OOPS MINI PROJECT

LIBRARY MANAGEMENT SYSTEM

AIM:

To construct a database for the Library management system and connect it with my SQL using java.

Algorithm:

```
Start
1. Initialize the System:
2. Load the necessary JDBC driver.
3. Set up database connection credentials (URL, username, password).
4. Display Main Menu:
5. Show available options:
    a. Add a Book.
    b. Search for a Book.
    c. Register a Member.
    d. Issue a Book.
    e. Return a Book.
    f. Exit.
6. Handle User Choice:
7. Prompt the user to select an option from the menu.
8. Use switch or if-else to direct the flow to the respective operation.
9. Perform Operations:
10. Add Book:
    a. Collect book details (e.g., title, author, ISBN, quantity).
    b. Insert the data into the Books table in the database.
11. Search for a Book:
    a. Accept search criteria (e.g., title, author).
    b. Query the Books table and display matching records.
12. Register a Member:
    a. Collect member details (e.g., name, contact).
    b. Insert the data into the Members table.
13. Issue a Book:
    a. Check if the book is available.
    b. Update the Books table to decrement quantity.
    c. Add a record in the Transactions or IssuedBooks table.
14. Return a Book:
    a. Accept details about the return (e.g., book ID, member ID).
    b. Update the Books table to increment quantity.
    c. Update or delete the corresponding Transactions record.
15. Handle Errors Gracefully:
16. Catch and log exceptions (e.g., database errors, invalid inputs).
17. Provide user-friendly error messages.
```

18. Exit:
19. Close the database connection.
20. End the program gracefully.

SQL QUERIES:

Create the database

```
CREATE DATABASE library_management;

USE library_management;

-- Users table
CREATE TABLE users (
   user_id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(50) UNIQUE NOT NULL,
   password VARCHAR(50) NOT NULL,
   email VARCHAR(50) NOT NULL,
   is_admin BOOLEAN DEFAULT FALSE
);

-- Books table
CREATE TABLE books (
   book_id INT AUTO_INCREMENT PRIMARY KEY,
   title VARCHAR(100) NOT NULL,
   author VARCHAR(100),
   category VARCHAR(50),
   isbn VARCHAR(20),
   status ENUM('available', 'issued') DEFAULT 'available'
);


-- Transactions table
CREATE TABLE transactions (
   transaction_id INT AUTO_INCREMENT PRIMARY KEY,
   user_id INT,
   book_id INT,
   issue_date DATE,
   due_date DATE,
   return_date DATE,
   FOREIGN KEY (user_id) REFERENCES users(user_id),
   FOREIGN KEY (book_id) REFERENCES books(book_id)
);
USE library_management;
INSERT INTO books (book_id,title, author, category, isbn, status)
VALUES
(1,'The Great Gatsby', 'F. Scott Fitzgerald', 'Fiction', '9789516235', 'available'),
(2,'1984', 'George Orwell', 'Dystopian', '97804515248', 'available'),
(3,'To Kill a Mockingbird', 'Harper Lee', 'Fiction', '9780061120', 'issued'),
(4,'Moby-Dick', 'Herman Melville', 'Adventure', '9781853260', 'available'),
(5,'Pride and Prejudice', 'Jane Austen', 'Romance', '97801414395', 'available'),
(6,'Project Java',"done by govarthan and Gunaseelan","and Gunaseelan",'964263836','available');
select * from books
ALTER TABLE books ADD year INT;
INSERT INTO books (book_id,title, author, category, isbn, status)
```

```
VALUES (6,'Project Java',"done by govarthan and Gunaseelan","and
Gunaseelan",'964263836','available');
INSERT INTO books (book_id,title, author, category, isbn, status)
VALUES (7,'java mini project',"Done by Gunaseelan ","and Gunaseelan",'94284263836','available');
```

addBook.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "Gova@12345";  // Update this with your actual password
$dbname = "library_management";  // Ensure this matches your database name

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$title = $_POST['title'];
$author = $_POST['author'];
$category = $_POST['category'];
$isbn = $_POST['isbn'];
$status = $_POST['status'];
$sql = "INSERT INTO books (title, author, category, isbn, status)
VALUES ('$title', '$author', '$category', '$isbn', '$status')";
if ($conn->query($sql) === TRUE) {
    echo "New book added successfully!";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

