



**R. K. DESAI ACHCHHARIWALA COLLEGE OF COMPUTER
AND APPLIED SCIENCES, VAPI – 396195 (GUJARAT)**

PROJECT REPORT

on

“Chat Room”

*AS A PARTIAL REQUIREMENT
FOR THE DEGREE OF*

Bachelor Of Computer Application (B.C.A.)

Veer Narmad South Gujarat University

Submitted By:

Prajapati Henil Vinodbhai

Suryavanshi Shivam Vishav Mohan

Exam Seat No:

605

586

Guided By:

Prof. Neha Patel

**R.K.DESAI ACHCHHARIWALA COLLEGE OF COMPUTER AND
APPLIED SCIENCES, G.I.D.C. , VAPI – 396195 (GUJARAT)**

PROJECT CERTIFICATE

This is to certify that the project entitled

“ChatRoom”

Carried out by

*The partial fulfillment for the award of
Bachelor Of Computer Application
to the VeerNarmadSouthGujaratUniversity, Surat
is a work of require standard.*

Signature:

Guide By:

Prof. Neha Patel.

Date:

I/C Principal:

Dr.Mittal Shah

Date:

Examiner:

Date:



Chat Room

Group Chat & ChatGPT

Acknowledgement

I would like to take this opportunity to express my sincere gratitude and appreciation to everyone who contributed to the successful completion of the "Chatroom" project.

First and foremost, I would like to thank our project supervisor for providing us with guidance and support throughout the entire project. Their insights and expertise were invaluable in shaping the direction of our work.

I would also like to extend my heartfelt thanks to all the members of our team for their hard work and dedication. Each member brought unique skills and perspectives to the project, and their contributions were essential to its success.

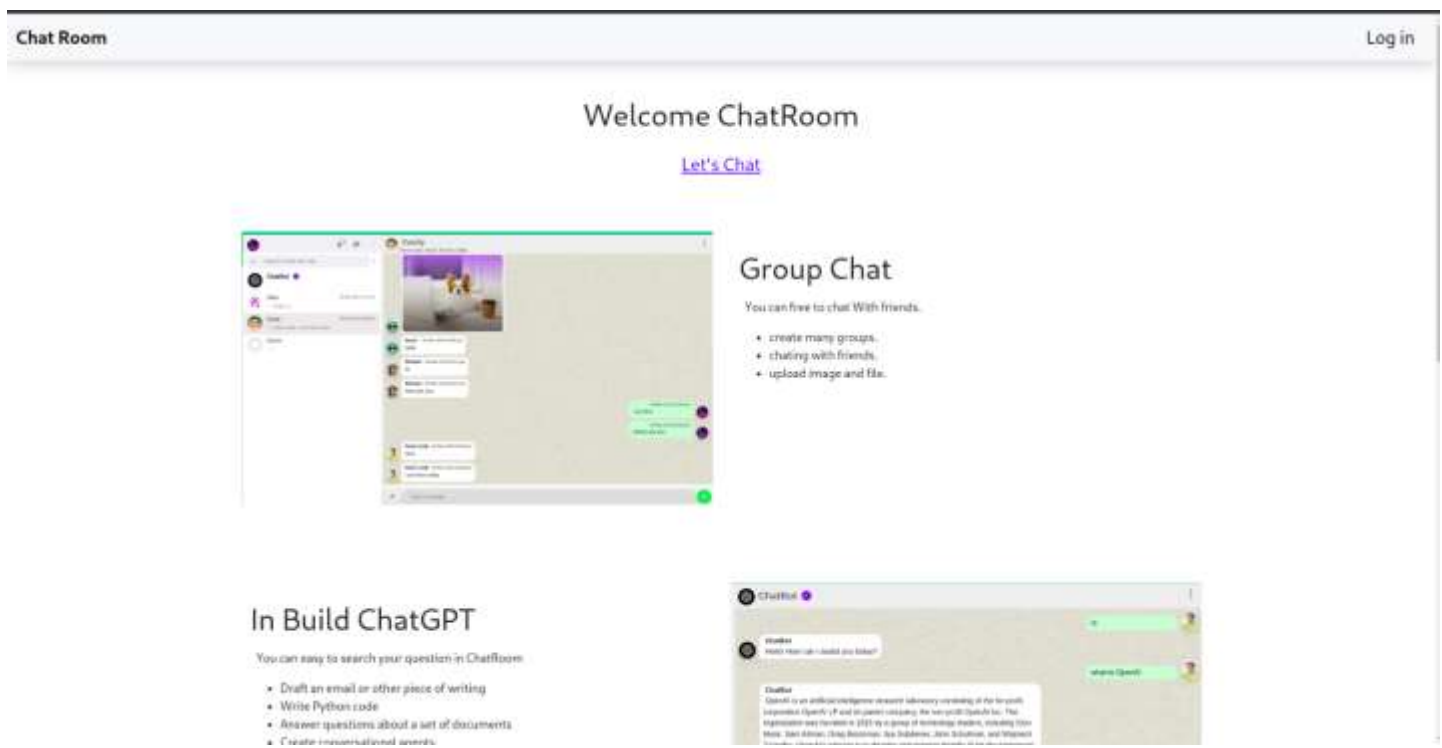
Index

No	Title	Page No
1	Introduction	1
2	Project Details	2
3	Database Dictionary	4
4	Flow Chart	9
5	Screen Shot & details	14
6	Conclusion	43
7	References	44

Introduction

The Chatroom project is a web-based chat application developed using **Laravel**, a PHP web application framework, **CSS**, and **jQuery**, a JavaScript library. This project aims to provide users with a communication platform that allows them to exchange messages, files and communicate with one another. The project features a user-friendly interface that allows users to create accounts, join chat rooms, and send messages to one another.

Project Overview the Chatroom project was developed to provide users with an online platform to communicate with one another. The project has two main components: the front-end and the back-end. The front-end is developed using jQuery, CSS and is responsible for rendering the user interface. The back-end is developed using Laravel and is responsible for handling user authentication, data storage, and communication between users.



Project Details:

Project Features:

- **User registration and login:** Users can create accounts and log in to the application using their credentials.
- **Chat rooms:** Users can create chat rooms and invite other users to join them.
- **messaging:** Users can send and receive messages or files.
- **User profiles:** Users can view their profiles and edit their information.
- **Notification system:** Users receive notifications when they are invited to join a chat room.
- **Chatbot:** To integrate ChatGPT and DELL-2 into the Chatroom project, APIs provided by OpenAI. user can ask questions and find images.
- **Admin Dashboard:** provide admin dashboard it uses admin can manage site. It can be changing user permission.

Technology Used:

The Chatroom project uses the following technology used:

- **Laravel:** A PHP web application framework used to develop the back-end of the project and version **9.52**.
- **jQuery:** A JavaScript library used to develop the front-end of the project.
- **MySQL:** A database management system used to store user information and chat messages.
- **CSS:** CSS is a styling language used for describing the presentation of a document written in HTML.
- **Bootstrap:** Bootstrap is a front-end framework used for designing responsive and mobile-first web applications.

▪ **Laravel:**

Laravel is a free, open-source PHP web framework designed for building modern web applications. It was created by Taylor Otwell in 2011 and has since become one of the most popular PHP frameworks, known for its elegant syntax, expressive and intuitive syntax, and robust features. Laravel follows the Model-View-Controller (MVC) architectural pattern and provides many built-in tools and features such as routing, authentication, caching, and database migration that make it easy to build scalable and maintainable web applications. Laravel also has a large and active community of developers who contribute to its growth and provide support to other developers through forums, blogs, and social media channels.

▪ **MySQL:**

MySQL is a popular open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) to manage and manipulate data. It was first released in 1995 and is now owned by Oracle Corporation. MySQL is widely used in web applications as a backend database because of its scalability, reliability, and ease of use. MySQL supports many features such as multiple storage engines, transactions, indexes, and user-defined functions. It is also known for its performance, stability, and security features. MySQL is compatible with many programming languages and can be used with various operating systems, including Windows, Linux, and macOS

Architecture:

The Chatroom application is built using the Model-View-Controller (MVC) architecture pattern. The application uses Laravel as the back-end framework and jQuery for real-time communication between the server and the client. The front-end is designed using CSS and Bootstrap.

Database:

The application uses MySQL as the database. The database schema consists of tables for users, rooms, messages, room_user, notification, etc.

Conclusion:

The Chatroom is a communication platform that allows users to chat with each other. The application is built using Laravel, jQuery, CSS, and Bootstrap. The application allows users to create accounts, log in, and communicate with other users. The application also includes features like user status, user profiles, chatroom creation, and moderation.

Database Dictionary




→ User Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 🔑	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	name 🔑	varchar(255)	utf8mb4_unicode_ci		No	None		
3	email 🔑	varchar(255)	utf8mb4_unicode_ci		No	None		
4	photo	varchar(255)	utf8mb4_unicode_ci		No	default.jpg		
5	email_verified_at	timestamp			Yes	NULL		
6	password	varchar(255)	utf8mb4_unicode_ci		No	None		
7	role	enum('admin', 'user', 'suspend')	utf8mb4_unicode_ci		No	user		
8	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL		
9	created_at	timestamp			Yes	NULL		
10	updated_at	timestamp			Yes	NULL		




→ Room Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 🔑	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_unicode_ci		No	None		
3	photo	varchar(255)	utf8mb4_unicode_ci		No	default.jpg		
4	created_by 🔑	bigint(20)		UNSIGNED	No	None		
5	created_at	timestamp			Yes	NULL		
6	updated_at	timestamp			Yes	NULL		



→ Room_User Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	room_id 	bigint(20)		UNSIGNED	No	None		
3	user_id 	bigint(20)		UNSIGNED	No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		




→ Message Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	chat	text	utf8mb4_unicode_ci		No	None		
3	room_id 	bigint(20)		UNSIGNED	No	None		
4	user_id 	bigint(20)		UNSIGNED	No	None		
5	created_at	timestamp			Yes	NULL		
6	updated_at	timestamp			Yes	NULL		

→ **Files Table:**

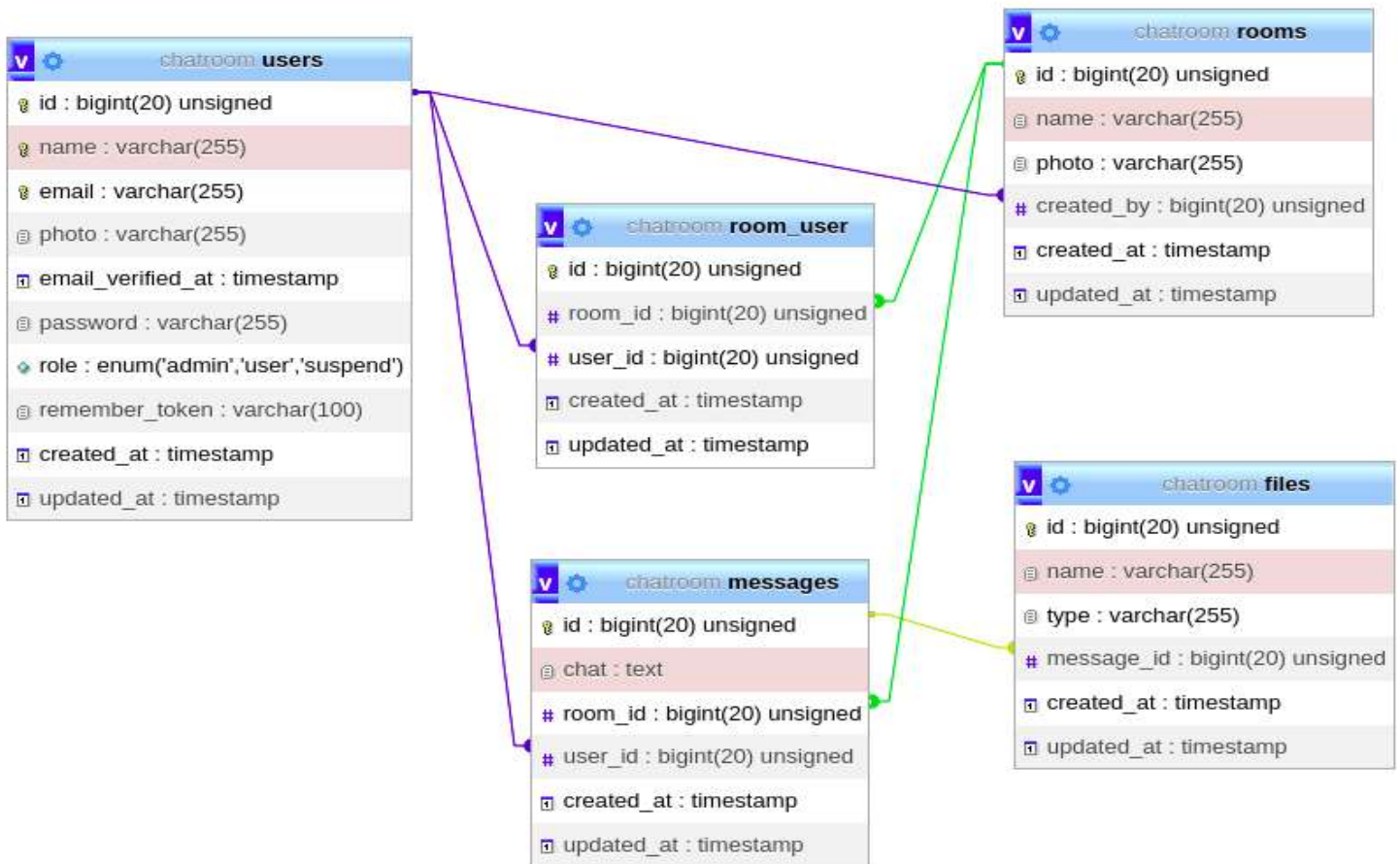
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_unicode_ci		No	None		
3	type	varchar(255)	utf8mb4_unicode_ci		No	None		
4	message_id 	bigint(20)		UNSIGNED	No	None		
5	created_at	timestamp			Yes	NULL		
6	updated_at	timestamp			Yes	NULL		

→ **Notification Table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	char(36)	utf8mb4_unicode_ci		No	None		
2	type	varchar(255)	utf8mb4_unicode_ci		No	None		
3	notifiable_type 	varchar(255)	utf8mb4_unicode_ci		No	None		
4	notifiable_id 	bigint(20)		UNSIGNED	No	None		
5	data	text	utf8mb4_unicode_ci		No	None		
6	read_at	timestamp			Yes	NULL		
7	created_at	timestamp			Yes	NULL		
8	updated_at	timestamp			Yes	NULL		

➤ Database:

- ❖ Following database use for chatroom data management like message, files and rooms.



