

**Work Integrated Learning Programmer**  
**M.Tech Software Engineering**



**FULL STACK APPLICATION DEVELOPMENT**

**Submitted by**

**Raghunath Reddy G**

**2023MT93051**

**2023mt93051@wilp.bits-pilani.ac.in**

## Contents

<b>1. Problem statement: BOOK EXCHANGE PLATFORM</b>	<b>3</b>
<b>2. Key Features:</b>	<b>3</b>
<b>3. Architecture design:</b>	<b>3</b>
<b>4. Each Micro service is internal design shown below (user service, Book services)</b>	<b>4</b>
<b>5. Tech-stack used:</b>	<b>5</b>
<b>6. Flow diagram:</b>	<b>5</b>
<b>7. Database Design:</b>	<b>6</b>
<b>8. MS-SQL Database design table details</b>	<b>6</b>
<b>9. Backend API implementation endpoint.</b>	<b>7</b>
<b>9.1 Swagger screen of the “user service”</b>	<b>7</b>
<b>9.2 Backend API Book Management Service endpoints</b>	<b>8</b>
<b>9.3 Below are the books API Request /response example</b>	<b>10</b>
<b>10 .Frontend React application screen shot:-</b>	<b>11</b>
<b>10.1 Code structure:</b>	<b>11</b>
<b>11. Application working Screenshots</b>	<b>11</b>
<b>12. GitHub Repositories (<a href="https://github.com/raghunathreddy/BITS-Software-Engineering/tree/main/3rdsem/fullstack-development">https://github.com/raghunathreddy/BITS-Software-Engineering/tree/main/3rdsem/fullstack-development</a>)</b>	<b>16</b>
<b>12.1. github Backend services:-</b>	<b>16</b>
<b>12.2 github Frontend React application:-</b>	<b>16</b>
<b>13. Demonstration Video</b>	<b>16</b>
<b>13.1 Demo video uploaded in YouTube</b>	<b>16</b>

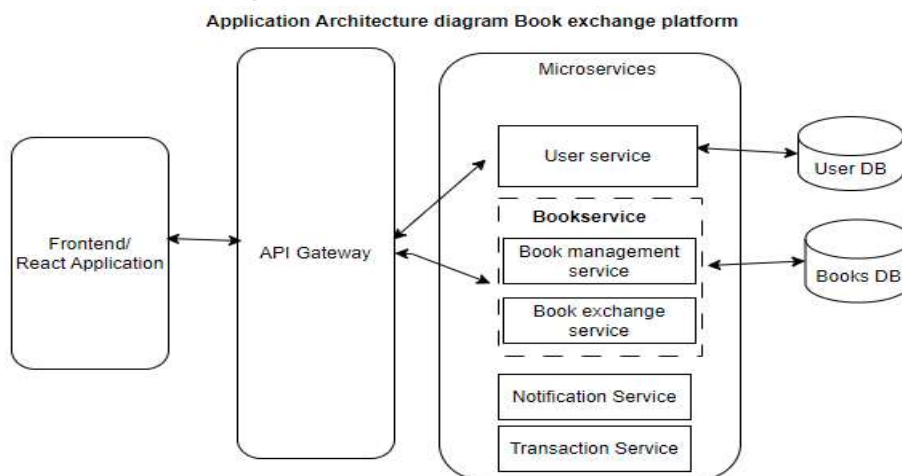
## 1. Problem statement: BOOK EXCHANGE PLATFORM

Book lovers frequently accumulate a collection of books they have read and look for other recommendations. They are always eager to explore new reading material. Traditional methods of exchanging books, such as local book swaps or lending among friends, are limited in scope and accessibility. Therefore, it is imperative to have a digital platform that can facilitate book exchanges on a larger scale. This platform should connect users with similar reading interests, enabling them to trade books easily and efficiently. This project aims to develop a full-stack web application that serves as a centralized platform for users to exchange, lend, and borrow books with other users. The platform should provide a user-friendly interface, robust search and recommendation features, and secure transaction capabilities.

## 2. Key Features:

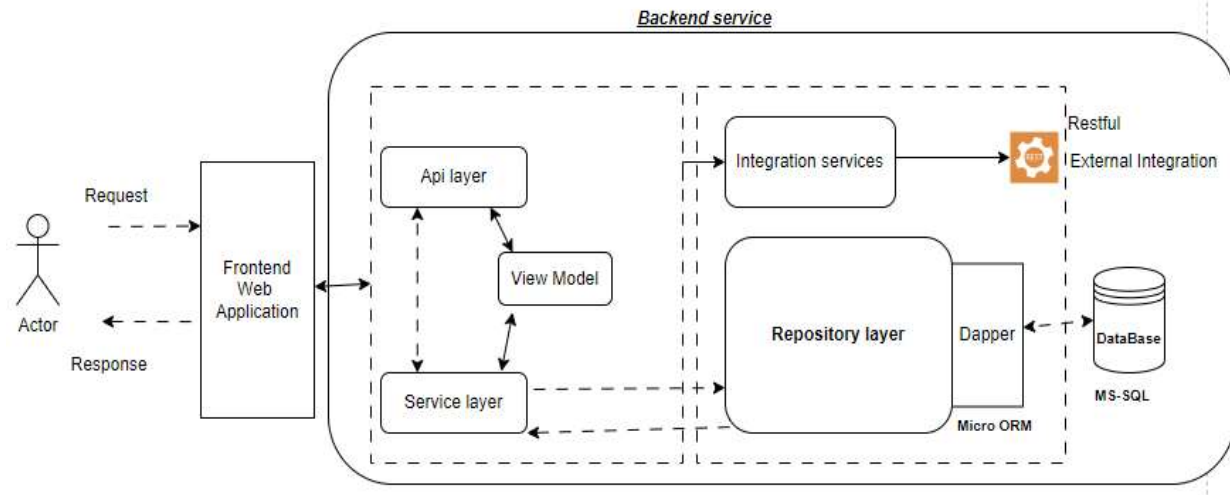
- **User Authentication:** Implement a secure user authentication system to allow users to register, log in, and manage their accounts.
- **Book Listing:** Enable users to list books they want to exchange or lend, including details such as title, author, genre, condition, and availability status.
- **Book Search:** Provide users with advanced search and filtering options to discover books based on criteria like genre, author, title, location, and availability.
- **Exchange Requests:** Allow users to send and receive exchange requests for specific books, including negotiation options for terms such as delivery method and duration.
- **Messaging System:** Implement a messaging system to facilitate user communication regarding book exchanges, including negotiation details, logistics, and scheduling. (Mock API s can be used)
- **User Profiles:** Enable users to create profiles with information about their reading preferences, favorite genres, and books they currently own or wish to acquire.
- **Transaction Management:** Provide tools for users to track the status of their exchange transactions, including pending requests, accepted exchanges, and completed transactions.

