**COMP0034 Coursework\_1**

***General introduction of file structures:***

In the rest API, I define 3 tables in the database including player table, trainer table and the data table. The player table and the trainer table are the tables containing information of users. The data table contains the results of output of machine learning models and the input features reading from the excel. Detailed structure of the database is shown in the fig.1.

图形用户界面, 文本

中度可信度描述已自动生成

fig.1 The ERD diagrams of the database

The structure of API module is shown in fig.2:

电脑萤幕的截图

描述已自动生成

fig.2 The structure of the API

***Application part:***

As shown in fig.2, the API contains several parts. \_\_init\_\_.py contains the setting of the create\_app() functions to create the flask instance. I used blueprints to make the structures of the module clearer and easy to debug. bp\_delete, bp\_get, bp\_put and bp\_post contain functions with certain http requests. I deploy the error handlers in the blueprints. Detailed explanation is within code. Extension file is used to import db and ma instances, which avoids location errors when other files importing db and ma. In models.py, I defined the models of the databases from designs of ERD in fig.1. The router file is used for directly call API using code. schemas.py avails packages Marshmallow is used to serialize and deserialize JSON files and database objects in schemas.py. The data file contains the csv file of IMU data.

***Test part:***

Overall, I run 20 tests for the routes in API. Not only that I tested all the functions in blueprints, but also, I tested some edge situations and whether some errors raised normally with the functions in blueprints. I also tested with different values and see whether the test codes run normally.

As the fig.1, test codes locate in the test directory. In conftest.py, I defined some json variables for the player, trainer, and data row, which makes the structure clearer and easier to debug. Other files are used for the testing of certain functions. Detailed explanations are inside the test files.

The fig.3 demonstrates that all the tests have passed, and fig.4 demonstrates the coverage reports:

电脑萤幕的截图

描述已自动生成

fig.3 the test screen shot.

文本

描述已自动生成

fig.4 the coverage reports

From fig.4, we can see that not all the contents of blueprints are tested. That should be the reason of error handler and exception I set. I tested some of the errors but not all of them.

***Tools and Techniques:***

Instead of importing logging packages, I directly allow pytest to report info level of information from the system configuration to help me to debug.

电脑屏幕截图

描述已自动生成

fig.5 the configuration of pytest

Also, I used DB Browser to help me to check the content of database, which helped me to debug the API.

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

Fig.6 the evidence of using DB browser

The menu is in Chinese, but the content is shown in Fig.6.

Also, I use git for version control often; here are part of records:

文本

描述已自动生成

Fig.7 the evidence of using git

Also, I add explanation of file in Readme.md and dependencies in requriment.txt.

文本

描述已自动生成

Fig.8 A part of Readme.md

文本

描述已自动生成

Fig.9 the dependencies in requirements.

Also, I add .gitignore to ignore the virtual environment I used for the project. 图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Fig.10 the exists of ignore file

I used CI to use pytest and deploy the workflow in the GitHub. The .github zip in src file demonstrates the CI test code in yml format.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Fig.11 the exists of yml in workflow.

图形用户界面

描述已自动生成Fig.12 the screen shot of the actions for the projects

I also used black package to lint the files to make the files format in a good style:

文本

描述已自动生成

Fig.13 the evidence of using of black packages.

I define pyproject.toml

电脑屏幕的手机截图

描述已自动生成

Fig.14 the evidence of defining pyproject.toml.

***Improvements I would like to do but I do not have time:***

I would like to change the password type of players and trainers to hash type of password. But the workload for that is large, which needs at least 8 hours to do that. All my test functions and blueprints about it need to change. I hardly have time to do that.

***References:***

*Acknowledgement of AI used:*

Chatgpt4.0 and Chatgpt 3.5(published by OPEN AI) is used in help me for coding and proving solutions for my projects. Some of codes are generated and I correct them to fit my project. Some of the codes like yml for CI is directly generated without any changes from the Chatgpt. They also suggested new methods to use in debugging. They also advised new ideas about some concepts taught in comp0034. Also, I used it to polish some written works in the Readme markdown.

*Cite of the dataset:*

AlSahly,Abdullah. (2022). accelerometer\_gyro\_mobile\_phone\_dataset. UCI Machine Learning Repository. <https://doi.org/10.3390/s22176513>.

*GitHub URL:*

https://github.com/ucl-comp0035/comp0034-cw1i-King-in-black.git