Library Management System for Group-Based Student Borrowing

Oluwatoyin Kode

Group Members (Temitope Oyemade, Adedeji Oludayo, Edith Jiokengbogni)

Department of Computer Science

Bowie State University

1. Executive Summary

This document outlines the design, development, and deployment of a Library Management System tailored to manage book borrowing and return processes in a classroom setting. The system addresses the needs of 78 library books, a class of students categorized into groups, and their specific borrowing preferences. It incorporates features such as borrowing status, return tracking, group membership identification, and book ratings.

2. Problem Definition

The challenge is to design a database system for a class with students divided into four groups (A-D), each consisting of three members. Each student has three preferred books from a longlist of 78 books. The system should manage book borrowing, return dates, and due notifications. The database must allow searching by student name to display borrowed books and corresponding due dates. Additional features include average book rating calculations and listing books by rating performance.

3. Requirements

- Functional Requirements:
    - Book Management: Create and manage a database of 78 books with relevant details (title, author, rating, etc).
    - Student Management: Store student information, including names, groups, and book preferences.
    - Borrowing Records: Track borrowing status, due dates, and return information.
    - Queries: Users can search by student name, group membership, and book availability.
    - Reporting: Display books with above and below-average ratings.
- Non-Functional Requirements:
    - Usability: User-friendly interface for queries.
    - Accessibility: Hosted on GitHub Pages.

4. Proposed Solution

We propose a Library Management System with the following features:

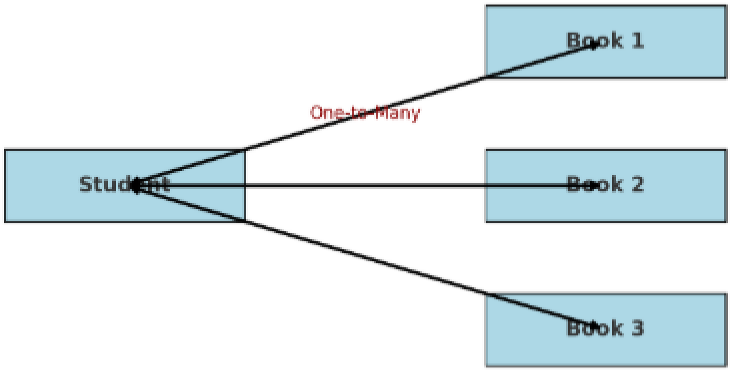
- A relational database to manage books, students, and group memberships.
- A user interface (UI) for querying borrowing details and viewing group information.
- A backend system to calculate average ratings and display books above or below the average.
- A method to dynamically update and query the database.

The system will be developed using Java for the backend, MySQL for the database, and a web interface hosted on GitHub Pages.

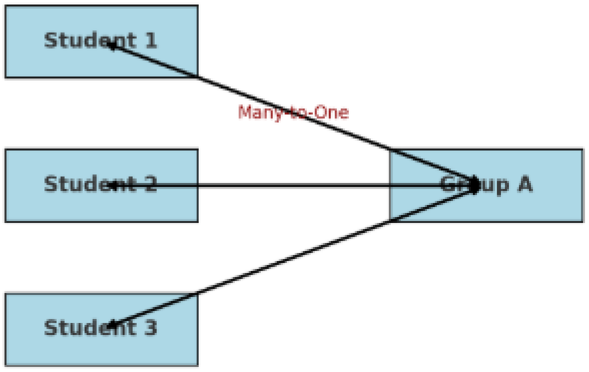
5. Design and Methodologies

5.1. Database Design

- Entities:
  - Books (BookID, Title, Author, Rating, BorrowedStatus)
  - Students (StudentID, FirstName, LastName, Group, PreferredBooks)
  - Groups Table: (group_id, group_name)
- Relationships and Entity-Relationship Diagram (ERD):
  - One-to-Many: A student can borrow multiple books.



  - Many-to-One: Multiple students (3) belong to a group.

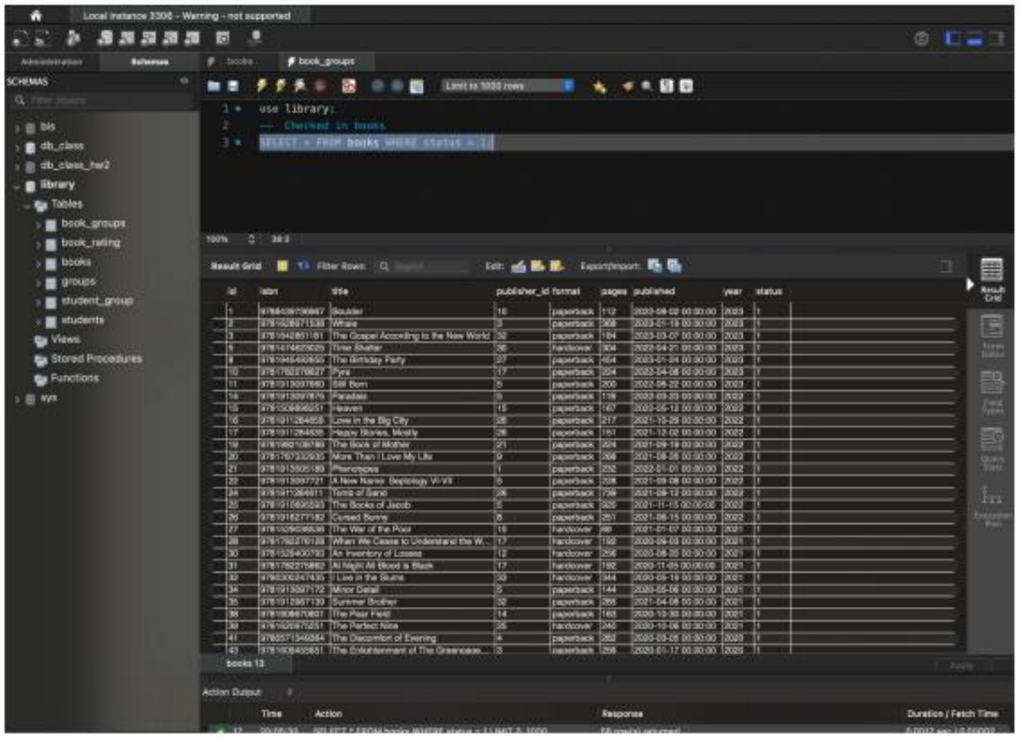


Normalization: Ensuring no data redundancy and efficient querying by dividing the database into normalized tables.

- SQL Queries:

- Query for borrowed books and return status:

SELECT * FROM books WHERE status = 1;



```
use library;
-- Checked in books.
SELECT * FROM books WHERE status = 1;
```

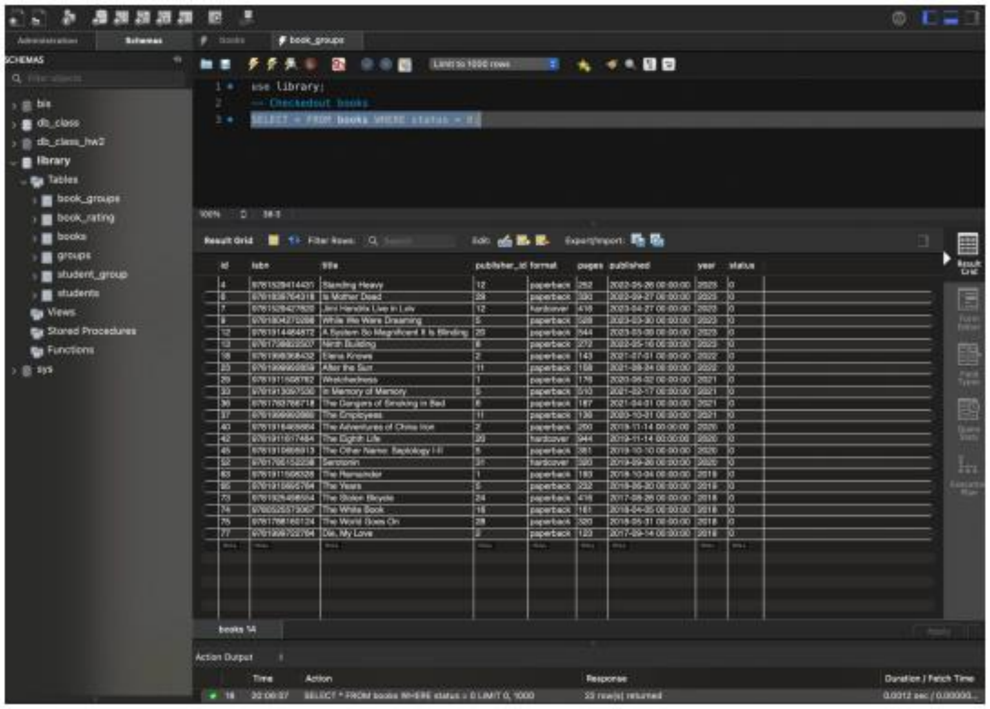
id	isbn	title	publisher_id	format	pages	published	year	status
1	9788439736867	Boulder	10	paperback	112	2022-08-02 00:00:00	2023	1
2	9781529971538	Whale	3	paperback	368	2023-01-19 00:00:00	2023	1
3	9781642861181	The Gospel According to the New World	32	paperback	184	2023-03-07 00:00:00	2023	1
5	9781474823025	Time Shelter	30	hardcover	304	2022-04-21 00:00:00	2023	1
8	9781945492655	The Birthday Party	27	paperback	464	2023-01-24 00:00:00	2023	1
10	9781782276627	Pyre	17	paperback	224	2022-04-08 00:00:00	2023	1
11	9781913097660	Still Born	5	paperback	200	2022-06-22 00:00:00	2023	1
14	9781913097875	Paradais	5	paperback	118	2022-03-23 00:00:00	2022	1
15	9781509896251	Heaven	15	paperback	167	2022-05-12 00:00:00	2022	1
16	9781911284658	Love in the Big City	28	paperback	217	2021-10-28 00:00:00	2022	1
17	9781911284635	Happy Stories, Mostly	28	paperback	161	2021-12-02 00:00:00	2022	1
19	9781662108789	The Book of Mother	21	paperback	224	2021-09-19 00:00:00	2022	1
20	9781787332935	More Than I Love My Life	9	paperback	268	2021-08-26 00:00:00	2022	1
21	9781913505189	Phenotypes	1	paperback	232	2022-01-01 00:00:00	2022	1
22	9781913097721	A New Name: Septology VI-VII	5	paperback	208	2021-09-08 00:00:00	2022	1
24	9781911284611	Tomb of Sand	28	paperback	739	2021-08-12 00:00:00	2022	1
25	9781910695593	The Books of Jacob	5	paperback	920	2021-11-15 00:00:00	2022	1
26	9781916277182	Cursed Bunny	8	paperback	251	2021-08-15 00:00:00	2022	1
27	[illegible]	The War of the Poor	16	hardcover	88	2021-01-07 00:00:00	2021	1
28	9781782276128	When We Cease to Understand the W...	17	hardcover	192	2020-09-03 00:00:00	2021	1
30	9781529400733	An Inventory of Losses	12	hardcover	256	2020-08-20 00:00:00	2021	1
31	9781782276862	At Night All Blood is Black	17	hardcover	192	2020-11-05 00:00:00	2021	1
32	9780300247435	I Live in the Slums	33	hardcover	344	2020-05-19 00:00:00	2021	1
34	9781913097172	Minor Detail	5	paperback	144	2020-05-06 00:00:00	2021	1
35	9781912987139	Summer Brother	32	paperback	288	2021-04-08 00:00:00	2021	1
38	9781908670601	The Pear Field	14	paperback	192	2020-10-30 00:00:00	2021	1
39	9781620975251	The Perfect Nine	25	hardcover	240	2020-10-06 00:00:00	2021	1
41	9780571349364	The Discomfort of Evening	4	paperback	282	2020-03-05 00:00:00	2020	1
43	9781908455651	The Enlightenment of The Greengage...	3	paperback	256	2020-01-17 00:00:00	2020	1

books 13

Action Output

	Time	Action	Response	Duration / Fetch Time
17	20:06:30	SELECT * FROM books WHERE status = 1 LIMIT 0, 1000	58 row(s) returned	0.0012 sec / 0.00002

SELECT * FROM books WHERE status = 0;



```
use library;
-- Checkedout books
SELECT * FROM books WHERE status = 0;
```

id	isbn	title	publisher_id	format	pages	published	year	status
4	9781529414431	Standing Heavy	12	paperback	252	2022-05-26 00:00:00	2023	0
6	9781838764318	Is Mother Dead	29	paperback	330	2022-09-27 00:00:00	2023	0
7	9781529427820	Jimi Hendrix Live in Lviv	12	hardcover	416	2023-04-27 00:00:00	2023	0
9	[illegible]	While We Were Dreaming	5	paperback	528	2023-03-30 00:00:00	2023	0
12	9781914484872	A System So Magnificent It Is Blinding	20	paperback	544	2023-03-09 00:00:00	2023	0
13	9781739822507	Ninth Building	8	paperback	272	2022-05-16 00:00:00	2023	0
18	9781999368432	Elena Knows	2	paperback	143	2021-07-01 00:00:00	2022	0
23	[illegible]	After the Sun	11	paperback	158	2021-08-24 00:00:00	2022	0
29	9781911508782	Wretchedness	1	paperback	176	2020-06-02 00:00:00	2021	0
33	9781913097530	In Memory of Memory	5	paperback	510	2021-02-17 00:00:00	2021	0
36	9781782766718	The Dangers of Smoking in Bed	6	paperback	187	2021-04-01 00:00:00	2021	0
37	[illegible]	The Employees	11	paperback	130	2020-10-01 00:00:00	2021	0
40	9781916465864	The Adventures of China Iron	2	paperback	260	2019-11-14 00:00:00	2020	0
42	9781911617464	The Eighth Life	20	hardcover	944	2019-11-14 00:00:00	2020	0
45	9781910695913	The Other Name: Septology I-II	5	paperback	381	2019-10-10 00:00:00	2020	0
52	9781786152238	Serotonin	31	hardcover	320	2019-09-26 00:00:00	2020	0
63	9781911508328	The Remainder	1	paperback	183	2018-10-04 00:00:00	2019	0
65	9781910695794	The Years	5	paperback	232	2018-06-20 00:00:00	2019	0
73	[illegible]	The Stolen Bicycle	24	paperback	416	2017-08-26 00:00:00	2018	0
74	9780525573067	The White Book	16	paperback	161	2018-04-05 00:00:00	2018	0
75	9781786160124	The World Goes On	28	paperback	320	2018-05-31 00:00:00	2018	0
77	9781999722764	Die, My Love	2	paperback	123	2017-09-14 00:00:00	2018	0