- ❖ The following tables are used for sending mails using the jobs table, notifications and that database table managed by Laravel.

chatroom jobs
id : bigint(20) unsigned
queue : varchar(255)
payload : longtext
attempts : tinyint(3) unsigned
reserved_at : int(10) unsigned
available_at : int(10) unsigned
created_at : int(10) unsigned

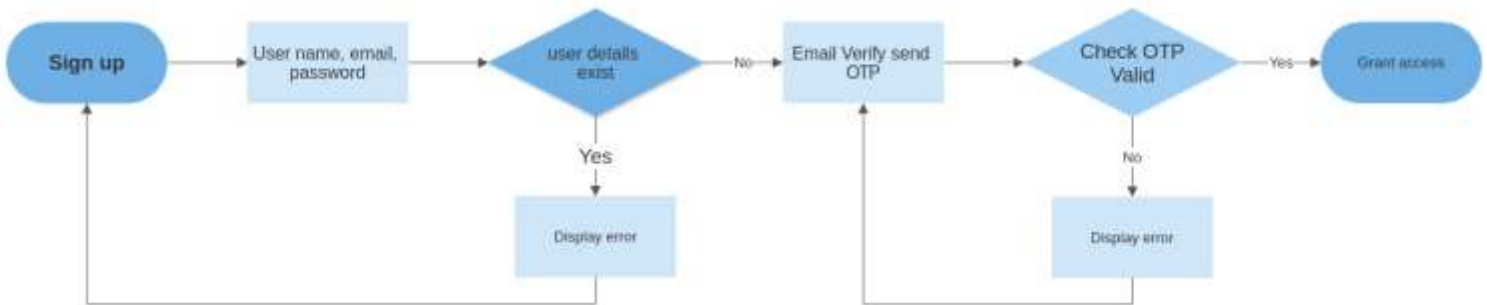
chatroom failed_jobs
id : bigint(20) unsigned
uuid : varchar(255)
connection : text
queue : text
payload : longtext
exception : longtext
failed_at : timestamp

chatroom notifications
id : char(36)
type : varchar(255)
notifiable_type : varchar(255)
notifiable_id : bigint(20) unsigned
data : text
read_at : timestamp
created_at : timestamp
updated_at : timestamp

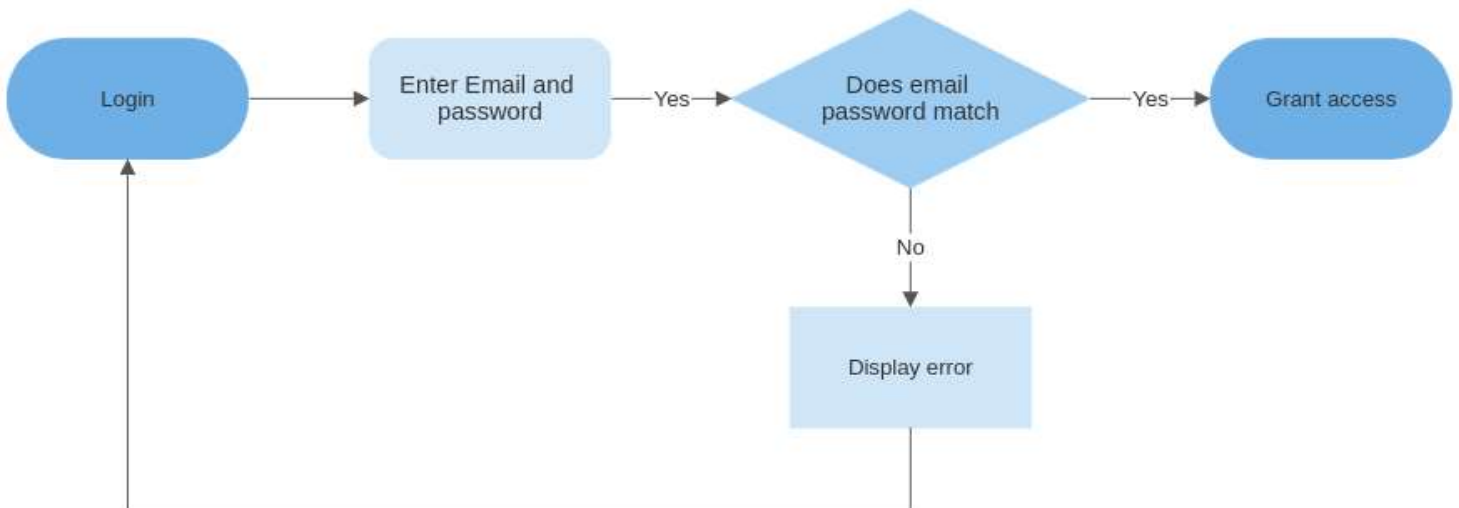
chatroom migrations
id : int(10) unsigned
migration : varchar(255)
batch : int(11)

➤ Flow Chart

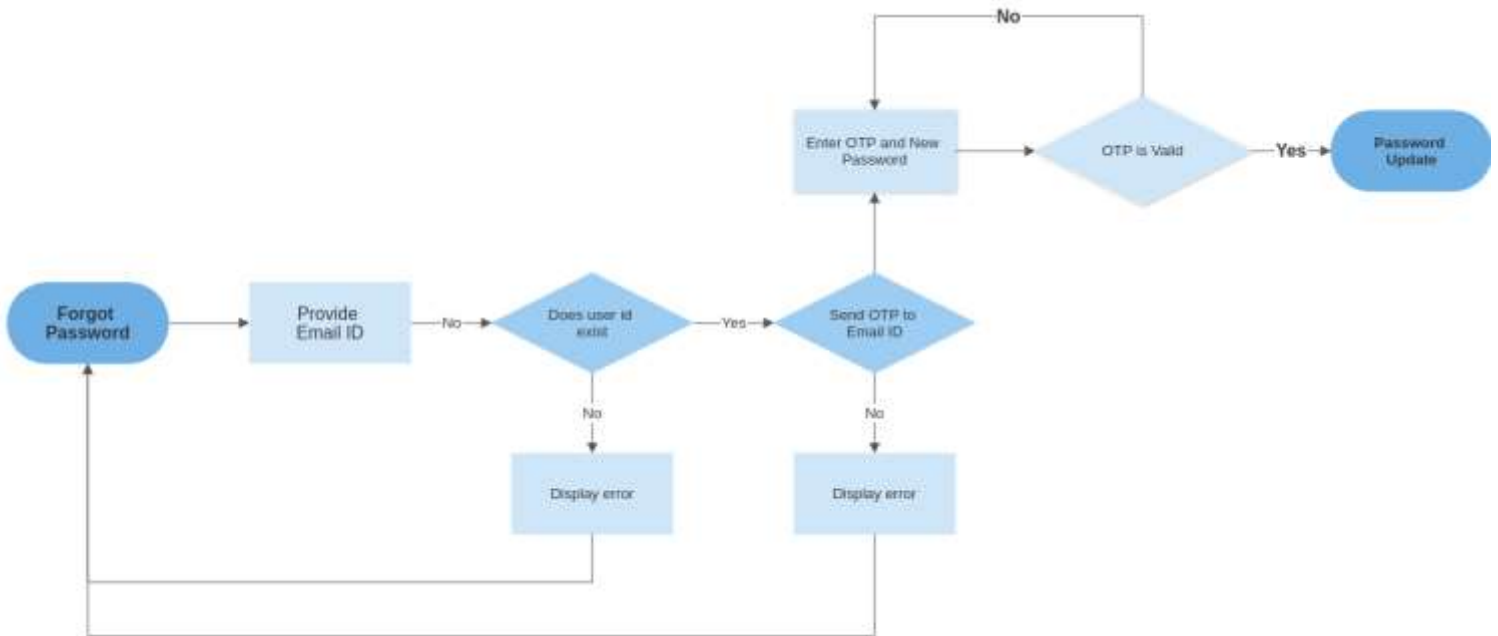
- Sign Up:



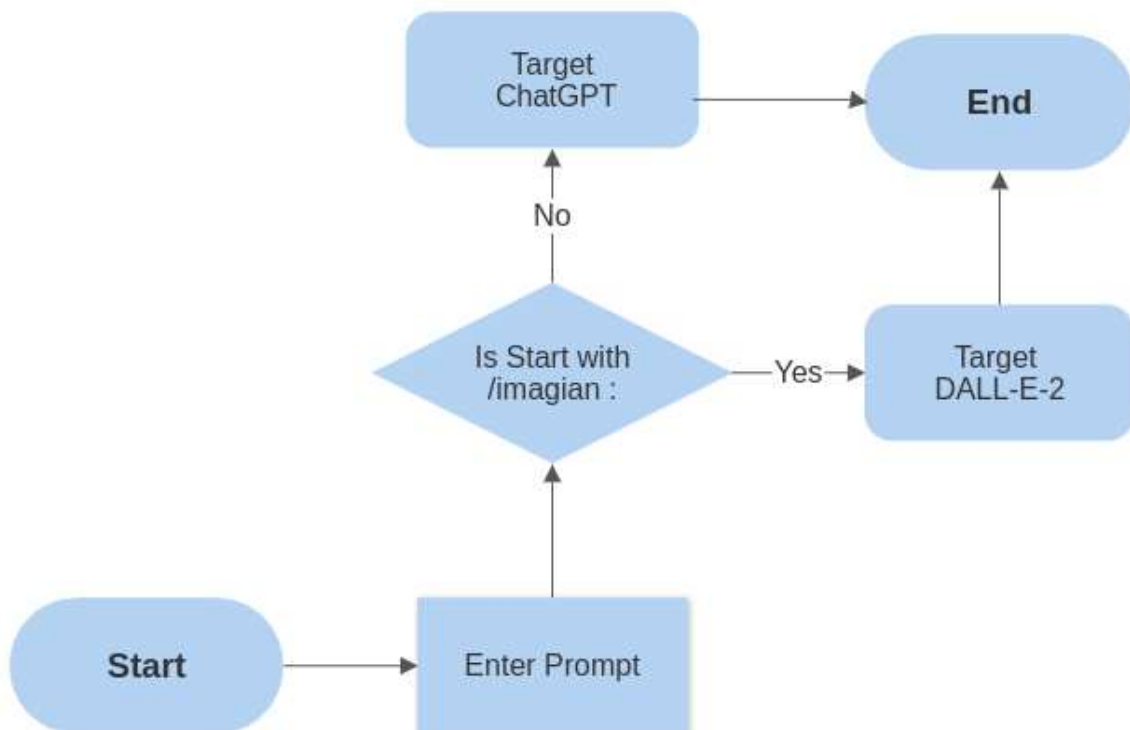
- Login:



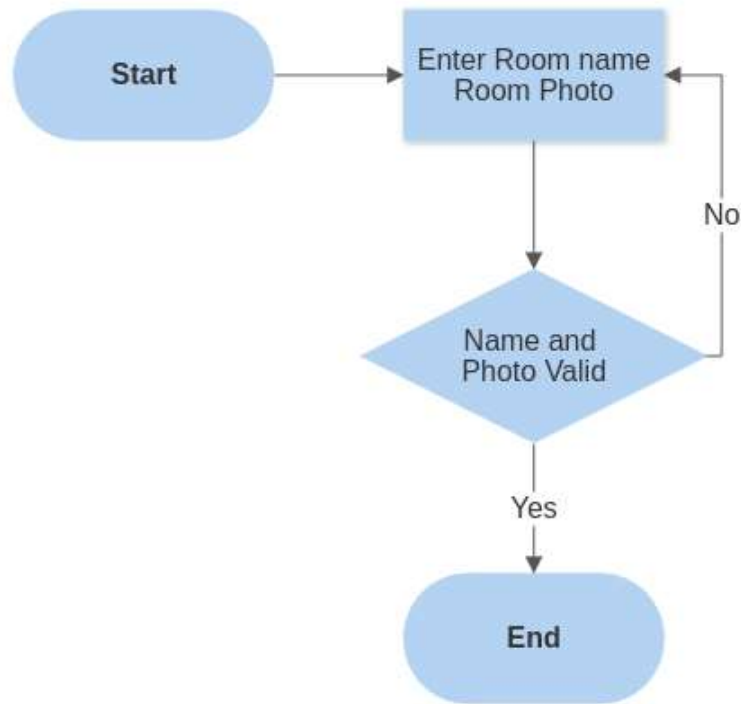
- Forgot Password:



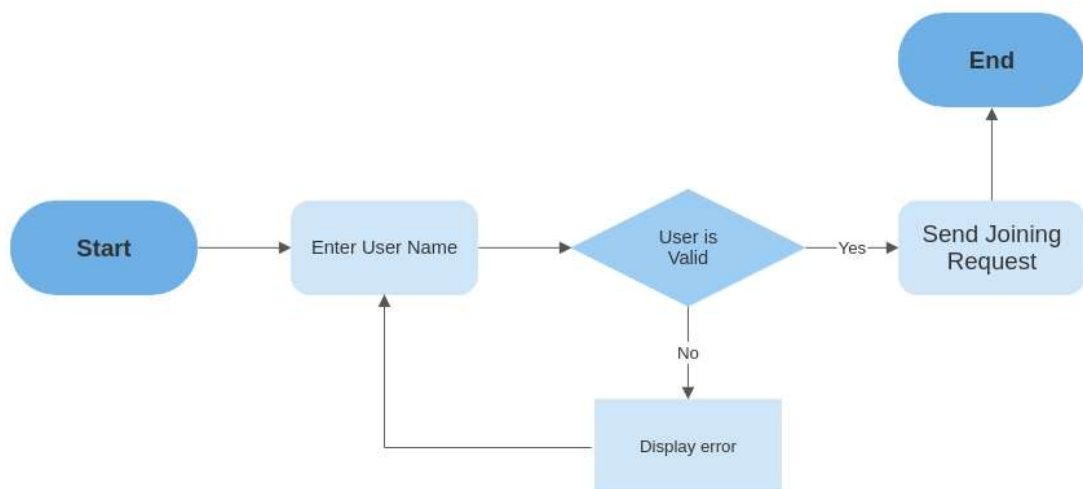
- Chat Bot:



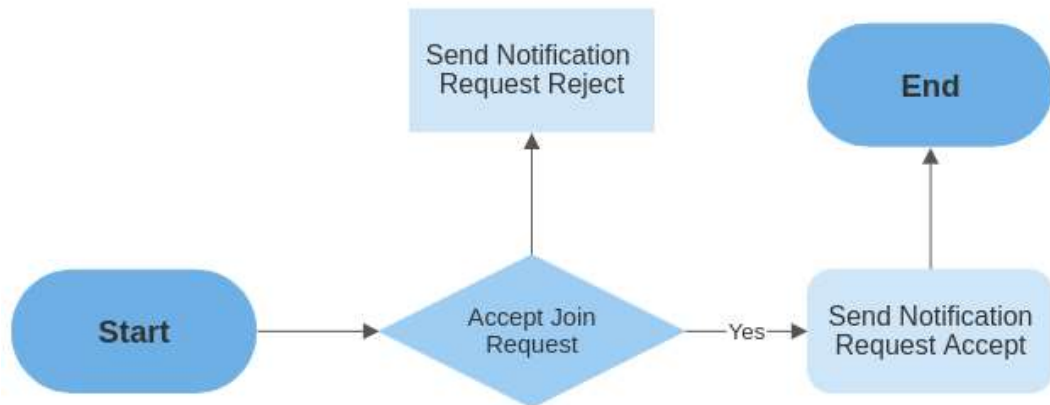
- Create Room:



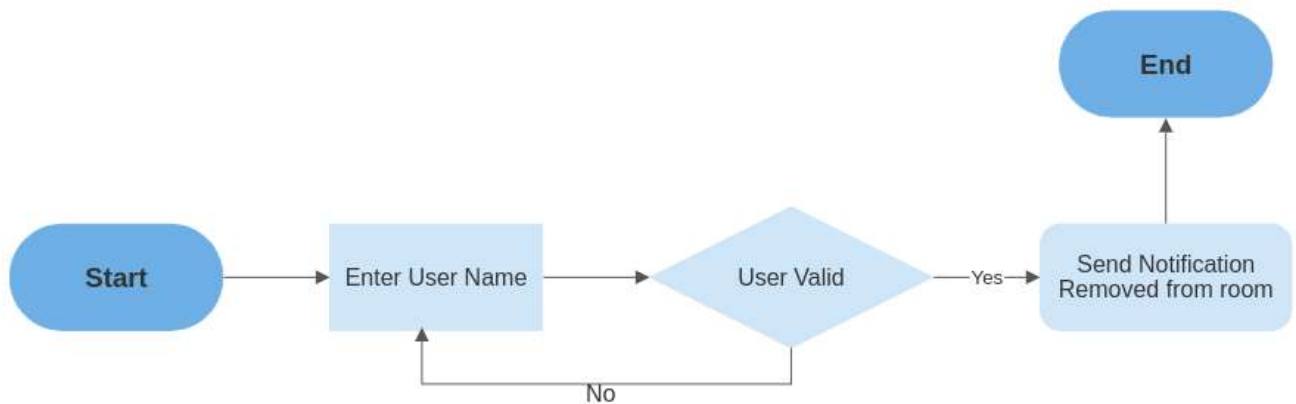
- Add Member to Room:



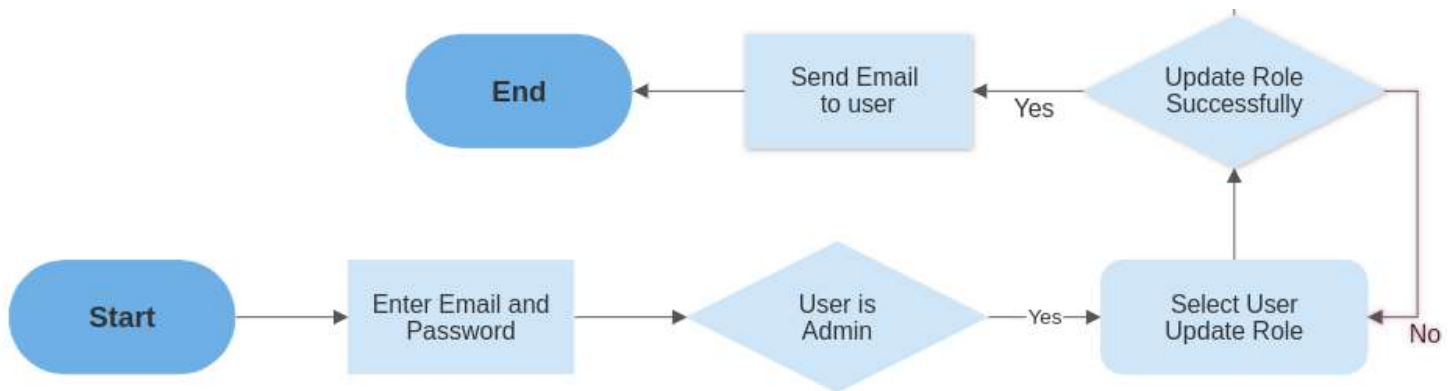
- Accept/Reject Joining Room Request:



- Remove Member to Room:



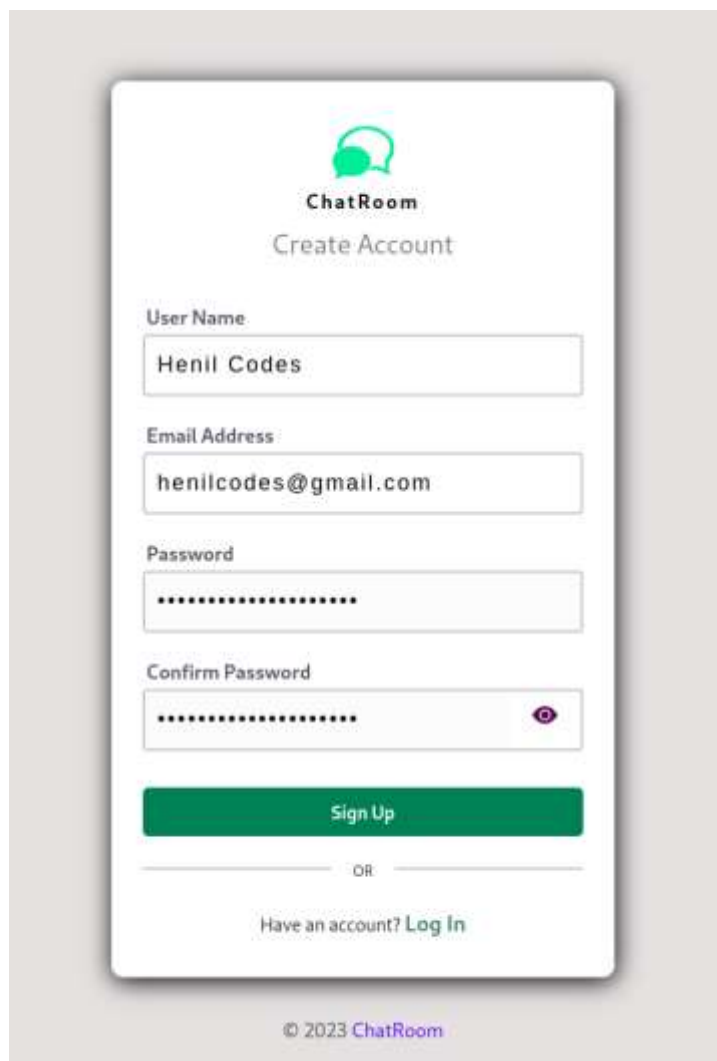
- Change User Role:



Screen Shot

Sign Up Page:

The Sign-Up page is a web page that allows users to create accounts to access the features of a website or application.



The screenshot displays the 'Create Account' page for ChatRoom. At the top, there is a green speech bubble icon with the text 'ChatRoom' and 'Create Account' below it. The form consists of four input fields: 'User Name' with the value 'Henil Codes', 'Email Address' with the value 'henilcodes@gmail.com', 'Password' with masked characters, and 'Confirm Password' with masked characters and a toggle eye icon. A green 'Sign Up' button is positioned below the password fields. Below the button is a horizontal line with 'OR' in the center. At the bottom of the form, there is a link that says 'Have an account? Log In'. The footer of the page shows '© 2023 ChatRoom'.

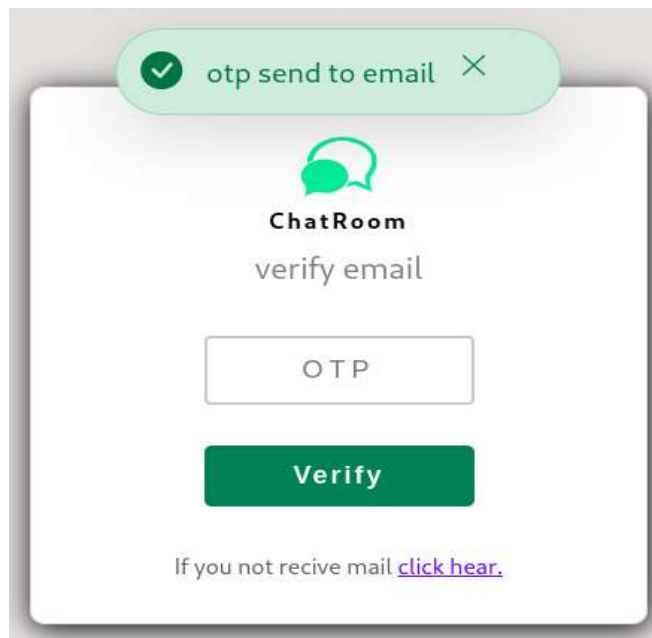
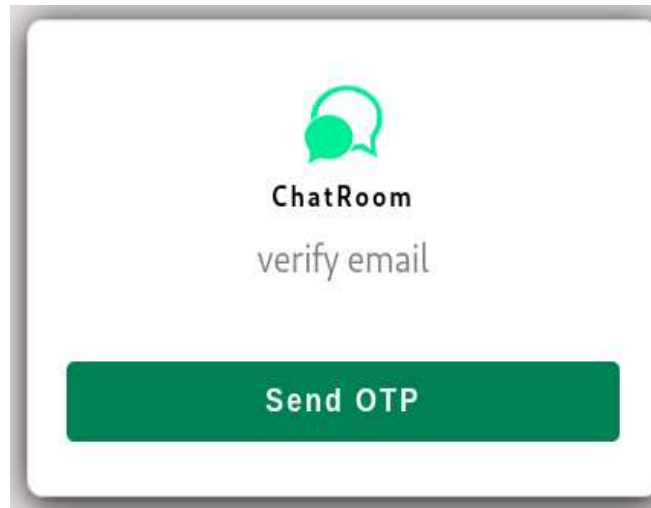
Components:

Registration Form: This is the main component of the Sign-Up page. It contains fields where users can enter their information, such as name, email address, username, and password. The form should include appropriate validation for each field, such as checking for the correct format of email addresses and password strength.

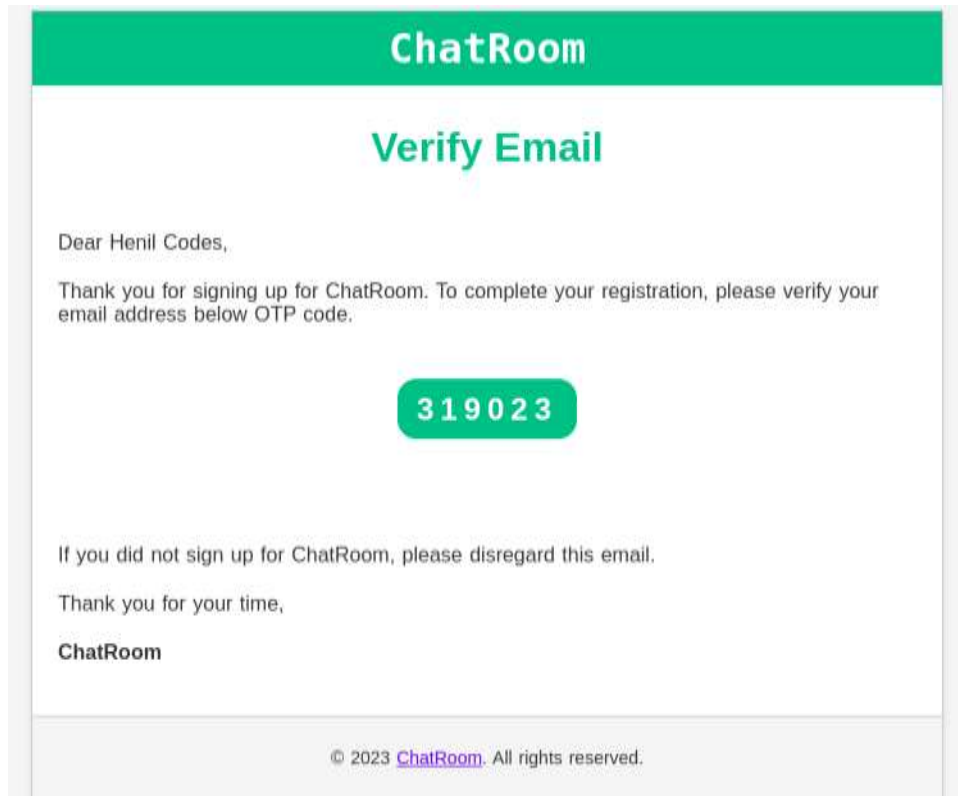
Functionality:

1. **User Authentication:** The Sign-Up page should validate the user's information and create a new account if the information is valid. The user's information should be stored securely in a database and protected from unauthorized access.
2. **Email Verification:** To confirm the user's email address, the website or application should send a verification email to the email address provided during the Sign-Up process. The user must enter OTP email to verify their email address and activate their account.
3. **Error Handling:** The Sign-Up page should provide clear error messages if the user enters incorrect or incomplete information. The error messages should be displayed next to the field with the error, and a general error message should be displayed at the top of the page.
4. **Redirect:** Once the user has successfully created an account, the Sign-Up page should redirect the user to the website or application's home page or to a page where they can complete their profile.

→ After Sign up verify email account:



- Users can receive OTP on mail.



➤ Login Page:

The Login page is a web page that allows users to access the features of a website or application by logging in with their credentials. We will describe the key components and functionality of a Login page.

ChatRoom

Welcome back!

Email Address

henilcodes@gmail.com

Password

.....

Log in

OR

Don't have an account? [Sign up](#)

Forgot Your Password?

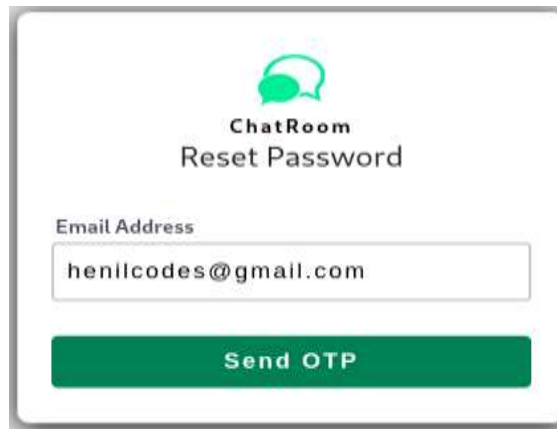
Components:

1. **Login Form:** This is the main component of the Login page. It contains fields where users can enter their login credentials, such as username or email address and password. The form should include appropriate validation for each field, such as checking for the correct format of email addresses and password strength.
2. **Forgot Password:** A Forgot Password link should be provided on the Login page. This allows users who have forgotten their password to reset it by providing their email address and sending OTP to email for verify user.

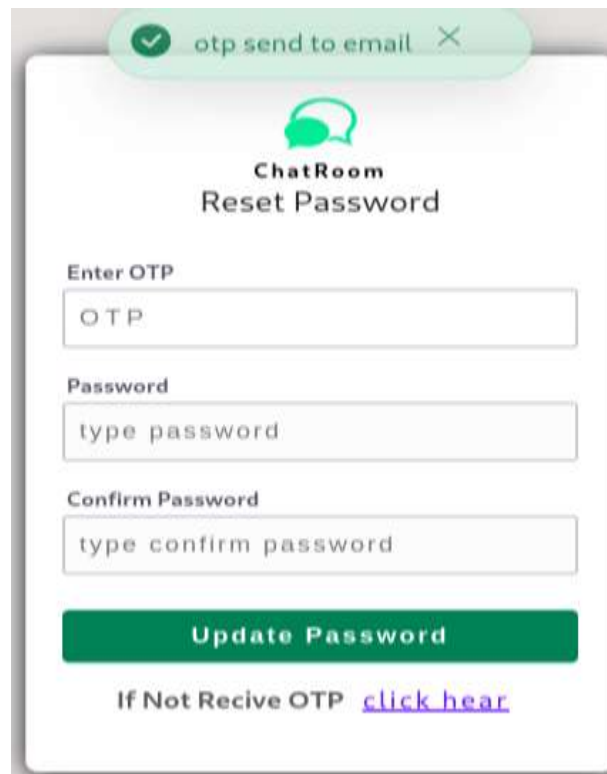
Functionality:

1. **User Authentication:** The Login page should validate the user's login credentials and allow access to their account if the credentials are correct. The user's information should be stored securely in a database and protected from unauthorized access.
2. **Error Handling:** The Login page should provide clear error messages if the user enters incorrect login credentials. The error messages should be displayed next to the field with the error, and a general error message should be displayed at the top of the page.
3. **Redirect:** Once the user has successfully logged in, the Login page should redirect the user to the website or application's home page or to the page they were trying to access before they were prompted to log in.
4. **Session Management:** After the user has successfully logged in, the website should create a session for the user. The session should be used to keep the user logged in as they navigate through the website or application. The session should expire after a certain period of inactivity or when the user logs out.

→ **Forgot Password:**

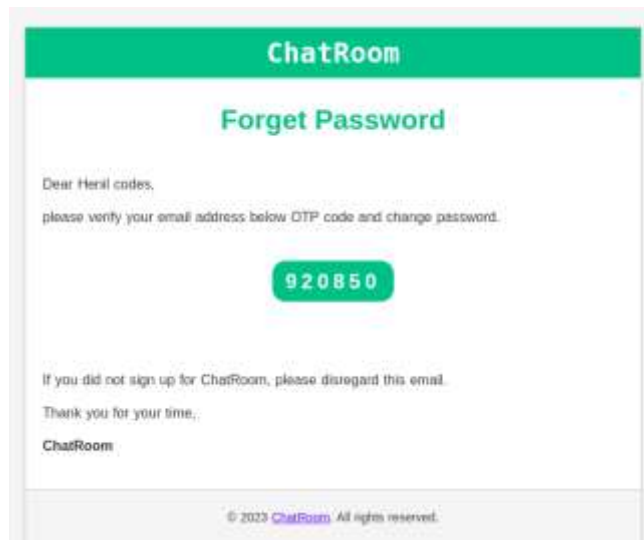


The image shows a mobile app screen for resetting a password. At the top, there is a green speech bubble icon with the text "ChatRoom" and "Reset Password" below it. Below the title, there is a label "Email Address" and a text input field containing "henilcodes@gmail.com". At the bottom, there is a green button labeled "Send OTP".



