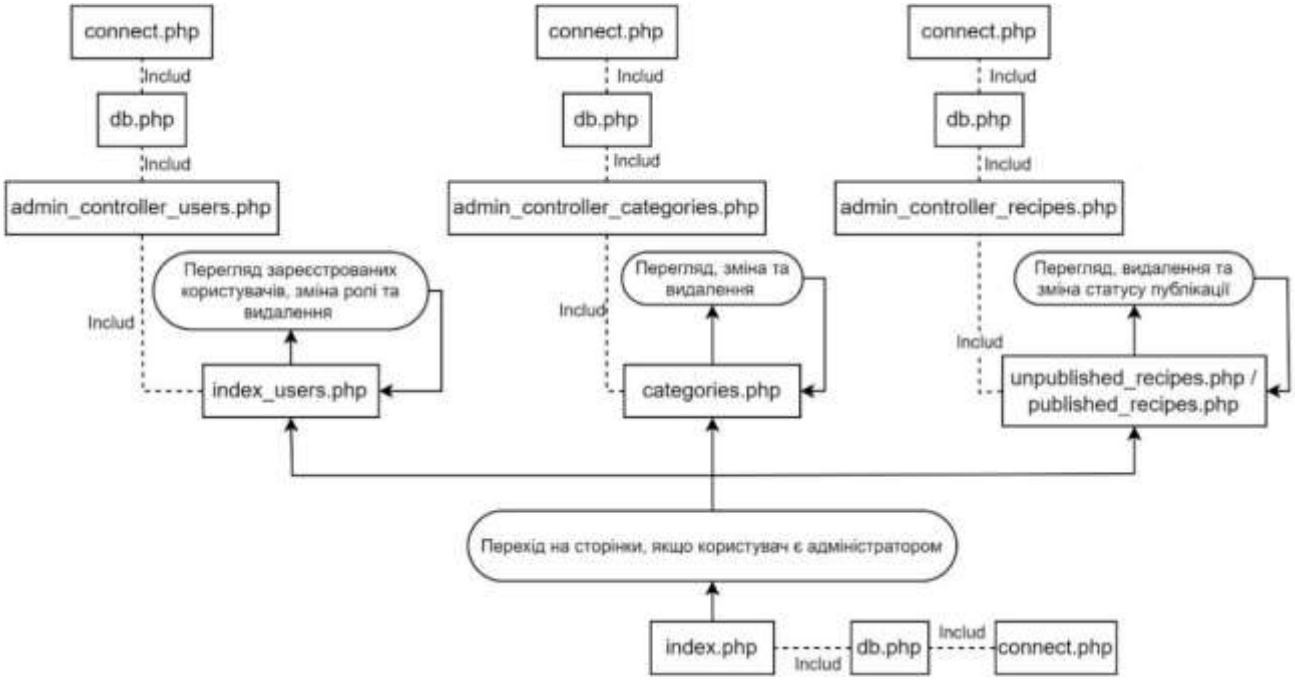

Рисунок 2.4 – Схема взаємодії модулів адміністративної частини вебсайту

У центрі адміністративної схеми розташований файл index.php, який після підключення службових залежностей перевіряє права користувача. Якщо користувач має статус адміністратора, він отримує доступ до розділів керування користувачами, категоріями та рецептами.

Розділ користувачів працює через index_users.php і контролер admin_controller_users.php. Він дозволяє переглядати облікові записи, змінювати ролі та видаляти профілі. Розділ категорій обслуговується через categories.php і admin_controller_categories.php, які відповідають за створення, редагування й видалення рубрик. Модерація рецептів реалізується через сторінки published_recipes.php, unpublished_recipes.php і контролер admin_controller_recipes.php.

Усі адміністративні дії проходять через спільні механізми підключення до бази даних. Це забезпечує єдину точку контролю, підвищує безпеку та зменшує ймовірність SQL-ін’єкцій. Крім того, винесення логіки в контролери полегшує

внесення змін, оскільки оновлення окремого функціоналу не потребує переписування всієї адміністративної частини [17].

На рисунку 2.5 показано логіку взаємодії компонентів користувацького інтерфейсу.

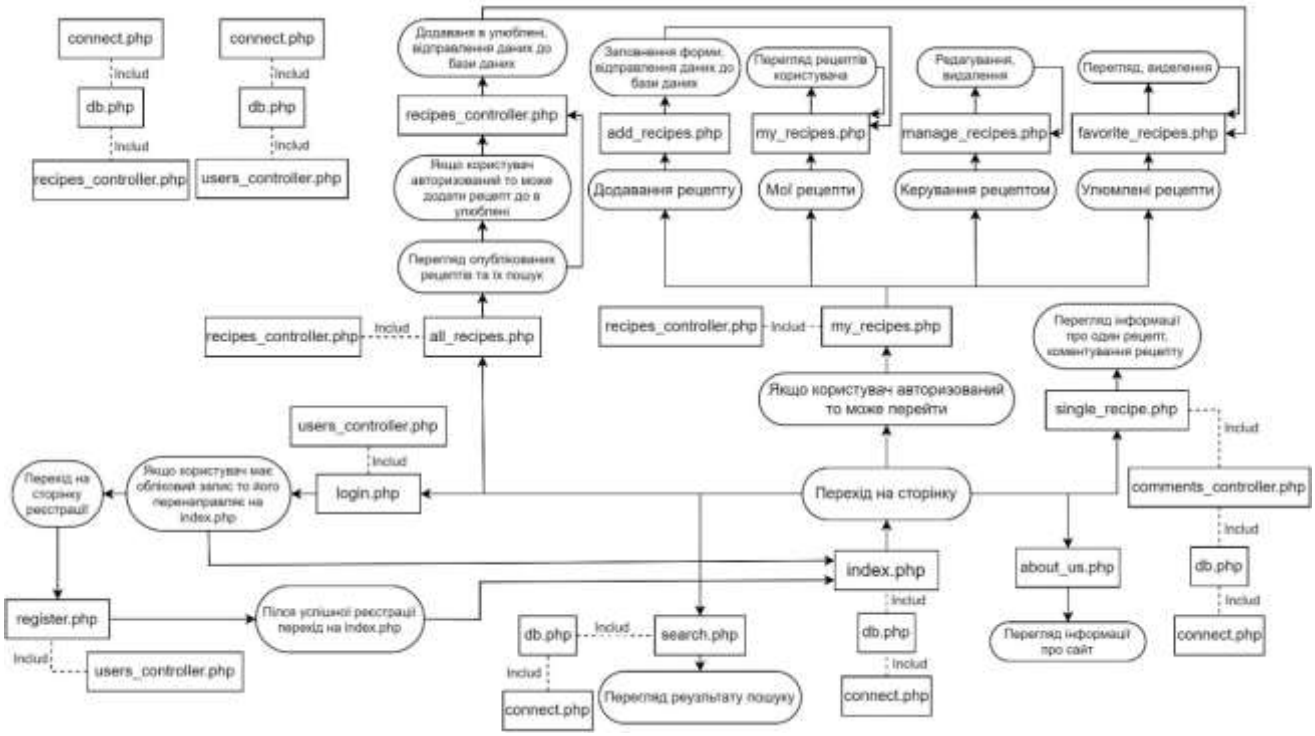

Рисунок 2.5 – Схема взаємодії компонентів інтерфейсу користувача вебсайту

У публічній частині вебсайту головну роль відіграють сторінки перегляду, пошуку та керування рецептами. Користувач може перейти з головної сторінки до каталогу, скористатися пошуком, відкрити окрему страву, зареєструватися або увійти в систему. Після авторизації йому стають доступними сторінки додавання, редагування та збереження рецептів.

Сценарій створення рецепта складається з кількох етапів. Спочатку користувач відкриває головну сторінку та проходить авторизацію. Якщо обліковий запис уже активний, система одразу перенаправляє його до персонального розділу. Далі автор натискає кнопку додавання рецепта й заповнює форму: вказує назву, інгредієнти, опис приготування, фото та інші параметри.

Після надсилання форми система виконує автоматичну перевірку. Якщо обов’язкові поля не заповнені або дані мають неправильний формат, користувач повертається до форми й виправляє помилки. Якщо перевірка успішна, рецепт переходить до черги модерації.

Подальше рішення ухвалює адміністратор. Він може схвалити рецепт, повернути його автору на доопрацювання або відхилити. Якщо матеріал відповідає правилам, він публікується в загальному каталозі та стає доступним для пошуку, коментування й додавання до вибраного. Якщо рецепт потребує уточнень, автор отримує можливість внести зміни й повторно подати його на перевірку. Якщо ж запис не відповідає тематиці або містить заборонений контент, він видаляється.

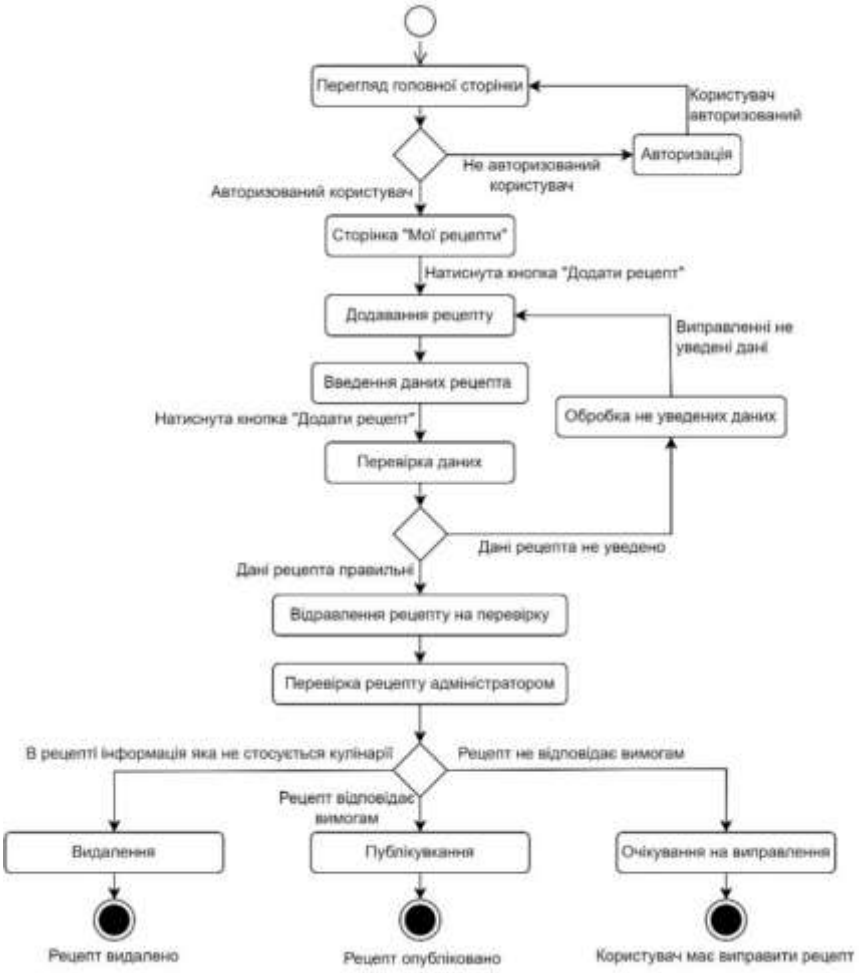

Рисунок 2.6 – Схема поетапного створення та публікації рецепта на вебсайті

Такий сценарій забезпечує контроль якості контенту та водночас залишає автору зрозумілий механізм виправлення помилок. У результаті платформа підтримує впорядкованість матеріалів і створює прозорий процес публікації.

2.5 Побудова діаграми класів вебсайту

Діаграма класів є засобом статичного моделювання, який показує основні об’єкти системи, їхні атрибути, методи та зв’язки. На відміну від сценарних або послідовнісних діаграм, вона не описує перебіг подій у часі, а відображає внутрішню структуру програмної системи. Така модель допомагає розробникам зрозуміти логіку побудови вебсайту, правильно організувати код і спростити подальшу підтримку [19].

Класову модель вебсайту наведено на рисунку 2.7.

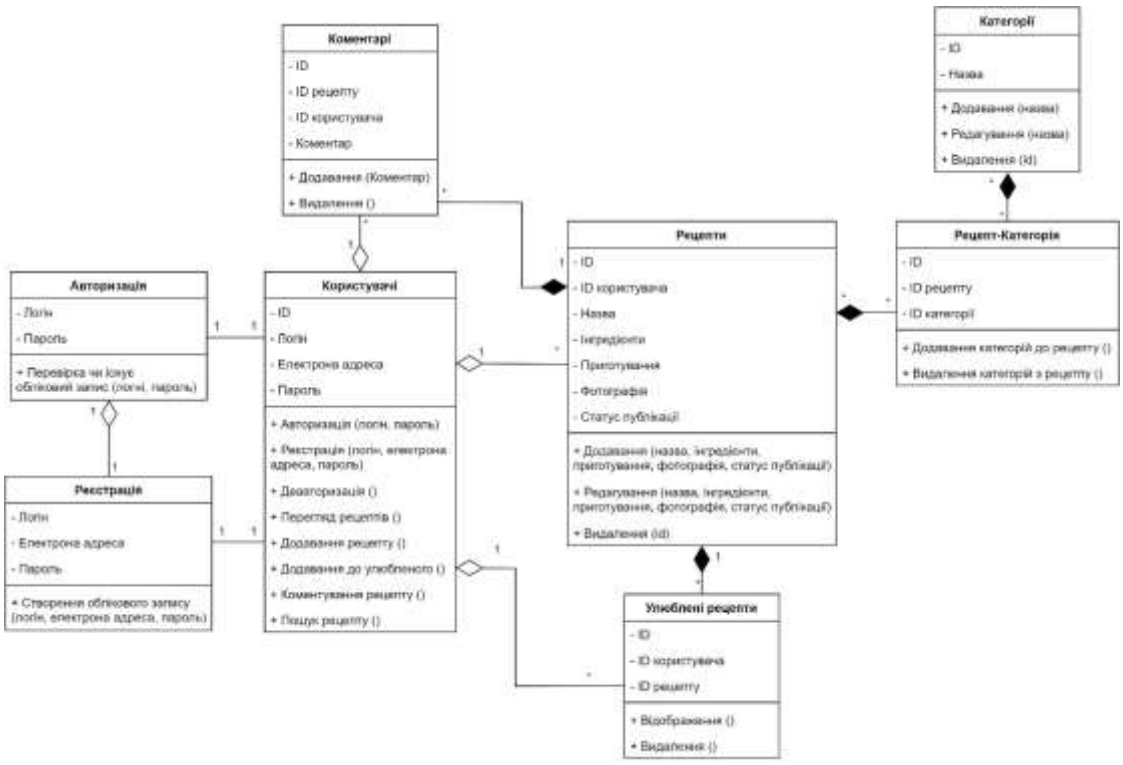

Рисунок 2.7 – Діаграма класів вебсайту

Одним із центральних елементів моделі є клас User, який описує обліковий запис користувача. Він містить ідентифікатор, логін, пароль, роль, дату створення профілю та інші службові дані. До методів цього класу належать

реєстрація, авторизація, редагування профілю, створення рецептів і додавання коментарів.

Клас Recipe описує кулінарну публікацію. Він містить назву, список інгредієнтів, кроки приготування, зображення, статус публікації та посилання на автора. Один користувач може створити багато рецептів, тому між класами User і Recipe реалізується зв’язок «один-до-багатьох».

Клас Registration відповідає за процедуру створення нового профілю. Він перевіряє коректність введених даних, унікальність email або логіна, формує хеш пароля та передає дані для збереження в базі. Клас Authorization працює з уже наявними обліковими даними, перевіряє пароль і створює активну сесію користувача.

Клас Comment використовується для збереження відгуків під рецептами. Кожен коментар належить одному користувачу й одному рецепту, але один рецепт може мати багато коментарів. Це дозволяє організувати обговорення страв і відстежувати активність учасників.

Клас SavedRecipe виконує роль проміжної сутності між користувачем і рецептом. Він зберігає інформацію про те, які рецепти користувач додав до вибраного. Такий підхід запобігає дублюванню даних і дає змогу швидко перевірити, чи збережена конкретна страва в персональній колекції.

