1. INTRODUCTION

1.1 INTRODUCTION

Intra-College Communication System is a software which has been developed to override the problems prevailing in the practicing manual system. This software is to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it provides it is user-friendly. Intra-College Communication System, as described above, can lead to error free, secure, reliable and fast networking system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. These systems will ultimately allow you to better manage resources.

1.2. PURPOSE

The main objective of this document is to illustrate the requirements of the project "Intra-College Communication System". The document gives the detailed description of both functional and non-functional requirements proposed by the client. The purpose of this project is to build a connectivity between students and faculties to reduce the manual work for conveying any information regarding to the college. It tracks all the details about the Posts, Shares and Photos etc

.

1.3. OBJECTIVE

- To gather all the college members, starting from administrative level to the basic clerical section into a single frame.
- To convey all the notice(s), event(s), corrigendum(s) etc. and moreover what is going around the college.
- To grow the teacher and student relationship stronger.
- To provide 24 X 7 connectivity with college campus.

2. SYSTEM STUDY

2.1. EXISTING SYSTEM

- Reduce manual work, every institution, whether big or small, has challenges to overcome and managing the information's of friends, users, shares, videos, photos.
- The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required take the database backup

2.2. PROPOSED SYSTEM

- The proposed system works based on a few conditions. We are expecting the colleges to have a domain registered and hand out emails based on that domains which helps reduce the manual work required to verify every student or a personal logging in to the system.
- In the case of a college where they are not able provide the emails, we have an admin page designed which can be used to manually check if a user can or cannot use the platform.
- The verification process is very much necessary as we don't want intruders from some other college to disrupt any kind of workflow happening on the network.
- If the college has a proper digital record of all the students studying at the college, we can also setup an One Time Password based login from the registry of the college, which again is the student's personal responsibility to notify the college when there is any change in such phone number.

2.3. FEASIBILITY STUDY

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological and operational. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it .During the stage of our feasibility study, we had to under go the following

steps as described under:

Identify the origin of data at different levels of the system.

- Identify the expectation of end user from the finished product/system.
- Analyze the drawback(s) of the existing system.
- Technical feasibility study: It lays out details on how a good or service will be delivered, which includes transportation, business location, technology needed, materials and labor.
- Financial feasibility study: It is a projection of the amount of funding or startup capital needed, what sources of capital can and will be used and what kind of return can be expected on the investment.
- Organizational feasibility study: It is a definition of the corporate and legal structure of the business; this may include information about the founders, their professional background and the skills they possess necessary to get the company off the ground and keep it operational.

3. HARDWARE AND SOFTWARE REQUIRMENTS

3.1. Hardware Requirements

Processor	AMD Phenom II X2 550 Processor 3.10 Ghz
Memory	4.0 GB
Hard disk space	1 TB
Monitor	21.5 Samsung sync master 2233sw
Others	Other required standard computer peripherals, such as keyboard and mouse

Table no 3.1.1

3.2. Software Requirements

Server Environment	
Operating System	Microsoft Windows 7
Programming tool	Adobe Dreamweaver CS4
Database Server	MySQL database
Web server	Apache 2.2.11
Client Environment	
Internet browser	Chrome, Firefox…

Table no 3.2.1

4. SOFTWARE REQUIREMENT SPECIFICATION

4.1. INTRODUCTION

Software Requirement Specification (SRS) is essential information, which shapes the establishment of the software development process. SRS records the necessities of a framework as well as has a depiction of its significant components.

The focus in this stage is one of the users of the system and not the system solutions. The result of the requirement specification document states the intention of the software, properties and constraints of the desired system. SRS constitutes the understanding amongst customers and designers with respect to the substance of the product that will be created. SRS should be precise and totally signify the framework prerequisites as it makes a colossal commitme2nt to the general development plan.

One of the most essential information is SRS (Software Requirement Specification). It gives the detailed information about establishment of software development process. It records the important necessities of the frame work also holds the depiction of the important components. These things will be in the IEEE standards. The recommendations would shape the explanation behind giving clear image of the item to be made filling in as measure for execution of an understanding among client and the developer. One of the important steps involved in the development process is system requirements. This SRS (Software Requirement Specification) is followed after resource analysis phase. Its main task is to decide what a software product does. In this stage the main focus is the user, and not the system solution. SRS (Software Requirement Specification) gives the results like intention of the software, properties and constraints of the desired system. The main advantage of SRS (Software Requirement Specification) is that it gives a clear understanding among the clients and the product developers with respect to the product that is developed. SRS (Software Requirement Specification) which is documented should accurate and the prerequisites of the frame work should be signified as it makes colossal commitment to the general development plan process.

4.1.1 NEED FOR COMPUTARIZATION

Social media has changed the way we all interact with each other online. It gives us the ability to discover what's happening in the world in real-time, to connect with each other and stay in touch with long-distance friends, and in order to have access to endless amounts of information at our fingertips. In many senses, social media has helped many individuals find common ground with others online, making the world seem more approachable.

According to a survey by Pew Research Center, the use of social media is correlated with having more friends and more diverse personal networks, especially within emerging economies. For many teenagers, friendships can start virtually, with 57% of teens meeting a friend online.

Businesses are also using social media marketing to target their consumers right on their phones and computers, building a following in order to build a loyal fan base, and create a culture behind their own brand. Some companies, such as Denny's, have created entire personas on Twitter in order to market to younger consumers using their own language and personas

4.1.2 PURPOSE

The Web-based social networking services make it possible to connect people who share interests and activities across political, economic, and geographic borders. Through e-mail and instant messaging, online communities are created where a gift economy and reciprocal altruism are encouraged through cooperation. Information is suited to a gift economy, as information is a non rival good and can be gifted at practically no cost.

Facebook and other social networking tools are increasingly the object of scholarlyyTesearch. Scholars in many fields have begun to investigate the impact of social- networking sites, investigating how Such sites may play into issues of identity, privacy, social capital, youth culture, and education.

Several websites are beginning to tap into the power of the social networking model for philanthropy. Such models provide a means for connecting otherwise fragmented industries and small organizations without the resources to reach a broader audience with interested users. Social networks are providing a different way for individuals to communicate digitally.

4.1. 3 SCOPE

The tremendous popularity of using the social media networking could never have been realized before. In fact, social media have become an important tool of marketing in true sense of customer orientation. But this kind of social networking site which can access only for college campus not from outside the college will make a dramatic changes inside the college campus.

- The scope of Intra-College Communication System is widening and today it offers a strong support to the college campus in providing the much desired touch of concern.
- The bright future prospect of college networking is also proven with the fact that the technology is integrated in mobile phones as well.
- The project is developing with the power of interpersonal communication on a globalized outlook.

4.2 TOOLS AND TECHNOLOGIES USED

The system is designed on the popular MERN stack

- M: Mongo DB – the database widely popular for being non-relational based database that stores data in form of documents rather than a traditional table format. MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License which is deemed non-free by several distributions.
- E: Express JS – Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node,js, released as free and open source software under the MIT License, It is designed for building web applications and APIs. It has been called de facto standard server framework for Node.js.

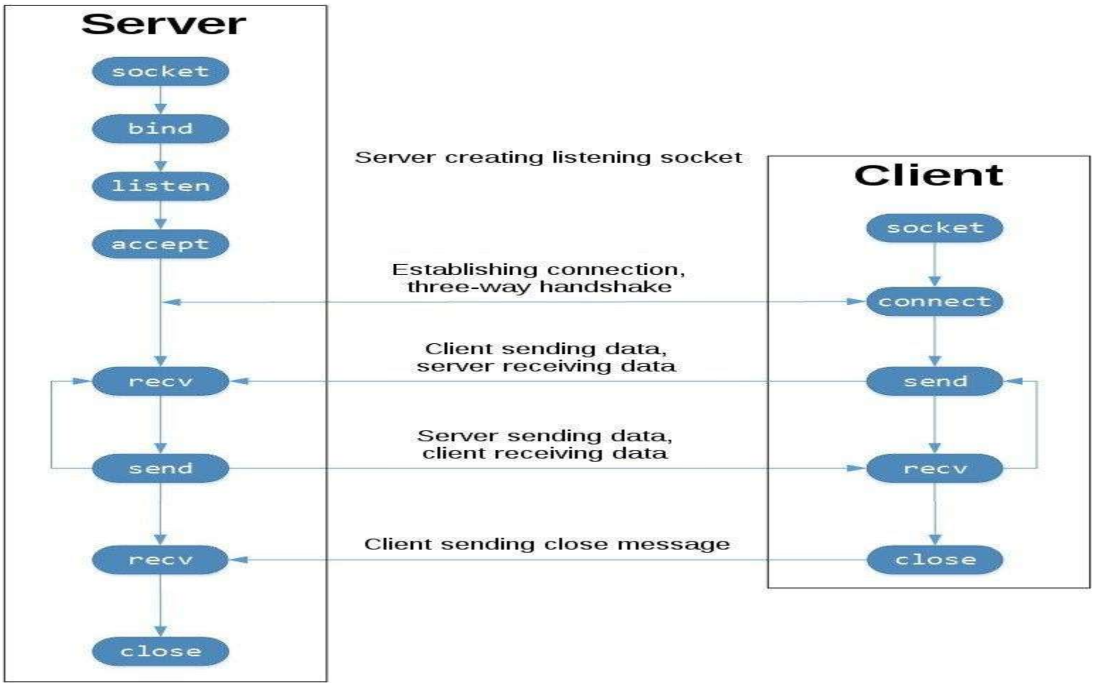
- R: React JS – React is a free and open-source front end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and several other companies.
- N: Node JS – Node,js is a cross platform, open source server environment that can run on Windows, Linux, Unix, macOS, and many more. Node.js is a back-end Java Script runtime environment, runs on the V8 Java Script Engine, and executes Java Script code outside a web browser.

Other tools that were used while development

- HTML5: Hyper Text Markup Language is a very popular standard markup language for documents designed to be displayed on a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as Java Script or Type Script.
- CSS3: Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XMLCSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
- Git: Git is a distributed version control system: tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows
- Docker: Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called *containers*. The service has both free and premium tiers. The software that hosts the containers is called Docker Engine. It was first started in 2013 and is developed by Docker, Inc.

4.2.1 SOCKET PROGRAMMNG

- Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server.

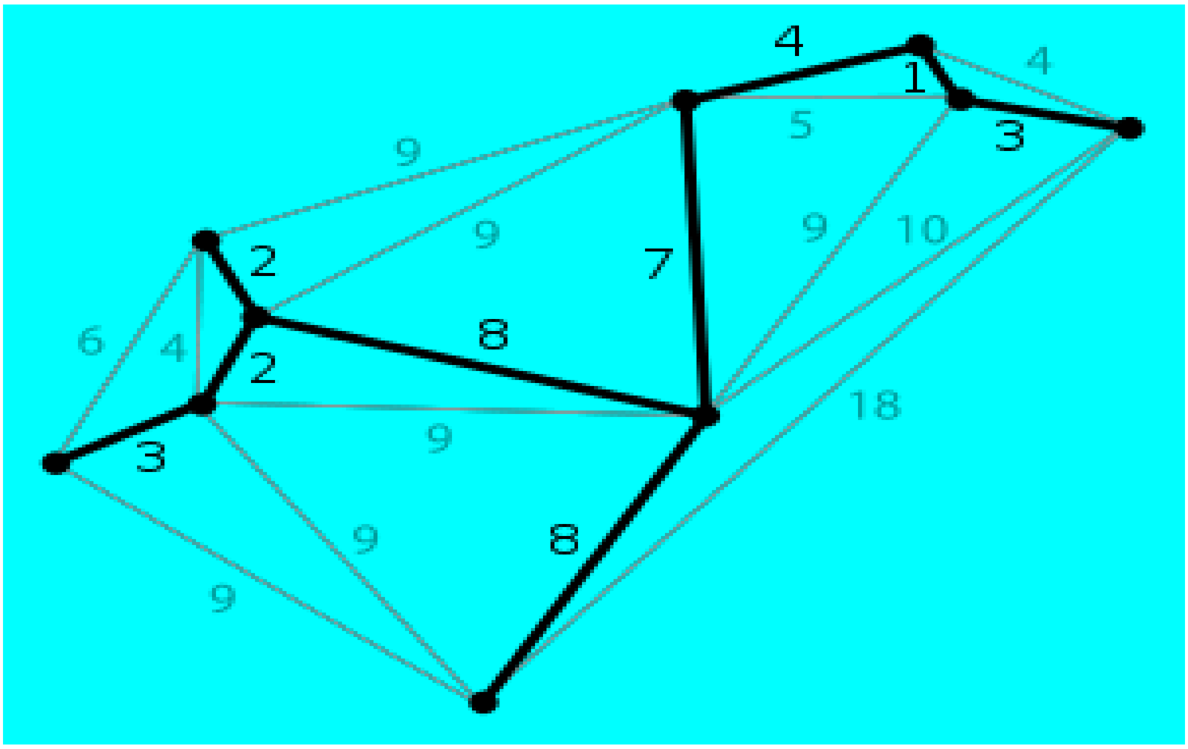

Server
socket
bind
listen
accept
recv
send
recv
close
Server creating listening socket
Establishing connection, three-way handshake
Client sending data, server receiving data
Server sending data, client receiving data
Client sending close message
Client
socket
connect
send
recv
close

- Transmission Control Protocol (TCP) Socket Flow

Figure no 4.2.1

4.2.2 SPAN TREE

- A single graph can have many different spanning trees.
- A spanning tree is defined as a tree which is a subset of the graph that have the same vertices as graph and edges same as a graph, but one less edge than the given graph makes the graph a spanning tree where all the vertices are covered with one less than edges of the given graph which makes it cycle free graph. In general, we can define a spanning tree as a tree that does not have any cycles, and the given graph can never be a disconnected graph as every connected and undirected graph can have at least one spanning tree that holds an equal number of vertices as a graph and edges one less than the given graph.



Span tree image

Figure no 4.2.2

4.3 FUNCTIONAL REQUIRMENTS

- Authentication: Authentication allows specific user to perform some tasks or use the application. Since this is an intra college system, the system should be thorough if some user is a member of the college or not
- Authorization levels: the different levels make sure that there is some kind of governing entity on the network. Different levels of personals can be given different responsibilities like verifying a student's background, making sure the posts do not carry any kind of elements that might cause imbalance in the college being functional without any troubles and so on.
- Reporting: Create frequent reports on things like number of users, bugs, crashes, user interests, upgrades and many more which helps in better run time of the system and also to make it more user friendly and promote the system among the students.

- Compliance to laws and Regulations

4.4 NON-FUNCTIONAL REQUIRMENTS

- Performance Requirements: The system should uphold the end-client's necessities. Thesystem is fit for preparing when huge quantities of documents are given as info and furthermore it should be intuitive and the deferrals included ought to be less. So, in each activity reaction of the framework, there are no immediate delays.
- Safety and Security Requirements: The framework should be planned in a protected manner by applying security measures. Data transmission should be safely sent to hubs with no changes in data. A unique special case dealing with a mechanism should be set up to keep away from the framework errors.
- Flexibility: If there is a slight modification to the business rationale, none of us need to introduce the whole framework in singular client's PCs.
- Reusability: For the show layer, the business rationale's reusability is more obvious. As this section has been created and evaluated, we may apply it to other projects and it will be beneficial down the road.
- Security: Since the client cannot access the database directly, the architecture is more secure.

5. SYSTEM DESIGN

5.1 USECASE DIAGRAM

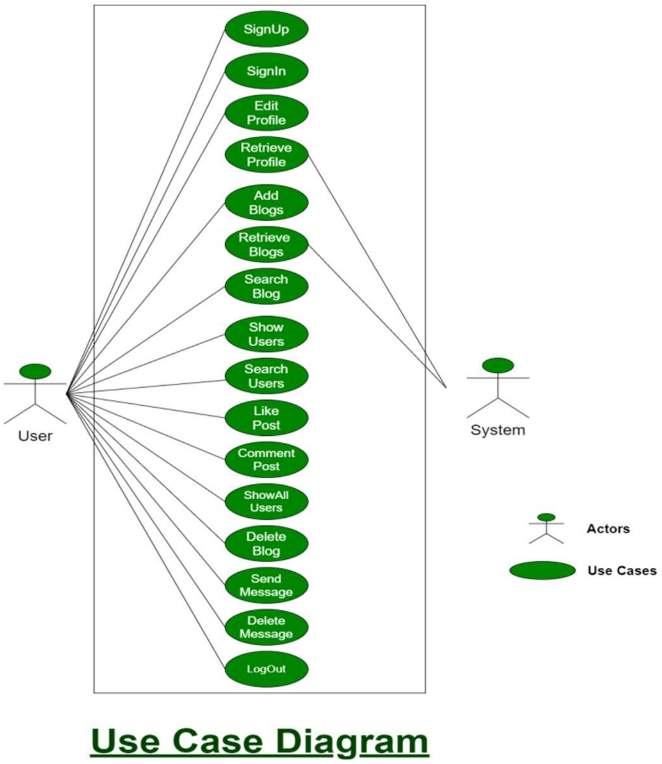


Figure no : 5.1.1

5.2 CLASS DIAGRAM

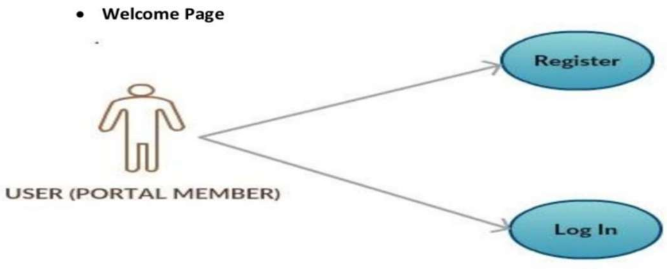


Figure no : 5.2.1

- After Login Page

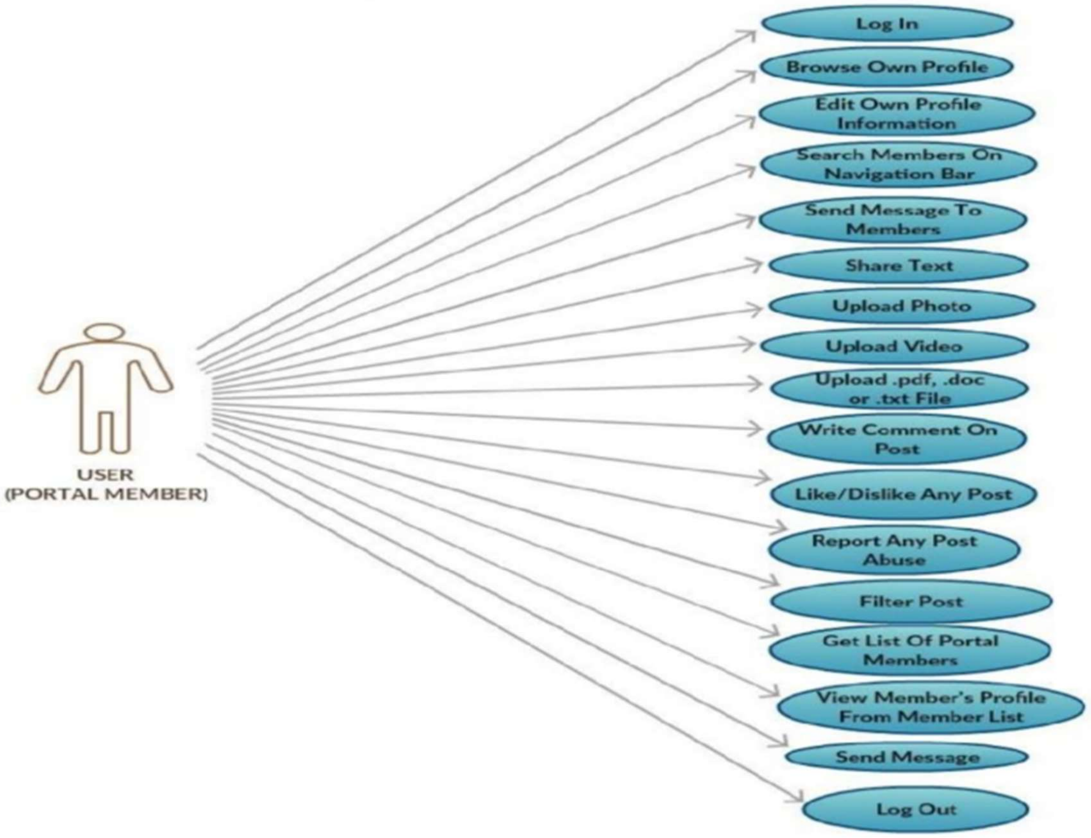


Figure no : 5.2.2

- Admin

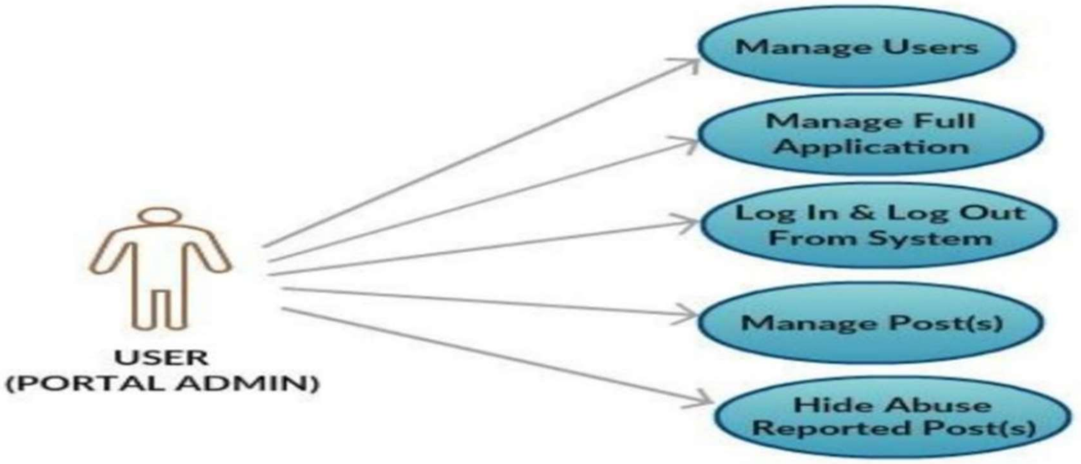


Figure no : 5.2.3

5.3 SEQUENCE DAIGRAM

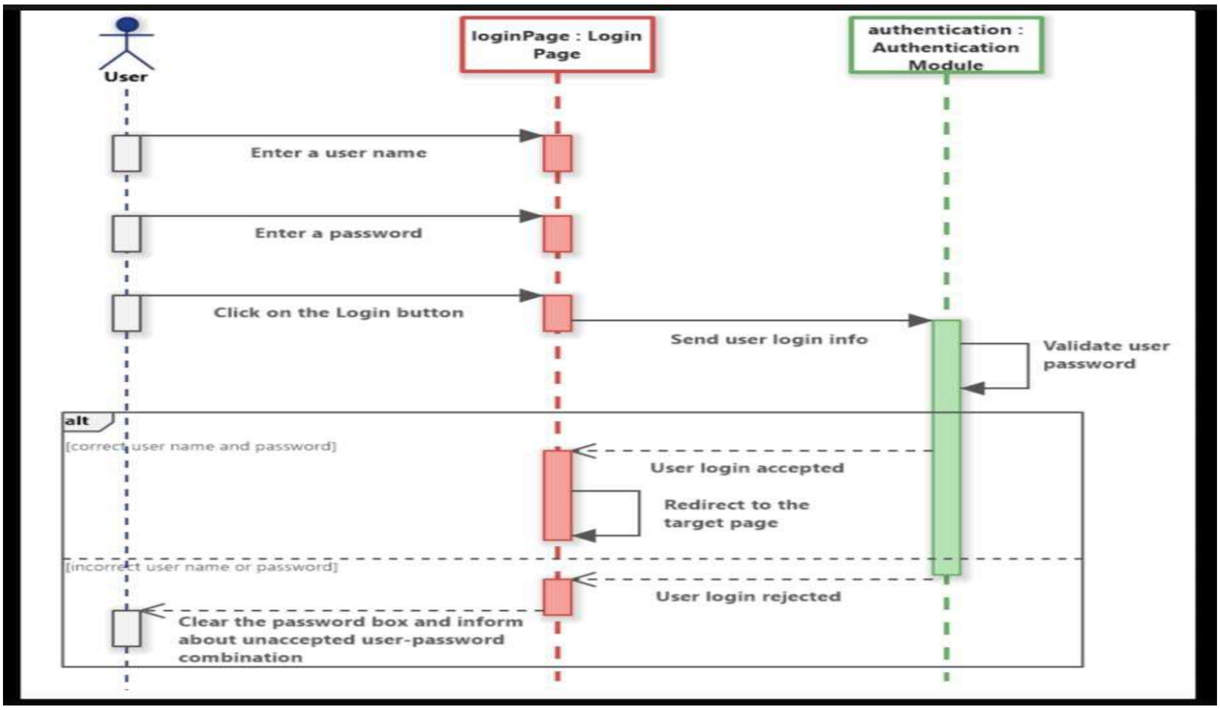

User
loginPage : Login Page
authentication : Authentication Module
Enter a user name
Enter a password
Click on the Login button
Send user login info
Validate user password
alt
[correct user name and password]
User login accepted
Redirect to the target page
[incorrect user name or password]
User login rejected
Clear the password box and inform about unaccepted user-password combination

