

Library Management System for Group-Based Student Borrowing

Edith Jiokengbogni

<https://ebogni17.github.io/LibraryManagement/>

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:

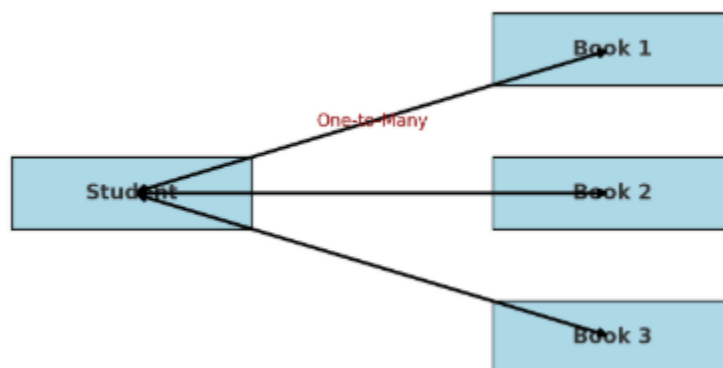
- A relational database to manage books, students, borrowing transactions, and group memberships.
- A user interface (UI) for querying borrowing details and viewing group information.
- A backend system to calculate average ratings and display books above or below the average.
- A method to dynamically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

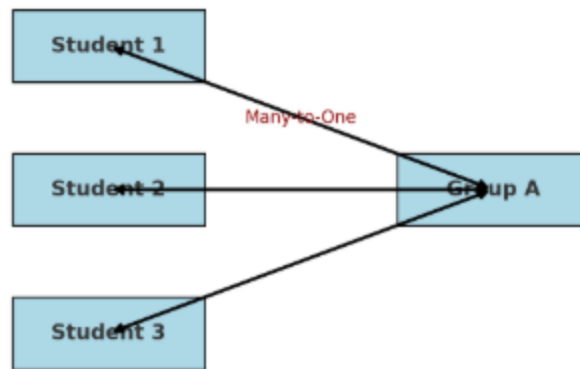
5. Design and Methodologies

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	978452870887	Shutter	18	paperback	112	2023-08-02 00:00:00	2023	1
2	978102871133	White	2	paperback	368	2023-07-18 00:00:00	2023	1
3	9781042801181	The Gospel According to the New World	52	paperback	184	2023-03-07 00:00:00	2023	1
4	9781017465325	Time Shelter	26	hardcover	324	2022-04-27 00:00:00	2023	1
5	9787945452655	The Birthday Party	27	paperback	464	2023-07-04 00:00:00	2023	1
6	9781782778227	Yes	17	paperback	324	2023-04-08 00:00:00	2023	1
7	9781013037880	Salt Skin	5	paperback	256	2022-06-22 00:00:00	2023	1
8	9781013091875	Paradise	15	paperback	178	2023-09-09 00:00:00	2023	1
9	9781050846251	Heaven	18	paperback	147	2023-05-12 00:00:00	2023	1
10	9781011324525	Love in the Big City	28	paperback	317	2021-10-20 00:00:00	2022	1
11	9781011284525	Shake Shakes, Mandy	28	paperback	183	2021-12-02 00:00:00	2022	1
12	9781011284525	The Book of Mother	21	paperback	324	2021-09-18 00:00:00	2023	1
13	9781011284525	More Than I Love My Life	9	paperback	268	2021-08-28 00:00:00	2023	1
14	9781013050180	Phenotypes	1	paperback	232	2023-01-07 00:00:00	2023	1
15	9781013057722	A New Name, Baptiste M. V.B.	5	paperback	228	2021-09-08 00:00:00	2023	1
16	9781013057722	Temp of Sand	28	paperback	728	2021-08-12 00:00:00	2023	1
17	978101060533	The Book of Jacob	15	paperback	320	2021-11-15 00:00:00	2023	1
18	978101060533	Cursed Burns	5	paperback	293	2021-06-15 00:00:00	2023	1
19	978102846558	The War of the Phase	18	hardcover	95	2021-07-02 00:00:00	2023	1
20	9781782378128	When We Cease to Understand the W...	17	hardcover	186	2020-09-03 00:00:00	2023	1
21	9781028400723	An Inventory of Losses	12	hardcover	256	2020-08-28 00:00:00	2023	1
22	9781782378882	All Night All Blood is Black	17	hardcover	182	2020-11-05 00:00:00	2023	1
23	9781028400723	Live in the State	28	hardcover	344	2020-09-25 00:00:00	2023	1
24	9781013057722	Minor Detail	5	paperback	144	2020-05-06 00:00:00	2023	1
25	9781013057722	Summer Brother	32	paperback	268	2021-04-08 00:00:00	2023	1
26	9781088670627	The Pear Field	14	paperback	160	2020-10-02 00:00:00	2023	1
27	9781028400723	The Perfect Name	28	hardcover	340	2020-10-02 00:00:00	2023	1
28	9781013057722	The Discomfort of Evening	4	paperback	382	2020-03-05 00:00:00	2023	1
29	9781008400651	The Enlightenment of The Greenhouse...	13	paperback	328	2020-01-17 00:00:00	2023	1

