Web Development Using Application Frameworks
Coding Assignment: Deployment
Instructions

Overview

The Deployment coding assignment is the last in a series of assignments in which we have developed the EZU database system, a full C-R-U-D database application for simplified university record keeping. In this assignment, we make changes to our Django project to support configuration for both development and production environments. Following that, we deploy the EZU application to a production server at PythonAnywhere.

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.

Important Note Regarding Passwords

PLEASE NOTE: The tutorial videos for this assignment use the old password scheme for logging into our *courseinfo* Django application. Regardless of what might be shown in the tutorial video, please give all users the new password that we are using in the current semester: {iSchoolUI}

Tutorial Parts

This is a 4-part tutorial.

Part 0 – Prepare to Adapt to Current-Semester Changes, Replace 3 Template Files
I recorded this Part 0 tutorial video for the Spring 2022 semester. The remaining tutorial videos for this assignment were recorded in Spring 2021. So, a few of their details are out of date. This Part, and its tutorial video will prepare you to make appropriate adjustments:

1. **Replace 3 Template Files**
   In the starter files for previous assignments, I appear to have included coding errors in 3 of the template files. You may have found these errors and already corrected them in your project. Just to be safe, I have included corrected versions of these 3 template files in the starter files for Part 0. We will begin by retrieving these corrected template files and replacing them in our EZU project. The starter files are contained in:

   - starter_files_for_deployment_assignment_part_0.zip

   The template files to be replaced include:

   - instructor_detail.html
   - registration_detail.html
   - student_detail.html

2. **Use Proper Anaconda Virtual Environment**

   Regardless of the instructions given during the videos for the following tutorial parts, remember to use the Anaconda virtual environment that we built for the current semester on your development machine:

   - e4_trainor_django_course

3. **Use New Version of Pip Freeze Command**

   When preparing a requirements file for our PythonAnywhere installation, we use the Pip Freeze command. The syntax of this command has changed since tutorial videos were recorded last year. The new syntax is:

   - pip list --format=freeze

4. **Use Python 3.10 on PythonAnywhere**

   The tutorial video for setting up your environment at PythonAnywhere includes instructions on selecting the correct version of Python. We are using Python 3.10. If you don't see that Python 3.10 is available, then you need to follow the instructions for changing the system image for your PythonAnywhere account. The system image that includes Python 3.10 is named *haggis*.

5. **Use the New Password Scheme**

   The tutorial videos for this assignment use the old password scheme for logging into our *courseinfo* Django application. Regardless of what might be shown in the tutorial video, please give all users the new password that we are using in the current semester: {iSchoolUI}

6. S**et Up the Correct Teacher Account on PythonAnywhere**

   When setting up your account on PythonAnywhere, make sure to set your Teacher Account to *trainortas*. Don't use the old teacher account name that is mentioned in the older tutorial video.

Part 1 – Create Multiple Settings Files

In this part of the tutorial, we work together to refactor the settings file for our EZU project. The result is a hierarchy of settings files that address both development and production environments. To accomplish this, we do the following activities:

1. Refactor the single settings.py file that is located in our configuration directory. We replace this single file with a directory named settings that contains 3 settings files: *base*, *development*, and *production*.

2. Test using the *development* settings configuration and repair problems created by the side effects of having moved the settings files. Please note that the error message shown by the development server may be different than those shown in the tutorial video. Despite this, the remedies that I demonstrate in the tutorial will work to clear error messages and get the development server working properly.

3. Test using the *production* settings configuration and repair problems created by the side effects of having moved the settings files.

Part 2 – Finish Preparing Project for Deployment

In this part of the tutorial, we work together to further prepare our Django project for production deployment. Activities include:

1. Create requirements file that can be used to control virtual environment on production server.

2. Check that our project is under version control using Git and that we have proper settings in .gitignore .

3. Remove test data migrations for students and instructors from the migrations chain.

4. Test migrations as they will be run in the production environment.

5. Test setting up users as they will be set up in the production environment.

Part 3 – Deploy Project to Production Using PythonAnywhere, Test

In this part of the tutorial, we work together to deploy our Django project to a production server at PythonAnywhere. Activities include:

1. Check your PythonAnwhere account to confirm that it supports Python 3.10. If not, change your system image. The system image that supports Python 3.10 is *haggis*. After you change and save the system image setting, make sure the the 3 *Default Python Versions* (python should run, python3 should run, the editor Run button should use) settings are also set to Python 3.10.

