

Employee Management System

Project Member and Mentor



Member



Ms. Snehal G. Rakhonde.



Ms. Pratibha vijapure.



Mr.Vivek Kumar.



Ms.Harshita.



Ms. Ashvini.



Mentor



Prof. Indrika 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.

Technologies To Be Used

☐ **Hardware Requirement**

- **Operating System : Windows OS**
- **Processor : intel i3 or higher**
- **RAM : 4 GB or +**
- **Dedicated Graphics card.**

Software Requirement

Back-End

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

Annotations

- **Annotations are used to provide supplemental information about a program.**
- **Annotations start with '@'.**
- **@Id: This id annotation is used to declare attribute as primary.**
- **@Generated Value: Used to generate value automatically when use insert new record**
- **@Sequence Generator: used to assign initial value from where user want to generate .**
- **@Column: When user want to add constraint on column that time this @ is used.**

- **@Email: used to compare email format**
- **@Range: Used to assign rang for number type of input.**

Eg. age@size: Used in collection

@length: Used to assign range like Minimum and Maximus for the string type data.

@NotNull: This gives manadatory filed to enter data.

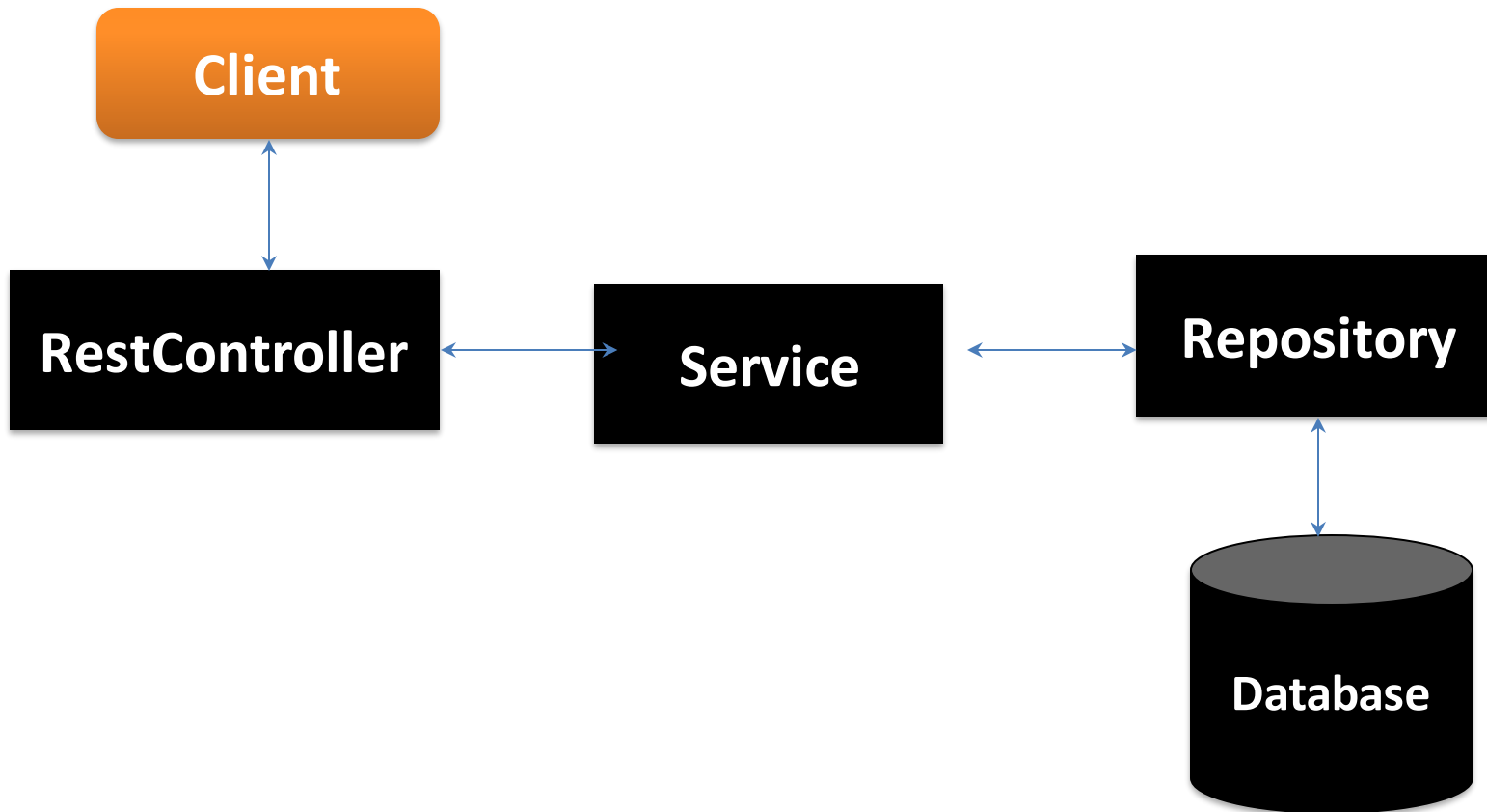
@Many To One:to create the many-to-one relationship between the Student and Address entities