`SELECT * FROM books WHERE status = 0;`

The screenshot shows a database management interface with a query editor and a results grid. The query executed is `SELECT * FROM books WHERE status = 0`. The results grid displays 27 rows of book data, including columns for id, isbn, title, publisher_id, format, pages, published, year, and status.

id	isbn	title	publisher_id	format	pages	published	year	status
4	976183941430	Standing Heavy	12	paperback	263	2020-05-20 08:00:00	2020	0
6	9761839764318	To Mother Dead	28	paperback	335	2020-09-27 06:00:00	2020	0
7	9761839427820	Just Handle Live in Law	12	hardcover	418	2020-04-27 06:00:00	2020	0
8	9761839417228	With the Year's Straining	5	paperback	269	2020-07-30 06:00:00	2020	0
12	9761814664572	A Soldier So Magnificent & So Blending	20	paperback	364	2020-03-09 06:00:00	2020	0
13	9761758625507	North Building	8	paperback	272	2020-05-16 06:00:00	2020	0
16	9761808368432	Elms Known	2	paperback	143	2021-01-01 06:00:00	2020	0
20	9761809099809	After the Best	11	paperback	178	2021-03-01 06:00:00	2020	0
29	9761811560782	Witchhazel	1	paperback	176	2020-04-02 06:00:00	2021	0
33	9761813007536	In Memory of Memory	5	paperback	310	2021-02-17 06:00:00	2021	0
36	9761763766718	The Dangers of Smoking in Bed	6	paperback	187	2021-04-01 06:00:00	2021	0
37	9761809099809	The Employee	11	paperback	178	2020-03-01 06:00:00	2021	0
40	9761816406084	The Adventures of China Inn	2	paperback	260	2018-11-14 06:00:00	2020	0
42	9761811017484	The Light Life	39	hardcover	344	2018-11-14 06:00:00	2020	0
46	9761811066813	The Other Name: Epitaphy I	6	paperback	383	2018-10-10 06:00:00	2020	0
52	9761763766718	Jerusalem	35	hardcover	369	2018-09-20 06:00:00	2020	0
63	9761811560526	The Remains	1	paperback	180	2018-10-04 06:00:00	2018	0
66	9761811066574	The Years	2	paperback	252	2018-05-20 06:00:00	2018	0
70	9761808368432	The Queen Bicycle	24	paperback	416	2017-09-26 06:00:00	2018	0
74	9760202972907	The White Book	16	paperback	183	2018-05-25 06:00:00	2018	0
75	9761768166134	The World (Quee On	28	paperback	320	2018-09-01 06:00:00	2018	0
77	9761809722764	On My Love	8	paperback	123	2017-09-14 06:00:00	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 'SCHEMAS' tree lists various database objects, including 'library' and 'book_groups'. The main area displays a SQL query in a dark-themed 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 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, an 'Action Output' pane shows the execution details, including 'Duration / 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;

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'library' selected. The main editor contains a SQL query: `use library; -- Your code should show which books for each group -- SELECT book_id,b.isbn, b.title,g.name as group_name FROM book_groups bg, books b, library.groups g WHERE bg.book_id = b.id AND bg.group_id = g.id;`. The 'Result Grid' at the bottom shows 41 rows of data with columns: book_id, isbn, title, and group_name. The data is grouped by group_name, with GROUP A having 18 rows, GROUP B having 18 rows, and GROUP C having 5 rows.

