CHAT SPHERE - DISCUSSION ROOM WEB APP

MINOR PROJECT-II

Submitted by

Bipasha Adhikari (9920103032)

Shreya Sharma (9920103035)

Aayush Rathore (9920103036)

Under the supervision of

Dr. Rashmi Kushwah

Department of Computer Science & Engineering And Information Technology

Jaypee Institute of Information Technology University, Noida

April 2023

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to Dr. Rashmi Kushwah, Assistant Professor (Senior Grade) in the Department of Computer Sciences & Engineering at Jaypee Institute of Information Technology (JIIT), India for her generous guidance, help, and valuable suggestions.

I also wish to extend my thanks to my classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

Signature(s) of Students

Bipasha Adhikari (9920103032)

Shreya Sharma (9920103035)

Aayush Rathore (9920103036)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Noida

Date: 22 April, 20

Name: Bipasha Adhikari

Enrolment No.: 9920103032

Name: Shreya Sharma

Enrolment No.: 9920103035

Name: Aayush Rathore

Enrolment No.: 9920103036

CERTIFICATE

This is to certify that the work titled “Chat Sphere – Discussion Room Web App” submitted by Bipasha Adhikari, Shreya Sharma, and Aayush Rathore of B.Tech of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Signature of Supervisor:

Name of Supervisor: Dr Rashmi Khuswah

Assistant Professor (Senior Grade) in the Department of Computer Sciences & Engineering

Jaypee Institute of Information Technology (JIIT), Sector-128, Noida- 201304

Date: 22 April 2023

ABSTRACT

Chat Sphere is a web app designed to foster online discussions and debates in a safe and inclusive environment. With its easy-to-use interface, users can create and join discussion rooms, which can be public or private, and engage in real-time conversations with others. The app allows users to share multimedia content such as images, videos, and links to enhance their discussions.

One of the key features of Chat Sphere is its commitment to promoting civil discourse. The app includes moderation tools that enable users to report and flag inappropriate content, ensuring that discussions remain respectful and productive. The app is also designed to be inclusive, providing a platform for people from diverse backgrounds and perspectives to engage in meaningful discussions.

In addition to text-based discussions, Chat Sphere includes chat capabilities that enable users to participate in real-time conversations. This feature allows users to interact with each other in a more conversational manner and build a sense of community.

To further personalize the user experience, Chat Sphere offers a range of customization options. Users can choose from different themes, upload custom avatars, and add custom background images to their discussion rooms. The app also includes a notification system that alerts users when new messages or discussions are added to their rooms, ensuring that they never miss out on important conversations.

Overall, Chat Sphere provides a user-friendly and inclusive platform for people to engage in discussions and debates on a wide range of topics. Its moderation tools and customizable features make it an ideal choice for individuals and organizations looking to create safe and productive online spaces for civil discourse.

LIST OF NOMENCLATURE

App.	Application
Env	Environment
Sys.	System
Lang.	Language
Mang.	Management

LIST OF ABBREVIATIONS

IDE	Integrated Development Environment
GCC	GNU Compiler Collection
SDK	Software Development kits
OCR	Optical character recognition OR Optical character reader
I/P	Input
O/P	Output
w/ or w/o	With or Without
DBMS	Database Management Systems
MVT	Model View Template

Table of Contents

CHAPTER-1

INTRODUCTION

1.1 Web Programming

Web programming is the process of creating dynamic websites and web apps. It involves designing, coding, testing, and maintaining the website or app. Web programming is a rapidly growing field due to the increasing demand for web-based apps and the rise of e-commerce. (Roca, n.d.)

Web programming involves the use of various programming languages such as HTML, CSS, and JavaScript. HTML (Hypertext Markup Language) is used to structure content on a web page, while CSS (Cascading Style Sheets) is used to style the content. JavaScript is used to make the web page dynamic by adding interactivity and behavior.

Web programming also involves the use of server-side technologies such as PHP, Python, Ruby, and ASP.NET. These languages allow web programmers to create dynamic websites that can interact with users, store and retrieve data from databases, and communicate with web servers.

To simplify the coding process, web programmers use various frameworks and libraries, such as React, Angular, Vue, and jQuery. These frameworks provide pre-built components, tools, and libraries that help web programmers create complex apps quickly and efficiently. Additionally, these frameworks and libraries help ensure the website or app is secure and optimized for performance.

Web programming also involves the use of databases such as MySQL, PostgreSQL, and MongoDB. Databases allow web programmers to store and retrieve large amounts of data and provide a way to organize and manage data efficiently. Web programmers use DBMS such as phpMyAdmin and pgAdmin to interact with databases.

Web programming requires strong problem-solving skills, attention to detail, and an understanding of user experience design principles. In addition, web programmers need to stay up-to-date with the latest trends and best practices to create effective and efficient web apps. They also need to ensure that the website or app is optimized for search engines and follows accessibility guidelines.

In conclusion, web programming is a complex and constantly evolving field that involves designing, coding, testing, and maintaining dynamic websites and web apps. Web programmers use various programming languages, frameworks, and tools to create effective and efficient web-based solutions.

1.1.1. Overview

There are two broad divisions of web development – front-end development (also called Client-side development) and back-end development (also called server-side development). Front-end development refers to constructing what a user sees when they load a web app – the content, design, and how you interact with it. This is done with three codes – HTML, CSS, and JavaScript. HTML stands for HYPER TEXT MARKUP LANGUAGE, is a special code for 'marking up' text in order to turn it into a web page. Every web page on the net is written in HTML, and it is the backbone of every web app. CSS, short for Cascading Style Sheet, is a code for setting style rules for the appearance of web pages. CSS handles the cosmetic side 10 of the web. Finally, JavaScript is a scripting language that's widely used to add functionality and interactivity to web pages.

1.1.2 Description

Our project is based on Python Django is a web framework.

Django is a popular Python-based web framework that allows developers to build complex web apps quickly and efficiently. Django follows the Model-View-Controller (MVC) architectural pattern and is based on the principle of Don't Repeat Yourself (DRY).

Django includes many built-in features that make web development faster and easier, such as a built-in ORM (Object-Relational Mapping) that allows developers to work with databases without writing SQL code, a template system that allows for easy separation of the design and the content, and a built-in administrative interface for managing site content.

Django is also highly customizable, allowing developers to add or remove features based on their specific needs. Django includes a large ecosystem of third-party packages and libraries that can be easily integrated into projects.

Django's security features are also robust, with built-in protection against common web app security vulnerabilities such as cross-site scripting (XSS) and SQL injection. Django also includes features such as password hashing, user authentication, and permissions and access controls to help secure web apps.

Overall, Django is a powerful web framework that is well-suited for building complex and scalable web apps. Its built-in features, customizable nature, and security features make it a popular choice for web development.

1.2 Chat Communication System

A chat communication system is a software app that allows users to exchange messages in real-time. Chat systems can be implemented in various ways, such as a standalone app, a web-based app, or integrated into a larger software system.

A chat communication system typically consists of a server and a client app. The server acts as a central point of communication and manages the exchange of messages between clients. The client app is installed on the user's device and connects to the server to send and receive messages.

