

Web Development Using Application Frameworks

Coding Assignment: Controller

Instructions

Overview

The Controller coding assignment is the next in a series of assignments in which we will be developing the EZU database system, a full C-R-U-D database application for a simplified university record keeping. In the first part of this assignment, we refactor the controller code (urls.py and views.py) for the list pages. In the second part of the tutorial, we create template files for the detail pages and enable them by creating entries in urls.py and views.py.

Tools

I am expecting you to use the tools that are demonstrated in the tutorial videos: Anaconda and PyCharm.

Tool Versions

Use the versions of PyCharm Professional, Anaconda, and Python that we installed during Week 1 of the course when we created the *e4_trainor_django_course* virtual env. These versions are documented in *Instructions for Tool Versions, Installation, and Virtual Environments*.

Starter Files

Please note that I have provided starter files for use in Part 2 of the tutorial. You may download them from the Weekly Schedule:

- starter_files_for_controller_assignment.zip

Tutorial Parts

Exercise 1 of this tutorial has two parts.

In Part 1, we refactor the urls.py and views.py entries for the list pages. In the video, I demonstrate refactoring and testing for the following list pages.

- Instructor
- Section

At the end of the Part 1 tutorial video, you are instructed to perform the same refactoring and testing for the remaining list pages on your own:

- Course
- Semester
- Student
- Registration

In Part 2, we create template files for the detail pages and enable them by creating appropriate entries in `urls.py` and `views.py`. During the video, I demonstrate coding and testing for the following detail pages.

- Instructor
- Section
- Semester

At the end of the Part 2 tutorial video, you are instructed to perform similar coding and testing for the remaining detail pages on your own:

- Course
- Student
- Registration

Exercises

1. Exercise 1 (Regular)

Follow both Part 1 and Part 2 of the tutorial instructions exactly. Be sure to do manual testing on your code.

2. Exercise 2 (Challenge)

The challenge exercises for all of the EZU-based coding assignments will be based on finding ways to maximize the usefulness of the Django Admin app in the activities of the current assignment. When working on these challenge exercises, you may consult any resources that you may find helpful. I am offering the following resources as a starting point:

1. [Django Software Foundation \(2025\)](#). *The Django admin site* in Django Documentation.
2. [Real Python \(2025\)](#). *Customize the Django Admin With Python*.
3. [Tomazic, N. \(2024\)](#). *Customizing the Django Admin* in TestDriven.io.

This week's challenge exercise is to do the following:

1. Write a short report that identifies ways in which we might customize the Django Admin app to help with the current week's work of writing and testing the code for this assignment including:
 - a. `urls.py`
 - b. `views.py`
 - c. Further template files
2. Identify 1 or 2 customizations. For each customization briefly explain how that customization might make our work easier or might make us more effective. Also, identify any coding work that would be required to implement the customization.

3. Place your report in a text file named *template_assignment_challenge.txt*. Place that text file directly into your Django Project directory.
4. Format your work as a business report with proper paragraphs and sentences. Pay attention to grammar, choice of words, and spelling. You do not need to cite sources.

Code and Document Deliverables

You are expected to submit one properly organized PyCharm Django project that is ready to be tested using PyCharm. Please refer to my tutorial video for details. Even if you have decided to do Exercise 2, just submit one Django project.

Non-Code Deliverables

Please be sure that the project you submit includes the following:

1. A test user (username = "tester", password = "{iSchoolUI}")
2. Sufficient test data present in the database to allow for testing all functions

Submission Method

Follow the process that I demonstrated in the tutorial video on submitting your work. This involves:

- Locating the properly named directory associated with your project in the file system.
- Compressing that directory into a single .ZIP file using a utility program.
- Submitting the properly named zip file to the submission activity for this assignment.

File and Directory Naming

Please use the following naming scheme for naming your PyCharm project:

surname_givenname_ezu

If this were my own project, I would name my PyCharm project as follows:

trainor_kevin_ezu

Use a zip utility to create one zip file that contain the PyCharm project directory. The zip file should be named according to the following scheme:

surname_givenname_ezu.zip

If this were my own project, I would name the zip file as follows:

trainor_kevin_ezu.zip

PLEASE NOTE: All file and directory names must be in lower case. Deductions will be made for submissions that do not conform to this standard.

Due Date

Please see the Weekly Schedule for the date and time when this assignment is due.

Last Revised
2025-02-17