AGRO-WORLD

Project Report Submitted By

HARITHA KRISHNAN

Reg. No: AJC20MCA-2040

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS (2 Year)
(MCA)
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]

2021-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**AGRO-WOLRD"** is the bonafide work of **HARITHA KRISHNAN (Reg.No: AJC20MCA-2040)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-2022.

Ms. Jetty Benjamin **Internal Guide**	**Ms. Nimmy Francis** **Coordinator**

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **“AGRO-WOLRD”** 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 Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

Date:	**HARITHA KRISHNAN**
KANJIRAPPALLY	**Reg.No : AJC20MCA-2040**

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 coordinators **Ms**. **Nimmy Francis** 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. Jetty Benjamin** 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.

HARITHA KRISHNAN

ABSTRACT

The main objective of this project for the better performance of krishibhavan service and to manage that services through computerized system. The concept of this project is that the farmer get a clear idea about what happened in krishibhavan office through online mode. This application helps users can view the schemes/services, with the eligibility criteria users can also apply the services/schemes. Office staff verify each user’s requests and monitor virtually. This application pass valid information to valid users. Users can easily access the information at very quick time.

CONTENT

List of Abbreviation

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

CHAPTER 1

INTRODUCTION

1. PROJECT OVERVIEW

The concept of this project is that the farmer get a clear idea about what happened in krishibhavan office through online mode. This department deals with formulation and implementation of both food crops and cash crops. The project is based on web application. Farmers can book eligible categories of product with in time.

The main objective of this project for the better performance of krishibhavan service and to manage that services through computerized system.

1.1 PROJECT SPECIFICATION

The proposed system is a website in which user can book online for servicing. It is cheaper and less time consuming for get information from the office.

The system mainly includes 5 modules. They are:

1. **Admin Module**

Admin must have a login into this system. He has the overall control of the system. Admin can add or update service and schemes details, manage user data etc. Admin can View all the registered users and also manage all that data.

- ✓ Login
- ✓ Create services/schemes
- ✓ Mange services /schemes
- ✓ Manage staff and officers
- ✓ Get details of users as and when required.

2.User Module

Customer can register and they can view services and apply for the services according to their needs and download required documents.

- ➢ Register:
- ➢ Users can get online form for register with details such as name, mobile number, id proof and land proof.
- ➢ Login
- ➢ user login to the site with their username and password
- ➢ View services and schemes
- ➢ users can view all the schemas with their eligible categories

3.Staff Module

- Login
- Staff login to the site with their username and password
- Manage users
- Registered users had been handled by staffs of the department.
- View application
- View which of the applications are newly arrived.
- View status
- Staff can be view each of the user’s current state.
- Process Application
- Staffs are evaluated each of the application by the eligibility and check that which of the applications are pending state.
- Update profile
- Updated staff’s profile details.

4.Officer Module

.

- Control over application
- Login
- Officers login to the site with their username and password
- View new request from user
- Officers can view the new requests created by the users.
- Approve application
- Each user application verified by the officers and matching the eligible categories to approve applications.
- Update status
- After approving application pending applications are counted.
- Approve for certificate
- If any certificate attest requests from the users are clarified by Officers.
- Generate report
- Audit
- Able to recommend new services to local government bodies
- Fielding
- Officers visit farmers field for the verification were their documents are valid or not.
- Manage staff
- Manage Records
- View status of staff
- Update profile

5.Payment Module

In payment system office provide different types of schemes, some of the scheme have payment and some of the scheme have not. Which is view by users.

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 wells 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 issue objected 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.

2.1 EXISTING SYSTEM

Existing system is not a fully automated system. User should directly reach to collect the information which they need by offline.

It take more time to get information for their appropriate needs.

2.2 DRAWBACKS OF EXISTING SYSTEM

- Human effort is needed.
- It is difficult to maintain important information in user.
- More manual hours need to process application.

2.3 PROPOSED SYSTEM

The proposed system is designed 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 Krishi Bhavan. In our proposed system. The issues of existing system can be overcome by the proposed system .It makes the process is more easier and people can know about the services through the application .This application keeps the data in a centralized way which is available to all the users simultaneously. No specific training is required for the staff to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. It is very easy to record the information of online services and schemes in the database.

2.4 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:

- **Better security:-**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

- **Ensure data accuracy:-**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

- **Better service: -**

The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

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 work ability, which is the impact on the agro-organization, ability to meet their farmers needs and effective use of farmer's resources. Thus, when a new application is proposed of agriculture 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 is cost estimates for appropriate land development, crop husbandry, harvesting, 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, AGRO-WORLD 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 suggest done?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. 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 for development. The System used was also of good performance of Processor Intel Pentium Gold; RAM 4GB and, Hard disk 500 GB

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.

3.2 SYSTEM SPECIFICATION

3.1.4 Hardware Specification

Processor	-	Intel Pentium Gold
RAM	-	4 GB
Hard disk	-	500 GB

3.1.5 Software Specification

Front End	-	HTML, CSS
Back end	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, J Query, PHP

3.2 SOFTWARE DESCRIPTION

3.2.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and2.1million web servers. Originally created by RasmusLedorfin1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page, it now stands for PHP: Hypertext Preprocessor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed and supported by Oracle Corporation. The MySQL Website 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 setup 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 in consistent, 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 versionsexist.Inthismanual,“SQL92”referstothestandardreleasedin1992, “SQL: 1999”refers to the standard released in1999,and“SQL:2003”referstothe current version of the standard. We use the phrase “the SQL standard ”to mean the current version of the SQL Standard at anytime.

- **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 down 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 back ends, 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 stand alone 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 processor a system insufficient 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 modeling of real-world objects and systems.

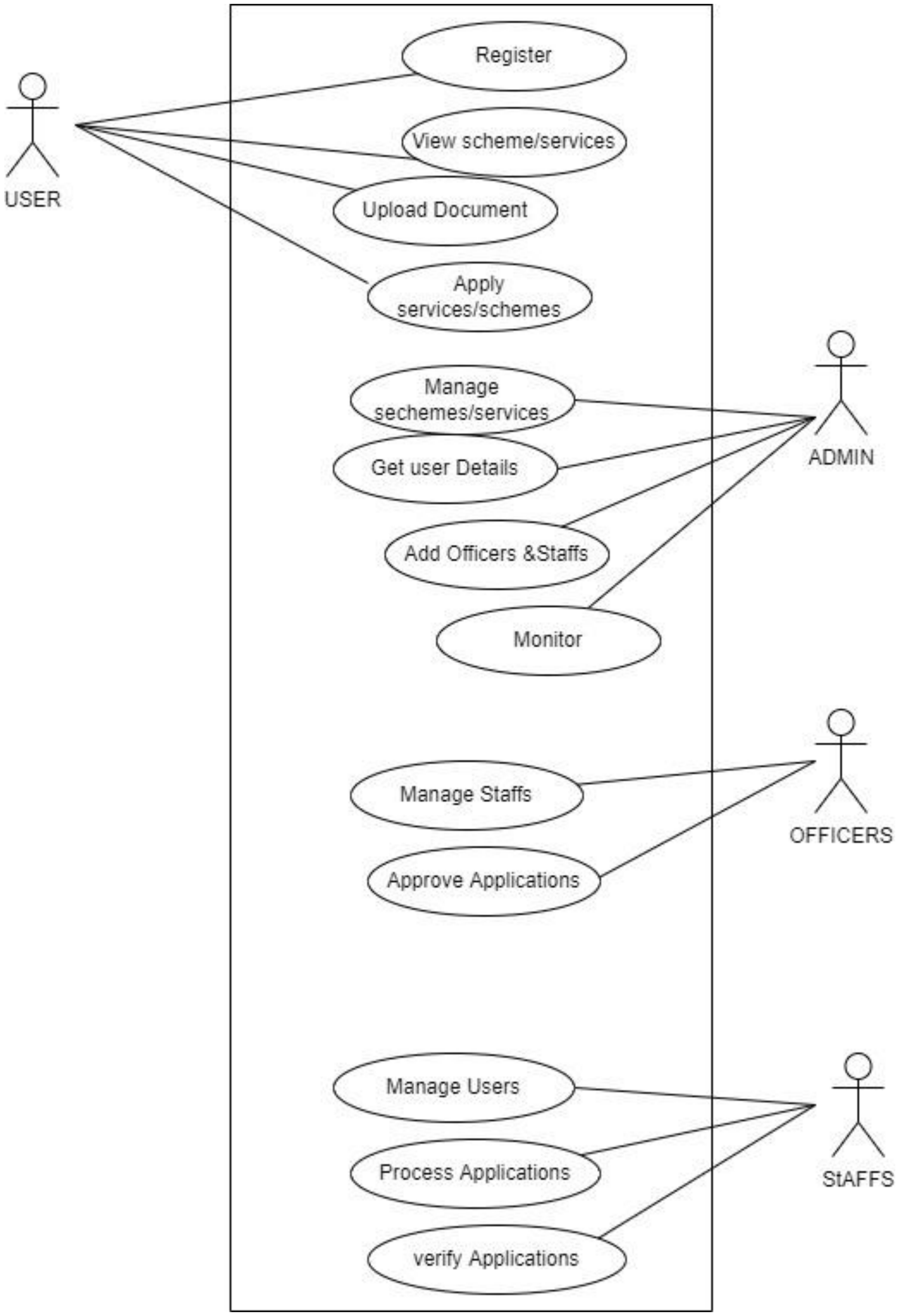
System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include 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.


