

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНІКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Магістр»

«Веб-система для підтримки діяльності садівника»

(тема кваліфікаційної роботи українською мовою)

«Web-Based System to Support Gardener Activities»

(тема кваліфікаційної роботи англійською мовою)

Виконала: здобувачка заочної форми навчання
спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

Засоба Юлія Михайлівна

(прізвище, ім'я, по-батькові здобувача)

Керівник к.т.н., доцент Перелигін Б.В.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент к.т.н., доцент кафедри інженерії ПЗ

національного університету, Тройніна А.С.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

Інформаційних технологій

№ від 2024 р.

Завідувачка кафедри

(підпис)

(прізвище, ім'я)

Захищено на засіданні ЕК №

протокол № від 2024 р.

Оцінка / /

(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

(підпис)

(прізвище, ім'я)

Одеса 2024

АНОТАЦІЯ

Метою магістерської кваліфікаційної роботи є забезпечити інтегровану платформу, яка допоможе користувачам приймати обґрунтовані рішення щодо догляду за рослинами та покращить ефективність їхньої садівничої діяльності.

Методи розробки базуються на застосування моделі BERT для проектування чатів, на мові програмування JavaScript та фреймворку React, Node.js з фреймворком Express.js і базі даних PostgreSQL.

Результатом роботи є веб-система підтримки садівників, що включає інтерактивний каталог рослин, базу знань, чат-підтримку з використанням BERT, систему рекомендацій, нагадування та календар подій, забезпечуючи ефективний догляд за рослинами і підвищення обізнаності користувачів.

Ключові слова: модель BERT, каталог рослин, система рекомендацій, календар подій, JavaScript, React, Node.js, Express.js, PostgreSQL.

ABSTRACT

The goal of this work is to provide an integrated platform that will help users make informed decisions about plant care and improve the efficiency of their horticultural activities.

The development methods are based on the application of the BERT model for the design of chats, on the JavaScript programming language and the React framework, Node.js with the Express.js framework and the PostgreSQL database.

The result is a web-based support system for gardeners that includes an interactive plant catalog, a knowledge base, BERT-enabled chat support, a recommendation system, reminders, and an event calendar, providing effective plant care and user awareness.

Keywords: BERT model, plant directory, recommendation system, event calendar, JavaScript, React, Node.js, Express.js, PostgreSQL.

ЗМІСТ

АНОТАЦІЯ.....	3
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ	6
ВСТУП.....	7
1 СИСТЕМНИЙ АНАЛІЗ І СПЕЦИФІКАЦІЯ ВИМОГ	9
1.1 Актуальність розробки.....	9
1.2 Аналіз існуючих рішень.....	11
1.3 Компоненти веб-системи для інформаційної підтримки	16
1.4 Моделювання бізнес-процесів	23
1.4 Функціональні вимоги	26
2 ПРОЕКТУВАННЯ ВЕБ-СИСТЕМИ	29
2.1 Застосування моделі BERT для проектування чатів	29
2.2 Побудова моделі каталогу рослин.....	33
2.3 Алгоритм роботи BERT у чаті підтримки.....	35
2.4 Моделювання діаграми Use-Case	39
2.5 Архітектура веб-системи	49
3 РОЗРОБКА ВЕБ-СИСТЕМИ	52
3.1 Вибір технологій.....	52
3.2 Концептуальні класи веб-системи.....	56
3.3 Розробка бази даних	60
3.4 Тренування нейромережі	66
4 ТЕСТУВАННЯ РОБОТИ ВЕБ-СИСТЕМИ	69
4.1 Функціональне тестування	69
4.2 Тестування користувацького інтерфейсу (UI-тестування).....	74
ВИСНОВКИ	79
СПИСОК ВИКОРИСТОВАНИХ ДЖЕРЕЛ.....	80

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

Скорочення

BERT – Bidirectional Encoder Representations from Transformers

CRUD операції – Create, Read, Update, Delete

MLM – Masked Language Model

ORM – Object-Relational Mapping

Умовні позначення

Backend – Логічний шар

Frontend – Клієнтський шар

IDEF0 – методологія функціонального моделювання і графічного опису процесів

NLP – Обробка природної мови

One-way Data Flow – Одностороння передача даних

React – JavaScript бібліотека

UML Sequence Diagram – Діаграма послідовності дії

Test set – Тестова вибірка

Training set – Тренувальна вибірка

UI – Користувацький інтерфейс

Use Cases – випадків використання

Модель BERT– Bidirectional Encoder Representations from Transformers)

ВСТУП

Предметна область «Садівництво» охоплює всі аспекти догляду за рослинами, озеленення територій, вирощування квітів, дерев, кущів та інших рослин. Ця сфера є важливою складовою сільського господарства та ландшафтного дизайну, а також сприяє збереженню довкілля і підвищенню якості життя.

Роботи в садівництві тісно пов'язані із сезонними змінами та погодними умовами. Садівники повинні враховувати оптимальні строки для посадки, обрізки, поливу та збору врожаю, а також адаптуватися до змін клімату. Для цього слід складати графіки, що включає планування щоденних, тижневих і сезонних робіт, таких як догляд за рослинами, підготовка ґрунту, боротьба з шкідниками та збирання врожаю. Крім цього, садівникам необхідно управління запасами насіння, добрив, пестицидів, садового інструменту та техніки, що допомагає уникнути втрат і забезпечує своєчасне поповнення ресурсів.

Садівники потребують точних і сучасних даних про сорти рослин, умови їхнього вирощування, боротьбу з шкідниками та хворобами. Це знання включаються до рекомендацій щодо оптимального використання ресурсів і покращення врожайності. Використання літератури, баз даних, веб-ресурсів для отримання необхідної інформації про нові сорти рослин, сучасні методи вирощування та догляду.

Ручне управління всіма аспектами садівництва може бути трудомістким і неефективним, особливо на великих територіях або при обслуговуванні кількох об'єктів одночасно. Веб-система може значно полегшити управління всіма процесами, підвищуючи ефективність і якість роботи. Також, наявність бази знань у веб-системі, що буде містити інформацію про рослини, догляд за ними та умови вирощування, значно спростить та підвищить ефективність роботи садівників.

Метою магістерської кваліфікаційної роботи є забезпечити інтегровану платформу, яка допоможе користувачам приймати обґрунтовані рішення щодо догляду за рослинами та покращить ефективність їхньої садівничої діяльності.

Для досягнення мети необхідно:

1. Вивчити потреби садівників і визначити ключові функціональні компоненти системи, після чого розробити архітектуру веб-системи.
2. Реалізувати інтерактивний каталог рослин.
3. Створити базу знань: організувати інформацію про рослини, хвороби, шкідників і методи їх вирішення.
4. Розробити чат-підтримку: інтегрувати модель BERT для аналізу повідомлень і покращення взаємодії користувачів із системою, а також реалізувати нагадування та календар подій.
5. Забезпечити інтерактивний і зручний інтерфейс.
6. Протестувати систему.

Об'єктом дослідження є процес надання інформаційної та комунікаційної підтримки садівникам для догляду за рослинами, вирішення проблем зі шкідниками та хворобами, а також підвищення рівня знань користувачів.

Предметом дослідження є методи та інструменти розробки веб-системи, яка забезпечує інтерактивну взаємодію з користувачами через каталог рослин, базу знань, рекомендації, нагадування та календар подій.

1 СИСТЕМНИЙ АНАЛІЗ І СПЕЦИФІКАЦІЯ ВИМОГ

1.1 Актуальність розробки

Садівництво в сучасних умовах стикається з низкою проблем, пов'язаних із плануванням робіт, управлінням ресурсами та пошуком інформації про рослини. Ці виклики значно впливають на ефективність та успішність діяльності садівників.

1. Проблеми планування робіт:

- сезонність: більшість садівничих робіт залежать від пори року, що вимагає точного планування. Помилки в плануванні можуть призвести до втрати врожаю або погіршення якості рослин. Наприклад, невчасний посів або обрізка можуть негативно вплинути на розвиток рослин;
- кліматичні зміни: непередбачувані зміни в погодних умовах, спричинені глобальним потеплінням, ускладнюють довгострокове планування. Посухи, аномальні морози чи рясні дощі можуть знищити або пошкодити врожай, якщо роботи не адаптовані до нових умов;
- обмеженість робочої сили: сучасні садівники часто стикаються з дефіцитом кваліфікованих працівників, що впливає на здатність виконувати роботи вчасно та якісно. Це створює потребу в автоматизації та більш ефективному розподілі робочих завдань.

2. Проблеми управління ресурсами:

- управління запасами: необхідність управління великими обсягами інвентарю, включаючи насіння, добрива, інструменти, створює виклики. Невчасне поповнення запасів або неправильне зберігання можуть призвести до значних фінансових втрат;
- використання добрив та пестицидів: необхідно правильно розраховувати кількість добрив і пестицидів, щоб не нашкодити

рослинам і довкіллю. Недостатня або надмірна кількість хімікатів може знизити врожайність або навіть знищити рослини;

- енергозатратність: сучасні садівники змушені шукати способи зниження енергозатратності, пов'язаної з використанням техніки, поливу, обігріву теплиць. Це вимагає переходу на більш ефективні технології або використання відновлюваних джерел енергії.

3. Проблеми пошуку інформації про рослини:

- доступ до актуальної інформації: садівники постійно потребують точних і надійних даних про різні види рослин, включаючи інформацію про їх вирощування, шкідників, хвороби, оптимальні умови для росту. Інформація повинна бути актуальною, оскільки нові хвороби і шкідники можуть швидко поширюватися;
- складність інтеграції даних: багато джерел інформації є розрізненими, і садівникам важко інтегрувати ці дані в єдину систему для прийняття рішень. Це може спричинити помилки у виборі методів вирощування або догляду за рослинами;
- диджиталізація знань: не всі садівники мають доступ до цифрових технологій або достатню комп'ютерну грамотність для використання онлайн-ресурсів і спеціалізованого програмного забезпечення. Це обмежує їхні можливості отримувати та використовувати найсучаснішу інформацію [1].

Сучасне садівництво стикається з багатьма викликами, пов'язаними з плануванням робіт, управлінням ресурсами та пошуком інформації про рослини. Для подолання цих проблем необхідно використовувати новітні технології, такі як веб-системи для управління садівничою діяльністю, які можуть допомогти автоматизувати планування, оптимізувати управління ресурсами та забезпечити доступ до актуальної інформації про рослини. Ці рішення можуть значно підвищити ефективність і стійкість садівництва в умовах сучасних викликів.

1.2 Аналіз існуючих рішень

Garden Plan Pro (рис. 1.1) [2] це мобільний додаток, розроблений спеціально для планування садів та городів. Він призначений для садівників, які бажають ефективно планувати свої грядки, керувати рослинами та оптимізувати простір у саду.

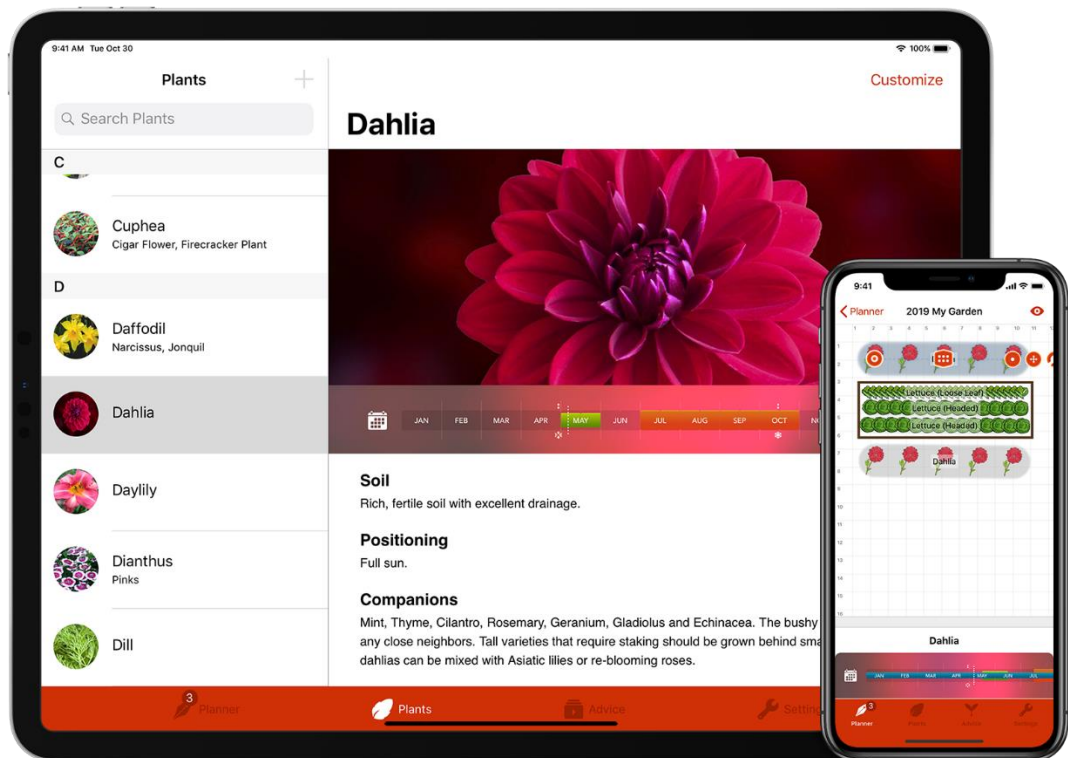


Рисунок 1.1 – Мобільний додаток Garden Plan Pro

Основні функції Garden Plan Pro:

1. Інструменти для планування:

- додаток дозволяє створювати детальні схеми саду або городу, використовуючи різні інструменти для розміщення рослин, доріжок, теплиць та інших елементів;
- користувачі можуть вибирати з великої бази даних рослин, з урахуванням специфіки їхнього росту, вимог до ґрунту та клімату.

2. Календар садівника:

- додаток автоматично генерує календар робіт на основі створеного плану саду, вказуючи на оптимальні строки посадки, поливу, підживлення і збору врожаю;
- календар враховує кліматичні умови регіону, щоб допомогти садівнику не пропустити важливі етапи догляду за рослинами.

3. Рекомендації та нагадування:

- garden plan pro надає персоналізовані рекомендації щодо догляду за рослинами, враховуючи особливості конкретних культур;
- користувачі можуть налаштовувати нагадування про важливі події, такі як посів, підживлення або збір урожаю.

4. Сумісність із кліматом та регіоном:

- додаток враховує кліматичні умови користувача, надаючи рекомендації, що відповідають місцевим особливостям;
- є можливість зберігати й аналізувати інформацію про погоду, що впливає на ріст рослин.

5. Синхронізація та резервне копіювання:

- додаток дозволяє синхронізувати дані між різними пристроями, що зручно для садівників, які працюють з декількома гаджетами;
- регулярне резервне копіювання даних забезпечує збереження всіх планів і історії робіт.

Garden Plan Pro ідеально підходить для як новачків, так і досвідчених садівників. Він допомагає організувати процес планування та догляду за садом, зберігаючи інформацію в одному місці і надаючи рекомендації, що підвищують продуктивність та ефективність роботи.

Цей додаток є платним, і доступний для користувачів на платформах iOS. Він отримав позитивні відгуки за свою простоту у використанні та багатофункціональність [3].

Наступне існуюче рішення для садівників – це додаток «Garden Compass Plant / Disease Identifier» (рис. 1.2) . Garden Compass Plant / Disease Identifier

допомагає користувачам ідентифікувати рослини, шкідників або хвороби. Просто сфотографувавши рослину чи хворобу, група експертів допоможе визначити вид та порекомендує рішення для вирішення проблеми.

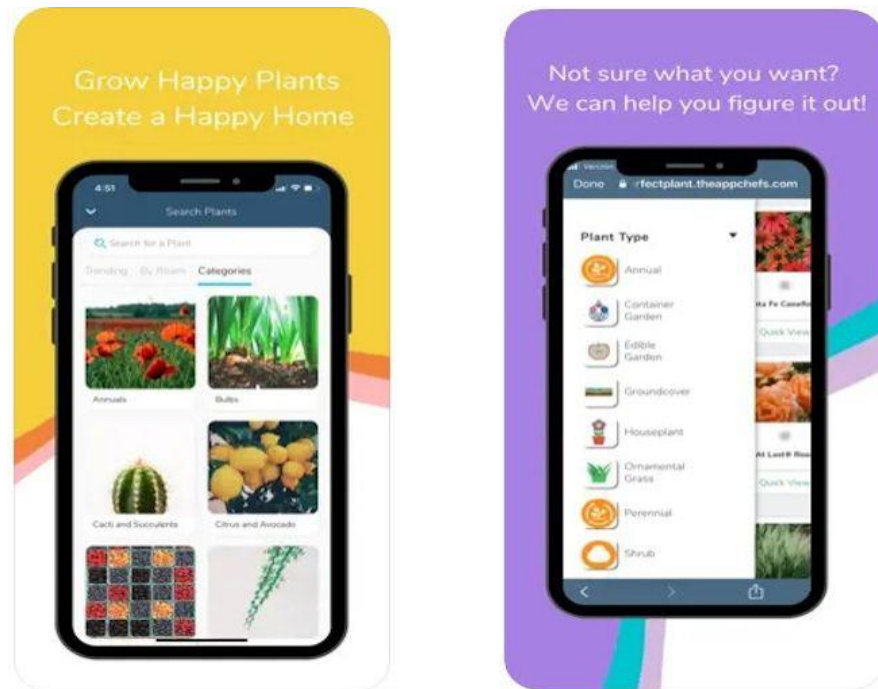


Рисунок 1.2 – Скрін вікна додатка «Garden Compass Plant/Disease Identifier»

Основні функції «Garden Compass Plant / Disease Identifier»:

1. Ідентифікація рослин: користувачі можуть зробити фото рослини, яку потрібно ідентифікувати, і завантажити його в додаток. Після цього команда експертів аналізує зображення і надає інформацію про рослину.

