**Question : 1. Create an Employee node**

Answer : Create Record via GraphQL

1. CREATE(EMP:Employee{id:100, name: "Johan", salary: 20000,imageUrl:" Johanurl.com" })
2. CREATE(EMP:Employee{id:101, name: "emp1", salary: 30000,imageUrl:"imageUrl1.com" })
3. CREATE(EMP:Employee{id:102, name: "emp2", salary: 40000,imageUrl:"imageUrl2.com" })
4. CREATE(EMP:Employee{id:103, name: "emp3", salary: 40000,imageUrl:"imageUrl3.com" })
5. CREATE(EMP:Employee{id:104, name: "emp3", salary: 40000,imageUrl:"imageUrl4.com" })

And Fetch data from GraphQL via following GraphQL Query

MATCH (emp:Employee)

WHERE emp.name = 'emp3' or emp.id=101

RETURN emp

ALSO I COMPLETED QUESTION ONE VIA JAVA CODE

Below share detail URL and different CRUD operation. option

**Question : 2**. Return all Employee nodes to the client. Any format is fine. For the actual implementation use Java (Spring boot, Dropwizard, Spark java etc) , Javascript (NodeJS + Express etc.) or Python.

Answer :2

Option ==1.

<http://localhost:8080/employee/new>

<http://localhost:8080/>

<http://localhost:8080/employee>

Display Existing record and Add new Record (link to)

A screenshot of a computer

Description automatically generated

Option =2

http://localhost:8080/employee/show/{id}

<http://localhost:8080/employee/show/40>

display Employee detail whose id =40

A screenshot of a computer

Description automatically generated

option ==3

[http://localhost:8080/employee/view/{empname}](http://localhost:8080/employee/view/%7bempname%7d)

<http://localhost:8080/employee/view/BB>

display Employee(s) detail whose name =BB

A screenshot of a computer

Description automatically generated

option=4

[http://localhost:8080/employee/view/{empname}/{empid}](http://localhost:8080/employee/view/%7bempname%7d/%7bempid%7d)

<http://localhost:8080/employee/view/BB/40>

display Employee(s) detail whose name =BB or id=44

A screenshot of a computer

Description automatically generated

option=5

http://localhost:8080/employee/edit/{empid}

<http://localhost:8080/employee/edit/22>

display Employee detail whose id =22 with Edit option.

A screenshot of a computer

Description automatically generated

Option =6

[http://localhost:8080/employee/delete/{emp\_id}](http://localhost:8080/employee/delete/%7bemp_id%7d)

<http://localhost:8080/employee/delete/23>

delete Employee detail whose id =23 and display remaining emp list.

A screenshot of a computer

Description automatically generated

Option =7

http://localhost:8080/employee/view/{emp\_name/{emp\_id}

<http://localhost:8080/employee/view/BB/23>

display Employee List in detail whose empname =BB or emp\_id =23

A screenshot of a computer

Description automatically generated

[**http://localhost:8080/product/**](http://localhost:8080/product/)

**or**

[**http://localhost:8080/product/new**](http://localhost:8080/product/new)

**A screenshot of a computer

Description automatically generated**

[**http://localhost:8080/product/show/39**](http://localhost:8080/product/show/39)

