Elizabeth Thomas
ID: 101097106

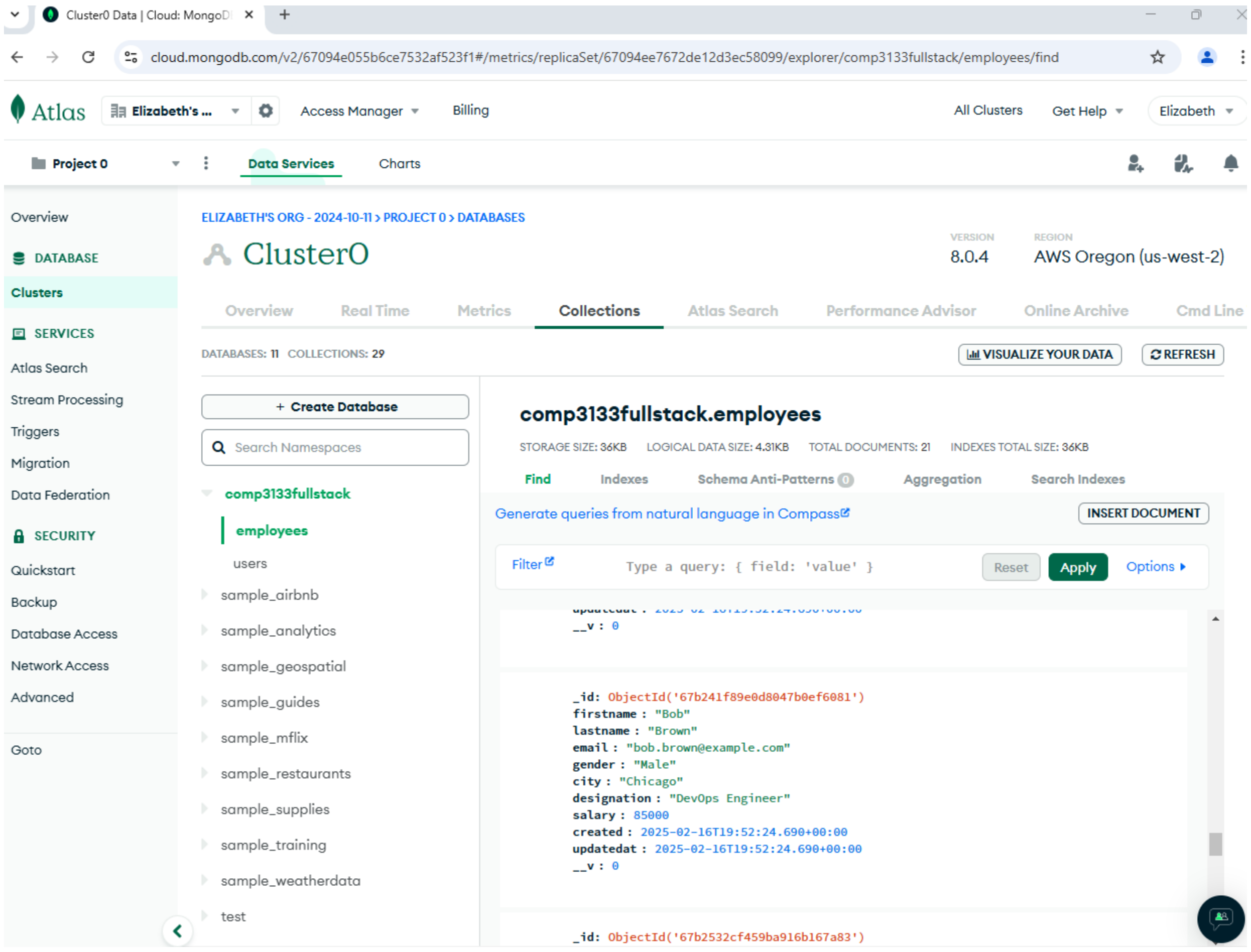
COMP 3133 – Full Stack Development – II

Assignment – I

Submission Checklist:

Make single docx file with all labeled screenshots.

MongoDB Console Screenshots: o Provide screenshots of your MongoDB database showing collections.



```
__v : 0

_id: ObjectId('67b241f89e0d8047b0ef6081')
firstname : "Bob"
lastname : "Brown"
email : "bob.brown@example.com"
gender : "Male"
city : "Chicago"
designation : "DevOps Engineer"
salary : 85000
created : 2025-02-16T19:52:24.690+00:00
updatedat : 2025-02-16T19:52:24.690+00:00
__v : 0

_id: ObjectId('67b2532cf459ba916b167a83')
```

Elizabeth Thomas
ID: 101097106

2. Postman API Collection: o Export your Postman collection and include it in your submission.

✓ Submitted in D2L

4. Project ZIP File:

✓ Submitted in D2L

5. GitHub Project Link: Provide the link to your GitHub repository.

https://github.com/eliza2526/comp3133_assignment1

6. Sample User Detail: Include a sample user's details for testing login.

- username: user1 password: password1
- username: user2 password: password2
- username: newuser password: password123

o Add any comments or notes that could be helpful.

Employee Ids for CRUD Operations

- 67b1880b6e2a5d4d5d96f88a
- 67b2532cf459ba916b167a83
- 67b1880b6e2a5d4d5d96f88b
- 67b1880b6e2a5d4d5d96f88c
- 67b1880b6e2a5d4d5d96f88d
- 67b188e16e523fd9ee8c3819
- 67b191d9af7d514b043db89c

