

Thematic Chatrooms Documentation

Chapters

Introduction.....	2
Usage.....	2
User Authentication.....	2
User Interface.....	3
API.....	7
Production.....	7
Set Up.....	8

Introduction:

The thematic chatrooms project is a web application that offers users chatrooms to join, based on different themes, such as music and art. Inside the chatrooms, the users can interact with other users in real-time conversations.

Usage:

The core usage of the Thematic Chatrooms app includes:

Account Log in: connect to the app with your username and password (create an account if you don't have one).

Theme selection: choose a chatroom theme from the available list.

Join a chatroom: Join the specific chatroom by clicking the "Join" button.

Chat interface: send and receive text messages from other users in the same chatroom as you, in real time.

User Authentication:

The web app offers a secure user authentication system:

Sign Up: Creates a new account with a unique username and strong password.

Log In: Account access by providing the username and password.

Passwords are securely encrypted before being stored in a database.

Certain functionality will force the users to follow the specific characteristics for a username and passwords.

Username:

- 1) Must contain only letters and numbers.
- 2) The length must be between 5 to 14 characters.
- 3) Cannot contain any spaces.

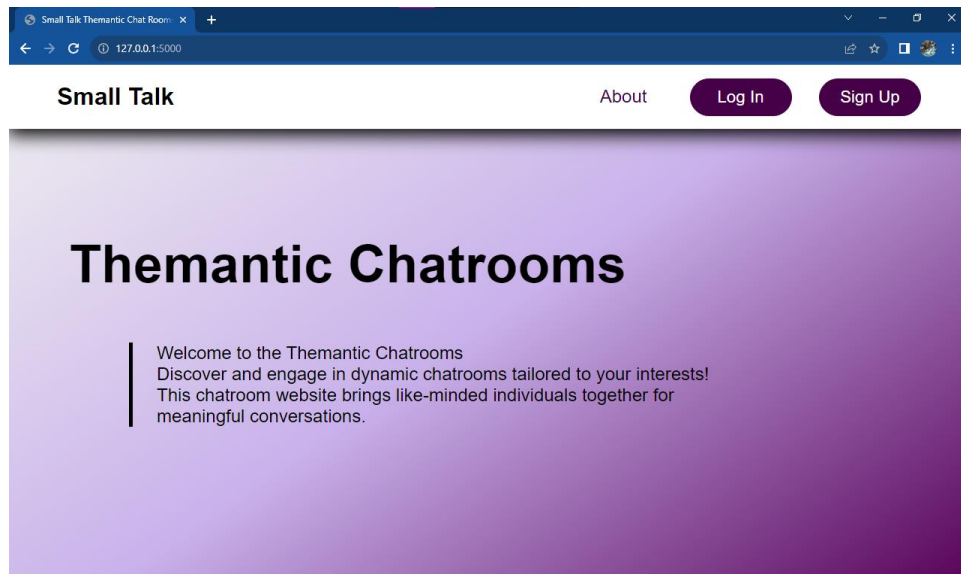
Password:

- 1) Must contain letters, digits and special characters (@, !, %, \$, #, &, *).
- 2) The length must be between 5 to 14 characters.
- 3) Cannot contain any spaces.

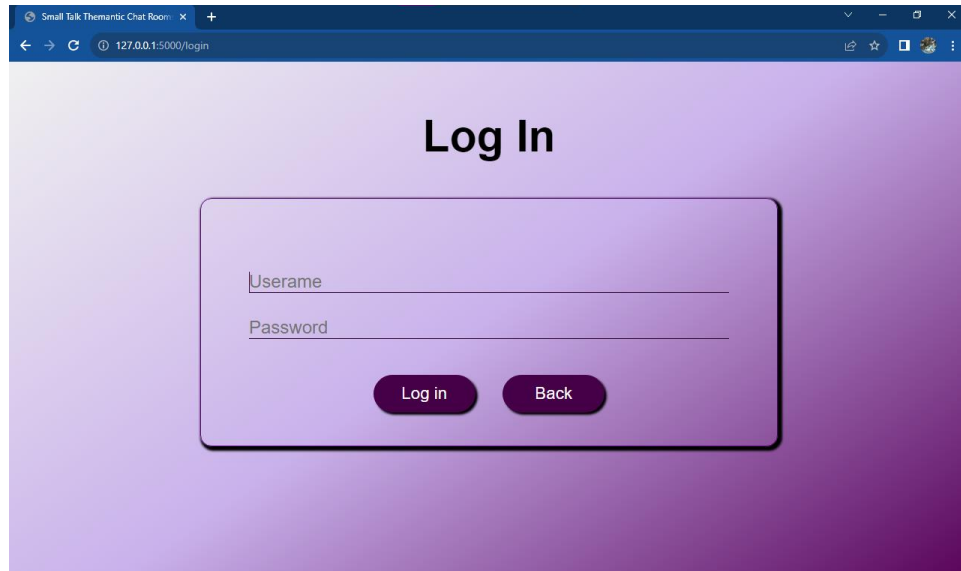
User Interface:

The user interface consists of:

The opening panel:



The sign up and log in forms:



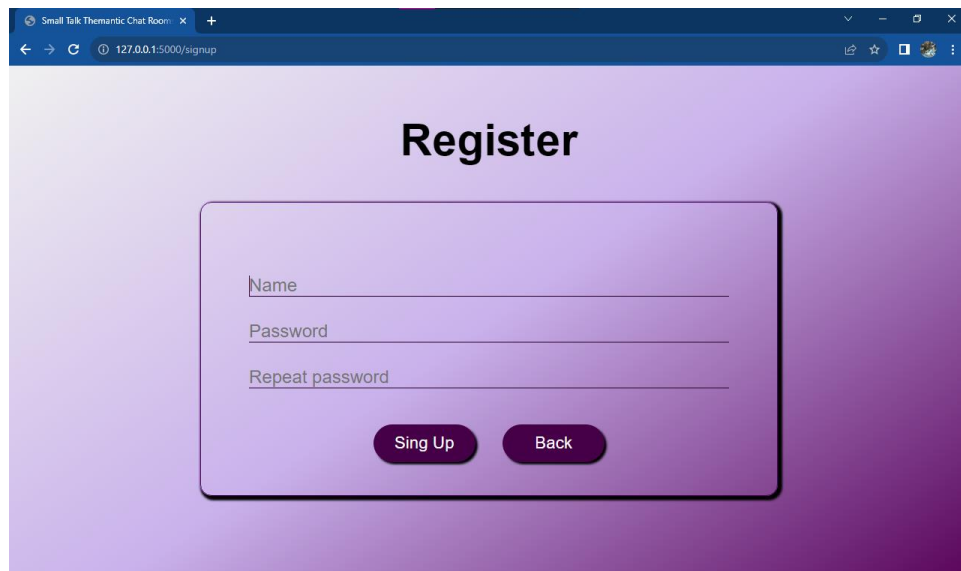
A screenshot of a web browser window showing the 'Log In' form. The browser's address bar displays '127.0.0.1:5000/login'. The form is centered on a purple gradient background and is enclosed in a rounded rectangle. It features two input fields: 'Username' and 'Password'. Below these fields are two buttons: 'Log in' and 'Back'.

Log In

Username

Password

Log in Back



A screenshot of a web browser window showing the 'Register' form. The browser's address bar displays '127.0.0.1:5000/signup'. The form is centered on a purple gradient background and is enclosed in a rounded rectangle. It features three input fields: 'Name', 'Password', and 'Repeat password'. Below these fields are two buttons: 'Sing Up' and 'Back'.