Figure no : 5.3.1

5.4 ACTIVITY DAIGRAM

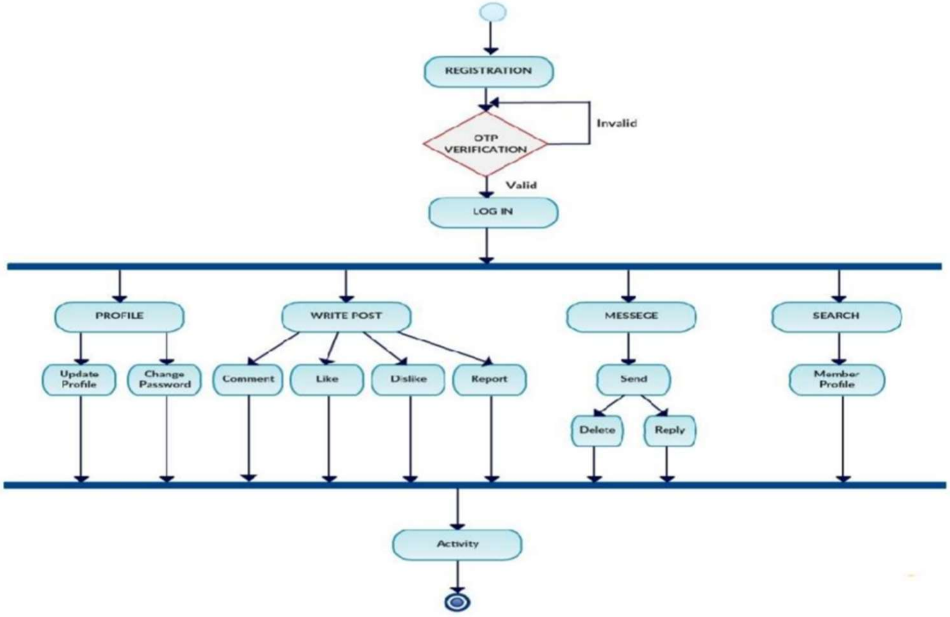


Figure no : 5.4.1

5.5 DATAFLOW DAIGRAM

- level 0

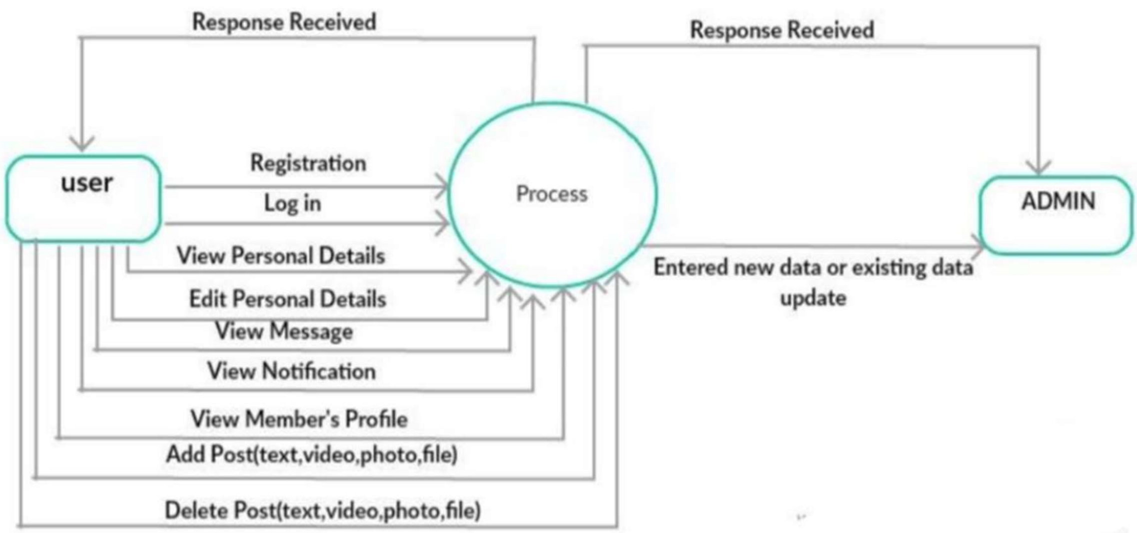


Figure no : 5.5.1

- level 1

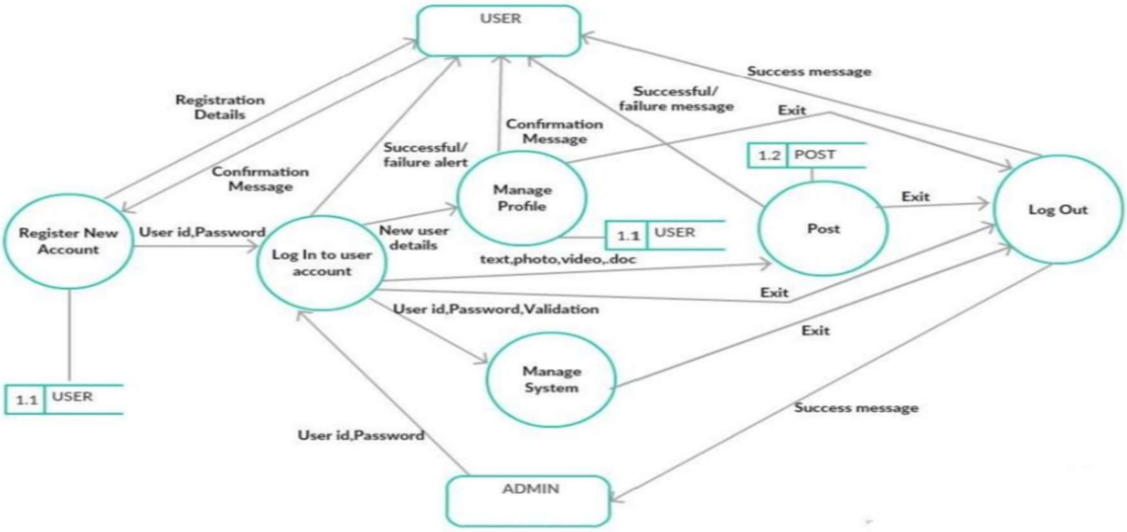


Figure no : 5.5.2

- level 2

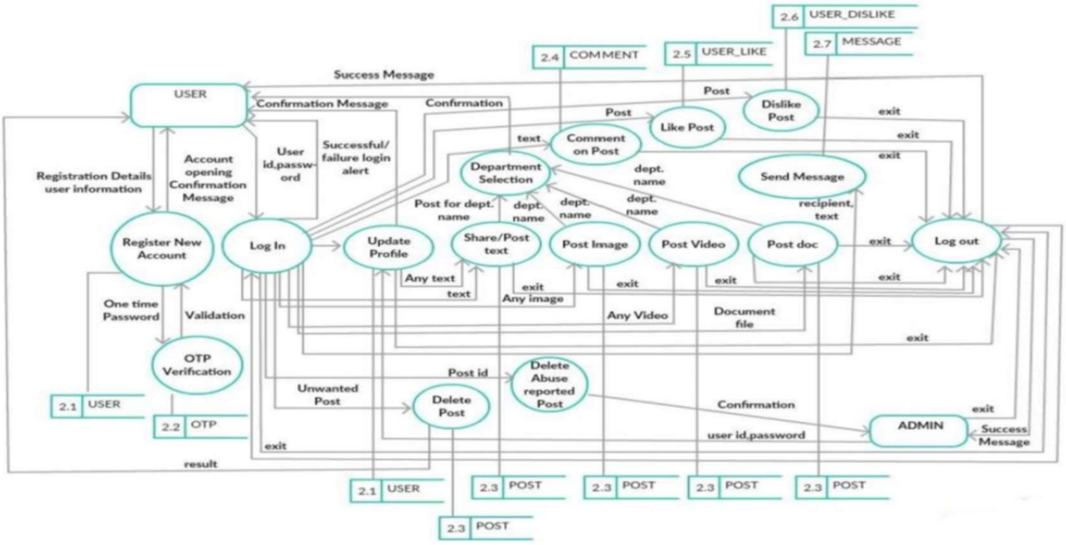


Figure no : 5.5.3

5.6 ER DAIGRAM

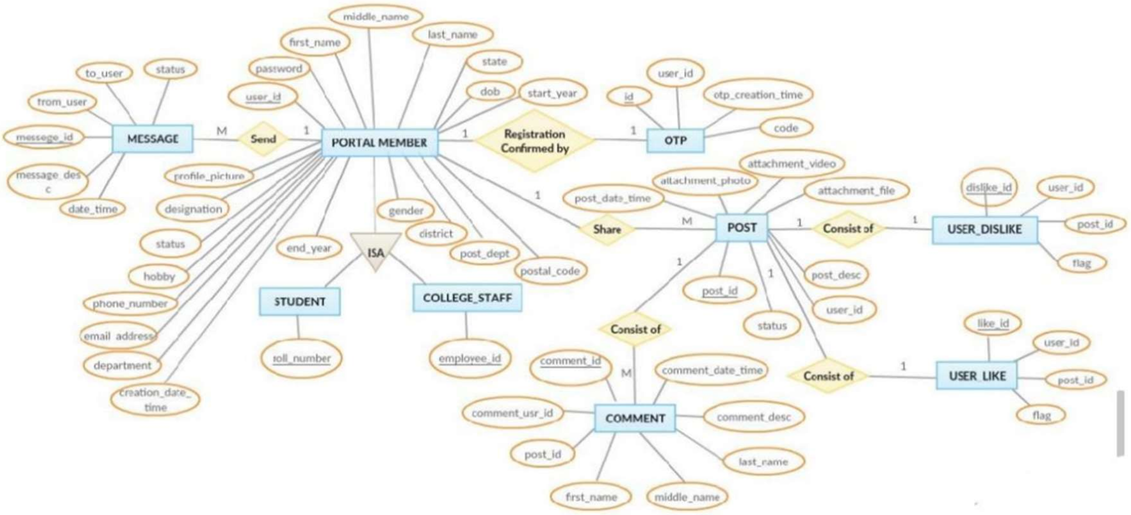


Figure no : 5.6.1

6. IMPLEMENTATION

6.1. MODULES

6.1.1. SERVER

The server class acts as a wrapper for all server functions for our social networking site. It essentially act as a link between all of the information such as accounts, account details, pages notes, etc to our database. When any other model object such as a page is pulled from the server, a temporary copy is made, If that temporary copy is changed in any way, the new version must be sent to the server in order to update the permanent copy.

6.1.2. ACCOUNT

Each user who wants to use the site must create an account. This is the head class that all other objects use to determine what a user does and when the user did it. The account's information has four purposes: hold the login information, hold friend information, hold profile information, and hold privacy information with such a large amount of information to keep track of, the Account class would be very large and difficult to work with. Therefore to ease the load, the Account class was broken up into three different classes. There is the actual account class which keeps track of login information and friend information. It also holds the other two classes within it. Profile information was outsourced to the Account Details class, and privacy settings were outsourced to the Privacy Settings class. The only time the Account class needs to be updated is when the user changes his/her username and/or password. All other settings are handled by the Account Details and Privacy Settings classes.

6.1.3. POST

A Wall Post object works very similar to a Message object. The only difference is that the Wall Post objects are viewable to all friends of the user. Another difference is that a Wall Post object can hold comments. Ifa user adds a wall post to his or her own account, their status will change to the new wall post.

6.1.4. GROUPS/FRIENDS

The most important feature is being able to add and remove friends. In our Social Networking site, making friends is a fairly straightforward process. Users can type in the name of a friend in the search bar at the top of their home page. The database is queried for an account that has the search term contained in the full name, any media files with the search terms in the description, any pages with the search terms in the titles, any links with the search terms in the titles, and any notes with the search terms in the title. For example, User A could search for User B in the search bar. After clicking on User B's profile, User A will see a button that says Send Friend Request. Clicking on it will send an alert to User B that User A wants to be a friend. The friend request will now be in the friends list of User B, where he/she can either accept it or ignore it, letting it sit there indefinitely. If user B accepts the request, User A will be added to User B°s friends list and vice versa. Being friends has its advantages. For example, only friends can chat to each other. Also, friends can view any part of a profile that is marked as friends only. Finally, for a user to view his/her friends and incoming friend requests, he/she just clicks on the friends tab which brings up a frame.

6.1.5. ACCOUNT CREATION

When a user accesses the site for the first time, he/she must create an account before using any of the site features. The account creation process is broken into three sections. The first section deals with the login information and is required for the user to fill out. This includes the email, password, and password confirmation. The purpose behind the password confirmation is to ensure that the user didn't accidentally mistype when creating a password. The second section deals with information about who you are such as name, location, and gender. Most of these fields are optional except for your name and gender. It wouldn't be much of a social network if everyone was named anonymous. The final section deals with information about the users likes and dislikes, such as interests and activities. Unlike the other two sections, this section is completely optional. Once the user clicks create account, a new account, account details, and privacy settings are added to the server, and the user is brought back to the login page.

