МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет економіки і підприємництва

Кафедра інформаційних технологій

Допущено до захисту

"_____" _____ 2025 р.

Зав. кафедри, к.т.н., доцент

_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Розробка веб-застосунку для пошуку кулінарних рецептів

Виконав - студент 4 курсу, групи 41-кн
освітньої програми «Комп'ютерні науки»
спеціальності 122 Комп'ютерні науки
Ющенко Дмитро Віталійович
Науковий керівник - д.т.н., професор
Кулаков Павло Ігорович

Умань - 2025

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ……..... 7
ВСТУП…… 8
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ…… 10
1.1. Предмет дослідження та технології розробки …… 10
1.2. Порівняльний аналіз аналогів ……. 16
1.3. Постановка задачі..... 19
РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ, МЕТОДІВ І АЛГОРИТМІВ РОБОТИ ВЕБ-ЗАСТОСУНКУ…….. 21
2.1. Розробка моделі веб-застосунку……..... 21
2.2. Розробка методів пошуку рецептів та замовлення продуктів …… 24
2.3. Розробка загального алгоритму роботи застосунку та діаграм станів..... 28
РОЗДІЛ 3. РОЗРОБКА ВЕБ-ЗАСТОСУНКУ……... 32
3.1. Розробка структури бази даних..... 32
3.2. Розробка модуля авторизації та реєстрації користувача…… 37
3.3. Розробка головної вкладки та модуля перегляду рецептів веб-застосунку…… 42
ВИСНОВОК…… 48
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ…… 49
ДОДАТОК А. Лістинг програми……... 51

Вступ

Організація процесу приготування страв хоч і може спершу видатися простим завданням, особливо в домашніх умовах, насправді, це може нести свої незручності та ризики. Інколи трапляються ситуації, коли стає проблематично підібрати страву для приготування, виходячи з наявних інгредієнтів. Також, можлива інша проблема, яка полягає у відсутності зручного способу придбати необхідні продукти. Причинами таких проблем може бути: деякі рецепти вимагають специфічних інгредієнтів, які можуть бути важко знайти в місцевих магазинах, це особливо стосується екзотичних або регіональних продуктів; навіть якщо уже заплановано приготувати страву з вже наявних продуктів, виявляється, що ключові інгредієнти можуть бути відсутніми, що змінює плани; пошук альтернативних інгредієнтів або магазинів, де їх можна придбати, може забирати багато часу і зусиль, це може бути особливо проблематично для зайнятих людей або тих, хто живе в місцях з обмеженим доступом до різноманітних магазинів; знаходження замінників для відсутніх інгредієнтів часто вимагає додаткових витрат на нові продукти або компромісні варіанти, що може підвищити витрати на приготування страв; якщо людина уже запланувала покупку необхідних продуктів заздалегідь, іноді стаються несподівані ситуації, коли змінюється робочий графік або виявляється, що продукти у магазині неналежної якості. Отже, актуальною є розробка вебзастосунку для підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів.

Актуальність

Актуальність теми зумовлена тим, що організація процесу приготування страв у домашніх умовах має ряд незручностей і ризиків. Трапляються ситуації, коли проблематично підібрати страву для приготування виходячи з наявних інгредієнтів, або виникає потреба придбати необхідні продукти, що вирішується за допомогою розробленого застосунку.

Об’єкт дослідження - процес підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів.

Предмет дослідження - методи і засоби реалізації вебзастосунку для підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів.

Мета дослідження

Метою та завданням кваліфікаційної роботи бакалавра є удосконалення процесу пошуку рецептів та замовлення необхідних продуктів за рахунок розробки спеціального вебзастосунку, в якому реалізовані нові методи пошуку рецептів та замовлення продуктів.

Відповідно до поставленої мети потрібно вирішити такі завдання:

- здійснити аналіз стану питання та методів розв’язання задачі;
- розробити модель, загальний алгоритм роботи та діаграми станів застосунку;
- розробити метод замовлення продуктів;
- розробити метод пошуку рецептів;
- здійснити програмну реалізацію веб-застосунку;
- описати системні вимоги для веб-застосунку для підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Предмет дослідження та технології розробки

Застосунки для підбору рецептів є популярним і затребуваним напрямком у сфері кулінарних застосунків та сервісів. Такі застосунки спрямовані на надання користувачам широкого доступу до великих баз даних рецептів та можливості персоналізувати процес пошуку та вибору страв під свої смаки, дієтичні переваги та наявні продукти.

Сучасний стан розвитку таких застосунків характеризується постійним вдосконаленням функціональності та додаванням нових можливостей. Провідні рішення, такі як Cookpad, Yummly та Mealime, пропонують користувачам цілий спектр функцій: пошук рецептів за різноманітними критеріями, збереження улюблених страв, автоматичне формування списків покупок, а подекуди і інтеграцію з сервісами доставки продуктів.

Актуальність розробки нових застосунків для підбору рецептів обґрунтовується низкою важливих факторів. По-перше, сучасні користувачі потребують зручних інструментів для пошуку страв на основі наявних продуктів, особливо коли бракує окремих інгредієнтів для улюблених рецептів. По друге, планування та ефективне придбання необхідних продуктів часто є проблемою, яку можна вирішити завдяки інтеграції застосунків з сервісами доставки.

Крім того, постійно зростає попит на мобільні та веб-застосунки, що спрощують повсякденні завдання, зокрема приготування їжі. Характерними особливостями таких кулінарних застосунків є великі бази даних рецептів, різноманітні можливості пошуку та фільтрації, функції збереження улюблених страв, детальна інформація про інгредієнти та покрокові інструкції приготування, а також елементи персоналізації рекомендацій під індивідуальні вподобання користувачів.

Найбільшого поширення такі застосунки отримали на мобільних платформах iOS та Android [1], а також у вигляді веб-сервісів [2 - 3], оскільки

вони забезпечують найбільш зручний та доступний спосіб взаємодії з кулінарними рецептами та можливостями замовлення продуктів "на ходу" [4 - 5]. Розвиток застосунків для підбору рецептів та планування покупок продуктів є важливим трендом у сфері кулінарних онлайн-сервісів. Поєднання великих баз даних рецептів, персоналізованих рекомендацій та можливості замовлення необхідних інгредієнтів робить такі застосунки дедалі затребуванішими серед сучасних користувачів, які прагнуть спростити процес планування та приготування домашніх страв.

Для розробки вебзастосунку для підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів можуть бути використані різноманітні підходи та інструменти. Одним із популярних варіантів є застосування MERN stack [6] - сучасного технологічного стеку, що включає в себе MongoDB [7], Express.js [8], React.js [9] та Node.js [10].

MERN stack є одним з найкращих варіантів для розробки такого вебзастосунку з кількох ключових причин. По-перше, використання єдиної мови програмування JavaScript, як для frontend, так і для backend частини, значно спрощує розробку та підтримку проекту. По друге, MERN stack характеризується високим рівнем гнучкості та масштабованості завдяки модульній архітектурі на основі мікросервісів. Крім того, екосистема бібліотек та інструментів, доступних для MERN stack, є надзвичайно широкою та активно розвивається, що прискорює процес розробки.

Вибір NoSQL бази даних MongoDB також є вдалим рішенням для такого типу застосунку, оскільки гнучкість у структурі даних, масштабованість та високі показники продуктивності роблять її ідеальним вибором для зберігання та обробки даних про рецепти, інгредієнти, користувачів тощо. Приклад використання MongoDB наведено на рисунку 1.1. Крім того, MERN stack дозволяє створювати SPA (Single Page Application) - застосунки, що забезпечують плавну та інтерактивну взаємодію користувача, що відповідає сучасним вимогам до користувацького досвіду.

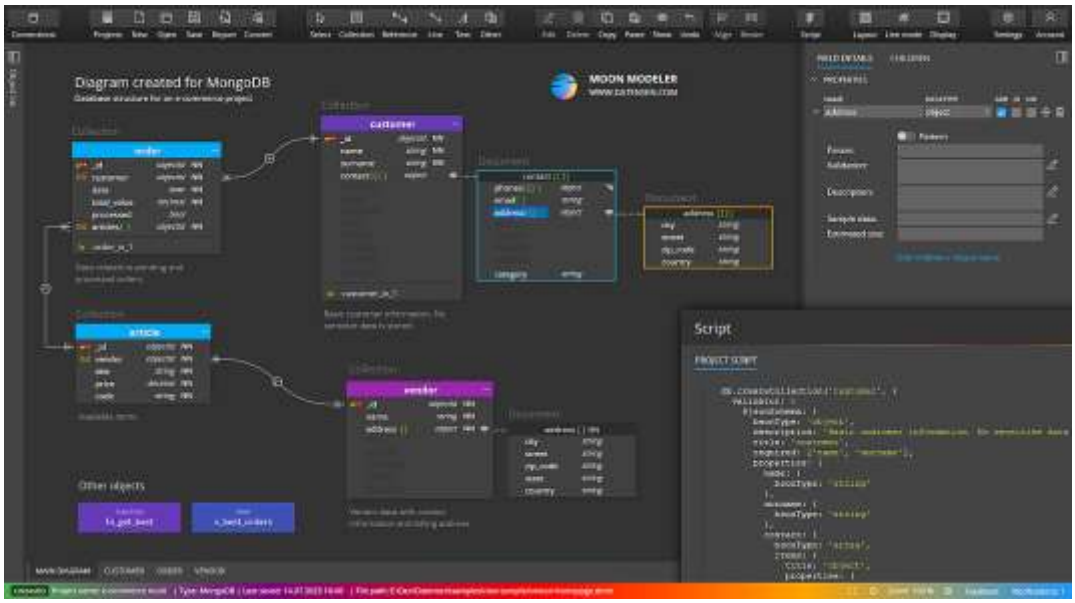


Рисунок 1.1 - Приклад використання MongoDB

Порівняння традиційного багатосторінкового застосунку та односторінкового застосунку наведено на рисунку 1.2.

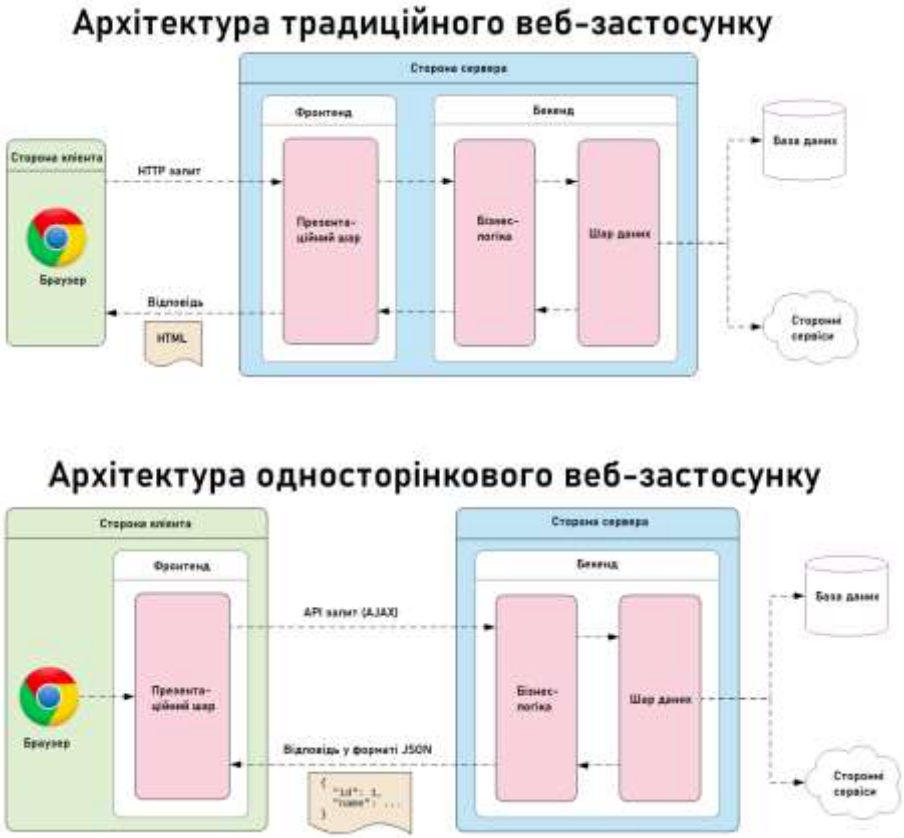


Рисунок 1.2 - Архітектура багатосторінкового за односторінкового застосунків

Express.js – це мінімалістичний та гнучкий веб-фреймворк для Node.js. Він надає набір функціональностей для створення веб-застосунків та API. Express.js спрощує процес маршрутизації, обробки запитів та відповідей, а також управління сесіями та cookies. Розробники цінують Express.js за його простоту, швидкість розробки та велику екосистему плагінів (middleware), які дозволяють легко розширювати функціональність програми. Він є основою для багатьох популярних фреймворків Node.js вищого рівня. Архітектура Express.js наведена на рисунку 1.3.

