

COLLEGE MANAGEMENT SYSTEM

Project Member:

Ms. Neha Gupta

Mr. Gopal Singh Devada

Mr. Suresh

Mr. Shashank

Mr. Pradeep

Mr. Harish Kumar

Guided By :

Prof. Indrakka Mali

Title :

- Introduction
- Objective
- Technology to be Used
- Proposed System
- Snapshots
- Diagrams
- Advantages
- Conclusion

Introductoin:

- The college management system helps Educational Institutions especially colleges in various ways, such as storing data, maintain student profiles, analyzing administrative and academic data, improving communication, and engaging students.
- A college management system is **a cloud based educational ERP software that enables HEIs to manage online admission & fees, students' attendance, library books, etc.** It can also generate students' performance reports & simplify the hassles of faculty.

- College Management System is an end-to-end solution for colleges to improve operational efficiency & institutional outcomes by automating Student-Faculty lifecycle & campus administration.
- The COLLEGE MANAGEMENT SYSTEM can be used to store student information like attendance and student details etc.
- Admin can create report regarding any student any time using this system. Using this system you can register new student and their course details.

Objective :

- The main objective of the College Management System is **to manage the details of College, Faculty, Course, Batch, Session.**
- It manages all the information about College, Student, Session, College.
- The project is totally built at administrative end and thus only the administrator is guaranteed the access

- **This project deals with the various functioning in College management process.**
- The main idea is to implement a proper process to system .
- In our existing system contains a many operations registration, student search, fees, details of the student etc.
- The organization can maintain computerized records without redundant entries.

Software & Hardware used:

Software Requirements :

- Operating System : Windows 7 or Windows 10
- Language : Java
- IDE : Spring Boot, PostMan
- Backend : Microsoft MySQL server

Hardware Requirements :

- CPU : Intel Pentium IV processor
- RAM : 512 MB or above
- Hard Disk : 2 GB hard disk space or minimum

Proposed System:

- The college management system helps Educational Institutions especially colleges in various ways, such as storing data, maintain student profiles, analyzing administrative and academic data, improving communication, and engaging students.
- **This is a web-related application that permits us to approach the entire knowledge regarding the college, students, faculties etc.**
- This application is also called as College management system. It offers an actual trip of the college campus.

- First we have to run the Application on the browser, the index page will be shown after that if the user is registered then he/she can login the page and he/she can entering in the application.
- If the user is not registered then he/she needs to register themselves and then he/she can continuing to login and he/she can access any time.
- After that related pages will be there like if user is admin so he/she can access all the operations like insert, delete, update and search the details.

- If the user is customer then he/she can view the related record and he/she can update the record.
- And after that he/she can logged out by themselves.

Back-End :

