

Case Study

On

Library Management System

INTRODUCTION

- The Project aims to automate the process of **Library Management System** and Make the process of recruitment process.
- It provides a platform where the **librarian can view, add, update, and delete books** available in the library.
- It allows the **librarian to manage student details**, including adding and viewing registered students.
- It provides information about **book availability**, helping in proper issue and return of books.

ABSTRACT

Library Management System (LMS). The main objective of the **Library Management System** is to develop software that manages library activities in a college or institution.

It provides an interactive platform where the **librarian can manage all library operations** such as book records, student details, and book issue/return activities from a single console.

The system helps in maintaining **complete details of all books** in a digital format. It allows the librarian to easily **add, update, view, and delete books**, as well as manage student information efficiently.

What we are going to build: Client Requirement

The client wants a **Library Management System** where the **admin (librarian)** can manage all library-related activities such as **books, students, and book issue/return records.**

The system should provide a centralized platform to efficiently control and track library operations.

We have to build a **simple Library Management System** with the following requirements:

- **Admin (Librarian)** should be able to **create, update, delete, and view books** in the library.

- **Students** should be able to **borrow more than one book** (as per library rules).
- The system should **track book availability** and update quantity automatically after issue and return.
- **Admin** should be able to **create, update, delete, and view student records**.
- The system should maintain **complete issue and return history** of books.
- The admin should be able to **view all issued and returned book records** at any time.

What we are going to build: Some technical terms.

- Login and Register APIs for Admin (Librarian)
- Book APIs with pagination and sorting
- Proper user input validation
- Exception handling for errors like book not found or out of stock
- Role-based security (Admin access)
- JWT-based authentication for secure APIs

What are the technologies and tools we are going use ?

- Framework: **Spring Boot** Java Framework.
 - Java 8+
 - Maven
 - STS
 - Apache Tomcat
 - Spring core, Spring security(jwt) , Spring data JPA(Hibernate) et.c
- MySQL Database.
- Postman

SYSTEM REQUIREMENTS

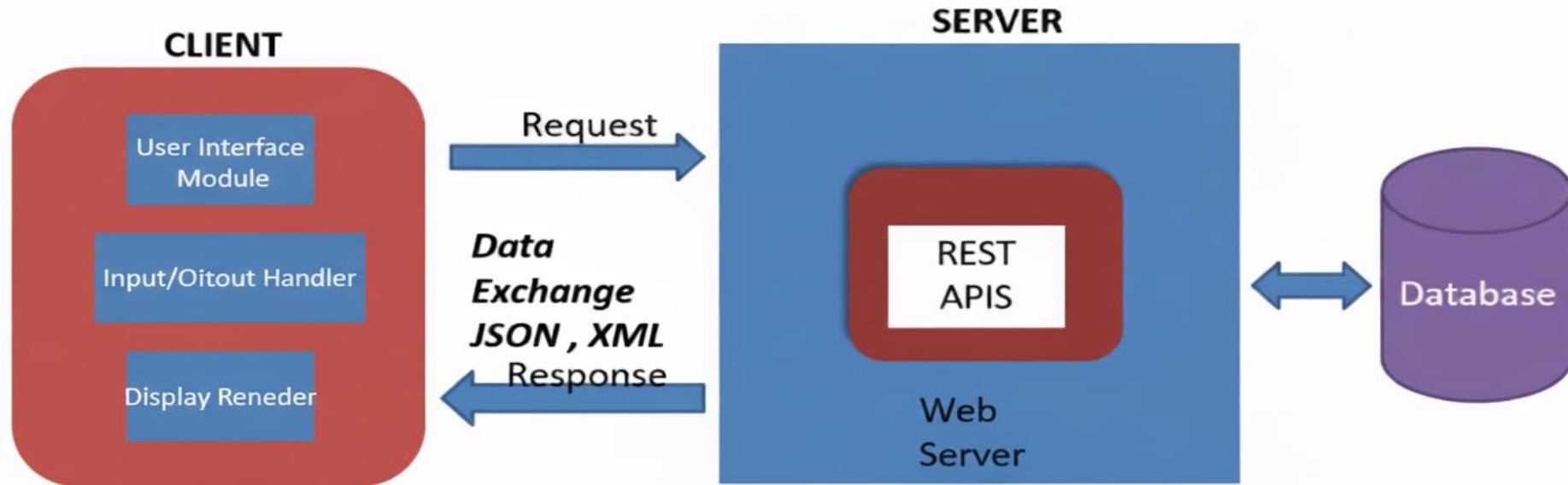
SOFTWARE REQUIREMENT :

