CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

E Wed is a web application for matrimonial that, provides the efficient matrimonial website, a provision for viewing the latest status of the service. It is a free matrimonial website. The web service provides you the maximum benefit by providing the information about various users located at various places, there by managing them effectively.

This system provides services in web server and bride and groom. That needs a particular person can register their name in the system and get the username and password through which they can find their soul mates.System also provides provision to view the profile of the registered user. provided that they give correct username and password.

1.2 PROJECT SPECIFICATION

The proposed system is a website in which user can register and view matching profiles online. Also that the customers did not need to search for profiles everywhere instead simply need to the web were he/she can view all the vehicles and their details

The system includes 5 modules. They are:

1. Registration
2. Login
3. Matches
   - Add,View, edit and delete profiles
   - View matched profiles
   - Searching for profiles

4. Message
5.Payment

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

E Wed is a web application for matrimonial that, provides the efficient matrimonial website, a provision for viewing the latest status of the service. It is a free matrimonial website. The web service provides you the maximum benefit by providing the information about various users located at various places, there by managing them effectively.

This system provides services in web server and bride and groom. That needs a particular person can register their name in the system and get the username and password through which they can find their soul mates. System also provides provision to view the profile of the registered user. provided that they give correct username and password.

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. Customer can register and they can specify the preference for matching more accurately if needed. Each customer can create their own profile. The proposed system rectify the drawbacks of the present system. The main problem is regarding time as well as cost. The cost of advertisement and searching of the partners through newspapers and consultancy are difficult.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system customers can view all information about his profiles, matches , preference ,bill etc.

2.3 DRAWBACKS OF EXISTING SYSTEM

- No proper online management of system
- It is difficult to maintain important information.
- More manual hours need to generate required reports.

2.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for growth of service center; on such consideration the system is proposed. In our proposed system there is admin who can view all the customers. It allows customers to make their service and do their transactions by using online payment method. Users of this proposed system are admin , admin-users and customer.

The software programme that reduces the amount of time needed to manually maintain records and provide reports This programme maintains the data in a central location that is simultaneously accessible to all users. Managing historical data in a database is extremely simple. To utilise this programme, distributors don't need any special training. They may easily utilise the tool that reduces the number of manual hours needed to complete routine tasks, improving performance. The databases may easily be updated with information on online sales and purchases.

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:

- **Eliminate the chance for critical data misuse: -**

Only Admin have access to the vehicle owners and users' data, no other members are permitted. Therefore, the data will be kept private.

- **Ensure data accuracy: -**

The suggested solution does away with human mistakes made when entering user information during registration.

- **Better service: -**

Hard copy storage won't be a strain for the system. For performing the same activity, 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

In this stage, the project's viability is evaluated, and a business proposal is presented with a very generic project plan and some cost projections. The proposed system's practicality must be investigated during system analysis. This will guarantee that the suggested solution won't burden the business. Understanding the main system requirements is crucial for the feasibility analysis.

Three key considerations involved in the feasibility analysis are:

3.1.1 Economical Feasibility

This study is being done to see what kind of financial impact the organization's system will have. The corporation has a finite amount of money to invest in the system's research and development. The costs must be supported by evidence. As a result, the developed system came in under budget, which was made possible by the fact that most of the technology were public domain. All that must be purchased are the customized goods.

3.1.2 Technical Feasibility

This study is being done to evaluate the system's technological requirements, or its technical viability. Any system that is created must not place a heavy burden on the available technical resources. As a result, the client will face high expectations. The created system must have a low bar for implementation, requiring either no changes at all or very few.

3.1.3 Behavioral Feasibility

The goal of the study is to determine how much the user accepts the system. This includes the instruction needed for the user to operate the system effectively. The system shouldn't make the user feel threatened; instead, they should view it as a need. The techniques used to inform and acquaint the user with the system are the only factors that affect the level of acceptance by the users. As the system's ultimate user, his confidence must be increased so that he may offer some helpful criticism, which is encouraged.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

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

3.2.2 Software Specification

Front End	-	HTML, CSS ,BOOTSTRAP,JQUERY
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server-side scripting language used for general-purpose programming as well as web development. More than 244 million websites and 2.1 million web servers currently use PHP. The PHP group now produces the reference implementation of PHP, which was first developed by Rasmus Ledorf in 1995. PHP, a recursive acronym that once meant for personal Home page, now stands for PHP: Hypertext Preprocessor. A web server's PHP processor module interprets PHP code to produce the final web page. In order to handle data, PHP commands can be directly inserted into an HTML source file rather than calling an external file. Due to limitations on the use of the name PHP, it has also developed to include a command-line interface capability and can be used standalone, which makes it incompatible with the GNU General Public License (GPL). Most web servers support the free deployment of PHP, which is also available as a standalone shell on practically all platforms and operating systems.

3.3.2 MySQL

Oracle Corporation created, distributed, and provided support for MySQL, the most well-known Open-Source SQL database management system. The most recent details regarding MySQL software are available on the MySQL website.

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

A systematic collection of data is called a database. It might be anything, such as a straightforward grocery list, a photo gallery, or the enormous amount of data in a business network. A database management system like MySQL Server is required in order to add, access, and process data that is stored in a computer database. Database management systems, whether used as stand-alone programmes or as a component of other applications, are essential to computing because computers are excellent at processing vast volumes of data.

- **MySQL databases are relational.**

Instead of placing all the data in one huge warehouse, a relational database keeps the data in individual tables. Physical files that are designed for speed contain the database structures. The logical model provides a flexible programming environment with objects like databases, tables, views, rows, and columns. One-to-one, one-to-many, unique, compulsory or optional, and "pointers" between distinct tables are a few examples of the rules you might build up to regulate the relationships between various data fields. 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. MySQL stands for "Structured Query Language" with the SQL prefix. The most popular standard language for accessing databases is SQL. Depending on your programming environment, you might explicitly enter SQL (for example, to generate reports), incorporate SQL statements into other languages' code, or use a language-specific API that obscures the SQL syntax. By way of the ANSI/ISO SQL Standard, SQL is defined. Since its inception in 1986, the SQL standard has gone through various iterations. In this document, "SQL92" refers to the 1992 standard, "SQL: 1999" to the 1999 standard, and "SQL: 2003" to the most recent version of the standard. The SQL Standard as it exists at any one time is referred to as "the SQL standard."

- **MySQL software is Open Source.**

Anyone can use and modify the software because it is open source. The MySQL software is available for free download and usage online by anyone. You are free to examine the source code and modify it as necessary. The GPL (GNU General Public License) is used by the MySQL software to specify what you are allowed to do and are not allowed to do with the software in certain circumstances. You can purchase a commercially licenced version from us if the GPL makes you uncomfortable or if you need to integrate MySQL code into a for-profit application. For further details, see the MySQL Licensing Overview.

- **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 utilise 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.**

The MySQL Database Software is a client/server system that includes a multi-threaded SQL server that supports several client programmes and libraries, administration tools, and a broad variety of application programming interfaces (APIs). Additionally, we provide MySQL Server as an integrated multi-threaded library that you can link into your programme to create a standalone offering that is smaller, faster, and simpler to operate.

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." One way to describe it is as the process of using different methodologies and concepts to specify a device, a process, or a system in enough detail to allow for its physical reality. Regardless of the development paradigm that is employed, software design forms the technical core of the software engineering process. The architectural detail needed to construct a system or product is developed through the system design. This programme has also through the best possible design phase, fine tuning all efficiency, performance, and accuracy levels, as in the case of any systematic technique. A user-oriented document is converted into 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.

Unified Modeling Language is known as UML. Compared to other popular programming languages like C++, Java, COBOL, etc., UML is unique. A visual language called UML is used to create software blueprints. A general-purpose visual modelling language for software system visualization, specification, construction, and documentation is what UML is known as. 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 been standardized to the point where it is now an OMG standard. A comprehensive UML diagram that depicts a system is made up of all the elements and relationships. The most crucial aspect of the entire procedure is the UML diagram's aesthetic impact. It is completed by using all the additional components. The following nine diagrams are part of UML.

- 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 organising 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. UML (Unified Modeling Language), a standard language for the modelling of real-world objects and systems, uses use case diagrams.

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 task focused on customer support are just a few examples of system objectives. For instance, use cases in a product sales context can involve customer service, item ordering, catalogue updating, and payment processing. There are four elements in a use case diagram.

• The actors, often people involved in the system who are defined according to their roles; the boundary, which establishes the system of interest in relation to its environment.

• The relationships between and among the actors and the use cases, which are the precise roles played by the players within and around the system.

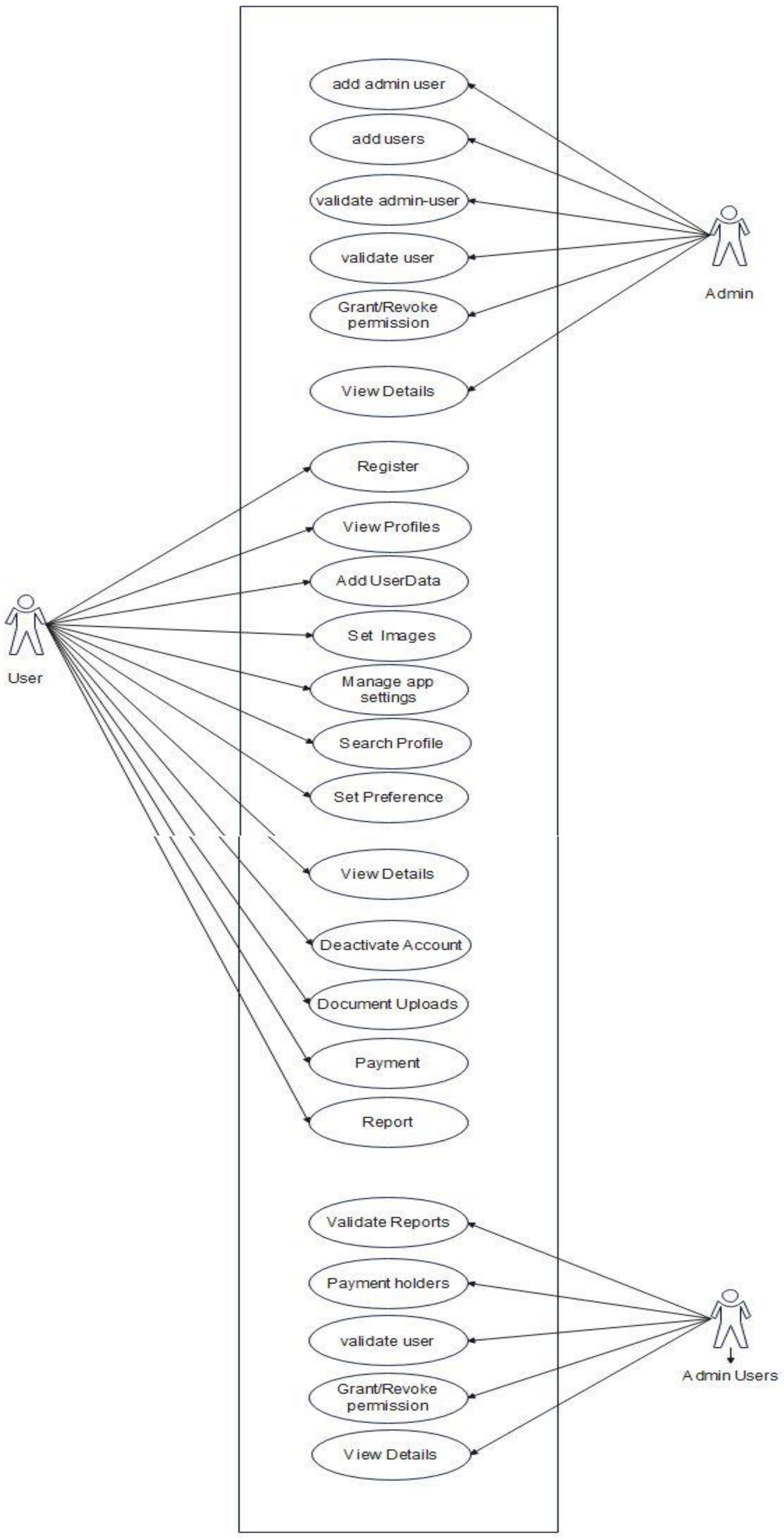
Use case diagrams are created to depict a system's functional requirements. To create an effective use case diagram after identifying the aforementioned things, we must adhere to the following rules.

• 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 the actors names that fit them.

• Clearly depict links and dependencies in the diagram.

• Keep in mind that the diagram's primary function is to indicate the needs; do not attempt to include all possible relationships.


add admin user
add users
validate admin-user
validate user
Grant/Revoke permission
View Details
Admin
Register
View Profiles
Add UserData
Set Images
Manage app settings
Search Profile
Set Preference
View Details
Deactivate Account
Document Uploads
Payment
Report
User
Validate Reports
Payment holders
validate user
Grant/Revoke permission
View Details
Admin Users

4.2.2 SEQUENCE DIAGRAM

A sequence diagram essentially shows how things interact with one another sequentially, or the order in which these interactions occur. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence diagrams show the actions taken by the components of a system in chronological order. Businesspeople and software engineers frequently use these diagrams to record and comprehend the requirements for new and current systems.

Sequence Diagram Notations –

i. **Actors –** In a UML diagram, an actor represents a particular kind of role in which it communicates with the system's objects. An actor is always beyond the purview of the system that we want to use the UML diagram to represent. We employ actors to portray a variety of roles, including those of human users and other outside subjects. In a UML diagram, an actor is represented using a stick person notation. In a sequence diagram, there might be several actors.

ii. **Lifelines –** A named piece that shows a specific participant in a sequence diagram is called a lifeline. In essence, a lifeline represents each incident in a sequence diagram. The top of a sequence diagram is where the lifeline pieces are placed.

iii. **Messages –** Using messages, communication between objects is demonstrated. The messages are displayed on the lifeline in chronological sequence. Arrows are how messages are represented. A sequence diagram's main components are lifelines and messages.

Messages can be broadly classified into the following categories:

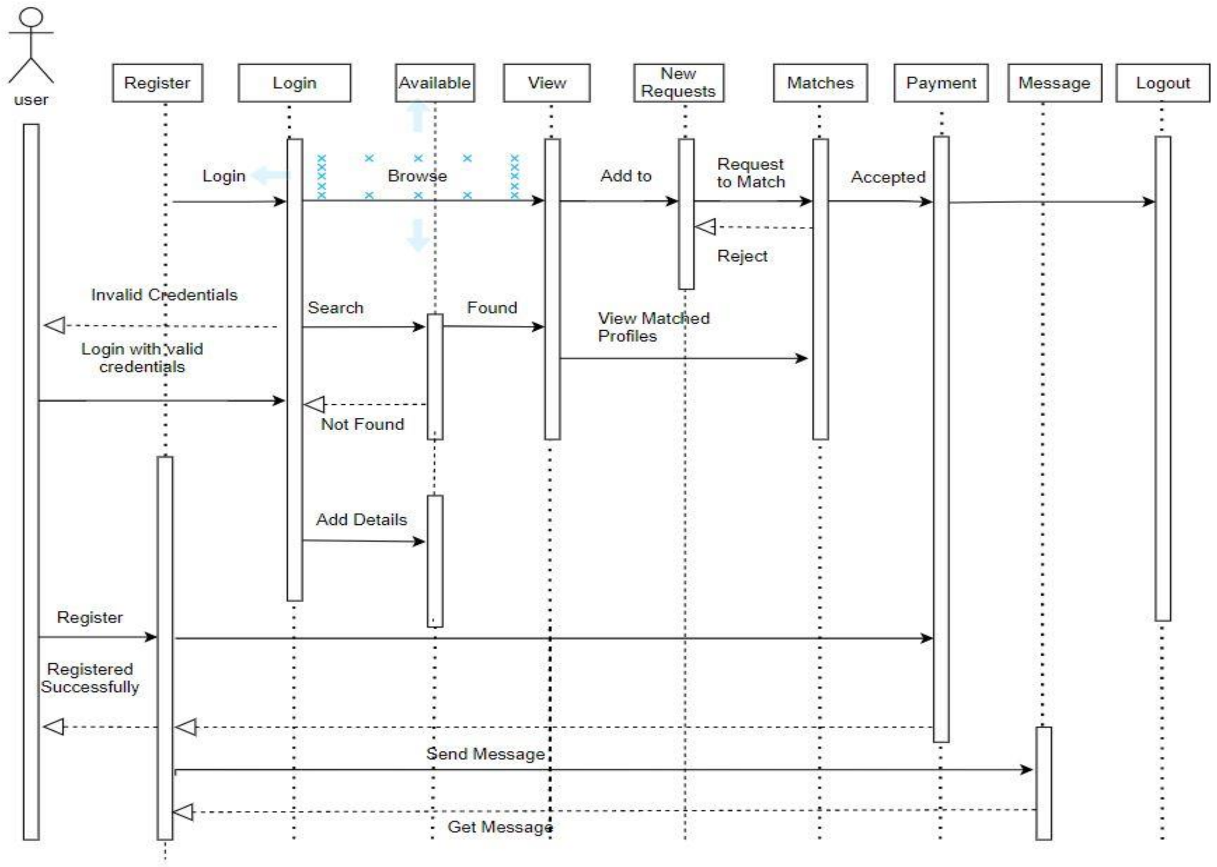
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message

- Reply Message
- Found Message

iv. **Guards –** In the UML, we utilise 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 –

- They are also used to display specifics of UML use case diagrams, as well as to describe and depict the logic underlying complex functions, operations, or procedures.
- Used to comprehend the precise operation of present or upcoming systems.
- Visualize the flow of information between different system elements or objects.

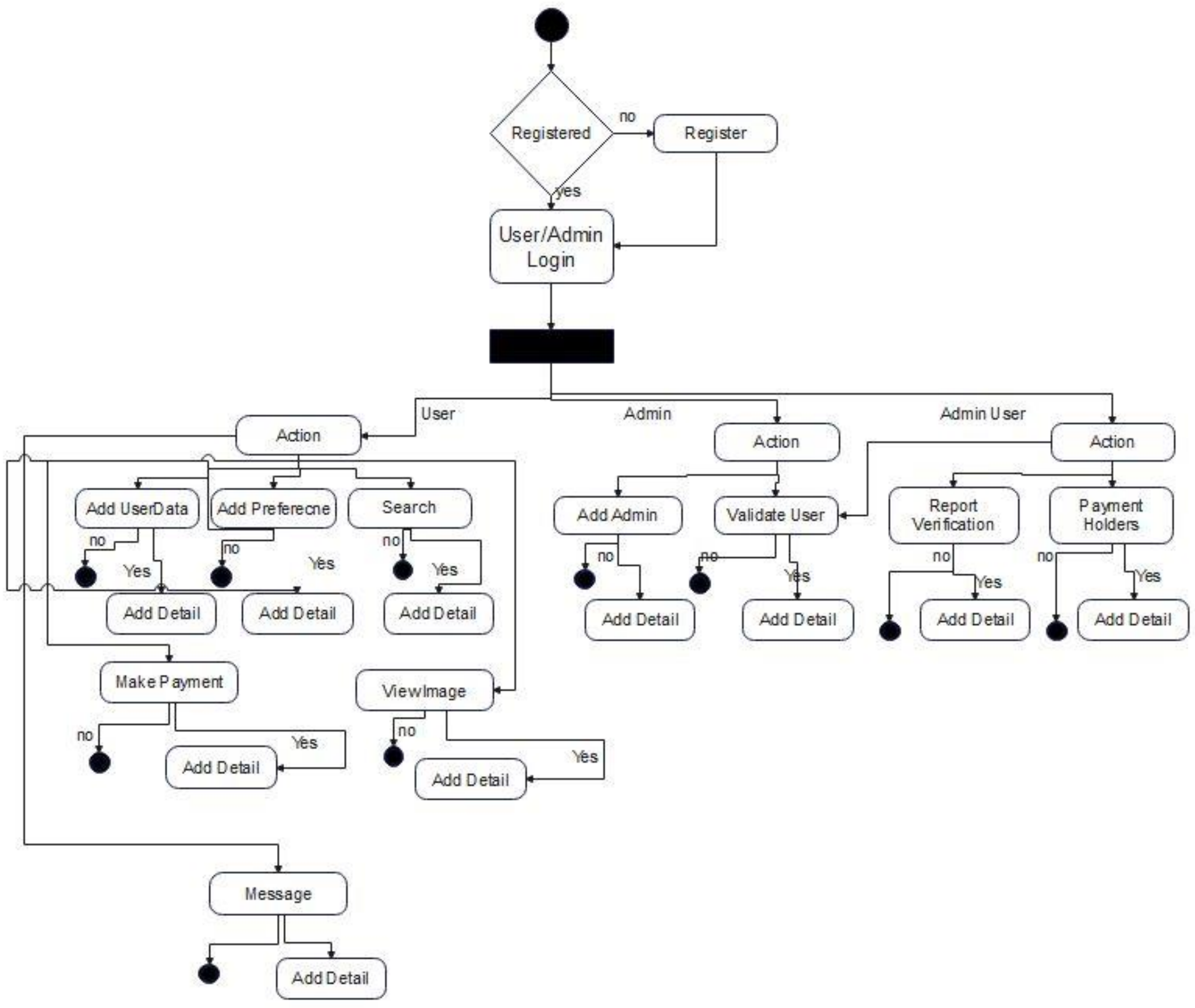


4.2.3 ACTIVITY DIAGRAM

An activity diagram is essentially a flowchart that shows how one activity leads to another. The action might be referred to as a system operation.

One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams use many features, such as fork, join, etc., to cope with all types of flow control.

