

# Library Management System for Group-Based Student Borrowing

Adedeji Oludayo

<https://heziazza.github.io/LibraryDBMgt/>

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:

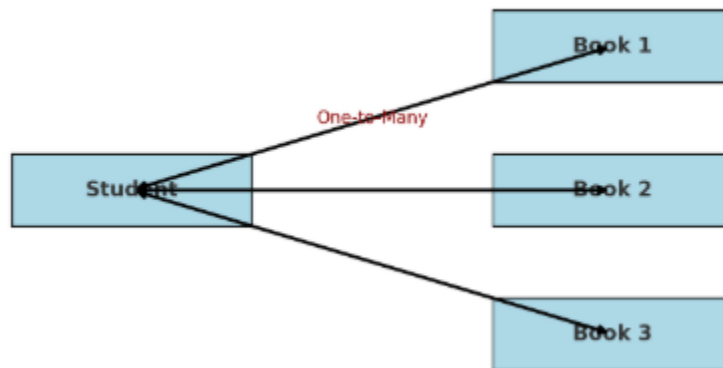
- A relational database to manage books, students, borrowing transactions, and group memberships.
- A user interface (UI) for querying borrowing details and viewing group information.
- A backend system to calculate average ratings and display books above or below the average.
- A method to dynamically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

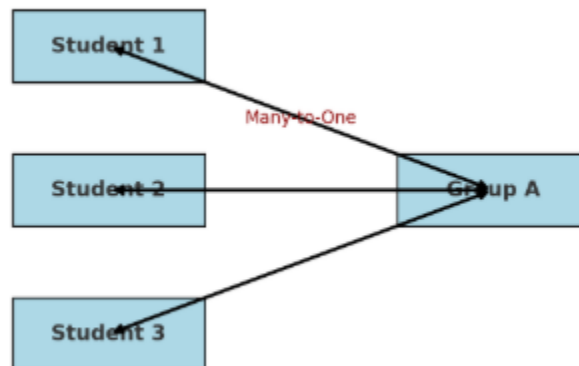
## 5. Design and Methodologies

### 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	97860970667	Doublet	18	paperback	112	2023-08-02 00:00:00	2023	1
2	9781028071338	Wings	3	paperback	368	2023-07-18 00:00:00	2023	1
3	9781945891181	The Cheaper According to the New World	26	paperback	184	2023-03-02 00:00:00	2023	1
5	9781714502025	Time Staller	26	hardcover	324	2022-04-27 00:00:00	2023	1
8	9781945406555	The Birthday Party	27	paperback	464	2023-01-04 00:00:00	2023	1
10	9781762776827	Pyra	17	paperback	294	2023-04-08 00:00:00	2023	1
11	9781913097963	Self Born	5	paperback	300	2023-05-22 00:00:00	2023	1
14	9781913091870	Headless	5	paperback	178	2023-03-03 00:00:00	2023	1
15	978150886851	Heaven	16	paperback	147	2023-05-12 00:00:00	2023	1
16	9781911264858	Love in the Big City	28	paperback	217	2021-10-28 00:00:00	2022	1
17	9781911264858	Swamp Books, Swamp	28	paperback	183	2021-12-02 00:00:00	2022	1
18	9781911264858	The Book of Esther	21	paperback	254	2021-09-19 00:00:00	2022	1
20	9781767932935	More Than I Love My Life	9	paperback	268	2021-08-26 00:00:00	2023	1
21	9781913505189	Phenotypes	1	paperback	292	2022-01-01 00:00:00	2022	1
22	9781913507721	A New Name: Sepulchre Writ	5	paperback	208	2021-09-08 00:00:00	2022	1
24	9781911264858	Temp of Love	28	paperback	128	2021-08-12 00:00:00	2022	1
25	9781911662553	The Books of Jacob	35	paperback	240	2021-11-15 00:00:00	2023	1
26	9781918277182	Curved Bones	5	paperback	293	2021-06-15 00:00:00	2022	1
27	9781918268628	The War of the Post	18	hardcover	360	2021-07-02 00:00:00	2023	1
28	9781762776738	When We Cease to Understand the W...	13	hardcover	188	2020-04-03 00:00:00	2023	1
30	9781525400793	An Inventory of Losses	12	hardcover	256	2020-08-20 00:00:00	2023	1
31	9781762770862	All Night All Blood & Black	17	hardcover	182	2020-11-03 00:00:00	2023	1
32	9781623247435	I Live in the Future	28	hardcover	344	2020-09-19 00:00:00	2023	1
34	9781913507172	Salmon Detail	5	paperback	144	2020-05-04 00:00:00	2023	1
35	9781911957139	Summer Brother	32	paperback	285	2021-04-08 00:00:00	2023	1
36	9781908610601	The Pear Field	14	paperback	163	2020-10-08 00:00:00	2023	1
41	9781911340584	The Discomfort of Evening	25	hardcover	240	2020-10-08 00:00:00	2023	1
43	9781909400651	The Solifamentum of The Greenhouse	3	paperback	258	2020-07-17 00:00:00	2023	1

