

Employee Management System

Project Member :

Member

- ❖ Ms.Snehal Rokhonde
- ❖ Mr.Vivek Kumar
- ❖ Mrs. Pratibha Vijapure
- ❖ Ms. Ashwini Murahde
- ❖ Ms. Harshita JK

Guided By

- ❖ Prof. Indrakka Mali

Index

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

Introduction

- ❑ Employee management system is a distributed application, developed to maintain the details of employees working in any organization.
- ❑ It maintains the information about the personal details of their employees, also the details about the payroll system which enables to generate the payslip.
- ❑ The application is actually a suit of applications developed using java. Its simple to understand even if your not smiler with this application.

- ☐ **The application is reduced as much as possible to avoid errors while entering the data.**
- ☐ **It also provides error message while entering invalid data.**
- ☐ **No formal knowledge is needed for the user to use this system.**
- ☐ **Thus, by this all it proves it is user-friendly.**
- ☐ **Employee Management System, It can assist the user to concentrate on their other activities rather concentrate on the record-keeping.**
- ☐ **Thus, it will help the organization in better utilization of resources.**

Objective

- ❑ The purpose of the Employee Management System is to automate the existing manual system with the help of computerized equipment and full-fledged computer software, fulfilling their requirements so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easy to work with.
- ❑ Employee Management systems, as described above, can lead to error-free, secure, reliable and fast management systems.

- ☐ It can assist the user to concentrate on their activities rather to concentrate on the record keeping.
- ☐ Thus, it will help an organization in better utilization of resources.
- ☐ The organization can maintain computerized records without redundant entries.
- ☐ That means that one need not be distracted by information that is not relevant while being able to reach the information.

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

Back-End

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

Proposed System

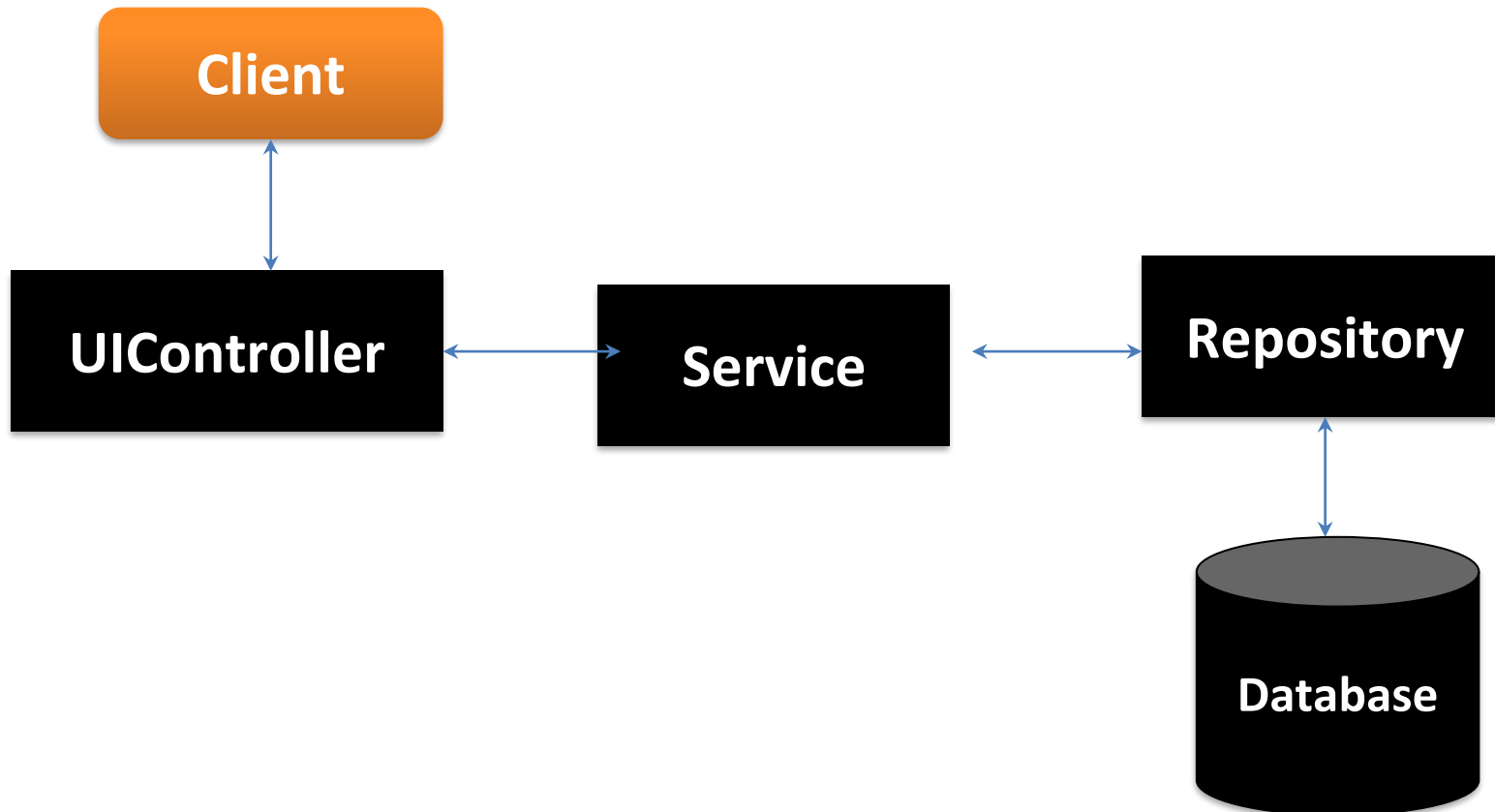
- ❑ First we have to run the Application on the browser, the index page will be shown after that if the user is registered then he/she can login the page and he/she can entering in the application.
- ❑ If the user is not registered then he/she needs to register themselves and then he/she can continuing 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.

