

Team name: Thu1030-Dipto-Group6
Team number: 6

Product Owner: Dipto Pratyaksa

Team members:

Scrum Master: Kim Ta

Testing: Megan Dalton

Development team: Declan Baker, Dineth Abeysinghe

New code/test files:

- `ordermicroservices` was added to BackEnd containing the following files and directories:

```
✓ ordermicroservices
  ✓ .mvn / wrapper
    maven-wrapper.jar
    maven-wrapper.properties
    MavenWrapperDownloader.java
  ✓ src
    ✓ main
      ✓ java / com / sept / orders
        ✓ config
          WebConfig.java
        ✓ controllers
          OrdersController.java
        ✓ daos
          OrdersDao.java
        ✓ mapper
          OrderMapper.java
        ✓ models
          Order.java
          TokenBody.java
          OrdersApplication.java
      ✓ resources
        ✓ static
        ✓ templates
          application.properties
    > test
    > target
    .gitignore
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
    system.properties
```

- The following new files and directories were added to bookmicroservices:
 - src/main/java/com/bookeroo/app/models/Review.java
 - src/main/java/com/bookeroo/app/services/
 - src/main/java/com/bookeroo/app/services/ImageUploadService.java
 - src/main/java/com/bookeroo/mapper/ReviewMapper.java

```

    <bookmicroservices>
        > .mvn
        <src>
            <main>
                <java / com / bookeroo>
                    <app>
                        <config>
                            J WebConfig.java
                        <controllers>
                            J BooksController.java
                        <daos>
                            J BooksDao.java
                        <models>
                            J Book.java
                            J Review.java
                        <services>
                            J ImageUploadService.java
                            J AppApplication.java
                        <mapper>
                            J BookMapper.java
                            J ReviewMapper.java
                        <resources>
                            J application.properties
                    <test / java / com / bookeroo / app>
                        J AppApplicationTests.java
                > target
                <.gitignore>
                <Dockerfile>
                mvnw
                mvnw.cmd
                pom.xml
                system.properties
  
```

- The following new files and directories were added to loginmicroservices:
 - src/main/java/com/rmit/sept/bk_loginmicroservices/mapper/SellerMapper.java
 - src/main/java/com/rmit/sept/bk_loginmicroservices/model/Seller.java

- src/main/java/com/rmit/sept/bk_loginmicroservices/payload/RequestResponse.java
- src/main/java/com/rmit/sept/bk_loginmicroservices/payload/ResetRequest.java
- src/main/java/com/rmit/sept/bk_loginmicroservices/payload/Token.java

```

    < loginmicroservices
      > .mvn
      < src
        < main
          < java / com / rmit / sept / bk_loginservices
            < config
              J WebConfig.java
            < exceptions
              J CustomResponseEntityExceptionHandler.java
              J InvalidLoginResponse.java
              J UsernameAlreadyExistsException.java
              J UsernameAlreadyExistsResponse.java
            < mapper
              J SellerMapper.java
              J UserMapper.java
            < model
              J Seller.java
              J User.java
            < payload
              J JWTLoginSucessReponse.java
              J LoginRequest.java
              J RequestResponse.java
              J ResetRequest.java
              J Token.java
            < Repositories
              J UserRepository.java
            < security
              J JwtAuthenticationEntryPoint.java
              J JwtAuthenticationFilter.java
              J JwtTokenProvider.java
              J SecurityConfig.java
              J SecurityConstant.java

```

```

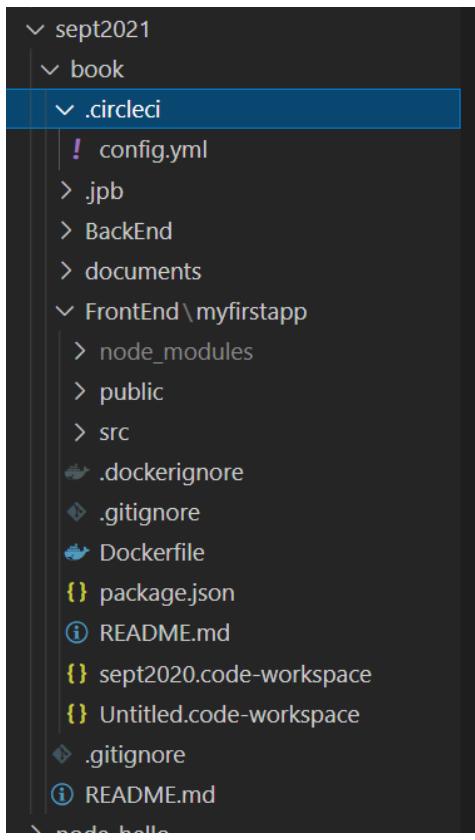
    < services
      J CustomUserDetailsService.java
      J MapValidationErrorService.java
      J UserService.java
    < validator
      J UserValidator.java
    < web
      J UserController.java
      J ms_login.java
    > resources
    > test
    > target
    ♦ .gitignore
    ♣ loginmicroservices.iml
    ♦ mvnw
    Windows mvnw.cmd
    ♣ pom.xml

