Awesome Book Market

Xinwei Lin & Xufeng Liu

Group 17

Link to our Website: http://flip1.engr.oregonstate.edu:20200/

A. Executive Summary:

From the beginning, we misunderstood the purpose of creating such a website. We thought that this is a website created for users to actually use and buy books. We were wrong. After reviewing assignment documentation, feedback from our peers, and also the instructor's announcement, we now consider our website like a management system that is provided for the software developer of the company, so they can intuitively see the data and their relationships. It was the biggest mistake that we made throughout the term, and it means many of our work and design should be redefined and much of them were wasted. We worked together to have a new version of the website and removed useless parts, for instance, user login and user sign up. Instead of creating a website for users, we created a website for developers where we emphasize the data stored in the database and lists them. There are minor fixes after considering feedback from peers: not listing IDs in the table, creating new data will use dropdowns that show the representative names rather than IDs. We finally made a complete working website.

þi

CS 340 TEAM EVALUATION FORM

DECEMBER 8, 2020

RATE YOUR TEAMS PERFORMANCE USING THE SCALE BELOW.

1 = Strongly Disagree 2 = Disagree 3 = Agree 4 = Strongly Agree

GROUP NUMBER		
NAME OF GROUP TEAM MEMBERS:		
SCALE AND COMMENTS	RATING	ADDITIONAL COMMENTS
HOW PREPARED WAS YOUR TEAM? Research, reading, and assignment complete	4	We submitted all assignment on time.
HOW RESPONSIVE & COMMUNICATIVE WERE YOU BOTH AS A TEAM? Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc.	4	We both are very responsive and communicate a lot.
DID BOTH GROUP MEMBERS PARTICIPATE EQUALLY Contributed best academic ability	4	Yes, we did work equally.
DID YOU BOTH FOLLOW THE INITIAL TEAM CONTRACT? Were both team members both positive and productive?	4	Yes, we all followed team contract throughout the term.

Are there any suggestions for improvement for your team and what are your goals moving forward?
(Better communication, follow the contract better, modify the initial team contract, more contribution, etc?)?

It is our last assignment, we are very glad to have each other as team member in this course because we had a good time working with one and another.

B. Project Outline and Database Outline:

Overview

Awesome Book Market sells first-hand books and second-hand books in the same place. Selling or buying books is your own choice on this website. By using the database, we save all information about the book, buyer, and seller, and for each purchase, we will link

the information and combine it as ordered. People will be able to check the order after they make a purchase. It will be flexible for people to buy and sell their books, and we keep track of the transaction to provide users a safe environment.

The book market trades 50,000 books in a year including old and new books. For each trade, there will be a special ID for each transaction, so users are able to check the history. A DB driven website will record orders of books of buying or selling. We assume the website will have around 2000 visits a day and there will be 5000 active users and have more than 50,000 user accounts saved in our database in the first year. For those users, they are going to create about 100 orders per day.

