|  |  |
| --- | --- |
|  | **2012** |
|  |  |
|  |  |
|  |  |

**A Project Report on**

|  |  |
| --- | --- |
|  | **PROJECT WORK**  **B.Sc. CS**  **Semester-VI** |
|  |  |

**LIBRARY MANAGEMENT SYSTEM**

Submitted as partial fulfillment of the requirement

for B.Sc**. (CS) - Semester VI**

Prepared for Subject: **PROJECT WORK** (CSCC24)

**Prepared By**

**Name of Student : Rathod Mitesh**

**Enrollment No : 22BSCCS22052**

****

Shri Maneklal M. Patel Institute of Sciences and Research**,**

**Department of B.Sc. (CS), Gandhinagar.**

Shri Maneklal M. Patel Institute of Sciences and Research**,**





**B.Sc.(CS)Gandhinagar.**

Date: 17 /03 /2025

**CERTIFICATE**

This is to certify that the project report compiled by Rathod Mitesh. Student of **Shri Maneklal M. Patel Institute of Sciences and Research, B.Sc.(CS) Department,** has completed his Semester- VI project satisfactorily within the department under the guidance of **Baljeet Kaur Dave.**

To the best of our knowledge this is an original work by him/her. He has worked on project **“Library Management System”** for the duration from 18/11/2024 to 29/03/2025.

During his tenure at this department, he/she was found to be sincere and meticulous in his work. We appreciate his enthusiasm & dedication towards the work assigned to him/her. We are hopeful that he/she will prove to be good professional & wish him/her grand success for the future.

**Internal Project Guide**

Mr. /Ms. **Baljeet Kaur Dave**

Department of Computer Sciences

**Prof. (Dr)** **B. R. Pandya**

Head of Department,

Department of Computer Sciences



**ACKNOWLEDGMENT**

I am grateful to my internal guide ***Baljeet Kaur Dave*** for providing me valuable guidance for the PROJECT WORK and other domain related information.

(I would like to express my endless thanks to ***Baljeet Kaur Dave***, for giving invaluable advice, direction, support and transferring knowledge throughout the project work.)

Also, my hearty gratitude to Head of Department, **Prof. (Dr.) B. R. Pandya** for giving me an opportunity for learning through the process of project work by which I can improve my technical skills along with practical exposure and its applicability in the industry.

**Name of Student: Rathod Mitesh**

Enrollment No: 22BSCCS22052

**Scope**

1. Objective

* The **Library Management System** is designed to automate library operations and manage books, users, and borrowing activities efficiently.
* It provides an **admin panel** that allows the librarian to **monitor and control** the entire system.
* The system includes features for **adding, updating, and deleting books** along with tracking borrowing requests.
* Users can **search for books, borrow books, and update their profiles** in a secure and streamlined manner.
* The project aims to **reduce manual efforts, eliminate paperwork, and enhance the efficiency** of library operations.

1. Functional Requirement

**Admin Functionalities:**

* Manage Books
  + Add a new book
  + Delete books
  + Update book details
  + View books with ISBN numbers
* Manage Borrow Requests
* Approve or Reject book borrow requests
* View pending requests
* Manage Users
* Add, update, and delete users
* Track user details
* Issue & Return Books
  + Issue books to students
  + Mark books as returned
* Fine Management
  + Apply late return fines after 15 days
  + Maintain defaulter list
* Authentication & Authorization
  + Secure admin login

**User Functionalities:**

* User Registration & Login
  + New user registration
  + Secure user login
* Profile Management
  + Update profile details
* Book Search & Borrowing
  + Search books by title, author, or category
  + Request books for borrowing
* Issued Books & Due Date Tracking
  + View issued books
  + Check due dates
* Fine Payment
  + View pending fines
  + Pay fines
* Cancel Borrow Requests
  + Users can cancel their borrow request if not yet approved

**Library Management System Functionalities**

**Admin Functionalities:**

1. Record Library Activities

* This feature allows the **admin** to track all activities like book borrow requests, returns, and fines.
* The system logs all transactions, ensuring a **clear history of operations.**

2. Manage Books

* Add Book
  + The **admin can add new books** by entering details such as **title, author, category, ISBN, quantity, etc.**
  + Each book is uniquely identified by an **ISBN number.**
* Delete Book
  + The **admin can delete books** if they are outdated or unavailable.
* Update Book
  + The **admin can update book details**, including title, author, and stock availability.
* View Books
  + The **admin can view all books** in the library with filters for searching by **ISBN, author, or category.**

3. Manage Users

* The **admin can add, update, and delete user accounts** for students and staff.
* This helps the librarian **keep track of registered users** in the system.

Update User

* The **admin and users can update their profile information**, such as **name, email, and mobile number.**

Delete User

* The **admin can remove user accounts** from the system if they are inactive or no longer part of the institution.

4. Manage Borrow Requests

* Users **request books** for borrowing, and the **admin reviews and approves/rejects the requests.**
* **Pending requests are stored** in the database until processed by the admin.

5. Issue & Return Books

* The **admin issues books** to users after approving the borrow request.
* **Return books are recorded**, and the system updates availability status accordingly.

6. Fine Management

Defaulter List

* A **defaulter list is generated** for users who **fail to return books within 15 days.**
* The **fine amount is calculated automatically** based on overdue days.

7. Reports & Analytics

View Issued Books

* The **admin can view a list of issued books** and check **due dates and return status.**

Fine Collection Report

* This report provides a **detailed summary of fines collected** from late returns.

Defaulter Report

* The admin can **generate a defaulter list** of students who have pending fines.

8. Authentication & Authorization

* The system includes a secure login for both admin and users.
* The admin has higher access permissions, while users can only access their own data.

**User Functionalities**

1. New User Registration

* **Users (students and staff) can register** by providing their details.
* After successful registration, they can **log in and access the system.**

2. User Login

* Users can **log in using their email and password.**
* The system verifies the credentials before allowing access.

3. Search Books

* Users can **search books by title, author, or category** to find the required books.

4. Borrow Books

* Users can **request to borrow books** through the system.
* **The admin approves/rejects requests** based on book availability.

5. View Issued Books

* Users can **see the list of books issued to them** along with **due dates.**

6. Fine Payment

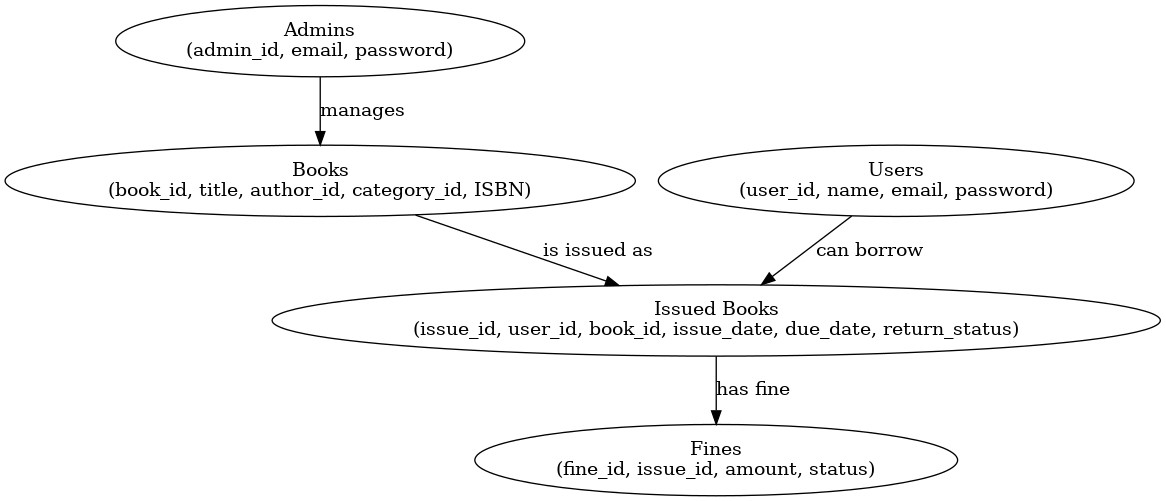
* Users who **exceed the due date of 15 days are fined.**
* The system **shows the pending fine amount.**
* Users can **pay fines manually.**

7. Cancel Borrow Requests

* If a **borrow request is still pending**, users can **cancel it before approval.**

**Future Enhancements:**

* Due Date Notifications:
  + Send **email and SMS reminders** for book return deadlines.
* Online Payment Integration:
  + Automate **fine payments** using **payment gateways like Razorpay.**

ER Diagram:

Data Dictionary

Admin table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Column name | Data type | constraints | description |
| 1. | admin\_id | Int (11) | Primary key | Unique ID for admin |
| 2. | Email | Varchar (100) | UNIQUE, NOT NULL | Email for login |
| 3. | Password | Varchar (15) | Not null | Encrypted password |

Example:

|  |  |  |
| --- | --- | --- |
| Admin\_id | Email | Password |
| 1 | meet@gmail.com | mitesh06 |

