Implement a Restful spring boot application for student.

1. create a new Spring Boot project. Include the following dependencies:

   File---->new ----→spring boot starter project

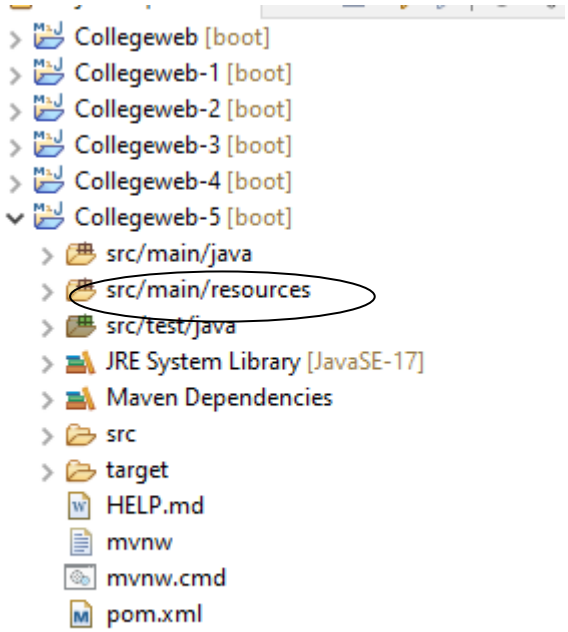
   Add dependencies as

   Spring Web

   Spring Data JPA

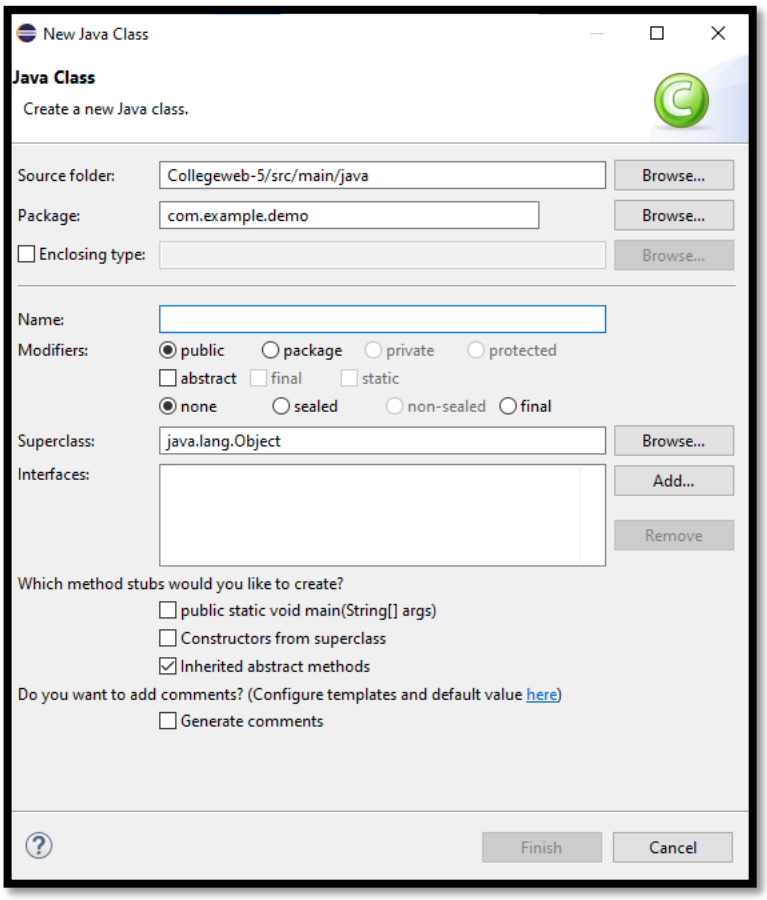
   H2 Database

2) In project explorer select



Select src/main/java as shown in above figure

Right click and select new , add class



Enter name of class as “Student”

Copy and paste following code in Student.java

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Student {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String firstName;
    private String lastName;
    private int age;