API endpoint

- http://localhost:4000/graphql

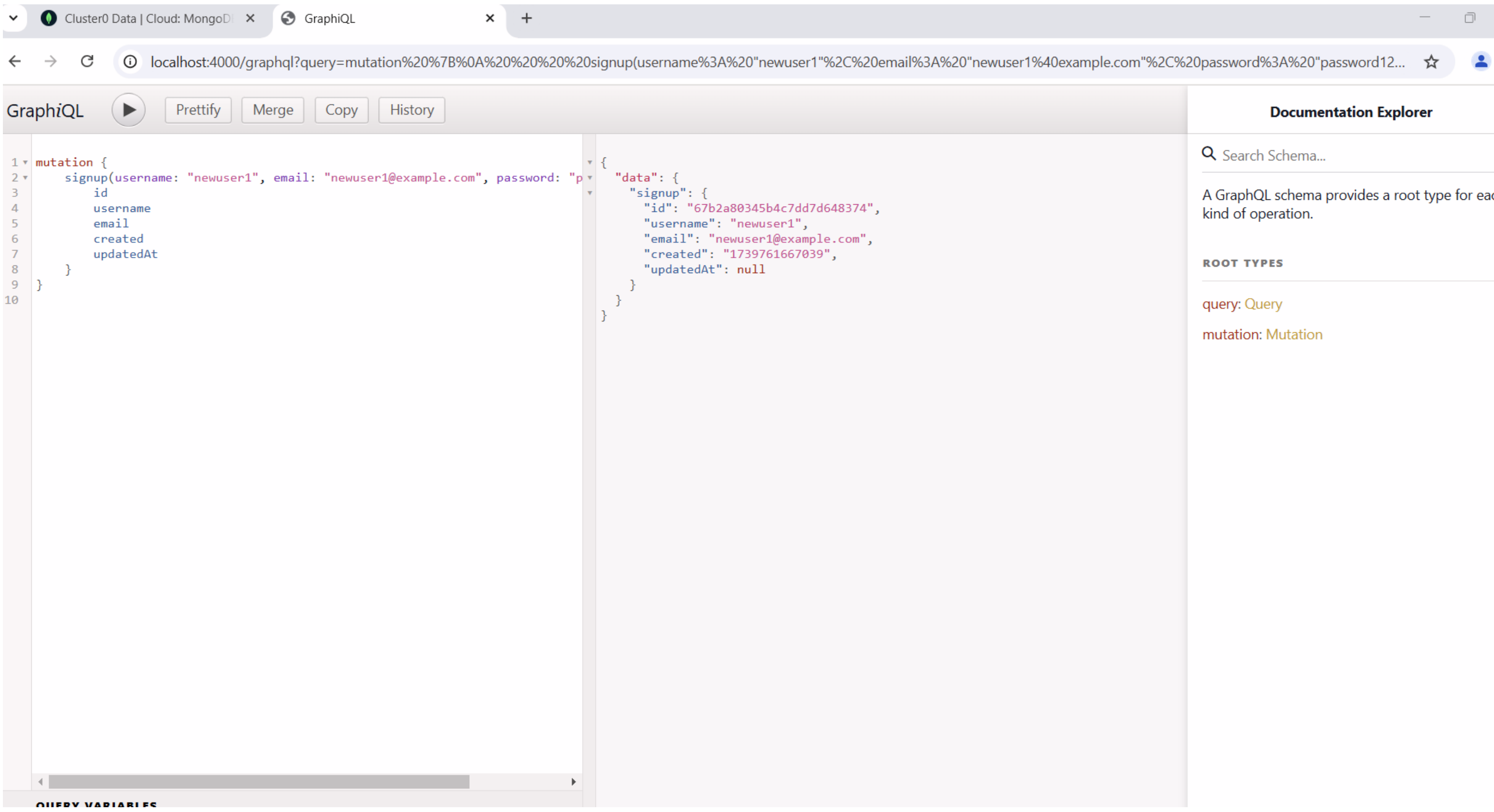
Elizabeth Thomas
ID: 101097106

3. Screenshots: Include screenshots of each API being tested along with responses.

GraphQL Implementation

Following is the list of to develop which accept all data as JSON Object whenever needed:

1 Mutation Signup Allow user to create new account


Cluster0 Data | Cloud: MongoD
GraphiQL
localhost:4000/graphql?query=mutation%20%7B%0A%20%20%20%20signup(username%3A%20"newuser1"%2C%20email%3A%20"newuser1%40example.com"%2C%20password%3A%20"password12...
GraphiQL
Prettify
Merge
Copy
History
Documentation Explorer
Search Schema...
QUERY VARIABLES

```
mutation {
    signup(username: "newuser1", email: "newuser1@example.com", password: "p
        id
        username
        email
        created
        updatedAt
    }
}
```

```
{
  "data": {
    "signup": {
      "id": "67b2a80345b4c7dd7d648374",
      "username": "newuser1",
      "email": "newuser1@example.com",
      "created": "1739761667039",
      "updatedAt": null
    }
  }
}
```

A GraphQL schema provides a root type for ea
kind of operation.

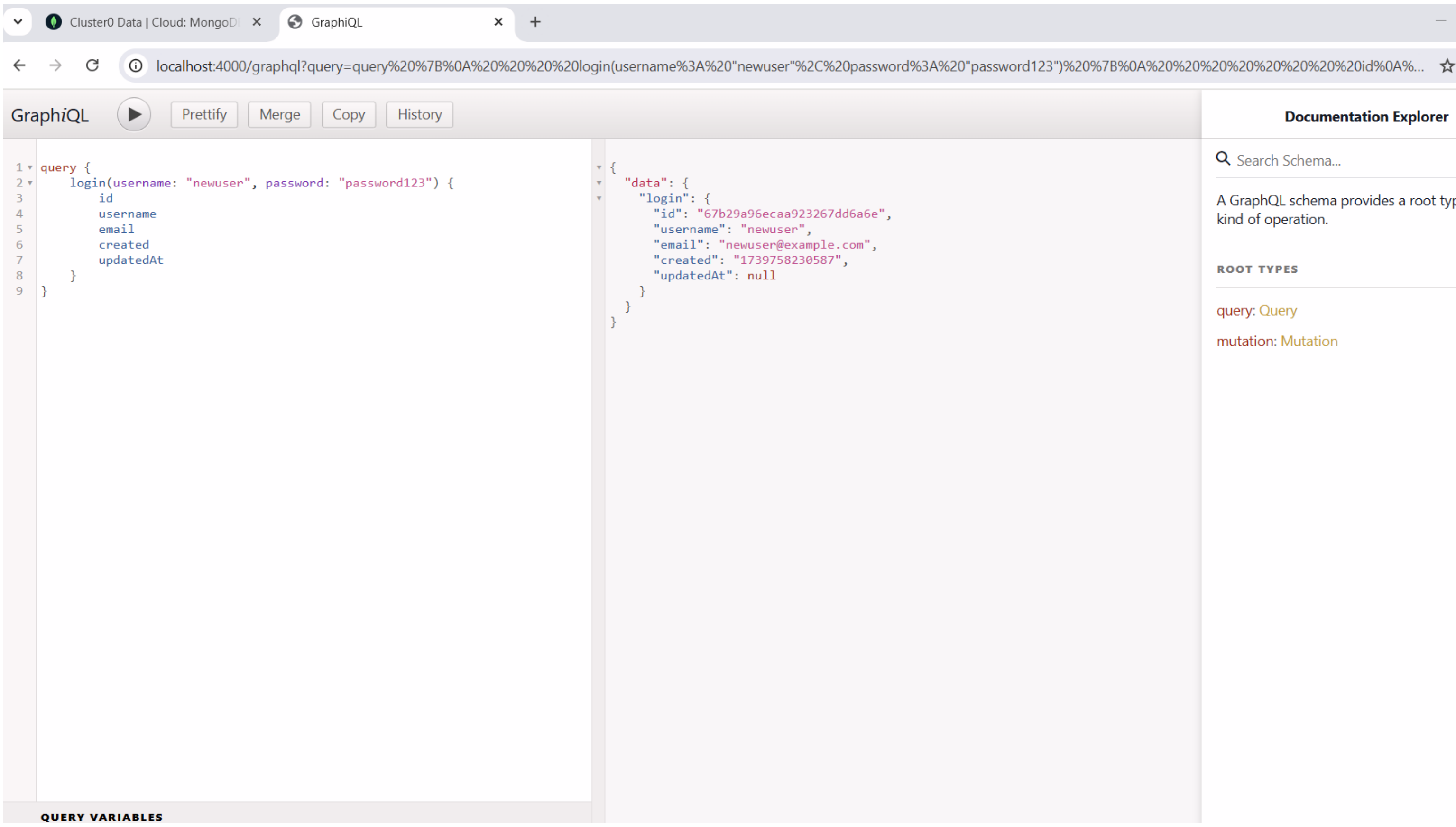
ROOT TYPES

query: Query

mutation: Mutation

Elizabeth Thomas
ID: 101097106

2 Query Login Allow user to access the system



localhost:4000/graphql?query=query%20%7B%0A%20%20%20%20login(username%3A%20"newuser"%2C%20password%3A%20"password123")%20%7B%0A%20%20%20%20%20%20%20%20id%0A%...

```
query {
    login(username: "newuser", password: "password123") {
        id
        username
        email
        created
        updatedAt
    }
}
```

