Java Academy

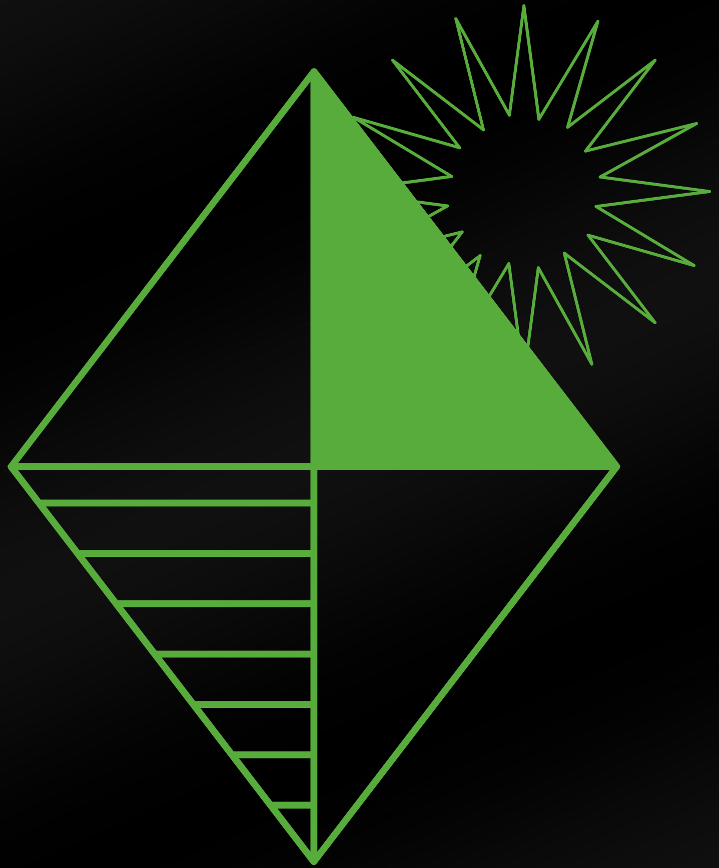
Student Grades Administration Service

Using:
Spring Data JPA, Security and MockMvc

By:
Pedro Jahir Hinojosa García

Buissness need

The educational institution aims to enhance its existing student management system by developing a secure, role-based API for managing student data.



Core requirements

- Secure Data Access
- Role-Based Authorization
- Efficiency & Automation
- Scalable Solution

