Python Chatapp

**Project folder structure –**

**FlaskChat/**

**├── flask\_chat\_venv/ # Python virtual environment (ignored in Git)**

**├── static/**

**│ └── styles/**

**│ └── style.css # CSS stylesheet**

**├── templates/**

**│ ├── base.html # Base template**

**│ ├── home.html # Home page template**

**│ └── room.html # Chat room template**

**└── main.py # Main application file**

**Styles.css :**

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;

}

body {

  background-color: #f5f5f5;

}

#root {

  min-height: 100vh;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

}

#home-container {

  background-color: white;

  padding: 2rem;

  border-radius: 10px;

  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

  width: 400px;

}

#home-header {

  text-align: center;

  margin-bottom: 1.5rem;

  color: #4a4a4a;

}

#error {

  color: #ff3860;

  margin-bottom: 1rem;

  text-align: center;

}

#chat-widget-home {

  display: flex;

  flex-direction: column;

  gap: 1rem;

}

#name-input {

  display: flex;

  flex-direction: column;

  gap: 0.5rem;

}

#name-input label {

  font-weight: 500;

  color: #363636;

}

#name-input input {

  padding: 0.5rem;

  border: 1px solid #dbdbdb;

  border-radius: 4px;

  font-size: 1rem;

}

#code-label {

  font-weight: 500;

  color: #363636;

}

#chat-widget-home > div:nth-child(2) {

  display: flex;

  gap: 0.5rem;

  align-items: flex-end;

}

#chat-widget-home > div:nth-child(2) input {

  flex-grow: 1;

  padding: 0.5rem;

  border: 1px solid #dbdbdb;

  border-radius: 4px;

  font-size: 1rem;

}

#join,

#create {

  padding: 0.5rem 1rem;

  background-color: #485fc7;

  color: white;

  border: none;

  border-radius: 4px;

  font-size: 1rem;

  cursor: pointer;

}

#join:hover,

#create:hover {

  background-color: #3a56c0;

}

hr {

  border: none;

  border-top: 1px solid #dbdbdb;

  margin: 1rem 0;

}

#create {

  width: 100%;

}

/\* Room styles \*/

#room-container {

  background-color: white;

  padding: 2rem;

  border-radius: 10px;

  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

  width: 600px;

  max-height: 90vh;

  display: flex;

  flex-direction: column;

}

#room-subsection {

  display: flex;

  justify-content: space-between;

  align-items: center;

  margin-bottom: 1rem;

}

#room-code-display span {

  color: #485fc7;

  font-weight: bold;

}

#leave-chat-btn {

  color: #ff3860;

  text-decoration: none;

  font-weight: 500;

}

#leave-chat-btn:hover {

  text-decoration: underline;

}

#chat-room-widget {

  display: flex;

  flex-direction: column;

  flex-grow: 1;

  gap: 1rem;

}

#msgs-container {

  flex-grow: 1;

  border: 1px solid #dbdbdb;

  border-radius: 4px;

  padding: 1rem;

  overflow-y: auto;

  max-height: 60vh;

  display: flex;

  flex-direction: column;

}

#messages {

  list-style: none;

  display: flex;

  flex-direction: column;

  gap: 0.5rem;

}

.message-item {

  max-width: 70%;

  padding: 0.5rem 1rem;

  border-radius: 1rem;

}

.self-message-item {

  align-self: flex-end;

  background-color: #485fc7;

  color: white;

}

.peer-message-item {

  align-self: flex-start;

  background-color: #f5f5f5;

  color: #363636;

}

.member-activity {

  align-self: center;

  color: #7a7a7a;

  font-size: 0.9rem;

}

.muted-text {

  color: #d1d5ff;

}

.muted-text-white {

  color: #7a7a7a;

}

#message-box {

  display: flex;

  gap: 0.5rem;

}

#message-input {

  flex-grow: 1;

  padding: 0.5rem;

  border: 1px solid #dbdbdb;

  border-radius: 4px;

  font-size: 1rem;

}

#send-btn {

  padding: 0.5rem 1rem;

  background-color: #485fc7;

  color: white;

  border: none;

  border-radius: 4px;

  font-size: 1rem;

  cursor: pointer;

}

#send-btn:hover {

  background-color: #3a56c0;

}

**base.html :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Flask Chat</title>

    <script src="https://cdn.tailwindcss.com"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.0.1/socket.io.js"></script>