## 3. Architecture design:

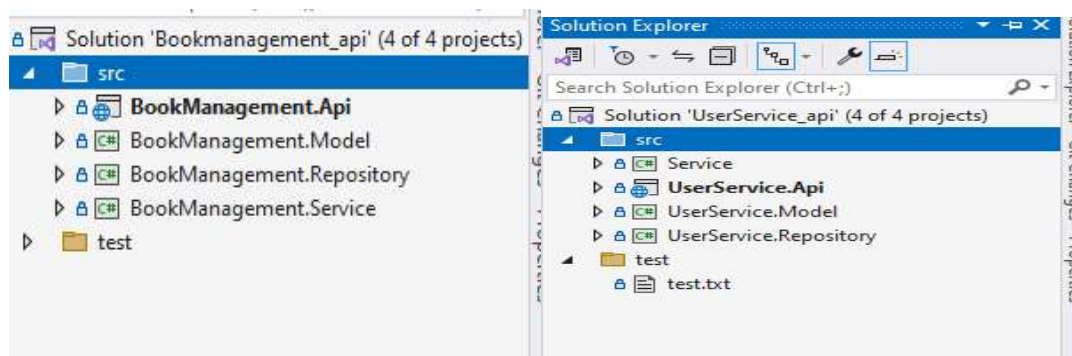


#### 4. Each Micro service is internal design shown below (user service, Book services)

Microservice is internal design shown below (user service, Bookservices)



- Backend api code snippet in n-layer architecture



The backend service is designed using the **microservice** each micro-service is independently developed & deployed with frequently add the feature in each iteration and deploy to production.

These microservices are *designed based on **business domain context & isolated within the boulder***. Each services is communicated via messaging or direct service to service call using restful template.

- **API layer:** will have the Http-Methods like Http GET,POST PUT,DELETE which is exposed as via swagger to consume
- **Service Layer:-** will contain the business logic implemented w.r.t to domain. And also it bind the data from DTO-models to Domain-Data-models.
- **Repository layer:** will have DB connection logic & all DB query and Integrated to Dapper-ORM which will connect to database and load the data to data models.
- **Model layer:** will have the POCO classes or Domain model classes which is replica of the database table which is used to bind the data from table.

## 5. Tech-stack used:

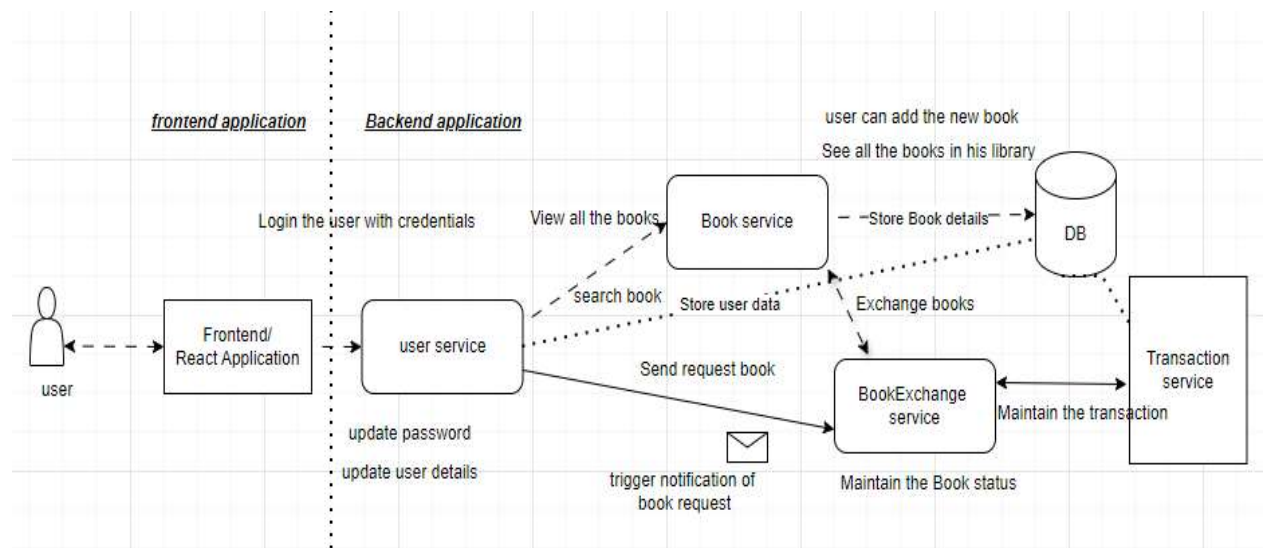
### Back-End stack: -

- **.NET core:** *.Net core restful API* service is user which is used to build the back business logic and expose it as restful service it will connect to database and pull the data using sql queries.
- **MS SQL** : is used as database to store the application user data & book exchange data , this is relational database stores the data in form of rows & columns'
- **Dapper**: it is **MICRO** ORM which connect the repository & Data model and binds the data from SQL DB to Data model.