Це корисно для тих, хто стикається з новими або незнайомими видами рослин у своєму саду або під час прогулянки.

2. Діагностика хвороб і шкідників: додаток також дозволяє користувачам завантажити фото рослин, що мають ознаки хвороби або пошкоджені шкідниками. Експерти аналізують ці зображення і надають рекомендації щодо лікування або боротьби зі шкідниками.

Це особливо важливо для підтримки здоров'я рослин і запобігання поширенню хвороб у саду.

3. База даних і поради: додаток містить велику базу даних рослин, хвороб і шкідників, що дозволяє користувачам самостійно шукати інформацію про догляд за конкретними видами рослин.

Крім ідентифікації, Garden Compass надає рекомендації щодо поливу, підживлення, обрізки та інших аспектів догляду за рослинами.

4. Садові завдання: Garden Compass допомагає користувачам організувати їхній сад, створюючи список завдань з догляду за рослинами на основі отриманих рекомендацій.
5. Професійна підтримка: однією з ключових переваг додатку є те, що відповіді на запити надають реальні фахівці з ботаніки та садівництва, а не автоматизовані системи. Це підвищує точність і надійність інформації.

Цей додаток особливо корисний для початківців і досвідчених садівників, які потребують швидкої допомоги в ідентифікації рослин або діагностиці проблем. Він надає можливість отримати професійну консультацію прямо зі смартфона, що може допомогти запобігти втратам у саду та підтримувати рослини в найкращому стані [3].

Додаток доступний для платформ iOS і Android, і отримав позитивні відгуки за простоту у використанні та якісну підтримку з боку експертів.

«The Landscaper's Companion-Plant and Garden Reference Guide» (рис. 1.3) [4] це мобільний додаток, розроблений для садівників, ландшафтних дизайнерів і любителів рослин. Він є довідковим інструментом, який допомагає користувачам швидко знаходити інформацію про різні види рослин, їх характеристики, вимоги до догляду і багато іншого.

Основні функції The Landscaper's Companion:

1. Велика база даних рослин: додаток містить інформацію про понад 26 000 видів рослин, включаючи дерева, кущі, багаторічні та однорічні рослини, сукуленти, ліани та багато інших.

Кожна рослина супроводжується детальною інформацією, такою як зона вирощування, вимоги до ґрунту, потреби у світлі та воді, розмір, сезонність та інші ключові характеристики.



Рисунок 1.3 – Скрін екранів додатку «The Landscaper's Companion-Plant and Garden Reference Guide»

2. Пошук та фільтрація: користувачі можуть швидко знайти потрібну рослину за допомогою функцій пошуку та фільтрації, наприклад, за типом рослини, кольором квітки, розміром або вимогами до догляду. Це зручно для ландшафтних дизайнерів, які можуть підбирати рослини відповідно до конкретних умов або бажаного дизайну.
3. Закладки та нотатки: додаток дозволяє користувачам зберігати рослини у вигляді закладок, а також додавати власні нотатки. Це корисно для планування саду або ландшафтного дизайну, а також для збереження власних спостережень.
4. Фотографії та ілюстрації: кожна рослина супроводжується фотографіями або ілюстраціями, що допомагає візуально ідентифікувати рослину і краще зрозуміти її зовнішній вигляд.

5. Регулярні оновлення: база даних додатку регулярно оновлюється новою інформацією про рослини, що забезпечує актуальність даних для користувачів.

Цей додаток ідеально підходить для професійних ландшафтних дизайнерів, садівників і ентузіастів, які потребують надійного джерела інформації про рослини. Він також може бути корисним для студентів-біологів або агрономів, які вивчають ботаніку та ландшафтний дизайн.

The Landscaper's Companion доступний на платформах iOS і Android, і має позитивні відгуки за свою велику базу даних, зручність у використанні та можливість швидкого доступу до важливої інформації про рослини.

1.3 Компоненти веб-системи для інформаційної підтримки

Для інформаційної підтримки садівників слід включити до складу веб-системи декілька компонент:

- база знань;
- чат підтримки;
- система рекомендацій;
- нагадування та календар подій.

База знань у веб-системі для підтримки діяльності садівників буде зберігати різноманітну та структуровану інформацію, яка допоможе користувачам отримати відповіді на запитання про рослини, догляд за ними, а також вирішення проблем, пов'язаних з хворобами або шкідниками. Основні компоненти Бази знань включатимуть:

1. Інформація про рослини:

- назви та класифікація: наукові та народні назви рослин, їх класифікація (наприклад, дерева, кущі, багаторічники, однорічники, сукуленти тощо);
- опис і характеристики: короткий опис кожної рослини, включаючи її зовнішній вигляд, розмір, форму, колір квітів та листя;

- умови вирощування: рекомендації щодо оптимальних умов для вирощування, такі як світло, вологість, тип ґрунту, температура;
- зони вирощування: інформація про кліматичні зони, де найкраще ростуть певні рослини;
- період цвітіння: інформація про час цвітіння та врожайності рослин, коли це доречно.

2. Догляд за рослинами:

- полив: рекомендації щодо частоти та об'єму поливу для різних рослин;
- підживлення: типи добрив, частота їх використання, особливості підживлення у різні сезони;
- обрізка та формування: інструкції щодо обрізки, формування кущів та дерев, включаючи оптимальні часи для цього;
- пересадка: керівництво з пересадки, вибір оптимального горщика або місця у саду;
- зимівля та захист: рекомендації щодо захисту рослин від морозів, підготовка до зими.

3. Хвороби та шкідники:

- ідентифікація хвороб: опис типових хвороб рослин, симптоми, які допоможуть їх розпізнати;
- ідентифікація шкідників: інформація про шкідників, які можуть пошкодити рослини, способи їх розпізнання;
- методи лікування: рекомендації щодо лікування рослин від хвороб і шкідників, включаючи використання пестицидів, біологічних засобів та профілактичні заходи;
- профілактика: інформація про профілактичні заходи для запобігання хворобам і шкідникам.

4. Сезонні роботи:

- календар садівника: збір інформації про сезонні роботи в саду, такі як посадка, обрізка, збирання врожаю, підготовка до зими;

- сезонні рекомендації: рекомендації щодо догляду за рослинами залежно від сезону (весна, літо, осінь, зима).

5. Додаткові матеріали:

- фото та ілюстрації: візуальні матеріали для ідентифікації рослин, шкідників, а також для демонстрації правильних методів догляду;
- відеоінструкції: навчальні відео про догляд за рослинами, боротьбу з хворобами і шкідниками, правильну техніку обрізки, пересадки тощо;
- посилання на додаткові ресурси: перелік корисних посилань на зовнішні ресурси або наукові публікації для більш детального вивчення.

Чат підтримки у веб-системі для садівників є важливим інструментом, який допомагає користувачам отримувати відповіді на свої запитання, спілкуватися з іншими садівниками та фахівцями, а також швидко знаходити потрібну інформацію. Чат буде потужним інструментом для взаємодії користувачів, обміну знаннями та отримання оперативної допомоги. Він дозволить користувачам швидко вирішувати проблеми, отримувати поради та ділитися досвідом, що значно покращить ефективність їхньої роботи та успішність садівництва в цілому.

Основні функції чату підтримки включають:

1. Задання питань та отримання відповідей:

- реальний час: користувачі можуть задавати запитання у чаті та отримувати відповіді від інших користувачів або модераторів у режимі реального часу;
- експертні відповіді: платформа може залучати експертів у садівництві, які надаватимуть професійні поради щодо вирощування рослин, боротьби з хворобами або планування робіт;
- відгуки від інших садівників: користувачі можуть обмінюватися досвідом, давати поради та допомагати один одному в межах спільноти.

2. Пошук у чаті та базі знань:

- інтеграція з базою знань: чат підтримки може бути інтегрований з базою знань, що дозволить користувачам швидко знаходити відповіді на типові запитання або отримувати посилання на відповідні статті;
- історія чату: можливість пошуку в історії чату дозволяє користувачам знаходити раніше обговорені питання і рішення, зберігаючи час на повторне задавання схожих питань.

3. Сповіщення та нагадування:

- нагадування: користувачі можуть отримувати сповіщення про нові відповіді на свої запитання, а також про важливі події в чаті або оновлення бази знань;
- персоналізовані рекомендації: чат може автоматично надавати рекомендації на основі питань користувача або історії його запитів.

4. Модерація:

- забезпечення порядку: модератори можуть контролювати зміст чату, видаляти спам, блокувати небажаних користувачів та забезпечувати дотримання правил спільноти;
- фільтри контенту: система може автоматично фільтрувати повідомлення, що містять неналежний контент, забезпечуючи безпечне та продуктивне середовище для обговорення.

5. Додаткові функції:

- формування груп: користувачі можуть об'єднуватися в групи за інтересами (наприклад, вирощування певних видів рослин, органічне садівництво) для обговорення конкретних тем;
- формування запитів до експертів: можливість створення запитів до експертів у разі потреби глибшого аналізу або консультації з фахівцем.

Система рекомендацій у веб-системі для садівників є інструментом, який допомагає користувачам ухвалювати обґрунтовані рішення щодо догляду за рослинами, планування робіт та вирішення проблем. Така система використовує дані про рослини, умови вирощування, досвід інших користувачів і наукові знання, щоб надавати персоналізовані поради.

Основні аспекти системи рекомендацій:

1. Персоналізовані рекомендації:

- врахування індивідуальних умов: система може надавати рекомендації з урахуванням кліматичної зони, типу ґрунту, рівня освітленості та інших специфічних умов вирощування. наприклад, поради щодо поливу або підживлення можуть бути адаптовані відповідно до місця проживання садівника;
- історія роботи користувача: рекомендації можуть базуватися на попередньому досвіді користувача, його виборі рослин, попередніх запитах та виконаних роботах;

2. Сезонні поради:

- автоматичні нагадування: на основі календаря та типу рослин, система може автоматично нагадувати про необхідні дії (наприклад, посадка, обрізка, підживлення) у відповідні періоди року;
- сезонні роботи: система може пропонувати користувачам рекомендації щодо підготовки рослин до зміни сезонів (наприклад, захист від морозів, підготовка до літа).

3. Інформаційні поради з догляду:

- вибір рослин: рекомендації щодо підбору рослин, які підходять для конкретних умов, ґрунту, освітлення, та естетичних потреб користувача;
- профілактика та лікування хвороб: система може запропонувати методи профілактики хвороб або лікування, якщо користувач вказує симптоми, які спостерігаються у рослин.

4. Рекомендації на основі даних інших користувачів:

- соціальні рекомендації: аналізуючи досвід інших садівників у системі, можна надавати рекомендації на основі схожих ситуацій або рослин, які показали найкращі результати;
- тренди та популярні рослини: рекомендації щодо популярних або трендових рослин у певному регіоні або серед схожих користувачів.

5. Інтеграція з іншими компонентами системи:

- інтеграція з базою знань: рекомендації можуть бути доповнені посиланнями на відповідні статті, інструкції або відео з бази знань;
- інтеграція з чатом підтримки: якщо система не може надати однозначну рекомендацію, користувач може швидко задати питання в чаті підтримки або отримати доступ до відповідних тем, що обговорювались іншими садівниками.

Наприклад, користувач вводить інформацію про свої рослини, умови вирощування або конкретні проблеми, і система рекомендацій пропонує оптимальні дії, пояснюючи, чому саме ці заходи є найбільш доцільними. Якщо система виявить, що користувач посадив рослини, які не сумісні за умовами вирощування, вона може запропонувати зміни в розміщенні рослин або додатковий догляд.

Нагадування та календар подій будуть допомагати організувати процес догляду за рослинами та ефективно планувати роботи у саду чи на городі. Ці функції забезпечують своєчасне виконання всіх необхідних завдань, включаючи полив, обрізку, внесення добрив, обробку рослин від шкідників та інші заходи.

Основні можливості:

1. Персоналізовані нагадування: садівники можуть створювати індивідуальні нагадування для різних дій. Це включає регулярні завдання, такі як щоденний полив або тижневе внесення добрив, а

також одноразові нагадування, наприклад, про обробку рослин перед настанням холодів або посадку нових рослин на початку сезону.

Налаштування нагадувань дозволяє забезпечити, щоб жодне завдання не було пропущене, що особливо важливо для збереження здоров'я рослин і підтримки їхнього росту.

2. Календар подій: система надає можливість відображати всі заплановані події та завдання в зручному календарному форматі. Це дозволяє садівникам легко переглядати свій розклад і вносити зміни в плани догляду за рослинами відповідно до своїх потреб.

У календарі можна відстежувати як поточні, так і майбутні завдання, що дає змогу підготуватися до важливих подій, таких як збирання врожаю або пересадка.

3. Адаптація до погодних умов: система може враховувати прогнозовані зміни погоди та попереджати садівників про необхідність додаткового догляду або захисту рослин. Наприклад, якщо очікується різке похолодання, система нагадає про потребу вкривати рослини або обробити їх перед морозами.

Така функціональність дозволяє більш гнучко реагувати на погодні умови і захистити рослини від несприятливих факторів.

4. Оптимізація збору врожаю: нагадування та календар подій допомагають визначити оптимальний час для збору врожаю, враховуючи поточні умови та стан рослин. Це дозволяє садівникам вчасно підготуватися до збирання та забезпечити максимальну врожайність.

Система нагадувань також може пропонувати час для збору врожаю на основі історичних даних та прогнозів, що дозволяє мінімізувати ризики втрат.

1.4 Моделювання бізнес-процесів

IDEF0 діаграма допомагає відобразити структуру системи, описати її основні функції, а також взаємозв'язки між різними компонентами системи. Для веб-системи підтримки діяльності садівників основний процес може бути позначений як «Підтримка діяльності садівників», який ділиться на декілька підпроцесів (рис. 1.4).

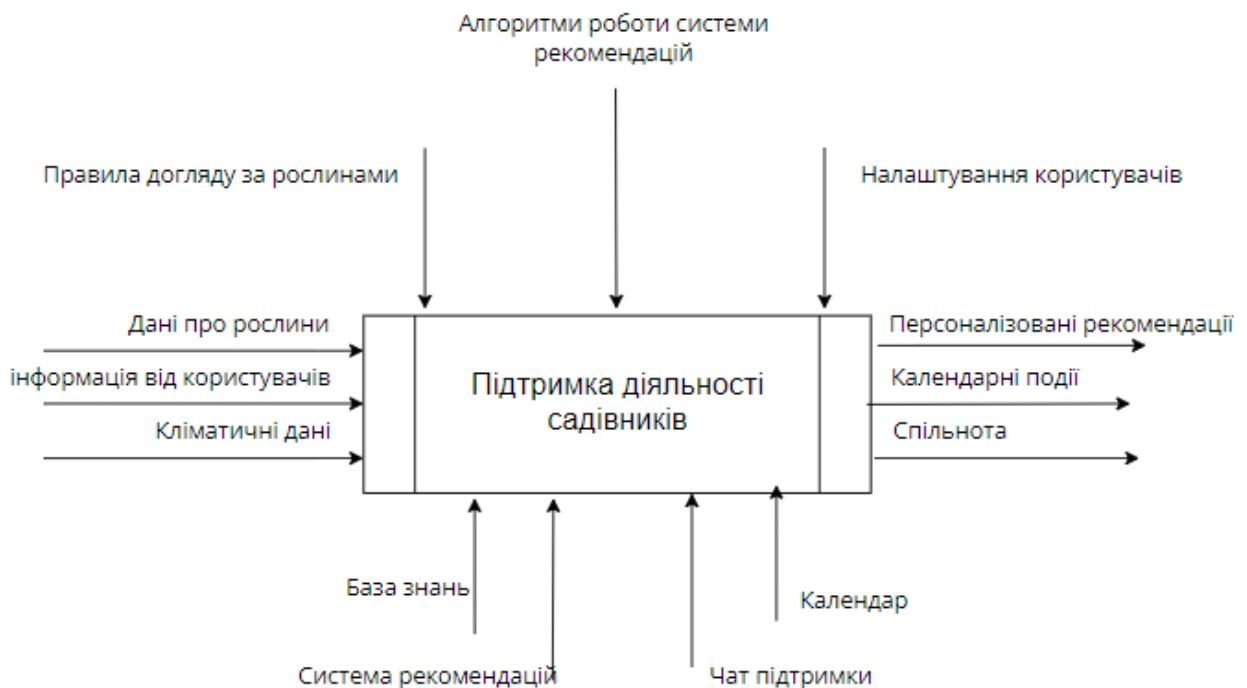


Рисунок 1.4 – Опис основного процесу «Підтримка діяльності садівників»

До опису процесу «Підтримка діяльності садівників» входить:

- Входи: дані про рослини, інформація від користувачів, кліматичні дані, запити користувачів на створення нагадувань або календарних подій;
- Виходи: персоналізовані рекомендації, нагадування, календарні події, аналітичні звіти, підтримка в режимі реального часу;
- Механізми (механізми виконання): база знань про рослини, система рекомендацій, модуль нагадувань, календарний модуль, чат підтримки;

- Контроль: бравила догляду за рослинами, рекомендації експертів, налаштування користувачів, алгоритми роботи системи рекомендацій.