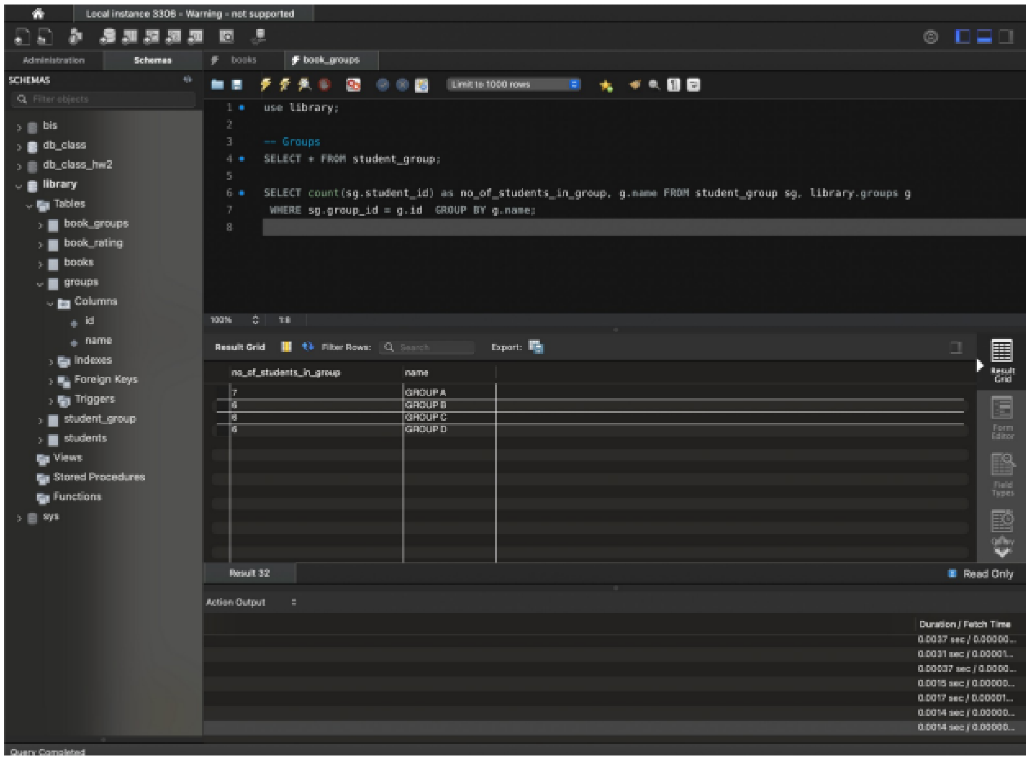
books 14

Action Output

	Time	Action	Response	Duration / Fetch Time
18	20:06:37	SELECT * FROM books WHERE status = 0 LIMIT 0, 1000	22 row(s) returned	0.0012 sec / 0.00000...

- Query for group affiliations and preferred books.

```
CREATE TABLE `groups` (
 `id` int NOT NULL AUTO_INCREMENT,
 `name` varchar(45) DEFAULT NULL,
 PRIMARY KEY (`id`),
 UNIQUE KEY `id_UNIQUE` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
SELECT count(sg.student_id) as no_of_students_in_group, g.name FROM student_group sg,
library.groups g
 WHERE sg.group_id = g.id GROUP BY g.name;
```



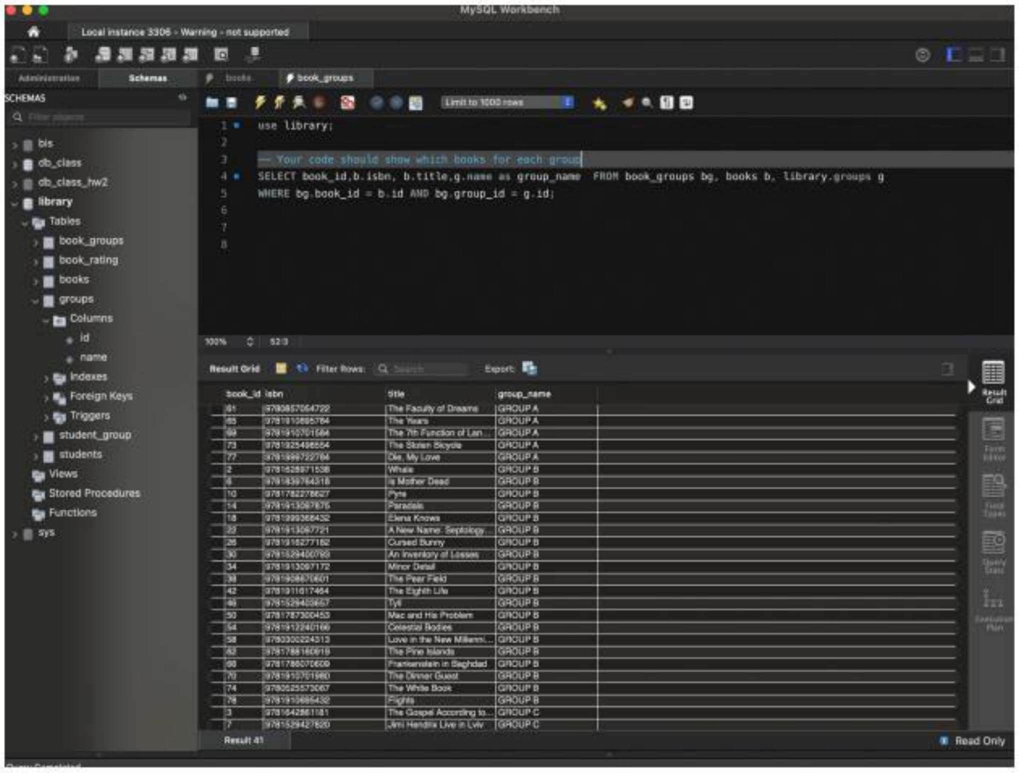
```
use library;

-- Groups
SELECT * FROM student_group;

SELECT count(sg.student_id) as no_of_students_in_group, g.name FROM student_group sg, library.groups g
 WHERE sg.group_id = g.id  GROUP BY g.name;
```

```
no_of_students_in_group   name
7                         GROUP A
6                         GROUP B
8                         GROUP C
6                         GROUP D
```

```
Duration / Fetch Time
0.0037 sec / 0.00000...
0.0031 sec / 0.00001...
0.00037 sec / 0.0000...
0.0015 sec / 0.00000...
0.0017 sec / 0.00001...
0.0014 sec / 0.00000...
0.0014 sec / 0.00000...
```

Query Completed

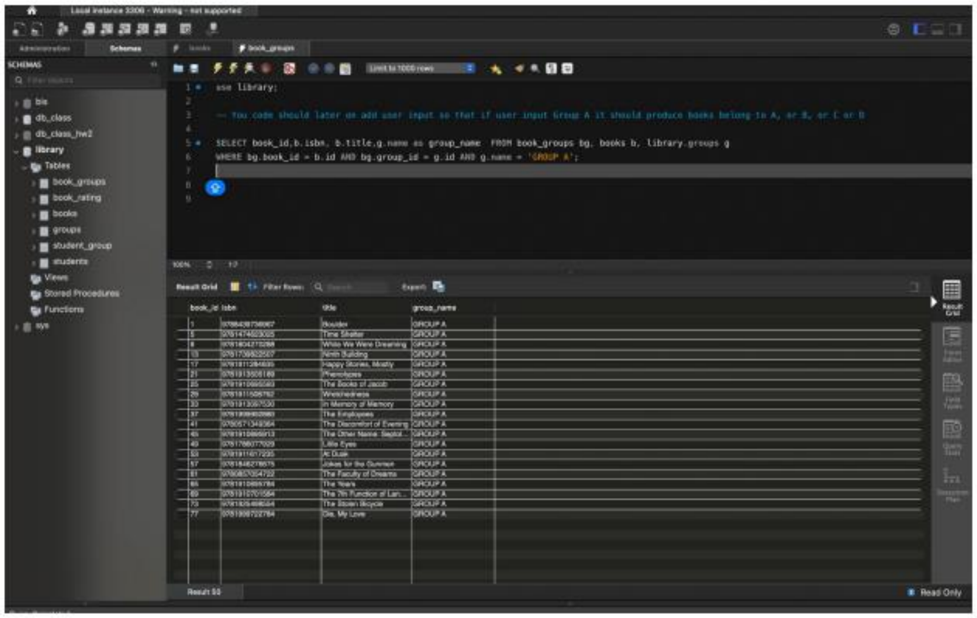
SELECT book_id,b.isbn, b.title,g.name as group_name FROM book_groups bg, books b, library.groups g
WHERE bg.book_id = b.id AND bg.group_id = g.id;



```
use library;

-- Your code should show which books for each group
SELECT book_id,b.isbn, b.title,g.name as group_name  FROM book_groups bg, books b, library.groups g
WHERE bg.book_id = b.id AND bg.group_id = g.id;
```

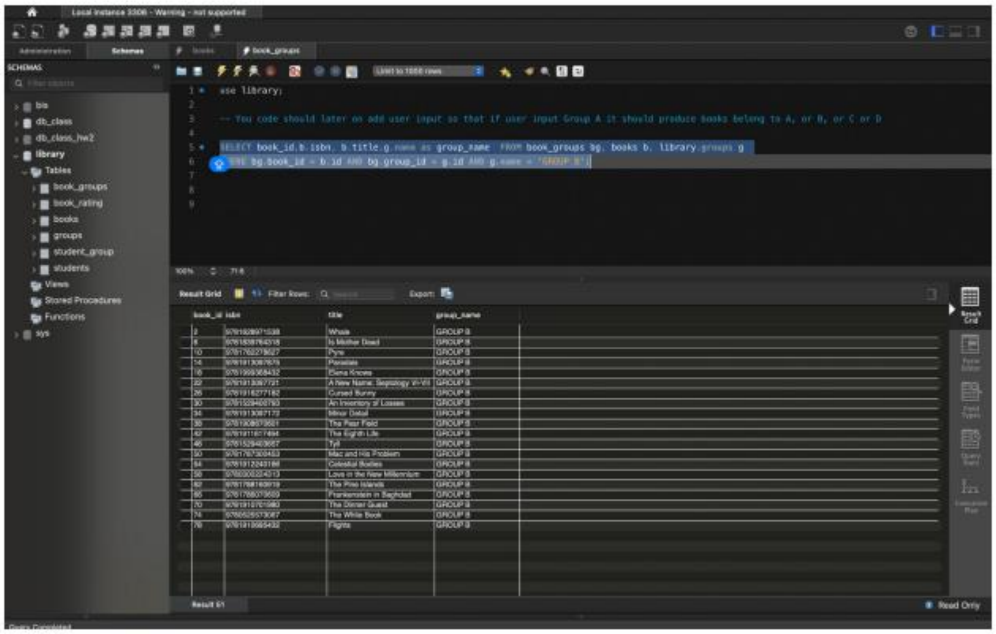
book_id	isbn	title	group_name
61	9780857064722	The Faculty of Dreams	GROUP A
65	9781910695784	The Years	GROUP A
69	9781910701584	The 7th Function of Lan...	GROUP A
73	9781925498554	The Stolen Bicycle	GROUP A
77	9781999722784	Die, My Love	GROUP A
2	9781628971538	Whale	GROUP B
6	9781839764318	Is Mother Dead	GROUP B
10	9781782278627	Pyre	GROUP B
14	9781913097675	Paradais	GROUP B
18	9781999368432	Elena Knows	GROUP B
22	9781913097721	A New Name: Septology...	GROUP B
26	9781916277182	Cursed Bunny	GROUP B
30	9781529400793	An Inventory of Losses	GROUP B
34	9781913097172	Minor Detail	GROUP B
38	9781908670601	The Pear Field	GROUP B
42	9781911617464	The Eighth Life	GROUP B
46	9781529403657	Tyll	GROUP B
50	9781787300453	Mac and His Problem	GROUP B
54	9781912240166	Celestial Bodies	GROUP B
58	9780300224313	Love in the New Millenni...	GROUP B
62	9781788160919	The Pine Islands	GROUP B
66	9781786070609	Frankenstein in Baghdad	GROUP B
70	9781910701980	The Dinner Guest	GROUP B
74	9780525573067	The White Book	GROUP B
78	9781910695432	Flights	GROUP B
3	9781642861181	The Gospel According to...	GROUP C
7	9781529427620	Jimi Hendrix Live in Lviv	GROUP C

SELECT book_id,b.isbn, b.title,g.name as group_name FROM book_groups bg, books b, library.groups g
WHERE bg.book_id = b.id AND bg.group_id = g.id AND g.name = 'GROUP A';



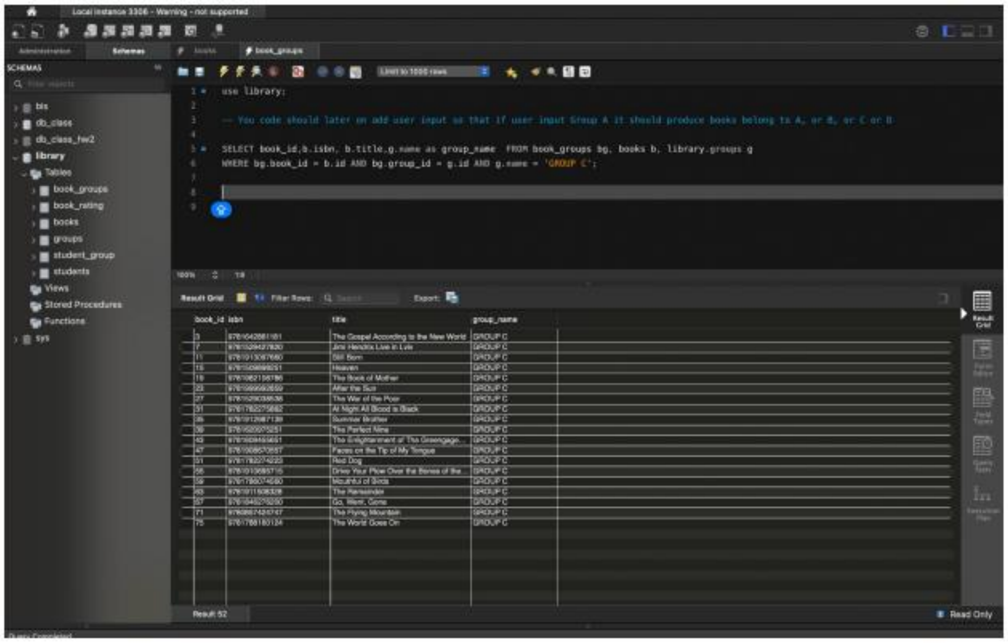
```sql
use library;

-- You code should later on add user input so that if user input Group A it should produce books belong to A, or B, or C or D

SELECT book_id,b.isbn, b.title,g.name as group_name  FROM book_groups bg, books b, library.groups g
WHERE bg.book_id = b.id AND bg.group_id = g.id AND g.name = 'GROUP A';
```

book_id	isbn	title	group_name
1	9788408738907	Boulder	GROUP A
5	9781474623025	Time Shelter	GROUP A
9	9781804273288	While We Were Dreaming	GROUP A
13	9781739822507	Ninth Building	GROUP A
17	9781911284635	Happy Stories, Mostly	GROUP A
21	9781913505189	Phenotypes	GROUP A
25	9781910695583	The Books of Jacob	GROUP A
29	9781911508782	Wretchedness	GROUP A
33	9781913097530	In Memory of Memory	GROUP A
37	9781800960880	The Employees	GROUP A
41	9780571349364	The Discomfort of Evening	GROUP A
45	9781910695913	The Other Name: Septol...	GROUP A
49	9781786077929	Little Eyes	GROUP A
53	9781911617235	At Dusk	GROUP A
57	9781846278675	Jokes for the Gunmen	GROUP A
61	9780857054722	The Faculty of Dreams	GROUP A
65	9781910695784	The Years	GROUP A
69	9781910701584	The 7th Function of Lan...	GROUP A
73	9781925498564	The Stolen Bicycle	GROUP A
77	9781908722784	Die, My Love	GROUP A



```sql
use library;

-- You code should later on add user input so that if user input Group A it should produce books belong to A, or B, or C or D

SELECT book_id,b.isbn, b.title,g.name as group_name  FROM book_groups bg, books b, library.groups g
WHERE bg.book_id = b.id AND bg.group_id = g.id AND g.name = 'GROUP B';
```