SELECT \* FROM books WHERE status = 0;

id	isbn	title	publisher_id	format	pages	published	year	status
4	9781528144301	Standing Heavy	12	paperback	282	2023-05-20 00:00:00	2023	0
6	9781626764318	Is Mother Dead	28	paperback	380	2023-09-27 00:00:00	2023	0
7	9781626764318	Am Harvill, Live It Life	12	hardcover	418	2023-04-27 00:00:00	2023	0
9	9781804271088	White Sea Vows Dreaming	5	paperback	368	2023-02-20 00:00:00	2023	0
12	9781814854572	A System to Magnificent & to Binding	29	paperback	344	2023-03-09 00:00:00	2023	0
13	9781758255507	North Building	8	paperback	272	2023-05-16 00:00:00	2023	0
18	9781688785432	Umbra Arcana	12	paperback	143	2021-07-01 00:00:00	2022	0
23	9781809955539	After the Sea	11	paperback	118	2021-08-31 00:00:00	2022	0
29	9781811506782	Watchwords	1	paperback	178	2020-04-02 00:00:00	2021	0
33	9781413505530	In Memory of Memory	5	paperback	510	2021-03-27 00:00:00	2021	0
38	9781762776718	The Gospels of Breathing in Red	6	paperback	187	2021-04-01 00:00:00	2021	0
37	9781809955586	The Empress	11	paperback	138	2020-10-31 00:00:00	2021	0
40	9781818408064	The Adventures of China Iron	2	paperback	290	2019-11-14 00:00:00	2020	0
42	9781811517484	The Light Life	29	hardcover	344	2019-11-14 00:00:00	2020	0
45	9781811566613	The City Names: Sepulchre L&B	5	paperback	363	2019-10-10 00:00:00	2020	0
53	9781765152258	Serotonin	35	hardcover	390	2019-09-26 00:00:00	2020	0
63	9781811506528	The Hermit	1	paperback	180	2018-10-04 00:00:00	2018	0
65	9781718652194	The Years	15	paperback	224	2018-09-20 00:00:00	2018	0
70	9781818408064	The Stone Skye	24	paperback	418	2017-08-29 00:00:00	2018	0
74	9781802575907	The White Book	18	paperback	181	2018-04-26 00:00:00	2018	0
76	9781768168134	The World Lives On	28	paperback	320	2018-05-31 00:00:00	2018	0
77	9781809103194	Oh, My Love	3	paperback	132	2017-09-14 00:00:00	2018	0

- Query for group affiliations and preferred books.

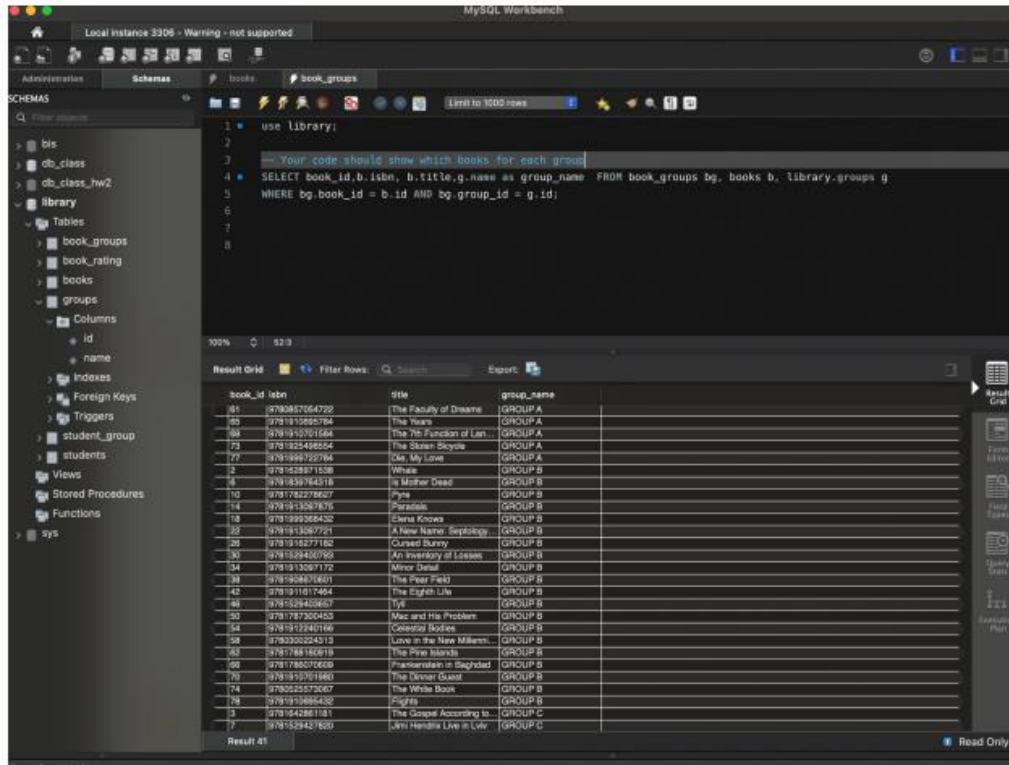
```
CREATE TABLE `groups` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci  
SELECT count(sg.student_id) as no_of_students_in_group,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

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'book\_groups', 'book\_rating', 'books', 'groups', 'student\_group', and 'students'. The 'groups' table is selected. The main editor window shows a SQL query:   
`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;`  
The 'Result Grid' at the bottom displays the query results in a table with 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

  
The 'Action Output' pane at the bottom right shows the execution details for the query, including the duration and fetch time for each row.

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;



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';

Local Instance SQL - Warning - not supported

Administration Schemas book book\_groups

SCHEMAS

- filter objects
- db
- db\_class
- db\_class\_hw2
- library
  - Tables
    - 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 A';
7
8
9

```

100% 1/7

Result Grid Filter Rows Export

book_id	isbn	title	group_name
1	00000000000000000000	Booker	GROUP A
2	00000000000000000000	Time Machine	GROUP A
3	00000000000000000000	White Man's Burden	GROUP A
4	00000000000000000000	Search for God	GROUP A
5	00000000000000000000	Angels' Strife, Liberty	GROUP A
6	00000000000000000000	Phenomena	GROUP A
7	00000000000000000000	The Book of Job	GROUP A
8	00000000000000000000	Whitehead's	GROUP A
9	00000000000000000000	In Memory of Memory	GROUP A
10	00000000000000000000	The Endgame	GROUP A
11	00000000000000000000	The Descent of Evening	GROUP A
12	00000000000000000000	The Other Name, Secret	GROUP A
13	00000000000000000000	Julia Lynn	GROUP A
14	00000000000000000000	At Day	GROUP A
15	00000000000000000000	James for the Remnant	GROUP A
16	00000000000000000000	The Faculty of Dreams	GROUP A
17	00000000000000000000	The Name	GROUP A
18	00000000000000000000	The New	GROUP A
19	00000000000000000000	The Book of Job	GROUP A
20	00000000000000000000	One, My Love	GROUP A

