Tribhuvan University

Academia International College

Final Year Project Report

On

Crop Recommendation System

[CSC 412]

Under the supervision of

"Mr. Bishwas Mathema"

Submitted by
Prabisha Bajracharya (T.U. Exam Roll No. 26501/077)

Sagar Chaudhary (T.U. Exam Roll No. 26512/077)

Rojan Chitrakar (T.U. Exam Roll No. 26508/077)

Submitted to

Department of Computer Science and Information Technology

Academia International College

Institute of Science and Technology

Tribhuvan University

January, 2025

Tribhuvan University

Academia International College

Final Year Project Report

On

Crop Recommendation System

[CSC 412]

A final year project submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Information Technology awarded by Tribhuvan University

Submitted by

Prabisha Bajracharya (T.U. Exam Roll No. 26501/077)

Sagar Chaudhary (T.U. Exam Roll No. 26512/077)

Rojan Chitrakar (T.U. Exam Roll No. 26508/077)

Submitted to

Department of Computer Science and Information Technology

Academia International College

Institute of Science and Technology

Tribhuvan University

January, 2025

Tribhuvan University

Institute of Science and Technology

Academia International College

Department of Computer Science and Information Technology

Email: mail@academiacollege.edu.np

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by Prabisha Bajracharya (26501/077), Sagar Chaudhary (26512/077), Rojan Chitrakar (26508/077) entitled "Crop Recommendation System" be accepted as fulfilling in partial requirements for the degree of Bachelors of Science in Computer Science and Information Technology. In my best knowledge, this is an original work in Computer Science and Information Technology

……

Mr. Bishwas Mathema

Project Supervisor

HOD/Program Coordinator

Department of Computer Science and Information Technology

Academia International College

Gwarko, Lalitpur

Tribhuvan University

Department of Computer Science and Information Technology

Academia International College

Certificate of Approval

This is to certify that this project prepared by Prabisha Bajracharya, Sagar Chaudhary, Rojan Chitrakar entitled "Crop Recommendation System" in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…… Mr. Bishwas Mathema Supervisor Academia Int'l College	…… Mr. Bishwas Mathema HOD/Program Coordinator Academia Int'l College
…….. External Examiner Academic designation	……..……. Internal Examiner Central Department of CSIT Tribhuvan University

Acknowledgement

We extend our heartfelt gratitude to Academia International College for providing us with the opportunity and resources to undertake this project as part of our curriculum. The supportive environment and facilities played a crucial role in enabling us to complete the project efficiently.

Special thanks to our supervisor, **Mr. Bishwas Mathema** (HOD/Program Coordinator, Academia International College), for his consistent guidance, support, and feedback throughout the report's creation. We are generously obligated to him for providing this excellent opportunity to expand our knowledge. It helped us a lot to realize what we studied for.

We sincerely thank all the individuals, families, friends, colleagues, and teachers who supported and guided us throughout this project. Your valuable insights and feedback have been instrumental in helping us complete this project successfully within the limited time frame, making this a rewarding and enriching experience.

Thank You,

Prabisha Bajracharya (T.U. Exam Roll No. 26501/077)

Sagar Chaudhary (T.U. Exam Roll No. 26512/077)

Rojan Chitrakar (T.U. Exam Roll No. 26508/077)

Abstract

The "Crop Recommendation System" is a web-based application that assists farmers in determining the best crops for their farms based on soil and environmental factors. Using the Random Forest algorithm, the system evaluates input data such as soil type, pH level, and weather characteristics to select crops that maximize production and productivity.

The system has user and administrative interfaces, secure authentication, and requires internet connectivity to work. The major goal is to give farmers with accurate and timely advice that will improve decision-making and reduce agricultural hazards. This technology reduces human error in crop selection, assures effective data management, and gives user-friendly insights. The project, built with Python, Flask, and SQL database, intends to bridge the gap between technology and agriculture by empowering farmers and contributing to sustainable farming methods.

Keywords: crop recommendation system, Random Forest, Flask, Python, MySQL, agriculture, soil analysis

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DFD	Data Flow Diagram
ER	Entity Relationship
HTML	Hyper Markup Language
IDE	Integrated Development Environment
SQL	Structured Query Language
VS	Code Visual Studio Code

Chapter 1: Introduction

1.1 Introduction

Agriculture is crucial to the world economy, with millions of farmers relying on crops for a living. However, one of the most difficult issues farmers have been deciding on the best crop for their area, taking into account varied soil conditions, climate, and environmental factors. To address this issue, we provide a web-based Crop Recommendation System based on the Random Forest algorithm, designed to assist farmers in making informed crop-selection decisions.

The system uses the Random Forest algorithm, to anticipate the best crops based on important criteria such as soil type, temperature, rainfall, humidity, and ph. By examining past data, the model can make accurate crop recommendations, increasing agricultural output and lowering crop failure risks.

The system, which is accessible via a user-friendly online interface developed using Flask, HTML, and Bootstrap, allows farmers to simply input crucial parameters such as soil qualities and rainfall, as well as obtain specific crop suggestions. This dynamic system is intended to be intuitive, delivering rapid feedback and empowering farmers to make data-driven decisions that boost productivity and sustainability.

The Crop Recommendation System is a modern approach that blends technology and agriculture, allowing farmers to maximize production while adjusting to changing environmental conditions. With this technology, we hope to bridge the gap between ancient farming practices and modern data science, resulting in smarter, more sustainable agriculture.

1.2 Problem Statement

Farmers frequently struggle to select the best crops for their property due to differences in soil conditions, climate, and other environmental factors. This leads to low crop yields, wasteful resource use, and greater economic risk. There is a need for an efficient, data-driven solution that can assist farmers in making informed crop selection decisions, assuring maximum production and sustainability. The Crop Recommendation System intends to

address this issue by offering individualized crop recommendations based on important criteria such as soil type, temperature, rainfall, and pH levels, allowing farmers to make data-driven decisions for improved agricultural results.

Here are some ideas for the issue statement of your Crop Recommendation System:

- Crop Selection Difficulty: Farmers face challenges in selecting the ideal crop for their property owing to a variety of factors such as soil characteristics, weather, and climate.
- Impact of Poor Crop Choices: Incorrect crop selection leads to reduced yields, poor resource utilization, and financial losses for farmers.
- Environmental Variability: Soil quality, temperature, humidity, and rainfall patterns differ greatly among regions, making it difficult for farmers to anticipate which crops would grow.

1.3 Objective

The objective of the Crop Recommendation System is to provide farmers with accurate, data-driven crop suggestions based on environmental factors such as soil type, temperature, and rainfall. By leveraging the Random Forest algorithm, the system aims to help farmers optimize crop selection, improve yield, and enhance sustainability, ensuring more informed decision-making in agriculture.

1.4 Scope and Limitations

1.4.1 Scope

The scope of Crop Recommendation System include:

- Personalized crop recommendations: Offers customized crop recommendations based on specific parameters such as soil type, temperature, and rainfall for various regions.
- Data-Driven Decision Making: Helps farmers make informed decisions by analyzing environmental data using machine learning algorithms.
- User-Friendly Interface: Available via a simple online interface, making it simple for farmers with little technical understanding to utilize.

- Adaptability: The system can be upgraded with fresh data or models to improve accuracy and responsiveness to changing environmental circumstances.

1.4.2 Limitations

Limitations of the Crop Recommendation System are:

- No Real-Time Updates: Lacks real-time changes in weather, pests, or other factors.
- No Integration with Farming Equipment: Doesn't interact with IoT or smart farming tools for real-time monitoring.
- Farmer Awareness: Many Nepali farmers may lack knowledge of input parameters, reducing effectiveness.
- Internet Dependence: Requires internet, which can be unavailable in remote areas.

1.5 Methodology

The Agile methodology is a modern project management and software development technique that emphasizes flexibility, cooperation, and a customer-centric perspective. It is commonly used by top companies such as Facebook, Google, and Amazon. Agile takes an iterative and incremental strategy, focusing on the rapid delivery of functional products [7].

We used Agile methodology to develop our Crop Recommendation System, which allowed for flexibility and iterative progress throughout the project. Regular supervision and input from supervisors appear to be the most effective way for project improvement. This strategy enabled us to work iteratively and adjust to changes. Incorporate regular input to achieve high-quality results.

We began by consulting with our supervisor to define the project requirements, breaking them down into smaller, manageable tasks and prioritizing them accordingly. The project was organized into distinct tasks, each with specific objectives clearly outlined. The backend included connecting the user registration/login and Random Forest models with our Flask web app. The model was trained on 2200 datasets and continuously improved throughout the training process.

Frequent meetings with our supervisor provided valuable feedback, enabling us to refine the system. These discussions focused on improving aspects such as user-friendliness and model

accuracy, which were successfully incorporated into the system. Once all components were integrated, we conducted user testing and subsequently deployed the system.

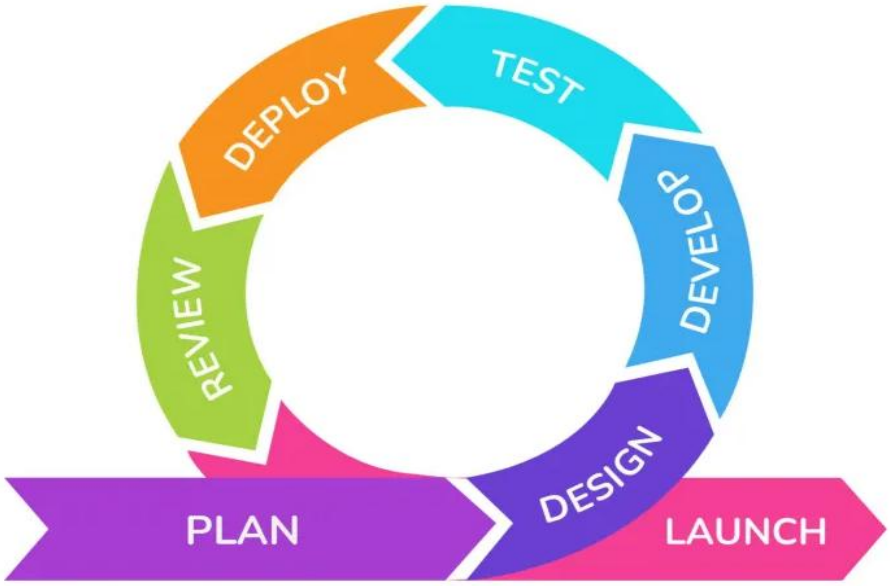


Figure 1.1 Agile Methodology for Software Development

1.6 Report Organization

This report is divided into six main chapters, each dedicated to a specific aspect of the project, providing a clear and organized presentation of information. The chapters are outlined as follows:

Chapter 1: Introduction

The first chapter provides an overview of the project, including its background, the problem being addressed, the objectives, and the scope and limitations of the study. It also outlines the methodology adopted during development and offers a roadmap of how the report is organized.