Proyect foundations

spring®

Security

Safety

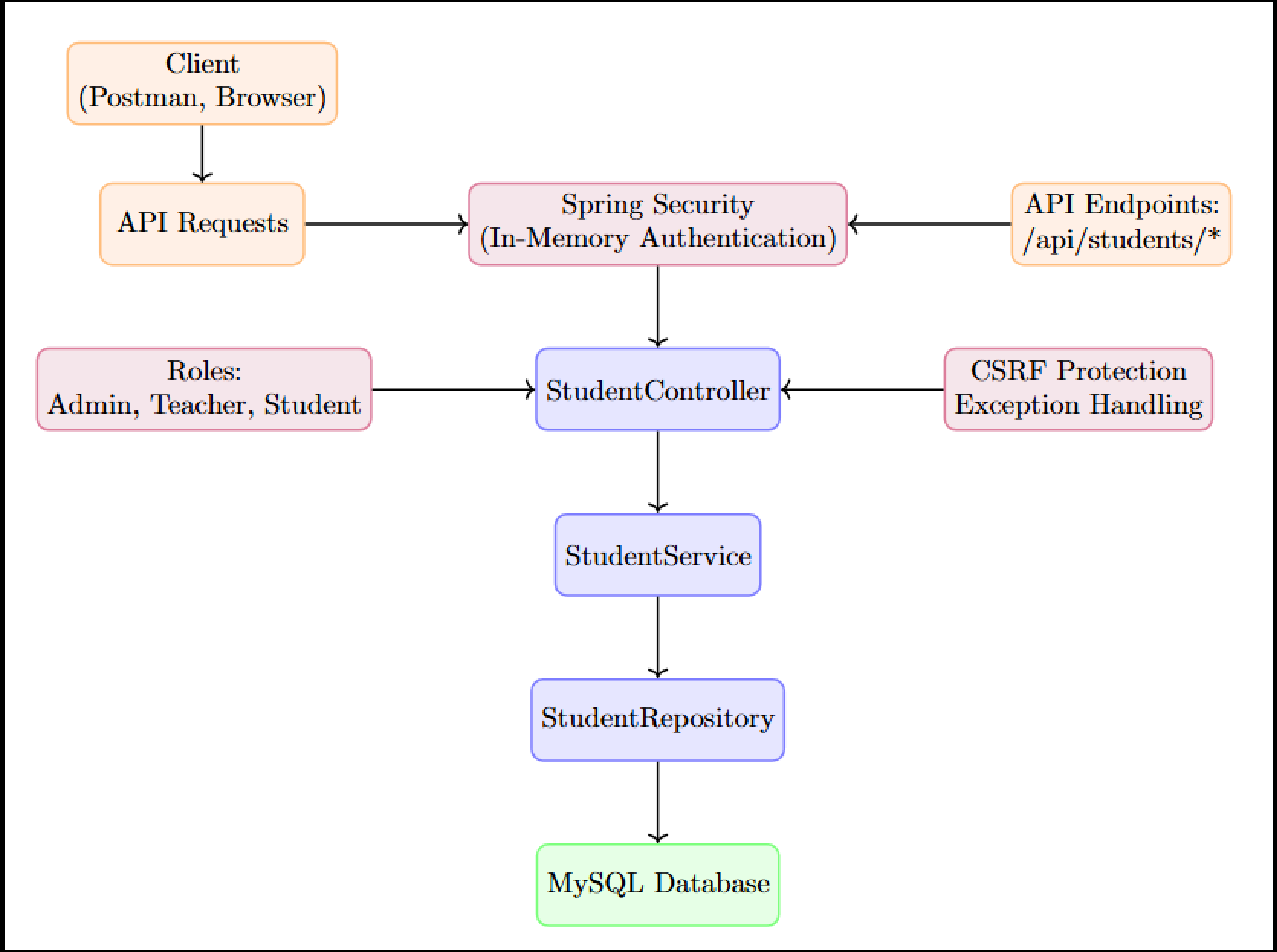
Data JPa

Database conection

MockMVC

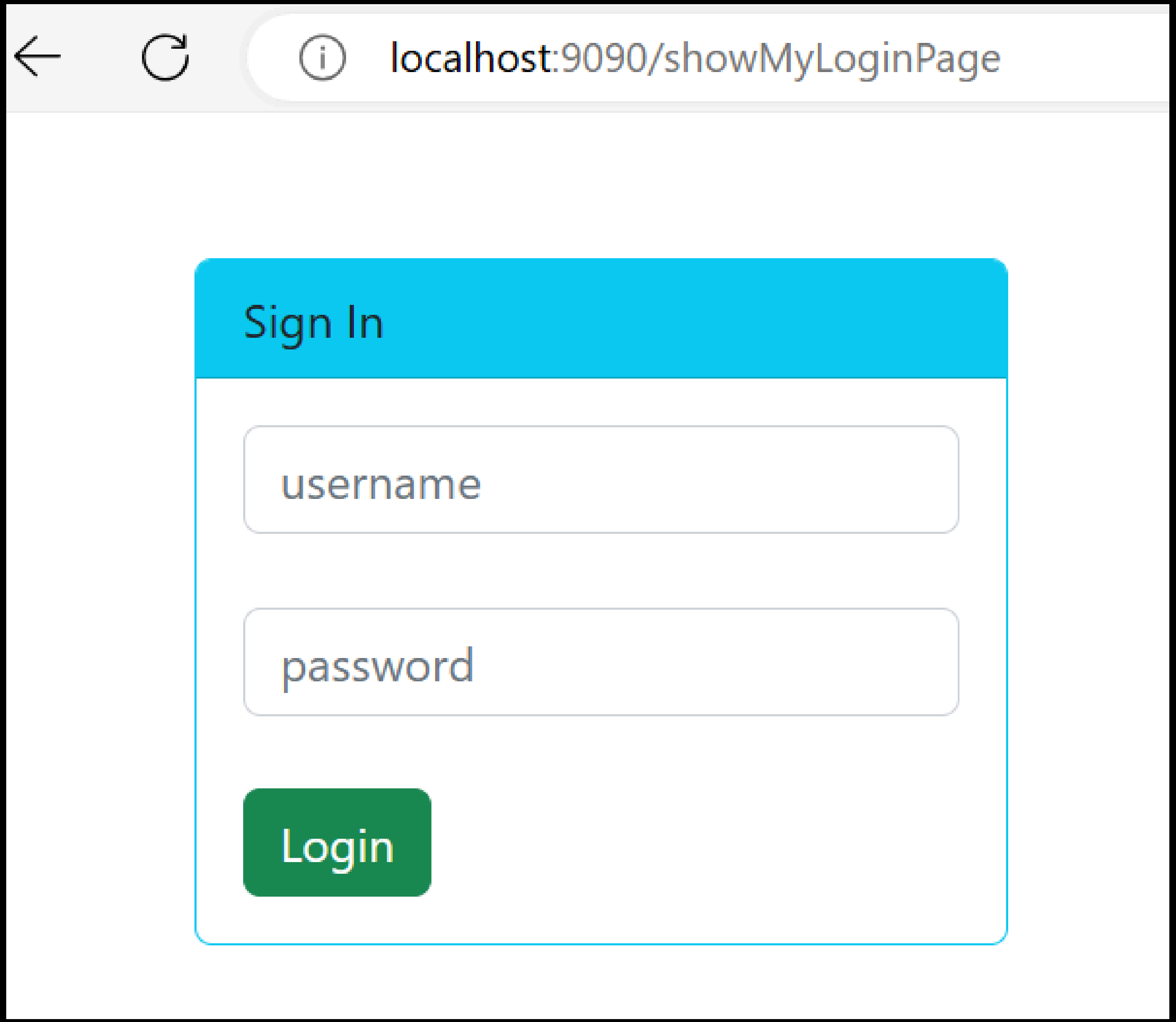
Testing

Code Diagram


Client (Postman, Browser)
API Requests
Spring Security (In-Memory Authentication)
API Endpoints: /api/students/*
Roles: Admin, Teacher, Student
StudentController
CSRF Protection Exception Handling
StudentService
StudentRepository
MySQL Database