Result 00 Read Only

Local Instance SQL - Warning - not supported

Administration Schemas book book\_groups

SCHEMAS

- filter objects
- db
- db\_class
- db\_class\_hw2
- library
  - Tables
    - 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

```

100% 71/4

Result Grid Filter Rows Export

book_id	isbn	title	group_name
3	00000000000000000000	White	GROUP B
4	00000000000000000000	In Mother David	GROUP B
5	00000000000000000000	Pige	GROUP B
6	00000000000000000000	Phenomena	GROUP B
7	00000000000000000000	China Kines	GROUP B
8	00000000000000000000	A New Name, Secretly St-43	GROUP B
9	00000000000000000000	Curved Runny	GROUP B
10	00000000000000000000	An Inventory of Losses	GROUP B
11	00000000000000000000	Never Lined	GROUP B
12	00000000000000000000	The Road Past	GROUP B
13	00000000000000000000	The Right Life	GROUP B
14	00000000000000000000	Ty	GROUP B
15	00000000000000000000	Mac and His Problem	GROUP B
16	00000000000000000000	Conductor Bumper	GROUP B
17	00000000000000000000	Love is the New Millennium	GROUP B
18	00000000000000000000	The First Island	GROUP B
19	00000000000000000000	Perseus in the Night	GROUP B
20	00000000000000000000	The Outer World	GROUP B
21	00000000000000000000	The White Book	GROUP B
22	00000000000000000000	Agnes	GROUP B

Result 01 Read Only

Local Instance 3306 - Warning - not supported

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
1	978-0438811811	The Grapes According to the New World	GROUP C
7	978-0102942782	Jane Heredia Live & Loud	GROUP C
11	978-0113077867	Old Fern	GROUP C
13	978-0109848251	Heaven	GROUP C
18	978-0821581981	The Book of Esther	GROUP C
25	978-0699585552	After the Sun	GROUP C
27	978-0696366356	The Way of Zen Poet	GROUP C
31	978-0783215682	At Night All Blood is Black	GROUP C
36	978-012687139	Summer Brother	GROUP C
38	978-0620295811	The Perfect Nine	GROUP C
43	978-0694546611	The Enlightenment of The Greenyago	GROUP C
44	978-0694546611	Paras on the Top of My Tongue	GROUP C
47	978-0783219223	Heart Log	GROUP C
58	978-013983715	Drive Your Paw Over the Bones of the	GROUP C
59	978-0139201650	Necktie & 21 Sticks	GROUP C
20	978-011908329	The Harlequin	GROUP C
87	978-0482152320	Go, West, Go!	GROUP C
71	978-0814201621	The Flying Machine	GROUP C
74	978-0788180124	The World Goes On	GROUP C

Local Instance 3306 - Warning - not supported

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
1	978-0438811811	Standing Heavy	GROUP B
8	978-0104526255	The Slightly Pinky	GROUP B
12	978-0114584872	A System On May/June 11 is Missing	GROUP B
16	978-0191284865	Love in the Big City	GROUP B
