Amrit Campus

Tribhuvan University

Institute of Science and Technology

A Project Report

ON

Crop Recommendation System with integrated crop cure and marketplace functionality

Submitted to

Department of Computer Science and Information Technology

Amrit Campus

Lainchaur, Kathmandu

Nepal

In partial fulfilment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted By:

Ashish Khatri 23123/076

Kritan Sitaula 23149/076

Nischal Kafle 23165/076

Under the supervision of

Mr. Rakesh Kumar Bachchan

March, 2024

AMRIT CAMPUS

Affiliated to Tribhuvan University

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by Ashish Khatri (23123/076), Kritan Sitaula (23149/076), and Nischal Kafle (23165/076) entitled "**Crop Recommendation System – with integrated cure recommendation and marketplace**" built in partial fulfillment of requirement of the degree of BSc. in Computer Science and Information Technology (B.Sc. CSIT).

Mr. Rakesh Kumar Bachchan
Supervisor

Department of Computer Science and Information Technology

Amrit Science Campus

Lainchaur, Kathmandu

AMRIT CAMPUS

Affiliated to Tribhuvan University

Certificate Of Approval

This is to certify that this project prepared by Ashish Khatri, Kritan Sitaula, Nischal Kafle entitled "**Crop Recommendation System – with integrated cure recommendation and marketplace**" has been submitted to the Department of Computer Science and Information Technology for acceptance in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Mr. Rakesh Kumar Bachchan
Supervisor/Lecturer
Amrit Campus

Head of Department
Department of Computer Science
and Information Technology
Amrit Campus

External Examiner
IOST, Tribhuvan University

Internal Examiner
Amrit Campus

Acknowledgement

We would like to express our sincere gratitude to all those who have contributed to the completion of this project report.

First and foremost, we extend our heartfelt thanks to **Mr. Rakesh Kumar Bachchan**, our project supervisor, for his invaluable guidance, support, and mentorship throughout the duration of this project. His expertise, encouragement, and constructive feedback have been instrumental in shaping the direction and success of our work.

We are deeply thankful to our faculty members and academic advisors for their unwavering support and encouragement throughout the project. Their insights, expertise, and encouragement have been invaluable in guiding us through the various stages of the project.

We are grateful to our friends and family members for their unwavering support, encouragement, and understanding during the challenging periods of this project. Their words of encouragement and belief in our abilities have been a source of strength and motivation.

Last but not least, we express our gratitude to all the individuals and organizations who have directly or indirectly contributed to the success of this project. Your support and collaboration have been invaluable, and we are deeply appreciative of your contributions.

Thank you all for being part of this journey and for your invaluable support and encouragement.

Ashish Khatri
Kritan Sitaula
Nischal Kafle

Abstract

This project aims in developing a software system for crop recommendation, plant cure recommendation, direct-to-consumer marketing of the agricultural products. The system takes the value soil nutrients – Nitrogen, Phosphorous, Potassium and pH and climatic conditions – Temperature, Humidity and Rainfall as input and apply Random Forest Classification Algorithm for prediction of suitable crops under supplied conditions. Similarly, the system predicts plant cure recommendation on the basis of value of soil nutrient – Nitrogen, Phosphorous and Potassium. Also, the system allows the admin user to sell seeds as well as products using the integrated online marketplace. Farmer users can order the seeds of the crop directly from this software solution. It also allows the farmer user to sell and market their agricultural products online.

Keywords: *crop-recommendation; cure-recommendation; soil-nutrients; climatic-conditions, direct-to-consumer;*

Table of Contents

Supervisor's Recommendation i
Certificate Of Approval ii
Acknowledgement iii
Abstract iv
Table of Contents v
List of Tables vii
List of Figures viii
Chapter 1 - Introduction 1
1.1 Introduction 1
1.2 Problem Statement 2
1.3 Objectives 3
1.4 Scope and Limitation 3
1.5 Development Methodology 4
1.6 Report Organization 4
Chapter 2 - Background Study and Literature Review 6
2.1 Background Study 6
Crop Recommendation System 6
Plant cure recommendation system 6
Online Marketplace for Agronomics 6
2.2 Literature Review 7
Chapter 3 - System Analysis 9
3.1 Requirement Analysis 9
3.2 Feasibility Analysis 11
Chapter 4 - System Design 13
4.1 Design 13

4.2 Algorithm Detail....................18

Random Forest Classifier....................18

CART Algorithm....................19

Chapter 5 Implementation and Testing....................23

5.1 Data Collection....................23

5.2 Implementation....................24

5.2.1 Tools Used....................24

5.2.2 Implementation Details of Modules....................24

5.2 Testing....................25

5.2.1 Test Cases for Unit Testing....................25

5.2.2 Test Cases for System Testing....................29

5.3 Result Analysis....................33

Chapter 6 Conclusion and Future Recommendation....................35

6.1 Conclusion....................35

6.2 Future Recommendation....................35

References....................i

Appendices....................i

List of Tables

Table 3.1: Working Schedule 12
Table 5.1: Tools Used 24
Table 5.2: Unit Testing of Authentication and Authorization 25
Table 5.3: Unit test of Banner Management 26
Table 5.4: Unit Testing of Product module 26
Table 5.5: Unit Testing of Seed module 27
Table 5.6: Unit testing of User Module 28
Table 5.7: Unit testing of cart module 28
Table 5.8: Unit testing of order module 29
Table 5.9: Unit testing of transaction module 29

List of Figures

Figure 1.1: Project's Abstract 1
Figure 3.1: Use Case Diagram 9
Figure 4.1: Class Diagram 13
Figure 4.2: Crop Recommendation's State Diagram 13
Figure 4.3: Cure Recommendation's State Diagram 14
Figure 4.4: Ecommerce's State Diagram 14
Figure 4.5: Activity Diagram 15
Figure 4.6: Sequence Diagram 16
Figure 4.7: Component Diagram 17
Figure 4.8: Deployment Diagram 17
Figure 4.9: Random Forest Classification 18
Figure 4.10: Crop Recommendation System Flow Diagram 21
Figure 4.11: CART Algorithm Flow Diagram 22
Figure 5.1: Snapshot of dataset head 23
Figure 5.2: Shape of data 23
Figure 5.3: Testing of Admin CMS 30
Figure 5.4: User Data in Database 30
Figure 5.5: Testing of crop recommendation module 31
Figure 5.6: Testing for farmer CMS 31
Figure 5.7: Testing of customer 32
Figure 5.8: Paypal testing 32
Figure 5.9: Accuracy Rates for Different Train-Test Splits 34

Chapter 1 - Introduction

1.1 Introduction

The fusion of technology and traditional farming practices is necessary for the modernization of agriculture. A software tailored to address the diverse needs of farmers by data-driven decision-making and integration of agricultural processes is introduced in this project. This college project aims to bridge the gap between traditional farming methodologies and technology through the development of an innovative software solution. This software is engineered to address three key stages of the farming cycle: planning, cultivation, and marketing.

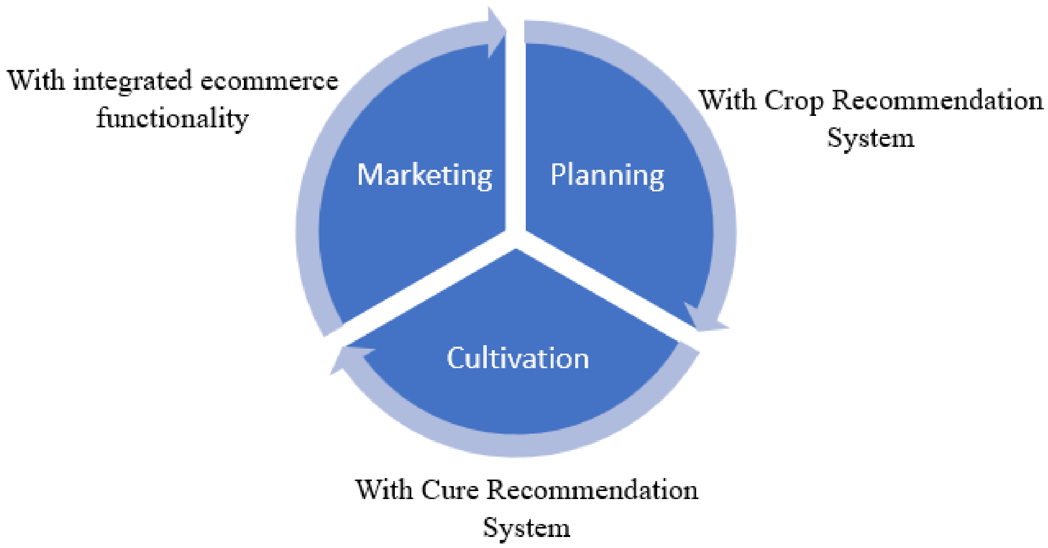


Figure 1.1: Project's Abstract

1. The first stage of this software solution allows users to input data regarding soil nutrients, such as Nitrogen, Phosphorous, Potassium, and pH levels, alongside climatic parameters including temperature, humidity, and rainfall patterns. Utilizing a Random Forest classification algorithm, the software intelligently analyzes this data to recommend suitable crops for cultivation, tailored to the specific conditions of the farm using huge dataset. Furthermore, this software also simplifies online ordering of seeds for the recommended crops.
2. This software extends support to farmers during the cultivation phase by providing plant care recommendations. By inputting soil nutrient levels and the name of the

cultivated crop, farmers can receive tailored suggestions for optimal fertilization and pest management strategies, optimizing crop health and productivity.

3. Moreover, upon harvesting, this software facilitates direct online sales of farm produce to consumers through a user-friendly e-commerce platform. By cutting out intermediaries and enabling direct-to-consumer transactions, farmers can establish stronger connections with consumers, enhance market reach, and command better prices for their produce.

This software solution for the farming cycle stands to benefit multiple stakeholders within the agricultural ecosystem. Firstly, farmers can gain significant advantages through the software's capabilities. By providing input on soil nutrients and climatic conditions, farmers can receive tailored crop recommendations, enabling them to make informed decisions about which crops to cultivate. This empowers farmers to optimize their agricultural practices, maximize crop yields, and minimize resource inputs. Additionally, the direct online ordering feature of recommended seeds allows farmers to easily access quality seeds tailored to their specific needs. Furthermore, the plant cure recommendations provided by the software assist farmers in managing plant health and addressing agricultural challenges, which leads to improved crop quality and profitability. On the consumer side, the integration of e-commerce capabilities enables direct-to-consumer marketing, offering consumers convenient access to fresh produce from local farms. This benefits consumers by providing access to high-quality, locally grown agricultural products. Overall, the software solution facilitates enhanced productivity, sustainability, and market connectivity, ultimately driving positive outcomes for farmers, consumers, and the agricultural industry as a whole.

1.2 Problem Statement

Current agricultural practices suffer various problems. Among which, lack of expertise in crop selection, lack of expertise during cultivation and lack of online platforms to sell farm products are some. This project has tried to address these issues with three different software solution integrated in one package. A unified software solution is needed to integrate soil analysis, crop selection, procurement, plant care recommendations, and direct-to-consumer marketing, addressing these challenges and empowering farmers with streamlined processes and enhanced market connectivity.

1.3 Objectives

The objectives of the project can be stated as:

1. To develop the software solution that provides recommendation on selection of crop to grow during initial phase of agricultural process.
2. To provide plant cure recommendations during the cultivation phase.
3. To provide platform to sell agricultural product after harvesting.

1.4 Scope and Limitation

1.4.1 Scope

As a web-based system, the system will be easily accessible to those with a device supporting a browser and an Internet Connection. The software will allow users to input soil nutrient data (Nitrogen, Phosphorous, Potassium, pH) and climatic conditions (temperature, humidity, rainfall). The software will facilitate direct online ordering of recommended crop seeds from authorized suppliers, ensuring timely access to quality inputs for farmers. Upon cultivation, the software will provide personalized plant cure recommendations based on soil nutrient levels (N, P, K) and crop name, helping farmers in optimizing crop health and productivity. Farmers will be empowered to sell their harvested produce directly to consumers through an online platform integrated with a robust Content Management System (CMS), facilitating seamless transactions and market connectivity.

1.4.2 Limitation

Some of the limitation of this software solution is listed below:

1. Farmers need a different device to measure the soil nutrients value before providing input in this system.
2. The accuracy of crop recommendation and cure recommendation highly relies on the input data regarding NPK, pH, temperature, humidity and rainfall. Variability and inadequacy in data quality highly impacts the accuracy.
3. System doesn’t take account for specific regional or microclimatic variations that could influence crop suitability and management practices.

4. Raising technological literacy among farmers, making digital infrastructure accessible to farmers and comprehensive support and training to them are challenging.

1.5 Development Methodology

For this project, XP Methodology (Extreme Programming) was adopted. With a small team of three, pair programming can be particularly effective. Task was done together in pairs, alternating roles between writing code and reviewing, thereby promoting knowledge sharing, collaboration, and higher code quality.

Continuous Integration (CI) was practiced by frequently integrating work in shared git repository (link: https://github.com/CSIT-final-year-project/Godmel). TDD (Test Driven Development) was practiced and ensured software meets specified requirements and remains maintainable over the time of the project.

1.6 Report Organization

This report is organized as per the standard guidelines provided in syllabus of Project Work of B.Sc. CSIT. The general structure of the report is briefly described below.

First page of the report is Cover page. It is followed by Supervisor's recommendation page, Approval Letter, Acknowledgement page, Abstract page, Table of contents, list of figures and tables page which are labelled with page numbers in roman numeral from i-viii.

