

Library Management System for Group-Based Student Borrowing

Oluwatoyin Kode

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:

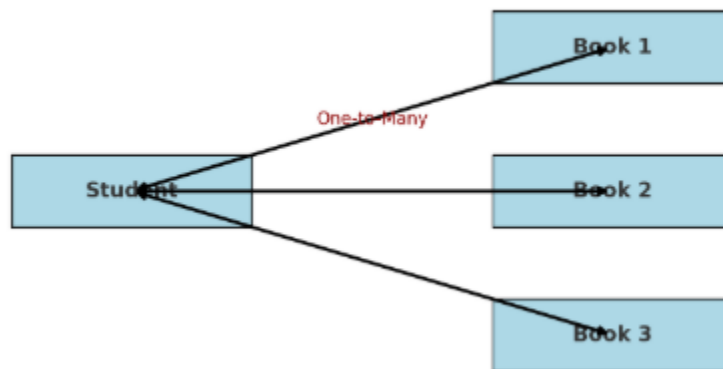
- A relational database to manage books, students, borrowing transactions, and group memberships.
- A user interface (UI) for querying borrowing details and viewing group information.
- A backend system to calculate average ratings and display books above or below the average.
- A method to dynamically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

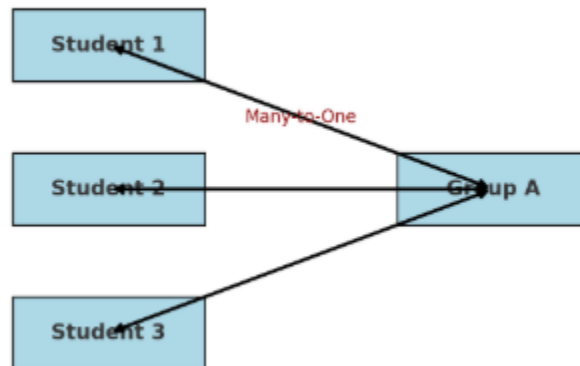
5. Design and Methodologies

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	978009796967	Booklet	15	paperback	112	2003-04-02 00:00:00	2003	1
2	978028071133	Wings	7	paperback	368	2003-05-12 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	2002-04-21 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	2002-04-08 00:00:00	2003	1
7	978191309780	Sail Down	5	paperback	200	2002-06-22 00:00:00	2003	1
8	978191309783	Paradise	9	paperback	118	2002-03-23 00:00:00	2002	1
9	978150484051	Heaven	15	paperback	140	2002-05-12 00:00:00	2002	1
10	978191126460	Love in the Big City	28	paperback	312	2001-05-28 00:00:00	2002	1
11	978191126460	Heavy Stones, Mostly	28	paperback	183	2001-12-02 00:00:00	2002	1
12	978190210878	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	2001-06-26 00:00:00	2002	1
14	978191309780	Paradise	11	paperback	222	2002-01-01 00:00:00	2002	1
15	978191309772	A New Name: Baptology V-VII	5	paperback	228	2001-09-08 00:00:00	2002	1
16	978191126461	Tomb of David	28	paperback	178	2001-09-12 00:00:00	2002	1
17	978191060503	The Books of Jacob	15	paperback	240	2001-11-15 00:00:00	2002	1
18	978191027782	Conrad Burns	35	paperback	251	2001-06-18 00:00:00	2002	1
19	978150484050	The War of the Pear	18	hardcover	98	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	978152540073	An Inventory of Losses	12	hardcover	296	2000-06-20 00:00:00	2001	1
22	978176270802	At Night All Wood is Black	17	hardcover	180	2000-11-03 00:00:00	2001	1
23	978030351435	I Live in the Status	48	hardcover	144	2000-03-18 00:00:00	2001	1
24	978191309772	Minor Detail	5	paperback	144	2000-05-08 00:00:00	2001	1
25	978191309773	Summer Brother	32	paperback	288	2001-04-08 00:00:00	2001	1
26	978180867601	The Pear Field	14	paperback	140	2000-10-30 00:00:00	2001	1
27	978030351435	The Perfect Rule	28	hardcover	140	2000-12-04 00:00:00	2001	1
28	978035154004	The Discomfort of Evening	4	paperback	240	2000-03-05 00:00:00	2000	1
29	978190040061	The Enlightenment of The Greenback	13	paperback	328	2000-01-17 00:00:00	2000	1

SELECT * FROM books WHERE status = 0;

