

Library Management System for Group-Based Student Borrowing

Adedeji Oludayo

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:

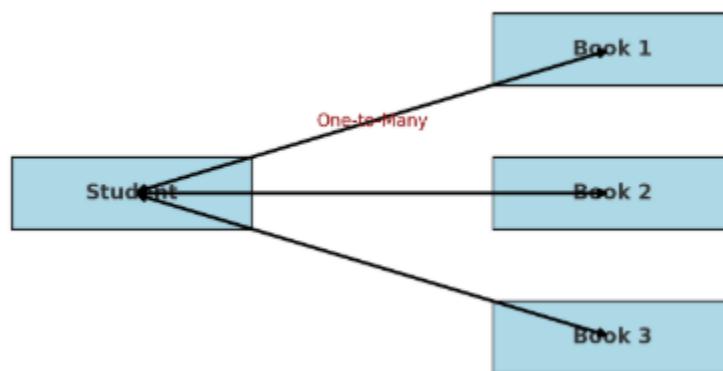
- A relational database to manage books, students, borrowing transactions, and group memberships.
- A user interface (UI) for querying borrowing details and viewing group information.
- A backend system to calculate average ratings and display books above or below the average.
- A method to dynamically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

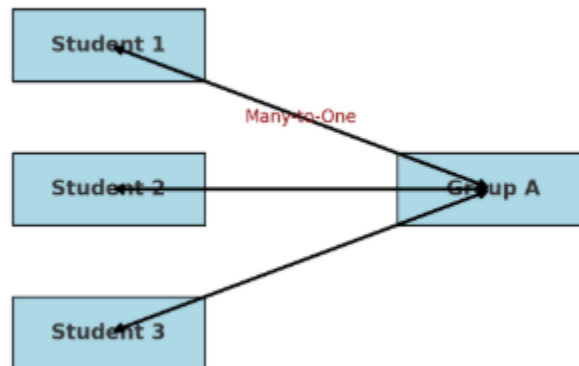
5. Design and Methodologies

5.1. Database Design

- Entities:
 - Books (BookID, Title, Author, Rating, BorrowedStatus)
 - Students (StudentID, FirstName, LastName, Group, PreferredBooks)
 - Borrowing (TransactionID, BookID, StudentID, BorrowDate, ReturnDate)
 - 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;

id	isbn	title	publisher_id	format	pages	published	year	status
1	978009706987	Booklet	10	paperback	112	2003-04-02 00:00:00	2003	1
2	978028071133	Wings	7	paperback	368	2003-07-12 00:00:00	2003	1
3	978104280181	The Gospel According to the New World	32	paperback	184	2003-03-07 00:00:00	2003	1
4	978107460320	Time Shifter	36	hardcover	304	2003-04-21 00:00:00	2003	1
5	978194544040	The Birthday Party	27	paperback	484	2003-01-04 00:00:00	2003	1
6	978176270801	Yes	17	paperback	324	2003-04-08 00:00:00	2003	1
7	978101300780	Sail Down	5	paperback	200	2002-06-22 00:00:00	2003	1
8	978101300783	Paradise	9	paperback	118	2002-03-23 00:00:00	2002	1
9	978100480051	Heaven	15	paperback	140	2002-05-12 00:00:00	2002	1
10	978101126460	Love in the Big City	28	paperback	317	2001-10-28 00:00:00	2002	1
11	978101126460	Heavy Stones, Mostly	28	paperback	183	2001-12-02 00:00:00	2002	1
12	978100480051	The Book of Mother	21	paperback	224	2001-09-19 00:00:00	2002	1
13	978176270801	More Than I Love My Life	9	paperback	368	2001-08-26 00:00:00	2002	1
14	978101300510	Hardcore	11	paperback	322	2003-01-01 00:00:00	2002	1
15	978101300772	A New Name: Baptology V-VII	5	paperback	228	2001-09-08 00:00:00	2002	1
16	978101126461	Tomb of David	28	paperback	178	2001-08-12 00:00:00	2002	1
17	978101060053	The Books of Jacob	5	paperback	240	2001-11-15 00:00:00	2002	1
18	978101027782	Conrad Burns	35	paperback	251	2001-06-15 00:00:00	2002	1
19	978100480051	The War of the Pear	18	hardcover	95	2001-01-07 00:00:00	2001	1
20	978176270801	When We Cease to Understand the W...	17	hardcover	180	2000-04-03 00:00:00	2001	1
21	978100480051	An Inventory of Losses	12	hardcover	296	2000-08-20 00:00:00	2001	1
22	978100480051	At Night All Wood is Black	17	hardcover	180	2000-11-03 00:00:00	2001	1
23	978000001430	I Live in the Status	48	hardcover	144	2000-03-18 00:00:00	2001	1
24	978101300772	Minor Detail	5	paperback	144	2000-05-08 00:00:00	2001	1
25	978101300772	Summer Brother	32	paperback	285	2001-04-08 00:00:00	2001	1
26	978100480051	The Pear Field	14	paperback	140	2000-10-30 00:00:00	2001	1
27	978000001430	The Perfect Rule	25	hardcover	140	2000-12-04 00:00:00	2001	1
28	978000134004	The Discomfort of Evening	4	paperback	240	2000-03-05 00:00:00	2000	1
29	978100480051	The Enlightenment of The Greenback...	13	paperback	328	2000-01-17 00:00:00	2000	1

SELECT * FROM books WHERE status = 0;

id	isbn	title	publisher_id	format	pages	published	year	status
4	978103041400	Standing Heavy	12	paperback	282	2003-05-26 00:00:00	2003	0
5	978103041401	So Many Good	28	paperback	343	2003-09-27 00:00:00	2003	0
6	978103041402	Jim Hendrix Live in Live	12	paperback	419	2003-04-27 00:00:00	2003	0
7	978103041403	White We Were Dreaming	5	paperback	568	2003-03-30 00:00:00	2003	0
8	978103041404	A System So Magnificent It Is Bending	20	paperback	544	2003-03-30 00:00:00	2003	0
9	978103041405	North Building	8	paperback	212	2003-03-10 00:00:00	2003	0
10	978103041406	Evans Knows	2	paperback	143	2001-07-01 00:00:00	2002	0
11	978103041407	After the Sex	11	paperback	118	2001-08-24 00:00:00	2002	0
12	978103041408	Witchcraft	1	paperback	178	2000-08-02 00:00:00	2001	0
13	978103041409	In Memory of Memory	3	paperback	241	2001-03-07 00:00:00	2001	0
14	978103041410	The Gargles of Smoking in Red	6	paperback	187	2001-04-01 00:00:00	2001	0
15	978103041411	The Employees	11	paperback	136	2000-10-01 00:00:00	2001	0
16	978103041412	The Adventures of China Iron	2	paperback	250	2000-11-14 00:00:00	2000	0
17	978103041413	The Light Life	20	hardcover	244	2000-11-04 00:00:00	2000	0
18	978103041414	The Other Names: Baptology I-II	5	paperback	381	2000-10-10 00:00:00	2000	0
19	978103041415	Sermon	30	hardcover	350	2000-09-26 00:00:00	2000	0
20	978103041416	The Remembrance	1	paperback	180	2000-10-24 00:00:00	2000	0
21	978103041417	The News	5	paperback	233	2000-05-20 00:00:00	2000	0
22	978103041418	The Shaken Blythe	24	paperback	418	2000-08-26 00:00:00	2000	0
23	978103041419	The White Book	18	paperback	181	2000-04-25 00:00:00	2000	0
24	978103041420	The World Does On	28	paperback	320	2000-05-31 00:00:00	2000	0
25	978103041421	Let My Love	2	paperback	133	2001-03-14 00:00:00	2000	0

- 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;

```

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' tree lists various database objects including 'library' and 'groups'. The main area displays a SQL query in a text editor:

```

1 use library;
2
3 -- Groups
4 SELECT + FROM student_group;
5
6 SELECT count(sg.student_id) as no_of_students_in_group, g.name FROM student_group sg, library.groups g
7 WHERE sg.group_id = g.id GROUP BY g.name;
8

```

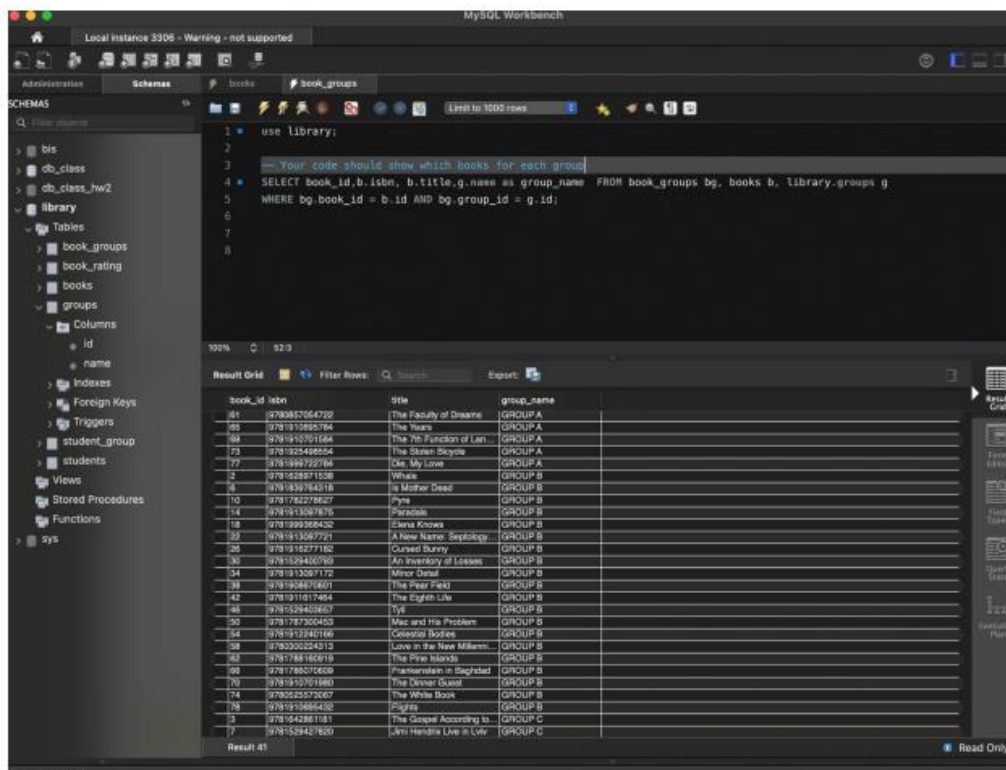
Below the query editor, a 'Result Grid' shows the execution results. The grid has two columns: 'no_of_students_in_group' and 'name'. The results are as follows:

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

At the bottom, an 'Action Output' section displays performance metrics for each step of the query execution:

Duration / Fetch Time
0.0037 sec / 0.00000...
0.0031 sec / 0.00000...
0.0037 sec / 0.00000...
0.0016 sec / 0.00000...
0.0017 sec / 0.00001...
0.0014 sec / 0.00000...
0.0014 sec / 0.00000...

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;



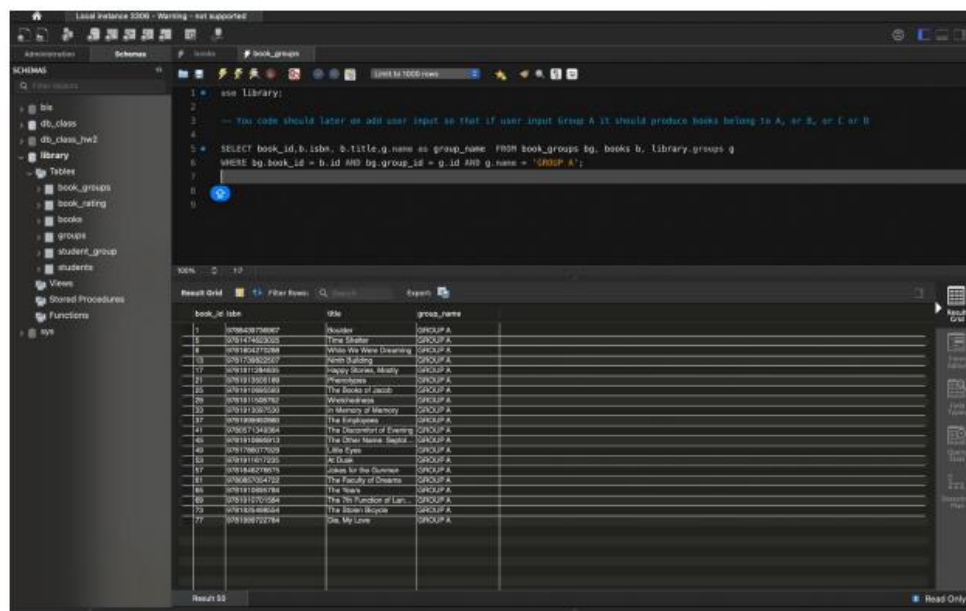
The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists the 'library' database. The 'Query Editor' contains the following SQL query:

```
1 use library;
2
3 -- Your code should show which books for each group
4 SELECT book_id,b.isbn, b.title,g.name as group_name FROM book_groups bg, books b, library.groups g
5 WHERE bg.book_id = b.id AND bg.group_id = g.id;
```

The 'Result Grid' displays the following data:

book_id	isbn	title	group_name
81	0290857064729	The Faculty of Dreams	GROUP A
85	0781010895784	The Years	GROUP A
89	0781010761664	The 7th Function of Lan...	GROUP A
73	0781025486854	The Seven Beyond	GROUP A
77	0781998792784	Oh, My Love	GROUP A
12	0781028871538	Whales	GROUP B
18	0781839764518	Is Mother Dead?	GROUP B
10	0781782278627	Pyre	GROUP B
14	0781873897615	Paradise	GROUP B
18	0781999366432	Elena Knows	GROUP B
23	0781913067721	A New Name: Septology	GROUP B
26	0781918277182	Cursed Bunny	GROUP B
33	0781929400182	An Inventory of Losses	GROUP B
34	0781913267172	Minor Detail	GROUP B
38	0781908870601	The Peer Field	GROUP B
42	0781910174654	The Eighth Life	GROUP B
44	0781929400182	Type	GROUP B
50	0781787300453	Mac and His Problems	GROUP B
54	0781912240166	Celestial Bodies	GROUP B
58	0783002245113	Love in the New Millenn...	GROUP B
62	0781788166619	The Pine Islands	GROUP B
68	0781788070658	Phantomas in Baghdad	GROUP B
70	0781910761880	The Dinner Guest	GROUP B
74	078025573067	The White Book	GROUP B
76	0781919386432	Pygals	GROUP B
78	0781942861181	The Clasp According to...	GROUP C
7	0781529427850	Lani Hantala Live in Lw...	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';



The screenshot shows the MySQL Workbench interface with the same query as before, but with a filter applied to the 'Result Grid' to show only rows where the group name is 'GROUP A'.

The 'Query Editor' contains the following SQL query:

```
1 use library;
2
3 -- Your 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
4 SELECT book_id,b.isbn, b.title,g.name as group_name FROM book_groups bg, books b, library.groups g
5 WHERE bg.book_id = b.id AND bg.group_id = g.id AND g.name = 'GROUP A';
```

The 'Result Grid' displays the following data:

book_id	isbn	title	group_name
1	078040278987	Skuller	GROUP A
8	0781914623203	The Starline	GROUP A
18	0781804277288	While We Were Dreaming	GROUP A
19	0781758825207	Wish Building	GROUP A
17	078117124449	Happy Science Academy	GROUP A
21	0781913008169	Phenomena	GROUP A
15	0781910000493	The Book of Jacob	GROUP A
19	0781910067602	Reverendence	GROUP A
33	0781913067030	In Memory of Memory	GROUP A
17	0781919000089	The Englishman	GROUP A
41	0780571348364	The Descent of Evening	GROUP A
40	0781910066113	The Other Name: Seid...	GROUP A
40	0781788077509	S. and P.	GROUP A
85	0781918677255	Ac Diah	GROUP A
17	0781844027895	Shore for the Summer	GROUP A
18	0780402504192	The Faculty of Dreams	GROUP A
85	0781010895784	The Years	GROUP A
89	0781010761664	The 7th Function of Lan...	GROUP A
73	0781025486854	The Seven Beyond	GROUP A
77	0781998792784	Oh, My Love	GROUP A

Local Instance 3306 - Warning - not supported

Administration Schemas book_group

SCHEMAS

Filter objects

- db
- db_class
- db_class_hw2
- library
 - book_groups
 - book_rating
 - books
 - groups
 - student_group
 - students
- Views
- Stored Procedures
- Functions
- sys

