**ЗМІСТ**

[1 ВСТУП 2](#_Toc201344909)

[1.1 Огляд програмного забезпечення 2](#_Toc201344910)

[1.2 Мета роботи 2](#_Toc201344911)

[1.3 Межі 3](#_Toc201344912)

[2 ЗАГАЛЬНИЙ ОПИС 4](#_Toc201344913)

[2.1 Перспективи продукту 4](#_Toc201344914)

[2.2 Функції продукту 5](#_Toc201344915)

[2.3 Характеристики користувачів 6](#_Toc201344916)

[2.4 Загальні обмеження 6](#_Toc201344917)

[2.5 Припущення та залежності 7](#_Toc201344918)

[3 КОНКРЕТНІ ВИМОГИ 9](#_Toc201344919)

[3.1 Вимоги до зовнішніх інтерфейсів 9](#_Toc201344920)

[3.1.1 Інтерфейс користувача 9](#_Toc201344921)

[3.1.2 Програмний інтерфейс 13](#_Toc201344922)

[3.1.3 Комунікаційний інтерфейс 13](#_Toc201344923)

[3.1.4 Обмеження пам’яті 14](#_Toc201344924)

[3.1.5 Функції продукту 14](#_Toc201344925)

[3.1.6 Припущення та залежності 15](#_Toc201344926)

[3.2. Властивості програмного продукту 16](#_Toc201344927)

[3.3 Атрибути програмного продукту 19](#_Toc201344928)

[3.3.1 Надійність 19](#_Toc201344929)

[3.3.2 Доступність 19](#_Toc201344930)

[3.3.3 Безпека 19](#_Toc201344931)

[3.3.4 Супроводжуваність 20](#_Toc201344932)

[3.3.5 Переносимість 20](#_Toc201344933)

[3.3.6 Продуктивність 20](#_Toc201344934)

[3.4 Вимоги бази даних 20](#_Toc201344935)

# **1 ВСТУП**

## 1.1 Огляд програмного забезпечення

Кулінарія є важливою частиною повсякденного життя, а приготування їжі часто вимагає не лише навичок, а й креативності, часу та планування. У сучасному світі, де значна частина користувачів має обмежений час і водночас прагне дотримуватися індивідуальних дієтичних уподобань, виникає потреба в інтелектуальних системах, що допомагають у виборі відповідних страв.

Система персоналізованого підбору кулінарних рецептів є прикладом сучасного веб-застосунку, орієнтованого на індивідуальні потреби користувача. Вона дозволяє користувачам зберігати свої вподобання (тип дієти, небажані інгредієнти, алергени), переглядати рецепти, відмічати улюблені страви, фільтрувати їх за різними параметрами (складність, кухня, час приготування), а також отримувати персоналізовані рекомендації.

У процесі розробки застосунку було реалізовано механізми авторизації та реєстрації з використанням JWT, створено інтерфейси для різних ролей користувачів (звичайний користувач, модератор, адміністратор), реалізовано REST API для взаємодії з базою даних MongoDB та розроблено клієнтську частину за допомогою React. Система поєднує у собі простоту використання, гнучкість налаштувань та ефективні алгоритми фільтрації, рекомендацій, що дозволяють користувачам легко знаходити відповідні рецепти згідно зі своїми потребами.

## 1.2 Мета роботи

Метою роботи є проєктування та реалізація веб-системи, яка забезпечує персоналізований підбір кулінарних рецептів з урахуванням індивідуальних уподобань користувача. Система має надавати користувачам можливість зручно шукати та фільтрувати рецепти, зберігати улюблені страви, встановлювати параметри дієти, алергенів та небажаних інгредієнтів, а також отримувати рекомендації на основі заданих вподобань.

Досягнення цієї мети передбачає реалізацію сучасної архітектури веб-застосунку, інтеграцію бази даних, побудову API для взаємодії клієнта з сервером, а також створення інтуїтивного інтерфейсу для кінцевих користувачів.

