Crop Elevate
Crop Distribution from Panchayath to Farmers

Mini Project Report

Submitted by

Aleena Joseph

Reg. No.: AJC19MCA-I007

In Partial fulfillment for the Award of the Degree of

INTEGRATED MASTER OF COMPUTER APPLICATIONS

(INMCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2023-2024

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, “**CROP ELEVATE”** is the bonafide work of **ALEENA JOSEPH (Regno: AJC19MCA-I007)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

Mr. Ajith G.S	**Ms. Meera Rose Mathew**
Internal Guide	**Coordinator**

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **"CROP ELEVATE"** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2023-2024.

Date: 30-10-2023	**ALEENA JOSEPH**
KANJIRAPPALLY	**Reg: AJC19MCA-I007**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Mr. Ajith G.S** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

ALEENA JOSEPH

ABSTRACT

.

Crop Elevate

It is a management project that mainly aims to develop a platform that facilitates the exchange of crops to the farmers. Here the farmers can get the best crop that provides the best productivity within a limited period of time. The farmers of the different Wards who are interested in farming can register and the crops are given only to the farmers who met certain criteria that are given. The members can send notifications to the Secretary and the secretary can select them based on the priority. The crops are given to the farmers by the Panchayath Secretary only after the proper approval from the Ward Member.

• Farmer Registration: Farmers who are interested can create their user accounts on the platform.
• Income Analysis and Verification: The ward Member can access farmer income details and can analyse and verify the accuracy of the provided information.
• User Dashboard: Each user the Panchayath Secretary, Ward Member, and Farmer has a personalized dashboard displaying relevant information.
• Crop Recommendation System: The platform offers a personalized crop recommendation system, using the details provided by the farmers. The Panchayath Secretary can review it and take proper decisions based on it.
• Crop Database: A comprehensive database is maintained, containing information on various crops, expected profits
• Mobile Compatibility: The platform is designed to be accessible and user friendly across different devices including mobile phones. Mainly help the farmers to get the best crop for their field that give the best productivity in a limited period of time.

<u>Modules</u>

Farmer

- Registration
- Login
- Manage Profile
- Notifications
- Crop Selection

- Recommendation to member

Secretary (Admin)

- Login
- Manage Members
- Notifications
- Manage Meetings
- Reports
- Digital Notice board

Member

- Login
- Manage Profile
- Notifications
- Verify Framer

CONTENT

List of Abbreviation

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language.

CSS - Cascading Style Sheet

SQL - Structured Query Language

UML - Unified Modelling Language

JS - JavaScript

AJAX - Asynchronous JavaScript and XML Environment

RDBMS - Relational Database Management System

DBMS – Database Management System

NF – Normal Form

BCNF – Boyce-Codd Normal Form

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The project mainly aims to develop a easy to access platform for the users. Here the farmers can register for crops through online and the panchayath will provide the crops for the farmers who met the criterias. The Ward member plays an important role in the approval of the crops to the farmers. The panchayath secretary (Admin) will provide the final approval. The sales person will collect the products (products grown from the crops) and will sold it in the respective shops.

1.2 PROJECT SPECIFICATION

Developing a user-friendly online platform that enables farmers to register their crops. The system will evaluate the criteria set by the panchayat for crop approval. Ward members will play a pivotal role in this approval process, and the final approval will be granted by the panchayat secretary (Admin). Additionally, the platform will facilitate the collection of products grown from these crops by salespersons and their subsequent sale in designated shops. This system should include user registration and authentication, crop evaluation, approval workflows, and inventory management for the sales process. The platform should be accessible, intuitive, and efficient to serve the needs of both farmers and the administrative team.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System study is the process of analyzing, designing, and evaluating a system to understand its functioning, identify potential improvements, and ensure it aligns with the organization's goals and requirements.

System study is a critical phase in the system development life cycle, helping organizations make informed decisions about their systems and ensuring they align with business objectives. It can be applied to various types of systems, including software, business processes, and hardware infrastructure.

2.2 EXISTING SYSTEM

The existing system is very complex and is time consuming since the farmers need to apply for the crop through the hand written form and if any errors are identified by the member then the farmer need to collect the form from the member and can correct the errors and need to give the form back to the member and the member will give the form to the panchayath and the panchayath will give the crops. The list of the farmers to whom the crops are allotted is shown in a meeting called(Gramasabha)conducted for each ward. This is very time consuming.

2.2.1 NATURAL SYSTEM STUDIED

In the current system the farmers can apply for the crop by filling a hand written form and the farmers need to visit the panchayath for the further updates.it is difficult for the farmers. The farmers will get the current updates only after the meeting conducted by the panchayath secretary with the ward members.so, it is difficult for the farmers.

2.2.2 DESIGNED SYSTEM STUDIED

In the current system the farmers can apply for the crop by filling a hand written form and the farmers need to visit the panchayath for the further updates.it is difficult for the farmers.so the system is mainly designed to help the farmers to apply for the crop in an online manner so that they can easily get the current updates and can also get the current update that the application is approved or not or need any modification in the application form etc.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Time consuming
- Updates Of details is Difficult

- Getting updates is difficult
- Clearing clarifications is difficult

2.4 PROPOSED SYSTEM

The proposed system mainly hep the farmers to apply for the crop samplings easily in an online mode. Through this system the farmers can easily register for the crop samplings. The copy of the application form is sent to the ward member so that the ward member can verify that the farmer is in the correct ward or not. When the member approve the application the message is sent to the farmer that the ward member has approved the application. The copy of the approved application from the ward member is sent to the admin, here it is the Panchayath Secretary. The panchayath secretary will approve the application and the crop is given to the farmer by the panchayath secretary. the up-to-date updates are displayed in the farmer page.so the system will help the farmers to get the updates easily.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Saves time
- Get current updates
- Easily clarify the doubts
- Easily correct errors

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is a comprehensive analysis and evaluation conducted to assess the practicality, viability, and potential success of a proposed project, initiative, or investment. Its main objective is to determine whether the project is technically, economically, socially, and operationally feasible before proceeding with implementation. The study involves examining various aspects, including technical requirements, resource availability, financial considerations, social and environmental impacts, operational compatibility, and potential risks. By conducting a feasibility study, decision-makers can gain valuable insights into the project's likelihood of success, identify potential challenges, and explore alternative solutions.

3.1.1 Economical Feasibility

Assesses whether the necessary software may bring an organization financial benefits. It includes the expenses related to the software development team, the expected cost of the necessary hardware and software, the expense of conducting a feasibility study, and so on. System that supplies crops to farmers must be carefully evaluated in terms of its financial sustainability and prospective profitability. In this analysis, costs for infrastructure, technology, personnel, crop procurement, transportation, and overhead expenses are all evaluated. The analysis covers various funding sources and looks at the initial investment needed for development and deployment. Pricing policies are developed to find a balance between farmer affordability and system sustainability. As a result of the economic feasibility assessment, stakeholders may make well-informed decisions and assure the system's financial stability, which benefits farmers directly and promotes a long-term crop exchange model within the community.

3.1.2 Technical Feasibility

Assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. The technical feasibility study for a system that supplies crops to farmers looks at how well the suggested solution can be created, put into practice, and run. The availability and suitability of necessary technology, as well as data collecting and integration skills, are important factors. The system should support existing agricultural technology and databases and have a user-friendly interface that is accessible to users with diverse levels of technical expertise. Making sure the system is technically prepared to promote the exchange of crops to farmers by enabling efficient and reliable operations with the right technology and infrastructure is achieved by conducting an extensive technical feasibility assessment.

3.1.3 Behavioral Feasibility

Behavioral feasibility, also known as behavioral analysis, is an essential aspect of project or business feasibility studies. It assesses whether a proposed project, system, or business change can be accepted and adopted by the people who will use it. behavioral feasibility is about understanding and addressing the human and cultural aspects of a project to ensure its successful implementation. It is an essential component of feasibility studies to ensure that a project not only makes sense from a technical or financial perspective but is also supported by those it affects.

3.1.4 Feasibility Study Questionnaire

1. **How the crops are given to the Farmers? Is there any criterias?**
   The crops are mainly given to the farmers mainly based on the annual income. If there is a farmer with high annual income and a one with the low annual income the crop will be provided to the one with the low income.
2. **How the applications forms are submitted?**
   The application forms are filled and submitted as a hard copy.it is a form that need to be hand written.
3. **If any data is not entered or any field is not added how the field can be updated?**
   The panchayath together with the Ward Members will evaluate the form and if there is any corrections the Ward Member will inform the farmer to update it and give back the form.
4. **How the farmers will get the updates?**
   The farmers will get the updates in the Gramasabha or they need to visit the panchayath for the new updations.
5. **How will you inform them that their registration is approved or not?**
   We will inform them by calling them and inform them that the application is approved.
6. **How the payment is done by the farmers?**
   The farmers can done the payment by cash.
7. **Did you give any advices for growing the crops?**
   No not at all. We didn’t provide any advices.
8. **How will you inform the farmers that their crops had arrived?**
   We will call or message the corresponding farmers and inform them.

9. **From where the farmers will collect the crops?**

   The farmers can collect the crops from the Krishi Bhavan.

10. **Did you provide any shops where the farmers can sell the products that are grown from their crops?**

    No we did not have such facilities, farmers can to sell their products by their own.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- intel core i3
RAM	- 4 G B
Hard disk	- 1 T B

3.2.2 Software Specification

Front End – HTML,CSS

Back End - Django

Database - MYSQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 Django

Django is a high-level Python web framework that simplifies and streamlines web development. It provides a structured and efficient way to build web applications and is known for its "batteries-included" philosophy, meaning it includes many built-in features and tools to help developers. Django is a versatile framework that simplifies web development by providing a robust set of tools and conventions. It is widely used for building web applications, from simple blogs to complex, data-driven websites.

3.3.2 SQLite

SQLite is a lightweight, self-contained, serverless, and open-source relational database management system (RDBMS) that is renowned for its simplicity, efficiency, and portability. It is designed to be embedded within applications, allowing them to store, manage, and retrieve structured data using SQL queries without the need for a separate database server. SQLite is widely used in a variety of software applications, including mobile apps, desktop programs, and embedded systems, due to its small footprint, fast performance, and support for ACID (Atomicity, Consistency, Isolation, Durability) compliance.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

Any designed system or product's development starts with the design phase. An efficient system depends on well-executed design, which is a creative process. A crucial step in the creation of sophisticated software, hardware, and information systems is design of systems. It entails the planning and coordination of numerous elements to produce a unified and effective system that satisfies particular needs. System design offers several advantages in various domains.

4.2 UML DIAGRAM

Unified Modeling Language (UML) diagrams are a set of standardized graphical notations used in software engineering, systems engineering, and various other fields to visually represent and document various aspects of a system. These diagrams help communicate and analyze the design, structure, behavior, and interactions within a system. Here are some common UML diagram types and their definitions:

- Class Diagram
- Object Diagram
- Sequence Diagram
- Use-case Diagram
- Activity Diagram
- State-Chart Diagram
- Collaboration Diagram
- Deployment Diagram
- Component Diagram

4.2.1 USE CASE DIAGRAM

Use-case diagrams aid in capturing system requirements and model a system's behavior in UML. The scope and high-level functions of a system are described in use-case diagrams. The interactions between the system and its actors are also depicted in these diagrams. Use-case diagrams show what the system does and how the actors use it, but they do not show how the system works internally.

The context and requirements of either the entire system or the key components of the system are illustrated and defined by use-case diagrams. A complex system can be represented by a single use-case diagram, or its various components can be represented by a number of use-case diagrams. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process.

The primary components of a use case diagram include:

- ***Actors:***

  Actors are outside parties who communicate with the system. Users, roles, or other systems may be included. On the diagram, actors are shown as stick figures.
- ***Use cases:***

  Use cases are particular services or functionalities that the system offers. On the diagram, each use case is represented by an oval and corresponds to a distinct aspect of system functionality.
- ***Associations:***

  The connections between actors and use cases show how they interact or relate to one another. An association shows that a particular actor is involved in a given use case.
- ***System boundary:***

  The system boundary, which is typically a box, encloses all the use cases and actors and establishes the scope of the system being modeled.

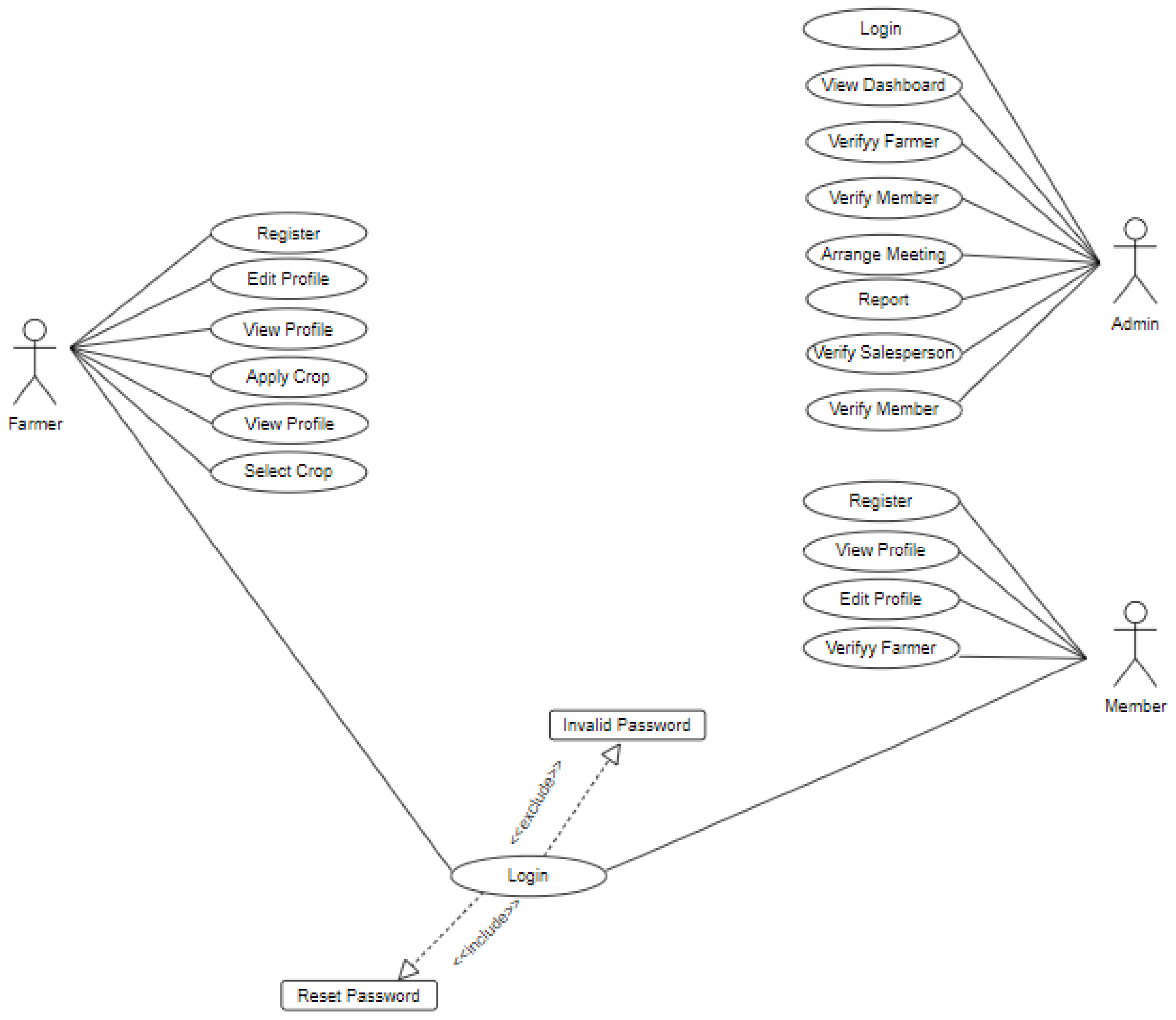


Fig:1 Use case Diagram for Crop Elevate

4.2.2 SEQUENCE DIAGRAM

A sequence diagram is a graphical representation used in software engineering and system design to visualize and describe the interactions and order of execution between various components or objects within a system. It provides a dynamic view of how different parts of a system communicate with one another and the sequence in which messages or method calls are exchanged.

In a sequence diagram:

Lifelines: These represent the entities (objects, components, or actors) involved in the interaction.

Messages: Arrows and notations between lifelines indicate the messages or method calls being exchanged between the entities.

Activation Bars: These show the duration of time during which an entity is active in processing a message.

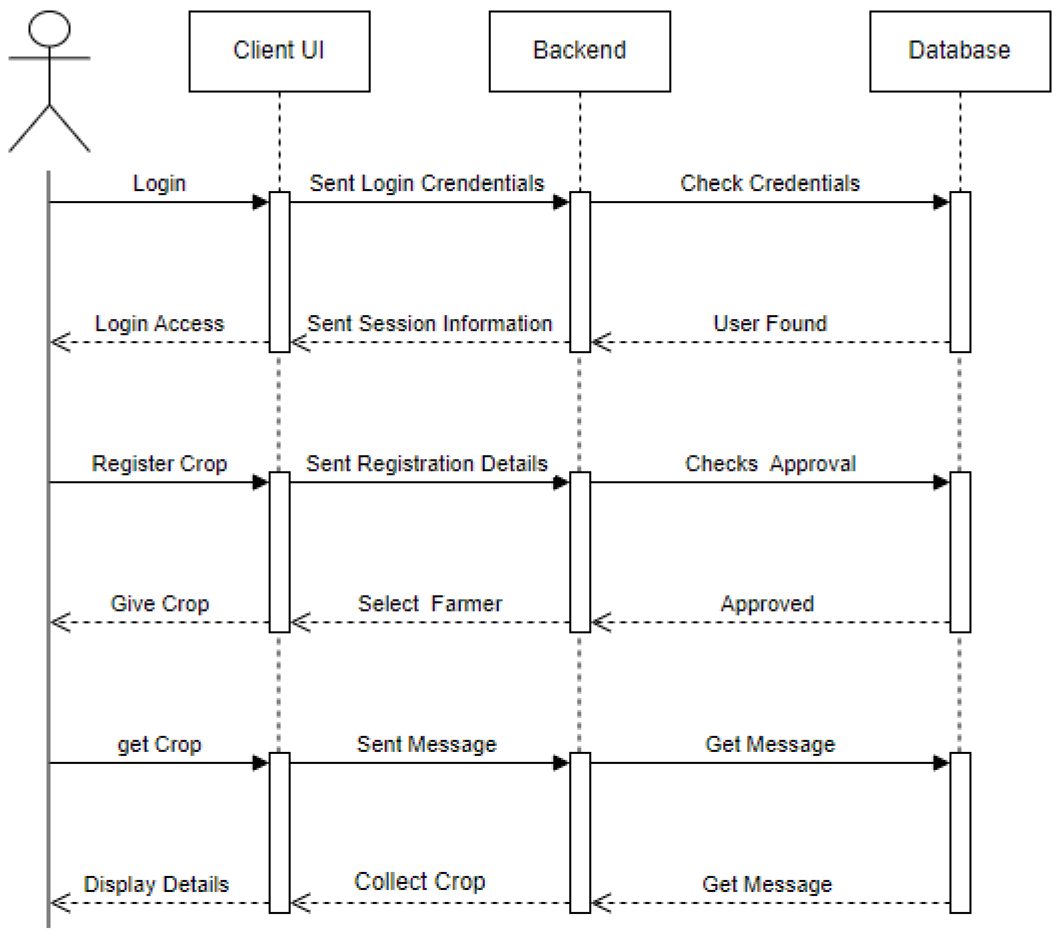


Fig:2 Sequence Diagram for Crop Elevate

4.2.3 State Chart Diagram

A state chart diagram, also known as a state machine diagram or state diagram, is a visual representation used in software engineering and system modeling to describe the various states that an object, component, or system can exist in and the transitions between those states. It is particularly useful for modeling the behavior of systems with complex state-dependent logic. Here are the key components of a state chart diagram:

State: A state represents a condition or situation in which an object or system can exist. Each state is depicted as a rectangle with a label describing the state's name or characteristics.

