EMPLOYMENT EXCHANGE AUTOMATION SYSTEM

Project Report Submitted By

AVANI P A

Reg. No: AJC20MCA-2032

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

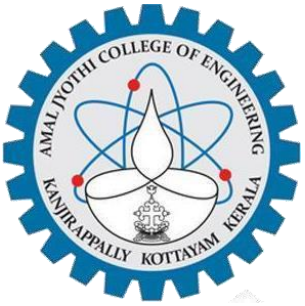
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2020-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, "**EMPLOYMENT EXCHANGE AUTOMATION SYSTEM"** is the bonafide work of **AVANI P A (Reg.No:AJC20MCA- 2032)** 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. Anit James
Internal Guide

Ms. Nimmy Francis
Coordinator

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

External Examiner

DECLARATION

I hereby declare that the project report **"EMPLOYMENT EXCHANGE AUTOMATION SYSTEM"** 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:21-07-2022 **AVANI P A**

KANJIRAPPALLY **Reg. No: AJC20MCA-2032**

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**. **Anit James** 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.

AVANI P A

ABSTRACT

The main objective of employment exchange automation system is for better performance of employment exchange services which are fully automated. Through fully automated system the users can easily get their services and schemes without time waste. The user's account is verified after their registration. So, they can access their services and scheme according through their eligibility criteria. They users can also know their seniority after their registration. Users can download their certificate. And also, officers and can download their salary slip. The Employment Exchange Automation System helps job seekers find employment. Employers are sponsored to hire candidates who have registered with an automation system for the employment exchange for positions that have been announced or advertised.

.

CONTENT

List of Abbreviation

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

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The Employment Exchange Automation System project is for the better performance of employment exchange service and to manage those services through computerized system. And monitoring the activities. The automated employment exchange system helps job seekers find jobs. Employers sponsor candidates who have registered with an employment exchange system to fill open positions that have been announced or advertised.

1.2 PROJECT SPECIFICATION

These issues can be resolved by the proposed system. The process is more easier an their is no expense is required and people can know about the services, scheme without going to the employment exchange system. And get their services and scheme through onlinemode. Also they can apply through online without going to employment exchange system. Notifications of the jobs are made available for users. Staff can verify the documents through online and approve the application. The system includes 5 modules. They are:

➢Admin:

- Login
- Mange services /schemes
- Manage staff and officers
- Manage attendance
- Manage leave details
- Salary for Officers and Staff
- Manage Notification/Alerts

➢ Users:

- Register
- Login
- Account verification
- View services and schemes
    a) Apply for loan
    b) Vocational Guidance
    c) Self Employment
    d) Coaching cum-Guidance
    e) Guidance for scheduled caste and scheduled tribes

f) Certificate registration

- Apply for services and schemes
- Manage job offers
- Manage Profile
- Apply for new request
- Download certificates

➢Staff:

- Login
- View application
- Process Application
- View salary slip
- Generate Reports
- Manage attendance
- Apply leave
- Manage Profile

➢Officers:

- Login
- View new request from user
- Verify and Approve application
- Manage Profile
- Apply leave
- View salary slip
- Approve for certificate
- Generate report

➢ Recruiters(Private /Public)

- Register and login
- Vacancy details
- Manage jobseekers

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is the process of acquiring and analyzing data, diagnosing issues, and using the data to suggest system changes. The system users and system developers must communicate extensively during this problem-solving process. Any system development process should start with a system analysis or research. The system is meticulously examined and assessed. The system analyst takes on the role of the interrogator and probes deeply into how the current system functions. The input to the system is identified, and the system is seen as a whole. The various procedures can be correlated to the outputs from the organization. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

The process must be thoroughly studied using a variety of methodologies, including questionnaires and interviews. To reach a conclusion, the information gathered by these sources must be carefully examined. Understanding how the system works is the conclusion. The current system is the name of this system. Now that the issue with the current system has been thoroughly examined, trouble spots have been located. The designer now acts as a problem-solver and works to resolve the issues the business is having. Proposals are made in place of the solutions. The proposal is then analytically compared to the current system, and the best one is chosen. The user is given the opportunity to approve or reject the suggestion. The proposal is reviewed on user requestand suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

The process of acquiring and analyzing data in order to use it for future system studies is known as preliminary study. Initial research is a problem-solving activity that necessitates close coordination between system users and developers. It conducts a number of feasibility studies. These investigations provide an approximate estimate of the system activities, which can be used to determine the tactics to be used for an efficient system research and analysis.

In this system study the whole system study has been taken from the officer who is working in the district Employment Exchange Office Kattappana.Through this system study lots of data’s and information are collected.

2.2 EXISTING SYSTEM

The existing system have the disadvantages such as time consuming and expensive. The candidate must go to their respective employment exchange system to get their services or scheme. The proposed system rectify the drawbacks of the present system. The current system must be changed in order to include new data, increase its efficiency, and make it more adaptable and secure.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Human effort is required.
- It is challenging to keep crucial information in the user.
- More manual hours need to process application.

2.4 PROPOSED SYSTEM

The proposed system is intended to address every drawback of the current system. For the Employment Exchange System, it is essential to create a more user-friendly and visually striking system.. In this proposed system we can time consuming can be reduced. It makes the process more easier and there is no expense is required and people can know about the services without entering into the Employment Exchange System.

2.5 ADVANTAGES OF PROPOSED SYSTEM

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

- **More secure:-**

Unauthorized access must be prevented in order for data to remain secure. Data protection means that they are shielded from different types of erasure.

- **Ensure data accuracy:-**

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

- **Better service: -**

The Employment Exchange Automation system provide better services and scheme. Also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is conducted to determine whether the project will, upon completion, fulfil the objectives of the organization in relation to the work, effort, and time invested in it. A feasibility study enables the developer to predict the project's usefulness and potential future. An evaluation of a system proposal's viability takes into account the system's impact on the organization, user needs, and efficient use of resources. As a result, before a new application is accepted for development, it often undergoes a feasibility assessment. The document provides the feasibility of the project that is being designed and lists various are asthat were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economic Feasibility

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

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

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

The cost of project Employment Exchange Automation System was divided according to the system used, its development cost. According to all the calculations the project was developed in alow cost. As it is completely developed using open-source software.

3.1.2 Technical Feasibility

First, a technical assessment of the system is required. Based on an outline design of the system requirement in terms of input, output, programs, and procedures, the assessment of this feasibility must be made. After identifying a general system, the study must next recommend the kind of equipment, necessary procedures for constructing the system, and ways to operate the system once it has been designed. Technical issues raised during the investigation are:

- Does the current technology allow for the suggested solutions?
- Can the system expand if it is improved?

The project should be created in a way that achieves the required performance and functions while staying within the bounds. Even if the technology could become outdated after a while, the system can still be used because newer versions of the same software work with previous versions. There are therefore few restrictions associated with this undertaking. The system's front end and back end have both been constructed using PHP, and the project can be developed technically. The system's Pentium processor, 4GB of RAM, and 500GB of hard drive all performed well.

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

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

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible. There is sufficient support for users. Without going to Employment Exchange, we can get the services and scheme according to our needs.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Pentium

RAM - 4 GB

Hard disk - 500 GB

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

3.3.2 MySQL

Oracle Corporation is responsible for the development, distribution, and maintenance of MySQL, the most widely used Open-Source SQL database management system. The most recent information on MySQL software can be found on the MySQL website.

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

A structured data collection is known as a database. A simple grocery list, a photo gallery, or the enormous volume of data on a business network might all be examples. You require a database management system like MySQL Server in order to add, access, and process data kept in a computer database. Database management systems are essential to computing because computers are excellent at handling vast volumes of data, whether as standalone utility or as a component of other programs.

- **MySQL databases are relational.**

As opposed to placing all the data in one huge warehouse, a relational database keeps the data in individual tables. Physical files with speed optimization have been created from the database structures. With objects like databases, tables, views, rows, and columns, the logical model provides a flexible programming environment. You set up rules defining the connections between various data fields, such as "pointers" between various tables and one-to-one, one-to-many, unique, necessary, and optional relationships. With a well-designed database, your application won't ever encounter inconsistent, duplicate, orphan, out-of-date, or missing data since the database enforces these rules. "Structured Query Language," or SQL, is what "MySQL" stands for. Database access is typically done using SQL, a standardized language. Depending on your programming environment, you may enter SQL directly (for instance, to generate reports), incorporate SQL statements into other languages' code, or use a language-specific API that obscures the SQL syntax. The ANSI/ISO SQL Standard provides a definition of SQL. Since it was first created in 1986, the SQL standard has undergone many changes. The terms "SQL92" and "SQL:1999" in this document refer to the standards released in 1992, 1999, and 2003, respectively. "SQL:2003" refers to the standard's most recent revision. The SQL Standard that is in effect at any one time is referred to as "the SQL standard" in this document.

- **MySQL software is Open Source.**

Anyone can use and modify software that is considered open source. The MySQL software is available for free download from the Internet and is usable by anyone. You can examine the source code and modify it as necessary if you so choose. The GPL (GNU General Public License) is used by the MySQL software to specify when it is permitted and prohibited to download the program. You can purchase a commercially likened version from us if you don't like the GPL or need to integrate MySQL code into a business application. More details can be found in the MySQL Licensing Overview.

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

You should give it a try if that's what you're searching for. On a desktop or laptop, MySQL Server can run without much oversight alongside your other programs, web servers, and other devices. If you use a whole system for MySQL, you can change the settings to utilize all the RAM, CPU power, and I/O capacity available.

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

The MySQL Database Software is a client/server system that includes a multi-threaded SQL server that supports many back ends, a number of unique client program and libraries, administrative tools, and a broad variety of application programming interfaces (APIs). In order to create a smaller, faster, more manageable, and standalone project, we now supply MySQL Server as an embedded multi-threaded library that you can connect into your program.

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. The process of using different methodologies and concepts to specify a process or a system in enough detail to allow for its physical implementation is referred to as "design." It may be described as the process of using different ideas and approaches to specify a device, processor, or system in too little detail to allow for its physical implementation. Regardless of the development paradigm that is employed, software design forms the technical core of the software engineering process. A system or product's architectural details are developed as part of the system design. This program, like any systematic technique, has undergone the best design phase possible, fine refining all efficiency, performance, and accuracy levels. A user-oriented document is transitioned during the design phase into a document for programmers or database staff. Logical and physical design are the two stages of system development.

4.2 UML DIAGRAM

Software system artefacts can be specified, visualized, built, and documented using the UML standard language. The Object Management Group (OMG) is the organization that developed UML, and the OMG received a draught of the UML 1.0 definition in January 1997.

Unified Modeling Language, or UML, is different from other popular programming languages like C++, Java, COBOL, etc. As a general-purpose visual modelling language to visualize, specify, construct, and record software systems, UML is a pictorial language used to create software blueprints. 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. Object oriented analysis and design are directly related to UML. UML has been standardized and is now an OMG standard. An entire UML diagram representing a system is created using all the elements and linkages. The most crucial aspect of the entire process is how the UML diagram looks. The following nine diagrams are utilized along with all the other components to create it.

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

4.2.1 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between system components. A approach for identifying, outlining, and organizing system requirements is called a use case. The word "system" here refers to a thing that is being created or run, like a website for mail-order product sales and services. In UML (Unified Representing Language), a common notation for modelling actual items and systems, use case diagrams are used.

The planning of overall requirements, validating a hardware design, testing and debugging a software product in development, producing an online help reference, or carrying out a customer-service-focused activity are some examples of system objectives. Items ordering, catalogue updates, payment processing, and customer relations are a few examples of use cases in a product sales system. Four elements make up a use case diagram.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

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

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

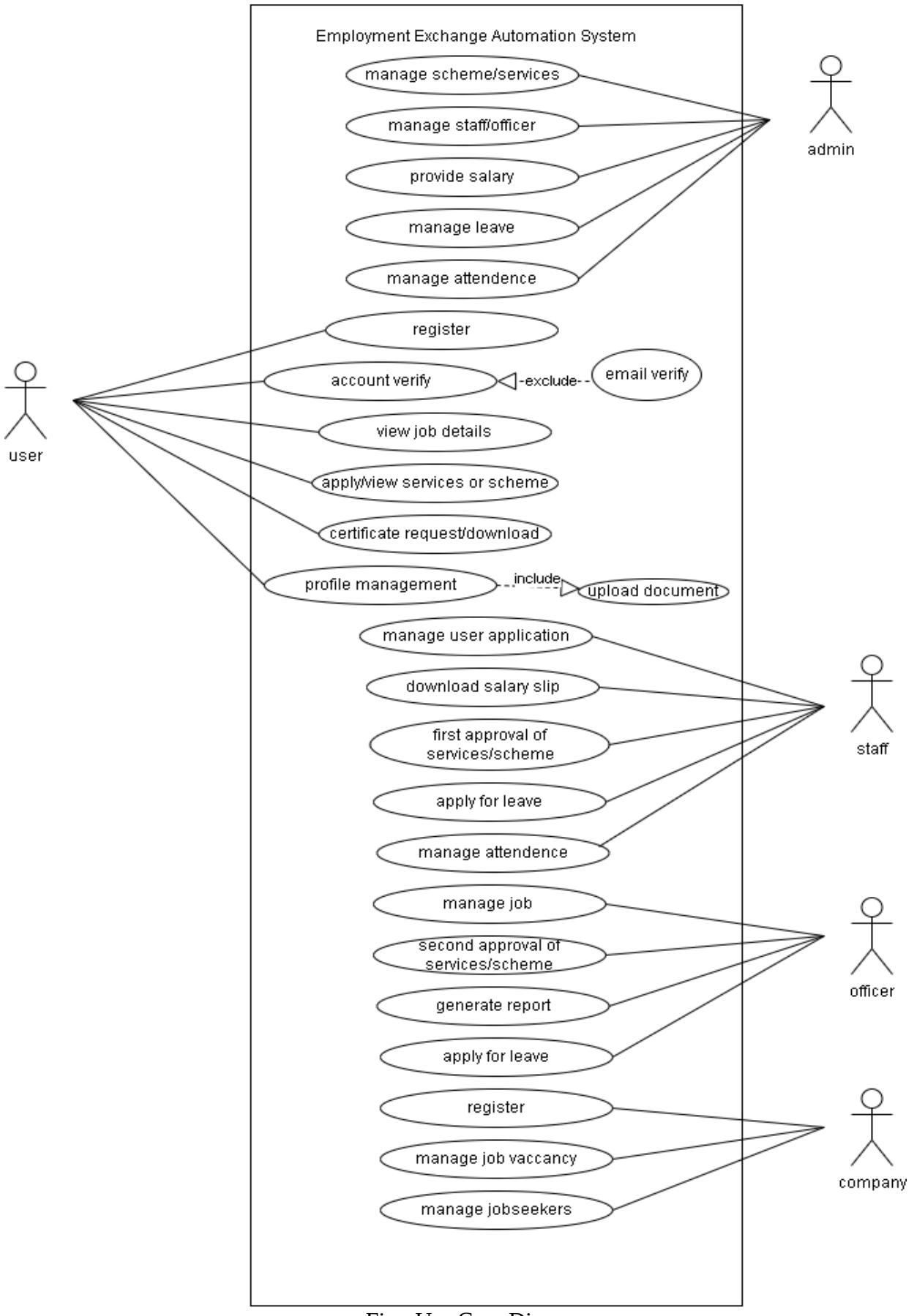


Fig : Use Case Diagram

4.2.2 SEQUENCE DIAGRAM

A sequence diagram only represents the sequential order in which objects interact with one another, or the order in which these interactions occur. The terms event diagrams and event scenarios can also be used to describe a sequence diagram. Sequence diagrams show the actions that the components of a system take. Software engineers and businesspeople alike frequently use these diagrams to document and comprehend the requirements for both new and current systems.

