



UE21CS352B - Object Oriented Analysis & Design using Java

Mini Project Report

“Library Management System”

Submitted by:

| | |
|---------------------------|----------------------|
| Suhas Gowda Harish | PES1UG21CS633 |
| Suma Grandhi | PES1UG21CS638 |
| Sunil Parameswaran | PES1UG21CS644 |
| Tanvi Jain | PES1UG21CS664 |

6th Semester k Section

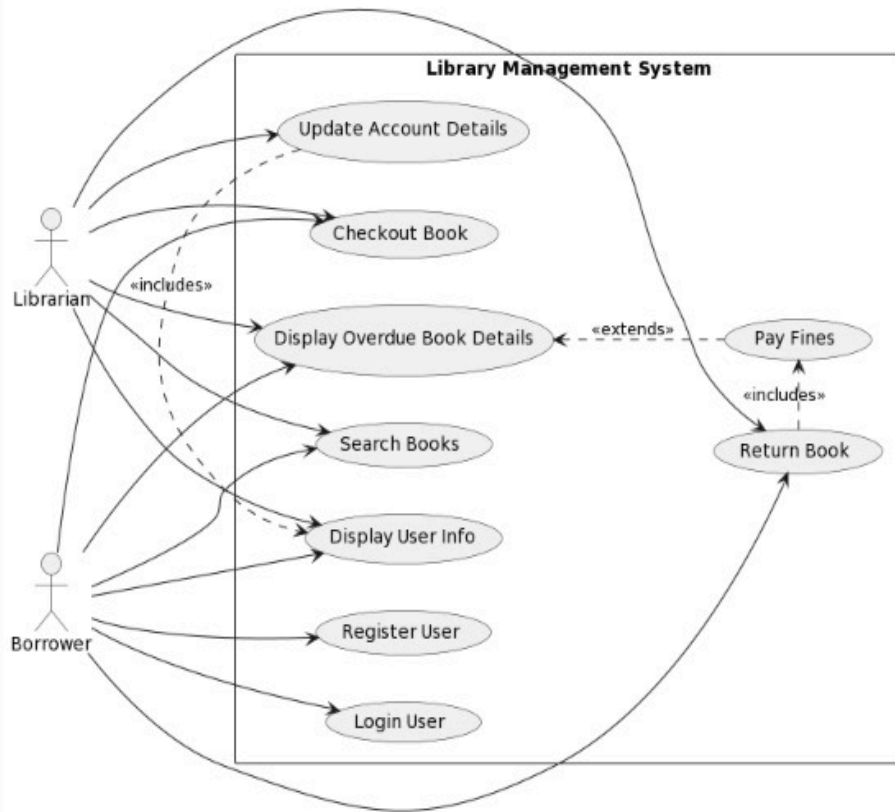
Bhargavi Mokashi
Assistant Professor

January - May 2024

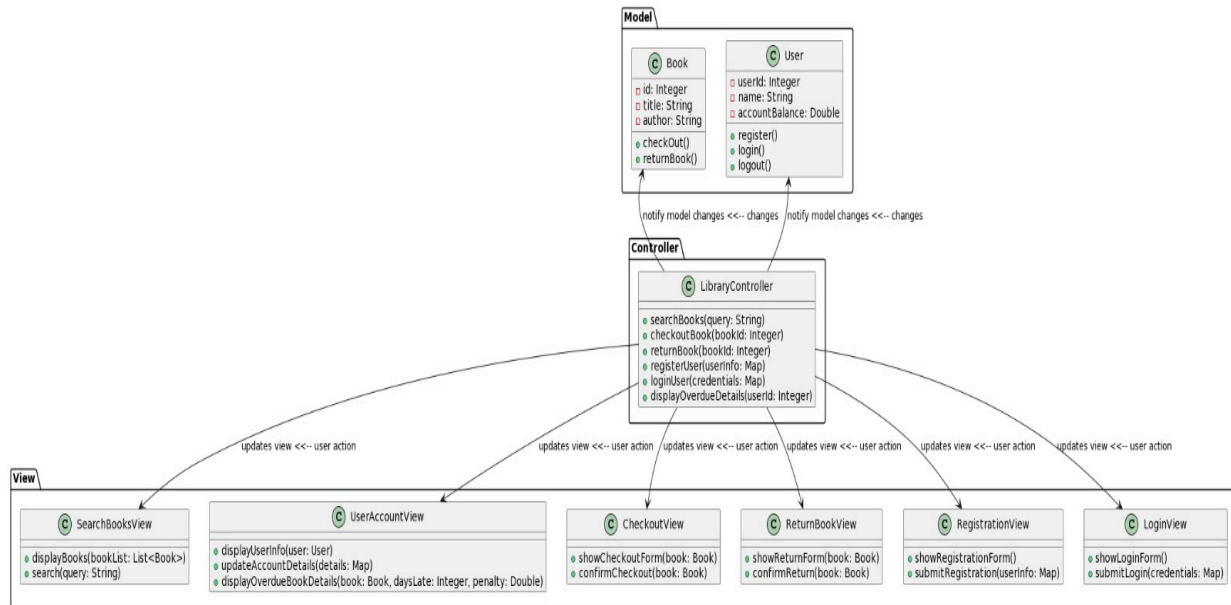
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

