Phase II: User Requirements and Application Specifications

Submission Deadline: 18.03.2024, 23:59

1. **Chosen Development Model:**

For building an online bookstore Agile is chosen because of it being an relatively simple and not complex project . We also choose the Agile methodology for its flexibility and adaptability to changes in project requirements. It will allow for iterative development, frequent feedback, and continuous improvement, which are crucial for an online bookstore website where we plan to deliver the software to the customers and get rapid feedback to continue with the development of the project.

2. **User Requirements:**

* Stakeholders:

- End-users:Customers which visit the bookstore website to effortlessly browse, search, and buy books.

- Clients:Business owners or managers who have an interest in the performance of the website and who own or run the bookstore.

- Developers:The development team in charge of developing, maintaining, and designing the website.

* User Stories:

- As an end-user: I would like to be able to search for books easily by title so that I can quickly find the books I'm interested in.

- Benefit: Enhanced user experience and higher chance of locating books that are wanted.

- As an administrator :I would like to keep the catalog up-to-date, I wish to manage book listings, which includes adding new books and removing out-of-date ones.

- Benefit: Ensures that the books that are provided to clients are accurate and relevant.

-As the end user: I would like that in order to make accurate purchasing selections I want to see comprehensive book descriptions that include summary, category and price.

- Benefit: Improves consumer experience when shopping and helps in decision-making.

-As the end user: I would like to be able to easily contact the business by sending a message.

-Enhances user experience, costumer relationships and satisfaction.

3. **Functional Requirements:**

* Brief Description:

- The system shall allow users to sign up an account or log in if they are already registered.

-Users shall be able to view product listings.

-Users shall be able to browse books by categories.

-Users shall be able to search for books by title.

-Users shall be able to view detailed information about each book, including its description category and price .

- Users shall be able to add books to their shopping cart, proceed to checkout, and complete the purchase.

-The users shall be able to contact the business “Librari Art’s” through a contact form

- The system shall provide administrators with tools to manage book listings.

* Acceptance Criteria:

- The user registration and login capabilities have been implemented and tested.

-Users can browse through books.

-Users can filter books according to their categories.

-Users can search for books based on their titles.

-Users can view the full details about each book.

- Users can add books to their cart, look at the contents, and successfully finish the checkout process.

- Users can contact the business via a contact form.

- Administrators can add or remove book listings.

4. **Non-Functional Requirements:**

* Brief Description:

- The system must be responsive and offer a consistent user interface across all platforms.

- It must be safe, with authentication procedures securing user information.

-The website needs to be responsive to different screen sizes and devices.

- The website must be expandable to handle rising traffic and a growing book inventory.

-The website must have an easy-to-use interface and navigation.

* Acceptance Criteria:

- The average loading time of the website is 3 seconds, guaranteeing a quick and responsive experience.

- Authentication systems are used to secure user accounts.

-The website functions flawlessly on a range of screen sizes, including those of computers, laptops, tablets, and smartphones.

- At least 450 concurrent users may be supported by the system without noticeably degrading performance.

-The user interface (UI) of the website is evaluated across a range of groups to ensure that it is as easy to use as possible for users.

5. **Application Specifications:**

* Architecture:

-The architecture is based on a microservices model, which allows different system components to be created and deployed independently. A microservices design allows to develop different aspects of the website separately.Microservices integrate each business function (for example, catalog management, product listing, and login/signup capability) as a discrete service. This method makes it easier to change or add new features to the system without affecting other components. In this architecture type, failure in one service does not always affect the entire system.

* High-level Descriptions Of System Components:

*User Interface (UI):*

-The UI component is the front-end of the bookstore's website, where customers interact with the system.

-It has online sites for browsing books, searching for specific titles, viewing book details, managing the book catalog, placing orders etc.

-To retrieve data and conduct actions, the UI makes API calls to the backend services.

*Authentication Service:*

-User authorization and authentication are managed by this service.