Fig 3 : Activity Diagram for E-Wed

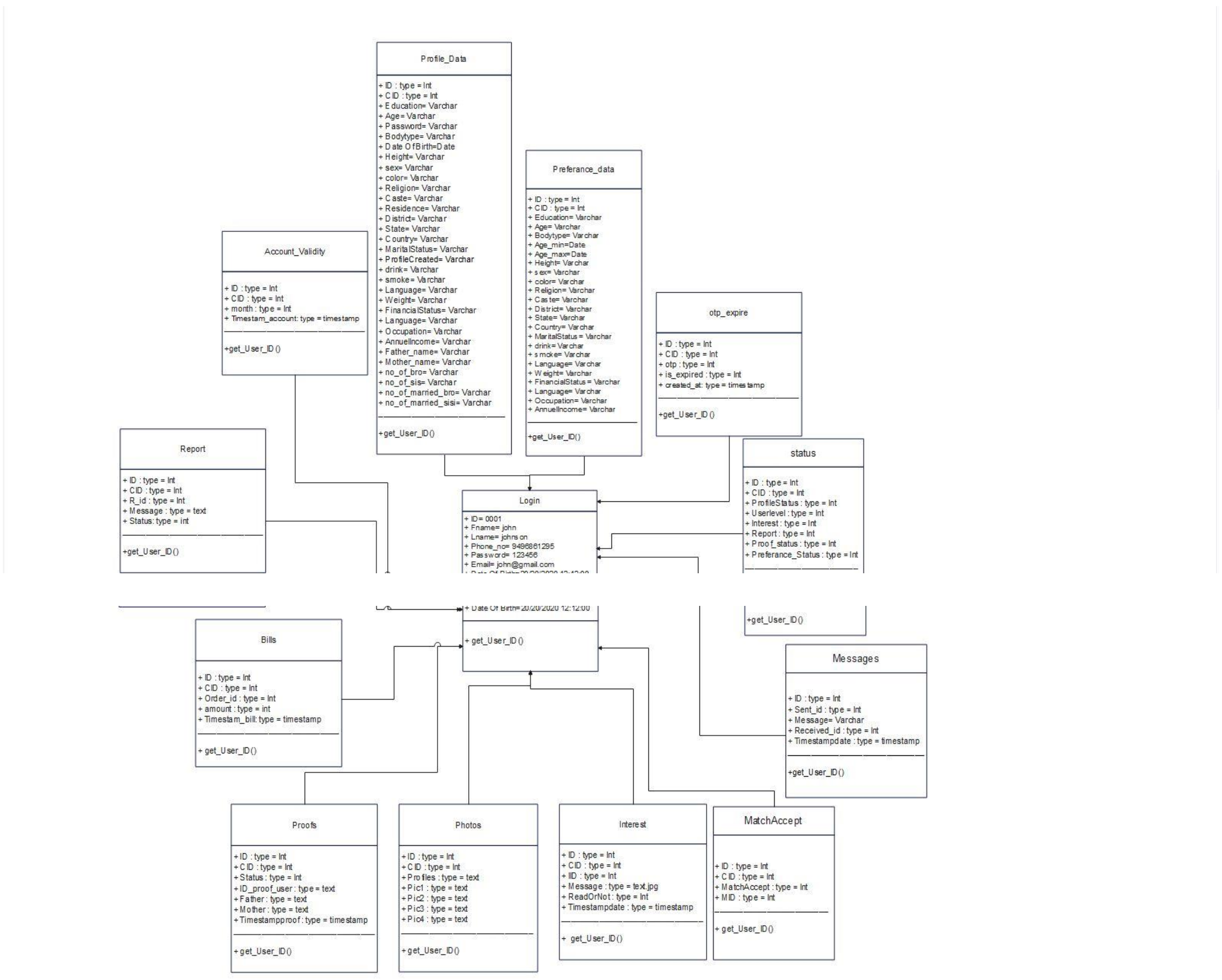


4.2.4 CLASS DIAGRAM

Static diagrams include class diagrams. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualising, 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 translated with object-oriented languages, class diagrams are extensively utilised in the modelling of object-oriented systems.

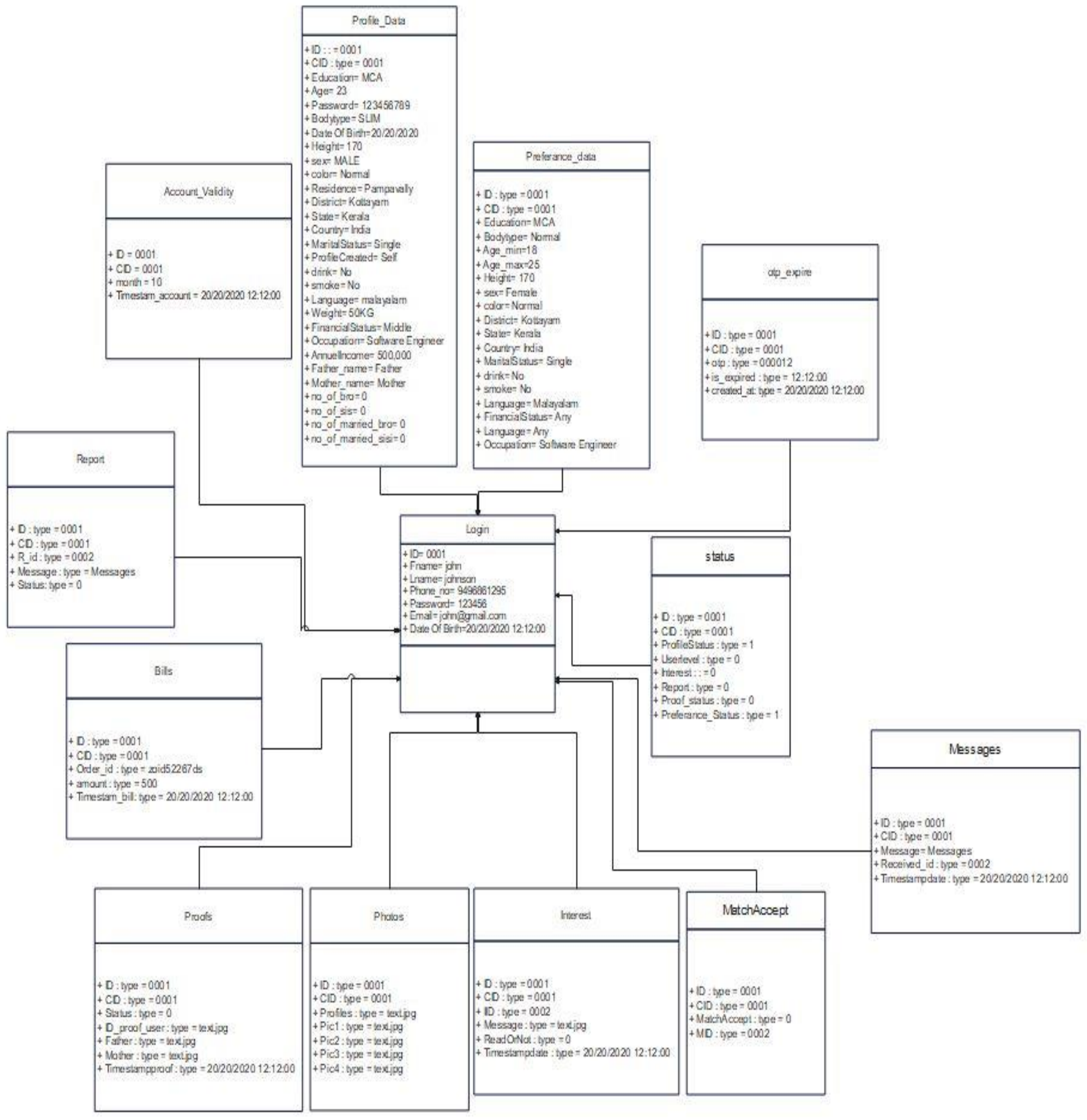
Fig 4 : Class diagram for E-Wed



4.2.5 OBJECT DIAGRAM

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.

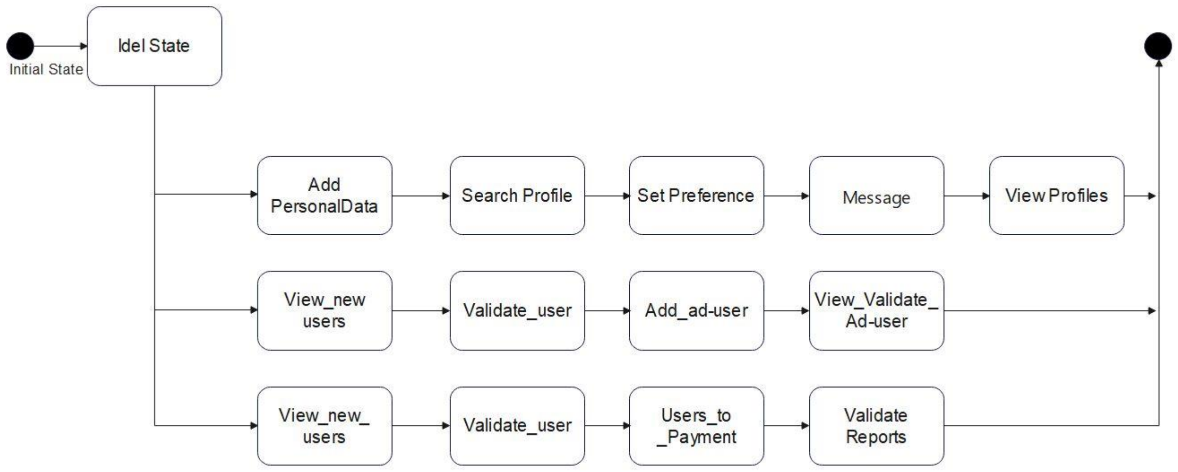
Fig 5 : Object diagram for E-Wed



4.2.6 STATECHART DIAGRAM

One of the five UML diagrams used to depict a system's dynamic nature is the statechart diagram. Throughout an object's existence, they define several states, and these states are altered by events. The reactive systems can be modelled with statechart diagrams. A system that reacts to internal or external events is known as a reactive system.

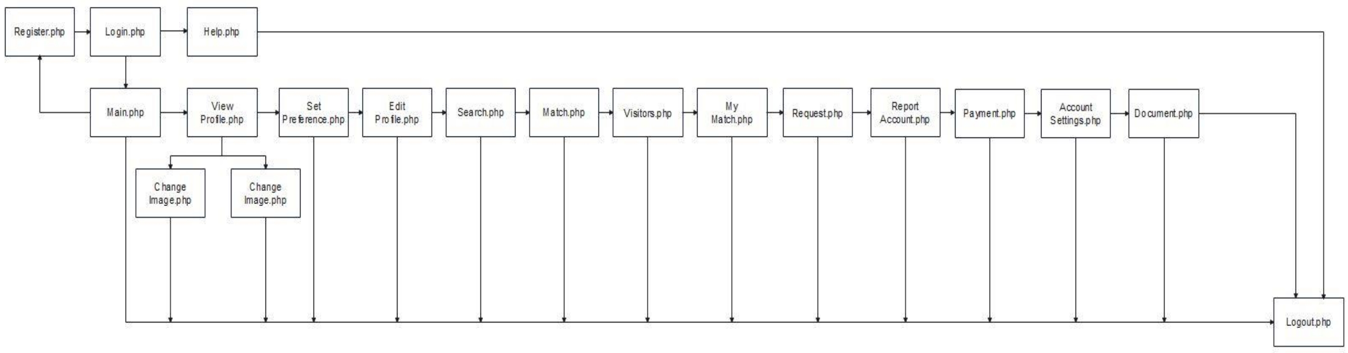
Fig 6 : Statechart diagram for E-Wed



4.2.7. COMPONENT DIAGRAM

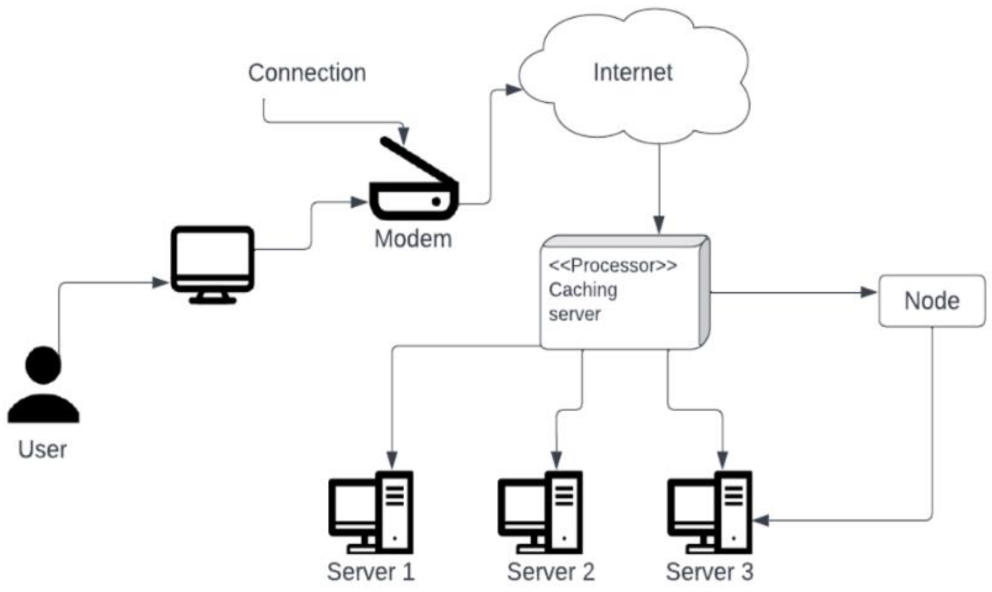
A component diagram, often called a UML component diagram, shows how the physical parts of a system are wired up and organised. Component diagrams are frequently used to model implementation details and confirm that all necessary system functions have been accounted for.

Fig 6 : Component diagram for E-Wed



4.2.8. DEPLOYMENT DIAGRAM

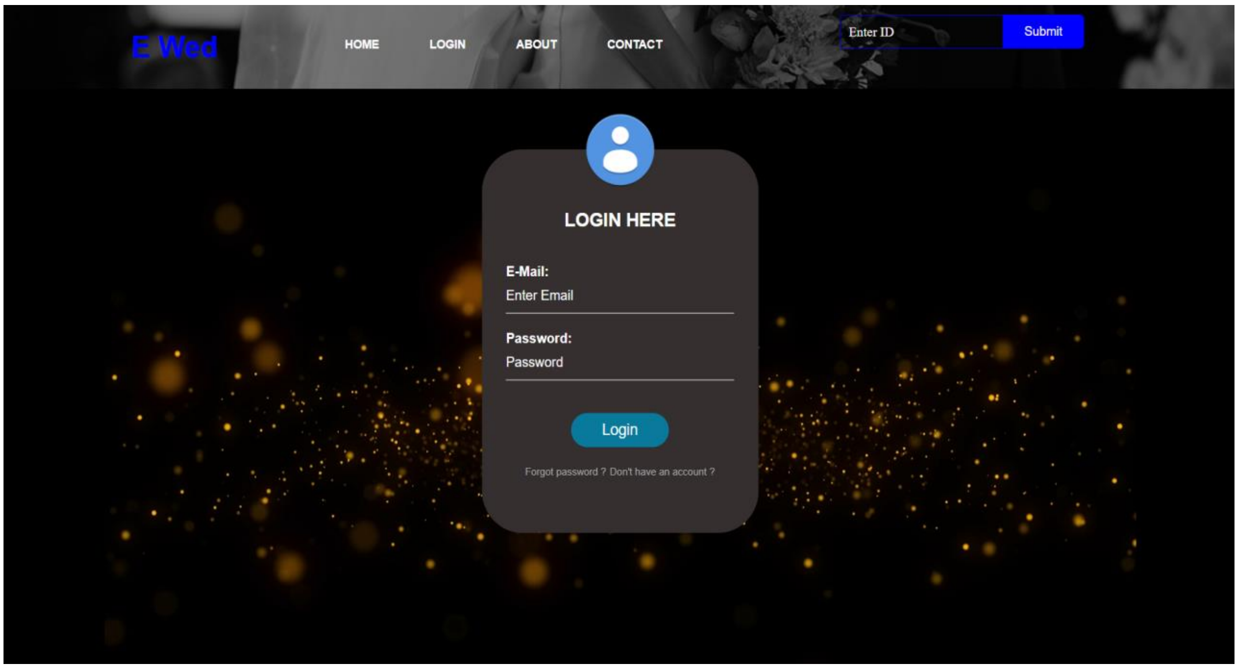
An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system.



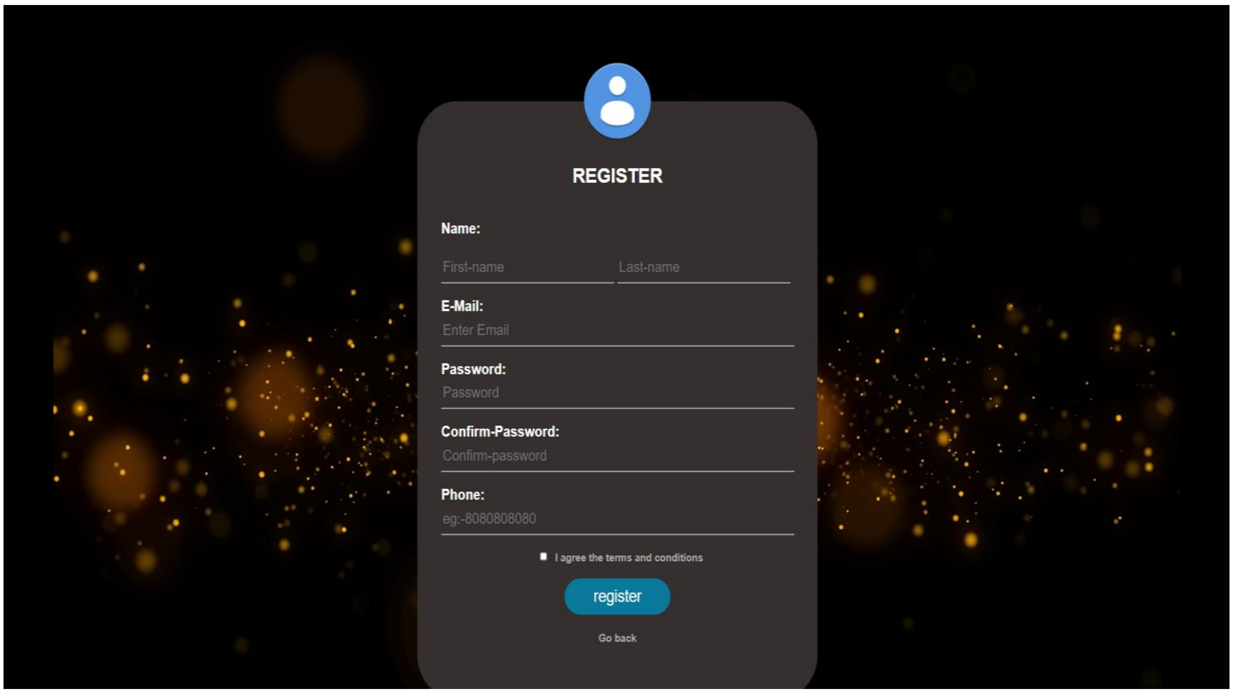
4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

User Login



User Registration



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 construct the system in issue is converted from an information level design to a design in the second stage. 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 arrangement aims to accomplish the two main goals listed below.

- 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 database 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 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: 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 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 normalising data structures in a way that reduces duplication and fosters integrity. Using the normalisation 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 normalised. 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 in a tuple must have a single value from the attribute's domain and its domain must only include atomic values. The 1NF forbids "relations within relations" or "relations as attribute values within tuples," in other words. By 1NF, only single atomic or indivisible values are allowed for attribute values. Putting the data into First Normal Form is the first step. Data that is of a similar type in each table can be moved into separate tables to solve this problem. According to the project's requirements, a Primary Key or Foreign Key is assigned to each table. For each nested or non-atomic attribute, we create additional relations in this. 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 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 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

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. A relation is only considered to be in third normal form if it is in second normal form, and furthermore, the relation's non-key characteristics should not depend on other non-key attributes.