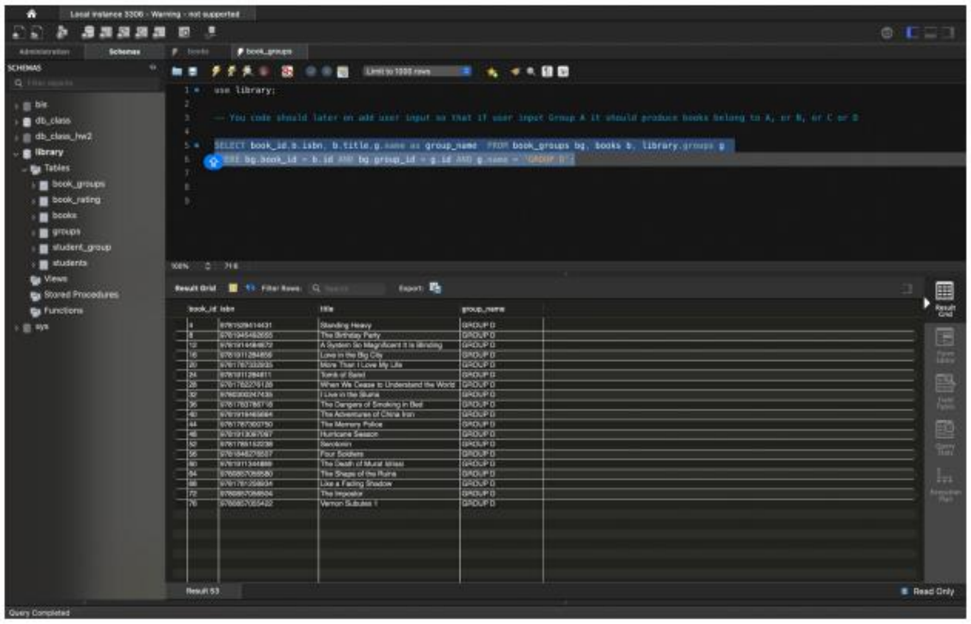
book_id	isbn	title	group_name
2	9781802971538	Whale	GROUP B
6	9781838764318	Is Mother Dead	GROUP B
10	9781782278627	Pyre	GROUP B
14	9781913097675	Paradais	GROUP B
18	9781999368432	Elena Knows	GROUP B
22	9781913097731	A New Name: Septology VI-VII	GROUP B
26	9781916277182	Cursed Bunny	GROUP B
30	9781529400793	An Inventory of Losses	GROUP B
34	9781913097172	Minor Detail	GROUP B
38	9781908673601	The Pear Field	GROUP B
42	9781911617494	The Eighth Life	GROUP B
46	9781529403657	Tyll	GROUP B
50	9781787303453	Mac and His Problem	GROUP B
54	9781912243186	Celestial Bodies	GROUP B
58	9780300224013	Love in the New Millennium	GROUP B
62	9781788163919	The Pine Islands	GROUP B
66	9781786073669	Frankenstein in Baghdad	GROUP B
70	9781910701980	The Dinner Guest	GROUP B
74	9780525573087	The White Book	GROUP B
78	9781910695432	Flights	GROUP B



```sql
use library;

-- You code should later on add user input so that if user input Group A it should produce books belong to A, or B, or C or D

SELECT book_id,b.isbn, b.title,g.name as group_name  FROM book_groups bg, books b, library.groups g
WHERE bg.book_id = b.id AND bg.group_id = g.id AND g.name = 'GROUP C';
```

book_id	isbn	title	group_name
3	9781642861161	The Gospel According to the New World	GROUP C
7	9781529427820	Jimi Hendrix Live in Lviv	GROUP C
11	9781913087680	Still Born	GROUP C
15	9781509899251	Heaven	GROUP C
19	9781082156786	The Book of Mother	GROUP C
23	9781999992659	After the Sun	GROUP C
27	9781529038536	The War of the Poor	GROUP C
31	9781782275862	At Night All Blood is Black	GROUP C
35	9781912987139	Summer Brother	GROUP C
39	9781620975251	The Perfect Nine	GROUP C
43	9781609455651	The Enlightenment of The Greengage...	GROUP C
47	9781908670557	Faces on the Tip of My Tongue	GROUP C
51	9781782274223	Red Dog	GROUP C
55	9781913085715	Drive Your Plow Over the Bones of the...	GROUP C
59	9781780074560	Mouthful of Birds	GROUP C
63	9781911508328	The Remainder	GROUP C
67	9781846276260	Go, Went, Gone	GROUP C
71	9780857424747	The Flying Mountain	GROUP C
75	9781788180124	The World Goes On	GROUP C

Result 52



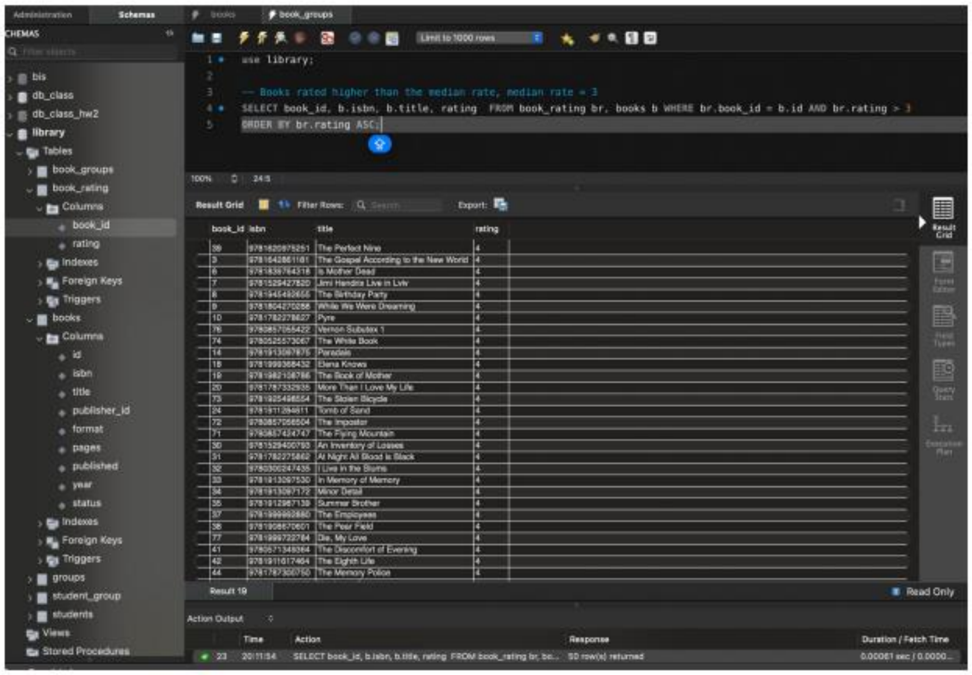
```sql
use library;

-- You code should later on add user input so that if user input Group A it should produce books belong to A, or B, or C or D

SELECT book_id,b.isbn, b.title,g.name as group_name  FROM book_groups bg, books b, library.groups g
WHERE bg.book_id = b.id AND bg.group_id = g.id AND g.name = 'GROUP D';
```

book_id	isbn	title	group_name
4	9781529414431	Standing Heavy	GROUP D
8	9781945492655	The Birthday Party	GROUP D
12	9781914484672	A System So Magnificent It is Blinding	GROUP D
16	9781911284856	Love in the Big City	GROUP D
20	9781787332935	More Than I Love My Life	GROUP D
24	9781911284811	Tomb of Sand	GROUP D
28	9781782276128	When We Cease to Understand the World	GROUP D
32	9780300247435	I Live in the Slums	GROUP D
36	9781783786716	The Dangers of Smoking in Bed	GROUP D
40	9781916465684	The Adventures of China Iron	GROUP D
44	9781787300750	The Memory Police	GROUP D
48	9781913087097	Hurricane Season	GROUP D
52	9781788163238	Serotonin	GROUP D
56	9781846276537	Four Soldiers	GROUP D
60	9781911344889	The Death of Murat Idrissi	GROUP D
64	9780857056580	The Shape of the Ruins	GROUP D
68	9781781258934	Like a Fading Shadow	GROUP D
72	9780857058504	The Impostor	GROUP D
76	9780857055422	Vernon Subutex 1	GROUP D

Result 53

- Query for book ratings above or below average.

Books rated higher than the median rate, median rate = 3
SELECT b.id,b.isbn, b.title, rating FROM book_rating br, books b WHERE br.book_id = b.id AND br.rating > 3;

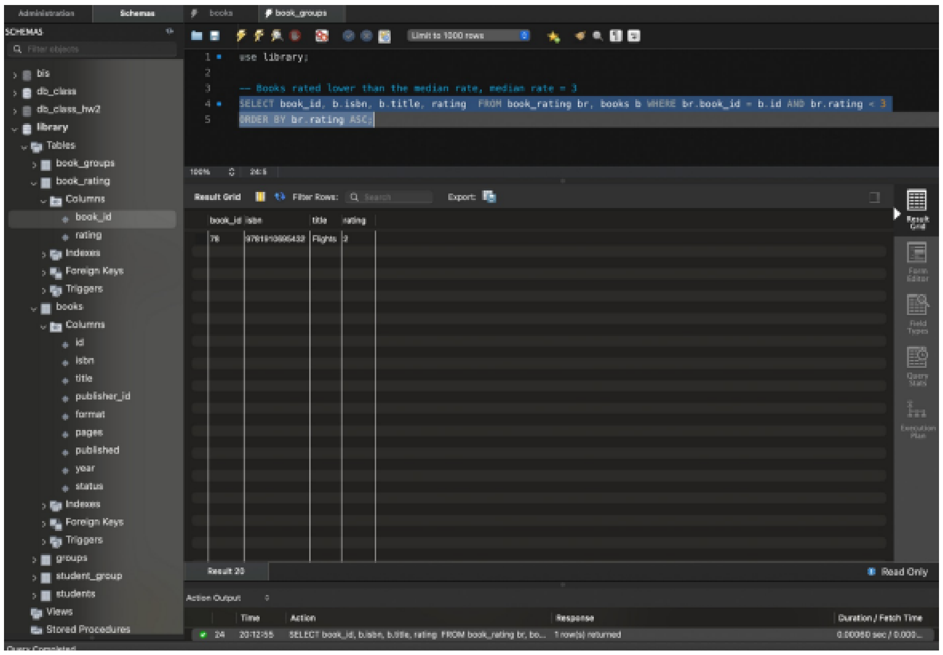


```
use library;

-- Books rated higher than the median rate, median rate = 3
SELECT book_id, b.isbn, b.title, rating  FROM book_rating br, books b WHERE br.book_id = b.id AND br.rating > 3
ORDER BY br.rating ASC;
```

book_id	isbn	title	rating
39	9781620975251	The Perfect Nine	4
3	9781642861181	The Gospel According to the New World	4
6	9781839764318	Is Mother Dead	4
7	9781529427820	Jimi Hendrix Live in Lviv	4
8	9781945492655	The Birthday Party	4
9	9781804270288	While We Were Dreaming	4
10	9781782278627	Pyre	4
76	9780857055422	Vernon Subutex 1	4
74	9780525573067	The White Book	4
14	9781913097875	Paradais	4
18	9781999368432	Elena Knows	4
19	9781982108786	The Book of Mother	4
20	9781787332935	More Than I Love My Life	4
73	9781925498554	The Stolen Bicycle	4
24	9781911284611	Tomb of Sand	4
72	9780857056504	The Impostor	4
71	9780857424747	The Flying Mountain	4
30	9781529400793	An Inventory of Losses	4
31	9781782275862	At Night All Blood is Black	4
32	9780300247435	I Live in the Slums	4
33	9781913097530	In Memory of Memory	4
34	9781913097172	Minor Detail	4
35	9781912987139	Summer Brother	4
37	9781999992880	The Employees	4
38	9781908670601	The Pear Field	4
77	9781999722784	Die, My Love	4
41	9780571348084	The Discomfort of Evening	4
42	9781911617464	The Eighth Life	4
44	9781787300750	The Memory Police	4

```
23  20:11:54  SELECT book_id, b.isbn, b.title, rating FROM book_rating br, bo...  50 row(s) returned  0.00061 sec / 0.0000...
```

Books rated lower than the median rate, median rate = 3 SELECT b.id,b.isbn, b.title, rating FROM book_rating br, books b WHERE br.book_id = b.id AND br.rating < 3;

