Software Design

Specifications

for

Group 5 Online Book Exchange

Version 3.0 approved

Prepared by Alexander Voisan, Angel Villalobos, Fernando Torres, Guang Wu, Mychal Salgado, Portia Wu

Cal State LA University

May 11th, 2022

Table of Contents 2

[Revision History](#_1ttp2sodvb5d) 4

[1.Introduction](#_19o1ohgrpzm9) 5

[1.1 Purpose](#_3mm8hxvf1hx4) 5

[1.2 Document Scope](#_vz1jt3m7h4ep) 5

[1.3 Intended Audience](#_69hgdrvojvv6) 5

[1.4 System Identities](#_lw59bncx6wfb) 5

[1.5 References](#_vqi3ykvwugd9) 6

[1.6 Define any important terms, acronyms, or abbreviations See Section S - Glossary](#_12s90tu0wpe) 6

[2. System Overview](#_el4ycotgypua) 7

[3. Design Considerations](#_v5o03lqb5ae5) 8

[3.1 Assumptions and Dependencies](#_k0ecczlna4xt) 8

[3.2 General Constraints](#_w7oa4sl78rhy) 8

[3.3 Goals and Guidelines](#_wibewio5wkd9) 8

[3.4 Development Methods](#_gez1kf7l7fwv) 8

[4. Architectural Strategies](#_eyx1neeqt266) 9

[4.1 Products used (programming language, database, library, etc.)](#_ei533vokg8rz) 9

[4.2 Reuse of existing software components implemented into system](#_40xfn0ngi7i4) 9

[4.3 Future plans for extending or enhancing the software](#_2k9phri2f5h9) 9

[4.4 User interface paradigms (or system input and output models)](#_yl4rjq7punnc) 9

[4.5 Hardware and/or software interface paradigms](#_fsr76fidq1e8) 10

[4.6 Error detection and recovery](#_rjvtejt6i02g) 10

[4.7 Memory management policies](#_tncbqhj19na7) 10

[4.8 External databases and/or data storage management and persistence](#_jk9ugp6qc0if) 10

[4.9 Distributed data or control over a network](#_6hi96k2tdgae) 10

[4.10 Generalized approaches to control](#_rrqthkok7sia) 10

[4.11 Concurrency and synchronization](#_h650h6373usn) 10

[4.12 Communication mechanisms](#_9ctocn1dzu5u) 10

[5. System Architecture](#_r1ca2edw7yor) 11

[6. Detailed System Design](#_bom1ijy9c48) 13

[6.1 Module-1 About Us](#_rmkojc7m3qyy) 13

[6.2 Module-2 Register Page](#_j3hrt2ruet6g) 13

[6.3 Module-3 Login Page](#_dijwzs9l014i) 14

[6.4 Module-4 Post Book](#_zcprdyavhhhu) 15

[6.5 Module-5 Display Book](#_9e0muh7dso2p) 16

[6.6 Module-6 My Books](#_d7jmce6brypc) 17

[6.7 Module-7 Search Book](#_3qpvvtv39cx7) 18

[6.8 Module-8 Book Details](#_f8svdbjv8g0c) 19

[6.9 Module-9 FAQ](#_e0gbgs3wl8eu) 20

[6.10 Module-10 Wish List](#_apz4blyzmw8w) 21

[7. Graphical User Interface](#_t2cwrjjn052j) 22

[8. Glossary](#_x8z9ryqwvjef) 26

# 

# 

# 

# 

# 

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name  First Draft | Date  4/27/22 | Reason For Changes  First Draft of SDS | Version  1 |
| Second Draft | 5/4/22 | Second Draft of SDS, improvements from prior feedback. | 2 |
| Final Draft | 5/11/22 | Last draft of SDS, improvements and corrections due to new features. | 3 |
|  |  |  |  |

# 

# 1.Introduction

## 1.1 Purpose

The purpose of this document:

1. To describe the inner workings of the project and explain the concepts in which we used to create it.

2. To provide the reader with information on the technological needs to use this project.

3. Explain what the intended use of the project is for.

4. Promote and explain current and future features the project will contain.

## 1.2 Document Scope

This document is intended for Developers, Project Managers, Customers, and Testers as well as document writers to help understand what the project contains. The specified sections each reader may want to focus on are:

Developers - Section(s) 1.3, and 2 through 5. Understand the scope of the project as well as the internal workings and what users will be seeing on their end. Project Managers - Section(s) 3 through 5. Understanding the functions of the project and learning how they may be interconnected.

Customers - Understanding if the product meets your needs and will function for you. Section(s) 1.3, and 3 through 5

Testers - Using internal documentation to confirm function descriptions and to confirm the workings of functions. Section(s) 3

## 1.3 Intended Audience

This document’s intended audience is Testers, Customers and Developers

## 1.4 System Identities

Group 5 Online Book Exchange Version 3.0

## 1.5 References

1.5.1 General information on Bootstrap

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

1.5.2 Information on Django queries

<https://docs.djangoproject.com/en/3.2/topics/db/queries/>

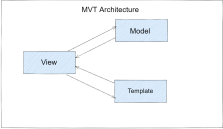
1.5.3 General information and documentation on Django

https://docs.djangoproject.com/en/3.2/

## 1.6 Define any important terms, acronyms, or abbreviations See Section S - Glossary

# 

# 2. System Overview



The MVT (Model View Template) is a software design pattern. It is a collection of three important components: Model View and Template. The Model helps to handle databases. It is a data access layer which handles the data.

The Template is a presentation layer which handles the User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and render a template.

# 

# 3. Design Considerations

## 3.1 Assumptions and Dependencies

PyCharm and Django will be needed for an easy and clean install and use of the software and Bootstrap on the local server if you wish to have your own version of it.