login Methods

Browser


localhost:9090/showMyLoginPage

Sign In

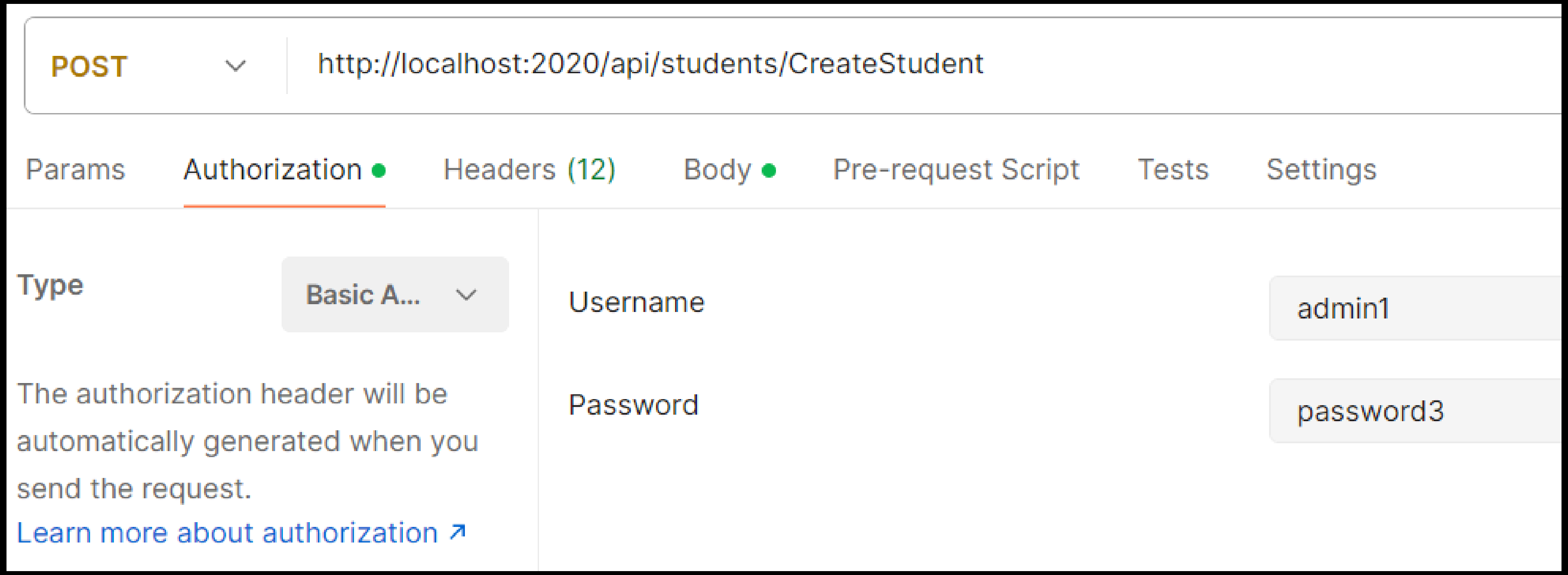
username

password

Login

All requests need autentification

Postman


POST http://localhost:2020/api/students/CreateStudent

Params | Authorization ● | Headers (12) | Body ● | Pre-request Script | Tests | Settings

Type: Basic A...

The authorization header will be automatically generated when you send the request.

Learn more about authorization ↗

Username: admin1

Password: password3

Only-ADMIN request

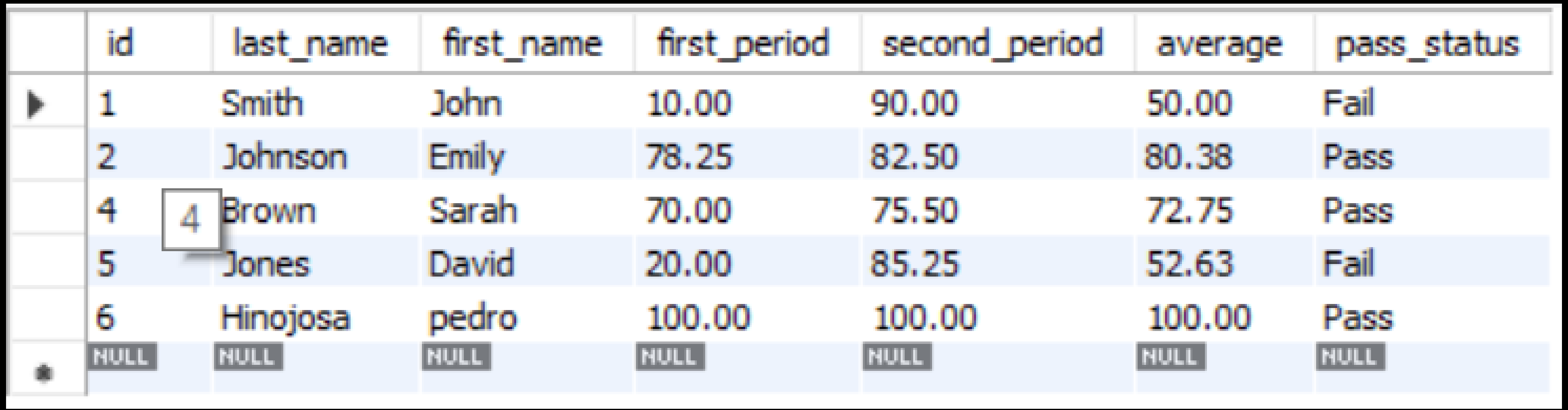
Deleting

Adding

Postman

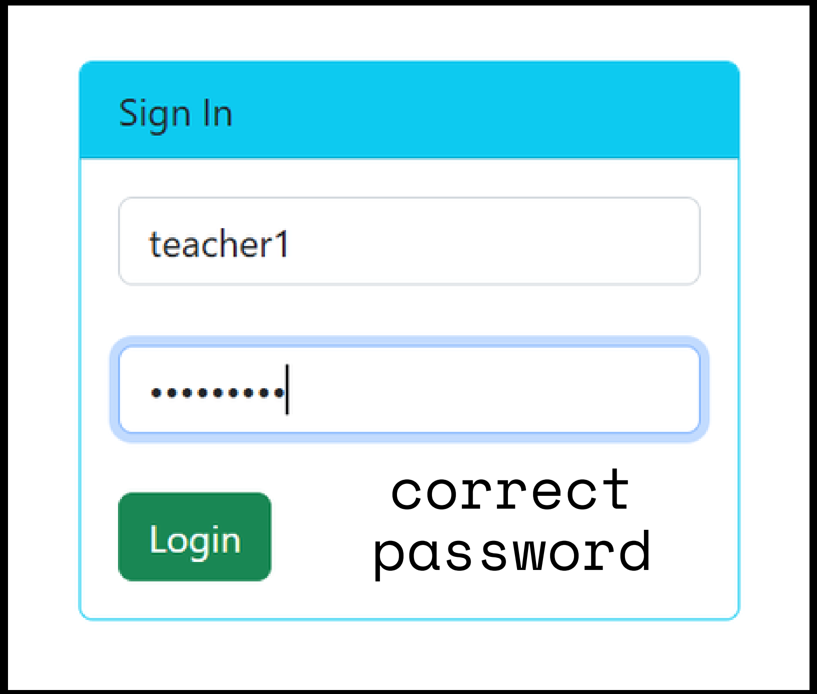
Needs API

After requests



	id	last_name	first_name	first_period	second_period	average	pass_status
▶	1	Smith	John	10.00	90.00	50.00	Fail
	2	Johnson	Emily	78.25	82.50	80.38	Pass
	4	Brown	Sarah	70.00	75.50	72.75	Pass
	5	Jones	David	20.00	85.25	52.63	Fail
	6	Hinojosa	pedro	100.00	100.00	100.00	Pass
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Theacher requests