```

1 use library;
2
3 -- 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
4
5 SELECT book_id, b isbn, b title, g.name as group_name FROM book_groups bg, books b, library_groups g
6 WHERE bg.book_id = b.id AND bg.group_id = g.id AND g.name = 'GROUP B';
7
8
9

```

71.6

Result Grid Filter Rows: Export

book_id	isbn	title	group_name
3	0761608071038	White	GROUP B
4	0761608071039	To Another Road	GROUP B
10	0761608071037	Pyre	GROUP B
14	0761613067610	Parasite	GROUP B
18	0761608071044	Down Arrow	GROUP B
20	0761613067781	A New Name: Sepulchre Street	GROUP B
26	0761618277182	Ground Runny	GROUP B
30	0761608071035	An Invention of Solace	GROUP B
34	0761613067773	Minor Detail	GROUP B
38	0761608071031	The Paper Feast	GROUP B
42	076161071644	The Lullaby Lib	GROUP B
46	0761608071032	Tyrl	GROUP B
50	0761608071033	Mace and His Problem	GROUP B
54	0761613067184	Celestial Rules	GROUP B
58	0761608071034	Love in the New Millennium	GROUP B
62	0761608071036	The First House	GROUP B
66	0761608071030	Parasitism in Baghdad	GROUP B
70	0761613067180	The Outer Guest	GROUP B
74	0761608071037	The White Book	GROUP B
78	0761613067182	Figaro	GROUP B

Result 01

Read Only

Local Instance 3306 - Warning - not supported

Administration Schemas book_group

SCHEMAS

Filter objects

- db
- db_class
- db_class_hw2
- library
 - book_groups
 - book_rating
 - books
 - groups
 - student_group
 - students
- Views
- Stored Procedures
- Functions
- sys