## 1.3 Межі

Розроблена веб-система орієнтована на забезпечення базового функціоналу підбору кулінарних рецептів відповідно до індивідуальних вподобань користувача. Межі системи визначаються обсягом реалізованих функцій і не включають складні алгоритми машинного навчання чи глибокої персоналізації на основі історії активностей користувача.

У межах даного проєкту реалізовано авторизацію, збереження вподобань, перегляд, пошук, фільтрацію та додавання рецептів до улюблених. Система не передбачає підтримку мобільних платформ, багатомовності чи адміністрування великої кількості користувачів одночасно. Також відсутня можливість додавання рецептів самими користувачами, усі дані попередньо внесені в базу.

Таким чином, система є прототипом, який демонструє основні можливості персоналізованого підбору рецептів, і може бути основою для подальшого масштабування та розвитку.

# **2 ЗАГАЛЬНИЙ ОПИС**

## 2.1 Перспективи продукту

Система персоналізованого підбору кулінарних рецептів створена з метою підвищення зручності у виборі страв, враховуючи індивідуальні харчові вподобання користувача. Продукт орієнтований на широку аудиторію, від користувачів із особливими дієтичними потребами до звичайних користувачів, які шукають нові ідеї для приготування їжі.

У перспективі, програмна система має значний потенціал для подальшого розвитку та вдосконалення. Одним із основних напрямів є розширення платформи шляхом створення мобільного застосунку. Це дозволить користувачам отримувати доступ до рекомендацій і взаємодіяти з системою не лише через браузер, а й на смартфонах та планшетах. Мобільний клієнт забезпечить більшу гнучкість, спростить доступ до рецептів у повсякденному житті (наприклад, під час покупок або приготування їжі), а також сприятиме збільшенню аудиторії продукту.

Ще одним важливим кроком є впровадження алгоритмів машинного навчання для покращення персоналізованих рекомендацій. Замість простих фільтрів, що базуються на збігах уподобань, система зможе аналізувати попередні дії користувача, враховувати оцінки рецептів, час перегляду, обрані страви тощо. Це дозволить створити більш точні та релевантні рекомендації з урахуванням індивідуальних патернів поведінки користувачів. Алгоритми кластеризації, колаборативної фільтрації чи нейронні мережі зможуть суттєво підвищити якість сервісу.

Крім того, підтримка багатомовного інтерфейсу стане важливим кроком до розширення цільової аудиторії. Додавання таких мов, як англійська, німецька, французька чи польська, дозволить охопити користувачів з інших країн, забезпечити більшу доступність функціоналу і зробити продукт придатним до масштабування на міжнародному рівні.

Також перспективним є розширення бази даних рецептів, як за рахунок додавання нових страв вручну адміністраторами або модераторами, так і через створення інтерфейсу для надсилання рецептів від самих користувачів. Це дозволить сформувати живу спільноту навколо платформи та підвищить її актуальність.

У майбутньому система може також включати нові функції: створення списку покупок, календар харчування, збереження історії приготувань, інтеграцію з соціальними мережами для обміну рецептами, що значно підвищить залученість користувачів і зробить платформу багатофункціональним інструментом у сфері кулінарії.

## 2.2 Функції продукту

Програмна система забезпечує користувачів основними функціями, спрямованими на персоналізований підбір кулінарних рецептів. Після авторизації користувач має змогу заповнити власні уподобання: обрати тип дієти, зазначити алергени та небажані інгредієнти. Ця інформація зберігається в системі та використовується для подальшого формування рекомендацій.

Ключовою функціональністю продукту є автоматичний підбір рецептів, що відповідають вподобанням користувача. Система фільтрує рецепти за заданими параметрами, виключаючи ті, що містять небажані компоненти, та формує добірку релевантних страв. Також реалізовано функцію пошуку рецептів та фільтрацію за категоріями, що дозволяє зручно орієнтуватися в каталозі.