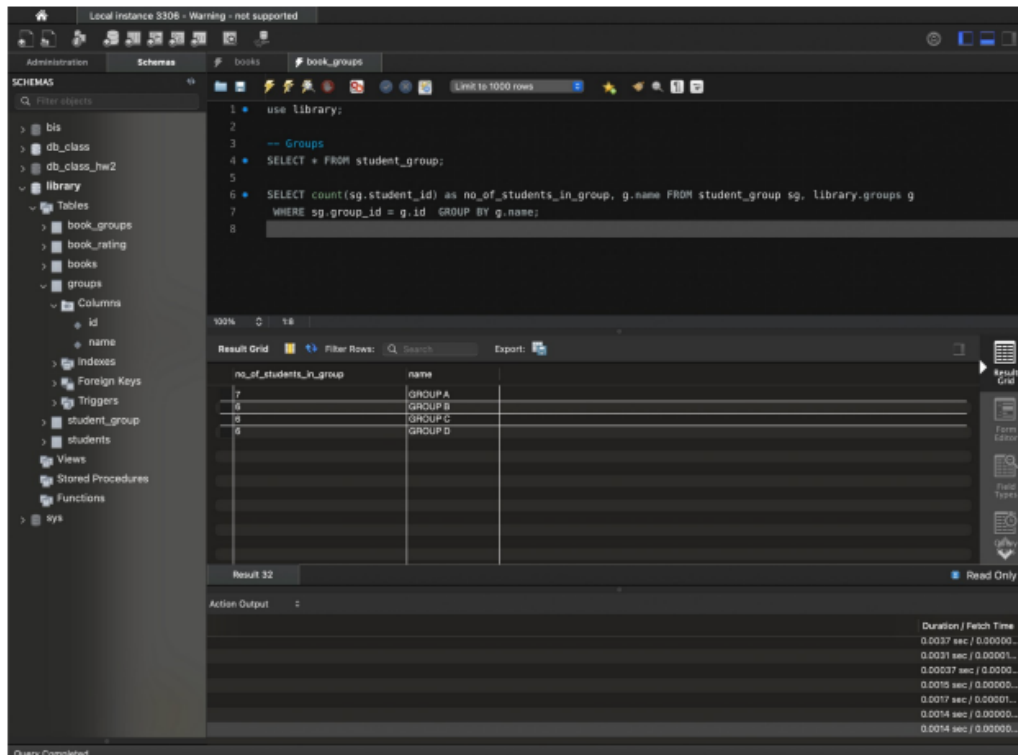
id	isbn	title	publisher_id	format	pages	published	year	status
4	978153941401	Standing Heavy	12	paperback	282	2003-05-26 00:00:00	2003	0
5	978163976401	So Many Good	28	paperback	342	2002-09-27 00:00:00	2003	0
6	978153942780	Jim Hendrix: Live in Live	12	paperback	419	2003-04-27 00:00:00	2003	0
7	978180427038	White We Were Dreaming	5	paperback	568	2003-03-30 00:00:00	2003	0
8	978171464872	A System So Magnificent It Is Bending	25	paperback	544	2003-03-30 00:00:00	2003	0
9	978176270802	North Building	8	paperback	212	2002-05-10 00:00:00	2002	0
10	978180636842	Evans Knows	2	paperback	143	2017-07-01 00:00:00	2002	0
11	978180605058	After the Sex	11	paperback	118	2001-06-24 00:00:00	2002	0
12	978171507802	Witchhunts	1	paperback	178	2000-06-02 00:00:00	2001	0
13	978191309780	In Memory of Memory	5	paperback	241	2001-03-07 00:00:00	2001	0
14	978176270801	The Gargles of Smoking in Bed	6	paperback	187	2001-04-01 00:00:00	2001	0
15	978180605058	The Employees	11	paperback	136	2000-10-01 00:00:00	2001	0
16	978171464804	The Adventures of China Iron	2	paperback	250	2019-11-14 00:00:00	2000	0
17	978191077484	The Light Life	25	hardcover	244	2019-11-14 00:00:00	2000	0
18	978181060613	The Other Names: Baptology I-II	5	paperback	381	2019-10-10 00:00:00	2000	0
19	978176270801	Servants	30	hardcover	350	2019-09-26 00:00:00	2000	0
20	978171150630	The Remembrance	1	paperback	180	2019-10-24 00:00:00	2019	0
21	978171150634	The News	1	paperback	233	2019-05-20 00:00:00	2019	0
22	978182496004	The Shaken Blythe	24	paperback	418	2017-08-26 00:00:00	2018	0
23	978030351507	The White Book	18	paperback	181	2018-04-25 00:00:00	2018	0
24	978176270801	The World Does On	28	paperback	320	2018-05-31 00:00:00	2018	0
25	978180636842	Let My Love	2	paperback	133	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;

```

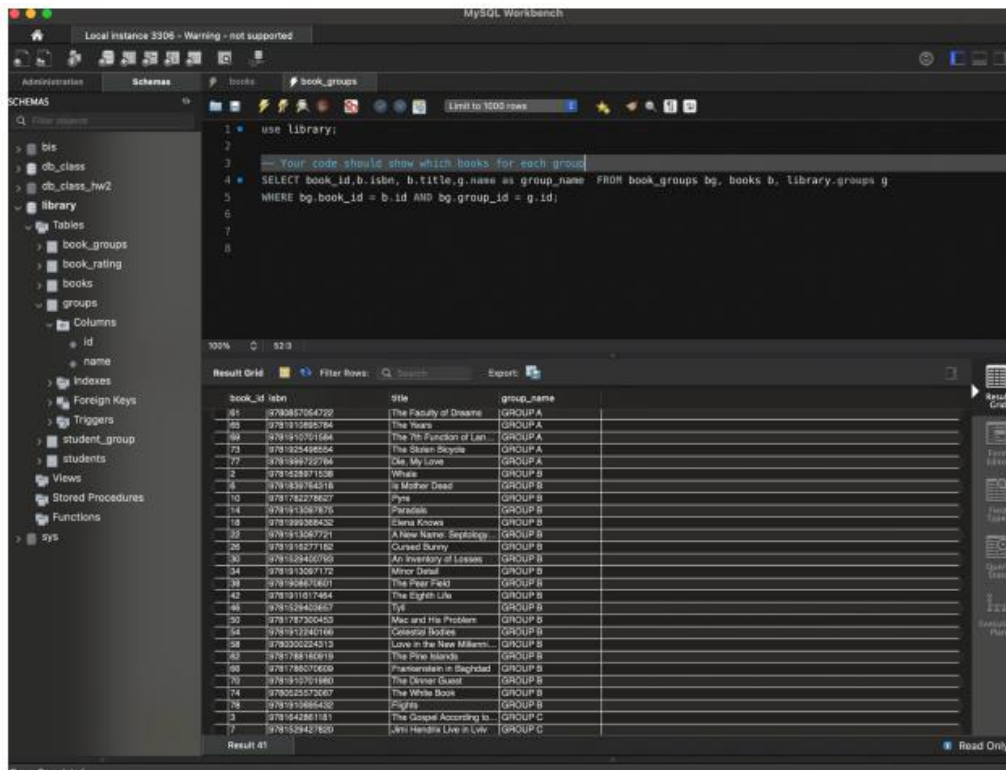


The screenshot shows a database management tool interface. On the left, a sidebar lists various database objects including 'library' and 'groups'. The main area displays a SQL query in a text editor. Below the editor, a 'Result Grid' shows the output of the query. The grid has two columns: 'no_of_students_in_group' 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 of the interface, there is an 'Action Output' section and a 'Duration / Fetch Time' section showing performance metrics for each row of the result set.

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;



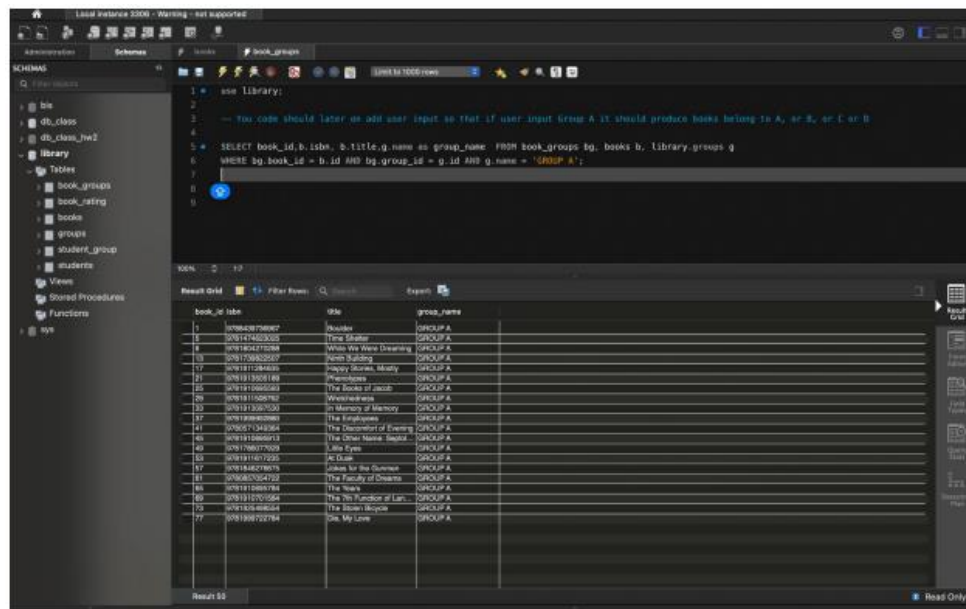
The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'library' database selected. The 'Query Editor' in the center contains the following SQL query:

```
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' at the bottom displays the results of the query, showing 41 rows. The columns are 'book_id', 'isbn', 'title', and 'group_name'. The results are grouped by 'group_name'.