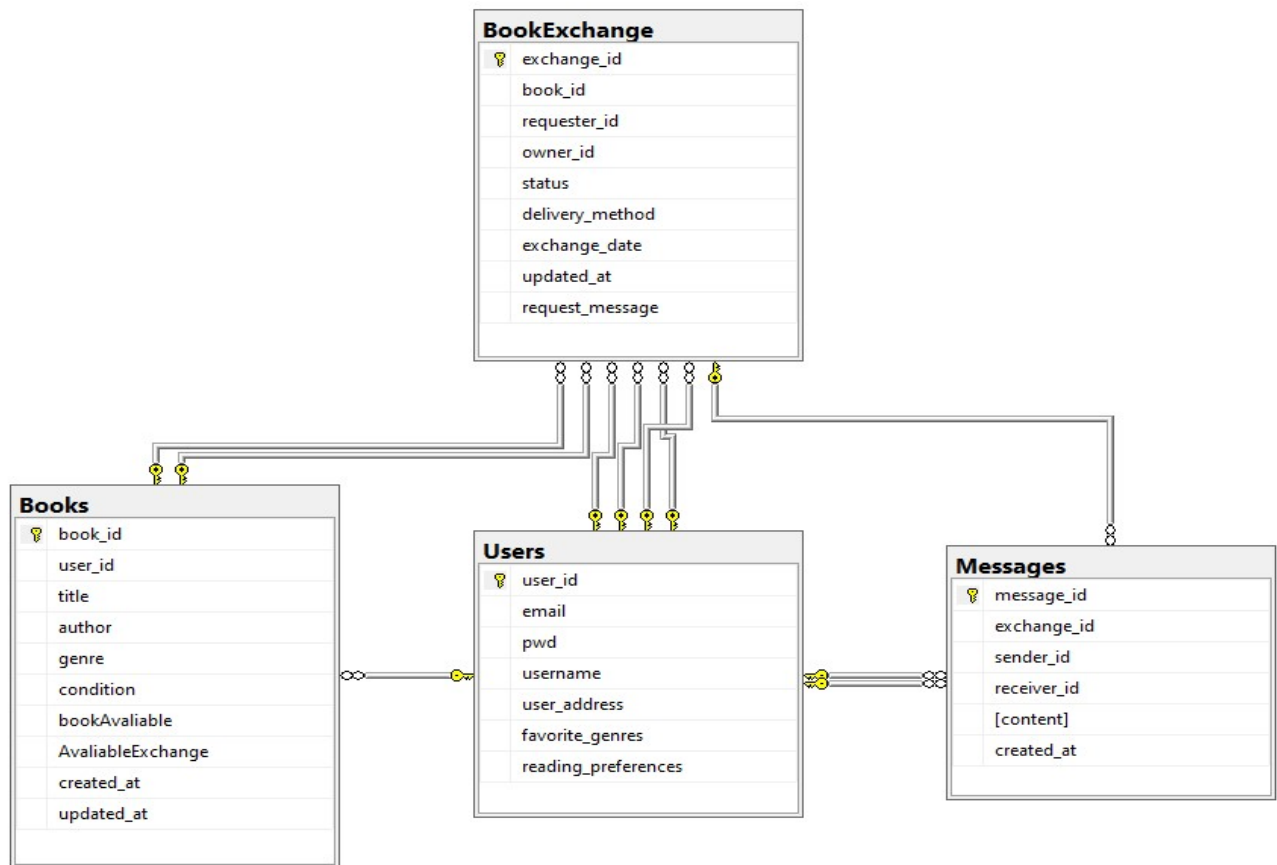
### Front-End stack:

- **React Web application:** - React is used to build the interactive web application with a user friendly. It's as responsive UI which uses a *bootstrap .css styling*

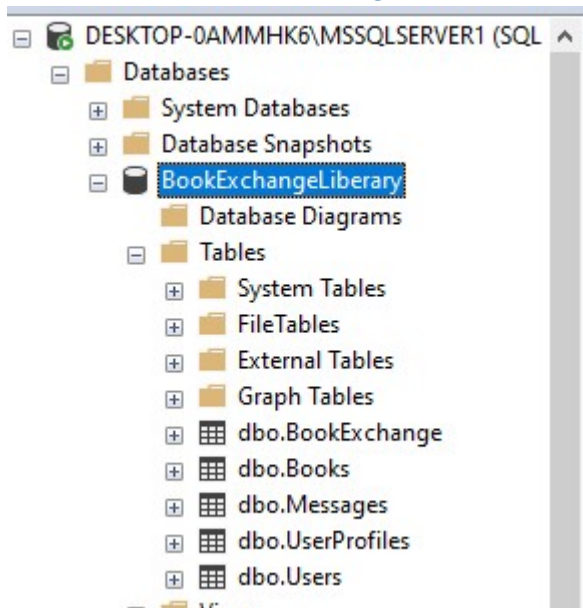
## 6. Flow diagram:



## 7. Database Design:



## 8. MS-SQL Database design table details



## 9. Backend API implementation endpoint.

- **User Story 1: User Authentication**

As a user, I want to securely register, log in, and manage my account, So that I can access and use the book exchange platform.

Acceptance Criteria:

The platform must allow users to register with a valid email and password.

