

E-GRAMA PANCHAYAT

Project Report Submitted By

ASHTAMI PRASAD

Reg. No:AJC20MCA-2029

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**

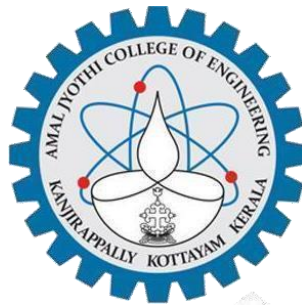
[Affiliated to APJ Abdul Kalam Technological University, Kerala.
Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally,
Kottayam, Kerala – 686518]

2020-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, “**E-GRAMA PANCHAYAT**” is the bonafide work of **ASHTAMI PRASAD (Reg.No:AJC20MCA-2029)** 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. Shelly Shiju George
Internal Guide

Ms. Nimmy Francis
Coordinator

External Examiner

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

DECLARATION

I hereby declare that the project report “**E-GRAMA PANCHAYAT**” is a bonafided 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:

KANJIRAPPALLY

ASHTAMI PRASAD

Reg.No:AJC20MCA-2029

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 **Rev. Fr. Dr. Rubin Thottupurathu Jose** and **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. Shelly Shiju George** 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.

ASHTAMI PRASAD

ABSTRACT

The main objective of this project for the better performance of grama panchayat service and to manage those services through computerized system. The target of this project is to create an E-Gram Panchayat. The internet can be used to access the online application known as E-Grama Panchayat. It is possible to use this system to keep an eye on grama panchayat operations.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	EXISTING SYSTEM	6
2.3	DRAWBACKS OF EXISTING SYSTEM	6
2.4	PROPOSED SYSTEM	6
2.5	ADVANTAGES OF PROPOSEDSYSTEM	6
3	REQUIREMENT ANALYSIS	8
3.1	FEASIBILITY STUDY	9
3.1.1	ECONOMICAL FEASIBILITY	9
3.1.2	TECHNICAL FEASIBILITY	9
3.1.3	BEHAVIORAL FEASIBILITY	10
3.2	SYSTEM SPECIFICATION	11
3.2.1	HARDWARE SPECIFICATION	11
3.2.2	SOFTWARE SPECIFICATION	11
3.3	SOFTWARE DESCRIPTION	11
3.3.1	PHP	11
3.3.2	MYSQL	12
4	SYSTEM DESIGN	14
4.1	INTRODUCTION	15
4.2	UML DIAGRAM	15
4.2.1	USE CASE DIAGRAM	16
4.2.2	SEQUENCE DIAGRAM	18
4.2.3	STATECHART DIAGRAM	20
4.2.4	ACTIVITY DIAGRAM	21
4.2.5	CLASS DIAGRAM	22
4.2.6	OBJECT DIAGRAM	23
4.2.7	COMPONENT DIAGRAM	24
4.2.8	DEPLOPMENT DIAGRAM	25

4.3	USER INTERFACE DESIGN	26
4.4	DATA BASE DESIGN	29
5	SYSTEM TESTING	36
5.1	INTRODUCTION	37
5.2	TEST PLAN	38
5.2.1	UNIT TESTING	38
5.2.2	INTEGRATION TESTING	39
5.2.3	VALIDATION TESTING	39
5.2.4	USER ACCEPTANCE TESTING	39
6	IMPLEMENTATION	43
6.1	INTRODUCTION	44
6.2	IMPLEMENTATION PROCEDURE	44
6.2.1	USER TRAINING	45
6.2.2	TRAINING ON APPLICATION SOFTWARE	45
6.2.3	SYSTEM MAINTENANCE	45
6.2.4	HOSTING	45
7	CONCLUSION & FUTURE SCOPE	47
7.1	CONCLUSION	48
7.2	FUTURE SCOPE	48
8	BIBLIOGRAPHY	49
9	APPENDIX	51
9.1	SAMPLE CODE	58
9.2	SCREEN SHOTS	81
9.3	PLAGARISM REPORT	86

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

1.1 PROJECT OVERVIEW

The major goal of this initiative is to use computerized systems to manage and enhance the performance of grama panchayat services.

1.2 PROJECT SPECIFICATION

The suggested system can fix these problems. It simplifies the procedure, requires no money, and allows individuals to learn about the services without going into the grama panchayat.

E-Grama Panchayat mainly contain four users.

- Admin
- Officers
- Users
- Staff

1. Admin Module

- The system's admin has logged in.
- The admin may add or alter services or schemes
- He/ She has the overall control of the system.
- Add staff and officers.
- Admin can view the applications of users.
- Approve or reject leave.
- View Complaint and replay to the complaint.
- Approve or reject leaves from staff and officer.
- View attendance

2. Staff:

- Managing user's application.
- Get details of users as and when required
- Apply Leave

3. Officers:

- Login
- View new request from user
- Apply leave
- Process application
- Generate report

4. Users:

- View services and apply for the services according to their needs.
- View schemes and apply for the schemes according to their needs.
- Upload required documents.
- Pay property tax.
- File complaint.

CHAPTER 2

SYSTEM STUDY

2.1. INTRODUCTION

The procedure of collecting and analyzing data, identifying problems, and using the data to recommend system modifications. During this problem-solving process, there must be extensive contact between system developers and users. Any system development process should start with a system evaluation or research. The system is thoroughly examined and assessed. The system analyst takes on the role of the interrogator and asks detailed questions regarding the operation of the current system. The system's input is acknowledged, and the system as a whole is viewed. The various processes can be connected to the organizational outputs. System analysis involves comprehending the problem, identifying the significant and crucial variables, analyzing and synthesizing the numerous components, and choosing the best or, at the very least, most acceptable course of action.

The procedure needs to be thoroughly investigated utilizing a range of approaches, such as surveys and interviews. The data acquired by various sources must be thoroughly scrutinized in order to make a choice. The conclusion is knowing how the system functions. This system is known as the current one. Trouble locations have been identified now that the problem with the current system has been thoroughly analyzed. The designer now assumes the position of a problem-solver and makes an effort to address the issues the business is facing. Proposals are used in place of the solutions. Following an analytical comparison of the plan and the current system, the best choice is chosen. The proposal is presented to the user with the option to accept or reject it. The proposal is evaluated and then necessary adjustments are made in response to user requests. This loop ends once the user is satisfied with the proposal.

Preliminary research is the process of collecting and analyzing data in order to use it for upcoming system investigations. Initial research requires strong collaboration between system users and developers since it involves problem-solving. It carries out several feasibility studies. The system activities are roughly estimated by these studies, which can be used to choose the methods to employ for effective system research and analysis.

2.2 EXISTING SYSTEM

The current system is not entirely automated. User will register and they can apply for service and schemes. Each user can create their own profile. The proposed system rectifies the drawbacks of the present system.

The current system must be altered in order to include new data, increase its efficiency, and make it more adaptable and secure. The new system allows users to view all services and plans.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Human effort is needed.
- Maintaining significant information in the user is challenging.
- More manual hours need to process application.

2.4 PROPOSED SYSTEM

The suggested system is intended to address every drawback of the current system. For grama panchayat, it is essential to have a system that is more user-friendly and appealing. The issues of existing system can be overcome by the proposed system. It makes the process easier and there is no expense required and people can know about the services without entering into the grama panchayat.

The data is kept by this program in a single, easily accessible area for all users. It is quite easy to manage historical data in a database. The staff can utilize this application with no special training. They can readily be used to cut down on the amount of time needed to conduct routine tasks manually, improving output. The database may easily be updated with information on online services and programs.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system's implementation and design are both rather simple. The system utilizes hardly any system resources and operates in almost every situation. It features the following things:

➤ **Better security:-**

Data security depends on preventing unauthorized access. Data protection refers to their defense against various forms of deletion. The four interrelated issues that makeup the system security challenge are security, integrity, privacy, and confidentiality. By requiring a username and password to log in, security is preserved. Data security will be guaranteed because we maintain the documents in secure databases.

➤ **Ensure data accuracy:-**

Human error in the user information entry process during registration is eliminated by the suggested fix

➤ **Better service: -**

With this solution, hard copy storage won't be a hassle. By completing the same activity in a different manner, we can also save time and resources. The data can be kept for a longer time without losing any information.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is completed to determine whether the project would ultimately succeed in meeting the goals of the organization given the work, effort, and time put into it. The creator of the project can predict the project's potential future and usefulness thanks to a feasibility study. A system's efficacy is determined by considering how it affects the organization, user needs, and resource efficiency. As a result, before a new application is given the go-ahead for development, a feasibility review is typically carried out.

The viability of the project is described in the article, along with a list of the several aspects—such as technical, economic, and operational viability—that were carefully taken into account. It has the following characteristics:

3.1.1 Economical Feasibility

The creation of an E-Gram Panchayat is the goal of this project. A web-based application called E-Grama Panchayat is available online and may be used anywhere. The monitoring of grama panchayat operations may be done using this system. Following are some important financial questions that were brought up during the original investigation:

- The expenses carry out an extensive system investigation
- The price of the equipment and software.
- The lower costs or fewer costly errors are examples of advantages.

There are no extra costs associated with human labor because the offered solution was created as part of a project. Furthermore, the system may be developed on a budget given that all required resources are now accessible.

System costs, development costs, and hosting costs make up the three cost categories for the DREAMS project. All estimates show that the project was created at a reasonable cost, given that open-source technologies were used exclusively during its creation.

3.1.2 Technical Feasibility

The system needs to be technologically assessed first. An overview design of the system's requirements in terms of input, output, program, and procedures must serve as the foundation for the viability assessment. After constructing an outline system, the inquiry must next recommend the kind of tools, the procedures needed to create the system, and the methods to use the system once it has been created.

During the investigation, the following technical concerns were encountered:

- Does the available technology permit the proposed fixes?
- If the system is developed, can it expand?

The project should be planned so that the necessary functionality and performance are met within the constraints. The idea incorporates encryption and calls for a high resolution scanning device. Even though the technology may eventually become obsolete, the system can still be used because a newer version of the same program works with an earlier version. Consequently, there aren't many restrictions on this project. Technically, the project can be completed because the system was created utilizing PHP for the front end and a MySQL server for the back end. Technically, the project can be completed because the system was created utilizing PHP for the front end and a MySQL server for the back end. The system was also well-performing, with an Intel i3 core processor, 4GB of RAM, and a 1TB hard drive.

3.1.3 Behavioral Feasibility

The proposed system contains the following inquiries:

- Is there enough assistance for users?
- Will anyone be harmed by the proposed system?

The project would be advantageous since it will accomplish the goals when it is developed and put into action. After carefully assessing all behavioral aspects, it is determined that the project is behaviorally feasible.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	Intel corei3
RAM	-	4 GB
Hard disk	-	1 TB

3.2.2 Software Specification

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

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server-side scripting language used for creating websites as well as other types of program. PHP is now used by 2.1 million web servers and more than 244 million webpages. The reference version of PHP, which Rasmus Ledorf created in 1995, is now created by the PHP group. Personal Home Page (PHP) is a recursive acronym that is now used to refer to PHP: Hypertext Preprocessor. The PHP processor module on a web server translates PHP code to create the finished web page. PHP instructions can be directly included in an HTML source file to handle data instead of utilizing an external file. The GNU General Public License is incompatible with PHP since it has evolved to incorporate a command-line interface capability and can be used independently due to restrictions on the use of the word (GPL). On practically all platforms and operating systems, as well as on the majority of web servers, PHP is freely installable as a standalone shell.

3.3.2 MySQL

The most well-known Open Source SQL database management system, MySQL, was developed, distributed, and supported by Oracle Corporation. On the MySQL website, you may find the most latest information regarding the MySQL program.

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

A planned collection of data is called a database. Anything might be it, including a straightforward shopping list, a photo gallery, or the enormous amount of data in a company network. Data included in a computer database must be added to, accessed, and processed using a database management system, such as MySQL Server. Database management systems—whether used as standalone program or as a component of other applications—are essential to computing because computers are so adept at processing massive volumes of data.

- **MySQL databases are relational.**

Instead of combining all the data into one huge warehouse, a relational database keeps it in individual tables. The database structures are kept in physically fast-loading files. The logical model, which consists of objects like databases, tables, views, rows, and columns, provides a flexible programming environment. One-to-one, one-to-many, unique, required or optional, and "pointers" between other tables are just a few examples of the rules you may make to control the relationships between various data fields. The database upholds these rules, making sure that your application never runs into problems with inconsistent, duplicate, orphan, outdated, or missing data. The "Structured Query Language" prefix "SQL" in MySQL stands for this phrase. SQL is the most widely used standard language for database access. You might explicitly enter SQL (for instance, to generate reports), embed SQL statements into other languages' code, or use a language-specific API that hides the SQL syntax depending on your programming environment. SQL is specified by the ANSI/ISO SQL Standard. Since its inception in 1986, the SQL standard has undergone multiple revisions. In this handbook, "SQL92" refers to the standard that was published in 1992, "SQL:1999" to the standard that was published in 1999, and "SQL:2003" to the version of the standard that is now in use. The SQL Standard as it exists at any one time is referred to as "the SQL standard."

- **MySQL software is Open Source.**

Considering that the software is open source, anyone can use and alter it. Anyone can use and download the MySQL software for free online. You have the right to look at the source code and make any necessary changes. The GPL (GNU General Public License) which specifies what you may and may not do with the program in certain circumstances, is used by the MySQL software. If the GPL makes you uncomfortable or if you need to incorporate MySQL code into a for-profit application, you can buy a commercially licensed version from us. Check out the MySQL Licensing Overview for more information. The MySQL Database Server is very fast, reliable, scalable, and easy to use.

You ought to give it a shot if that is what you're after. In addition to your other apps, web servers, and other software, MySQL Server can function smoothly on a desktop or laptop while requiring little to no maintenance. You can modify the settings to utilize all the RAM, CPU power, and I/O capacity if you dedicate an entire machine to MySQL.

