**Library Management System**

**Submitted By:**

**Group 14**

* **Santhosh Mani(A20518627)**
* **Sheeram Venkatesh(A20519833)**
* **Ravishankar Sivagnanam(A20503091)**

**Index**

* **Abstract**
* **Softwares Used**
* **Language Used**
* **Phase 1**
* **Phase 2**
* **Phase 3**
* **Output Screenshots**
* **Conclusion**

**Abstract**

The objective is to construct a library management application with a database backend to hold data on users, documents, and other items. We collect data for a single library's patrons for this project. The purpose of Online Library Management is to computerize, automate, and handle all other processes related to the management of member information, book issuance, and book returns. In many cases, the library's maintenance is made more accessible by computerization. Due to a significant reduction in manual labor, it lessens management's workload. The library's books, reports, journal articles, magazines, and theses are arranged systematically, and the library management system makes it possible to keep track of them. They have characteristics like authors, editors, edition, type, year of publication, etc. With the help of Java and MySQL, the project seeks to create and put into use a library management system that allows librarians and users to log in. Members and documents can be added, viewed, searched for, and deleted by librarians. Members may borrow, examine, search for, and return documents.

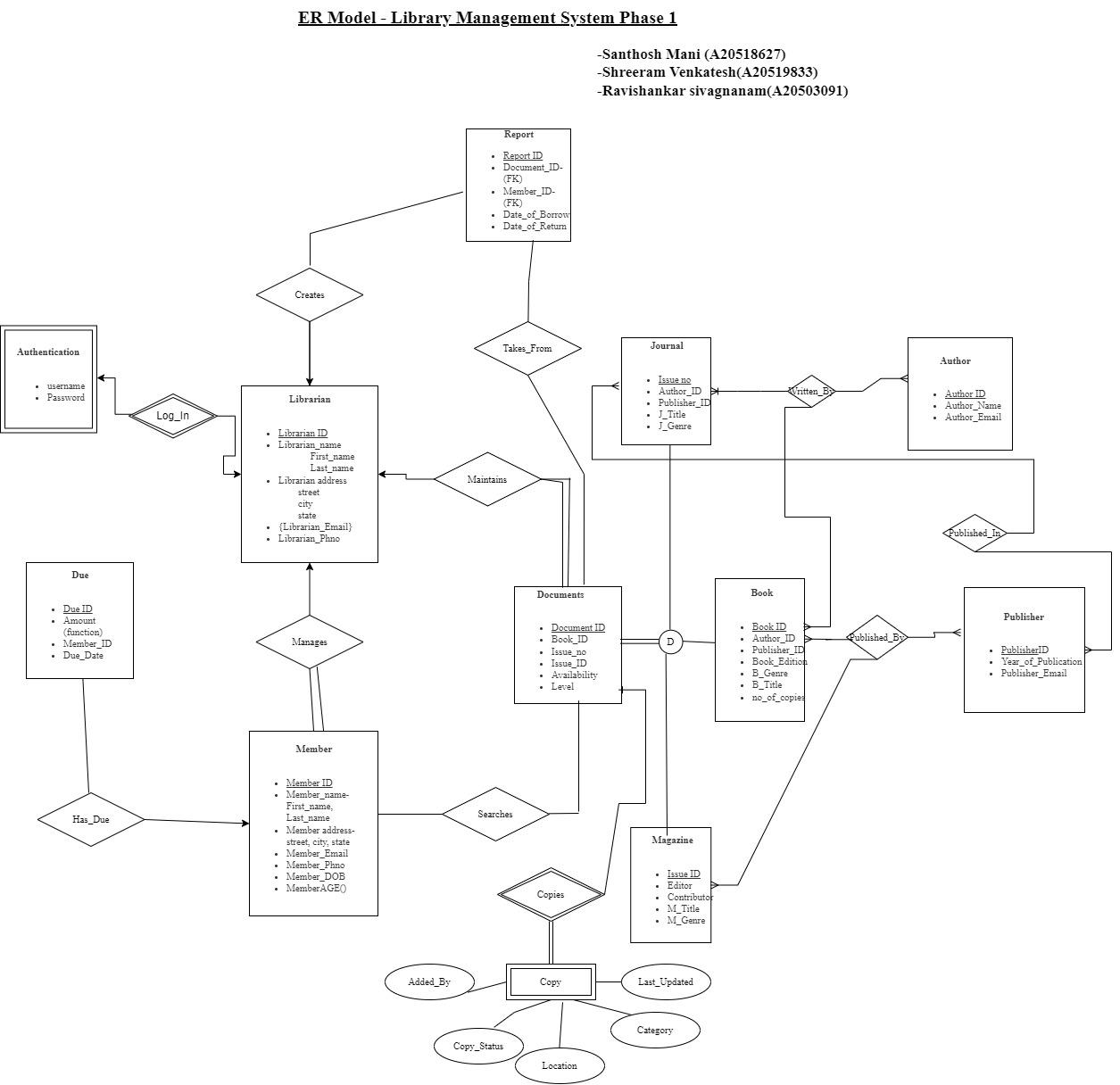
**Software Used :**

* PGAdmin - PGAdmin is a web-based GUI application used to communicate with Postgres database sessions on local and remote servers. You can use PGAdmin to carry out any kind of database administration necessary for a Postgres database.
* Eclipse using Maven - Computer programming is done with Eclipse, an integrated development environment. A base workspace and an expandable plug-in system are included for environment customization.

**Languages Used:**

* Postgresql
* Java
* HTML

**Phase 1: ER Model**



**Phase 2**

ER Model to Relational Schema: -

ENTITIES: -

Strong Entity: -