Клас Category містить перелік кулінарних рубрик. Він має методи додавання, редагування та видалення категорій. Перед видаленням система може перевіряти, чи не пов’язані з категорією наявні рецепти.

Клас RecipeCategory реалізує зв’язок між рецептами й категоріями. Оскільки один рецепт може належати до кількох рубрик, а одна рубрика містить багато рецептів, цей клас забезпечує відношення «багато-до-багатьох».

Стрілки на діаграмі відображають типи асоціацій між класами та їхню кардинальність. Позначення «1» і «∞» показують, скільки об’єктів одного класу може бути пов’язано з об’єктами іншого класу. Завдяки такому поданню легко простежити, як користувачі, рецепти, категорії, коментарі та збережені матеріали взаємодіють між собою.

Отже, діаграма класів є важливим елементом проєктування, оскільки вона фіксує структуру майбутнього програмного продукту й допомагає уникнути неузгодженостей під час реалізації.

2.6 Формування файлової структури та ієрархії каталогів проєкту

Продумана файлова структура є важливою умовою якісної розробки вебпроєкту. Вона допомагає швидко знаходити потрібні модулі, розмежовує відповідальність між частинами системи та спрощує роботу над проєктом у команді. Якщо файли згруповано логічно, легше оновлювати окремі компоненти, додавати новий функціонал і виправляти помилки [20].

Для вебсайту було сформовано деревоподібну структуру каталогів, у якій окремо розміщено адміністративні сторінки, бізнес-логіку, автентифікацію, стилі, скрипти, зображення та шаблони рецептів. Схему файлової структури подано на рисунку 2.8.

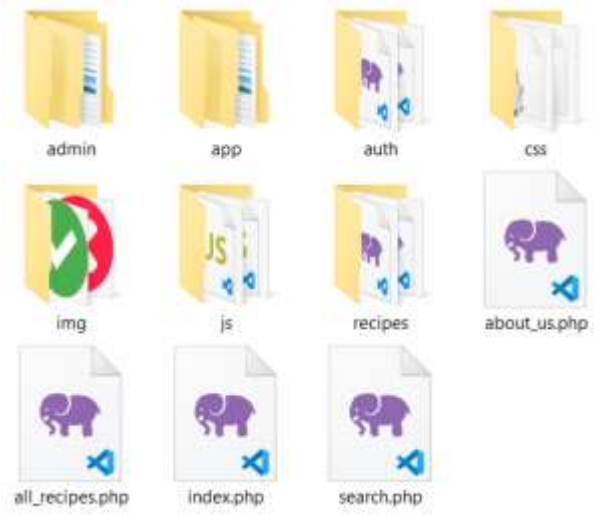

Рисунок 2.8 – Схема файлової структури

Каталог admin містить файли адміністративної частини. У ньому розміщуються сторінки та контролери для керування користувачами, категоріями й рецептами. Такий поділ дає змогу відокремити бек-офіс від публічної частини сайту.

Каталог app виконує роль ядра системи. У ньому зберігаються службові класи, моделі, конфігураційні файли та компоненти, що забезпечують роботу бізнес-логіки. Саме цей блок об’єднує основні механізми взаємодії з даними.

Папка auth призначена для сценаріїв автентифікації. Тут можуть розміщуватися файли реєстрації, входу, виходу із системи й відновлення пароля. Винесення цих функцій в окрему директорію робить структуру сайту зрозумілішою.

Каталог css містить таблиці стилів, які визначають зовнішній вигляд вебсайту. У ньому можуть зберігатися як базові CSS-файли, так і скомпільовані стилі з препроцесорів.

Папка img використовується для збереження зображень. Сюди можуть потрапляти фотографії страв, системні іконки та інші графічні ресурси. Для зручності резервного копіювання й упорядкування файли можна структурувати за датами або типами.

Каталог js містить клієнтські сценарії, які відповідають за інтерактивність вебсайту: фільтрацію, валідацію форм, асинхронний пошук, обробку подій і динамічне оновлення сторінок.

Папка recipes призначена для шаблонів і сторінок, пов’язаних із переглядом рецептів. У ній можуть бути окремі файли для відображення заголовка, списку інгредієнтів, інструкцій приготування, коментарів і схожих страв.

У корені проєкту розміщуються основні публічні файли: index.php, all_recipes.php, search.php і about_us.php. Файл index.php є початковою точкою входу, all_recipes.php виводить каталог рецептів, search.php обробляє пошукові запити, а about_us.php містить інформацію про платформу.

Такий розподіл файлів за функціональними зонами робить проєкт зручним для розроблення й супроводу. Кожен тип ресурсів має власне місце, тому зменшується ризик плутанини, дублювання коду та конфліктів під час внесення змін.

2.7 Висновок до другого розділу

У другому розділі було виконано концептуальне й технічне проєктування вебсайту для обміну кулінарними рецептами. Насамперед обґрунтовано використання трирівневої архітектури, яка розділяє систему на клієнтський рівень, прикладну логіку та рівень даних. Такий підхід забезпечує гнучкість, спрощує тестування й дозволяє масштабувати окремі компоненти незалежно один від одного.

Було розроблено навігаційну структуру вебсайту, у якій визначено основні розділи для гостей, зареєстрованих користувачів і адміністраторів. Завдяки цьому сформовано зрозумілу логіку переходів і розмежовано права доступу до функціоналу.

Окрему увагу приділено проєктуванню бази даних. Визначено основні сутності системи, зокрема користувачів, рецепти, категорії, коментарі та збережені рецепти. Запропонована модель забезпечує нормалізацію даних, підтримує референційну цілісність і створює умови для ефективного пошуку, фільтрації та подальшого розширення функціоналу.

Також було описано поведінкову модель вебсайту, роботу основних модулів і логіку взаємодії контролерів, сторінок та бази даних. Розглянуто сценарій створення рецепта — від авторизації користувача й заповнення форми до модерації та публікації матеріалу. Такий процес дозволяє підтримувати якість контенту й забезпечує прозорий механізм роботи з авторськими публікаціями.

Діаграма класів деталізувала об’єктну структуру системи та показала зв’язки між користувачами, рецептами, коментарями, категоріями й збереженими матеріалами. Це створює основу для подальшої реалізації програмної частини вебсайту.

Крім того, було сформовано файлову ієрархію проєкту, у якій окремо виділено адміністративну частину, бізнес-логіку, автентифікацію, статичні ресурси та публічні сторінки. Така організація полегшує супровід коду, командну роботу й подальше розширення системи.

Отже, у межах другого розділу сформовано цілісну технічну основу вебсайту. Запропоновані архітектурні, структурні та програмні рішення забезпечують можливість стабільної реалізації, безпечної експлуатації та поступового розвитку платформи для обміну кулінарними рецептами..

РОЗДІЛ 3. ПЕРЕВІРКА ЯКОСТІ, ВАЛІДАЦІЯ ТА ТЕСТУВАННЯ ВЕБСАЙТУ

3.1 Обґрунтування вибору хостингового середовища для розміщення вебсайту

Після завершення локальної розробки важливим етапом є перенесення вебсайту в середовище реальної експлуатації. Для цього необхідно обрати хостинг-платформу, яка забезпечить стабільну роботу ресурсу, достатню швидкість завантаження сторінок, захищене зберігання даних і можливість подальшого розширення. Від якості хостингу залежить не лише технічна працездатність сайту, а й загальне враження користувачів від взаємодії з платформою.

Хостинг можна розглядати як орендований серверний простір із налаштованим програмним середовищем, через яке вебсайт стає доступним у мережі Інтернет. Якщо провайдер має часті перебої, обмежені ресурси або застаріле серверне обладнання, це може призвести до повільного завантаження сторінок, помилок під час додавання рецептів, збоїв авторизації чи втрати довіри з боку користувачів. Тому під час вибору платформи враховувалися такі критерії: стабільність роботи, доступність технічної підтримки, підтримка бази даних, наявність SSL-сертифіката, простота керування файлами та можливість масштабування [21].

Для розгортання вебсайту було обрано платформу ProFreeHost. Це рішення є доцільним для стартової версії кулінарного вебсайту, оскільки сервіс надає базові можливості, достатні для розміщення MVP-продукту без значних фінансових витрат. Такий підхід особливо важливий на початковому етапі, коли основні ресурси доцільніше спрямовувати на розроблення функціоналу, тестування та покращення користувацького досвіду.

Серед переваг ProFreeHost варто відзначити підтримку PHP і MySQL, можливість роботи з базою даних через phpMyAdmin, доступ до FTP/FTPS для завантаження файлів, а також підключення SSL-сертифіката. Наявність

phpMyAdmin спрощує адміністрування бази даних: можна переглядати таблиці, виконувати резервне копіювання, імпортувати або експортувати дані та контролювати структуру записів без використання складних серверних інструментів.

Окремою перевагою є відсутність початкових витрат. Для навчального або демонстраційного проєкту це дозволяє швидко перевірити працездатність вебсайту в реальних умовах, оцінити стабільність роботи й підготувати платформу до подальшого розвитку. У разі збільшення кількості користувачів або навантаження на сервер передбачена можливість переходу на платний тариф із ширшими ресурсами [22].

Таким чином, ProFreeHost можна вважати виправданим вибором для першого розгортання вебсайту. Він забезпечує базову інфраструктуру, необхідну для роботи кулінарної платформи, дає змогу протестувати функціонал у реальному середовищі та залишає можливість подальшого масштабування.

3.2 Проведення валідації та тестування вебсайту

Після розміщення вебсайту на віддаленому сервері важливим етапом є перевірка його якості. Цей процес охоплює валідацію коду та тестування функціональних і нефункціональних характеристик системи. Обидва напрями доповнюють один одного: валідація підтверджує правильність технічної реалізації, а тестування демонструє, як система поводиться в реальних сценаріях використання.

Валідація спрямована на перевірку відповідності коду чинним вебстандартам. Для HTML і CSS застосовуються спеціалізовані валідатори, зокрема сервіси W3C, а для PHP-коду — статичні аналізатори, які дозволяють виявити синтаксичні помилки, некоректні виклики функцій, неініціалізовані змінні або потенційно небезпечні фрагменти. Якісний код зменшує ризик некоректного відображення сторінок у різних браузерах і спрощує подальший супровід проєкту [23].

Тестування вебсайту передбачає перевірку основних користувацьких сценаріїв: реєстрації, авторизації, перегляду рецептів, пошуку, фільтрації, додавання рецептів, збереження матеріалів до вибраного та роботи адміністративної частини. Окрім цього, виконуються перевірки адаптивності, кросбраузерної сумісності, стабільності роботи під навантаженням і базових аспектів безпеки. Метою такого комплексу є підтвердження того, що вебсайт відповідає функціональним вимогам, коректно працює на різних пристроях і не містить критичних помилок [24].

Для демонстрації процесу валідації було обрано головну сторінку вебсайту, реалізовану у файлі index.php. Код сторінки перевірено за допомогою онлайн-сервісу PHP Code Checker. Аналізатор виконав перевірку синтаксису, коректності викликів функцій, ініціалізації змінних і загальної відповідності коду правилам PHP.

Результат перевірки засвідчив відсутність помилок і попереджень. Це означає, що головна сторінка не містить критичних синтаксичних недоліків, які могли б призвести до некоректного виконання скрипта або збоїв у роботі ресурсу. Фрагмент звіту наведено на рисунку 3.1..

Рисунок 3.1 – Результати перевірки сторінки index.php засобом PHP Code Checker

На рисунку 3.1 показано результат аналізу файла index.php, у якому повідомлення «No errors or warnings found» підтверджує відсутність виявлених

помилок. Нульові значення в лічильниках помилок і попереджень свідчать про те, що код головної сторінки є придатним для подальшого використання в робочому середовищі.

Аналогічну перевірку було виконано для інших публічних сторінок і файлів адміністративної частини. Критичних зауважень не виявлено, що підтверджує коректність PHP-реалізації основних модулів вебсайту. Це знижує ймовірність появи помилок у продакшн-середовищі та підвищує передбачуваність роботи ресурсу.

Окремо було перевірено таблиці стилів за допомогою сервісу W3C CSS Validator. Перевірка стосувалася основних CSS-файлів, які відповідають за вигляд головної сторінки, адаптивну верстку та поведінку елементів інтерфейсу. Результати наведено на рисунку 3.2.

Рисунок 3.2 – Результати перевірки CSS-стилів головної сторінки засобом W3C CSS Validator

Звіт валідатора засвідчив відсутність синтаксичних помилок, некоректних значень властивостей, застарілих декларацій або проблем із медіазапитами. Це означає, що стилі відповідають вимогам CSS і мають коректно відображатися в сучасних браузерах. Такий результат є важливим для забезпечення стабільного вигляду вебсайту на різних пристроях і роздільних здатностях екрана.

Отже, проведена валідація підтвердила технічну коректність основних PHP- і CSS-файлів. Це створює підґрунтя для подальшого функціонального тестування та експлуатаційної перевірки системи.

Формальна правильність коду ще не гарантує однакового вигляду вебсайту в усіх браузерах. Тому важливим етапом стало тестування кросбраузерної сумісності. Його мета полягає в тому, щоб перевірити, чи однаково коректно відображаються сторінки, шрифти, сітки, кнопки, форми та JavaScript-елементи в різних браузерних середовищах [25].

Для перевірки було обрано чотири популярні браузери: Google Chrome, Microsoft Edge, Mozilla Firefox і Opera. Вони використовують різні або модифіковані рушії рендерингу, тому дають змогу оцінити поведінку сайту в найпоширеніших умовах. Перевірялися головна сторінка, каталог рецептів, форма додавання рецепта, сторінки авторизації та елементи особистого кабінету.

Тестування проводилося на кількох роздільних здатностях екрана, зокрема 1920×1080, 1366×768 і 375×667 px. Особливу увагу приділено роботі Flex- і Grid-контейнерів, поведінці навігаційного меню, коректності кнопок, відображенню карток рецептів, плавності анімацій і відсутності помилок у консолі браузера.

На рисунку 3.3 показано відображення головної сторінки у браузері Google Chrome.

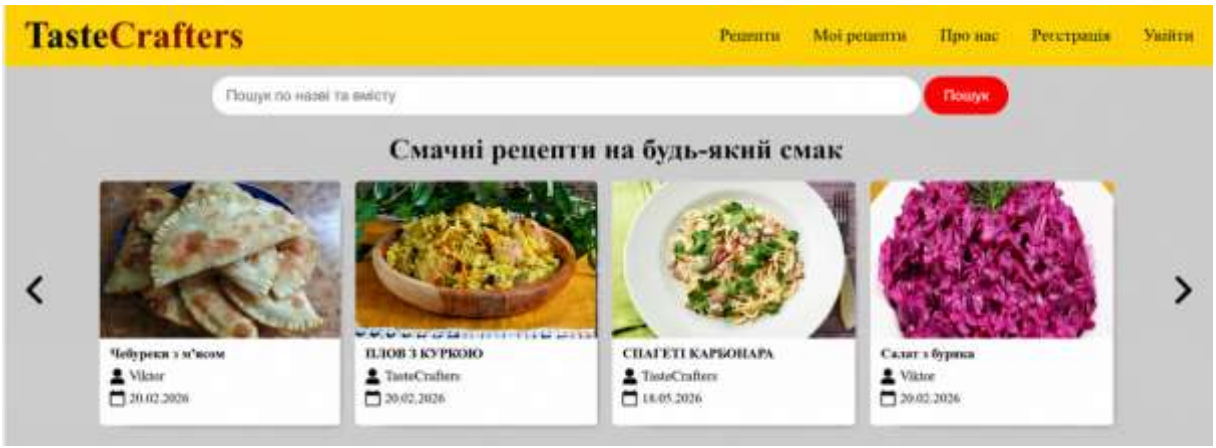

Рисунок 3.3 – Перевірка відображення головної сторінки вебсайту у браузері Google Chrome

У Chrome сторінка відобразилася відповідно до еталонного макета. Навігаційна панель, пошуковий блок, сітка рецептів і кнопки працювали коректно. JavaScript-обробники виконувалися без затримок, а консоль не містила

критичних помилок. Це підтвердило правильну роботу сайту в одному з найпоширеніших браузерів.

