

DOCUMENTATION

Documentation of the Bookstore Website using HTML, CSS, JavaScript, SQL and Python.
Using the Django Framework.

The project is a Bookstore website. Allows users to buy books and manage their accounts.

The Implementation of the Bookstore Website uses the Django Framework.

The framework is a Python-based web framework that is used to create web applications.

It uses a Model-View-Controller (MVC) architecture and the HTML structure, with forms used to create dynamic web pages.

The framework also adopts the MVC architecture and uses the Django ORM to create the database.

The administration of the website is done using the Django Admin which allows a superuser to manage the website.

The application uses a base settings structure that allows for adding components to the website.

The design of the website is made using the CSS framework and JavaScript framework.

The SQL

Statements for each transaction in the database are as follows:

- Connect to the database:

```
connection = sqlite3.connect('bookstore.db')
cursor = connection.cursor()
```

- Create a table for the users:

```
cursor.execute("""CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT,
    password TEXT,
    email TEXT,
    first_name TEXT,
    last_name TEXT,
    address TEXT,
    city TEXT,
    state TEXT,
    zipcode TEXT,
    phone TEXT,
    is_admin INTEGER
)""")
```

- Create a table for the books:

```
cursor.execute("""CREATE TABLE books (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    title TEXT,  
    author TEXT,  
    year INTEGER,  
    isbn INTEGER,  
    price REAL,  
    quantity INTEGER  
)""")
```

- Create a table for the orders:

```
cursor.execute("""CREATE TABLE orders (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    user_id INTEGER,  
    book_id INTEGER,  
    quantity INTEGER,  
    date_added TEXT  
)""")
```

- Create a table for the order items:

```
cursor.execute("""CREATE TABLE order_items (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    order_id INTEGER,  
    book_id INTEGER,  
    quantity INTEGER  
)""")
```

- Create a table for the reviews:

```
cursor.execute("""CREATE TABLE reviews (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    user_id INTEGER,  
    book_id INTEGER,  
    rating INTEGER,  
    comment TEXT  
)""")
```

- Create a table for the writers:

```
cursor.execute("""CREATE TABLE writers (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    first_name TEXT,  
    last_name TEXT,  
    email TEXT,  
    phone TEXT  
)""")
```

```
)""")
```

- Create a table for the categories:

```
cursor.execute("""CREATE TABLE categories (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name TEXT  
)""")
```

The Database is created using the SQLite3 module and contains the tables
django_migrations, auth_user, auth_user_groups, auth_user_user_permissions,
auth_group, auth_group_permissions, auth_permission, auth_message,
django_session, auth_user_temp, store_category, store_review, store_slide,
store_writer, store_book, store_order_item, account_user, account_profile,
account_post, account_comment.

- USER GUIDE FOR THE BOOKSTORE WEBSITE:

- To setup environment for the website:

- Open the terminal and navigate to the directory where the project is located.

- Activate the virtual environment.

- Run the command:

- \$ source venv/bin/activate

- Run the following command:

- python manage.py startapp bookstore

- The command will create a new folder called bookstore in the directory where the project is located.

- The command will create a new file called bookstore/__init__.py in the directory where the project is located.

- The command will create a new file called bookstore/admin.py in the directory where the project is located.

- The command will create a new file called bookstore/apps.py in the directory where the project is located.

- The command will create a new file called bookstore/urls.py in the directory where the project is located.

- The command will create a new file called bookstore/views.py in the directory where the project is located.

- The command will create a new file called bookstore/wsgi.py in the directory where the project is located.

- The command will create a directory called accounts, files, cart, order, search and store.

- The command will create the database called db.sqlite3

- The command will create a new file called manage.py in the directory where the project is located.

- To run the website:
 - Open the terminal and navigate to the directory where the project is located.
 - Run the following command:
 - `python manage.py run server`
 - The command will run the website.
- To create a new user on the website:
 - Open the port on `http://127.0.0.1:8000`
 - The user will be able to register on the website.
- To login to the website:
 - Enter the created username and password on the website.
- To log out of the website:
 - Click on the username on the website.
 - Click on the logout button.
- To add a book to the website:
 - Click on the username on the right side of the website.
 - Click on the add book button.
 - Enter the book title, author, year, ISBN, price and quantity.
 - Click on the Save button.
- To search for a book on the website:
 - Click the search bar on the top of the website.
 - Enter the book title.
 - Click on the search button.

After placing books in the cart users can choose to checkout or continue shopping.

- To checkout:
 - Click on the cart icon on the website.
 - Click on the checkout button.
 - Enter in user and shipping information.
 - Click on the place continue to checkout button.
 - Then the user will be greeted with the information entered
 - Users can download a pdf of the order.

API Documentation is Published Here:

<https://documenter.getpostman.com/view/20492702/Uyr5nJsJ>