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

A client/server system, the MySQL Database Software features a multi-threaded SQL server that supports a number of client programs and libraries, management tools, and a wide range of application programming interfaces (APIs). Additionally, we offer MySQL Server as a multi-threaded embedded library that you can integrate into your program to create a more manageable, quicker, and more compact standalone solution.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Any engineered system or product's development process begins with design. A creative process is design. The secret to an efficient system is a decent design. Design is the process of thoroughly outlining a process or system so that it can be physically executed by using a variety of methodologies and concepts. It might be defined as the process of employing various concepts and methods to specify a device, processor, or system with insufficient specificity to enable physical implementation. Software design serves as the technical foundation of the software engineering process, regardless of the development paradigm used. The architectural information needed to construct a system or product is generated by the system design. As with any systematic technique, this software underwent the best design phase possible, fine-tuning all efficiency, performance, and accuracy levels. A user-oriented document becomes a document for programmers or database staff throughout the design phase. The two stages of system design development are logical design and physical design.

4.2 UML DIAGRAM

A common language known as UML is used to specify, visualize, build, and document the software system artefacts. The Object Management Group (OMG) was responsible for developing UML, and a draught of the UML 1.0 definition was presented to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from other well-known programming languages like C++, Java, COBOL, etc. Software blueprints are made using a visual language called UML. UML is referred to as an all-purpose visual modelling language for the visualization, specification, construction, and documentation of software systems. UML is not just used to represent software systems, despite the fact that this is its most common application. It is also used to model systems that are not software-based. For instance, the manufacturing facility's process flow, etc. Although UML is not a programming language, tools can be used to generate code using UML diagrams in a variety of languages. The analysis and design of objects-oriented systems are directly related to UML. UML has become an OMG standard after some standardization. The

visual impact of the UML diagram is the most important factor in the entire process.

Utilizing all the other elements completes it. The UML includes the next 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

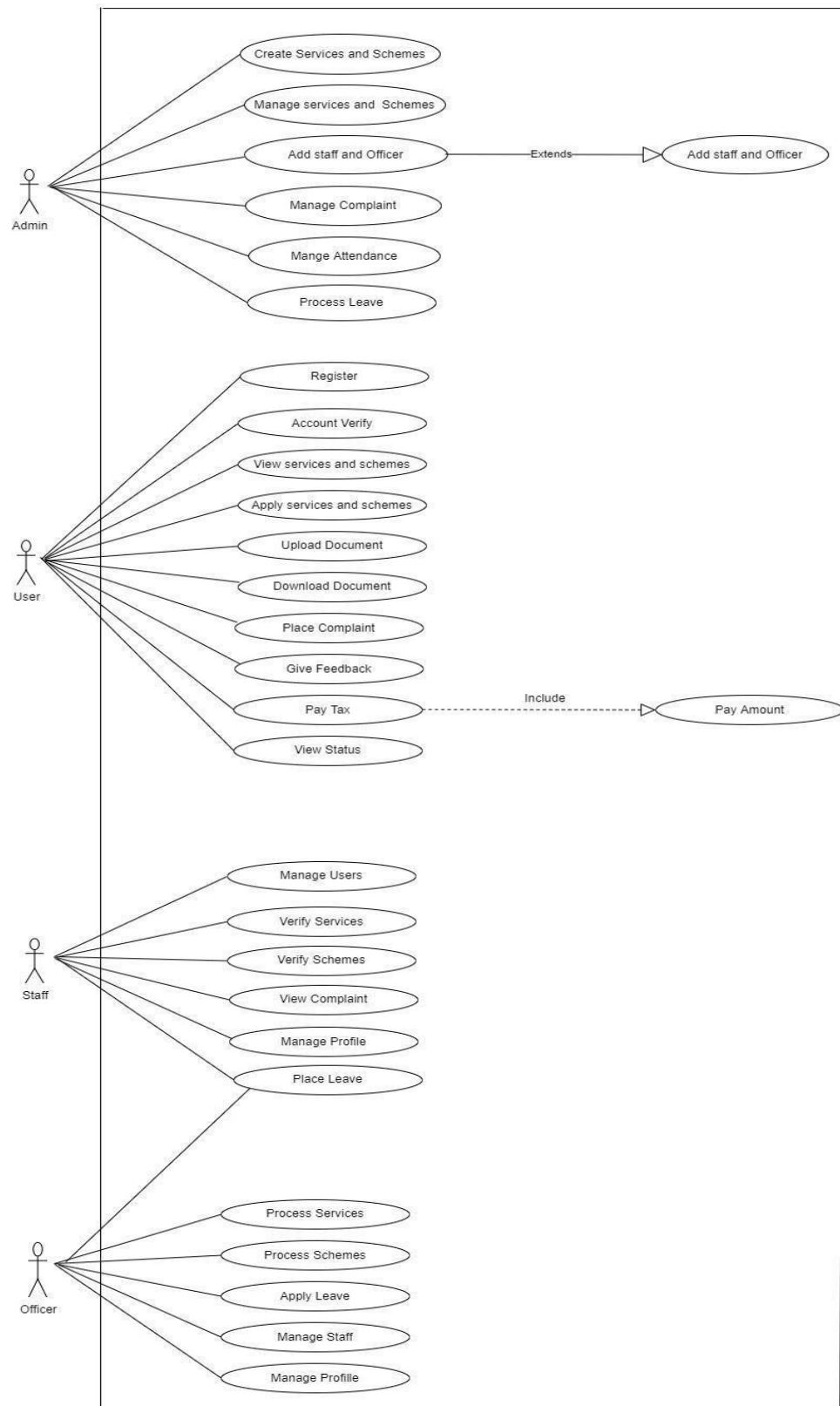
A use case diagram serves as an example of the interactions between system parts. A use case is a tool for identifying, categorizing, and organizing system requirements. In this context, the word "system" refers to something that is being created or run, such as a website for product sales and mail-order services. UML (Unified Representing Language), a common notation for modelling real products and systems, uses use case diagrams. Planning general requirements, validating hardware designs, testing and debugging software products while they are still in development, creating online help resources, and finishing customer support-focused tasks are a few examples of system objectives. For instance, customer support, item ordering, catalogue updating, and payment processing are examples of use cases in a setting of product sales. A use case diagram consists of four components. The border, which establishes the relationship between the system of interest and its environment.

- The actors within and surrounding the system perform the use cases, which are the specialized roles.
- The actors and use cases are connected to one another and interact.

Use case diagrams are created to depict a system's functional requirements. To create an effective use case diagram after identifying the aforementioned things, we must adhere to the following rules.

- A use case's naming is very significant. The name should be selected in a way that makes it clear what functions are being performed. Give a suitable name for actors.
- Clearly depict links and dependencies in the diagram.

- Since the primary goal of the diagram is to define the needs, avoid attempting to include all possible relationships.
- Use notes whenever required to clarify some important points.



4.2.2 SEQUENCE DIAGRAM

A sequence diagram fundamentally depicts the sequential order in which events occur or how they interact with one another. Event diagrams and event scenarios are other names for sequence diagrams. Sequence diagrams display the actions performed by a system's parts in a time-based manner. These diagrams are widely used by businesspeople and software engineers to document and explain the requirements for new and existing systems.

Sequence Diagram Notations –

- **Actors** – An actor in a UML diagram symbolizes a specific kind of role in which it interacts with the system's elements. An actor is never within the scope of the system that we want to describe using the UML diagram. For a range of roles, including those of human users and other external subjects, we use actors. An actor is shown using the stick person notation in a UML diagram. A sequence diagram could contain multiple actors.
- **Lifelines** – A lifeline is a named component that displays a particular participant in a sequence diagram. A lifeline essentially stands in for each incident in a sequence diagram. In a sequence diagram, the lifeline components are at the top.
- **Messages** – It is shown how things can communicate with one another through messages. On the lifeline, the messages are presented in reverse chronological order. Messages are represented by arrows. The two major elements of a sequence diagram are lifelines and messages.

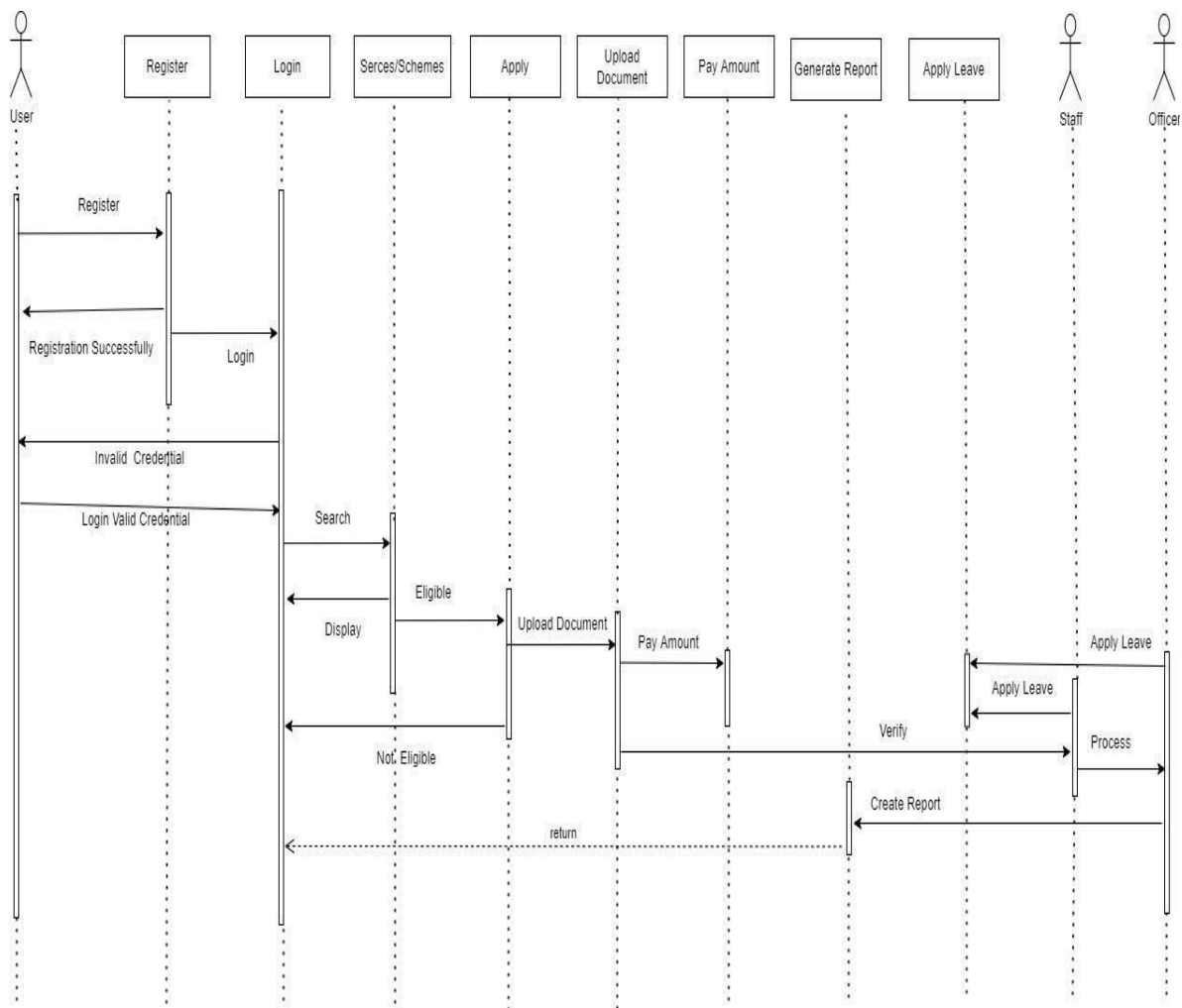
The following categories serve as general classifications for messages:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Replay Message
- Found Message
- Lost Message

- **Guards** – In the UML, we utilize guards to model circumstances. When we need to limit the flow of messages under the guise of a condition being met, we use them. Software engineers rely on guards to inform them of the limitations imposed by a system or specific process.

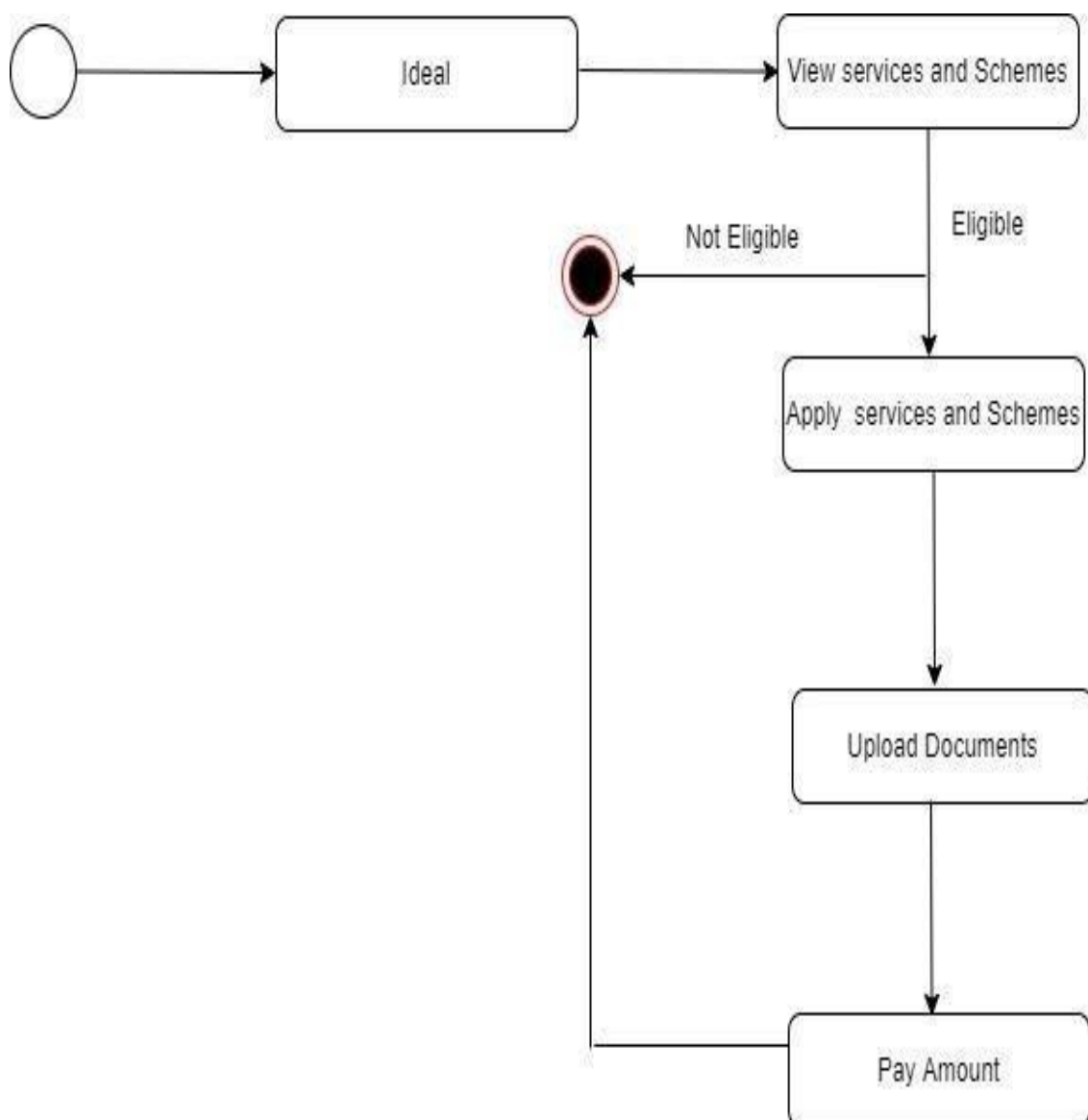
Uses of sequence diagrams –

- Employed to model and visualize the reasoning behind a complex function, operation, or process.
- They are also used to show details of UML use case diagrams.
- Used to comprehend the precise operation of present or upcoming systems.
- Visualize the flow of information between various components or objects.



