

## Database

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists several databases: cluster0.4wh1dmz.mongodb.net, admin, comp3123\_assignment1, comp3123\_assignment2 (selected), employees, users, config, local, sample\_mflix, taskapp, test, cluster0.t4hxj0z.mongodb.net, cluster0.ztfe6cp.mongodb.net, and localhost:27017. A message at the bottom left indicates 'Compass 1.48.2 installed successfully'.

The main area displays the 'users' collection from the 'comp3123\_assignment2' database. It shows one document with the following fields:

```
_id: ObjectId('692b987f4f1831db7d5f6954')
username : "arash"
email : "arash123@gmail.com"
password : "$2a$10$82W23I51qwrzPgF55LlnR02EhDyt01ecboujJZzjFj6vzECj3SEhS"
created_at : 2025-11-30T01:06:07.113+00:00
updated_at : 2025-11-30T01:06:07.116+00:00
__v : 0
```

Below the document preview, there are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The status bar at the bottom right shows the command: '- DELETE /api/v1/emp/employees/\_id=692b987f4f1831db7d5f6954 - Delete employee'.

## Create user

The screenshot shows the Postman application interface. The top navigation bar includes Home, Workspaces, API Network, and various status icons. A sidebar on the left contains sections for Collections, Environments, History, Flows, and Files (BETA). The main workspace is titled "HTTP COMP3123 / Create User". A POST request is being made to `http://localhost:5000/api/v1/user/signup`. The "Body" tab is selected, showing a raw JSON payload:

```
1 {  
2   "username": "johndoe",  
3   "email": "johndoe@example.com",  
4   "password": "password123"  
5 }
```

Below the request, the response details are shown: **201 Created**, 298 ms, 423 B. The response body is displayed as:

```
{ } JSON ▾
```

```
1 {  
2   "message": "User created successfully.",  
3   "user_id": "692bbb85f75fcedd57636c67"  
4 }
```

At the bottom, there are tabs for Cloud View, Console, Runner, Vault, and Help.

## Login user

The screenshot shows the Postman application interface. The top navigation bar includes Home, Workspaces, API Network, and various status indicators. The left sidebar has sections for Collections, Environments, History, Flows, and Files (BETA). The main workspace shows a collection named "COMP3123 / Create User". A POST request is being made to `http://localhost:5000/api/v1/user/login`. The request body is set to raw JSON:

```
1 {  
2   "email": "johndoe@example.com",  
3   "password": "password123"  
4 }
```

The response status is 200 OK, with a response time of 188 ms and a response size of 718 B. The response body is displayed as JSON:

```
{ } JSON ▾ ▶ Preview ▾ Visualize ▾  
1 {  
2   "message": "Login successful.",  
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJlc2Vx2lkIjoiNjkyYmJjODVmNzVmY2VkdDU3NjM2YzY3IiwidXNlcm5hbWUiOijqb2huZG9lIiwiZW1haWwiOijqb  
2huZG9lQGV4YW1wbGUuY29tIiwiWF0IjoxNzY0NDczNzcyLCJleHAiOjE3NjQ1NjAxNzJ9.  
Wiv18kRqeF4q-DHqNSCIEYZV9y95MwttIp05Rfx0fo",  
4   "user_id": "692bbb85f75fcedd57636c67",  
5   "username": "johndoe",  
6   "email": "johndoe@example.com"  
7 }
```

At the bottom, there are tabs for Cloud View, Console, Runner, Vault, and other tools.

## Create employee

The screenshot shows the Postman application interface for creating a new employee. The top navigation bar includes Home, Workspaces, API Network, and various status indicators. The left sidebar features sections for Collections, Environments, History, and Flows, with 'Environments' currently selected. The main workspace displays a POST request to 'http://localhost:5000/api/v1/emp/employees'. The request body contains the following JSON payload:

```
1 {  
2   "first_name": "Jane",  
3   "last_name": "Smith",  
4   "email": "jane.smith@company.com",  
5   "position": "Software Engineer",  
6   "salary": 85000,  
7   "date_of_joining": "2024-01-15",  
8   "department": "Engineering"  
9 }
```

The response section shows a successful '201 Created' status with a response time of 185 ms and a response size of 431 B. The response body is displayed as JSON:

```
1 {  
2   "message": "Employee created successfully.",  
3   "employee_id": "692bbc5f75fcedd57636c6b"  
4 }
```

At the bottom, there are tabs for Cloud View, Console, Runner, Vault, and other tools.