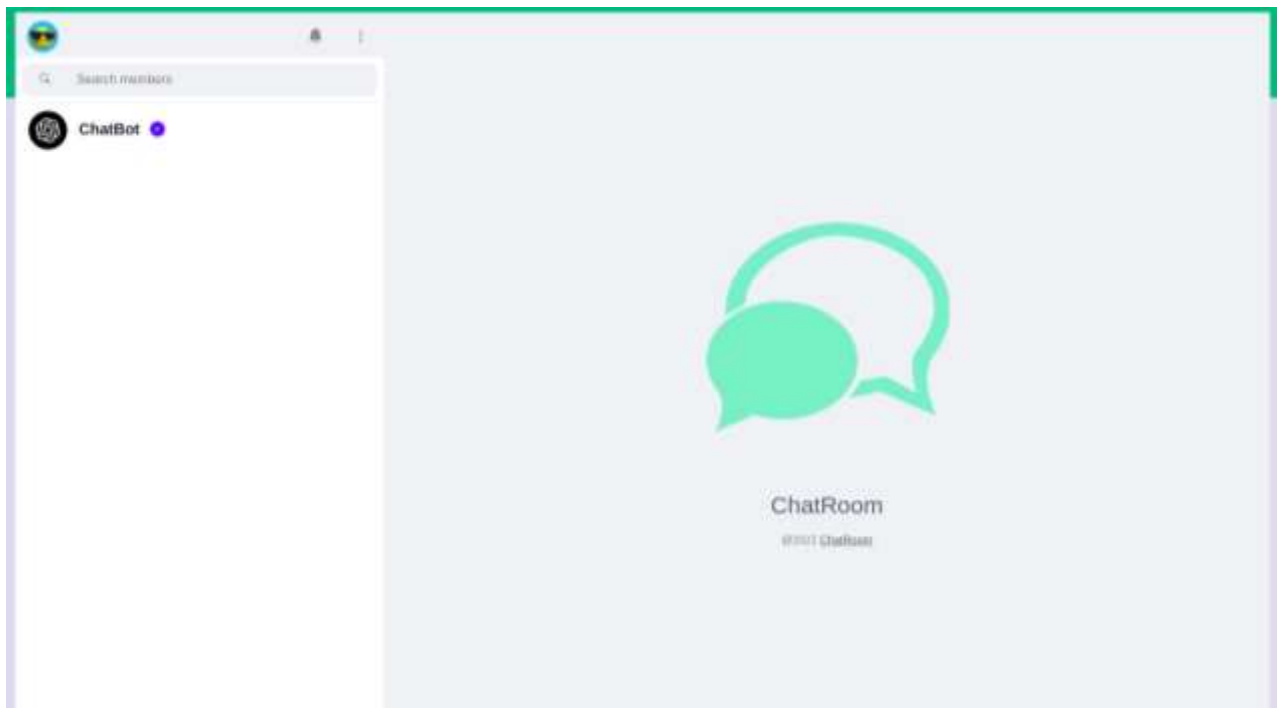
The image shows the next step in the password reset process. At the top, there is a green notification bar with a checkmark icon, the text "otp send to email", and a close button (X). Below the notification bar, there is a green speech bubble icon with the text "ChatRoom" and "Reset Password" below it. Below the title, there are three input fields: "Enter OTP" with a placeholder "OTP", "Password" with a placeholder "type password", and "Confirm Password" with a placeholder "type confirm password". At the bottom, there is a green button labeled "Update Password". Below the button, there is a link that says "If Not Recive OTP [click hear](#)".

- Users receive OTP on mail.



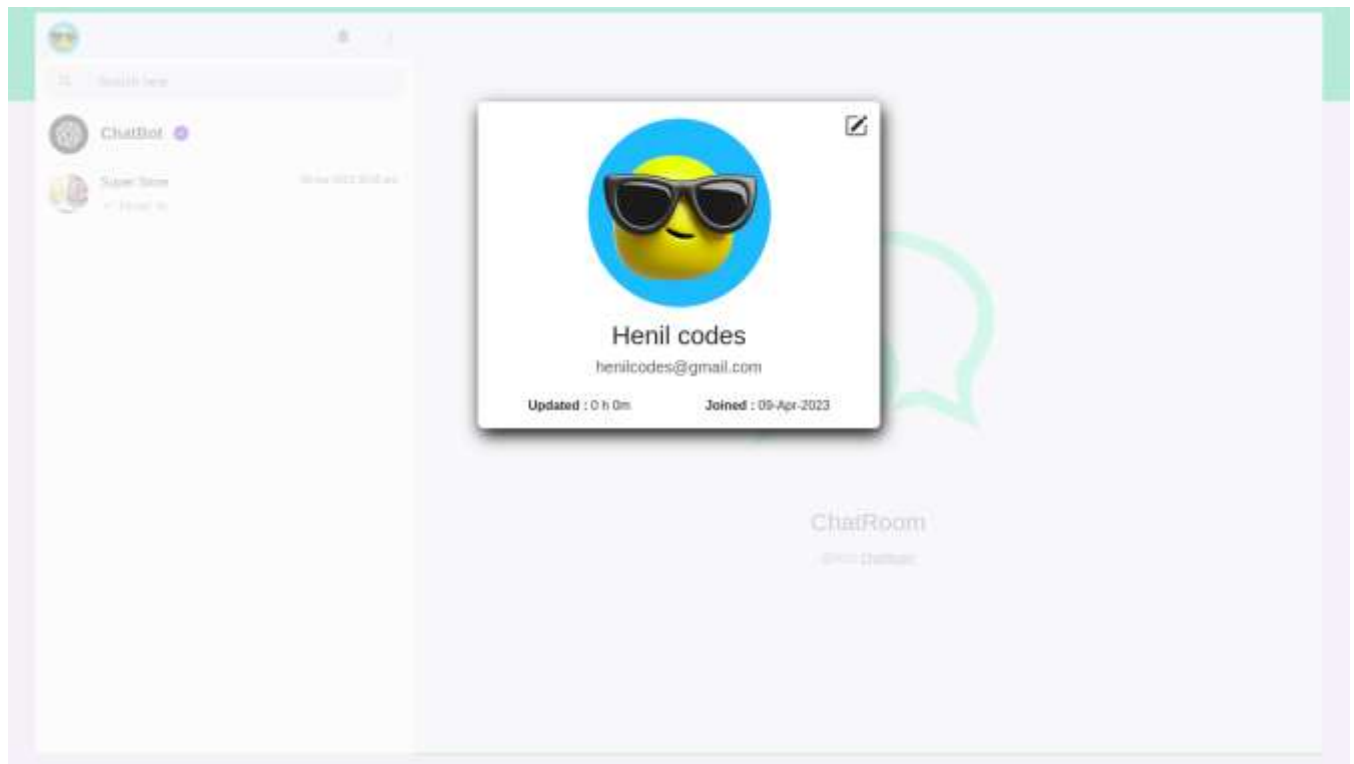
➤ User Side:

- After verifying email, the user redirects to chatroom. Chatroom in by default provide personal assistant as name Chatbot and provide default profile photo.



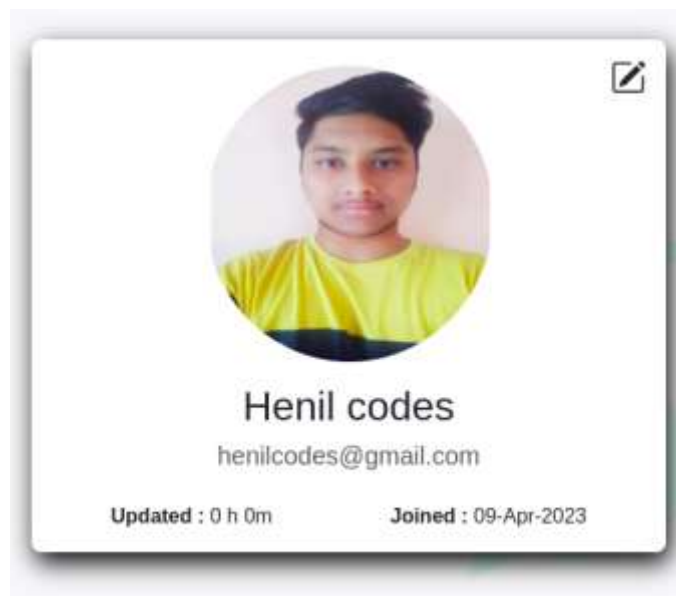
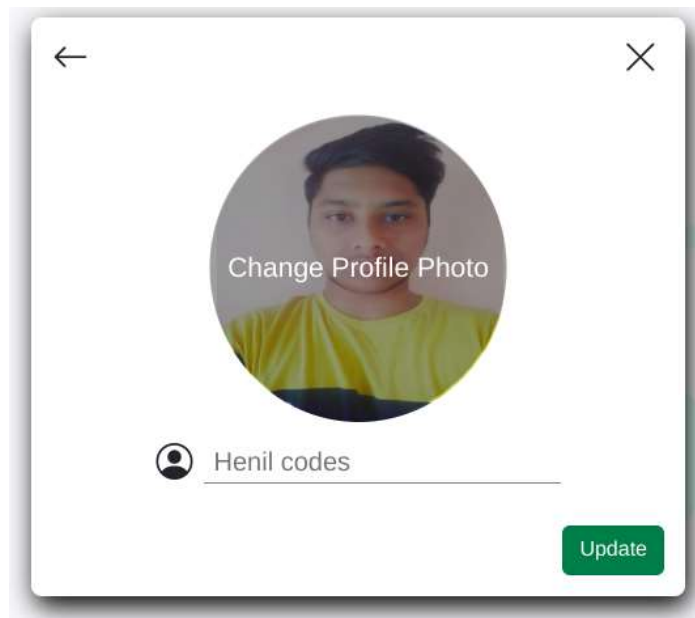
ChatRoom

→ When you click profile photo then displays user's details like username, email address, photo, account created and update time. Users can change username and photo after login successfully.



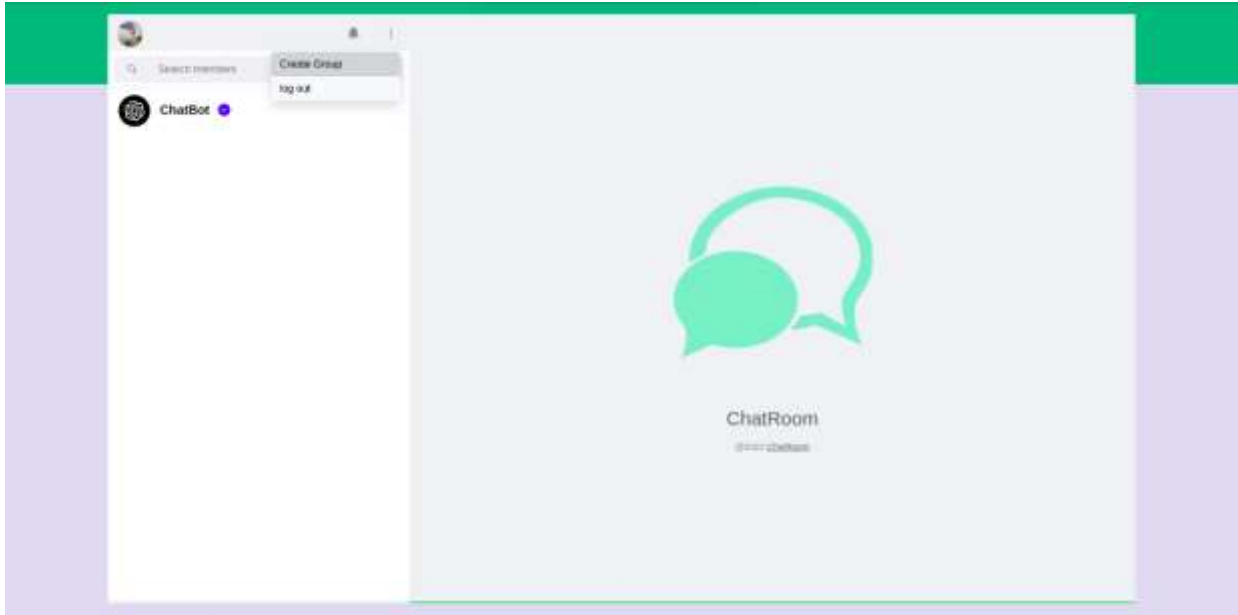
ChatRoom

→ Users can change username and photo after login successfully.

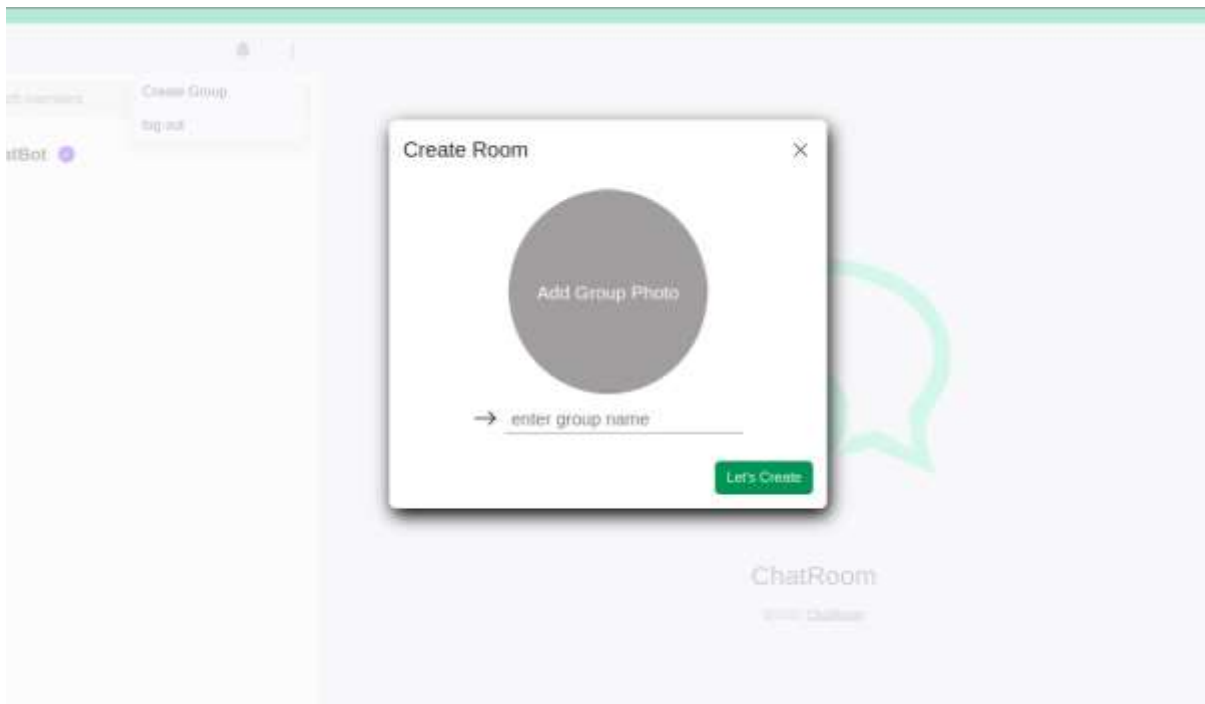


➤ Create Group:

User can click on 3 dots and open the form a create group.

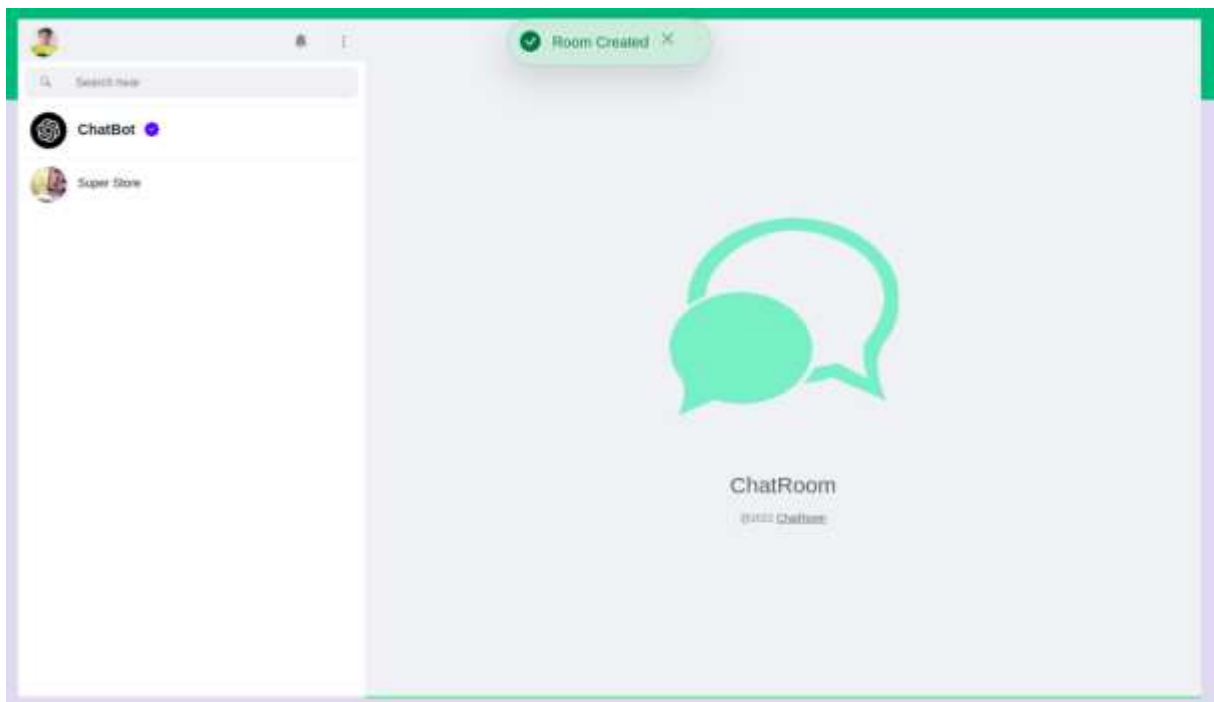
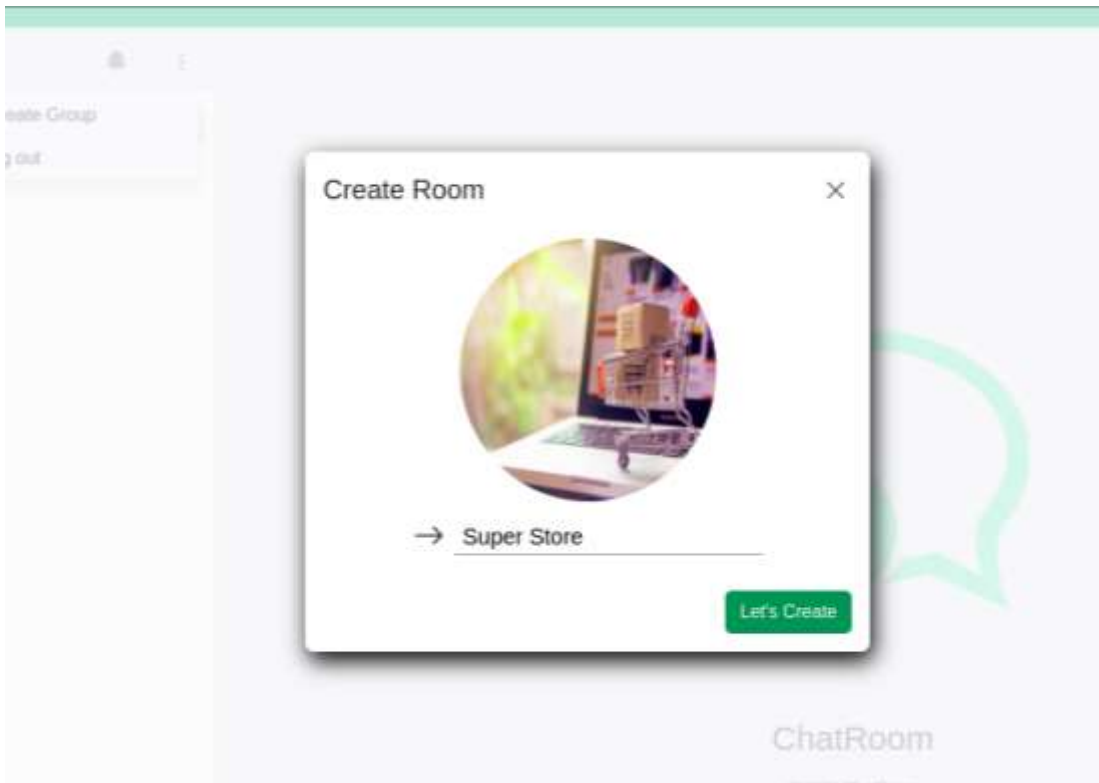