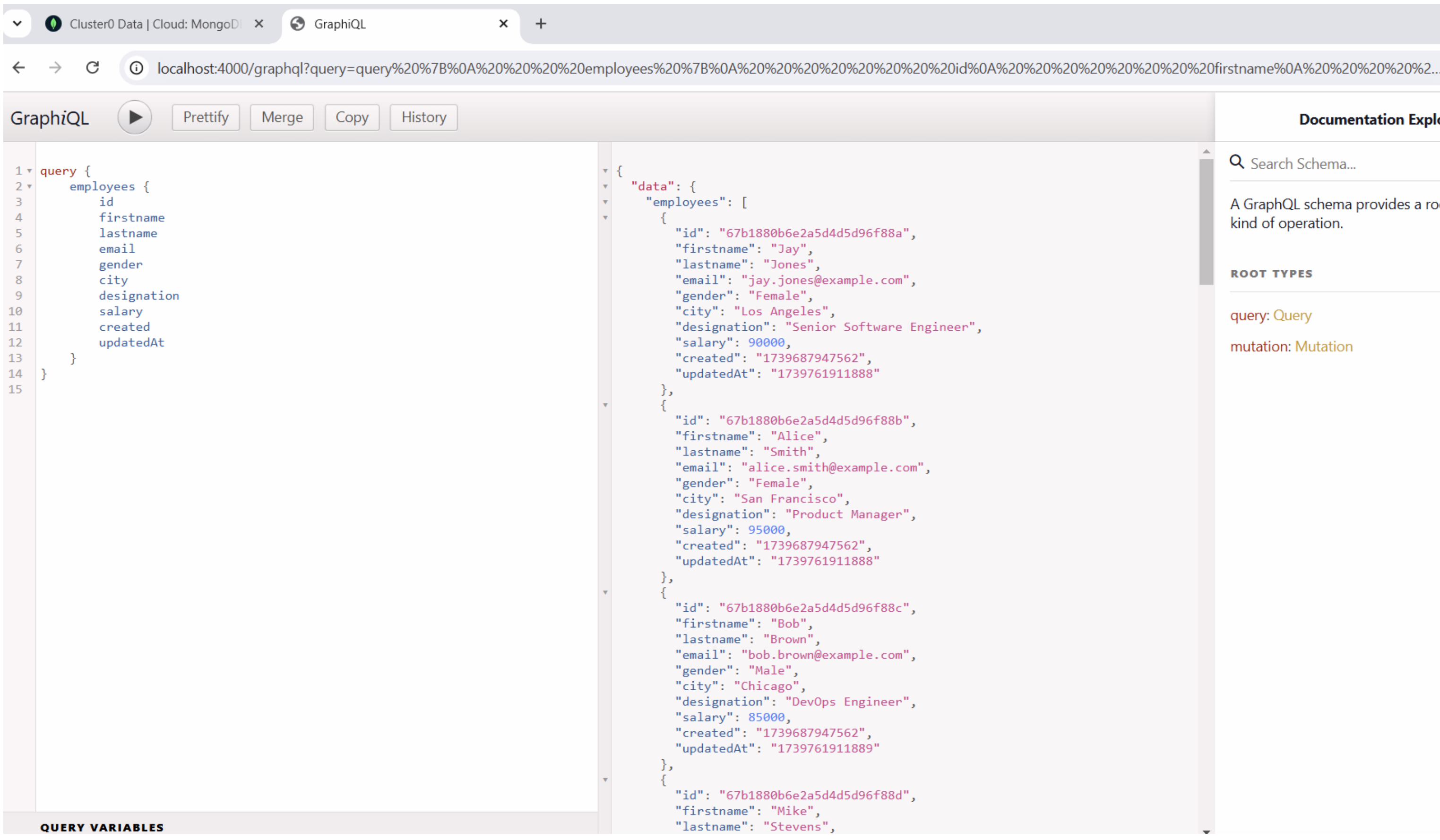
```
{
  "data": {
    "login": {
      "id": "67b29a96ecaa923267dd6a6e",
      "username": "newuser",
      "email": "newuser@example.com",
      "created": "1739758230587",
      "updatedAt": null
    }
  }
}
```

A GraphQL schema provides a root typ
kind of operation.

query: Query
mutation: Mutation

Elizabeth Thomas
ID: 101097106

3 Query Get all employees User can get all employee list


Cluster0 Data | Cloud: MongoD
GraphiQL
localhost:4000/graphql?query=query%20%7B%0A%20%20%20%20employees%20%7B%0A%20%20%20%20%20%20%20%20id%0A%20%20%20%20%20%20%20%20firstname%0A%20%20%20%20%2...
GraphiQL
Prettify
Merge
Copy
History
Documentation Explo
Search Schema...
QUERY VARIABLES

```
query {
    employees {
        id
        firstname
        lastname
        email
        gender
        city
        designation
        salary
        created
        updatedAt
    }
}
```

```
{
  "data": {
    "employees": [
      {
        "id": "67b1880b6e2a5d4d5d96f88a",
        "firstname": "Jay",
        "lastname": "Jones",
        "email": "jay.jones@example.com",
        "gender": "Female",
        "city": "Los Angeles",
        "designation": "Senior Software Engineer",
        "salary": 90000,
        "created": "1739687947562",
        "updatedAt": "1739761911888"
      },
      {
        "id": "67b1880b6e2a5d4d5d96f88b",
        "firstname": "Alice",
        "lastname": "Smith",
        "email": "alice.smith@example.com",
        "gender": "Female",
        "city": "San Francisco",
        "designation": "Product Manager",
        "salary": 95000,
        "created": "1739687947562",
        "updatedAt": "1739761911888"
      },
      {
        "id": "67b1880b6e2a5d4d5d96f88c",
        "firstname": "Bob",
        "lastname": "Brown",
        "email": "bob.brown@example.com",
        "gender": "Male",
        "city": "Chicago",
        "designation": "DevOps Engineer",
        "salary": 85000,
        "created": "1739687947562",
        "updatedAt": "1739761911889"
      },
      {
        "id": "67b1880b6e2a5d4d5d96f88d",
        "firstname": "Mike",
        "lastname": "Stevens",
```

A GraphQL schema provides a ro
kind of operation.