28	978-0178233335	More Than I Love My Life	GROUP B
29	978-0191284811	Bank of Soul	GROUP B
35	978-0178233118	When We Learn to Understand the World	GROUP B
37	978-0302947435	I Live in the Shark	GROUP B
39	978-0178278149	The Dangers of Smoking in Bed	GROUP B
41	978-0191456841	The Adventures of Chick & Leo	GROUP B
44	978-0178760780	The Memory Palace	GROUP B
46	978-0191287197	The Purple Mirror	GROUP B
50	978-0178816238	Sanctuary	GROUP B
56	978-0348578237	Four Brothers	GROUP B
60	978-0191134488	The Book of Moral Stories	GROUP B
64	978-0801768580	The Shape of the Moon	GROUP B
66	978-0178238817	Like a Flying Shadow	GROUP B
72	978-0801768580	The Invisible	GROUP B
76	978-0801768580	Vampire Sublime 1	GROUP B

- 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;



1 use Library;

2

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

4 SELECT book\_id, b.isbn, b.title, rating FROM book\_rating br, books b WHERE br.book\_id = b.id AND br.rating > 3

5 ORDER BY br.rating ASC;

book_id	isbn	title	rating
29	9781620975251	The Perfect Nine	4
3	9781642861161	The Gospel According to the New World	4
6	9781935174318	To Mother, I Said	4
27	9781529427820	Joni Hendrix Live in Life	4
8	9781544493655	The Birthday Party	4
9	9781854270258	When We Were Dreaming	4
10	9781782278627	Pyle	4
76	9780887055422	Vernon Scholtes 1	4
74	9782525573667	The White Book	4
14	9781911897875	Paradise	4
18	9781999368432	Elena Knows	4
19	9781982104786	The Book of Mother	4
20	9781781332835	More Than I Love My Life	4
23	9781625448654	The Broken Bonds	4
24	9781911264811	Tomb of Sand	4
72	9780887056504	The Imposter	4
21	9780887432427	The Winding Mountain	4
26	9781529450793	An Inventory of Losses	4
21	9781782275862	At Night All Blood Is Black	4
30	9780300247435	I Live in the Storm	4
23	9781911897830	In Memory of Memory	4
34	9781911897772	Minor Detail	4
25	9781812867138	Sunstar Brother	4
26	9781999498865	The Synagogue	4
28	9781629427801	The Hair Field	4
77	9781999722764	Da, My Love	4
41	178057134884	The Discomfort of Evening	4
42	9781918174854	The Dying Life	4
44	9781787300702	The Memory Palace	4

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;

1 use Library;

2

3 -- Books rated lower than the median rate, median rate = 3

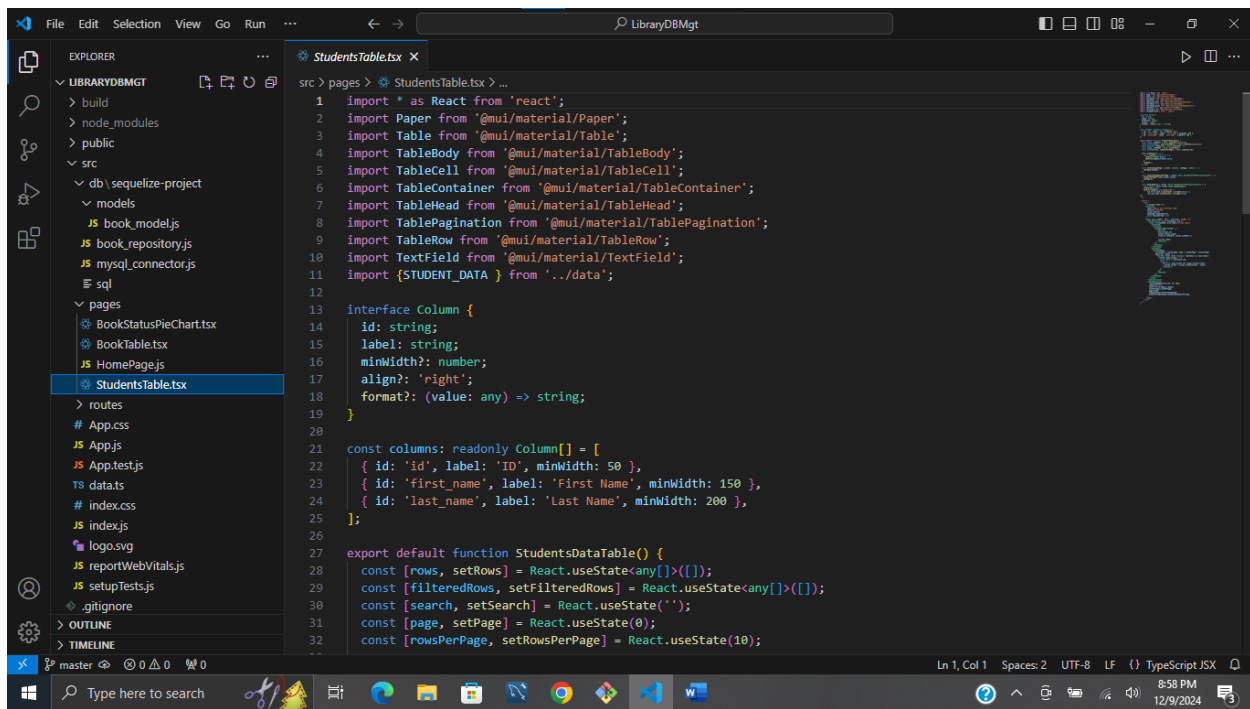
4 SELECT book\_id, b.isbn, b.title, rating FROM book\_rating br, books b WHERE br.book\_id = b.id AND br.rating < 3

5 ORDER BY br.rating ASC;

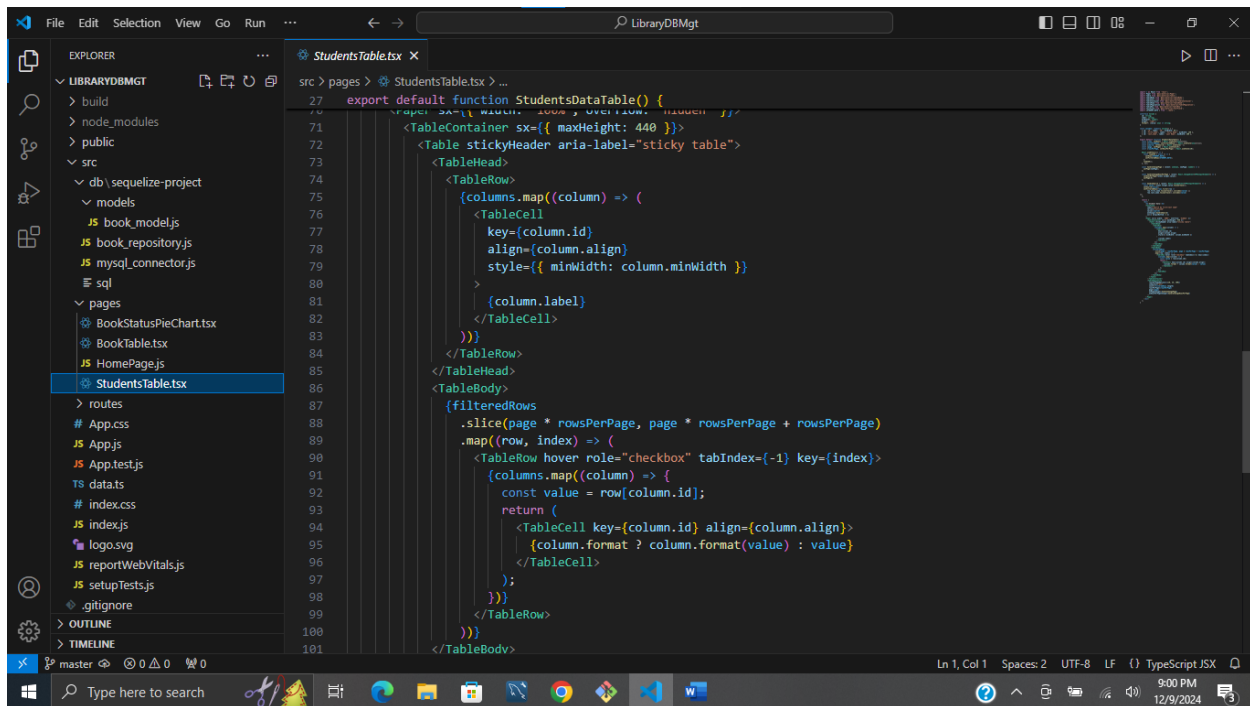
book_id	isbn	title	rating
76	978180965432	Right	2

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



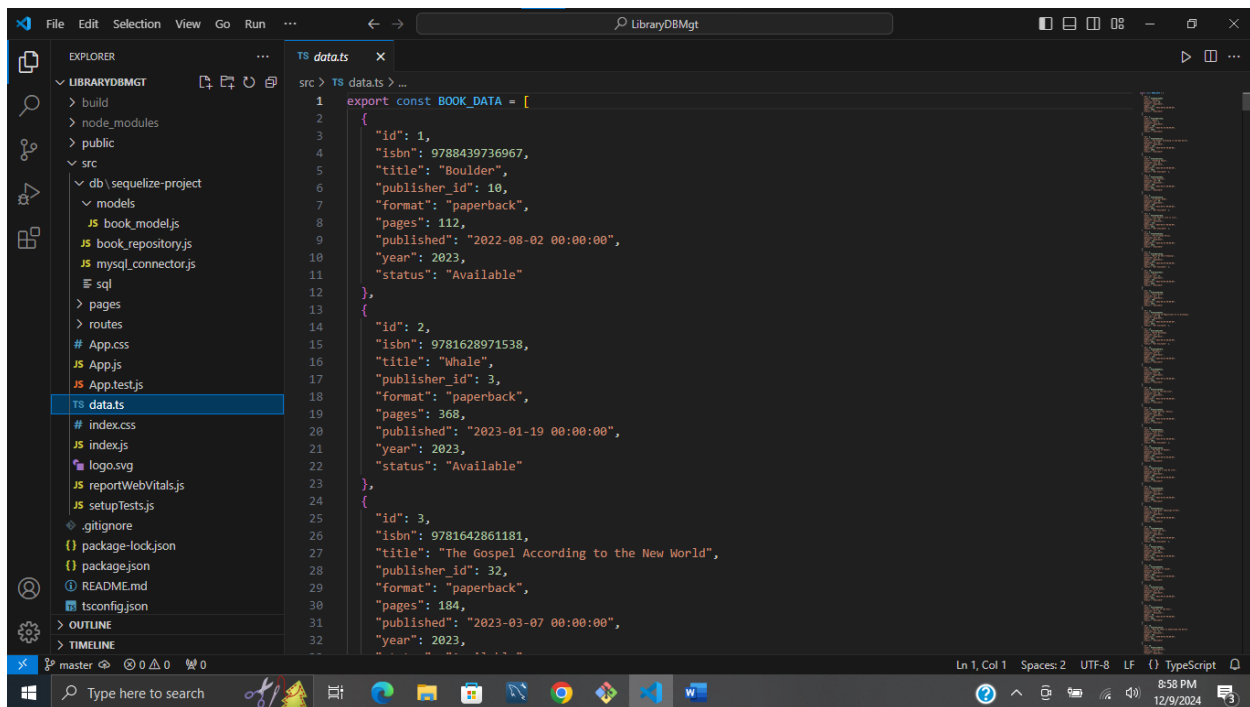


