

COMP3123 – Assignment 02

Olympia Santaguida-Antunes

Student# 101469745

Containers Running

Ask Gordon BETA

Containers

Images

Volumes

Kubernetes

Builds

Models

MCP Toolkit BETA

Docker Hub

Docker Scout

Extensions

Containers [Give feedback](#)

Container CPU usage ⓘ
3.16% / 2000% (20 CPUs available)

Container memory usage ⓘ
467.92MB / 15.07GB

Show charts

Q Search

Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<input type="radio"/> minikube	b1351ac7afe6	k8s-minikube/kicbas		0%	12 days ago	<div><div></div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input checked="" type="radio"/> 101469745_assignment02	-	-	-	9.14%	1 minute ago	<div><div></div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input checked="" type="radio"/> employee_mongodb	f5aa81578943	mongo:latest	27017:27017	1.42%	1 minute ago	<div><div></div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input checked="" type="radio"/> employee_backend	d000915e49e7	101469745_assignm	3000:3000	0.22%	1 minute ago	<div><div></div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input checked="" type="radio"/> employee_frontend	4fad02d0c91c	101469745_assignm	3001:3000	7.5%	1 minute ago	<div><div></div><div></div><div></div><div></div></div>

Showing 5 items

Engine running

RAM 3.44 GB CPU 0.05% Disk: 21.22 GB used (limit 1006.85 GB)

Terminal [Update available](#)

Frontend: SignUp

A screenshot of a web browser displaying the 'Employee Management App' Sign Up form. The browser's address bar shows 'http://localhost:3001/signup'. The form is centered on a purple gradient background. It contains the following fields: Username (olympia15), Email (olympia15@google.ca), Password (masked with dots), and Confirm Password (masked with dots). A blue 'Sign Up' button is at the bottom of the form. Below the button is a link: 'Already have an account? Login here'.

Employee Management App
Sign Up

Username
olympia15

Email
olympia15@google.ca

Password
.....

Confirm Password
.....

Sign Up

Already have an account? [Login here](#)

Frontend: SignUp Success

A screenshot of the same web browser showing the 'Employee Management App' Sign Up form after a successful registration. A dark grey toast notification is displayed over the form, containing the text 'localhost:3001 Signup successfull Please login.' and an 'OK' button. The 'Sign Up' button has been replaced by a button labeled 'Creating Account...'. The rest of the form fields and the 'Login here' link remain visible.

Employee Management App
Sign Up

Username
olympia15

Email

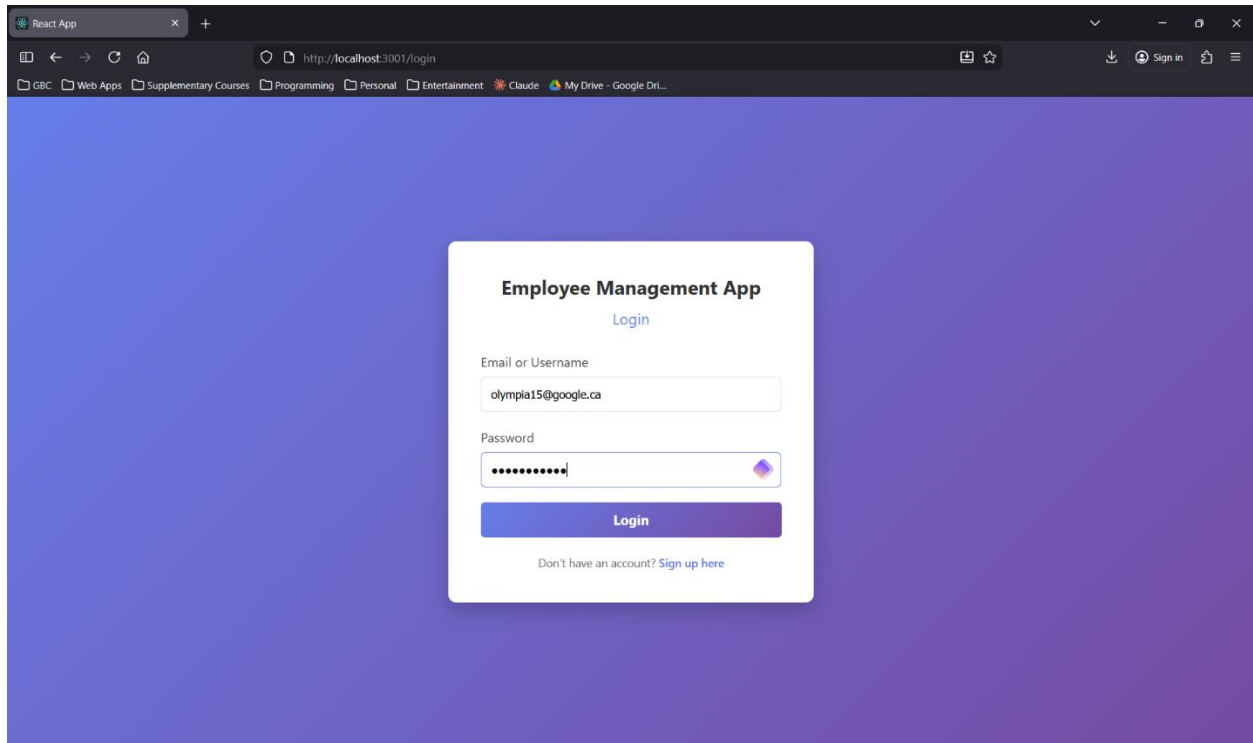
localhost:3001
Signup successfull Please login.
OK

Confirm Password
.....