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	0781099792784	Oh, My Love	GROUP A
12	0781028711538	White	GROUP B
18	0781039764518	is Mother Dead	GROUP B
10	0781782278627	Pine	GROUP B
14	0781073097615	Paradise	GROUP B
18	0781099366432	Elena Knows	GROUP B
23	0781013067721	A New Name: Septology	GROUP B
26	0781010277182	Cursed Bunny	GROUP B
33	0781029400182	An Inventory of Lies	GROUP B
34	0781013067722	Minor Detail	GROUP B
38	0781030870601	The Peer Field	GROUP B
42	0781010174654	The Eighth Life	GROUP B
44	0781029400183	Type	GROUP B
50	0781787300453	Mac and His Problem	GROUP B
54	0781012340166	Celestial Bodies	GROUP B
58	0780302245113	Love in the New Millenn...	GROUP B
62	0781788166619	The Pine Islands	GROUP B
68	0781788070658	Phantom in Baghdad	GROUP B
70	0781010761660	The Dinner Guest	GROUP B
74	078025573067	The White Book	GROUP B
76	0781010986432	Phages	GROUP B
78	0781042861181	The Clasp According to...	GROUP C
7	0781529427850	Lani Hendrix 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. The 'Query Editor' in the center 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' at the bottom displays the results of the query, showing 22 rows. The columns are 'book_id', 'isbn', 'title', and 'group_name'. The results are filtered to show only books belonging to 'GROUP A'.

book_id	isbn	title	group_name
1	078040278967	Skuller	GROUP A
8	0781010895784	The Years	GROUP A
18	0781025486854	White We Were Dreaming	GROUP A
19	0781788070658	White Building	GROUP A
17	0781010761664	Empy Stone, Empty	GROUP A
21	0781010761664	Phenomena	GROUP A
25	0781010761664	The Book of Jacob	GROUP A
26	0781010761664	Reverence	GROUP A
33	0781010761664	In Memory of Memory	GROUP A
17	0781010761664	The Englishman	GROUP A
41	078051348364	The Descent of Evening	GROUP A
46	0781010761664	The Other Name: Sept...	GROUP A
48	0781788070658	Is the First	GROUP A
55	0781010761664	Acorn	GROUP A
57	0781040278627	Green in the Summer	GROUP A
18	078040278967	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	0781099792784	Oh, My Love	GROUP A

Local Instance 3306 - Warning - not supported

Administration Schemas book_group

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';
```

71.6

book_id	isbn	title	group_name
3	0761608071038	White	GROUP B
4	0761608071039	To Another Road	GROUP B
10	076176279027	Pyre	GROUP B
14	0761913067610	Parasite	GROUP B
18	0761909484418	Down Arrow	GROUP B
20	0761913067781	A New Name, Segments of the	GROUP B
26	0761918277182	Ground Runny	GROUP B
30	0761528601013	An Invention of a Lesson	GROUP B
34	0761913067773	Minor Detail	GROUP B
38	0761528601014	The Paper Fold	GROUP B
42	0761710716444	The Lullaby Life	GROUP B
46	0761528601017	Ty	GROUP B
50	0761762790283	Mind and the Problem	GROUP B
54	0761913067884	Celestial Rules	GROUP B
58	0761528601015	Love in the New Millennium	GROUP B
62	0761762790289	The First Invention	GROUP B
66	0761762790290	Parasite in Baghdad	GROUP B
70	0761913067880	The Outer Guest	GROUP B
74	0761608071047	The White Book	GROUP B
78	0761913067832	Figura	GROUP B

Result 01

Read Only

Local Instance 3306 - Warning - not supported

Administration Schemas book_group

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';
```

1.8

book_id	isbn	title	group_name
15	0761040801101	The Gospel According to the New World	GROUP C
17	0761028427820	Jane Mendon Live in Life	GROUP C
19	0761913067680	Self Germ	GROUP C
23	0761040801101	Heaven	GROUP C
25	0761062110180	The Book of Mafar	GROUP C
29	0761062090302	After the Sea	GROUP C
33	0761028601030	The War of the Peace	GROUP C
37	0761762790282	All Night All Blood is Black	GROUP C
41	0761913067779	Summer Brother	GROUP C
45	0761028601031	The Perfect Film	GROUP C
49	0761028601031	The Employment of The Greenage	GROUP C
53	0761062090303	Parasite on the Top of My Tongue	GROUP C
57	0761762790283	Red Dog	GROUP C
61	0761913067810	Drive Your Head Over the Bones of the	GROUP C
65	0761762790289	He, and, of, the	GROUP C
69	0761913067839	The Penetration	GROUP C
73	0761040801102	Go, West, Come	GROUP C
77	0761028601032	The Flying Machine	GROUP C
81	0761762790284	The World Goes On	GROUP C

Result 02

Read Only

Local Instance 2020 - Warning: not supported

Schemas: Administration, Schemas, book_groups

Query Editor:

```

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, 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';

```

Result Grid:

book_id	isbn	title	group_name
1	9787028414421	Standing Heavy	GROUPD
2	9787028414422	The Stranger Party	GROUPD
3	9787028414423	A System to Manage a System	GROUPD
4	9787028414424	Love in the Big City	GROUPD
5	9787028414425	More Than I Love My Life	GROUPD
6	9787028414426	Bank of Basel	GROUPD
7	9787028414427	When We Came to Understand the World	GROUPD
8	9787028414428	I Live in the Storm	GROUPD
9	9787028414429	The Dangers of Creating in Bed	GROUPD
10	9787028414430	The Architecture of Dark Iron	GROUPD
11	9787028414431	The Memory Police	GROUPD
12	9787028414432	Hurricane Season	GROUPD
13	9787028414433	Reveries	GROUPD
14	9787028414434	Four Seasons	GROUPD
15	9787028414435	The Book of Moral Tales	GROUPD
16	9787028414436	The Shape of the Ruins	GROUPD
17	9787028414437	Like a Facing Shadow	GROUPD
18	9787028414438	The Inevitable	GROUPD
19	9787028414439	Version Sublime I	GROUPD

- 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;