The body of the report is split into six chapters:

1. Chapter 1 is titled as Introduction. It covers following six topics:
    a. Introduction
    b. Problem Statement
    c. Objectives
    d. Scope and Limitation
    e. Development Methodology
    f. Report Organization
2. Chapter 2 is titled as Background Study and Literature Review which contains:
    a. Background Study of Subject Matter
    b. Literature Review
3. Chapter 3 is for System Analysis which briefly covers:
    a. Requirement Analysis

        i. Functional Requirements
        ii. Non-Functional Requirements
    b. Feasibility Study
        i. Technical
        ii. Operational
        iii. Economic
        iv. Schedule
    c. Analysis
        i. Class Diagrams
        ii. State and Sequence Diagram
        iii. Activity Diagram
4. Chapter 4 depicts System Design. Following sub-headings are discussed in this chapter.
    a. Design
        i. Refinement of Class, State, Sequence and Activity Diagrams
        ii. Component Diagram
        iii. Deployment Diagram
    b. Detail of used algorithm
5. Chapter 5 includes briefing of Implementation details and result of testing which is structured as:
    a. Implementation
        i. Tools Used
        ii. Implementation Details of Modules
    b. Testing
        i. Unit Testing
        ii. System Testing
    c. Result analysis
6. Chapter 6 is for conclusion and future recommendation.
    a. Conclusion
    b. Future Recommendation

Furthermore, a page for references and citation is kept after above chapters. At last, snapshots of system and some important code is organized in appendix section.

Chapter 2 - Background Study and Literature Review

2.1 Background Study

Crop Recommendation System

A crop recommendation system is like having a smart assistant for farmers. It helps them decide which crops would be best on their land by analyzing a bunch of different information, like what kind of soil they have, what the weather is like, and how much rain they typically get. By looking at all this data, the system can suggest which crops are most likely to do well in that area. It's kind of like having a knowledgeable friend who knows all about farming and can give you personalized advice on what to plant. This way, farmers can make better decisions and hopefully get better harvests.

To make a crop recommendation system, set of data is required which contains the suitable soil nutrients, weather data under which specific crop can be grown. Once this data is collected and preprocessed, machine learning can be used to make predictions based on the nature of these data. These programs look for patterns and connections between different factors and what crops tend to grow well under those conditions. Then, based on all this analyzing, the system can suggest which crops are likely to do best in that particular area.

Plant cure recommendation system

Plant cure recommendation system is the system which is tailored to provide farmer with suggestion of caring the plant based on existing soil nutrient. This project aims to achieve functionality with the help of dataset which contains specific value of soil nutrient N, P and K for each crop under which that crop can do well. The system takes input of soil nutrient values and compare provided data with data in dataset. If any value is found to be high or low, it provides suggestion based on excessiveness or lacking of those nutrients.

Online Marketplace for Agronomics

E-commerce is very effective tools for selling and buying goods. Such online marketplace dedicated for agriculture is highly beneficial for farmers as well as consumers. For farmers, having e-commerce functionality means they can sell their produce directly to consumers without going through middlemen and also they can easily access buying of seeds online. This not only boosts their profits but also builds trust because consumers know exactly where their food comes from. Plus, being online lets farmers showcase their goods to a wider audience, promoting the freshness of local produce. For consumers, it's all about convenience. They can buy farm-fresh food online, often at better prices. This direct connection between farmers and consumers strengthens local farming and encourages sustainable food choices. Ultimately, e-commerce links everyone together, making transactions transparent and relationships stronger, all while supporting a healthier environment.

2.2 Literature Review

[1] Archana et. al has proposed a smart system that can assist farmers in crop management by considering sensed parameters (temperature, humidity) and other parameters (soil type, location of farm, rainfall) that predicts the most suitable crop to grow in that environment.

In [2] Anguraj. Ka and other has proposed new technologies include the use of Internet of Things (IOT) and Machine Learning. The real time data from the field area can be collected using IOT system. The collected data from the field area is fed to the trained model. The trained model then makes the predictions using the data. The result produced by the model greatly helps is sowing the suitable crops in the particular field area.

[3] A. Mythili, determines a model is proposed for predicting the soil type and suggest a suitable crop that can be cultivated in that soil. The model has been tested using various machine learning algorithms such as KNN, SVM and logistic regression. The accuracy of the present model is maximum than the existing models.

[4] Rani Holambe et. al proposed a system would assist the farmers in making an informed decision about which crop to grow depending on a variety of environmental and geographical factors. The ML and IoT based suggestions will significantly educate the farmer and help them minimize costs and make strategic decisions by replacing intuition and passed-down knowledge with far more reliable data-driven ML models. This allows for a scalable, reliable solution to an important problem affecting hundreds of millions of people.

Reference Paper [5] determines real time sampling of soil properties using MODIFIED SUPPORT VECTOR REGRESSION, a popular machine learning algorithm and four modules. The Modules include Sensor interfaced to IoT device, Agri cloud, Analyzing the real time sensor data and Agri user interface (AUI). The first module is portable IoT device (NodeMCU) with soil moisture sensor and pH sensor, environmental sensors. Agri cloud module consists of storage. Analyzing the real time data module is processing of types of crops and small plants suggested using modified support vector machine algorithm. Agri-user interface is a basic web interface. Thus, with the help of soil properties farmer will be able to get types of crops and small plants is grown in farmland with help of Modified support vector machine algorithm.

Aruul Mozhi Varman S proposed an IOT and deep learning based smart agriculture systems. This system monitors and collects the soil parameters from the field with the help of a wireless sensor network. The collected data is then uploaded in the cloud. Finally, the systems suggest best irrigation practices to the farmers by predicting the crop to be sown for next crop rotation. This information will be sent as an SMS to the farmers. The parameters include soil temperature, atmospheric temperature, and humidity [6]. This system suggests further improving the effectiveness by predicting the suitable time for applying pesticides, fertilizer, and manures.

In paper [7] proposed a system would assist the farmers in making an informed decision about which crop to grow depending on a variety of environmental and geographical factors. The ML and IoT based suggestions will significantly educate the farmer and help them minimize costs and make strategic decisions by replacing intuition and passed-down knowledge with far more reliable data-driven ML models. This allows for a scalable, reliable solution to an important problem affecting hundreds of millions of people

Chapter 3 - System Analysis

3.1 Requirement Analysis

Functional Requirements:

This system allows three kinds of user privileges. One is admin, another farmer and other is customer. The functionality available for each privilege are represented with Use case diagram below.

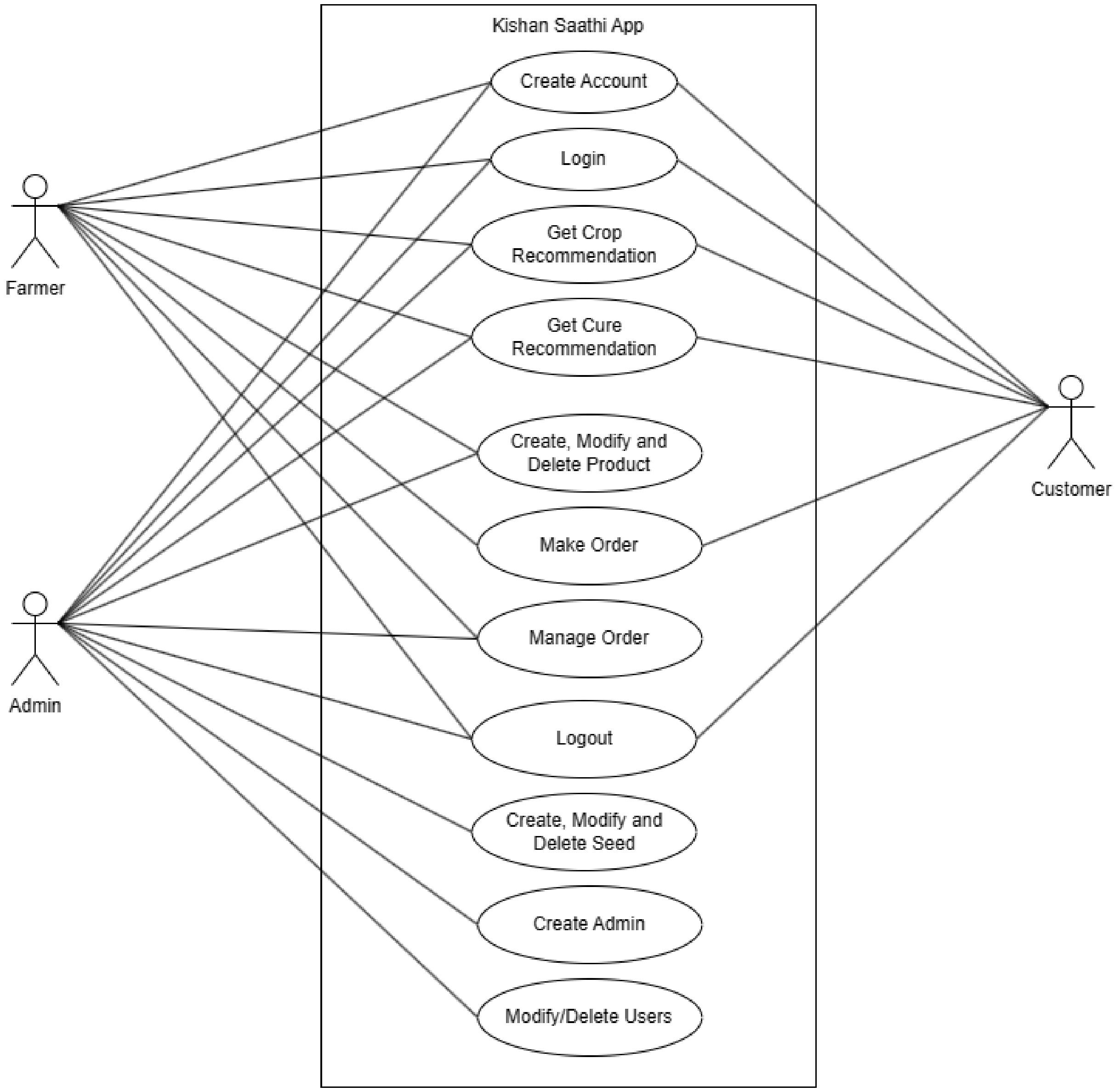


Figure 3.1: Use Case Diagram

- Admin
    1. Use ‘Crop Recommendation System” and “Crop Cure Recommendation System”
    2. Create seeds that is recommended for purchase to any user who obtains prediction of the crop
    3. Modify/Delete the created seed
    4. Create farm products to list in market place from where any user can make an order
    5. Modify/delete products
    6. Create another admin
    7. Modify/Delete users
    8. Manage Orders
- Farmer
    1. Use ‘Crop Recommendation System” and “Crop Cure Recommendation System”
    2. Make order for seeds of crop recommended by crop recommendation system
    3. Create farm products to list in market place from where any user can make an order
    4. Modify/delete products
    5. Modify/Delete users
    6. Manage Orders
- Customer
    1. Create Account
    2. Make orders for seeds or farm products

Hardware & Software Requirements

For this project, laptop with support of HTML5, CSS, and other implementation tools were required. The minimum system requirements for the laptop include: i5 11th generation, 16GB RAM, 512 SSD, Windows 11 OS.

The software requirements for the project includes powerful IDE that is Visual Studio Code, Jupyter Notebook for training models, MongoDB for database, Microsoft edge browser, postman for API testing and many other open source programs. Draw.io and MS Word are also needed for documentation.

Non-Functional Requirements

The non-functional requirements of this software are listed below:

1. Secured
2. Easily accessible
3. Responsive Design
4. User Friendly
5. High Performance

3.2 Feasibility Analysis

A feasibility study is conducted to assess the practicality and viability of implementing the proposed software solution for the farming cycle. This study evaluates various aspects, including technical, economic, operational, and scheduling feasibility, to determine whether the project is feasible and worth pursuing.

1. Technical Feasibility:

   This software solution encompasses complex functionalities such as soil nutrient analysis, crop classification, e-commerce integration, and data management. There was rich set of historical data regarding crop nature based on soil nutrients data. Different classification algorithms is available to train these data. Trained model was integrated into webapp along with developing ecommerce functionality in same web app. So, the project is technically feasible.
2. Economic Feasibility:

   Software requirement for development of this project is available free for community development. However, hardware requirement might be high to setup IOT as NPK sensor for soil nutrient, moisture sensor for moisture measurement in soil, Arduino board, etc. are needed. But as for now this project is just a prototype so all these parameters can be taken through user and weather data through online free APIs like open weather map. So, for prototyping, this project is economically feasible.
3. Operational Feasibility:

   Operational feasibility assessment indicates that the software solution can be effectively implemented within the existing agricultural ecosystem. User acceptance testing and stakeholder engagement activities needs to be done before putting it into practice.

4. Scheduling Feasibility

The project expands over a period of about 4 months. The project schedule is feasible.