Наступний крок – це декомпозиція основної функції на підфункції, а саме на управління базою знань, генерацію та управління рекомендаціями, управління нагадуваннями та календарними подіями та підтримка користувачів (рис. 1.5-1.6). Опишемо кожен складову функцію окремо.

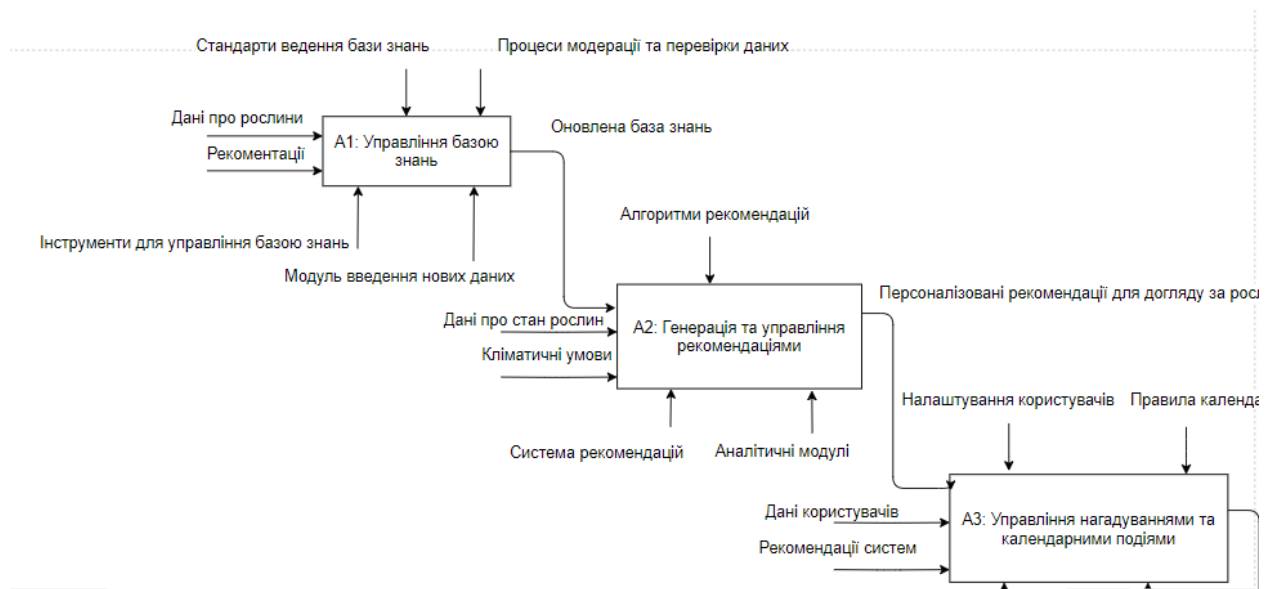


Рисунок 1.5 – Декомпозиція першого рівня (A1-A2)

A1 «Управління базою знань»:

- Входи: дані про рослини, рекомендації експертів;
- Виходи: оновлена база знань, доступ до інформації про догляд за рослинами;
- Механізми: інструменти для управління базою знань, модуль введення нових даних;
- Контроль: стандарти ведення бази знань, процеси модерзації та перевірки даних.

A2 «Генерація та управління рекомендаціями»:

- Входи: дані про стан рослин, кліматичні умови, історія роботи користувача.
- Виходи: персоналізовані рекомендації для догляду за рослинами.
- Механізми: система рекомендацій, аналітичні модулі.
- Контроль: алгоритми рекомендацій, бази даних історичних даних.



Рисунок 1.6 – Декомпозиція першого рівня (A3-A4)

A3 «Управління нагадуваннями та календарними подіями»:

- Входи: дані користувачів, рекомендації системи;
- Виходи: створені нагадування, календарні події;
- Механізми: модуль календаря, модуль нагадувань, інтеграція з погодними сервісами;
- Контроль: налаштування користувачів, правила календарних подій.

A4 «Підтримка користувачів (чат та довідка)»:

- Входи: запити користувачів, дані про рослини;

- Виходи: відповіді на запити, поради, посилання на релевантні статті з бази знань;
- Механізми: чат підтримки, модуль пошуку в базі знань;
- Контроль: модерація чату, правила надання консультацій, доступ до експертів.

Ця структура IDEF0 діаграми відображає, як різні компоненти системи працюють разом для забезпечення комплексної підтримки садівників у догляді за рослинами, плануванні робіт та отриманні своєчасних рекомендацій.

1.4 Функціональні вимоги

Ці функціональні вимоги визначають ключові можливості веб-системи для підтримки діяльності садівників, забезпечуючи ефективне управління інформацією, плануванням робіт, комунікацією між користувачами та наданням персоналізованих рекомендацій для успішного догляду за рослинами.

1. Управління базою знань.

Функція зберігання та доступу до інформації про рослини: система повинна забезпечувати зберігання інформації про рослини, включаючи наукові та народні назви, умови вирощування, методи догляду, періоди цвітіння, хвороби та шкідників. Користувачі повинні мати можливість шукати та переглядати інформацію за різними критеріями (назва, тип рослини, умови догляду тощо).

Функція додавання та оновлення даних: адміністратори повинні мати можливість додавати нові записи до бази знань, оновлювати існуючі дані та видаляти застарілу інформацію. Система повинна підтримувати модерацію доданих даних, щоб забезпечити їх точність і актуальність.

2. Генерація та управління рекомендаціями.

Функція надання персоналізованих рекомендацій: система повинна надавати користувачам рекомендації щодо догляду за рослинами на основі введених даних, таких як тип рослини, кліматичні умови, історія попередніх робіт. Рекомендації повинні враховувати сезонність та кліматичні умови, що впливають на догляд за рослинами.

Функція оновлення рекомендацій у реальному часі: система повинна автоматично оновлювати рекомендації на основі змінних факторів, таких як погода або оновлені дані про стан рослин.

3. Управління нагадуваннями та календарними подіями.

Функція створення та редагування нагадувань: користувачі повинні мати можливість створювати персоналізовані нагадування для регулярних (наприклад, щоденний полив) та одноразових (наприклад, обробка рослин перед морозами) завдань. Система повинна підтримувати редагування і видалення нагадувань.

Функція відображення календаря: система повинна відображати календар з усіма запланованими подіями, дозволяючи користувачам переглядати та керувати своїм розкладом догляду за рослинами. Календар повинен відображати події з урахуванням рекомендацій системи та запланованих користувачем дій.

Функція інтеграції з зовнішніми календарями: система повинна підтримувати синхронізацію з популярними календарними сервісами (Google Calendar, Apple Calendar тощо), щоб користувачі могли бачити всі свої події в одному місці.

4. Чат підтримки та пошук інформації.

Функція задати питання у чаті: користувачі повинні мати можливість задавати питання у чаті, отримувати відповіді від інших користувачів або від адміністратора/експерта. Система повинна підтримувати можливість групових обговорень і обміну досвідом між користувачами.

Функція пошуку в чаті та базі знань: система повинна надавати можливість пошуку відповідей на запитання в історії чату та базі знань, щоб

користувачі могли знайти вже обговорені теми. Результати пошуку повинні включати відповідні статті з бази знань, пов'язані питання з чату та рекомендації системи.

5. Підтримка користувачів.

Функція модерації чату: адміністратори повинні мати можливість модерації чату, видалення небажаних повідомлень і забезпечення дотримання правил спільноти.

Функція управління користувачами: система повинна дозволяти адміністраторам управляти обліковими записами користувачів, включаючи їх блокування, надання додаткових прав доступу або видалення.

Проведено аналіз предметної області. Розглянуто ключові компоненти інформаційної підтримки, включаючи базу знань, чат для обговорень, механізми надання рекомендацій, систему нагадувань та календар подій. Також детально проаналізовано проблеми спілкування в чатах і принципи функціонування моделі BERT, яка використовує алгоритми штучного інтелекту для оцінки коректності повідомлень.

Було здійснено огляд існуючих програмних аналогів. Для кожного аналізованого рішення виконано оцінку можливостей інформаційної підтримки, наявності чатів для взаємодії користувачів, можливості створення подій, а також рівень зручності користувацького інтерфейсу.

2 ПРОЕКТУВАННЯ ВЕБ-СИСТЕМИ

2.1 Застосування моделі BERT для проектування чатів

Модель BERT (Bidirectional Encoder Representations from Transformers) – це сучасна нейронна мережа, розроблена для роботи із задачами обробки природної мови (NLP). Вона забезпечує ефективне розуміння та генерацію тексту, зробивши значний внесок у розвиток глибокого навчання в цій сфері. BERT продемонструвала високі результати у таких завданнях, як машинний переклад, класифікація текстів і витяг інформації. Основні особливості моделі BERT:

1. Bidirectional Context.

Bidirectional Context (двосторонній контекст) — це ключова концепція, яка лежить в основі роботи моделі BERT (Bidirectional Encoder Representations from Transformers). Вона дозволяє моделі враховувати контекст слова з обох боків у реченні, що робить її більш точною при розумінні тексту.

Традиційний підхід (односторонній контекст): у більшості моделей попереднього покоління (наприклад, LSTM або GloVe) контекст слова визначався лише на основі попередніх слів (лівий контекст) або, рідше, наступних слів (правий контекст). Такий підхід не дозволяв повністю зрозуміти значення слова в реченні.

Двосторонній контекст у BERT: у моделі BERT кожне слово аналізується з урахуванням усіх слів, що його оточують, а саме, лівий контекст включає всі слова, які передують поточному, а правий контекст всі слова, які йдуть після поточного. Наприклад, у реченні: "Ключ знаходиться у замку, а не у дверях," модель розуміє, що слово "ключ" має значення "інструмент", а не "мелодія", завдяки аналізу всього речення. [5].

У Transformer-архітектурі BERT використовує механізм уваги, який дозволяє кожному слову "спілкуватися" з іншими словами у реченні, незалежно від їхньої позиції. Завдяки цьому кожне слово отримує контекст як зліва, так і справа. Для навчання BERT маскує випадкові слова в тексті та

навчається передбачати їх, аналізуючи всі слова з обох боків. Це змушує модель враховувати двосторонній контекст для кожного слова.

Цей підхід дозволяє моделі «зрозуміти» слова в більш широкому контексті, враховуючи всі можливі зв'язки в реченні. Такий двонапрямковий (bidirectional) підхід робить BERT особливо ефективним для завдань NLP, так як він може краще усвідомлювати семантичні залежності та значення слів у тексті.

2. Transformer Architecture.

Transformer Architecture – це революційна нейронна мережа, представлена в роботі "Attention Is All You Need" (2017), яка стала основою багатьох сучасних моделей для обробки природної мови (NLP), таких як BERT, GPT, T5 та інших. Її ключова особливість – ефективне використання механізму уваги (attention), що дозволяє обробляти довгі залежності в тексті. Механізм уваги.

Transformer використовує увагу для виділення важливих елементів у вхідних даних. Це дозволяє моделі акцентувати увагу на суттєвих частинах послідовності, зменшуючи вплив другорядних деталей. Замість обмеження фіксованим контекстом, Transformer аналізує всю вхідну послідовність, що забезпечує гнучкість і ефективність у вирішенні задач розуміння контексту. Архітектура передбачає кілька голів уваги, кожна з яких аналізує різні аспекти вхідних даних. Це дозволяє паралельно враховувати різні контексти, підвищуючи точність і швидкість обробки.

Transformer ідеально підходить для задач, де потрібно обробляти вхідні та вихідні послідовності, як-от машинний переклад. Завдяки здатності працювати з послідовностями будь-якої довжини, модель демонструє високу універсальність для завдань NLP. [6].

3. Pre-training.

Переднавчання є важливим етапом у роботі моделі BERT, завдяки якому вона отримує загальні знання про мову та її структуру. Це дозволяє моделі ефективно виконувати широкий спектр завдань NLP.

Основні компоненти переднавчання:

1. Маскована модель мови (Masked Language Model, MLM).

У процесі переднавчання частина слів у вхідному тексті замінюється спеціальним токеном [MASK]. Завдання моделі – передбачити замасковані слова, використовуючи контекст з обох боків (ліворуч і праворуч). Цей підхід допомагає моделі навчитися розуміти значення слів у їхньому контексті.

2. Передбачення наступного речення (Next Sentence Prediction, NSP).

Моделі подаються пари речень, і вона визначає, чи є друге речення логічним продовженням першого. Завдяки цьому BERT краще виявляє взаємозв'язки між реченнями, що важливо для завдань, які потребують аналізу тексту на рівні документа.

3. Розуміння зв'язків у тексті.

Завдяки двосторонньому контексту (bidirectional context) BERT аналізує текст на глибокому рівні, вивчаючи як семантичні, так і синтаксичні залежності між словами. Це робить модель ефективною у завданнях з розуміння змісту тексту, таких як витяг інформації, аналіз настроїв і класифікація текстів. Переднавчання дає моделі потужну базу для подальшого донавчання на конкретних завданнях, що робить BERT універсальним інструментом для обробки природної мови.

Переднавчання дає моделі потужну базу для подальшого донавчання на конкретних завданнях, що робить BERT універсальним інструментом для обробки природної мови. [5].

4. Fine-tuning.

Fine-tuning є важливим етапом застосування моделі BERT, спрямованим на її адаптацію до специфічних завдань у сфері обробки природної мови (NLP). Цей процес дозволяє моделі максимально ефективно виконувати поставлені завдання, враховуючи особливості конкретних даних.

Оскільки BERT уже переднавчена на великих корпусах даних, для Fine-tuning зазвичай потрібен значно менший обсяг специфічних даних, що знижує ризик перенавчання і спрощує процес адаптації. Під час цього етапу модель

оптимізує свої параметри, враховуючи особливості даних завдання, і це дозволяє їй забезпечувати точність і релевантність результатів.

У результаті Fine-tuning дозволяє моделі BERT стати універсальним інструментом для вирішення широкого спектра завдань NLP, забезпечуючи адаптацію до конкретного контексту, точність і ефективність роботи.

5. Мовна універсальність BERT – це ключова особливість, яка визначає його ефективність у багатьох міжнародних завданнях обробки природної мови (NLP).

BERT вирізняється своєю здатністю ефективно обробляти текстові дані на різних мовах одночасно. Ключовою перевагою є те, що під час переднавчання модель вчиться розпізнавати мовні особливості, синтаксичні структури та семантичні зв'язки в різних мовах, що робить її надзвичайно універсальною для завдань, що включають тексти з різних культур та лінгвістичних груп.

Отже, використання функції BERT для чату:

1. Розуміння запитів:

- класифікація намірів, а саме, модель може визначати, що саме цікавить користувача (наприклад, питання про полив, добриво, хвороби рослин);
- контекстуальне розуміння, тобто BERT враховує контекст, тому зможе розпізнавати схожі запитання навіть при різних формулюваннях.

2. Генерація відповідей:

- автоматичні відповіді на запитання на основі контексту. Якщо у базі знань є відповідь на запит, модель може відразу її запропонувати;
- додаткова обробка, а саме є можливість комбінувати BERT з іншими методами (наприклад, пошук за ключовими словами) для отримання більш точних і релевантних відповідей.

3. Перевірка вже заданих запитів:

- історія запитів допомагає при надходженні нового запиту, а саме система може шукати в цій історії, використовуючи BERT для порівняння;
- відкриття існуючих відповідей, тобто якщо система виявляє, що питання вже було задано, вона може відобразити відповідь з історії. Це дозволить користувачам отримувати інформацію швидше і зменшить навантаження на систему.

2.2 Побудова моделі каталогу рослин

Для побудови моделі каталогу рослин за вказаною структурою можна визначити основні елементи, які будуть включені до кожної категорії. Ось детальніше пояснення кожного з елементів моделі:

$Cat = Cat_i, i=1..N$: це загальна структура каталогу, де N – кількість категорій рослин.

$Cat_i = \langle Name, Chat, Recommendations, Event \rangle$: тут кожна категорія містить такі елементи:

- **Name**: назва категорії (наприклад, «Декоративні», «Овочеві»);
- **Chat**: посилання на чат, де користувачі можуть спілкуватися щодо конкретної категорії. Це може бути або окремий канал, або секція в загальному чаті;
- **Recommendations**: рекомендації для користувачів, пов'язані з даною категорією рослин (наприклад, поради щодо догляду, підбору рослин);
- **Event**: події, пов'язані з категорією (наприклад, дати сезонних робіт, фестивалі рослин).