TABLE DESIGN

Login

Primary key **id**

SI No	Name	Type	Description
1.	id	Int(10)	id
2.	fname	Int(10)	Fname
3.	lname	Varchar(20)	Lname
4.	email	Varchar(20)	Email
5.	password	Varchar(20)	Password
6.	phone	Varchar(20)	Phone
7.	timestamp	Varchar(20)	Timestamp

Interest

Primary key **id**

Foreign key **cid,iid** references id (login)

SI No	Name	Type	Description
1.	id	Int(20)	Id
2.	cid	Int(10)	Cid[sender]
3.	iid	Varchar(20)	Iid [receiver]
4.	message	Varchar(20)	message
5.	ReadOrNot	Varchar(20)	ReadOrNot
6.	timeStampDate	Varchar(30)	TimeStampDate
7.	description	Varchar(50)	Description

Matchaccept

Primary key : **id**

Foreign key **cid,mid** references id (login)

SI No	Name	Data Type	Description
1.	id	Int(20)	id
2.	cid	Varchar(20)	cid[sender]
3.	MatchAccept	Varchar(20)	MatchAccept Or Not
4.	mid	Varchar(20)	mid[receiver]

Messages

Primary key **id**

Foreign key **sent_by,received_by** references id (login)

SI No	Name	Data Type	Description
1.	id	Int(20)	Id
2.	sent_by	Int(10)	Sent_by
3.	Received_by	Varchar(50)	Received_by
4.	message	Varchar(20)	Message
5.	CreatedAt	Varchar(20)	CreatedAt

Partnerdetails

Primary key **id**

Foreign key **cid** references id (login)

SI No	Name	Data Type	Description
1.	id	Int(20)	Id
2.	cid	Int(20)	User id
3.	agemin	Int(20)	Year
4.	agemax	Int (20)	Year
5.	maritalstatus	Varchar(20)	Marital staus
6.	heightmin	Int (20)	Height
7.	heightmax	Int (30)	Height
8.	religion	Varchar(20)	Religion
9.	caste	Varchar(20)	Caste
10.	mothertounge	Varchar(25)	Language
11.	educationqualification	Varchar(100)	Education Qualification
12.	occupation	Varchar(20)	Occupation Description
13.	country	Varchar(20)	Country
14.	District	Varchar(20)	District
15.	State	Varchar(20)	State
16.	Drink	Varchar(20)	Drink
17.	Smoke	Varchar(20)	Smoke
18.	Color	Varchar(20)	Color
19.	Financialstatus	Varchar(20)	Financial status
20.	bodytype	Varchar(20)	Bodytype

OTP_expire

Primary key **id**

Foreign key **cid** references id (login)

SI No	Name	Data Type	Description
1.	Id	Int(20)	Id
2.	otp	Int(20)	Otp
3.	cid	Int(20)	cid
4.	Is_expired	Int(20)	Expire time
5.	Created_at	datetime	Otp created time

Photo

Primary key **id**

Foreign key **cid** references id (login)

SI No	Name	Data Type	Description
1.	Id	Int(20)	Id
2.	cid	Int(20)	cid
3.	profiles	Text	Picture
4.	Pic1	Text	Picture
5.	Pic2	Text	Picture
6.	Pic3	Text	Picture
7.	Pic4	text	Picture

Profiledata

Primary key **id**

Foreign key **cid** references id (login)

SI No	Name	Data Type	Description
1.	id	Int(20)	Id
2.	cid	Int(20)	User id
3.	age	Int(20)	Year
4.	occudiscribtion	Varchar(20)	Occupation details
5.	maritalstatus	Varchar(20)	Marital staus
6.	height	Int (20)	Height
7.	annualincome	Int (30)	Annualincome
8.	religion	Varchar(20)	Religion
9.	caste	Varchar(20)	Caste
10	mothertounge	Varchar(25)	Mothertounge
11	educationqualification	Varchar(100)	Educationqualification
12	occupation	Varchar(20)	Occupation
13	country	Varchar(20)	Country
14	District	Varchar(20)	District
15	State	Varchar(20)	State
16	Drink	Varchar(20)	Drink
17	Smoke	Varchar(20)	Smoke
18	Color	Varchar(20)	Color
19	Financialstatus	Varchar(20)	Financial status
20	bodytype	Varchar(20)	Bodytype
21	noofbro	Int(20)	Noofbro
22	noofsis	Int(20)	Noofsis
23	marriedbro	Varchar(20)	Marriedbro
24	marriedsis	Varchar(20)	Marriedsis

Status

Primary key **id**

Foreign key **cid** references id (login)

SI No	Name	Data Type	Description
1.	Id	Int(2)	Id
2.	cid	Int(2)	cid
3.	profilestatus	Int(2)	Profilestatus
4.	userlevel	Int(2)	Userlevel
5.	interest	Int(2)	Interest
6.	report	Int(2)	Report
7.	Proof_status	Int(2)	Proof_status
8.	Preference_status	Int(2)	Preference_status

Tbl_account_validity

Primary key **id**

Foreign key **cid** references id (login)

SI No	Name	Data Type	Description
1.	Id	Int(2)	Id
2.	cid	Int(2)	cid
3.	month	Int(2)	month
4.	Timestamp_account	Timestamp	Timestamp_account

Tbl_report

Primary key **id**

Foreign key **cid** references id (login)

SI No	Name	Data Type	Description
1.	Id	Int(2)	Id
2.	cid	Int(2)	Cid
3.	rid	Int(2)	Rid
4.	message	text	Messages
5.	status	Int(15)	Status

tlb_proof

Primary key **id**

Foreign key **cid** references id (login)

SI No	Name	Data Type	Description
1.	Id	Int(2)	Id
2.	cid	Int(2)	cid
3.	status	Int(2)	status
4.	Id_proof_pic	Int(2)	User id proof
5.	Imestamp_proof	Int(2)	Interest
6.	father	Int(2)	User father id proof
7.	mother	Int(2)	User mother id proof

tlb_bill

Primary key **id**

Foreign key **cid** references id (login)

SI No	Name	Data Type	Description
1.	Id	Int(2)	Id
2.	cid	Int(2)	cid
3.	Order_id	Int(2)	Payment Id
4.	amount	Int(2)	Amount Payed
5.	Bill_timestamp	timestamp	When the payment occured

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the process of carefully controlling the execution of software in order 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. They are:

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

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

If a 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

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

5.1 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 programme, 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). Testing's precise goals should be laid forth in quantifiable language. The test plan should include information on the mean time to failure, cost to find and fix defects, remaining defect density or frequency of occurrence, and test work hours per regression test.

The levels of testing include:

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

5.2.1 Unit Testing

The smallest unit of software design—the software component or module—is the focus of unit testing, which concentrates verification work. Important control pathways are examined in order to find faults inside the module's border using the component level design description as a guide. The scope set for unit testing and the relative complexity of tests. Unit testing can be carried out simultaneously for numerous components and is white-box focused. Information flow into and out of the programme unit under test is monitored by the modular interface to ensure appropriate operation. To make sure that temporary data is kept in its original form during all phases of an algorithm's execution, the local data structure is inspected. To confirm that each 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 other test, tests of data flow over a module interface are necessary. All other tests are irrelevant if data cannot enter and depart the system properly. An important duty during the unit test is the selective examination of execution pathways. Error circumstances must be foreseen in good design, and error handling paths must be put 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 programme structure that has been determined by design using unit tested components. The programme as a whole is tested. Correction is challenging since the size of the overall programme 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 integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive programme structure also developed when discrepancies in programme structures were eliminated.

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. To completely exercise all functional requirements for a programme, the software engineer can create sets of input conditions using Black Box testing.

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 under consideration has its user acceptance assessed; in this case, it must meet the needs of the company. When creating and making modifications as needed, the software should stay in touch with the user and viewpoint system. The following points are considered in this:
Output screen designs, input screen designs, and
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 system under study is evaluated using the test data after it has been prepared. 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.

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 doesexactly what it was designed to do. It tests for bugs, defects, and any other issues that can arisewith 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 sequences can be incredibly in-depth, and provide detailedreports 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. Automation testing makes this process an easier one. It's also able to analyzea 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.

5.2.5 Selenium Testing

Selenium is an open-source tool that automates web browsers. It provides a single interface that 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.

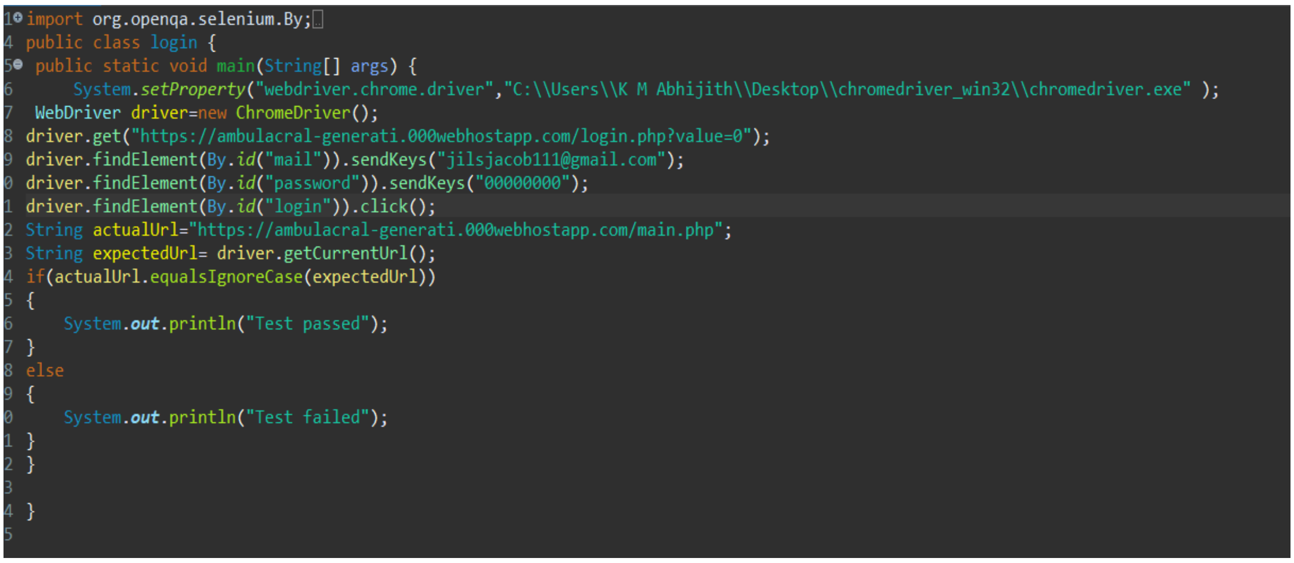
Test cases for a Login Page

Project Name: E-Wed	
Login Test Case	
Test Case ID: Fun_1	**Test Designed By: Jilse Jacob**
Test Priority (Low/Medium/High): High	**Test Designed Date:** 17-07-2022
Module Name: Login Screen	**Test Executed By: Dr. Juby Mathew**
Test Title: Verify login with valid username and password	**Test Execution Date:** 18-05-2022
Description: Test the Login Page	
Pre-Condition: User has valid username and password	

Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigation to Login Page		Login Page should be display ed	Login page displayed	Pass
2	Provide Valid username	Username : jilsjacob111@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboad	Pass
3	Provide Valid Password	Password: 00000000			
4	Click on Sign In button				
5	Provide Invalid username or password	Username: jilsjacob111@gmail.com Password: 00000000	User should not be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null username or Password	Username : null Password: null			
7	Click on Sign In button				

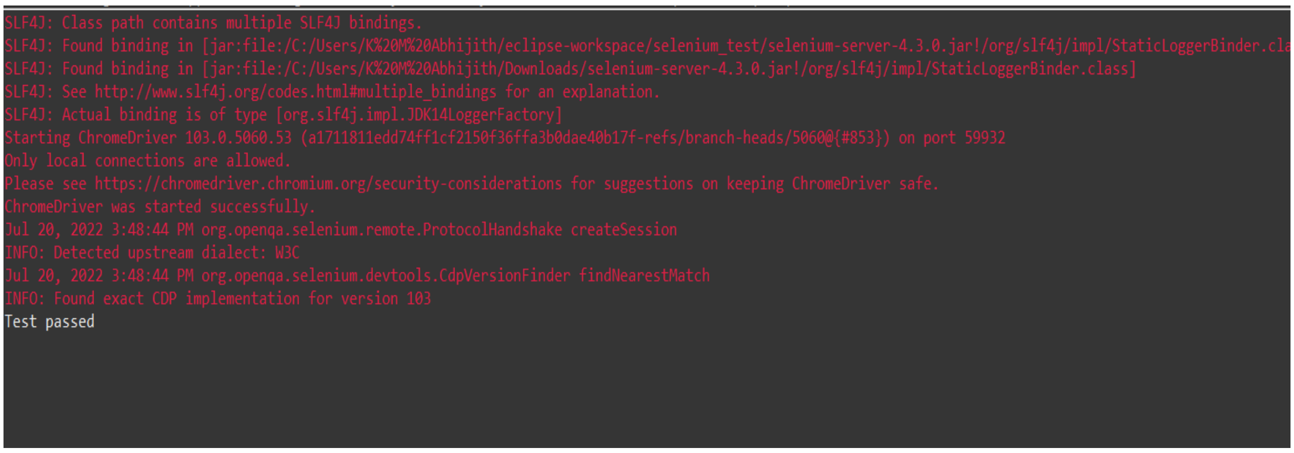
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database.

Code package



```
import org.openqa.selenium.By;
public class login {
 public static void main(String[] args) {
     System.setProperty("webdriver.chrome.driver","C:\\Users\\K M Abhijith\\Desktop\\chromedriver_win32\\chromedriver.exe" );
 WebDriver driver=new ChromeDriver();
driver.get("https://ambulacral-generati.000webhostapp.com/login.php?value=0");
driver.findElement(By.id("mail")).sendKeys("jilsjacob111@gmail.com");
driver.findElement(By.id("password")).sendKeys("00000000");
driver.findElement(By.id("login")).click();
String actualUrl="https://ambulacral-generati.000webhostapp.com/main.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl))
{
    System.out.println("Test passed");
}
else
{
    System.out.println("Test failed");
}
}

}
```



```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/Users/K%20M%20Abhijith/eclipse-workspace/selenium_test/selenium-server-4.3.0.jar!/org/slf4j/impl/StaticLoggerBinder.cla
SLF4J: Found binding in [jar:file:/C:/Users/K%20M%20Abhijith/Downloads/selenium-server-4.3.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.JDK14LoggerFactory]
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 59932
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 20, 2022 3:48:44 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 20, 2022 3:48:44 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 newly revised system design into an operational one, and it simply refers to placing a new system design into operation.

The user department now bears the most of the workload, faces the most disruption, and has the biggest 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 that satisfies organisational needs must be implemented properly. 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 system analysis and design work needed to implement the three key components of education and training, system testing, and changeover will increase in complexity as a system is implemented.

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

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 first have doubts regarding the software, but we must

Make sure that there isn't an increase in resistance because one must make sure that:
The new system's advantages must be known to the active user. Their faith in the software is increased.
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. 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. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports, and execute other important tasks through user training. 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

6.2.1 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

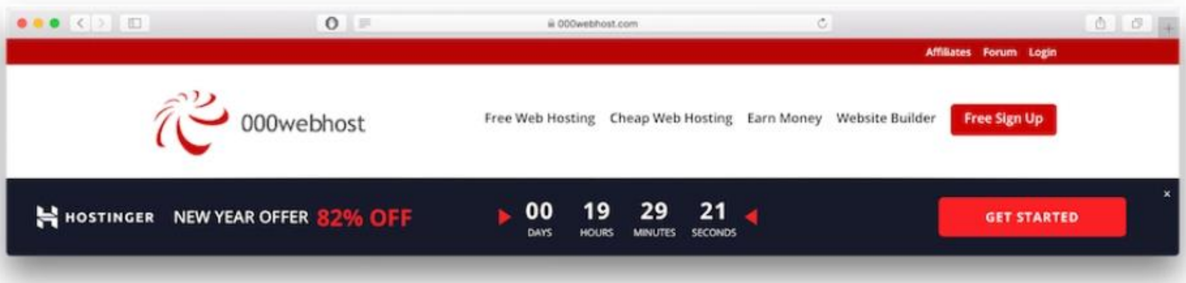
When a facilitating supplier distributes space on a web server for an online site to store its files, they are facilitating an online site. Web facilitating makes the records that include web site (code, pictures, etc.) accessible for seeing online. Every website you've ever gone to is facilitated on a server. The sum of space apportioned on a server to website depends on the sort of facilitating. The most sorts of facilitating are shared, committed, VPS and affiliate. They are separated by the kind of innovation utilized for the server, the level of administration given and the extra administrations on offer.

000Webhost

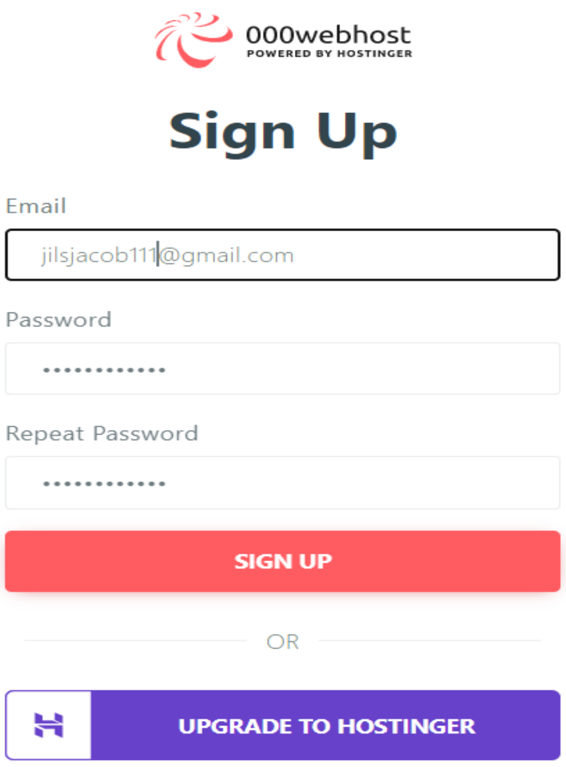
Free web facilitating tends to be so constrained in capabilities and highlights that client must pay to induce what they need. However, 000webhost may be a free website facilitating arrangement that gives an cluster of important highlights, counting web site builder, WordPress back, and no advertisements. Clients can overhaul to a paid arrange to induce indeed more highlights and back, but based on our surveys, 000webhost is the most excellent free web facilitating arrangement for those who are genuinely on a tight budget.

Setting up website in 000webhost

Step 1: Head on to 000webhost.com



Step 2 : Sign up using valid email



Sign Up

Email

jilsjacob111@gmail.com

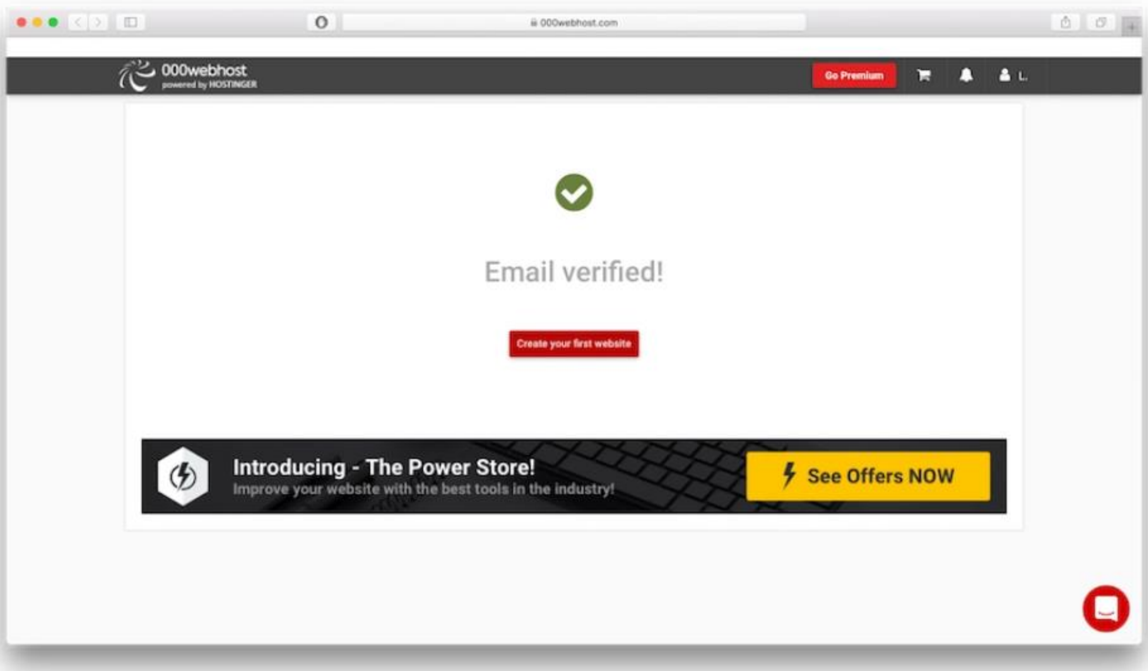
Password

••••••••••••

Repeat Password

••••••••••••

Step 3 : Verify mail id by clicking on the link send by 000webhost.com

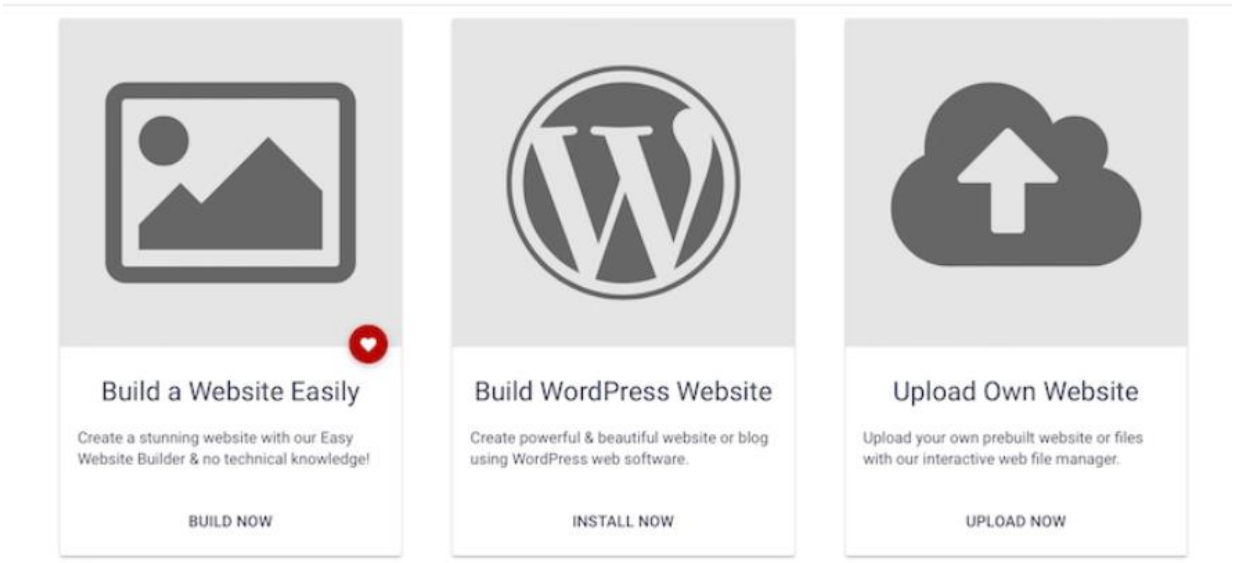


Email verified!

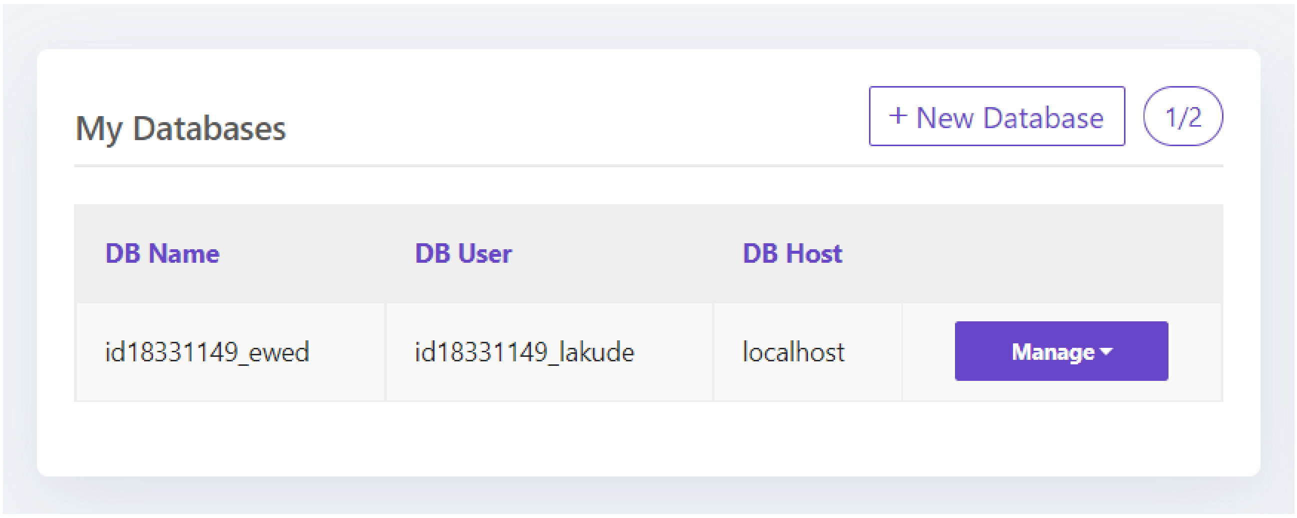
Introducing - The Power Store!
Improve your website with the best tools in the industry!

Step 4 : search for a domain name and confirm (Provided by 000webhost)

Step 5: Upload website files


Build a Website Easily
Create a stunning website with our Easy Website Builder & no technical knowledge!
BUILD NOW
Build WordPress Website
Create powerful & beautiful website or blog using WordPress web software.
INSTALL NOW
Upload Own Website
Upload your own prebuilt website or files with our interactive web file manager.
UPLOAD NOW

Step 6: Upload database

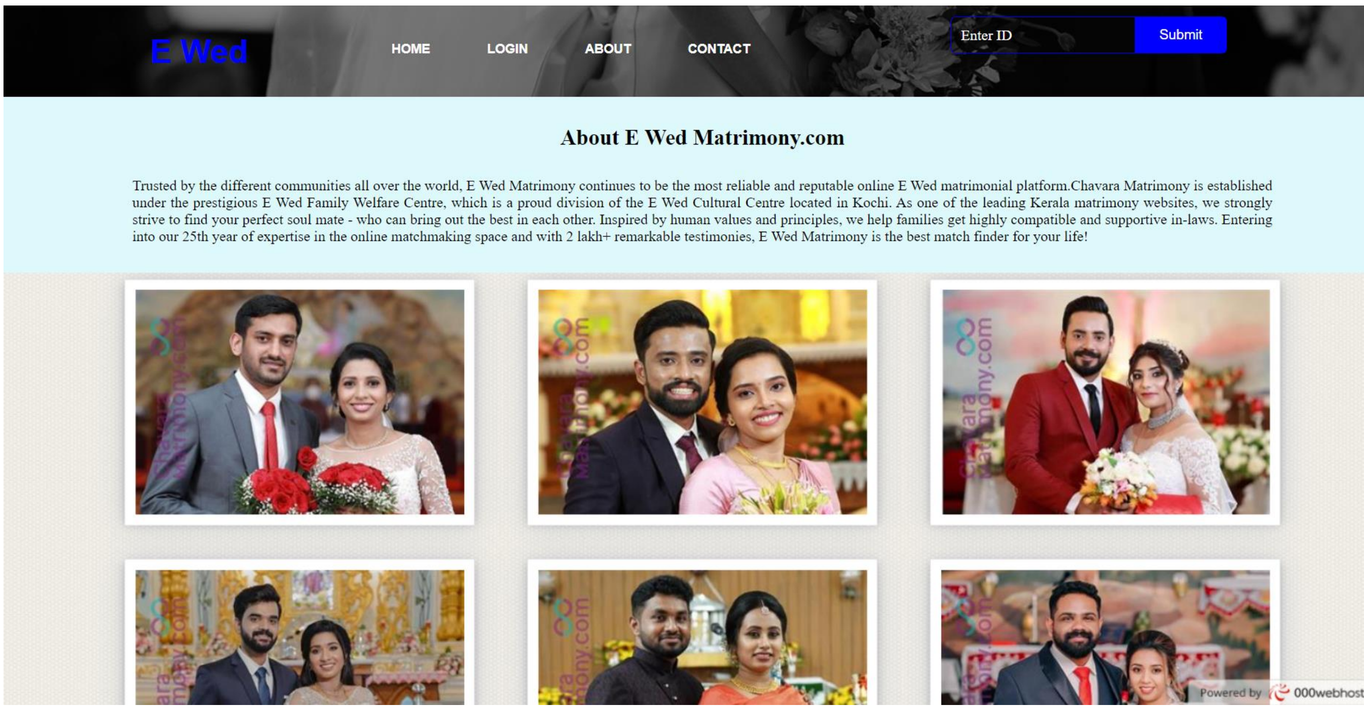


My Databases

DB Name	DB User	DB Host	
id18331149_ewed	id18331149_lakude	localhost	Manage

Step 7: update connection file with details provided by the system

Step 8 : Hosted website



About E Wed Matrimony.com

Trusted by the different communities all over the world, E Wed Matrimony continues to be the most reliable and reputable online E Wed matrimonial platform.Chavara Matrimony is established under the prestigious E Wed Family Welfare Centre, which is a proud division of the E Wed Cultural Centre located in Kochi. As one of the leading Kerala matrimony websites, we strongly strive to find your perfect soul mate - who can bring out the best in each other. Inspired by human values and principles, we help families get highly compatible and supportive in-laws. Entering into our 25th year of expertise in the online matchmaking space and with 2 lakh+ remarkable testimonies, E Wed Matrimony is the best match finder for your life!

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to find partners online and view all information.

7.2.FUTURE SCOPE

- The proposed system is designed in such a way that the payment should be done in online mode as well as cash on delivery.
- Users can able to do advanced search options
- Vehicle users can able to add complaints and feedbacks etc.
- Data security can be enhanced.
- Reduce the fuel consumption.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

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

WEBSITES:

- 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

Main.php