2. Make sure the *trainortas* is setup as the teacher for your PythonAnywhere account. Please note that the tutorial video mentions an older account setting for teacher. Do not use that one. Use *trainortas*.

3. Create a *bash* console.

4. Clone your project repository on the server. Please note that the tutorial video shows how to work with a Git repository hosted on BitBucket. More recently, I have been recommending that students host their repository on GitHub. The features of the server websites for these two products are similar but not the same. Please take hints from the video and try to avoid getting disoriented by details from BitBucket shown on the video.

5. If your database file was under version control, then delete the database file.

6. Create a virtual environment named *e4_ezu* using virtualenv. Note that this does not match the name that we are using on our development machine. This difference is expected.

7. Populate the virtual environment with Python packages using *pip* and the *base.txt* requirements file in the configuration directory of your project. Remember to use the new syntax for Pip Freeze:

   - pip list --format=freeze

8. Configure your production server using the PythonAnywhere *Web* link:

   a. Under the *Code* heading:
      i. Set *Source code* to your main project directory.
      ii. Do not change setting for *Working directory*.
      iii. Click on *WSGI configuration file* link and configure the file.
      iv. Do not change setting for *Python version* (should already be set).

   b. Under Virtualenv heading:
      i. Click on link and set to location of *e4_ezu* in your *.virtualenvs* directory.

9. Using the --*settings* option to point to the production settings file, run *manage.py* to accomplish the following:
   a. Run migrations on the new database. Remember to run the Group-Permissions migration a second time as a workaround to a bug.

b. Create the *tester* superuser. Even though the tutorial video mentions our old password scheme, please remember to use the new password scheme: {iSchoolUI}

c. Collect static files using the *collectstatic* command.

10. Using the PythonAnywhere *Web* link, configure staticfiles setting on the Web page. Remember to set values for both the *URL* and the *Directory*.

11. Using the PythonAnywhere *Web* link, start the production server using *Reload* button. Remember that you need to use this button to restart the server whenever you make changes to your code or configuration.

12. Login to the Django *Admin* app and create users with proper group membership and passwords (refer to documentation from the *Authentication and Authorization* assignment).

13. Use the *courseinfo* app to populate the application with a minimum set of test data (create 4 instances of everything).

14. Test the application as deployed and configured on the production server.

Exercises

1. **Exercise 1 (Required)**
   Follow Parts 0 through 3 of the tutorial instructions exactly.

2. **Exercise 2 (Optional Challenge Exercise)**
   For our convenience, we are deploying our system to the PythonAnywhere server with SQLite3 as the database. SQLite3 is typically only used for development and other relational database products are typically used on the productions server.

   - Which production-quality database server products are available on the PythonAnywhere platform?
   - Which of these database server products would you recommend for EZU? Why?
   - Would more product choices be available if we had a paying rather than a beginner account? Which would you choose if we had a paying account?
   - What further actions would we need to take in our deployment to use the recommended database server product instead of SQLite3? Be specific regarding the steps.

   Please write no more than 1 page of text (single spaced). Convert your document to a PDF file named *plan_for_a_production_database_server.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 the Challenge Exercise, just submit one Django project.

Non-Code Deliverables

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

1. Make sure that all usernames mentioned in the tutorial are created with the expected password (password = “{iSchoolUI}”). PLEASE NOTE: We have changed the password 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

Please note that this assignment has a different submission procedure than previous assignments. The procedure has 2 parts:

1. Deploy your EZU code to PythonAnywhere.
2. Submit the URL of the EZU application as installed on your PythonAnywhere account to the Canvas submission activity for this assignment. For example, the URL for my deployment of EZU at PythonAnywhere is *http://trainor1.pythonanywhere.com*

Deploy to PythonAnywhere

Before starting the tutorial, create an account on PythonAnywhere that can be used for deployment. Instructions for creating an account can be found on the Weekly Schedule. If possible, use the same username at PythonAnywhere as you are using at Illinois. Remember to identify our TAs as your teacher so that they will have access to your code. Their PythonAnywhere username is *trainortas*.

Deployment is accomplished by following all parts of the tutorial (see above).

Due Date

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

Last Revised

2023-04-20