Task 1 : Create a Spring boot project configure with Github and database write junit test case for a student details name , age ,salary and create a jar and deploy it.

Student application:

package com.upendra.student;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import io.swagger.v3.oas.annotations.OpenAPIDefinition;

import io.swagger.v3.oas.annotations.info.Info;

@OpenAPIDefinition(info = @Info(title = "student"))

@SpringBootApplication

public class StudentApplication {

```
	public static void main(String[] args) {

		SpringApplication.run(StudentApplication.class, args);

	}

}
```

Controller:

package com.upendra.student.controller;

import java.security.PublicKey;

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;

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

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

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

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


import com.upendra.student.model.Student;

import com.upendra.student.service.StudentService;


@RestController
@RequestMapping("/student")
public class StudentCont {


        @Autowired
        StudentService studentService;


        @GetMapping("/getByName/{name}")
        public List<Student> getBySytudent(@PathVariable("name") String name) {
                return studentService.findByName(name);


        }
        @GetMapping("/getByAge/{age}")
        public List<Student> getByAge(@PathVariable("age") int age) {
                return studentService.findByAge(age);


        }


        @GetMapping("/getBySalary/{salary}")
        public List<Student> getBySalary(@PathVariable("salary") long salary) {
                return studentService.findBySalary(salary);
```

```java
        }

        @PostMapping("/post")
        public Student post(@RequestBody Student student)
        {
                return studentService.post(student);
        }

}
```

Service:

```java
package com.upendra.student.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.bind.annotation.RequestBody;

import com.upendra.student.model.Student;
import com.upendra.student.repo.StudentRepo;

@Service
public class StudentService {

        @Autowired
        StudentRepo repo;

        public List<Student> findByName(String name) {
```

```java
                return repo.findByName(name);

        }


        public List<Student> findByAge(int age) {

                return repo.findByAge(age);

        }

        public List<Student> findBySalary(long salary) {

                return repo.findBySalary(salary);

        }

        public Student post(Student student)

        {

                return repo.save(student);

        }

}
```

Repository

```java
package com.upendra.student.repo;


import java.util.List;


import org.springframework.data.mongodb.repository.MongoRepository;

import org.springframework.stereotype.Repository;


import com.upendra.student.model.Student;


@Repository

public interface StudentRepo extends MongoRepository<Student, String>{


        List<Student> findByName(String name);
```

```java
        List<Student> findBySalary(long salary);

        List<Student> findByAge(int age);
}
```

Test Classes:

FOR CONTROLER
```java
package com.upendra.student.controller;


import static org.junit.jupiter.api.Assertions.*;
import static org.mockito.Mockito.when;


import java.util.ArrayList;
import java.util.List;


import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;
import org.springframework.boot.test.mock.mockito.MockBean;
import org.springframework.test.web.servlet.MockMvc;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;


import com.upendra.student.model.Student;
import com.upendra.student.service.StudentService;


@WebMvcTest(value = StudentCont.class)
class StudentContTest {


        @Autowired
```

```java
MockMvc mockMvc;

@MockBean

StudentService studentService;



@Test

void testGetByName() {

        List<Student> list = new ArrayList<>();


        Student student = new Student("Upendra", 44, 40000000);

        list.add(student);

        when(studentService.findByName("upendra")).thenReturn(list);

}


@Test

void testGetBySalary() {

        List<Student> list = new ArrayList<>();


        Student student = new Student("Upendra", 44, 40000000);

        list.add(student);

        when(studentService.findBySalary(246465)).thenReturn(list);

}

@Test

void testGetByAge() {

        List<Student> list = new ArrayList<>();


        Student student = new Student("Upendra", 44, 40000000);

        list.add(student);

        when(studentService.findByAge(0)).thenReturn(list);
```
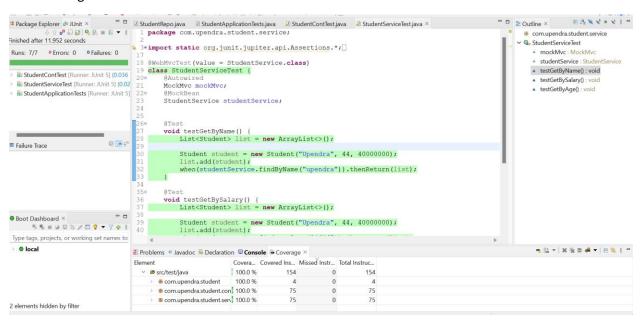
```
            }



}



FOR SERVICE:

package com.upendra.student.service;


import static org.junit.jupiter.api.Assertions.*;

import static org.mockito.Mockito.when;


import java.util.ArrayList;

import java.util.List;


import org.junit.jupiter.api.Test;

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

import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;

import org.springframework.boot.test.mock.mockito.MockBean;

import org.springframework.test.web.servlet.MockMvc;

import org.springframework.web.servlet.config.annotation.EnableWebMvc;


import com.upendra.student.model.Student;


@WebMvcTest(value = StudentService.class)

class StudentServiceTest {

        @Autowired

        MockMvc mockMvc;

        @MockBean
```

```java
StudentService studentService;


@Test
void testGetByName() {

        List<Student> list = new ArrayList<>();


        Student student = new Student("Upendra", 44, 40000000);

        list.add(student);

        when(studentService.findByName("upendra")).thenReturn(list);

}


@Test
void testGetBySalary() {

        List<Student> list = new ArrayList<>();


        Student student = new Student("Upendra", 44, 40000000);

        list.add(student);

        when(studentService.findBySalary(246465)).thenReturn(list);

}
@Test
void testGetByAge() {

        List<Student> list = new ArrayList<>();


        Student student = new Student("Upendra", 44, 40000000);

        list.add(student);

        when(studentService.findByAge(0)).thenReturn(list);

}
```

}

Test coverage Screen shot :

Github Screen shot:

Curl

```
curl -X 'GET' \
  'http://localhost:8081/student/getByAge/22' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8081/student/getByAge/22
```

Server response

Code	Details
200	**Response body**

```
[
  {
    "name": "upendra",
    "age": 22,
    "salary": 28000
  }
]
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Wed,22 Nov 2023 10:02:02 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

Code	Description	Links