    <script>

        tailwind.config = {

            theme: {

                extend: {

                    animation: {

                        'gradient-x': 'gradient-x 6s ease infinite',

                        'gradient-y': 'gradient-y 6s ease infinite',

                        'gradient-xy': 'gradient-xy 6s ease infinite',

                    },

                    keyframes: {

                        'gradient-x': {

                            '0%, 100%': {

                                'background-size': '200% 200%',

                                'background-position': 'left center'

                            },

                            '50%': {

                                'background-size': '200% 200%',

                                'background-position': 'right center'

                            }

                        },

                        'gradient-y': {

                            '0%, 100%': {

                                'background-size': '400% 400%',

                                'background-position': 'center top'

                            },

                            '50%': {

                                'background-size': '200% 200%',

                                'background-position': 'center bottom'

                            }

                        },

                        'gradient-xy': {

                            '0%, 100%': {

                                'background-size': '400% 400%',

                                'background-position': 'left center'

                            },

                            '50%': {

                                'background-size': '200% 200%',

                                'background-position': 'right center'

                            }

                        }

                    }

                }

            }

        }

    </script>

    <style>

        .message-enter {

            animation: message-enter 0.3s ease-out;

        }

        @keyframes message-enter {

            from {

                opacity: 0;

                transform: translateY(10px);

            }

            to {

                opacity: 1;

                transform: translateY(0);

            }

        }

    </style>

</head>

<body class="min-h-screen bg-gray-100">

    {% block content %}{% endblock %}

</body>

</html>

**Home.html :**

{% extends 'base.html' %}

{% block content %}

<div class="min-h-screen flex items-center justify-center bg-gradient-to-br from-indigo-500 via-purple-500 to-pink-500 animate-gradient-xy">

    <div class="w-full max-w-md px-8 py-12 bg-white/90 backdrop-blur-sm rounded-xl shadow-2xl space-y-6 transition-all duration-300 hover:shadow-3xl">

        <div class="text-center">

            <h1 class="text-4xl font-bold text-transparent bg-clip-text bg-gradient-to-r from-indigo-600 to-pink-600 mb-2">Flask Chat</h1>

            <p class="text-gray-600">Connect and chat in real-time</p>

        </div>

        {% if error %}

        <div class="bg-red-100 border-l-4 border-red-500 text-red-700 p-4 animate-bounce">

            <p>{{ error }}</p>

        </div>

        {% endif %}

        <form method="post" class="space-y-6">

            <div>

                <label for="name" class="block text-sm font-medium text-gray-700">Your Name</label>

                <input type="text" id="name" name="name" value="{{ name }}"

                       class="mt-1 block w-full px-4 py-2 border border-gray-300 rounded-md shadow-sm focus:ring-indigo-500 focus:border-indigo-500 transition-all duration-200"

                       placeholder="Enter your name">

            </div>

            <div>

                <label for="code" class="block text-sm font-medium text-gray-700">Room Code</label>

                <div class="mt-1 flex rounded-md shadow-sm">

                    <input type="text" id="code" name="code" value="{{ code }}"

                           class="flex-1 min-w-0 block w-full px-4 py-2 rounded-l-md border border-gray-300 focus:ring-indigo-500 focus:border-indigo-500 transition-all duration-200"

                           placeholder="Enter room code">

                    <button type="submit" name="join"

                            class="inline-flex items-center px-4 py-2 border border-l-0 border-gray-300 rounded-r-md bg-indigo-600 text-white hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500 transition-all duration-200">

                        Join

                    </button>

                </div>

            </div>

            <div class="relative">

                <div class="absolute inset-0 flex items-center">

                    <div class="w-full border-t border-gray-300"></div>

                </div>

                <div class="relative flex justify-center text-sm">

                    <span class="px-2 bg-white text-gray-500">OR</span>

                </div>

            </div>

            <button type="submit" name="create"

                    class="w-full flex justify-center py-3 px-4 border border-transparent rounded-md shadow-sm text-white bg-gradient-to-r from-purple-600 to-pink-600 hover:from-purple-700 hover:to-pink-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-purple-500 transition-all duration-300 transform hover:scale-105">

                Create New Room

            </button>

        </form>

    </div>

</div>

{% endblock %}

**room.html :**

{% extends 'base.html' %}

{% block content %}

<div class="min-h-screen flex flex-col bg-gradient-to-br from-blue-500 via-indigo-500 to-purple-600 animate-gradient-x">