Register
View scheme/services
Upload Document
Apply services/schemes
USER
Manage sechemes/services
Get user Details
Add Officers &Staffs
Monitor
ADMIN
Manage Staffs
Approve Applications
OFFICERS
Manage Users
Process Applications
verify Applications
StAFFS

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 –

- **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.
- **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

- **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.

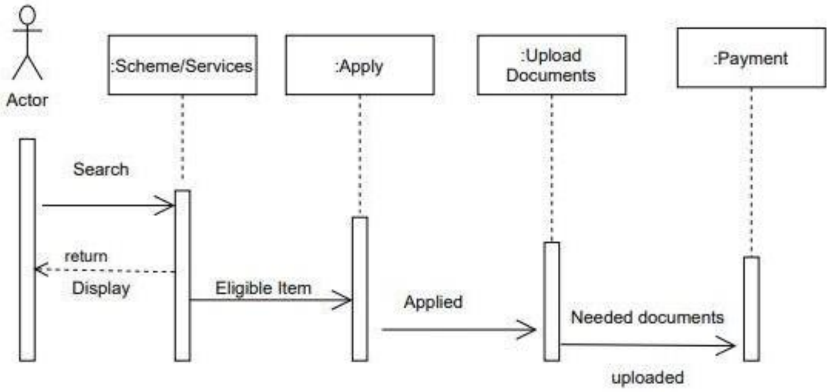
Messages can be broadly classified into the following categories:

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

- Self-Message
- Reply Message
- Found Message
- Lost Message
- **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.

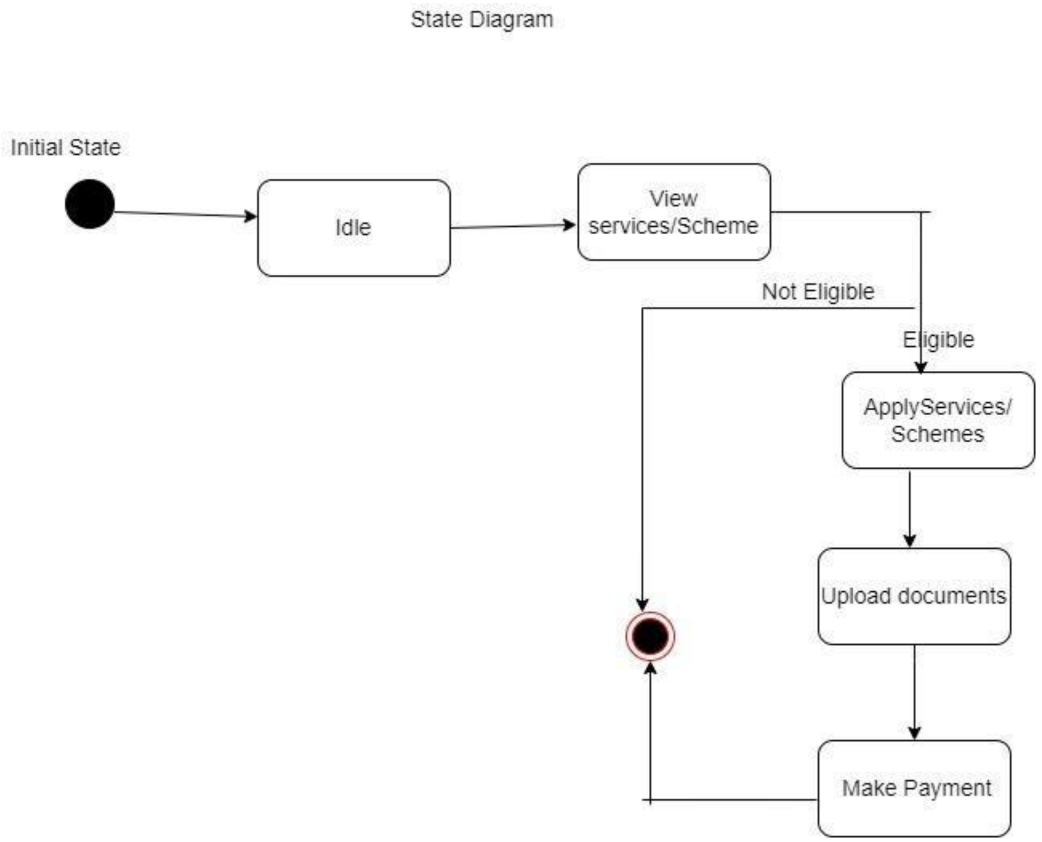
Uses of sequence diagrams –

- 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.



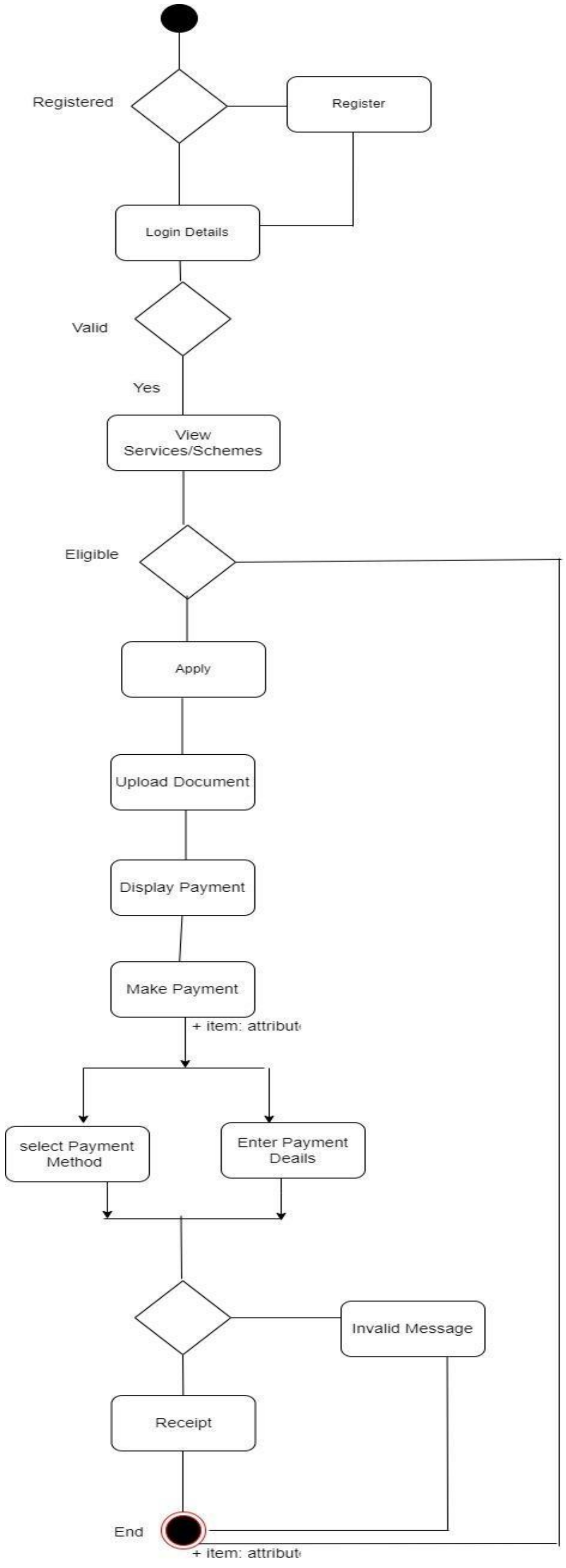
4.2.3. STATE CHART DIAGRAM

State Diagram are used to capture the behavior of a software system. UML State machine diagrams can be used to model the behavior of a class, a subsystem, a package, or even an entire system. It is also called a State chart or State Transition diagram. State chart diagrams provide us an efficient way to model the interactions or communication that occur within the external entities and a system. These diagrams are used to model the event-based system. A state of an object is controlled with the help of an event. State chart diagrams are used to describe various states of an entity within the application system.