```

1 use library;
2
3 -- 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
4
5 SELECT book_id, b isbn, b title, g.name as group_name FROM book_groups bg, books b, library_groups g
6 WHERE bg.book_id = b.id AND bg.group_id = g.id AND g.name = 'GROUP C';
7
8
9

```

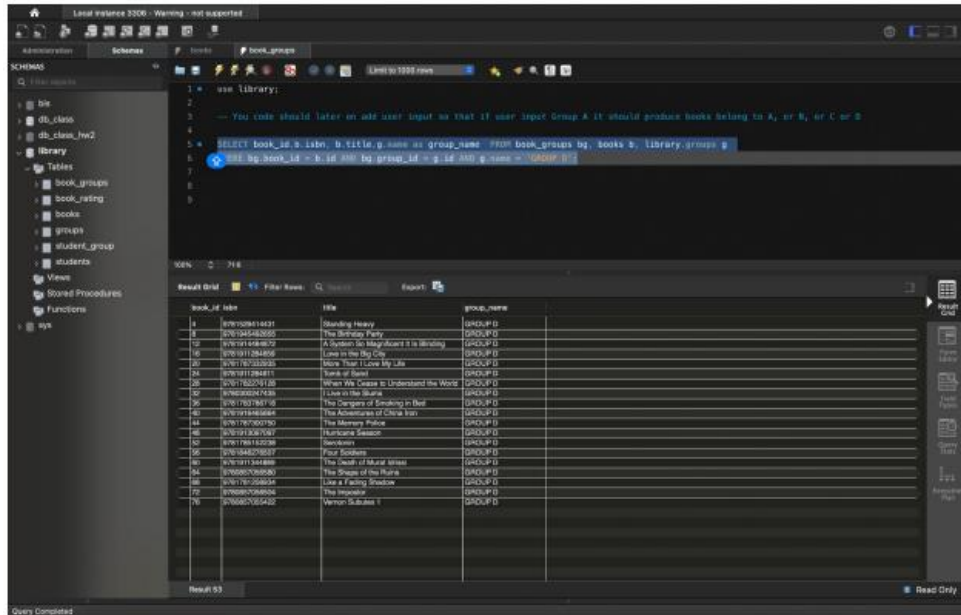
1.8

Result Grid Filter Rows: Export

book_id	isbn	title	group_name
5	0761608071031	The Gospel According to the New World	GROUP C
17	0761608071030	Jane Mendis Live in Life	GROUP C
19	0761613067680	Self-Gem	GROUP C
23	0761608071031	Heaven	GROUP C
18	0761607101780	The Book of Mafur	GROUP C
22	0761608071030	After the Sea	GROUP C
27	0761608071030	The War of the Pines	GROUP C
31	0761607101780	All Night All Blood is Black	GROUP C
35	0761613067773	Summer Brother	GROUP C
39	0761608071031	The Perfect Film	GROUP C
43	0761608071031	The Employment of The Greenage	GROUP C
47	0761608071031	Parasite on the Top of My Tongue	GROUP C
51	0761607101780	Hot Dog	GROUP C
55	0761613067180	Drive Your Head Over the Bones of the	GROUP C
59	0761608071030	Hearts of the Sea	GROUP C
63	0761613067180	The Penetration	GROUP C
67	0761608071030	Go West, Go	GROUP C
71	0761608071031	The Pines, the Sea	GROUP C
75	0761608071034	The World Goes On	GROUP C

Result 02

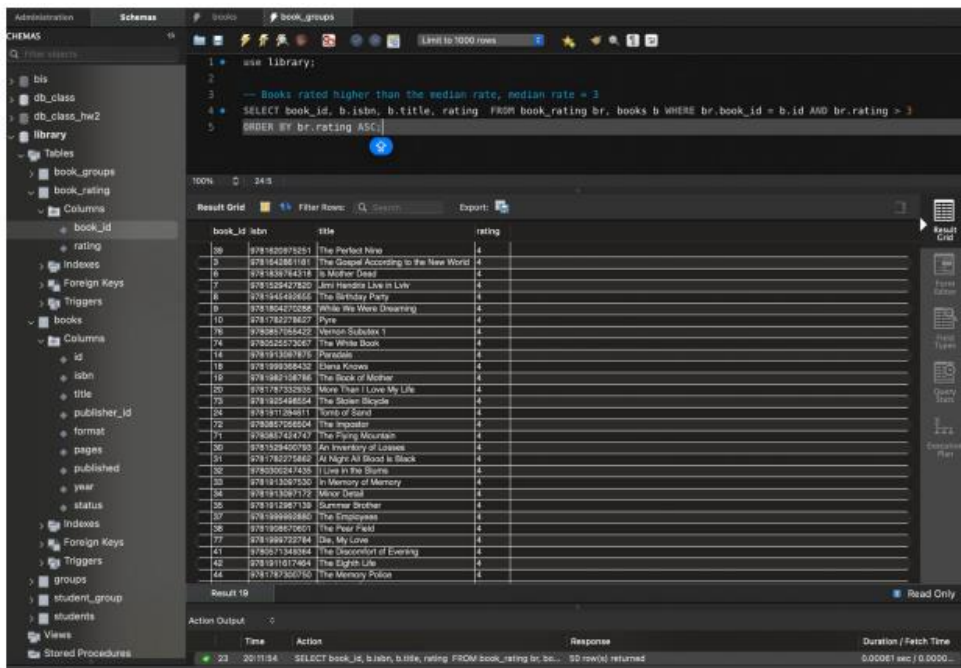
Read Only



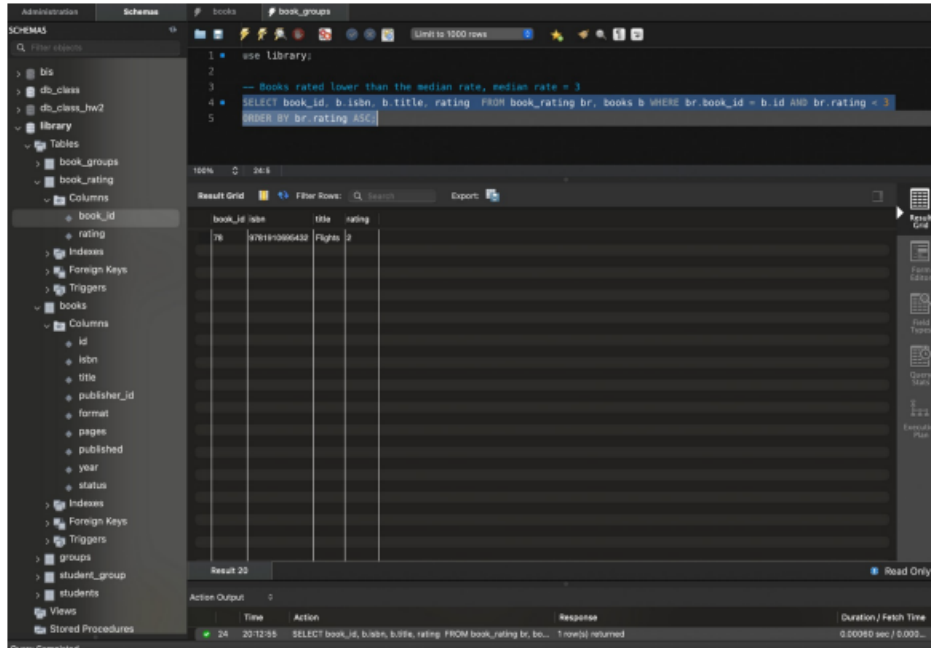
- Query for book ratings above or below average.

Books rated higher than the median rate, median rate = 3

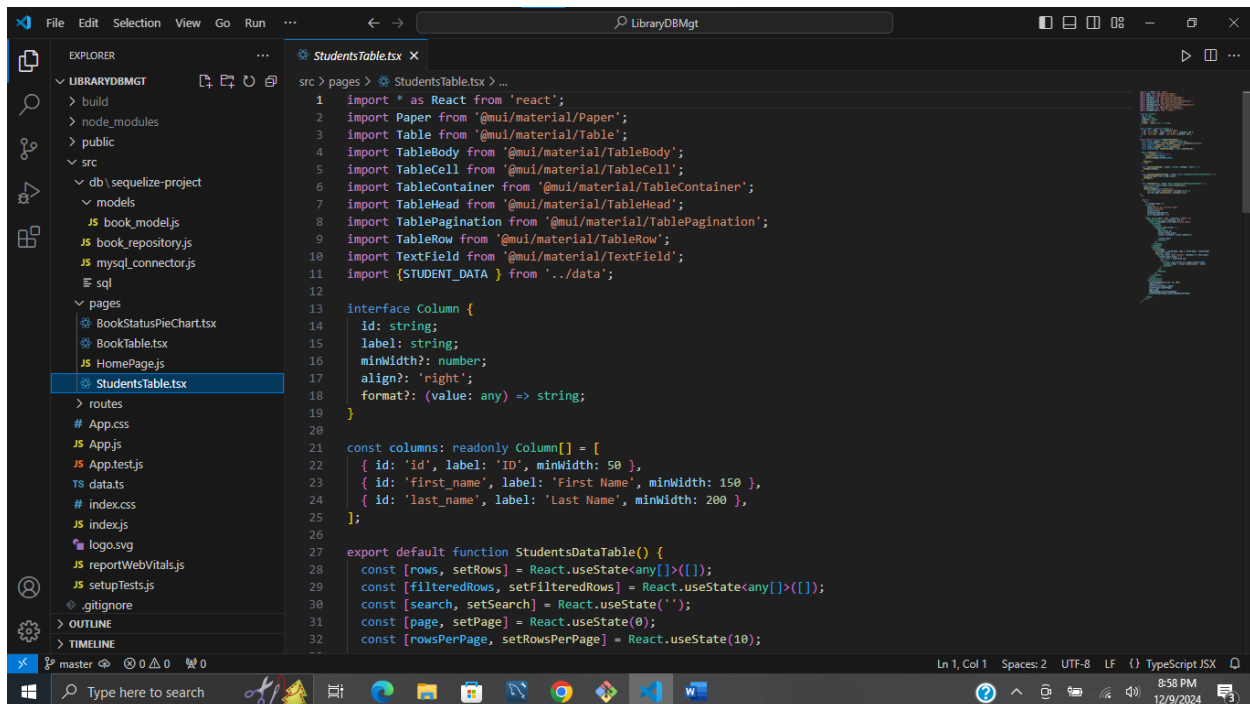
SELECT b.id,b.isbn, b.title, rating FROM book_rating br, books b WHERE br.book_id = b.id AND br.rating > 3;

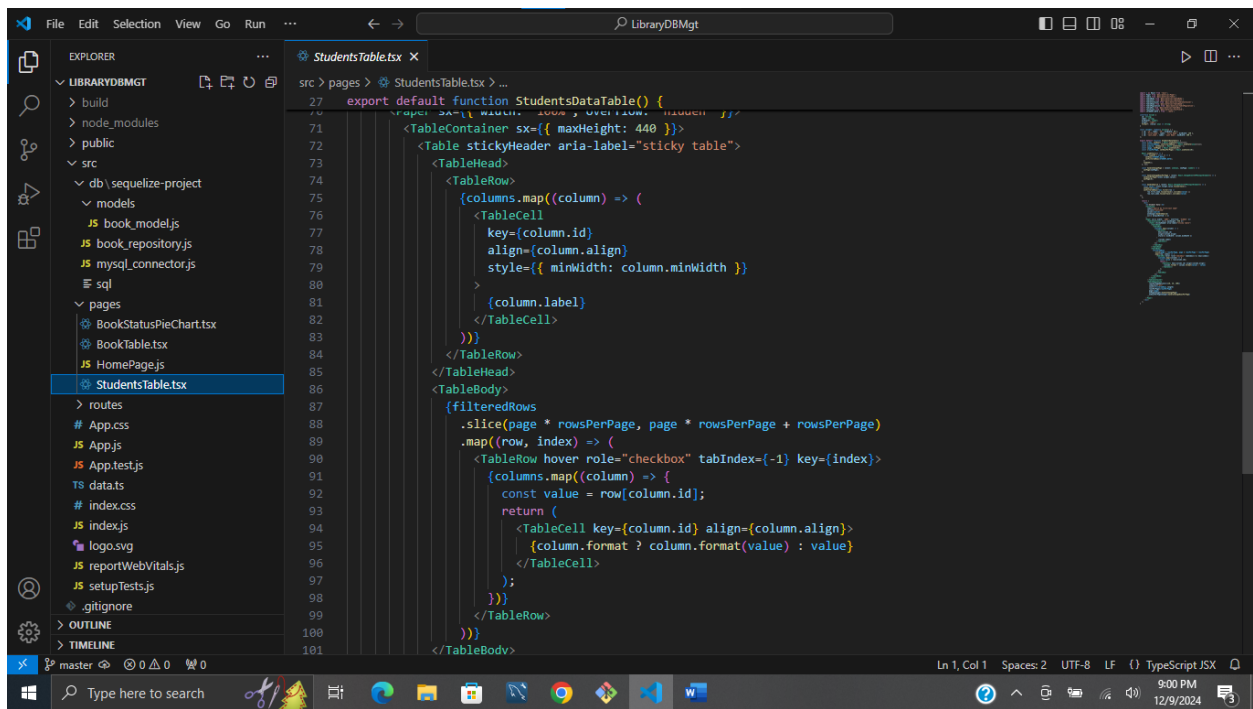


Books rated lower than the median rate, median rate = 3 SELECT b.id,b.isbn, b.title, rating FROM book_rating br, books b WHERE br.book_id = b.id AND br.rating < 3;



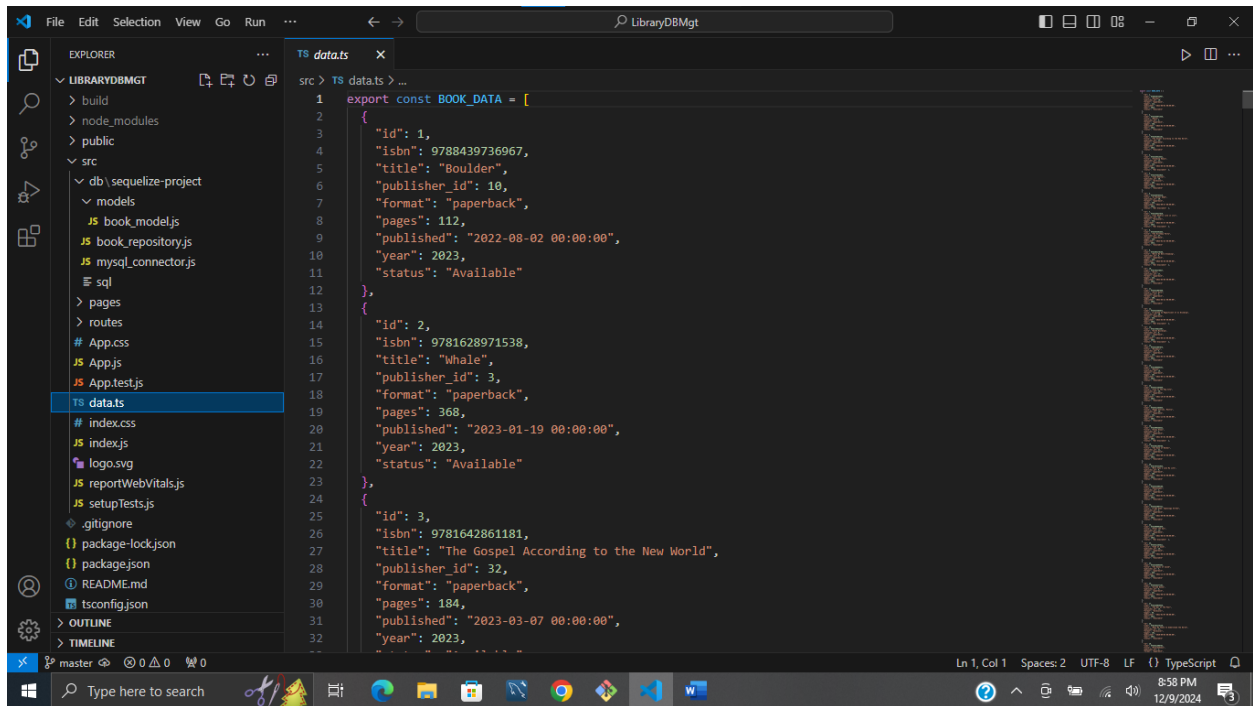
