KRISHI BHAVAN MANAGEMENT SYSTEM

Project Report Submitted By

LISA ELISABATH JOJI

Reg. No:AJC17MCA-I032

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]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, “**KRISHI BHAVAN MANAGEMENT SYSTEM”** is the bonafide work of **LISA ELISABATH JOJI (Reg.No:AJC17MCA-I032)** 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 2017-22.

Ms. Grace Joseph
Internal Guide

Ms. Meera Rose Mathew
Coordinator

Rev.Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **"KRISHI BHAVAN MANAGEMENT SYSTEM"** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

Date:	**LISA ELISABATH JOJI**
KANJIRAPPALLY	**Reg. No: AJC17MCA-I032**

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 their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms. Grace Joseph** for her 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.

LISA ELISABATH JOJI

ABSTRACT

Krishi Bhavan Management System is a web application to manage various services required by the users as well as to manage those services by the officers. It provides services in a quick time according to the requirements that are to be fulfilled. This system provides consistency of data and develops the user friendly and interactive website which will reduce the paper works, faster and easy work and save the time. The user can also reduce the time and effort for filing an application by using this system. In this system there are mainly four users: User, Administrator, Officers and Assistant Officers. Admin is the one who manages all the accounts and this system. Admin can add the officers and view the users of the system. Users can apply for any of the services available within the system such as crop insurance, paddy royalty, crop base price allowance and also can apply for soil test. This system helps users to get agricultural allowances for their crops and helps agricultural officers as well as agricultural assistant officers complete their works in an easy manner.

CONTENT

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

"**KRISHI BHAVAN MANAGEMENT SYSTEM**" is a web application which is meant to help the users get their allowances or services in an easy manner. The user can also reduce thetime and effort by using this system. The proposed system includes four users they are administrator, user, officer and assistant officer. Registered users can login to the site and canapply for various services and can get the allowance. Administrator can view users, officers and assistant officers. He can approve or reject the officers as well as assistant officers. The administrator has the central control over the whole system. Administrator can accept or reject the services requested by the user.

1.2 PROJECT SPECIFICATION

The proposed system focuses on rectifying the problems within the existing system. This system aims in providing a more user-friendly website through which more number of common people can achieve their needs. It also focuses on providing successful completion of requirements within short period of time. It incorporates various functionalities such as crop insurance, paddy royalty, crop base price and soil test requests and so on which will in turn make the system more worth.

The system includes 4 modules. They are:

1. Admin

- Admin must have a login into this system. He has the overall control of the system.
- Admin can enable or disable the services.
- Admin can view all services.
- Admin can also add the employees.

2. User

- Users can register and login to the system and can apply for any of the services provided by the system.
- The user is provided with the following services – crop insurance, paddy royalty ,crop base price and soil test.

3. Agricultural Officer

- Agricultural officer can accept or reject the service request made by a user.
- The officer can manage the requests made by the user.

4. Agricultural Assistant Officer

- Agricultural assistant officer can accept or reject the service request made by a user.
- The officer can manage the requests made by the user.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. Customer can register and they can upload their projects. Each user can apply for various agricultural related requests. The proposed system rectify the drawbacks of the present system.

Krishi Bhavan Management System is envisaged to make available relevant information and services to the farming community and private sector through the use of information and communication technologies, to supplement the existing delivery channels provided for by the department.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system users can apply for various services available and the agricultural officer or the agricultural assistant officer can approve or reject the corresponding requests made by the user.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Project often delayed with no progress visibility.
- Maintenance of large paper works.
- Time required for completion of a service.

2.4 PROPOSED SYSTEM

The proposed system focuses on rectifying the problems within the existing system. This system aims in providing a more user-friendly website through which more number of common people can achieve their needs. It also focuses on providing successful completion of requirements within short period of time. It incorporates various functionalities such as soil test requests, paddy royalty farm plans and so on which will in turn make the system more worth.

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for business growth; on such consideration the system is proposed.

In our proposed system there is admin who can view all the officers and assistant officers. It allows users to apply for crop insurance, paddy royalty, and crop base price. Users of this proposed system are user, admin, officer and assistant officer. The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. Our website is multifunctional which includes user introduction, officer introduction, project details etc.

The proposed system helps the customer to get their requests done smoothly, efficiently and in less time and further the officer can approve or reject the requests. This system is made to help customers, officers as well as assistant officers.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- Integrated soil test

  The existing system lacks the functionality to submit for soil test within the system.
  By the integration of soil test to the proposed system the user will be able to request for soil test without the aid of third party websites and will be able to manage the request from the user dashboard.

- Provision to carry out payment

  By the addition of payment option to the user services can be processed in a faster phase. The existing system lacks the management of services fastly due to the lack of payment option.

- Centralized system access management

  Centralized user management is made possible by placing user registration openly and other modules within the admin. Thereby users, the external parties of the website have no access to internal administration of modules.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.
- The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, KRISHI BHAVAN MANAGEMENT SYSTEM was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project.The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible fordevelopment.

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

KRISHI BHAVAN MANAGEMENT SYSTEM, GUI is simple so that users can easily use it. KRISHI BHAVAN MANAGEMENT SYSTEM is simple enough so that no training is needed.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- Intel core i3
RAM	- 4 GB
Hard disk	- 1 TB

3.2.2 Software Specification

Front End	-	HTML,CSS, Bootstrap
Backend	-	MySQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, JQuery, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is a recursive acronym for "PHP: Hypertext Preprocessor". PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. Though its reputation is mixed, PHP is still extremely popular and is used in over 75% of all websites where the server-side programming language is known. PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website. It is not only used to build the web apps of many tech giants like Facebook but is also used to build many CMS (Content Management System) like WordPress, Drupal, Shopify, WooCommerce etc.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their

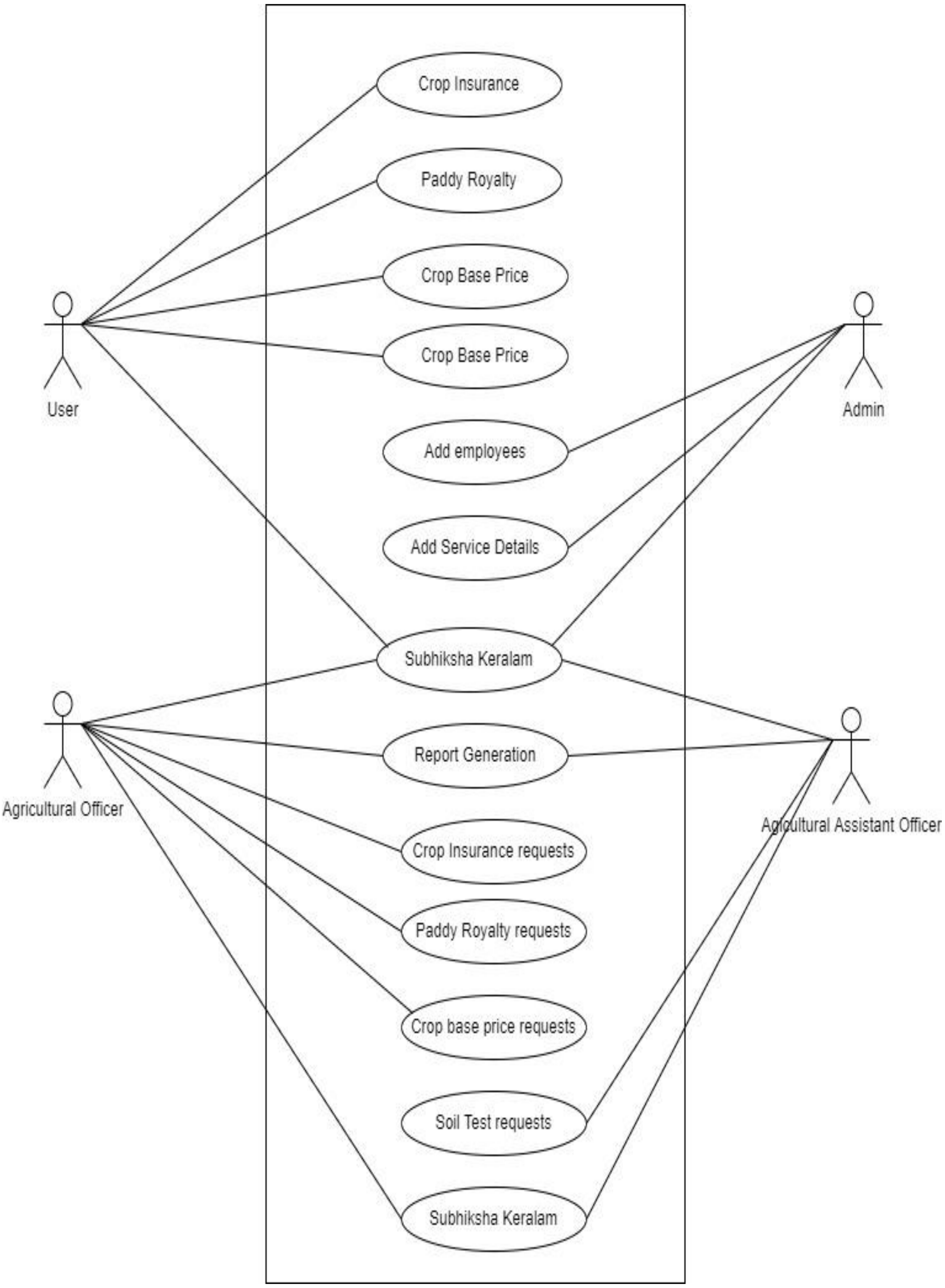
roles.

- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig 1: Use case diagram for Krishi Bhavan Management System



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message

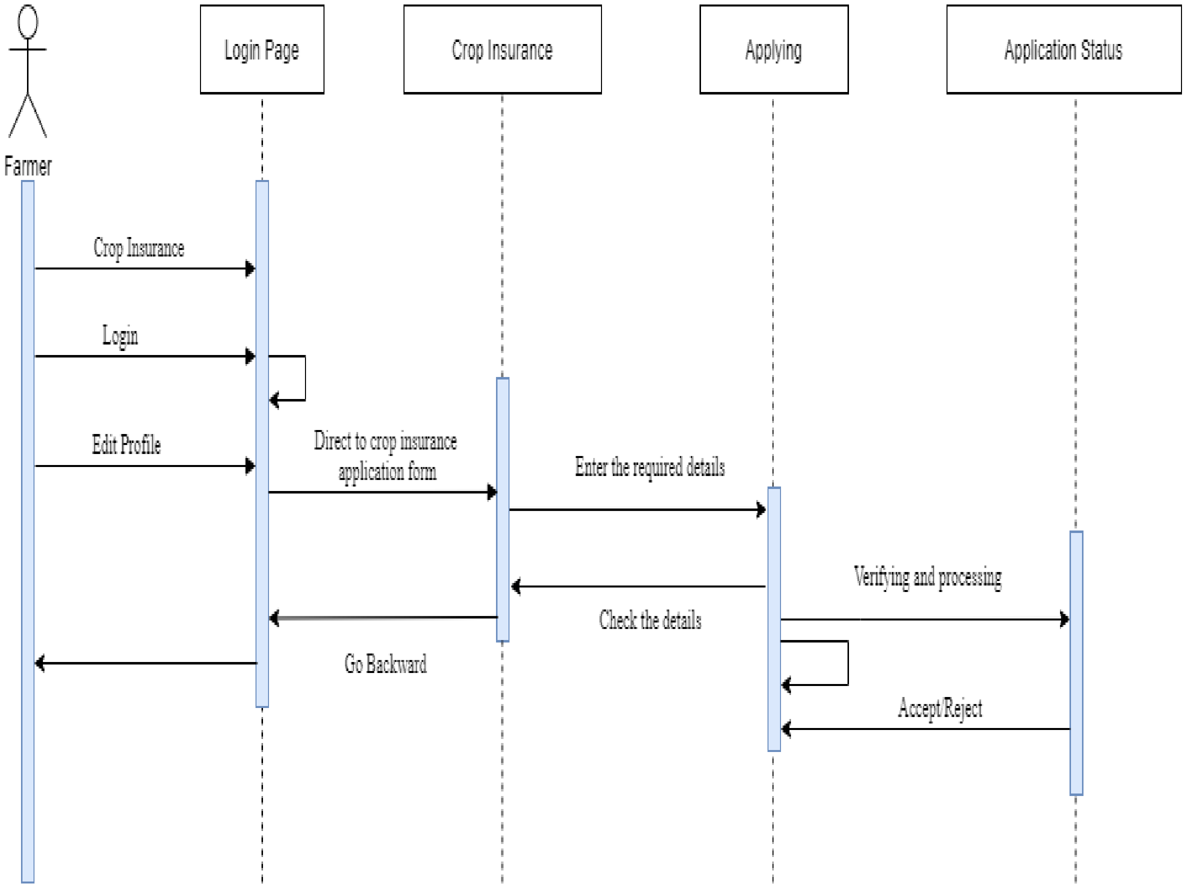
- Found Message

iv. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Fig 2: Sequence diagram for Krishi Bhavan Management System



4.2.3 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class Diagram Notations –

- **Class Name** - The name of the class appears in the first partition.
- **Class Attributes** - Attributes are shown in the second partition.

The attribute type is shown after the colon.

Attributes map onto member variables (data members) in code.

- **Class Operations (Methods) -**

  Operations are shown in the third partition. They are services the class provides.

  Operations map onto class methods in code.

- **Class Visibility**

  The +, - and # symbols before an attribute and operation name in a class denote the visibility

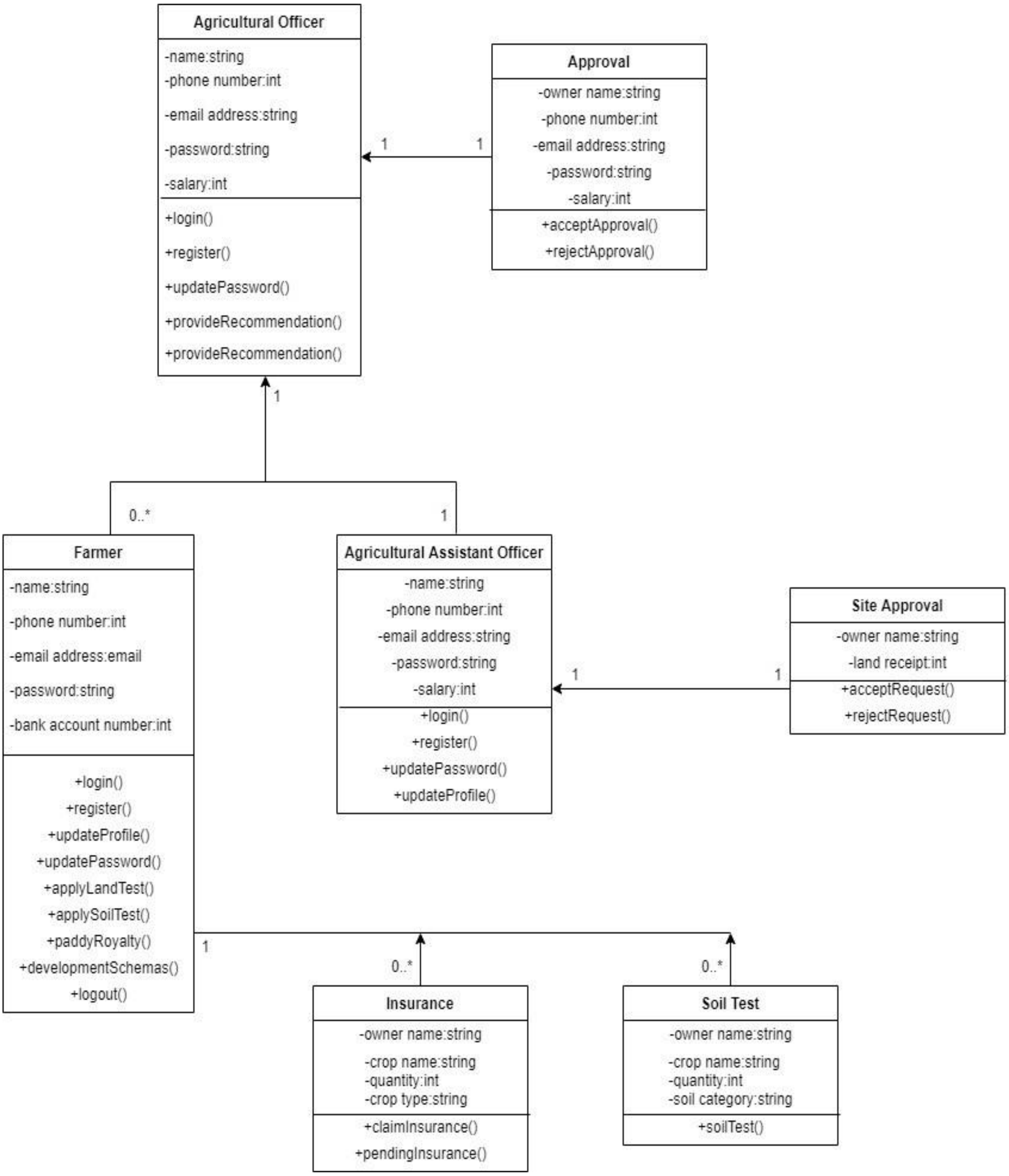
  of the attribute and operation.