Chapter 2: Background Study and Literature Review

This chapter lays the theoretical groundwork for the project, explaining essential concepts, terminologies, and theories. It also presents a detailed review of similar projects and related

studies by other researchers, establishing a link between existing knowledge and the project's objectives.

Chapter 3: System Analysis

The third chapter focuses on into the analysis phase of the system development process. It outlines the functional and non-functional requirements and includes a feasibility study addressing technical, operational, economic, and scheduling aspects. Additionally, models like ER diagrams and DFDs are utilized to analyze the data and system processes.

Chapter 4: System Design

The design phase is covered in this chapter, focusing on database design (converting ER diagrams into relational models and ensuring normalization), the layout of forms and reports, and the creation of user-friendly interfaces. The algorithms powering the system's functionality are also detailed here.

Chapter 5: Implementation and Testing

This chapter explains how the system was developed using various tools and technologies. It describes the implementation of different modules, highlighting the methods, classes, and processes used. It also details the testing process, including unit and system testing, along with an analysis of the results to confirm the system's expected performance.

Chapter 6: Conclusion and Future Recommendations

The final chapter summarizes the project's key achievements and findings. It reflects on the objectives and evaluates the extent of their fulfillment. Additionally, it provides recommendations and suggestions for future improvements, aiming to enhance the project further based on the outcomes.

Chapter 2: Background and Literature Review

2.1 Background

Nepal remains a primarily agrarian country, with agriculture employing 62% of the population and accounting for approximately 23% of the GDP in 2022. However, the sector confronts various obstacles, including insufficient government investment, falling productivity, and little commercialization. Despite Nepal's inherent agricultural diversity and fertile regions, the country's reliance on imports for staple goods such as rice and vegetables has increased its trade deficit significantly [5].

A crop recommendation system suited to Nepal's distinct agro-climatic zones presents a possible option. The system makes data-driven recommendations to farmers to assist them choose the best crops by monitoring soil fertility, weather conditions, and parameters such as NPK levels, rainfall, and temperature. This not only increases production but also decreases reliance on imports, which helps to ensure food security and economic stability. Furthermore, such systems encourage efficient resource utilization and sustainable activities to prevent climate change.

In this project, we will use machine learning to select appropriate crops depending on soil fertility, weather conditions, and other environmental data. We will use the Random Forest algorithm for classification to predict the best crops for a specific region. The system will be simply accessible and user-friendly, allowing farmers to make educated decisions and increase agricultural yields with more accuracy.

2.2 Literature Review

The growing demand for sustainable agriculture practices and resource efficiency has led to a considerable increase in interest in the development of machine learning-based crop recommendation systems in recent years. In order to forecast the best crops based on variables like soil type, pH levels, and weather patterns, researchers have investigated a variety of machine-learning models, including as Decision Trees, Random Forests, and K-Nearest Neighbors.

In October 2020, Pradeepa Bandara, Thilini Weerasooriya, Ruchirawya T.H, W.J.M. Nanayakkara, Dimantha M.A.C, Pabasara M.G.P. proposed a system which used an Arduino board to collect data on several parameters like pH, moisture, humidity, temperature, erosion to predict suitable crop. The dataset contained around five different crops. The crop prediction was be done by Native Bayes Classifier, Support Vector Machine (SVM). By applying Native Bayes algorithm acquired higher productivity and accuracy of around 96% [1].

In April 2022, Lakshman Kumar Sreu and Sai Maanas Gandham have implemented the Decision tree and Random Forest classifier for the crop recommendation system. In 2022 Ranjith Macharla, Dr. K Shirisha developed a Crop Recommendation System using Random Forest Classifier algorithm. They first compared the accuracy of other models like of Decision Tree Classifier in which they got the accuracy of 90%, KNN Classifier in which they got an average accuracy 97%, Random Forest Classifier in which they got an accuracy of 98.8%. 80% of dataset they used for the training the model and 20% of dataset they used for training the model. Among all the models, Random Forest Classifier got the highest accuracy so, they implemented their system using Random Forest Classifier model [2].

They used the dataset from the Kaggle and the Indian government websites. In March 2023, G. Buvaanyaa and Dr. S. Radhimeenakshi published research paper about Crop Recommendation System Using Random Forest Algorithm. Out of 100% samples, they trained 75% samples, and 25% samples used for the testing and found that the average accuracy of crop recommendation system was 91.99% [4].

In February 2024, researchers Monisha H M Dhanush Jogi K Adarsha Dalavai S M Manvith Dalli of B.M.S College of Bangalore, India published a research paper on A Web Based Crop Recommendation System Using Various Machine Learning Algorithms. They tested classifier algorithms like Logistic Regression, Decision Tree, Random Forest and XGBoost and found that among all the classifiers, highest accuracy was of the Random Forest Classifier which was 98.86%. Also, they concluded to use the Random Forest because overfitting will not happen easily and Random Forest gives the accurate results [3].

Chapter 3: System Analysis

3.1 System Analysis

The crop suggestion system is designed to help users choose the most appropriate crops for their farms depending on environmental and soil conditions. Built with Flask, the system is powered by a Random Forest algorithm that was trained on a dataset of 2,200 samples. This extensive dataset covers parameters such as soil qualities (NPK levels), temperature, rainfall, and other information, ensuring accurate and dependable recommendations. The system's user-friendly design and easy access make it an effective tool for encouraging data-driven farming practices and increasing production.

3.1.1 Requirement Analysis

The system requirement can be functional requirements and non-functional requirements.

i. Functional requirements

The system takes into account the following functional requirements:

- User Registration and Login: Users can register and log in to access personalized features and admins can manage user accounts.
- Crop recommendation: Accepts input factors such as NPK concentrations, rainfall, temperature, and pH and uses the trained Random Forest model to recommend the best crops.
- Dashboard: Provides users with a dashboard to view their information and Admins can view their information and user information and user list.
- Contact and Feedback: Includes a contact form for users to provide feedback or report issues.

ii. Non-functional requirements

The system takes into account the following non-functional requirements:

- Performance: The system should respond to user queries and provide recommendations within few seconds.
- Multilingual Support: The system supports multiple languages for accessibility across diverse users.

- Scalability: The system should be capable of handling increased data volumes and user loads while maintaining performance.

3.1.2 Feasibility Analysis

A feasibility study is an evaluation and analysis procedure meant to determine a project or proposal's functioning, practicality, and chance of success. The process includes assessing several factors, such as operational, legal, technical, financial, and scheduling aspects, to determine whether the project can be successfully completed within the allocated budget, time, and resources.

i. Technical Feasibility

From a technical perspective, the idea is possible and compatible with both hardware and software available today. Since the Crop Recommendation System depends on widely recognized technology and scalable frameworks, it is possible in theory. For precise crop predictions, the system makes use of the Random Forest algorithm, a tried-and-true machine learning method. Python is used in its implementation, making use of model-building packages such as Scikit-learn, which are suitable for examining agricultural data. Flask, HTML, and Bootstrap were used in the making of the web interface to provide a scalable, responsive, and lightweight solution that works on every type of device.

The following lists the technical specifications for this project:

- A laptop with at least 8GB RAM, SSD (at least 128GB) and GPU.
- High speed internet

Over 90 percent of today's personal PCs can run this application, which has low hardware and software requirements.

ii. Operational Feasibility

There aren't many system requirements or human resources needed to finish this project. In addition to the system's user-friendly design, farmers with no background in technology can

enter data using an internet browser. A responsive and user-friendly design that works on PCs and mobile phones is ensured by the use of Flask, HTML, and Bootstrap. This method is easy for farmers to include into their regular decision-making procedures. Adoption resistance is decreased by inputs like pH, rainfall, and soil type that match data farmers already record.

There are currently three developers working on the project, which is enough for the project's needs. The goal of this project is to create a crop recommendation system using the Random Forest Classifier Algorithm as its base.

iii. Economic Feasibility

Although it makes use of freely available IDEs (like Visual Studio Code) and open-source tools (like Python and Jupyter Notebook), the project is affordable. It is an effective choice for this project because many websites provide free website hosting as long as a storage limit is met.

The method provides long-term financial benefits by improving crop selection, which raises agricultural yields, decreases resource waste, and lowers farmers economic risks. The possible financial benefits from increased productivity and sustainability outweigh the initial expenses which include design, execution, and user training. Plus, the advantages of the project easily overcome the drawbacks. The project is therefore currently economically feasible.

iv. Schedule

Scheduling feasibility involves estimating the project's completion time. After all of these areas are considered, the feasibility analysis helps identify any obstacles the proposed project may face, including: Internal Project Constraints: Technical, Technology, Budget, and Resources.