Користувачі можуть додавати рецепти до обраного списку, формуючи власну колекцію улюблених страв.

Для адміністраторів та модераторів система надає додаткові функції керування. Модераторам доступні можливості додавання нових рецептів, включаючи введення назви, опису, списку інгредієнтів, інструкцій з приготування та зображення, їх редагування та видалення рецептів. Окремо передбачено панель адміністратора для керування користувачами, включаючи призначення (створення) нових адміністраторів та модераторів ситеми.

Усі функції реалізовано через вебінтерфейс, з інтерактивним відгуком на дії користувача, що забезпечує інтуїтивну роботу з системою.

## 2.3 Характеристики користувачів

Програмна система орієнтована на кілька категорій користувачів, кожна з яких має власні права доступу та функціональні можливості. Основними типами користувачів є звичайні користувачі, модератори та адміністратори. Для кожної з ролей реалізовано відповідний інтерфейс та набір функцій, що відповідають потребам конкретної категорії.

Звичайний користувач − це основна категорія, яка взаємодіє із системою з метою підбору кулінарних рецептів. Він має змогу проходити реєстрацію, авторизуватись, вказувати власні вподобання щодо дієти, алергенів і небажаних інгредієнтів, переглядати рецепти, шукати їх, фільтрувати, додавати до улюблених, а також створювати нові рецепти. Інтерфейс для звичайного користувача є максимально інтуїтивним та приємним для взаємодії.

Модератор, користувач із додатковими правами, відповідальний за створення контенту (рецептів). Модератор також має можливість редагувати або видаляти рецепти, стежити за відповідністю даних загальним вимогам до контенту. Інтерфейс модератора розширено відповідними інструментами для зручного керування рецептами.

Адміністратор, користувач із повними правами доступу, який має можливість керування всіма користувачами системи, в тому числі він може призначати та змінювати ролі, редагувати облікові записи, видаляти користувачів.

## 2.4 Загальні обмеження

Розроблена система має такі загальні обмеження:

* функціонування системи залежить від стабільного підключення до мережі Інтернет;
* вебзастосунок оптимізований для сучасних браузерів (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari);
* повний функціонал доступний лише після проходження авторизації;
* кожна адреса електронної пошти може бути використана лише для створення одного облікового запису;
* інтерфейс реалізовано лише українською мовою (багатомовна підтримка не реалізована);
* система доступна лише у вебверсії (мобільна версія або мобільний застосунок відсутні);
* для збереження авторизаційного стану та вподобань використовується локальне сховище браузера (localStorage);
* у системі наразі не реалізовано автоматичне резервне копіювання бази даних;
* база рецептів є обмеженою й потребує поступового розширення для підвищення якості персоналізованих рекомендацій.

## 2.5 Припущення та залежності

У процесі розробки вебсистеми було визначено низку припущень, а також виявлено залежності, що впливають на її функціонування:

* передбачається, що користувач має базові навички роботи з веббраузерами та розуміє принципи реєстрації, авторизації та налаштування персональних параметрів;
* вважається, що користувач вводить достовірну та коректну інформацію при створенні профілю та налаштуванні вподобань;
* система покладається на наявність стабільного з’єднання з Інтернетом для обміну даними з сервером;
* функціональність зберігання сесії користувача базується на коректній роботі localStorage у браузері користувача;
* серверна частина залежить від роботи MongoDB як основного джерела збереження даних;
* для авторизації та перевірки прав доступу передбачається безперебійна генерація й обробка JWT-токенів;
* клієнтська частина взаємодіє з REST API, і залежить від стабільної роботи відповідних маршрутів серверного застосунку;
* актуальність та коректність роботи рекомендаційної системи залежить від повноти даних у базі рецептів та від наявності збережених вподобань у користувача;
* безпека паролів користувачів забезпечується через використання бібліотеки bcryptjs, що передбачає наявність цієї залежності на сервері;
* припускається, що оновлення структури бази даних або серверної логіки буде здійснюватися з дотриманням зворотної сумісності.