6.2. DATABASE TABLES

- Database: College_campus

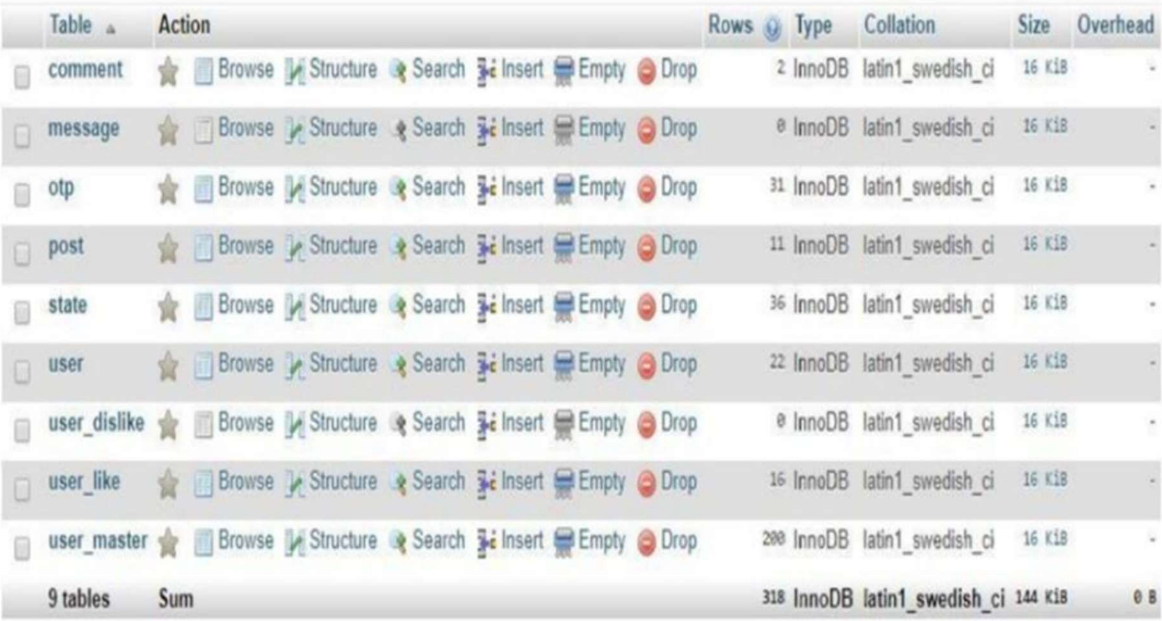
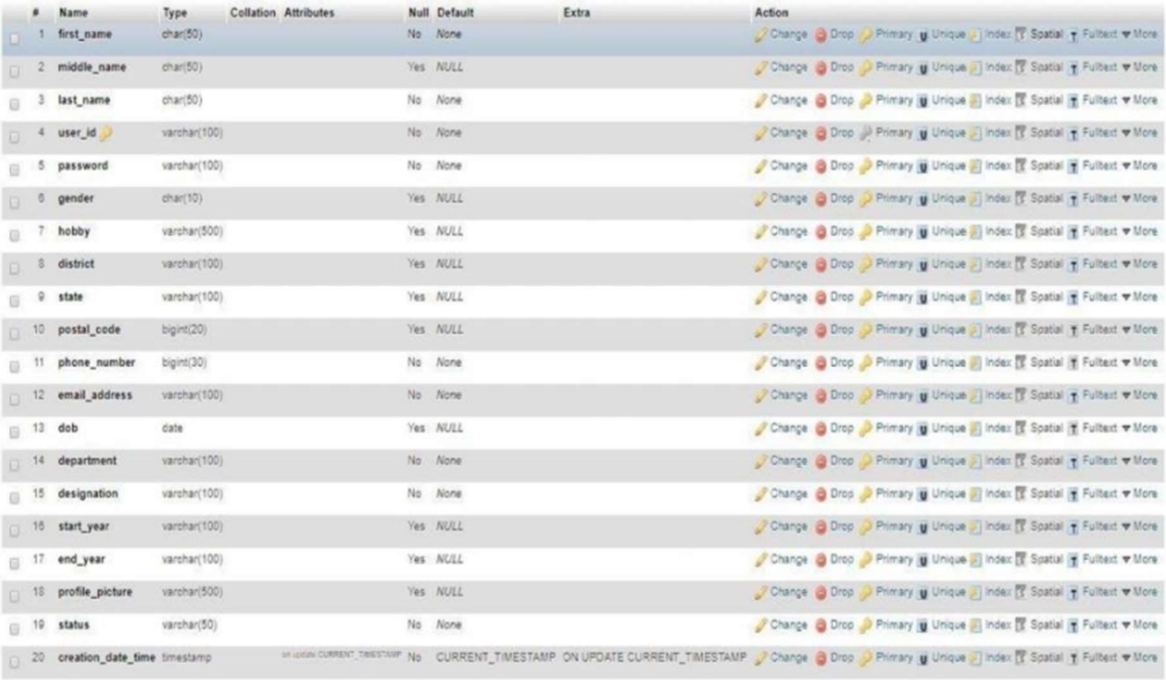


	Table	Action	Rows	Type	Collation	Size	Overhead
□	comment	☆ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
□	message	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
□	otp	☆ Browse Structure Search Insert Empty Drop	31	InnoDB	latin1_swedish_ci	16 KiB	-
□	post	☆ Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16 KiB	-
□	state	☆ Browse Structure Search Insert Empty Drop	36	InnoDB	latin1_swedish_ci	16 KiB	-
□	user	☆ Browse Structure Search Insert Empty Drop	22	InnoDB	latin1_swedish_ci	16 KiB	-
□	user_dislike	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
□	user_like	☆ Browse Structure Search Insert Empty Drop	16	InnoDB	latin1_swedish_ci	16 KiB	-
□	user_master	☆ Browse Structure Search Insert Empty Drop	200	InnoDB	latin1_swedish_ci	16 KiB	-
	9 tables	Sum	318	InnoDB	latin1_swedish_ci	144 KiB	0 B

Table no: 6.2.1

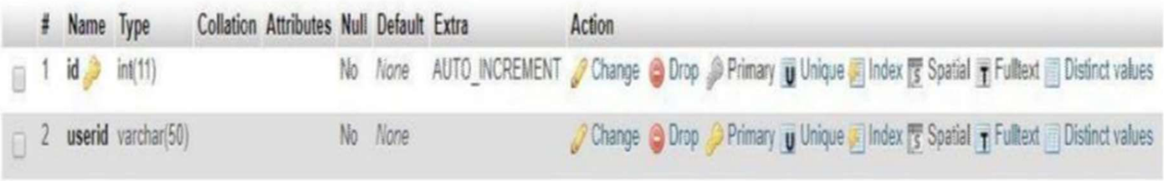
1. User



	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
□	1	first_name	char(50)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext More
□	2	middle_name	char(50)			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	3	last_name	char(50)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext More
□	4	user_id	varchar(100)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext More
□	5	password	varchar(100)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext More
□	6	gender	char(10)			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	7	hobby	varchar(500)			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	8	district	varchar(100)			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	9	state	varchar(100)			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	10	postal_code	bigint(20)			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	11	phone_number	bigint(30)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext More
□	12	email_address	varchar(100)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext More
□	13	dob	date			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	14	department	varchar(100)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext More
□	15	designation	varchar(100)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext More
□	16	start_year	varchar(100)			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	17	end_year	varchar(100)			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	18	profile_picture	varchar(500)			Yes	*NULL*		Change Drop Primary Unique Index Spatial Fulltext More
□	19	status	varchar(50)			No	*None*		Change Drop Primary Unique Index Spatial Fulltext More
□	20	creation_date_time	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Change Drop Primary Unique Index Spatial Fulltext More

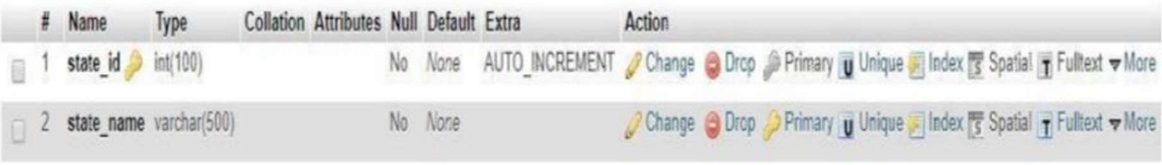
Table no: 6.2.2

2. user_master



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	userid	varchar(50)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

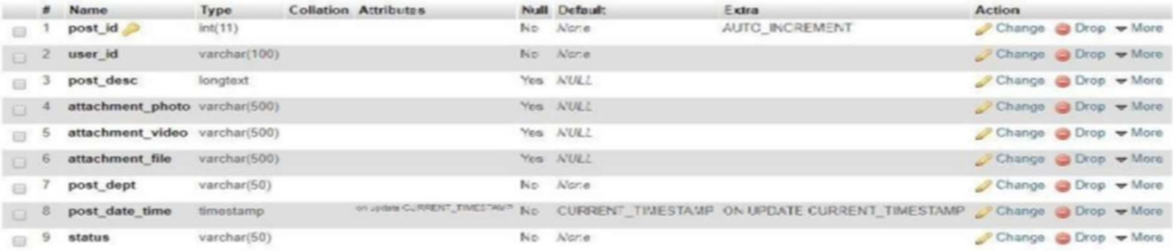
3. state



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	state_id	int(100)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext More
2	state_name	varchar(500)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

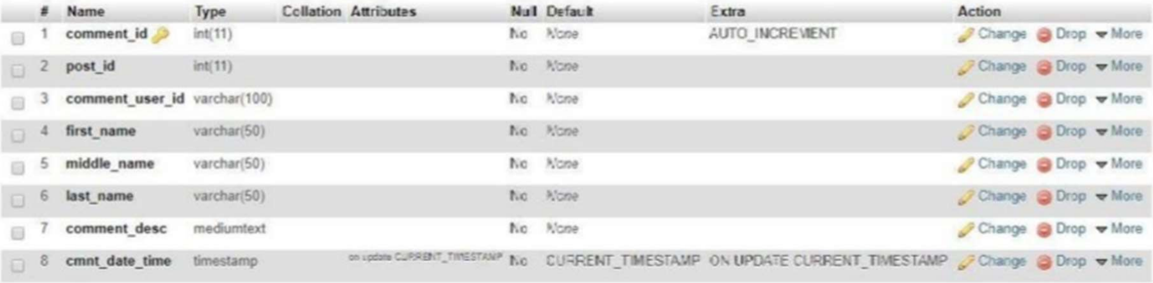
Table no: 6.2.3 and 6.2.4

4. post



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	post_id	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	user_id	varchar(100)			No	None		Change Drop More
3	post_desc	longtext			Yes	NULL		Change Drop More
4	attachment_photo	varchar(500)			Yes	NULL		Change Drop More
5	attachment_video	varchar(500)			Yes	NULL		Change Drop More
6	attachment_file	varchar(500)			Yes	NULL		Change Drop More
7	post_dept	varchar(50)			No	None		Change Drop More
8	post_date_time	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Change Drop More
9	status	varchar(50)			No	None		Change Drop More

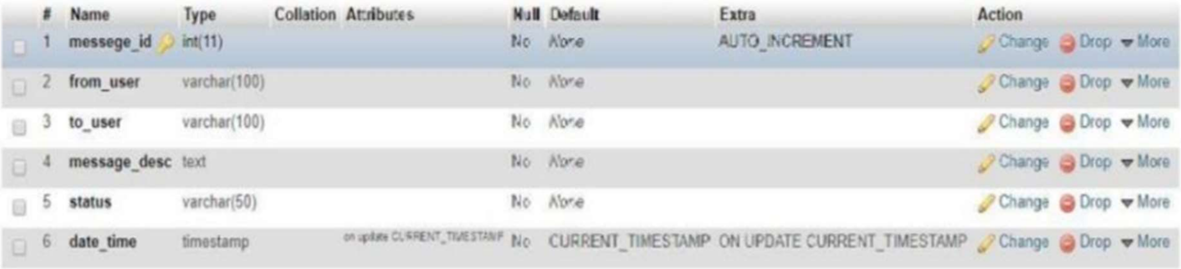
5. comment



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	comment_id	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	post_id	int(11)			No	None		Change Drop More
3	comment_user_id	varchar(100)			No	None		Change Drop More
4	first_name	varchar(50)			No	None		Change Drop More
5	middle_name	varchar(50)			No	None		Change Drop More
6	last_name	varchar(50)			No	None		Change Drop More
7	comment_desc	mediumtext			No	None		Change Drop More
8	cmnt_date_time	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Change Drop More