Приклад реалізації для категорій декоративних та овочевих рослин буде мати наступний вигляд:

```

C1 = <"Декоративні", "link_to_chat_decorative",
      "Рекомендуємо поливати раз на тиждень. Використовуйте
      добрива кожні два тижні.", "Обрізка рослин: березень,
      квітень">
C2 = <"Овочеві", "link_to_chat_vegetables",
      "Збирайте овочі вранці, коли рослини ще вологі.",
      "Посадка насіння: квітень, травень">

```

Побудова дерева каталогу рослин – це важливий етап, який допоможе організувати інформацію так, щоб користувачі могли легко знаходити потрібні дані. По-перше, необхідно побудувати основу для дерева каталогу, яку з часом можна буде доповнювати новими категоріями та даними. До основних категорій будемо відносити наступні:

1. Типи рослин:
 - декоративні;
 - овочеві;
 - фруктові;
 - чагарники;
 - дерева.
2. Середовище вирощування:
 - внутрішні рослини;
 - вуличні рослини;
 - квіти.
3. Кліматичні умови:
 - теплолюбні;
 - холодостійкі;
 - посухостійкі.

Наступний крок, це визначення підкатегорій для більш детального класифікування, а саме:

1. Декоративні (квітучі, листяні, кактуси та сукуленти).
2. Овочеві (коренеплоди, листові, фрукти).

Створення логічного і зручного дерева каталогу допоможе користувачам швидше знайти потрібну інформацію і підвищить загальну ефективність веб-системи підтримки садівників.

Рослини



Рисунок 2.1 – Структура каталогу рослин

2.3 Алгоритм роботи BERT у чаті підтримки

Модель BERT ефективно обробляє та розуміє природну мову, що дозволяє її використовувати для автоматизації відповідей на запитання користувачів, класифікації повідомлень і знаходження найбільш релевантної

інформації. Алгоритм роботи BERT для чату підтримки можна розбити на кілька кроків.

1. Попередня обробка даних:

- a) токенизація, тобто кожне повідомлення користувача розбивається на окремі слова або токени за допомогою спеціального токенизатора, який підтримує BERT (наприклад, WordPiece Tokenizer);
- b) додавання спеціальних tokenів: до початку та кінця тексту додаються спеціальні токени ([CLS] для початку і [SEP] для кінця), які допомагають моделі визначити початок та завершення повідомлення;
- c) перетворення вектори: токени перетворюються на індекси (вектори) відповідно до словника BERT;

2. Введення даних у BERT:

- a) подача в модель: перетворені токени передаються в BERT як вхідний шар. Модель обробляє текст двонаправлено, аналізуючи контекст кожного слова як зліва направо, так і справа наліво;
- b) контекстуальне кодування: BERT створює контекстуально залежні представлення кожного токена. Це означає, що значення кожного слова в реченні визначається з урахуванням інших слів навколо нього.

3. Обробка BERT:

- a) шар трансформера: BERT використовує багат шаровий трансформер, що дозволяє моделі зрозуміти, як слова у реченні взаємопов'язані на різних рівнях (глибокі залежності між словами, граматичні конструкції, сенс фрази);
- b) вихідні представлення: після проходження через модель, на виході кожен токен отримує свій векторний представник, який враховує як контекст цього слова в реченні, так і загальний сенс речення.

4. Класифікація або відповідь:

- а) класифікація або предикція: якщо мета – класифікація запитань (наприклад, вибір категорії питання чи визначення тону повідомлення), вектор для всього речення використовується для того, щоб передати його в класифікатор (звичайно це шар нейронної мережі) для визначення категорії запиту (наприклад, «полив», «добрива», «хвороби рослин»);
- б) пошук відповіді: якщо BERT використовується для пошуку відповіді на питання, модель може порівнювати питання користувача з базою знань або попередніми діалогами. BERT може знаходити найрелевантніші відповіді, порівнюючи контекст запиту і можливі відповіді через семантичну схожість їх представлень.

5. Післяобробка:

- а) відновлення тексту: вихід моделі у вигляді векторів токенів перетворюється назад у зрозумілий текст (у випадку, якщо необхідна автоматична відповідь на питання);
- б) виведення відповіді: якщо BERT виконував класифікацію, результат класифікації може бути переданий для виконання наступних дій, наприклад, показати користувачу статтю з бази знань, відповідну категорії питання. Якщо ж це пошук відповіді, користувачу буде надана найбільш релевантна відповідь.

6. Оцінка релевантності (опціонально): якщо система підтримує взаємодію з користувачем для покращення якості, після виведення відповіді користувач може оцінити релевантність відповіді, і ця інформація може бути використана для навчання моделі або подальшої оптимізації результатів.

Таким чином, BERT у чаті підтримки може відповідати на запити користувачів на основі глибокого аналізу природної мови, забезпечуючи розуміння та відповідь у контексті запитів.

Діаграма послідовності дії (UML Sequence Diagram) відображає взаємодію між акторами (користувачем, чатботом, BERT, базою даних) у

хронологічному порядку. Основна мета діаграми послідовності – показати, як відбувається обмін повідомленнями між об'єктами у системі з часом. Вона демонструє покроковий процес обробки запиту користувача, починаючи від введення тексту до отримання відповіді через чатбот, обробку за допомогою BERT та бази знань.

Діаграма послідовності дії (рис. 2.3) включатиме таких акторів, як:

1. User (Користувач).
2. Chatbot (Чатбот на основі BERT).
3. BERT (Модель для обробки природної мови).
4. Database (База знань).

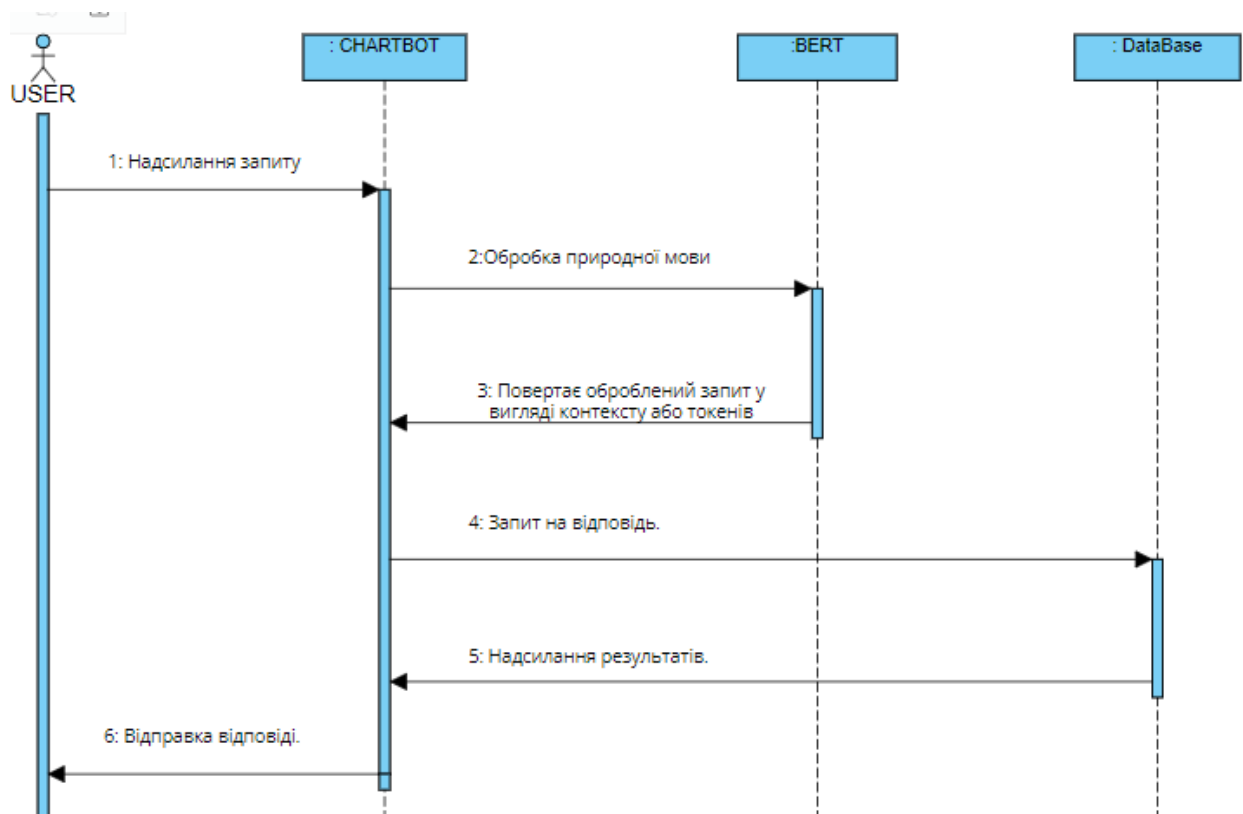


Рисунок 2.3 – Діаграма послідовності дії для обробки запиту користувача у чаті підтримки

Опис кроків діаграми:

Крок №1 «Користувач -> Чатбот»: користувач надсилає запит у вигляді текстового повідомлення через інтерфейс чату, наприклад: "Як часто поливати троянди?". Мета: ініціація взаємодії з чатботом.

Крок №2 «Чатбот -> BERT»: чатбот відправляє текстове повідомлення до NLP-моделі BERT, яка виконує аналіз тексту. Мета: визначити сенс, намір та витягти основні інформаційні блоки (токенізація, семантичний аналіз) для кращого розуміння запиту.

Крок №3 «BERT -> Чатбот»: BERT аналізує текст запиту, виділяє ключові слова, фрази, визначає сенс запиту та контекст. Мета: зробити запит користувача зрозумілим для системи.

Крок №4 «Чатбот -> База знань»: чатбот на основі даних, отриманих від BERT, формує пошуковий запит до бази знань, щоб знайти відповідну інформацію. Мета: отримати точну інформацію з бази знань, що відповідає запиту користувача.

Крок №5 «База знань -> Чатбот»: база знань шукає дані, пов'язані із запитом (наприклад, знайти інформацію про полив троянд) і повертає цю інформацію чатботу. Мета: надати актуальну відповідь на запит.

Крок №6 «Чатбот -> Користувач»: чатбот відправляє користувачу сформовану відповідь, наприклад: «Троянди слід поливати кожні 2-3 дні, в залежності від погодних умов». Мета: надати користувачу корисну відповідь на його запит.

2.4 Моделювання діаграми Use-Case

Для веб-системи підтримки садівників можна визначити кілька основних акторів та use cases (випадків використання). Це допоможе структурувати розуміння функціоналу й інтерфейсу.

Виділено наступних акторів:

1. Садівник (Користувач): основний користувач системи, який доглядає за рослинами.
2. Експерт: фахівець із садівництва, який надає рекомендації або відповідає на запитання користувачів у чаті.
3. Модератор, який відповідає за модерацію чату та контроль якості контенту в системі.
4. Адміністратор системи керує системою загалом, налаштовує базу знань, додає новий контент, керує сповіщеннями та календарем.
5. Гість, тобто, нерегулярний користувач, який має обмежений доступ до бази знань (перегляд певної інформації без створення акаунта).

На рис.2.1 представлено загальну діаграму варіантів використання для веб-системи з підтримки діяльності садівників.

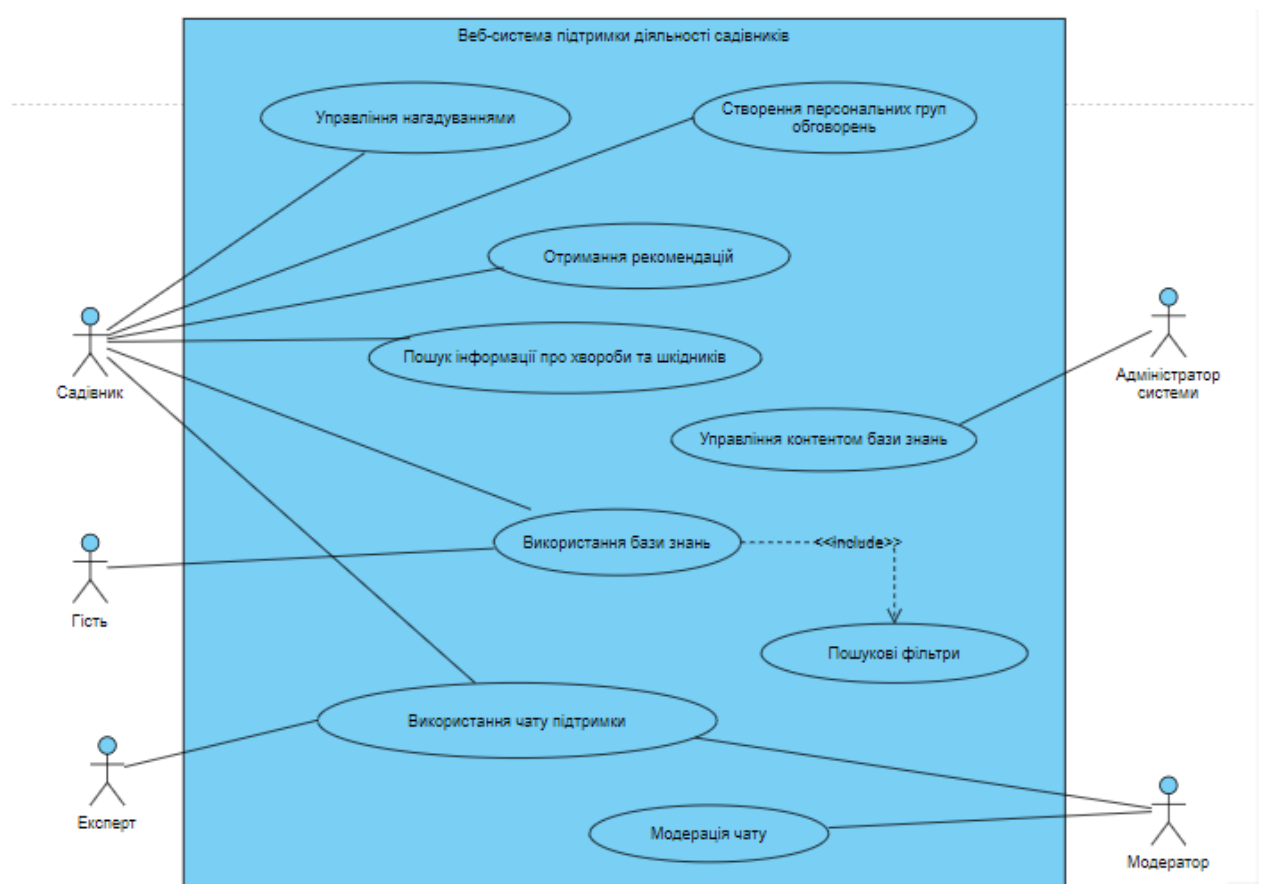


Рисунок 2.1 – Діаграма Use-Case для веб-системи

Розглянемо детально представлені на діаграмі варіанти використання (ВВ) щодо визначених акторів.

ВВ №1 «Використання бази знань».

Актор: Садівник, Гість.

Мета: отримання інформації про догляд за рослинами, боротьбу з хворобами або шкідниками.

Основний сценарій:

1. Користувач відкриває систему.
2. Пошук інформації: користувач вводить ключове слово в пошуковий рядок, наприклад, «базилік» або «жовте листя».
3. Обробка пошуку: система знаходить релевантні статті з бази знань, сортує їх за популярністю, релевантністю чи датою.
4. Відображення результатів: користувач переглядає список статей.
5. Перегляд статті: користувач відкриває потрібну статтю й читає інструкції або рекомендації.

Альтернативні сценарії:

- якщо користувач вводить неточні або неіснуючі терміни, система пропонує список схожих запитів;
- гість бачить лише обмежену кількість статей або скорочений контент і отримує пропозицію зареєструватися для доступу до повної інформації.

ВВ №2 «Отримання рекомендацій».

Актор: Садівник.

Мета: отримати індивідуальні поради по догляду за конкретними рослинами залежно від місцевих умов, сезону тощо.

Основний сценарій:

1. Введення даних: користувач вводить інформацію про рослини, наприклад, які рослини в нього є, де вони ростуть (сад, балкон), місцевий клімат, бажані умови (виращування на ґрунті або в контейнерах).

2. Аналіз даних: система обробляє введену інформацію та використовує базу знань, щоб створити рекомендації (полив, пересадка, удобрення).
3. Відображення рекомендацій: система показує інструкції та графік догляду за кожною рослиною.
4. Додавання до календаря: користувач може автоматично додати нагадування в календар (системний або мобільний) для своєчасного виконання завдань.

Альтернативні сценарії: якщо користувач не вводить всі необхідні дані, система може використовувати стандартні рекомендації або запитувати додаткову інформацію (геолокацію, сезон).

ВВ №2 «Використання чату підтримки».

Актори: Садівник, Експерт, Модератор.

Мета: надання консультацій у реальному часі, можливість спілкування з іншими садівниками.

Основний сценарій:

1. Вхід у чат: користувач відкриває вікно чату через інтерфейс платформи.
2. Задання запитання: користувач формулює запитання, наприклад, "Листя на моїй герані пожовкло. Що робити?".
3. Очікування відповіді: система автоматично надає відповідь (якщо чат-бот реалізований) або передає питання іншим користувачам чи експертам.
4. Отримання відповіді: інший користувач або експерт відповідає у чаті.
5. Оцінка відповіді: користувач може оцінити відповідь (наприклад, лайком або коментарем), щоб показати, чи вона була корисною.

