[ZANG] - [Chuanjie]

Student ID: 699382953

[chuanjiezang@gmail.com](mailto:chuanjiezang@gmail.com)

Analysis of the Chatting Application

I designed our chat application for real-time interaction, prioritizing essential features like user authentication, message exchanges, and maintaining chat logs. During the analysis, I emphasized data security, scalable infrastructure for an increasing user base, and an intuitive GUI for seamless interaction. I aimed for a modular architecture separating the GUI, server logic, and database to ease updates and maintenance.

Design of the Chatting Application

In the design phase, I laid out a comprehensive plan covering the database structure, application architecture, and GUI. I created a database schema to manage user details and messages securely. Adopting the MVC pattern, I ensured a clean separation of concerns—enhancing code manageability. The GUI was crafted to be user-friendly, displaying messages neatly and simplifying the login/logout process.

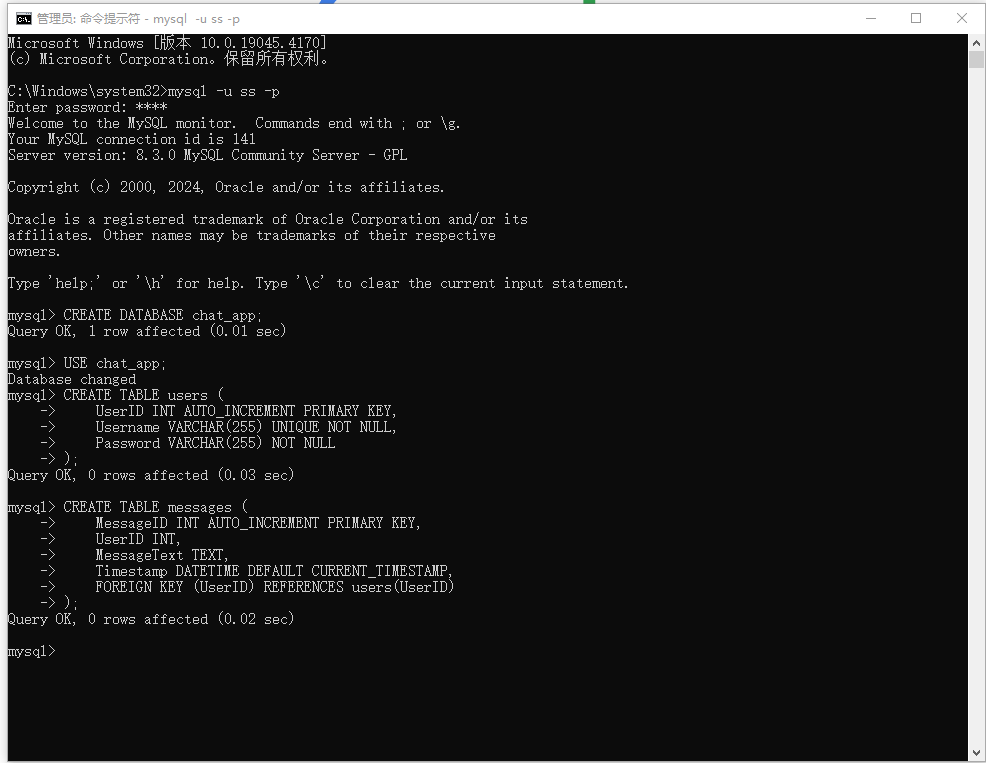
Communication Between the Code and the Database

Our application communicates with the database via a DBMS interface, employing SQL for data operations. The backend, developed in PHP, interacts with a MySQL database through PDO or mysqli, securing SQL query execution. I used prepared statements to safeguard against SQL injection, with database interactions encapsulated within the MVC's model component for user and message management.

GUI Design

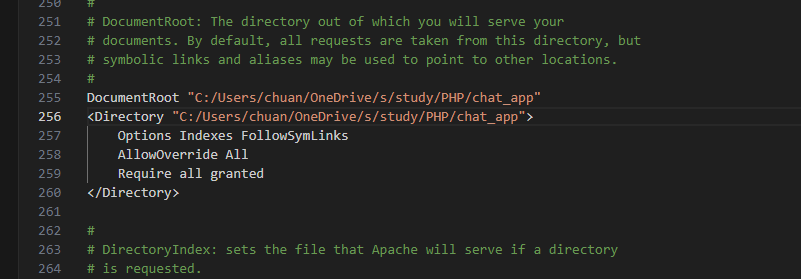
The GUI design focuses on user experience, balancing simplicity with comprehensive functionality. It features a chat window for chronological message display, a text input field, and a send button. Responsive design ensures the app's adaptability across various devices. I also considered enhancing engagement through online user displays, read receipts, and new message notifications. The GUI's minimalist aesthetic reduces distractions, offering customization options for personalization.