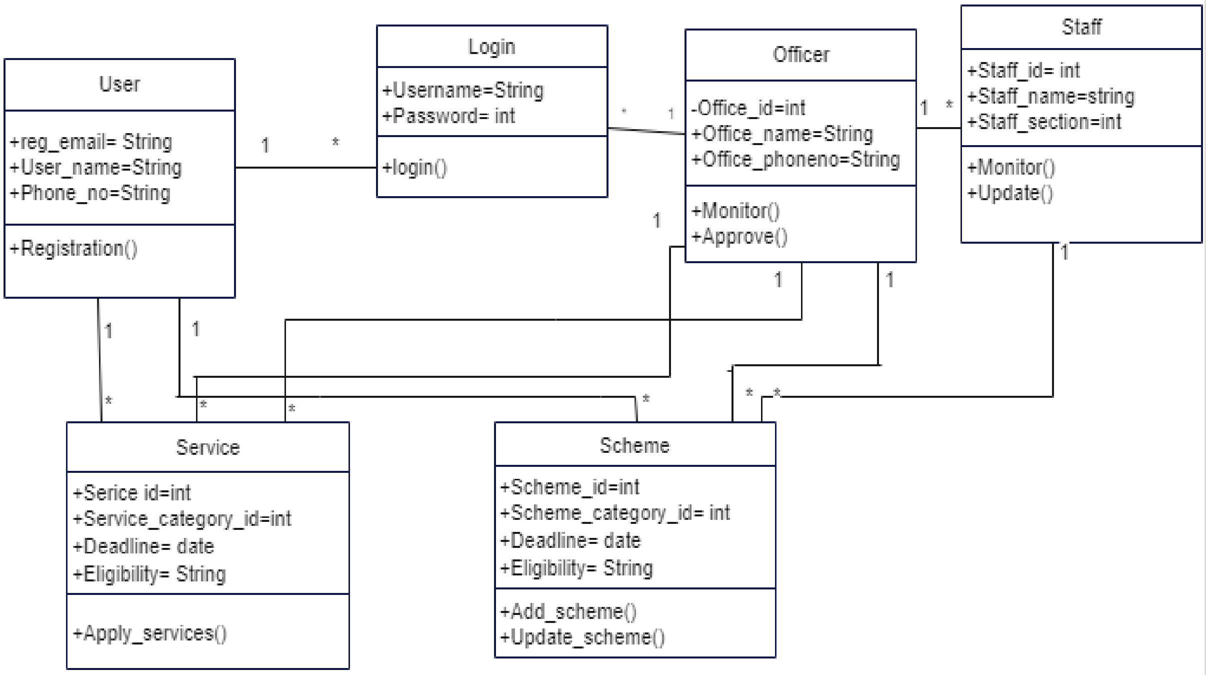
4.2.4. ACTIVITY DIAGRAM

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


Registered
Register
Login Details
Valid
Yes
View Services/Schemes
Eligible
Apply
Upload Document
Display Payment
Make Payment
+ item: attribut
select Payment Method
Enter Payment Deails
Invalid Message
Receipt
End
+ item: attribut

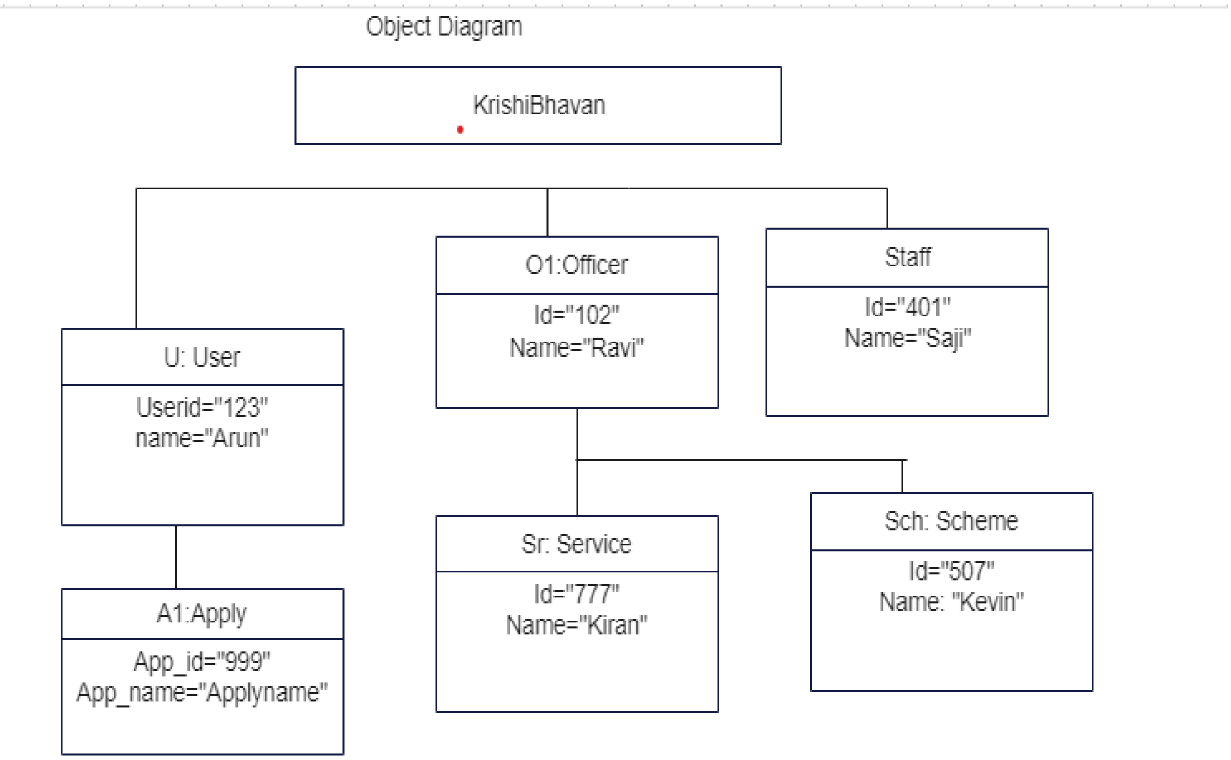
4.2.5. CLASS DIAGRAM

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 shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.



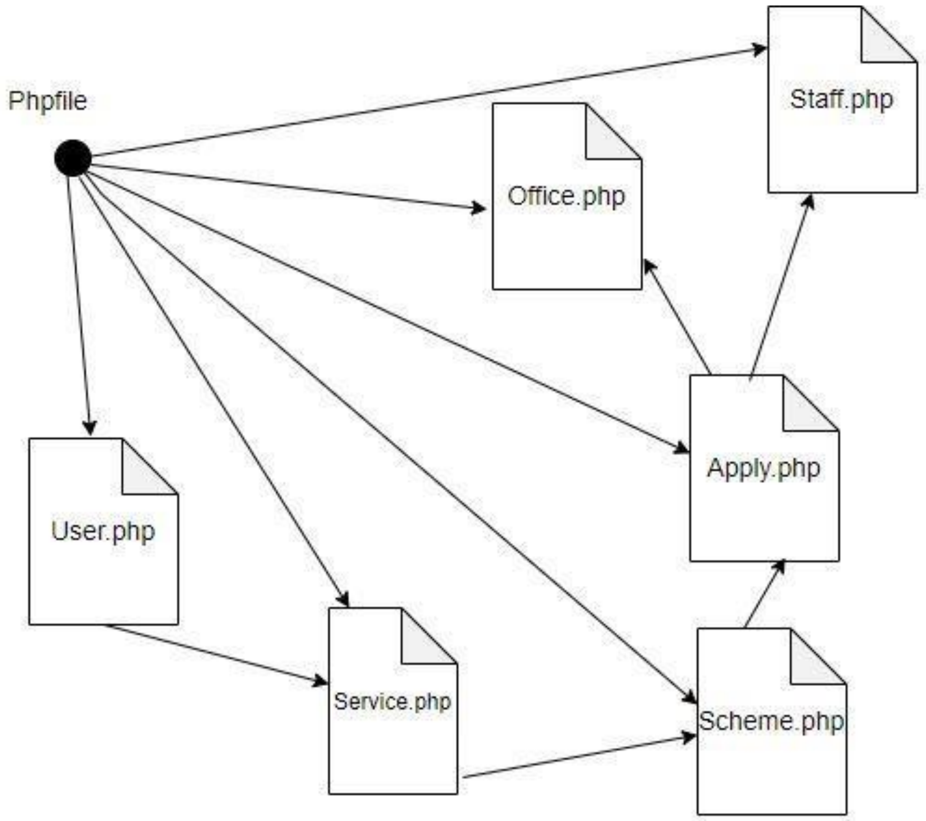
4.2.6. OBJECT DIAGRAM

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.



