МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет комп’ютерних наук

Кафедра програмної інженерії

Специфікація

Тема:

Веб-застосунок для ідентифікації та підбору кімнатно-садових рослин

Студент гр. ПЗПІ-21-10 Артем ПЕЧЕНЕВСЬКИЙ

Харків 2025

# 1 ВСТУП

### 1.1 Огляд продукту

Даний документ описує вимоги до веб-застосунку для ідентифікації та підбору кімнатно-садових рослин. Програмний продукт призначено для використання як фахівцями-ботаніками, так і звичайними користувачами, які прагнуть дізнатися більше про рослини.  
Основна функціональність веб-застосунку полягає в ідентифікації рослини за її зображенням, відображенні детальної інформації про неї.

### 1.2 Мета

Метою створення веб-застосунку є полегшення догляду за рослинами для звичайних користувачів та ентузіастів за допомогою інструментів ідентифікації, представлення додаткової інформації про рослини та інтегрованого форуму для обговорення.

### 1.3 Межі

Проєкт реалізовано в обмеженому обсязі відповідно до цілей дипломної роботи.  
У межах цієї розробки не передбачено реалізацію таких можливостей:

- інтеграції з іншими сервісами або платформами (наприклад, соціальними мережами);

- мобільного застосунку або офлайн-режиму роботи;

- підтримки мультимовного інтерфейсу та перемикання мов;

- розширених можливостей адміністративного керування (наприклад, система ролей, журнал подій);

- складної аналітики або статистики користувацької активності.

Система реалізована як **веб-застосунок, що працює у браузері на десктопних пристроях**, з базовим функціоналом: реєстрація, перегляд, пошук, збереження та керування інформацією про рослини.

У майбутньому функціональні можливості можуть бути розширені в межах наступних етапів розробки або оновлень.

### 1.4 Посилання

- Методичні вказівки до кваліфікаційної роботи бакалавра, ХНУРЕ, 2025 (https://software.nure.ua/students-2/)

- ДСТУ - <https://software.nure.ua/wp-content/uploads/2021/02/DSTU_3008-2015.pdf>

- Головна сторінка Supabase - https://supabase.com/

- Головна сторінка **Render - https://render.com/**

### 1.5 Означення та абревіатури

У цьому документі можуть використовуватися скорочення та терміни:

**- API** — прикладний програмний інтерфейс (Application Programming Interface)

**- БД** — база даних

# 2. ЗАГАЛЬНИЙ ОПИС

### 2.1 Перспективи продукту

У перспективі передбачається розширення функціоналу веб-застосунку шляхом створення мобільної версії, інтеграції з інтелектуальним помічником на базі штучного інтелекту, а також реалізації системи push-сповіщень для нагадування користувачам про необхідні дії з догляду за рослинами (наприклад, полив, підживлення тощо).

### 2.2 Функції продукту

Користувач веб-застосунку зможе:

- Завантажити фотографію рослини;

- Отримати ймовірну назву рослини;

- Отримати коротку та розширену інформацію щодо догляду за рослиною;

### 2.3 Характеристики користувачів

Типовим користувачем системи є звичайний власник кімнатних чи садових рослин віком від 14 до 55 років. Середовище орієнтоване на людей без глибоких ботанічних знань, які мають базові навички користування ПК та інтернетом. Основною мотивацією користувача є бажання ідентифікувати невідому рослину та отримати поради щодо догляду за нею.

### 2.4 Загальні обмеження

На даному етапі реалізації веб-застосунок функціонує лише за наявності підключення до інтернету. Підтримується лише десктопна версія сайту. Обмеження на розмір зображення встановлені на рівні 5 МБ.

### 2.5 Припущення й залежності

Система передбачає, що користувач має змогу зробити фото на мобільний пристрій та завантажити його на ПК для подальшого використання в системі. Для обробки зображень використовується сторонній стабільно працюючий API. Сервер системи повинен бути доступним у момент здійснення запиту користувача.

# 3. КОНКРЕТНІ ВИМОГИ

## 3.1 Вимоги до зовнішніх інтерфейсів

### 3.1.1 Інтерфейс користувача

Веб-застосунок включає такі основні сторінки:

**- Головна сторінка** — короткий опис сервісу та кнопка для переходу до завантаження фото;

**- Сторінка завантаження фото** — інтерфейс для додавання зображення та запуску ідентифікації;

**- Сторінки рослин** — інформація про рослини, включаючи їх опис, рекомендації з догляду.

**- Профіль користувача** — особисті дані та історія запитів.

Усі інтерфейси адаптовані під десктопну версію та оптимізовані для простого використання навіть недосвідченими користувачами.

### 3.1.2 Апаратний інтерфейс

У межах розробки даного веб-застосунку апаратний інтерфейс **не використовується**, оскільки система працює виключно в браузері та не взаємодіє з фізичними пристроями (сенсори, контролери тощо).

### 3.1.3 Програмний інтерфейс (API)

Система використовує клієнт-серверну архітектуру.  
Веб-застосунок (клієнт) взаємодіє з сервером через REST API для:

- завантаження зображення;

- отримання результатів ідентифікації;

- реєстрації та автентифікації користувача;

- перегляду профілю користувача.

Стороннє API для ідентифікації рослин буде обрано та реалізовано на подальших етапах розробки.