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 –

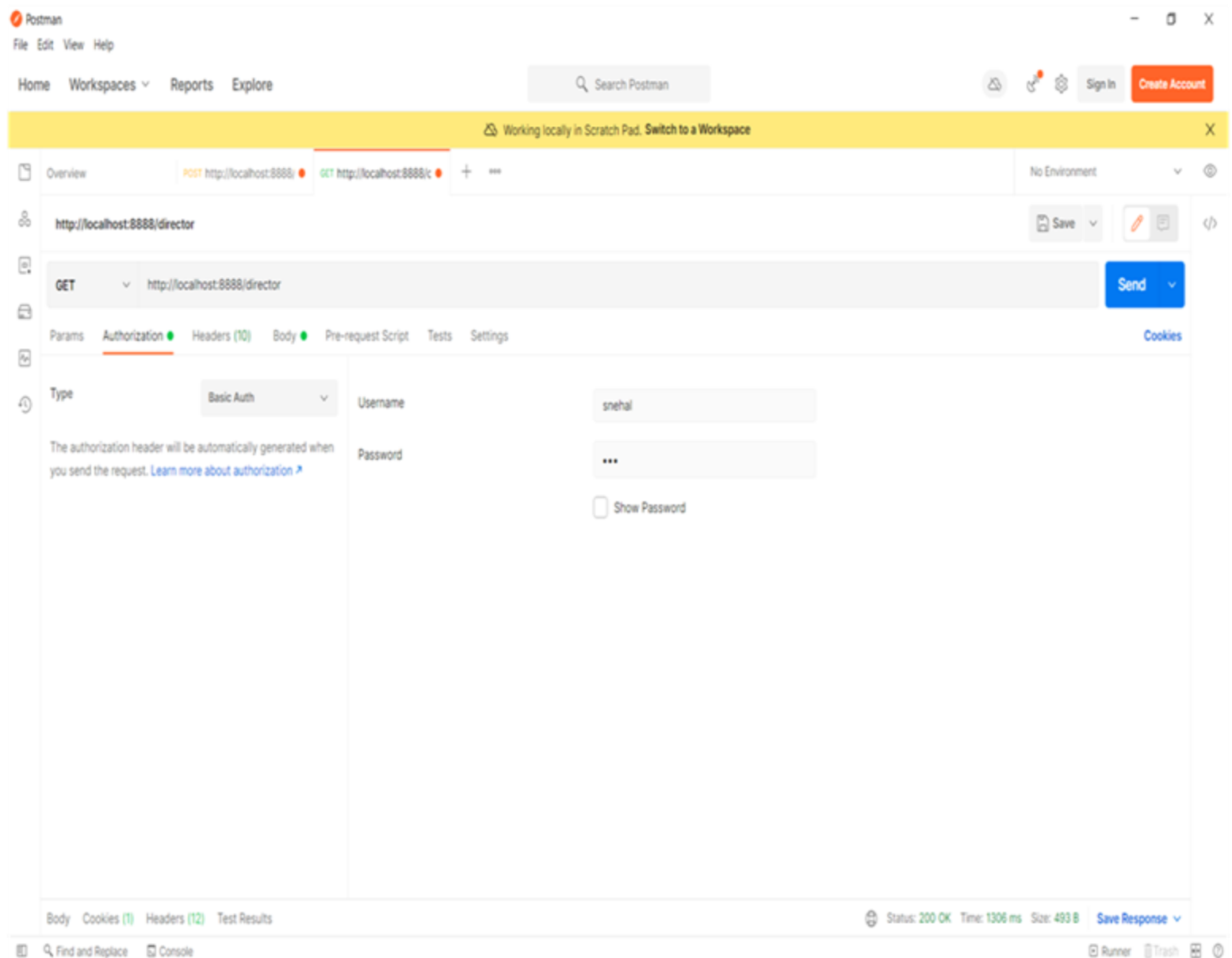


- ❑ **RestController** - In Spring Boot, the RestController class is responsible for processing incoming requests, preparing a model, and returning the view to be rendered as a response.
- ❑ **Service** - Components are the class file which contains @Service annotation. These class files are used to 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 screenshot shows the Postman interface with a GET request to `http://localhost:8888/director/All` executed successfully. The response is a JSON array containing two employee records.