Альтернативні сценарії:

- якщо немає експертів онлайн, користувач може звернутися до чат-бота або залишити питання для відповіді пізніше;

– модератор втручається в обговорення у випадку порушення правил.

ВВ №4 «Управління нагадуваннями».

Актор: Садівник.

Мета: налаштування нагадувань про необхідні дії (полив, обрізка, пересадка тощо).

Основний сценарій:

1. Додавання рослин: користувач додає рослини, що ростуть у нього, до свого профілю.
2. Налаштування нагадувань: користувач встановлює частоту нагадувань, наприклад, полив раз на тиждень, підживлення раз на місяць.
3. Отримання сповіщень: користувач отримує нагадування у системі або на свій телефон про необхідні дії.
4. Підтвердження виконання: користувач відмічає в системі, що виконав дію (наприклад, полив рослину).

Альтернативний сценарій: система автоматично пропонує рекомендації щодо графіка на основі типу рослини або сезонності.

ВВ №5 «Пошук інформації про хвороби та шкідників».

Актор: Садівник.

Мета: ідентифікація хвороб або шкідників рослин за описом симптомів або фото.

Основний сценарій:

1. Введення симптомів: користувач описує симптоми або завантажує фотографію рослини (наприклад, жовте листя, плями на листках).
2. Аналіз системою: система порівнює симптоми з базою даних хвороб і шкідників.
3. Відображення результатів: користувач отримує список можливих захворювань або шкідників з рекомендаціями щодо лікування.
4. Прийняття рішення: користувач обирає відповідні дії або звертається до експерта для підтвердження діагнозу.

Альтернативний сценарій: система може запросити додаткові дані, якщо інформації недостатньо для точного діагнозу.

ВВ №6 «Модерація чату».

Актор: Модератор.

Мета: адміністратор контролює активність в чаті, забезпечує дотримання правил спільноти, видаляє небажаний контент і взаємодіє з користувачами для підтримки продуктивної атмосфери.

Основний сценарій:

1. Авторизація: адміністратор входить в систему через спеціальну панель з розширеними правами доступу до чату.
2. Перегляд активності в чаті:
 - адміністратор переглядає поточні розмови, повідомлення користувачів і обговорення;
 - система може виділяти підозрілий або потенційно небажаний контент для полегшення модерації (на основі автоматичних фільтрів).
3. Модерація повідомлень:
 - видалення повідомлень: адміністратор може видаляти образливі, спамні або неправомірні повідомлення, які порушують правила платформи;
 - редагування повідомлень: якщо дозволено, адміністратор може редагувати повідомлення для корекції некоректних фраз або видалення особистих даних;
 - фільтрація повідомлень: система може автоматично блокувати певні слова або фрази, але адміністратор може переглядати та підтверджувати/відхиляти автоматично приховані повідомлення.
4. Блокування користувачів:
 - тимчасове блокування: адміністратор може тимчасово заблокувати користувачів, які порушують правила, з можливістю розблокування після певного часу;

- постійне блокування: у разі серйозних порушень або повторних скарг, адміністратор може заблокувати користувача назавжди;
- можливість відправки попереджень користувачу перед блокуванням.

5. Перегляд історії чату:

- адміністратор має доступ до історії повідомлень, щоб перевірити контекст обговорень або знайти старі повідомлення для аналізу;
- історія чату допомагає вирішувати конфлікти між користувачами, якщо вони виникають.

6. Реагування на скарги:

- користувачі можуть подавати скарги на інші повідомлення або учасників чату;
- адміністратор переглядає скарги, перевіряє контекст повідомлень і ухвалює рішення щодо їхнього видалення або блокування користувача.

7. Модерація приватних і групових чатів:

- адміністратор може мати доступ до приватних повідомлень користувачів, якщо вони подали скарги на особисту переписку;
- модерація групових чатів за інтересами (наприклад, спеціальні чати для обговорення конкретних типів рослин).

8. Забезпечення правил етикету та взаємодії:

- адміністратор стежить за тим, щоб користувачі дотримувались правил поведінки в чаті, включаючи заборону на рекламу, спам, ненормативну лексику або дискримінаційні висловлювання;
- забезпечення позитивної атмосфери для всіх учасників платформи.

9. Повідомлення про модераційні дії:

- адміністратор може повідомляти користувачам про свої дії (видалення повідомлень, блокування) через систему сповіщень;

- надсилання пояснення, чому конкретне повідомлення було видалене або користувач заблокований.

10.Встановлення автоматичних фільтрів:

- адміністратор може налаштовувати автоматичні фільтри, які блокують або позначають повідомлення за ключовими словами, зокрема ненормативну лексику або образливі висловлювання;
- фільтри можуть оновлюватися для адаптації до нових типів небажаного контенту.

Альтернативні сценарії:

- а) адміністратор може отримувати автоматичні сповіщення про підозрілі повідомлення або користувачів.
- б) при необхідності адміністратор може залучати сторонніх експертів для вирішення конфліктних ситуацій або скарг.

Варіант використання «Модерація чату» для Адміністратора включає контроль і підтримку порядку в чаті, забезпечуючи коректну взаємодію між користувачами, зокрема запобігання образам, спаму та порушенням правил.

ВВ №7 «Створення персональних груп обговорень»

Актор: Садівник.

Опис: користувачі можуть створювати тематичні групи для обговорення певних видів рослин або тем.

Основний сценарій:

1. Користувач створює групу (наприклад, "Органічне садівництво").
2. Інші користувачі приєднуються до групи та обговорюють питання, пов'язані з темою.

Альтернативний сценарій: користувачі можуть ділитися досвідом, обмінюватися порадами та задавати питання, як у загальному чаті, так і в групі.

ВВ №8 «Управління контентом бази знань».

Актор: Адміністратор системи.

Мета: адміністратор управляє контентом платформи, підтримуючи його актуальність, коректність і структуру.

Основний сценарій:

1. Авторизація: адміністратор входить в систему через панель управління з доступом до розширених функцій.
2. Огляд контенту:
 - адмін переглядає статті, посібники, зображення, відео та інший контент, який є на платформі;
 - можливість перегляду списку опублікованих, чернеткових та застарілих матеріалів.
3. Редагування контенту:
 - адміністратор може редагувати текст статей, додавати чи змінювати зображення, відео, інструкції;
 - поле редагування включає текстовий редактор з можливістю форматування, додавання посилань, медіа-контенту (зображення, відео).
4. Створення нового контенту:
 - адміністратор створює нові статті або посібники, додає інформацію про нові хвороби чи шкідників, оновлює рекомендації по догляду;
 - включає завантаження медіа-файлів (фото, відео) для ілюстрацій.
5. Модерація користувацького контенту:
 - адміністратор переглядає й модерує питання та відповіді, додані користувачами (садівниками);
 - можливість схвалювати або відхиляти матеріали, додані користувачами, на основі внутрішніх правил або політики контенту.
6. Видалення або архівація контенту:
 - якщо інформація застаріла або більше не є актуальною, адміністратор може видалити статтю або помістити її в архів;
 - видалення небажаного контенту.

7. Категоризація та теги:

- адміністратор створює та редагує категорії, до яких належать статті або інший контент («Овочі», «Квіти», «Проблеми з шкідниками»);
- можливість додавання тегів для полегшення пошуку та навігації користувачів по контенту.

8. Аналіз та оптимізація:

- аналіз переглядів статей, відгуків користувачів, взаємодії з контентом;
- на основі цих даних адміністратор оптимізує контент, додаючи або редагуючи інформацію для покращення користувацького досвіду.

Альтернативні сценарії:

- а) адміністратор може отримати автоматичні сповіщення про нові публікації користувачів або повідомлення про застарілий контент.
- б) якщо стаття застаріла, адміністратор може запросити автора або експерта для її оновлення.

Отже, функція адміністратору по модерації чату включає в себе багато окремих задач, для візуальної наглядності побудуємо окрему діаграму варіантів використання для актора-Адміністратора (рис. 2.2).

Таким чином, основна мета адміністратора – підтримка високої якості та актуальності контенту на платформі, забезпечення його доступності й корисності для кінцевих користувачів (Садівників).

Адміністратор контролює активність в чаті, забезпечує дотримання правил спільноти, видаляє небажаний контент і взаємодіє з користувачами для підтримки продуктивної атмосфери.

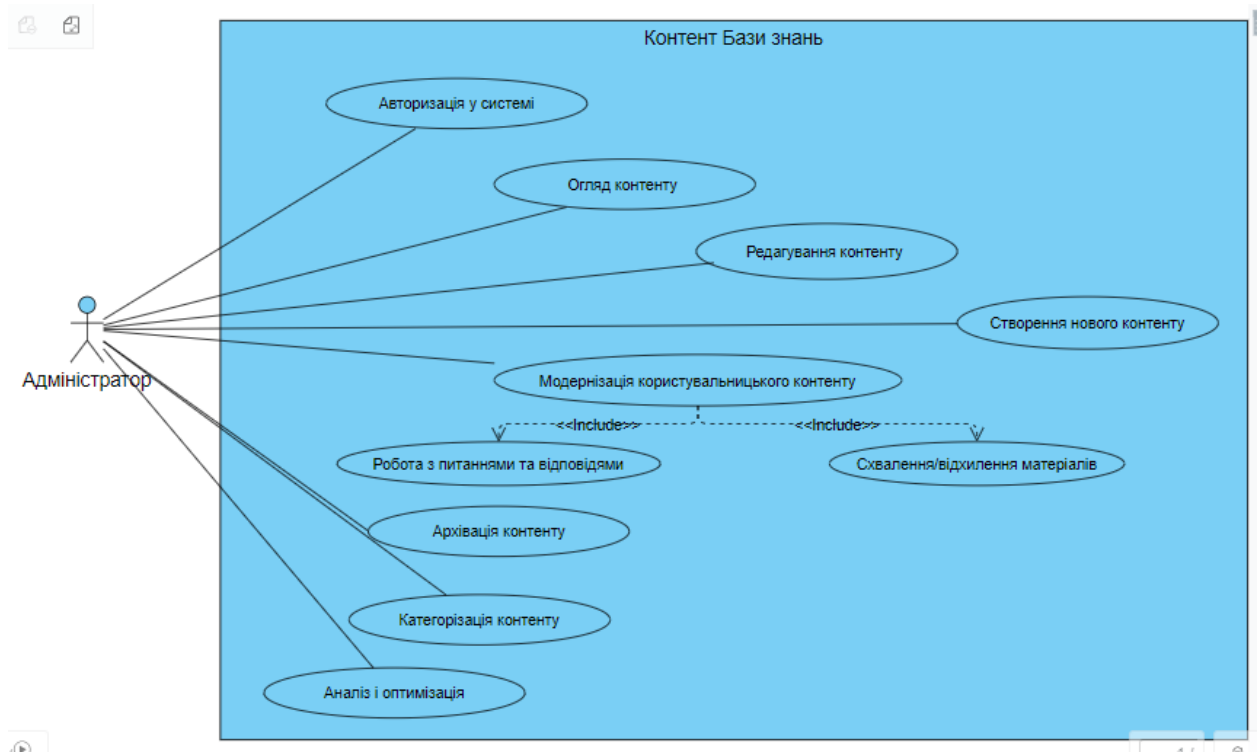


Рисунок 2.2 – Діаграма Use-Case для актора-Адміністратора

2.5 Архітектура веб-системи

Архітектура веб-системи для інформаційної підтримки садівників може бути розділена на кілька шарів та компонентів. Вона базується на модульному підході, де кожен модуль відповідає за свою функціональність (база знань, чат підтримки, система рекомендацій тощо).

1. Клієнтський шар (Frontend):

- користувацький інтерфейс (UI), через який користувач взаємодіє із системою (веб-сайт або мобільний додаток). Він складається з різних сторінок (чат підтримки, пошук в базі знань, рекомендації, календар подій, налаштування нагадувань);
- чат підтримки: інтерфейс для взаємодії з чатботом. Включає введення тексту, кнопки швидких дій, сповіщення.

2. Логічний шар (Backend):

- API Gateway виступає шлюзом для всіх запитів між клієнтською частиною і backend-серверами. Він обробляє всі запити від користувача до бази знань, чатбота, системи рекомендацій та календаря подій;
- чатбот (на основі BERT): обробляє запити користувачів через обробку природної мови (NLP). Він використовує модель BERT для аналізу тексту, розуміння намірів користувача та надання відповідей;
- модуль бази знань містить всю інформацію про рослини, догляд, хвороби, сезонні роботи тощо. Відповідає за надання структурованої інформації на запити, отримані через чатбот або інші компоненти;
- система рекомендацій використовує алгоритми машинного навчання для генерації персоналізованих рекомендацій на основі інформації про рослини, умови вирощування, сезонність і досвід користувача;
- модуль нагадувань і календаря подій відповідає за створення, управління та відображення нагадувань користувачам. Він інтегрований з системою рекомендацій та базою знань для автоматичних рекомендацій і нагадувань (наприклад, про полив або пересадку);
- система аутентифікації та ролей для управління доступом до різних компонентів системи, включаючи ролі адміністратора і модератора;
- модерація чату, тобто функціонал для перегляду історії повідомлень, управління користувачами (блокування, фільтрація контенту), видалення спаму та небажаних повідомлень.

3. Дані (Data Layer).

4. Модуль аналітики та машинного навчання.

Збирає інформацію про запити користувачів, частоту звернень, успішність наданих рекомендацій і аналізує ці дані для покращення роботи системи.

Взаємодія компонентів:

1. Користувач через клієнтський інтерфейс взаємодіє з чатботом і отримує відповіді на свої запитання.
2. Чатбот використовує модель BERT для обробки природної мови та звертається до бази знань або системи рекомендацій для генерації відповідей.
3. Модулі бази даних зберігають всю інформацію і надають її на запити, а модуль аналітики допомагає персоналізувати відповіді та рекомендації.

У другому розділі розглянуто особливості застосування моделі BERT для проектування чатів. Описані такі аспекти, як двонаправлений контекст (Bidirectional Context), архітектура трансформерів (Transformer Architecture), попереднє навчання (Pre-training), донавчання (Fine-tuning) та використання мовних моделей.

Створено модель каталогу рослин, яка включає назву категорії рослини, посилання на чат для обговорень цієї категорії, рекомендації щодо догляду та відповідні події. Визначено та детально описано категорії рослин, а також їх наповнення. Надано опис варіантам використання до основної діаграми Use-Case та визначено архітектуру веб-системи.

3 РОЗРОБКА ВЕБ-СИСТЕМИ

3.1 Вибір технологій

Для Frontend були обрані такі засоби розробки, як: HTML, CSS, JavaScript та фреймворк React. Їх комбінація дозволяє створити масштабовану, ефективну та сучасну веб-систему з гнучким інтерфейсом. React особливо корисний для динамічних елементів системи, таких як чат підтримки або рекомендаційна система, що покращить взаємодію користувачів із веб-системою.

HTML визначає структуру веб-сторінок, що є необхідним для будь-якого веб-застосунку. HTML5 підтримує семантичні теги, що робить структуру коду більш зрозумілою і полегшує розуміння сторінки для розробників, а також пошукових систем (SEO). CSS дозволяє легко змінювати вигляд елементів без зміни HTML-коду. Це дозволяє розділяти структуру від стилів, полегшуючи підтримку та розвиток дизайну. CSS дозволяє створювати адаптивні інтерфейси, які виглядають добре на різних розмірах екранів (мобільних, планшетах, настільних ПК), що важливо для зручності користувачів. Використання сучасних CSS-технологій (Flexbox, Grid) спрощує верстку складних макетів інтерфейсу для ваших компонентів (база знань, календар, чат тощо).

JS – це основна мова програмування для додавання інтерактивності на веб-сторінках. З його допомогою ви можете реалізувати динамічний контент (чат підтримки, оновлення бази знань без перезавантаження сторінки). JS підтримується всіма браузерами, що забезпечує надійну і стабільну роботу вашої веб-системи на будь-якій платформі. JS легко інтегрується з зовнішніми API, що дозволяє обмінюватися даними з сервером, надсилати запити до бази знань, рекомендаційної системи тощо [7].

React (JavaScript бібліотека) – React дозволяє розробляти інтерфейс у вигляді незалежних компонентів, які можуть повторно використовуватись. Наприклад, модуль чату підтримки або компонент нагадувань може бути

окремим, незалежним блоком, що значно спрощує підтримку та розвиток. Одностороння передача даних (One-way Data Flow) робить додатки легшими для тестування та налагодження, що особливо важливо при масштабуванні. React дозволяє писати компоненти з використанням JSX (JavaScript XML), що поєднує JavaScript і HTML в одному файлі, роблячи код більш інтуїтивно зрозумілим та зручним для розробників.

React використовує «віртуальний DOM», що робить оновлення сторінки більш ефективним. Це підвищує продуктивність веб-додатку, оскільки оновлюються лише змінені елементи сторінки. React має велику базу плагінів, бібліотек і компонентів. Ви можете використовувати готові рішення для певних задач, таких як маршрутизація (React Router) або керування станом (Redux, Context API). React легко інтегрується з існуючими технологіями, базами даних та бекенд-фреймворками. Це дозволяє реалізувати швидку взаємодію з сервером через REST API чи GraphQL [8].

На бекенді системи найкраще використовувати Node.js з фреймворком Express.js, базу даних PostgreSQL та ORM Sequelize для управління даними. Для аутентифікації користувачів зручно використовувати JWT, а для чату в реальному часі – Socket.io.

