Project 1 SY306

Team Name: The Lean Meme Dream Team

Laurel Wynn

Chris Eatmon

Thomas Miller

Zach Makkonen

**Descriptions of project roles:**

Laurel and Thomas implemented the index.html, config.py, index.py, signup.py and users.py files that implemented the log in and sign up portions of the messageboard. In the log in function inside of the index.py file, a session ID and cookie is created. Thomas and Chris created the messageBoard.css file.

Chris and Zach both worked on the messageBoard.py file, which generates the messageboard and messages after a user logs into the website. It checks the session cookie to determine the role of the user. Chris tested the session ID functionality and implementation of the message board portion of the website. Zach tested logout implementation and functionality, as well as troubleshooting for various portions of the project.

**Users:**

Username: ‘m197102’ Password: ‘1234’

Username: ‘admin’ Password: ‘password’

**Technical Overview:**

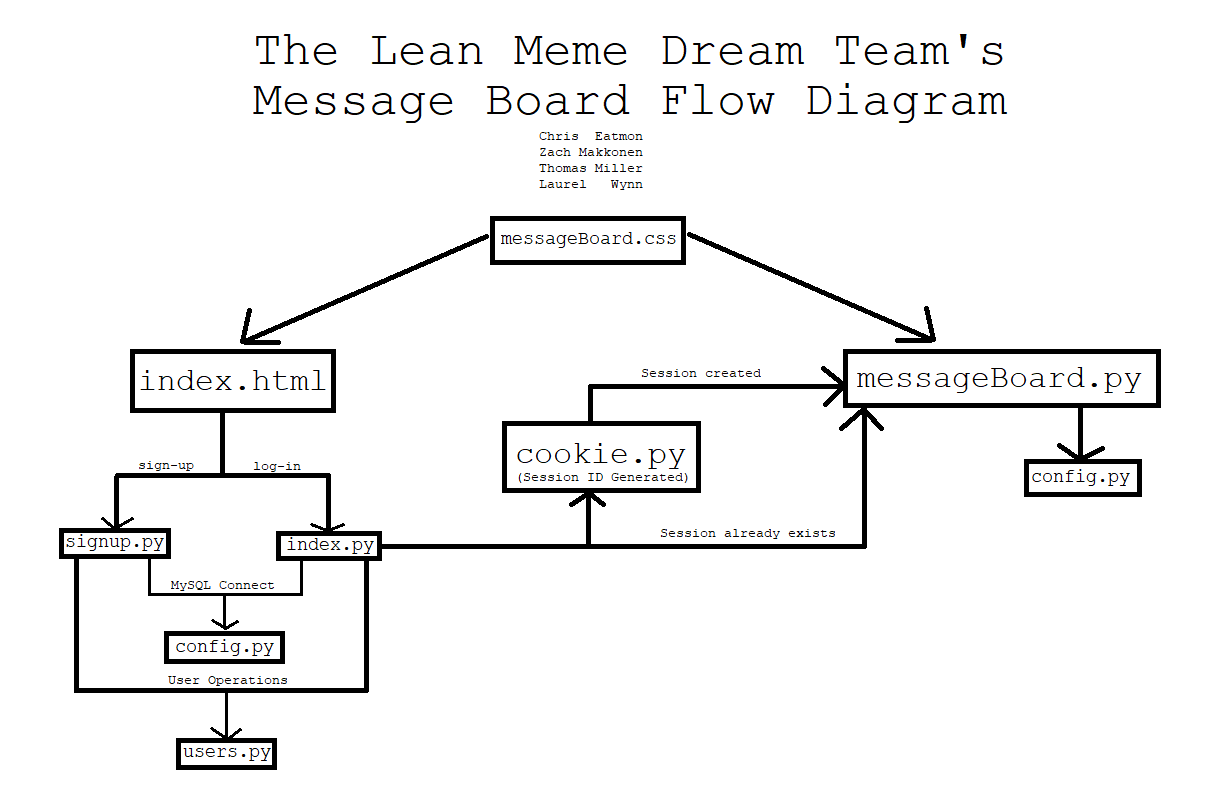
Upon first visit to the site, the user will be met with the index.html page which has HTML forms for logging in, as well as signing up for the message board. If the user desires to login, they will enter credentials, which will be checked by using the index.py script, which will take the username and password, and run a mySQL query to make sure that the user exists, as well as that the hashed password stored on the server matches the hashed password that the user entered. If the query is successful, a session ID will be created by hashing the current time, the system will open a session ID file by that name located in /tmp and write to it the username and permissions level (regular for any user signing up on index.html). Then cookie.py used to create a matching cookie with the same ‘sid’ field. The cookie is then printed to be stored by the browser. After this, the user will be redirected to messageBoard.py. If the query failed to find matching credentials, the user will be redirected to index.py, which will display an error message and give the user a button to go back to index.html to try again.