Request Details:

- Method: GET
- URL: `http://localhost:8888/director/All`
- Body: none

Response Details:

- Status: 200 OK
- Time: 268 ms
- Size: 1.53 KB

Response Body (JSON):

```
[{"empId": 1, "empName": "rina", "empSalary": 32000.0, "empEmailId": "rina@gmail.com", "empage": 23, "empCellNo": "1234567899", "empAddress": "amravti", "empDesignation": "DIRECTOR", "empExperience": 3.0, "empUsername": "priya", "empPassword": "$2a$12$8NYa.gTmd0N14pwA/bybq.ZpdX5CwkCjw7f0Urwvz8Bg9zph/eQe6", "managerId": 0}, {"empId": 999, "empName": "snehal", "empSalary": 32000.0, "empEmailId": "snehal@gmail.com", "empAddress": "amravti"}]
```

3 Get Employee List: /director/{managerId}

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', along with a search bar and buttons for 'Sign In' and 'Create Account'. 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 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 object with the following fields:

```
1 {
2   "empId": 1002,
3   "empName": "sneha",
4   "empSalary": 230000.0,
5   "empEmailId": "sneha@gmail.com",
6   "empage": 26,
7   "empCellNo": "1789567891",
8   "empAddress": "nashik",
9   "empDesignation": "EMPLOYEE",
10  "empExperience": 2.0,
11  "empUsername": "sneha",
12  "empPassword": "$2a$10$3kxU/DyuNKHVqafo3Cc8BeH.MQcI1tYClzCSVfI7p7w10KLFWNLEm",
13  "managerId": 1000
14 }
15
16 }
```

The status bar at the bottom indicates 'Status: 200 OK', 'Time: 27 ms', and 'Size: 655 B'. There are also buttons for 'Save Response', 'Find and Replace', 'Console', 'Runner', and 'Trash'.

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 tabs (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 workspace shows 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

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.gTmd@N14pnA/bYbq.Zpdx5CwkCfw7i0UirWvz@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 displays the Postman application interface. At the top, the menu bar includes File, Edit, View, and Help. Below it, navigation tabs for Home, Workspaces, Reports, and Explore are visible, along with a search bar and buttons for Sign In and Create Account. A yellow banner indicates the user is working locally in Scratch Pad.

The main workspace shows a collection of requests. The selected request is a POST to `http://localhost:8888/director/addDept`. The request body is configured as raw JSON with the following content:

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

The response section at the bottom shows a status of 200 OK, a time of 96 ms, and a size of 415 B. The response body is displayed in a pretty-printed JSON format:

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

The bottom status bar includes options for Find and Replace, Console, Runner, and Trash.

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

The screenshot displays the Postman interface for a PUT request. The URL bar shows `http://localhost:8888/director/update/department/3`. The request body is a JSON object with the following structure:

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

The response is also a JSON object, displayed in the 'Body' tab:

```
{  "deptId": 3,  "deptName": "sci",  "deptLocation": "yavataml",  "employee": []}
```

The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 175 ms, and a size of 415 B.

7 delete department : /director/delete/department/{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 also present. A yellow banner at the top 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 'DELETE' method to the URL 'http://localhost:8888/director/delete/department/3'. The 'Body' tab is selected, showing a JSON payload:

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

. The 'Send' button is visible. Below the request details, there are tabs for 'Params', 'Authorization', 'Headers (10)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is active, and the 'JSON' format is selected. The response section at the bottom shows a status of '200 OK', a time of '90 ms', and a size of '377 B'. The response body is displayed in the 'Text' view, showing the message: '1 Record deleted succeseffully...!'. The bottom of the image 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

Save

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

Send

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 succeseffully...!