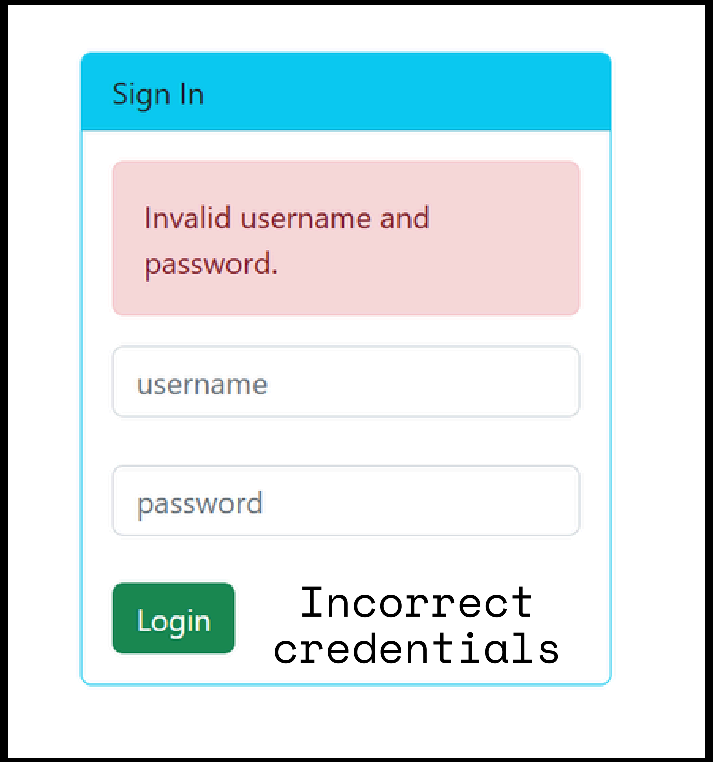
Sign In

teacher1

•••••••••

Login

correct password



Sign In

Invalid username and password.