Authors table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Column name | Data type | constraints | Description |
| 1. | author\_id | Int (11) | Primary key | Unique author ID |
| 2. | name | Varchar (20) | Not null | Author's name |

Example:

|  |  |
| --- | --- |
| author\_id | name |
| 1 | J.K. Rowling |
| 2 | George orwell |
| 3 | Mark Twain |
| 4 | J.R.R. Tolkien |
| 5 | Agatha Christie |

Category table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Column name | Data type | constraints | Description |
| 1. | category\_id | Int (11) | Primary key | Id of category |
| 2. | category\_name | Varchar (100) | NOT NULL | Name of category |

Example:

|  |  |
| --- | --- |
| category\_id | category\_name |
| 1 | Dystopian |
| 2 | Fantasy |
| 3 | Math’s |
| 4 | Mystery |
| 5 | Science |

Book details:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Column name | Data type | constraints | Description |
| 1. | book\_id | Int (15) | Primary key | Unique book ID |
| 2. | Title | Varchar (25) | Not null | Book of title |
| 3. | author\_Id | Varchar (13) | Foreign key | Reference to author |
| 4. | category\_id | Varchar (15) | Foreign key | Book category ID |
| 5. | isbn | Int (10) | Unique, not null | International standard book |
| 6. | available\_copies | Int (5) | null | Number of copies available |

Example:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Book\_id | title | Isbn | Author\_id | Available\_ copies | Cat\_id |
| 1 | 1984 | 9780451524935 | Secker & Warburg | 10 | 1 |
| 2 | Harry potter & the philosopher’s stone | 978074532743 | bloomsbury | 10 |  |
| 3 | Brave New world | 9780060850524 | Harper &brothers | 10 | 3 |
| 4 | The Hobbit | 9780547928227 | Houghton Mifflin Harcourt | 10 | 4 |

Authors Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Column name | Data type | constraints | description |
| 1. | author\_id | Int (11) | Primary key | Unique author ID |
| 2. | Name | Varchar (30) | Not null | Author's name |

Example:

|  |  |
| --- | --- |
| Book author\_id | Name |
| 1 | Krish |
| 2 | mitesh |
| 3 | sujal |
| 4 | het |
| 5 | mit |

User table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Column name | Data type | constraints | description |
| 1. | User\_id | Int (11) | Primary key | Unique ID for users |
| 2. | Name | Varchar (100) | Not null | Full name of the user |
| 3. | Email | Varchar (100) | Foreign key | Email address |
| 4. | Password | Varchar (100) | Not null | Encrypted password |
| 5. | Mobile | Int (12) | Not null | Mo.no of user |

Example:

|  |  |  |  |
| --- | --- | --- | --- |
| User\_id | Name | Email | password |
| 1 | Carol Jones | Carol.jones345@gamil.com | jones\_190 |
| 2 | David White | David.white123@gmail.com | white\_899 |
| 3 | Emma Wilson | Emma.wilson867@gmail.com | wilson\_880 |

Issue book­­­­­­­­­­­­­­­­­­­­­­­­­-:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Column name | Data type | constraints | Description |
| 1. | Issue\_id | Int(11) | Primary key | Unique issue record ID |
| 2. | user\_id | Int(11) | Foreign key | User who borrowed the book |
| 3. | Book\_id | Int(11) | Foreign key | Book that was issued |
| 4. | Issue\_date | date | null | Date when book was issued |
| 5. | Return date | Date | null | Date of return |
| 6. | fine | Int(6) | null | Fine of book |
| 7. | Return status | Enum(‘issued,returned’) | DEFAULT 'Not Returned' | Status ('Returned'/'Not Returned') |
| 8. | Due Date | date | Not null | |  | | --- | |  |  |  | | --- | | Date when book is due | |

Example:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Issue\_header | user\_id | Book\_id | Issue\_date | Return\_date | Fine |
| 1 | 1 | 1 | 2024-09-01 | 2024-09-16 | 10 |
| 2 | 2 | 2 | 2024-09-05 | 2024-09-20 | 20 |
| 3 | 3 | 3 | 2024-08-17 | 2024-08-29 | 30 |
| 4 | 4 | 4 | 2024-09-20 | 2024-09-27 | 40 |
| 5 | 5 | 5 | 2024-10-10 | 2024-10-20 | 50 |

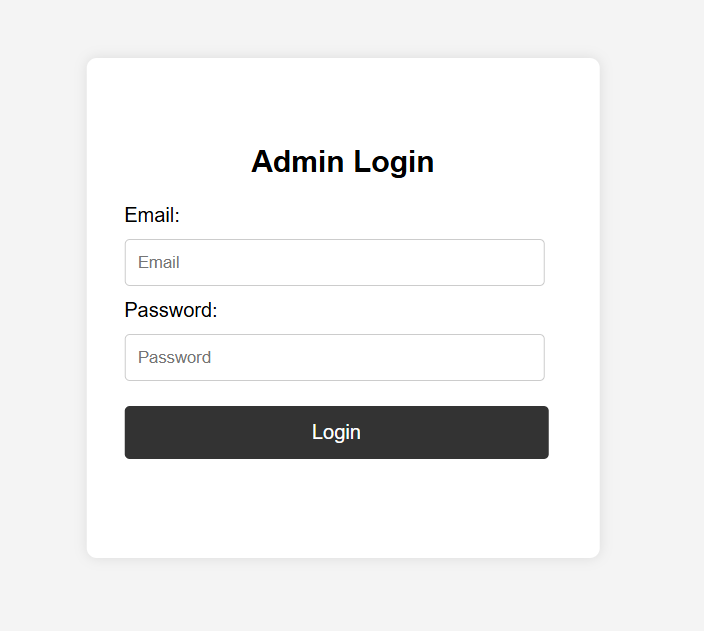
Fine:-

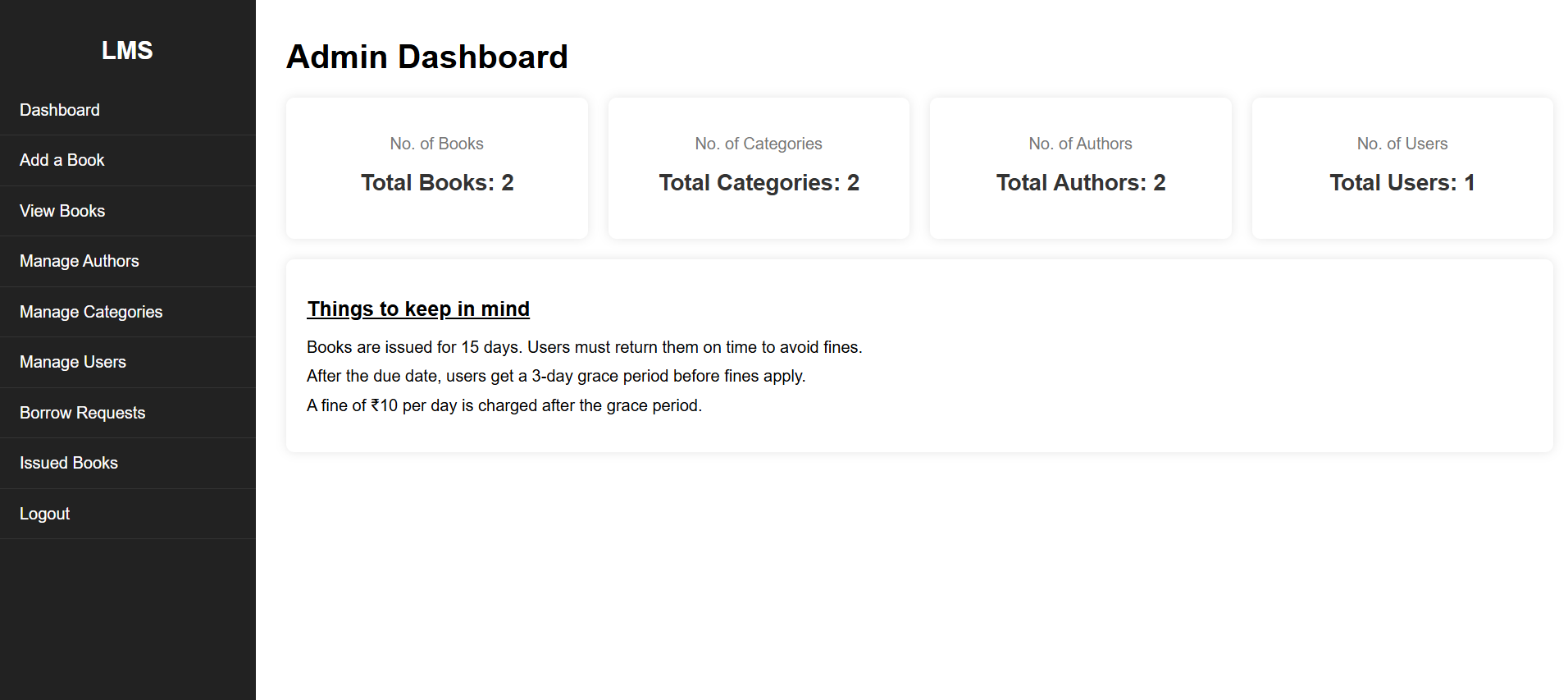
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Column name | Data type | constraints | description |
| 1. | Fine\_id | Int(11) **t(11)** | Primary key | Unique fine ID |
| 2. | User\_id | Int(11) | Foreign key | User who borrowed the book |
| 3. | Issue\_id | Int(11) | Foreign key | Issued book transaction |
| 4. | Amount | Int(11) | null | Amount of fine |
| 5. | Status | Enum(‘unpaid’,’paid’) | Null | Fine payment status ('Paid'/'Unpaid') |
| 6. | Payment\_date | timestamp | Null | Date of payment |

