Software Design Specification (SDS)
For PotatoSavior

Prepared By
Md Nazmus Saquib Khan

1. Introduction

1.1 Purpose:
The purpose of this document is to outline the detailed design and architecture of the PotatoSavior website, a consulting platform aimed at supporting farmers by diagnosing potato plant health issues and providing guidance through educational resources and expert services.

1.2 Scope:
The design includes four primary sections:

- **Company Information Page**: Provides details about PotatoSavior's mission and services.
- **Potato Plant Care Page**: Offers care tips and downloadable guides.
- **Disease Information Page**: Explains common diseases like early and late blight.
- **Plant Health Check Page**: Implements a diagnostic tool using a deep learning model for image-based analysis.

2. System Overview

2.1 Architecture:
The system follows a three-tier architecture:

- **Frontend**: Built with HTML, CSS, and Bootstrap for styling.
- **Backend**: Developed using FastAPI to handle API requests and integrate the deep learning model.
- **Database**: PostgreSQL for storing user inquiries, uploaded images, and diagnostic results.

2.2 Technology Stack:

- Frontend: HTML, CSS, Bootstrap.
- Backend: FastAPI (Python).
- Database: PostgreSQL.
- Deep Learning Model: CNN

3. Detailed Design

3.1 Company Information Page

- **Functionality:**
    - Displays static content about the company's mission and services.
    - Includes an inquiry form for consulting services.
- **Design Elements:**

- Header with navigation links.
- Content sections for mission, vision, and form.

3.2 Potato Plant Care Page

- **Functionality:**
  - Provides care tips with instructions.

- **Design Elements:**
  - Organized sections with tips and images.

3.3 Disease Information Page

- **Functionality:**
  - Displays detailed information on early and late blight.
  - Links users to consulting services for additional help.
- **Design Elements:**
  - Sections with text and images for each disease.

3.4 Plant Health Check Page

- **Functionality:**
  - Enables users to upload images of potato plants.
  - Runs diagnostic analysis and displays results.
- **Design Elements:**
  - Image upload form with drag-and-drop functionality.
  - Loading indicator and results display.
- **APIs:**
  - Upload images to the backend for model analysis.
  - Return diagnostic results and recommendations.

4. Data Flow

- **Frontend to Backend:**
  - User actions (e.g., image upload or inquiry form submission) trigger API calls.
- **Backend Processing:**
  - Validates inputs, stores data in the database, and integrates the deep learning model.
- **Database:**
  - Stores form submissions, diagnostic results, and related metadata.

5. Security

- Implement HTTPS for secure communication.
- Restrict image uploads to supported formats (JPEG, PNG) and size (max 5MB).

6. Testing Plan

6.1 Unit Testing:

- Test individual components such as the inquiry form and image upload feature.

6.2 Integration Testing:

- Verify communication between the frontend, backend, and database.

6.3 Performance Testing:

- Measure the response time of the diagnostic tool.

6.4 User Acceptance Testing:

- Ensure the platform meets usability standards for the target audience.

7. Deployment

- Monitor user activity and system performance for future improvements.