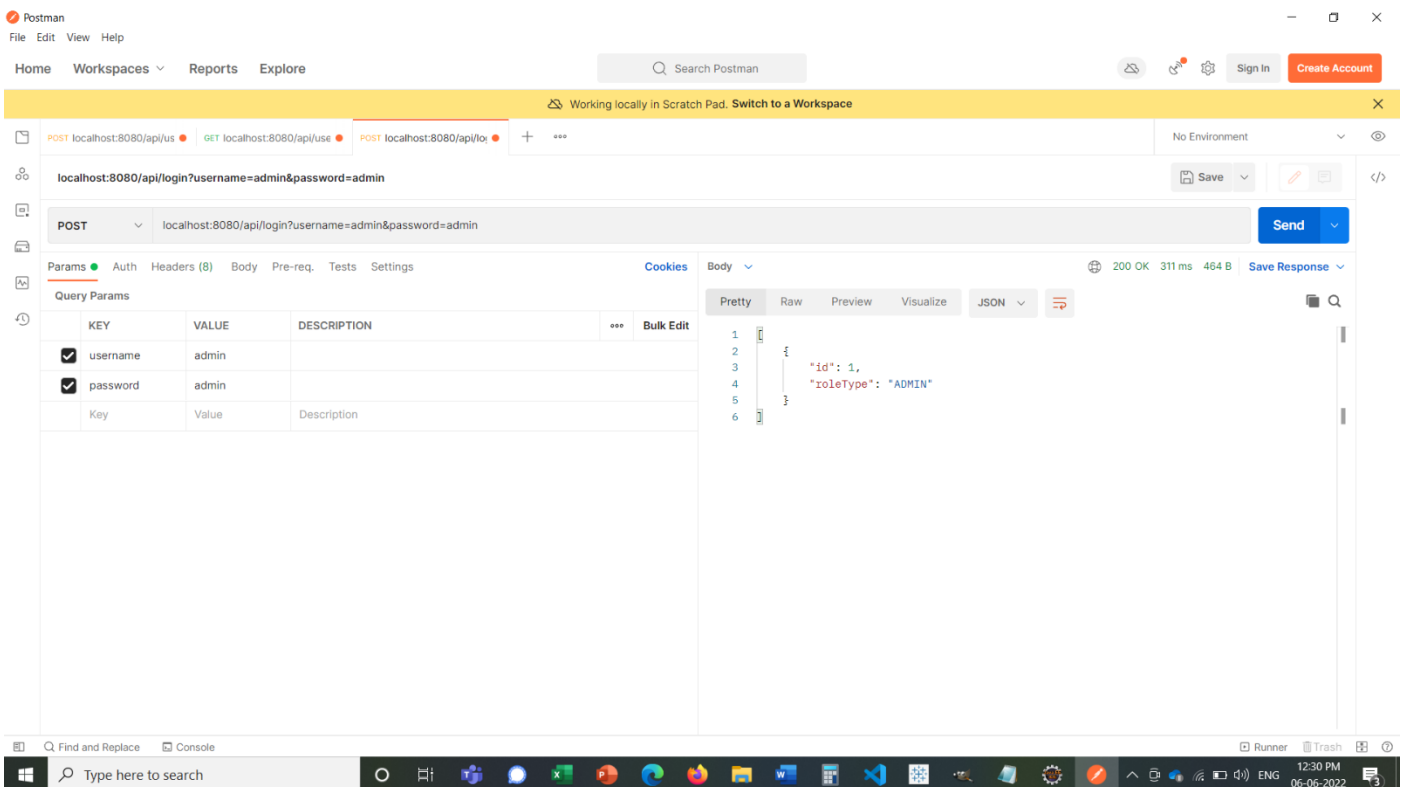
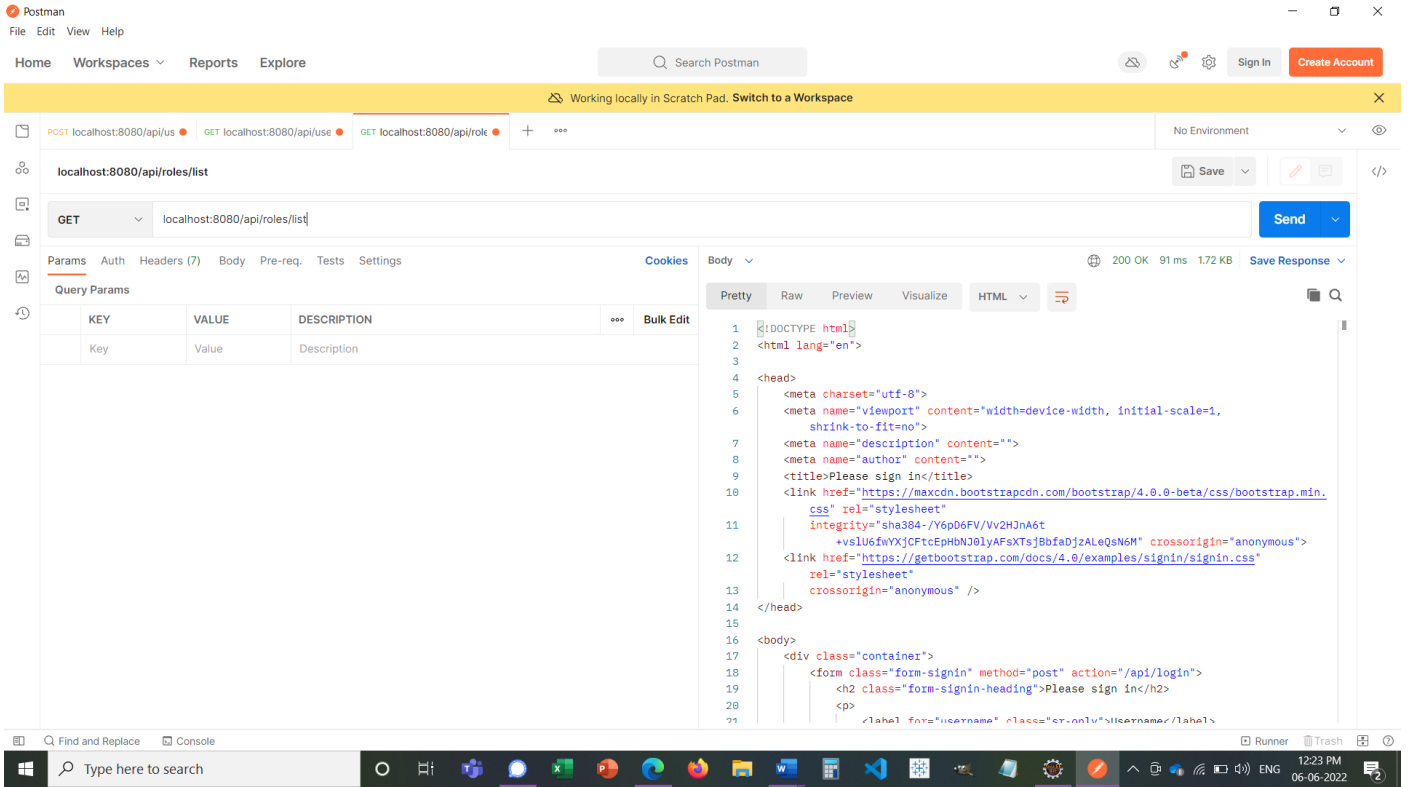