**Example:-**

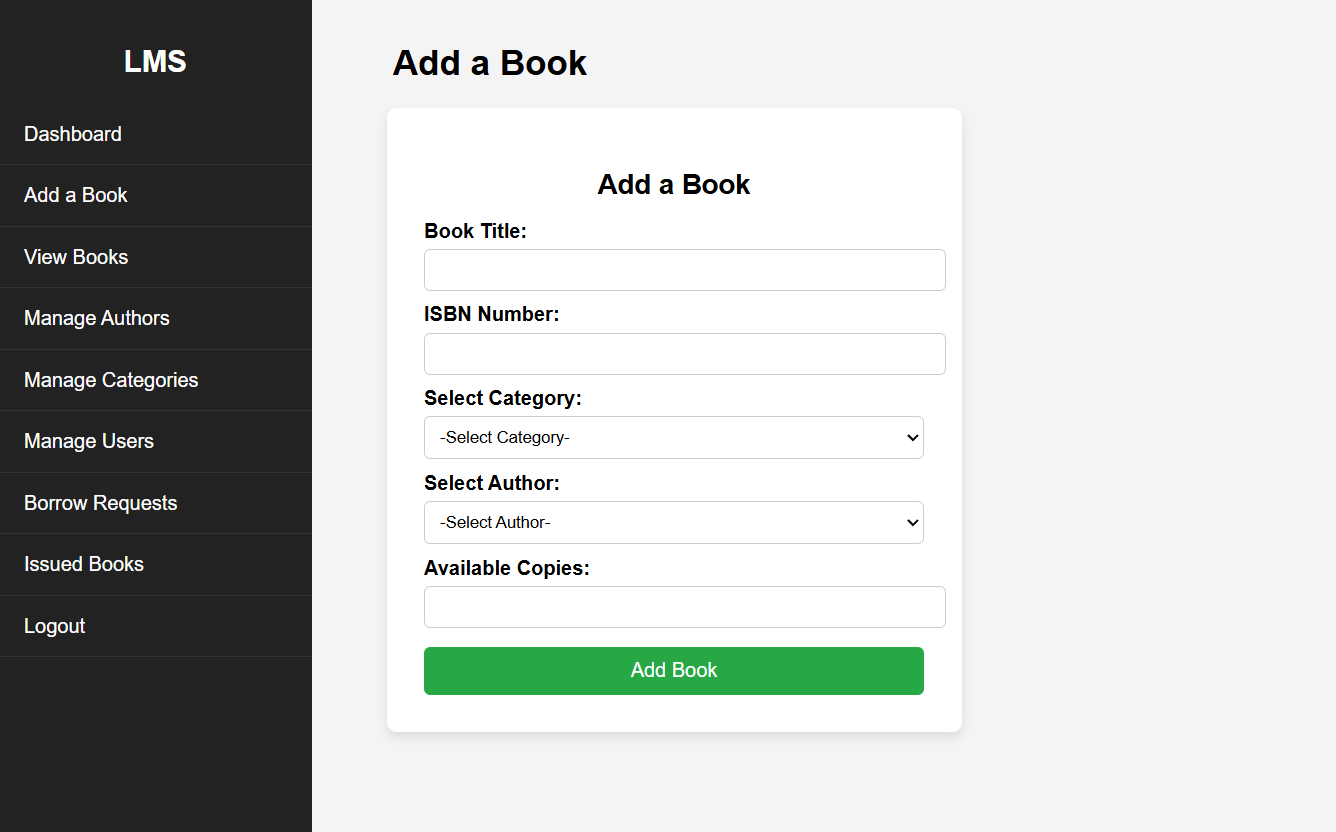
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fine\_id | User\_id | Issue\_id | Amount | Status | Payment\_date |
| 1 | 1 | 1 | 60 | Paid | 20/2/2025 |
| 2 | 2 | 2 | 20 | Unpaid | 10/3/2025 |
| 3 | 3 | 3 | 50 | Paid | 15/3/2025 |
| 4 | 4 | 4 | 10 | Unpaid | 17/3/2025 |

**Screen Layouts**

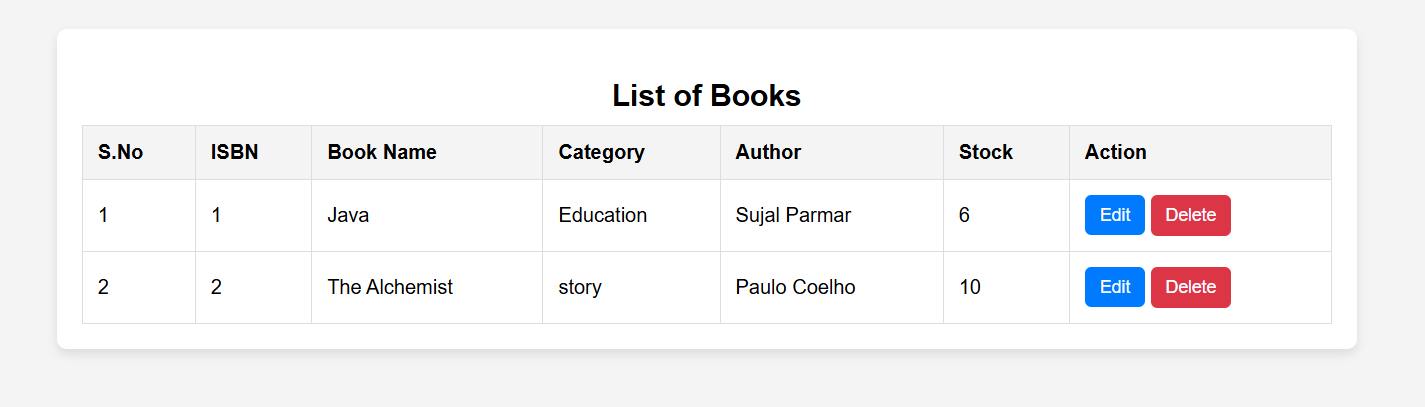
****



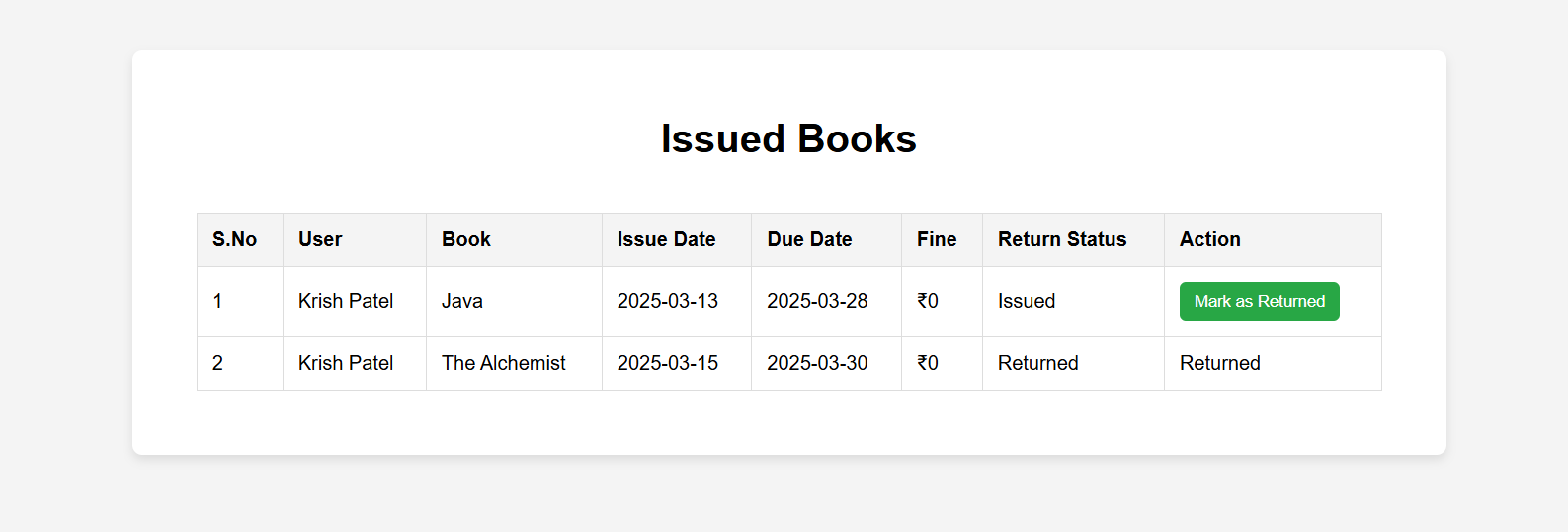
* **Description:** **This feature allows librarians to enter various records into the system, such as, returns, and non-availability of books.**