Find and Replace Console

Runner Trash

Type here to search

37°C 02:09 30-04-2022

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

The screenshot shows the Postman interface with a POST request to `http://localhost:8888/director/addEmp/2`. The request body is a JSON object with the following fields:

```
{  "empName": "prithvi",  "empSalary": 238000.0,  "empEmailId": "pri@gmail.com",  "empAge": 26,  "empCellNo": "1809567891",  "empAddress": "raygad",  "empDesignation": "EMPLOYEE",  "empExperience": 2.0,  "empUsername": "prithvi",  "empPassword": "$2a$10$VLm0PTtCJfa.0CB0HniczeA7ySt3hK0NncDH2ErC/4isJ3/1a6rKi",  "managerId": 1000}
```

The response status is 200 OK, with a time of 498 ms and a size of 655 B. The response body is a JSON object with the following fields:

```
{  "empId": 1051,  "empName": "prithvi",  "empSalary": 238000.0,  "empEmailId": "pri@gmail.com",  "empAge": 26,  "empCellNo": "1809567891",  "empAddress": "raygad",  "empDesignation": "EMPLOYEE",  "empExperience": 2.0,  "empUsername": "prithvi",  "empPassword": "$2a$10$VLm0PTtCJfa.0CB0HniczeA7ySt3hK0NncDH2ErC/4isJ3/1a6rKi",  "managerId": 1000}
```

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

The screenshot shows the Postman application interface. At the top, there's a menu bar with 'Postman', 'File', 'Edit', 'View', and 'Help'. Below it, a navigation bar includes 'Home', 'Workspaces', 'Reports', and 'Explore', along with a search bar and buttons for 'Sign In' and 'Create Account'. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main workspace shows a PUT request to 'http://localhost:8888/director/updateEmp/1051'. The request body is in JSON format, containing employee details. The response is also in JSON format, showing the updated employee data. The status is '200 OK'.

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

http://localhost:8888/director/updateEmp/1051

PUT | http://localhost:8888/director/updateEmp/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   "empId": 1051,
3   "empName": "shivaji",
4   "empSalary": 238000.0,
5   "empEmailId": "pri@gamil.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.0CB0HnrczeA7ySt3hKNWncDKzErC/4is33/la6rKi",
13  "managerId": 1000
14 }
```

Body | Cookies (1) | Headers (11) | Test Results

Status: 200 OK | Time: 52 ms | Size: 655 B | Save Response

Pretty | Raw | Preview | Visualize | JSON

```
1 {
2   "empId": 1051,
3   "empName": "shivaji",
4   "empSalary": 238000.0,
5   "empEmailId": "pri@gamil.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.0CB0HnrczeA7ySt3hKNWncDKzErC/4is33/la6rKi",
13  "managerId": 1000
14 }
```

Find and Replace | Console | Runner | Trash

Type here to search

37°C | 02:17 | 30-04-2022

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

The screenshot displays the Postman application interface. At the top, the menu bar includes 'File', 'Edit', 'View', and 'Help'. Below it, the navigation bar shows 'Home', 'Workspaces', 'Reports', and 'Explore'. A search bar labeled 'Search Postman' is positioned 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 DELETE method to the URL 'http://localhost:8888/director/delete/1051'. The request body is empty. The response is displayed in the bottom panel, showing a status of '200 OK', a time of '39 ms', and a 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.

11 Updateself : /director/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 interface shows a PUT request to 'http://localhost:8888/director/updateself/999'. The request body is a JSON object with the following fields: 'empId' (999), 'empName' ('shivaaji'), 'empSalary' (230000.0), 'empEmailId' ('pri@gamil.com'), 'empage' (23), 'empCellNo' ('1809567891'), 'empAddress' ('raygad'), 'empDesignation' ('DIRECTOR'), 'empExperience' (2.0), 'empUsername' ('prithvi'), 'empPassword' ('123'), and 'managerId' (@). The response status is '200 OK' with a time of '52 ms' and size of '594 B'. The response body is displayed in 'Pretty' format, showing the same JSON object. The Windows taskbar at the bottom shows the date as '02-27 30-04-2022' and the temperature as '37°C'.

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

http://localhost:8888/director/updateself/999

PUT http://localhost:8888/director/updateself/999

Send

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: 52 ms Size: 594 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "empId": 999,
3   "empName": "shivaaji",
4   "empSalary": 230000.0,
5   "empEmailId": "pri@gamil.com",
6   "empage": 23,
7   "empCellNo": "1809567891",
8   "empAddress": "raygad",
9   "empDesignation": "DIRECTOR",
10  "empExperience": 2.0,
11  "empUsername": "prithvi",
12  "empPassword": "123",
13  "managerId": @
14 }
```