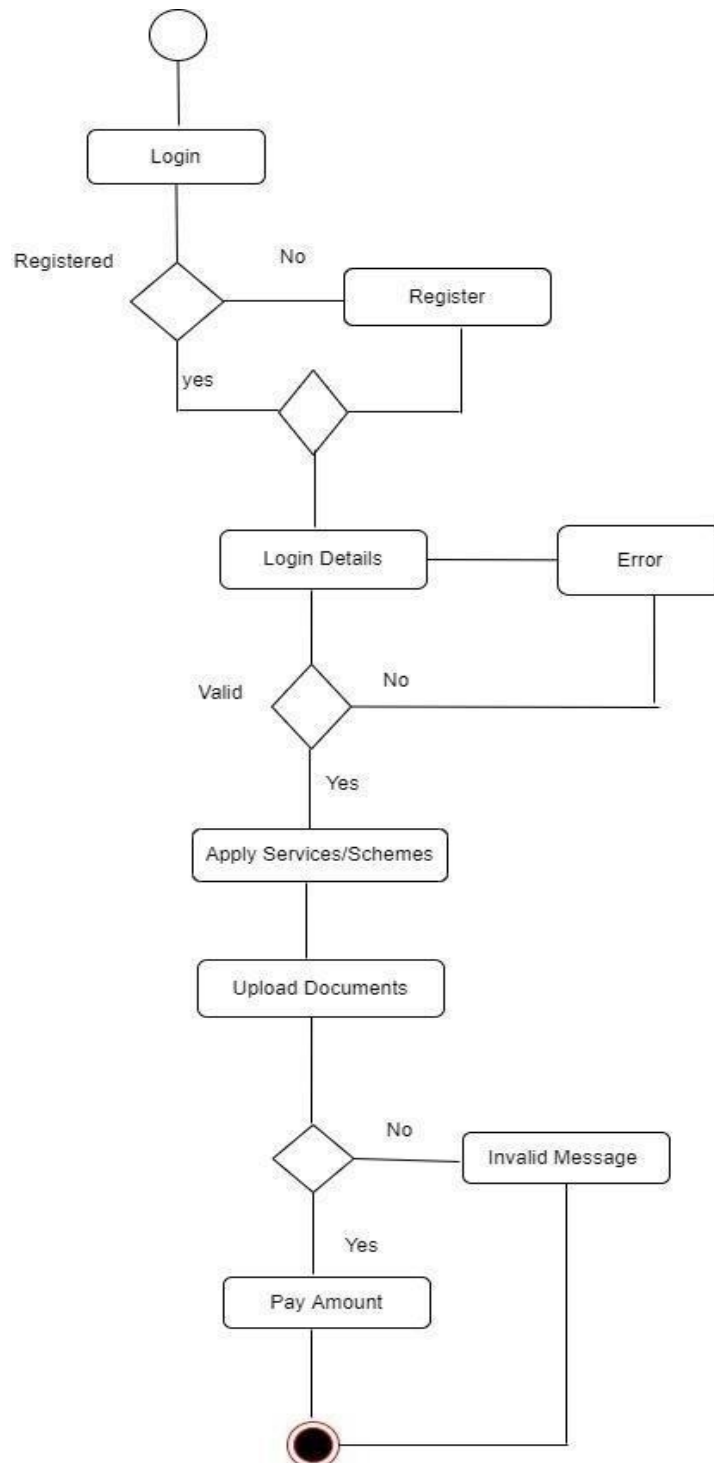
4.2.3 STATE CHART DIAGRAM

State diagrams are used to depict how a software system behaves. A class, a subsystem, a package, or even an entire system's behavior can be represented by a state machine diagram in a UML model. State charts and state transition diagrams are other names for it. State chart diagrams give us a useful approach to represent the communications or interactions that take place between external entities and a system. The event-based system is modelled using these diagrams. With the aid of an event, a state of an object can be managed. To depict the multiple states of an entity within the application system, state chart diagrams are utilized.



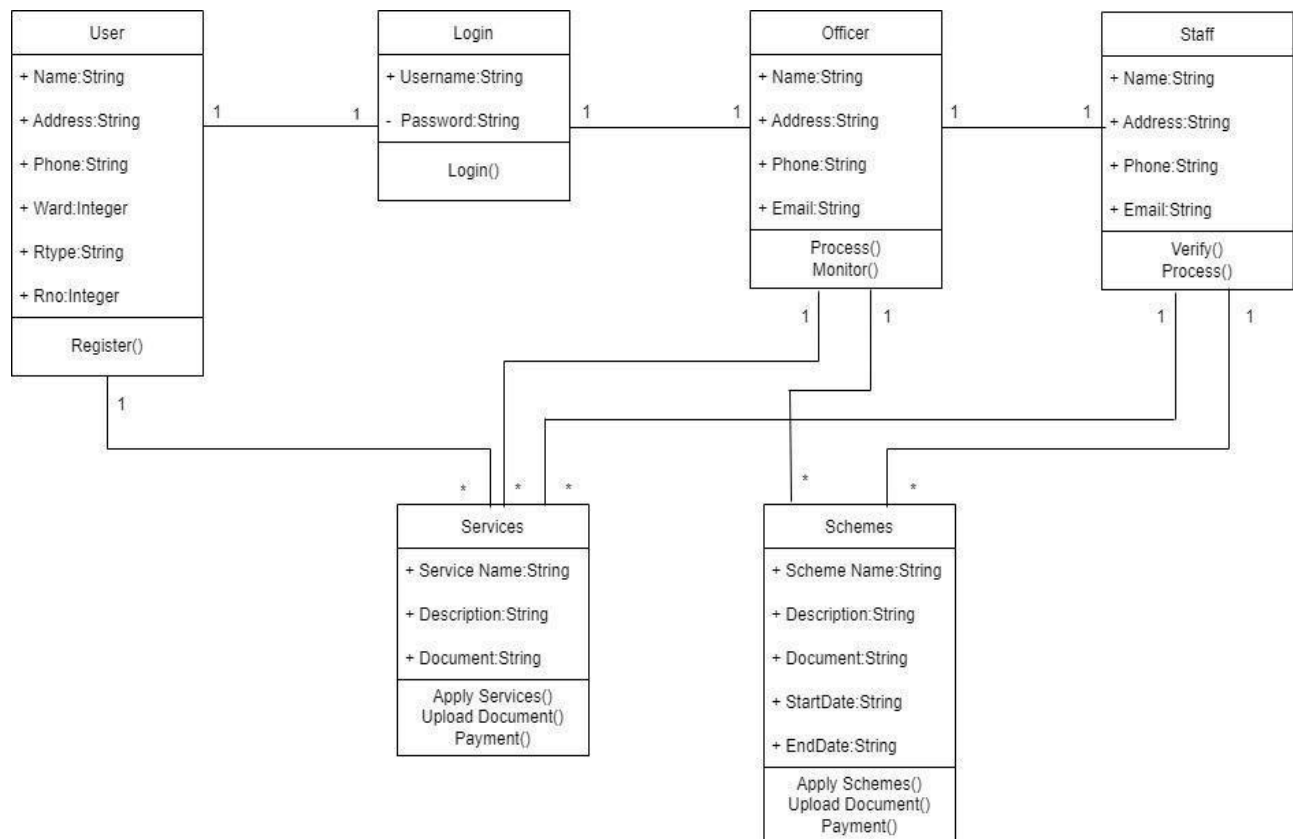
4.2.4 ACTIVITY DIAGRAM

Activity diagrams demonstrate the coordination of several layers of activity abstraction necessary to deliver a service. Usually, a few operations are required to complete an event, especially when those operations must coordinate numerous different tasks. How the events in a single use case interact to one another is another frequent requirement, particularly in use cases where activities may overlap and require coordination.



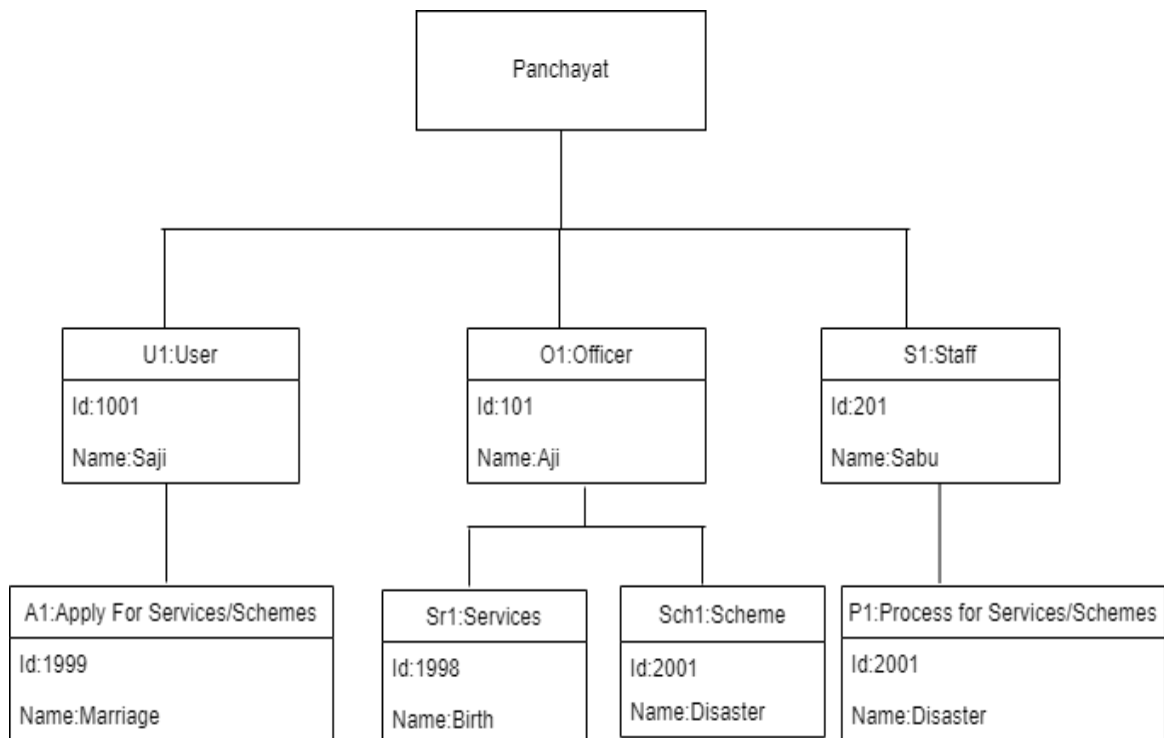
4.2.5 CLASS DIAGRAM

Class diagrams are a type of static diagram. It represents the static view of the application. Class diagrams are used for visualizing, describing, and documenting various system components as well as for producing executable code for software applications. A class diagram describes the constraints imposed on the system along with the properties and operations of a class. Class diagrams are widely used in the modelling of object-oriented systems since they are the only UML diagrams that can be directly translated using object-oriented languages. A class diagram shows a collection of classes, interfaces, affiliations, collaborations, and constraints. Additionally, known as a "structural diagram".