  + denotes public attributes or operations

  - denotes private attributes or operations

  # denotes protected attributes or operations

Fig 3: Class diagram for Krishi Bhavan Management System



4.2.4 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

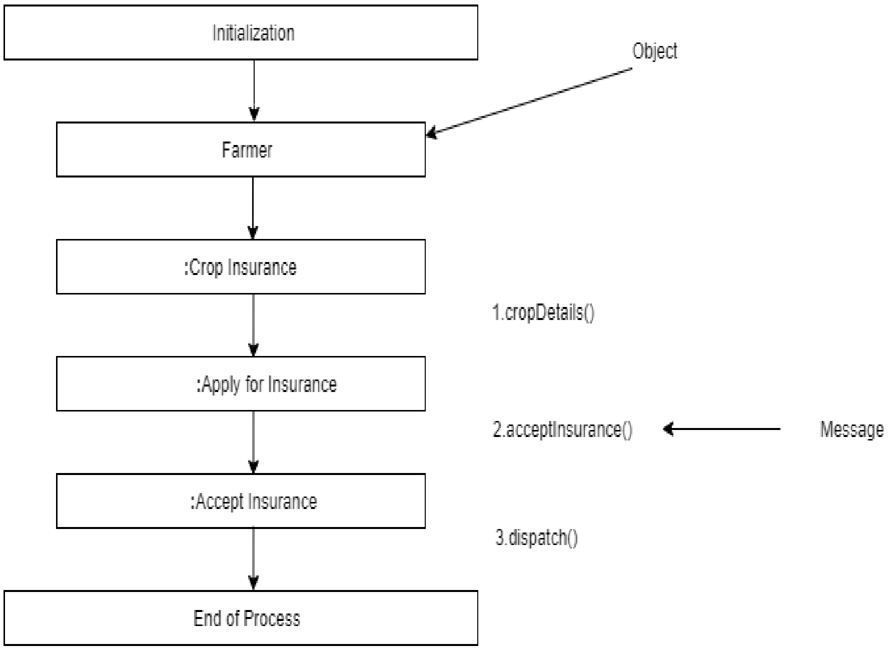
Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

Notations of a collaboration diagram

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. The four major components of a collaboration diagram are:

- Objects- Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name: class name. If an object has a property or state that specifically influences the collaboration, this should also be noted.
- Actors- Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.
- Links- Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.
- Messages- Messages between objects are shown as a labeled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.

Fig 4: Collaboration diagram for Krishi Bhavan Management System



4.2.5 ACTIVITY DIAGRAM

An activity diagram is a behavioral diagram i.e., it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. An activity diagram is used to model the workflow depicting conditions, constraints, sequential and concurrent activities. An activity diagram can be used to illustrate a business process (high level implementation) to a standalone algorithm.

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to represent business workflows.

The purpose of an activity diagram can be described as −

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swim lane, etc.

The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Before drawing an activity diagram, we should identify the following elements −

- Activities
- Association
- Conditions
- Constraints

Fig 5: Activity diagram of user for Krishi Bhavan Management System

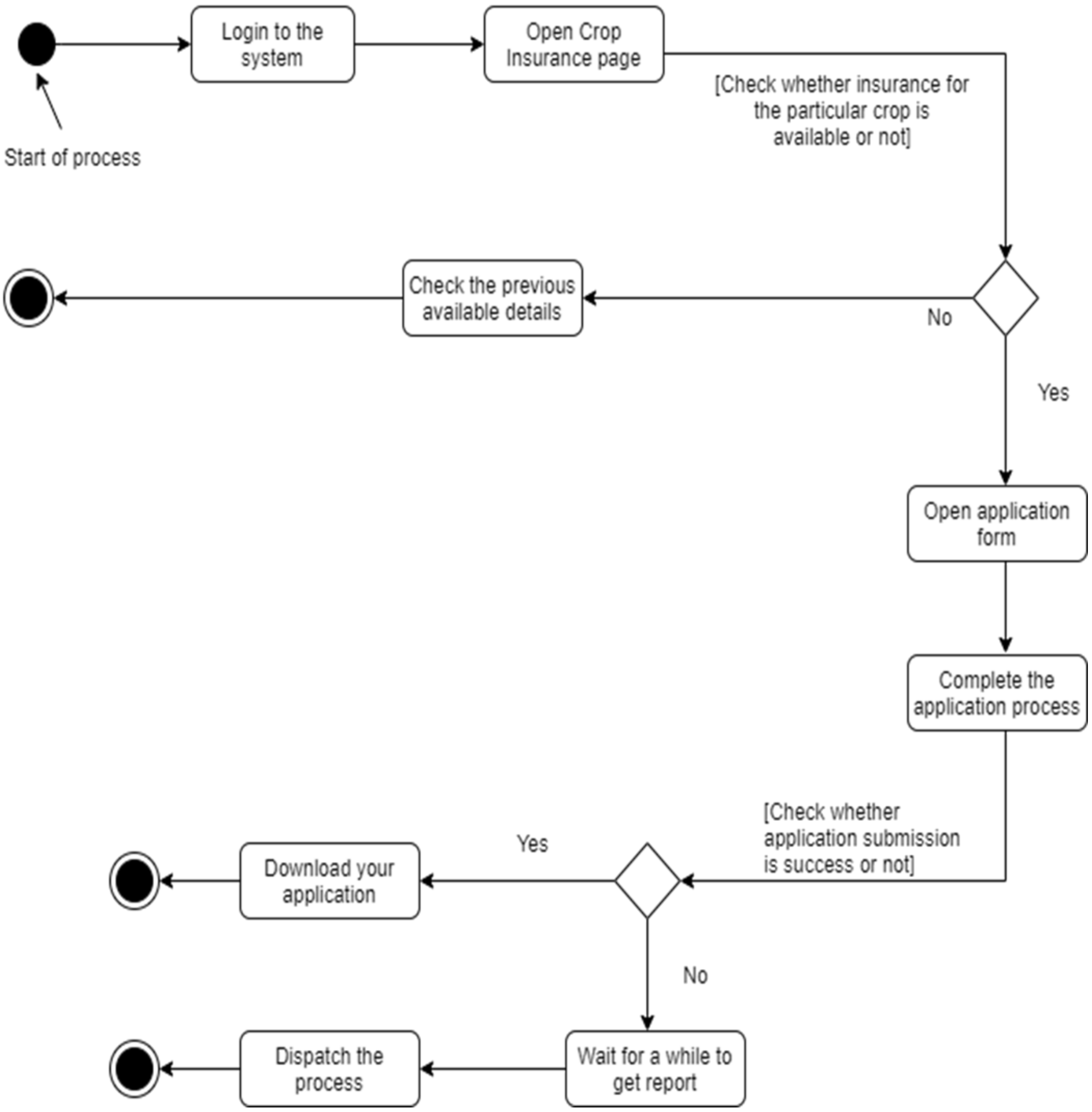

Login to the system
Open Crop Insurance page
Start of process
[Check whether insurance for the particular crop is available or not]
Check the previous available details
No
Yes
Open application form
Complete the application process
[Check whether application submission is success or not]
Yes
Download your application
No
Dispatch the process
Wait for a while to get report

Fig 6: Activity diagram of officer for Krishi Bhavan Management System

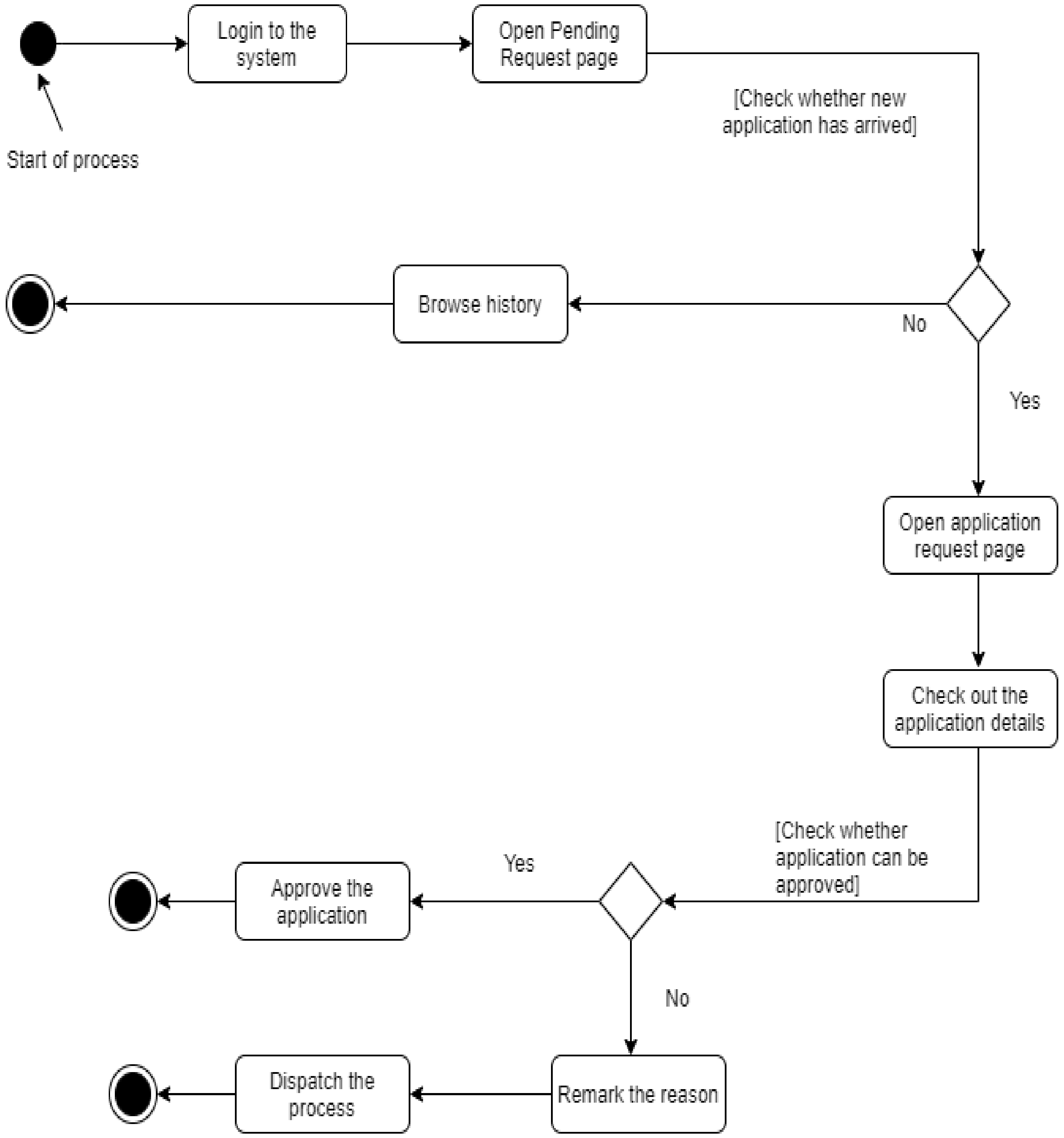
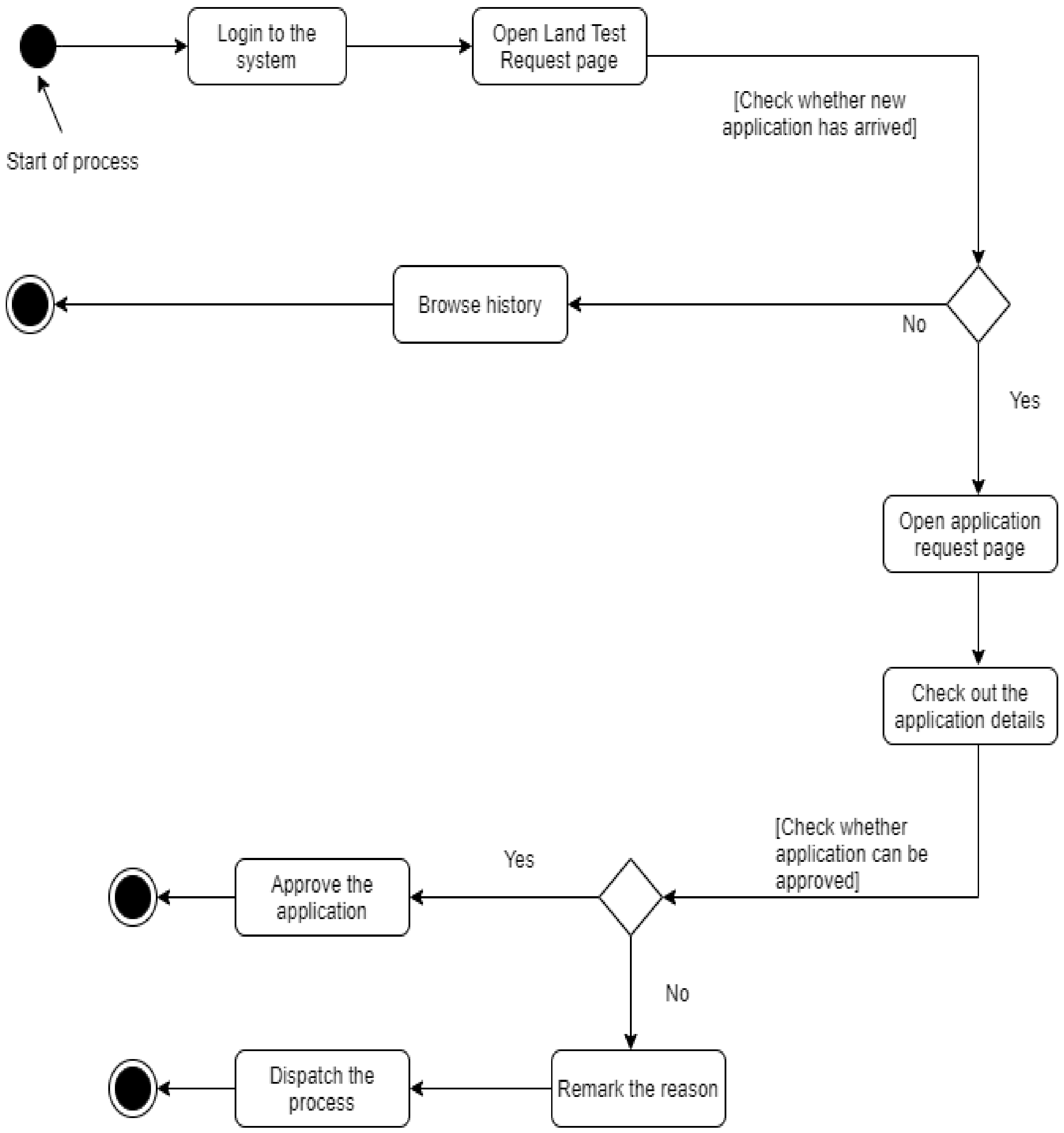

Login to the system
Open Pending Request page
[Check whether new application has arrived]
Start of process
Browse history
No
Yes
Open application request page
Check out the application details
[Check whether application can be approved]
Yes
Approve the application
No
Dispatch the process
Remark the reason

Fig 7: Activity diagram of agricultural assistant officer for Krishi Bhavan Management System


Login to the system
Open Land Test Request page
Start of process
[Check whether new application has arrived]
Browse history
No
Yes
Open application request page
Check out the application details
[Check whether application can be approved]
Yes
Approve the application
No
Dispatch the process
Remark the reason

4.2.6 STATE CHART DIAGRAM

State chart diagram describes different states of a component in a system. The states are specific to a component/object of a system.

A state chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

As State chart diagram defines the states, it is used to model the lifetime of an object.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

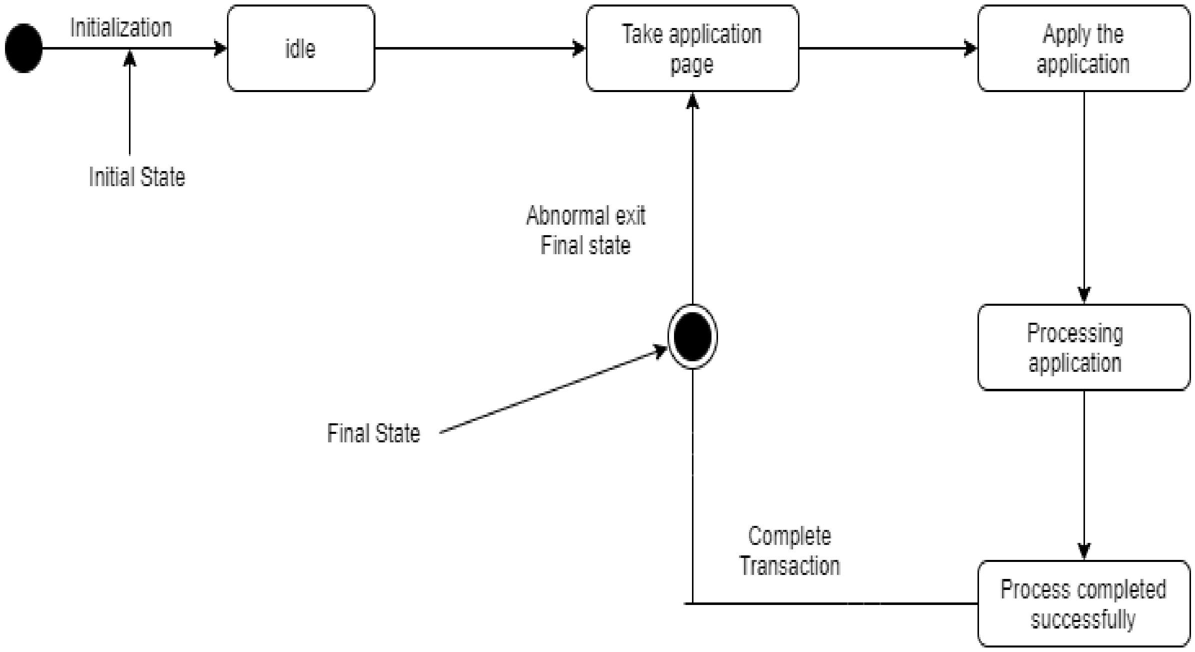
State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

How to Draw a State chart Diagram?

