LIBRARY MANAGEMENT SYSTEM

Name : P. Nethra Sri

Department : ECE

Register no. : 43130286

INTRODUCTION

The **Library Management System (LMS)** is a software application developed to automate and manage the day-to-day activities of a library. The traditional manual system of maintaining book records, student details, and transactions is time-consuming and error-prone. This project aims to replace the manual process with a computerized system that is efficient, accurate, and easy to use.

The system helps librarians manage books, students, and borrowing activities effectively. It provides quick access to information, reduces paperwork, and improves overall library operations. The Library Management System ensures proper maintenance of library data and enhances productivity.

ABSTRACT

The **Library Management System** is designed to manage library resources such as books, students, and book transactions in a digital manner. The main objective of this project is to simplify the process of issuing and returning books while maintaining accurate records.

This system allows the admin to add, update, delete, and view books and student details. It also maintains transaction records for issued and returned books. The backend of the system is developed using **Spring Boot**, **MySQL**, and **Postman** for testing REST APIs. The system ensures data consistency, security, and faster access to information.

CLIENT REQUIREMENT

Client wants a Library Management System where admin can check all student details, book details and transaction records.

Admin should be able to track which student issued which book and return status.

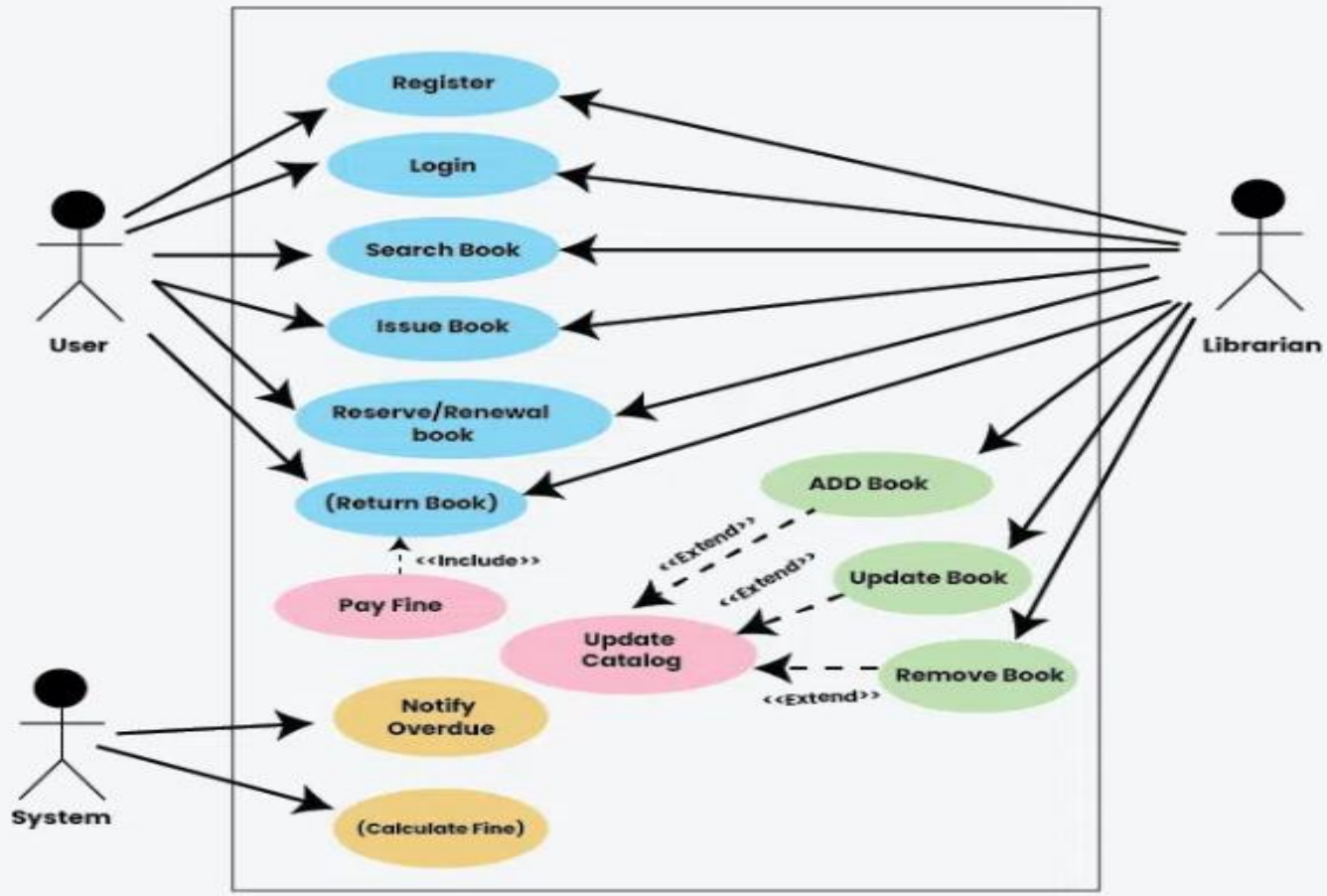
WE HAVE TO BUILD SIMPLE LIBRARY MANAGEMENT SYSTEM

- Admin should be able to create, delete, update and view students.
- Admin should be able to create, delete, update and view books.
- Student should be able to issue more than one book.
- Book issue should be categorized student wise.
- Transactions should be categorized date wise.

TECHNOLOGIES AND TOOLS USED

- Framework : Spring Boot
- Java 8+
- Maven
- Spring Tool Suite (STS)
- Apache Tomcat
- Spring Core, Spring Security (JWT), Spring Data JPA
- MySQL Database
- Postman REST Client

Use Case Diagram for Library Management System Design



SYSTEM REQUIREMENTS

SOFTWARE REQUIREMENTS :

- Back End : Spring Boot, JPA Hibernate, Postman
- DB Server : MySQL Server 8.0
- Web Server : Apache Tomcat
- IDE : Spring Tool Suite (STS)
- Browser : Google Chrome
- Test Tool : JUnit

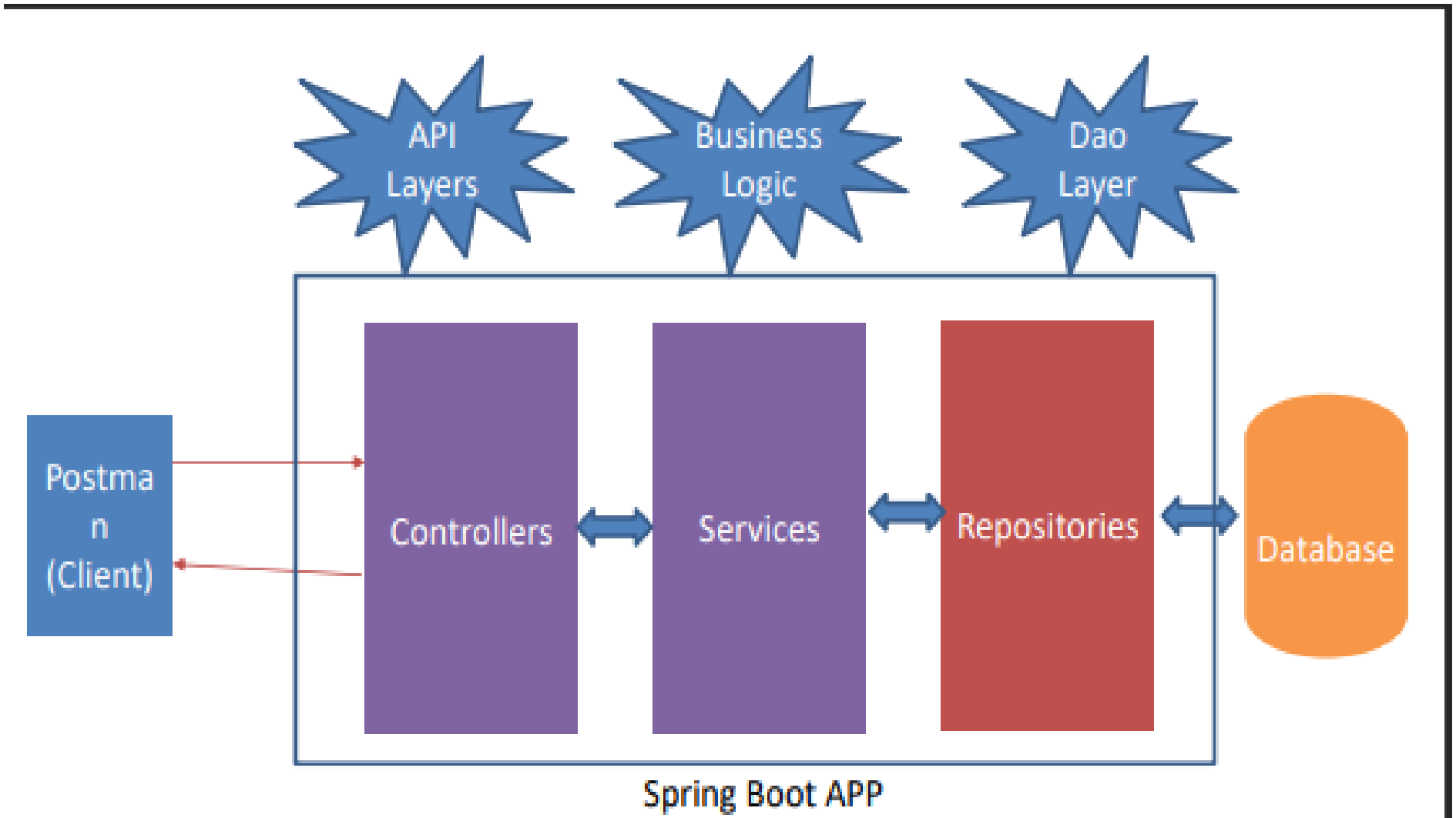
CLIENT SERVER ARCHITECTURE :