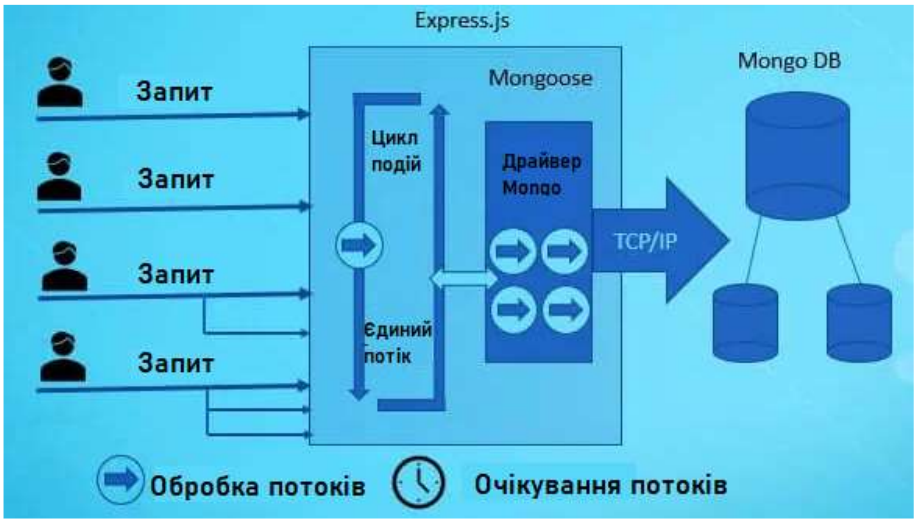


Рисунок 1.3 - Архітектура Express.js

React.js – це бібліотека JavaScript з відкритим кодом для створення користувацьких інтерфейсів. Розроблена та підтримувана Facebook, React використовує компонентний підхід до побудови UI, де кожен компонент може управляти власним станом. Ключовою особливістю React є віртуальний DOM (Document Object Model), який оптимізує оновлення інтерфейсу, порівнюючи віртуальне представлення з реальним DOM і оновлюючи лише змінені частини. React також відомий своєю односпрямованою передачею даних, що робить код більш передбачуваним та простішим для налагодження. Він широко використовується для створення складних, інтерактивних веб-застосунків. Архітектура ReactJS наведена на рисунку 1.4.

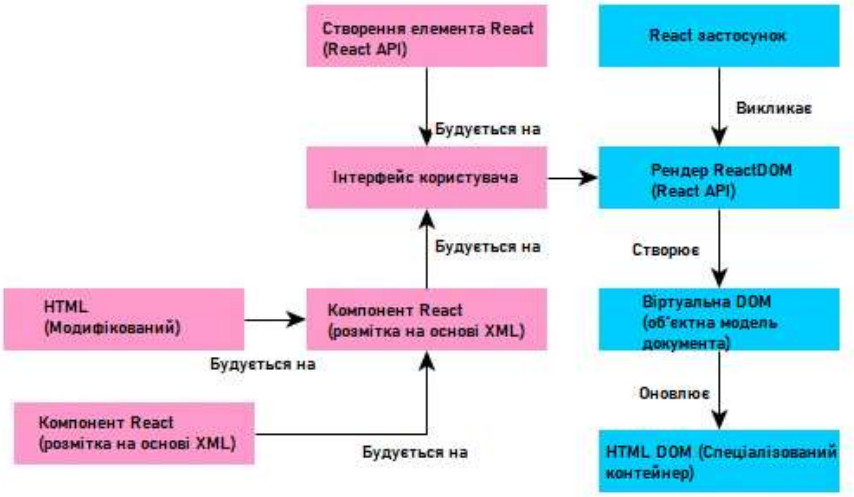


Рисунок 1.4 - Архітектура ReactJS

Node.js – це середовище виконання JavaScript на стороні сервера, побудоване на двигуні V8 від Google Chrome. Воно дозволяє розробникам використовувати JavaScript для написання серверних скриптів, створення веб-серверів та інструментів командного рядка. Node.js використовує неблокуючу, подієво-орієнтовану модель, що робить його легким та ефективним для застосунків, які працюють з даними в реальному часі та інтенсивно використовують введення/виведення. Його пакетний менеджер npm є найбільшим у світі реєстром бібліотек з відкритим кодом. Node.js широко використовується для створення масштабованих мережевих застосунків, API та мікросервісів. Архітектура Node.js продемонстрована на рисунку 1.5.

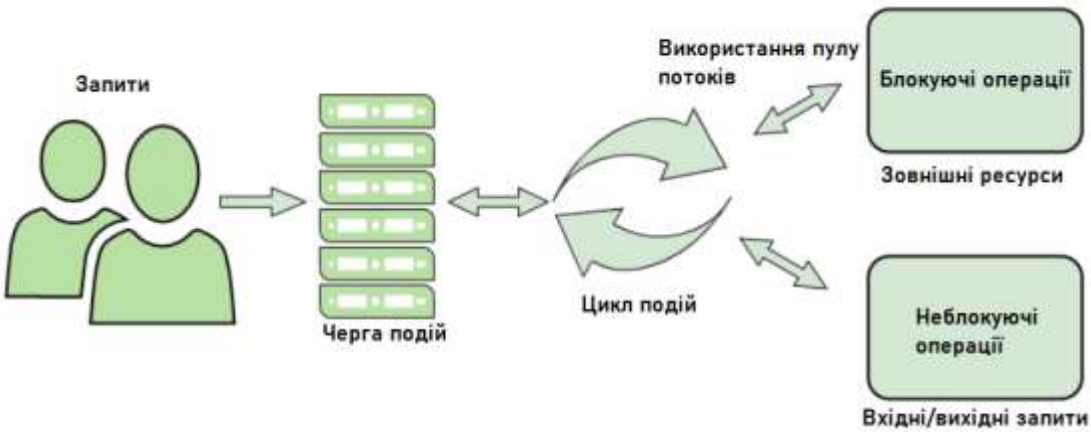


Рисунок 1.5 - Архітектура Node.js

Крім MERN stack, для розробки вебзастосунку можуть бути використані і інші підходи та інструменти. Наприклад, для frontend-частини можна застосувати сучасні JavaScript-фреймворки та бібліотеки, такі як React, Angular або Vue.js, а для backend-частини - мови програмування, такі як Node.js, Python (Django, Flask) або PHP (Laravel, Symfony). Незалежно від обраного технологічного стеку, важливо також приділити увагу архітектурним підходам, вибору систем контролю версій, інструментів автоматизації, менеджерів пакетів, IDE та інших допоміжних засобів. Крім того, необхідно забезпечити інтеграцію з зовнішніми сервісами, таким як API сервісів доставки продуктів, API пошуку рецептів та інгредієнтів, платіжні системи, а також приділити увагу забезпеченню безпеки веб-застосунку. Особливу увагу також слід приділити розробці інтуїтивно зрозумілого та зручного користувацького інтерфейсу, використовуючи сучасні бібліотеки UI-компонентів та проводячи юзабіліті-тестування для вдосконалення UX. Комплексне застосування описаних підходів та інструментів, зокрема MERN stack, дозволить створити високоякісний, масштабований та безпечний веб-застосунок для підбору рецептів, який задовольнятиме потреби сучасних користувачів.

Отже, на основі проведеного аналізу можна зробити висновок, що застосування MERN stack є найкращим підходом для розробки вебзастосунку для підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів. Переваги MERN stack для такого типу застосунку є очевидними. Використання єдиної мови програмування JavaScript як для frontend, так і для backend частини значно спрощує розробку та підтримку проекту. Модульна архітектура на основі мікросервісів забезпечує високу гнучкість та масштабованість системи. Крім того, велика екосистема бібліотек та інструментів, доступних для MERN stack, прискорює процес розробки. Вибір NoSQL бази даних MongoDB та можливість створення SPA-застосунків також є вагомими перевагами MERN stack для вебзастосунку з функціями пошуку рецептів та замовлення продуктів. MongoDB дозволяє ефективно управляти

різноструктурованими даними про рецепти, інгредієнти та користувачів, а SPA-архітектура забезпечує плавну і інтуїтивно зрозумілу взаємодію користувача. Порівняно з іншими технологічними стеками, MERN stack надає найбільш збалансований та комплексний набір інструментів і підходів для реалізації вебзастосунку з функціоналом, що відповідає сучасним вимогам користувачів. Тому його застосування можна вважати найкращим рішенням для розробки такого типу веб-сервісу.

1.2 Порівняльний аналіз аналогів

При визначенні вимог до розроблюваного вебзастосунку для підбору рецептів, було проведено аналіз існуючих аналогів. Це необхідно для кращого розуміння потреб користувачів та виявлення функціоналу, який ще не реалізовано в цих застосунках.

Cookpad [3] - це мобільний застосунок та веб-сайт, який надає користувачам можливість знаходити, ділитися та зберігати рецепти страв. Він пропонує велику базу даних рецептів, створених користувачами з усього світу, що дозволяє знайти рецепти на будь-який смак та в будь-якому стилі кулінарії. Інтерфейс застосунка Cookpad наведено на рисунку 1.6.

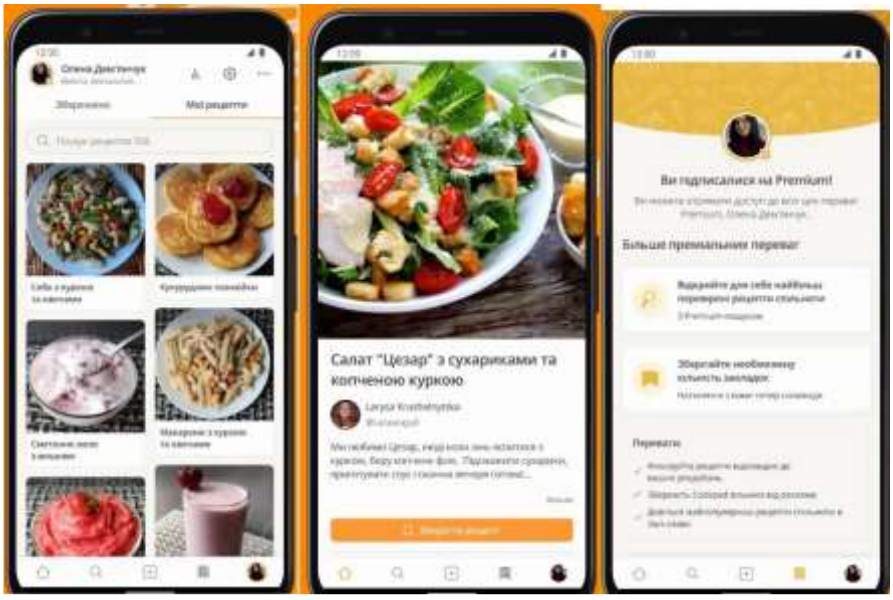


Рисунок 1.6 - Інтерфейс застосунку Cookpad

Основні характеристики Cookpad включають:

- користувачі можуть шукати рецепти за різними критеріями, такими як інгредієнти, типи страв, кухні, складність приготування та інші;

- користувачі можуть зберігати рецепти, які їм подобаються, у власному обліковому записі для подальшого використання;

- користувачі можуть ділитися своїми власними рецептами з іншими користувачами спільноти Cookpad.

Yummly [4] - це онлайн-платформа та мобільний застосунок, який надає користувачам можливість знаходити, зберігати та ділитися рецептами страв. Він відомий своєю великою базою даних рецептів, інтеграцією зі смарт-приладами та іншими корисними функціями. Головну сторінку веб-сайту Yummly наведено на рисунку 1.7.

