

Exercise 2.8: Deploying a Django Project

Learning Goals

- Enhance user experience and look and feel of your web application using CSS and JS
- Deploy your Django web application on a web server
- Curate project deliverables for your portfolio

Reflection Questions

- Explain how you can use CSS and JavaScript in your Django web application.

In a Django web application, you can use CSS and JavaScript by placing your static files (CSS, JavaScript, images, etc.) in a static directory within your app. You reference them in your templates using the `{% load static %}` tag and then link to the files with `{% static 'path/to/file.css' %}` or `{% static 'path/to/file.js' %}`. This allows you to style your application with CSS and add interactivity with JavaScript. Ensure `STATIC_URL` and `STATICFILES_DIRS` are configured in your `settings.py`.

- In your own words, explain the steps you'd need to take to deploy your Django web application.

Here are the steps to deploy your Django web application to PythonAnywhere:

Sign Up / Log In: Create an account or log in to PythonAnywhere.

Create a New Web App:

Go to the "Web" tab.

Click "Add a new web app."

Select the "Django" option.

Choose a Version: Select the appropriate version of Python for your project.

Enter Your Project Directory: Specify the path to your Django project.

Configure WSGI File:

Edit the WSGI file to point to your Django application.

Ensure the path to your virtual environment is correct.

Set Up Virtual Environment:

Go to the "Consoles" tab.

Start a new Bash console.

Create and activate a virtual environment.

Install your project requirements using `pip install -r requirements.txt`.

Database Configuration:

Set up your database (if applicable) in the PythonAnywhere console.

Run migrations using `python manage.py migrate`.

Static Files:

Collect static files using `python manage.py collectstatic`.

Configure the static files section in the "Web" tab to serve them correctly.

Check Settings:

Ensure `ALLOWED_HOSTS` in your `settings.py` includes your PythonAnywhere domain.

Configure any necessary environment variables.

Reload Web App: Go back to the "Web" tab and click the "Reload" button for your web app.

Access Your Site: Visit your PythonAnywhere domain to see your deployed application.

- (Optional) Connect with a few Django web developers through LinkedIn or any other network. Ask them for their tips on creating a portfolio to

showcase Python programming and Django skills. Think about which tips could help you improve your portfolio.

- You've now finished Achievement 2 and, with it, the whole course! Take a moment to reflect on your learning:

- What went well during this Achievement?

The basics of Python went really well for me. I found the concepts of data types and control structures, like loops and conditionals, to be intuitive. Understanding functions was particularly satisfying, as it allowed me to write cleaner and more organized code.

- What's something you're proud of?

I am particularly proud of successfully deploying my Django web application. The deployment process was challenging for me due to the different steps involved, such as setting up the server, configuring the database, and ensuring all dependencies were correctly installed. Overcoming these hurdles made me feel accomplished and gave me confidence in my skills.

- What was the most challenging aspect of this Achievement?

The most challenging aspect was the deployment process. I struggled with configuring the server settings and understanding how to link my database correctly. There were moments of frustration, especially when I encountered errors that I didn't know how to resolve. However, researching and troubleshooting these issues ultimately taught me a lot.

- Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Django skills?

Yes, this Achievement definitely met my expectations. It exceeded them, actually, by showing me the practical applications of Python and Django in web development. I feel much more confident now in using Django to build web applications, and I'm excited to explore more advanced topics, such as API development and user authentication.