Initial State: An initial state is the starting point of a state machine. It is often marked with a small filled circle connected to the initial state by an arrow.

Final State: A final state represents the end of a particular state or the completion of a process. It is typically marked with a small filled circle enclosed by a larger circle.

Transition: A transition represents a change of state and is depicted as an arrow connecting two states. It is labeled with an event that triggers the transition and a condition (if any) that must be satisfied for the transition to occur.

Event: An event is an occurrence or trigger that initiates a state transition. Events are associated with transitions and can be external stimuli, user actions, or time-based events.

Action: Actions are executable operations or behaviors associated with a state or transition.

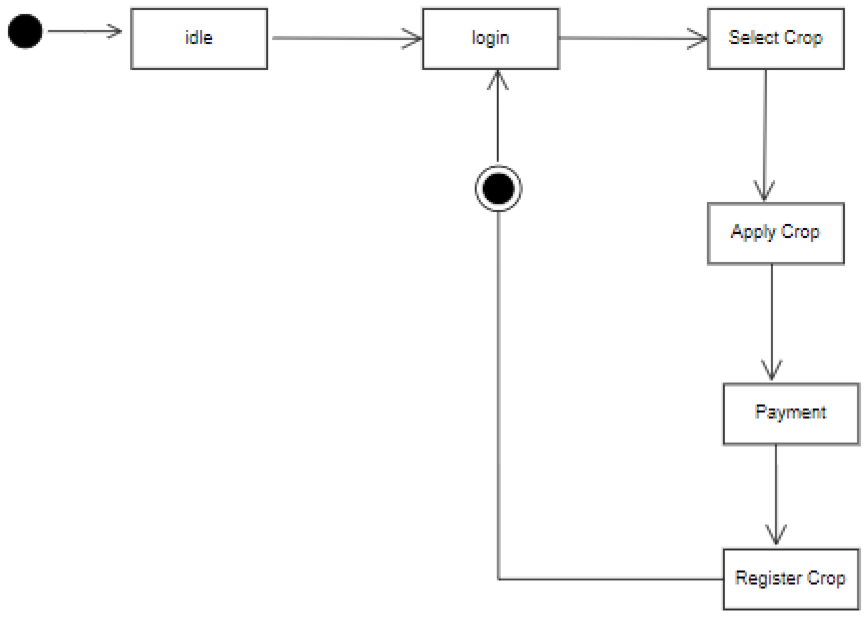


Fig:3 State Chart Diagram for Crop Elevate

4.2.4 Activity Diagram

An activity diagram is a type of UML (Unified Modeling Language) diagram that is used to represent the workflow or flow of activities within a system, process, or business. It is a visual tool that helps model, understand, and document the steps or actions in a process or system.

Key components of an activity diagram include:

Activity: Activities are represented as rounded rectangles and represent specific tasks, actions, or operations in the process. They can be detailed or high-level, depending on the level of abstraction needed.

Action: Actions are represented by rectangles with rounded corners and represent individual steps within an activity. They provide a more granular view of the work being done.

Control Flow: Control flow arrows show the sequence in which activities and actions are executed. They connect activities and actions, indicating the order of execution.

Decision Nodes: Decision nodes are represented as diamonds and are used to represent decision points in the process. Depending on conditions, the process can take different paths.

Merge Nodes: Merge nodes, also shown as diamonds, represent points where different paths converge.

Forks and Joins: Forks represent points where the process splits into multiple parallel activities, while joins indicate where those parallel paths converge back into a single flow.

Activity diagrams are valuable for business process modeling, software design, and system analysis.

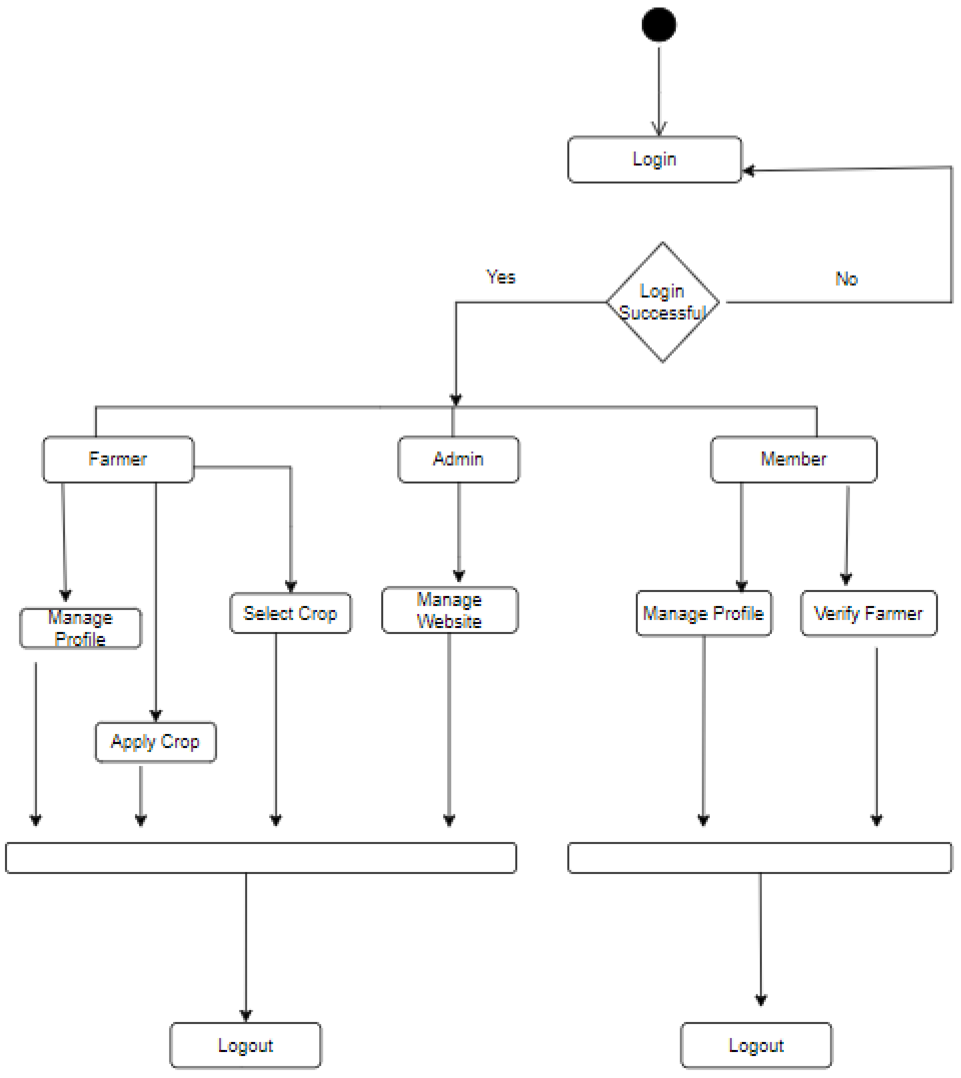


Fig:4 Activity Diagram for Crop Elevate

4.2.5 Class Diagram

A class diagram is a type of UML (Unified Modeling Language) diagram used in software engineering to visualize and document the structure and relationships of classes and objects within a system. It provides a high-level overview of the system's structure and serves as a blueprint for designing and implementing software systems. Class diagrams are an essential part of object-oriented modeling and are commonly used during the software development process.

Components of a Class Diagram:

Class: Represents a blueprint or template for creating objects.

Contains attributes (data members) and methods (functions) that define the behavior of objects of that class.

Attributes: Variables that represent the data or characteristics of a class. Typically displayed as name: type pairs (e.g., name: string).

Methods: Functions or operations that define the behavior of a class. Represented with their names, input parameters, and return types.

Associations: Lines connecting classes to indicate relationships between them.

Multiplicity notations (e.g., 0..1, 1, 1..*) describe how many objects participate in the relationship.

Inheritance (Generalization): Represents an "is-a" relationship where one class (subclass or child) inherits attributes and methods from another class (superclass or parent).

Interfaces: Specifies a contract that a class must adhere to. A class can implement one or more interfaces, ensuring that it provides specific methods.

Aggregation and Composition: Aggregation represents a "has-a" relationship where one class contains or is part of another class.

Composition is a stronger form of aggregation, indicating that the parts belong exclusively to the whole.

Dependency: Shows that one class relies on another class but is not tightly coupled to it.

Class diagrams are used for design and documentation purposes, helping software developers, designers, and other stakeholders understand the structure and relationships within a software system.

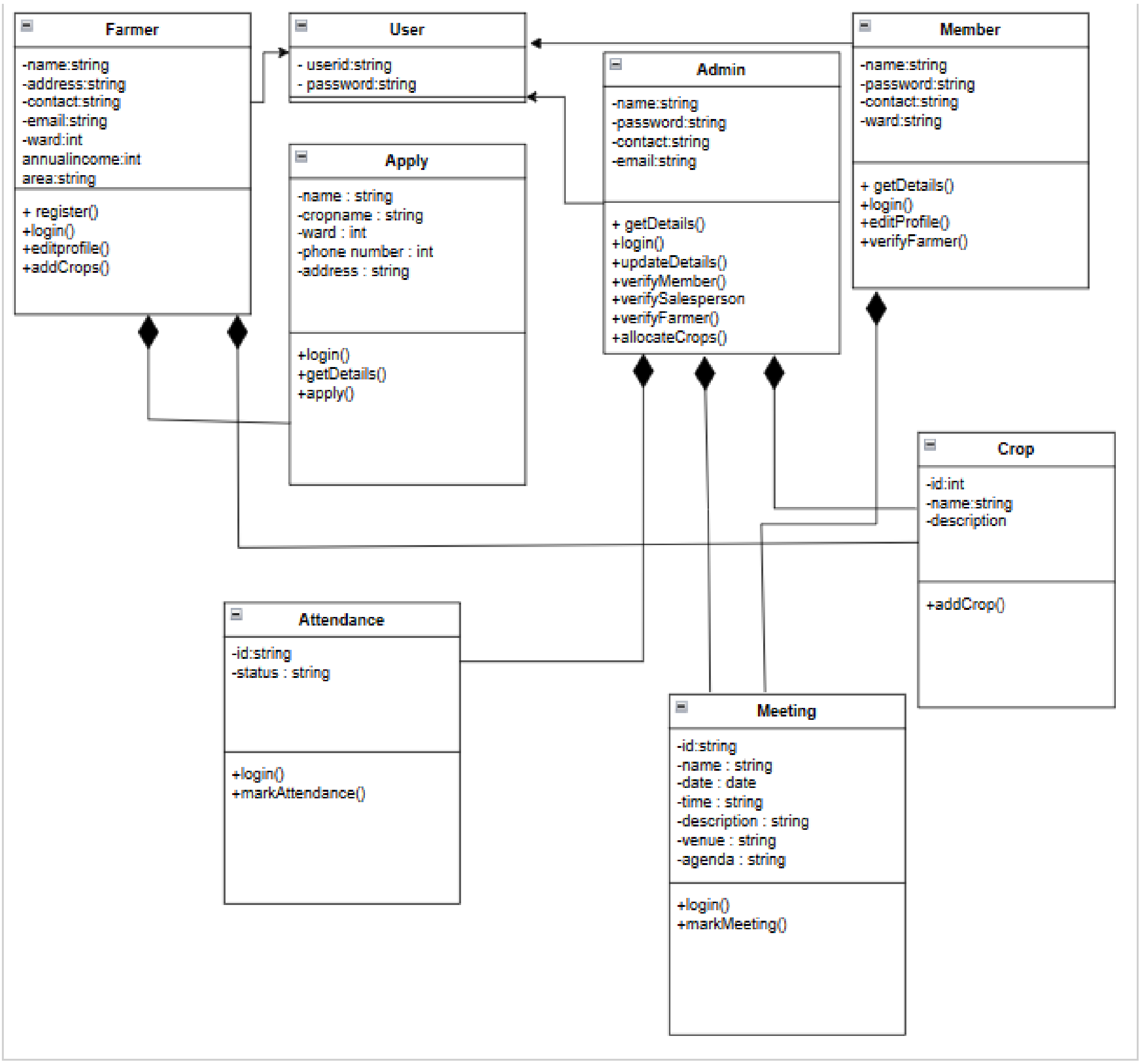


Fig:5 Class Diagram for Crop Elevate

4.2.6 Object Diagram

An object diagram is a structural diagram in the Unified Modeling Language (UML) that represents instances of classes or objects at a specific point in time within a system or a specific scenario. It provides a detailed view of the relationships and interactions between objects, offering a snapshot of how objects collaborate at a particular moment.

Key Features of Object Diagrams:

Objects: In an object diagram, individual objects or instances of classes are depicted. These objects can represent real-world entities or software components.

Links: Object diagrams illustrate the relationships or associations between objects. Links between objects show how they are connected or interact with each other.

Attributes: Object diagrams may display object attributes and their values, providing a detailed look at the state of objects at the given moment.

Multiplicity: Object diagrams can show the cardinality of associations, indicating how many objects participate in a particular relationship.

Scenario-Specific: Object diagrams are typically scenario-specific, meaning they depict objects and their relationships within a specific context or use case.

Object diagrams are useful for visualizing and verifying how objects collaborate and communicate within a system during a particular scenario or event.

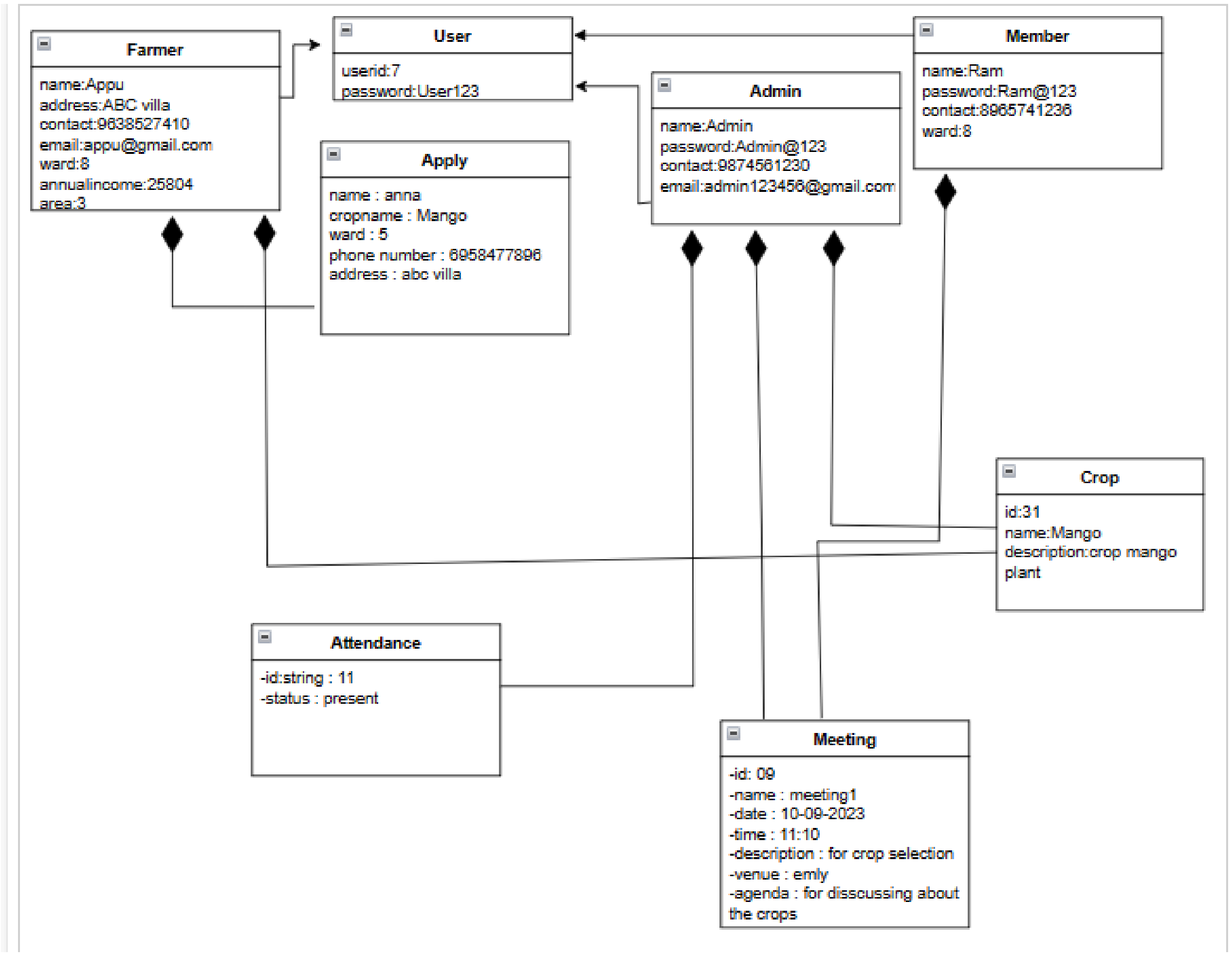


Fig:6 Object Diagram for Crop Elevate

4.2.7 Component Diagram

A component diagram is a type of structural diagram in the Unified Modeling Language (UML) that represents the high-level architecture and structure of a system or application. It focuses on the organization of components, which can be software modules, libraries, or other encapsulated parts of a system. Component diagrams are used to model the physical and logical aspects of a system's components and how they interact.