Modern chat communication systems often include advanced features such as group chat, file sharing, voice and video calling, and screen sharing. These features enhance the user experience and make it easier for users to collaborate and communicate effectively.

Chat communication systems can be used for various purposes, such as team collaboration in the workplace, customer support in businesses, or social networking between friends and family. They are particularly useful for remote teams, enabling team members to communicate and collaborate from different locations.

There are many chat communication systems available in the market, both open-source and proprietary. Some popular examples of chat communication systems include Slack, Microsoft Teams, Discord, WhatsApp, and Facebook Messenger. These systems offer different features and capabilities, catering to different use cases and user needs.

Overall, chat communication systems are an essential tool for modern communication and collaboration, offering an efficient and effective way for people to communicate and work together in real-time.

CHAPTER – 2

BACKGROUND STUDY

College is an important time in a person's life for many reasons, but two of the most significant are the opportunity to learn skills and build a network. While the primary focus of the college is usually academics, the benefits of college extend beyond the classroom.

In College, students have the chance to learn many new skills that will be beneficial to their future careers. In college, students have access to a variety of courses, programs, and extracurricular activities that can help them develop their skills in areas such as critical thinking, problem-solving, communication, and leadership. These skills are highly valued by employers and are essential for success in any field.

The development of a chat web app project for discussions is an innovative initiative that aims to bring together people from different parts of the world to engage in meaningful discussions on various topics. The project is designed to provide users with an interactive platform that fosters open and respectful communication, promoting intellectual discourse and knowledge sharing. (“The Six Biggest New Ideas In Chat” 2006)

The chat web app project has a user-friendly interface that allows users to create or join chat rooms on various topics of their interest. The app enables users to customize their profiles with avatars and information about themselves, facilitating the creation of a community where users can connect and interact with one another.

The chat rooms are organized by categories, making it easy for users to find chat rooms that interest them. The project utilizes modern web technologies such as Django, socket.io to provide real-time communication between users. Python Django is a web framework that allows to quickly create efficient web pages. The use of these technologies enables the app to provide users with an interactive experience that mimics the feel of a face-to-face conversation.

The chat web app project's primary goal is to encourage meaningful and respectful discussions among users on various topics. The project aims to create a platform that promotes intellectual discourse and knowledge sharing while also facilitating the creation of a community that encourages social interaction. (“Development of Chat Application”, n.d.)

College is an ideal time to build a network. During their time in college, students have the opportunity to meet and interact with people from a variety of backgrounds and with different interests and goals. These connections can be valuable both during and after college. Students can build relationships with professors, fellow students, alumni, and professionals in their fields of interest. These relationships can lead to internships, job opportunities, mentorship, and other benefits that can help students achieve their goals. Building a network is an essential part of career success, and college provides an ideal Env. for students to start building their professional networks.

But networking with people from your college to have discussions related to many things(lang., doubts related to placements, etc) is a task. This thing motivated us to make a website for college students needing counseling or discussion so they don't waste their valuable college time.

So we have come up with an idea to make a website to solve this problem. Our project is designed to provide users with a platform to engage in meaningful discussions with each other on various topics. With the increasing need for virtual communication in today's world, we have created this web app to offer a convenient and efficient way for users to connect and exchange ideas and perspectives. Our chat web app is designed to foster a collaborative and inclusive Env, providing a user-friendly interface that is easy to navigate. Users can create their accounts, set up their profiles, and join discussion groups that match their interests.

CHAPTER – 3

DETAILED DESIGN

Login: Users must log in to the site to create a room or join an existing one. This helps to ensure the security and privacy of the conversations that take place on the platform.

Profile creation: Users can create a profile on the site if they do not already have an account. This profile information is stored in a sqlite3 database. Users can include information such as their name, profile picture, bio, and other details that they want to share with others on the platform.

Room creation/joining: Users can create a room of their own or join an existing one. This allows users to connect with others who share similar interests and engage in discussions with them.

Browse topics: Users can browse all the rooms based on a certain topic in the "Browse Topics" section. This makes it easier for users to find rooms that are relevant to their interests and join them.

Recent activities: Users can see the most recent activities on the site, such as who messaged in which room and what was the message, in the "Recent Activities" section. This allows users to stay up-to-date on what's happening on the platform and join in on conversations that interest them.

Profile viewing: Users can view their own profile as well as other users' profiles. This allows users to learn more about each other and connect with people who share similar interests.

Overall, these features help to create a vibrant and engaging community on the platform where users can connect with each other and engage in interesting conversations.

CHAPTER – 4

IMPLEMENTATION

Our project is a web app based on Django framework. Python Django is a web framework that allows to quickly create efficient web pages. Django is also called batteries included framework because it provides built-in features such as Django Admin Interface, default database – SQLite3, etc. When you're building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a mang. panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use.

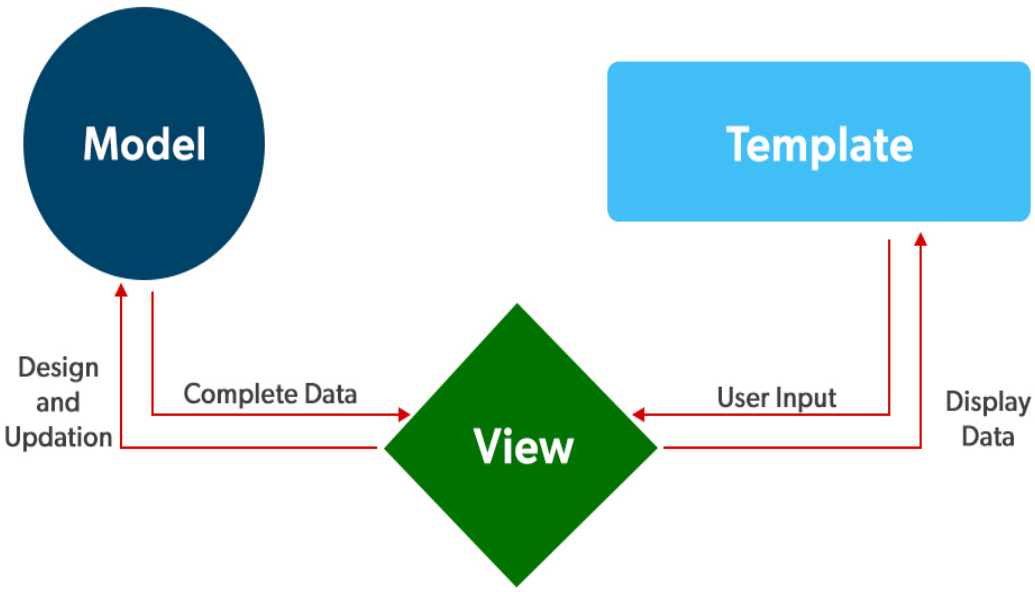
4.1. Django Architecture

Django is based on MVT architecture which has the following three parts –

Model: The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire app and is represented by a database (generally relational databases such as MySql, Postgres).

View: The View is the user interface that you see in your browser when you render a website. It is represented by HTML/CSS/Javascript and Jinja files.