- Sequelize models: These are models to import the codes from MySQL database
 - Student data Model:





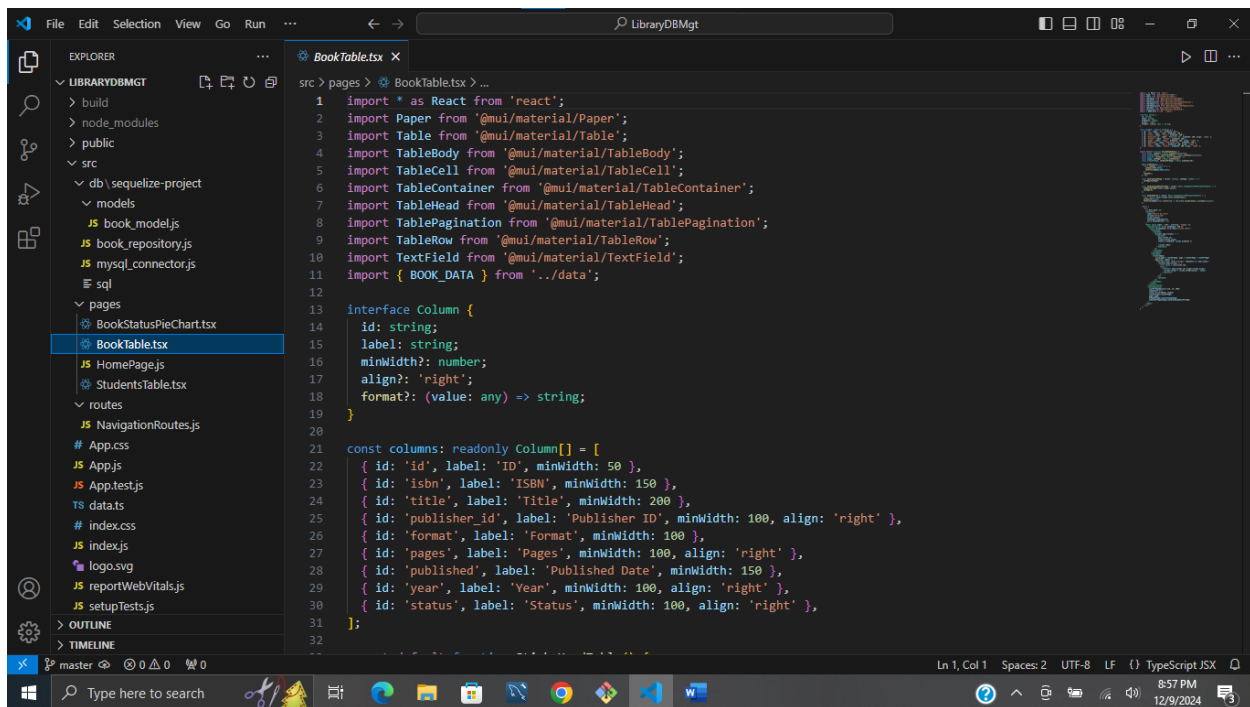
```
src > pages > StudentsTable.tsx > ...
27 export default function StudentsDataTable() {
28   <TableContainer sx={{ maxHeight: 440 }}>
29     <Table stickyHeader aria-label="sticky table">
30       <TableHead>
31         <TableRow>
32           {columns.map((column) => (
33             <TableCell
34               key={column.id}
35               align={column.align}
36               style={{ minWidth: column.minWidth }}
37             >
38               {column.label}
39             </TableCell>
40           ))}
41         </TableRow>
42       </TableHead>
43       <TableBody>
44         {filteredRows
45           .slice(page * rowsPerPage, page * rowsPerPage + rowsPerPage)
46           .map((row, index) => (
47             <TableRow hover role="checkbox" tabIndex={-1} key={index}>
48               {columns.map((column) => {
49                 const value = row[column.id];
50                 return (
51                   <TableCell key={column.id} align={column.align}>
52                     {column.format ? column.format(value) : value}
53                   </TableCell>
54                 );
55               })}
56             </TableRow>
57           ))}
58       </TableBody>
59     </Table>
60   </TableContainer>
61 )}
```

○ Book data Model:



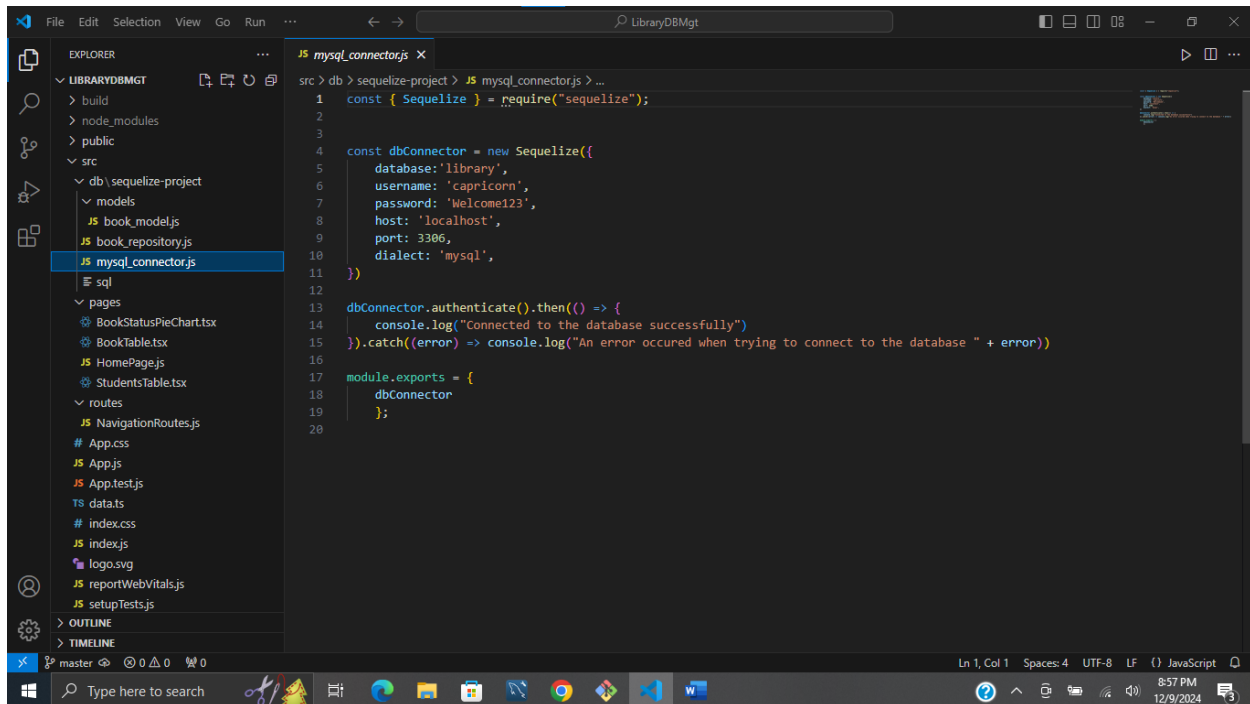
```
src > TS datats > ...
1 export const BOOK_DATA = [
2   {
3     "id": 1,
4     "isbn": 9788439736967,
5     "title": "Boulder",
6     "publisher_id": 10,
7     "format": "paperback",
8     "pages": 112,
9     "published": "2022-08-02 00:00:00",
10    "year": 2023,
11    "status": "Available"
12  },
13  {
14    "id": 2,
15    "isbn": 9781628971538,
16    "title": "Whale",
17    "publisher_id": 3,