→ For create a room required room profile photo and name.



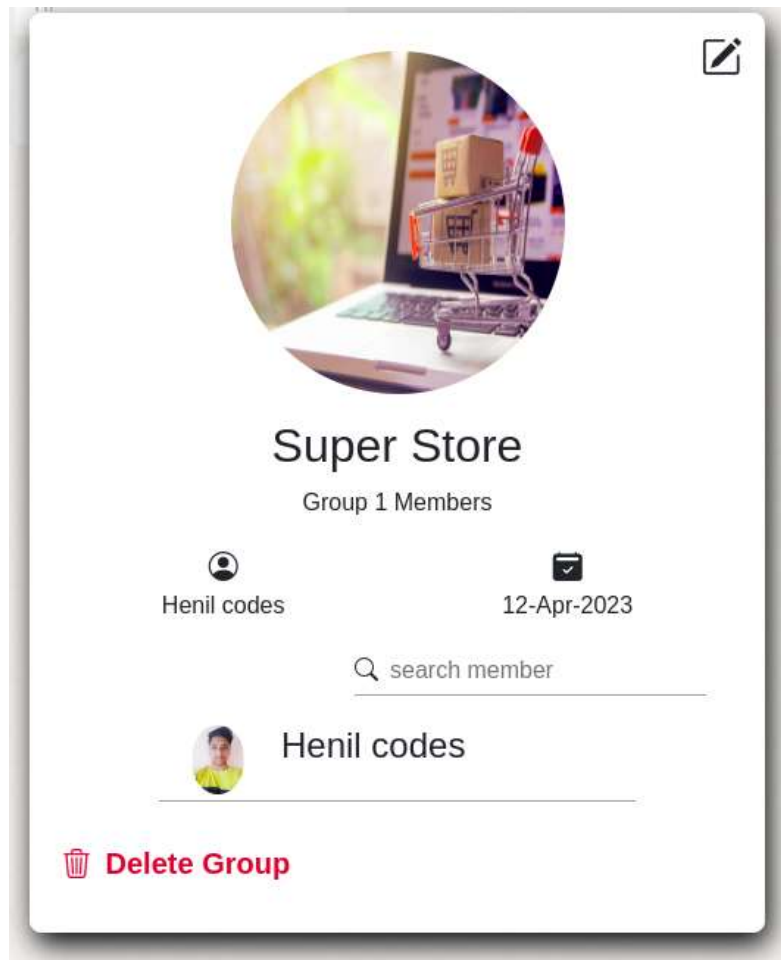
ChatRoom

→ Enter the room name and choose the room profile photo then let us create room.



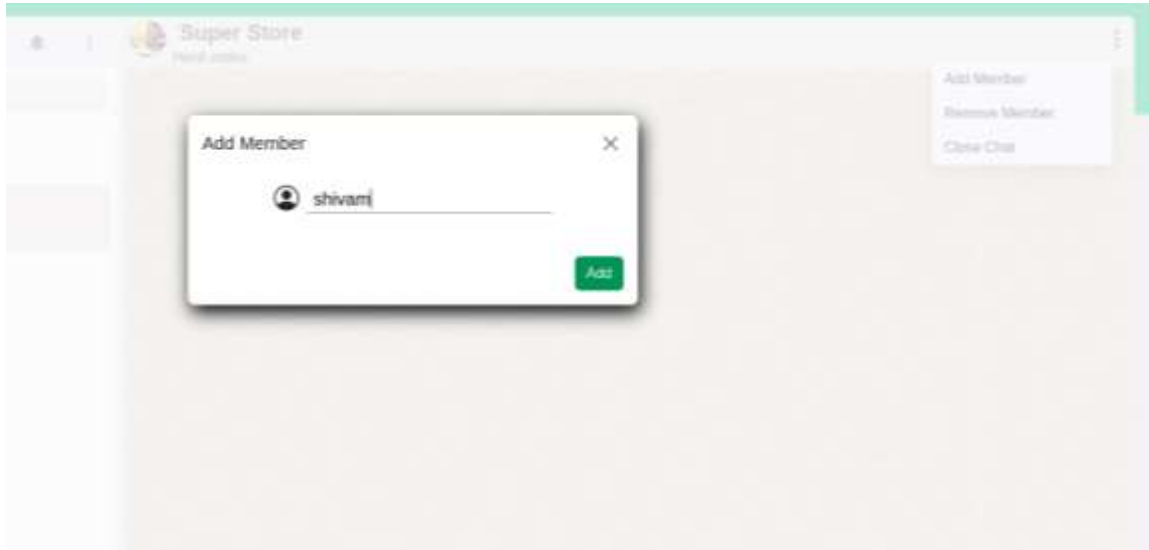
➤ Group Profile

→ Group profile in display group name, photo, admin name, group created date, total member, and list.

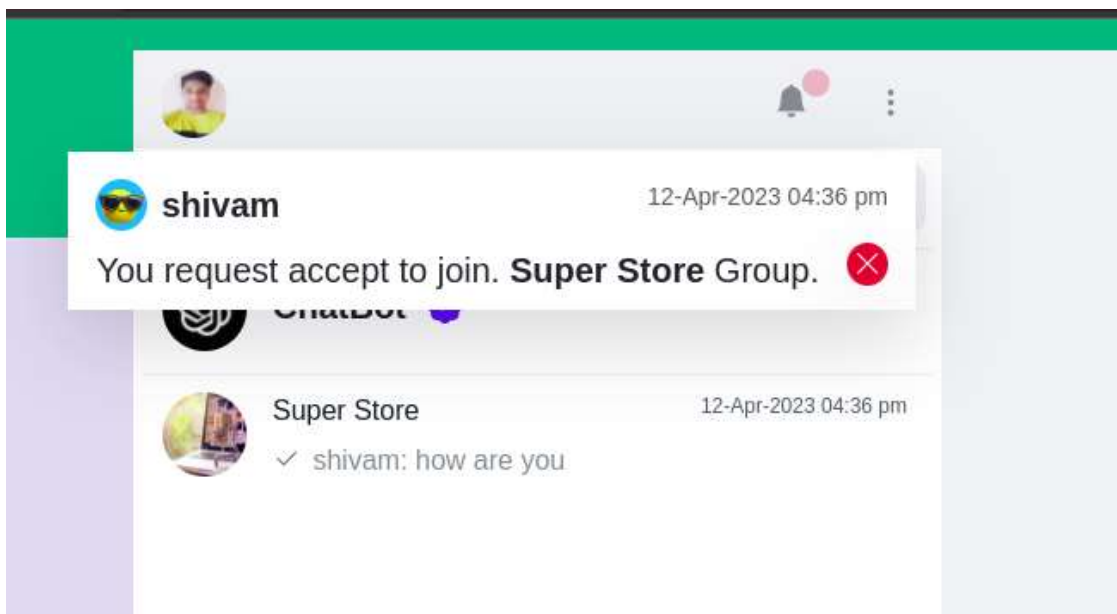


➤ Add Group members:

→ Enter username and if user is valid then send notification to user dashboard.

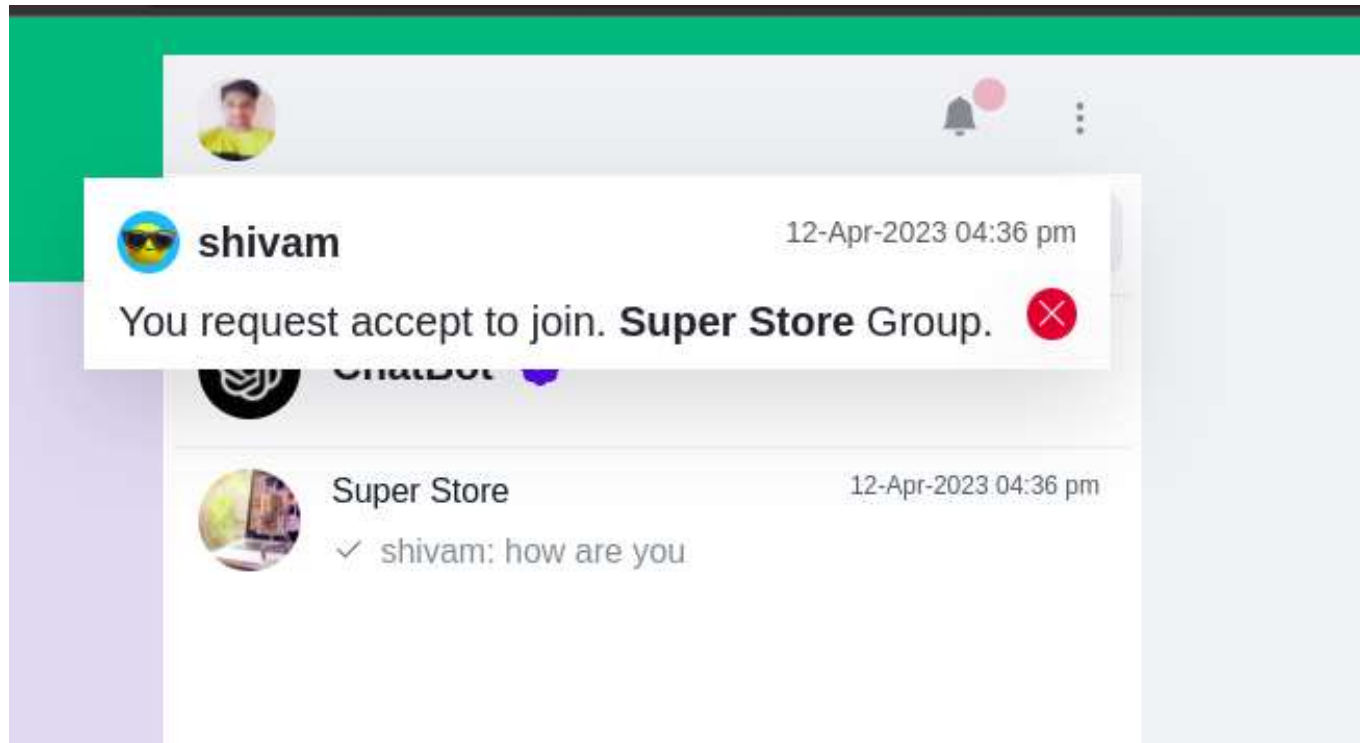


→ The following screen shot is another user dashboard. User is receiving notifications a join Group if the user accepts request, then user join in room else reject invitation request.



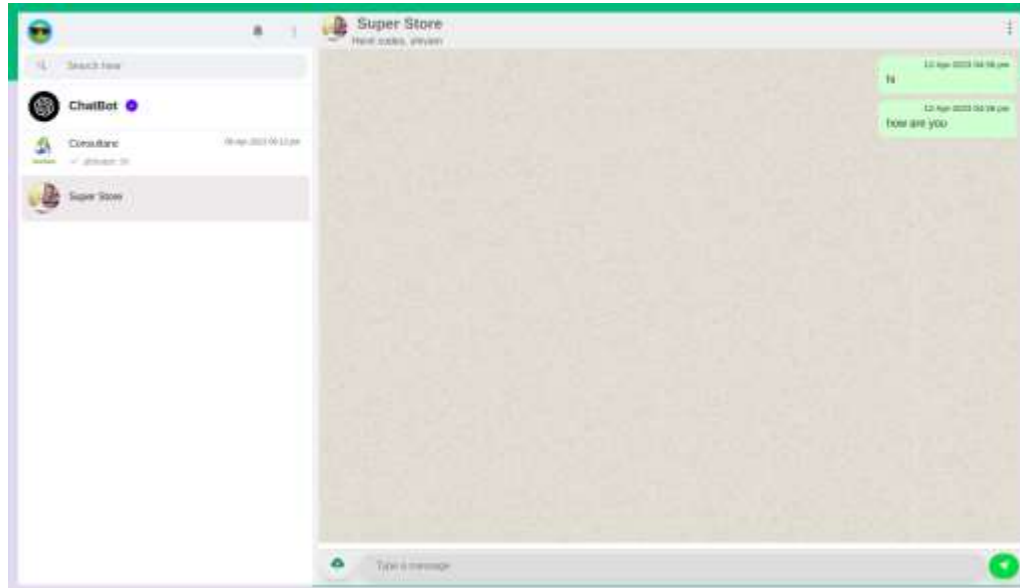
ChatRoom

→ Now the group admin receives room invitations accepted request and start chat.