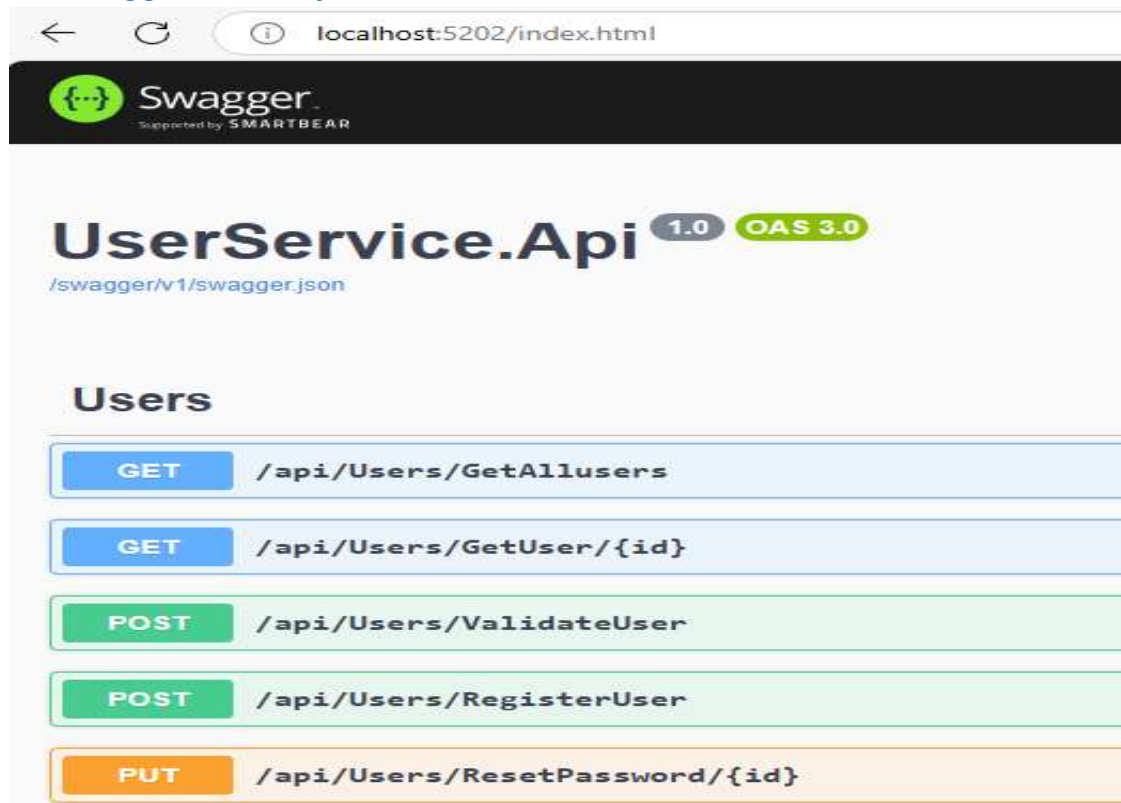
Passwords must be stored securely using encryption.

Users should be able to reset their password via a password recovery system.

Users should be able to log out from their account.

Entity	Http Method	URI	Description
Users	GET	/api/Users/GetAllusers	Get all the users list
	GET	/api/Users/GetAllusers/\${userid}	Search the User by ID
	POST	/api/Users/RegisterUser	Add the new User
	PUT	/api/users/resetPassword/ \${id}	Update the user password w.r.t userID
	DELETE	/api/users/\${id}	Deleted the user

### 9.1 Swagger screen of the “user service”





## 9.2 Backend API Book Management Service endpoints

- **User Story 2: Book Listing**

As a user, I want to list books that I want to exchange or lend, So that others can browse and request the books I offer.

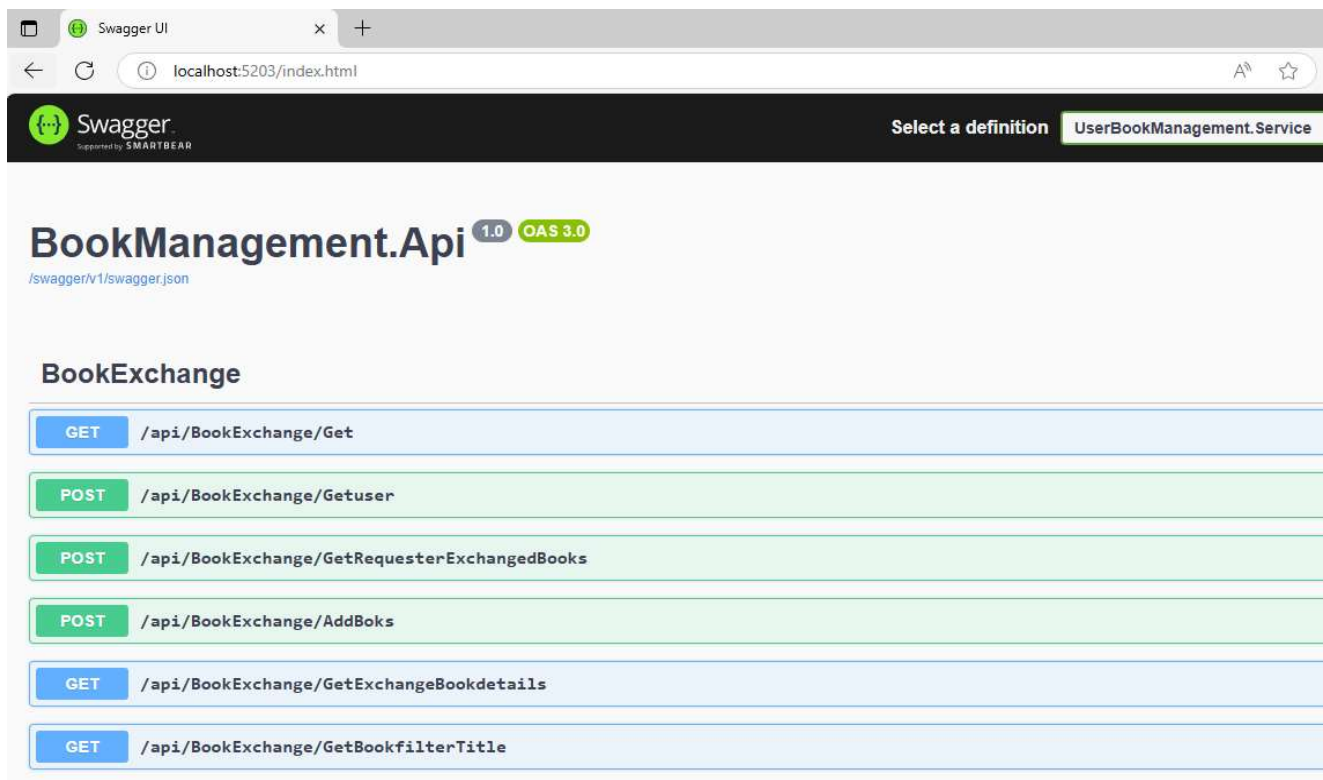
Acceptance Criteria:

Users should be able to add a book to their list by providing details such as title, author, genre, condition, and availability status.