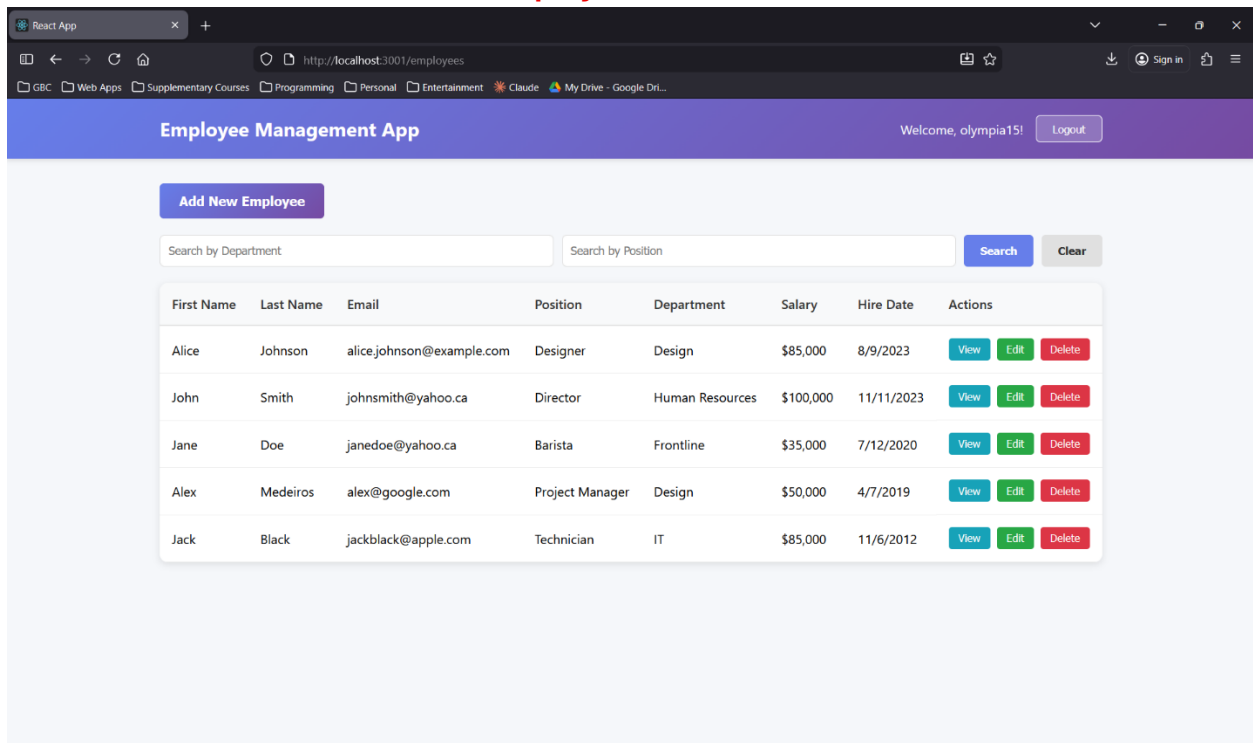
Creating Account...

Already have an account? [Login here](#)

Frontend: Login



Frontend: Search Screen with All Employees



Frontend: Add Employee

A screenshot of a web browser displaying a form titled "Add New Employee". The form is centered on a light blue background. It contains several input fields: "First Name" (Sarah), "Last Name" (Thornton), "Email" (saraht@rogers.com), "Position" (Intern), "Department" (Design), "Salary" (30000), and "Hire Date" (06 / 01 / 2024). At the bottom, there are two buttons: "Add Employee" (purple) and "Cancel" (grey). The browser's address bar shows "http://localhost:3001/employees/add".

Add New Employee

First Name: Sarah

Last Name: Thornton

Email: saraht@rogers.com

Position: Intern

Department: Design

Salary: 30000

Hire Date: 06 / 01 / 2024

Add Employee **Cancel**

Frontend: Add Success

A screenshot of the same "Add New Employee" form, but with a success message overlay. The overlay is a dark grey box with the text "localhost:3001" and "Employee added successfully!". Below the text is a red "OK" button. The form's "Add Employee" button is now disabled and labeled "Adding...". The "Cancel" button remains visible. The browser's address bar shows "http://localhost:3001/employees/add".