- Client sends request through Web Application or Postman.
- Server processes request using Controllers, Services and Repositories.
- Data exchange happens using JSON format.

HTTP REQUEST METHODS

- GET - Fetch data
- POST - Insert data
- PUT - Update data
- DELETE - Delete data

- URI is used to identify resources.


API Layers
Business Logic
Dao Layer
Postman (Client)
Controllers
Services
Repositories
Database
Spring Boot APP

ADMIN MODULE

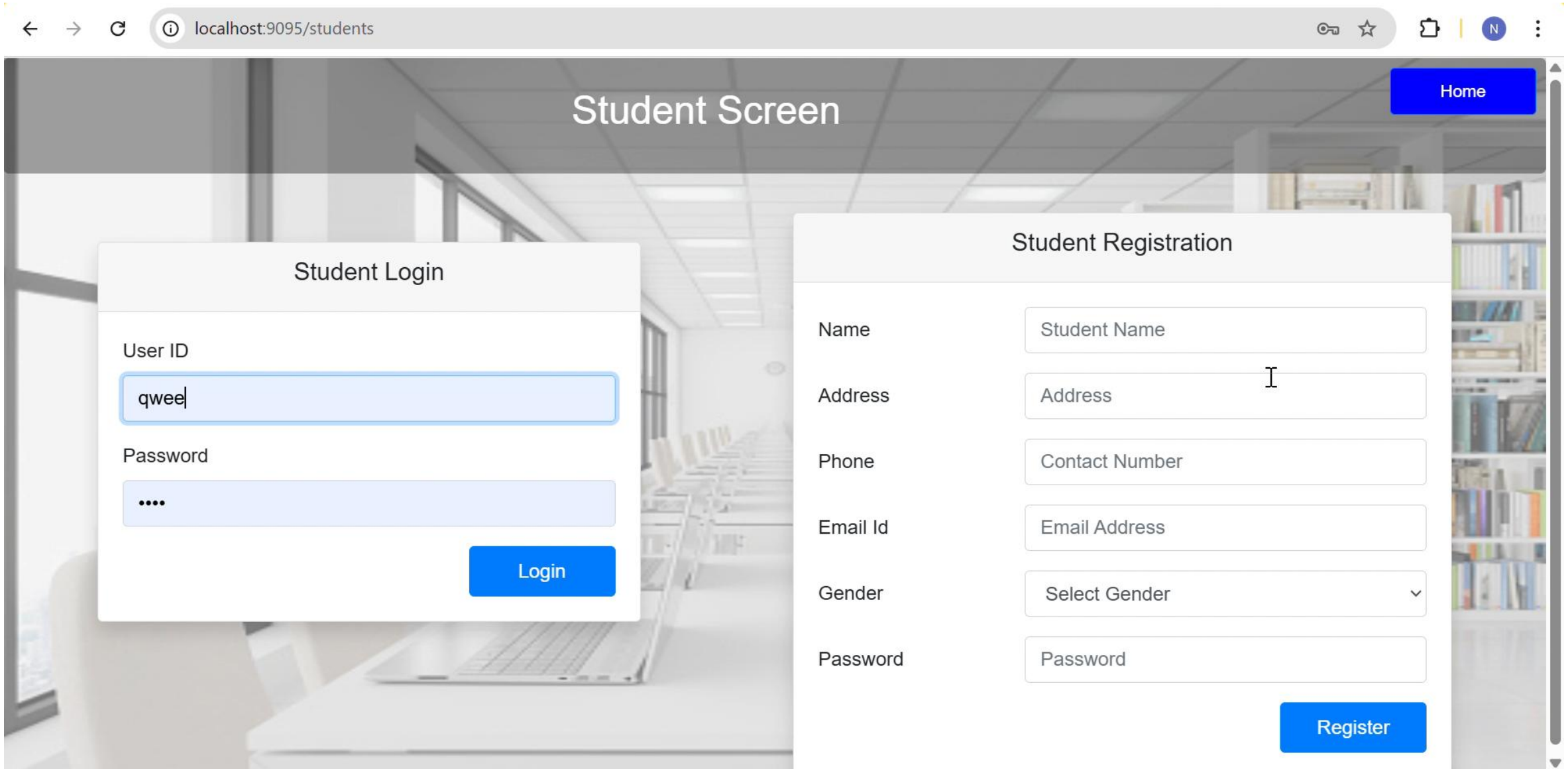
- Admin can move after login successfully.
- Admin can save admin details using Postman tool.
- Admin can add details using POST mapping.
- Admin can get all details using GET mapping.
- Admin can update details using PUT mapping.
- Admin can delete details using DELETE mapping.
- After completion of Admin module, Student module is created as child
- table.