**Report** (Report\_ID, Librarian\_ID, Document\_ID, Member\_ID, Date\_of\_Borrow, Date\_of\_Return)

**Librarian** (Librarian\_ID, First\_name, Last\_name, Street, City, State, Librarian\_Phno)

**Librarian\_Email** (Librarian\_ID, Library\_Email)

**Due** (Due\_ID, Member\_ID, Due\_Date) – Member ID is the Foreign Key

**Member** (Member\_ID, Librarian\_ID, First\_name, Last\_name, Street, City, State, Member\_Phno, Member\_DOB) - Librarian ID is the Foreign Key

**Member\_Email** (Member\_ID, Member\_Email)

**Journal** (Issue\_no, Author\_ID, Publisher\_ID, J\_Title, J\_Genre)

**Author** (Author\_ID, Author\_name, Author\_Email)

**Book** (Book\_ID, Author\_ID, Publisher\_ID, Book\_Edition, B\_Genre, B\_Title, no\_of\_copies)

**Publisher** (Publisher\_ID, Year\_of\_Publication, Publisher\_Email)

**Magazine** (Issue\_ID, Editor, Contributor, M\_Title, M\_Genre)

**Documents** (Document\_ID, Book\_ID, Issue\_no, Issue\_ID, Availability, Level)

Weak Entity: -

**Copy** (Document\_ID, Added\_By, Copy\_Status, Location, Category, Last\_Updated)

**Authentication** (Librarian\_ID, Username, Password)

RELATION: -

**Takes\_From** (Report\_ID, Document\_ID)

**Searches** (Document\_ID, Member\_ID)

**Written\_By** (Book\_ID, Author\_ID, Issue\_no)

**Published\_in** (Publisher\_ID, Issue\_no)

**Published\_By** (Issue\_ID, Book\_ID, Publisher\_ID)

Weak Relation: -

**Log**\_**In** (Librarian\_ID)

Report Table consist of attributes Report\_ID, Librarian\_ID, Document\_ID, Member\_ID, Date\_of\_Borrow, Date\_of\_Return. It shows the overall list of documents, that are borrowed by the members and signed by the librarians.

The Librarian consists of attributes Librarian\_ID, First\_name, Last\_name, Street, City, State, and Librarian\_Phno which are the details of the librarians.

The due table consists of the details of the due dates for each document to return

The Member table consists of attributes Member\_ID, Librarian\_ID, First\_name, Last\_name, Street, City, State, Member\_Phno, Member\_DOB.

The Journal table consists of details of the journals in the library like Issue\_no, Author\_ID, Publisher\_ID, J\_Title, and J\_Genre.