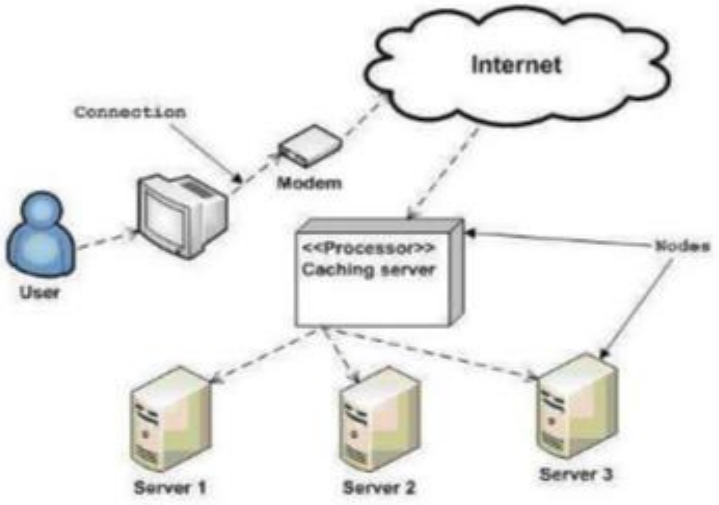
4.2.7. COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Physical aspects are the elements such as executable, libraries, files, documents, etc. which reside in a node. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.



4.2.8 DEPLOYMENT DIGRAM

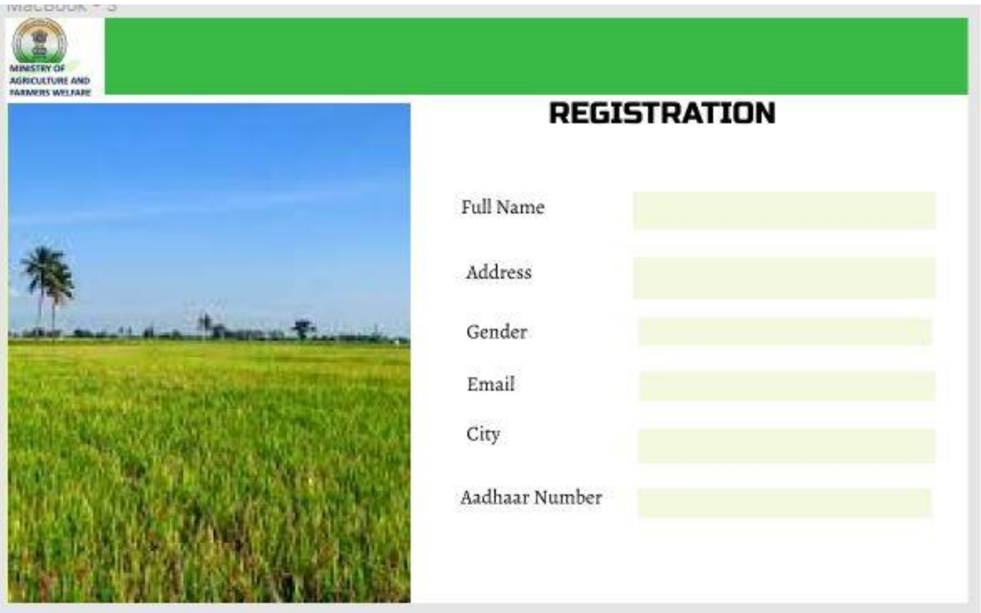
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.



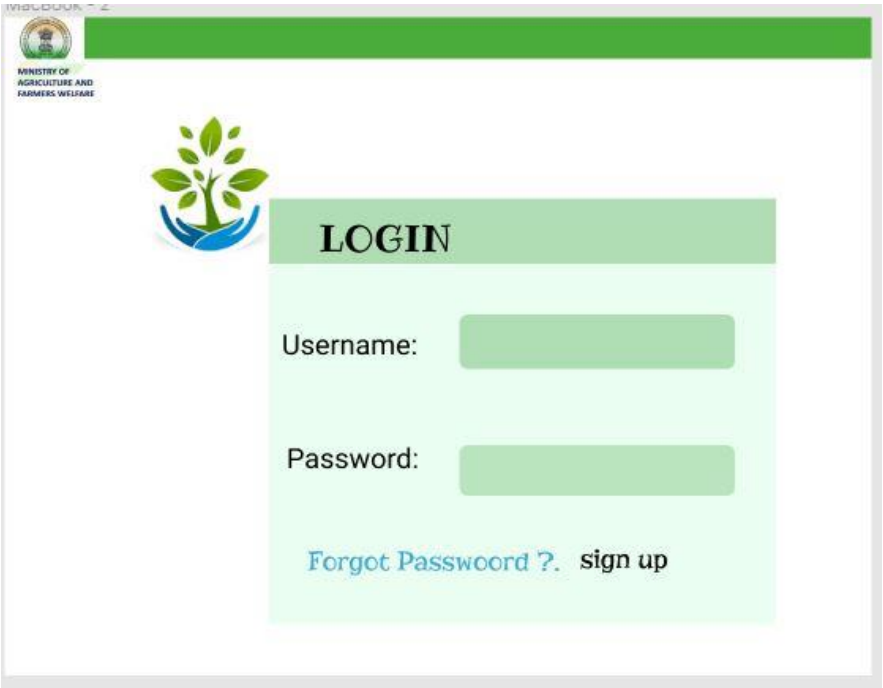
4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

