Leaners Academy

Problem Statement:

Learner's Academy is a school that has an online management system. The system keeps track of its classes, subjects, students, and teachers. It has a back-office application, The administrator can:

- Set up a master list of all the subjects for all the classes
- Set up a master list of all the teachers
- Set up a master list of all the classes
- Assign classes for subjects from the master list
- Assign teachers to a class for a subject (A teacher can be assigned to different classes for different subjects)
- Get a master list of students (Each student must be assigned to a single class)

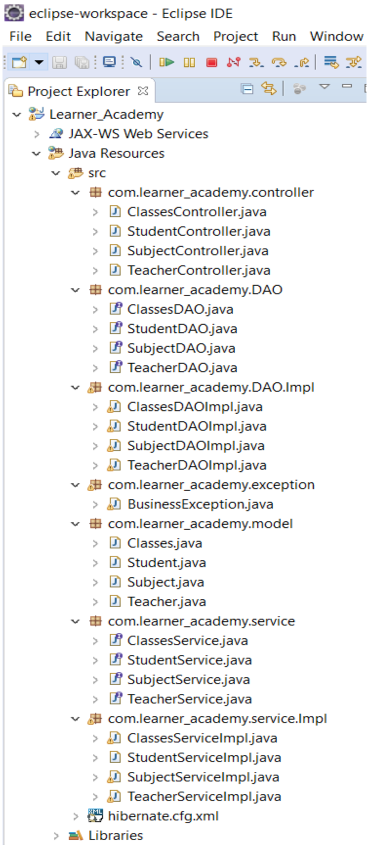
There will be an option to view a Class Report which will show all the information about the class, such as the list of students, subjects, and teachers

The goal of the company is to deliver a high-end quality product as early as possible.

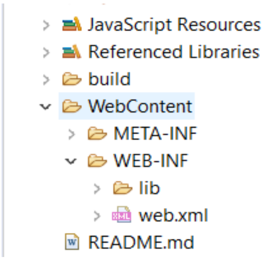
Tools used in the Project:

Programming Language and related	***Java, Hibernate, Jersey***
IDE	***Eclipse***
SQL	***MySQL***
Version Control System	***Git & GitHub***
Software Development Methodology	***Agile - Scrum - Jira***
API Client	***POSTMAN***

Hierarchy of codes developed:

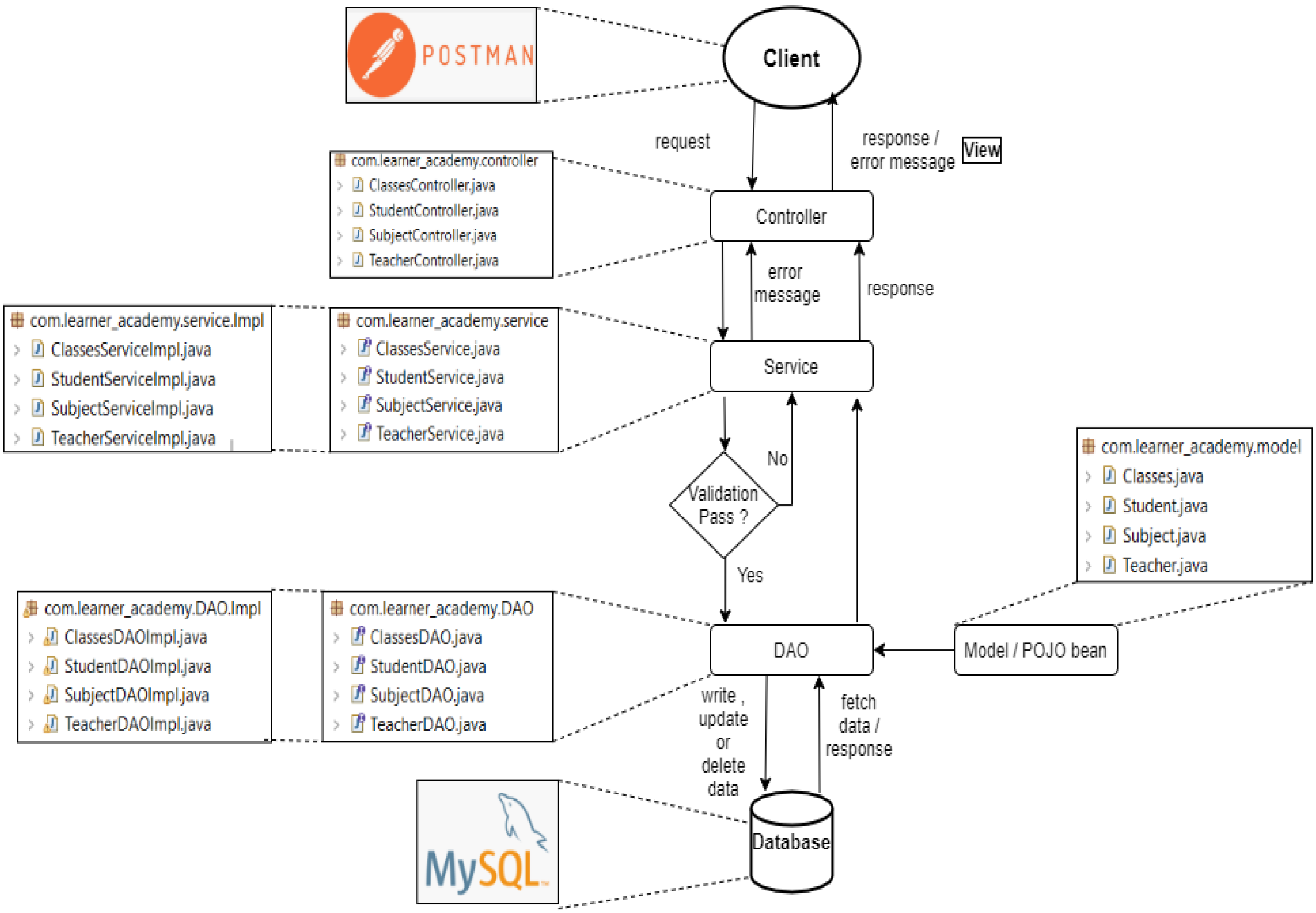


```
Learner_Academy
  JAX-WS Web Services
  Java Resources
    src
      com.learner_academy.controller
        ClassesController.java
        StudentController.java
        SubjectController.java
        TeacherController.java
      com.learner_academy.DAO
        ClassesDAO.java
        StudentDAO.java
        SubjectDAO.java
        TeacherDAO.java
      com.learner_academy.DAO.Impl
        ClassesDAOImpl.java
        StudentDAOImpl.java
        SubjectDAOImpl.java
        TeacherDAOImpl.java
      com.learner_academy.exception
        BusinessException.java
      com.learner_academy.model
        Classes.java
        Student.java
        Subject.java
        Teacher.java
      com.learner_academy.service
        ClassesService.java
        StudentService.java
        SubjectService.java
        TeacherService.java
      com.learner_academy.service.Impl
        ClassesServiceImpl.java
        StudentServiceImpl.java
        SubjectServiceImpl.java
        TeacherServiceImpl.java
      hibernate.cfg.xml
    Libraries
```



```
> JavaScript Resources
> Referenced Libraries
> build
v WebContent
  > META-INF
  v WEB-INF
    > lib
    > web.xml
  README.md
```

