

Web Development Using Application Frameworks Instructions for the Final Project

General Description

You will be expected to plan, gather requirements for, design, code, and test a Web application using Django as your Final Project. Expectations include:

- The Web application should fully demonstrate the Django application framework features covered in the class. These include:
 - Django models
 - Django URL configurations
 - Django views (including class-based views and generic class-based views)
 - Django templates
 - Django user authentication and permission schemes
- The Web application should include significant database add/change/delete functionality.
- The Web application should be sufficiently interesting to you that you are likely to continue to develop and maintain it after the course is complete.

More details regarding project expectations are presented below.

Project Size

The size of the project should be comparable to the size of the completed EZU tutorial project. Size will be assessed based upon the number of *Django function points* in your Final Project. Django function points are a measurement of application size that I devised using a simplified version of a traditional approach to measuring application size called [function points](#). The formula for Django function points is:

$$\text{number of model classes} + \text{number of URL patterns}$$

Using the formula above, the completed EZU tutorial project has a Django function point score of 44.

Nature of the Application

Provided that you meet the other expectations for the project, you have fairly wide latitude regarding the nature of the application. Whatever application you choose, please make sure that it has a significant database component.

Design and Coding

The design and coding of your project should be consistent with the standards and practices presented in the Pinkham text and demonstrated in the EZU tutorial example.

How Important is Creativity

The grading of your final project will be based upon the grading rubric document which will be published separately in our Weekly Schedule. Please consult this document for details of the grading scheme.

Creativity is important on this project in that it may help to keep your interest in the project high. Designing a novel application, providing interesting features, providing remarkably high usability, including dramatic styling, or creating an impressive portfolio piece are all ways to express yourself and increase your own commitment to this work. So, if you wish to include these in your project, you might expect them to increase your own satisfaction.

Nevertheless, you should be aware that there are no points allocated in the grading rubric for such a level of creativity. Your application doesn't need to represent a promising business opportunity such that it might be the new Facebook or the new Twitter. It doesn't need to demonstrate important knowledge of other coursework from the iSchool like machine learning or natural language processing. It doesn't need to provide more usability than that provided in the EZU tutorial example. And it definitely does not need to have better styling than the EZU tutorial example.

When you examine the grading rubric, you will see that **points are awarded for building an application that is comparable to the EZU tutorial example in its size, scope, completeness, and professional build quality**. As stated above, please refer to the grading rubric document for details of the grading scheme.

Tools

I am expecting you to use the tools that we have been using throughout the semester: Anaconda, PyCharm, and Git.

Tool Versions

In the current semester, I am expecting you to use Python 3.10 and Django 4.1. If you are considering a final project configuration that would require a different version of Python or Django, please get my permission before making this choice.

Our plan for testing your submission will be to use the *e4_trainor_django_course* virtual env that we have been using throughout the course. **If your final project submission requires a different virtual environment, then you must include a *requirements.txt* file for your project that we can use to build the virtual environment for our testing of your work.** This *requirements.txt* file might include a different version of Python (if approved), a different version of Django (if approved), and any additional Python packages for features that you are using that are not covered during the course.

Testing

Your application should be fully tested when submitted. This means that you should have devised a manual testing plan for the application and followed it. Reading the items that you submit as documentation should allow us to follow that same manual testing plan when we evaluate your submission. This manual testing will meet the requirements for *Exercise 1* (see below).

If you are also submitting *Exercise 2*, you will be expected to create a full set of automated tests for your project. See *Exercise 2* (below) for more details.

Documentation

In order to properly evaluate your submission, we will need for you to provide us with documentation. This single document should be placed within your Django project in the same directory as your *settings.py* file and named:

README.PDF

This README.PDF document should include enough information for us to properly evaluate, test, and grade your submission. It should include at least the following:

1. Special Virtual Environment (if applicable)

If I have approved a different virtual environment for your project (see above), then you must provide a *requirements.txt* file and instructions for building the virtual environment that we will use to test your work.

2. Application Description

A short description of your application and its intended user communities. This includes both regular users and administrative users.

3. Authentication and Authorization Scheme

A description of the authentication and permissions scheme that you have implemented and how it corresponds to the communities described above. A table of user groups and permissions assigned to user groups should be included.

4. User IDs and Passwords

Lists of user IDs and passwords with an explanation of how they map to the communities described above. At a minimum, your application should have a Django superuser defined named *tester* that has a password of *{iSchoolUI}*.

5. Testing Instructions

Instructions for manual testing of your application using the sufficient test data, user IDs, and passwords that you have provided.

6. Other Relevant Information

Any further information that will help us understand your application for the purposes of fairly and fully evaluating it.

Exercise 1 (Regular)

The project described above will constitute the work for Exercise 1, the Regular portion of this assignment.

Exercise 2 (Challenge)

The requirements for *Exercise 1* do not include automated unit testing. For this Challenge Exercise, code automated unit tests for your entire Django project using the approaches shown in Chapters 5, 6, 7, and 9, of the Vincent Book.

Format

Submit 1 file of type .ZIP.

Submission

Please note that the name of the file submitted and the name of your PyCharm project directory must be in all lower case. Deductions will be made for submissions that do not follow this standard.

Zip up your PyCharm project directory and submit that zip file to the submission activity identified for this assignment in the Weekly Schedule.

When submitting your project files, please use the following naming conventions:

- Name your PyCharm project directory using the following naming pattern:

`surname_givename_final_project`

If this were my project, I would name it:

`trainor_kevin_final_project`

- Consequently, the zip file that you submit will have the following naming pattern:

`surname_givename_final_project.zip`

If this were my project file, I would name it:

`trainor_kevin_final_project.zip`

Submission Deadline

The submission deadline and submission activity will be indicated in the Weekly Schedule.

Grading

A separate grading rubric document will be posted to the Weekly Schedule.

Last Revised

2023-02-27