Core 5 Topics:

1. GitHub Repository, Environment Setup & Project Structure

I set up my github repository in the beginning and used it throughout the project to save and log my progress.

2. Wizard of Oz Prototyping / UI/UX Planning

Before I started my project, I drew wireframes so I would have an idea of what the UI would look like in its finished state.

3. Models + ORM Basics

I created four main models for my website: Book, BookList, BookListItem, and ReadingProgress. Each of these make up the core parts of my website and support any future improvements I would make (BookList supports multiple lists being made in the future).

4. Views + Templates + URLs (Function or Class-based)

I created views, templates and urls for all my pages. I wanted a limited number of pages so I had a lot going on in all my views to make sure the functions I wanted were on the same page as each other.

5. User Authentication for Internal Users

I made sure to load my models to the admin page and created a superuser for myself and the instructor using the username and password listed in the README file. Only authenticated internal users can access it.

6. Deployment (Production Setup)

I successfully deployed my live website on PythonAnywhere.

Functional Add-ons:

1. ORM Queries + Data Summaries

I utilized ORM queries in my Dashboard page and MyBooks page to allow users to send queries to the OpenLibrary API and their own book list. I also included data summaries in the stats page.

2. Static Files (CSS/JS Integration)

I used bootstrap and CSS to customize my website and make it user friendly, including colors, containers for their tbr list and current read in the dashboard page and separate boxes for the books in the MyBooks page.

3. Charts / Visualization (Matplotlib or Plotly)

I used matplotlib to create a bar chart view and a pie chart view. Both these views render as pngs and are displayed in the stats page, showing how many books the user has read each month this year and displaying the number of books in each list (reading, to-read, finished).

4. Forms + Basic Input / CRUD

I use forms for the login/sign up pages and have numerous post/delete functions within my website. A user can delete a book from their list, update each book's status and the number of pages they have read so far in their current read.

5. Integrate External APIs

I integrated multiple external APIs for this project with OpenLibrary, including their api to search books, access each edition of a book, and access the cover of each edition of the book. This was probably the hardest thing to implement in this project and one I would still like to play with after this class to improve the UI of the search results.

6. Data Presentation & Export

I made it possible for users to download their "mybooks" data in both json and csv formats using the export views we learned a few weeks ago.

7. User Authentication for External Users

Users have to either create an account or signup for the website to access any of the pages I made, and once they create one, they have a blank account to add their own data into.