[Fall-2021] 36-650 Course Project

Accept this assignment by accessing GitHub classroom via the following URL:
https://classroom.github.com/a/g3GuDb1B

In this project, you will use FIFA Dataset available on Kaggle
https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset?select=players_20.csv

This dataset contains Soccer player statistics for 2015-2020. In this project, you only need the statistics for 2020 to conduct the following tasks:

General Expectations

- Follow coding best practices with well-documented code.
- Add your dataset under data folder

Task-I: Build and populate necessary tables (30% of course project grade)

- Create the DB table to store the dataset and populate it with data.
- Identify constraints as needed and document them in your Readme.md file.
- Your tables should be created in schema with the name "fifa".
- In your ReadMe.md, attach a screenshot of your table infrastructure (e.g. DbVisualizer screenshot).

Task-II: Conduct analytics on your dataset (40% of course project grade)

Develop Python functions to answer the following questions (given that x, y and z) will be user-entered parameters. Each question should be answered by one or more Python functions. Think about how to implement these requirements with the least possible SQL statements/transactions.

1. List the **x** players who achieved highest improvement across all skillsets.
2. What are the **y** clubs that have largest number of players with contracts ending in 2021?
3. List the **z** clubs with largest number of players in the dataset where z >= 5.
4. What is the most popular nation_position and team_position in the dataset? (list the most popular for each)
5. What is the most popular nationality for the players in the dataset?

Task- III Test your Code and Document it (30% of course project grade)

- Document all your functions properly.
- In the Python functions you developed above, make sure to handle errors properly.
- Develop unit tests to cover ALL the functions that were developed above. Your unit tests should cover happy and sad path scenarios.
- In your Readme file, provide a summary on the scenarios you covered in your unit tests.

Task- IV (20% Extra-credit) Provide Portable, Cross-platform Solution

- Dockerize your application. The end goal is to have docker compose containing Python and PostgreSQL, and all other software/applications you need to run your application.
- Your Dockerfile should encapsulate your source code files as well.
- Push your Docker Image to Docker Hub.
- In your Readme file, provide the details on how to run the Docker Image from Docker hub directly.

Task V (15% Extra credit) – Difficulty level: Challenging

Use Python's Seaborn library to develop the following visualizations. Each visualization should be created using a separate Python Function.

- Develop Python function that graphically displays the 10 players who have achieved the highest improvement across all skillsets
- Develop Python function that graphically displays the 5 players with highest value (value_eur).
- Develop Python function that graphically displays the 10 players with the largest number of player_traits.
    - If there are more than 10 players, e.g. (players 9, 10, and 11 include same number of player traits), include all of them in the visualization.

Submission Guidelines:

- Post URL for your GitHub repository. Use the starter code that is provided above as the starter for your code.
- Your GitHub repository should have a ReadMe.md file that lists the "exact" steps on how to get this application working on a new machine (via Docker). I will follow the steps in your ReadMe file and if I can't get it running on my machine, I will deduct considerable number of points from your project grade.
- You should record a video demonstrating two elements:
    1. Code Walkthrough while you are explaining your code changes.
    2. Demoing the running application while you are navigating through EVERY functionality that is working in your application. I will use this video to help assessing your grade. You may lose points for the functionalities that are not demonstrated in the demo.
- Your video size may be large to be uploaded to GitHub. You may use Box to upload the video and add the URL to your ReadMe.md file in your GitHub repository.
    1. Make sure that your video is publicly shared. Private videos won't be visible to the instructor and TAs and therefore, your project grade will be impacted

Common Penalties:

- Late submissions on Canvas or GitHub: 100% reduction (won't be graded)
- Not submitting the GitHub video (for both code walkthrough and functionality demo): 20% penalty (calculated from maximum project grade).
- Not providing clear details in the ReadMe file on how to run the application (or any variables that need to be updated/replaced): 10% penalty (calculated from maximum project grade)

Suggested Project Task Schedule (includes only core requirements and doesn't include extra-credit):

Week	Task
End of Week-5	Complete Task-I, Write the SQL needed for Task-II
End of Week-7	Complete Task-IV
End of Week-10	Work on and Complete Task-II
End of Week-11	Complete the Error handling and Documentation needed for Task-III
End of Week-12	Complete Task-III