18    "format": "paperback",
19    "pages": 368,
20    "published": "2023-01-19 00:00:00",
21    "year": 2023,
22    "status": "Available"
23  },
24  {
25    "id": 3,
26    "isbn": 9781642861181,
27    "title": "The Gospel According to the New World",
28    "publisher_id": 32,
29    "format": "paperback",
30    "pages": 184,
31    "published": "2023-03-07 00:00:00",
32    "year": 2023,
33  }
34 ]
```

○ Book Table Model:



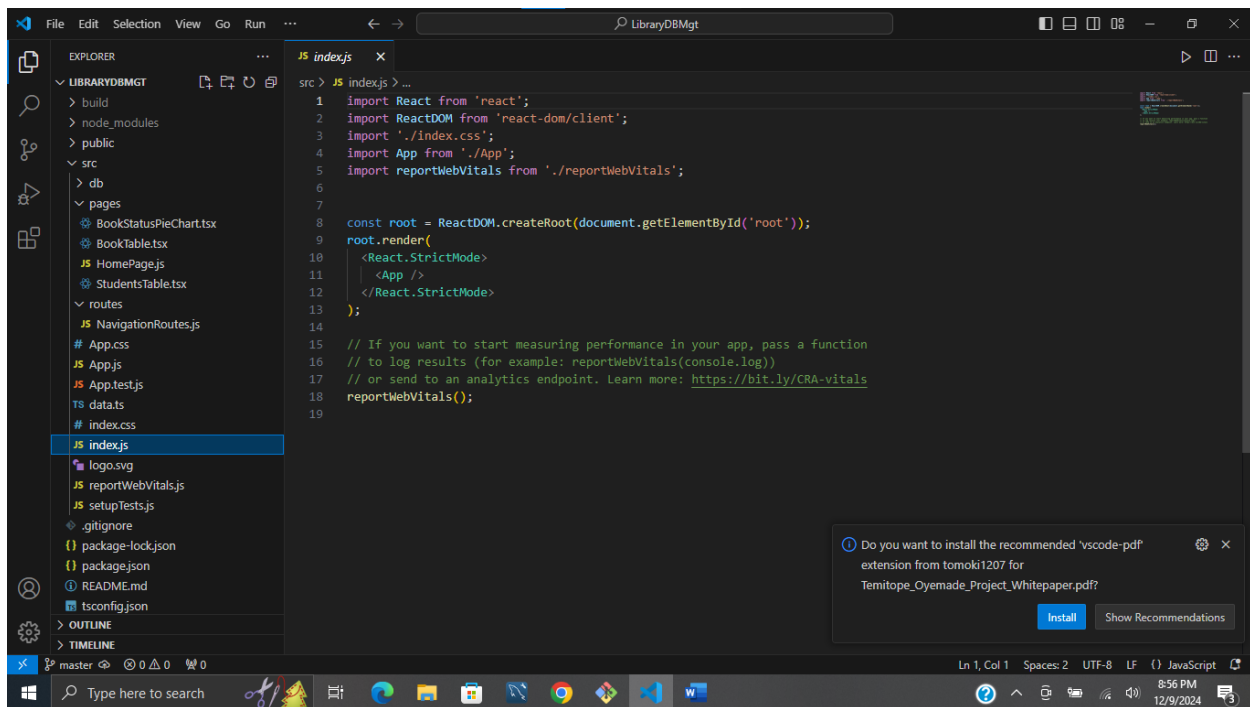
```
1 import * as React from 'react';
2 import Paper from '@mui/material/Paper';
3 import Table from '@mui/material/Table';
4 import TableBody from '@mui/material/TableBody';
5 import TableCell from '@mui/material/TableCell';
6 import TableContainer from '@mui/material/TableContainer';
7 import TableHead from '@mui/material/TableHead';
8 import TablePagination from '@mui/material/TablePagination';
9 import TableRow from '@mui/material/TableRow';
10 import TextField from '@mui/material/TextField';
11 import { BOOK_DATA } from '../data';
12
13 interface Column {
14   id: string;
15   label: string;
16   minWidth?: number;
17   align?: 'right';
18   format?: (value: any) => string;
19 }
20
21 const columns: readonly Column[] = [
22   { id: 'id', label: 'ID', minWidth: 50 },
23   { id: 'isbn', label: 'ISBN', minWidth: 150 },
24   { id: 'title', label: 'Title', minWidth: 200 },
25   { id: 'publisher_id', label: 'Publisher ID', minWidth: 100, align: 'right' },
26   { id: 'format', label: 'Format', minWidth: 100 },
27   { id: 'pages', label: 'Pages', minWidth: 100, align: 'right' },
28   { id: 'published', label: 'Published Date', minWidth: 150 },
29   { id: 'year', label: 'Year', minWidth: 100, align: 'right' },
30   { id: 'status', label: 'Status', minWidth: 100, align: 'right' },
31 ];
```

- Database Connector:

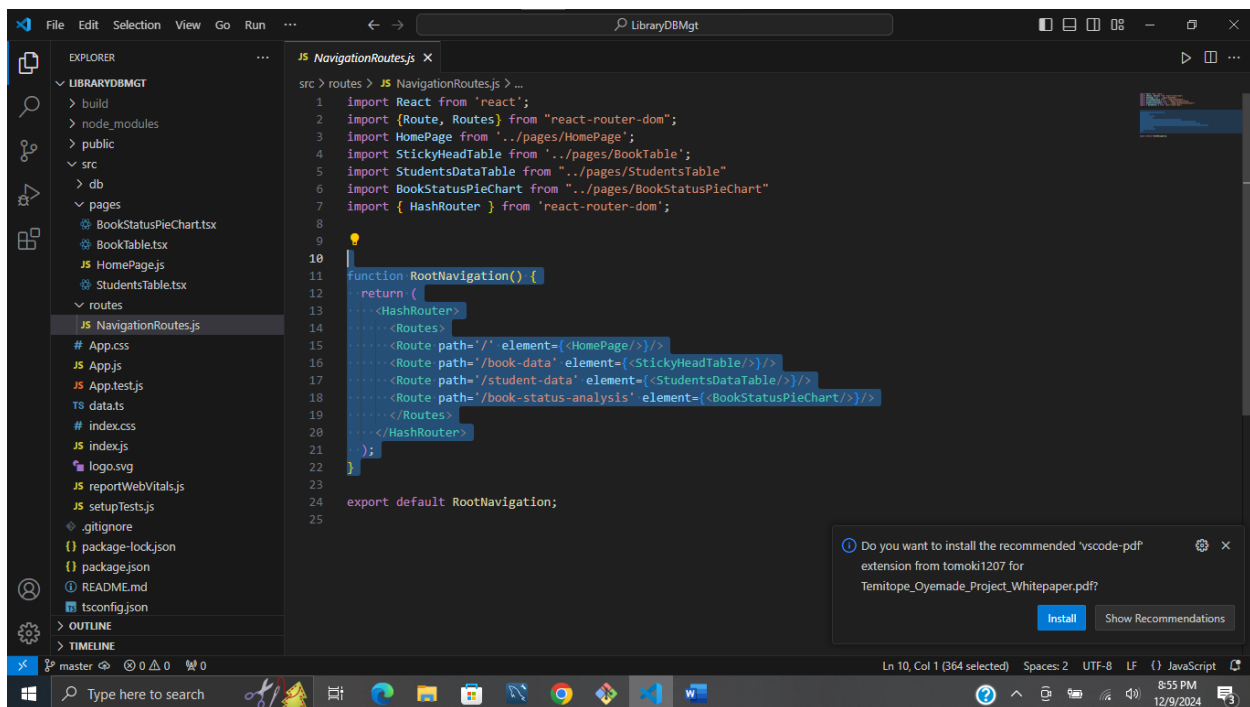