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 –

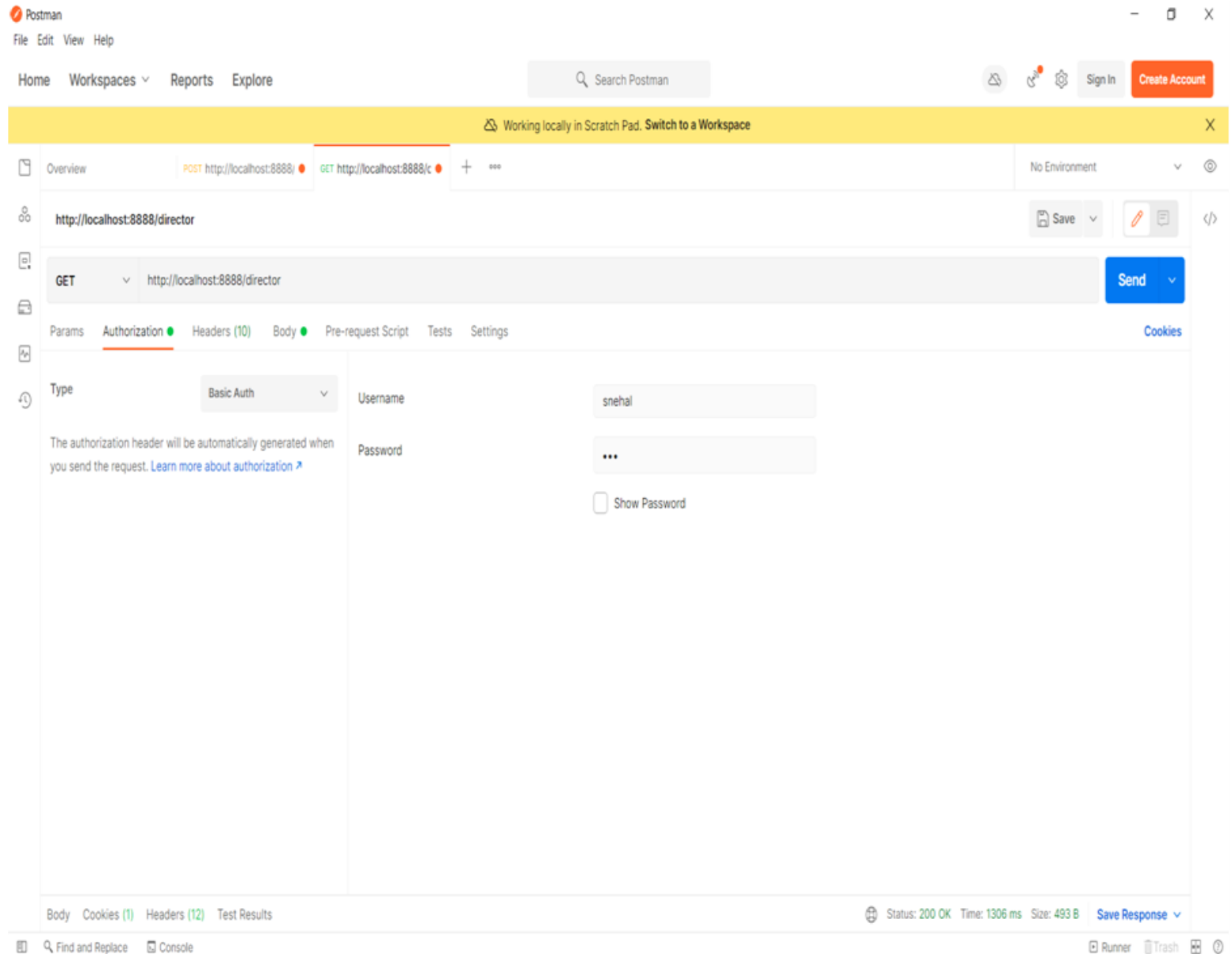


- ❑ **UIController** - In Spring Boot, the UIController class is responsible for processing incoming requests, preparing a model, and returning the view to be rendered as a response.
- ❑ **Service** - Components are the class file which contains @Service annotation. These class files are used to write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

Snapshots

LOG IN PAGE



2 Get All Record : / director/All

The image shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar is also present. Below this, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into several sections. The top section shows the request details: a GET request to 'http://localhost:8888/director/All'. The 'Body' tab is selected, showing a JSON response. The response is a list of two employee records. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 268 ms', and 'Size: 1.53 KB'. The bottom of the interface has a 'Find and Replace' bar and a 'Console' section.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Overview POST http://localhost:8888/ GET http://localhost:8888/c

http://localhost:8888/director/All

GET http://localhost:8888/director/All

Params Authorization Headers (10) Body Pre-request Script Tests Settings