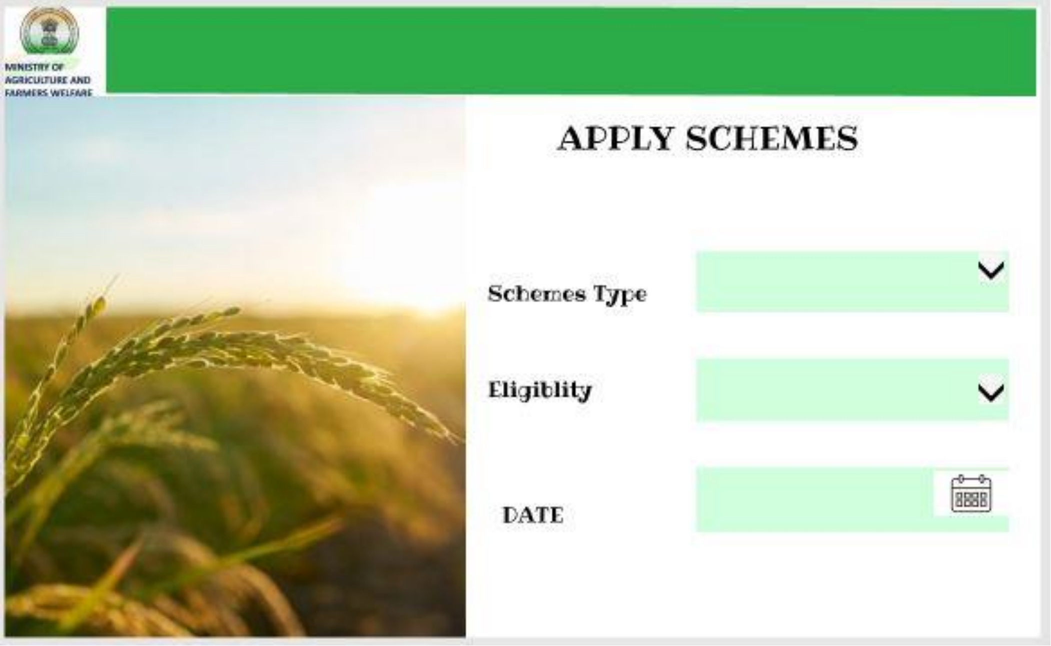
Form Name : User Registration



Form Name : User Login

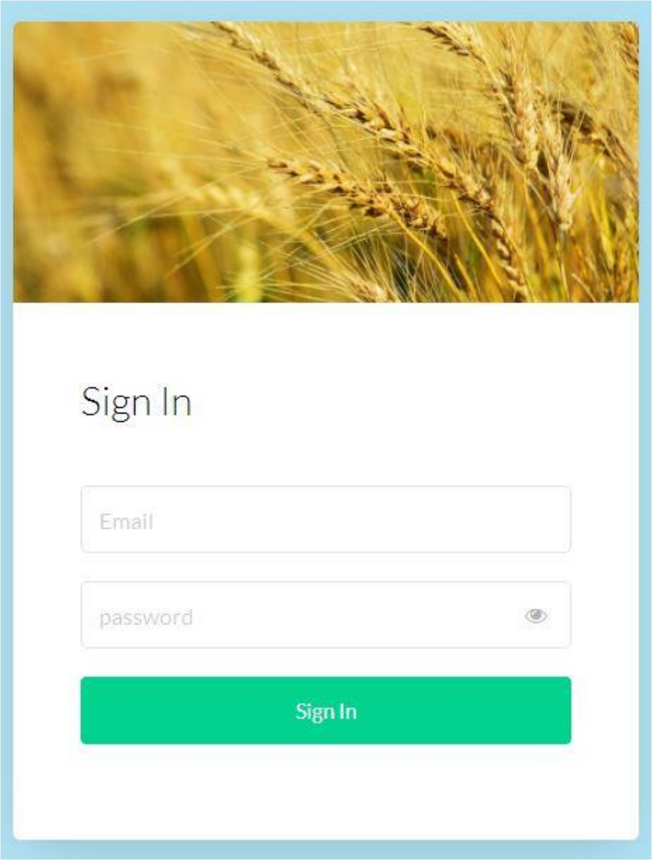


Form Name : Apply Schemes

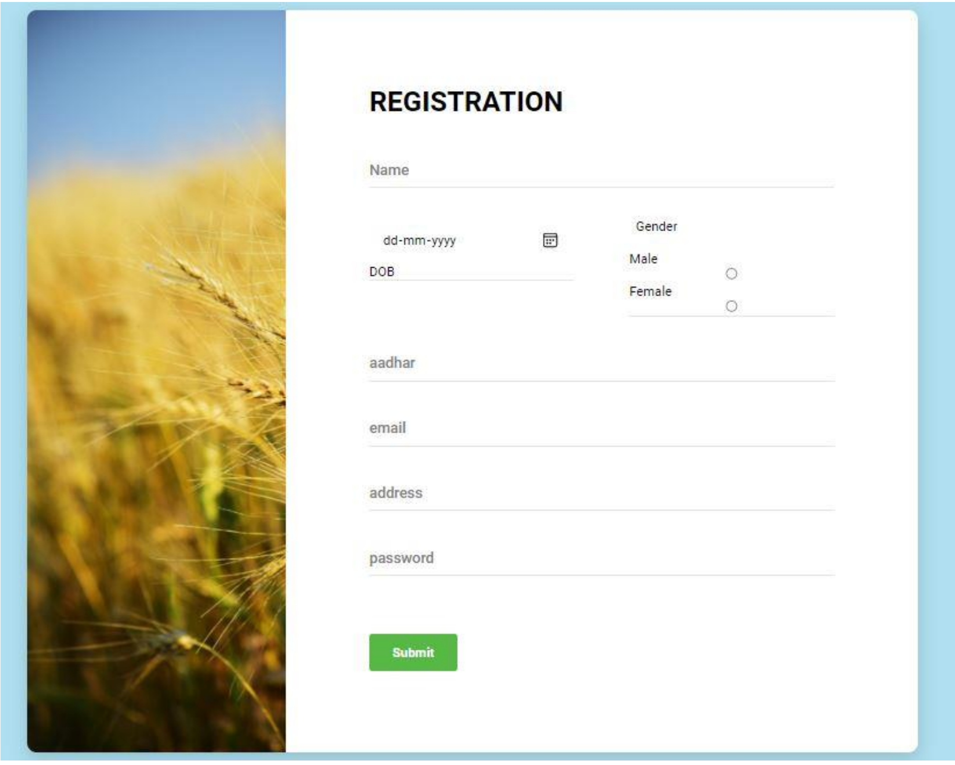


4.3.2 OUTPUT DESIGN

User Login



User Registration



REGISTRATION

Name

dd-mm-yyyy

DOB

Gender

Male ○

Female ○

aadhar

email

address

password

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 data base 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 are 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 as attribute 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 attributes 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

tbl_Login
Primary key : login_id
Foreignkey : reg_email

Filedname	Datatype	Size	Description
Login_id	int	10	Primary key of Tbl_Login
Reg_email	varchar	15	Foreign key Tbl_registration
Password	varchar	12	Store password

tbl_Registration
Primary key : reg_email
Foreign key : UserType_id

Fieldname	Datatype	Size	Description
Reg_email	Varchar	15	Primary key of Tbl_Registration
userType_id	Int	10	Foreign key of Tbl_UserType
Fname	Varchar	25	First name
Lname	Varchar	25	Last name
Mobile_no	Varchar	15	Moblie number
Aadhaar_proof	Varchar	30	Aadhaar proof
Land_proof	Varchar	30	Land proof

tbl_scheme

Primary key: Scheme_id
Foreign key : Sche_catid

Fieldname	Datatype	size	Description
Scheme_id	int	20	Primary key
Sch_deadline	date		End date of scheme for apply
Sch_description	varchar	30	Description of schemes

tbl_Service

Primary key: Service_id
Foreign key: Ser_catid

Filedname	Datatype	Size	Description
Service_id	int	20	Primary key of Tbl_service
Ser_deadline	date		End date of service
Ser_Description	varchar	30	Description for services

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 appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are 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 is 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 objective so testing should be stated in measurable terms. So that theme any 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 setup 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 where removed and ensured that all modules are working, and gives the expected result.

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 in consistence is 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 will fully 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.

5.2.4 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:

5.2.3.1 Input Screen Designs,

5.2.3.2 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 understudy 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 the process of converting a new revised system design into an operation alone. At this stage the main workload, the greatest up heavy land 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 are liable 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 system becomes 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. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet. In the existing system farmers visit krishi-office to get the information, they had not get any proper notification about the schemes/services are available. Users are not fully satisfied the procedures of krishi-office because user interaction is very low compare to online mode. But in the case of proposed system that introduces user view all information about the scheme/services. Apply the services/schemes by their wish. The main objective of this project for the better performance of krishibhavan service and to manage that services through computerized system. Users can provide any feedback via to the application.

FUTURE SCOPE

- Future Scope of this project is introducing create virtual guideline classes to the farmers.
- Generate a payment system that help users will do all their payment transaction via this application.
- Create a feedback facility to the users.
- Generate notification system.

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 DesignDescriptions.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample Code

Login.php

```
<?php


include 'connection.php';

if(isset($_POST['email']) && isset($_POST['email'])){
   $email=$_POST['email'];
   $password=$_POST['password'];

   $sql="SELECT * FROM tbl_login where email ='$email' AND password = '$password'";
   $result =mysqli_query($conn,$sql);

   if(mysqli_num_rows($result) === 1){
               $row = mysqli_fetch_assoc($result);
               if ($row['email'] == $email && $row['password'] == $password) {
                   echo "Successfully logged in";
                   exit();
          }
          else{
                     echo "Incorrect Email and Password";
                     exit();
          }
       }
}


?>
```

