

CropCare

Submitted by: Rakshitha T G

USN: 24SUPMCAGL066

MCA – Sapthagiri NPS University

Github: <https://github.com/Rakshithagovind05/cropcare>

S. No	Content	Page No.
1	Introduction	1
2	Problem Statement	1
3	Objectives	2
4	Proposed Solution	2
5	Technology Stack and Decision Rationale	3
6	Database Design	4
7	System Workflow	4
8	Modules Implemented <ul style="list-style-type: none">• Login & Registration Module• Crop Module• Disease Detection Module• Remedies Module• Upload Module	5
9	Future Enhancements	9
10	Conclusion	9

CropCare

1.INTRODUCTION:

Agriculture is one of the most important sectors of our economy, and plant diseases often lead to severe reductions in both the quality and quantity of crops. Farmers frequently face challenges in identifying diseases and finding appropriate remedies on time. To address this problem, **CropCare** has been developed — a smart web-based farming application that assists farmers in detecting crop diseases and suggesting suitable treatments.

This application is built using **Spring Boot, Java, HTML, CSS, and MySQL**. It provides an easy-to-use interface where users can explore various crop diseases, view symptoms, and access recommended remedies. The system also includes modules for user registration, crop information, disease details, and image upload for future AI-based analysis. By integrating technology with agriculture, CropCare aims to promote **smart farming**, reduce crop loss, and support sustainable agricultural practices.

2.Problem Statement:

Farmers often struggle to identify crop diseases accurately and in a timely manner, leading to reduced yield, poor crop quality, and economic loss. Traditional methods of disease detection depend heavily on expert knowledge, which is not always accessible in rural areas. Moreover, farmers may lack awareness of the correct remedies or preventive measures for specific diseases.

There is therefore a pressing need for an easy-to-use digital platform that helps farmers quickly identify crop diseases, understand their causes, and find effective solutions. The **CropCare** web application aims to solve this problem by providing a centralized, accessible, and reliable system that offers disease information, symptoms, and suitable remedies for major crops — all through a simple web interface.

3. Objectives

The main objective of the **CropCare** web application is to assist farmers in identifying crop diseases easily and to suggest appropriate remedies for effective crop management. To achieve this goal, the system focuses on the following specific objectives:

1. **Simplify Disease Identification:**
Provide an intuitive platform that allows farmers to recognize crop diseases through clear descriptions and images.
2. **Suggest Accurate Remedies:**
Offer scientifically recommended treatments and preventive measures for common crop diseases.
3. **Centralize Crop Information:**
Maintain a comprehensive database of major crops, their diseases, and corresponding remedies for quick reference.
4. **Enhance Accessibility:**
Ensure that farmers can access disease and remedy information anytime through a simple, web-based interface.
5. **Enable User Management:**
Provide secure registration and login features for managing user access and personalization.
6. **Support Future Enhancements:**
Establish a framework for future integration of AI or machine learning models to automatically detect crop diseases from images.

4. Proposed Solution:

To overcome the challenges faced by farmers in identifying and managing crop diseases, the proposed solution is a **web-based application named CropCare**. This system serves as a smart platform that provides users with detailed information about various crops, their common diseases, symptoms, and effective remedies.

The application is developed using **Spring Boot** as the backend framework and **HTML/CSS** for the frontend, ensuring a simple and interactive user interface. A **MySQL database** is integrated to store crop data, user information, and disease details efficiently.

Key features of the proposed system include:

- **User Registration and Login Module** – Ensures secure access for users.
- **Crop Information Module** – Displays detailed information about major crops.
- **Disease Detection Module** – Lists disease symptoms and identification tips.

- **Remedy Module** – Suggests preventive measures and treatments for each disease.
- **Upload Module** – Allows users to upload images of affected crops for future AI-based detection.

By providing these features in a single, easy-to-use web application, **CropCare** empowers farmers to detect diseases early, apply suitable remedies, and ultimately improve productivity and reduce losses.

5 .Technology Stack and Decision Rationale

Layer	Technology	Rationale
Frontend	HTML5, CSS3	Provides a responsive and user-friendly interface for easy navigation.
Backend	Spring Boot (Java)	Simplifies backend development with REST APIs and MVC architecture.
Database	MySQL	Offers reliable and structured data management for crops, users, and diseases.
Validation	Spring Boot Validation API	Ensures secure and valid input data during user registration and login.
Build Tool	Maven	Automates project builds and dependency management.
IDE	Eclipse / IntelliJ IDEA	Provides a robust development environment for Java and Spring Boot.
Server	Embedded Tomcat	Handles web requests without external configuration.