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

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 268 ms Size: 1.53 KB Save Response

Pretty Raw Preview Visualize JSON

```
3 {
4   "empId": 1,
5   "empName": "rina",
6   "empSalary": 32000.0,
7   "empEmailId": "rina@gmail.com",
8   "empage": 23,
9   "empCellNo": "1234567899",
10  "empAddress": "amravti",
11  "empDesignation": "DIRECTOR",
12  "empExperience": 3.0,
13  "empUsername": "priya",
14  "empPassword": "$2a$12$8NYa.gTmd0N14pwA/bybq.Zpdx5CwkCjw7f0UzWvz8Bg9zph/eQw6",
15  "managerId": 0
16 },
17 {
18   "empId": 999,
19   "empName": "snehal",
20   "empSalary": 32000.0,
21   "empEmailId": "snehal@gmail.com",
22   "empage": 23,
23   "empCellNo": "1234567899",
24   "empAddress": "amravti",
25   "empDesignation": "DIRECTOR",
26   "empExperience": 3.0,
27   "empUsername": "priya",
28   "empPassword": "$2a$12$8NYa.gTmd0N14pwA/bybq.Zpdx5CwkCjw7f0UzWvz8Bg9zph/eQw6",
29   "managerId": 0
30 }
```

Find and Replace Console

Runner Trash

3 Get Employee List: /director/{managerId}

The screenshot shows the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it, a navigation bar includes 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar labeled 'Search Postman' is also present. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'.