****

* **Description:** **To add book the admin has to enter details about the book through the form given and add ISBN number.**

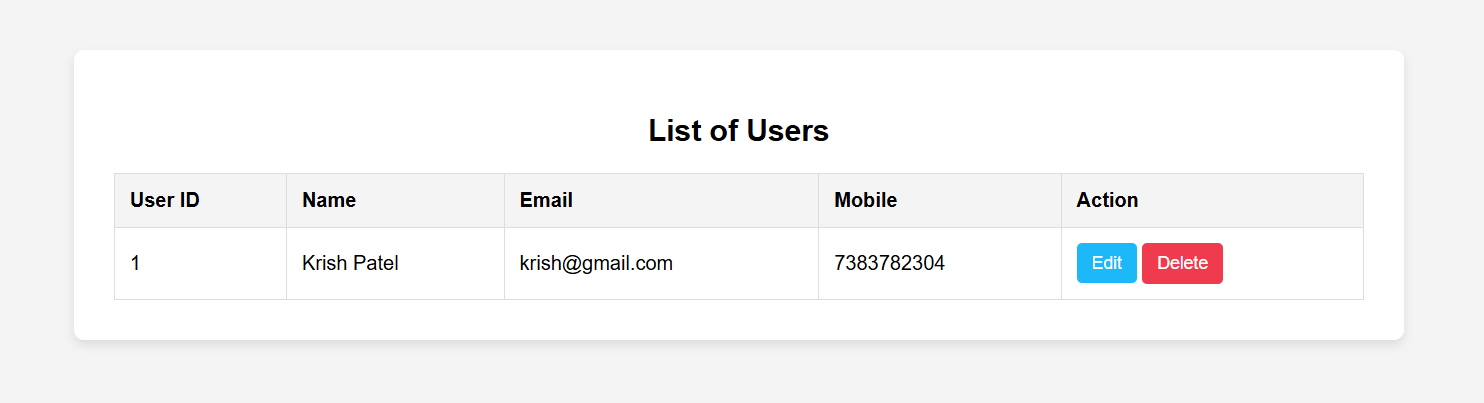
****

* **Description:** **It show how many books are available in library.**

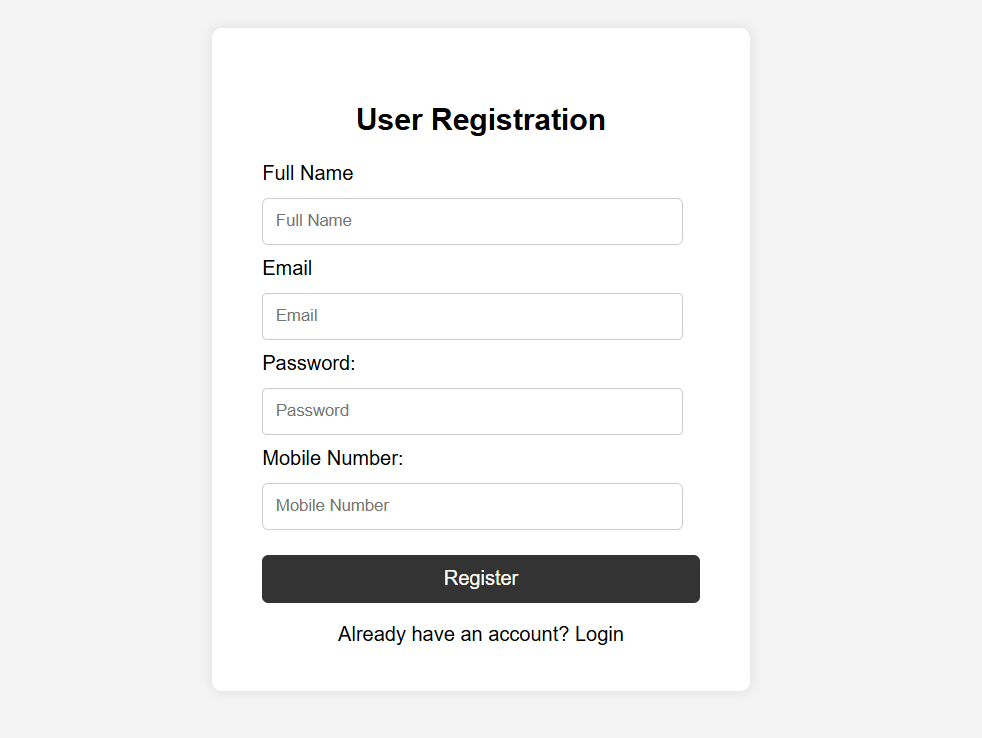
****

**Description:** **This feature allows librarians to show the details of the student who did not return the books before the deadline.**

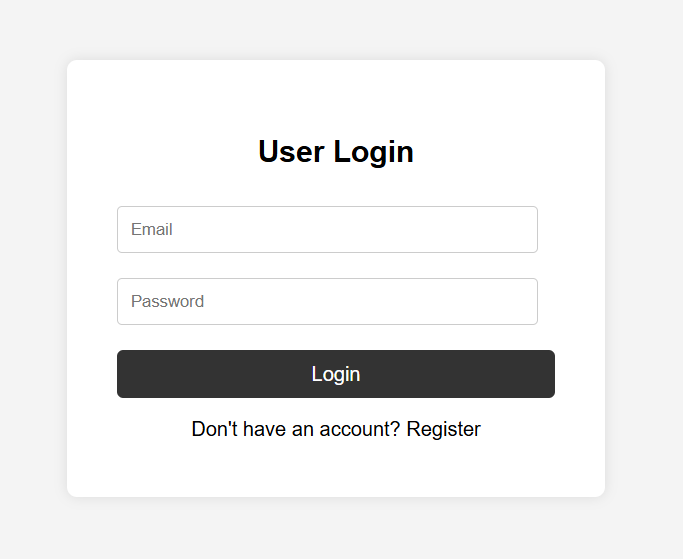
* **This feature enables librarians to view a list of students who have failed to return their books within the 15-day deadline, displaying detailed information for each defaulter.**

****

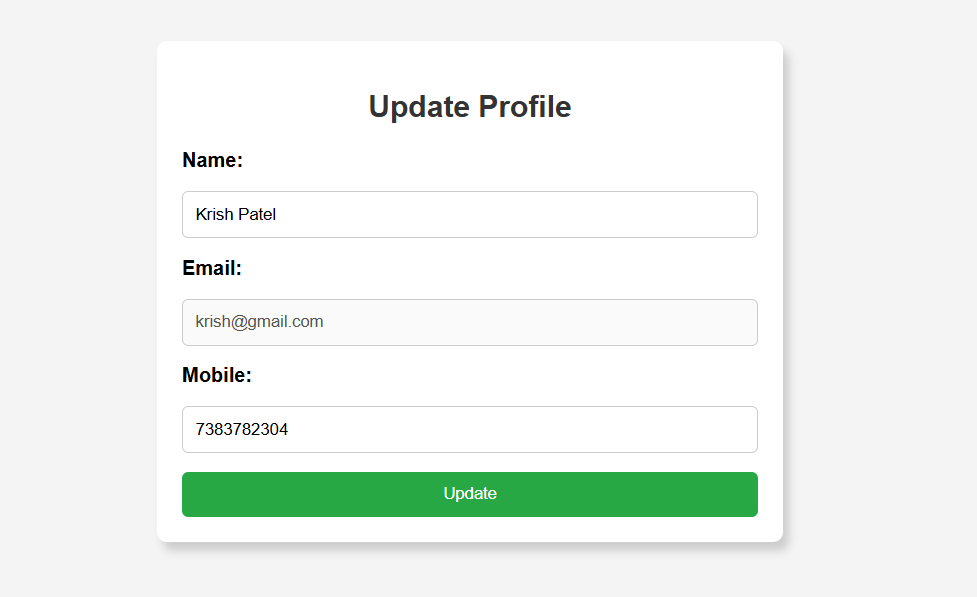
**Description:- Me admin can see how many users in library.**