```
<?php
session_start();
include("connection.php");
include("function.php");


$id = $_SESSION['id'];
if (!isset($_SESSION['id'])) {
  header("Location:login.php");
}


$query_account_validity = "SELECT * FROM tbl_account_validity where cid = $id";
$result_account_validity = mysqli_query($con, $query_account_validity);
$data_result_account_validity = mysqli_fetch_assoc($result_account_validity);

$month = $data_result_account_validity["month"];
$url = $data_result_account_validity["timestamp_account"];
$today = date("Y-m-d");
"today" . $today;
$dt_today = new DateTime("$today");
$dt_today->format("Y M D");
$fullArray = explode(' ', $url);
$dt = strval($fullArray[0]);
$fullArray2 = explode('-', $dt);
$dt_yy = strval($fullArray2[0]);
$dt_mm = strval($fullArray2[1]);
$dt_dd = strval($fullArray2[2]);
$check_datess = "$dt_yy-$dt_mm-$dt_dd";
$dt = new DateTime("$check_datess");
$dt->format("Y M D");
$dt->modify("+5 month");
$dt->format("Y M D");
$s = $dt->format("Y M d");
$dt->format("Y M D");


$k = 0;
if ($dt_today <= $dt) {
} else {
  $k = 1;
}
```

```
if ($_SERVER['REQUEST_METHOD'] == "POST") {
  $value = $_POST['value'];
  header("Location: search.php?id=" . $value);
}

?>
<html>

<head>
  <title>E Wed</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</head>

<body style="background-color:#e8eff4">
  <nav class="navbar navbar-light bg-dark fixed-top">
    <div class="container-fluid">

      <a class="navbar-brand" href="#" style="color:blue;margin-left:20px">
        <h2 style=" margin-left: 20px;">E Wed</h2>
      </a>



      <form name="profile" method="POST">
        <div class="row" style="margin-top:10px;">
          <div class="col">
            <input class="form-control me-2" type="search" id="value" name="value"
placeholder="Search By ID" style="width:200%" aria-label="Search" required />
          </div>
          <div class="col">
            <button class="btn btn-outline-success" type="submit" style="margin-left:
100%;">Search</button>
          </div>
        </div>
      </form>

      <?php
      $fetchimage = "SELECT * FROM photos where cid = $id";
      $fetchimageresult = mysqli_query($con, $fetchimage);

      while ($fetchimageresultdata = mysqli_fetch_assoc($fetchimageresult)) {
        $user_name = "SELECT fname,lname FROM login where id = $id";
        $user_name_result = mysqli_query($con, $user_name);
        while ($user_name_result_data = mysqli_fetch_assoc($user_name_result)) {
      ?>
          <h7 style="color:white">
            <?php

            echo $user_name_result_data['fname'];
            ?>
            <br>
          <?php
          echo $user_name_result_data['lname'];
        }
          ?>
          </h7>
          <?php

          $name = $fetchimageresultdata["profiles"];
          ?>
          <img src="image_upload/images/<?php echo $name; ?>" style="width: 40px;height:
40px;border-radius:25px;border:2px solid white;margin: -25% ;" alt="Error" />

        <?php } ?>




        <button class="navbar-toggler" type="button" data-bs-toggle="offcanvas" data-bs-
target="#offcanvasNavbar" aria-controls="offcanvasNavbar" style="background:grey">
          <span class="navbar-toggler-icon"></span>
        </button>

        <div style="color:blue;background:black;border:white;" class="offcanvas offcanvas-end"
tabindex="-1" id="offcanvasNavbar" aria-labelledby="offcanvasNavbarLabel">
          <div class="offcanvas-header">
            <h4 class="offcanvas-title" id="offcanvasNavbarLabel" Style="color: blue;">E
Wed</h4>
            <button style="border:1px solid white;" type="button" class="btn-close text-reset"
data-bs-dismiss="offcanvas" aria-label="Close">
              <h2 style="margin-top:-15px;color:blue">
                <b>x</b>
              </h2>
            </button>
          </div>

          <div class="offcanvas-body">
            <?php
            include("side_menu.php");
            ?>
            <form class="d-flex">
          </form>
        </div>
      </div>
    </div>
  </nav><br><br><br><br>

  <?php
  if ($k == 1) {
  ?>
    <script>
      alert("please Pay to Use !!!");
    </script>

    <?php

    ?>
    </center>
    <div class="container">
      <br><br><br>
      <div class="alert alert-primary container" style="text-align:center">
        <?php echo "Please pay to use !!!"; ?>
      </div>
    </div>
    </center>
  <?php



  } else {
  ?>
    <br>
    </center>
    <div class="row container rounded " style="margin-left: 100px">
      <div class="col  rounded border border-info" style="padding:
15px;background:white;color:black">
        <center>
          <b>
            Account Expires : </b>
          <h4 style="color:red"><?php echo $s ?></h4>

        </center>
      </div>



      <?php
      $preferancetable = "SELECT * from partnerdetails where cid = $id";
      $preferanceresult = mysqli_query($con, $preferancetable);
      while ($preferance_row = mysqli_fetch_assoc($preferanceresult)) {
```

```
$pre_agemin = $preferance_row['agemin'];
$pre_agemax = $preferance_row['agemax'];
$pre_heightmin = $preferance_row['heightmin'];
$pre_heightmax = $preferance_row['heightmax'];

$pre_maritalstatus = $preferance_row['maritalstatus'];

if ($pre_maritalstatus == 'ANY') {


  $pre_maritalstatus = '%';
}

$pre_religion = $preferance_row['religion'];

if ($pre_religion == 'ANY') {


  $pre_religion = '%';
}

$pre_caste = $preferance_row['caste'];

if ($pre_caste == 'ANY') {


  $pre_caste = '%';
}

$pre_mothertounge = $preferance_row['mothertounge'];
if ($pre_mothertounge == 'ANY') {

  $values_table = "SELECT l_value from language";
  $values_result = mysqli_query($con, $values_table);
  while ($value_row = mysqli_fetch_array($values_result)) {
    $pre_mothertounge = $value_row["l_value"];

    $pre_mothertounge = '%';
  }
}


$pre_educationqualification = $preferance_row['educationqualification'];
if ($pre_educationqualification == 'ANY') {

  $values_table = "SELECT e_value from education";
  $values_result = mysqli_query($con, $values_table);
  while ($value_row = mysqli_fetch_array($values_result)) {
    $pre_educationqualification = $value_row["e_value"];

    $pre_educationqualification = '%';
  }
}

$pre_occupation = $preferance_row['occupation'];

if ($pre_occupation == 'ANY') {

  $values_table = "SELECT o_value from occupation";
  $values_result = mysqli_query($con, $values_table);
  while ($value_row = mysqli_fetch_array($values_result)) {
    $pre_occupation = $value_row["o_value"];

    $pre_occupation = '%';
  }
}

$pre_country = $preferance_row['country'];

if ($pre_country == 'ANY') {

  $pre_country = '%';
}

$pre_state = $preferance_row['state'];

if ($pre_state == 'ANY') {

  $values_table = "SELECT s_value from state";
  $values_result = mysqli_query($con, $values_table);
  while ($value_row = mysqli_fetch_array($values_result)) {
    $pre_state = $value_row["s_value"];

    $pre_state = '%';
  }
}

$pre_drink = $preferance_row['drink'];

if ($pre_drink == 'ANY') {
  $pre_drink = "'YES','NO'";

  $pre_drink = '%';
}

$pre_smoke = $preferance_row['smoke'];

if ($pre_smoke == 'ANY') {
  $pre_smoke = "'YES','NO'";

  $pre_smoke = '%';
}
```

```
    $pre_color = $preferance_row['color'];

    if ($pre_color == 'ANY') {
      $pre_color = "'Dark','Normal','White'";

      $pre_color = '%';
    }

    $pre_financialstatus = $preferance_row['financialstatus'];

    if ($pre_financialstatus == 'ANY') {


      $pre_financialstatus = '%';
    }

    $pre_bodytype = $preferance_row['bodytype'];

    if ($pre_bodytype == 'ANY') {


      $pre_bodytype = '%';
    }
  }


  $p_s = 0;
  $fetch_proof_status = "SELECT preferance_status FROM status where cid = $id";
  $fetch_proof_status_result = mysqli_query($con, $fetch_proof_status);
  $data_fetch_proof_status_result = mysqli_fetch_assoc($fetch_proof_status_result);
  if ($data_fetch_proof_status_result["preferance_status"] == 1) {
    $p_s = 1;
  }


  if ($p_s == 0) {



    $table = "SELECT * from profiledata where cid in (SELECT cid from status where
profilestatus = 1) and sex != (SELECT sex from profiledata where cid = $id)
  and age between $pre_agemin and $pre_agemax
  and heigth between $pre_heightmin and $pre_heightmax
  and maritalstatus like '$pre_maritalstatus'
  and religion like '$pre_religion'
  and caste like '$pre_caste'
  and mothertounge like '$pre_mothertounge'
  and educationqualification like '$pre_educationqualification'
  and occupation like '$pre_occupation'
  and country like '$pre_country'
  and states like '$pre_state'
  and drink like '$pre_drink'
  and smoke like '$pre_smoke'
  and color like '$pre_color'
  and financialstatus like '$pre_financialstatus'
  and bodytype like '$pre_bodytype' ";
      } else {
        $table = "SELECT * from profiledata where cid in (SELECT cid from status where
profilestatus = 1) and sex != (SELECT sex from profiledata where cid = $id)
  and age between $pre_agemin and $pre_agemax
  and maritalstatus like '$pre_maritalstatus'
  and religion like '$pre_religion'
  and country like '$pre_country' ";
      }




      $result = mysqli_query($con, $table);
      if (mysqli_num_rows($result) > 0) {



      ?>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;
        <div class="col  rounded border border-info" style="padding:
15px;background:white;color:black;">
          <center>
            <b>
              Matching Profile :
            </b>
            <h4 style="color:blue">
              <?php echo mysqli_num_rows($result); ?>
            </h4>
          </center>
        </div>
    </div>
    </center>
    <br>
    <?php


        while ($row = mysqli_fetch_assoc($result)) {
    ?>
      <div class=" container rounded" style="padding: 15px;box-shadow: 1px 1px 5px 5px
lightblue;background:white;color:black">
        <div class="row">
```

```
            <div class="col">
              <?php
              $value = $row["cid"];
              $fetchimage = "SELECT * FROM photos where cid = $value";
              $fetchimageresult = mysqli_query($con, $fetchimage);
              while ($fetchimageresultdata = mysqli_fetch_assoc($fetchimageresult)) {
                $name = $fetchimageresultdata["profiles"];

              ?>
                <a href="viewprofile.php?id=<?= $row["cid"] ?>">
                  <img class=" rounded" src="image_upload/images/<?php echo $name; ?>"
    style="width: 400px;height: 400px;margin-left: 30px;" alt="Error" />
                </a>
              <?php } ?>
            </div>




            <div class="col">

              <tr><br>
                <td><span class="#" style="color:blue">ID :</span>
                  <?php echo $row["cid"]; ?>
                </td>
                <br><br>
                <?php


                ?>
                <td><span class="#" style="color:blue">Name :</span>
                  <?php echo $row["fname"] . " " . $row["lname"]; ?>
                </td>
                <br><br>
                <?php

                ?>
                <td><span class="#" style="color:blue">Education :</span>
                  <?php echo $row["educationqualification"]; ?>
                </td>
                <br><br>
                <?php

                ?>
                <td><span class="#" style="color:blue">Date of Birth :</span>
                  <?php echo $row["dob"]; ?>
                </td>
                <br><br>
                <?php
```

```
        ?>
        <td><span class="#" style="color:blue">Finance Status:</span>
          <?php echo $row["financialstatus"]; ?>
        </td>
        <br><br>
        <?php

        ?>
        <td><span class="#" style="color:blue">Age :</span>
          <?php echo $row["age"]; ?>
        </td>
        <br><br>
        <?php

        ?>
        <td><span class="#" style="color:blue">Height :</span>
          <?php echo $row["heigth"]; ?>
        </td>
        <br><br>
        <?php

        ?>
        <td><span class="#" style="color:blue">Weight :</span>
          <?php echo $row["weights"]; ?>
        </td>
        <br>
       </tr>
       <?php

       ?>
      </div>
     </div>
    </div>
    <br><?php

     }
    } else {
       ?>
   <center class="alert alert-info container">
    <h3>
     No Matching Profile !!!
    </h3>
   </center>
<?php
    }
   }
?>
```

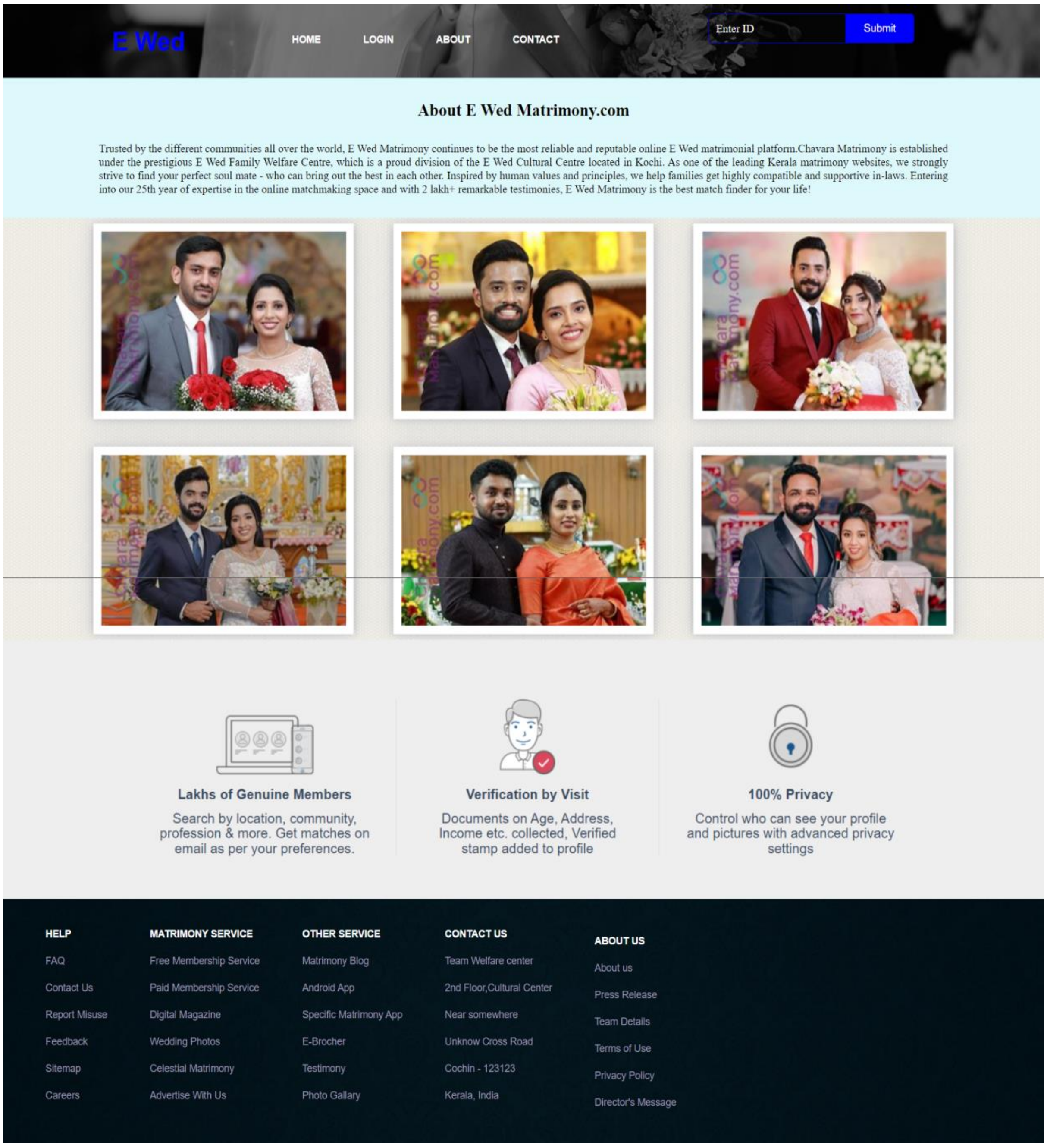
```
</div>
</div>
</div>
<br>