Template: A template consists of static parts of the desired HTML O/P as well as some special syntax describing how dynamic content will be inserted. ("Django Tutorial" 2023)



4.2 Setting up the Virtual Environment

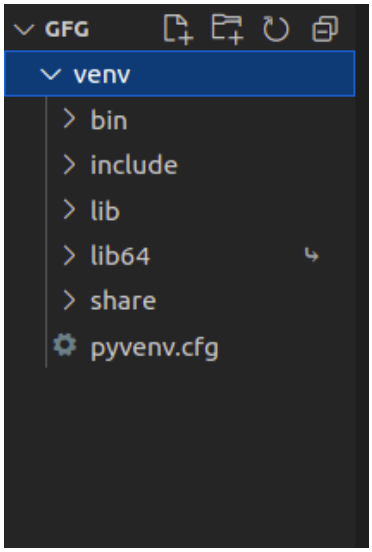
Most of the time when you'll be working on some Django projects, you'll find that each project may need a different version of Django. This problem may arise when you install Django in a global or default env. To overcome this problem we will use virtual env in Python. This enables us to create multiple different Django env on a single computer. To create a virtual env type the below command in the terminal. (guide, n.d.)

python3 -m venv ./name

Here the name suggests the name of the virtual environment. Let's create our virtual env with the name as venv only. So the command to create it will be –

python3 -m venv ./venv

After running the above command you will see a folder named venv with the following sub-directories.



After creating the virtual env let's activate it. To activate it type the below command in the terminal.

source ./venv/bin/activate

In the above command ./ is used to tell the current working directory.

Note: If you have your virtual env set up in another location and your terminal opened up in another location, then provide the location to the venv folder i.e. our virtual env folder.

After you run the above command you should see (venv) at the starting of every line of your terminal.

4.3 Installing Django

We can install Django using the pip command. To install this type the below command in the terminal.

pip install django

4.4 Starting the project

To initiate a project of Django on Your PC, open Terminal and Enter the following command

django-admin startproject projectName

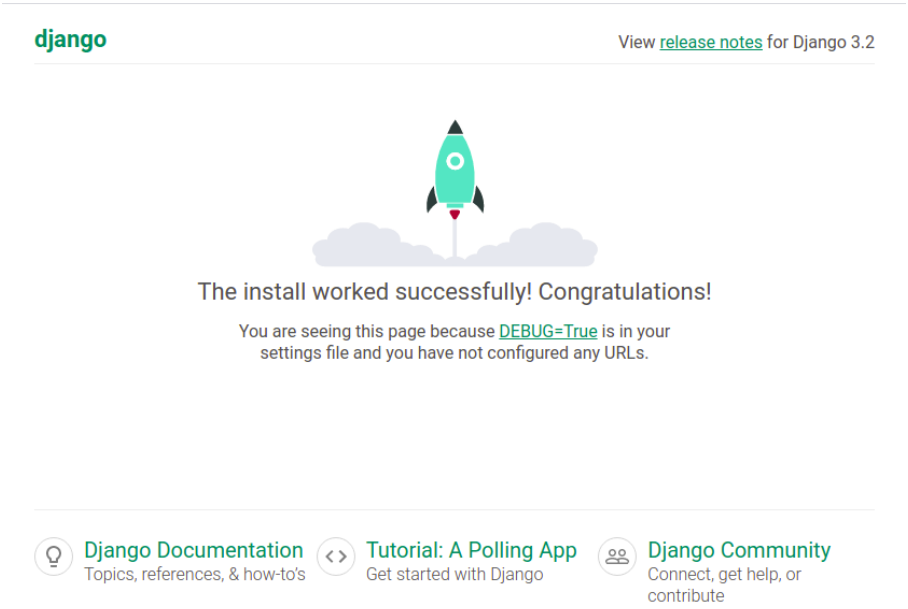
A New Folder with the name projectName will be created. To enter in the project using the terminal enter command

cd projectName

Now let's run the server and see everything is working fine or not. To run the server type the below command in the terminal.

python manage.py runserver

After running the server go to http://127.0.0.1:8000/ and you'll see something like this –



4.4.1 How to implement the functionalities

Creating a Django app: Once the project is created, create a new app within the project. This can be done using the command "python manage.py startapp appname" in the terminal. Replace "appname" with the name of your app.

Creating models: To store user information, room information, and recent activities, create models for each of them. This can be done in the app's models.py file. For example, the user model can have fields such as username, password, email, bio, and profile picture.

Creating views: Create views for each of the functionalities mentioned above. For example, create a view for the login page, signup page, room creation page, room search page, profile page, etc. These views can be defined in the app's views.py file.

Creating templates: Create templates for each of the views. These templates can be HTML files that display the content of the views. For example, the login page template can have fields for username and password, and a submit button that sends the information to the server.

Configuring URLs: Configure the app's URLs to map to the appropriate views. For example, the login page URL can be mapped to the login view.

Implementing authentication: Django provides built-in authentication views and functionality, which can be used to implement login and signup functionality. This can be done by configuring the settings.py file and using the appropriate views and templates.

Implementing room creation and search: Create views and templates for creating and searching for rooms. The room data can be stored in the database, and the views can query the database to display the appropriate information.

Implementing browse topics: Create a view and template for browsing topics. The topics can be stored in the database, and the rooms can be linked to the appropriate topics using foreign keys.

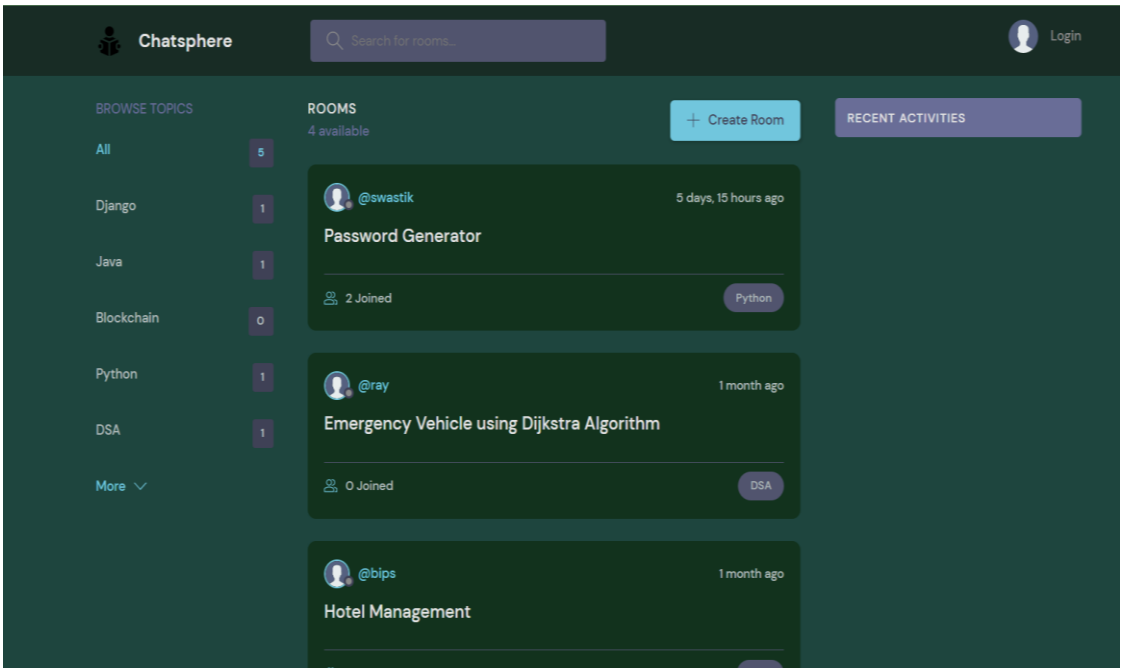
Implementing recent activities: Create a model for storing recent activities, and a view and template for displaying the activities. The activities can be stored in the database, and the view can query the database to display the appropriate information.