# **3 КОНКРЕТНІ ВИМОГИ**

## 3.1 Вимоги до зовнішніх інтерфейсів

### 3.1.1 Інтерфейс користувача

Інтерфейс користувача вебсистеми створений із використанням технологій HTML, CSS, JavaScript та фреймворку React. Основна увага приділялася забезпеченню простоти, доступності та інтуїтивно зрозумілої навігації для користувачів з різними ролями (звичайний користувач, модератор, адміністратор).

На головній сторінці відображається список доступних кулінарних рецептів (див. рис. 3.1). Кожен рецепт представлений у вигляді картки з назвою, зображенням, описом, тривалістю приготування та позначками (тегами). Користувач може скористатися полем пошуку для знаходження рецептів за назвою або ключовими словами. Також доступна фільтрація за категорією страви, складністю, часом приготування, типом кухні тощо. Якщо користувач зареєстрований, у кожної картки рецепту буде іконка у вигляді серця для додавання рецепта до улюблених.

Зображення, що містить текст, знімок екрана, овоч, їжа

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.1 – Скріншот головної сторінки веб-системи (рисунок виконаний самостійно)

Сторінки авторизації та реєстрації дозволяють користувачеві увійти до системи або створити новий акаунт. Для авторизації потрібно ввести email та пароль (див. рис. 3.2). У разі успішного входу користувача перенаправляє до відповідної сторінки згідно з його роллю. Реєстрація передбачає введення імені, email, пароля та підтвердження згоди на обробку персональних даних. Пароль зберігається у хешованому вигляді.

Зображення, що містить текст, знімок екрана, дизайн

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.2 – Скріншот сторінки входу (рисунок виконаний самостійно)

На сторінці профілю, користувач може переглядати свої персональні дані (ім’я, email) та змінювати налаштування вподобань: вказати тип дієти, перелік алергенів та інгредієнтів, які не бажає бачити в рецептах. Ці параметри використовуються системою для формування персоналізованих рекомендацій.

Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.3 – Скріншот сторінки профілю користувача (рисунок виконаний самостійно)

Інтерфейс сторінки вподобань зосереджений на формуванні добірок рецептів за користувацькиими вподобаннями.

Зображення, що містить текст, їжа, меню, Закуска

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.4 – Скріншот сторінки рекомендацій (рисунок виконаний самостійно)

Сторінка улюблених рецептів дозволяє користувачеві швидко переглядати всі рецепти, які він раніше позначив як улюблені. Рецепти виводяться у вигляді карток, де можна, натиснувши на хрестик у картці рецепту, прибрати рецепт зі списку.

Зображення, що містить текст, знімок екрана, Закуска, їжа

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.5 – Скріншот сторінки з улюбленими рецептами (рисунок виконаний самостійно)

На сторінці адміністратора (доступна лише для ролі “admin”) є можливість перегляду спису усіх зареєстрованих користувачів, редагування їхніх даних та видалення акаунтів (див. рис. 3.6). Також можна додавати нових користувачів до системи. Панель має форму таблиці з функцією фільтрації за роллю.

Зображення, що містить текст, знімок екрана, число, Шрифт

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.6 – Скріншот сторінки управління користувачами (рисунок виконаний самостійно)

Сторінка модератора (див. рис. 3.7) призначена для користувачів з роллю “moderator”. Дозволяє переглядати, додавати, редагувати та видаляти рецепти. При додаванні рецепта модератор може вказати назву, опис, список інгредієнтів, категорію, час приготування, рівень складності, зображення. Для редагування використовується форма з попередньо заповненими полями. Видалення рецепта супроводжується підтвердженням дії.