## 3.2 General Constraints

Our current Software constraints are as followed

Hardware limitations:

Hardware will have to scale given the size of the user base and amount of book information sellers upload onto the site. Storage will be the main factor in this, as user base size will rely on CPU and RAM constraints to give them a smooth experience.

Security:

Admin logins will have to implement 2FA for a more secure account.

## 3.3 Goals and Guidelines

The goal is to allow users to be able to communicate with each other as well as more directly with the publishers in order to give a better understanding of what books to sell.

## 3.4 Development Methods

● GitHub

● PyCharm

● MVC

# 4. Architectural Strategies

## 4.1 Products used (programming language, database, library, etc.)

1. Python

2. Django Framework

3. BootStrap

## 4.2 Reuse of existing software components implemented into system

● BootStrap

● Django

## 4.3 Future plans for extending or enhancing the software

Future plans include but are not limited to:

1. Wishlist System

2. Shopping Cart

3. Genre System

4. Report System - *Admins Only are allowed to see open reports* 5. Forum System

## 4.4 User interface paradigms (or system input and output models)

● Use of a Form System from Django to input / output data which interacts with SQLite3

## 4.5 Hardware and/or software interface paradigms

PyCharm

## 4.6 Error detection and recovery

We Use Django to automatically catch errors the users will tend to create when adding books or comments into the database

## 4.7 Memory management policies

Django

## 4.8 External databases and/or data storage management and persistence

SQLite3

## 4.9 Distributed data or control over a network

N/A

## 4.10 Generalized approaches to control

Users are given minimal control over any data they are inputting into the server and Django checks data to confirm it conforms to the proper rules set in place

## 4.11 Concurrency and synchronization

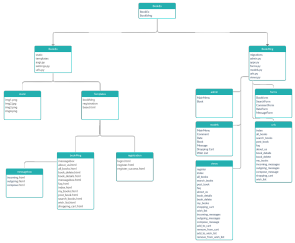
SQLite3 handles the data in the backend and insures that all users are seeing the correct data across the site

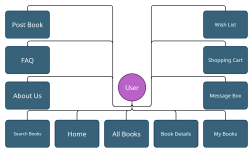
## 4.12 Communication mechanisms

● Django - Allows forms to be used which allows the users to use the front end to send data to the backend which would be stored in a SQLite3 database

● HTTPS

# 5. System Architecture





5.1.1 About us

This page displays the information of the students in the project

5.1.2 Register

This page allows users to register for an account

5.1.3 Login

This page allows registered users to login to access the rest of the website

5.1.4 Display Books

This page displays all the books in the database

5.1.5 Post Book

This page allows users to post a book on the website

f5.1.6 Search Book

This page allows users to search for any books given the condition of the user.

5.1.7 Home Page

The main page of the website does not need users to be registered to view the page. Users can view the page by clicking on the book icon on the nav bar

5.1.8 My Book

The page displays every book the user has posted

5.1.9 Book Detail

The user can see the details of any book by clicking on the name.

5.1.10 Wish List

The user can see a list of books they have added to their favorites list.

5.1.11 Shopping Cart

The user can see a list of books they have added to their shopping cart and can choose to buy them here.

5.1.12 Rate Book

The user can rate a book on a scale of 1(not good) to 5(excellent).

# 6. Detailed System Design

## 6.1 Module-1 About Us



6.1.1 Definition

The page is the about us that displays the information of the students working on the project

6.1.2 Responsibilities

The page displays the information with text and with links to external websites

6.1.3 Constraints

Currently no constraints.

6.1.4 Composition

The page consists of Bootstrap, HTML, Python and uses a Django server

6.1.5 Uses/Interactions

Navigation bar on the top of the page as well as links to the GitHub accounts of the group members

6.1.6 Resources

None

6.1.7 Processing

All the page does is display text and links

6.1.8 Interface

● Python

● HTML

● Bootstrap

● Django

## 6.2 Module-2 Register Page



6.2.1 Definition

The page is where users create an account

6.2.2 Responsibilities

The page retrieves the user’s information and stores in the database

6.2.3 Constraints

User must follow the requirements on the page when filling out the password section

6.2.4 Composition

The page consists of Bootstrap, HTML, Python and uses a Django server

6.2.5 Uses/Interactions

Navigation bar on the top of the page. Text fields where users input information for their username and password

6.2.6 Resources

Stores users in the SQL database

6.2.7 Processing

Gathers the data to send to database

6.2.8 Interface

● Python

● HTML

● Bootstrap

● Django

● SQL

## 6.3 Module-3 Login Page



6.3.1 Definition

The page is where users login with their username and passwords they created

6.3.2 Responsibilities

The page retrieves the user’s information and checks with the data on the database

6.3.3 Constraints

User must input their username and password to access the rest of the website or the user must go to the register page

6.3.4 Composition

The page consists of Bootstrap,HTML,Python and uses a Django server

6.3.5 Uses/Interactions

Navigation bar on the top of the page. Text fields where users inputs their username and password

6.3.6 Resources

Retrieves date on users in the SQL database

6.3.7 Processing

Checks the credentials with those on the database.

6.3.8 Interface

● Python

● HTML

● Bootstrap

● Django

● SQL

## 6.4 Module-4 Post Book



6.4.1 Definition

The page allows users to post books.

6.4.2 Responsibilities

The page gathers the name,price,website and picture to create a book in the database.

6.4.3 Constraints

Users must be signed in and fill out all fields.

6.4.4 Composition

The page consists of Bootstrap,HTML,Python and uses a Django server

6.4.5 Uses/Interactions

Navigation bar on the top of the page. Text fields where users inputs name,price,website and picture

6.4.6 Resources

Saves book in the SQL database

6.4.7 Processing

Checks if the user creates a successful book then sends the data to the database.