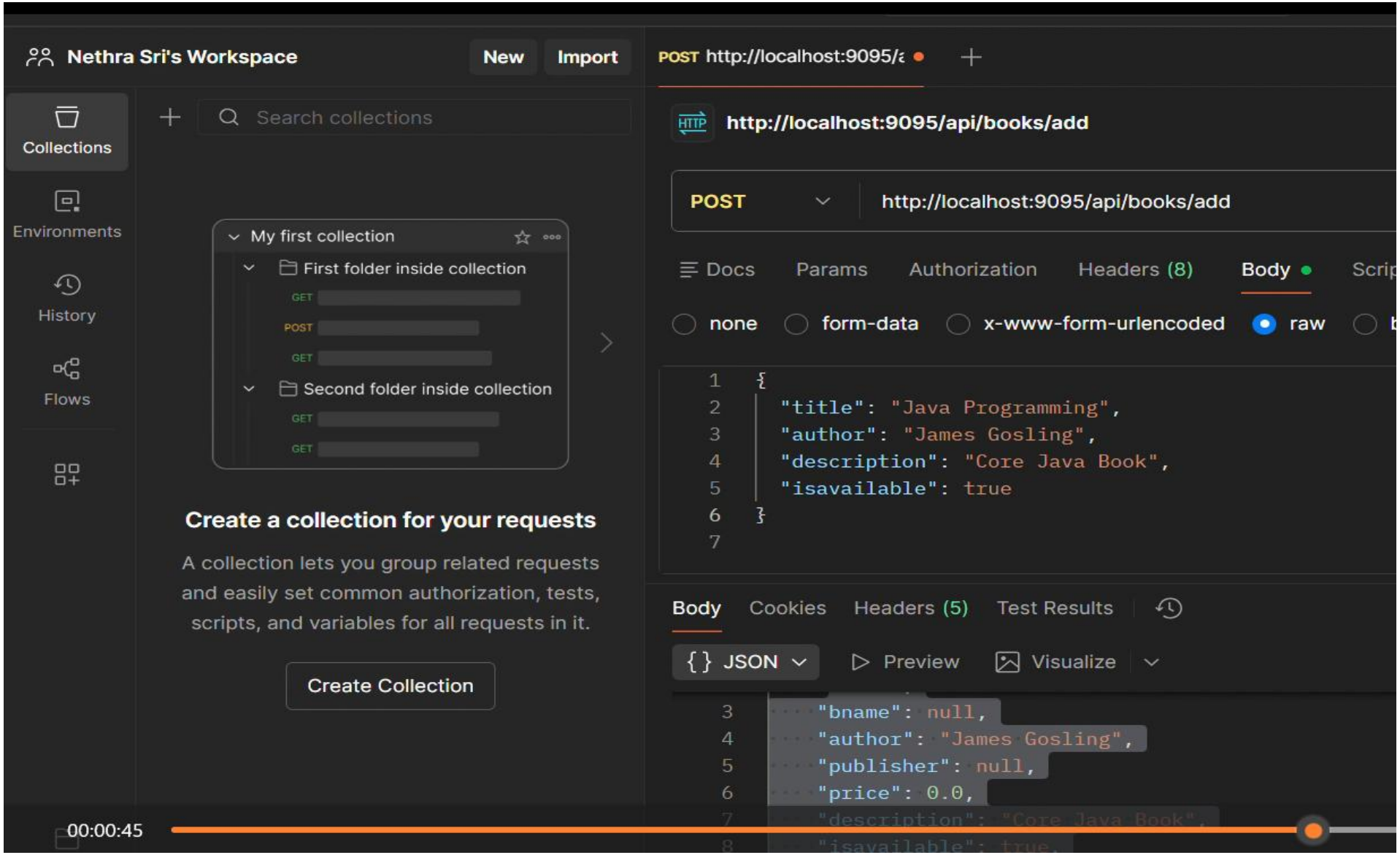
STUDENT MODULE

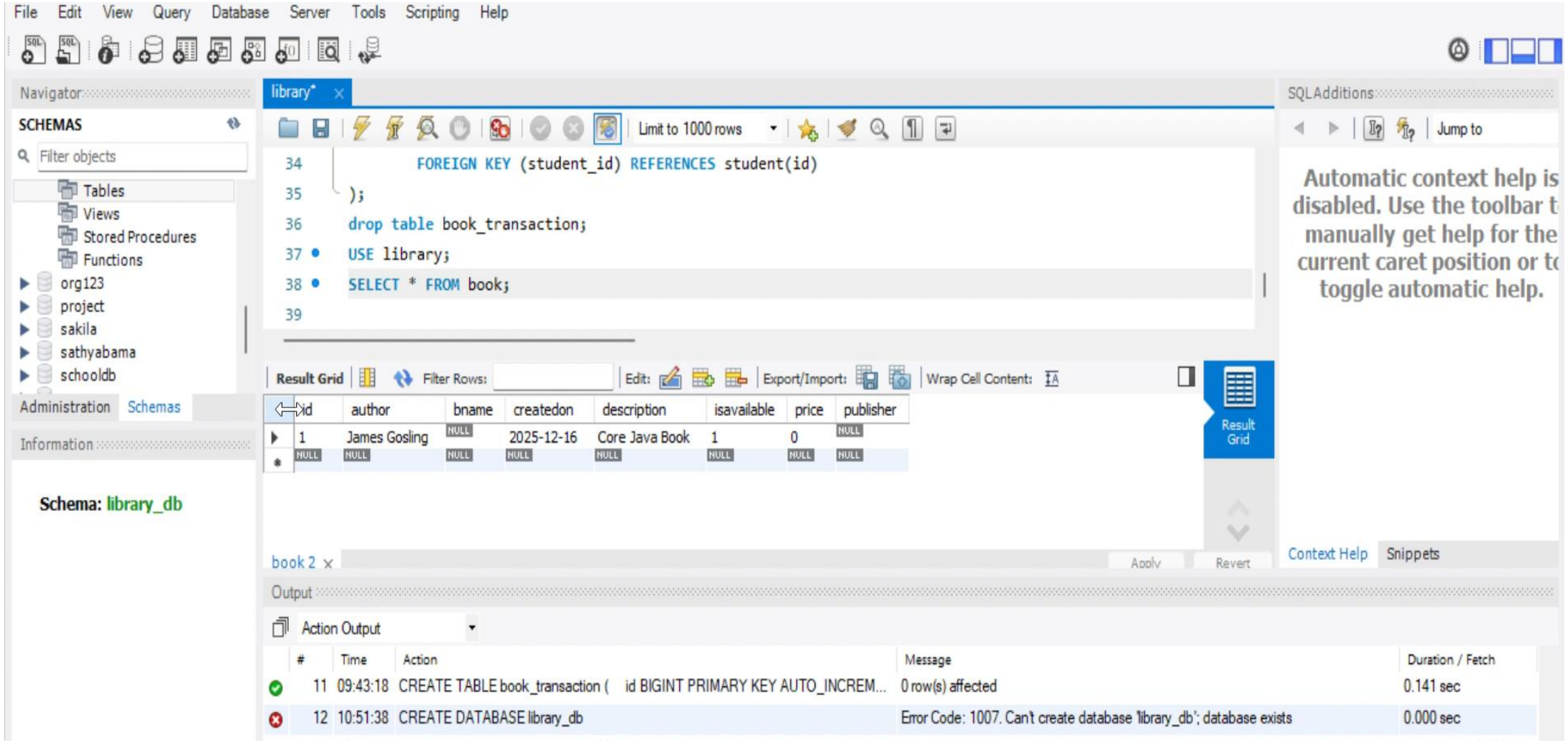
- After completing Admin Module, we move to Student Module.
- Student details are added using Postman tool.
- POST mapping is used to save student details.
- GET mapping is used to get list of students.
- PUT mapping is used to update student details.
- DELETE mapping is used to delete student details.
- Student module is connected with Book module.