Результат перевірки в Microsoft Edge подано на рисунку 3.4.

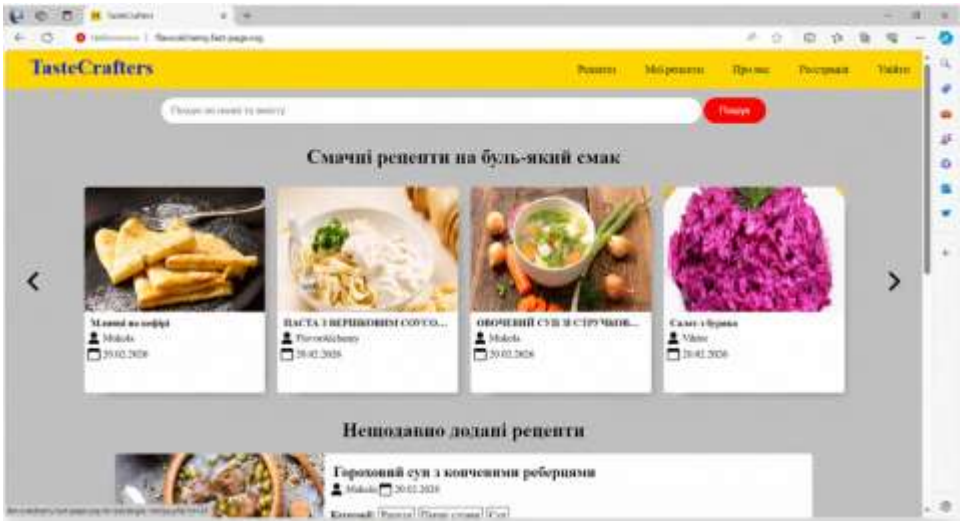

Рисунок 3.4 – Перевірка відображення головної сторінки вебсайту у браузері Microsoft Edge

У Microsoft Edge основні елементи інтерфейсу зберегли своє розташування, відступи та кольорову схему. Відмінностей у типографіці, ширині контейнерів або роботі інтерактивних компонентів не виявлено. Це свідчить про стабільне відтворення сайту в браузерах на основі рушія Blink.

На рисунку 3.5 продемонстровано роботу вебсайту в Opera.

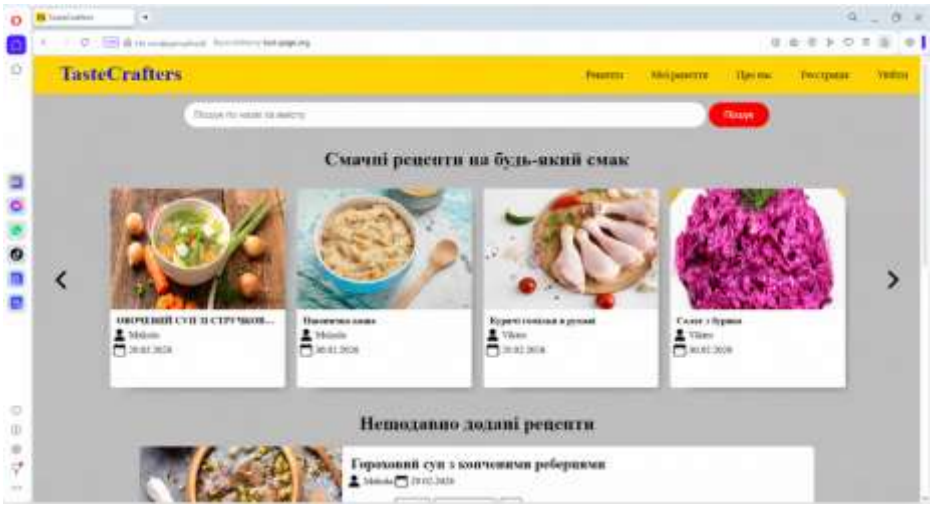

Рисунок 3.5 – Перевірка відображення головної сторінки вебсайту у браузері Opera

В Opera перевірялися завантаження зображень, робота випадних меню, плавність CSS-переходів і відсутність зсувів елементів під час взаємодії. Макет відтворився без помітних візуальних дефектів, а функції браузера, зокрема режим економії трафіку, не вплинули на працездатність сторінки.

Результат перевірки у Mozilla Firefox наведено на рисунку 3.6.

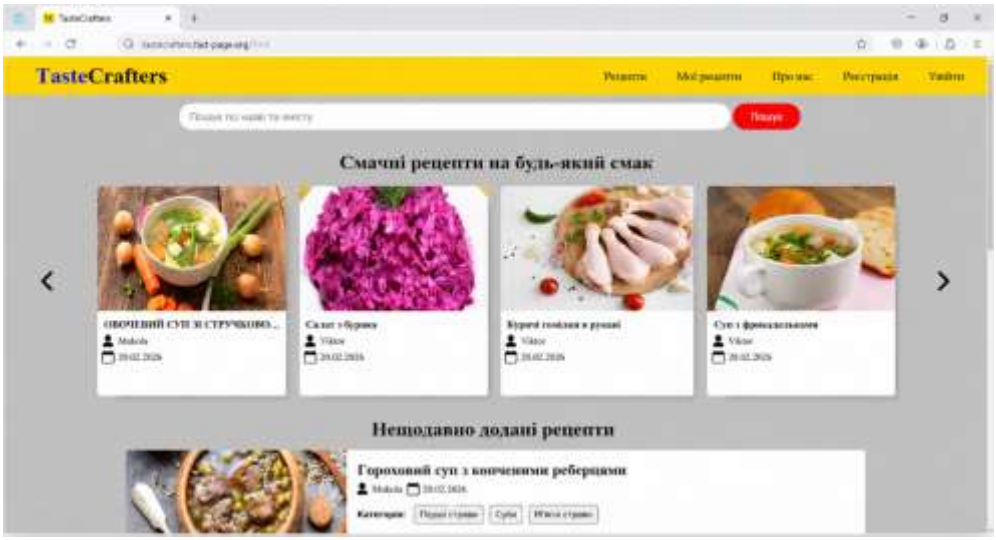

Рисунок 3.6 – Перевірка відображення головної сторінки вебсайту у браузері Mozilla Firefox

У Firefox коректно відобразилися CSS Grid-сітка, Flex-контейнери, навігаційне меню та блоки з рецептами. Також перевірено перемикання теми й роботу кешованих елементів. Порушень у відображенні або роботі інтерактивних компонентів не виявлено.

За результатами перевірки у Chrome, Edge, Opera та Firefox можна зробити висновок, що вебсайт має належний рівень кросбраузерної сумісності. Основні функції — авторизація, пошук, фільтрація, додавання до вибраного, коментування та перегляд рецептів — працюють однаково стабільно в усіх перевірених браузерах.

Адаптивність є однією з ключових вимог до сучасного вебсайту, оскільки значна частина користувачів взаємодіє з онлайн-сервісами через смартфони та планшети. Адаптивний дизайн передбачає, що макет автоматично підлаштовується під різні розміри екранів, зберігаючи зручність читання, доступність кнопок і стабільність розміщення елементів.

Для реалізації адаптивності у вебсайті використано медіазапити CSS із кількома брейкпоінтами, зокрема 576 px, 768 px, 992 px і 1200 px. Це дає змогу змінювати структуру сітки, розміри шрифтів, відступи та розташування блоків залежно від ширини екрана. Додатково застосовано підхід mobile first, за якого спочатку проєктується зручний вигляд для мобільних пристроїв, а потім інтерфейс розширюється для планшетів і настільних комп'ютерів.

Перевірку адаптивності виконано на двох смартфонах: Redmi Note 6 Pro та iPhone 11 Pro. На рисунку 3.7 показано роботу кількох ключових сторінок сайту на екрані Redmi Note 6 Pro.

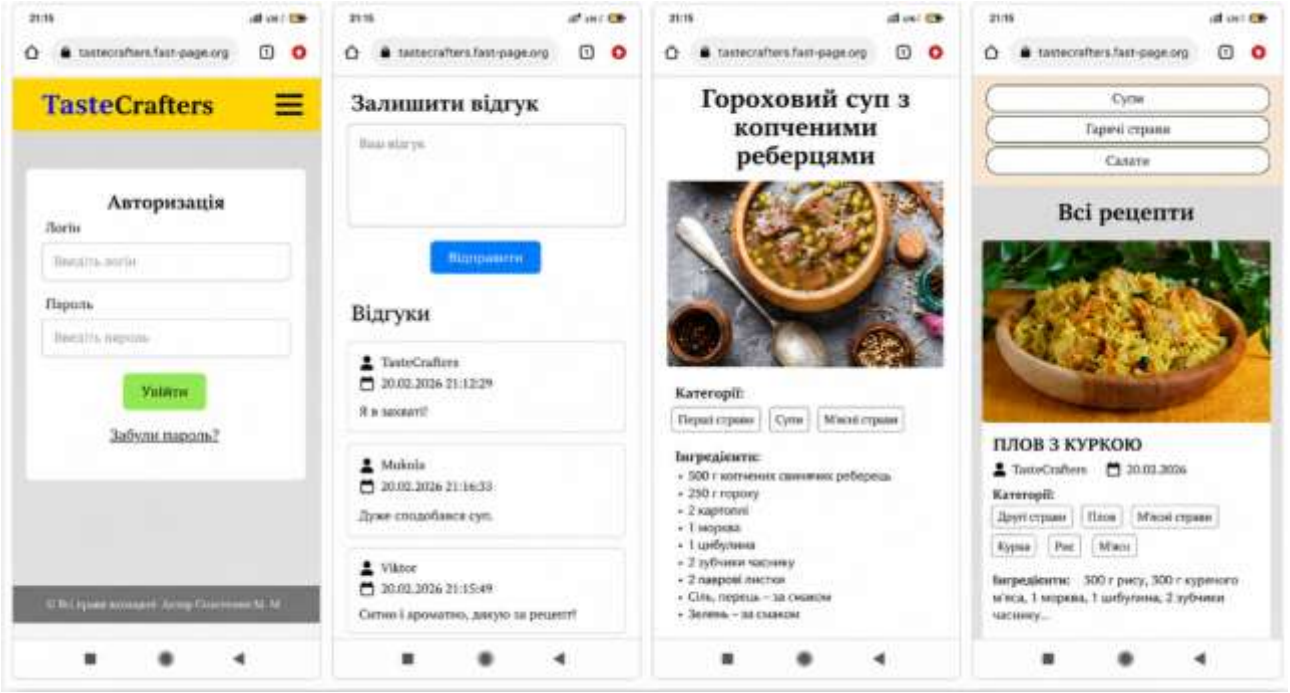

Рисунок 3.7 – Перевірка адаптивності ключових сторінок вебсайту на смартфоні Redmi Note 6 Pro

На пристрої Redmi Note 6 Pro сторінки відобразилися коректно: меню відкривалося у мобільному форматі, поля форм займали доступну ширину контейнера, кнопки залишалися зручними для натискання, а текст не виходив за межі екрана. Горизонтального прокручування або накладання блоків не зафіксовано.

На рисунку 3.8 наведено результат перевірки на iPhone 11 Pro.

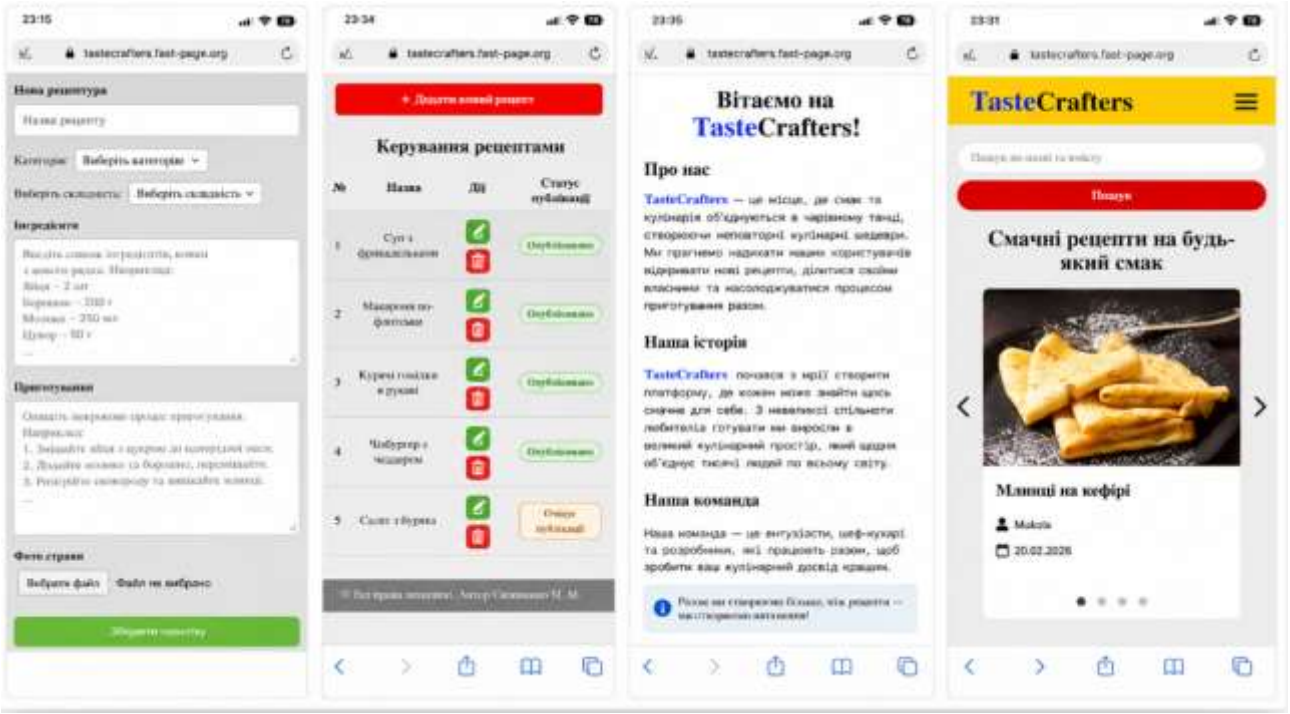

Рисунок 3.8 – Перевірка адаптивності ключових сторінок вебсайту на смартфоні iPhone 11 Pro

На iPhone 11 Pro інтерфейс також зберіг правильну структуру. Відступи, розміри тексту, кнопки та блоки з рецептами відповідали мобільному макету. Сторінки прокручувалися плавно, форми не виходили за межі екрана, а інтерактивні елементи залишалися доступними для користувача.

Результати перевірки підтвердили, що інтерфейс вебсайту коректно масштабується на різних мобільних пристроях. Відсутність обрізаних елементів, накладань і зайвого горизонтального скролу свідчить про правильне використання медіазапитів і гнучких одиниць вимірювання. Отже, сайт може зручно використовуватися як на смартфонах, так і на екранах більшого розміру.

3.3 Перевірка функціональної роботи та супровід вебсайту в умовах експлуатації

Етап експлуатації охоплює практичну перевірку роботи сайту після його розміщення на сервері. На цьому етапі оцінюється не лише технічна справність окремих модулів, а й загальна зручність користування ресурсом. До основних

напрямів перевірки належать робота головної сторінки, автентифікація, каталог рецептів, особистий кабінет користувача та адміністративна панель.

Метою експлуатаційної перевірки є підтвердження того, що вебсайт стабільно виконує свої функції, коректно реагує на дії користувачів, забезпечує збереження даних і дозволяє адміністратору контролювати якість контенту.

Головна сторінка є початковою точкою взаємодії користувача з платформою. Вона формує перше враження про вебсайт, демонструє основні можливості сервісу та спрямовує відвідувача до ключових розділів. Її вигляд подано на рисунку 3.9.