**The Book table consists of attributes** Book\_ID, Author\_ID, Publisher\_ID, Book\_Edition, B\_Genre, B\_Title, and no\_of\_copies which are the details of books borrowed by the members.

**The Magazine table consists of attributes** Issue\_ID, Editor, Contributor, M\_Title, and M\_Genre which are the details of magazines in the library

**The Documents table consists of attributes** Document\_ID, Book\_ID, Issue\_no, Issue\_ID, Availability, Level

**Publisher** table consists of attributes Publisher\_ID, Year\_of\_Publication, Publisher\_Email

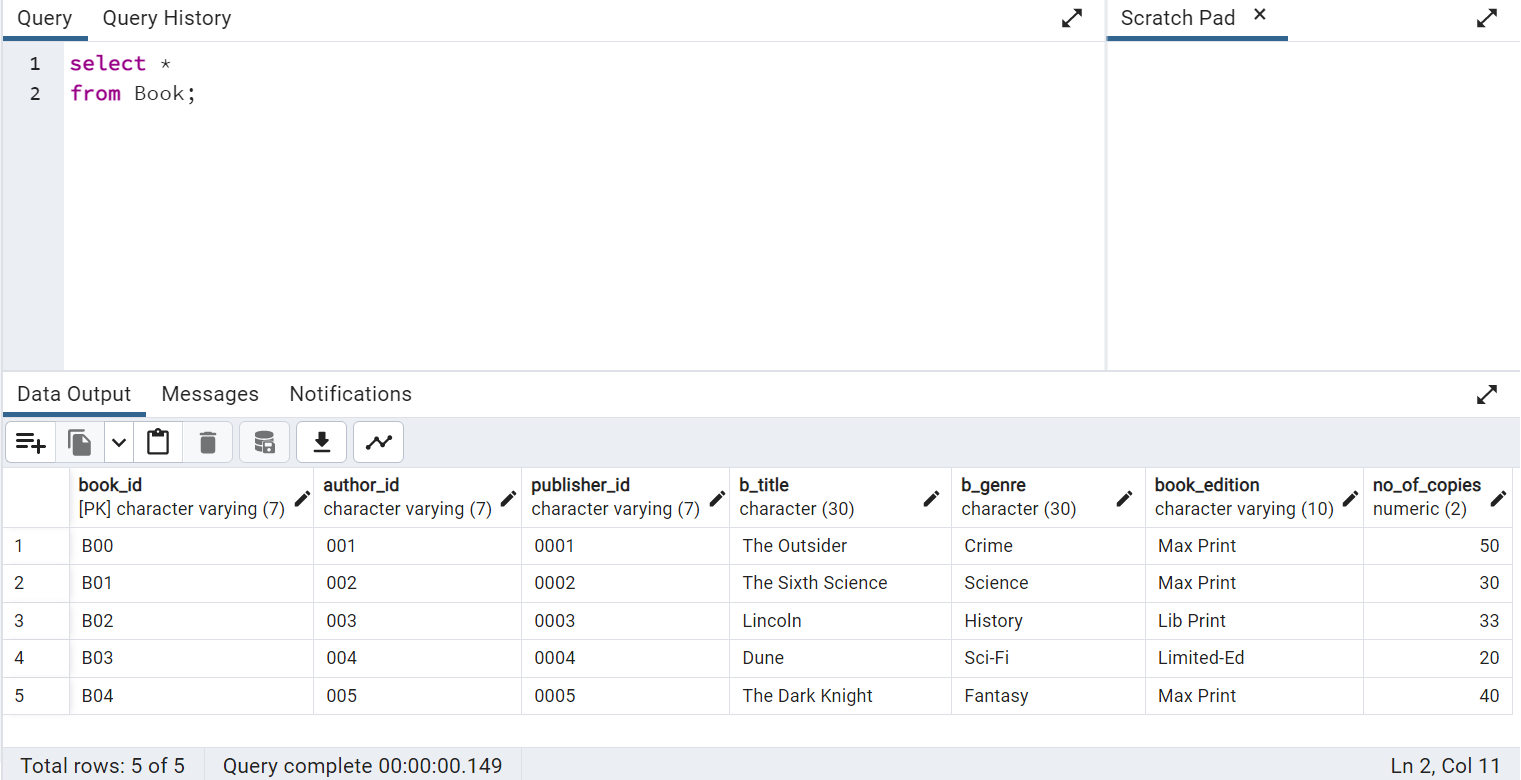
**The author** table consists of the details of the Author like Author\_ID, Author\_name, Author\_Email

