Exercise 2.7: Data Analysis and Visualization in Django

Learning Goals

- Work on elements of two-way communication like creating forms and buttons
- Implement search and visualization (reports/charts) features
- Use QuerySet API, DataFrames (with pandas), and plotting libraries (with matplotlib)

Reflection Questions

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.**
   The CareerFoundry website collects data about courses and users (students, mentors and tutors). Analyzing the collected data could help the company make sense of it and make decisions. For example, CareerFoundry collects student's opinion and feedback about each exercise in a course. This helps the company make decisions about the courses (they can easily check which are the courses and exercises that need more attention and act accordingly to improve the courses and exercises). They can collect data about submission history to track course progress. This allows the company to reach out to students that are behind schedule providing a better service (possibly improving the success rate of the courses) or to monitor mentor/tutor response times to see if they meet the service level agreements with students. Analyzing students and mentors/tutors data can also help the company make decisions about marketing campaigns, sales efforts, recruitment, etc. The potential of data analysis is endless.

2. **Read the Django official documentation on QuerySet API. Note down the different ways in which you can evaluate a QuerySet.**
   You can evaluate a QuerySet in the following ways:
   - **Iteration.** A QuerySet is iterable, and it executes its database query the first time you iterate over it.
   - **Slicing.** A QuerySet can be sliced, using Python's array-slicing syntax. Slicing an unevaluated QuerySet usually returns another unevaluated QuerySet, but Django will execute the database query if you use the "step" parameter of slice syntax, and will return a list.
   - **Pickling/Caching.** The results are read from the database.
     - If you pickle a QuerySet, this will force all the results to be loaded into memory prior to pickling. Pickling is usually used as a precursor to caching and when the cached queryset is reloaded, you want the results to already be present and ready for use (reading from the database can take some time, defeating the purpose of caching). This means that when you unpickle a QuerySet, it contains

the results at the moment it was pickled, rather than the results that are currently in the database.

- **repr().**
- **len().**This returns the length of the result list.
- **list().**
- **bool().**Testing a QuerySet in a boolean context, such as using bool(), or, and or an if statement, will cause the query to be executed. If there is at least one result, the QuerySet is True, otherwise False.

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.**

   Working with a QuerySet is fine as long as your database is small, with only a few thousand records.

   A pandas DataFrame is a two-dimensional data structure (like a spreadsheet) where data is stored in tabular form in rows and columns. The size of a DataFrame is mutable, which means you can delete or append elements. You can perform arithmetic operations on both row and column labels.

   DataFrame is also better for data processing because of its excellent representation of data, efficient handling of huge data, and extensive feature set (of Pandas).

   Depending on the purpose, a DataFrame may be more performant as it works with data loaded into memory.