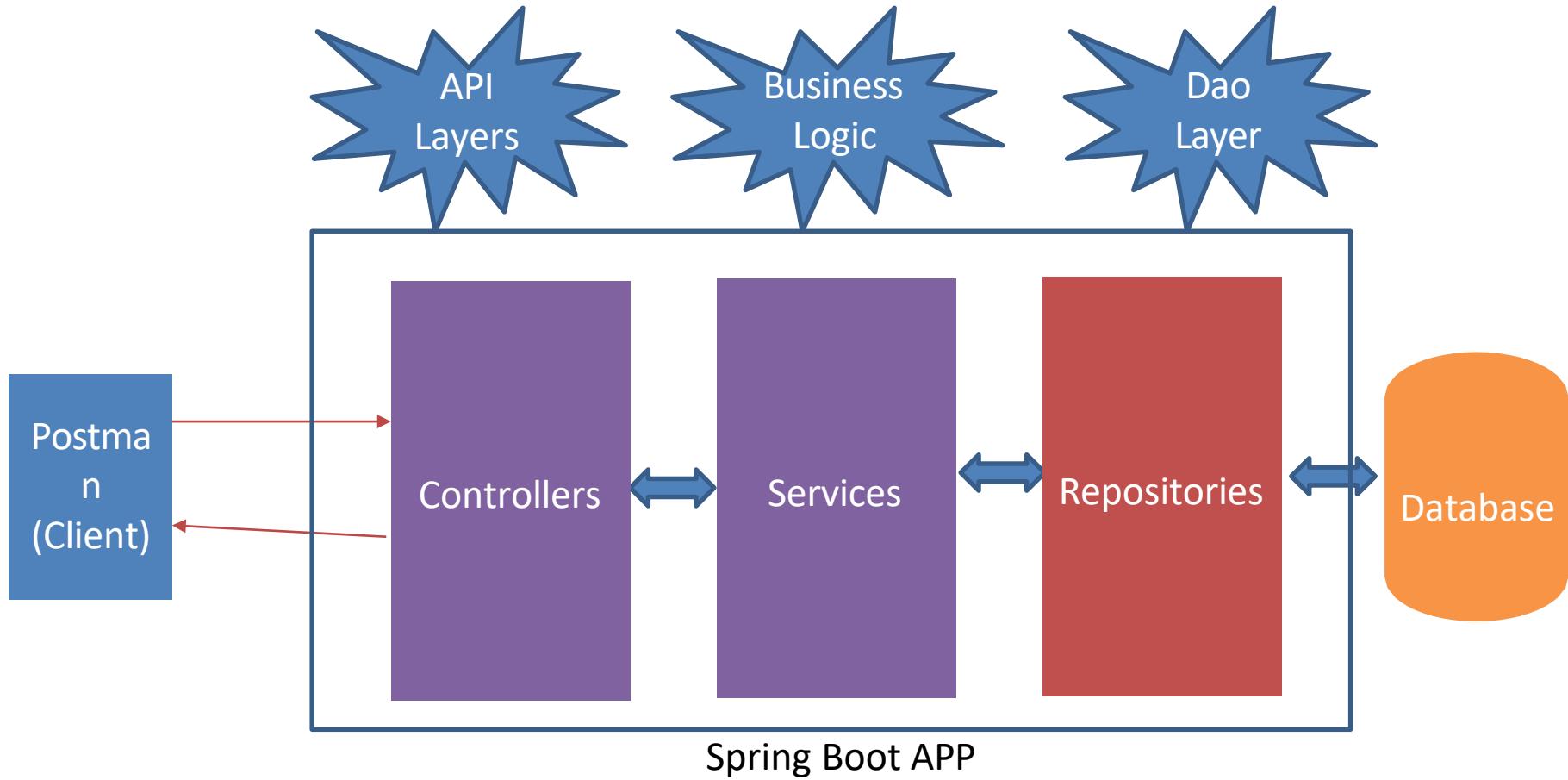
- Back End:** Spring Boot , JPA Hibernate and Postman Tool
- DB Server:** MySQL Server 8.0.CE
- Web Server:** Apache Tomcat
- IDE:** Spring Tool Suite(STS)
- Browser:** Google Chrome

PROJECT MODULE

- ❖ Admin (Librarian) Module
- ❖ Book Management Module
- ❖ Book Delete Module
- ❖ Book Addition Module

Client – Server Architecture





ADMIN MODULE

- In Admin Module Can move after login successfully. Admin Can save their details by using Postman Tool.
- Admin module can save admin details by using post mapping, Admin can get all details by using Get Mapping , Admin can Update details by using Put Mapping , Admin can delete their details By using Delete Mapping.
- In Admin can contain Test methods we mention Save , List of Admins , update and Delete methods.
- After Completion Admin we can add the books of record to the database .

