**Spring REST Assignment**

**1)**

**HomeController.java**

package rest.REST.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class HomeController {

    @GetMapping("/hello")

    @ResponseBody

    public String helloWorld() {

        return "Hello World!!";

    }

}

RestApplication.java

package rest.REST;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

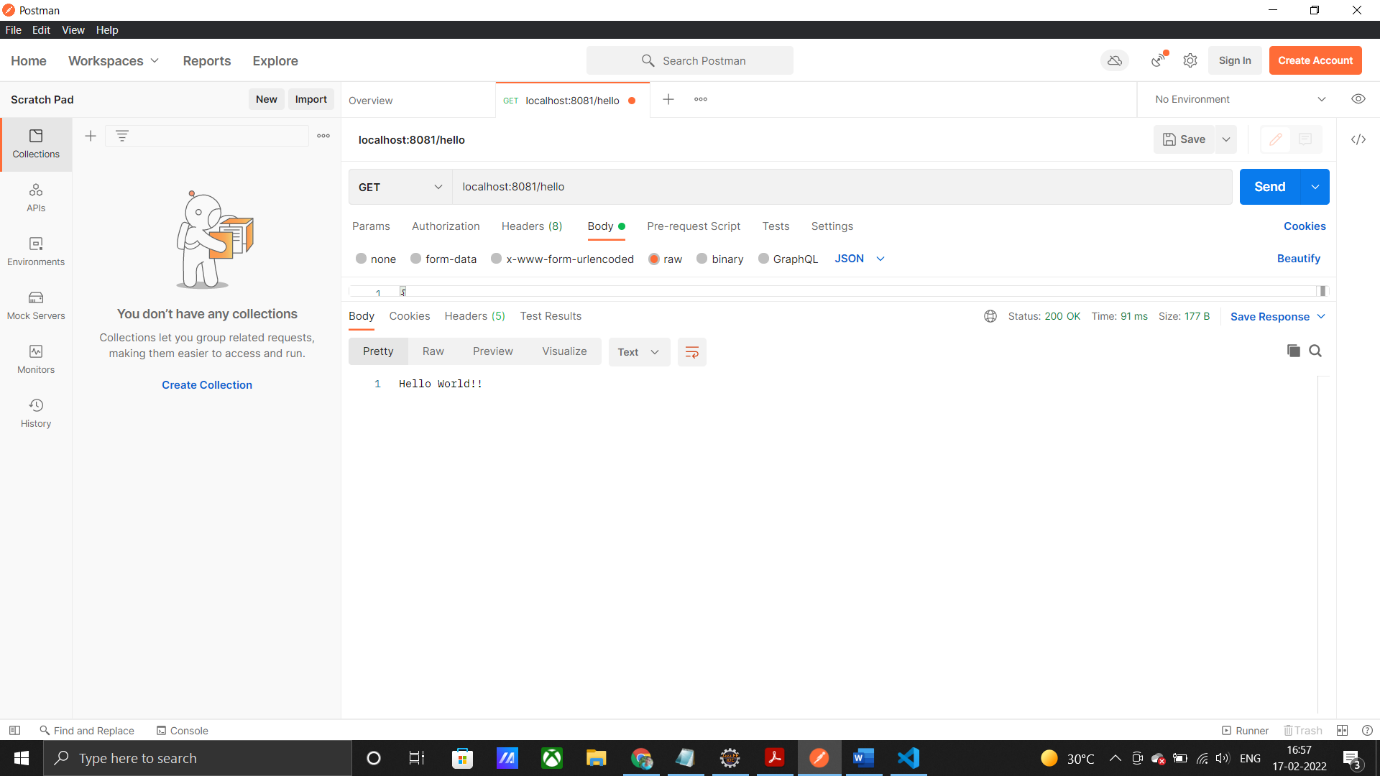
public class RestApplication {

    public static void main(String[] args) {

        SpringApplication.run(RestApplication.class, args);

    }

}



RestApplicationTests.java

package rest.REST;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

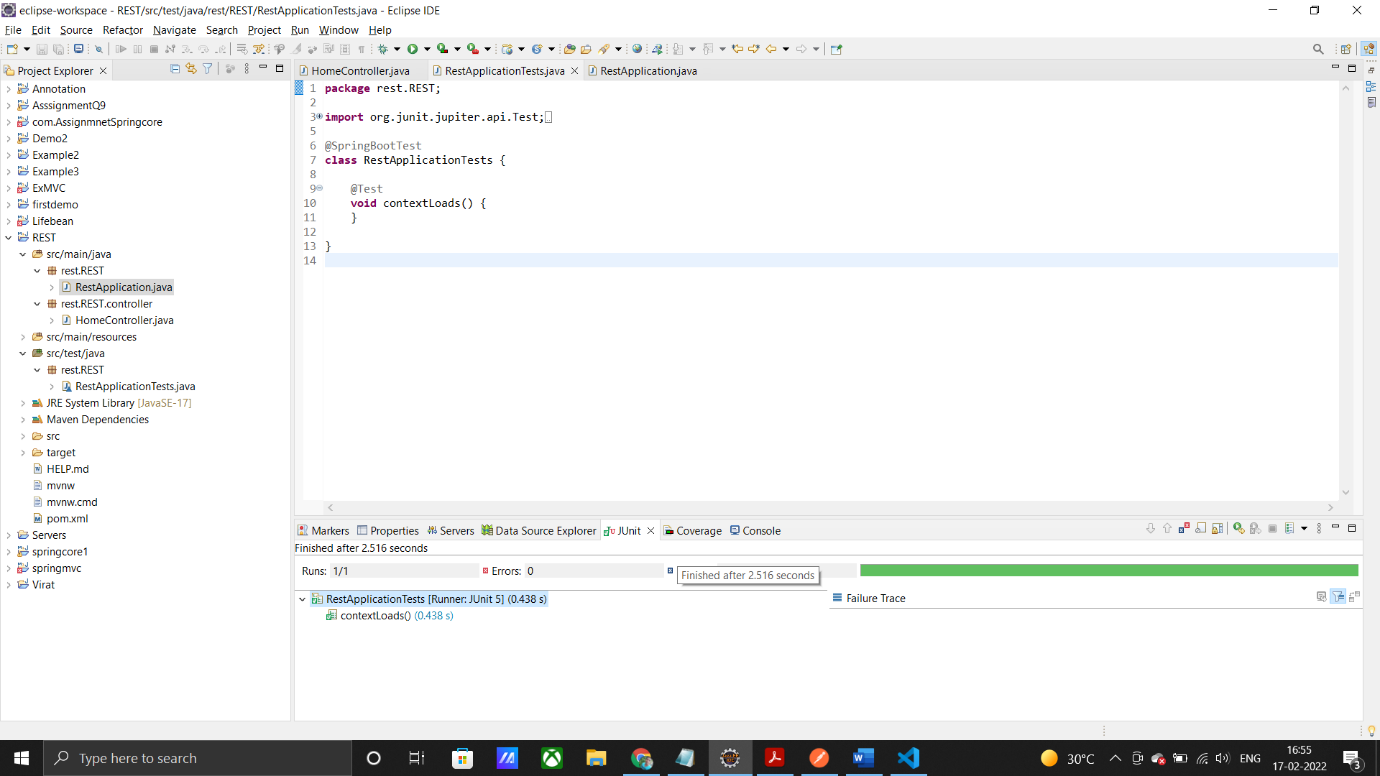
class RestApplicationTests {

    @Test

    void contextLoads() {

    }

}



2)

HomeController.java

package com.example.Assignment2.controller;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.bind.annotation.RestController;

import com.example.Assignment2.entities.User;

@RestController

public class HomeController {

    User user = new User("Himanshu", "himan!123");

    @PostMapping("/login")

    @ResponseBody

    public String login(@RequestBody User user) {

        System.out.println(user);

        if (this.user.getUsername().equals(user.getUsername()) && this.user.getPassword().equals(user.getPassword())) {

            return "Valid User";

        }

        return "Invalid User";

    }

}

User.java

package com.example.Assignment2.entities;

public class User {

    private String username;

    private String password;

    public User() {

        super();

    }

    public User(String username, String password) {

        this.username = username;

        this.password = password;

    }

    public String getUsername() {

        return username;

    }

    public void setUsername(String username) {

        this.username = username;