DatabaseConnection.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String DB_URL =
"jdbc:mysql://localhost:3306/library_management?useSSL=false&serverTimezone=UTC";
    private static final String USER = "root";  // Replace with your MySQL username
    private static final String PASS = "Gova@12345";
    public static Connection getConnection() throws SQLException {
        Connection conn = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println("Connection established successfully.");
        } catch (ClassNotFoundException e) {
            System.err.println("MySQL JDBC Driver not found.");
            e.printStackTrace();
        } catch (SQLException e) {
            System.err.println("Failed to connect to the database.");
            throw e;
        }
        return conn;
    }
    public static void closeConnection(Connection conn) {
        try {
            if (conn != null && !conn.isClosed()) {
                conn.close();
                System.out.println("Connection closed.");
            }
        } catch (SQLException e) {
            System.err.println("Failed to close the connection.");
            e.printStackTrace();
        }
    }
}
```

LibraryManager.java

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class LibraryManager {

    public void addBook(String title, String author, String category, String isbn) {
        String query = "INSERT INTO books (title, author, category, isbn) VALUES (?, ?, ?, ?)";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {

            stmt.setString(1, title);
            stmt.setString(2, author);
            stmt.setString(3, category);
            stmt.setString(4, isbn);
            stmt.executeUpdate();
            System.out.println("Book added to the library.");

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Main.java

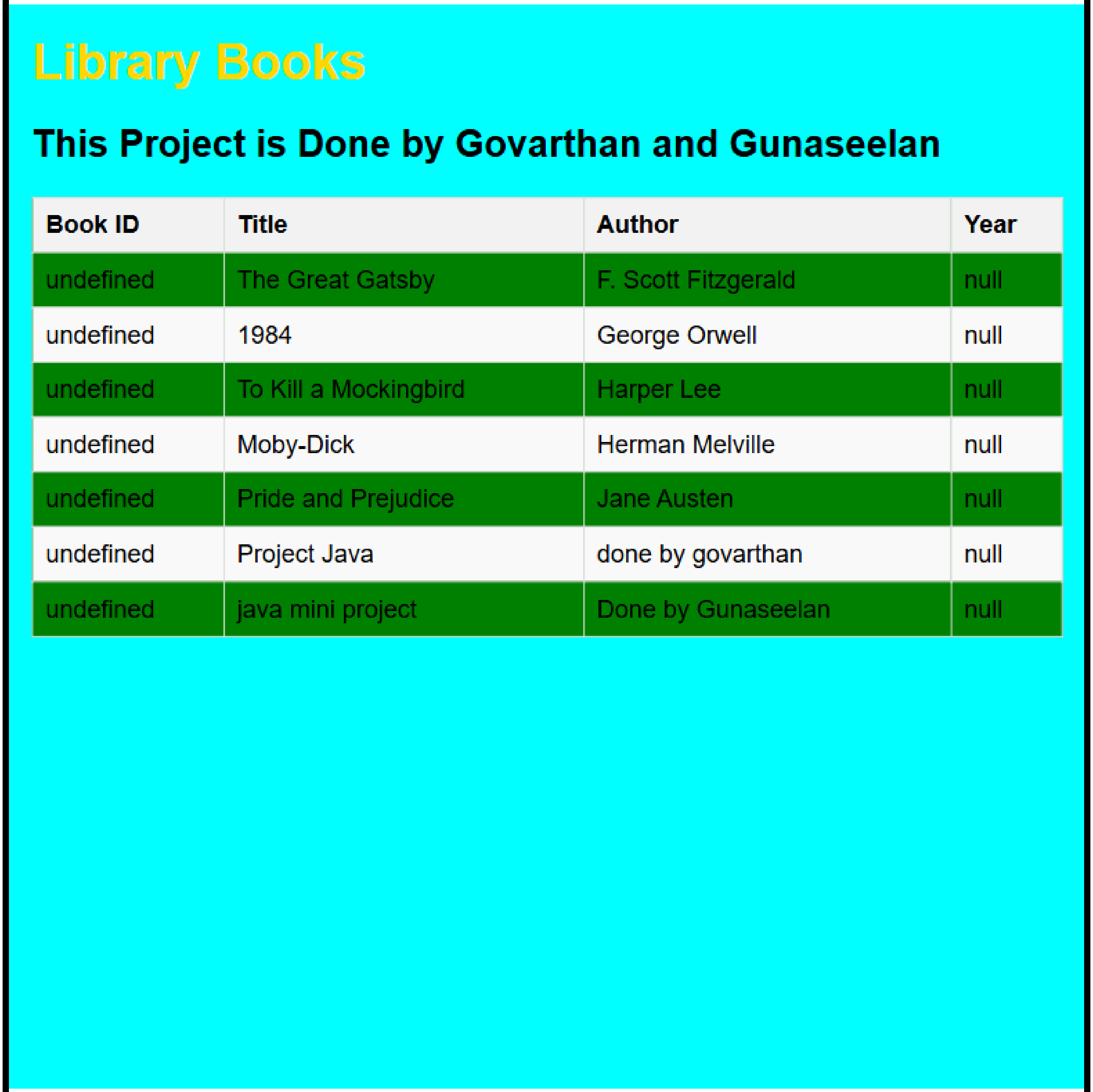
```
import java.sql.Connection;
```

