

Name: Deep Patel

Stu ID: 101415117

GitHub Repository Link: [https://github.com/Deeppatel91/COMP3133\\_101415117\\_Assignment1](https://github.com/Deeppatel91/COMP3133_101415117_Assignment1)

COMP 3133 Assignment-1

Test It Locally: <http://localhost:5000/graphql>

Sample User for Testing:

Username: ico

Email: ico@example.com

Password: Password@123

Screen Shots:

### 1) User Sign-Up Screenshots: Mutation

The screenshot displays the Postman interface for a GraphQL API. The collection is named 'comp3133' and the specific request is 'Assignmet-1 COMP-3133 / user Signup'. The request is a POST to 'http://localhost:5000/graphql' using the GraphQL format. The query is as follows:

```
1 mutation {
2   signup(username: "imark", email: "mark@gmail.com", password: "Password@123") {
3     token
4     user {
5       id
6       username
7       email
8     }
9   }
10 }
11
```

The response is a 200 OK status with a response time of 304 ms and a body size of 558 B. The JSON response is:

```
{
  "data": {
    "signup": {
      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3YTkwODdmNmY0N2E2ZGRlZmVmNSIsIm1hdCI6MTczOTEzNjgzMiwiZXhwIjoxNzMSMTQwNDMyfQ.1SaunKaKqXeYHwPI2Xqvy87KZLIRqVvmHIWMPvWqXg",
      "user": {
        "id": "67a91f4887f6f47a6ddefef5",
        "username": "imark",
        "email": "mark@gmail.com"
      }
    }
  }
}
```

The interface also shows a sidebar with collections and environments, and a bottom status bar with system information like temperature and time.

## 2) User Login: Query

HTTP

Assignmnet-1 COMP-3133 / user login

Save

Share

POST

http://localhost:5000/graphql

Send

Params

Authorization

Headers (8)

Body •

Scripts

Settings

Cookies

☐ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☒ GraphQL

Auto Fetch

Schema Fetched

QUERY

GRAPHQL VARIABLES

```
1 query {  
2   login(username: "imark", password: "Password@123") {  
3     token  
4     user {  
5       id  
6       username  
7       email  
8     }  
9   }  
}
```

1

Body

Cookies

Headers (8)

Test Results

200 OK • 172 ms • 557 B • Save Response

{ } JSON

Preview

Visualize

```
1 {  
2   "data": {  
3     "login": {  
4       "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
           eyJpZCI6ImYyOTkxZjQwODdmNmY0N2E2ZGRlZmVmNSIsIm1hdCI6MTczOTEZzjkxMywiYXhwIjoxNzM5MTQwWnTEzfQ.  
           xDbL-THG1IQbbhUTfAK_CV0pft2ldpSSSugg8Xu638",  
5       "user": {  
6         "id": "67a91f4087f6f47a6ddeFeF5",  
7         "username": "imark",  
8         "email": "mark@gmail.com"  
9       }  
10    }  
11  }  
12 }
```

### 3) User Database:

comp3133\_\_101415117\_assignment1.users

STORAGE SIZE: 36KB   LOGICAL DATA SIZE: 589B   TOTAL DOCUMENTS: 3   INDEXES TOTAL SIZE: 108KB

Find   Indexes   Schema Anti-Patterns 0   Aggregation   Search Indexes

Generate queries from natural language in Compass [↗](#) INSERT DOCUMENT

Filter [↗](#)

Type a query: { field: 'value' }

Reset   Apply   Options [▶](#)

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('67a913a947638966d69c2f27')
username: "ico"
email: "ico@example.com"
password: "$2a$10$APeWv4YmWxHBSBWZn3RI3.BkkMTCVgMBXW6wt6T8gkgjDv8mf8LEK"
created_at: 2025-02-09T20:44:25.257+00:00
updated_at: 2025-02-09T20:44:25.257+00:00
__v: 0
```