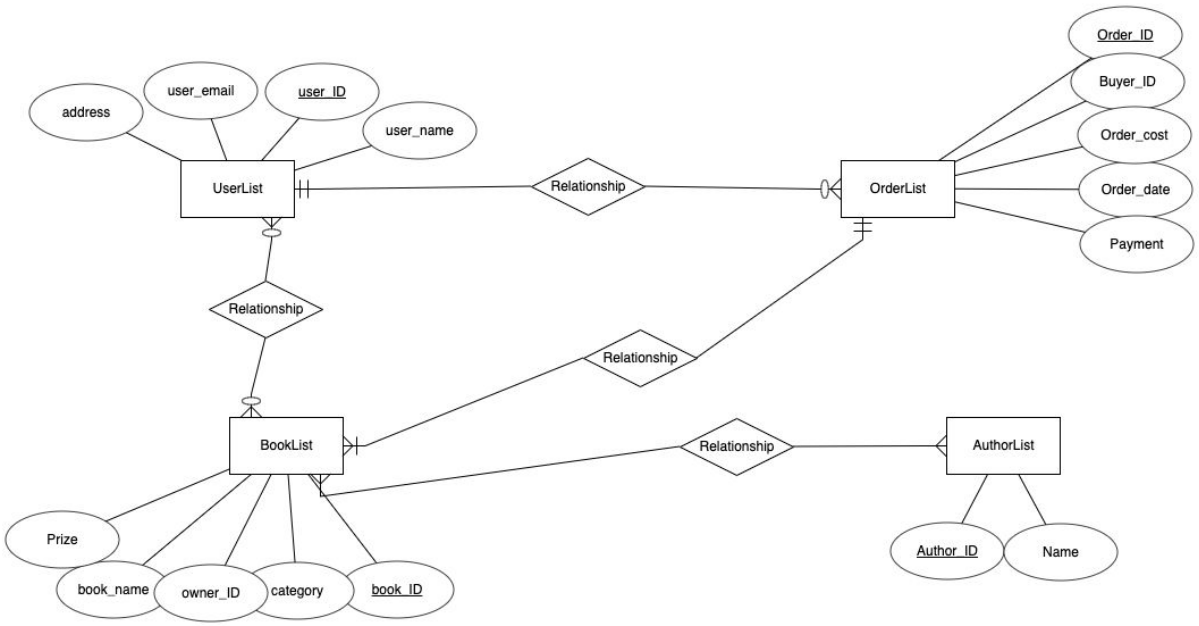
Database outline

- UserList (will be implemented by Lin)
    - User_ID: primary key, unique, int, auto_increment, not NULL
    - User_name: varchar, not NULL
    - Password: varchar, not NULL
    - User_email: varchar, not NULL
    - Address: varchar, not NULL
    - Relationships: A 1:M relationship between the User and Order. An M:M relationship between User and Address. A 1:M relationship between User and User_Address. A 1:M relationship between User and user_Order. An M:M relationship between UserList and Book.
    - User Entity will hold the user's information. A User can buy many Books in one Order, or separately. A User can post a Book in a Post for selling, as many Posts as he wants. A User can have many addresses, and an Address can be used by multiple users.
- BookList (will be implemented by Lin)
    - Book_ID: primary key, unique, int, auto_increment, not NULL
    - Category: varchar, not null
    - Book_name: varchar, not null

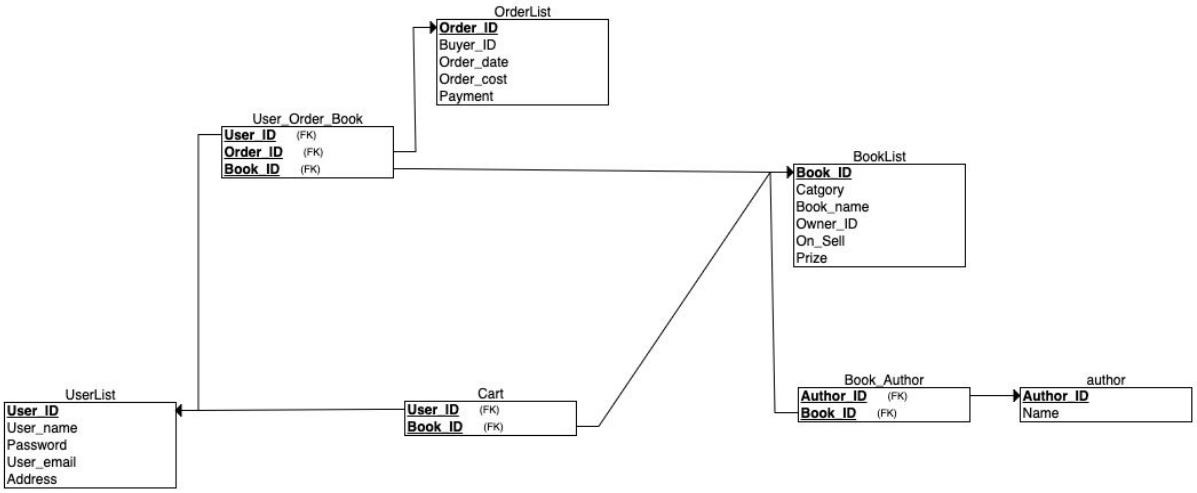
    - Owner_ID: int, not null, FK
    - Prize: int, not null
    - On_Sell: boolean, not null
    - Relationships: A 1:M relationship between Order and Book. A M:M relationship between Book and Author. A 1:M relationship between Book and Book_Author. A 1:1 relationship between Book and order_book.
    - The Book entity holds the information of a Book. A Book can only appear in a Post or Order. A Book can have many Authors. Authors can write many Books.
- OrderList (will be implemented by Lin)
    - Order_ID: primary key, unique, int, auto_increment, not NULL
    - Buyer_ID: int, not null, FK
    - Order_date: DATETIME, not NULL
    - Order_cost: int, not NULL
    - Payment: varchar, not Null
    - Relationships: A 1:M relationship between Order and User. A 1:M relationship between Order and Book. A 1:1 relationship between Order and User_Order.
    - Order is the Entity that holds the information of a transaction which indicates the book is sold from someone to someone. An order can have a single User for selling, a single User for buying, and multiple Books. If no one buy the book, the seller_ID will be NULL to indicate that the order is not completed yet.
- AuthorList (will be implemented by Liu)
    - Author_ID: primary key, unique, int, auto_increment, not null.
    - Name: unique, varchar, not NULL.
    - Relationships: A M:M relationship between Author and Book. A 1:M relationship between Author and Book_Author.
    - Author is the Entity that holds the information of a single author. An Author can write many Books, and a Book can have many Authors.
- Book_Author (will be implemented by Liu)
    - Author_ID: int, not NULL, FK