Table 3.1 Project Schedule Table

Task	Start Date	End Date	Duration
Project Initiation	3-Aug	17-Aug	14
Data Collection and Preparation	18-Aug	1-Sep	14
Model Development and Preparation	2-Sep	23-Sep	21
System Design and Development	24-Sep	14-Oct	20
Testing and QA	15-Oct	22-Oct	7
Deployment	23-Oct	30-Oct	7
Documentation	18-Aug	13-Nov	87

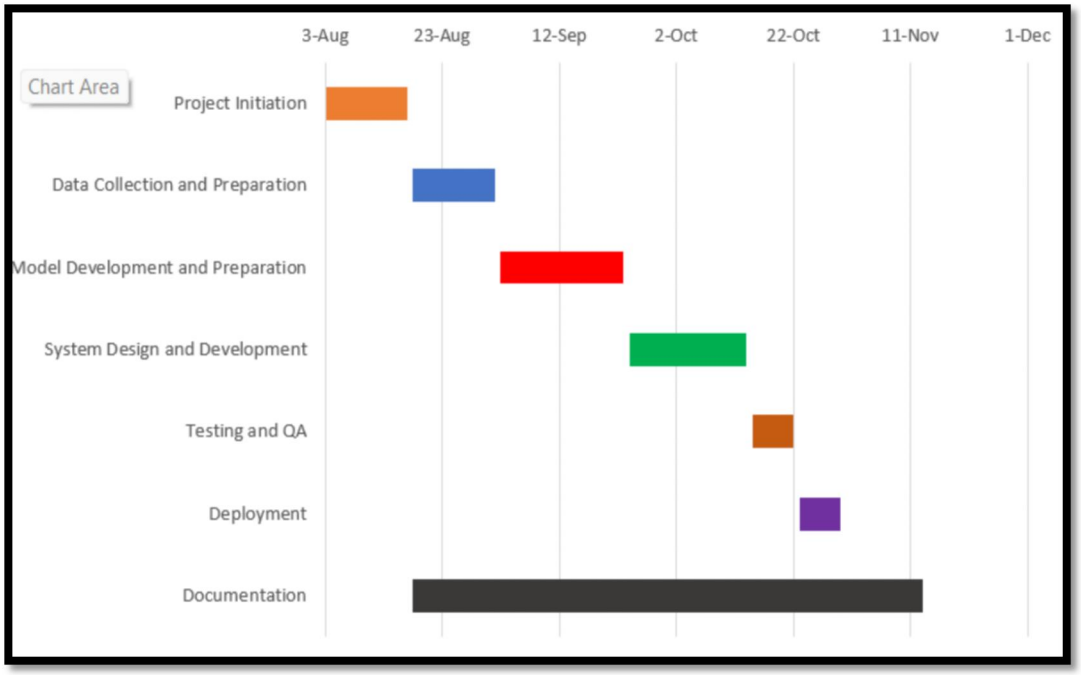


Figure 3.1 Project Gantt Chart

3.1.3 Data Modeling using ER Diagrams

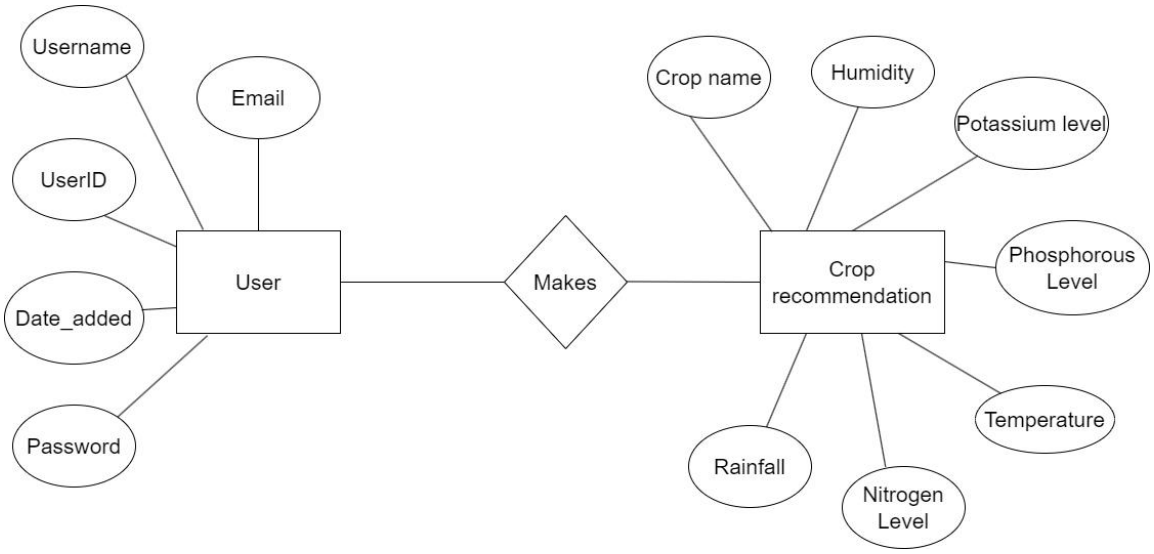


Figure 3.2 ER diagram of Crop Recommendation System

3.1.4 Process Modeling using DFD

The processes such as input of soil data, crop recommendation, and the flow of results between users and the system are illustrated below in the form of a Data Flow Diagram (DFD).

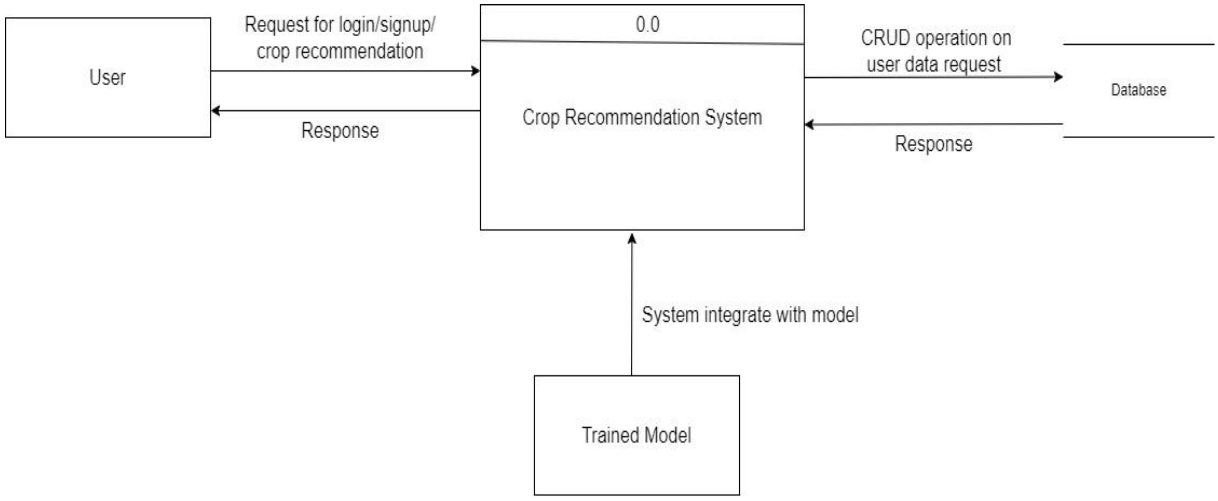


Figure 3.3 Level-0 DFD (Context Diagram) of Crop Recommendation System

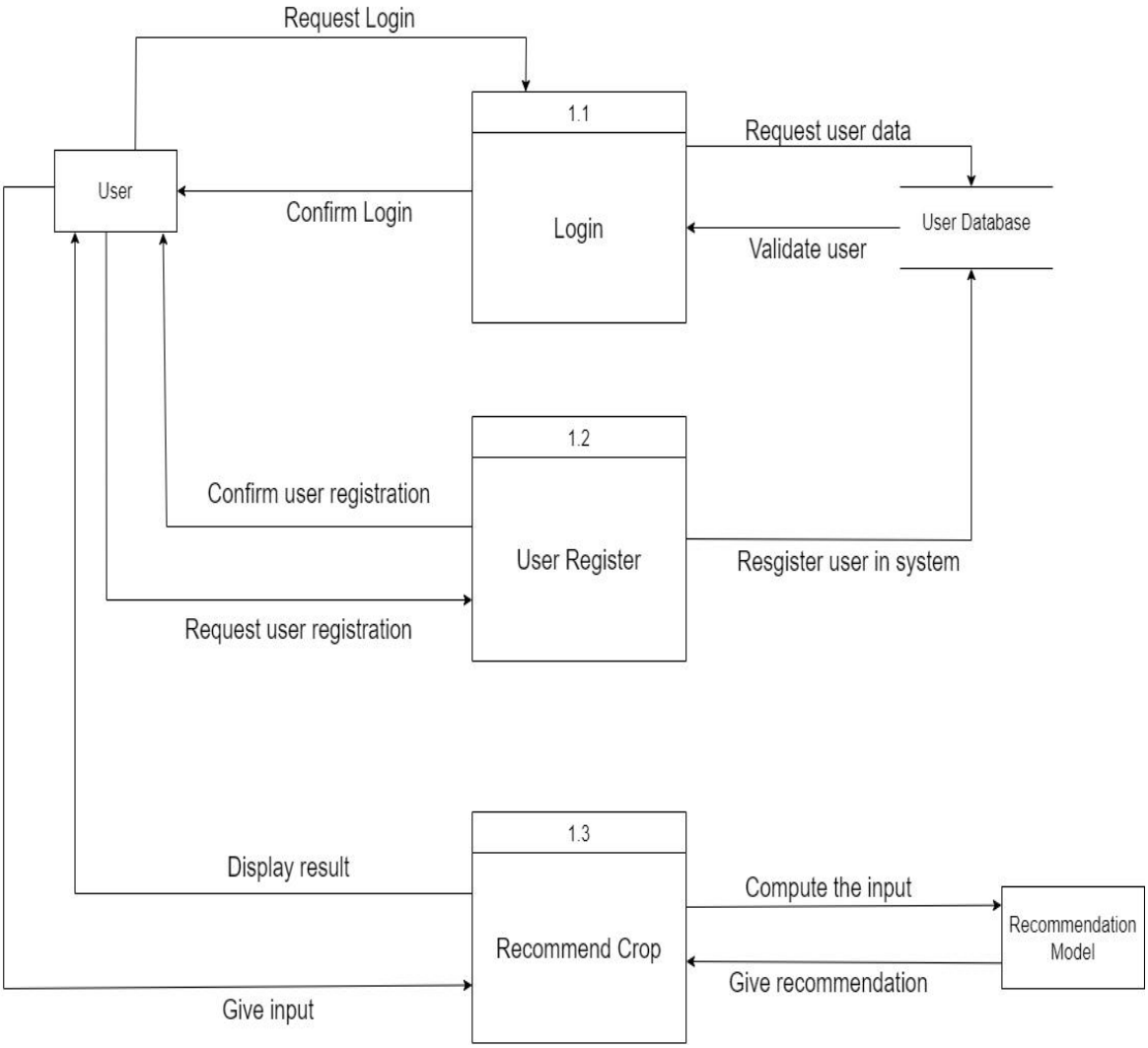


Figure 3.4 Level-1 DFD of Crop Recommendation System

Chapter 4: System Design

4.1 Design

The Crop Recommendation System (CRS) is designed to simplify the process of selecting suitable crops for farming. Our focus is on creating a user-friendly system that leverages smart recommendations to provide accurate and timely information, making it accessible and beneficial for all users.