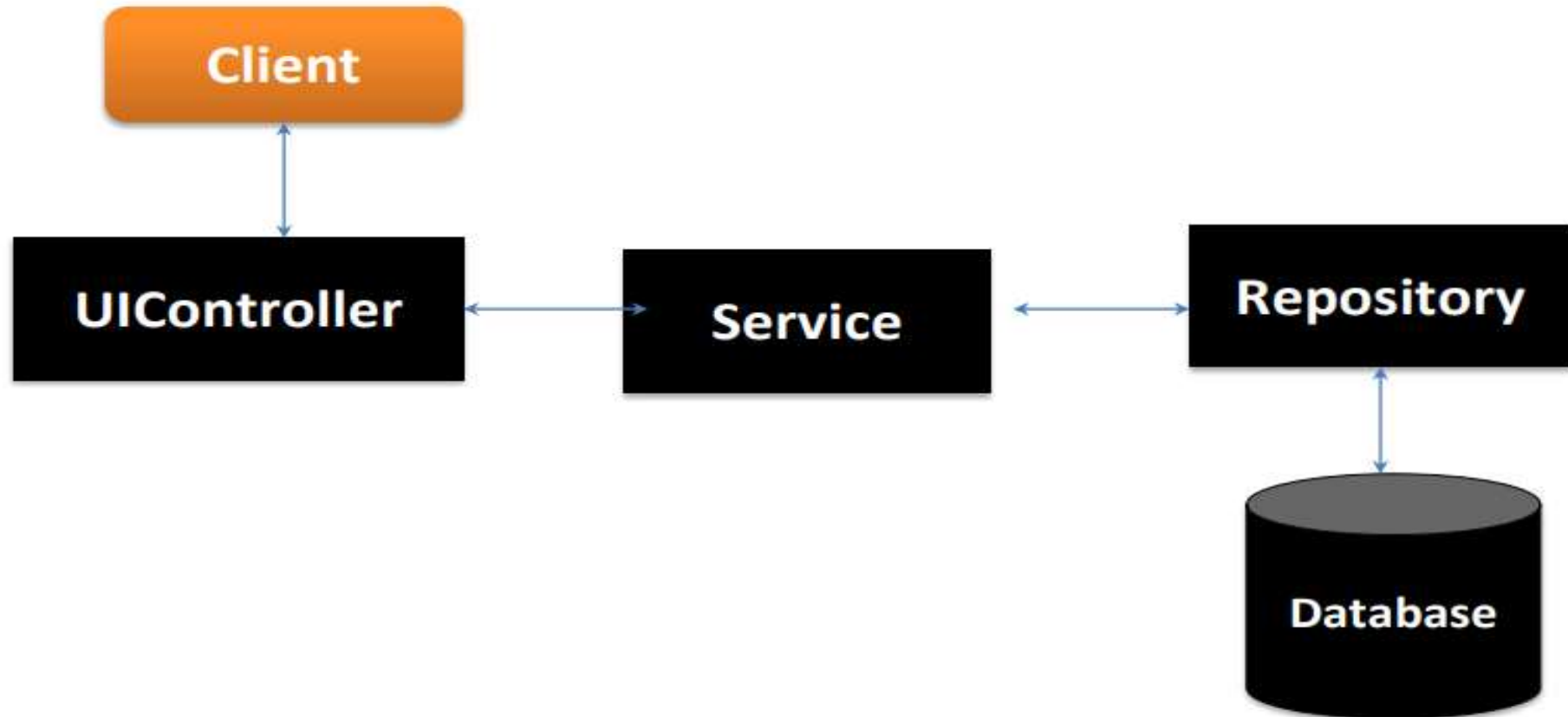
- Core Java
 - Spring Boot
 - Spring Data JPA
 - Hibernate
- Spring Boot Web
- MySQL Database
- JUnit

About Back-End:

- There are mainly four operations will be performed by the user and admin Ex. Insert, update, retrieve/fetch, delete.
- These operation will be performed using the spring boot framework, core java and spring boot web , spring data Jpa , and hibernate.
- And thymeleaf is the design pattern for the user interface.
- For connectivity purpose we are using MySQL database. It is connected to the java and database

Working Of Back-End ;

➤ There are mainly three stages



- **UIController** - In Spring Boot, the UIController class is responsible for processing incoming requests, preparing a model, and returning the view to be rendered as a response.
- **Service - Components** are the class file which contains @Service annotation.
- These class files are used to write business logic in a different layer.

- Repository - Repository is a specialization of @Component annotation which is used to indicate that the class provides the mechanism for storage, retrieval, update, delete and search operation on objects
- Repository is directly connected with the database and then it can return to the repository and response to the service and it can return to the controller and then response send to the client.

Advantages:

- Better and improved admission rate
- Better teacher performance
- Drastic reduction in manual tasks
- Elimination of paper based tasks
- Improved fee management

- Makes the institute smart and ready for the future.

Improved institute reputation

better resource utilization

- Duplication of the student and teacher data is avoided.

snap shots:

Add Student:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' tabs, a search bar, and buttons for 'Sign In' and 'Create Account'. Below this is a yellow banner indicating 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into three sections: a left sidebar with 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'; a central area with a 'Scratch Pad' tab and a message 'You don't have any collections'; and a right panel for editing a request. The request is a POST to 'http://localhost:8893/student'. The 'Body' tab is selected, showing a JSON payload with student details. Below the request editor, the 'Test Results' section shows the response in 'Pretty' format, which is a JSON object containing student information. The bottom of the screen shows the Windows taskbar with the search bar and various application icons.