    - Book_ID: int, not NULL, FK
    - Relationships: A 1:M relationship between Book and Book_Author, A 1:M relationship between Book_Author and Author.
    - This is a joint Entity that connects Book and Author. Connect two 1:M and compare to M:M.
- User_Order_Book (will be implemented by Liu)
    - User_ID: int, not NULL, FK
    - Order_ID: int, not NULL, FK
    - Book_ID, int, not NULL, FK
    - Relationships: A 1:M relationship between User and User_Order. A 1:1 relationship between User_Order and Order.
    - This is the Entity that will implement 1:M relationship between User and Order. A User can have many orders, but an order will only have one seller.
- Cart (will be implemented by Lin)
    - User_ID: int, not NULL, FK
    - Book_ID: int, not NULL, FK
    - Relationships: A 1:M relationship between User and Cart. A 1:M relationship between Book and Cart.
    - User can add as many books as they want into their Cart, but a single book can be added to many users' Cart. Also, A Cart can be empty, and a book doesn't necessarily need to be added to a Cart.

C. Entity-Relationship Diagram

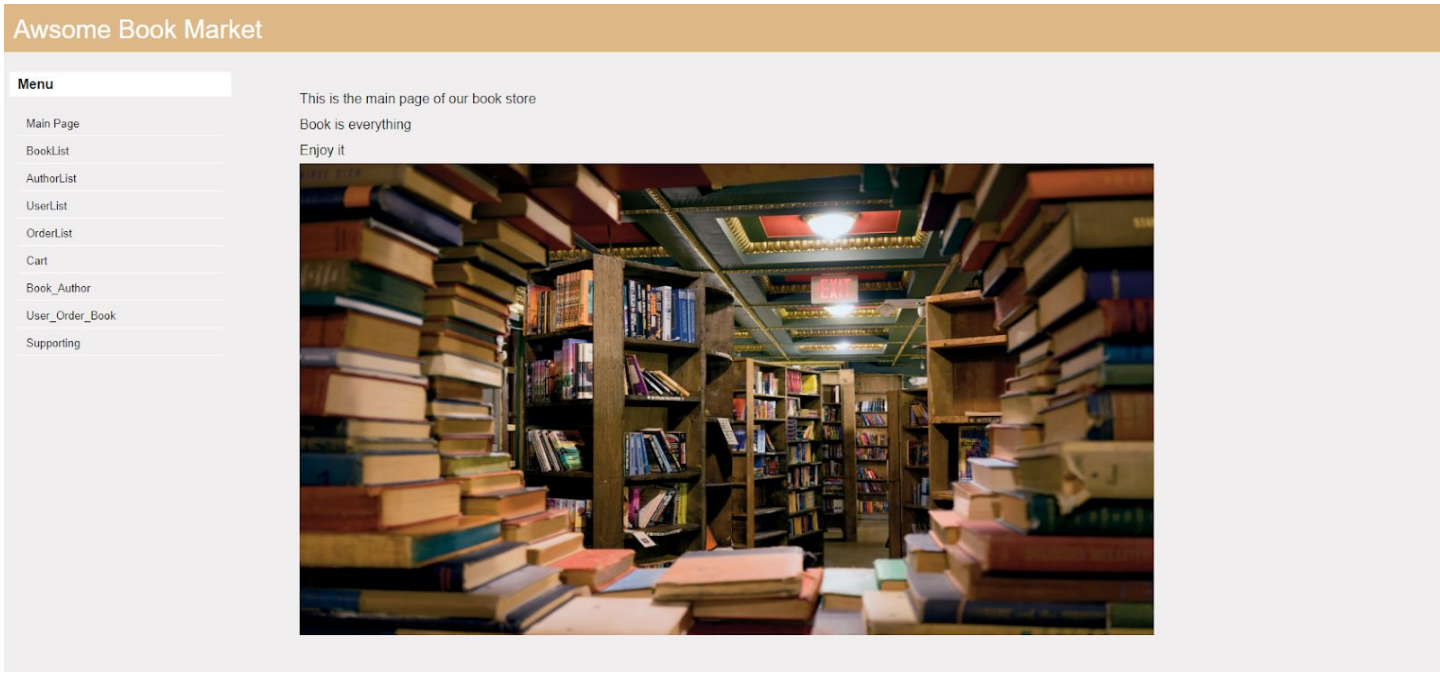


D. Schema

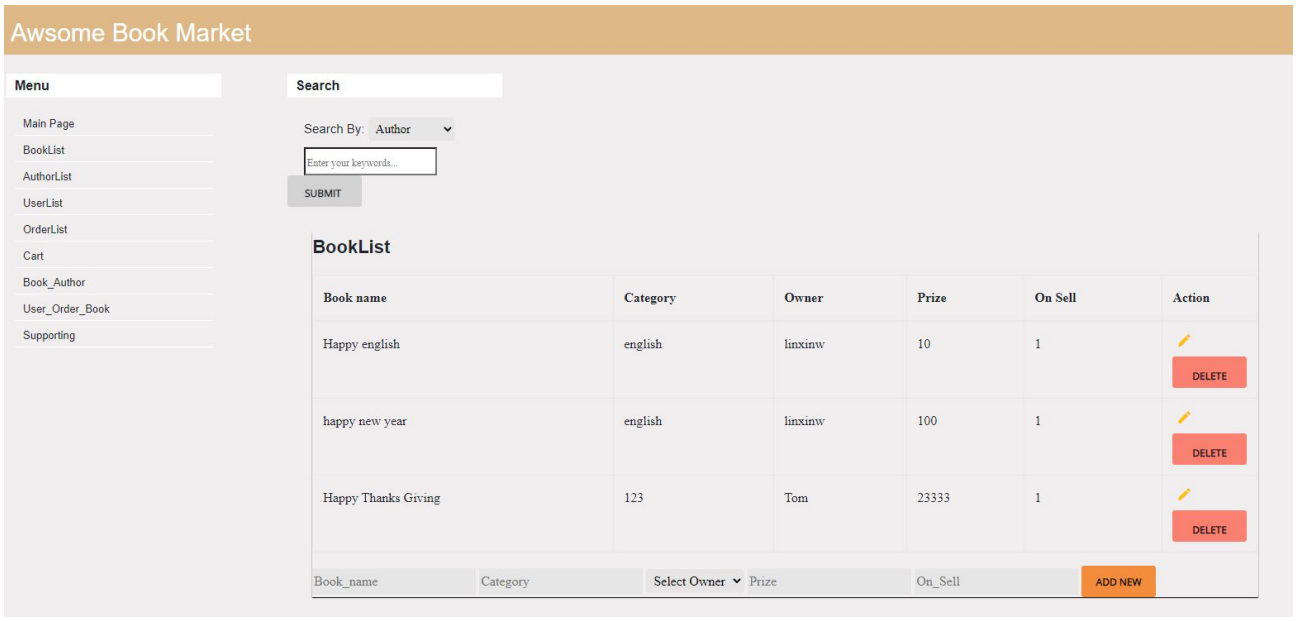


E. UI Screen Shots with Informative Titles

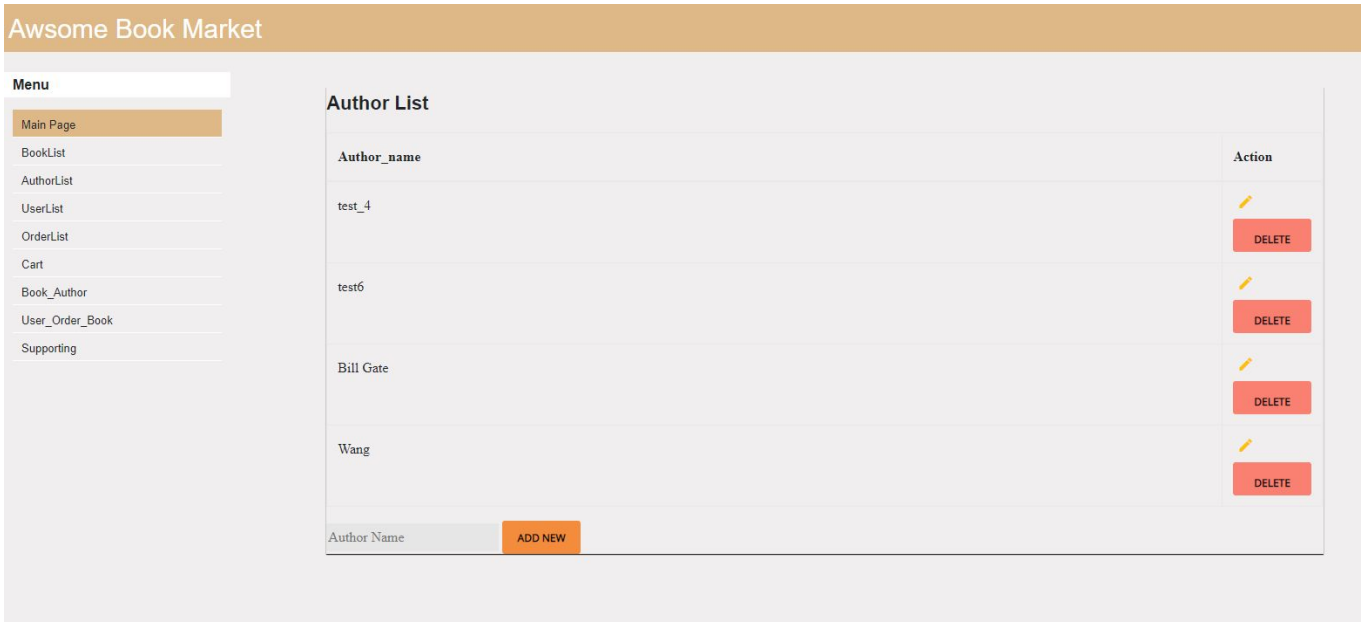
DISPLAY Main page



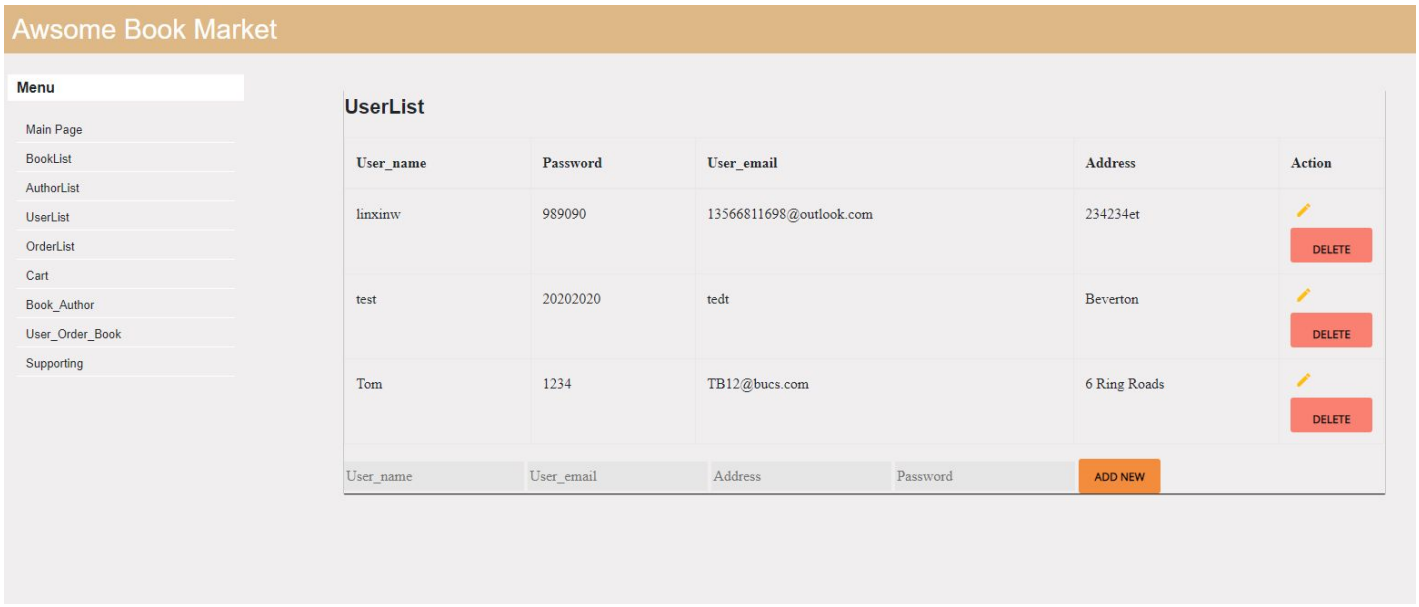
DISPLAY, INSERT, Delete, Filter, Edit Book List page



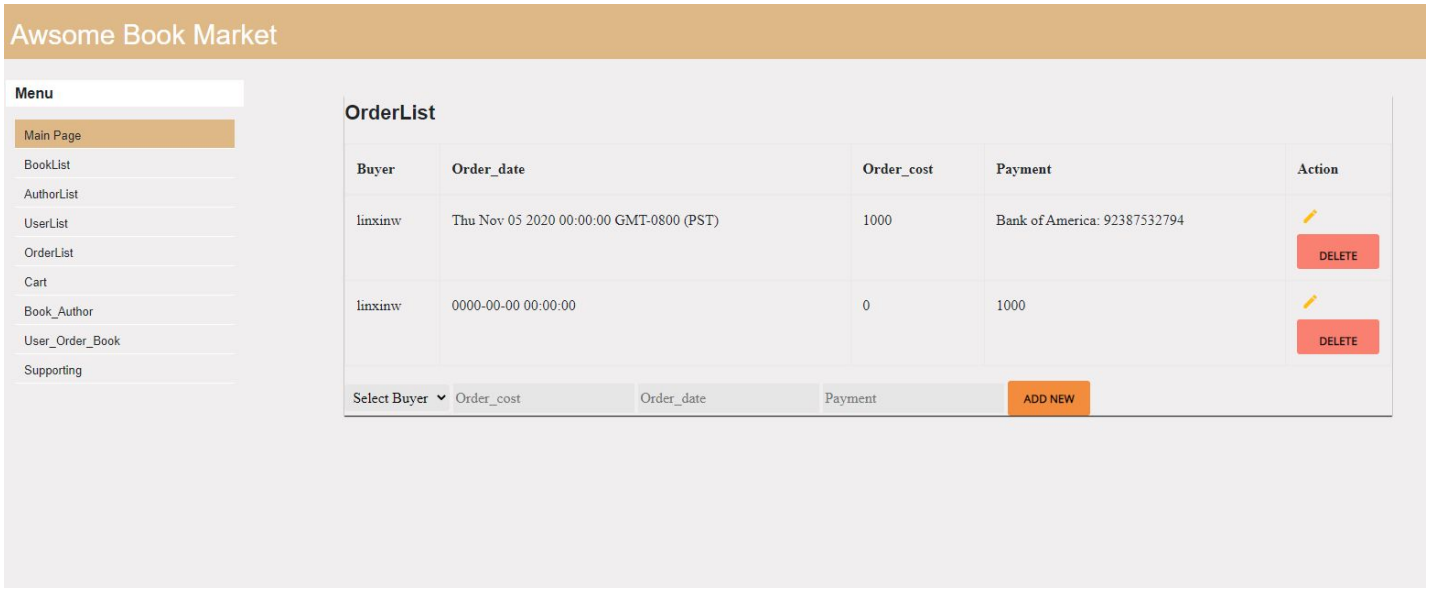
DISPLAY, INSERT, Delete, Edit Author List page