    }

    public String getPassword() {

        return password;

    }

    public void setPassword(String password) {

        this.password = password;

    }

    @Override

    public String toString() {

        return "User [password=" + password + ", username=" + username + "]";

    }

}

3)

(Example3Application.java)

package example.Example3;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Example3Application {

    public static void main(String[] args) {

        SpringApplication.run(Example3Application.class, args);

    }

}

(controller.java)

package example.Example3.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.bind.annotation.RestController;

import example.Example3.entities.zipcode;

import example.Example3.services.zipcodeService;

@RestController

public class controller {

    @Autowired

    private zipcodeService ZipcodeService;

    @GetMapping("/country-details/{id}")

    @ResponseBody

    public zipcode getByZipcode(@PathVariable("id") int id) {

        System.out.println("Zipcode is: " + id);

        return ZipcodeService.getzipcodeByzipcode(id);

    }

    @GetMapping("/country-details")

    public List<zipcode> getAll() {

        for (zipcode z : ZipcodeService.getAll()) {

            System.out.println(z);

        }

        return ZipcodeService.getAll();

    }

}

(zipcode.java)

package example.Example3.entities;

//Sample Input: 99501

//Sample output: {"state": "AK", City: "ANCHORAGE", "country: "US"}

public class zipcode {

 private int zipcode;

 private String state;

 private String city;

 private String country;

 public zipcode() {

     super();

 }

 public zipcode(int zipcode, String state, String city, String country) {

     this.zipcode = zipcode;

     this.state = state;

     this.city = city;

     this.country = country;

 }

 public int getZipcode() {

     return zipcode;

 }

 public void setZipcode(int zipcode) {

     this.zipcode = zipcode;

 }

 public String getState() {

     return state;

 }

 public void setState(String state) {

     this.state = state;

 }

 public String getCity() {

     return city;

 }

 public void setCity(String city) {

     this.city = city;

 }

 public String getCountry() {

     return country;

 }

 public void setCountry(String country) {

     this.country = country;

 }

 @Override

 public String toString() {

     return "zipcode [city=" + city + ", country=" + country + ", state=" + state + ", zipcode=" + zipcode + "]";

 }

}

(zipcodeService.java)

package example.Example3.services;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import example.Example3.entities.zipcode;

@Component

public class zipcodeService {

    private static List<zipcode> list = new ArrayList<zipcode>();

    // zipcode Zip = new zipcode(99501, "AK", "ANCHORAGE", "US");

    static {

        list.add(new zipcode(99501, "AK", "ANCHORAGE", "US"));

    }

    public zipcode getzipcodeByzipcode(int zipcode) {

        return list.stream().filter(z -> z.getZipcode() == zipcode).findFirst().get();

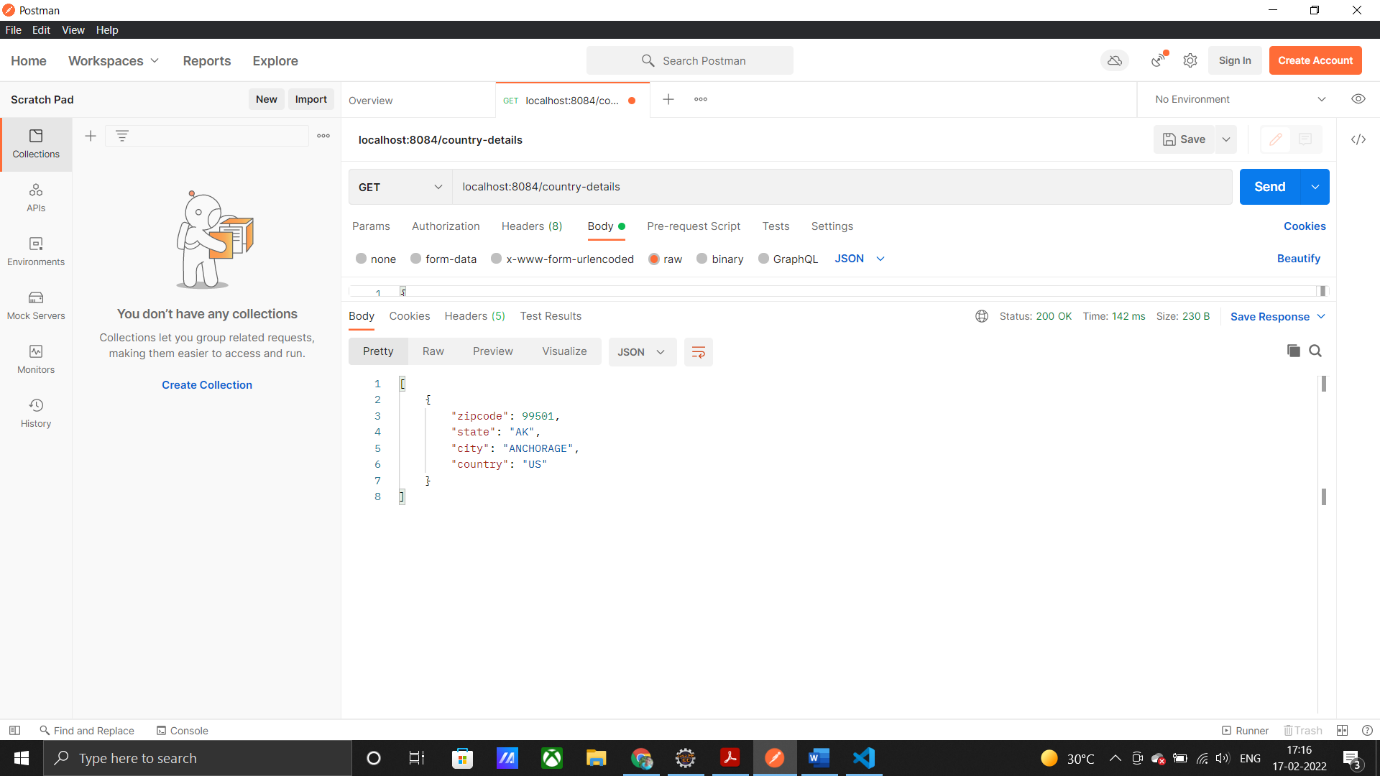
    }

    public List<zipcode> getAll() {

        return list;

    }

}



(Example3ApplicationTests.java)