```
1 const { Sequelize } = require("sequelize");
2
3
4 const dbConnector = new Sequelize({
5   database: 'library',
6   username: 'capricorn',
7   password: 'welcome123',
8   host: 'localhost',
9   port: 3306,
10  dialect: 'mysql',
11 })
12
13 dbConnector.authenticate().then(() => {
14   console.log("Connected to the database successfully")
15 }).catch((error) => console.log("An error occurred when trying to connect to the database " + error))
16
17 module.exports = {
18   dbConnector
19 };
20
```

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

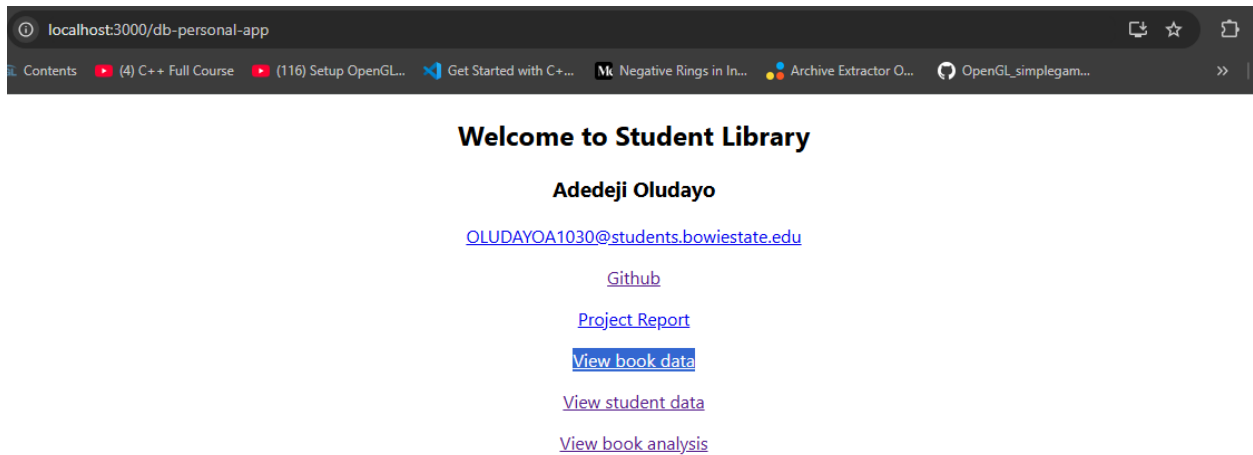


- HashRoute: HashRoute was imported for connectivity with GitHub



5.2. Front-end Demonstration

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



Webpage displaying Book Data

A screenshot of a web browser window showing a 'Book Data' table. The browser's address bar shows 'heziazza.github.io/LibraryDBMgt/#/book-data'. The page has a dark header with the title 'Book Data' and a search bar labeled 'Search by title'. Below the search bar is a table with 9 columns: ID, ISBN, Title, Publisher ID, Format, Pages, Published Date, Year, and Status. The table contains 7 rows of book data. At the bottom right of the table, it says 'Rows per page: 10' and '1-10 of 78'. The Windows taskbar is visible at the bottom of the screen.

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

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

Hezi/ x Aded/ x Chat/ x CS50 x Gemini x React x page: x React x React x Hezi/ x trave: x + -

heziazza.github.io/LibraryDBMgt/#/book-data

YI Contents (4) C++ Full Course (116) Setup OpenGL... Get Started with C+... Mk Negative Rings in In... Archive Extractor O... OpenGL_simplegam...

All Bookmarks

Book Data

Search by title

bou

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

Rows per page: 10 1-1 of 1 < >

Type here to search

9:35 PM 12/9/2024

Hezi/ x Aded/ x Chat/ x CS50 x Gemini x React x page: x React x React x Hezi/ x trave: x + -

heziazza.github.io/LibraryDBMgt/#/book-data

YI Contents (4) C++ Full Course (116) Setup OpenGL... Get Started with C+... Mk Negative Rings in In... Archive Extractor O... OpenGL_simplegam...

All Bookmarks

Book Data

Search by title