Screenshots of the Rest Application:

A) Accessing the URL to get a list of Roles <http://localhost:8080/api/roles/list>
It redirects to a login page. We login with “ADMIN” credentials using the URL Pasted below.
<http://localhost:8080/api/login>
Once successfully logged in, we see the list of roles as the response returned.



B) Add the “USER” role to the roles table

The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu items (File, Edit, View, Help), and navigation links (Home, Workspaces, Reports, Explore). A search bar and utility buttons (Sign In, Create Account) are also present. The main workspace displays a POST request to `localhost:8080/api/roles/add`. The request body is set to JSON and contains the following content:

```
1 {
2   "roleType": "USER"
3 }
```

The right-hand pane shows the response status as 200 OK, with a response time of 180 ms and a body size of 394 B. The response body is currently empty.

B.1) List all the roles to see the USER role added.

The screenshot shows the Postman application interface with a GET request to `localhost:8080/api/roles/list`. The response status is 200 OK, with a response time of 15 ms and a body size of 491 B. The response body is displayed in JSON format:

```
1 [
2   {
3     "id": 1,
4     "roleType": "ADMIN"
5   },
6   {
7     "id": 2,
8     "roleType": "USER"
9   }
10 ]
```

The left-hand pane shows the Params tab with a table for Query Params:

KEY	VALUE	DESCRIPTION
Key	Value	Description

C) Add a new user with “USER” role to the users table

The screenshot shows the Postman interface with a POST request to `localhost:8080/api/users/add/USER`. The request body is a JSON object:

```
{  "username": "user",  "password": "$2a$12$eCjm.rKIQmZuzVjub.Q4dez0BBB1FYpTQ4D4oSgCnf8LctbSrCrE2"}
```

