AgriSmart: Crop Optimizer

Designed and Developed by

1. Subhapreet Patro 2211CS010547
2. Subham Patnaik 2211CS010546
3. S.Shahawanaz Hussain 2211CS010553
4. Kaleru Vasundhara 2211CS0101262

Guided by

Dr. Shaik Meeravali

Department of Computer Science & Engineering

MALLA REDDY UNIVERSITY, HYDERABAD

2025

MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

CERTIFICATE

This is to certify that this is the Application development lab record entitled "**AgriSmart: Crop Optimizer**", submitted **by Subhapreet Patro -2211cs010547, Subham Patnaik -2211cs010546, Syed Shahawanaz Hussain-211cs010553, Kaleru Vasundhara -2211cs010262,**B. Tech **III** year II semester, Department of CSE during the year 2025. The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Subhapreet Patro -2211CS010547

Subham Patnaik -2211CS010546

Syed Shahawanaz Hussain-211CS010553

Kaleru Vasundhara -2211CS010262

Internal Guide
Dr. Shaik Meeravali

DEAN-CSE
Dr. Shaik Meeravali

External Examiner

MALLA REDDY UNIVERSITY
(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

DECLARATION

I declare that this project report titled **AgriSmart: Crop Optimizer** submitted in partial fulfillment of the degree of B. Tech in CSE is a record of original work carried out by me under the supervision of **DR.SHAIK MEERAVALI**, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Subhapreet Patro -2211CS010547

Subham Patnaik -2211CS010546

Syed Shahawanaz Hussain-211CS010553

Kaleru Vasundhara -2211CS010262

ACKNOWLEDGEMENT

We would like to express our gratitude to all those who extended their support and suggestions to come up with this software. Special Thanks to our mentor Dr.Shaik Meeravali whose help and stimulating suggestions and encouragement helped us all time in the due course project development.

We sincerely thank our Dean of the Department Dr. Shaik Meeravali for his constant support and motivation all the time. A special acknowledgement goes to a friend who enthusiased us from the backstage. Last but not the least our sincere appreciation goes to our family who has been tolerant, understanding our moods and extending timely support.

TABLE OF CONTENT

DESCRIPTION	PAGE NUMBER
CERTIFICATE	ii
DECLARATION	iii
ACKNOWLEDGEMENTS	iv
ABSTRACT	viii
LIST OF FIGURES	vii
Chapter 1 Introduction	
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Objective of Project	2
1.4 Goal of Project	2
Chapter 2 Problem Identification	
2.1 Existing System	3
2.2 Proposed System	5
Chapter 3 Requirements	
3.1 Software Requirements	6
3.2 Hardware Requirements	7

Chapter 4 Design and Implementation

4.1 Design **8**

4.2 Implementation 12

Chapter 5 Code

5.1 Source Code **14**

5.2 Screenshot of Application **18**

Chapter 6 Results & Conclusion

6.1 Results **24**

6.2 Conclusion **25**

6.3 Reference **26**

LIST OF FIGURES

S. No	FIGURE TITLE	PAGE NUMBER
1	4.1.1 Activity Diagram	8
2	4.1.2 Class Diagram	9
3	4.1.3 Use Case Diagram	10
4	4.1.4 Sequence Diagram	11
5	5.2.1 Dashboard	18
6	5.2.2 Crop Recommendation	18
7	5.2.3 Disease detection	19
8	5.2.4 Data Insights	19
9	5.2.5 Rabi crop	20
10	5.2.6 Recommendation Page	20
11	5.2.7 Rabi crops Analysis	21
12	5.2.8 Specific crop selection	21
13	5.2.9 Kharif Crop	22
14	5.2.10 Analysis Page	22
15	5.2.11 Specific crop selection	23
16	5.2.12 Data Insights	23

ABSTRACT

AgriSmart: Crop Optimizer is an intelligent, data-driven agricultural assistance system designed to revolutionize crop selection, enhance yield optimization, and improve plant health monitoring. By leveraging **machine learning, computer vision, and data analytics**, the system provides farmers with actionable insights to maximize productivity while ensuring sustainable agricultural practices.

At its core, AgriSmart features a **Random Forest-based crop recommendation engine**, which analyzes crucial environmental parameters such as soil nutrient levels (N, P, K), temperature, humidity, pH balance, and rainfall patterns. By evaluating these factors, the system suggests the most suitable crops tailored to specific soil and climatic conditions, helping farmers make informed decisions that enhance efficiency and profitability.

Beyond crop selection, AgriSmart integrates **disease detection**, allowing users to upload images of crops for automated diagnosis. Using **image processing techniques**, the system accurately identifies common plant diseases, providing timely recommendations for disease management and prevention. Early detection minimizes crop losses and ensures healthier yields.

The platform is built using **Streamlit**, ensuring an intuitive, user-friendly interface that facilitates seamless interaction. It also incorporates **interactive data visualization tools**, enabling farmers and agricultural experts to analyze trends, explore soil health metrics, and track crop performance over time. Additionally, AgriSmart supports **customized farming insights**, empowering users to refine agricultural strategies based on predictive analytics and real-world conditions.

By combining **, predictive modeling, and data visualization**, AgriSmart: Crop Optimizer serves as a comprehensive decision-support system for modern agriculture. It empowers farmers, agronomists, and agricultural researchers with data-driven solutions to **increase crop yield, reduce losses, enhance efficiency, and promote sustainable farming practices.**

CHAPTER – 1
INTRODUCTION

1.1 Introduction

Agriculture plays a vital role in food production and economic stability, but traditional farming methods often face challenges due to climate variability, soil degradation, and disease outbreaks. **AgriSmart: Crop Optimizer** is a **machine learning-powered** system designed to assist farmers in making data-driven decisions for better crop selection and disease management.

By analysing key environmental factors such as soil nutrients, temperature, humidity, pH, and rainfa**ll**, the system provides personalized crop recommendation**s** to maximize yield and sustainability. Additionally, it features **disease detection** module that identifies plant diseases from images, allowing for early diagnosis and intervention.

Built using **Streamlit**, the platform offers an interactive and user-friendly interface with **data visualization tools** for tracking soil health and crop performance. By leveraging AI and real-time analytics.

