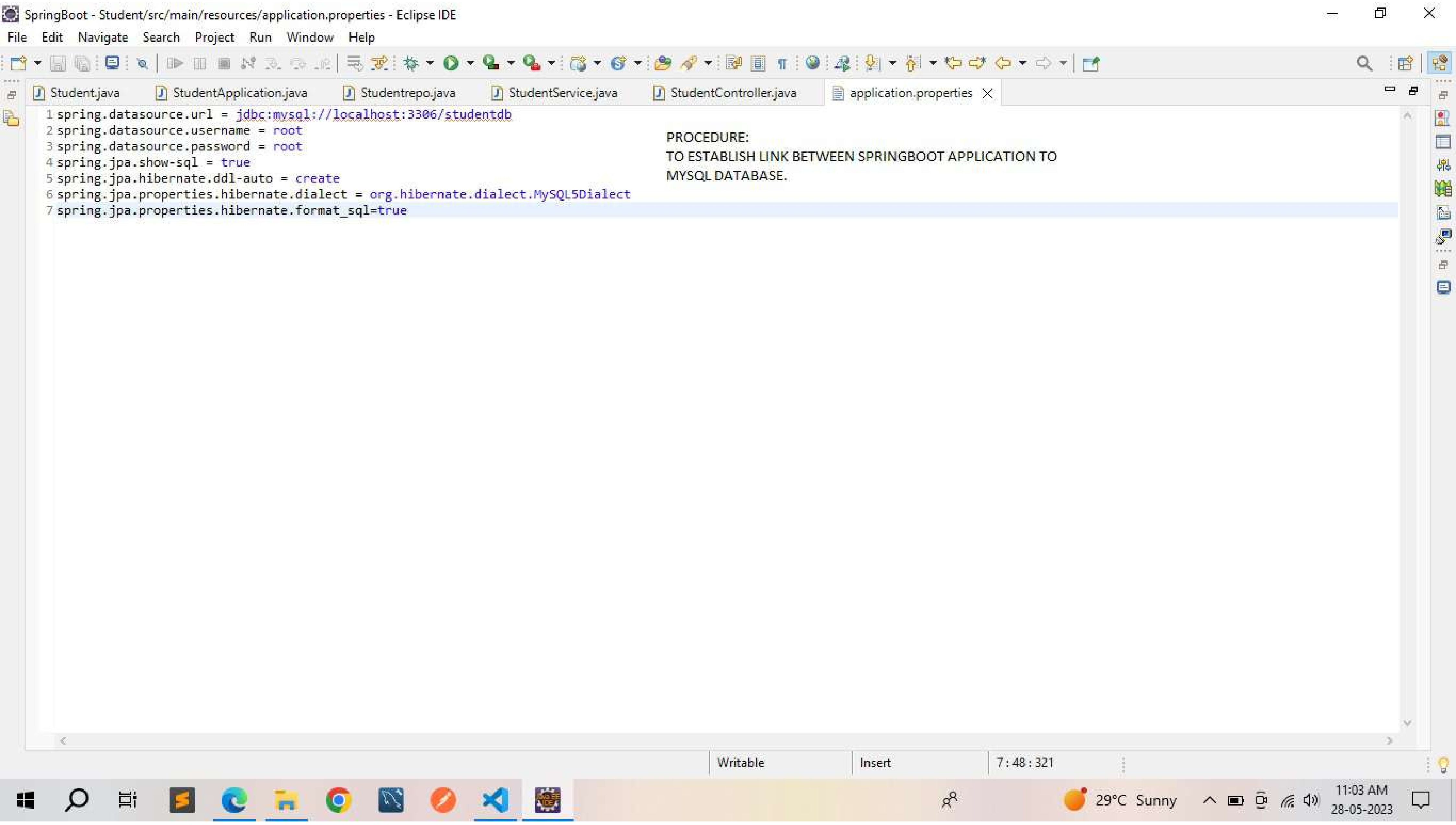
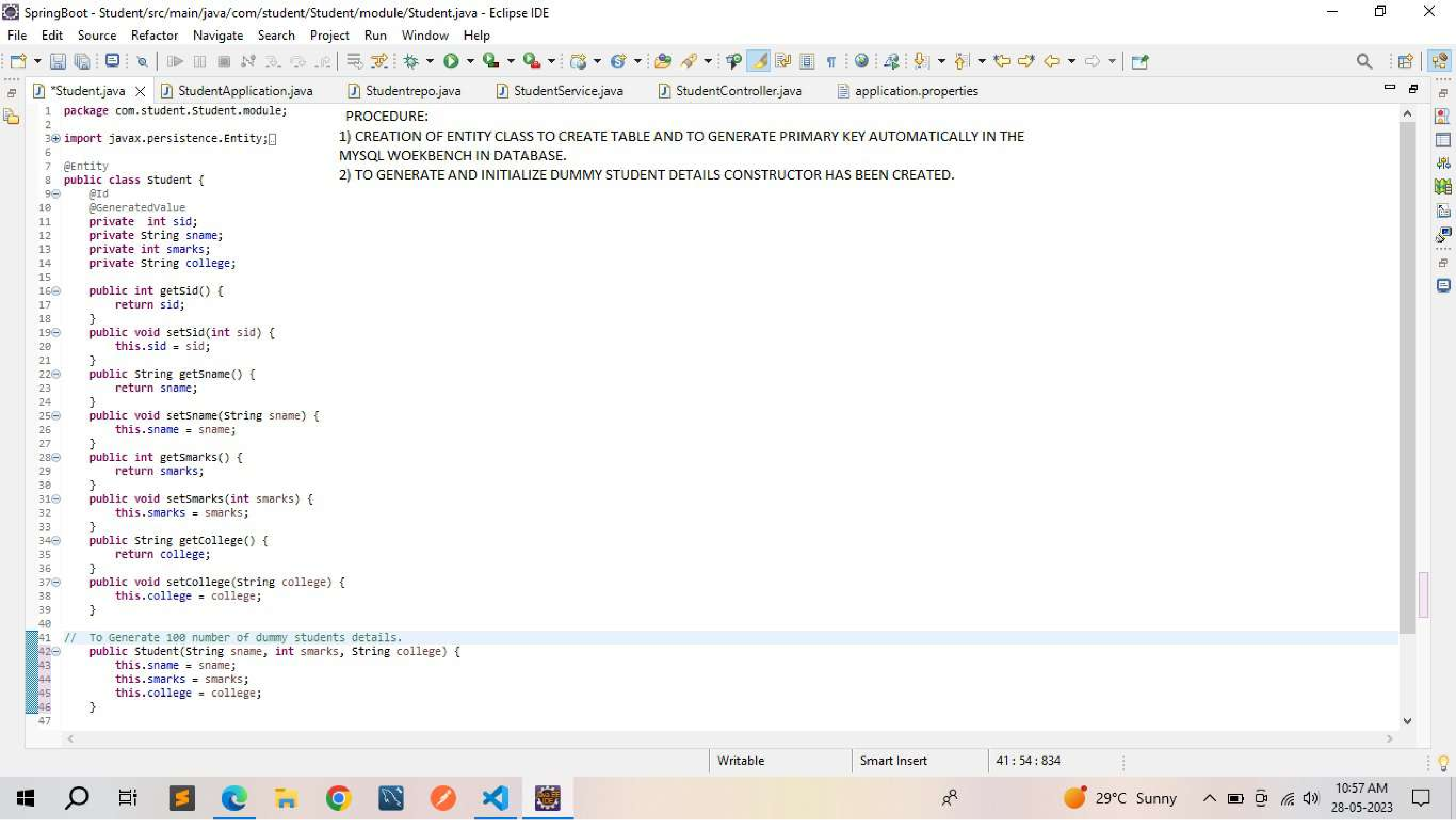


```
spring.datasource.url = jdbc:mysql://localhost:3306/studentdb
spring.datasource.username = root
spring.datasource.password = root
spring.jpa.show-sql = true
spring.jpa.hibernate.ddl-auto = create
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
spring.jpa.properties.hibernate.format_sql=true
```