Add New Employee

First Name: Sarah

Last Name: Thornton

Email: saraht@rogers.com

Position: Intern

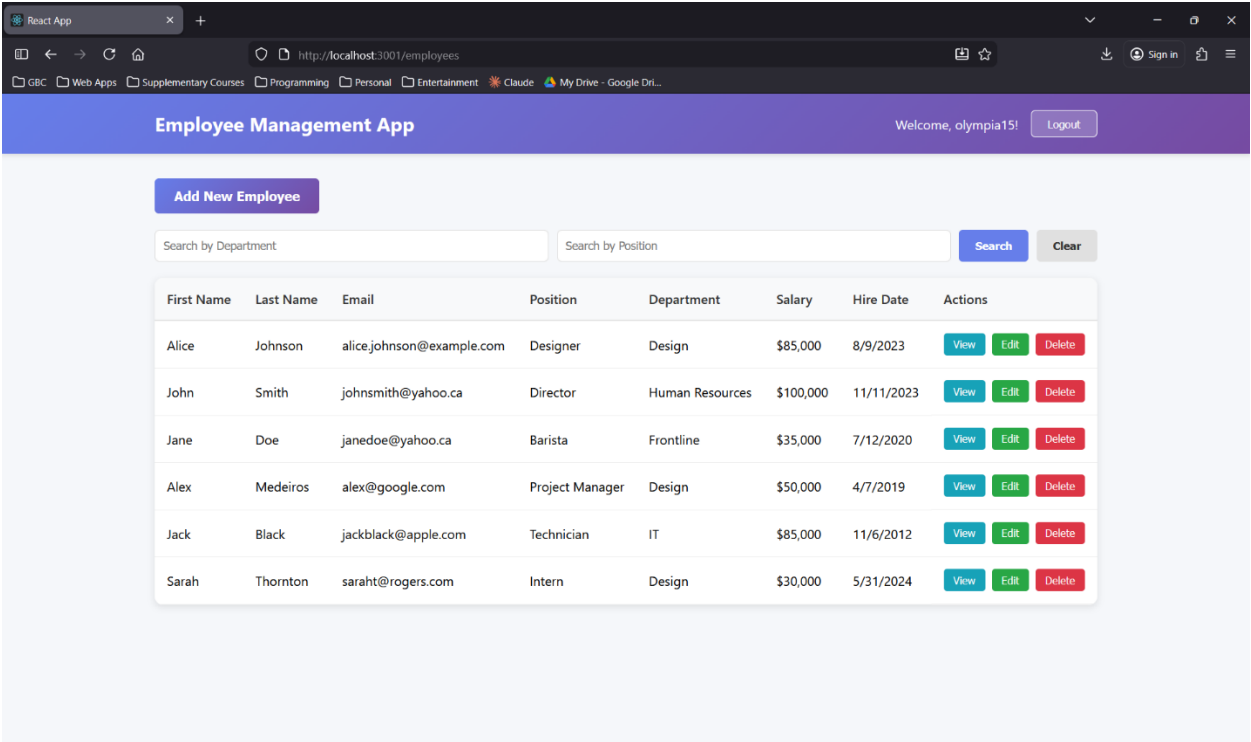
Salary: 30000

Hire Date: 06 / 01 / 2024

