

George Brown College

Student: Breno Lopes Mafra

Student ID: 101485572

Assignment 2 – Screenshots

1 - Docker Containers up and running:

The screenshot shows the Docker Desktop application interface. At the top, there's a blue header bar with various icons: a 'PERSONAL' button, a search bar, a 'K' icon, a question mark, a bell with a red notification badge (2), a graduation cap, a gear, a three-dot menu, and a 'B' button. Below the header, the main interface is titled 'Containers' with a 'Give feedback' link. It displays two resource usage metrics: 'Container CPU usage' (2.15% / 800%) and 'Container memory usage' (270.61MB / 7.57GB). A 'Show charts' button is also present. The main area is a table listing five containers:

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	Actions
<input type="checkbox"/>	101485572_comp3123_assignment2	-	-	-	
<input type="checkbox"/>	comp3123-mongodb	29b875a59821	mongo:7.0	27017:27017	
<input type="checkbox"/>	comp3123-mongo-express	0342e4f8ee87	mongo-exp	8081:8081	
<input type="checkbox"/>	comp3123-frontend	8eff5fb05a54	101485572	3000:5173	
<input type="checkbox"/>	comp3123-backend	b5f8e2688066	101485572	5001:5000	

2 – Postman Screenshots:

2.1 Sign Up (POST <http://localhost:5001/api/auth/signup>)

The screenshot shows the Postman interface for a collection named "comp3123_assignment2". The left sidebar lists various environments, flows, and history items. The main workspace displays a POST request for "Sign Up" to the URL <http://localhost:5001/api/auth/signup>. The "Body" tab is selected, showing a JSON payload:

```
1 {  
2   "name": "Breno Lopes Mafra",  
3   "email": "brenolopesmafра@gmail.com",  
4   "password": "123456"  
5 }
```

Below the request, the response is shown as a 201 Created status with a response time of 188 ms and a response size of 379 B. The response body is a JSON object with an id field:

```
1 {  
2   "id": "69136e48ac6ff0e791d25411"  
3 }
```

2.2 Log In (POST <http://localhost:5001/api/auth/login>)

The screenshot shows the Postman interface for the same collection "comp3123_assignment2". The left sidebar lists various environments, flows, and history items. The main workspace displays a POST request for "Log In" to the URL <http://localhost:5001/api/auth/login>. The "Body" tab is selected, showing a JSON payload:

```
1 {  
2   "email": "brenolopesmafра@gmail.com",  
3   "password": "123456"  
4 }
```

Below the request, the response is shown as a 200 OK status with a response time of 142 ms and a response size of 525 B. The response body is a JSON object with a token field:

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJpZCI6IjY5MTM2ZTQ4YWm2ZmYwZTc5MWQyNTQxMSIisImIhdCI6MTc2Mjg4MTEExMSwi  
ZXhwIjoxNzYzNDg10TExfQ.B00Fz43FvgZolkv0VYFhZIaL4D99-J0jZ7BtSNY0QHc"  
3 }
```

2.3 Create Employee (POST <http://localhost:5001/api/employees>)

The screenshot shows the Postman interface with the following details:

- Collection:** comp3123_assignment2
- Request Type:** POST
- URL:** http://localhost:5001/api/employees
- Headers:** (9 items)
- Body:** form-data (4 fields)
 - firstName: Nazyrha
 - lastName: Nassar Salles
 - email: nahhsalles@gmail.com
 - address: Toronto
- Response:** 201 Created (77 ms, 663 B)
 - JSON response:

```
1 {  
2   "firstName": "Nazyrha",  
3   "lastName": "Nassar Salles",  
4   "email": "nahhsalles@gmail.com",  
5   "address": "Toronto",  
6   "salary": 5000,  
7   "department": "IT",  
8   "position": "Developer",  
9   "picture": "1762881552300-809468459.jpg",  
10  "_id": "69137010ac6ff0e791d25414",  
11  "createdAt": "2025-11-11T17:19:12.318Z",  
12  "updatedAt": "2025-11-11T17:19:12.318Z",  
13  "__v": 0  
14 }
```

2.4 Get All Employees (GET <http://localhost:5001/api/employees>)

The screenshot shows the Postman interface with the following details:

- Collection:** comp3123_assignment2
- Request Type:** GET
- URL:** http://localhost:5001/api/employees
- Headers:** (7 items)
 - Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI...
- Response:** 200 OK (65 ms, 660 B)
 - JSON response:

```
1 [  
2   {  
3     "_id": "69137010ac6ff0e791d25414",  
4     "firstName": "Nazyrha",  
5     "lastName": "Nassar Salles",  
6     "email": "nahhsalles@gmail.com",  
7     "address": "Toronto",  
8     "salary": 5000,  
9     "department": "IT",  
10    "position": "Developer",  
11    "picture": "1762881552300-809468459.jpg",  
12    "createdAt": "2025-11-11T17:19:12.318Z",  
13    "updatedAt": "2025-11-11T17:19:12.318Z",  
14    "__v": 0  
15   }  
16 ]
```

2.5 Get Employee By ID (GET <http://localhost:5001/api/employees/{id}>)

The screenshot shows the Postman interface with the collection 'comp3123_assignment2' selected. The left sidebar lists various API endpoints. The 'GET Get Employee By ID' endpoint is currently selected and highlighted in grey. The main panel displays the request details for this endpoint.

Request URL: `http://localhost:5001/api/employees/69137010ac6ff0e791d25414`

Headers:

Key	Value
Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI...
Key	Value

Response: 200 OK

Body:

```
1 {  
2   "_id": "69137010ac6ff0e791d25414",  
3   "firstName": "Nazyra",  
4   "lastName": "Nassar Salles",  
5   "email": "nahhsalles@gmail.com",  
6   "address": "Toronto",  
7   "salary": 5000,  
8   "department": "IT",  
9   "position": "Developer",  
10  "picture": "1762881552300-809468459.jpg",  
11  "createdAt": "2025-11-11T17:19:12.318Z",  
12  "updatedAt": "2025-11-11T17:19:12.318Z",  
13  "__v": 0  
14 }
```

2.6 Get Employee By Name (GET <http://localhost:5001/api/employees/search?name={name}>)

The screenshot shows the Postman interface with the collection 'comp3123_assignment2' selected. The left sidebar lists various API endpoints. The 'GET Search Employee By Name' endpoint is currently selected and highlighted in grey. The main panel displays the request details for this endpoint.

Request URL: `http://localhost:5001/api/employees/search?name=Nazyra`

Headers:

Key	Value
Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI...
Key	Value

Response: 200 OK

Body:

```
1 [  
2   {  
3     "_id": "69137010ac6ff0e791d25414",  
4     "firstName": "Nazyra",  
5     "lastName": "Nassar Salles",  
6     "email": "nahhsalles@gmail.com",  
7     "address": "Toronto",  
8     "salary": 10000,  
9     "department": "IT",  
10    "position": "Developer",  
11    "picture": "1762881552300-809468459.jpg",  
12    "createdAt": "2025-11-11T17:19:12.318Z",  
13    "updatedAt": "2025-11-11T17:24:39.084Z",  
14    "__v": 0  
15  }  
16 ]
```

2.7 Get Employee By Department (GET <http://localhost:5001/api/employees/search?department={department}>)

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows collections like 'comp3123_assignment2' with various endpoints: POST Sign Up, POST Log In, POST Create Employee, GET Get Employees, GET Get Employee By ID, GET Search Employee By Name, GET Search Employee By Department (which is selected), GET Search Employee By Position, PUT Update Employee, and DEL Delete Employee.
- Top Bar:** Shows the collection 'comp3123_assignment2' and various operations: New, Import, PUT Update, DEL Delete, GET Search (x2), and the currently selected GET Search.
- Header Bar:** Shows the URL 'http://localhost:5001/api/employees/search?department=IT' and a 'Send' button.
- Headers Tab:** Shows the 'Authorization' header with the value 'eyJhbGciOiJIUzI1NiIsInR5cCI...'. Other headers like 'Content-Type' and 'Accept' are also present.
- Body Tab:** Shows a JSON response with a single employee object:

```

1  [
2    {
3      "_id": "69137010ac6ff0e791d25414",
4      "firstName": "Nazryra",
5      "lastName": "Nassar Salles",
6      "email": "nahhsalles@gmail.com",
7      "address": "Toronto",
8      "salary": 10000,
9      "department": "IT",
10     "position": "Developer",
11     "picture": "1762881552300-809468459.jpg",
12     "createdAt": "2025-11-11T17:19:12.318Z",
13     "updatedAt": "2025-11-11T17:24:39.084Z",
14     "__v": 0
15   }
16 ]

```

2.8 Get Employee By Position (GET <http://localhost:5001/api/employees/search?position={position}>)

The screenshot shows the MongoDB Compass interface with the following details:

- Left Sidebar:** Shows collections like 'comp3123_assignment2' with various endpoints: POST Sign Up, POST Log In, POST Create Employee, GET Get Employees, GET Get Employee By ID, GET Search Employee By Name, GET Search Employee By Department, GET Search Employee By Position (which is selected), PUT Update Employee, and DEL Delete Employee.
- Top Bar:** Shows the collection 'comp3123_assignment2' and various operations: New, Import, DEL Delete, GET Search (x2), and the currently selected GET Search.
- Header Bar:** Shows the URL 'http://localhost:5001/api/employees/search?position=Developer' and a 'Send' button.
- Headers Tab:** Shows the 'Authorization' header with the value 'eyJhbGciOiJIUzI1NiIsInR5cCI...'. Other headers like 'Content-Type' and 'Accept' are also present.
- Body Tab:** Shows a JSON response with a single employee object:

```

1  [
2    {
3      "_id": "69137010ac6ff0e791d25414",
4      "firstName": "Nazryra",
5      "lastName": "Nassar Salles",
6      "email": "nahhsalles@gmail.com",
7      "address": "Toronto",
8      "salary": 10000,
9      "department": "IT",
10     "position": "Developer",
11     "picture": "1762881552300-809468459.jpg",
12     "createdAt": "2025-11-11T17:19:12.318Z",
13     "updatedAt": "2025-11-11T17:24:39.084Z",
14     "__v": 0
15   }
16 ]

```

2.9 Update Employee (PUT <http://localhost:5001/api/employees/{id}>)

The screenshot shows the MongoDB Compass interface. On the left sidebar, under the 'comp3123_assignment2' collection, the 'PUT Update Employee' option is selected. In the main panel, a PUT request is being made to <http://localhost:5001/api/employees/69137010ac6ff0e791d25414>. The 'Body' tab is active, displaying the following JSON payload:

```
1 {  
2   "_id": "69137010ac6ff0e791d25414",  
3   "firstName": "Nazyrha",  
4   "lastName": "Nassar Salles",  
5   "email": "nahhsalles@gmail.com",  
6   "address": "Toronto",  
7   "salary": 10000,  
8   "department": "IT",  
9   "position": "Developer",  
10 }  
11  
12 }
```

The response shows a 200 OK status with a response time of 68 ms and a body size of 659 B. The response JSON is identical to the payload sent.

2.10 Delete Employee (DELETE <http://localhost:5001/api/employees/{id}>)