PROCEDURE:
TO ESTABLISH LINK BETWEEN SPRINGBOOT APPLICATION TO MYSQL DATABASE.

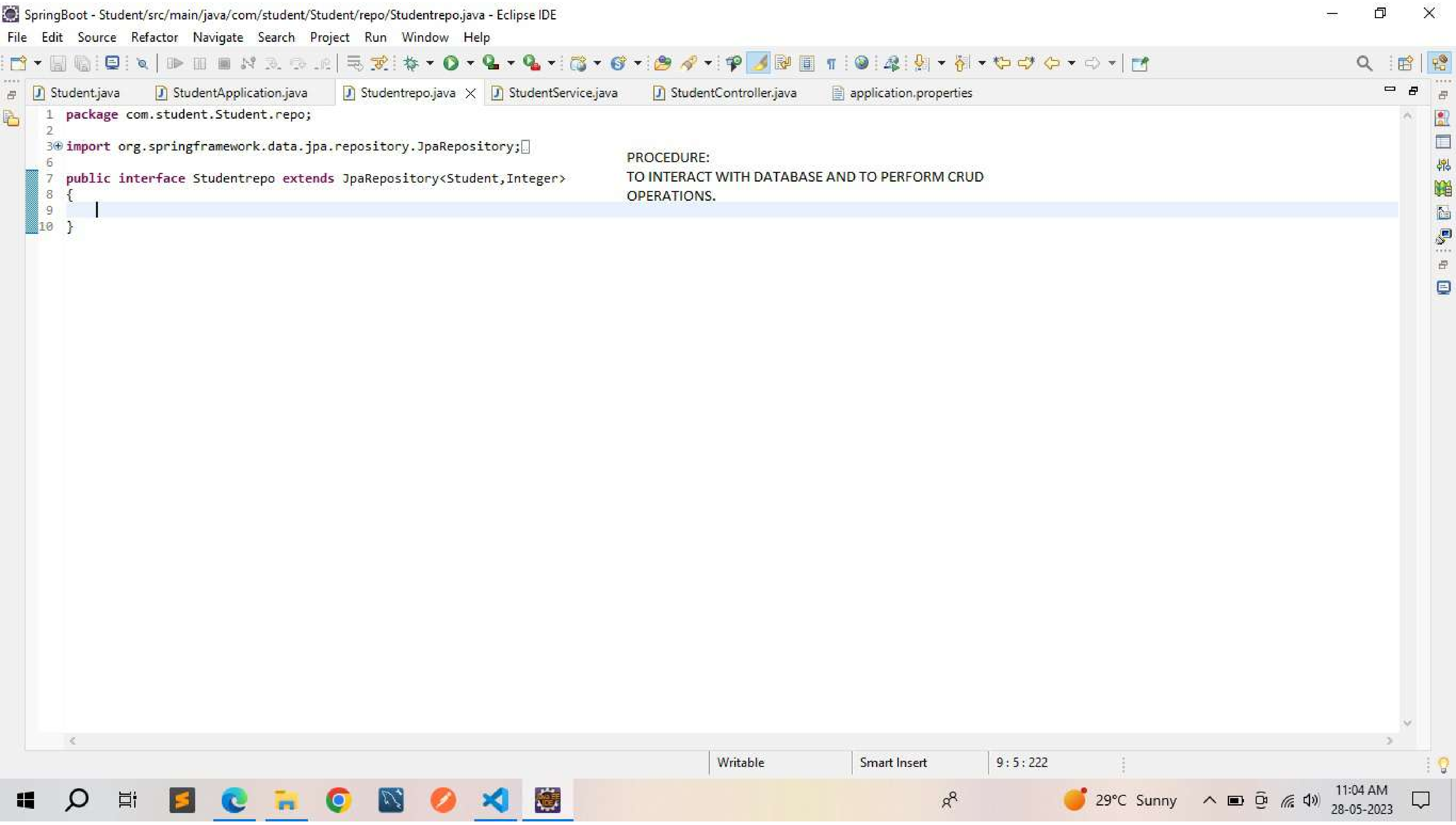


PROCEDURE:

1) CREATION OF ENTITY CLASS TO CREATE TABLE AND TO GENERATE PRIMARY KEY AUTOMATICALLY IN THE MYSQL WOEKBENCH IN DATABASE.