Table 3.1: Working Schedule

Tasks	**Start Date**	**End Date**	**Days to complete**
Planning	11 Oct	1 Nov	21
Requirement Analysis	15 Oct	6 Nov	22
Data collection	7 Nov	8 Nov	2
Model Training	8 Nov	13 Dec	35
Backend Development	13 Dec	20 Jan	37
Frontend Development	16 Dec	25 Jan	39
Testing	20 Jan	Feb 2	12
Documentation	Feb 22	Mar 3	9

Chapter 4 - System Design

4.1 Design

Refinement of Class Diagram

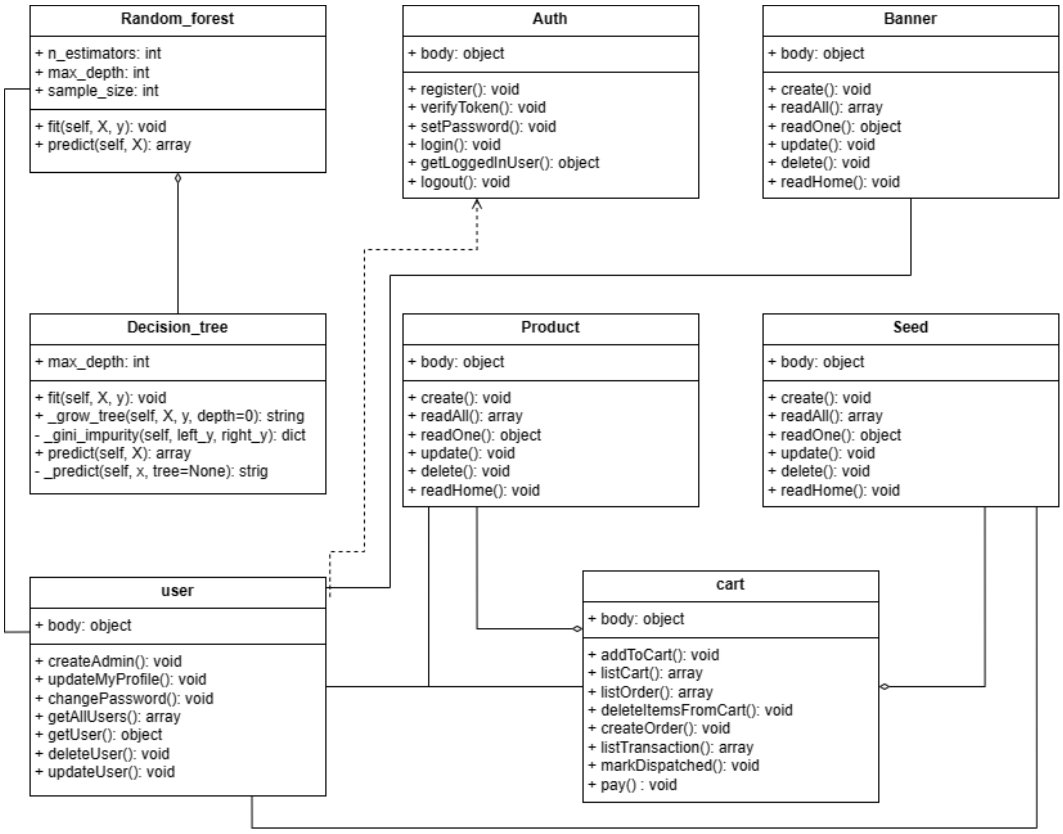


Figure 4.1: Class Diagram

State Diagram for Crop Recommendation System

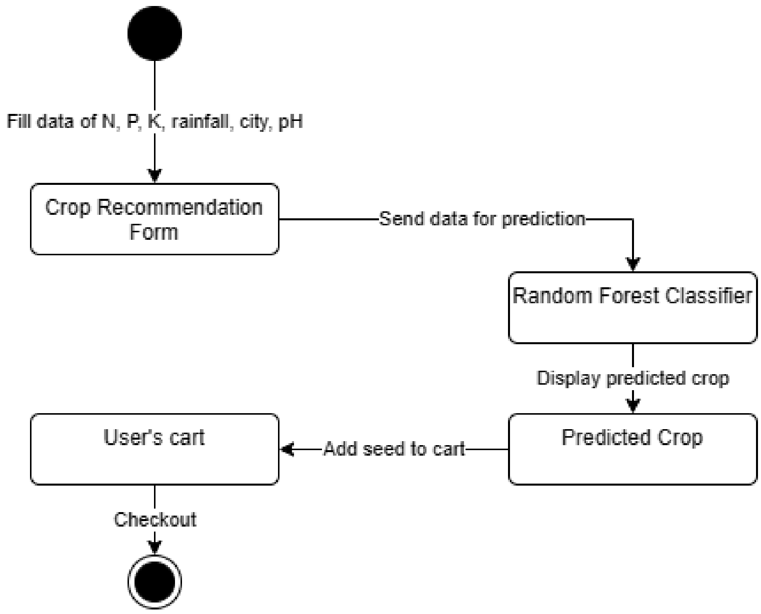


Figure 4.2: Crop Recommendation's State Diagram

Cure Recommendation System’s State Diagram

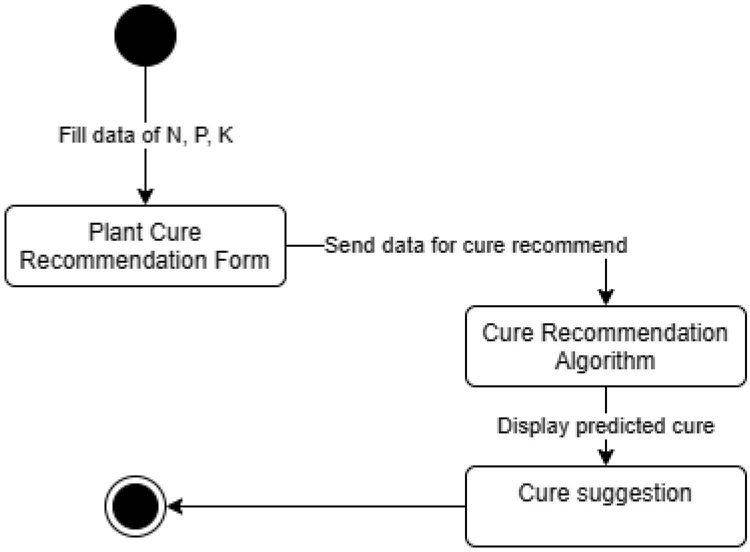


Figure 4.3: Cure Recommendation's State Diagram

Online Marketplace State Diagram

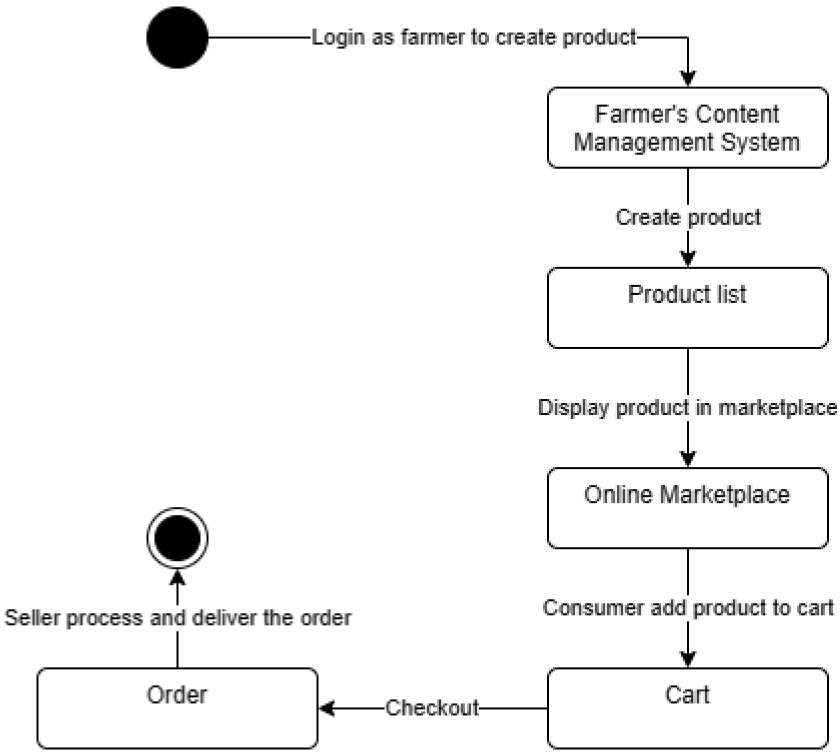


Figure 4.4: Ecommerce's State Diagram

Refinement of Activity Diagram

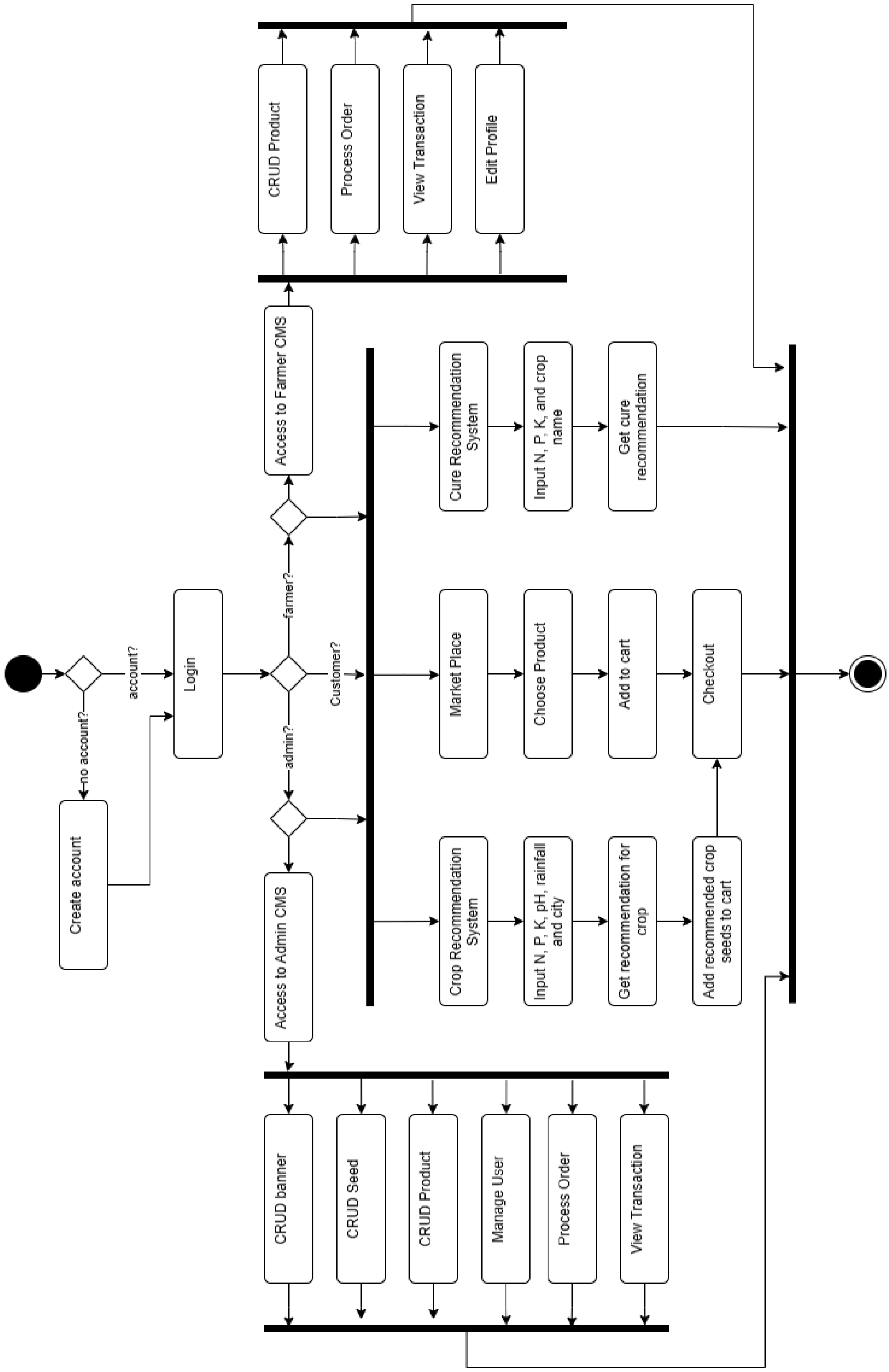


Figure 4.5: Activity Diagram

Refinement of Sequence Diagram

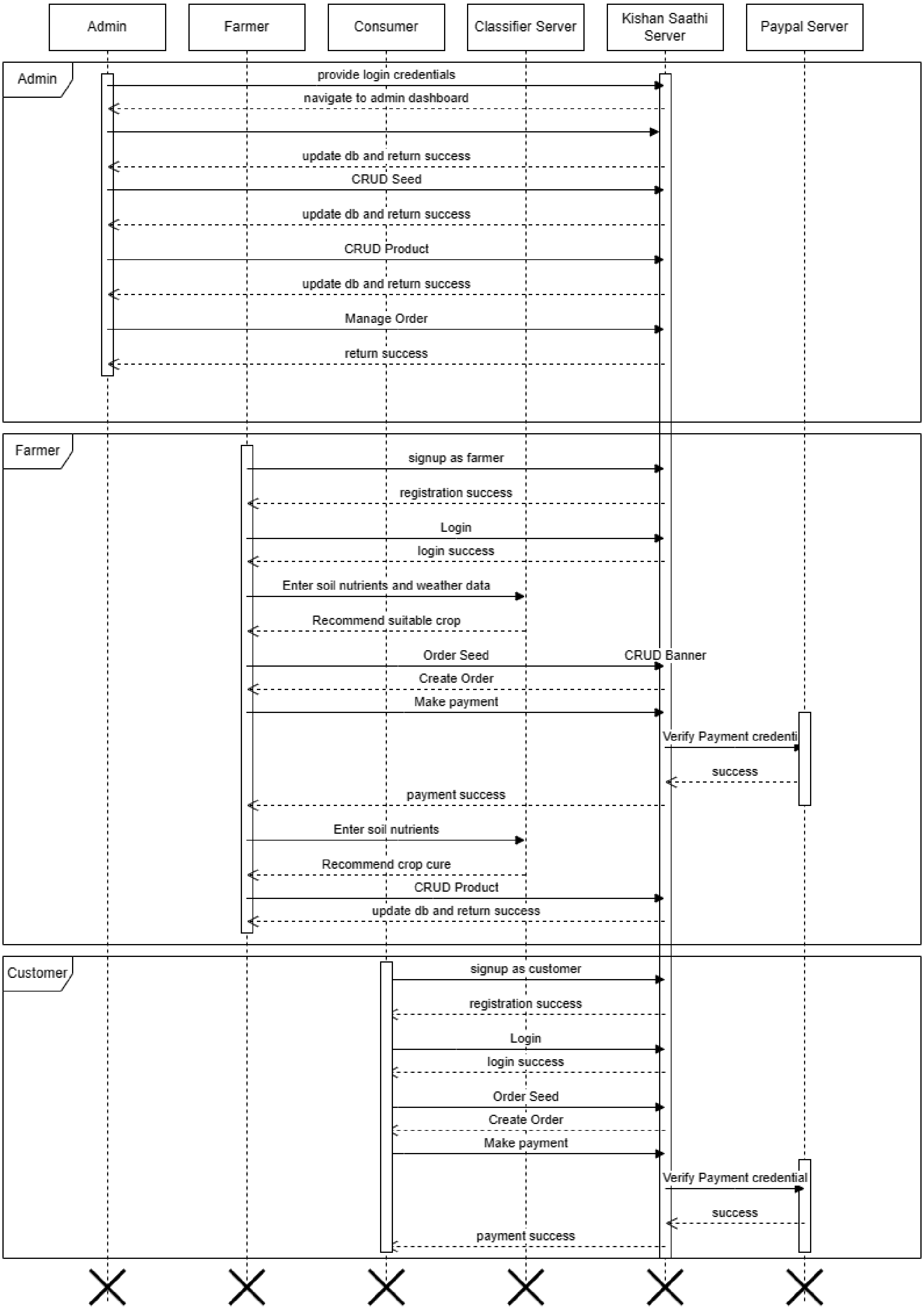


Figure 4.6: Sequence Diagram

Component Diagram

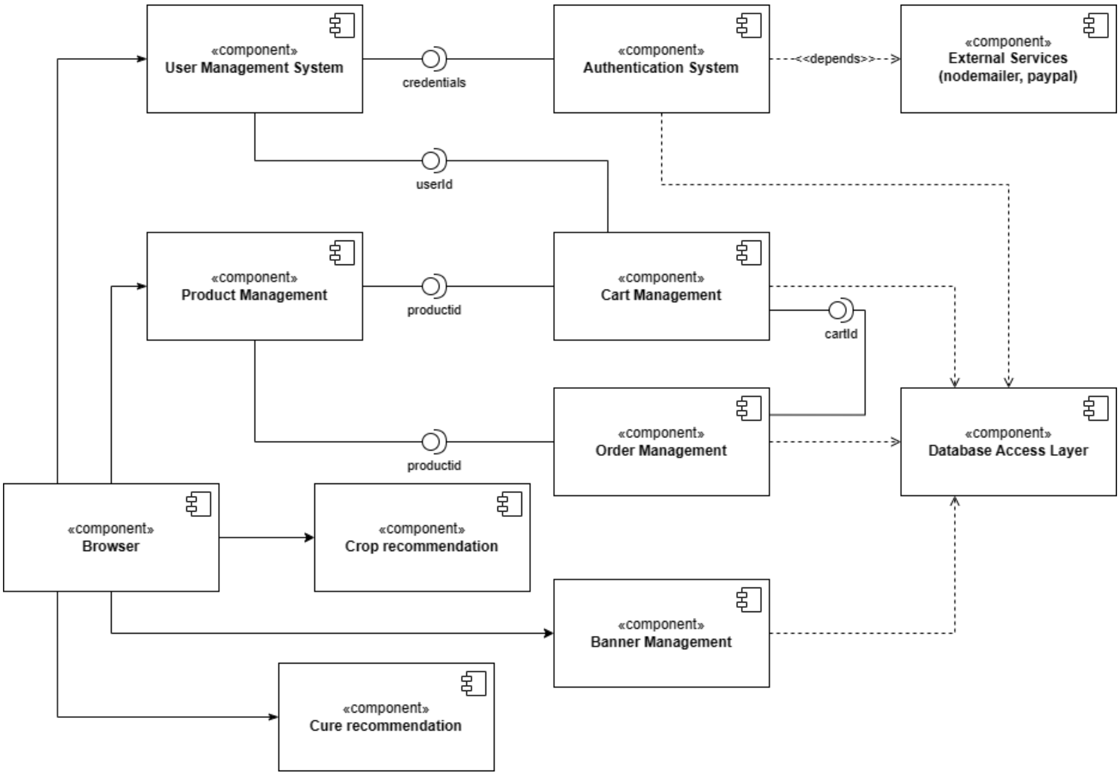


Figure 4.7: Component Diagram

Deployment Diagram

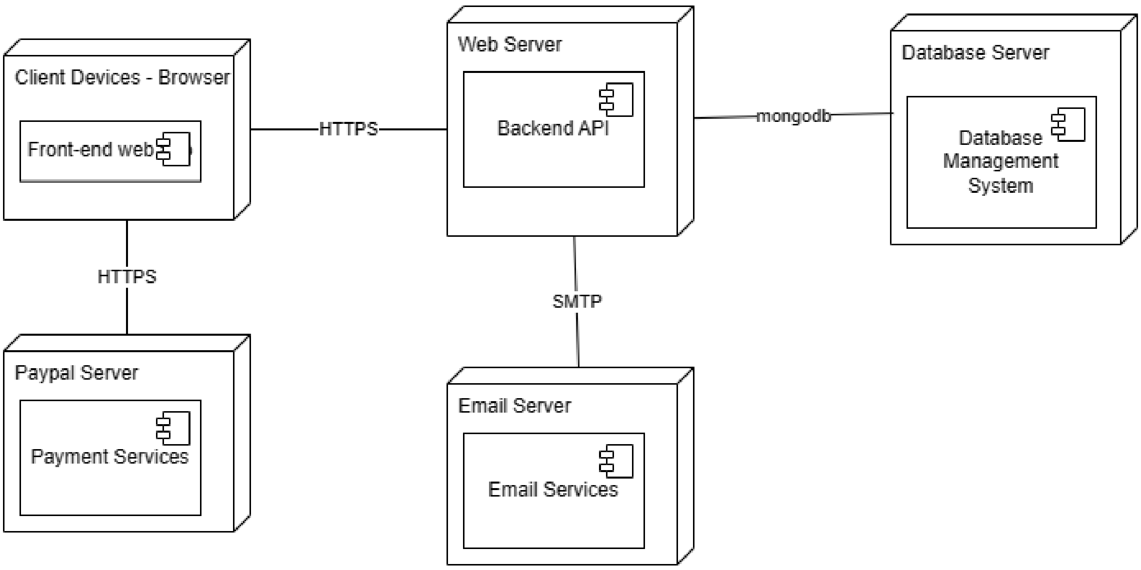


Figure 4.8: Deployment Diagram

4.2 Algorithm Detail

Random Forest Classifier

The Random Forest or Random Decision Forest is a supervised Machine learning algorithm used for classification, regression, and other tasks using decision trees. Random Forests are particularly well-suited for handling large and complex datasets, dealing with high-dimensional feature spaces, and providing insights into feature importance. This algorithm's ability to maintain high predictive accuracy while minimizing overfitting makes it a popular choice across various domains, including finance, healthcare, and image analysis, among others.

The Random Forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is a set of decision trees (DT) from a randomly selected subset of the training set and then it collects the votes from different decision trees to decide the final prediction.

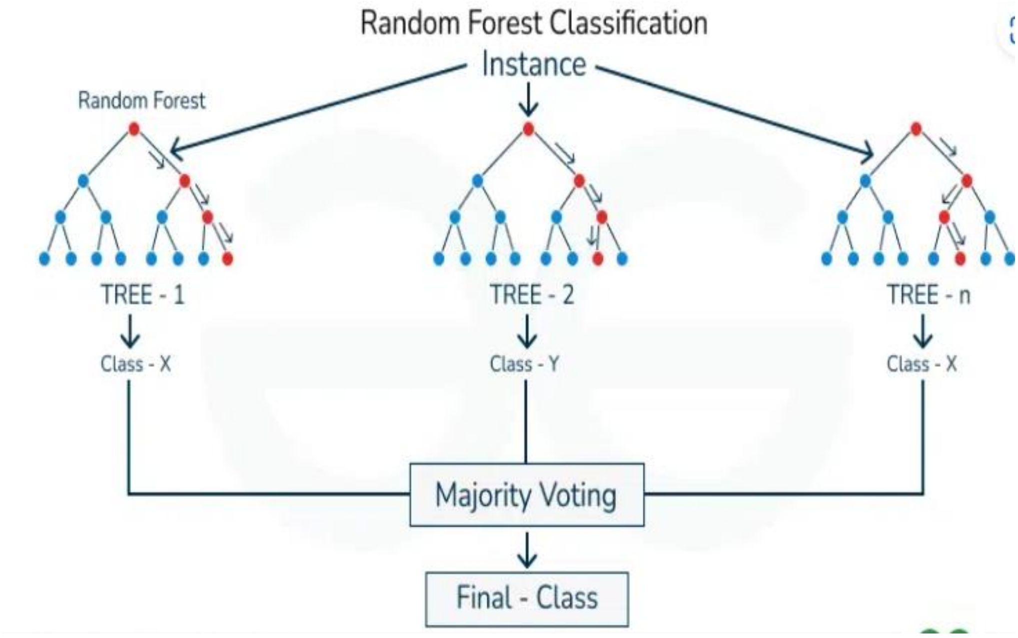


Figure 4.9: Random Forest Classification

Each decision tree in the random forest is constructed using a subset of the training data and a random subset of features introducing diversity among the trees, making the model more robust and less prone to overfitting.

During the training phase, each tree is built by recursively partitioning the data based on the features. At each split, the algorithm selects the best feature from the random subset, optimizing for information gain or Gini impurity. The process continues until a predefined stopping criterion is met, such as reaching a maximum depth or having a minimum number of samples in each leaf node.

For creating Decision Tree, CART Algorithm was used. CART stands for Classification and Regression Tree Algorithm.

CART Algorithm

Classification and Regression Trees (CART) is a decision tree algorithm that is used for both classification and regression tasks. It is a supervised learning algorithm that learns from labelled data to predict unseen data.

Tree structure: CART builds a tree-like structure consisting of nodes and branches. The nodes represent different decision points, and the branches represent the possible outcomes of those decisions. The leaf nodes in the tree contain a predicted class label or value for the target variable.

Splitting criteria: CART uses a greedy approach to split the data at each node. It evaluates all possible splits and selects the one that best reduces the impurity of the resulting subsets. For classification tasks, CART uses Gini impurity as the splitting criterion. The lower the Gini impurity, the more pure the subset is. For regression tasks, CART uses residual reduction as the splitting criterion. The lower the residual reduction, the better the fit of the model to the data.

Pruning: To prevent overfitting of the data, pruning is a technique used to remove the nodes that contribute little to the model accuracy. Cost complexity pruning and information gain pruning are two popular pruning techniques. Cost complexity pruning involves calculating the cost of each node and removing nodes that have a negative cost. Information gain pruning involves calculating the information gain of each node and removing nodes that have a low information gain.

The CART algorithm works via the following process:

- The best-split point of each input is obtained.
- Based on the best-split points of each input in Step 1, the new “best” split point is identified.
- Split the chosen input according to the “best” split point.

- Continue splitting until a stopping rule is satisfied or no further desirable splitting is available.

CART algorithm uses Gini Impurity to split the dataset into a decision tree. It does that by searching for the best homogeneity for the sub nodes, with the help of the Gini index criterion.

Complete algorithm of crop recommendation system:

1. Input the data for preprocessing
2. Encode the label data
3. Split the data into train dataset and test dataset
4. Implement Random Forest classification algorithm:
    a. Initialize Random Forest with number of trees (*n_estimators*)
    b. Train the data:
        i. Initialize an empty list '*self.estimators*' to store decision trees. Also initialize other attributes with provided values that is *n_estimators*, *max_depth*, and *sample_size*
        ii. Iterate self.n_estimators times to create a new decision tree for each iteration. Decision Tree was created with specified maximum depth('*max_depth*') using CART algorithm. (The implementation detail of CART algorithm is given in next list.)
        iii. If *sample_size* is provided (not None), random sampling is performed to select a subset of the training data (X and y). It helps in reducing overfitting by creating diversity among decision trees. If *sample_size* is not provided, entire training set is used.
        iv. The selected subset of training data is used to train the decision tree.
        v. After training, the trained decision tree is added to the list of estimators (*self.estimators*). This list will contain all the decision trees trained during the random forest fitting process.
        vi. After all the decision trees have been trained and added to the list, the training is completed.
    c. Save the model
    d. Perform testing by feeding testing dataset as input

e. Decode the output which will be predicted result

System Flow Diagram

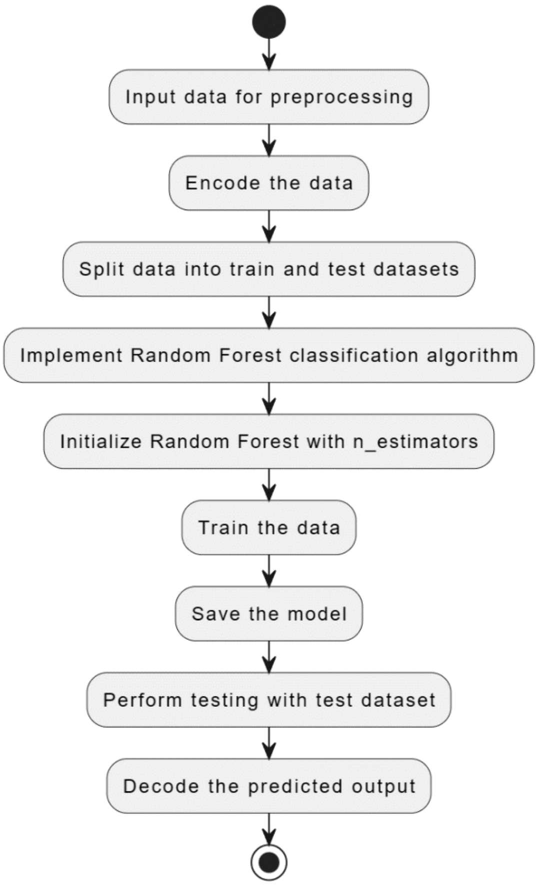


Figure 4.10: Crop Recommendation System Flow Diagram

The step for generating decision tree with CART algorithm are as follows:

1. From above algorithm of random forest, *max_depth* is supplied to create a decision tree in CART Algorithm. It determines the max depth of the tree. If it is not supplied, the tree can grow until certain stopping criteria are met.
2. CART Algorithm takes input features X and corresponding target labels y.
3. *_grow_tree()* method is called recursively to build the decision tree
4. At each step, *_grow_tree()* method checks if stopping criteria are met:
    a. If the maximum depth is reached (*self.max_depth*) or if all samples belong to the same class, stop splitting and return a leaf node containing the majority class label and some metadata.
    b. If stopping criteria are not met, it proceeds to find the best split:
        i. Iterate through each feature and each unique value within that feature.

ii. Calculate the Gini impurity for the split based on this threshold.

$$Gini(y) = 1 - \sum_{i=1}^{n} p_i^2$$

Where:

n is the number of classes.

p_i is the probability of randomly picking an element of class i from the set y.

iii. Choose the split with the lowest Gini impurity as the best split

c. If no split decreases impurity (indicating no meaningful split), return a leaf node with the majority class label and metadata.

5. Once the best split is found, split the dataset into left and right subsets based on the chosen feature and threshold. Recursively call *_grow_tree* on both subsets.