Key Features of Component Diagrams:

Components: In a component diagram, components are represented as boxes or rectangles, each depicting a discrete part of the system. These components can be at different levels of abstraction, such as individual classes, packages, or larger system modules.

Relationships: Component diagrams show the relationships and dependencies between components. These relationships can include associations, dependencies, aggregations, and more, indicating how components interact with each other.

Interfaces: Components can have interfaces that define the services or functionality they provide or require from other components. Interfaces are depicted using lollipops or sockets.

Ports: Ports are used to represent the interaction points of components. They show where connections, such as communication channels or data flows, are established between components.

Connectors: Connectors are lines or arrows that illustrate the connections between components. They depict the flow of data or communication pathways between components.

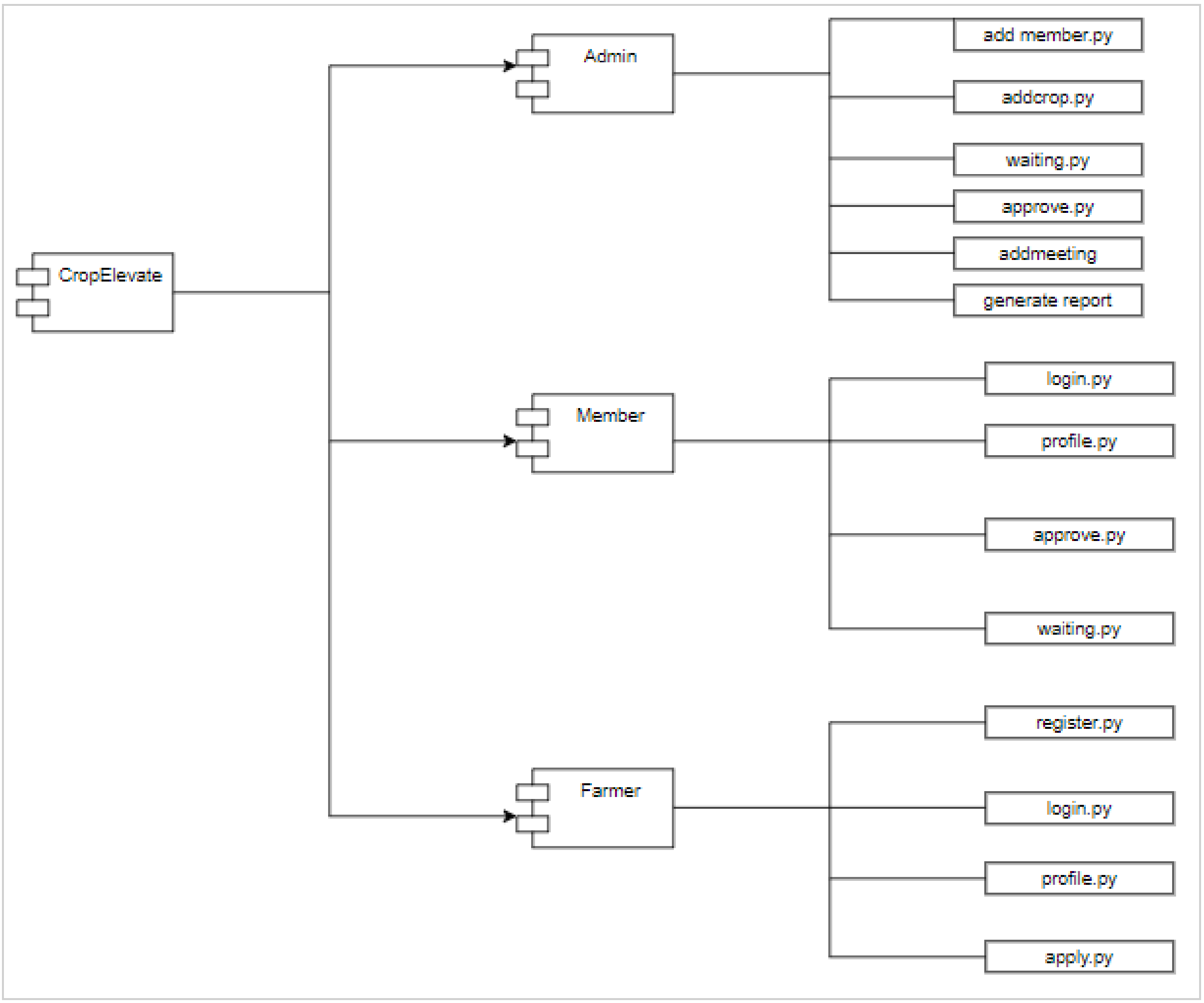


Fig:7 Component Diagram for Crop Elevate

4.2.8 Deployment Diagram

A deployment diagram is a type of diagram in the Unified Modeling Language (UML) that focuses on the physical deployment of software components within a system or application. It illustrates how software artifacts, such as executable files, libraries, and components, are distributed across hardware nodes and interconnected to form a complete system.

Key Features of Deployment Diagrams:

Nodes: In a deployment diagram, nodes represent hardware entities such as servers, workstations, or devices. These nodes serve as the execution environments for the software components.

Artifacts: Software artifacts, including components, files, and databases, are represented in the diagram to illustrate their deployment on specific nodes.

Relationships: Deployment diagrams show the relationships between nodes and artifacts. These relationships can include associations, dependencies, and usage relationships, indicating how software components interact with hardware nodes.

Deployment Specifications: Deployment specifications can be associated with artifacts to provide additional information about deployment options, configurations, and requirements.

Deployment diagrams are valuable for system architects and developers to visualize the physical infrastructure and software distribution of a system. They help in planning and understanding how software components will be deployed on hardware, including considerations for scalability, performance etc.

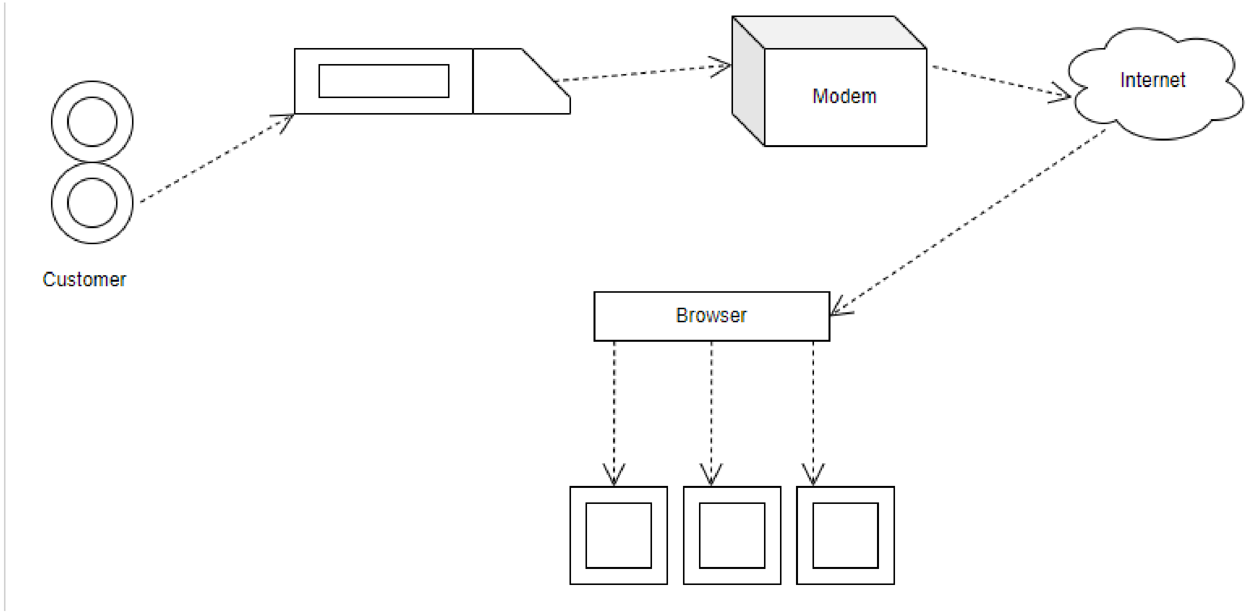


Fig:8 Deployment Diagram for Crop Elevate

4.3 USER INTERFACE DESIGN USING FIGMA

Form Name: signup

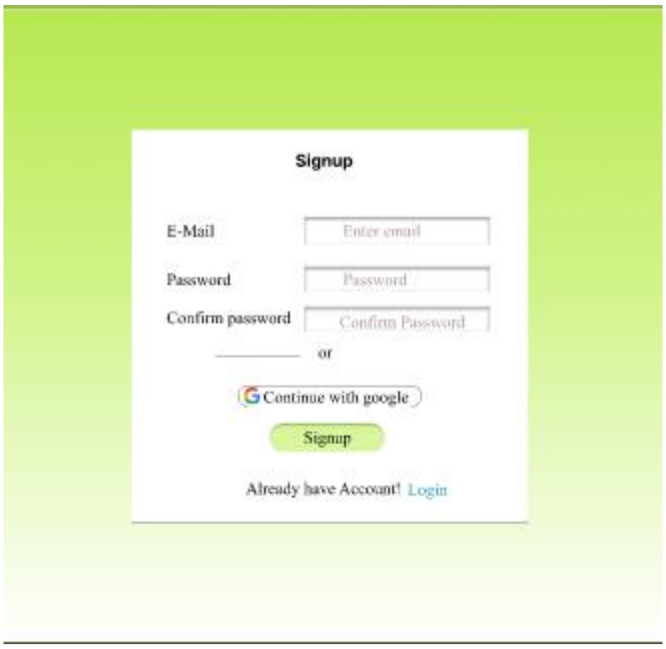


Fig:9 signup

Form Name: Login

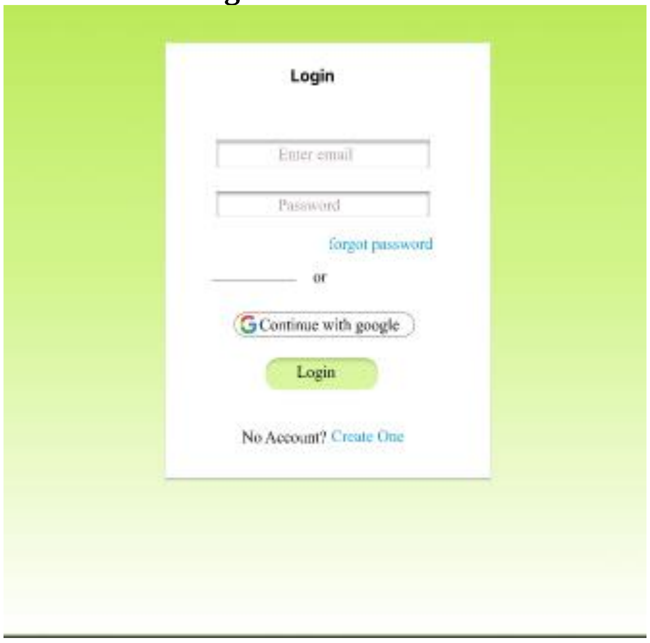
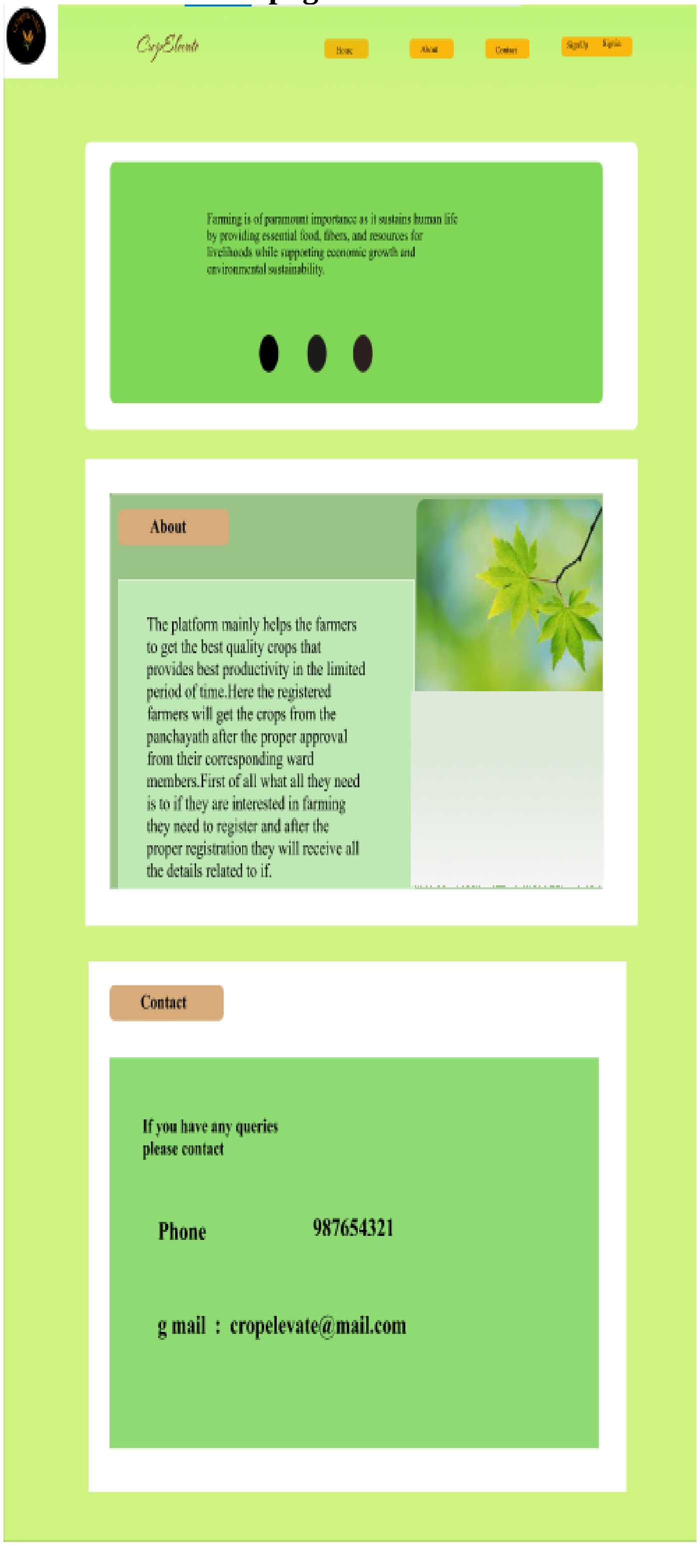


Fig:10 Login

Form Name: Homepage



Farming is of paramount importance as it sustains human life by providing essential food, fibers, and resources for livelihoods while supporting economic growth and environmental sustainability.

The platform mainly helps the farmers to get the best quality crops that provides best productivity in the limited period of time.Here the registered farmers will get the crops from the panchayath after the proper approval from their corresponding ward members.First of all what all they need is to if they are interested in farming they need to register and after the proper registration they will receive all the details related to it.

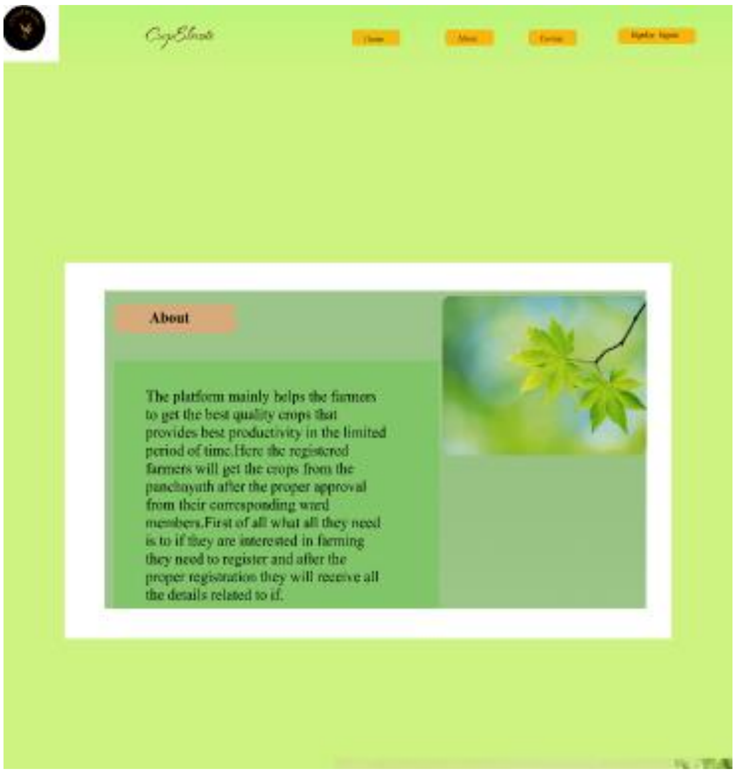
If you have any queries please contact

Phone 987654321

g mail : cropelevate@mail.com

Fig:11 Homepage

Form Name: About

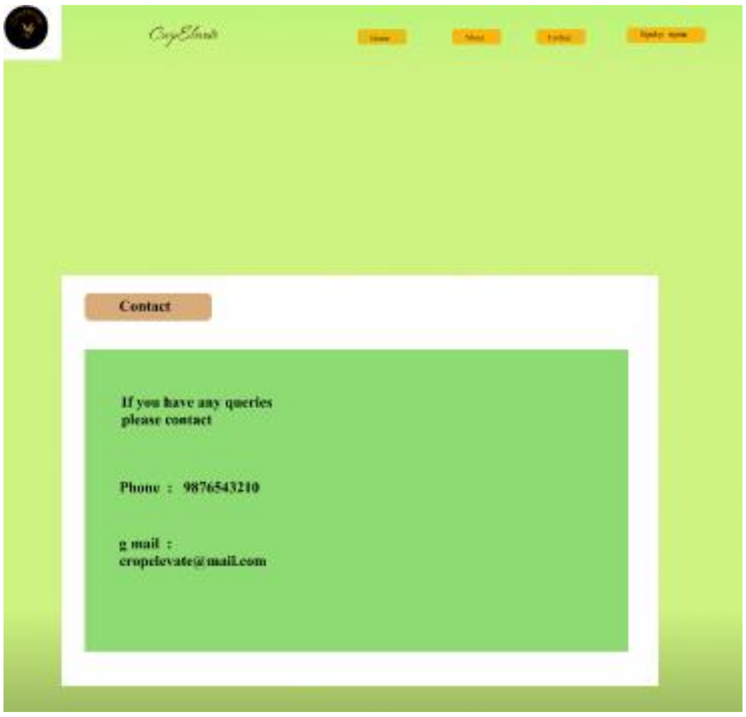


About

The platform mainly helps the farmers to get the best quality crops that provides best productivity in the limited period of time.Here the registered farmers will get the crops from the panchayath after the proper approval from their corresponding ward members.First of all what all they need is to if they are interested in farming they need to register and after the proper registration they will receive all the details related to if.