4.1.1 Architectural Design

The Crop Recommendation System (CRS) has a three-tier design that follows the client-server model. The presentation layer provides the client with an easy-to-use interface, while the application layer guarantees that crop recommendations are accurate based on user input. The data layer securely manages and retrieves critical data, such as soil and environmental factors, to support the recommendation process. Together, these layers operate smoothly to provide quick, precise answers and useful insights in an easy-to-use system.

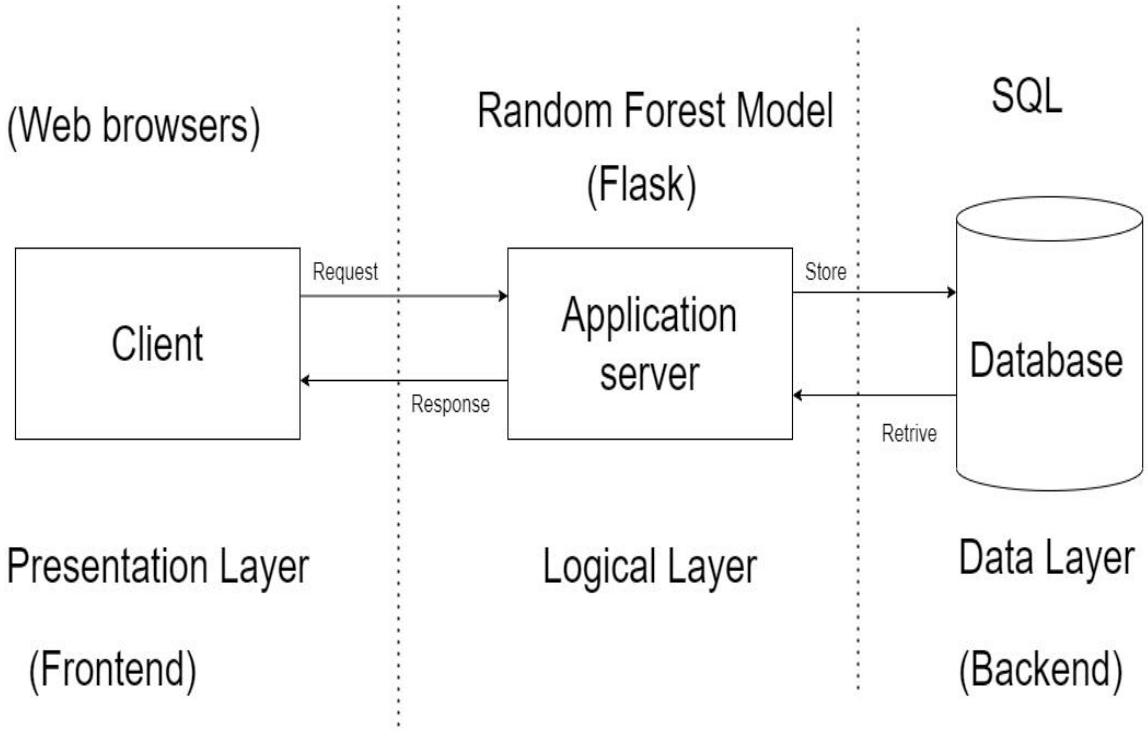


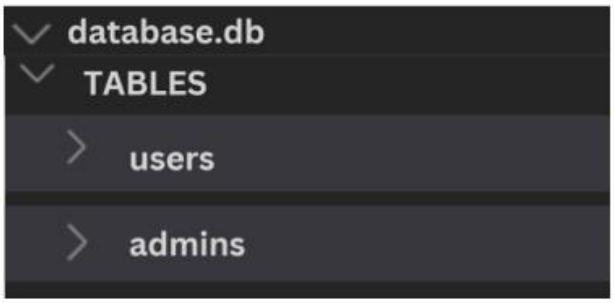
Figure 4.1 Three tire architecture of Crop Recommendation System

4.1.2 Database Design

The success of the Crop Recommendation System (CRS) depends on its well-structured database, which is essential for storing user information. An SQL database is used to efficiently manage and store the user data related to the system, ensuring smooth operation and easy access to the necessary information.