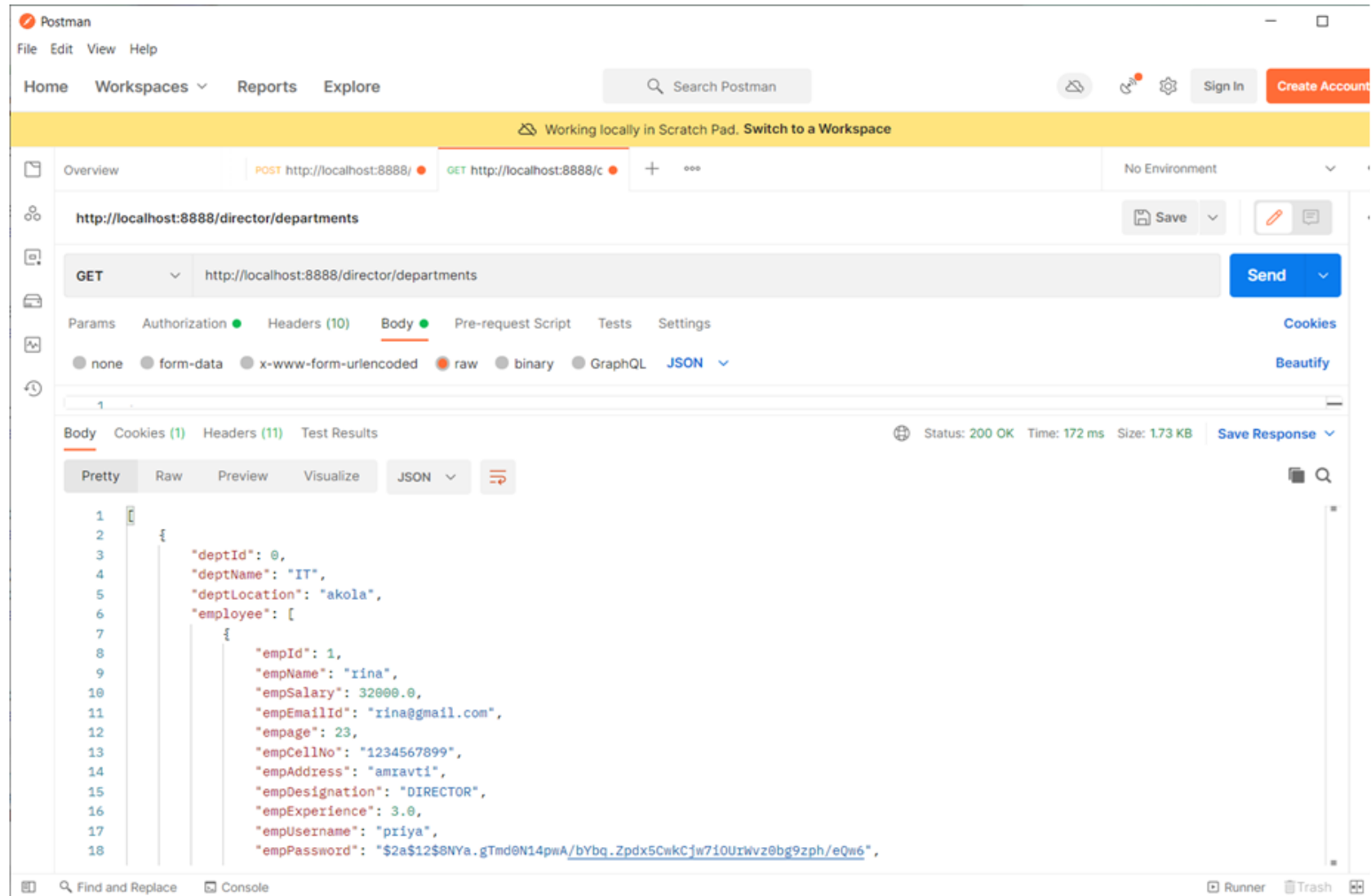
The main workspace shows a GET request to `http://localhost:8888/director/1000`. The request is configured with the following details:

- Method: GET
- URL: `http://localhost:8888/director/1000`
- Body: JSON (selected)

The response is displayed in the 'Body' tab, showing a JSON array of employee data. The status is 200 OK, with a response time of 27 ms and a size of 655 B.

```
1 [
2   {
3     "empId": 1002,
4     "empName": "sneha",
5     "empSalary": 230000.0,
6     "empEmailId": "sneha@gmail.com",
7     "empage": 26,
8     "empCellNo": "1789567891",
9     "empAddress": "nashik",
10    "empDesignation": "EMPLOYEE",
11    "empExperience": 2.0,
12    "empUsername": "sneha",
13    "empPassword": "$2a$10$3kxU/DyuNKHVqaJo3CcBBEh.HQcI1tYClzCSVfI7p7w10KLFwNLEm",
14    "managerId": 1000
15  }
16 ]
```

4 Get All Departments : /director/departments



The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu items (File, Edit, View, Help), navigation links (Home, Workspaces, Reports, Explore), a search bar, and user options (Sign In, Create Account). A yellow banner indicates "Working locally in Scratch Pad. Switch to a Workspace".