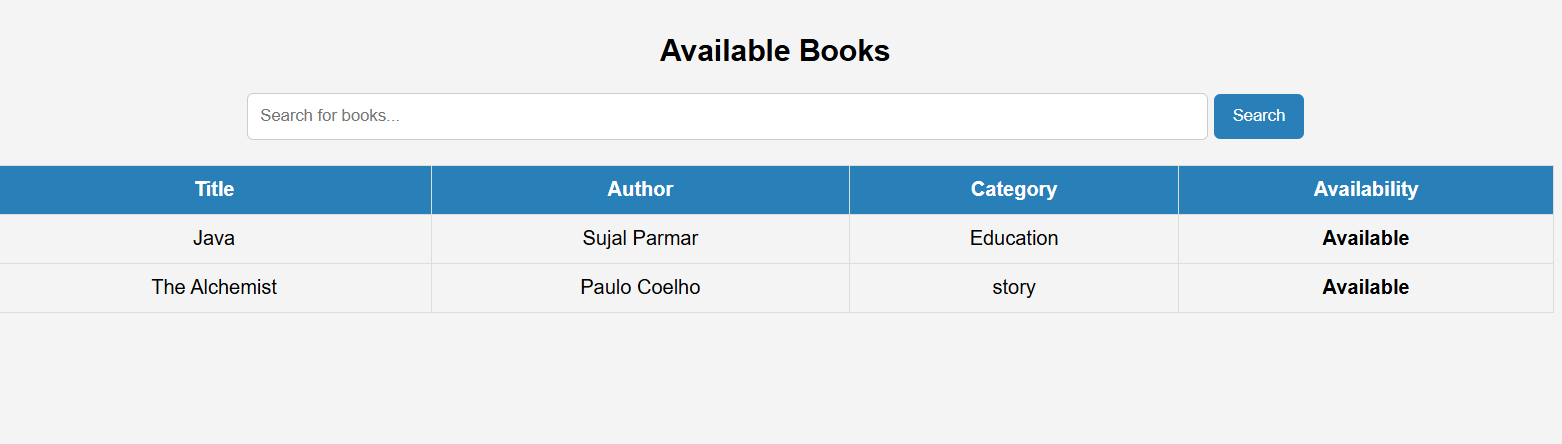
* **Description:** **This feature allows new users (students, teachers, etc.) to sign up for the system by providing the necessary details.**

****

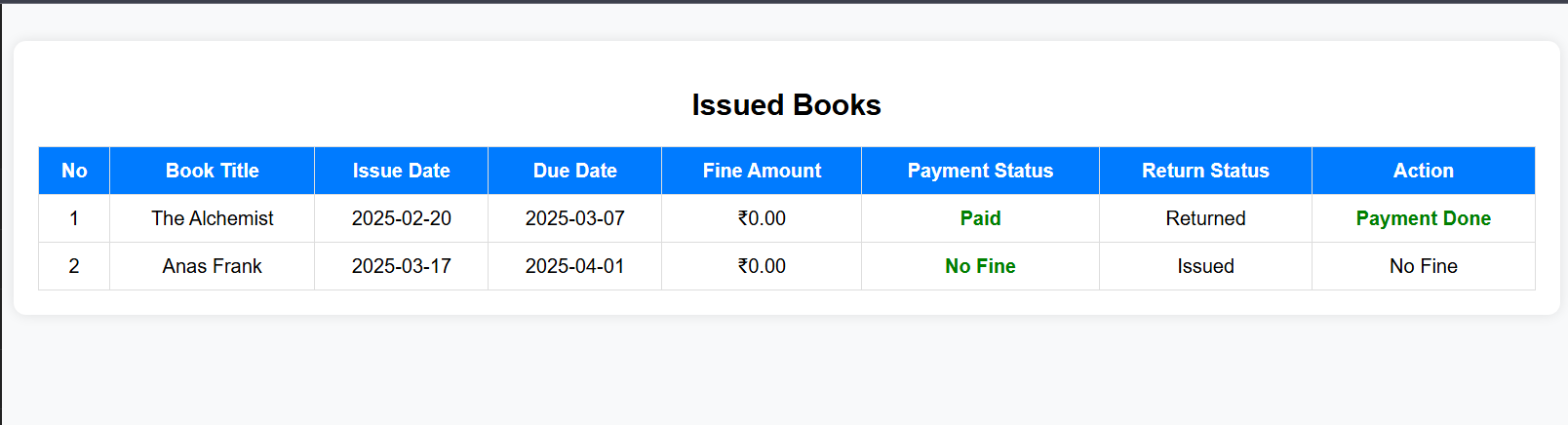
**Description:** **This feature provides authenticated access for registered users to use the system with the help of email.**

****

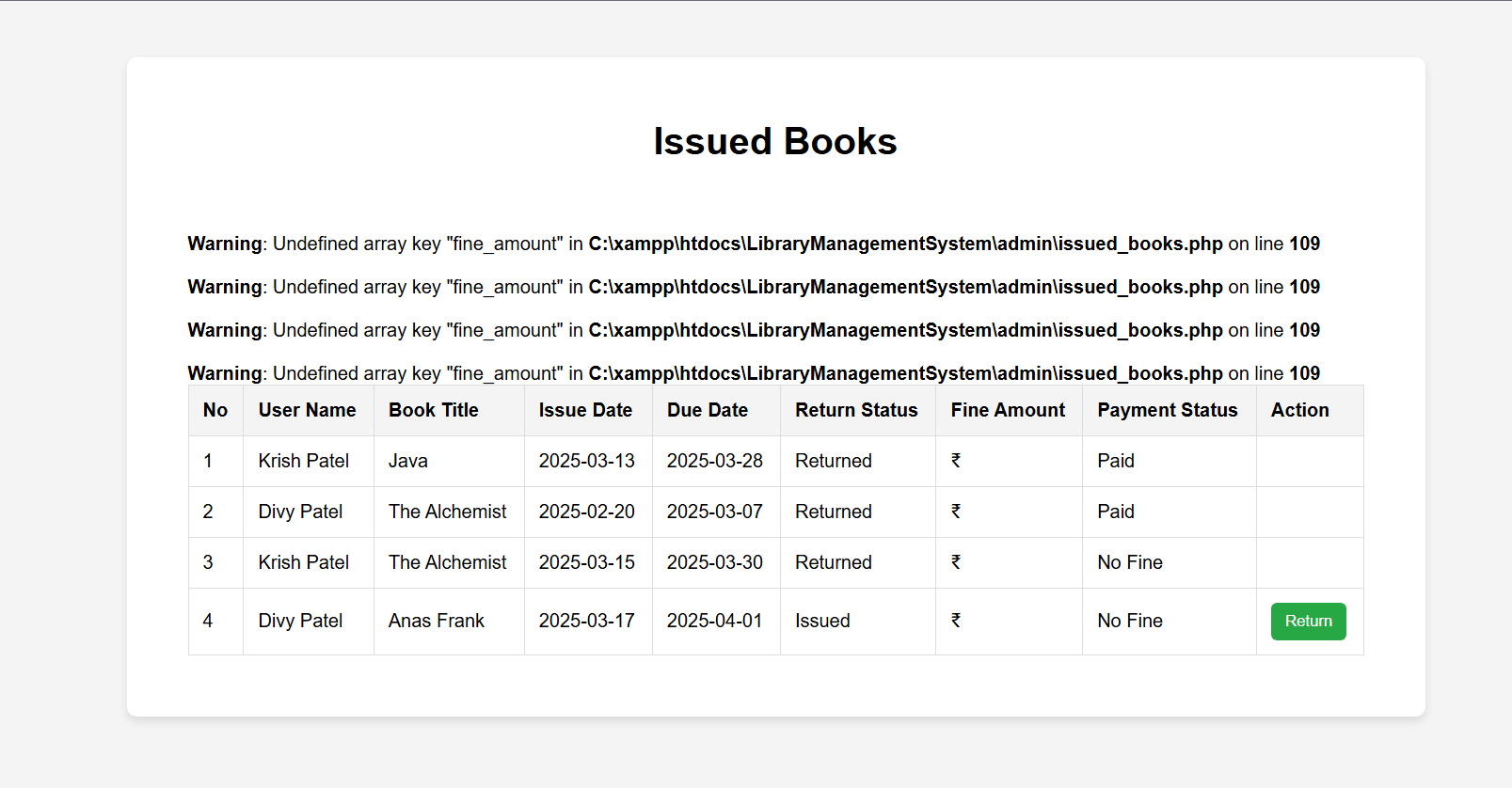
* **Description:** **To update student details the admin insert the student id and update the details needed.**