users

id
username
name
email
date_added
password_hash



admins

id
username
name
email
date_added
password_hash

Figure 4.2 Database design of Crop Recommendation System

4.1.3 Forms and Interface Design

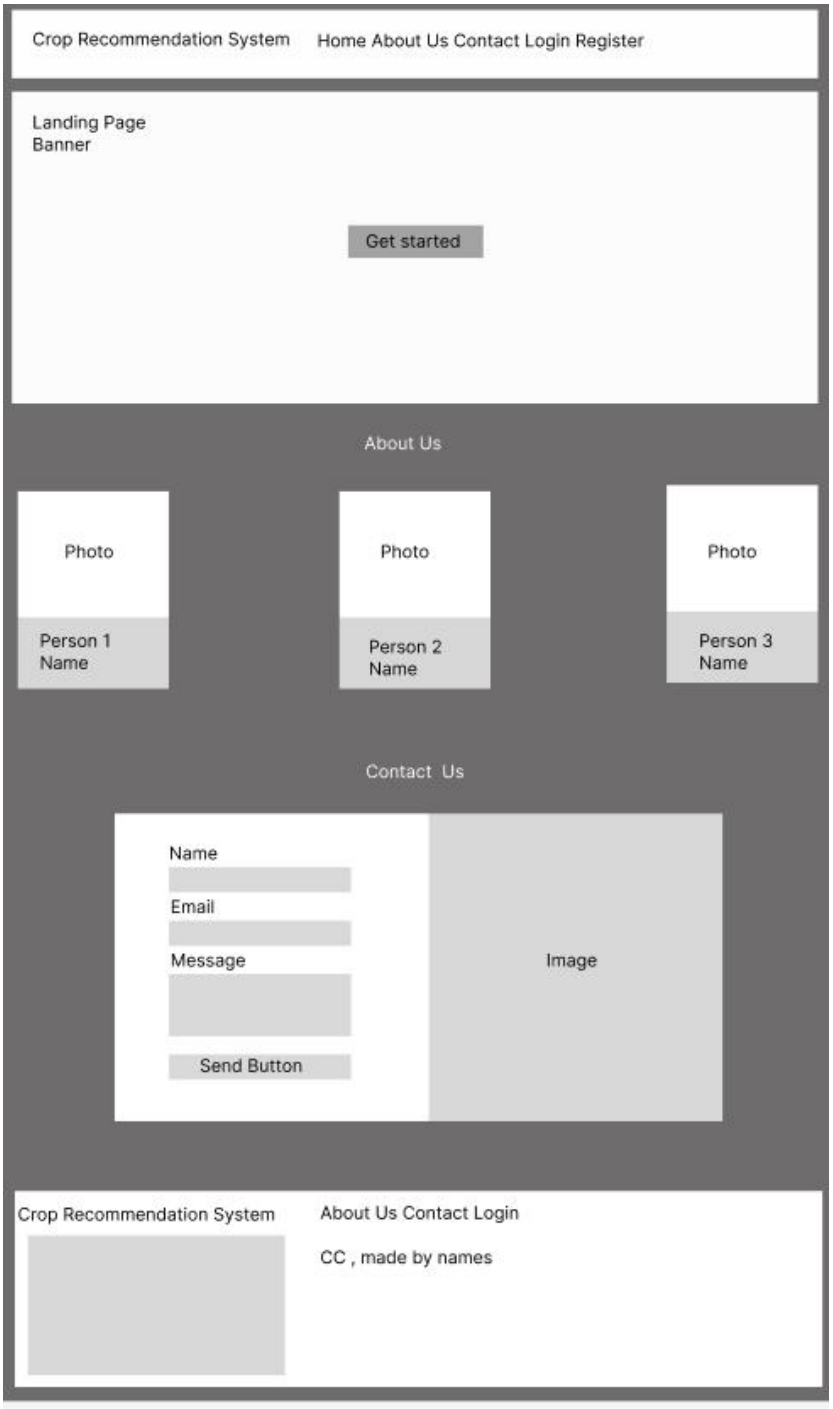


Figure 4.3 Home page design

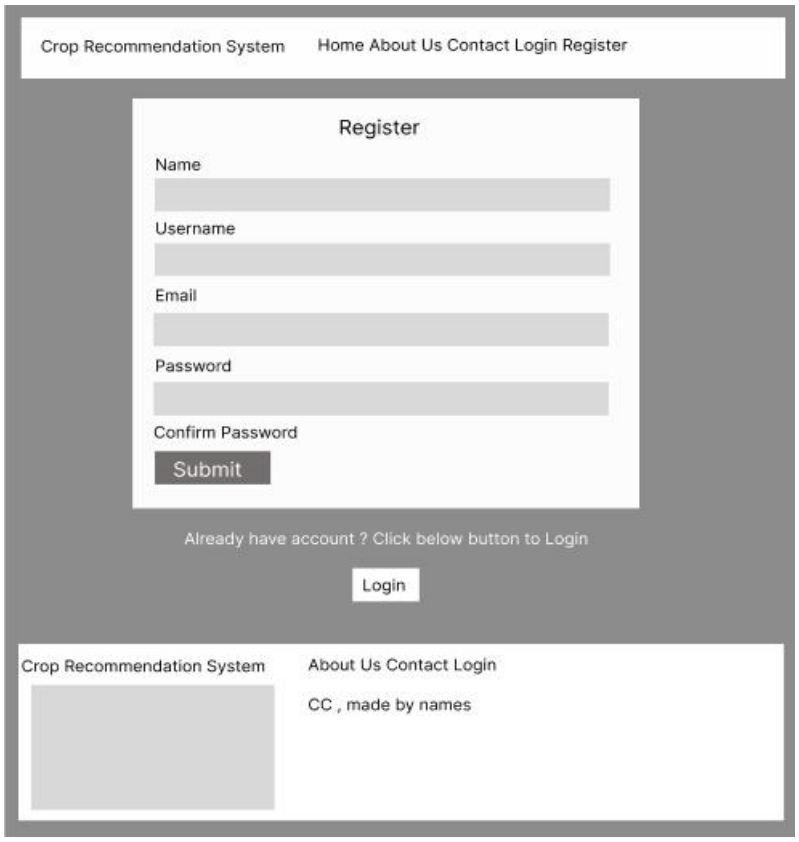


Figure 4.4 Registration page design

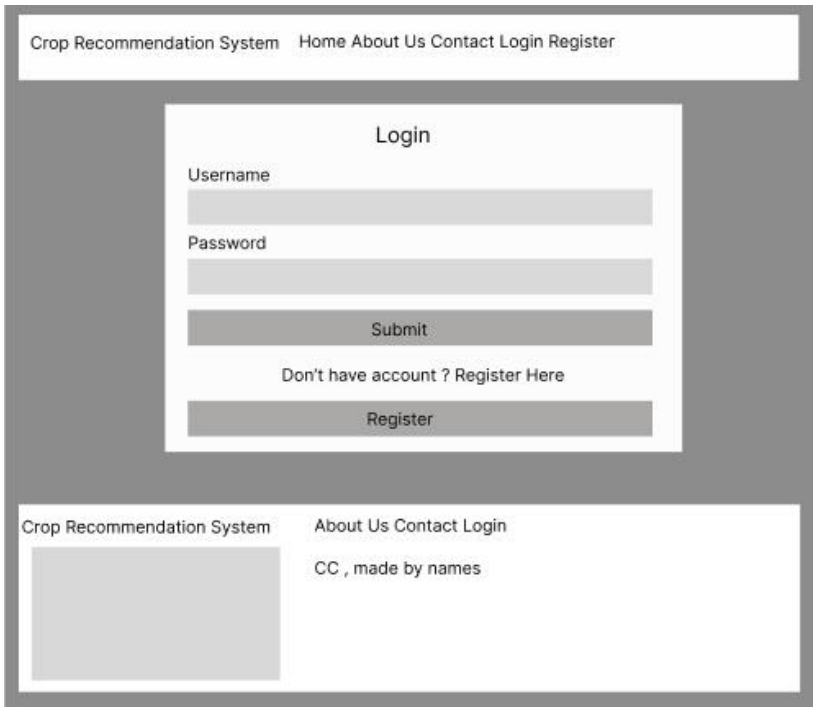


Figure 4.5 Login page design

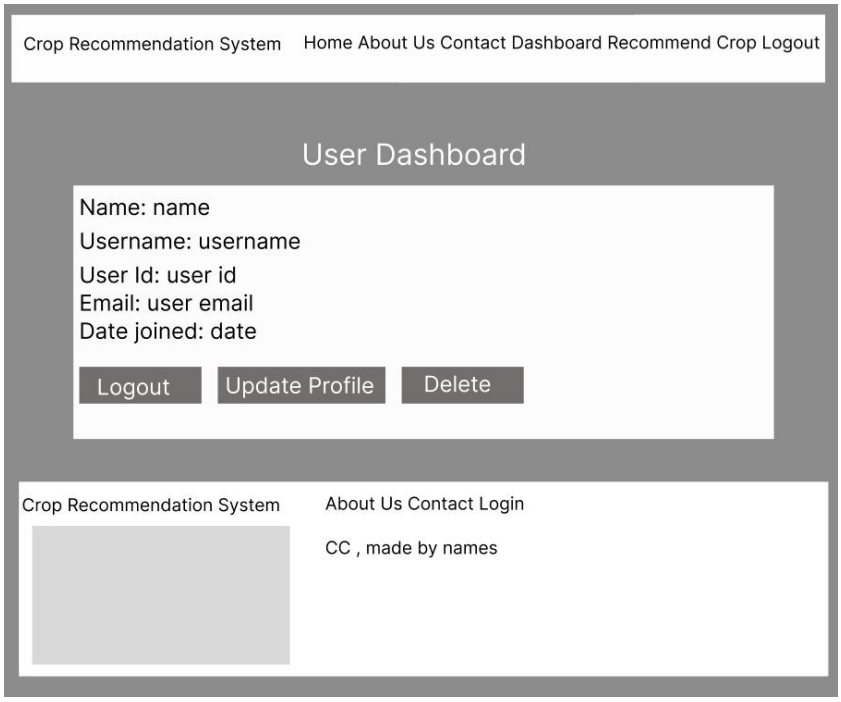


Figure 4.6 User dashboard design

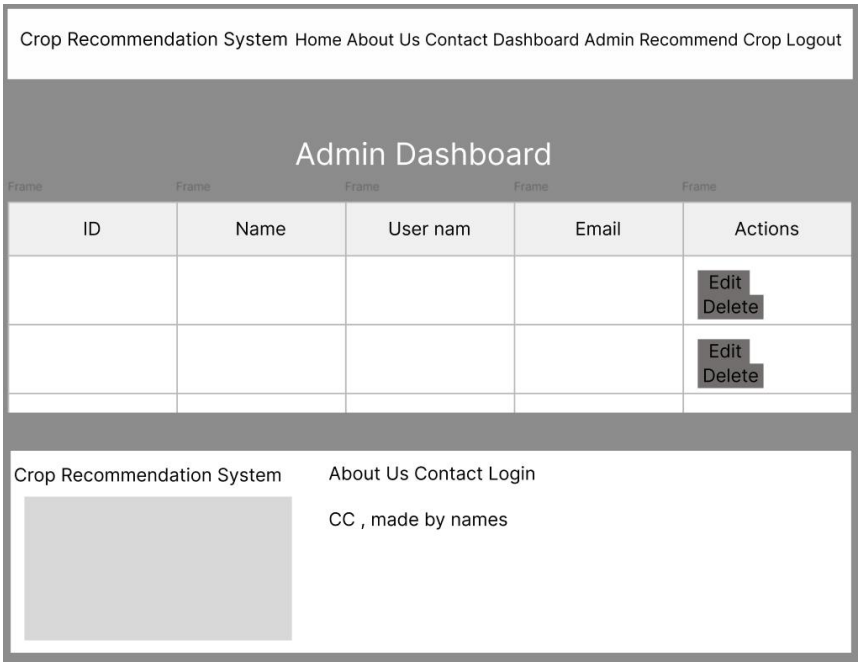


Figure 4.7 Admin dashboard design

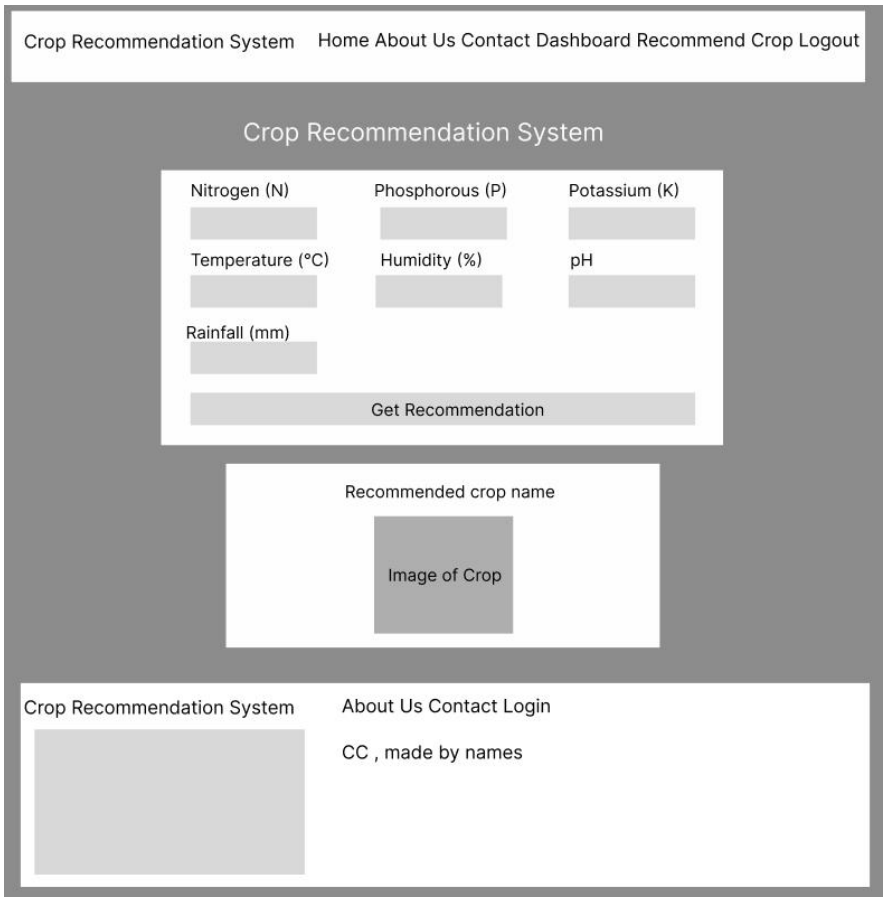


Figure 4.8 Crop recommendation module design

4.2 Algorithm Details

4.2.1 Random Forest Algorithm

Random Forest is a common machine learning algorithm that falls within the supervised learning category. It can be applied to both classification and regression issues in machine learning. It is based on the concept of ensemble learning, which involves merging numerous classifiers to solve a complex problem and improve model performance. According to its description, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Rather than depending on a single decision tree, the random forest collects forecasts from each tree and predicts the ultimate output based on the majority of predictions.