Addscheme.php

```
<?php
include 'connection.php';
if(isset($_POST['submit'])){

$name=$_POST['name'];
$schemedescri=$_POST['schemedescription'];
$startdate=$_POST['startdate'];
$enddate=$_POST['enddate'];
$document=$_POST['document'];
$quali=$_POST['qualification'];

$sql=mysqli_query($conn,"INSERT INTO `scheme`( `name`, `schemedescription`, `startdate`,
`enddate`, `document`, `qualification`) VALUES
('$name','$schemedescri','$startdate','$enddate','$document','$quali')");
echo '<script>alert("Successfully Inserted")</script>';
header('Addsheme.php');
}
?>


<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>AgroWorld:Dashboard</title>
    <!-- plugins:css -->
    <link rel="stylesheet" href="assets/vendors/mdi/css/materialdesignicons.min.css">
    <link rel="stylesheet" href="assets/vendors/css/vendor.bundle.base.css">
    <!-- endinject -->
    <!-- Plugin css for this page -->
    <link rel="stylesheet" href="assets/vendors/jvectormap/jquery-jvectormap.css">
    <link rel="stylesheet" href="assets/vendors/flag-icon-css/css/flag-icon.min.css">
    <link rel="stylesheet" href="assets/vendors/owl-carousel-2/owl.carousel.min.css">
    <link rel="stylesheet" href="assets/vendors/owl-carousel-2/owl.theme.default.min.css">
    <!-- End plugin css for this page -->
    <!-- inject:css -->
    <!-- endinject -->
    <!-- Layout styles -->
    <link rel="stylesheet" href="assets/css/style.css">
    <!-- End layout styles -->
    <link rel="shortcut icon" href="assets/images/favicon.png" />
  </head>
  <body>
    <div class="container-scroller">
      <!-- partial:partials/_sidebar.html -->
      <nav class="sidebar sidebar-offcanvas" id="sidebar">
        <div class="sidebar-brand-wrapper d-none d-lg-flex align-items-center justify-content-center fixed-top">
          <a class="sidebar-brand brand-logo" href="index.html"><img src="assets/images/logo.svg" alt="logo" /></a>
          <a class="sidebar-brand brand-logo-mini" href="index.html"><img src="assets/images/logo-mini.svg" alt="logo" /></a>
        </div>
        <ul class="nav">
          <li class="nav-item profile">
            <div class="profile-desc">
              <div class="profile-pic">
                <div class="count-indicator">
                  <img class="img-xs rounded-circle " src="assets/images/faces/face15.jpg" alt="">
                  <span class="count bg-success"></span>
                </div>
                <div class="profile-name">
                  <h5 class="mb-0 font-weight-normal">Admin</h5>

                </div>
              </div>
              <a href="#" id="profile-dropdown" data-toggle="dropdown"><i class="mdi mdi-dots-vertical"></i></a>
              <div class="dropdown-menu dropdown-menu-right sidebar-dropdown preview-list" aria-
```

```
labelledby="profile-dropdown">


              <div class="dropdown-divider"></div>
              <a href="#" class="dropdown-item preview-item">
                <div class="preview-thumbnail">
                  <div class="preview-icon bg-dark rounded-circle">
                    <i class="mdi mdi-calendar-today text-success"></i>
                  </div>
                </div>
                <div class="preview-item-content">
                  <p class="preview-subject ellipsis mb-1 text-small">To-do list</p>
                </div>
              </a>
            </div>
          </div>
        </li>

        <li class="nav-item menu-items">
          <a class="nav-link" href="index.html">
            <span class="menu-icon">
              <i class="mdi mdi-speedometer"></i>
            </span>
            <span class="menu-title">Dashboard</span>
          </a>
        </li>
      </ul>
    </nav>
    <!-- partial -->
    <div class="container-fluid page-body-wrapper">
      <!-- partial:partials/_navbar.html -->
      <nav class="navbar p-0 fixed-top d-flex flex-row">
        <div class="navbar-brand-wrapper d-flex d-lg-none align-items-center justify-content-center">
          <a class="navbar-brand brand-logo-mini" href="index.html"><img src="assets/images/logo-
mini.svg" alt="logo" /></a>
        </div>
        <div class="navbar-menu-wrapper flex-grow d-flex align-items-stretch">
          <button class="navbar-toggler navbar-toggler align-self-center" type="button" data-
toggle="minimize">
            <span class="mdi mdi-menu"></span>
          </button>

          <ul class="navbar-nav navbar-nav-right">

            <li class="nav-item dropdown">
              <a class="nav-link" id="profileDropdown" href="#" data-toggle="dropdown">
                <div class="navbar-profile">
                  <img class="img-xs rounded-circle" src="assets/images/faces/face15.jpg" alt="">
                  <p class="mb-0 d-none d-sm-block navbar-profile-name">Admin</p>
                  <i class="mdi mdi-menu-down d-none d-sm-block"></i>
                </div>
              </a>
              <div class="dropdown-menu dropdown-menu-right navbar-dropdown preview-list" aria-
labelledby="profileDropdown">
```

```
                <div class="dropdown-divider"></div>
                <a class="dropdown-item preview-item">
                  <div class="preview-thumbnail">
                    <div class="preview-icon bg-dark rounded-circle">
                      <i class="mdi mdi-logout text-danger"></i>
                    </div>
                  </div>
                  <div class="preview-item-content">
                    <p class="preview-subject mb-1">Log out</p>
                  </div>
                </a>
                </div>
            </li>
          </ul>
          <button class="navbar-toggler navbar-toggler-right d-lg-none align-self-center" type="button"
data-toggle="offcanvas">
            <span class="mdi mdi-format-line-spacing"></span>
          </button>
        </div>
      </nav>

                  <div class="main-panel">
        <div class="content-wrapper">

          <div class="row">
            <div class="col-12 grid-margin stretch-card">
              <div class="card">
                <div class="card-body">
                  <h4 class="card-title">Manage Schemes</h4>
                  <p class="card-description">Add Schemes</p>
                  <form class="forms-sample" action="" method="POST">
                    <div class="form-group">
                      <label for="name"> Scheme Name</label>
                      <input type="text" class="form-control" name="name" id="name"
placeholder="Scheme Name" required  onchange="Validate();">

              </div>
              <span id="msg1" style="color:red;"></span>
<script>
function Validate()
{
    var val = document.getElementById('name').value;

    if (!val.match(/^[A-Z] [A-Za-z]{3,}$/))
    {
        document.getElementById('msg1').innerHTML="Start with a Capital letter & Only alphabets
without space are allowed!!";
                         document.getElementById('name').value = "";
        return false;
    }
document.getElementById('msg1').innerHTML=" ";
    return true;
}
</script>
```

```
                    <div class="form-group">
                      <label for="schemedescription">Scheme Description</label>
                                              <input type="text" class="form-control" name="schemedescription"
id="schemedescription" placeholder="Scheme Description" required onchange="Validate();">


                                         </div>

                    <div class="form-group">
                      <label for="startdate">Start Date</label>
                      <input type="date" class="form-control" name="startdate" id="startdate"
placeholder="Start Date">
                    </div>
                                                   <span id="msg1" style="color:red;"></span>
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script type="text/javascript">
    $(function () {
        var today = new Date();
        var month = ('0' + (today.getMonth() + 1)).slice(-2);
        var day = ('0' + today.getDate()).slice(-2);
        var year = today.getFullYear();
        var date = year + '-' + month + '-' + day;
        $('[id*=date]').attr('min', date);
    });
</script>

                                                   <div class="form-group">
                      <label for="enddate">End Date</label>
                      <input type="date" class="form-control"  name ="enddate" id="enddate"
placeholder="End Date">
                    </div>
                                                   <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script type="text/javascript">
    $(function () {
        var today = new Date();
        var month = ('0' + (today.getMonth() + 1)).slice(-2);
        var day = ('0' + today.getDate()).slice(-2);
        var year = today.getFullYear();
        var date = year + '-' + month + '-' + day;
        $('[id*=date]').attr('min', date);
    });
</script>

                   <!-- <div class="form-group">
                      <label>File upload</label>
                      <input type="file" name="img[]" class="file-upload-default">
                      <div class="input-group col-xs-12">
                        <input type="text" class="form-control file-upload-info" disabled
placeholder="Upload Image">
                        <span class="input-group-append">
                          <button class="file-upload-browse btn btn-primary" type="button">Upload</button>
                        </span>
```

```
                  </div>-->
                 </div>

                                    <div class="form-group">
                  <label for="document">Document required</label>
                  <input type="file" class="form-control" name="document" id="document"
placeholder="Document required" required>

 </div>

                 <div class="form-group">
                  <label for="qualification">Qualification</label>
                  <textarea class="form-control" name="qualification" id="qualification" required
rows="4"></textarea>
                 </div>
                 <button type="submit" name="submit" class="btn btn-primary mr-2">Submit</button>

                </form>
               </div>
              </div>






             </div>
            </div>
           </div>

          <!-- partial -->

           <!-- content-wrapper ends -->
           <!-- partial:partials/_footer.html -->
           <footer class="footer">

           </footer>
           <!-- partial -->
          </div>
          <!-- main-panel ends -->
         </div>
         <!-- page-body-wrapper ends -->
        </div>
        <!-- container-scroller -->
        <!-- plugins:js -->
        <script src="assets/vendors/js/vendor.bundle.base.js"></script>
        <!-- endinject -->
        <!-- Plugin js for this page -->
        <script src="assets/vendors/chart.js/Chart.min.js"></script>
        <script src="assets/vendors/progressbar.js/progressbar.min.js"></script>
        <script src="assets/vendors/jvectormap/jquery-jvectormap.min.js"></script>
        <script src="assets/vendors/jvectormap/jquery-jvectormap-world-mill-en.js"></script>
        <script src="assets/vendors/owl-carousel-2/owl.carousel.min.js"></script>
        <!-- End plugin js for this page -->
        <!-- inject:js -->
```

```
    <script src="assets/js/off-canvas.js"></script>
    <script src="assets/js/hoverable-collapse.js"></script>
    <script src="assets/js/misc.js"></script>
    <script src="assets/js/settings.js"></script>
    <script src="assets/js/todolist.js"></script>
    <!-- endinject -->
    <!-- Custom js for this page -->
    <script src="assets/js/dashboard.js"></script>
    <!-- End custom js for this page -->
  </body>
</html>
```