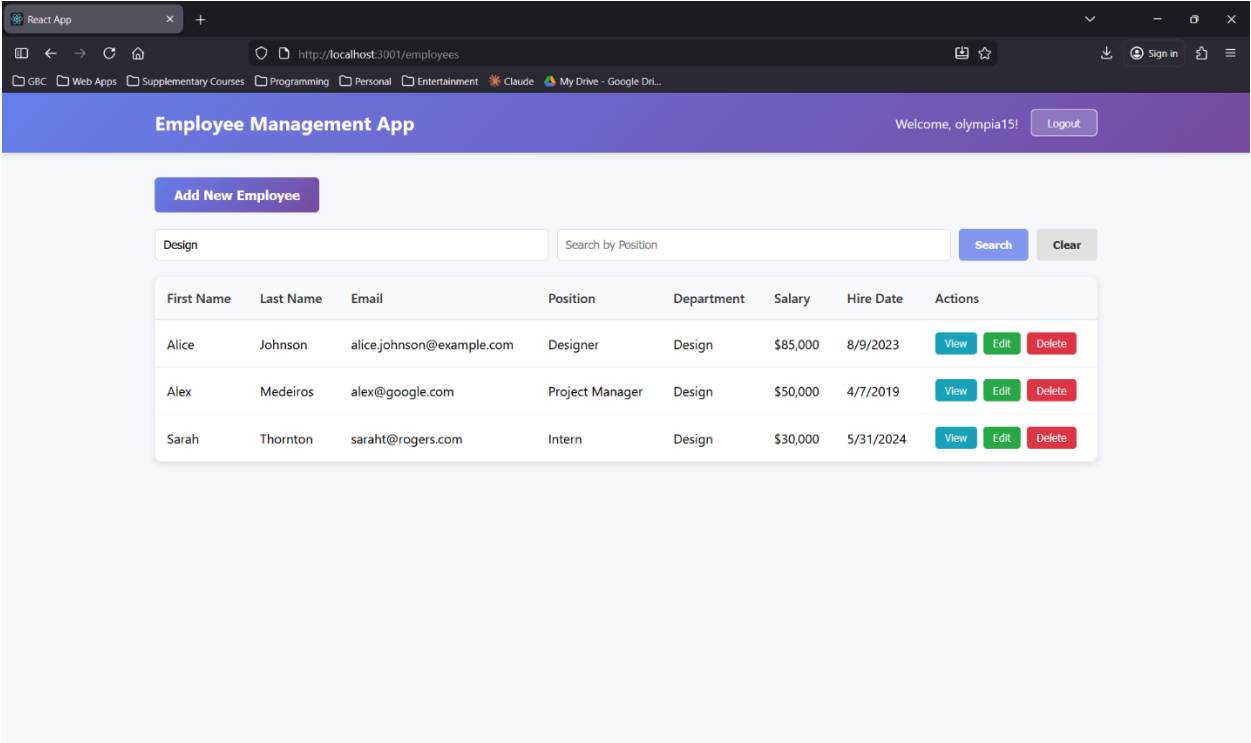
Adding... **Cancel**

localhost:3001
Employee added successfully!
OK

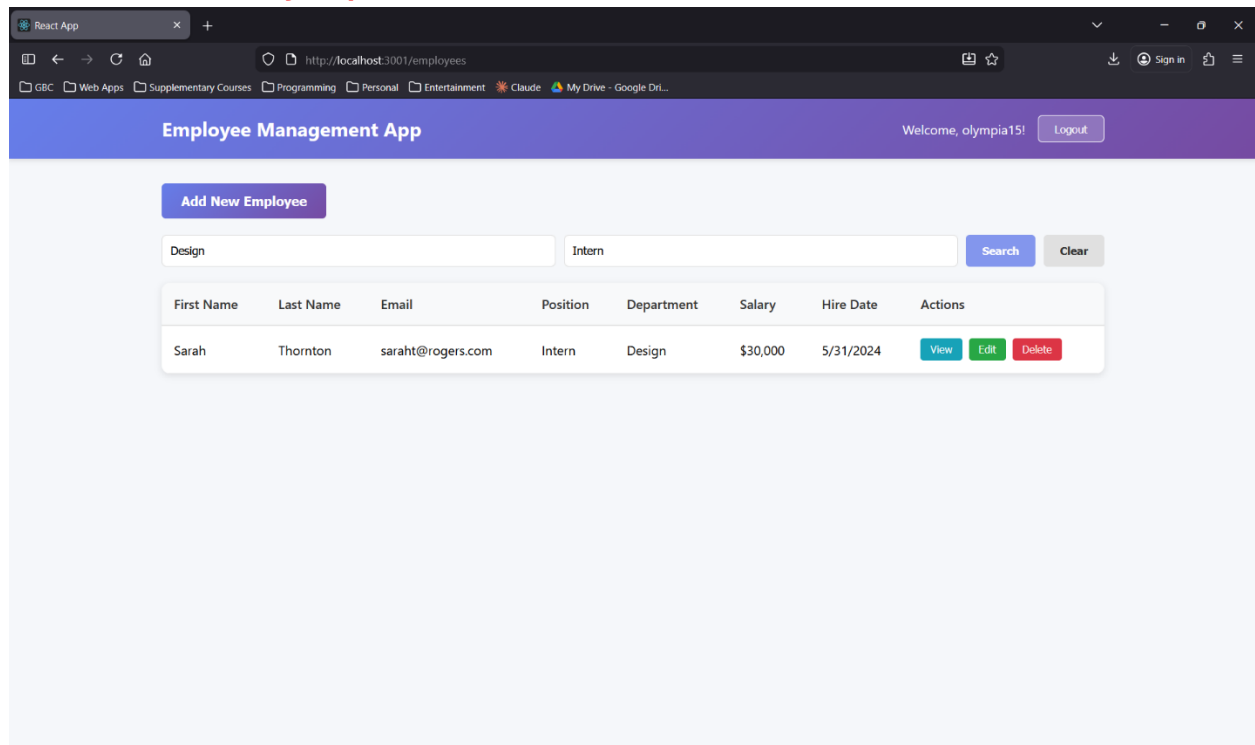
Frontend: Search Screen with Newly Added Employee