Fig:12 About

Form Name: Contact



Contact

If you have any queries please contact

Phone : 9876543210

g mail : cropelevate@mail.com

Fig:13 Contact

4.4 DATABASE DESIGN

Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept of designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored.

The main objectives behind database designing are to produce physical and logical design models of the proposed database system. To elaborate this, the logical model is primarily concentrated on the requirements of data and the considerations must be made in terms of monolithic considerations and hence the stored physical data must be stored independent of the physical conditions. On the other hand, the physical database design model includes a translation of the logical design model of the database by keep control of physical media using hardware resources and software systems such as Database Management System (DBMS).

4.4.1 Relational Database Management System (RDBMS)

A Relational Database Management System (RDBMS) is a type of database management system that organizes and stores data in a structured format, primarily using tables with rows and columns. It is based on the principles of the relational model, which was introduced by Edgar F. Codd in the 1970s. RDBMSs are designed to manage and manipulate data efficiently while maintaining the integrity and relationships between different data elements.

Key characteristics of RDBMS:

Tables: Data is organized into tables, also known as relations, where each table consists of rows (tuples) and columns (attributes). Each row represents a unique record, while each column represents a specific piece of data.

Schema: RDBMSs define a schema that specifies the structure of tables, including data types, constraints, and relationships between tables. This ensures data consistency and integrity.

SQL (Structured Query Language): RDBMSs use SQL as the standard language for querying and manipulating data. SQL allows users to perform operations like selecting, inserting, updating, and deleting data.

ACID Properties: RDBMSs ensure data integrity through ACID (Atomicity, Consistency, Isolation,

Durability) properties. These properties guarantee that database transactions are processed reliably.
Data Relationships: RDBMSs support the establishment of relationships between tables, enabling the creation of complex queries that retrieve related data from multiple tables.
Normalization: RDBMSs promote data normalization, which reduces data redundancy and enhances data integrity by minimizing the risk of update anomalies.

Popular RDBMSs include MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, and SQLite. These systems are widely used in various applications, ranging from enterprise-level data management to web development and mobile applications, due to their reliability and data consistency features.

4.4.2 Normalization

Normalization in the context of databases refers to the process of organizing and structuring data within a relational database to eliminate data redundancy and ensure data integrity. It involves breaking down large tables into smaller, related tables and establishing relationships between them. The main objective of normalization is to reduce the potential for data anomalies, such as update anomalies, insert anomalies, and delete anomalies, which can occur when data is not properly structured. The process typically involves defining a set of rules or normal forms, including First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), and beyond, to guide the design and organization of database tables. Each normal form has specific criteria that a table must meet to be considered normalized. Normalization is a critical step in database design, ensuring efficient data storage, retrieval, and maintenance while minimizing data inconsistencies.
First Normal Form (1NF): In 1NF, a table is considered normalized if it has no repeating groups (columns with multiple values), and all attributes contain atomic (indivisible) values. It ensures that each cell in a table contains a single, indivisible value.
Second Normal Form (2NF): 2NF builds on 1NF and further eliminates partial dependencies. To be in 2NF, a table must meet 1NF criteria, and all non-key attributes must depend on the entire primary key, not just part of it. This avoids redundancy related to composite primary keys.
Third Normal Form (3NF): 3NF extends 2NF by removing transitive dependencies. A table is in 3NF if it meets 2NF criteria and has no attributes that depend on other non-key attributes. It eliminates indirect relationships between attributes.
Boyce-Codd Normal Form (BCNF): BCNF is a stricter form of 3NF that ensures that for every non-trivial functional dependency, the left-hand side is a superkey. It helps avoid anomalies related

to key attributes.

Fourth Normal Form (4NF): 4NF deals with multi-valued dependencies, ensuring that if an attribute is multi-valued, it's stored in a separate table with a relationship to the main table.

Fifth Normal Form (5NF): 5NF addresses cases where an attribute depends on a combination of super keys and is often used in specialized situations to further reduce data redundancy.

4.4.3 Sanitization

sanitization refers to the process of cleansing and validating input data to ensure that it is safe and does not pose security risks to the database or the application using it. Sanitization is a critical practice to prevent various security vulnerabilities, including SQL injection, Cross-Site Scripting (XSS), and other forms of cyberattacks.

sanitization is a fundamental practice in database design to protect against vulnerabilities and ensure that data is stored and retrieved safely. It is a crucial element in maintaining the integrity and security of the database and the applications that interact with it.

4.4.4 Indexing

An index is a structural component within a database that serves to optimize the speed of various table operations. These indexes can be established on one or more columns, enhancing the efficiency of data retrieval and record sorting. When designing an index, it is imperative to carefully consider which columns will be frequently used in SQL queries, warranting the creation of one or more indexes on these columns.

In practical terms, an index resembles a specialized table, storing a primary key or an index field and a reference to each corresponding record in the primary table. These indexes remain concealed from users and exclusively serve the database's search engine, swiftly pinpointing records. The creation of indexes is facilitated through the utilization of the "CREATE INDEX" statement applied to tables.

While the presence of indexes in tables may result in slightly longer execution times for "INSERT" and "UPDATE" statements due to the need to modify index values, it significantly accelerates the execution of "SELECT" statements by expediting the process of locating records.

4.5 TABLE DESIGN

1. Tbl_Login

Primary key: Login_id

No	Field	Data_Type	Key constraint
1	Login_id	Int	Primary key, Auto Increment
2	Status	Int	
3	Email	VARCHAR	NOT NULL
4	Password	VARCHAR	NOT NULL
5	Approve	VARCHAR	NOT NULL

2.Tbl_Profile

Primary key: Profile_id
Foreign key: Login_id references table Tbl_Login

No	Field	Data_Type	Key constraint
1	Profile_id	Int	Primary Key, Auto Increment
2	Login_id	Int	Foreign Key
3	Name	VARCHAR	NOT NULL
4	Address	VARCHAR	NOT NULL
5	Place	VARCHAR	NOT NULL
6	Ward	VARCHAR	NOT NULL
7	Contact	VARCHAR	NOT NULL
8	Dob	VARCHAR	NOT NULL
9	Land	VARCHAR	NOT NULL
10	Annual income	VARCHAR	NOT NULL
11	Licence no	VARCHAR	NOT NULL

3. Tbl_Crop

Primary key: Crop_id
Foreign key: Login_id references table Tbl_Login

No	Field	Data_Type	Key Constraints
1	Crop_id	INT	Primary Key, Auto Increment
2	Login_id	INT	Foreign Key
3	Crop Name	VARCHAR	NOT NULL
4	Yield (After how many years)	VARCHAR	NOT NULL

4. Tbl_Member

Primary key: Member_id
Foreign key: Login_id references table Tbl_Login

No	Field	Data_Type	Key Constraints
1	Member_id	INT	Primary Key, Auto Increment
2	Login_id	INT	Foreign Key
3	Name	VARCHAR	NOT NULL
4	Email	VARCHAR	NOT NULL
5	Password	VARCHAR	NOT NULL
6	dob	VARCHAR	NOT NULL
7	Address	VARCHAR	NOT NULL
8	District	VARCHAR	NOT NULL
9	Thaluk	VARCHAR	NOT NULL
10	Panchayath	VARCHAR	NOT NULL
11	Ward Name	VARCHAR	NOT NULL
12	Ward No	INT	NOT NULL
13	Postal	INT	NOT NULL
14	Phone	INT	NOT NULL
15	Bio	VARCHAR	NOT NULL

5. Tbl_Apply_Crop

Primary key: Apply_Crop_id
Foreign key: Login_id references table Tbl_Login,
Crop_id references table Tbl_Crop

No	Field	Data_Type	Key Constraints
1	Apply_Crop_id	INT	Primary Key, Auto Increment
2	Login_id	INT	Foreign Key
3	Crop_id	INT	Foreign Key
4	Cname	VARCHAR	NOT NULL
5	Farmer Name	VARCHAR	NOT NULL
6	Address	VARCHAR	NOT NULL
7	contactNo	VARCHAR	NOT NULL
8	Ward No	VARCHAR	NOT NULL
9	Annual Income	INT	
10	Land	VARCHAR	NOT NULL
11	File_upload	VARCHAR	NOT NULL
12	Crop giventime	VARCHAR	NOT NULL
13	Is_approved	VARCHAR	NOT NULL
14	Is_approvedd	VARCHAR	NOT NULL
15	Is_given	VARCHAR	NOT NULL

6. Tbl_Meeting

Primary key: Meeting_id
Foreign key: Login_id references table Tbl_Login

No	**Field**	**Data_Type**	**Key Constraints**
1	Meeting_id	INT	Primary Key, Auto Increment
2	Login_id	INT	Foreign Key
3	Meeting_date	VARCHAR	NOT NULL
4	Meeting_time	VARCHAR	NOT NULL
5	desmeeting	VARCHAR	NOT NULL
6	Last_edited_at	VARCHAR	NOT NULL
7	Meeting_venue	VARCHAR	NOT NULL
8	Meeting_agenda	VARCHAR	NOT NULL
9	Report	VARCHAR	NOT NULL
10	Is_active	VARCHAR	NOT NULL

7. Tbl_Ward_Attendance

Primary key: attendance_id
Foreign key: Login_id references table Tbl_Login,
Meeting_id references table Tbl_Meeting,
Member_id references table Tbl_Member

No	**Field**	**Data_Type**	**Key Constraints**
1	attendance_id	INT	Primary Key, Auto Increment
2	Login_id	INT	Foreign Key
3	Member_id	INT	Foreign Key
4	Meeting_id	INT	Foreign Key
5	Is_present	VARCHAR	NOT NULL

8. Tbl_Farmer

Primary key: Farmer_id
Foreign key: Login_id references table Tbl_Login

No	**Field**	**Data_Type**	**Key Constraints**
1	Farmer_id	INT	Primary Key, Auto Increment
2	Login_id	INT	Foreign Key
3	First_name	VARCHAR	NOT NULL
4	Last_name	VARCHAR	NOT NULL
5	Birth_date	VARCHAR	NOT NULL
6	email	VARCHAR	NOT NULL
7	Profile_pic	VARCHAR	NOT NULL
8	Gender	VARCHAR	NOT NULL
9	Annual_income	INT	NOT NULL
10	Land	VARCHAR	NOT NULL
11	House_no	INT	NOT NULL
12	Address	VARCHAR	NOT NULL
13	Phone_number	INT	NOT NULL
14	Ward	VARCHAR	NOT NULL
15	Pin_code	INT	NOT NULL
16	File_upload	VARCHAR	NOT NULL
17	Profile_photo	VARCHAR	NOT NULL

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software testing is a methodical process where a software program is meticulously evaluated to ensure it functions as intended. This process incorporates various verification and validation techniques to assess the software's compliance with specifications. Verification encompasses activities like reviews, analyses, inspections, and walkthroughs. Meanwhile, validation involves examining the software to ensure it aligns with the desired outcomes.

Two key aspects of software testing are static analysis and dynamic analysis. Static analysis delves into the software's source code to pinpoint potential issues, whereas dynamic analysis observes its behavior during runtime, collecting valuable data like execution traces, timing profiles, and test coverage details.

Key aspects of system testing include:

Functional Testing: Verifying that the software performs its intended functions correctly, addressing features, and requirements as specified in the design and requirements documents.

Non-Functional Testing: Evaluating non-functional aspects, such as performance, scalability, security, and usability to ensure the system's quality and robustness.

Integration Testing: Assessing how different components or modules within the system work together and interface seamlessly.

End-to-End Testing: Testing the system as a whole to ensure that data and processes flow smoothly from end to end.

Regression Testing: Ensuring that system changes or updates do not introduce new defects or disrupt existing functionalities.

System testing is performed after unit and integration testing and serves as a comprehensive assessment before the software's release to ensure that it meets user expectations and performs reliably in its operational environment.

5.2 TEST PLAN

A test plan is a formal document that defines the scope, objectives, resources, schedule, and approach for testing a software system. It outlines the testing strategy, test cases, and criteria for success, serving as a guide for testers, developers, and stakeholders. The primary goal of a test plan is to ensure that the software is thoroughly tested to identify defects and validate that it meets its intended requirements and quality standards.

Key components of a test plan include:

Introduction: Provides an overview of the software under test, its purpose, and the objectives of testing.

Scope: Defines what aspects of the software will be tested and any items that are out of scope.

Testing Objectives: Specifies the goals of testing, including what needs to be achieved validated.

Test Strategy: Outlines the overall approach to testing, including the types of testing (e.g., functional, performance, security), and the testing environments.

Test Schedule: Provides a timeline for testing activities, including start and end dates for different testing phases.

Test Cases: Describes individual test cases, including input data, expected results, and dependencies.

Resources: Identifies the roles and responsibilities of team members, as well as the tools and infrastructure required for testing.

Risks and Mitigation: Highlights potential risks during testing and strategies to mitigate them.

Exit Criteria: Specifies the conditions that must be met to consider testing complete.

Approvals: Identifies stakeholders responsible for approving the test plan.

5.2.1 Unit Testing

Unit testing is a software testing technique where individual units or components of a software application are tested in isolation. A unit, in this context, is the smallest testable part of the software, typically a function, method, or class. The primary goal of unit testing is to verify that each unit of code performs as expected and meets its specified requirements.

During unit testing, the modular interface is tested to ensure that data enters and exits the software unit under test properly. The local data structure is inspected to ensure that data temporarily stored retains its integrity during each step of an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once, and all error handling paths are tested to ensure that the software can handle errors correctly.

5.2.2 Integration Testing

Integration testing is a software testing technique that evaluates the interactions and data flow between interconnected components, modules, or services within a software application. The primary objective is to identify and rectify issues that may arise when these units are combined. This testing phase ensures that various parts of the software work harmoniously, data is exchanged correctly, and interfaces between components function as intended.

5.2.3 Validation Testing or System Testing

Validation testing and system testing are both integral parts of the software testing process, each serving distinct purposes in ensuring software quality and reliability. Validation testing, often referred to as acceptance testing, is the process of evaluating a software system to determine whether it meets the specified requirements and satisfies the needs of the end users or stakeholders. It focuses on validating that the software performs as intended and delivers the expected functionality.

5.2.4 Output Testing or User Acceptance Testing

User acceptance testing is performed to ensure that the system meets the business requirements and user needs. It is important to involve the end users during the development process to ensure that the software aligns with their needs and expectations. During user acceptance testing, the input and output screen designs are tested with different types of test data. The preparation of test data is critical to ensure comprehensive testing of the system. Any errors identified during testing are addressed and corrected, and the corrections are noted for future reference.

5.2.5 Automation Testing

Automation Testing is a critical aspect of software testing that involves using automated tools and scripts to perform tests on a software application.
Automation testing is the process of using automation tools, frameworks, and scripts to execute test cases and verify the functionality of a software application automatically.

5.2.6 Selenium Testing

Selenium is an open-source automated testing framework used to verify web applications across different browsers and platforms. Selenium allows for the creation of test scripts in various programming languages such as Java, C#, and Python. Jason Huggins, an engineer at Thought Works, developed Selenium in 2004 while working on a web application that required frequent testing. He created a JavaScript program called "JavaScriptTestRunner" to automate browser actions and improve testing efficiency. Selenium has since evolved and continues to be developed by a team of contributors.

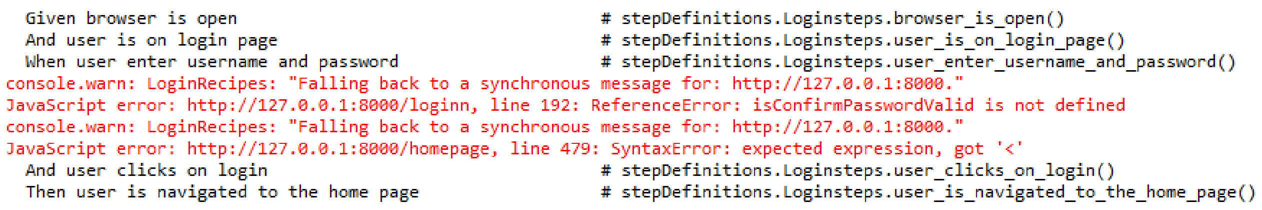
In addition to Selenium, another popular tool used for automated testing is Cucumber. Cucumber is an open-source software testing framework that supports behavior-driven development (BDD). It allows for the creation of executable specifications in a human-readable format called Gherkin. One of the advantages of using Cucumber is its ability to bridge the gap between business stakeholders and technical teams. By using a common language, Cucumber facilitates effective communication and collaboration during the testing process. It promotes a shared understanding of the requirements and helps ensure that the developed software meets the intended business goals.