CART Algorithm Flow Diagram

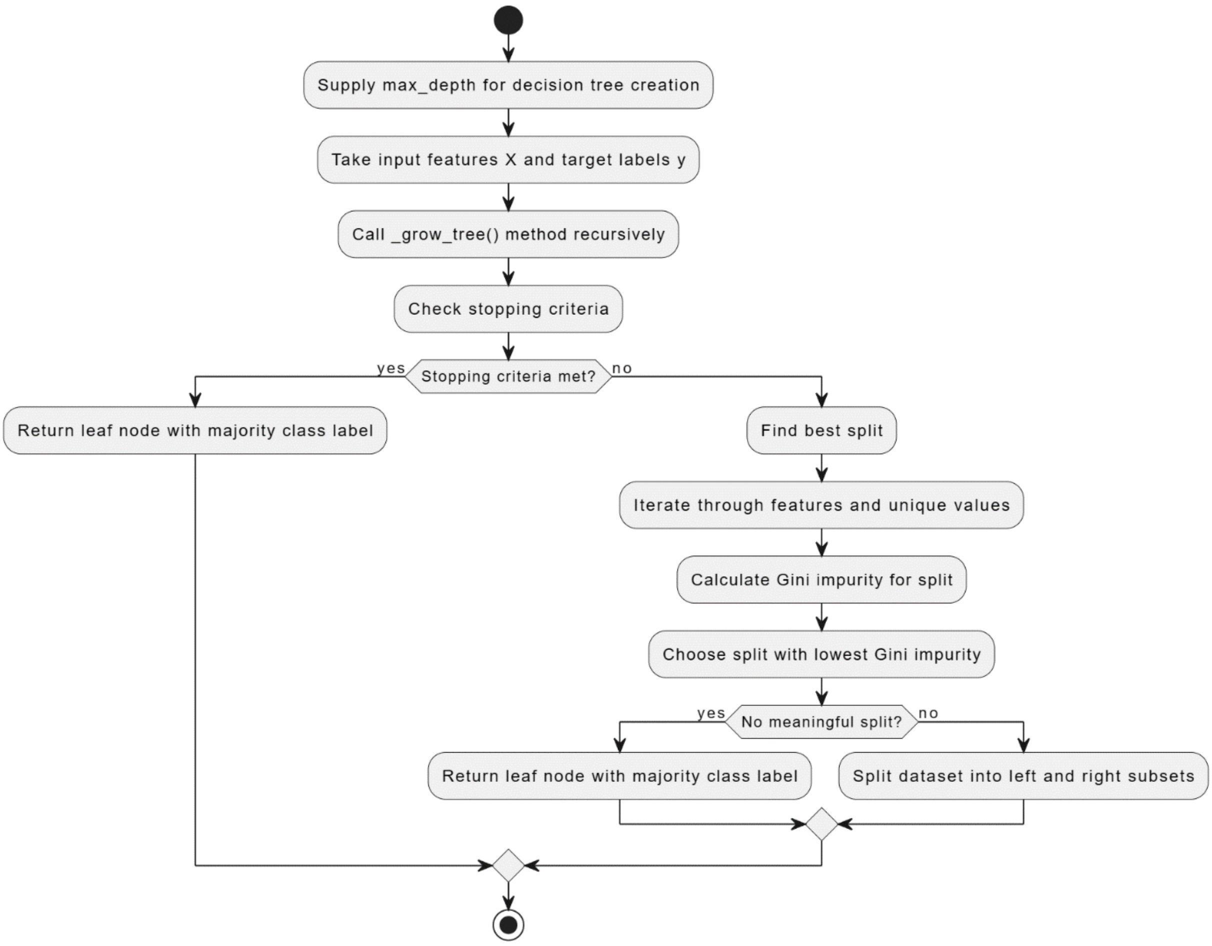


Figure 4.11: CART Algorithm Flow Diagram

Chapter 5 Implementation and Testing

5.1 Data Collection

For crop recommendation system, the dataset was taken from Kaggle [8]. Dataset contains 2200 rows and 8 columns with no missing values. First 7 columns contain numeric data and last column contains label data. The 8 data fields are:

- N - ratio of Nitrogen content in soil
- P - ratio of Phosphorous content in soil
- K - ratio of Potassium content in soil
- temperature - temperature in degree Celsius
- humidity - relative humidity in %
- ph - ph value of the soil
- rainfall - rainfall in mm
- label – crop name

Following is the looks of few data:

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

Figure 5.1: Snapshot of dataset head

Shape of the data:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   N            2200 non-null   int64
 1   P            2200 non-null   int64
 2   K            2200 non-null   int64
 3   temperature  2200 non-null   float6
 4   humidity     2200 non-null   float6
 5   ph           2200 non-null   float6
 6   rainfall     2200 non-null   float6
 7   label        2200 non-null   object
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB
```

Figure 5.2: Shape of data

5.2 Implementation

5.2.1 Tools Used

Table 5.1: Tools Used

Tools	Specifications
Programming Language	Python, JavaScript
IDE	VS Code, Jupyter Notebook
Frontend	React JS, React Bootstrap
Backend	Node JS, Express JS, Flask
Database	MongoDB
Diagrams	Draw.io
Documentation	MS Word

5.2.2 Implementation Details of Modules

The system is broken down into three major modules:

A. **Crop Recommendation Module**

Crop Recommendation Module was developed as flask application using Python. It uses classifier trained in Python Jupyter Notebook. The API was integrated in frontend application build over React JS library.

B. **Cure Recommendation Module**

Cure Recommendation Module was also developed as flask application using Python. The API was integrated in frontend application build over React JS library.

C. **E-Commerce Module**

E-commerce module was developed as Node application using Express Framework. This module consists several different modules:

a. Auth Module
b. Banner Module
c. Seed Module
d. Product Module
e. Cart Module
f. User Module

The API was integrated in frontend application build over React JS library.

5.2 Testing

5.2.1 Test Cases for Unit Testing

Table 5.2: Unit Testing of Authentication and Authorization

Test	Description	Test Step	Expected Result	Status
TA1	User Registration test	Go to sign up page, fill registration form	After filling registration form, user will get verification link at specified email	pass
TA2	Verification	Check verification email and click on the link provided to verify	If verified, user will redirect to password set page	pass
TA3	Set Password	In password set page, create password. Only password with Uppercase, Lowercase, Number, Symbol and at least 8 characters are allowed.	Password should be stored in database in encrypted form and user should be redirect into login page.	pass
TA4	Login	Provide login credential in login page.	For Admin, redirect in ADMIN CMS. For Farmer, redirect in FARMER CMS. For Consumer, redirect in HOMEPAGE.	pass
TA5	Permission Test	Browse Admin Dashboard URL, while login as User	Block Admin Dashboard and redirect in homepage	pass
TA6	Logout	Click on logout	User should be logged out	pass

Table 5.3: Unit test of Banner Management

Test	Description	Test Step	Expected Result	Status
TB1	Banner Create	Go to Banner management in sidebar of admin panel. Click on create and provide necessary information to create banner.	Banner should be created and displayed in homepage.	pass
TB2	Banner List	Go to Banner Management page	There should be list of all the banners created by logged in user	pass
TB3	Banner Update	Click on the yellow pencil icon button at the last column of banner listing and edit the necessary information	The edited information should be reflected.	pass
TB4	Banner Delete	Click on the red trash icon button at the last column of banner listing	Confirmation dialog should appear and upon confirming the related banner data should be deleted from database	pass

Table 5.4: Unit Testing of Product module

Test	Description	Test Step	Expected Result	Status
TP1	Product Create	Go to Product management in sidebar of admin panel. Click on create and provide necessary information to create product.	Product should be created and displayed in Marketplace.	pass

TP2	Product List	Go to Product Management page	There should be list of all the products created by logged in user	pass
TP3	Product Update	Click on the yellow pencil icon button at the last column of product listing and edit the necessary information	The edited information should be reflected.	pass
TP4	Product Delete	Click on the red trash icon button at the last column of product listing	Confirmation dialog should appear and upon confirming the related product data should be deleted from database	pass

Table 5.5: Unit Testing of Seed module

Test	Description	Test Step	Expected Result	Status
TS1	Seed Create	Go to Seed management in sidebar of admin panel. Click on create and provide necessary information to create product.	Seed should be created and displayed as product to add on cart when predicted by crop recommendation system.	pass
TS2	Seed List	Go to Seed Management page	There should be list of all the seeds created by logged in user.	pass
TS3	Seed Update	Click on the yellow pencil icon button at the last column of seed listing and edit the necessary information.	The edited information should be reflected.	pass

TS4	Seed Delete	Click on the red trash icon button at the last column of seed listing.	Confirmation dialog should appear and upon confirming the related seed data should be deleted from database.	pass

Table 5.6: Unit testing of User Module

Test	Description	Test Step	Expected Result	Status
TU1	Create Admin	Login as admin and go to user management. Click on create admin to create a new admin.	New user with role admin should be created.	pass
TU2	User List	Go to User Management page	There should be list of all the Users.	pass
TU3	User Update	Click on the yellow pencil icon button at the last column of user listing and edit the necessary information.	The edited information should be reflected.	pass
TU4	User Delete	Click on the red trash icon button at the last column of user listing.	Confirmation dialog should appear and upon confirming the related user data should be deleted from database.	pass

Table 5.7: Unit testing of cart module

Test	Description	Test Step	Expected Result	Status
TAC1	Product Added to cart	In Marketplace, click add to cart button of any product	A modal for selecting quantity will popped and the product should be added to cart afterward.	pass

TAC2	Product checkout	In cart, select all the product to checkout, enter the delivery address and click checkout	Total should be displayed, and after clicking checkout order should be updated in seller's profile	pass

Table 5.8: Unit testing of order module

Test	Description	Test Step	Expected Result	Status
TO1	Order Listing	Go to order management page	All the new order should be listed	pass
TO2	Mark Dispatched	At the last column, click the mark dispatched button	The order should be completed and completed order should be displayed in transaction page	pass

Table 5.9: Unit testing of transaction module

Test	Description	Test Step	Expected Result	Status
TT1	Transaction Listing	Go to transaction page	All the completed order should be displayed	pass

5.2.2 Test Cases for System Testing

To test the proper functioning of the system, following test cases are taken:

1. Admin user was logged in and created new banner, seeds, product and new admin account from default admin account. Changes is expected to reflect.

   Following is the result. Few Banners, Seeds, Products and new admin is created from admin CMS.

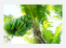
   Banner:.

Title	Link	Image	Status
Banner 3	https://www.ashish.com		active
Banner 2	https://www.ashish.com		active
Banner 1	https://www.ashish.com		active

Seed:

Seed	Price	Image	Status
Watermelon seed	60(-2% off) = 58.8		active
Rice seed	60(-2% off) = 58.8		active
Promegranate Seed	60(-2% off) = 58.8		active
Pigeonpea seed	60(-2% off) = 58.8		active

Product:

Product	Price	Image	Status
Banana	150(-5% off) = 142.5		active
Apple	500(-7% off) = 465		active

New Admin:

Name	Role	Email	Image	Status
Ashish admin	admin	ashish@admin.com		active

Figure 5.3: Testing of Admin CMS

2. A new farmer user is signed up with the required credentials. Then the user first uses the crop recommendation system. A random data will be provided to the crop recommendation form. To get the crop recommendation. User must be able to order the recommended crop from the recommendation. The order must be shown in admin panel. Then farmer user will test cure recommendation system. A test data will be fed and the result for curing the crop should be obtained. From farmer CMS, few products were created, which must be displayed in Farmer Marketplace.

   Result:

   New Farmer account:


```
_id: ObjectId('65fbdd2c5d671f7478414108')
name : "Ashish Farmer"
email : "ashish@farmer.com"
role : "farmer"
token : null
image : "1711004972545_8930.jpg"
status : "active"
password : "$2a$10$AMmoQhjbaWMjJ/BNbTvi7eSLifM.zb9KL6.PFY5XKWUf8JlDmKBm."
resetToken : null
resetExpiry : null
createdAt : 2024-03-21T07:09:32.558+00:00
updatedAt : 2024-03-21T07:10:03.753+00:00
__v : 0
```


Figure 5.4: User Data in Database

Crop recommendation:

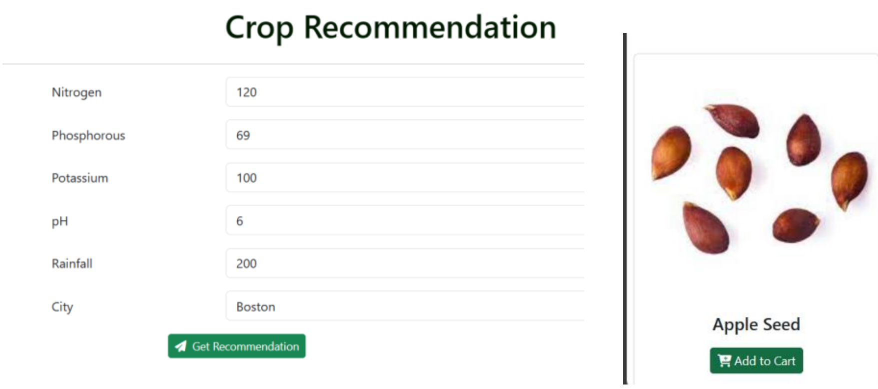
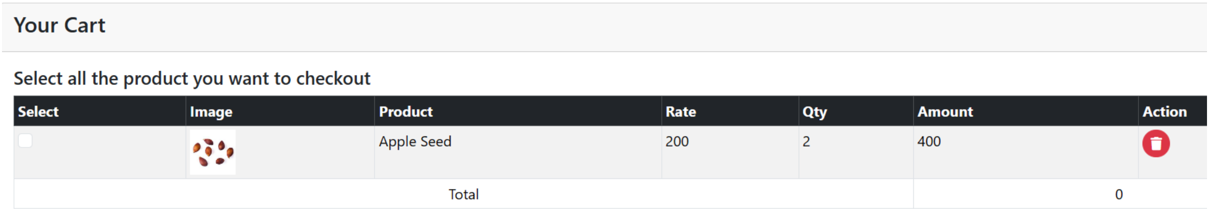
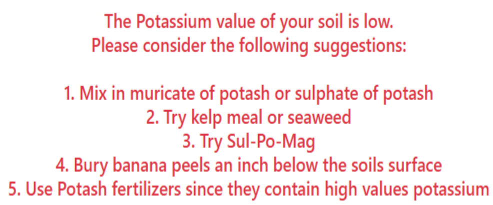


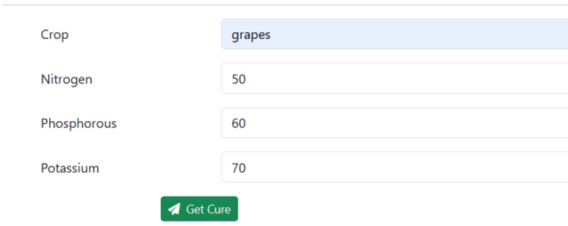
Figure 5.5: Testing of crop recommendation module



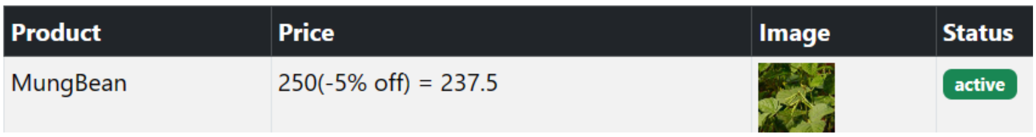
Select	Image	Product	Rate	Qty	Amount	Action
		Apple Seed	200	2	400	
		Total			0	

Cure recommendation:



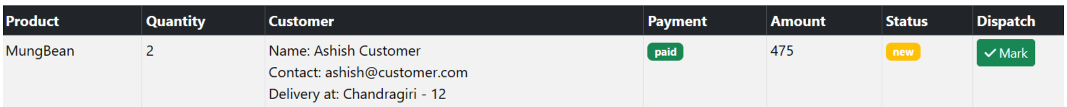


Product created in farmer account



Product	Price	Image	Status
MungBean	250(-5% off) = 237.5		active

Order list:



Product	Quantity	Customer	Payment	Amount	Status	Dispatch
MungBean	2	Name: Ashish Customer Contact: ashish@customer.com Delivery at: Chandragiri - 12	paid	475	new	Mark

Figure 5.6: Testing for farmer CMS

3. A new customer user is signed up with the required credentials. Then the customer user will navigate to farmer market place. Customer user must be able to place order. The order should be updated in farmer's panel.

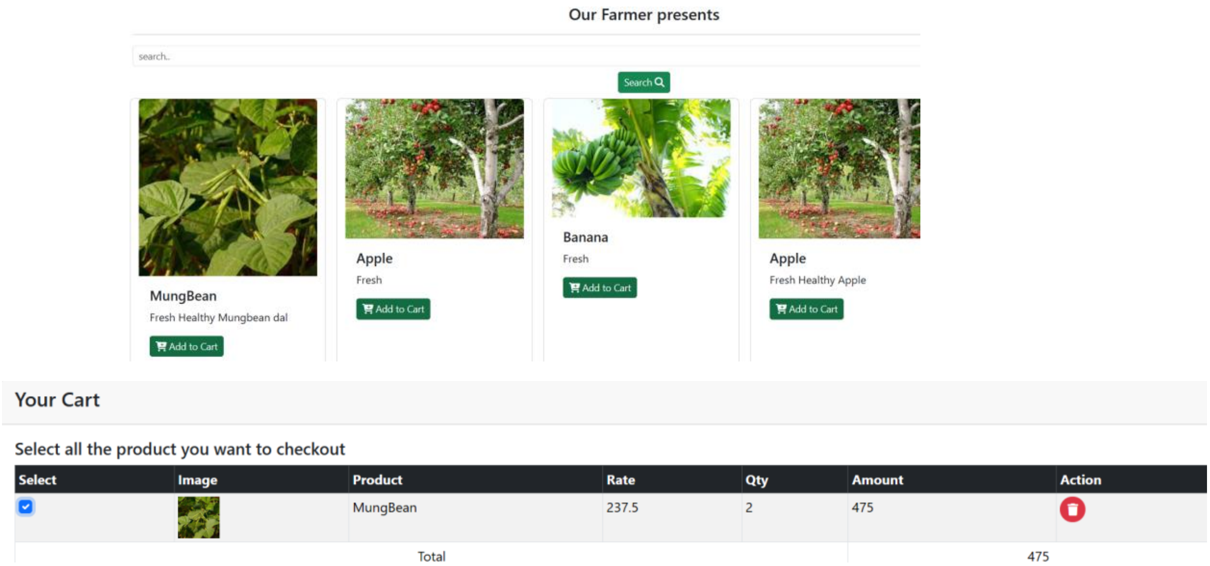


Figure 5.7: Testing of customer

4. Additionally, paypal integration is also tested while ordering.

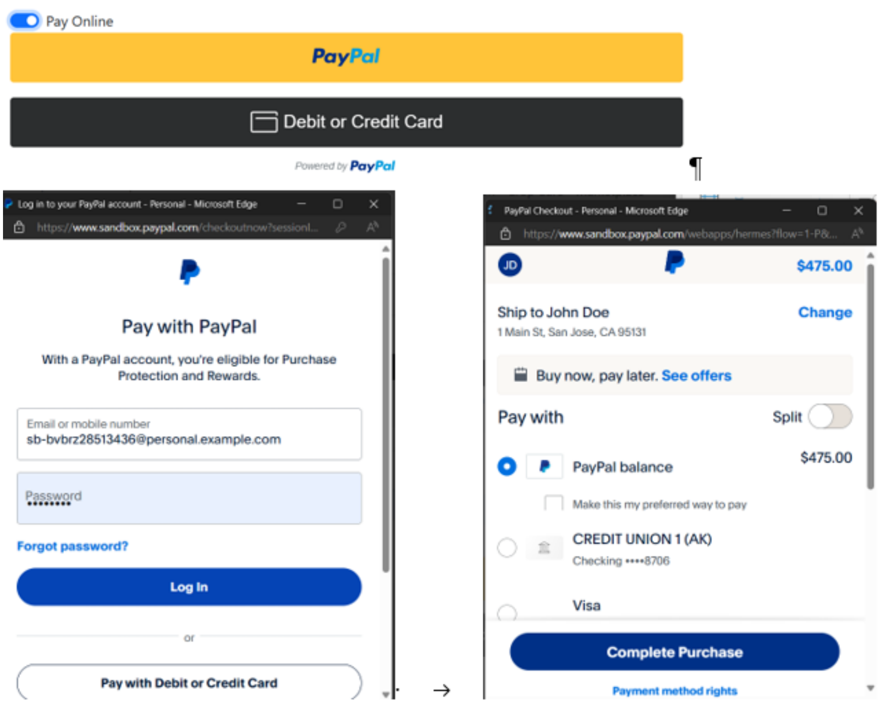


Figure 5.8: Paypal testing

5.3 Result Analysis

To evaluate the performance of the model, K-fold cross validation algorithm was used where the data was divided 8 parts i.e. 8 fold cross validation.

The accuracy of the model was calculated by:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

```python
def accuracy_score(y_true, y_pred):
    correct_predictions = np.sum(y_true == y_pred)
    total_predictions = len(y_true)
    accuracy = correct_predictions / total_predictions
    return accuracy
```

The accuracy rate of the model for 10% train data and 90% test data was 96.26%.