```
src > pages > StudentsTable.tsx > ...
1 import * as React from 'react';
2 import Paper from '@mui/material/Paper';
3 import Table from '@mui/material/Table';
4 import TableBody from '@mui/material/TableBody';
5 import TableCell from '@mui/material/TableCell';
6 import TableContainer from '@mui/material/TableContainer';
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 {STUDENT_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: 'first_name', label: 'First Name', minWidth: 150 },
24   { id: 'last_name', label: 'Last Name', minWidth: 200 },
25 ];
26
27 export default function StudentsDataTable() {
28   const [rows, setRows] = React.useState<any[]>([]);
29   const [filteredRows, setFilteredRows] = React.useState<any[]>([]);
30   const [search, setSearch] = React.useState('');
31   const [page, setPage] = React.useState(0);
32   const [rowsPerPage, setRowsPerPage] = React.useState(10);
```

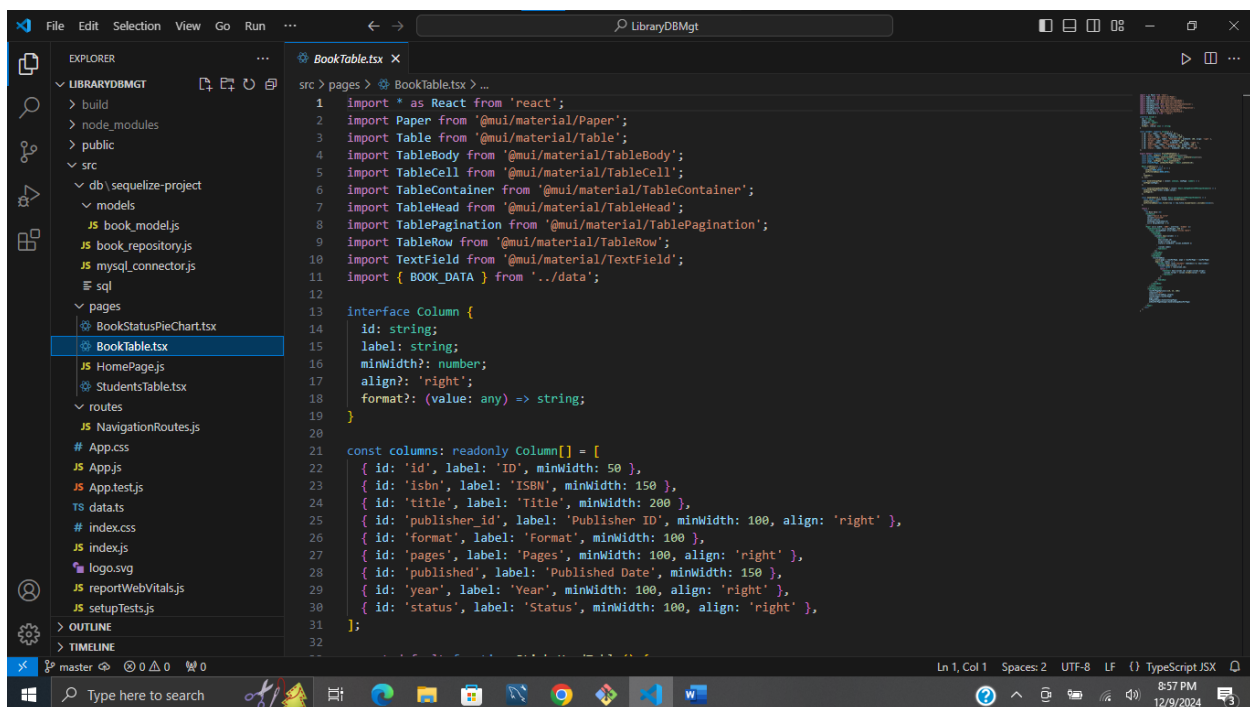