6.4.8 Interface

● Python

● HTML

● Bootstrap

● Django

● SQL

## 6.5 Module-5 Display Book



6.5.1 Definition

The page displays all the books currently stored in the SQL database

6.5.2 Responsibilities

The page has to go through every book in the SQL database and display them on the page

6.5.3 Constraints

Too many books might cause the webpage to take more time to load

6.5.4 Composition

The page consists of Bootstrap,HTML,Python and uses a Django server

6.5.5 Uses/Interactions

Navigation bar on the top of the page. Users can click on the links on the name to get more details on the book. Books will show their front covers on this page. Getting more details on the book allows users to comment as well as rate the book. Users can also change their ratings on books whenever they want as well. Users may not rate their own books.

6.5.6 Resources

Retrieves data from SQL database

6.5.7 Processing

Has to go through all of the books in the database

6.5.8 Interface

● Python

● HTML

● Bootstrap

● Django

● SQL

## 6.6 Module-6 My Books



6.6.1 Definition

The page displays all the books currently stored from the user in the SQL database

6.6.2 Responsibilities

The page has to go through every book from the user in the SQL database and display them on the page

6.6.3 Constraints

Too many books might cause the webpage to take more time to load

6.6.4 Composition

The page consists of Bootstrap,HTML,Python and uses a Django server

6.6.5 Uses/Interactions

Navigation bar on the top of the page. Users can click on the links on the name to get more details on the book. Users can click on the delete button to delete books the user has posted. Users will see comments and the rating of their book in book details.

6.6.6 Resources

Retrieves data from SQL database

6.6.7 Processing

Has to go through all of the books from the user in the database

6.6.8 Interface

● Python

● HTML

● Bootstrap

● Django

● SQL

## 6.7 Module-7 Search Book



6.7.1 Definition

The page allows users to search books on the website

6.7.2 Responsibilities

The page has to go through every book in the SQL database and display them on the page based on what the user inputed

6.7.3 Constraints

Too many books might cause the webpage to take more time to load

6.7.4 Composition

The page consists of Bootstrap,HTML,Python and uses a Django server

6.7.5 Uses/Interactions

Navigation bar on the top of the page. Text field where users input what they want to search. Users can click on the links on the name to get more details on the book.

6.7.6 Resources

Retrieves data from SQL database

6.7.7 Processing

Has to go through all of the books in the database matching what the user inputed

6.7.8 Interface

● Python

● HTML

● Bootstrap

● Django

● SQL

## 6.8 Module-8 Book Details



6.8.1 Definition

The page allows users to view a specific books’ details, depending on what book the user chooses.

6.8.2 Responsibilities

The page has to go through every book in the SQL database and display the book with the same id as the book that was chosen in the prior page. This page will also find all comments that have been published for the book and show them on the page. In addition, the current user will be able to add comments and add a rating for the selected book.

6.8.3 Constraints

Too many comments and books might cause the webpage to take more time to load

6.8.4 Composition

The page consists of Bootstrap, CSS, HTML, Python, and uses a Django server

6.8.5 Uses/Interactions

Navigation bar on the top of the page. Text field where the user can input a comment. Star ratings that users can select to rate the current book Users can click on the buttons on the bottom of the page to post a comment or to submit a rating.

6.8.6 Resources

Retrieves data from SQL database

6.8.7 Processing

Has to go through all of the books in the database matching what the user inputed

6.8.8 Interface

● Python

● HTML

● Bootstrap

● Django

● SQL

● CSS

## 6.9 Module-9 FAQ



6.9.1 Definition

The page allows users to view a specific question page, depending on what question the user chooses the site will display the selected question only.

6.9.2 Responsibilities

The page has the responsibility of helping teach users about the website, and how certain pages function for the user

6.9.3 Constraints

Users must be logged in and a longer FAQ page may be hard to navigate.

6.9.4 Composition

The page consists of Bootstrap, CSS, HTML, Python, and uses a Django server

6.9.5 Uses/Interactions

Navigation bar on the top of the page. Large dropdown list of questions the user may have in which the questions are answered.

6.9.6 Resources

None

6.9.7 Processing

No processing needed as it only displays text

6.9.8 Interface

● Python

● HTML

● Bootstrap

● Django

● SQL

● CSS

## 6.10 Module-10 Wish List



6.10.1 Definition

The page displays all the books currently stored for the users list in the SQL database

6.10.2 Responsibilities

The page has to go through books from the user in the SQL database and display them if the user has marked them for being onto their wishlist

6.10.3 Constraints

Too many books might cause the webpage to take more time to load

6.10.4 Composition

The page consists of Bootstrap,HTML,Python and uses a Django server

6.10.5 Uses/Interactions

Navigation bar on the top of the page. Users can click on the links on the name to get more details on the book. Users can click on the delete button to delete books from their wish list. Users will see the seller and price of each book.