```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.8,
random_state=42)
```

```python
print(f" accuracy : {accuracy_score(y_test,ypred)}")
```

```
accuracy : 0.9626262626262626
```

Likewise, the accuracy for 20% train, 80% test data was 98.69%. The accuracy for 30% train, 70% test data was 98.70%. The accuracy for 40% train, 60% test data was 98.93%. The accuracy for 50% train, 50% test data was 99.45%. The accuracy for 60% train, 40% test data was 99.31%. The accuracy for 70% train, 30% test data was 99.09%. The accuracy for 80% train, 20% test data was 99.3%.

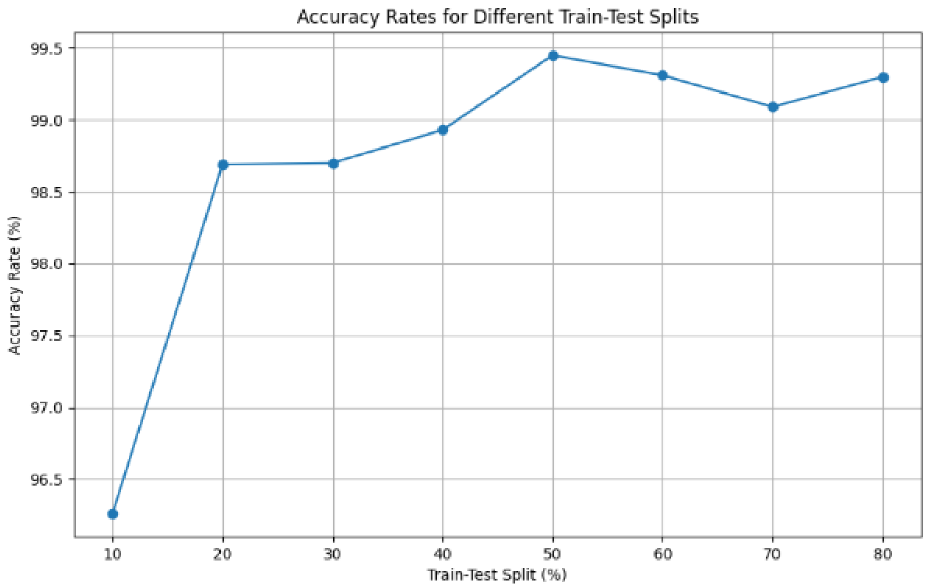


Figure 5.9: Accuracy Rates for Different Train-Test Splits

The accuracy of the crop recommendation algorithm consistently improves as the proportion of training data increases. The model achieves exceptionally high accuracy rates, with the highest accuracy of 99.45% obtained when using a 50-50 split between training and testing data. Even with smaller training proportions (e.g., 10% or 20%), the algorithm demonstrates robust performance, indicating its effectiveness in predicting suitable crops across different scenarios. The results suggest that the algorithm can provide reliable crop recommendations with high accuracy, making it a valuable tool for farmers and agricultural professionals.

Chapter 6 Conclusion and Future Recommendation

6.1 Conclusion

The development of the proposed software solution for the farming cycle represents a significant step towards revolutionizing agricultural practices and empowering farmers with data-driven decision-making tools. By seamlessly integrating soil analysis, crop selection, procurement, plant care recommendations, and direct-to-consumer marketing functionalities, the software addresses key challenges faced by farmers and enhances productivity, sustainability, and market access within the agricultural sector.

Through the implementation of advanced algorithms such as the Random Forest classification algorithm, the software effectively predicts suitable crops based on soil nutrients and climatic conditions, enabling farmers to make informed decisions regarding crop selection. The direct online ordering feature further streamlines the procurement process, providing farmers with convenient access to quality seeds tailored to their specific requirements.

Additionally, the software's plant care recommendation module offers valuable guidance to farmers throughout the cultivation process, facilitating optimal fertilization and pest management strategies based on soil nutrient levels and crop type. The integration of e-commerce capabilities enables farmers to sell their produce directly to consumers, eliminating intermediaries and fostering stronger connections between producers and consumers.

6.2 Future Recommendation

In future, the crop recommendation system and cure recommendation system can be automated with used of sensors. NPK sensor is available in the market which measures Nitrogen, Phosphorous and Potassium content of the soil. Soil moisture sensor with pH indicator is available in the market to measure humidity and pH value of the soil. Other data can be fetched from different API. The wi-fi module can be attach in Arduino board along with other sensor to get real time data from the field and get real time recommendation without the need of inputting soil data by user. This will not only made the process easier but also provide real time data.

Other future recommendations are as follows:

a. Enhanced Data Analytics: Continuously improve the software's data analytics capabilities by incorporating additional factors such as soil texture, crop history, and environmental factors to further enhance crop prediction accuracy and recommendation precision.
b. Mobile Application Development: Develop a mobile application version of the software to provide farmers with on-the-go access to essential features and functionalities, enhancing usability and convenience.
c. Expansion of Market Connectivity: Explore opportunities to expand the software's market connectivity beyond direct-to-consumer sales, such as partnerships with local markets, restaurants, and food delivery services to broaden the reach and impact of farmers' produce.
d. Integration with IoT Devices: Incorporate Internet of Things (IoT) devices such as soil sensors, weather stations, and drones to enable real-time data collection and monitoring, facilitating proactive decision-making and resource optimization.