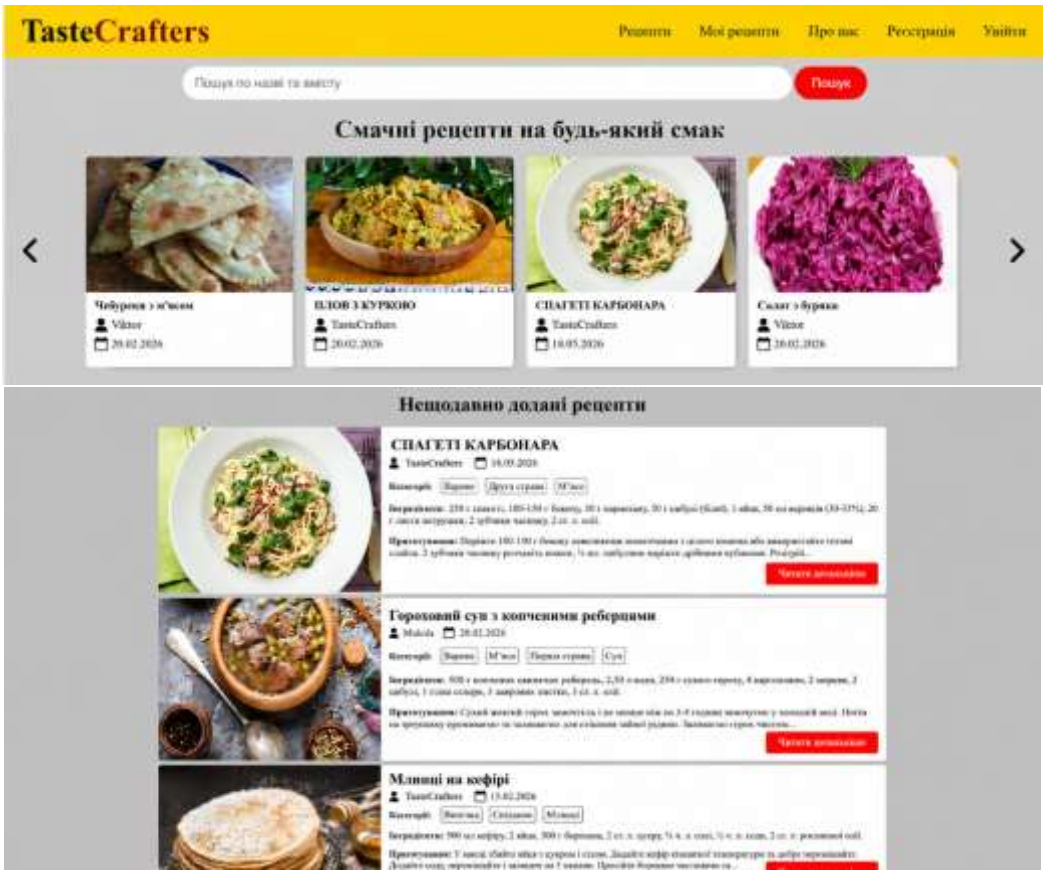

Рисунок 3.9 – Головна сторінка вебсайту

У верхній частині сторінки розташовано навігаційне меню, через яке користувач може перейти до каталогу рецептів, сторінки входу, реєстрації, особистого кабінету або інформаційного розділу. Меню адаптується до ширини екрана, тому залишається зручним як на комп’ютері, так і на мобільному пристрої.

Одним із центральних елементів головної сторінки є пошуковий блок. Він дає змогу швидко знайти потрібний рецепт за назвою, інгредієнтом або ключовим словом. Для підвищення зручності передбачено підказки, які допомагають користувачу швидше сформулювати запит.

Також на головній сторінці розміщено блок із рекомендованими або популярними рецептами. Картки містять назву страви, зображення, короткий опис, автора та дату публікації. Такий формат дозволяє швидко переглянути кілька варіантів і перейти до детальної сторінки рецепта.

Нижче може відображатися розділ із нещодавно доданими рецептами. Він демонструє, що платформа регулярно оновлюється, а користувачі можуть швидко знайти нові кулінарні ідеї. Кожна картка має кнопку для переходу до повного опису або додавання до вибраного.

Під час експлуатаційної перевірки головна сторінка завантажувалася стабільно, інтерактивні елементи працювали без помітних затримок, а структура сторінки залишалася коректною на різних екранах. Це підтверджує її готовність до використання як основного навігаційного центру вебсайту.

Модуль автентифікації забезпечує перехід користувача від статусу гостя до повноцінного учасника платформи. Він складається зі сторінок реєстрації та авторизації, які відповідають за створення облікового запису, вхід у систему та подальшу роботу з персональними функціями [29].

Сторінку реєстрації показано на рисунку 3.10.

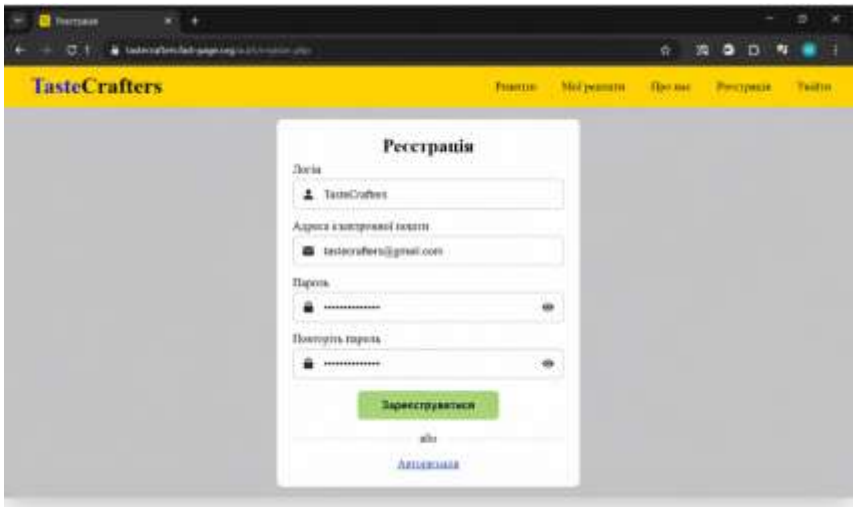

Рисунок 3.10 – Форма реєстрації нового користувача

Форма реєстрації містить поля для введення нікнейма, електронної пошти, пароля та підтвердження пароля. Також може бути передбачений чекбокс згоди з правилами користування платформою. Під час заповнення форми система перевіряє наявність обов'язкових даних, правильність email-адреси та збіг паролів.

Після надсилання форми дані проходять серверну перевірку. Система перевіряє унікальність логіна або електронної пошти, після чого пароль хешується та зберігається в базі даних у захищеному вигляді. Якщо всі дані коректні, створюється новий обліковий запис із роллю звичайного користувача. У разі помилки користувач отримує повідомлення з поясненням, які саме поля потрібно виправити.

Сторінку авторизації наведено на рисунку 3.11.

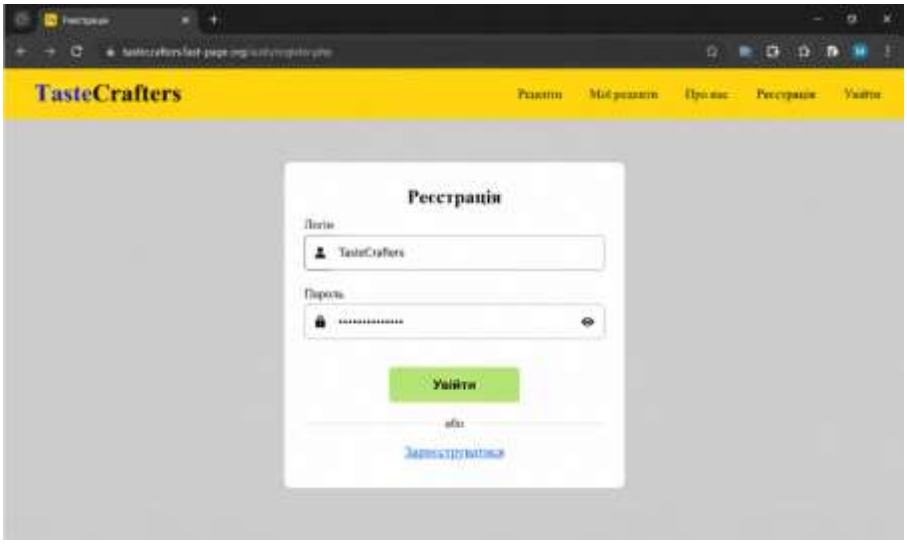

Рисунок 3.11 – Форма входу «Авторизація»

Форма входу містить поля для введення логіна або email-адреси та пароля. Після надсилання даних система шукає відповідний обліковий запис у базі та перевіряє пароль шляхом порівняння введеного значення з хешем. Якщо дані правильні, створюється активна сесія користувача, після чого він отримує доступ до особистого кабінету, додавання рецептів і збереження улюблених страв.

Для підвищення безпеки форми можуть містити CSRF-токени, а повторні невдалі спроби входу — обмежуватися. Паролі не зберігаються у відкритому вигляді, що зменшує ризик компрометації даних користувачів.

Під час тестування модуль реєстрації та авторизації успішно опрацював як позитивні, так і негативні сценарії. Порожні поля, некоректний email, дублікат облікового запису або неправильний пароль коректно оброблялися системою з виведенням відповідних повідомлень. Успішний вхід завершувався переходом до функцій авторизованого користувача.

Розділ «Рецепти» є основною публічною частиною вебсайту. У ньому користувачі переглядають опубліковані страви, шукають потрібні рецепти, застосовують фільтри та переходять до детального опису приготування. Фактично цей розділ виконує функцію електронної бібліотеки кулінарного контенту.

Інтерфейс каталогу подано на рисунку 3.12.

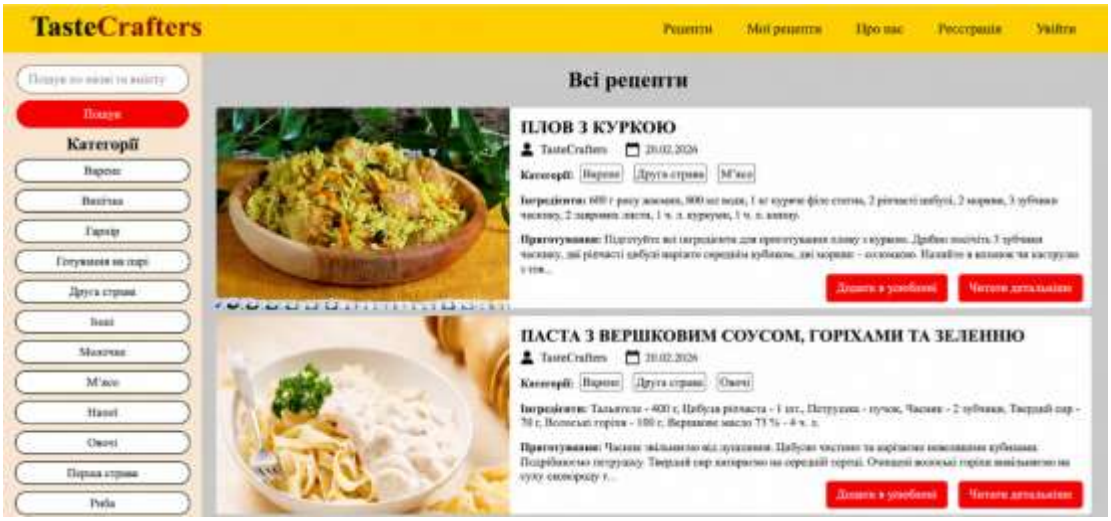

Рисунок 3.12 – Категорія «Рецепти» вебсайту

У лівій частині сторінки розташована панель фільтрації. Вона містить пошуковий рядок і перелік категорій, за якими можна звузити результати. Користувач може обрати потрібну рубрику, наприклад «М’ясо», «Десерти», «Сніданки» або «Вегетаріанські страви», після чого в основній області залишаються лише рецепти, що відповідають вибраному критерію.

Основна частина сторінки представлена плиткою карток рецептів. Кожна картка містить зображення страви, назву, короткий опис, автора та додаткові відомості. При натисканні на картку користувач переходить до повної сторінки

рецепта. Для авторизованих користувачів також доступна функція додавання рецепта до вибраного.

Механізм додавання до улюблених працює без повного перезавантаження сторінки. Після натискання на відповідну іконку система надсилає асинхронний запит, створює запис у таблиці saved_recipes і змінює стан іконки. Користувач одразу бачить результат дії, що покращує загальну зручність взаємодії.

Під час перевірки встановлено, що пошук повертає релевантні результати, фільтрація за категоріями працює коректно, а додавання до вибраного виконується без затримок. Каталог рецептів відповідає вимогам зручності, швидкодії та функціональності.

Після входу в систему користувач отримує доступ до особистого функціонального простору. У ньому можна створювати власні рецепти, редагувати опубліковані матеріали, переглядати статус публікацій і формувати добірку улюблених страв. Такий підхід перетворює користувача з пасивного читача на активного учасника кулінарної спільноти.

Сторінку додавання нового рецепта наведено на рисунку 3.13.

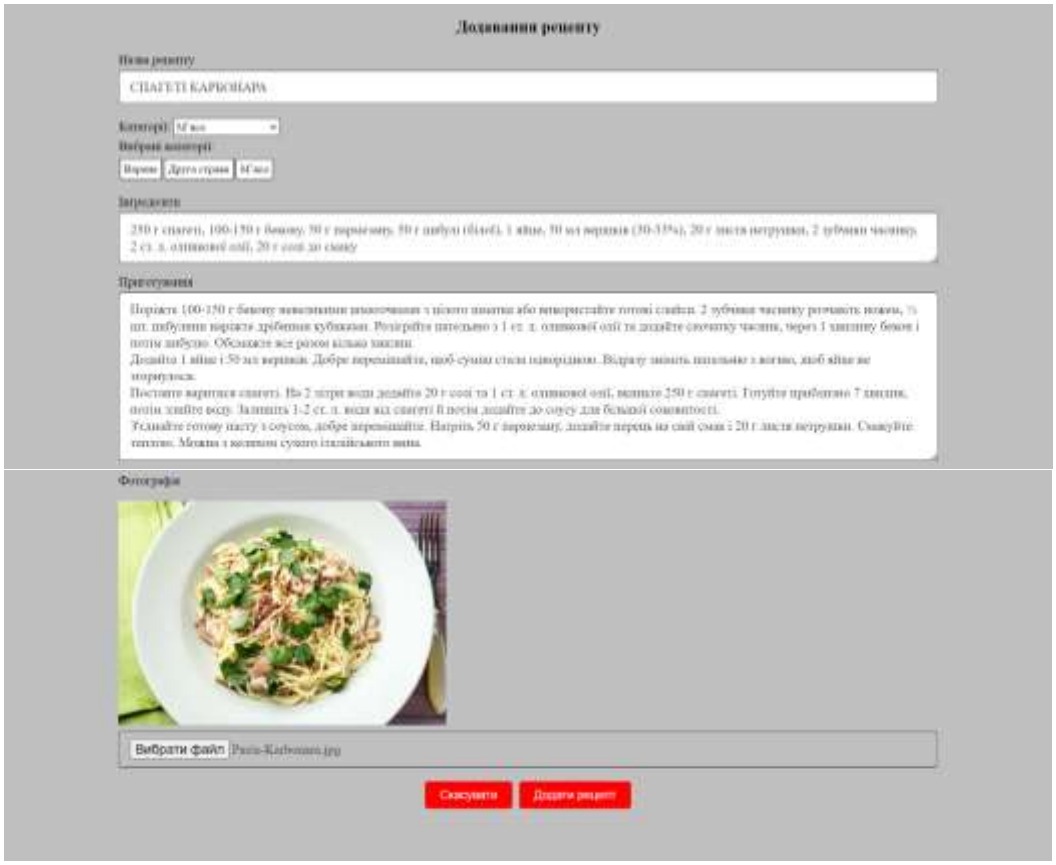

Рисунок 3.13 – Додавання нового рецепту

Форма додавання рецепта містить поля для назви страви, опису, інгредієнтів, кроків приготування, категорії та зображення. Для зручності користувача передбачено поділ інформації на логічні блоки. Інгредієнти можуть вводитися окремими рядками із зазначенням кількості та одиниці вимірювання, а інструкція приготування — у вигляді послідовних кроків.

Після завантаження зображення система перевіряє його формат, розмір і відповідність базовим вимогам. Дані форми передаються до контролера рецептів, який виконує валідацію, зберігає інформацію в базі даних і, за потреби, розміщує файл зображення у відповідній директорії. Після успішного додавання рецепт може бути переданий на модерацію.