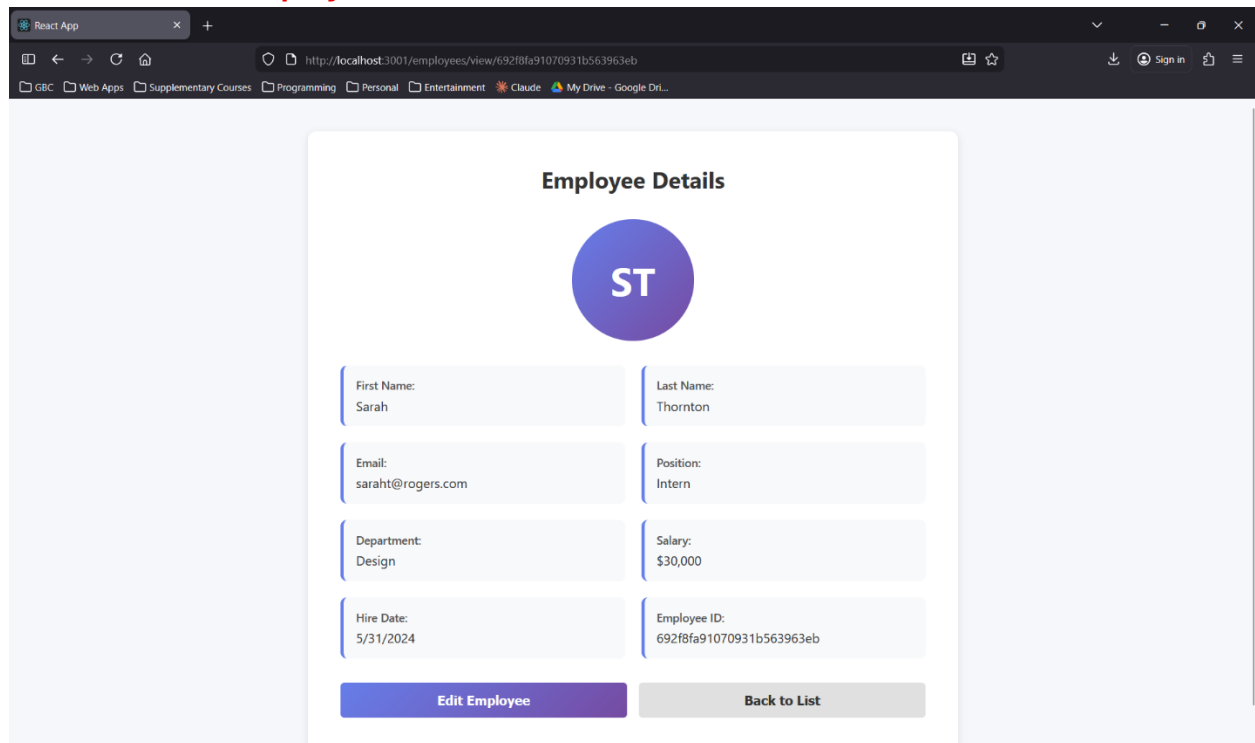
Frontend: Search by Department



Frontend: Search by Department and Position



Frontend: View Employee Details



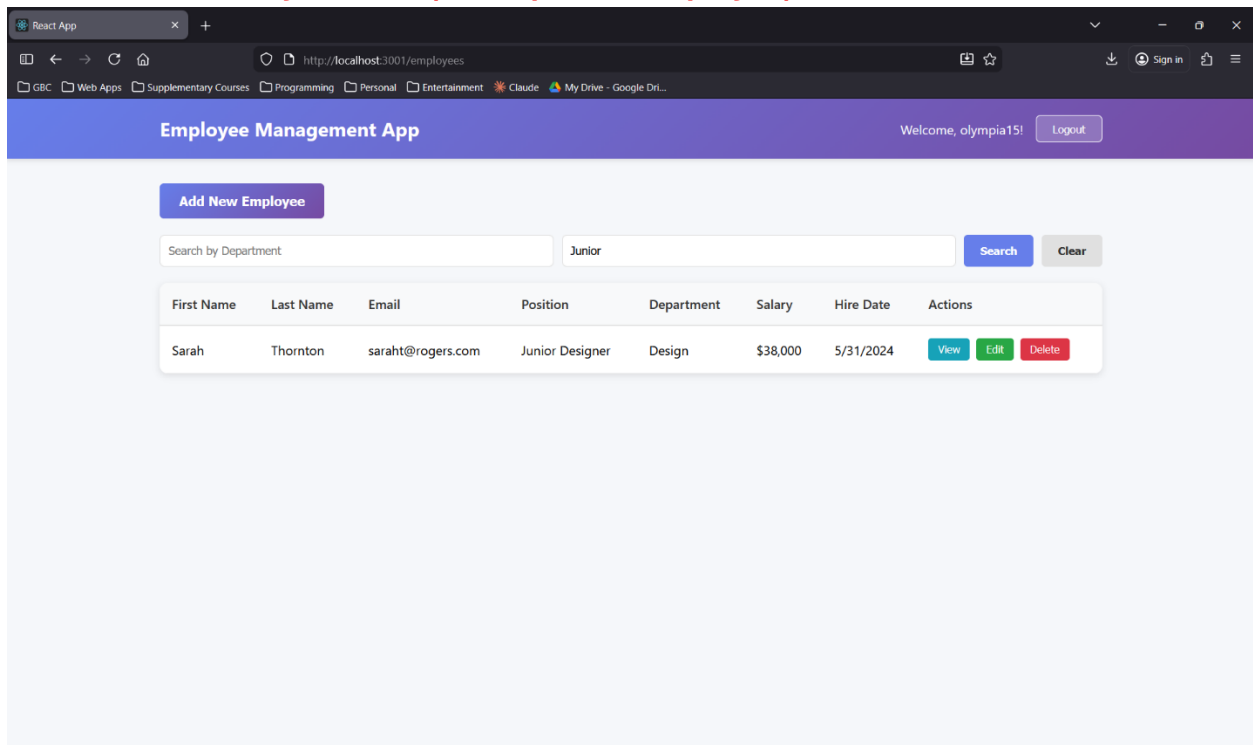
Frontend: Edit Employee

A screenshot of a web browser displaying a React App. The address bar shows the URL `http://localhost:3001/employees/edit/692f8fa91070931b563963eb`. The browser's tab is labeled 'React App'. The page content is a white modal form titled 'Edit Employee'. The form contains several input fields: 'First Name' with the value 'Sarah', 'Last Name' with 'Thornton', 'Email' with 'saraht@rogers.com', 'Position' with 'Junior Designer', 'Department' with 'Design', 'Salary' with '38000', and 'Hire Date' with '06 / 01 / 2024'. At the bottom of the form are two buttons: a purple 'Update Employee' button and a grey 'Cancel' button.