The more trees in the forest, the higher the accuracy and the lower the risk of overfitting. The following diagram demonstrates how the Random Forest algorithm works:

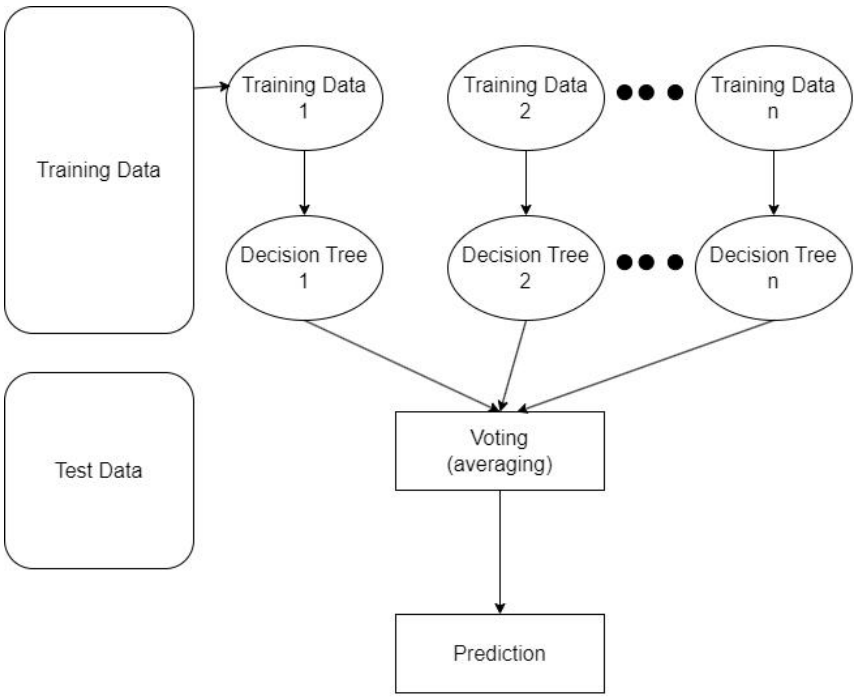


Figure 4.9 Working of Random Forest Algorithm

Algorithm for Random Forest in Machine Learning:

Step 1: Select random samples from a given data or training set.

Step 2: This algorithm will construct a decision tree for every training data.

Step 3: Voting will take place by averaging the decision tree.

Step 4: Finally, select the most voted prediction result as the final prediction result.

Random Forest Model

The Random Forest model of the Crop Recommendation System works as a group of decision-makers, with each decision tree making its own prediction. The model combines 100 decision trees' individual predictions into a final recommendation. While additional trees improve accuracy, they also require more computer resources. Furthermore, the model is set up to produce consistent results, assuring repeatability regardless of how many times the code is executed.

Training the Model: The model gains knowledge from past data during the training stage. It investigates input characteristics like as soil conditions and environmental influences, as well as the resulting crop outcomes. As more data is processed, the Random Forest model

improves its ability to recognize patterns and relationships, allowing it to provide accurate crop suggestions for similar situations in the future.

Making Predictions: Once the model has been trained and learned from the data, it can be tested against new, previously unknown data. 75% of the data was used for training the model, while 25% was used for testing. Using the patterns discovered during training, the model predicts the most suitable crops for the new input data.

Measuring Accuracy: To evaluate the model's performance, we compare its predictions with the actual outcomes. The closer the predicted crops are to the real recommendations, the higher the model's accuracy. This accuracy is expressed as a percentage, indicating how frequently the model makes correct predictions.

4.2.3 Evaluation Matrix

To evaluate the performance and reliability of a classification model, we use specific metrics. Here's a simple breakdown of these metrics.

1. Accuracy

Accuracy provides an overall picture of how often the model is correct. This is the proportion of correct forecasts (both positive and negative) to total predictions.

Formula:

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

2. Pression

Precision shows how often the model is accurate when recommending a specific crop. For example, if the model suggests 100 crops, precision indicates how many of those recommendations are truly suitable for the given conditions.

Formula:

$$\text{Pression} = \frac{TP}{(TP+FP)}$$

3. Recall

Recall measures how many actual positives the model correctly identifies. It's like asking: out of all the suitable crops, how many did the model correctly recommend?

Formula: =TP/ (TP +FN)

Recall = $\frac{TP}{(TP+FN)}$

4. Confusion Matrix

A confusion matrix visualizes the model's performance. The table displays the accuracy of the model's predictions for each category. This table displays both the model's strengths and weaknesses, providing insight into its performance.

There are various types of errors that occur.

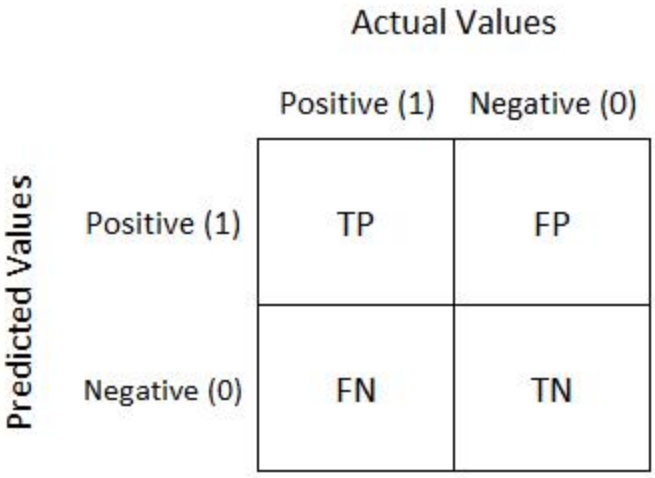


Figure 4.10 Confusion Matrix

Chapter 5: Implementation and Testing

5.1 Implementation

The Crop Recommendation System implementation process consists of several steps, including defining requirements and planning, selecting the appropriate methodology, selecting a framework, designing the system, developing frontend components, implementing backend functionality, testing and quality assurance, and optimizing the system for improved performance.

5.1.1 Tools used

Python: Backend programming language used for the project.

• Flask: Web framework utilized for developing the web application.

• Draw.io: Used for creating diagrams like DFD and ER diagrams.

• SQL Database: Relational database used for storing user information.

• Jupyter Notebook: Used for building and training the Random Forest model.

• VS Code: IDE for writing and running code.

• Microsoft Word: Used for preparing project documentation.

• Microsoft Excel: Used for creating Gantt charts and writing the test cases.

5.1.2 Implementation details of Modules

Some of the modules implemented in our system include the following:

User Management Module:

This module allows new users to sign up by creating an account and enables existing users to log in securely. It also provides functionality for users to log out, ensuring the safe termination of their session. The module stores essential user details, such as username, email, and password, in the user database.

Crop Recommendation Module:

This module allows users to input agricultural data, such as soil parameters and environmental conditions, to receive crop recommendations. The recommendation system

uses a Random Forest algorithm, which processes the input data to suggest the most suitable crops based on historical patterns. The system not only provides crop recommendations but also displays useful insights, helping farmers make informed decisions. It connects with the database to retrieve relevant crop information and recommendations based on the user's inputs.

Session Management Module:

Flask-Login handles user session management by securely storing session data in a cookie on the client-side. When a user logs in, Flask-Login creates a session by storing their user ID, which is retrieved and validated with each request. It uses login_user() to log the user in and logout_user() to clear the session when they log out. Flask-Login ensures that only authenticated users can access certain features, like crop recommendations, using the @login_required decorator. The session is automatically managed, providing a seamless experience while ensuring security and persistence across multiple requests.

5.2 Testing

Software testing ensures system reliability, functionality, and quality standards. The process involves analyzing both individual components and the entire system to find and address any possible concerns.

5.2.1 Test Cases for Unit Testing

Unit testing ensures that each module or component performs its function correctly.

Table 5.1 Test Cases for Registration/Login

Test Case Id	Test case description	Steps to perform	Expected Outcome	Actual Outcome	Status
TC-01	Register new user with valid inputs	Provide email: johndoe@gmail.com Created Password: password123	Sign up Successfully and enter to home page.	As expected	Pass
TC-02	Login new user with new data to check registration works or not	Email: johndoe@gmail.com Password: password123	Login successfully.	As expected	Pass
TC-03	User login with valid input	Email: elon@gmail.com Password: elon123	Login successfully.	As expected	Pass
TC-04	User login with invalid input	Email: mark@gmail.com Password: 1234	Login failed display error message.	As expected	Pass

Table 5.2 Test Cases for Crop Recommendation Model

Test case Id	Test case description	Steps to perform	Expected Outcome	Actual Outcome	Status
TC-01	Filling all fields	Filling all fields to get crop recommendations	A suitable crop is recommended based on the input values.	As expected	Pass
TC-02	Empty Fields	Trying to press the recommend button with all fields empty	Error message display.	As expected	Pass

Table 5.3 Test Cases for Admin Panel

Test case Id	Test case description	Steps to perform	Expected Outcome	Actual Outcome	Status
TC-01	Admin view the users list	Navigate to admin dashboard	User list table to be displayed.	As expected	Pass
TC-02	Admin Delete or Edit user information	Navigate to admin dashboard and click on action buttons	Updated details to be reflected in the user list and database.	As expected	Pass

5.2.2 Test Cases for System Testing

System testing ensures that all components function properly. This phase enables seamless integration of modules and a smooth user experience.

