ONLINE STORAGE FOR DOCUMENT SHARING

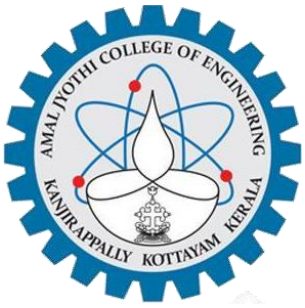
Project Report Submitted By

HARIKRISHNAN R

Reg. No: AJC16MCA-I27

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

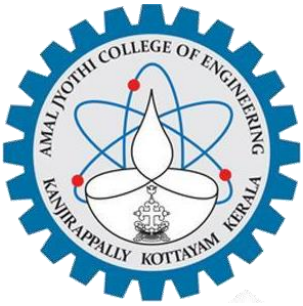
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2020-2021

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, "**ONLINE STORAGE AND DOCUMENT SHARING FORUM"** is the bonafide work of **HARIKRISHNAN R (Reg.No:AJC16MCA-I27)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2020-21.

Ms. Jetty Benjamin
Internal Guide

Rev.Fr.Rubin Thottupuram
Coordinator

Rev.Fr.Rubin Thottupuram
Head of the Department

DECLARATION

I hereby declare that the project report **"ONLINE STORAGE AND DOCUMENT SHARING FORUM"** 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 Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2020-2021.

Date: **HARIKRISHNAN R**

KANJIRAPPALLY **Reg. No: AJC16MCA-I27**

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. Z V Lakaparampil** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr. Rubin Thottupuram** for helping us. I extend my whole hearted thanks to the project coordinators **Rev.Fr. Rubin Thottupuram** and **Ms**. **Sona Maria Sebastian** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms. Jetty Benjamin** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

HARIKRISHNAN R

ABSTRACT

Online Storage and Document Sharing Forum to support learning purposes. To date, online storage has become one of important tools for document storage and management. Online storage has reduced the dependency to the storage devices that are bound to size limit, cost and risk. A part of the storage capability, online storage can be used to share documents by allowing others to access the individual or a group of documents. The widespread use of online storage in the last few years can be attributed to the existence of appealing applications such as file backup, data archival and file sharing. Such systems helps us to store files or data in separate area where you can access it at any time, from anywhere with ease and no cost.

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

"XS-FORUM System" The is a web based "**Online Storage and Document Sharing Forum**" project. The Main purpose of this application will manage file storing, file sharing and also provide platform for sharing queries and having chat section site. This system helps us to store the files or data in separate area where you can access it at any time, from anywhere with ease and no cost. This provide faster as well as simple UI to maintain which brings lesser vagueness in the usage of the site. Such usability or such easiness provide the system to be used by the students those who are started to familiarize the computer working and can be used or targeted for the educational institution for student activates or institutional platform for coding-skills.

Moreover, this not only create the space for storing also the system is provided with a vast area for developers to communicate to each other, personal chat heads can also be maintained by the users. In addition to that building a team for particular project. Are also possible through this system and private space for that team are also provided to ensure the several brains put together to bring a wonderful creative ideas, these team-space can be maintained by the team leader or by the member.

However, current tech update, online-compilers , Q&A section for all users(registered as well as unregistered), addition of business—modules—which provide venture capitals to seek for well program minded youth(find job section).

This system overall helps to increase the potential and knowledge of the users or the community using the system. They Provides varies opportunity to work with others on other projects basics of our interest, give us a platform for bring in other brain to our own projects, finding similar taste and give us a vast area of job opportunities.

1.2 PROJECT SPECIFICATION

The proposed system is made to help the users for an easy and convenient way of store, share and create projects

We will also provide users to give feedbacks, about the interface and for improvements etc.

The system includes 2 modules. They are:

1. Admin Module

All major provisions such as account blocking for any fraudulent activity from the user or another. Looking for the credibility of the answers given by the each users for the queries by others, also marks or small signs are enabled if the answer is 100% sure by the admin sometimes they are auto enabled by the site, going through the feedback from each every one and rectifying their issue over the site and other things. Maintain Database is series of their duty for the functioning the of the site and live news updates over the site. Verification of job details is the major work for the admin they analysis the genuinty in the job alerts given with in the sites by the registered venture capitals or any other if there are not scam the public view to such news are enabled by the admin.Chat clearing and scam massage are deleted from the site db.

2. End-Users Module

The user or further called as the end-users can be divided into several categories:

a. Not a registered a credentials:

They are only allowed to surf through the system, and ask any question or any queries, also provide the online compiler to test the any code which is given in the Q&A section. They are limited to access for the site full potentials. Moreover additional functionalities such as online compilation of specific codes current news update and all details about the site and creator and contacts are provide

b. Registered credentials:

They are real users of the system and they are provided with the functionalities such as:
These users have right to store their file online on the site, they can answer to the queries mentioned in the site by other registered or unregistered users. Those who are looking for members to work with them for a specified project they can create or look for other team members through the function forming Team where they can find perfect partners for their job and also the one who hiring the team members can also mention the constraints for being selected, these teams are having their own chat area for them, option to select the leader role and the leader having the upper hands over others members for operating the sorting and manipulating the to-do list, he can go through each and every profile in the team and can ask for adding further details. Other than this each every users are provided with the view of all real time users and can have private chat with them.
However to become a registered user one must register through the registration link or the signup

section and the sign in to use these functionalities. Moreover additional functionalities such as online compilation of specific codes current news update and have the rights to select the s[pacific type of news to be displayed on the page can be selected and all details about the site and creator and contacts are provide and provision for feedback.

In-addition to this the each users can see their own profile and can edit them, they are also give information about their performance in the site such as showing the number of queries he/she answered, file he stored and past activity or the recent activity. Also provide a provision for resume Building according to the profile.

c. Venture-capitals/start-ups (job providers):

They need to be registered and sign in as a business account member giving all relevant details in the site. Then they can post there job opportunities in the company on this site and other relevant event can also be posted on this sites, appropriate feedback are given by the user for their queries.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

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

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

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

2.2 EXISTING SYSTEM

The existing system is some module concentrated system. Here in this system needs to focus on certain functionalities and they lack up-to-date services. There is no capabilities of interaction and sharing and customizing under one roof. This system gives us very less security for saving data; some data may be lost due to mismanagement. It's a limited system and fewer users friendly. Searching of particular information is very critical it takes lot of time. The users cannot able to restrict the file sharing options. The users only know his information only not others. It is very critical to share public information to all users.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Less convenient
- Human effort is needed.

2.4 PROPOSED SYSTEM

The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach. User Friendliness is provided in the application with various controls provided by system Rich User Interface. The system makes the overall project management much easier and flexible. It can be accessed over the Intranet. Various classes have been used for file uploading , down loading, sharing, chat, and also for creating group for project and communication.

The user information files can be stored in centralized database which can be maintained by the system. This can give the good security for user information because data is not in client machine. Authentication is provided for this application only registered users can access. User can share is data to others, and also he can get data from others. There is no risk of data management at any level while the project development is under process. Report generation features is provided using Data reports to generate different kind of reports.

Similarly users can serach memebers for completing their project as a group also startups can look for hiring purpose

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:

- **One for all:-**

Everything comes under single roof creates less vagueness about all and need not to have separate application for each services.

- **Better security: -**

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

- **Ensure data accuracy: -**

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

- **Better service: -**

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

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

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

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

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

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

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

3.1.2 Technical Feasibility

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

Technical issues raised during the investigation are:

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

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

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.

XS-FORUM, GUI is simple so that users can easily use it. **XS-FORUM** is simple enough so that no training is needed.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

PIV 2.8 GHz Processor and Above

RAM 512MB and Above

HDD 40 GB Hard Disk Space and Above

3.2.2 Software Specification

WINDOWS OS

Internet Information Server 5.0 (IIS)

SQL Server 2005 Enterprise Edition

Front End - HTML, CSS

Backend - MYSQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

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. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, 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:HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface

capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

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

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

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

- **MySQL databases are relational.**

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

- **MySQL software is Open Source**

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

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

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

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

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

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

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

4.2 UML DIAGRAM

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

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML

diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- Statechart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

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

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

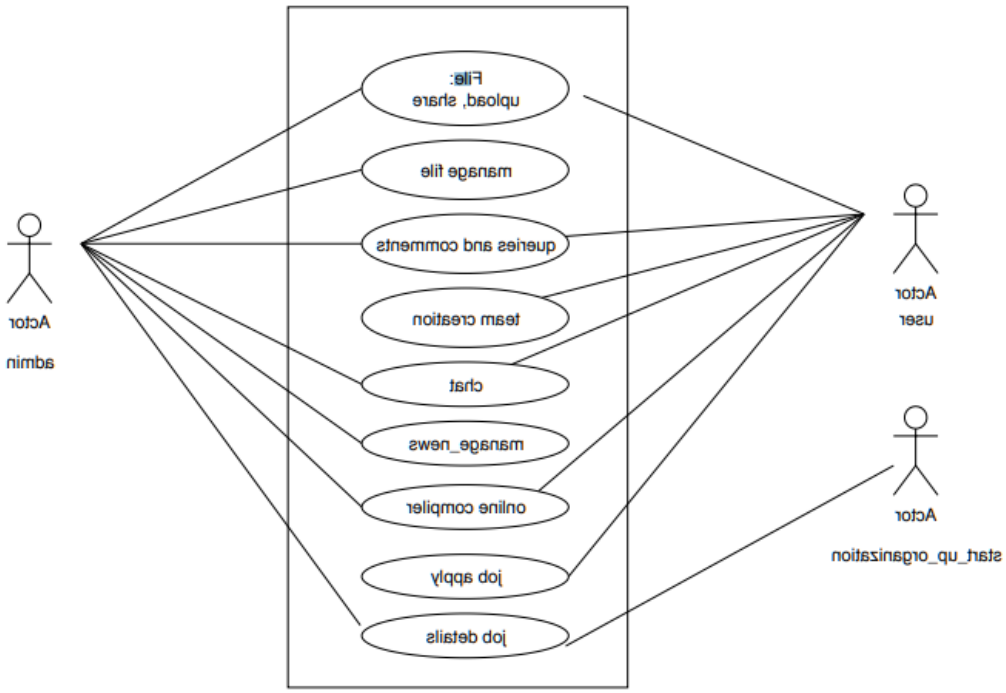
- The boundary, which defines the system of interest in relation to the world around it.
-

- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

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

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

Fig 1 : Use case diagram for **online storing and document sharing forum**



4.2.2 SEQUENCE DIAGRAM

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

Sequence Diagram Notations –

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

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

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

Messages can be broadly classified into the following categories:

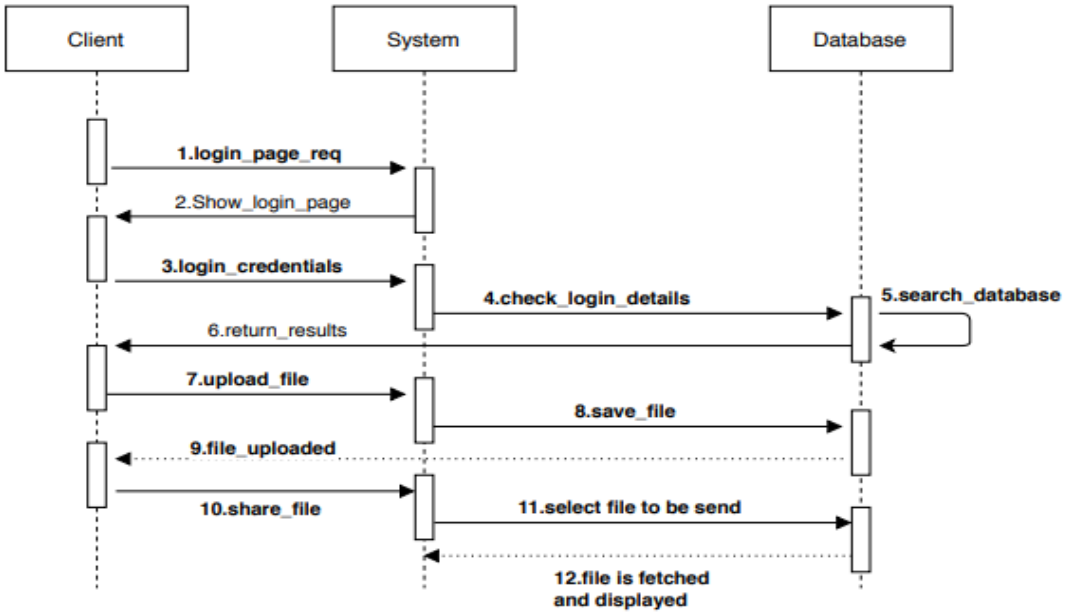
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

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.
- Visualise how messages and tasks move between objects or components in a system.

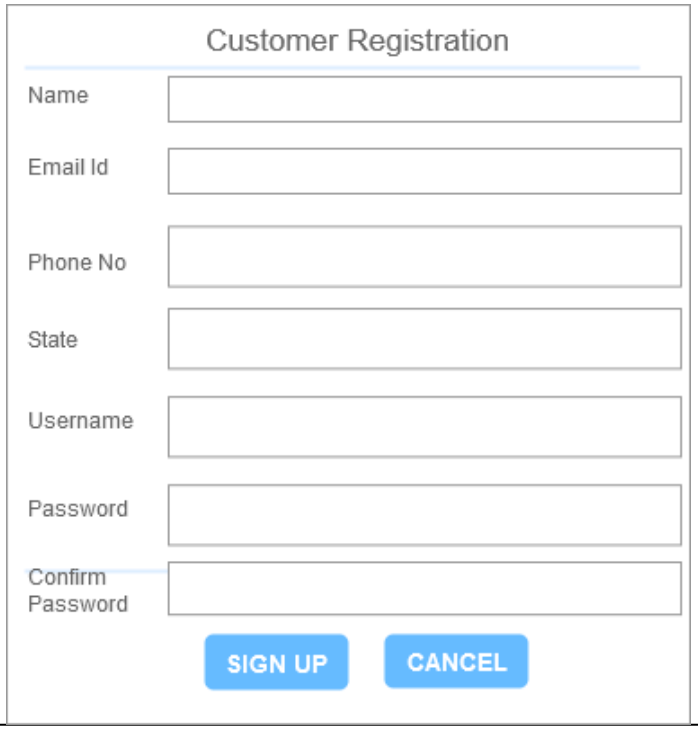
Fig 1 : Sequence diagram for **online storing and document sharing forum**