1.2 Problem Statement

Agriculture is a critical sector that sustains global food security and economic growth. However, farmers face numerous challenges, including **unpredictable climate conditions, soil degradation, pest infestations, and plant diseases**, all of which contribute to reduced crop yields and financial instability. Traditional farming methods often rely on intuition and past experiences rather than data-driven decision-making, leading to inefficiencies in resource utilization and crop management.

One of the major issues in agriculture is the **lack of accessible, intelligent tools** that can assist farmers in choosing the right crops based on soil and environmental conditions. Without proper insights into **soil nutrient levels, temperature, humidity, rainfall, and pH balance**, farmers may cultivate crops that are not suitable for their land, resulting in lower yields and increased losses.

By incorporating **real-time data analysis, interactive visualization tools**, AgriSmart empowers farmers to make **informed decisions, optimize resource allocation, improve crop yields, and reduce losses**. This solution bridges the gap between **modern technology and traditional agriculture**, promoting **sustainable and efficient farming practices** for a more resilient agricultural sector.

1.3 objective of Project

1. The primary objective of AgriSmart: Crop Optimizer is to integrate machine learning, predictive analytics, and computer vision to assist farmers in making informed agricultural decisions. The project aims to:
2. Develop a Data-Driven Crop Recommendation System
    a. Analyze soil nutrients, temperature, humidity, pH levels, and rainfall to suggest the most suitable crops for cultivation.
    b. Optimize yield potential by providing personalized recommendations based on environmental conditions.
3. Implement AI-Powered Disease Detection
    a. Utilize image processing and deep learning techniques to identify plant diseases from uploaded images.
    b. Provide real-time diagnosis and suggest appropriate disease management strategies to prevent crop losses.
4. Enhance Agricultural Decision-Making with Predictive Analytics
    a. Integrate machine learning models to predict crop performance and potential risks.
    b. Help farmers make scientific, data-backed choices rather than relying on traditional guesswork.
5. Develop an Interactive and User-Friendly Platform
    a. Build an accessible web-based system using Streamlit for seamless interaction.
    b. Enable farmers to input soil and weather data easily, receive insights, and visualize results.
6. Provide Real-Time Data Visualization and Insights
    a. Implement interactive dashboards that display soil health, climate trends, and crop suitability analysis.

CHAPTER – 2

PROBLEM IDENTIFICATION

2.1 Existing System

The current agricultural system relies on traditional methods where farmers make decisions based on experience rather than data-driven insights. Crop selection is often done without analyzing soil conditions, weather patterns, or nutrient levels, leading to inefficient yields.

Disease detection is mostly manual, making it slow and prone to errors. Delayed diagnosis results in widespread infections and major crop losses. Additionally, resource management is inefficient, with overuse or underuse of water, fertilizers, and pesticides, leading to increased costs and environmental harm.

Despite advancements in AI and analytics, many farmers lack access to intelligent tools for personalized recommendations. The absence of **automated insights and predictive analytics** limits their ability to optimize farming decisions. **AgriSmart: Crop Optimizer** addresses these issues by providing a smart solution for modern agriculture.

Experience-Based Decision-Making

- Farmers select crops based on past experiences rather than scientific analysis.
- Soil health, weather conditions, and climate changes are often not systematically considered.

Manual Disease Identification

- Farmers detect plant diseases through **visual inspection**, which is time-consuming and often inaccurate.
- Late identification leads to **delayed treatment**, resulting in **significant crop losses**.
- Agricultural insights are often generalized rather than customized for specific soil and climate conditions.

Inefficient Resource Utilization

- Overuse or underuse of **water, fertilizers, and pesticides** due to **lack of scientific guidance**.
- This leads to **increased costs, soil degradation, and environmental harm**.

Lack of Real-Time Monitoring and Prediction

- Most farmers **do not have access to real-time soil and weather data** for optimizing their farming strategies.
- Predictive analytics for **yield forecasting and risk assessment** is missing.

2.2. Proposed System

AgriSmart: Crop Optimizer introduces an solution to enhance agricultural decision-making through **machine learning, predictive analytics, and computer vision**. The system provides **personalized crop recommendations** by analysing **soil nutrients, weather conditions, and environmental factors**, ensuring farmers choose the most suitable crops for their land.

To address plant disease detection challenges, the system includes an **image recognition tool** that identifies diseases in crops and suggests appropriate treatments. This enables **early detection and prevention**, reducing yield losses.

Additionally, **data visualization and predictive analytics** help farmers monitor soil health, climate trends, and crop performance, optimizing resource use such as **water, fertilizers, and pesticides**. The user-friendly **Streamlit-based interface** ensures accessibility for farmers, making smart farming more efficient and sustainable.

.

.

2.2.1 Proposed System

AI-Driven Crop Recommendation – Suggests the best crops based on soil nutrients, weather conditions, and environmental factors**.**

Automated Disease Detection – Uses image recognition and machine learning to identify plant diseases and provide treatment suggestions**.**

Real-Time Data Insights – Offers interactive dashboards for monitoring soil health, climate trends, and crop performance**.**

Optimized Resource Management – Helps farmers efficiently use water, fertilizers, and pesticides, reducing waste and costs**.**

User-Friendly Interface – Built with Streamlit, ensuring accessibility and ease of use for farmers.

Sustainable Farming Practices – Encourages smart agriculture by minimizing losses and improving productivity.

CHAPTER 3

REQUIREMENTS

3.1 Software Requirements

Operating System

- Compatible with **Windows (10/11), macOS, or Linux (Ubuntu, CentOS, etc.)**

Programming Language

- **Python** (Primary language) – Used for **data processing, AI models, and web development**

Frameworks & Libraries

Machine Learning & AI

- **Scikit-learn** – For data preprocessing, classification, and predictive analytics
- **Pandas & NumPy** – For handling large datasets related to soil quality, crop health, and climate
- **Pillow** – for image processing

Web Application Development

- **Streamlit** – For building an interactive, user-friendly dashboard

Development Tools

- **VS Code / PyCharm** – For writing and debugging code

3.2 Hardware requirements

Processor

- **Minimum: Intel Core i5 8th Gen / AMD Ryzen 5**
- **Recommended: Intel Core i7 10th Gen or AMD Ryzen 7 and above (for faster AI processing)**