</body>

</html>
```

9.2 Screen Shots

Home page



About E Wed Matrimony.com

Trusted by the different communities all over the world, E Wed Matrimony continues to be the most reliable and reputable online E Wed matrimonial platform.Chavara Matrimony is established under the prestigious E Wed Family Welfare Centre, which is a proud division of the E Wed Cultural Centre located in Kochi. As one of the leading Kerala matrimony websites, we strongly strive to find your perfect soul mate - who can bring out the best in each other. Inspired by human values and principles, we help families get highly compatible and supportive in-laws. Entering into our 25th year of expertise in the online matchmaking space and with 2 lakh+ remarkable testimonies, E Wed Matrimony is the best match finder for your life!

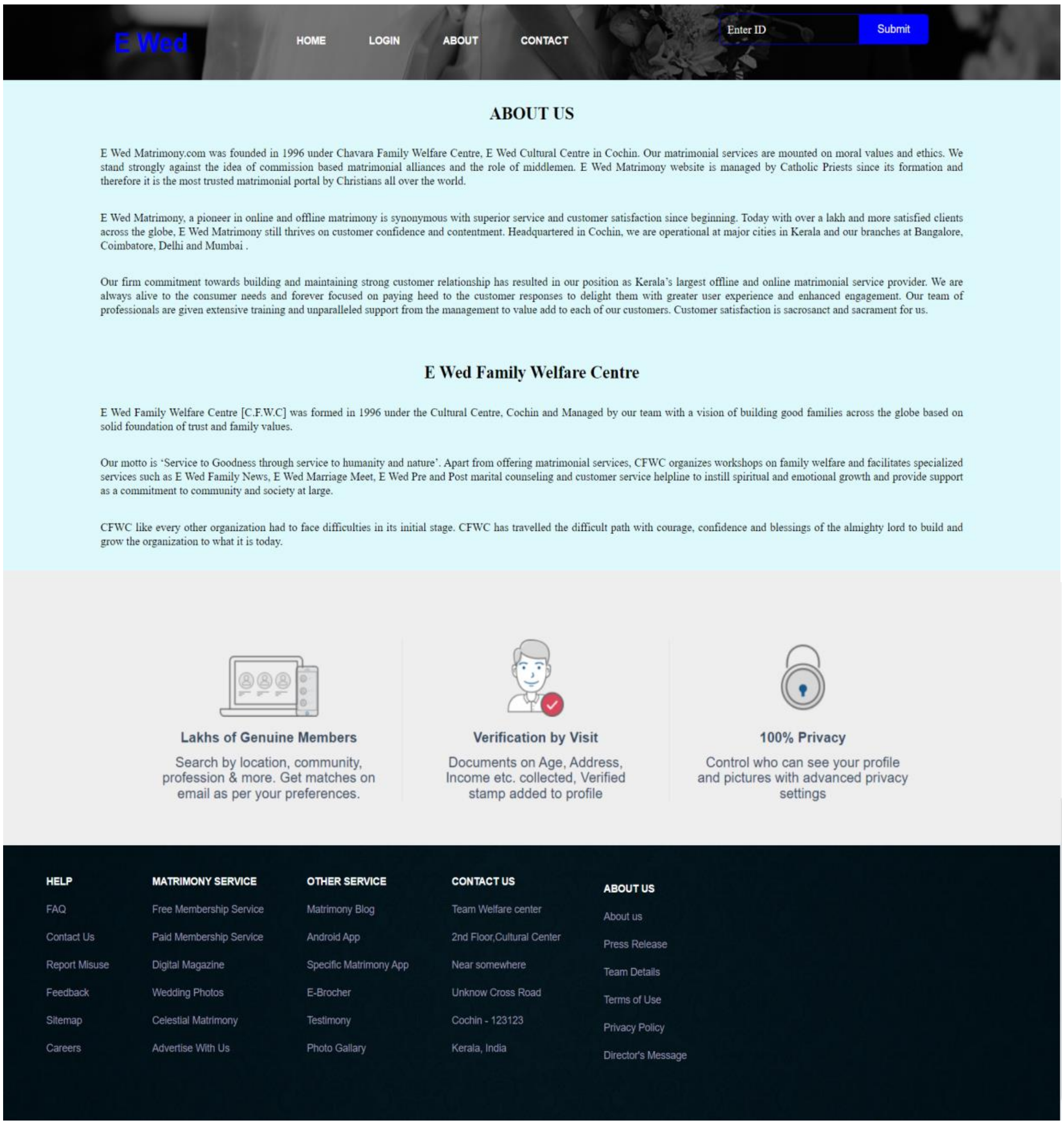
Lakhs of Genuine Members
Search by location, community, profession & more. Get matches on email as per your preferences.

Verification by Visit
Documents on Age, Address, Income etc. collected, Verified stamp added to profile

100% Privacy
Control who can see your profile and pictures with advanced privacy settings

HELP	MATRIMONY SERVICE	OTHER SERVICE	CONTACT US	ABOUT US
FAQ	Free Membership Service	Matrimony Blog	Team Welfare center	About us
Contact Us	Paid Membership Service	Android App	2nd Floor,Cultural Center	Press Release
Report Misuse	Digital Magazine	Specific Matrimony App	Near somewhere	Team Details
Feedback	Wedding Photos	E-Brocher	Unknow Cross Road	Terms of Use
Sitemap	Celestial Matrimony	Testimony	Cochin - 123123	Privacy Policy
Careers	Advertise With Us	Photo Gallary	Kerala, India	Director's Message

About Us page



ABOUT US

E Wed Matrimony.com was founded in 1996 under Chavara Family Welfare Centre, E Wed Cultural Centre in Cochin. Our matrimonial services are mounted on moral values and ethics. We stand strongly against the idea of commission based matrimonial alliances and the role of middlemen. E Wed Matrimony website is managed by Catholic Priests since its formation and therefore it is the most trusted matrimonial portal by Christians all over the world.

E Wed Matrimony, a pioneer in online and offline matrimony is synonymous with superior service and customer satisfaction since beginning. Today with over a lakh and more satisfied clients across the globe, E Wed Matrimony still thrives on customer confidence and contentment. Headquartered in Cochin, we are operational at major cities in Kerala and our branches at Bangalore, Coimbatore, Delhi and Mumbai .

Our firm commitment towards building and maintaining strong customer relationship has resulted in our position as Kerala's largest offline and online matrimonial service provider. We are always alive to the consumer needs and forever focused on paying heed to the customer responses to delight them with greater user experience and enhanced engagement. Our team of professionals are given extensive training and unparalleled support from the management to value add to each of our customers. Customer satisfaction is sacrosanct and sacrament for us.

E Wed Family Welfare Centre