book_id	isbn	title	group_name
181	9780857064729	The Faculty of Dreams	GROUP A
485	9781815885184	The Years	GROUP A
58	9781910261686	The 7th Function of Lan...	GROUP A
73	9781825488854	The Seven Moons	GROUP A
77	9781844792784	On My Love	GROUP A
12	9781528971538	White	GROUP B
8	9781839764218	is Mother Dead	GROUP B
110	9781742276257	Pigs	GROUP B
114	9781913267876	Parables	GROUP B
18	9781299368432	Elena Knows	GROUP B
29	9781913267721	A New Name: Septology	GROUP B
25	9781918277182	Curved Runny	GROUP B
30	9781912446785	An Inventory of Lissas	GROUP B
34	9781913267172	Minor Detail	GROUP B
38	9781924871401	The Pear Field	GROUP B
42	9781911817484	The Eager Liar	GROUP B
48	9781529432657	Ty	GROUP B
50	9781787304453	Mac and His Problem	GROUP B
54	9781912240166	Celestial Bodies	GROUP B
58	9782020224113	Love in the New Milenn...	GROUP B
63	9781784182819	The Five Islands	GROUP B
66	9781788070609	Phantomstein in Baghdad	GROUP B
70	9781815761840	The Dinner Guest	GROUP B
74	9782025272007	The White Book	GROUP B
78	9781913884582	Priglas	GROUP B
13	9781842861181	The Gospel According to...	GROUP C
7	9781529427620	Joni Hendrix Live in Livi	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 left sidebar displays the 'SCHEMAS' tree with 'library' selected. The main editor contains a SQL query: `use library; -- 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 -- 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 'Result Grid' at the bottom shows 53 rows of data for GROUP A, with columns: book_id, isbn, title, and group_name. The data is grouped by group_name, with GROUP A having 53 rows.

book_id	isbn	title	group_name
1	9780488798897	Books	GROUP A
15	9781744523265	Tene Malor	GROUP A
16	9781844278784	When We Were Drawing	GROUP A
118	9781758252597	Acute Building	GROUP A
117	9781811284085	Happy Street, Italy	GROUP A
8	9781913267148	Phonology	GROUP A
25	9781912202583	The Books of Jacob	GROUP A
26	9781911026782	Whorehouse	GROUP A
113	9781913267530	In Memory of Memory	GROUP A
37	9781919480880	The Employees	GROUP A
41	9782571348364	The Document of Evening	GROUP A
46	9781913884513	The Silver House: Book	GROUP A
49	9781788077929	Like Eyes	GROUP A
13	9781911677225	A Lion	GROUP A
17	9781914827887	John for the Queen	GROUP A
181	9780857064729	The Faculty of Dreams	GROUP A
18	9781815885184	The Years	GROUP A
58	9781910261686	The 7th Function of Lan...	GROUP A
19	9781825488854	The Seven Moons	GROUP A
77	9781844792784	On My Love	GROUP A

Local Instance 3308 - Warning - not supported

Administration Schemas Books book_groups

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

```

100% 714

Result Grid Filter Rows: 0 Search Export

book_id	isbn	title	group_name
3	9781838071338	Whisk	GROUP B
8	9781838764718	In Mother David	GROUP B
10	9781762279627	Pops	GROUP B
15	9781913287678	Parasite	GROUP B
16	9781969368432	China Kitchens	GROUP B
20	9781313367715	A New Normal: Surviving 9/11	GROUP B
25	9781914277182	Curved Runny	GROUP B
30	9781328407193	An Ordinary at Losses	GROUP B
34	9781913471713	Water Land	GROUP B
35	9781308795031	The Run Fast	GROUP B
42	9781911277885	The Right Life	GROUP B
46	9781524036037	Tyrl	GROUP B
50	9781767320483	Mac and His Problem	GROUP B
54	9781917348186	Cosideration	GROUP B
58	9781630224213	Love in the New Millennium	GROUP B
60	9781184140819	The Pine Islands	GROUP B
65	9781766795669	Persepolis in Baghdad	GROUP B
70	9781911767080	The Center Guard	GROUP B
74	9781625575987	The White Book	GROUP B
76	9781313865450	Agnes	GROUP B

Result 01

Read Only

Local Instance 3308 - Warning - not supported

Administration Schemas Books book_groups

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