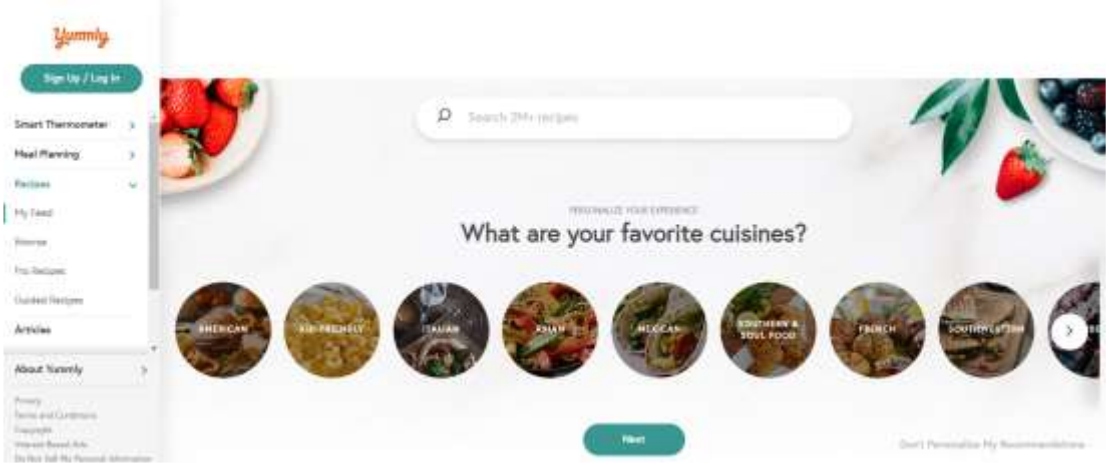


Рисунок 1.7 - Головна сторінка Yummly

Основні характеристики Yummly включають:

- Yummly використовує алгоритми рекомендацій для адаптації підбору рецептів під індивідуальні вподобання та потреби користувачів;

- користувачі можуть шукати рецепти за різними категоріями, такими як тип страв (сніданок, обід, вечеря), кухні світу, дієтичні обмеження та інші;

- користувачі можуть зберігати рецепти у власному обліковому записі на платформі для подальшого використання;

- Yummly автоматично генерує список покупок на основі обраних рецептів, що дозволяє користувачам зручно планувати покупки продуктів.

Mealime [5] - це мобільний застосунок, спрямований на полегшення процесу планування та приготування їжі для користувачів. Він пропонує різноманітні рецепти з можливістю персоналізації та генерації списків покупок для зручності користувачів. Застосунок Mealime проілюстровано на рисунку 1.8.

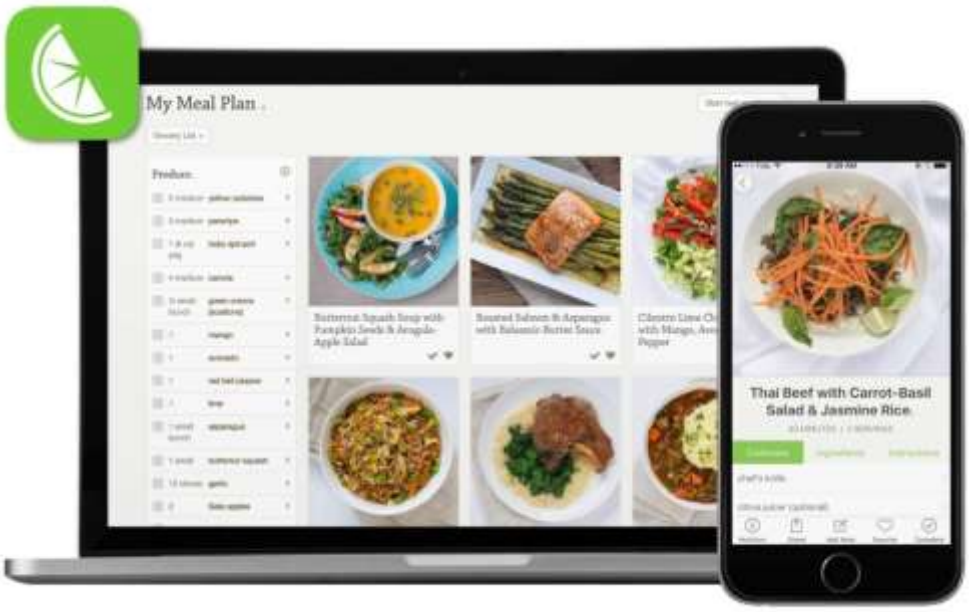


Рисунок 1.8 - Інтерфейс застосунку Mealime

Основні характеристики Mealime включають:

- користувачі можуть вибирати рецепти на кожен день тижня з великого списку доступних опцій;

- веб-застосунок враховує індивідуальні дієтичні обмеження та вподобання користувачів, пропонуючи рецепти, які відповідають їхнім потребам;

- веб-застосунок автоматично генерує списки покупок на основі обраних рецептів, допомагаючи користувачам ефективно планувати покупки продуктів;

- кожен рецепт супроводжується детальними інструкціями приготування та фотографіями, що дозволяє користувачам легко готувати страви навіть без великого досвіду в кулінарії.

Для наочної демонстрації відмінностей розглянутих застосунків було зведено їх переваги і недоліки у таблицю порівняння (таблиця 1.1).

Таблиця 1.1 – Порівняльний аналіз аналогів

	Cookpad	Yummly	Mealime	Mealthy
Пошук рецептів	1	1	1	1
Збереження улюблених рецептів	1	1	1	1
Детальна інформація про рецепти	1	1	1	1
Список покупок	0	1	1	1
Замовлення продуктів з магазину	0	0	0	1
Сумарний бал	**3**	**4**	**4**	**5**

Проаналізувавши результати з таблиці 1, можна зробити висновок, що розроблюваний застосунок має вищий сумарний бал за Cookpad на 40%, за Yummly та Mealime на 20%.

1.3 Постановка задачі

Метою кваліфікаційної роботи бакалавра є удосконалення процесу пошуку рецептів та замовлення необхідних продуктів за рахунок розробки спеціального веб-застосунку, в якому реалізовані нові методи пошуку рецептів та замовлення продуктів. Застосунок, який необхідно розробити, повинен мати зручний та інтуітивний інтерфейс, можливість легкого редагування компонентів, можливість обробки великої кількості запитів на сервер, повинен мати розвинуту базу даних, яка зможе зберігати усі потрібні ресурси.

Завданням розробки є аналіз предметної області та проведення порівняльного аналізу аналогів, обґрунтування вибору технологій для розробки веб-застосунку, опис системних вимог до веб-застосунку для підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів, розробка моделі,

загального алгоритму роботи та діаграми станів веб-застосунку, розробка методу замовлення продуктів, який на відміну від відомих, дозволяє підбирати інгредієнти, необхідні для приготування обраного рецепту, з можливістю їх доставки, розробка методу пошуку рецептів, який на відміну від відомих, дозволяє здійснювати пошук рецептів за наявними інгредієнтами, що полегшує користувачам підбір страви для приготування, розробка веб-застосунку.

РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ, МЕТОДІВ І АЛГОРИТМІВ РОБОТИ ВЕБ-ЗАСТОСУНКУ

2.1 Розробка моделі веб-застосунку

Веб-застосунки для пошуку рецептів оперують з різноманітними типами даних для реалізації своєї функціональності. Основними категоріями даних, якими вони працюють, є дані про рецепти, дані про інгредієнти, дані про користувачів, а також дані про замовлення.

Дані про рецепти включають інформацію, таку як назва страви, інгредієнти та їх кількість, інструкції приготування, час та складність приготування, категорія страви, кухня, фотографії готової страви, а також відеоінструкції. Ця інформація є ключовою для пошуку та відображення рецептів користувачам. Дані про інгредієнти містять назву, опис та характеристики інгредієнтів, їх категоризацію, ціну та одиницю виміру, а також інформацію про наявність в магазинах. Такі дані використовуються для формування списків покупок та замовлення необхідних продуктів.

Дані про користувачів включають персональну інформацію, список улюблених рецептів, історію замовлень, а також відомості про дієтичні обмеження та вподобання. Ця категорія даних дозволяє персоналізувати рекомендації та функціонал застосунку для кожного користувача.

Окремо веб-застосунки обробляють дані про замовлення, такі як інформація про замовлені продукти, дату та час замовлення, статус замовлення, дані про доставку, а також спосіб оплати та суму замовлення. Ці дані є необхідними для реалізації можливості замовлення та доставки продуктів через застосунок.

Ефективне управління та інтеграція цих різноманітних наборів даних є ключовим аспектом у розробці сучасних вебзастосунків для пошуку рецептів та замовлення продуктів харчування. Застосунок, що розробляється, не є

виключенням і також використовує та оперує цими категоріями даних для реалізації своїх основних функцій.

Для кращого розуміння архітектури розроблюваного веб-застосунка було побудовано модель системи у вигляді діаграми компонентів (рисунок 2.1).

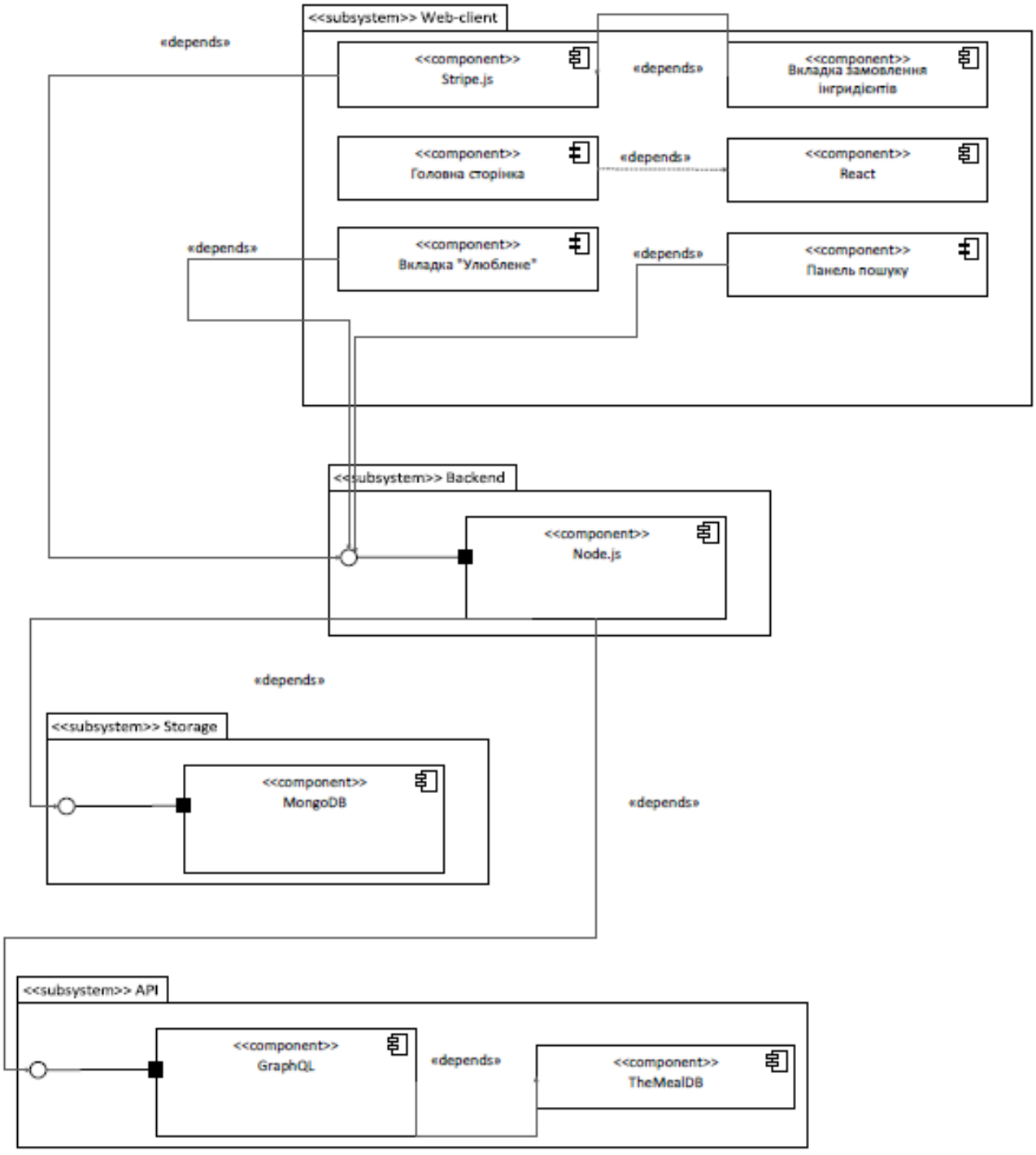


Рисунок 2.1 - Модель системи у вигляді діаграми компонентів

Компонент «Web-client» являє собою фронтенд частину застосунку, яка відповідає за взаємодію з користувачем через веб-браузер. В ній використовується React - популярна JavaScript бібліотеку для створення користувацьких інтерфейсів.

