The Application Library Management System is created using

> SpringBoot v2.2.1
>
> Spring Security v5.2.1
>
> ReactJS and Bootstrap
>
> Log4j
>
> Hibernate
>
> Postgresql 12.0

I have used

> **Jasypt Encoder   (v1.9.3)**
>
> **Bcrypt Password Encoder**

to encrypt all the passwords used in DB and application properties.

I have further used

> **JWT Token** for API Authentication
>
> **Role Based Authentication**
>
> Added **CORS Support**
>
> and Method Level Security

and created Junit SpringBootTest cases.

I have created a **Pageable Search** in spring and created fuzzy search UI with custom footbar for search result navigation, also added custom exception handling in the app.

Admin Can

- Search Books
- Add Book
- View all the issued books

Student Can

- Search Books
- View his issued books

Following roles were created for Library Management,

> ROLE_ADMIN,
>
> ROLE_SUPERVISOR
>
> ROLE_STUDENT

How to Install and Use the Application,

1. Run the SQL Scripts in the **librarydb_sql.sql** file , this will create the db named '*librarydb*' and a user named '*librarydbuser*' with password '*libarydbuser1*' and required tables with seed data for the application.

2. Download the Library Springboot Application code from git link https://github.com/meetarun/libmgmt

   Import the same in your preferred IDE or compile the code directly using maven command

   ```
   mvn install  or mvn clean install
   ```

   this will create the jar in target folder and .m2\repository\com\apus\arunlib\0.0.1-SNAPSHOT**arunlib-0.0.1-SNAPSHOT.jar**

3. Now start the jar using java -jar <jar path>

4. Download the UI code from the link, https://github.com/meetarun/libmgmt

5. Start the UI using the npm commands

   ```
   npm install
   npm start
   ```

6. Now access the UI with the URL http://localhost:3000/

I have also provided the **Postman Collection** of the Library APIs in the file `libraryDB_Target.postman_collection.json`, The application can be accessed via this endpoints using valid jwtToken in Authorization Header.
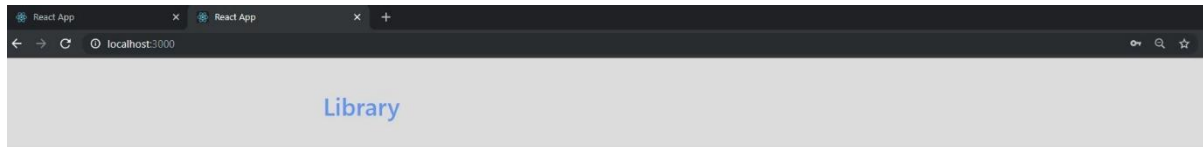
The user passwords can be encrypted using `PwdEncrypterUtil` class which has main method.