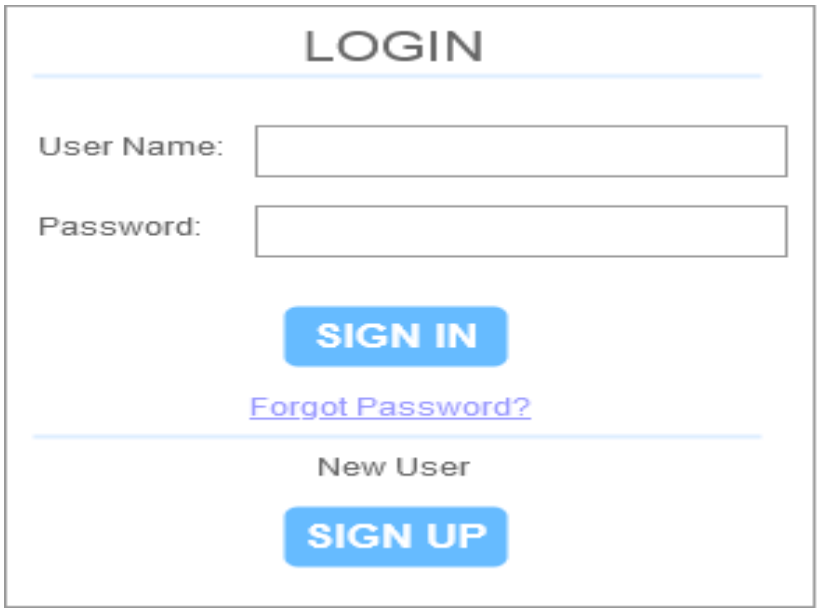
4.5 USER INTERFACE DESIGN

4.5.1-INPUT DESIGN

Form Name : Customer Registration

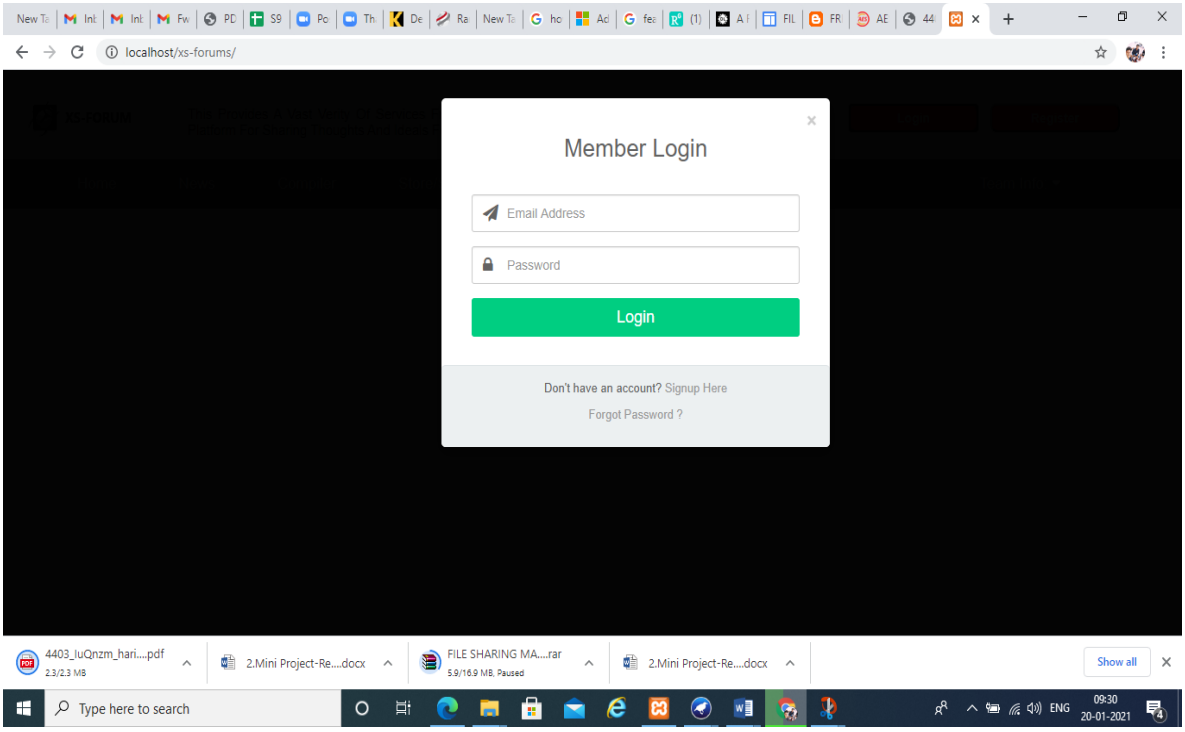


Form Name : User Login

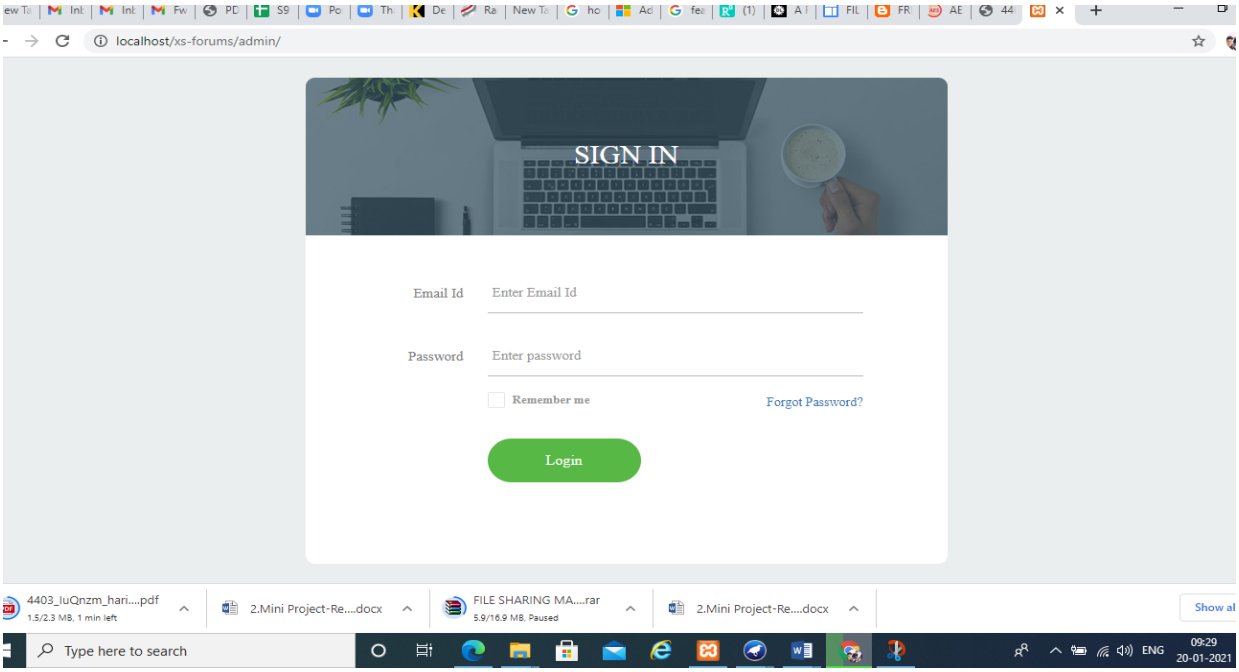


4.5.2 OUTPUT DESIGN

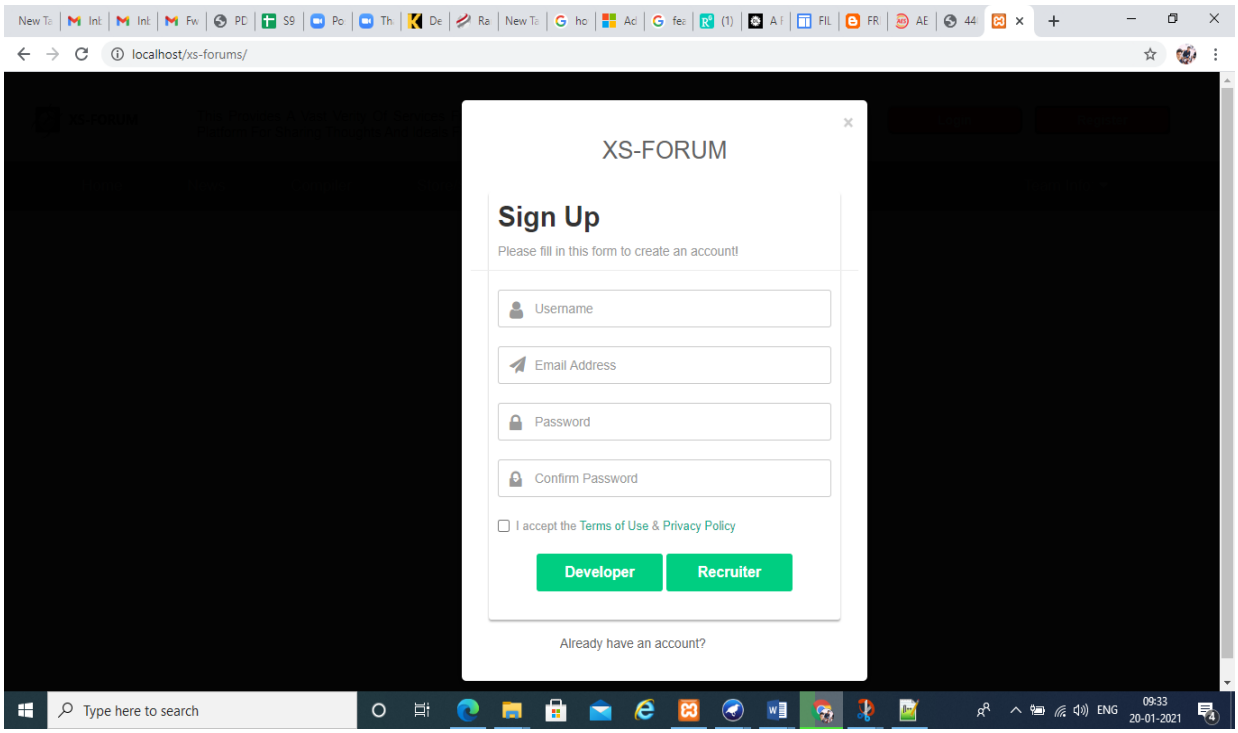
User Login



Admin Login



User Registration



4.6. DATABASE DESIGN

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

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

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

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

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

Relations, Domains & Attributes

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

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

Relationships

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

4.6.2 Normalization

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

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

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

First Normal Form

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

Second Normal Form

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

Third Normal Form

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

TABLE DESIGN

Table No 01 : tbl_reg_users: This table store the login details and the type of the user.

Primary Key : regid

Foreign Key :

Field Name	Type	Size	Description
Regid	int	10	Primary key
Email	Varchar	25	email for login
Password	VARCHAR	25	Password for login
Type	Varchar	25	Type of user

Table No 02 : tbl_account: This table is for storing the details or profile of the registered users

Primary Key : accid

Foreign Key : regid

Field Name	Type	Size	Description
Acid	int	10	Primary key
name	varchar	25	Name of the reg user
username	varachar	50	Username for registered user
phone	varchar	25	Mobile number
about	text		User description
stastus	varchar	25	User is online/offline for chat purpose
valid	varchar	25	To check the validity of the user
regid	int	10	Foreign key from (tbl_reg_users)

Table No 03 : tbl_company_details: This table is for storing the details or profile of the registered company users

Primary Key : compid

Foreign Key : regid

Field Name	Type	Size	Description
compid	int	10	Primary key
Name	Varchar	25	Name of representation of the company
cname	varchar	25	Company name
email	varchar	50	Reg company heads mail
phone	varchar	25	Mobile number company reg user
about	text		About company user
Position	Varchar	25	Position in company
valid	varchar	25	To check the validity of the user
regid	int	10	Foreign key from (tbl_reg_users)

Table No 04 : tbl_file_store: Used to store Data/files by the registered users

Primary Key : fileid

Foreign Key : regid

Field Name	Type	Size	Description
fileid	int	10	Primary key
File	blob		File uploaded
filename	varchar	25	
Date	Date		Uploaded date
description	VARCHAR	255	Short note on type of file
regid	int	10	Forign key from (tbl_reg_users)

Table No 05 : tbl_questions: questions asked by the users are stored here for displaying purpose

Primary Key : qid

Foreign Key : regid

Field Name	Type	Size	Description
qid	int	10	Primary key
date	Datetime		Date on which question posted

content	text		Content of the question
valid	Varchar	25	Check the genuinely
answer_count	Int	20	Number of answers for each question
regid	int	10	Foreign key from (tbl_reg_users)

Table No 06 : tbl_answers: answers given by users for each questions are stored

Primary Key : aid

Foreign Key : qid

Field Name	Type	Size	Description
aid	int	10	Primary key
qdate	Datetime		Date on which answer posted
content	text		Content of the answer
valid	Varchar	25	Check the genuinity
approved	varchar	25	Check approval of content by admin or compilation by site
regid	int	10	Foreign key from (tbl_reg_users)
qid	int	10	Foreign key from (tbl_questions)

Table No 07 : tbl_grp_prj: Table for storing each group details for their communication ease

Primary Key : gid

Foreign Key : regid

Field Name	Type	Size	Description
gid	int	10	Primary key
Gname	varchar	25	Group name
role	varchar	25	Each role for memeber in team (ex:team leader)
Valid_member	int	25	To check the validity of the user in group
valid	int	25	Exist or not
Notification_valid	Int	10	Recruit for joining team is closed or not
regid	int	10	Foreign key from (tbl_reg_users)

Table No 08 : tbl_todo_list: Used to store day to day task given by leaders

Primary Key : todo_id

Foreign Key : gid

Field Name	Type	Size	Description
Todo_id	int	10	Primary key
title	Varchar	25	Title for next task
Content	Mediumtext		List out things to do
date	Datetime		Date of declaration
gid	int	10	Foreign key from (tbl_group)

Table No 09 : tbl_individual_chat: Used for enabling msg between individual registered user

Primary Key : msgid

Foreign Key : from_user_id

Field Name	Type	Size	Description
Msgid	int	10	Primary key
from_user_id	int	25	Foreign key from (tbl_reg_users(regid))
to_user_id	int		List out things to do
Msg	text		Content of msg

Table No 10 : tbl_grp_chat: Used for enabling msg in a group

Primary Key : msgid

Foreign Key : from_user_id

Field Name	Type	Size	Description
msgid	int	10	Primary key
from_user_id	int	25	Foreign key from (tbl_reg_users(regid))
msg	text		Content of msg
gid	int	10	Foreign key from (tbl_group)

Table No 11 : tbl_job: Job entry given by each job providing registered organistation

Primary Key : jobid

Foreign Key : regid

Field Name	Type	Size	Description
jobid	int	10	Primary key
Company	varchar	100	Company name
tittle	varchar	50	Job title
Job_description	text		About job details
Eligibility	Text		Mention criterie for applying
Last_date	datetime		Mention the last date submission
valid	varchar	25	Validity of the job
compid	int	10	Foreign key from (tbl_reg_users)..company regid

Table No 12 : tbl_jobapplication: Table for storing the applicant details those who applied for job

Primary Key : appid

Foreign Key : jobid

Field Name	Type	Size	Description
appid	Int	10	Primary key
name	varchar	25	Name of the reg user
email	varachar	50	Reg email id
phone	varchar	25	Mobile number
Address	Varchar	100	Address of the applicant
City	Varchar	25	City he reside
State	Varchar	23	State he reside
Resume	Blob		Resume of the applicant
about	Text		About himself
valid	varchar	25	Valid application or closed application
jobid	Int	10	Foreign key from (tbl_job)

regid	Int	10	Foreign key from (tbl_reg_users)

Table No 13 : tbl_enquiry: Table for storing the enquiries or complaints by the reg users

Primary Key : eqid

Foreign Key : regid

Field Name	Type	Size	Description
eqid	int	10	Primary key
subject	varchar	25	Title of enquiry
about	text		Content of enquiry
status	varchar	25	Check whether rectified or not
regid	int	10	Foreign key from (tbl_reg_user)

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

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

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

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

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

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

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

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

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

5.2 TEST PLAN

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

The levels of testing include:

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

5.2.1 Unit Testing

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

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

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

5.2.2 Integration Testing

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

5.2.3 Validation Testing or System Testing

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

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

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

5.2.4 Output Testing or User Acceptance Testing

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

- Input Screen Designs,
- Output Screen Designs,

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

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

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

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

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

The implementation state involves the following tasks:

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

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In

many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

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

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

6.2.1 User Training

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

6.2.2 Training on the Application Software

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

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System

maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge about the web development. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that startups can hire peoples according to their needs.
- Many internships are viable through platform
- Helps to find out common communities
- Communication through chats.
- Transferring data with high security
- Helps to have a virtual common area for a small ventures.
- Data security can be enhanced.