6.10.6 Resources

Retrieves data from SQL database

6.10.7 Processing

Has to go through the books in the users list in the database

6.10.8 Interface

● Python

● HTML

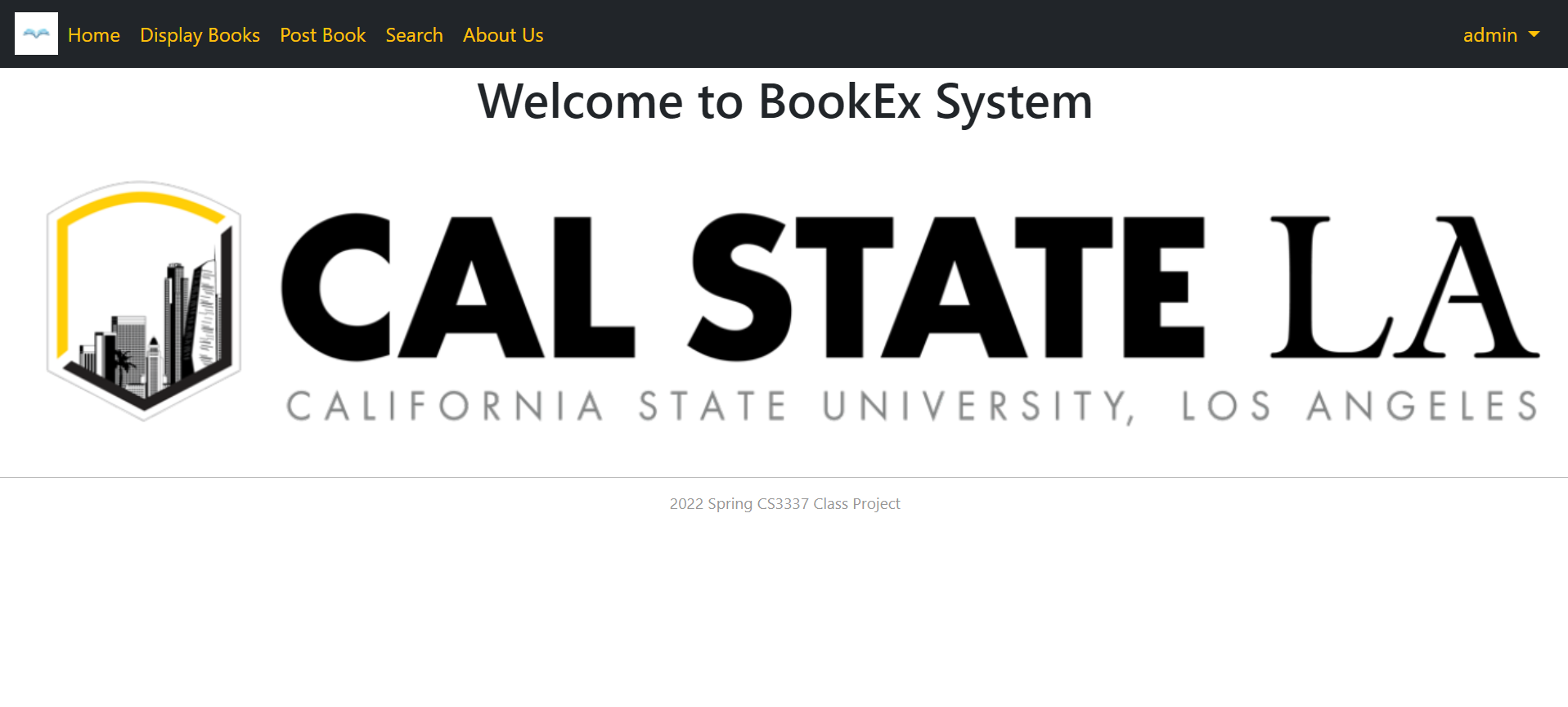
● Bootstrap

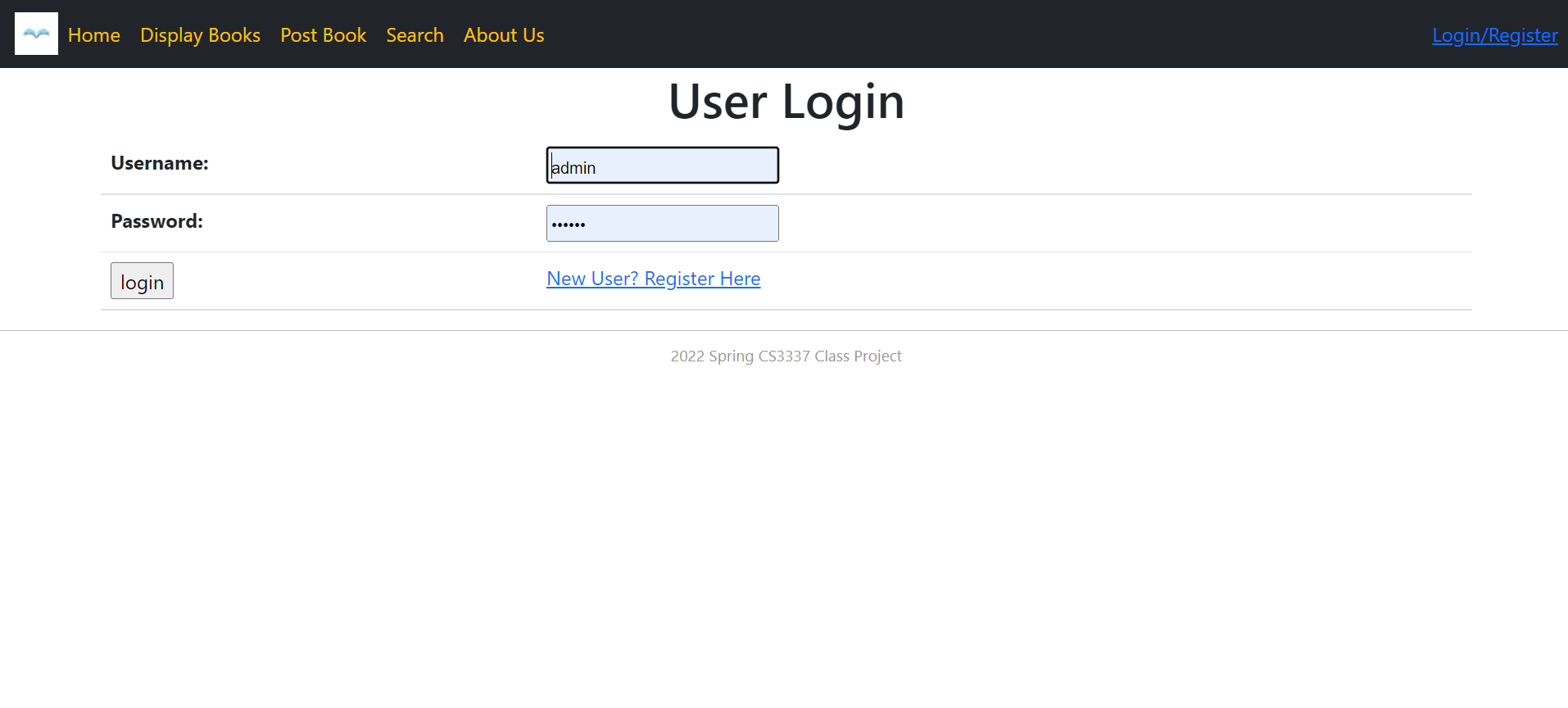