package example.Example3;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

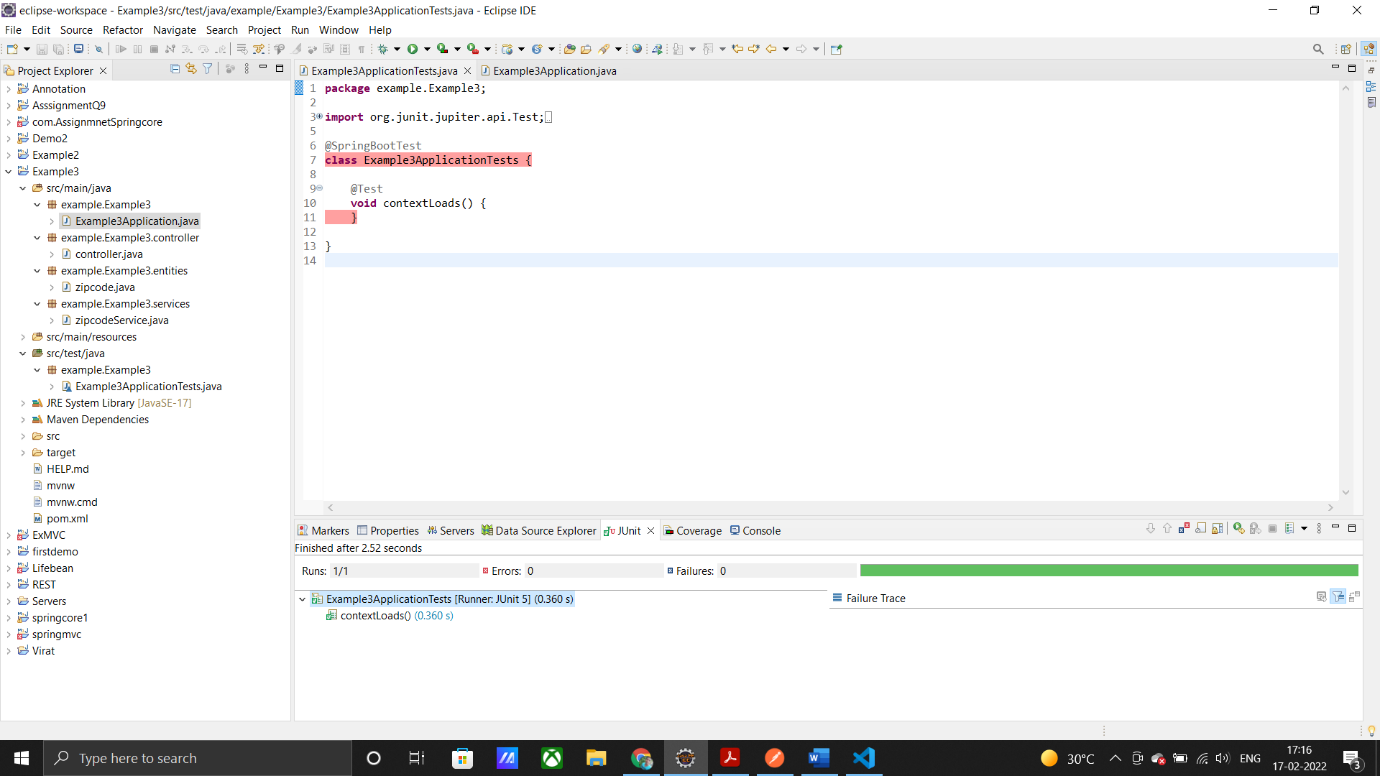
class Example3ApplicationTests {

    @Test

    void contextLoads() {

    }

}



4)

(controller.java)

package Assignment04.controller;

import service.Service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class controller {

    @Autowired

    private Service service;

    @GetMapping(path = "/credit-card/{id}")

    public String creditCard(@PathVariable("id") long number) {

        boolean ans = service.isValid(number);

        if (ans == true) {

            return "Credit card is valid";

        }

        return "Credit card is not valid";

    }

}

(Application4Assignment.java)

package bablu;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Assignment4Application {

    public static void main(String[] args) {

        SpringApplication.run(Assignment4Application.class, args);

    }

}

(Service.java)

package service;

import org.springframework.stereotype.Component;

@Component

public class Service {

    public boolean isValid(long number) {

        return (getSize(number) >= 13 &&

                getSize(number) <= 16) &&

                (prefixMatched(number, 4) ||

                        prefixMatched(number, 5) ||

                        prefixMatched(number, 37) ||

                        prefixMatched(number, 6))

                &&

                ((sumOfDoubleEvenPlace(number) +

                        sumOfOddPlace(number)) % 10 == 0);

    }

    // Get the result from Step 2

    public static int sumOfDoubleEvenPlace(long number) {

        int sum = 0;

        String num = number + "";

        for (int i = getSize(number) - 2; i >= 0; i -= 2)

            sum += getDigit(Integer.parseInt(num.charAt(i) + "") \* 2);

        return sum;

    }

    public static int getDigit(int number) {

        if (number < 9)

            return number;

        return number / 10 + number % 10;

    }

    public static int sumOfOddPlace(long number) {

        int sum = 0;

        String num = number + "";

        for (int i = getSize(number) - 1; i >= 0; i -= 2)

            sum += Integer.parseInt(num.charAt(i) + "");

        return sum;

    }

    public static boolean prefixMatched(long number, int d) {

        return getPrefix(number, getSize(d)) == d;

    }

    public static int getSize(long d) {

        String num = d + "";

        return num.length();

    }

    public static long getPrefix(long number, int k) {

        if (getSize(number) > k) {

            String num = number + "";

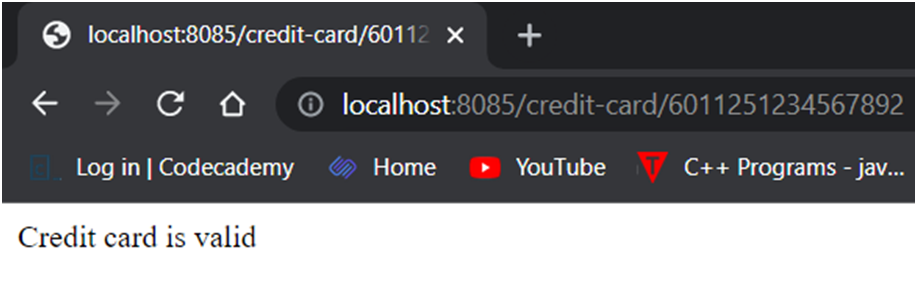
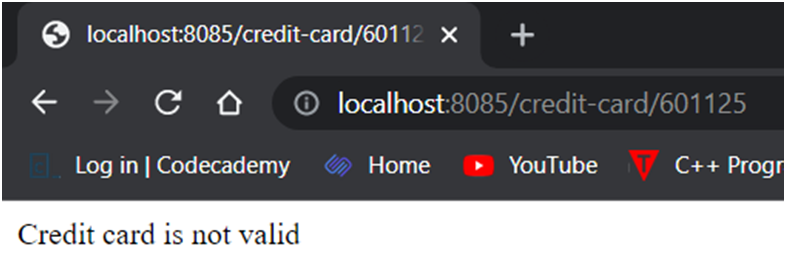
            return Long.parseLong(num.substring(0, k));

        }

        return number;

    }

}



6)