-It offers features for logging in, and registering users.

-User credentials are validated by the Authentication Service.

*Catalog Service:*

-The book catalog and associated data are managed by the Catalog Service.

-Users can browse books, look for titles, and read book data (such as the price, genre, and description).

-In order to update catalog information and obtain book data, this service interacts with the Database Service

*Order Service:*

-Order management is the responsibility of the Order Service, which also manages features like adding products to the shopping cart, changing quantity, and creating order invoices.

-To obtain the information required about the orders and save the invoices, the Order Service communicates with the Database Service.

*User Services:*

-Book catalogs are managed by the user service.

-It manages functions like updating and removing out-of-date books from the book listing.

-In order to store and retrieve user data, this service interacts with the Database Service and the Authentication Service.

*Contact Form:*

-The contact form enables the user to contact the business

-Through filling the forms fields ( name,email,phone number, message) the clients can send a message directly to the business

-This services communicates with the Database service to store the data

*Interactions:*

-To authenticate users and retrieve user-specific data, the User Interface communicates with the Authentication Service.

-Book information is retrieved from the Database Service via the Catalog Service, which allows users to look up books and read details.

-With the Order Service, users can add products to their shopping cart and check out.

- Users can contact the business through using the Contact Service.

-The User Service is in charge of the book catalog and communicates with the Database Service to store and retrieve data, as well as the Authentication Service for user authentication and authorization.

* Database Model:

*Tables:*

1.Clients:

-Fields: client\_id(Primary Key),Client\_username,Client\_password

2.Contact:

-Fields: contact\_id(Primary Key),name,email,phone number, message

3.Admin:

-Fields: admin\_id(Primary Key), admin\_username, admin\_password

4.Books:

-Fields: book\_id(Primary Key), book\_title,book\_img,book\_description,book\_price,category

5.Sales:

-Fields:sale\_id(Primary Key), book\_id (Foreign Key references Books table),client\_id(Foreign Key references clients Table),sale\_date,quantity,subtotal,total

### *Relationships:*

Sales and Clients: One client may have many sales.

Clients and sales have a one-to-many relationship .

Books & Sales: A single book may be sold more than once.

Books and sales have a one-to-many relationship.

Books and Admin: One admin can supervise several books.

Admin and Books have a one-to-many relationship .

### *Constraints:*

-Primary Key Constraints:

A primary key constraint is present in every table to guarantee that every record has a unique identity.

Examples: client\_id in Clients table, book\_id in Books table, etc.

*-*Foreign Key Constraints:

The Sales table references the Clients and Books tables through the usage of foreign key restrictions.

Ensures the link between tables and guarantees referential integrity.

*-*Normalization:

Tables are 1st Normal Form (1NF) and further normalized for better efficiency and organization.