The db connection and test user I nput passwords can be encrypted and decrypted using `JasyptEncoderUtil`
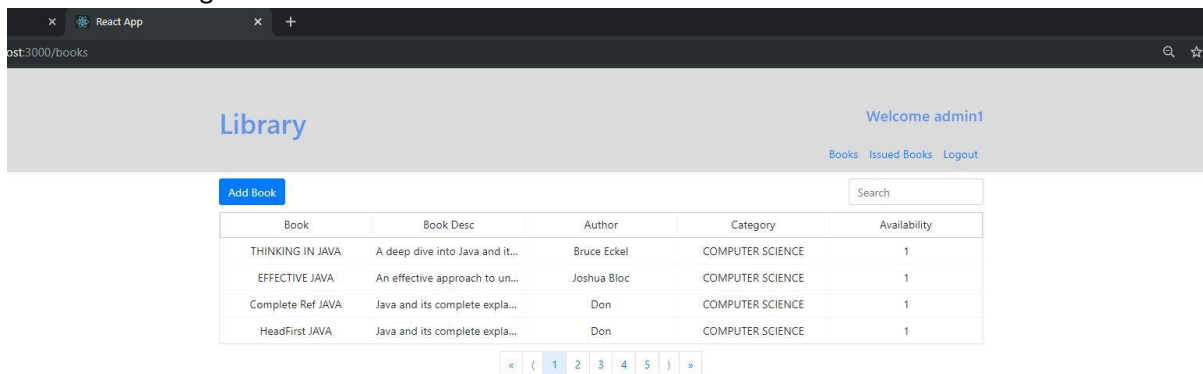
Provided in the application package.
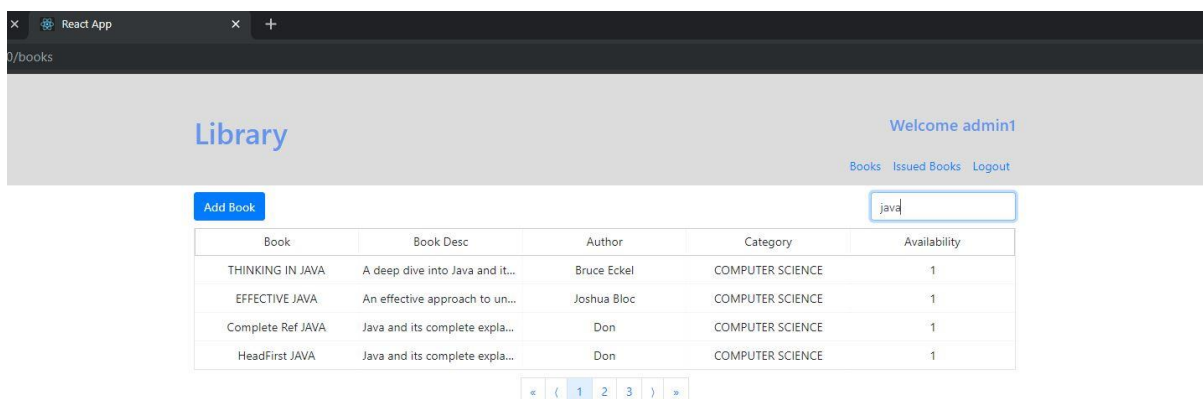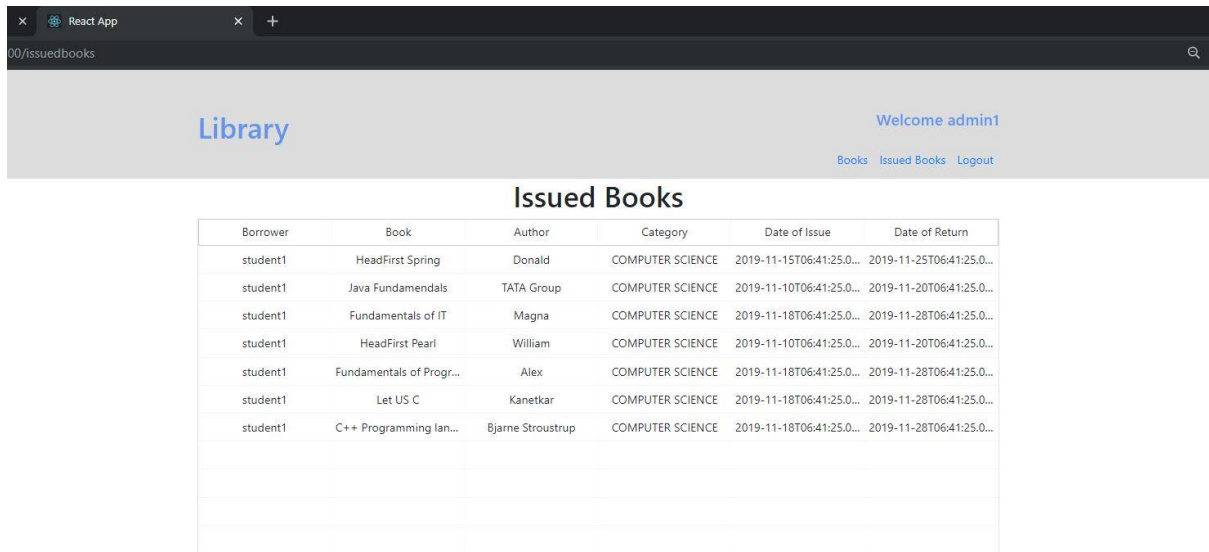
Few Screenshots of the Application