Each book listing must have a unique ID associated with a user's profile.

Users should be able to edit or delete book listings at any time.

The book listing must be displayed in the user's profile and searchable by others.



Entity	Http Method	URI	Description
ExchangeBook Service	GET	/api/BookExchange/Get	Get all the Books for Exchange
	POST	/api/BookExchange/GetUser	Search the Books available w.r.t UserID
	POST	/api/ BookExchange /AddBooks	Add the New books for available for Exchange
	GET	/api/ BookExchange /GetBookFilterByTitle	Search the Books available for Exchange w.r.t Title
	POST	/api/ BookExchange /GetRequestExchangeBooks	Raise a request for books exchange



### User Story 3: Book Search

As a user, I want to search for books based on criteria such as title, author, genre, and location, Acceptance Criteria:

The platform must provide a search bar where users can enter keywords like title, author, or genre.

The platform should allow users to filter search results by availability status, genre, and location.

Users must be able to view detailed information about a book (title, author, condition, etc.) when clicking on a search result. The search results should be paginated or load incrementally to handle large datasets.

Books			
GET	/api/Books/Get		
POST	/api/Books/Getuser		
POST	/api/Books/SearchBook		
POST	/api/Books/AddBoks		

Entity	Http Method	URI	Description
Books Service	GET	/api/Books/Get	Get all the Books in library /store
	POST	/api/ Books /GetUser	Search the Books available w.r.t UserID
	POST	/api/ Books /SearchBook	Search book w.r.t title , gener, author
	POST	/api/ Books /AddBooks	Add the book in this list

### Backend code structure

```
8 namespace BookManagement.Api.Controllers
9 {
10     public class BookExchangeController : ControllerBase
11     {
12         public void AddBoks([FromBody] DtoBookExchange addExchangebook)
13         {
14             _bookExchangeService.AddExchangedBooks(addExchangebook);
15         }
16
17         [HttpGet]
18         public List<DtoBookExchangeTx> GetExchangeBookdetails()
19         {
20             return _bookExchangeService.GetAllExchangedBooksTrx();
21             //return new string[] { "value1", "value2" };
22         }
23
24         [HttpGet]
25         public List<DtoBookExchangeTx> GetBookFilterTitle(string? title)
26         {
27             var resultbooks= _bookExchangeService.GetAllExchangedBooksTrx();
28             return resultbooks.Where(x => x.title.Contains(title)).ToList();
29             //return new string[] { "value1", "value2" };
30         }
31     }
32 }
```

The Solution Explorer on the right shows the project structure for 'BookManagement.Api' (4 of 4 projects). It includes folders for src, test, and various files like appsettings.json, Program.cs, and interfaces.