*Technologies Used:*

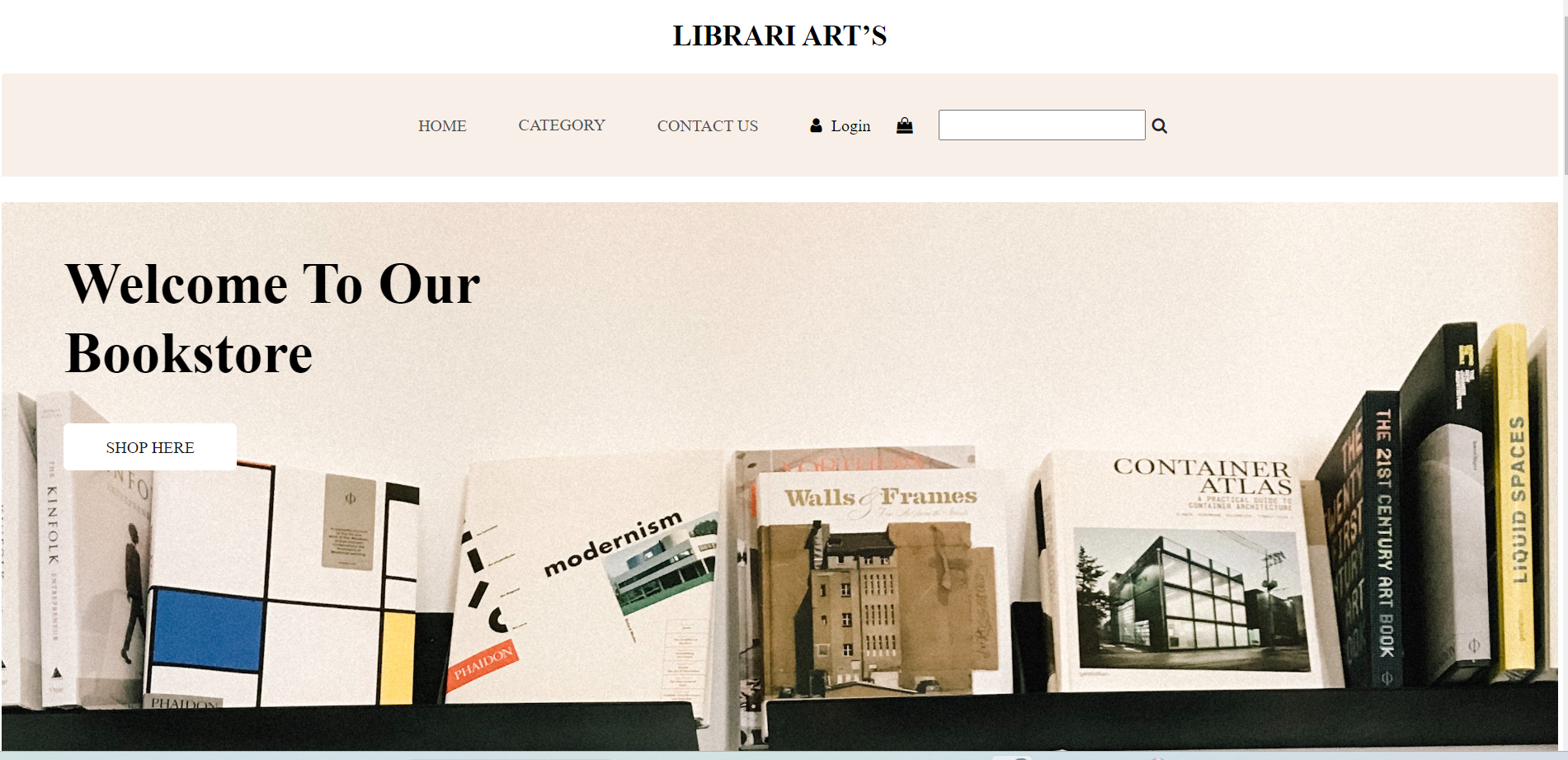
- Frontend: The website pages are given structure and style using HTML and CSS. These webpages will include interactive features that captivate users thanks to JavaScript. Finally, we will use Bootstrap to ensure that the website functions properly across a range of devices with varying screen sizes.

- Backend: A local web server environment for testing and development will be provided via XAMPP, while PHP will be utilized for server-side functionality.