The main interface displays a GET request to `http://localhost:8888/director/departments`. The request is configured with the following settings:

- Method: GET
- URL: `http://localhost:8888/director/departments`
- Body: JSON (selected)

The response is displayed in the "Body" tab, showing a JSON object with the following structure:

```
1 {
2   "deptId": 0,
3   "deptName": "IT",
4   "deptLocation": "akola",
5   "employee": [
6     {
7       "empId": 1,
8       "empName": "rina",
9       "empSalary": 32000.0,
10      "empEmailId": "rina@gmail.com",
11      "empage": 23,
12      "empCellNo": "1234567899",
13      "empAddress": "amravti",
14      "empDesignation": "DIRECTOR",
15      "empExperience": 3.0,
16      "empUsername": "priya",
17      "empPassword": "$2a$12$8NYa.gTmd0N14pwA/bYbq.Zpdx5CwkCjw7i0Urwvz@bg9zph/eQw6",
18    }
19  ]
20 }
```

The status bar at the bottom shows "Status: 200 OK", "Time: 172 ms", and "Size: 1.73 KB". The bottom of the interface includes a "Find and Replace" bar, a "Console" tab, and a "Runner" button.

5 Add department: /director/addDept

The screenshot shows the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it, a navigation bar includes 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar labeled 'Search Postman' is on the right. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'.

The main workspace shows a collection of requests. The selected request is a POST to 'http://localhost:8888/director/addDept'. The request is configured with the following details:

- Method:** POST
- URL:** http://localhost:8888/director/addDept
- Body Type:** raw (selected), with a content type of 'JSON'.
- Body Content:**

```
1 -
2 -
3 {  "deptName": "phy",
4   "deptLocation": "yavataml"
5 }
6 -
```

The response is displayed at the bottom, showing a status of 200 OK, a time of 96 ms, and a size of 415 B. The response body is a JSON object:

```
1 {
2   "deptId": 3,
3   "deptName": "phy",
4   "deptLocation": "yavataml",
5   "employee": []
6 }
```

The bottom of the interface includes a 'Find and Replace' bar and a 'Console' area.

6 update department: /director/update/department/{deptId}

The screenshot shows the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it, a navigation bar includes 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar labeled 'Search Postman' is also present. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface shows a PUT request to 'http://localhost:8888/director/update/department/3'. The request body is in JSON format, containing 'deptName': 'sci' and 'deptLocation': 'yavataml'. The response is also in JSON format, showing 'deptId': 3, 'deptName': 'sci', 'deptLocation': 'yavataml', and 'employee': []. The status is '200 OK' with a time of '175 ms' and a size of '415 B'.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Overview POST http://localhost:8888/ PUT http://localhost:8888/c + ... No Environment

http://localhost:8888/director/update/department/3 Save

PUT http://localhost:8888/director/update/department/3 Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

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

```
1 -
2 -
3 {
4   "deptName": "sci",
5   "deptLocation": "yavataml"
6 }
```

Body Cookies (1) Headers (11) Test Results Status: 200 OK Time: 175 ms Size: 415 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "deptId": 3,
3   "deptName": "sci",
4   "deptLocation": "yavataml",
5   "employee": []
6 }
```

Find and Replace Console Runner Trash

7 delete department : /director/delete/department/{deptId}

The screenshot displays the Postman application interface. At the top, the menu bar includes 'File', 'Edit', 'View', and 'Help'. Below it, the 'Home' tab is active, showing 'Workspaces', 'Reports', and 'Explore'. A search bar labeled 'Search Postman' is present. A yellow banner at the top indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main workspace shows a 'DELETE' request to 'http://localhost:8888/director/delete/department/3'. The request body is a JSON object:

```
{  \"deptName\": \"sci\",  \"deptLocation\": \"yavataml\"}
```

. The response is 'Record deleted succeseefully...!' with a status of 200 OK. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 02:09 on 30-04-2022.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Overview POST http://localhost:8888/ DEL http://localhost:8888/d

http://localhost:8888/director/delete/department/3

DELETE http://localhost:8888/director/delete/department/3

Params Authorization Headers (10) Body Pre-request Script Tests Settings

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

```
1 {
2   \"deptName\": \"sci\",
3   \"deptLocation\": \"yavataml\"
4 }
5
6
```

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 90 ms Size: 377 B Save Response

Pretty Raw Preview Visualize Text

1 Record deleted succeseefully...!

8 add employee : /director/addEmp/{deptId}

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Reports', and 'Explore'. Below that, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main area displays a POST request to 'http://localhost:8888/director/addEmp/2'. The request body is in JSON format, containing fields like 'empName', 'empSalary', 'empEmailId', 'empage', 'empCellNo', 'empAddress', 'empDesignation', 'empExperience', 'empUsername', 'empPassword', and 'managerId'. The response is also in JSON format, showing the same fields with values. The status is '200 OK', time is '498 ms', and size is '655 B'. The bottom of the screen shows a Windows taskbar with various icons and the system clock.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Overview POST http://localhost:8888/ POST http://localhost:8888/ + ...

No Environment

Save

Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings

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

1 2 3 4 5