Local Instance 2020 - Warning: not supported

Schemas: Administration, Schemas, book_groups

Query Editor:

```

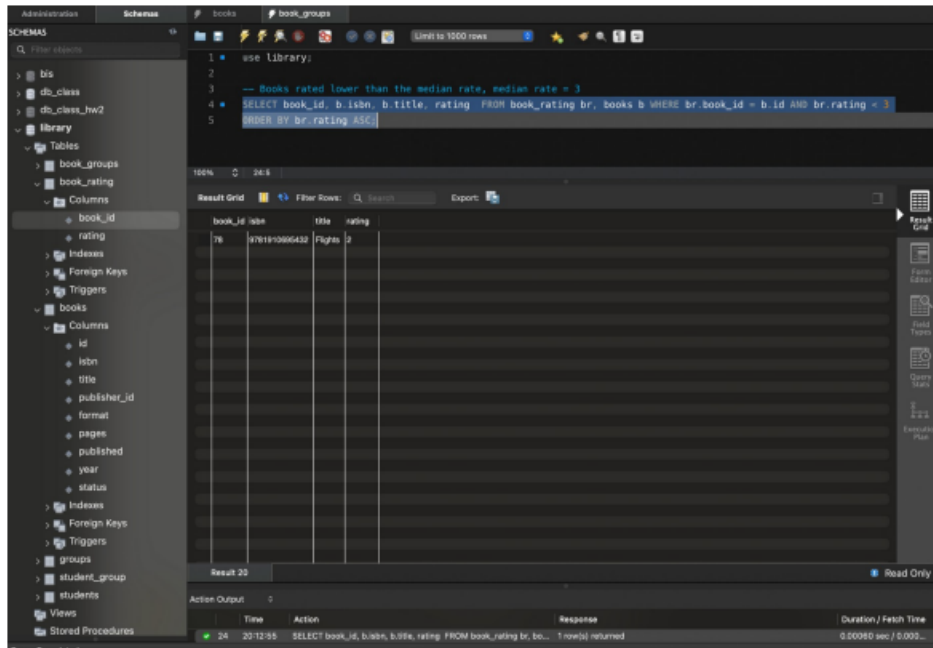
1 use library;
2
3 -- Books rated higher than the median rate, median rate = 3
4 SELECT book_id, b.isbn, b.title, rating FROM book_rating br, books b WHERE br.book_id = b.id AND br.rating > 3
5 ORDER BY br.rating ASC;

```

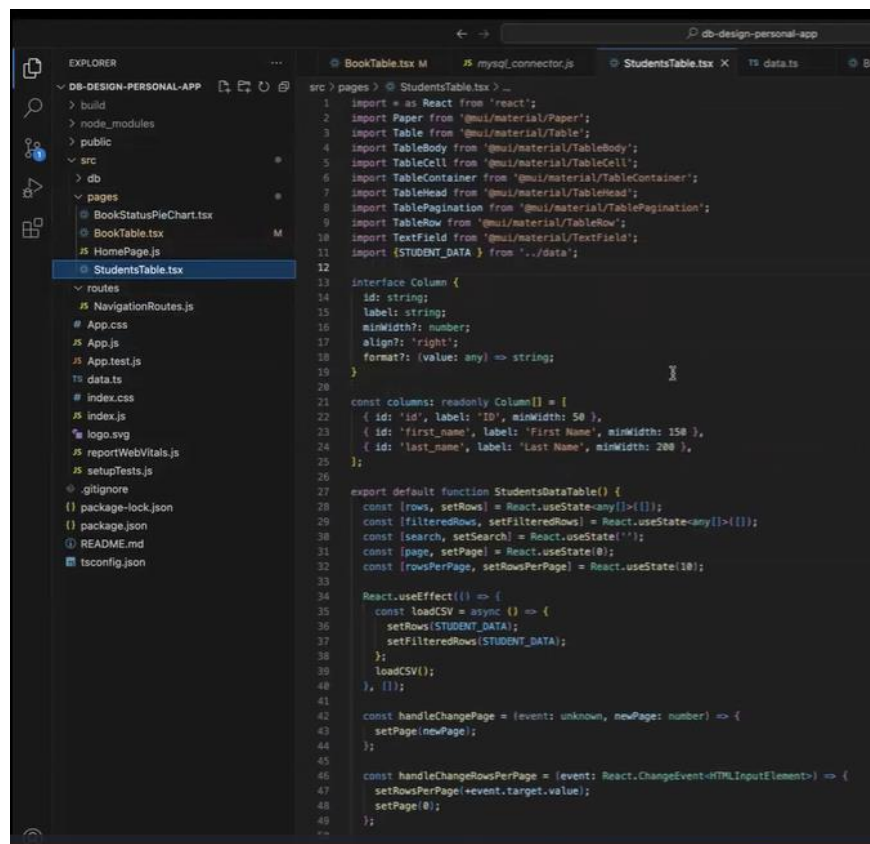
Result Grid:

book_id	isbn	title	rating
20	9781808782851	The Perfect Nine	4
21	9781808782852	The Gospel According to the New World	4
22	9781808782853	In Mother's Land	4
23	9781808782854	Jerry Hansen: Live in Life	4
24	9781808782855	The Skyline Party	4
25	9781808782856	White We Were Unearthing	4
26	9781808782857	Pyre	4
27	9781808782858	Version Sublime I	4
28	9781808782859	The White Book	4
29	9781808782860	Paradise	4
30	9781808782861	Diana Knows	4
31	9781808782862	The Book of Mother	4
32	9781808782863	More Than I Love My Life	4
33	9781808782864	The Broken Bicycle	4
34	9781808782865	Home of Sand	4
35	9781808782866	The Imposter	4
36	9781808782867	The Piping Mountain	4
37	9781808782868	An Inventory of Losses	4
38	9781808782869	At Night All Roads to Black	4
39	9781808782870	I Live in the Storm	4
40	9781808782871	In Memory of Memory	4
41	9781808782872	Minor Detail	4
42	9781808782873	Summer Brother	4
43	9781808782874	The Employee	4
44	9781808782875	The Four Field	4
45	9781808782876	Big My Love	4
46	9781808782877	The Succession of Evening	4
47	9781808782878	The Eight Life	4
48	9781808782879	The Memory Police	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;