The screenshot shows the MongoDB Compass interface. On the left sidebar, under the 'comp3123_assignment2' collection, the 'DEL Delete Employee' option is selected. In the main panel, a DELETE request is being made to <http://localhost:5001/api/employees/69137010ac6ff0e791d25414>. The 'Headers' tab is active, showing the following header:

Key	Value
Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI...

The response shows a 200 OK status with a response time of 91 ms and a body size of 305 B. The response JSON is {"success": true}.

2.11 Simulated error (Without token authorization)

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists collections, environments, flows, and history. In the main area, a collection named "comp3123_assignment2" is selected. A sub-menu for this collection shows various operations: POST Sign Up, POST Log In, POST Create Employee, GET Get Employees, GET Get Employee By ID, GET Search Employee By Name, GET Search Employee By Department, GET Search Employee By Position, PUT Update Employee, and DEL Delete Employee. The "POST Create Employee" operation is currently selected. To the right, a request configuration window is open for "Create Employee". It shows a "POST" method, the URL "http://localhost:5001/api/employees", and a "Headers (8)" tab. Below the headers, there is a table for setting key-value pairs. At the bottom of the request window, the status is shown as "401 Unauthorized" with a response time of 29 ms and a size of 323 B. The response body is displayed as JSON: { "error": "Unauthorized" }.