● Django

● SQL

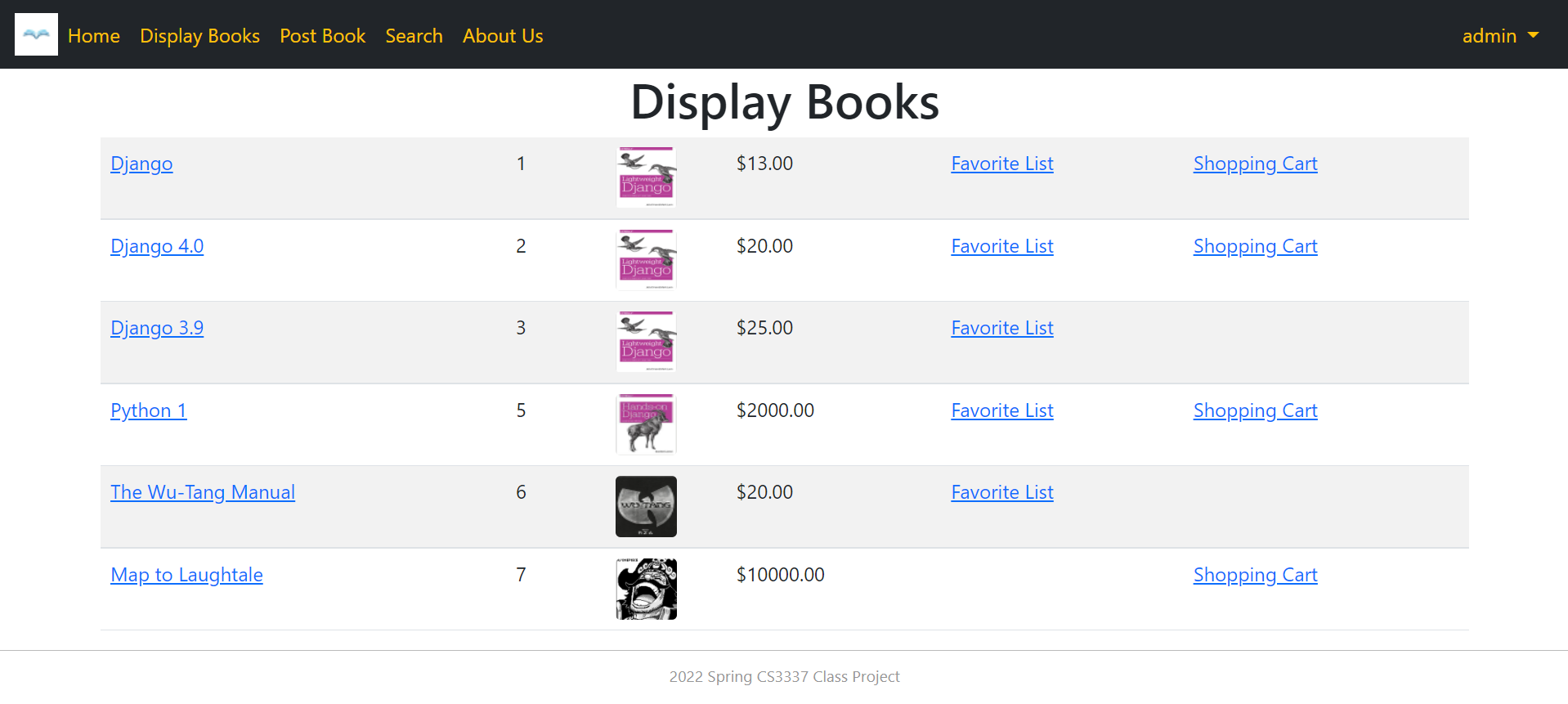
# 7. Graphical User Interface

Homepage of the website

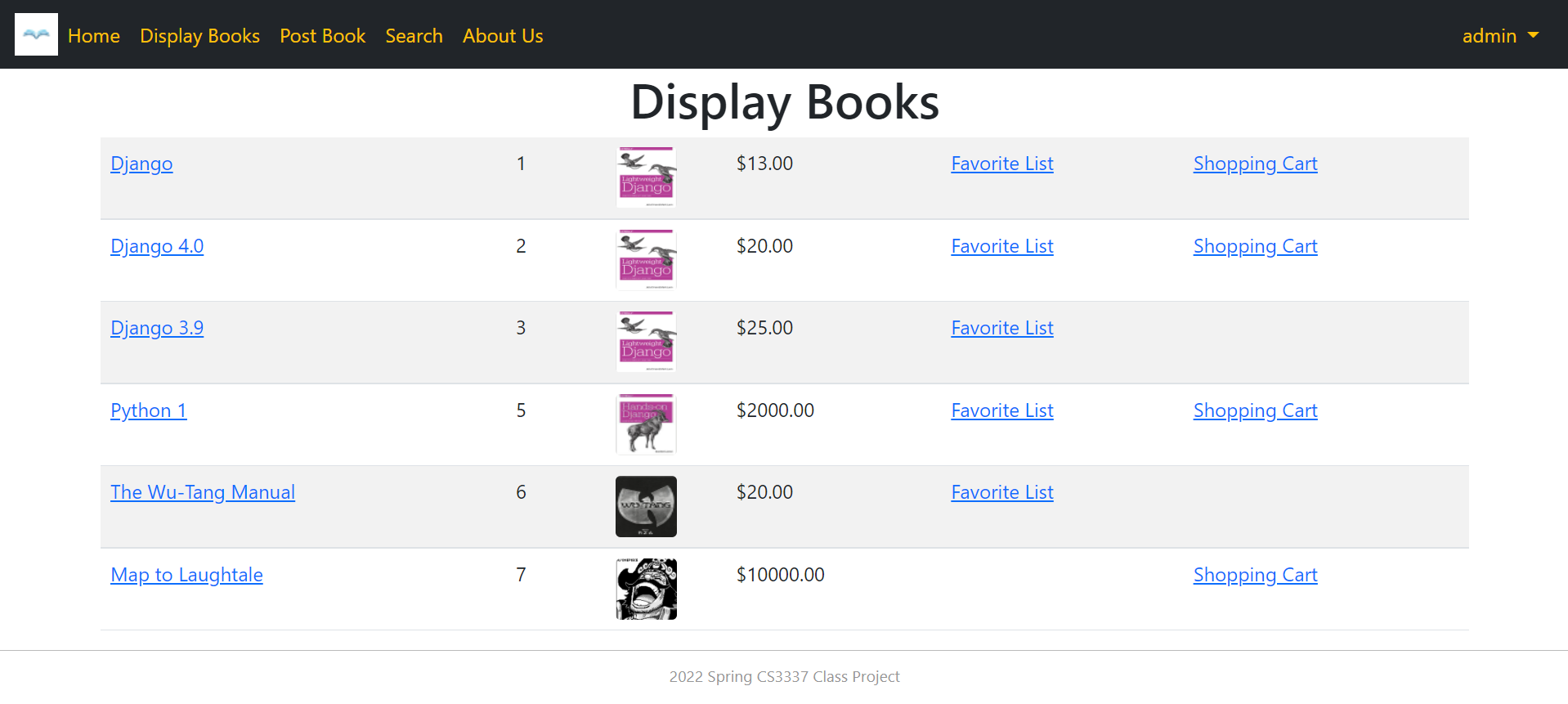
Login/Registration Page



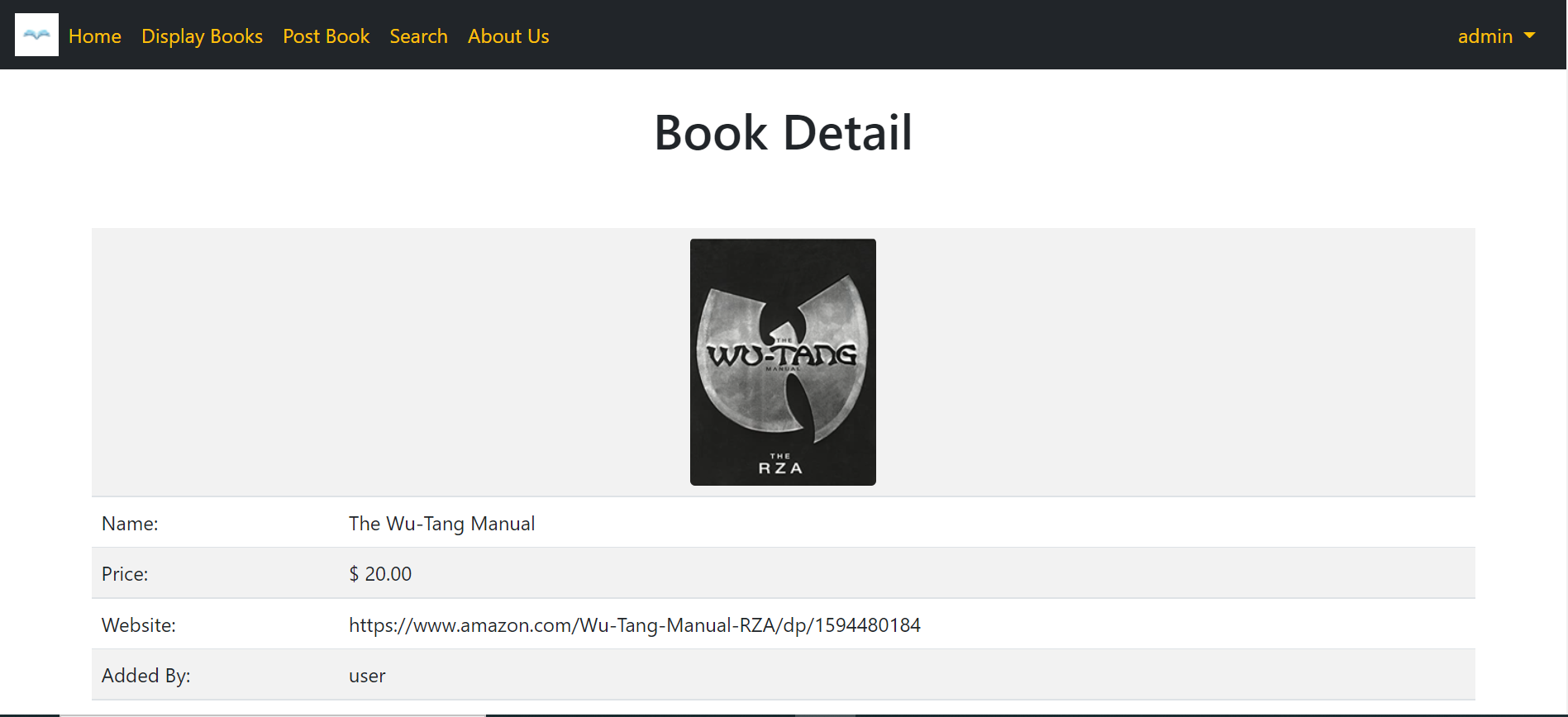
Display Books