**A screenshot of a computer

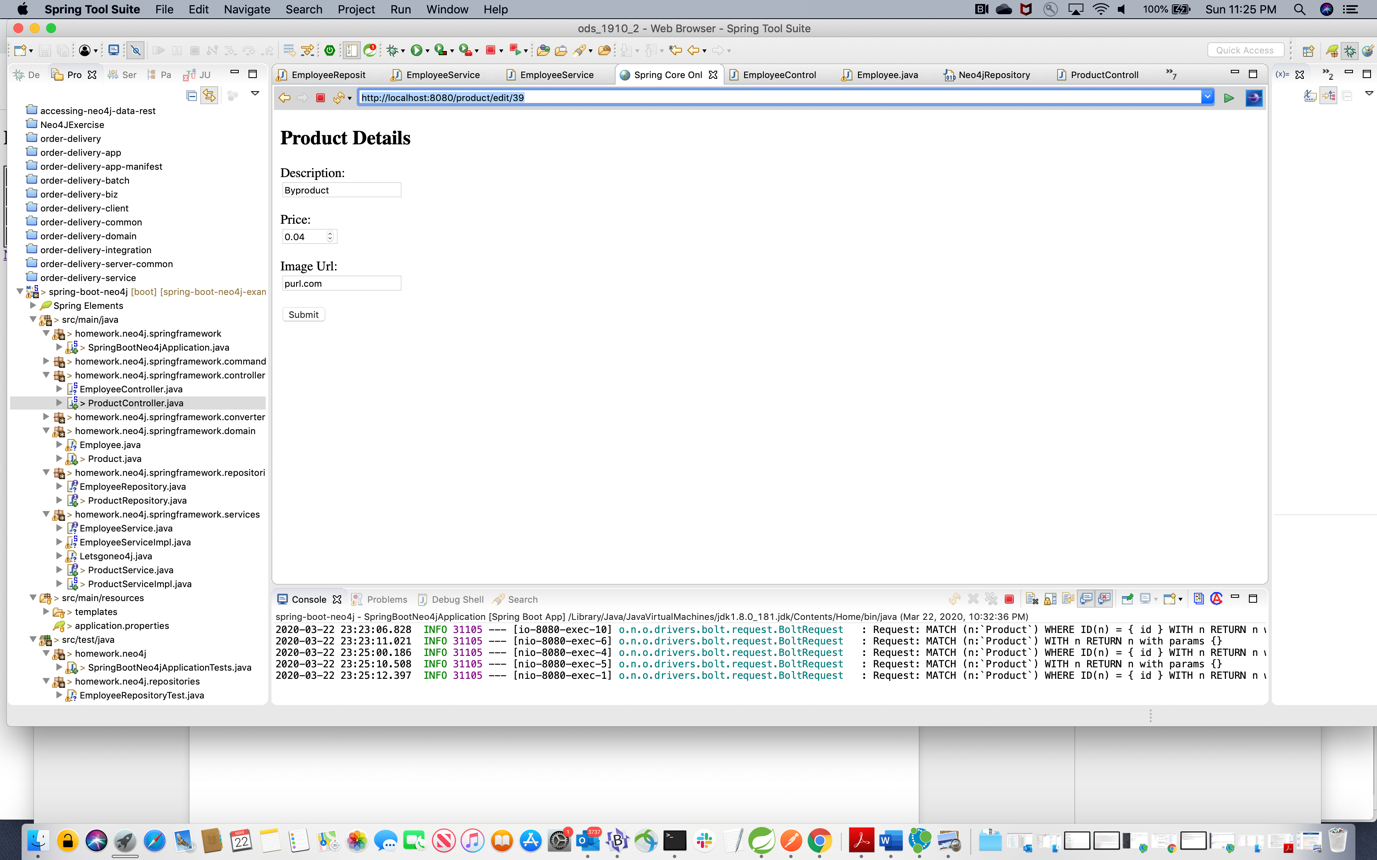
Description automatically generated**

[**http://localhost:8080/product/list**](http://localhost:8080/product/list)

**A screenshot of a computer

Description automatically generated**

[**http://localhost:8080/product/edit/39**](http://localhost:8080/product/edit/39)

****

[**http://localhost:8080/product/delete/22**](http://localhost:8080/product/delete/22)

**A screenshot of a computer

Description automatically generated**

**Bonus task 1**: develop a simple UI that can be used directly from the web browser to drive the API - any stack will do, from plain HTML forms to JQuery to Angular/React/etc.

**Answer: ALL URL for CRUD operation is given in question-2 Answer**