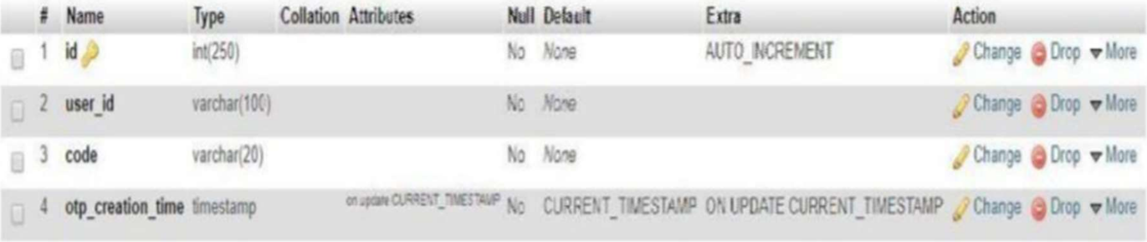
Table no: 6.2.5 and 6.2.6

6. message



	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
□	1	messege_id	int(11)			No	None	AUTO_INCREMENT	Change Drop More
□	2	from_user	varchar(100)			No	None		Change Drop More
□	3	to_user	varchar(100)			No	None		Change Drop More
□	4	message_desc	text			No	None		Change Drop More
□	5	status	varchar(50)			No	None		Change Drop More
□	6	date_time	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Change Drop More

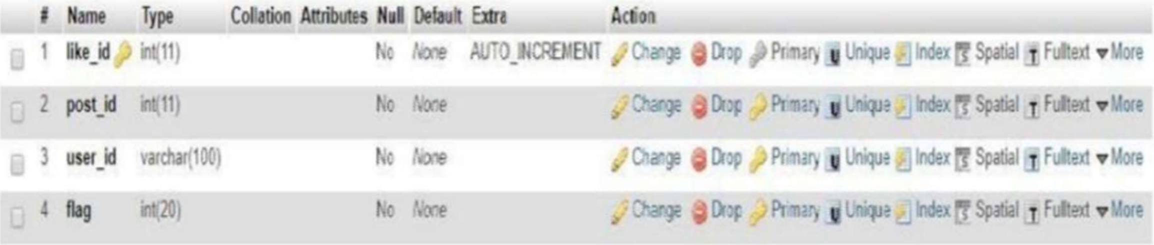
7. otp



	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
□	1	id	int(250)			No	None	AUTO_INCREMENT	Change Drop More
□	2	user_id	varchar(100)			No	None		Change Drop More
□	3	code	varchar(20)			No	None		Change Drop More
□	4	otp_creation_time	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Change Drop More

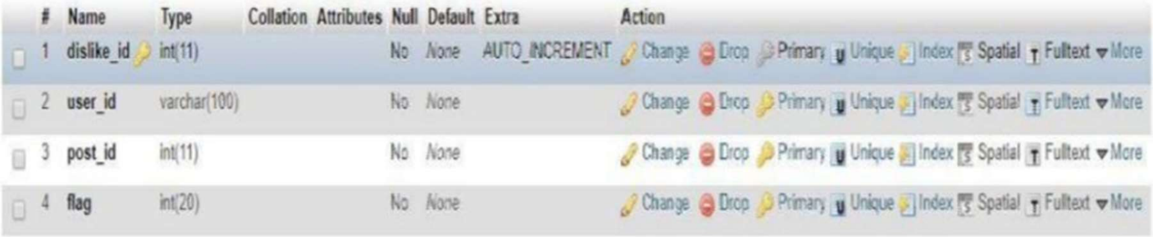
Table no: 6.2.7 and 6.2.8

8. user_like



	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
□	1	like_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext More
□	2	post_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
□	3	user_id	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
□	4	flag	int(20)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

9. user_dislike



	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
□	1	dislike_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext More
□	2	user_id	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
□	3	post_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
□	4	flag	int(20)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

Table no: 6.2.9 and 6.2.10

7. SOURCE CODE

Server Side:

Index.js

```
import express from "express";

import bodyParser from "body-parser";

import mongoose from "mongoose";

import cors from "cors";

import dotenv from "dotenv";

import multer from "multer";

import helmet from "helmet";

import morgan from "morgan";

import path from "path";

import { fileURLToPath } from "url";

import authRoutes from "./routes/auth.js";

import userRoutes from "./routes/users.js";

import postRoutes from "./routes/posts.js";

import { register } from "./controllers/auth.js";

import { createPost } from "./controllers/posts.js";

import { verifyToken } from "./middleware/auth.js";

import User from "./models/User.js";

import Post from "./models/Post.js";

import { users, posts } from "./data/index.js";
```

```
[userIds[6], true],

[userIds[3], true],

]),

comments: [

"I lied again, one more random comment",

"Why am I doing this?",

"I'm bored",

"I'm still bored",

"All I want to do is play video games",

"I'm going to play video games",

],

},

{

_id: new mongoose.Types.ObjectId(),

userId: userIds[6],

firstName: "Carly",

lastName: "Vowel",

location: "Chicago, IL",

description:

"Just a short description. I'm tired of typing. I'm going to play video games now.",

picturePath: "post5.jpeg",

userPicturePath: "p8.jpeg",
likes: new Map([
[userIds[1], true],
[userIds[3], true],
2[userIds[5], true],
[userIds[7], true],
]),
comments: [
"I lied again, one more random comment",
"Why am I doing this?",
"Man I'm bored",
"What should I do?",
"I'm going to play video games",
],
},
{
_id: new mongoose.Types.ObjectId(),
userId: userIds[7],
firstName: "Jessica",
lastName: "Dunn",
location: "Washington, DC",
description:
"For the last time, I'm going to play video games now. I'm tired of typing. I'm going to play video games now.",
picturePath: "post6.jpeg",
userPicturePath: "p9.jpeg",
likes: new Map([
[userIds[1], true],
[userIds[2], true],
]),

comments: [
"Can I play video games now?",
"No let's actually study",
"Never mind, I'm going to play video games",
"Stop it.",
"Michael, stop it.",
],
},
];
```

Posts.js

```
import express from "express";
import { getFeedPosts, getUserPosts, likePost } from "../controllers/posts.js";
import { verifyToken } from "../middleware/auth.js";

const router = express.Router();
```

```
/* READ */
router.get("/", verifyToken, getFeedPosts);
router.get("/:userId/posts", verifyToken, getUserPosts);

/* UPDATE */
router.patch("/:id/like", verifyToken, likePost);

export default router;
```

Users.js

```
import express from "express";
import {
getUser,
getUserFriends,
addRemoveFriend,
} from "../controllers/users.js";
import { verifyToken } from "../middleware/auth.js";

const router = express.Router();
```

Posts.js

```
import Post from "../models/Post.js";
import User from "../models/User.js";

/* CREATE */
export const createPost = async (req, res) => {
try {
const { userId, description, picturePath } = req.body;
const user = await User.findById(userId);
const newPost = new Post({
userId,
firstName: user.firstName,
lastName: user.lastName,
location: user.location,
description,
userPicturePath: user.picturePath,
picturePath,
likes: {},
comments: [],
});
await newPost.save();
```

```
const post = await Post.find();

res.status(201).json(post);

} catch (err) {

res.status(409).json({ message: err.message });

}

};


/* READ */

export const getFeedPosts = async (req, res) => {

try {

const post = await Post.find();

res.status(200).json(post);

} catch (err) {

res.status(404).json({ message: err.message });

}

};


export const getUserPosts = async (req, res) => {

try {

const { userId } = req.params;

const post = await Post.find({ userId });

res.status(200).json(post);
```

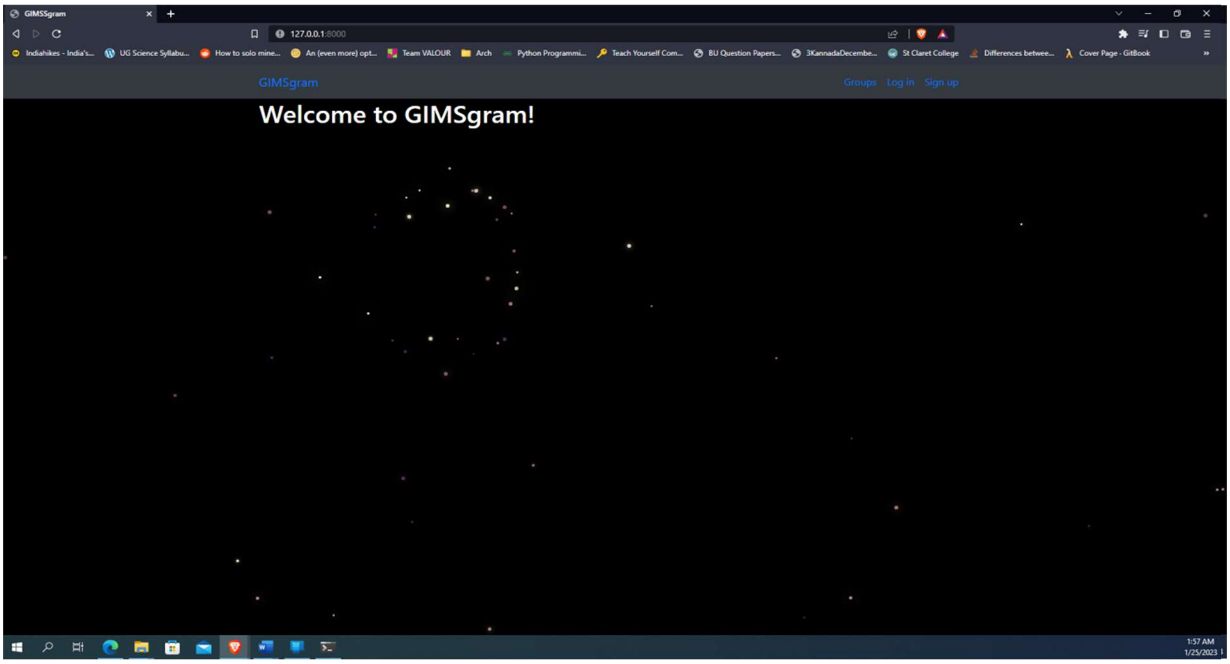
```
} catch (err) {
res.status(404).json({ message: err.message });
}
};
```

Theme.js

```
// color design tokens export
export const colorTokens = {
grey: {
0: "#FFFFFF",
10: "#F6F6F6",
50: "#F0F0F0",
100: "#E0E0E0",
200: "#C2C2C2",
300: "#A3A3A3",
400: "#858585",
500: "#666666",
600: "#4D4D4D",
700: "#333333",
800: "#1A1A1A",
900: "#0A0A0A",
1000: "#000000",
},
primary: {
50: "#E6FBFF",
100: "#CCF7FE",
200: "#99EEFD",
300: "#66E6FC",
400: "#33DDFB",
500: "#00D5FA",
600: "#00A0BC",
700: "#006B7D",
800: "#00353F",
900: "#001519",
},
};
```

8. SCREENSHOTS

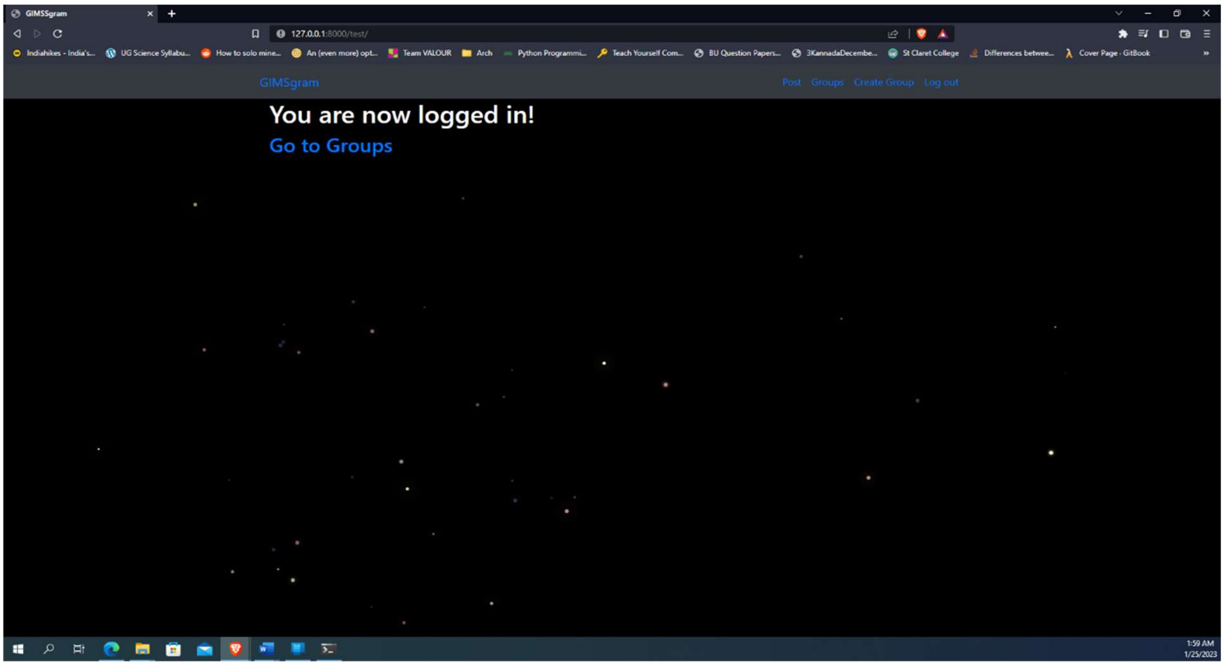
Home page



Welcome to GIMSgram!