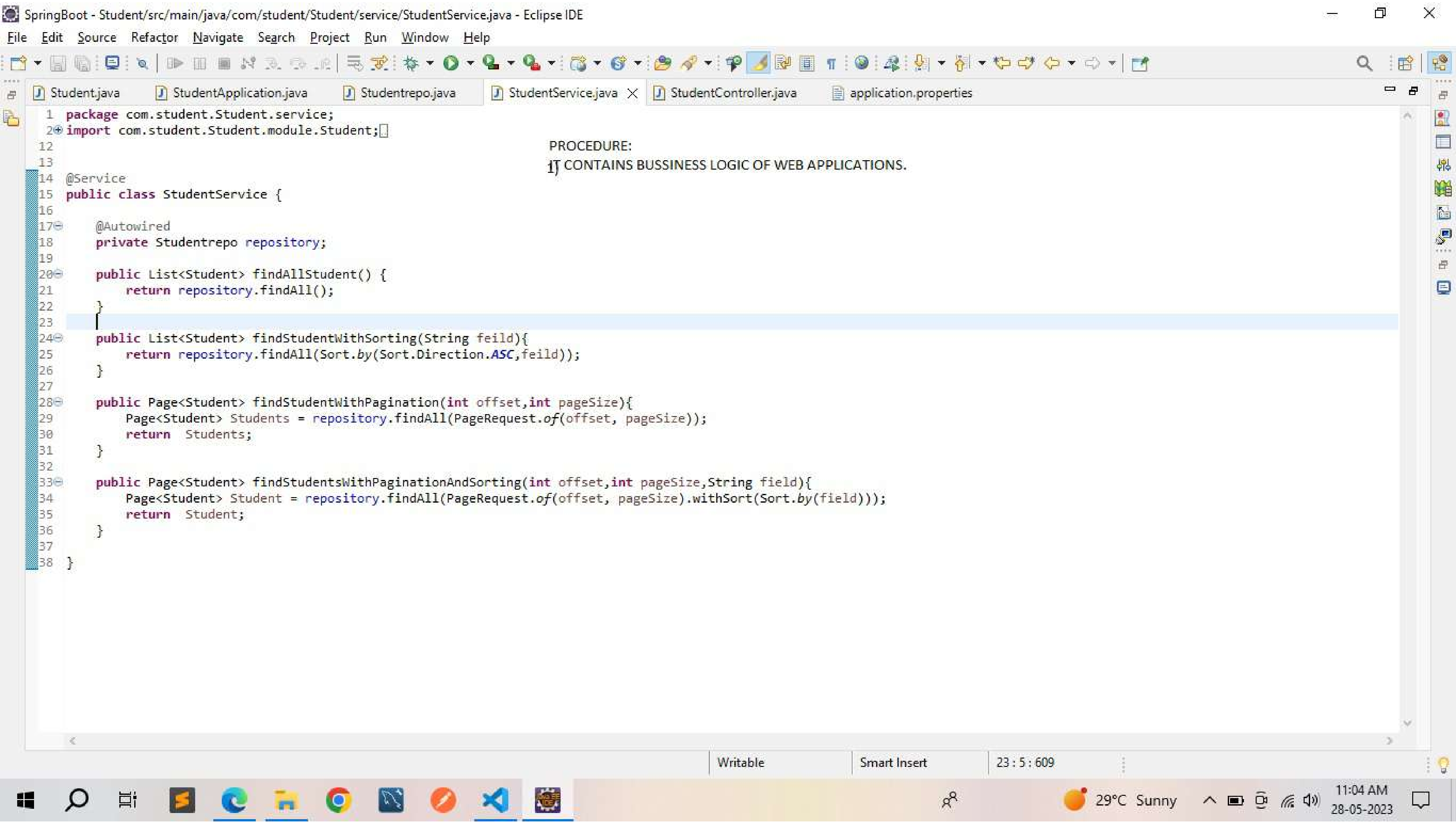
2) TO GENERATE AND INITIALIZE DUMMY STUDENT DETAILS CONSTRUCTOR HAS BEEN CREATED.

```java
package com.student.Student.module;

import javax.persistence.Entity;

@Entity
public class Student {
    @Id
    @GeneratedValue
    private  int sid;
    private String sname;
    private int smarks;
    private String college;

    public int getSid() {
        return sid;
    }
    public void setSid(int sid) {
        this.sid = sid;
    }
    public String getSname() {
        return sname;
    }
    public void setSname(String sname) {
        this.sname = sname;
    }
    public int getSmarks() {
        return smarks;
    }
    public void setSmarks(int smarks) {
        this.smarks = smarks;
    }
    public String getCollege() {
        return college;
    }
    public void setCollege(String college) {
        this.college = college;
    }

//  To Generate 100 number of dummy students details.
    public Student(String sname, int smarks, String college) {
        this.sname = sname;
        this.smarks = smarks;
        this.college = college;
    }