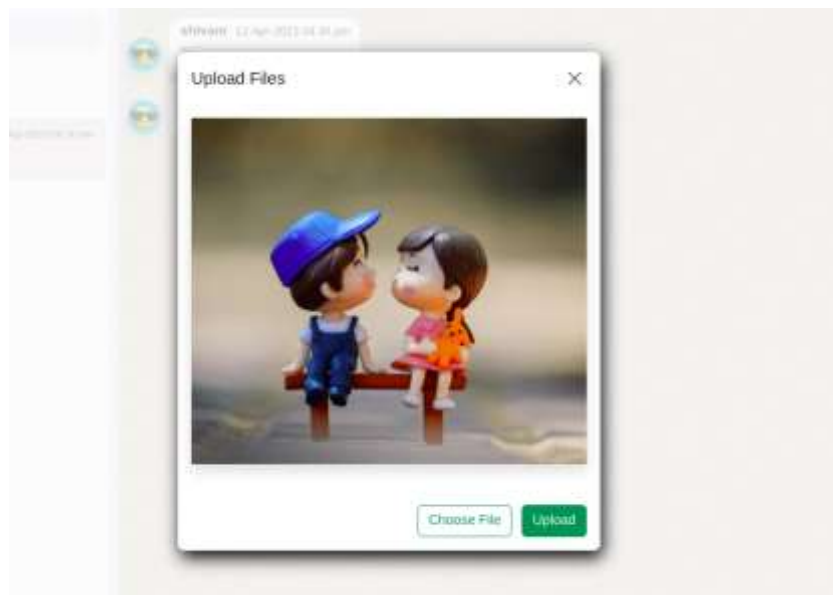


➤ send messages and upload files:

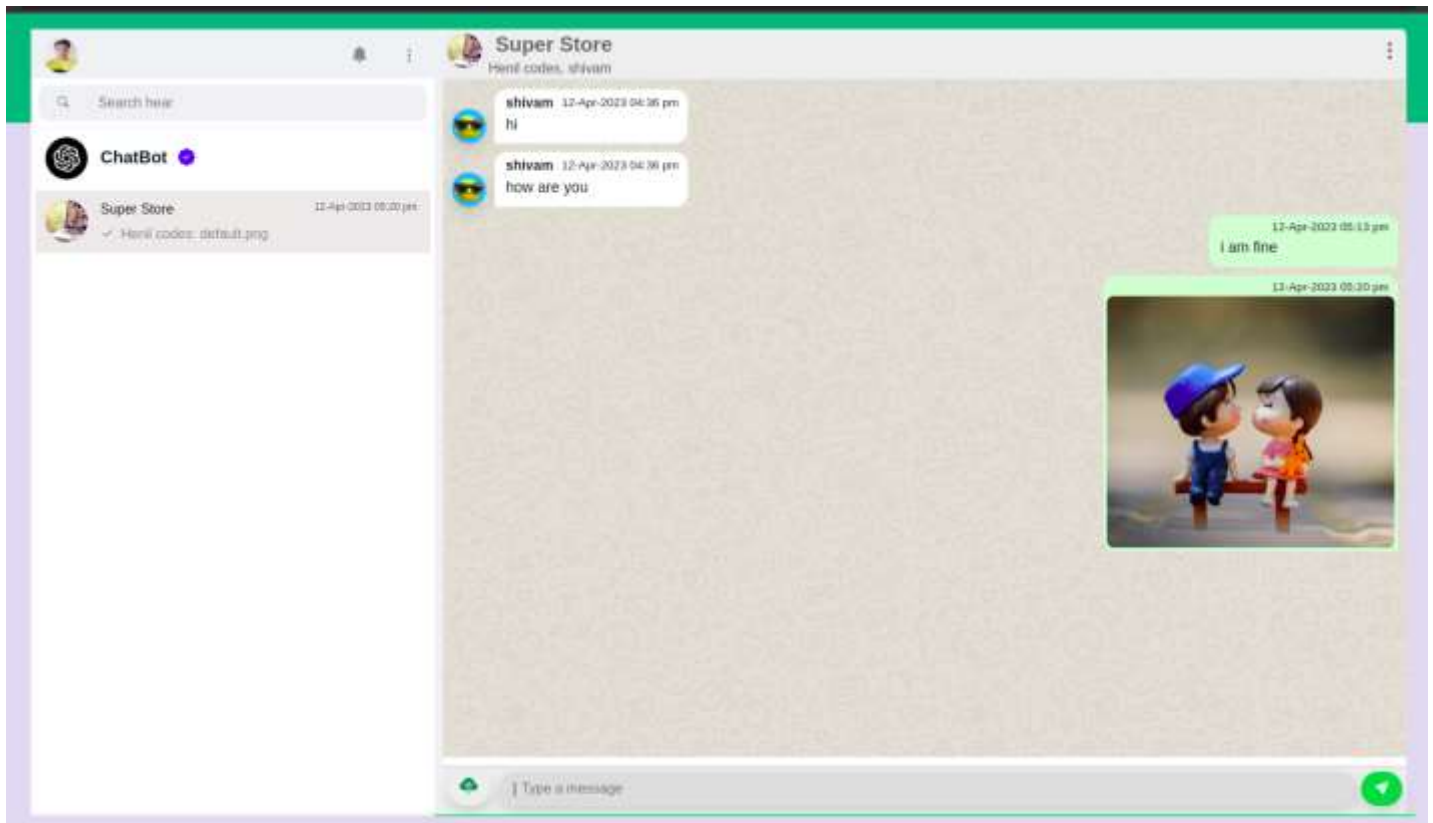
→ When the User accepts invitations request, join group, and start chat.



→ Another user is replying to messages and uploading files.

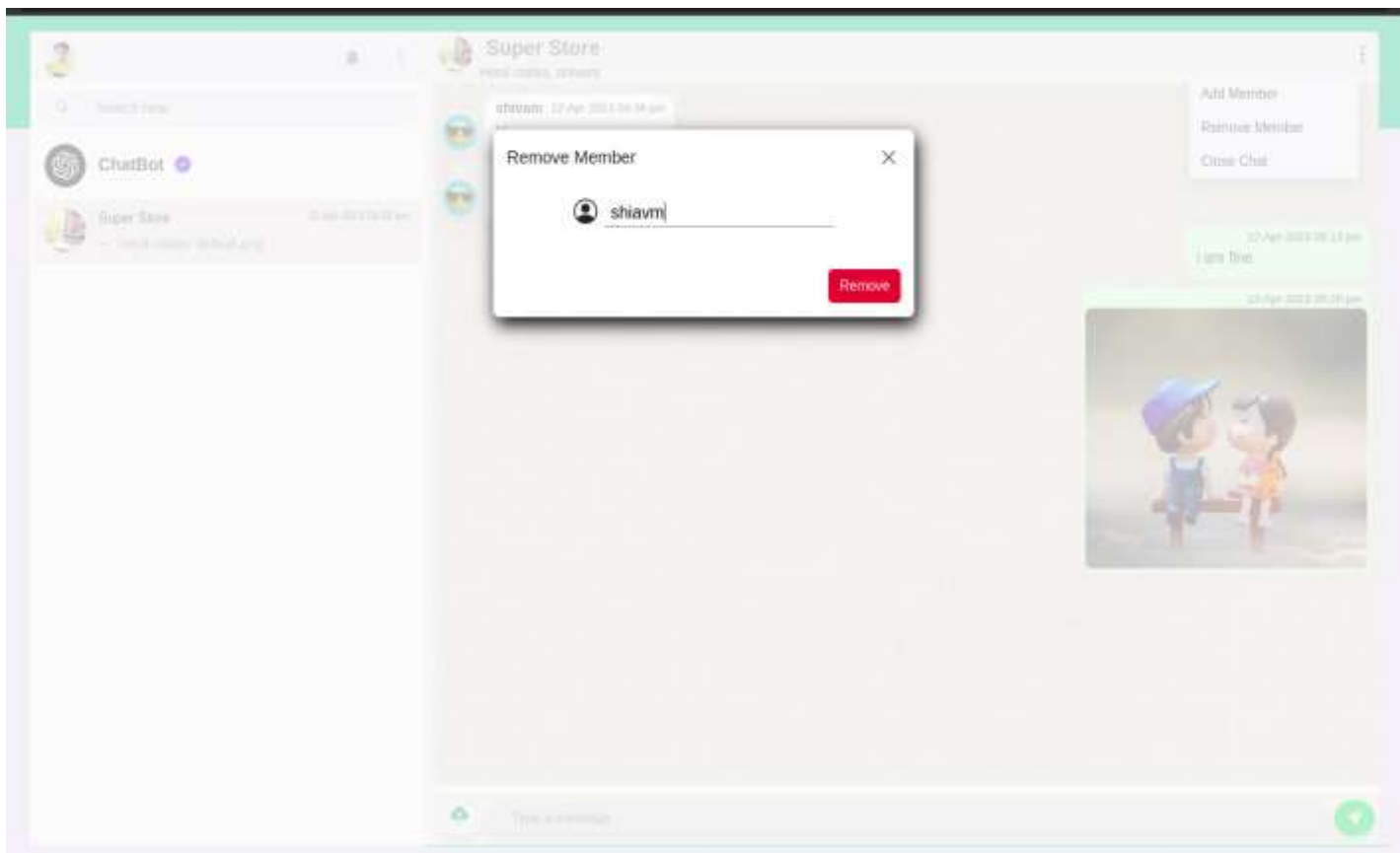


ChatRoom



➤ Remove Group members:

→ Only group admin can remove users from groups. If user is valid then user remove from group.



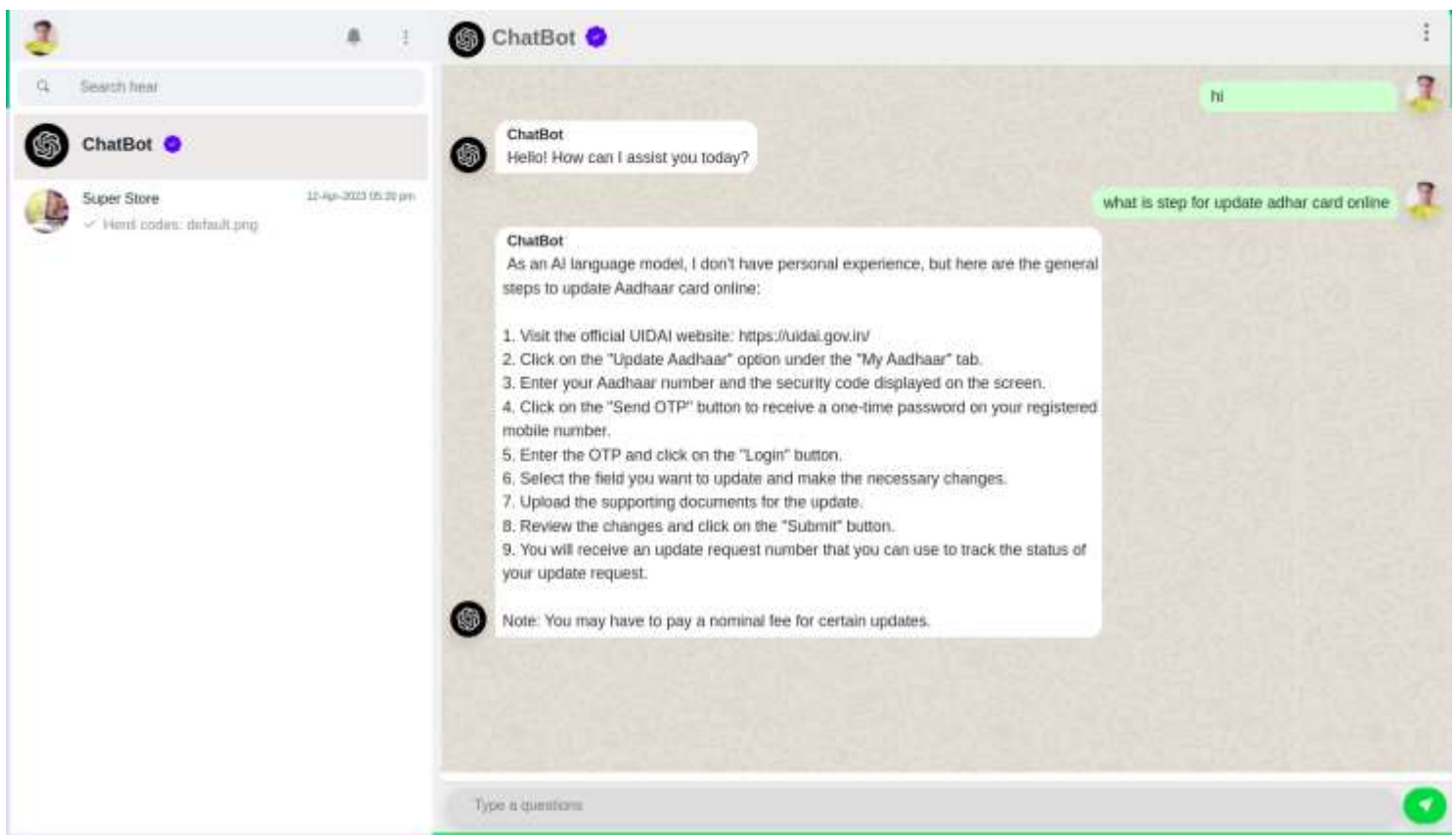
➤ Chat Bot:

→ Chat bot in two features is inbuilt. Using OpenAI API.

- ChatGPT
- DALLE-2

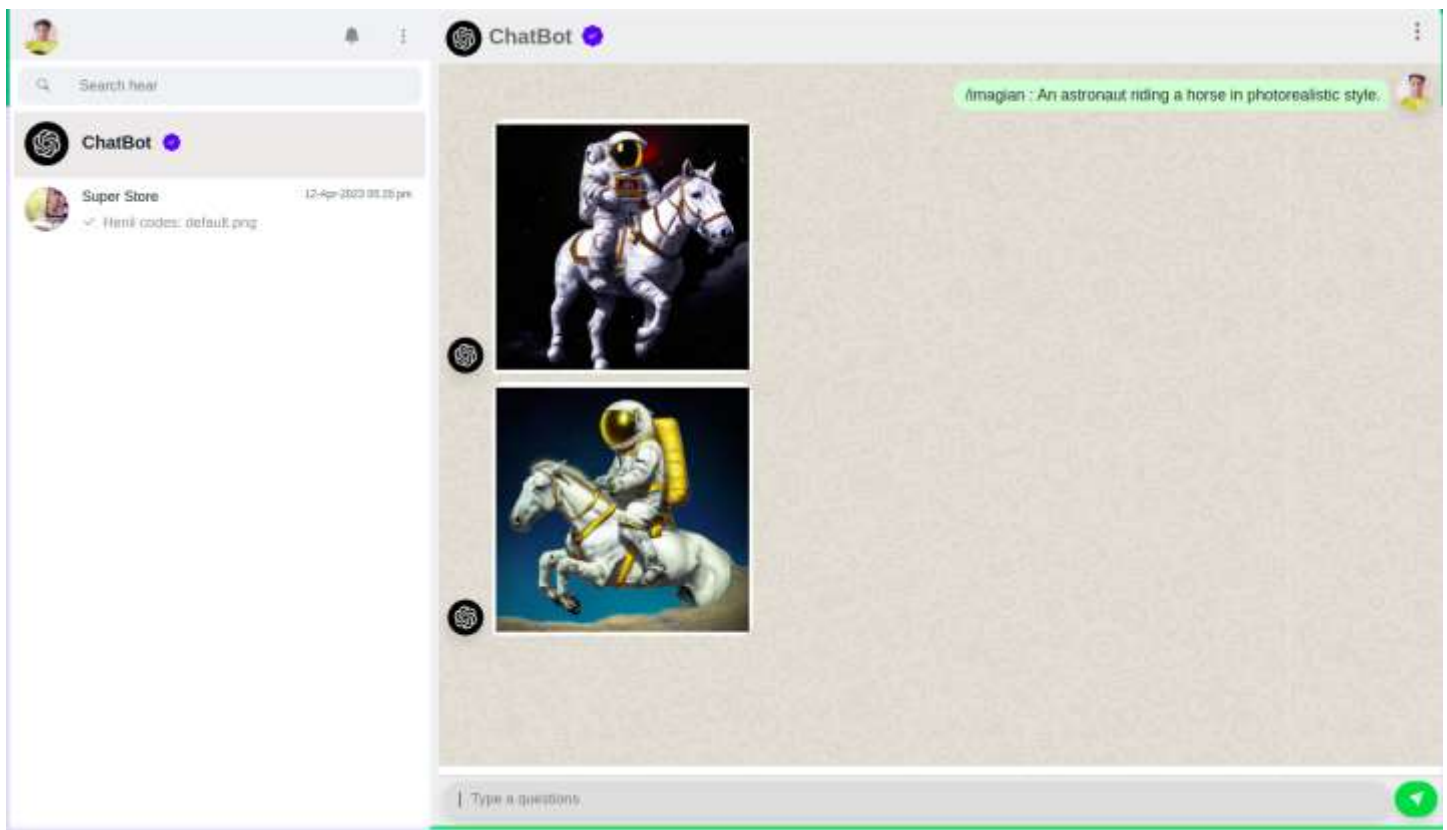
▫ ChatGPT

→ Users can directly ask questions to ChatBot.



▫ DALLE-2

→ If the user wants to use those features, then user required add prefix **/imagine** along the over prompt.