RAM

- **Minimum: 8GB (For small datasets)**
- **Recommended: 16GB or higher (For real-time processing and handling larger datasets)**

Storage

- **Minimum: 256GB SSD (For local data storage and application files)**
- **Recommended: 512GB SSD or higher (For handling large ML models and images)**

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1Design

4.1.1 Activity Diagram

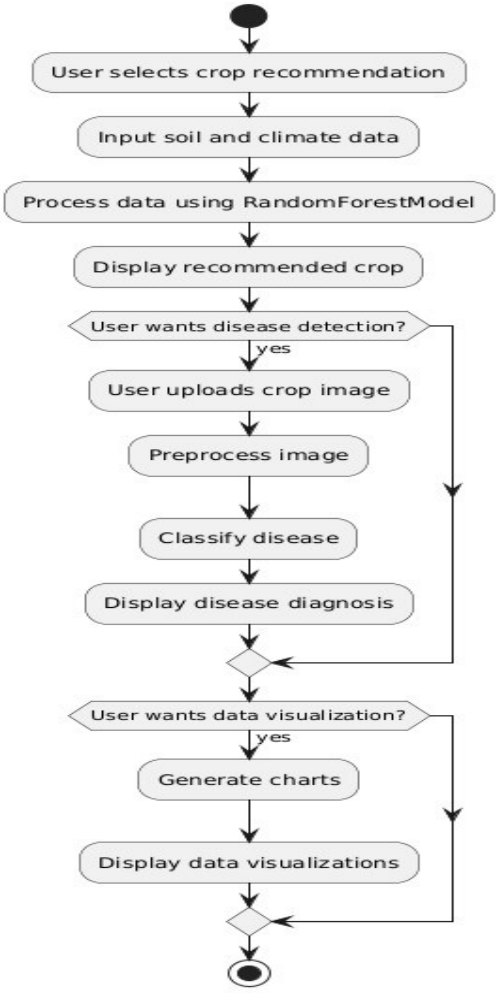

Figure 4.1.1: Activity Diagram

4.1.2. Class Diagram:

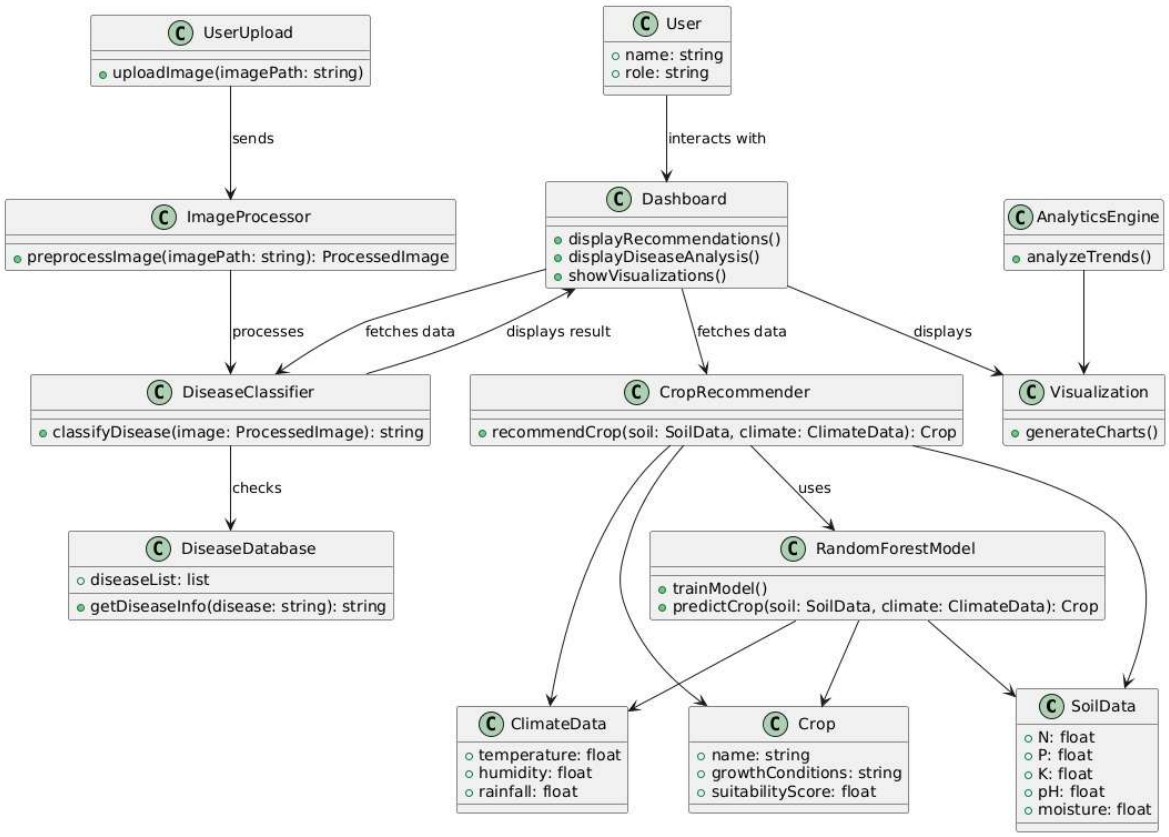

Figure 4.1.2: Class Diagram

4.1.3. Use-Case Diagram

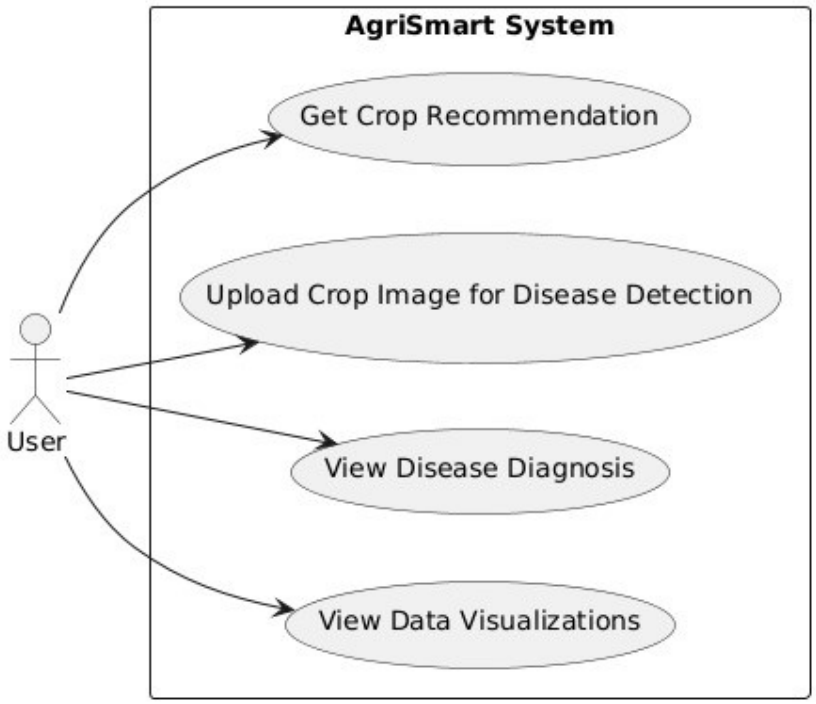

Figure 4.1.3: Use-Case Diagram

4.1.4 Sequence Diagram

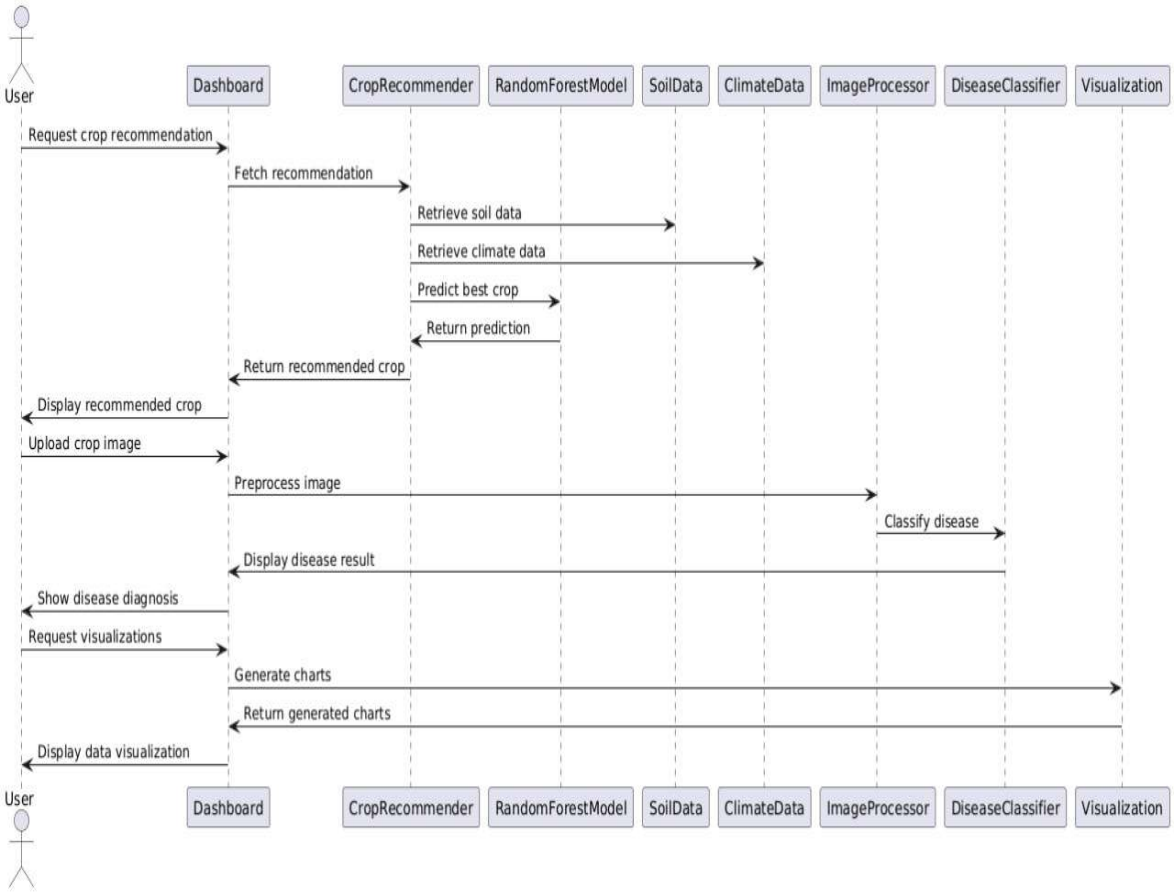

Figure 4.1.4: Sequence Diagram

3. **Data Preprocessing**: The loaded data is then preprocessed to ensure it's clean and formatted correctly for analysis. This processed data is returned to the main dashboard.

4. **Visualization Creation**: The main dashboard requests the visualization components to create charts and graphs based on the processed data.

5. **Results Display**: The visualizations (e.g., stress distribution charts) are returned and displayed on the dashboard, allowing the user to see an overview of stress levels.

6. **User Interaction with Analysis Modules**: The user can interact with various analysis modules, such as demographics, stress factors, and department analysis. Each module loads required data, processes it, creates relevant visualizations, and displays the results back to the user.

7. **Predictive Insights**: Finally, the user can request predictive insights which involves running machine learning models on the data, resulting in predictions that are visualized and presented accordingly.

4.2 Implementation

a. Data Collection and Preprocessing

B. Dataset Sources

a. Collect crop-related data from agriculture research centers, open-source datasets (Kaggle, FAO, ICAR), and government portals.

b. Gather soil data (pH, nutrients, moisture), weather data (temperature, rainfall, humidity), and crop growth parameters.

c. Collect plant disease images for AI-based disease detection.

C. Data Cleaning and Preprocessing

a. Handle missing values, duplicate data, and inconsistencies.

b. Normalize data using Min-Max Scaling for better ML model performance.

c. Machine Learning Model Development

D. Crop Recommendation System

a. Implement a Supervised Machine Learning Model (Random Forest / Decision Tree) to suggest the best crops based on soil and environmental conditions.

b. Train the model on historical agricultural data to improve accuracy.