```

- The following are the contents of FrontEnd:

```
> node_modules
> public
└ src
  > actions
  > components
    > reducers
    > securityUtils
    # App.css
    JS App.js
    JS App.test.js
    # index.css
    JS index.js
    └ logo.svg
    JS serviceWorker.js
    JS setupTests.js
    JS store.js
  └ .dockerignore
  └ .gitignore
  └ Dockerfile
  { package-lock.json
  { package.json
  { README.md
  { sept2020.code-workspace
  { Untitled.code-workspace
```

- To the root of the file a .circleci folder was created containing a config.yml file. A dockerfile was added to the same root directory for dockerization



To the FrontEnd directory, the App.test.js file was modified to include new tests written to test the application's front end through Jest.



book	113	! Failed	build-and-test	circleci-project-setup	2m ago	2m 15s	Ci	Gx	...
bf3ec7d final push of jest testing									
Jobs			✓ build 129			1m 31s			
			! test 130			37s			

Execution of tests ^

New user stories:

- User story #44: Public user reviews books on the app
- User story #45: Public user reviews users on the app

Credentials for default admin account:

- Username: dipto@mail.com
- Password: septsept

Summary of functionality:

After logging in, an admin would be presented with the home page, featuring an updated navbar in the header. An admin account can access a Transaction History page by clicking the link in the header. On the Transaction History page, the admin could view a summary book transactions that have occurred between users, as well as download a csv file by clicking the button at the bottom of the page scrolling down (Appendix A.2.). This caters to both user stories #1 and #2 because it shows information about the book being sold, the seller and the buyer. Clicking the Inbox link on the header, an admin is taken to a page where they may approve or reject public users who want to change their account type to shop owner (Appendix A.3.). On the edit books page, an admin could choose a book and edit its details, this communicates with the SQL database to update the details of the book via the books DAO in the back end, which is called upon by the books controller (Appendix A.4.). Similarly, on the User List page, the admin is presented with a UI that communicates with the SQL database to update the table of users, where they may edit, block or add a new admin user by clicking the appropriate buttons intuitively (Appendix A.5.). Clicking add user would open a page where the admin could register an account for a new admin (Appendix A.6.). Requests pertaining to any user are handled in loginmicroservices to maintain consistency with the base code.

The search function is completed with the addition of searching by category. This feature is similar to the other search features. The user simply enters a category of a book in the search bar in the catalogue (Appendix B.1.).

A shop owner/publisher account can access the Sell Book page from the link in the navbar in the header, which takes them to a page to fill information about a new/used book that they want to sell. The post request is handled in bookmicroservices similarly to admin adding books (Appendix C.1.).

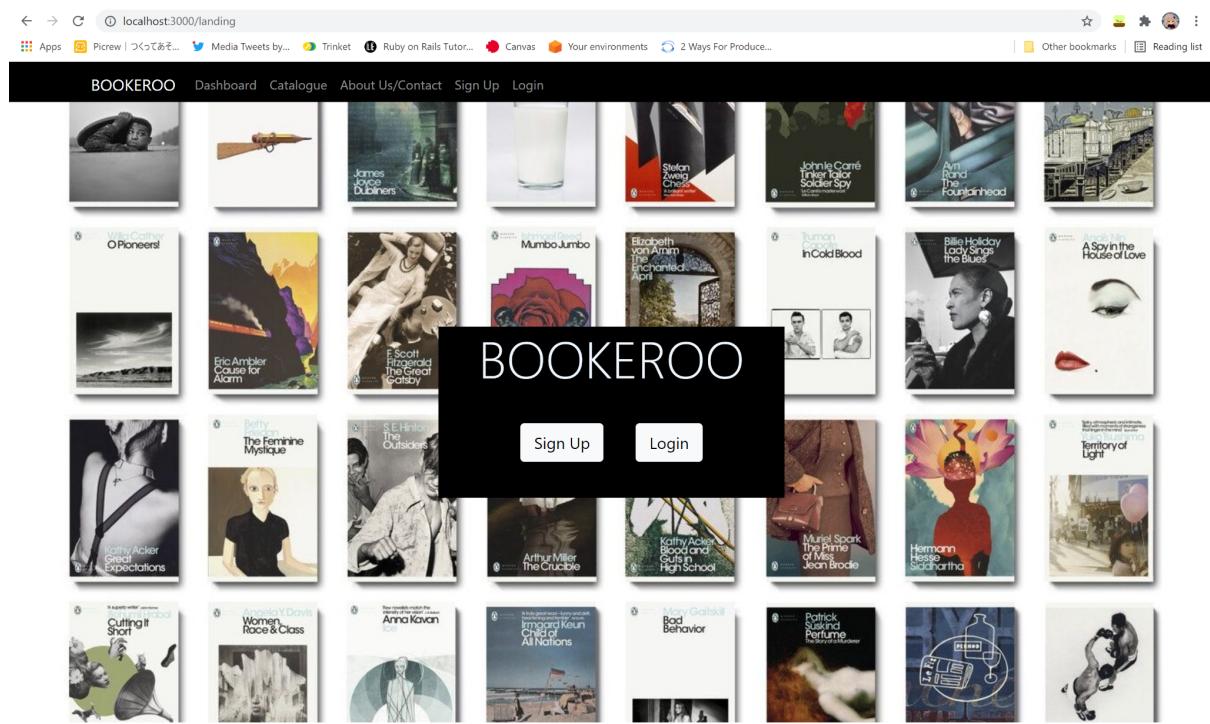
Shop owners/publishers and public users can access a current orders page, where they can oversee past and current orders, as well as cancel them (Appendix D.1.).