```

100% 14

Result Grid Filter Rows: 0 Search Export

book_id	isbn	title	group_name
3	9781643881181	The Gospel According to the New World	GROUP C
7	9781028427820	Jane Harolds Love in Life	GROUP C
11	9781913087660	Old Born	GROUP C
13	9781028482517	Harriet	GROUP C
18	9781962718518	The Book of Mother	GROUP C
23	9781969940859	After the Sea	GROUP C
27	9781967006336	The War of the Pines	GROUP C
31	9781782575682	At Night All Blood is Black	GROUP C
36	9781912887139	Summer Brother	GROUP C
38	9781962059551	The Perfect King	GROUP C
43	9781624434651	The Enlightenment of the Greenage	GROUP C
47	9781624510887	Peace on the Top of My Tongue	GROUP C
51	9781782576222	Heart King	GROUP C
56	9781913983715	Drive Your Pine Over the Bones of the	GROUP C
59	9781718274749	Death of a King	GROUP C
63	9781911963839	The Penetration	GROUP C
67	9781048276200	Go, West, Go	GROUP C
71	9781911442417	The Flying Mirror	GROUP C
74	9781786118014	The World Goes On	GROUP C

Result 02

Read Only

Local instance 2026 - 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 (93 rows):

book_id	isbn	title	group_name
1	0787028414021	Standing Heavy	GROUPD
11	0787028414021	The Stranger Party	GROUPD
12	0787028414021	A System to Magnificent 5 is Missing	GROUPD
13	0787028414021	Love in the Big City	GROUPD
14	0787028414021	How I Love My Life	GROUPD
24	0787028414021	Bank of Sand	GROUPD
25	0787028414021	When We Learn to Understand the World	GROUPD
26	0787028414021	I Live in the World	GROUPD
27	0787028414021	The Dangers of Smoking in Bed	GROUPD
28	0787028414021	The Assassination of John F. Kennedy	GROUPD
44	0787028414021	The Memory Palace	GROUPD
45	0787028414021	Hurricane Season	GROUPD
46	0787028414021	Reveries	GROUPD
47	0787028414021	Four Seasons	GROUPD
48	0787028414021	The Book of Moral Tales	GROUPD
49	0787028414021	The Shape of the Future	GROUPD
72	0787028414021	Like a Falling Shadow	GROUPD
73	0787028414021	The Invention	GROUPD
74	0787028414021	Vernon Sublime 1	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 2026 - Warning: not supported

Schemas: Administration, Schemas, book_groups

Query Editor:

```

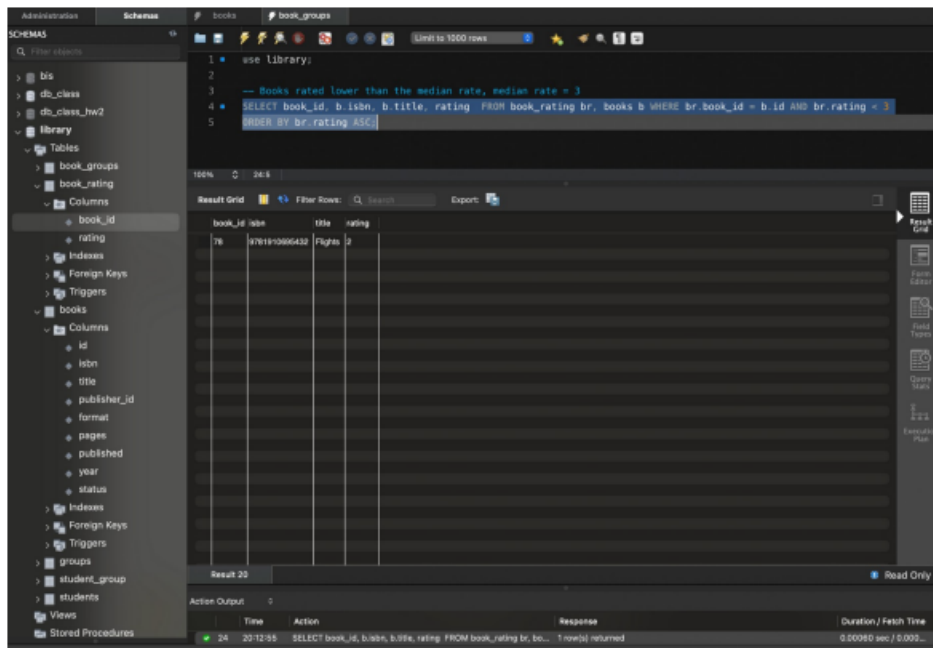
1 use library;
2
3 -- Books rated higher than the median rate, median rate = 3
4
5 SELECT book_id, b.isbn, b.title, rating FROM book_rating br, books b WHERE br.book_id = b.id AND br.rating > 3
6 ORDER BY br.rating ASC;