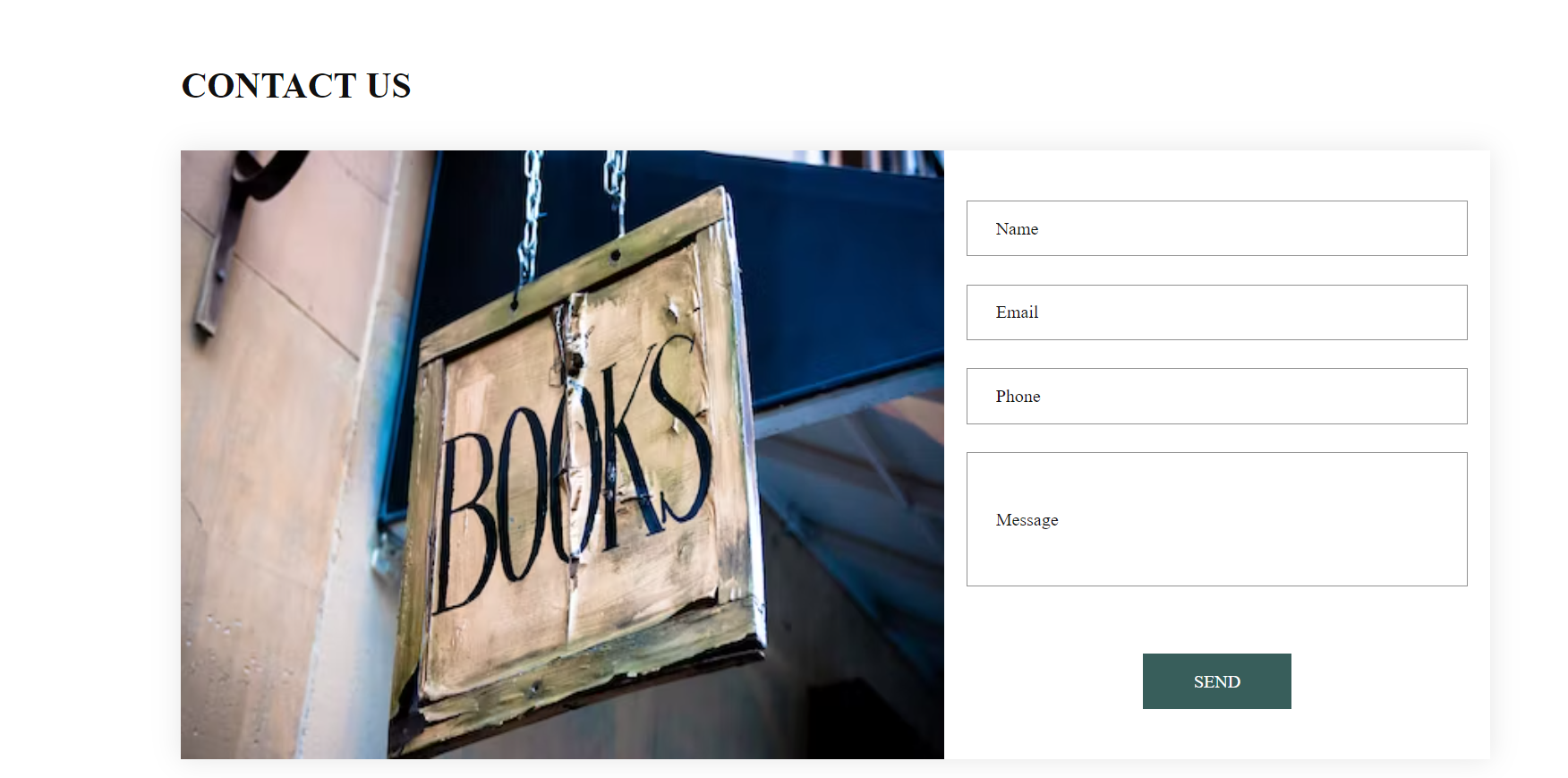
-Database: MySQL is used to store the tables and each table's data.