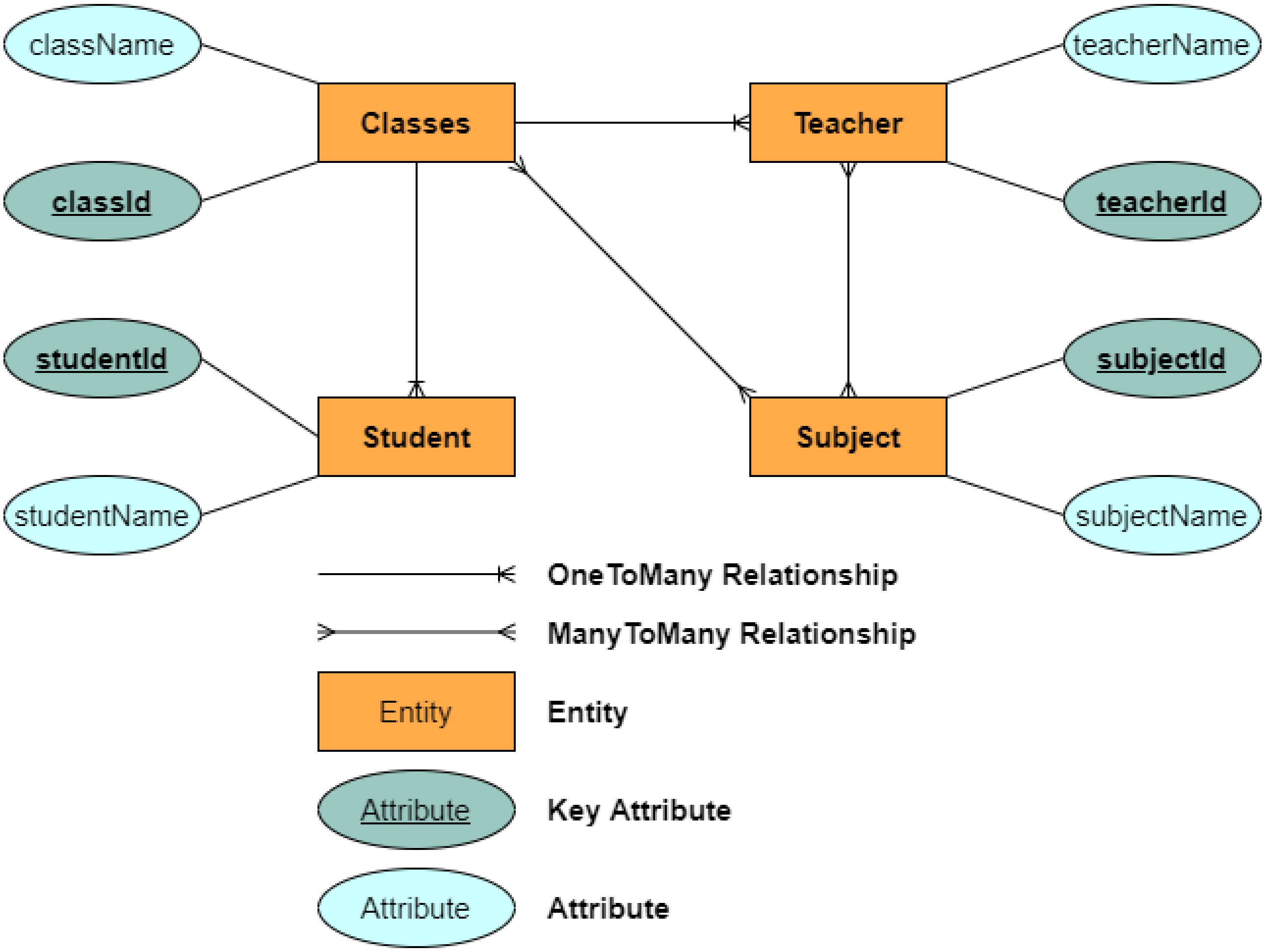
Application Architecture and Flow Chart:



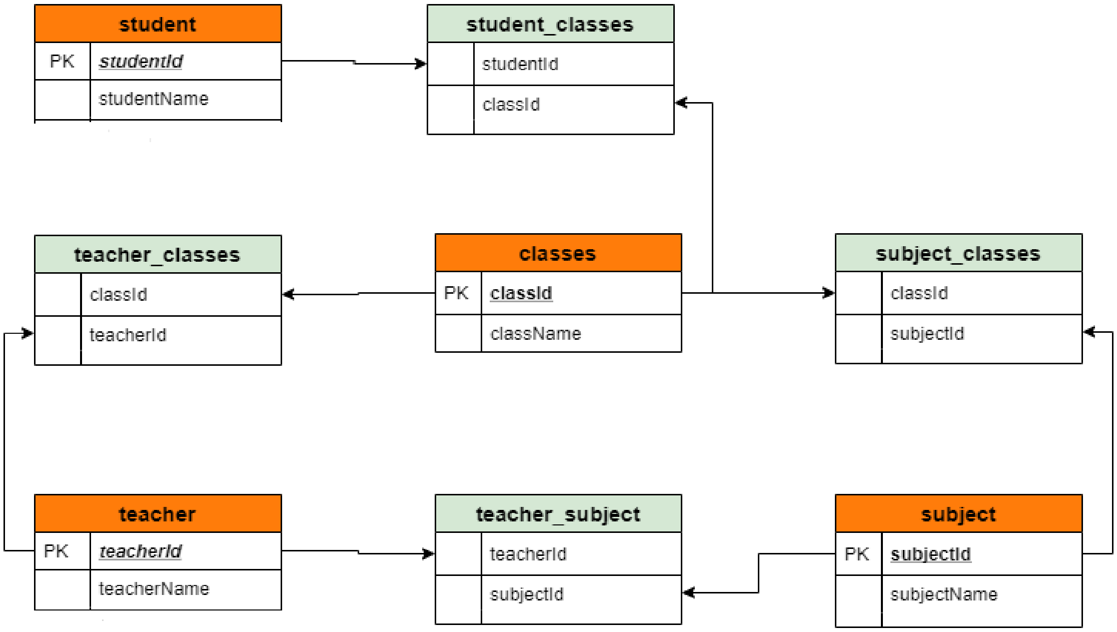
Exception Handling:

Some of the exceptions and errors are handled, so that the application won’t close, exit or throw an exception.

Entity Relationship Diagram



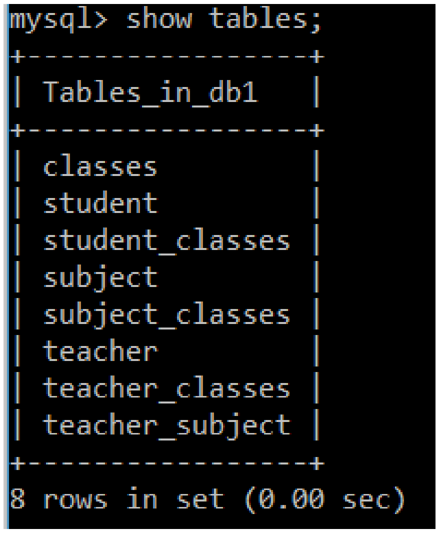
Entity and Tables:



Database Tables and Mapping

Legend	
🟧	*Entity & Table*
🟩	*Table*

Tables created in Database:



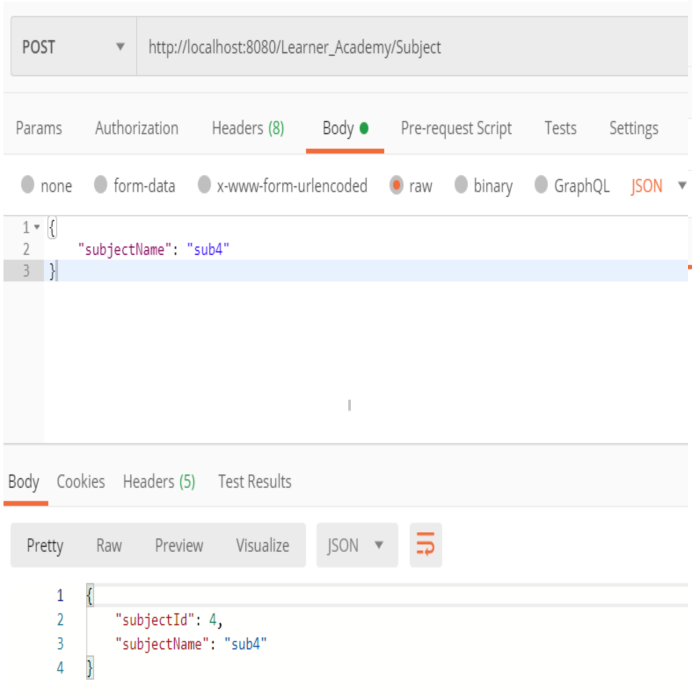
```
mysql> show tables;
+-----------------+
| Tables_in_db1   |
+-----------------+
| classes         |
| student         |
| student_classes |
| subject         |
| subject_classes |
| teacher         |
| teacher_classes |
| teacher_subject |
+-----------------+
8 rows in set (0.00 sec)
```

Project Requirement and Verification in POSTMAN Tool:

Project Requirement		Verifying in POSTMAN tool	
S.No.	**Requirement**	**HTTP Method**	**Request URL**
1	Set up a master list of all the subjects	GET	http://localhost:8080/Learner_Academy/Subject
2	Set up a master list of all the teachers	GET	http://localhost:8080/Learner_Academy/Teacher
3	Set up a master list of all the classes	GET	http://localhost:8080/Learner_Academy/Classes
4	Assign subject for class	GET	http://localhost:8080/Learner_Academy/Classes
5	Assign teacher to subject	GET	http://localhost:8080/Learner_Academy/Classes
6	Get a master list of students	GET	http://localhost:8080/Learner_Academy/Student
7	An option to view a Class Report which will show all the information about the class, such as the list of students, subjects, and teachers	GET	http://localhost:8080/Learner_Academy/Classes

Screenshots taken while executing project requirements and their verification:

1) ***Set up a master list of all the subjects:***

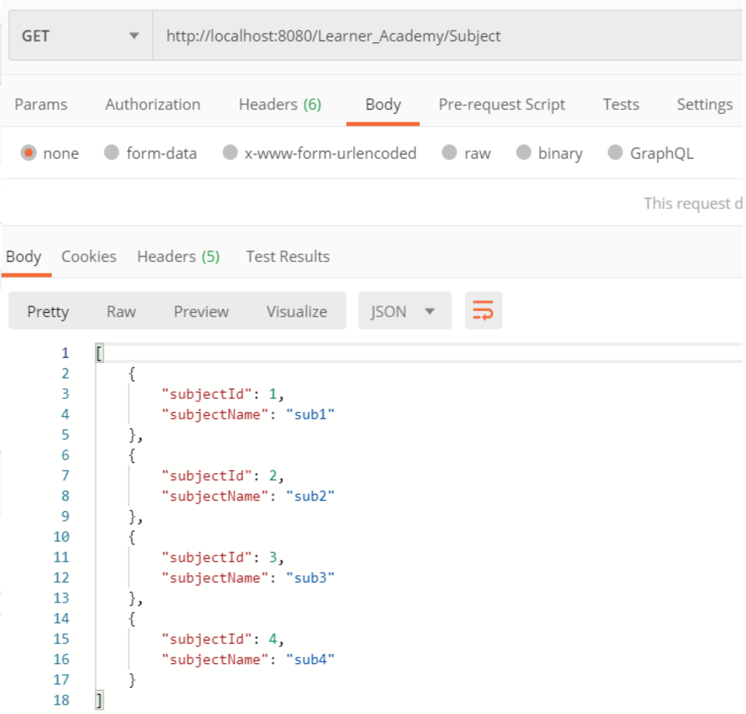


POST http://localhost:8080/Learner_Academy/Subject

```
{
    "subjectName": "sub4"
}
```

Response:

```
{
    "subjectId": 4,
    "subjectName": "sub4"
}
```

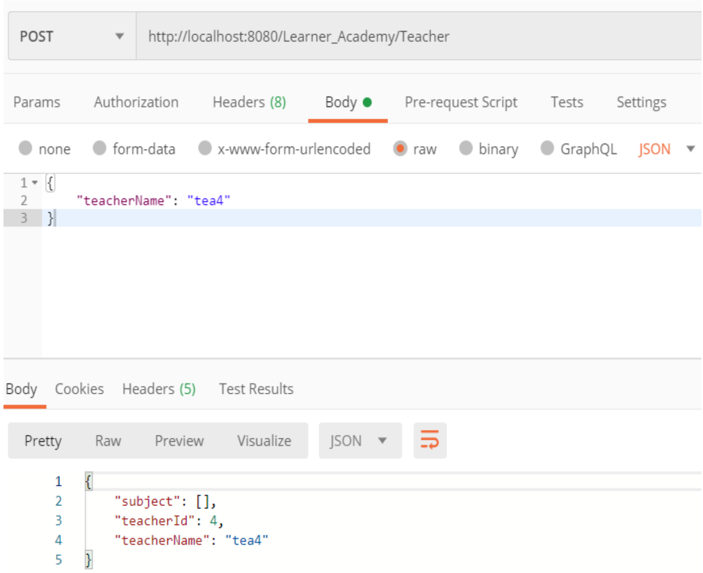


GET http://localhost:8080/Learner_Academy/Subject

Response:

```
[
    {
        "subjectId": 1,
        "subjectName": "sub1"
    },
    {
        "subjectId": 2,
        "subjectName": "sub2"
    },
    {
        "subjectId": 3,
        "subjectName": "sub3"
    },
    {
        "subjectId": 4,
        "subjectName": "sub4"
    }
]
```