On the page of a book listing, customers can purchase books on the app via PayPal, which has been implemented using PayPal API. This allows them to enter their PayPal credentials and receive the fee. Users may also share a link to the book listing. Finally, users can submit a review by filling in the text box on the left and clicking the submission button.

Appendix:

A.

A.1.



A.2.

localhost:3000/transactionHistory

BOOKEROO Catalogue About Us/Contact Add Book Edit Book User List BlackList Inbox Transaction History Current Orders Settings Log Out

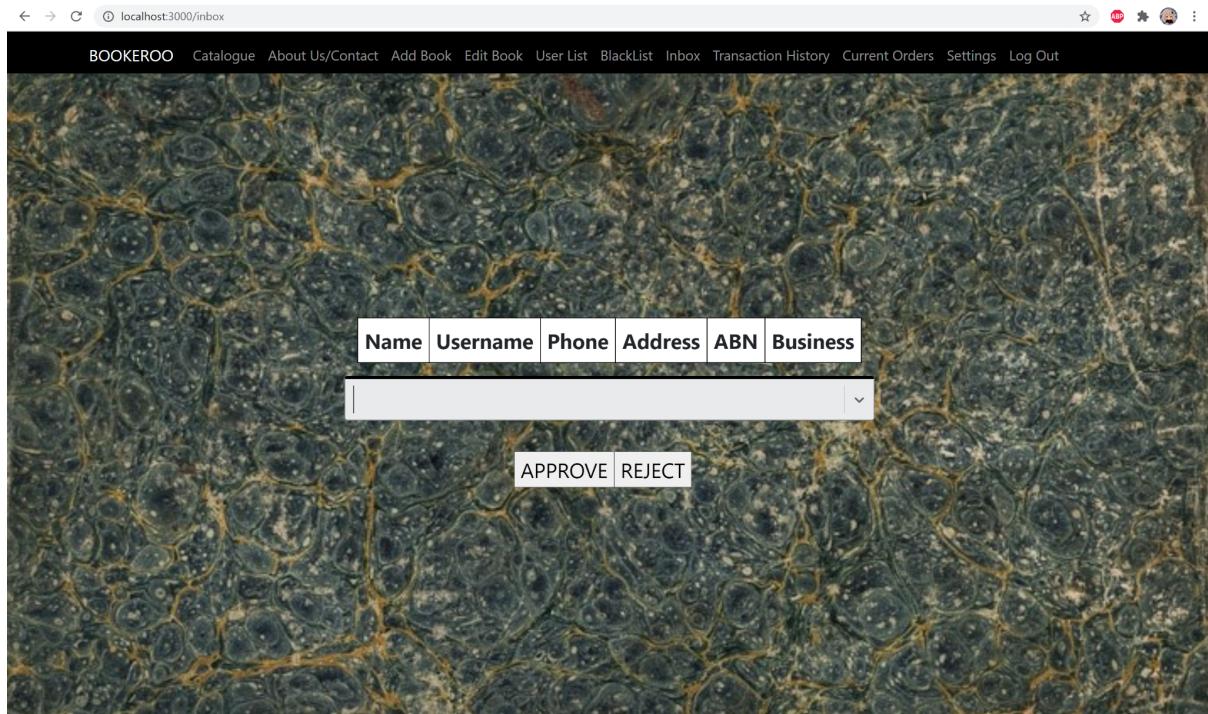
Order ID	Seller ID	Buyer ID	Qty	Price	Book ID	Title	Date
1	2	3	4	5	6	book	1634521656000
2	1	2	1	3	1	Here comes the sun	1634521656000
3	1	2	1	3	1	Here comes the sun	1634521656000
4	13	0	6	3	1	Dark Emu	1634521656000
5	23	12	2	35	29	Against The Day	1634521656000
6	23	12	1	25	17	All Quiet on the Western Front	1634521656000
7	18	12	1	20	9	Little Women	1634521656000
8	18	12	1	20	9	Little Women	1634521656000
13	13	12	1	20	1	Dark Emu	1634563479000
14	13	12	1	20	1	Dark Emu	1634563591000
15	13	12	1	20	1	Dark Emu	1634563890000
17	13	12	1	20	1	Dark Emu	1634564678000
18	13	12	1	20	1	Dark Emu	1634564748000
19	13	12	1	20	1	Dark Emu	1634564905000