13 Date :/director/getByDate/2022-05-01

Postman

File Edit View Help

Home Workspaces ▾ Reports Explore

Search Postman

Working locally in Scratch Pad. [Switch to a Workspace](#)

Overview POST http://localhost:8888/ GET http://localhost:8880/c + ...

http://localhost:8880/director/getByDate/2022-05-01

GET http://localhost:8880/director/getByDate/2022-05-01

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

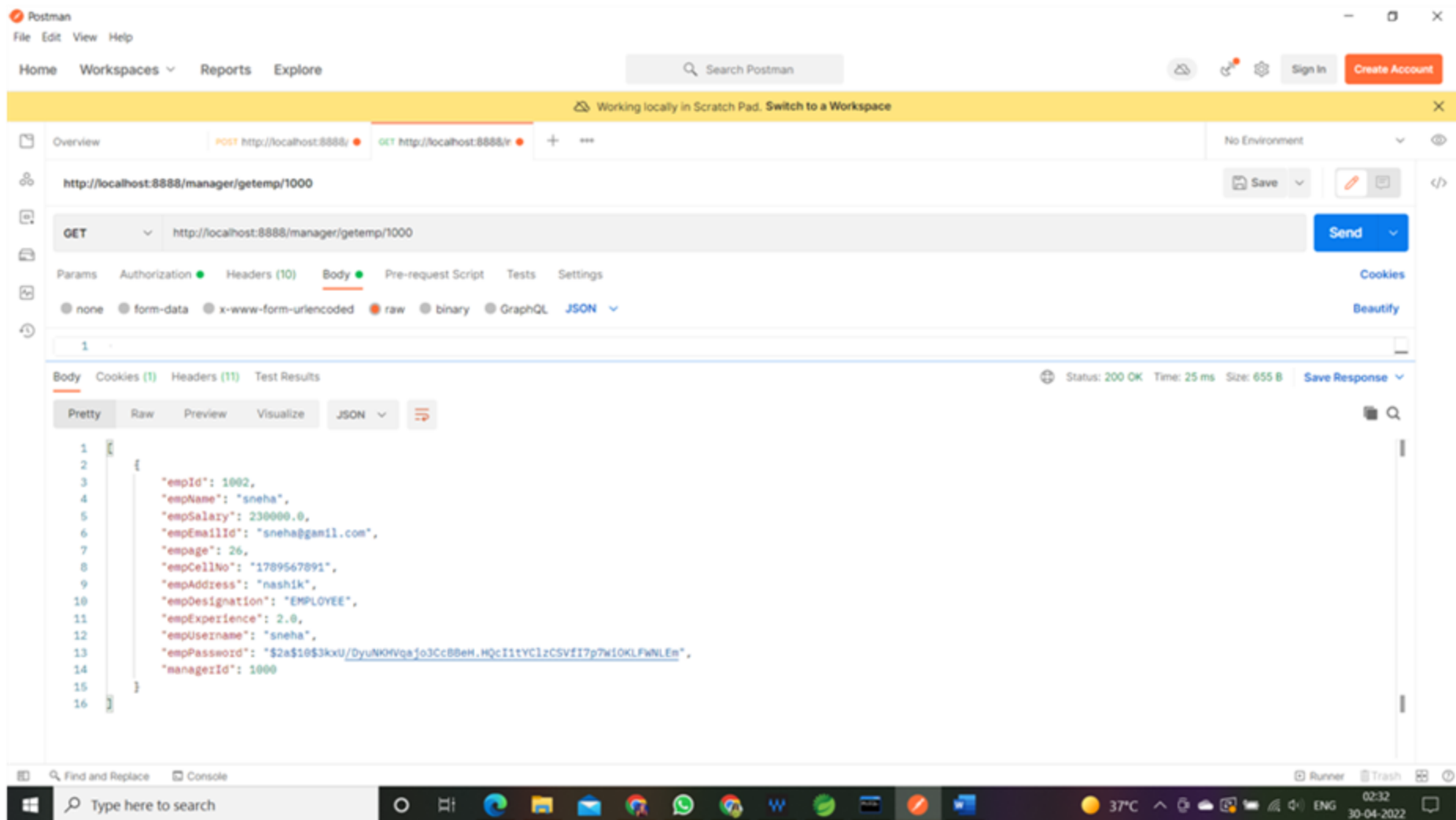
Body Cookies (1) Headers (11) Test Results Status: 200 OK 1