Sequence Diagram Notations –

i. **Actors –** A type of role that interacts with the system and its objects is represented by an actor in a UML diagram. An actor is always outside the purview of the system that the UML diagram is intended to represent. To portray a variety of roles, including those of human users and other external subjects, we employ actors. Stick person notation is used to represent an actor in a UML diagram. A sequence diagram can have several actors.

ii. **Lifelines –** A named element that shows a specific participant in a sequence diagram is called a lifeline. In a sequence diagram, each instance is essentially represented by a lifeline. The lifeline components in a sequence diagram are at the top.

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

Messages can be broadly classified into the following categories:

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

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

Uses of sequence diagrams –

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

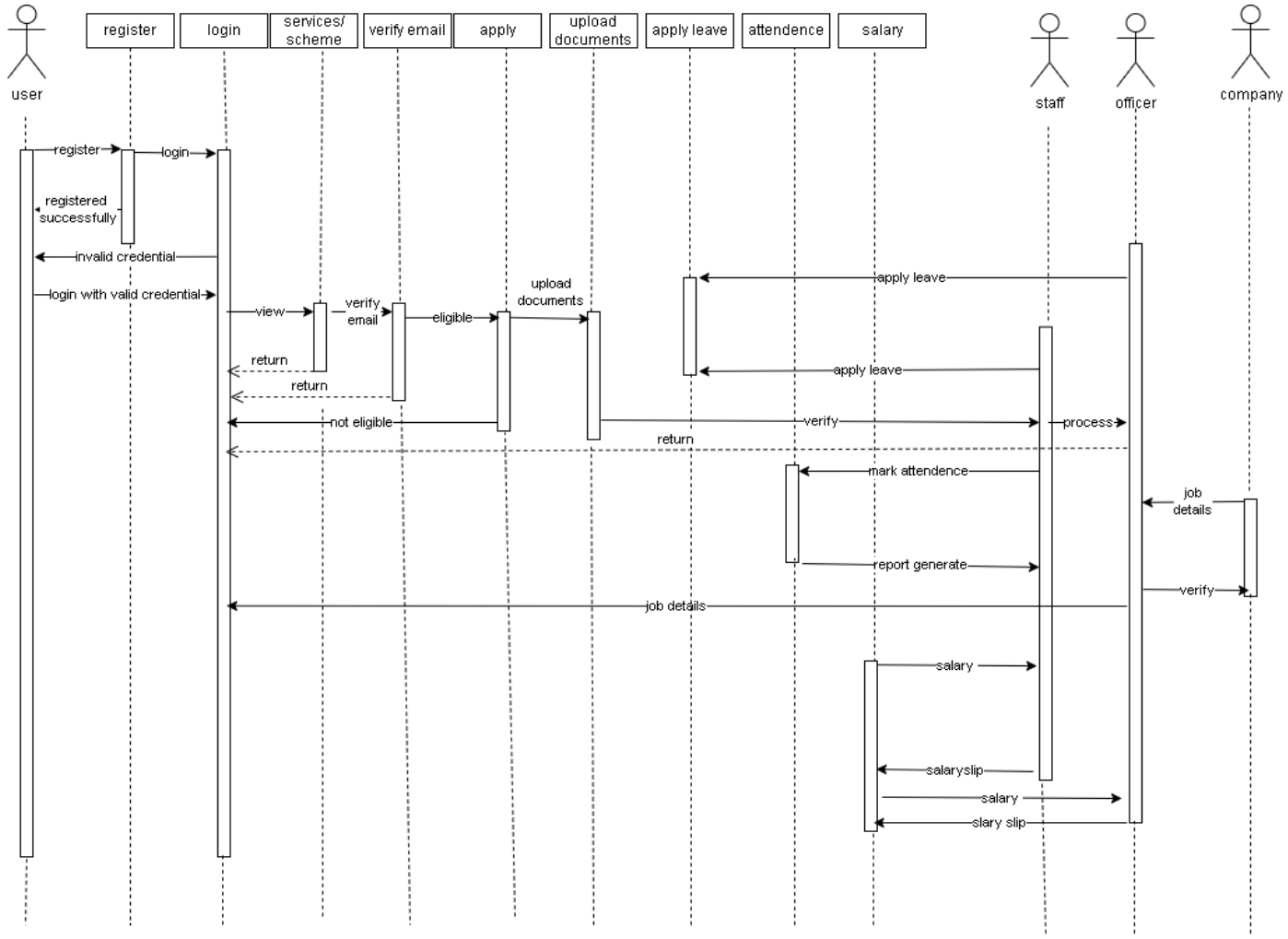


Fig: sequence diagram

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 a complete 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.

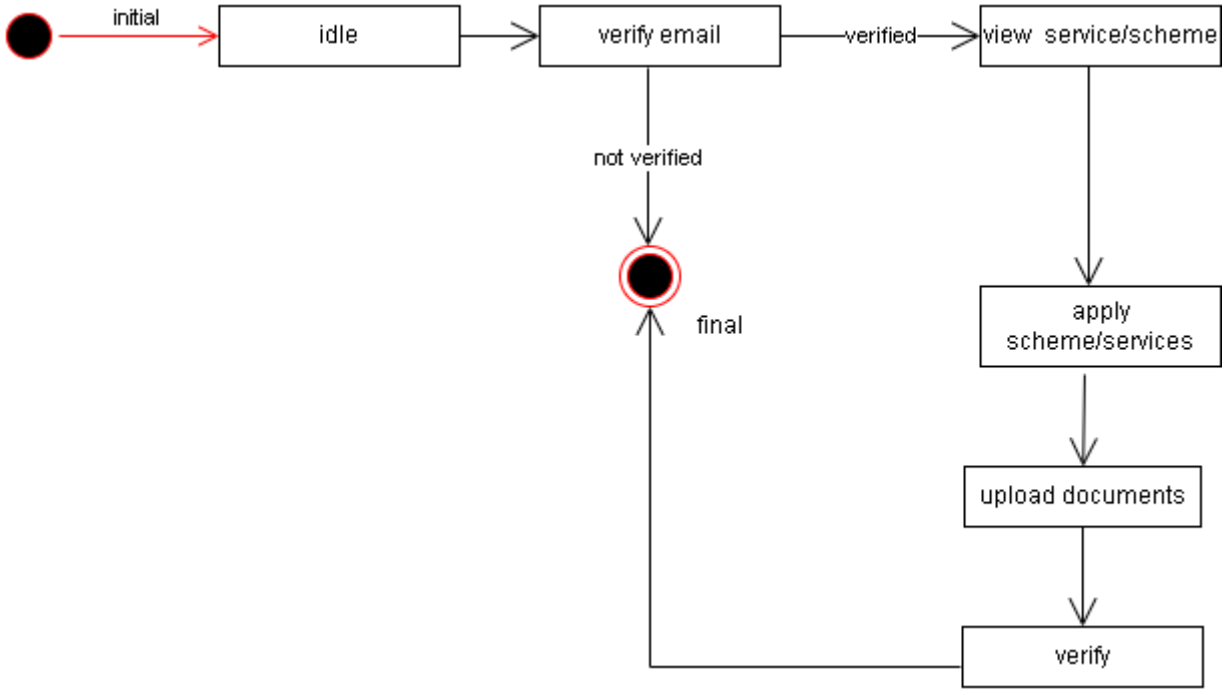


Fig : State Chart Diagram

4.2.4 Activity Diagram

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

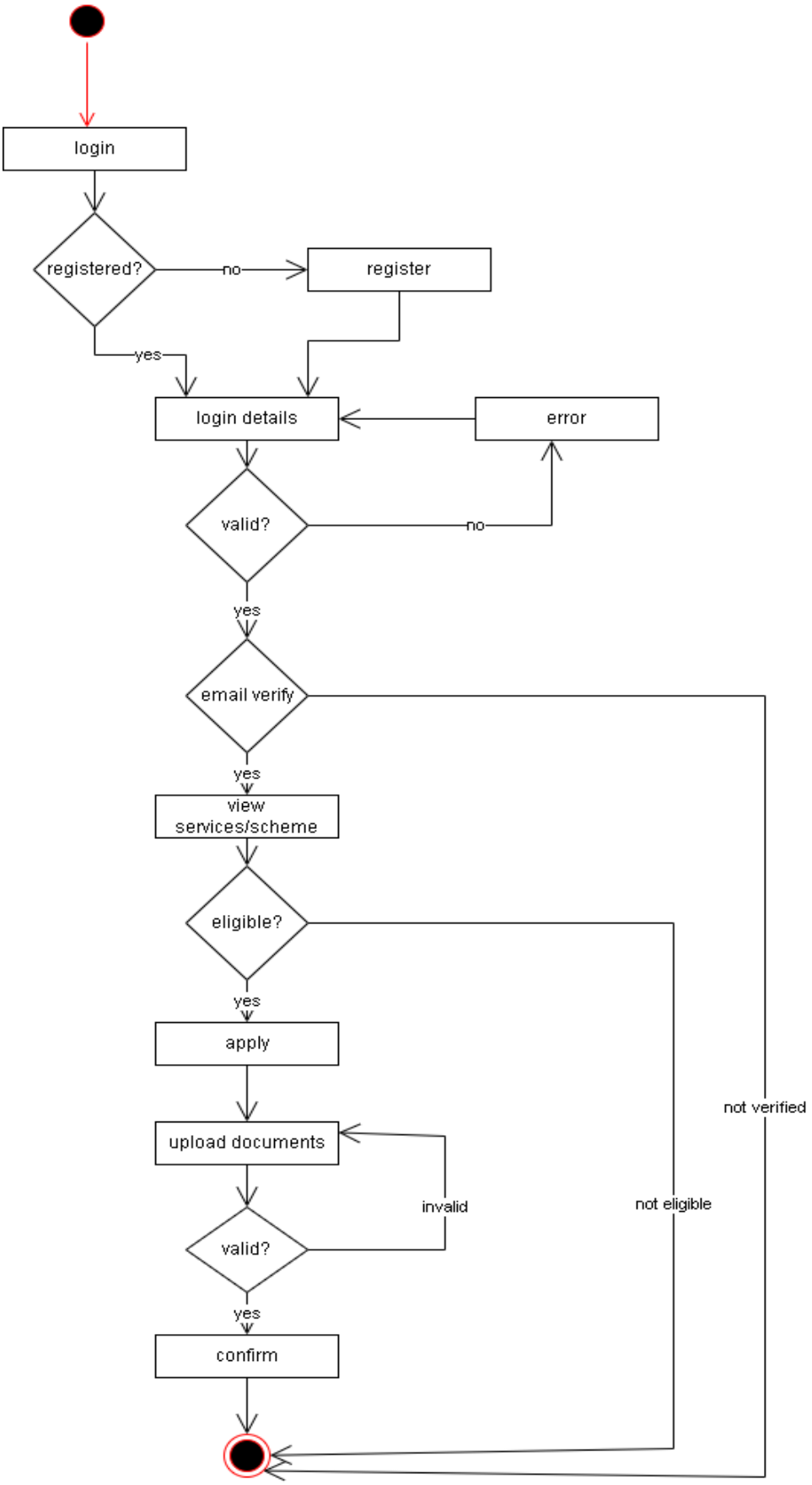


Fig : Activity Diagram

4.2.5 Class Diagram

Static diagrams are class diagrams. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualizing, explaining, and documenting various elements of systems. The characteristics and functions of a class are described in a class diagram, along with the restrictions placed on the system. Because they are the only UML diagrams that can be directly mapped with object-oriented languages, class diagrams are frequently employed in the modelling of object-oriented systems. A collection of classes, interfaces, affiliations, collaborations, and constraints are displayed in a class diagram. It also goes by the name "structural diagram."