Implementing profiles: Create a view and template for displaying user profiles. The user information can be stored in the database, and the view can query the database to display the appropriate information.

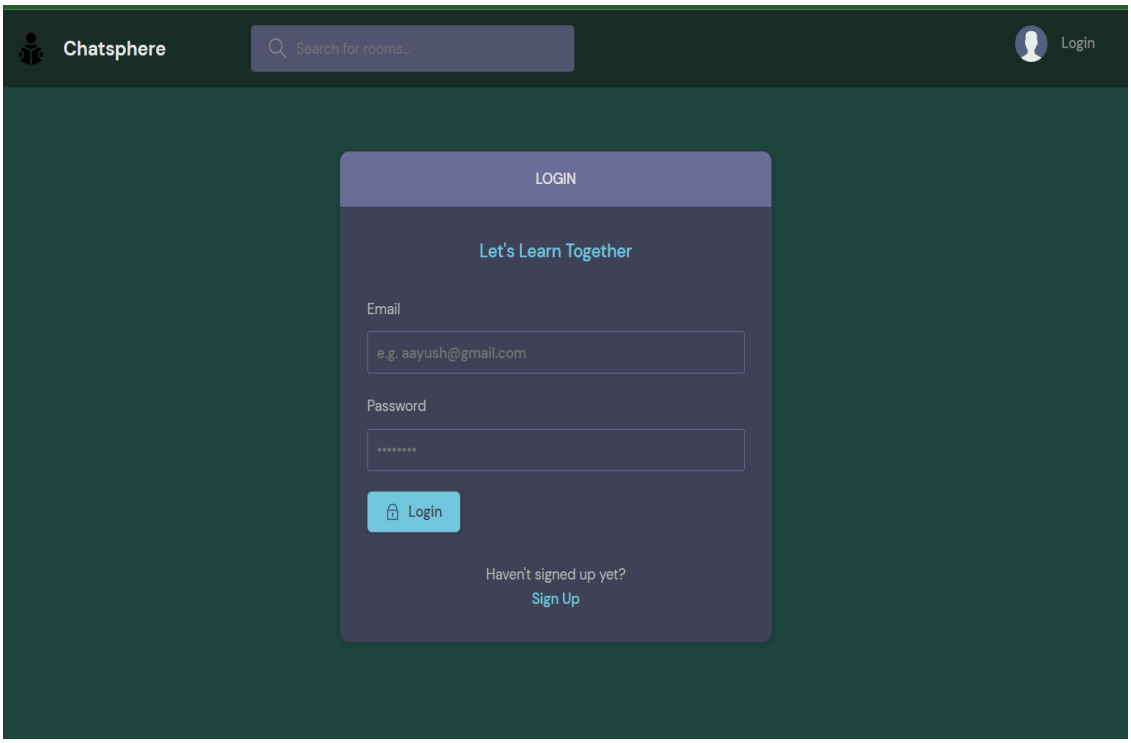
CHAPTER – 5

RESULT

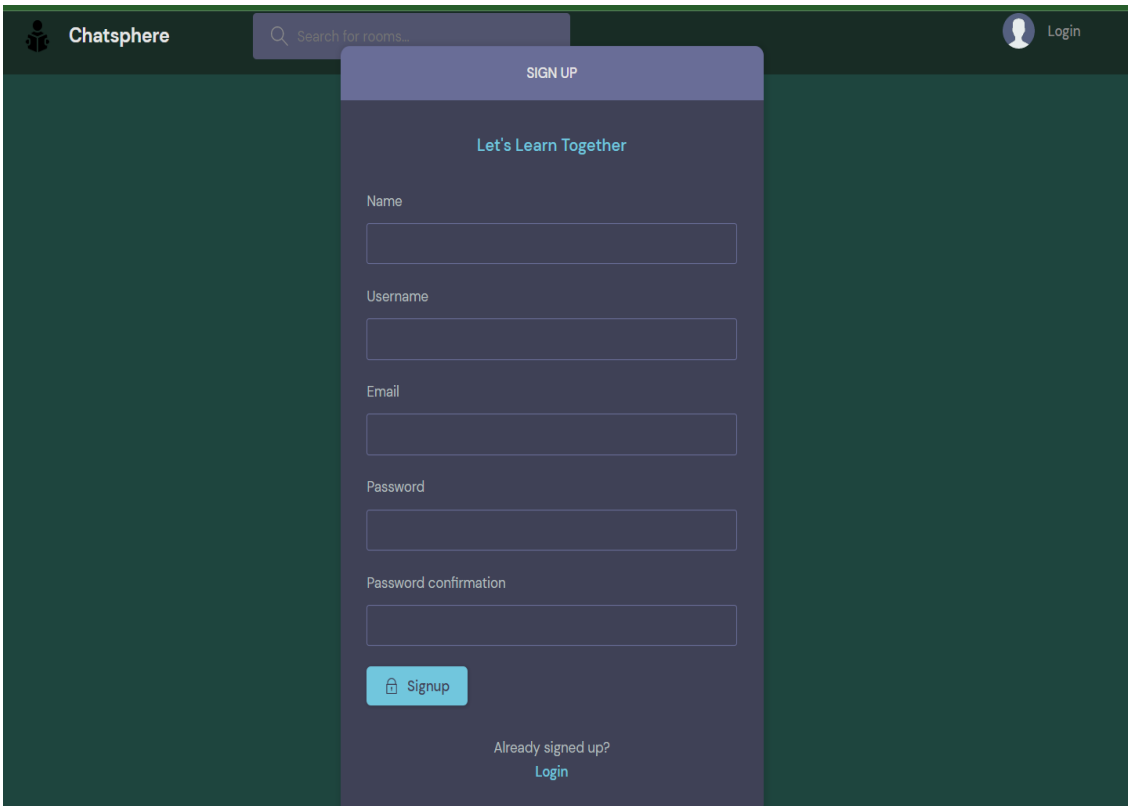
1. **Home Page**
   View before logging in. User can see the existing rooms but cannot join until they log in



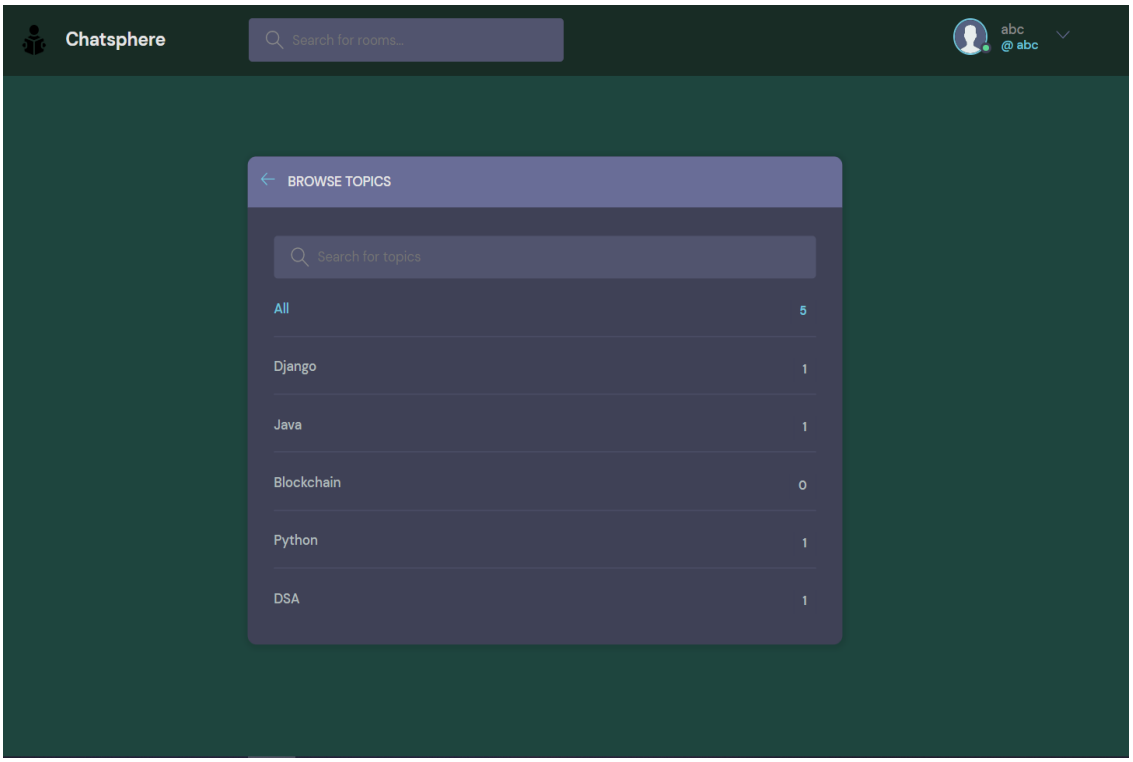
2. **Login Page**
   Where existing user can login to there accounts.



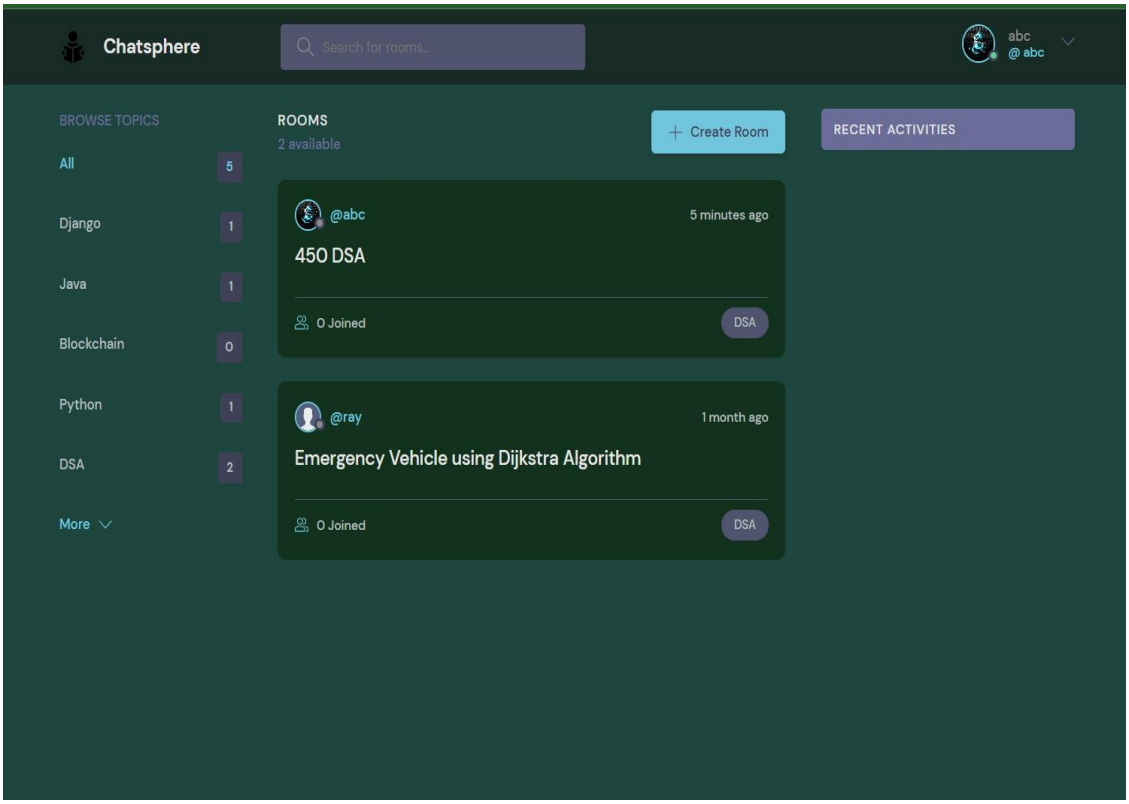
3. **Signup Page**
   Where a new user can enter his/her credentials and create a new account.



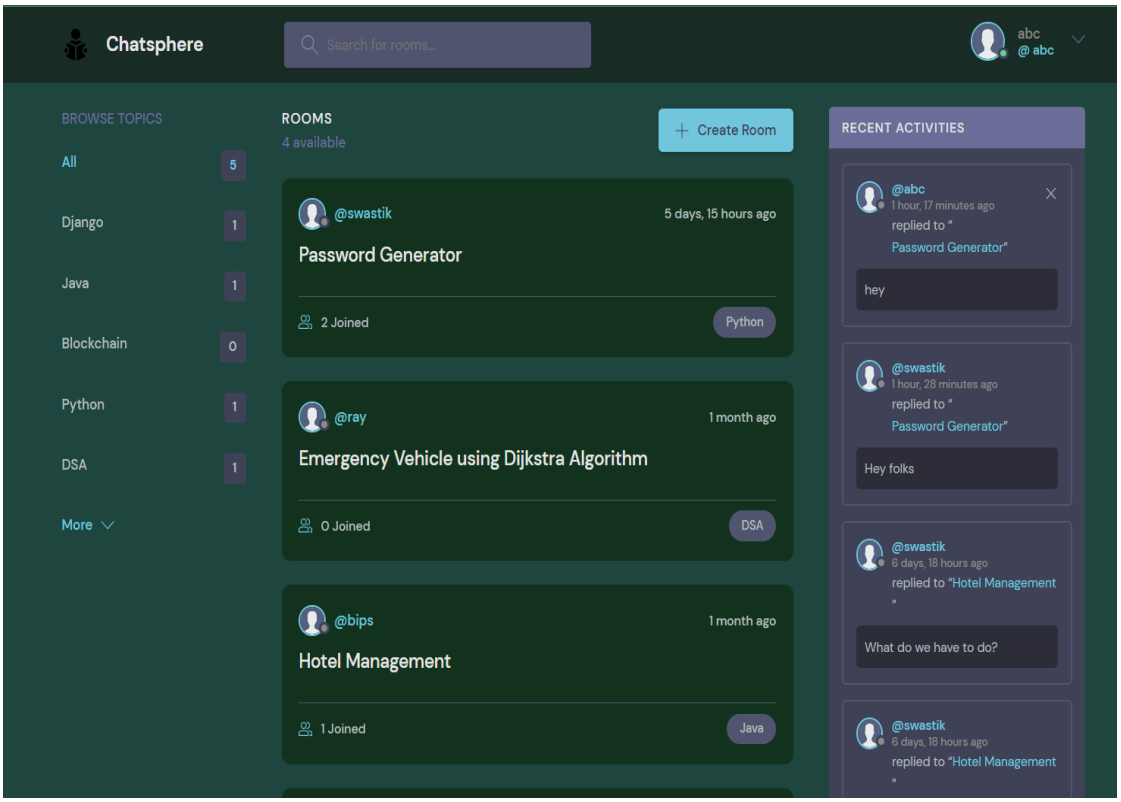
4. **Browser topic page**
   A search bar where an user can search groups by entering topics.