Вкладка замовлення інгредієнтів залежить від бібліотеки Stripe.js для інтеграції з платіжною системою Stripe [11]. Це бібліотека JavaScript від компанії Stripe, яка надає набір програмних інтерфейсів (API) для інтеграції платіжної системи Stripe у веб-застосунки. Вона дозволяє створювати безпечні форми оплати, збирати дані платіжних карток, виконувати процедури 3D Secure, керувати підписками та багато іншого. Основною перевагою Stripe.js є забезпечення безпечного збору даних платіжної картки. Бібліотека надає захищене поле для введення номера картки, дати закінчення терміну дії, CVC-коду та інших даних. Ці дані шифруються та відправляються безпосередньо на сервери Stripe, оминаючи сервер застосунка. Після збору даних картки Stripe.js повертає одноразовий токен, який можна використовувати для створення платежів або зберігання даних картки на стороні Stripe. Stripe.js також дозволяє перевіряти дійсність і правильність введених даних платіжної картки перед відправкою на сервер. Крім того, бібліотека підтримує інтеграцію з популярними платіжними системами, такими як Apple Pay та Google Pay, а також надає можливості для локалізації та кастомізації форм оплати відповідно до потреб конкретного застосунка. У розроблюваній системі Stripe.js використовується для сторінки замовлення та оплати продуктів, які користувач хоче придбати. Він забезпечує формування електронного чека на оплату та переводить користувача на сторінку оплати, де користувач проводить оплату за замовленні інгредієнти.

Головна сторінка відповідає за відображення основного контенту системи. Вона використовує фреймворк React для демонстрації візуальної частини користувачу.

Клієнтська частина системи забезпечується даними з допомогою серверної частини, для якої використовується Node.js. Node.js - це крос-платформове

середовище виконання з відкритим кодом, яке використовує JavaScript для написання серверної частини застосунків. На прикріпленій моделі системи Node.js представлений у вигляді компонента "Node.js". Ключовою особливістю Node.js є використання неблокуючої архітектури, яка забезпечує ефективну роботу з великою кількістю паралельних з'єднань. Це досягається завдяки подіє-орієнтованій моделі та асинхронному введенню/виведенню, що робить Node.js ідеальним вибором для розробки масштабованих мережевих застосунків з великою пропускною здатністю. Node.js також пропонує багатий екосистему модулів, що дозволяє розробникам легко інтегрувати різноманітні функції у свої застосунки.

Бібліотеки та фреймворки, такі як Express.js для веб-розробки, Socket.IO для реалізації двостороннього зв'язку в реальному часі та Mongoose для роботи з базами даних MongoDB, значно спрощують і прискорюють процес розробки. Node.js використовується в системі для забезпечення роботи її серверної частини. Він зв’язує клієнтську частину зі сховищем та з API для отримання рецептів.

Сховище представлене базою даних MongoDB – популярною нереляційною базою даних з відкритим кодом. Використовується для зберігання даних застосунку, таких як користувацька інформація, вміст сайту тощо.

TheGraphQL [12] використовується для зв’язування серверної частини системи з TheMealDB - API, що містить базу даних рецептів.

2.2. Розробка методів пошуку рецептів та замовлення продуктів

Метод пошуку рецептів призначений в першу чергу для полегшення пошуку рецептів тоді, коли є певний набір інгредієнтів, з яких необхідно приготувати страву. Метод пошуку рецептів складається з наступних етапів:

- користувач переходить на сторінку пошуку рецептів.
- користувач обирає пошук за певними інгредієнтами.
- користувач обирає наявні у нього інгредієнти із запропонованого списку.

- система формує перелік рецептів, що містять вказані інгредієнти.
- користувач обирає один із запропонованих рецептів.
- відкривається нове вікно перегляду рецепту.

Блок-схема алгоритму роботи цього методу наведена на рисунку 2.2.



Рисунок 2.2 - Блок-схема алгоритму роботи методу пошуку рецептів

У вікні перегляду рецепту користувач може ознайомитися із рецептом, переглянути відео приготування, а також повний список інгредієнтів, з якого

можна обрати та замовити інгредієнти, яких не вистачає для приготування. Також, користувач може зберегти в «Улюблене» обраний рецепт. Можливість замовлення продуктів, яких не вистачає для приготування страви за певним рецептом, є особливою можливістю, яку пропонує розроблюваний веб-застосунок та якої немає у більшості передових аналогів. Ця можливість може бути дуже корисною для користувачів, адже для них сильно спрощений процес підбору та отримання необхідних інгредієнтів. Окрім того, що існує можливість одразу визначити всі інгредієнти, яких не вистачає, також можна одразу дізнатися вартість, яку буде коштувати та чи інша страва.

Метод замовлення продуктів складається з наступних етапів:

- користувач обирає рецепт, який він хоче приготувати, із запропонованих варіантів або шляхом пошуку.

- вебзастосунок аналізує список інгредієнтів, необхідних для приготування обраного рецепту. Він перевіряє, які з цих інгредієнтів вже є в наявності у користувача.

- на основі аналізу, застосунок формує список інгредієнтів, яких бракує у користувача для приготування страви.

- вебзастосунок пропонує користувачеві замовити відсутні інгредієнти.

- користувач може обрати необхідні продукти зі списку, вказати бажану кількість.

- користувач вказує зручний час та адресу для доставки замовлених продуктів.

- після підтвердження, вебзастосунок передає інформацію про замовлення до сервісів доставки продуктів.

- сервіси доставки опрацьовують отримане замовлення, підтверджують його та організовують доставку продуктів на вказану адресу.

- замовлені продукти доставляються користувачеві у зазначений час та на вказану адресу.

Блок-схема розробленого методу наведена на рисунку 2.3.

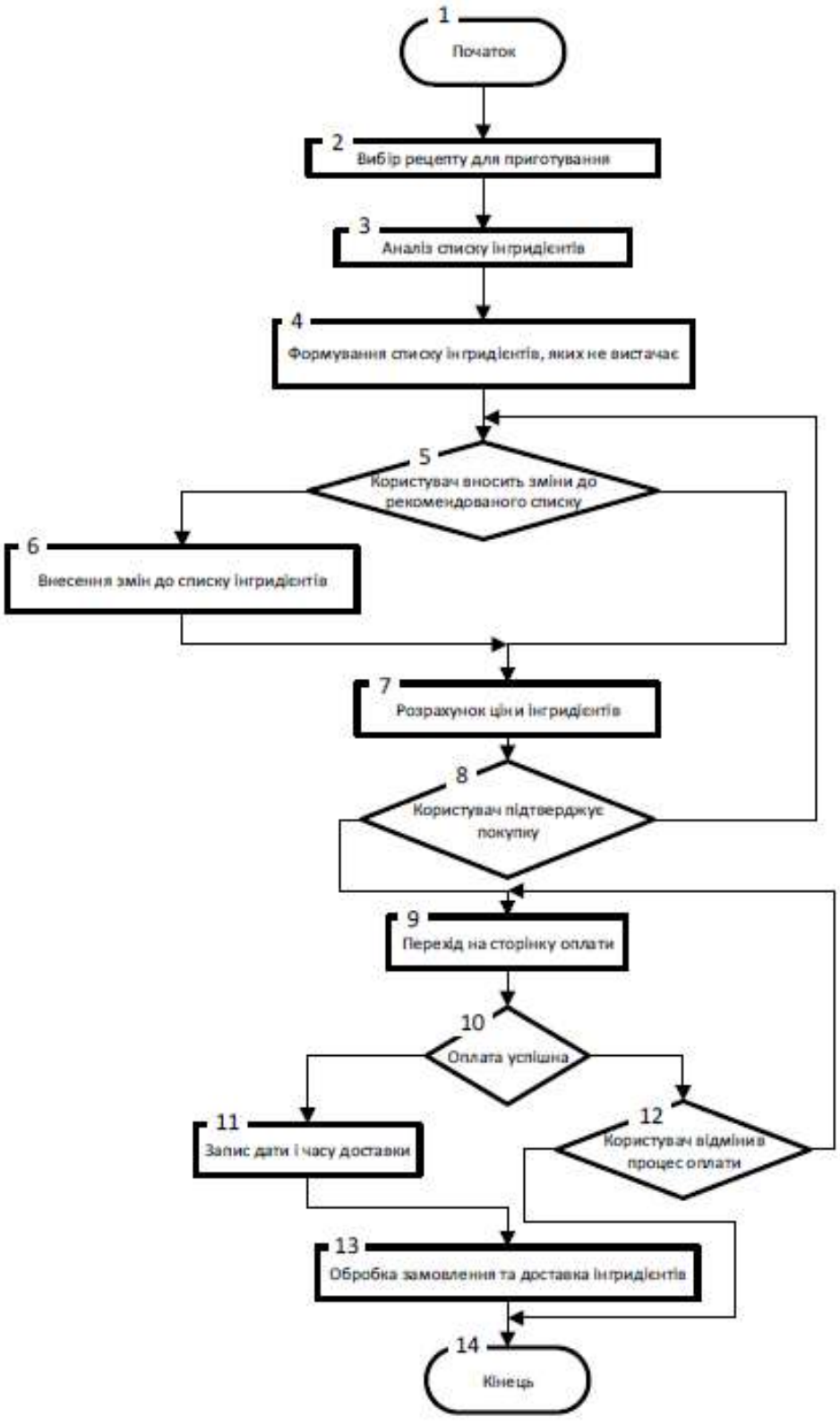


Рисунок 2.3 - Блок-схема методу замовлення продуктів

Таким чином, розроблений метод допомагає користувачам економити час на складанні списку інгредієнтів, що необхідно придбати для приготування рецепту, а також дозволяє замовити необхідні продукти харчування.

2.3. Розробка загального алгоритму роботи застосунку та діаграм станів

Діаграми станів - це графічний спосіб моделювання систем, що складається з різних станів та переходів між ними. Вони широко використовуються для візуалізації поведінки систем у різних умовах. Діаграми станів корисні для аналізу складних систем, таких як програмне забезпечення, електричні мережі або бізнес-процеси.

Було розроблено діаграми станів, що описують поведінку компонентів веб-застосунку. Діаграма станів процесу додавання продуктів в кошик наведена на рисунку 2.4.

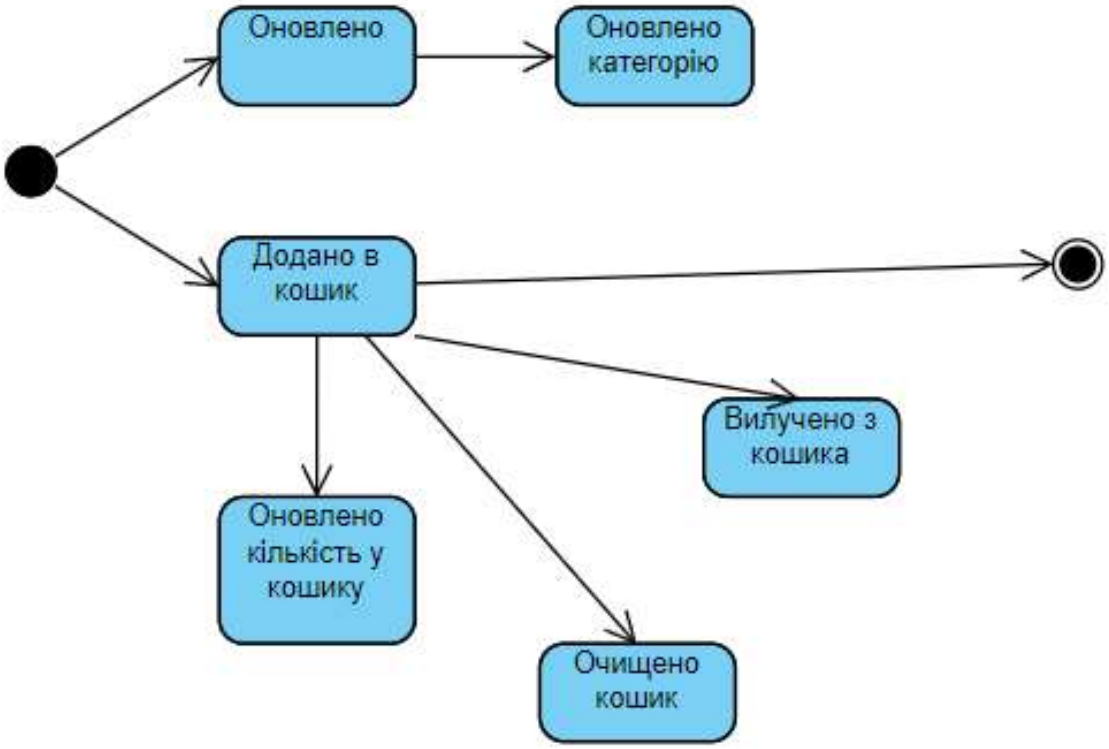


Рисунок 2.4 - Діаграма станів процесу додавання продуктів до кошика

Ця діаграма описує випадки, коли продукти додаються до кошика, вилучаються з нього, коли кошик повністю очищується або оновлюється. Також, в кошику можна оновити кількість доданого до нього певного продукту. Діаграма станів процесу пошуку рецептів наведено на рисунку 2.5.