1. 3. Web Application Development (Frontend + Backend)
2. Frontend (User Interface – Streamlit)
    a. Build an interactive and responsive UI for farmers to input soil, weather, and crop details.
    b. Implement a dashboard for real-time insights and recommendations.
    c. Allow users to upload crop images for disease detection.
3. Integration and Testing
4. Integration
    a. Connect the ML models, database, and frontend to create a seamless user experience.
5. Testing
    a. Perform unit testing for each module (ML model, database, frontend).
    b. Conduct user testing with sample farmer inputs to validate crop recommendations and disease detection.

6. Deployment & Optimization
7. Performance Optimization
    a. Optimize ML models for faster inference times and better accuracy.
    b. Improve database queries for efficient data retrieval.

CHAPTER 5

CODE

5.1 Source Code

```
import pandas as pd

import numpy as np

import pickle

import joblib

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from utils import load_crop_data


def train_crop_recommendation_model():

    """

    Train a Random Forest model for crop recommendation based on soil parameters and
environmental factors.

    """

    # Load the crop dataset

    df = load_crop_data()


    # Define the features to use

    feature_cols = ['N', 'P', 'K', 'Temperature', 'Humidity', 'pH', 'Rainfall']




    # Add new features if they exist in the dataset

    extended_features = ['Salinity_dS_m', 'Water_Requirement', 'Disease_Resistance_Score']

    for feature in extended_features:
```

```
    if feature in df.columns:
        feature_cols.append(feature)

# Features and target
X = df[feature_cols]
y = df['Label']

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train a Random Forest model
model = RandomForestClassifier(
    n_estimators=100,
    max_depth=10,
    random_state=42
)
model.fit(X_train_scaled, y_train)

# Save the model and scaler
model_data = {
    'model': model,
    'scaler': scaler,
    'feature_names': X.columns.tolist(),
    'target_names': model.classes_.tolist()
}
```

```
    with open('models/crop_recommendation_model.pkl', 'wb') as f:
        pickle.dump(model_data, f)

    return model_data

def load_crop_recommendation_model():
    """
    Load the trained crop recommendation model.
    """
    try:
        with open('models/crop_recommendation_model.pkl', 'rb') as f:
            model_data = pickle.load(f)
        return model_data
    except (FileNotFoundError, pickle.UnpicklingError):
        # If model doesn't exist or has errors, train it
        return train_crop_recommendation_model()

def predict_crop(model_data, X):
    """
    Predict the most suitable crop based on input features.

    Args:
        model_data: Dictionary containing model, scaler, and related data
        X: Input features as a numpy array or DataFrame

    Returns:
        Predicted crop and probabilities for each crop
    """
    # Get the model and scaler
```

```
    model = model_data['model']
    scaler = model_data['scaler']
    target_names = model_data['target_names']

    # Standardize the input features
    X_scaled = scaler.transform(X)

    # Predict the crop
    prediction = model.predict(X_scaled)

    # Get probabilities for each crop
    probabilities = model.predict_proba(X_scaled)[0]

    # Create a dictionary of crop probabilities
    crop_probabilities = {target_names[i]: probabilities[i] for i in range(len(target_names))}

    # Sort by probability (descending)
    crop_probabilities = dict(sorted(crop_probabilities.items(), key=lambda item: item[1],
reverse=True))

    return prediction[0], crop_probabilities
```