E Wed Family Welfare Centre [C.F.W.C] was formed in 1996 under the Cultural Centre, Cochin and Managed by our team with a vision of building good families across the globe based on solid foundation of trust and family values.

Our motto is 'Service to Goodness through service to humanity and nature'. Apart from offering matrimonial services, CFWC organizes workshops on family welfare and facilitates specialized services such as E Wed Family News, E Wed Marriage Meet, E Wed Pre and Post marital counseling and customer service helpline to instill spiritual and emotional growth and provide support as a commitment to community and society at large.

CFWC like every other organization had to face difficulties in its initial stage. CFWC has travelled the difficult path with courage, confidence and blessings of the almighty lord to build and grow the organization to what it is today.

Lakhs of Genuine Members

Search by location, community, profession & more. Get matches on email as per your preferences.

Verification by Visit

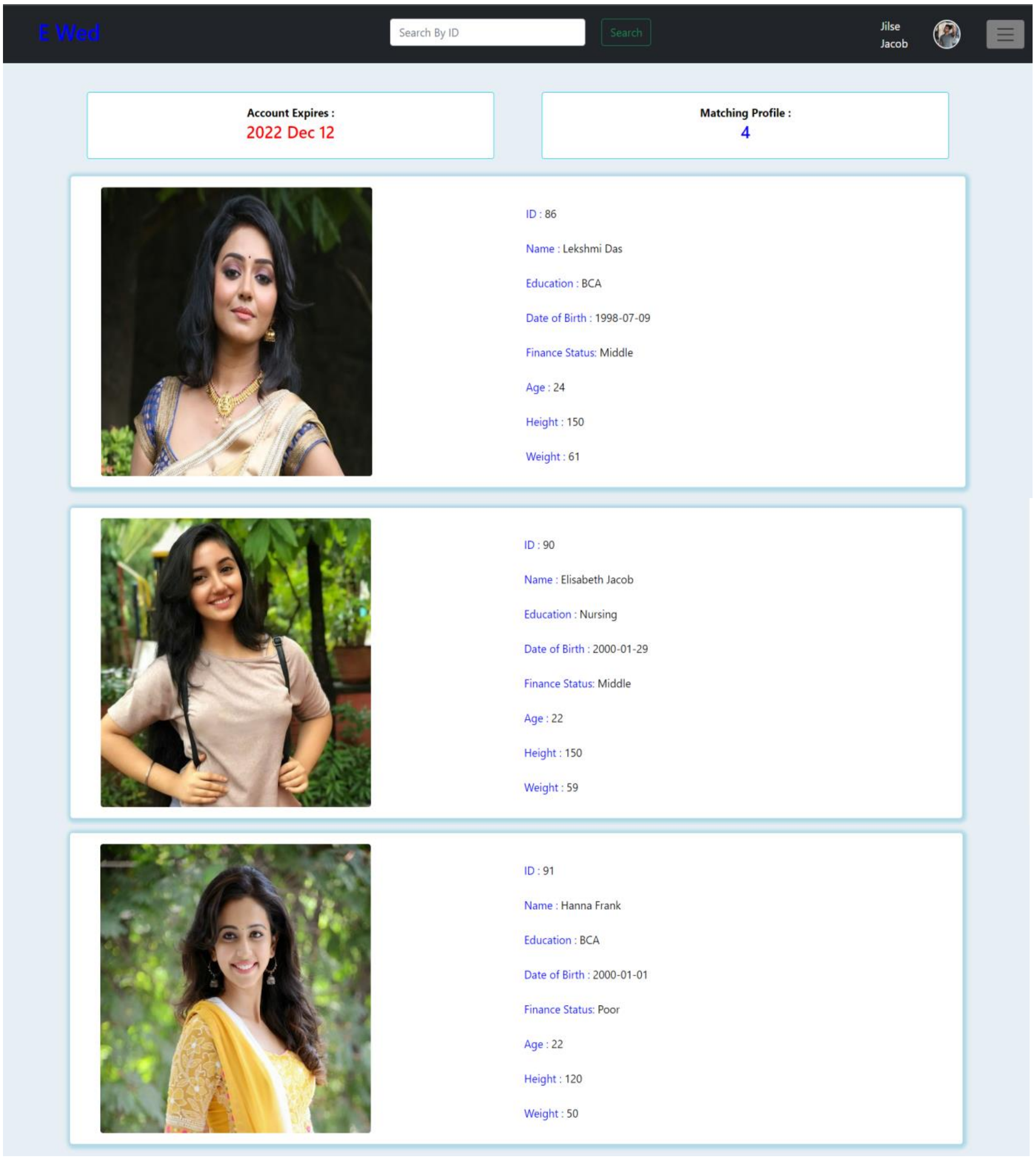
Documents on Age, Address, Income etc. collected, Verified stamp added to profile

100% Privacy

Control who can see your profile and pictures with advanced privacy settings

HELP	MATRIMONY SERVICE	OTHER SERVICE	CONTACT US	ABOUT US
FAQ	Free Membership Service	Matrimony Blog	Team Welfare center	About us
Contact Us	Paid Membership Service	Android App	2nd Floor,Cultural Center	Press Release
Report Misuse	Digital Magazine	Specific Matrimony App	Near somewhere	Team Details
Feedback	Wedding Photos	E-Brocher	Unknow Cross Road	Terms of Use
Sitemap	Celestial Matrimony	Testimony	Cochin - 123123	Privacy Policy
Careers	Advertise With Us	Photo Gallary	Kerala, India	Director's Message

User Home page



Account Expires :
2022 Dec 12

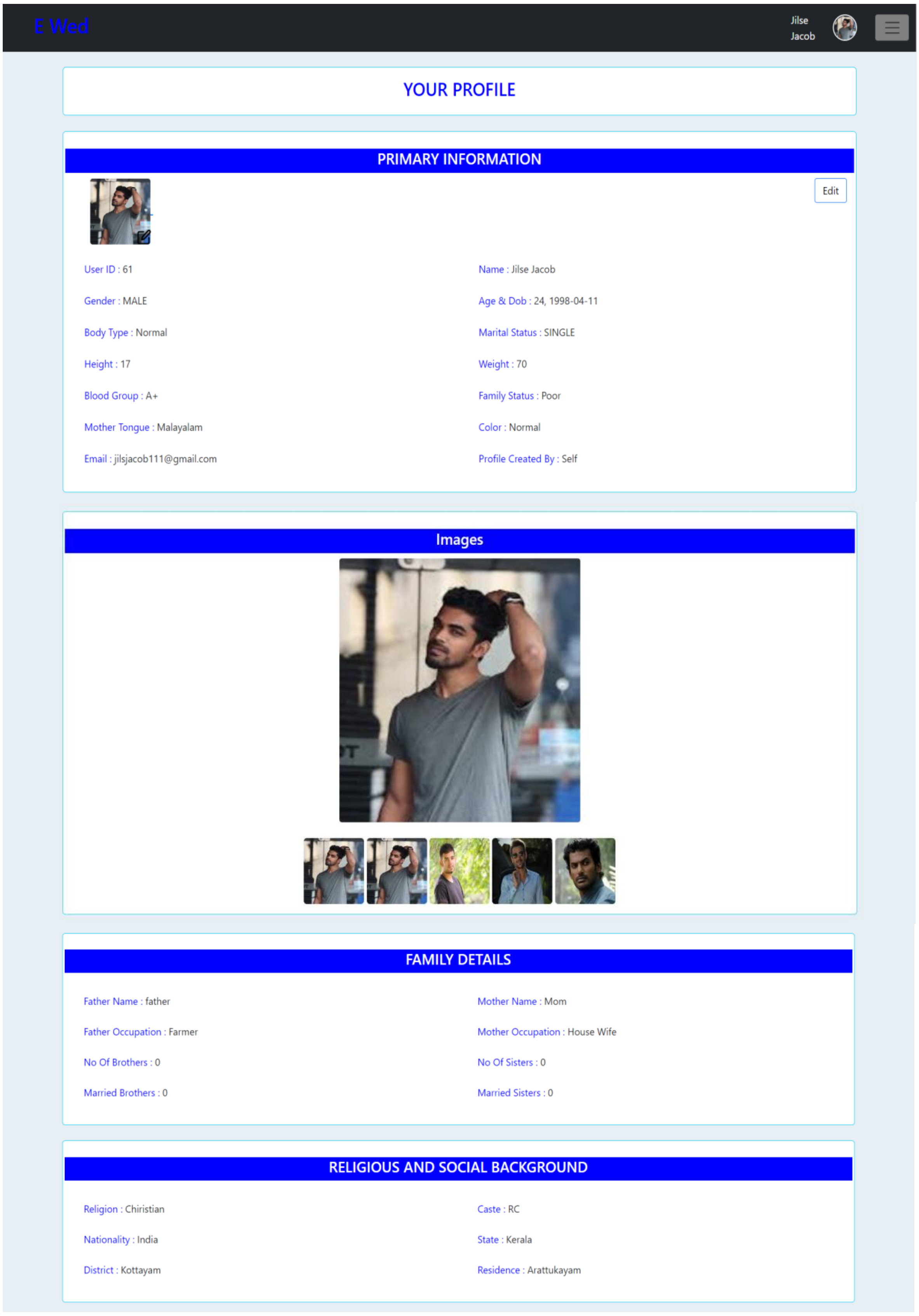
Matching Profile :
4

ID : 86
Name : Lekshmi Das
Education : BCA
Date of Birth : 1998-07-09
Finance Status: Middle
Age : 24
Height : 150
Weight : 61

ID : 90
Name : Elisabeth Jacob
Education : Nursing
Date of Birth : 2000-01-29
Finance Status: Middle
Age : 22
Height : 150
Weight : 59

ID : 91
Name : Hanna Frank
Education : BCA
Date of Birth : 2000-01-01
Finance Status: Poor
Age : 22
Height : 120
Weight : 50

User Profile page



YOUR PROFILE

PRIMARY INFORMATION

User ID : 61
Name : Jilse Jacob
Gender : MALE
Age & Dob : 24, 1998-04-11
Body Type : Normal
Marital Status : SINGLE
Height : 17
Weight : 70
Blood Group : A+
Family Status : Poor
Mother Tongue : Malayalam
Color : Normal
Email : jilsjacob111@gmail.com
Profile Created By : Self

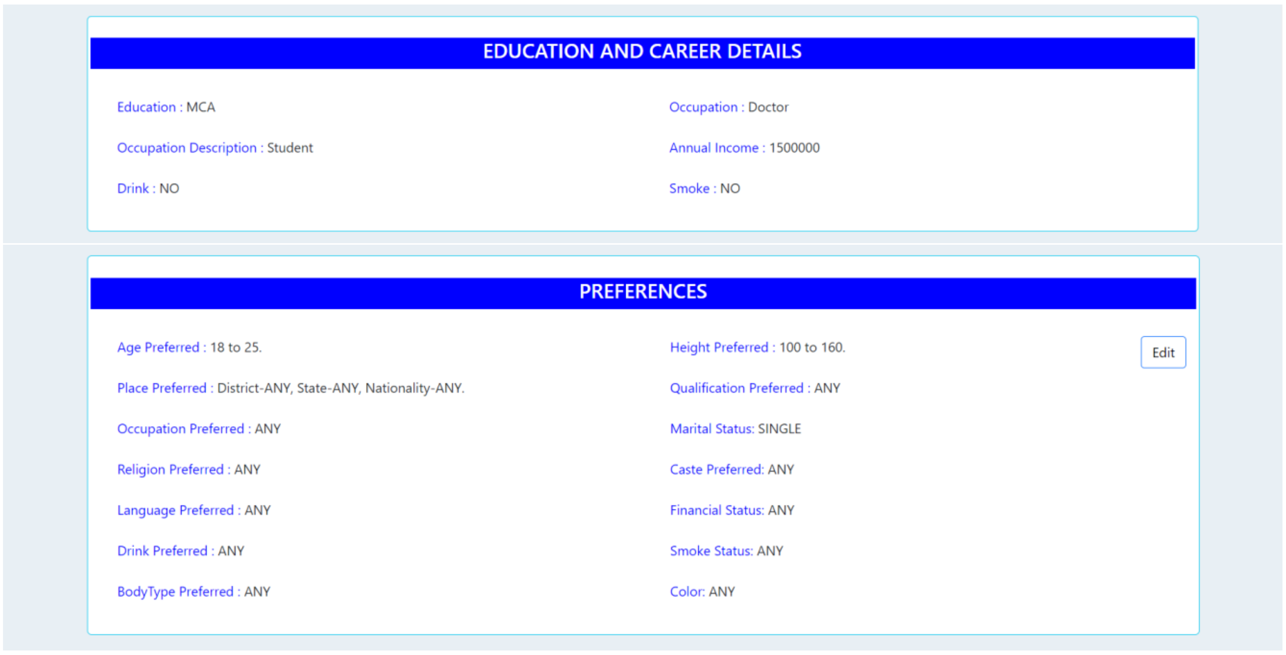
Images

FAMILY DETAILS

Father Name : father
Mother Name : Mom
Father Occupation : Farmer
Mother Occupation : House Wife
No Of Brothers : 0
No Of Sisters : 0
Married Brothers : 0
Married Sisters : 0

RELIGIOUS AND SOCIAL BACKGROUND

Religion : Chiristian
Caste : RC
Nationality : India
State : Kerala
District : Kottayam
Residence : Arattukayam

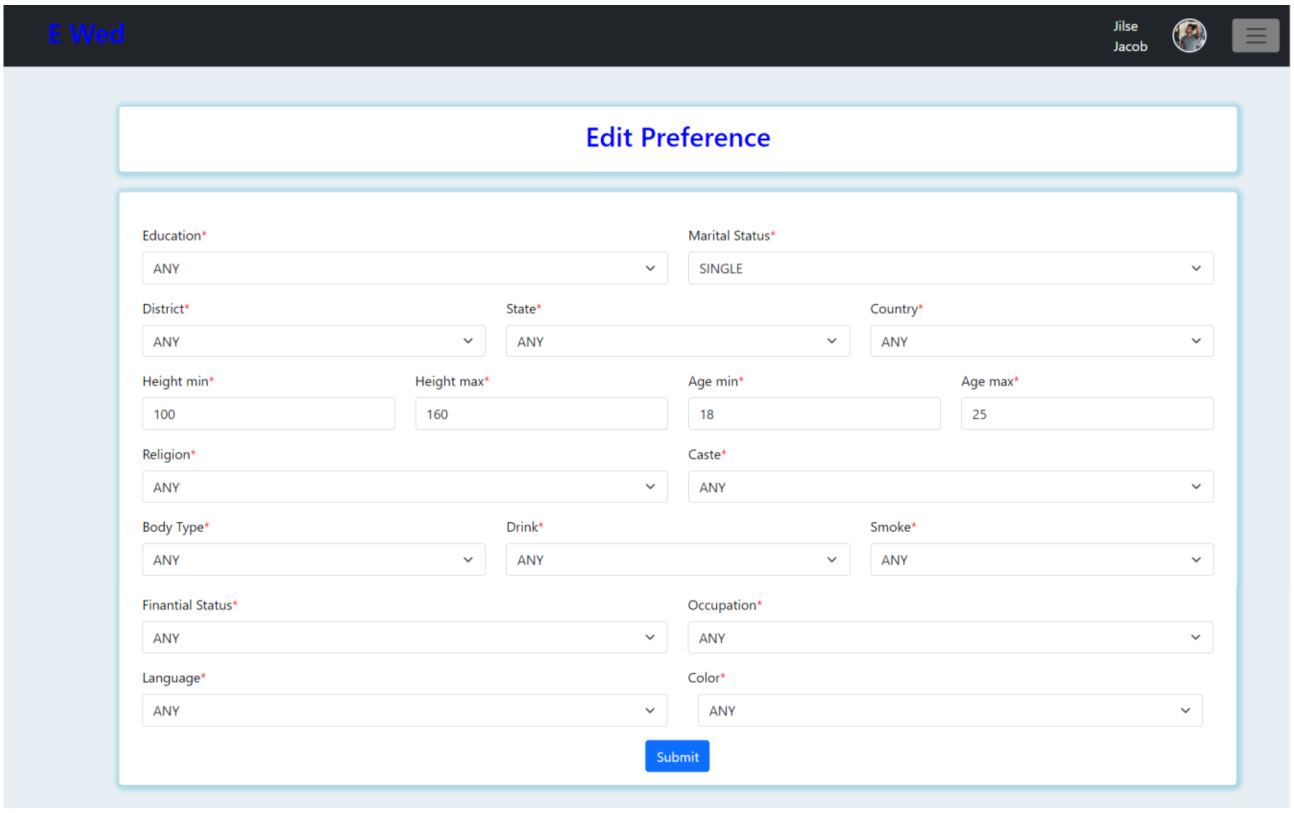

EDUCATION AND CAREER DETAILS

Education : MCA	Occupation : Doctor
Occupation Description : Student	Annual Income : 1500000
Drink : NO	Smoke : NO

PREFERENCES

Age Preferred : 18 to 25.	Height Preferred : 100 to 160.