```

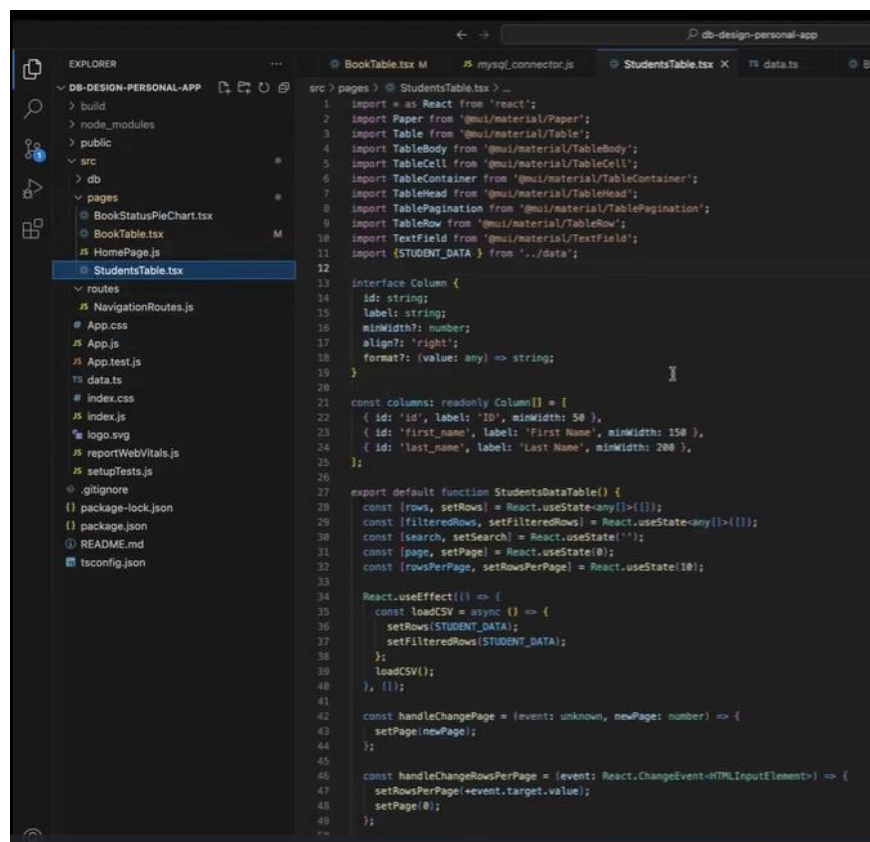
Result Grid (19 rows):

book_id	isbn	title	rating
30	0787028414021	The Perfect Nine	4
31	0787028414021	The General According to the New World	4
4	0787028414021	Is Mother Dead	4
7	0787028414021	Jimi Hendrix Live in Life	4
8	0787028414021	The Whoddy Party	4
9	0787028414021	While My Wives Unwearing	4
10	0787028414021	Pyre	4
76	0787028414021	Vernon Sublime 1	4
74	0787028414021	The White Book	4
14	0787028414021	Paradise	4
18	0787028414021	Elena Knows	4
19	0787028414021	The Book of Mother	4
20	0787028414021	How I Love My Life	4
73	0787028414021	The Spoken Bicycle	4
24	0787028414021	Womb of Sand	4
72	0787028414021	The Imposter	4
71	0787028414021	The Flying Mountain	4
30	0787028414021	An Inventory of Losses	4
31	0787028414021	All Nigger All Blood in Black	4
32	0787028414021	I Live in the World	4
33	0787028414021	In Memory of Memory	4
34	0787028414021	Minor Detail	4
35	0787028414021	Summer Brother	4
37	0787028414021	The Encyclopedia	4
38	0787028414021	The Pear Field	4
77	0787028414021	Die, My Love	4
41	0787028414021	The Devolution of Evening	4
42	0787028414021	The Eighth Life	4
44	0787028414021	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;