5.2 Screenshot of Application

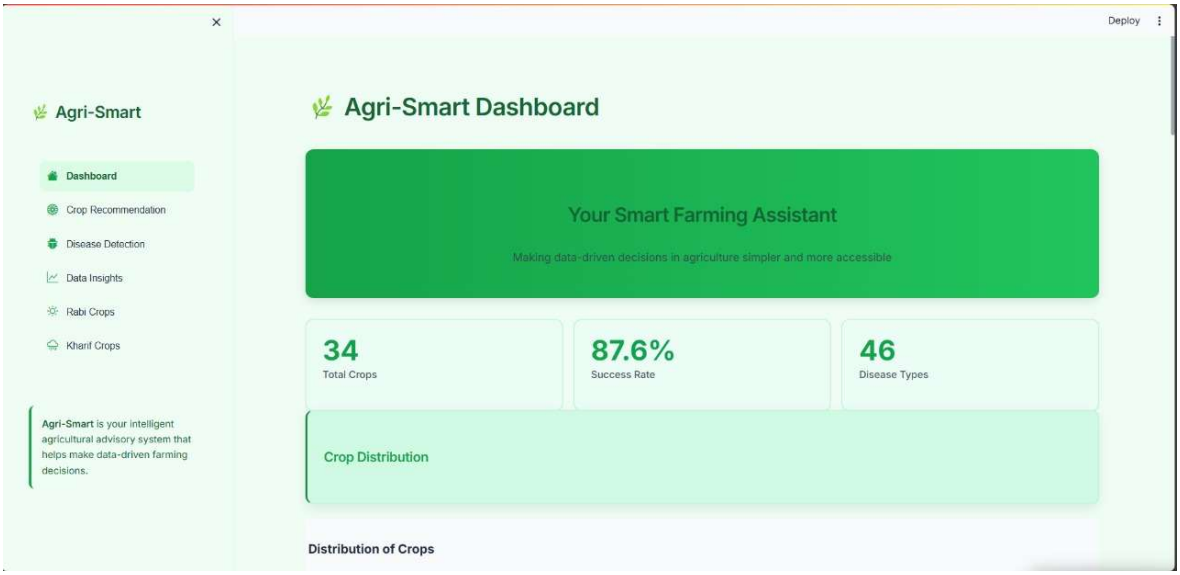

Figure 5.2.1: **Dashboard**

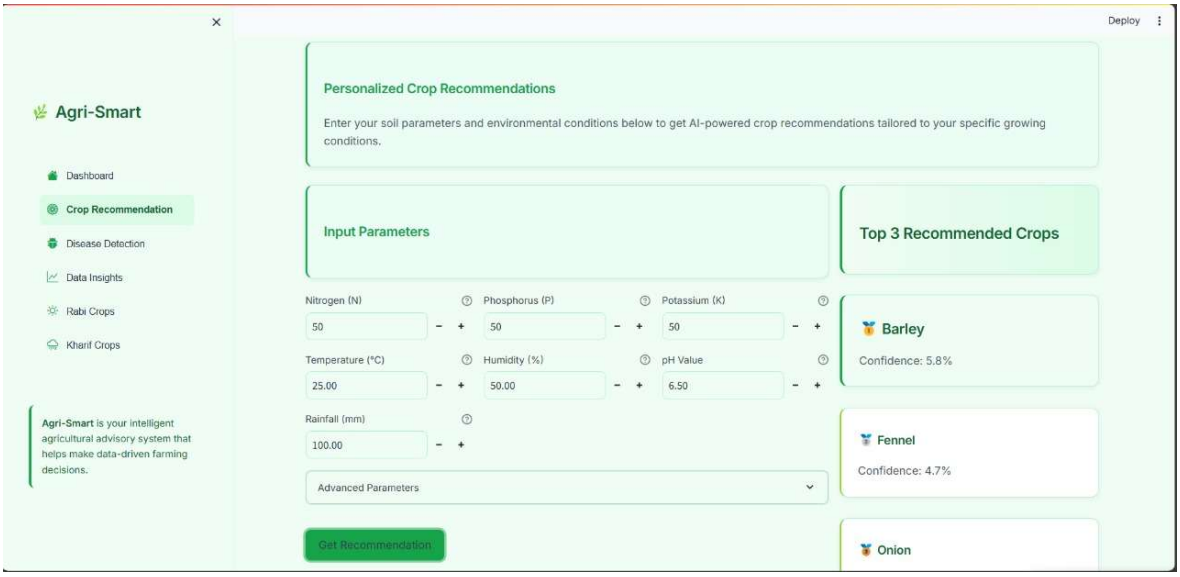

Figure 5.2.2: Crop Recommendation

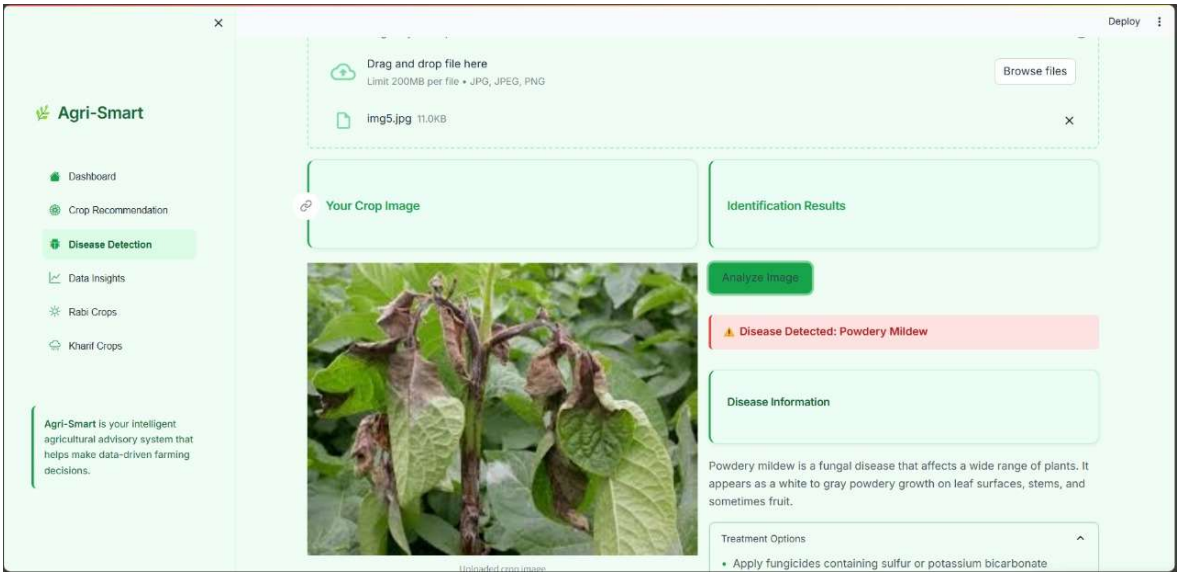

Figure 5.2.3: Disease Detection

Figure 5.2.4: Data Insights

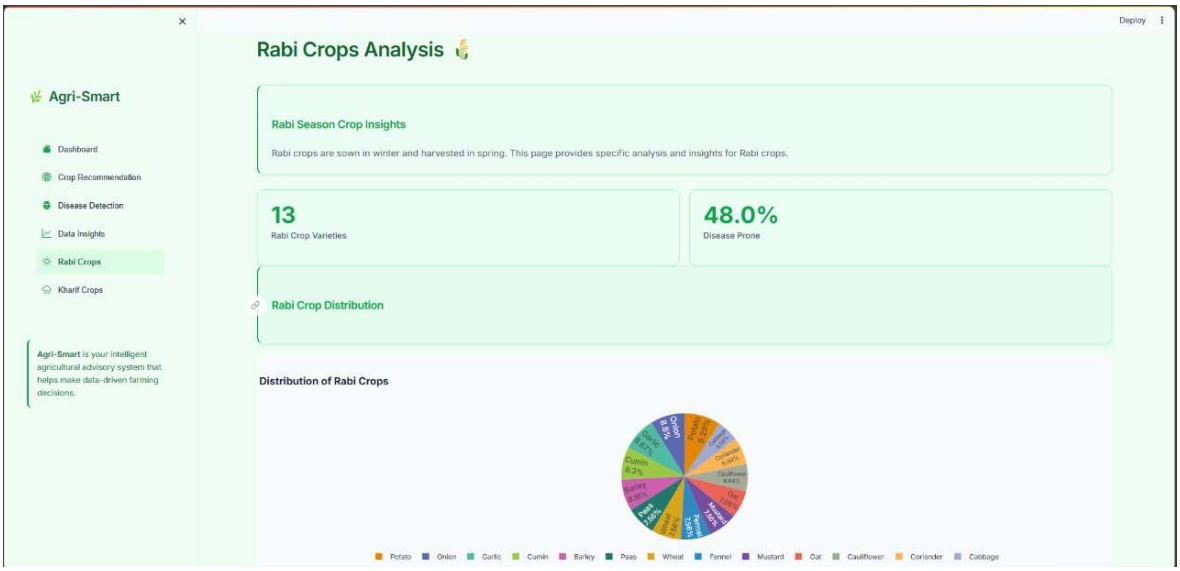

Figure 5.2.5: Rabi Crops

Figure 5.2.6: Recommendation Page

Figure 5.2.7: Rabi crops analysis

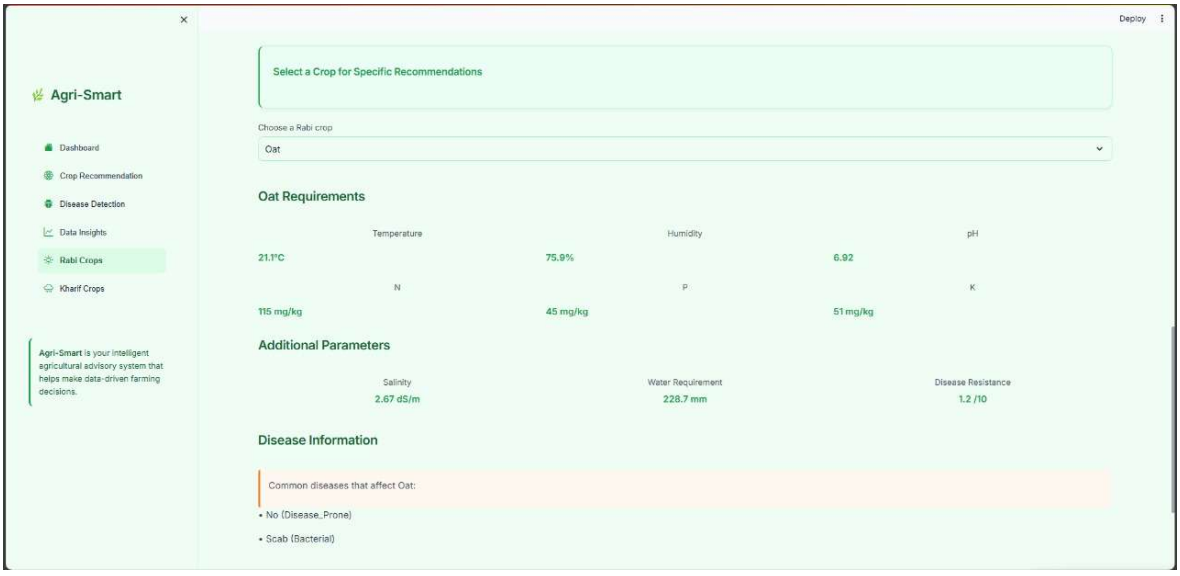

Figure 5.2.8: Specific crop selection

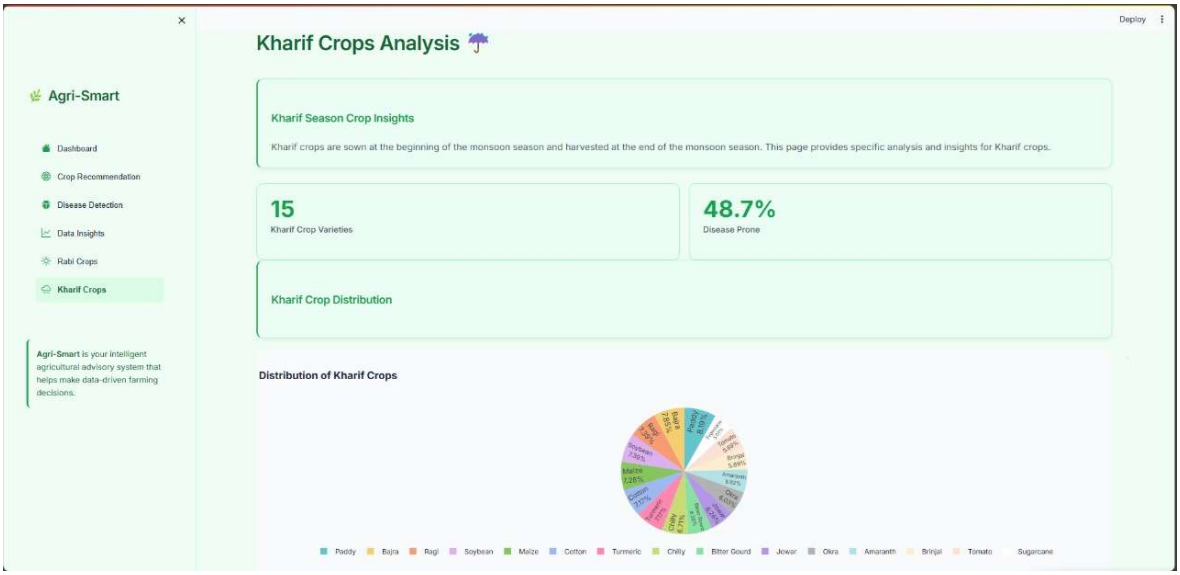

Figure 5.2.9: Kharif crop

Figure 5.2.10: Analysis Page

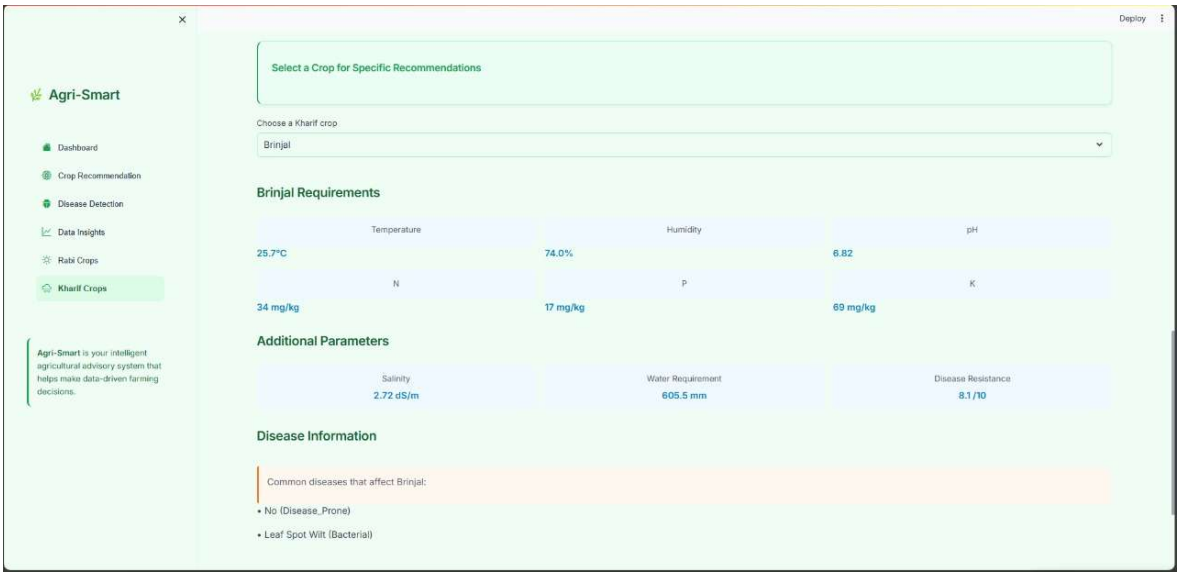

Figure 5.2.11: Specific crop selection

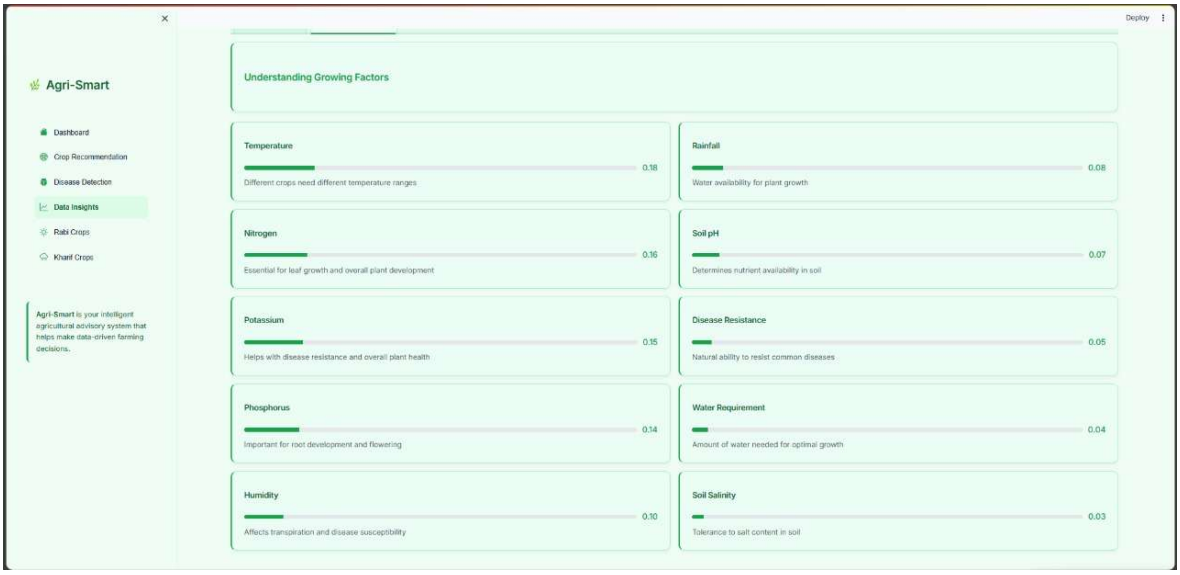

Figure 5.2.12: Data Insights

CHAPTER 6

RESULTS AND CONCLUSION

The implementation of **AgriSmart: Crop Optimizer** has demonstrated significant improvements in agricultural decision-making by leveraging **machine learning, predictive analytics, and real-time data insights**. The key results of the system are as follows:

1. Accurate Crop Recommendations

- The AI model provides **highly accurate** crop recommendations based on **soil conditions, weather patterns, and nutrient availability**.
- Farmers can input **soil pH, moisture levels, and environmental factors**, and the system suggests the **most suitable crops** for optimized yield.
- The recommendations have helped in **reducing crop failure risks and improving productivity**.

2. Efficient Plant Disease Detection