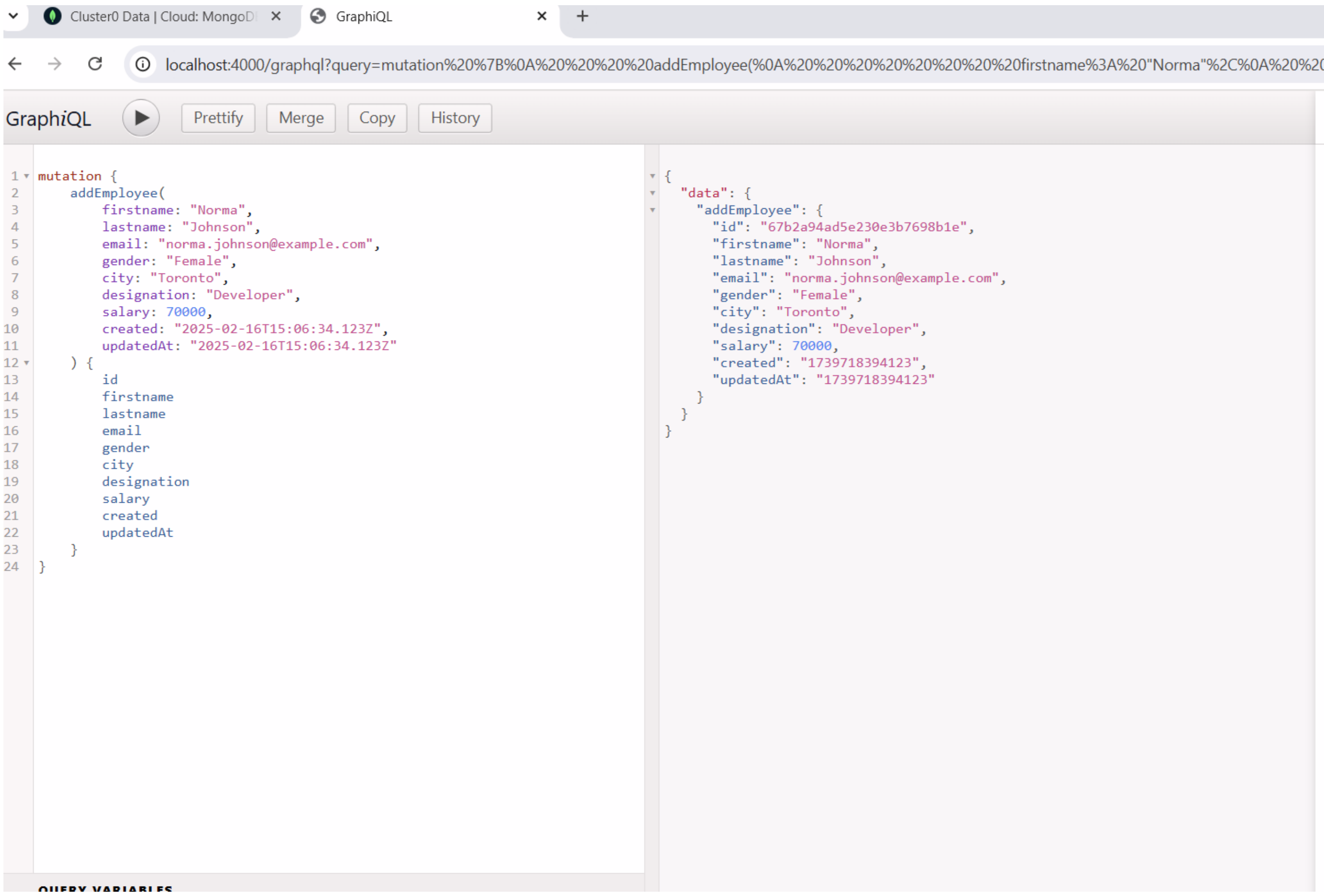
ROOT TYPES

query: Query

mutation: Mutation

Elizabeth Thomas
ID: 101097106

4 Mutation Add New employee User can create new employee



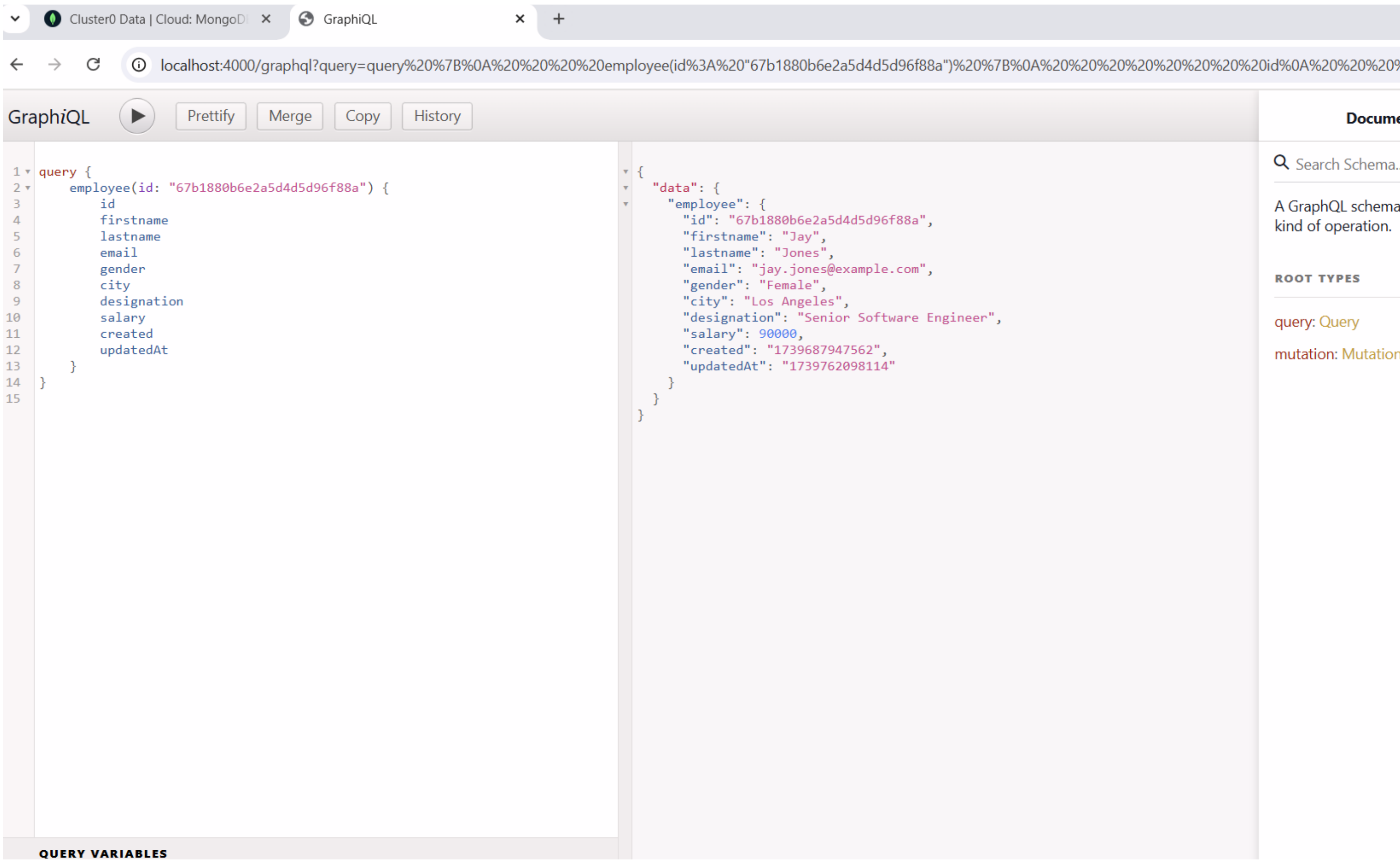
localhost:4000/graphql?query=mutation%20%7B%0A%20%20%20%20addEmployee(%0A%20%20%20%20%20%20%20%20firstname%3A%20"Norma"%2C%0A%20%20

```
mutation {
    addEmployee(
        firstname: "Norma",
        lastname: "Johnson",
        email: "norma.johnson@example.com",
        gender: "Female",
        city: "Toronto",
        designation: "Developer",
        salary: 70000,
        created: "2025-02-16T15:06:34.123Z",
        updatedAt: "2025-02-16T15:06:34.123Z"
    ) {
        id
        firstname
        lastname
        email
        gender
        city
        designation
        salary
        created
        updatedAt
    }
}
```

```
{
  "data": {
    "addEmployee": {
      "id": "67b2a94ad5e230e3b7698b1e",
      "firstname": "Norma",
      "lastname": "Johnson",
      "email": "norma.johnson@example.com",
      "gender": "Female",
      "city": "Toronto",
      "designation": "Developer",
      "salary": 70000,
      "created": "1739718394123",
      "updatedAt": "1739718394123"
    }
  }
}
```

Elizabeth Thomas
ID: 101097106

5 Query Search employee by eid User can get employee details by employee id


Cluster0 Data | Cloud: MongoD
GraphiQL
localhost:4000/graphql?query=query%20%7B%0A%20%20%20%20employee(id%3A%20"67b1880b6e2a5d4d5d96f88a")%20%7B%0A%20%20%20%20%20%20%20%20id%0A%20%20%20%20

GraphiQL Prettify Merge Copy History

```
query {
    employee(id: "67b1880b6e2a5d4d5d96f88a") {
        id
        firstname
        lastname
        email
        gender
        city
        designation
        salary
        created
        updatedAt
    }
}
```