    <div class="container mx-auto px-4 py-8 flex-1 flex flex-col">

        <div class="flex justify-between items-center mb-6">

            <div>

                <h1 class="text-3xl font-bold text-white">Flask Chat 💬</h1>

                <div class="flex items-center mt-2">

                    <span class="text-white/80">Room:</span>

                    <span class="ml-2 px-3 py-1 rounded-full bg-white/20 text-white font-medium">{{ room }}</span>

                </div>

            </div>

            <a href="/" class="px-4 py-2 rounded-lg bg-white/10 hover:bg-white/20 text-white backdrop-blur-sm transition-all duration-200 flex items-center">

                <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5 mr-1" viewBox="0 0 20 20" fill="currentColor">

                    <path fill-rule="evenodd" d="M9.707 14.707a1 1 0 01-1.414 0l-4-4a1 1 0 010-1.414l4-4a1 1 0 011.414 1.414L7.414 9H15a1 1 0 110 2H7.414l2.293 2.293a1 1 0 010 1.414z" clip-rule="evenodd" />

                </svg>

                Leave

            </a>

        </div>

        <div class="flex-1 flex flex-col bg-white/10 backdrop-blur-sm rounded-xl shadow-xl overflow-hidden">

            <div id="msgs-container" class="flex-1 p-4 overflow-y-auto">

                <ul id="messages" class="space-y-4"></ul>

            </div>

            <div class="p-4 bg-white/20 border-t border-white/10">

                <div class="flex space-x-2">

                    <input type="text" id="message-input" placeholder="Type your message..."

                           class="flex-1 px-4 py-3 rounded-lg bg-white/90 focus:outline-none focus:ring-2 focus:ring-indigo-500 transition-all duration-200">

                    <button id="send-btn" onclick="sendMessage()"

                            class="px-6 py-3 rounded-lg bg-indigo-600 text-white hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-indigo-500 focus:ring-offset-2 transition-all duration-200 transform hover:scale-105">

                        Send

                    </button>

                </div>

            </div>

        </div>

    </div>

</div>

<script>

    var socketio = io();

    socketio.on("message", function (message) {

        createChatItem(message.message, message.sender);

    });

    function createChatItem(message, sender) {

        var messages = document.getElementById("messages");

        var isCurrentUser = "{{ user }}" === sender;

        var isSystem = sender === "";

        if (isSystem) {

            var content = `

                <li class="text-center text-white/70 text-sm py-2 message-enter">

                    ${message}

                </li>

            `;

        } else {

            var content = `

                <li class="message-enter ${isCurrentUser ? 'flex justify-end' : 'flex justify-start'}">

                    <div class="${isCurrentUser ? 'bg-indigo-600' : 'bg-white'} ${isCurrentUser ? 'text-white' : 'text-gray-800'} max-w-xs lg:max-w-md px-4 py-2 rounded-lg ${isCurrentUser ? 'rounded-tr-none' : 'rounded-tl-none'} shadow">

                        ${!isCurrentUser ? `<div class="font-bold text-xs ${isCurrentUser ? 'text-indigo-100' : 'text-indigo-600'}">${sender}</div>` : ''}

                        <div class="py-1">${message}</div>

                        <div class="text-xs ${isCurrentUser ? 'text-indigo-100' : 'text-gray-500'} text-right">

                            ${new Date().toLocaleTimeString([], {hour: '2-digit', minute:'2-digit'})}

                        </div>

                    </div>

                </li>

            `;

        }

        messages.innerHTML += content;

        messages.scrollTop = messages.scrollHeight;

    }

    function sendMessage() {

        var msgInput = document.getElementById("message-input");

        if (msgInput.value.trim() === "") return;

        socketio.emit("message", { message: msgInput.value });

        msgInput.value = "";

    }

    // Handle Enter key press

    document.getElementById("message-input").addEventListener("keypress", function(e) {

        if (e.key === "Enter") {

            sendMessage();

        }

    });

    // Load previous messages

    {% for message in messages %}

    createChatItem("{{ message.message }}", "{{ message.sender }}");