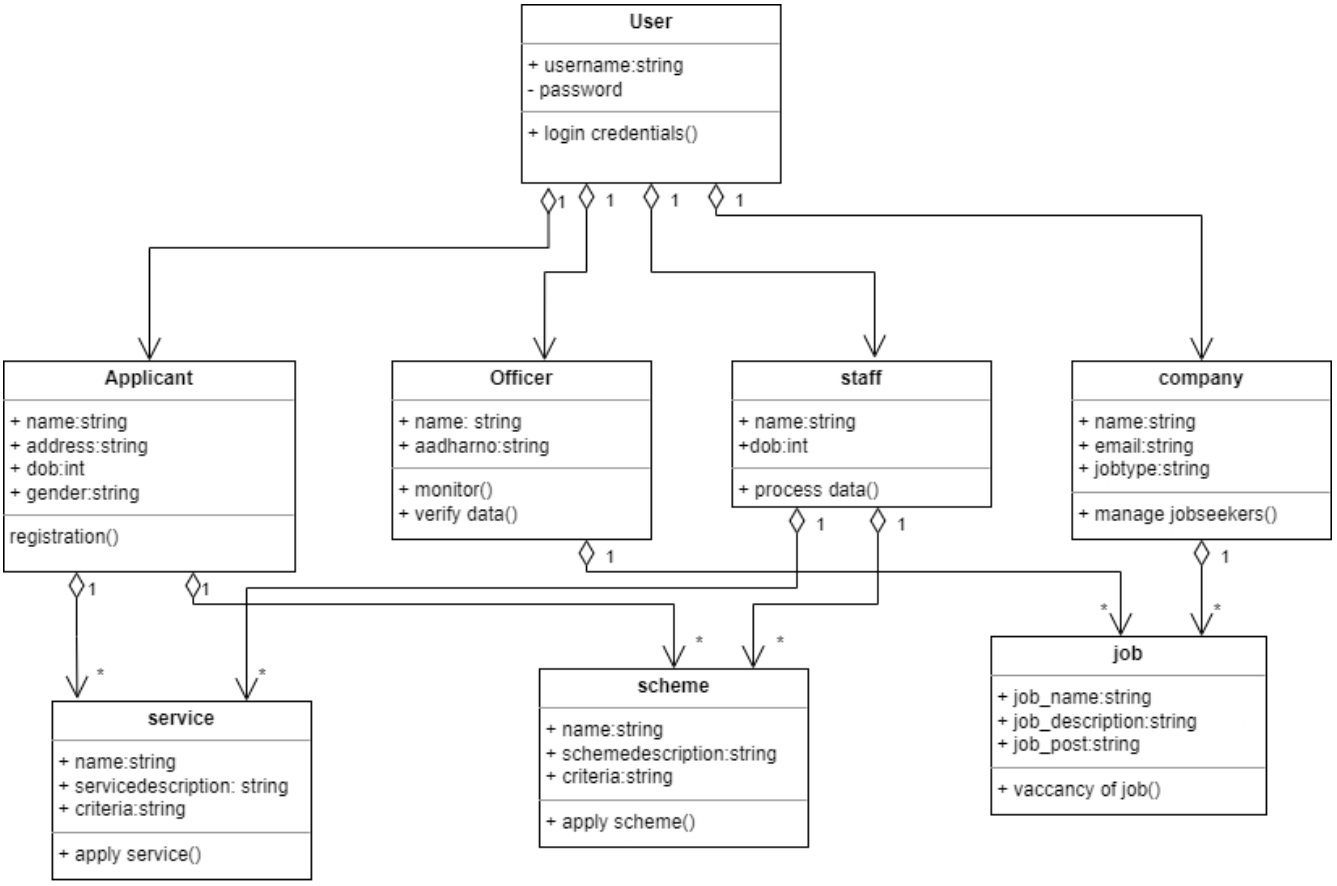


Fig : Class Diagram

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

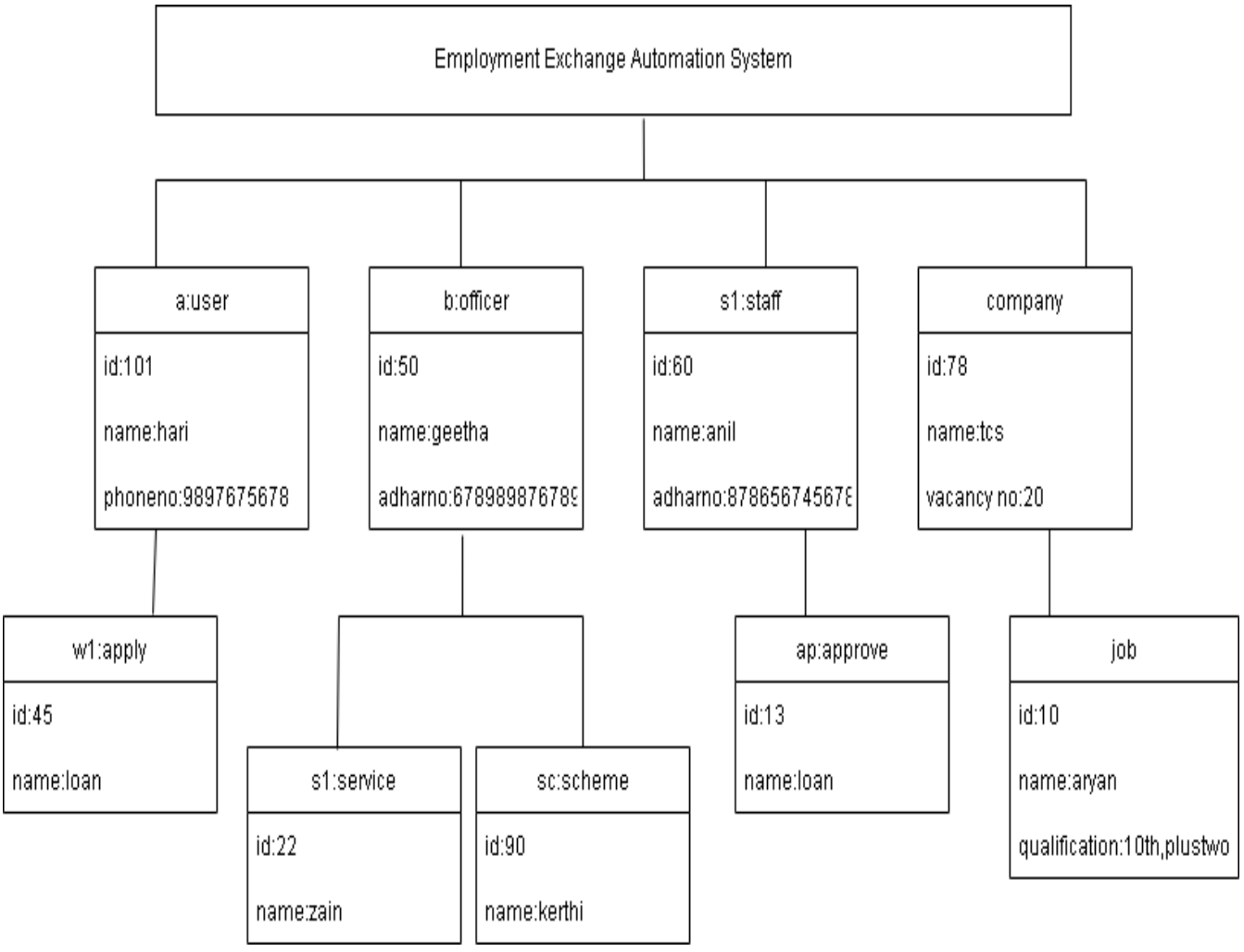


Fig : Object Diagram

4.2.7 Component Diagram

In terms of nature and behavior, component diagrams vary. The physical components of a system are modelled using component diagrams. The components that make up a node's physical aspects include executables, libraries, files, documents, etc. Component diagrams are used to depict how a system's components are arranged and related to one another. Additionally, executable systems are created using these representations.

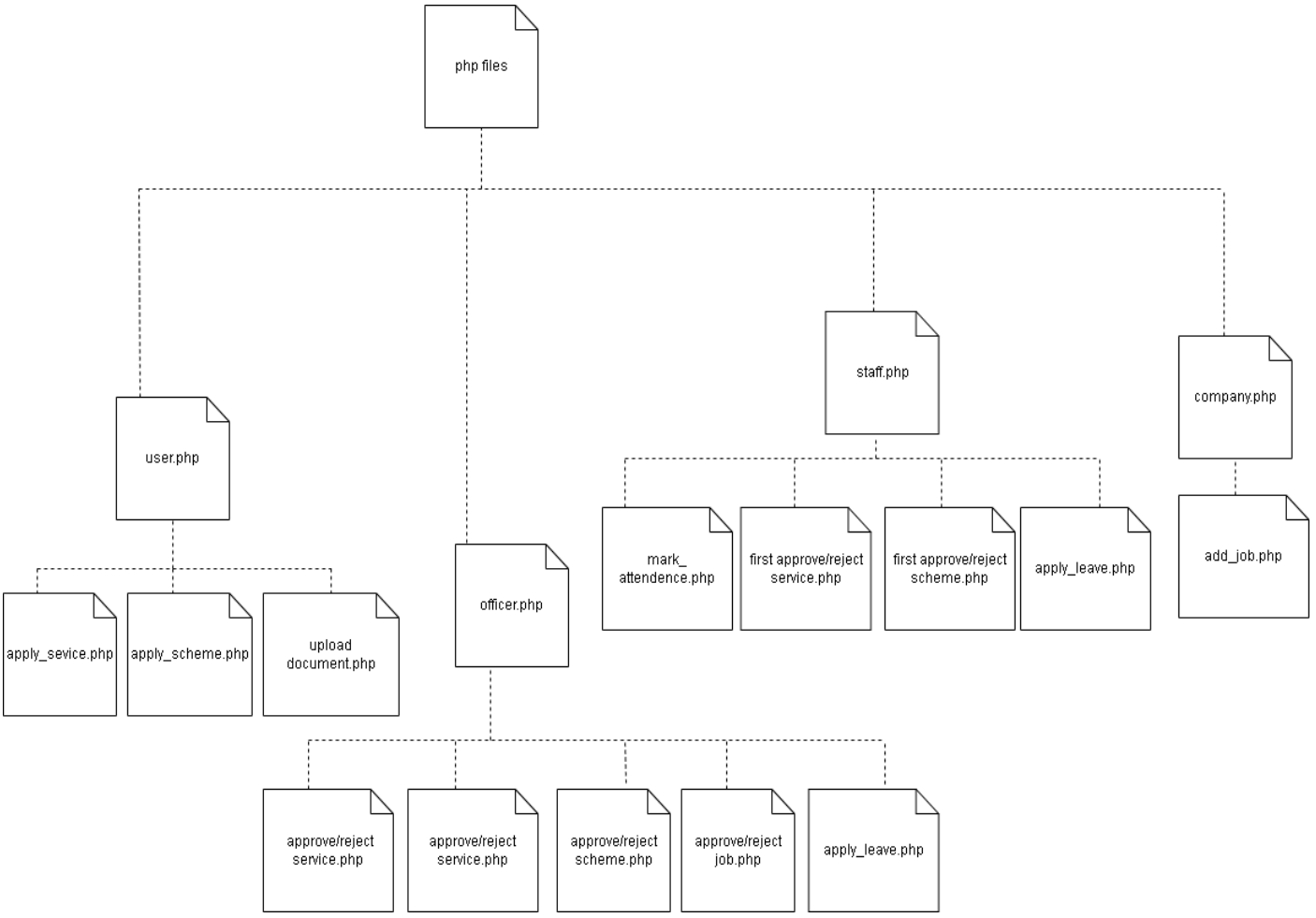


Fig : Component Diagram

4.2.8 Deployment Diagram

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

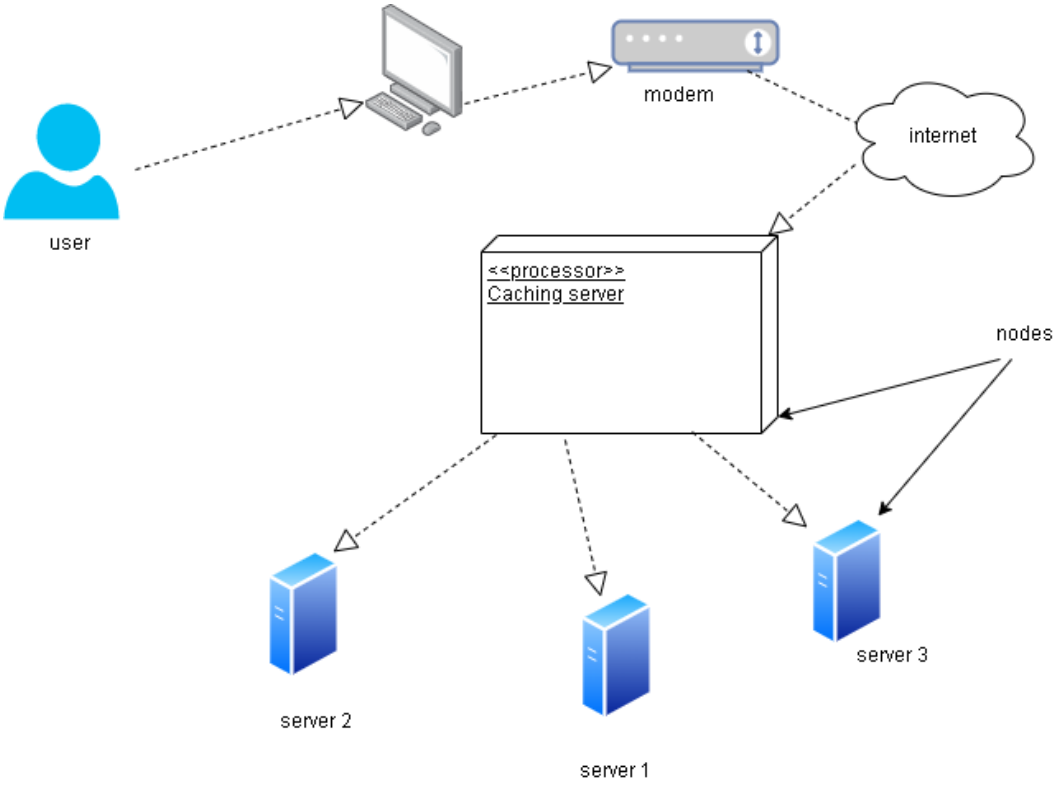
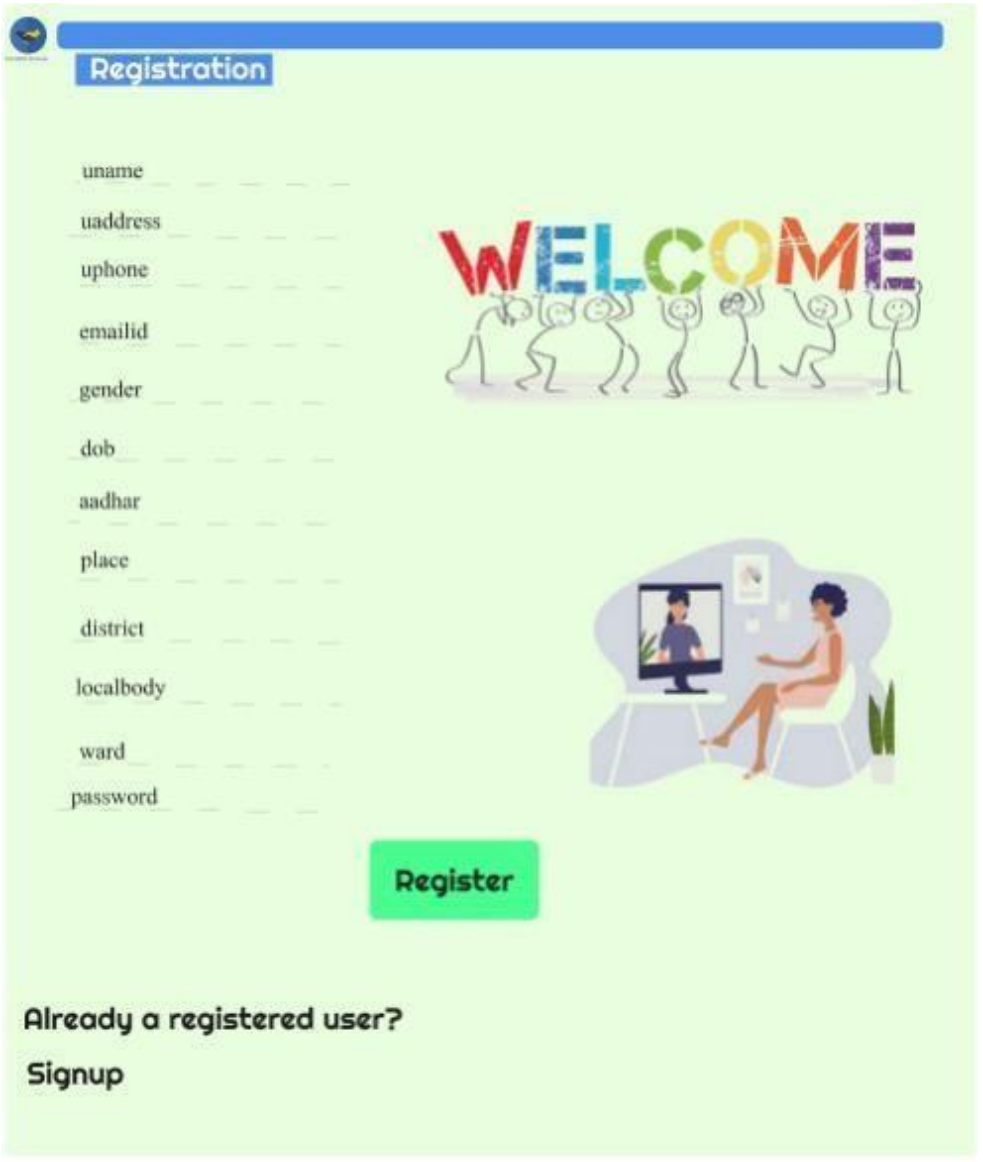