Home Workspaces Explore Search Postman Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad New Import Overview POST http://localhost:8893/ DEL http://localhost:8888/d + ... No Environment

http://localhost:8893/student Save

POST http://localhost:8893/student Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
5 ..... "studentAddress": "salem",
6 ..... "studentEmail": "suresh@gmail.com",
7 ..... "studentAge": 21,
8 ..... "studentFees": 50000.0
9 .....
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 1823 ms Size: 335 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2 "studentId": 4,
3 "studentName": "suresh",
4 "studentMobileNo": "7904641213",
5 "studentAddress": "salem",
6 "studentEmail": "suresh@gmail.com",
7 "studentAge": 21,
8 "studentFees": 50000.0
9
```

Find and Replace Console Runner Trash 16:00 25-09-2022

Get All Student Details:

HomeWorkspacesExplore

Search Postman

Sign InCreate Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch PadNewImportOverviewGET http://localhost:8893/sDEL http://localhost:8888/d+...No Environment

Collections+...APIsEnvironmentsMock ServersMonitorsHistory

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

http://localhost:8893/studentSave

GET http://localhost:8893/studentSend

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencoderawbinaryGraphQLJSON

5... "studentAddress": "salem",

6... "studentEmail": "suresh@gmail.com",

7... "studentAge": 21,

8... "studentFees": 50000.0

9...

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 44 msSize: 660 BSave Response

PrettyRawPreviewVisualizeJSON

1{

2{

3"studentId": 2,

4"studentName": "navin",

5"studentMobileNo": "5758687979",

6"studentAddress": "salem",

7"studentEmail": "navin@gmail.com",

8"studentAge": 21,

9"studentFees": 50000.0

10},

11{

12"studentId": 2

Find and ReplaceConsole

RunnerTrash

Type here to search

AQI 3116:0125-09-2022

Update Student:

HomeWorkspacesExplore

Search Postman

Sign InCreate Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch PadNewImport

OverviewPUT http://localhost:8893/sDEL http://localhost:8888/d+...

No Environment

Collections

APIs

Environments

Mock Servers

Monitors

History

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

http://localhost:8893/student/4

Save

Send

PUThttp://localhost:8893/student/4

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

3....."studentName": "suresh",

4....."studentMobileNo": "7904641213",

5....."studentAddress": "mettux",

6....."studentEmail": "suresh@gmail.com",

7....."studentAge": 21,

8....."studentFees": 50000.0

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 694 msSize: 331 BSave Response

PrettyRawPreviewVisualizeJSON

1

2"studentId": 4,

3"studentName": "suresh",

4"studentMobileNo": "7904641213",

5"studentAddress": "mettux",

6"studentEmail": "suresh@gmail.com",

7"studentAge": 21,

8"studentFees": 50000.0

9

Find and ReplaceConsole

RunnerTrash

29°C Partly sunny16:0225-09-2022

Delete Student By Id:

HomeWorkspacesExplore

Search Postman

Sign InCreate Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch PadNewImport

OverviewDEL http://localhost:8893/sDEL http://localhost:8888/d+...

No Environment

Collections

APIs

Environments

Mock Servers

Monitors

History

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

http://localhost:8893/student/4

Save

Send

DELETE

http://localhost:8893/student/4

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

3"studentName": "suresh",

4"studentMobileNo": "7904641213",

5"studentAddress": "mettur",

6"studentEmail": "suresh@gmail.com",

7"studentAge": 21,

8"studentGender": "Male"

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 179 msSize: 181 BSave Response

PrettyRawPreviewVisualizeText

1 record is deleted

Find and ReplaceConsole

RunnerTrash

29°C Partly sunny16:0325-09-2022

Get Student By Id:

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore'. A search bar labeled 'Search Postman' is present. On the right, there are links for 'Sign In' and 'Create Account'. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'.

The main interface is divided into three sections:

- Left Panel (Collections):** Shows a message 'You don't have any collections' with a cartoon character holding a box. It also includes a 'Create Collection' button and a list of sidebar options: Collections, APIs, Environments, Mock Servers, Monitors, and History.
- Middle Panel (Request Editor):** Displays a REST client request for 'http://localhost:8893/student/2' using the GET method. The request is configured with the following settings:
 - Method: GET
 - URL: http://localhost:8893/student/2
 - Params: none
 - Authorization: none
 - Headers: 8
 - Body: raw
 - Pre-request Script: none
 - Tests: none
 - Settings: none
- Right Panel (Response Viewer):** Shows the response status as '200 OK' with a time of '65 ms' and a size of '328 B'. The response body is displayed in JSON format, showing student details for ID 2.

The response body is as follows:

```
{  "studentId": 2,  "studentName": "navin",  "studentMobileNo": "5758687979",  "studentAddress": "salem",  "studentEmail": "navin@gmail.com",  "studentAge": 21,  "studentFees": 50000.0}
```

The bottom of the screen shows the Windows taskbar with the search bar, task view button, and several open applications. The system clock shows '16:05' on '25-09-2022' with a weather forecast of '29°C Partly sunny'.

Add Teacher:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options. A search bar labeled 'Search Postman' is also present. On the right, there are buttons for 'Sign In' and 'Create Account'. Below this, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'.

The main interface is divided into three panels. The left panel, titled 'Scratch Pad', shows a sidebar with icons for 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The central area of this panel contains a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run. Create Collection'.

The right panel shows a REST client request. The URL is 'http://localhost:8893/teacher'. The method is 'POST'. The request body is in JSON format, containing the following data:

```
1 {
2   "teacherName": "senthil",
3   "teacherSalary": 50000.0
4 }
```

Below the request body, the response is displayed in the 'Body' tab. The response is a JSON object with the following data:

```
1 {
2   "teacherId": 5,
3   "teacherName": "senthil",
4   "teacherSalary": 50000.0
5 }
```

The status of the request is '201 Created', with a response time of '142 ms' and a size of '232 B'. The bottom of the interface shows a Windows taskbar with various application icons and a system tray displaying the date and time as '16:07 25-09-2022'.

Get All Teacher:

HomeWorkspacesExplore

Search Postman

Sign InCreate Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch PadNewImportOverviewGET http://localhost:8893/tDEL http://localhost:8888/d+...No Environment

Collections+Collections

You don't have any collections

Collections let you group related requests, making them easier to access and run.

Create Collection

APIsEnvironmentsMock ServersMonitorsHistory

http://localhost:8893/teacherSave

GEThttp://localhost:8893/teacherSend

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookiesBeautify

noneform-datax-www-form-urlencodetrawbinaryGraphQLJSON

1234

... "teacherName": "senthil",
... "teacherSalary": 50000.0

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 44 msSize: 421 BSave Response

PrettyRawPreviewVisualizeJSON

12345678910111213

{
 "teacherId": 2,
 "teacherName": "mekala",
 "teacherSalary": 40000.0
},
{
 "teacherId": 3,
 "teacherName": "priya",
 "teacherSalary": 40000.0
},
{
 "teacherId": 4,

Find and ReplaceConsole

RunnerTrash

Type here to search

29°C Partly sunny16:0825-09-2022

Get Teeacer By Id:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options, along with a search bar and 'Sign In'/'Create Account' buttons. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The left sidebar shows 'Scratch Pad' with 'New' and 'Import' buttons, and a 'Collections' section with a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run. Create Collection'. The main area shows a GET request to 'http://localhost:8893/teacher/2'. The request body is a JSON object:

```
{  "teacherName": "senthil",  "teacherSalary": 50000.0}
```

. The response is shown in the 'Body' tab, displaying a JSON object:

```
{  "teacherId": 2,  "teacherName": "mekala",  "teacherSalary": 40000.0}
```

. The status is '200 OK', time is '112 ms', and size is '226 B'. The bottom status bar shows 'Runner', 'Trash', and system information: '29°C Partly sunny', '16:08', and '25-09-2022'.

Home Workspaces ▾ Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad New Import Overview GET http://localhost:8893/t DEL http://localhost:8888/d + ... No Environment

Collections

APIs

Environments

Mock Servers

Monitors

History

You don't have any collections
Collections let you group related requests, making them easier to access and run.
Create Collection

Save

GET http://localhost:8893/teacher/2 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1  {
2    "teacherName": "senthil",
3    "teacherSalary": 50000.0
4  }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 112 ms Size: 226 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "teacherId": 2,
3    "teacherName": "mekala",
4    "teacherSalary": 40000.0
5  }
```

Find and Replace Console

Runner Trash

Type here to search

29°C Partly sunny 16:08 25-09-2022

Update Teacher:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options. A search bar labeled 'Search Postman' is also present. On the right side of the top bar, there are links for 'Sign In' and 'Create Account', along with window management icons.

