Matthew Cao

Professor Harding

CS 3300

October 19, 2023

GE05 Part 1 Documentation

* How do you set up the generic list and detailed views for a model to be displayed in the web app.

To set up generic list and detailed views, a number of files need to be edited and created. First, URL mappings are added to my existing urls.py file. Second, class-based views are added to my views.py file. Third, I create new HTML templates for displaying additional student details on my web app based on what links are visited. My original base template file also needs to be altered. To sum up, I basically went through the following source, which provided step-by-step guidelines: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Generic\_views.

* What are the CRUD actions and routes you have created so far?

In regard to CRUD actions, I have done C, R, and D so far. For create and delete, I have created and removed several student, portfolios, and projects so far for testing purposes. To make sure my web app is functioning properly, I need to create and delete student profiles and check if they are displaying on my web app. When it comes to reading, my web app is currently set up to read information from my database in order to display it. For example, when I add a student, it is stored in my database and later accessed by both my Django admin site and web app. My database is the foundation for my web app, so I am constantly reading from it to obtain necessary information. The updating element of CRUD hasn’t been implemented yet because there isn’t a way for me to update information from the web app itself.

Source: https://stackify.com/what-are-crud-operations/

* How do you pass information from the database to display the active portfolios on the home page?

In my views.py file, I added the following lines of code to pass active portfolio

Information to the home page of my web app:

student\_active\_portfolios = Student.objects.select\_related('portfolio').all().filter(portfolio\_\_is\_active=True)

print("active portfolio query set", student\_active\_portfolios)

return render( request, 'portfolio\_app/index.html', {'student\_active\_portfolios':student\_active\_portfolios})

The code filters out portfolios set to active and prints them with my index.html template.

* + Explore the python shell command line to see what is returned for

student\_active\_portfolios = Student.objects.select\_related('portfolio').all().filter(portfolio\_\_is\_active=True)

The code returns only the portfolios that are set to active, that is the portfolios with the “active” checkbox selected when the portfolios were created.

* Explore the code in index .html [Built-in template tags and filters | Django documentation](https://docs.djangoproject.com/en/4.2/ref/templates/builtins/)

My index.html file is used for the home page of my web app. There are header tags in the file that correspond to the headers being displayed.

A close up of text

Description automatically generated

A blue and white sign

Description automatically generated

It is also important to note that my index.html file is an extension of my base\_template.html file. My base template is used as a template for my entire web app, while the index file specifically focuses on the home page.



My other templates such as student\_list.html are also extensions of my base template and contain the same extending line of code.