Micky Batista Melo

IS211 Final Project

Username: admin

Password: password

For the final project, I decided to do a blog application. I first started by creating a database named ‘blog’ with a table named ‘posts’ and columns named ‘post\_id’, ‘created’, ‘title’, ‘content’, and ‘author’. For the ‘created’ column, I used the SQL timestamp function to create a timestamp every time a post is created. To connect the application to the database, I created a function named ‘get\_db\_connection’ that I use in other functions, where database connectivity is needed.

The app consists of a root URL (‘/’), which displays a list of the posts that have been created on the app and are stored on the ‘blog’ database. These posts are listed in reverse chronological order. To do this, I created a html template named index that display a post’s author, title, timestamp and has a reference to a URL link that displays a posts content by post id. To display the posts from the database, I created a for loop in index.html, which gets the post id, title, author, and timestamp. There is a function named “index”, which renders the index.html template and has a SQL SELECT statement to choose the data that needs to be displayed, and an ORBER By statement to order the posts in chronological order by ordering ‘created’ (the timestamps) in descending order

The next step in creating the app was creating a login interface to grant access to the blog’s dashboard. This interface is located at the ‘/login’ route. There is a function named “login”, which requests a username and password from the user. I decided to make the username, “admin” and the password, “password”, and instructed the program to display an error if the values do not match and if they do match, to redirect the user to the ‘/dashboard’ route. The html template for login contains a html form for the user to input this information.

The dashboard of the blog displays the post id and title of the posts that have been created and allow the user to edit or delete these posts and to also create new posts. The dashboard html template has a table in which a for loop was used to display the post id and title and the data is retrieved by a function named dashboard, which selects and retrieves this data by using the SQL SELECT statement.

A function named edit(id) updates the title, content, and author of a post by id, using SQL UPDATE statement to modify these values. The values are obtained and stored in the database by using the “GET” and “POST” method, which request and send data to the server. The html template named edit has a form for the users to insert these values. At the dashboard html template, there is a label that redirects the user to the ‘/<int:id>/edit’ route to allow the user to do this. The delete function works like the edit function but uses the SQL DELETE statement to delete the selected item and there is no html template for the delete function.

To add a post, there is a function named create that uses the “GET” and “POST” methods to obtain and store a post’s title, author, and content. I also used the flashing system to displayed messages if the user failed to include a title, author, or content. The user’s input is inserted into the database by using the SQL SELECT statement, and the user is then redirected to the ‘/’ route, where the posts are displayed. To allow the user to input their data, there is a html template named create that has a form, which instructs the user to input a title, author and content for the post.