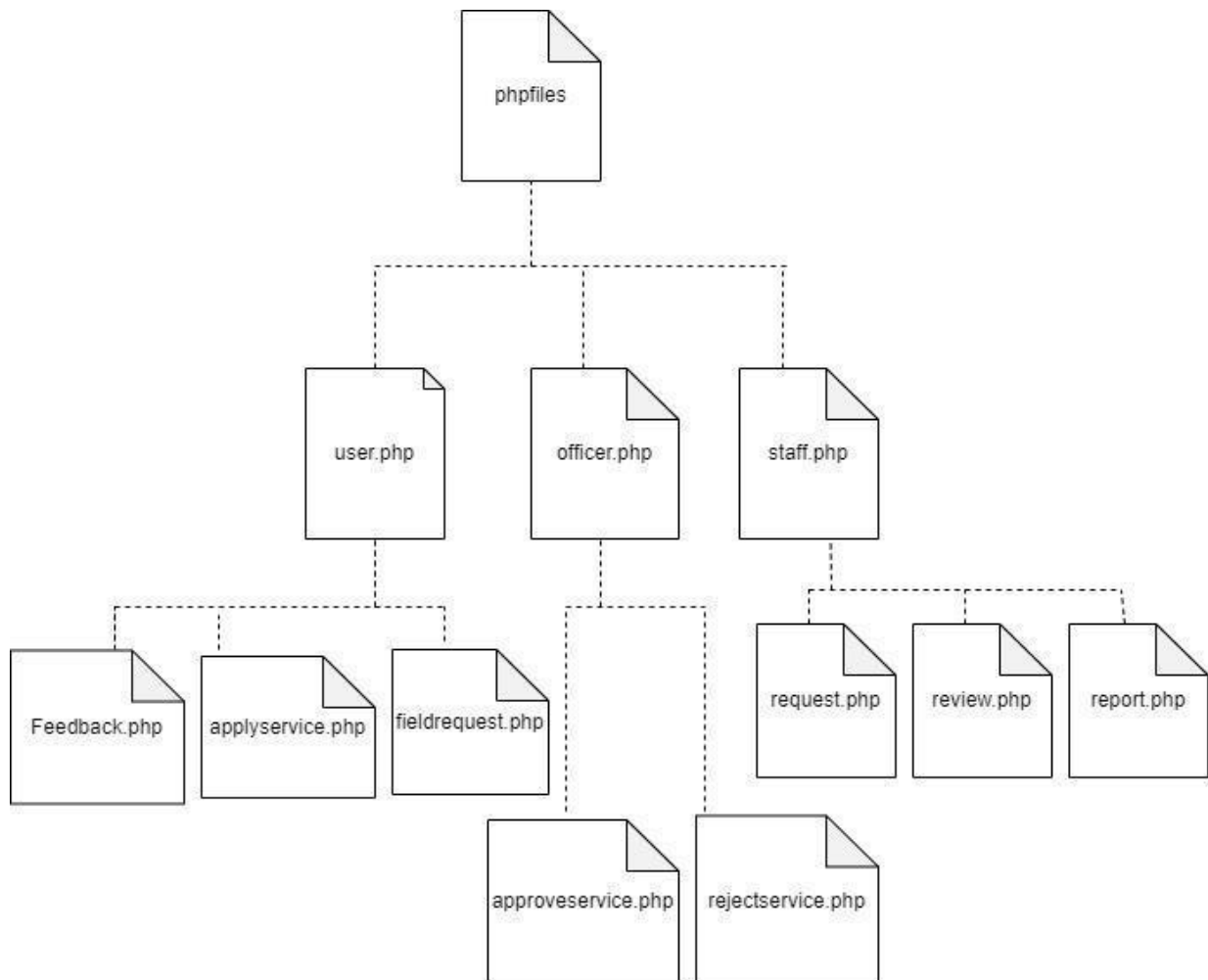
4.2.6 OBJECT DIAGRAM

Since class diagrams are the source of object diagrams, class diagrams are a prerequisite for object diagrams. An instance of a class diagram is represented by an object diagram. Class and object diagrams both use the same fundamental ideas. The static view of a system is also represented by object diagrams, but this static view represents a momentary snapshot of the system. To represent a group of items and their connections as an instance, object diagrams are employed.



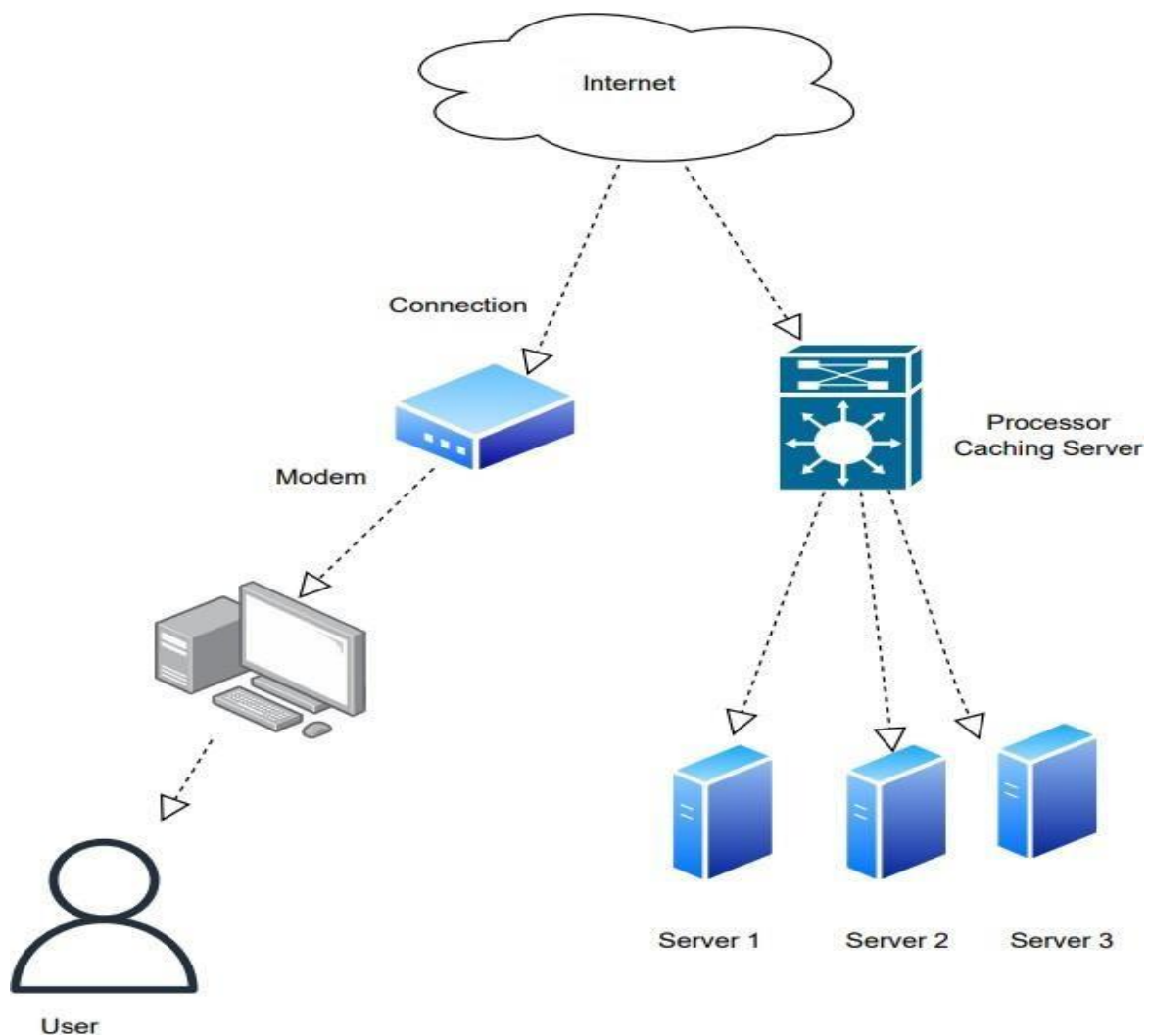
4.2.7 COMPONENT DIAGRAM

Component diagrams vary in their behavior and character. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other things that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams can also be used to create systems that can be executed.



4.2.7 DEPLOYMENT DIAGRAM

Deployment diagrams show the topology of a system's physical components, where the software components are installed. Deployment diagrams are used to describe a system's static deployment view. The key elements of deployment diagrams are nodes and connections between them.



4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

Form Name : User Registration



The image displays a user registration form for 'E Panchayath'. The form is divided into two main sections. The left section has a light green background and features a circular logo with a laptop, a checkmark, and a 'Login' button, with the word 'Register' in large red text below it. The right section has a light gray background and is titled 'REGISTRATION'. It contains a series of input fields for user information: Aadhar Number, Name, Address, City, PinCode, Phone Number, Gender (a dropdown menu), Ward Number, Ration Card Number, Ration Card Type (a dropdown menu), Id Card Number, House Number, UserName, and Password. At the bottom of the right section, there is a red 'REGISTER' button and a link for 'Already a Registered User? Signup'.

E Panchayath

Register

REGISTRATION

Aadhar Number

Name

Address

City

PinCode

Phone Number

Gender

Ward Number

Ration Card Number

Ration Card Type

Id Card Number

House Number

UserName

Password

REGISTER

Already a Registered User? [Signup](#)

Form Name : User Login



The login form features a green header with the E-Grama Panchayat logo and a large illustration of a tree with people sitting around its base. The word 'WELCOME' is written in colorful letters at the bottom left. The main content area is light gray and contains the title 'LOGIN' in large black letters. Below the title are input fields for 'User Name' and 'Password', with a padlock icon next to the password field. A 'Forgot Password?' link is positioned below the password field. A green 'Login' button is centered below these fields. At the bottom, there is a link that says 'Don't Have an Account ? Register'.

E Panchayath

LOGIN

User Name _____

Password _____ 

[Forgot Password ?](#)

Login

Don't Have an Account ? [Register](#)

Form Name : Apply Services



The 'Apply For Services' form has a green header with the E-Grama Panchayat logo. The title 'Apply For Services' is in large, bold, black letters. The form contains several input fields: 'Name', 'Address', 'City', 'Post', 'Date of Birth' (with a calendar icon), 'Gender' (with a dropdown arrow), 'Service Type' (with a dropdown arrow), 'Upload Document' (with a document icon), and 'Applied Date' (with a calendar icon). A large green 'Apply' button is at the bottom center. On the right side, there is an illustration of a person sitting at a desk with a computer, surrounded by other people and a tree.

E Panchayath

Apply For Services

Name _____

Address _____

City _____

Post _____

Date of Birth 

Gender 

Service Type 

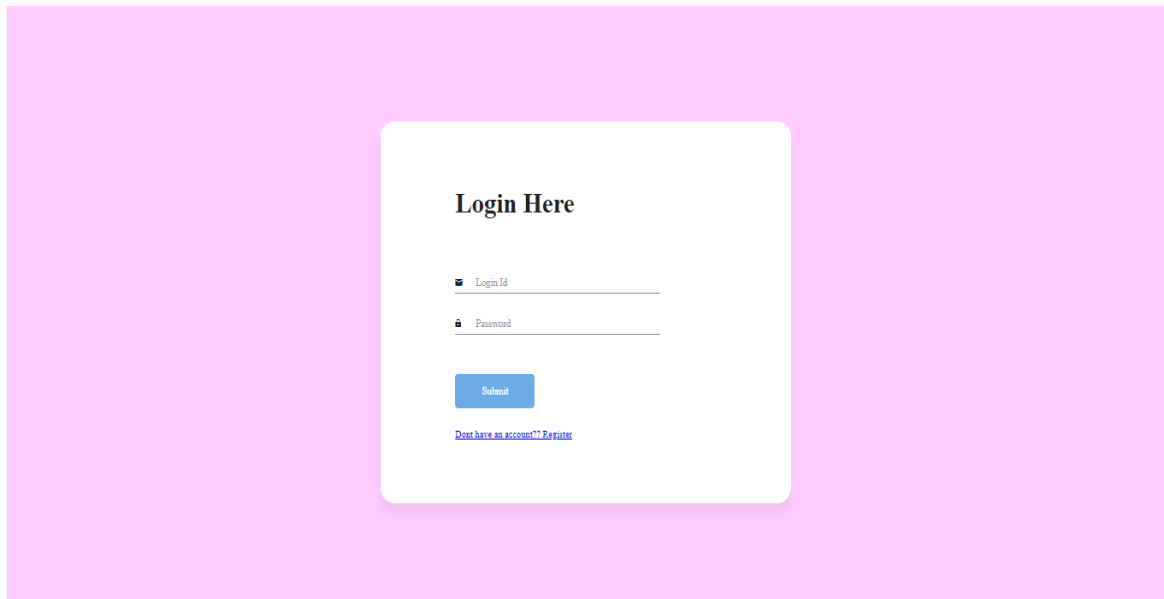
Upload Document 

Applied Date 

Apply

4.3.2 OUTPUT DESIGN

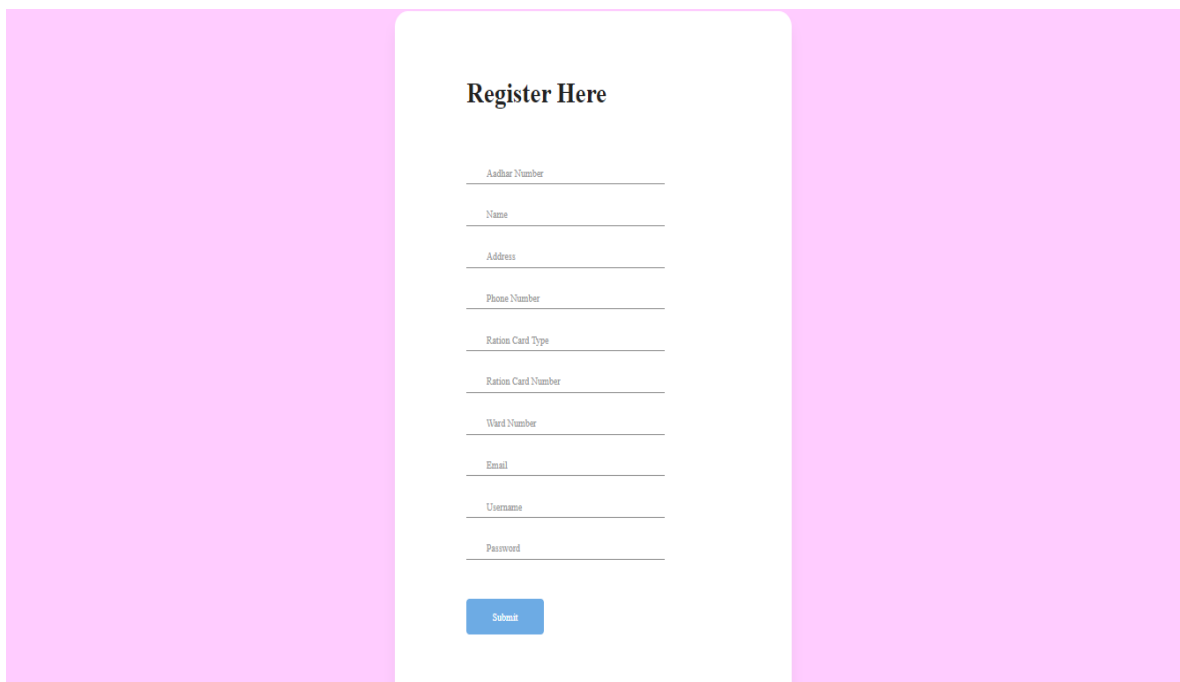
User Login

A user login form titled "Login Here" is centered on a light pink background. The form is a white rounded rectangle with a subtle drop shadow. It contains two input fields: "Login Id" with a key icon and "Password" with a lock icon. Below these is a blue "Submit" button. At the bottom, there is a blue link that says "Don't have an account? Register".

Login Here

[Don't have an account? Register](#)

User Registration

A user registration form titled "Register Here" is centered on a light pink background. The form is a white rounded rectangle with a subtle drop shadow. It contains ten input fields: "Aadhar Number", "Name", "Address", "Phone Number", "Ration Card Type", "Ration Card Number", "Ward Number", "Email", "Username", and "Password". Below these is a blue "Submit" button.