    // Getters and setters
}
```

Now select source menu of eclipse and select generate getters and setters ……..

Add getter and setters for all fields

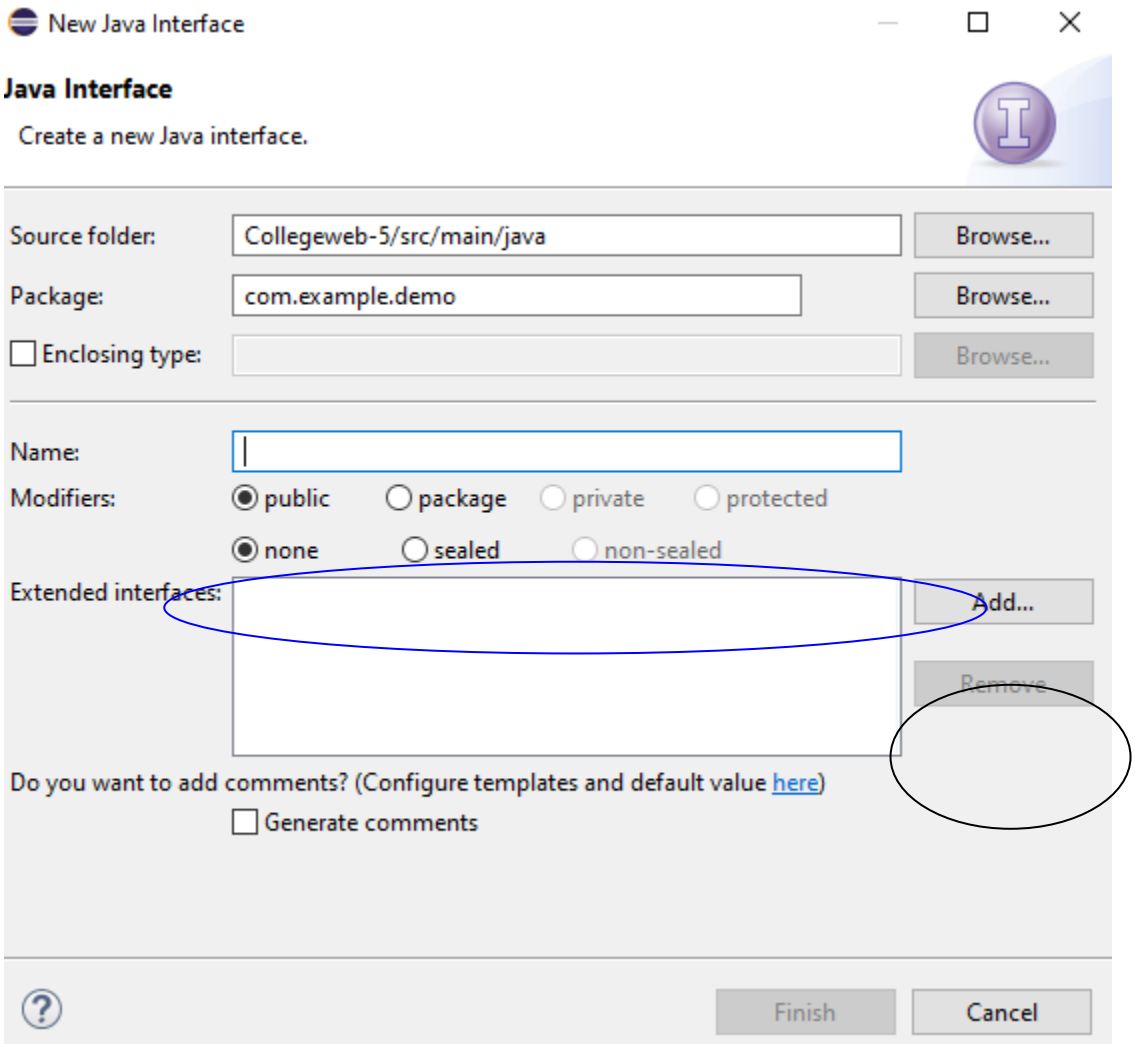
Again select source menu of eclipse and select generate tostring function …..

3) create student repository interface

Select src/main/java

Right click and select new

Select interface



Type Name as StudentRepository.

Click on add button and select interface as JpaRepository

Write following code in this java file

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

public interface StudentRepository extends JpaRepository<Student, Long> {

}
```

4) Create a Student Service:

Create a service class to handle business logic.