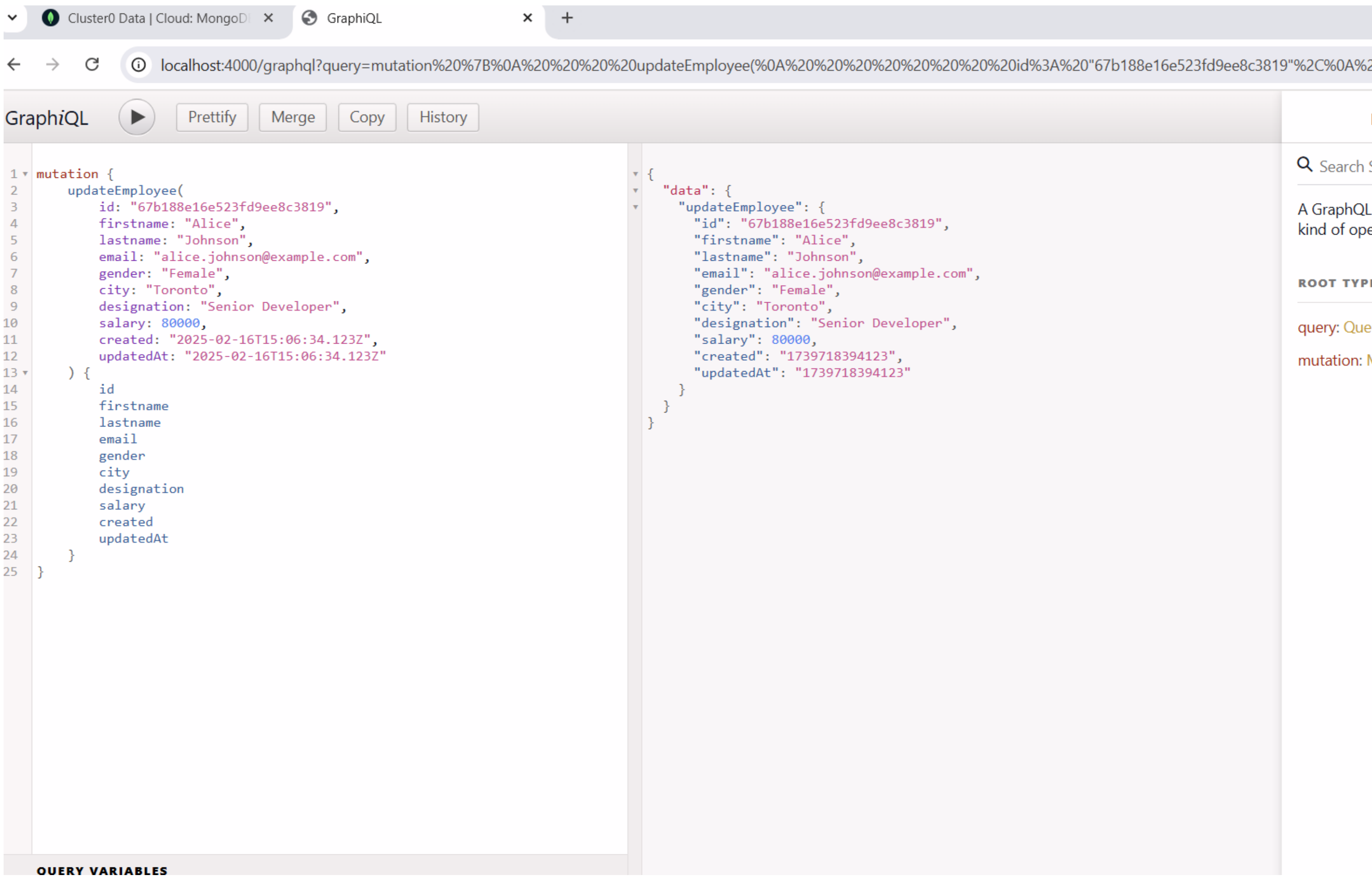
```
{
  "data": {
    "employee": {
      "id": "67b1880b6e2a5d4d5d96f88a",
      "firstname": "Jay",
      "lastname": "Jones",
      "email": "jay.jones@example.com",
      "gender": "Female",
      "city": "Los Angeles",
      "designation": "Senior Software Engineer",
      "salary": 90000,
      "created": "1739687947562",
      "updatedAt": "1739762098114"
    }
  }
}
```

QUERY VARIABLES

Docume
Search Schema.
A GraphQL schema
kind of operation.
ROOT TYPES
query: Query
mutation: Mutation