### 9.3 Below are the books API Request /response example

localhost:5203/index.html

#### Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5203/api/Books/Getuser?user_id=1' \
  -H 'accept: text/plain' \
  -d ''
```

Request URL

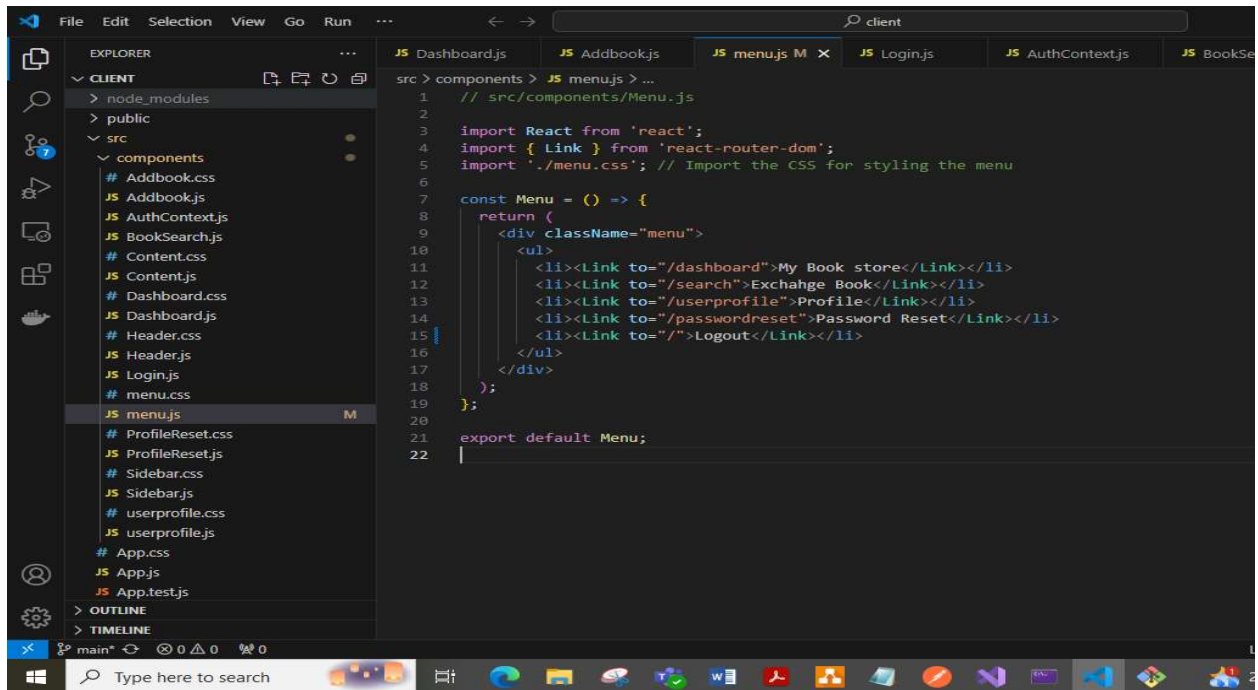
```
http://localhost:5203/api/Books/Getuser?user_id=1
```

Server response

Code	Details
200	<p>Response body</p> <pre>[   {     "book_id": 2,     "user_id": 1,     "title": "HereThere and Everywhere",     "author": "Sudha Murty",     "genre": "Motivation",     "condition": "New",     "bookAvaliable": "Available",     "avaliableExchange": "yes",     "created_at": "2024-10-31T00:00:00",     "updated_at": "2024-10-31T00:00:00"   },   {     "book_id": 5,     "user_id": 1,     "title": "The Power of Your Subconscious Mind",     "author": "Joseph Murphy",     "genre": "Spritual",     "condition": "Good",     "bookAvaliable": "Available",     "avaliableExchange": "yes",     "created_at": "2024-10-25T16:40:16.46",     "updated_at": "2024-10-31T16:40:16.46"   },   {     "book_id": 1,     "user_id": 1,     "title": "The Power of Your Subconscious Mind",     "author": "Joseph Murphy",     "genre": "Spritual",     "condition": "Good",     "bookAvaliable": "Available",     "avaliableExchange": "yes",     "created_at": "2024-10-25T16:40:16.46",     "updated_at": "2024-10-31T16:40:16.46"   } ]</pre>

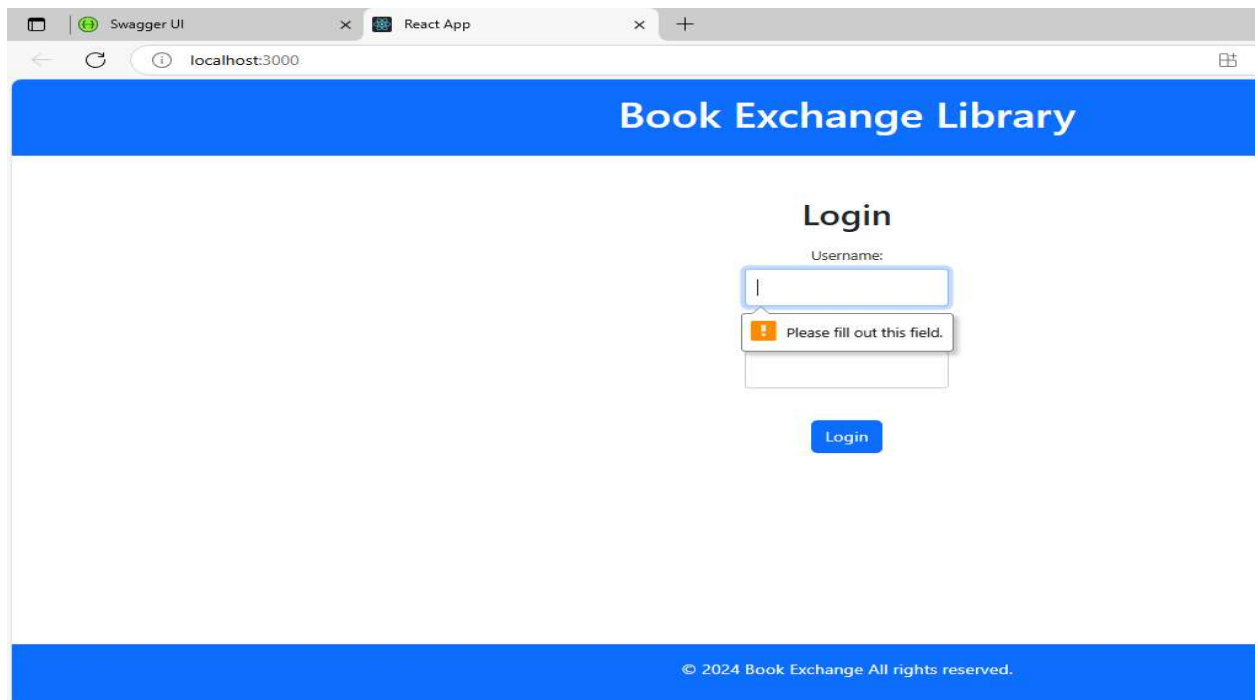
## 10. Frontend React application screen shot:-

### 10.1 Code structure:

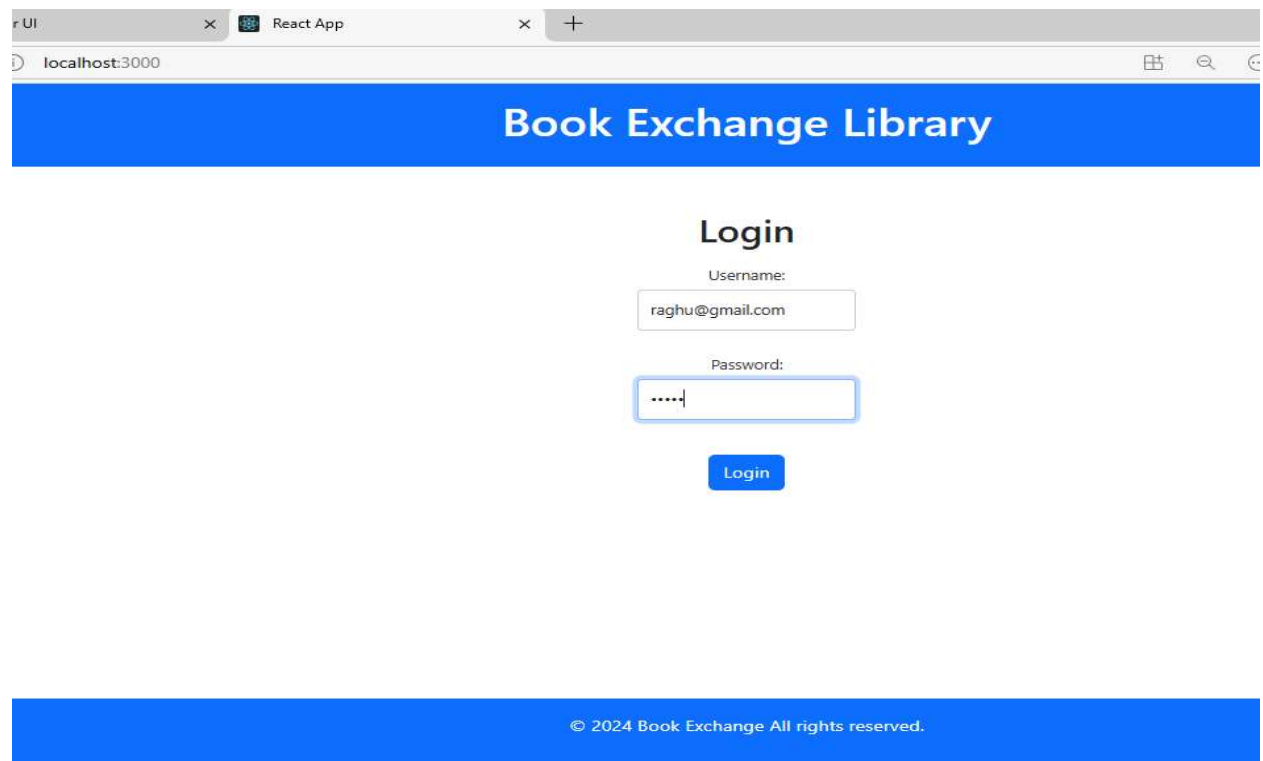


## 11. Application working Screenshots

### 11.1: Login screen:- login with validation

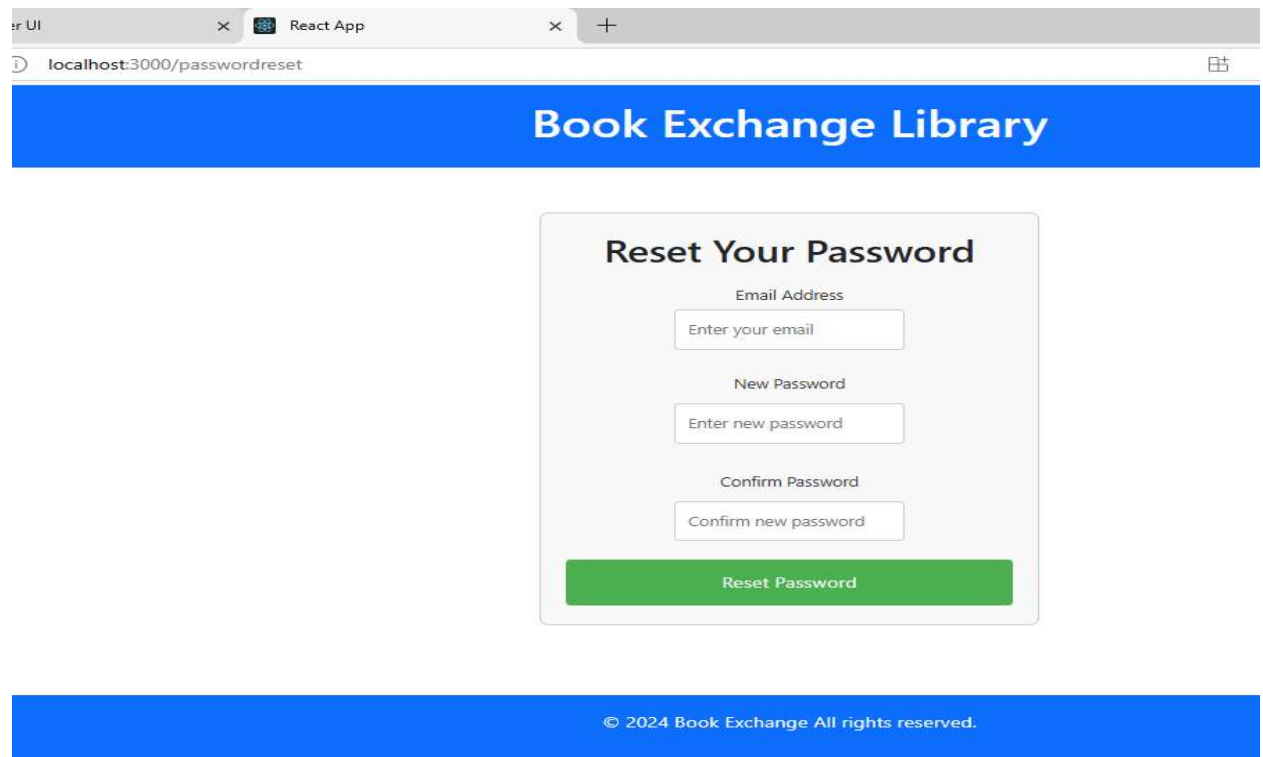


### 11.2 Login screen: - login with username & password



The screenshot shows a web browser window with two tabs: 'r UI' and 'React App'. The address bar displays 'localhost:3000'. The page features a blue header with the text 'Book Exchange Library'. Below the header, the title 'Login' is centered. Underneath, there are two input fields: 'Username:' with the value 'raghu@gmail.com' and 'Password:' with masked characters '....'. A blue 'Login' button is positioned below the password field. At the bottom of the page, a blue footer contains the text '© 2024 Book Exchange All rights reserved.'

### 11.3 Password Reset:



The screenshot shows a web browser window with two tabs: 'r UI' and 'React App'. The address bar displays 'localhost:3000/passwordreset'. The page features a blue header with the text 'Book Exchange Library'. Below the header, a light gray box contains the title 'Reset Your Password'. Inside this box, there are three input fields: 'Email Address' with the placeholder 'Enter your email', 'New Password' with the placeholder 'Enter new password', and 'Confirm Password' with the placeholder 'Confirm new password'. A green 'Reset Password' button is located at the bottom of the gray box. At the bottom of the page, a blue footer contains the text '© 2024 Book Exchange All rights reserved.'