State chart diagram is used to describe the states of different objects in its life cycle. Emphasis is placed on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately.

State chart diagrams are very important for describing the states. States can be identified as the condition of objects when a particular event occurs.

Fig 8: State chart diagram for Krishi Bhavan Management System



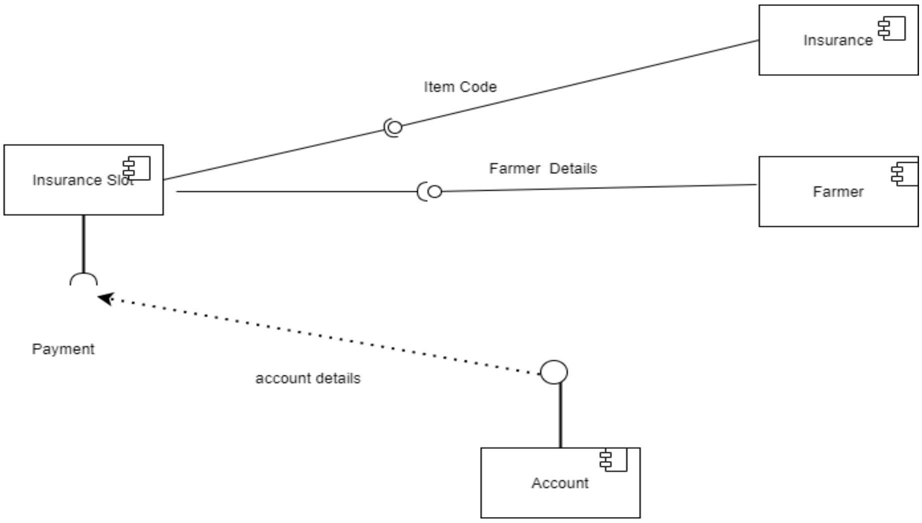
4.2.7 COMPONENT DIAGRAM

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces. The component diagrams have remarkable importance. It is used to depict the functionality and behavior of all the components present in the system, unlike other diagrams that are used to represent the architecture of the system, working of a system, or simply the system itself.

In UML, the component diagram portrays the behavior and organization of components at any instant of time. The system cannot be visualized by any individual component, but it can be by the collection of components.

Fig 9: Component diagram for Krishi Bhavan Management System



4.2.8 OBJECT DIAGRAM

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

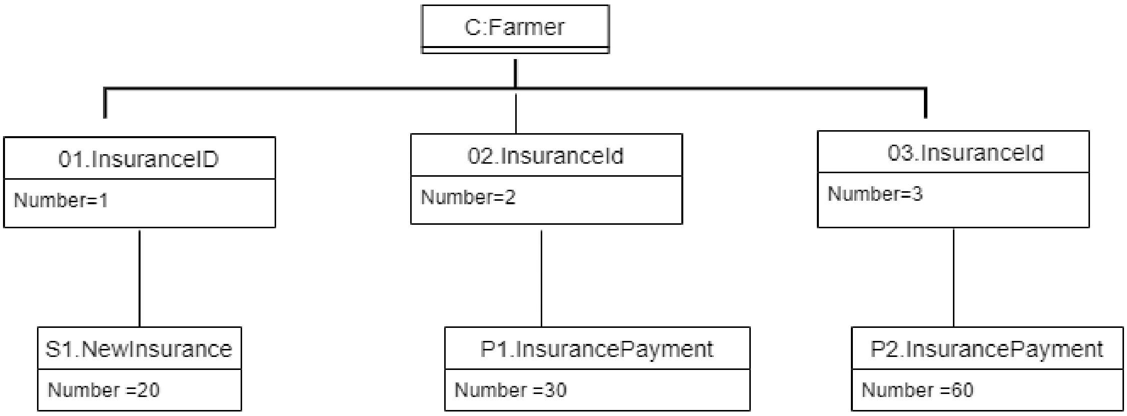
Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

Notations Used in Object Diagram–

- Objects or Instance specifications – When we instantiate a classifier in a system, the object we create represents an entity which exists in the system. We can represent the changes in object over time by creating multiple instance specifications. We use a rectangle to represent an object in an Object Diagram. An object is generally linked to other objects in an object diagram.
- Links – We use a link to represent a relationship between two objects.
- Dependency Relationships – We use a dependency relationship to show when one element depends on another element.
- Association – Association is a reference relationship between two objects (or classes).
- Aggregation – Aggregation represents a "has a" relationship.
- Composition – Composition is a type of association where the child cannot exist independent of the other.

Fig 10: Object diagram for Krishi Bhavan Management System



4.2.9 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

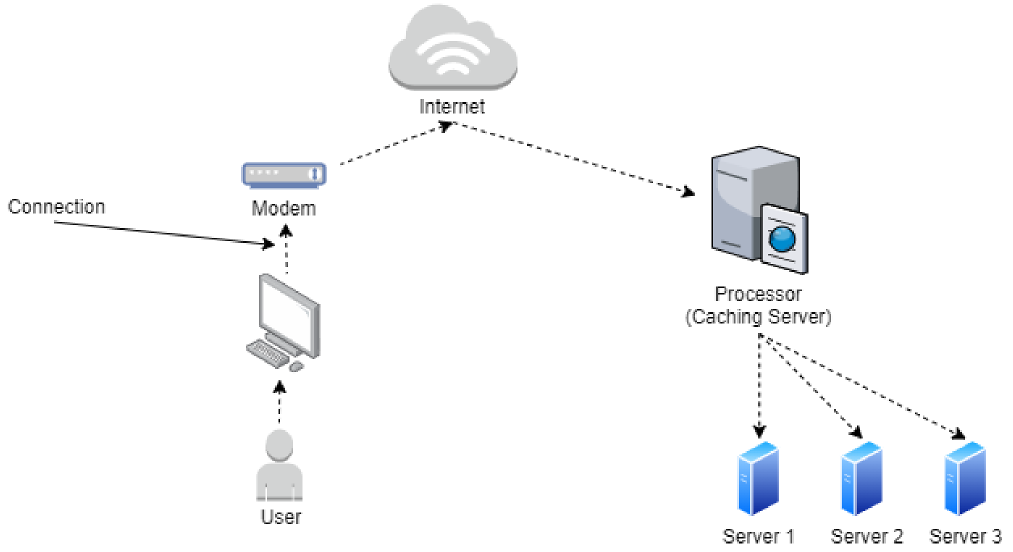
Deployment diagram represents the deployment view of a system. It is related to the component diagram because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.

Notation of Deployment diagram –

The deployment diagram consist of the following notations:

- Component: A rectangle with two tabs that indicates a software element.
- Artifact: A product developed by the software, symbolized by a rectangle with the name and the word "artifact" enclosed by double arrows.
- Interface: A circle that indicates a contractual relationship. Those objects that realize the interface must complete some sort of obligation.
- Node: A hardware or software object, shown by a three-dimensional box.

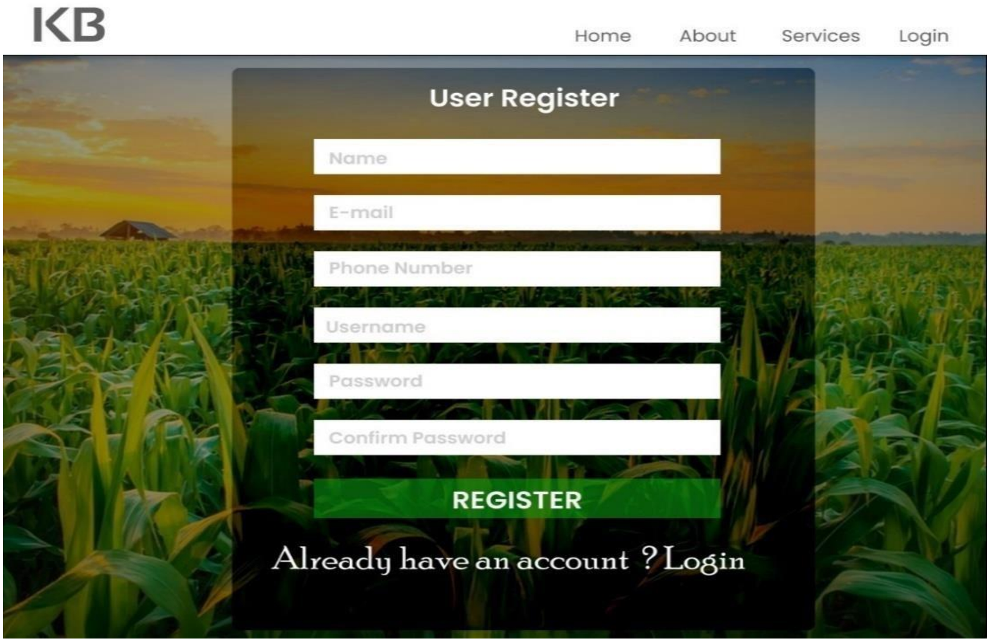
Fig 11 : Deployment diagram for Krishi Bhavan Management System



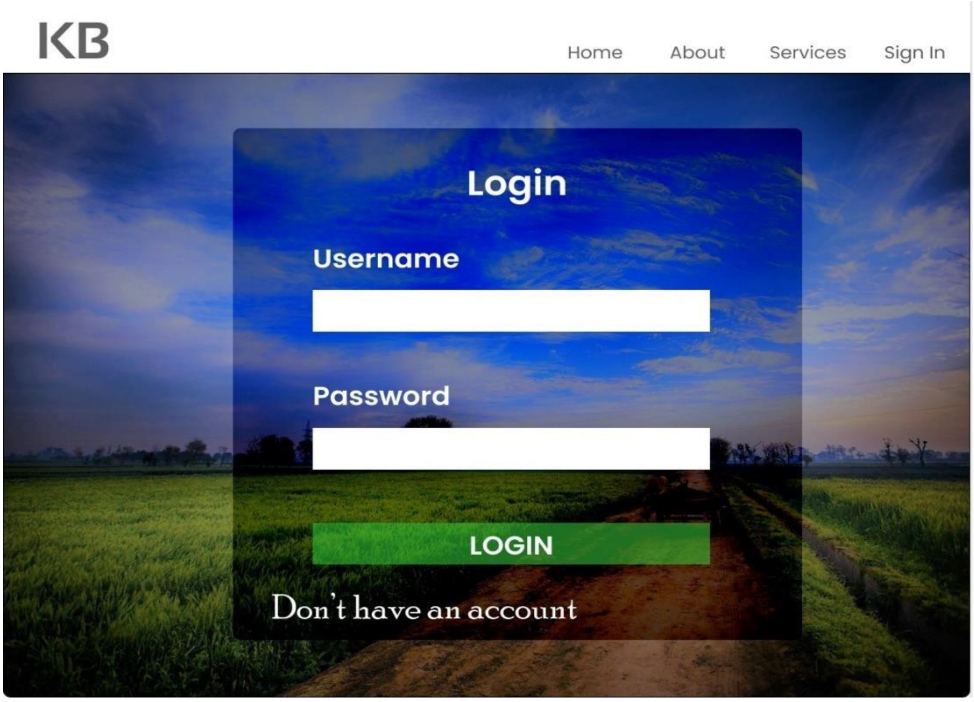
4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1-INPUT DESIGN