Register Here

4.4 DATABASE DESIGN

A database is a structured system with the capacity to store information and allows users to retrieve stored information quickly and effectively. Any database's primary goal is its data, which demands protection.

There are two stages to the database design process. The user needs are obtained in the first step, and a database is created to as clearly as possible meet these criteria. This process, known as information level design, is carried out independently of all DBMSs.

The design for the specific DBMS that will be used to implement the system in question is converted from an information level design to a design in the second step. Physical Level Design is the stage where the characteristics of the particular DBMS that will be used are discussed. Parallel to the system design is a database design. The database's data organization aims to accomplish the two main goals listed below.

- Data Integrity
- Data independence

4.4.1 Relational Database Management System (RDBMS)

The database is represented as a collection of relations in a relational paradigm. A table or file of records with values can be compared to each relation. In formal relational model terminology, a row is referred to as a tuple, a column heading is referred to as an attribute, and the table is referred to as a relation. Numerous tables, each with a unique name, makeup a relational data base. Each row in a tale reflects a set of related values.

Relations, Domains & Attributes

A relation is a table. Tuples are the units of a table's rows. An ordered group of n elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A group of atomic values make up a domain D. Choosing a data type from which the domain's data values are derived is a typical way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name. Each value in a relation is atomic and cannot be broken down.

Relationships

- Key is used to create table relationships. Primary Key and Foreign Key are the two principal keys that are most crucial. With the use of these keys, relationships for entity integrity and referential integrity can be created.
- Entity Integrity forbids the use of null values for any Primary Key.
- No Primary Key may contain null values, according to Referential Integrity.
- Referential Integrity Each unique Foreign Key value must have a corresponding Primary Key value in the same domain to maintain referential integrity. Super Key and Candidate Keys are additional keys.

4.4.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. The formal process of normalizing data structures in a way that reduces duplication and fosters integrity. Using the normalization technique, superfluous fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. A row in a table is uniquely identified by a key. Primary keys and foreign keys are the two different kinds of keys. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalized.

It means placing things in their natural form, as the name suggests. By using normalization, the application developer aims to establish a coherent arrangement of the data into appropriate tables and columns, where names may be quickly related to the data by the user. By removing recurring groups from the data, normalization prevents data redundancy, which puts a heavy strain on the computer's resources. These consist of:

Normalize the data.

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

First Normal Form

The domain of an attribute may only contain atomic values, and each attribute's tuple value must be a single value from that attribute's domain in accordance with the First Normal Form. By 1NF, single atomic or indivisible values are the only attribute values that are allowed, and 1NF prohibits "relations within relations" or "relations as attribute values within tuples". Data entry into Initial Normal Form is the first phase. This can be accomplished by separating data into tables of a similar type in each table. Depending on the needs of the project, a Primary Key or Foreign Key is assigned to each table. For each nested attribute or non-atomic attribute in this, we create new relations. Depending on the needs of the project, a Primary Key or Foreign Key is assigned to each table. For each nested relation or non-atomic attribute, new relations are formed in this process. This got rid of data groups that were repeated. If a relation solely meets the constraints that include the primary key, it is said to be in first normal form.

Second Normal Form

No non-key attribute should, in accordance with Second Normal Form, be functionally dependent on a portion of the primary key for relations when the main key has several attributes. For each partial key and its dependent attributes, we decompose and set up a new relation in this manner. Maintain the relationship between the original primary key and any properties that are entirely dependent on it. This procedure aids in removing data that depends only on a small portion of the key. If and only if a relation satisfies all the requirements for first normal form for the primary key and all of the non-primary key qualities of the relation are completely dependent on the primary key alone, then that relation is said to be in second normal form.

Third Normal Form

Relation should not have a non-key attribute that is functionally determined by another non-key attribute or by a collection of non-key attributes, according to the Third Normal Form. The primary key should not be transitively dependent, in other words. The non-key attributes that functionally determine other non-key attributes are decomposed in this way put up in relation. This procedure is used to eliminate anything not wholly dependent on the Primary Key. Only when a relation is in second normal form and, more importantly, when its non-key characteristics do not depend on those of other non-key attributes, is it considered to be in third normal form.

TABLE DESIGN**Table: login****Primary Key: logid**

Sno	Field Name	Data Type	Constraints
1	logid	Integer(11)	PrimaryKey
2	uname	varchar(20)	NotNull
3	psw	varchar(20)	NotNull
4	role	int(5)	NotNull

Table: registartion**Primary key: id****Primary key:adhno**

sno	Field Name	Data Type	Constraints
1	id	Integer(10)	Primary Key
2	adhno	bigint(20)	Primary Key
3	name	varchar(20)	Not Null
4	adr	varchar(30)	NotNull
5	phno	int(15)	NotNull
6	rtype	varchar(10)	NotNull
7	rno	int(20)	NotNull
8	wno	int(10)	NotNull
9	email	varchar(20)	NotNull
10	uname	varchar(100)	NotNull
11	psw	varchar(100)	NotNull
12	status	varchar(10)	NotNull

Table: cscheme**Primary Key: scid**

Sno	Field Name	Data Type	Constraints
1	scid	int(11)	PrimaryKey
2	name	varchar(100)	NotNull
3	start	Date	NotNull
4	end	Date	NotNull
5	descrip	varchar(100)	NotNull
6	doc1	varchar(100)	NotNull
7	doc2	varchar(100)	NotNull
8	doc3	varchar(100)	NotNull
9	doc4	varchar(100)	NotNull
10	doc5	varchar(100)	NotNull
11	quali	varchar(100)	NotNull
12	status	varchar(100)	NotNull

Table: cservice**Primary Key: sid**

Sno	Field Name	Data Type	Constraints
1	sid	int(100)	PrimaryKey
2	name	varchar(100)	NotNull
3	description	varchar(100)	NotNull
4	doc1	varchar(100)	NotNull
5	doc2	varchar(100)	NotNull
6	doc3	varchar(100)	NotNull
7	doc4	varchar(100)	NotNull
8	doc5	varchar(100)	NotNull
9	status	varchar(100)	NotNull

Table: cstaffofficer**Primary Key: soid**

Sno	Field Name	Data Type	Constraints
1	soid	int(10)	PrimaryKey
2	name	varchar(100)	NotNull
3	address	varchar(100)	NotNull
4	phone	varchar(100)	NotNull
5	email	varchar(100)	NotNull
6	date	Date	NotNull
7	gender	varchar(100)	NotNull
8	status	varchar(100)	NotNull

Table: complaint**Primary Key: atid**

Sno	Field Name	Data Type	Constraints
1	atid	Integer(10)	Primary Key
2	name	varchar(100)	NotNull
3	date	date	NotNull
4	poa	Varchar(10)	NotNull
5	status	varchar(100)	NotNull

Table: attendance**Primary Key: atid**

<u>Sln</u>	<u>Field Name</u>	<u>Data Type</u>	<u>Constraints</u>
1	<u>atid</u>	Integer(10)	<u>Primary Key</u>
2	<u>name</u>	varchar(100)	<u>NotNull</u>
3	<u>date</u>	date	<u>NotNull</u>
4	<u>poa</u>	Varchar(11)	<u>NotNull</u>
5	<u>status</u>	varchar(100)	<u>NotNull</u>

Table: applyservice**Primary Key: apserid**

<u>Sln</u>	<u>Field Name</u>	<u>Data Type</u>	<u>Constraints</u>
1	<u>apserid</u>	<u>int(10)</u>	<u>Primary Key</u>
2	<u>name</u>	varchar(100)	<u>NotNull</u>
3	<u>mname</u>	varchar(20)	<u>NotNull</u>
4	<u>maidenname</u>	varchar(20)	<u>NotNull</u>
5	<u>address</u>	varchar(100)	<u>NotNull</u>
4	<u>phone</u>	<u>int(10)</u>	<u>NotNull</u>
7	<u>relation</u>	varchar(20)	<u>NotNull</u>
8	<u>dob</u>	date	<u>NotNull</u>
9	<u>place</u>	varchar(10)	<u>NotNull</u>
10	<u>child</u>	varchar(50)	<u>NotNull</u>
11	<u>cname</u>	varchar(20)	<u>NotNull</u>
12	<u>service</u>	varchar(30)	<u>NotNull</u>
13	<u>file1</u>	varchar(255)	<u>NotNull</u>
14	<u>file2</u>	varchar(255)	<u>NotNull</u>
15	<u>file3</u>	varchar(255)	<u>NotNull</u>
16	<u>file4</u>	varchar(255)	<u>NotNull</u>
17	<u>file5</u>	varchar(255)	<u>NotNull</u>
18	<u>status</u>	varchar(10)	<u>NotNull</u>

Table: feedback**Primary Key: fid**

<u>Sln</u>	<u>Field Name</u>	<u>Data Type</u>	<u>Constraints</u>
1	<u>fid</u>	<u>int(10)</u>	<u>Primary Key</u>
2	<u>rate</u>	varchar(100)	<u>NotNull</u>
3	<u>descri</u>	varchar(100)	<u>NotNull</u>

Table: propertytax**Primary Key: ptid**

Sln o	Field Name	Data Type	Constraints
1	ptid	int(10)	Primary Key
2	name	varchar(100)	NotNull
3	address	varchar(100)	NotNull
4	amount	int(20)	NotNull
5	cent	varchar(100)	NotNull
6	total	varchar(100)	NotNull

Table: replay**Primary Key: ptid**

Sln o	Field Name	Data Type	Constraints
1	rid	int(10)	Primary Key
2	replay	varchar(20)	NotNull
3	status	varchar(100)	NotNull

Table: applyscheme**Primary Key: apid**

Sln o	Field Name	Data Type	Constraints
1	apid	int(10)	Primary Key
2	scheme	varchar(30)	NotNull
3	doc1	varchar(20)	NotNull
4	doc2	varchar(20)	NotNull
5	doc3	varchar(20)	NotNull
6	doc4	varchar(20)	NotNull
7	doc5	varchar(20)	NotNull
8	status	varchar(20)	NotNull

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the practice of carefully controlling the execution of software to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and walkthroughs as well. Verifying that what has been specified matches what the user truly desired is the process of validation.

The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code, searching for issues and obtaining statistics without actually running the code. Dynamic analysis examines how software behaves while it is running in order to offer data like execution traces, timing profiles, and test coverage details.

Testing is a collection of activities that can be planned ahead of time and carried out in a methodical manner. Testing starts with individual modules and progresses to the integration of the full computer-based system. There are many rules that can be used as testing objectives, and testing is necessary for the system testing objectives to be successful. As follows:

A program is tested by being run with the goal of identifying any errors.

- A test case with a high likelihood of detecting an unknown fault qualifies as a good testcase.
- A test that finds a mistake that hasn't been noticed is successful.

If a test is successfully carried out in accordance with the aforementioned aims, it will reveal software bugs. Additionally, testing shows that the software functions seem to operate in accordance with the specification and that the performance requirements seem to have been satisfied.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Testing for correctness is meant to ensure that a program performs exactly as it was intended to. This is much harder than it might initially seem, especially for big programs.

5.2 TEST PLAN

A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer program, its documentation, and associated data structures are all created by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfils the purpose for which it was intended. In order to solve the inherent issues with allowing the builder evaluate what they have developed, there is an independent test group (ITG). It is best to describe the precise objective soft testing in quantifiable words. Defect density or frequency of occurrence, cost to detect and correct the problems, remaining test work hours per regression test, and time to failure should all be specified in the test plan.

The levels of testing include:

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

5.2.1 Unit Testing

Unit testing concentrates verification work on the software component or module, which is the smallest unit of software design. The component level design description is used as a guide when testing crucial control paths to find faults inside the module's perimeter. the level of test complexity and the untested area determined for unit testing. The uni testing is white-box oriented, and step can be conducted in parallel for multiple components. To guarantee that data enters and exits the software unit under test properly, the modular interface is tested. To make sure that data temporarily stored retains its integrity during each step of an algorithm's execution, the local data structure is inspected. To confirm thateach statement in a module has been executed at least once, boundary conditions are evaluated. Finally, each path for managing errors is examined. Before starting any additional tests, data flow tests via a module interface must be completed. All other tests are irrelevant if data cannot enter and depart the system properly. An important duty duringthe unit test is the selective examination of execution pathways. Error circumstances must

be expected in good design, and error handling paths must be set up to cleanly reroute or halt work when an error does arise. The final step of unit testing is boundary testing. Software frequently fails at its limits.

In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. All unused code was eliminated, and it was confirmed that every module was functional and produced the desired outcome.

5.2.2 Integration Testing

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a program structure that has been determined by design using unit tested components. The program as a whole is tested. Correction is challenging since the size of the overall program makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All of the modules were incorporated into the system after unit testing was completed in order to check for interface consistency. A distinctive program structure also developed when discrepancies in program structures were eliminated.

5.2.3 Validation Testing or System Testing

This marks the conclusion of the testing procedure. This required comprehensive testing of the system, which covered all forms, codes, modules, and class modules. Popular names for this type of testing include system tests and black box testing.

The functional requirements of the software are the main emphasis of the black box testing approach. That example, using Black Box testing, a software engineer can create sets of input conditions that will fully test every program requirement.

The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors.

5.2.4 Out Testing or User Acceptance Testing

User approval of the system under consideration is tested; in this case, it must meet the needs of the company. When developing, the program should stay in touch with the user and perspective system to make modifications as needed. With regard to the following points, this is done:

- Input Screen Designs,
- Output Screen Designs,

The aforementioned testing is carried out using a variety of test data. The preparation of test data is essential to the system testing process. The system under study is tested using the test data after the test data preparation. Errors in the system are once again found during testing, fixed using the methods described above, and logged for use in the future.

Automation Testing

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it's a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what's been found, and this information can be compared with earlier test runs.

Benefits of Automation Testing

Detailed reporting capabilities - Automation testing uses well-crafted test cases for various scenarios. These scripted can be incredibly in depth, and provide detailed reports that simply wouldn't be possible when done by a human. Improved bug detection - One of the main reasons to test a product is to detect bugs and other defects and other defects. Automation testing makes this process an easier one. It's also able to analyze a wider test coverage than humans may be able to.

