

Web Programming Language – CS 6314

BYOB.COM (BUY YOUR OWN BOOK) (BOOKSTORE) TEAM 21

By:

Faraz Khalid (fxk150530)

Humayoon Akhtar Qaimkhani (hxq190001)

Raisaat Atifa Rashid (rar150430)

Spring 2020 – University of Texas at Dallas

Introduction

For our final project for the class, the team decided to make a bookstore. As a team, we took inspiration from the Moravian bookstore, a bookstore founded in 1745 in and is renowned for being the oldest bookstore in the United States. To further explain what we mean by taking inspiration from the Moravian bookstore, we mean that we wanted to keep our interface simple. The UI was designed keeping in mind that our website would target age groups ranging from teenagers to elderly.

In our first group meeting, while brainstorming ideas on what functionalities should be implemented and how the UI should look like, we came up with several mock-ups, some very detailed and extravagant while some as simple as having the bare minimum things displayed on the webpage to please the eyes of the user. Eventually, we decided to follow what once Steve Jobs said and we quote, “focus and simplicity. Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple.” Hence, we made the decision and went with a simple UI, that we have now.

In another one of our group meetings, we decided on the following function requirements for the end users that were kept in mind while working on the project.

- The filter feature shall filter by categories.
- The user shall be able to filter by name.
- The user shall be able to filter by price ranges.
- The user shall be able to add books to their cart.
- The user shall be able to remove books from their cart.
- The user shall be able to checkout after they are done shopping.

- The user shall be able to see their order history.
- The search time shall be less than 2 seconds.
- The password shall be hashed, for security purposes.
- User shall have all credible fields when signing up, as per the requirements mentioned by Dr. Yuruk.

From an administrator perspective, the following functional requirements were signed off by all the team members, in accordance with the requirements mentioned by Dr. Yuruk

- The admin UI should be similar to the user UI, with privileged functionalities added such as adding, removing and updating the books.
- The admin shall be able to add a book.
- The admin shall be able to update the book.
- The admin shall be able to remove a book.
- The admin shall be able to see all the books in the database.

Below mentioned are some of the non-functional requirements that were gathered during our meeting

- The website shall be scalable, giving options to add more books.
- The website's response time for each action shall be less than 2 seconds.
- The website shall be available with no downtime pages.

Database Design

For database, we worked with MySQL.

Our database consists of the following tables

- user
Has information about the user.
- admin
Has information about the admins of the website.
- cart
Has information about what the user adds to the cart.
- books
Has information about the books.
- orders
Has information about the orders from the users.
- order_address
Contains the address for each order.

Languages/Framework used for implementation:

The following was used for developing the application

- HTML
- CSS
- JavaScript
- JQuery
- Bootstrap 4
- PHP

Work Division:

Initially, we all started working on ideas together, such as creating a mock-up and initially designing how our database would look like. As the project further progressed, we distributed tasks accordingly.

To get the most out of the project, we went through the Software Development Lifecycle Model, starting with requirements analysis, then moving on to software architecture, and then implementation. We all worked together to find and document all our functional and non-functional requirements that we wanted to include in the project.

For the implementation phase, all team members were asked to work on both, the front-end design and back-end implementation so that everyone could contribute equally. Faraz started with the initial web design and got help from both, Raisaat and Humayoon to build the user interface for admin and users. Raisaat started development on adding everything to the database and getting data for the database (book information), and with the help of the other two teammates was able to get everything up and running. Humayoon spent major time making sure the implementation on the user side was working as it is supposed to, with everything being added to the cart appropriately and checking for all the validity that comes with it. Again, the team pitched in making sure everything was in accordance to the requirements set by Dr. Yuruk.

To sum up, each phase and each particular functionality is an effort of not one but all of the team members. After working on one thing as a team, we then moved forward together, with everyone pitching in each task as it helped us gather more ideas and develop our website with great efficiency. For most part, we implemented the trio-programming methodology (a variation of pair-programming but not with two but three people).