e. Community Engagement and Training: Organize workshops, training sessions, and community outreach programs to educate farmers on the benefits and usage of the software, fostering adoption and maximizing its impact on agricultural communities.
f. Continuous Improvement and Updates: Implement a feedback mechanism to gather user feedback and suggestions for software improvements, ensuring ongoing refinement and enhancement of features based on user needs and evolving industry trends.

References

[1] 2019 IEEE "Smart Management of Crop Cultivation using IoT and Machine Learning" Archana Gupta, Dharmil Nagda, Pratiksha Nikhare, Atharva Sandbhor

[2] "Crop Recommendation on Analyzing Soil Using Machine Learning" Anguraj.Ka, Thiyaneswaran.Bb, Megashree.Gc, Preetha Shri.J.Gd, Navya.Se, Jayanthi. Jf, 2020.

[3] "Classification of Soil and Crop Suggestion using Machine Learning Techniques", A. Mythili, IEEE 2019.

[4] "IOT based Crop Recommendation, Crop Disease Prediction and Its Solution" Rani Holambe, Pooja Patil, Padmaja Pawar, Saurabh Salunkhe, Mr. Hrushikesh Joshi, 2019 IRJET

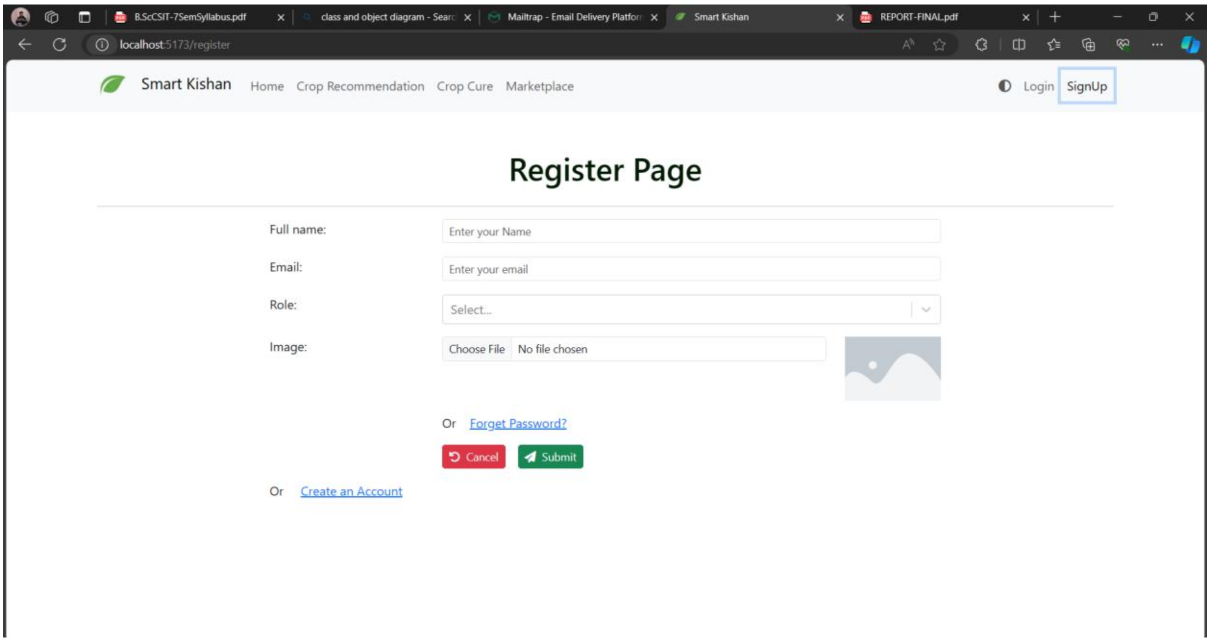
[5] Radhika, Narendiran, "Kind of Crops and Small Plants Prediction using IoT with Machine Learning," International Journal of Computer & Mathematical Sciences, 2018.

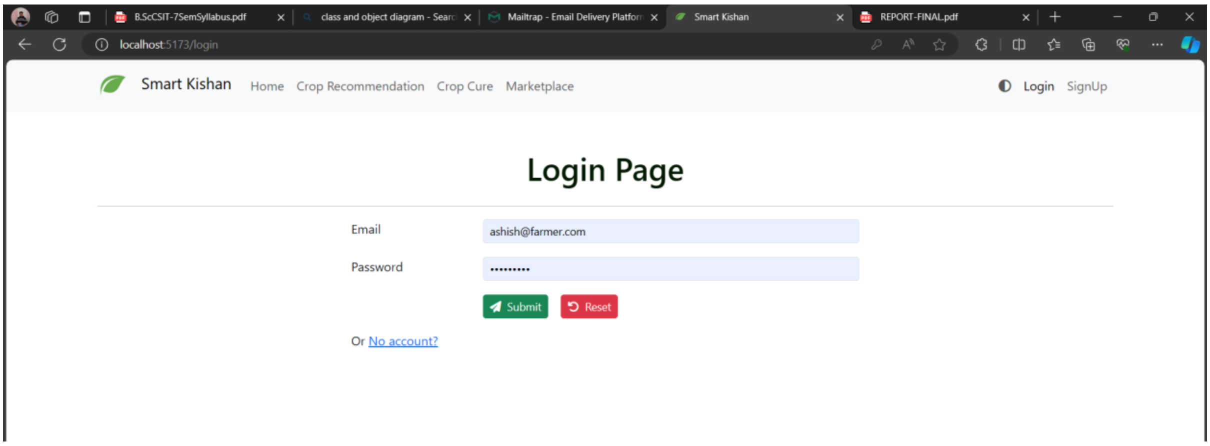
[6] Mehta, P., Shah, H., Kori, V., Vikani, V., Shukla, S., & Shenoy, M.,2018. "Survey of unsupervised machine learning algorithms on precision agricultural data", IEEE

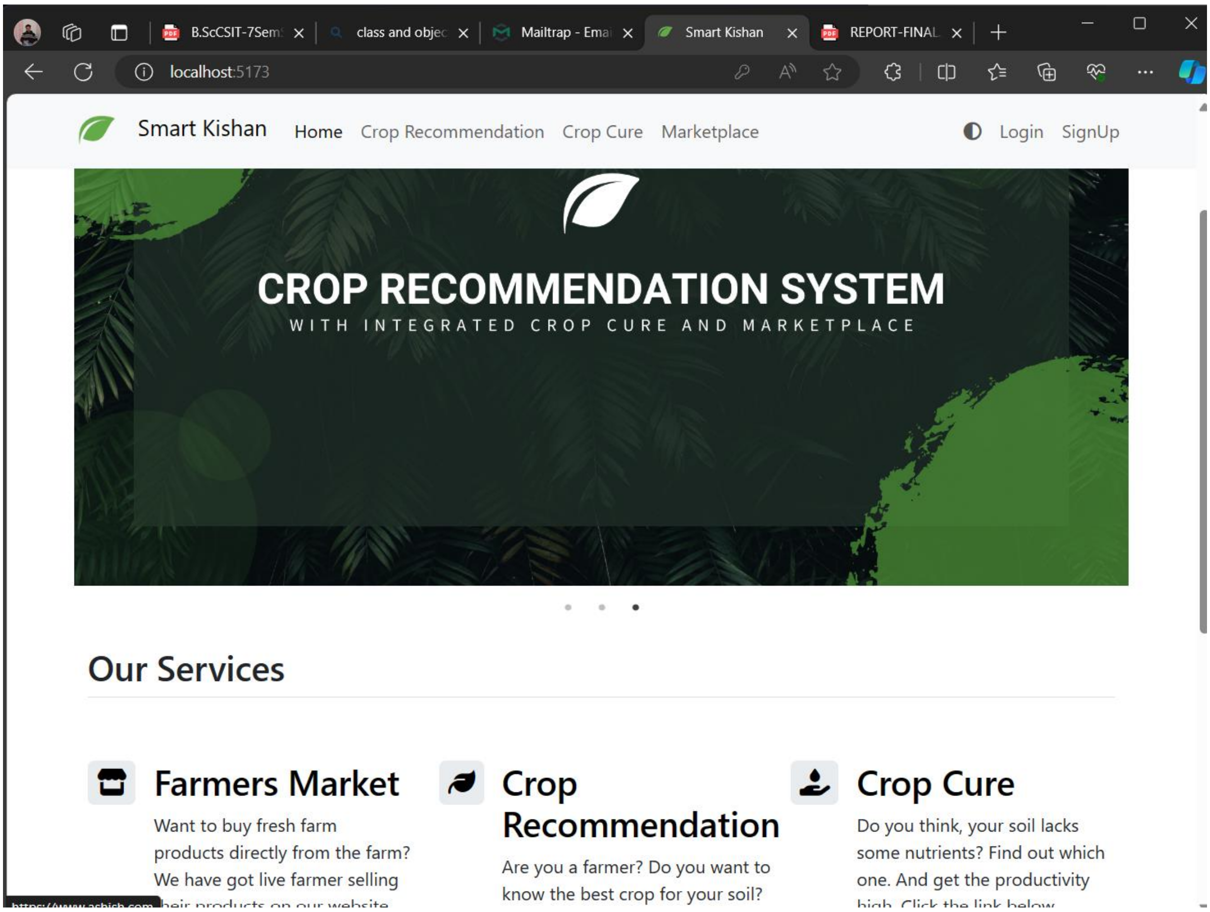
[7] "IOT based Crop Recommendation, Crop Disease Prediction and Its Solution" Rani Holambe, Pooja Patil, Padmaja Pawar, Saurabh Salunkhe , Mr. Hrushikesh Joshi, 2019 IRJET

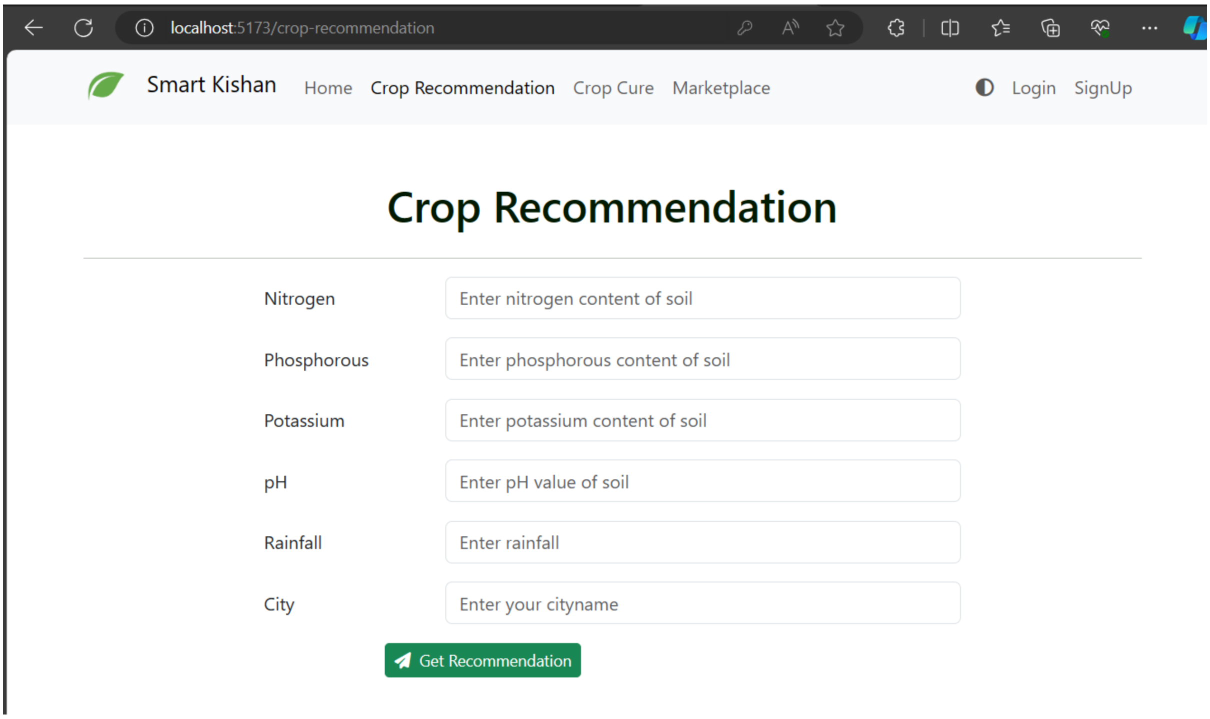
[8] Atharva Ingle, "Crop Recommendation Dataset" [Online]. Available: https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset/data. [Accessed 09 December 2023].

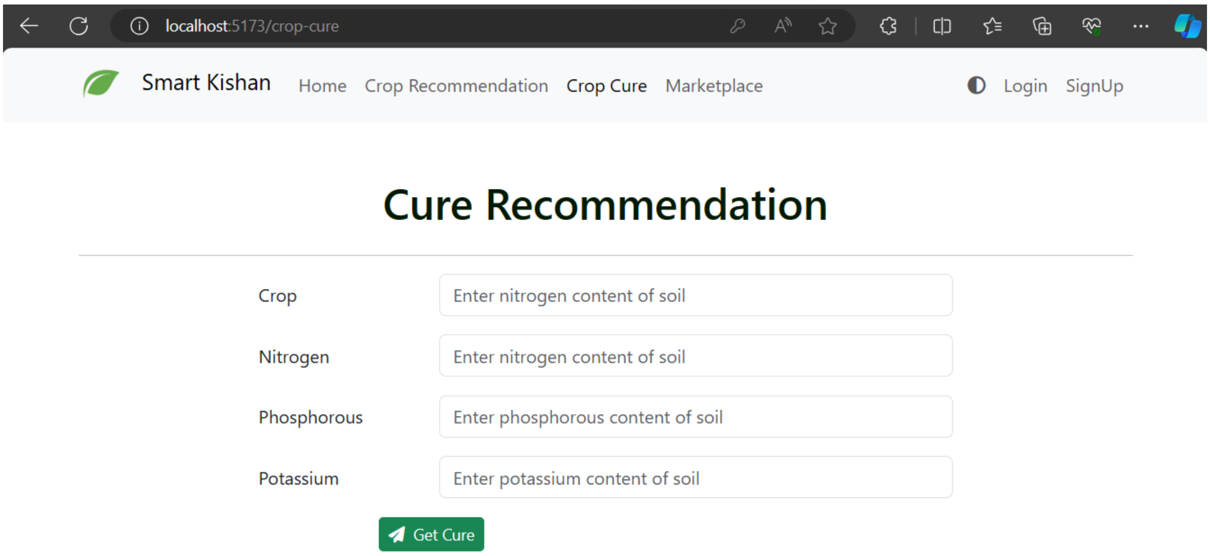
Appendices

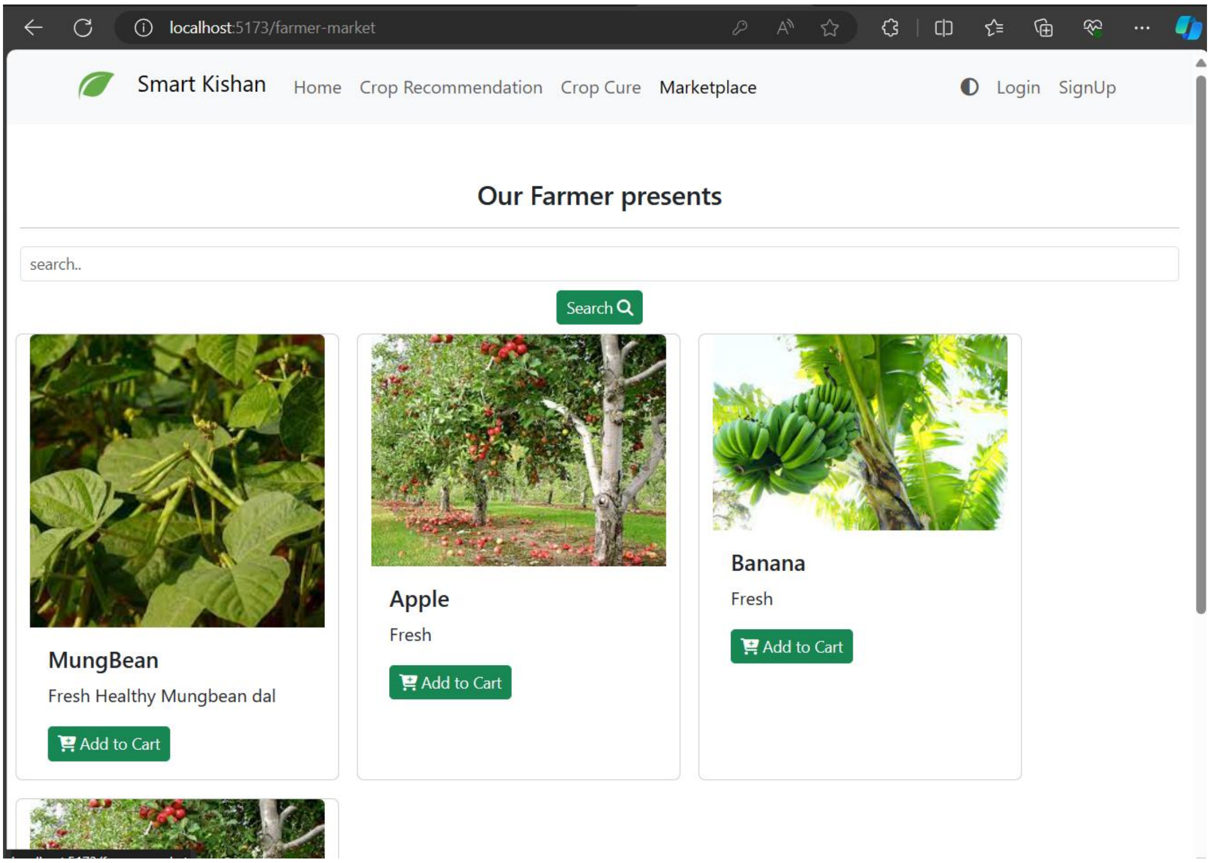

localhost:5173/register
Smart Kishan
Home
Crop Recommendation
Crop Cure
Marketplace
Login
SignUp
Register Page
Full name:
Enter your Name
Email:
Enter your email
Role:
Select...
Image:
Choose File
No file chosen
Or
Forget Password?
Cancel
Submit
Or
Create an Account

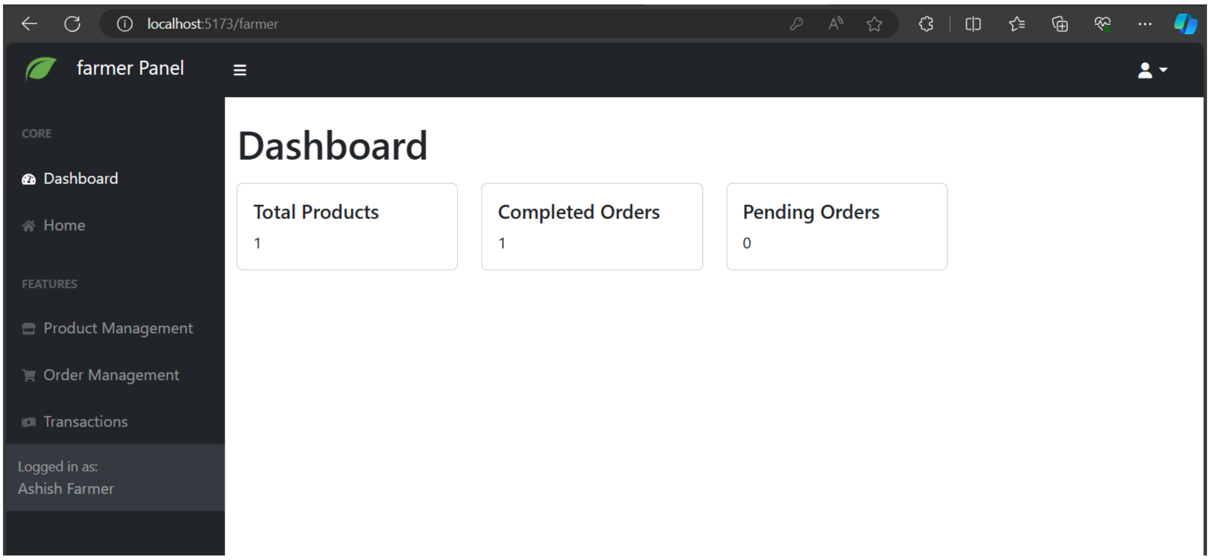

localhost:5173/login
Smart Kishan
Home
Crop Recommendation
Crop Cure
Marketplace
Login
SignUp
Login Page
Email
ashish@farmer.com
Password
Submit
Reset
Or No account?

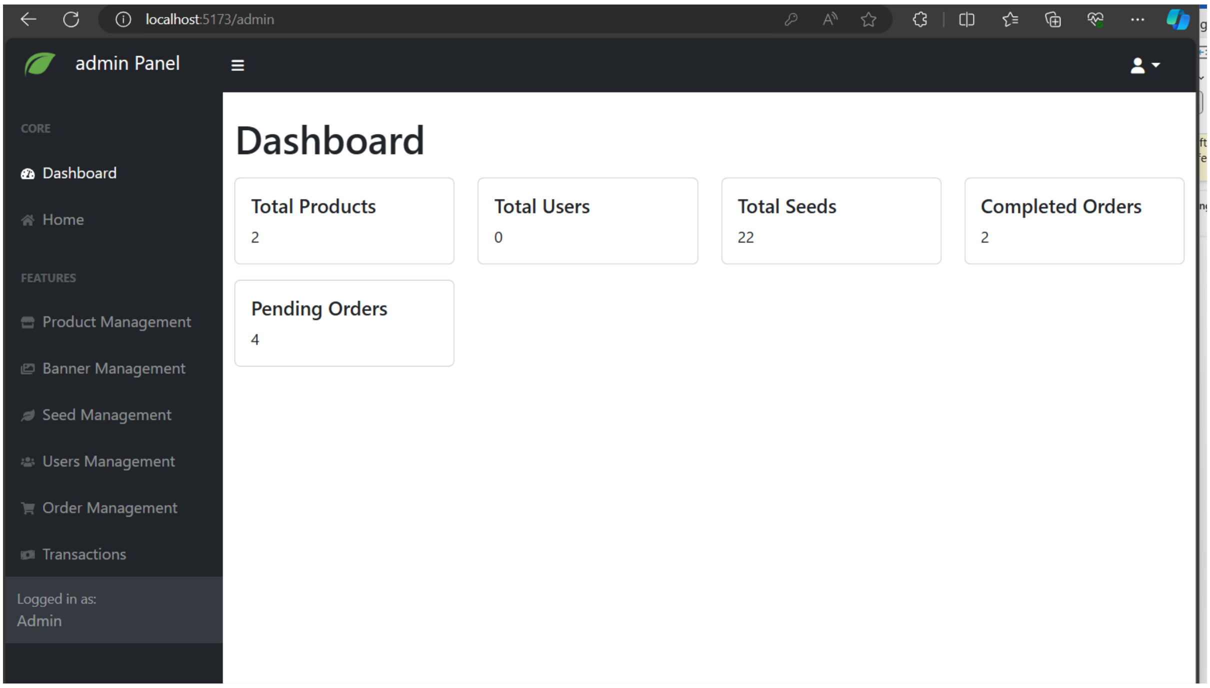

B.ScCSIT-7Sem
class and objec
Mailtrap - Emai
Smart Kishan
REPORT-FINAL
localhost:5173
Smart Kishan
Home
Crop Recommendation
Crop Cure
Marketplace
Login
SignUp
CROP RECOMMENDATION SYSTEM
WITH INTEGRATED CROP CURE AND MARKETPLACE
Our Services
Farmers Market
Want to buy fresh farm products directly from the farm? We have got live farmer selling
Crop Recommendation
Are you a farmer? Do you want to know the best crop for your soil?
Crop Cure
Do you think, your soil lacks some nutrients? Find out which one. And get the productivity


localhost:5173/crop-recommendation
Smart Kishan
Home
Crop Recommendation
Crop Cure
Marketplace
Login
SignUp
Crop Recommendation
Nitrogen
Enter nitrogen content of soil
Phosphorous
Enter phosphorous content of soil
Potassium
Enter potassium content of soil
pH
Enter pH value of soil
Rainfall
Enter rainfall
City
Enter your cityname
Get Recommendation


localhost:5173/crop-cure
Smart Kishan
Home
Crop Recommendation
Crop Cure
Marketplace
Login
SignUp
Cure Recommendation
Crop
Enter nitrogen content of soil
Nitrogen
Enter nitrogen content of soil
Phosphorous
Enter phosphorous content of soil
Potassium
Enter potassium content of soil
Get Cure


localhost:5173/farmer-market
Smart Kishan
Home
Crop Recommendation
Crop Cure
Marketplace
Login
SignUp
Our Farmer presents
search..
Search
MungBean
Fresh Healthy Mungbean dal
Add to Cart
Apple
Fresh
Add to Cart
Banana
Fresh
Add to Cart


localhost:5173/farmer
farmer Panel
CORE
Dashboard
Home
FEATURES
Product Management
Order Management
Transactions
Logged in as:
Ashish Farmer
Dashboard
Total Products
1
Completed Orders
1
Pending Orders
0


localhost:5173/admin
admin Panel
CORE
Dashboard
Home
FEATURES
Product Management
Banner Management
Seed Management
Users Management
Order Management
Transactions
Logged in as:
Admin
Dashboard
Total Products
2
Total Users
0
Total Seeds
22
Completed Orders
2
Pending Orders
4

