ORM Querying Lab

Introduction

Object-Relational Mappers (ORM) give software developers the tools to interact with databases purely through the applications object-oriented programming language (such as Python). Rather than writing raw SQL queries with our applications we can instead interact through class objects called "models". This project is an exercise in becoming familiar and comfortable using ORMs such as Django's ORM (Django covered soon!). In this project we will be interacting with multiple models to query data in unique was and inspect what SQL is actually being executed "under-the-hood".

Technologies

Python, MySQL

Learning Objective

The objective of this project is to get more comfortable using the Django ORM query methods to retrieve data from a MySQL database. Many of your future projects, including the capstone, will require you to use similar Django ORM queries that you will be introduced to here. Your proficiency will grow with each project.

Resources

PowerPoints

- Querying Using an ORM

Other Resources

- Basic Github Operations Document

- Official Django ORM Documentation:

- https://docs.djangoproject.com/en/4.0/topics/db/queries/

- https://docs.djangoproject.com/en/4.0/ref/models/querysets/

Tasks

1. **Complete the required queries in each view.py function.**
2. **Print your query results to the terminal and attempt to match the provided "Expected Terminal Result" as shown in the comments under each function. (Also shown in the browser)**
3. **Utilize the Django Debug Toolbar to view the actual SQL that is executed by your ORM queries.**
4. **Use the web page's navigation buttons to execute the corresponding functions**

Setup Steps:

Work through steps to complete project setup (Also shown in overview video).

1. Create a github repo, select Python gitignore.
2. Unzip the starter code and pushes it to the repo.
3. When opening the project, make sure to open the folder that contains the pipfile as the top level.
4. Create a new database in MySQL workbench named 'school_db'.
5. Update the password in local_settings.py to reflect your personal MySQL password.
6. Enter 'pipenv install' command.
7. Enter 'pipenv shell' command.
8. Change VS Code's python interpreter to reflect newly created venv (see ORM Query Lab Setup video – **minute mark: 3:29**)
9. Enter 'python manage.py migrate' in the correct directory.
10. You can now start the application with the VS Code debugger by pressing F5 or selecting Run > Start Debugging
11. Open up the URL as shown in the terminal (http://127.0.0.1:8000/)