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
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample Code

Login.php

```
<?php
session_start();
include('database.php');
if(isset($_POST['login']))
{
$email=$_POST['email'];
$pass=$_POST['password'];
$password = md5($pass);
$sql ="SELECT regid,email,type FROM tbl_reg_users WHERE email='$email' and password ='$password'
and verification = '1'";
$query=mysqli_query($con,$sql);
$r=mysqli_num_rows($query);

        if($r > 0)
        {
                while($results=mysqli_fetch_array($query))
                {
                        if($results['type'] == 'user')
                        {
                                $id = $results['regid'];
                                $_SESSION['id'] = $results['regid'];
                                $s = "update tbl_account set stastus='online' where regid='$id'";
                                $q=mysqli_query($con,$s);
                                $sql1 = "SELECT username FROM tbl_account WHERE regid=$id";
                                $query1=mysqli_query($con,$sql1);
                                $r1=mysqli_num_rows($query1);
                                if($r1 > 0)
                                {
                                        while($results1=mysqli_fetch_array($query1))
                                        {
                                                $_SESSION['uname']=$results1['username'];
                                        }
                                }
                        $_SESSION['login']=$results['regid'];
                        $_SESSION['email']=$results['email'];
                        $_SESSION['type'] = $results['type'];
                                if ($results['type'] == 'user')
                                {
                                echo "<script type='text/javascript'> document.location = 'index.php';
</script>";
                                }
                        }
                        else
                        {
                                $id = $results['regid'];
                                $_SESSION['id'] = $results['regid'];
                                $s = "update tbl_cmpny_account set stastus='online' where regid='$id'";
```

```
                                $q=mysqli_query($con,$s);
                                $sql1 = "SELECT username FROM tbl_cmpny_account WHERE
regid=$id";
                                $query1=mysqli_query($con,$sql1);
                                $r1=mysqli_num_rows($query1);
                                if($r1 > 0)
                                {
                                        while($results1=mysqli_fetch_array($query1))
                                        {
                                                $_SESSION['uname']=$results1['username'];
                                        }
                                }
                        $_SESSION['login']=$results['regid'];
                        $_SESSION['email']=$results['email'];
                                if ($results['type'] == 'company')
                                {
                                echo "<script type='text/javascript'> document.location = 'index.php';
</script>";
                                }
                        }
                }
        }
        else
        {
          echo "<script>alert('Invalid Details/Account is Not verified');</script>";
        }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Elegant Modal Login Modal Form with Icons</title>
<link href="https://fonts.googleapis.com/css?family=Roboto|Varela+Round" rel="stylesheet">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<style>

        .modal-login {
                color: #636363;
                width: 350px;
                margin: 30px auto;
        }
        .modal-login .modal-content {
                padding: 20px;
                border-radius: 5px;
                border: none;
```

```
    }
    .modal-login .modal-header {
            border-bottom: none;
            position: relative;
            justify-content: center;
    }
    .modal-login h4 {
            text-align: center;
            font-size: 26px;
    }
    .modal-login  .form-group {
            position: relative;
    }
    .modal-login i {
            position: absolute;
            left: 13px;
            top: 11px;
            font-size: 18px;
    }
    .modal-login .form-control {
            padding-left: 40px;
    }
    .modal-login .form-control:focus {
            border-color: #00ce81;
    }
    .modal-login .form-control, .modal-login .btn {
            min-height: 40px;
            border-radius: 3px;
    }
    .modal-login .hint-text {
            text-align: center;
            padding-top: 10px;
    }
    .modal-login .close {
position: absolute;
            top: -5px;
            right: -5px;
    }
    .modal-login .btn {
            background: #00ce81;
            border: none;
            line-height: normal;
    }
    .modal-login .btn:hover, .modal-login .btn:focus {
            background: #00bf78;
    }
    .modal-login .modal-footer {
            background: #ecf0f1;
            border-color: #dee4e7;
            text-align: center;
            margin: 0 -20px -20px;
            border-radius: 5px;
            font-size: 13px;
```

```
                justify-content: center;
        }
        .modal-login .modal-footer a {
                color: #999;
        }
        .trigger-btn {
                display: inline-block;
                margin: 100px auto;
        }
</style>

</head>
<body>
<!-- Modal HTML -->

<div id="myModallog" class="modal fade">
        <div class="modal-dialog modal-login">
                <div class="modal-content">
                        <div class="modal-header">
                                <h4 class="modal-title">Member Login</h4>
                                <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">&times;</button>
                        </div>
                        <div class="modal-body">
                                <form method="post">
                                        <div class="form-group">
                                <i class="fa fa-paper-plane"></i>
                                <input type="email" class="form-control" name="email"
placeholder="Email Address" required="required">
                                        </div>
                                        <div class="form-group">
                                                <i class="fa fa-lock"></i>
                                                <input type="password" name='password' class="form-
control" placeholder="Password" required="required">
                                        </div>
                                        <div class="form-group">
                                                <input type="submit" name='login' id='signin'class="btn
btn-primary btn-block btn-lg" value="Login">
                                        </div>
                                </form>
                        </div>
                        <div class="modal-footer text-center">
      <p>Don't have an account? <a href="#myModalreg" data-toggle="modal" data-
dismiss="modal">Signup Here</a></p>
      <p><a href="#myModalgetotp" data-toggle="modal" data-dismiss="modal">Forgot Password
?</a></p>
    </div>
                </div>
        </div>
</div>
</body>
</html>
```