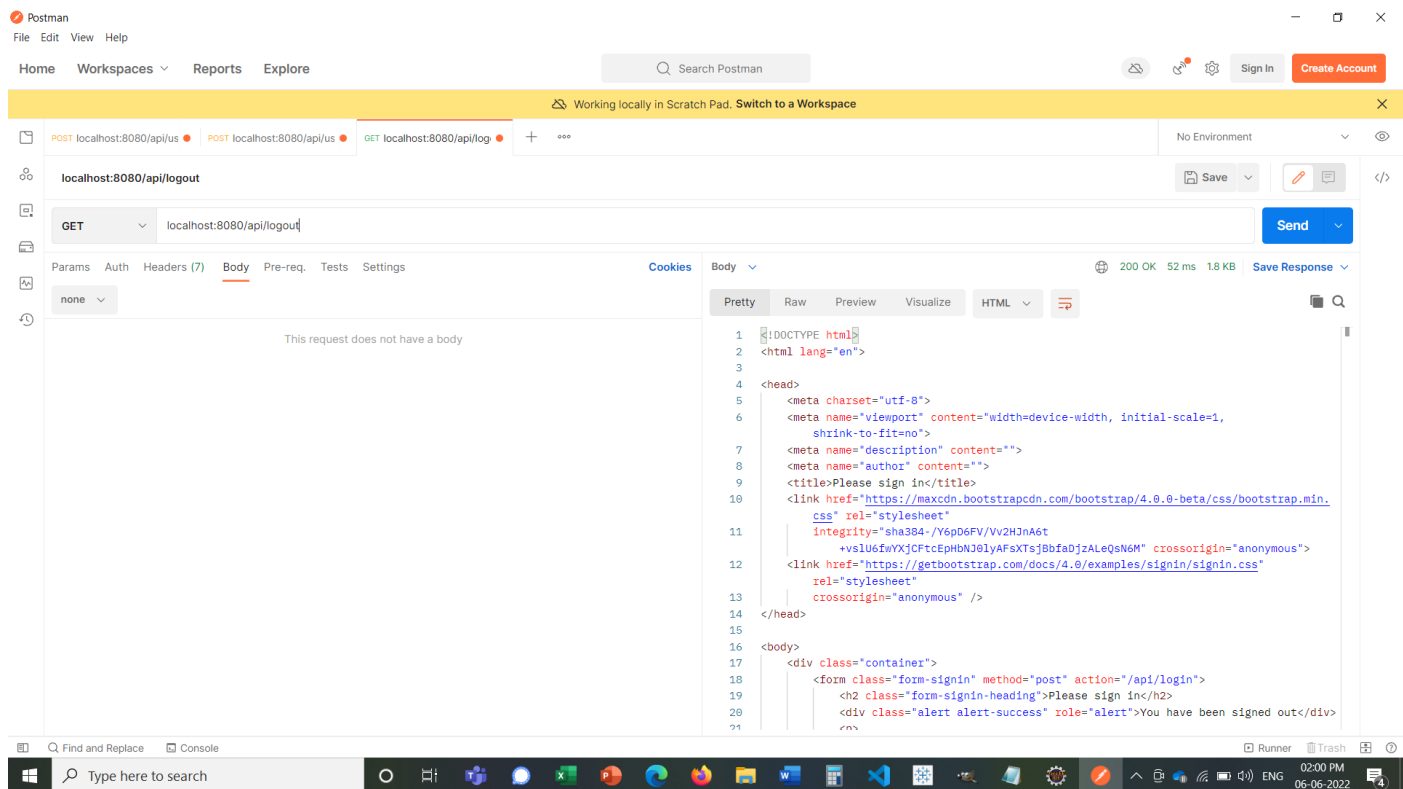
The response status is 200 OK, with a response time of 258 ms and a body size of 394 B. The response body is empty.

C.1) List all users to confirm that user with USER role has been added

The screenshot shows the Postman interface with a GET request to `localhost:8080/api/users/list`. The response status is 200 OK, with a response time of 19 ms and a body size of 722 B. The response body is a JSON array of two user objects:

```
[  {    "userId": 1,    "username": "admin",    "password": "$2a$12$XT80go.mJ2zia3M37AfeIe90Bb2F/BCtgrFa3BvXsYY6Vs.X3RmWm",    "roles": [      {        "id": 1,        "roleType": "ADMIN"      }    ]  },  {    "userId": 2,    "username": "user",    "password": "$2a$12$eCjm.rKIQmZuzVjub.Q4dez0BBB1FYpTQ4D4oSgCnf8LctbSrCrE2",    "roles": [      {        "id": 2,        "roleType": "USER"      }    ]  }]
```