username

password

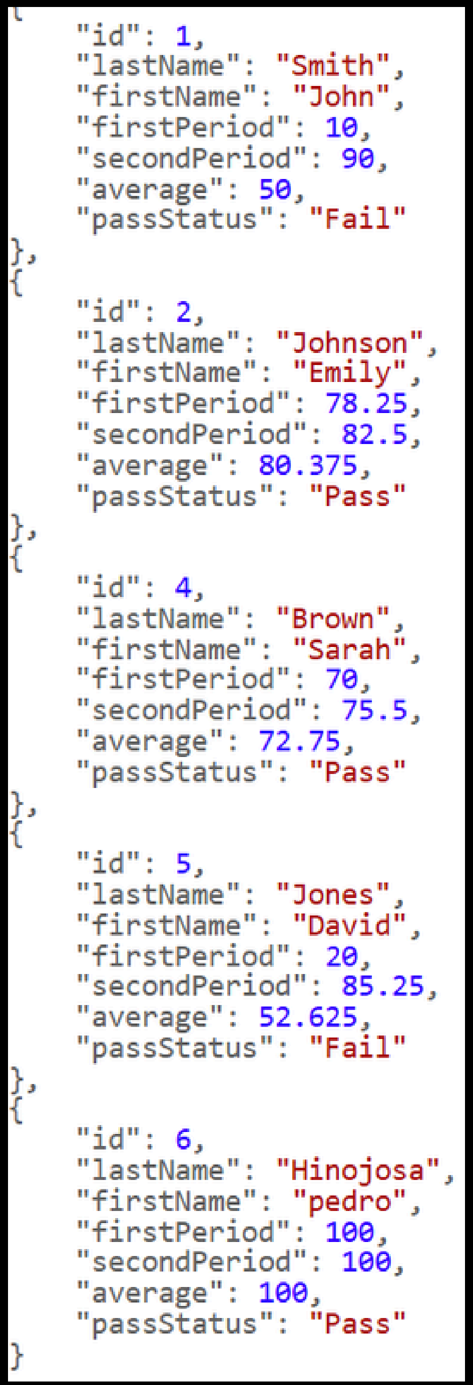
Login

Incorrect credentials

After passing Security

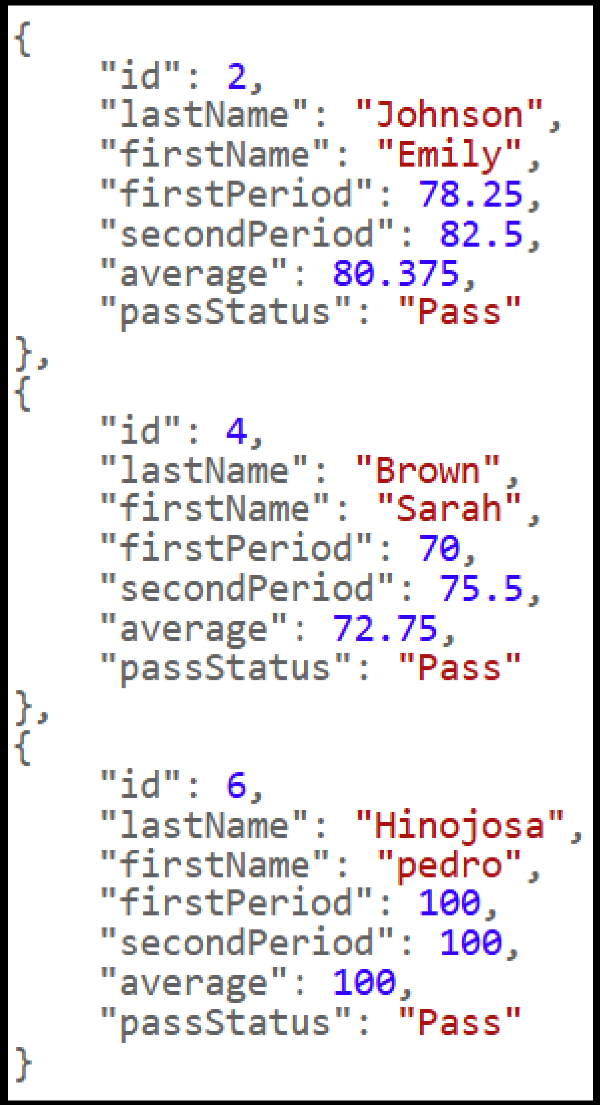
Not passing Security

See all students

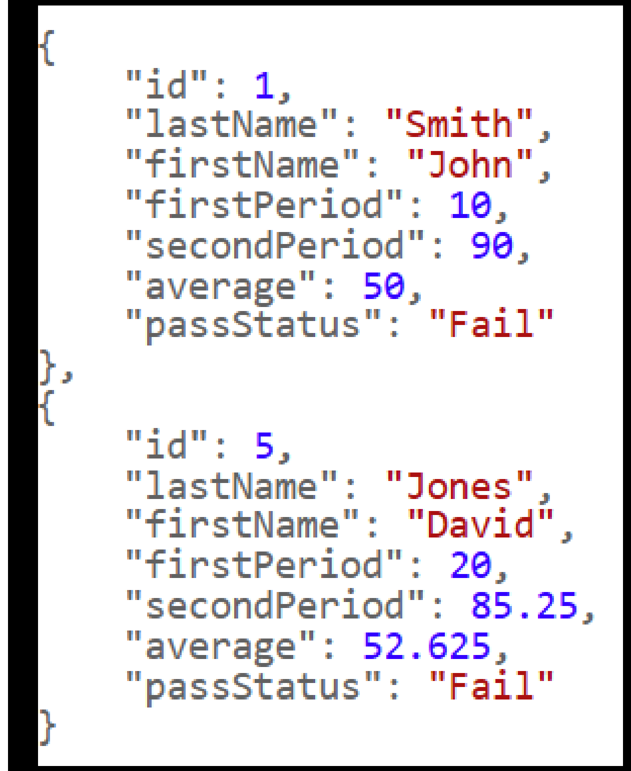


```
    "id": 1,
    "lastName": "Smith",
    "firstName": "John",
    "firstPeriod": 10,
    "secondPeriod": 90,
    "average": 50,
    "passStatus": "Fail"
},
{
    "id": 2,
    "lastName": "Johnson",
    "firstName": "Emily",
    "firstPeriod": 78.25,
    "secondPeriod": 82.5,
    "average": 80.375,
    "passStatus": "Pass"
},
{
    "id": 4,
    "lastName": "Brown",
    "firstName": "Sarah",
    "firstPeriod": 70,
    "secondPeriod": 75.5,
    "average": 72.75,
    "passStatus": "Pass"
},
{
    "id": 5,
    "lastName": "Jones",
    "firstName": "David",
    "firstPeriod": 20,
    "secondPeriod": 85.25,
    "average": 52.625,
    "passStatus": "Fail"
},
{
    "id": 6,
    "lastName": "Hinojosa",
    "firstName": "pedro",
    "firstPeriod": 100,
    "secondPeriod": 100,
    "average": 100,
    "passStatus": "Pass"
}
```

Pass/Fail Students



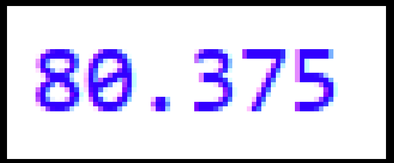
```
{
    "id": 2,
    "lastName": "Johnson",
    "firstName": "Emily",
    "firstPeriod": 78.25,
    "secondPeriod": 82.5,
    "average": 80.375,
    "passStatus": "Pass"
},
{
    "id": 4,
    "lastName": "Brown",
    "firstName": "Sarah",
    "firstPeriod": 70,
    "secondPeriod": 75.5,
    "average": 72.75,
    "passStatus": "Pass"
},
{
    "id": 6,
    "lastName": "Hinojosa",
    "firstName": "pedro",
    "firstPeriod": 100,
    "secondPeriod": 100,
    "average": 100,
    "passStatus": "Pass"
}
```



```
{
    "id": 1,
    "lastName": "Smith",
    "firstName": "John",
    "firstPeriod": 10,
    "secondPeriod": 90,
    "average": 50,
    "passStatus": "Fail"
},
{
    "id": 5,
    "lastName": "Jones",
    "firstName": "David",
    "firstPeriod": 20,
    "secondPeriod": 85.25,
    "average": 52.625,
    "passStatus": "Fail"
}
```