PostgreSQL є одним з найпотужніших та найбільш популярних систем керування реляційними базами даних (СКБД) з відкритим вихідним кодом. Його вибір для вашого проекту є виправданим з багатьох причин:

1. Підтримка реляційних і нереляційних даних PostgreSQL підтримує як реляційні (табличні), так і нереляційні дані (JSON, XML, Hstore). Це дає змогу зберігати складні дані (структури JSON) без потреби в окремій NoSQL базі.
2. Масштабованість PostgreSQL ефективно працює з великими обсягами даних, що робить його чудовим вибором для вашої системи, де буде зберігатися багато інформації про рослини, хвороби, рекомендації, чати тощо.

3. Потужна система запитів PostgreSQL має складну підтримку SQL і надає можливості складних запитів, що робить його підходящим для реалізації складної логіки та аналітики. Це важливо для роботи з базою знань і системою рекомендацій.
4. Можливості розширень PostgreSQL підтримує вбудовані розширення, як PostGIS для геопросторових даних, що може бути корисним для системи, якщо, включити локаційні дані для подій чи рослин.
5. ACID-сумісність і транзакційність PostgreSQL повністю підтримує ACID-принципи (Atomicity, Consistency, Isolation, Durability), що забезпечує надійність і цілісність даних. Це важливо для стабільної роботи системи, особливо коли користувачі надсилають повідомлення, залишають коментарі або отримують нагадування.
6. Надійність і резервне копіювання PostgreSQL має вбудовані механізми резервного копіювання та відновлення, що забезпечує захист даних у разі збою. Можна легко налаштувати резервне копіювання всіх критично важливих даних (користувачів, чату, рослин тощо).
7. Інтеграція з ORM (Object-Relational Mapping) Якщо ви використовуєте React і Node.js, можна інтегрувати PostgreSQL з ORM, такими як Sequelize або TypeORM, для управління складною структурою бази даних на рівні об'єктів, що спрощує роботу з базою.
8. Безпека PostgreSQL має потужні механізми для контролю доступу, включаючи SSL, аутентифікацію та підтримку керування правами користувачів. Це важливо для захисту даних користувачів, чату та іншої чутливої інформації.
9. Гнучке налаштування PostgreSQL дозволяє налаштовувати продуктивність під специфічні вимоги, оптимізуючи базу для високого навантаження або швидкого відгуку системи.

10. Безкоштовність та велика спільнота PostgreSQL є абсолютно безкоштовним і має велику спільноту розробників, що постійно покращує його функціонал та забезпечує наявність багатьох ресурсів для вирішення проблем [9].

Node.js – це популярний JavaScript-пантайм, який дозволяє виконувати код на стороні сервера. Він ідеально підходить для роботи з React (на фронтенді), оскільки підтримує асинхронну модель обробки запитів, що добре підходить для роботи з базою даних, обробки чатів і подій в реальному часі. Завдяки асинхронності Node.js може обробляти велику кількість одночасних запитів, що важливо для систем, які вимагають швидкої реакції (наприклад, чат). JavaScript на фронті і бекенді: єдина мова для обох частин системи спрощує підтримку і розробку. Node.js легко працює з WebSocket для створення чатів у реальному часі, що є ключовою функцією системи [10].

Express.js – це мінімалістичний веб-фреймворк для Node.js, який полегшує розробку REST API і управління запитами. Він забезпечує простоту маршрутизації, обробки запитів, відповіді на них, а також інтеграцію з middleware для різних функцій (наприклад, аутентифікація). Express.js підходить для організації маршрутів (наприклад, запити на отримання рослин, хвороб, рекомендацій, обробки чатів тощо). Швидке створення REST API: для взаємодії з фронтендом (React) та базою даних.

JWT (JSON Web Tokens) для аутентифікації – це стандарт для безпечної передачі інформації між клієнтом і сервером у форматі JSON. JSON Web Tokens легко інтегруються з Express.js. Інформація в JWT підписана, що гарантує її цілісність. JWT можна використовувати токен для авторизації в різних частинах системи (чати, рекомендації, нагадування).

Sequelize – це ORM для Node.js, яка спрощує роботу з базою даних PostgreSQL, дозволяючи взаємодіяти з таблицями як з об'єктами. Це значно спрощує CRUD операції (Create, Read, Update, Delete) та надає зручний API для роботи з даними. Замість написання сирих SQL запитів можна працювати

з об'єктами JavaScript. Sequelize дозволяє автоматизувати зміни в структурі бази даних, що спрощує управління схемою [11].

Passport.js – це модуль аутентифікації для Node.js, який підтримує безліч стратегій (локальна аутентифікація, через соціальні мережі тощо). Можна використовувати як локальну аутентифікацію (логін через email і пароль), так і OAuth для входу через Google або Facebook. Він легко налаштовується і інтегрується з Express.js.

3.2 Концептуальні класи веб-системи

Для розробки веб-системи підтримки садівників можна виділити кілька ключових класів. Вони охоплюють як бекенд, так і фронтенд. Класи для такої системи включають логіку для роботи з користувачами, чатом, базою знань, рекомендаціями, нагадуваннями та іншими компонентами.

1. Клас «User» (Користувач): представляє користувача системи, який може взаємодіяти з різними компонентами.

Атрибути:

- id: унікальний ідентифікатор користувача;
- name: ім'я користувача;
- email: електронна пошта;
- password: пароль (зашифрований);
- role: роль користувача (садівник, модератор, адміністратор);
- preferences: уподобання користувача.

Методи:

- login(): авторизація користувача;
- register(): реєстрація користувача;
- editprofile(): редагування профілю;
- logout(): вихід з системи.

2. Клас «Admin» (Адміністратор): спеціальний клас, що розширює функції користувача, даючи доступ до адміністрування.

Атрибути: успадковує від класу «User».

Методи:

- manageusers(): управління користувачами (додавання, редагування, видалення);
- managecontent(): управління контентом у базі знань;
- manageevents(): створення і редагування подій.

3. Клас «Moderator» (Модератор): користувач з можливістю модерації контенту та чату.

Атрибути: успадковує від класу «User».

Методи:

- moderatechat(): модерація чату, перевірка на порушення правил;
- editcontent(): можливість редагувати та публікувати статті в базі знань.

4. Клас «KnowledgeBase» (База знань): для роботи з базою знань, яка містить статті, поради, інструкції тощо.

Атрибути:

- id: унікальний ідентифікатор статті;
- title: назва статті;
- content: текст статті;
- author: автор статті;
- tags: теги для категоризації.

Методи:

- searcharticles(): пошук статей по базі знань;
- viewarticle(): перегляд статті;
- addarticle(): додавання нової статті (доступно модераторам або адміністраторам).

5. Клас «ChatSession» (Чат-сесія): відображає сесію чату користувача з чат-ботом або модератором.

Атрибути:

- id: унікальний ідентифікатор сесії;

- user: користувач, який ініціював чат;
- messages[]: масив повідомлень;
- status: статус сесії (активна, завершена).

Методи:

- sendmessage(): надсилання повідомлення в чат;
- getchathistory(): отримання історії повідомлень;
- closesession(): завершення сесії.

6. Клас «RecommendationSystem» (Система рекомендацій): генерує персоналізовані рекомендації для користувача на основі уподобань або даних з чату. Атрибут тільки один:

- recommendations[]: масив рекомендацій для користувача.

Методи:

- generaterecommendation(): створення рекомендацій на основі інтересів користувача або запитів;
- viewrecommendation(): перегляд наданої рекомендації.

7. Клас «Reminder» (Нагадування): клас для керування нагадуваннями користувачів про важливі події або дії.

Атрибути:

- id: унікальний ідентифікатор нагадування;
- user: користувач, якому призначене нагадування;
- message: текст нагадування;
- datetime: дата і час, коли нагадування має бути відправлене;
- status: статус нагадування (активне, виконане).

Методи:

- setreminder(): встановлення нового нагадування;
- deletereminder(): видалення нагадування;
- sendreminder(): надсилання нагадування користувачу.

8. Клас «Event» (Подія): для роботи з подіями, в яких можуть прийняти участь користувачі веб-системи (семінари, майстер-класи, інші заходи для садівників).

Атрибути:

- id: унікальний ідентифікатор події;
- title: назва події;
- description: опис події;
- datetime: дата і час події;
- location: місце проведення (якщо офлайн);
- participants[]: список зареєстрованих користувачів.

Методи:

- createevent(): створення нової події;
- registerforevent(): реєстрація на подію;
- vieweventdetails(): перегляд деталей події.

9. Клас «Plant» (Рослина): в ньому зберігається інформація про рослини.

Атрибути:

- id: унікальний ідентифікатор рослини;
- name: назва рослини;
- type: тип рослини (садова, кімнатна тощо);
- careinstructions: інструкції по догляду;
- season: сезонність.

Методи:

- viewplantdetails(): перегляд інформації про рослину;
- addplanttofavorites(): додавання рослини до улюблених.

10. Клас «Notification» (Сповіщення): для керування сповіщеннями, які система відправляє користувачам.

Атрибути:

- id: унікальний ідентифікатор сповіщення;
- user: користувач, якому призначене сповіщення;
- message: текст сповіщення;
- datetime: дата і час відправки сповіщення.

Методи:

- sendnotification(): відправка сповіщення;

- `viewnotificationhistory()`: перегляд історії сповіщень.

У React будуть функціональні компоненти для відображення інформації, роботи з формами (реєстрація, чат, перегляд подій), та інтерфейсу бази знань (Frontend-класи (UI/UX)). На сервері використовуються класи для обробки запитів, управління даними (база знань, користувачі), генерації рекомендацій, та управління нагадуваннями і подіями (Backend-класи).

3.3 Розробка бази даних

Для веб-системи підтримки садівників, яка включає базу знань, чат підтримки, систему рекомендацій та нагадування, структура бази даних може бути організована в кілька основних таблиць. Опишемо кожну з них детально:

1. Таблиця «Users» (Користувачі):

- `user_id` (INT, PK) – унікальний ідентифікатор користувача
- `username` (VARCHAR) – ім'я користувача
- `email` (VARCHAR) – адреса електронної пошти
- `password_hash` (VARCHAR) – хеш пароля
- `created_at` (DATETIME) – дата створення облікового запису

2. Таблиця «Plants» (Рослини):

- `plant_id` (INT, PK) – Унікальний ідентифікатор рослини
- `scientific_name` (VARCHAR) – наукова назва
- `common_name` (VARCHAR) – народна назва
- `description` (TEXT) – опис рослини
- `care_instructions` (TEXT) – інструкції з догляду
- `growing_conditions` (TEXT) – умови вирощування
- `flowering_period` (VARCHAR) – період цвітіння

3. Таблиця «Diseases» (Хвороби):

- `disease_id` (INT, PK) – унікальний ідентифікатор хвороби
- `name` (VARCHAR) – назва хвороби
- `symptoms` (TEXT) – симптоми

- treatment (TEXT) – методи лікування
- prevention (TEXT) – профілактика

4. Таблиця «Pests» (Шкідники):

- pest_id (INT, PK) – унікальний ідентифікатор шкідника
- name (VARCHAR) – назва шкідника
- description (TEXT) – опис шкідника
- damage (TEXT) – пошкодження, які завдає шкідник
- treatment (TEXT) – методи лікування

5. Таблиця «KnowledgeBase» (База знань):

- article_id (INT, PK) – унікальний ідентифікатор статті
- title (VARCHAR) – заголовок статті
- content (TEXT) – вміст статті
- created_at (DATETIME) – дата створення статті
- updated_at (DATETIME) – дата оновлення статті

6. Таблиця «ChatMessages» (Чат-повідомлення):

- message_id (INT, PK) – унікальний ідентифікатор повідомлення
- user_id (INT, FK) – ідентифікатор користувача, який надіслав повідомлення
- content (TEXT) – вміст повідомлення
- timestamp (DATETIME) – дата та час надходження повідомлення

7. Таблиця «Recommendations» (Рекомендації):

- recommendation_id (INT, PK) – унікальний ідентифікатор рекомендації
- user_id (INT, FK) – ідентифікатор користувача
- plant_id (INT, FK) – ідентифікатор рослини
- recommendation_text (TEXT) – текст рекомендації
- created_at (DATETIME) – дата створення рекомендації

8. Таблиця «Reminders» (Нагадування):

- reminder_id (INT, PK) – унікальний ідентифікатор нагадування
- user_id (INT, FK) – ідентифікатор користувача

- reminder_text (TEXT) – текст нагадування
- reminder_date (DATETIME) – дата та час нагадування

9. Таблиця «Reviews» (Відгуки):

- review_id (INT, PK) – унікальний ідентифікатор відгуку
- user_id (INT, FK) – унікальний ідентифікатор користувача
- article_id/plant_id (INT, FK) – зовнішні ключі
- rating (INT) – рейтинг
- comment (TEXT) – текст відгуку
- created_at (DATETIME) – дата створення відгуку

10. Таблиця «UserQueries» (Історія запитів):

- query_id (INT, PK) – унікальний ідентифікатор запиту
- user_id (INT, FK) – унікальний ідентифікатор користувача
- query_text (TEXT) – запит користувача
- timestamp (DATETIME) – дата запиту

11. Таблиця «UserActivity» (Активність користувача):

- activity_id (INT, PK) – унікальний ідентифікатор активності
- user_id (INT, FK) – унікальний ідентифікатор користувача
- article_id/plant_id (INT, FK) – зовнішні ключі
- action_type (VARCHAR) – тип дії (перегляд, збереження тощо)
- timestamp (DATETIME) – дата

12. Таблиця «Events» (Календар подій):

- event_id (INT, PK) – унікальний ідентифікатор події
- user_id (INT, FK) – ідентифікатор користувача
- event_name (VARCHAR) – назва події
- event_date (DATETIME) – дата події
- description (TEXT) – опис події

ER-діаграму БД представлено на рисунку 3.2:

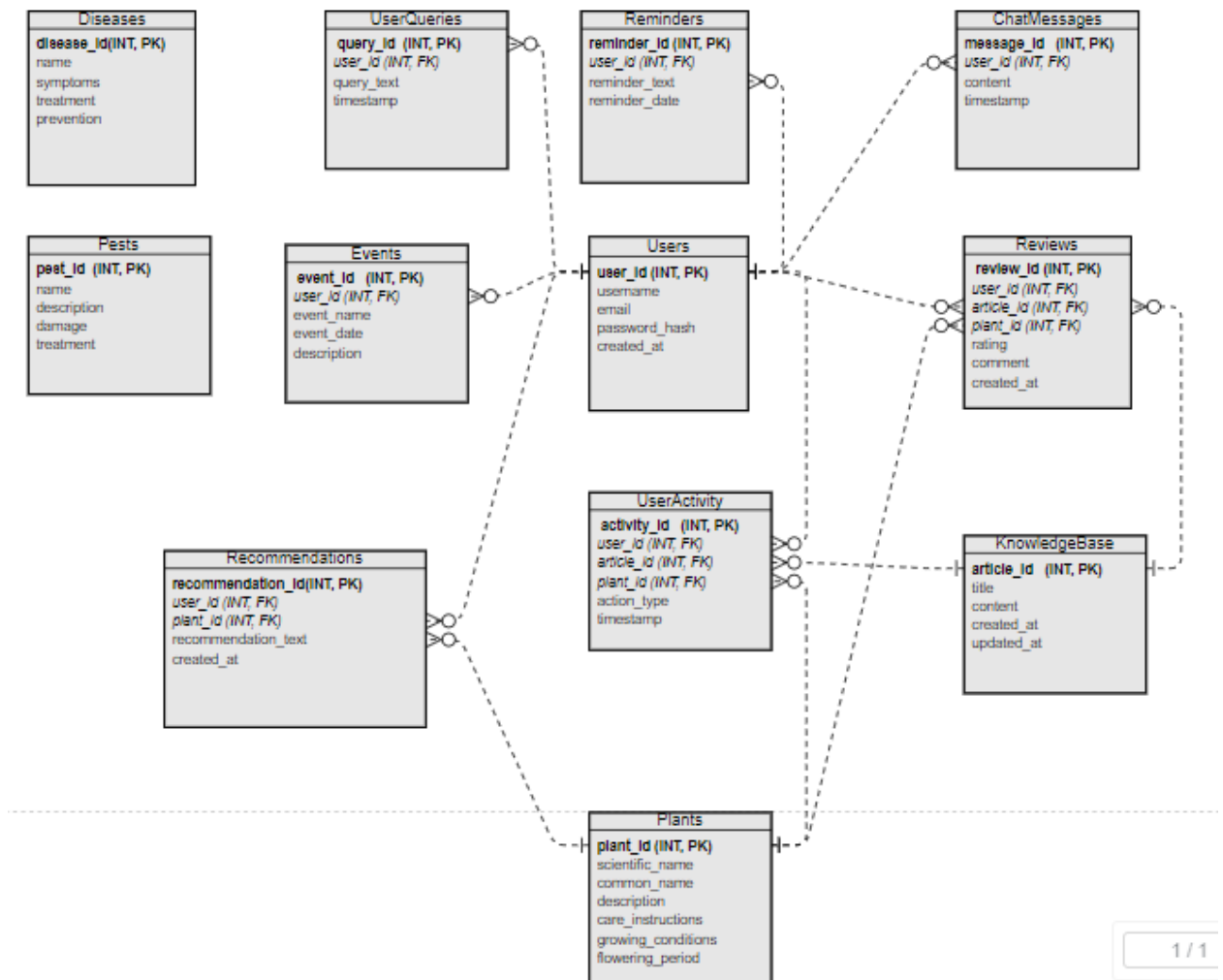


Рисунок 3.2 – Діаграма ER

На першому етапі слід прописати SQL-код для створення таблиці користувачів, так як саме вона буде основною. Крім цього, є таблиці «Рослини», «Хвороби», «Шкідники» і «База знань», які не мають зовнішніх ключей, але інші таблиці будуть пов'язані з ними. Їх створюємо в першу чергу також.

```

-- Таблиця Users (Користувачі)
CREATE TABLE Users (
    user_id SERIAL PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
  