    {% endfor %}

</script>

{% endblock %}

**main.py :**

from \_\_future\_\_ import annotations  # Must be at the very top

from flask import Flask, request, render\_template, redirect, url\_for, session

from flask\_socketio import SocketIO, join\_room, leave\_room, send

from string import ascii\_letters

import random

app = Flask(\_\_name\_\_)

app.config["SECRET\_KEY"] = "supersecretkey"

socketio = SocketIO(app)

# A mock database to persist data

rooms = {}

def generate\_room\_code(length: int, existing\_codes: list[str]) -> str:

    while True:

        code\_chars = [random.choice(ascii\_letters) for \_ in range(length)]

        code = ''.join(code\_chars)

        if code not in existing\_codes:

            return code

@app.route('/', methods=["GET", "POST"])

def home():

    session.clear()

    if request.method == "POST":

        name = request.form.get('name')

        create = request.form.get('create', False)

        code = request.form.get('code')

        join = request.form.get('join', False)

        if not name:

            return render\_template('home.html', error="Name is required", code=code)

        if create != False:

            room\_code = generate\_room\_code(6, list(rooms.keys()))

            new\_room = {

                'members': 0,

                'messages': []

            }

            rooms[room\_code] = new\_room

        if join != False:

            # no code

            if not code:

                return render\_template('home.html', error="Please enter a room code to enter a chat room", name=name)

            # invalid code

            if code not in rooms:

                return render\_template('home.html', error="Room code invalid", name=name)

            room\_code = code

        session['room'] = room\_code

        session['name'] = name

        return redirect(url\_for('room'))

    else:

        return render\_template('home.html')

@app.route('/room')

def room():

    room = session.get('room')

    name = session.get('name')

    if name is None or room is None or room not in rooms:

        return redirect(url\_for('home'))

    messages = rooms[room]['messages']

    return render\_template('room.html', room=room, user=name, messages=messages)

@socketio.on('connect')

def handle\_connect():

    name = session.get('name')

    room = session.get('room')

    if name is None or room is None:

        return

    if room not in rooms:

        leave\_room(room)

    join\_room(room)

    send({

        "sender": "",

        "message": f"{name} has entered the chat"

    }, to=room)

    rooms[room]["members"] += 1

@socketio.on('message')

def handle\_message(payload):

    room = session.get('room')

    name = session.get('name')

    if room not in rooms:

        return

    message = {

        "sender": name,

        "message": payload["message"]

    }

    send(message, to=room)

    rooms[room]["messages"].append(message)

@socketio.on('disconnect')

def handle\_disconnect():

    room = session.get("room")

    name = session.get("name")

    leave\_room(room)

    if room in rooms:

        rooms[room]["members"] -= 1

        if rooms[room]["members"] <= 0:

            del rooms[room]

        send({

        "message": f"{name} has left the chat",

        "sender": ""

    }, to=room)

if \_\_name\_\_ == "\_\_main\_\_":

    socketio.run(app, debug=True)

**requirements.txt :**

bidict==0.23.1

blinker==1.8.2

click==8.1.8

colorama==0.4.6

flask==3.0.3

Flask-SocketIO==5.5.1

h11==0.16.0

importlib-metadata==8.5.0

itsdangerous==2.2.0

jinja2==3.1.6

MarkupSafe==2.1.5

python-engineio==4.12.1

python-socketio==5.13.0

simple-websocket==1.1.0

werkzeug==3.0.6

wsproto==1.2.0

zipp==3.20.2

**\**

**Updates :**

**Here i need to store the previous room i had joined and show at start page show as My conections at side with room name and without entering code i need to enter that room and i need have my image icon and along with room name need see total members in room**

* **start page :**

**list all rooms with room icon, room name, total participants, last opened**

**and keep a button at top new eoon and then view this connect chat form to enter into nre chat**

**from start page listing a;; my previous rroms i no need to enter room code again and need to enter**

**directly into room and save all msgs in the room**

**and make sure there will be profile page also at top in start page where i can have my profile pic, name, contacn and other details which are actually in the room details also**

* **Room page :**

**The only person who created room has option to edit room name, room icon and remove memebers in group.**

**I can change the chat wallpaper .**

**i can customise the messages for room for individual participants as their own.**

**Can view room propife. can view member profile**

**update the chat text box with included emoji's, audio, media, doc, files transfer**

**store all chat in local storgae for now.**

**it shoild looks like whatsapp right now**

**Update these features adn functionalities for my app and give all files in detailed now.**