localhost:3000/transactionHistory

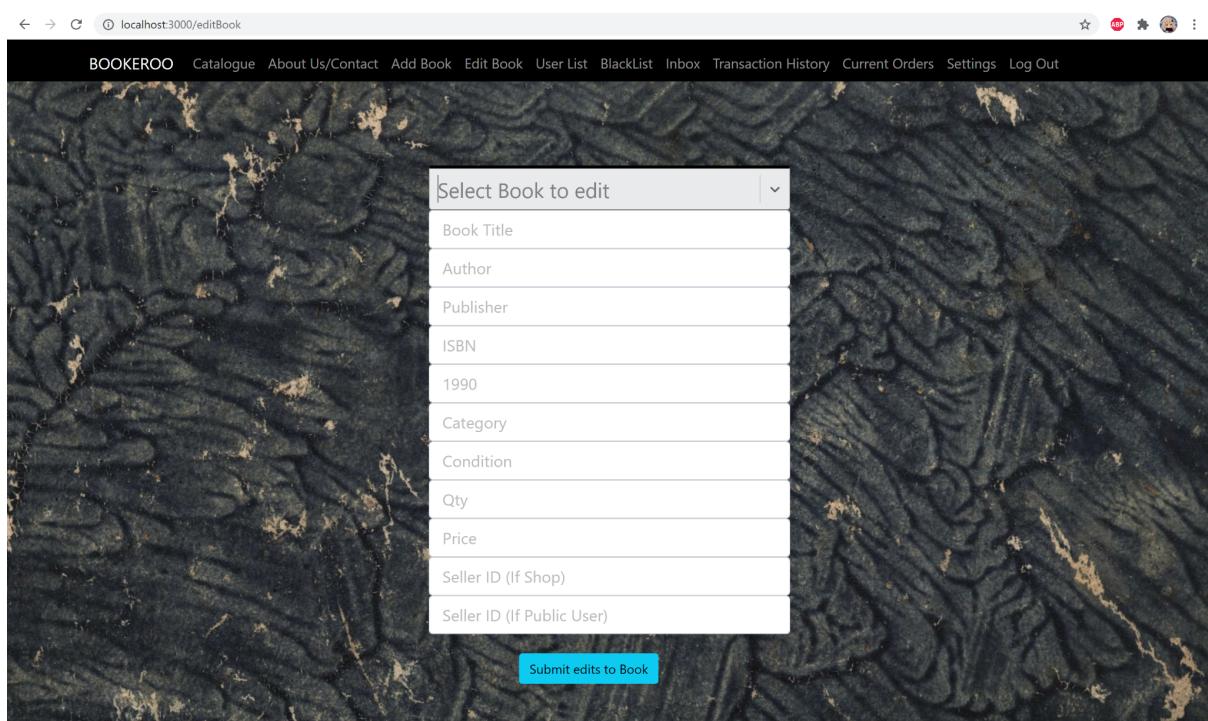
13	13	12	1	20	1	Dark Emu	1634563479000
14	13	12	1	20	1	Dark Emu	1634563591000
15	13	12	1	20	1	Dark Emu	1634563890000
17	13	12	1	20	1	Dark Emu	1634564678000
18	13	12	1	20	1	Dark Emu	1634564748000
19	13	12	1	20	1	Dark Emu	1634564905000
20	18	12	1	20	9	Little Women	1634525381000
21	13	12	1	3	1	Dark Emu	1634529068000
22	13	12	1	3	1	Dark Emu	1634644512000
23	23	12	1	25	17	All Quiet on the Western Front	1634531753000
24	23	12	1	25	17	All Quiet on the Western Front	1634531872000
25	23	12	1	25	17	All Quiet on the Western Front	1634532013000
26	23	12	1	25	17	All Quiet on the Western Front	1634645642000
27	23	12	1	25	17	All Quiet on the Western Front	1634534072000