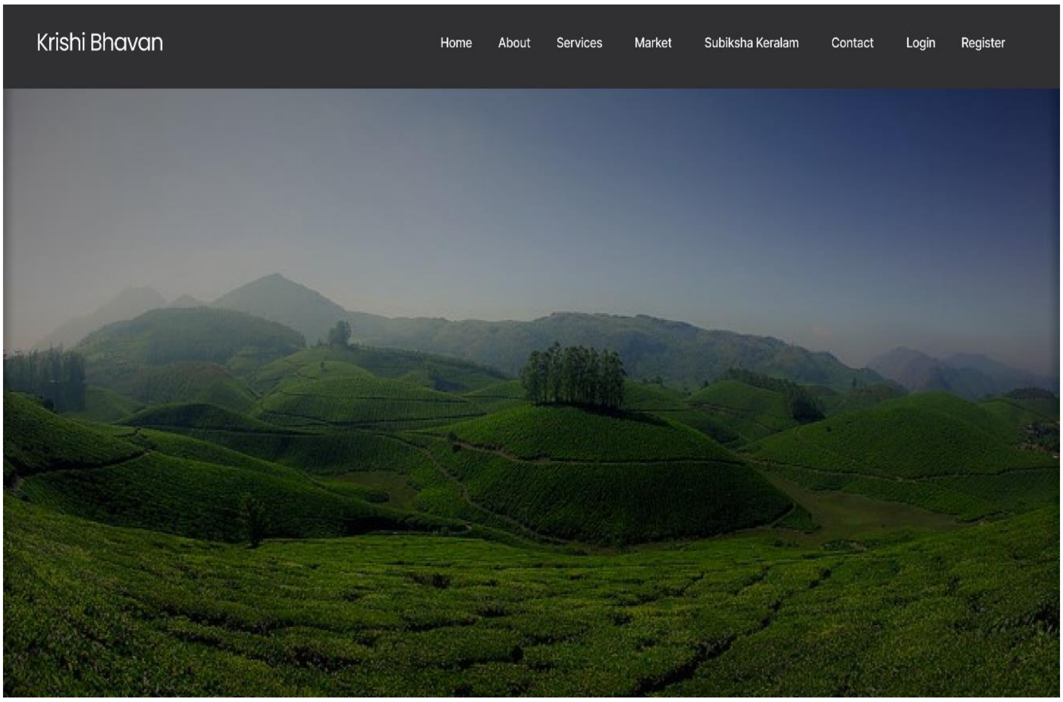
Form Name : User Registration



Form Name : User Login

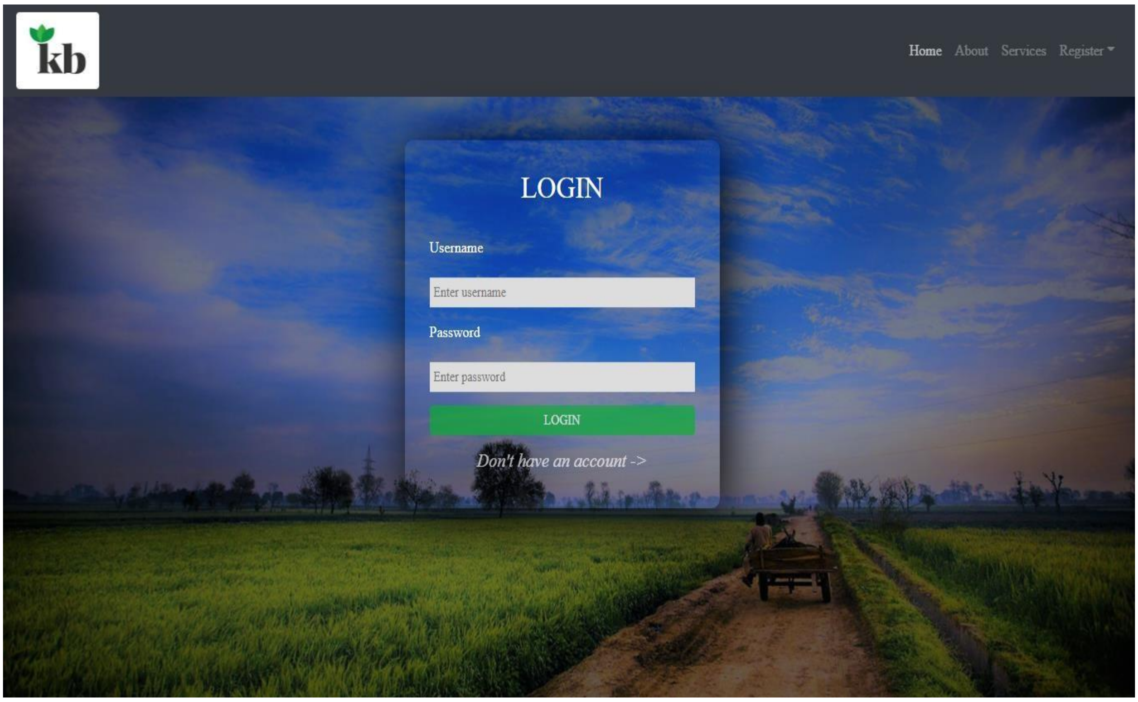


Form Name : User Home Page

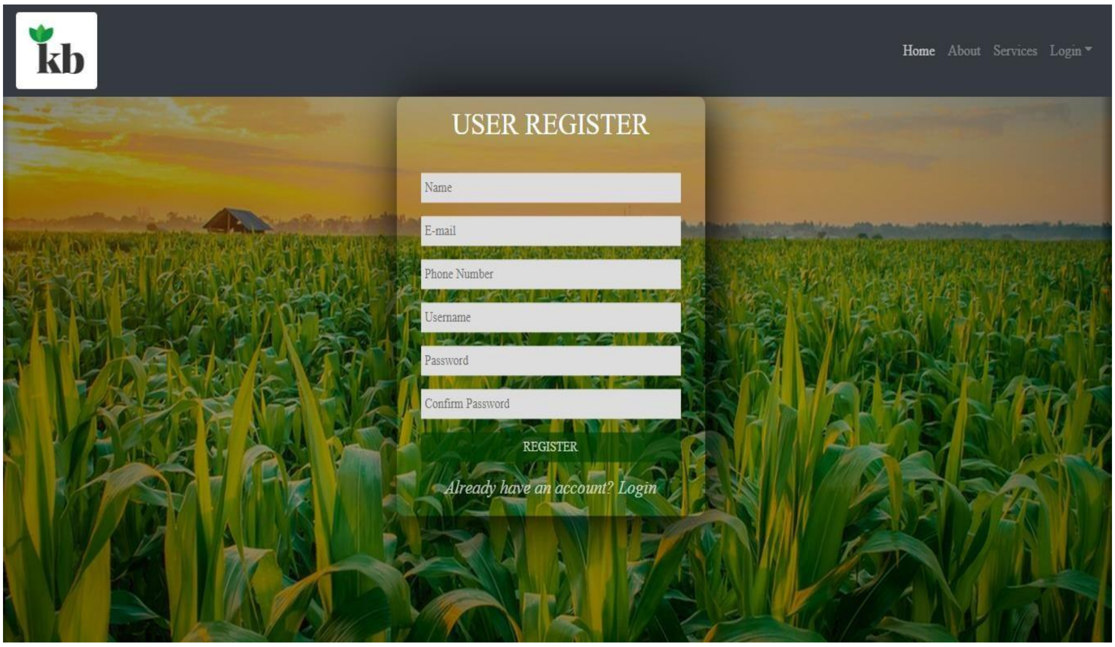


4.3.2 OUTPUT DESIGN

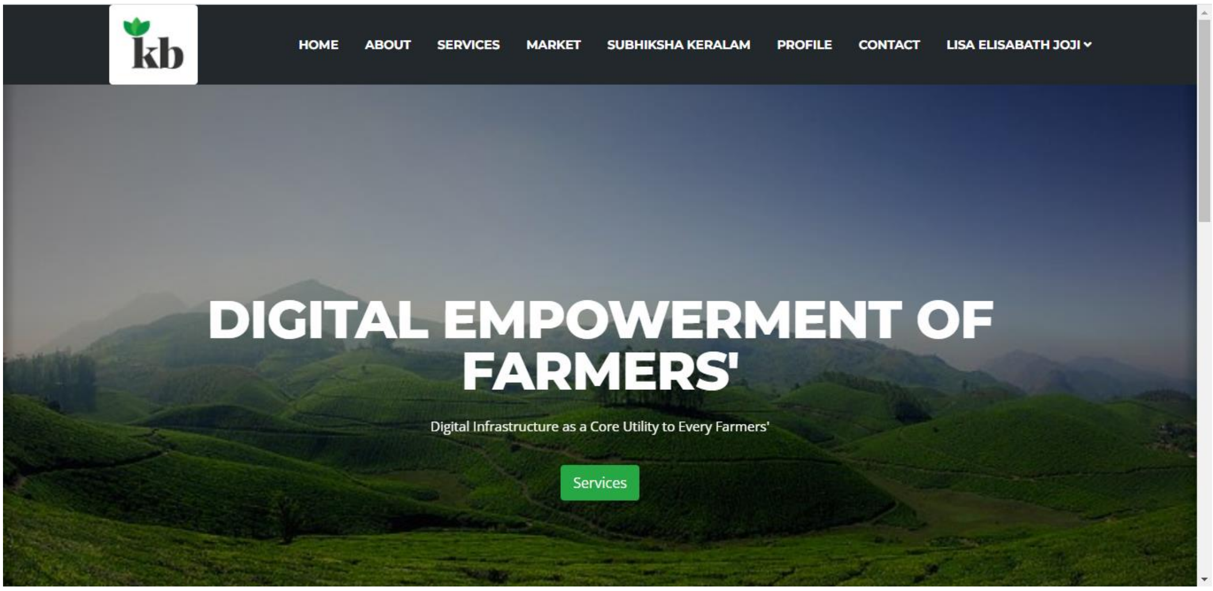
User Login



User Registration



Landing Page



DIGITAL EMPOWERMENT OF FARMERS'

Digital Infrastructure as a Core Utility to Every Farmers'

4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.4.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key is Super Key and Candidate Keys.

4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words, 1NF disallows “relations within relations” or “relations asattribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attribute of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN

Table No : 01

Table Name : tb_userreg

Primary Key : reg_id

Table Description : To store user registration information

Fieldname	Data Type	Size	Description
reg_id	int	50	Primary key
name	varchar	50	Name of the user
email	varchar	50	Email of the user
phone_number	varchar	50	Phone number of the user
username	varchar	50	Username of the user
password	varchar	50	Password
usertype	varchar	20	Type of user
status	varchar	10	State

Table No : 02

Table Name : tbl_employreg

Primary Key : emp_id

Table Description : To store employee registration information

Fieldname	Data Type	Size	Description
emp_id	int	50	Primary key
type_id	int	50	type_id from tbl_type
name	varchar	50	Name of the employee
address	varchar	50	Address of the employee
email	varchar	50	Email of the employee
phone	bigint	50	Phone number of the employee
password	varchar	50	Password
status	int	50	State

Table No : 03

Table Name : tb_service_details

Primary Key : serv_id

Table Description : To store service details information

Fieldname	Data Type	Size	Descriptions
serv_id	int	50	Primary key
service_name	varchar	100	Name of the service
description	varchar	200	Description of the service
scheme	varchar	200	Scheme of the service
component	long int	200	Component of the service
eligibilty	varchar	200	Eligibility for the service

Table No : 04

Table Name : tbl_crop_insurance

Primary Key : crop_id

Foreign Key : survey_number

Table Description : To store crop insurance requests

Fieldname	Data Type	Size	Descriptions
crop_id	int	50	Primary key
user_id	int	10	user_id from tbl_userreg
surveynumber	varchar	50	Survey number of the land
crop_name	varchar	50	Name of the crop insured
seeds	varchar	100	Quantity of seeds insured
ageforinsurance	long int	20	Crop age to claim insurance
imageproof	varchar	50	Image for proof
astatus	int	50	State
date	varchar	200	Date applied

Table No : 05

Table Name : tbl_paddy_royalty

Primary Key : ID

Foreign Key : survey_number

Table Description : To store paddy royalty requests

Fieldname	Data Type	Size	Descriptions
ID	int	50	Primary key
user_id	int	10	user_id from tbl_userreg
surveynumber	varchar	50	Survey number of the land
landname	varchar	50	Known name of the land
landarea	int	20	Area of land
landavailable	int	20	Land available for cultivation
cropdetails	varchar	200	Details of cultivated crops
imageproof	varchar	50	Image for proof
astatus	int	50	State
date	varchar	200	Date applied

Table No : 06

Table Name : tbl_baseprice

Primary Key : ID

Foreign Key : crop_name

Table Description : To store requests for crop base price

Fieldname	Data Type	Size	Descriptions
ID	int	50	Primary key
user_id	int	10	user_id from tbl_userreg
cropname	varchar	50	Crop details
cropnumber	int	50	Quantity of crop cultivated
photo	varchar	50	Image for proof

astatus	varchar	50	State
date	varchar	200	Date applied

Table No : 07

Table Name : tbl_soiltest

Primary Key : id

Table Description : To store request for soil test

Fieldname	Data Type	Size	Descriptions
id	int	50	Primary key
user_id	int	10	user_id from tbl_userreg
surveynumber	varchar	50	Survey number of the land
landname	varchar	50	Known name of the land
landarea	varchar	50	Area of land
soiltype	varchar	50	Type of soil
astatus	varchar	50	State
date	varchar	200	Date applied

Table No : 08

Table Name : tbl_type

Primary Key : type_id

Table Description : To store employee type

Fieldname	Data Type	Size	Descriptions
type_id	int	11	Primary key
type	varchar	50	Type of employee
status	int	11	State

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appears to be working according to the specification, that performance requirement appears to havebeen met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness is supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers are always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.1.1 Test Case

Test Case 1

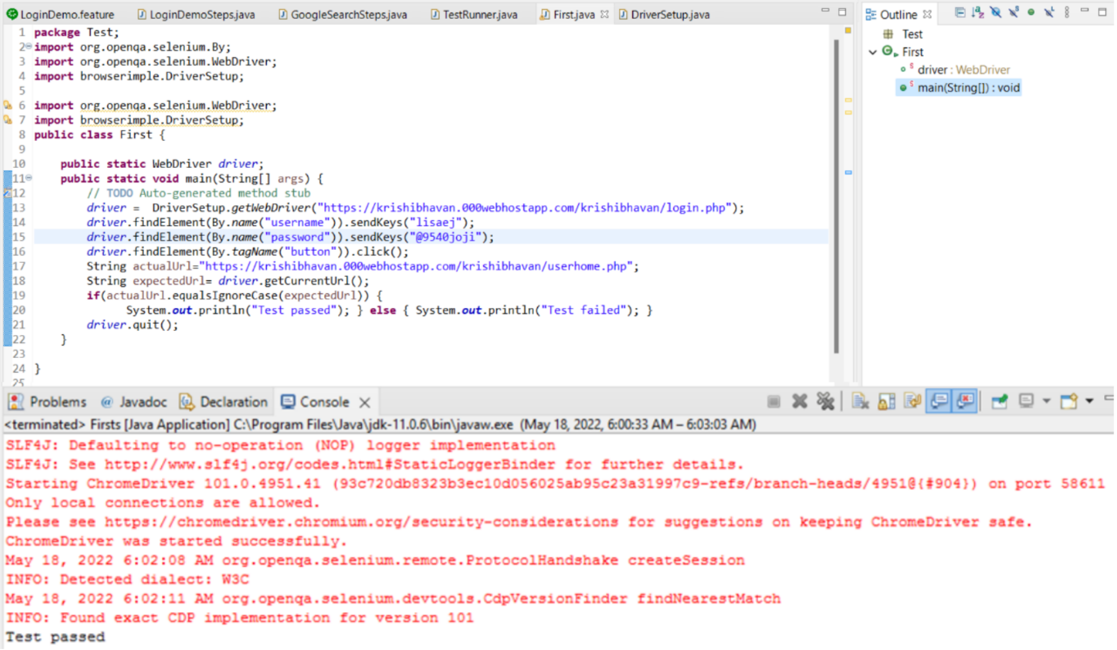
Project Name: Mobile Repair Management Website	
Login Test Case	
Test Case ID: Fun_1	**Test Designed By: Lisa Elisabath Joji**
Test Priority (Low/Medium/High): High	**Test Designed Date:** 17-05-2022
Module Name: Login Screen	**Test Executed By: Ms. Grace Joseph**
Test Title: Verify login with valid username and password	**Test Execution Date:** 18-05-2022
Description: Test the Login Page	
Pre-Condition: User has valid email id and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Username	email: lisajoji@gmail.com	User should be able to Login	User Logged in and navigated to user Dashboard	Pass
3	Provide Valid Password	Password: @9540joji			

4	Click on Sign In button			with records	
5	Provide Invalid username or pass word	email: lisajoji@gmail.com Password: @9540joji	User should not be able to Login	Message for enter valid username or password displayed	Pass
6	Provide Null Phone number orPassword	Username: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database					

Code:

```
package Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import browserimple.DriverSetup;


import org.openqa.selenium.WebDriver;

import browserimple.DriverSetup;

public class First {


public static WebDriver driver;

public static void main(String[] args) {

        // TODO Auto-generated method stub
        Driver=DriverSetup.getWebDriver("https://krishibhavan.000webhostapp.com/kris
        hibhavan/login.php");

        driver.findElement(By.name("email")).sendKeys("lisajoji@gmail.com");

        driver.findElement(By.name("password")).sendKeys("@9540joji");

        driver.findElement(By.tagName("button")).click();

        String
actualUrl="https://krishibhavan.000webhostapp.com/krishibhavan/userhome.php";

        String expectedUrl= driver.getCurrentUrl();

        if(actualUrl.equalsIgnoreCase(expectedUrl)) {

                System.out.println("Test passed"); }

        Else

         { System.out.println("Test failed"); }

        driver.quit();

}

}
```



```
package Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import browserimple.DriverSetup;

import org.openqa.selenium.WebDriver;
import browserimple.DriverSetup;
public class First {

    public static WebDriver driver;
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        driver =  DriverSetup.getWebDriver("https://krishibhavan.000webhostapp.com/krishibhavan/login.php");
        driver.findElement(By.name("username")).sendKeys("lisaej");
        driver.findElement(By.name("password")).sendKeys("@9540joji");
        driver.findElement(By.tagName("button")).click();
        String actualUrl="https://krishibhavan.000webhostapp.com/krishibhavan/userhome.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
              System.out.println("Test passed"); } else { System.out.println("Test failed"); }
        driver.quit();
    }

}
```

```
<terminated> Firsts [Java Application] C:\Program Files\Java\jdk-11.0.6\bin\javaw.exe  (May 18, 2022, 6:00:33 AM – 6:03:03 AM)
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 58611
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 6:02:08 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 6:02:11 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

5.2.1.2 Test Case

Test Case 2

Project Name: Mobile Repair Management Website	
Login Test Case	
Test Case ID: Fun_2	**Test Designed By: Lisa Elisabath Joji**
Test Priority (Low/Medium/High): High	**Test Designed Date:** 25-05-2022
Module Name: Admin	**Test Executed By: Ms. Grace Joseph**
Test Title: Register new employee	**Test Execution Date:** 25-05-2022
Description: Test the registration of new employee by admin	
Pre-Condition: Admin should be logged into the system	

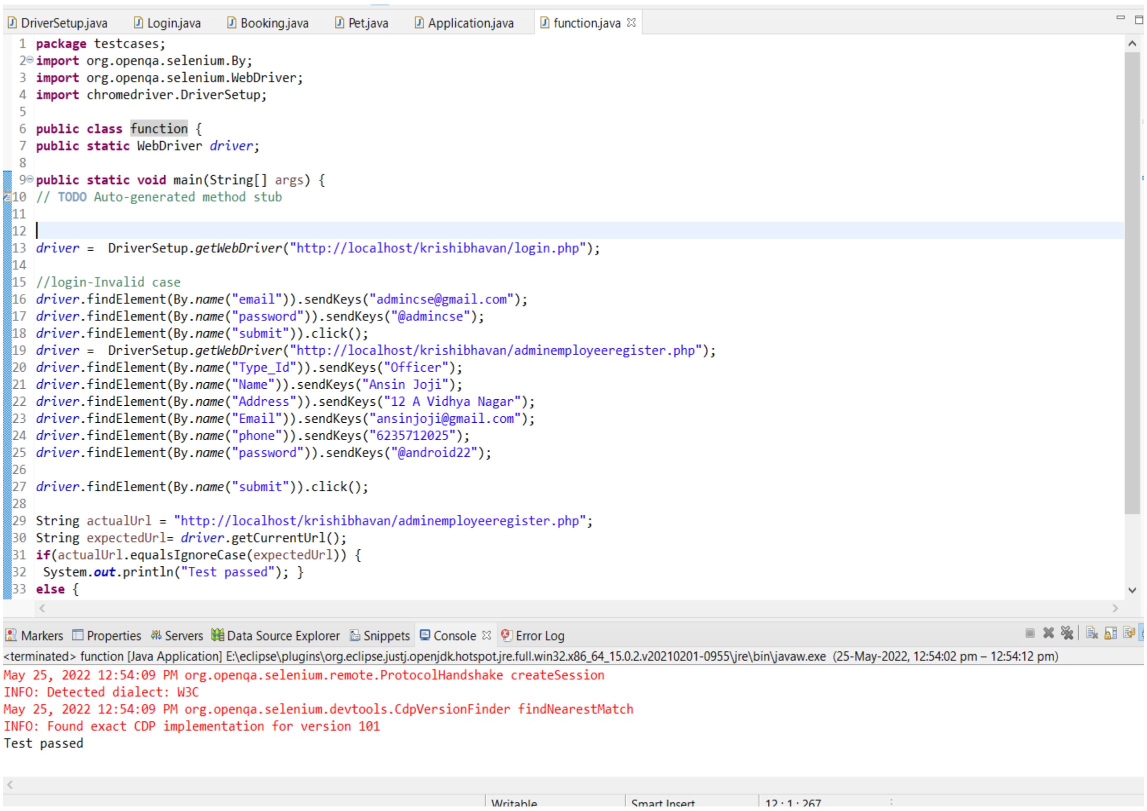
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Username	email: admincse@gmail.com	User should be able to Login	User Logged in and navigated to user Dashboard with records	Pass
3	Provide Valid Password	Password: @admincse			
4	Click on Sign In button				
5	Navigation to Employee Register page		Employee Register Page should be displayed	Employee Register Page displayed	Pass
6	Provide Valid Type_Id	Type_Id: Officer	Admin should be able to register new employee	Admin should be able to input values and redirect to admin dashboard	Pass
7	Provide Valid Name	Name: Ansin Joji			
8	Provide Valid Address	Address: 12 A Vidhya Nagar			

9	Provide Valid Address	Address: 12 A Vidhya Nagar			
10	Provide Valid Email	Email: ansinjoji@gmail.com			
11	Provide Valid Phone	Phone: 6235712025			
12	Provide Valid Password	Password: @android22			
13	Click on Add button				
Post-Condition: Admin registration details are validated with database and successfully registered.The Account session details are logged in database					

Code:

```
package testcases;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import chromedriver.DriverSetup;


public class Function {

  public static WebDriver driver;

  public static void main(String[] args) {

    driver =  DriverSetup.getWebDriver("http://localhost/krishibhavan/login.php");

    //login-Invalid case

    driver.findElement(By.name("email")).sendKeys("admincse@gmail.com");

    driver.findElement(By.name("password")).sendKeys("@admincse");

    driver.findElement(By.name("submit")).click();

    driver                                                                    =
DriverSetup.getWebDriver("http://localhost/krishibhavan/adminemployeeregister.php");
```