```



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

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

public interface Studentrepo extends JpaRepository<Student,Integer>
{

}
```

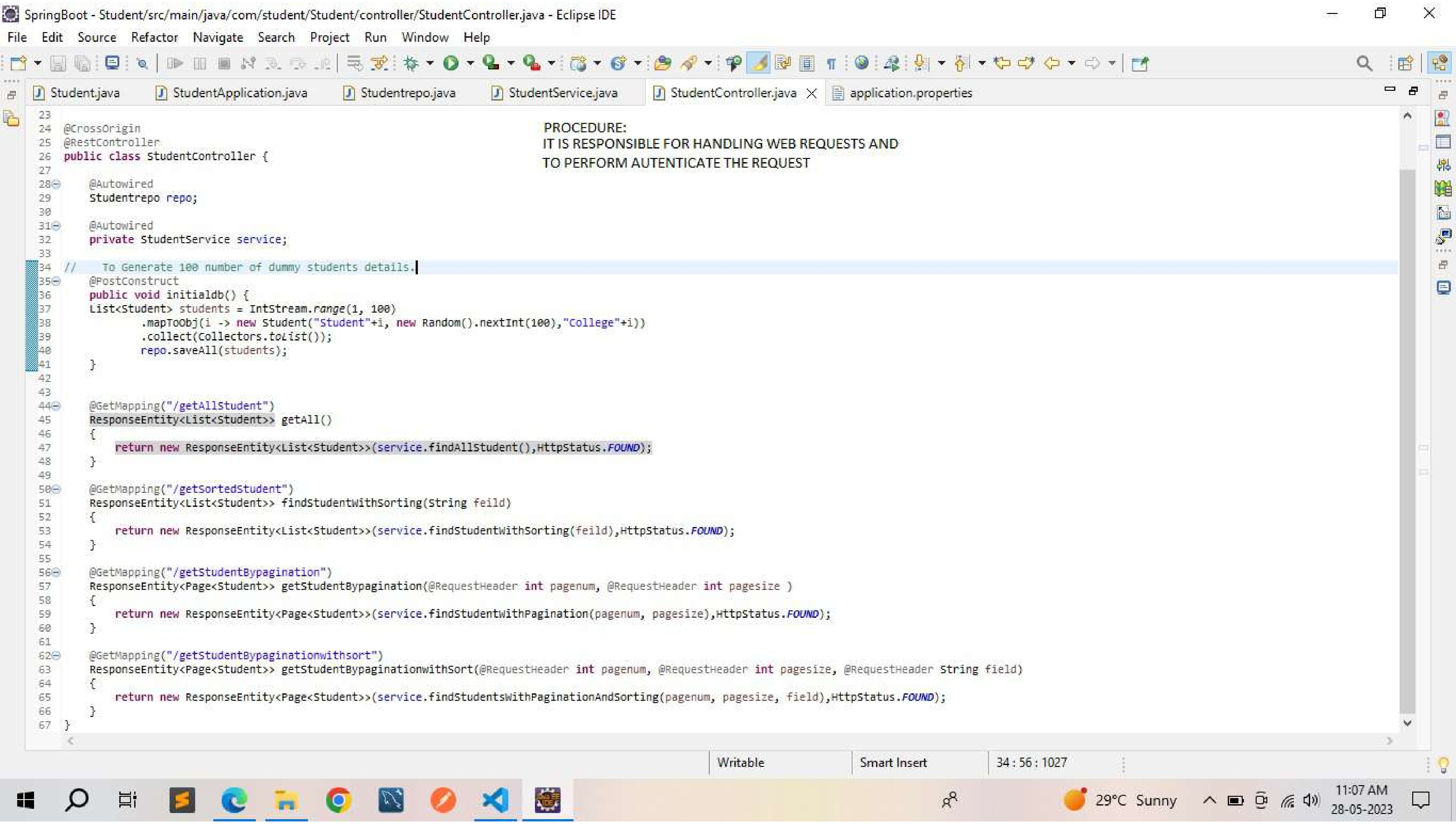
PROCEDURE:
TO INTERACT WITH DATABASE AND TO PERFORM CRUD OPERATIONS.


SpringBoot - Student/src/main/java/com/student/Student/service/StudentService.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Student.java StudentApplication.java Studentrepo.java StudentService.java StudentController.java application.properties
Writable Smart Insert 23 : 5 : 609
29°C Sunny
11:04 AM
28-05-2023

PROCEDURE:

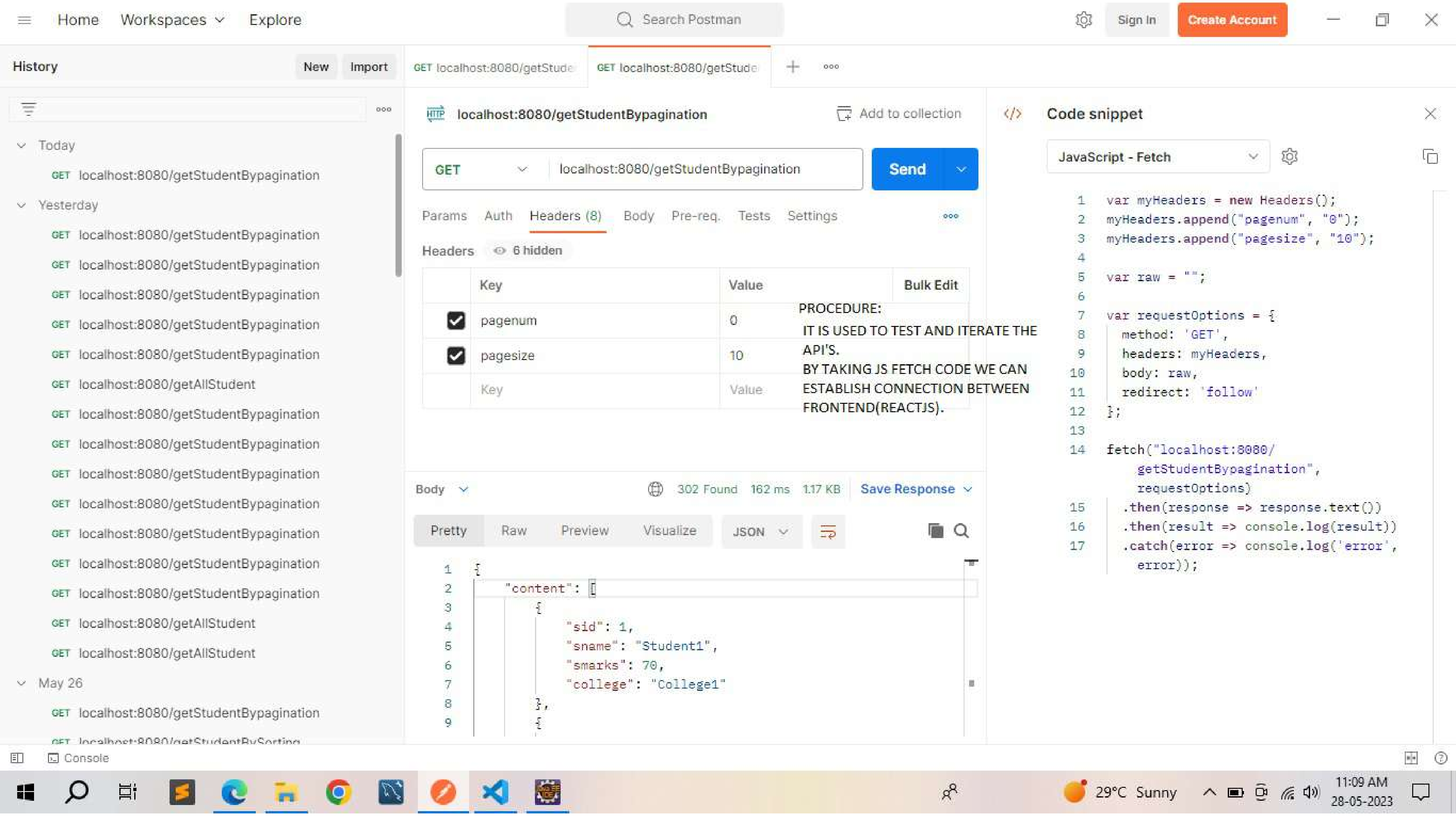
IT CONTAINS BUSSINESS LOGIC OF WEB APPLICATIONS.