На рисунку 3.14 показано сторінку керування рецептами з боку користувача.

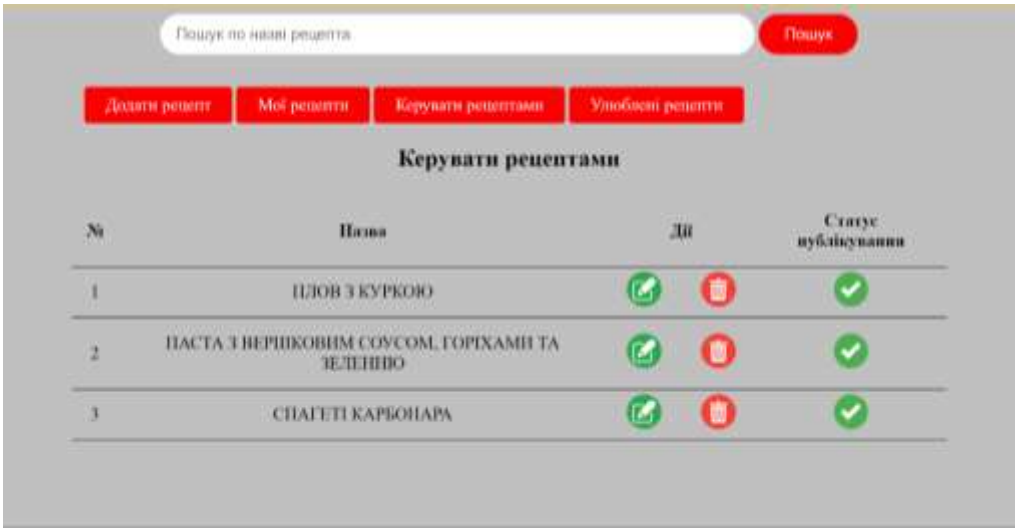

Рисунок 3.14 – Сторінка керування рецептами користувача

Цей розділ дає змогу автору переглядати власні публікації, редагувати їх або видаляти. Для кожного рецепта відображається назва, дата створення та статус. Якщо матеріал ще не схвалено адміністратором, користувач бачить відповідний індикатор. Після публікації статус змінюється, а рецепт стає доступним у загальному каталозі.

Користувач може повернутися до редагування, якщо потрібно виправити опис, додати деталі або замінити фотографію. Такий механізм дозволяє підтримувати актуальність і якість власного контенту.

Сторінку улюблених рецептів наведено на рисунку 3.15.

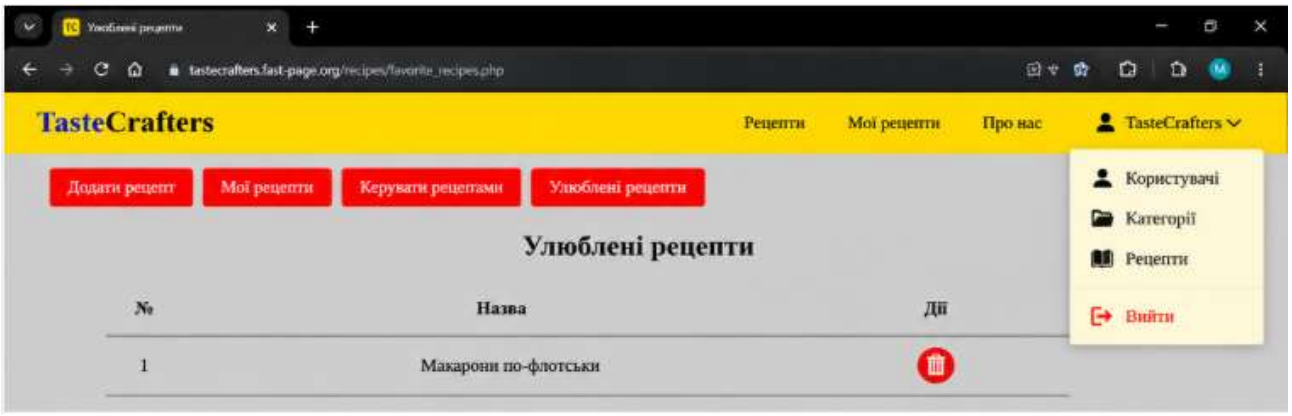

Рисунок 3.15 – Сторінка збережених рецептів користувача

Розділ «Улюблені рецепти» містить страви, які користувач зберіг для подальшого перегляду. Коли користувач натискає кнопку додавання до вибраного, у таблиці saved_recipes створюється зв’язок між його обліковим записом і відповідним рецептом. Після цього страва відображається в особистій добірці.

Кожна картка в цьому розділі має кнопку для видалення з вибраного. Це дозволяє користувачу самостійно керувати своєю колекцією й залишати лише актуальні рецепти. Якщо список порожній, система може запропонувати переглянути популярні або нещодавно додані страви.

Отже, особистий кабінет забезпечує користувачеві повний набір інструментів для створення, редагування, відстеження й збереження кулінарного контенту.

Адміністративна панель є закритою частиною вебсайту, доступною лише користувачам із відповідними правами. Вона призначена для керування обліковими записами, категоріями та рецептами, а також для контролю якості матеріалів, що публікуються на платформі [30].

Після авторизації адміністратор отримує доступ до спеціального меню, у якому розміщено основні модулі: «Користувачі», «Категорії» та «Рецепти». Інтерфейс керування показано на рисунку 3.16.

Рисунок 3.16 – Інтерфейс керування модулями адміністративної панелі

Модуль «Користувачі» дає змогу переглядати список зареєстрованих профілів, бачити логін, електронну адресу, дату створення та роль користувача. Адміністратор може змінювати ролі, блокувати порушників або видаляти облікові записи. Це забезпечує контроль за спільнотою та дотриманням правил користування платформою.

Модуль «Категорії» використовується для керування кулінарними рубриками. Адміністратор може створювати нові категорії, перейменовувати наявні або видаляти ті, що втратили актуальність. Перед видаленням система має перевіряти, чи не пов’язані з категорією рецепти. Це дозволяє уникнути помилок у структурі даних і зберегти цілісність каталогу.

Модуль «Рецепти» призначений для модерації контенту. Він поділяється на сторінки опублікованих і неопублікованих матеріалів. У розділі неопублікованих рецептів адміністратор переглядає нові записи, перевіряє їхній зміст, якість опису, відповідність тематиці та наявність зображень. Перелік рецептів, що очікують публікації, подано на рисунку 3.17.

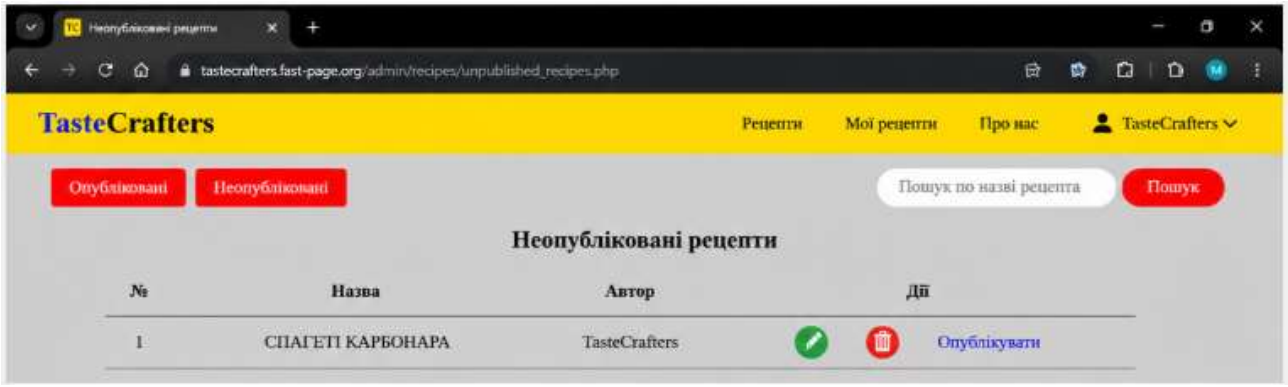

Рисунок 3.17 – Список рецептів, що очікують на публікацію

Якщо рецепт відповідає вимогам, адміністратор схвалює його, після чого запис переходить до загального каталогу. Якщо матеріал потребує виправлення, його можна повернути автору або відхилити. Такий підхід дозволяє підтримувати належний рівень якості публікацій і не допускати розміщення випадкового чи некоректного контенту.

На рисунку 3.18 показано перелік рецептів зі статусом «Опубліковано».

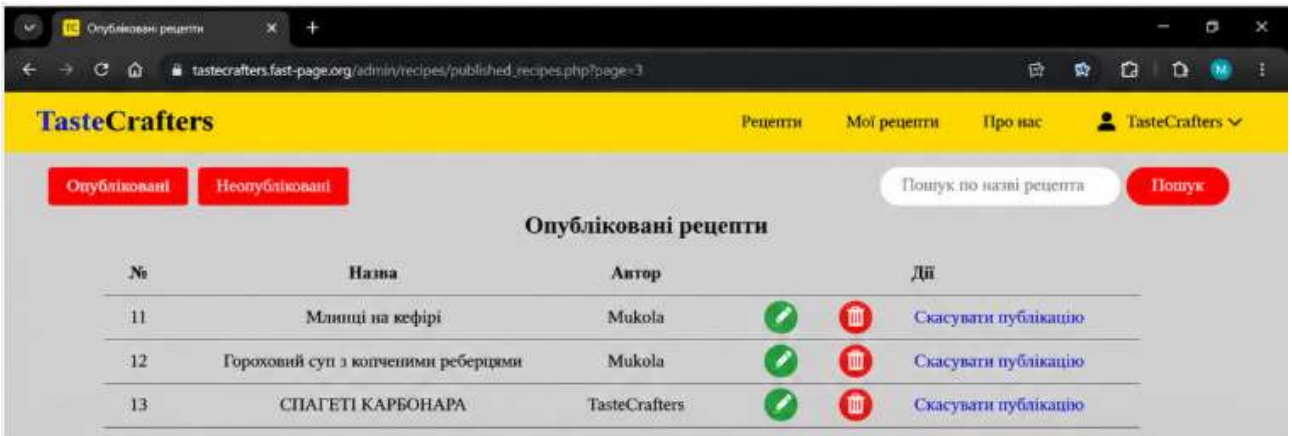

Рисунок 3.18 – Список опублікованих рецептів

У розділі опублікованих рецептів адміністратор може переглядати вже доступні матеріали, приховувати записи або вносити зміни в разі виявлення помилок. Це забезпечує постійний контроль над контентом навіть після його публікації.

Функціональна перевірка адміністративної панелі показала, що основні дії виконуються коректно. Керування користувачами, категоріями та рецептами відбувається без збоїв, а зміни в базі даних зберігаються правильно. Отже,

адміністративне середовище забезпечує необхідний рівень контролю над роботою вебсайту.

3.4 Висновок до третього розділу

У третьому розділі було виконано комплексну перевірку вебсайту після його розгортання в робочому середовищі. Насамперед обґрунтовано вибір хостинг-платформи ProFreeHost, яка забезпечує базові умови для розміщення стартової версії вебсайту: підтримку PHP і MySQL, доступ до phpMyAdmin, можливість підключення SSL-сертифіката та відсутність початкових фінансових витрат.

Проведена валідація PHP- і CSS-файлів підтвердила технічну коректність кодової бази. Перевірка за допомогою PHP Code Checker не виявила синтаксичних помилок або попереджень, а W3C CSS Validator підтвердив правильність таблиць стилів. Це свідчить про відповідність основних файлів вебстандартам і знижує ризик проблем під час експлуатації.

Кросбраузерне тестування у Google Chrome, Microsoft Edge, Opera та Mozilla Firefox показало, що вебсайт однаково коректно відображає основні сторінки та інтерактивні елементи. Навігація, пошук, фільтрація, перегляд рецептів, авторизація й додавання до вибраного працювали стабільно в усіх перевірених браузерах.

Адаптивне тестування на смартфонах Redmi Note 6 Pro та iPhone 11 Pro підтвердило правильну роботу мобільної версії сайту. Інтерфейс зберігав читабельність, елементи не накладалися один на одного, форми залишалися зручними для заповнення, а горизонтальне прокручування не виникало.

У межах експлуатаційної перевірки було проаналізовано роботу головної сторінки, сторінок реєстрації й авторизації, каталогу рецептів, особистого кабінету користувача та адміністративної панелі. Встановлено, що користувач може створювати обліковий запис, входити в систему, переглядати рецепти, додавати їх до вибраного, публікувати власні матеріали та відстежувати їхній

статус. Адміністратор, своєю чергою, має інструменти для керування користувачами, категоріями та модерацією рецептів.

Отже, результати валідації, кросбраузерної перевірки, адаптивного тестування та експлуатаційного аналізу підтвердили працездатність вебсайту. Розроблена платформа є технічно коректною, зручною для користувачів і придатною до подальшого розвитку.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Надання долікарської допомоги у разі ураження електричним струмом

Ураження електричним струмом може статися в будь-якому місці та в будь-який час. Незалежно від місця події, важливо знати основні правила надання долікарської допомоги, адже швидкі та правильні дії можуть врятувати життя. Також варто зауважити, що чим більший струм електричного удару, тим він потенційно смертельніший, тому вміння правильно реагувати у таких ситуація набуває особливого значення.

Будь-яка людина, яка зазнає дії електричного струму, не в змозі покликати на допомогу або самостійно визволитися від джерела струму. Контакт зі струмопровідними частинами часто викликає судоми м’язів, що заважає говорити або рухатися.

Ознаками ураження електричним струмом є:

- зупинка чи утруднене дихання;
- дискомфорт чи біль в грудях;
- дезорієнтація та сплутаність свідомості;
- відсутній чи нерегулярний пульс;
- судомні скорочення;
- втрата свідомості;
- опіки у місці контакту із джерелом струму [37].

У разі ураженні електричним струмом необхідно звільнити потерпілого від частин, які проводять струм. За можливості швидко відключити від мережі електрообладнання з яким контактує постраждалий, адже зволікання може призвести до її смерті. Щоб звільнити постраждалого від електропровідних частин необхідно використовувати діелектричні засоби захисту (рукавиці, килимки, боти та калоші) або сухі рукавиці, сухий одяг та палицю. Також варто використовувати інструменти, які мають ізольовані ручки для перерізання провідників.

Якщо відтягувати постраждалого від джерела струму за сухий одяг потрібно уникати контакту з металевими предметами та відкритими частинами тіла постраждалого. Якщо відтягувати постраждалого за ноги, то не потрібно контактувати з його взуттям, адже воно може стати провідником електричного струму. Також, той, хто надає допомогу, зобов'язаний мати одягнуті діелектричні рукавиці або обмотані руки матеріалом, який не проводить електричний струм для їх ізоляції. Вставання на гумовий килимок, суху дошку або непровідний матеріал того хто надає долікарську допомогу дозволяє ізолювати його.

Звільнивши постраждалого від дії електричного струму, йому необхідно надати першу медичну допомогу й викликати службу швидкої медичної допомоги. В наслідок дії електричного струму на організм людини, прийнято виділяти три стани:

1. Постраждалий у свідомості. В такому випадку необхідно забезпечити зручне положення й зберігати спокій для прибуття медичної допомоги. Важливо, щоб постраждалий не рухався і не проявляти фізичну активність, оскільки це може погіршити його стан.

2. Постраждалий втратив свідомість, але дихає. Як і в попередньому стані, необхідно поставити людину в стабільне положення, розстебнути одяг, який перешкоджає диханню, і забезпечити доступ свіжого повітря. Також можна спробувати привести постраждалого до тями, бризнувши йому на обличчя воду або давши понюхати нашатирний спирт.

3. Постраждалий не дихає або його дихання переривчасте. Цей стан є найсерйознішим порівняно з попередніми. В такому випадку потрібно негайно розпочати серцево-легеневу реанімацію, яка включає непрямий масаж серця та штучне дихання. Без цієї реанімації постраждалий може померти ще до прибуття швидкої допомоги. Перед тим, як виконувати штучне дихання, потрібно надати постраждалому правильне положення та перевірити ротову порожнину на наявність сторонніх тіл, крові або блювотних мас, і за необхідності очистити її [38].