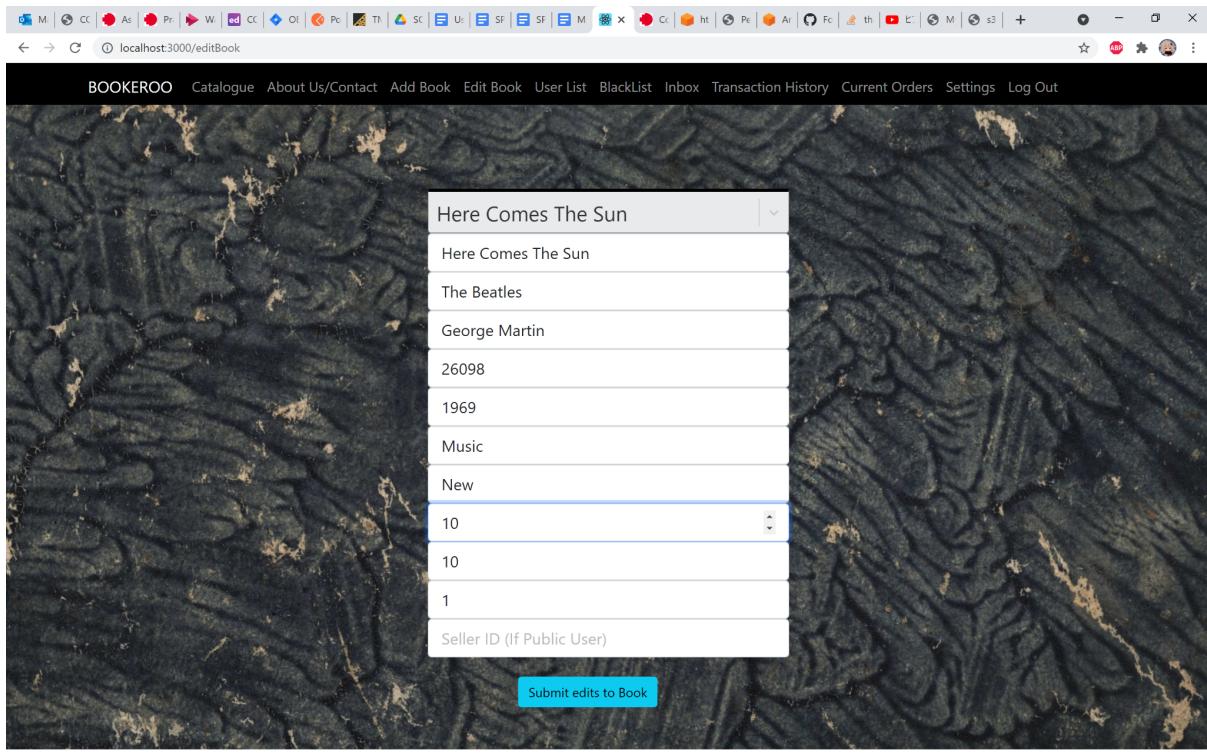
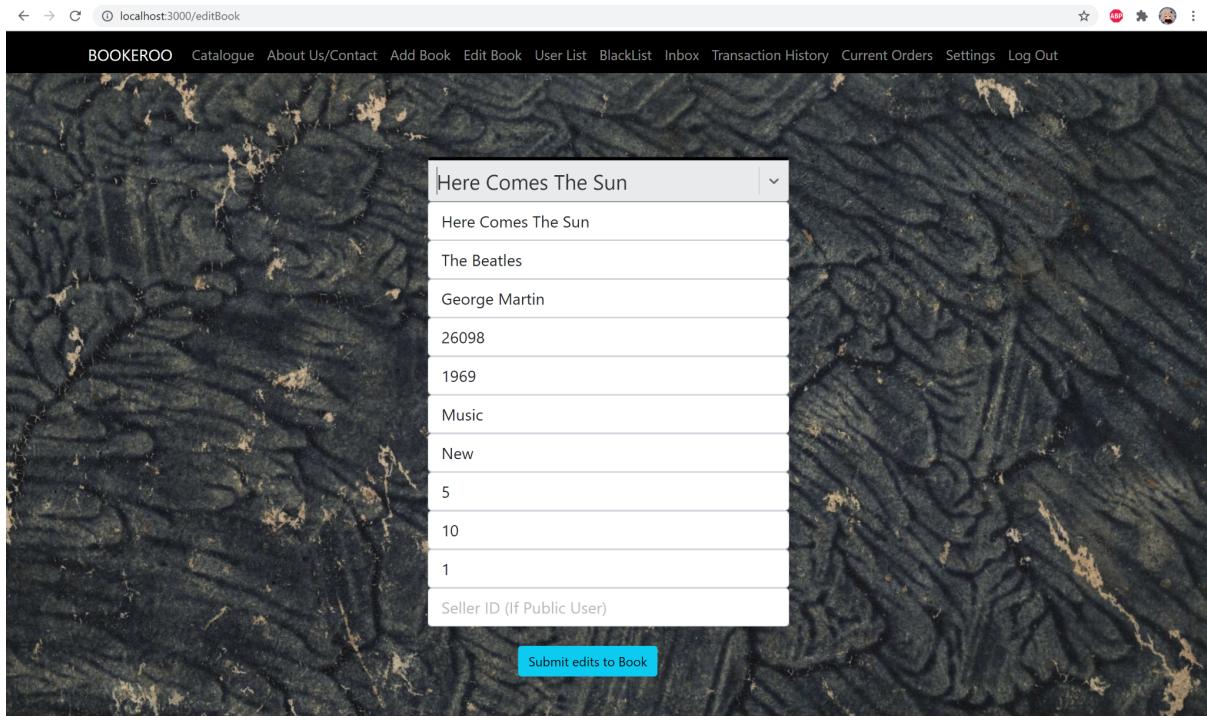
[Download](#)
Transactions as CSV