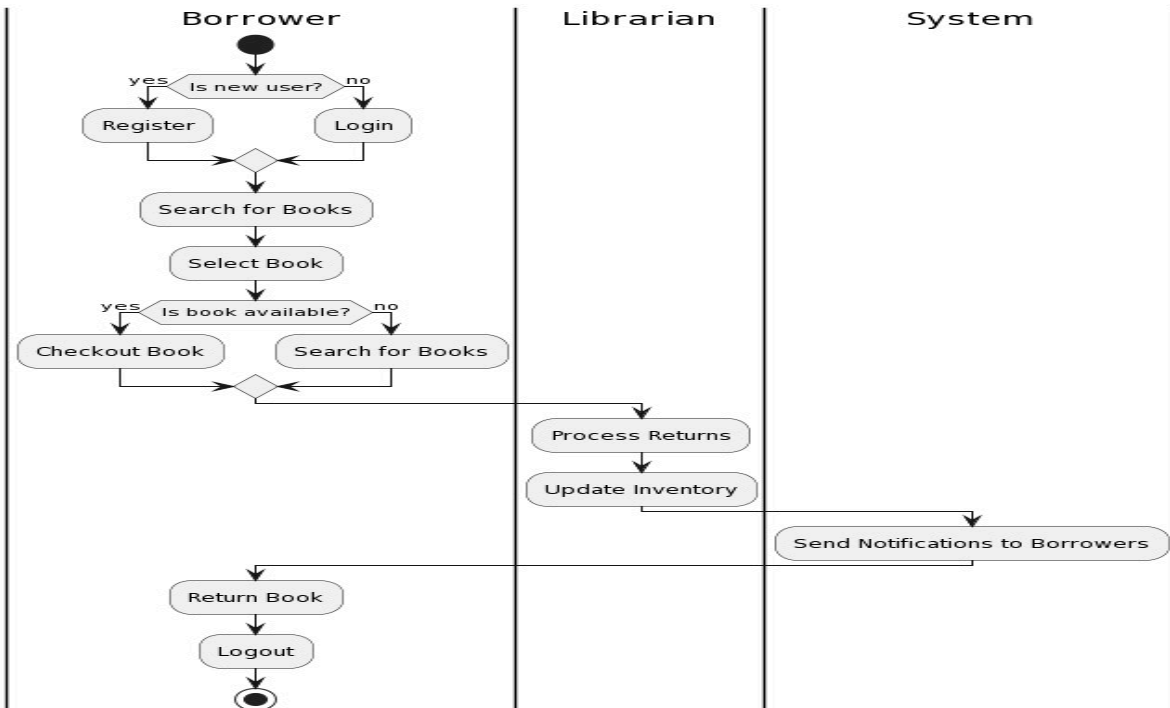
USE CASE DIAGRAM:



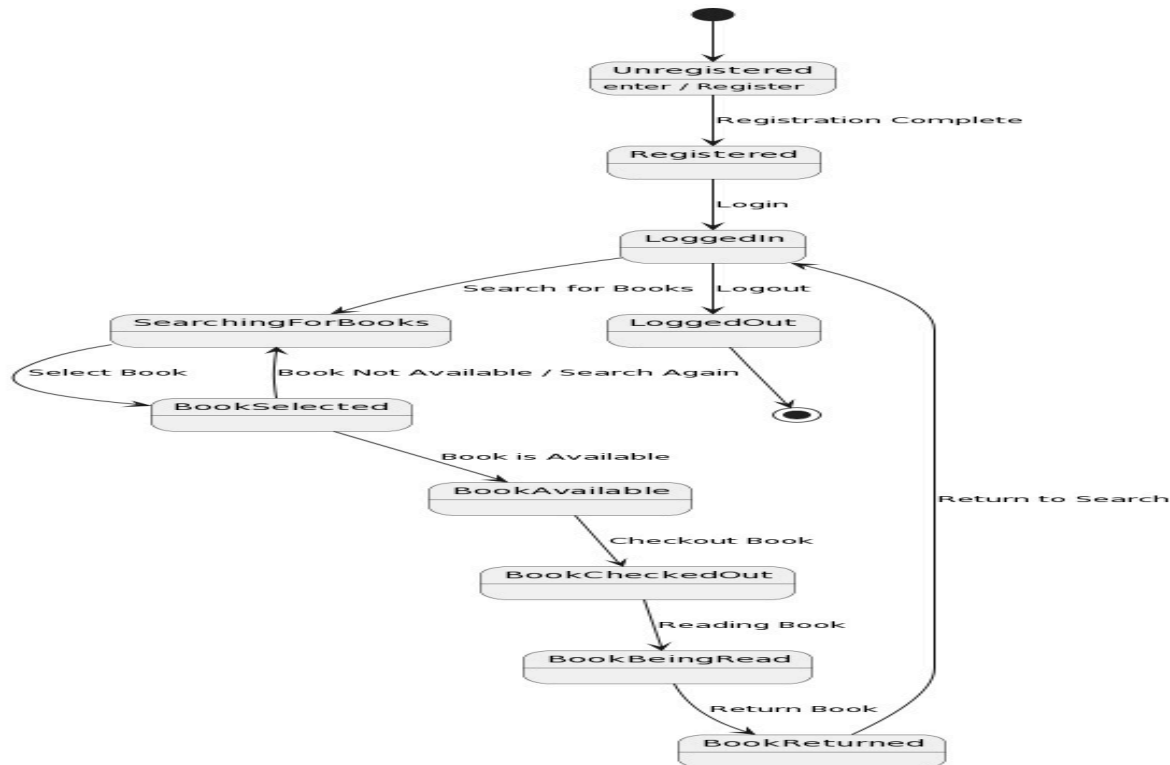
CLASS DIAGRAM:



ACTIVITY DIAGRAM:



STATE DIAGRAM:



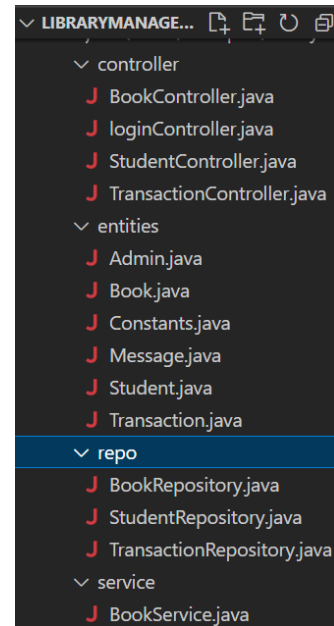
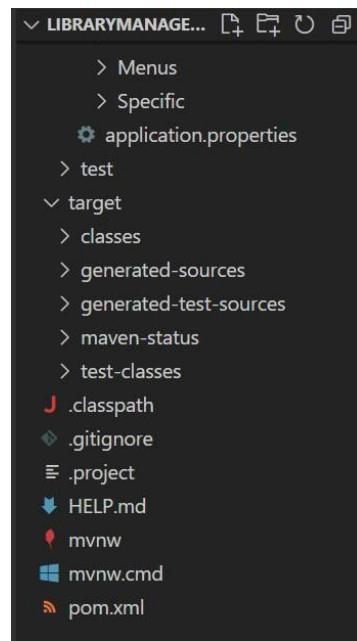
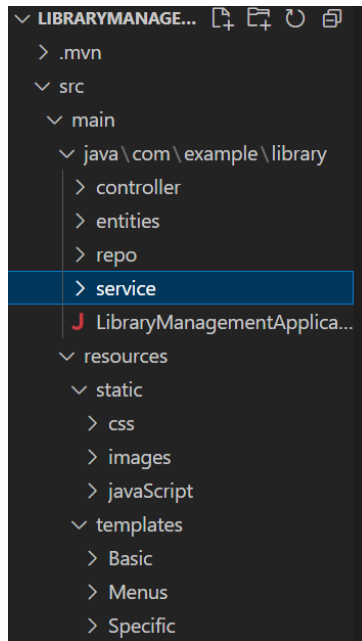
MODEL – VIEW – CONTROLLER ARCHITECTURE:

The Model View Controller (MVC) design pattern specifies that an application consists of a data model, presentation information, and control information. The pattern requires that each of these be separated into different objects. MVC is more of an architectural pattern, but not for complete application. MVC mostly relates to the UI / interaction layer of an application. You're still going to need business logic layer, maybe some service layer and data access layer.

The **Model** contains only the pure application data, it contains no logic describing how to present the data to a user. (It's just a data that is shipped across the application like for example from back-end server view and from front-end view to the database. In java programming, Model can be represented by the use of POJO (Plain-old-java-object) which is a simple java class.

The **View** presents the model's data to the user. The view knows how to access the model's data, but it does not know what this data means or what the user can do to manipulate it. View just represent, displays the application's data on screen. View page are generally in the format of .html or .jsp in java programming (which is flexible).

The **Controller** exists between the view and the model. It is where the actual business logic is written. It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model. Since the view and the model are connected through a notification mechanism, the result of this action is then automatically reflected in the view.



DESIGN PRINCIPLES:

The following principles have been used in our project:

S - Single Responsibility Principle:

This principle has been used in the creation of Admin functionality and Books functionalities.

O - Open Closed Principle:

This principle is used in the implementation of specifying features of the controller part of our project (BookController.java, loginController.java, StudentController.java, TransactionController.java)

I - Interface Segregation:

This principle has been used to segregate the interface functionalities of different use cases in our project.

D - Dependency Inversion:

This principle has been used to implement to remove complexity of high level modules, that do not depend the lower level modules.

Design Patterns:

1. Singleton Design Pattern:

Singleton pattern is one of the simplest design patterns in Java. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object. This pattern involves a single class which is responsible to create an object while making sure that only single object gets created. This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class. In our project all the functionalities of the Book implantation like adding a book, issuing a book, tracking a book, fine payment have all been implementing by using this design pattern.

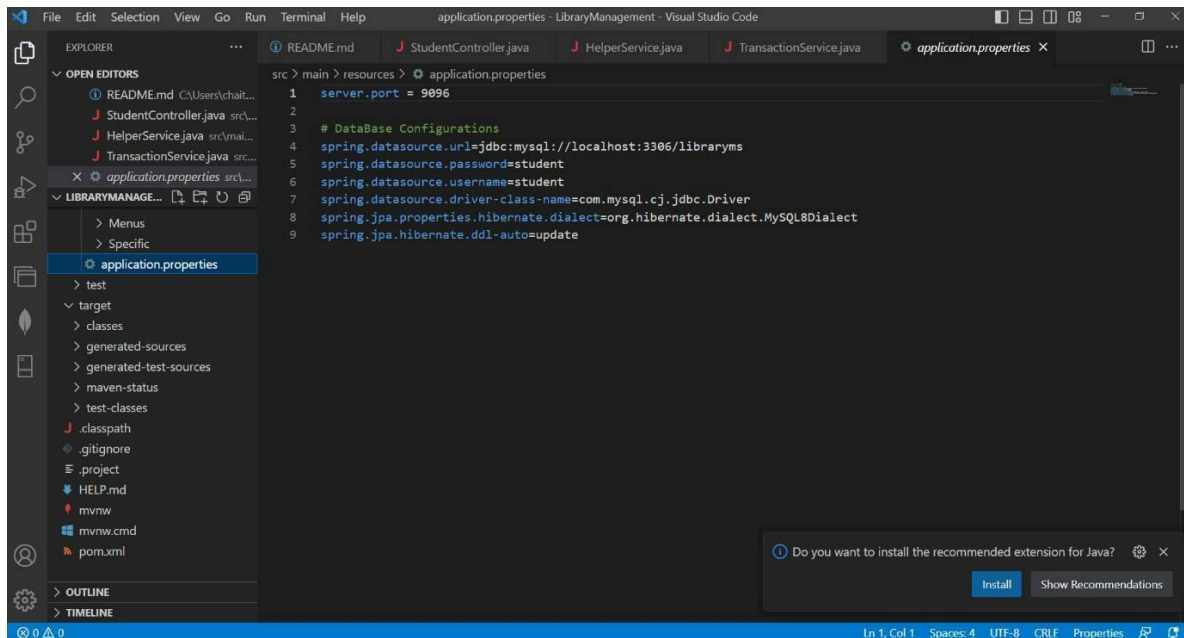
2. Service Layer Pattern:

Service Layer is an abstraction over domain logic. It defines application's boundary with a layer of services that establishes a set of available operations and coordinates the application's response in each operation. In our project this patter is defined in the classifying the services that a user needs, hence mapping the each service to a particular user request.