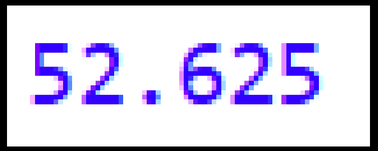
Every Average

2.-



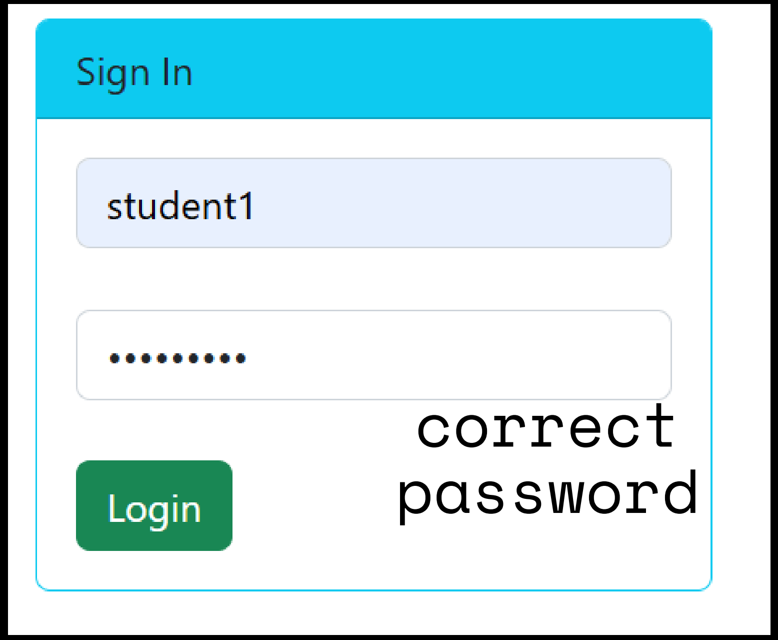
```
80.375
```

5.-



```
52.625
```

Student request


Sign In

student1

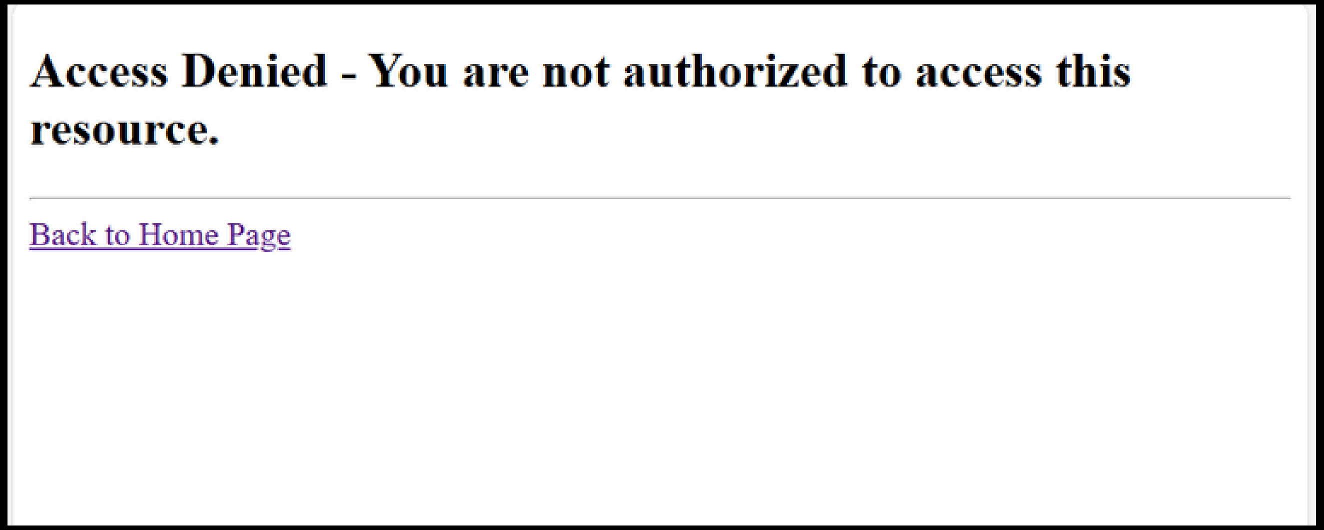
•••••••••

Login

correct password

After passing Security

If the credentials are correct but the user doesn't have permission

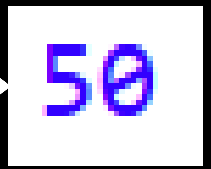

Access Denied - You are not authorized to access this resource.