На рисунку 4.1 зображено схематичне представлення виконання серцево-легеневої реанімації.

Рисунок 4.1 – Схематичне представлення виконання серцево-легеневої реанімації

Непрямий масаж серця слід розпочати якомога швидше. Натискання варто здійснювати в нижній частині грудини, при цьому глибина натискання має бути від 5 до 6 сантиметрів. Частота натискань має бути 100-120 на хвилину з мінімальною кількістю перерв. Після кожного натискання грудна клітка повинна повністю розпрямлятися, не допускається опертися на грудну клітку. Масаж слід виконувати на твердій поверхні.

Використання штучного дихання передбачає 30 чергових натискань на грудну клітку, за яким слідує проведення 2 вдихів. Якщо неможливо провести штучну вентиляцію легенів, то потрібно безперервно натискати на грудну клітку [39].

Лікаря слід викликати незалежно від стану постраждалого. Навіть якщо потерпілий прийшов до тями після удару електричним струмом й отримав першу допомогу. Важливо відвести потерпілого до лікарні для подальшого спостереження. Оскільки наслідки ураження можуть проявитися пізніше й мати серйозні наслідки.

Отже, надання першої допомоги при ураженні електричним струмом є надзвичайно важливим і потребує спеціальних навичок. Найкращим рішенням буде негайно викликати медичну допомогу для забезпечення безпечного та

ефективного лікування постраждалого. Крім того, працюючи з веб-сайтами на комп’ютерах варто дотримуватися правил безпеки, оскільки комп’ютер є джерелом електричної небезпеки. Ураження струмом може призвести до летальних наслідків, тому вчасне надання першої допомоги є критичним для збереження життя. Щоб уникнути уражень електричним струмом при роботі з комп’ютером, необхідно встановлювати захисні пристрої, які забезпечують ізольованість електропровідних частин.

4.2 Основні вимоги охорони праці та безпеки під час роботи користувачів із персональним комп’ютером

Перелік загальних вимог безпеки з охорони праці для користувачів ПК доволі великий. Перед початком роботи з ПК, потрібно пройти навчання з питань безпеки, включаючи інструктаж з охорони праці та пожежної безпеки.

Кожен користувач ПК повинен дотримуватися ряду правил для забезпечення безпеки та легальності використання. Перш за все, необхідно уникати доступу сторонніх осіб до ПК, що запобігає порушенням приватності та безпеки даних. Крім того, важливо використовувати лише ліцензійне програмне забезпечення, щоб уникнути порушень авторських прав та забезпечити надійну роботу системи. Не менш важливо дотримуватися вимог з охорони праці, що включає в себе правильну організацію робочого місця, уникнення перевантажень та забезпечення заходів безпеки при взаємодії з ПК.

Перед початком роботи за ПК для забезпечення безпеки охорони праці слід дотримуватися таких вимог:

- Протерти клавіатуру й екран монітора.
- Відстань від екрану до очей має бути 60-80 сантиметрів.
- Забезпечити, щоб пряме світло не впливало на екран ПК.
- Регулювати освітленість приміщення за допомогою сонцезахисних пристроїв.
- Провести налаштування положення робочого стільця, кут нахилу спинки, та екрану монітора.

- Підготувати робоче місце, переконатися, що воно чисте та безпечне.
- Переконатися, що всі пристрої (монітор, клавіатура та інші) правильно підключені до системного блоку.
- Переконатися, що в елементах ПК відсутні струмопровідні елементи.
- Перевірити стан ізоляції кабелів живлення та їх надійність.
- Переконатися у належному функціонуванні елементів керування.
- Перевірити надійність заземлення електричного устаткування.
- При виявлені будь-яких несправностей не вмикати ПК і повідомити відповідальну особу [40].

Під час користування ПК важливо дотримуватися основних вимог безпеки:

- Увімкнути ПК вимикачами на корпусах у послідовності: периферійне обладнання, монітор, системний блок.
- Відрегулювати яскравості монітору.
- При роботі з текстовою інформацією, використовувати чорні знаки на світлому фоні.
- Для нейтралізації статичної електрики підвищуйте вологість повітря за допомогою зволожувачів.
- Для зменшення навантаження роботи на ПК, потрібно розподіляти рівномірно, враховуючи її складність.
- Тривалість безперервної роботи за ПК не повинна перевищувати дві години, після чого слід зробити перерву на 15 хвилин.
- Забороняється захаращувати робоче місце сторонніми предметами.
- Забороняється самостійно розбирати та ремонтувати елементи ПК.
- Забороняється класти будь-які предмети на ПК та його елементи.
- Забороняється вимикати апаратуру під час роботи.
- Забороняється переключати з’єднувальні шнури при включеному живленні.
- У разі виявлення запаху горілого, потрібно вимкнути пристрій та звернутися до відповідальної особи.

Після закінчення роботи з ПК потрібно зберегти інформацію, при необхідності файли створити резервну копію даних. Потім слід вимкнути ПК та, за потреби відключити периферійні пристрої від електромережі. Після цього потрібно прибрати робоче місце.

Під час роботи на ПК можуть виникнути небезпечні ситуації, такі як коротке замикання, перевантаження блоку живлення, перегрівання, пожежа або поломка обладнання. У разі таких ситуацій необхідно негайно відключити ПК від електромережі та повідомити відповідальну особу. Забороняється допускати сторонніх осіб у небезпечну зону. Якщо сталася аварія, важливо зберегти стан робочого місця та обладнання та повідомити відповідальній особі для отримання подальших інструкцій та запобігти подібним ситуація в майбутньому. У разі пожежі потрібно повідомити відповідальну особу, викликати рятувальну службу та, якщо можливо, гасити пожежу за допомогою вогнегасників, але лише після відключення обладнання від електромережі. У разі наявності потерпілих потрібно надати долікарську допомогу та викликати швидку медичну допомогу [41].

4.3 Висновок до четвертого розділу

В даному розділу кваліфікаційної роботи описано долікарську допомогу при ураженні електричним струмом. Проаналізовано ознаки ураження електричним струмом, стани в яких може перебувати потерпілий та процес надання першої медичної допомоги потерпілому.

Проаналізовано вимоги безпеки охорони праці для користувачів перед початком роботи, під час виконання роботи та після завершення виконання роботи за ПК. Дотримання цих вимог дозволяє запобігти виникненню різноманітних небезпечних ситуацій.

ВИСНОВКИ

У межах кваліфікаційної роботи було досліджено, спроєктовано, реалізовано та протестовано вебсайт для обміну кулінарними рецептами. Отримані результати підтвердили доцільність створення україномовної онлайн-платформи, яка поєднує рецептурний контент, соціальну взаємодію користувачів і можливості персоналізованого використання сервісу.

У першому розділі розглянуто предметну область і визначено основні потреби потенційної аудиторії. Встановлено, що користувачі зацікавлені в зручних, мобільних і локалізованих кулінарних ресурсах, які дають змогу не лише переглядати рецепти, а й зберігати їх, коментувати, оцінювати та публікувати власні матеріали. Також було проаналізовано наявні сервіси, визначено їхні переваги й недоліки, сформовано функціональні та нефункціональні вимоги до майбутнього вебсайту. Окремо описано ролі користувачів: гостя, зареєстрованого користувача, автора й адміністратора, що дало змогу чітко розмежувати права доступу та сценарії взаємодії із системою.

У другому розділі виконано концептуальне й технічне проєктування вебсайту. Для реалізації обрано трирівневу архітектуру, яка розділяє клієнтську частину, прикладну логіку та рівень роботи з даними. Такий підхід спрощує підтримку, тестування й подальше масштабування системи. Було розроблено структуру бази даних із ключовими сутностями: користувачі, рецепти, категорії, коментарі та збережені рецепти. Запропонована модель мінімізує дублювання даних і забезпечує коректні зв’язки між об’єктами. Також побудовано діаграми, описано користувацькі сценарії, модульну організацію та файлову структуру проєкту, що створило цілісну основу для програмної реалізації.

У третьому розділі проведено валідацію, тестування та експлуатаційну перевірку вебсайту. Перевірка PHP- і CSS-файлів підтвердила технічну коректність коду та відповідність основним вебстандартам. Кросбраузерне тестування у Google Chrome, Microsoft Edge, Opera та Mozilla Firefox показало стабільне відображення сторінок і правильну роботу інтерактивних елементів. Адаптивність було перевірено на мобільних пристроях, де сайт зберіг зручність

перегляду, читабельність тексту та коректне розташування елементів інтерфейсу. Крім того, протестовано основні сценарії роботи: реєстрацію, авторизацію, перегляд і фільтрацію рецептів, додавання матеріалів до вибраного, публікацію рецептів, керування особистим кабінетом та адміністрування контенту.

У четвертому розділі розглянуто питання безпеки життєдіяльності та охорони праці. Зокрема, описано порядок надання домедичної допомоги в разі ураження електричним струмом, а також наведено основні правила безпечної роботи за персональним комп’ютером. Увагу приділено організації робочого місця, правильному розташуванню монітора, освітленню, дотриманню перерв і використанню справного електрообладнання.

Отже, у результаті виконання роботи створено функціональний MVP вебсайту для обміну кулінарними рецептами. Розроблена платформа є зручною для користувачів, технічно працездатною та придатною до подальшого розвитку. У перспективі її можна розширити шляхом упровадження рекомендаційних алгоритмів, інтеграції з продуктовими сервісами, створення мобільного застосунку та вдосконалення персоналізованих функцій.

ПЕРЕЛІК ДЖЕРЕЛ

1. Пательня [Електронний ресурс]. Режим доступу до ресурсу: https://patelnya.com.ua/ (дата звернення: 03.06.2026).

2. Кукорама [Електронний ресурс]. Режим доступу до ресурсу: https://cookorama.net/uk/ (дата звернення: 03.06.2026).

3. Функціональні те нефункціональні вимоги [Електронний ресурс]. Режим доступу до ресурсу: https://visuresolutions.com/uk/requirements-management-traceability-guide/functional-vs-non-functional-requirements/ (дата звернення: 03.06.2026).

4. What is Use Case Diagram? [Електронний ресурс]. Режим доступу до ресурсу: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/ (дата звернення: 03.06.2026).

5. Варіанти використання та сценарії (Use Cases and Scenarios) [Електронний ресурс]. Режим доступу до ресурсу: https://www.maxzosim.com/use-cases-and-scenarios/ (дата звернення: 04.06.2026).

6. Методи розробки сайтів [Електронний ресурс]. Режим доступу до ресурсу: https://webstudio2u.net/ua/webdesign/354-site-develop-methods.html (дата звернення: 04.06.2026).

7. Створюємо сайт з нуля: шаблон чи індивідуальна розробка? [Електронний ресурс]. Режим доступу до ресурсу: https://rubarbs.com/ua/article/create-a-website-from-scratch-a-template-or-individual-development (дата звернення: 04.06.2026).

8. Етапи життєвого циклу розробки ПЗ [Електронний ресурс]. Режим доступу до ресурсу: https://icstudio.online/post/etapi-zhittyevogo-ciklu-rozrobki-pz (дата звернення: 04.06.2026).

9. Програмні системи створення веб-сайтів, CMS [Електронний ресурс]. Режим доступу до ресурсу: http://www.znannya.org/?view=WebDev (дата звернення: 05.06.2026).

10. Створення сайту на PHP [Електронний ресурс]. Режим доступу до ресурсу: https://itstatti.in.ua/stvorennya-sajtiv/18-stvorennya-sajtu-na-php.html (дата звернення: 05.06.2026).

11. База даних MySQL [Електронний ресурс]. Режим доступу до ресурсу: https://promoter.net.ua/articles/baza-danix-mysql.html (дата звернення: 05.06.2026).

12. Знайомство з Visual Studio Code [Електронний ресурс]. Режим доступу до ресурсу: https://romul.name/blog/znayomstvo-z-visual-studio-code/ (дата звернення: 05.06.2026).

13. Архітектура веб-додатків [Електронний ресурс]. Режим доступу до ресурсу: https://medium.com/@IvanZmerzlyi/архітектура-веб-додатків-ca4c82f75bcf (дата звернеуня: 06.06.2026).

14. Структура сайту: основні види та правила їх розробки [Електронний ресурс]. Режим доступу до ресурсу: https://webtune.com.ua/statti/web-rozrobka/struktura-sajtu/ (дата звернення: 06.06.2026).

15. Основні етапи проектування бази даних [Електронний ресурс]. Режим доступу до ресурсу: https://javarush.com/ua/quests/lectures/ua.questhibernate.level17.lecture01 (дата звернення: 06.06.2026).

16. Моделювання даних (Data Modelling) [Електронний ресурс]. Режим доступу до ресурсу: https://www.maxzosim.com/data-modelling/ (дата звернення: 07.06.2026).

17. Everything about Functional Block Diagrams [Електронний ресурс]. Режим доступу до ресурсу: https://edrawmax.wondershare.com/diagram-tips/function-block-diagram.html (дата звернення: 07.06.2026).

18. Діаграма станів [Електронний ресурс]. Режим доступу до ресурсу: https://vuzlit.com/1009781/diagrama_staniv (дата звернення: 07.06.2026).

19. Що таке діаграма класів UML і найкращий творець діаграм UML [Електронний ресурс]. Режим доступу до ресурсу: https://javarush.com/ua/quests/lectures/ua.questhibernate.level17.lecture01 (дата звернення: 08.06.2026).

20. Структурування каталогу: розкладемо все по поличках [Електронний ресурс]. Режим доступу до ресурсу: https://fractus.com.ua/uk/blog/strukturuvannya-katalogu-rozklademo-vse-po-polichkah/ (дата звернення: 08.06.2026).

21. Як правильно вибрати хостинг для сайту [Електронний ресурс]. Режим доступу до ресурсу: https://brainlab.com.ua/uk/blog-uk/dlya-chogo-potriben-hosting (дата звернення: 09.06.2026).

22. ProFreeHost – Free Hosting Review | Speed And Performance Analysis [Електронний ресурс]. Режим доступу до ресурсу: https://hexane.co.in/profreehost-free-hosting-review-speed-performance-analysis-alternatives/ (дата звернення: 09.06.2026).

23. Перевірка валідності сайту [Електронний ресурс]. Режим доступу до ресурсу: https://cityhost.ua/uk/blog/proverka-validnosti-sayta.html (дата звернення: 09.06.2026).

24. Тестування веб-проектів: основні етапи та поради [Електронний ресурс]. Режим доступу до ресурсу: https://qalight.ua/baza-znaniy/testuvannya-veb-proektiv-osnovni-etapi-ta-poradi/ (дата звернення: 09.06.2026).

25. Кросбраузерність – що це таке, і як її можна перевірити: огляд сервісів [Електронний ресурс]. Режим доступу до ресурсу: https://107.com.ua/blog/krosbrayzernist-sho-ce-take-i-iak-її-mojna-pereviriti-ogliad-servisiv/ (дата звернення: 10.06.2026).

26. Перевірка адаптивності сайту за допомогою браузера [Електронний ресурс]. Режим доступу до ресурсу: https://webtune.com.ua/statti/internet-marketing/yak-pereviryty-adaptyvnist-za-dopomogoyu-brauzera/ (дата звернення: 10.06.2026).

27. Як зробити пошук по сайту [Електронний ресурс]. Режим доступу до ресурсу https://www.zahidknyga.com.ua/instrukcii/kak-sdelat-poisk-po-sajtu.html (дата звернення: 10.06.2026).

28. Responsive Display [Електронний ресурс]. Режим доступу до ресурсу: https://kenwheeler.github.io/slick/ (дата звернення: 11.06.2026).

29. Як верифікувати користувача на сайту: дзвінки, SMS, електрона пошта [Електронний ресурс]. Режим доступу до ресурсу: https://cityhost.ua/uk/blog/yak-

verifikuvati-koristuvacha-na-sayti-dzvinki-sms-elektronna-poshta.html (дата звернення: 11.06.2026).

30. Що таке адмін-панель сайту та як туди зайти [Електронний ресурс]. Режим доступу до ресурсу: https://hostiq.ua/blog/ukr/admin-panel/ (дата звернення: 11.06.2026).