- 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                       <TableCell key={column.id} align={column.align}>
100                         {column.format ? column.format(value) : value}
101                       </TableCell>
102                     ))}
103                   </TableRow>
104                 ))}
105             </TableBody>
106           </Table>
107         </Paper>
108       </div>
109     );
110   }
111 }

```

- Book data Model:

```

src > ts data.ts > @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": "9781528414431",
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": "9781474622825",
49    "title": "Time Shelter",
50    "publisher_id": 38,
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/ORA-vitals
18 reportWebVitals();
  
```

- HashRoute: HashRoute was imported for connectivity with GitHub

```

src > routes > .js NavigationRoutes.js > RootNavigation
1  import React from 'react';
2  import {Route, Routes} from "react-router-dom";
3  import HomePage from '../pages/HomePage';
4  import StickyHeadTable from '../pages/BookTable';
5  import StudentsDataTable from '../pages/StudentsTable';
6  import BookStatusPieChart from '../pages/BookStatusPieChart';
7  import { HashRouter } from 'react-router-dom';
8
9
10
11 function RootNavigation() {
12   return (
13     <HashRouter>
14     <Routes>
15     <Route path="/" element={HomePage}/>
16     <Route path="/book-data" element={StickyHeadTable}/>
17     <Route path="/student-data" element={StudentsDataTable}/>
18     <Route path="/book-status-analysis" element={BookStatusPieChart}/>
19     </Routes>
20   </HashRouter>
21 );
22 }
23
24 export default RootNavigation;
25

```

5.2. Front-end Demonstration

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	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

← → ↻ https://ebogni17.github.io/LibraryManagement/#/book-data ☆

Getting Started DreamBox Learning M... chrome - Bing Presentation Session S... Amazon.com - Online... Login | CompTIA LastPass Amazon.com - Online... Booking.com Exp >> Other Bookmarks

ook Data

Search by title

ti

ID	ISBN	Title	Publisher ID	Format	Pages	Published Date	Year	Status
5	9781474623025	Time Shelter	30	hardcover	304	2022-04-21 00:00:00	2023	Available
11	9781913097660	Still Born	5	paperback	200	2022-06-22 00:00:00	2023	Available
47	9781908670557	Faces on the Tip of My Tongue	14	paperback	136	2019-10-22 00:00:00	2020	Available
54	9781912240166	Celestial Bodies	19	paperback	243	2018-05-21 00:00:00	2019	Available