Pretty Raw Preview Visualize JSON ▾

```
1 [
2   {
3     "empId": 1000,
4     "empName": "pratibha",
5     "empSalary": 23000.0,
6     "empEmailId": "pratibha@gmail.com",
7     "empage": 24,
8     "empCellNo": "1254567891",
9     "empAddress": "nashik",
10    "empDesignation": "MANAGER",
11    "empExperience": 4.0,
12    "empUsername": "pratibha",
13    "empPassword": "$2a$10$pZvmjKgDyUndiV11b3R6ze0V90EitwDec0.iRpU6W/btauUyXD/0y",
14    "managerId": 999,
15    "joiningDate": "2022-05-01"
16  },
17  {
18    "empId": 1003,
19    "empName": "harshita",
20    "empSalary": 23000.0,
21    "empEmailId": "har@gmail.com",
22    "empage": 24,
```

Find and Replace Console

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



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 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/manager/getemp/1000'. The 'Body' tab is selected, showing a JSON response. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 25 ms', and 'Size: 655 B'. The Windows taskbar is visible at the very bottom.

Overview **POST** http://localhost:8888/ **GET** http://localhost:8888/ No Environment

http://localhost:8888/manager/getemp/1000 Save

GET http://localhost:8888/manager/getemp/1000 Send

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

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

1

Body Cookies (1) Headers (11) Test Results Status: 200 OK Time: 25 ms Size: 655 B Save Response

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "empId": 1002,
3   "empName": "sneha",
4   "empSalary": 230000.0,
5   "empEmailId": "sneha@gmail.com",
6   "empAge": 26,
7   "empCellNo": "1789567891",
8   "empAddress": "nashik",
9   "empDesignation": "EMPLOYEE",
10  "empExperience": 2.0,
11  "empUsername": "sneha",
12  "empPassword": "$2a$10$3kxU/DyuNQHvGaJo3CcB8eH.MQcI1tYClzCSvfI7p7w1OKLFWNlEm",
13  "managerId": 1000
14 }
15
16
```

Find and Replace Console Runner Trash

Type here to search 37°C 02:32 30-04-2022

14 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 is present on the right. A yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main area shows a REST client setup for a POST request to 'http://localhost:8888/manager/2'. The 'Body' tab is selected, showing a JSON payload. The 'Send' button is visible. Below the request, the response is displayed in the 'Body' tab, showing a successful status (200 OK) and a JSON response. The response contains employee details such as empId, empName, empSalary, empEmailId, empAge, empCellNo, empAddress, empDesignation, empExperience, empUsername, empPassword, and managerId.

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 Beautify

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

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

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

1

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": 1052,
3   "empName": "sai",
4   "empSalary": 230000.0,
5   "empEmailId": "sai@gmail.com",
6   "empAge": 26,
7   "empCellNo": "1809587891",
8   "empAddress": "piratapgad",
9   "empDesignation": "EMPLOYEE",
10  "empExperience": 2.0,
11  "empUsername": "sai",
12  "empPassword": "$2a$10$yaHejLREm1EPvDAB33sPI.dmvZYMQYBz1j9Trv1NIMXgZs0GjNafa",
13  "managerId": 1000
14 }
```

Find and Replace Console Runner Trash

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

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 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 states 'Working locally in Scratch Pad. Switch to a Workspace'. The main workspace shows a PUT request to 'http://localhost:8888/manager/update/1052'. The 'Body' tab is selected, showing a JSON payload with employee details. The response status is '200 OK' with a time of '59 ms' and size of '653 B'. The bottom of the screen shows a Windows taskbar with various application icons and a system tray with the date '30-04-2022' and time '02:43'.

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/

No Environment

Save

PUT http://localhost:8888/manager/update/1052

Send

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: 59 ms Size: 653 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "empId": 1052,
3   "empName": "soyara",
4   "empSalary": 230000.0,
5   "empEmailId": "sai@gamil.com",
6   "empage": 26,
7   "empCellNo": "1809587891",
8   "empAddress": "piratapgad",
9   "empDesignation": "EMPLOYEE",
10  "empExperience": 2.0,
11  "empUsername": "sai",
12  "empPassword": "$2a$10$yaHejLREm1EPvDABJ3sPI.dmvZYMQYBz1j9TrV1NIXKgZs0GjNafa",
13  "managerId": 1000
14 }
```

Find and Replace Console

Runner Trash

Type here to search

37°C

02:43 30-04-2022

16 UpdateSelf : /manager/updateself/{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 area displays a PUT request to 'http://localhost:8888/manager/updateself/1000'. The 'Body' tab is selected, showing a JSON payload. The response status is '200 OK' with a time of '226 ms' and a size of '648 B'. The response body is displayed in the 'Body' tab, showing a JSON object with employee details.

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/

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

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$10$uTFZZUu9WcgjXevzFij@iue6ZkwQIbdhZ5vjoZLmpK1gk3TjdZMd3m",
13  "managerId": 999
14 }
```