- **Simplifies testing** - Testing is a routine part of the operations of most SaaS and tech companies. Making it as simple as possible is key. Using automation is extremely beneficial. When automating test tools, the test scripts can be reused.
- **Speeds up the testing process** - Machines and automated technology work faster than humans. Along with improved accuracy, this is why we use them. In turn, this shortens your software development cycles.
- **Reduces human intervention** - Tests can be run at any time of day, even overnight, without needing humans to oversee it. Plus, when it's conducted automatically, this can also reduce the risk of human error.

5.2.1 Selenium Testing

Selenium is an opensource tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, Node JS, PHP, Perl, Python, and C#, among others. The Selenium testing tool is used to automate tests across browsers for web applications. It's used to ensure high quality web applications whether they are responsive, progressive, or regular. Selenium is an open- source tool.

Project Name: E-Grama Panchayat					
Login Test Case					
Test Case ID: Fun_1			Test Designed By: Ashtami Prasad		
Test Priority (Low/Medium/High): High			Test Designed Date:17-07-2022		
Module Name: Login			Test Executed By: Ms. Shelly Shiju George		
Test Title: Login			Test Execution Date: 18-07-2022		
Description: Login Module					
Pre-Condition: User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Naviga tion to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid usernam e	Username : harish	User should be able to Login	User Logged in and navigated to User Dashboard	Pass
3	Provide Valid Password	Password: harish			
4	Click on Sign In button				
Post Condition:User is Authenticated with database,correctly login to account					

```

1 package test;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.chrome.ChromeDriver;
5 public class login2 {
6     public static void main(String[] args) {
7         System.setProperty("webdriver.chrome.driver", "C:\\Users\\Teena\\Downloads\\chromedriver_win32\\chromedriver.exe" );
8         WebDriver driver=new ChromeDriver();
9         driver.get("http://localhost/grama/login.php");
10        driver.findElement(By.id("uname")).sendKeys("harish");
11        driver.findElement(By.id("psw")).sendKeys("harish");
12        driver.findElement(By.id("signup")).click();
13        String actualUrl="http://localhost/grama/user/index.php";
14        String expectedUrl= driver.getCurrentUrl();
15        if(actualUrl.equalsIgnoreCase(expectedUrl))
16        {
17            System.out.println("Test passed");
18        }
19        else
20        {
21            System.out.println("Test fail");
22        }
23    }
24 }
25
26
27
28

```

```

1 package test1;
2 import org.openqa.selenium.By;
3
4 public class complaint {
5     public static void main(String[] args) {
6         System.setProperty("webdriver.chrome.driver", "C:\\Users\\Hp\\Downloads\\chromedriver_win32\\chromedriver.exe" );
7         WebDriver driver=new ChromeDriver();
8         driver.get("http://localhost/grama/login.php");
9         driver.findElement(By.id("uname")).sendKeys("harish");
10        driver.findElement(By.id("psw")).sendKeys("harish");
11        driver.findElement(By.id("signup")).click();
12        driver.get("http://localhost/grama/User/complaint.php");
13        driver.findElement(By.id("name")).sendKeys("harish");
14        driver.findElement(By.id("email")).sendKeys("harish23@gmail.com");
15        driver.findElement(By.id("phone")).sendKeys("2128588886");
16        driver.findElement(By.id("complaint")).sendKeys("Denial of service");
17        driver.findElement(By.id("description")).sendKeys("services replay not given");
18
19        driver.findElement(By.id("add")).click();
20        String actualUrl="http://localhost/grama/User/complaint.php";
21        String expectedUrl= driver.getCurrentUrl(); //driver.findElement(By.id("qty")).sendKeys("3");
22        //driver.findElement(By.id("cart1")).click();
23        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
24            System.out.println("Test passed");
25        } else {
26            System.out.println("Test failed");
27        }
28    }
29 }
30 }
31
32
33

```

```

<terminated> login2 [Java Application] C:\Users\Teena\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (19-Jul-2022, 12:30:32 pm)
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 51226
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 19, 2022 12:30:37 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 19, 2022 12:30:37 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed

```

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. It can be regarded as the most important stage in creating a successful new system since it gives users assurance that the system will operate as intended and be reliable and accurate. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a new, revised system design into a standalone operation, and it simply refers to convening a new system design into operation. At this point, the user department is responsible for the majority of the workload, the most disruption, and the most influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result. Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system must be implemented properly in order to satisfy organizational needs. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards can it be put into use. The system personnel assess the system's viability. The effort necessary for system analysis and design to implement the three key components of education and training, system testing, and changeover will increase in proportion to how complicated the system being implemented is. The implementation state involves the following tasks:

- Careful planning.
- Examination of system and its limitations
- Designing strategies to implement the transition

6.1 IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. People have early reservations about the software, but we must watch out that they do not become more resistant by making sure that:

- The new system's advantages must be known to the active user.
- Their faith in the program is strengthened.
- The user receives the appropriate instruction so that he feels confident using the application.

Before examining the system, the user must be aware that the server software needs to be running on the server in order to access the results. The actual process won't happen if the server object is not active and functioning on the server.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. The people who will be involved must have faith in their ability to contribute to the goal and benefits anticipated from the computer-based 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

The user will need to receive the necessary basic training on computer awareness after which the new application software will need to be taught to them. This will explain the fundamental principles of how to use the new system, including how the screens work, what kind of help is displayed on them, what kinds of errors are made when entering data, how each entry is validated, and how to change the date that was 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. Depending on the user group and hierarchical level, this training could be different.

6.2.3 System Maintenance

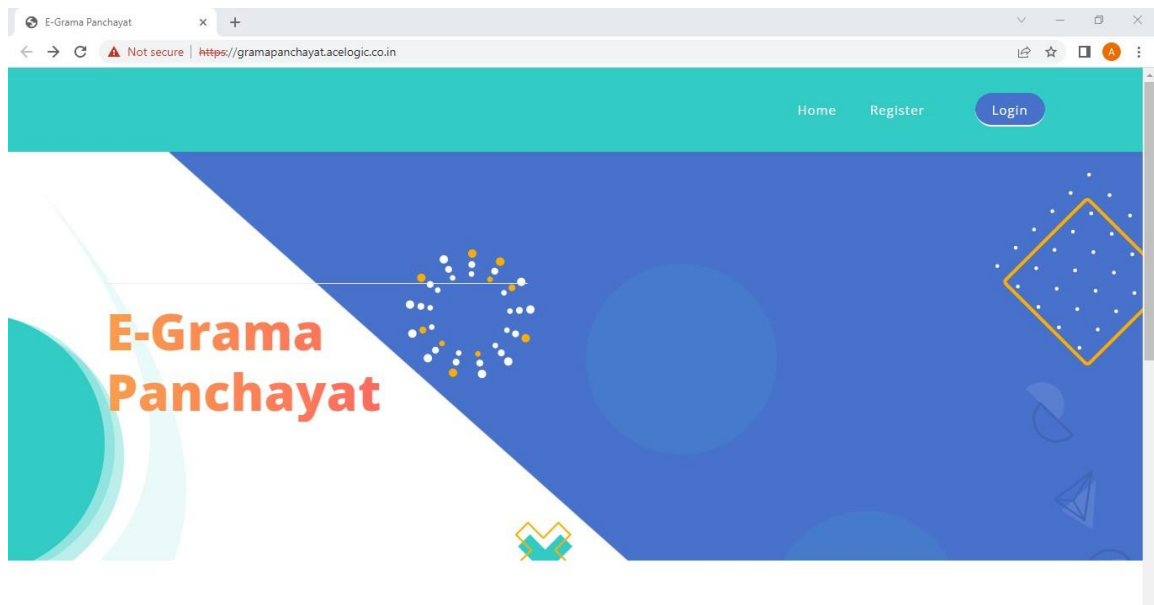
The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software maintenance involves much more than just "Finding Mistakes".

6.2.4 Hosting

000webhost is a free website hosting solution that provides an array of valuable features, including a website builder, WordPress support, and no ads.

Setting up infinity:

1. Create an Account or Log in into your account.
2. Create a Hosting Account.
3. Create a Custom Domain or a Subdomain provided by Infinity Free.
4. Manage your account.
5. Upload your Files
6. Creating Your Database.
7. Changing your PHP connection file configuration.



CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

There is no use of widely utilized technologies like the internet because the system's operating technology is outdated. The suggested system gives the ability for users to browse all information and access services and plans online.

7.2 FUTURE SCOPE

- This application is simple to use in a variety of circumstances, and we may add new functionality as needed.
- Reusability is possible in this application as and when necessary.
- The main future scope of this project is fully automated.
- Mobile Application
- Advanced searching

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”,2009.
- Roger S Pressman, “*Software Engineering*”,1994.
- Pankaj Jalote, “*Software 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:

- 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 SESSION_Start();
$con=mysqli_connect('localhost','root','','grama'); if(isset($_POST['login']))
{
    $uname=$_POST['uname'];
    $psw=$_POST['psw'];
    $q3=mysqli_query($con,"select * from login where uname='$uname' and psw='$psw'");
    if(mysqli_num_rows($q3)>0){
        $row = mysqli_fetch_array($q3);
        $usertype=$row['role'];
        $user=$row['uname']; if($user==$uname)
        {
            echo'<script>alert("login Successfully")</script>'; if($usertype == 0) //User
            {
                #echo $_SESSION['id1'];
                $_SESSION['name'] = $uname;
                header('location:User/index.php');
            }
            if($usertype == 1) //Admi
            {
                echo'<script>alert("login Successfully")</script>'; header('location:Admin/index.php');
            }
            if($usertype == 2) //staff
            {
                $_SESSION['name'] = $uname;
                echo'<script>alert("login Successfully")</script>'; header('location:Staff/index.php');
            }
            if($usertype == 3) //officer
            {
                $_SESSION['name'] = $uname;
                header('location:Officer/index.php');
```

```
}  
if($usertype == 4) //officer  
{  
    $_SESSION['name'] = $uname;  
    header('location:Clerk/index.php');  
  
}  
else  
{  
  
  
}  
}  
}?>  
  
}  
  
header('location:login.php');  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<meta http-equiv="X-UA-Compatible" content="ie=edge">  
<title>E-grama panchayat</title>  
<!-- Font Icon -->  
<link rel="stylesheet" href="fonts/material-icon/css/material-design-iconic-  
font.min.css">  
<!-- Main css -->  
<link rel="stylesheet" href="css/style.css">
```

```
</head>
<body>
<div class="main">
<!-- Sign up form -->
<section class="signup">
<div class="container">
<div class="signup-content">
<div class="signup-form">
<h2 class="form-title">Login Here</h2>
<?php if(isset($_GET['error']))
{
?>
<h3 class="form-title" ><text>Incorrect Username or Password</text></h3>
<?php
}
?>
<form method="POST" action="" class="register-form" id="register-form"
autocomplete="off">
<div class="form-group">
<label for="name"><i class="zmdi zmdi-account material-icons- name"></i></label>
</div>
<div class="form-group">
<label for="email"><i class="zmdi zmdi-email"></i></label>
<input type="text" name="uname" placeholder="Login Id" required>
</div>
<div class="form-group">
<label for="pass"><i class="zmdi zmdi-lock"></i></label>
<input type="password" name="psw" placeholder="Password" required>

</div>

<div class="form-group form-button">
```

```
<input type="submit" name="login" id="signup" class="form-submit"
value="Submit"/><br><br>
```

```
<script>
function myFunction()
{
document.getElementById('btn') alert("Services Added!");
document.getElementById('btn').prop('disabled',true);
}
```

```
</script>
```

```
Register</a>
```

```
<a href="registration.php" style="color:blue">Dont have an account??
```

```
</div>
```

```
</form>
```

```
</div>
```

```
<div class="signup-image">
```

```
<!-- <figure></figure> -->
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- Sing in Form -->
```

```
</div>
```

```
<!-- JS -->
```

```
<script src="vendor/jquery/jquery.min.js"></script>
<script src="js/main.js"></script>
</body><!-- This templates was made by Colorlib (https://colorlib.com) -->
</html>
```

Add Schemes

```
<?php
include 'connect.php';
$date= date('d-m-y');
if(isset($_POST['add'])){
$name=$_POST['name'];
$startdate=$_POST['startdate'];
$enddate=$_POST['enddate'];
$schemedescri=$_POST['schemedescri'];
$document1=$_POST['doc1'];
$document2=$_POST['doc2'];
$document3=$_POST['doc3'];
$document4=$_POST['doc4'];
$document5=$_POST['doc4'];
$quali=$_POST['quali'];
$dup=mysqli_query($conn,"select * from cscheme where name='$name'");
if(mysqli_num_rows($dup)>0){
echo'<script>alert("Scheme Already exist")</script>';
}
else{
$schemeid=mysqli_insert_id($conn);
$sql=mysqli_query($conn,"INSERT INTO `cscheme`(`name`,`start`,`end`,`descri`,`doc1`,`doc2`,`doc3`,`doc4`,`doc5`,`quali`,`status`)values('$name','$startdate','$enddate','$schemedescri','$document1','$document2','$document3','$document4','$document5','$quali','Active')");
echo'<script>alert("Scheme Already exist")</script>'; header('viewschemes.php');
}
}
?>
```



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Admin </title>
<meta content="width=device-width, initial-scale=1.0" name="viewport">
<meta content="" name="keywords">
<meta content="" name="description">
<!-- Favicon -->
<link href="img/favicon.ico" rel="icon">

<!-- Google Web Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Heebo:wght@400;500;600;700&display
=swap" rel="stylesheet">

<!-- Icon Font Stylesheet -->
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet">
<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css"
rel="stylesheet">
<!-- Libraries Stylesheet -->
<link href="lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
<link href="lib/tempusdominus/css/tempusdominus-bootstrap-4.min.css" rel="stylesheet"
/>
<!-- Customized Bootstrap Stylesheet -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<!-- Template Stylesheet -->
<link href="css/style.css" rel="stylesheet">
</head>
<body>
<div class="container-xxl position-relative bg-white d-flex p-0">
```

```

<!-- Spinner Start -->
<div id="spinner" class="show bg-white position-fixed translate-middle w-100 vh-100
top-50 start-50 d-flex align-items-center justify-content-center">
<div class="spinner-border text-primary" style="width: 3rem; height: 3rem;"
role="status">
<span class="sr-only">Loading...</span>
</div>
</div>
<!-- Spinner End -->
<!-- Sidebar Start -->
<div class="sidebar pe-4 pb-3">
<nav class="navbar bg-light navbar-light" style="background-color:blue;">
<a href="index.html" class="navbar-brand mx-4 mb-3">
<h3 class="text-primary">ADMIN</h3>
</a>
<div class="d-flex align-items-center ms-4 mb-4">
<div class="position-relative">
<!--  -->

<!-- <div class="bg-success rounded-circle border border-2 border-white position-
absolute end-0 bottom-0 p-1"></div> -->
</div>
<div class="ms-3">
<!-- <h6 class="mb-0">Jhon Doe</h6> -->
<!-- <span>Admin</span> -->
</div>
</div>
<div class="navbar-nav w-100">
<a href="index.php" class="nav-item nav-link active">Dashboard</a>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs- toggle="dropdown">Services</a>
<div class="dropdown-menu bg-transparent border-0">
<a href="viewservices.php" class="dropdown-item">View Services</a>