Fig : Depolyment Diagram

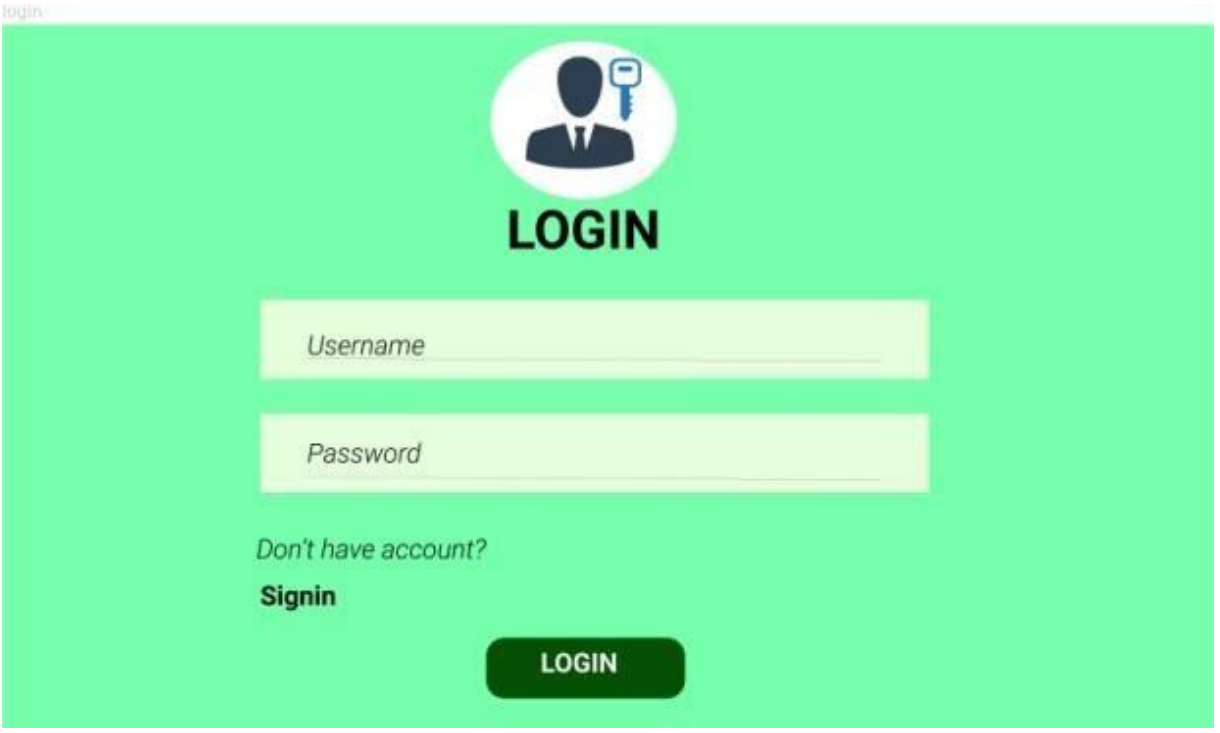
4.3 USER INTERFACE DESIGN

4.3.1 INPUT DESIGN

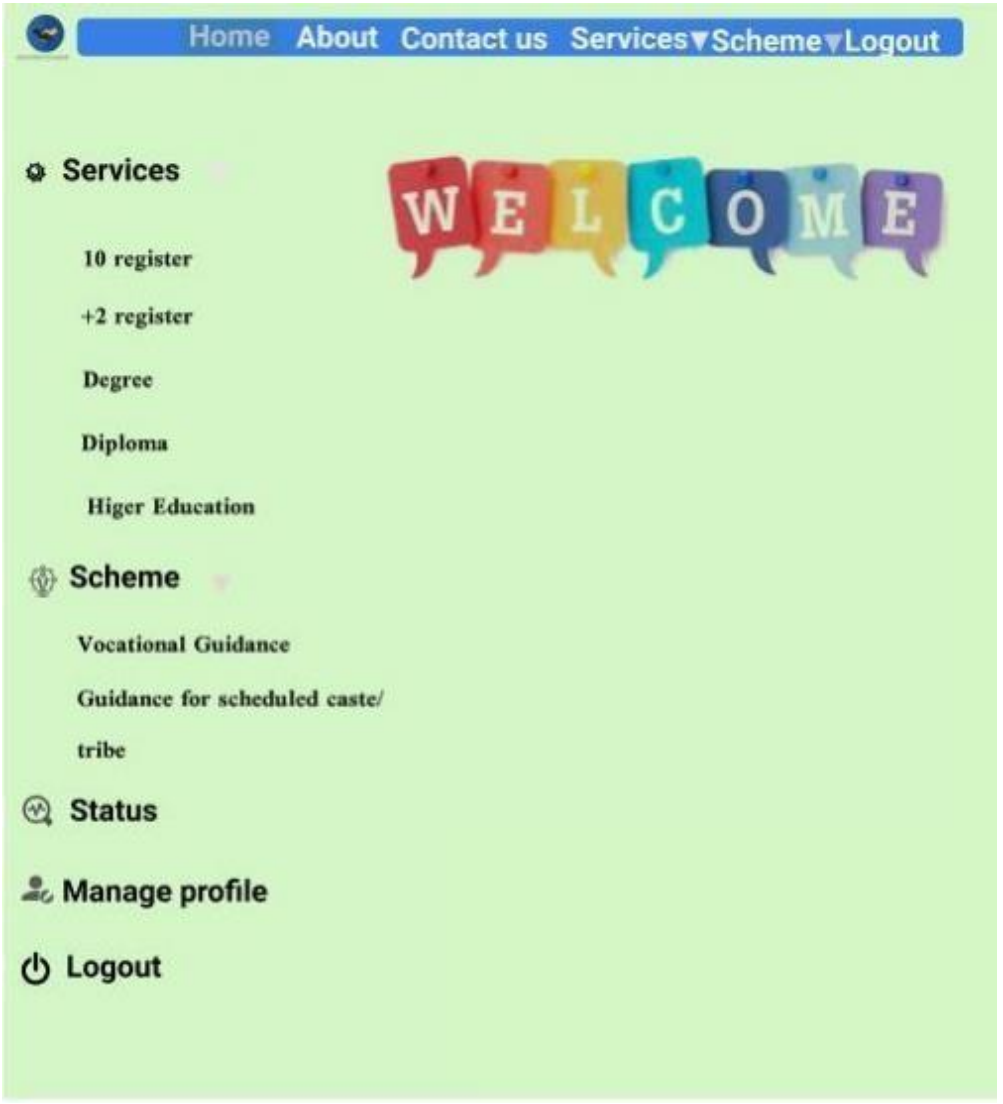
Form Name : User Registration



Form Name : User Login

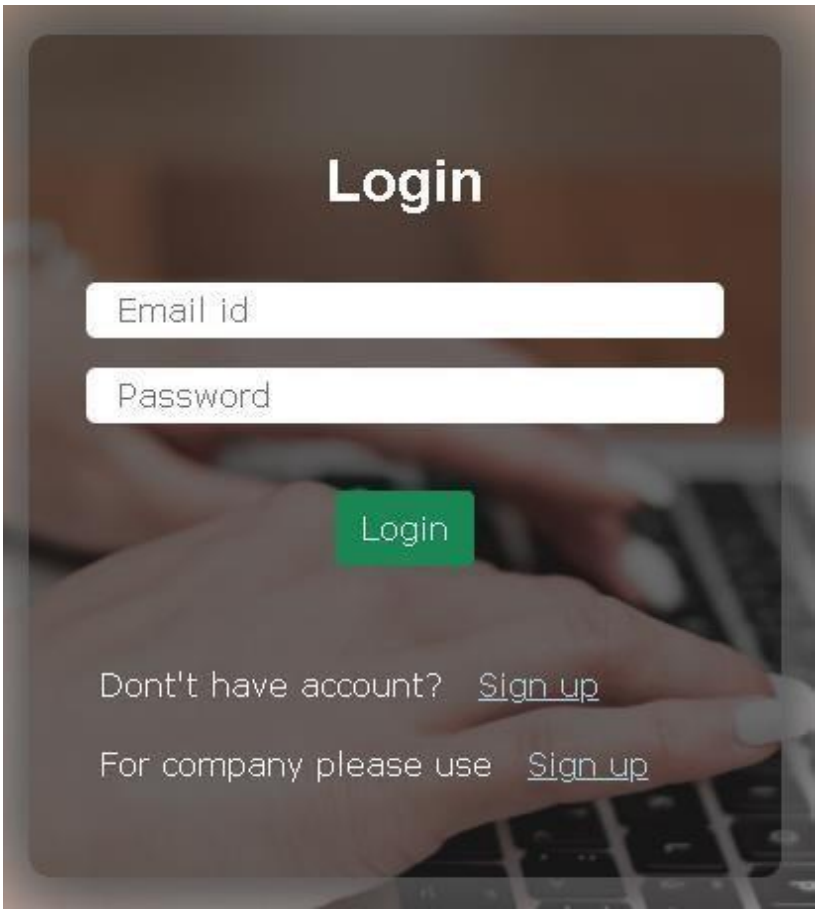


Form Name : Service request

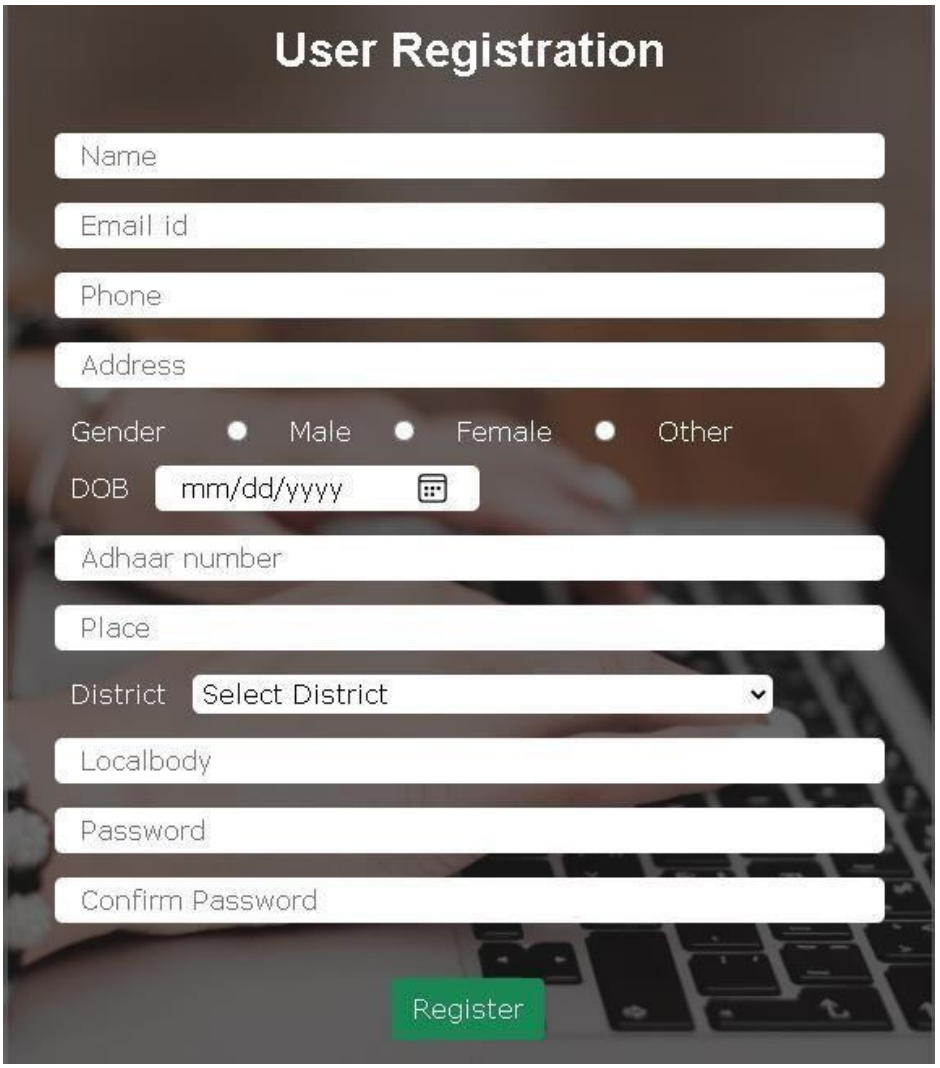


4.3.2 OUTPUT DESIGN

User Login



User Registration



4.4 DATABASE DESIGN

A database is a structured system with the ability to store information and allow users to quickly and effectively access that information. Any database must be protected because its primary goal is its data. A two-level procedure goes into designing a database. In the first step, user needs are compiled, and a database is developed to as clearly as possible meet these criteria. Information Level Design is the name of this stage, which is carried out independently of any specific DBMS.

This information level design is converted into a design for the particular DBMS that will be used to implement the system in question in the second stage. Physical Level Design is the phase in which the qualities of the chosen DBMS are discussed. The system design and the database design are concurrent. The database's data organization aims to accomplish the following two main goals.

- Data Integrity
- Data independence

4.4.1 Relational Database Management System (RDBMS)

In a relational model, the database is shown as a set of relations. Each relation resembles a file or table of records with values. 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 in formal relational model language. A relational data base is made up of a number of tables, each with its own name. In a story, each row represents a group of associated values.

Relations, Domains & Attributes

A table is a relation. 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

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity forbids the use of null values for any Primary Key.
- No Primary Key may contain null values, according to Referential Integrity.
- Referential Integrity: A Primary Key value in the same domain must correspond to each unique Foreign Key value. Super Key and Candidate Keys are additional keys..

4.4.2 Normalization

The simplest possible arrangement 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 normal type 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.

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

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

First Normal Form

According to the First Normal Form, each attribute's domain must only include atomic values, and each attribute's value in a tuple must be a single value from that domain. In other words, 1NF forbids using relationships as attribute values within tuples or relations within relations. Single atomic or indivisible values are the only attribute values that are permitted under 1NF. The data must first be entered into First Normal Form. This can be accomplished by separating data into tables of a similar type in each table. According to the project's requirements, a Primary Key or Foreign Key is assigned to each table. For each non-atomic property or nested relation, new relations are created in this process. Repeated data groupings were removed as a result. A relation is considered to be in first normal form if and only if it complies with the main key-only requirement.

Second Normal Form

No non-key attribute should be functionally dependent on a portion of the main key for relations where the primary key has several attributes, according to Second Normal Form. This involves breaking down each partial key into its dependent characteristics and setting up a new relation for each one. Keep the original primary key and any properties that are entirely dependent on it in your database. This procedure aids in removing data that depends only on a small portion of the key. If a connection meets all the requirements for first normal form for the main key and every non-primary key feature of the relation is completely dependent on its primary key alone, then and only then is the relation said to be in second normal form.

Third Normal Form

A non-key attribute of a Relation should not be functionally determined by another non-key property or by a collection of non-key attributes, according to the Third Normal Form. In other words, the primary key shouldn't be transitively dependent on anything. In this, we break down the relation into its component parts and build up the non-key qualities that functionally determine the other non-key attributes. To eliminate anything that does not totally dependent on the Primary Key, this step is conducted. If a relation is solely in second normal form and moreover does not depend on another relation's non-key qualities, it is said to be in third normal form.

Indexing

By reducing the number of disk accesses needed when a query is completed, indexing helps a database perform better. It is a data structure method used to locate and access data in a database

rapidly. Several database columns are used to generate indexes. The primary key or candidate key of the table is duplicated in the first column, which is the Search key. To make it easier to find the related data, these values are kept in sorted order. Recall that the information may or may not be kept in sorted order.