2) ***Set up a master list of all the teachers:***

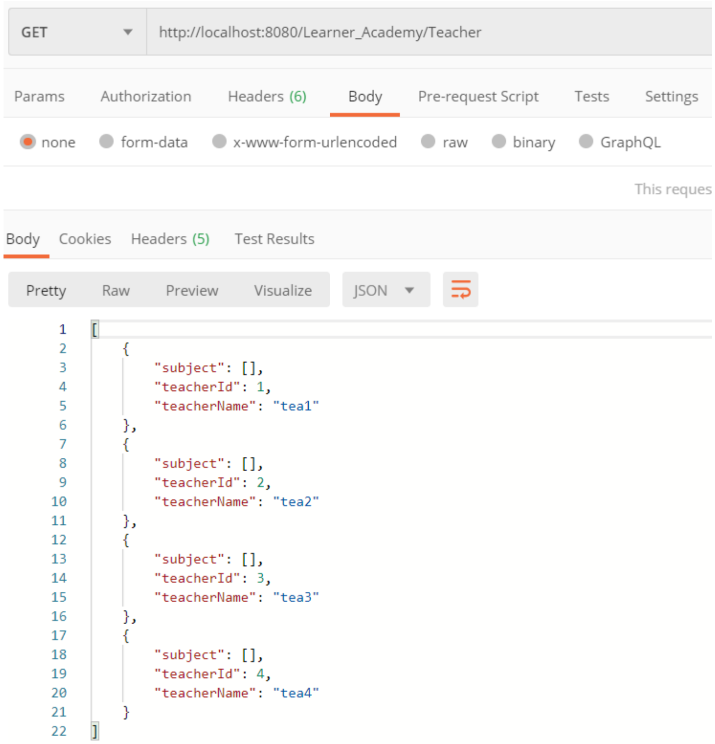


POST http://localhost:8080/Learner_Academy/Teacher

Params
Authorization
Headers (8)
Body ●
Pre-request Script
Tests
Settings
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON ▾

```
{
    "teacherName": "tea4"
}
```

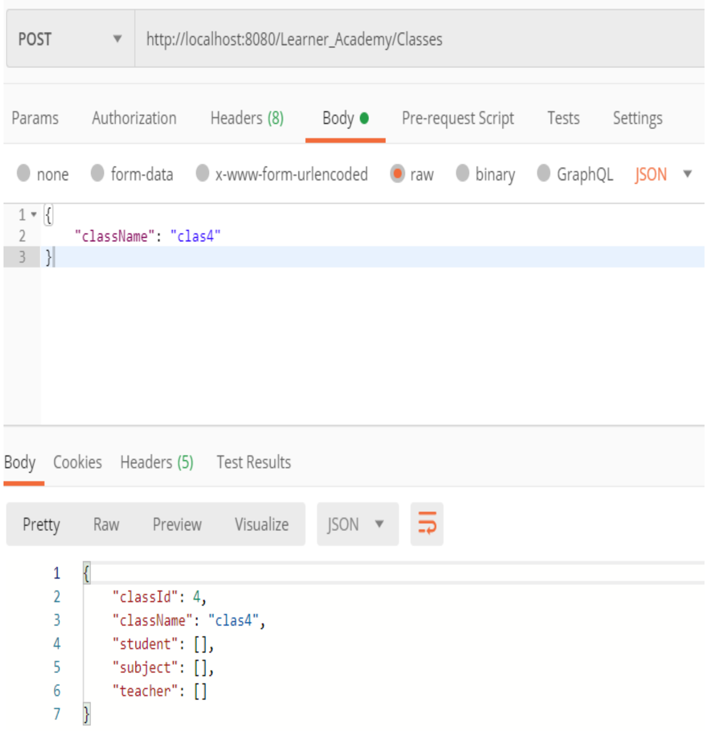
```
{
    "subject": [],
    "teacherId": 4,
    "teacherName": "tea4"
}
```