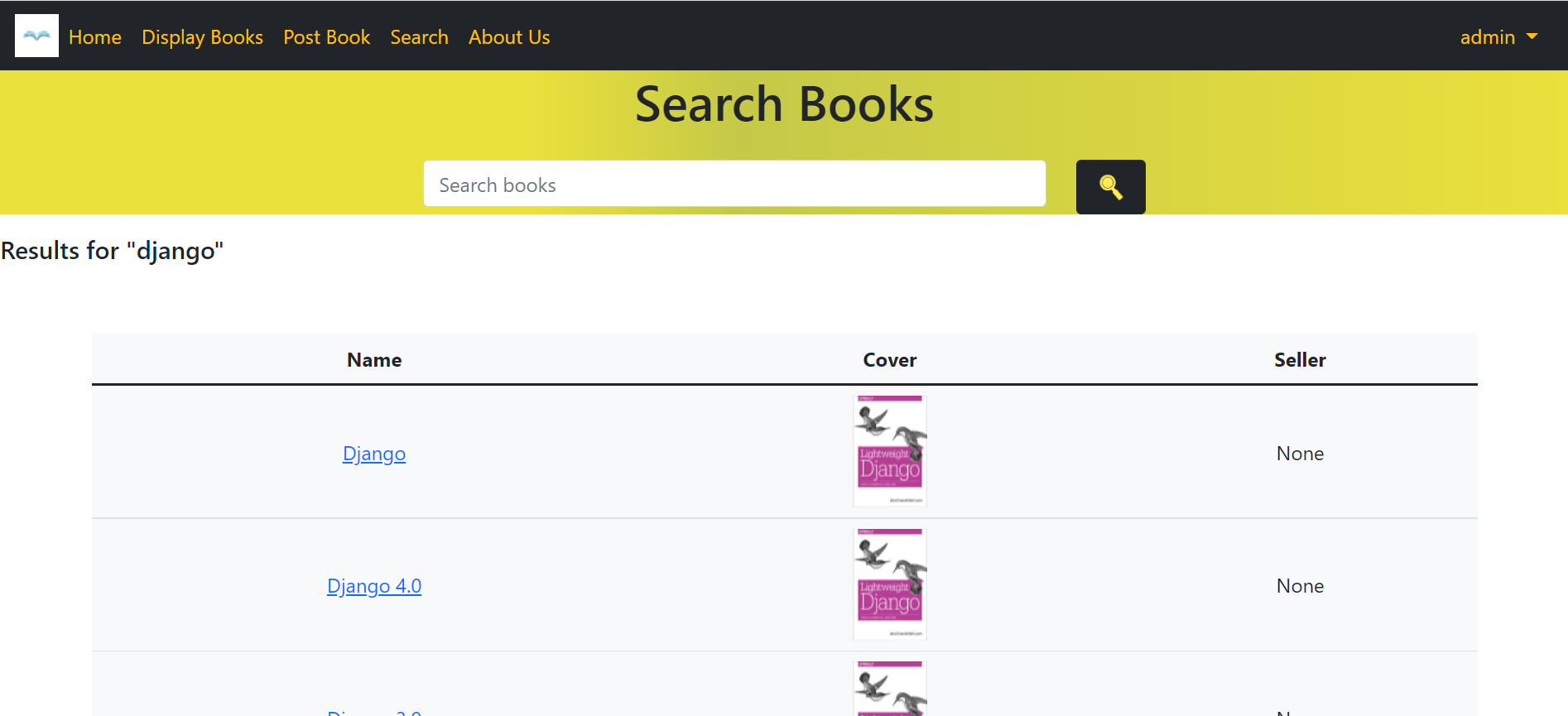
Post Books



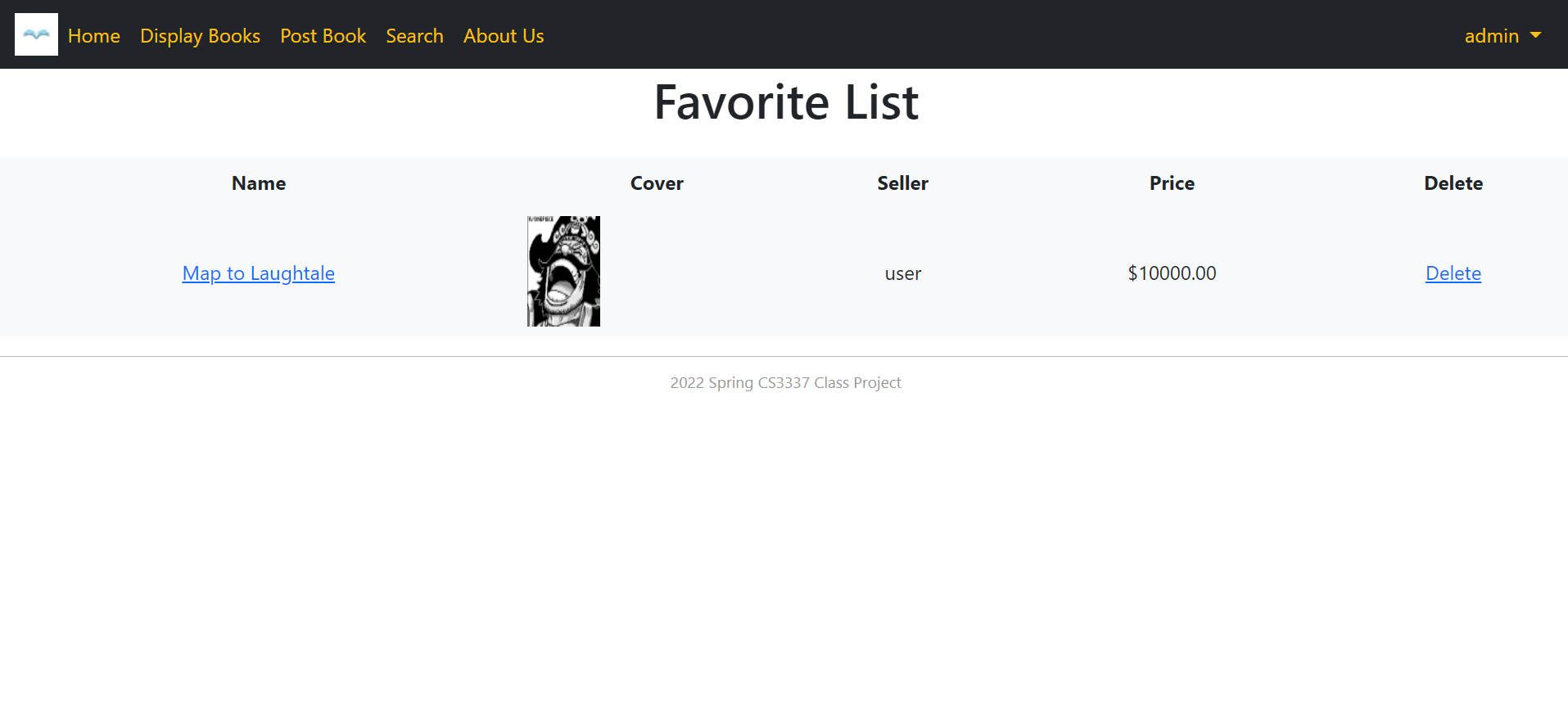
Book Details



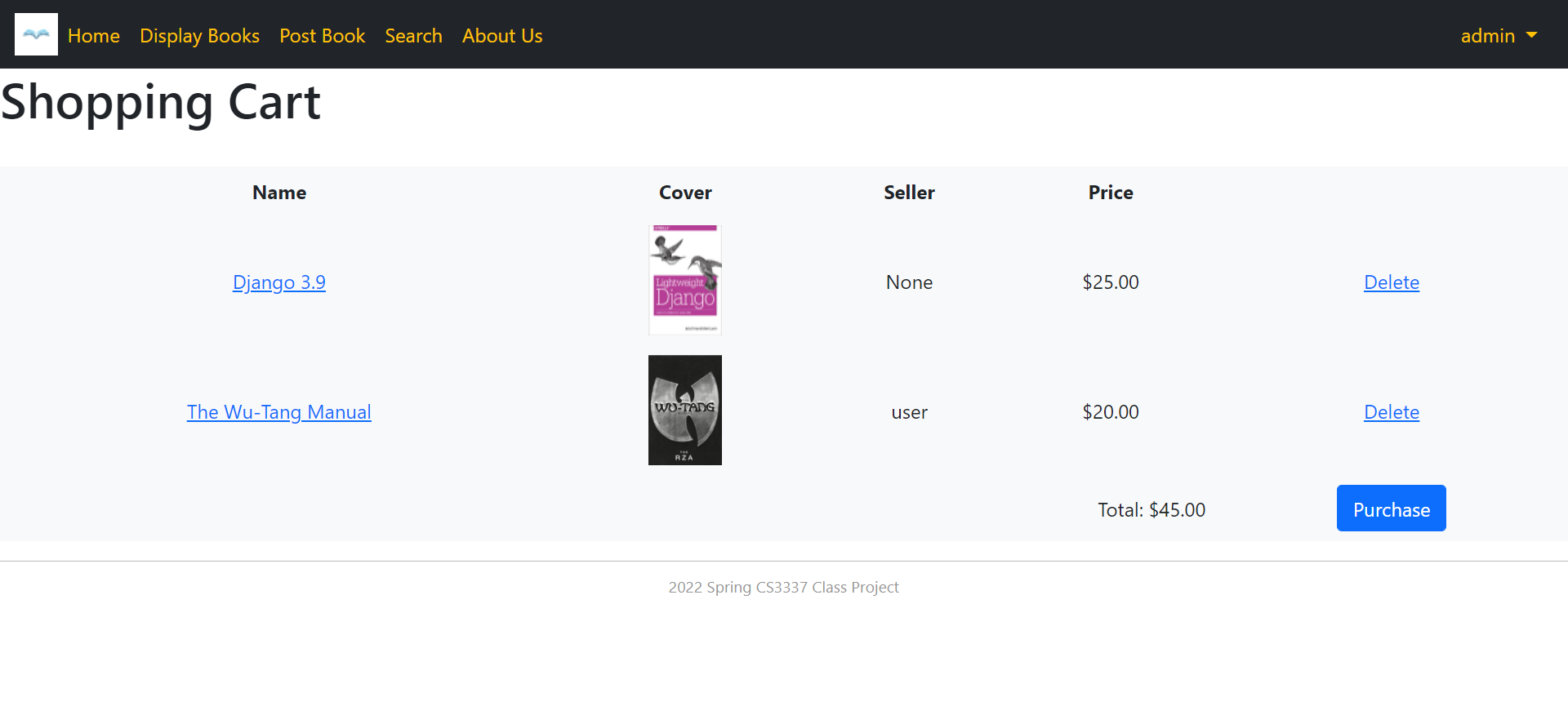
Book Search



Favorite List



Shopping Cart



# 8. Glossary

8.1 Django: Python based free and open source web framework that follows the Model-View-Template (MVT) architecture.

8.2 Bootstrap: Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

8.3 SQLite3: Relational database management system contained in a C library. Not a client–server database engine. Rather, it is embedded into the end program.

8.4 HTTPS: Hypertext Transfer Protocol. It is used for secure communications over a network and is widely used. The communication protocol is encrypted using Transport Layer Security.

8.5 CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.