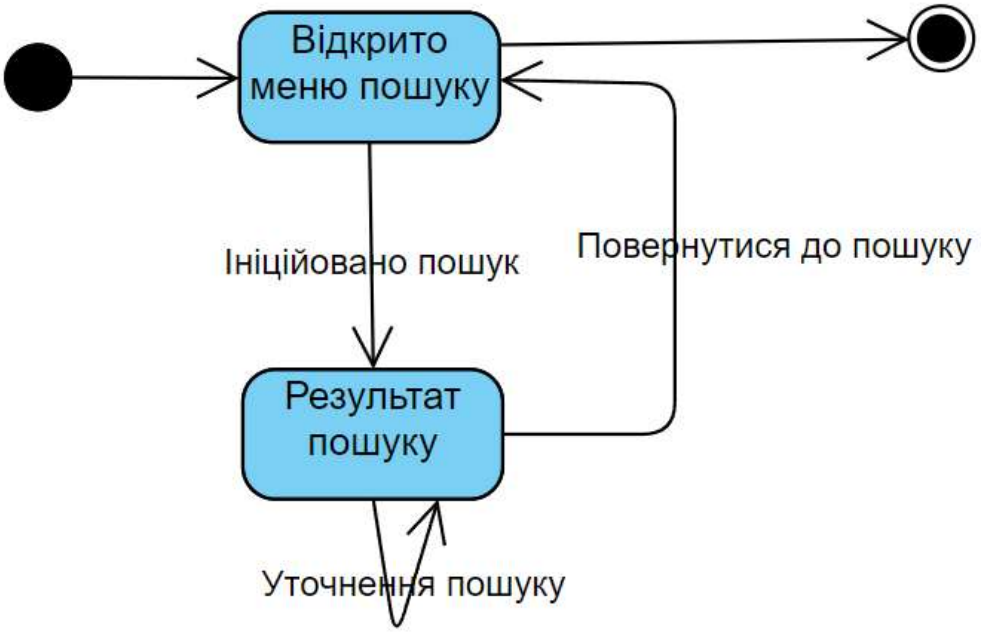


Рисунок 2.5 - Діаграма станів процесу пошуку рецептів

В ній описано початковий стан, коли відкрито пошук рецептів, але ще не введено запит, запуск пошуку, перегляд результатів та повернення назад для повторного пошуку.

На рисунку 2.6 наведено діаграму станів процесу реєстрації. В ній описано такі кроки, як введення даних, надсилання форми для реєстрації, а також стан успіху, коли дані введені правильно та реєстрація пройшла успішно, та стан помилки, коли введені дані не відповідають вимогам, або виникла інша помилка при реєстрації.

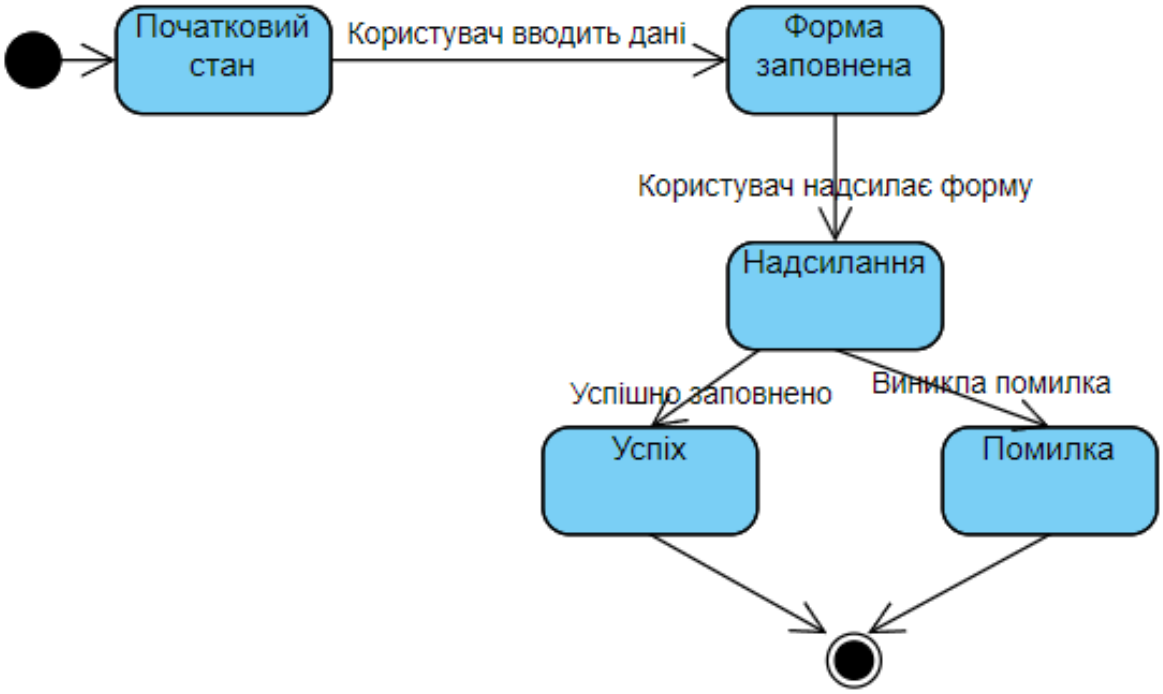


Рисунок 2.6 - Діаграма станів процесу реєстрації

Якщо користувач зареєстрований в системі, тоді він повинен ввести свої дані для входу у свій аккаунт. Якщо він не зареєстрований, тоді йому потрібно пройти процес реєстрації, після чого він зможе користувася усім функціоналом. Загальний алгоритм роботи веб-застосунку наведено на рисунку 2.7.

При пошуку рецептів, користувач спершу обирає категорію необхідного рецепту, після чого знаходить рецепт, що найбільше йому підходить. Також, він може скористатися меню пошуку рецепту за назвою. Користувач може додати рецепти в «Улюблене». Якщо користувач вирішив обрати перелік інгредієнтів, відкривається відповідна вкладка, з якої користувач також може виконати замовлення необхідних інгредієнтів.

Таким чином, було розроблено загальну блок-схему роботи веб-застосунку та діаграми станів різних її складових.

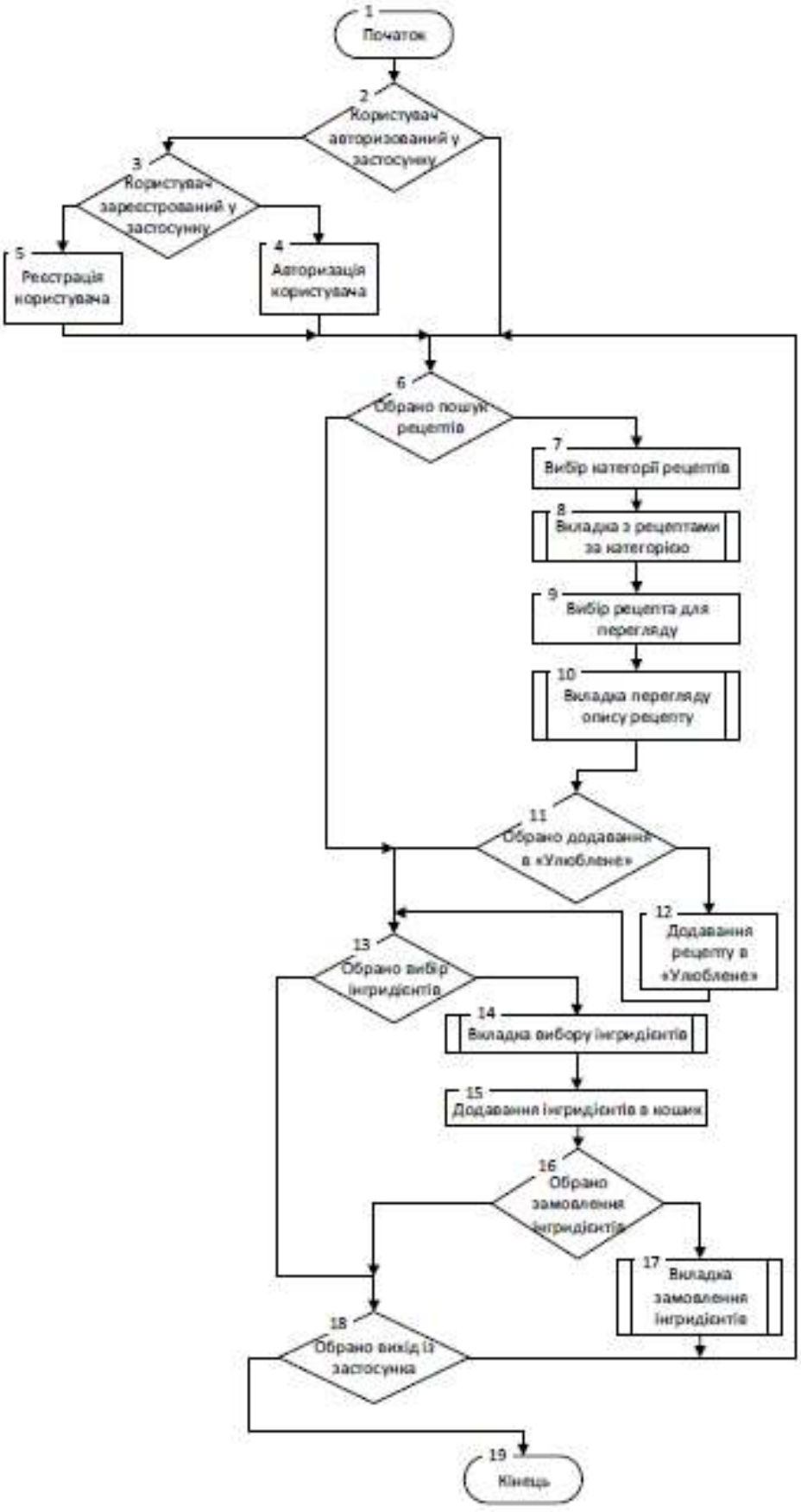


Рисунок 2.7 - Блок-схема загального алгоритму роботи вебзастосунку

РОЗДІЛ 3. РОЗРОБКА ВЕБ-ЗАСТОСУНКУ

3.1 Розробка структури бази даних

JavaScript [13] - це високорівнева, інтерпретована мова програмування, яка є однією з основних технологій веб-контенту разом з HTML та CSS. JavaScript використовується для створення інтерактивних веб-сторінок та веб-застосунків, а також для розробки серверних застосунків за допомогою Node.js. Однією з основних переваг JavaScript є його простота та легкість у вивченні. JavaScript має синтаксис, схожий на синтаксис мов програмування C та Java, що полегшує його вивчення розробникам, які вже знайомі з цими мовами. Крім того, JavaScript має велику кількість бібліотек та фреймворків, таких як jQuery, React та Angular, які дозволяють розробникам швидко створювати складні вебзастосунки. JavaScript є інтерпретованою мовою, що означає, що код JavaScript виконується безпосередньо в веббраузері, без необхідності компіляції.

TypeScript [14] - це відкрита мова програмування, розроблена компанією Microsoft, яка є надбудовою над мовою JavaScript. TypeScript додає статичну типізацію, класи, інтерфейси та інші особливості, відсутні в JavaScript, що робить його більш підходящим для розробки великих та складних програмних проєктів. Однією з основних переваг TypeScript є його здатність працювати з існуючими JavaScript-бібліотеками та фреймворками. TypeScript-код компілюється в JavaScript, що дозволяє використовувати його в будь-якому місці, де підтримується JavaScript. Крім того, TypeScript має вбудовані типи для багатьох популярних JavaScript-бібліотек, таких як jQuery, React та Angular, що полегшує їх використання в TypeScript-проєктах.