```
public class Main {
    public static void main(String[] args) throws Exception{
        Connection conn = DatabaseConnection.getConnection();
        DatabaseConnection.closeConnection(conn);
    }
}
```

Server.js

```
const express = require('express');
const mysql = require('mysql2');
const path = require('path');

const app = express();

app.use(express.static(path.join(__dirname, 'public')));

const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'Gova@12345',
  database: 'library_management'
});

app.get('/api/books', (req, res) => {
  connection.query('SELECT * FROM books', (err, results) => {
    if (err) {
      res.status(500).json({ error: 'Failed to fetch data from database' });
      return;
    }
    res.json(results);
  });
});

app.listen(3000, () => {
  console.log('Server running on http://localhost:3000');
});
```

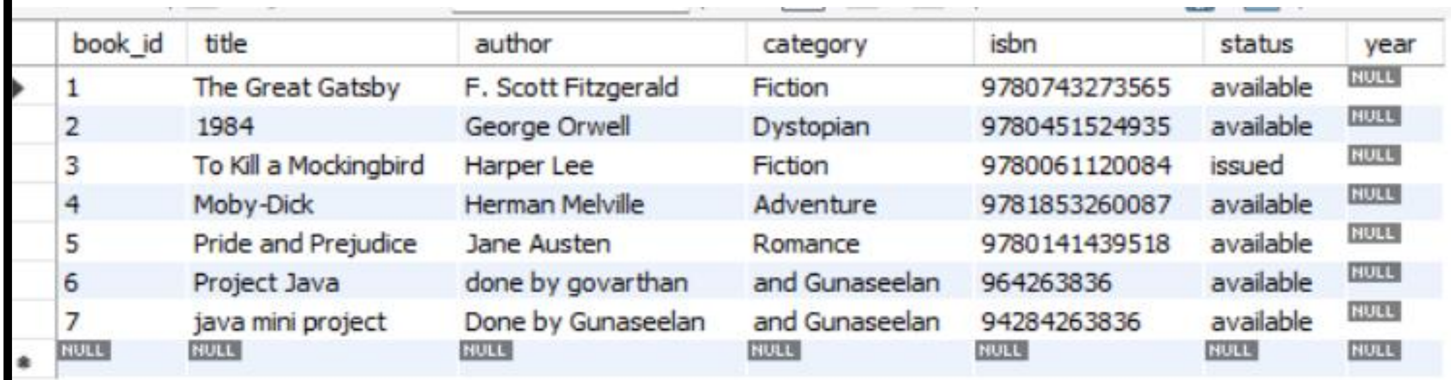
OUTPUT:



Library Books

This Project is Done by Govarthan and Gunaseelan

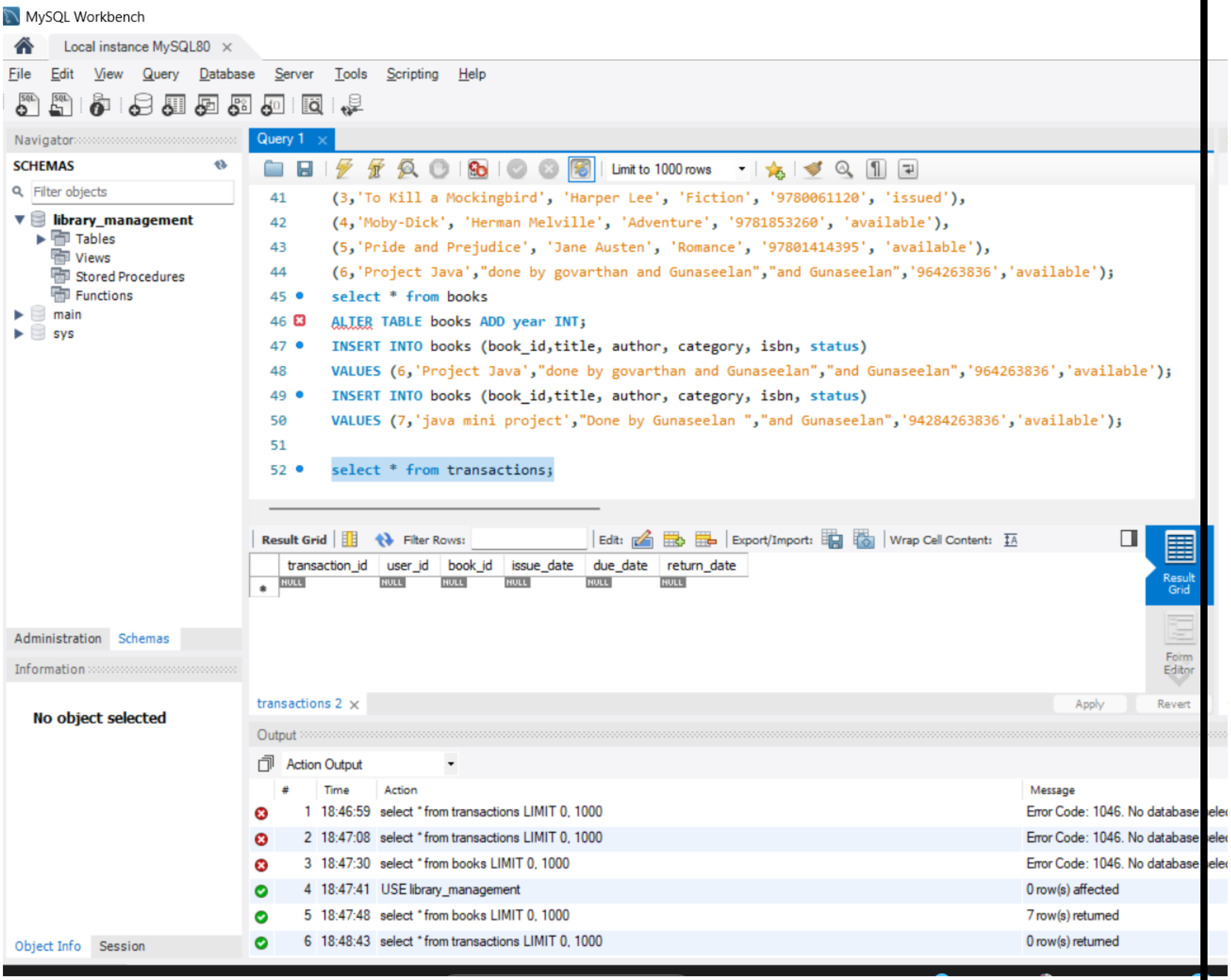
Book ID	Title	Author	Year
undefined	The Great Gatsby	F. Scott Fitzgerald	null
undefined	1984	George Orwell	null
undefined	To Kill a Mockingbird	Harper Lee	null
undefined	Moby-Dick	Herman Melville	null
undefined	Pride and Prejudice	Jane Austen	null
undefined	Project Java	done by govarthan	null
undefined	java mini project	Done by Gunaseelan	null



book_id	title	author	category	isbn	status	year
1	The Great Gatsby	F. Scott Fitzgerald	Fiction	9780743273565	available	NULL
2	1984	George Orwell	Dystopian	9780451524935	available	NULL
3	To Kill a Mockingbird	Harper Lee	Fiction	9780061120084	issued	NULL
4	Moby-Dick	Herman Melville	Adventure	9781853260087	available	NULL
5	Pride and Prejudice	Jane Austen	Romance	9780141439518	available	NULL
6	Project Java	done by govarthan	and Gunaseelan	964263836	available	NULL
7	java mini project	Done by Gunaseelan	and Gunaseelan	94284263836	available	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Note: Some character types are not supporting in website.

	transaction_id	user_id	book_id	issue_date	due_date	return_date
*	NULL	NULL	NULL	NULL	NULL	NULL