Elizabeth Thomas
ID: 101097106

6 Mutation Update employee by eid User can update employee details

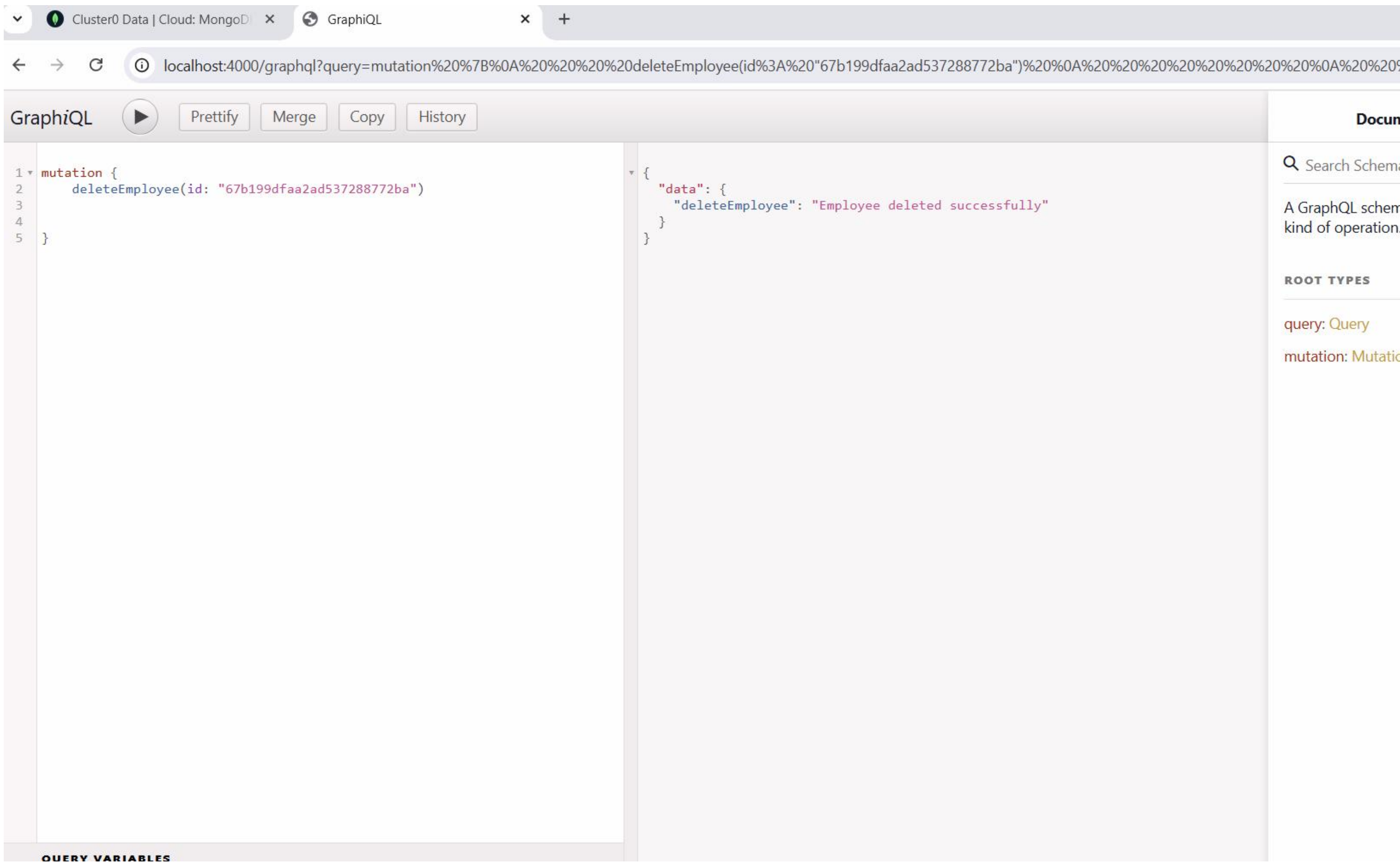

Cluster0 Data | Cloud: MongoD
GraphiQL
localhost:4000/graphql?query=mutation%20%7B%0A%20%20%20%20updateEmployee(%0A%20%20%20%20%20%20%20%20id%3A%20"67b188e16e523fd9ee8c3819"%2C%0A%2
GraphiQL
Prettify
Merge
Copy
History
Search
A GraphQL
kind of ope
ROOT TYPE
query: Que
mutation: M
QUERY VARIABLES

```
mutation {
    updateEmployee(
        id: "67b188e16e523fd9ee8c3819",
        firstname: "Alice",
        lastname: "Johnson",
        email: "alice.johnson@example.com",
        gender: "Female",
        city: "Toronto",
        designation: "Senior Developer",
        salary: 80000,
        created: "2025-02-16T15:06:34.123Z",
        updatedAt: "2025-02-16T15:06:34.123Z"
    ) {
        id
        firstname
        lastname
        email
        gender
        city
        designation
        salary
        created
        updatedAt
    }
}
```

```
{
  "data": {
    "updateEmployee": {
      "id": "67b188e16e523fd9ee8c3819",
      "firstname": "Alice",
      "lastname": "Johnson",
      "email": "alice.johnson@example.com",
      "gender": "Female",
      "city": "Toronto",
      "designation": "Senior Developer",
      "salary": 80000,
      "created": "1739718394123",
      "updatedAt": "1739718394123"
    }
  }
}
```

Elizabeth Thomas
ID: 101097106

7 Mutation Delete employee by eid User can delete employee by employee id

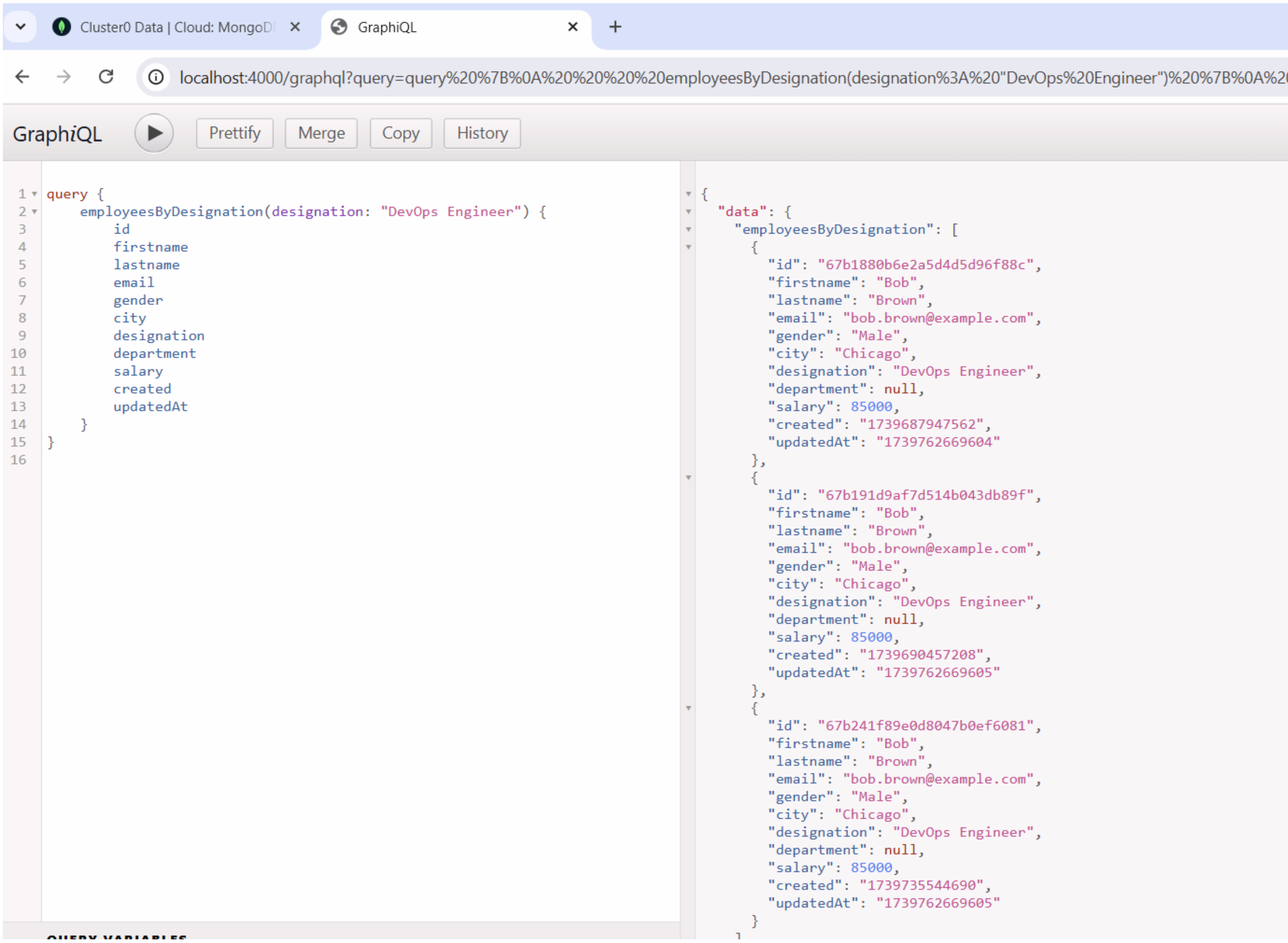

Cluster0 Data | Cloud: MongoD
GraphiQL
localhost:4000/graphql?query=mutation%20%7B%0A%20%20%20%20deleteEmployee(id%3A%20"67b199dfaa2ad537288772ba")%20%0A%20%20%20%20%20%20%20%20%0A%20%20
GraphiQL
Prettify
Merge
Copy
History
Docun
Search Schema
A GraphQL schem
kind of operation
ROOT TYPES
query: Query
mutation: Mutatic
QUERY VARIABLES

```
mutation {
    deleteEmployee(id: "67b199dfaa2ad537288772ba")


}
```

```
{
  "data": {
    "deleteEmployee": "Employee deleted successfully"
  }
}
```

Elizabeth Thomas
ID: 101097106

8 Query Search Employee by designation or department User can get employee list by designation or department


Cluster0 Data | Cloud: MongoD
GraphiQL

localhost:4000/graphql?query=query%20%7B%0A%20%20%20%20employeesByDesignation(designation%3A%20"DevOps%20Engineer")%20%7B%0A%2

GraphiQL Prettify Merge Copy History

```
query {
    employeesByDesignation(designation: "DevOps Engineer") {
        id
        firstname
        lastname
        email
        gender
        city
        designation
        department
        salary
        created
        updatedAt
    }
}
```