```java
package com.student.Student.service;
import com.student.Student.module.Student;[]


@Service
public class StudentService {

    @Autowired
    private Studentrepo repository;

    public List<Student> findAllStudent() {
        return repository.findAll();
    }

    public List<Student> findStudentWithSorting(String feild){
        return repository.findAll(Sort.by(Sort.Direction.ASC,feild));
    }

    public Page<Student> findStudentWithPagination(int offset,int pageSize){
        Page<Student> Students = repository.findAll(PageRequest.of(offset, pageSize));
        return  Students;
    }

    public Page<Student> findStudentsWithPaginationAndSorting(int offset,int pageSize,String field){
        Page<Student> Student = repository.findAll(PageRequest.of(offset, pageSize).withSort(Sort.by(field)));
        return  Student;
    }

}
```



PROCEDURE:
IT IS RESPONSIBLE FOR HANDLING WEB REQUESTS AND
TO PERFORM AUTENTICATE THE REQUEST

```java

@CrossOrigin
@RestController
public class StudentController {

    @Autowired
    Studentrepo repo;

    @Autowired
    private StudentService service;

//    To Generate 100 number of dummy students details.
    @PostConstruct
    public void initialdb() {
    List<Student> students = IntStream.range(1, 100)
            .mapToObj(i -> new Student("Student"+i, new Random().nextInt(100),"College"+i))
            .collect(Collectors.toList());
            repo.saveAll(students);
    }


    @GetMapping("/getAllStudent")
    ResponseEntity<List<Student>> getAll()
    {
        return new ResponseEntity<List<Student>>(service.findAllStudent(),HttpStatus.FOUND);
    }

    @GetMapping("/getSortedStudent")
    ResponseEntity<List<Student>> findStudentWithSorting(String feild)