Register.php

```
<?php
include('database.php');
$msg=" lkhdfglhdsflghl";
$username = $_POST['username'];
		$pass = $_POST['password'];
		$password = md5($pass);
		$email = $_POST['email'];
if(isset($_POST['signupdev']))
{
$verification_id=rand(111111111,999999999);
$sql="INSERT INTO  tbl_reg_users (email, password,type,vid) VALUES('$email','$password','user','$verification_id')";
$query = mysqli_query($con,$sql);
$id = mysqli_insert_id($con);
$sql1="INSERT INTO  tbl_account (username, regid) VALUES('$username',$id)";
$query1 = mysqli_query($con,$sql1);
if ($query1)
{
		echo "<script>alert('We have just sent a verification link to $email. Please check your inbox and click on the link to get started. If you cant find this email (which could be due to spam filters), just request a new one here.');</script>";
				//mysqli_query($con,"insert into user(name,email,password,verification_status,verification_id) values('$username','$email','$password',0,'$verification_id')");

			$mailHtml="Please confirm your account registration by clicking the button or link below: <a href='http://127.0.0.1/php/email_verification/verification_email.php?id=$verification_id'>http://127.0.0.1/php/email_verification/verification_email.php?id=$verification_id</a>";

			smtp_mailer($email,'Account Verification',$mailHtml);

}
else
{
		echo "<script>alert('Something went wrong. Please try again');</script>";
}}
else
{
		if(isset($_POST['signuprec']))
		{
			$verification_id=rand(111111111,999999999);
		$sql="INSERT INTO  tbl_reg_users (email, password,type,vid) VALUES('$email','$password','company','$verification_id')";
$query = mysqli_query($con,$sql);
$id = mysqli_insert_id($con);
$sql1="INSERT INTO  tbl_cmpny_account (username, email, regid) VALUES('$username','$email','$id')";
$query1 = mysqli_query($con,$sql1);
if ($query)
{
			$msg="We've just sent a verification link to <strong>$email</strong>. Please check your inbox and click on the link to get started. If you can't find this email (which could be due to spam filters), just request a new one here.";

			$mailHtml="Please confirm your account registration by clicking the button or link below: <a href='http://127.0.0.1/php/email_verification/verification_email.php?id=$verification_id'>http://127.0.0.1/php/email_verification/verification_email.php?id=$verification_id</a>";

			smtp_mailer($email,'Account Verification',$mailHtml);
}
else
{
		echo "<script>alert('Something went wrong. Please try again');</script>";
		}}
}

function smtp_mailer($to,$subject, $msg){
```

```
		require_once("smtp/class.phpmailer.php");
		$mail = new PHPMailer();
		$mail->IsSMTP();
		$mail->SMTPDebug = 1;
		$mail->SMTPAuth = true;
		$mail->SMTPSecure = 'TLS';
		$mail->Host = "smtp.sendgrid.com";
		$mail->Port = 80;
		$mail->IsHTML(true);
		$mail->CharSet = 'UTF-8';
		$mail->Username = "harisgs07@gmail.com";
		$mail->Password = "ambilyradha7@";
		$mail->SetFrom("harisgs07@gmail.com",'admin');
		$mail->Subject = $subject;
		$mail->Body =$msg;
		$mail->AddAddress($to);
		if(!$mail->Send()){
			return 0;
		}else{
			return 1;
		}
}

?>

<script>
function checkAvailabilityemail1() {
$("#loaderIcon").show();
jQuery.ajax({
url: "check.php",

data:'email='+$("#emailid").val(),
type: "POST",
success:function(data){
$("#user-availability-status").html(data);
$("#loaderIcon").hide();
},
error:function (){}
});
}
</script>
<script>
function validatingpassword1() {
$("#loaderIcon").show();
jQuery.ajax({
url: "check.php",

data:'pass='+$("#passwordid").val(),
type: "POST",
success:function(data){
$("#user-availability-statusa").html(data);
$("#loaderIcon").hide();
},
error:function (){}
});
}
</script>

<script type="text/javascript">
function valid()
{
if(document.signup.password.value!= document.signup.confirm_password.value)
{
alert("Password and Confirm Password Field do not match  !!");
```

```
document.signup.confirm_password.focus();
return false;
}

return true;
}
</script>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://fonts.googleapis.com/css?family=Roboto:400,700" rel="stylesheet">
<title>Bootstrap Sign up Form with Icons</title>
<link href="https://fonts.googleapis.com/css?family=Roboto|Varela+Round" rel="stylesheet">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<style>
        div.modal-content
        {
                width:450px;
        }
        .form-control, .form-control:focus, .input-group-addon {
                border-color: #e1e1e1;
        }
  .form-control, .btn {
    border-radius: 3px;
  }
        .signup-form {
                width: 390px;
                margin: 0 auto;
                padding: 10px 0;
        }
  .signup-form form {
                color: #999;
                border-radius: 3px;
        margin-bottom: 15px;
    background: #fff;
    box-shadow: 0px 2px 2px rgba(0, 0, 0, 0.3);
    padding: 10px;
  }
        .signup-form h2 {
                color: #333;
                font-weight: bold;
    margin-top: 0;
  }
  .signup-form hr {
    margin: 0 -30px 20px;
  }
        .signup-form .form-group {
                margin-bottom: 20px;
        }
        .signup-form label {
                font-weight: normal;
                font-size: 13px;
        }
        .signup-form .form-control {
                min-height: 38px;
                box-shadow: none !important;
        }
```

```
            .signup-form .input-group-addon {
                        max-width: 42px;
                        text-align: center;
            }
            .signup-form input[type="checkbox"] {
                        margin-top: 2px;
            }
    .signup-form .btn{
        font-size: 16px;
        font-weight: bold;
                        background: #19aa8d;
                        border: none;
                        min-width: 140px;
    }
            .signup-form .btn:hover, .signup-form .btn:focus {
                        background: #179b81;
        outline: none;
            }
            .signup-form a {
                        color: #fff;
                        text-decoration: underline;
            }
            .signup-form a:hover {
                        text-decoration: none;
            }
            .signup-form form a {
                        color: #19aa8d;
                        text-decoration: none;
            }
            .signup-form form a:hover {
                        text-decoration: underline;
            }
            .signup-form .fa {
                        font-size: 21px;
            }
            .signup-form .fa-paper-plane {
                        font-size: 18px;
            }
            .signup-form .fa-check {
                        color: #fff;
                        left: 17px;
                        top: 18px;
                        font-size: 7px;
                        position: absolute;
            }
            .signup-form .message{
                        color:red;
            }
</style>
</head>
<body>
<div id="myModalreg" class="modal fade">
            <div class="modal-dialog modal-login">
                        <div class="modal-content">
                                    <div class="modal-header">
                                                <h4 class="modal-title">XS-FORUM</h4>
                                                <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">&times;</button>
                                    </div>
<div class="signup-form">
   <form  method="post" name ='signup' onSubmit="return valid();">
                        <h2>Sign Up</h2>
                        <p>Please fill in this form to create an account!</p>
                        <hr>
```

```
<div class="form-group">
                                        <i class="fa fa-user"></i>
                                        <input type="text" class="form-control" name="username" placeholder="Username"
required="required">
</div>
<div class="form-group">
                                        <i class="fa fa-paper-plane"></i>
                                        <input type="email" class="form-control" id='emailid' name="email"
onBlur="checkAvailabilityemail1()" placeholder="Email Address" required="required">
                                        <span id="user-availability-status" style="font-size:12px;"></span>
</div>
                <div class="form-group">
                                        <i class="fa fa-lock"></i>
                                        <input type="text" class="form-control" name="password" id='passwordid'
placeholder="Password"  onBlur="validatingpassword1()"  required="required">
                                        <span id="user-availability-statusa" name='yu' style="font-size:12px;"></span>
</div>
                <div class="form-group">
                                                <i class="fa fa-lock"></i>
                                                <i class="fa fa-check"></i>
                                        <input type="text" class="form-control" name="confirm_password" placeholder="Confirm
Password" required="required">

</div>
<div class="form-group">
                        <label class="checkbox-inline"><input type="checkbox" required="required"> I accept the <a
href="#">Terms of Use</a> &amp;amp; <a href="#">Privacy Policy</a></label>
                </div>
                <div class="form-group">
  <center> <button type="submit" id='submitdev' name='signupdev' class="btn btn-primary btn-lg">Developer</button>
                        <button type="submit" id='submitrec' name='signuprec' class="btn btn-primary btn-lg">Recruiter
</button></center>
                </div>

</form>
        <div class="text-center">Already have an account? <a href='#myModallog' data-dismiss='modal' data-
toggle='modal'>Login here</a></div>
        <?php
        include('login.php');
        ?>
</div>
</div>
</div>
</div>
</body>
</html>
```

d. File upload

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

  $move = "Uploads/";
  $a = $_FILES['storefile']['name'];
  $b = $move.$a;
```

```
$type = pathinfo($b, PATHINFO_EXTENSION);
  $allowtype = array('gif');
  if(in_array($type,$allowtype))
  {
        echo"<script>alert('Noway it is possible');</script>";
  }
  else
  {
  move_uploaded_file($_FILES['storefile']['tmp_name'], $move.$a);

              $sql = "insert into tbl_file_stored_users (file,regid) values('$a','4')";
   $query = mysqli_query($con,$sql);
   include('adminhome.php');
   echo "<script>$('.txt').load('fileupload.php');</script>";
  }

}
?>
<form method='POST' action='fileupload.php' enctype="multipart/form-data">
 <div class="form-group">
   <label for="exampleFormControlFile1">Example file input</label>
   <input type="file" name='storefile' class="form-control-file"
id="exampleFormControlFile1">
  <button type='submit' name='sb' value='submit'> Upload </button>
 </div>
</form>
```

Details_fetching

```
<script>
function update_stastu(id,val)
  {
        var dev ='company';
        alert(id);
        var id = id;
        var val = val;
        $.ajax
        ({
              type:"POST",
              url:"enable_disable.php",
              data : {id:id,val:val,dev1:dev1},
              success:function(data){
              if(val == "Enabled")
              {
```

```
alert('hari');
                        $('#' + id).attr("class","btn btn-outline-danger");
                        $('#' + id).html('Disabled');
                        $('#' + id).attr("value","Disabled");
                }
                else
                {

$('#' + id).attr("class","btn btn-outline-success");
                        $('#' + id).html('Enabled');
                        $('#' + id).attr("value","Enabled");
                }
                }
        })
  }
</script>

<?php
include('database.php');
?>
  <div class="table-responsive"style="font-size:15px;">
  <table class="table table-hover table-dark table-lg" style="margin-top:5%;" border='1'>
        <caption class="top">List of users
        </caption>
          <thead>
                <tr style='text-align:center;'>
                  <th scope="col">Sl.No</th>
                  <th scope="col">Representative Name</th>
                  <th scope="col">Company Name</th>
                  <th scope="col">Phone</th>
                  <th scope="col">About</th>
                  <th scope="col">Possition</th>
                  <th scope="col">Username</th>
                  <th scope="col">Email-Id</th>
                  <th scope="col">Valid</th>
                  <th scope="col">Stastus</th>
                  <th scope="col">Check</th>
                </tr>
          </thead>
          <tbody>
  <?php
  $sql ="SELECT * from tbl_cmpny_account LIMIT 9";
  $query=mysqli_query($con,$sql);
  $r=mysqli_num_rows($query);
  if($r>0)
  {
```

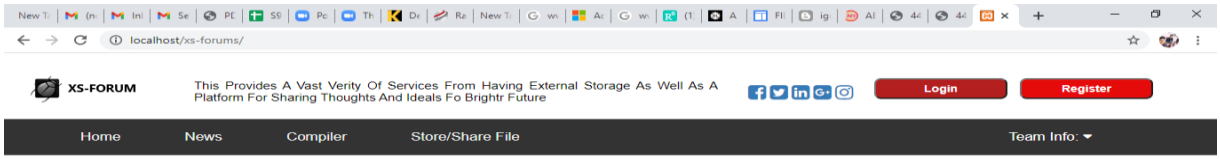
```
while($result = mysqli_fetch_array($query))
        {
                //$_SESSION['d']= $result['acid'];

                ?>
                <tr class='g' style='text-align:center;'>

                 <td ><?php echo $result['compid'];?></td>

 <td  style='text-align:left;'><?php echo $result['repname'];?></td>
                  <td style='text-align:left;'><?php echo $result['cname'];?></td>
                  <td><?php echo $result['phone'];?></td>
                  <td><?php echo $result['about'];?></td>
                   <td><?php echo $result['position'];?></td>
                   <td><?php echo $result['username'];?></td>
                   <td><?php echo $result['email'];?></td>
                   <td><?php echo $result['stastus'];?></td>
                   <td ><?php echo $result['valid'];?></td>
                   <td >

                    <?php
                   if ($result['valid']=='' || $result['valid']== 1 )
                   {
                           ?>
                           <button style="font-size:15px;"  type="button"  id=<?php echo
$result['compid'];?> class="btn btn-outline-success" value="Enabled"
onclick="update_stastu(<?php echo $result['compid'];?>,this.value)">Enabled</button>