Implemented in (StudentService.java, TransactionService.java, BookService.java, HelpService.java)

Database:

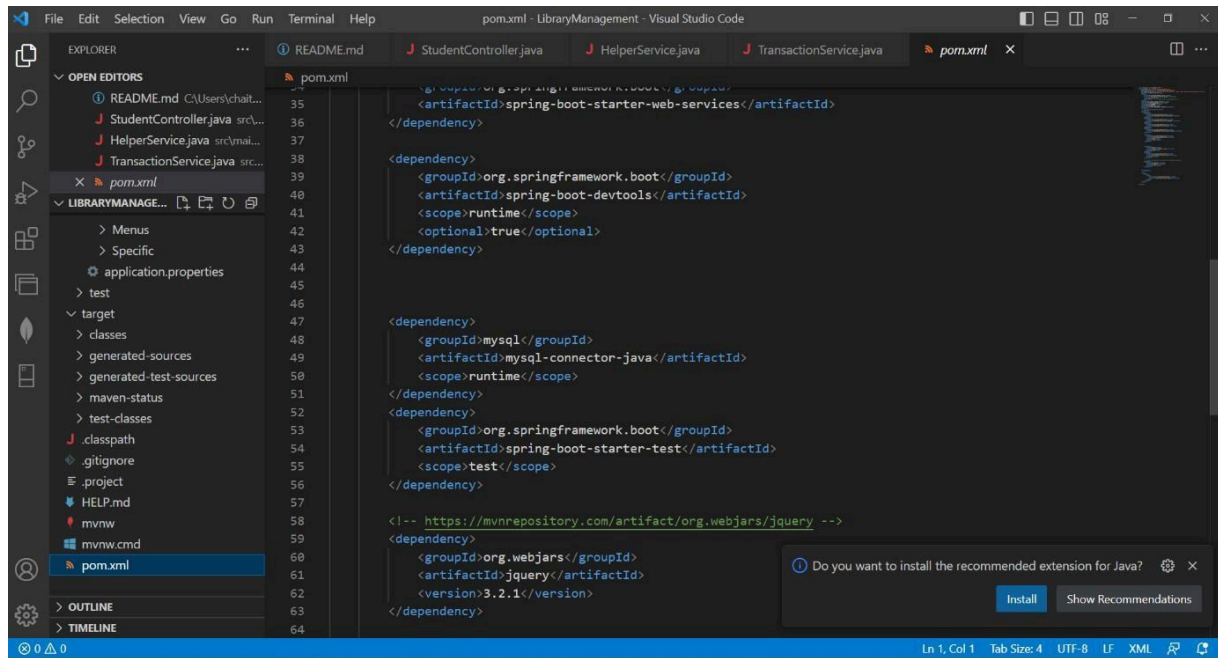
We have used MySQL Database, to make our project data persistent, along with the implementation of Spring MVC Framework.



The screenshot shows the Visual Studio Code interface with the 'application.properties' file open in the editor. The file is located in the 'src > main > resources' directory. The code in the file is as follows:

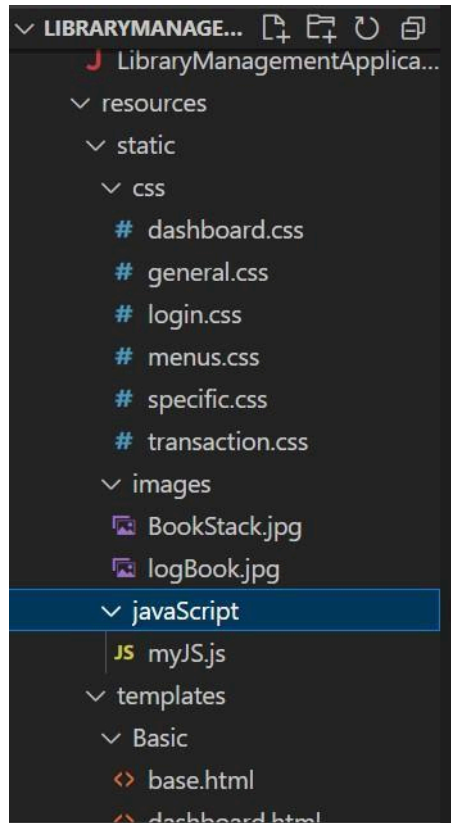
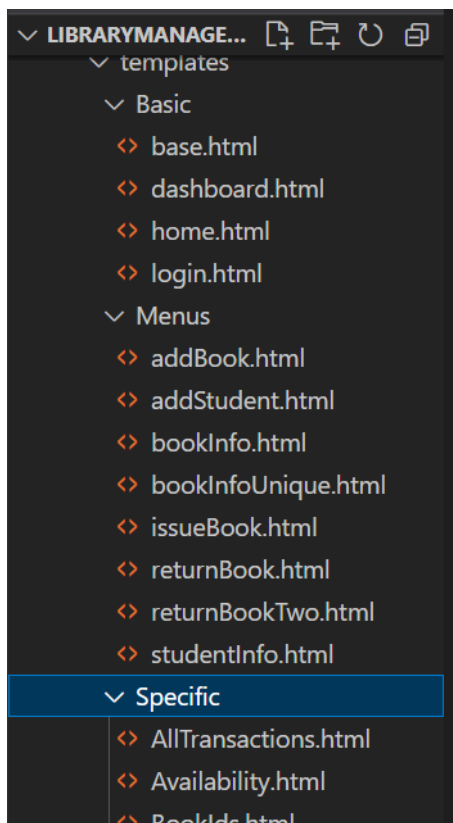
```
1 server.port = 9096
2
3 # DataBase Configurations
4 spring.datasource.url=jdbc:mysql://localhost:3306/libraryms
5 spring.datasource.password=student
6 spring.datasource.username=student
7 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
8 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
9 spring.jpa.hibernate.ddl-auto=update
```