Table 5.4 Test Cases for Responsiveness of System

Test case Id	Test case description	Steps to perform	Expected Outcome	Actual Outcome	Status
TC-01	Validate connection between backend and model	Send processed data to the model and check the models response	Accurate prediction.	As expected	Pass
TC-02	Test data flow from model to result display	Submit data and process backend and fetch prediction results and display output on the screen.	Results are displayed. clearly and accurately	As expected	Pass
TC-03	Verify data flow between input form and backend	Fill in the input form with valid data, submit and check backend processing.	Data is correctly validated, processed, and stored.	As expected	Pass

Table 5.5 Test Cases for Responsiveness

Test case Id	Test case description	Steps to perform	Expected Outcome	Actual Outcome	Status
TC-01	Check UI adaptability.	Interact with the UI by resizing the browser or switching between devices	UI adjusts seamlessly without glitches or misalignment.	As expected	Pass
TC-02	Test concurrent user access	Simulate multiple users accessing the system simultaneously	System performs smoothly without noticeable delays.	As expected	Pass

Table 5.6 Test Cases for Usability Testing

Test case Id	Test case	Steps to	Expected	Actual	Status

	description	perform	Outcome	Outcome	
TC-01	Test page navigation	Navigate between various pages or modules of web app	Navigation to be smooth.	As expected	Pass
TC-02	Verify clarity of error messages	Enter invalid data in forms or simulate incorrect actions	Error messages are clear, helpful and easy to understand.	As expected	Pass

5.3 Result Analysis

The Crop Recommendation System is designed to provide accurate crop suggestions based on environmental factors and soil conditions. To evaluate its performance and user experience, various metrics and real-world scenarios were analyzed, offering valuable insights into its effectiveness and potential areas for enhancement.

5.3.1 Evaluating Model Performance

- Accuracy

The system achieves a high validation accuracy rate of 98%, demonstrating its ability to recommend the most suitable crops in the majority of cases.

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,precision_score,recall_score,f1_score
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9890909090909091
```

Figure 5.1 Accuracy of model

- Precision:

Precision indicates how accurately the system avoids incorrect recommendations. For example, if the model suggests a specific crop, it is correct 98.9% of the time.

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,precision_score,recall_score,f1_score

print("\nPrecision:\n", precision_score(y_test, y_pred, average='micro'))
```

```
Precision:
 0.9890909090909091
```

Figure 5.2 Precision of model

- Recall:

This metric measures the system's ability to detect all actual disease cases, with a recall of 98.9%. It ensures that the model doesn't miss many cases.

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,precision_score,recall_score,f1_score

print("\nRecall:\n", recall_score(y_test, y_pred, average='micro'))
```

```
Recall:
 0.9890909090909091
```

Figure 5.3 Recall score of model

- F1 Score:

The F1 score, which balances precision and recall, is 98.9%, highlighting the system's reliability in recommending suitable crops while minimizing errors.

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,precision_score,recall_score,f1_score

print("\nF1 Score:\n", f1_score(y_test, y_pred, average='micro'))
```

```
F1 Score:
 0.9890909090909091
```

Figure 5.4 F1 score of model

- Confusion Matrix:

The system's predictions were evaluated using a confusion matrix, revealing key insights:

- It performed exceptionally well in recommending crops with distinct and well-defined input patterns.
- However, crops with similar or overlapping features were more challenging to classify accurately.
- While the majority of predictions were accurate, occasional errors were observed, including false positives (recommending an unsuitable crop) and false negatives (failing to recommend a suitable crop).

```
Confusion Matrix:
 [[29  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 23  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 24  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 31  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 26  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 19  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 27  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 24  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 15  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  1  0  0 25  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 22  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 19  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 31  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 34  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0]
 [ 0  0  0  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0 20  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 21]]
```

Figure 5.5 Confusion matrix of the model

Chapter 6: Conclusion and Future Recommendations

6.1 Conclusion

The Crop Recommendation System overcomes the gap between advanced data science and conventional farming methods, representing a revolutionary approach to current agriculture. The technology gives farmers precise, data-driven crop suggestions by using the Random Forest algorithm, allowing them to make decisions that are specific to their own surroundings.

By improving resource use, this advanced instrument not only raises agricultural output and lowers the chance of crop failure, but it also encourages sustainable farming methods. The widespread use is made possible by its user-friendly interface, which guarantees availability even for farmers with little technological knowledge.

The system handles important issues like crop selection complexity and environmental variability, but it could be even more responsive and inclusive in the future if it included real-time data, IoT integration, and multilingual support. The method is a big step toward more intelligent and sustainable agriculture, even with drawbacks like data dependency and technology accessibility.

Basically, by helping farmers to adopt precision farming, the Crop Recommendation System delivers better yields, reduced risks, and a sustainable agricultural future. Agriculture and technology working together has the potential to transform farming methods globally by making them more capable of adapting to constantly changing environmental circumstances.

6.2 Future Recommendations

The Crop Recommendation System's success depends on its ability to integrate modern technologies, enhance its use, and solve issues that restrict its success. The system can increase crop suggestion accuracy by implementing advanced machine learning algorithms. To capture complex relationships in the data, algorithms like Neural Networks, XGBoost, and Gradient Boosting could be investigated. Using collective learning to combine several models may help improve the prediction strength.

Future advancements for the Crop Recommendation System include:

- Real-Time Data Integration: For precise and quick recommendations, use real-time data such as insect activity, soil moisture levels, and weather forecasts.
- IoT Integration: To collect accurate on-field data, use Internet of Things devices such as crop health monitors, weather stations, and soil sensors.
- Multilingual Support: Provide local language options, such Nepali, to increase accessibility for farmers who don't understand English.
- Data Visualization: To assist farmers in understanding farm data and crop recommendations, introduce advanced tools such as heatmaps, charts, and graphs.
- Mobile Applications: Provide mobile applications with offline support to increase usefulness in remote areas.
- Sustainability Metrics: To support sustainable agriculture, share information about water use, carbon footprint, and environmentally friendly methods.

References

[1] T. W. R. T. W. N. D. M. P. M. Pradeepa Bandara, "Crop Recommendation System," International Journal of Computer Applications, 2020.

[2] S. M. G. Lakshman Kumar Seru, "Crop Recommendation System Using Machine Learning," Sathyabama Institute of Science and Technology, 2022.

[3] D. K. S. Ranjith Macharla, "Crop Recommendation System," International Research Journal of Modernization in Engineering Technology and Science, 2022.

[4] G. B. a. D. S. Radhimeenakshi, "Crop Recommendation System Using Random Forest Algorithm," International Journal of Research Culture Society ISSN, 2023.

[5] M. Pradhan, "Nepal Economic Forum," [Online]. Available: https://nepaleconomicforum.org/nepals-agricultural-landscape-accessing-the-governments-stance/.

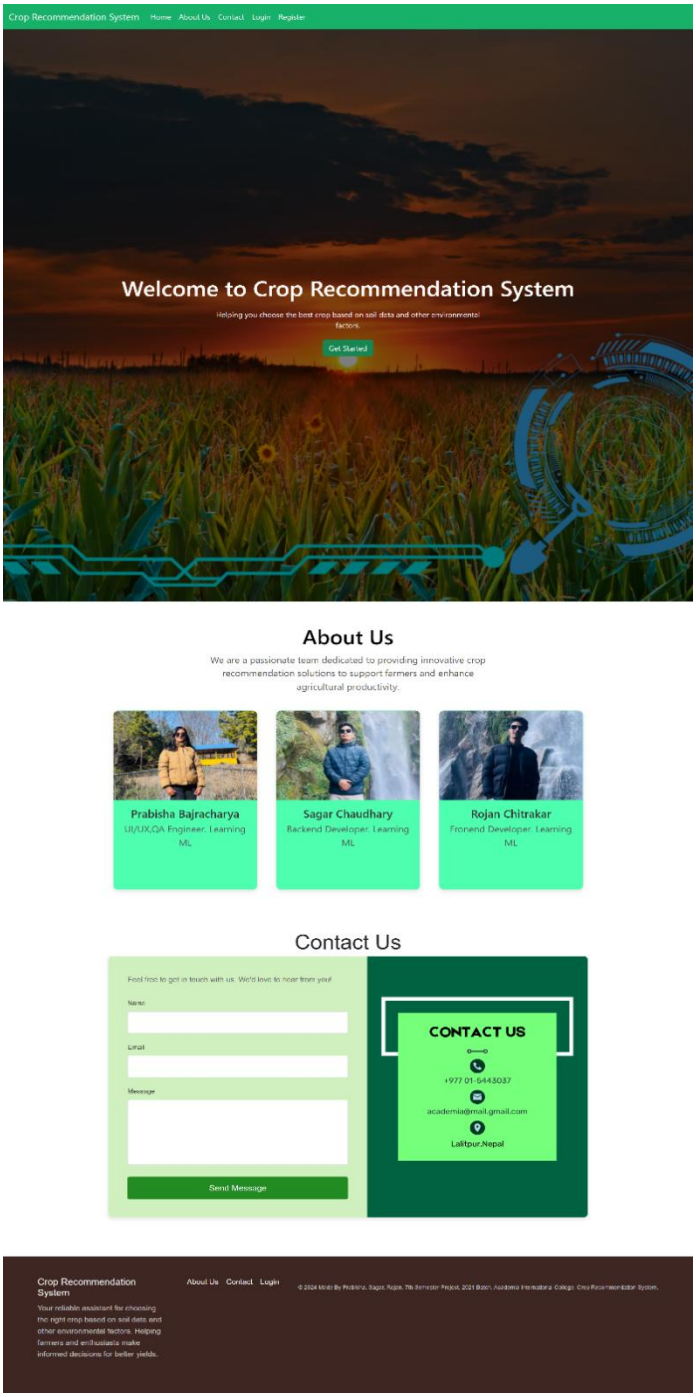
[6] G. f. geeks, "geeks for geels," [Online]. Available: https://www.geeksforgeeks.org/what-is-agile-methodology/.