controller.java

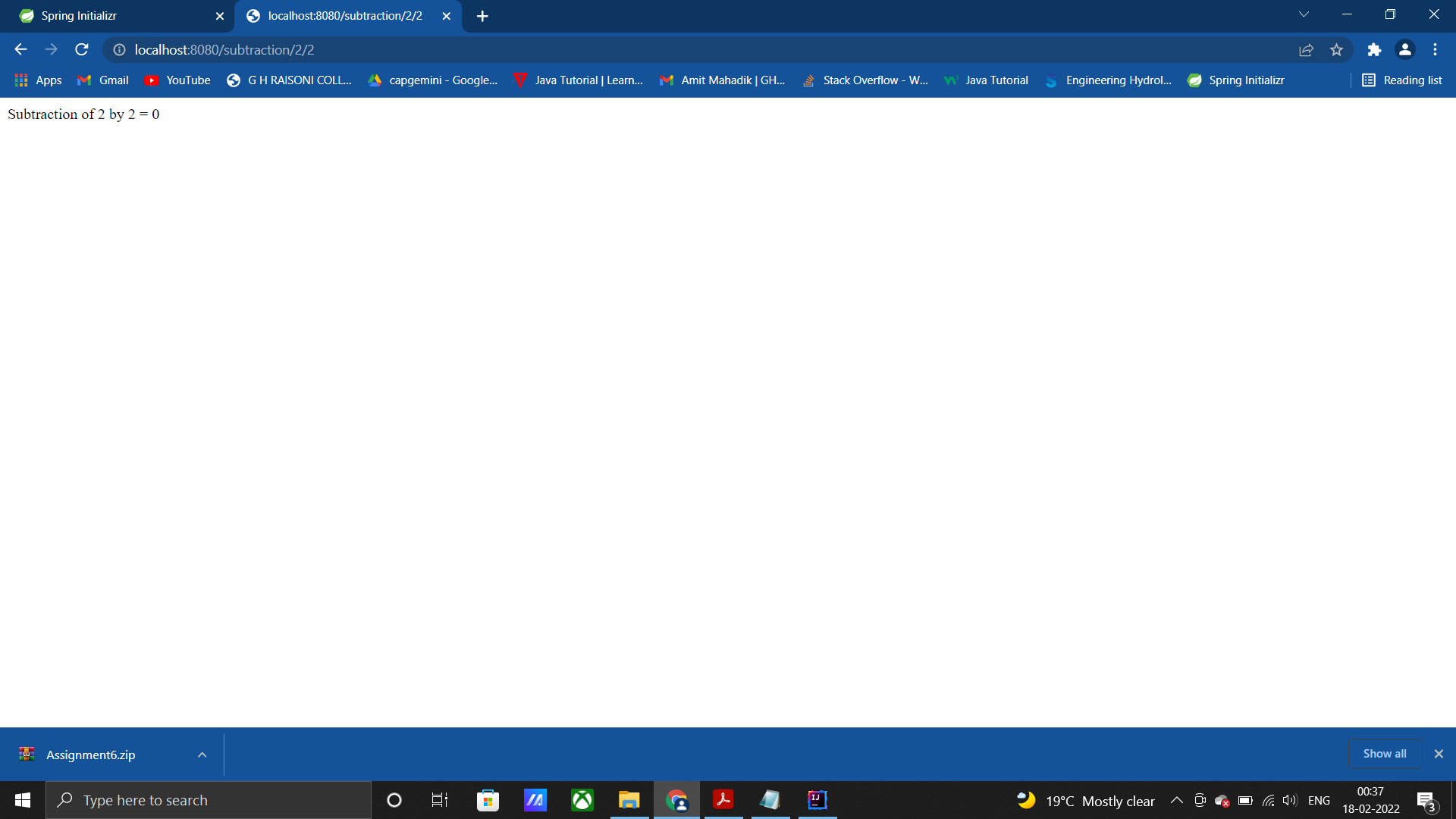
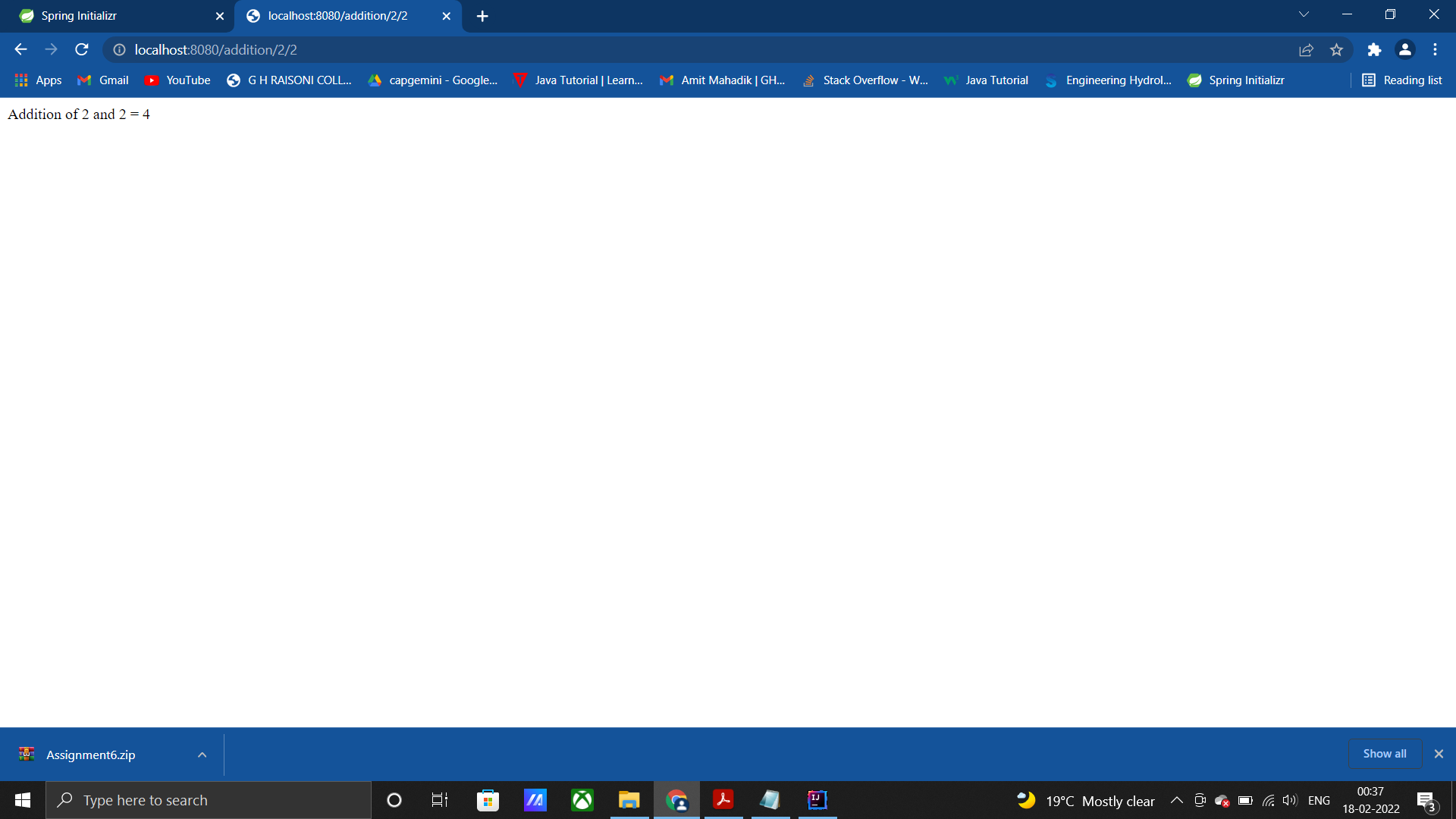
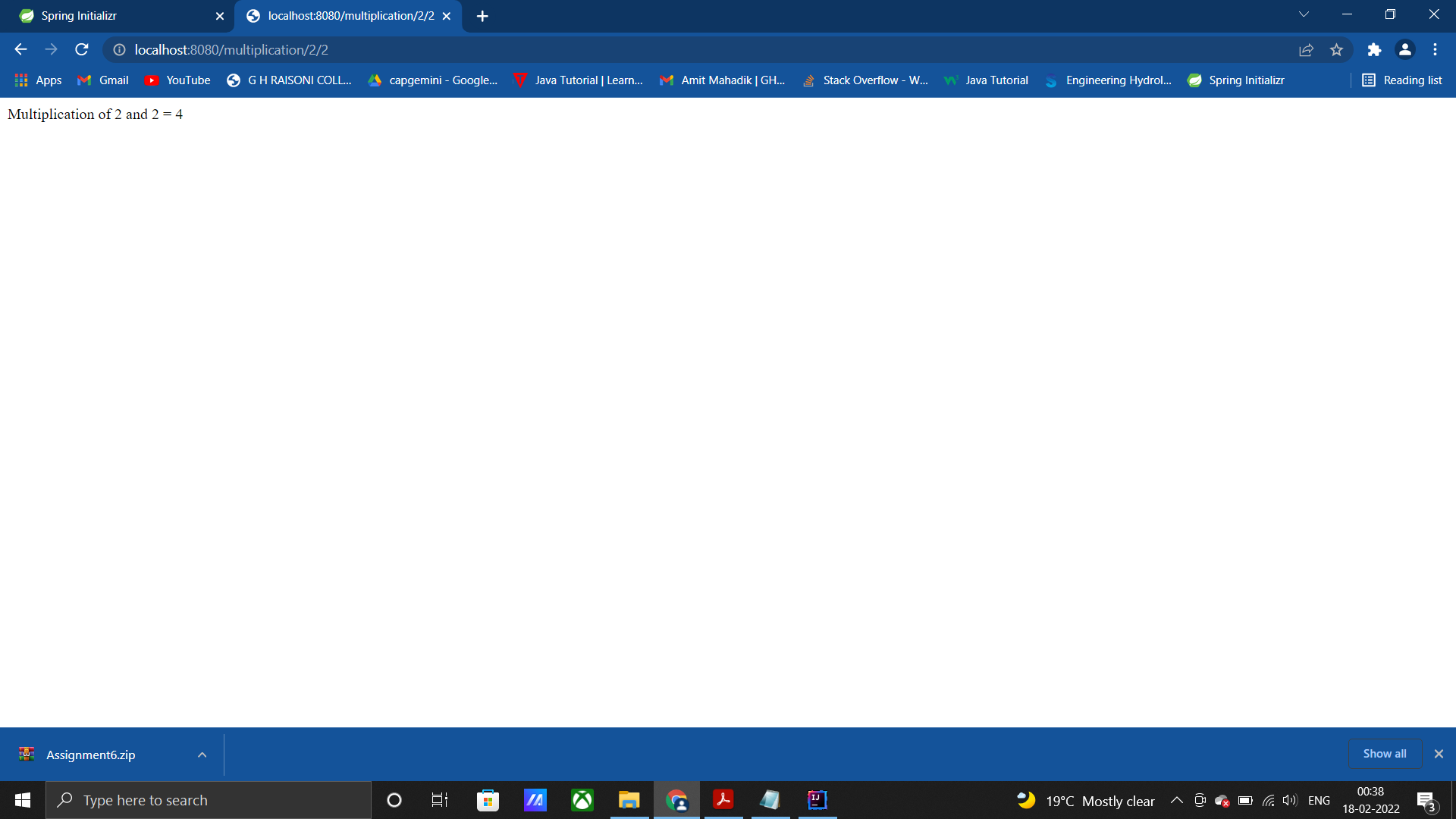
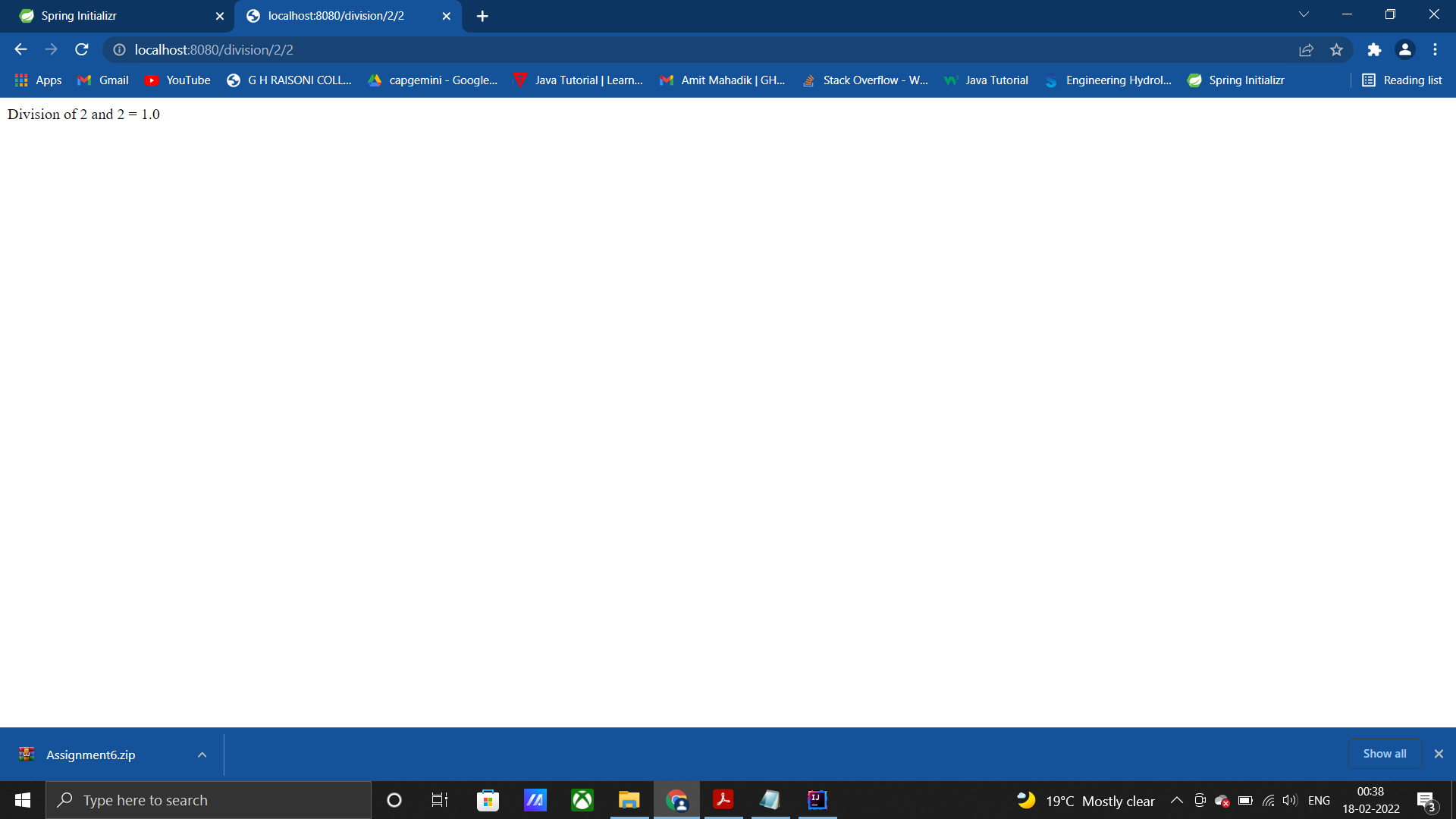
package com.example.Assignment6.controller;  
  
