Exercise 2.7: Data Analysis and Visualization in Django

1. *Consider your favorite website/application (you can also take CareerFoundry).*
   Think about the various data that your favorite website/application collects. Write
   down how analyzing the collected data could help the website/application.

Analysis of various collected data (on the CareerFoundry platform), such as user
activity, course progress, preferences and feedback, can help improve the user
experience and optimize the learning process. It can also identify trends in user
behaviour, enabling better personalization, improved course offerings, and better user
retention. Analyzing forum interactions can help identify the most popular topics or
common challenges faced by learners, guiding the creation of new courses or
community events.

2. *Read the Django official documentation on QuerySet API. Note down the*
   different ways in which you can evaluate a QuerySet.

Ways to Evaluate a QuerySet (Django):

- len(): To get the number of records.
- list(): Converts the QuerySet to a Python list.
- exists(): Checks if any records exist.
- count(): Counts the number of records in the QuerySet.
- Iteration: Looping through a QuerySet to retrieve records.
- filter() allows for narrowing down the query results based on specified criteria.
- exclude() helps in filtering out records that do not match the specified criteria.
- order_by() is used to sort the results based on given fields.

3. *In the Exercise, you converted your QuerySet to DataFrame. Now do some*
   research on the advantages and disadvantages of QuerySet and DataFrame, and
   explain the ways in which DataFrame is better for data processing.

Advantages of QuerySet:

- Directly connected to the database.
- Supports lazy evaluation (queries the database only when needed).

- Efficient for simple data retrieval and manipulation in Django.

Disadvantages of QuerySet:

• QuerySets have limitations in data manipulation and transformation capabilities compared to DataFrame.

Advantages of DataFrame:

- More powerful for complex data processing and analysis (e.g., filtering, grouping, and statistical operations).
- Offers better performance for in-memory operations.

DataFrame is better for data processing because it provides a wide range of functions for handling and manipulating data, making it ideal for tasks like analytics, visualization, and large-scale computations.