Login Page

Admin Home Page

Admin Book Search Page

Admin Issued Books Page
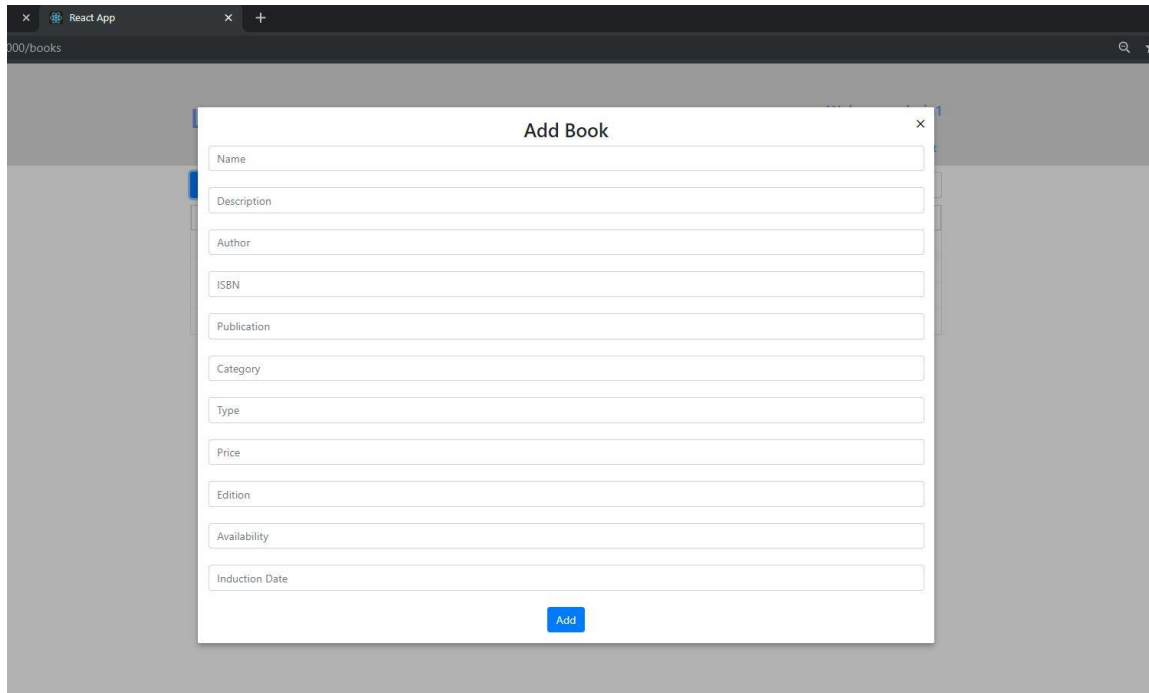

Admin Add Book Page

Student Home Page

Library

Welcome student1

Books   Issued Books   Logout

Search

Book	Book Desc	Author	Category	Availability
THINKING IN JAVA	A deep dive into Java and it...	Bruce Eckel	COMPUTER SCIENCE	1
EFFECTIVE JAVA	An effective approach to un...	Joshua Bloc	COMPUTER SCIENCE	1
Complete Ref JAVA	Java and its complete expla...	Don	COMPUTER SCIENCE	1
HeadFirst JAVA	Java and its complete expla...	Don	COMPUTER SCIENCE	1

« ‹ 1 2 3 4 5 › »

Student Issued Books Page

Library

Welcome student1

Books   Issued Books   Logout

Issued Books

Borrower	Book	Author	Category	Date of Issue	Date of Return
student1	HeadFirst Spring	Donald	COMPUTER SCIENCE	2019-11-15T06:41:25.0...	2019-11-25T06:41:25.0...
student1	Java Fundamendals	TATA Group	COMPUTER SCIENCE	2019-11-10T06:41:25.0...	2019-11-20T06:41:25.0...
student1	Fundamentals of IT	Magna	COMPUTER SCIENCE	2019-11-18T06:41:25.0...	2019-11-28T06:41:25.0...
student1	HeadFirst Pearl	William	COMPUTER SCIENCE	2019-11-10T06:41:25.0...	2019-11-20T06:41:25.0...
student1	Fundamentals of Progr...	Alex	COMPUTER SCIENCE	2019-11-18T06:41:25.0...	2019-11-28T06:41:25.0...
student1	Let US C	Kanetkar	COMPUTER SCIENCE	2019-11-18T06:41:25.0...	2019-11-28T06:41:25.0...
student1	C++ Programming lan...	Bjarne Stroustrup	COMPUTER SCIENCE	2019-11-18T06:41:25.0...	2019-11-28T06:41:25.0...