Final Project Overview

For my project I created a web application that allows users to share and maintain a log of different exercises that they complete. I completed this project using the Django framework as it has built in features to handle APIs, databases, and website hosting. For the most part I feel like the overall build of my project follows a model view controller pattern.

The models that I created were a user class and an exercise class. The user class contains information on the users that are going to be creating exercises such as height and weight which is used for calculating calories burned during activities, while also containing basic information for an account such as a username and password. The exercise class contains information about the exercise that is being posted such as time, type, date, distance, and other similar fields. It also has a foreign key that links it to the user that created it.

The most interesting part of Django to me is how the views and urls files work together as the view and controller of the web app. The views class contains all the logic on different functions that the app can do and the urls file then directs a user to the correct page based on what they do and what they are wanting to see. The pages that I have in my project are a login page, I know my login is not done in a secure way, but I didn’t realize at the time of creating my project that Django had built in methods for authenticating users. When I realized it, I would have had to change a large amount of my project. The home page shows all exercises posted to the app and every page has a menu that allows the users to navigate to each different part of the app.

There is also an API that I created to return JSONobjects that would allow third-party connections to the webapp and database. The java app is also a model view controller but is set up differently. The user and exercise models still have there own class but the view and controller are in the same class but the controller are private classes within the overall class for the JFrame.

Overall, I enjoyed learning about the different structures that projects can be built with. I feel like using both Django and to an extent java swing shows a lot of similarities but also the differences that come with each. In Django it is much more designed to be on the web and be used as a backend the frontend capabilities seem a lot more limited than java swing with JFrame and the other libraries that are associated with it. That also could be because I am not the most creative with front end especially html.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated