Web Development Using Application Frameworks
Coding Assignment: Template
Instructions

Overview

The Template 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 Template coding assignment, we create list pages for most or the EZU database tables. This includes creating the template page, creating a function-based view, and creating an entry in urls.py.

Tools

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

Tool Versions

In the current semester, I am expecting you to use Python 3.10 and Django 4.1.

Starter Files

Please note that I have provided a starter file for `base.html`. You do not need to type this HTML code by hand. You can just copy this file into the proper directory structure within your PyCharm project.

Tutorial Parts

This is a one-part tutorial. During the video, I demonstrate the required coding and testing activities for two of the six EZU database tables:

- Instructor
- Section

At the end of the tutorial video, you are instructed to perform the coding and testing for the remaining tables on your own:

- Course
- Semester
- Student
- Registration

Note that we will not be creating pages in our EZU application for the year and period tables. These are infrequently changed infrastructure tables. We will assume that a system administrator will use the Django Admin application to maintain those tables when necessary.

Exercises

1. **Exercise 1 (Regular)**
   Follow the tutorial instructions exactly.

2. **Exercise 2 (Challenge)**
   My tutorial video for this assignment does not include automated unit testing. For the Challenge exercise, code automated unit tests using the same approach that we used for the Vincent Chapter 5 coding assignment. Automated unit tests for the model classes made up the challenge exercise in our previous coding assignment. In this week's challenge, include automated unit tests for code that we created in urls.py, in views.py, and in our templates.

Code 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}". PLEASE NOTE: We have changed the password that in the current semester. The old password is mentioned in some of the tutorial videos. Please be sure to use the new password instead.
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
2023_02_22