    {
        return new ResponseEntity<List<Student>>(service.findStudentWithSorting(feild),HttpStatus.FOUND);
    }

    @GetMapping("/getStudentBypagination")
    ResponseEntity<Page<Student>> getStudentBypagination(@RequestHeader int pagenum, @RequestHeader int pagesize )
    {
        return new ResponseEntity<Page<Student>>(service.findStudentWithPagination(pagenum, pagesize),HttpStatus.FOUND);
    }

    @GetMapping("/getStudentBypaginationwithsort")
    ResponseEntity<Page<Student>> getStudentBypaginationwithSort(@RequestHeader int pagenum, @RequestHeader int pagesize, @RequestHeader String field)
    {
        return new ResponseEntity<Page<Student>>(service.findStudentsWithPaginationAndSorting(pagenum, pagesize, field),HttpStatus.FOUND);
    }
}
```



Today
GET localhost:8080/getStudentBypagination

Yesterday
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getAllStudent
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBypagination
GET localhost:8080/getAllStudent
GET localhost:8080/getAllStudent

May 26
GET localhost:8080/getStudentBypagination
GET localhost:8080/getStudentBySorting

localhost:8080/getStudentBypagination

GET localhost:8080/getStudentBypagination

Headers 6 hidden

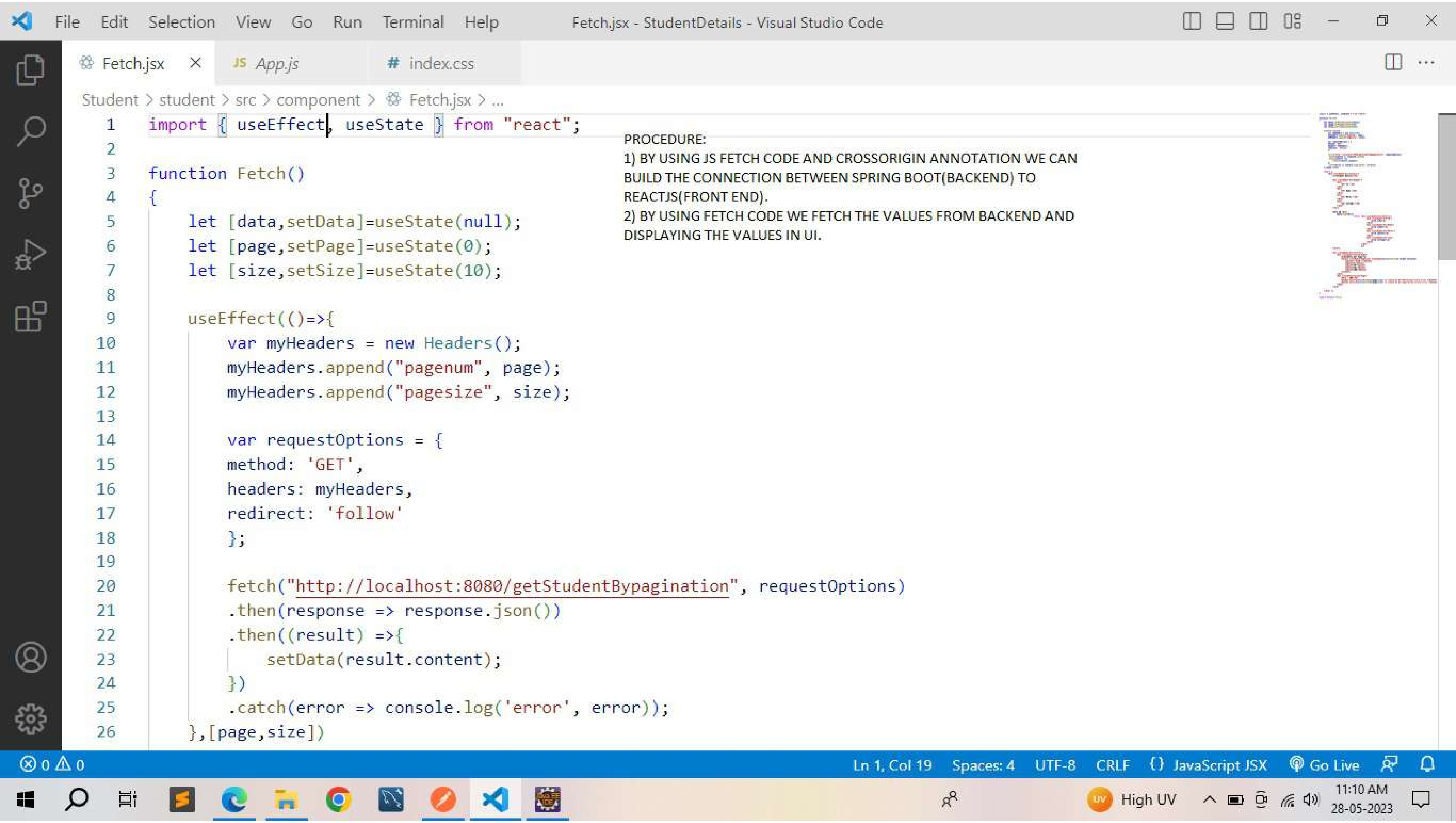
	Key	Value	Bulk Edit
✓	pagenum	0	
✓	pagesize	10	
	Key	Value	

PROCEDURE:
IT IS USED TO TEST AND ITERATE THE API'S.
BY TAKING JS FETCH CODE WE CAN ESTABLISH CONNECTION BETWEEN FRONTEND(REACTJS).

Body 302 Found 162 ms 1.17 KB

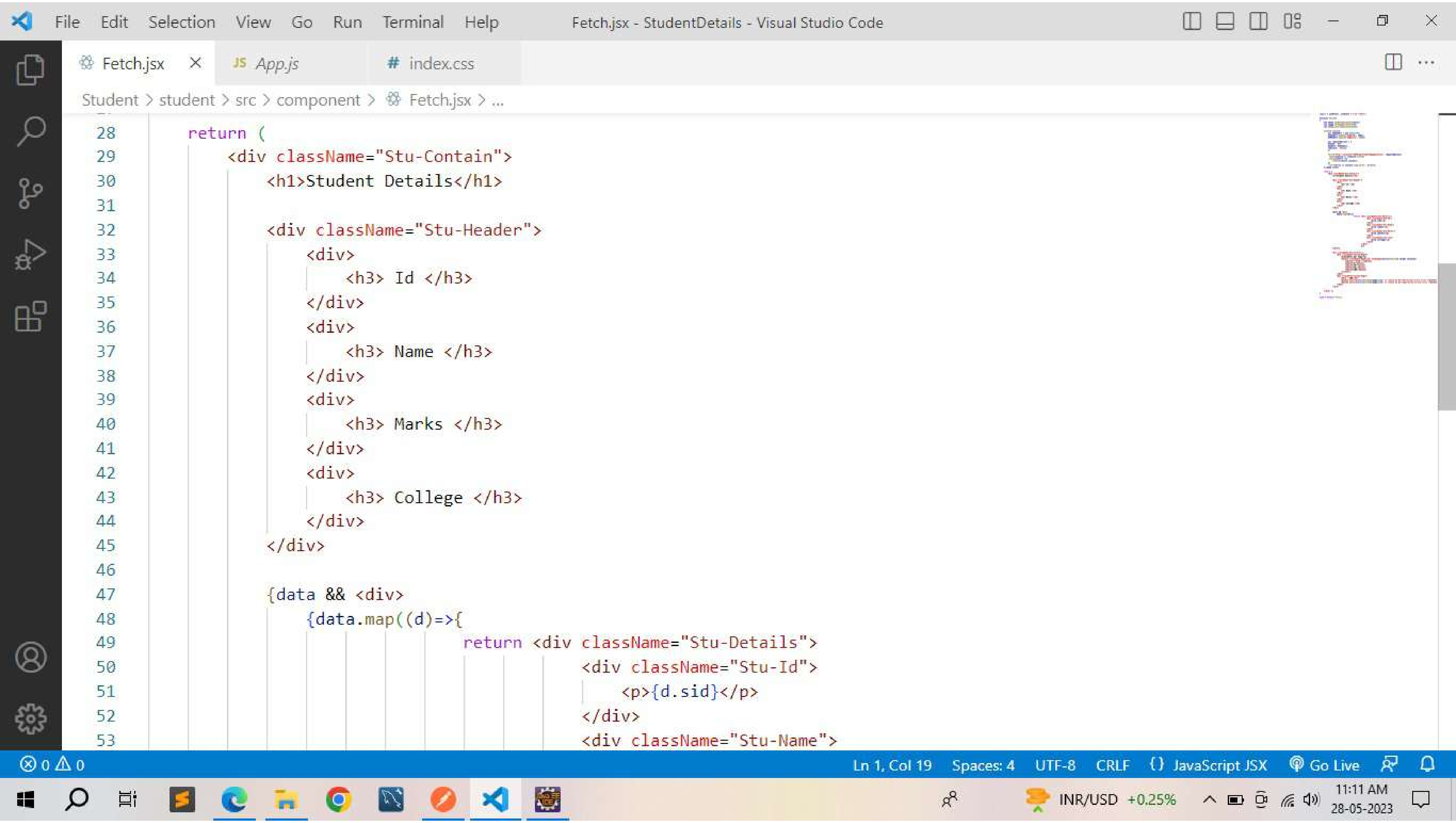
```
{
    "content": [
        {
            "sid": 1,
            "sname": "Student1",
            "smarks": 70,
            "college": "College1"
        },
        {
```