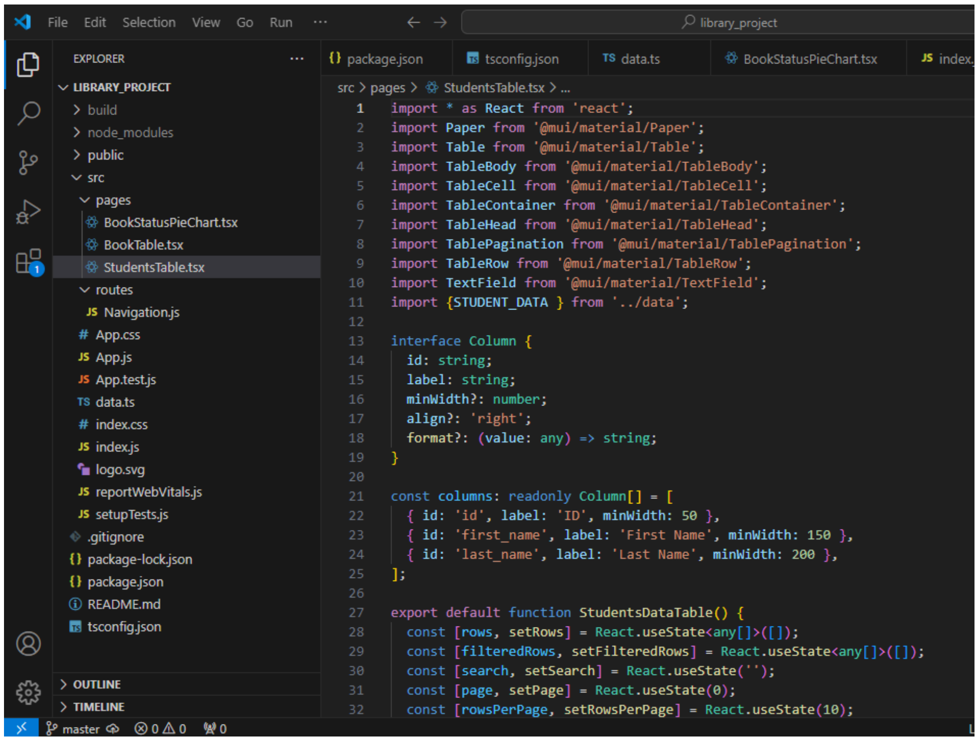


```
use library;

-- Books rated lower than the median rate, median rate = 3
SELECT book_id, b.isbn, b.title, rating  FROM book_rating br, books b WHERE br.book_id = b.id AND br.rating < 3
ORDER BY br.rating ASC;
```

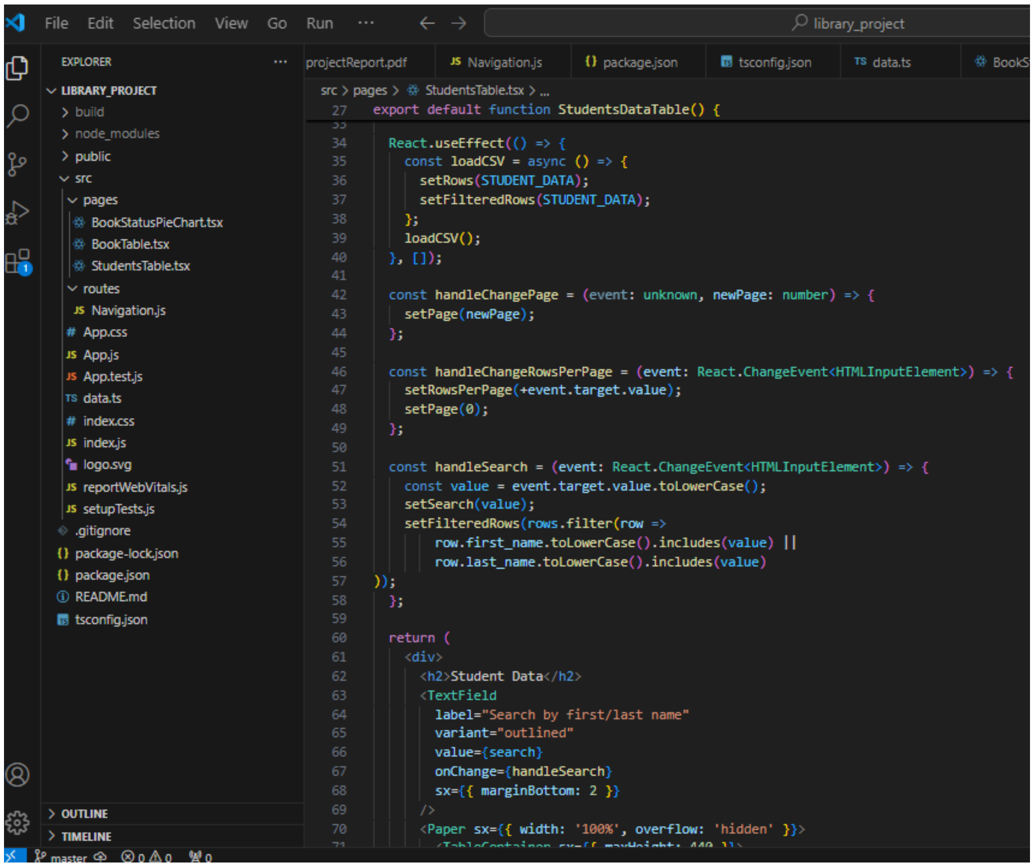
book_id	isbn	title	rating
78	9781910695432	Flights	2

```
24  20:12:55  SELECT book_id, b.isbn, b.title, rating FROM book_rating br, bo...  1 row(s) returned  0.00060 sec / 0.000...
```

- Sequelize Tables and Models: These are models to import the codes from MySQL database
  - Student Table:

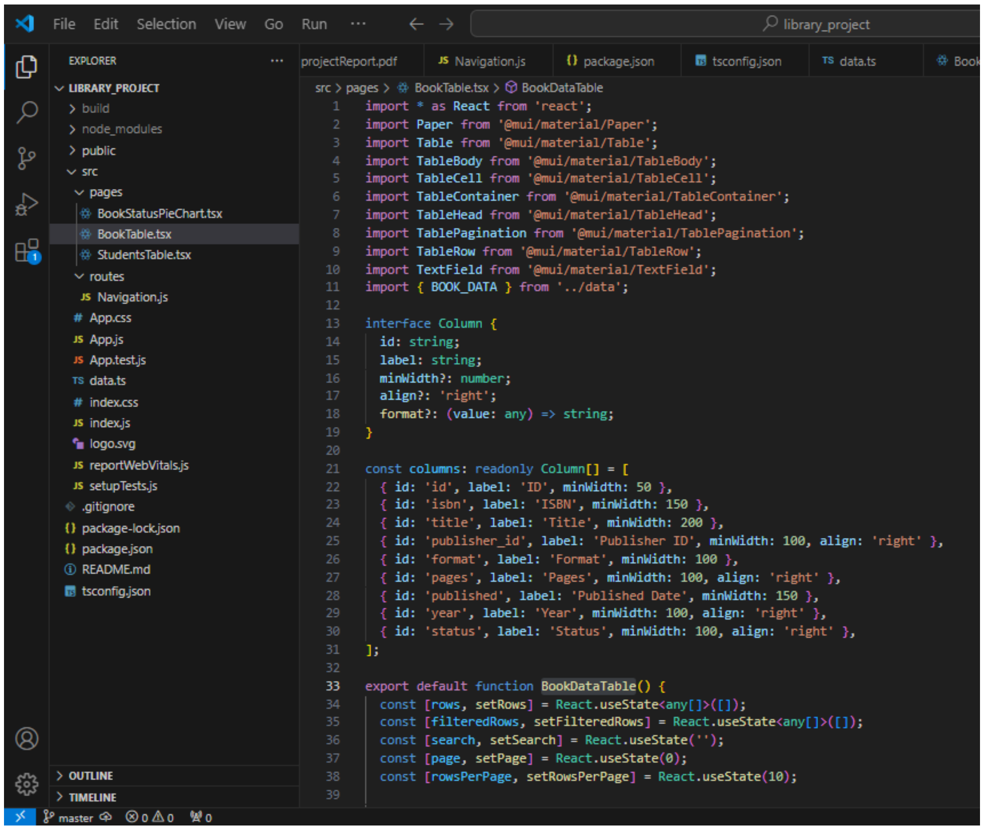


```tsx
import * as React from 'react';
import Paper from '@mui/material/Paper';
import Table from '@mui/material/Table';
import TableBody from '@mui/material/TableBody';
import TableCell from '@mui/material/TableCell';
import TableContainer from '@mui/material/TableContainer';
import TableHead from '@mui/material/TableHead';
import TablePagination from '@mui/material/TablePagination';
import TableRow from '@mui/material/TableRow';
import TextField from '@mui/material/TextField';
import {STUDENT_DATA } from '../data';

interface Column {
  id: string;
  label: string;
  minWidth?: number;
  align?: 'right';
  format?: (value: any) => string;
}

const columns: readonly Column[] = [
  { id: 'id', label: 'ID', minWidth: 50 },
  { id: 'first_name', label: 'First Name', minWidth: 150 },
  { id: 'last_name', label: 'Last Name', minWidth: 200 },
];

export default function StudentsDataTable() {
  const [rows, setRows] = React.useState<any[]>([]);
  const [filteredRows, setFilteredRows] = React.useState<any[]>([]);
  const [search, setSearch] = React.useState('');
  const [page, setPage] = React.useState(0);
  const [rowsPerPage, setRowsPerPage] = React.useState(10);
```

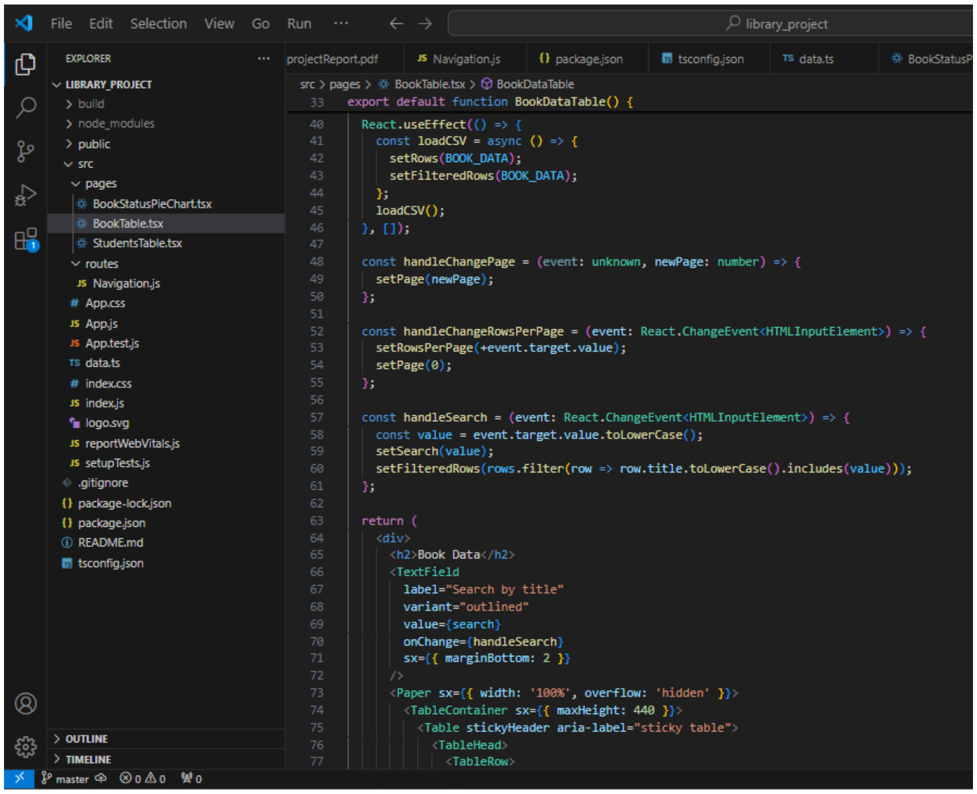


```tsx
export default function StudentsDataTable() {

  React.useEffect(() => {
    const loadCSV = async () => {
      setRows(STUDENT_DATA);
      setFilteredRows(STUDENT_DATA);
    };
    loadCSV();
  }, []);

  const handleChangePage = (event: unknown, newPage: number) => {
    setPage(newPage);
  };

  const handleChangeRowsPerPage = (event: React.ChangeEvent<HTMLInputElement>) => {
    setRowsPerPage(+event.target.value);
    setPage(0);
  };

  const handleSearch = (event: React.ChangeEvent<HTMLInputElement>) => {
    const value = event.target.value.toLowerCase();
    setSearch(value);
    setFilteredRows(rows.filter(row =>
        row.first_name.toLowerCase().includes(value) ||
        row.last_name.toLowerCase().includes(value)
));
  };

  return (
    <div>
      <h2>Student Data</h2>
      <TextField
        label="Search by first/last name"
        variant="outlined"
        value={search}
        onChange={handleSearch}
        sx={{ marginBottom: 2 }}
      />
      <Paper sx={{ width: '100%', overflow: 'hidden' }}>
```

- Book Table:

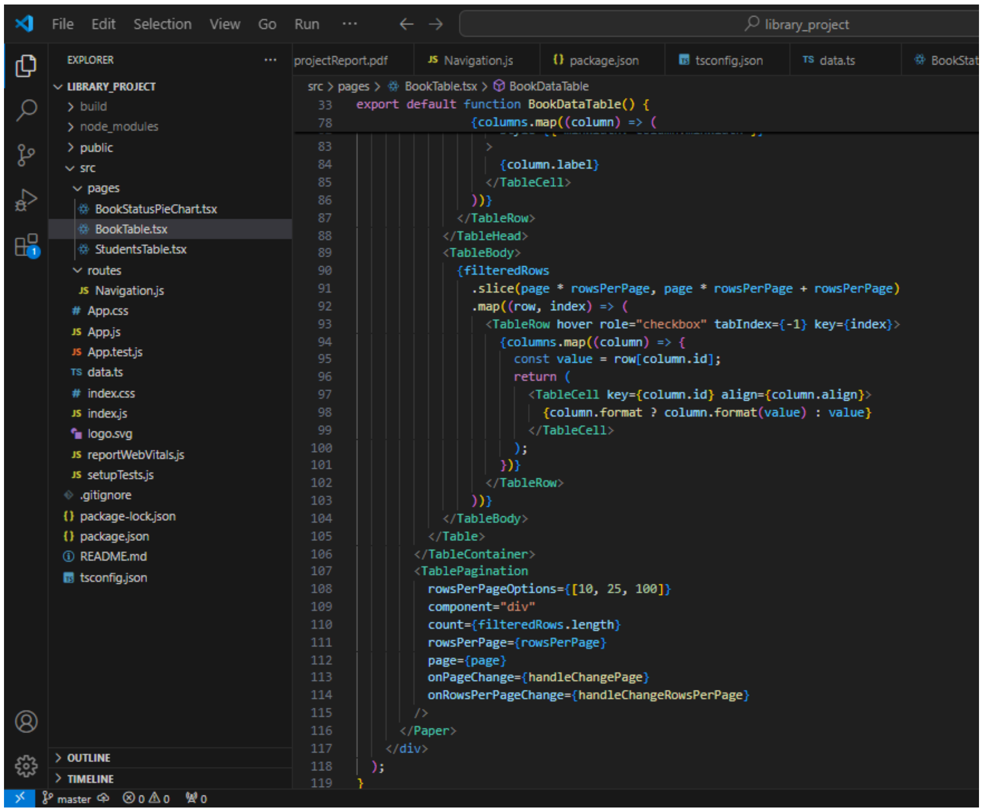


```tsx
import * as React from 'react';
import Paper from '@mui/material/Paper';
import Table from '@mui/material/Table';
import TableBody from '@mui/material/TableBody';
import TableCell from '@mui/material/TableCell';
import TableContainer from '@mui/material/TableContainer';
import TableHead from '@mui/material/TableHead';
import TablePagination from '@mui/material/TablePagination';
import TableRow from '@mui/material/TableRow';
import TextField from '@mui/material/TextField';
import { BOOK_DATA } from '../data';

interface Column {
  id: string;
  label: string;
  minWidth?: number;
  align?: 'right';
  format?: (value: any) => string;
}

const columns: readonly Column[] = [
  { id: 'id', label: 'ID', minWidth: 50 },
  { id: 'isbn', label: 'ISBN', minWidth: 150 },
  { id: 'title', label: 'Title', minWidth: 200 },
  { id: 'publisher_id', label: 'Publisher ID', minWidth: 100, align: 'right' },
  { id: 'format', label: 'Format', minWidth: 100 },
  { id: 'pages', label: 'Pages', minWidth: 100, align: 'right' },
  { id: 'published', label: 'Published Date', minWidth: 150 },
  { id: 'year', label: 'Year', minWidth: 100, align: 'right' },
  { id: 'status', label: 'Status', minWidth: 100, align: 'right' },
];

export default function BookDataTable() {
  const [rows, setRows] = React.useState<any[]>([]);
  const [filteredRows, setFilteredRows] = React.useState<any[]>([]);
  const [search, setSearch] = React.useState('');
  const [page, setPage] = React.useState(0);
  const [rowsPerPage, setRowsPerPage] = React.useState(10);

```



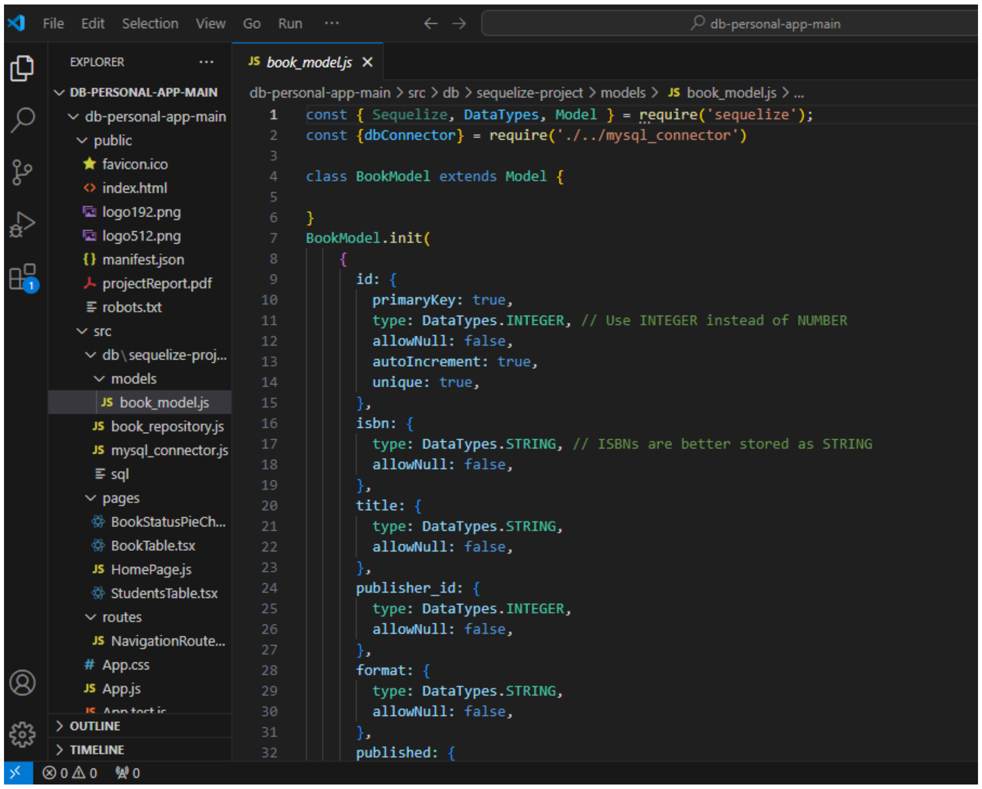
export default function BookDataTable() {

```tsx
  React.useEffect(() => {
    const loadCSV = async () => {
      setRows(BOOK_DATA);
      setFilteredRows(BOOK_DATA);
    };
    loadCSV();
  }, []);

  const handleChangePage = (event: unknown, newPage: number) => {
    setPage(newPage);
  };

  const handleChangeRowsPerPage = (event: React.ChangeEvent<HTMLInputElement>) => {
    setRowsPerPage(+event.target.value);
    setPage(0);
  };

  const handleSearch = (event: React.ChangeEvent<HTMLInputElement>) => {
    const value = event.target.value.toLowerCase();
    setSearch(value);
    setFilteredRows(rows.filter(row => row.title.toLowerCase().includes(value)));
  };

  return (
    <div>
      <h2>Book Data</h2>
      <TextField
        label="Search by title"
        variant="outlined"
        value={search}
        onChange={handleSearch}
        sx={{ marginBottom: 2 }}
      />
      <Paper sx={{ width: '100%', overflow: 'hidden' }}>
        <TableContainer sx={{ maxHeight: 440 }}>
          <Table stickyHeader aria-label="sticky table">
            <TableHead>
              <TableRow>
```

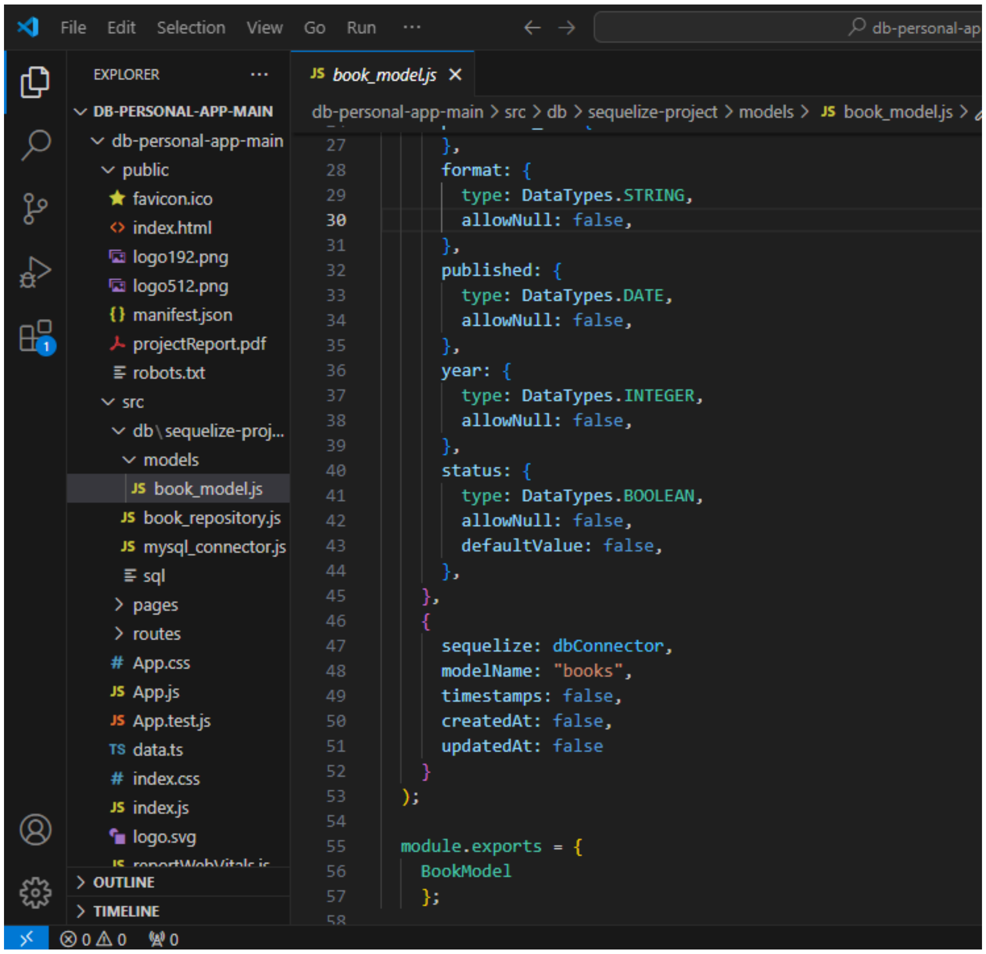


```
export default function BookDataTable() {
              {columns.map((column) => (
                >
                  {column.label}
                </TableCell>
              ))}
            </TableRow>
          </TableHead>
          <TableBody>
            {filteredRows
              .slice(page * rowsPerPage, page * rowsPerPage + rowsPerPage)
              .map((row, index) => (
                <TableRow hover role="checkbox" tabIndex={-1} key={index}>
                  {columns.map((column) => {
                    const value = row[column.id];
                    return (
                      <TableCell key={column.id} align={column.align}>
                        {column.format ? column.format(value) : value}
                      </TableCell>
                    );
                  })}
                </TableRow>
              ))}
          </TableBody>
        </Table>
      </TableContainer>
      <TablePagination
        rowsPerPageOptions={[10, 25, 100]}
        component="div"
        count={filteredRows.length}
        rowsPerPage={rowsPerPage}
        page={page}
        onPageChange={handleChangePage}
        onRowsPerPageChange={handleChangeRowsPerPage}
      />
    </Paper>
  </div>
);
}
```

- Book Model:

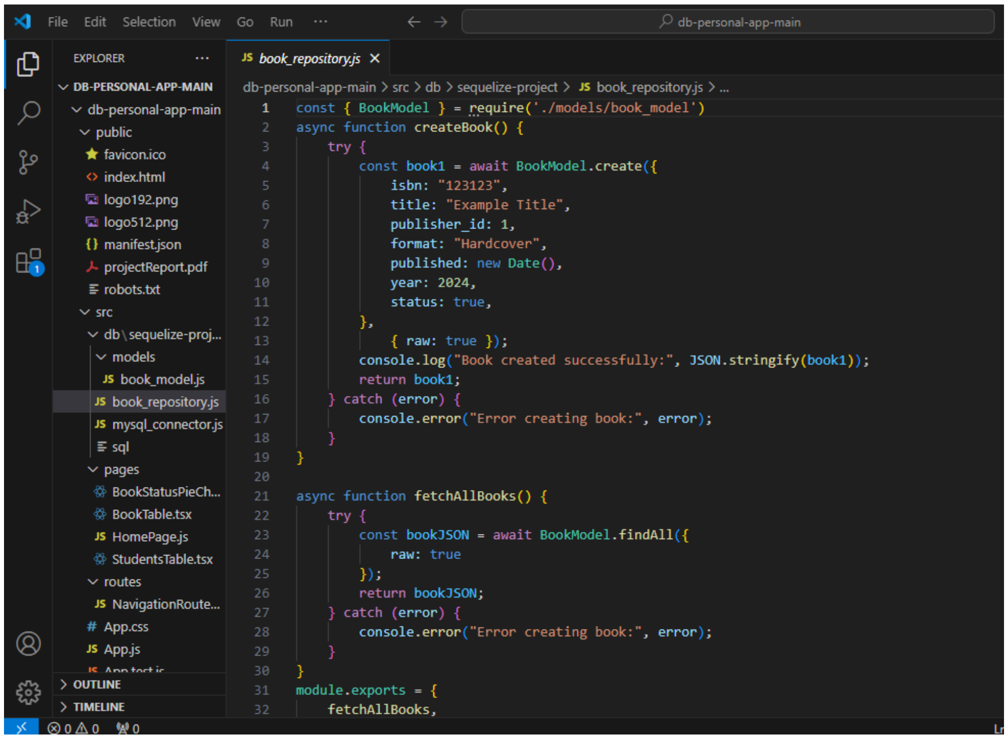


```
const { Sequelize, DataTypes, Model } = require('sequelize');
const {dbConnector} = require('./../mysql_connector')

class BookModel extends Model {

}
BookModel.init(
  {
    id: {
      primaryKey: true,
      type: DataTypes.INTEGER, // Use INTEGER instead of NUMBER
      allowNull: false,
      autoIncrement: true,
      unique: true,
    },
    isbn: {
      type: DataTypes.STRING, // ISBNs are better stored as STRING
      allowNull: false,
    },
    title: {
      type: DataTypes.STRING,
      allowNull: false,
    },
    publisher_id: {
      type: DataTypes.INTEGER,
      allowNull: false,
    },
    format: {
      type: DataTypes.STRING,
      allowNull: false,
    },
    published: {
```