69	9781910701584	The 7th Function of Language	7	hardcover	390	2017-05-03 00:00:00	2018	Available

Rows per page: 10 ▾ 1–5 of 5 < >

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

← → ↻ https://ebogni17.github.io/LibraryManagement/ ☆

Getting Started DreamBox Learning M... chrome - Bing Presentation Session S... Amazon.com - Online... Login | CompTIA LastPass Amazon.com - Online... Booking.com Exp >> Other Bookmark

Welcome to Student Library

Edith Jiokeng

jiokengbognie0517@students.bowiestate.edu

[Github](#)

[Project Report](#)

[View book data](#)

[View student data](#)

[View book analysis](#)

Webpage displaying Student Data

← → ↻ https://ebogni17.github.io/LibraryManagement/#/student-data ☆

Getting Started DreamBox Learning M... chrome - Bing Presentation Session S... Amazon.com - Online... Login | CompTIA LastPass Amazon.com - Online... Booking.com Exp >> Other Boo

Student Data

Search by first/last name

ID	First Name	Last Name
2	PRECIOUS	ADELA KUN
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 “jio” for jioke ng all the name that have “jio” are displayed or “ja” for Jared, all the name with “ja” are displayed.

← → ↻ https://ebogni17.github.io/LibraryManagement/#/student-data ☆

Getting Started DreamBox Learning M... chrome - Bing Presentation Session S... Amazon.com - Online... Login | CompTIA LastPass Amazon.com - Online... Booking.com Exp >> Other Bookmarks

Student Data

Search by first/last name

ed

ID	First Name	Last Name
8	VARUNKUMARREDDY	DODDA
14	EDITH	JIOKENGBOGNI
16	EDDY	KOUNDJOUYEM
18	XAVIER	MEDY
20	ADEDEJI	OLUDAYO
24	JARED	ROBINSON

Rows per page: 10 ▾ 1-6 of 6 < >

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

← → ↻ https://ebogni17.github.io/LibraryManagement/ ☆

Getting Started DreamBox Learning M... chrome - Bing Presentation Session S... Amazon.com - Online... Login | CompTIA LastPass Amazon.com - Online... Booking.com Exp >> Other Boc

Welcome to Student Library

Edith Jiocheng

jiokengbognie0517@students.bowiestate.edu

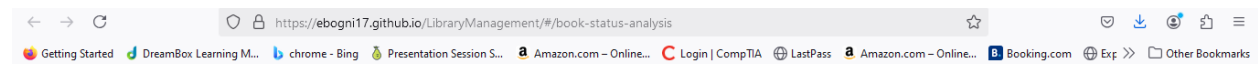
[Github](#)

[Project Report](#)

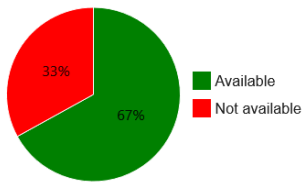
[View book data](#)

[View student data](#)

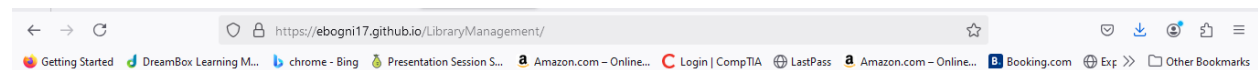
[View book analysis](#)



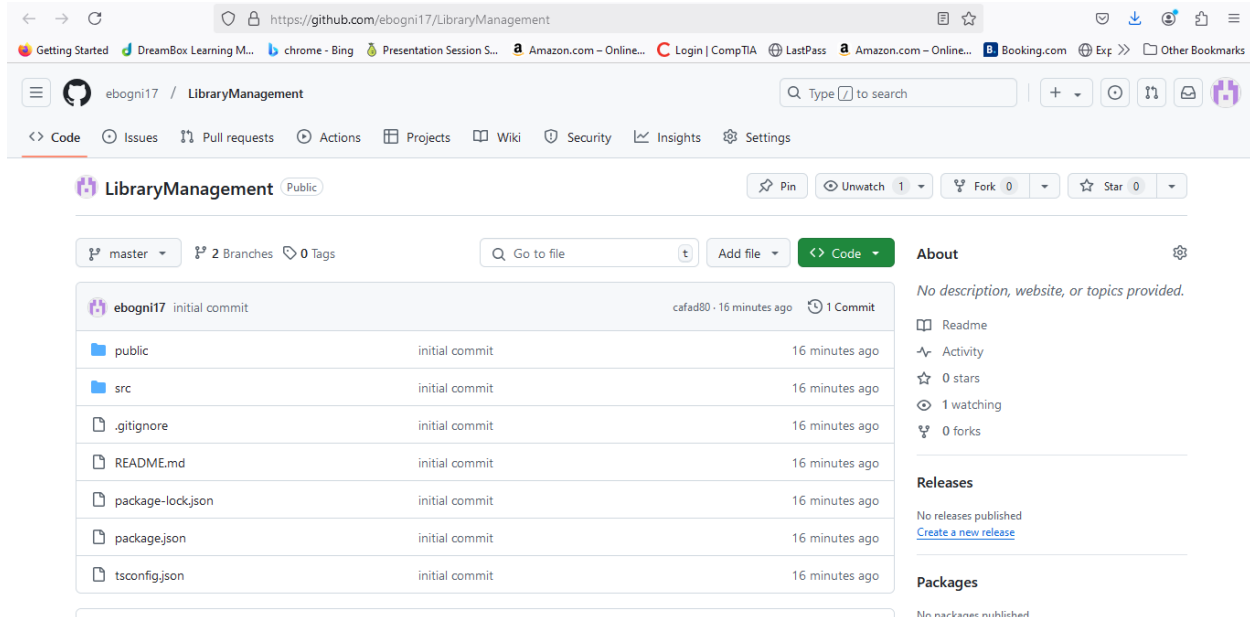
Pie chart showing analysis of available books



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/ebogni17/LibraryManagement>

GitHub Page Link: <https://ebogni17.github.io/LibraryManagement/>