Data Sanitization

An automated procedure called "sanitization" is used to get a value ready for use in a SQL query. This process typically involves checking the value for particular characters that have a special significance for the target database. To prevent a SQL injection attack, you must sanitize (filter) the input string while processing a SQL query based on user input. For instance, the user and password input is a typical scenario. In that particular scenario, the server response would provide access to the 'target user' account without requiring a password check

4.5 TABLE DESIGN

1. tbl_login

Table description: To store login details

Primary key : **lid**

Foreign key:**login_id** (regid from tbl_registration)

SI.No	Name	Data type	Description
1.	lid	Int (10)	Login id
2.	login_id	Int (10)	Regid from table tbl_registration
3.	emailid	Varchar (100)	Email Id
4.	password	Varchar (100)	Password
5.	usertype	Int (10)	Usertype
6.	designation	Varchar (200)	Designation
7.	status	Int (10)	Current status
8.	verified	Int (10)	Email verified or not

2. tbl_registration

Table description: To store registration details

Primary key : **regid**

SI.No	Name	Data type	Description
1.	regid	Int (10)	Registration id
2.	uname	Varchar (200)	user name
3	uaddress	Varchar (200)	Address
4.	uphone	Bigint (10)	user phone
5.	aadharno	Bigint (12)	Aadhar number
6.	company_registration	Int (20)	Company registration
7.	gender	Varchar (10)	Gender
8.	dob	Date	Date of birth
9.	place	Varchar (200)	Place
10.	district	Varchar (200)	District
11.	localbody	Varchar (200)	Localbody
12.	companyname	Varchar (200)	Companyname
13.	description	Varchar (200)	Company Description

3. tbl_services

Table description: To store services details

Primary Key : **serviceid**

SI.No	Name	Data type	Description
1.	serviceid	Int (10)	service id
2.	name	Varchar (300)	service name
3.	subservice	Varchar (300)	service subname
4.	service_description	Varchar (2000)	Description
5.	criteria	Varchar (300)	Criteria
6.	docs_needed	Varchar (400)	Documents
7.	category	Varchar (300)	Category
8.	status	Int (10)	current status

4. tbl_slide

Table description: To store notification details

Primary key **id**

s

SI.No	Name	Data type	Description
1.	id	**Int (10)**	SlideId
2.	sildename	Varchar (100)	Name
3.	slide	Varchar (800)	Image

5. tbl_user_schemes

Table description: To store applied services of user

Primary key **scheme_id**

Foregin key **doc_fid** (doc_id from tbl_documents)

Foregin key **reg_fid** (regid from tbl_registration)

Foregin key **cscheme_id (**schemeid from tbl_cscheme)

SI.No	Name	Data type	Description
1.	user_scheme_id	**Int (20)**	User scheme Id
2.	doc_fid	Int (20)	Document id
3.	reg_fid	Int (20)	Register id
4.	cscheme_id	Int (20)	Scheme id
5.	status	Int (20)	Current status
6.	comments	Varchar (500)	Comments
7.	applied_date	date	Date
8.	rejection_status	Int (10)	Rejection
9.	approval_status	Int (100)	Approval

6. tbl_attendence

Table description: To store attendance details of officers and staff

Primary key **attendance_id**

Foreign key **attendance_reg_id (**regid from tbl_registartion**)**

SI.No	Name	Data type	Description
1.	attendance_id	Int (10)	Attendence id
2.	attendance_reg_id	Int (10)	Regidtartion id from table tbl_registration
3.	time_in	time	Time
4.	attend_date	date	Today's date
5.	time_out	time	Timeout
6.	full_half_day	Varchar (100)	Leave days

7. tbl_documents

Table description: To store documents of users

Primary key **doc_id**

Foreign key **fid** (regid from tbl_registration)

SI.No	Name	Data type	Description
1.	doc_id	Int (10)	Document id
2.	fid	Int (10)	Register id from tbl_registation table
3.	adhaar	Varchar (300)	Aadharimage
4	pan	Varchar (300)	Pan card
5.	birth	Varchar (300)	Birth certificate
6.	death	Varchar (300)	Death certificate
7.	income	Varchar (300)	Income certificate
8.	widow	Varchar (300)	Widow certificate
9.	tenth	Varchar (300)	Tenth certificate
10.	plus2	Varchar (300)	Plus 2 certificate

8. tbl_cscheme

Table description: To store new schemes

Primary key **schemeid**

SI.No	Name	Data type	Description
1.	schemeid	Int (10)	Service id
2.	name	Varchar (200)	Service name
3.	subscheme	Varchar (200)	Service subname
4.	schemedescription	Varchar (900)	Description
5.	criteria	Varchar (200)	Criteria
6.	docs_needed	Varchar (800)	Documents needed
7.	category	Varchar (200)	Category
8.	status	Int (10)	Current status

9. tbl_alert

Table description: To store alert messages

Primary key **atid**

SI.No	Name	Data type	Description
1.	atid	Int (10)	Alert Id
2.	message	Varchar (500)	Message
3.	status	Int (5)	Current status

10. tbl_applied_leaves

Table description: To store leave details of officers and staff

Primary key **applied_leaves_id**

Foreign key **applied_leaves_reg_id** (regid from tbl_registration)

SI.No	Name	Data type	Description
1.	applied_leaves_id	Int (10)	Applied leave Id
2.	applied_leaves_reg_id	Int (10)	Applied leave registered id
3.	leave_type	Varchar (255)	Leave types
4.	applied_date	date	Applied leave date
5.	from_date	date	Leave applied from date
6.	to_date	date	Leave applied to date
7.	leave _reason	Varchar (800)	Reason for leave
8.	rejection_reason	Varchar (255)	Reason for rejection
9.	leave_status	Int (10)	Status of leave

11. tbl_jobs

Table description: To store job information which is provided by company

Primary key **jobs_id**

Foreign key **company_id** (regid from tbl_registration**)**

SI.No	Name	Data type	Description
1.	jobs_id	Int (10)	Job id
2.	company_id	Int (10)	Registration id from tbl_registartion
3.	job_name	Varchar (300)	Job name
4	job_description	longtext	Job description
5.	job_post	Varchar (300)	Job post
6.	job_branch	Varchar (400)	Job branch
7.	job_vaccancy_no	Varchar (400)	Vacancy number
8.	job_qualification	Varchar (800)	Qualification
9.	submitted_on	date	Submitted date
10.	job_start_date	date	Interview start date
11.	job_end_date	date	Interview end date
12.	status	Int (10)	current status
13.	job_type	Int (2)	job type
14.	company_employee_no	Mediumint (255)	employee number
15.	job_pay_range	Bigint (255)	Salary
16.	approved_by	Varchar (200)	approved by
17.	approved_on	date	Aproned on date
18.	rejection_reason	mediumtext	Rejection reason

12. tbl_leave

Table description: To store leave types of users

Primary key **leave_id**

Foregin key **leave_reg_id** (regid from **t**bl_registration)

SI.No	Name	Datatype	Description
1.	leave_id	Int (10)	Leave_id
2.	leave_regid	Int (10)	Registration id from tbl_registration
3.	earned_leave	Int (20)	Earned leave
4.	casual_leave	Int (20)	Casual leave
5.	sick_leave	Int (20)	Sick leave
6.	hospital_leave	Int (20)	Hospital leave
7.	maternity_leave	Int (20)	Maternity leave

13. tbl_salary

Table description: To store salary details of officers and staff

Primary key **salary_id**

Foregin key **salary_regid** (regid from tbl_registration)

SI.No	Name	Datatype	Description
1.	salary_id	Int (10)	salary_id
2.	salary_regid	Int (10)	Registration id from tbl_registration
3.	month	Varchar (10)	Month
4.	year	Int (10)	Year
5.	salary	Int (200)	Salary
6.	salary_usertype	Int (100)	Usertype

14. tbl_qualification

Table description: To store qualification details of users

Primary key **qualification_id**

Foregin key **quali_reg_id** (regid from tbl_registration)

SI.No	Name	Datatype	Description
1.	qualification_id	Int (10)	qaualification_id
2.	quali_reg_id	Int (10)	Foregin key for the table
3.	qualification	Varchar (200)	Qualification type
4.	institution	Varchar (200)	Institution
5.	board	Varchar (200)	Education board
6.	register_no	Int (12)	Register number
7.	percentage	Int (50)	Percentage of mark
8.	cgpa	Int (50)	Cgpa mark
9.	passoutmonth	Varchar (10)	Passout month
10.	passoutyear	Int (10)	Passout year

15. tbl_user_details

Table description: To store user details

Primary key **user_details_id**

Foreign key **user_reg_id (**regid from tbl_registration**)**

SI.No	Name	Data type	Description
1.	user_details_id	Int (10)	User details id
2.	user_req_id	Int (10)	Register id from tbl_registration
3.	marital_status	Int (100)	Martial status
4	ward	Int (100)	Ward
5.	caste	Varchar (200)	Caste
6.	religion	Varchar (200)	Religion
7.	qualifications	Varchar (500)	Qualification
8.	display_picture	Varchar (900)	Display image

16. tbl_user_services

Table description : To store users applied services

Primary key **user_service_id**

Foreign key **doc_id (**doc_id from tbl_documents)

Foreign key **reg_id** (regid from tbl_registration**)**

Foreign key **service_id (**service_id reference from tbl_services)

SI.No	Name	Data type	Description
1.	user_service_id	Int (10)	User service id
2.	doc_id	Int (10)	Document id
3.	reg_id	Int (10)	Register id
4	service_id	Int (10)	Service id
5.	status	Int (20)	Current status
6.	comments	Varchar (500)	Comments
7.	applied_date	date	Applied date
8.	rejection_status	Int (10)	Rejection status
9.	approval_type	Int (20)	Approval type

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

The act of meticulously putting software through its paces to see whether it operates as intended is known as software testing. The terms "verification" and "validation" are widely used in relation to software testing. Examining or testing items, especially software, for consistency and adherence to a given specification is the process of validation. Software testing is only one sort of verification; reviews, analyses, inspections, and walkthroughs are other techniques applied in verification. Validation is the process of making sure that what has been specified is what the user actually wants.

Other procedures that are typically connected to software testing include static analysis and dynamic analysis. Without actually running the code, static analysis examines the software's source code to look for errors and gather statistics. Dynamic analysis looks at the behavior of software while it is in use to provide information like execution traces, timing profiles, and test coverage specifics.

A series of activities known as testing can be organized in advance and carried out in a methodical manner. Starting with the module level, testing progresses to the integration of the full computer-based system. Testing is necessary for the system testing objectives to be successful, and there are many rules that can be used as testing objectives. These are:

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

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

Software bugs would be found if testing was successfully completed in accordance with the aforementioned objectives. Additionally, testing shows that the program appears to function as intended and that the performance criteria appear to have been satisfied.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness is supposed to verify that a program does exactly what it was designed to do.

This is much more difficult than it may at first appear, especially for large programs

5.2 TEST PLAN

A test plan denotes a sequence of desired steps to be taken in order to complete various testing procedures. The Test Plan serves as a roadmap for the course of activity. A computer program is produced by software developers; however, An independent test group (ITG) was created to address the inherent issues with allowing the builder to test what they have created. The specified objective soft testing ought to do the greatest in quantifiable aspects. Therefore, the test plan should include information on the likelihood of failure, the cost to locate and fix the flaws, the remaining defect density or frequency of occurrence, and the number of test work hours required for each regression test.The levels of testing include:

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

5.2.1 Unit Testing

Unit testing focusses 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. Unit testing is white-box focused, and it may be done simultaneously for several 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. A module's boundary conditions are checked to make sure that each statement has been run at least once. All methods for handling errors are then tested. Before starting any other test, data flow tests across a module interface must be completed. All other tests are meaningless if data cannot enter and exit properly. The unit test must include the selective testing of execution pathways. Error handling pathways must be set up to reroute traffic or cleanly terminate, as is required by good design processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

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 modules were incorporated into the system after unit testing was completed in order to check for any interface consistency issues. Additionally, variations in program architectures were eliminated, and a singular program structure emerged.

5.2.3 Validation Testing or System Testing

The testing process comes to an end here. This involved testing the entire system in its entirety, including all forms, code, 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 Output Testing or User Acceptance Testing

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

5.2.4.1 Input Screen Designs,

5.2.4.2 Output Screen Designs,

The testing mentioned above is carried out using several test data types. In the system testing process, the preparation of test data is crucial. The test data are prepared, and then the system understudy is examined using those test data. During system testing, faults in the test data are once again found and fixed using the testing procedures described above. The fixes are also logged for use in the future.

5.2.5 Automation Testing

Automation testing is the process of testing software and other tech products to ensure it meetsstri ct requirements. Essentially, it's a test to double-check that the equipment or software doesexactly what it was designed to do. It tests for bugs, defects, and any other issues th at canarisewith product development. Automation testing can be run at any time of the day. It us esscripted 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 indepth, and provide detailed reports that 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. Automation testing makes this process an easier one. It's also able to analyze wider test coverage than humans may be able to.

- **Simplifies testing** - Testing is a routine part of the operations of most SaaS and techcompanies. 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 shortensyour software development cycles.

- **Reduces human intervention** - Tests can be run at any time of day, even overnight, with outneeding humans to oversee it. Plus, when it's conducted automatically, this can also reduce risk of human error.

5.2.6 Selenium Testing

Selenium is an opensource tool that automates web browsers. It provides a single interface tha t lets you write test scripts in programming languages like Ruby, Java, NodeJS, 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.

Test cases for a Login Page

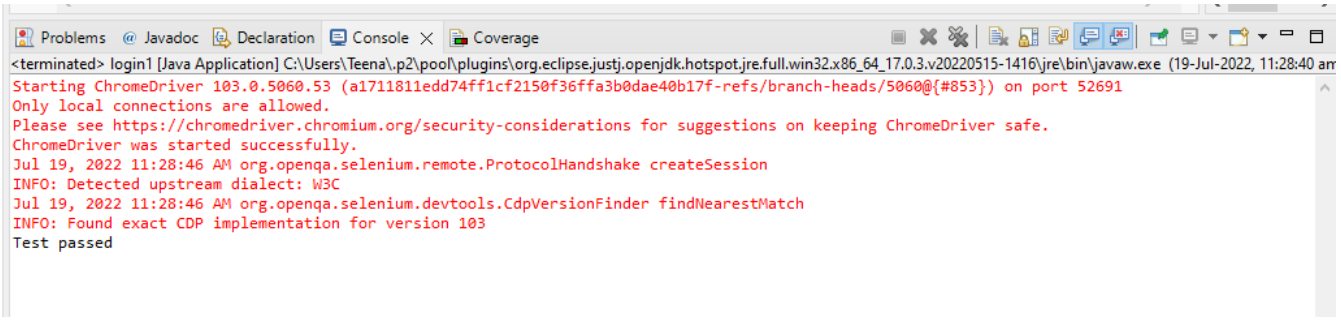
Project Name: Employment Exchange Automation System					
Login Test Case					
Test Case ID: login1			**Test Designed By: Avani P A**		
Test Priority (Low/Medium/High): High			**Test Designed Date:**19-07-2022		
Module Name: login			**Test Executed By: Ms. Anit James**		
Test Title: Verify login with valid usernameand password			**Test Execution Date: 19**-07-2022		
Description: Test the Login Page					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Login Page		Login page should be displayed	Login page displayed	Pass
2	Provide valid username	Username : avanipa1999@gmail.com	User should be able to login	User logged in and navigated to user dashboard	Pass
3	Provide valid password	Password: Password@2022			
4	Click on Sign In button				
5	Provide invalid username or password	Username: kerthi@gmail.com Password: Passwor@d	User should be able to login	Message for enter valid username or password	Pass
6	Provide null username and password	Username :null Password :null			
7	Click on Sign In Button				

Post -Condition : User is validated with database and successfully login into account. The Account session on details are logged in database.

Code package

```
package test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class login1 {
 public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","C:\\Users\\Teena\\Downloads\\chromedriver_win32\\chromedriver.exe" );
  WebDriver driver=new ChromeDriver();
  driver.get("http://localhost/emp/login.php");
  driver.findElement(By.id("email")).sendKeys("avanipa1999@gmail.com");
  driver.findElement(By.id("password")).sendKeys("Password@2022");
  driver.findElement(By.id("register")).click();
  String actualUrl="http://localhost/emp/index.php";
  String expectedUrl= driver.getCurrentUrl();
  if(actualUrl.equalsIgnoreCase(expectedUrl))
  {
  System.out.println("Test passed");
  }
  else
  {
  System.out.println("Test fail");
  }
 }
}
```

Output



```java
package test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class login1 {
 public static void main(String[] args) {
     System.setProperty("webdriver.chrome.driver","C:\\Users\\Teena\\Downloads\\chromedriver_win32\\chromedriver.exe" );
  WebDriver driver=new ChromeDriver();
  driver.get("http://localhost/emp/login.php");
  driver.findElement(By.id("email")).sendKeys("avanipa1999@gmail.com");
  driver.findElement(By.id("password")).sendKeys("Password@2022");
  driver.findElement(By.id("register")).click();
  String actualUrl="http://localhost/emp/index.php";
  String expectedUrl= driver.getCurrentUrl();
  if(actualUrl.equalsIgnoreCase(expectedUrl))
  {
  System.out.println("Test passed");
  }
  else
  {
  System.out.println("Test fail");
  }
 }
}
```

_

Test cases for a Registration Page

Project Name: Employment Exchange Automation System					
Registration Test Case					
Test Case ID: register3			**Test Designed By: Avani P A**		
Test Priority (Low/Medium/High): High			**Test Designed Date:**19-07-2022		
Module Name: Registration			**Test Executed By: Ms. Anit James**		
Test Title: Verify registration with their details			**Test Execution Date:** 19-07-2022		
Description: Test the registration page					
Pre-Condition: User with valid address					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to registration		Registration should be displayed	Registration page displayed	Pass
2	Provide valid details	name : Geethanjali email: trm4749@gmail.com phone:7859686235 address: Pulikkunnanmukaili gender:female date: 18-07-1999 adhaar:785963256325 place: Alappuzha district:Idukki localbody:Mulipality	User should be able to register	User registered and navigated to registration page	Pass
3	Provide valid password	passwd: Password@20222 confirm_passwd: Password@20222			
4	Click on registration button				
5	Provide Invalid details	name : Geethanjali email: trm4749@gmail.com phone:0000000000 address: Pulikkunnanmukaili			Pass

		gender:female date: 18-07-1999 adhaar:785963256325 place: Alappuzha district:Idukki localbody:Mulipality			
6	Provide Null details	No details provided	User should not be able to login	Enter valid details is displayed	
7	Click on Sign In button				

Post -Condition : User is validated with database and successfully registered with the account.