## Get all employees

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for Home, Workspaces, API Network, and several recent items. On the left sidebar, there are sections for Collections, Environments, History, and Flows. The main workspace shows a collection named "test" under the "HTTP" tab. A GET request is defined with the URL `http://localhost:5000/api/v1/emp/employees`. The request body is set to raw JSON:

```
1 {
2     "first_name": "Jane",
3     "last_name": "Smith",
4     "email": "jane.smith@company.com",
5     "position": "Software Engineer",
6     "salary": 85000,
7     "date_of_joining": "2024-01-15",
8     "department": "Engineering"
9 }
```

Below the request, the response is displayed in a JSON viewer. The status is 200 OK, with a response time of 95 ms and a size of 811 B. The response body contains two employee objects:1 [
2 {
3 "employee\_id": "692b989a4f1831db7d5f695a",
4 "first\_name": "a",
5 "last\_name": "a",
6 "email": "s@gmail.com",
7 "position": "s",
8 "salary": 21,
9 "date\_of\_joining": "2025-11-29T00:00:00.000Z",
10 "department": "s",
11 "profile\_picture": null
12 },
13 {
14 "employee\_id": "692bbbc5f75fcedd57636c6b",
15 "first\_name": "Jane",
16 "last\_name": "Smith",
17 "email": "jane.smith@company.com",
18 "position": "Software Engineer",
19 "salary": 85000,
20 "date\_of\_joining": "2024-01-15T00:00:00.000Z",
21 "department": "Engineering",
22 "profile\_picture": null
23 }
24 ]

At the bottom, there are tabs for Body, Cookies, Headers (10), Test Results, and a preview section showing the JSON structure.

## Get employee by id

The screenshot shows the Postman application interface. In the top navigation bar, 'Home' and 'Workspaces' are visible. The main workspace title is 'COMP3123 / test'. A search bar and various status indicators are present in the header.

The left sidebar includes sections for 'Collections', 'Environments', 'History', 'Flows', and 'Files (BETA)'. The 'Environments' section is currently selected.

The main content area shows a 'PUT' request being sent to the URL `http://localhost:5000/api/v1/emp/employees/692bbbc5f75fcedd57636c6b`. The 'Body' tab is active, displaying the following JSON payload:

```
1 {  
2   "position": "Senior Software Engineer",  
3   "salary": 95000  
4 }
```

Below the request, the response is shown under the 'Body' tab. The response code is 200 OK, with a timestamp of 105 ms and a size of 670 B. The response body is:

```
1 {  
2   "message": "Employee details updated successfully.",  
3   "employee": {  
4     "employee_id": "692bbbc5f75fcedd57636c6b",  
5     "first_name": "Jane",  
6     "last_name": "Smith",  
7     "email": "jane.smith@company.com",  
8     "position": "Senior Software Engineer",  
9     "salary": 95000,  
10    "date_of_joining": "2024-01-15T00:00:00.000Z",  
11    "department": "Engineering",  
12    "profile_picture": null  
13  }  
14 }
```

At the bottom, there are links for 'Cloud View', 'Console', 'Runner', 'Vault', and help icons.