31. V. Kozlovskyi, Y. Balanyuk, H. Martyniuk, O. Nazarevych, L. Scherbak and G. Shymchuk, «Information Technology for Estimating City Gas Consumption During the Year,» 2022 International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, 2022, pp. 1-4, doi: 10.1109/SIST54437.2022.9945786.

32. Approach to gas consumption process forecasting on the basis of a mathematical model in the form of a random cyclic process / Serhii Lupenko, Iaroslav Lytvynenko, Oleg Nazarevych, Grigorii Shymchuk, Volodymyr Hotovych // ICAAEIT 2021, 15-17 December 2021. – Tern. : TNTU, Zhytomyr «Publishing house „Book-Druk“» LLC, 2021. – P. 213–219. – (Mathematical modeling in power engineering and information technologies).

33. Lytvynenko, S. Lupenko, O. Nazarevych, G. Shymchuk and V. Hotovych, «Mathematical model of gas consumption process in the form of cyclic random process,» 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), LVIV, Ukraine, 2021, pp. 232-235, doi: 10.1109/CSIT52700.2021.9648621.

34. Additive mathematical model of gas consumption process / Iaroslav Lytvynenko, Serhii Lupenko, Oleh Nazarevych, Hryhorii Shymchuk, Volodymyr Hotovych // Scientific Journal of TNTU. – Tern. : TNTU, 2021. – Vol 104. – No 4. – P. 87–97.

35. O. Nazarevych, Y. Leshchyshyn, S. Lupenko, V. Hotovych, G. Shymchuk and N. Shabliy, «Method of Gas Consumption Change-point Detection Based on Seasonally Multicomponent Model,» 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), Deggendorf, Germany, 2020, pp. 152-155, doi: 10.1109/ACIT49673.2020.9208924.

36. Y. Leshchyshyn, L. Scherbak, O. Nazarevych, V. Gotovych, P. Tymkiv and G. Shymchuk, «Multicomponent Model of the Heart Rate Variability Change-point,» 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), Polyana, Ukraine, 2019, pp. 110-113, doi: 10.1109/MEMSTECH.2019.8817379.

37. Панченко С. Електробезпека / С. Панченко, О. Акімов, М. Бабаєв, В. Блиндюк, В. Панченко, О. Супрун, Д. Сушко. – УкрДУЗТ, 2018. – 295 с. – ISBN 978-617-654-085-4.

38. Удар електричним струмом: перша допомога, наслідки після ураження [Електронний ресурс]. Режим доступу до ресурсу: https://www.uzhnu.edu.ua/uk/news/strum.htm (дата звернення: 13.06.2026).

39. Інструкція з охорони праці при роботі на персональному комп'ютері [Електронний ресурс]. Режим доступу до ресурсу: https://pro-op.com.ua/article/485-nstruktsya-z-ohoroni-prats-pri-robot-na-personalnomu-kompyuter (дата звернення: 14.06.2026).

40. Катренко А. Охорона праці в галузі комп'ютингу / А. Катренко, Л. Катренко. – Манголія 2006, 2026. – 544 с. – ISBN 978-617-574-049-1.

41. Шимчук, Г. В., Назаревич, О. Б., Литвиненко, Я. В., Готович, В. А., Никитюк, В. В., & Боднарчук, І. О. (2025). Грід-системи та технології хмарних обчислень. Навчальний посібник для здобувачів освітнього рівня «магістр» спеціальностей: F3 «Комп'ютерні науки», F6 «Інформаційні системи та технології».

42. Шимчук Г., Голотенко О., Небесний Р., Готович В. Застосування мови Scala у системах паралельних і хмарних обчислень. Наука і техніка сьогодні. 2026. № 4(58). С. 4794–4807. DOI: 10.52058/2786-6025-2026-4(58)-4794-4807.

43. Шевченко Н., Шимчук Г., Готович В., Голотенко О., Литвиненко С., Петрошук М. Математична модель для прогнозування змін у бездротових сенсорних мережах. Наука і техніка сьогодні. 2026. № 4(58). С. 4767–4782. DOI: 10.52058/2786-6025-2026-4(58)-4767-4782.

ДОДАТКИ

Додаток А

Таблиця реєстру варіантів використання вебсайту для акторів

Таблиця А.1 – Реєстр варіантів використання для акторів

Актор	Варіант використання	Призначення
1	**2**	**3**
Незареєстрований користувач	Перегляд опублікованих рецептів	Отримання інформації про опубліковані рецепти на сайті
	Пошук рецептів	Пошук рецептів на сайті
	Реєстрація	Створення облікового запису для доступу до додаткових функцій
Зареєстрований користувач	Перегляд опублікованих рецептів	Отримання інформації про опубліковані рецепти на сайті
	Пошук рецептів	Пошук рецептів на сайті
	Авторизація	Перевірка наявності певного користувача на основі логіну та паролю
Користувач	Перегляд опублікованих рецептів	Отримання інформації про опубліковані рецепти на сайті
	Пошук рецептів	Пошук рецептів на сайті
	Додавання рецептів	Додавання власних рецептів
	Видалення рецептів	Видалення власних рецептів
	Додавання до улюблених	Додавання рецептів до списку улюблених
	Коментування рецепту	Залишення коментарів під рецептами
	Вихід з облікового запису	Вихід з облікового запису користувача, завершення сесії
Адміністратор	Перегляд опублікованих рецептів	Отримання інформації про опубліковані рецепти на сайті

Продовження таблиці А.1

1	2	3
Адміністратор	Пошук рецептів	Пошук рецептів на сайті
	Додавання рецептів	Додавання власних рецептів
	Видалення рецептів	Видалення власних рецептів
	Додавання до улюблених	Додавання рецептів до списку улюблених
	Перегляд користувачів	Перегляд всіх зареєстрованих користувачів
	Видалення облікового запису	Видалення облікового запису користувачів із сайту.
	Коментування рецепту	Залишення коментарів під рецептами
	Видалення коментарів рецепту	Видалення будь-яких коментарів під рецептом
	Перевірка рецептів	Перевірка рецептів на відповідність вимогам сайту
	Публікування рецепту	Публікація рецепту авторизованим користувачем
	Вихід з облікового запису	Вихід з облікового запису адміністратора

Додаток Б

Лістинг медіазапитів вебсайту

```
@media only screen and (max-width: 2560px){
    .last-recipes{width: 80%;}
    .recipe-preview{width: 70%; padding: 10px; float: right;}
    .recipe-img{width: 30%; height: 100%; float: left;}
    .img-right{height: 20px;}
}
@media only screen and (max-width: 1430px){
    .last-recipes{width: 100%; padding: 10px; }
    .recipe-last{width: 100%;}
    .preparation{display: -webkit-box; -webkit-line-clamp: 3; -
webkit-box-orient: vertical; overflow: hidden;}
    .ingredients{display: -webkit-box; -webkit-line-clamp: 2; -
webkit-box-orient: vertical; overflow: hidden;}
}
@media only screen and (max-width: 768px){
    .recipe-last{height: auto;}
    .recipe-img{width: 100%;}
    .recipe-preview{width: 100%;}
    .read-more{position: static; display: block; width: 100%;
text-align: center;}
    .recipe-slider .next{right: 10px;}
    .recipe-slider .prev{left: 10px;}
    .search form {display: flex; flex-direction: column; align-
items: center; justify-content: center;}
    .search input{width: 100%; margin-bottom: 10px; height: 40px;
padding-left: 15px;}
    .search button{height: 40px; width: 100%;}
    .footer-text{font-size: 1em;}
    .slider-title{font-size: 30px;}
    .last-recipes-title{font-size: 30px;}
}
@media only screen and (max-width: 425px){
    .recipe-slider .next{right: 5px;}
    .recipe-slider .prev{left: 5px;}
}

@media only screen and (max-width: 1200px){
    .recipes_content{padding: 5px 20px;}
}
@media only screen and (max-width: 900px){
    .action-links a {display: flex; flex-direction: column;}
    .image-upload{width: 60%;}
    .recipes_content{padding: 5px 10px;}
    .content {width: 100%; margin: 15px auto;}
}
@media only screen and (max-width: 850px) {
    .edit_users{margin: 0 20%;}
    th, td {padding: 10px;}
}
@media only screen and (max-width: 800px) {
    .btn{display: flex; flex-direction: column; text-align:
center; margin-bottom: 10px; width: 100%; margin-right: 0;}
```

```
    .btn-group-add{display: flex; flex-direction: column; text-
align: center; justify-content: center; margin-left: 0;}
    .content{width: 100%; margin: 15px auto;}
    .search {margin-left: 0;}
    .action-links{display: flex; flex-direction: column; text-
align: center; justify-content: center;}
}
@media only screen and (max-width: 600px){
    .content{width: 100%;}
    .recipes_content{padding: 20px 10px 20px;}
    .btn-group-add{padding: 0; margin: 0;}
    .image-upload{width: 100%; padding: 0 5px;}
    .action-links{display: flex; flex-direction: column;}
    .action-links .delete{padding: 3px 0 3px 0;}
    .edit_users{margin: 0 5%;}
    th, td {padding: 5px; }
}
@media only screen and (max-width: 510px){
    .action-links a {display: flex; flex-direction: column; font-
size: 0.7rem;}
    th, td {padding: 5px;}
    .footer-text{font-size: 1em;}
    table{font-size: 1rem;}
}
@media only screen and (max-width: 425px) {
    .recipes_content{padding: 20px 3px 20px;}
    a{font-size: 0.9rem;}
    label{margin-left: 10px;}
    .text-input{width: 99%; margin-left: 2px;}
    .search form{display: flex; flex-direction: column;}
    .search input{width: 100%; margin-bottom: 10px;}
    .search button{width: 100%;}
    .search{width: 100%;}
    .action-links a img{height: 25px;}
    table{font-size: 0.6rem;}
    a{font-size: 0.6rem;}
}
@media only screen and (max-width: 375px){
    th, td {padding: 0.5px;}
    .action-links a {padding: 0 1px;}
}

@media only screen and (max-width: 1024px) {
    .recipes{width: 100%; padding: 10px;}
    .recipe{width: 100%;}
}
@media only screen and (max-width: 900px) {
    .content-wrapper{flex-direction: column;}
    .categories{width: 100%;}
    .categories-all{display: none;}
    .search form {display: flex; flex-direction: column; align-
items: center; justify-content: center;}
    .search input{width: 100%; margin-bottom: 10px; height: 40px;
padding-left: 15px; padding: 3%;}
    .search button{height: 40px; width: 100%; padding: 0; font-
size: 1.3rem;}
```

```
    .dowm{display: inline;}
}
@media only screen and (max-width: 768px) {
    .recipe{height: auto;}
    .recipe-img{width: 100%;}
    .recipe-preview{width: 100%;}
    .btn-group { position: static; bottom: 0; right: 0;}
    .btn{position: static; display: block; width: 100%; text-
align: center; margin-left: 0; margin-bottom: 10px;}
}
@media only screen and (max-width: 425px) {
    .footer-text{font-size: 1em;}
}

@media only screen and (max-width: 2560px) {
    .img-right{height: 20px;}
}
@media only screen and (max-width: 860px) {
    header{position: relative; z-index: 9999;    }
    header ul{width: 100%; background: #dcbb00; max-height: 0;
overflow: hidden;}
    .showing{max-height: 100em;}
    header ul li{width: 100%;}
    header ul li ul{ position: static; display: block; width:
100%; z-index: 88888;}
    header ul li ul li a{padding: 10px; background: #dcbb00;
padding-left: 50px;}
    header ul li ul li a:hover{background: #F0E68C;}
    .menu{display: block; position: absolute; right: 20px; top:
10px; font-size: 1.5em;}
    .logo{margin-left: 0.5em;}
    header ul li a{margin-left: 0;}
}

@media only screen and (max-width: 1200px){
    .recipes_content{
        padding: 30px 20px 50px;
    }
}
@media only screen and (max-width: 1024px) {
    .content_my_recipes{width: 100%; padding: 10px;}
    .recipe{width: 100%;}
    .btn-group-my{margin-left: 0;}
}
@media only screen and (max-width: 900px){
    .action-links a {display: flex; flex-direction: column;}
    .image-upload{width: 60%;}
}
@media only screen and (max-width: 800px) {
    .recipe{height: auto;}
    .recipe-img{width: 100%;}
    .recipe-preview{width: 100%;}
    .read-more{ position: static; display: block; width: 100%;
text-align: center;}
    .search form {display: flex; flex-direction: column; align-
items: center; justify-content: center; }
```

```
    .search input{width: 100%; margin-bottom: 10px; height: 40px;
padding-left: 15px;}
    .search button{height: 40px; width: 100%;}
    .btn{display: flex; flex-direction: column; text-align:
center; margin-bottom: 10px;}
    .btn-group-add, .btn-group-manage{margin-left: 0;}
    .content{width: 100%; margin: 15px auto;}
}
@media only screen and (max-width: 600px){
    .content{width: 100%;}
    .recipes_content{padding: 20px 10px 20px;}
    .btn-group-add, .btn-group-manage{padding: 0 10px;}
    .image-upload{width: 100%; padding: 0 5px;}
    .action-links{display: flex; flex-direction: column;}
    .action-links .delete{padding: 3px 0 3px 0;}
}
@media only screen and (max-width: 425px) {
    .footer-text{font-size: 1em;}
    .recipes_content{padding: 20px 3px 20px;}
    a{font-size: 0.9rem;}
    th, td{padding: 10px;}
    .footer-text{font-size: 1em;}
    label{margin-left: 10px;}
    .text-input{width: 99%; margin-left: 2px;}
}

@media only screen and (max-width: 768px) {
    .single-recipe-img{width: 100%;}
    .categories{width: 100%; padding: 20px 0 0 20px;}
    .ingredients{width: 100%; padding: 20px 0 0 20px;}
    .content, .comments{width: 90%;}
}
@media only screen and (max-width: 650px) {
    .content, .comments{width: 95%;}
    .comment-header{align-items: flex-start; flex-direction:
column; margin-bottom: 0.7rem;}
    .comment-header span{margin-bottom: 3px;}
}
@media only screen and (max-width: 450px){
    .content, .comments{width: 98%; padding: 10px 10px;}
    .single{padding: 20px 15px;}
    .footer-text{font-size: 1em;}
}
```

Додаток В

Лістинг швидкого меню вебсайту

```
<header>
    <a href="<?php echo BASE_URL ?>" class="logo">
        <h1 class="logo-text"><span class="span-
blue">Flavor</span>Alchemy</h1>
    </a>
    <img src="<?php echo BASE_URL ?>img/menu.png" alt="Меню"
height="40px" class="menu">
    <ul class="nav">
        <li><a href="<?php echo BASE_URL
?>all_recipes.php">Рецепти</a></li>
        <li><a href="<?php echo BASE_URL
?>recipes/my_recipes.php">Мої рецепти</a></li>
        <li><a href="<?php echo BASE_URL ?>about_us.php">Про
нас</a></li>
        <?php if(isset($_SESSION['id'])):?>
            <li>
                <a href="#">
                    <img src="<?php echo BASE_URL ?>img/user.png"
alt="Аватар" height="16px" class="img-right"><?php echo
$_SESSION['login']; ?>
                    <img src="<?php echo BASE_URL ?>img/down.png"
alt="Стрілка вниз" height="20px">
                </a>
                <ul>
                    <?php if ($_SESSION['admin']): ?>
                        <li><a href="<?php echo BASE_URL
?>admin/users/index_users.php">Користувачі</a></li>
                        <li><a href="<?php echo BASE_URL
?>admin/categories/categories.php">Категорії</a></li>
                        <li><a href="<?php echo BASE_URL
?>admin/recipes/published_recipes.php">Рецепти</a></li>
                    <?php endif; ?>
                    <li><a href="<?php echo BASE_URL
?>auth/logout.php" class="logout">Вийти</a></li>
                </ul>
            </li>
        <?php else: ?>
            <li><a href="<?php echo BASE_URL
?>auth/register.php">Реєстрація</a></li>
            <li><a href="<?php echo BASE_URL
?>auth/login.php">Увійти</a></li>
        <?php endif; ?>
    </ul>
</header>
```