[7] S. Laoyan, "asana," [Online]. Available: https://asana.com/resources/agile-methodology.

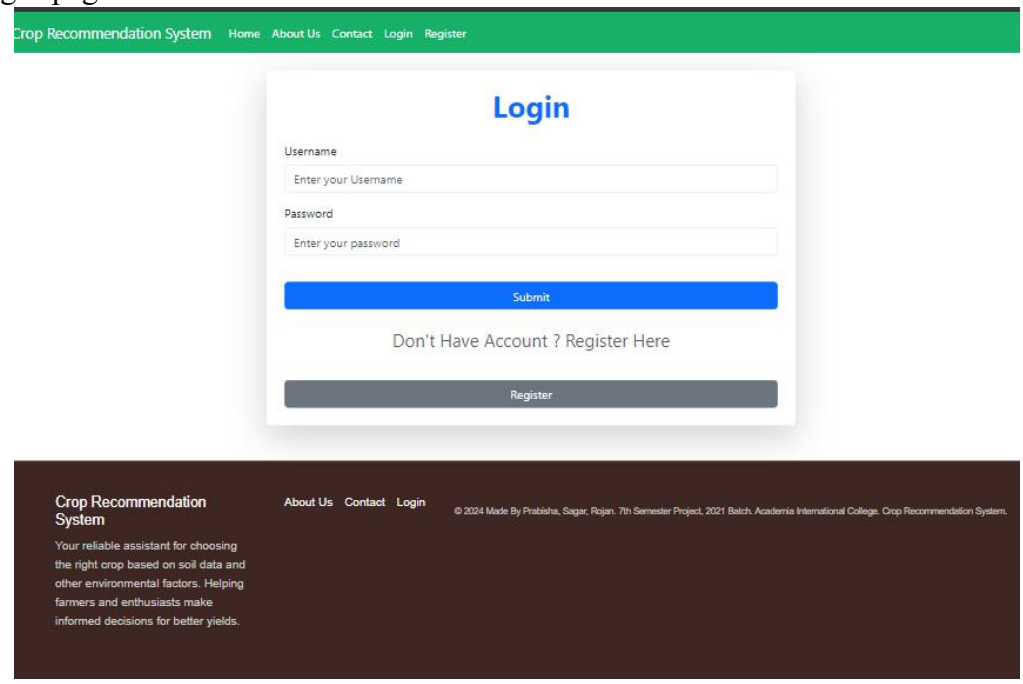
Appendices

Screenshots of the web application

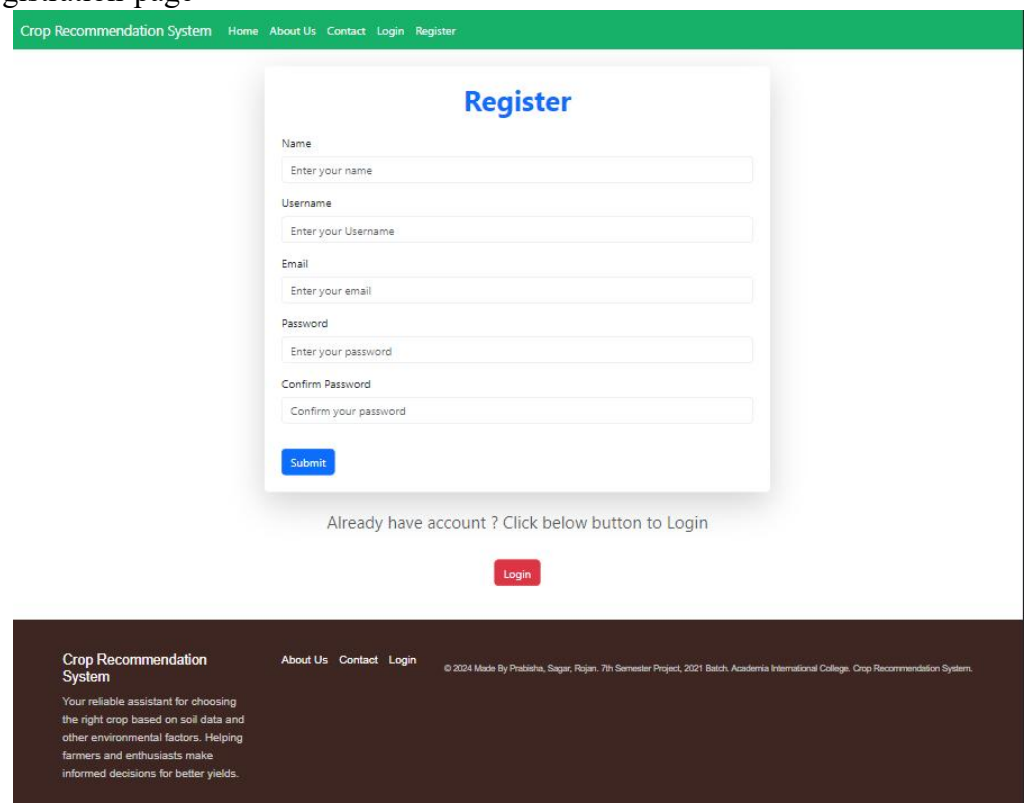
a. Home-page



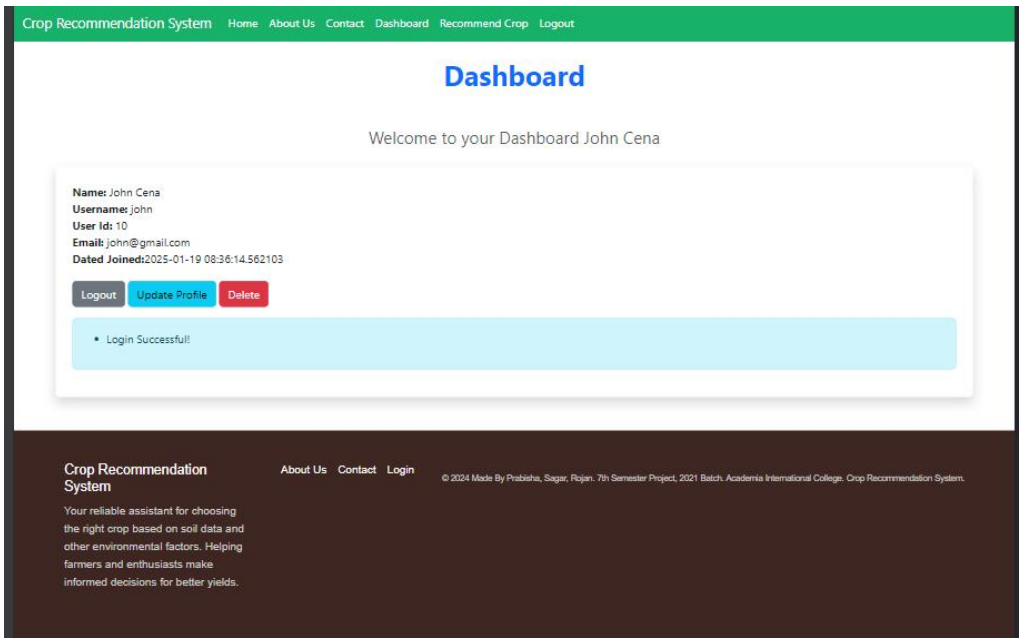
b. Login-page



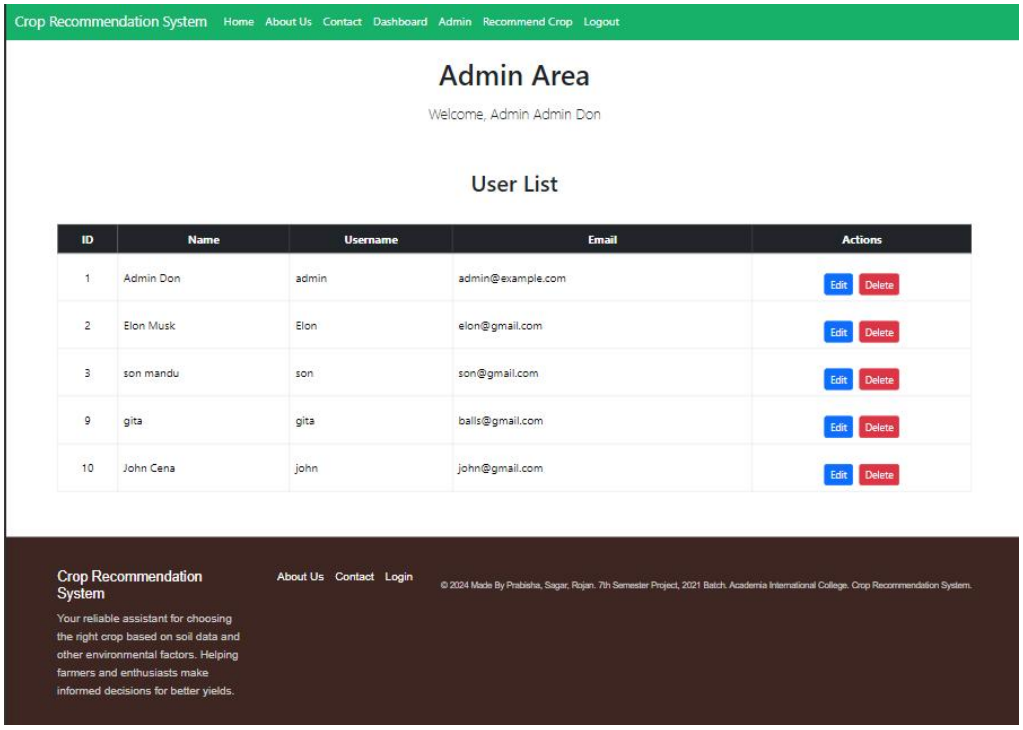
c. Registration-page



d. User dashboard



e. Admin dashboard



f. Crop Recommendation Module