* User Interface Design:

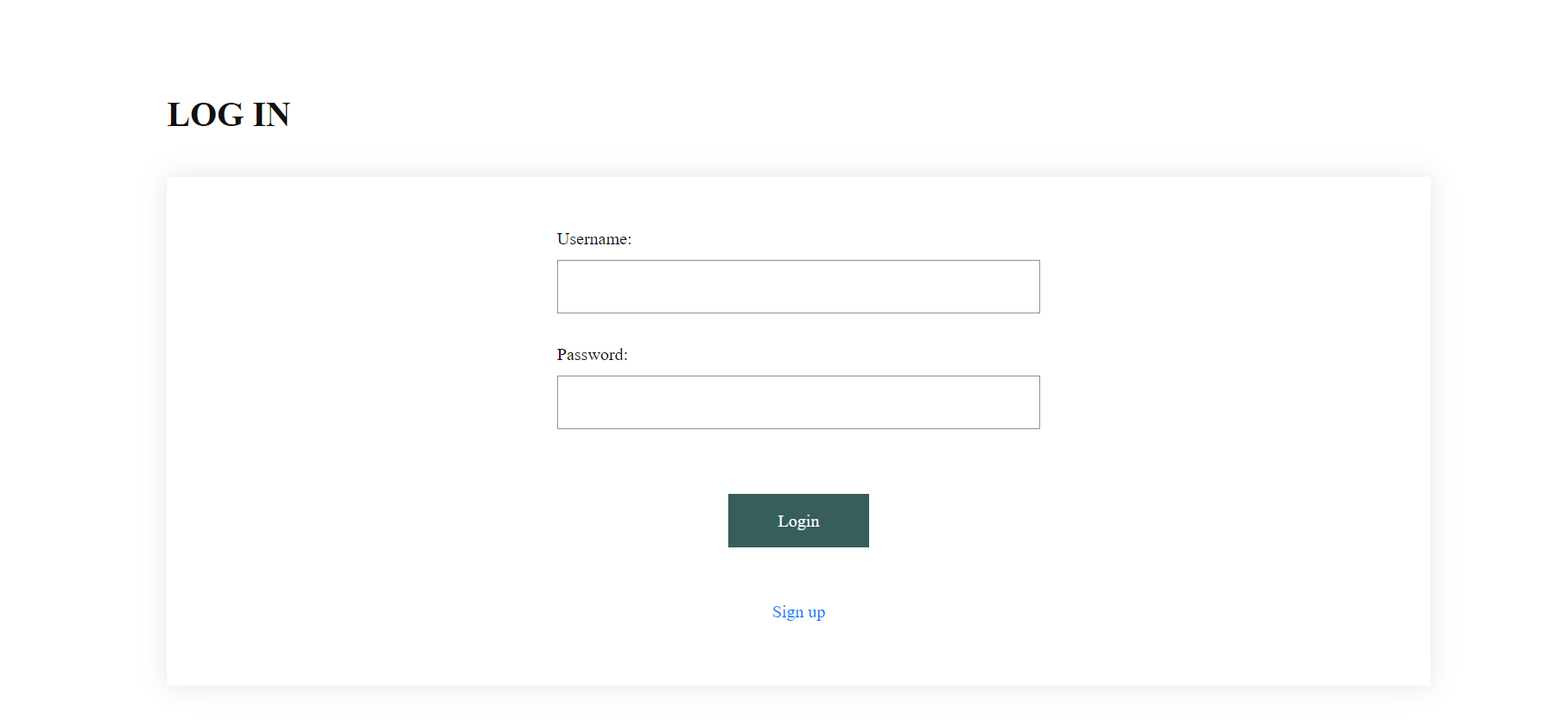
Home Page:



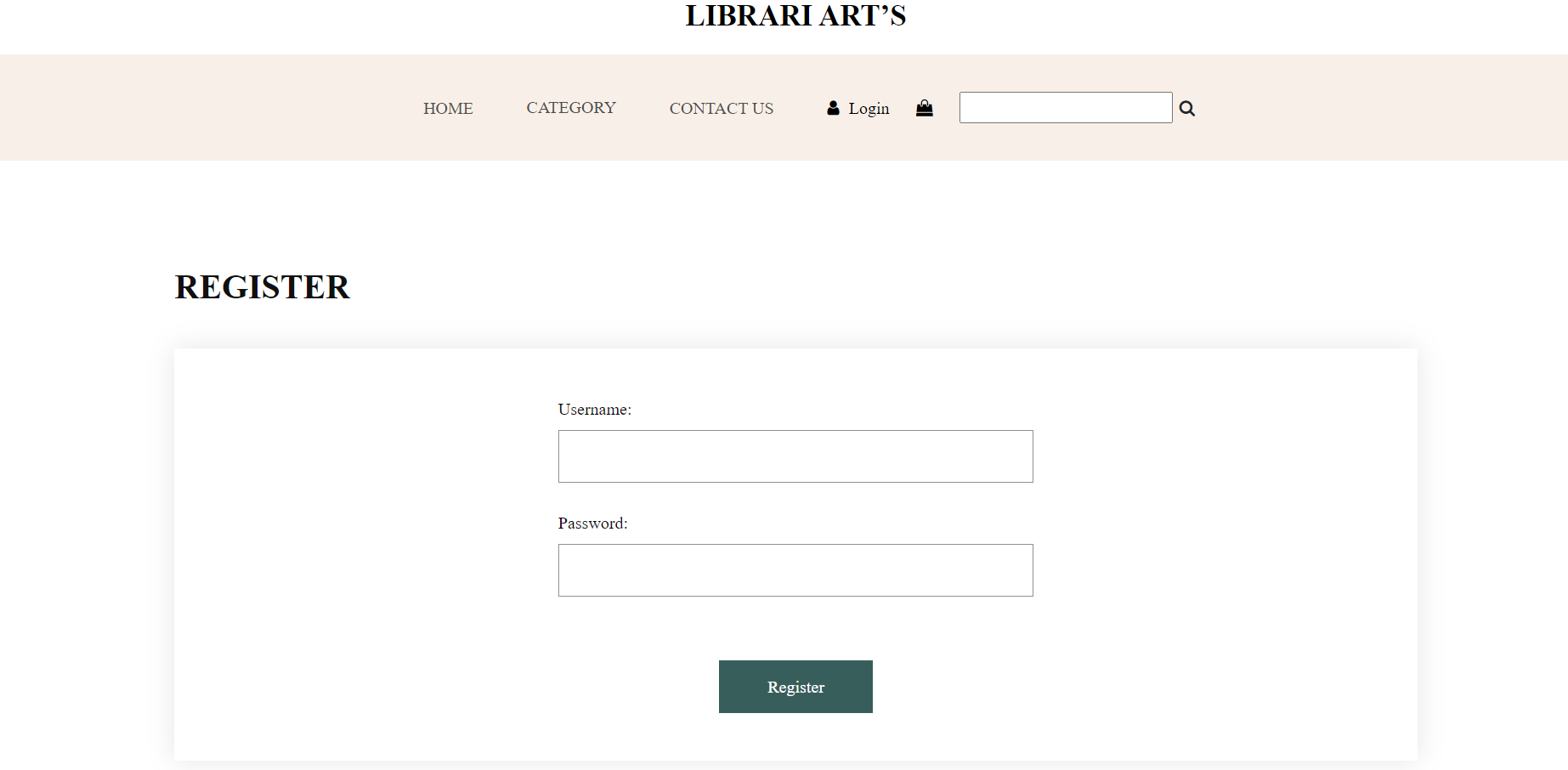
Contact:

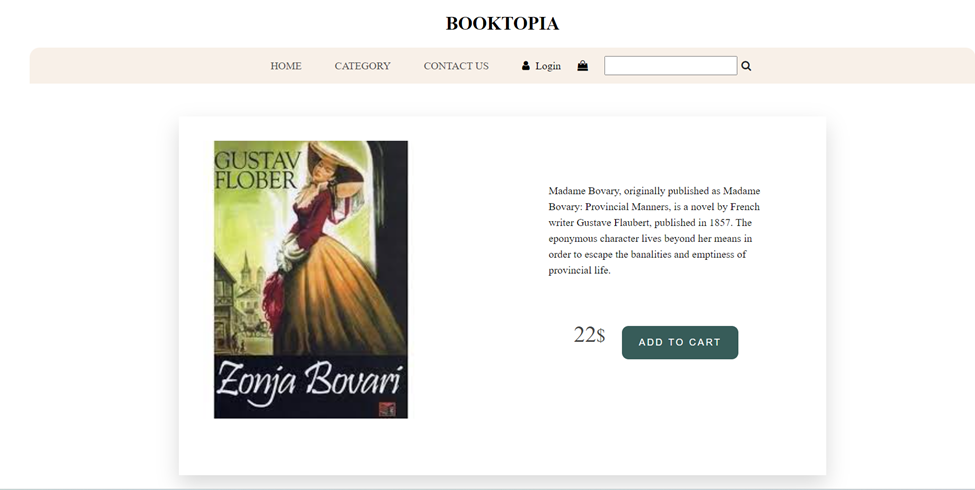


Login:

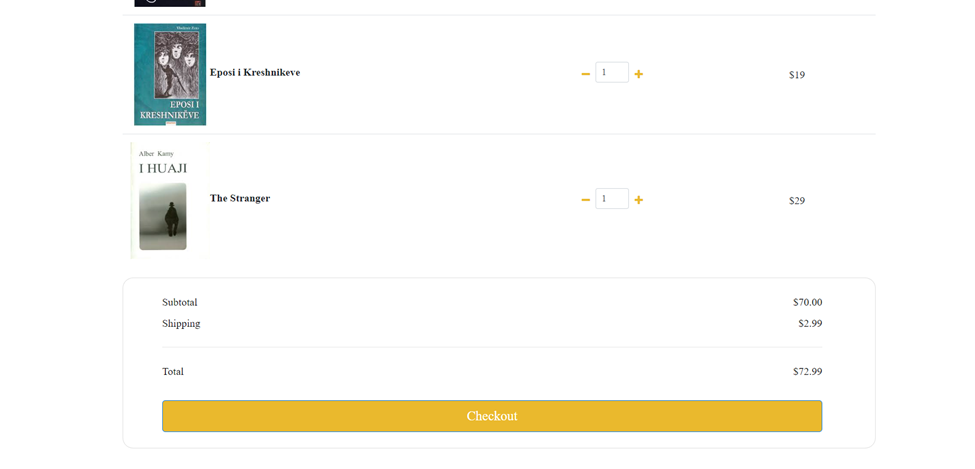


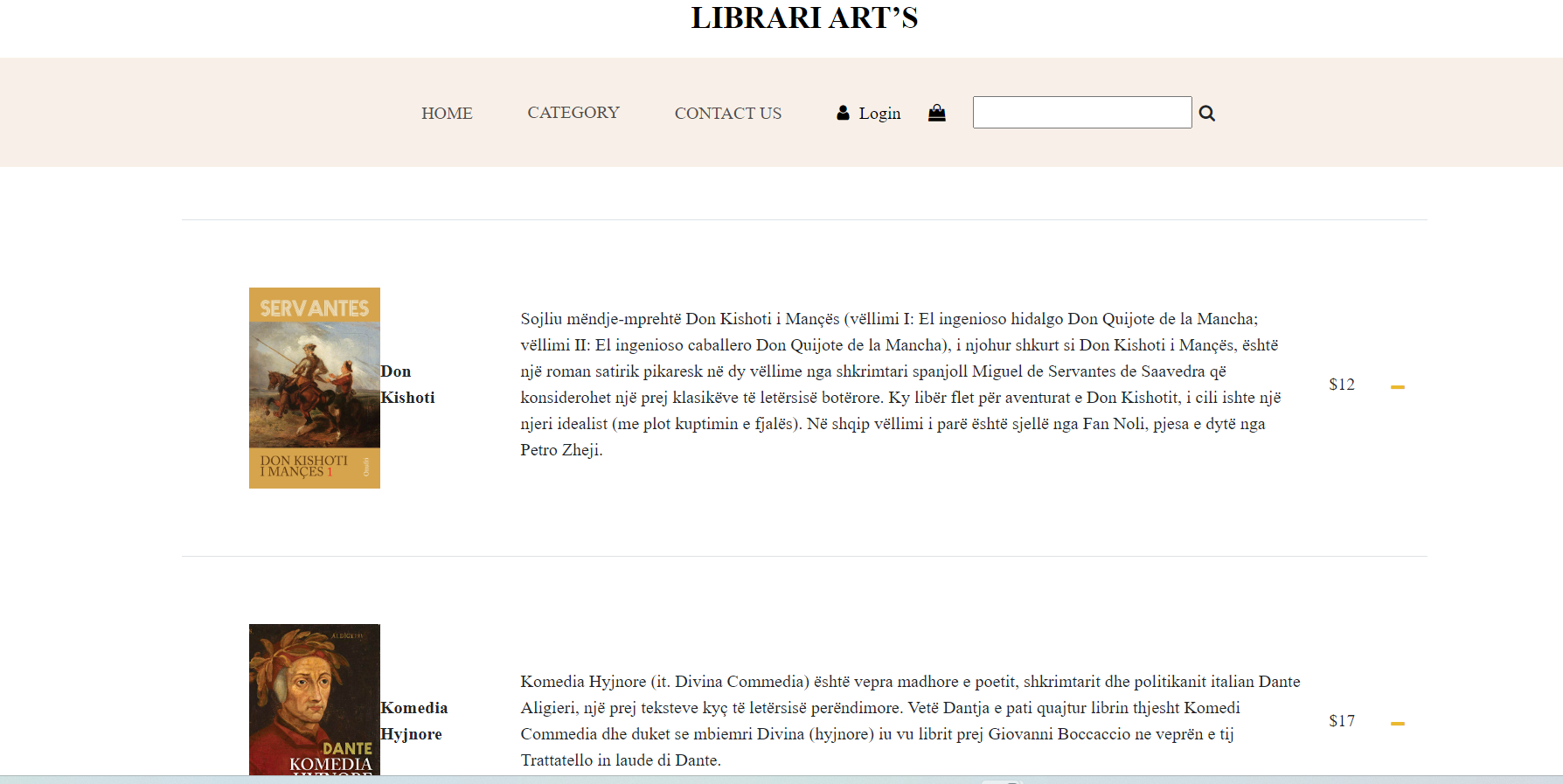
Register:



BookDetail:

Shopping cart:



Admin Management:

*Visual Representation:*

* User Registration/Login:
  + When users visit the website,and click the login button a login/registration screen will appear.
  + By entering the required details, including their username and password, new users can register.
  + Users who have registered can sign in with their credentials.
* Browsing Books:
  + Users will be taken to the homepage after logging in, where they can browse books in several categories.
  + Users can click on a particular book to access its details.
* Adding to Cart:
  + On the book's detail page, users can click the "Add to Cart" button to add books to their purchasing basket.
  + The system will add the new item(s) and update the cart summary accordingly.
* Checkout Process:
  + Users can go to the checkout page when they're ready to make a purchase.
  + They can check what's in their cart, adjust the quantities, and then proceed the payment.

Admin Management:

* + Book listings can be managed by administrators through a different admin interface.
  + To keep up the bookstore catalog, administrators can add new books, update old ones, and delete titles that are no longer relevant.

Contact:

* Customers can get in touch with the company.
* By completing the required fields, users can send a message to the company (ex. Name, email etc).

*Security Measures:*

-Encryption:Hashing user credentials prior to storage is done through encryption.

- Authentication: In order to log in, users must have valid credentials.

-Continuous updates and audits of security.