import com.example.Assignment6.service.service;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.ResponseBody;  
  
@Controller  
@ResponseBody  
public class controller {  
   
 @Autowired  
 private service s;  
  
 @GetMapping("/addition/{num1}/{num2}")  
 public String Addition(@PathVariable("num1") int num1, @PathVariable("num2") int num2) {  
 System.*out*.println("Number 1 = " + num1 + "/nNumber 2 = " + num2);  
 return ("Addition of " + num1 + " and " + num2 + " = " + s.Addition(num1, num2));  
 }  
  
 @GetMapping("/subtraction/{num1}/{num2}")  
 public String Subtraction(@PathVariable("num1") int num1, @PathVariable("num2") int num2) {  
 return ("Subtraction of " + num1 + " by " + num2 + " = " + s.subtraction(num1, num2));  
 }  
  
 @GetMapping("/multiplication/{num1}/{num2}")  
 public String Multiplication(@PathVariable("num1") int num1, @PathVariable("num2") int num2) {  
 return ("Multiplication of " + num1 + " and " + num2 + " = " + s.multiplication(num1, num2));  
 }  
  
 @GetMapping("/division/{num1}/{num2}")  
 public String Division(@PathVariable("num1") int num1, @PathVariable("num2") int num2) {  
 return ("Division of " + num1 + " and " + num2 + " = " + s.division(num1, num2));  
 }  
  
 @GetMapping("/square-root/{num1}")  
 public String SquareRoot(@PathVariable("num1") int num1) {  
 return ("Addition of " + num1 + " = " + s.square(num1));  
 }  
  
}

service.java

package com.example.Assignment6.service;  
  
import org.springframework.stereotype.Component;  
  
@Component  
public class service {  
  
 public int Addition(int a, int b) {  
 System.*out*.println(a + b);  
 return a + b;  
 }  
  
 public int subtraction(int a, int b) {  
 System.*out*.println(a - b);  
 return a - b;  
 }  
  
 public int multiplication(int a, int b) {  
 System.*out*.println(a \* b);  
 return a \* b;  
 }  
  
 public double division(int a, int b) {  
 try {  
 System.*out*.println(a / b);  
 return a / b;  
 } catch (Exception e) {  
 e.printStackTrace();  
 return -1;  
 }  
 }  
  
 public double square(int a) {  
 System.*out*.println(Math.*sqrt*(a));  
 return Math.*sqrt*(a);  
 }  
}

Assiggnment6Application.java

package com.example.Assignment6;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class Assignment6Application {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(Assignment6Application.class, args);  
 }  
  
}



5)

EmployeeController.java

package com.example5.com.example.assignment5.controller;

import java.util.List;

import com.example5.com.example.assignment5.entities.Employee;

import com.example5.com.example.assignment5.services.EmployeeService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class EmployeeController {

    @Autowired

    private EmployeeService employeeService;

    @PostMapping("/employee")

    public String AddEmployee(@RequestBody Employee employee) {

        System.out.println(employee);

        employeeService.setEmployee(employee);

        return "Employee Added Successfully";

    }

    @GetMapping("/employee")

    public List<Employee> getAllEmployees() {

        return employeeService.findAllEmployees();

    }

    @GetMapping("/employee/{id}")

    public Employee getEmployee(@PathVariable("id") int id) {

        return employeeService.findEmployeeById(id);

    }

    @DeleteMapping("/employee/{id}")

    public String deleteEmployeeString(@PathVariable("id") int id) {

        employeeService.deleteEmployee(id);

        return "Employee Deleted Successfully";

    }

    @PutMapping("/employee/{id}")

    public String updateEmployee(@RequestBody Employee e, @PathVariable("id") int id) {

        employeeService.updateEmployee(e, id);

        return "Employee Updated Successfully";

    }

}

Employee.java

package com.example5.com.example.assignment5.entities;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "employee")

public class Employee {

    @Id

    private int id;

    private String name;

    private String department;

    private String designation;

    private long salary;

    public Employee() {

        super();