Find and Replace Console

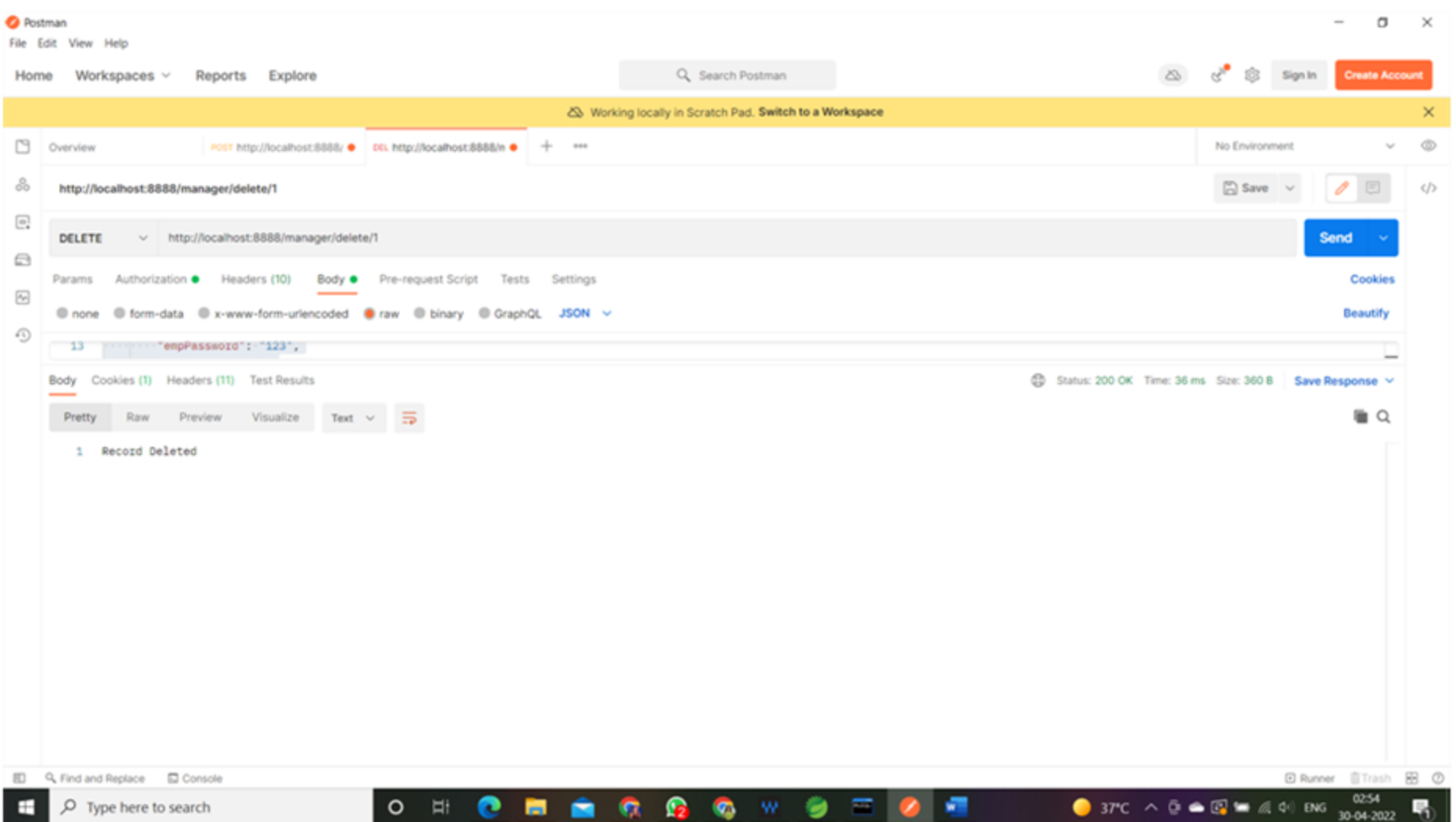
Runner Trash

Type here to search

37°C

02:50
30-04-2022

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



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

The screenshot shows the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it, navigation tabs include 'Home', '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 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 bottom section shows the response details, including status '200 OK', time '178 ms', and size '650 B'. The response body is displayed in a 'Pretty' JSON format.

