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

1. **Explain how you can use CSS and JavaScript in your Django web application.**
   CSS and JavaScript (JS) are used to improve the look, feel and overall user experience of an application. CSS provides style to web pages and JS provides interactive elements that engage a user (to web pages).
   I can use CSS and JS to enhance my user interface in many different ways. Some of these include:
   - Creating a navigation bar;
   - Adjusting the style of text, forms, tables, and buttons;
   - Implementing dynamic elements (showing or hiding information with the click of a button, changing color on mouse hover, playing audio or video on a page, displaying animations, zooming in or out an image, displaying countdowns or timers, etc.).

2. **In your own words, explain the steps you'd need to take to deploy your Django web application.**
   To deploy my Django web application I need to go through several steps:
   1. Prepare GitHub repository and upload code.
      a. Navigate to the new repository in your system.
      b. Push your content to the remote GitHub repository.
      c. Upload content.
      d. Status check.
      e. Commit.
      f. Send changes online.
      g. Check GitHub.
   2. Update Django application for hosting.
      a. Procfile - create a new file called Procfile (capital P and no extension) in the root folder.
      b. Gunicorn – install Gunicorn.
      c. Database configuration.
         i. Install dj-database-url.
         ii. Configure settings.py.
         iii. psycopg2 (Python Postgres database support)
      d. Static files.
      e. WhiteNoise.

        i. Install WhiteNoise.
        ii. Update settings.py.
    f. DEBUG.
    g. SECRET_KEY.
    h. Requirements.
3. Deploy using Heroku.
    a. Create an account.
    b. Install the client.
    c. Log in.
    d. Create and upload the website.
        i. Create application.
        ii. Upload.
        iii. Set up database table(s).
        iv. Create superuser.
        v. Visit site in browser.
        vi. Add data.
    e. Set SECRET_KEY.

3. **(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.**

-

4. **You've now finished Achievement 2 and, with it, the whole course! Take a moment to reflect on your learning:**
    a. **What went well during this Achievement?**
    b. **What's something you're proud of?**
    c. **What was the most challenging aspect of this Achievement?**
    d. **Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Django skills?**

The most challenging aspect of this Achievement was the adaptation to the "Django way" and the exercise about "Data Analysis and Visualization in Django", namely the section about "Visualization and Charting". I run into several issues while implementing the charts and had to explore other resources to complete the task.
I was expecting to find more information about tests.
I also had several doubts regarding the key features and requirements of the project brief. The exercises/tasks don't seem to be completely aligned with the project brief.
I'm proud of reaching the end of the course (and the program). It gave me some confidence, but I need to continue working in my Django (technical) skills.