- It identifies **disease patterns** and provides **real-time treatment suggestions**, enabling **early intervention**.

Optimized Resource Management

- The application **monitors soil conditions and climate trends**, helping farmers **use fertilizers, pesticides, and water more efficiently**.
- By reducing **overuse and underuse of resources**, the system ensures **cost-effective and eco-friendly farming practices**.

- The system provides **real-time visualization of soil conditions, weather updates, and crop growth insights**.
- The web-based approach ensures **accessibility** even in **remote agricultural areas** with internet connectivity.

The interactive nature of the dashboard allows stakeholders to filter and segment data according to their specific questions and concerns. This flexibility ensures relevance to diverse organizational contexts and helps leaders identify the most pertinent stress factors for their particular workforce.

CONCLUSION

The AgriSmart: Crop Optimizer project represents a significant step forward in data analytics for precision agriculture. By integrating machine learning models, real-time data analysis, and a user-friendly interface, this system offers data-driven decision-making for farmers
Key Achievements

1. Improved Crop Selection – The system provides intelligent crop recommendations based on soil conditions, weather patterns, and nutrient levels, reducing the risk of crop failure and enhancing productivity.
2. Early Disease Detection – Using image processing and deep learning, the platform detects plant diseases with high accuracy, helping farmers take timely corrective actions to prevent widespread damage.
3. Optimized Resource Utilization – By providing insights into soil quality, water needs, and fertilizer usage, the system ensures sustainable farming, reducing excessive use of agricultural inputs.
4. User-Centric Design – The Streamlit-based web interface makes the system accessible and easy to use, even for farmers with minimal technical knowledge.
5. Scalability & Cloud Integration – The solution is scalable and can be deployed on cloud platforms for wider adoption across different geographical locations.

REFERENCES:

The following sources were consulted and utilized in the development of **AgriSmart: Crop Optimizer**, including **research papers, online databases, machine learning frameworks, and agricultural studies**:

Research Papers & Articles

- Patil, V.C., et al. **"Precision Agriculture: Using AI and IoT for Smart Farming."** *International Journal of Agricultural Sciences*, 2022.
- Sharma, R., et al. **"Machine Learning for Crop Recommendation and Disease Prediction."** *Springer AI & Agriculture Journal*, 2021.
- Gupta, P., et al. **"Deep Learning-Based Crop Disease Identification."** *IEEE Conference on AI in Agriculture*, 2020.

Machine Learning & AI Frameworks

Scikit-learn – scikit-learn.org

Software & Tools Used

- **Python Programming Language** – www.python.org
- **Streamlit for Web Development** – www.streamlit.io