Test Case 1:

Code

```
package stepDefinitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class Loginsteps {
WebDriver driver=null;
@Given("browser is open")
public void browser_is_open() {
System.out.println("Inside step browser");
System.out.printf("webdirver.gecko.marionette","C:\\Users\\user\\eclipse-
workspace\\pro1\\src\\test\\resources\\drivers\\geckodriver.exe");
driver=new FirefoxDriver();}
@And("user is on login page")
public void user_is_on_login_page() throws Exception {
driver.navigate().to("http://127.0.0.1:8000/loginn");
}
@When("user enter username and password")
public void user_enter_username_and_password() throws Exception {
driver.findElement(By.id("email")).sendKeys("anna@gmail.com");
driver.findElement(By.id("password")).sendKeys("Anna@123");}
@And("user clicks on login")
public void user_clicks_on_login() throws Exception {
driver.findElement(By.id("submit")).click();}
@Then("user is navigated to the home page")
public void user_is_navigated_to_the_home_page() throws Exception {
driver.findElement(By.className("header")).isDisplayed();
Thread.sleep(2000);
driver.close();
driver.quit();
}}
```

Screenshot



```
  Given browser is open                                      # stepDefinitions.Loginsteps.browser_is_open()
  And user is on login page                                  # stepDefinitions.Loginsteps.user_is_on_login_page()
  When user enter username and password                      # stepDefinitions.Loginsteps.user_enter_username_and_password()
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
JavaScript error: http://127.0.0.1:8000/loginn, line 192: ReferenceError: isConfirmPasswordValid is not defined
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
JavaScript error: http://127.0.0.1:8000/homepage, line 479: SyntaxError: expected expression, got '<'
  And user clicks on login                                   # stepDefinitions.Loginsteps.user_clicks_on_login()
  Then user is navigated to the home page                    # stepDefinitions.Loginsteps.user_is_navigated_to_the_home_page()
```

Fig:14 Selenium_Login

Test Report

Test Case 1

Project Name: CropElevate	
Login Test Case	
Test Case ID: 1	**Test Designed By:** Aleena Joseph
Test Priority (Low/Medium/High): High	**Test Designed Date:** 29/09/2023
Module Name: Login Page	**Test Executed By:** Mr. Ajith G.S
Test Title: User Login	**Test Execution Date:** 29/09/2023
Description: Verify the login page with valid email and password	
Pre-Condition: User has valid username and password	

Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)
1	Navigate to login page		Dashboard should displayed.	Login page is displayed	Pass
2	Provide valid email	anna@gmail.com	User should be able to login	User logged in the user dashboard is displayed	pass
3	Provide valid password	Anna@123			
4	Clicks on login button				
5	Navigate to homepage				pass

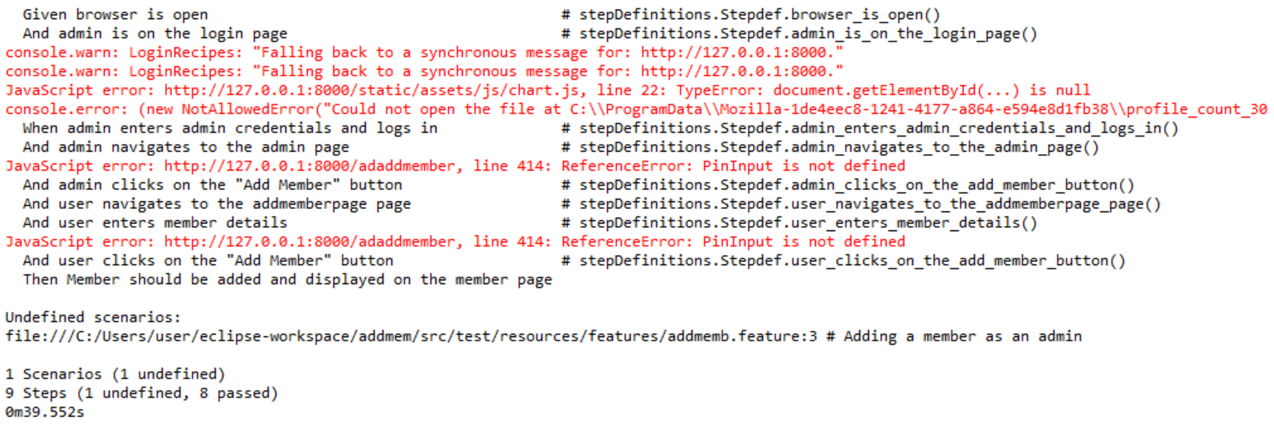
Post-Condition: Successfully Logged in

Test Case 2:

Code

```
package stepDefinitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Then;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Stepdef {
private WebDriver driver; // Initialize your WebDriver instance
@Given("browser is open")
public void browser_is_open() {
System.out.println("Inside Step - Browser Is Open");
System.setProperty("webdriver.gecko.marionette", "C:\\Users\\user\\eclipse-
workspace\\addmem\\src\\test\\resources\\drivers\\geckodriver.exe");
driver = new FirefoxDriver();
driver.manage().window().maximize();
}
@And("admin is on the login page")
public void admin_is_on_the_login_page() throws Exception{
driver.navigate().to("http://127.0.0.1:8000/adminlogin");
Thread.sleep(3000);
}
@When("admin enters admin credentials and logs in")
public void admin_enters_admin_credentials_and_logs_in()throws Throwable {
driver.findElement(By.name("email")).sendKeys("admin123@gmail.com");
driver.findElement(By.name("password")).sendKeys("Admin@123");
driver.findElement(By.id("login")).click();
Thread.sleep(3000);
}
@And("admin navigates to the admin page")
public void admin_navigates_to_the_admin_page()throws Exception
{
Thread.sleep(3000);
driver.findElement(By.id("memind")).click();
}
@And("admin clicks on the \"Add Member\" button")
public void admin_clicks_on_the_add_member_button() throws InterruptedException
{
driver.findElement(By.id("addmem")).click();
Thread.sleep(3000);
}
@And("user navigates to the addmemberpage page")
public void user_navigates_to_the_addmemberpage_page() throws
InterruptedException {
Thread.sleep(3000);
}
@And("user enters member details")
public void user_enters_member_details() throws InterruptedException {
driver.findElement(By.name("name")).sendKeys("Ravi");
driver.findElement(By.name("email")).sendKeys("ravi1@gmail.com");
driver.findElement(By.name("password")).sendKeys("Ravi@123");
driver.findElement(By.name("address")).sendKeys("ABC Villa");
driver.findElement(By.name("dis")).sendKeys("Kottayam");
driver.findElement(By.name("taluk")).sendKeys("Kanjirappally");
driver.findElement(By.name("panchayat")).sendKeys("Erumely");
driver.findElement(By.name("wardno")).sendKeys("4");
driver.findElement(By.name("wardname")).sendKeys("Cheruvally Estate");
driver.findElement(By.name("pin")).sendKeys("686504");
driver.findElement(By.name("phone")).sendKeys("8896329685");
driver.findElement(By.name("bio")).sendKeys("Ward Member Of The Ward 4
Cheruvally Estate");
}
@And("user clicks on the \"Add Member\" button")
public void user_clicks_on_the_add_member_button(){
driver.findElement(By.id("createmem")).click();
}
@Then("member should be added and displayed on the member page")
public void member_should_be_added_and_displayed_on_the_member_page()
throws InterruptedException {        // Implement code to verify that the added category
is displayed on the member page
driver.findElement(By.id("memberlist")).isDisplayed();
Thread.sleep(2000);
driver.close();
driver.quit();
}
}
```

Screenshot



```
  Given browser is open                                # stepDefinitions.Stepdef.browser_is_open()
  And admin is on the login page                       # stepDefinitions.Stepdef.admin_is_on_the_login_page()
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
JavaScript error: http://127.0.0.1:8000/static/assets/js/chart.js, line 22: TypeError: document.getElementById(...) is null
console.error: (new NotAllowedError("Could not open the file at C:\\ProgramData\\Mozilla-1de4eec8-1241-4177-a864-e594e8d1fb38\\profile_count_30
  When admin enters admin credentials and logs in      # stepDefinitions.Stepdef.admin_enters_admin_credentials_and_logs_in()
  And admin navigates to the admin page                # stepDefinitions.Stepdef.admin_navigates_to_the_admin_page()
JavaScript error: http://127.0.0.1:8000/adaddmember, line 414: ReferenceError: PinInput is not defined
  And admin clicks on the "Add Member" button          # stepDefinitions.Stepdef.admin_clicks_on_the_add_member_button()
  And user navigates to the addmemberpage page         # stepDefinitions.Stepdef.user_navigates_to_the_addmemberpage_page()
  And user enters member details                       # stepDefinitions.Stepdef.user_enters_member_details()
JavaScript error: http://127.0.0.1:8000/adaddmember, line 414: ReferenceError: PinInput is not defined
  And user clicks on the "Add Member" button           # stepDefinitions.Stepdef.user_clicks_on_the_add_member_button()
  Then Member should be added and displayed on the member page

Undefined scenarios:
file:///C:/Users/user/eclipse-workspace/addmem/src/test/resources/features/addmemb.feature:3 # Adding a member as an admin

1 Scenarios (1 undefined)
9 Steps (1 undefined, 8 passed)
0m39.552s
```

Fig:15 Selenium_Add_Member

Test report

Test Case 2

Project Name: CropElevate	
Add Member Test Case	
Test Case ID: 2	**Test Designed By:** Aleena Joseph
Test Priority (Low/Medium/High): High	**Test Designed Date:** 29/09/2023
Module Name: Add Member Page	**Test Executed By:** Mr. Ajith G.S
Test Title: Adding Member	**Test Execution Date:** 29/09/2023
Description: To add a member by the admin	
Pre-Condition: admin has valid username and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Admin is on login page		Dashboard should displayed.	Login page is displayed	Pass
2	Provide the Email	Admin123@gmail.com	Admin login	Admin logged in the admin dashboard is displayed	pass
3	Provide valid password	Admin@123			
4	Admin navigates to admin page				
5	Admin clicks on add member button.		Admin click on add member button Admin enters the member details	Admin clicks on add member button Admin enters the member details	pass
6	Navigates to add member page		clicks on the create member button the member is created	clicks on the create member button the member is created	Pass

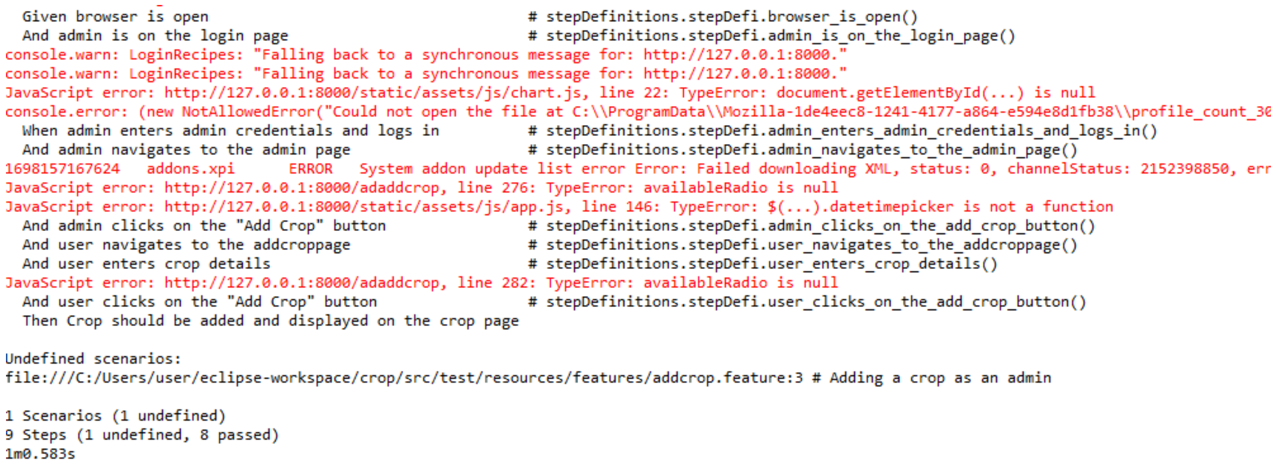
7	Enter the member details				
8	Click on add member button.		Member is displayed on the Member Page	Member is displayed on the Member Page	
9	Member should be added and displayed in the admin member page				
Post-Condition: The member is created successfully					

Test Case 3:

Code

```
package stepDefinitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Then;
import org.openqa.selenium.firefox.FirefoxDriver;
public class stepDefi {
private WebDriver driver; // Initialize your WebDriver instance
@Given("browser is open")
public void browser_is_open() {
System.out.println("Inside Step - Browser Is Open");
System.setProperty("webdriver.gecko.marionette",                "C:\\Users\\user\\eclipse-
workspace\\crop\\src\\test\\resources\\drivers\\geckodriver.exe");
driver = new FirefoxDriver();
driver.manage().window().maximize();}
@And("admin is on the login page")
public void admin_is_on_the_login_page() throws Exception{
driver.navigate().to("http://127.0.0.1:8000/adminlogin");
Thread.sleep(3000);}
@When("admin enters admin credentials and logs in")
public void admin_enters_admin_credentials_and_logs_in()throws Throwable {
driver.findElement(By.name("email")).sendKeys("admin123@gmail.com");
driver.findElement(By.name("password")).sendKeys("Admin@123");
driver.findElement(By.id("login")).click();
Thread.sleep(3000);
}
@And("admin navigates to the admin page")
public void admin_navigates_to_the_admin_page()throws Exception
{
Thread.sleep(3000);
driver.findElement(By.id("cid")).click();
}
@And("admin clicks on the \"Add Crop\" button")
public void admin_clicks_on_the_add_crop_button() throws InterruptedException {
driver.findElement(By.id("addcro")).click();
Thread.sleep(3000);
}
@And("user navigates to the addcroppage")
public void user_navigates_to_the_addcroppage() throws InterruptedException {
Thread.sleep(3000);
}
@And("user enters crop details")
public void user_enters_crop_details() throws InterruptedException {
driver.findElement(By.name("cname")).sendKeys("Mango");
driver.findElement(By.name("cdescription")).sendKeys("Mango Plant ");
driver.findElement(By.name("start_date")).sendKeys("2023-10-15");
driver.findElement(By.name("end_date")).sendKeys("2023-10-28");
driver.findElement(By.name("count")).sendKeys("200");
}
@And("user clicks on the \"Add Crop\" button")
public void user_clicks_on_the_add_crop_button(){
driver.findElement(By.id("createcrop")).click();}
@Then("crop should be added and displayed on the crop page")
public void crop_should_be_added_and_displayed_on_the_crop_page() throws
InterruptedException {            // Implement code to verify that the added category is
displayed on the member page
driver.findElement(By.id("croplist")).isDisplayed();
Thread.sleep(2000);
driver.close();
driver.quit();
}}
```

Screenshot



```
  Given browser is open                                      # stepDefinitions.stepDefi.browser_is_open()
  And admin is on the login page                             # stepDefinitions.stepDefi.admin_is_on_the_login_page()
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
JavaScript error: http://127.0.0.1:8000/static/assets/js/chart.js, line 22: TypeError: document.getElementById(...) is null
console.error: (new NotAllowedError("Could not open the file at C:\\ProgramData\\Mozilla-1de4eec8-1241-4177-a864-e594e8d1fb38\\profile_count_3
  When admin enters admin credentials and logs in            # stepDefinitions.stepDefi.admin_enters_admin_credentials_and_logs_in()
  And admin navigates to the admin page                      # stepDefinitions.stepDefi.admin_navigates_to_the_admin_page()
1698157167624   addons.xpi      ERROR   System addon update list error Error: Failed downloading XML, status: 0, channelStatus: 2152398850, err
JavaScript error: http://127.0.0.1:8000/adaddcrop, line 276: TypeError: availableRadio is null
JavaScript error: http://127.0.0.1:8000/static/assets/js/app.js, line 146: TypeError: $(...).datetimepicker is not a function
  And admin clicks on the "Add Crop" button                  # stepDefinitions.stepDefi.admin_clicks_on_the_add_crop_button()
  And user navigates to the addcroppage                      # stepDefinitions.stepDefi.user_navigates_to_the_addcroppage()
  And user enters crop details                               # stepDefinitions.stepDefi.user_enters_crop_details()
JavaScript error: http://127.0.0.1:8000/adaddcrop, line 282: TypeError: availableRadio is null
  And user clicks on the "Add Crop" button                   # stepDefinitions.stepDefi.user_clicks_on_the_add_crop_button()
  Then Crop should be added and displayed on the crop page

Undefined scenarios:
file:///C:/Users/user/eclipse-workspace/crop/src/test/resources/features/addcrop.feature:3 # Adding a crop as an admin

1 Scenarios (1 undefined)
9 Steps (1 undefined, 8 passed)
1m0.583s
```

Fig:16 Selenium_Add_Crop

Test Case 3

Project Name: CropElevate					
Add Crop Test Case					
Test Case ID: 3			**Test Designed By:** Aleena Joseph		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 29/09/2023		
Module Name: Add crop Page			**Test Executed By:** Mr. Ajith G.S		
Test Title: Adding Crop			**Test Execution Date:** 29/09/2023		
Description: To add a crop by the admin					
Pre-Condition: Admin has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Admin is on login page		Dashboard should displayed.	Login page is displayed	Pass
2	Provide the Email	Admin123@gmail.com	Admin login	Admin logged in the admin dashboard is displayed	pass
3	Provide valid password	Admin@123			
4	Admin navigates to admin page				
5	Admin clicks on add crop button.		Admin click on add crop button and Admin enters the crop details	Admin clicks on add crop button and Admin enters the crop details	pass
6	Navigates to add crop page		clicks on the create crop button the crop is created	clicks on the create crop button the crop is created	Pass
7	Enter the crop details				
8	Click on add crop button.		Crop is displayed in the crop page	Crop is displayed in the crop page	Pass

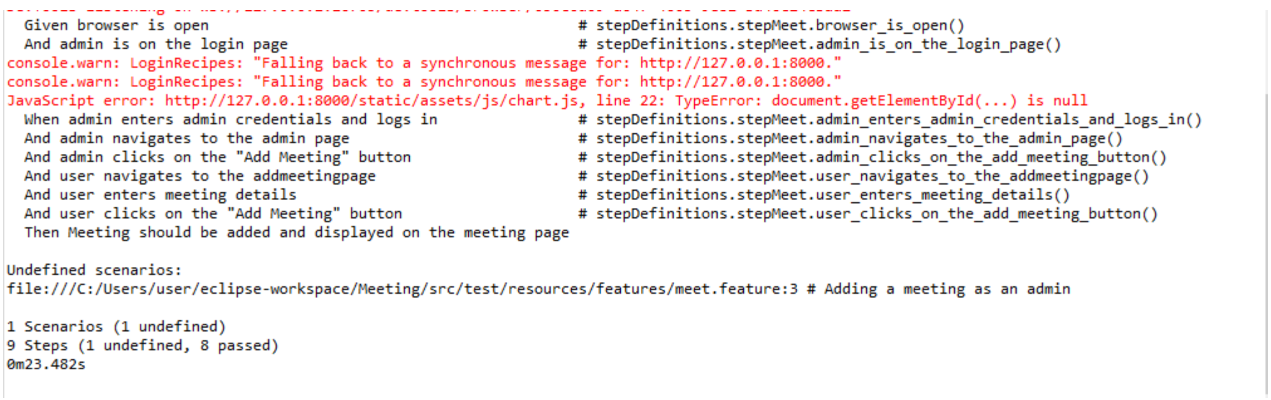
9	Crop should be added and displayed in the admin crop page				
Post-Condition: Crop is added successfully					

Test Case 4:

Code

```
package stepDefinitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Then;
import org.openqa.selenium.firefox.FirefoxDriver;
public class stepMeet {
private WebDriver driver; // Initialize your WebDriver instance
@Given("browser is open")
public void browser_is_open() {
System.out.println("Inside Step - Browser Is Open");
System.setProperty("webdriver.gecko.marionette", "C:\\Users\\user\\eclipse-
workspace\\crop\\src\\test\\resources\\drivers\\geckodriver.exe");
driver = new FirefoxDriver();
driver.manage().window().maximize();}
@And("admin is on the login page")
public void admin_is_on_the_login_page() throws Exception{
driver.navigate().to("http://127.0.0.1:8000/adminlogin");
Thread.sleep(3000);
}
@When("admin enters admin credentials and logs in")
public void admin_enters_admin_credentials_and_logs_in()throws Throwable {
driver.findElement(By.name("email")).sendKeys("admin123@gmail.com");
driver.findElement(By.name("password")).sendKeys("Admin@123");
driver.findElement(By.id("login")).click();
Thread.sleep(3000);}
@And("admin navigates to the admin page")
public void admin_navigates_to_the_admin_page()throws Exception
{
Thread.sleep(3000);
driver.findElement(By.id("met")).click();
driver.findElement(By.id("meetid")).click();
}
@And("admin clicks on the \"Add Meeting\" button")
public void admin_clicks_on_the_add_meeting_button() throws InterruptedException {
driver.findElement(By.id("addmem")).click();
Thread.sleep(3000);
}
@And("user navigates to the addmeetingpage")
public void user_navigates_to_the_addmeetingpage() throws InterruptedException {
Thread.sleep(3000);}
@And("user enters meeting details")
public void user_enters_meeting_details() throws InterruptedException {
driver.findElement(By.name("meeting_date")).sendKeys("2023-11-05");
driver.findElement(By.name("meeting_time")).sendKeys("11:00:00");
driver.findElement(By.name("meeting_venue")).sendKeys("Erumely Panchayath");
driver.findElement(By.name("meeting_agenda")).sendKeys("Meeting with the ward members");
}
@And("user clicks on the \"Add Meeting\" button")
public void user_clicks_on_the_add_meeting_button(){
driver.findElement(By.id("createmeet")).click();
}
@Then("meeting should be added and displayed on the meeting page")
public void meeting_should_be_added_and_displayed_on_the_meeting_page() throws
InterruptedException {        // Implement code to verify that the added category is displayed on the
member page
driver.findElement(By.id("meetinglist")).isDisplayed();
Thread.sleep(2000);
driver.close();
driver.quit();
}
}
```

Screenshot



```
  Given browser is open                                    # stepDefinitions.stepMeet.browser_is_open()
  And admin is on the login page                           # stepDefinitions.stepMeet.admin_is_on_the_login_page()
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
JavaScript error: http://127.0.0.1:8000/static/assets/js/chart.js, line 22: TypeError: document.getElementById(...) is null
  When admin enters admin credentials and logs in          # stepDefinitions.stepMeet.admin_enters_admin_credentials_and_logs_in()
  And admin navigates to the admin page                    # stepDefinitions.stepMeet.admin_navigates_to_the_admin_page()
  And admin clicks on the "Add Meeting" button             # stepDefinitions.stepMeet.admin_clicks_on_the_add_meeting_button()
  And user navigates to the addmeetingpage                 # stepDefinitions.stepMeet.user_navigates_to_the_addmeetingpage()
  And user enters meeting details                          # stepDefinitions.stepMeet.user_enters_meeting_details()
  And user clicks on the "Add Meeting" button              # stepDefinitions.stepMeet.user_clicks_on_the_add_meeting_button()
  Then Meeting should be added and displayed on the meeting page

Undefined scenarios:
file:///C:/Users/user/eclipse-workspace/Meeting/src/test/resources/features/meet.feature:3 # Adding a meeting as an admin

1 Scenarios (1 undefined)
9 Steps (1 undefined, 8 passed)
0m23.482s
```

Fig:17 Selenium_Meeting

Test Case 4	
Project Name: CropElevate	
Meeting Test Case	
Test Case ID: 4	**Test Designed By:** Aleena Joseph
Test Priority (Low/Medium/High): High	**Test Designed Date:** 29/09/2023
Module Name: Add meeting Page	**Test Executed By:** Mr. Ajith G.S

Test Title: Adding meeting			**Test Execution Date:** 29/09/2023		
Description: To add a meeting by the admin					
Pre-Condition: Admin has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actua l Result**	**Status(Pass/ Fail)**
1	Admin is on login page		Dashboard should displayed.	Login page is displayed	Pass
2	Provide the Email	Admin123@ gmail.com	Admin login	Admin logged in and the admin dashboard is displayed	pass
3	Provide valid password	Admin@123			
4	Admin navigates to admin page				
5	Admin clicks on add meeting button.		Admin click on add meeting button and Admin enters the meeting details	Admin clicks on add meeting button and Admin enters the meeting details	pass
6	Navigates to add meeting page		clicks on the create meeting button the meeting is created	clicks on the create meeting button the meeting is created	Pass
7	Enter the meeting details		Meeting is displayed in the meeting page	Meeting is displayed in the meeting page	Pass
8	Click on add meeting button.				
9	Meeting should be added and displayed in the admin meeting page				
Post-Condition: Meeting created successfully					

CHAPTER 6
IMPLEMENTATION

6.1 INTRODUCTION

The implementation phase of a project is a critical stage in the process of turning a design into a fully functional system. During this phase, the primary goal is to instill confidence in users that the system will work effectively and accurately. Key considerations during implementation include user training and documentation. In some cases, the conversion process may run in parallel with user training or occur at a later stage.

This phase places a significant workload on the user department, leading to substantial changes and a major impact on the existing system. Poorly planned or uncontrolled implementation can result in confusion and chaos. Whether the new system is entirely novel, replacing an existing manual or automated system, or modifying an existing one, proper implementation is essential to meet the organization's needs.

System implementation encompasses all the activities required to transition from the old system to the new one. It's essential to ensure that thorough testing has been conducted to verify that the system functions according to the specified requirements. System personnel play a crucial role in evaluating the feasibility of the system. Implementation demands extensive effort in three main areas: education and training, system testing, and changeover. Careful planning, investigation of system and constraints, and the design of methods to facilitate the changeover are all integral aspects of the implementation phase.

6.2 IMPLEMENTATION PROCEDURES

Software implementation is a crucial phase in the software development process, involving the installation of the software in its designated environment to ensure that it functions as intended. In some cases, the software may be developed for users who are not directly involved in its creation. At the outset, there may be apprehensions and doubts about the new software, but it's essential to mitigate any potential resistance. This can be achieved through the following measures:

User Awareness: Actively inform users about the advantages and benefits of the new system. Building their confidence in the software's capabilities is key to its successful adoption.

User Guidance: Provide comprehensive guidance and training to users to make them comfortable with the application. Ensuring that users are well-versed in the software's usage is essential for a smooth transition.

It's important to note that for the software to function as intended, users should be aware that the server program must be operational on the server. Without the server component running, the desired processes will not take place. Effective software implementation is critical to ensure that the software meets its intended purpose and contributes to organizational success.

6.2.1 User Training

User training serves the crucial purpose of preparing individuals for system testing and the transition to a new system. To realize the anticipated advantages of computer-based systems, it is imperative that those involved have a strong grasp of their roles within the new framework. As system complexity increases, the necessity for training becomes more pronounced. Through user training, individuals acquire the necessary skills to input data accurately, effectively address error messages, proficiently query the database, and execute routines for generating reports and carrying out other essential functions. In essence, user training empowers individuals to confidently navigate and utilize the system, ensuring its successful adoption and the attainment of its objectives

6.2.2 Training on the Application Software

After users have received basic computer awareness training, they must be instructed on how to use the new application software. This thorough training should cover the system's underlying philosophy as well as the practical aspects of using the software. This entails being aware of the logical sequence of screens, screen design principles, the various kinds of on-screen assistance that are offered, foreseeing and resolving possible data input errors, and the related validation checks for every data entry. Additionally, users should have the skills and resources needed to correct any inaccurate data. Additionally, this training ought to be tailored to the individual requirements of every user or group, taking into account their particular needs for efficient system use.

6.2.3 System Maintenance

The maintenance phase plays a pivotal role in the software development cycle, serving as the period when software is actively utilized to fulfill its intended purposes. Effective maintenance is of paramount importance to ensure that the system retains its functionality, reliability, and adaptability to evolving environmental conditions. Maintenance activities encompass more than just identifying and rectifying errors or glitches; they may also entail software updates, functional modifications, and performance enhancements, among other tasks. In essence, software maintenance is a continuous, ongoing process that demands constant scrutiny, assessment, and refinement of the system to align with the ever-changing needs and demands of its users.

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In conclusion, the envisioned management system represents a significant step forward in supporting and empowering farmers in their agricultural endeavors. By facilitating the exchange of crops and providing personalized crop recommendations, it addresses the critical need for improved productivity and efficient resource utilization. The collaboration between farmers, Panchayath Secretaries, and Ward Members ensures data accuracy and verification, fostering trust and community involvement. This system exemplifies the potential of technology to bridge the gap between local governance bodies and farmers, enhancing decision-making in agriculture. Ultimately, it stands as a promising solution that not only streamlines the process of crop allocation but also serves as a model for future initiatives aimed at improving the lives and livelihoods of farmers.

7.2 FUTURE SCOPE

The future scope of the proposed crop exchange and recommendation management system is vast and holds great potential. Firstly, the system can be integrated with emerging technologies such as artificial intelligence and machine learning to continually enhance its crop recommendation algorithms. By analyzing historical data, local conditions, and farmer feedback, the system can evolve to provide even more accurate and tailored crop suggestions. Furthermore, the platform can extend its services to include comprehensive agricultural information, pest and disease management guidance, and weather forecasts to equip farmers with a holistic support system for their farming needs.

Secondly, the system can serve as a foundation for fostering agricultural communities and collaborations. It could facilitate knowledge sharing among farmers, organize cooperative buying or selling groups, and enable collective decision-making. Moreover, by incorporating financial services like microloans and insurance options, it can help farmers manage risks and improve their financial stability. The system's future scope also involves partnerships with research institutions and government agencies to leverage their resources and expertise in the quest for sustainable and innovative agricultural practices, thus ensuring the long-term success and resilience of the agricultural sector.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- PankajJalote, “Software engineering: a precise approach”
- Roger S Pressman, “Software Engineering”

WEBSITES:

- www.w3schools.com
- www.bootstrap.com
- https://chat.openai.com/chat
- www.jquery.com

CHAPTER 9
APPENDIX

9.1 Sample Code

HTML

Login

```
<!DOCTYPE html>
<html>
<head>
<title>LOGIN</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<style>
h1
{
color:white;
}
*{
margin: 0;
padding: 0;
box-sizing: border-box;
font-family: 'Poppins', sans-serif;
}
.image-container
{
background-size: cover;
min-height: 80vh;
display: flex;
align-items: center;
justify-content: center;
}
.form-container
{
background-image:linear-gradient(rgb(153, 245, 93), rgb(241, 243, 145));
padding: 50px;
border-radius: 20px;
align-items:center;
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.8);
}
.form-link
{
text-align: center;
margin-top: 10px;
}
.form-link span,
.form-link a
{
font-size: 14px;
font-weight: 400;
color: blue;
}
```

```
.image-container img
{
max-width: 100%;
height: auto;
display: block;
}
</style>
</head>
<body>
<div class="container-fluid">
<div class="row">
<div class="col-md-6 image-container">
<img src="https://i.pinimg.com/736x/0b/da/44/0bda44489659f5dc3b00971f59e5c419.jpg"
alt="Image">
</div>
<div class="col-md-6">
<div class="container form-container mt-5">
<h1>Login Form</h1>
{% for messages in messages %}
<h3 style="color: red">{{ messages }}</h3>
{% endfor %}
<form action="#" id="registrationForm" method="POST">
{% csrf_token %}
<div class="form-group">
<label for="email">Email</label>
<input type="email" class="form-control" placeholder="Enter Email" id="email" name="email"
required>
<small id="emailError" class="form-text text-danger"></small>
</div>
<!-- <div class="form-group">
<label for="username">Username</label>
<input type="text" class="form-control" id="username" name="username" placeholder="Enter
Username" required>
<small id="usernameError" class="form-text text-danger"></small>
</div> -->
<div class="form-group">
<label for="password">Password</label>
<input type="password" class="form-control" id="password" placeholder="Enter Password"
name="password" required>
<small id="passwordError" class="form-text text-danger"></small>
</div>
{% comment %} <form action="{% url 'password_reset' %}" method="post">
{% csrf_token %}
<div class="form-group">
<a href="{% url 'password_reset' %}" class="forgot-password">Forgot Password?</a>
</div>
</form> {% endcomment %}
<button type="submit" id="submit" class="btn btn-primary">Submit</button>
<pre>Don't Have an Account? <a href="signup" style="color:blue; font-size:
14px;">Signup</a></pre
</form>
```

```
</div>
</div>
</div>
</div>
</div>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<script>
const registrationForm = document.getElementById('registrationForm');
const emailInput = document.getElementById('email');
const passwordInput = document.getElementById('password');
function validateEmail() {
const emailValue = emailInput.value;
const emailError = document.getElementById('emailError');
if (!/^\S+@\S+\.\S+$/.test(emailValue)) {
emailError.textContent = 'Enter a valid email address.';
return false;
} else {
emailError.textContent = '';
return true;
}
}
function validatePassword() {
const passwordValue = passwordInput.value;
const passwordError = document.getElementById('passwordError');
if (!/(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*\W).{8,}/.test(passwordValue)) {
passwordError.textContent = 'Password should contain at least 8 characters, one uppercase letter,
one lowercase letter, one number, and one special character.';
return false;
} else {
passwordError.textContent = '';
return true;
}
}
registrationForm.addEventListener('submit', function(event) {
const isEmailValid = validateEmail();
const isPasswordValid = validatePassword();
if ( !isEmailValid || !isPasswordValid || !isConfirmPasswordValid) {
event.preventDefault();
}
});
</script>
</body>
</html>
```

Admin_Dashboard

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0">
<title>CropElevate</title>
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/bootstrap.min.css' %}">
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/font-awesome.min.css' %}">
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/style.css' %}">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.0/css/all.min.css">
<style>
.header
{
background-color:green;
}
.header .header-left
{
background-color:green;
}
.green-background {
background-color: green;
}
{% comment %} .sidebar{
background-color: green;
} {% endcomment %}
.sidebar
{
background-color:green;
}
.sidebar-menu li a {
color:white;
}
.sidebar-menu li a:hover {
color: white;
}
.sidebar-menu li.active a {
color: green;
background-color: #f3f3f3;
}
.table{
margin-left:80px;
width:450px;
}
.card
{
width:70% ;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.8);
}
.card member-panel
{
margin-left:40px;
}
.card2
{
width:75% ;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.8);
margin-left:900px;
margin-top:-480px;
}
</style>
<!--[if lt IE 9]>
<script src="assets/js/html5shiv.min.js"></script>
<script src="assets/js/respond.min.js"></script>
<![endif]-->
</head>
<body>
<div class="main-wrapper">
<div class="header">
<div class="header-left">
<a href="{% url 'adindex' %}" class="logo">
<img src="{% static 'assets/img/adminlogo1.png' %}" width="35" height="35" alt="">
<span>CropElevate</span>
</a>
</div>
<a id="toggle_btn" href="javascript:void(0);"><i class="fa fa-bars"></i></a>
<a id="mobile_btn" class="mobile_btn float-left" href="#sidebar"><i class="fa fa-bars"></i></a>
<ul class="nav user-menu float-right">
<li class="nav-item dropdown d-none d-sm-block">                  <div class="dropdown-menu
notifications">
<div class="topnav-dropdown-header">
<span>Notifications</span>
</div>
<div class="drop-scroll">
<ul class="notification-list">
</ul>
</div>
</div>
</li>
<li class="nav-item dropdown d-none d-sm-block">
<a href="javascript:void(0);" id="open_msg_box" class="hasnotifications nav-link"><i class="fa
fa-comment-o"></i> <span class="badge badge-pill bg-danger float-right"></span></a>
</li>
<li class="nav-item dropdown has-arrow">
<a href="#" class="dropdown-toggle nav-link user-link" data-toggle="dropdown">
<span class="user-img">
<img class="rounded-circle" src="{% static 'assets/img/user.jpg' %}" width="24" alt="Admin">
<span class="status online"></span>
```

```
</span>
{% comment %} <span>Admin</span> {% endcomment %}
</a>
<div class="dropdown-menu">
<a class="dropdown-item" href="{% url 'adminlogin' %}">Logout</a>
</div>
</li>
</ul>
<div class="dropdown mobile-user-menu float-right">
<a href="#" class="dropdown-toggle" data-toggle="dropdown" aria-expanded="false"><i
class="fa fa-ellipsis-v"></i></a>
<div class="dropdown-menu dropdown-menu-right">
<a class="dropdown-item" href="{% url 'adminlogin' %}">Logout</a>
</div>
</div>
</div>
<div class="sidebar" id="sidebar">
<div class="sidebar-inner slimscroll">
<div id="sidebar-menu" class="sidebar-menu">
<ul>
<li class="menu-title" style="color:#c9edbe; font-weight:bold;">Admin</li>
<li class="active">
<a href="{% url 'adindex' %}"><i class="fa fa-dashboard"></i> <span>Dashboard</span></a>
</li>
<li>
<a href="{% url 'admember' %}" id="memind"><i class="fa fa-user"></i>
<span>Members</span></a>
</li>
<li>
<a href="{% url 'adcrop' %}" id="cid"><i class="fas fa-seedling fa-2x"></i>
<span>Crops</span></a>
</li>
<li class="submenu green-background"> <!-- Add the "green-background" class -->
<a href="#"><i class="fa fa-user"></i> <span></span>Applied <span class="menu-
arrow"></span></a>
<ul style="display: none;background-color:green;">
<li>
<a href="{% url 'adpendingapproval' %}"><i class="fas fa-exclamation-circle"></i>
</i> <span>Pending</span></a>
</li>
<li>
<a href="{% url 'adapproval' %}"><i class="fa fa-calendar-check-o"></i> <span>Approved
List</span></a>
</li>
<li>
<a href="{% url 'adpendinglist' %}"><i class="fas fa-hourglass-half"></i>
</i> <span>Waiting List</span></a>
</li>
</ul>
</li>
<li class="submenu green-background"> <!-- Add the "green-background" class -->
```