PHP (Hypertext Preprocessor) [15] - це серверна мова програмування, яка використовується для створення динамічних вебсторінок та вебзастосунків. PHP є вільним та відкритим програмним забезпеченням, яке розповсюджується під ліцензією PHP License. PHP підтримує багато платформ, включаючи Windows, Linux, macOS та Unix, та може бути використаний з багатьма серверами,

включаючи Apache, Nginx та Microsoft IIS. PHP є інтерпретованою мовою, що означає, що код PHP виконується безпосередньо на сервері, без необхідності компіляції. Це дозволяє розробникам швидко створювати та тестувати вебзастосунки. PHP також підтримує об'єктно-орієнтоване програмування, що дозволяє створювати складні програмні структури та робить код більш модульним та легким у підтримці. PHP має велику кількість бібліотек та фреймворків, таких як Laravel, Symfony, CodeIgniter, Yii, що дозволяють розробникам швидко створювати складні вебзастосунки. PHP також підтримує роботу з багатьма базами даних, включаючи MySQL, PostgreSQL, Oracle, SQLite, що дозволяє створювати вебаплікації, які працюють з великими обсягами даних. PHP має вбудовану підтримку для HTML, CSS та JavaScript, що дозволяє легко створювати динамічні вебсторінки. PHP також підтримує роботу з файлами, що дозволяє завантажувати, зберігати та видаляти файли на сервері. PHP також має вбудовану підтримку для сесій та cookie, що дозволяє зберігати інформацію про користувачів між запитами.

Щодо середовища розробки, то Visual Studio Code [16] є одним з найкращих середовищ розробки вебзастосунків через його широкий спектр функцій, гнучкість і зручність використання. VS Code має вбудовану підтримку багатьох мов програмування, включаючи JavaScript, TypeScript, HTML, CSS, Python, Java, C++ і багато інших. Це означає, що розробники можуть використовувати одне середовище розробки для всіх своїх веб-проектів, незалежно від мови програмування, яку вони використовують.

Розробка структури бази даних є ключовим етапом у процесі створення програмного забезпечення [17]. Правильно спроектована база даних забезпечує ефективне зберігання, організацію та доступ до даних, що відповідає вимогам проєкту. Першим кроком у розробці структури бази даних є аналіз вимог до даних. Це включає вивчення функціональних та нефункціональних вимог, а також потреб користувачів у доступі до інформації. Другим кроком є проектування схеми бази даних. На цьому етапі визначається структура таблиць,

включаючи поля та типи даних, а також зв'язки між таблицями. Важливо враховувати нормалізацію даних для забезпечення їхньої цілісності та уникнення аномалій. Третій крок - реалізація бази даних. Це включає створення таблиць, індексів, обмежень цілісності даних та інших об'єктів бази даних за допомогою мови запитів (наприклад, SQL) або інструментів адміністрування баз даних.

Для роботи бази даних, було створено таблиці Category, User, Comment, Product, Order, OrderProduct, Recipe, UserSavedRecipes.

Таблиця Category містить записи про категорії продуктів та рецептів та складається з таких полів:

- _id - унікальний ідентифікатор категорії. Тип даних – автоматично зростаючий номер.

- name - назва категорії. Тип даних - VARCHAR(255).

Таблиця User містить дані про користувачів системи та складається з таких полів:

- _id - унікальний ідентифікатор користувача. Тип даних: SERIAL (автоматично зростаючий номер).

- username - ім'я користувача. Тип даних: VARCHAR(255). Поле є обов'язковим.

- Email - лектронна адреса користувача. Тип даних: VARCHAR(255). Поле є обов'язковим.

- password – пароль користувача. Тип даних: VARCHAR(255). Поле є обов'язковим.

Таблиця Post містить дані про пости або статті на кулінарну тематику, що опубліковані в системі та складається з таких полів:

- _id - унікальний ідентифікатор поста. Тип даних: SERIAL (автоматично зростаючий номер).

- postText – текст поста. Тип даних: TEXT. Поле є обов'язковим.

- postAuthor – ідентифікатор автора поста. Тип даних: INT. Поле є обов'язковим.

- createdAt - дата і час створення поста. Тип даних: TIMESTAMP. Поле є обов'язковим.

Таблиця Comment містить записи коментарів, що користувачі залишили в системі та складається з таких полів:

- _id - унікальний ідентифікатор коментаря. Тип даних: SERIAL (автоматично зростаючий номер).

- commentText - текст коментаря. Тип даних: TEXT. Поле є обов'язковим.

- commentAuthor - ідентифікатор автора коментаря. Тип даних: INT. Поле є обов'язковим. Зовнішній ключ, що посилається на таблицю User.

- createdAt - дата і час створення коментаря. Тип даних: TIMESTAMP. Поле є обов'язковим.

- postId - ідентифікатор поста, до якого належить коментар. Тип даних: INT.

Таблиця Product містить записи про продукти харчування та складається з таких полів:

- _id - унікальний ідентифікатор продукту. Тип даних: SERIAL (автоматично зростаючий номер).

- name - назва продукту. Тип даних: VARCHAR(255).

- description - опис продукту. Тип даних: TEXT.

- image - URL зображення продукту. Тип даних: VARCHAR(255).

- quantity - кількість продукту. Тип даних: INT.

- price - ціна продукту. Тип даних: FLOAT.

- categoryId - ідентифікатор категорії, до якої належить продукт. Тип даних: INT. Зовнішній ключ, що посилається на таблицю Category.

Таблиця Order містить записи про замовлення продуктів, що здійснені користувачами та містить такі поля:

- _id - унікальний ідентифікатор замовлення. Тип даних: SERIAL (автоматично зростаючий номер).

- purchaseDate - дата і час покупки. Тип даних: TIMESTAMP.

- userId - ідентифікатор користувача, який зробив замовлення. Тип даних: INT. Зовнішній ключ, що посилається на таблицю User.

Таблиця OrderProduct є проміжною таблицею, що зв’язує записи із замовленнями з продуктами та складається з таких полів:

- productId - ідентифікатор продукту. Тип даних: INT. Зовнішній ключ, що посилається на таблицю Product. Поле є частиною первинного ключа.

Таблиця Recipe містить в собі записи рецептів та складається з таких полів:

- _id - унікальний ідентифікатор рецепта. Тип даних: SERIAL (автоматично зростаючий номер).

- idMeal - унікальний ідентифікатор страви. Тип даних: VARCHAR(255).

- strMeal - назва страви. Тип даних: VARCHAR(255).

- strCategory - категорія страви. Тип даних: VARCHAR(255).

- strArea - регіон або країна походження страви. Тип даних: VARCHAR(255).

- strInstructions -інструкції по приготуванню страви. Тип даних: TEXT.

- strMealThumb - URL зображення страви. Тип даних: VARCHAR(255).

- strTags - теги страви. Тип даних: TEXT.

- strYoutube - URL відео з рецептом на YouTube. Тип даних: VARCHAR(255).

- strIngredients - інгредієнти страви, збережені у форматі тексту. Тип даних: TEXT.

Таблиця UserSavedRecipes містить збережені користувачем рецепти та складається з таких полів:

- userId - ідентифікатор користувача, який зберіг рецепт. Тип даних: INT. Зовнішній ключ, що посилається на таблицю User. Поле є частиною первинного ключа.

- recipeId - ідентифікатор рецепта. Тип даних: INT. Зовнішній ключ, що посилається на таблицю Recipe. Поле є частиною первинного ключа. Схема бази даних наведена на рисунку 3.1.

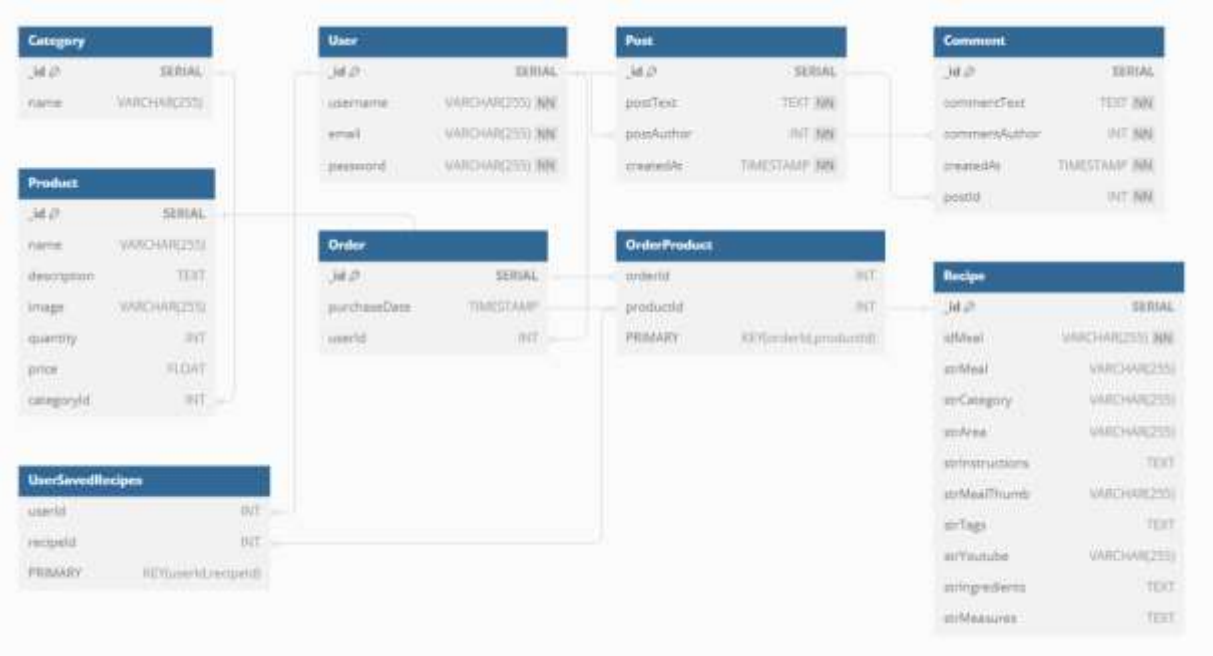


Рисунок 3.1 - Схема бази даних

Отже, було розроблено схему бази даних, в якій зберігатимуться дані про користувачів, рецепти, збережені рецепти, замовлені продукти, коментарі. Проведено детальний опис усіх таблиць та полів бази даних.

3.2. Розробка модуля авторизації та реєстрації користувача

Процес реєстрації користувача є базовим для отримання ним доступу до функціоналу системи. Реєстрація аккаунту користувача дозволяє прив’язати дані користувача до збережених ним в аккаунті даних в базі даних, що згодом дозволить отримати до них доступ при авторизації користувача в системі. При вході в аккаунт з будь-якого пристрою, дані будуть відображені користувачу, адже вони зберігаються на віддаленому сервері, а не на пристрої користувача. Було розроблено функціонал для авторизації та реєстрації користувача. Він об’єднаний у сервіс AuthService та складається з кількох функцій, що покривають всі випадки використання системи, що стосуються реєстрації та авторизації. Блок-схема алгоритму роботи функції перевірки токена авторизації наведена на рисунку 3.2.