Зображення, що містить текст, знімок екрана, число, Шрифт

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.7 – Скріншот сторінки модератора (рисунок виконаний самостійно)

### 3.1.2 Програмний інтерфейс

Програмна система побудована за клієнт-серверною архітектурою, де комунікація між клієнтською частиною (вебінтерфейсом) та серверною логікою здійснюється через програмний інтерфейс прикладного програмування (API). Взаємодія відбувається з використанням REST-підходу та передачі даних у форматі JSON, що забезпечує простоту інтеграції та масштабованість системи.

Програмний інтерфейс забезпечує можливості для:

* обробки запитів авторизації та реєстрації користувачів;
* збереження та оновлення персональних даних користувачів;
* керування вподобаннями користувачів;
* роботи з кулінарними рецептами: їх створення, редагування, перегляд, видалення, пошук та фільтрація;
* додавання та перегляду обраних (улюблених) рецептів користувача;
* розмежування доступу до функцій системи на основі ролей (звичайний користувач, модератор, адміністратор);
* формування персоналізованих рекомендацій на основі введених вподобань.

Доступ до функціоналу, пов’язаного з обліковими записами та налаштуванням користувача, здійснюється з використанням механізму токенів автентифікації (JWT), що дозволяє безпечно зберігати сесію та ідентифікувати користувача протягом роботи із системою.

Інтерфейс також підтримує валідацію введених даних та повідомлення про помилки для підвищення зручності та захищеності взаємодії між клієнтом і сервером.

### 3.1.3 Комунікаційний інтерфейс

Комунікація між клієнтською частиною системи (вебінтерфейс, реалізований за допомогою React) та серверною частиною (створеною на основі Node.js та Express) здійснюється через протокол HTTP(S). Усі запити до серверу формуються відповідно до принципів REST-архітектури та передаються у форматі JSON.

Основні аспекти комунікації включають:

* Передачу запитів та відповідей між клієнтом і сервером із використанням методів GET, POST, PUT, DELETE для виконання CRUD-операцій.
* Автентифікацію та авторизацію користувача за допомогою JWT-токенів, які зберігаються на клієнті (у localStorage) та передаються в заголовках HTTP-запитів (Authorization header).
* Валідацію запитів і відповідей на стороні сервера для забезпечення цілісності та безпеки переданих даних.

Уся комунікація відбувається за захищеним протоколом HTTPS, що гарантує конфіденційність переданої інформації між клієнтською частиною та сервером.

### 3.1.4 Обмеження пам’яті

Для серверного застосунку обмеження пам’яті є мінімальними, оскільки система розгортається на окремому сервері, ресурси якого можуть масштабуватися відповідно до навантаження.

Для веб-застосунку пам’ять обмежується можливостями браузера, в якому він виконується. Основні дані (наприклад, JWT-токен, уподобання користувача) тимчасово зберігаються у localStorage, обсяг якого, як правило, не перевищує 5–10 МБ.

### 3.1.5 Функції продукту

Розроблена система забезпечує користувачам широкий набір операцій, спрямованих на взаємодію з кулінарними рецептами та персоналізацію користувацького досвіду. Основною функцією є можливість реєстрації та авторизації користувачів з подальшою перевіркою їхніх даних через механізм JWT-аутентифікації. Після входу в систему користувач отримує доступ до головної сторінки, де представлено повний перелік доступних рецептів. Інтерфейс дозволяє виконувати пошук рецептів за ключовими словами, а також фільтрацію за такими параметрами, як тип кухні, складність приготування чи основні інгредієнти.

Користувач має змогу переглядати свій профіль, налаштовувати вподобання: обирати тип дієти, зазначати алергени чи небажані інгредієнти. На основі цих параметрів система формує персоналізовані рекомендації, які дозволяють швидше знаходити відповідні рецепти. Також реалізована функціональність додавання рецептів до списку улюблених, що дає змогу зберігати вподобані страви для подальшого використання.