3. MongoDB Database:

3.1 Database

The screenshot shows the Mongo Express interface. The address bar indicates the connection is to "localhost:8081/db/comp3123_db/". The main title is "Viewing Database: comp3123_db". Below the title, there is a "Collections" section with two entries: "employees" and "users". Each entry has four buttons: "View", "Export", "[JSON]", and "Import". To the right of each entry is a red "Del" button. Above the "employees" entry, there is a "Collection Name" input field and a "+ Create collection" button. Below the collections section is a "Database Stats" table with the following data:

Database Stats	
Collections (incl. system.namespaces)	2
Data Size	394 Bytes
Storage Size	61.4 KB
Avg Obj Size #	197 Bytes
Objects #	2
Indexes #	4
Index Size	123 KB

3.2 Users table

The screenshot shows the Mongo Express interface for the 'users' collection. At the top, there are buttons for 'New Document' and 'New Index'. Below that is a search bar with 'Simple' and 'Advanced' tabs, and a 'Find' button. A red banner at the bottom says 'Delete all 2 documents retrieved'. The main table has columns: '_id', 'name', 'email', 'password', and 'createdAt'. It lists two documents:

_id	name	email	password	createdAt
69136e48ac6ff0e791d25411	Breno Lopes Mafra	brenolopesmafra@gmail.com	\$2b\$10\$eKuzQlkCTbJVYn30Ve9BMe5dUyqsRvNCqTrWdeXMec...	Tue Nov 2025 17:11 GMT+0 (Coordinated Universal Time)
691381f8534b8a6041c0a6b7	Test	test@gmail.com	\$2b\$10\$yYbvLN9ueeB8uYkMSkTDEOUx41yIBSDP3X92XHfYWM...	Tue Nov 2025 18:35 GMT+0 (Coordinated Universal Time)

3.3 Employees Table

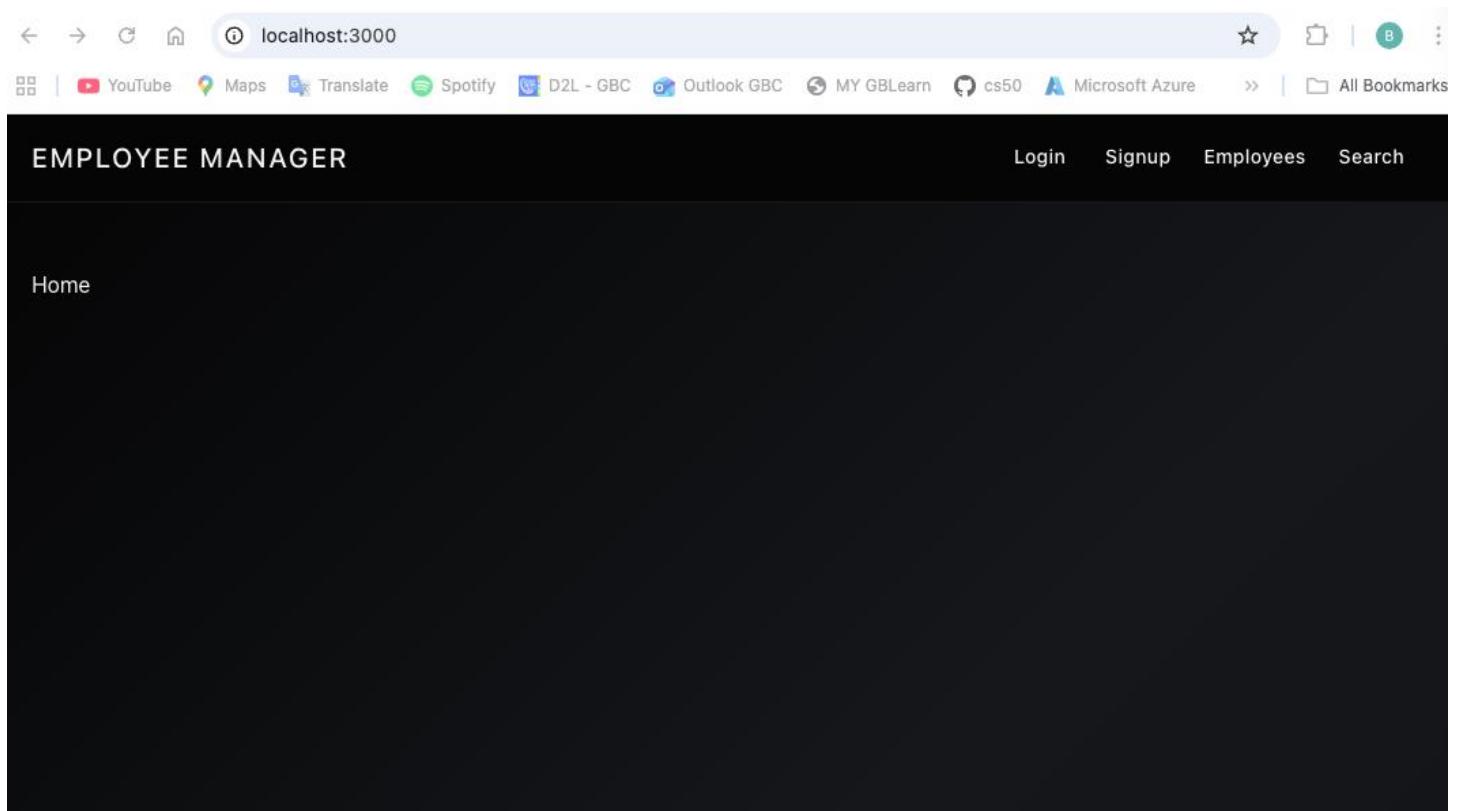
The screenshot shows the Mongo Express interface for the 'employees' collection. At the top, there are buttons for 'New Document' and 'New Index'. Below that is a search bar with 'Simple' and 'Advanced' tabs, and a 'Find' button. A red banner at the bottom says 'Delete all 1 documents retrieved'. The main table has columns: '_id', 'firstName', 'lastName', 'email', 'address', 'salary', 'department', 'position', 'picture', and 'createdAt'. It lists one document:

_id	firstName	lastName	email	address	salary	department	position	picture	createdAt
69139991534b8a6041c0a6d6	Test	Doe	test@gmail.com	Toronto	2300	IT	Staff	1762892177731-27687155.png	Tue Nov 2025 20:16:30 GMT+0 (Coordinated Universal Time)

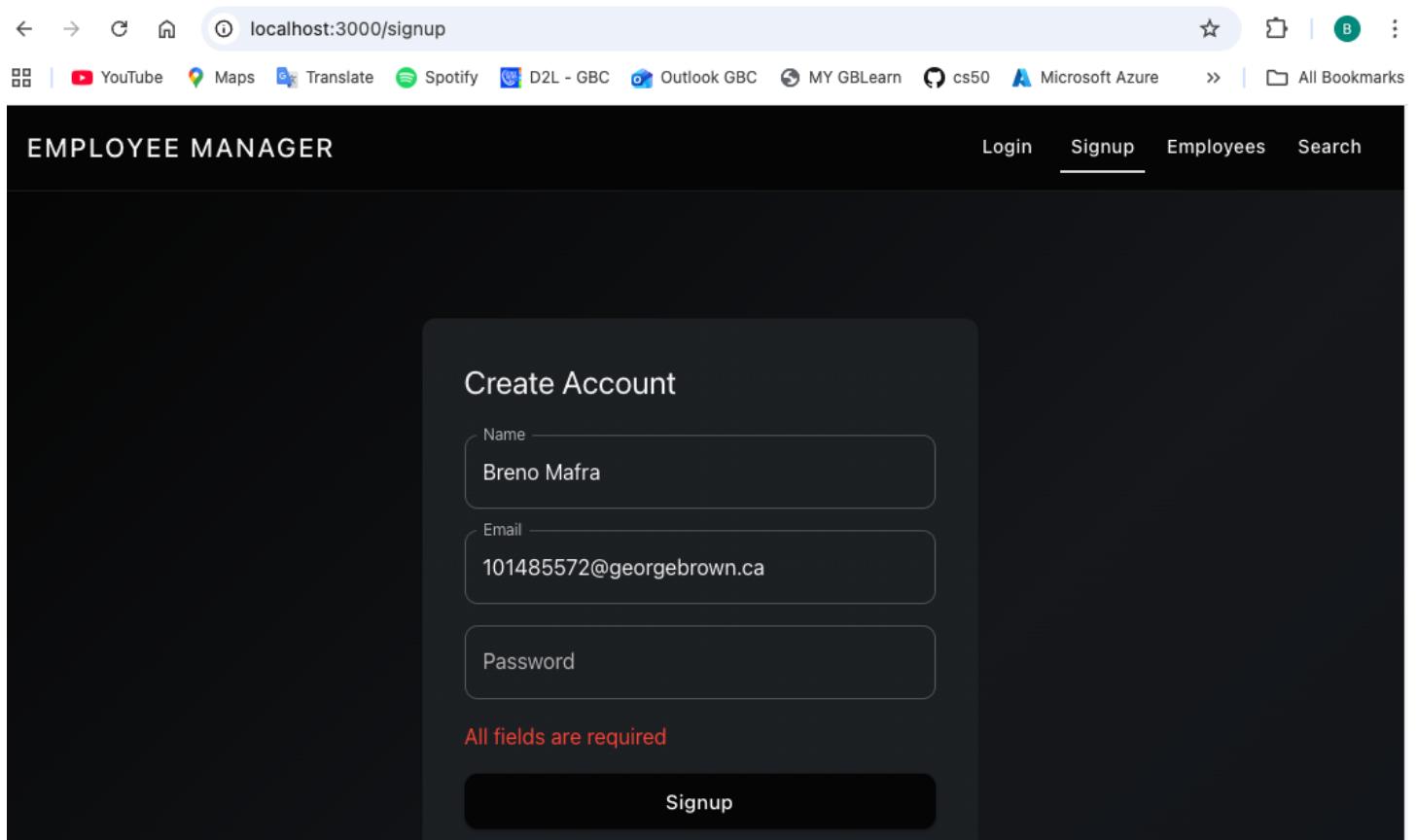
Rename Collection

4. Front-End Screenshots (using Docker on localhost:3000/)

4.1 Home Screen



4.2 Sign Up Screen



localhost:3000/signup

EMPLOYEE MANAGER

Login Signup Employees Search

Create Account

Name: Breno Mafra