```
_id: ObjectId('67a91f4087f6f47a6dde5f5')
username: "imark"
email: "mark@gmail.com"
password: "$2a$10$.86eU8YrIRI5vL4N3EFUJeV7N4cCg4xAV4TYDQPMWIAWN5uU0pUuq"
created_at: 2025-02-09T21:22:53.228+00:00
```

#### 4) Add Employee Protected JWT Token Required Authorized -Mutation

The screenshot shows the GraphQL Playground interface for a POST request to `http://localhost:5000/graphql`. The request body is a GraphQL mutation to add a new employee with the following details: first\_name: "melina", last\_name: "hazeri", email: "melina@gmail.com", gender: "Female", designation: "devops", salary: 88000.00, date\_of\_joining: "2023-09-11", and department: "CICD". The response is a 200 OK status with a JSON body containing the employee's ID and photo.

**QUERY**

```
1 mutation {
2   addEmployee(
3     first_name: "melina"
4     last_name: "hazeri"
5     email: "melina@gmail.com"
6     gender: "Female"
7     designation: "devops"
8     salary: 88000.00
9     date_of_joining: "2023-09-11"
10    department: "CICD"
11  )
12 }
```

**GRAPHQL VARIABLES**

```
1
```

**Body** Cookies Headers (8) Test Results | 200 OK • 139 ms • 428 B • Save Response

**JSON** Preview Visualize

```
1 {
2   "data": {
3     "addEmployee": {
4       "id": "67a9207987f6f47a6ddefeb",
5       "first_name": "melina",
6       "last_name": "hazeri",
7       "email": "melina@gmail.com",
8       "employee_photo": "melina.jpeg"
9     }
10  }
11 }
```

Postbot Ctrl Alt P

#### 5) Update Employee Protected JWT Token Required Authorized - Mutation

The screenshot shows the GraphQL Playground interface for a POST request to `http://localhost:5000/graphql`. The request body is a GraphQL mutation to update an employee with the following details: id: "67a9207987f6f47a6ddefeb", first\_name: "Melian", and salary: 89999.00. The response is a 200 OK status with a JSON body containing the updated employee's ID, first\_name, and salary.

**QUERY**

```
1 mutation {
2   updateEmployee(
3     id: "67a9207987f6f47a6ddefeb"
4     first_name: "Melian"
5     salary: 89999.00
6   ) {
7     id
8     first_name
9     salary
10  }
11 }
```

**GRAPHQL VARIABLES**

```
1
```

**Body** Cookies Headers (8) Test Results | 200 OK • 177 ms • 366 B • Save Response

**JSON** Preview Visualize

```
1 {
2   "data": {
3     "updateEmployee": {
4       "id": "67a9207987f6f47a6ddefeb",
5       "first_name": "Melian",
6       "salary": 89999
7     }
8   }
9 }
```

Postbot Ctrl Alt P

## 6) Get All Employee Protected JWT Token Required Authorized – Query

The screenshot shows the GraphQL Playground interface. The URL is `http://localhost:5000/graphql` and the method is `POST`. The query is:

```
1 query {
2   getAllEmployees {
3     id
4     first_name
5     last_name
6     email
7     designation
8     salary
9     employee_photo
10    department
11  }
12 }
```

The response is a 200 OK status with a response time of 89 ms and a size of 1.48 KB. The JSON response is:

```
{
  "data": {
    "getAllEmployees": [
      {
        "id": "67a8e6ccaa72bf56cf7a1f3",
        "first_name": "Georgia",
        "last_name": "Smith",
        "email": "alice@example.com",
        "designation": "Software Engineer",
        "salary": 90888,
        "employee_photo": "alice.jpg",
        "department": "Engineering"
      }
    ]
  }
}
```

## 7) Employee By ID Protected JWT Token Required Authorized – Query

The screenshot shows the GraphQL Playground interface. The URL is `http://localhost:5000/graphql` and the method is `POST`. The query is:

```
1 query {
2   getEmployeeById(id: "67a8e6ccaa72bf56cf7a1f3") {
3     id
4     first_name
5     last_name
6     email
7     designation
8     salary
9   }
10 }
11
12
13
14 }
```

The response is a 200 OK status with a response time of 97 ms and a size of 507 B. The JSON response is:

```
{
  "data": {
    "getEmployeeById": {
      "id": "67a8e6ccaa72bf56cf7a1f3",
      "first_name": "Georgia",
      "last_name": "Smith",
      "email": "alice@example.com",
      "designation": "Software Engineer",
      "salary": 90888,
      "department": "Engineering",
      "employee_photo": "alice.jpg"
    }
  }
}
```

## 8) Delete Employee Protected JWT Token Required Authorized – Mutation

HTTP Assignmnet-1 COMP-3133 / delete employee

POST http://localhost:5000/graphql Send

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL Auto Fetch Schema Fetched

QUERY

```
1 mutation {
2   deleteEmployee(id: "67a8f6b5698c0dbbf1c72c2d")
3 }
4
```

GRAPHQL VARIABLES

```
1
```

Body Cookies Headers (8) Test Results 200 OK • 209 ms • 360 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "data": {
3     "deleteEmployee": "Employee with ID 67a8f6b5698c0dbbf1c72c2d deleted successfully"
4   }
5 }
```

## 9) Search Designation Employee Protected JWT Token Required Authorized – Query

HTTP Assignmnet-1 COMP-3133 / designated post

POST http://localhost:5000/graphql Send

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL Auto Fetch Schema Fetched

QUERY

```
1 query {
2   searchEmployeeByDesignation(designation: "DOGE") {
3     id
4     first_name
5     last_name
6     email
7     designation
8     salary
9     department
10    employee_photo
11  }
12 }
13
```

GRAPHQL VARIABLES

```
1
```

Body Cookies Headers (8) Test Results 200 OK • 102 ms • 503 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "data": {
3     "searchEmployeeByDesignation": [
4       {
5         "id": "67a91b835068060f1bfceb5e",
6         "first_name": "trump",
7         "last_name": "trump",
8         "email": "trump@example.com",
9         "designation": "DOGE",
10        "salary": 7999,
11        "department": "ICE",
12        "employee_photo": "republican.jpeg"
13      }
14    ]
15  }
16 }
```

## 10) Search Department Employee Protected JWT Token Required Authorized – Query

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:5000/graphql`
- Method:** POST
- Body Type:** GraphQL
- Query:**

```
1 query {
2   searchEmployeeByDepartment(department: "CICD") {
3     id
4     first_name
5     last_name
6     email
7     designation
8   }
9 }
```
- GraphQL Variables:** 1
- Status:** 200 OK, 91 ms, 503 B
- Response Body (JSON):**

```
1 {
2   "data": {
3     "searchEmployeeByDepartment": [
4       {
5         "id": "67a9207987f6f47a6ddefefb",
6         "first_name": "Melian",
7         "last_name": "hazeri",
8         "email": "melina@gmail.com",
9         "designation": "devops",
10        "salary": 89999,
11        "department": "CICD",
12        "employee_photo": "melina.jpeg"
13      }
14    ]
15  }
16 }
```

## 11) Employee Database Documentation:

The screenshot shows a database management interface with the following details:

- Database:** comp3133\_\_101415117\_assignment1.employees
- Storage Size:** 36KB
- Logical Data Size:** 1.38KB
- Total Documents:** 5
- Indexes Total Size:** 72KB
- Find Tab:** Active, showing a list of documents.
- Document 1:**

```
{
  "_id": ObjectId('67a9207987f6f47a6ddefefb'),
  "first_name": "Melian",
  "last_name": "hazeri",
  "email": "melina@gmail.com",
  "gender": "Female",
  "designation": "devops",
  "salary": 89999,
  "date_of_joining": "2023-09-11T00:00:00.000+00:00",
  "department": "CICD",
  "employee_photo": "melina.jpeg",
  "created_at": "2025-02-09T21:39:05.579+00:00",
  "updated_at": "2025-02-10T16:14:20.981+00:00",
  "__v": 0
}
```

- Passwords are hashed before saving.
- JWT is required for protected routes (except login/signup).
- GraphQL API tested using Postman.
- Postman collection is attached for easy testing.