Для адміністраторів реалізовано функції перегляду списку користувачів, їх редагування та видалення. Модератори мають змогу додавати нові рецепти до системи, а також редагувати або видаляти наявні. Усі операції реалізовані з урахуванням ролей користувачів та відповідного розмежування прав доступу, що забезпечує безпечну та структуровану взаємодію з системою.

### 3.1.6 Припущення та залежності

У процесі розробки вебсистеми було сформовано низку припущень і залежностей.

Передбачається, що користувач володіє базовими навичками взаємодії з веббраузерами, здатний зареєструватися, авторизуватися, налаштувати вподобання та користуватися функціоналом інтерфейсу без сторонньої допомоги. При цьому система очікує, що користувач вводить достовірну інформацію під час реєстрації та встановлення персональних налаштувань.

Система залежить від стабільного з’єднання з Інтернетом, оскільки всі дані обробляються сервером, а вебклієнт постійно взаємодіє з ним через REST API. Відповідно, функціональність вебінтерфейсу також залежить від підтримки localStorage у браузері для збереження JWT-токена та інших даних користувача.

Зі сторони сервера проєкт використовує MongoDB як основне сховище даних. Для захисту облікових записів під час авторизації застосовується бібліотека bcryptjs, яка забезпечує надійне хешування паролів. Генерація та перевірка прав доступу базуються на JWT-токенах, що також є критичною залежністю.

Рекомендаційна система функціонує на основі аналізу збережених вподобань користувача. Її ефективність напряму залежить від повноти бази рецептів, актуальності параметрів вподобань та коректного відбору релевантних даних.

Клієнтська частина побудована з використанням React і взаємодіє з сервером, реалізованим на Node.js з Express, тому працездатність програми залежить також від підтримки відповідного середовища виконання

Передбачається, що всі оновлення структури бази даних або серверної логіки здійснюватимуться з дотриманням зворотної сумісності, щоб уникнути порушення існуючого функціоналу.

## 3.2. Властивості програмного продукту

Розроблена система персоналізованого підбору кулінарних рецептів реалізована на основі клієнт-серверної архітектури, у якій клієнтська частина (вебінтерфейс) взаємодіє з серверною частиною через REST API. Комунікація між компонентами відбувається з використанням протоколу HTTPS, що гарантує захищену передачу даних.

Усі функції системи згруповані в окремі логічні підсистеми: управління користувачами, вподобаннями, рецептами та рекомендаціями. Вони реалізовані у вигляді сервісів на сервері, кожен з яких відповідає за обмежену область відповідальності. Це забезпечує модульність, спрощує подальше тестування та розширення функціональності.

Для покращення зручності та безпеки використано такі рішення:

* JWT-аутентифікація: кожен користувач після авторизації отримує токен, який додається до всіх наступних запитів;
* Валідація та обробка помилок: як на клієнтській, так і на серверній стороні впроваджено перевірку введених даних;
* Збереження вподобань: унікальні вподобання користувача (дієта, алергени, небажані інгредієнти) використовуються для персоналізації добірки рецептів.

Одним із основних сценаріїв у системі є пошук кулінарних рецептів на основі ключових слів або параметрів фільтрації. Діаграма активності на рисунку 3.1 ілюструє типовий процес взаємодії між вебклієнтом, сервером та базою даних. Спочатку користувач відкриває розділ пошуку рецептів у вебінтерфейсі. Далі він вводить ключові слова або назву страви, а також задає параметри фільтрації (тип кухні, складність, час приготування тощо). Після натискання кнопки «Знайти», вебклієнт формує HTTP GET-запит до серверного API. Сервер обробляє запит і формує відповідний запит до бази даних MongoDB із врахуванням переданих параметрів. База даних виконує пошук рецептів, що відповідають умовам, і повертає результати серверу. Сервер обробляє результати й надсилає їх клієнту у форматі JSON. У результаті, користувач отримує список рецептів. Якщо результати знайдено − він може переглянути деталі обраної страви та зберегти її в улюблене. Якщо за запитом жодного рецепта не знайдено − система відображає відповідне повідомлення.