                  <?php }
                   else
                   {
                           ?>
                           <button style="font-size:15px;"  type="button"  id=<?php echo
$result['compid'];?> class="btn btn-outline-danger" value="Disabled"
onclick="update_stastu(<?php echo $result['compid'];?>,this.value)">Disabled</button>

                  </td>
                  <?php
                    }
                    ?>



                  </tr>
                 <?php
```

```
}

  }?>

            </tbody>
  </table>
  <div  aria-label="..." class="float-right">
 <ul style="margin-top:-50px; list-style-type: none; " >
   <li class="page-item disabled" style="float: left;">

 <span class="page-link">Previous</span>
   </li>
   <li class="page-item" style="float: left;"><a class="page-link" href="#">1</a></li>


<li class="page-item active" style="float: left;">
     <span class="page-link">
       2
       <span class="sr-only">(current)</span>
     </span>
   </li>
   <li class="page-item" style="float: left;"><a class="page-link" href="#">3</a></li>
   <li class="page-item" style="float: left;">
     <a class="page-link" href="#">Next</a>
   </li>
 </ul>
</div>
</div>
```

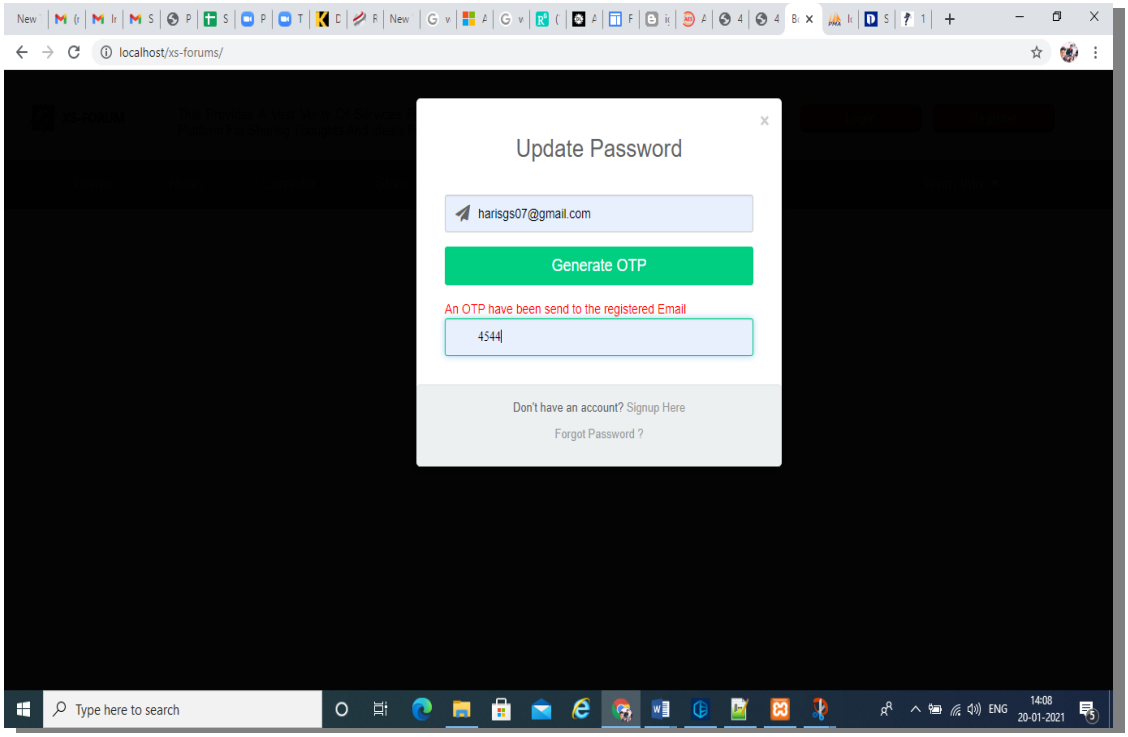
9.2 Screen Shots

