Web Development Using Application Frameworks
Coding Assignment: Link Pages
Instructions

Overview
The Link Pages 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 Link Pages coding assignment, we populate the `href` attributes in all of the pages that we created during earlier coding assignments. The result is a highly interlinked Web site that allows the user to pick any of the model classes as their entry point to this information system.

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*.

Tutorial Parts
This is a three-part tutorial.

In Part 1, we begin by providing the code required to redirect the root URL to the appropriate page. Following that, we populate the `href` attributes for the links that are part of the navigation section in the base.html template. During the video, I demonstrate coding and testing for the following navigation links.

- Instructors
- Sections

At the end of the Part 1 tutorial video, you are instructed to perform similar coding and testing for the remaining navigation links on your own:

- Courses
- Semesters
- Students
- Registrations

In Part 2, we populate the `href` attributes for all of the links on the list pages. During the video, I demonstrate coding and testing for the following list pages.

- Instructor
- Section

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

- Course
- Semester
- Student
- Registration

In Part 3, we populate the `href` attributes for all of the links on the detail pages. During the video, I demonstrate coding and testing for the following detail pages.

- Instructor
- Section

At the end of the Part 3 tutorial video, you are instructed to perform similar coding and testing for the remaining detail pages on your own:

- Course
- Semester
- Student
- Registration

Exercises

1. **Exercise 1 (Regular)**
   Follow Parts 1, 2, and 3 of the tutorial instructions exactly.

2. **Exercise 2 (Challenge)**
   My EZU tutorial videos do not include unit testing code. In the current semester, we are working in groups during class breakout sessions to create unit testing code for the version of EZU that we created in the previous assignment. To get credit for this Challenge Exercise, you will need to copy the unit testing code developed by your group in the classroom and incorporate it into your own copy of EZU. For this assignment, that means you will incorporate the unit tests for the code that your group developed for the following assignment:

   - Controller Assignment

   Remember that unit testing code in your copy of EZU should also still include the unit tests created by your group for earlier assignments.

   When incorporated into your copy of EZU, the unit tests should run and show all tests passing. To achieve this, you may need to refactor your code so that names used in the unit test code agree with names used in your EZU project. Also, you will need to correct any errors that are exposed by failing tests.

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
2024-02-20