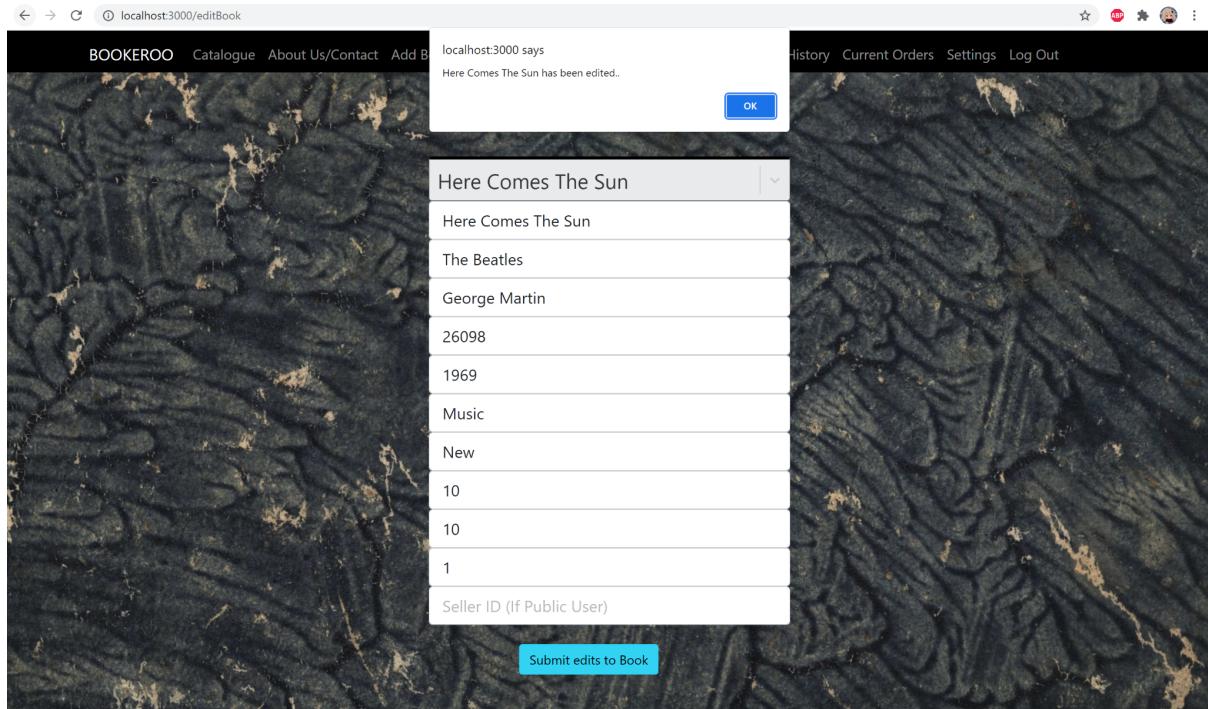
A.3.



A.4.