D) Logout and then login with user having USER role

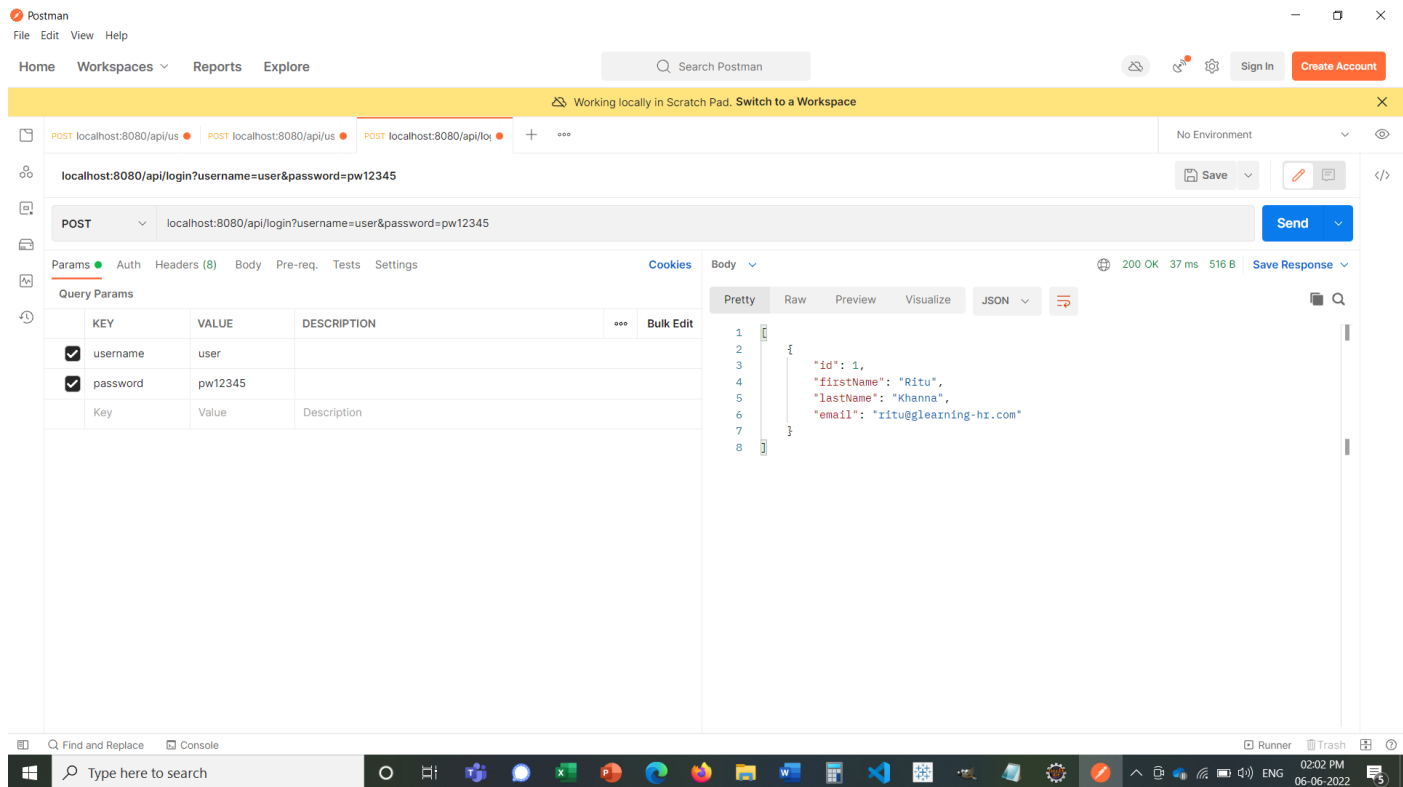


Postman interface showing a GET request to `localhost:8080/api/logout`. The response is an HTML page with a login form and a success message.

Request: `GET localhost:8080/api/logout`

Response (HTML):

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1,
7     shrink-to-fit=no">
8   <meta name="description" content="">
9   <meta name="author" content="">
10  <title>Please sign in</title>
11  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.
12    css" rel="stylesheet"
13    integrity="sha384-/Y6pD6FV/Vv2H3nA6t
14    +vsIU6fWVXJCFtcEpHbNJelyAFsXTsJB0fa0jzALeQsN6M" crossorigin="anonymous">
15  <link href="https://getbootstrap.com/docs/4.0/examples/signin/signin.css"
16    rel="stylesheet"
17    crossorigin="anonymous" />
18 </head>
19 <body>
20   <div class="container">
21     <form class="form-signin" method="post" action="/api/login">
22       <h2 class="form-signin-heading">Please sign in</h2>
23       <div class="alert alert-success role="alert">You have been signed out</div>
```



Postman interface showing a POST request to `localhost:8080/api/login?username=user&password=pw12345`. The response is a JSON object with user details.

Request: `POST localhost:8080/api/login?username=user&password=pw12345`

Response (JSON):

```
1 {
2   "id": 1,
3   "firstName": "Ritu",
4   "lastName": "Khanna",
5   "email": "ritu@glearning-hr.com"
6 }
7
8
```

E) The user cannot access roles table or users table. But the user can access employees table.
User can only perform read operations on the employees table.
User cannot add, update or delete an employee record

User to access roles table

The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu items (File, Edit, View, Help), and navigation links (Home, Workspaces, Reports, Explore). A search bar and utility buttons (Sign In, Create Account) are also present. The main workspace displays a collection of requests. The selected request is a GET request to `localhost:8080/api/roles/list`. The response status is `403 Forbidden` with a response time of `90 ms` and a body size of `498 B`. The response body, viewed in 'Pretty' format, contains the message: `1 Hi user, you do not have permission to access this page!`. The left sidebar shows the 'Query Params' section with a table structure.

KEY	VALUE	DESCRIPTION
Key	Value	Description

User to access users table

The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu items (File, Edit, View, Help), and navigation links (Home, Workspaces, Reports, Explore). A search bar and utility buttons (Sign In, Create Account) are also present. The main workspace displays a collection of requests. The selected request is a GET request to `localhost:8080/api/users/list`. The response status is `403 Forbidden` with a response time of `13 ms` and a body size of `498 B`. The response body, viewed in 'Pretty' format, contains the message: `1 Hi user, you do not have permission to access this page!`. The left sidebar shows the 'Query Params' section with a table structure.

KEY	VALUE	DESCRIPTION
Key	Value	Description

User to access the employees table

The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu items (File, Edit, View, Help), and navigation links (Home, Workspaces, Reports, Explore). A search bar and user options (Sign In, Create Account) are also present. The main workspace displays a GET request to `localhost:8080/api/employees/list`. The response is a JSON object representing an employee:

```
{
  "id": 1,
  "firstName": "Ritu",
  "lastName": "Khanna",
  "email": "ritu@glearning-hr.com"
}
```

The bottom status bar shows the request was successful (200 OK) with a response time of 19 ms and a body size of 516 B.

User to perform search operation on the employees table using firstName

The screenshot shows the Postman application interface with a GET request to `localhost:8080/api/employees/search/firstName/Ritu`. The response is the same JSON object as in the previous screenshot:

```
{
  "id": 1,
  "firstName": "Ritu",
  "lastName": "Khanna",
  "email": "ritu@glearning-hr.com"
}
```

The bottom status bar shows the request was successful (200 OK) with a response time of 24 ms and a body size of 516 B.

User to add employee record

Postman interface showing a POST request to `localhost:8080/api/employees/add`. The request body is JSON:

```
{  "firstName": "Amar",  "lastName": "Sharma",  "email": "amar@glearning-faculty.com"}
```

The response is `403 Forbidden` with the message: `1 Hi user, you do not have permission to access this page!`

F) Login with ADMIN role credentials to perform CRUD operations on employees table.

Postman interface showing a POST request to `localhost:8080/api/login?username=admin&password=admin`. The request body is JSON:

```
{  "id": 1,  "firstName": "Ritu",  "lastName": "Khanna",  "email": "ritu@glearning-hr.com"}
```

The response is `200 OK` with the message: `1 Hi user, you do not have permission to access this page!`

Admin to add an employee record

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Reports', and 'Explore'. Below it, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main workspace shows a collection of requests, with the selected one being a POST request to 'localhost:8080/api/employees/add'. The request body is set to 'JSON' and contains the following data:

```
1 {
2   "firstName": "Amar",
3   "lastName": "Sharma",
4   "email": "amar@glearning-faculty.com"
5 }
```

The response is displayed on the right, showing a status of '200 OK' with a response time of '20 ms' and a body of '1'. The bottom status bar shows the system clock as 02:30 PM on 06-06-2022.

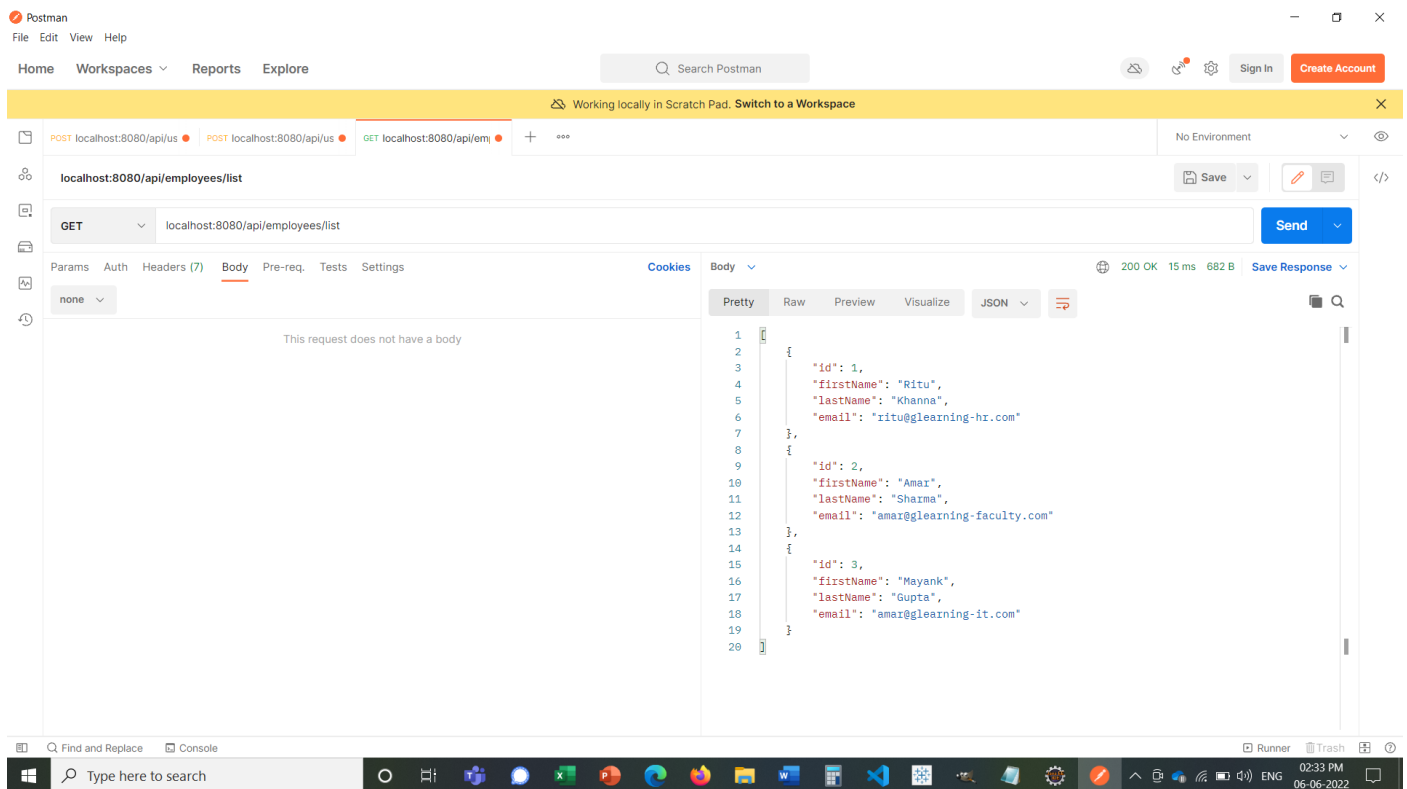
Admin to add one more employee record

The screenshot shows the Postman application interface, similar to the first one. The selected request is a POST to 'localhost:8080/api/employees/add' with a JSON body:

```
1 {
2   "firstName": "Mayank",
3   "lastName": "Gupta",
4   "email": "amar@glearning-it.com"
5 }
```

The response on the right shows '200 OK' with a response time of '13 ms' and a body of '1'. The bottom status bar shows the system clock as 02:31 PM on 06-06-2022.

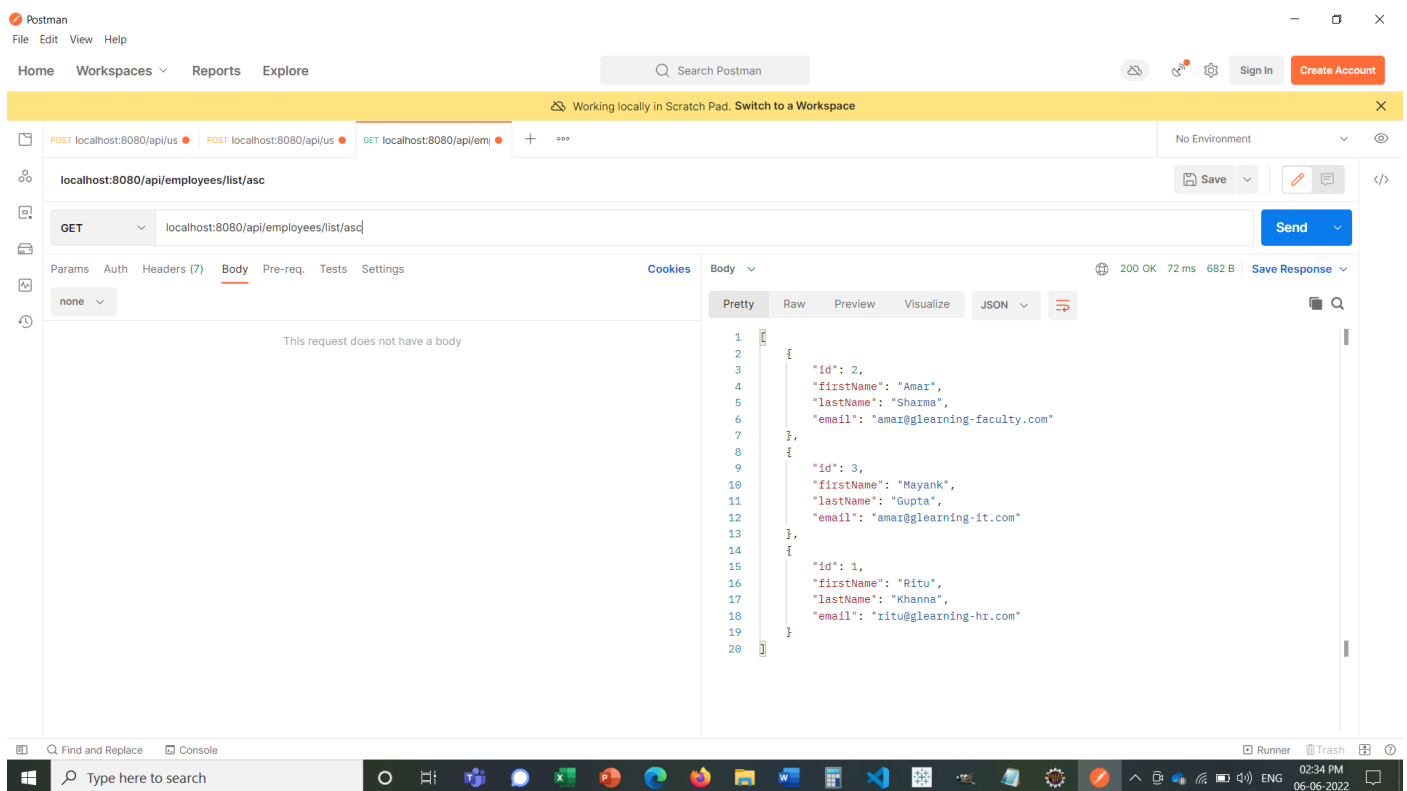
Admin to list all employee records to verify the added employees



Postman interface showing a GET request to `localhost:8080/api/employees/list`. The response is a 200 OK status with a JSON body containing three employee records:

```
1 {
2   {
3     "id": 1,
4     "firstName": "Ritu",
5     "lastName": "Khanna",
6     "email": "ritu@glearning-hr.com"
7   },
8   {
9     "id": 2,
10    "firstName": "Amar",
11    "lastName": "Sharma",
12    "email": "amar@glearning-faculty.com"
13  },
14  {
15    "id": 3,
16    "firstName": "Mayank",
17    "lastName": "Gupta",
18    "email": "amar@glearning-it.com"
19  }
20 }
```

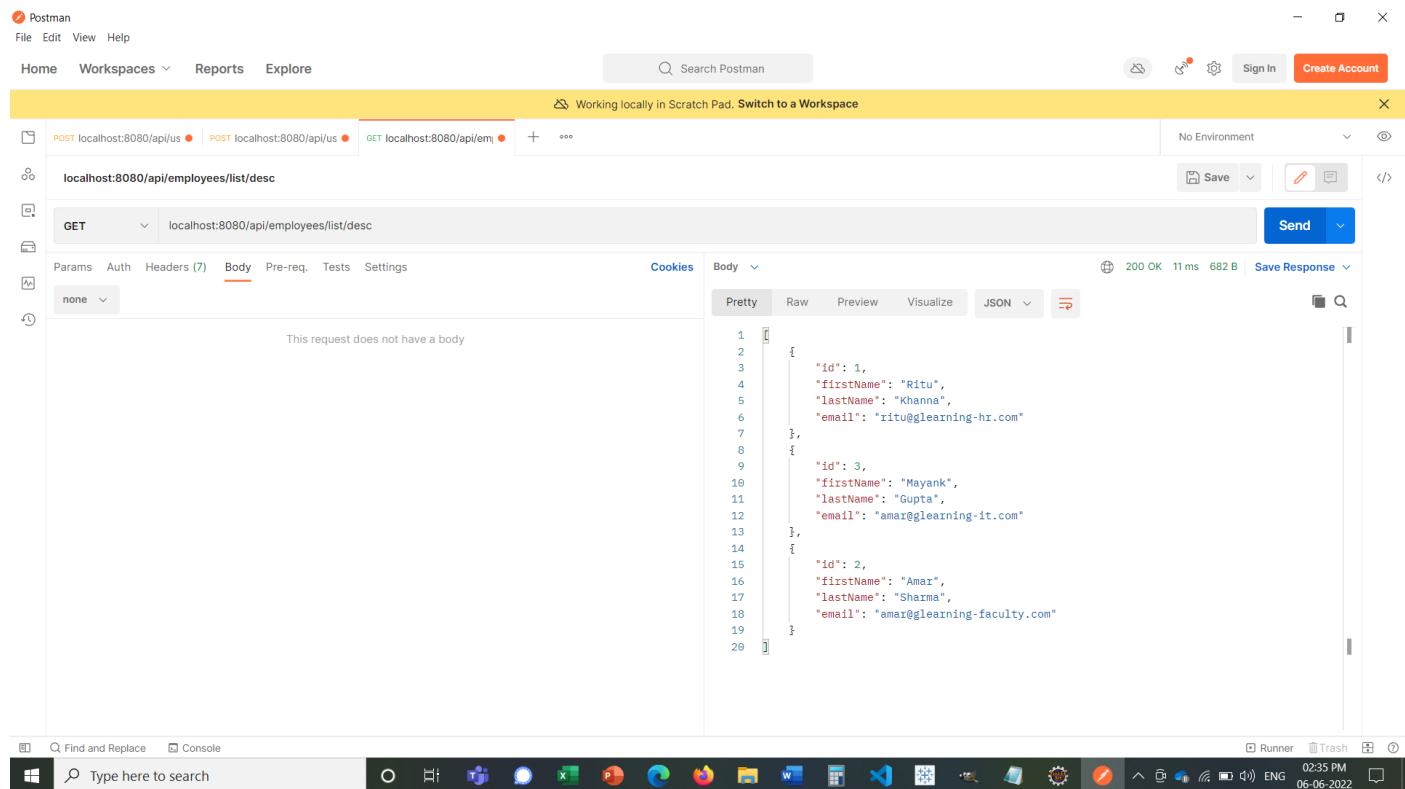
Admin to list all employees in ascending order



Postman interface showing a GET request to `localhost:8080/api/employees/list/asc`. The response is a 200 OK status with a JSON body containing three employee records sorted by ID in ascending order:

```
1 {
2   {
3     "id": 2,
4     "firstName": "Amar",
5     "lastName": "Sharma",
6     "email": "amar@glearning-faculty.com"
7   },
8   {
9     "id": 3,
10    "firstName": "Mayank",
11    "lastName": "Gupta",
12    "email": "amar@glearning-it.com"
13  },
14  {
15    "id": 1,
16    "firstName": "Ritu",
17    "lastName": "Khanna",
18    "email": "ritu@glearning-hr.com"
19  }
20 }
```

Admin to list all employees in descending order

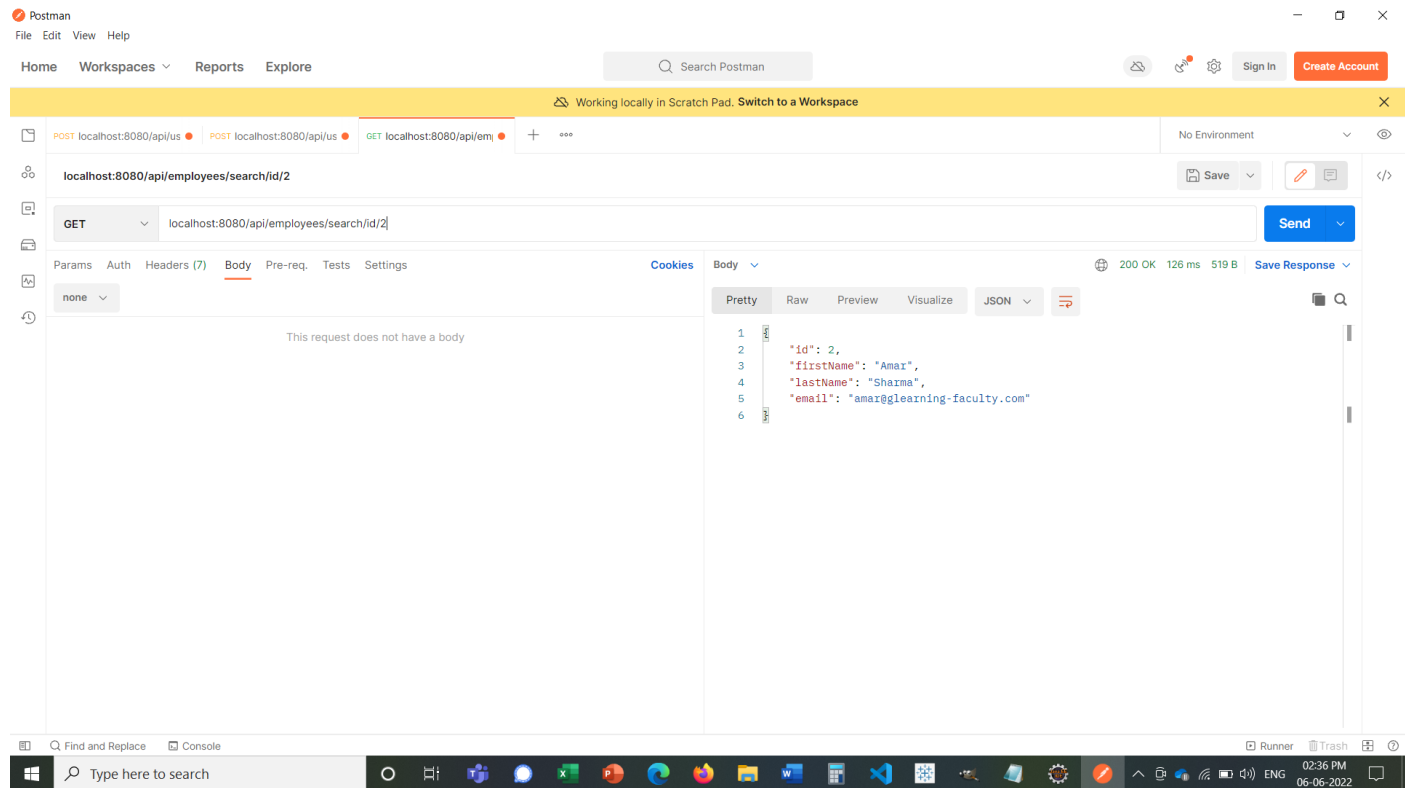


Postman interface showing a GET request to `localhost:8080/api/employees/list/desc`. The response is a JSON array of three employees, sorted by ID in descending order.