## Update employee

The screenshot shows the Postman application interface for testing a REST API. The top navigation bar includes Home, Workspaces, API Network, and various status indicators. The left sidebar contains sections for Collections, Environments, History, Flows, and Files (BETA). The main workspace shows a collection named "test" under "HTTP". A PUT request is selected with the URL `http://localhost:5000/api/v1/emp/employees/692bbbc5f75fcedd57636c6b`. The "Body" tab is active, displaying the following JSON payload:

```
1 {  
2     "position": "Senior Software Engineer",  
3     "salary": 95000  
4 }
```

The response section shows a successful **200 OK** status with a response time of 105 ms and a response size of 670 B. The response body is displayed as JSON:

```
1 {  
2     "message": "Employee details updated successfully.",  
3     "employee": {  
4         "employee_id": "692bbbc5f75fcedd57636c6b",  
5         "first_name": "Jane",  
6         "last_name": "Smith",  
7         "email": "jane.smith@company.com",  
8         "position": "Senior Software Engineer",  
9         "salary": 95000,  
10        "date_of_joining": "2024-01-15T00:00:00.000Z",  
11        "department": "Engineering",  
12        "profile_picture": null  
13    }  
14 }
```

The bottom navigation bar includes Cloud View, Console, Runner, Vault, and other application icons.

## Delete employee

The screenshot shows the Postman application interface for testing an API endpoint. The main window displays a DELETE request to the URL `http://localhost:5000/api/v1/emp/employees?eid=692bbbc5f75fcedd57636c6b`. The request body contains the following JSON data:

```
1 {  
2   "position": "Senior Software Engineer",  
3   "salary": 95000  
4 }
```

The response section shows a successful `200 OK` status with a response time of 95 ms and a response size of 385 B. The response body is:

```
{ } JSON ▾ Preview Visualize ▾  
1 {  
2   "message": "Employee deleted successfully."  
3 }
```

The left sidebar of the Postman interface includes sections for Collections, Environments, History, Flows, and Files (BETA). The top navigation bar shows Home, Workspaces, API Network, and various search and settings icons.

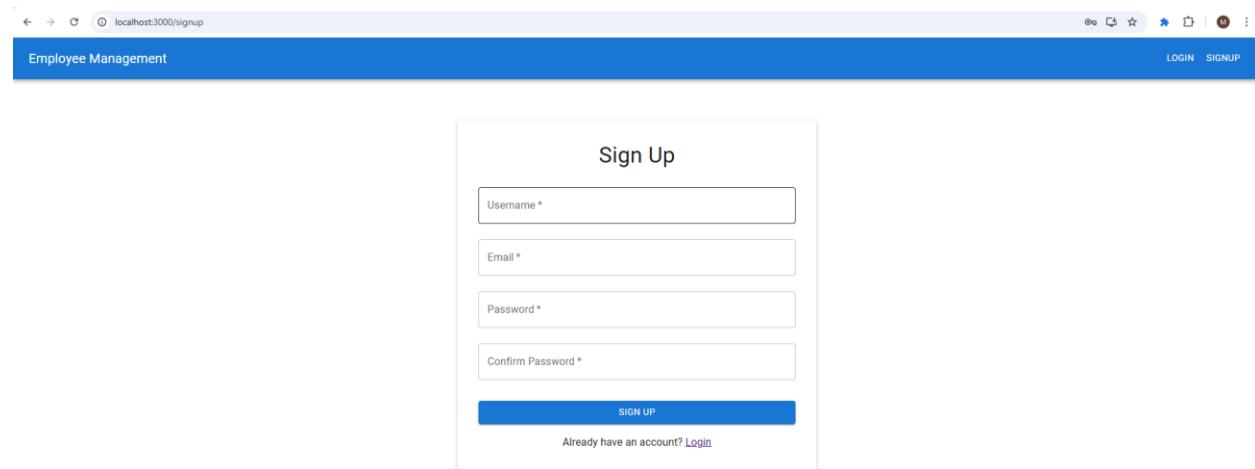
## Search employee by department and position

The screenshot shows the Postman application interface for testing an API endpoint. The left sidebar contains navigation links: Home, Workspaces, API Network, Collections, Environments, History, Flows, and Files (BETA). The main workspace displays a collection named "COMP3123 / test". A GET request is selected with the URL `http://localhost:5000/api/v1/emp/employees/search?department=Engineering&position=Engineer`. The "Body" tab is active, showing the response body as an empty JSON array: `[ ]`. The status bar at the bottom indicates a 200 OK response with 95 ms latency and 341 B size.