Below the top bar, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into three sections:

- Left Panel (Scratch Pad):** Contains a sidebar with icons for 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main area of this panel shows a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run. Create Collection'.
- Middle Panel (Request Editor):** Shows a PUT request to 'http://localhost:8893/teacher/5'. The request body is in JSON format, containing:

```
{  "teacherName": "senthil",  "teacherSalary": 45000.0}
```
- Right Panel (Response Viewer):** Displays the response from the server. The status is '200 OK', the time taken is '289 ms', and the size is '227 B'. The response body is in JSON format, containing:

```
{  "teacherId": 5,  "teacherName": "senthil",  "teacherSalary": 45000.0}
```

At the bottom of the application, there's a status bar with a 'Find and Replace' search bar, a 'Console' tab, and a system tray showing the date and time as '16:08 25-09-2022'.

Delete Teacher By Id:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options, along with a search bar and user account links. Below this, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into a left sidebar with navigation options like 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The central area shows a 'Scratch Pad' with a 'New' button and an 'Import' button. A message states 'You don't have any collections' with a 'Create Collection' link. The right pane shows an active request to 'http://localhost:8893/teacher/5' with a 'DELETE' method. The request body is a JSON object:

```
{  "teacherName": "senthil",  "teacherSalary": 45000.0}
```

. The response status is '200 OK' with a time of '159 ms' and a size of '181 B'. The response body is 'record is deleted'. The bottom of the image shows the Windows taskbar with the search bar, task icons, and system tray information including temperature, weather, and time.

Home Workspaces Explore Search Postman Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad New Import Overview

DEL http://localhost:8893/t DEL http://localhost:8888/d + ... No Environment

http://localhost:8893/teacher/5 Save

DELETE http://localhost:8893/teacher/5 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "teacherName": "senthil",
3   "teacherSalary": 45000.0
4 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 159 ms Size: 181 B Save Response

Pretty Raw Preview Visualize Text

```
1 record is deleted
```

Find and Replace Console Runner Trash 29°C Partly sunny 16:09 25-09-2022

Add Subject:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' links, a search bar, and buttons for 'Sign In' and 'Create Account'. Below this is a yellow banner indicating 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into three sections: a left sidebar, a central workspace, and a right-hand details panel.

Left Sidebar: Contains a 'Scratch Pad' section with 'New' and 'Import' buttons. Below it is a 'Collections' section with a plus icon and a list of icons for 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. A message states 'You don't have any collections' with a 'Create Collection' link.

Central Workspace: Shows a REST client request. The URL is 'http://localhost:8893/subject'. The method is 'POST'. The request body is in JSON format, containing:

```
{  "subjectName": "java"}
```

. The 'Body' tab is selected, and the 'raw' radio button is chosen. The 'Send' button is visible.

Right-hand Details Panel: Shows the response of the request. The status is '201 Created', time is '302 ms', and size is '241 B'. The response body is in JSON format, containing:

```
{  "subjectId": 5,  "subjectName": "java",  "enrolledstudent": [],  "teacher": null}
```

. The 'Body' tab is selected, and the 'Pretty' radio button is chosen.

Bottom Bar: Includes a 'Find and Replace' search bar, a 'Console' tab, and a system tray with a taskbar showing various application icons, the date '25-09-2022', and the time '16:11'.

Get All Subject:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options. A search bar labeled 'Search Postman' is present. On the right, there are links for 'Sign In' and 'Create Account'. Below this, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'.

The main interface is divided into three sections:

- Left Sidebar:** Contains a 'Scratch Pad' section with 'New' and 'Import' buttons. Below it, a 'Collections' section shows a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run.' with a 'Create Collection' button. Further down are links for 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'.
- Center Panel:** Shows a request configuration for 'http://localhost:8893/subject'. The method is 'GET'. The 'Body' tab is selected, showing a JSON payload:

```
{  "subjectName": "java"}
```

. The 'Send' button is visible.
- Right Panel:** Displays the response details. The status is '200 OK', time is '63 ms', and size is '690 B'. The 'Body' tab is selected, showing a detailed JSON response:

```
{  "subjectId": 2,  "subjectName": "python",  "enrolledstudent": [    {      "studentId": 2,      "studentName": "navin",      "studentMobileNo": "5758687979",      "studentAddress": "salem",      "studentEmail": "navin@gmail.com",      "studentAge": 21,      "studentFees": 50000.0    }  ]}
```

At the bottom, there's a Windows taskbar with the Start button, a search bar, and several application icons. The system tray shows the date and time as '16:11 25-09-2022' and the weather as '29°C Partly sunny'.