****

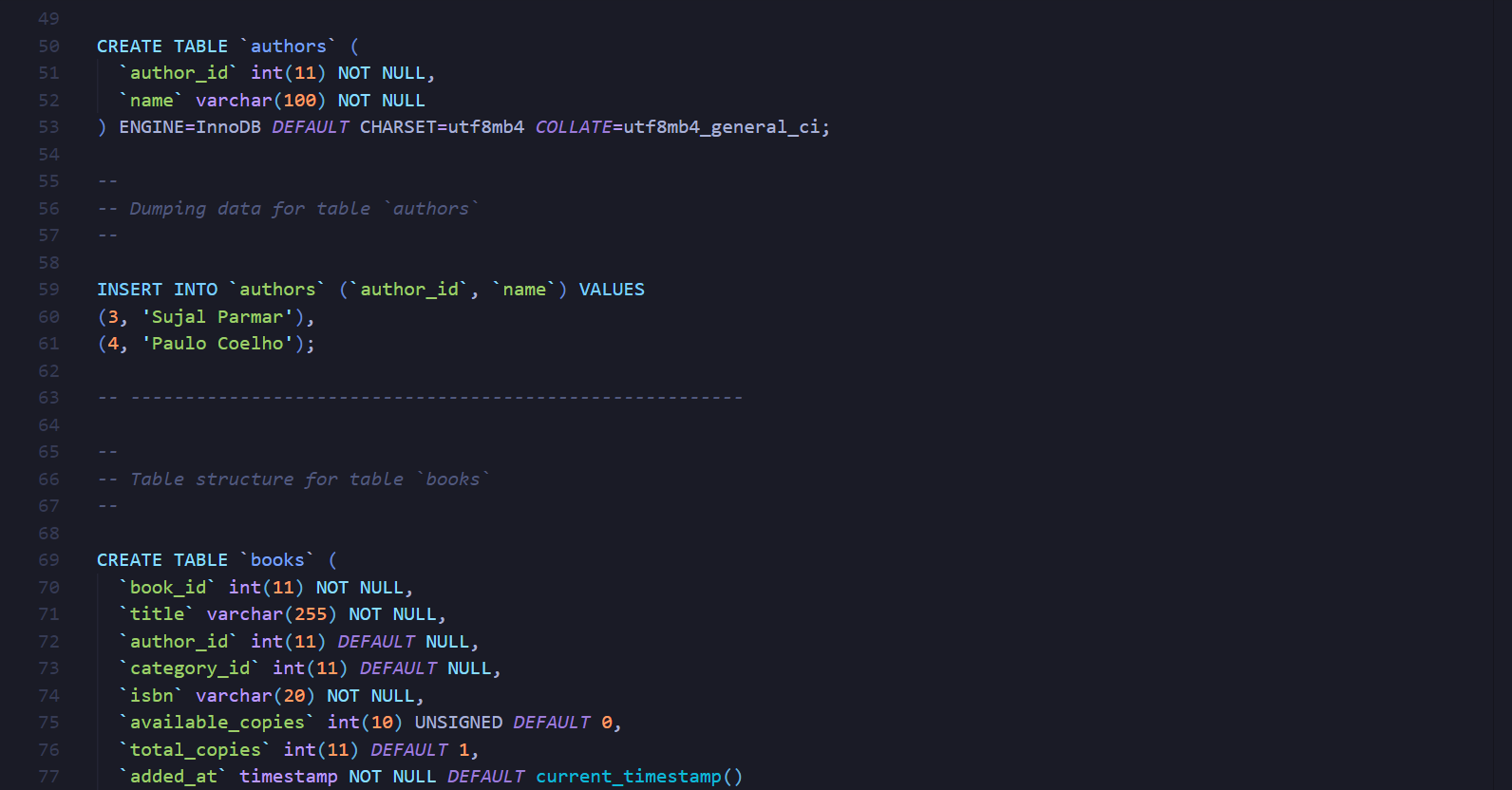
**Description:- Users can see how many books are available and unavailable**

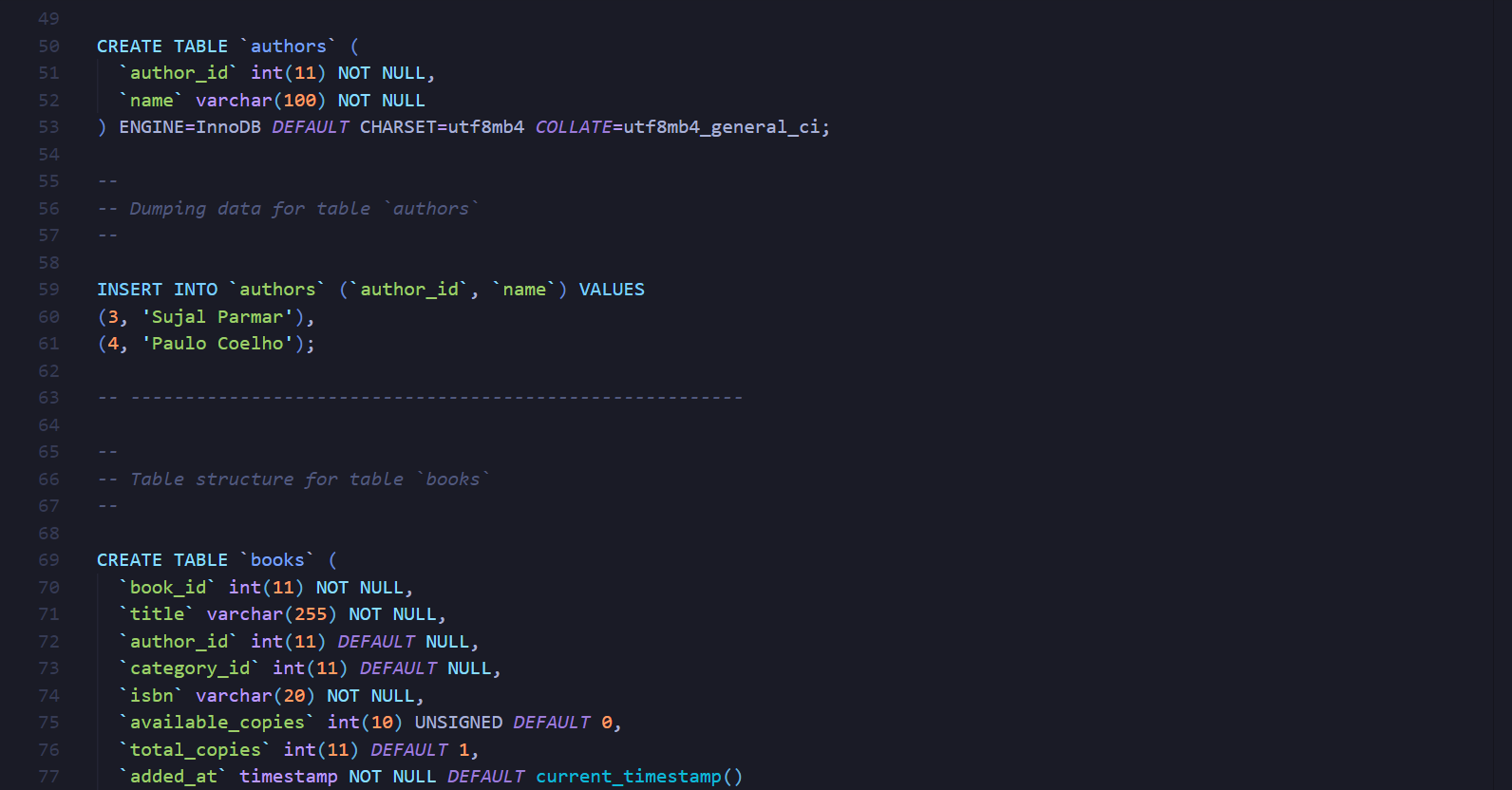
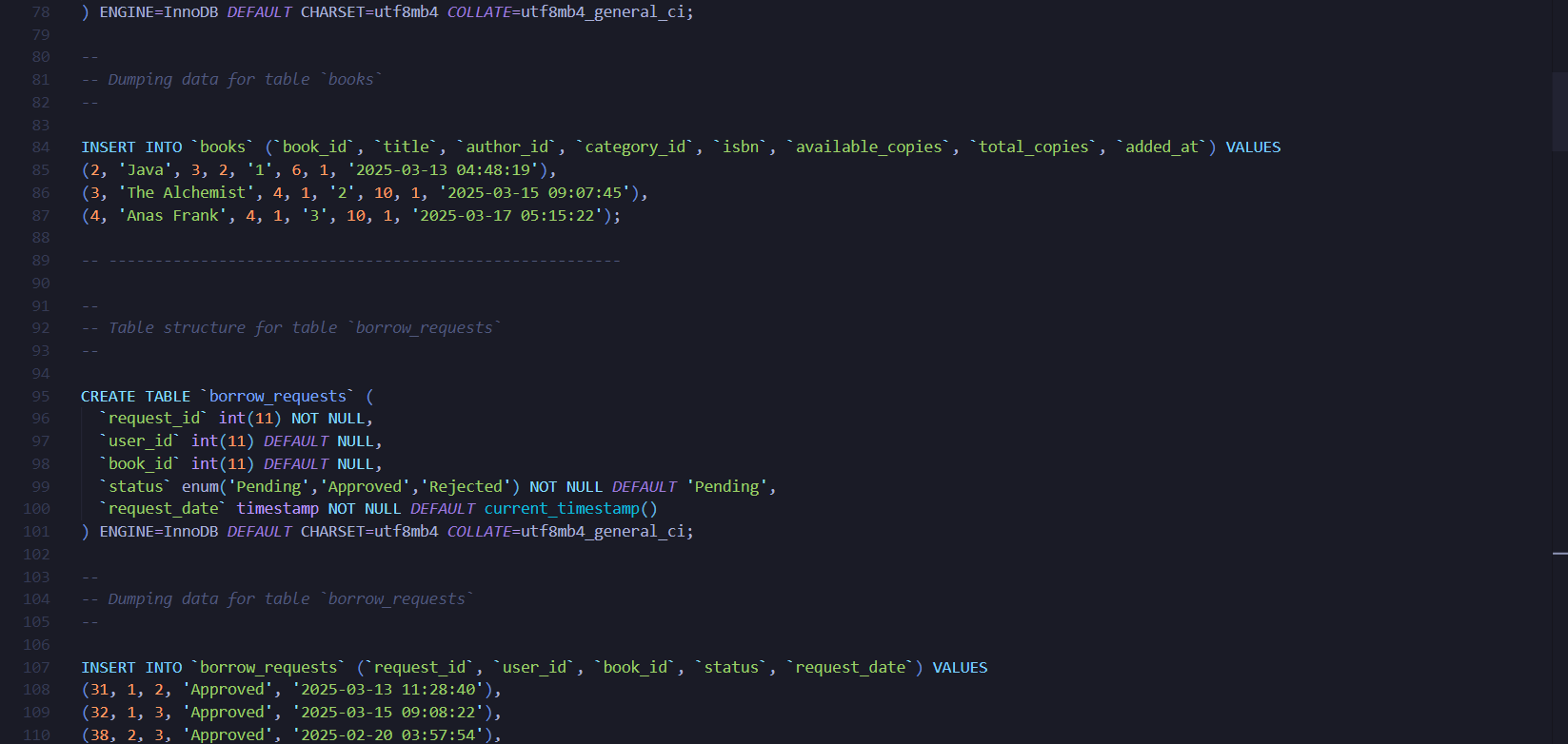
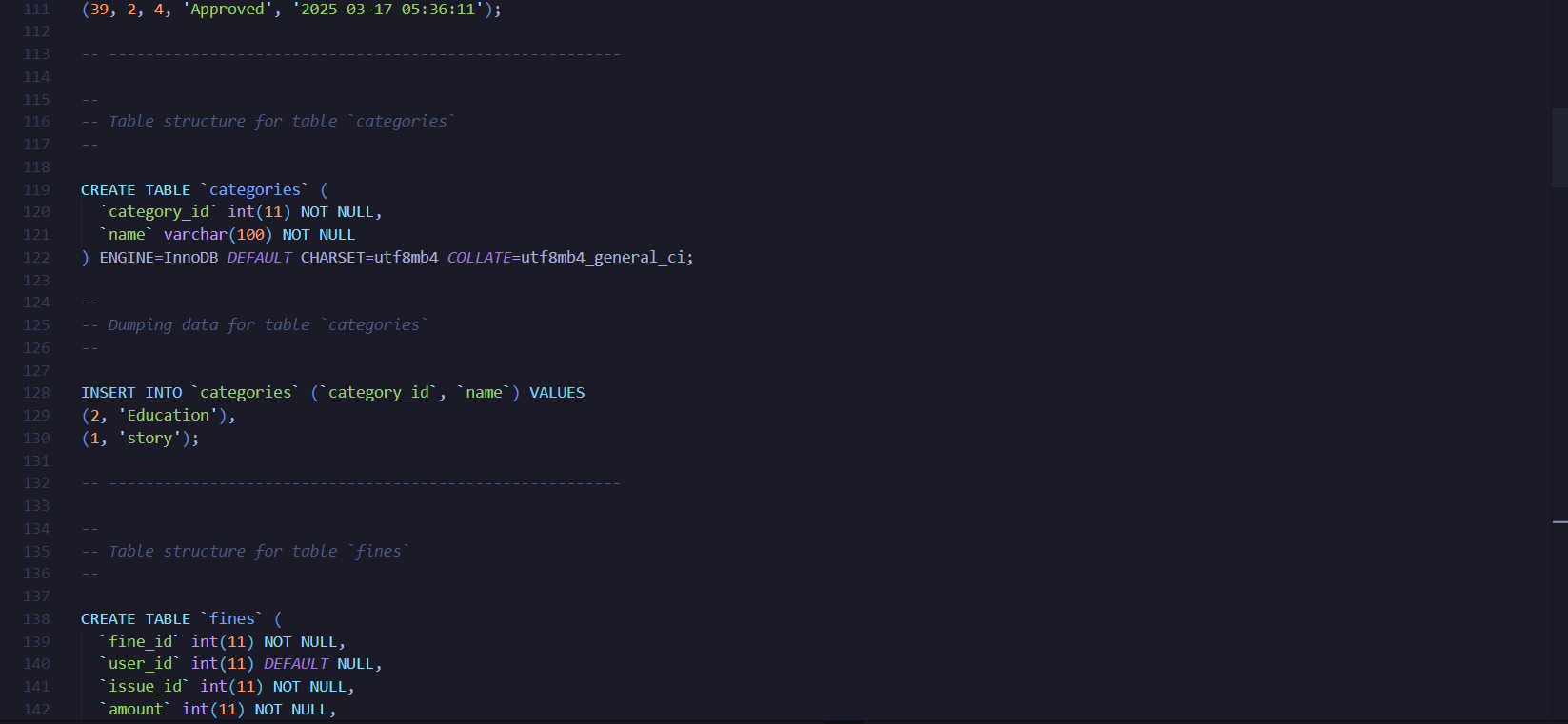
****