```

```

CREATE TABLE Plants (
    plant_id SERIAL PRIMARY KEY,
    scientific_name VARCHAR(255) NOT NULL,
    common_name VARCHAR(255),
    description TEXT,
    care_instructions TEXT,
    growing_conditions TEXT,
    flowering_period VARCHAR(255)
);

CREATE TABLE Diseases (
    disease_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    symptoms TEXT,
    treatment TEXT,
    prevention TEXT
);

CREATE TABLE Pests (
    pest_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    description TEXT,
    damage TEXT,
    treatment TEXT
);

CREATE TABLE KnowledgeBase (
    article_id SERIAL PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    content TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

Наступний крок, це створення таблиць, які мають зовнішні ключі з попередніми. Це таблиці «Чат-повідомлення», «Рекомендації», «Нагадування», «Календар подій», «Історія запитів». Всі вони мають поле «user_id» для зв'язку з таблицею користувачів:

```

-- Таблиця ChatMessages (Чат-повідомлення)
CREATE TABLE ChatMessages (
    message_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES Users(user_id) ON DELETE CASCADE,
    content TEXT,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Таблиця Recommendations (Рекомендації)
CREATE TABLE Recommendations (
    recommendation_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES Users(user_id) ON DELETE CASCADE,

```

```

        plant_id INT REFERENCES Plants(plant_id) ON DELETE
CASCADE,
        recommendation_text TEXT,
        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    );
CREATE TABLE Reminders (
    reminder_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES Users(user_id) ON DELETE CASCADE,
    reminder_text TEXT,
    reminder_date TIMESTAMP
);
CREATE TABLE Reviews (
    review_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES Users(user_id) ON DELETE CASCADE,
    article_id INT REFERENCES KnowledgeBase(article_id),
    plant_id INT REFERENCES Plants(plant_id),
    rating INT CHECK (rating >= 1 AND rating <= 5),
    comment TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE TABLE UserQueries (
    query_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES Users(user_id) ON DELETE CASCADE,
    query_text TEXT,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE TABLE Events (
    event_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES Users(user_id) ON DELETE CASCADE,
    event_name VARCHAR(255),
    event_date TIMESTAMP,
    description TEXT
);

```

Остання таблиця «Активність користувача» має три зовнішніх ключа, тому її створюємо останньою.

```

-- Таблиця UserActivity (Активність користувача)
CREATE TABLE UserActivity (
    activity_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES Users(user_id) ON DELETE CASCADE,
    article_id INT REFERENCES KnowledgeBase(article_id),
    plant_id INT REFERENCES Plants(plant_id),
    action_type VARCHAR(255),
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

3.4 Тренування нейромережі

Для тренування та тестування BERT-моделі для чату підтримки садівників, тренувальна і тестова вибірки мають складатися з текстових даних, які відображають типові питання та відповіді користувачів. Це будуть реальні діалоги з користувачами або змодельовані питання, які охоплюють різні аспекти садівництва, такі як догляд за рослинами, боротьба зі шкідниками та хворобами, рекомендації тощо.

Тренувальна вибірка (Training set) використовується для навчання моделі. У цьому випадку модель бачить запит користувача та очікувану відповідь, і на основі цього вчиться генерувати правильні відповіді.

Тестова вибірка (Test set) використовується після тренування для оцінки, наскільки добре модель може відповідати на нові або невідомі запити. Тестова вибірка не повинна використовуватися під час навчання, щоб перевірити здатність моделі узагальнювати.

1. Приклад тренувальної вибірки (Training Set):

Текст запиту (вхід): «Як доглядати за трояндами влітку?».

Очікувана відповідь (вихід): «Влітку троянди потребують регулярного поливу, мульчування та обробки від шкідників. Обрізка відцвілих квітів сприяє новому цвітінню.»

Текст запиту (вхід): «Чим обробити рослину від попелиці?».

Очікувана відповідь (вихід): «Для боротьби з попелицею можна використовувати інсектицидні мила або хімічні засоби, такі як перметрин. Також допомагає обробка рослини мильним розчином.»

Текст запиту (вхід): «Коли краще пересаджувати півонії?»

Очікувана відповідь (вихід): «Півонії найкраще пересаджувати восени, коли вони перебувають у стані спокою. Це зазвичай кінець серпня або вересень.»

2. Приклад тестової вибірки (Test Set):

Текст запиту (вхід): «Яка краща температура для вирощування орхідей?»

Очікувана відповідь (вихід): «Орхідеї люблять теплу температуру близько 18-24°C вдень і трохи нижчу вночі.»

Текст запиту (вхід): «Що робити, якщо на листях рослини з'являються жовті плями?»

Очікувана відповідь (вихід): «Жовті плями на листках можуть свідчити про нестачу вологи, хворобу або шкідників. Перевірте умови поливу та огляньте рослину на наявність шкідників.»

На рисунку 3.3 представлено діаграму послідовності для навчання нейронної мережі.

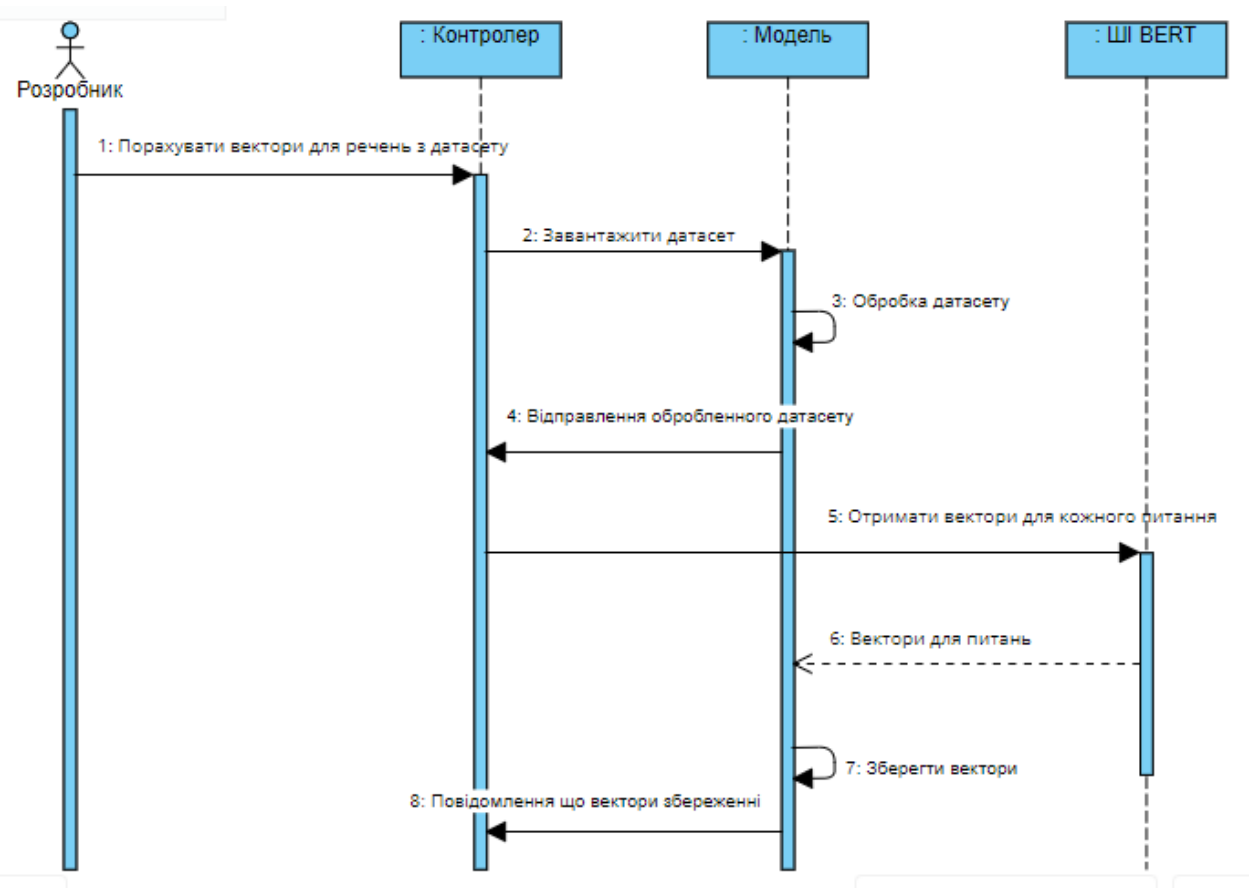


Рисунок 3.3 – Діаграма навчання моделі

У третьому розділі наведено обґрунтування вибору технологій фронтенду та бекенду для реалізації веб-системи. Визначені концептуальні класи системи, а також створено структуру бази даних, яка містить інформацію про користувачів, рослини, категорії, чати, повідомлення, рекомендації, події у вигляді ER-діаграми та SQL-коду. Надано опис алгоритму навчання нейронної мережі на тренувальних і тестових виборках.

4 ТЕСТУВАННЯ РОБОТИ ВЕБ-СИСТЕМИ

4.1 Функціональне тестування

Функціональне тестування веб-системи підтримки садівників включає перевірку того, чи правильно працюють основні функції системи відповідно до вимог. Було підготовлено групу тест-кейсів – конкретні сценарії тестування, які допоможуть перевірити кожен функцію окремо. Розглянемо їх більш детально (табл.4.1-4.10).

Таблиця 4.1 – Тест-кейс «Реєстрація нового користувача»

ID тест-кейсу	ТС-001
Модуль	Реєстрація
Опис	Перевірка можливості зареєструвати нового користувача в системі.
Попередні умови	Користувач не зареєстрований в системі.
Кроки:	1. Відкрити сторінку реєстрації. 2. Ввести ім'я користувача, email та пароль. 3. Натиснути кнопку «Зареєструватися».
Очікуваний результат	Користувач успішно зареєстрований, і система перенаправляє на сторінку профілю або повідомляє про успішну реєстрацію.

Таблиця 4.2 – Тест-кейс «Вхід у систему»

ID тест-кейсу	ТС-002
Модуль	Аутентифікація
Опис	Перевірка входу в систему за допомогою коректного логіна та пароля.
Попередні умови	Користувач зареєстрований у системі.

Продовження таблиці 4.2

Кроки	1. Відкрити сторінку входу. 2. Ввести email і пароль. 3. Натиснути кнопку "Увійти".
Очікуваний результат	Користувач успішно авторизується, і система перенаправляє на головну сторінку або особистий кабінет.

Таблиця 4.3 – Тест-кейс «Додавання нової рослини в базу»

ID тест-кейсу	ТС-003
Модуль	База знань – Рослини
Опис	Перевірка можливості додати нову рослину в базу знань.
Попередні умови	Користувач має права адміністратора.
Кроки:	1. Перейти в розділ «Рослини». 2. Натиснути кнопку «Додати рослину». 3. Ввести наукову назву, народну назву, опис, інструкції з догляду, умови вирощування та період цвітіння. 4. Натиснути кнопку «Зберегти».
Очікуваний результат	Рослина успішно додана до бази знань, і вона з'являється в списку.

Таблиця 4.4 – Тест-кейс «Пошук рослини в базі знань»

ID тест-кейсу	ТС-004
Модуль	База знань – Пошук
Опис	Перевірка функції пошуку рослин за назвою.
Попередні умови	В системі є рослини.
Кроки	1. Перейти на сторінку бази знань. 2. Ввести в поле пошуку наукову або народну назву рослини і натиснути кнопку "Пошук".

Продовження таблиці 4.4

Очікуваний результат	Система відображає результати, які відповідають запиту.
----------------------	---

Таблиця 4.5 – Тест-кейс «Відправка повідомлення в чат підтримки»

ID тест-кейсу	ТС-005
Модуль	Чат підтримки
Опис	Перевірка можливості надсилання повідомлення в чат для отримання підтримки.
Попередні умови	Користувач увійшов у систему.
Кроки:	1. Перейти до розділу чату. 2. Ввести текст повідомлення у відповідне поле. 3. Натиснути кнопку "Відправити".
Очікуваний результат	Повідомлення відправлено і з'являється в чаті. Відповідь чат-бота відображається після аналізу введеного повідомлення.

Таблиця 4.6 – Тест-кейс «Отримання персоналізованих рекомендацій»

ID тест-кейсу	ТС-006
Модуль	Система рекомендацій
Опис	Перевірка можливості отримання персоналізованих рекомендацій на основі обраної рослини.
Попередні умови	Користувач вибрав рослину в системі.
Кроки:	1. Перейти на сторінку обраної рослини. 2. Натиснути кнопку "Отримати рекомендації".
Очікуваний результат	Система відображає рекомендації щодо догляду за рослиною, з урахуванням індивідуальних умов користувача.

Таблиця 4.7 – Тест-кейс «Створення нагадування про подію»

ID тест-кейсу	ТС-007
Модуль	Нагадування
Опис	Перевірка можливості створення нагадування про подію для рослини.
Попередні умови	Користувач увійшов у систему.
Кроки:	1. Перейти в розділ «Нагадування». 2. Натиснути кнопку «Створити нагадування». 3. Ввести текст нагадування та вибрати дату. 4. Натиснути кнопку «Зберегти».
Очікуваний результат	Нагадування успішно створено, і воно з'являється в календарі подій.

Таблиця 4.8 – Тест-кейс «Оцінка та відгук на статтю»

ID тест-кейсу	ТС-008
Модуль	Відгуки
Опис	Перевірка можливості залишити відгук і оцінку статті в базі знань.
Попередні умови	Користувач увійшов у систему та переглянув статтю.
Кроки:	1. Перейти до статті в базі знань. 2. Прокрутити вниз до розділу "Відгуки". 3. Ввести коментар і вибрати рейтинг (від 1 до 5). 4. Натиснути кнопку «Залишити відгук».
Очікуваний результат	Відгук успішно збережено, і він відображається під статтею.

Крім вищеописаних функцій, слід перевірити роботу нагадування на запланований час користувачем та видалення користувачем свого облікового запису з системи.

Таблиця 4.9 – Тест-кейс «Перевірка отриманого нагадування»

ID тест-кейсу	ТС-009
Модуль	Нагадування
Опис	Перевірка того, чи нагадування відображається у встановлений час.
Попередні умови	Нагадування створене та заплановане на поточний день.
Кроки:	1. Дочекатися дати й часу, на які було встановлено нагадування. 2. Перевірити, чи з'явилося сповіщення.
Очікуваний результат	Система відправляє сповіщення з текстом нагадування у встановлений час.

Таблиця 4.10 – Тест-кейс «Видалення користувачем свого облікового запису»

ID тест-кейсу	ТС-010
Модуль	Акаунт користувача
Опис	Перевірка можливості видалення облікового запису користувача.
Попередні умови	Користувач увійшов у систему.
Кроки:	1. Перейти в налаштування облікового запису. 2. Натиснути кнопку «Видалити обліковий запис». 3. Підтвердити видалення.
Очікуваний результат	Обліковий запис успішно видалено, і користувач більше не має доступу до системи.

Ці тест-кейси охоплюють основні функціональні можливості системи, дозволяючи перевірити коректність роботи всіх важливих компонентів веб-застосунку.

Результати перевірки кожного тест-кейсу представлено в таблиці 4.11.

Таблиця 4.11 – Результати функціонального тестування

ID тест-кейсу	Назва тест-кейсу	Актори	Результат тестування
ТС-001	Реєстрація нового користувача	Новий користувач	Успішно
ТС-002	Вхід у систему	Зареєстрований користувач	Успішно
ТС-003	Додавання нової рослини в базу	Адміністратор	Успішно
ТС-004	Пошук рослини в базі знань	Користувач	Успішно
ТС-005	Відправка повідомлення в чат підтримки	Користувач	Успішно
ТС-006	Отримання персоналізованих рекомендацій	Користувач	Успішно
ТС-007	Створення нагадування про подію	Користувач	Успішно
ТС-008	Оцінка та відгук на статтю	Користувач	Успішно
ТС-009	Перевірка отриманого нагадування	Користувач	Успішно
ТС-010	Видалення користувачем свого облікового запису	Користувач	Успішно

4.2 Тестування користувацького інтерфейсу (UI-тестування)

UI-тестування фокусується на перевірці правильності відображення та функціонування елементів інтерфейсу, таких як кнопки, поля введення, форми, меню та інші інтерактивні компоненти. Це включає перевірку зручності використання (юзабіліті), зовнішнього вигляду і реакції елементів на дії користувача, а також відповідність дизайну вимогам.