```
27 export default function StudentsDataTable() {
28   // ... (previous code) ...
71 <TableContainer sx={{ maxHeight: 440 }}>
72   <Table stickyHeader aria-label="sticky table">
73     <TableHead>
74       <TableRow>
75         {columns.map((column) => (
76           <TableCell
77             key={column.id}
78             align={column.align}
79             style={{ minWidth: column.minWidth }}
80           >
81             {column.label}
82           </TableCell>
83         ))}
84       </TableRow>
85     </TableHead>
86     <TableBody>
87       {filteredRows
88         .slice(page * rowsPerPage, page * rowsPerPage + rowsPerPage)
89         .map((row, index) => (
90           <TableRow hover role="checkbox" tabIndex={-1} key={index}>
91             {columns.map((column) => {
92               const value = row[column.id];
93               return (
94                 <TableCell key={column.id} align={column.align}>
95                   {column.format ? column.format(value) : value}
96                 </TableCell>
97               );
98             })}
99           </TableRow>
100         ))}
101     </TableBody>
```

- Book data Model:



### ○ Book Table Model:



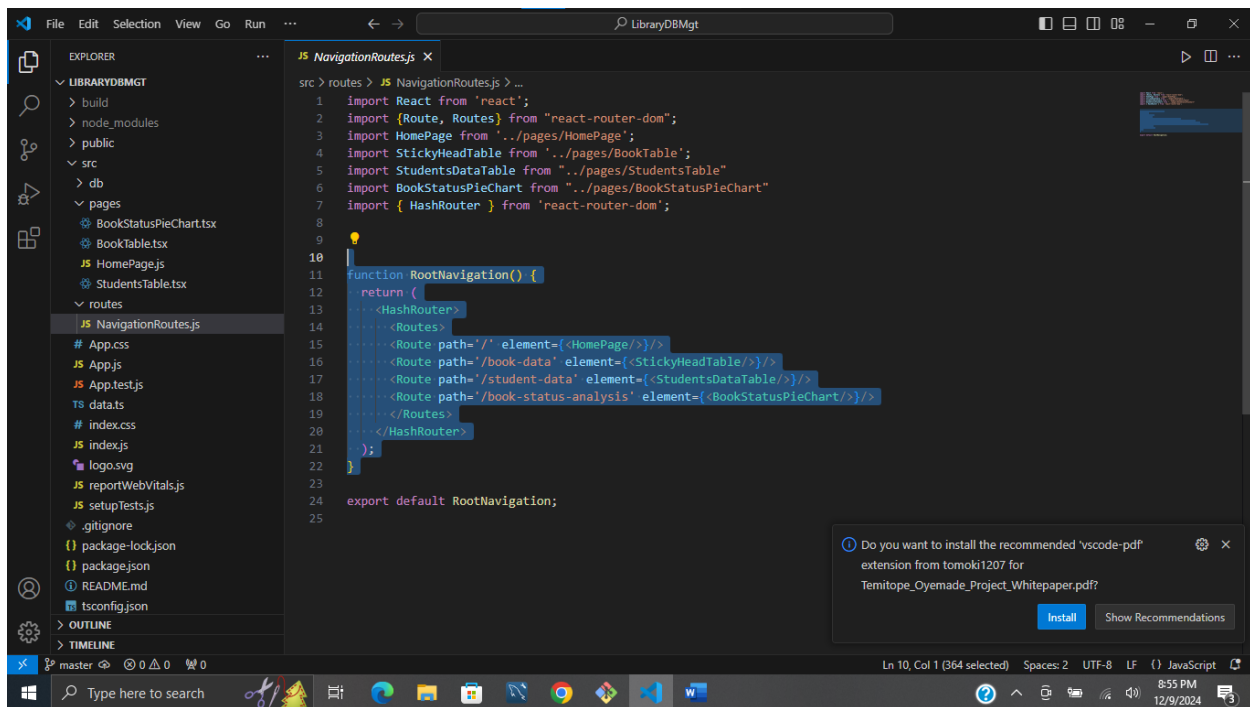
### ○ Database Connector:

```
src > db > sequelize-project > JS mysql_connectorjs > ...
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

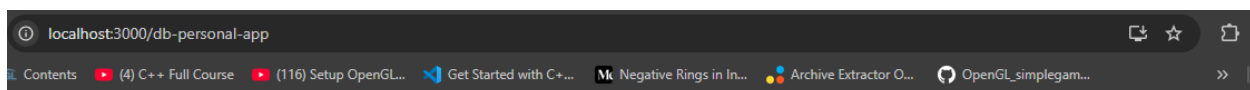
```
src > JS indexjs > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import reportWebVitals from './reportWebVitals';
6
7
8  const root = ReactDOM.createRoot(document.getElementById('root'));
9  root.render(
10   <React.StrictMode>
11     <App />
12   </React.StrictMode>
13 );
14
15 // If you want to start measuring performance in your app, pass a function
16 // to log results (for example: reportWebVitals(console.log))
17 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
18 reportWebVitals();
19
```

- 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



### Welcome to Student Library

Adedeji Oludayo

[OLUDAYOA1030@students.bowiestate.edu](mailto:OLUDAYOA1030@students.bowiestate.edu)

[Github](#)

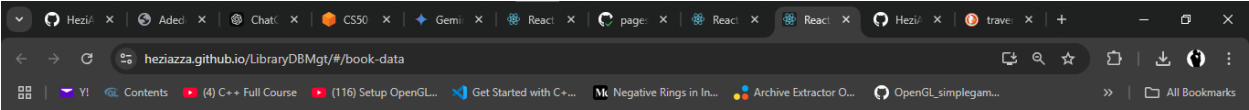
[Project Report](#)

[View book data](#)

[View student data](#)

[View book analysis](#)

Webpage displaying Book Data



Book Data

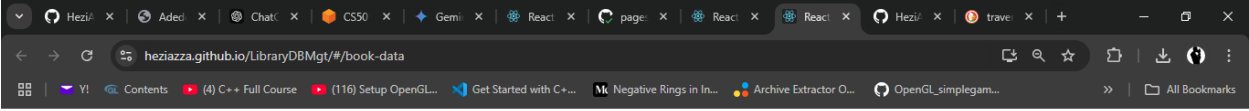
Search by title

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	9781639764318	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 Whale, or Paradais – as you begin to type, it begins to filter out your search



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

Rows per page: 10 1–1 of 1



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

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	9781639764318	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

Windows taskbar: Type here to search, 9:35 PM 12/9/2024

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

localhost:3000/db-personal-app

## Welcome to Student Library

Adedeji Oludayo

[OLUDAYOA1030@students.bowiestate.edu](mailto:OLUDAYOA1030@students.bowiestate.edu)

[Github](#)

[Project Report](#)

[View book data](#)

[View student data](#)

[View book analysis](#)

Webpage displaying Student Data

Hezi... x | Adeb... x | Chat... x | CSS0... 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+... Negative Rings in In... Archive Extractor O... OpenGL\_simplegam...

All Bookmarks

### Student Data

Search by first/last name

ID	First Name	Last Name
2	PRECIOUS	ADELAKUN
3	ANNA	AFOAKWAH
4	JORDAN	BERTRAND
5	SYDNEY	BRUTUS
6	SIRAK	CHAMISO
7	DEVINE	CHINEMERE
8	VARUNKUMARREDDY	DODDA

Rows per page: 10 1-10 of 24 < >

Windows Search Type here to search

Taskbar icons: File Explorer, Edge, Mail, Calendar, Photos, Music, Settings, Chrome, VS Code, Word

System tray: Network, Volume, 9:36 PM 12/9/2024, Notifications

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 < >

Type here to search

9:36 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/#/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