Add Class with name StudentService and add following code

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

import org.springframework.stereotype.Service;
```

```
import java.util.List;
import java.util.Optional;

@Service
public class StudentService {

    @Autowired
    private StudentRepository studentRepository;

    public List<Student> getAllStudents() {
        return studentRepository.findAll();
    }

    public Optional<Student> getStudentById(Long id) {
        return studentRepository.findById(id);
    }

    public Student saveStudent(Student student) {
        return studentRepository.save(student);
    }
```

```
    public void deleteStudent(Long id) {
        studentRepository.deleteById(id);
    }
}
```

5) Create a Student Controller:

Create a REST controller to handle HTTP requests.

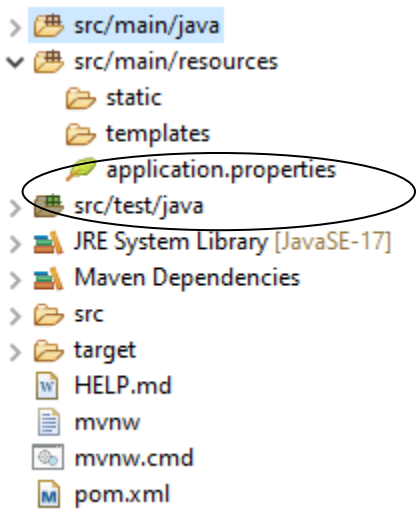
Add class with name StudentController

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Optional;

@RestController
@RequestMapping("/api/students")
public class StudentController {
```

```
    @Autowired
    private StudentService studentService;

    @GetMapping
    public List<Student> getAllStudents() {
        return studentService.getAllStudents();
    }

    @GetMapping("/{id}")
    public Optional<Student> getStudentById(@PathVariable Long
id) {
        return studentService.getStudentById(id);
    }

    @PostMapping
    public Student saveStudent(@RequestBody Student student) {
        return studentService.saveStudent(student);
    }

    @DeleteMapping("/{id}")
    public void deleteStudent(@PathVariable Long id) {
```

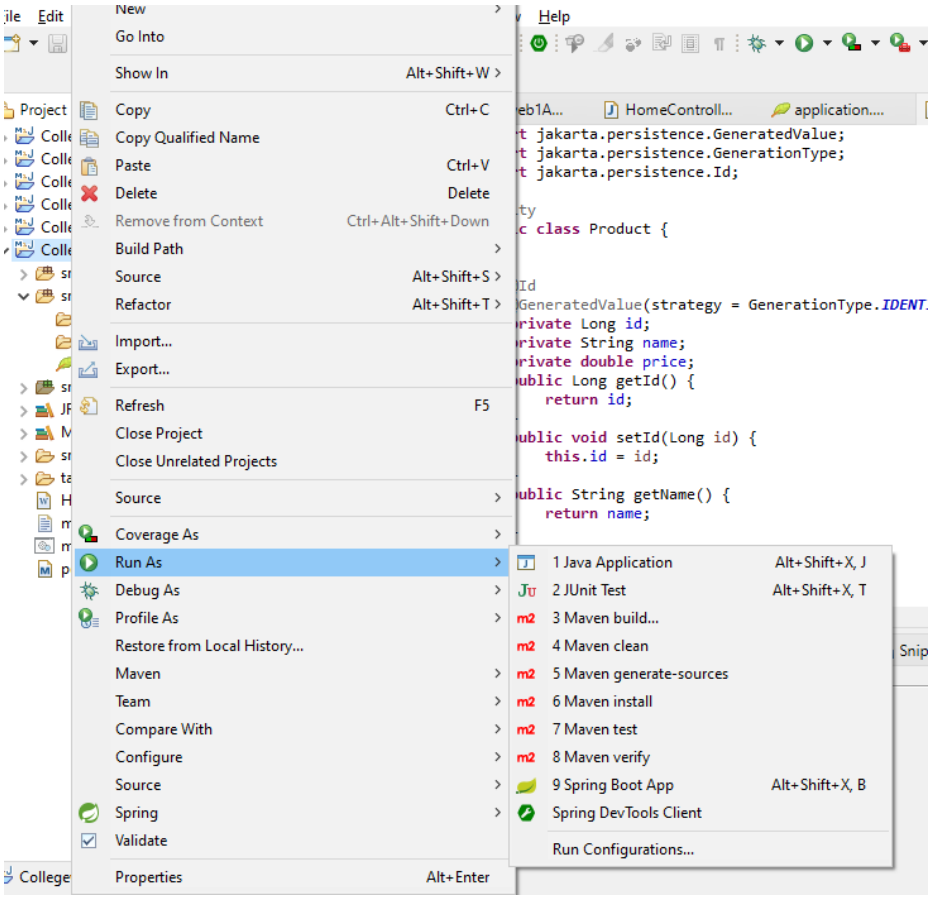
```
        studentService.deleteStudent(id);
    }
}
```