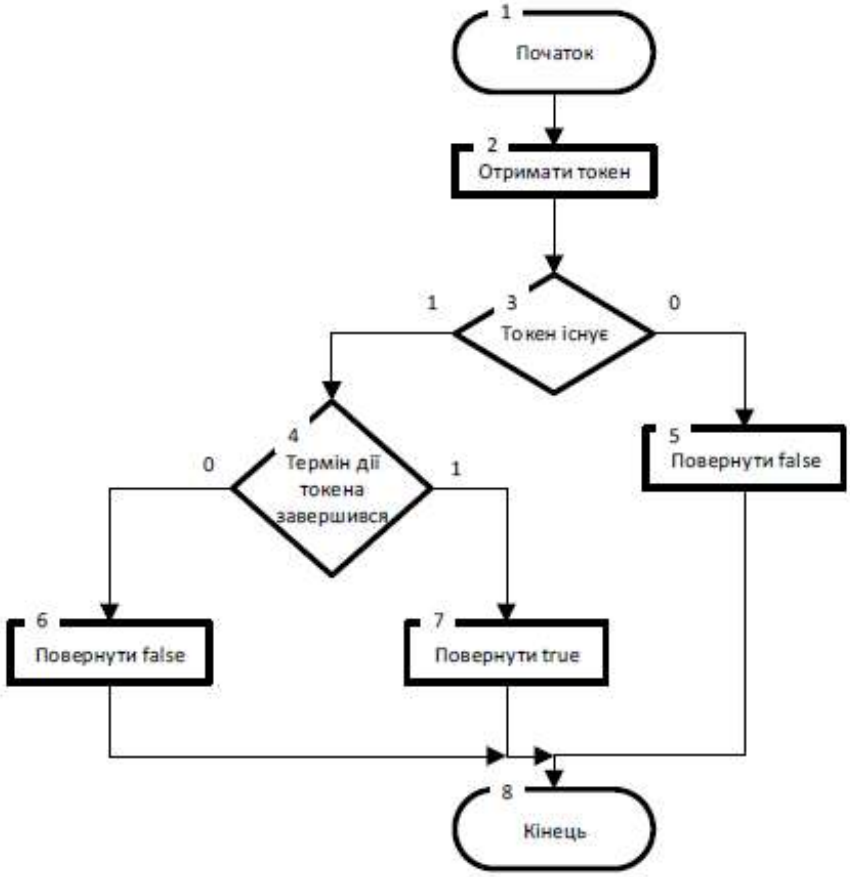


Рисунок 3.2 - Блок-схема алгоритму роботи функції перевірки токена авторизації

Спершу проводиться перевірка, чи вийшов термін дії токена авторизації користувача. Якщо токен не виявлено, тоді повертається значення false, адже в такому випадку неможливо перевірити, чи вийшов термін токена, якого не існує, а також те, що користувач ніколи не був авторизований у системі. Якщо токен існує та його термін дії не завершився, система повертає false. Якщо він завершився, тоді повертається true і в такому випадку генерується новий токен авторизації.

Наступний крок - отримання профілю користувача та його авторизація. Блок-схема алгоритму роботи функції для отримання профілю користувача та його токена авторизації наведена на рисунку 3.3.



Рисунок 3.3 - Блок-схема алгоритму роботи функції
для отримання профілю користувача

Після цього, проводиться авторизація користувача в профіль, збереження створеного токена авторизації в базу даних, перенаправлення користувача на головну сторінку. Було розроблено графічну схему сторінки реєстрації. Вона наведена на рисунку 3.4.

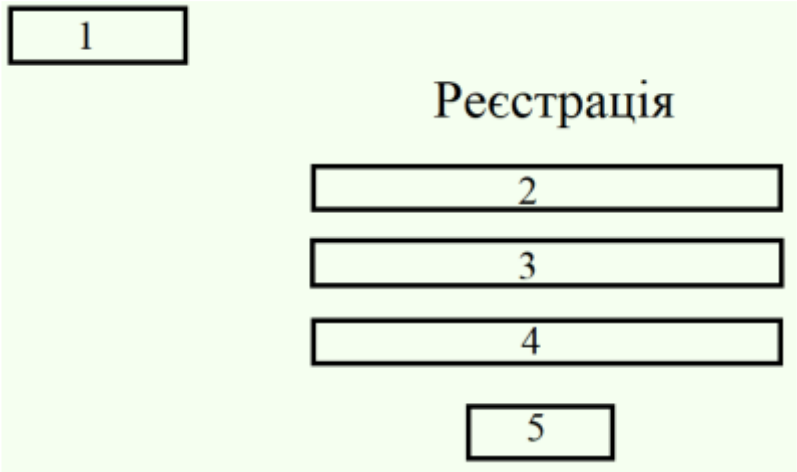


Рисунок 3.4 - Графічна схема сторінки реєстрації

Елементи сторінки реєстрації:

1 - Логотип застосунку.

2 - Поле імені користувача.

3 - Поле електронної пошти.

4 - Поле для паролю.

5 - Кнопка підтвердження реєстрації.

Також, було розроблено графічну схему сторінки авторизації, яка подана на рисунку 3.5. Вона дозволяє користувачу авторизуватися в застосунок.

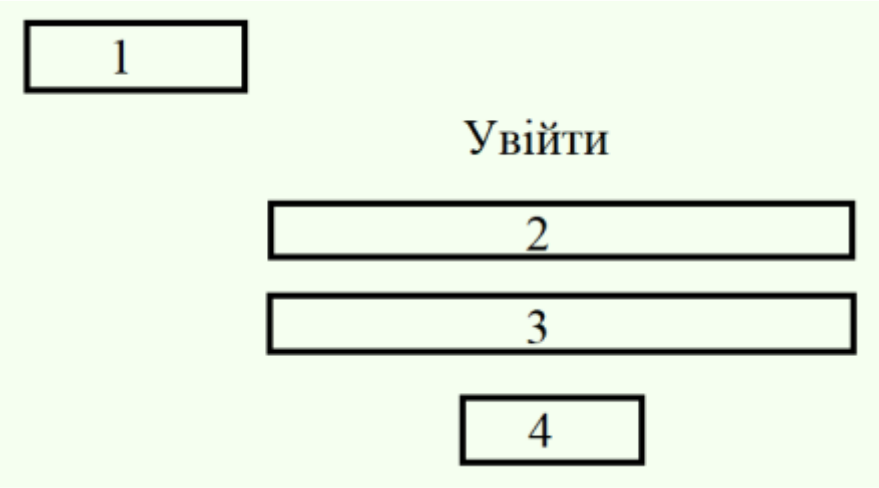


Рисунок 3.5 - Графічна схема сторінки авторизації

Елементи сторінки авторизації:

1 - Логотип.

2 - Поле електронної пошти.

3 - Поле для паролю.

4 - Кнопка «Увійти».

Для кращого розуміння можливих сценаріїв роботи та зв’язку між собою сторінок авторизації та реєстрації, було побудовано блок-схему, яка наведена на рисунку 3.6.

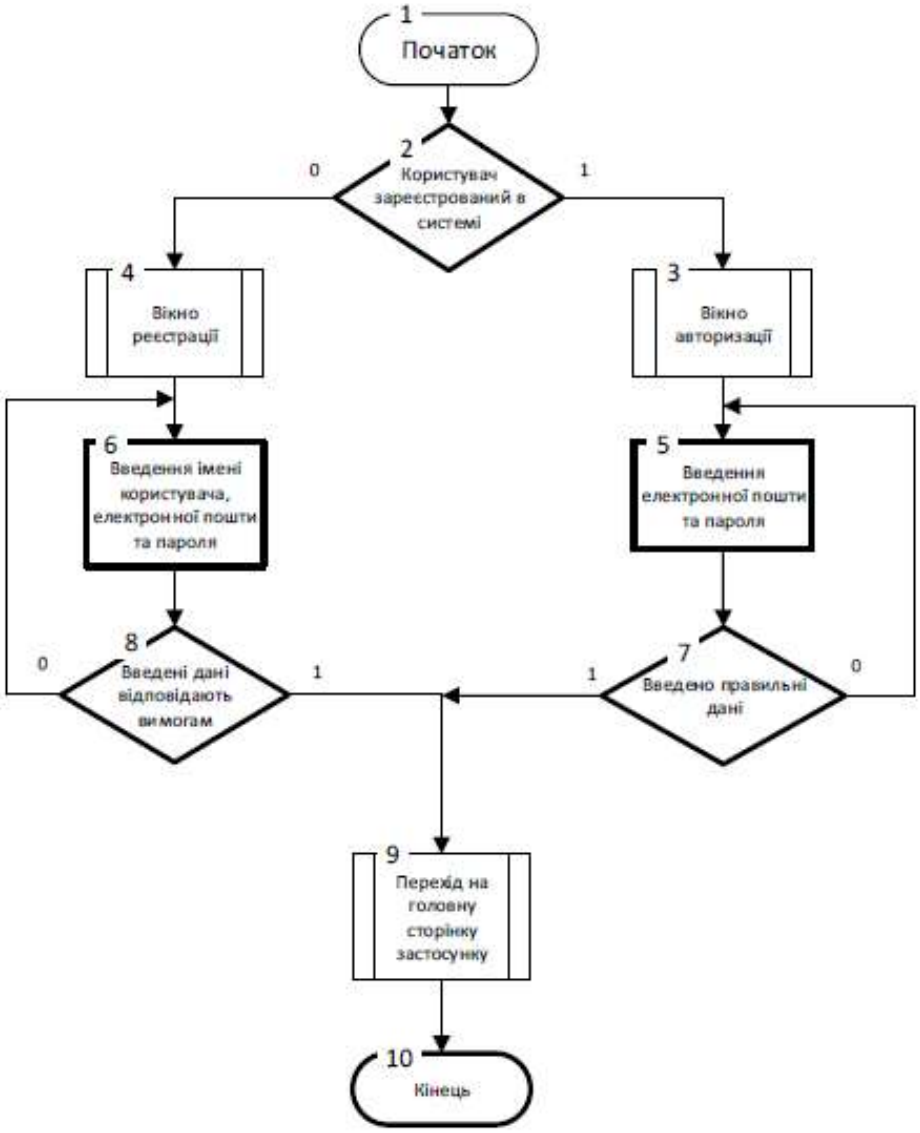


Рисунок 3.6 - Блок-схема алгоритму процесу реєстрації та авторизації.

Нижче наведено фрагмент програмного коду, який відповідає за процес реєстрації та авторизації.

```
const handleChange = (event) => {
const { name, value } = event.target;
setFormState({
...formState,
[name]: value,
});
```

```
};
const handleFormSubmit = async (event) => {
event.preventDefault();
console.log(formState);
try {
        const { data } = await login({

variables: { ...formState },
});
Auth.login(data.login.token);
const idMeals = data.login.user.savedRecipes.map((recipe) => {
return recipe.idMeal;
});
saveMealIds(idMeals);
} catch (e) {
console.error(e);
}
setFormState({
email: '',
password: '',
});
        };
```

3.3. Розробка головної вкладки та модуля перегляду рецептів веб-застосунку

Головна вкладка вебзастосунку є основною та запускається при переході на веб-сайт застосунку [18]. Вона надає доступ до інших вкладок застосунку. Найбільше місця на головній сторінці займає перелік рецептів, які відбираються системою випадковим чином. При кожному оновленні сторінки цей перелік

оновлюється. Також, у верхній частині робочої області є пошуковик рецептів за повною назвою або ж ключовими словами, а також за назвою інгредієнту, який міститься у рецептах. Окрім цього, додано два випадаючих списки, які оновлюють перелік запропованих рецепти на головній сторінці. Можна обирати серед певного переліку національних кухонь, а також за певними категоріями страв. Структурна схема інтерфейсу головної вкладки веб-застосунку наведено на рисунку 3.7.

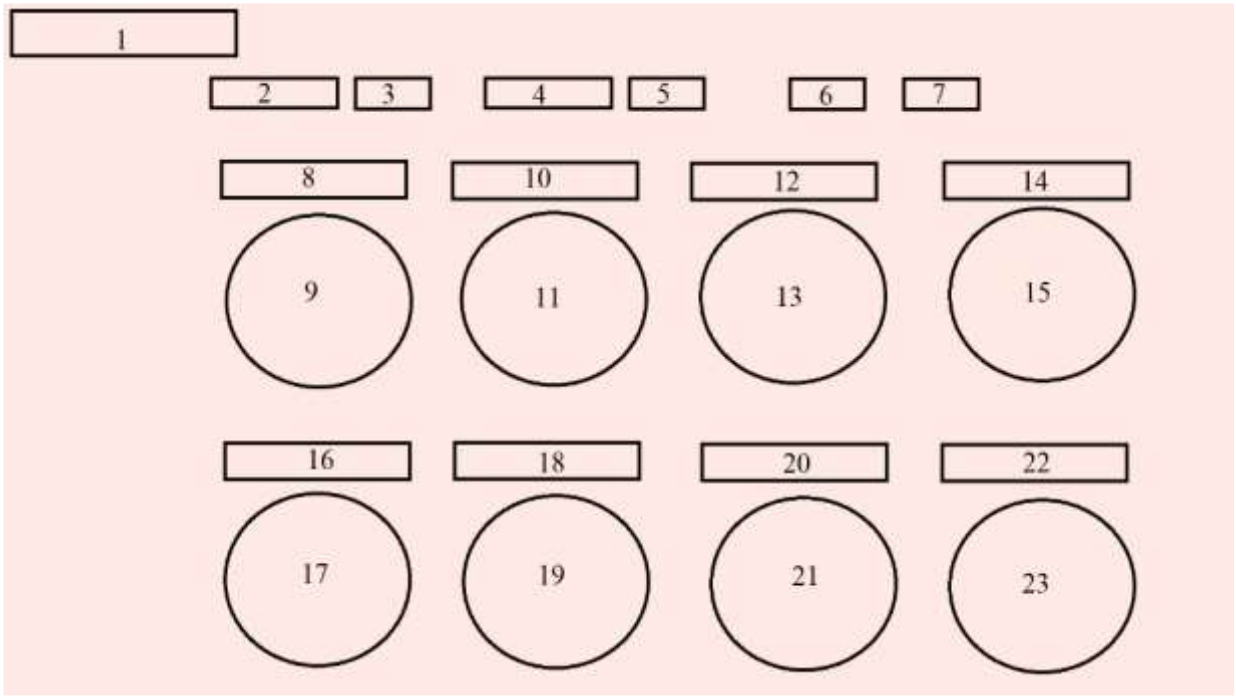


Рисунок 3.7 - Структурна схема інтерфейсу головної вкладки веб-застосунку

Елементи головної вкладки:

1 - Логотип.