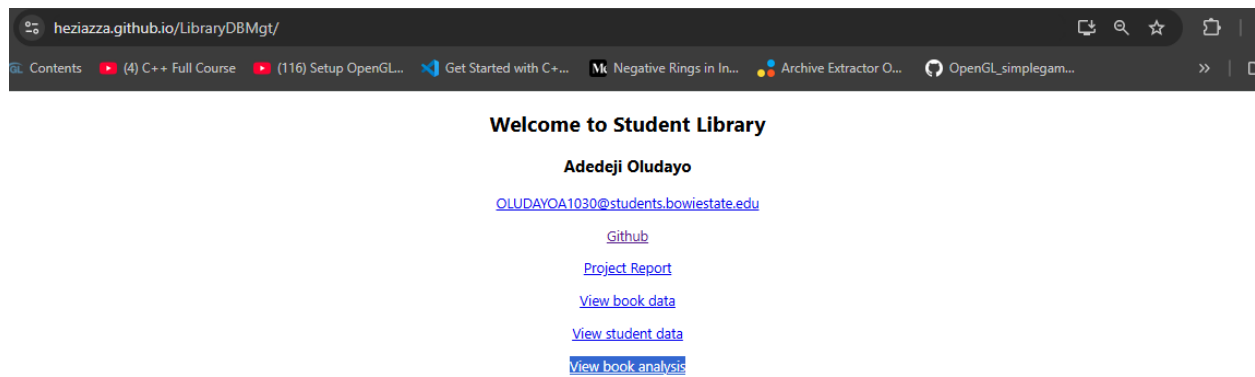
15	OLUWATOYIN	KODE
23	TEMITOPE	OYEMADE

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

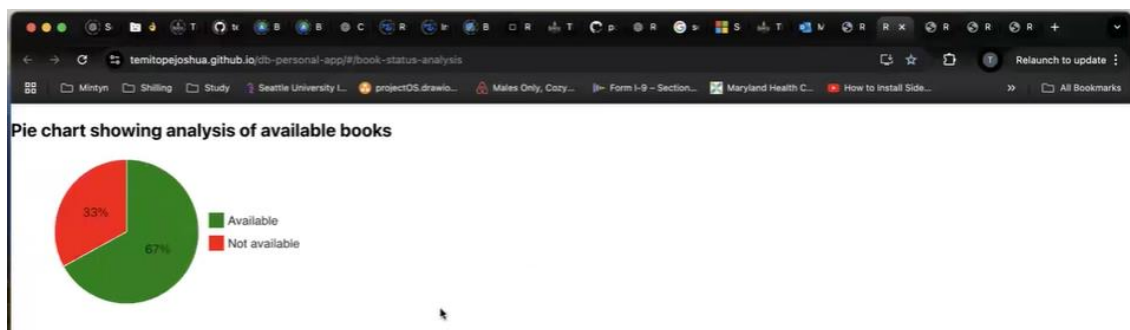
Type here to search

9:37 PM 12/9/2024

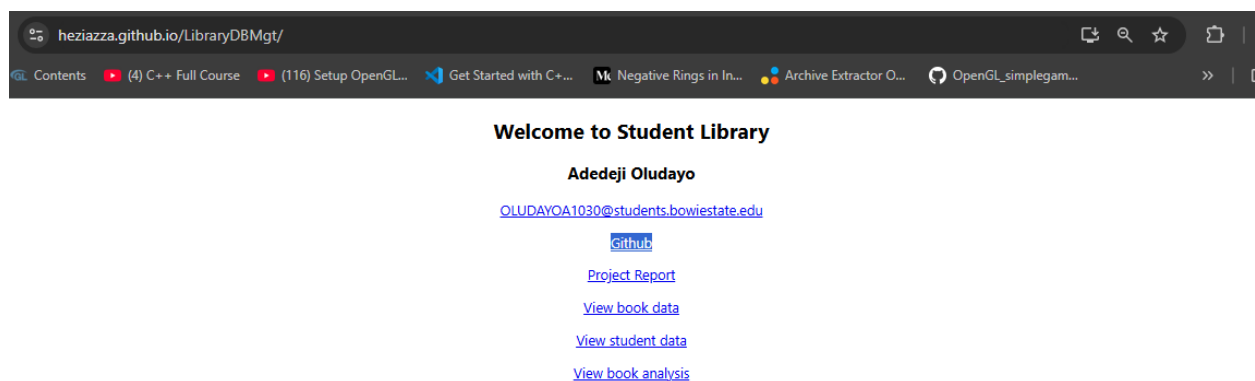
On the front end, Click on “View book analysis” to view the data analysis of book availability.



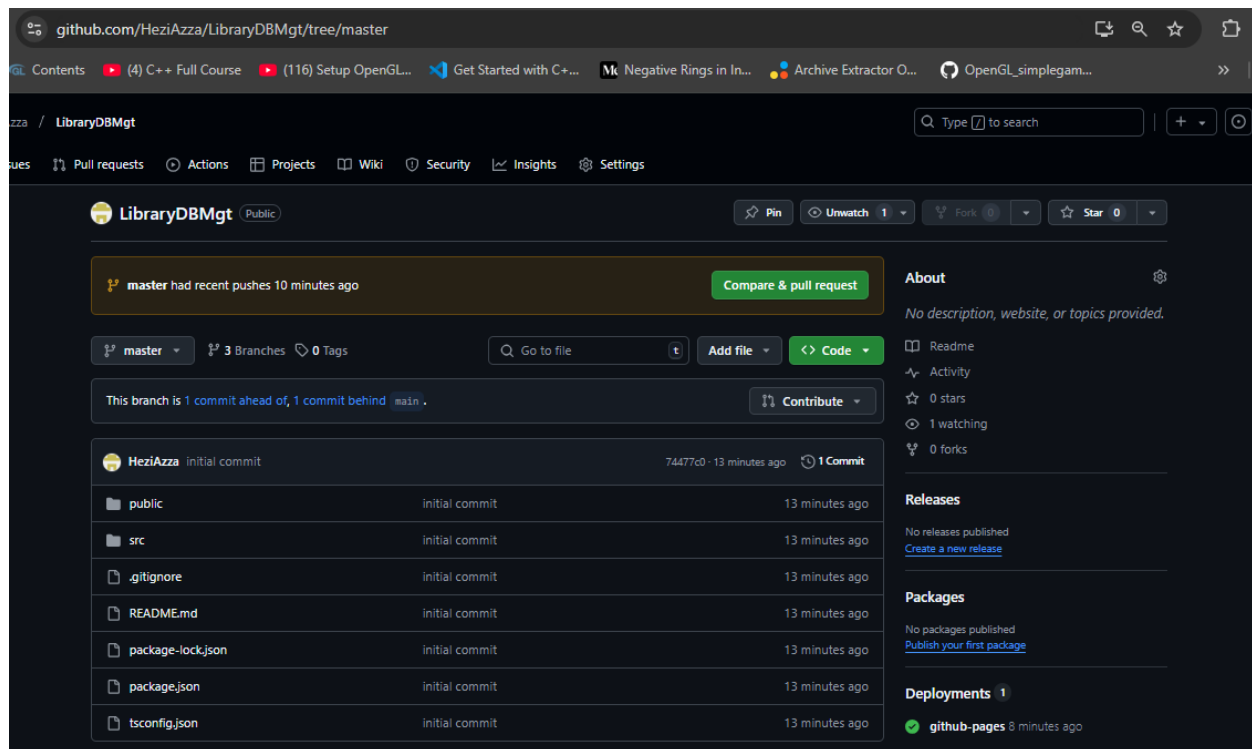
The pie chart shows a quick view of the percentage of books that are available and unavailable.



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

<https://heziazza.github.io/LibraryDBMgt/>

Source Code Repository Link: <https://github.com/HeziAzza/LibraryDBMgt>