Додаток Д

Лістинг скрипту слайдера

```
$('.recipe-body').slick({
    slidesToShow: 5,
    slidesToScroll: 1,
    autoplay: true,
    autoplaySpeed: 2000,
    nextArrow: $('.next'),
    prevArrow: $('.prev'),
    responsive: [
          {
               breakpoint: 2560,
               settings: {
                    slidesToShow: 5,
                    slidesToScroll: 5,
                    infinite: true,
                    dots: true
               }
          },
          {
               breakpoint: 1900,
               settings: {
                    slidesToShow: 4,
                    slidesToScroll: 4
               }
          },
          {
               breakpoint: 1440,
               settings: {
                    slidesToShow: 3,
                    slidesToScroll: 3
               }
          },
          {
               breakpoint: 800,
               settings: {
                    slidesToShow: 2,
                    slidesToScroll: 2
               }
          },
          {
               breakpoint: 500,
               settings: {
                    slidesToShow: 1,
                    slidesToScroll: 1
               }
          }
    ]
});
```

Додаток Е

Лістинг скрипту аутентифікації вебсайту

```
<?php include($_SERVER['DOCUMENT_ROOT'] . '/app/database/db.php');
$table = 'users';
define("BASE_URL", "http://flavoralchemy.fast-page.org/");
$admin_users = SELECT_ALL($table);
$errors = array();
$id = '';
$login = '';
$admin = '';
$email = '';
$password = '';
$repeat_password = '';
if(isset($_POST['register-btn'])){
    $errors = array();
    if(empty($_POST['login'])){
        array_push($errors, 'Введіть логін!');
    }elseif(strlen($_POST['login']) < 6 || strlen($_POST['login'])
> 20) {
        array_push($errors, 'Логін повинен містити від 6 до 20
символів!');
    }
    if(empty($_POST['email'])){
        array_push($errors, 'Введіть адресу електроної пошти!');
    }
    if(empty($_POST['password'])){
        array_push($errors, 'Введіть пароль!');
    }elseif(strlen($_POST['password']) < 4 ||
strlen($_POST['password']) > 20) {
        array_push($errors, 'Пароль повинен містити від 4 до 20
символів!');
    }
    if($_POST['repeat-password'] !== $_POST['password']){
        array_push($errors, 'Паролі не співпадають!');
    }
    $existingUser = SELECT_ONE('users', ['email' =>
$_POST['email']]);
    if (isset($existingUser)){
        array_push($errors, 'Така електрона адреса вже зайнята!');
    }
    $existingUser = SELECT_ONE('users', ['login' =>
$_POST['login']]);
    if (isset($existingUser)){
        array_push($errors, 'Такий логіні вже зайнятий!');
    }
    if(count($errors) === 0){
        unset($_POST['register-btn'], $_POST['repeat-password']);
        $_POST['password'] = password_hash($_POST['password'],
PASSWORD_DEFAULT);
        $_POST['admin'] = 0;
        $user_id = INSERT('users', $_POST);
        $user = SELECT_ONE('users', ['id' => $user_id]);

        $_SESSION['id'] = $user['id'];
```

```
            $_SESSION['login'] = $user['login'];
            $_SESSION['admin'] = $user['admin'];
            $_SESSION['message'] = "Реєстрація пройшла успішно!";
            $_SESSION['type'] = 'success';
            header('Location: '. BASE_URL);
            exit();
        }else{
            $login = $_POST['login'];
            $admin = isset($_POST['admin']) ? 1 : 0;
            $email = $_POST['email'];
            $password = $_POST['password'];
            $repeat_password = $_POST['repeat-password'];
        }
    }
    if (isset($_POST['login-btn'])){
        $errors = array();
        if(empty($_POST['login'])){
            array_push($errors, 'Введіть логін!');
        }
        if(empty($_POST['password'])){
            array_push($errors, 'Введіть пароль!');
        }
        if(count($errors) === 0){
            $user = SELECT_ONE('users', ['login' => $_POST['login']]);
            dd($user['password']);

            dd($_POST['password']);
            if ($user && password_verify($_POST['password'],
    $user['password'])){
                $_SESSION['id'] = $user['id'];
                $_SESSION['login'] = $user['login'];
                $_SESSION['admin'] = $user['admin'];
                $_SESSION['message'] = "Авторизація пройшла успішно!";
                $_SESSION['type'] = 'success';
                if ($_SESSION['admin']){
                    header('Location: '. BASE_URL);
                }else{
                    header('Location: '. BASE_URL);
                }
                exit();
            }else{
                array_push($errors, 'Неправильний логін або пароль!');
            }
        }
        $login = $_POST['login'];
        $password = $_POST['password'];
    }
    ?>
```

Додаток Ж

Лістинг сторінки “Рецепти”

```
<?php define("BASE_URL", "http://flavoralchemy.fast-page.org/");
include("./app/controllers/recipes_controller.php");
setlocale(LC_TIME, 'uk_UA.utf8');
if (!isset($_SESSION['id']) && isset($_GET['saved_recipe_id'])) {
    $_SESSION['message'] = 'Вам потрібно авторизуватися!';
    $_SESSION['type'] = 'error';
    header('Location: ' . BASE_URL . 'auth/login.php');
    exit();
}
$recipeTitle = "Всі рецепти";
setlocale(LC_COLLATE, 'uk_UA.utf8');
$categories_recipes = SELECT_ALL("categories");
usort($categories_recipes, function($a, $b) {return
strcoll($a['name'], $b['name']);});
if (isset($_GET['search']) && !empty($_GET['search'])) {
    $recipeTitle = "Результат пошуку";
    if (isset($_GET['category_id']) &&
!empty($_GET['category_id'])) {
        $categoryId = $_GET['category_id'];
        $searchedRecipes =
SEARCH_RESIPES_ON_GATEGORIES($_GET['search'], $categoryId);
        $recipes = $searchedRecipes;
    } else {
        $searchedRecipes = SEARCH_RESIPES($_GET['search']);
        $recipes = $searchedRecipes;
    }
} elseif (isset($_GET['category_id']) &&
!empty($_GET['category_id'])) {
    $categoryId = $_GET['category_id'];
    $recipesId = RECIPES_BY_CATEGORY($categoryId);
    $recipeIds = array();
    foreach ($recipesId as $recipe) {$recipeIds[] =
$recipe['recipe_id'];}
    $recipes = array();
    foreach ($recipeIds as $recipeId) {
        $selectedRecipe = SELECT_ONE('recipes', array('id' =>
$recipeId, 'published' => 1));
        if ($selectedRecipe) {$recipes[] = $selectedRecipe;}
    }
    $existingcat = SELECT_ONE('categories', ['id' =>
$categoryId]);
    $recipeTitle = "Рецепти в категорії";
    $recipeTitle .= " \"" . $existingcat['name'] . "\"";
} else {$recipes = ALL_RECIPES(1);}
$currentPage = isset($_GET['page']) ? max(1, (int)$_GET['page']) :
1;?>
<!DOCTYPE html>
<html lang="en"><head>
    <meta charset="UTF-8"><meta name="viewport"
content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./css/header.css">
    <link rel="stylesheet" href="./css/style.css">
```

```
    <link rel="stylesheet" href="./css/all_recipes.css">
    <link rel="icon" href="./img/FA.ico" type="image/x-icon">
    <!-- JQuery --><script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.1/jquery.mi
n.js"></script>
    <!-- Slick library --><script
src="./js/slick.min.js"></script>
    <!-- My script --><script src="./js/scripts.js"></script>
    <title>Кулінарні рецепти</title></head>
<body>
<div class="body">
    <?php include("./app/include/header.php"); ?>
    <main>
        <div class="content-wrapper">
            <div class="categories">
                <div class="search"><form action="all_recipes.php"
method="get"><input type="search" name="search" placeholder="Пошук
по назві та вмісту" required><input type="hidden"
name="category_id" value="<?php echo isset($_GET['category_id']) ?
$_GET['category_id'] : ''; ?>"><button
type="submit">Пошук</button></form></div>
                <h2 class="cat-title">Категорії <img src="<?php
echo BASE_URL ?>img/down.png" alt="" height="20px"
class="dowm"></h2>
                <div class="categories-all"><?php
foreach($categories_recipes as $cat): ?><div class="<?php if
($cat['id'] == $categoryId) echo 'active active_categories'; else
echo 'style_cat'; ?>" data-id="<?php echo $cat['id'] ?>"
style="margin-bottom: 10px;"><a href="?category_id=<?php echo
$cat['id']; ?>&page=" style="text-decoration: none;"><div><?php
echo $cat['name'] ?></div></a></div> <?php endforeach; ?></div>
            </div>
            <div class="recipes clearfix">
                <h1 class="recipes-title"><?php echo $recipeTitle;
?></h1>
                <?php $perPage = 5;
                    $recipesCount = count($recipes);
                    $totalPages = ceil($recipesCount / $perPage);
                    $offset = ($currentPage - 1) * $perPage;
                    $recipes = array_slice($recipes, $offset,
$perPage);
                    foreach($recipes as $recipe): ?>
                        <div class="recipe clearfix">
                            <a
href="./recipes/single_recipe.php?id=<?php echo $recipe['id'];
?>"><img src="./img/<?php echo $recipe['image']; ?>" alt=""
class="recipe-img"></a>
                            <div class="recipe-preview">
                                <h2 class="title_recipe"><a
href="./recipes/single_recipe.php?id=<?php echo $recipe['id'];
?>"><?php echo $recipe['title']; ?></a></h2>
                                <div class="text-center-user"><img
src="./img/user.png" alt="" height="16px" class="img-right"> <?php
$userlogin = SELECT_USER($recipe['user_id']); echo
$userlogin[0]['login']; ?>   <img src="./img/calendar.png"
```

```
alt="" height="16px" class="img-right"> <?php echo date('d.m.Y',
strtotime($recipe['create_date'])); ?></div>
                                <p class="previem-text"><span
style="font-weight: bold;">Категорії:</span>
                                    <?php $category_recipe =
SELECT_ALL('recipe_categories', ['recipe_id' => $recipe['id']]);
                                    $category_id = array();
                                    foreach ($category_recipe
as $category) {$category_id[] = $category['category_id'];}
                                    $category_ids = implode(',
', $category_id);
                                    $categoryNames =
CategoryNamesWithIds($category_ids);
                                    if
(!empty($categoryNames)): ?><?php foreach ($categoryNames as
$category): ?><a href="all_recipes.php?category_id=<?php echo
$category['id']; ?>&page=" class="selected-category"><?php echo
$category['name']; ?></a><?php endforeach; ?><?php endif; ?></p>
                                <p class="ingredients previem-
text"><span style="font-weight: bold;">Інгредієнти:</span> <?php
echo mb_substr($recipe['ingredients'], 0, 150, 'UTF-8') . '...';
?></p>
                                <p class="preparation previem-
text"><span style="font-weight: bold;">Приготування:</span> <?php
echo mb_substr($recipe['preparation'], 0, 200, 'UTF-8') . '...';
?></p>
                                <div class="btn-group"><a
href="all_recipes.php?saved_recipe_id=<?php echo $recipe['id'];
?>" class="btn">Додати в улюблені</a><a
href="./recipes/single_recipe.php?id=<?php echo $recipe['id']; ?>"
class="btn">Читати детальніше</a></div>
                            </div></div>
                    <?php endforeach; ?>
                        <?php if ($totalPages > 1): ?><ul
class="pagination"><?php for ($i = 1; $i <= $totalPages; $i++):
?><li>
                                    <?php
if(isset($_GET['category_id'])): ?><a href ="?search=<?php echo
urlencode($_GET['search']); ?>&category_id=<?php echo $categoryId;
?>&page=<?php echo $i; ?>" <?php if ($i == $currentPage) echo
'class="active"'; ?>><?php echo $i; ?></a>
                                    <?php
elseif(isset($_GET['search'])): ?><a href ="?search=<?php echo
urlencode($_GET['search']); ?>&page=<?php echo $i; ?>" <?php if
($i == $currentPage) echo 'class="active"'; ?>><?php echo $i;
?></a>
                                    <?php else: ?><a href
="?page=<?php echo $i; ?>" <?php if ($i == $currentPage) echo
'class="active"'; ?>><?php echo $i; ?></a>
                                    <?php endif; ?>
                        </li><?php endfor; ?></ul><?php endif; ?>
            </div>
        </div>
    </main>
    <?php include("./app/include/footer.php"); ?>
</div></body></html>
```

Додаток З

Лістинг скрипту додавання рецепту на вебсайту

```
<?php include($_SERVER['DOCUMENT_ROOT'] . '/app/database/db.php');
$table = 'recipes';
$recipes = SELECT_ALL($table);
$errors = array();
$id = "";
$title = "";
$ingredients = "";
$preparation = "";
$published = "";
if(isset($_POST['add-recipe'])){
$errors = array();
if(empty($_POST['title'])){
array_push($errors, 'Введіть назву рецепту!'); }
if(empty($_POST['categories'])){
array_push($errors, 'Виберіть категорії!');}
if(empty($_POST['ingredients'])){
array_push($errors, 'Введіть інгридієнти рецепту!');}
if(empty($_POST['preparation'])){
array_push($errors, 'Введіть приготування рецепту!');}
if(isset($_FILES['image']['name'])){
$image_name = time() . '_' . $_FILES['image']['name'];
$destination = '../img/' . $image_name;
$result = move_uploaded_file($_FILES['image']['tmp_name'],
$destination);
if($result){$_POST['image'] = $image_name;
}else{
array_push($errors, "Помилка завантаження фотографії!");}
}else{
array_push($errors, "Потрібно завантаження фотографію!");}
if(count($errors) == 0){
$cat = $_POST['categories'];
unset($_POST['add-recipe'], $_POST['selectCategories'],
$_POST['categories']);
$_POST['user_id'] = $_SESSION['id'];
$_POST['published'] = isset($_POST['published']) ? 1 : 0;
$recipe_id = INSERT($table, $_POST);
$selectedCategories = explode(',', $cat);
foreach ($selectedCategories as $category_id) {
INSERT('recipe_categories', array('recipe_id' => $recipe_id,
'category_id' => $category_id));
}
$_SESSION['message'] = "Рецепт додано успішно!";
$_SESSION['type'] = "success";
header('Location: ./manage_recipes.php');
exit();
}else{
$title = $_POST['title'];
$ingredients = $_POST['ingredients'];
$preparation = $_POST['preparation'];
$published = isset($_POST['published']) ? 1 : 0;    }
}?>
```

Додаток И

Лістинг структури таблиці керування рецептами вебсайту

```
<table>
    <thead>
        <th>№</th>
        <th>Назва</th>
        <th>Дії</th>
        <th>Статус публікування</th>
    </thead>
    <tbody>
        <?php $perPage = 5;
        $recipesCount = count($recipes);
        $totalPages = ceil($recipesCount / $perPage);
        $offset = ($currentPage - 1) * $perPage;
        $recipes = array_slice($recipes, $offset, $perPage);
        foreach ($recipes as $key => $recipe): ?>
            <tr>
                <td><?php echo $key+1; ?></td>
                <td><a href="./single_recipe.php?id=<?php echo
$recipe['id']; ?>"><?php echo $recipe['title']; ?></a></td>
                <td style="white-space: nowrap;" class="icon"><div
class="action-links">
                    <a href="edit_recipe.php?id=<?php echo
$recipe['id']; ?>" class="edit">
                        <img src="../img/edit.png"
alt="Редагувати" height="35px" title="Редагувати">
                    </a>
                    <a href="edit_recipe.php?delete_id=<?php echo
$recipe['id']; ?>" class="delete">
                        <img src="../img/delete.png"
alt="Видалити" height="35px" title="Видалити">
                    </a>
                </div></td>
                <td class="icon">
                    <?php if ($recipe['published']): ?>
                        <div>
                            <img src="../img/accept.png"
alt="Опублікований" height="35px" title="Опублікований">
                        </div>
                    <?php else: ?>
                        <div>
                            <0img src="../img/cross.png"
alt="Неопублікований" height="35px" title="Неопублікований">
                        </div>
                    <?php endif; ?>
                </td>
            </tr>
        <?php endforeach; ?>
    </tbody>
</table>
```