The Explorer sidebar on the left shows the project structure, including the 'application.properties' file. A notification at the bottom right asks if the user wants to install the recommended extension for Java.



Front End:

HTML and CSS have been used to implement a user-friendly front end.



Project Screenshots:

Starting MCV using Spring Maven

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

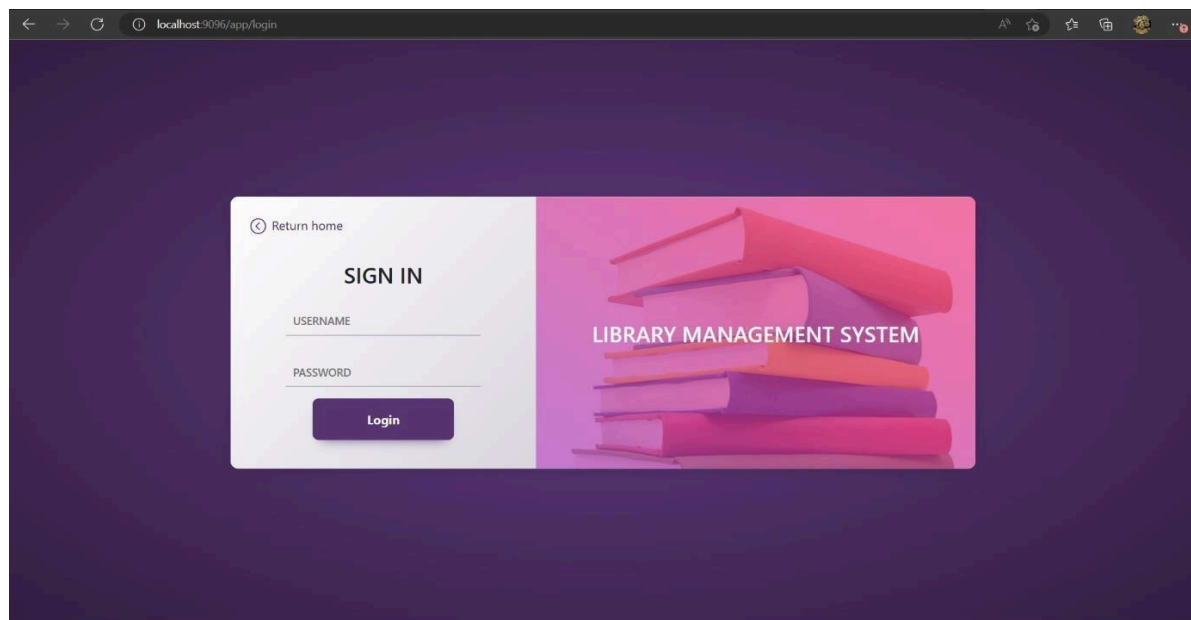
C:\Users\chait\Desktop\LibraryManagement (2)\LibraryManagement>mvn spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example.manage.library:LibraryManagement >-----
[INFO] Building LibraryManagement 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> spring-boot-maven-plugin:2.5.5:run (default-cli) > test-compile @ LibraryManagement >>>
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:resources (default-resources) @ LibraryManagement ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] Copying 1 resource
[INFO] Copying 34 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ LibraryManagement ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @ LibraryManagement ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] skip non existing resourceDirectory C:\Users\chait\Desktop\LibraryManagement (2)\LibraryManagement\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ LibraryManagement ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] <<< spring-boot-maven-plugin:2.5.5:run (default-cli) < test-compile @ LibraryManagement <<<
[INFO]
[INFO] --- spring-boot-maven-plugin:2.5.5:run (default-cli) @ LibraryManagement ---
[INFO] Attaching agents: []