Addservice.php

```
<?php
include './connection.php';
if(isset($_POST['submit'])){

$name=$_POST['name'];
$servicedescription=$_POST['servicedescription'];
$startdate=$_POST['startdate'];
$enddate=$_POST['enddate'];
$document=$_POST['document'];
$qualification = $_POST['qualification'];



$sql=mysqli_query($conn,"INSERT INTO
        tbl_service(name,servicedescription,startdate,enddate,document,qualification) VALUES
        ('$name','$servicedescription','$startdate','$enddate','$document','$qualification')");




echo '<script>alert("Successfully Inserted")</script>';
header('AddServices.php');
}
?>



<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>AgroWorld:Dashboard</title>
  <!-- plugins:css -->
  <link rel="stylesheet" href="assets/vendors/mdi/css/materialdesignicons.min.css">
  <link rel="stylesheet" href="assets/vendors/css/vendor.bundle.base.css">
  <!-- endinject -->
  <!-- Plugin css for this page -->
  <link rel="stylesheet" href="assets/vendors/jvectormap/jquery-jvectormap.css">
  <link rel="stylesheet" href="assets/vendors/flag-icon-css/css/flag-icon.min.css">
  <link rel="stylesheet" href="assets/vendors/owl-carousel-2/owl.carousel.min.css">
  <link rel="stylesheet" href="assets/vendors/owl-carousel-2/owl.theme.default.min.css">
  <!-- End plugin css for this page -->
  <!-- inject:css -->
  <!-- endinject -->
  <!-- Layout styles -->
  <link rel="stylesheet" href="assets/css/style.css">
  <!-- End layout styles -->
  <link rel="shortcut icon" href="assets/images/favicon.png" />
</head>
<body>
  <div class="container-scroller">
    <!-- partial:partials/_sidebar.html -->
    <nav class="sidebar sidebar-offcanvas" id="sidebar">
      <div class="sidebar-brand-wrapper d-none d-lg-flex align-items-center justify-content-center fixed-top">
        <a class="sidebar-brand brand-logo" href="index.html"><img src="assets/images/logo.svg" alt="logo" /></a>
        <a class="sidebar-brand brand-logo-mini" href="index.html"><img src="assets/images/logo-mini.svg" alt="logo" /></a>
      </div>
      <ul class="nav">
        <li class="nav-item profile">
          <div class="profile-desc">
            <div class="profile-pic">
              <div class="count-indicator">
                <img class="img-xs rounded-circle " src="assets/images/faces/face15.jpg" alt="">
                <span class="count bg-success"></span>
              </div>
          <div class="profile-name">
            <h5 class="mb-0 font-weight-normal">Admin</h5>


          </div>
        </div>
        <a href="#" id="profile-dropdown" data-toggle="dropdown"><i class="mdi mdi-dots-vertical"></i></a>
        <div class="dropdown-menu dropdown-menu-right sidebar-dropdown preview-list" aria-labelledby="profile-dropdown">



          <div class="dropdown-divider"></div>
          <a href="#" class="dropdown-item preview-item">
            <div class="preview-thumbnail">
              <div class="preview-icon bg-dark rounded-circle">
                <i class="mdi mdi-calendar-today text-success"></i>
              </div>
            </div>
            <div class="preview-item-content">
              <p class="preview-subject ellipsis mb-1 text-small">To-do list</p>
            </div>
          </a>
        </div>
      </div>
    </li>

    <li class="nav-item menu-items">
      <a class="nav-link" href="index.html">
        <span class="menu-icon">
          <i class="mdi mdi-speedometer"></i>
        </span>
        <span class="menu-title">Dashboard</span>
      </a>
    </li>
  </ul>
</nav>
<!-- partial -->
<div class="container-fluid page-body-wrapper">
  <!-- partial:partials/_navbar.html -->
```

```
<nav class="navbar p-0 fixed-top d-flex flex-row">
  <div class="navbar-brand-wrapper d-flex d-lg-none align-items-center justify-content-center">
    <a class="navbar-brand brand-logo-mini" href="index.html"><img src="assets/images/logo-mini.svg" alt="logo" /></a>
  </div>
  <div class="navbar-menu-wrapper flex-grow d-flex align-items-stretch">
    <button class="navbar-toggler navbar-toggler align-self-center" type="button" data-toggle="minimize">
      <span class="mdi mdi-menu"></span>
    </button>

    <ul class="navbar-nav navbar-nav-right">

      <li class="nav-item dropdown">
        <a class="nav-link" id="profileDropdown" href="#" data-toggle="dropdown">
          <div class="navbar-profile">
            <img class="img-xs rounded-circle" src="assets/images/faces/face15.jpg" alt="">
            <p class="mb-0 d-none d-sm-block navbar-profile-name">Admin</p>
            <i class="mdi mdi-menu-down d-none d-sm-block"></i>
          </div>
        </a>
        <div class="dropdown-menu dropdown-menu-right navbar-dropdown preview-list" aria-labelledby="profileDropdown">


          <div class="dropdown-divider"></div>
          <a class="dropdown-item preview-item">
            <div class="preview-thumbnail">
              <div class="preview-icon bg-dark rounded-circle">
                <i class="mdi mdi-logout text-danger"></i>
              </div>
            </div>
            <div class="preview-item-content">
              <p class="preview-subject mb-1">Log out</p>
            </div>
          </a>
          </div>
      </li>
    </ul>
    <button class="navbar-toggler navbar-toggler-right d-lg-none align-self-center"
type="button" data-toggle="offcanvas">
      <span class="mdi mdi-format-line-spacing"></span>
    </button>
  </div>
 </nav>


           <div class="main-panel">
  <div class="content-wrapper">

    <div class="row">




      <div class="col-12 grid-margin stretch-card">
        <div class="card">
          <div class="card-body">
            <h4 class="card-title">Manage Services</h4>
            <p class="card-description">Add Services</p>
            <form class="forms-sample" action="" method="POST">
              <div class="form-group">
                <label for="name"> Service Name</label>
                <input type="text" class="form-control" name="name" id="name"
placeholder="Service Name">
              </div>
              <div class="form-group">
                <label for="servicedescription">Service Description</label>
                <input type="text" class="form-control" name="servicedescription"
id="servicedescription" placeholder="Service Description">
              </div>
              <div class="form-group">
                <label for="startdate">Start Date</label>
                <input type="date" class="form-control" name="startdate" id="startdate"
placeholder="Start Date">
              </div>
                                        <span id="msg1" style="color:red;"></span>
<script type="text/javascript"
   src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
```

```
<script type="text/javascript">
   $(function () {
       var today = new Date();
       var month = ('0' + (today.getMonth() + 1)).slice(-2);
       var day = ('0' + today.getDate()).slice(-2);
       var year = today.getFullYear();
       var date = year + '-' + month + '-' + day;
       $('[id*=date]').attr('min', date);
   });
</script>

                                    <div class="form-group">
                   <label for="enddate">End Date</label>
                   <input type="date" class="form-control"  name ="enddate" id="enddate"
     placeholder="End Date">
                 </div>
                                    <span id="msg1" style="color:red;"></span>
<script type="text/javascript"
     src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script type="text/javascript">
   $(function () {
       var today = new Date();
       var month = ('0' + (today.getMonth() + 1)).slice(-2);
       var day = ('0' + today.getDate()).slice(-2);
       var year = today.getFullYear();
       var date = year + '-' + month + '-' + day;
       $('[id*=date]').attr('min', date);
   });
</script>

                <!-- <div class="form-group">
                   <label>File upload</label>
                   <input type="file" name="img[]" class="file-upload-default">
                   <div class="input-group col-xs-12">
                     <input type="text" class="form-control file-upload-info" disabled
     placeholder="Upload Image">
                     <span class="input-group-append">
                       <button class="file-upload-browse btn btn-primary"
     type="button">Upload</button>
```

```
          </span>
         </div>-->
        </div>

                      <div class="form-group">
         <label for="document">Document required</label>
         <input type="file" class="form-control" name="document" id="document" 
placeholder="Document required" required>
        </div>

        <div class="form-group">
         <label for="qualification">Qualification</label>
         <textarea class="form-control" name="qualification" id="qualification" 
rows="4"></textarea>
        </div>
        <button type="submit" name="submit" class="btn btn-primary mr-
2">Submit</button>

       </form>
      </div>
     </div>
    </div>





   </div>
  </div>

 <!-- partial -->

  <!-- content-wrapper ends -->
  <!-- partial:partials/_footer.html -->
  <footer class="footer">

  </footer>
  <!-- partial -->
 </div>
```

```
      <!-- main-panel ends -->
    </div>
    <!-- page-body-wrapper ends -->
  </div>
  <!-- container-scroller -->
  <!-- plugins:js -->
  <script src="assets/vendors/js/vendor.bundle.base.js"></script>
  <!-- endinject -->
  <!-- Plugin js for this page -->
  <script src="assets/vendors/chart.js/Chart.min.js"></script>
  <script src="assets/vendors/progressbar.js/progressbar.min.js"></script>
  <script src="assets/vendors/jvectormap/jquery-jvectormap.min.js"></script>
  <script src="assets/vendors/jvectormap/jquery-jvectormap-world-mill-en.js"></script>
  <script src="assets/vendors/owl-carousel-2/owl.carousel.min.js"></script>
  <!-- End plugin js for this page -->
  <!-- inject:js -->
  <script src="assets/js/off-canvas.js"></script>
  <script src="assets/js/hoverable-collapse.js"></script>
  <script src="assets/js/misc.js"></script>
  <script src="assets/js/settings.js"></script>
  <script src="assets/js/todolist.js"></script>
  <!-- endinject -->
  <!-- Custom js for this page -->
  <script src="assets/js/dashboard.js"></script>
  <!-- End custom js for this page -->
 </body>
</html>
```