CUSTOMER PAGES

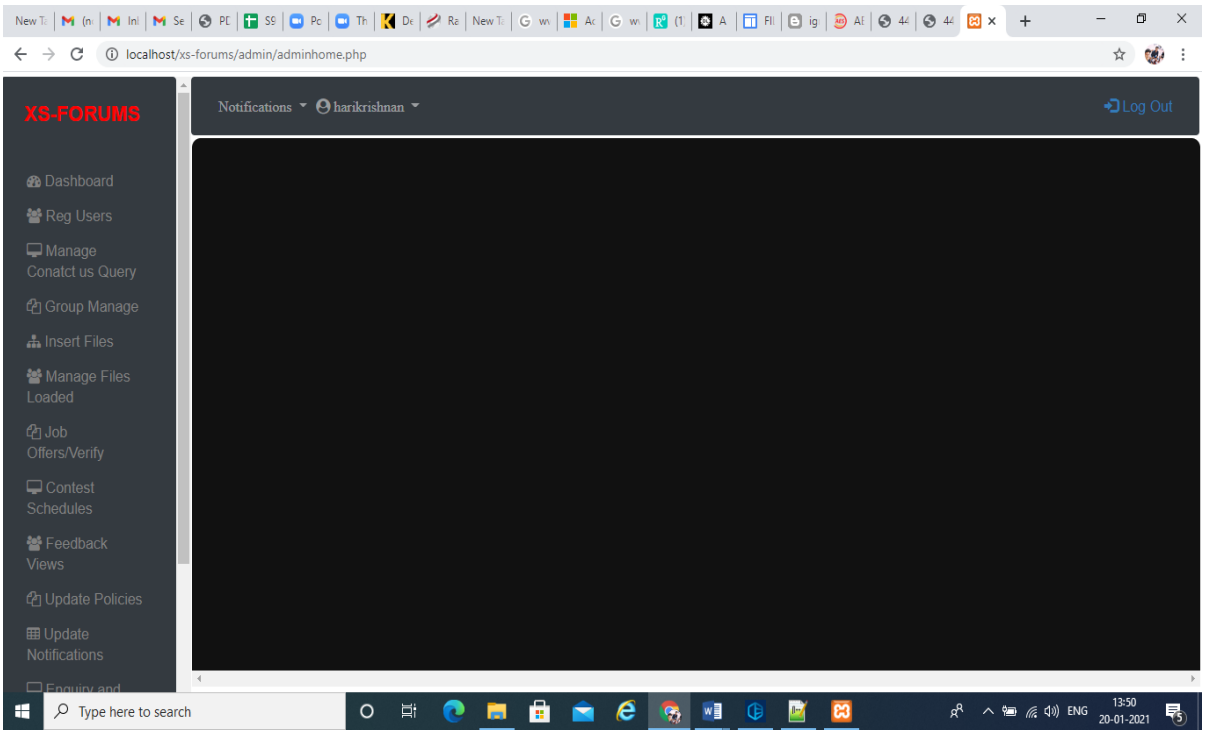
Customer Home page



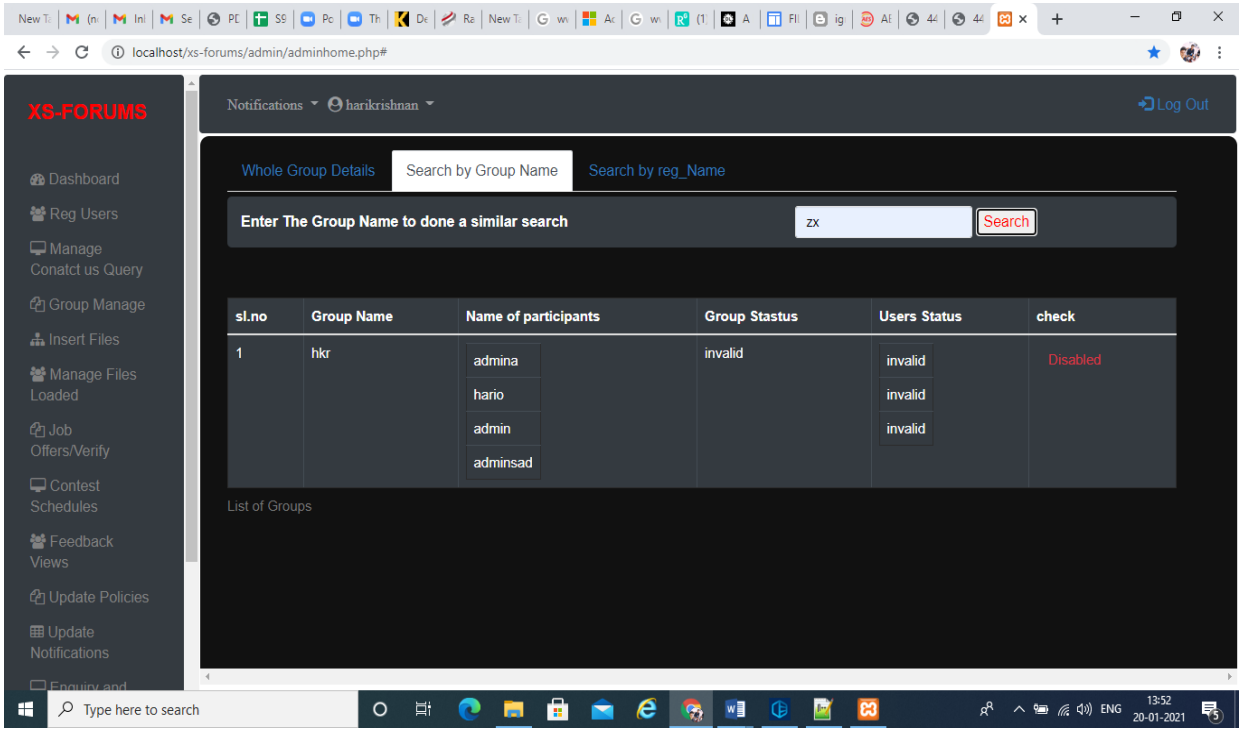
Forgot password



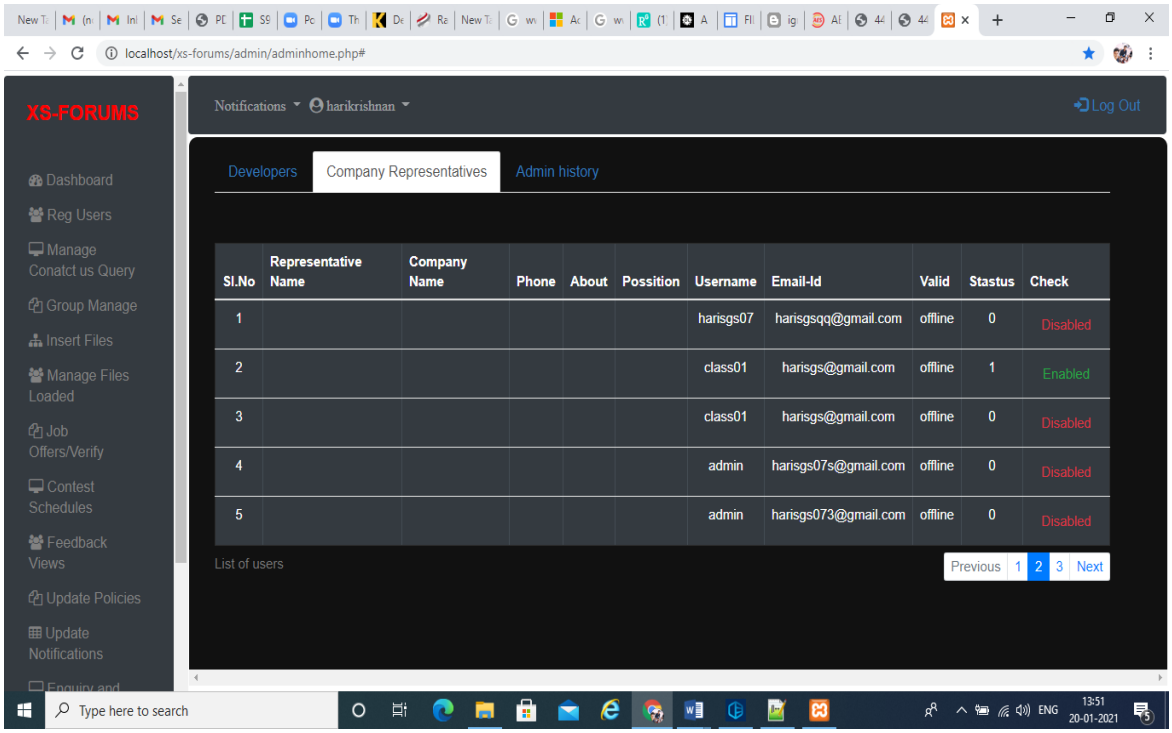
Admin Dashboard page



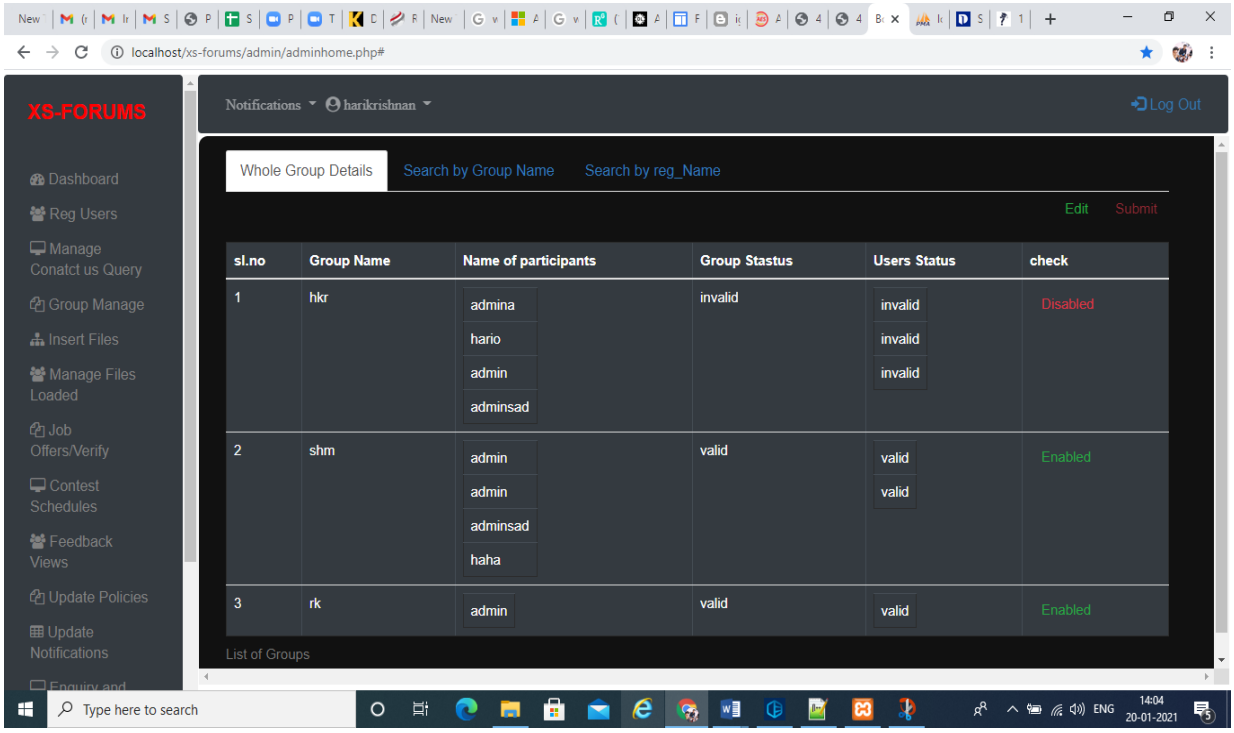
Search group



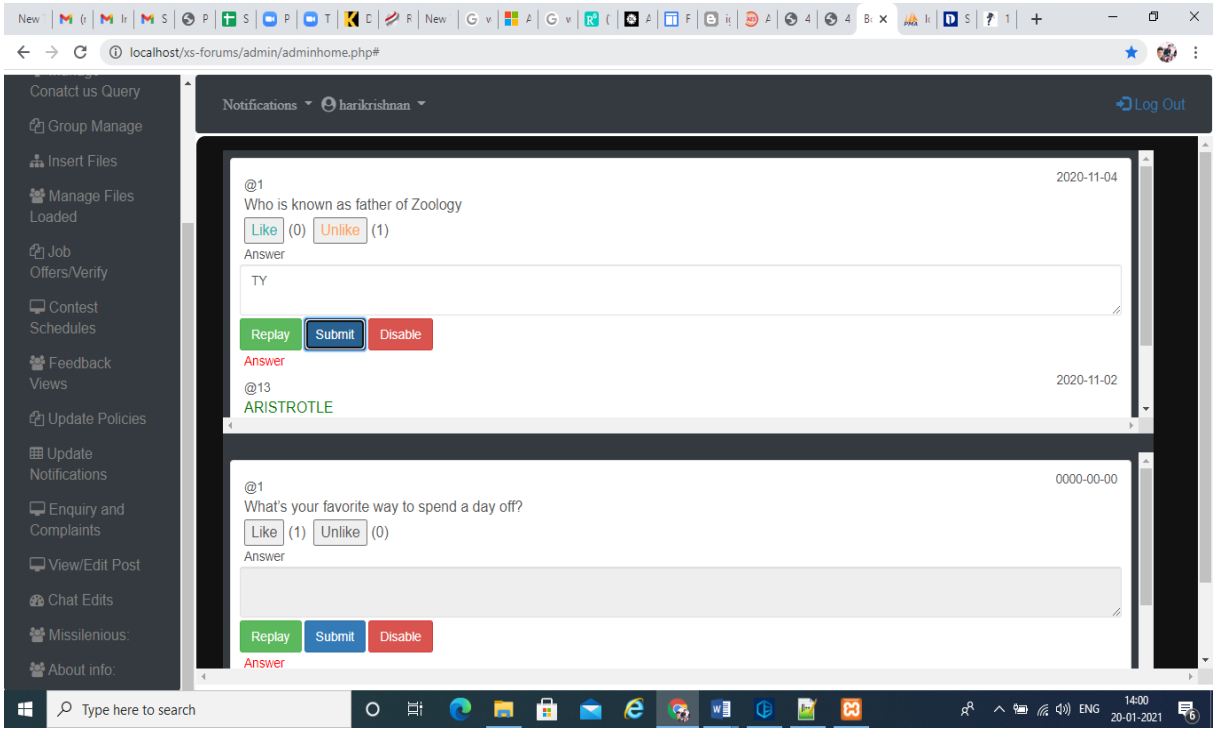
View all registered customers



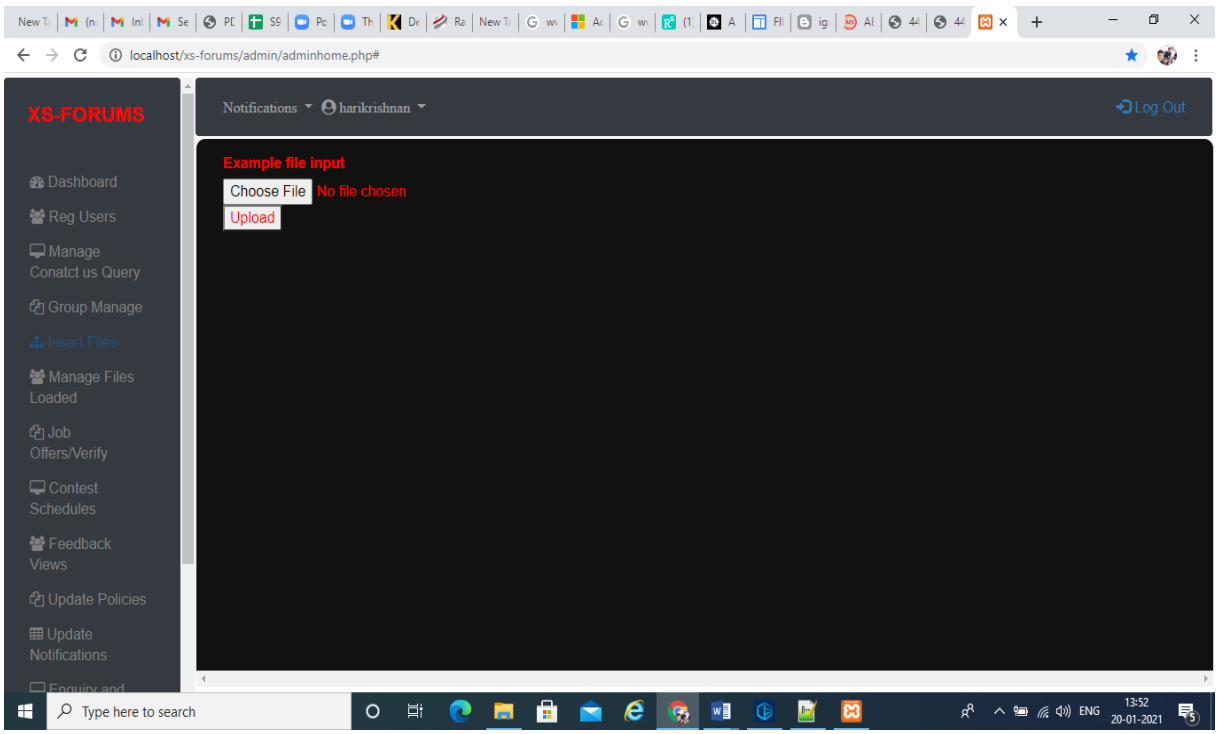
View all registered group details



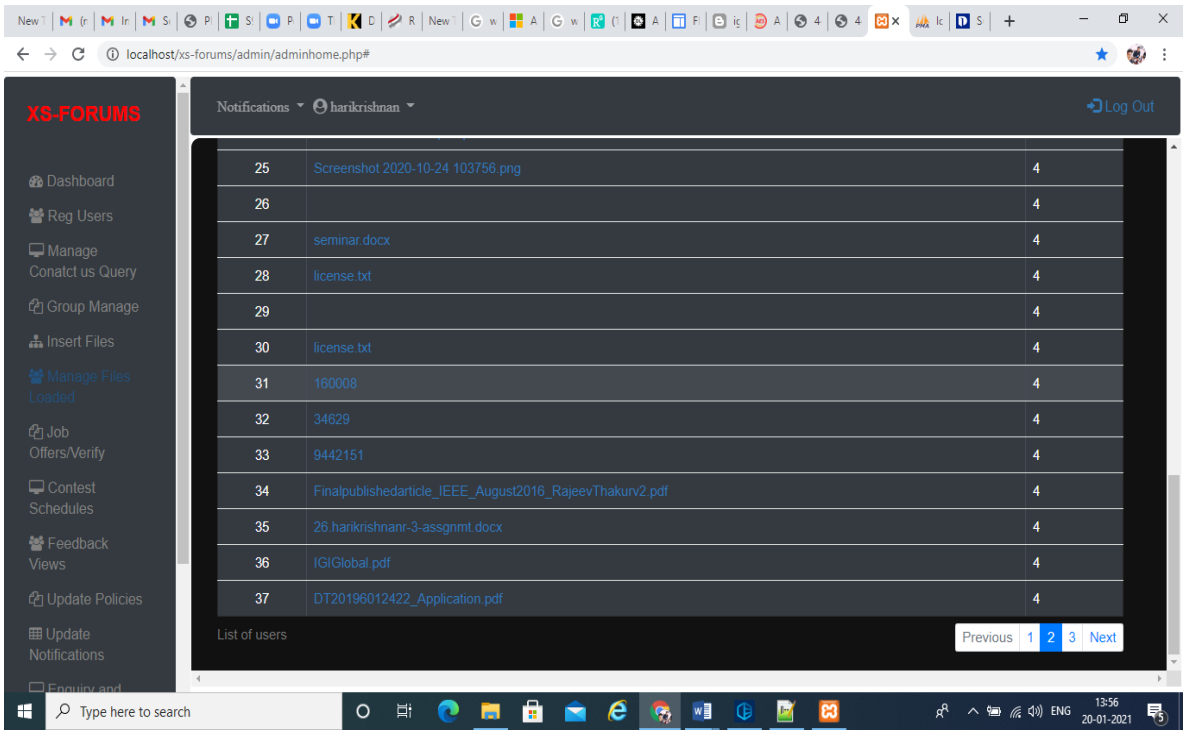
View/edit post



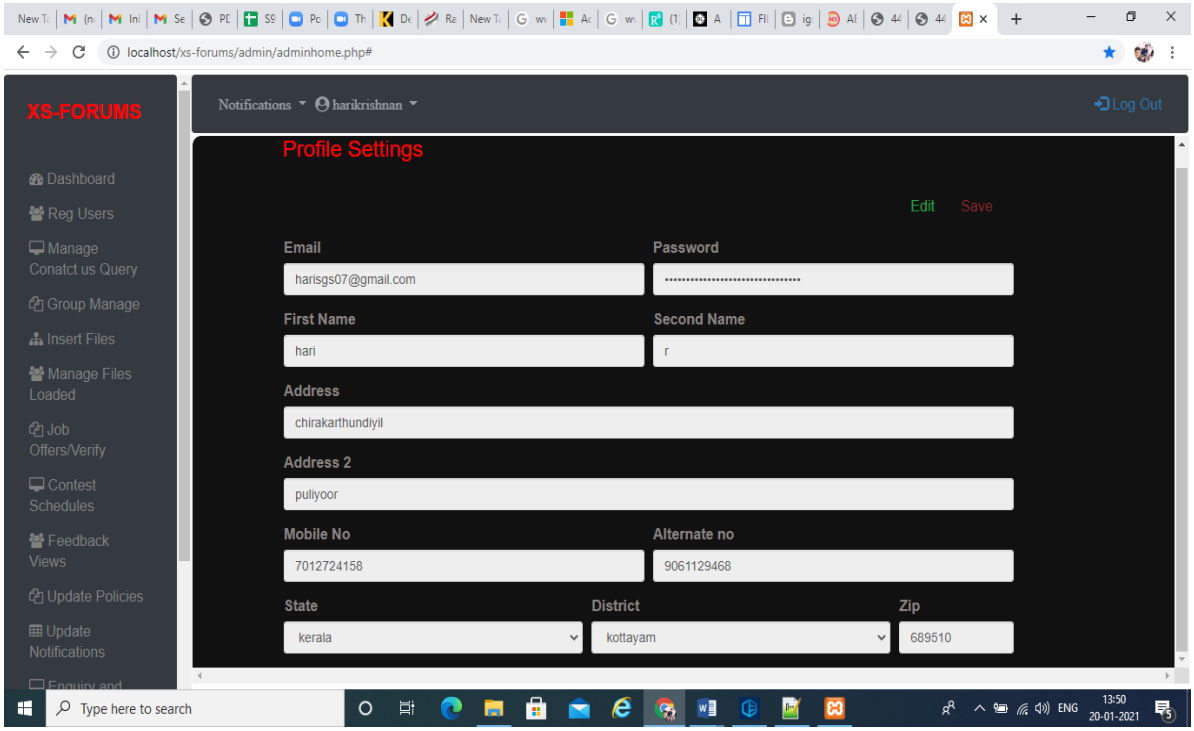
Insert Files



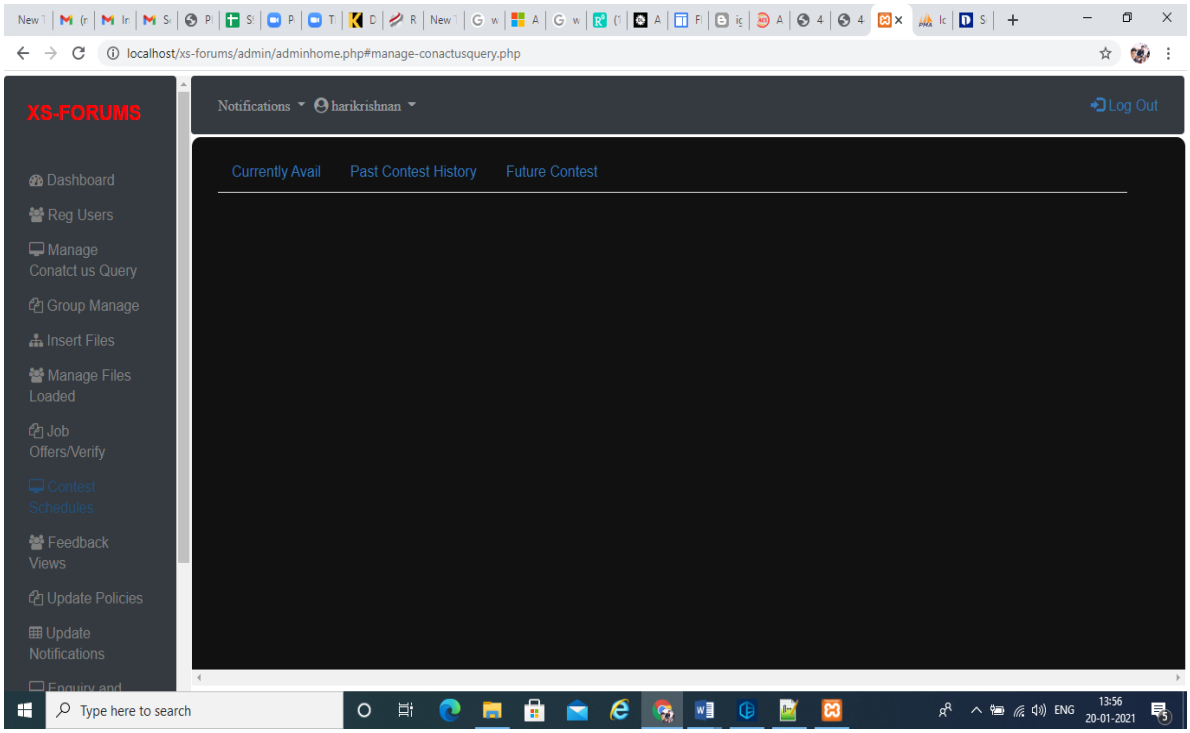
Manage files



Profile update



Contest details



Feed back