Figure no:8.1.1

Home page after login

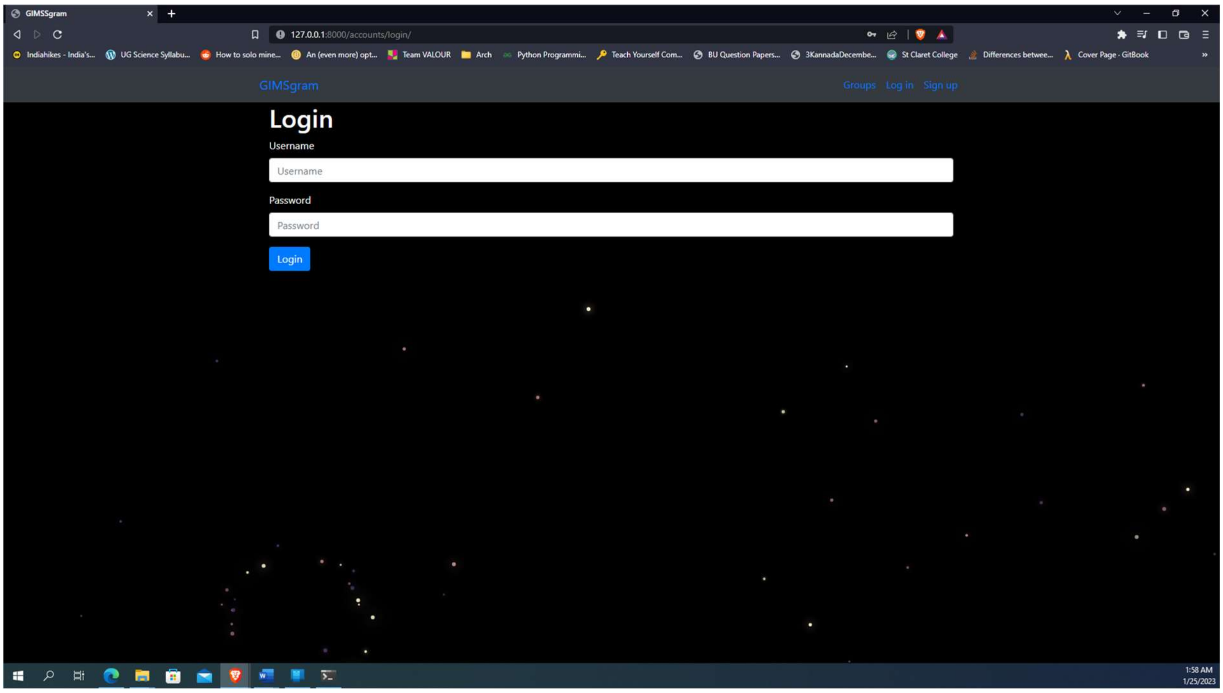


You are now logged in!

Go to Groups

Figure no:8.1.2

Login page



GIMSgram   Groups Log in Sign up

Login

Username

Username

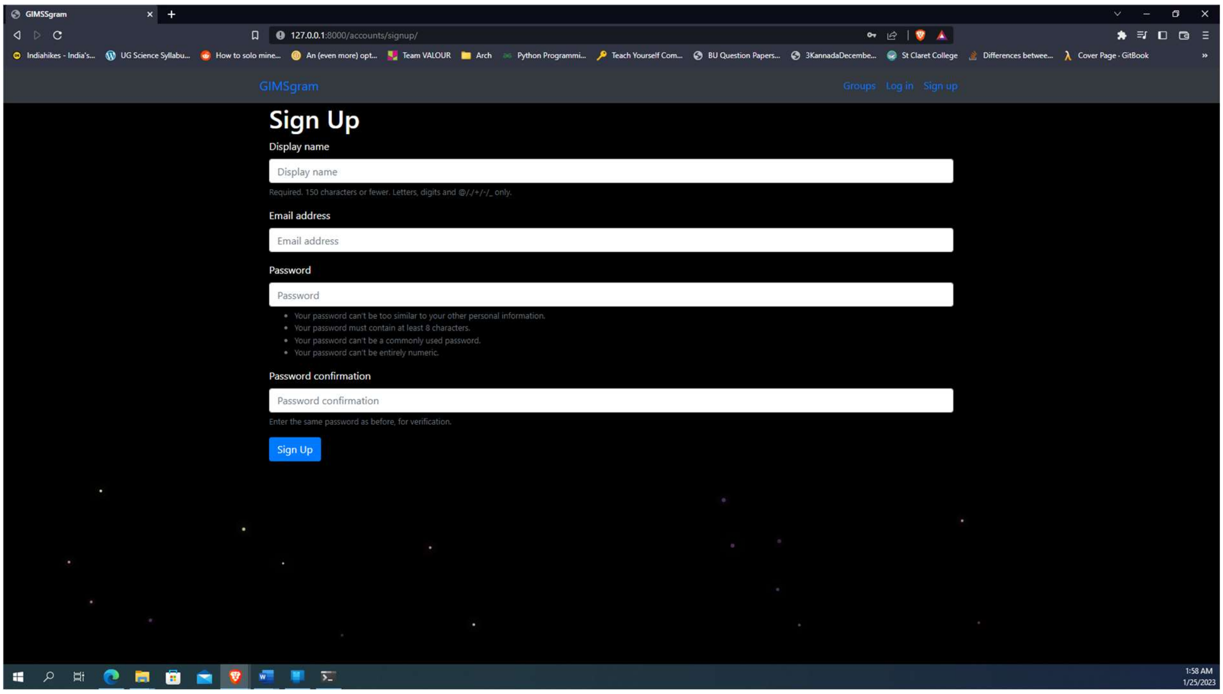
Password

Password

Login

Figure no:8.1.3

Signup page



GIMSgram   Groups Log in Sign up

Sign Up

Display name

Display name

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email address

Email address

Password

Password

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation

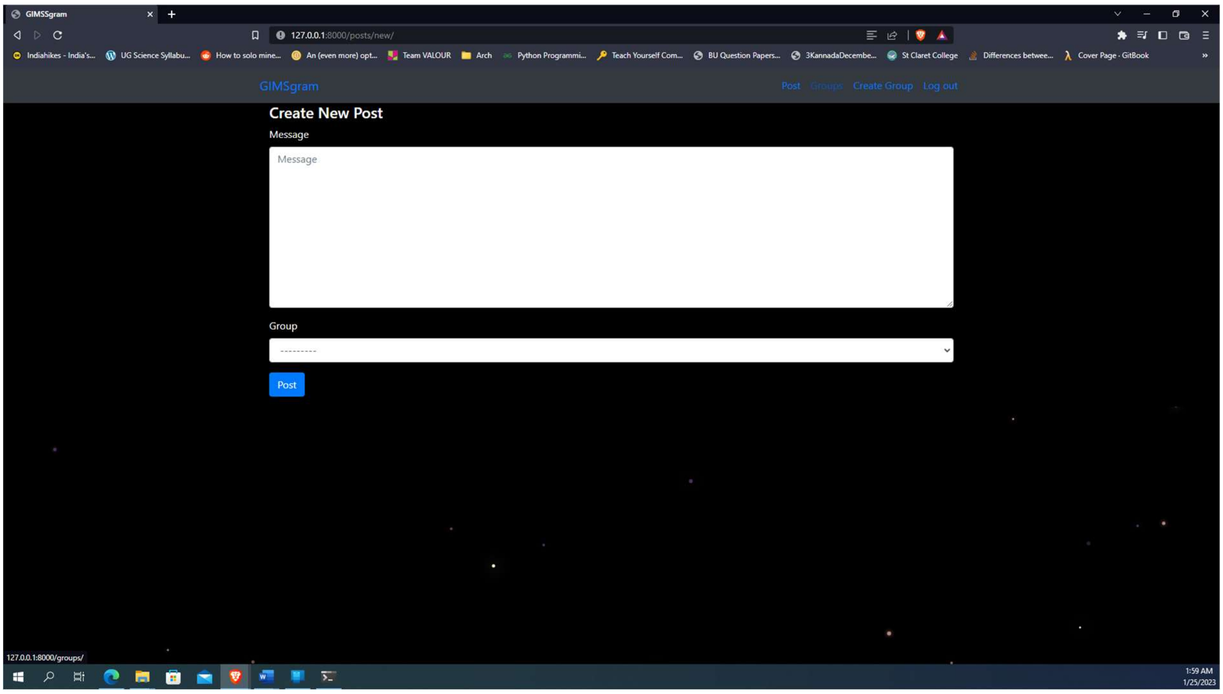
Password confirmation

Enter the same password as before, for verification.

Sign Up

Figure no:8.1.4

Post creation page



GIMSgram

Post Groups Create Group Log out

Create New Post

Message

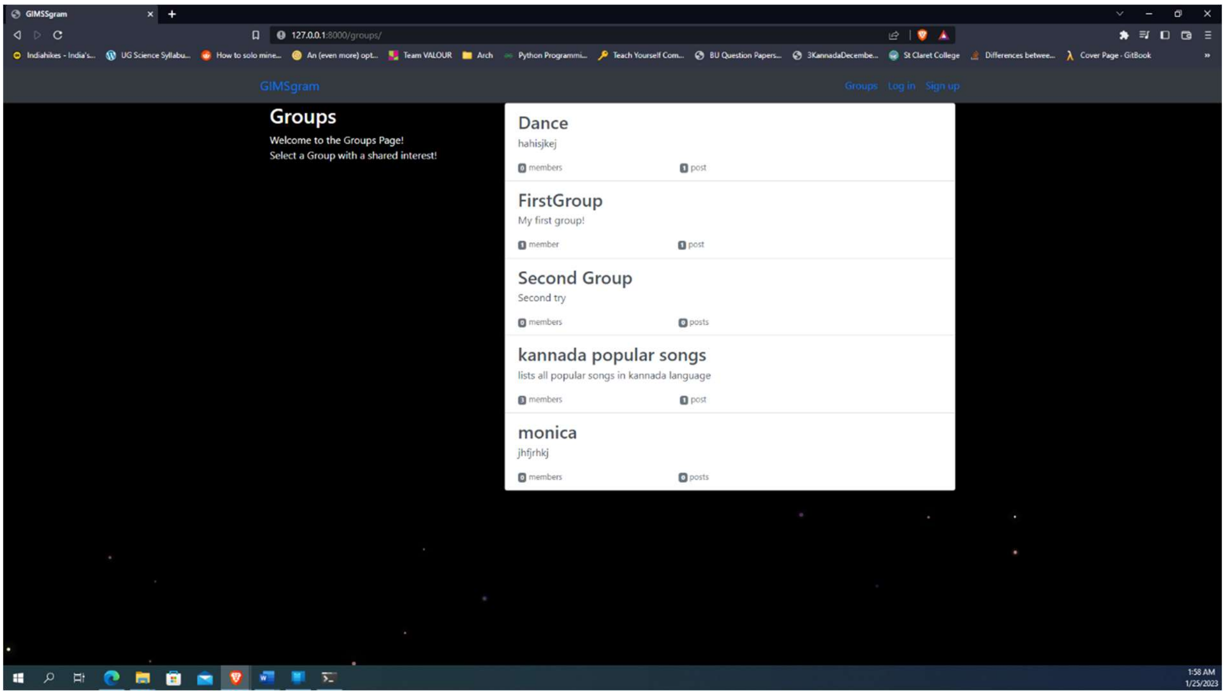
Message

Group

Post

Figure no:8.1.5

Groups page



GIMSgram

Groups Log in Sign up

Groups

Welcome to the Groups Page!
Select a Group with a shared interest!