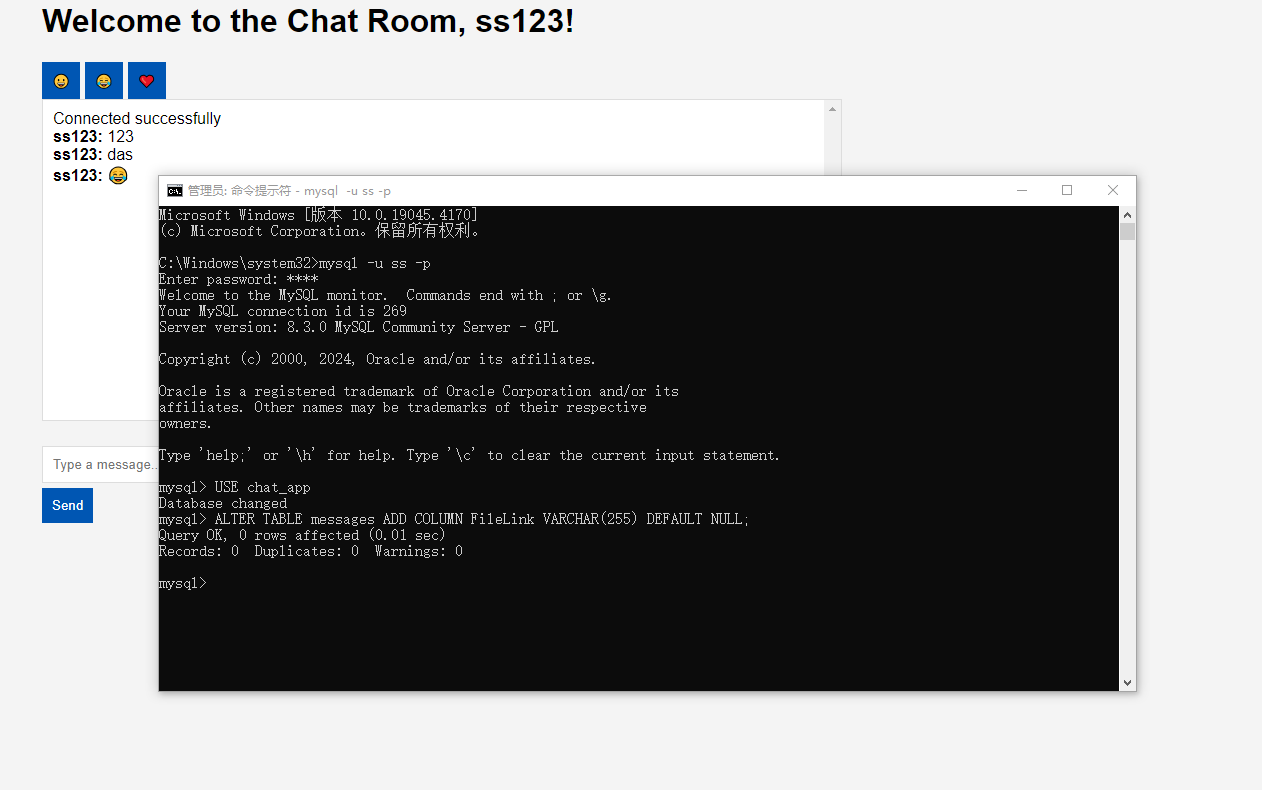
First step: create database and form.



Create the files register.html, register.php, db\_connect.php in the chat\_app folder.

Set httpd.conf in Apache24 and restart Apache24 service.





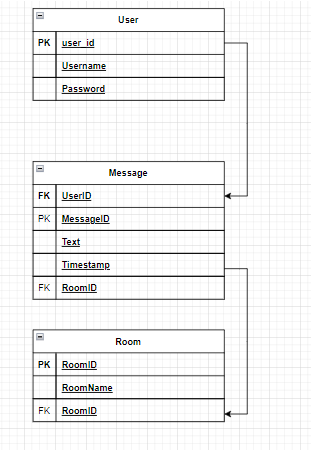
Here is how i set the MySQL, Im using ss as mysql name, password:7147

I use database: chat\_app

And create tables to store.

1. **Entity Relation Diagram**

I use drawio to draw.



1. **Data Flow Diagram**

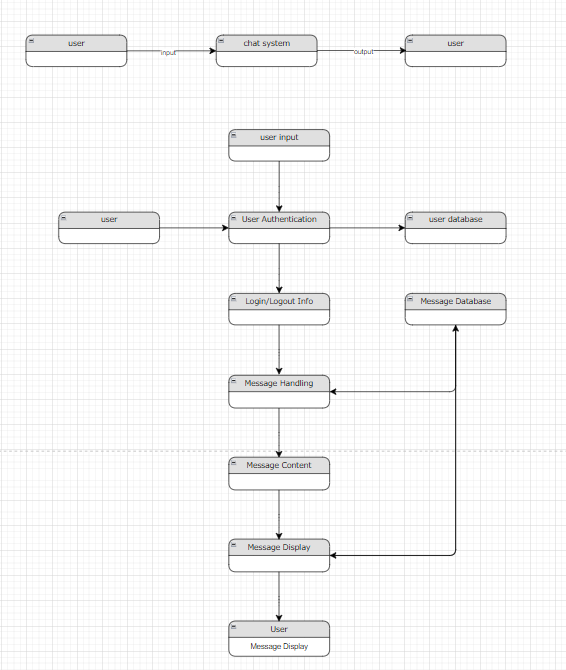
Description:

Users interact with the chat system by entering login information (username and password).