Place Preferred : District-ANY, State-ANY, Nationality-ANY.	Qualification Preferred : ANY
Occupation Preferred : ANY	Marital Status: SINGLE
Religion Preferred : ANY	Caste Preferred: ANY
Language Preferred : ANY	Financial Status: ANY
Drink Preferred : ANY	Smoke Status: ANY
BodyType Preferred : ANY	Color: ANY

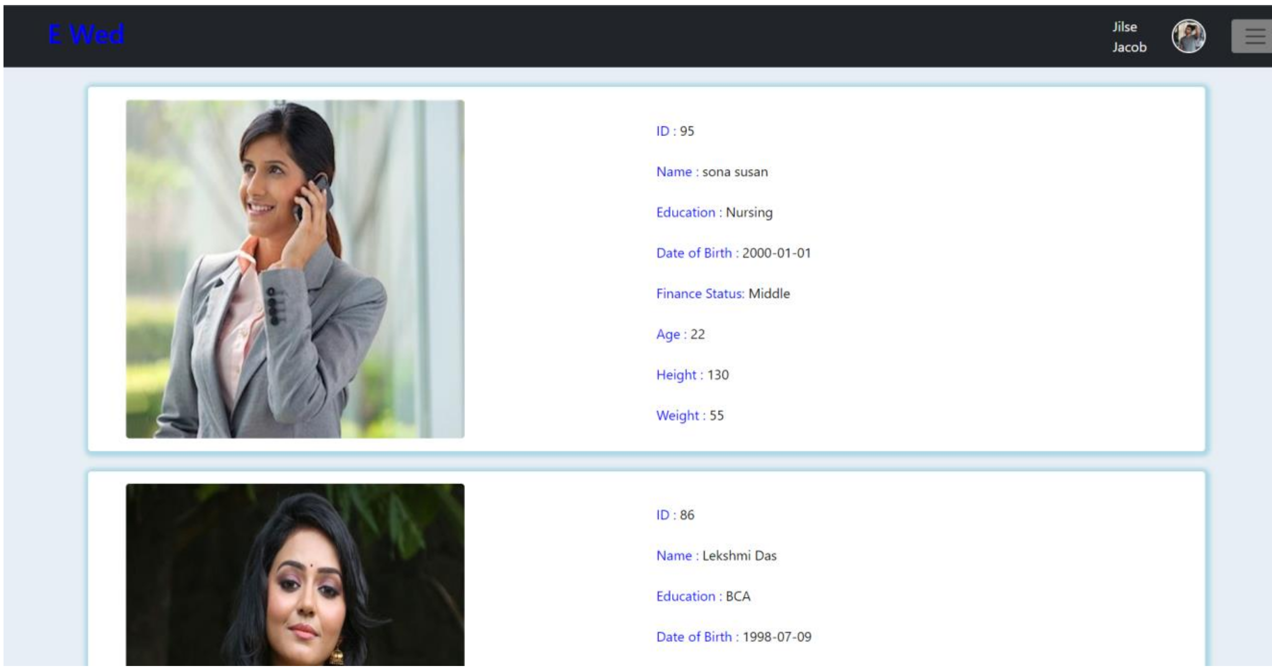
Vehicle Filter page



Edit Preference

Education*	ANY
Marital Status*	SINGLE
District*	ANY
State*	ANY
Country*	ANY
Height min*	100
Height max*	160
Age min*	18
Age max*	25
Religion*	ANY
Caste*	ANY
Body Type*	ANY
Drink*	ANY
Smoke*	ANY
Finantial Status*	ANY
Occupation*	ANY
Language*	ANY
Color*	ANY

Matches Page



ID : 95

Name : sona susan

Education : Nursing

Date of Birth : 2000-01-01

Finance Status: Middle

Age : 22

Height : 130

Weight : 55

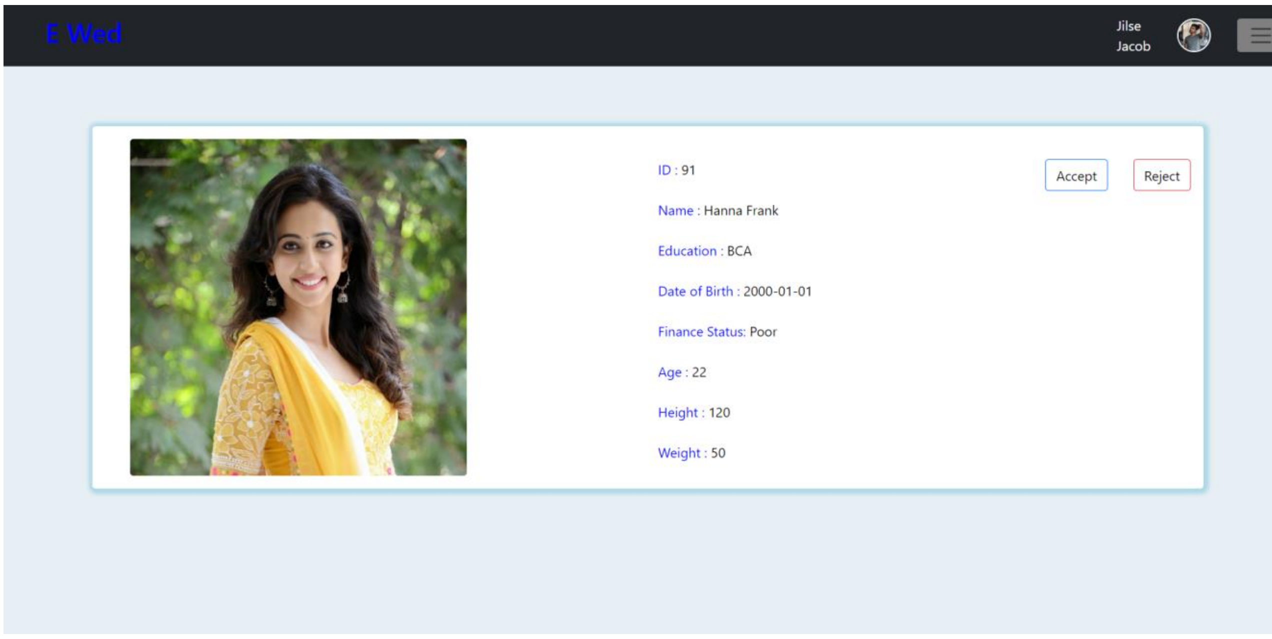
ID : 86

Name : Lekshmi Das

Education : BCA

Date of Birth : 1998-07-09

Requests page



ID : 91

Name : Hanna Frank

Education : BCA

Date of Birth : 2000-01-01

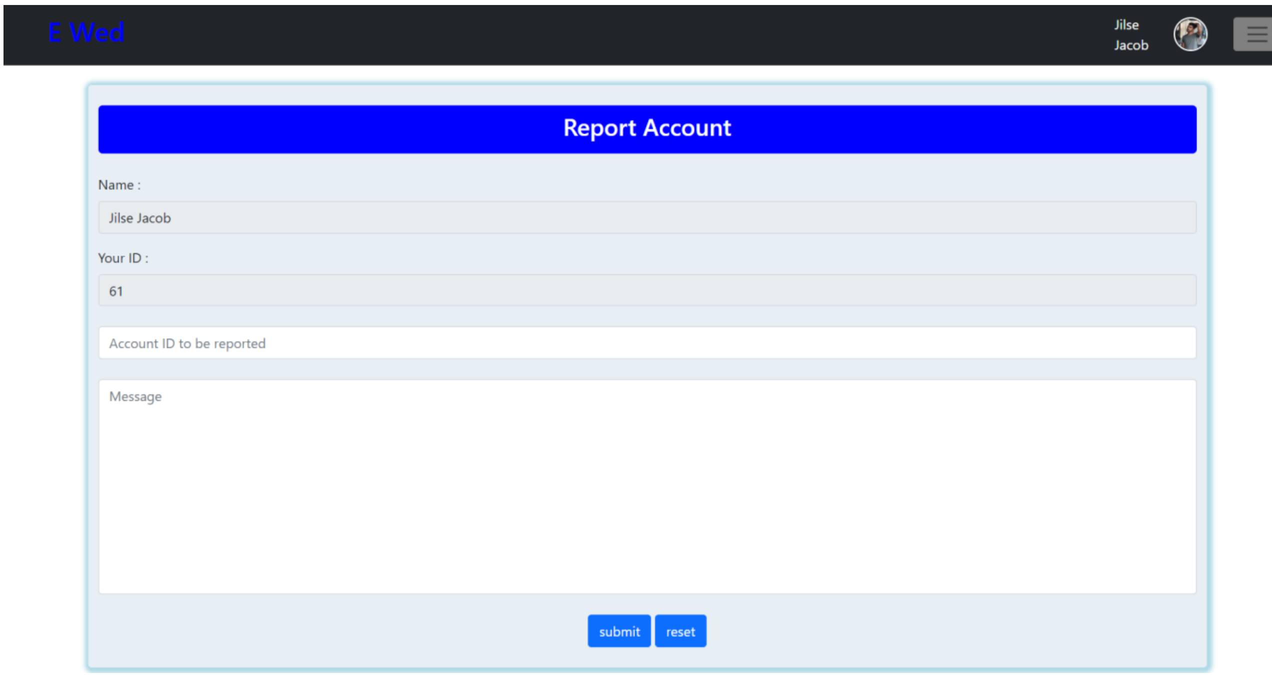
Finance Status: Poor

Age : 22

Height : 120

Weight : 50

Report Account Page



Report Account

Name :

Jilse Jacob

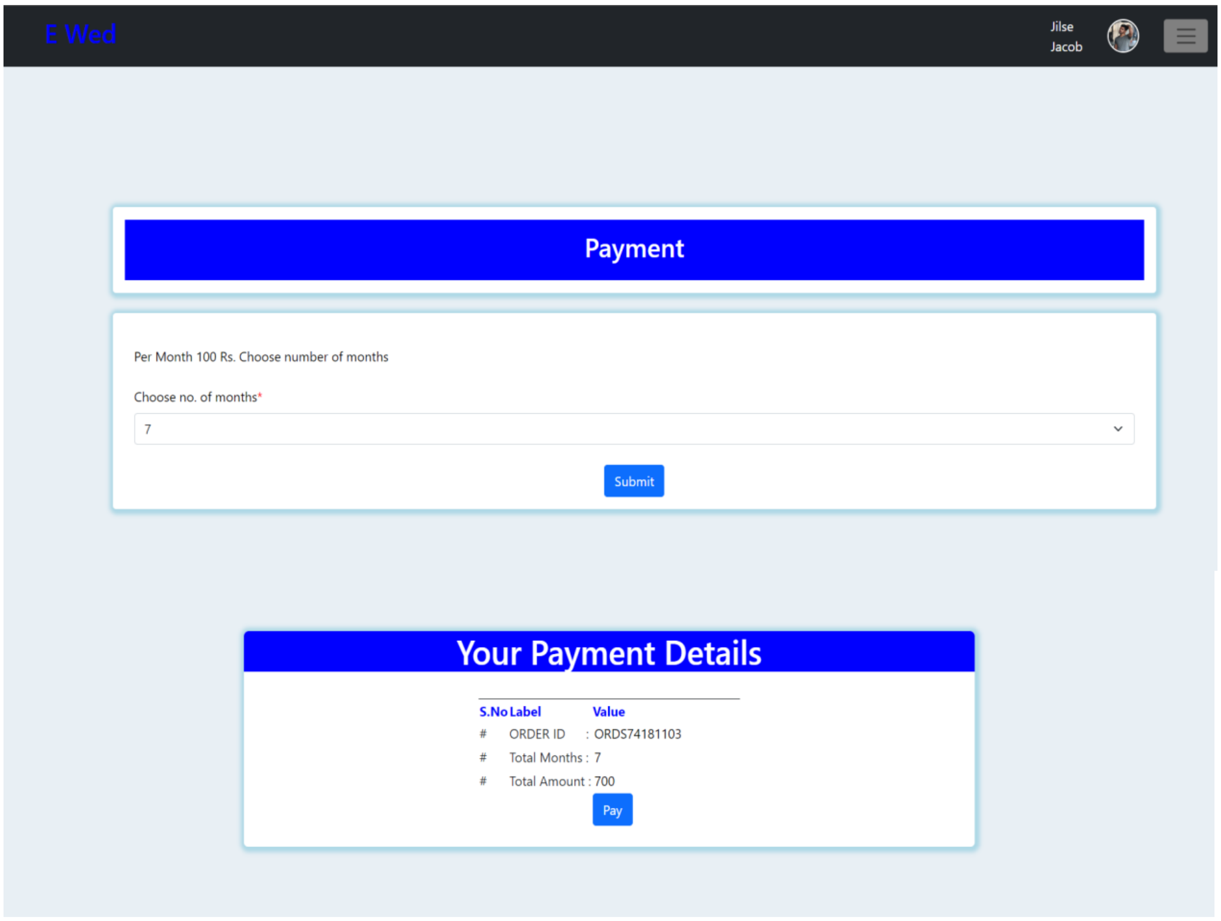
Your ID :

61

Account ID to be reported

Message

Payment pages



Payment

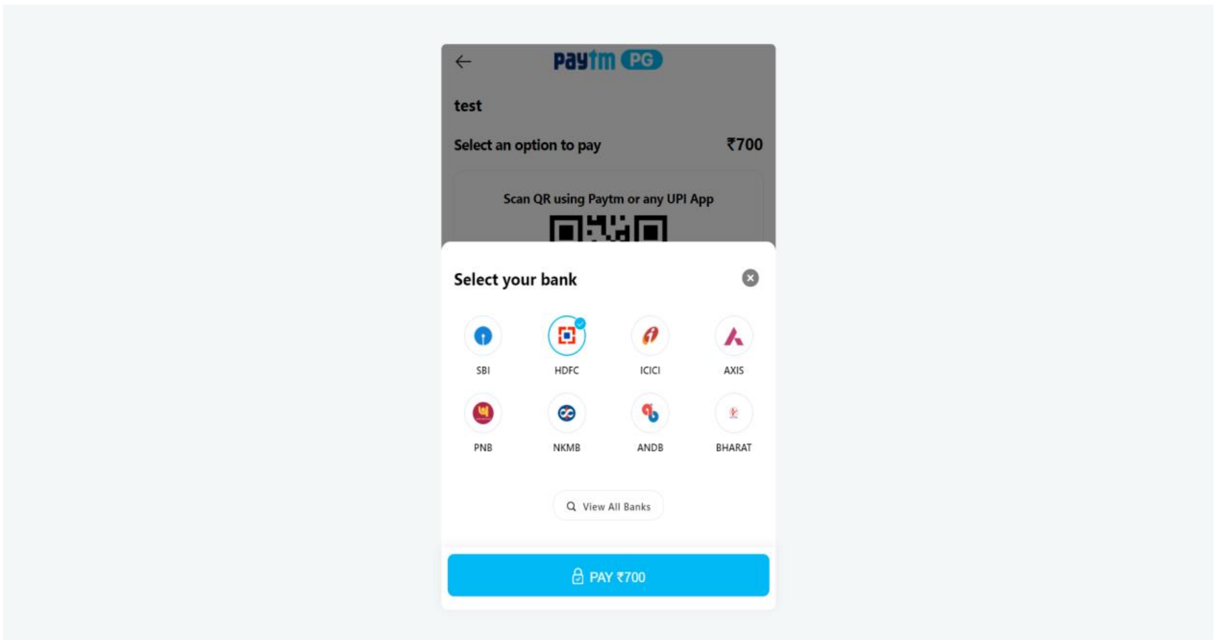
Per Month 100 Rs. Choose number of months

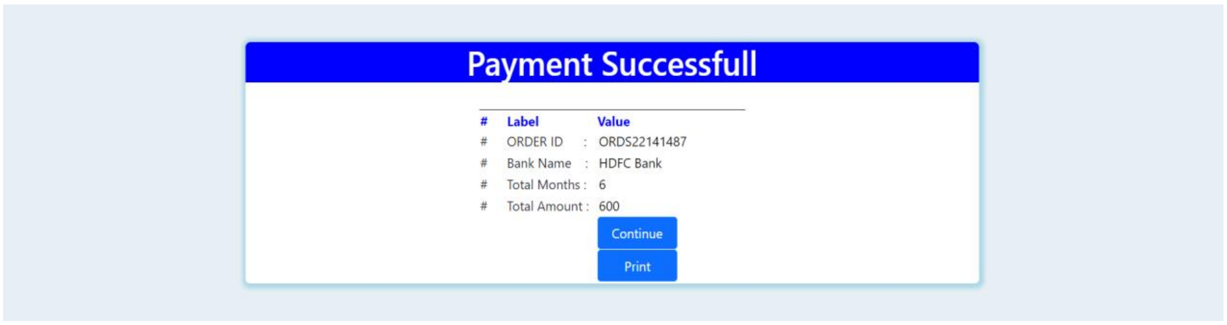
Choose no. of months*

7

Your Payment Details

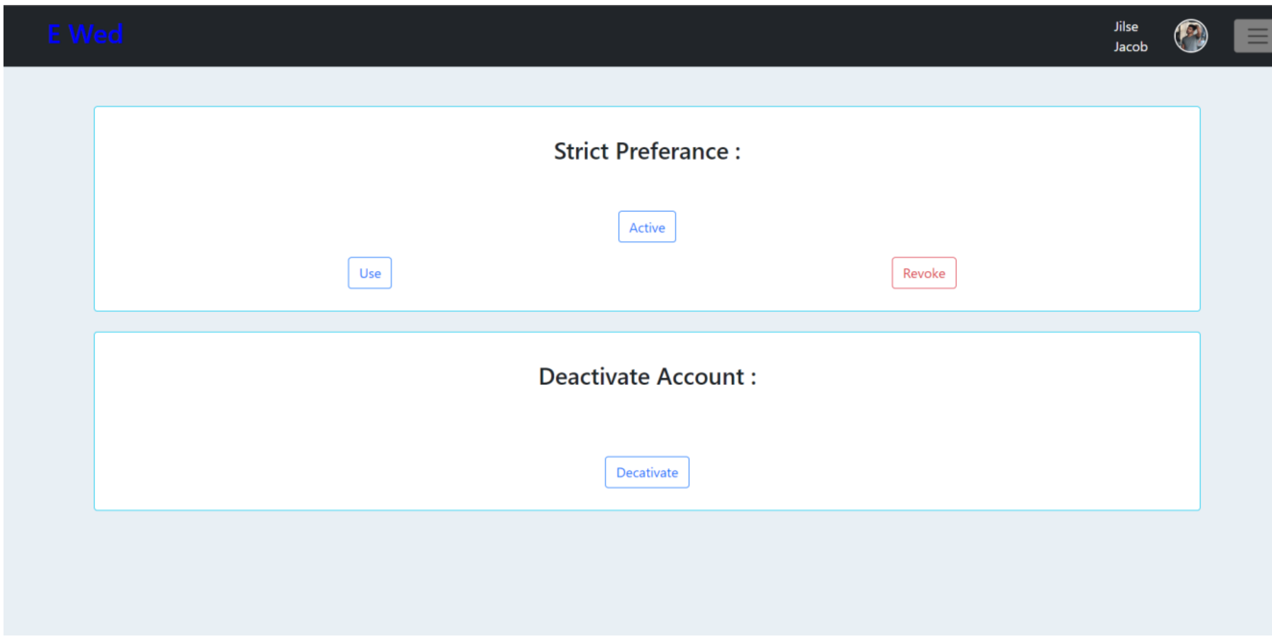
S.No	Label	Value
#	ORDER ID	: ORDS74181103
#	Total Months :	7
#	Total Amount :	700




Payment Successfull

#	Label		Value
#	ORDER ID	:	ORDS22141487
#	Bank Name	:	HDFC Bank
#	Total Months :		6
#	Total Amount :		600

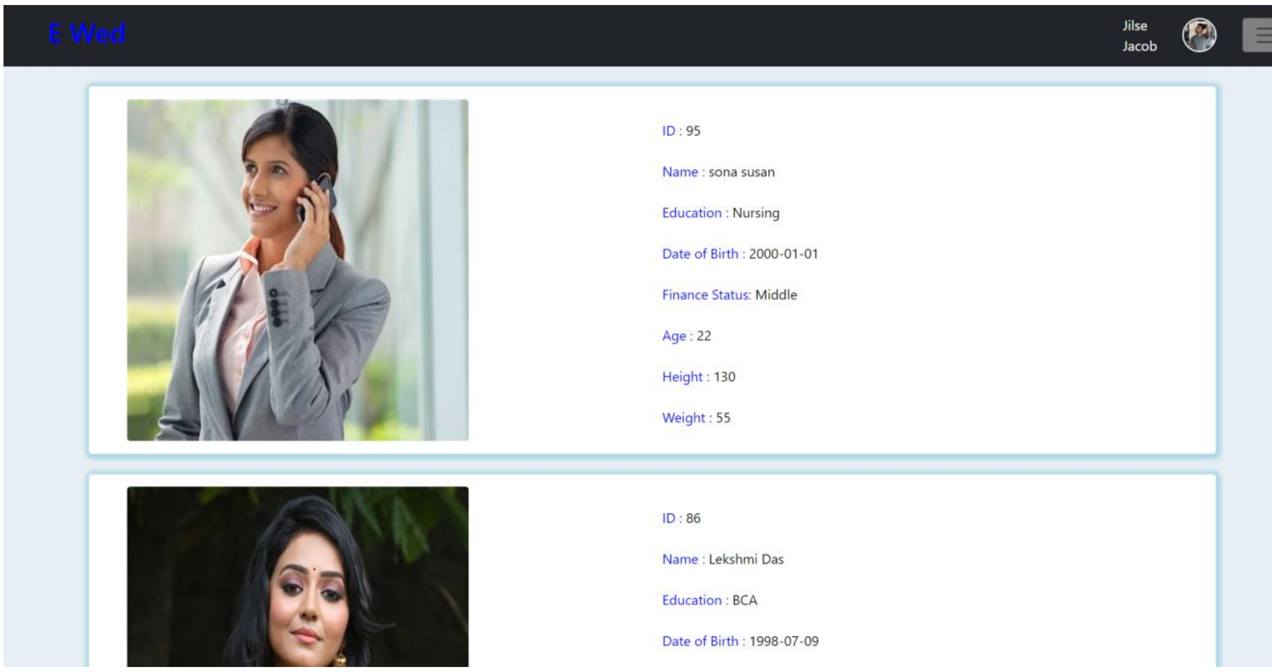
Settings page



Strict Preferance :

Deactivate Account :

Profile Visiters Page



ID : 95

Name : sona susan

Education : Nursing

Date of Birth : 2000-01-01

Finance Status: Middle

Age : 22

Height : 130

Weight : 55

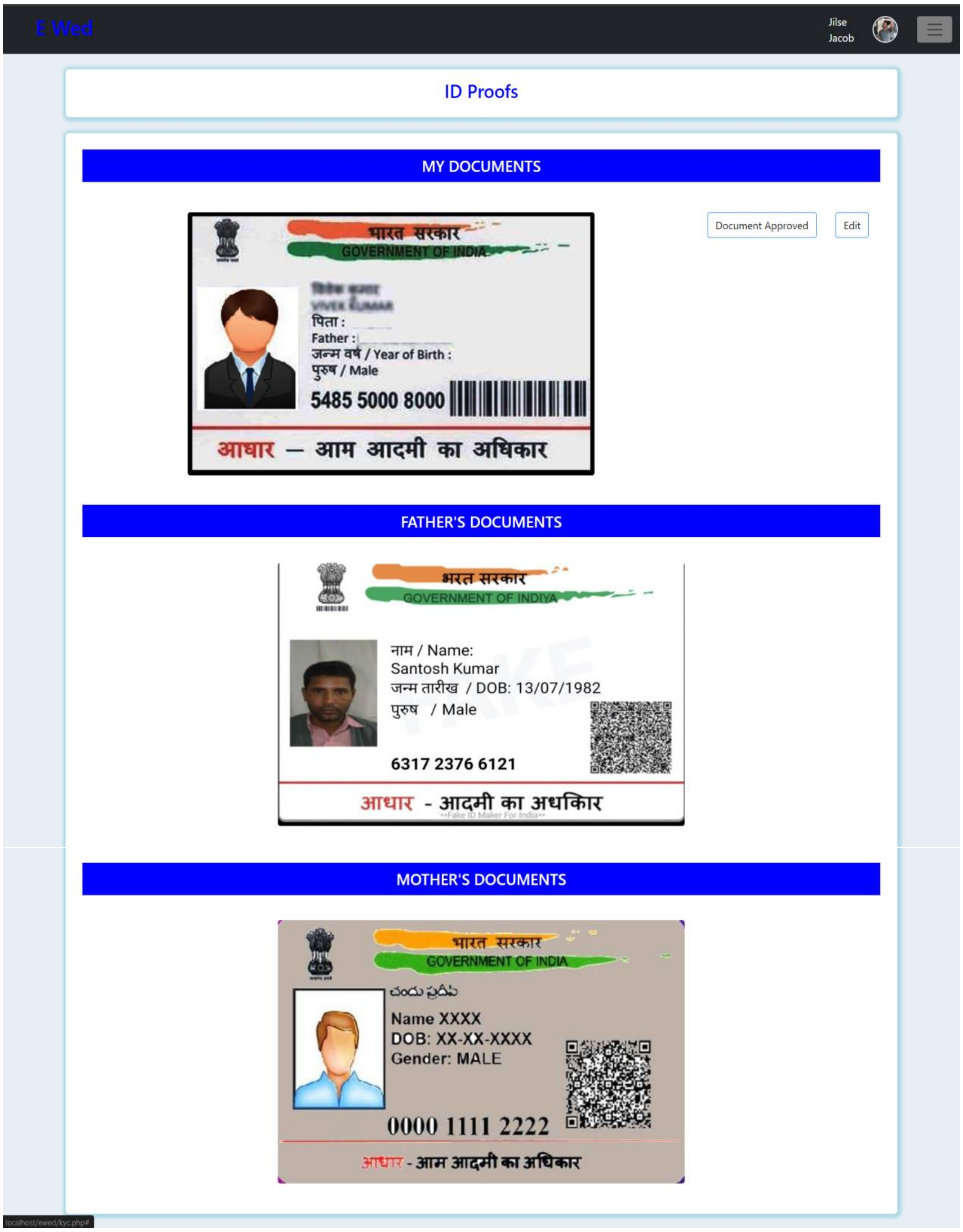
ID : 86

Name : Lekshmi Das

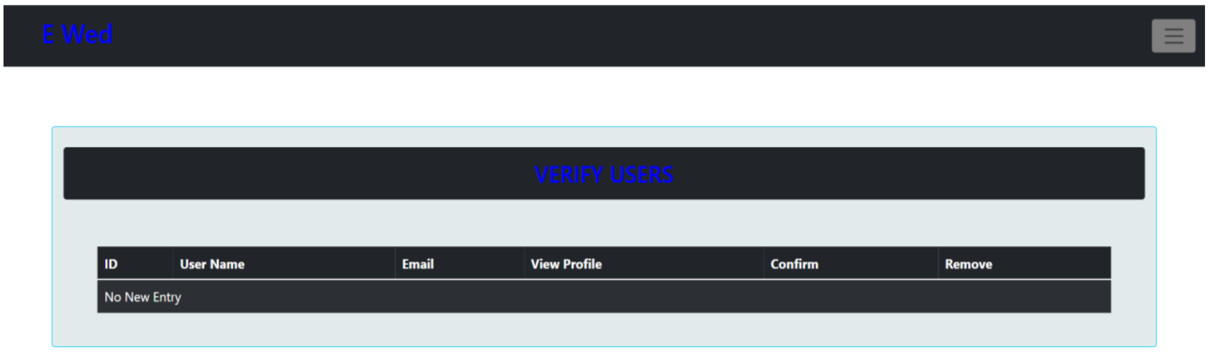
Education : BCA

Date of Birth : 1998-07-09

Documents page



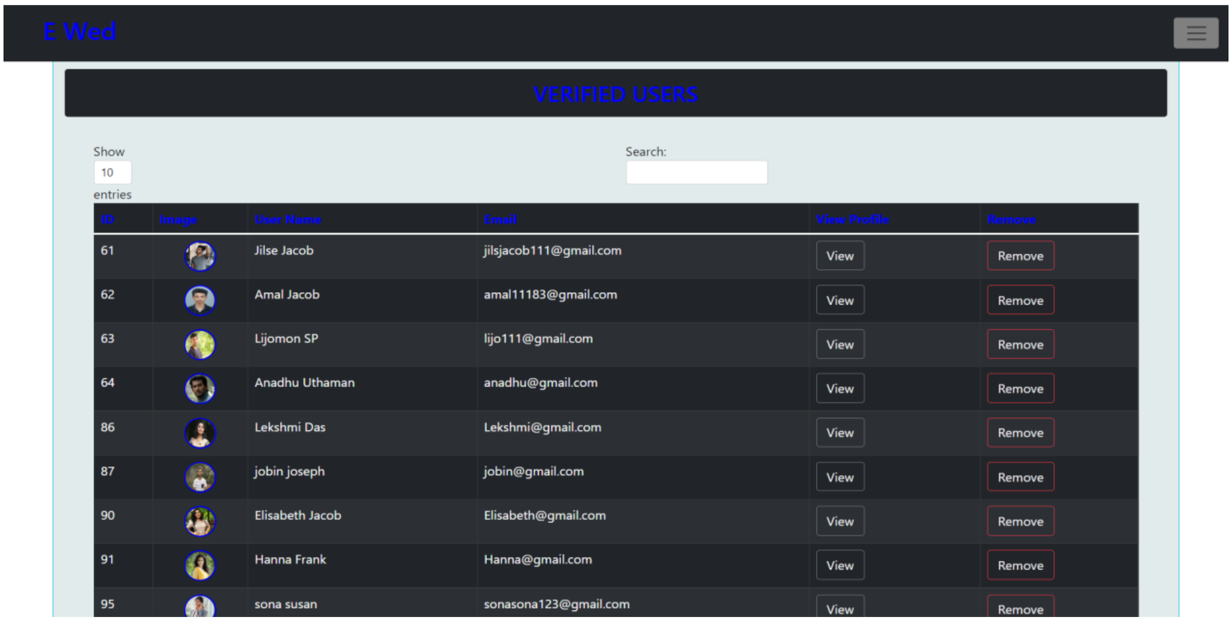
Admin New Users page



VERIFY USERS

ID	User Name	Email	View Profile	Confirm	Remove
No New Entry					

Admin View User Details Page

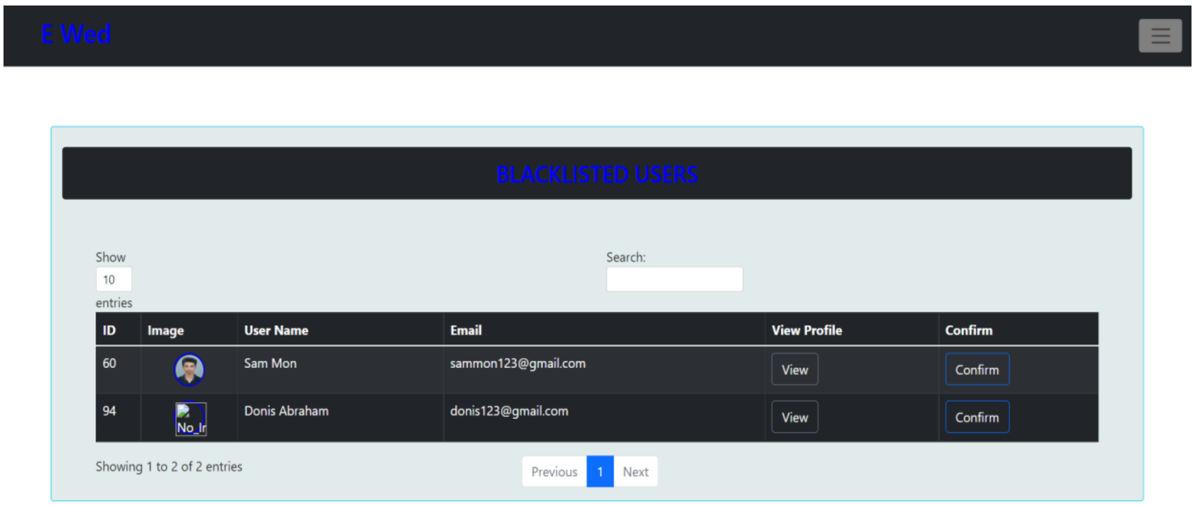


VERIFIED USERS

Show 10 entries Search:

ID	Image	User Name	Email	View Profile	Remove
61		Jilse Jacob	jilsjacob111@gmail.com	View	Remove
62		Amal Jacob	amal11183@gmail.com	View	Remove
63		Lijomon SP	lijo111@gmail.com	View	Remove
64		Anadhu Uthaman	anadhu@gmail.com	View	Remove
86		Lekshmi Das	Lekshmi@gmail.com	View	Remove
87		jobin joseph	jobin@gmail.com	View	Remove
90		Elisabeth Jacob	Elisabeth@gmail.com	View	Remove
91		Hanna Frank	Hanna@gmail.com	View	Remove
95		sona susan	sonasona123@gmail.com	View	Remove

Admin View Blacklisted Users Page



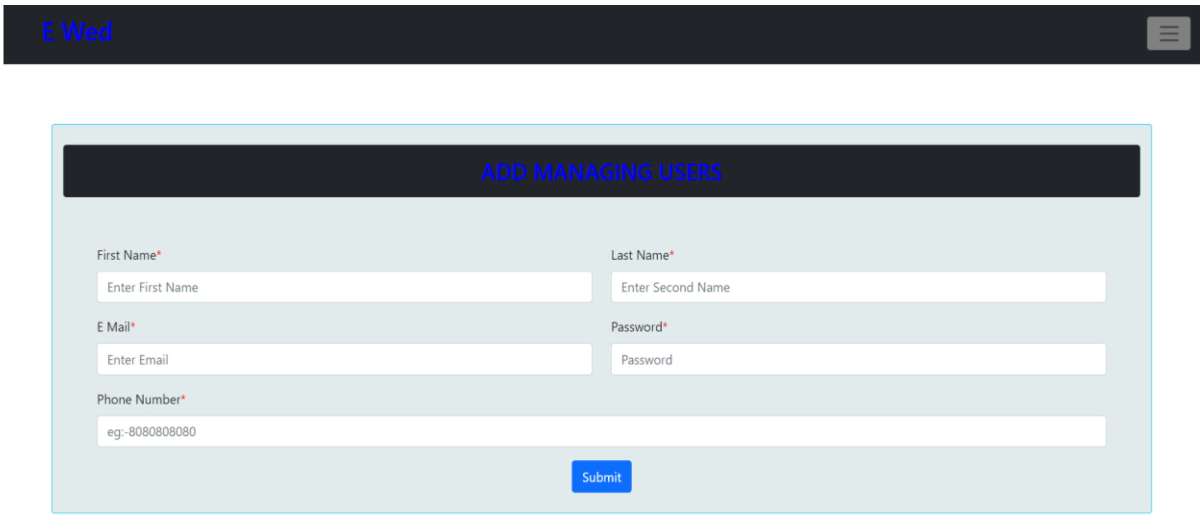
BLACKLISTED USERS

Show 10 entries Search:

ID	Image	User Name	Email	View Profile	Confirm
60		Sam Mon	sammon123@gmail.com	View	Confirm
94	No_Ir	Donis Abraham	donis123@gmail.com	View	Confirm