## Search by position

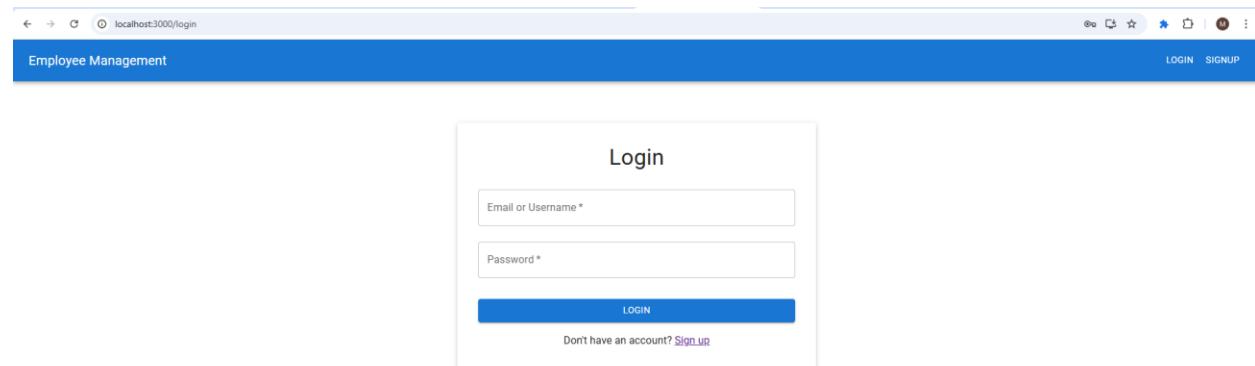
The screenshot shows the Postman application interface for testing an API endpoint. The left sidebar contains navigation links: Home, Workspaces, API Network, Collections, Environments, History, Flows, and Files (BETA). The main workspace displays a test environment named "COMP3123 / test". A GET request is being made to the URL `http://localhost:5000/api/v1/emp/employees/search?position=Software`. The "Body" tab is selected, showing the option "none" is chosen. The response section shows a 200 OK status with a response body of `[{"id": 1, "name": "Software"}]`. The bottom navigation bar includes Cloud View, Console, Runner, Vault, and other utility icons.

## Signup



A screenshot of a web browser showing the Signup page for "Employee Management". The URL is localhost:3000/signup. The page has a blue header bar with the title "Employee Management" and navigation links for "LOGIN" and "SIGNUP". The main content area is titled "Sign Up" and contains four input fields: "Username \*", "Email \*", "Password \*", and "Confirm Password \*". Below these fields is a blue "SIGN UP" button. At the bottom of the form, there is a link "Already have an account? [Login](#)".

## Login



A screenshot of a web browser showing the Login page for "Employee Management". The URL is localhost:3000/login. The page has a blue header bar with the title "Employee Management" and navigation links for "LOGIN" and "SIGNUP". The main content area is titled "Login" and contains two input fields: "Email or Username \*" and "Password \*". Below these fields is a blue "LOGIN" button. At the bottom of the form, there is a link "Don't have an account? [Sign up](#)".

## All employees

The screenshot shows a web browser window for 'Employee Management' at localhost:3000/employees. The page title is 'Employees'. At the top right, it says 'Welcome, arash' with links for 'EMPLOYEES' and 'LOGOUT'. A search bar at the top has fields for 'Department' and 'Position', with 'SEARCH' and 'CLEAR' buttons. Below the search bar is a table header row with columns: Photo, Name, Email, Position, Department, Salary, and Actions. A single employee record is listed: a placeholder photo, name 'a a', email 's@gmail.com', position 's', department 's', salary '\$21', and actions (edit, update, delete).

## Add employee

The screenshot shows a web browser window for adding a new employee at localhost:3000/employees/add. The title is 'Add Employee'. It features a placeholder profile picture with an 'UPLOAD PHOTO' button. Below are input fields for First Name\*, Last Name\*, Email\*, Position\*, Department\*, and Salary\*. There is also a date input field for Date of Joining\* with a calendar icon. At the bottom are 'CANCEL' and 'ADD EMPLOYEE' buttons.