    }

    public Employee(int id, String name, String department, String designation, long salary) {

        this.id = id;

        this.name = name;

        this.department = department;

        this.designation = designation;

        this.salary = salary;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public String getDepartment() {

        return department;

    }

    public void setDepartment(String department) {

        this.department = department;

    }

    public String getDesignation() {

        return designation;

    }

    public void setDesignation(String designation) {

        this.designation = designation;

    }

    public long getSalary() {

        return salary;

    }

    public void setSalary(long salary) {

        this.salary = salary;

    }

    @Override

    public String toString() {

        return "Employee [department=" + department + ", designation=" + designation + ", id=" + id + ", name=" + name

                + ", salary=" + salary + "]";

    }

}

EmployeeService.java

package com.example5.com.example.assignment5.services;

import java.util.List;

import com.example5.com.example.assignment5.dao.EmployeeRepo;

import com.example5.com.example.assignment5.entities.Employee;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Component;

@Component

public class EmployeeService {

    @Autowired

    private EmployeeRepo employeeRepo;

    public void setEmployee(Employee employee) {

        employeeRepo.save(employee);

        System.out.println("Employee added successfully");

    }

    public void setAllEmployee(List<Employee> employees) {

        employeeRepo.saveAll(employees);

        System.out.println("Added all the employees");

    }

    public List<Employee> findAllEmployees() {

        return employeeRepo.findAll();

    }

    public Employee findEmployeeById(int id) {

        return employeeRepo.findById(id);

    }

    public void updateEmployee(Employee e, int id) {

        Employee employee = employeeRepo.findById(id);

        employee.setName(e.getName());

        employee.setDepartment(e.getDepartment());

        employee.setDesignation(e.getDesignation());

        employee.setSalary(e.getSalary());

        employeeRepo.save(employee);

        System.out.println("Employee Updated Successfully");

    }

    public void deleteEmployee(int id) {

        employeeRepo.deleteById(id);

        System.out.println("Employee Deleted Successfully");

    }

}

EmployeeRepo.java

package com.example5.com.example.assignment5.dao;

import com.example5.com.example.assignment5.entities.Employee;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface EmployeeRepo extends JpaRepository<Employee, Integer> {

    public Employee findById(int id);

}

Application.properties

spring.datasource.name=example6

spring.datasource.url=jdbc:mysql://localhost:3306/example6

spring.datasource.username=root

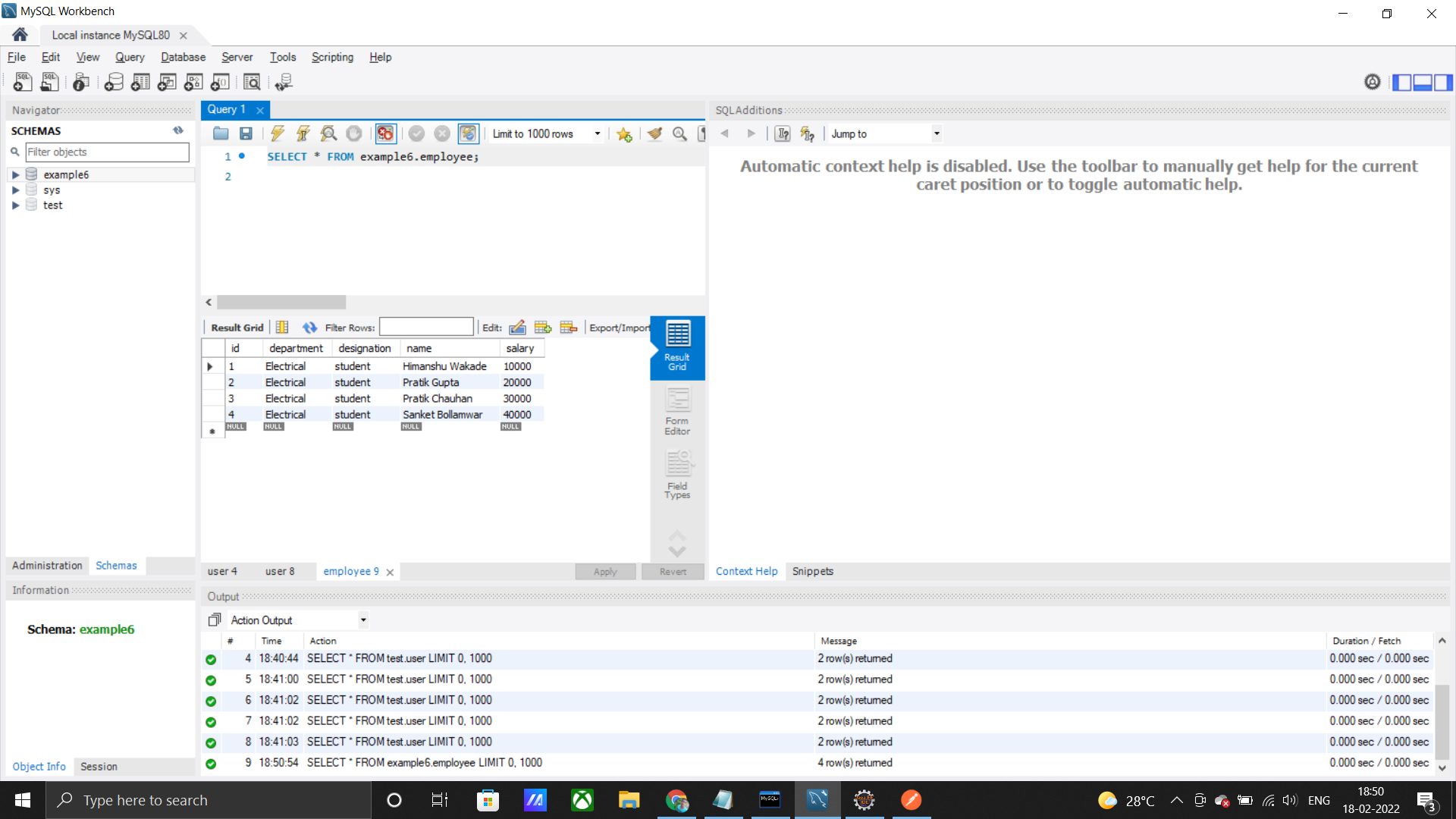
spring.datasource.password=my@sql

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

spring.jpa.hibernate.ddl-auto=update

output:



7)

OrderController.java

package com.springrest.Assignment07.controller;  
  
import java.util.List;  
import java.util.Optional;  
  
import com.springrest.Assignment07.entities.Order;  
import com.springrest.Assignment07.service.OrderService;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class OrderController {  
  
 @Autowired  
 private OrderService orderService;  
  
 @PostMapping("/orders")  
 public ResponseEntity<Order> addOrder(@RequestBody Order order) {  
 try {  
 System.*out*.println(order);  
 orderService.addOrder(order);  
 return ResponseEntity.*ok*(order);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
 @GetMapping("/orders")  
 public ResponseEntity<List<Order>> getAllOrder() {  
 List<Order> list = orderService.getAllOrders();  
  
 if (list.size() <= 0) {  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 return ResponseEntity.*of*(Optional.*of*(list));  
  
 }  
  
 @GetMapping("/orders/{id}")  
 public ResponseEntity<Order> getOrderById(@PathVariable("id") int id) {  
  
 try {  
 Order order = orderService.getOrderById(id);  
 return ResponseEntity.*of*(Optional.*of*(order));  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
  
 }  
  
 @GetMapping("/orders/name")  
 public ResponseEntity<Order> getOrderByName(@RequestParam(value = "name") String name) {  
 try {  
 Order order = orderService.getOrderByName(name);  
 return ResponseEntity.*of*(Optional.*of*(order));  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 }  
  
 @PutMapping("/orders/{id}")  
 public ResponseEntity<Order> updateOrder(@RequestBody Order order, @PathVariable("id") int id) {  
 Order order1 = null;  
 try {  
 order1 = orderService.updateOrder(order);  
 return ResponseEntity.*ok*().body(order1);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
 @DeleteMapping("/orders/{id}")  
 public ResponseEntity<Order> deleteOrder(@PathVariable("id") int id) {  
 try {  
 Order order = orderService.getOrderById(id);  
 orderService.deleteOrder(id);  
 return ResponseEntity.*ok*().body(order);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
}