Code:

```
package test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class register3 {
 public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","C:\\Users\\Teena\\Downloads\\chromedriver_win32
\\chromedriver.exe" );
  WebDriver driver=new ChromeDriver();
  driver.get("http://localhost/emp/register?tag=9mAaBlHVqZoioLgHYUK3Tw==");
  driver.findElement(By.id("name")).sendKeys("Geethanjali");
  driver.findElement(By.id("email")).sendKeys("trm4749@gamil.com");
  driver.findElement(By.id("phone")).sendKeys("7859686235");
  driver.findElement(By.id("address")).sendKeys("Pulikkunnanmukaili");
  driver.findElement(By.id("gender")).sendKeys("female");
  driver.findElement(By.id("date")).sendKeys("18-07-1999");
  driver.findElement(By.id("adhaar")).sendKeys("785963256325");
  driver.findElement(By.id("place")).sendKeys("Alappuzha");
  driver.findElement(By.id("district")).sendKeys("Idukki");
  driver.findElement(By.id("localbody")).sendKeys("Mulipality");
  driver.findElement(By.id("passwd")).sendKeys("Password@20222");
```

```
driver.findElement(By.id("confirm_passwd")).sendKeys("Password@20222");
driver.findElement(By.id("register")).click();
String actualUrl="http://localhost/emp/login";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl))
{
System.out.println("Test passed");
}
else
{
System.out.println("Test failed");
}
}
}
```

login.java | *register3.java | login1.java | register.java

```
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 register3 {
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/emp/register?tag=9mAaBlHVqZoioLgHYUK3Tw==");
10   driver.findElement(By.id("name")).sendKeys("Geethanjali");
11   driver.findElement(By.id("email")).sendKeys("trm4749@gamil.com");
12   driver.findElement(By.id("phone")).sendKeys("7859686235");
13   driver.findElement(By.id("address")).sendKeys("Pulikkunnanmukaili");
14   driver.findElement(By.id("gender")).sendKeys("female");
15   driver.findElement(By.id("date")).sendKeys("18-07-1999");
16   driver.findElement(By.id("adhaar")).sendKeys("785963256325");
17   driver.findElement(By.id("place")).sendKeys("Alappuzha");
18   driver.findElement(By.id("district")).sendKeys("Idukki");
19   driver.findElement(By.id("localbody")).sendKeys("Mulipality");
20   driver.findElement(By.id("passwd")).sendKeys("Password@20222");
21   driver.findElement(By.id("confirm_passwd")).sendKeys("Password@20222");
22   driver.findElement(By.id("register")).click();
23   String actualUrl="http://localhost/emp/login";
24   String expectedUrl= driver.getCurrentUrl();
25   if(actualUrl.equalsIgnoreCase(expectedUrl))
26   {
27   System.out.println("Test passed");
28   }
29   else
30   {
31   System.out.println("Test failed");
32   }
33  }
34 }
```

Problems | Javadoc | Declaration | Console | Coverage

```
<terminated> login1 [Java Application] C:\Users\Teena\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (19-Jul-2022, 11:28:40 am
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 52691
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 11:28:46 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 19, 2022 11:28:46 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed
```

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operation alone. At this stage the main workload, the greatest up Haviland the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion. Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide are liable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

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

6.2 IMPLEMENTATION PROCEDURES

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

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

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

6.2.1 User Training

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

6.2.2 Training on the Application Software

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

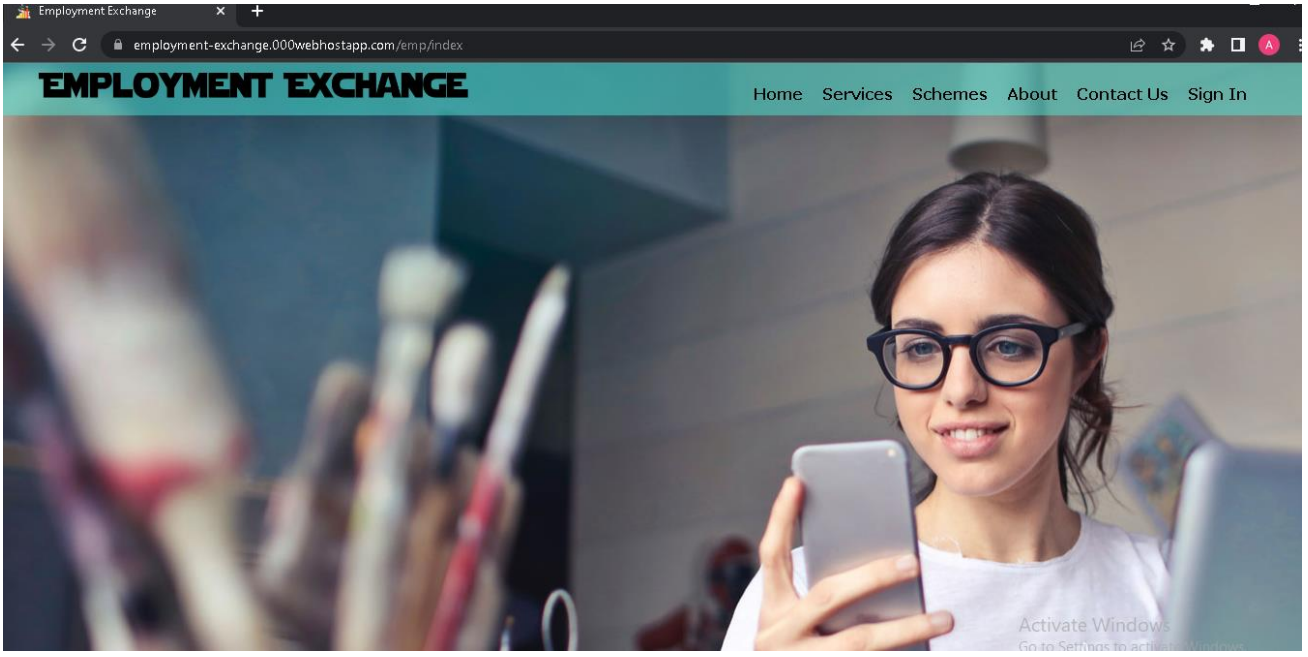
6.2.3 System Maintenance

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

Software maintenance is of course, far more than "Finding Mistakes".

6.2.4 Hosting

Hosting (also known as Web site hosting, Web hosting, and Webhosting) is the business of housing, serving, and maintaining files for one or more Web sites. More important than the computer space that is provided for Web site files is the fast connection to the Internet. Most hosting services offer connections on T-carrier system lines. Typically, an individual business hosting its own site would require a similar connection and it would be expensive.



6.2.5 Hosting Method Steps

- Go to the 000webhost homepage
- Sign up to make an account.
- Make sure you verify your email.
- Click + button to create a new website.
- Enter your details and click create.
- Click Manage Website to start building website.
- Choose a platform for your website.
- In website Builder, Drag and Drop the project files.
- Insert database in Database Management.
- Get the custom Domain name.
- Publish the website.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system employment exchange is to manage the services through computerized system. And the eligible candidate gets notified by the job and their relating details. The company can register through this system and add their vacancy details. After the approval of the officers the job notification are send to the users who has the qualification. If the user applies for one services or scheme they cannot again for that service or scheme. They can edit their account pages. The officers can apply for leave by online. And they get notified of their salary details through mail.

7.2 FUTURE SCOPE

In future the system will ascertaining the eligibility of renewal of the employment exchange registration card of the candidate through online. And the system can be develop in android application .They can add complaints and feedbacks.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”,2009.
- Roger S Pressman, “Software Engineering”,1994.
- PankajJalote, “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
- https://www.bing.com/search?q=php+tutorial&cvid=ecf6e80c06744639a6bf650f62a97f0d&aqs=edge.3.69i57j0l5j69i60l3.4955j0j1&pglt=43&FORM=ANNTA1&PC=U531#

CHAPTER 9

APPENDIX

9.1 Sample Code

Login.php

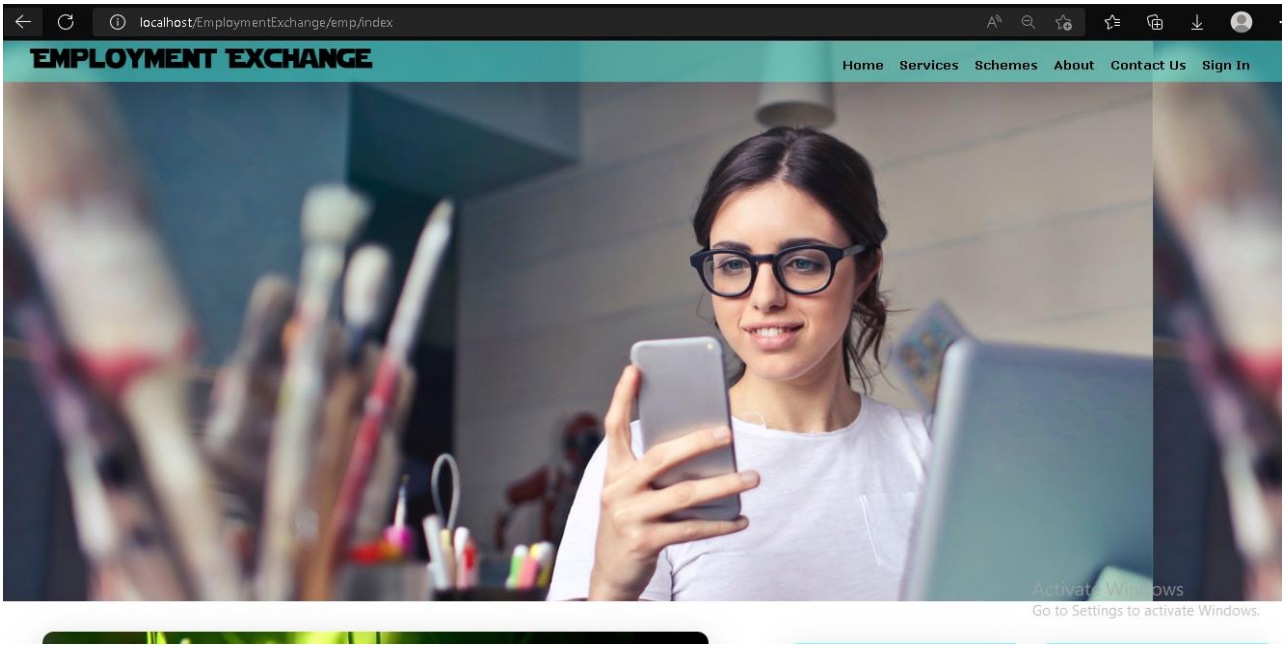
```
<?php

require_once './include/connect.php';
require './encdec.php';

session_start();

if(isset($_POST['login'])){
  $email = mysqli_real_escape_string($conn, $_POST['email']);
  $password = mysqli_real_escape_string($conn, $_POST['password']);

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

  if($row = mysqli_fetch_assoc($result)){

    // For admin
    if($row['usertype']==4){
      $hash = password_verify($password, $row['password']);
      if($hash == true){
        $_SESSION['superuser'] = $email;
        header('Location: superuser/dashboard');
        exit();
      }elseif($hash == false){
        $status = encrypt($userType);
        $error = encrypt("Error");
        header("Location: login?error=$error");
        exit();
      }
    }

    // For normal users
    $active = $row['status'];
    if($active==1){
      $hash = password_verify($password, $row['password']);
      if($hash == false){
        $status = encrypt($userType);
        $error = encrypt("Error");
        header("Location: login?error=$error");
        exit();
      }
      else if($hash == true){
        $_SESSION['regid'] = $row['login_id'];
        $_SESSION['usertype'] = $row['usertype'];
        $_SESSION['verified'] = $row['verified'];
```

```
        header("Location: ./index");
        exit();
       }
     }else{
      echo "<script>";
       echo "alert('Your account is deactivated, Kindly check with Admin to reactivate')";
      echo "</script>";
      header('Location: ./index');
      exit();
     }

   }

}

?>

<!DOCTYPE html>
<html>

<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Login Page</title>
  <link rel="stylesheet" href="./assets/css/css/bootstrap.min.css">
  <link rel="stylesheet" href="./assets/css/css/login.css">
  <script src="./assets/jquery-3.6.0.js"></script>
  <link rel="shortcut icon" type="image/jpg" href="./assets/images/logo.png"/>
</head>

<body>

  <div class="container">
    <div class="row justify-content-md-center" style="margin-top: 10%;">
      <div id="col1" class="col-2"></div>
      <div id="col2" class="col-8 col-md-4 login-main-div"><br><br>
        <center><b class="login-head">Login</b></center><br>
        <?php
          if(isset($_GET['error'])){
            if ($error=="Error"){
              $message = $_GET['error'];
              $message = "Invalid username or password";
        ?>
         <center><div id="alert" class="alert-warning"><?php echo $message ?></div></center><br>
        <?php
            }
          }
        ?>
        <form method="POST" action="" style="width: 100%;">
          <div class="container">
            <input type="email" name="email" class="inp px-3" placeholder="Email id" ><br>
            <input type="password" name="password" class="inp px-3" placeholder="Password"
```

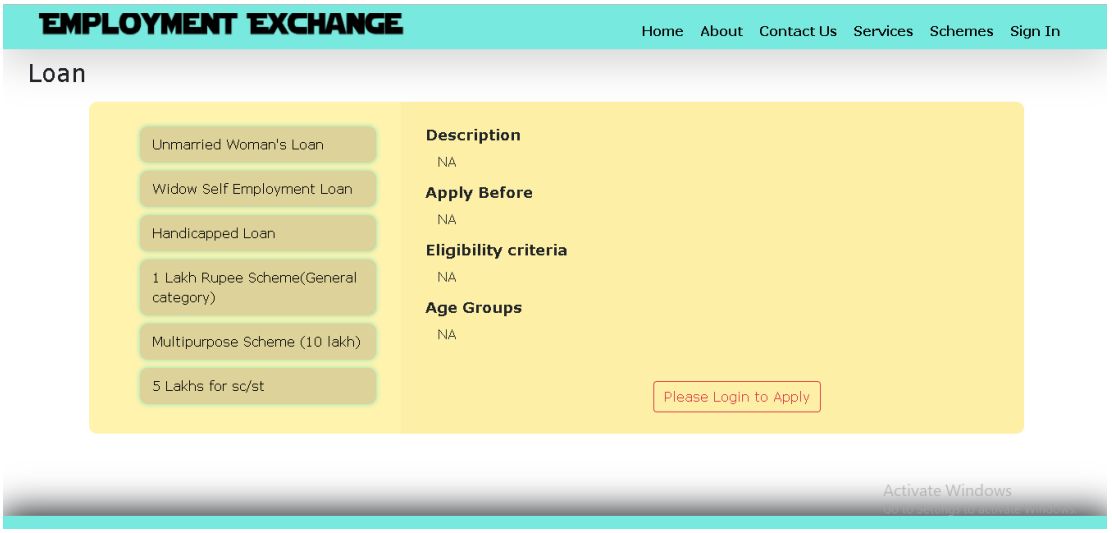
```
><br><br>
          <center><input type="submit" name="login" value="Login" class="btn btn-success"
id="register"></center><br>
        </div>
        <div class="form-row py-4">
          <div class="offset-1 col-lg-10">
            <p  style="color: white;">Dont't have account?   <a href="./register?tag=<?php echo
encrypt("0") ?>" style="color: lightblue;">Sign up</a></p>
            <p  style="color: white;">For company please use   <a href="./register?tag=<?php
echo encrypt("1") ?>" style="color: lightblue;">Sign up</a></p>
          </div>
        </div>
       </form>
      </div>
      <div id="col3" class="col-2"></div>
    </div>
  </div>
</body>

<script>
  $(document).ready(function () {

    if ($(window).width() < 991) {
      $("#col1").attr('class', 'col-1');
      $("#col2").attr('class', 'col-10');
      $("#col3").attr('class', 'col-1');
      $("#col2").addClass('login-main-div');
    }
  });
</script>

</html>
```