Phase 3:

Here are the outputs for Webapplication

Output 1-5 we are inserting values in Postgresql and checking if it getting displayed in the query

Output 1:



Output 2:



Output 3:



Output 4:



Output 5:

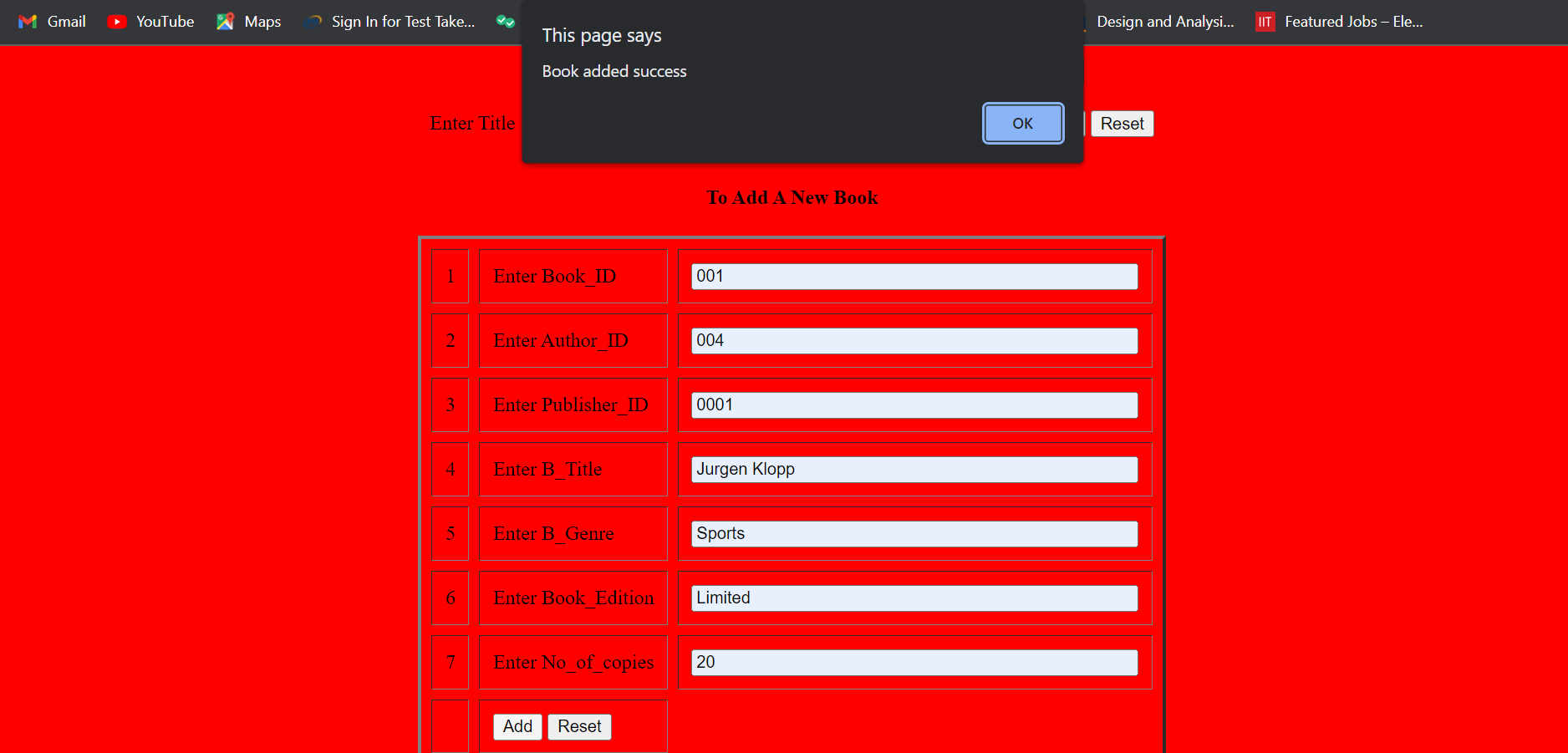


Output 6 is the webapplication of the library management system

Output 6:



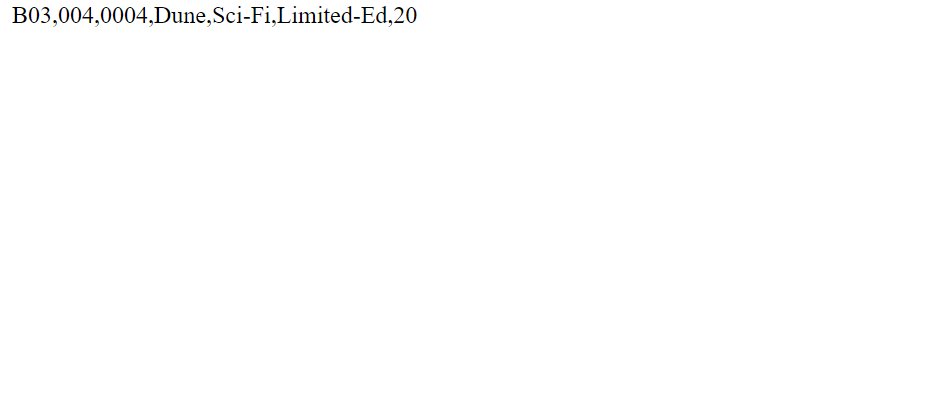
Output 7: We are adding the book details into the database



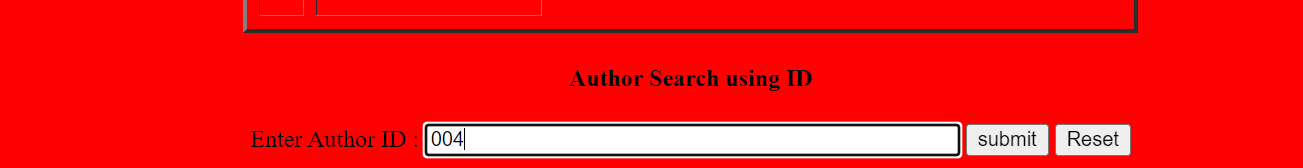
Output 8: We are searching the bookdetails using book title



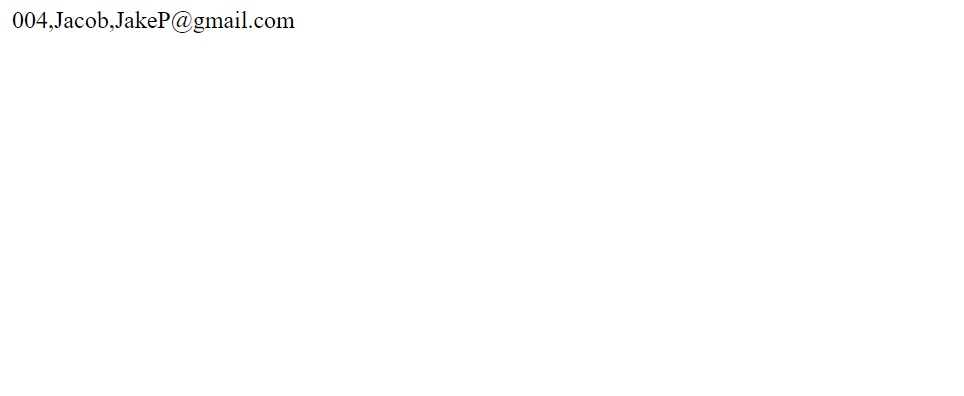
Output 9: Displays output of the search query



Output 10:We are searching author details using Author ID



Output 11:Displays the author details



CONCLUSION:

We have constructed a web based application using java eclipse maven and PostgreSQL interface as per our outcomes from Phase 1 and Phase 2 and we have shared the artifacts using a google drive link.