```
1 {
2   "empName": "prithvi",
3   "empSalary": 238888.8,
4   "empEmailId": "pri@gmail.com",
5   "empage": 26,
6   "empCellNo": "1889567891",
7   "empAddress": "raygad",
8   "empDesignation": "EMPLOYEE",
9   "empExperience": 2.0,
10  "empUsername": "prithvi",
11  "empPassword": "$2a$10$Vlm8PTtCJfa.0CBKHniczeA7ySt3hKNNncDHZErC/4isJJ/la6rKi",
12  "managerId": 1000
13 }
```

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 498 ms Size: 655 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "empId": 1051,
3   "empName": "prithvi",
4   "empSalary": 238888.8,
5   "empEmailId": "pri@gmail.com",
6   "empage": 26,
7   "empCellNo": "1889567891",
8   "empAddress": "raygad",
9   "empDesignation": "EMPLOYEE",
10  "empExperience": 2.0,
11  "empUsername": "prithvi",
12  "empPassword": "$2a$10$Vlm8PTtCJfa.0CBKHniczeA7ySt3hKNNncDHZErC/4isJJ/la6rKi",
13  "managerId": 1000
14 }
```

Find and Replace Console

Runner Trash

Type here to search

37°C 02:14 30-04-2022

9 update employee : /director/updateEmp/{empId}

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar is also present. Below this, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into two sections: the top section for request configuration and the bottom section for response viewing.

Request Configuration:

- Method:** PUT
- URL:** http://localhost:8888/director/updateEmp/1051
- Body Type:** JSON
- Body Content:**

```
1  
2  
3  
4 {  
  "empName": "shivaji",  
}
```

Response Viewing:

- Status:** 200 OK
- Time:** 52 ms
- Size:** 655 B
- Body Content (JSON):**

```
1  
2 {  
  "empId": 1051,  
  "empName": "shivaji",  
  "empSalary": 230000.0,  
  "empEmailId": "pri@gamil.com",  
  "empage": 26,  
  "empCellNo": "1889567891",  
  "empAddress": "raygad",  
  "empDesignation": "EMPLOYEE",  
  "empExperience": 2.0,  
  "empUsername": "prithvi",  
  "empPassword": "$2a$10$Vlm8PTtCJfa.0CBKHnrczeA7ySt3hKNnncDHZErC/4is33/la6rKi",  
  "managerId": 1000  
}
```

The bottom of the image shows the Windows taskbar with various application icons and the system clock indicating 02:17 on 30-04-2022.

10 delete employee: /director /delete /{empld}

The screenshot displays the Postman application interface. At the top, the menu bar includes 'File', 'Edit', 'View', and 'Help'. Below it, the 'Home' tab is active, showing 'Workspaces', 'Reports', and 'Explore'. A search bar labeled 'Search Postman' is present. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main workspace shows a 'DELETE' request to 'http://localhost:8888/director/delete/1051'. The 'Body' tab is selected, showing a 'none' body type. The 'Send' button is visible. Below the request, the response is shown with a status of '200 OK', time of '39 ms', and size of '363 B'. The response body is 'Record deleted...'. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 02:19 on 30-04-2022.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Overview POST http://localhost:8888/ DEL http://localhost:8888/d

http://localhost:8888/director/delete/1051

DELETE http://localhost:8888/director/delete/1051

Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings

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

1
2
3

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 39 ms Size: 363 B Save Response

Pretty Raw Preview Visualize Text

1 Record deleted...

Find and Replace Console

Type here to search

Runner Trash

37°C 02:19 30-04-2022

11 Updateself : /director/updateself/{empId}

The screenshot displays the Postman application interface. At the top, the Postman logo and menu items (File, Edit, View, Help) are visible. Below the menu, there are tabs for Home, Workspaces, Reports, and Explore. A search bar labeled "Search Postman" is present. A yellow banner indicates "Working locally in Scratch Pad. Switch to a Workspace".