GET http://localhost:8080/Learner_Academy/Teacher

```
[
    {
        "subject": [],
        "teacherId": 1,
        "teacherName": "tea1"
    },
    {
        "subject": [],
        "teacherId": 2,
        "teacherName": "tea2"
    },
    {
        "subject": [],
        "teacherId": 3,
        "teacherName": "tea3"
    },
    {
        "subject": [],
        "teacherId": 4,
        "teacherName": "tea4"
    }
]
```

3) ***Set up a master list of all the classes:***

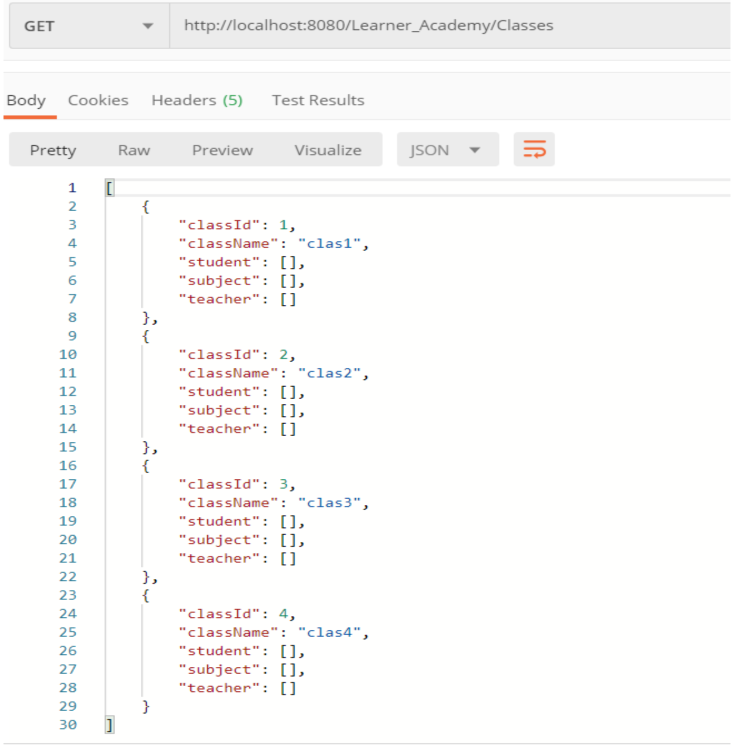


POST http://localhost:8080/Learner_Academy/Classes

Params
Authorization
Headers (8)
Body ●
Pre-request Script
Tests
Settings
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON ▾

```
{
    "className": "clas4"
}
```

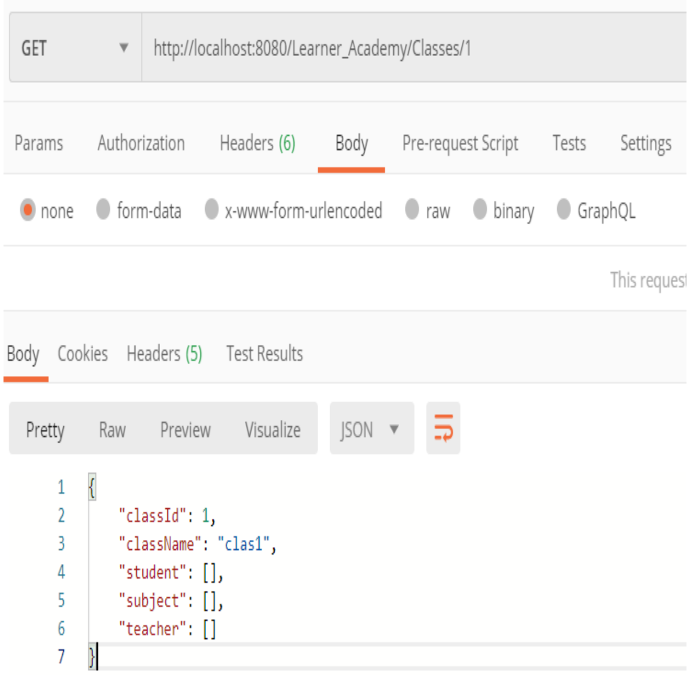
```
{
    "classId": 4,
    "className": "clas4",
    "student": [],
    "subject": [],
    "teacher": []
}
```