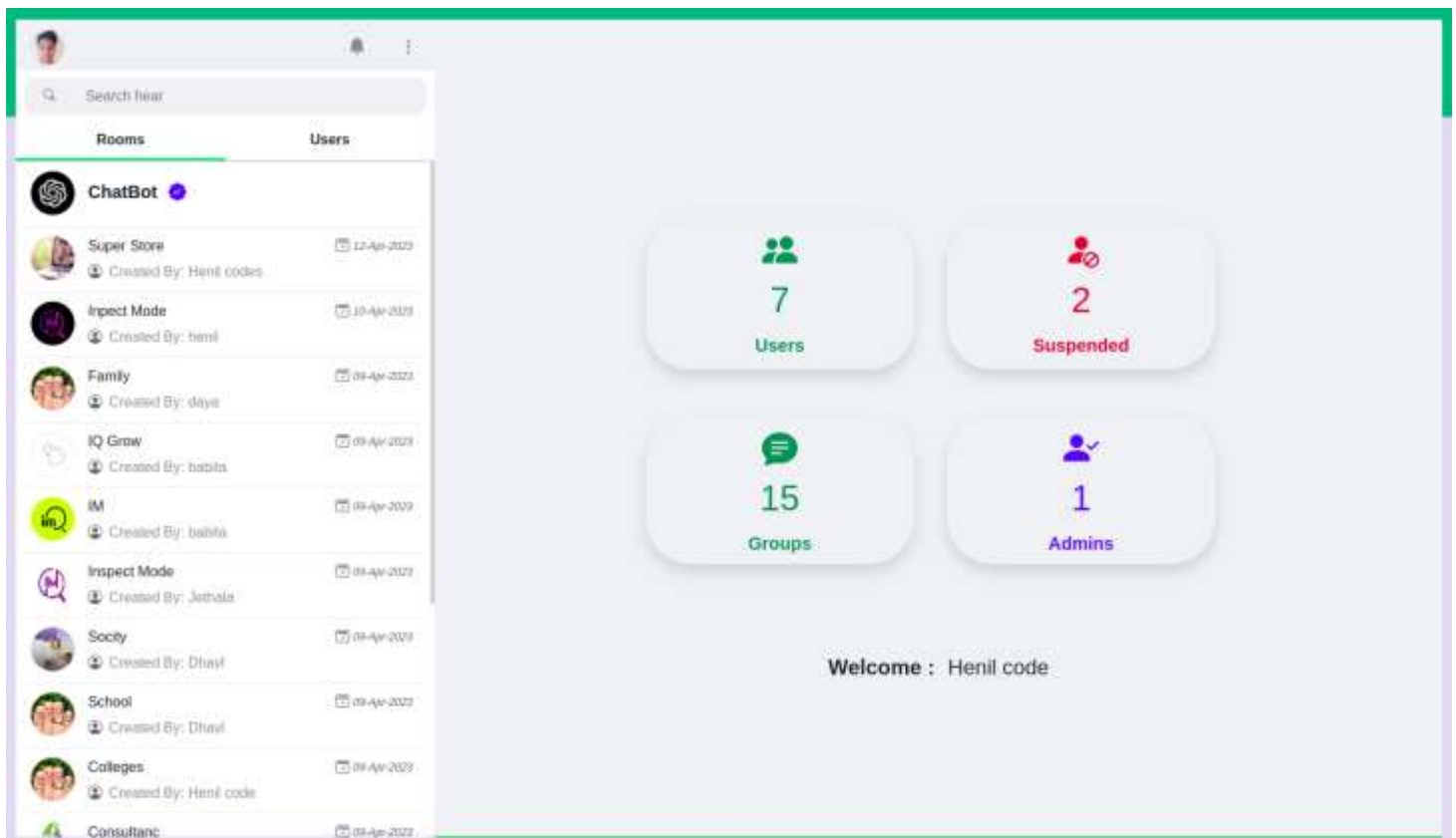


➤ Admin Side:

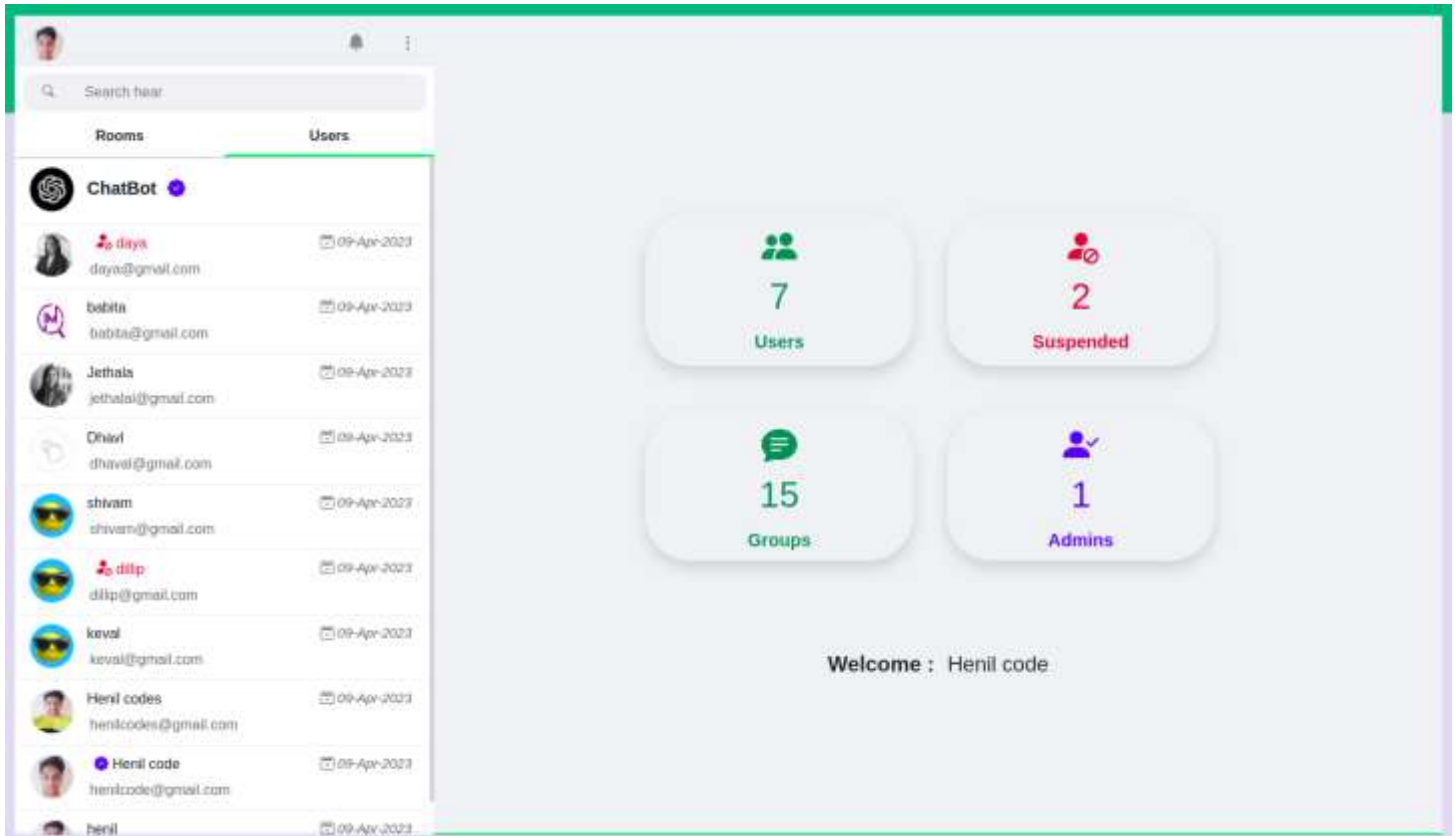
- When admin login in site they can access more features like change user permissions, watch total groups details, watch user details.

➤ Admin dashboard:

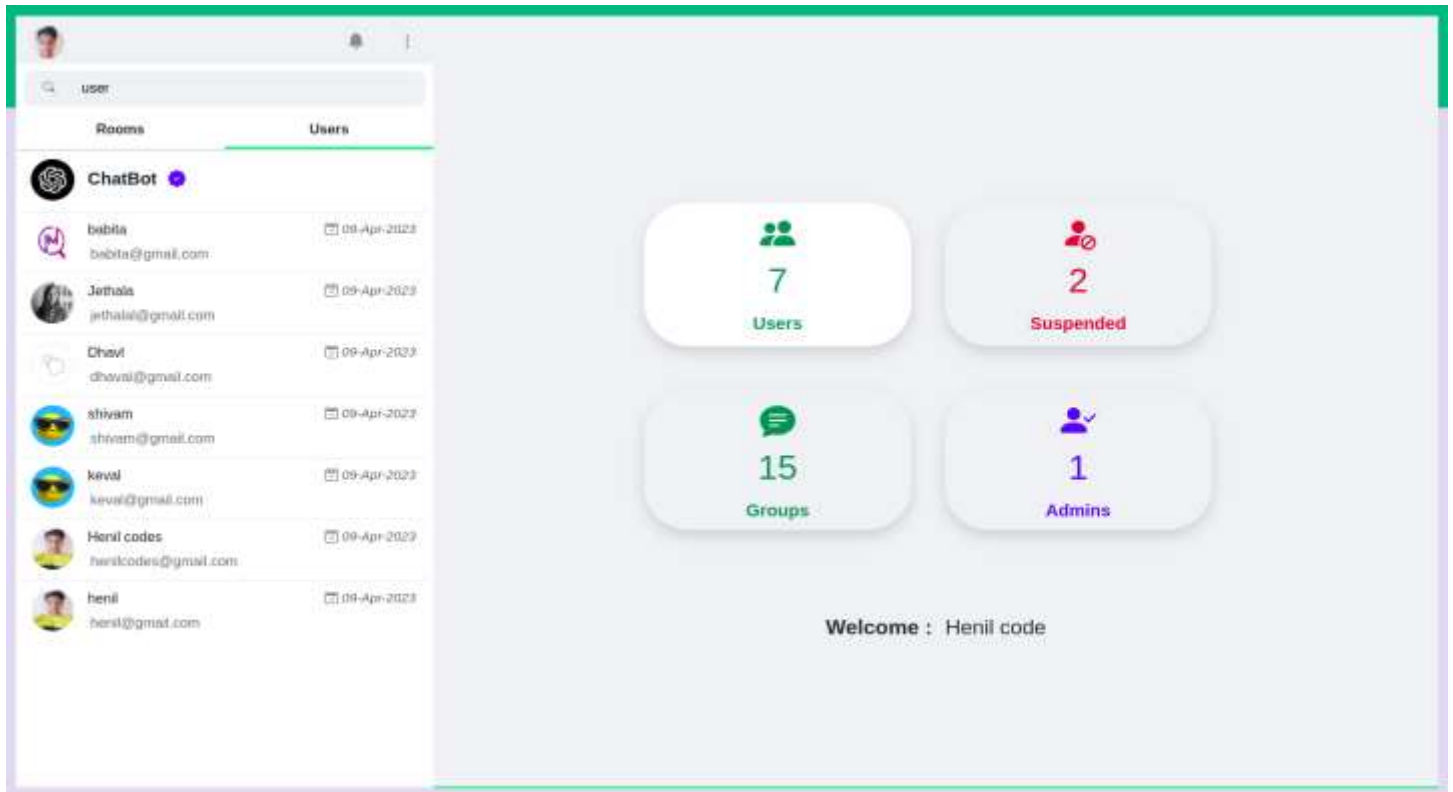
- Dashboard displays total active, suspended and admin users.
- Display total created group list and which user are created with date.



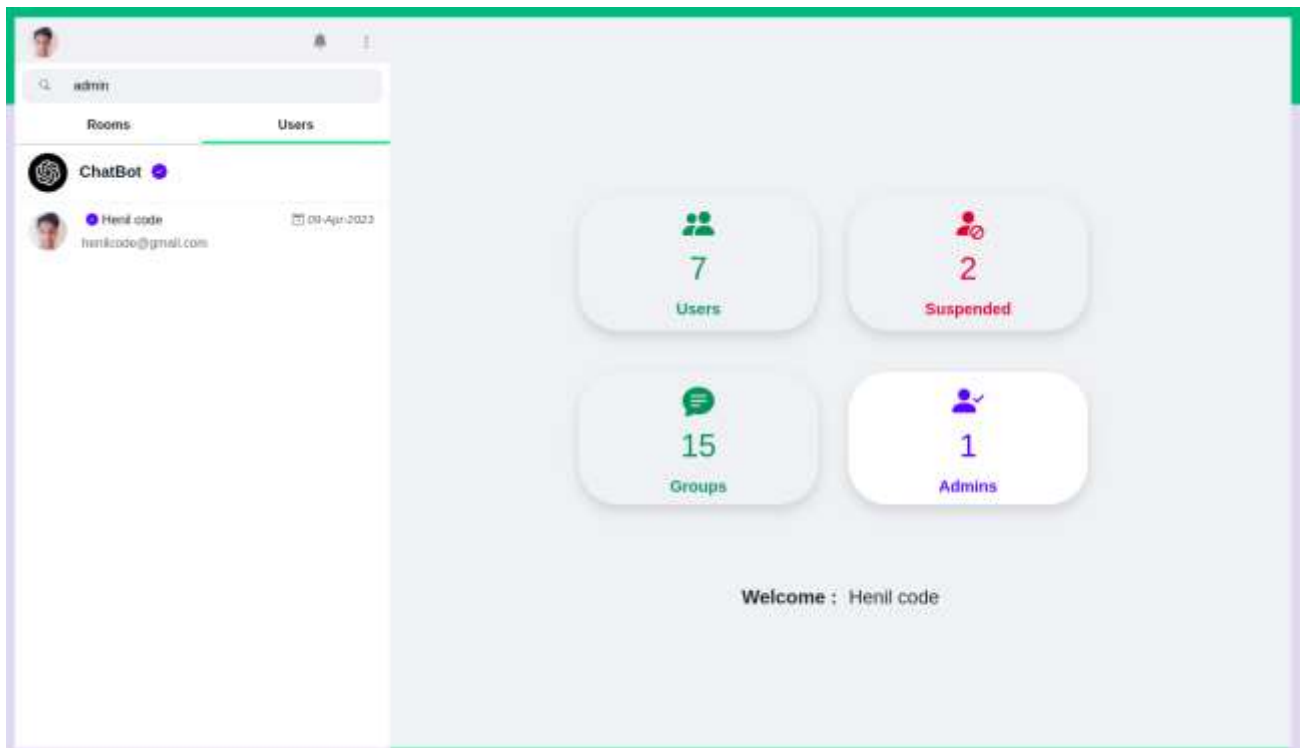
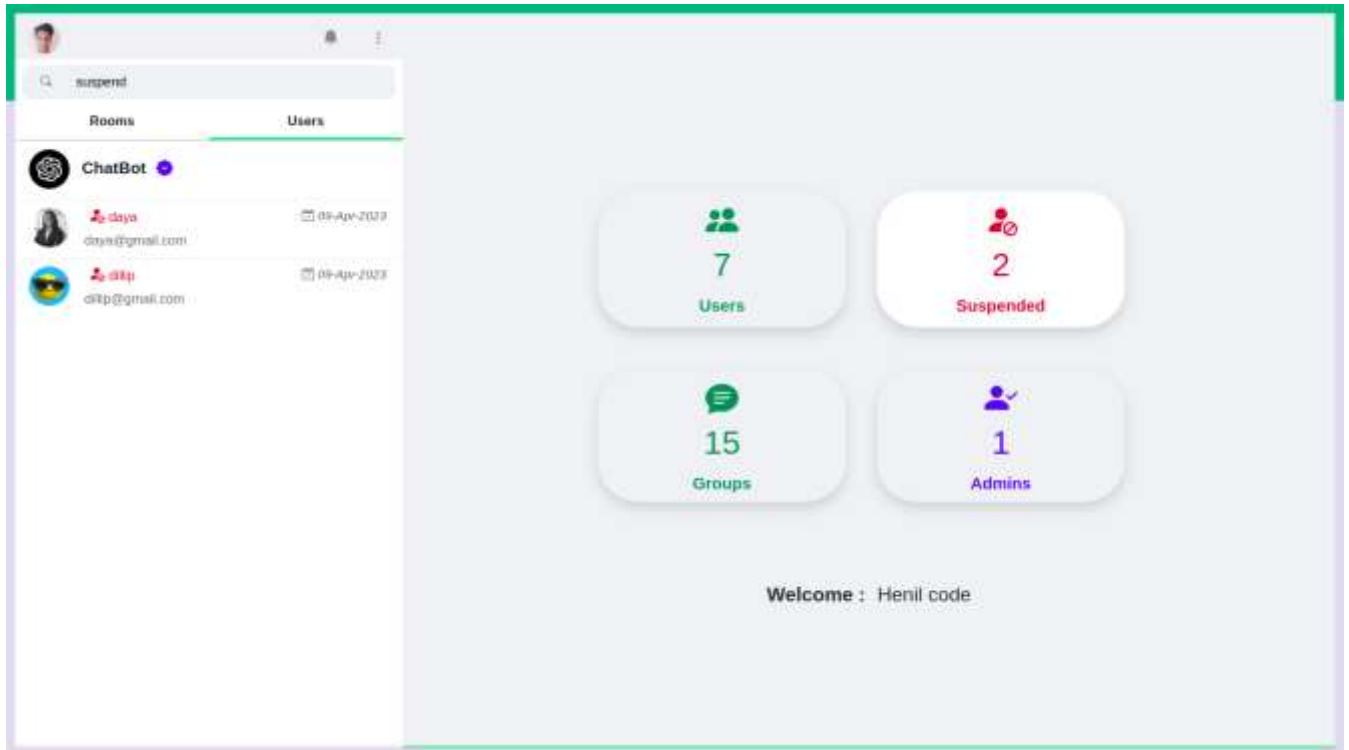
- Admin can check the total register user list with joining date and email.