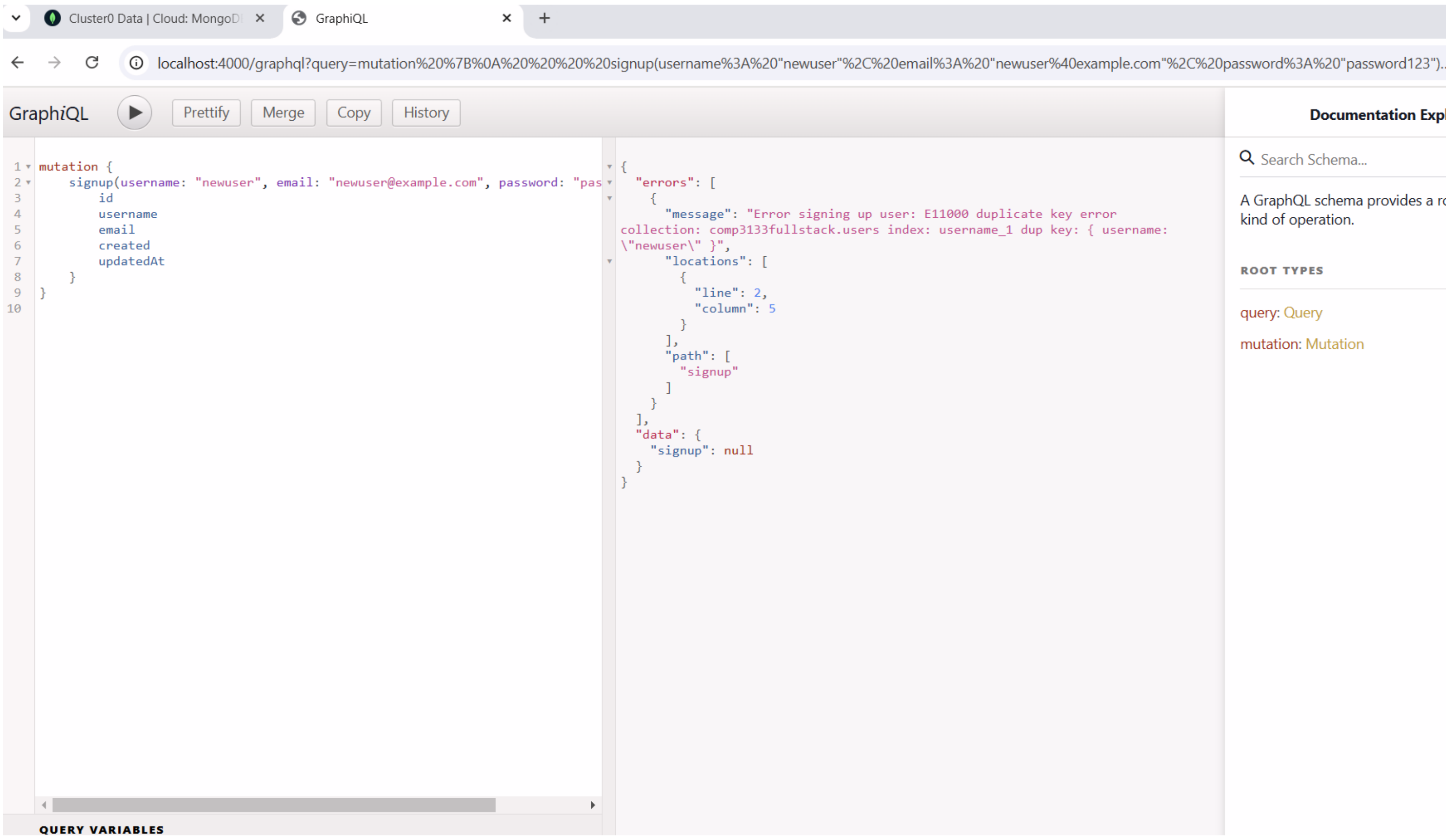
```
{
  "data": {
    "employeesByDesignation": [
      {
        "id": "67b1880b6e2a5d4d5d96f88c",
        "firstname": "Bob",
        "lastname": "Brown",
        "email": "bob.brown@example.com",
        "gender": "Male",
        "city": "Chicago",
        "designation": "DevOps Engineer",
        "department": null,
        "salary": 85000,
        "created": "1739687947562",
        "updatedAt": "1739762669604"
      },
      {
        "id": "67b191d9af7d514b043db89f",
        "firstname": "Bob",
        "lastname": "Brown",
        "email": "bob.brown@example.com",
        "gender": "Male",
        "city": "Chicago",
        "designation": "DevOps Engineer",
        "department": null,
        "salary": 85000,
        "created": "1739690457208",
        "updatedAt": "1739762669605"
      },
      {
        "id": "67b241f89e0d8047b0ef6081",
        "firstname": "Bob",
        "lastname": "Brown",
        "email": "bob.brown@example.com",
        "gender": "Male",
        "city": "Chicago",
        "designation": "DevOps Engineer",
        "department": null,
        "salary": 85000,
        "created": "1739735544690",
        "updatedAt": "1739762669605"
      }
    ]
```

QUERY VARIABLES

Elizabeth Thomas
ID: 101097106

Validation Errors:

User already exists:

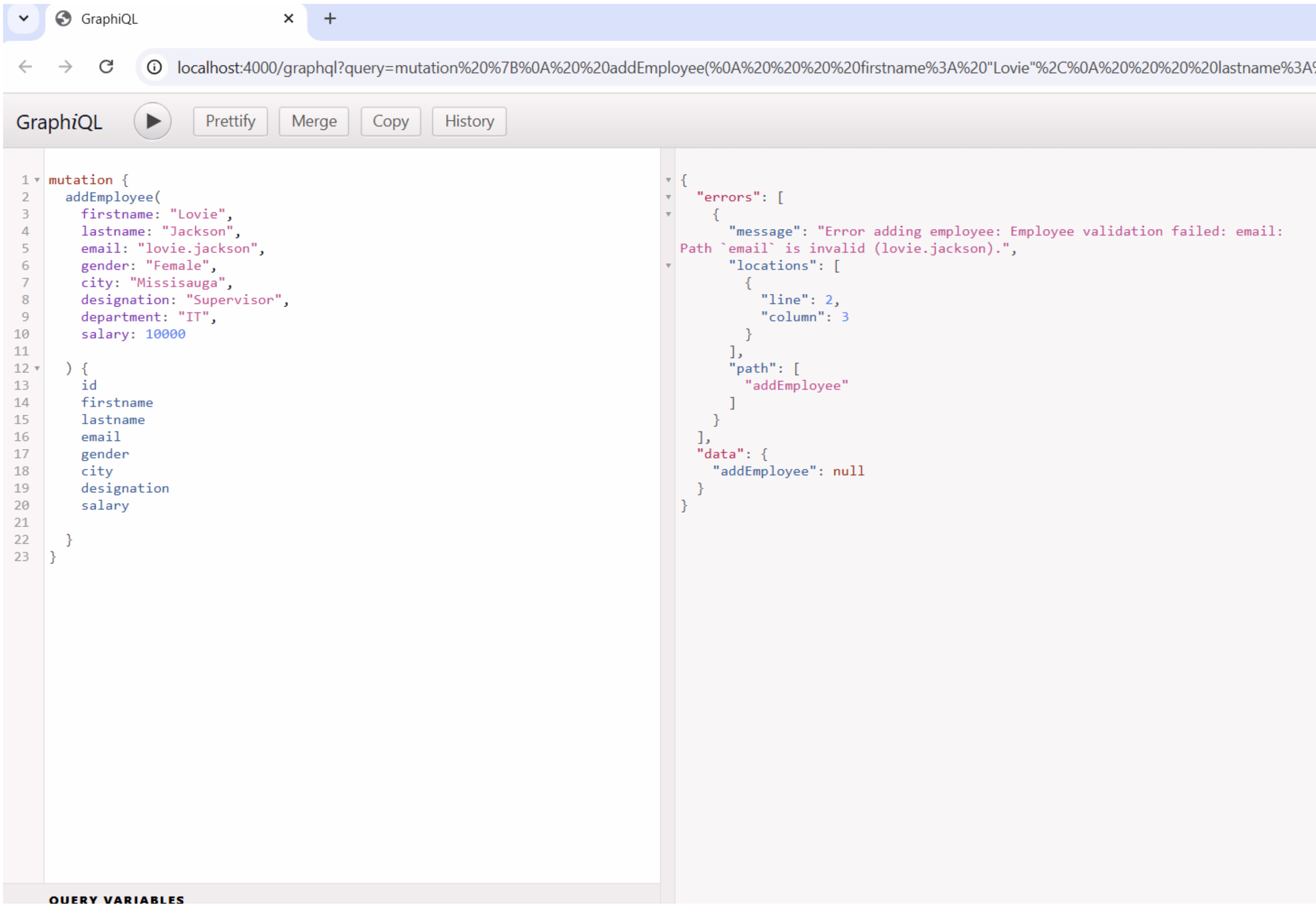

Cluster0 Data | Cloud: MongoD
GraphiQL
localhost:4000/graphql?query=mutation%20%7B%0A%20%20%20%20signup(username%3A%20"newuser"%2C%20email%3A%20"newuser%40example.com"%2C%20password%3A%20"password123")

```
mutation {
    signup(username: "newuser", email: "newuser@example.com", password: "pas
        id
        username
        email
        created
        updatedAt
    }
}
```

```
{
  "errors": [
    {
      "message": "Error signing up user: E11000 duplicate key error
collection: comp3133fullstack.users index: username_1 dup key: { username:
\"newuser\" }",
      "locations": [
        {
          "line": 2,
          "column": 5
        }
      ],
      "path": [
        "signup"
      ]
    }
  ],
  "data": {
    "signup": null
  }
}
```

Elizabeth Thomas
ID: 101097106

Email wrong format

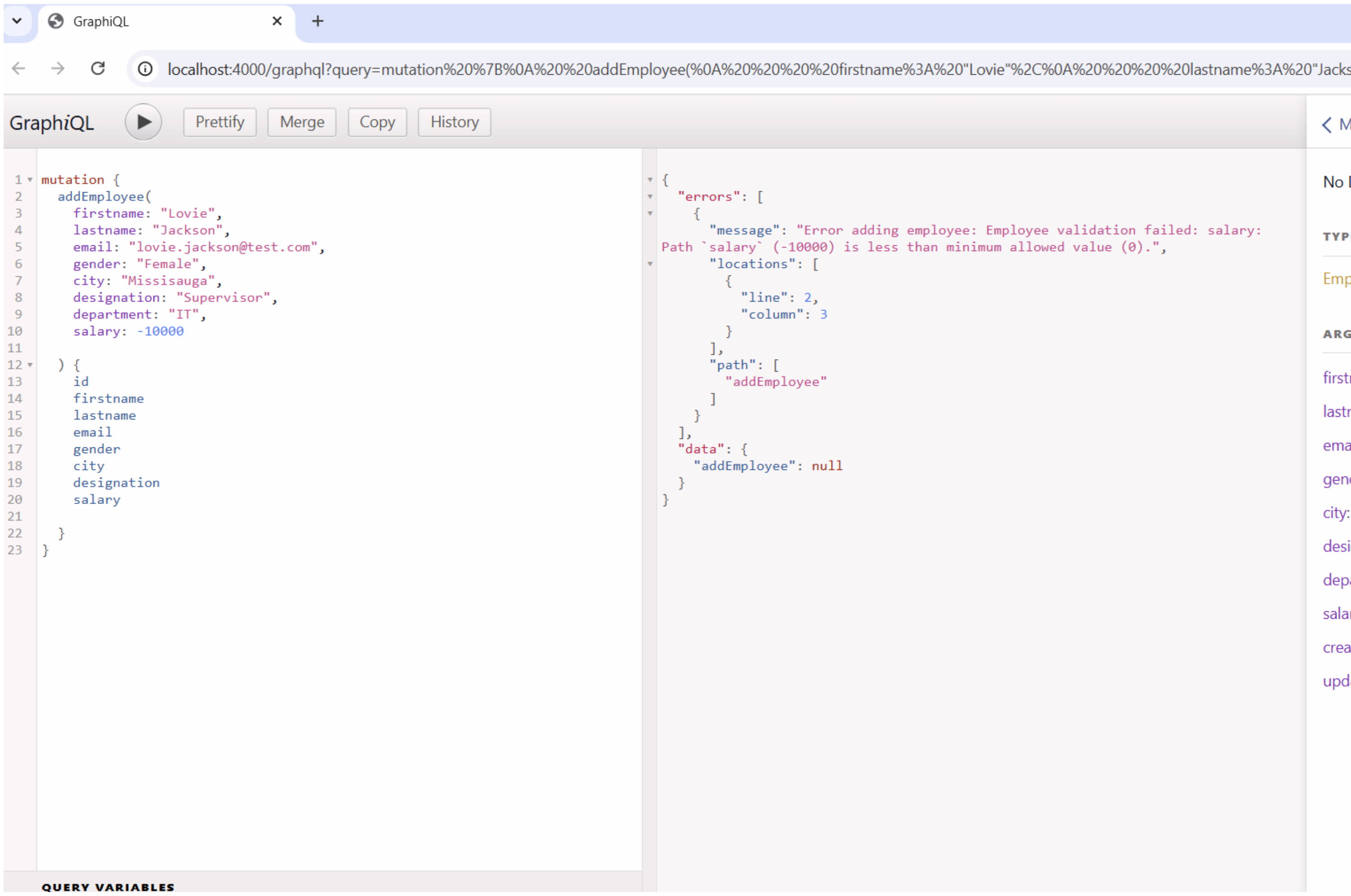

GraphiQL
localhost:4000/graphql?query=mutation%20%7B%0A%20%20addEmployee(%0A%20%20%20%20firstname%3A%20"Lovie"%2C%0A%20%20%20%20lastname%3A
GraphiQL
Prettify
Merge
Copy
History
QUERY VARIABLES

```
mutation {
  addEmployee(
    firstname: "Lovie",
    lastname: "Jackson",
    email: "lovie.jackson",
    gender: "Female",
    city: "Missisauga",
    designation: "Supervisor",
    department: "IT",
    salary: 10000

  ) {
    id
    firstname
    lastname
    email
    gender
    city
    designation
    salary

  }
}
```

```
{
  "errors": [
    {
      "message": "Error adding employee: Employee validation failed: email: Path `email` is invalid (lovie.jackson).",
      "locations": [
        {
          "line": 2,
          "column": 3
        }
      ],
      "path": [
        "addEmployee"
      ]
    }
  ],
  "data": {
    "addEmployee": null
  }
}
```

Elizabeth Thomas
ID: 101097106

Validation: salary less than zero



localhost:4000/graphql?query=mutation%20%7B%0A%20%20addEmployee(%0A%20%20%20%20firstname%3A%20"Lovie"%2C%0A%20%20%20%20lastname%3A%20"Jacks

```
mutation {
  addEmployee(
    firstname: "Lovie",
    lastname: "Jackson",
    email: "lovie.jackson@test.com",
    gender: "Female",
    city: "Missisauga",
    designation: "Supervisor",
    department: "IT",
    salary: -10000

  ) {
    id
    firstname
    lastname
    email
    gender
    city
    designation
    salary

  }
}
```

```
{
  "errors": [
    {
      "message": "Error adding employee: Employee validation failed: salary:
Path `salary` (-10000) is less than minimum allowed value (0).",
      "locations": [
        {
          "line": 2,
          "column": 3
        }
      ],
      "path": [
        "addEmployee"
      ]
    }
  ],
  "data": {
    "addEmployee": null
  }
}
```