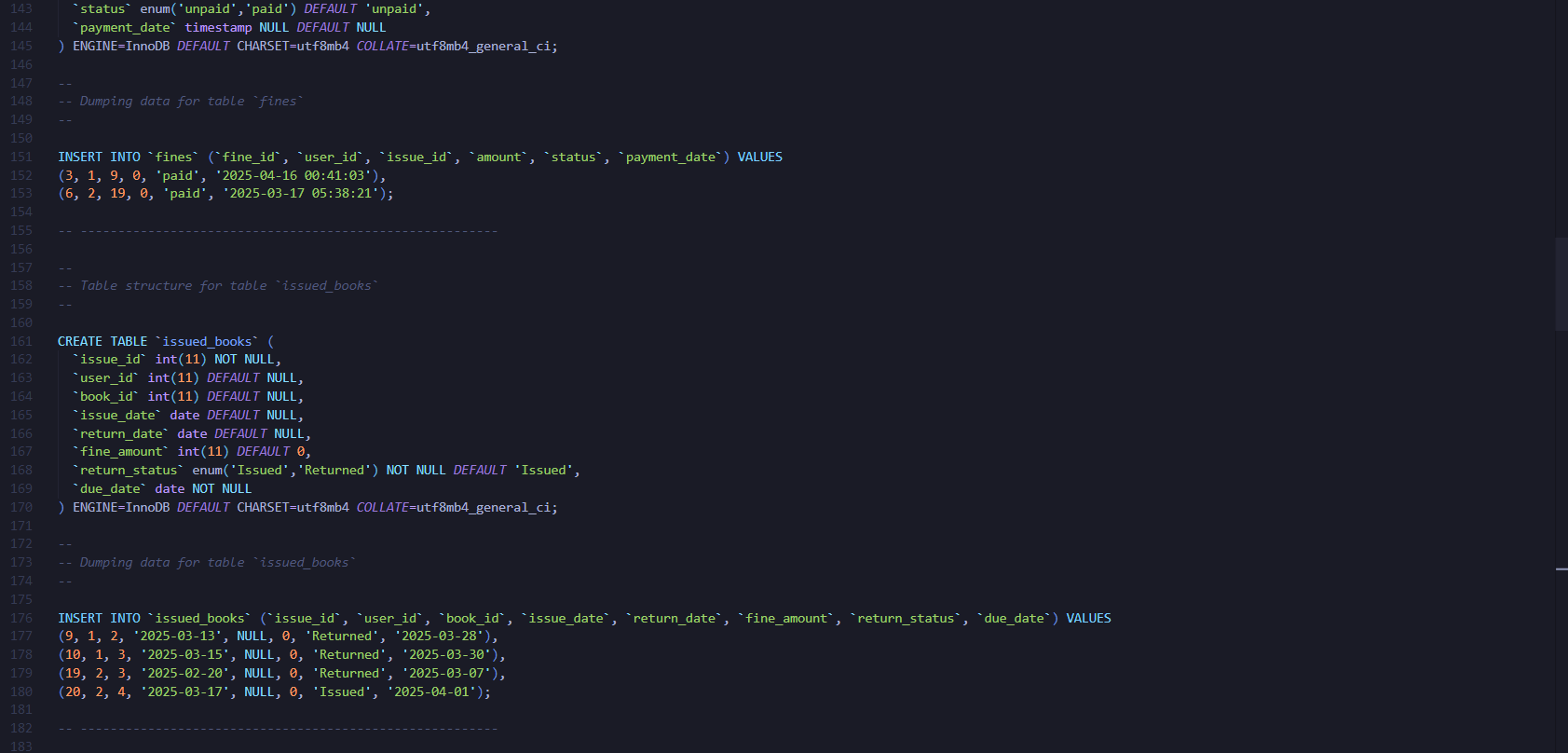
**Description:- Users can see issued books and it’s fine amount.**