Back to Home Page

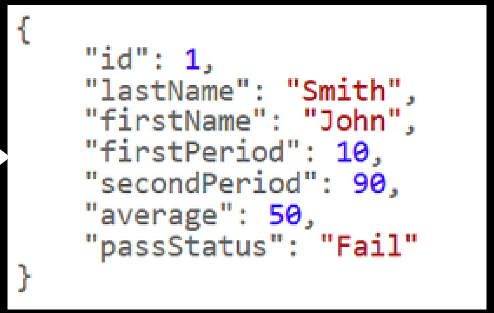
.html file

Can only access to it's own info

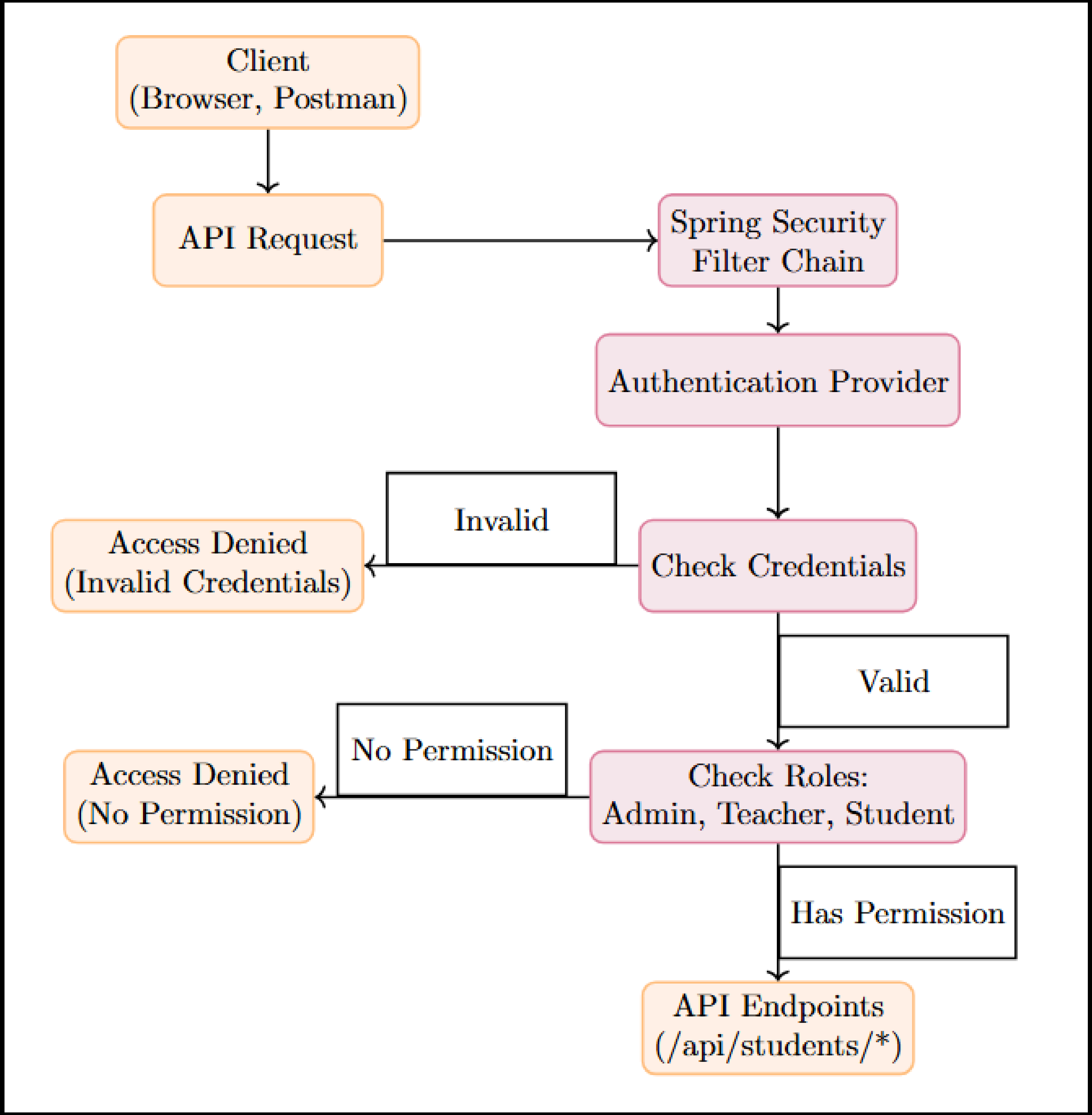
Average


50

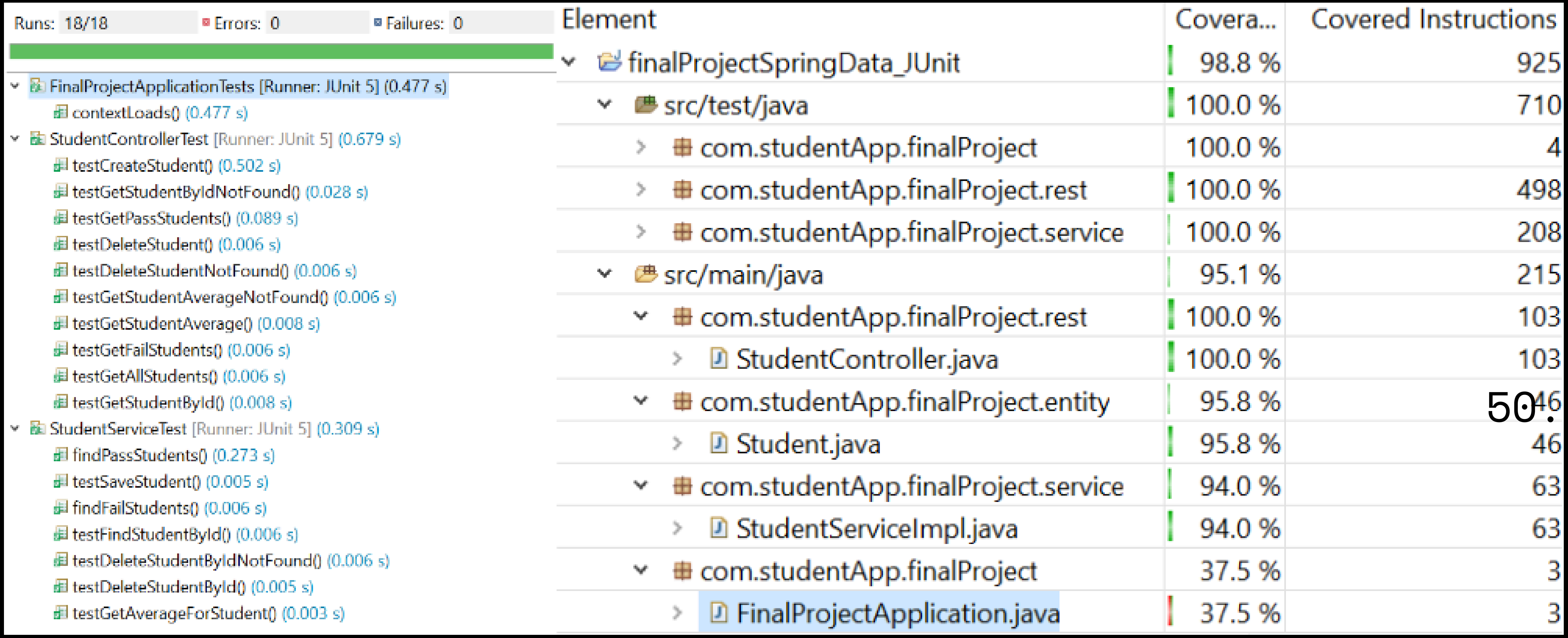
Full info


```
{
    "id": 1,
    "lastName": "Smith",
    "firstName": "John",
    "firstPeriod": 10,
    "secondPeriod": 90,
    "average": 50,
    "passStatus": "Fail"
}
```

Security Diagram


Client (Browser, Postman)
API Request
Spring Security Filter Chain
Authentication Provider
Check Credentials
Invalid
Access Denied (Invalid Credentials)
Valid
Check Roles: Admin, Teacher, Student
No Permission
Access Denied (No Permission)
Has Permission
API Endpoints (/api/students/*)