Email: 101485572@georgebrown.ca

Password:

Signup

4.3 Log In Screen

localhost:3000/login

EMPLOYEE MANAGER

Login Signup Employees Search

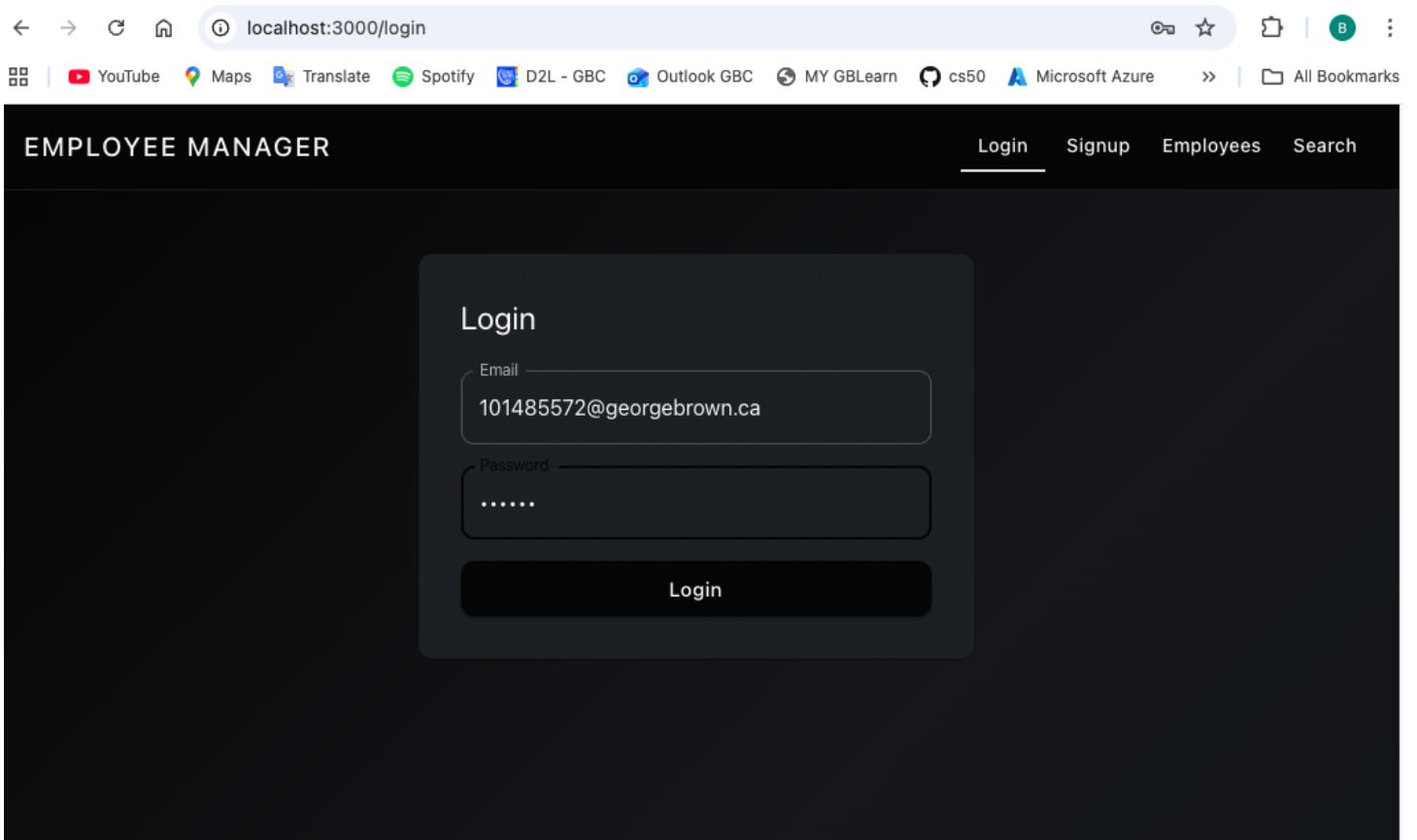
Login

Email: 101485572@georgebrown.ca

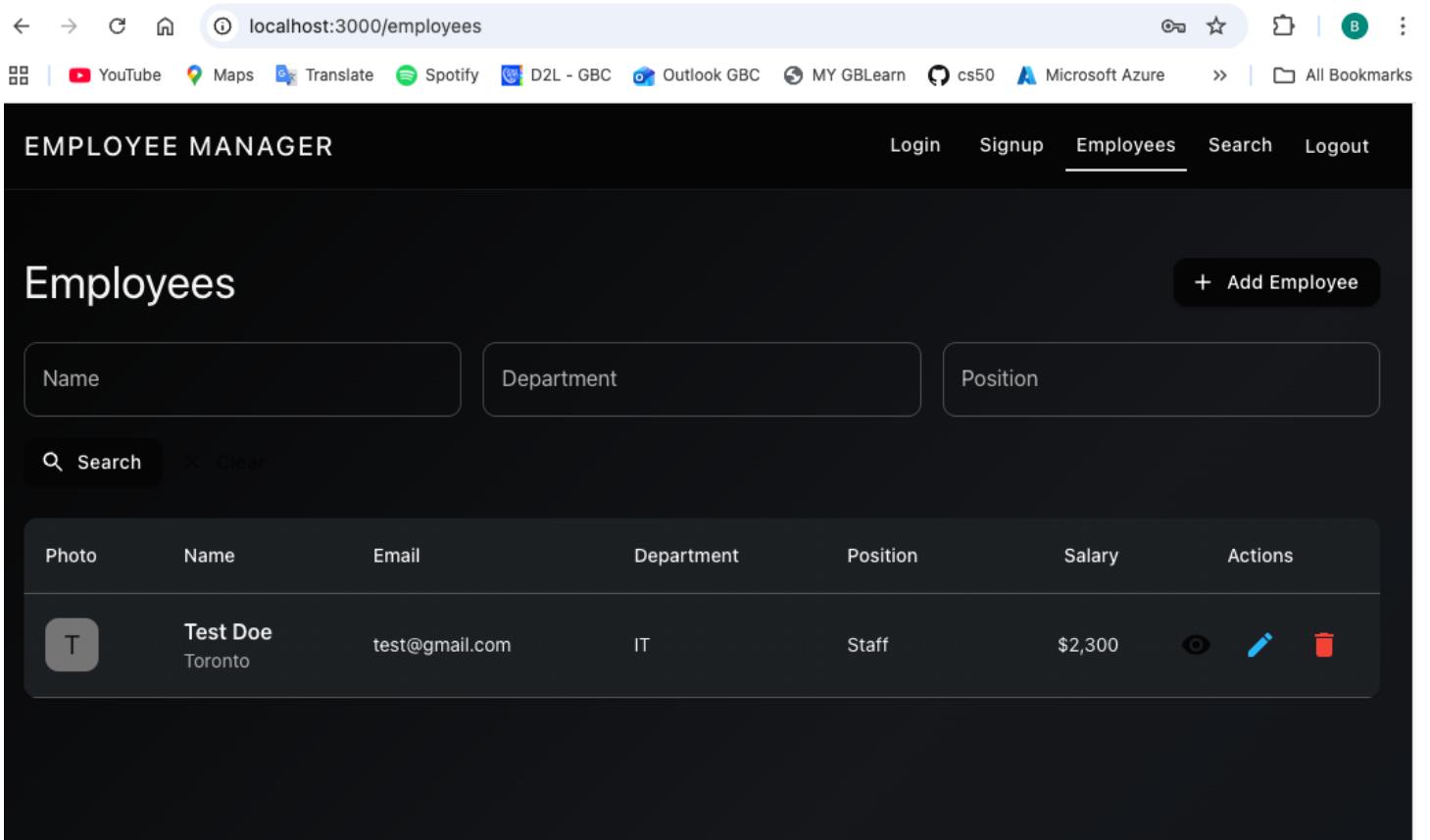
Password

All fields are required

Login



4.4 Employees List Screen



4.5 Add Employee Screen

localhost:3000/employees/new

EMPLOYEE MANAGER

Add Employee

All fields are required

EMPLOYEE MANAGER

Add Employee

First name:

Last name:

Email:

Department:

Position:

Salary:

Address:

localhost:3000/employees/new

EMPLOYEE MANAGER

Add Employee

EMPLOYEE MANAGER

Add Employee



First name:

Last name:

Email:

Department:

Position:

Salary:

Address:

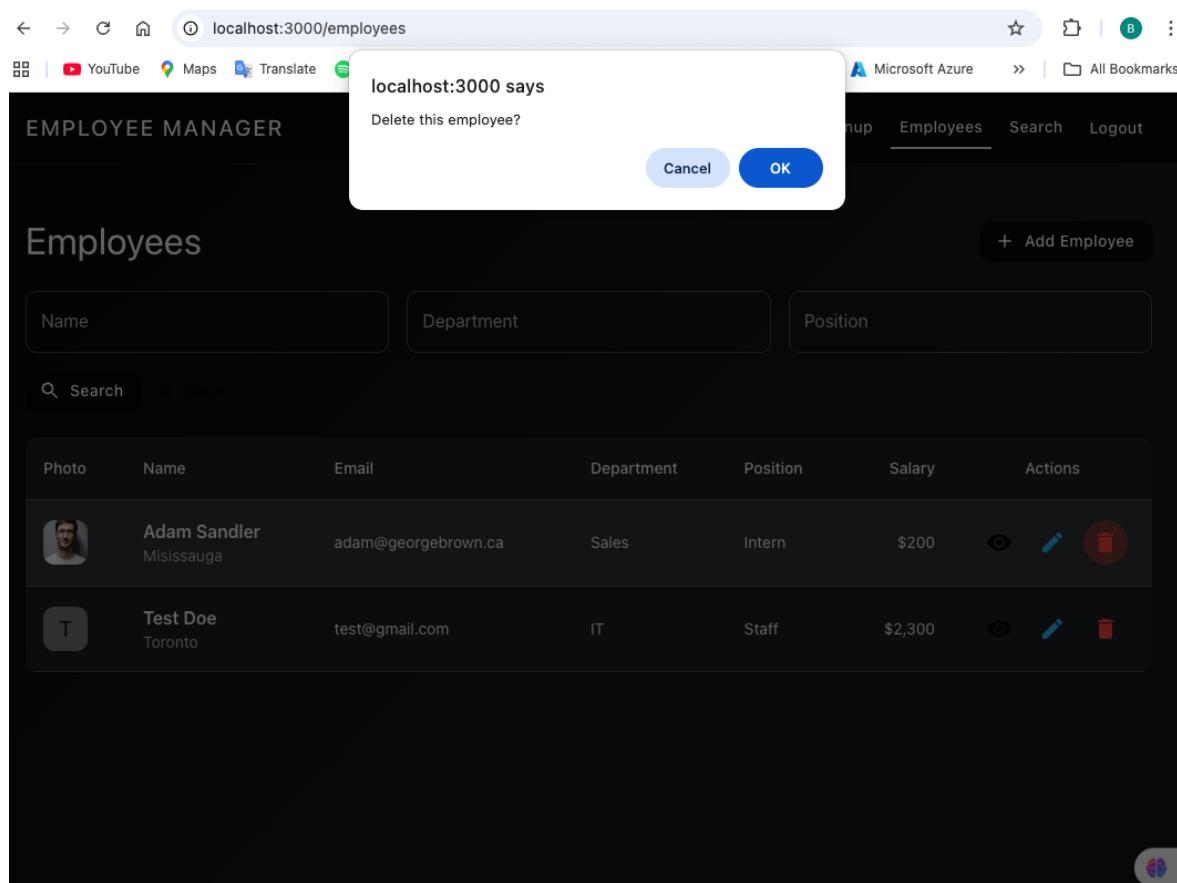
4.6 View Employee Screen

A screenshot of a web browser showing the 'EMPLOYEE MANAGER' application. The URL in the address bar is `localhost:3000/employees/69139e1ebc54233d824e581d`. The page title is 'EMPLOYEE MANAGER'. At the top right are links for 'Login', 'Signup', 'Employees', 'Search', and 'Logout'. Below the title, the employee's name 'Adam Sandler' is displayed with a back arrow and a 'Edit' button with a pencil icon. A 'Delete' button with a trash icon is also present. To the left is a portrait photo of Adam Sandler. To the right are his details: Email: adam@georgebrown.ca, Department: Sales, Position: Intern, Salary: \$200, and Address: Mississauga.