## view employee

The screenshot shows a web browser window with the URL `localhost:3000/employees/view/692b989a4f1831db7d5f695a`. The page title is "Employee Management". The main content area displays an employee profile with a placeholder photo (a gray circle with a white letter 'a'). The employee's name is listed as "a a". Below the name, there is a small "s" character. A horizontal line separates this from the contact information: "Email s@gmail.com Department s Salary \$21". Underneath, it says "Date of Joining 11/28/2025". At the bottom are two buttons: a light blue "BACK" button and a dark blue "EDIT" button with a pencil icon.

## Edit employee

The screenshot shows a web browser window with the URL `localhost:3000/employees/edit/692b989a4f1831db7d5f695a`. The page title is "Edit Employee". The main content area has a placeholder photo (a gray circle with a white person icon). Below the photo is a blue "UPLOAD PHOTO" button with a cloud icon. The form fields are as follows: "First Name \*" with value "a", "Last Name \*" with value "a", "Email \*" with value "s@gmail.com", "Position \*" with value "s", "Department \*" with value "s", "Salary \*" with value "21", and "Date of Joining \*" with value "11/29/2025". At the bottom are two buttons: a light blue "CANCEL" button and a dark blue "UPDATE EMPLOYEE" button with a pencil icon.

## Delete employee

The screenshot shows a web browser window for 'Employee Management' at the URL 'localhost:3000/employees'. The page title is 'Employees'. At the top right, there are links for 'Welcome, arash', 'EMPLOYEES', and 'LOGOUT'. A search bar with fields for 'Department' and 'Position', and buttons for 'SEARCH' and 'CLEAR', is located at the top center. Below the search bar is a table header with columns: Photo, Name, Email, Position, Department, Salary, and Actions. A single row of data is visible: a placeholder photo, 'aa', 's@gmail.com', 'S', 'S', '\$21', and three icons for edit, delete, and view. A modal dialog box titled 'Confirm Delete' is centered over the table, asking 'Are you sure you want to delete this employee?'. It contains 'CANCEL' and 'DELETE' buttons.

Deleted

The screenshot shows the same 'Employee Management' application after the deletion. The table below the search bar now displays the message 'No employees found'.

## Search by position

The screenshot shows a web browser window with the URL 'localhost:3000/employees'. The page title is 'Employee Management'. On the right, there is a welcome message 'Welcome, arash' and links for 'EMPLOYEES' and 'LOGOUT'. The main content area is titled 'Employees' and features a search bar with fields for 'Department' (set to 'software') and 'Position' (set to 'software'), along with 'SEARCH' and 'CLEAR' buttons. Below the search bar is a table with columns: Photo, Name, Email, Position, Department, Salary, and Actions. A single row is visible, representing an employee named 'a a' with email 'a@gmail.com', position 'software', department 'tech', salary '\$200', and actions represented by three icons: a magnifying glass, a pencil, and a trash can.

## Employees

+ ADD EMPLOYEE

Photo	Name	Email	Position	Department	Salary	Actions
	a a	a@gmail.com	software	tech	\$200	

## Search by department

The screenshot shows a web browser window with the URL 'localhost:3000/employees'. The page title is 'Employee Management'. On the right, there is a welcome message 'Welcome, arash' and links for 'EMPLOYEES' and 'LOGOUT'. The main content area is titled 'Employees' and features a search bar with fields for 'Department' (set to 'tech') and 'Position' (empty), along with 'SEARCH' and 'CLEAR' buttons. Below the search bar is a table with columns: Photo, Name, Email, Position, Department, Salary, and Actions. A single row is visible, representing an employee named 'a a' with email 'a@gmail.com', position 'software', department 'tech', salary '\$200', and actions represented by three icons: a magnifying glass, a pencil, and a trash can.

## Employees

+ ADD EMPLOYEE

Photo	Name	Email	Position	Department	Salary	Actions
	a a	a@gmail.com	software	tech	\$200	