```
<a href="#"><i class="fas fa-users fa-2x"></i> <span></span> Meeting<span class="menu-
arrow"></span></a>
<ul style="display: none;background-color:green;">
<li>
<a href="{% url 'admeeting' %}"><i class="fas fa-users fa-2x"></i>
</i> <span>Meeting List</span></a>
</li>
<li>
<a href="{% url 'adattendance' %}"><i class="fa fa-calendar-check-o"></i>
<span>Attendance</span></a>
</li>
</ul>
</li>
<li>
<a href="{% url 'adreport' %}"><i class="fas fa-file-text"></i>
</i> <span>Report</span></a>
</li>
<li>
<a href="{% url 'adcalendar' %}"><i class="far fa-calendar-alt fa-2x"></i>
</i>
<span>Calendar</span></a>
</li>
</ul>
</div>
</div>
</div>
<div class="page-wrapper">
<div class="content">
<div class="row">
<div class="col-md-6 col-sm-6 col-lg-6 col-xl-3">
<div class="dash-widget">
<span class="dash-widget-bg1"><i class="fa fa-user-o" aria-hidden="true"></i></span>
<div class="dash-widget-info text-right">
<h3>{{me_count}}</h3>
<span class="widget-title1">Members <i class="fa fa-check" aria-hidden="true"></i></span>
</div>
</div>
</div>
<div class="col-md-6 col-sm-6 col-lg-6 col-xl-3">
<div class="dash-widget">
<span class="dash-widget-bg2"><i class="fa fa-calendar-check-o"></i></span>
<div class="dash-widget-info text-right">
<h3>{{c_count}}</h3>
<span class="widget-title2">Crops <i class="fa fa-check" aria-hidden="true"></i></span>
</div>
</div>
</div>
</div>
<div class="col-12 col-md-6 col-lg-8 col-xl-8">
<div class="card">
<div class="card-header">
```

```
<h4 class="card-title d-inline-block" style="font-weight:bold;">Crop List</h4>
<a href="{% url 'adcrop' %}" class="btn btn-secondary float-right mr-2">View All</a>
<!-- Add your button here -->
</div>
<div class="table-responsive ml-auto"> <!-- Added ml-auto class -->
<table class="table table-bordered table-striped"> <!-- Added table styles -->
<thead>
<tr>
<th>Crop Name</th>
<th>Crop Count</th>
</tr>
</thead>
<tbody>
{% for crop in crops %}
<tr>
<td>{{ crop.Namec }}</td>
<td>{{ crop.count }}</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<div class="col-12 col-md-6 col-lg-6 col-xl-6"> <!-- Create a column for the second card -->
<div class="card2">
<div class="card2-header"><br>
<h4 class="card-title d-inline-block" style="margin-left:15px; font-size:22px;">Member
List</h4>
<a href="{% url 'admember' %}" class="btn btn-secondary float-right mr-2" style="margin-
right:70px;">View all</a>
</div>
<div class="table-responsive ml-auto">
<table class="table table-bordered table-striped">
<thead>
<tr>
<th>Member Name</th>
<th>Ward</th>
</tr>
</thead>
<tbody>
{% for member in me %}
<tr>
<td>{{ member.Name }}</td>
<td>{{ member.wardno }}</td>
</tr>
{% endfor %}
</tbody>
</table>
```

```
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="notification-box">
<div class="msg-sidebar notifications msg-noti">
<div class="topnav-dropdown-header">
<span>Messages</span>
</div>
</ul>
</div>
</div>
</div>
</div>
</div>
<div class="sidebar-overlay" data-reff=""></div>
<script src="{% static 'assets/js/jquery-3.2.1.min.js' %}"></script>
<script src="{% static 'assets/js/popper.min.js' %}"></script>
<script src="{% static 'assets/js/bootstrap.min.js' %}"></script>
<script src="{% static 'assets/js/jquery.slimscroll.js' %}"></script>
<script src="{% static 'assets/js/Chart.bundle.js' %}"></script>
<script src="{% static 'assets/js/chart.js' %}"></script>
<script src="{% static 'assets/js/app.js' %}"></script>
</body>
</html>
```

Member_Approval

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<!-- departments23:21-->
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0">
{% comment %} <link rel="shortcut icon" type="image/x-icon" href="{% static
'assets/img/favicon.ico' %}"> {% endcomment %}
<title>CropElevate</title>
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/bootstrap.min.css' %}">
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/dataTables.bootstrap4.min.css'
%}">
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/font-awesome.min.css' %}">
<link rel="stylesheet" type="text/css" href="{% static 'assets/css/style.css' %}">
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.0/css/all.min.css">
<style>
body {
font-family: Arial, sans-serif;
```

```
background-color: #b3f59d;
{% comment %} background-image: url('{% static "img/img2.jfif" %}'); {% endcomment %}
{% comment %} background-size: cover;
background-repeat: no-repeat;
background-attachment: fixed; {% endcomment %}
margin: 0;
padding: 0;
}
.header
{
background-color:#88a60f;
}
.header .header-left
{
background-color:#88a60f;
{% comment %} background-image:linear-gradient(rgb(153, 245, 93), rgb(241, 243, 145)); {%
endcomment %}
}
.sidebar-menu li a {
color:black;
{% comment %} color:white; {% endcomment %}
}
.sidebar-menu li a:hover {
color:#88a60f;
{% comment %} color: white; {% endcomment %}
}
.sidebar-menu li.active a {
color:#88a60f;
background-color: #f3f3f3;
}
{% comment %} body {
background-image:linear-gradient(rgb(153, 245, 93), rgb(241, 243, 145));
font-family: Arial, sans-serif;
background-color: #f2f2f2;
margin: 0;
padding: 0;
} {% endcomment %}
.container {
max-width: 800px;
margin: 0 auto;
background-color: #fff;
padding: 20px;
box-shadow: 0 0 10px rgba(0, 0, 0, 1.9);
border-radius: 5px;
}
h1 {
text-align: center;
color: #333;
}
.details {
margin-top: 20px;
```

```
}
table {
width: 100%;
border-collapse: collapse;
margin-top: 20px;
}
table, th, td {
border: 1px solid grey;
}
th, td {
padding: 10px;
text-align: left;
}
th {
background-color: lightgreen;
}
.no-details {
text-align: center;
color: #777;
}
.text-right {
text-align: right;
}
.btn-primary {
background-color: #007BFF;
color: #fff;
border: none;
padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
}
.btn-primary:hover {
background-color: #0056b3;
}
.apply-button {
background-color: #4CAF50;
color: white;
border: none;
padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
margin-top: 20px;
}
```

```
</style>
<!--[if lt IE 9]>
<script src="assets/js/html5shiv.min.js"></script>
<script src="assets/js/respond.min.js"></script>
<![endif]-->
</head>
<body>
<div class="main-wrapper">
<div class="header">
<div class="header-left">
<a href="{% url 'mindex' %}" class="logo">
<img src="{% static 'assets/img/adminlogo1.png' %}" width="35" height="35" alt="">
<span>CropElevate</span>
</a>
</div>
<a id="toggle_btn" href="javascript:void(0);"><i class="fa fa-bars"></i></a>
<a id="mobile_btn" class="mobile_btn float-left" href="#sidebar"><i class="fa fa-bars"></i></a>
<ul class="nav user-menu float-right">
<li class="nav-item dropdown d-none d-sm-block">
<a href="#" class="dropdown-toggle nav-link" data-toggle="dropdown"><i class="fa fa-bell-
o"></i> <span class="badge badge-pill bg-danger float-right"></span></a>
<div class="dropdown-menu notifications">
<div class="topnav-dropdown-header">
<span>Notifications</span>
</div>
<div class="drop-scroll">
<ul class="notification-list">
</ul>
</div>
</div>
</li>
<li class="nav-item dropdown d-none d-sm-block">
<a href="javascript:void(0);" id="open_msg_box" class="hasnotifications nav-link"><i class="fa
fa-comment-o"></i> <span class="badge badge-pill bg-danger float-right"></span></a>
</li>
<li class="nav-item dropdown has-arrow">
<a href="#" class="dropdown-toggle nav-link user-link" data-toggle="dropdown">
<span class="user-img">
<span>{{ user.name }}</span>
<img class="rounded-circle" src="{% static 'assets/img/user.jpg' %}" width="40" alt="Admin">
<span class="status online"></span></span>
</a>
<div class="dropdown-menu">
<a class="dropdown-item" href="{% url 'meditprofile' %}">Edit Profile</a>
<a class="dropdown-item" href="{% url "loginn" %}">Logout</a>
</div>
</li>
</ul>
<div class="dropdown mobile-user-menu float-right">
<a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown" aria-
expanded="false"><i class="fa fa-ellipsis-v"></i></a>
```

```
<div class="dropdown-menu dropdown-menu-right">
<a class="dropdown-item" href="{% url 'meditprofile' %}">Edit Profile</a>
<a class="dropdown-item" href="{% url "loginn" %}">Logout</a>
</div>
</div>
</div>
<div class="sidebar" id="sidebar">
<div class="sidebar-inner slimscroll">
<div id="sidebar-menu" class="sidebar-menu">
<ul>
<li class="menu-title" style="color:black; font-weight:bold;">Member</li>
<li>
<a href="{% url 'mindex' %}"><i class="fa fa-dashboard"></i> <span>Dashboard</span></a>
</li>
<li class="active">
<a href="{% url 'mfregistered' %}"><i class="fas fa-user fa-2x"></i><span>Registered
Farmers</span></a>
</li>
<li>
<a href="{% url 'mcrop' %}"><i class="fas fa-seedling fa-2x"></i> <span>Crops</span></a>
</li>
<li>
<a href="{% url 'mapprove' %}"><i class="fas fa-check-circle"></i>  <span>Approved
List</span></a>
</li>
<li>
<a href="{% url 'mpending' %}"><i class="fas fa-hourglass-half"></i> <span>Pending
List</span></a>
</li
<li>
<a href="{% url 'mmeeting' %}"><i class="fas fa-users fa-2x"></i> <span>Meetings</span></a>
</li>
<li>
<a href="{% url 'mcalendar' %}"><i class="fa fa-calendar"></i> <span>Calendar</span></a>
</li>
</ul>
</div>
</div>
</div>
<div class="page-wrapper">
<div class="content">
<div class="row">
<div class="col-sm-5 col-5">
<h4 class="page-title"></h4>
</div>
</div>
</div>
<div class="row">
<div class="col-md-6">
<div class="table-responsive">
<table class="table table-striped custom-table">
```

```
</table>
</div>
</div>
</div>
<div class="container">
<div class="container">
<h1 style="color:black; font-weight:bold;">Registered Farmers</h1>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<div class="details">
<div id="indiv">
<label for="role-filter" class="filter-label">Filter by Status:</label>
<select id="role-filter" class="filter-select">
<option value="all">All</option>
<option value="approved">Approved</option>
<option value="rejected">Rejected</option>
<option value="pending">Pending</option>
<option value="waiting">Waiting</option>
</select>
<br><br>
<form class="search-form" id="searchform">
<label for="username" class="search-label">Search by Name:</label>
<input type="text" id="username" name="username" class="search-input">
<button type="button" id="search-button" class="search-button">Search</button>
</form>
<br><br>
</div>
<p id="no-user-message" style="display: none;">No such Farmer</p>
{% if pending_details %}
<table class="user-list">
<tr>
<th style="color:black;"><b>Crop Name</b></th>
<th style="color:black;"><b>Farmer Name</b></th>
<th style="color:black;"><b>Address</b></th>
<th style="color:black;"><b>Contact No</b></th>
<th style="color:black;"><b>Ward No</b></th>
<th style="color:black;"><b>Annual Income </b></th>
<th style="color:black;"><b>Status</b></th>
<th style="color:black;"><b>Approve</b></th>
<th style="color:black;"><b>File</b></th>
</tr>
{% for application in pending_details %}
<tr>
<td style="color:black;">{{ application.cname }}</td>
<td style="color:black;">{{ application.farmerName }}</td>
<td style="color:black;">{{ application.address }}</td>
<td style="color:black;">{{ application.contactNo }}</td>
<td style="color:black;">{{ application.wardNo }}</td>
<td style="color:black;">{{ application.AnnualIncome }}</td>
<td>
{% if application.is_approved == 'pending' %}
<span class="text-success" style="font-weight: bolder; font-size: large;"></span>
```

```
{% elif application.is_approved == 'approved' %}
<span class="text-success" style="font-weight: bolder; font-size: large;">Approved</span>
{% elif application.is_approved == 'rejected' %}
<span class="text-danger" style="font-weight: bolder; font-size: large;">Rejected</span>
{% elif application.is_approved == 'waiting' %}
<span class="text-waiting" style="font-weight: bolder; font-size: large;
color:#ccb023;">Waiting</span>
{% else %}
<span class="text-danger" style="font-weight: bolder; font-size: large;"></span>
{% endif %}
</td>
<td>
{% with annual_income=application.AnnualIncome|default_if_none:"0" %}
{% if annual_income|add:0 <= 30000 %}
{% if application.is_approved == 'pending' %}
<form method="post" action="{% url 'approve_certification' application.id %}">
{% csrf_token %}
<button class="btn btn-success approve-btn" type="submit">Approve</button>
</form>
<form method="post" action="{% url 'reject_certification' application.id %}">
{% csrf_token %}
<button class="btn btn-danger reject-btn" type="submit">Reject</button>
</form>
{% elif application.is_approved == 'approved' %}
<form method="post" action="{% url 'reject_certification' application.id %}">
{% csrf_token %}
<button class="btn btn-danger reject-btn" type="submit">Reject</button>
</form>
{% elif application.is_approved == 'rejected' %}
<form method="post" action="{% url 'approve_certification' application.id %}">
{% csrf_token %}
<button class="btn btn-success approve-btn" type="submit">Approve</button>
</form>
{% else %}
<p>Not working</p>
{% endif %}
{% else %}
<span class="text-danger">Income too high</span>
{% if application.is_approved == 'pending' %}
<form method="post" action="{% url 'approve_certification' application.id %}">
{% csrf_token %}
<button class="btn btn-success approve-btn" type="submit">Approve</button>
</form>
<form method="post" action="{% url 'set_status_waiting' application.id %}">
{% csrf_token %}
<button class="btn btn-warning waiting-btn" type="submit">Waiting</button>
</form>
{% elif application.is_approved == 'approved' %}
<form method="post" action="{% url 'set_status_waiting' application.id %}">
{% csrf_token %}
<button class="btn btn-warning waiting-btn" type="submit">Waiting</button>
```

```
</form>
{% elif application.is_approved == 'waiting' %}
<form method="post" action="{% url 'approve_certification' application.id %}">
{% csrf_token %}
<button class="btn btn-success approve-btn" type="submit">Approve</button>
</form>
{% endif %}
{% endif %}
{% endwith %}
</td>
<td>
{% if application.file_upload %}
<a href="{{ application.file_upload.url }}" target="_blank">Download File</a>
{% else %}
No File
{% endif %}
</td>
</tr>
{% endfor %}
</table>
{% else %}
<p>No details available.</p>
{% endif %}
</div>
</div>
<script>
$(document).ready(function () {
const input = $('#username');
const roleFilter = $('#role-filter');
const rows = $('.user-list tbody tr');
const noUserMessage = $('#no-user-message');
$('#search-button').on('click', filterUsers);
roleFilter.on('change', filterUsers);
function filterUsers() {
const searchTerm = input.val().toLowerCase();
const selectedRole = roleFilter.val();
let found = false; // Flag to track if a matching user is found
rows.each(function () {
const username = $(this).find('td:nth-child(2)').text().toLowerCase();
const role = $(this).find('td:nth-child(7)').text().toLowerCase();
// Check if the user matches the selected role (or if all roles are selected)
if ((selectedRole === 'all' || role.includes(selectedRole)) &&
(searchTerm.length === 0 || username.includes(searchTerm))) {
$(this).css('display', 'table-row');
found = true; // Matching user found
} else {
$(this).css('display', 'none');
}
});
// Show/hide "No such user" message
if (found) {
```

```
noUserMessage.css('display', 'none');
} else {
noUserMessage.css('display', 'block');
}
}
});
</script>
</table>
</div>
</div>
</div>
</div>
<div class="notification-box">
<div class="msg-sidebar notifications msg-noti">
<div class="topnav-dropdown-header">
<span>Messages</span>
</div>
</div>
</div>
</div>
</div>
<script>
document.querySelectorAll(".apply-button").forEach(button => {
button.addEventListener("click", function() {
const rowId = this.closest("tr").getAttribute("data-row-id");
const isApproved = true;  // Modify this condition as needed
if (isApproved) {
const rowHtml = this.closest("tr").outerHTML;
const mapproveTable = document.getElementById("mapprove-table");
mapproveTable.innerHTML += rowHtml;
}
this.closest("tr").remove();
});
});
</script>
<div class="sidebar-overlay" data-reff=""></div>
<script src="{% static 'assets/js/jquery-3.2.1.min.js' %}"></script>
<script src="{% static 'assets/js/popper.min.js' %}"></script>
<script src="{% static 'assets/js/bootstrap.min.js' %}"></script>
<script src="{% static 'assets/js/jquery.dataTables.min.js' %}"></script>
<script src="{% static 'assets/js/dataTables.bootstrap4.min.js' %}"></script>
<script src="{% static 'assets/js/jquery.slimscroll.js' %}"></script>
<script src="{% static 'assets/js/app.js' %}"></script>
</body>
</html>
```