These technologies were chosen for their **compatibility, scalability, and performance**, ensuring that CropCare remains efficient and accessible for users across various devices.

6. Database Design

The CropCare database is implemented in **MySQL** and follows a normalized structure for efficient data management.

It includes tables for users, crops, diseases, and remedies.

Database Schema Overview

Table	Purpose
users	Stores user credentials and registration details
crops	Contains information about different crop types
diseases	Lists crop diseases with symptoms and causes
remedies	Stores recommended treatments for diseases

7. System Workflow

The CropCare application follows the **MVC (Model–View–Controller)** architecture.

Frontend (View Layer)

- Built using HTML and CSS for responsive design
- Displays crop, disease, and remedy information
- Allows users to upload crop images

Backend (Controller & Service Layers)

- Handles user requests and business logic
- Interacts with the database to fetch and store data
- Implements validation and authentication

Database (Model Layer)

- Stores user, crop, and disease-related data
- Ensures secure and organized data handling

Workflow Summary

User → Web Interface (HTML)

↓

Spring Controller → Service Layer

↓

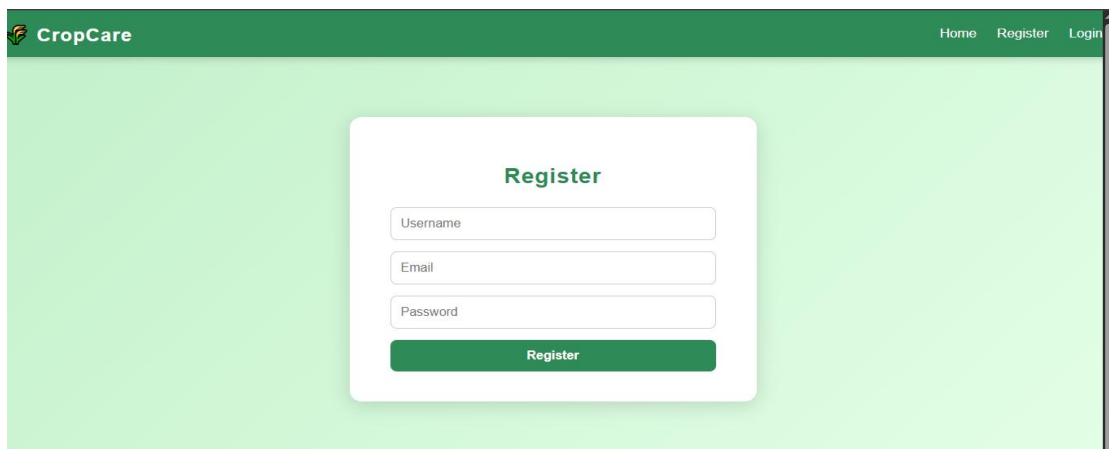
MySQL Database

↓

Response → Display in Browser

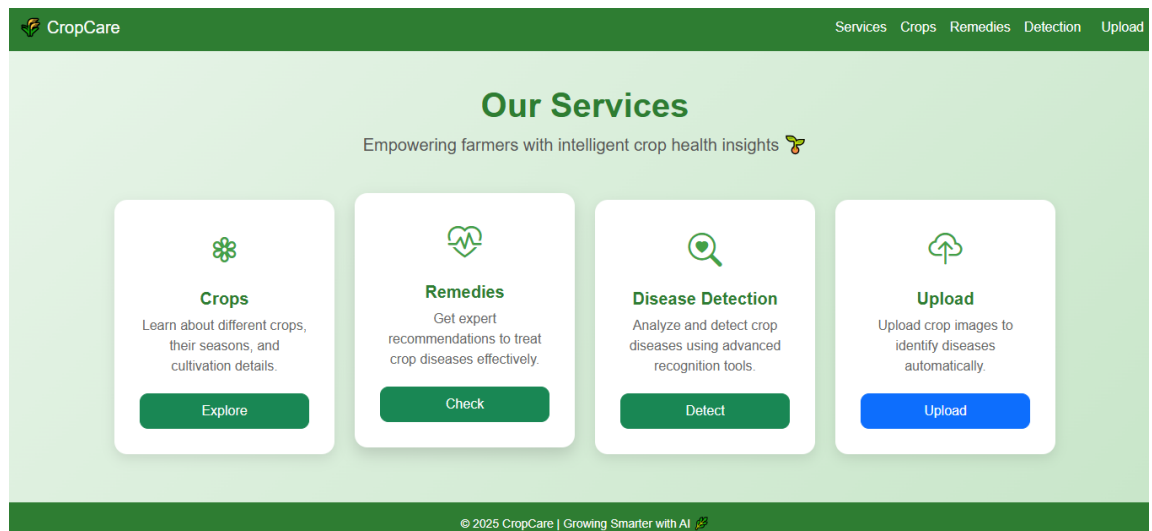
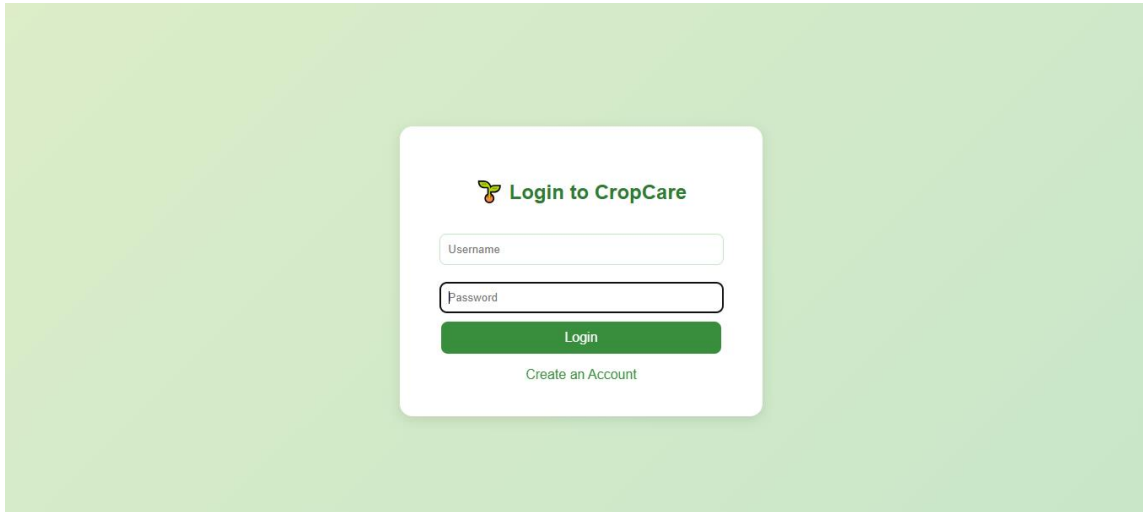
8.Modules Implemented:

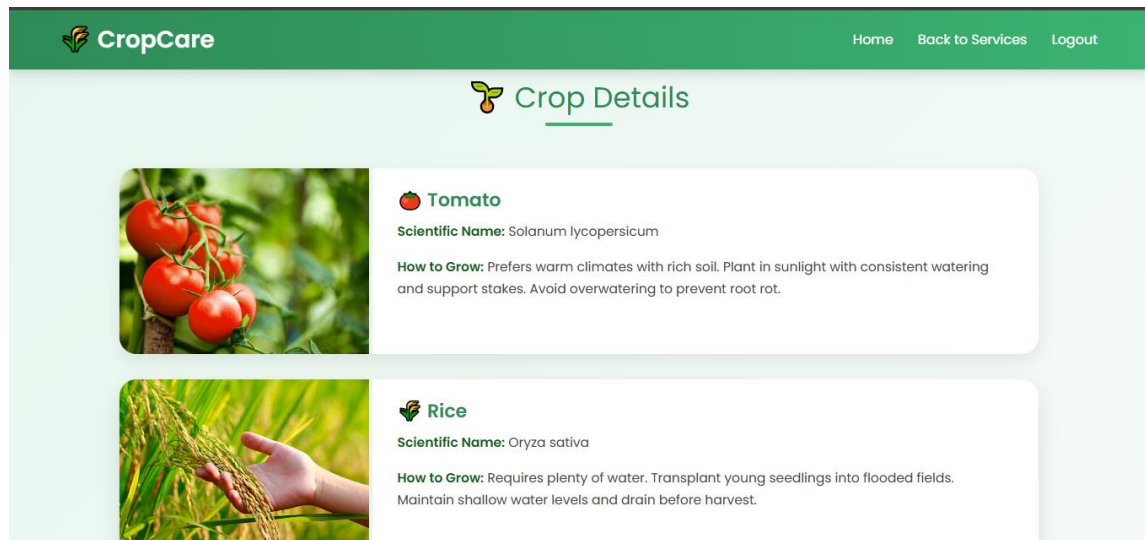
1. **Login & Registration Module** – Enables secure user access and management