Dance
hahisjkej
0 members 1 post

FirstGroup
My first group!
1 member 1 post

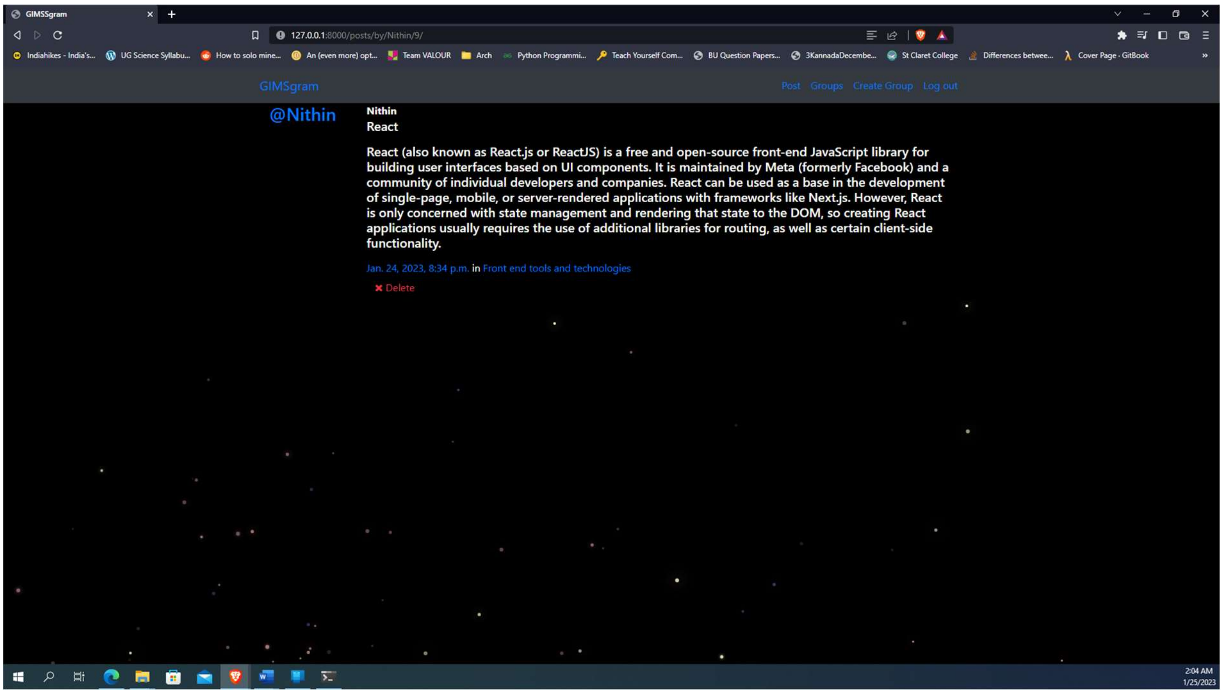
Second Group
Second try
0 members 0 posts

kannada popular songs
lists all popular songs in kannada language
3 members 1 post

monica
jhfjrhkj
0 members 0 posts

Figure no:8.1.6

Posts page



@Nithin

Nithin

React

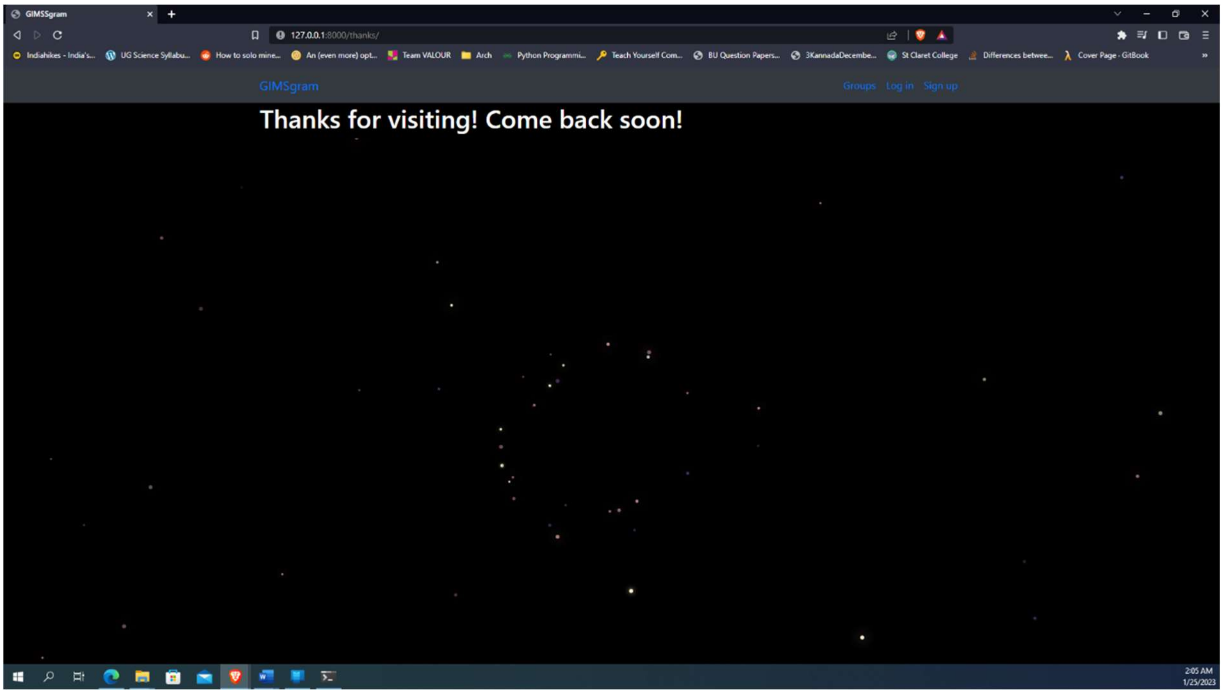
React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Jan. 24, 2023, 8:34 p.m. in Front end tools and technologies

✖ Delete

Figure no:8.1.7

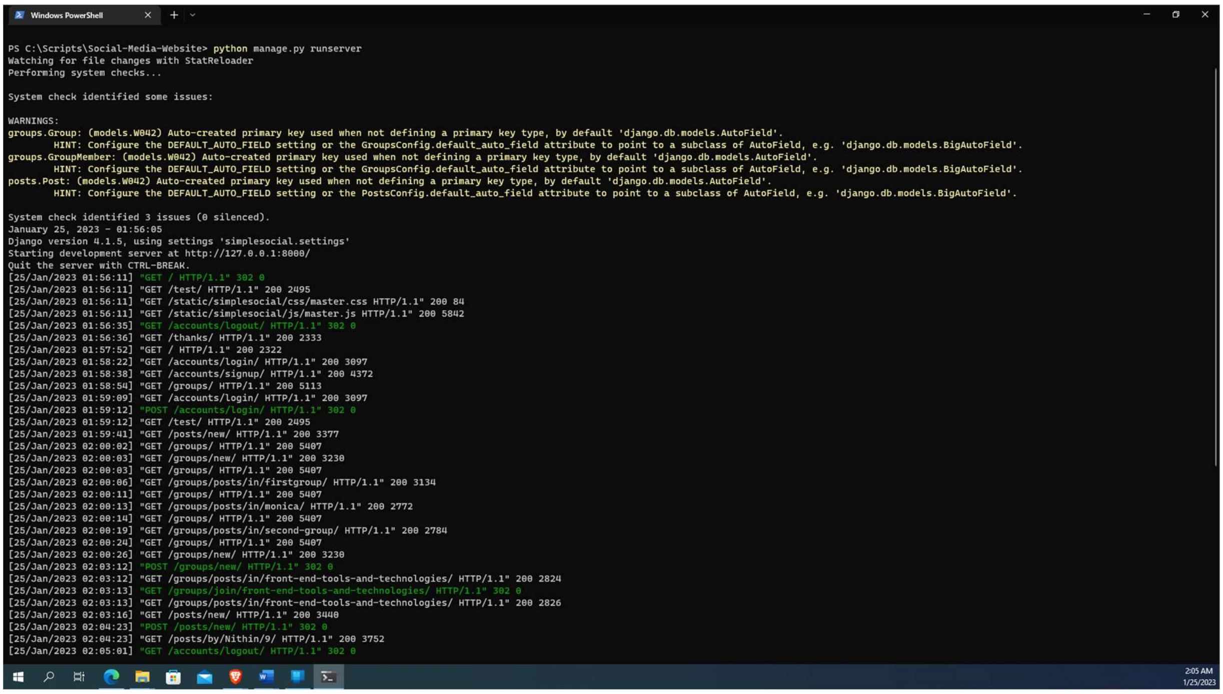
Logout page



Thanks for visiting! Come back soon!

Figure no:8.1.8

Command line server side



```
PS C:\Scripts\Social-Media-Website> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
groups.Group: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the GroupsConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
groups.GroupMember: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the GroupsConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
posts.Post: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
        HINT: Configure the DEFAULT_AUTO_FIELD setting or the PostsConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.

System check identified 3 issues (0 silenced).
January 25, 2023 - 01:56:05
Django version 4.1.5, using settings 'simplesocial.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[25/Jan/2023 01:56:11] "GET / HTTP/1.1" 302 0
[25/Jan/2023 01:56:11] "GET /test/ HTTP/1.1" 200 2495
[25/Jan/2023 01:56:11] "GET /static/simplesocial/css/master.css HTTP/1.1" 200 84
[25/Jan/2023 01:56:11] "GET /static/simplesocial/js/master.js HTTP/1.1" 200 5842
[25/Jan/2023 01:56:35] "GET /accounts/logout/ HTTP/1.1" 302 0
[25/Jan/2023 01:56:36] "GET /thanks/ HTTP/1.1" 200 2333
[25/Jan/2023 01:57:52] "GET / HTTP/1.1" 200 2322
[25/Jan/2023 01:58:22] "GET /accounts/login/ HTTP/1.1" 200 3097
[25/Jan/2023 01:58:38] "GET /accounts/signup/ HTTP/1.1" 200 4372
[25/Jan/2023 01:58:54] "GET /groups/ HTTP/1.1" 200 5113
[25/Jan/2023 01:59:09] "GET /accounts/login/ HTTP/1.1" 200 3097
[25/Jan/2023 01:59:12] "POST /accounts/login/ HTTP/1.1" 302 0
[25/Jan/2023 01:59:12] "GET /test/ HTTP/1.1" 200 2495
[25/Jan/2023 01:59:41] "GET /posts/new/ HTTP/1.1" 200 3377
[25/Jan/2023 02:00:02] "GET /groups/ HTTP/1.1" 200 5407
[25/Jan/2023 02:00:03] "GET /groups/new/ HTTP/1.1" 200 3230
[25/Jan/2023 02:00:03] "GET /groups/ HTTP/1.1" 200 5407
[25/Jan/2023 02:00:06] "GET /groups/posts/in/firstgroup/ HTTP/1.1" 200 3134
[25/Jan/2023 02:00:11] "GET /groups/ HTTP/1.1" 200 5407
[25/Jan/2023 02:00:13] "GET /groups/posts/in/monica/ HTTP/1.1" 200 2772
[25/Jan/2023 02:00:14] "GET /groups/ HTTP/1.1" 200 5407
[25/Jan/2023 02:00:19] "GET /groups/posts/in/second-group/ HTTP/1.1" 200 2784
[25/Jan/2023 02:00:24] "GET /groups/ HTTP/1.1" 200 5407
[25/Jan/2023 02:00:26] "GET /groups/new/ HTTP/1.1" 200 3230
[25/Jan/2023 02:03:12] "POST /groups/new/ HTTP/1.1" 302 0
[25/Jan/2023 02:03:12] "GET /groups/posts/in/front-end-tools-and-technologies/ HTTP/1.1" 200 2824
[25/Jan/2023 02:03:13] "GET /groups/join/front-end-tools-and-technologies/ HTTP/1.1" 302 0
[25/Jan/2023 02:03:13] "GET /groups/posts/in/front-end-tools-and-technologies/ HTTP/1.1" 200 2826
[25/Jan/2023 02:03:16] "GET /posts/new/ HTTP/1.1" 200 3440
[25/Jan/2023 02:04:23] "POST /posts/new/ HTTP/1.1" 302 0
[25/Jan/2023 02:04:23] "GET /posts/by/Nithin/9/ HTTP/1.1" 200 3752
[25/Jan/2023 02:05:01] "GET /accounts/logout/ HTTP/1.1" 302 0
```

Figure no:8.1.9

9. SYSTEM TESTING

9.1. INTRODUCTION

Testing is the major process involved in software quality assurance (QA). It is iterative process. Here test data is prepared and is used to test the modules individually. System testing makes sure that all components of the system function properly as a unit by actually forcing the system to fail.

The test causes should be planned before testing begins. Then as the testing progresses, testing shifts focus in an attempt to find errors in integrated clusters of modules and in the entire system. The philosophy behind testing is to find errors. Actually, testing is the estate of implementation that is aimed at ensuring that the system works actually and efficiently before implementation.

- Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. Thus, the system testing is a confirmation that all is correct and an opportunity to show the user that thesystem works. The final step involves Validation testing, which determines whether the software function as the user expected. The end-user rather than the system developer conduct this test most software developers as a process called “Alpha and Beta test” to uncover that only the end user seems able to find.

This is the final step in system life cycle. Here we implement the tested error-free system into real-life environment and make necessary changes, which runs in an online fashion. Here system maintenance is done every month or year based on company policies, and is checked for errors like runtime errors, long run errors and other maintenances like table verification and reports.

During the requirement analysis and design, the output is a document that is usually textual and non-executable. After the coding phase, computer programs are available that can be executed for testing purpose. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during the previous phases.

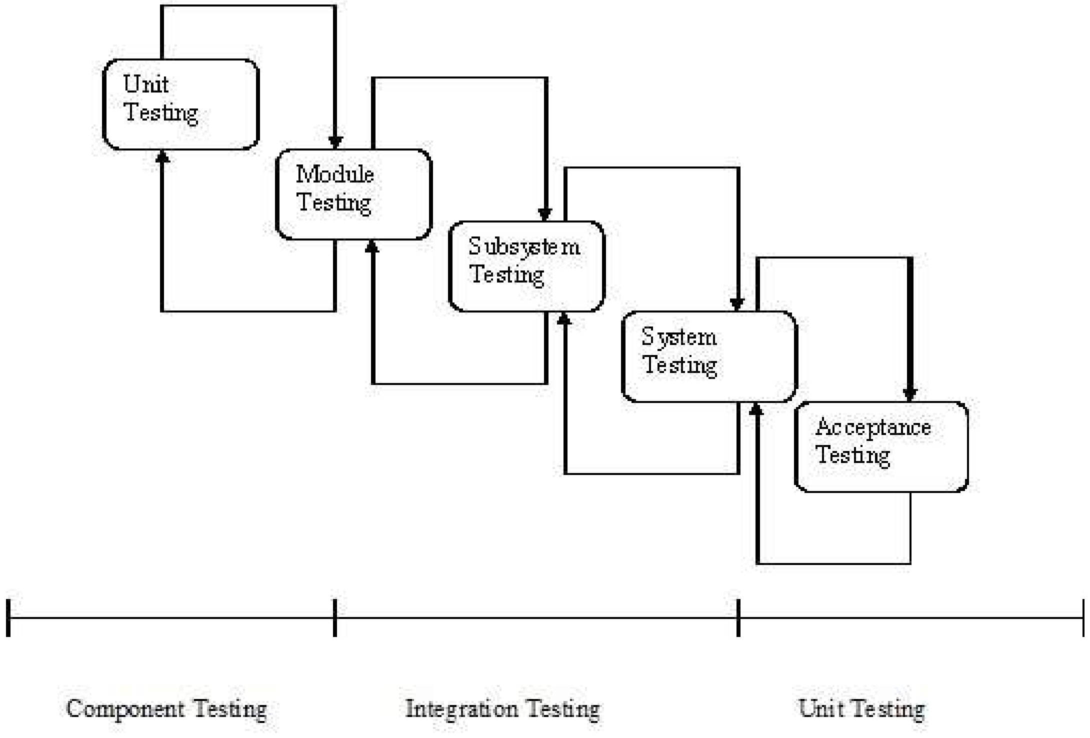


Figure No 9.1:1

9.2. TYPES OF TESTING

9.2.1. Unit Testing

Performance test consists of tests that ensure the fundamental programme logic is working properly and that programme inputs result in useful outputs. Every single line of code for an independent decision branch needs to be approved. It is finished following the conclusion of a single unit but before compromise. It is a rigorous but crucial test, and the outcome will be determined by statistics. Component-level checks that validate a workflow, software, or configuration are included in unit tests. It's due to the complexity of the software many programming parts. Unit tests make sure that each thrilling technique of its business connection adheres to the precise conclusions and contains sources of information and expected results that are depicted in a clear and understandable manner.

Given that this is not the ideal situation, where unit testing and coding are never driven as two separate phases, this testing is frequently handled as part of the problem lifecycle's linked code and unit court hearing.

9.2.2. Integration Testing

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps.

9.2.3. Validation Testing

To uncover functional errors, that is, to check whether functional characteristics confirm to specification or not specified.

9.2.4. System Testing

Utilitarian tests' standing and affiliation are determined by requirements, significant constraints, or excellent studies. Additionally, when relating to both Business measure streams, Data fields, defined metrics, and reformist cycles must all be thoroughly analyzed confirms that the connected programming structure's overall compliance with specifications. It assesses a plan to guarantee predictable results. Structure testing is demonstrated by the plan organized system blend test. System testing employs measure portrayals and transfers, with pre-driven cooperative partnerships and consolidation facilities maybe receiving special attention.

A sort of system testing is programming testing that is done throughout the board coupled structure evaluate the system's conformity with the corresponding requirements. Coordination testing that was successful is used as data when testing systems.

9.2.5. Acceptance Testing

When the system has no measure problem with its accuracy, the system passes through a final acceptance test. This test confirms that the system needs the original goal, Objective and requirements established during analysis. If the system fulfils all the requirements, it is finally acceptable and ready for operation.

9.2.6. White Box Testing

The element analyser takes into account the item's inherent capabilities, plan, and language in this difficult challenge rather than, at the very least, its inspiration. There's an excellent explanation for it. It is used to conduct tests in locations where a disclosure increase cannot.

9.2.7. Black Box Testing

Black box testing assesses a system solely from the outside, without the operator or tester knowing what is happening within the system to generate responses to test actions. A black box refers to a system whose behaviour has to be observed entirely by inputs and outputs.

Black box testing employs a number of different testing methods, including the decision table technique, the boundary value analysis technique, the state transition, the All-Pair testing method, the cause-effect graph method, the equivalence partitioning method, the error guessing method, the use case method, and the user story method.

9.3. TEST PLAN

A software project test plan is a document that describes the objectives, scope approach and focus of a software testing effort. This process of preparing a test plan is a useful way to think through the efforts needed to validate the acceptability of a software product. The completed document will help the people outside the test group understand 'Why and How' of production validation. Different test plans are used at different levels of testing.

9.3.1. TEST PLANS USED IN UNIT TESTING

Each module is tested for correctness whether it is meeting all the expected results. Condition loops in the code are properly terminated so that they don't enter into an infinite loop. Proper validations are done so as to avoid any errors related to data entry from user.

9.3.2. SYSTEM TESTING

System Testing refers to the manner of checking out a software program utility primarily based mostly on what its specification says its behavior need to be. In unique, we will growth check instances based completely totally on the specification of this device's behavior, without seeing an implementation of this system.

9.4. System Test Cases

Test Case No.	Testing scenario	Input	Expected result	Result
		Index Page Testing		
TC - 01	Run the application of Wi-Fi checker	Type link in the web browser and press enter web browser.	The home page of the application should be displayed.	The home page of the application is displaying.
TC - 02	Click on fetch list in the home page of the application.	Click on fetch list.	The Wi-Fi name connected by the machine should be displayed.	The Wi-Fi name connected by the machine is displaying.
TC - 03	Click on fetch list without connecting Wi-Fi.	Click on fetch list.	The Wi-Fi name connected by the machine should not be displayed.	The Wi-Fi name connected by the machine is not

				displaying.
TC - 04	Whitelisting of devices in backend.	Click on fetch list.	Should verify devices and accepting.	Verifying devices and accepting devices.

Table no:9.4.1

Test Case Number	Testing Scenario	Expected result	Result
	Registration Testing		
TC – 01	Clicking submit without entering details	Alert "Please fill all details"	Pass
TC – 02	Clicking submit without entering Username	Alert "Please fill Username"	Pass
TC – 03	Clicking submit without entering password	Alert "Please fill Password"	Pass
TC – 04	Clicking submit without entering email id	Alert "Please fill email id"	Pass

TC – 05	Clicking submit without entering phone number	Alert "Please fill contact number"	Pass
TC – 06	Clicking submit entering confirm password data which is not matching with password data	Alert "Password and Confirm Password do not match"	Pass
TC – 07	**Login Testing** Clicking submit without entering login details	Alert "Please enter the username and password"	Pass
TC – 08	Clicking submit without entering password	Alert "Please enter the password"	Pass
TC – 09	Clicking submit without entering Username	Alert "Please enter the Username"	Pass
TC – 10	Clicking submit entering wrong Username	Alert "Invalid User"	Pass
TC – 11	Clicking submit entering wrong password	Alert "Invalid User"	Pass
TC – 12	Clicking submit entering wrong Username and password	Alert "Invalid User"	Pass

Table no: 9.4.2

11. CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements. The objective of software planning is to provide a frame work that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

11. FUTURE ENHANCEMENT

In this system we would like to add up new features are

- User can post pictures
- liking and disliking of the user pictures
- Other college students can also use this website in future we will add up that features
- Other college students can login for this portal,where they get OTP for registration then after verification admin will approve them.

BIBLIOGRAPHY

1.Evans, Dave. "The internet of things: How the next evolution of the internet is changing everything." CISCO white paper 1.2011, pp. 1-11.

2.Asadi, Q. Wang, and V. Mancuso, “A survey on device-to-device communication in cellular networks,” IEEE Commun. Surveys Tuts., vol. 16, no. 4, pp. 1801–1819, 4th Quart., 2014.

3.N. Kato, “On device-to-device (D2D) communication [editor’s note],” IEEE Netw., vol. 30, no. 3, p. 2, May/Jun. 2016.

4.Bello, Oladayo, and SheraliZeadally. "Intelligent device-to-device communication in the internet of things." IEEE Systems Journal 10.3, 2016, pp.1172-1182.

5.Arabo, Abdullahi. "Mobile App collusions and its cyber security implications." Cyber Security and Cloud Computing (CSCloud), 2016 IEEE 3rd International Conference on. IEEE, 2016.

6.Elish, Karim O., Danfeng Yao, and Barbara G. Ryder. "On the need of precise inter-app ICC classification for detecting Android malware collusions." Proceedings of IEEE mobile security technologies (MoST), in conjunction with the IEEE symposium on security and privacy, 2015.

7.Haus, Michael, et al. "Security and Privacy in Device-to-Device (D2D) Communication: A Review." IEEE Communications Surveys & Tutorials 19.2, 2017, pp.1054-1079.

8.Dave Shackleford, “Application Whitelisting: Enhancing Host Security”, http://www.sans.org/reading_room/analysts_program/McAfee_09_App_ Whitelisting.pdf, Oct. 2009.

9.Pareek, Himanshu, Sandeep Romana, and P. R. L. Eswari. "Application whitelisting: approaches and challenges." International Journal of Computer Science, Engineering and Information Technology (IJCSEIT) 2.5, 2012.

10. Tehrani, Mohsen Nader, Murat Uysal, and Halim Yanikomeroglu. "Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions." IEEE Communications Magazine 52.5, 2014, pp.86-92.

11. Hassan, Khaled S., and Engy M. Maher. "Device-to-device communication distance analysis in interference limited cellular networks." Wireless Communication Systems (ISWCS 2013), Proceedings of the Tenth International Symposium on. VDE, 2013.

12. Tenzin Tendar. “Social Networking Project” April 09, 2014
https://www.slideshare.net/the10dar/document-33319174

APPENDIX

INTRODUCTION

Intra-College Communication System is a software which has been developed to override the Problems prevailing in the practicing manual system. This software is to eliminate and, in Some cases, reduce the hardships faced by this existing system. Moreover, this system is Effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also Provides error message while entering invalid data. No formal knowledge is needed for the User to use this system. Thus, by this all it provides it is user-friendly. Intra-College Communication System, as described above, can lead to error free, secure, reliable and fast Networking system. It can assist the user to concentrate on their other activities rather to Concentrate on the record keeping. Thus, it will help organization in better utilization of Resources.This is designed to assist in strategic planning, and will help you ensure that your organization Is equipped with the right level of information and details for your future goals. These systems Will ultimately allow you to better manage resources.

OBJECTIVE

- To gather all the college members, starting from administrative level to the basic clerical section into a single frame.
- To convey all the notice(s), event(s), corrigendum(s) etc. and moreover what is going around the college.
- To grow the teacher and student relationship stronger.
- To provide 24 X 7 connectivity with college campus.

EXISTING SYSTEM

- Reduce manual work, every institution, whether big or small, has challenges to overcome and Managing the information's of friends, users, shares, videos, photos.
- The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required take the database backup

PROPOSED SYSTEM

- The proposed system works based on a few conditions. We are expecting the colleges to have a domain registered and hand out emails based on that domains which helps reduce the manual work required to verify every student or a personal logging in to the system.

- In the case of a college where they are not able provide the emails, we have an admin page designed which can be used to manually check if a user can or cannot use the platform.The verification process is very much necessary as we don't want intruders from some other college to disrupt any kind of workflow happening on the network.

- If the college has a proper digital record of all the students studying at the college, we can also setup an One Time Password based login from the registry of the college, which again is the student's personal responsibility to notify the college when there is any change in such phone number.

MODULES

SERVER

The server class acts as a wrapper for all server functions for our social networking site. It essentially act as a link between all of the information such as accounts, account details, pages notes, etc to our database.

ACCOUNT

Each user who wants to use the site must create an account. This is the head class that all other objects use to determine what a user does and when the user did it.

POST

A Wall Post object works very similar to a Message object. The only difference is that the Wall Post objects are viewable to all friends of the user. Another difference is that a Wall Post object can hold comments

GROUPS/FRIENDS

The most important feature is being able to add and remove friends. In our Social Networking site, making friends is a fairly straightforward process. Users can type in the name of a friend in the search bar at the top

of their home page.

ACCOUNT CREATION

When a user accesses the site for the first time, he/she must create an account before using any of the site features. The account creation process is broken into three sections.

HARDWARE INTERFACE

Processor	AMD Phenom II X2 550 Processor 3.10 Ghz
Memory	4.0 GB
Hard disk space	1 TB
Monitor	21.5 Samsung sync master 2233sw
Others	Other required standard computer peripherals, such as keyboard and mouse

SOFTWARE INTERFACE

Server Environment	
Operating System	Microsoft Windows 7
Programming tool	Adobe Dreamweaver CS4
Database Server	MySQL database
Web server	Apache 2.2.11
Client Environment	
Internet browser	Chrome, Firefox…