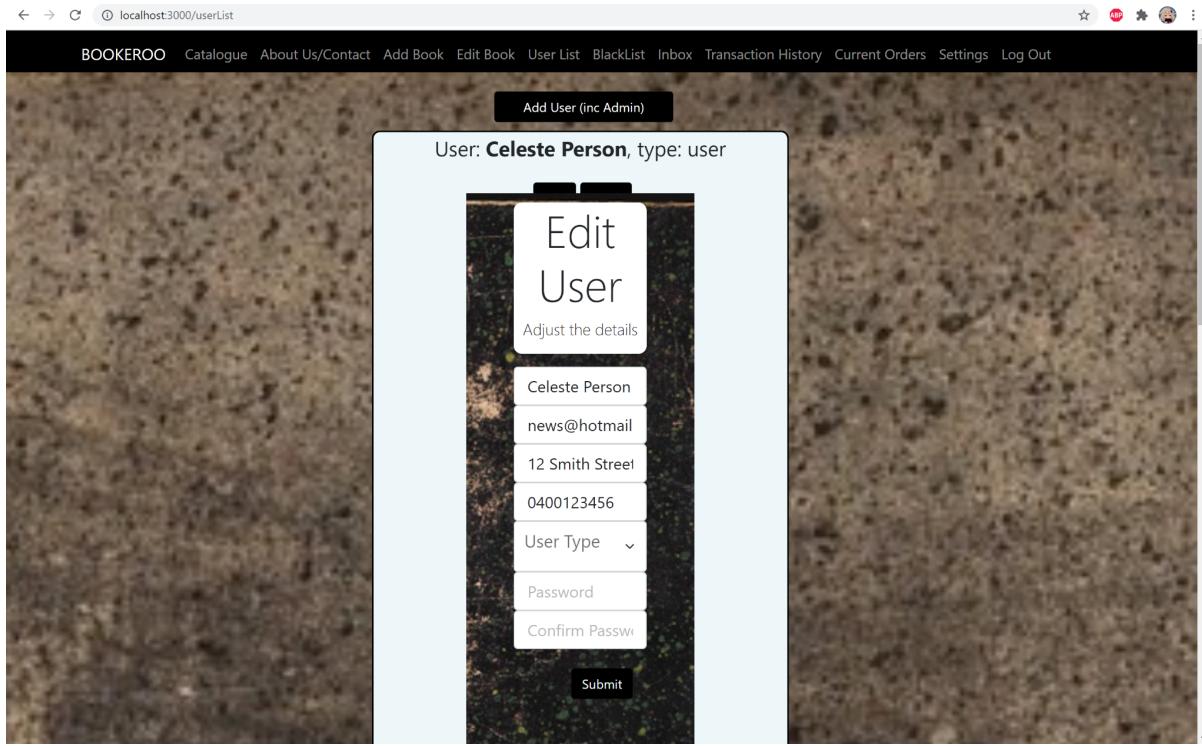




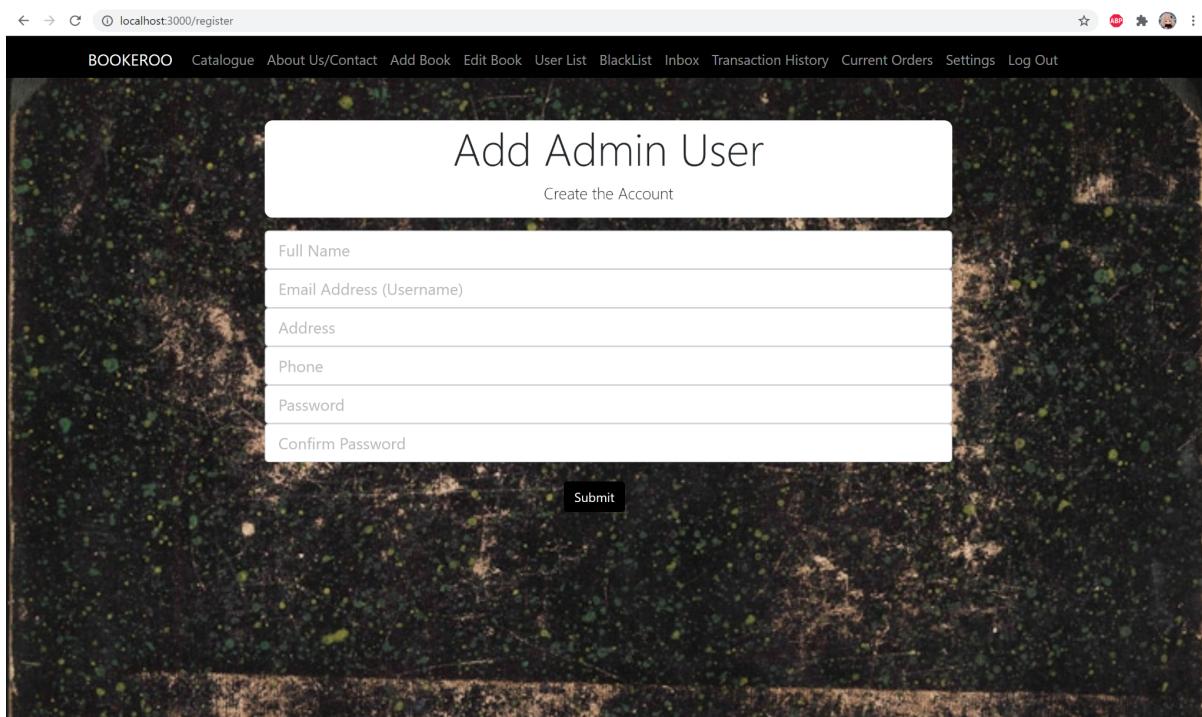
A.5.

A screenshot of a web browser displaying the BOOKEROO application at localhost:3000/userList. The page shows a list of users with their names, types, and actions. The users listed are:

- User: **Celeste Person**, type: user (Edit, Block)
- User: **Dipto**, type: admin (Edit, Block)
- User: **Grant**, type: publish (Edit, Block)
- User: **Ella**, type: user (Edit, Block)
- User: **dtest**, type: customer (Edit, Block)
- User: **Deeccc**, type: admin (Edit, Block)
- User: **Mateo**, type: shop (Edit, Block)



A.6.



B.

B.1.

← → ⌂ localhost:3000/catalogue/book

BOOKEROO Catalogue About Us/Contact Sign Up Login

Enter title, author, ISBN or category

The screenshot shows a grid of six book cards against a dark, textured background. Each card has an orange header with a white penguin icon and the text 'PENGUIN BOOKS'. The books listed are: 'Dark Emu' (Condition: Used, Sold By: Willy, Price: \$3, Qty: 5), 'Dubliners' (Not for sale), 'Little Women' (Condition: New, Sold By: Ella, Price: \$20, Qty: 2), 'All Quiet on the Western Front' (Condition: Used, Sold By: Willy, Price: \$3, Qty: 5), 'Here Comes The Sun' (Condition: New, Sold By: Unknown, Price: \$10, Qty: 10), and 'Against The Day' (Condition: Used, Sold By: Willy, Price: \$3, Qty: 5).

Dark Emu
Dubliners
Little Women
All Quiet on the Western Front
Here Comes The Sun
Against The Day

Condition: Used
Sold By: Willy
Price: \$3
Qty: 5

Condition: New
Sold By: Ella
Price: \$20
Qty: 2

Condition: Used
Sold By: Willy
Price: \$3
Qty: 5

Condition: New
Sold By: Unknown
Price: \$10
Qty: 10

Condition: Used
Sold By: Willy
Price: \$3
Qty: 5

← → ⌂ localhost:3000/catalogue/book

BOOKEROO Catalogue About Us/Contact Sign Up Login

music

The screenshot shows a single book card for 'Here Comes The Sun' from Penguin Books. The card includes the title, author, a penguin icon, and details: Condition: New, Sold By: Unknown, Price: \$10, and Qty: 10. The background is a dark, textured surface. A search bar at the top contains the text 'music'.