Зображення, що містить текст, схема, знімок екрана, Паралель

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.1 – Діаграма активності: пошук рецептів із фільтрацією (рисунок виконаний самостійно)

Іншою ключовою функціональністю вебсистеми є можливість надання персоналізованих рекомендацій кулінарних рецептів на основі вподобань користувача. На рисунку 3.2 зображено поетапну взаємодію між вебклієнтом і сервером.

Першим кроком користувач ініціює запит, натискаючи відповідну кнопку в інтерфейсі системи. Далі сервер приймає запит, витягує з бази даних актуальні вподобання користувача, які включають дієту, алергени та небажані інгредієнти. На основі цих даних формується список інгредієнтів, які повинні бути виключені з результатів. Сервер виконує пошук у базі даних, відкидаючи рецепти з небажаними компонентами, і формує добірку відповідних страв. Список рекомендованих рецептів повертається на клієнтську частину у форматі JSON. Вебінтерфейс відображає результати користувачу у вигляді карток рецептів.

Зображення, що містить текст, знімок екрана, Шрифт, квитанція

Вміст, створений ШІ, може бути неправильним.

Рисунок 3.2 – Діаграма активності: формування персональних рекомендацій (рисунок виконаний самостійно)

Цей механізм дозволяє системі адаптуватися до індивідуальних потреб користувача, що суттєво підвищує її практичну цінність. Крім того, така реалізація дає змогу легко розширити функціональність у майбутньому, наприклад, вдосконалити логіку добору на основі історії взаємодій.

Загальна структура системи дозволяє розширення її функціоналу, зокрема, впровадження механізмів зворотного зв’язку на рекомендації (оцінка їх відповідності), інтеграцію з календарем харчування чи формуванням списку покупок, додавання мобільного клієнта без зміни основного бекенду.

## 3.3 Атрибути програмного продукту

### 3.3.1 Надійність

Система забезпечує стабільну роботу за рахунок обробки помилок на сервері, валідації введених даних та захисту від збоїв при запитах до бази даних. У разі виникнення помилок користувач отримує зрозуміле повідомлення без припинення роботи застосунку.

### 3.3.2 Доступність

Вебзастосунок доступний цілодобово через браузер, за умови наявності стабільного інтернет-з’єднання. Вся функціональність відкривається після авторизації користувача.

### 3.3.3 Безпека

Для захисту даних реалізовано JWT-аутентифікацію, шифрування паролів (bcrypt), обмеження доступу за ролями, а також перевірку даних на обох сторонах (клієнт/сервер). Комунікація розгорнутої системи відбувається через захищений протокол HTTPS.

### 3.3.4 Супроводжуваність

Архітектура системи модульна, кожен компонент ізольований функціонально, що спрощує внесення змін, додавання нового функціоналу або оновлення залежностей. У коді дотримано структурованого стилю та іменування.

### 3.3.5 Переносимість

Застосунок легко переноситься на інші сервери чи хмарні середовища, оскільки використовує популярні технології (React, Node.js, MongoDB) та не має залежності від ОС.

### 3.3.6 Продуктивність

Система демонструє достатній рівень продуктивності для обробки запитів у реальному часі в межах обсягу демонстраційних даних. Відповіді на запити (пошук, фільтрація, перегляд рецептів) формуються з прийнятною швидкістю завдяки простій логіці обробки та прямій взаємодії з базою даних без складних обчислень чи надмірного навантаження на сервер.

### 3.4 Вимоги бази даних

Для зберігання даних використовується документоорієнтована база даних MongoDB. Основні вимоги до неї:

* підтримка зберігання напівструктурованих даних у форматі BSON;
* можливість реалізації зв’язків «один до одного» та «один до багатьох»;
* швидкий пошук по фільтрам (індексація за ключовими полями);
* забезпечення унікальності за email користувача;
* підтримка масштабування та реплікації для забезпечення надійності в майбутньому.