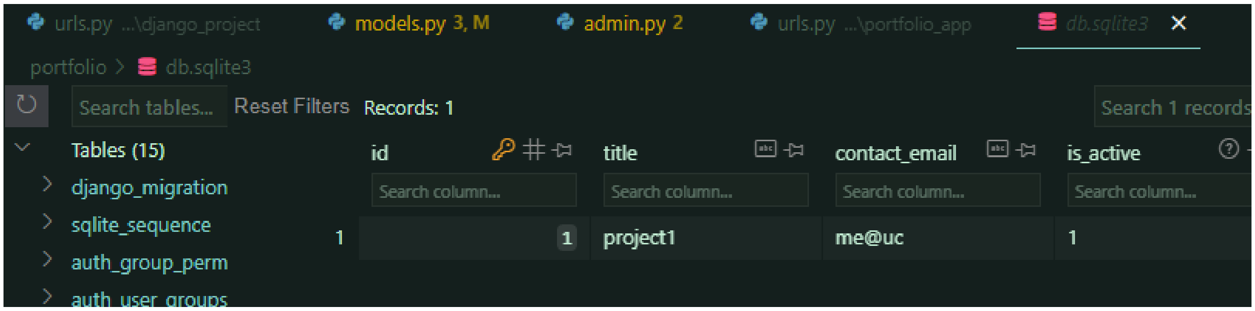
GE03 professional document
Juhan Hong
3/1/2024

Activate Record Pattern:

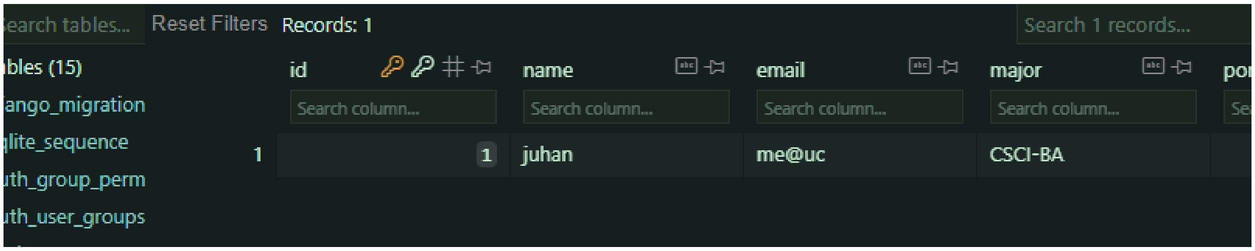
The active record pattern is a design pattern that expresses the application’s logic and data. Through GE03, I was able to experience and explore what that truely meant. With an example given such as the Student model, I was able to create other models that will be shown in the portfolio application such as Portfolio and Project sections. Instances of these classes represent individual records in the database, and methods defined in these classes allow you to interact with the data.

```
class Portfolio(models.Model):
    title = models.CharField(max_length=200)
    contact_email = models.CharField(max_length=200)
    is_active = models.BooleanField(default=False)
    about = models.TextField(blank=True)
```

You are able to witness the database happening in real time through the db.sqlite3 file.



	id	title	contact_email	is_active
1	1	project1	me@uc	1



	id	name	email	major	po
1	1	juhan	me@uc	CSCI-BA	

Object-Relational Mapping:

Django has an incredible ORM system that allows you to interact with the database using Python objects. Each model class I created in the GE03 gets represented in the database table as shown above. In GE03, I was able to create relationships such as one to one, and many to one. One of the fixes I had when the cmd window was telling me it was not working, I was able to solve it just by following the instructions it was giving me.

TypeError: ManyToOneRel.__init__() missing 2 required positional arguments: 'to' and 'field_name'
C:\Users\juhan\OneDrive\Desktop\school\CS3300\portfolio\portfolio_app\models.py changed, reloading.

```
class Project(models.Model):
    title = models.CharField(max_length=200)
    description = models.TextField(blank=True)
    portfolio = models.ManyToOneRel('portfolio', on_delete=models.CASCADE, to='portfolio_app.portfolio', field_name='portfolio', related_name='portfolio', related_query_name='portfolio')
```

In the given example above, the error stated that it was missing positional arguments, in which I was able to figure out what was needed for those arguments.

Routes in Admin app:
In GE03, we experienced the admin app capabilities of CRUD operations of the models through a web based interface. http://127.0.0.1:8000/admin/
Of course, I learned quickly that this was only possible after creating a superuser by python manage.py createsuperuser and update the admin.py.

```
from django.contrib import admin
from .models import Student, Portfolio, Project

admin.site.register(Student)
admin.site.register(Portfolio)
admin.site.register(Project)
```

In the admin app, I was able to test all the different models by creating each model such as portfolio, projects, and student. I was also able to see all the data being stored via the db.sqlite3 which showed that the data was now being stored in the database.

Source:
Cmd window error fix:
https://stackoverflow.com/questions/28122963/typeerror-init-missing-2-required-positional-arguments-client-socket-a
Model relations:
https://canvas.uccs.edu/courses/152508/pages/l09-models-and-database?module_item_id=2429637