```
    driver.findElement(By.name("Type_Id")).sendKeys("Officer");
    driver.findElement(By.name("Name")).sendKeys("Ansin Joji");
    driver.findElement(By.name("Address")).sendKeys("12 A Vidhya Nagar");
    driver.findElement(By.name("Email")).sendKeys("ansinjoji@gmail.com");
    driver.findElement(By.name("phone")).sendKeys("6235712025");
    driver.findElement(By.name("password")).sendKeys("@android22");
    driver.findElement(By.name("submit")).click();
    String actualUrl = "http://localhost/krishibhavan/adminemployeeregister.php";
    String expectedUrl= driver.getCurrentUrl();
    if(actualUrl.equalsIgnoreCase(expectedUrl)) {
      System.out.println("Test passed");
    }
    else {
      System.out.println("Test failed");
    }
    driver.quit();
  }
}
```



```java
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;

public class function {
public static WebDriver driver;

public static void main(String[] args) {
// TODO Auto-generated method stub


driver =  DriverSetup.getWebDriver("http://localhost/krishibhavan/login.php");

//login-Invalid case
driver.findElement(By.name("email")).sendKeys("admincse@gmail.com");
driver.findElement(By.name("password")).sendKeys("@admincse");
driver.findElement(By.name("submit")).click();
driver =  DriverSetup.getWebDriver("http://localhost/krishibhavan/adminemployeeregister.php");
driver.findElement(By.name("Type_Id")).sendKeys("Officer");
driver.findElement(By.name("Name")).sendKeys("Ansin Joji");
driver.findElement(By.name("Address")).sendKeys("12 A Vidhya Nagar");
driver.findElement(By.name("Email")).sendKeys("ansinjoji@gmail.com");
driver.findElement(By.name("phone")).sendKeys("6235712025");
driver.findElement(By.name("password")).sendKeys("@android22");

driver.findElement(By.name("submit")).click();

String actualUrl = "http://localhost/krishibhavan/adminemployeeregister.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
 System.out.println("Test passed"); }
else {
```

```
<terminated> function [Java Application] E:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe  (25-May-2022, 12:54:02 pm – 12:54:12 pm)
May 25, 2022 12:54:09 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 25, 2022 12:54:09 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that willfully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.1 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is theprocess of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As systembecomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system is equipped with all the basic functionality for users such as crop insurance, paddy royalty, crop base price and soil test. User can request for the various available services. And the admin has the provision to manage the overall interaction of each user to the system. Admin can enable or disable the services added by the admin itself. Admin can also add the officers or assistant officers and can also view the employees. User can apply for crop insurance and then the request can be viewed by the officer. The officer can process the request as per the criteria given from the officials. Agricultural officer can manage the request as well as can approve or reject the request made by the user. User can also apply for paddy royalty and then the request can be viewed by the officer. Agricultural officer can manage the request as well as can approve or reject the request made by the user. The officer can process the request as per the criteria given from the officials User can also apply crop base price and then the request can be viewed by the officer. Agricultural officer can manage the request as well as can process the request well. User can also apply soil test and then the request can be viewed by agricultural assistant officer. Agricultural assistant officer can manage the request as well as can process the request well.

7.2FUTURE SCOPE

- The current application provides all the basic functionalities which makes the users the edge to prefer the web application over physical interaction.
- The application can be enhanced with advanced applications using machine learning.
- The automatic calculation of land area can be implemented using artificial intelligence. Using the functionality, the agricultural assistant officer can survey the area of land with regards to verification of land area details provided by the applicant.
- Furthermore, various existing processes can fully be automated.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “So*ftware engineering*: a precise approach”, 2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- https://getbootstrap.com/docs/4.0/getting-started/introduction/
- https://stackoverflow.com/
- https://app.diagrams.net
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample Code

Customer

cropinsuranceapply.php

```
<?php
include("includes/connection.php");
?>
<header id="header" class="header-one">
  <?php
  include("includes/uheader.php");
  ?>
</header>
<?php

if (isset($_SESSION["id"]) != session_id()) {

  echo ("<script LANGUAGE='JavaScript'>
        window.alert('Login first');
        window.location.href='login.php';
      </script>");
} else {

  if (isset($_POST["submit"])) {

    $surveynum = $_POST["snum"];
    $crnam = $_POST["cname"];
    $cseed = $_POST["seed"];
    $crag = $_POST["age"];
    $pic = $_FILES["dataFile"]["name"];
    move_uploaded_file($_FILES["dataFile"]["tmp_name"], "images/" .
$_FILES["dataFile"]["name"]);
    $accnum = $_POST["acnumber"];
    $ifscbank = $_POST["ifsc"];
    $userid = $_SESSION["userid"];

    mysqli_query($con, "INSERT INTO
`tbl_cropinsurance`(`user_id`,`surveynumber`,`crop_name`, `seeds`,
`ageforinsurance`,`imageProof`,`accountnumber`,`ifsccode`, `date`) VALUES
('$surveynum','$userid','$crnam','$cseed','$crag','$pic','$accnum','$ifscbank', now())");
    header("location:viewcalrelfapply.php");
  }
?>

  <!DOCTYPE html>
  <html lang="en">

  <head>

    <!-- Basic Page Needs
================================================== -->
    <meta charset="utf-8">
```

```
<title>Services Details</title>

  <!-- Mobile Specific Metas
================================================== -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="description" content="Construction Html5 Template">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-
scale=5.0">

  <!-- Favicon
================================================== -->
  <link rel="icon" type="image/jpeg" href="images/logo/title.jpeg">

  <!-- CSS
================================================== -->
  <!-- Bootstrap -->
  <link rel="stylesheet" href="plugins/bootstrap/bootstrap.min.css">
  <!-- FontAwesome -->
  <link rel="stylesheet" href="plugins/fontawesome/css/all.min.css">
  <!-- Animation -->
  <link rel="stylesheet" href="plugins/animate-css/animate.css">
  <!-- slick Carousel -->
  <link rel="stylesheet" href="plugins/slick/slick.css">
  <link rel="stylesheet" href="plugins/slick/slick-theme.css">
  <!-- Colorbox -->
  <link rel="stylesheet" href="plugins/colorbox/colorbox.css">
  <!-- Template styles-->
  <link rel="stylesheet" href="css/style.css">


 </head>

 <body>
  <div class="body-inner">
   <div id="banner-area" class="banner-area" style="background-
image:url(images/banner/b2.png)">
    <div class="banner-text">
     <div class="container">
      <div class="row">
       <div class="col-lg-12">
        <div class="banner-heading">
         <h1 class="banner-title">Service</h1>
         <nav aria-label="breadcrumb">
          <ol class="breadcrumb justify-content-center">
           <li class="breadcrumb-item"><a href="adminhome.php">Home</a></li>
           <li class="breadcrumb-item"><a href="#">Services</a></li>
           <li class="breadcrumb-item active" aria-current="page">crop insurance</li>
          </ol>
         </nav>
        </div>
       </div><!-- Col end -->
      </div><!-- Row end -->
     </div><!-- Container end -->
    </div><!-- Banner text end -->
   </div><!-- Banner area end -->
 <section id="main-container" class="main-container">

<div class="container">
        <div class="row">

          <div class="col-xl-3 col-lg-4">
            <div class="sidebar sidebar-left">
              <div class="widget">
                <h3 class="widget-title"></h3>
                <ul class="nav service-menu">
                  <li class="active"><a href="#">Crop Insurance Apply</a></li>
                  <li><a href="viewcalrelfapply.php">Crop Insurance Requests</a></li>
                </ul>
              </div><!-- Widget end -->
            </div><!-- Sidebar end -->
          </div><!-- Sidebar Col end -->

          <div class="col-xl-8 col-lg-8">
            <div class="content-inner-page">
              <div class="pb-4">
                <h3 class="bg-secondary text-light font-weight-light">Appy for Crop 
Insurance</h3>

              </div>

              <script>
                function validate5() {
                  var name = document.getElementById("snum").value;
                  var letters = /^[^a-zA-Z\s]*$/;
                  if (!name.match(letters)) {
                    alert("Please enter survey number correctly..");
                    document.getElementById("snum").value = "";
                  }
                }

                function validate() {
                  var name = document.getElementById("snum").value;
                  var letters = /^[^a-zA-Z\s]*$/;
                  if (!name.match(letters)) {
                    alert("Please enter survey number correctly..");
                    document.getElementById("snum").value = "";
                  }
                }

                function validate1() {
                  var name = document.getElementById("cname").value;
                  var letters = /^[a-zA-Z\s]*$/;
                  if (!name.match(letters)) {
                    alert("Please enter land name correctly");
                    document.getElementById("cname").value = "";
                  }
                }

                function validate2() {
                  var name = document.getElementById("seed").value;
                  var letters = /^[0-9.]*$/;
```

```
           if (!name.match(letters)) {
           alert("Please enter land area detail correctly..");
           document.getElementById("seed").value = "";

 }
         }

         function validate3() {
          var name = document.getElementById("lno").value;
          var letters = /^[0-9.]*$/;
          if (!name.match(letters)) {
           alert("Please enter available land area detail correctly..");
           document.getElementById("lno").value = "";
          }
         }

         function validate4() {
          var name = document.getElementById("age").value;
          var letters = /^[0-9.]*$/;
          if (!name.match(letters)) {
           alert("Please enter available land area detail correctly..");
           document.getElementById("age").value = "";
          }
         }
        </script>
        <div class="container">

         <form action="#" method="POST">
          <div class="form-group">
           <label for="name">Survey Number</label>
           <input type="text" class="form-control" id="snum" name="snum"
onblur="validate5()" required />
           <label><span class="textfieldRequiredMsg"></span></label>
          </div>
          <div class="form-group">
           <label for="name">Crop Name</label>
           <input type="text" class="form-control" id="cname" name="cname"
onblur="validate()" required />
           <label><span class="textfieldRequiredMsg"></span></label>
          </div>
          <div class="form-group">
           <label for="desc">Seeds(In Grams)</label>
           <input type="text" class="form-control" id="seed" name="seed"
onblur="validate1()" required />
           <label><span class="textfieldRequiredMsg"></span></label>
          </div>
          <div class="form-group">
           <label for="desc">Crop Age (in months)</label>
           <input type="text" class="form-control" id="age" name="age"
onblur="validate3()" required />
           <label><span class="textfieldRequiredMsg"></span></label>
          </div>
          <div class="form-group">
           <label for="name">Upload file</label>
           <input type="file" name="dataFile" id="fileChooser" onchange="return
```

```
                        ValidateFileUpload()" required />
                  <label><span class="textfieldRequiredMsg"></span></label>
                </div>
                <div class="form-group">
                  <label for="desc">Account Number</label>

<input type="text" class="form-control" id="acnumber" name="acnumber"
onblur="validate3()" required />
                  <label><span class="textfieldRequiredMsg"></span></label>
                </div>
                <div class="form-group">
                  <label for="desc">IFSC Code</label>
                        <input type="text" class="form-control" id="ifsc" name="ifsc"
      onblur="validate3()"   required />
                  <label><span class="textfieldRequiredMsg"></span></label>
                </div>
                <button type="submit" class="btn btn-primary" name="submit">Apply</button>
              </form>

    </section><!-- Main container end -->

    <footer id="footer" class="footer bg-overlay">
      <?php include("includes/footer.php"); ?>
    </footer><!-- Footer end -->

    <!-- Javascript Files
  ================================================== -->

    <!-- initialize jQuery Library -->
    <script src="plugins/jQuery/jquery.min.js"></script>
    <!-- Bootstrap jQuery -->
    <script src="plugins/bootstrap/bootstrap.min.js" defer></script>
    <!-- Slick Carousel -->
    <script src="plugins/slick/slick.min.js"></script>
    <script src="plugins/slick/slick-animation.min.js"></script>
    <!-- Color box -->
    <script src="plugins/colorbox/jquery.colorbox.js"></script>
    <!-- shuffle -->
    <script src="plugins/shuffle/shuffle.min.js" defer></script>

    <!-- Template custom -->
    <script src="js/script.js"></script>

   </div><!-- Body inner end -->

  </body>

<?php
}

?>

  </html>
```

Administrator

adminemployeeregister.php

```
 <?php
include("includes/connection.php");
?>
<header id="header" class="header-one">
   <?php
   include("includes/adminheader.php");
   ?>
</header>
<?php
if (isset($_SESSION["id"]) != session_id()) {

   echo ("<script LANGUAGE='JavaScript'>
      window.alert('Login first');
      window.location.href='login.php';
    </script>");
} else {
?>
   <!DOCTYPE html>
   <html lang="en">

   <head>

      <!-- Basic Page Needs
==================================================== -->
      <meta charset="utf-8">
      <title>Services Details</title>

      <!-- Mobile Specific Metas
==================================================== -->
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="description" content="Construction Html5 Template">
      <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-
scale=5.0">

      <!-- Favicon
==================================================== -->
      <link rel="icon" type="image/jpeg" href="images/logo/title.jpeg">

      <!-- CSS
==================================================== -->
      <!-- Bootstrap -->
      <link rel="stylesheet" href="plugins/bootstrap/bootstrap.min.css">
      <!-- FontAwesome -->
      <link rel="stylesheet" href="plugins/fontawesome/css/all.min.css">
      <!-- Animation -->
```

```
        <link rel="stylesheet" href="plugins/animate-css/animate.css">
        <!-- slick Carousel -->
        <link rel="stylesheet" href="plugins/slick/slick.css">
        <link rel="stylesheet" href="plugins/slick/slick-theme.css">
        <!-- Colorbox -->
        <link rel="stylesheet" href="plugins/colorbox/colorbox.css">
        <!-- Template styles-->
        <link rel="stylesheet" href="css/style.css">


    </head>

    <body>
        <div class="body-inner">
            <div id="banner-area" class="banner-area" style="background-
image:url(images/banner/b2.png)">
                <div class="banner-text">
                    <div class="container">
                        <div class="row">
                            <div class="col-lg-12">
                                <div class="banner-heading">
                                    <h1 class="banner-title">Service</h1>
                                    <nav aria-label="breadcrumb">
                                        <ol class="breadcrumb justify-content-center">
                                            <li class="breadcrumb-item"><a
href="adminhome.php">Home</a></li>
                                            <li class="breadcrumb-item"><a href="#">Services</a></li>
                                            <li class="breadcrumb-item active" aria-current="page">Register
New Employee</li>
                                        </ol>
                                    </nav>
                                </div>
                            </div><!-- Col end -->
                        </div><!-- Row end -->
                    </div><!-- Container end -->
                </div><!-- Banner text end -->
            </div><!-- Banner area end -->

            <section id="main-container" class="main-container">
                <div class="container">
                    <div class="row">

                        <div class="col-xl-3 col-lg-4">
                            <!-- Sidebar end -->
                        </div><!-- Sidebar Col end -->

                        <div class="col-xl-8 col-lg-8">
                            <div class="content-inner-page">

                                <!--<h2 class="column-title mrt-0">Natural Disaster Relief</h2>-->
                    <h3 class="bg-danger text-light font-weight-light"> NEW EMPLOYEE
REGISTRATION</h3>

                <script>
                    function validate() {
                        var name = document.getElementById("sname").value;
                        var letters = /^[a-zA-Z\s]*$/;
                        if (!name.match(letters)) {
                            alert("Please enter service name correctly without a number");
                            document.getElementById("sname").value = "";
                        }
                    }

                    function validate1() {
                        var name = document.getElementById("desc").value;
                        var letters = /^[a-zA-Z\s]*$/;
                        if (!name.match(letters)) {
                            alert("Please enter service description correctly");
                            document.getElementById("desc").value = "";
                        }
                    }

                    function validate2() {
                        var name = document.getElementById("scheme").value;
                        var letters = /^[a-zA-Z\s]*$/;
                        if (!name.match(letters)) {
                            alert("Please enter scheme correctly without a number");
                            document.getElementById("scheme").value = "";
                        }
                    }

                    function validate3() {
                        var name = document.getElementById("comp").value;
                        var letters = /^[a-zA-Z\s]*$/;
                        if (!name.match(letters)) {
                            alert("Please enter component correctly without a number");
                            document.getElementById("comp").value = "";
                        }
                    }

                    function validate4() {
                        var name = document.getElementById("elig").value;
                        var letters = /^[a-zA-Z\s]*$/;
                        if (!name.match(letters)) {
                            alert("Please enter eligibility criteria correctly without a number");
                            document.getElementById("elig").value = "";
                        }
                    }
                </script>
                <div class="container">
```

```
                              <form action="addnewemployee.php" method="POST">
                                <div class="form-group">
                                   <label for="name">Employee Name</label>
                                   <select class="form-control" name="etype" id="etype">
                                      <option selected disabled>Employee role</option>
                                      <option value="1">Officer</option>
                                      <option value="2">Assistant Officer</option>
                                   </select> <label><span
class="textfieldRequiredMsg"></span></label>
                                   </div>
                                   <div class="form-group">
                                      <label for="name">Employee Name</label>
                                      <input type="text" class="form-control" id="ename"
   name="ename" onblur="validate()" required />
                                      <label><span class="textfieldRequiredMsg"></span></label>
                                   </div>
                                   <div class="form-group">
                                      <label for="name">Address</label>
                                      <input type="text" class="form-control" id="eadd" name="eadd"
   onblur="validate()" required />
                                      <label><span class="textfieldRequiredMsg"></span></label>
                                   </div>
                                   <div class="form-group">
                                      <label for="name">Email</label>
                                      <input type="text" class="form-control" id="eemail"
   name="eemail" onblur="validate()" required />
                                      <label><span class="textfieldRequiredMsg"></span></label>
                                   </div>
                                   <div class="form-group">
                                      <label for="name">Phone</label>
                                      <input type="text" class="form-control" id="ephone"
   name="ephone" onblur="validate()" required />
                                      <label><span class="textfieldRequiredMsg"></span></label>
                                   </div>
                                   <div class="form-group">
                                      <label for="name">Password</label>
                                      <input type="password" class="form-control" id="epass"
   name="epass" onblur="validate()" required />
                                      <label><span class="textfieldRequiredMsg"></span></label>
                                   </div>
                                   <button type="submit" class="btn btn-primary"
   name="adds">Add</button>
                                </form>
                             </div>

                          </div><!-- Content inner end -->
                       </div><!-- Content Col end -->

    </div><!-- Main row end -->
```

```
            </div><!-- Conatiner end -->
        </section><!-- Main container end -->

        <footer id="footer" class="footer bg-overlay">
            <?php include("includes/footer.php"); ?>
        </footer><!-- Footer end -->

        <!-- Javascript Files
  ================================================== -->

        <!-- initialize jQuery Library -->
        <script src="plugins/jQuery/jquery.min.js"></script>
        <!-- Bootstrap jQuery -->
        <script src="plugins/bootstrap/bootstrap.min.js" defer></script>
        <!-- Slick Carousel -->
        <script src="plugins/slick/slick.min.js"></script>
        <script src="plugins/slick/slick-animation.min.js"></script>
        <!-- Color box -->
        <script src="plugins/colorbox/jquery.colorbox.js"></script>
        <!-- shuffle -->
        <script src="plugins/shuffle/shuffle.min.js" defer></script>

        <!-- Template custom -->
        <script src="js/script.js"></script>

    </div><!-- Body inner end -->

  </body>

<?php
}

?>

</html>
```