9.2 Screenshots

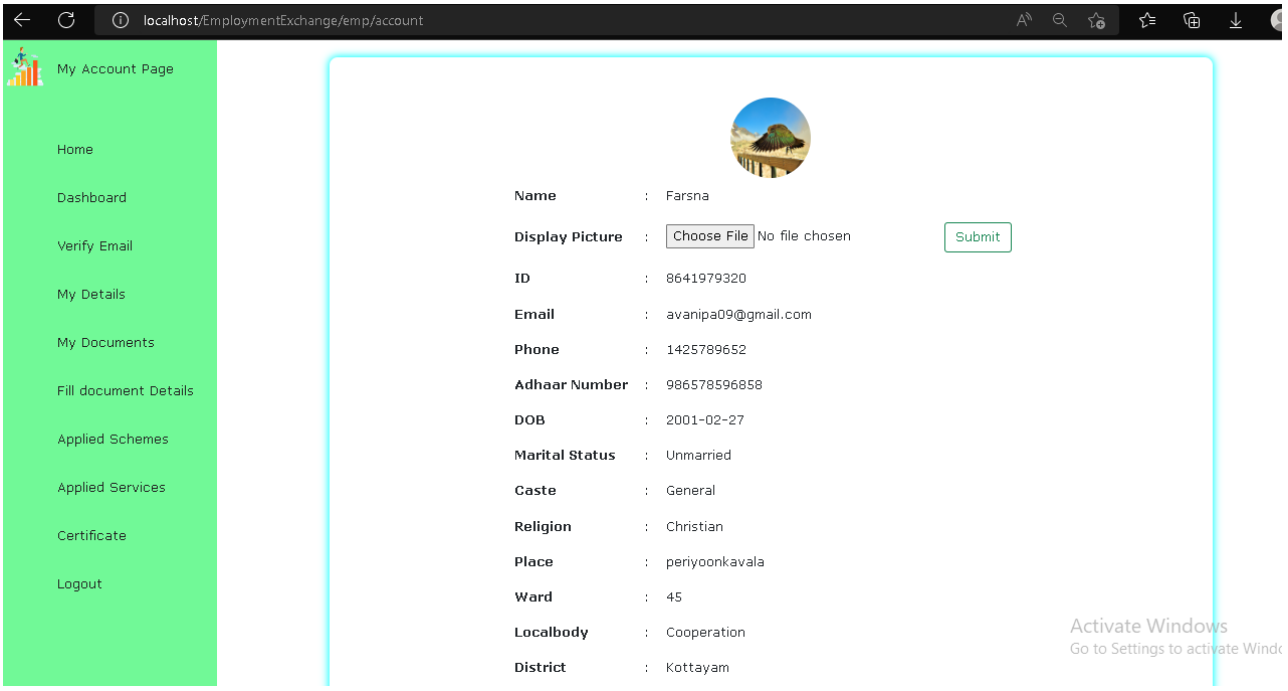
Home page



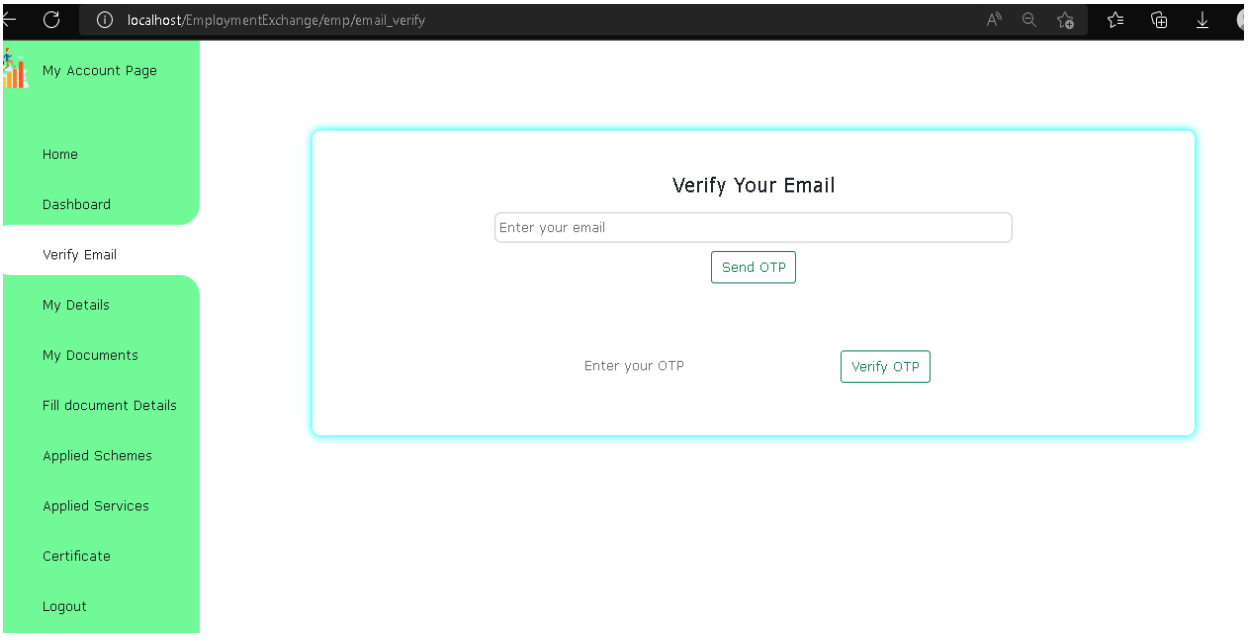
Scheme page



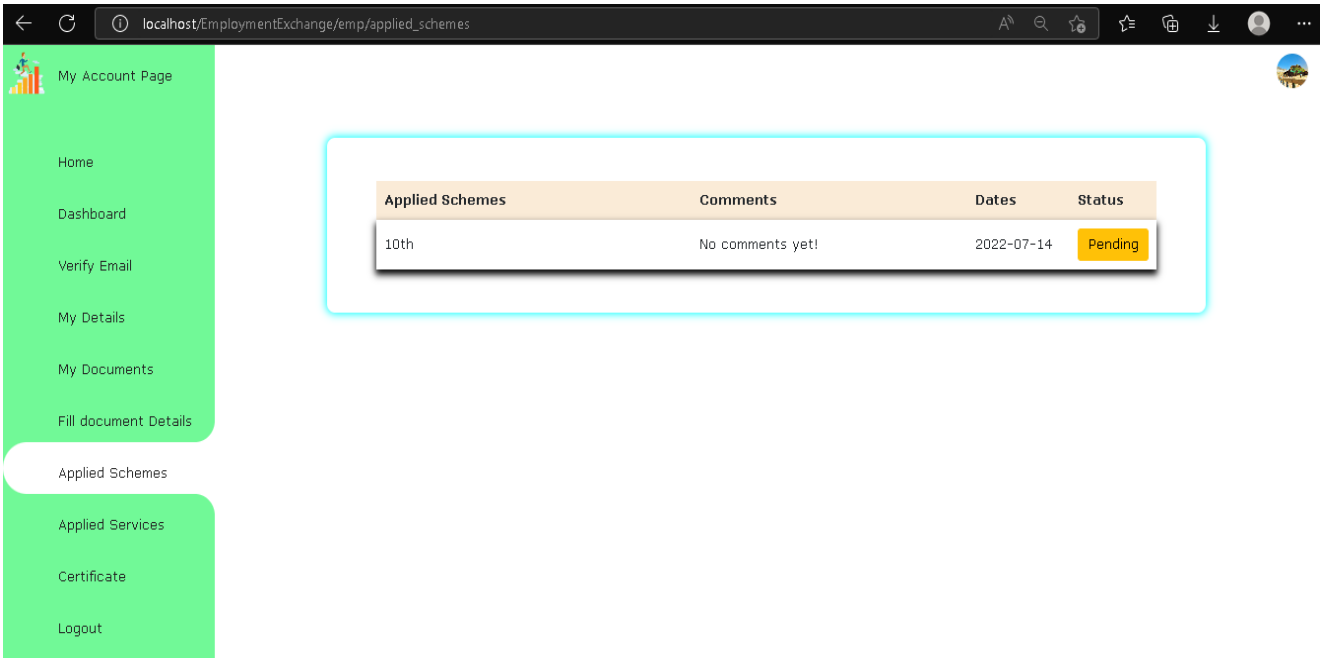
Account Page of User



Verify Email

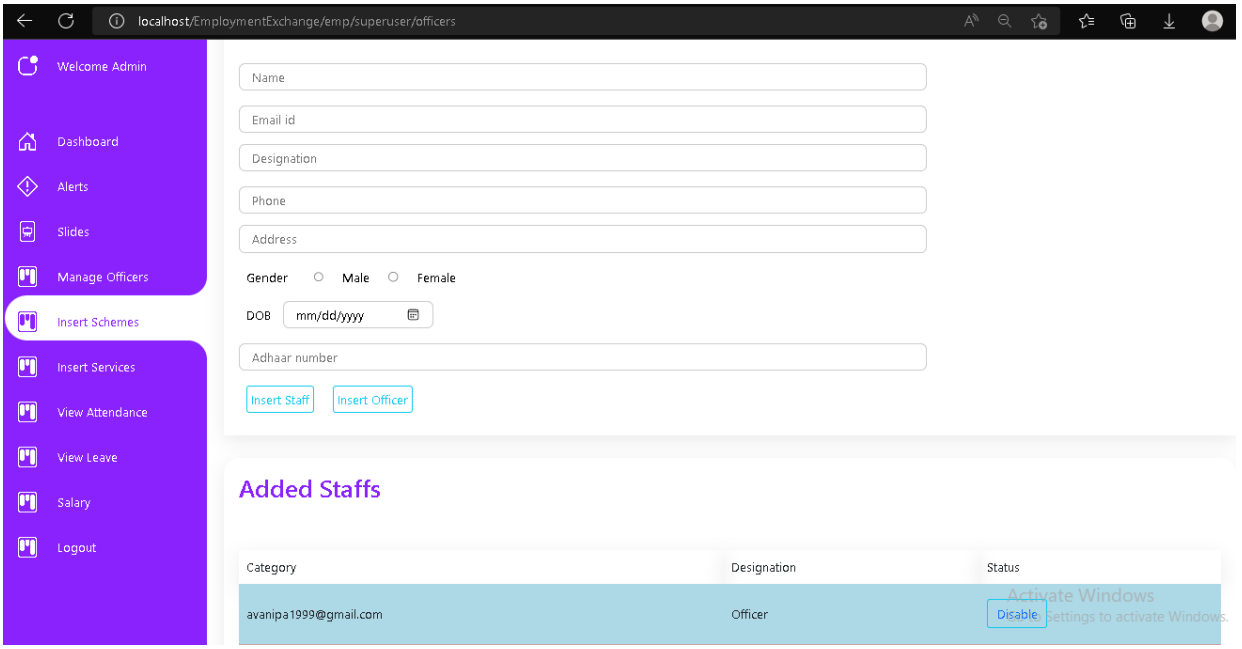


Applied Scheme

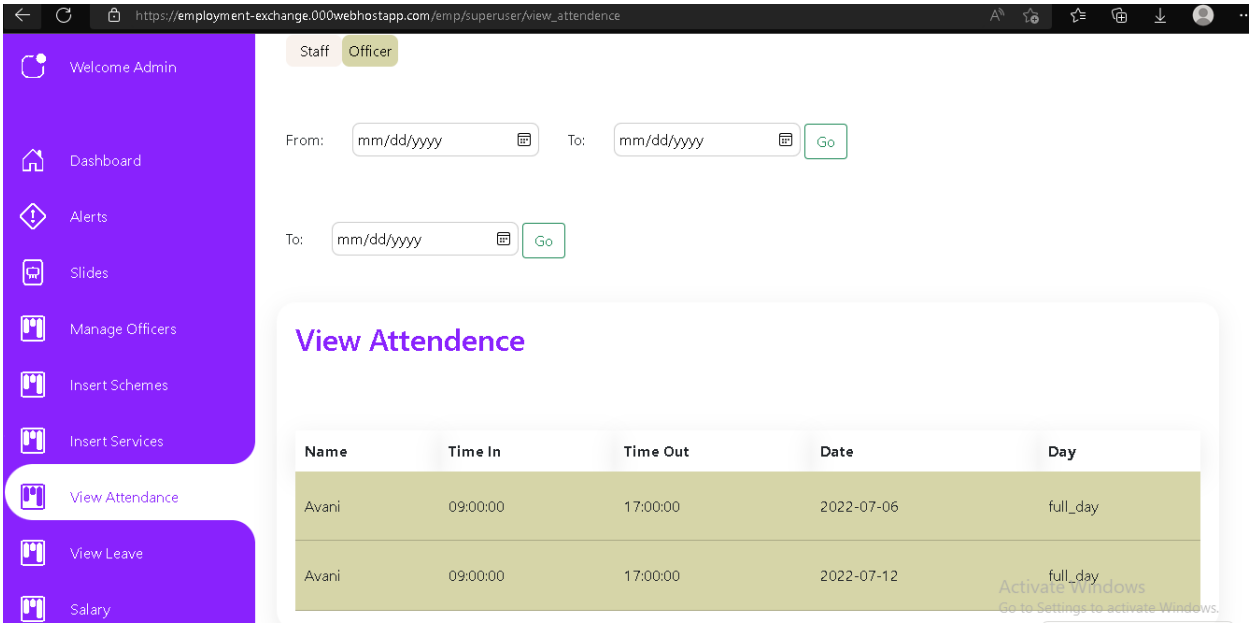


Admin Page

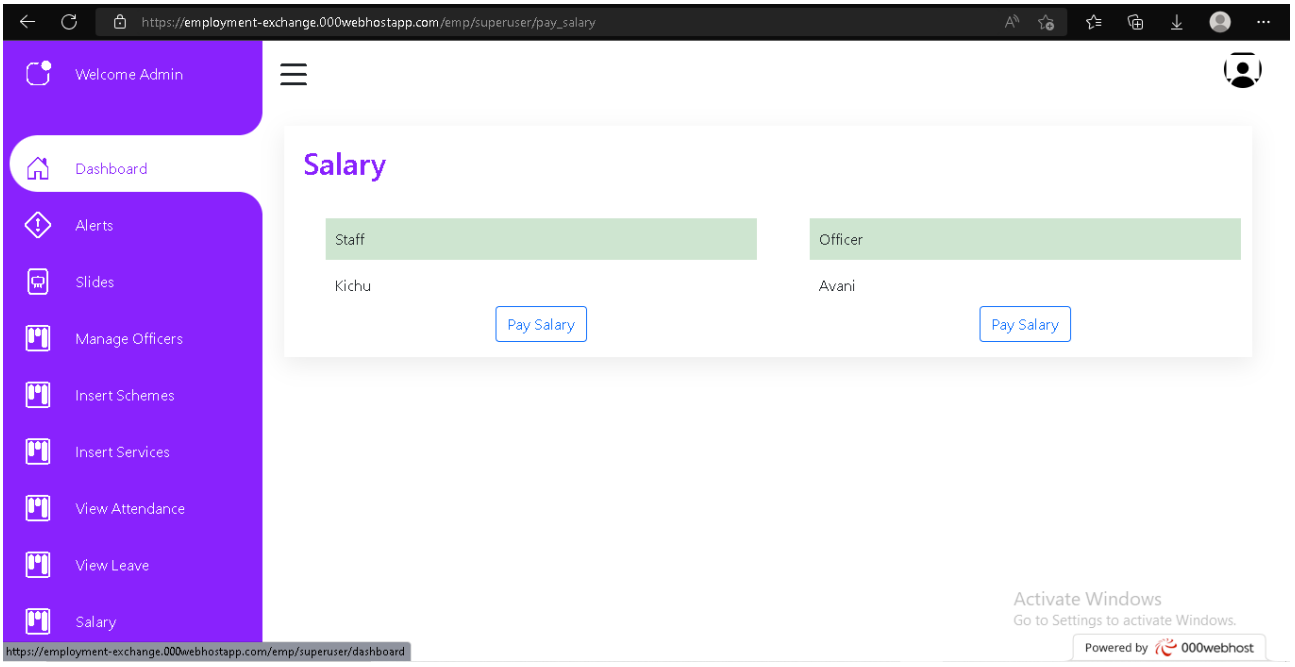
Add Officer/staff



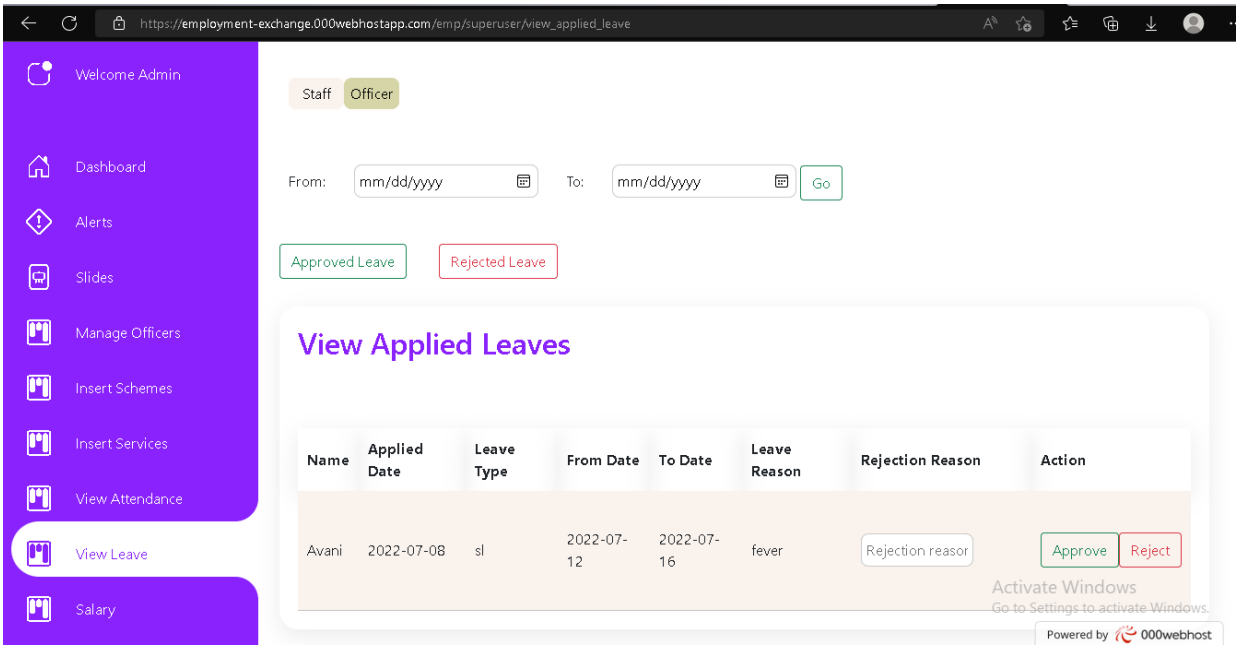
Manage attendance



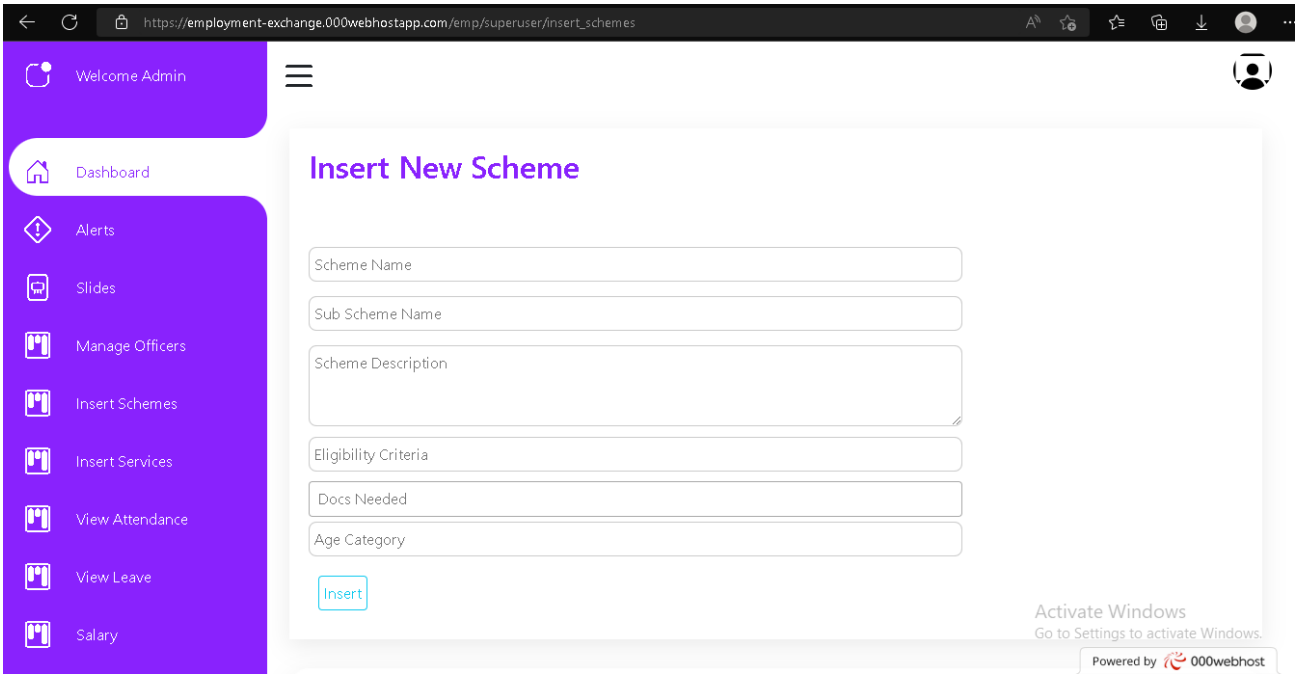
Manage salary



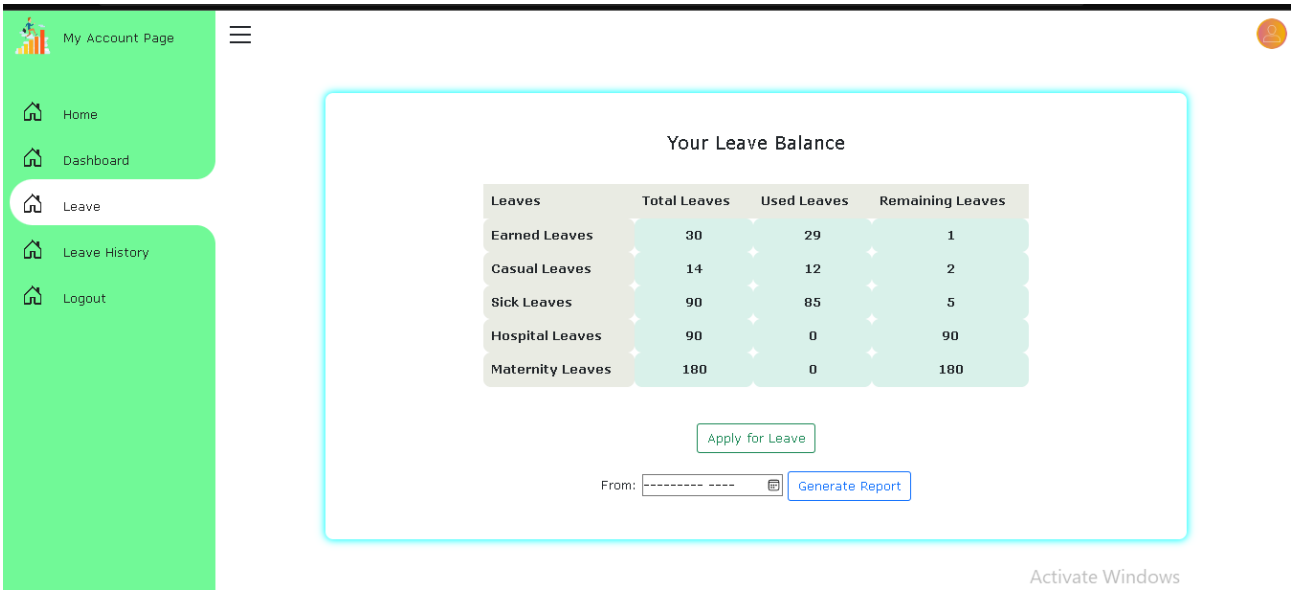
Manage leave



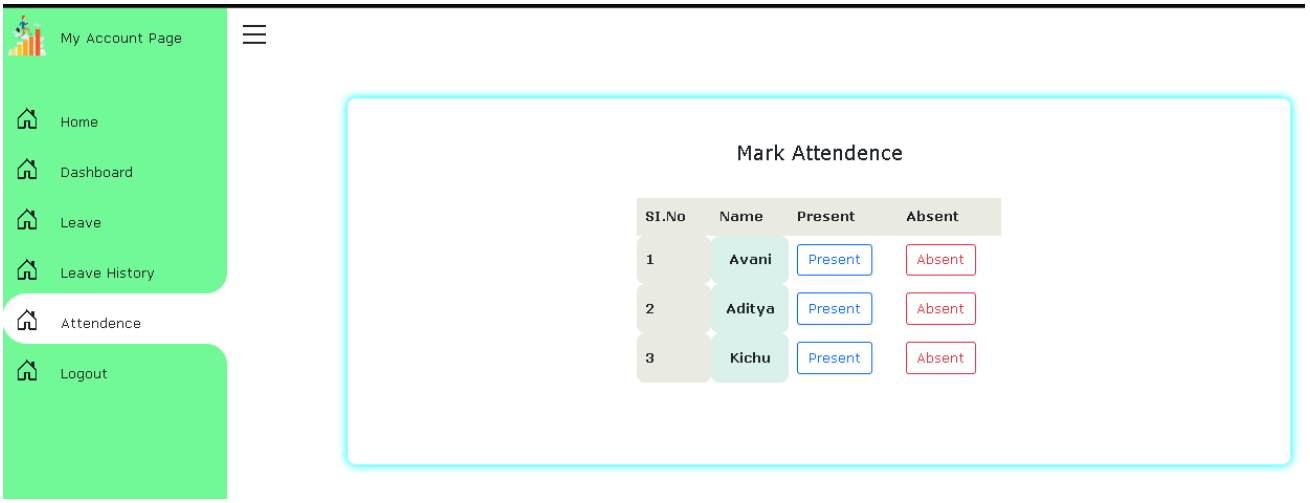
Insert Scheme



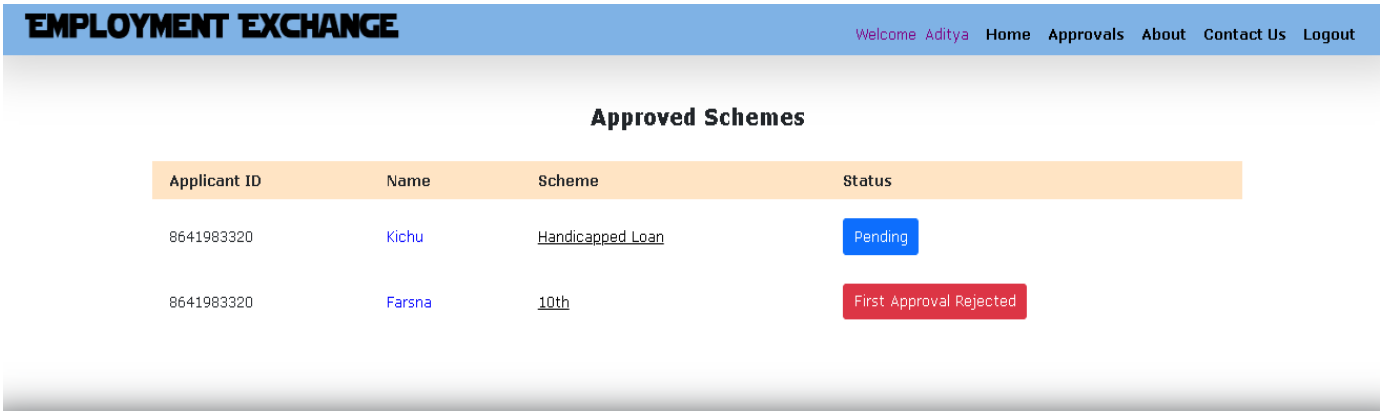
Officer leave



Attendance of staff and officer



Approvals of scheme



9.3 PLAGIARISM REPORT

turnitin **Similarity Report ID:** oid:10159:20022471

PAPER NAME

Employment_Exchange_Automation_System.pdf

WORD COUNT	CHARACTER COUNT
7718 Words	**40529 Characters**
PAGE COUNT	FILE SIZE
49 Pages	**901.4KB**
SUBMISSION DATE	REPORT DATE
Jul 22, 2022 12:07 PM GMT+5:30	**Jul 22, 2022 12:07 PM GMT+5:30**

16% Overall Similarity

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

- 16% Internet database
- 2% Publications database
- Crossref database

Excluded from Similarity Report

- Small Matches (Less then 10 words)

Summary

turnitin

Similarity Report ID: oid:10159:20022471