```

```
<a href="addservices.php" class="dropdown-item">Create Services</a>
```

```
</div>
```

```
</div>
```

```
<div class="nav-item dropdown">
```

```
<a href="#" class="nav-link dropdown-toggle" data-bs-  
toggle="dropdown">Schemes</a>
```

```
<div class="dropdown-menu bg-transparent border-0">
```

```
<a href="viewschemes.php" class="dropdown-item">View Schemes</a>
```

```
<a href="blokedcschemes.php" class="dropdown-item">View Blocked
```

```
Schemes</a>
```

```
<a href="addschemes.php" class="dropdown-item">Create Schemes</a>
```

```
</div>
```

```
</div>
```

```
<div class="nav-item dropdown">
```

```
<a href="#" class="nav-link dropdown-toggle" data-bs- toggle="dropdown">Staff or  
Officer</a>
```

```
<div class="dropdown-menu bg-transparent border-0">
```

```
<a href="viewstaffofficer.php" class="dropdown-item">View
```

```
Staff/Officer</a>
```

```
<a href="addstaffofficer.php" class="dropdown-item">Create Staff</a>
```

```
<a href="cofficer.php" class="dropdown-item">Create Officer</a>
```

```
</div>
```

```
</div>
```

```
<div class="nav-item dropdown">
```

```
<a href="#" class="nav-link dropdown-toggle" data-bs-
```

```
toggle="dropdown">Complaint</a>
<div class="dropdown-menu bg-transparent border-0">
<a href="viewcomplaint.php" class="dropdown-item">View Complaint</a>
</div>
</div>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs-
toggle="dropdown">Attendance</a>
<div class="dropdown-menu bg-transparent border-0">
<a href="viewattendance.php" class="dropdown-item">View Attendance</a>
</div>
</div>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs- toggle="dropdown">Leave</a>
<div class="dropdown-menu bg-transparent border-0">
<a href="viewleave.php" class="dropdown-item"> Leave Applications</a>
<a href="viewapprovedleave.php" class="dropdown-item">Approved
Leave</a> Leave</a>

<a href="viewrejectedleave.php" class="dropdown-item">Rejected

</div>
</div>

<div class="nav-item dropdown">
<a href="logout.php" class="nav-link dropdown-toggle" data-bs-
toggle="dropdown">Logout</a>
<div class="dropdown-menu bg-transparent border-0">
<a href=" ../login.php" class="dropdown-item">Logout</a>
</div>
</div>
```

```
</div>
</nav>
</div>
<!-- Sidebar End -->
<!-- Content Start -->
<div class="content">
<!-- Navbar Start -->
<nav class="navbar navbar-expand bg-light navbar-light sticky-top px-4 py-0">
<a href="index.html" class="navbar-brand d-flex d-lg-none me-4">

<h2 class="text-primary mb-0"><i class="fa fa-hashtag"></i></h2>

</a>
<a href="#" class="sidebar-toggler flex-shrink-0">
<i class="fa fa-bars"></i>
</a>
<form class="d-none d-md-flex ms-4">
<input class="form-control border-0" type="search" placeholder="Search">
</form>
<div class="navbar-nav align-items-center ms-auto">
<div class="nav-item dropdown">
<!-- <a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown"> -->
<div class="dropdown-menu dropdown-menu-end bg-light border-0 rounded-0 rounded-
bottom m-0">
<a href="#" class="dropdown-item">
<div class="d-flex align-items-center">


<div class="ms-2">
<h6 class="fw-normal mb-0">Jhon send you a message</h6>
<small>15 minutes ago</small>
```

```
</div>

</div>
</a>
<hr class="dropdown-divider">
<a href="#" class="dropdown-item">
<div class="d-flex align-items-center">


<div class="ms-2">
<h6 class="fw-normal mb-0">Jhon send you a message</h6>
<small>15 minutes ago</small>
</div>

</div>
</a>
<hr class="dropdown-divider">
<a href="#" class="dropdown-item">
<div class="d-flex align-items-center">

<div class="ms-2">
<h6 class="fw-normal mb-0">Jhon send you a message</h6>

<small>15 minutes ago</small>
</div>
</div>
</a>
<hr class="dropdown-divider">
<a href="#" class="dropdown-item text-center">See all message</a>
</div>
```

```

</div>
<div class="nav-item dropdown">
<div class="dropdown-menu dropdown-menu-end bg-light border-0 rounded-0 rounded-
bottom m-0">
</div>
</div>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">

<span class="d-none d-lg-inline-flex">Admin</span>
</a>
</div>
</div>
</nav>
<!-- Navbar End -->
<!-- Form Start -->
<div class="container-fluid pt-4 px-4">
<div class="row g-4">
<div class="col-sm-12 col-xl-6">
</div>
<div class="col-sm-12 col-xl-6">
</div>
<div class="col-sm-12 col-xl-6">
<div class="bg-light rounded h-100 p-4">
<form action="" method="POST" autocomplete="off" >
<h6 class="mb-4">Schemes Add</h6>
<div class="form-floating mb-3">
Scheme Name<input type="text" name="name" id="1" class="form- control" onchange
="ValidName();"
>
<label for="floatingInput"></label>
<span id="msg1" style="color:red;"></span>

```

```

<script>
function ValidName()
{
var val = document.getElementById('1').value;

if (!val.match(/^[A-Z][A-Za-z]{3,}$/))
{
document.getElementById('msg1').innerHTML="Start with a Capital letter & Only
alphabets are allowed!!";
document.getElementById('1').value = ""; return false;
}
document.getElementById('msg1').innerHTML=" "; return true;
}
</script>
</div>
<div class="form-floating mb-3">
Start Date<input type="date" name="startdate" class="form-control"

id="floatingInput"

placeholder="name@example.com" min="2022-09-07" >
<label for="floatingInput"></label>

</div>
<div class="form-floating mb-3">
End Date<input type="date" name="enddate" class="form-control"

id="floatingInput" 07">

placeholder="name@example.com" min="2022-09-07" max="2022-19-

<label for="floatingInput"></label>

```

```
</div>
<div class="form-floating">
Description<textarea name ="schemedescri" class="form-control" placeholder="Leave a
comment here"
id="floatingTextarea" style="height: 150px;"></textarea>
<div class="form-floating mb-3">
Required Document1<input type="name" name="document1" id="name" class="form-
control" id="floatingInput" onchange ="ValidName();">
<label for="floatingInput"></label>

</div>

<div class="form-floating mb-3">
Required Document2<input type="name" name="document2" id="name"

class="form-control" id="floatingInput" onchange ="ValidName();">
<label for="floatingInput"></label>

</div>

<div class="form-floating mb-3">
Required Document3<input type="name" name="document3"

id="name" class="form-control" id="floatingInput" onchange ="ValidName();">
<label for="floatingInput"></label>

</div>

<div class="form-floating mb-3">
```

Required Document4<input type="name" name="document4"

id="name" class="form-control" id="floatingInput" onchange ="ValidName();">

<label for="floatingInput"></label>

</div>

<div class="form-floating mb-3">

Required Document5<input type="name" name="document5"

id="name" class="form-control" id="floatingInput" onchange ="ValidName();">

<label for="floatingInput"></label>

</div>

id="floatingInput"

<label for="floatingTextarea"></label>

<div class="form-floating mb-3">

Qualification<input type="name" name="quali" class="form-control"

placeholder="name@example.com">

<label for="floatingInput"></label>

</div>

<button type="submit" name="add" class="btn btn-primary" >Add

Scheme</button>

function myFunction()

```
{

<script>

document.getElementById('btn') alert("Services Added!");
document.getElementById('btn').prop('disabled',true);
}

</script>

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

<!-- JavaScript Libraries -->

<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"></scrip
t>
<script src="lib/chart/chart.min.js"></script>
<script src="lib/easing/easing.min.js"></script>
<script src="lib/waypoints/waypoints.min.js"></script>
<script src="lib/owlcarousel/owl.carousel.min.js"></script>

<script src="lib/tempusdominus/js/moment.min.js"></script>
<script src="lib/tempusdominus/js/moment-timezone.min.js"></script>
<script src="lib/tempusdominus/js/tempusdominus-bootstrap-4.min.js"></script>

<!-- Template Javascript -->
```

```
<script src="js/main.js"></script>
</body>

</html>
```

Add Services.php

```
<?php
include 'connect.php'; session_start(); if(isset($_POST['add'])){
$name_error="";
$name=$_POST['name'];
$servicedescr=$_POST['desc'];
$document1=$_POST['document1'];
$document2=$_POST['document2'];
$document3=$_POST['document3'];
$document4=$_POST['document4'];
$document5=$_POST['document5'];

$dup=mysqli_query($conn,"select * from cservice where name='$name'");
if(mysqli_num_rows($dup)>0){
echo'<script>alert("Services Already exist")</script>';

}
else{
$service_id=mysqli_insert_id($conn);

$sql=mysqli_query($conn,"INSERT INTO `cservice`(`name`,`description`,`doc1`,`doc2`,`doc3`,`doc4`,`doc5`,`status`)values('$name','$servicedescr','$document1','$document2','$document3','$document4','$document5','Active')");
echo'<script>alert("Services Added!")</script>'; header('viewservices.html');
}
}
```

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Admin </title>

<meta content="width=device-width, initial-scale=1.0" name="viewport">

<meta content="" name="keywords">

<meta content="" name="description">

<!-- Favicon -->

<link href="img/favicon.ico" rel="icon">

<!-- Google Web Fonts -->

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link

href="https://fonts.googleapis.com/css2?family=Heebo:wght@400;500;600;700&display=swap" rel="stylesheet">

<!-- Icon Font Stylesheet -->

<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet">

<!-- Libraries Stylesheet -->

<link href="lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">

<link href="lib/tempusdominus/css/tempusdominus-bootstrap-4.min.css" rel="stylesheet"

/>

<!-- Customized Bootstrap Stylesheet -->

```
<link href="css/bootstrap.min.css" rel="stylesheet">

<!-- Template Stylesheet -->
<link href="css/style.css" rel="stylesheet">
</head>

<body>
<div class="container-xxl position-relative bg-white d-flex p-0">
<!-- Spinner Start -->
<div id="spinner" class="show bg-white position-fixed translate-middle w-100 vh-100
top-50 start-50 d- flex align-items-center justify-content-center">
<div class="spinner-border text-primary" style="width: 3rem; height: 3rem;"
role="status">
<span class="sr-only">Loading...</span>
</div>
</div>
<!-- Spinner End -->
<!-- Sidebar Start -->
<div class="sidebar pe-4 pb-3">
<nav class="navbar bg-light navbar-light" style="background-color:blue;">
<a href="index.html" class="navbar-brand mx-4 mb-3">
<h3 class="text-primary">ADMIN</h3>
</a>
<div class="d-flex align-items-center ms-4 mb-4">
<div class="position-relative">

<!--  -
->

<!-- <div class="bg-success rounded-circle border border-2 border-white position-
absolute end-0 bottom-0 p-1"></div> -->
</div>
<div class="ms-3">
```

```
<!-- <h6 class="mb-0">Jhon Doe</h6> -->
<!-- <span>Admin</span> -->
</div>
</div>
<div class="navbar-nav w-100">
<a href="index.php" class="nav-item nav-link active">Dashboard</a>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Services</a>
<div class="dropdown-menu bg-transparent border-0">
<a href="viewservices.php" class="dropdown-item">View Services</a>
<a href="addservices.php" class="dropdown-item">Create Services</a>

</div>
</div>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Schemes</a>
<div class="dropdown-menu bg-transparent border-0">
<a href="viewschemes.php" class="dropdown-item">View Schemes</a>
<a href="blokedschemas.php" class="dropdown-item">View Blocked Schemes</a>
<a href="addschemes.php" class="dropdown-item">Create Schemes</a>

</div>
</div>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Staff or
Officer</a>
<div class="dropdown-menu bg-transparent border-0">
<a href="viewstaffofficer.php" class="dropdown-item">View Staff/Officer</a>
<a href="addstaffofficer.php" class="dropdown-item">Create Staff</a>
<a href="cofficer.php" class="dropdown-item">Create Officer</a>

</div>
</div>
<div class="nav-item dropdown">
```

```
<a href="#" class="nav-link dropdown-toggle" data-bs-  
toggle="dropdown">Complaint</a>  
<div class="dropdown-menu bg-transparent border-0">  
<a href="viewcomplaint.php" class="dropdown-item">View Complaint</a>  
  
</div>  
</div>  
  
<div class="nav-item dropdown">  
<a href="#" class="nav-link dropdown-toggle" data-bs-  
toggle="dropdown">Attendance</a>  
<div class="dropdown-menu bg-transparent border-0">  
<a href="viewattendance.php" class="dropdown-item">View Attendance</a>  
  
</div>  
</div>  
  
<div class="nav-item dropdown">  
<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">Leave</a>  
<div class="dropdown-menu bg-transparent border-0">  
<a href="viewleave.php" class="dropdown-item"> Leave Applications</a>  
  
<a href="viewapprovedleave.php" class="dropdown-item">Approved Leave</a>  
<a href="viewrejectedleave.php" class="dropdown-item">Rejected Leave</a>  
  
</div>  
</div>  
  
<div class="nav-item dropdown">  
<a href="logout.php" class="nav-link dropdown-toggle" data-bs-  
toggle="dropdown">Logout</a>  
<div class="dropdown-menu bg-transparent border-0">  
<a href="..login.php" class="dropdown-item">Logout</a>  
  
</div>
```