```
var myHeaders = new Headers();
myHeaders.append("pagenum", "0");
myHeaders.append("pagesize", "10");

var raw = "";

var requestOptions = {
  method: 'GET',
  headers: myHeaders,
  body: raw,
  redirect: 'follow'
};

fetch("localhost:8080/
    getStudentBypagination",
    requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error',
    error));
```


File Edit Selection View Go Run Terminal Help
Fetch.jsx - StudentDetails - Visual Studio Code
Fetch.jsx
App.js
index.css
Student > student > src > component > Fetch.jsx > ...

```
import { useEffect, useState } from "react";

function Fetch()
{
    let [data,setData]=useState(null);
    let [page,setPage]=useState(0);
    let [size,setSize]=useState(10);

    useEffect(()=>{
        var myHeaders = new Headers();
        myHeaders.append("pagenum", page);
        myHeaders.append("pagesize", size);

        var requestOptions = {
        method: 'GET',
        headers: myHeaders,
        redirect: 'follow'
        };

        fetch("http://localhost:8080/getStudentBypagination", requestOptions)
        .then(response => response.json())
        .then((result) =>{
            setData(result.content);
        })
        .catch(error => console.log('error', error));
    },[page,size])
```

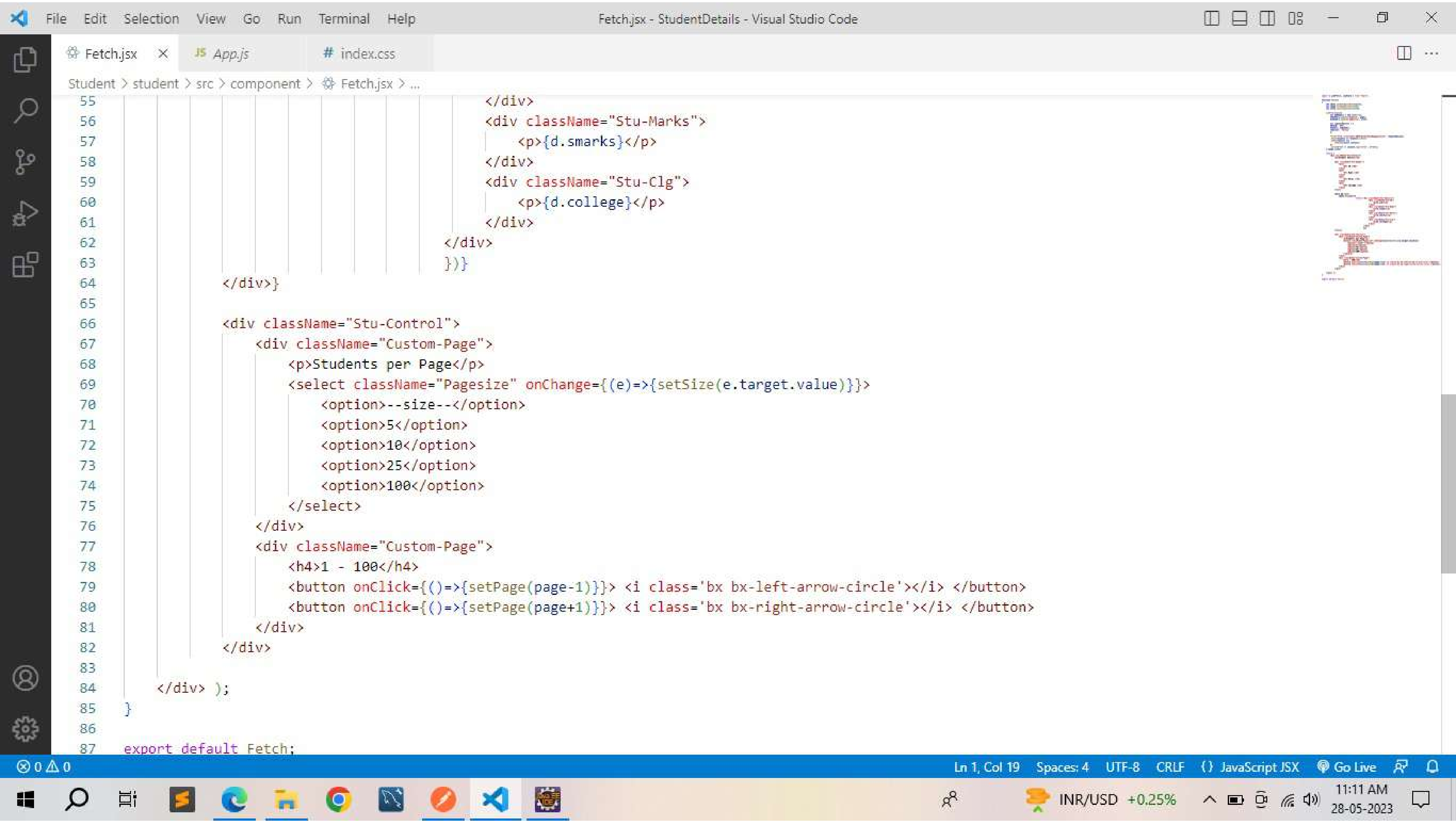
PROCEDURE:
1) BY USING JS FETCH CODE AND CROSSORIGIN ANNOTATION WE CAN BUILD THE CONNECTION BETWEEN SPRING BOOT(BACKEND) TO REACTJS(FRONT END).
2) BY USING FETCH CODE WE FETCH THE VALUES FROM BACKEND AND DISPLAYING THE VALUES IN UI.



File Edit Selection View Go Run Terminal Help
Fetch.jsx - StudentDetails - Visual Studio Code
Fetch.jsx
App.js
index.css
Student > student > src > component > Fetch.jsx > ...

```
    return (
        <div className="Stu-Contain">
            <h1>Student Details</h1>

            <div className="Stu-Header">
                <div>
                    <h3> Id </h3>
                </div>
                <div>
                    <h3> Name </h3>
                </div>
                <div>
                    <h3> Marks </h3>
                </div>
                <div>
                    <h3> College </h3>
                </div>
            </div>

            {data && <div>
                {data.map((d)=>{
                                return <div className="Stu-Details">
                                            <div className="Stu-Id">
                                                <p>{d.sid}</p>
                                            </div>
                                            <div className="Stu-Name">
```

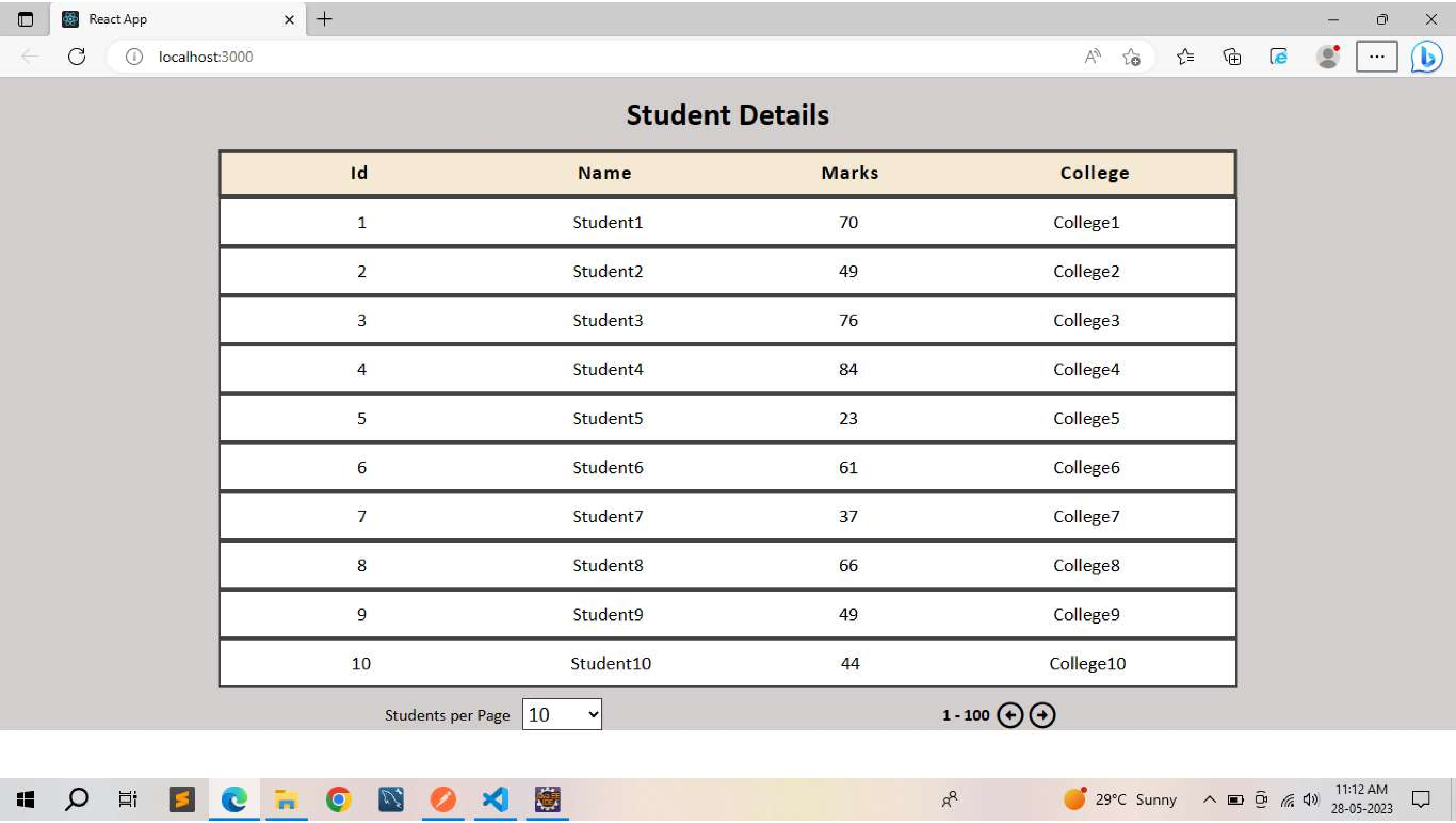
⊗ 0 ⚠ 0
Ln 1, Col 19 Spaces: 4 UTF-8 CRLF {} JavaScript JSX Go Live
INR/USD +0.25%
11:11 AM
28-05-2023



File Edit Selection View Go Run Terminal Help
Fetch.jsx - StudentDetails - Visual Studio Code
Fetch.jsx
App.js
index.css
Student > student > src > component > Fetch.jsx > ...