Testing without security
98% coverage

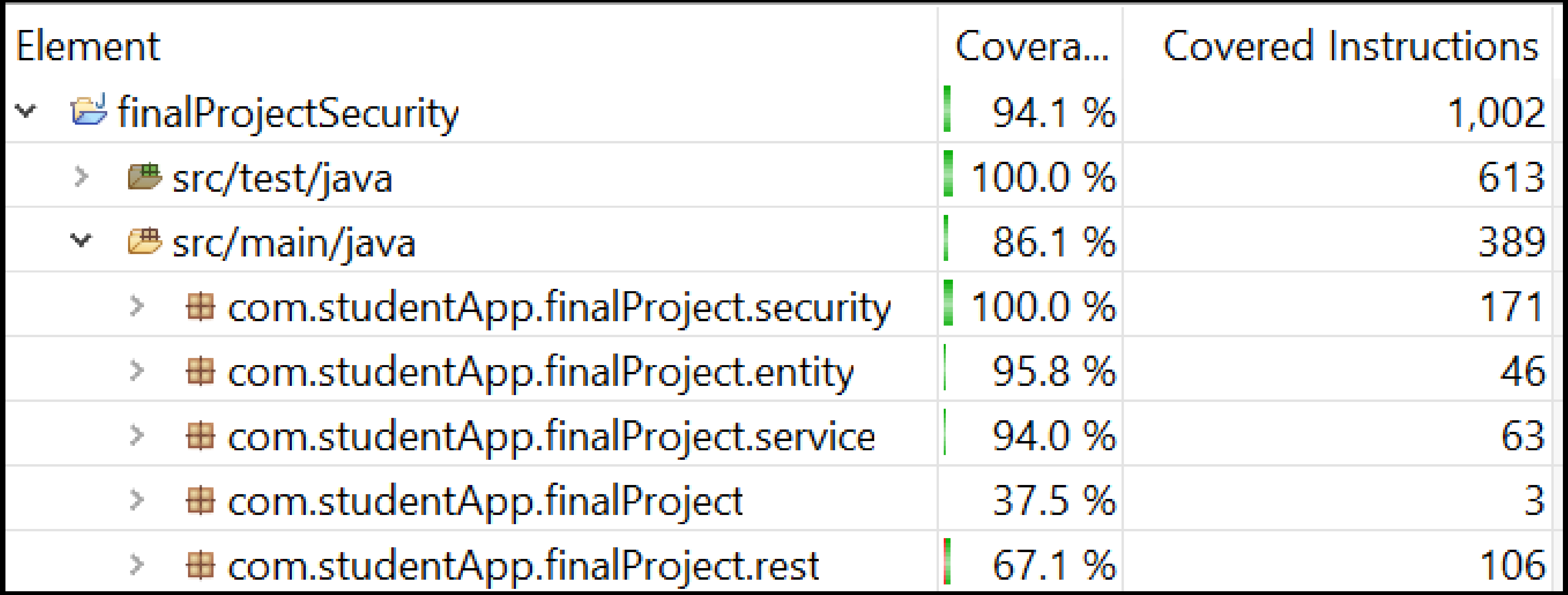


Runs: 18/18 Errors: 0 Failures: 0

```
FinalProjectApplicationTests [Runner: JUnit 5] (0.477 s)
    contextLoads() (0.477 s)
StudentControllerTest [Runner: JUnit 5] (0.679 s)
    testCreateStudent() (0.502 s)
    testGetStudentByIdNotFound() (0.028 s)
    testGetPassStudents() (0.089 s)
    testDeleteStudent() (0.006 s)
    testDeleteStudentNotFound() (0.006 s)
    testGetStudentAverageNotFound() (0.006 s)
    testGetStudentAverage() (0.008 s)
    testGetFailStudents() (0.006 s)
    testGetAllStudents() (0.006 s)
    testGetStudentById() (0.008 s)
StudentServiceTest [Runner: JUnit 5] (0.309 s)
    findPassStudents() (0.273 s)
    testSaveStudent() (0.005 s)
    findFailStudents() (0.006 s)
    testFindStudentById() (0.006 s)
    testDeleteStudentByIdNotFound() (0.006 s)
    testDeleteStudentById() (0.005 s)
    testGetAverageForStudent() (0.003 s)
```

Element	Covera...	Covered Instructions
finalProjectSpringData_JUnit	98.8 %	925
src/test/java	100.0 %	710
com.studentApp.finalProject	100.0 %	4
com.studentApp.finalProject.rest	100.0 %	498
com.studentApp.finalProject.service	100.0 %	208
src/main/java	95.1 %	215
com.studentApp.finalProject.rest	100.0 %	103
StudentController.java	100.0 %	103
com.studentApp.finalProject.entity	95.8 %	46
Student.java	95.8 %	46
com.studentApp.finalProject.service	94.0 %	63
StudentServiceImpl.java	94.0 %	63
com.studentApp.finalProject	37.5 %	3
FinalProjectApplication.java	37.5 %	3

Testing with security
94% coverage



Element	Covera...	Covered Instructions
finalProjectSecurity	94.1 %	1,002
src/test/java	100.0 %	613
src/main/java	86.1 %	389
com.studentApp.finalProject.security	100.0 %	171
com.studentApp.finalProject.entity	95.8 %	46
com.studentApp.finalProject.service	94.0 %	63
com.studentApp.finalProject	37.5 %	3
com.studentApp.finalProject.rest	67.1 %	106

Future steps

- -Change in-memory authentication to database authentication with encrypted password.

- -Add frontend menus for better visualization and custom endpoints depending on the request.

- -Migrate to cloud databases.

CONCLUSION

- Integrated RESTful APIs with MySQL for data persistence.
- Utilized Spring Security for authentication and authorization.
- Performed thorough unit and integration testing

✓ **Security**

Safety

✓ **Data JPa**

Database conection

✓ **MockMVC**

Testing