2 - Поле для пошуку страв за ключовими словами.

3 - Кнопка пошуку рецептів за ключовими словами.

4 - Поле для пошуку страв за інгредієнтами.

5 - Кнопка пошуку рецептів за ключовими словами.

6 - Випадаючий список категорій страв.

7 - Випадаючий список національних кухонь.

8, 10, 12, 14, 16, 18, 20, 22 - назви страв на головній сторінці.

9, 11, 13, 15, 17, 19, 21, 23 - фото страв.

Кількість страв, що продемонстровані на головній сторінці, може відрізнятися. Це залежить від обсягу категорій, що обираються користувачем. За замовчуванням, при початковому запуску головної сторінки відображається десять випадкових рецептів. При пошуку за назвою, список оновлюється та відображає ті рецепти, які містять введені в пошуку слова у назві. При пошуку за інгредієнтами, відображаються ті рецепти, в яких міститься вказаний інгредієнт. Для вибору рецептів за категорією та за національною кухнею розроблено випадаючі списки. Випадаючі списки, також відомі як dropdown menus, є поширеним елементом інтерфейсу користувача в вебзастосунках. Випадаючі списки можуть бути реалізовані за допомогою різних веб- технологій, таких як HTML, CSS та JavaScript. У HTML зазвичай використовується елемент <select>, щоб створити основу випадаючого списку, в якому елементи <option> представляють окремі опції [19, 20]. CSS використовується для стилізації вигляду випадаючого списку, наприклад, кольору, розміру шрифту та візуальних ефектів при наведенні курсора або виборі опції. JavaScript може бути використано для додавання інтерактивності та динамічної поведінки, наприклад, автоматичного заповнення опцій на основі введених користувачем даних або зміни вмісту сторінки відповідно до обраної опції. Основною перевагою використання випадаючих списків є те, що вони дозволяють користувачам швидко та легко обирати потрібну опцію з великого набору варіантів. Це особливо корисно, коли є обмежений простір на сторінці або коли список опцій дуже довгий. Коли користувач розкриває цікавий йому випадаючий список та обирає необхідну йому категорію, список страв оновлюється аналогічно, як при пошуку за інгредієнтами чи ключовими словами. При натисканні на страву, відкривається нова вкладка з її описом, рецептом та переліком інгредієнтів. Розроблено блок-схему алгоритму відображення рецепту, яка наведена на рисунку 3.8.

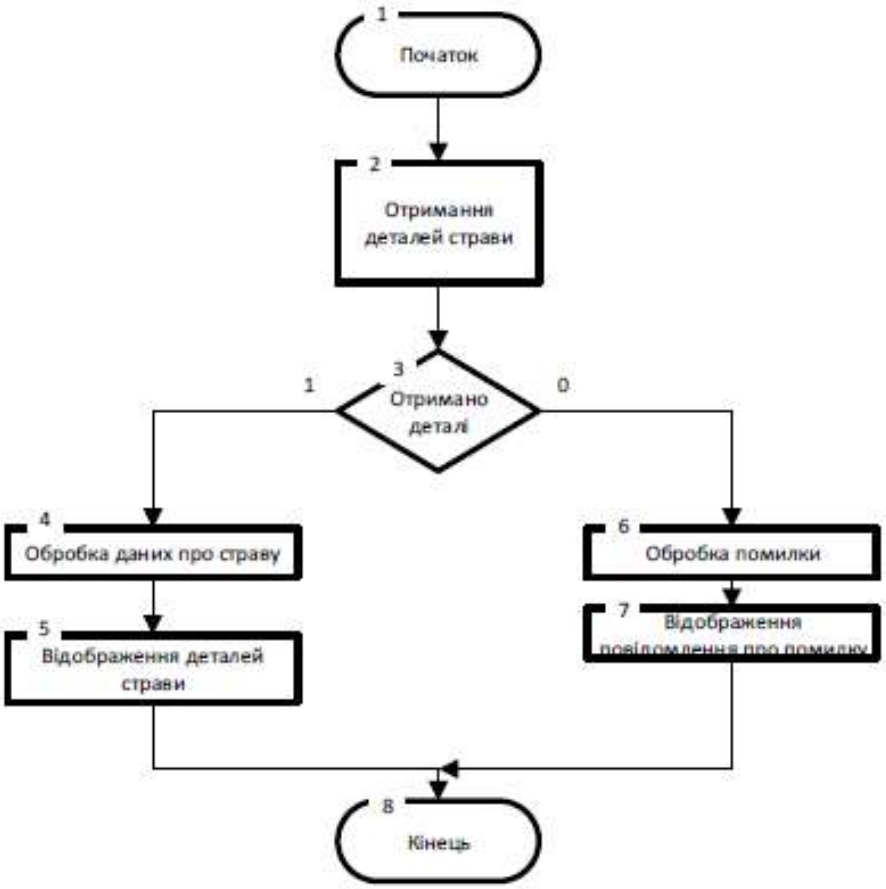


Рисунок 3.8 - Блок-схема алгоритму відображення рецепту

Структурна схема цієї вкладки наведена на рисунку 3.9.

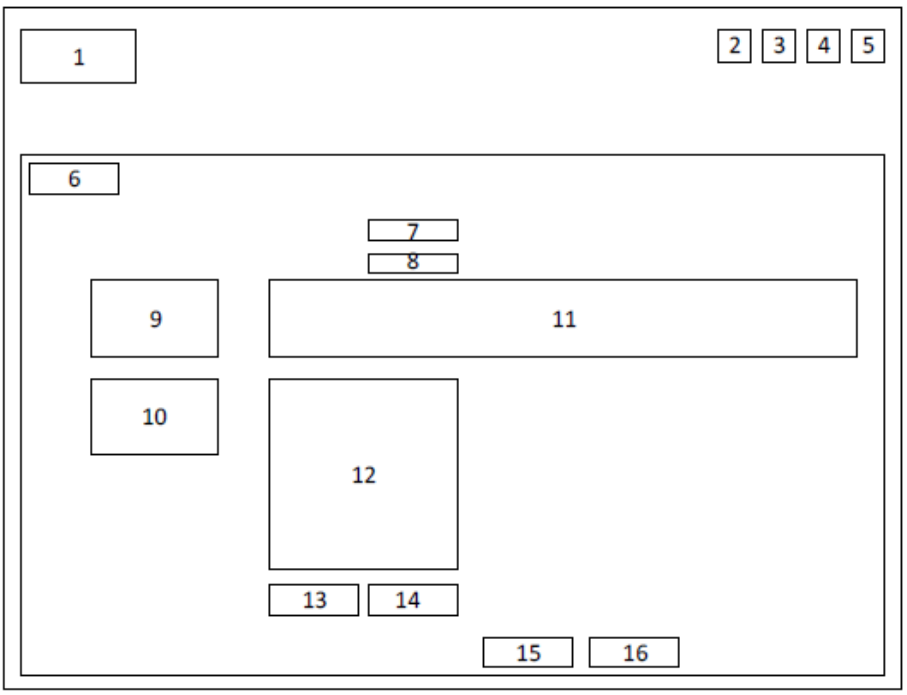


Рисунок 3.9 - Структурна схема вкладки перегляду рецепту

Окрім цього, розроблено блок-схему алгоритму роботи функції для додавання рецептів в «Улюблене» (рисунок 3.10).



Рисунок 3.10 - Блок-схема алгоритму роботи функції для додавання рецептів в «Улюблене»

Елементи вкладки перегляду рецепту:

1 - Логотип застосунку.

2 - Кнопка пошуку рецептів.

3 - Кнопка переходу в профіль користувача.

4 - Кнопка виходу з аккаунту.

5 - Кнопка переходу в кошик продуктів.

6 - Кнопка поверення на попередню сторінку.

7 - Назва страви.

8 - Категорія страви.

9 - Фото страви.

10 - Відео приготування страви на Youtube.

11 - Текст рецепту.

12 - Перелік інгредієнтів.

13 - Кнопка «Вибрати всі інгредієнти».

14 - Кнопка «Зняти вибір з усіх інгредієнтів».

15 - Кнопка «Зберегти рецепт».

16 - Кнопка «Додати продукти до кошика».

Висновок

В результаті виконання кваліфікаційної роботи бакалавра, було розроблено веб-застосунок для підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів. Розроблений веб-застосунок вирішує дві основні проблеми: складний процес пошуку рецептів та недоліки у процесі придбання необхідних продуктів для приготування страв. Для розробки було використано середовище розробки Visual Studio Code, мову програмування JavaScript.

Проведено аналіз веб-застосунків для підбору рецептів за інгредієнтами, порівняльний аналіз аналогів, аналіз методів розв’язання завдання. На їх основі було доведено доцільність та актуальність власної розробки, поставлено задачі дослідження. Розглянуто підходи до проектування веб-застосунків та визначено переваги та недоліки кожного підходу. Також проаналізовано концепції створення баз даних і конкретні реалізації систем управління базами даних, з яких було обрано оптимальний підхід для цієї розробки.

Проведено аналіз даних, які генеруються, використовуються, завантажуються та обробляються при роботі вебзастосунку для підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів. Було розроблено модель вебзастосунку, метод пошуку рецептів, метод замовлення продуктів, загальний алгоритм роботи та діаграми станів.

Проведено варіантний аналіз та здійснено вибір засобів реалізації веб-застосунку для підбору рецептів за інгредієнтами з можливістю придбання необхідних продуктів. Розроблено блок-схеми алгоритмів та структурні схеми інтерфейсу роботи модуля авторизації та реєстрації користувача, блок-схеми алгоритмів роботи та структурні схеми інтерфейсу головної вкладки, вкладки перегляду рецепту веб-застосунку.

Продемонстровано функціонал розробленого веб-застосунку та його здатність виконувати поставлені на початку розробки задачі. Визначено мінімальні та рекомендовані системні вимоги для роботи з веб-застосунком.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Розробка веб-додатків [Електронний ресурс]. Режим доступу: https://en.yeeply.com/blog/6-different-kinds-веб-app-development/#definition

2. Cookpad: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://cookpad.com/ua/homepage

3. Yummly: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://www.yummly.com/

4. Mealime: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://www.mealime.com/

5. MERN Stack explained: веб-сайт [Електронний ресурс] – Режим доступу до ресурсу: https://www.mongodb.com/resources/languages/mern-stack

6. MongoDB: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://www.mongodb.com/

7. Express.js: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://expressjs.com/

8. React: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://legacy.reactjs.org/

9. Node.js: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://nodejs.org/en

10. Stripe: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://stripe.com/

11. TheGraphQL: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://thegraph.com/docs/en/querying/graphql-api/

12. JavaScript: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://w3schoolsua.github.io/js/index.html#gsc.tab=0

13. TypeScript: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://www.typescriptlang.org/

14. PHP: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://www.php.net/

15. Visual Studio Code: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://code.visualstudio.com/

16. Web application testing: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://qatestlab.com/solutions/by-focus-area/web-application-testing/

17. Automated software testing: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://www.atlassian.com/continuous-delivery/softwaretesting/automated-testing

18. Non functional testing: веб-сайт [Електронний ресурс] - Режим доступу до ресурсу: https://www.perforce.com/blog/alm/what-non-functional-testing

19. Roger Pressman. Software Engineering: A Practitioner's Approach. New-York: McGraw Hill, 2023. 704с.

20. Cen Kaner. Lessons Learned in Software Testing. Wiley, 2022. - 320с.