6) In Application.properties write the code

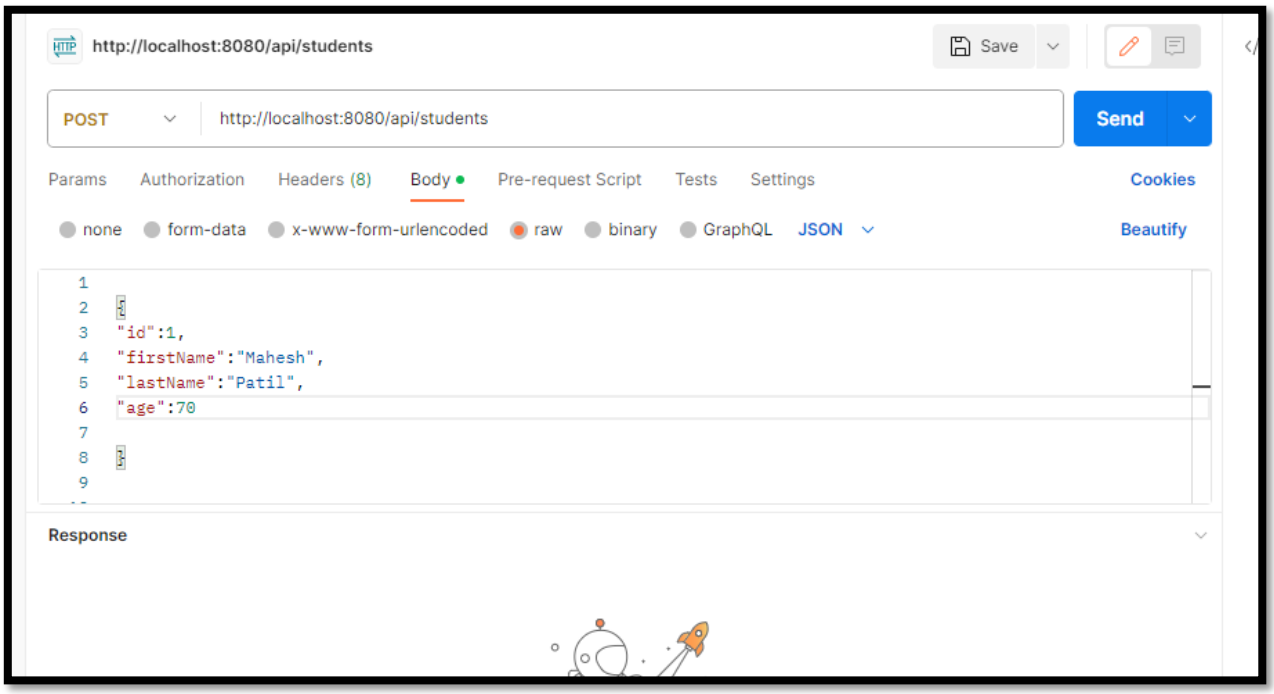



```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password
```

7) Run the application using run as spring boot application



8) now run postman application and set properties according to following figure



Select http type as “post”

Enter url as localhost:8080/api/students

Select body and enter information as

```
{
"id":1,
"firstName":"Mahesh",
"lastName":"Patil",
"age":70

}
```


Press send button

Create similar records and send

For display use http type as :- get

Enter url as localhost:8080/api/students

You will get all the records in postman

You can also type the same url on browser you will get all the records

Similarly you can delete record of student by using

Type as delete

And url as localhost:8080/api/students/1