Також можна окремо розглядати UX-тестування (User Experience), яке стосується не тільки зовнішнього вигляду, але і загального досвіду використання системи з точки зору користувача, включаючи навігацію та інтуїтивність. Перевіримо роботу інтерфейсу користувача.

Алгоритм отримання доступу до повного спектру меню веб-системи з підтримки діяльності садівника представлено на рисунку 4.1. На першому кроці користувачу слід пройти авторизацію/регістрацію, після підтвердження

введених даних йому буде відкрито доступ до чату підтримки та календаря подій.

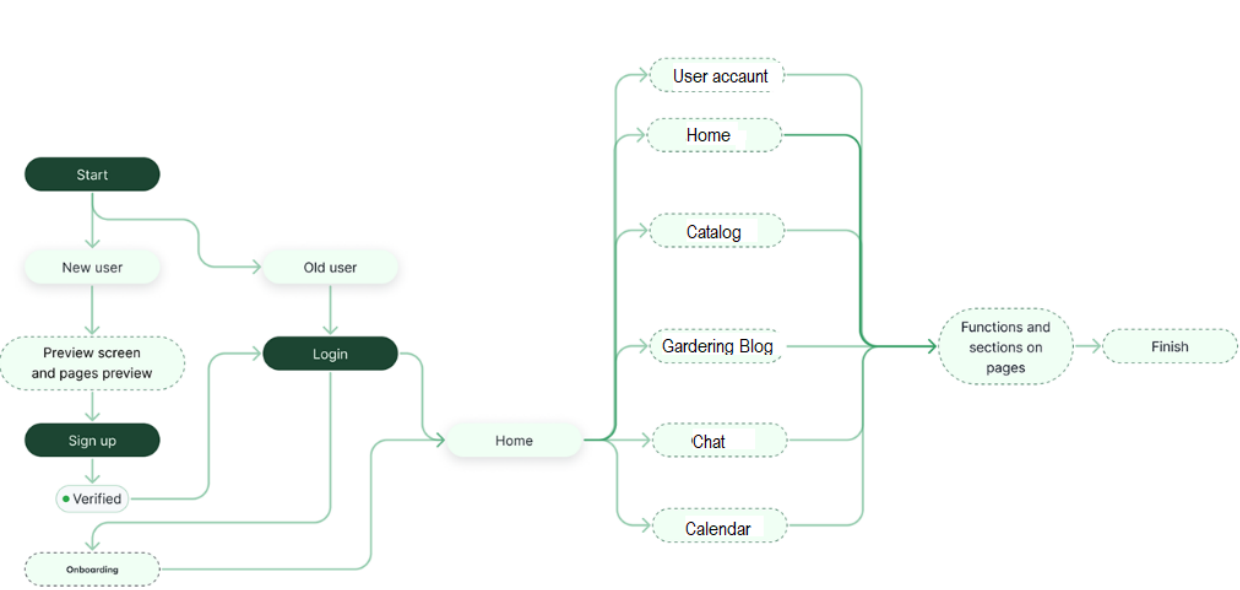


Рисунок 4.1 – Алгоритм отримання доступу для користувача системи

Стартову сторінку веб-системи з підтримки діяльності користувача представлено на рис.4.2.

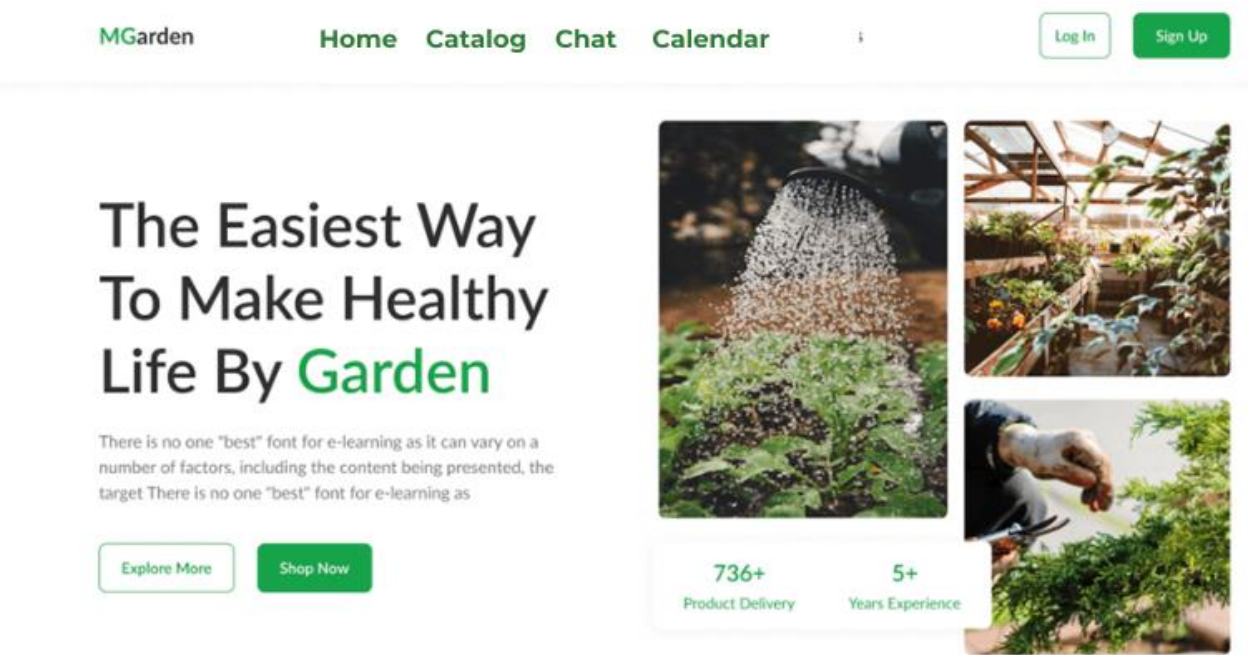


Рисунок 4.2 – Стартова сторінка «MGarden»

На закладці «Номе» (рис. 4.3) представлено стислу інформацію щодо веб-системи та перелік експертів, які будуть проводити консультації у чаті підтримки. Крім цього, тут розміщено актуальні пости (рис. 4.4) за перелік основних розділів інформації для садівників.



Рисунок 4.3 – Скріншот закладки «Номе»

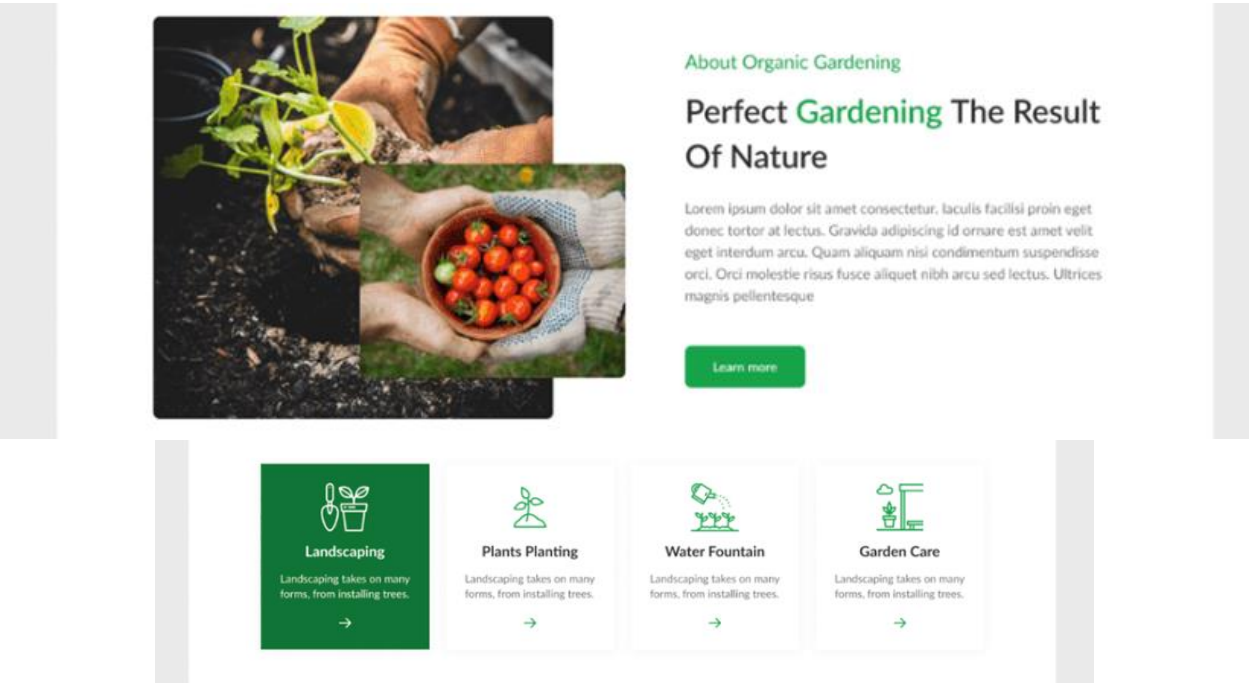


Рисунок 4.4 – Скріншот блоку нових публікацій

Закладка «Catalog» містить пошуковий рядок, який дозволяє користувачам швидко знаходити рослини, хвороби, шкідників або статті з бази знань за ключовими словами. Крім цього, ту розміщено фільтри: для точнішого пошуку, користувачі можуть відфільтровувати дані за категоріями:

- тип рослини (квіти, дерева, овочі тощо);
- умови вирощування (світло, вологість, температура);
- період цвітіння або збирання врожаю;
- тип проблеми (хвороби, шкідники, дефіцит поживних речовин тощо).

Так, обравши у пошуку зелень, користувачу відкриється перелік карточок елементів каталогу з фото та назвою до кожної з них (рис. 4.4). Для перегляду детальної інформації (опис рослини/хвороби/шкідника, умови вирощування, симптоми, методи лікування або боротьби, рекомендації щодо догляду) про обрану рослину слід натиснути на його зображення.

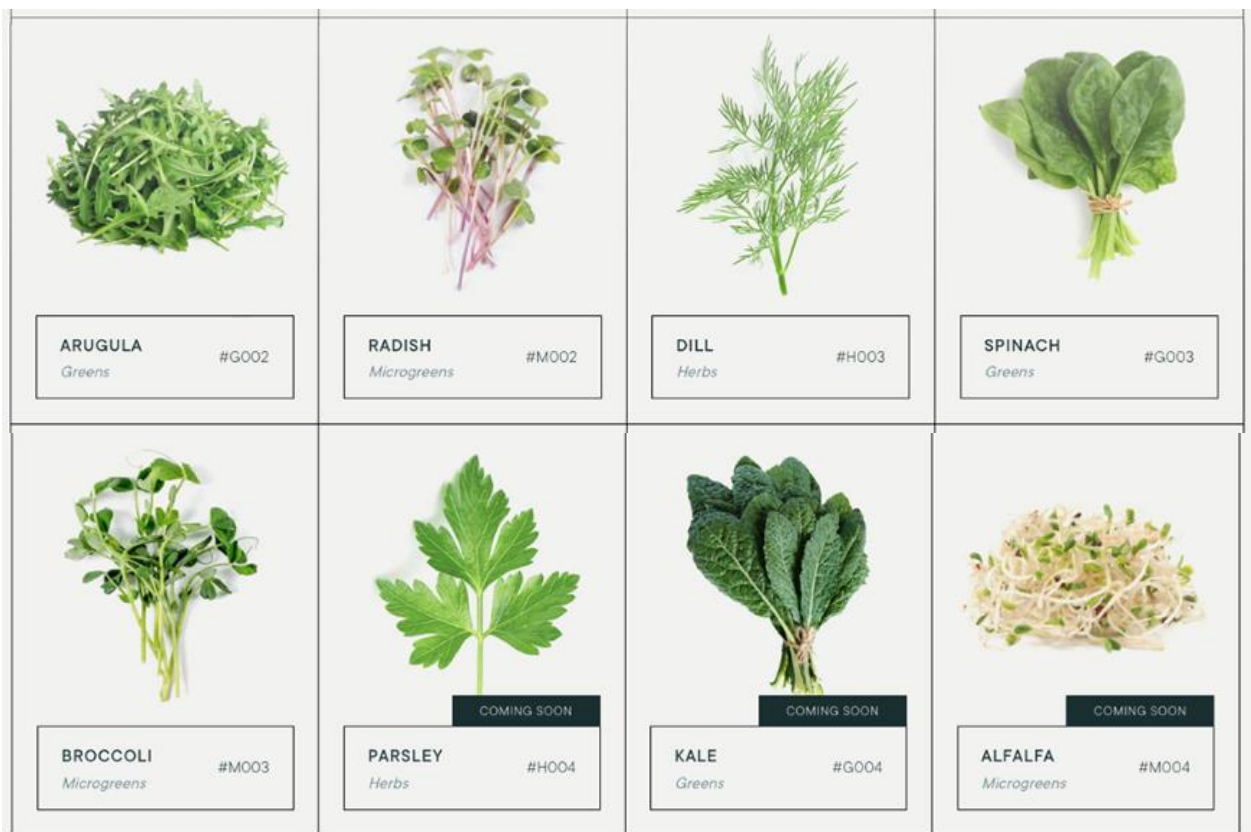


Рисунок 4.4 – Скріншот блоку «Зелень»

У четвертого розділі описані тест-кейси, за основі яких проведена перевірка виконання тестових сценаріїв для підтвердження коректності роботи програми. Виконано тестування користувацького інтерфейсу, під час якого перевірено коректність відображення та функціонування елементів інтерфейсу, таких як кнопки, поля введення, форми, меню та інші інтерактивні компоненти.

ВИСНОВКИ

Створення веб-системи підтримки садівників дозволяє вирішувати актуальні проблеми в догляді за рослинами, боротьбі з хворобами та шкідниками, а також покращувати комунікацію між користувачами. Це сприяє популяризації садівництва та забезпечує зручний інструмент для новачків і досвідчених садівників.

У результаті розроблено багатофункціональну систему, яка містить інтерактивний каталог рослин, базу знань, чат підтримки, систему рекомендацій, нагадування та календар подій. Це забезпечує користувачів всіма необхідними інструментами для ефективного управління садівництвом.

Використання сучасних технологій, таких як React для фронтенду, Node.js і PostgreSQL для бекенду, а також інтеграція моделі BERT для аналізу запитів у чаті, дозволяє забезпечити високий рівень зручності та ефективності роботи системи.

Система вирішує низку проблем, зокрема:

- недостатня обізнаність про догляд за рослинами;
- складність у діагностиці хвороб рослин і боротьбі зі шкідниками;
- потреба в оперативних нагадуваннях та рекомендаціях.

Проект є масштабованим, оскільки його структура дозволяє додавати нові функції, розширювати базу даних рослин, хвороб і шкідників, а також інтегрувати додаткові сервіси, наприклад, погодні прогнози чи рекомендації на основі локальних даних. Завдяки базі знань і рекомендаціям користувачі можуть здобувати нові знання та досвід у садівництві, що сприяє їхньому особистому розвитку.

Розроблена система зможе об'єднати садівників у спільноту, сприяючи обміну досвідом і взаємній підтримці через чат і соціальні функції.

СПИСОК ВИКОРИСТОВАНИХ ДЖЕРЕЛ

1. Основні моменти сучасного інтенсивного садівництва. URL: <https://schetelig.com.ua/stati-ua/osnovi-suchasnogo-intensivnogo-sadivnictva-konceptualni-momenti-ua.html> (дата звернення 15.08.2024)
2. Garden Plan Pro App. URL: <http://www.gardenplanpro.com/> (дата звернення 15.08.2024)
3. Best 9 gardener software for small businesses [in2024]. URL: <https://vev.co/blog/best-gardener-software-for-small-businesses> (дата звернення 15.08.2024)
4. Top 5 Garden Designing Apps for Professional Gardeners. URL: <https://treengarden.com/articles/detail/Top-5-Garden-Designing-Apps-for-Professional-Gardeners> (дата звернення 15.08.2024)
5. Як покращується обробка мови за допомогою моделі Google BERT з відкритим кодом. URL: <https://www.unite.ai/uk/how-language-processing-is-being-enhanced-through-googles-open-source-bert-model/> (дата звернення 03.09.2024)
6. Що таке обробка природної мови (NLP). URL: <https://metinvest.digital/ru/page/1052> (дата звернення 03.09.2024)
7. Why are JavaScript, HTML and CSS languages so popular. URL: <https://www.quora.com/Why-are-JavaScript-HTML-and-CSS-languages-so-popular-among-programmers>. (дата звернення 12.09.2024)
8. React. JavaScript-бібліотека для створення користувацьких інтерфейсів. URL: <https://uk.legacy.reactjs.org/> (дата звернення 16.09.2024)
9. PostgreSQL: The world's most advanced open source database. URL: <https://www.postgresql.org/> (дата звернення 22.09.2024)
10. Що таке Node JS простими словами. URL: <https://dan-it.com.ua/uk/blog/chto-jeto-takoe-node-js-prostymi-slovami/> (дата звернення 06.10.2024)

11. Як Sequelize допомагає в роботі з базами даних у Node.js. URL:
<https://foxminded.ua/sequelize-shcho-tse/> (дата звернення 13.10.2024)