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

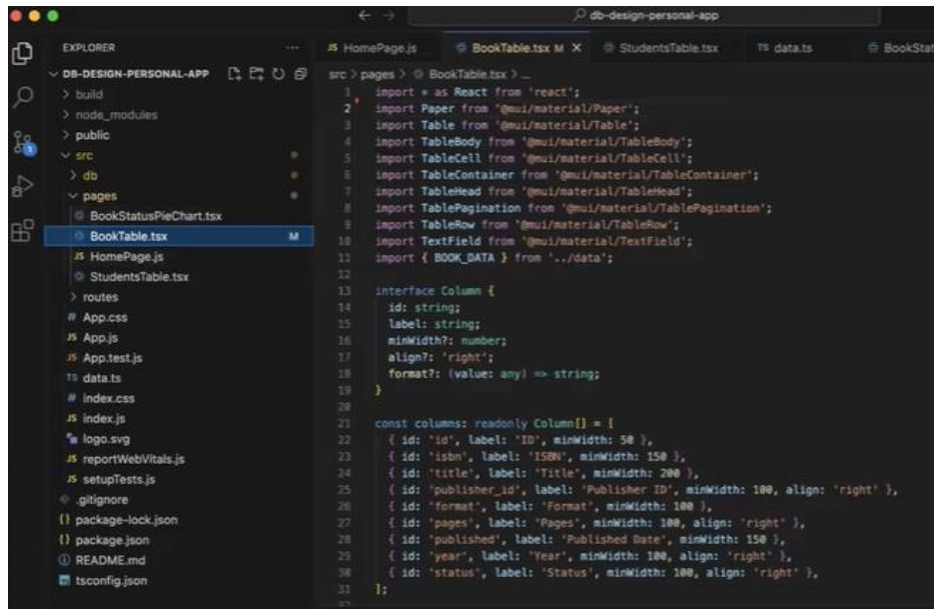



```
src > pages > StudentsTable.tsx > ...
27 export default function StudentsTable() {
58
59   const handleSearch = (event: React.ChangeEvent<HTMLInputElement>) => {
60     const value = event.target.value.toLowerCase();
61     setSearch(value);
62     setFilteredRows(rows.filter(row =>
63       row.first_name.toLowerCase().includes(value) ||
64       row.last_name.toLowerCase().includes(value)
65     ));
66   };
67
68   return (
69     <div>
70       <h2>Student Data</h2>
71       <TextField
72         label="Search by first/last name"
73         variant="outlined"
74         value={search}
75         onChange={handleSearch}
76         sx={{ marginBottom: 2 }}
77       />
78       <Paper sx={{ width: '100%', overflow: 'hidden' }}>
79         <TableContainer sx={{ maxHeight: 440 }}>
80           <Table stickyHeader aria-label="sticky table">
81             <TableHead>
82               <TableRow>
83                 {columns.map((column) => (
84                   <TableCell
85                     key={column.id}
86                     align={column.align}
87                     style={{ minWidth: column.minWidth }}
88                   >
89                     {column.label}
90                   </TableCell>
91                 ))}
92             </TableHead>
93             <TableBody>
94               {filteredRows
95                 .slice(page * rowsPerPage, page * rowsPerPage + rowsPerPage)
96                 .map((row, index) => (
97                   <TableRow hover role="checkbox" tabIndex={-1} key={index}>
98                     {columns.map((column) => (
99                       const value = row[column.id];
100                       return (
101                         <TableCell key={column.id} align={column.align}>
102                           {column.format ? column.format(value) : value}
103                         </TableCell>
104                       );
105                     ))}
106                   </TableRow>
107                 ))}
108             </TableBody>
109           </Table>
110         </TableContainer>
111       </Paper>
112     </div>
113   );
114 }
```

- Book data Model:

```
src > ts > data.ts > @00 BOOK_DATA
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    "status": "Available"
34  },
35  {
36    "id": 4,
37    "isbn": "9781529434431",
38    "title": "Standing Heavy",
39    "publisher_id": 12,
40    "format": "paperback",
41    "pages": 252,
42    "published": "2022-05-26 00:00:00",
43    "year": 2023,
44    "status": "Not Available"
45  },
46  {
47    "id": 5,
48    "isbn": "9781474623025",
49    "title": "Time Shelter",
50    "publisher_id": 30,
51    "format": "paperback",
52    "pages": 112,
53    "published": "2022-08-02 00:00:00",
54    "year": 2023,
55    "status": "Available"
56  }
57 ]
```

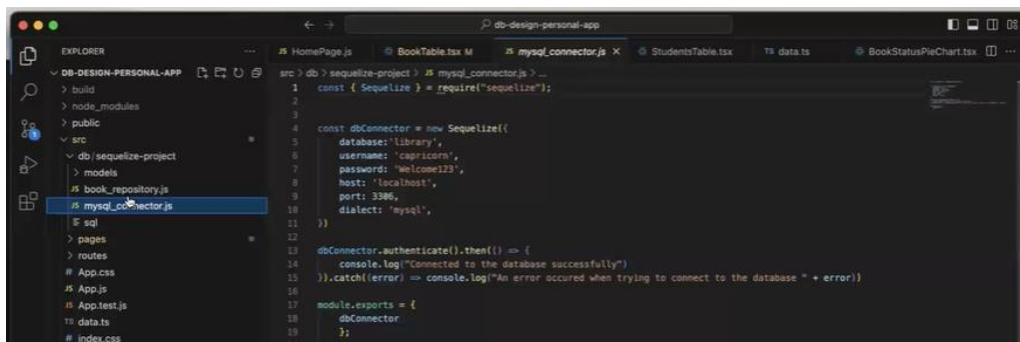
- Book Table Model:



```

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 ];
  
```

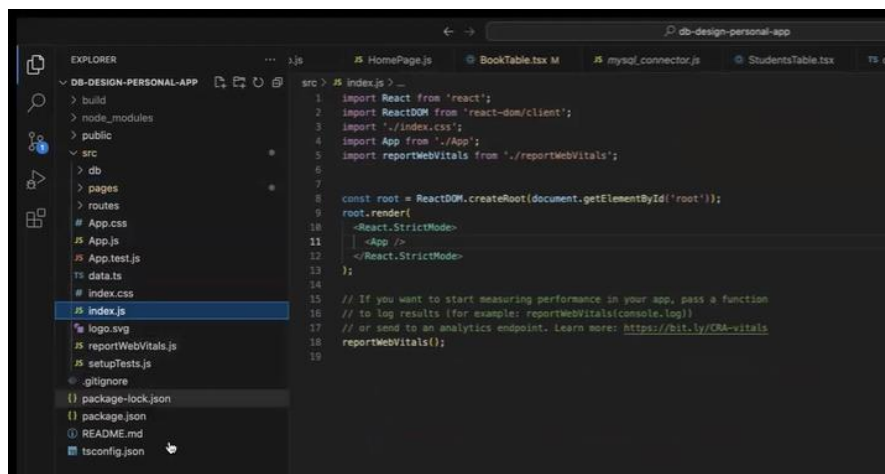
- Database Connector:



```

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 };
  
```

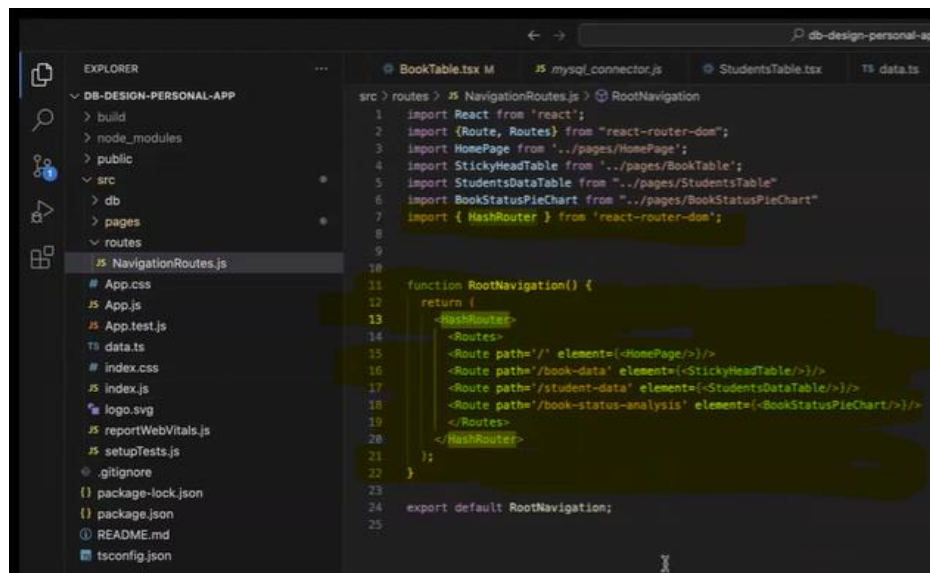
- Routers: These were used to connect the front-end our backend database
 - React: The front end was built with React



```

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();
  