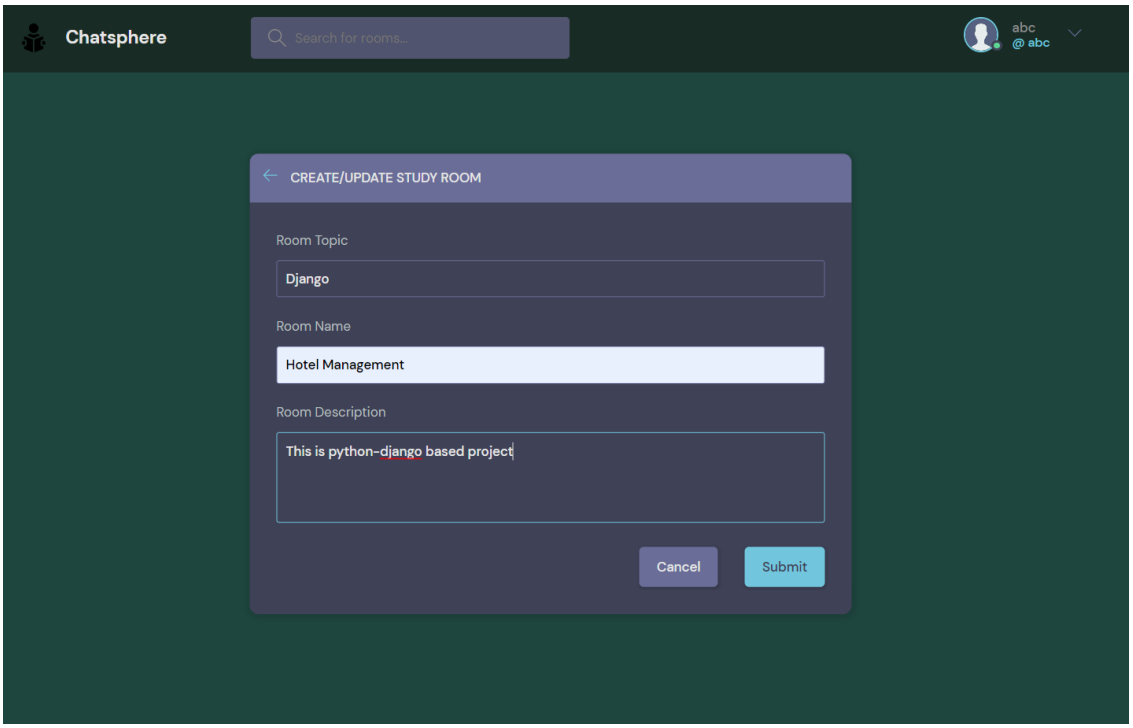
5. **From search option, we can separate**
   Sorted rooms on the basis of search.



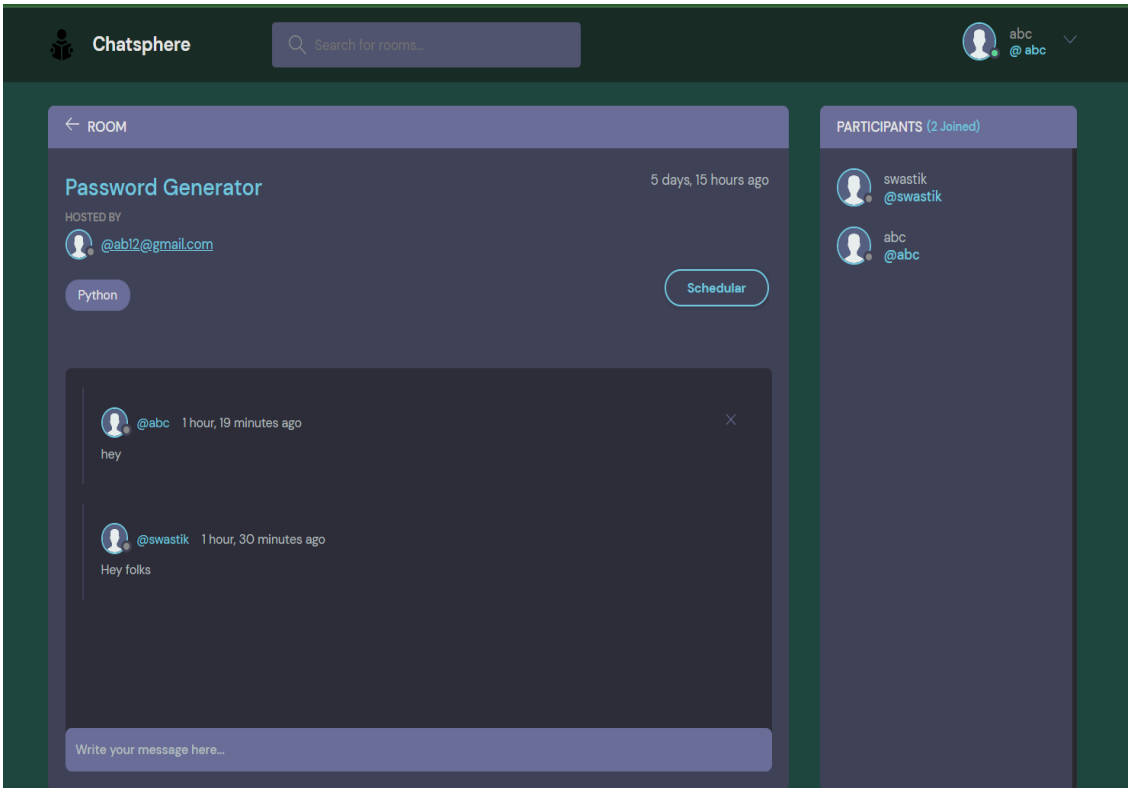
6. **After log in**
   User can see recent activities section and join room of his/her choice.