To sign up from index.html, the user enters a username, desired name and password. The user is then redirected to signup.py which will check to ensure that the username isn’t already taken. If it is already taken they will see a message indicating this and a button to redirect them to index.html. If the SQL query does not return a matching username, a SQL INSERT statement will insert the username, name, hashed password, and permissions level into the Users table. The user will then see a welcome message and will be redirected to index.html to login via a button.

Once the user successfully logs in, messageBoard.py will craft HTML that displays the username, button for logging out, a table of messages populated from a SQL SELECT \* query from the Messages table (each message will display the poster, time, and message content). The message will additionally have a button for deletion if either the logged in user was the one who posted it, or if the user has administrator-level permissions. There will also be a textarea field generated, which will allow for the posting of a new message on submission.

If a user chooses to logout, messageBoard.py will handle deletion of the session ID file on the server, as well as handling the client-side cookie deletion. After this, they will be redirected to the login page (index.html). If they delete a message, messageBoard.py will use a SQL DELETE statement to delete the associated message by utilziing the messageID parameter, and then reload the page. If a user submits a new message, the message will be created by a SQL INSERT statement, and the page will be reloaded again. All associated python and html files utilize messageBoard.css to ensure consistent, stylish formatting.

**Files:**

****

**Control flow description:**

Index.html

The main home page of the site. This gives a form to log in and another for a user to sign up. Ity uses messageBoard.css for the main style sheet.

Config.py

Gives the logon information for the SQL connection as well as creates a header and footer for the messageBoard.py file.

Index.py

The index.py file is called when a user attempts to sign into the webpage. It connects to the SQL database and then calls a log on function which uses a function called authenticateUser from the user.py file to check if the user and a SHA-512 hash of the password corresponds to the data in the database. This uses a SQL prepared statement to perform a select query and retrieve the hash stored as the users password. Depending on the result of this function, an error is printed or the user is successfully logged on. If successfully logged on, the main function of cookie.py is called to generate a session cookie and then the page is redirected to the messageboard itself in the messageBoard.py file.

Signup.py

Signup.py is called when a user submits the form to create a user account. It establishes a connection to the database and then attempts to add the user to the database using the addUser function in users.py. if the user is already in the database it gives an error page. If not, it adds the user, the user’s given name, a SHA-512 hash of the password, and the default role of ‘regular’ to the database. The role of ‘admin’ must be added when connected to the database itself and cannot be changed on the website. If a user is successfully added, it prints out a confirmation message and gives a button that links a user back to index.html where they can now log in. h

Cookie.py

Creates a session cookie, when once a user is verified to log in in index.py.   
messageBoard.py

When a user is successfully logged into the page, this file is called and generates the HTML code for the message board messages submitted. This file checks the session ID cookie for the user role. If valid, it generates a table of the messages and gives the option to add messages, and deletes them if the user role is ‘admin’. It implements this with several different functions in the code.

messageBoard.css

This file gives the CSS for the index.html, messageBoard.py pages, as well as the various error pages that may be generated at any point in the log in or sign up process .

**SQL Statements for table generation:**

**1.CREATE TABLE statement used for generating Users table**

CREATE TABLE Users(  
Username varchar(35) not null,  
 Name varchar(35) not null,  
 Password varchar(256) not null,  
Role varchar(35) not null,  
CONSTRAINT PK\_USER PRIMARY KEY (Username)  
);

**2.CREATE TABLE statement used for generating Messages table**

CREATE TABLE Messages(  
 Username varchar(35) not null,  
 Message varchar(255) not null,  
 TDate varchar(30) not null,   
 CONSTRAINT PK\_DATE PRIMARY KEY(TDate),  
 CONSTRAINT FK\_USER FOREIGN KEY (Username) REFERENCES Users (Username) ON DELETE NO ACTION On UPDATE CASCADE  
);