```
# Convert data to numpy array and preprocess
def preprocess_data(data):
    data_array = np.array(data)
    X = data_array[:, :-1].astype(float)
    y = data_array[:, -1]
    # Convert labels to integers using crop_dict
    y = np.array([crop_dict[label] for label in y])
    return X, y

# For crop recommendation
```

```
def crop_recommendation(condn):
    data = read_csv("Data/Crop_recommendation.csv")
    features, labels = preprocess_data(data)

    # Split data into training and testing sets
    train_size = int(0.8 * len(features))
    X_train, X_test = features[:train_size], features[train_size:]
    y_train, y_test = labels[:train_size], labels[train_size:]

    # Create a RandomForestClassifier object
    rf_classifier = RandomForest(n_estimators=100)

    # Train the model
    rf_classifier.fit(X_train, y_train)

    # Make predictions
    predictions = rf_classifier.predict(condn)

    # Reverse label encoding to get original labels
    original_labels =
[list(crop_dict.keys())[list(crop_dict.values()).index(pred)] for pred in
predictions]

    return original_labels
```

```
class DecisionTree:
    def __init__(self, max_depth=None):
        self.max_depth = max_depth

    def fit(self, X, y):
        self.tree = self._grow_tree(X, y)

    def _grow_tree(self, X, y, depth=0):
        num_samples, num_features = X.shape
        num_classes = len(np.unique(y))

        # Stopping criteria
        if (self.max_depth is not None and depth >= self.max_depth) or
num_classes == 1:
            return {'class': np.bincount(y).argmax(), 'num_samples':
num_samples, 'depth': depth}

        # Find the best split
        best_feature = None
        best_threshold = None
        best_gini = np.inf
        for feature in range(num_features):
            thresholds = np.unique(X[:, feature])
            for threshold in thresholds:
                left_indices = np.where(X[:, feature] <= threshold)[0]
                right_indices = np.where(X[:, feature] > threshold)[0]
                if len(left_indices) == 0 or len(right_indices) == 0:
                    continue
                gini = self._gini_impurity(y[left_indices], y[right_indices])
                if gini < best_gini:
                    best_feature = feature
                    best_threshold = threshold
                    best_gini = gini

        if best_gini == np.inf:
            return {'class': np.bincount(y).argmax(), 'num_samples':
num_samples, 'depth': depth}

        # Split the dataset
        left_indices = np.where(X[:, best_feature] <= best_threshold)[0]
        right_indices = np.where(X[:, best_feature] > best_threshold)[0]
        left_tree = self._grow_tree(X[left_indices, :], y[left_indices],
depth + 1)
        right_tree = self._grow_tree(X[right_indices, :], y[right_indices],
depth + 1)
```

```
class RandomForest:
    def __init__(self, n_estimators=100, max_depth=None, sample_size=None):
        self.n_estimators = n_estimators
        self.max_depth = max_depth
        self.sample_size = sample_size
        self.estimators = []

    def fit(self, X, y):
        for _ in range(self.n_estimators):
            tree = DecisionTree(max_depth=self.max_depth)
            if self.sample_size:
                indices = np.random.choice(len(X), size=self.sample_size, 
replace=True)
                X_sample = X[indices]
                y_sample = y[indices]
            else:
                X_sample = X
                y_sample = y
            tree.fit(X_sample, y_sample)
            self.estimators.append(tree)

    def predict(self, X):
        predictions = np.array([estimator.predict(X) for estimator in 
self.estimators])
        return np.apply_along_axis(lambda x: np.bincount(x).argmax(), axis=0, 
arr=predictions)
```

```
@ app.route('/fertilizer-predict', methods=['POST'])
def fert_recommend():
    title = 'Cure recommendation'
    crop_name = str(request.form['cropname'])
    N = int(request.form['nitrogen'])
    P = int(request.form['phosphorous'])
    K = int(request.form['potassium'])

    df = pd.read_csv('Data/fertilizer.csv')

    nr = df[df['Crop'] == crop_name]['N'].iloc[0]
    pr = df[df['Crop'] == crop_name]['P'].iloc[0]
    kr = df[df['Crop'] == crop_name]['K'].iloc[0]

    n = nr - N
    p = pr - P
    k = kr - K
    temp = {abs(n): "N", abs(p): "P", abs(k): "K"}
    max_value = temp[max(temp.keys())]
    if max_value == "N":
        if n < 0:
            key = 'NHigh'
        else:
            key = "Nlow"
    elif max_value == "P":
        if p < 0:
            key = 'PHigh'
        else:
            key = "Plow"
    else:
        if k < 0:
            key = 'KHigh'
        else:
            key = "Klow"

    response = Markup(str(fertilizer_dic[key]))
    res = {
            "result": response,
            "message": "Recommended cure"
        }
    return json.dumps(res)
```