7. **Create new Room page**
   Page where user can create a new room by entering topics, room name, description
   .

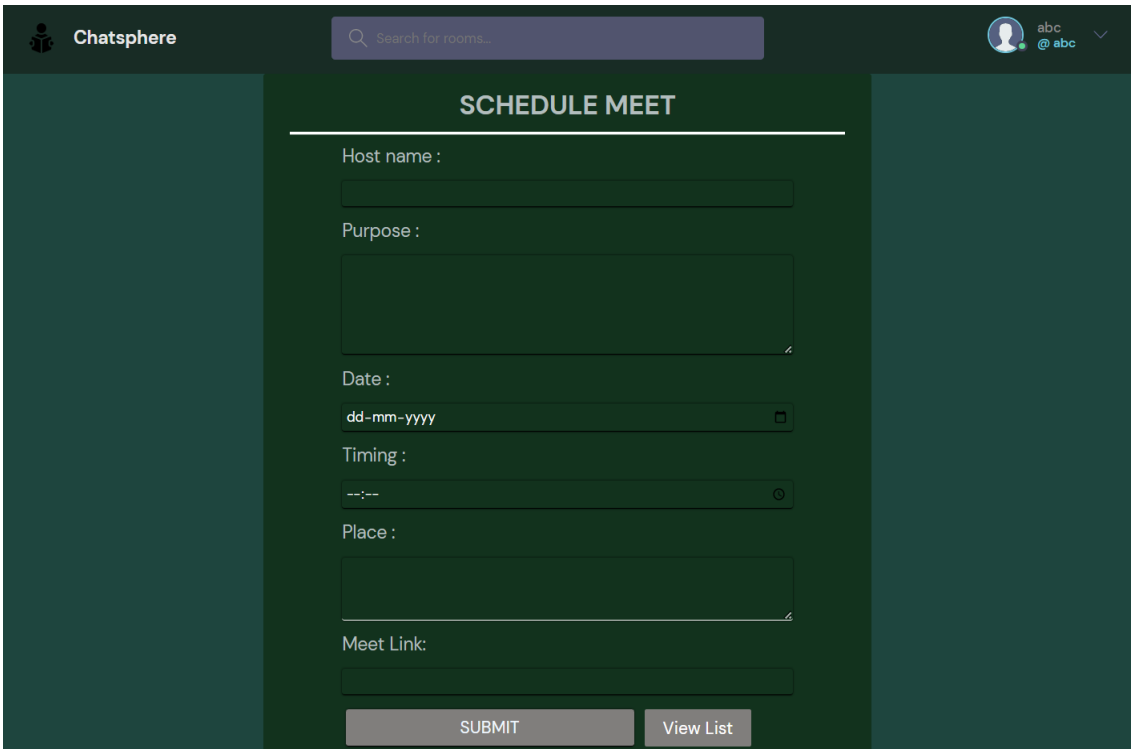


8. **Existing Room Page**
   Here a user can message discuss projects or can ask for guidance from his/her seniors or batchmates. It contains the list of participants



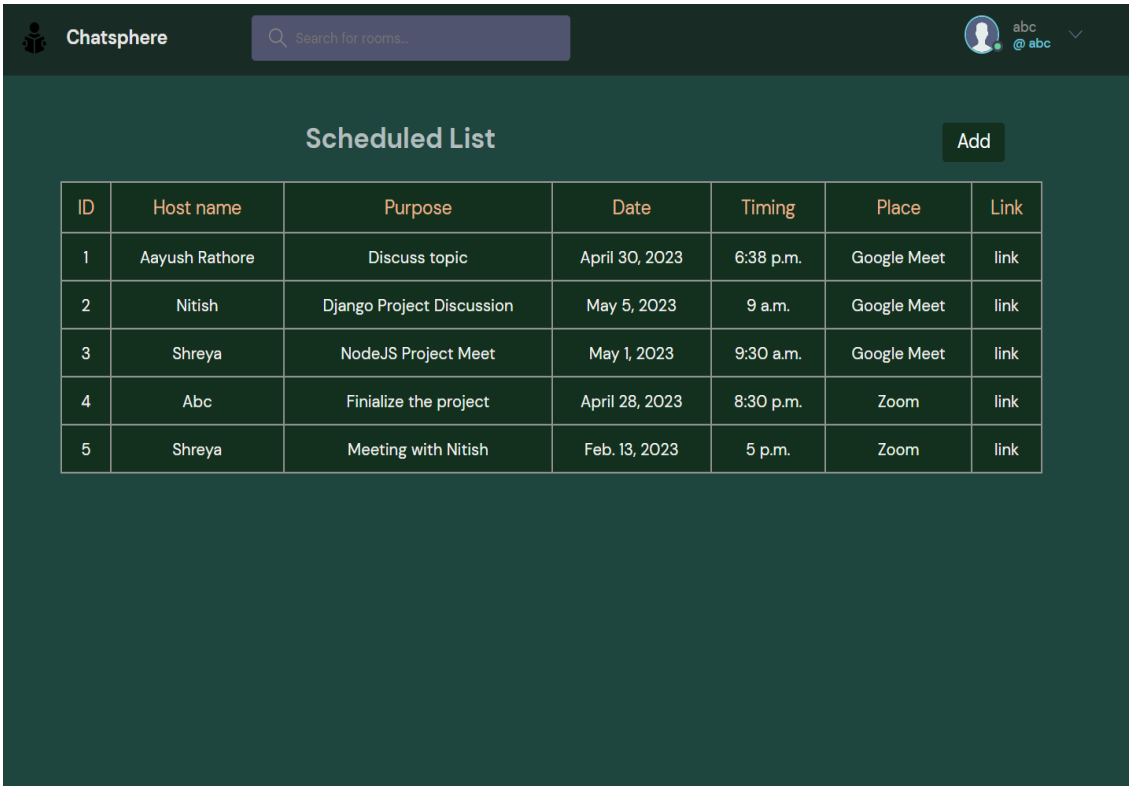
9. Schedule Meet Page

Here a user can enter details and schedule a meeting.



