Open-Source Report

Proof of knowing your stuff in CSE312

[Flask]

General Information & Licensing

Code Repository	https://github.com/ahssanja/CSE_312_Group_Project
License Type	BSD
License Description	• Is a permissive, open-source license that allows for unrestricted use even in commercial applications. • Requires attribution to be given to Flask in any software distributions. • Includes a disclaimer of liability for any damages that arise from use of the framework.
License Restrictions	• Flask is a micro web framework, so it may not be the best choice for large scale applications. • Not good for dynamic web pages and has limited support for async programming. • It does not provide many built-in security features so developers must implement these themselves.

Magic

Dispel the magic of this technology.

ANSWER

The technology used in the code is a combination of several libraries and frameworks, including Flask, SocketIO, and pymongo. After creating a TCP socket, the Flask framework listens for HTTP requests on a specified port. When a request is received, Flask routes the request to the appropriate function based on the URL.

For example, when the user accesses the URL "/", Flask calls the "loginhtml()" function, which sends the HTML file "LoginPage.html" as a response. Similarly, when the user submits a form with the method "POST" to the URL "/MadeNewAccount", Flask calls the "made_new_account()" function.

Within the "made_new_account()" function, the form data is extracted and validated. If the data is valid, it is stored in a dictionary and passed to the MongoDB database using the pymongo library.

When the user submits a form to the URL "/LoggedIn", the "login()" function is called. The email and password from the form are extracted and compared to the values stored in the database using the pymongo library. If the login details are valid, the user is redirected to the "LandingPage.html" file using Flask's "send_file()" function.

The SocketIO library is used to implement real-time communication between the server and client. When the client sends a "POST" request to the URL "/lookingforplayers", the server adds the client's IP address to the "lobby" list. When the client sends a "GET" request to the URL "/checklobby", the server checks if there are at least two players in the "lobby" list. If there are, the server creates a new "game" by removing the first two players from the "lobby" list and adding them to the "games" list. The server then sends the game ID to the clients using Flask's "jsonify()" function.

The code that uses the technology is spread across multiple files and functions, but the main logic can be found in the "app.py" file. Specifically, the "made_new_account()" and "login()" functions interact with the database using the pymongo library, while the "looking_for_players()" and "check_lobby()" functions use the SocketIO library to implement real-time communication.