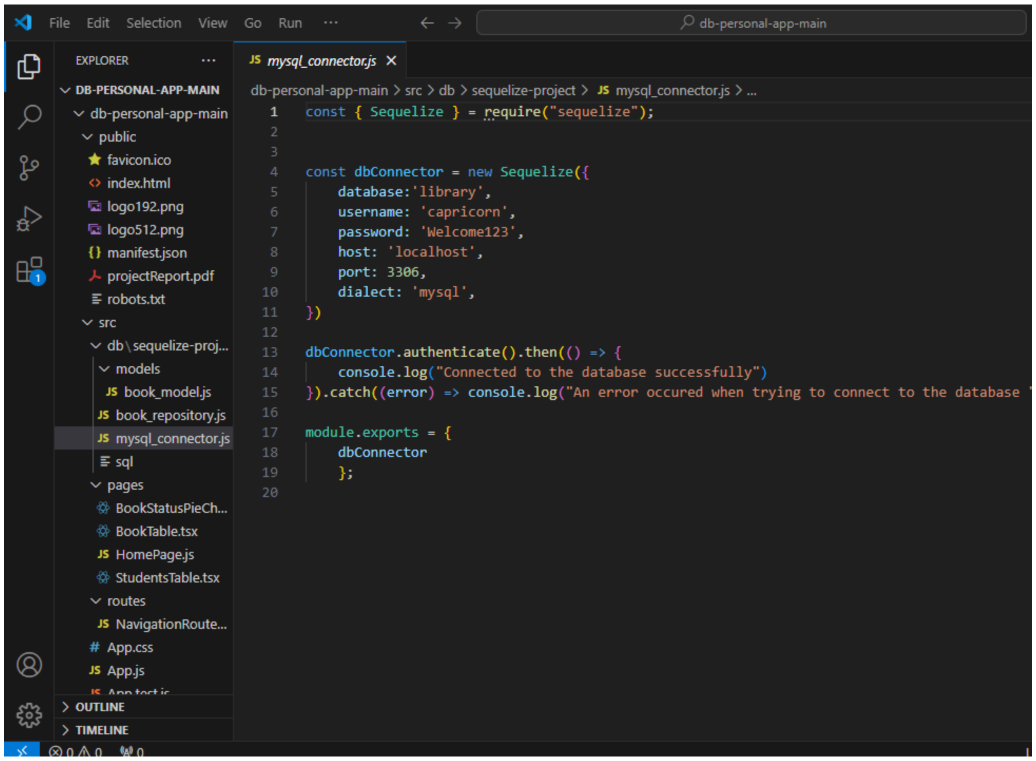
```
      },
      format: {
        type: DataTypes.STRING,
        allowNull: false,
      },
      published: {
        type: DataTypes.DATE,
        allowNull: false,
      },
      year: {
        type: DataTypes.INTEGER,
        allowNull: false,
      },
      status: {
        type: DataTypes.BOOLEAN,
        allowNull: false,
        defaultValue: false,
      },
    },
    {
      sequelize: dbConnector,
      modelName: "books",
      timestamps: false,
      createdAt: false,
      updatedAt: false
    }
  );

  module.exports = {
    BookModel
  };
```

- Book Repository Model:


File Edit Selection View Go Run
db-personal-app-main
EXPLORER
book_repository.js
db-personal-app-main > src > db > sequelize-project > book_repository.js > ...
OUTLINE
TIMELINE

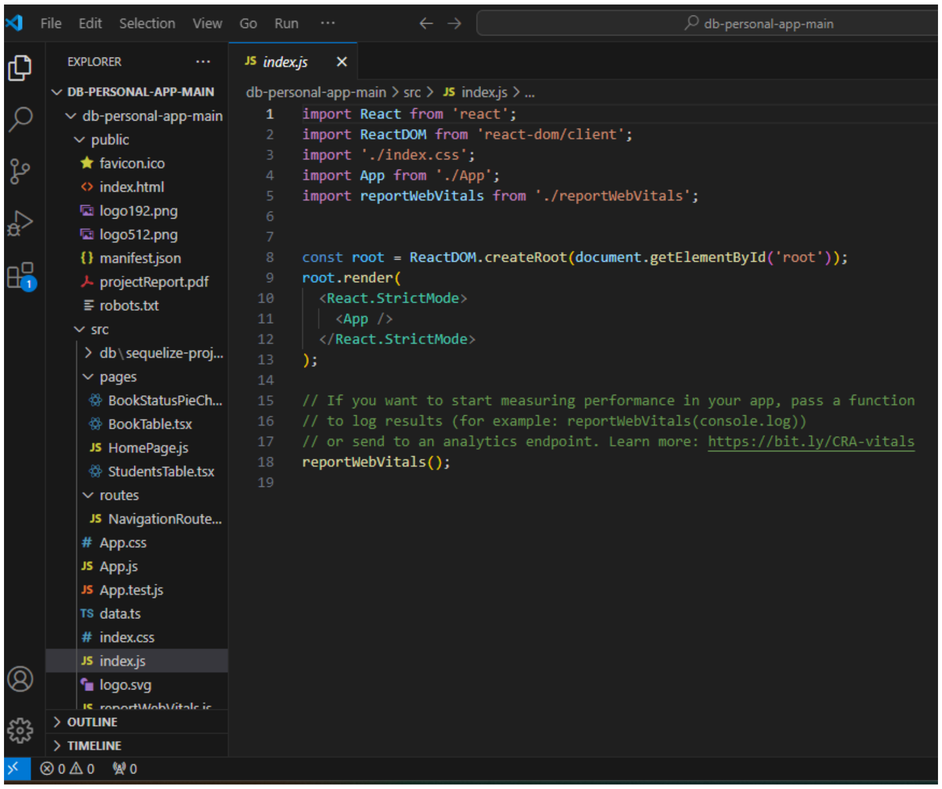
```javascript
const { BookModel } = require('./models/book_model')
async function createBook() {
    try {
        const book1 = await BookModel.create({
            isbn: "123123",
            title: "Example Title",
            publisher_id: 1,
            format: "Hardcover",
            published: new Date(),
            year: 2024,
            status: true,
        },
            { raw: true });
        console.log("Book created successfully:", JSON.stringify(book1));
        return book1;
    } catch (error) {
        console.error("Error creating book:", error);
    }
}

async function fetchAllBooks() {
    try {
        const bookJSON = await BookModel.findAll({
            raw: true
        });
        return bookJSON;
    } catch (error) {
        console.error("Error creating book:", error);
    }
}
module.exports = {
    fetchAllBooks,
```

- Database Connector:


File Edit Selection View Go Run
db-personal-app-main
EXPLORER
mysql_connector.js
db-personal-app-main > src > db > sequelize-project > mysql_connector.js > ...
OUTLINE
TIMELINE

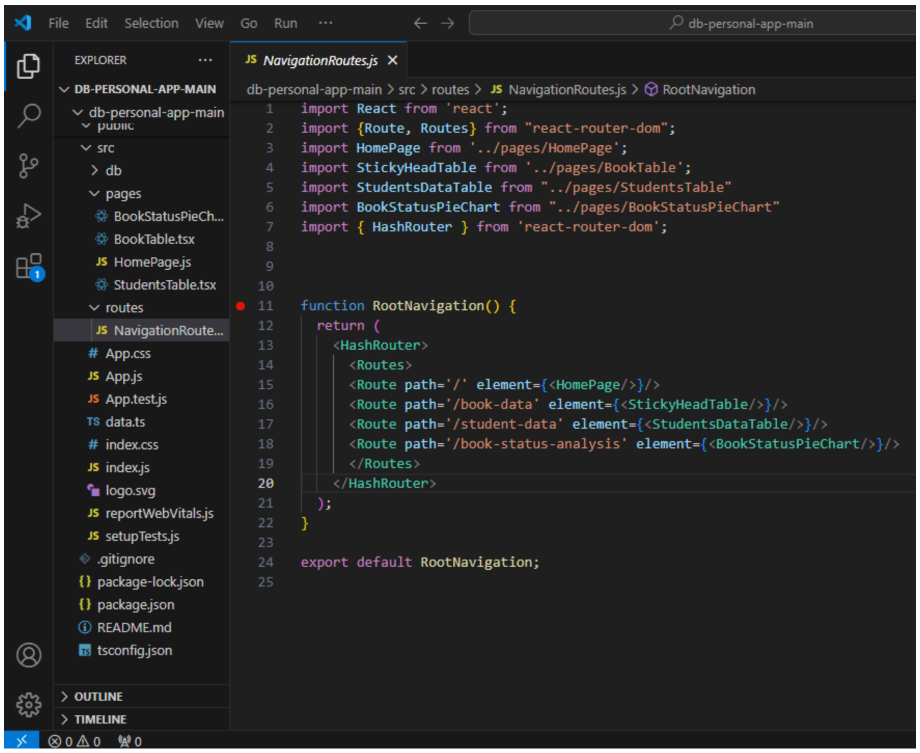
```javascript
const { Sequelize } = require("sequelize");


const dbConnector = new Sequelize({
    database:'library',
    username: 'capricorn',
    password: 'Welcome123',
    host: 'localhost',
    port: 3306,
    dialect: 'mysql',
})

dbConnector.authenticate().then(() => {
    console.log("Connected to the database successfully")
}).catch((error) => console.log("An error occured when trying to connect to the database "

module.exports = {
    dbConnector
    };
```

- Routers: These were used to connect the front-end our backend database
  - React: The front end was built with React



```js
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';


const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

- HashRoute: HashRoute was imported for connectivity with GitHub

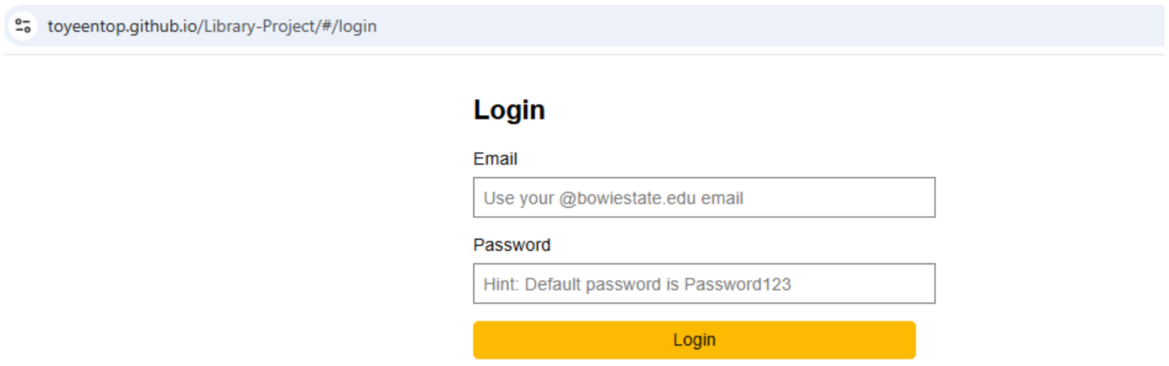


```js
import React from 'react';
import {Route, Routes} from "react-router-dom";
import HomePage from '../pages/HomePage';
import StickyHeadTable from '../pages/BookTable';
import StudentsDataTable from "../pages/StudentsTable"
import BookStatusPieChart from "../pages/BookStatusPieChart"
import { HashRouter } from 'react-router-dom';



function RootNavigation() {
  return (
    <HashRouter>
      <Routes>
      <Route path='/' element={<HomePage/>}/>
      <Route path='/book-data' element={<StickyHeadTable/>}/>
      <Route path='/student-data' element={<StudentsDataTable/>}/>
      <Route path='/book-status-analysis' element={<BookStatusPieChart/>}/>
      </Routes>
    </HashRouter>
  );
}