Http Request Methods

HTTP defines a set of **request methods** to indicate the desired action to be performed for a given resource

GET	http://localhost:9080/login	Return to the admin login page
GET	http://localhost:9080/books	Return to the books section
GET	http://localhost:9080/book/1	Return to the book of id 1
PUT	http://localhost:9080/update/1	Update a book of id 20
DELETE	http://localhost:9080/delete/2	Delete a book of id 2

Http Methods



URI (Uniform Resource Identifier)

- ✓ URI is used to identify resource.

GET	http://localhost:9080/login	Return the login page of admin
GET	http://localhost:9080/book/1	Return the book of id 1
POST	http://localhost:9080/addbook/	Create a new book
PUT	http://localhost:9080/update/2	Update a book of id 2
DELETE	http://localhost:9080/category/2	Delete a category of id 2

BOOK MANAGEMENT MODULE

- After successful login, the Admin can save, retrieve, update, and delete book details by using the Postman Tool via REST API mappings (Post, Get, Put, and Delete).
- **Testing and Validation:** The module contains dedicated Test methods to verify the functionality of Save, List of Books, Update, and Delete operations.
- **Hierarchical Structure:** The Book Module is created as a child table after the Admin module is complete.

BOOK DELETE MODULE

- **Safety Confirmation:** Implement a confirmation dialog (e.g., "Are you sure?") to prevent accidental clicks.
- **Status Validation:** Ensure the book isn't currently checked out or reserved before allowing deletion.
- **Soft Delete:** Instead of permanent erasure, mark the book as inactive in the database so it can be recovered if needed.
- **Audit Trail:** Automatically log the date, time, and user ID of the person who performed the deletion for accountability.

BOOK UPDATE MODULE

DATA DICTIONARY

TABLES OF DATABASE

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

user 2

Action Output

#	Time	Action
1	22:38:23	select * from publishers LIMIT 0, 1000
2	22:38:25	use library_management
3	22:38:27	show tables from library_management
4	22:38:32	select * from user LIMIT 0, 1000

Message

Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...
0 row(s) affected
13 row(s) returned
1 row(s) returned

Duration / Fetch

0.000 sec
0.000 sec
0.016 sec / 0.000 sec
0.000 sec / 0.000 sec

Object Info Session

20°C Mostly clear

Search

10:38 PM 12/18/2025

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Form Editor

Field Types

Apply Context Help Snippets

SQL Additions

Jump to

The screenshot shows the MySQL Workbench interface. In the top-left, there's a 'Navigator' panel with sections for MANAGEMENT, INSTANCE, and PERFORMANCE. Below it is an 'Information' panel. The main area has a 'Query 1' tab open with the following SQL code:

```
1 • create database library_management;
2 • use library_management;
3 • show tables from library_management;
4 • select * from author_details;
5 • select * from user;
6 • select * from authors;
7 • select * from author_details;
```

Below the code is a 'Result Grid' showing a single row of data from the 'user' table:

ID	Email	First Name	Last Name	Password
13	admin@admin.in	admin	admin	\$2a\$10\$KcGMXVfYAgMsCoC9MD6sN.NQ895sxq...

On the right, there's a vertical toolbar with icons for Result Grid, Form Editor, and Field Types. At the bottom, there's a 'Session' pane showing the execution history with details like time, action, message, duration, and fetch count. The system tray at the bottom shows the date and time (10:38 PM, 12/18/2025), battery level (20°C), and network status.