</div>

</div>

</nav>

</div>

<!-- Sidebar End -->

<!-- Content Start -->

<div class="content">

<!-- Navbar Start -->

<nav class="navbar navbar-expand bg-light navbar-light sticky-top px-4 py-0">

<h2 class="text-primary mb-0"><i class="fa fa-hashtag"></i></h2>

<i class="fa fa-bars"></i>

<input class="form-control border-0" type="search" placeholder="Search">

</form>

<div class="navbar-nav align-items-center ms-auto">

<div class="nav-item dropdown">

<!-- -->

m-0">

40px;">

<div class="dropdown-menu dropdown-menu-end bg-light border-0 rounded-0 rounded-bottom

<div class="d-flex align-items-center">

<h6 class="fw-normal mb-0">Jhon send you a message</h6>

<small>15 minutes ago</small>

</div>

</div>

<hr class="dropdown-divider">

<div class="d-flex align-items-center">

40px;">

40px;">

40px;">

<h6 class="fw-normal mb-0">Jhon send you a message</h6>

<small>15 minutes ago</small>

</div>

```
</div>
</a>
<hr class="dropdown-divider">
<a href="#" class="dropdown-item">
<div class="d-flex align-items-center">

<h6 class="fw-normal mb-0">Jhon send you a message</h6>
<small>15 minutes ago</small>
</div>
</div>
</a>
<hr class="dropdown-divider">
<a href="#" class="dropdown-item text-center">See all message</a>
</div>
</div>

<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">
Admin</span>
</a>

</div>
</div>
</nav>
<!-- Navbar End -->

<!-- Form Start -->
```

```

<div class="container-fluid pt-4 px-4">
<div class="row g-4">
<div class="col-sm-12 col-xl-6">

</div>
<div class="col-sm-12 col-xl-6">

</div>
<div class="col-sm-12 col-xl-6">
<div class="bg-light rounded h-100 p-4">
<form action="" method="POST" autocomplete="off" >
<h6 class="mb-4">Service Add</h6>
<?php if(isset($name_error)) { ?>
<p id="res3"><?php echo $name_error; ?></p><?php } ?>
<div class="form-floating mb-3">
Service Name<input type="name" name="name" id="name" class="form-control"

id="floatingInput" onchange ="ValidName();">
<label for="floatingInput"></label>

style="color:red;"></span>
<script>
function ValidName()
{

<span id="msg1"

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 are allowed!!";
document.getElementById('name').value = ""; return false;
}
document.getElementById('msg1').innerHTML=" "; return true;
}
</script>

```

```

</div>

```

```

<div class="form-floating">
<br><br> Description<textarea class="form-control" placeholder="Leave a comment
here" id="place" name="desc" style="height: 150px;" onchange
="ValidPlace();"></textarea>
<span id="msg2"
style="color:red;"></span>
<script>
function ValidPlace()
{
var val = document.getElementById('place').value;

if (!val.match(/^[A-Z][A-Za-z]{3,}$/))
{
document.getElementById('msg2').innerHTML="Start with a Capital letter & Only
alphabets are allowed!!";
document.getElementById('place').value = ""; return false;
}
document.getElementById('msg2').innerHTML=" "; return true;
}
</script>
</div>
<div class="form-floating mb-3">
Required Document1<input type="text" name="document1" id="name" class="form-

```

```
control" id="floatingInput" onchange="ValidName();">
<label for="floatingInput"></label>
```

```
</div>
```

```
<div class="form-floating mb-3">
```

```
Required Document2<input type="name" name="document2" id="name" class="form-
```

```
control" id="floatingInput" onchange="ValidName();">
```

```
<label for="floatingInput"></label>
```

```
</div>
```

```
<div class="form-floating mb-3">
```

```
Required Document3<input type="name" name="document3" id="name" class="form-
```

```
control" id="floatingInput" onchange="ValidName();">
```

```
<label for="floatingInput"></label>
```

```
</div>
```

```
<div class="form-floating mb-3">
```

```
Required Document4<input type="name" name="document4" id="name" class="form-
```

```
control" id="floatingInput" onchange="ValidName();">
```

```
<label for="floatingInput"></label>
```

```
</div>
```

```
<div class="form-floating mb-3">
Required Document5<input type="name" name="document5" id="name" class="form-
control" id="floatingInput" onchange="ValidName();">
<label for="floatingInput"></label>

</div>

<label for="floatingTextarea"></label><br>

<button type="submit" class="btn btn-primary" name="add" >Add Services</button>

<script> function myFunction()
{

document.getElementById('btn') alert("Services Added!");
document.getElementById('btn').prop('disabled',true);
}
</script>
</div>

</form>

</div>
</div>

<!-- JavaScript Libraries -->
<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"></scrip
```

```
t>
<script src="lib/chart/chart.min.js"></script>
<script src="lib/easing/easing.min.js"></script>
<script src="lib/waypoints/waypoints.min.js"></script>
<script src="lib/owlcarousel/owl.carousel.min.js"></script>
<script src="lib/tempusdominus/js/moment.min.js"></script>
<script src="lib/tempusdominus/js/moment-timezone.min.js"></script>
<script src="lib/tempusdominus/js/tempusdominus-bootstrap-4.min.js"></script>

<!-- Template Javascript -->
<script src="js/main.js"></script>
</body>

</html>
```

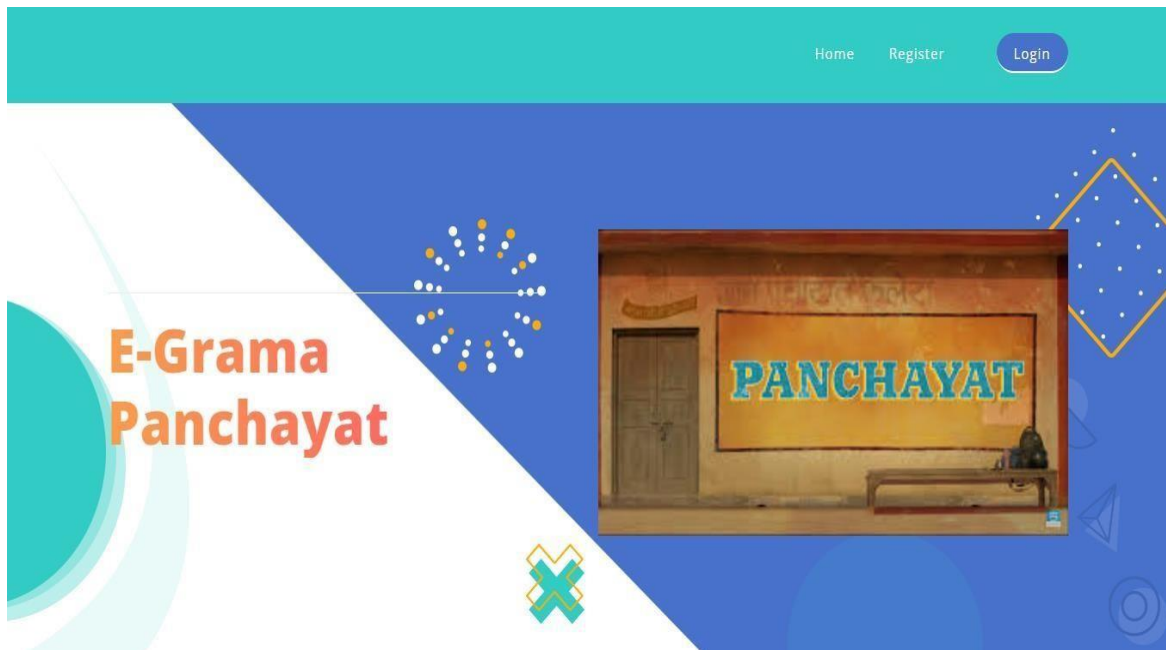
Removeschemes.php

```
<?php
include 'connect.php'
?>

<?php
$Sid1 = $_GET['sid'];
$del = mysqli_query($conn,"delete from cservice where sid = '$Sid1'"); if($del)
{
    mysqli_close($conn); header("location:viewservices.php"); exit;
}
else
{
    header("location:viewservices.php");
}
?>
```


9.2 ScreenShots

Home page



Add Services page

The screenshot shows the 'Add Services' page within an ADMIN interface. The top navigation bar includes the word 'ADMIN', a search bar, and a user profile icon labeled 'Admin'. A sidebar on the left lists various menu items: 'Dashboard', 'Services', 'Schemes', 'Staff or Officer', 'Complaint', 'Attendance', 'Leave', and 'Logout'. The main content area is titled 'Service Add' and contains three input fields: 'Service Name', 'Description', and 'Required Document1'. Below these, there is another input field labeled 'Required Document2'.

Manage Services page

ADMIN

Dashboard

Services

Schemes

Staff or Officer


Complaint

Leave

Attendance

Log Out

Search

 Admin

Recent Services

Show All

Name	Description	Status
Death Certificate	Certificate for death	<div>Edit</div>
Birth Certificate	Certificate for new child	<div>Edit</div>

Add Scheme page

ADMIN

Dashboard

Services

Schemes

Staff or Officer


Complaint

Attendance

Leave

Logout

Search

 Admin

Schemes Add

Scheme Name

Start Date

mm/dd/yyyy

End Date


mm/dd/yyyy

Description

Manage Scheme page

ADMIN

Search

 Admin

Dashboard

Services

Schemes

Staff or Officer

Complaint

Leave

Attendance

Log Out

Recent Scheme


Show All

Name	Description	Action
Flood	Foe heavy rain	<div>Block</div> <div>Edit</div>

Add Staff page

ADMIN

Search

 Admin

Dashboard

Services

Schemes

Staff or Officer

Complaint

Attendance

Leave

Logout

Add Staff / Officer

Name

Address

Phone

Email

Date of Joining


mm/dd/yyyy

Gender ☐ Male ☐ Female

Manage Complaint page

ADMIN

Search

 Admin ▾

Dashboard

Services ▾

Schemes ▾

Staff or Officer ▾

Complaint ▾

Leave ▾

Attendance ▾

Log Out

Recent Complaint

Show All

Name	Complaint	Description	Action
Ammu	Denial of Services	Services replay is not given	<div>Reply</div>

Pay Tax

Darvik

Search

Dashboard

Services ▾

Schemes ▾

Staff or Officer ▾

Complaint ▾

Leave ▾

Log Out

Property Tax

Name

Darvik

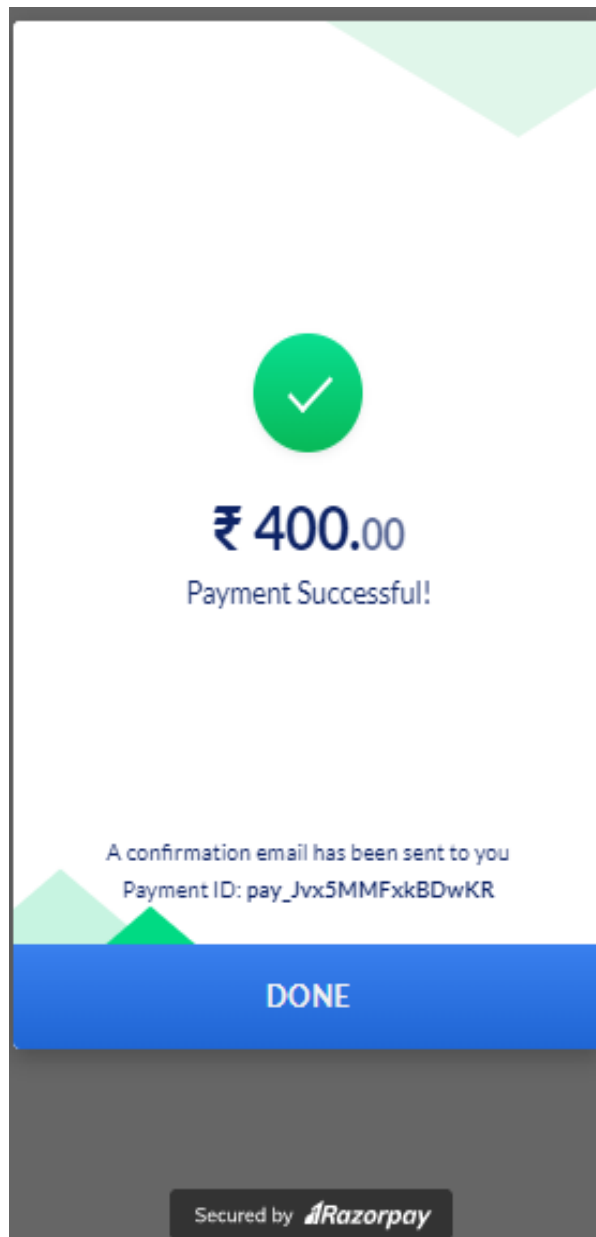
Address

kannuk

Cent

20

Calculate Amount

Pay Amount



Similarity Report ID: oid:10159:20025863

PAPER NAME

Doc15.pdf

AUTHOR

Ashtami Prasad

WORD COUNT

8493 Words

CHARACTER COUNT

46019 Characters

PAGE COUNT

56 Pages

FILE SIZE

1.2MB

SUBMISSION DATE

Jul 22, 2022 2:21 PM GMT+5:30

REPORT DATE

Jul 22, 2022 2:22 PM GMT+5:30**● 18% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 12% Internet database
- 6% Publications database
- Crossref database

● Excluded from Similarity Report

- Small Matches (Less than 20 words)

Summary



Similarity Report ID: oid:10159:20025863

18% Overall Similarity

Top sources found in the following databases:

- 12% Internet database
- 6% Publications database
- Crossref database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	G.S. Ajith, M.G. Girija, Jinson Devis. "Poly House Environment Monitori...	6%
	Crossref	
2	globalapptesting.com	3%
	Internet	
3	issuu.com	3%
	Internet	
4	coursehero.com	2%
	Internet	
5	ro.scribd.com	<1%
	Internet	
6	2019carscomingout.com	<1%
	Internet	
7	geeksforgeeks.org	<1%
	Internet	
8	docshare.tips	<1%
	Internet	
9	slideshare.net	<1%
	Internet	

Sources overview



Similarity Report ID: oid:10159:20025863

10	engineeringminiprojects.com	<1%
	Internet	
11	researchpublish.com	<1%
	Internet	
12	digital.com	<1%
	Internet	

[Sources overview](#)