BOOK MODULE

- After completing Student Module, we move to Book Module.
- Book details are added using Postman tool.
- POST mapping is used to add books.
- GET mapping is used to get book details.
- PUT mapping is used to update book details.
- DELETE mapping is used to delete book details.
- Book module is parent for Transaction module.


localhost:9095/students
Student Screen
Home
Student Login
User ID
qwee
Password
••••
Login
Student Registration
Name
Student Name
Address
Address
Phone
Contact Number
Email Id
Email Address
Gender
Select Gender
Password
Password
Register


Nethra Sri's Workspace
New
Import
Collections
Environments
History
Flows
Search collections
My first collection
First folder inside collection
GET
POST
GET
Second folder inside collection
GET
GET
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create Collection
POST http://localhost:9095/
http://localhost:9095/api/books/add
POST
http://localhost:9095/api/books/add
Docs
Params
Authorization
Headers (8)
Body
none
form-data
x-www-form-urlencoded
raw
{
"title": "Java Programming",
"author": "James Gosling",
"description": "Core Java Book",
"isavailable": true
}
Body
Cookies
Headers (5)
Test Results
JSON
Preview
Visualize
"bname": null,
"author": "James Gosling",
"publisher": null,
"price": 0.0,
"description": "Core Java Book",
"isavailable": true,
00:00:45


File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
Tables
Views
Stored Procedures
Functions
org123
project
sakila
sathyabama
schooldb
Administration Schemas
Information
Schema: library_db
library*
Limit to 1000 rows
34 FOREIGN KEY (student_id) REFERENCES student(id)
35);
36 drop table book_transaction;
37 USE library;
38 SELECT * FROM book;
39
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
id author bname createdon description isavailable price publisher
1 James Gosling NULL 2025-12-16 Core Java Book 1 0 NULL
NULL NULL NULL NULL NULL NULL NULL NULL
Result Grid
book 2
Apply Revert
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
11 09:43:18 CREATE TABLE book_transaction (id BIGINT PRIMARY KEY AUTO_INCREM... 0 row(s) affected 0.141 sec
12 10:51:38 CREATE DATABASE library_db Error Code: 1007. Can't create database 'library_db'; database exists 0.000 sec