```

- HashRoute: HashRoute was imported for connectivity with GitHub



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'DB-DESIGN-PERSONAL-APP' with folders like 'build', 'node_modules', 'public', 'src', 'db', 'pages', and 'routes'. The 'routes' folder is expanded, showing 'NavigationRoutes.js'. The code editor shows the content of 'NavigationRoutes.js', which includes imports for 'React', 'Route', 'Routes', 'HomePage', 'StickyHeadTable', 'StudentsDataTable', 'BookStatusPieChart', and 'HashRouter'. The code defines a 'RootNavigation' function that returns a JSX element with a 'HashRouter' component and four 'Route' elements. The routes are: '/' (HomePage), '/book-data' (StickyHeadTable), '/student-data' (StudentsDataTable), and '/book-status-analysis' (BookStatusPieChart). The function is exported as 'RootNavigation'.

5.2. Front-end Demonstration

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

toyeentop.github.io/Library-Project/

Welcome to Oluwatoyin's Personal Website

Contact me: toyeentop@yahoo.com

[LinkedIn](#)

[Github](#)

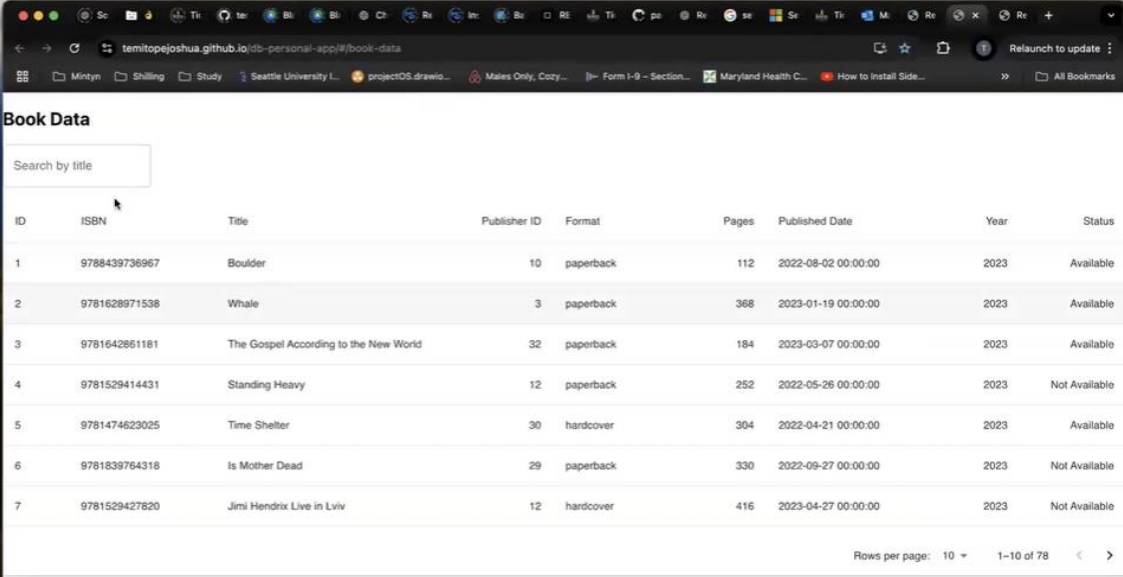
[Project Report](#)

[View book data](#)

[View student data](#)

[View book analysis](#)

Webpage displaying Book Data

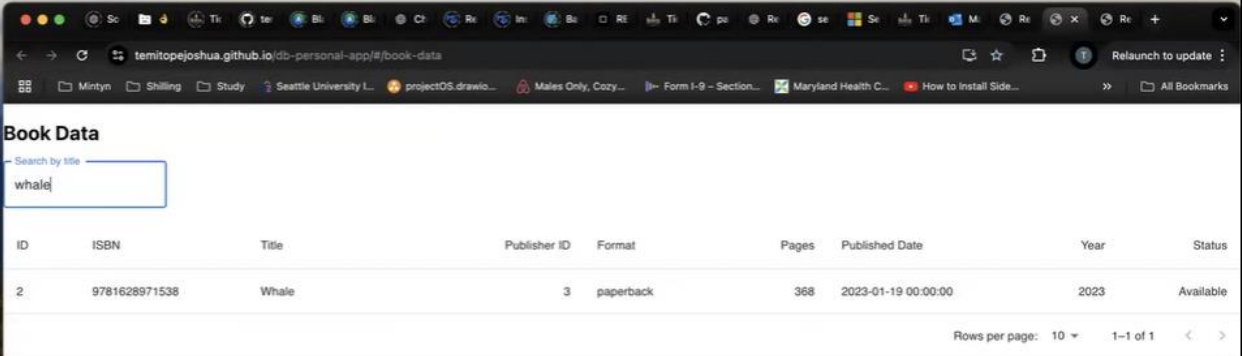


The screenshot shows a web browser window with the URL `temitopejoshua.github.io/db-personal-app/#/book-data`. The page title is "Book Data". Below the title is a search bar labeled "Search by title". Below the search bar is a table with 9 columns: ID, ISBN, Title, Publisher ID, Format, Pages, Published Date, Year, and Status. The table contains 7 rows of book data.

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

At the bottom right of the table, there is a pagination control showing "Rows per page: 10" and "1-10 of 78".