BOOKS DATABASE

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

books 2

Action Output

#	Time	Action
1	23:07:46	select * from books LIMIT 0, 1000
2	23:07:49	use library_management
3	23:07:51	select * from authors LIMIT 0, 1000
4	23:08:03	select * from books LIMIT 0, 1000

Message

Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH... 0.000 sec

0 row(s) affected 0.000 sec

3 row(s) returned 0.000 sec / 0.000 sec

3 row(s) returned 0.000 sec / 0.000 sec

Object Info Session

Humid Now

Search

11:08 PM
12/18/2025

SQLAdditions... SQL File 9* SQL File 10*

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Form Editor

Field Types

Context Help Snippets

The screenshot shows the MySQL Workbench interface. In the top-left, there's a sidebar with sections for Management, Instance, and Performance. The main area has a 'Query 1' tab open with a list of SQL queries related to library management. Below it is a 'Result Grid' showing data from the 'books' table. A vertical toolbar on the right contains icons for Result Grid, Form Editor, and Field Types. At the bottom, there's a taskbar with various application icons and system status indicators like battery level and signal strength.

CATEGORY DATABASE

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas Information

No object selected

categories 3

Output:

Action Output

#	Time	Action	Duration / Fetch
1	23:07:46	select * from books LIMIT 0, 1000	0.000 sec
2	23:07:49	use library_management	0.000 sec
3	23:07:51	select * from authors LIMIT 0, 1000	0.000 sec / 0.000 sec
4	23:08:03	select * from books LIMIT 0, 1000	0.000 sec / 0.000 sec
5	23:08:12	select * from categories LIMIT 0, 1000	0.000 sec / 0.000 sec

Message

Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH... 0.000 sec
0 row(s) affected
3 row(s) returned
3 row(s) returned
3 row(s) returned

Object Info Session

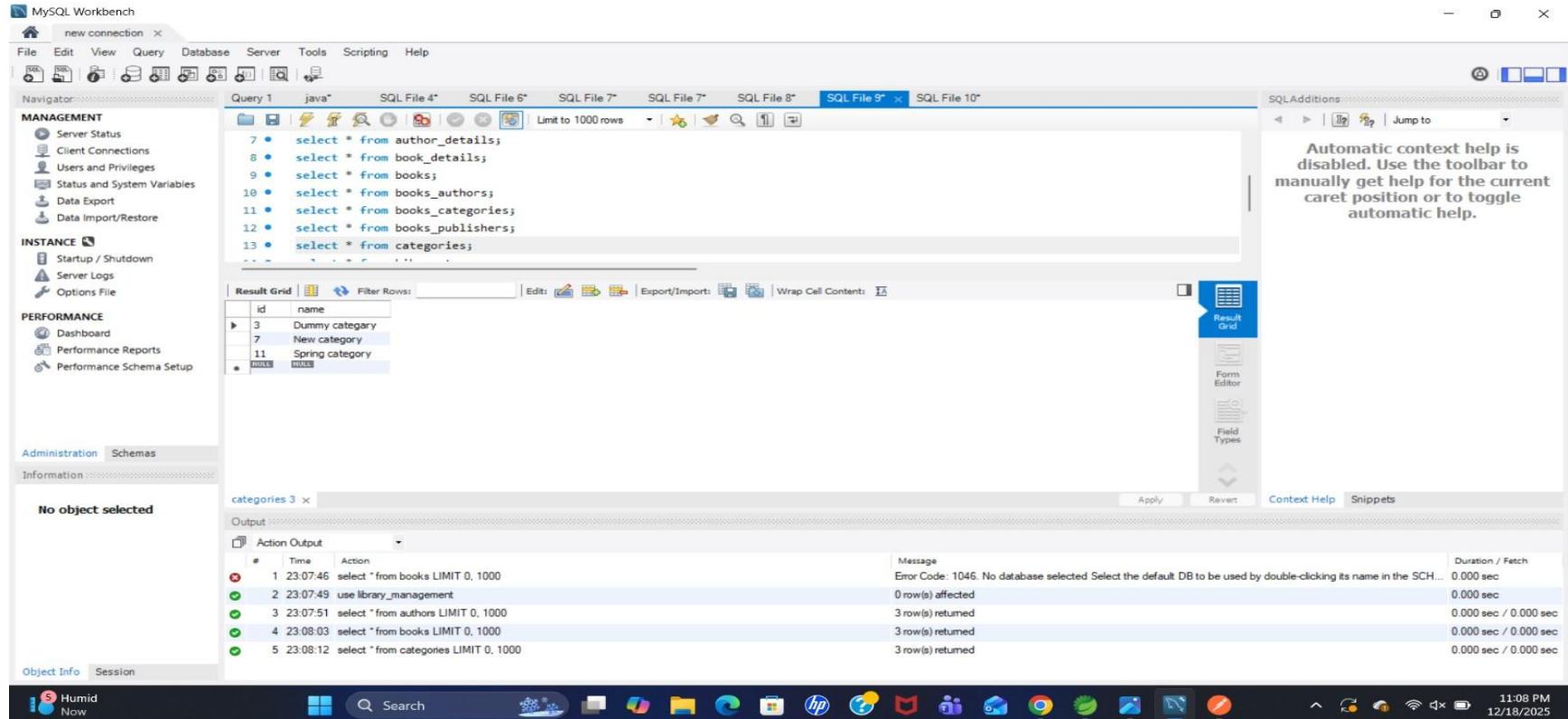
Humid Now

Search

11:08 PM 12/18/2025

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.