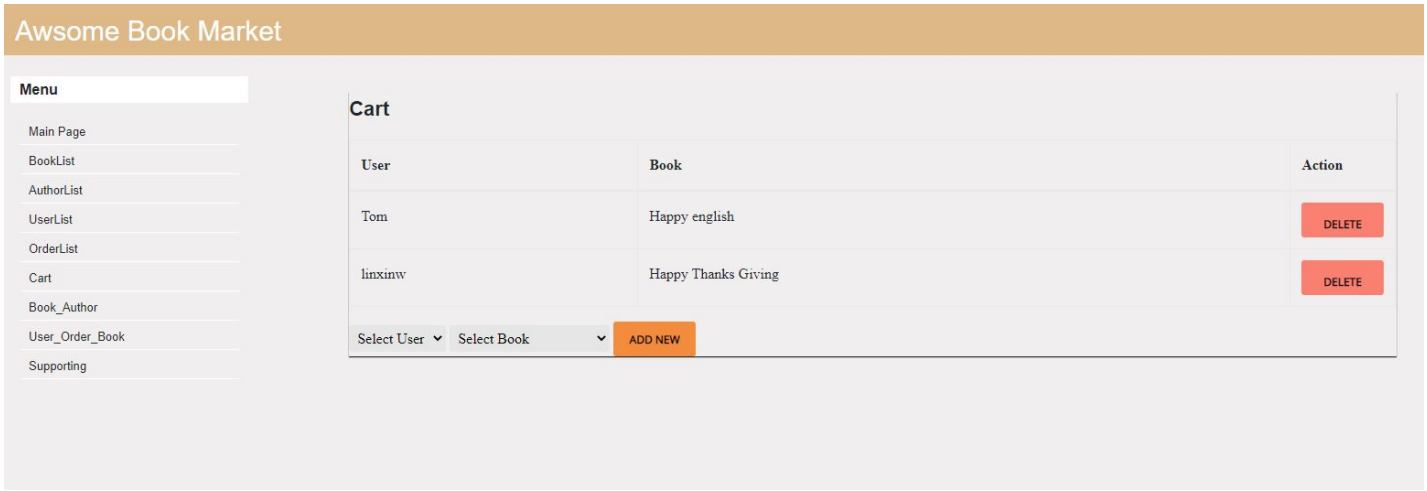
DISPLAY, INSERT, Delete, Edit User List page



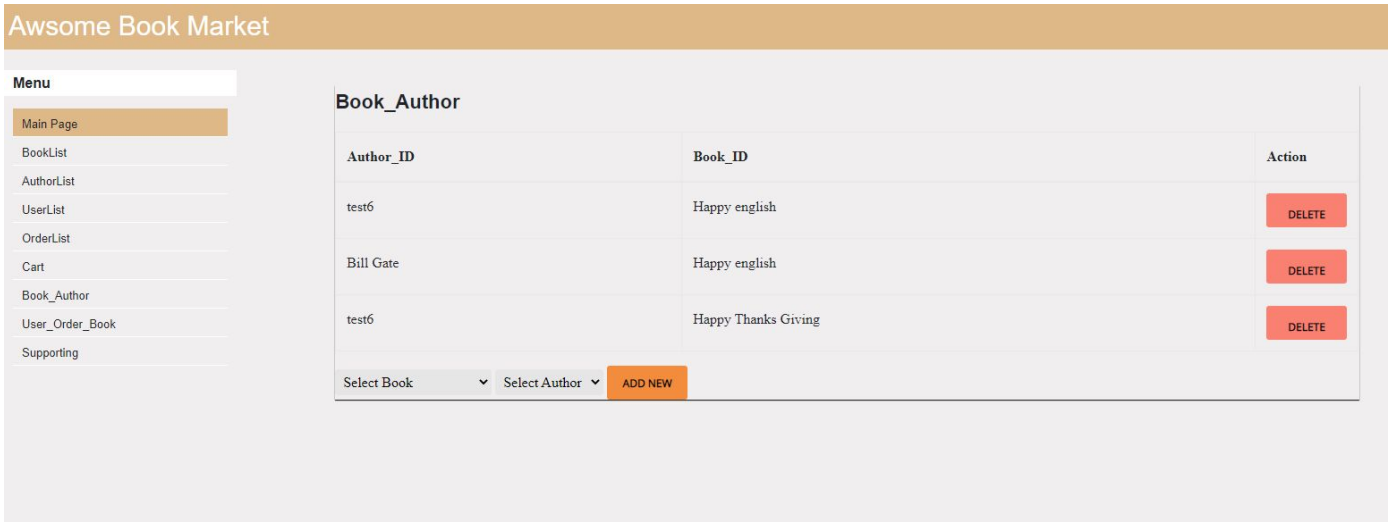
DISPLAY, INSERT, Delete, Edit Order List page