Showing 1 to 2 of 2 entries

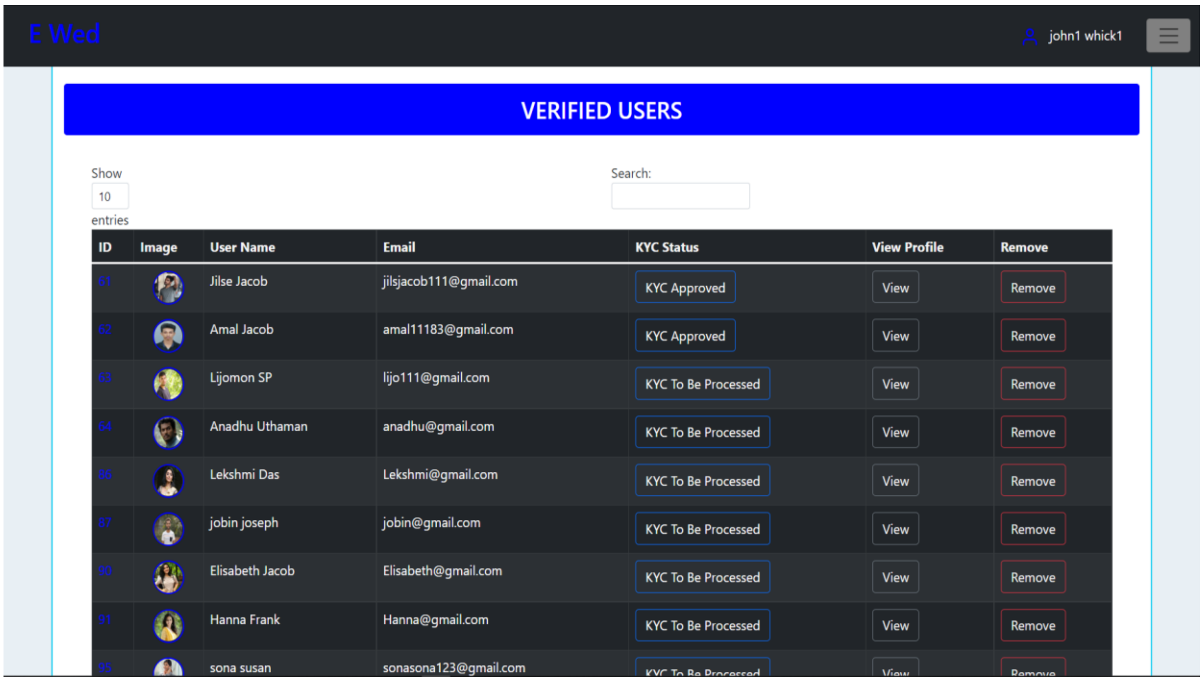
Add managing Users Page



ADD MANAGING USERS

First Name*	Last Name*
Enter First Name	Enter Second Name
E Mail*	Password*
Enter Email	Password
Phone Number*	
eg:-8080808080	

Admin User Index Page



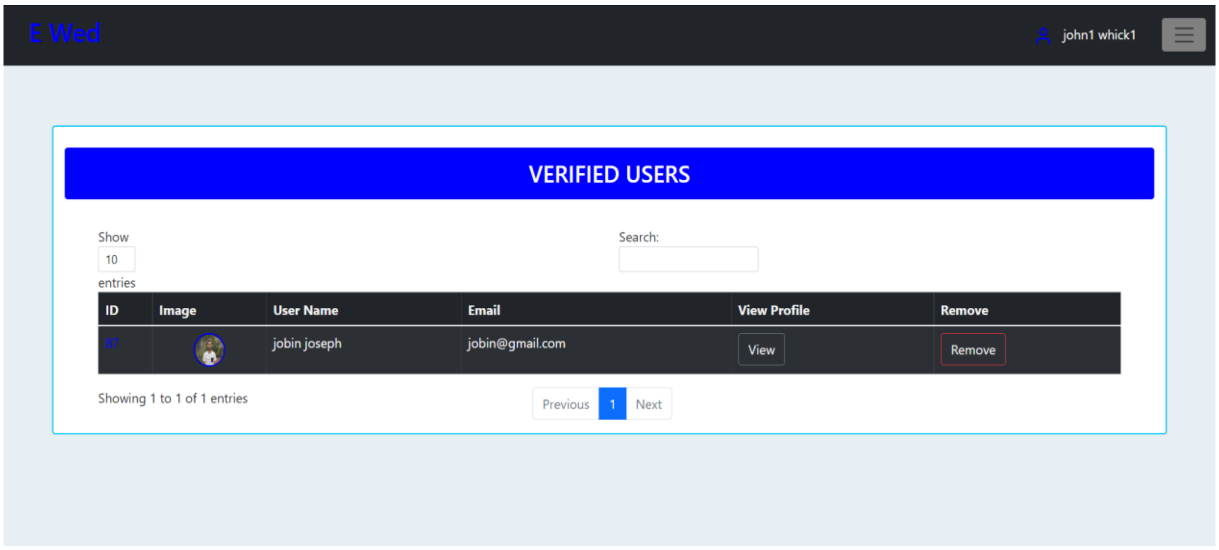
VERIFIED USERS

Show 10 entries

Search:

ID	Image	User Name	Email	KYC Status	View Profile	Remove
61		Jilse Jacob	jilsjacob111@gmail.com	KYC Approved	View	Remove
62		Amal Jacob	amal11183@gmail.com	KYC Approved	View	Remove
63		Lijomon SP	lijo111@gmail.com	KYC To Be Processed	View	Remove
64		Anadhu Uthaman	anadhu@gmail.com	KYC To Be Processed	View	Remove
86		Lekshmi Das	Lekshmi@gmail.com	KYC To Be Processed	View	Remove
87		jobin joseph	jobin@gmail.com	KYC To Be Processed	View	Remove
90		Elisabeth Jacob	Elisabeth@gmail.com	KYC To Be Processed	View	Remove
91		Hanna Frank	Hanna@gmail.com	KYC To Be Processed	View	Remove
95		sona susan	sonasona123@gmail.com	KYC To Be Processed	View	Remove

Admin User Index Page



VERIFIED USERS

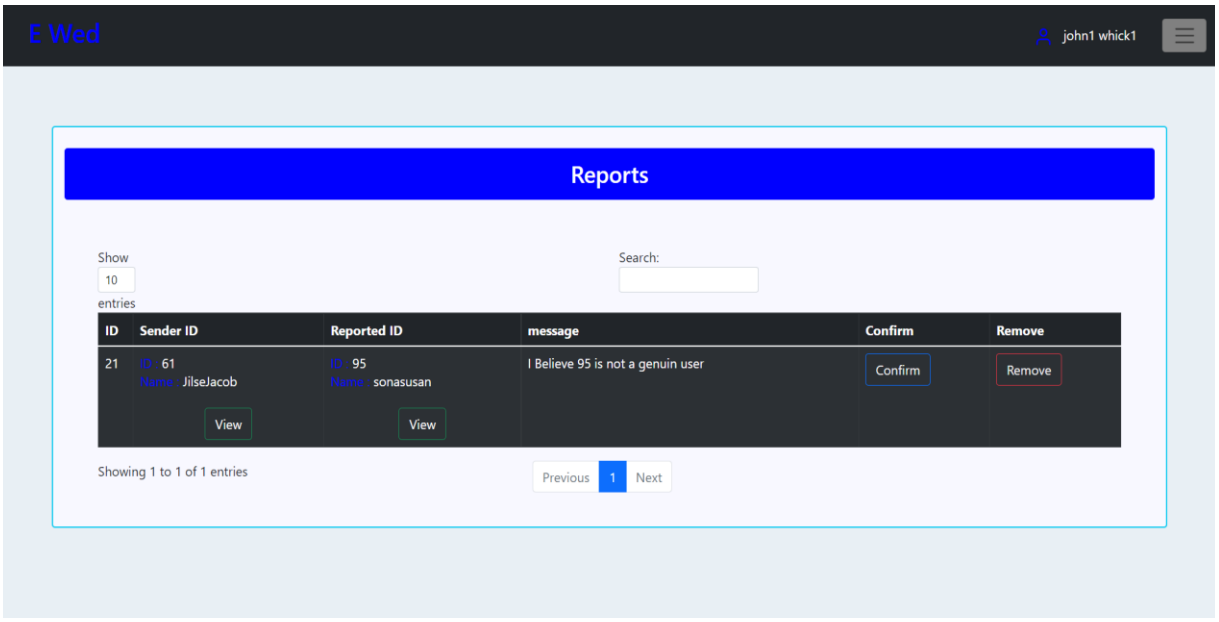
Show
10
entries

Search:

ID	Image	User Name	Email	View Profile	Remove
87		jobin joseph	jobin@gmail.com	View	Remove

Showing 1 to 1 of 1 entries

Admin View Feedback Page



Reports

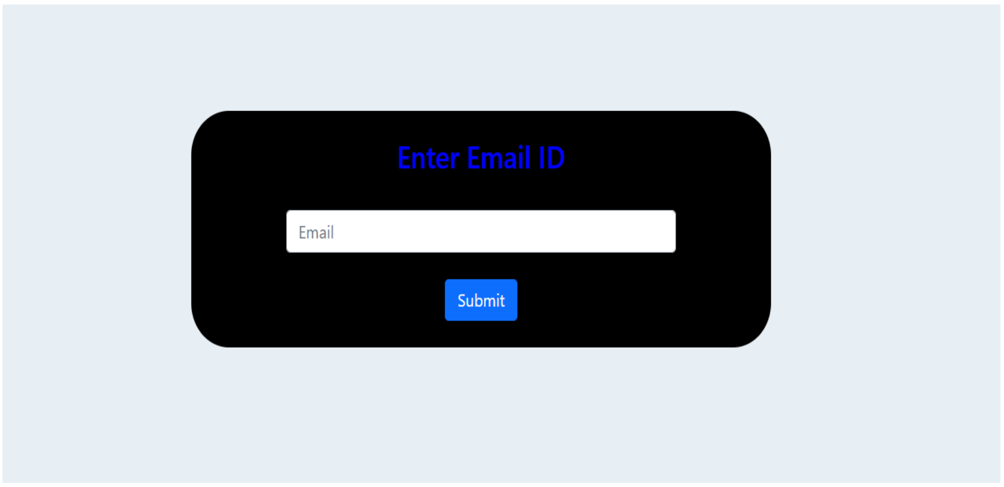
Show
10
entries

Search:

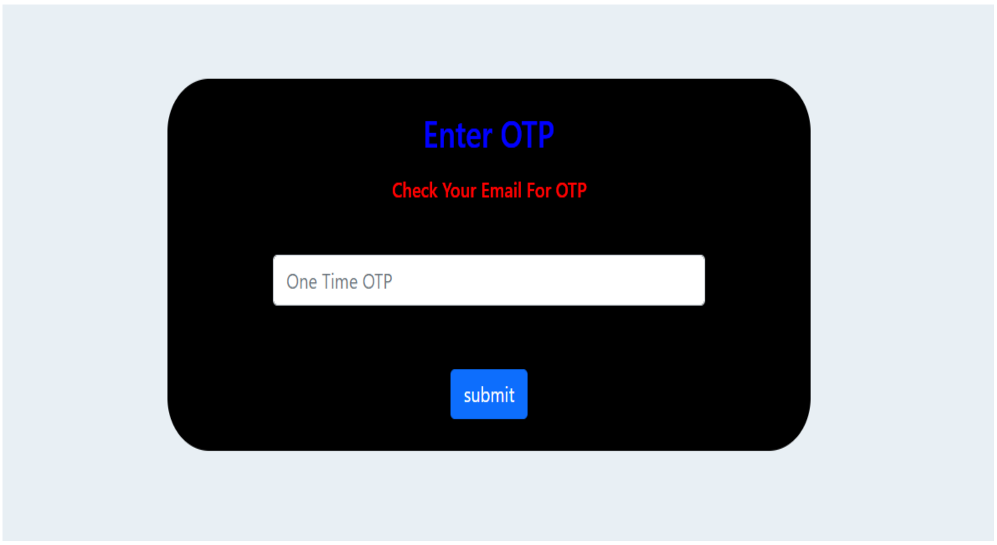
ID	Sender ID	Reported ID	message	Confirm	Remove
21	ID : 61 Name : JilseJacob View	ID : 95 Name : sonasusan View	I Believe 95 is not a genuin user	Confirm	Remove

Showing 1 to 1 of 1 entries

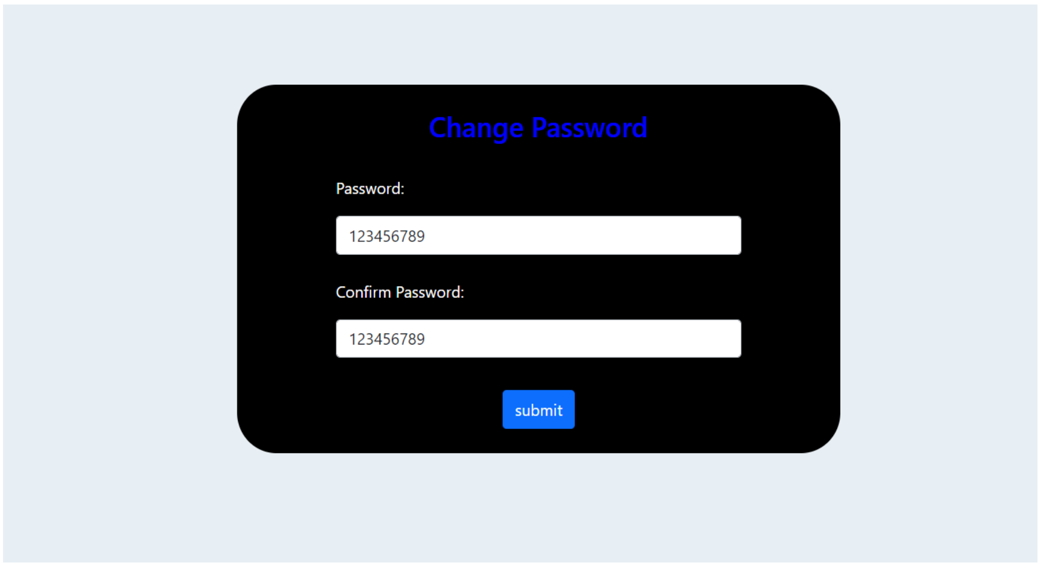
User forgot Password page



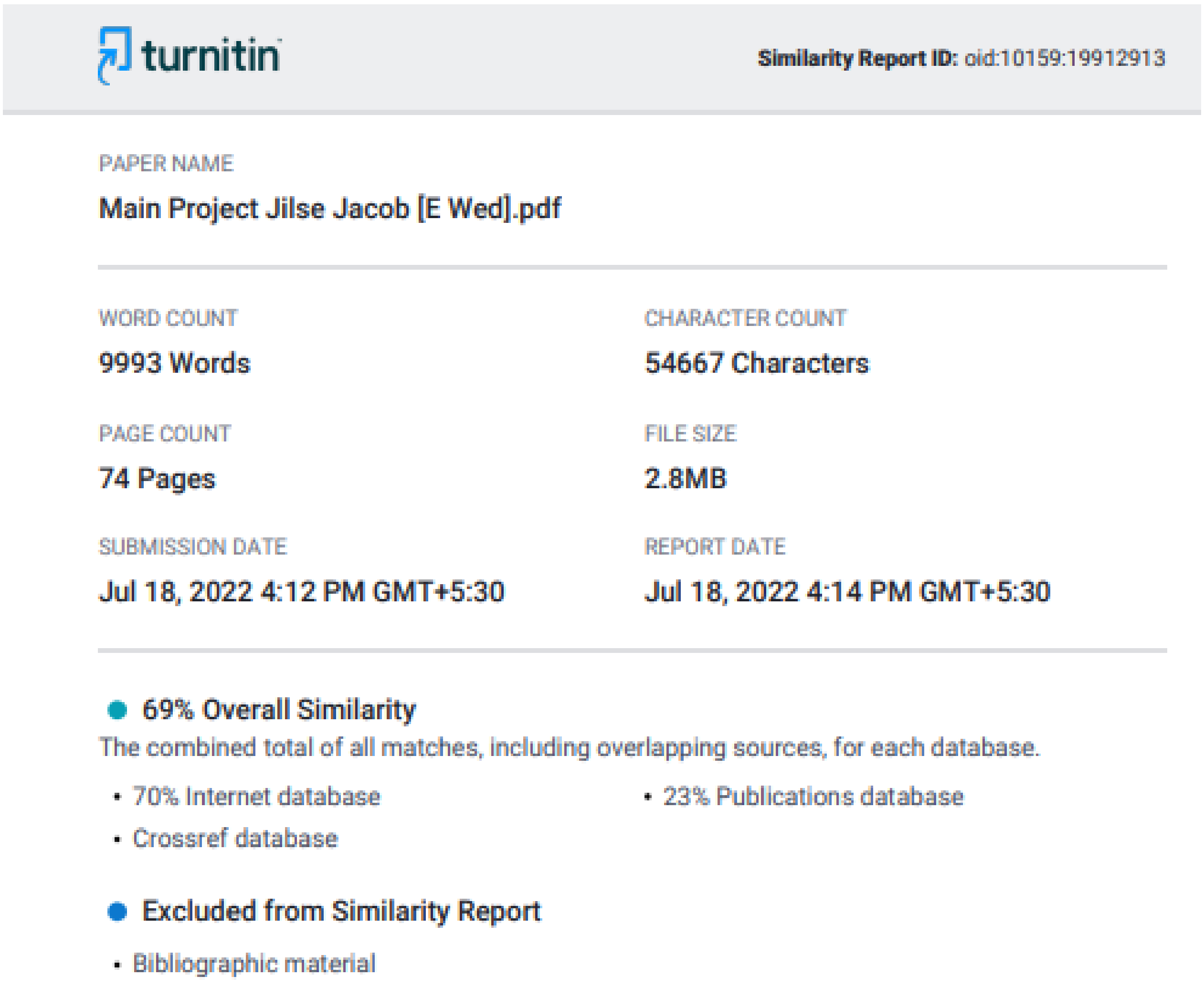
OTP Verification page



New Password Set Page



9.3 PLAGIARISM REPORT



turnitin **Similarity Report ID:** oid:10159:19912913

PAPER NAME

Main Project Jilse Jacob [E Wed].pdf

WORD COUNT	CHARACTER COUNT
9993 Words	54667 Characters
PAGE COUNT	FILE SIZE
74 Pages	2.8MB
SUBMISSION DATE	REPORT DATE
Jul 18, 2022 4:12 PM GMT+5:30	Jul 18, 2022 4:14 PM GMT+5:30

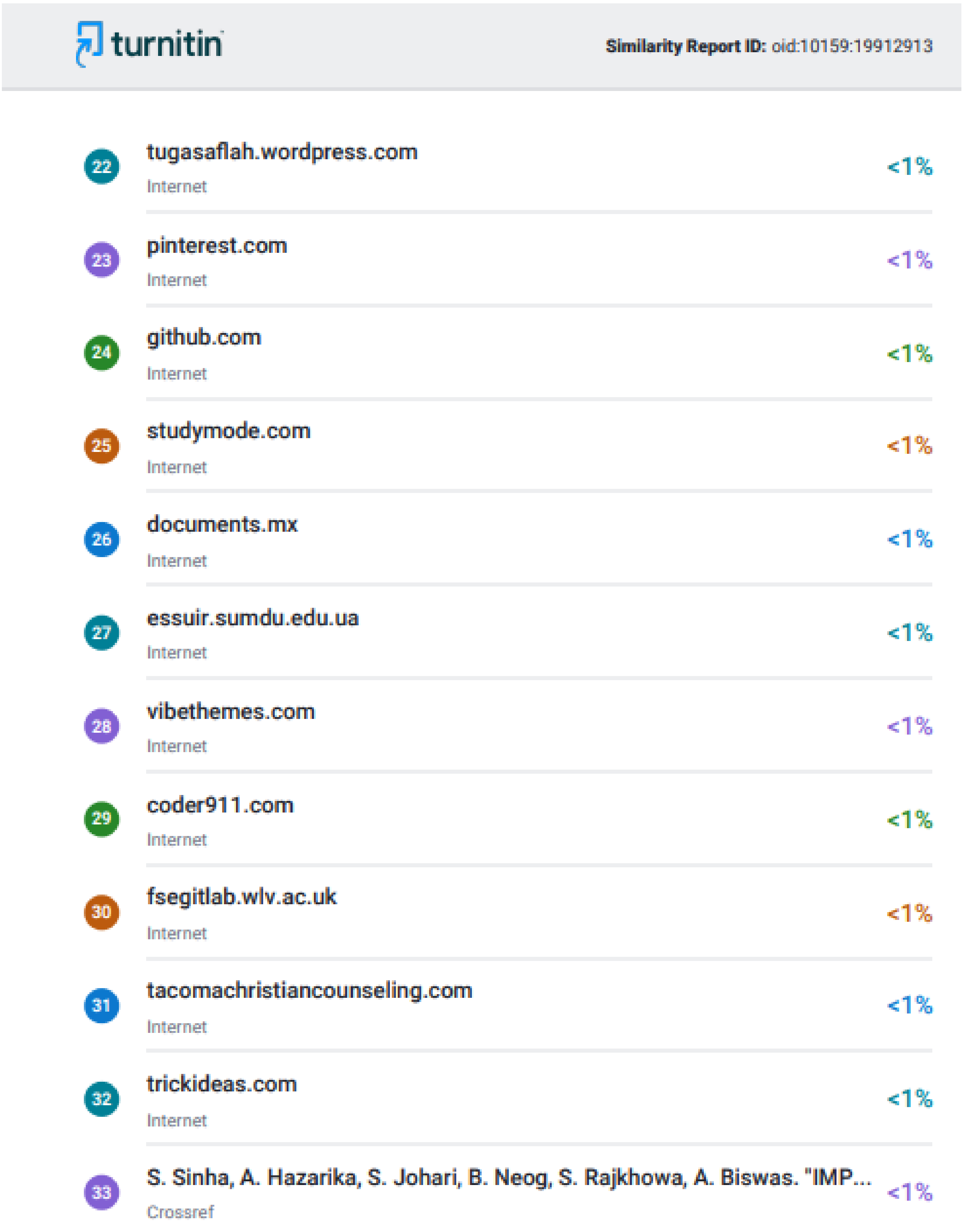
● 69% Overall Similarity

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

- 70% Internet database
- 23% Publications database
- Crossref database

● Excluded from Similarity Report

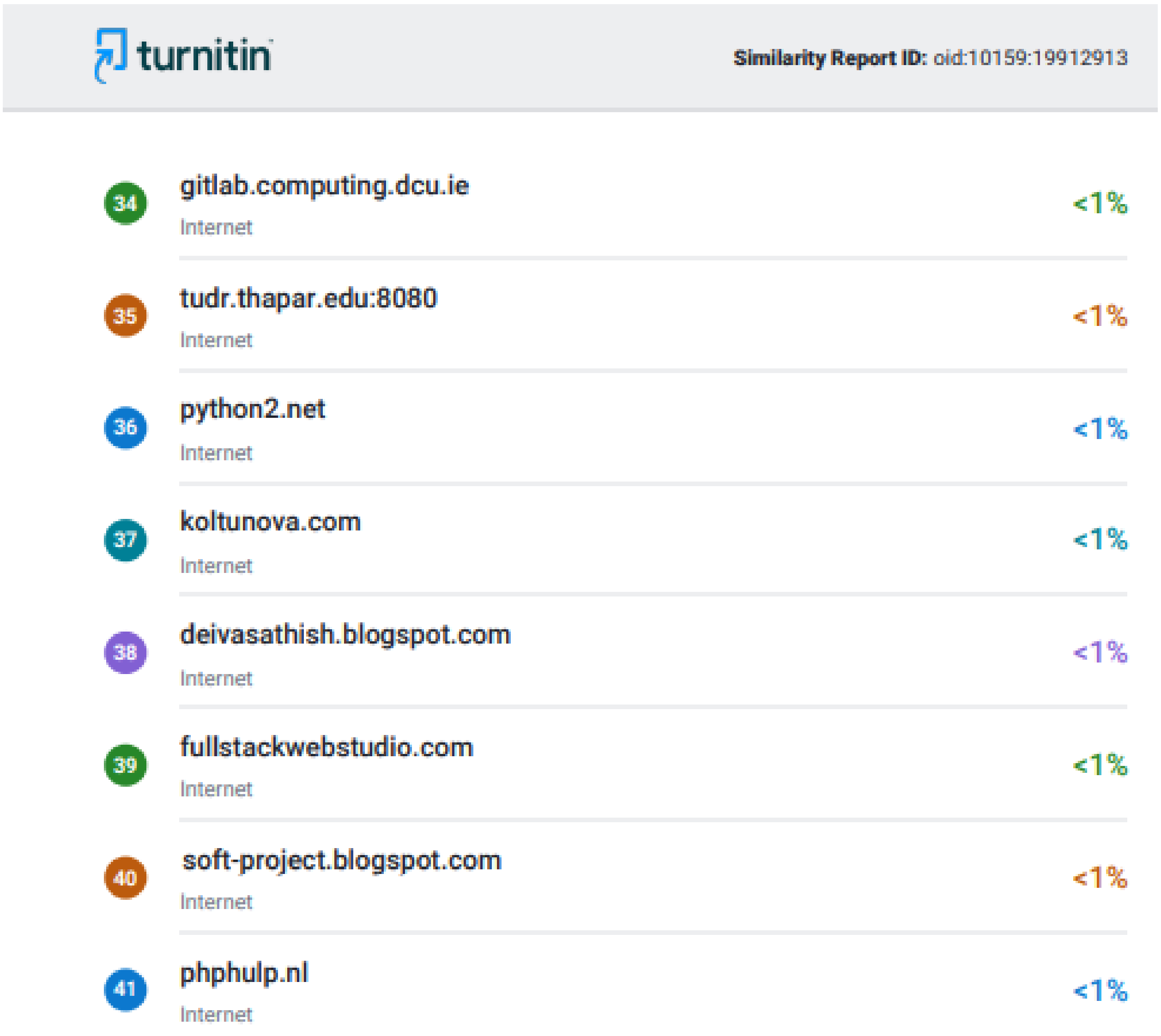
- Bibliographic material



turnitin

Similarity Report ID: oid:10159:19912913

22	tugasaflah.wordpress.com Internet	<1%
23	pinterest.com Internet	<1%
24	github.com Internet	<1%
25	studymode.com Internet	<1%
26	documents.mx Internet	<1%
27	essuir.sumdu.edu.ua Internet	<1%
28	vibethemes.com Internet	<1%
29	coder911.com Internet	<1%
30	fsegitlab.wlv.ac.uk Internet	<1%
31	tacomachristiancounseling.com Internet	<1%
32	trickideas.com Internet	<1%
33	S. Sinha, A. Hazarika, S. Johari, B. Neog, S. Rajkhowa, A. Biswas. "IMP... Crossref	<1%



turnitin

Similarity Report ID: oid:10159:19912913

34	gitlab.computing.dcu.ie Internet	<1%
35	tudr.thapar.edu:8080 Internet	<1%
36	python2.net Internet	<1%
37	koltunova.com Internet	<1%
38	deivasathish.blogspot.com Internet	<1%
39	fullstackwebstudio.com Internet	<1%
40	soft-project.blogspot.com Internet	<1%
41	phphulp.nl Internet	<1%