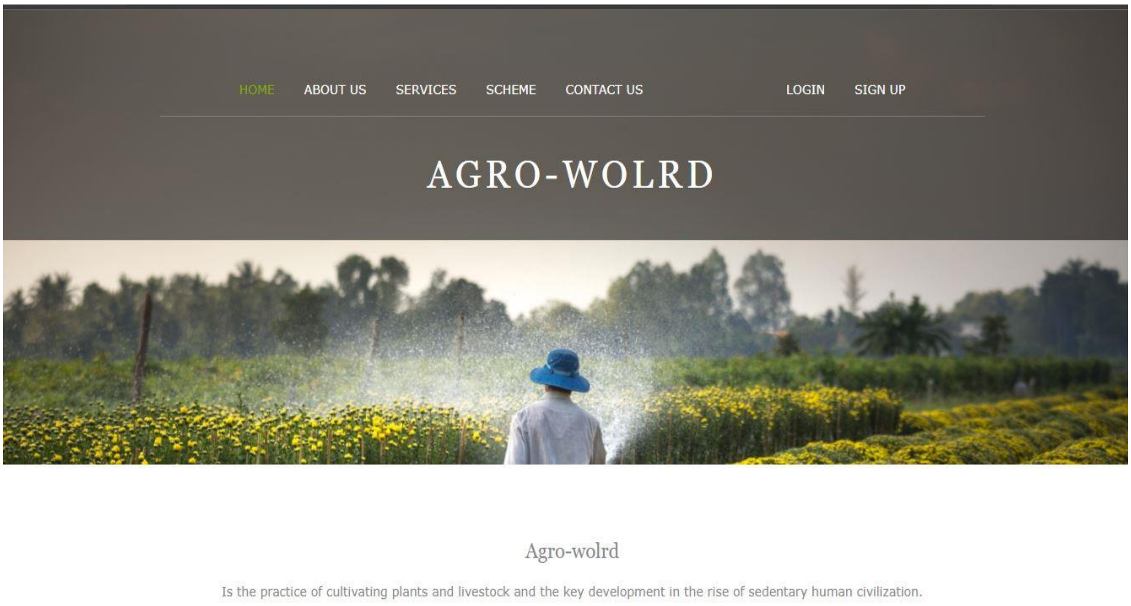
Delete1.php

```
<?php
include('connection.php');
?>
<?php
$id = $_GET['scheme_id'];
$del = mysqli_query($conn, "delete from scheme where scheme_id ='$id' ");
if($del)
{
mysqli_close($conn);
```

```
    header("location:listscheme.php");
    echo "alert";
    }
    else
    {
    header("location:listscheme.php");
    }
?>
```

9.2 ScreenShots

Home page

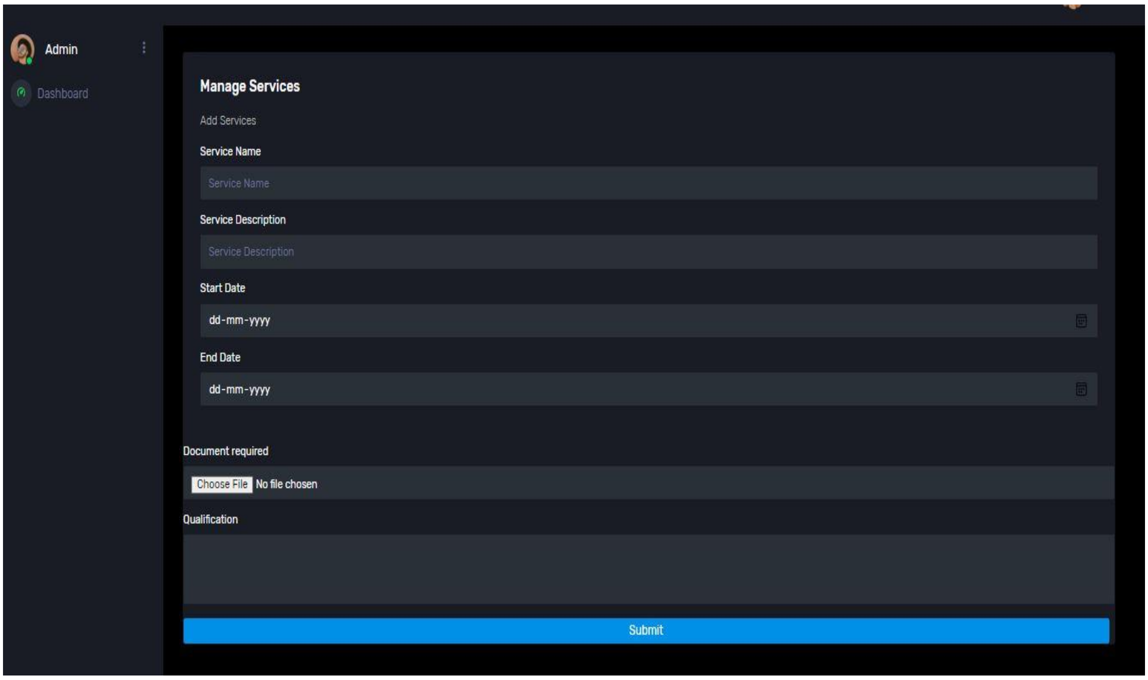


AGRO-WOLRD

Agro-wolrd

Is the practice of cultivating plants and livestock and the key development in the rise of sedentary human civilization.

Add Services page



Admin

Dashboard

Manage Services

Add Services

Service Name

Service Name

Service Description

Service Description

Start Date

dd-mm-yyyy

End Date

dd-mm-yyyy

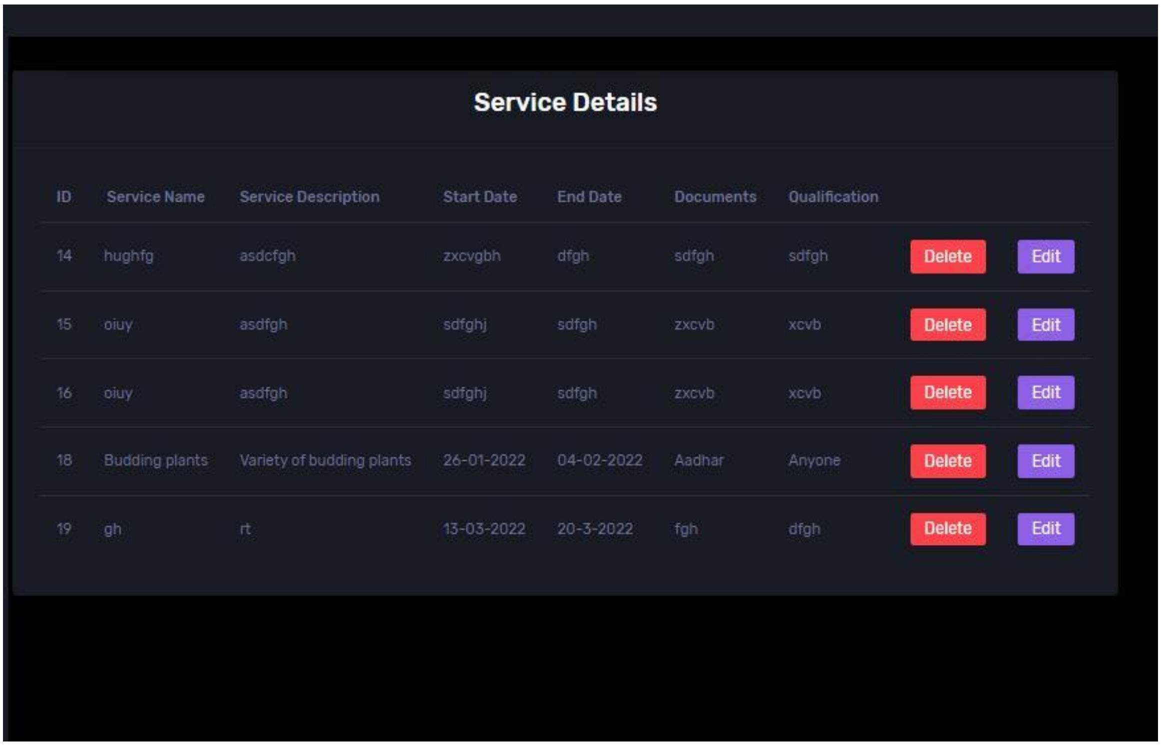
Document required

Choose File No file chosen

Qualification

Submit

Manage Services page



Service Details

ID	Service Name	Service Description	Start Date	End Date	Documents	Qualification		
14	hughfg	asdcfgh	zxcvgbh	dfgh	sdfgh	sdfgh	Delete	Edit
15	oiuy	asdfgh	sdfghj	sdfgh	zxcvb	xcvb	Delete	Edit
16	oiuy	asdfgh	sdfghj	sdfgh	zxcvb	xcvb	Delete	Edit
18	Budding plants	Variety of budding plants	26-01-2022	04-02-2022	Aadhar	Anyone	Delete	Edit
19	gh	rt	13-03-2022	20-3-2022	fgh	dfgh	Delete	Edit