The user authentication process verifies the user's identity and the data flows to the user database for verification.

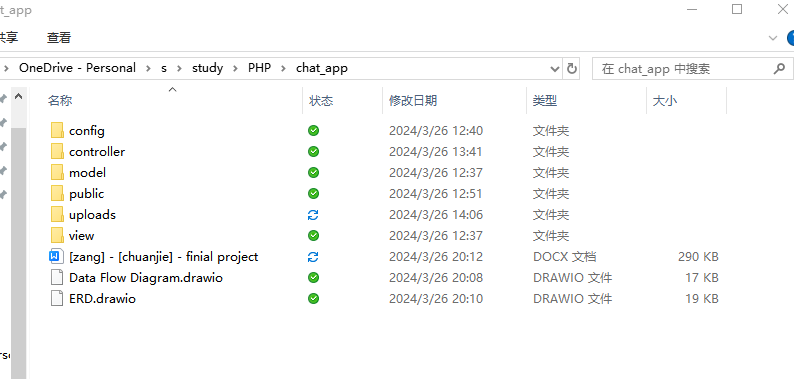
Authenticated users can send messages and the message handling process ensures that messages are stored and delivered to the correct recipient or chat room.

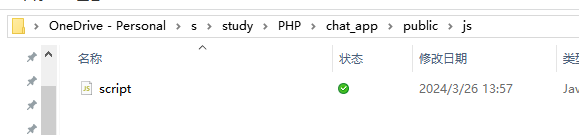
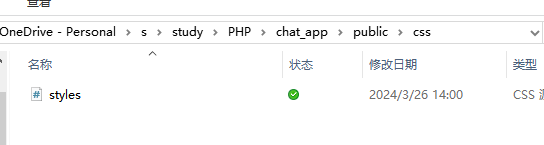
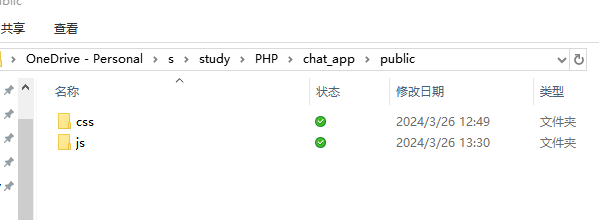
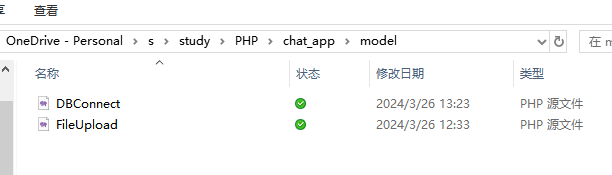
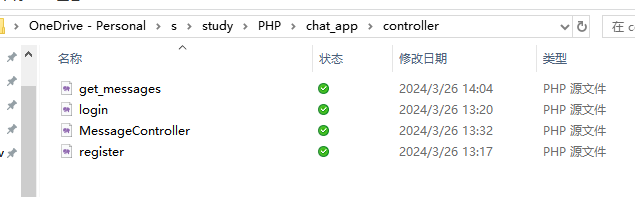
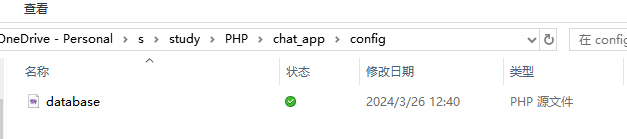
The message presentation process is responsible for retrieving messages from the message database and presenting them to the user.

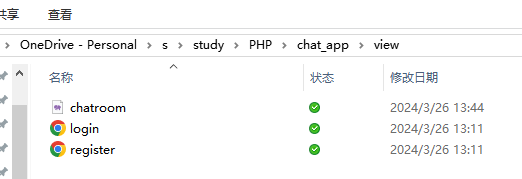
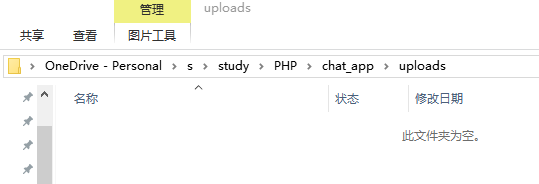


MVC:

Note that I set the root directory to the "chat\_app"







To use the program's process, input ‘localhost/view/login.html’

[ZANG] - [Chuanjie]

Student ID: 699382953

[chuanjiezang@gmail.com](mailto:chuanjiezang@gmail.com)