CSC561 NoSQL Databases
Lab 1, Part 3: Relational Database Using PostgreSQL

Part 3: Access from Programming Language (Python/Django ORM)

Pre-Requisite – Successfully complete and receive a grade for Lab 1b. If you complete Lab 1b early and would like me to grade it early, please send me an email. Otherwise, I will grade them right after the due date. After you receive a grade for Lab 1b, I will export your Lab 1b assignment to your GitHub repository and copy the files for Lab 1c into the same share used for Lab 1b. Additionally, I will remove the Laravel container and replace it with a Django container.

1. The tasks we want to complete Lab 1c are to:

    a. Access our PostgreSQL databases from a Django Python app. Django is a Python framework.
    b. Execute a query from Django using Django's Object-Relational Mapping (ORM).
    c. Display our query results on a webpage.

Utilizing a database might be all that you need in the real world, but accessing that database from a programming language such as Python will give you an advantage in the workplace. Knowing how to retrieve, manipulate and save data from a programming language is not a new feature of any programming language or database, so it will be helpful to understand how to set-up and perform some basic tasks, such as the ones we will cover now:

1. Install Visual Studio code integrated development environment (IDE). You can use other Python IDE's such as Atom.
2. Map a drive to the Windows share on the container with the Django code
3. Open up the Django project in the Visual Studio code IDE
4. Modify the Django view and template to execute the queries using the ORM relationships and display the data in the lab

Video Tutorials

Please review these video lectures:

Learning Django – LinkedIn Learning
[1 hour 26 min]:
https://www.linkedin.com/learning/learning-django/welcome?u=43607124&auth=true

Access from Programming Language (Python/Django ORM)
[20 min 16 sec]:
https://cdnapisec.kaltura.com/index.php/extwidget/preview/partner_id/1371761/uiconf_id/13362791/entry_id/1_cvm19bta/embed/dynamic

Writing your first Django app
https://docs.djangoproject.com/en/2.2/intro/tutorial01/

Please be sure to follow the steps in the Lab Setup video linked above to connect to your container. Use this table to determine which container is yours. You will log into the share with .\NetID for the username (.\tllos1 for example) and your UIN for the password.

Netid	Windows share	Url of Django application
agang2	\\10.64.3.56\agang2	https://csc570e.uis.edu:9444
bbala5	\\10.64.3.56\bbala5	https://csc570e.uis.edu:9445
bguti6	\\10.64.3.56\bguti6	https://csc570e.uis.edu:9446
brodr22	\\10.64.3.56\brodr22	https://csc570e.uis.edu:9447
chick7	\\10.64.3.56\chick7	https://csc570e.uis.edu:9448
eunsik2	\\10.64.3.56\eunsik2	https://csc570e.uis.edu:9449
jlund6	\\10.64.3.56\jlund6	https://csc570e.uis.edu:9450
jshei3	\\10.64.3.56\jshei3	https://csc570e.uis.edu:9451
mpavl3	\\10.64.3.56\mpavl3	https://csc570e.uis.edu:9452
rsayy2	\\10.64.3.56\rsayy2	https://csc570e.uis.edu:9453
sarya7	\\10.64.3.56\sarya7	https://csc570e.uis.edu:9454
skoch7	\\10.64.3.56\skoch7	https://csc570e.uis.edu:9455
szhen6	\\10.64.3.56\szhen6	https://csc570e.uis.edu:9456
zwold2	\\10.64.3.56\zwold2	https://csc570e.uis.edu:9457
smeka6	\\10.64.3.56\smeka6	https://csc570e.uis.edu:9458

Relationships will need to be set before writing the queries (Many-to-one (ForeignKey)). For Step 1 and 2, you will need to create a relationship from the Transactions table to the Users table, from the Transactions table to the Inventory table and from the Inventory table to the Status table. The models with relationships have been already implemented for the Status, User and Inventory tables. You will need to implement the Model class for the Transaction table in the checkout/models.py.

Your results will be the template checkout/templates/home.html, which already maps to the route of the root of the website. This view will display two tables with the data similar to what I showed in the video example for Status and Inventory (see video link above for Access from Programming Language).

1. Write a query to show all items that user1 has checked out with the corresponding checkout_time and display the results.

2. Write a query to show all users who checked out items before September 3 2018 and display the results, which should include the user first name, the item description, checkout_time and the status.

This is a very important step:

- For the Python/Django application to work and to have the automatic /admin site for your app you need the tables created by the commands (django-admin.py startproject website, python3 manage.py createsuperuser, python3 manage.py startapp checkout, python3 manage.py migrate) In order to create those tables you can download the sql file with the table information from http://csc570e.uis.edu/csc561/django.sql

- You will need to implement the Transactions model in checkout/models.py. Since it is referenced in other files such as checkout/admin.py, if you don't implement it then the server will respond with an Internal Server Error.

- Then insert the data into your database with the command:

  Ex: psql -U postgres -h 10.92.132.11 -p 5432 -d lab1 -a -f django.sql

  (replace the -h with the IP address of your PostgreSQL database and -d with your database name)

  This will make the /admin interface work for your project with the username=root and password=p@ssw0rd2019