10. Schedule list Page

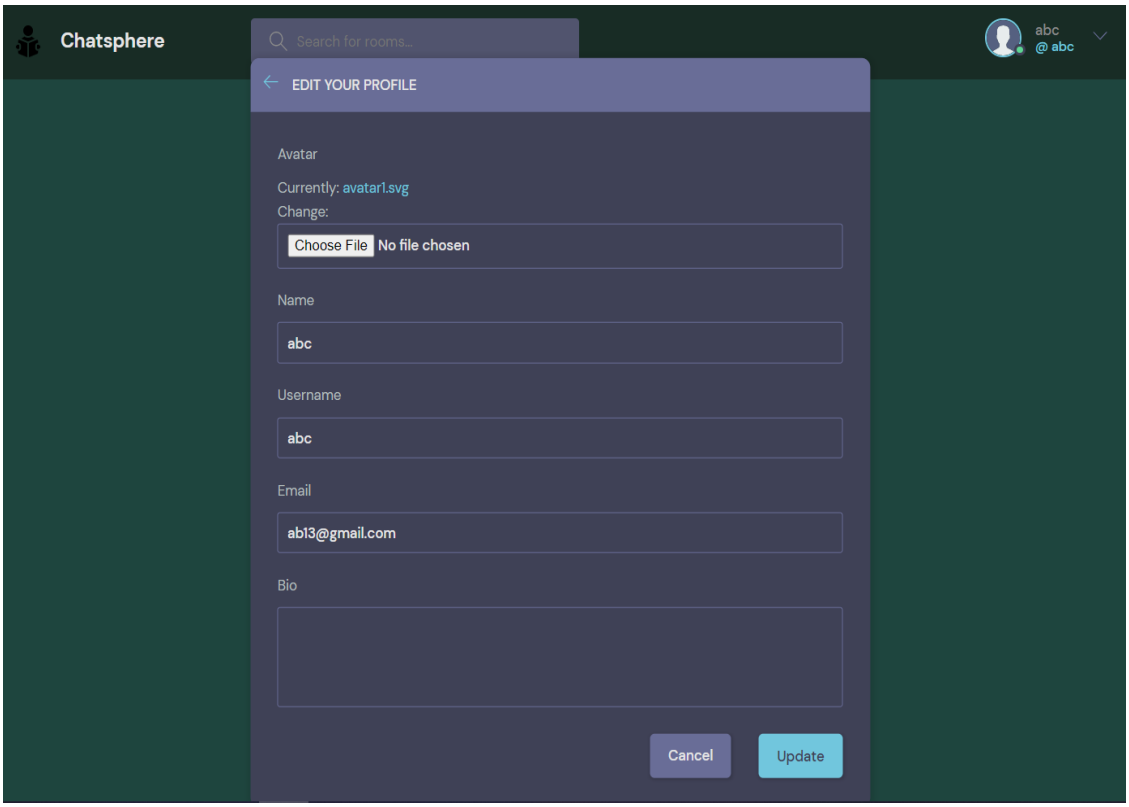
All the scheduled meet will be displayed on this page.



ID	Host name	Purpose	Date	Timing	Place	Link
1	Aayush Rathore	Discuss topic	April 30, 2023	6:38 p.m.	Google Meet	link
2	Nitish	Django Project Discussion	May 5, 2023	9 a.m.	Google Meet	link
3	Shreya	NodeJS Project Meet	May 1, 2023	9:30 a.m.	Google Meet	link
4	Abc	Finialize the project	April 28, 2023	8:30 p.m.	Zoom	link
5	Shreya	Meeting with Nitish	Feb. 13, 2023	5 p.m.	Zoom	link

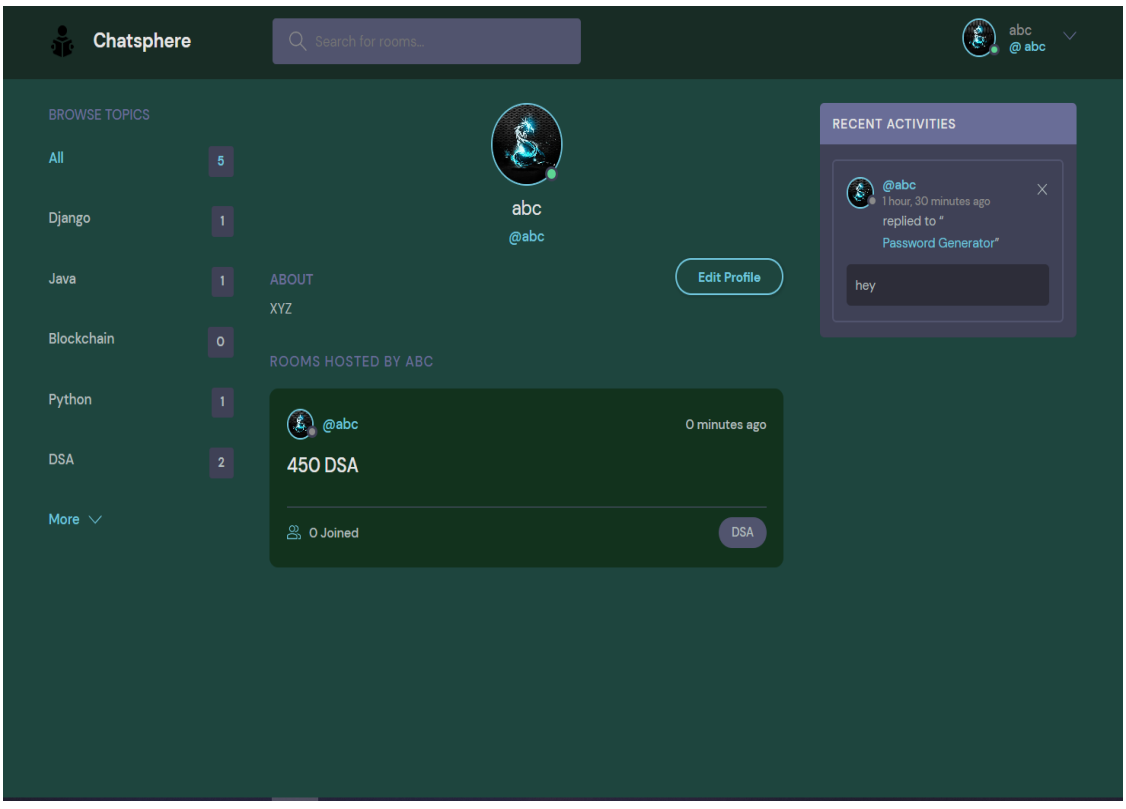
11. Edit Profile Page

Here a user can edit his/her profile.



12. User Profile Page

For viewing user profile.



CHAPTER-6

CONCLUSION AND FUTURE WORK

In conclusion, the Chat Discussion Room project using Django is a feature-rich and user-friendly platform that enables users to create, join and participate in discussions on a wide range of topics. The platform offers a range of features, including user authentication, profile creation, room creation/joining, topic browsing, recent activities viewing, and profile viewing. The use of Django as the web framework provides robust security, efficient database handling and easy scalability of the platform.

This project can be further enhanced with the addition of features such as private messaging, real-time chat using web sockets, notifications, and social media integration. Additionally, the platform can be improved by implementing responsive design, mobile app development, and SEO optimization.

Overall, the Chat Discussion Room project using Django has the potential to provide a valuable tool for groups, and organizations to connect and communicate effectively, making it a valuable addition to any online community.

REFERENCES

1. “Development of Chat Application.” n.d. IJRASET. Accessed April 24, 2023. https://www.ijraset.com/research-paper/development-of-chat-application.
2. “Django Tutorial.” 2023. GeeksforGeeks. https://www.geeksforgeeks.org/django-tutorial/.
3. guide, step. n.d. “Django Tutorial: Python Web Development.” DataCamp. Accessed April 24, 2023. https://www.datacamp.com/tutorial/web-development-django.
4. Roca, Claudia. n.d. “What is web programming and what is it for? (2023).” ThePowerMBA. Accessed April 24, 2023. https://www.thepowermba.com/en/blog/what-is-web-programming.
5. “The Six Biggest New Ideas In Chat.” 2006. TechCrunch. https://techcrunch.com/2006/11/24/the-six-biggest-new-ideas-in-chat/.