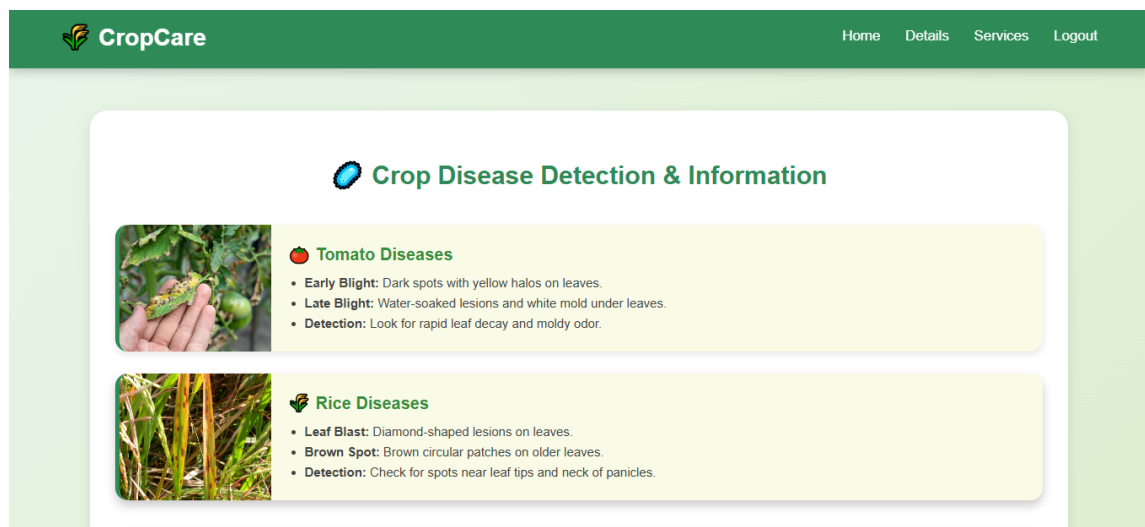
The screenshot displays the CropCare web application interface. At the top, there is a green header bar with the CropCare logo on the left and navigation links for Home, Register, and Login on the right. The main content area has a light green background. In the center, there is a white rounded rectangle containing the 'Register' form. The form is titled 'Register' in green text. It includes three input fields: 'Username', 'Email', and 'Password'. Below these fields is a green 'Register' button.