OrderRepo.java

package com.springrest.Assignment07.dao;  
  
import com.springrest.Assignment07.entities.Order;  
  
import org.springframework.data.mongodb.repository.MongoRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface OrderRepo extends MongoRepository<Order, Integer> {  
  
 public Order findById(int id);  
  
 public Order findByName(String name);  
}

OrderServica.java

package com.springrest.Assignment07.service;  
  
import java.util.List;  
  
import com.springrest.Assignment07.dao.OrderRepo;  
import com.springrest.Assignment07.entities.Order;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Component;  
  
@Component  
public class OrderService {  
  
 @Autowired  
 private OrderRepo orderRepo;  
  
 public void addOrder(Order order) {  
 System.*out*.println(order);  
 orderRepo.save(order);  
 System.*out*.println("Order Added Successfully");  
 }  
  
 public List<Order> getAllOrders() {  
 return orderRepo.findAll();  
 }  
  
 public Order getOrderById(int id) {  
 return orderRepo.findById(id);  
 }  
  
 public Order getOrderByName(String name) {  
 return orderRepo.findByName(name);  
 }  
  
 public Order updateOrder(Order order) {  
 return orderRepo.save(order);  
 }  
  
 public void deleteOrder(int id) {  
 orderRepo.deleteById(id);  
 }  
}

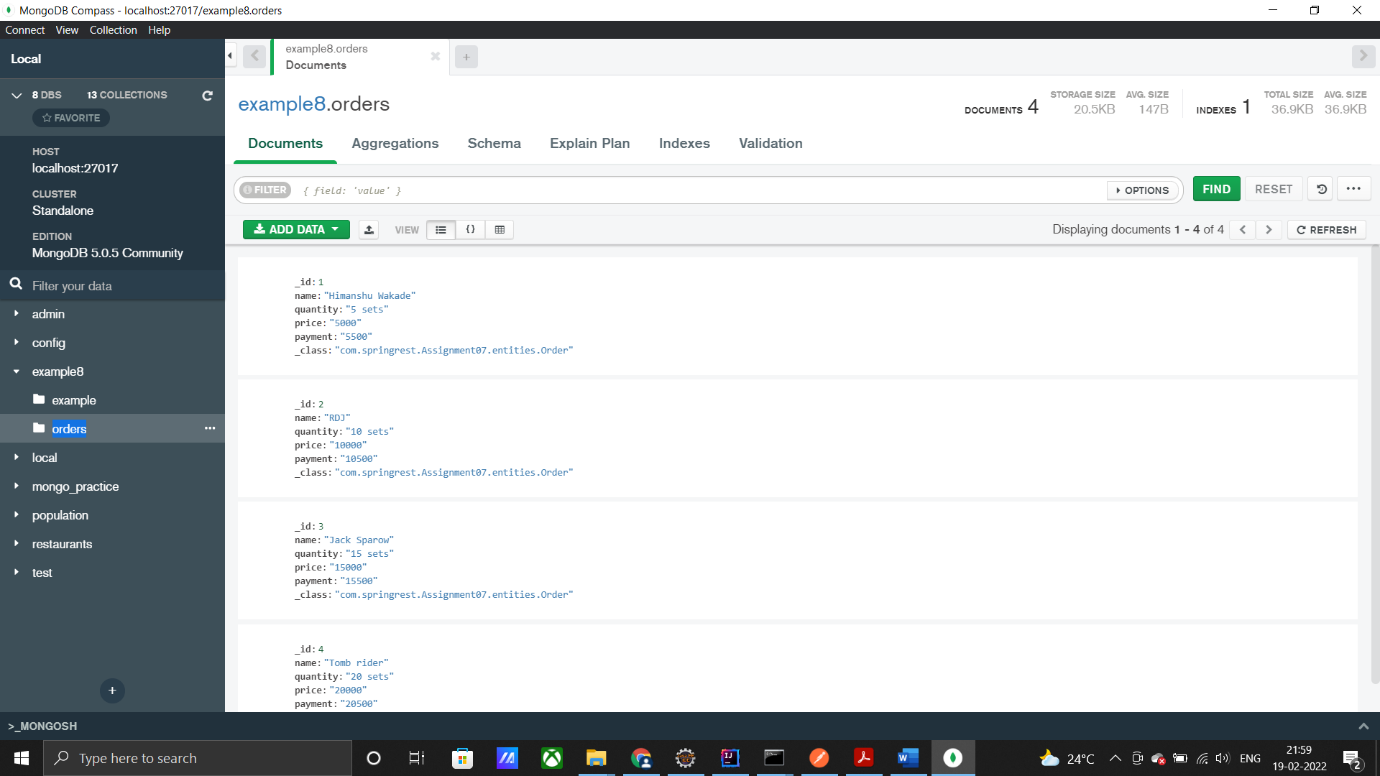
Order.java

package com.springrest.Assignment07.entities;  
  
import org.springframework.data.mongodb.core.mapping.Document;  
  
@Document(collection = "orders")  
public class Order {  
 private int id;  
 private String name;  
 private String quantity;  
 private String price;  
 private String payment;  
  
 public Order() {  
 super();  
 }  
  
 public Order(int id, String name, String quantity, String price, String payment) {  
 this.id = id;  
 this.name = name;  
 this.quantity = quantity;  
 this.price = price;  
 this.payment = payment;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getQuantity() {  
 return quantity;  
 }  
  
 public void setQuantity(String quantity) {  
 this.quantity = quantity;  
 }  
  
 public String getPrice() {  
 return price;  
 }  
  
 public void setPrice(String price) {  
 this.price = price;  
 }  
  
 public String getPayment() {  
 return payment;  
 }  
  
 public void setPayment(String payment) {  
 this.payment = payment;  
 }  
  
 @Override  
 public String toString() {  
 return "Order [id=" + id + ", name=" + name + ", payment=" + payment + ", price=" + price + ", quantity="  
 + quantity + "]";  
 }  
  
}

application.properties

spring.data.mongodb.host= localhost  
spring.data.mongodb.port= 27017  
spring.data.mongodb.database= example8

output:



8 and 9)

8th for product

9th for customer

CustomerController

package com.example8.assignment8.Controller;  
  
import java.util.List;  
import java.util.Optional;  
  
import com.example8.assignment8.Models.Customer;  
import com.example8.assignment8.Services.CustomerService;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestParam;  
  
@Controller  
public class CustomerController {  
  
 @Autowired  
 private CustomerService customerService;  
  
 @GetMapping("/customers")  
 public ResponseEntity<List<Customer>> getAll() {  
 try {  
 List<Customer> list = customerService.getAllCustomers();  
 return ResponseEntity.*of*(Optional.*of*(list));  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 }  
  
 @GetMapping("/customers/{id}")  
 public ResponseEntity<Customer> getCustomerById(@PathVariable("id") int id) {  
 try {  
 Customer customer = customerService.getCustomerById(id);  
 return ResponseEntity.*of*(Optional.*of*(customer));  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 }  
  
 @GetMapping("/customers/name")  
 public ResponseEntity<Customer> getCustomerByName(@RequestParam(value = "name") String name) {  
 try {  
 Customer customer = customerService.getCustomerByName(name);  
 return ResponseEntity.*of*(Optional.*of*(customer));  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 }  
  
 @PostMapping("/customers")  
 public ResponseEntity<Customer> addCustomers(@RequestBody Customer customer) {  
 try {  
 customerService.addCustomer(customer);  
 return ResponseEntity.*ok*().body(customer);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
 @PutMapping("/customers/{id}")  
 public ResponseEntity<Customer> updateCustomer(@RequestBody Customer customer, @PathVariable("id") int id) {  
 try {  
 Customer customer1 = customerService.updateCustomer(customer);  
 return ResponseEntity.*ok*().body(customer1);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
 @DeleteMapping("/customers/{id}")  
 public ResponseEntity<Customer> deleteCustomer(@PathVariable("id") int id) {  
 try {  
 Customer customer = customerService.getCustomerById(id);  
 customerService.deleteCustomer(id);  
 return ResponseEntity.*ok*().body(customer);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
}

OrderController

package com.example8.assignment8.Controller;  
  
import org.springframework.stereotype.Controller;  
  
@Controller  
public class OrderController {  
  
}

ProductController

package com.example8.assignment8.Controller;  
  
import java.util.List;  
import java.util.Optional;  
  
import com.example8.assignment8.Models.Product;  
import com.example8.assignment8.Services.ProductService;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
  