### 11.4 Password Reset: with validation

localhost:3000/passwordreset

## Book Exchange Library

### Reset Your Password

Email Address

raghu@gmail.com

New Password

....

Confirm Password

.....

Passwords do not match.

Reset Password

© 2024 Book Exchange All rights reserved.

### 11.5 Dashboard: once login it will show all the books collection in his library/store

localhost:3000/dashboard

## Book Exchange Library

My Book store

Exchahge Book

Profile

Password Reset

Logout

### My Books collection

Add\_NewBook

Book_ID	Book Title	Author	Genre	BookAvailable
2	HereThere and Everywhere	Sudha Murty	Motivation	Available
5	The Power of Your Subconscious Mind	Joseph Murphy	Spritual	Available
7	2 States	Chetan Bhagat	Novel	Available
1007	Fairy tales	Hans Christian	Novel	Available
1009	The Divine Comedy	Dante A	Spritual	Available

© 2024 Book Exchange All rights reserved.

### 11.6 Add the new book in the collection list:

Swagger UI React App

localhost:3000/addbook

## Book Exchange Library

[My Book store](#)  
[Exchahge Book](#)  
[Profile](#)  
[Password Reset](#)  
[Logout](#)

### Add New Book

Book Title:

Author:

Genre:

BookAvaiable:

condition:

[Add Book](#)

© 2024 Book Exchange All rights reserved.

### 11.7 List all the Books available for Exchanges with other

localhost:3000/search

## Book Exchange Library

[My Book store](#)  
[Exchahge Book](#)  
[Profile](#)  
[Password Reset](#)  
[Logout](#)

### Books Availible for Exchange

Search for books [Search](#)

Book_ID	Book_Owner	Title	Book_exchanged_To	Requested_status	Request_message	delivery_method	Exchange_date	Raise Request
2	Raghunath G	HereThere and Everywhere		Availible		courier		<a href="#">Request_Book</a>
5	Raghunath G	The Power of Your Subconscious Mind	Shankar	Exchanged	Need this book for week	courier	2024-11-14T16:00:41.28	<a href="#">Request_Book</a>
1007	Raghunath G	Fairy tales	Mark J	Exchanged	Mutual exchanged	Hand Exchanged	2024-11-13T16:00:41.28	<a href="#">Request_Book</a>

© 2024 Book Exchange All rights reserved.

### 11.8 Search the book with title shown below:

The screenshot shows a web browser at localhost:3000/search. The page has a blue header with 'Book Exchange Library' and a blue sidebar with links: 'My Book store', 'Exchange Book', 'Profile', 'Password Reset', and 'Logout'. The main content area is titled 'Books Available for Exchange'. It features a search bar with 'Fairy tales' and a 'Search' button. Below the search bar is a table with the following data:

Book_ID	Book_Owner	Title	Book_exchanged_To	Requested_status	Request_message	delivery_method	Exchange_date	Raise Request
1007	Raghunath G	Fairy tales	Mark J	Exchanged	Mutual exchanged	Hand Exchanged	2024-11-13T16:00:41.28	<button>Request_Book</button>

At the bottom of the page, there is a footer: '© 2024 Book Exchange All rights reserved.'

### 11.9. LoggedIn user profile:

The screenshot shows a web browser at localhost:3000/userprofile. The page has a blue header with 'Book Exchange Library' and a blue sidebar with links: 'My Book store', 'Exchange Book', 'Profile', 'Password Reset', and 'Logout'. The main content area displays the 'User Profile' for a user with ID 1. The profile information is as follows:

**User Profile**  
**User ID:** 1  
**Name:** Raghunath G  
**Email:** raghu@gmail.com  
**Favorite\_genres:** Novel,Friction  
**Reading Preferences:** Novel

At the bottom of the page, there is a footer: '© 2024 Book Exchange All rights reserved.'



## 12. GitHub Repositories (<https://github.com/raghunathreddy/BITS-Software-Engineering/tree/main/3rdsem/fullstack-development>)

### 12.1. github Backend services:-

- **UserService API :-**  
[https://github.com/raghunathreddy/BITS-Software-Engineering/tree/main/3rdsem/fullstack-development/UserService\\_api](https://github.com/raghunathreddy/BITS-Software-Engineering/tree/main/3rdsem/fullstack-development/UserService_api)
- **BookService API:-**  
[https://github.com/raghunathreddy/BITS-Software-Engineering/tree/main/3rdsem/fullstack-development/Bookmanagement\\_api](https://github.com/raghunathreddy/BITS-Software-Engineering/tree/main/3rdsem/fullstack-development/Bookmanagement_api)

### 12.2 github Frontend React application:-

- **React Application:-** <https://github.com/raghunathreddy/BITS-Software-Engineering/tree/main/3rdsem/fullstack-development/frontend/book-exchange-library/client>

## 13. Demonstration Video

- [https://github.com/raghunathreddy/BITS-Software-Engineering/blob/main/3rdsem/fullstack-development/document/FSAD\\_Raghunath\\_2023MT93051.mp4](https://github.com/raghunathreddy/BITS-Software-Engineering/blob/main/3rdsem/fullstack-development/document/FSAD_Raghunath_2023MT93051.mp4)

### 13.1 Demo video uploaded in YouTube

[https://www.youtube.com/watch?v=rNr2a8\\_wAAI](https://www.youtube.com/watch?v=rNr2a8_wAAI)

## 14. Documentation & Architecture diagram:-

- <https://github.com/raghunathreddy/BITS-Software-Engineering/tree/main/3rdsem/fullstack-development/document>