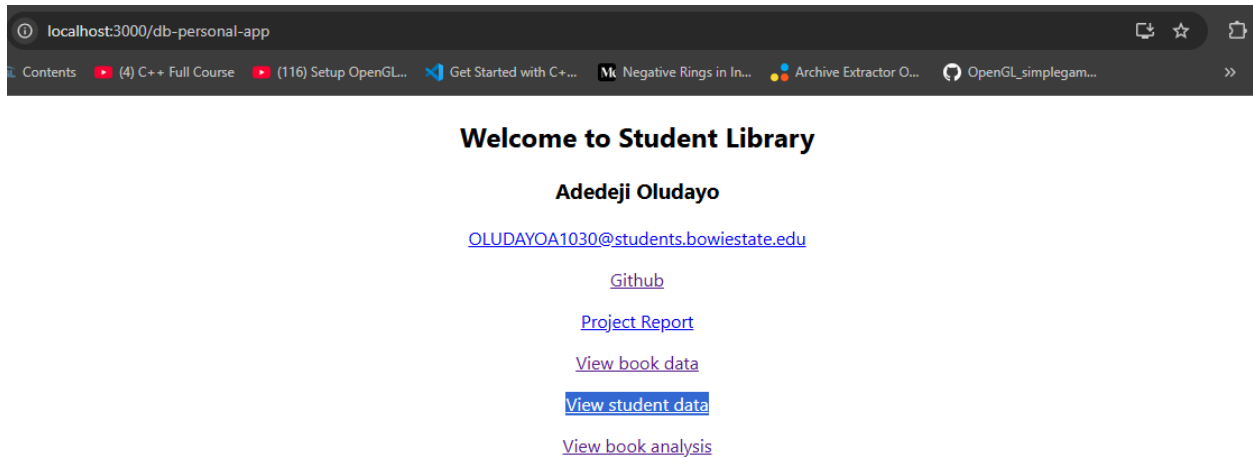
ID	ISBN	Title	Publisher ID	Format	Pages	Published Date	Year	Status
1	9788439736967	Boulder	10	paperback	112	2022-08-02 00:00:00	2023	Available
2	9781628971538	Whale	3	paperback	368	2023-01-19 00:00:00	2023	Available
3	9781642861181	The Gospel According to the New World	32	paperback	184	2023-03-07 00:00:00	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 < >

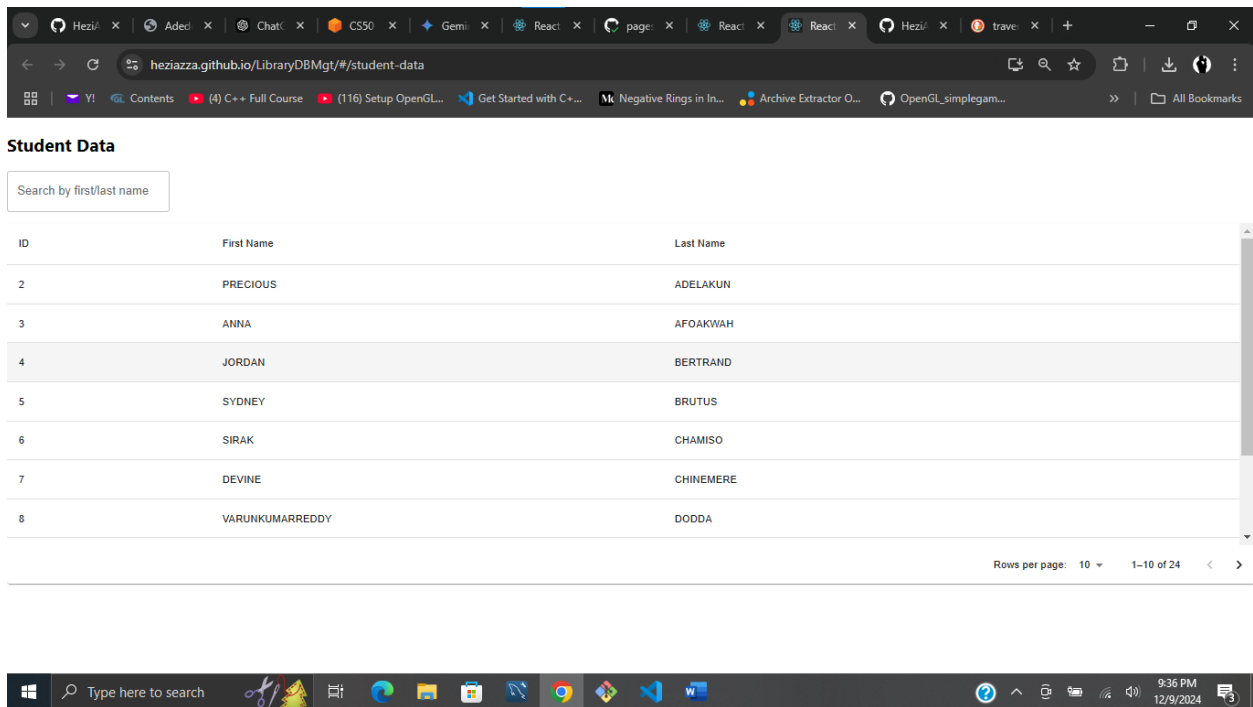
Type here to search

9:35 PM 12/9/2024

On the website, Click on “View student data” to view student data



Webpage displaying Student Data



Users can search student data by first or last name. For example as you begin to type “ade” for Adedeji all the name that have “ade” are displayed or “ja” for Jared, all the name with “ja” are displayed.

Hezi/ x Aded/ x Chat/ x CS50 x Gemini x React x page: x React x React x Hezi/ x trave: x + -

heziazza.github.io/LibraryDBMgt/#/student-data

Y! Contents (4) C++ Full Course (116) Setup OpenGL... Get Started with C+... Mk Negative Rings in In... Archive Extractor O... OpenGL_simplegam...

All Bookmarks

Student Data

Search by first/last name

ade

ID	First Name	Last Name
2	PRECIOUS	ADELAKUN
20	ADEDEJI	OLUDAYO
23	TEMITOPE	OYEMADE

Rows per page: 10 1-3 of 3 < >



Hezi/ x Aded/ x Chat/ x CS50 x Gemini x React x page: x React x React x Hezi/ x trave: x + -

heziazza.github.io/LibraryDBMgt/#/student-data

Y! Contents (4) C++ Full Course (116) Setup OpenGL... Get Started with C+... Mk Negative Rings in In... Archive Extractor O... OpenGL_simplegam...

All Bookmarks

Student Data

Search by first/last name

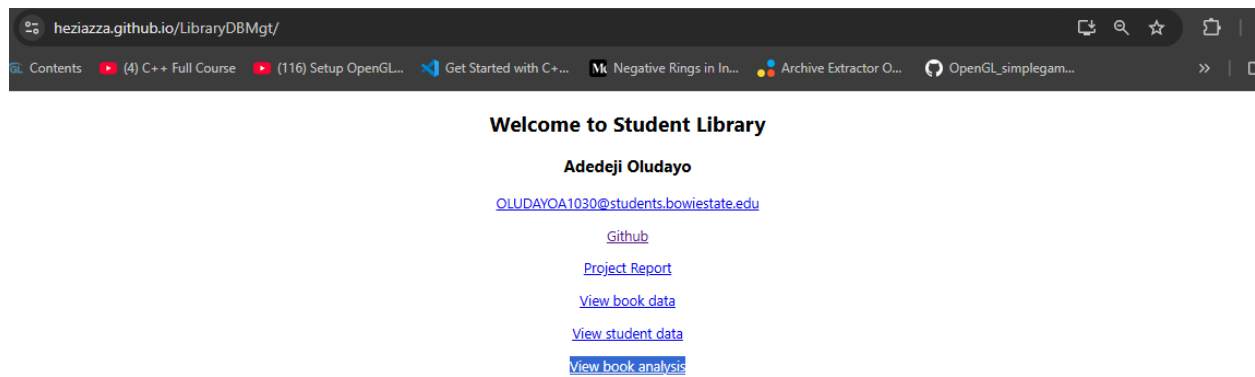
td

15	OLUWATOYIN	KODE
23	TEMITOPE	OYEMADE

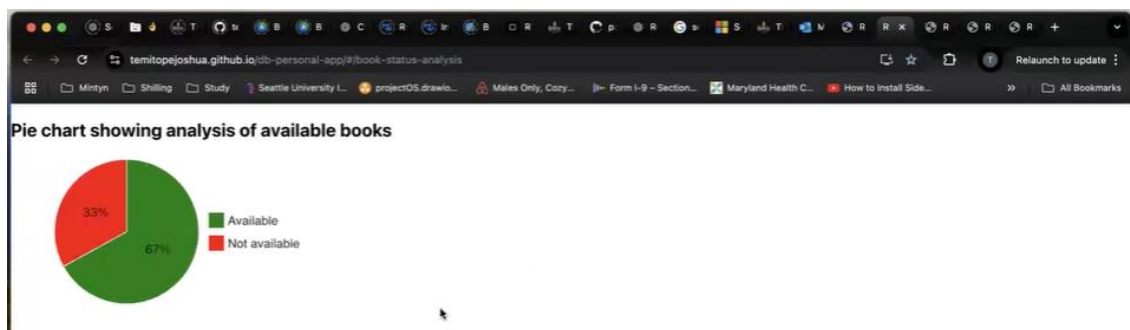
Rows per page: 10 1-2 of 2 < >



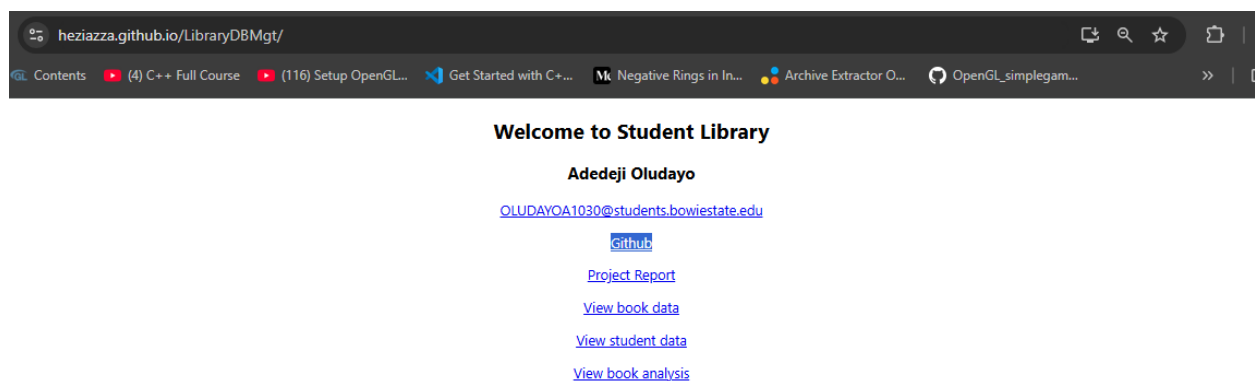
On the front end, Click on “View book analysis” to view the data analysis of book availability.



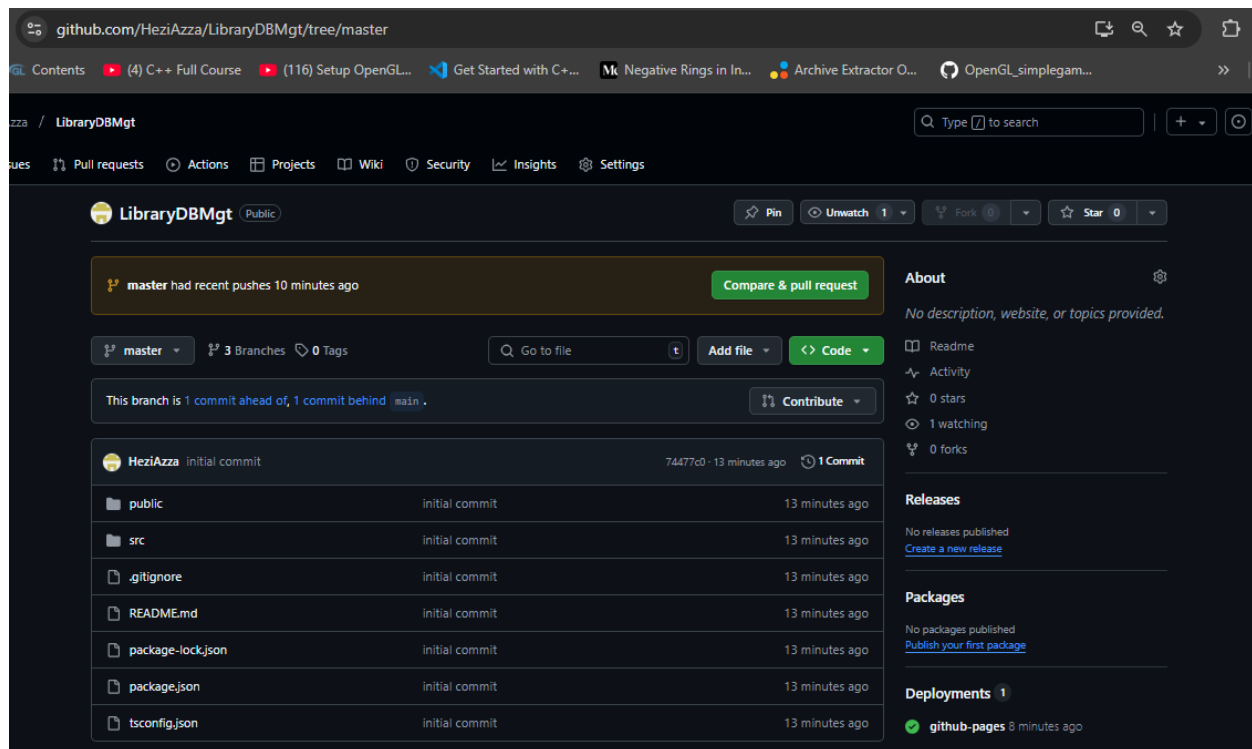
The pie chart shows a quick view of the percentage of books that are available and unavailable.



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



Website displaying GitHub Page with all the project packages and codes



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: [GitHub Repository Link]