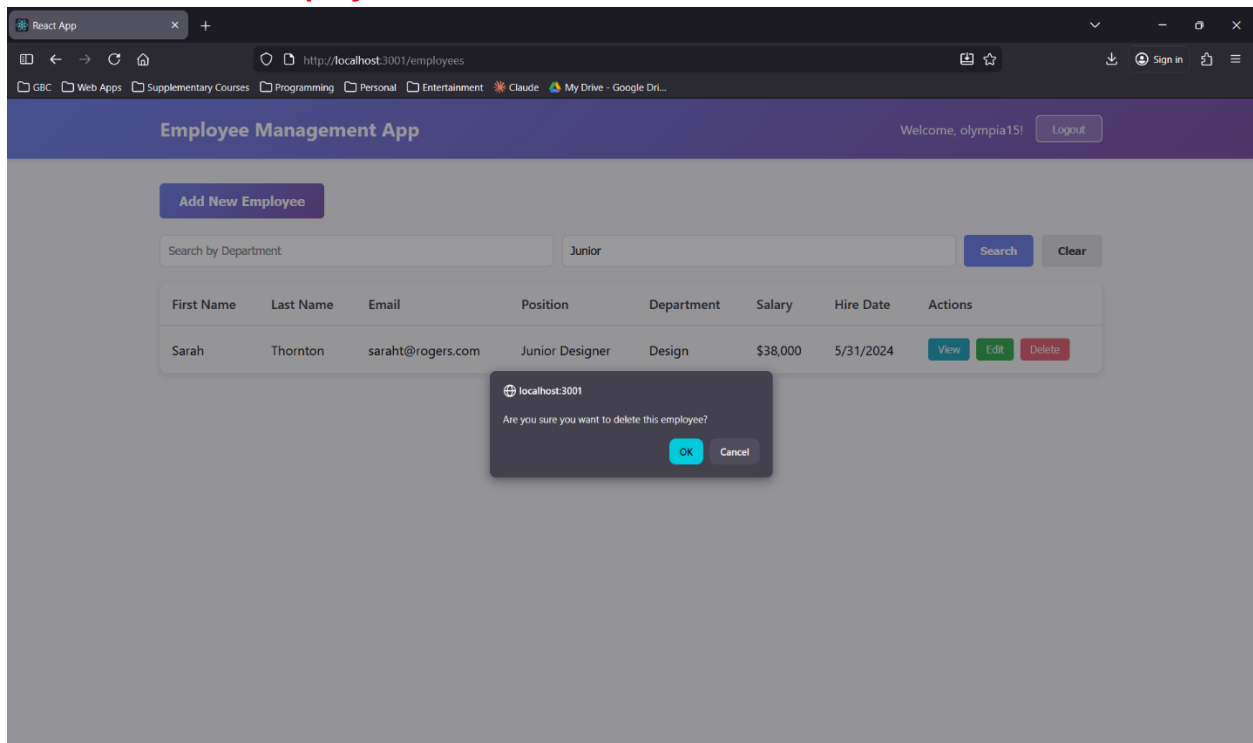
Frontend: Edit Success

A screenshot of the same web browser and 'Edit Employee' form, but in a state of success. A dark grey toast notification is displayed in the center of the form, containing the text 'localhost:3001' and 'Employee updated successfully!' with an 'OK' button. The 'Update Employee' button at the bottom of the form is now disabled and labeled 'Updating...'. The 'Cancel' button remains visible. The background of the page is dimmed.

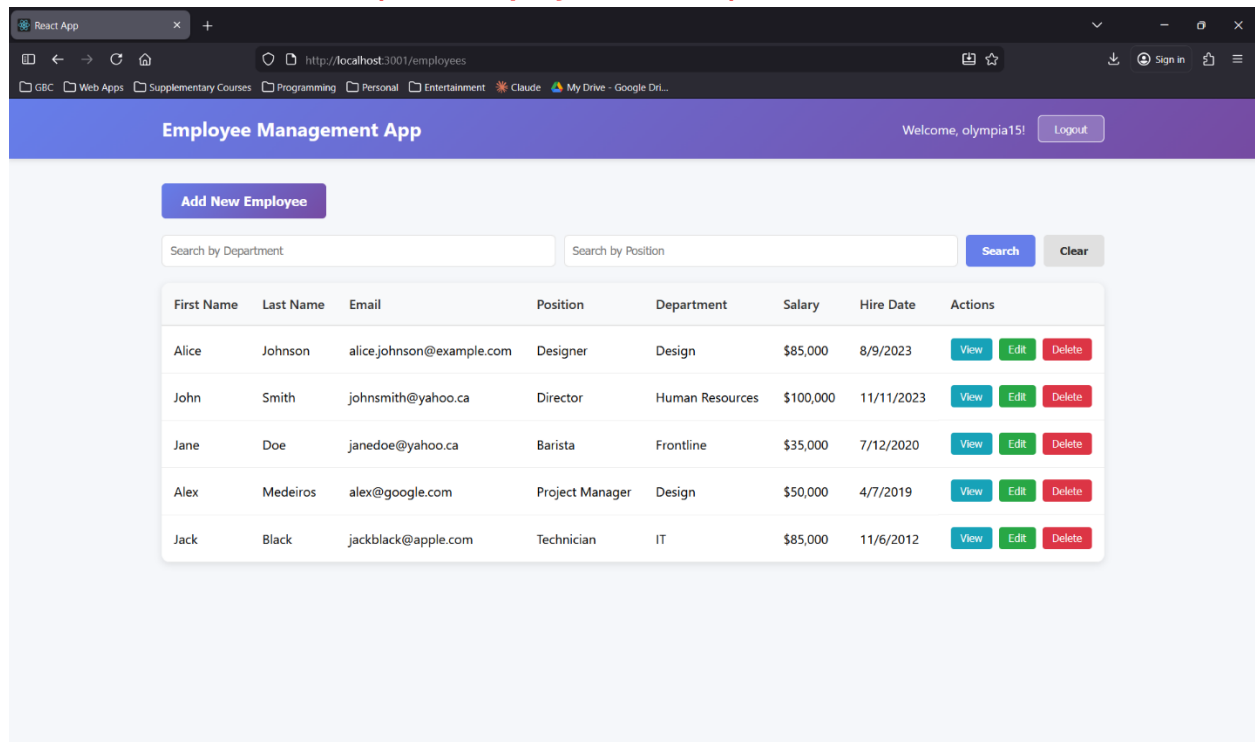
Frontend: Search by Position (with Updated Employee)