Register

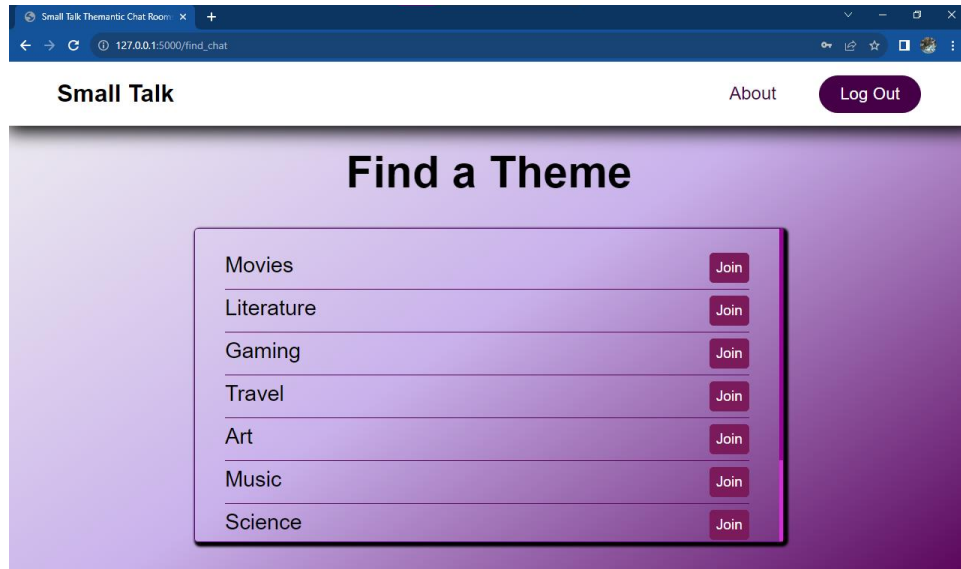
Name

Password

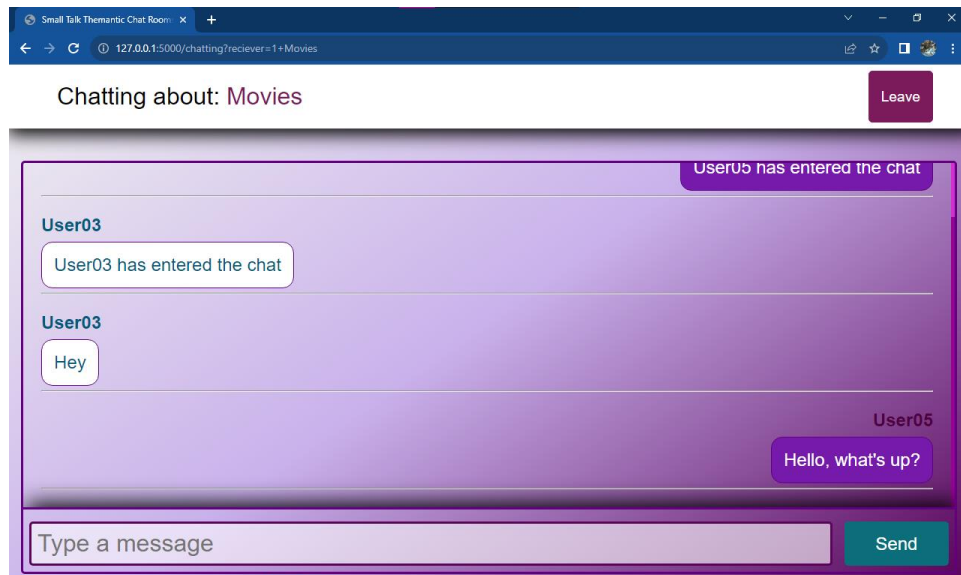
Repeat password

Sing Up Back

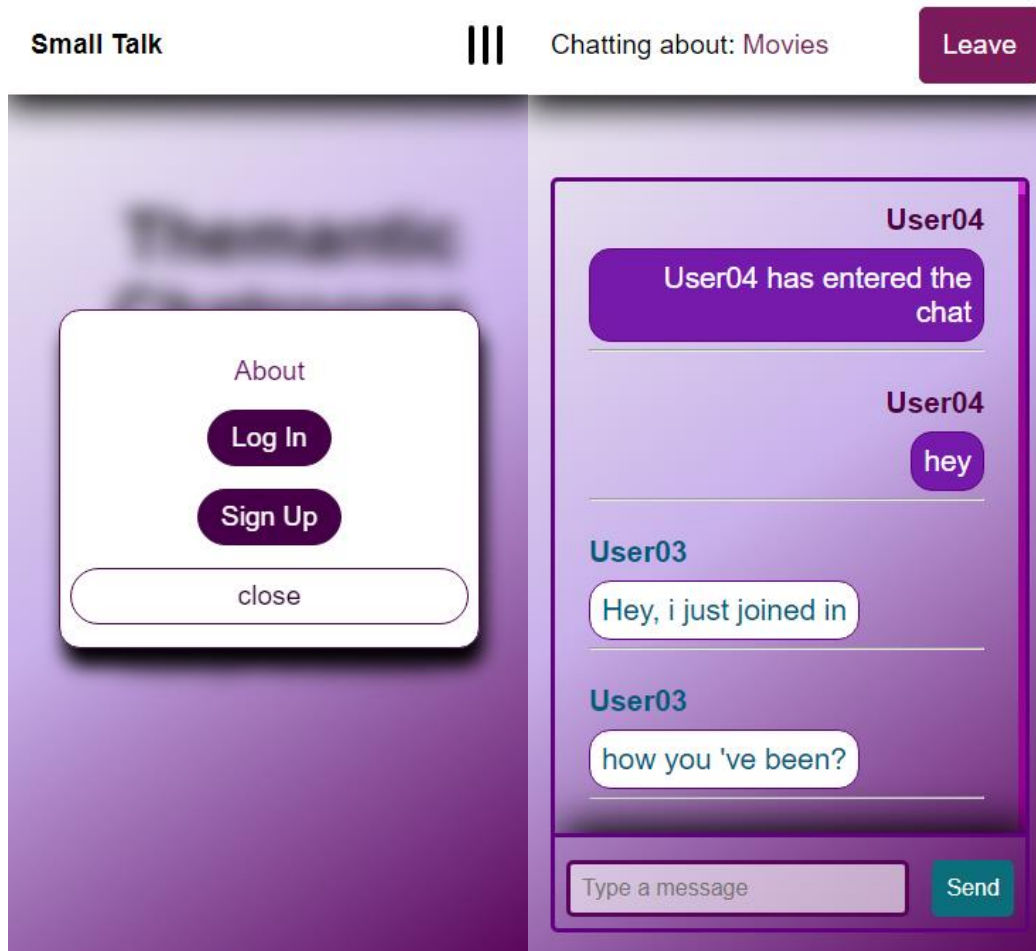
The thematic choice list:



The chatting interface:



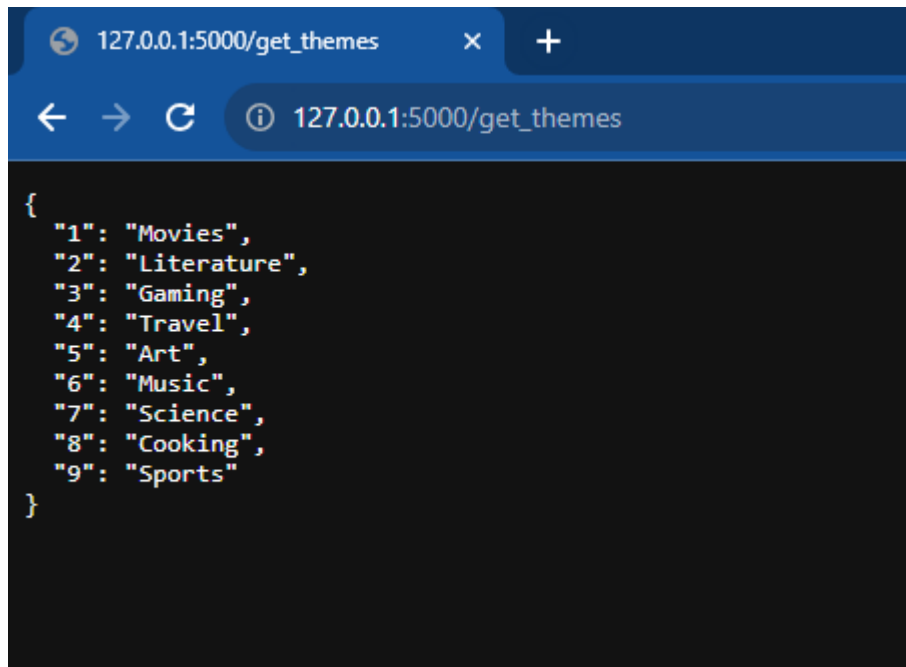
The app's user interface is customized for smaller screen sizes (phones & tablets) and includes animations, for smoother transitions and a better experience.



API:

The app offers an API for easier access to the current themes.

Api: get_themes



Production:

The technologies that were using during production are:

- 1) Python
- 2) Flask, python framework
- 3) Sqlite3 (SQL database)
- 4) HTML5
- 5) CSS
- 6) JavaScript

Python Files:

- 1) main.py: contains most of the python functionality.
- 2) account_standard.py: contains the functions that define the characteristics of passwords and usernames.
- 3) database.py: contains the code that defines the database schema and all the necessary connections.

- 4) chat_themes.py: contains a single dictionary with all the available themes. Adding or removing an item from the dictionary will change the available themes of the app.

Database File:

- 1) UsersDatabase.db: the database that stores all the accounts (usernames and encrypted passwords.).

Templates Files:

- 1) Base.html: it is the base html that contains a wrapper element, all the other templates files are alterations of the content of the base element.
- 2) index.html: contains the opening panel (also includes necessary JavaScript).
- 3) signUp.html: contains the sign-up form.
- 4) logIn.html: contains the log-in form.
- 5) findComv.html: contains the list of all the available themes.
- 6) chatting.html: contains the chatting interface and the socket code, responsible for the real-time communication, in combination with python.

Static files

- 1) styling.css: includes all the CSS code for the UI's styling.
- 2) bars-solid.svg: an image, acting as a button for the mobile navigation.

Dependencies:

- 1) flask
- 2) flask_socketio
- 3) hashlib

Sockets inclusion (head, script tag):

```
src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.0.1/socket.io.js"
integrity="sha512-
q/dWJ3kcmjBLU4Qc47E4A9kTB4m3wuTY7vkFJDTZKjTs8jhyGQnaUrxat0Ytd0ssMZhbNua9hE+E7Qv1j+
DyZwA=="
crossorigin="anonymous"
```

Set Up

To set up the project locally:

- 1) Clone the repository:


```
git clone https://github.com/leTabs/Thematic-Chat-Rooms
```

2) Install dependences:

```
pip install "dependency"
```

Note: pip is the standard way of installing python third party libraries. There are additional ways of installing third party python libraries.

It is good practice to include the dependencies files and the project files in the same directory.