Apply_Crop

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Apply for Crop</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<style>
body {
font-family: Arial, sans-serif;
background-color: #f2f2f2;
margin: 0;
padding: 0;
}
.container {
max-width: 800px;
margin: 0 auto;
background-color: #fff;
padding: 20px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.6);
border-radius: 5px;
}
h1 {
text-align: center;
color: #333;
}
form {
margin-top: 20px;
}
.form-group {
margin-bottom: 15px;
}
label {
font-weight: bold;
}
input[type="text"], input[type="tel"] {
width: 100%;
padding: 10px;
border: 1px solid #ddd;
border-radius: 5px;
}
select {
width: 100%;
padding: 10px;
border: 1px solid #ddd;
border-radius: 5px;
}
.apply-button {
```

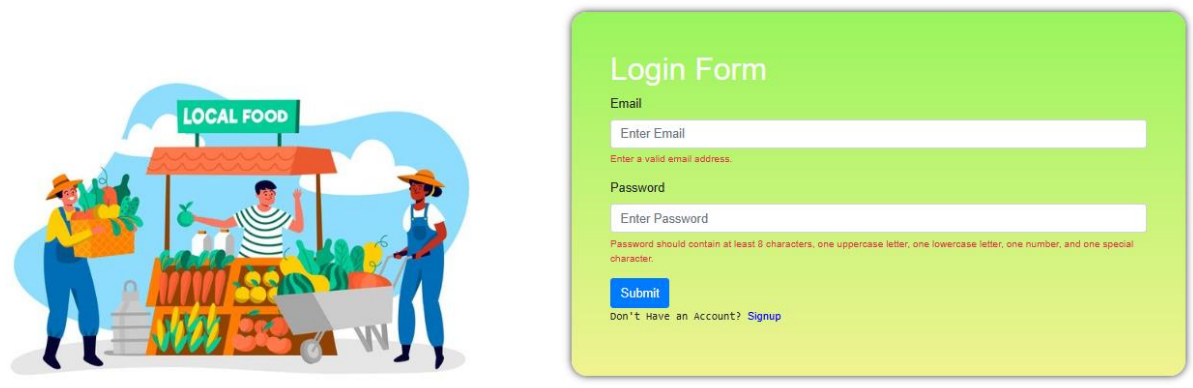
```
background-color: #4CAF50;
color: white;
border: none;
padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
margin-top: 20px;
}
.back-button {
background-color: lightblue;
color: white;
border: none;
padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
margin-top: 20px;
}
</style>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-success">
<h2 style="color:#0d4859; padding-left:20px; font-weight:bold;">CropElevate</h2>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
<ul class="navbar-nav">
<a href="{% url 'homepage' %}" class="nav-item nav-link active" style="margin-left: 80px;
color:white; font-size:20px;">Home</a>
<a href="{% url 'crop' %}" class="nav-item nav-link active" style="margin-left: 60px;
color:white; font-size:20px;">Notice</a>
<a href="{% url 'ceditprofile' %}" class="nav-item nav-link active" style="margin-left: 80px;
color:white; font-size:20px;">Edit Profile</a>
<a href="{% url 'fmyprofile' %}" class="nav-item nav-link active" style="margin-left: 80px;
color:white; font-size:20px;">My Profile</a>
</ul>
</div>
<div class="dropdown"style="float: left; margin-left:300px; margin-top:20px;" >
<button class="btn btn-secondary dropdown-toggle" type="button" id="userDropdown" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false" style="background-color:
#c1d18a; color:black; font-weight:bold;">
<i class="fa fa-user" aria-hidden="true"></i>
```

```
<span>{{ user.name }}</span>
</button>
<div class="dropdown-menu" aria-labelledby="userDropdown" style="background-color:
#c1d18a;">
{% if user.is_authenticated %}
<a class="dropdown-item" href="{% url 'loggout' %}">Logout</a>
{% else %}
<a class="dropdown-item" href="{% url 'loginn' %}">Login</a>
{% endif %}
</div>
</div>
</nav>
<br><br>
<div class="container">
{% if existing_certification %}
{% if existing_certification.is_approved == 'pending' %}
<p class="certification-message">You Have Applied For The crop.</p>
{% elif existing_certification.is_approved == 'approved' %}
<p class="certification-message">Your Application Is Approved By The Member.</p>
{% elif existing_certification.is_approved == 'rejected' %}
<p class="certification-message">Your Application is Rejected.</p>
{% elif existing_certification.is_approved == 'waiting' %}
<p class="certification-message">You Are In The Waiting List.</p>
{% else %}
<p> Not working </p>
{% endif %}
{% if existing_certification.is_approvedd == 'approved' %}
<p class="certification-message">Your Application Is Approved By The Panchayath
Secretary.</p>
{% elif existing_certification.is_approved == 'waiting' %}
<p class="certification-message">You Are in the Waiting list by Panchayath Secretarys.</p>
{% else %}
<p> Please Wait </p>
{% endif %}
{% if existing_certification.is_given == 'given' %}
<p class="certification-message">You Can Collect The Crop From The Panchayath</p>
{% elif existing_certification.is_given == 'notgiven' %}
<p class="certification-message">Crop Samplings Is Not Allocated to you at this time</p>
{% else %}
<p> Please Wait </p>
{% endif %}
{% else %}
<h1>Apply for Crop</h1>
<form method="POST" action="" autocomplete="off" id="apply"  enctype="multipart/form-
data">
{% csrf_token %}
<div class="col-md-6">
<label for="cropName" style="color:black;">Crop Name</label>
<input type="text" id="cname" name="cname" value="{{ crop_name }}" readonly>
</div>
<div class="col-md-6">
```

```
<label for="farmerName" style="color:black;">Farmer Name</label>
<input type="text" id="farmerName" name="farmerName" placeholder="Enter Name"  value="{{
farmer_name }} {{ farmer_lname }}" required readonly>
<small id="nameError" class="form-text text-danger"></small>
</div>
<div class="col-md-6">
<label for="address" style="color:black;">Address</label>
<input type="text" id="address" name="address" placeholder="Enter Address" value="{{
farmer_address }}" required readonly>
<small id="addressError" class="form-text text-danger"></small>
</div>
<div class="col-md-6">
<label for="wardNo" style="color:black;">Ward </label>
<div class="form-group form-focus">
<input type="text" id="wardno" name="wardno" placeholder="Enter Ward Number" value="{{
farmer_ward }}" required readonly>
{% comment %} <input type="text" id= "ward"class="form-control floating" value="{{
farmer_ward }}" name="ward" required readonly> {% endcomment %}
</div>
</div>
<div class="col-md-6">
<label for="phone" style="color:black;">Phone </label>
<div class="form-group form-focus">
<input type="text" id="phone_number" name="phone_number" placeholder="Enter Phone
Number" value="{{ farmer_phone_number }}" required readonly>
</div>
</div>
<div class="col-md-6">
<label for="annualIncome" style="color:black;">Annual Income </label>
<input type="text" id="annualIncome" name="annualIncome" placeholder="Enter Annual
Income" value="{{ farmer_annual_income }}" required readonly>
<small id="annualIncomeError" class="form-text text-danger"></small>
</div>
<div class="col-md-6">
<label for="land" style="color:black;">Amount of Land  (in cent )</label>
<div class="form-group form-focus">
<input type="text" class="form-control floating" value="{{ farmer_land }}" name="land"
id="land" style="border-radius:10px;" readonly>
<small id="landError" class="form-text text-danger"></small>
</div>
</div>
<div class="col-md-6">
<label for="file_upload" style="color:black;">Upload File</label>
<input type="file" name="file_upload" id="file_upload" value="{{ farmer_file_upload }}">
</div>
<button type="submit" class="apply-button">Apply</button>
</form>
{% endif %}
</div>
<script>
const farmerNameInput = document.getElementById('farmerName');
const addressInput = document.getElementById('address');
const contactNoInput = document.getElementById('contactNo');
const wardNoInput = document.getElementById('wardNo');
const annualIncomeInput = document.getElementById('annualIncome');
const nameError = document.getElementById('nameError');
const addressError = document.getElementById('addressError');
const contactNoError = document.getElementById('contactNoError');
const wardNoError = document.getElementById('wardNoError');
const annualIncomeError = document.getElementById('annualIncomeError');
function validateName() {
const nameValue = farmerNameInput.value.trim();
if (!nameValue) {
nameError.textContent = 'Name is required.';
nameError.style.color = 'red';
return false;
} else if (!/^[a-zA-Z]+$/.test(nameValue)) {
nameError.textContent = 'Name should contain only alphabetical characters.';
nameError.style.color = 'red';
return false;
} else {
nameError.textContent = ''; // Clear the error message
return true;
}
}
function validateAddress() {
const addressValue = addressInput.value.trim();
if (!addressValue) {
addressError.textContent = 'Address is required.';
addressError.style.color = 'red';
return false;
} else if (!/^[a-zA-Z0-9\s]+$/.test(addressValue)) {
addressError.textContent = 'Address should contain only alphabets, numbers, and spaces.';
addressError.style.color = 'red';
return false;
} else {
addressError.textContent = ''; // Clear the error message
return true;
}
}
function validateContactNo() {
const contactNoValue = contactNoInput.value.trim();
if (!contactNoValue) {
contactNoError.textContent = 'Contact No is required.';
contactNoError.style.color = 'red';
return false;
} else if (!/^\d{10}$/.test(contactNoValue)) {
contactNoError.textContent = 'Contact No should be a 10-digit number.';
contactNoError.style.color = 'red';
return false;
} else {
contactNoError.textContent = ''; // Clear the error message
return true;
}}
function validateWardNo() {
const wardNoValue = wardNoInput.value.trim();
if (!wardNoValue) {
wardNoError.textContent = 'Ward No is required.';
wardNoError.style.color = 'red';
return false;
} else if (!/^[a-zA-Z0-9]+$/.test(wardNoValue)) {
wardNoError.textContent = 'Ward No should contain only alphabets and numbers.';
wardNoError.style.color = 'red';
return false;
} else {
wardNoError.textContent = ''; // Clear the error message
return true;
}}
function validateAnnualIncome() {
const annualIncomeValue = annualIncomeInput.value.trim();
if (!annualIncomeValue) {
annualIncomeError.textContent = 'Annual Income is required.';
annualIncomeError.style.color = 'red';
return false;
} else if (!/^\d+$/.test(annualIncomeValue)) {
annualIncomeError.textContent = 'Annual Income should contain only numbers.';
annualIncomeError.style.color = 'red';
return false;
} else {
annualIncomeError.textContent = ''; // Clear the error message
return true;
}}
farmerNameInput.addEventListener('input', validateName);
addressInput.addEventListener('input', validateAddress);
contactNoInput.addEventListener('input', validateContactNo);
wardNoInput.addEventListener('input', validateWardNo);
annualIncomeInput.addEventListener('input', validateAnnualIncome);
document.getElementById('apply').addEventListener('submit', function(event) {
if (!validateName() || !validateAddress() || !validateContactNo() || !validateWardNo() ||
!validateAnnualIncome()) {
event.preventDefault();
}
});
</script>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

9.2 Screen Shots

Login



Login Form

Email

Enter Email

Enter a valid email address.

Password

Enter Password

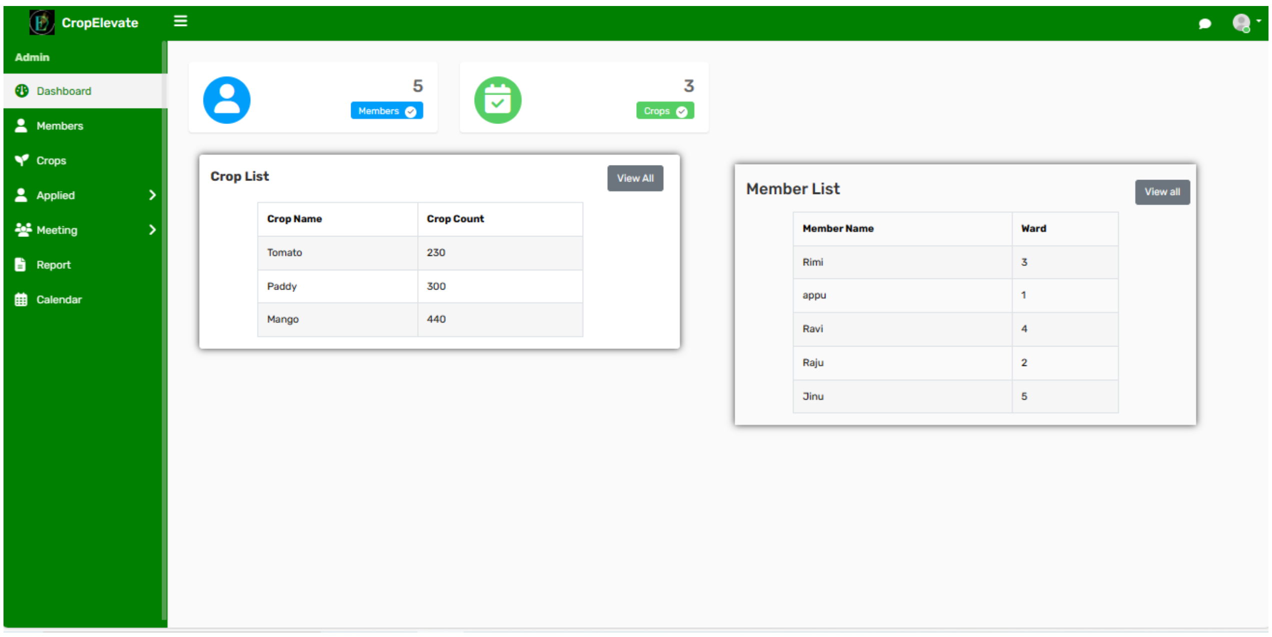
Password should contain at least 8 characters, one uppercase letter, one lowercase letter, one number, and one special character.

Submit

Don't Have an Account? Signup

Fig:18 Login

Admin_Dashboard



Crop List

View All

Crop Name	Crop Count
Tomato	230
Paddy	300
Mango	440

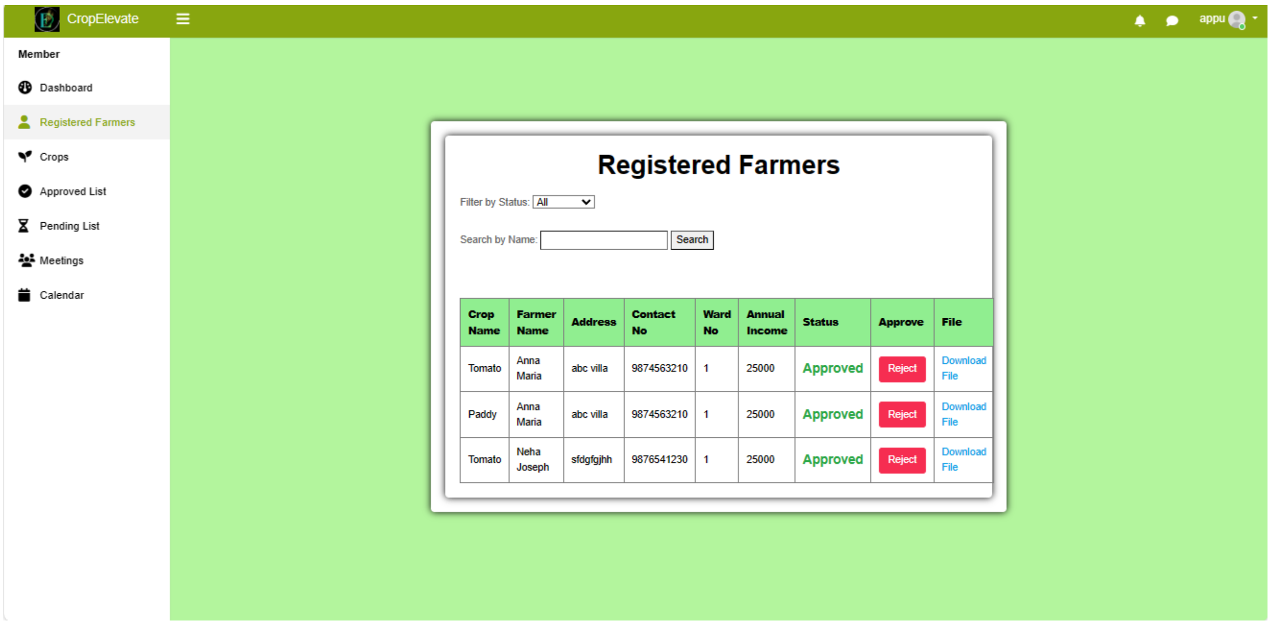
Member List

View all

Member Name	Ward
Rimi	3
appu	1
Ravi	4
Raju	2
Jinu	5

Fig:19 Admin_Dashboard

Member_Approval



Registered Farmers

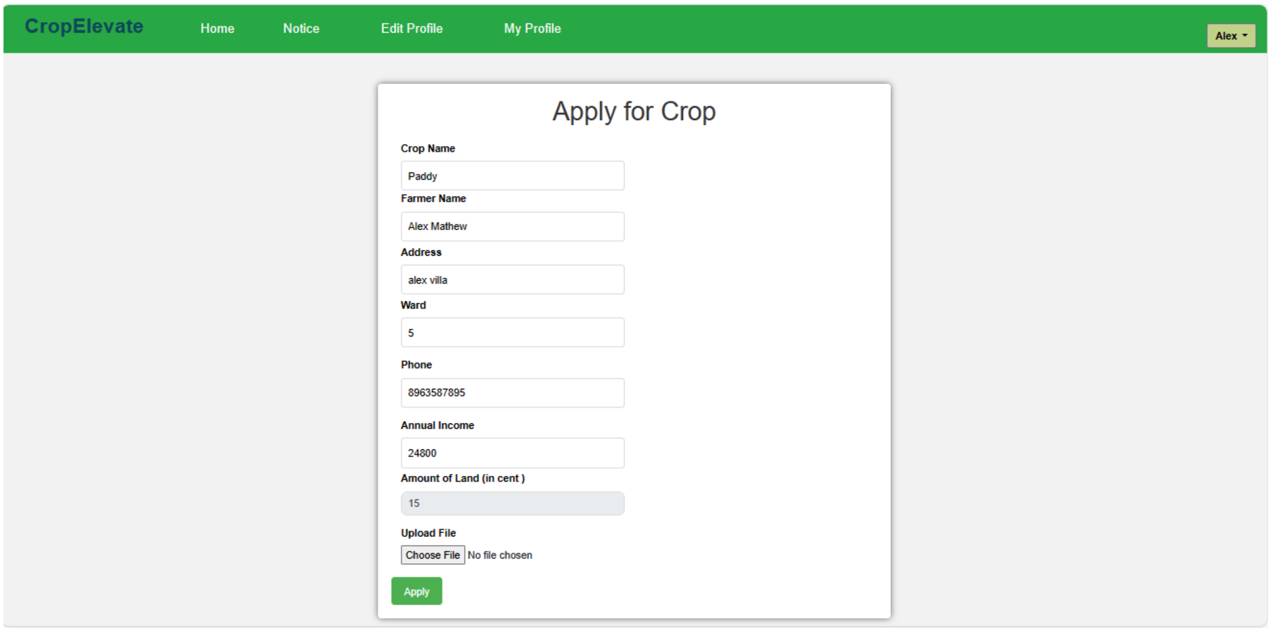
Filter by Status: All

Search by Name: Search

Crop Name	Farmer Name	Address	Contact No	Ward No	Annual Income	Status	Approve	File
Tomato	Anna Maria	abc villa	9874563210	1	25000	Approved	Reject	Download File
Paddy	Anna Maria	abc villa	9874563210	1	25000	Approved	Reject	Download File
Tomato	Neha Joseph	sfdgfgjhh	9876541230	1	25000	Approved	Reject	Download File

Fig:20 Member_Approval

Apply_Crop



Apply for Crop

Crop Name	Paddy
Farmer Name	Alex Mathew
Address	alex villa
Ward	5
Phone	8963587895
Annual Income	24800
Amount of Land (in cent)	15
Upload File	Choose File No file chosen

Apply

Fig:21 Apply_Crop