The main workspace shows a PUT request to the endpoint `http://localhost:8888/director/updateself/999`. The request body is a JSON object with the following fields:

```
1 {
2   "empId": 999,
3   "empName": "shivaji",
4   "empSalary": 230000.0,
5   "empEmailId": "pri@gamil.com",
6   "empage": 23,
7   "empCellNo": "1889567891",
8   "empAddress": "raygad",
9   "empDesignation": "DIRECTOR",
10  "empExperience": 2.0,
11  "empUsername": "prithvi",
12  "empPassword": "123",
13  "managerId": 0
14 }
```

The response status is 200 OK, with a time of 52 ms and a size of 594 B. The response is displayed in the "Body" tab, which is currently set to "Pretty" view. The response content is a JSON object with the same structure as the request body.

The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 02:27 on 30-04-2022.

12 Get employee List : /manager/getemp/{managerId}

The screenshot shows the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it, a navigation bar includes 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar labeled 'Search Postman' is present. A yellow banner at the top right says 'Working locally in Scratch Pad. Switch to a Workspace'. The main area displays a GET request to 'http://localhost:8888/manager/getemp/1000'. The 'Body' tab is selected, showing a JSON response. The response status is '200 OK' with a time of '25 ms' and a size of '655 B'. The JSON data is as follows:

```
{
  "empId": 1002,
  "empName": "sneha",
  "empSalary": 230000.0,
  "empEmailId": "sneha@gmail.com",
  "empAge": 26,
  "empCellNo": "1789567891",
  "empAddress": "nashik",
  "empDesignation": "EMPLOYEE",
  "empExperience": 2.0,
  "empUsername": "sneha",
  "empPassword": "$2a$10$3kxU/DyuNKMVqaJo3CcB8eH.HQcI1tYClzCSVfI7p7w10KLFWNLEm",
  "managerId": 1000
}
```

The bottom of the screen shows the Windows taskbar with various application icons and a system tray displaying '37°C', 'ENG', and the date '30-04-2022'.

12 Add employee : /manager/{deptId}

The image shows the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it, a navigation bar includes 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar labeled 'Search Postman' is on the right. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main area shows a collection of requests, with the selected one being a POST request to 'http://localhost:8888/manager/2'. The request body is in JSON format, containing employee details. The response is also in JSON format, showing a 200 OK status, a time of 262 ms, and a size of 650 B. The response body is displayed in the 'Body' tab, formatted as JSON.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Overview POST http://localhost:8888/ POST http://localhost:8888/ + ...

No Environment

Save

Send

POST http://localhost:8888/manager/2

Params Authorization Headers (10) Body Pre-request Script Tests Settings

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

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 262 ms Size: 650 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "empId": 1852,
3   "empName": "sai",
4   "empSalary": 238888.8,
5   "empEmailId": "sai@gamil.com",
6   "empage": 26,
7   "empCellNo": "1889587891",
8   "empAddress": "piratapgad",
9   "empDesignation": "EMPLOYEE",
10  "empExperience": 2.8,
11  "empUsername": "sai",
12  "empPassword": "$2a$10$yaHejLREm1EPvDABJ3sPI.dmvZYmqYBz1j9Trv1NIMXgZs0GjNafa",
13  "managerId": 1888
14 }
```

Find and Replace Console

Runner Trash

13 Update employee : /manager/update/{empId}

The image shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Reports', and 'Explore'. Below this is a search bar and a status bar indicating 'Working locally in Scratch Pad. Switch to a Workspace'. The main area is divided into tabs: 'Overview', 'POST http://localhost:8888/', and 'PUT http://localhost:8888/'. The 'PUT' tab is active, showing the URL 'http://localhost:8888/manager/update/1052'. The request body is set to 'JSON' and contains the following data:

```
{  "empId": 1052,  "empName": "soyara",  "empSalary": 230000.0,  "empEmailId": "sai@gmail.com",  "empage": 26,  "empCellNo": "1889587891",  "empAddress": "pistapad",  "empDesignation": "EMPLOYEE",  "empExperience": 2.0,  "empUsername": "sai",  "empPassword": "$2a$10$yaHejLREm1EPvDABJ3sPI.dmvZYMqYBz1j9TrV1NIMXgZs0GjNafa",  "managerId": 1000}
```

The response status is '200 OK' with a time of '59 ms' and a size of '653 B'. The response body is displayed in 'Pretty' format, showing the same JSON data as the request body.

At the bottom of the screen, there's a Windows taskbar with various icons and a search bar.

14 UpdateSelf : /manager/updateself/{empId}

The screenshot shows the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it, a navigation bar includes 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar labeled 'Search Postman' is present. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main workspace shows a PUT request to 'http://localhost:8888/manager/updateself/1000'. The request body is in JSON format, containing employee details. The response is a 200 OK status with a time of 226 ms and a size of 648 B. The response body is displayed in JSON format, showing the updated employee information.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Overview POST http://localhost:8888/ PUT http://localhost:8888/r + ...

No Environment

Save

Send

PUT http://localhost:8888/manager/updateself/1000

Params Authorization Headers (10) Body Pre-request Script Tests Settings

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

1
2
3

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 226 ms Size: 648 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "empId": 1000,
3   "empName": "soyara",
4   "empSalary": 230000.0,
5   "empEmailId": "sakhi@gmail.com",
6   "empage": 26,
7   "empCellNo": "1849587891",
8   "empAddress": "lonar",
9   "empDesignation": "MANAGER",
10  "empExperience": 2.0,
11  "empUsername": "sa",
12  "empPassword": "$2a$10$uTFZZUu9WcgjXevrFIj@iue6ZkwQIbdhZ5vjoZLmpK1gk3TjdZMd3m",
13  "managerId": 999
14 }
```

Find and Replace Console

Runner Trash

Type here to search

37°C

02:50
30-04-2022

15 Delete employee : /manager/delete/{empId}

The screenshot displays the Postman application interface. At the top, the Postman logo and menu (File, Edit, View, Help) are visible. Below the menu, there are navigation links: Home, Workspaces, Reports, and Explore. A search bar labeled 'Search Postman' is present. On the right, there are icons for cloud sync, a notification, settings, and buttons for 'Sign In' and 'Create Account'.

A yellow banner across the top of the main workspace reads: 'Working locally in Scratch Pad. Switch to a Workspace'. Below this, the 'Overview' tab is active, showing a list of requests. The selected request is a DELETE method to 'http://localhost:8888/manager/delete/1'. The request body is set to 'JSON' and contains the text: '13 {"empPassword": "123"}'. The 'Send' button is visible on the right.

Below the request editor, the response is displayed. The status is '200 OK', the time is '36 ms', and the size is '360 B'. The response body is shown in 'Pretty' format and contains the text: '1 Record Deleted'. The bottom of the interface shows a Windows taskbar with various application icons and a system tray displaying the temperature (37°C), date (30-04-2022), and time (02:54).

16 Update employee : /employee/update/{empId}

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar is also present. Below this, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main interface is divided into several sections. The top section shows the request details: a PUT request to 'http://localhost:8888/employee/update/1002'. The 'Body' tab is selected, showing a JSON payload. The 'Send' button is visible. Below the request details, the 'Body' tab is expanded, showing the JSON response in 'Pretty' format. The response is a JSON object with employee details. The bottom of the screen shows the Windows taskbar with various application icons and the system clock.

Postman

File Edit View Help

Home Workspaces Reports Explore

Search Postman

Sign In Create Account

Working locally in Scratch Pad. Switch to a Workspace

Overview POST http://localhost:8888/ PUT http://localhost:8888/employee/update/1002

http://localhost:8888/employee/update/1002

PUT http://localhost:8888/employee/update/1002

Params Authorization Headers (10) Body Pre-request Script Tests Settings

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

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 178 ms Size: 650 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "empId": 1002,
3   "empName": "sakhi",
4   "empSalary": 230000.0,
5   "empEmailId": "sara@gmail.com",
6   "empage": 26,
7   "empCellNo": "1848587891",
8   "empAddress": "lonar",
9   "empDesignation": "EMPLOYEE",
10  "empExperience": 2.0,
11  "empUsername": "sara",
12  "empPassword": "$2a$10$LndYRpHktspuDQRf687yC.de8ACpKJ/WHBxQUnhDexv/PYfssk8v6",
13  "managerId": 1000
14 }
```

Find and Replace Console

Runner Trash

Type here to search

37°C 03:02 30-04-2022

Advantages And Disadvantages

Advantages:

- Duplication of the employee data is avoided.
- As employee id is already stored in the master therefore human mistakes can be avoided Timely alerts ensure that penalties are avoided.

Conclusion

- ❑ Effectiveness, efficiency, and reliability are the key aspects that make this web-based Employee management system very useful for industrial and several other businesses.
- ❑ The proposed project is very flexible to handle new modules and features as per user requirements in future.
- ❑ It can also be integrated with other systems such as employee tracking system, employee information management system, employee registration system, etc.