Get Subject By Id:

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', and 'Explore'. A search bar is present. The main workspace is titled 'Working locally in Scratch Pad. Switch to a Workspace'. The left sidebar shows 'Collections' with a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run. Create Collection'. The main panel displays a REST client request for 'http://localhost:8893/subject/4' using the 'GET' method. The request body is a JSON object:

```
{  "subjectName": "java"}
```

. The response is displayed in the bottom panel, showing a status of '200 OK', time of '73 ms', and size of '245 B'. The response body is a JSON object:

```
{  "subjectId": 4,  "subjectName": "communication",  "enrolledstudent": [],  "teacher": null}
```

. The bottom status bar shows the Windows taskbar with the search bar, taskbar icons, and system tray information: '29°C Partly sunny', '16:11', and '25-09-2022'.

Home Workspaces Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad New Import Overview

GET http://localhost:8893/s DEL http://localhost:8888/d + ... No Environment

Save Save

http://localhost:8893/subject/4

GET http://localhost:8893/subject/4 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "subjectName": "java"
3 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 73 ms Size: 245 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "subjectId": 4,
3   "subjectName": "communication",
4   "enrolledstudent": [],
5   "teacher": null
6 }
```

Find and Replace Console Runner Trash

Type here to search

29°C Partly sunny 16:11 25-09-2022

Update Subject:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options. A search bar labeled 'Search Postman' is also present. On the right side of the top bar, there are links for 'Sign In' and 'Create Account', along with window management icons.

Below the top bar, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into several sections:

- Left Sidebar:** Contains navigation options like 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. A message states 'You don't have any collections' with a 'Create Collection' button.
- Top Bar (Scratch Pad):** Includes 'New' and 'Import' buttons, and tabs for 'Overview', 'PUT http://localhost:8893/s', and 'DEL http://localhost:8888/d'. It also shows 'No Environment' and a 'Save' button.
- Main Panel:** Displays a PUT request to 'http://localhost:8893/subject/4'. The request body is in JSON format:

```
{  "subjectName": "core java"}
```

. The 'Body' tab is selected, showing the JSON response:

```
{  "subjectId": 4,  "subjectName": "core java",  "enrolledstudent": [],  "teacher": null}
```

. The status bar at the bottom of the main panel shows 'Status: 200 OK', 'Time: 303 ms', and 'Size: 241 B'.

The bottom of the image shows the Windows taskbar with the search bar, task view button, and several application icons. The system tray on the right shows the date and time as '16:12 25-09-2022' and the weather as '29°C Partly sunny'.

Delete Subject By Id:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options, along with a search bar and user account links. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The left sidebar shows a 'Scratch Pad' section with 'New' and 'Import' buttons, and a 'Collections' section with a message 'You don't have any collections' and a 'Create Collection' button. The main workspace is divided into two tabs: 'Overview' and 'http://localhost:8893/s'. The 'http://localhost:8893/s' tab is active, showing a 'DELETE' request to 'http://localhost:8893/subject/4'. The request body is a JSON object:

```
{  "subjectName": "core-java"}
```

. The response is a 200 OK status with a message 'record is deleted'. The bottom status bar shows the time as 16:12 on 25-09-2022, with weather information (29°C Partly sunny) and system icons.

Home Workspaces Explore Search Postman Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad New Import Overview DEL http://localhost:8893/s DEL http://localhost:8888/d + ... No Environment

Collections + ...

APIs

Environments

Mock Servers

Monitors

History

You don't have any collections
Collections let you group related requests, making them easier to access and run.
Create Collection

http://localhost:8893/subject/4 Save

DELETE http://localhost:8893/subject/4 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "subjectName": "core-java"
3 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 227 ms Size: 181 B Save Response

Pretty Raw Preview Visualize Text

```
1 record is deleted
```

Find and Replace Console Runner Trash 16:12 25-09-2022 29°C Partly sunny

Assign Subject To Student:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options. A search bar labeled 'Search Postman' is present. On the right, there are links for 'Sign In' and 'Create Account'. Below this, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into three sections: 'Scratch Pad' on the left, a central request editor, and a response viewer at the bottom.

Scratch Pad: This section on the left contains a sidebar with icons for 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main area of the Scratch Pad shows a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run.' with a 'Create Collection' button.

Request Editor: The central part of the interface shows a PUT request to the URL 'http://localhost:8893/subject/3/student/5'. The request body is formatted as JSON and contains the following data:

```
1 {
2   "studentName": "suresh",
3   "studentMobileNo": "9686868799",
4   "studentAddress": "salem",
5   "studentEmail": "suresh@gmail.com",
6   "studentAge": 21,
7   "studentFees": 50000.0
8 }
```

The request is set to 'JSON' format. Below the request editor, the response viewer shows the status '200 OK' with a time of '196 ms' and a size of '402 B'. The response body is also in JSON format and contains the following data:

```
1 {
2   "subjectId": 3,
3   "subjectName": "html",
4   "enrolledstudent": [
5     {
6       "studentId": 5,
7       "studentName": "suresh",
8       "studentMobileNo": "9686868799",

```

The bottom of the screen shows a Windows taskbar with the search bar, task view button, and several application icons. The system tray on the right indicates the temperature is '29°C Partly sunny', the time is '16:16', and the date is '25-09-2022'.

Assign Subject To Teacher:

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options. A search bar labeled 'Search Postman' is present. On the right, there are links for 'Sign In' and 'Create Account'. Below this, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into a left sidebar and a central workspace. The sidebar contains a 'Scratch Pad' section with 'New' and 'Import' buttons, and a 'Collections' section with a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run. Create Collection'. The central workspace shows a PUT request to 'http://localhost:8893/subject/3/teacher/4'. The request body is in JSON format, containing student details: 'studentName': 'suresh', 'studentMobileNo': '9686868799', 'studentAddress': 'salem', 'studentEmail': 'suresh@gmail.com', and 'studentAge': 21. The response is also in JSON format, showing the subject details: 'subjectId': 3, 'subjectName': 'html', and 'enrolledstudent' as an array containing the student details. The status of the request is '200 OK' with a time of '377 ms' and a size of '464 B'. The bottom of the screen shows a Windows taskbar with various application icons and a system tray displaying the date and time as '16:17 25-09-2022'.

Home Workspaces Explore Search Postman Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad New Import Overview PUT http://localhost:8893/s DEL http://localhost:8888/d + ... No Environment

Collections

You don't have any collections
Collections let you group related requests, making them easier to access and run.
Create Collection

APIs Environments Mock Servers Monitors History

http://localhost:8893/subject/3/teacher/4 Save

PUT http://localhost:8893/subject/3/teacher/4 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "studentName": "suresh",
3   "studentMobileNo": "9686868799",
4   "studentAddress": "salem",
5   "studentEmail": "suresh@gmail.com",
6   "studentAge": 21
7 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 377 ms Size: 464 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "subjectId": 3,
3   "subjectName": "html",
4   "enrolledstudent": [
5     {
6       "studentId": 5,
7       "studentName": "suresh",
8       "studentMobileNo": "9686868799",
9       "studentAddress": "salem",
10      "studentEmail": "suresh@gmail.com",
11      "studentAge": 21,
12    }
13  ]
14 }
```

Find and Replace Console Runner Trash 16:17 25-09-2022 29°C Partly sunny

Profit and Loss of College:

HomeWorkspacesExplore

Search Postman

Sign InCreate Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch PadNewImportOverviewGET http://localhost:8893/pandlofcollegeDEL http://localhost:8888/d+...No Environment

Collections+...

You don't have any collections
Collections let you group related requests, making them easier to access and run.
Create Collection

APIsEnvironmentsMock ServersMonitorsHistory

http://localhost:8893/pandlofcollegeSave

GEThttp://localhost:8893/pandlofcollegeSend

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencoderawbinaryGraphQLJSONBeautify

```
1 {
2   "studentName": "suresh",
3   "studentMobileNo": "9686868799",
4   "studentAddress": "salem",
5   "studentEmail": "suresh@gmail.com",
6   "studentAge": 21
}
```

BodyCookiesHeaders (5)Test ResultsStatus: 200 OKTime: 164 msSize: 178 BSave Response

PrettyRawPreviewVisualizeText

1 Profit 30000.0

Find and ReplaceConsole

RunnerTrash

Windows Taskbar

29°C Partly sunny16:18 25-09-2022

Conclusion :

- **The project entitled as College Management System is the system that deals with the issues related to a particular institution.**
- This project is successfully implemented with all the features mentioned in system requirements specification.
- Awareness and right information about any college is essential for both the development of student as well as faculty.
- So this serves the right purpose in achieving the desired requirements of both the communities.