export default RootNavigation;
```

5.2. Front-end (Webpage) Demonstration

5.2. Front-end Demonstration

Click the link https://toyeentop.github.io/Library-Project/#/login to access the front-end of the database through GitHub.


toyeentop.github.io/Library-Project/#/login

Login

Email

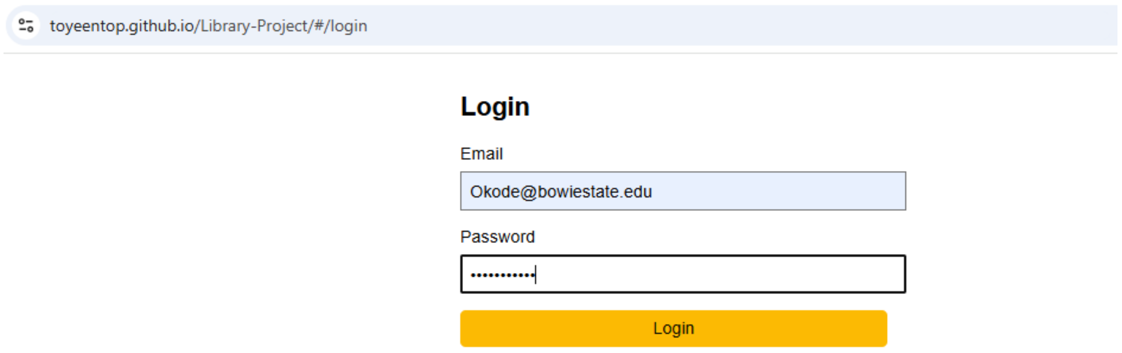
Use your @bowiestate.edu email

Password

Hint: Default password is Password123

Login

Enter email and password


toyeentop.github.io/Library-Project/#/login

Login

Email

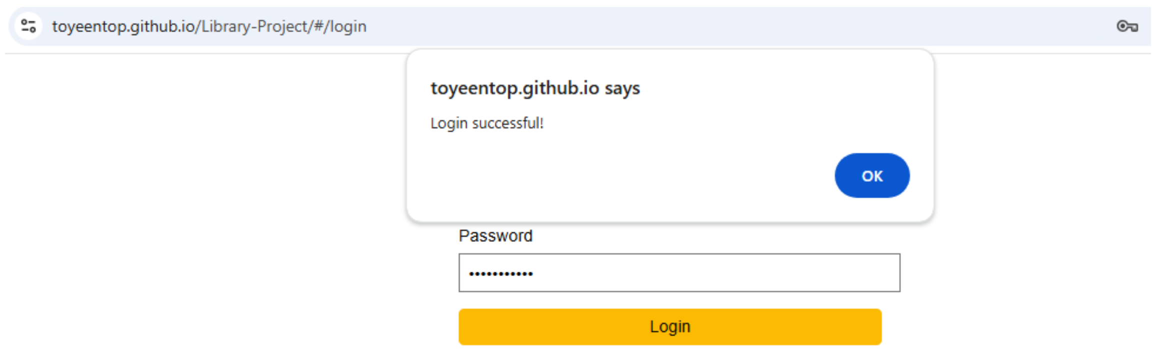
Okode@bowiestate.edu

Password

•••••••••••

Login

If the correct email and password are entered, access will be granted.

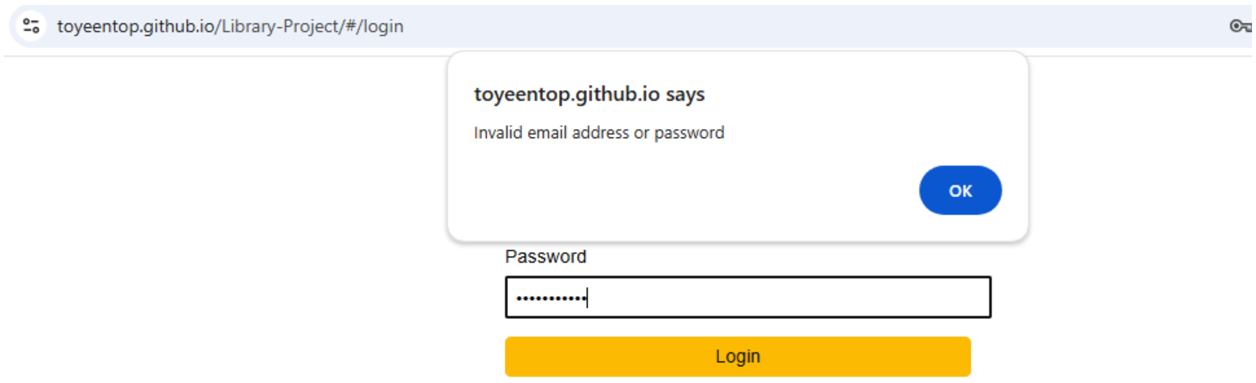


toyeentop.github.io/Library-Project/#/login

toyeentop.github.io says

Login successful!

Otherwise, access will not be granted.



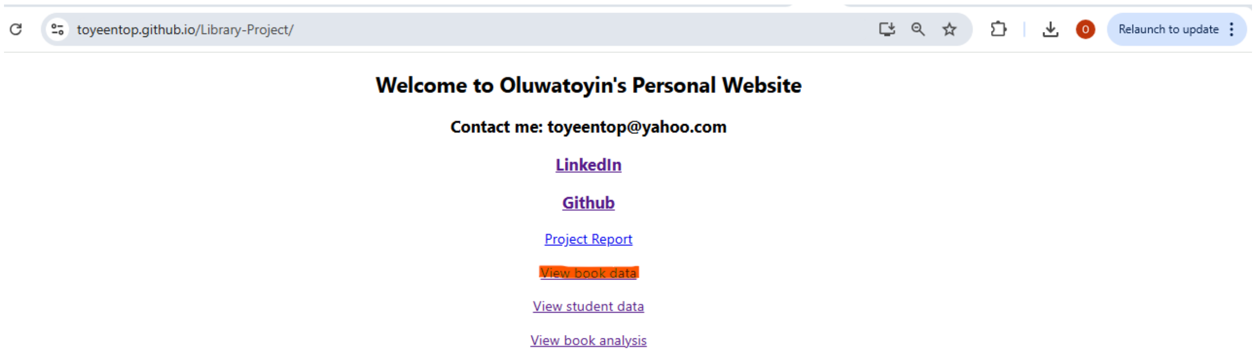
toyeentop.github.io/Library-Project/#/login

toyeentop.github.io says

Invalid email address or password

Once access is granted take any of the actions below to explore the page.

On the website, click on “View book data” to view the book data



toyeentop.github.io/Library-Project/

Relaunch to update

Welcome to Oluwatoyin's Personal Website

Contact me: toyeentop@yahoo.com

LinkedIn

Github

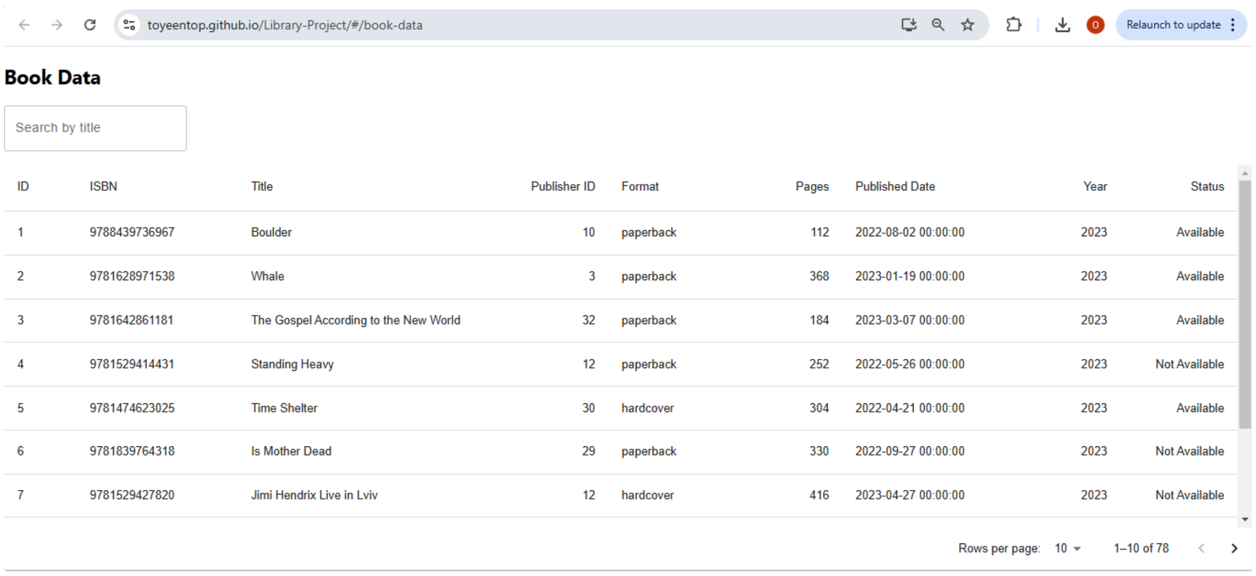
Project Report

View book data

View student data

View book analysis

Webpage displaying Book Data. The page displays 10 rows per page however, the settings can be modified to display 25 or 50 pages per page.



toyeentop.github.io/Library-Project/#/book-data

Relaunch to update

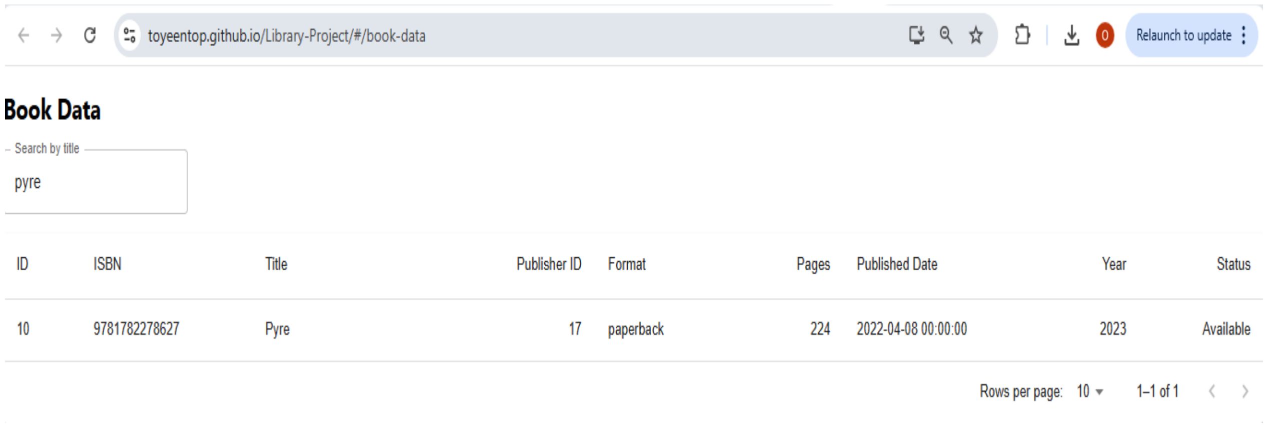
Book Data

Search by title

ID	ISBN	Title	Publisher ID	Format	Pages	Published Date	Year	Status
1	9788439736967	Boulder	10	paperback	112	2022-08-02 00:00:00	2023	Available
2	9781628971538	Whale	3	paperback	368	2023-01-19 00:00:00	2023	Available
3	9781642861181	The Gospel According to the New World	32	paperback	184	2023-03-07 00:00:00	2023	Available
4	9781529414431	Standing Heavy	12	paperback	252	2022-05-26 00:00:00	2023	Not Available
5	9781474623025	Time Shelter	30	hardcover	304	2022-04-21 00:00:00	2023	Available
6	9781839764318	Is Mother Dead	29	paperback	330	2022-09-27 00:00:00	2023	Not Available
7	9781529427820	Jimi Hendrix Live in Lviv	12	hardcover	416	2023-04-27 00:00:00	2023	Not Available

Rows per page: 10 1–10 of 78

Search by Title – for example Pyre, or gospel – as you begin to type, it begins to filter out your search

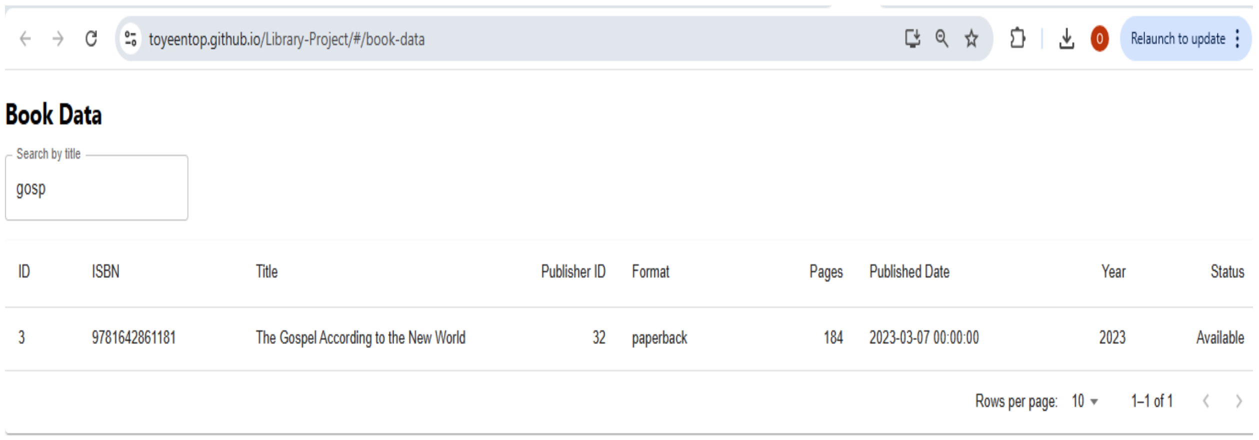


Book Data

Search by title: pyre

ID	ISBN	Title	Publisher ID	Format	Pages	Published Date	Year	Status
10	9781782278627	Pyre	17	paperback	224	2022-04-08 00:00:00	2023	Available

Rows per page: 10 1–1 of 1



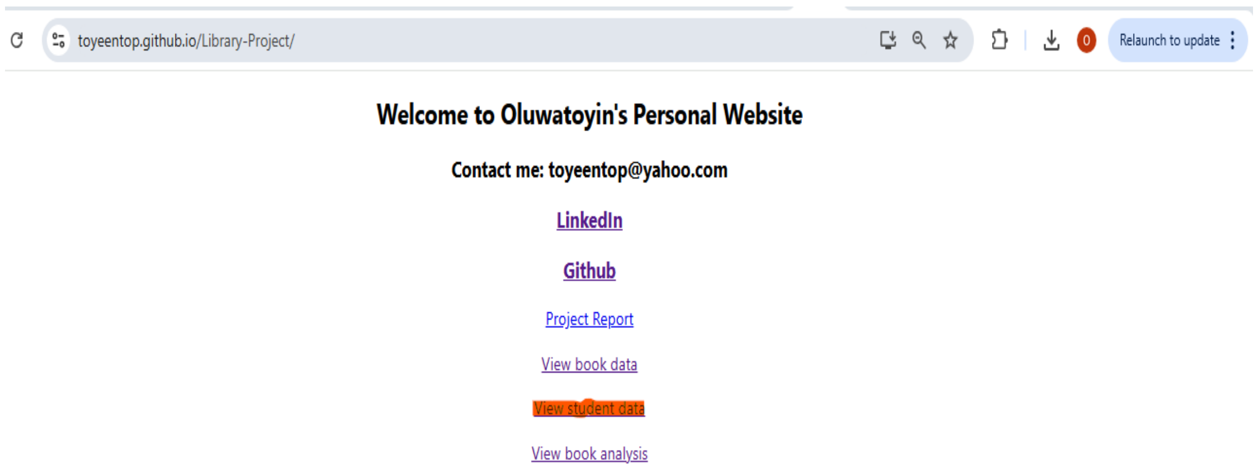
Book Data

Search by title: gosp

ID	ISBN	Title	Publisher ID	Format	Pages	Published Date	Year	Status
3	9781642861181	The Gospel According to the New World	32	paperback	184	2023-03-07 00:00:00	2023	Available

Rows per page: 10 1–1 of 1

On the website, Click on "View student data" to view student data.



Welcome to Oluwatoyin's Personal Website

Contact me: toyeentop@yahoo.com

LinkedIn

Github

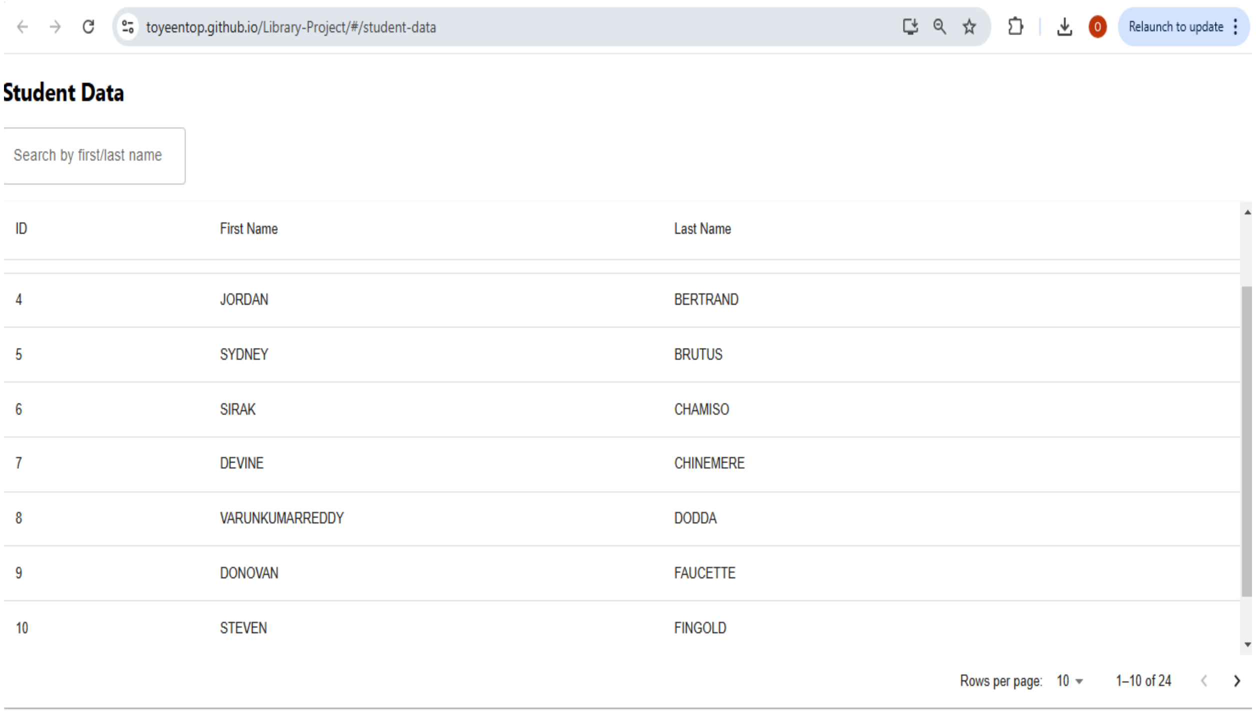
Project Report

View book data

View student data

View book analysis

Webpage displaying Student Data. The page displays 10 rows per page however, the settings can be modified to display 25 or 50 pages per page.



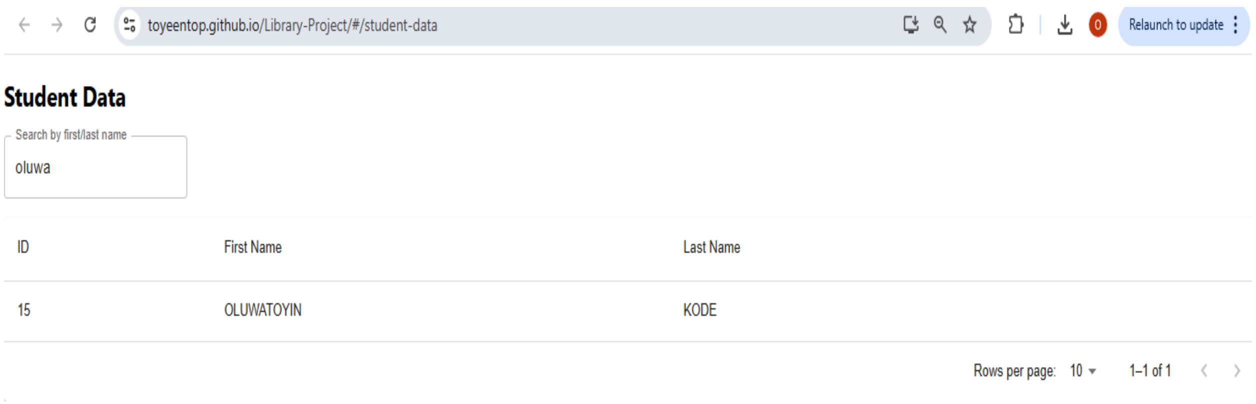
Student Data

Search by first/last name

ID	First Name	Last Name
4	JORDAN	BERTRAND
5	SYDNEY	BRUTUS
6	SIRAK	CHAMISO
7	DEVINE	CHINEMERE
8	VARUNKUMARREDDY	DODDA
9	DONOVAN	FAUCETTE
10	STEVEN	FINGOLD

Rows per page: 10 1–10 of 24

Users can search student data by first or last name. For example as you begin to type “oluwa” for Oluwatoyin and the name is displayed or “Pr” for Precious, and Precious is displayed.

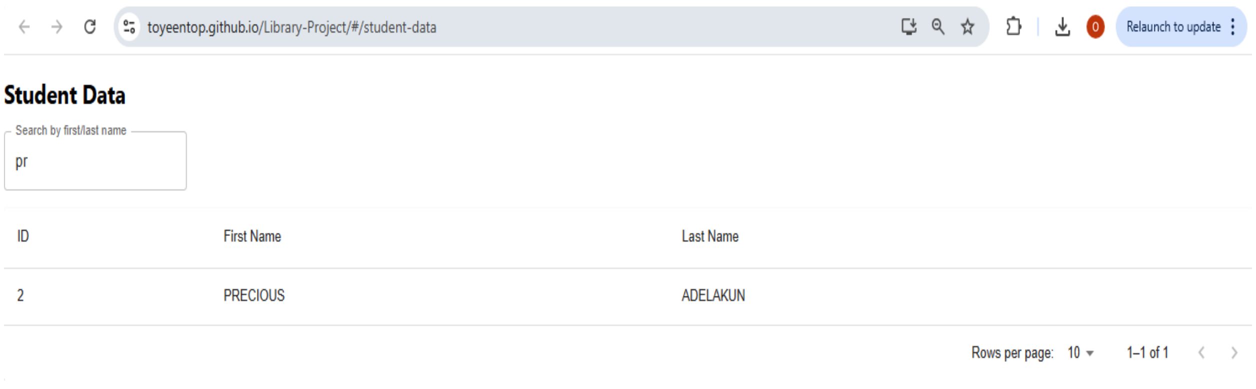


Student Data

Search by first/last name: oluwa

ID	First Name	Last Name
15	OLUWATOYIN	KODE

Rows per page: 10 1–1 of 1



toyeentop.github.io/Library-Project/#/student-data

Relaunch to update

Student Data

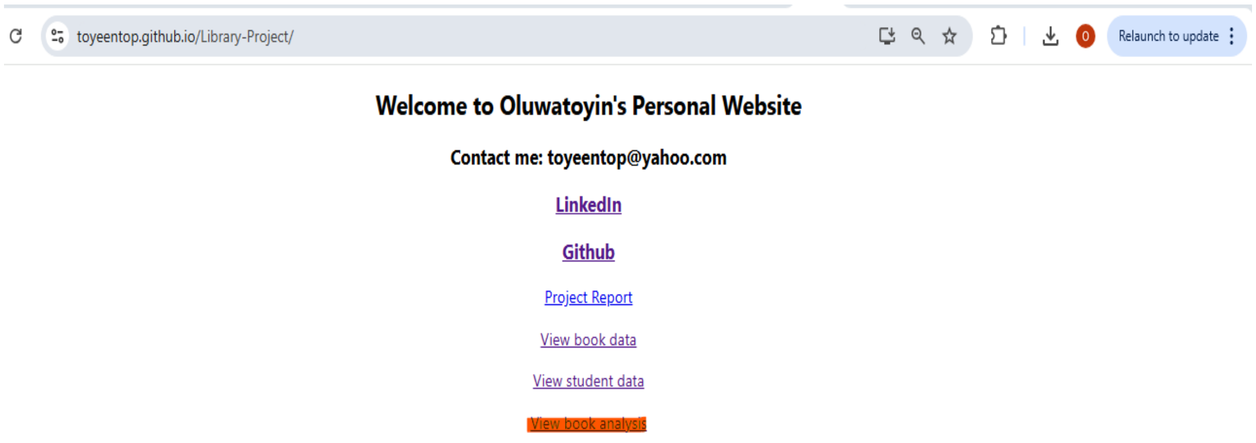
