Web Development Using Application Frameworks
Coding Assignment: Pagination and Staticfiles
Instructions

Overview

The Pagination and Staticfiles coding assignment is the next in a series of assignments in which we will be developing the EZ University database system, a full C-R-U-D database application for a simplified university record keeping. In the Pagination portion, we add pagination to the Instructor List and Student List pages. In the Staticfiles portion, we add layout and styling to the pages of the EZU application.

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 3.9.7 and Django 3.2.5.

Tutorial Parts

This is a three-part tutorial: Part 0, Part 1, and Part 2.

Part 0 – Run Migrations That Load Data for Pagination Testing

This part of the assignment is completely instructor-led. Play the tutorial video and follow along on your computer while you load test data for instructors and students. There is a starter zip file for this portion of the assignment that you need to download from the weekly schedule:

- `ezu_test_data_migration_starter_files.zip`

Part 1 – Add Pagination to Instructor List and Student List Pages
We begin this part of the tutorial by working together to code and test pagination capability for the following page:

- Instructor List

While accomplishing this goal, we will use the starter files provided in:

- `ez_university_pagination_tutorial_starter_files.zip`

After completing work on the Instructor List, you will work by yourself to code and test pagination capability for the following page:

- Student List

Part 2 – Staticfiles
In this part of the tutorial, we work together to do the following:

- Configure the Staticfiles feature in our Django project.
- Place .css and .png files into a *static* directory in our project.
- Update most of the template files in our project with new versions that contain styling hooks.

While accomplishing these goals, we will be using starter files provided in:

- `ez_university_staticfiles_tutorial_starter_files.zip`

The result of our efforts is a styled Web application that is substantially more visually appealing than the unstyled version that we have worked with up to this point.

Exercises

1. **Exercise 1 (Required)**
   Follow Parts 0 and 1 tutorial instructions exactly. These are the parts in which we load test data and develop the pagination features.

2. **Exercise 2 (Required)**
   Follow Part 2 of the tutorial instructions exactly. This is the part in which we add Staticfiles to the project.

3. **Exercise 3 (Optional Challenge Exercise)**
   Create a plan for using and testing Django pagination features in your Final Project. Your plan should include the following:

   - Which pages in your Final Project will include pagination features? Why these? Why not others?
   - How many test instances will you need to create in order to test the pagination features?
   - Where will you get or how will you generate the test data values?
   - How could you adapt the data migration code used in the first part of this assignment to load test data values of your own?

   Please write no more than 1 page of text (single spaced). Convert your document to a PDF file named *final_project_pagination_plan.pdf*. Place it in the *courseinfo* directory of your PyCharm project with program files like *urls.py* and *views.py.*

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.