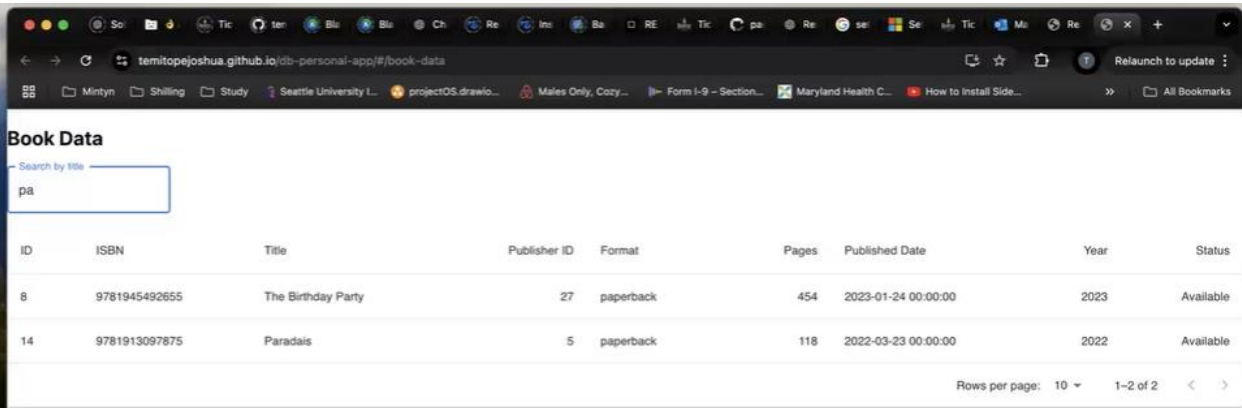
Search by Title – for example Whale, or Paradais – as you begin to type, it begins to filter out your search



The screenshot shows the same web browser window, but the search bar now contains the text "whale". The table below the search bar now only displays one row of data, which is the book "Whale".

ID	ISBN	Title	Publisher ID	Format	Pages	Published Date	Year	Status
2	9781628971538	Whale	3	paperback	368	2023-01-19 00:00:00	2023	Available

At the bottom right of the table, the pagination control shows "Rows per page: 10" and "1-1 of 1".



The screenshot shows the same web browser window, but the search bar now contains the text "pa". The table below the search bar now displays two rows of data, which are the books "The Birthday Party" and "Paradais".

ID	ISBN	Title	Publisher ID	Format	Pages	Published Date	Year	Status
8	9781945492655	The Birthday Party	27	paperback	454	2023-01-24 00:00:00	2023	Available
14	9781913097875	Paradais	5	paperback	118	2022-03-23 00:00:00	2022	Available

At the bottom right of the table, the pagination control shows "Rows per page: 10" and "1-2 of 2".

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

toyeentop.github.io/Library-Project/



Welcome to Oluwatoyin's Personal Website

Contact me: toyeentop@yahoo.com

[LinkedIn](#)

[Github](#)

[Project Report](#)

[View book data](#)

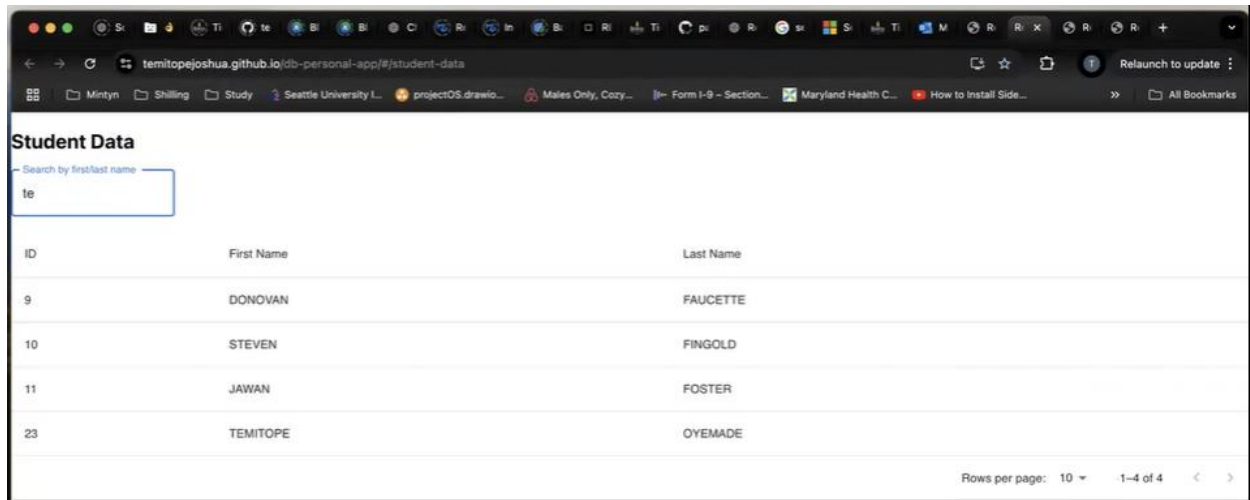
[View student data](#)

[View book analysis](#)

Webpage displaying Student Data

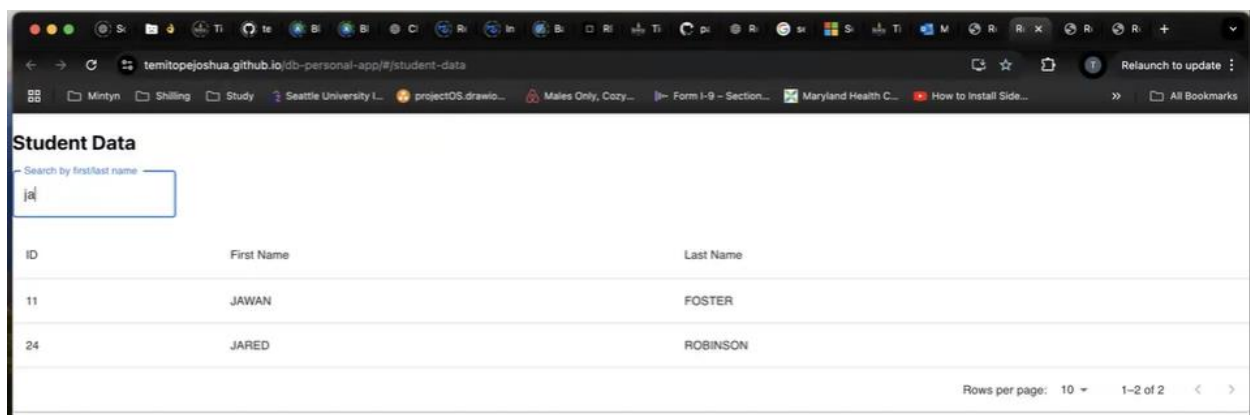
ID	First Name	Last Name
2	PRECIOUS	ADELAUN
3	ANNA	AFOAKWAH
4	JORDAN	BERTRAND
5	SYDNEY	BRUTUS
6	SIRAK	CHAMISO
7	DEVINE	CHINEMERE
8	VARUNKUMARREDDY	DODDA

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



The screenshot shows a web browser window with the URL `temitopejoshua.github.io/db-personal-app/#/student-data`. The page title is "Student Data". There is a search bar labeled "Search by first/last name" with the text "te" entered. Below the search bar is a table with three columns: "ID", "First Name", and "Last Name". The table contains four rows of data. At the bottom right of the table, it says "Rows per page: 10" and "1-4 of 4".