2. Crop Module – Displays details and characteristics of major crops.

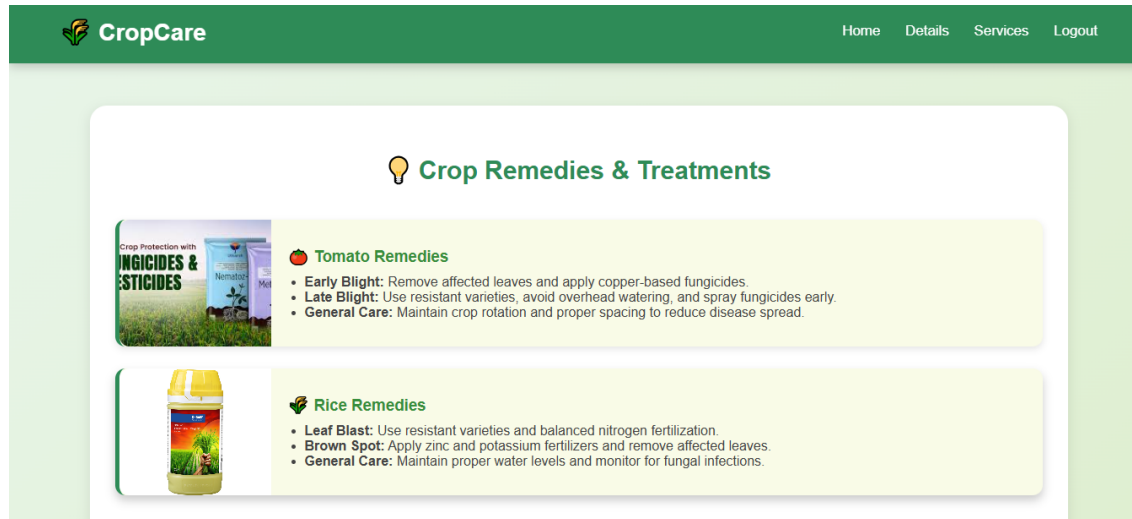




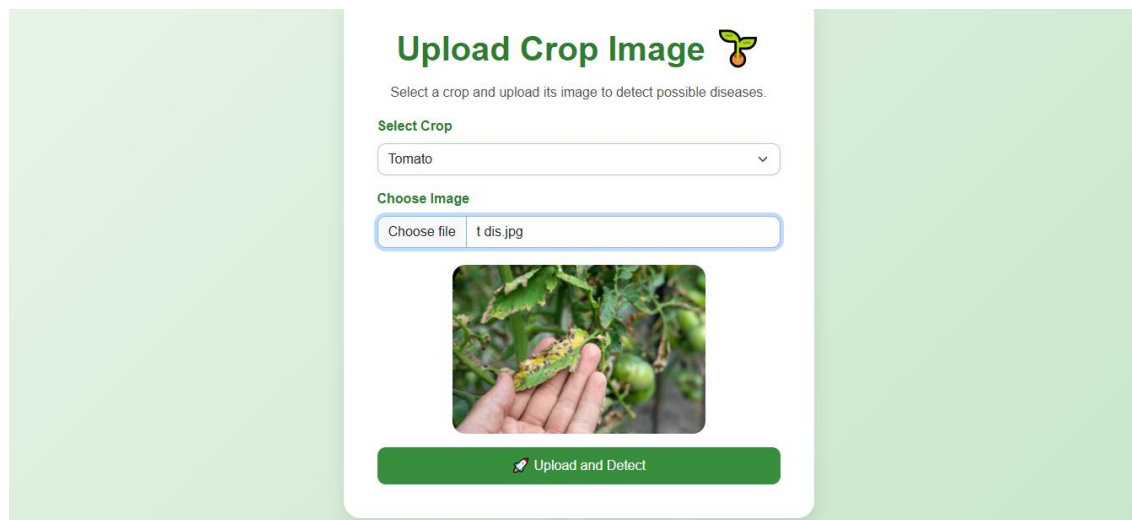
3. **Disease Detection Module** – Lists diseases with symptoms and identification guides.

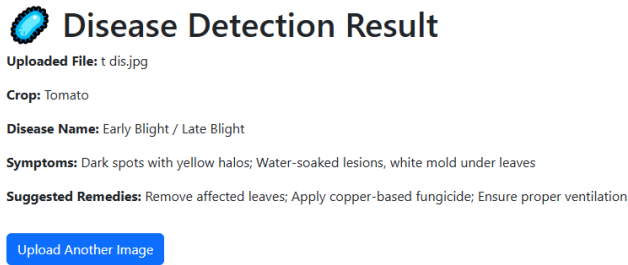


4. Remedies Module – Suggests appropriate treatments and preventive methods.



5. Upload Module – Allows users to upload crop images for potential AI-based disease detection.





9.Future Enhancements:

- Integration of **AI/ML** for automatic disease detection from images.
- Development of a **mobile application** for wider accessibility.
- Multi-language support to reach diverse farming communities.
- Addition of **data analytics** for predicting disease outbreaks.
- Cloud-based storage for larger datasets and scalability.

10.Conclusion

The development of **CropCare** demonstrates how modern web technologies can be used to address real-world agricultural problems.

By providing an easy-to-use platform for disease identification and remedy suggestion, CropCare contributes to improving agricultural efficiency and supporting farmers' livelihoods.

This project also enhanced technical expertise in **Spring Boot**, **database design**, and **web application development**, while emphasizing the importance of technology in promoting **smart and sustainable farming practices**.