Search by first/last name

pr

ID	First Name	Last Name
2	PRECIOUS	ADELAKUN

Rows per page: 10 1–1 of 1

On the webpage, Click on “View book analysis” to view the data analysis of book availability.



toyeentop.github.io/Library-Project/

Relaunch to update

Welcome to Oluwatoyin's Personal Website

Contact me: toyeentop@yahoo.com

LinkedIn

Github

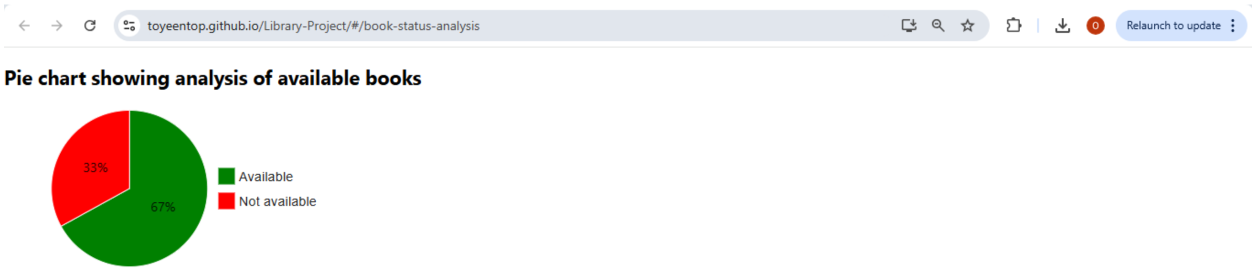
Project Report

View book data

View student data

View book analysis

The pie chart shows a quick view of the percentage of books that are available and unavailable.

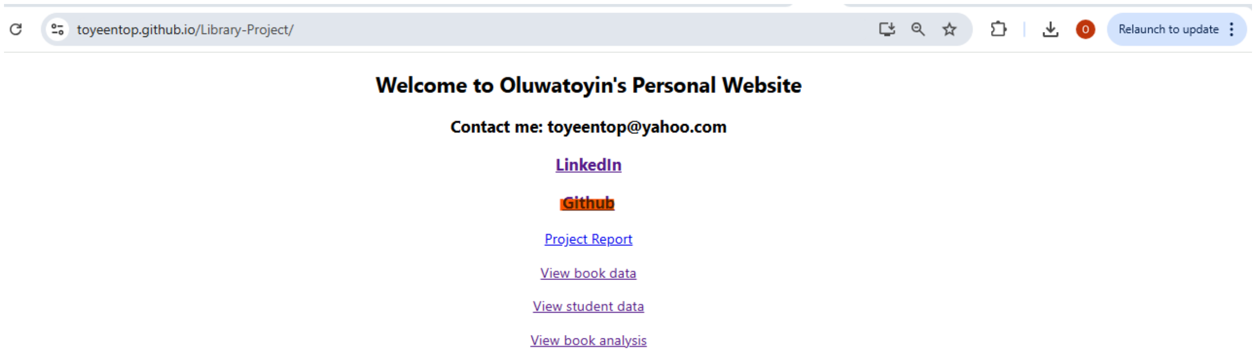


toyeentop.github.io/Library-Project/#/book-status-analysis

Relaunch to update

Pie chart showing analysis of available books

On the website, Click on “GitHub” to view the GitHub Page.


toyeentop.github.io/Library-Project/

Welcome to Oluwatoyin's Personal Website

Contact me: toyeentop@yahoo.com

LinkedIn

Github

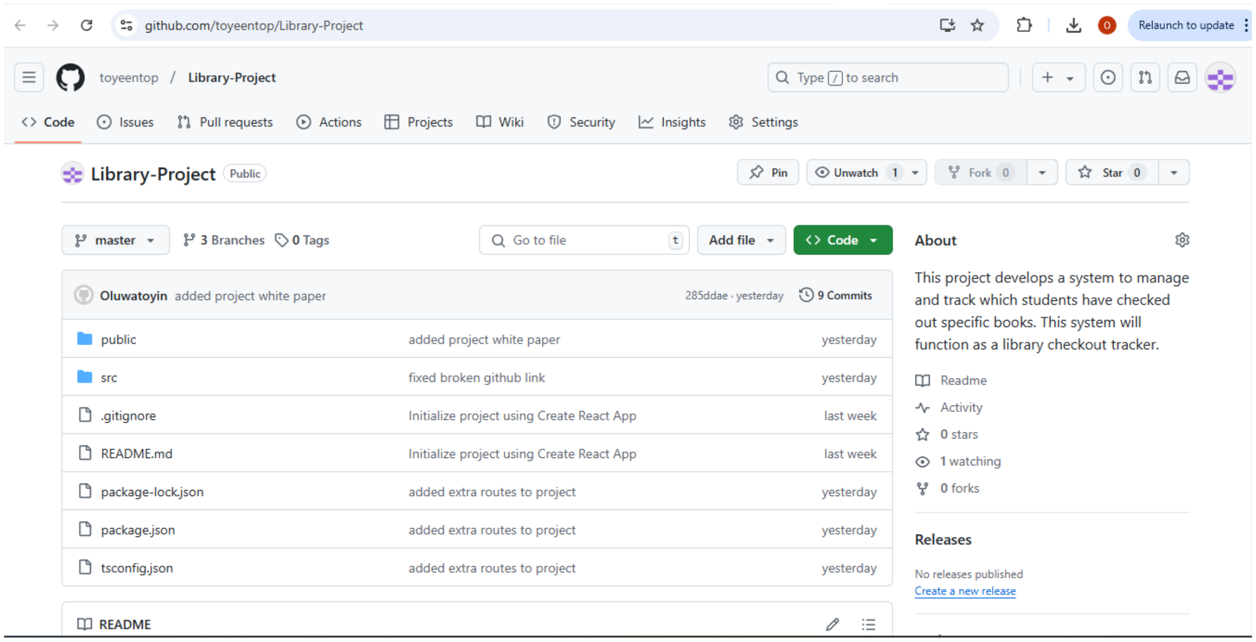
Project Report

View book data

View student data

View book analysis

Website displaying GitHub Page with all the project packages and codes


github.com/toyeentop/Library-Project

toyeentop / Library-Project

Library-Project Public

master · 3 Branches · 0 Tags

Oluwatoyin added project white paper · 285ddae · yesterday · 9 Commits

public	added project white paper	yesterday
src	fixed broken github link	yesterday
.gitignore	Initialize project using Create React App	last week
README.md	Initialize project using Create React App	last week
package-lock.json	added extra routes to project	yesterday
package.json	added extra routes to project	yesterday
tsconfig.json	added extra routes to project	yesterday

About

This project develops a system to manage and track which students have checked out specific books. This system will function as a library checkout tracker.

Readme · Activity · 0 stars · 1 watching · 0 forks

Releases

No releases published

Create a new release

README

5.3. Tools and Technologies

- Programming Language:
  - Java: Chosen for its platform independence, strong database connectivity, and robust features, Java was used to build the core application logic for interactive database management.
- Backend Framework:
  - Node.js: Utilized for its lightweight, event-driven architecture, Node.js efficiently handled concurrent requests and facilitated dynamic interactions between the client and the database.
- ORM (Object-Relational Mapping):
  - Sequelize: This ORM simplified database operations with MySQL, providing an abstraction layer over raw SQL queries. This reduced development time, enabled clean code, and improved maintainability.

- Database:
    - MySQL: Selected for its reliability, scalability, and support for SQL standards, MySQL efficiently managed the relational data needed for the project. Its robust querying capabilities allowed seamless handling of borrowing and returning operations.
- Web Hosting:
    - GitHub Pages: Used to host project documentation and static resources, ensuring easy public access, version control, and collaborative development.
- Frontend Framework:
    - React with HashRouter: React enabled the development of a dynamic and responsive user interface, while HashRouter provided seamless navigation with GitHub.

6. System Features

6.1. Borrowing Management

- Specify the name of the student.
- Display borrowed books, borrower details, and return dates.
- Dynamically update the borrowing and return status.

6.2. Rating Analysis

- Show books rated higher or lower than the average rating.
- Allow users to input new ratings to update average dynamically.

6.3. Group Management

- Display students in groups (A, B, C, D).
- Show books borrowed by students in each group.
- Allow querying of a student's group based on their name.

7. Conclusion and Recommendations

The Library Book Borrowing Management System successfully manages book inventory, student assignments, and borrowing activities while enabling meaningful queries and reports. Future improvements could include automated notifications for due dates and overdue penalties.

Appendices

Source Code Repository Link: https://github.com/toyeentop/Library-Project