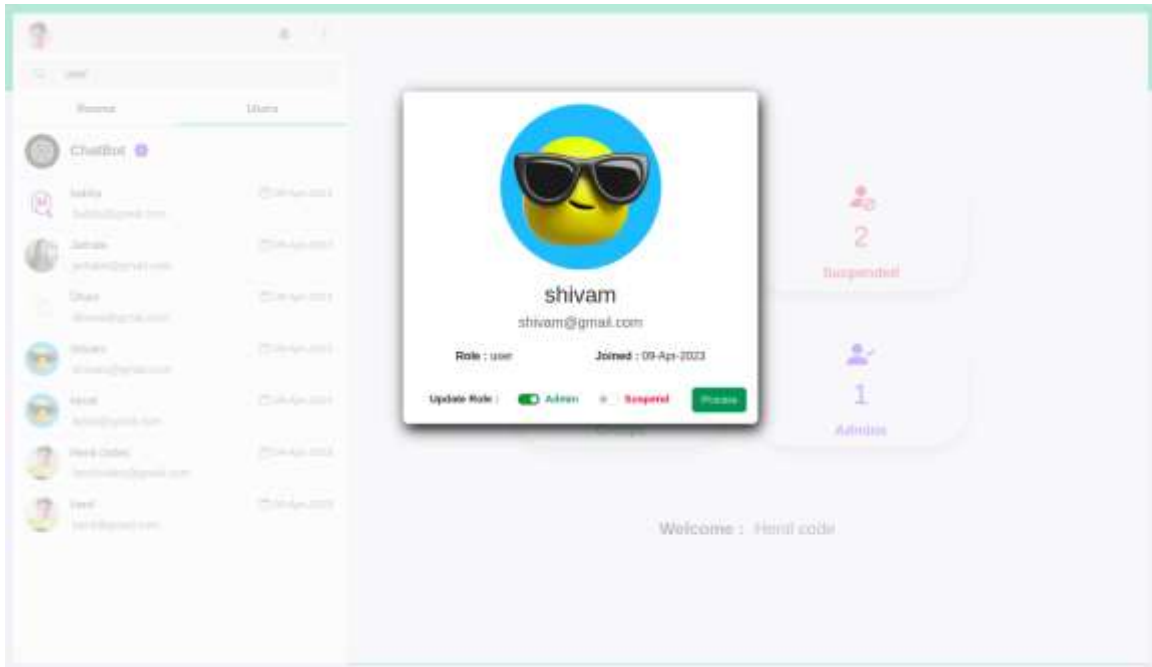
- Admin filter user list and find suspended, admin and active user list.



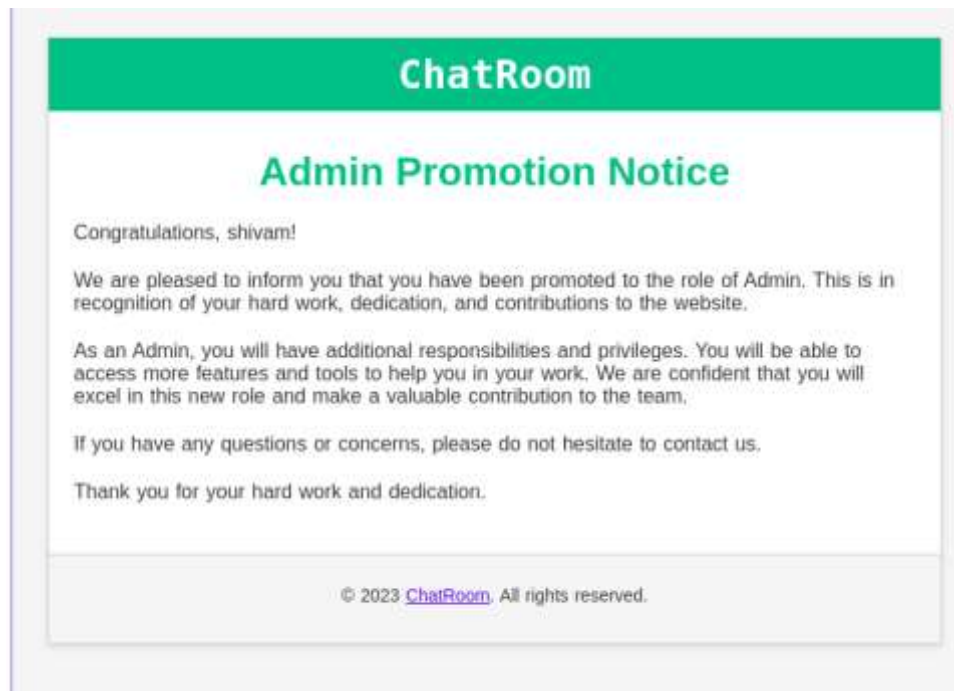
ChatRoom



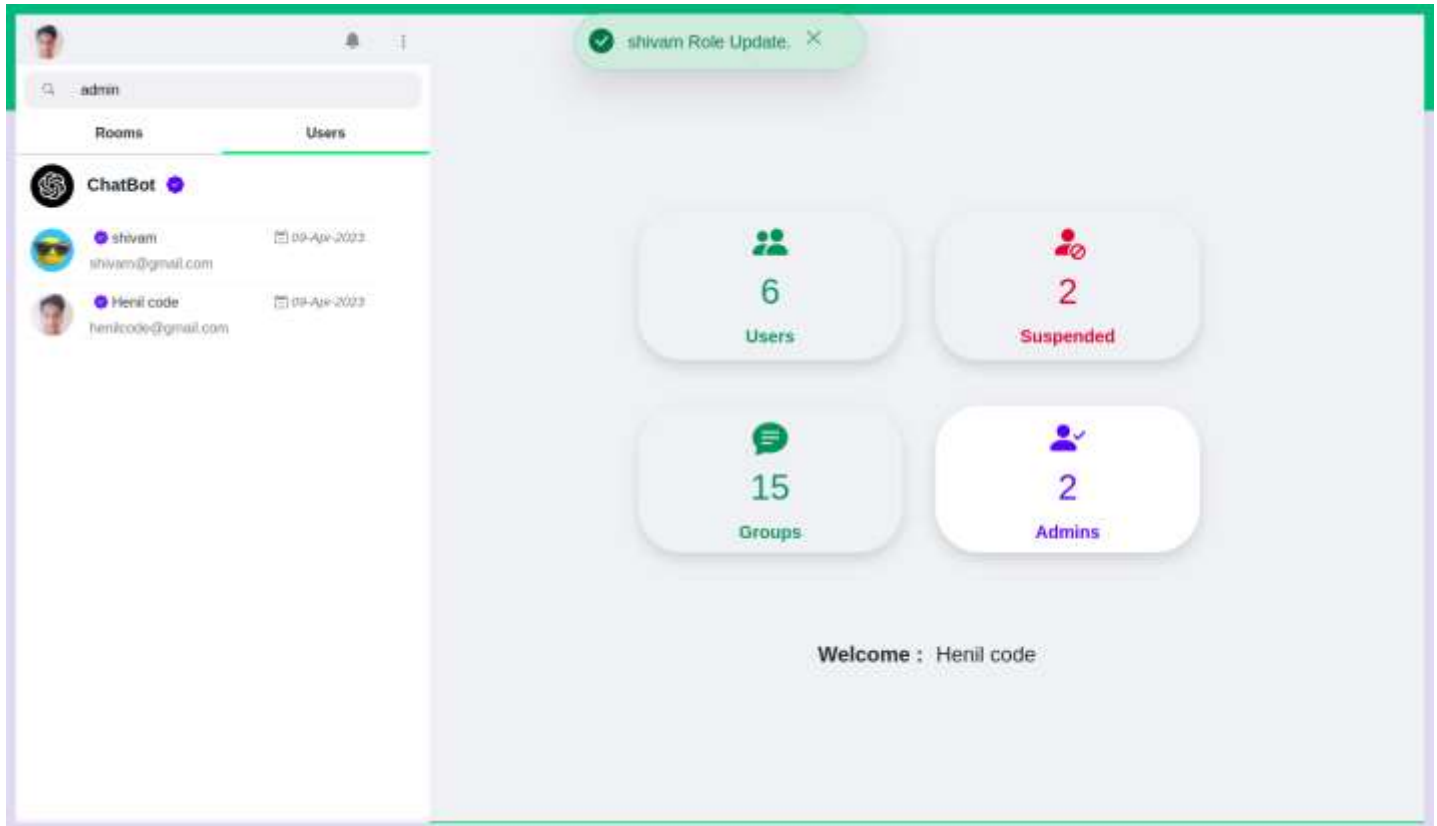
- Admin can change user permissions.
 - After Change permission send mail to user.



- Users can receive mail like.

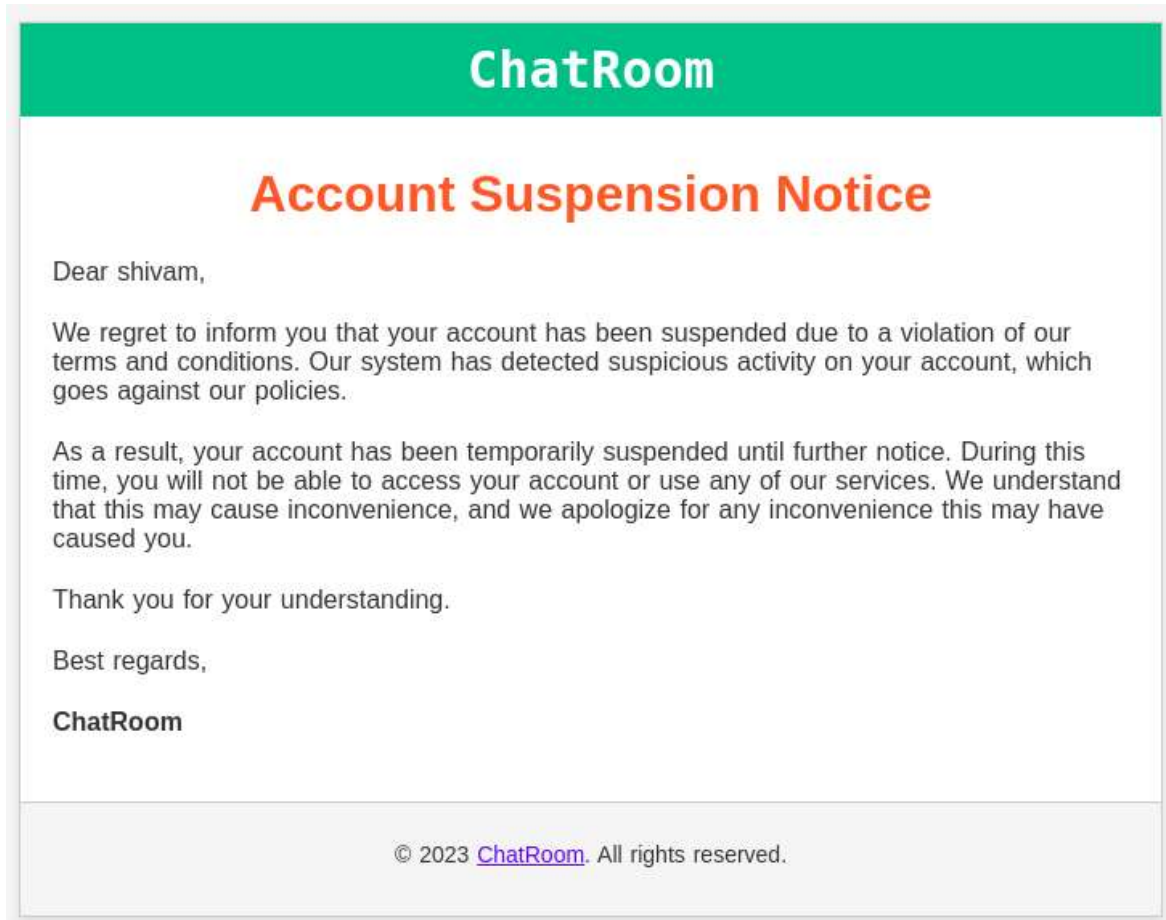


ChatRoom

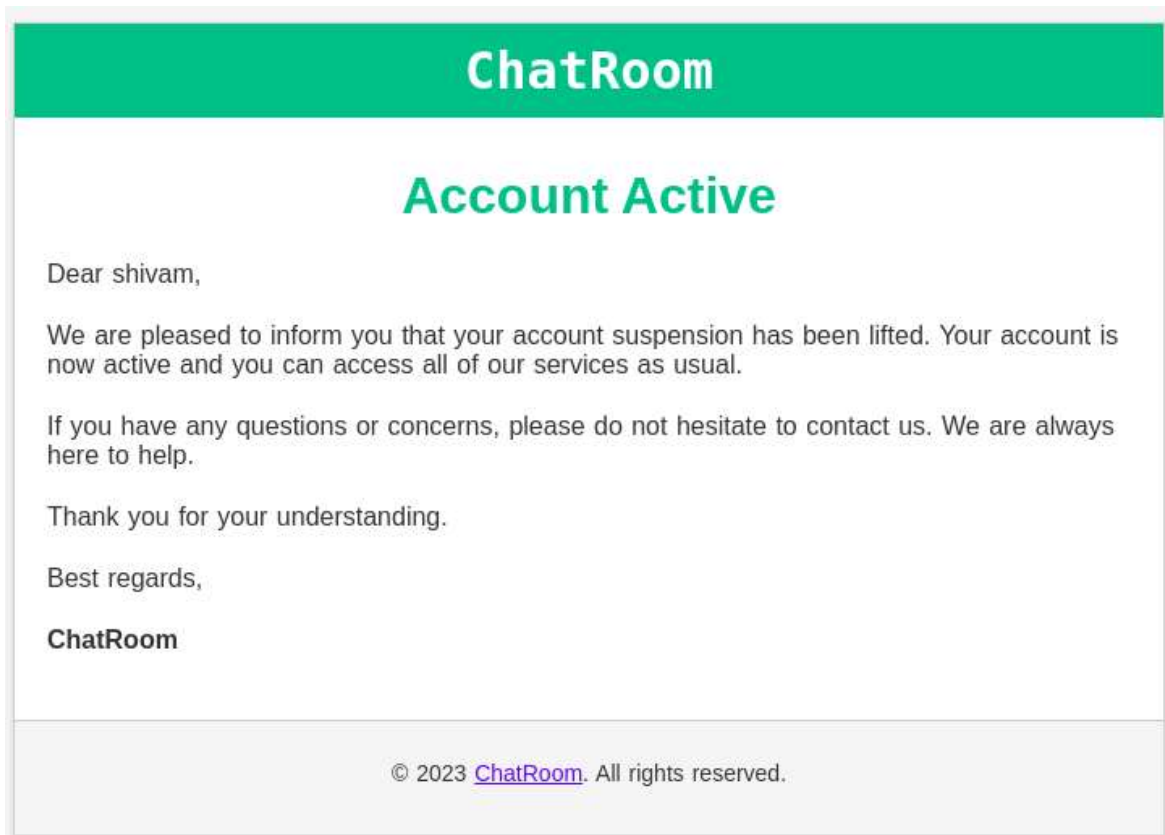


➤ Email template example.

- admins suspend user account, then the user receives mail.



- admin active user account, then the user receives that mail.



Conclusion

chatrooms have become an integral part of our digital communication landscape, providing a platform for individuals from all over the world to connect, share information, and build relationships.

While the early days of chatrooms were often marred by issues such as spam, trolling, and harassment, advancements in technology and improved moderation practices have helped to make chatrooms safer and more enjoyable for everyone involved.

Whether for socializing, networking, or learning, chatrooms offer a range of benefits for users, including the ability to connect with like-minded individuals, seek advice and support from others, and engage in real-time conversations on a variety of topics.

Reference:

- OpenAI
- ChatGPT
- Wikipedia

Thank You