@Controller  
@RequestMapping("/products")  
public class ProductController {  
  
 @Autowired  
 private ProductService productService;  
  
 @GetMapping("/")  
 public ResponseEntity<List<Product>> getAllProducts() {  
 try {  
 List<Product> products = productService.getAllProduct();  
 return ResponseEntity.*of*(Optional.*of*(products));  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 }  
  
 @GetMapping("/{id}")  
 public ResponseEntity<Product> getProductById(@PathVariable("id") int id) {  
 try {  
 Product product = productService.getProductById(id);  
 return ResponseEntity.*of*(Optional.*of*(product));  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 }  
  
 @GetMapping("/names")  
 public ResponseEntity<Product> getProductByName(@RequestParam(value = "name") String name) {  
 try {  
 Product product = productService.getProductByName(name);  
 return ResponseEntity.*of*(Optional.*of*(product));  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 }  
  
 @PostMapping("/")  
 public ResponseEntity<Product> addProduct(@RequestBody Product product) {  
 try {  
 productService.addProduct(product);  
 return ResponseEntity.*ok*().body(product);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
 @PutMapping("/{id}")  
 public ResponseEntity<Product> updateProduct(@RequestBody Product product, @PathVariable("id") int id) {  
 try {  
 productService.updateProduct(product);  
 return ResponseEntity.*ok*().body(product);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
 @DeleteMapping("/{id}")  
 public ResponseEntity<Product> deleteProduct(@PathVariable("id") int id) {  
 try {  
 Product product = productService.getProductById(id);  
 productService.deleteProduct(id);  
 return ResponseEntity.*ok*().body(product);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
}

CustomerRepo

package com.example8.assignment8.Dao;  
  
import com.example8.assignment8.Models.Customer;  
  
import org.springframework.data.mongodb.repository.MongoRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface CustomerRepo extends MongoRepository<Customer, Integer> {  
 public Customer findById(int id);  
  
 public Customer findByFirstName(String name);  
}

OrderRepo

package com.example8.assignment8.Dao;  
  
import com.example8.assignment8.Models.Order;  
  
import org.springframework.data.mongodb.repository.MongoRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface OrderRepo extends MongoRepository<Order, Integer> {  
  
}

ProductRepo

package com.example8.assignment8.Dao;  
  
import com.example8.assignment8.Models.Product;  
  
import org.springframework.data.mongodb.repository.MongoRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface ProductRepo extends MongoRepository<Product, Integer> {  
 public Product findById(int id);  
  
 public Product findByName(String name);  
}

Customer

package com.example8.assignment8.Models;  
  
import org.springframework.data.mongodb.core.mapping.Document;  
  
@Document(collection = "customer")  
public class Customer {  
 private int id;  
 private String firstName;  
 private String lastName;  
 private String street;  
 private String city;  
 private String state;  
 private String zip;  
 private String country;  
  
 public Customer() {  
 }  
  
 public Customer(int id, String firstName, String lastName, String street, String city, String state, String zip,  
 String country) {  
 this.id = id;  
 this.firstName = firstName;  
 this.lastName = lastName;  
 this.street = street;  
 this.city = city;  
 this.state = state;  
 this.zip = zip;  
 this.country = country;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public String getStreet() {  
 return street;  
 }  
  
 public void setStreet(String street) {  
 this.street = street;  
 }  
  
 public String getCity() {  
 return city;  
 }  
  
 public void setCity(String city) {  
 this.city = city;  
 }  
  
 public String getState() {  
 return state;  
 }  
  
 public void setState(String state) {  
 this.state = state;  
 }  
  
 public String getZip() {  
 return zip;  
 }  
  
 public void setZip(String zip) {  
 this.zip = zip;  
 }  
  
 public String getCountry() {  
 return country;  
 }  
  
 public void setCountry(String country) {  
 this.country = country;  
 }  
  
 @Override  
 public String toString() {  
 return "Customer [city=" + city + ", country=" + country + ", firstName=" + firstName + ", id=" + id  
 + ", lastName=" + lastName + ", state=" + state + ", street=" + street + ", zip=" + zip + "]";  
 }  
  
}

Order

package com.example8.assignment8.Models;  
  
import java.sql.Date;  
  
import org.springframework.data.mongodb.core.mapping.Document;  
  
@Document(collection = "order")  
public class Order {  
 private int id;  
 private float total;  
 private Date date;  
 private Customer customer;  
 private Product product;  
  
 public Order() {  
 super();  
 }  
  
 public Order(int id, float total, Date date, Customer customer, Product product) {  
 this.id = id;  
 this.total = total;  
 this.date = date;  
 this.customer = customer;  
 this.product = product;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public float getTotal() {  
 return total;  
 }  
  
 public void setTotal(float total) {  
 this.total = total;  
 }  
  
 public Date getDate() {  
 return date;  
 }  
  
 public void setDate(Date date) {  
 this.date = date;  
 }  
  
 public Customer getCustomer() {  
 return customer;  
 }  
  
 public void setCustomer(Customer customer) {  
 this.customer = customer;  
 }  
  
 public Product getProduct() {  
 return product;  
 }  
  
 public void setProduct(Product product) {  
 this.product = product;  
 }  
  
 @Override  
 public String toString() {  
 return "Order [customer=" + customer + ", date=" + date + ", id=" + id + ", product=" + product + ", total="  
 + total + "]";  
 }  
  
}

Product

package com.example8.assignment8.Models;  
  
import org.springframework.data.mongodb.core.mapping.Document;  
  
@Document(collection = "product")  
public class Product {  
 private int id;  
 private String name;  
 private float price;  
  
 public Product() {  
 }  
  
 public Product(int id, String name, float price) {  
 this.id = id;  
 this.name = name;  
 this.price = price;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public float getPrice() {  
 return price;  
 }  
  
 public void setPrice(float price) {  
 this.price = price;  
 }  
  
 @Override  
 public String toString() {  
 return "Product [id=" + id + ", name=" + name + ", price=" + price + "]";  
 }  
  
}

CustomerService

package com.example8.assignment8.Services;  
  
import java.util.List;  
  
import com.example8.assignment8.Dao.CustomerRepo;  
import com.example8.assignment8.Models.Customer;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Component;  
  
@Component  
public class CustomerService {  
  
 @Autowired  
 private CustomerRepo customerRepo;  
  
 public List<Customer> getAllCustomers() {  
 List<Customer> list = customerRepo.findAll();  
 return list;  
 }  
  
 public Customer getCustomerById(int id) {  
 Customer customer = customerRepo.findById(id);  
 return customer;  
 }  
  
 public Customer getCustomerByName(String name) {  
 return customerRepo.findByFirstName(name);  
 }  
  
 public void addCustomer(Customer customer) {  
 customerRepo.save(customer);  
 }  
  
 public Customer updateCustomer(Customer customer) {  
 return customerRepo.save(customer);  
 }  
  
 public void deleteCustomer(int id) {  
 customerRepo.deleteById(id);  
 }  
}

OrderService

package com.example8.assignment8.Services;  
  
import org.springframework.stereotype.Component;  
  
@Component  
public class OrderService {  
  
}

ProductService

package com.example8.assignment8.Services;  
  
import java.util.List;  
  
import com.example8.assignment8.Dao.ProductRepo;  
import com.example8.assignment8.Models.Product;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Component;  
  
@Component  
public class ProductService {  
  
 @Autowired  
 private ProductRepo productRepo;  
  
 // Adding Products  
 public void addProduct(Product product) {  
 productRepo.save(product);  
 }  
  
 // Retrieving All the products  
 public List<Product> getAllProduct() {  
 return productRepo.findAll();  
 }  
  
 // Getting all products by id  
 public Product getProductById(int id) {  
 return productRepo.findById(id);  
 }  
  
 // Getting all products by name  
 public Product getProductByName(String name) {  
 return productRepo.findByName(name);  
 }  
  
 // Updating product by id  
 public Product updateProduct(Product product) {  
 productRepo.save(product);  
 return product;  
 }  
  
 // Deleting product by id  
 public void deleteProduct(int id) {  
 productRepo.deleteById(id);  
 }  
  
}

application.properties

spring.data.mongodb.host=localhost  
spring.data.mongodb.port=27017  
spring.data.mongodb.database=Products