### **3.1.4 Комунікаційний протокол**

Для взаємодії між клієнтською та серверною частиною веб-застосунку використовується протокол **HTTP/HTTPS**, який забезпечує обмін запитами та відповідями. Протокол **HTTPS** рекомендований для захищеного з’єднання, особливо під час авторизації та передавання персональних даних.

Клієнт здійснює запити до сервера з використанням методів **GET**, **POST**, **PATCH**, **DELETE**. Відповіді повертаються у форматі **JSON**, що спрощує обробку даних і взаємодію між компонентами.

### **3.1.5 Обмеження пам’яті**

На поточному етапі проєктування не встановлено жорстких обмежень щодо використання **оперативної пам’яті**, **обсягу бази даних** або інших ресурсів пристрою. Єдине обмеження, реалізоване на рівні інтерфейсу, — це **максимальний розмір зображення**, яке завантажується користувачем: **до 5 МБ**.

Обмеження щодо розміру таблиць, кількості користувачів, кількості збережених записів тощо наразі не застосовуються, але можуть бути введені у наступних версіях системи після аналізу навантаження.

### **3.1.6 Операції**

Веб-застосунок передбачає виконання таких основних операцій:

- завантаження зображення рослини через інтерфейс;

- надсилання зображення на сервер для обробки;

- отримання результату ідентифікації;

- реєстрація нового користувача;

- авторизація через електронну пошту та пароль;

- збереження рослин у персональний список;

- перегляд детальної інформації про рослину.

Усі операції реалізовані за принципом запит–відповідь у межах клієнт-серверної архітектури.

### **3.1.7 Функції продукту**

Веб-застосунок виконує такі основні функції:

- автоматичне розпізнавання рослини за зображенням;

- відображення структурованої інформації про знайдену рослину (назва, опис, умови догляду);

- збереження рослин у персональний обліковий запис користувача;

### **3.1.8 Припущення й залежності**

Система передбачає, що користувач має змогу зробити якісне зображення рослини на мобільний пристрій та завантажити його у веб-застосунок зі свого персонального комп’ютера. Також передбачається наявність стабільного інтернет-з’єднання, оскільки всі ключові процеси (ідентифікація та збереження) виконуються через сервер.

Крім того, функціонування системи частково залежить від стабільності стороннього API для ідентифікації рослин, яке повинно мати достатню точність та підтримувати сучасні формати обміну даними.

## **3.2 Властивості програмного продукту**

Програмне забезпечення реалізує такі ключові можливості:

- Ідентифікація рослини за зображенням;

- Показ назви, опису, догляду;

- Обліковий запис користувача з можливістю збереження рослин;

- У майбутньому — сповіщення, редагування профілю, нові функції.

Для реалізації веб-застосунку використано такі основні технології:

**- HTML, CSS, JavaScript** — для створення інтерфейсу користувача (фронтенд);

**- Node.js та Express.js** — як серверна частина (бекенд) для обробки запитів і взаємодії з базою даних;

**- Supabase** — як повноцінне рішення для зберігання даних (на базі PostgreSQL), автентифікації користувачів (Supabase Auth), а також обробки запитів до REST API;

**- Render** — для хостингу серверної частини проєкту;

**- Figma** — для створення макетів інтерфейсу користувача;

**- GitHub** — для контролю версій, зберігання та відстеження змін у коді.

**Supabase Auth** використовується як тимчасове рішення для безпечної реєстрації та входу користувачів, управління сесіями, а також контролю доступу до функцій системи відповідно до ролей (звичайний користувач, ботанік, адміністратор). У майбутньому можливий перехід на альтернативні або більш розширені механізми авторизації.

**Середовище розробки:** для написання коду використовувалися текстовий редактор Notepad++ та середовище Visual Studio Code. У VS Code виконувалося структурування проєкту, редагування окремих файлів, а також робота з терміналом. Для зручного збереження змін і синхронізації проєкту з репозиторієм використовувався клієнт GitHub Desktop.

## **3.3 Атрибути програмного продукту**

### 3.3.1 Надійність

У разі помилок (наприклад, збій API) система повідомляє користувача та дозволяє повторити дію.

### 3.3.2 Доступність

У тестовому режимі — тимчасова недоступність допускається. Після релізу — 24/7 із плановими оновленнями.

### 3.3.3 Безпека

Паролі зберігаються у хешованому вигляді. Всі персональні дані недоступні іншим користувачам. HTTPS використовується для захисту.

### **3.3.4 Супроводжуваність**

Підтримка адміністратором та ботаніками. Кодова база модульна, зручна для оновлень і розширень.

### 3.3.5 Переносимість

Застосунок працює в будь-якому сучасному десктопному браузері (Chrome, Firefox).

### 3.3.6 Продуктивність

На початковому етапі: до 60 секунд на обробку фото. У майбутньому — до 15 секунд.

## **3.4 Вимоги бази даних**

Використовується реляційна БД (Supabase). Основні таблиці:

- users: email, пароль, роль;

- plants: назва, опис, тип, полив, фото;

- saved\_plants.

## **3.5 Інші вимоги**

- Функції доступні лише зареєстрованим користувачам;

- Роль “admin” має розширені можливості;

- Оптимізація під десктоп.