16% Overall Similarity

Top sources found in the following databases:

- 16% Internet database
- 2% Publications database
- Crossref database

TOP SOURCES

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

1	issuu.com Internet	7%
2	seminarprojects.com Internet	1%
3	ia801708.us.archive.org Internet	1%
4	tutorialspoint.com Internet	<1%
5	agustian9000.blogspot.com Internet	<1%
6	geeksforgeeks.org Internet	<1%
7	slideshare.net Internet	<1%
8	Ilona Veitaite, Audrius Lopata. "Chapter 12 Knowledge-Based UML Dyn... Crossref	<1%
9	dev.mysql.com Internet	<1%

Sources overview

turnitin **Similarity Report ID:** oid:10159:20022471

10	192.192.246.204 Internet	<1%
11	researchpublish.com Internet	<1%
12	justblogme.com Internet	<1%
13	cs.worcester.edu Internet	<1%
14	irjet.net Internet	<1%
15	docslib.org Internet	<1%
16	documents.mx Internet	<1%
17	ici2016.org Internet	<1%
18	seminarprojects.com Internet	<1%
19	dartmouth.edu Internet	<1%
20	docshare.tips Internet	<1%
21	justanswer.com Internet	<1%

Sources overview

turnitin

Similarity Report ID: oid:10159:20022471

22	Ilona Veitaite, Audrius Lopata. "Chapter 3 Knowledge-Based Generatio... Crossref	<1%
23	m.mu.edu.sa Internet	<1%
24	ucertify.com Internet	<1%
25	repository.president.ac.id Internet	<1%
26	tudr.thapar.edu:8080 Internet	<1%

Sources overview