Agricultural Officer

cropinsurancestatus.php

```
<?php
include("includes/connection.php");
include("includes/nav1.php");
if (isset($_SESSION["id"]) != session_id()) {

    echo ("<script LANGUAGE='JavaScript'>
        window.alert('Login first');
        window.location.href='login.php';
     </script>");
```

```
} else {

?>

    <?php
    if (isset($_POST['rbtn'])) {
        include "includes/connection.php";
        $id = $_POST['rbtn'];
        $sval = $_POST['status'];
        $sql = "UPDATE `tbl_cropinsurance` SET `astatus`= '$sval' WHERE `crop_id` = $id";
        $result = mysqli_query($con, $sql);
    }
    ?>

    <html>

    <head>
        <title>Officer Home</title>

        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

        <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>


    </head>

    <body>
        <div class="ml-5 mt-5">
            <h3 class="text-center">CROP INSURANCE REQUEST</h3><br><br>
            <table class="table table-bordered css-serial">
                <thead class="">
                    <tr>

                        <th scope="col">Name</th>
                        <th scope="col">Date Applied</th>
                        <th scope="col">Survey Number</th>
                        <th scope="col">Crop Name</th>
                        <th scope="col">Seeds Quantity (in grmas)</th>
                        <th scope="col">Crop age (in Months)</th>

                        <th scope="col">Account Number</th>
                        <th scope="col">IFSC Code</th>
                        <th scope="col">Status</th>
                        <th scope="col">Status</th>

                    </tr>
                </thead>
                <tbody>

                    <?php
                    include("includes/connection.php");
                    $sql = "SELECT
`crop_id`,`user_id`,`surveynumber`,`crop_name`,`seeds`,`ageforinsurance`,`imageProof`,`acc
ountnumber`,`ifsccode`,`astatus`,`date` FROM tbl_cropinsurance";
                    $result = mysqli_query($con, $sql);
                    $rows = mysqli_num_rows($result);
                    if (mysqli_num_rows($result) > 0) {
                        while ($row = mysqli_fetch_assoc($result)) {

                            $fetchName = "SELECT * FROM `tb_userreg` WHERE `reg_id`='" .
$row['user_id'] . "'";
                            $fetchNameResult = mysqli_query($con, $fetchName);
                            if (mysqli_num_rows($fetchNameResult) > 0) {
                                while ($fetchNameRow = mysqli_fetch_assoc($fetchNameResult)) {
                                    echo "<td scope='row'>" . $fetchNameRow['name'] . "</th>";
                                }
                            }
                            echo "
                            <td scope='row'>" . $row['date'] . "</th>
                            <td scope='row'>" . $row['surveynumber'] . "</th>
                            <td scope='row'>" . $row['crop_name'] . "</th>
                            <td scope='row'>" . $row['seeds'] . "</th>
                            <td scope='row'>" . $row['ageforinsurance'] . "</th>

                            <td scope='row'>" . $row['accountnumber'] . "</th>
                            <td scope='row'>" . $row['ifsccode'] . "</th>

                            <td scope='row'>";
                            if ($row['astatus'] == 1) {
                                echo "<p class='card-text'>Processed</p>";
                            } else if ($row['astatus'] == 2) {
                                echo "<p class='card-text text-secondary'>Approved</p>";
                            } else if ($row['astatus'] == 3) {
                                echo "<p class='card-text text-secondary'>Rejected</p>";
                            } else {
                                null;
                            }
                        echo "</th>";
                        echo "<td>

                       <form method='post' action='#'>
                       <select class='custom-select' id='inputGroupSelect01' name='status'>
                          <option selected>Choose...</option>
                          <option value='1'>Processed</option>
                          <option value='2'>Approved</option>
                          <option value='3'>Rejected</option>
                        </select>
                        <button class='btn btn-success' value='" . $row['crop_id'] . "'
name=rbtn>Update</button>
                       </form>
                       </td></tr>";
                     }
                   }
                   ?>

                </tbody>
              </table>
            </div>
        </body>

<?php
}
?>


        </html>
```

Agricultural Assistant Officer

soilteststatus.php

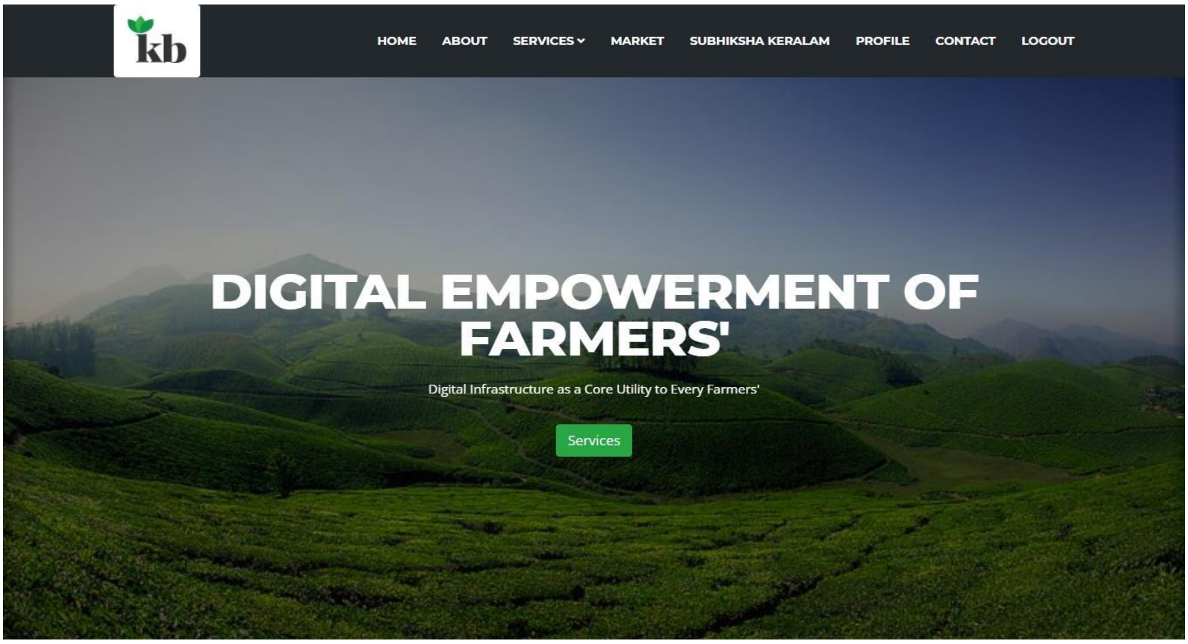
```
<?php
include("includes/connection.php");
include("includes/nav_assistant.php");
if (isset($_SESSION["id"]) != session_id()) {

  echo ("<script LANGUAGE='JavaScript'>
        window.alert('Login first');
        window.location.href='emplogin.html';
      </script>");
} else {

?>

  <?php
  if (isset($_POST['rbtn'])) {
    include "includes/connection.php";
    $id = $_POST['rbtn'];
    $sval = $_POST['status'];
    $sql = "UPDATE `tbl_soiltest` SET `status`= '$sval' WHERE Id = $id";
    //echo $sql;

    $result = mysqli_query($con, $sql);
    //echo "hai";
  }
  ?>

  <html>

  <head>
    <title>Officer Home</title>

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>


  </head>

  <body>
    <div class="ml-5 mt-5">
      <h3>Request for soil test</h3>
      <table class="table table-bordered css-serial">
        <thead class="">
          <tr>
            <th scope="col">Date Applied</th>
            <th scope="col">Name</th>
            <th scope="col">Survey Number</th>
            <th scope="col">Land Name</th>
            <th scope="col">Land Area</th>
            <th scope="col">Soil Type</th>
            <th scope="col">Status</th>
            <th scope="col">Action</th>
```

```
         </tr>
        </thead>
        <tbody>

         <?php
         include("includes/connection.php");
         $sql = "SELECT * FROM tbl_soiltest";
         $result = mysqli_query($con, $sql);
         if (mysqli_num_rows($result) > 0) {
           while ($row = mysqli_fetch_assoc($result)) {
             echo "
             <tr>
             <td scope='row'>" . $row['date'] . "</th>";
             $fetchName = "SELECT * FROM `tb_userreg` WHERE `reg_id`='" . $row['user_id']
 . "'";
             $fetchNameResult = mysqli_query($con, $fetchName);
             if (mysqli_num_rows($fetchNameResult) > 0) {
               while ($fetchNameRow = mysqli_fetch_assoc($fetchNameResult)) {
                 echo "<td scope='row'>" . $fetchNameRow['name'] . "</th>";
               }
             }
             echo "
             <td scope='row'>" . $row['date'] . "</th>
             <td scope='row'>" . $row['user_id'] . "</th>
             <td scope='row'>" . $row['surveynumber'] . "</th>
             <td scope='row'>" . $row['landname'] . "</th>
             <td scope='row'>" . $row['landarea'] . "</th>
             <td scope='row'>" . $row['soiltype'] . "</th>";
             if ($row['status'] == 1) {
               echo "<td scope='row'>Underprocessing</th>";
             } else if ($row['status'] == 2) {
               echo "<td scope='row'>processed</th>";
             } else if ($row['status'] == 3) {
               echo "<td scope='row'>approved</th>";
             } else if ($row['status'] == 4) {
               echo "<td scope='row'>rejected</th>";
             }
             echo "</th>";
             echo "<td>

                          <form method=post>
                          <select class='custom-select' id='inputGroupSelect01' name='status'>
                              <option selected>Choose...</option>
                              <option value='1'>Underprocessing</option>
                              <option value='2'>processed</option>
                              <option value='3'>approved</option>
                              <option value='4'>rejected</option>
                            </select>
                            <button class='btn btn-success' value='" . $row['Id'] . "'
 name=rbtn>Update</button>
                          </form>
                          </td>";

                  "</tr>";
                 }
               }
               ?>

             </tbody>
           </table>
         </div>
       </body>

     <?php
     }
     ?>


       </html>
```

9.2 Screenshots

CUSTOMER PAGES

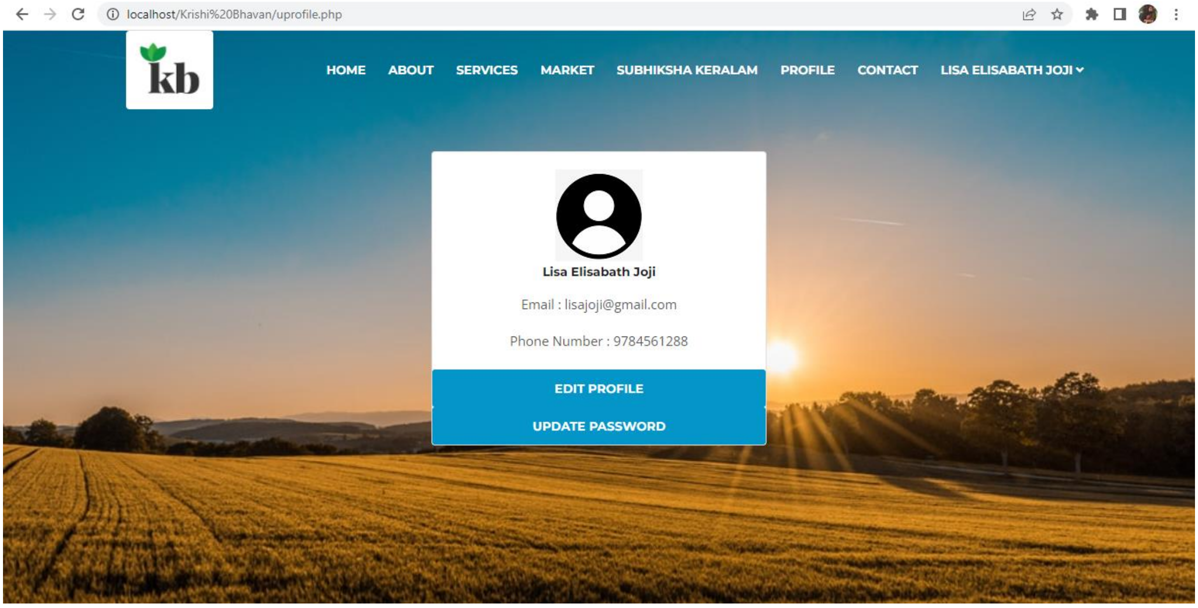
Landing Page



DIGITAL EMPOWERMENT OF FARMERS'

Digital Infrastructure as a Core Utility to Every Farmers'

Account page

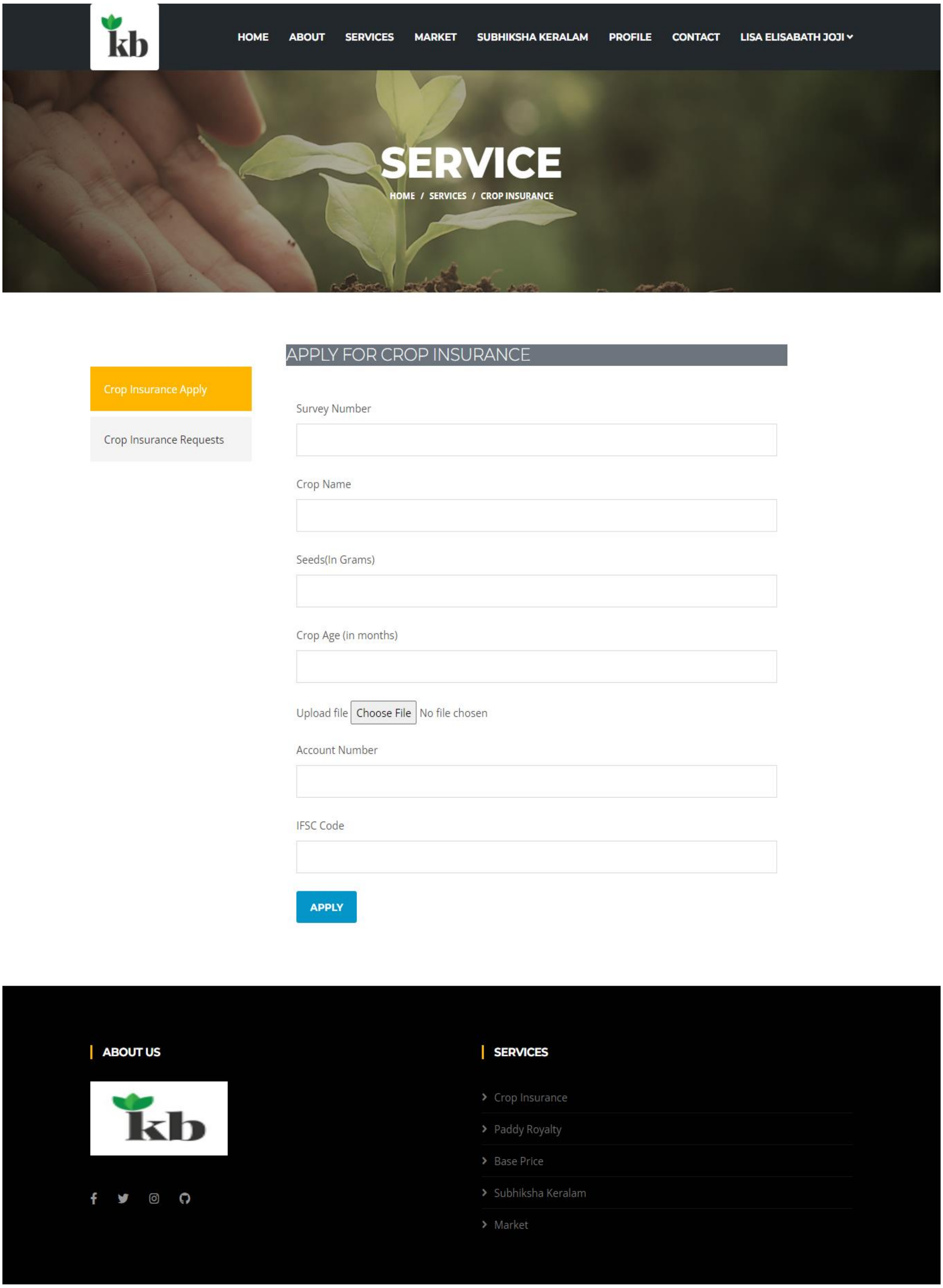


Lisa Elisabath Joji

Email : lisajoji@gmail.com

Phone Number : 9784561288

Apply for Crop Insurance



kb
HOME ABOUT SERVICES MARKET SUBHIKSHA KERALAM PROFILE CONTACT LISA ELISABATH JOJI
SERVICE
HOME / SERVICES / CROP INSURANCE
Crop Insurance Apply
Crop Insurance Requests

APPLY FOR CROP INSURANCE

Survey Number

Crop Name

Seeds(In Grams)

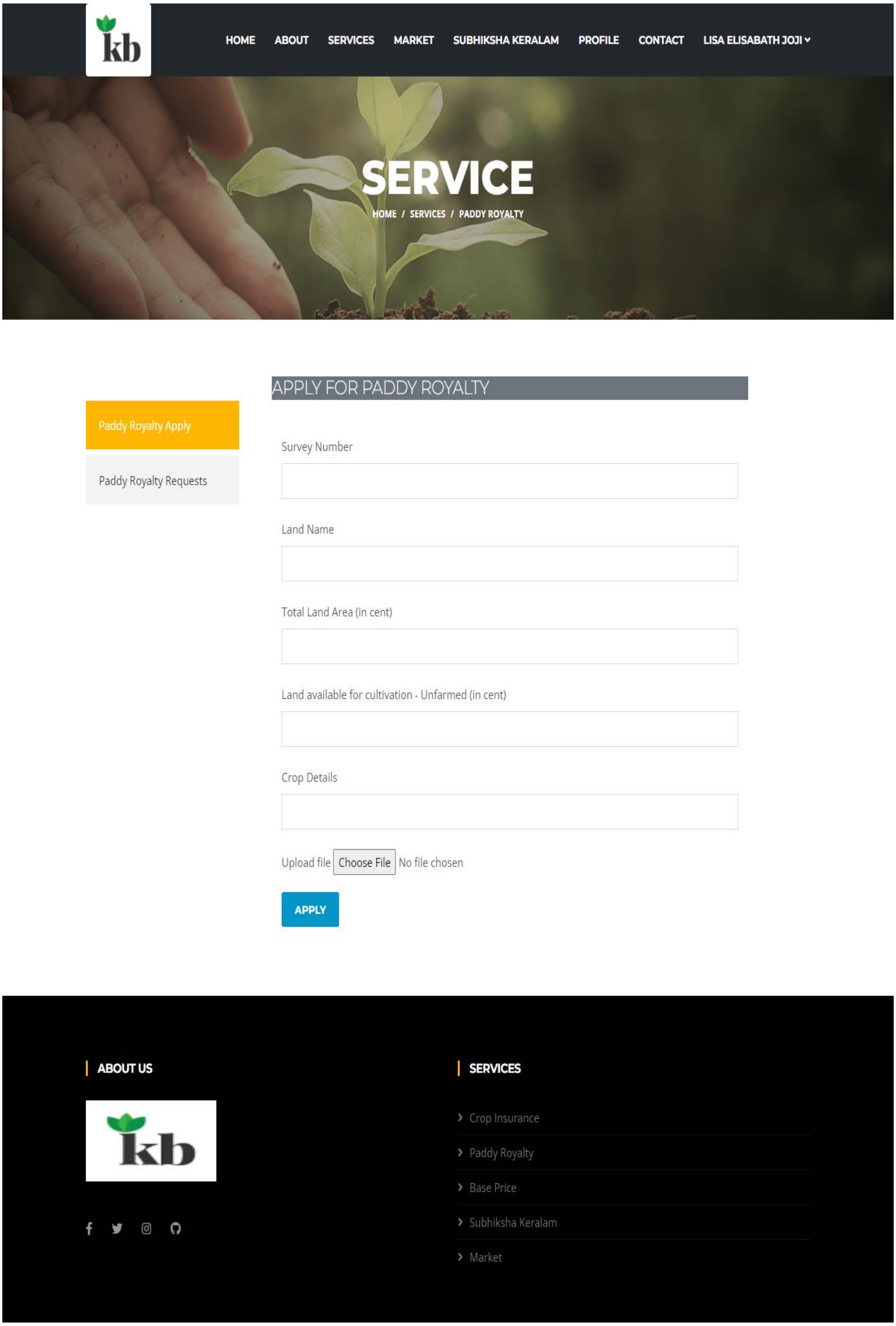
Crop Age (in months)

Upload file Choose File No file chosen

Account Number

IFSC Code

Apply for paddy royalty



APPLY FOR PADDY ROYALTY

Survey Number

Land Name

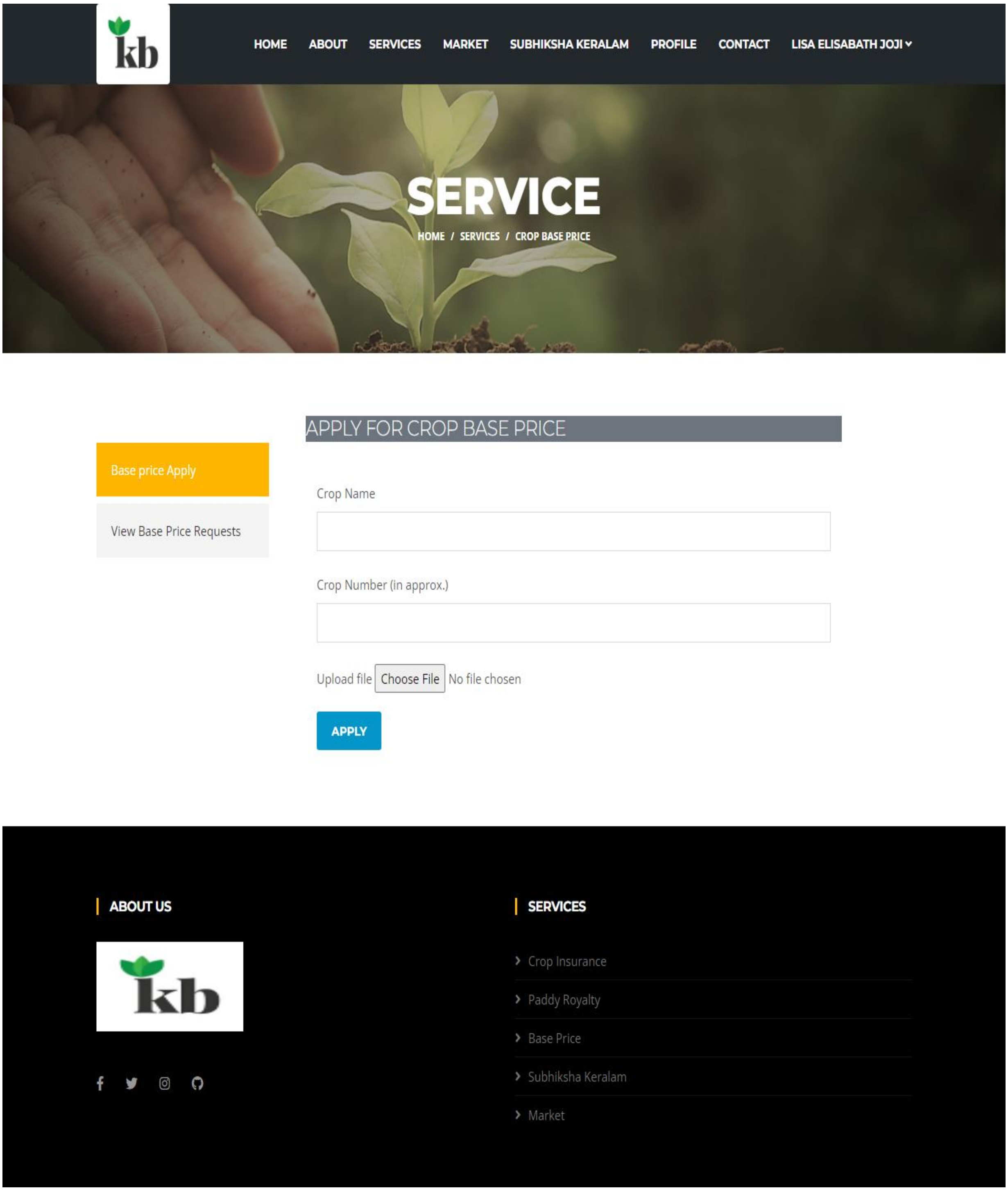
Total Land Area (in cent)

Land available for cultivation - Unfarmed (in cent)

Crop Details

Upload file Choose File No file chosen

Apply for crop base price



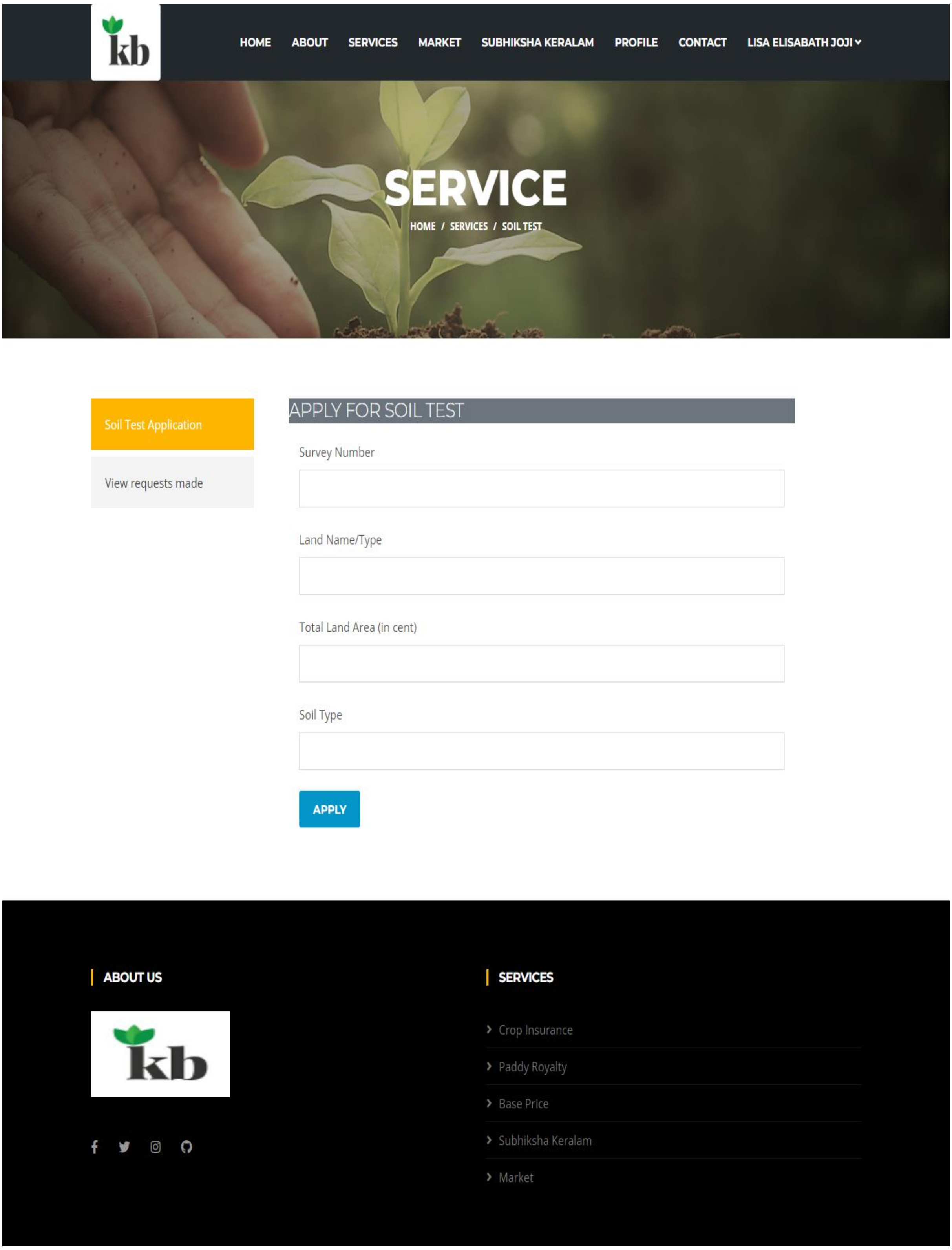
APPLY FOR CROP BASE PRICE

Crop Name

Crop Number (in approx.)

Upload file Choose File No file chosen

Apply for soil test



kb
HOME ABOUT SERVICES MARKET SUBHIKSHA KERALAM PROFILE CONTACT LISA ELISABATH JOJI
SERVICE
HOME / SERVICES / SOIL TEST
Soil Test Application
View requests made
APPLY

APPLY FOR SOIL TEST

Survey Number

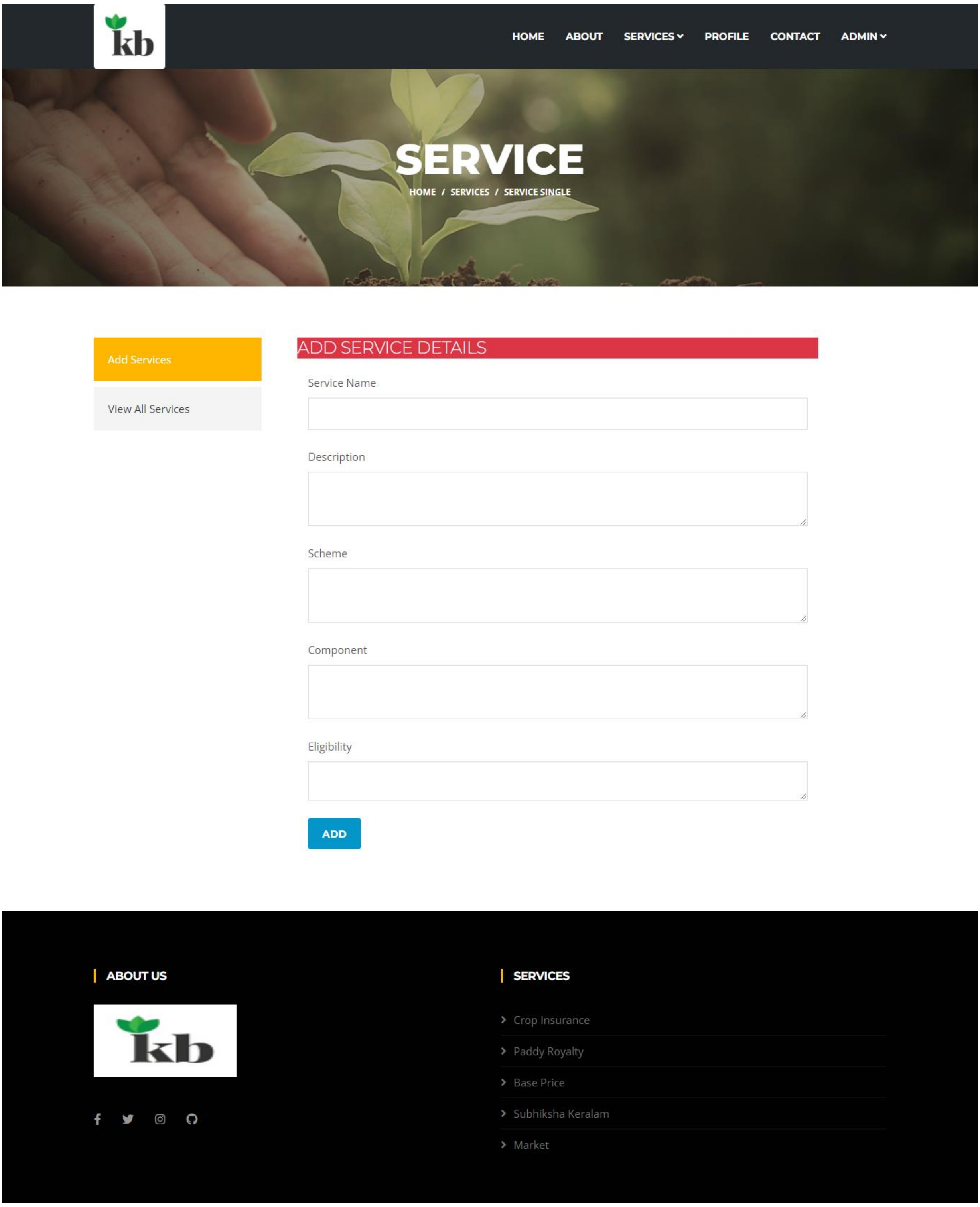
Land Name/Type

Total Land Area (in cent)