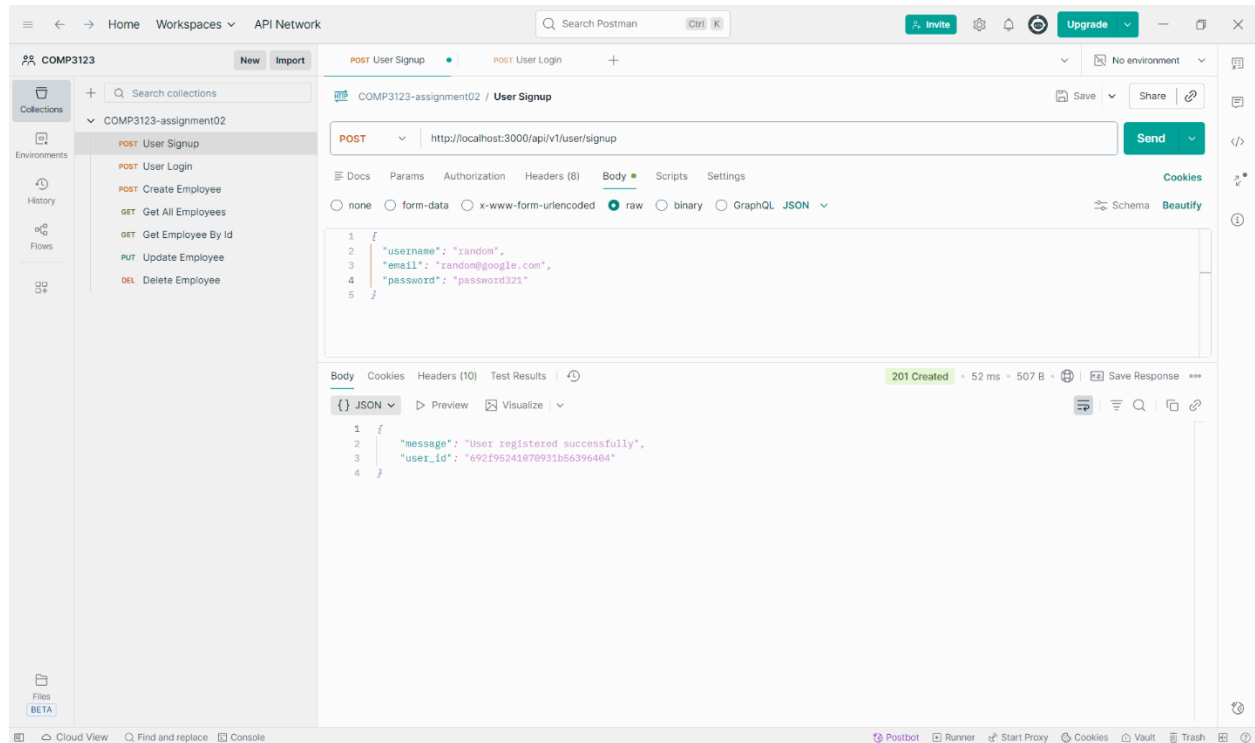
Frontend: Delete Employee Confirm



Frontend: Search Screen (After Employee Deleted)



Backend: SignUp



Backend: Login

The screenshot shows the Postman interface with a collection named 'COMP3123' and a sub-collection 'COMP3123-assignment02'. The 'User Login' endpoint is selected, showing a POST request to 'http://localhost:3000/api/v1/user/login'. The request body is a JSON object:

```
{  "email": "random@google.com",  "password": "password321"}
```

. The response is a 200 OK status with a JSON body:

```
{  "message": "Login successful",  "user": {    "id": "692f96241679931b56396484",    "username": "random",    "email": "random@google.com"  }}
```

Backend: Login With Wrong Password

The screenshot shows the Postman interface with the same collection and endpoint as the previous screenshot. The request body is a JSON object:

```
{  "email": "random@google.com",  "password": "wrongpassword"}
```

. The response is a 400 Bad Request status with a JSON body:

```
{  "status": "false",  "message": "Invalid email/username or password"}
```

Backend: Create Employee

The screenshot shows the Postman interface with a collection named 'COMP3123' and a sub-collection 'COMP3123-assignment02'. The 'Create Employee' endpoint is selected, showing a POST request to 'http://localhost:3000/api/v1/emp/employees'. The request body is a JSON object representing an employee. The response is a 201 status code with a success message and the employee ID.

```
POST http://localhost:3000/api/v1/emp/employees
```

```
{
  "firstName": "Olympia",
  "lastName": "Santaguida",
  "email": "olympia@outlook.com",
  "position": "Junior DevOps Engineer",
  "salary": 65000,
  "hireDate": "2025-06-18",
  "department": "IT"
}
```

```
{
  "message": "Employee created successfully.",
  "employee_id": "692f95eb1070931b5639640a"
}
```

