Daily workout- a workout management aid

Brief:

The overall objective for this project, as highlighted in the project specification, was to create an application which entails the following functionalities: create, read, update and delete (CRUD.)

I intend on creating an application that is specifically designed for an audience that likes to keep fit and healthy. My app will allow users to write a list of exercises they wish to carry out either at home or at the gym. In addition to this, users will be able to update their list of workouts as this is a variable that is likely to change throughout their fitness journey. To increase the efficiency of the app, I have included a delete function as well as an update function so users can easily tick off the exercises as they have completed them. Upon completion of an exercise, the user can click on the ‘finish exercise’ icon to update the list.

Throughout this project I have had the opportunity to utilise a number of supporting tools and technologies, including the following:

* SQL-Alchemy (extension)
* Python
* Jenkins
* Flask
* Pycharm
* Flask WTForm (extension)

The application was created using Flask, which is a popular web microframework, written in Python. Flask provided me with relevant tools, libraries and technologies needed to successfully build a web application for my chosen purpose. Furthermore, I was able to use an extension called ‘Flask SQLAlchemy.’ This is an object relational mapping tool used with the database to manipulate the data so it can grabbed, received, selected, retrieved and updated. I had access to a variety of useful defaults and extra helpers which aided in the accomplishment of tasks. Another extension used was, WTForms which made the code easier to read and work with. A list of extensions used during this project can be found on ‘Requirements.txt’ on Github.



Risk assessment:

Below is a list of potential risks I had planned in and my plan of action on how these risks could be avoided or minimised.

|  |  |  |  |
| --- | --- | --- | --- |
| Description of risk | Likelihood of risk occurring | Mitigating action (ways in which the risk can be reduced) | Contingent action (actions to be taken if risk occurs) |
| Bug in coding | Medium | -Define keys better.  -Allow time for debugging.  -Unit test the code throughout the process (write a code, test it, fix it.) | - Review GIT hub and reconnect with the last commit prior to the big issue. |
| Server failure | Low | -Commit code to GIT hub to ensure no work is lost if there is a server issue. | -Restart project from the last commit. |
| GCP server does not run code | High | -Expand RAM on your GCP server.  -Use another cloud service such as AWS.  -Set up GCP server using the correct configurations to avoid potential issues. | -Ensure GCP server is connected correctly. |
| Denied permission from terminal in GCP | Low | -Override by changing permissions. | -Restart GCP |
| Lack of preparation | Medium | -Use an ERD diagram  -Ensure you have sufficient time to meet deadlines. | -Prioritise tasks |
| Syntax errors | Low | -Use the ‘hints and tips’ provided on Pycharm. | -Run code and look for highlighted syntax error.  -Increase knowledge to fix errors. |

Testing:

The testing software used for this project was Jenkins- an open-source automation server that allows developers to reliably test their code. My initial test was carried out to assess whether the delete, add and the done functions were working in accordance with the design brief. I then checked to see if my class function to a robust standard. The testing process was an integral part of the project and also one that I found to be my biggest hurdle when trying to achieve my aim for this project. Having had good success with this topic during the online tutorials in class, I was optimistic that this stage would not pose a problem for me. However, I was unable to get an adequate 70% coverage. The highest I got was 67%, as evidenced by my .coverage history in my github repository. I could have improved my coverage by running tests that were not very useful or practical such as a 404 test. Although these tests would have improved my percentage of coverage, the results would not have impacted my app and its functionality I therefore decided against this and chose to carry out tests that were practical and would enhance my app.

Evaluation:

This project has been phenomenal for my growth as a developer and has provided me with clarity, confidence and a resilience when problem solving. Even though I have faced a variety of issues, I had the patience to persevere and think of alternative mehtods to achieve my goals. If I had the opportunity to carry out this project again, I would ensure I left ample time for the testing stage as I feel I have underestimated the time it takes to set up the correct configurations for testing. I am grateful to have had this issue arise at this early stage in my journey as a developer as it has highlighted areas where I must apply more focus in future projects. I have learnt to never overlook the source of an error, to pay attention to detail and to not overlook the simple tasks. As a result, I have created a checklist of possible minor issues that I could easily overlook when finding the root of an issue and I use this checklist where appropriate. This has increased my efficiency when dealing with issues that arise and I have found that often these issues can be fixed by simply carrying out an action from my checklist.

To further extend my app I would create a log in page for users to sign in on as this would give them a more personalised experience. Another useful feature that could be added in order to further improve my app would be a record of previous work out histories. This would allow users to go back and see what exercises they have completed as well as allowing them to track their workout progress.