Soil Type

ADMIN PAGES

Add Services



ADD SERVICE DETAILS

Service Name

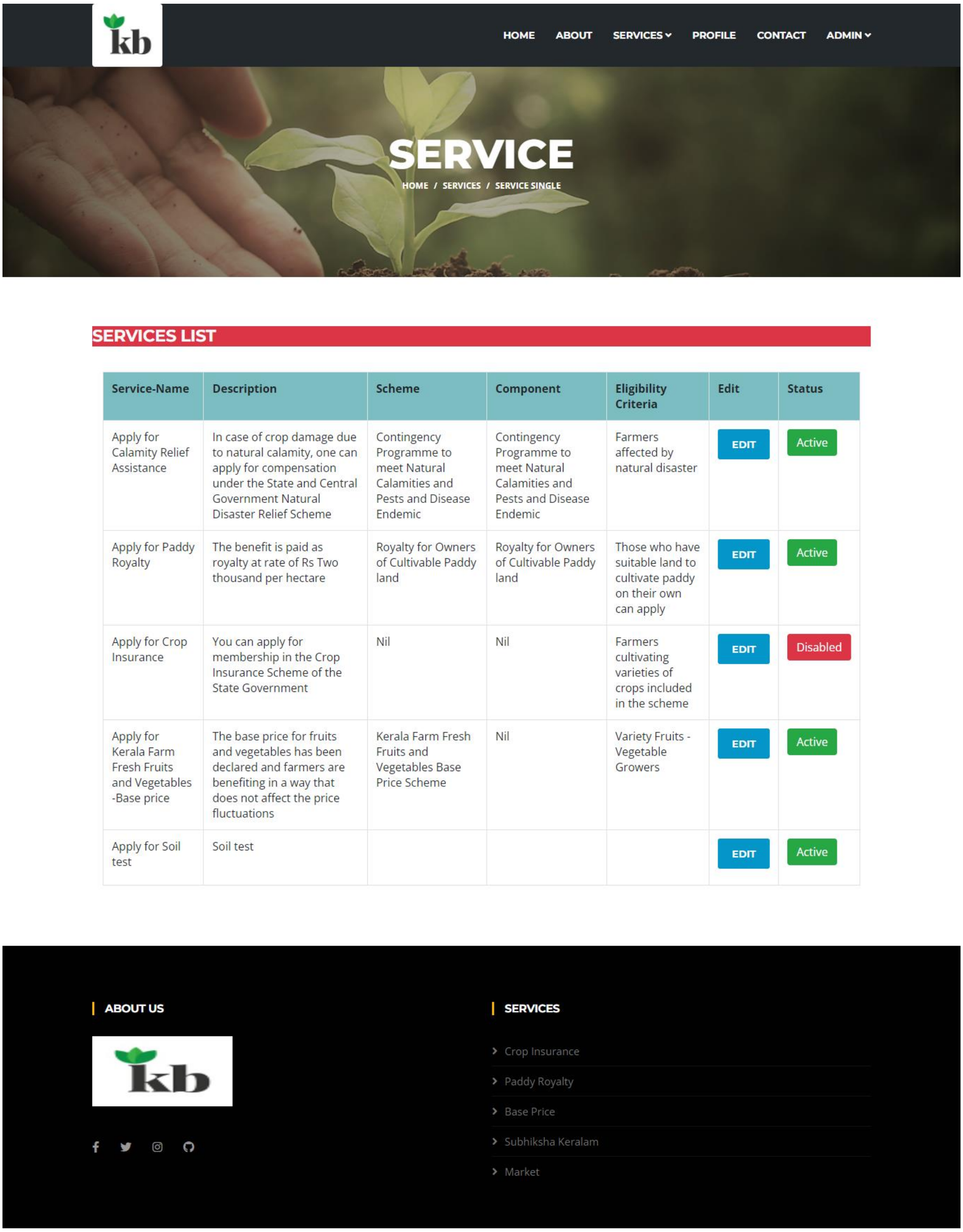
Description

Scheme

Component

Eligibility

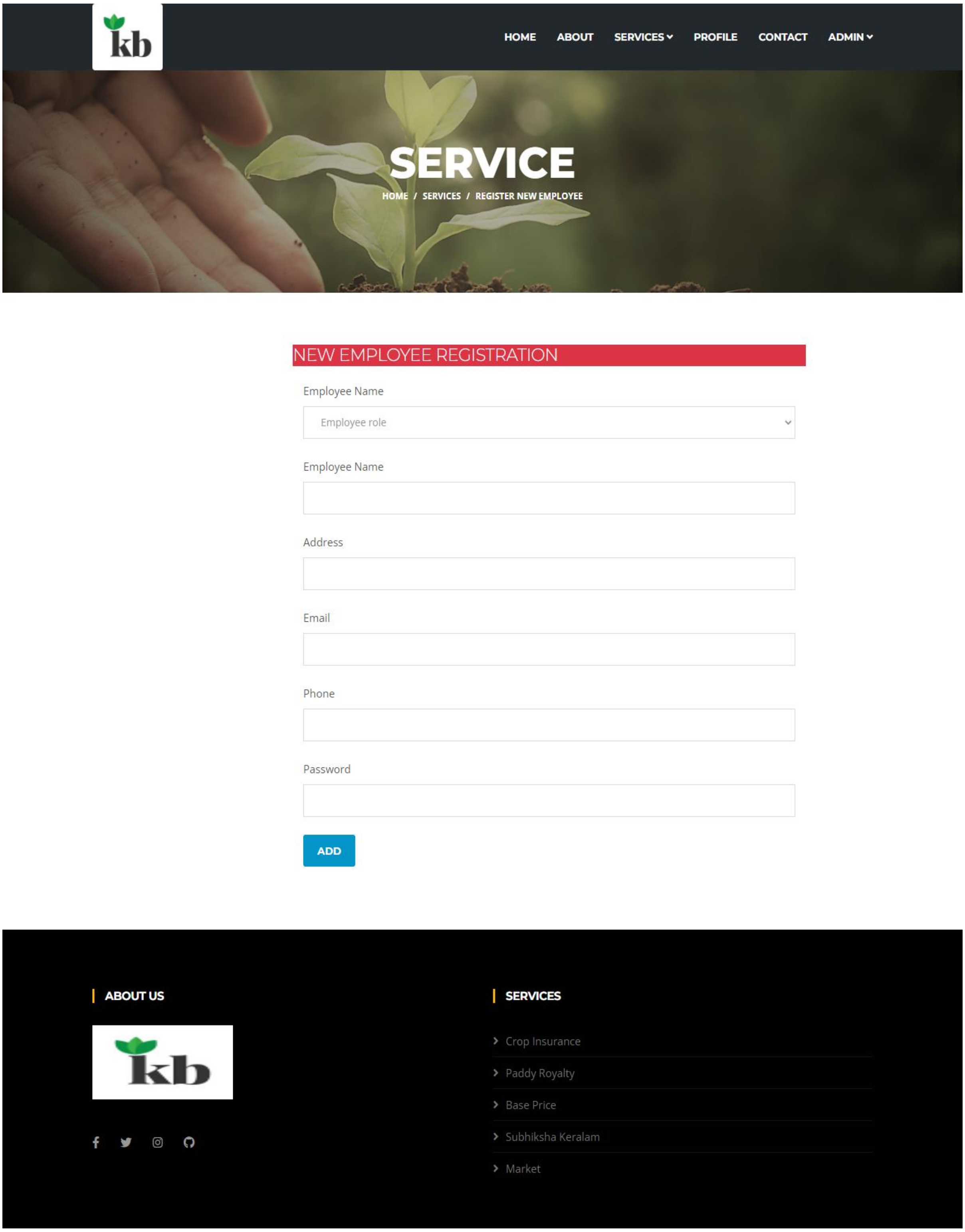
View Services



SERVICES LIST

Service-Name	Description	Scheme	Component	Eligibility Criteria	Edit	Status
Apply for Calamity Relief Assistance	In case of crop damage due to natural calamity, one can apply for compensation under the State and Central Government Natural Disaster Relief Scheme	Contingency Programme to meet Natural Calamities and Pests and Disease Endemic	Contingency Programme to meet Natural Calamities and Pests and Disease Endemic	Farmers affected by natural disaster	EDIT	Active
Apply for Paddy Royalty	The benefit is paid as royalty at rate of Rs Two thousand per hectare	Royalty for Owners of Cultivable Paddy land	Royalty for Owners of Cultivable Paddy land	Those who have suitable land to cultivate paddy on their own can apply	EDIT	Active
Apply for Crop Insurance	You can apply for membership in the Crop Insurance Scheme of the State Government	Nil	Nil	Farmers cultivating varieties of crops included in the scheme	EDIT	Disabled
Apply for Kerala Farm Fresh Fruits and Vegetables -Base price	The base price for fruits and vegetables has been declared and farmers are benefiting in a way that does not affect the price fluctuations	Kerala Farm Fresh Fruits and Vegetables Base Price Scheme	Nil	Variety Fruits - Vegetable Growers	EDIT	Active
Apply for Soil test	Soil test				EDIT	Active

Register New Employee



NEW EMPLOYEE REGISTRATION

Employee Name

Employee role

Employee Name

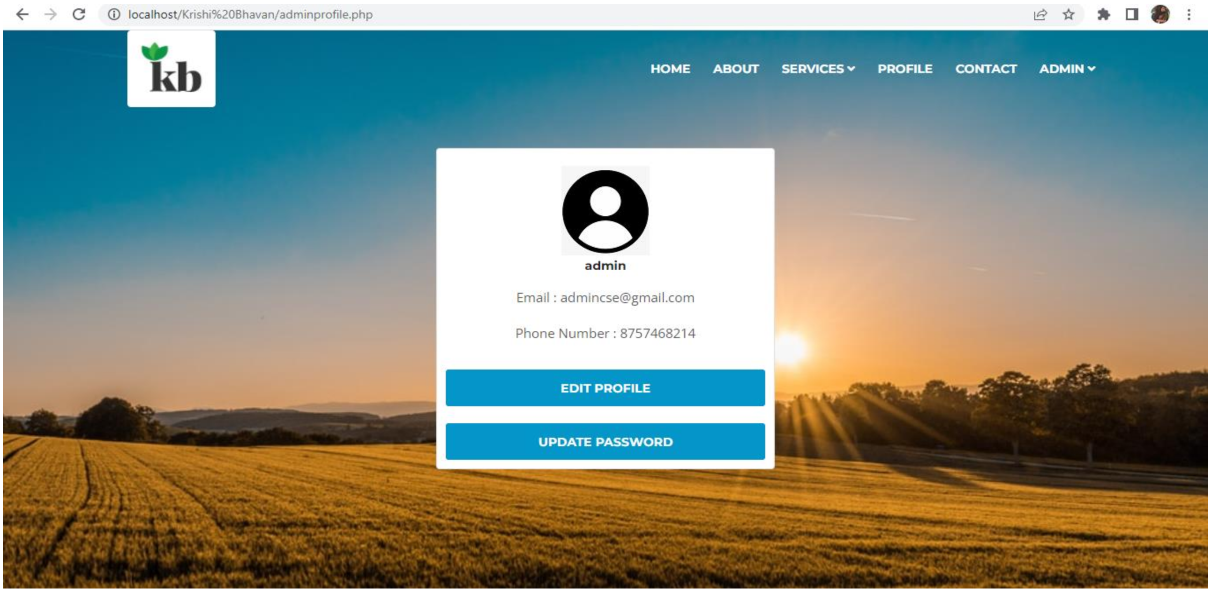
Address

Email

Phone

Password

Account Page



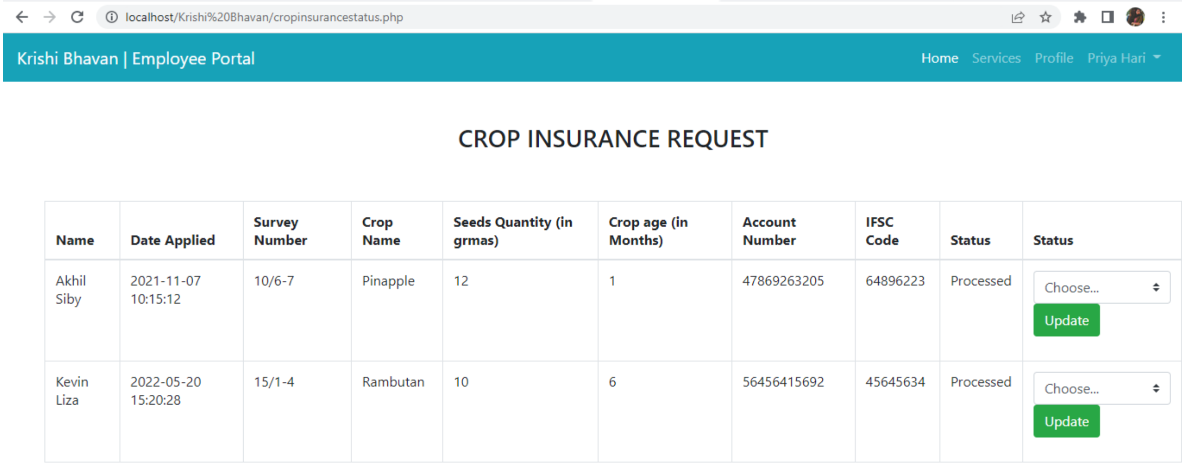
admin

Email : admincse@gmail.com

Phone Number : 8757468214

OFFICER PAGES

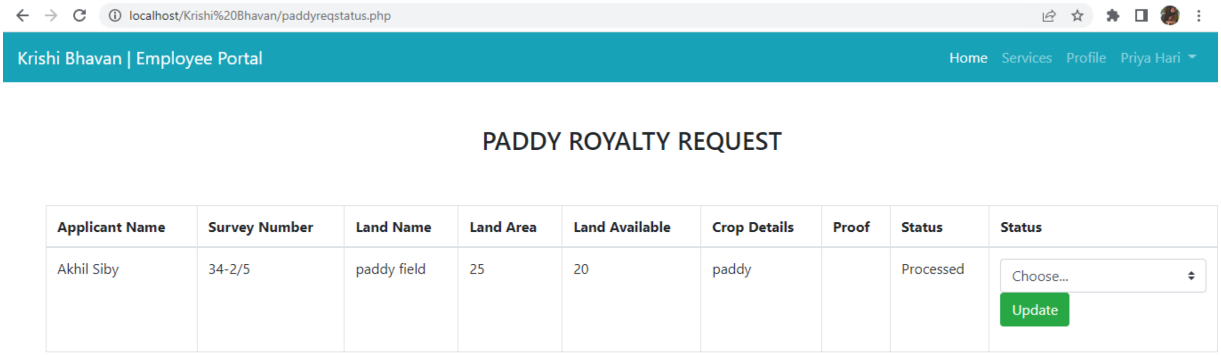
Crop Insurance Request


localhost/Krishi%20Bhavan/cropinsurancestatus.php
Krishi Bhavan | Employee Portal
Home
Services
Profile
Priya Hari

CROP INSURANCE REQUEST

Name	Date Applied	Survey Number	Crop Name	Seeds Quantity (in grmas)	Crop age (in Months)	Account Number	IFSC Code	Status	Status
Akhil Siby	2021-11-07 10:15:12	10/6-7	Pinapple	12	1	47869263205	64896223	Processed	Choose... Update
Kevin Liza	2022-05-20 15:20:28	15/1-4	Rambutan	10	6	56456415692	45645634	Processed	Choose... Update

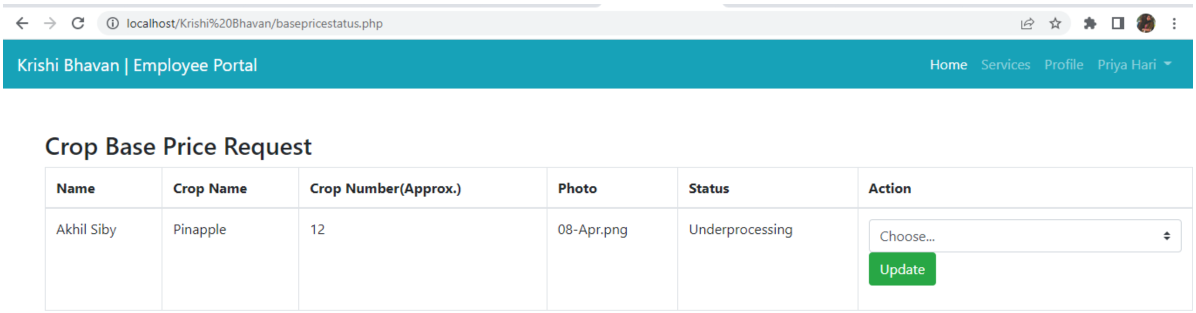
Paddy Royalty Request


localhost/Krishi%20Bhavan/paddyreqstatus.php
Krishi Bhavan | Employee Portal
Home
Services
Profile
Priya Hari
Choose...
Update

PADDY ROYALTY REQUEST

Applicant Name	Survey Number	Land Name	Land Area	Land Available	Crop Details	Proof	Status	Status
Akhil Siby	34-2/5	paddy field	25	20	paddy		Processed	Choose... Update

Crop Base Price Request

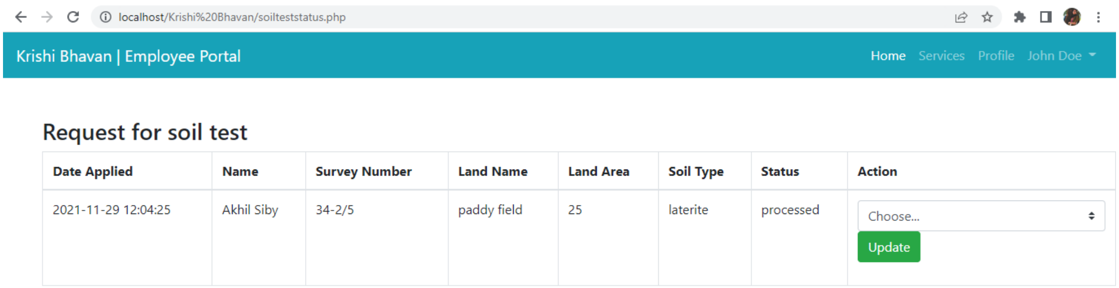

localhost/Krishi%20Bhavan/basepricestatus.php
Krishi Bhavan | Employee Portal
Home
Services
Profile
Priya Hari
Choose...
Update

Crop Base Price Request

Name	Crop Name	Crop Number(Approx.)	Photo	Status	Action
Akhil Siby	Pinapple	12	08-Apr.png	Underprocessing	Choose... Update

OFFICER PAGES

Soil Test Request


localhost/Krishi%20Bhavan/soilteststatus.php
Krishi Bhavan | Employee Portal
Home
Services
Profile
John Doe
Choose...
Update

Request for soil test

Date Applied	Name	Survey Number	Land Name	Land Area	Soil Type	Status	Action
2021-11-29 12:04:25	Akhil Siby	34-2/5	paddy field	25	laterite	processed	Choose... Update