Overview POST http://localhost:8888/ PUT http://localhost:8888/ + --- No Environment

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

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

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

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

Body Cookies (1) Headers (11) Test Results Status: 200 OK Time: 178 ms Size: 650 B Save Response

Pretty Raw Preview Visualize JSON

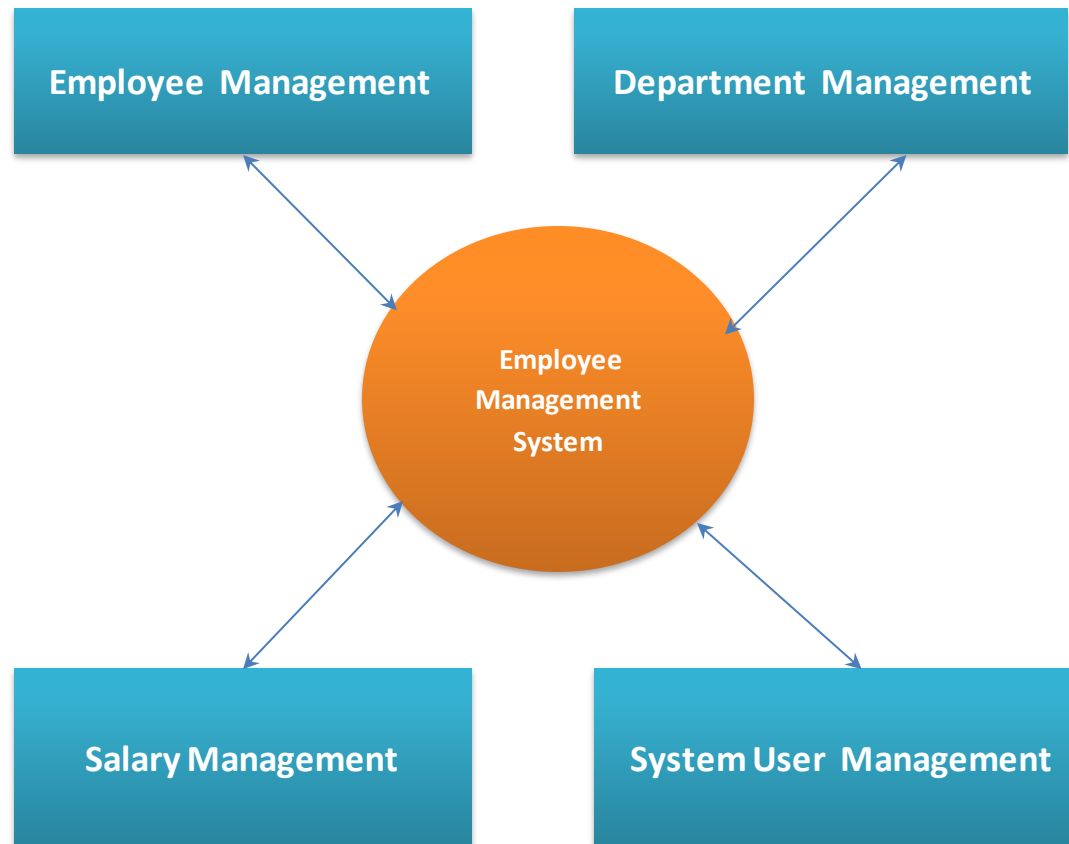
```
1 {
2   "empId": 1002,
3   "empName": "sakshi",
4   "empSalary": 230000.0,
5   "empEmailId": "sara@gmail.com",
6   "empage": 26,
7   "empCellNo": "1848587891",
8   "empAddress": "lonar",
9   "empDesignation": "EMPLOYEE",
10  "empExperience": 2.0,
11  "empUsername": "sara",
12  "empPassword": "$2a$10$LnDyRphKtspu0QRi687yC.de8ACpKJ/WH8xQUnhDexv/PYfssk8v6",
13  "managerId": 1000
14 }
```

Find and Replace Console Runner Trash

Type here to search 37°C 03:02 30-04-2022

Diagrams

Data Flow Diagram:



DFD – Employee Management System

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.