Infinite Tales Technical Document

Database setup:

The infintetales database was created on phpMyAdmin. 7 tables were created.

Table Details:

Address: contains the address of a user.

Book: contains the information of each book.

Cart: contains the current cart of the user.

Orders: contains the orders of the user.

Review: contains the product reviews.

Users: contains the login information of the users.

Wishlist: contains the wish listed items of the user.

Relationships:

- **users** is linked to **address** are linked with **addressId**
- **review** is linked to **users** through **userId**
- **review** is linked to **book** through **productId**
- **book** is linked to **review** through **reviewId**
- **wishlist** is linked to **users** through **accountId**
- **wishlist** is linked to **book** through **bookId**
- **orders** is linked to **book** through **bookId**
- **orders** is linked to **users** through **accountId**
- **cart** is linked to **users** through **accountId**
- **cart** is linked to **book** through **bookId**

Key Fields:

Address has **addressId** as **Primary Key**

Users has **accountId** as **Primary Key**

Review has **reviewId** as **Primary Key**

Book has **bookId** as **Primary Key**

Cart has **cartId** as **Primary Key**

Orders has **orderId** as **Primary Key**

Wishlist has **wishlistId** as **Primary Key**

Data Manipulation Techniques

CRUD Operations:

Create:

Adding a new book.

Submitting a review.

Creating account.

Create address.

Create Wishlist.

Create Order.

Create Cart.

Read:

Fetching book details for display.

Retrieving user reviews for a specific book.

Fetching address information.

Fetching user information.

Fetching cart details.

Fetching order details.

Update:

Modifying book information.

Editing or deleting a user review.

Editing address.

Add to Wishlist.

Add to Order.

Add to Cart.

Delete:

Removing a book from the system.

Deleting a review.

Remove item from wishlist.

Remove item from cart.

Virtual Server

To replicate a live server environment during development, XAMPP, a cross-platform web server solution, was chosen. Following the installation from https://www.apachefriends.org/, the XAMPP software was installed and initiated. This involved starting both the Apache HTTP Server and MySQL Database Server components.

Server Software Used:

Apache HTTP Server: It serves as the web server, handling HTTP requests and enabling access to PHP files.

MySQL Database Server: This relational database management system is employed for storing and retrieving data for the PHP web application.

Server Configuration:

Accessing phpMyAdmin: By navigating to localhost/phpmyadmin/, the phpMyAdmin interface was accessible. This tool facilitated the management of MySQL databases, allowing for database creation, modification, and interaction.

Localhost: The local server environment was easily accessed through localhost, providing a base URL for the PHP website.

The setup using XAMPP closely mimics the configuration of a live server by incorporating both web server and database server components. This local environment allows for seamless testing and development of PHP web applications. The use of localhost as the entry point provides a consistent experience with URLs similar to those used in live production environments.

Building a dynamic web application

To build this web application, first a prototype was made with HTML and CSS. HTML for defining the content and layout of your web pages, and CSS for styling the elements and making the bookstore visually appealing.

Server-side scripting with PHP was used to handle dynamic content generation. Interaction with the database to store and retrieve book, user, review, cart, order and address information using XAMPP.

Form handling to manage user input, such as submitting new books, updating book details, updating address, and creation of users.

Proper database design to store information about the tables, such as books which includes details like title, author, genre, and price.

Use of SQL queries to interact with the database for tasks like adding, updating, or retrieving information.

PHP scripting to dynamically generate HTML content based on the data retrieved from the database. This includes displaying a list of books, individual book details, and any other relevant information.

Providing informative error messages to users and logging errors on the server for debugging purposes.

Testing your application to ensure that all features work as expected.

Test Cases

Input	Process	Output
Book details	Insert new book record	Confirmation message or error
Review	Insert Review	Confirmation message or error
Create User	Insert information	Confirm creation or error handling
Update Address	Insert new address	Update address
Update wishlist	Insert/Remove book into wishlist	Update wishlist
Update cart	Insert/ Remove book into cart	Update cart
Update orders	Insert/ Remove books into orders	Update orders
Insert Review	Put in Review	Confirmation message or error