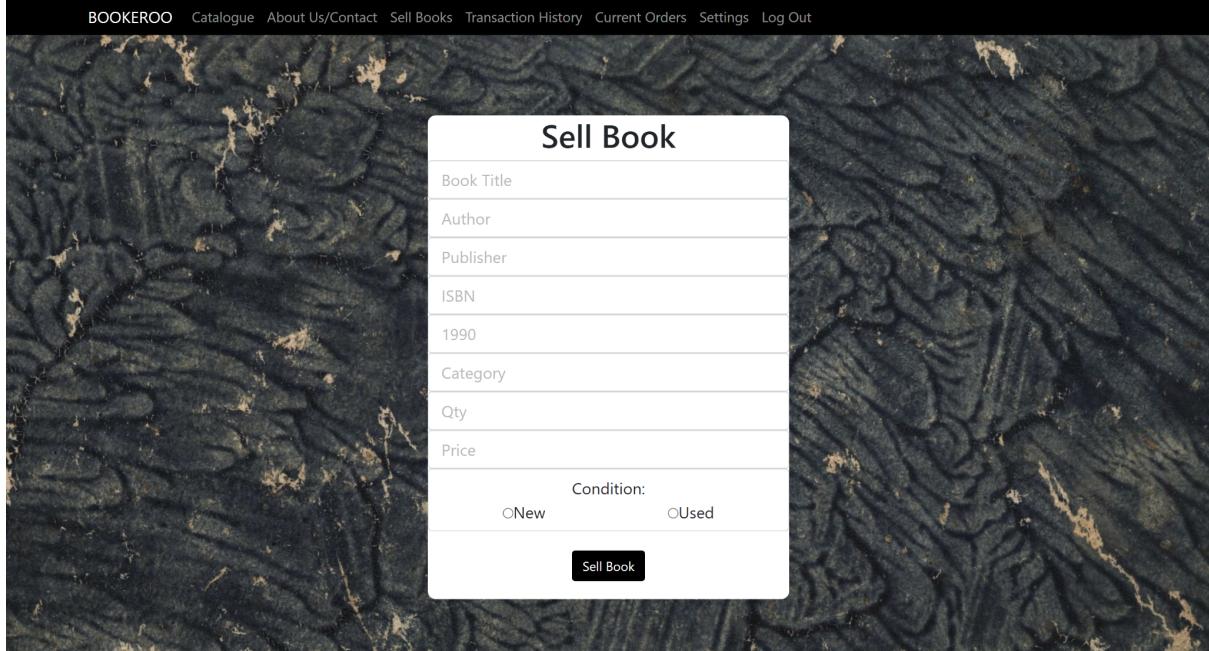
Here Comes The Sun

Condition: New
Sold By:
Price: \$10
Qty: 10

C.
C.1.

← → ⌛ ⚡ localhost:3000/addBook

BOOKEROO Catalogue About Us/Contact Sell Books Transaction History Current Orders Settings Log Out



Sell Book

Book Title
Author
Publisher
ISBN
1990
Category
Qty
Price

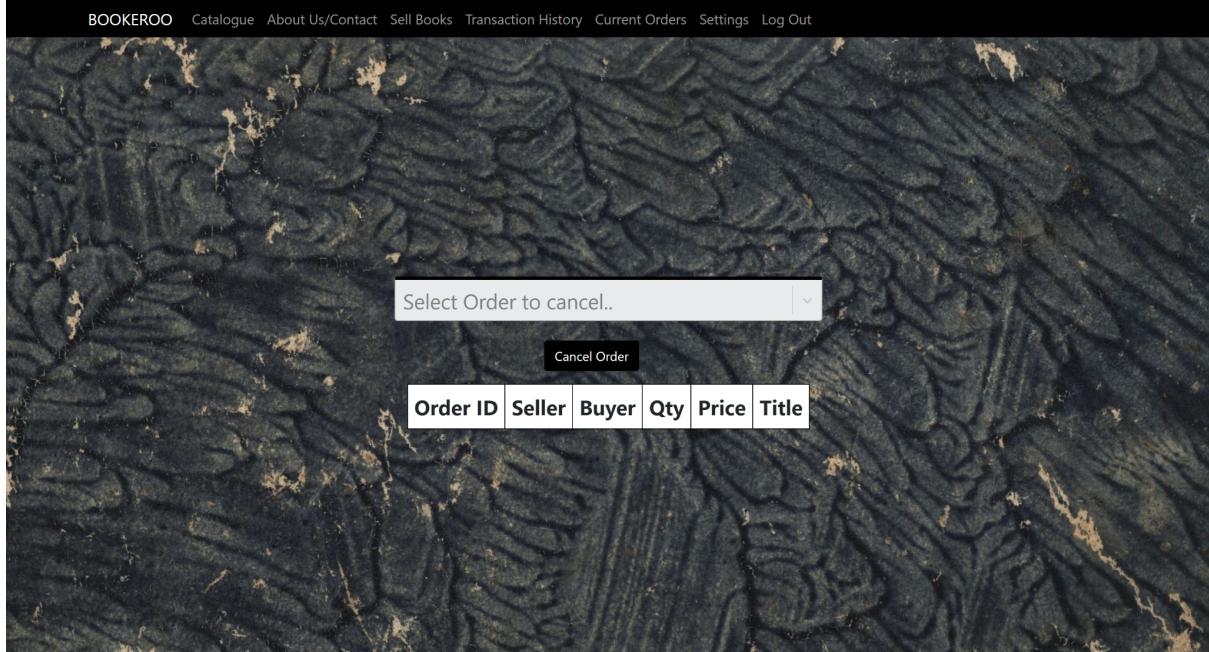
Condition:
 New Used

Sell Book

D.
D.1.

← → ⌛ ⚡ localhost:3000/currentOrders

BOOKEROO Catalogue About Us/Contact Sell Books Transaction History Current Orders Settings Log Out



Select Order to cancel..

Cancel Order

Order ID	Seller	Buyer	Qty	Price	Title
----------	--------	-------	-----	-------	-------