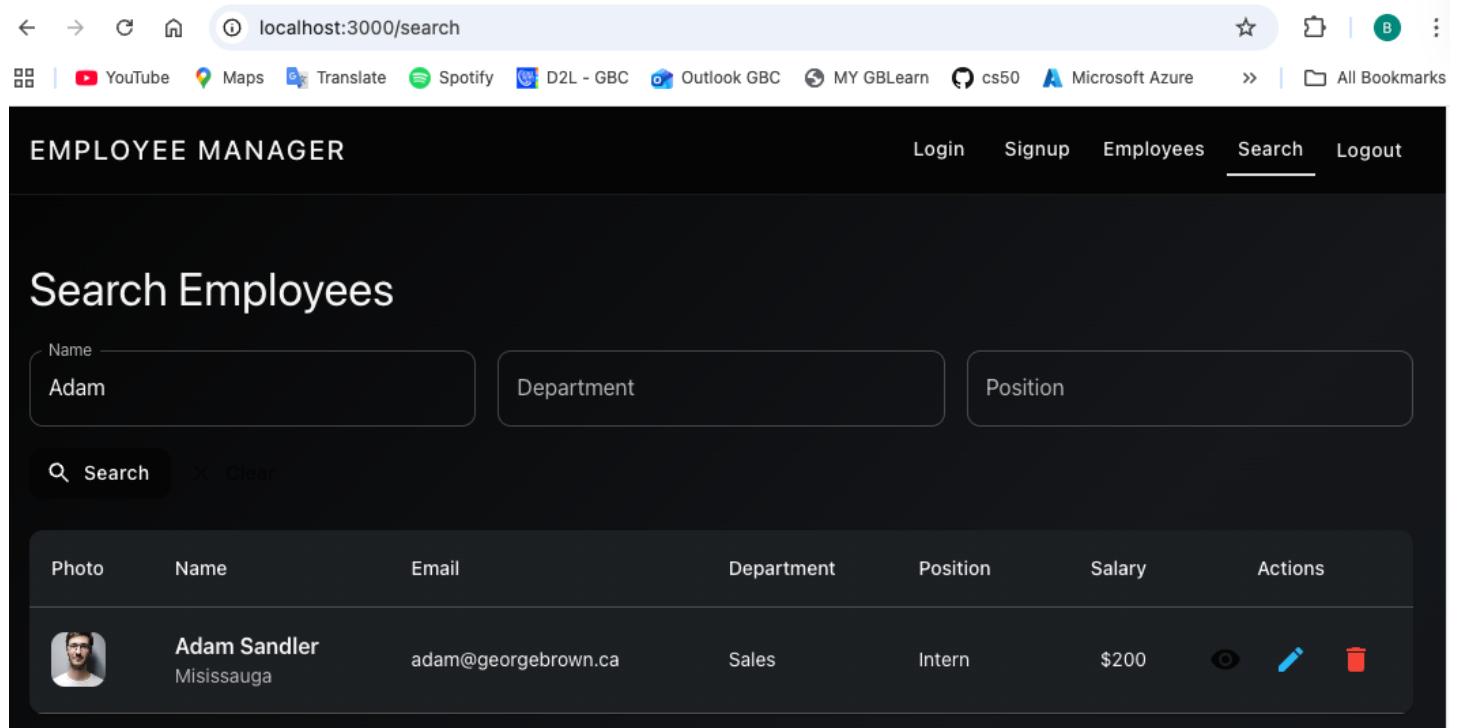
4.7 Update Employee Screen

A screenshot of a web browser showing the 'EMPLOYEE MANAGER' application. The URL in the address bar is `localhost:3000/employees/69139e1ebc54233d824e581d/edit`. The page title is 'EMPLOYEE MANAGER'. At the top right are links for 'Login', 'Signup', 'Employees', 'Search', and 'Logout'. Below the title, the heading 'Edit Employee' is shown with a back arrow. On the left is a portrait photo of Adam Sandler. Below it is a 'Upload photo' button. There are six input fields arranged in two rows of three: First name (Adam), Last name (Sandler), Email (adam@georgebrown.ca), Department (Sales); Position (Intern), Salary (200), and Address (Mississauga). At the bottom are 'Update' and 'Cancel' buttons.

4.8 Delete Employee Screen



4.9 Search Employee Screen



localhost:3000/search

EMPLOYEE MANAGER

Login Signup Employees Search Logout

Search Employees

Name Department Position

IT

Q Search X Clear

Photo	Name	Email	Department	Position	Salary	Actions
	Test Doe Toronto	test@gmail.com	IT	Staff	\$2,300	

localhost:3000/search

EMPLOYEE MANAGER

Login Signup Employees Search Logout

Search Employees

Name Department Position

Intern

Q Search X Clear

Photo	Name	Email	Department	Position	Salary	Actions
	Adam Sandler Mississauga	adam@georgebrown.ca	Sales	Intern	\$200	