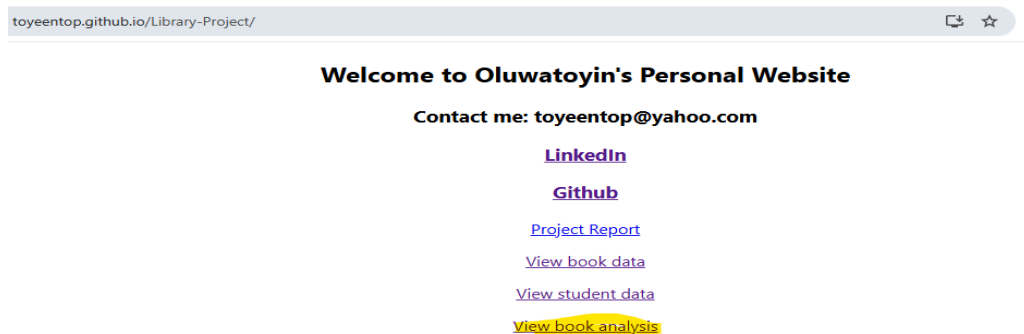
ID	First Name	Last Name
9	DONOVAN	FAUCETTE
10	STEVEN	FINGOLD
11	JAWAN	FOSTER
23	TEMITOPE	OYEMADE



The screenshot shows the same web browser window as the previous one, but with the search bar containing "ja". The table now displays two rows of data. At the bottom right of the table, it says "Rows per page: 10" and "1-2 of 2".

ID	First Name	Last Name
11	JAWAN	FOSTER
24	JARED	ROBINSON

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



The screenshot shows a web browser window with the URL `toyeentop.github.io/Library-Project/`. The page has a dark background with white text. It says "Welcome to Oluwatoyin's Personal Website" and "Contact me: toyeentop@yahoo.com". Below this are several links: "LinkedIn", "Github", "Project Report", "View book data", "View student data", and "View book analysis" (which is highlighted in yellow).

toyeentop.github.io/Library-Project/

Welcome to Oluwatoyin's Personal Website

Contact me: toyeentop@yahoo.com

[LinkedIn](#)

[Github](#)

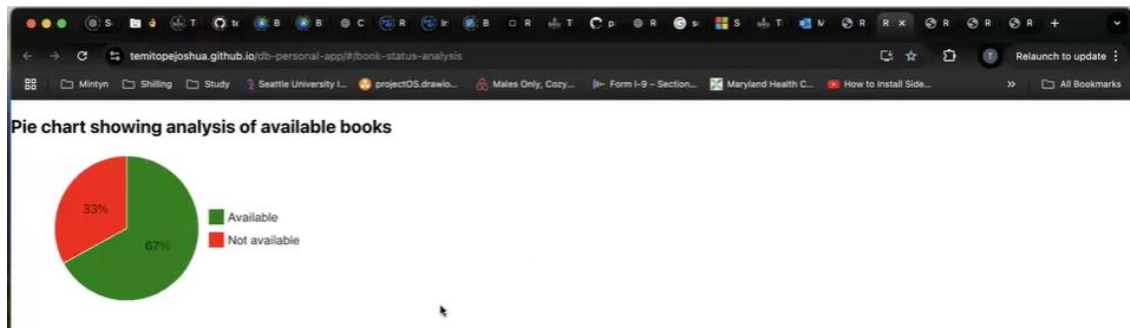
[Project Report](#)

[View book data](#)

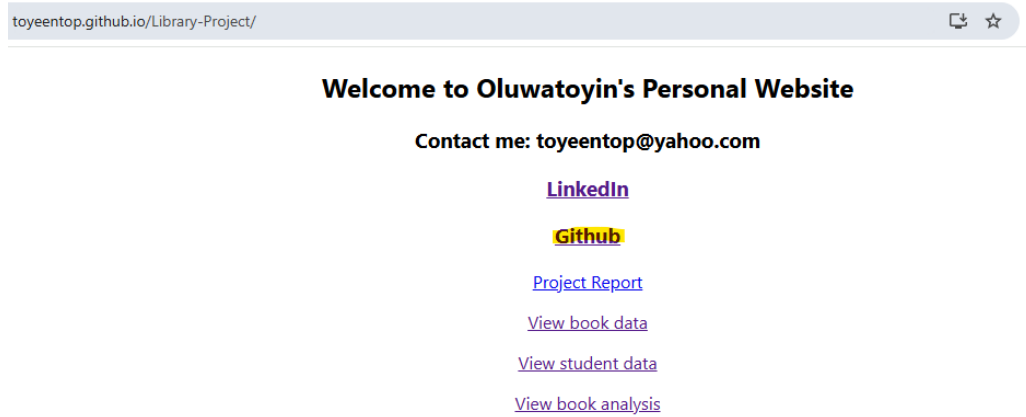
[View student data](#)

[View book analysis](#)

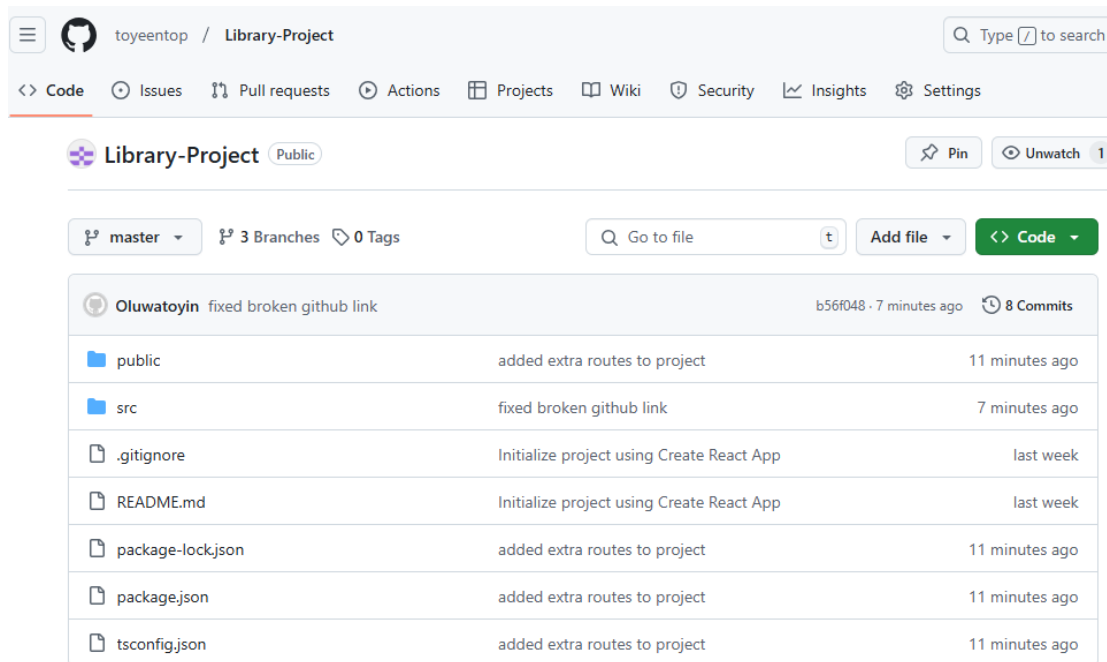
The pie chart shows a quick view of the percentage of books that are available and unavailable.



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: <https://github.com/toyeentop/Library-Project>