Feature Checklist

Team:

Book Browsing and Sorting

- □ Retrieve List of Books by Genre
    - ○ Logic: Given a specific genre, return a list of books for that genre.
    - ○ HTTP Request Type: GET
    - ○ Parameters Sent: Genre
    - ○ Response Data: JSON List of book objects
- □ Retrieve List of Top Sellers (Top 10 books that have sold the most copied)
    - ○ Logic: Return the top 10 books that have sold the most copies in descending order (most copies sold would be #1)
    - ○ HTTP Request Type: GET
    - ○ Parameters Sent: None
    - ○ Response Data : JSON List of book objects
- □ Retrieve List of Books for a particular rating and higher
    - ○ Logic: Filter by rating higher or equal to the passed rating value.
    - ○ HTTP Request Type: GET
    - ○ Parameters Sent: Rating
    - ○ Response Data: JSON List of book objects
- □ Discount books by publisher.
    - ○ Logic: Update the price of all books under a publisher by a discount percent.
    - ○ HTTP Request Type: PUT or PATCH
    - ○ Parameters Sent: Discount percent, Publisher
    - ○ Response Data: None

Each item is worth 10 points.

Total Achieved Functionality: ___ out of 4

Profile Management

- □ Create a User with username, password and optional fields (name, email address, home address)
    - ○ Logic: Provided the user fields, create the user in the database.
    - ○ HTTP Request Type: POST
    - ○ Parameters Sent: User Object
    - ○ Response Data: None

- □ Retrieve a User Object and its fields by their username
    - ○ Logic: Given a specific username, retrieve the user details.
    - ○ HTTP Request Type: GET
    - ○ Parameters Sent: Username
    - ○ Response Data: JSON User object.
- □ Update the user and any of their fields except for mail
    - ○ Logic: Given the username as a key lookup value and any other user field, update that user field with the new param value.
    - ○ HTTP Request Type: PUT / PATCH
    - ○ Parameters Sent: Username
    - ○ Response Data: None
- □ Create Credit Card that belongs to a User

    - Logic: Given a user name and credit card details, create a credit card for that user.
    - HTTP Request Type: POST
    - Parameters Sent: User name, Credit Card Object
    - Response Data: None

Each item is worth 10 points.

Total Achieved Functionality: ___ out of 4

Shopping Cart

- □ Retrieve the subtotal price of all items in the user’s shopping cart.
    - Logic: Give a user Id,return the subtotal of the books in the cart.
    - HTTP Request Type: GET
    - Parameters Sent: User Id
    - Response Data: Calculated Subtotal
- □ Add a book to the shopping cart.
    - Logic: Provided with a book Id and a User Id, add the book to the user’s shopping cart.
    - HTTP Request Type: POST
    - Parameters Sent: Book Id, User Id
    - Response Data: None
- □ Retrieve the list of book(s) in the user’s shopping cart.
    - Logic: Give a user Id, return a list of books that are in the shopping cart.
    - HTTP Request Type: GET
    - Parameters Sent: User Id
    - Response Data: List of Book Objects
- □ Delete a book from the shopping cart instance for that user.
    - Logic: Given a book If and a User Id, remove the book from the user’s shopping cart.
    - HTTP Request Type: DELETE
    - Parameters Sent: Book Id, User Id
    - Response Data: None

Each item is worth 10 points.

Total Achieved Functionality: ___ out of 4

Book Details

- □ An administrator must be able to create a book with the book ISBN, book name, book description, price, author, genre, publisher , year published and copies sold.
    - Logic: Given a Book’s info, add it to the system.
    - HTTP Request Type: POST
    - Parameters Sent: Book Object
    - Response Data: None
- □ Must be able retrieve a book’s details by the ISBN
    - Logic: Given a book id, retrieve the book information
    - HTTP Request Type: GET
    - Parameters Sent: Book Id
    - Response Data: Book object JSON
- □ An administrator must be able to create an author with first name, last name, biography and publisher
    - Logic: Given an Author’s Info, add it to the system.
    - HTTP Request Type: POST
    - Parameters Sent: Author Object

  - Response Data: None
- □ Must be able to retrieve a list of books associated with an author
  - Logic: Given an Author's Id, return the list of books for that author.
  - HTTP Request Type: GET
  - Parameters Sent: Author Id
  - Response Data: JSON list of Book Objects

Each item is worth 10 points.

Total Achieved Functionality: ___ out of 4

Book Rating and Commenting

- □ Must be able to create a rating for a book by a user on a 5 star scale with a datestamp
  - Logic: Create a rating for a book given by a user.
  - HTTP Request Type: POST
  - Parameters Sent: Rating, User Id, Book Id
  - Response Data: None
- □ Must be able to create a comment for a book by a user with a datestamp
  - Logic: Create a comment for a book given by a user.
  - HTTP Request Type: POST
  - Parameters Sent: Comment, User Id, Book Id
  - Response Data: None
- □ Must be able to retrieve a list of all comments for a particular book.
  - Logic: Retrieve a list of comments for the book
  - HTTP Request Type: GET
  - Parameters Sent: Book Id
  - Response Data: JSON list of comments
- □ Must be able to retrieve the average rating for a book
  - Logic: Given a book Id, calculate the average rating as a decimal.
  - HTTP Request Type: GET
  - Parameters Sent: Book Id
  - Response Data: Computed Average rating (decimal)

Each item is worth 10 points.

Total Achieved Functionality: ___ out of 4

Wish List Management

- □ Must be able to create a wishlist of books that belongs to user and has a unique name
  - Logic: Given a user Id and a wish list name, create the wishlist.
  - HTTP Request Type: POST
  - Parameters Sent: Wish list name, User Id
  - Response Data: None
- □ Must be able to add a book to a user's wishlist
  - Logic: Given a book Id and a wish list Id, add the book to that wish list.
  - HTTP Request Type: POST
  - Parameters Sent: Book Id, Wishlist Id
  - Response Data: None
- □ Must be able to remove a book from a user's wishlist into the user's shopping cart
  - Logic: : Given a book Id and a wish list Id, remove the book to that wish list.

- HTTP Request Type: DELETE
- Parameters Sent: Book Id, Wishlist Id
- Response Data: None

- □ Must be able to list the book’s in a user’s wishlist
  - Logic: Given a wishlist Id, return a list of the books in that wishlist.
  - HTTP Request Type: GET
  - Parameters Sent: Wishlist Id
  - Response Data: JSON LIST of books in the user’s wishlist.

Each item is worth 10 points.

Total Achieved Functionality: ___ out of 4