  ____ _
 / ___| | |
 \___ \| |_| |
  ___) | | |
 / ___| | |
 \___|_|_|_|

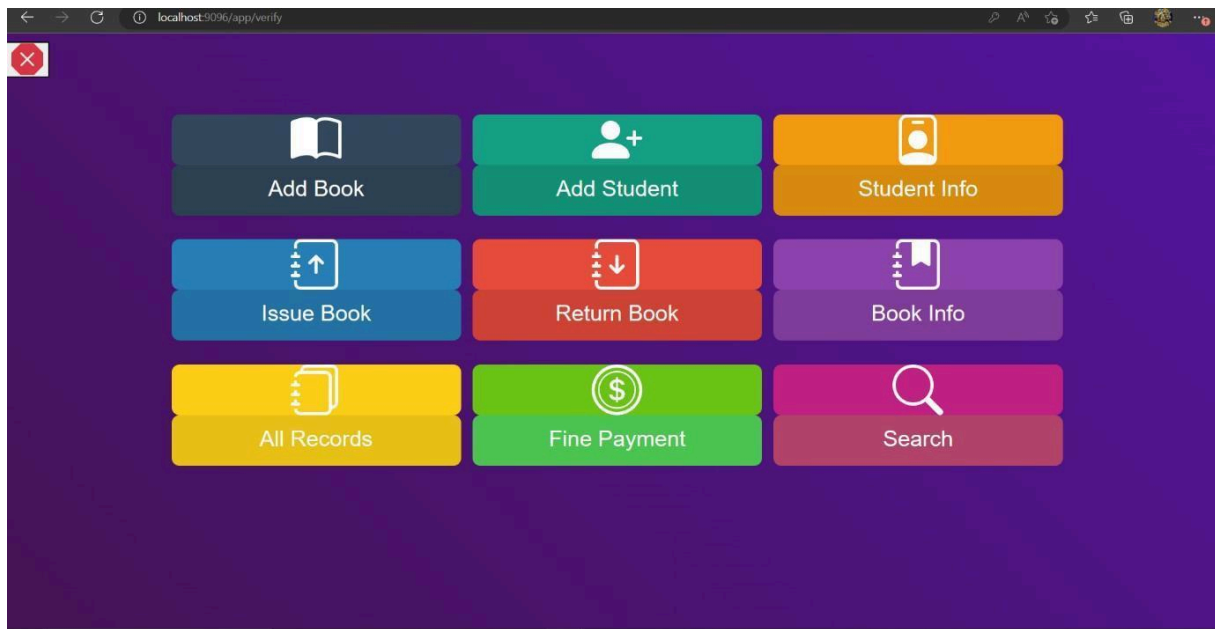
:: Spring Boot ::
              (v2.5.5)

2023-04-29 20:20:04.668 INFO 18752 --- [ restartedMain] c.e.l.LibraryManagementApplication : Starting LibraryManagementApplication using Java 19.0.2 on DESKTOP-5073HML with PID 18752 (C:\Users\chait\Desktop\LibraryManagement (2)\LibraryManagement\target\classes started by chait in C:\Users\chait\Desktop\LibraryManagement (2)\LibraryManagement)
2023-04-29 20:20:04.668 INFO 18752 --- [ restartedMain] c.e.l.LibraryManagementApplication : No active profile set, falling back to default profiles: default
2023-04-29 20:20:04.716 INFO 18752 --- [ restartedMain] .e.DevtoolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable.
2023-04-29 20:20:04.716 INFO 18752 --- [ restartedMain] .e.DevtoolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'.
2023-04-29 20:20:05.383 INFO 18752 --- [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
```

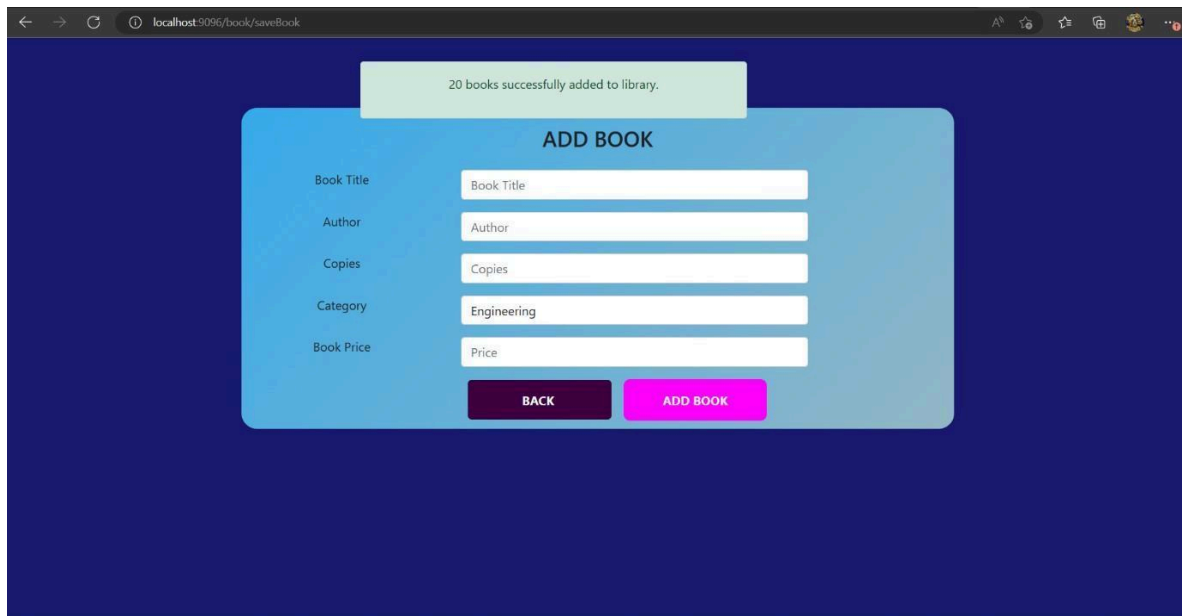
Login Page



Admin Dashboard













Adding Books



Book Info:

Back

| Sr No | Title | Author | Category | Price | Details |
|-------|---|----------------|-------------------|-------|---|
| 1 | Do Androids Dream of Electric Sheep? | Philip K. Dick | Others | 690 |   |
| 2 | The Curious Incident of the Dog in the Night-Time | Mark Haddon | Others | 590 |   |
| 3 | I Was Told There'd Be Cake: Essays | Sloane Crosley | General Knowledge | 560 |   |
| 4 | Design Principles and Pattern - A Java Approach | Bhargav | Engineering | 1000 |   |
| 5 | JAVA DESIGN PATTERNS | JOSEPH MOSH | Engineering | 250 |   |

ADD STUDENT

Student successfully registered

ADD STUDENT

First Name

Last Name

Branch

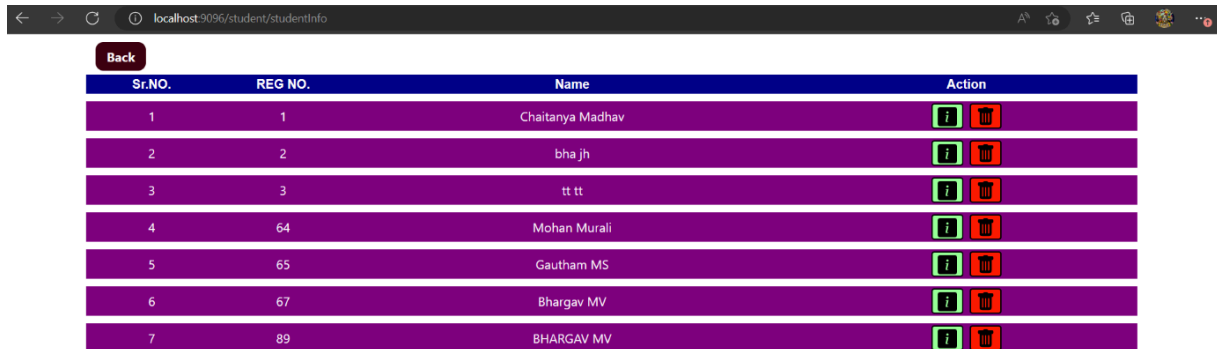
Year















Division

Roll Number

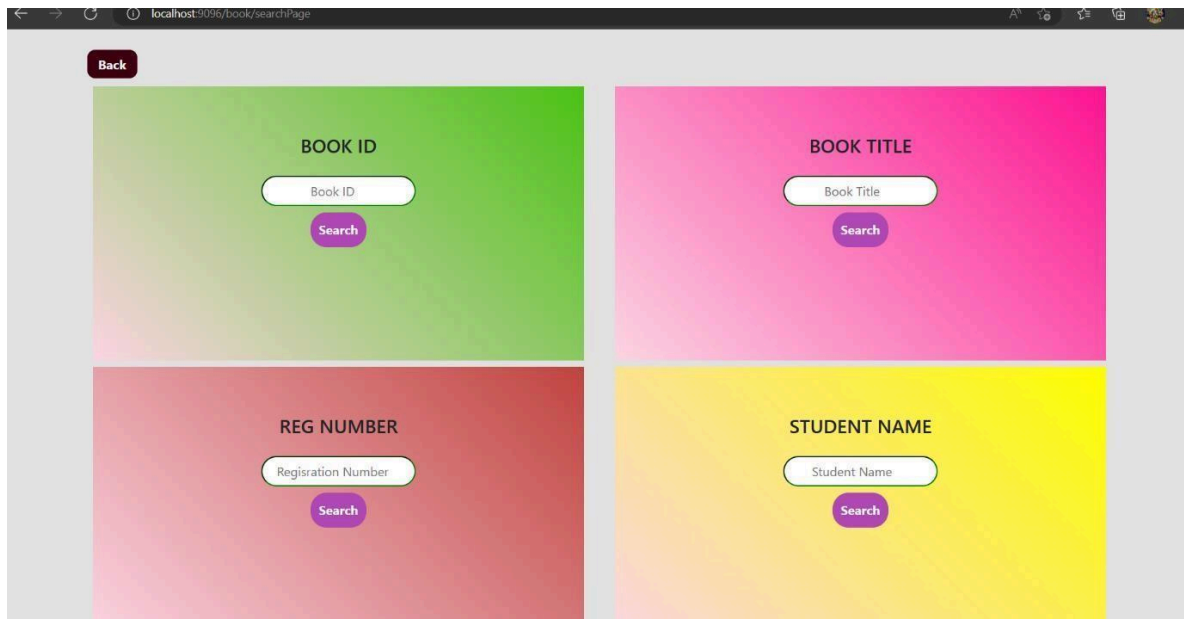
BACK ADD STUDENT

STUDENT INFO



| Sr.NO. | REG NO. | Name | Action |
|--------|---------|------------------|---|
| 1 | 1 | Chaitanya Madhav |   |
| 2 | 2 | bha jh |   |
| 3 | 3 | tt tt |   |
| 4 | 64 | Mohan Murali |   |
| 5 | 65 | Gautham MS |   |
| 6 | 67 | Bhargav MV |   |
| 7 | 89 | BHARGAV MV |   |

Searching a book from User Account:



Back

BOOK ID

Search

BOOK TITLE

Search

REG NUMBER

Search

STUDENT NAME

Search

Back End MySQL:

```
C:\Windows\System32\cmd.exe - mysql -u root
```

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]> show databases;
```

```
-----+-----+
Database
-----+-----+
crime634
crime_portal
criminal_management
crms_db
cs634
cs634_rns
cs634_university_db
dbt_peslug20cs634_crimereport
demo_crime
demo_db
gabbbbbage
information_schema
libraryms
mysql
performance_schema
peslug20cs634_crimedb
phpmyadmin
railway_system_lab_634
test
university_db_cs634
-----+-----+
0 rows in set (0.010 sec)
```

```
MariaDB [(none)]> use libraryms;
```

```
Database changed
```

```
MariaDB [libraryms]> show tables;
```

```
-----+-----+
Tables_in_libraryms
-----+-----+
book
hibernate_sequence
student
student_books
transaction
-----+-----+
0 rows in set (0.001 sec)
```

```
MariaDB [libraryms]>
```

```
C:\Windows\System32\cmd.exe - mysql -u root
```

```
MariaDB [libraryms]> select * from student;
```

```
-> ;
```

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near ':' at line 1
```

```
MariaDB [libraryms]> select * from student;
```

```
-----+-----+-----+-----+-----+-----+-----+-----+
reg_number | branch | division | fine | fine_paid | name | roll_number | surname | year |
-----+-----+-----+-----+-----+-----+-----+-----+
1 | Computer Science | C | 0 | 3134 | Chaitanya | 78 | Radhav | Third |
2 | Mechanical | A | 0 | bha | 680 | Jh | First |
3 | Mechanical | A | 0 | tt | 9 | tt | First |
64 | Mechanical | C | 0 | 0 | Mohan | 278 | Murali | Third |
65 | Computer Science | C | 0 | 0 | Gautham | 222 | MS | Third |
67 | EATC | B | 0 | 0 | Bhargav | 333 | MV | First |
89 | Computer Science | A | 0 | 0 | BHARGAV | 234 | MV | Third |
-----+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)
```