```
(3,'To Kill a Mockingbird', 'Harper Lee', 'Fiction', '9780061120', 'issued'),
(4,'Moby-Dick', 'Herman Melville', 'Adventure', '9781853260', 'available'),
(5,'Pride and Prejudice', 'Jane Austen', 'Romance', '97801414395', 'available'),
(6,'Project Java',"done by govarthan and Gunaseelan","and Gunaseelan",'964263836','available');
select * from books
ALTER TABLE books ADD year INT;
INSERT INTO books (book_id,title, author, category, isbn, status)
VALUES (6,'Project Java',"done by govarthan and Gunaseelan","and Gunaseelan",'964263836','available');
INSERT INTO books (book_id,title, author, category, isbn, status)
VALUES (7,'java mini project',"Done by Gunaseelan ","and Gunaseelan",'94284263836','available');

select * from transactions;
```

	transaction_id	user_id	book_id	issue_date	due_date	return_date
*	NULL	NULL	NULL	NULL	NULL	NULL

	#	Time	Action	Message
✗	1	18:46:59	select * from transactions LIMIT 0, 1000	Error Code: 1046. No database selec
✗	2	18:47:08	select * from transactions LIMIT 0, 1000	Error Code: 1046. No database selec
✗	3	18:47:30	select * from books LIMIT 0, 1000	Error Code: 1046. No database selec
✓	4	18:47:41	USE library_management	0 row(s) affected
✓	5	18:47:48	select * from books LIMIT 0, 1000	7 row(s) returned
✓	6	18:48:43	select * from transactions LIMIT 0, 1000	0 row(s) returned

RESULT:

The database construction for the Library management system has been successfully complected and connected with mySQL using java.

MINI PROECT MEMBERS:

1. **GOVARTHAN V-- 231401030 – CSBS 'A'**
2. **GUNASEELAN S – 231401009 – CSBS 'A'**