Backend: Get All Employees

The screenshot shows the Postman interface with the same collection and sub-collection. The 'Get All Employees' endpoint is selected, showing a GET request to 'http://localhost:3000/api/v1/emp/employees'. The response is a 200 status code with a list of three employees.

```
GET http://localhost:3000/api/v1/emp/employees
```

```
[
  {
    "employee_id": "692f83d8e9aa5e35ce2904e",
    "firstName": "Alex",
    "lastName": "Medeiros",
    "email": "alex@google.com",
    "position": "Project Manager",
    "salary": 50000,
    "hireDate": "2019-04-08T00:00:00.000Z",
    "department": "Design"
  },
  {
    "employee_id": "692f84110e9aa5e35ce29053",
    "firstName": "Jack",
    "lastName": "Black",
    "email": "jack@blackapple.com",
    "position": "Technician",
    "salary": 85000,
    "hireDate": "2012-11-07T00:00:00.000Z",
    "department": "IT"
  },
  {
    "employee_id": "692f95eb1070931b5639640a",
    "firstName": "Olympia",
    "lastName": "Santaguida",
    "email": "olympia@outlook.com",
    "position": "Junior DevOps Engineer",
    "salary": 65000,
    "hireDate": "2025-06-18T00:00:00.000Z",
    "department": "IT"
  }
]
```

Backend: Update Employee

COMP3123 assignment02 / Update Employee

PUT http://localhost:3000/api/v1/emp/employees/692f95eb1070931b5639640a

Body

```
1 {
2   "position": "Senior DevOps Engineer",
3   "salary": 75000
4 }
```

200 OK • 8 ms • 508 B

Body

```
1 {
2   "message": "Employee updated successfully.",
3   "employee_id": "692f95eb1070931b5639640a"
4 }
```

Backend: Get Employee By ID (with Updated Employee)

COMP3123 assignment02 / Get Employee By Id

GET http://localhost:3000/api/v1/emp/employees/692f95eb1070931b5639640a

Query Params

Key	Value	Description
Key	Value	Description

200 OK • 9 ms • 649 B

Body

```
1 {
2   "employee_id": "692f95eb1070931b5639640a",
3   "firstName": "Olympia",
4   "lastName": "Santaguida",
5   "email": "olympia@outlook.com",
6   "position": "Senior DevOps Engineer",
7   "salary": 75000,
8   "hireDate": "2025-06-10T00:00:00.000Z",
9   "department": "IT"
10 }
```

Backend: Delete Employee

The screenshot shows the Postman interface with a workspace named 'COMP3123'. The left sidebar lists collections, including 'COMP3123-assignment02' which contains several API endpoints. The main panel displays a 'DELETE' request to 'http://localhost:3000/api/v1/emp/employees?id=692f95eb1070931b5639640a'. The 'Params' tab is active, showing a query parameter 'id' with the value '692f95eb1070931b5639640a'. The 'Body' tab is selected, showing a '204 No Content' response status, indicating a successful deletion.

COMP3123-assignment02 / Delete Employee

DELETE http://localhost:3000/api/v1/emp/employees?id=692f95eb1070931b5639640a

Query Params

Key	Value	Description
id	692f95eb1070931b5639640a	

Body: 204 No Content • 8 ms • 322 B

Backend: Get Employee By ID (with Deleted Employee)

The screenshot shows the Postman interface with a workspace named 'COMP3123'. The left sidebar lists collections, including 'COMP3123-assignment02' which contains several API endpoints. The main panel displays a 'GET' request to 'http://localhost:3000/api/v1/emp/employees/692f95eb1070931b5639640a'. The 'Params' tab is active, showing a query parameter 'id' with the value '692f95eb1070931b5639640a'. The 'Body' tab is selected, showing a '404 Not Found' response status, indicating that the employee has been deleted and is no longer found.

COMP3123-assignment02 / Get Employee By Id

GET http://localhost:3000/api/v1/emp/employees/692f95eb1070931b5639640a

Query Params

Key	Value	Description
id	692f95eb1070931b5639640a	

Body: 404 Not Found • 7 ms • 480 B

```
{
  "status": "false",
  "message": "Employee not found."
}
```

Backend: Get Employee by ID (with Active Employee)

The screenshot displays the Postman interface for a REST client. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. A search bar for 'Search Postman' is present, along with buttons for 'Invite', 'Upgrade', and a 'No environment' dropdown. The left sidebar shows a collection named 'COMP3123' with a sub-collection 'COMP3123-assignment02'. The main panel is titled 'COMP3123-assignment02 / Get Employee By Id'. The request method is 'GET' and the URL is 'http://localhost:3000/api/v1/emp/employees/692f84118e9aa5e35ce29053'. The 'Params' tab is active, showing a table with columns 'Key', 'Value', and 'Description'. The 'Body' tab is also active, showing a JSON response. The response status is '200 OK' with a response time of '9 ms' and a size of '629 B'. The JSON body contains employee details.

GET http://localhost:3000/api/v1/emp/employees/692f84118e9aa5e35ce29053

Query Params

Key	Value	Description
Key	Value	Description

Body

```
1 {
2   "employee_id": "692f84118e9aa5e35ce29053",
3   "firstName": "Jack",
4   "lastName": "Black",
5   "email": "jackblack@apple.com",
6   "position": "Technician",
7   "salary": 85000,
8   "hireDate": "2012-11-07T00:00:00.000Z",
9   "department": "IT"
10 }
```

200 OK • 9 ms • 629 B • Save Response