```
200 OK 11 ms 682 B
```

```
[{"id": 1, "firstName": "Ritu", "lastName": "Khanna", "email": "ritu@glearning-hr.com"}, {"id": 3, "firstName": "Mayank", "lastName": "Gupta", "email": "amar@glearning-it.com"}, {"id": 2, "firstName": "Amar", "lastName": "Sharma", "email": "amar@glearning-faculty.com"}]
```

Admin to search employee by Id = 2



Postman interface showing a GET request to `localhost:8080/api/employees/search/id/2`. The response is a JSON object representing the employee with ID 2.

```
200 OK 126 ms 519 B
```

```
{"id": 2, "firstName": "Amar", "lastName": "Sharma", "email": "amar@glearning-faculty.com"}
```

Admin to search by firstName = Mayank

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Reports', and 'Explore'. Below it, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The main workspace shows a GET request to 'localhost:8080/api/employees/search/firstName/Mayank'. The 'Body' tab is selected, displaying a JSON response in 'Pretty' format:

```
{  "id": 3,  "firstName": "Mayank",  "lastName": "Gupta",  "email": "amar@glearning-it.com"}
```

. The status bar at the bottom shows '200 OK', '17 ms', and '517 B'.

Admin to update email id of the employee id=3 from amar@glearning-it.com to mayank@glearning-it.com.

The screenshot shows the Postman application interface for a PUT request. The request is to 'localhost:8080/api/employees/update'. The 'Body' tab is selected, showing a JSON body in 'Raw' format:

```
{  "id": 3,  "firstName": "Mayank",  "lastName": "Gupta",  "email": "mayank@glearning-it.com"}
```

. The status bar at the bottom shows '200 OK', '139 ms', and '394 B'.

Admin to search the employee by id=3 to verify the update

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Reports', and 'Explore'. Below it, 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 'localhost:8080/api/employees/search/id/3'. The 'Body' tab is active, showing a JSON response in 'Pretty' format:

```
1 {
2   "id": 3,
3   "firstName": "Mayank",
4   "lastName": "Gupta",
5   "email": "mayank@learning-it.com"
6 }
```

The status bar at the bottom indicates a 200 OK response with 13 ms latency and 517 B of data. The Windows taskbar at the bottom shows the time as 02:47 PM on 06-06-2022.

Admin to delete the employee with id=1

The screenshot shows the Postman application interface. The selected request is a DELETE request to 'localhost:8080/api/employees/delete/1'. The 'Body' tab is active, showing a response in 'Text' format:

```
1
```

The status bar at the bottom indicates a 200 OK response with 55 ms latency and 394 B of data. The Windows taskbar at the bottom shows the time as 02:47 PM on 06-06-2022.

Admin to list all the employees to confirm the delete

Postman
File Edit View Help

Home Workspaces Reports Explore

Search Postman

Working locally in Scratch Pad. Switch to a Workspace

POST localhost:8080/api/us POST localhost:8080/api/us GET localhost:8080/api/em/ + ... No Environment

localhost:8080/api/employees/list Save

GET localhost:8080/api/employees/list Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

none

This request does not have a body

200 OK 10 ms 604 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 2,
4     "firstName": "Amar",
5     "lastName": "Sharma",
6     "email": "amar@glearning-faculty.com"
7   },
8   {
9     "id": 3,
10    "firstName": "Mayank",
11    "lastName": "Gupta",
12    "email": "mayank@glearning-it.com"
13  }
14 }
```

Find and Replace Console

Runner Trash

Type here to search

02:48 PM 06-06-2022