```
MariaDB [libraryms]> select * from book;
```

```
-----+-----+-----+-----+-----+-----+-----+-----+
book_id | author | category | is_issued | price | title | student_reg_number |
-----+-----+-----+-----+-----+-----+-----+-----+
4 | Philip K. Dick | Others | 0 | 690 | Do Androids Dream of Electric Sheep? | NULL |
5 | Philip K. Dick | Others | 0 | 690 | Do Androids Dream of Electric Sheep? | NULL |
6 | Philip K. Dick | Others | 0 | 690 | Do Androids Dream of Electric Sheep? | NULL |
7 | Philip K. Dick | Others | 0 | 690 | Do Androids Dream of Electric Sheep? | NULL |
8 | Philip K. Dick | Others | 0 | 690 | Do Androids Dream of Electric Sheep? | NULL |
9 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
10 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
11 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
12 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
13 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
14 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
15 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
16 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
17 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
18 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
19 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
20 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
21 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
22 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
23 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
24 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
25 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
26 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
27 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
28 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
29 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
30 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
31 | Mark Haddon | Others | 0 | 590 | The Curious Incident of the Dog in the Night-Time | NULL |
-----+-----+-----+-----+-----+-----+-----+-----+
0 rows in set (0.001 sec)
```

Issue Book:

localhost:9096/transaction/issueRequest

Book is Issued to Student

ISSUE BOOK

Registration Number:

Book ID:

[BACK](#) [ISSUE BOOK](#)

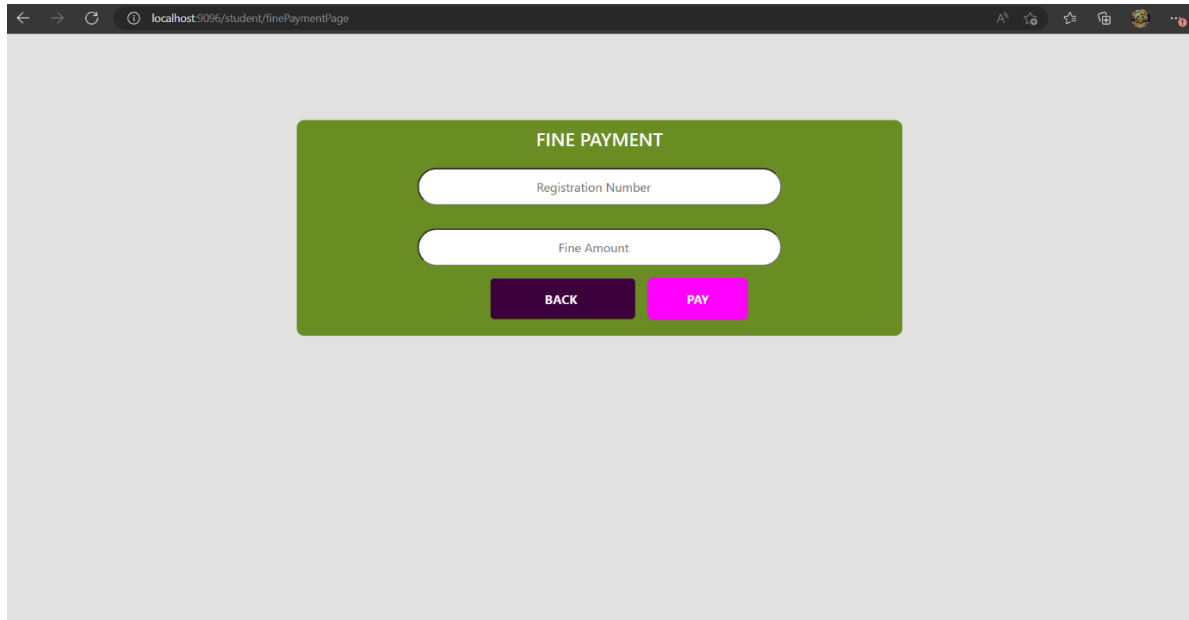
ALL Transactions List:

localhost:9096/transaction/allTransactions

[Back](#)

| All Transaction | | | |
|-----------------|---|---------------|------------------|
| BOOK ID | TITLE | REG NUMBER | STUDENT NAME |
| ISSUE DATE | EXPECTED RETURN | ACTUAL RETURN | FINE |
| 70 | JAVA DESIGN PATTERNS | 89 | BHARGAV MV |
| 2023-04-29 | 2023-05-14 | 2023-04-29 | Rs. 0 |
| 27 | The Curious Incident of the Dog in the Night-Time | 67 | Bhargav MV |
| 2023-04-28 | 2023-05-13 | | Rs. 0 |
| 49 | Design Principles and Pattern - A Java Approach | 1 | Chaitanya Madhav |
| 2023-04-28 | 2023-05-13 | | Rs. 0 |

Fine Payment:



The screenshot shows a web browser window with the address bar displaying 'localhost:9096/student/finePaymentPage'. The page content is a green rectangular form titled 'FINE PAYMENT'. Inside the form, there are two white input fields: the first is labeled 'Registration Number' and the second is labeled 'Fine Amount'. Below these fields are two buttons: a dark purple 'BACK' button and a bright pink 'PAY' button.

← → ↻ ⓘ localhost:9096/student/finePaymentPage

FINE PAYMENT

Registration Number

Fine Amount

BACK PAY