****

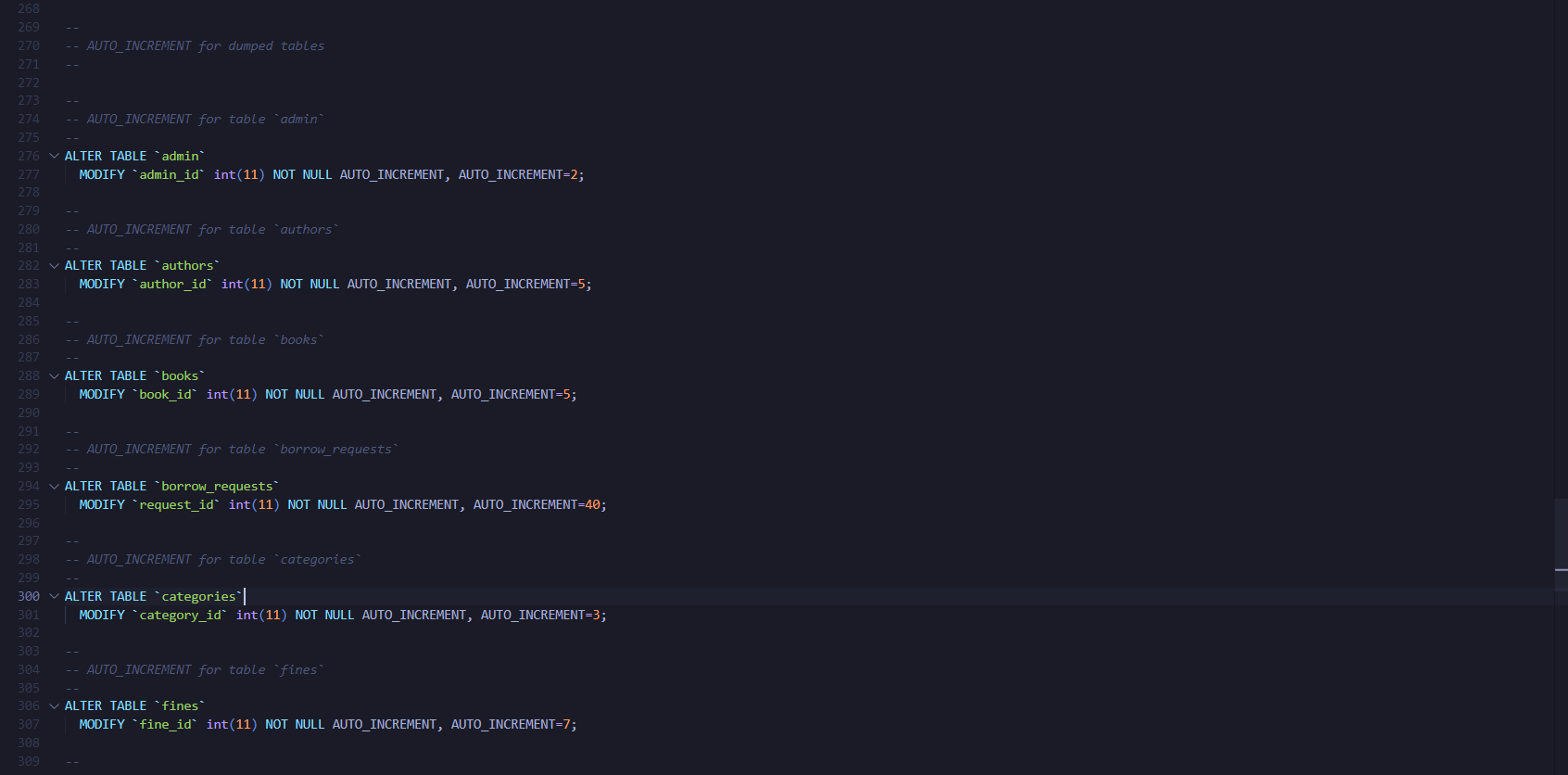
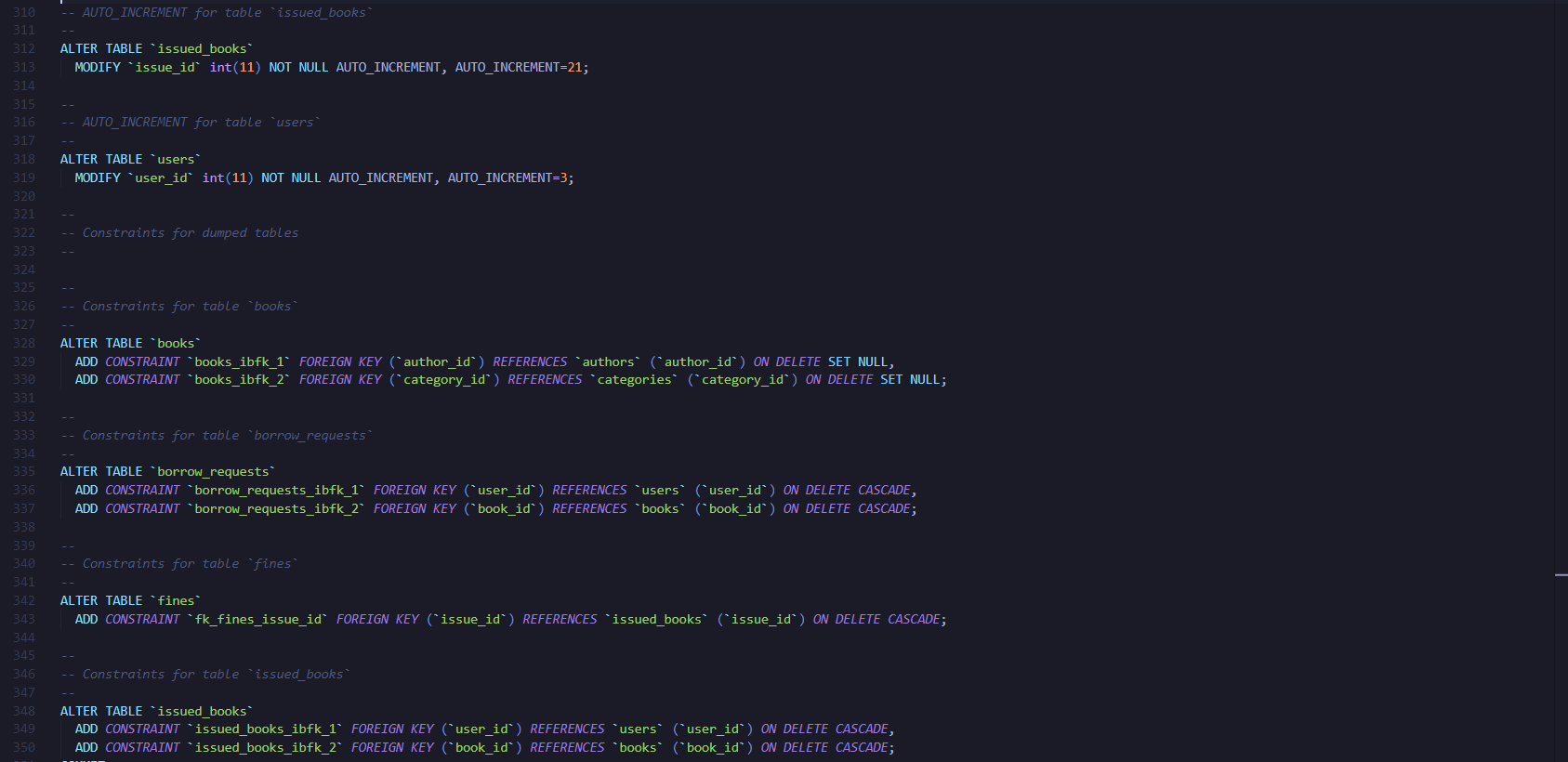
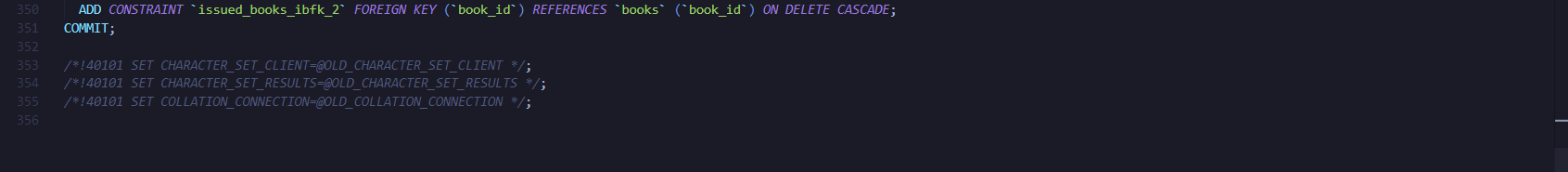
**Description:- Test cases with error code.**

**Code:**

****

****



** **

**THANK YOU**