```
                                        </div>
                                        <div className="Stu-Marks">
                                            <p>{d.smarks}</p>
                                        </div>
                                        <div className="Stu-Clg">
                                            <p>{d.college}</p>
                                        </div>
                                    </div>
                                    ))}
                </div>}

                <div className="Stu-Control">
                    <div className="Custom-Page">
                        <p>Students per Page</p>
                        <select className="Pagesize" onChange={(e)=>{setSize(e.target.value)}}>
                            <option>--size--</option>
                            <option>5</option>
                            <option>10</option>
                            <option>25</option>
                            <option>100</option>
                        </select>
                    </div>
                    <div className="Custom-Page">
                        <h4>1 - 100</h4>
                        <button onClick={()=>{setPage(page-1)}}> <i class='bx bx-left-arrow-circle'></i> </button>
                        <button onClick={()=>{setPage(page+1)}}> <i class='bx bx-right-arrow-circle'></i> </button>
                    </div>
                </div>

        </div> );
}

export default Fetch;
```

⊗ 0 ⚠ 0
Ln 1, Col 19 Spaces: 4 UTF-8 CRLF {} JavaScript JSX Go Live
INR/USD +0.25%
11:11 AM
28-05-2023

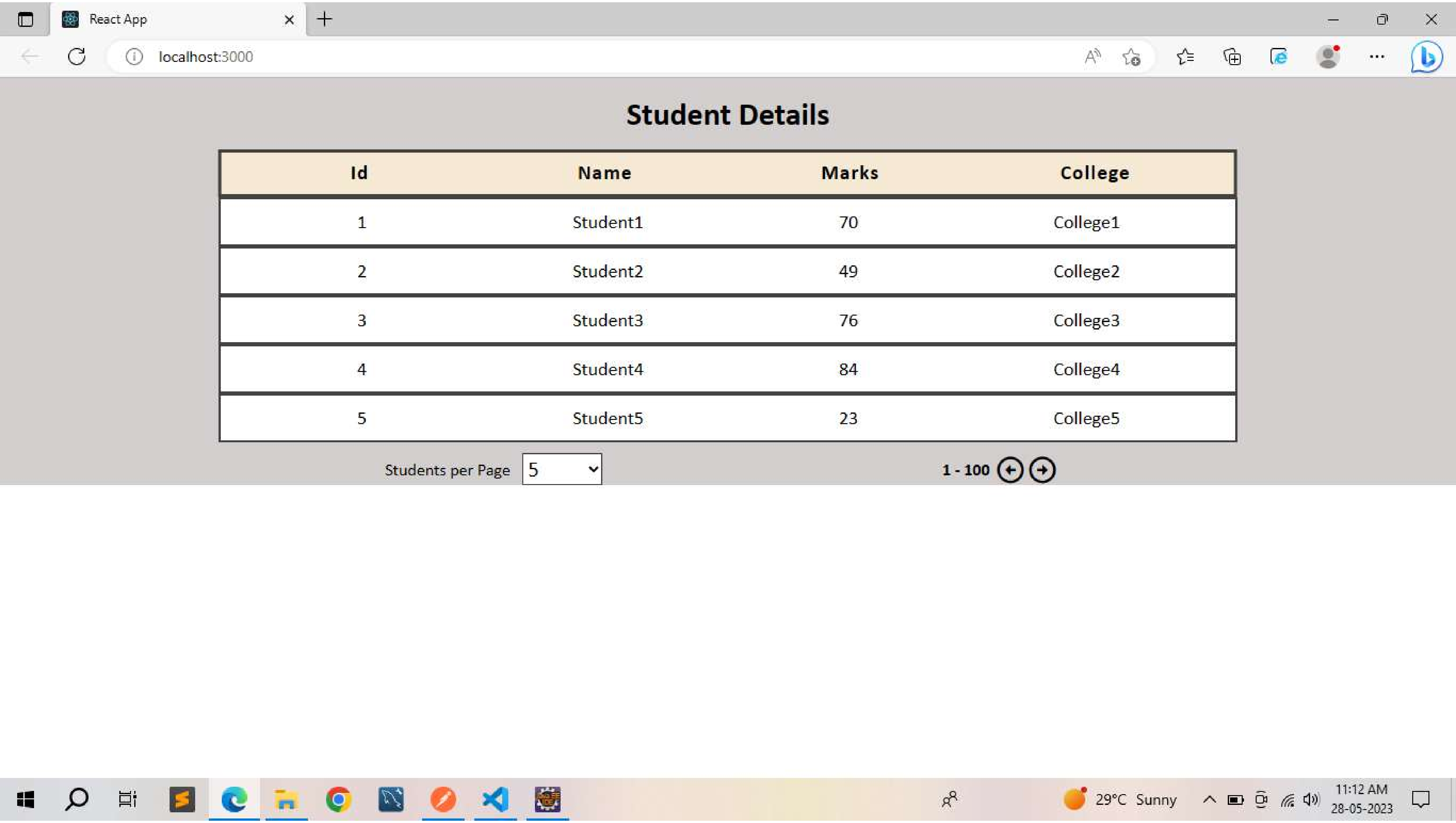


Student Details

Id	Name	Marks	College
1	Student1	70	College1
2	Student2	49	College2
3	Student3	76	College3
4	Student4	84	College4
5	Student5	23	College5
6	Student6	61	College6
7	Student7	37	College7
8	Student8	66	College8
9	Student9	49	College9
10	Student10	44	College10

Students per Page: 10

1 - 100



Student Details

Id	Name	Marks	College
1	Student1	70	College1
2	Student2	49	College2
3	Student3	76	College3
4	Student4	84	College4
5	Student5	23	College5

Students per Page: 5

1 - 100