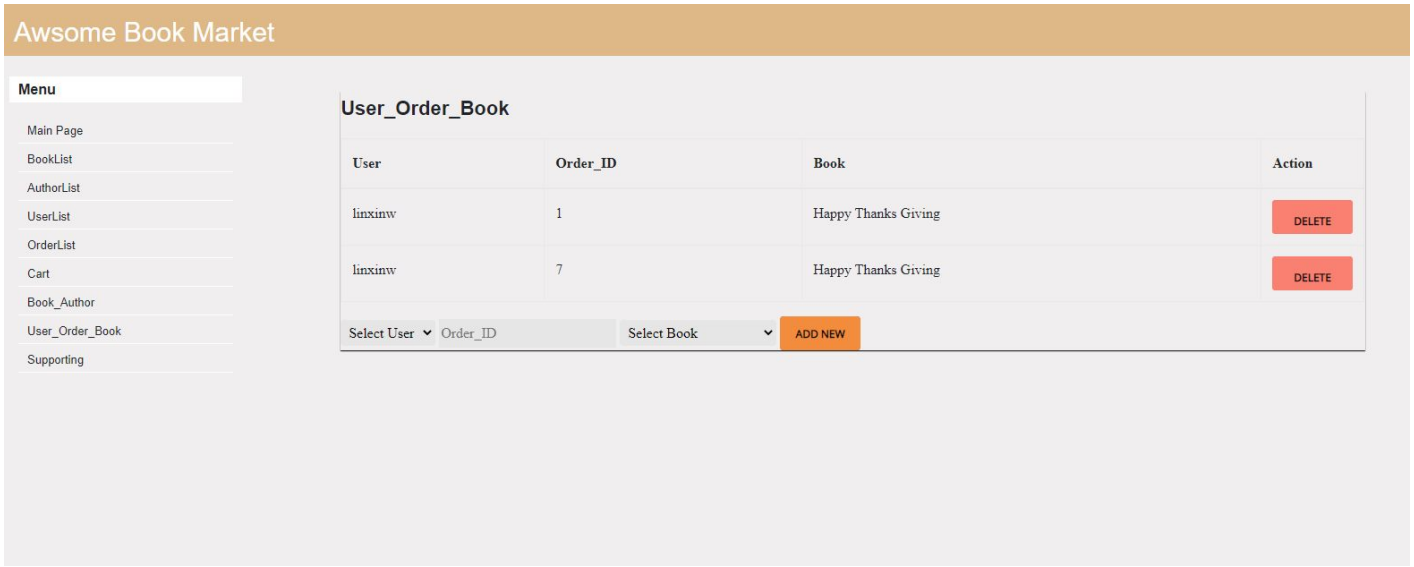
DISPLAY, INSERT, Delete Cart page



DISPLAY, INSERT, Delete Book Author page



DISPLAY, INSERT, Delete User Order Book page



DISPLAY Supporting page

Awsome Book Market

Menu

- Main Page
- BookList
- AuthorList
- UserList
- OrderList
- Cart
- Book_Author
- User_Order_Book
- Supporting

This is the Supporting page

You can get information about an order you placed on the website

Call xxx-xxx-xxx to get futher support