PUBLISHER DATABASE

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	23:07:46	select * from books LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
2	23:07:49	use library_management	0 row(s) affected	0.000 sec
3	23:07:51	select * from authors LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
4	23:08:03	select * from books LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
5	23:08:12	select * from categories LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
6	23:08:18	select * from publishers LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Humid Now

Search

11:08 PM
12/18/2025

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

POST METHOD FOR ADD DATA

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', 'Search Postman', 'Ctrl + K' (shortcuts), and various status icons like 'Invite', 'Settings', 'Bell', and 'Upgrade'. The left sidebar displays the user's workspace ('triveni garimella9990's Workspace'), collection ('My Collection'), environment ('Environments'), history ('History'), flows ('Flows'), and files ('Files BETA'). The main workspace shows a 'Post data' collection with a 'POST Get data' and a 'POST Post data' endpoint. The 'POST Post data' endpoint is selected, with its URL set to 'http://localhost:9080/category/2'. The 'Params' tab is active, showing a single parameter 'Key' with 'Value' and 'Description' columns. Below this, the 'Body' tab is selected, displaying the response body as an HTML snippet. The response status is '200 OK' with a duration of '16 ms' and a size of '2.57 KB'. The response body contains the following HTML code:

```
64 | <div class="form-group">
65 |   <div class="row">
66 |     <div class="col-sm-6 col-sm-offset-3">
67 |       <input type="submit" name="login-submit" id="login-submit"
68 |         class="form-control btn btn-success" value="Log In" />
69 |
70 |     </div>
71 |   </div>
72 | </div>
73 |
74 | </div>
75 | </div>
76 | </div>
77 | <div><footer class="page-footer font-small blue pt-4">
78 |   <!-- Copyright -->
79 |   <div class="footer-copyright text-center py-3">
80 |     © 2022 Copyright: <a href="https://www.knowledgefactory.net/">
81 |       www.knowledgefactory.net</a>
```

At the bottom, there are navigation links for 'Cloud View', 'Find and replace', 'Console', 'Terminal', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and system status indicators for battery, signal, and date.

GET METHOD FOR GET ALL DATA

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Collections, Environments, History, Flows, and Files (BETA). The main workspace is titled "triveni garimella9990's Workspace". A collection named "My Collection" is selected, containing two items: "GET Get data" and "POST Post data". The current view is a POST request for "Post data" with the URL "http://localhost:9080/books". The "Params" tab is active, showing a table with one row: "Key" (Value) and "Value" (Description). Below this, the "Body" tab displays the response body as an HTML document. The response is a 200 OK status with 375 ms duration and 2.57 KB size. The response body is as follows:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <meta charset="utf-8">
6 <meta http-equiv="x-ua-compatible" content="ie=edge">
7 <title>Login</title>
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <link rel="stylesheet"
10 href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
11 <link rel="stylesheet"
12 href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
13 </head>
14
15 <body>
16 <div><nav class="navbar navbar-expand-sm bg-info navbar-light">
17 <a class="navbar-brand" href="#">Library Management System</a>
```

At the bottom, the footer contains links for Cloud View, Find and replace, Console, Terminal, Runner, Start Proxy, Cookies, Vault, Trash, and system icons for search, file operations, and connectivity.

PUT METHOD FOR UPDATE DATA

The screenshot shows the Postman application interface. The top navigation bar includes Home, Workspaces, API Network, a search bar, and various toolbars. The left sidebar displays 'triveni garimella9990's Workspace' with sections for Collections, Environments, History, and Flows. The main workspace shows a collection named 'My Collection' with a 'Post data' request selected. The request details show a PUT method at the URL `http://localhost:9080/update/2`. The 'Params' tab is active, showing a table for Query Params with one row: Key (Value) and Value (Key). Below this, the 'Body' tab displays an HTML form structure:

```
44<div class="form-group">
45  <label for="username">Email</label>: <input type="text" id="username" name="username" placeholder="Enter Email" required="required" />
46</div>
47<div class="form-group">
48  <label for="password">Password</label>: <input type="password" id="password" name="password" class="form-control" placeholder="Enter Password" />
49</div>
50<div class="form-group">
51  <div class="row">
52    <div class="col-sm-6 col-sm-offset-3">
53      <input type="submit" name="login-submit" id="login-submit" class="form-control btn btn-success" value="Log In" />
54    </div>
55  </div>
56</div>
```

The response section shows a successful 200 OK status with a response time of 15 ms and a size of 2.57 KB. The bottom navigation bar includes Cloud View, Find and replace, Console, Terminal, Runner, Start Proxy, Cookies, Vault, Trash, and a timestamp of 10:58 PM on 12/18/2025.

STS Configuration

workspace-spring-tools-for-eclipse-4.32.2.RELEASE - Ibm/src/main/java/com/verinon/lbm/jpa/BookRepository.java - Spring Tools for Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

- > eshop [boot]
- > Ibm [boot]
- librarymanagementsystem [boot] [devtools]
 - > src/main/java
 - > src/main/resources
 - > src/test/java
 - > JRE System Library [JavaSE-17]
 - > Maven Dependencies
 - > src
 - > target
 - LICENSE
 - pom.xml
 - README.md

BookRepository.java

```
1 package com.verinon.lbm.jpa;
2
3 import com.verinon.lbm.pojo.BookPojo;
4
5
6 public interface BookRepository extends JpaRepository<BookPojo, Integer>
7 {
8     //List<BookPojo> findByBook_name(String book_name);
9 }
10
11
```

Outline X Task List

- com.verinon.lbm.jpa
- BookRepository

Console X

```
librarymanagementsystem - Application [Spring Boot App] C:\Program Files\Java\jdk-17.0.5\bin\javaw.exe (Dec 18, 2025, 10:32:38 PM elapsed: 0:30:52) [pid: 22424]
2025-12-18 22:32:41.882 INFO 22424 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitIrr
2025-12-18 22:32:41.927 INFO 22424 --- [ restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version : HCANN000001: Hibernate Commons Annotations
2025-12-18 22:32:42.081 INFO 22424 --- [ restartedMain] o.hibernate.annotations.common.Version : HHH000400: Using dialect: org.hibernate.dialect
2025-12-18 22:32:42.185 INFO 22424 --- [ restartedMain] org.hibernate.dialect.Dialect : HHH000490: Using JtaPlatform implementation
2025-12-18 22:32:42.786 INFO 22424 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : Initialized JPA EntityManagerFactory factory for persistence unit 'librarymanagementsystem'
2025-12-18 22:32:42.795 INFO 22424 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : spring.jpa.open-in-view is enabled by c
2025-12-18 22:32:42.848 WARN 22424 --- [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : Will not secure any request
2025-12-18 22:32:43.398 INFO 22424 --- [ restartedMain] o.s.s.web.DefaultSecurityFilterChain : LiveReload server is running on port 35
2025-12-18 22:32:43.718 INFO 22424 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : Tomcat started on port(s): 9080 (http)
2025-12-18 22:32:43.773 INFO 22424 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Started Application in 3.919 seconds (-)
2025-12-18 22:32:43.783 INFO 22424 --- [ restartedMain] c.k.d.l.Application : HikariPool-1 - Thread starvation or clos
2025-12-18 22:52:27.071 WARN 22424 --- [1-1 housekeeper] com.zaxxer.hikari.pool.HikariPool : Initializing Spring DispatcherServlet
2025-12-18 22:56:27.670 INFO 22424 --- [nio-9080-exec-3] o.a.c.c.C.[localhost].[/] : Tomcat started on port(s): 9080 (http)
2025-12-18 22:56:27.670 INFO 22424 --- [nio-9080-exec-3] o.s.web.servlet.DispatcherServlet : Started Application in 3.919 seconds (-)
2025-12-18 22:56:27.670 INFO 22424 --- [nio-9080-exec-3] o.s.web.servlet.DispatcherServlet : HikariPool-1 - Thread starvation or clos
2025-12-18 22:56:27.672 INFO 22424 --- [nio-9080-exec-3] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
2025-12-18 23:03:20.999 WARN 22424 --- [1-1 housekeeper] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Thread starvation or clos
```

librarymanagementsystem

Air Poor Now

Search

11:03 PM
12/18/2025