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 3.8 and Django 3.1.

Tutorial Parts

This is a 3-part tutorial.

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.

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 3.8. If not, change your system image.

2. Make sure the *trainor1* is setup as the teacher for your PythonAnywhere account.

3. Create a *bash* console.

4. Clone your project repository on the server.

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

6. Create a virtual environment named *e4_ezu* using virtualenv.

7. Populate the virtual environment with Python packages using *pip* and the *base.txt* requirements file in the configuration directory of your project.

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.

   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.

Code Deliverables

You are expected to deploy a properly organized EZU Django project that is ready to be tested on the PythonAnywhere server. Please refer to my tutorial video for details.

Non-Code Deliverables

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

1. A test user (username = “tester”, password = “(secret)”
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 a one-page document to Moodle to confirm that your project is ready for grading.

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 me as your teacher so that I will have access to your code. My PythonAnywhere username is *trainor1*.

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

Submit One-Page Document to Moodle

In order to facilitate the grading of your work, please submit a one-page PDF document to the Moodle submission activity for this assignment. The document should contain the following information:

1. Your name
2. Your PythonAnywhere username
3. A statement that your code is deployed and ready for grading

File Naming

The file that you submit to Moodle should be named using the following scheme:

surname_givenname_deployment_assignment.pdf

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

trainor_kevin_deployment_assignment.pdf

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

2021-04-20