GET http://localhost:8080/Learner_Academy/Classes

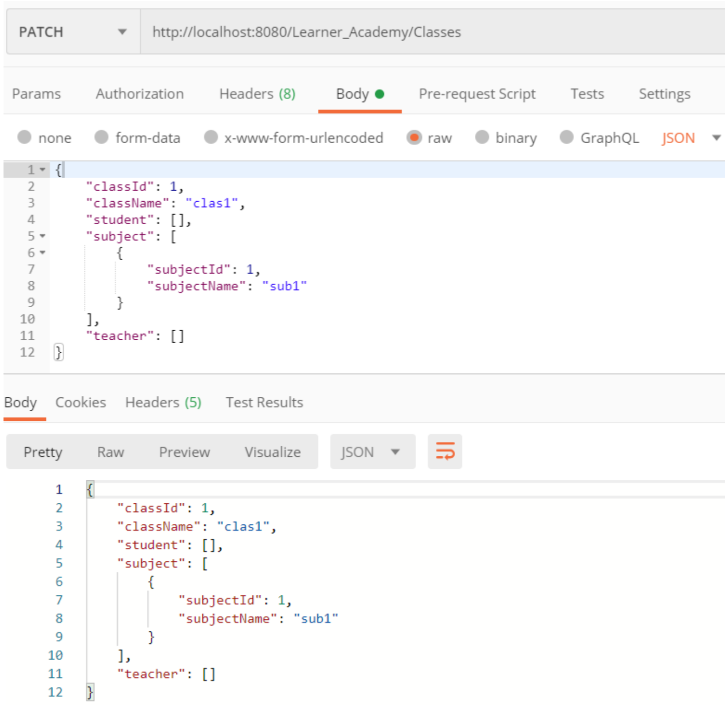
```
[
    {
        "classId": 1,
        "className": "clas1",
        "student": [],
        "subject": [],
        "teacher": []
    },
    {
        "classId": 2,
        "className": "clas2",
        "student": [],
        "subject": [],
        "teacher": []
    },
    {
        "classId": 3,
        "className": "clas3",
        "student": [],
        "subject": [],
        "teacher": []
    },
    {
        "classId": 4,
        "className": "clas4",
        "student": [],
        "subject": [],
        "teacher": []
    }
]
```

4) ***Assign subject for class:***



GET http://localhost:8080/Learner_Academy/Classes/1

```
{
    "classId": 1,
    "className": "clas1",
    "student": [],
    "subject": [],
    "teacher": []
}
```

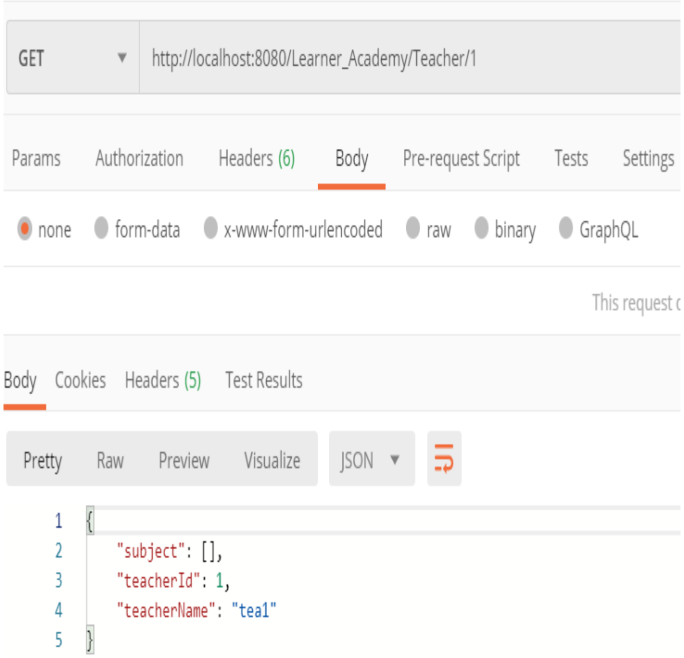


PATCH http://localhost:8080/Learner_Academy/Classes

```
{
    "classId": 1,
    "className": "clas1",
    "student": [],
    "subject": [
        {
            "subjectId": 1,
            "subjectName": "sub1"
        }
    ],
    "teacher": []
}
```

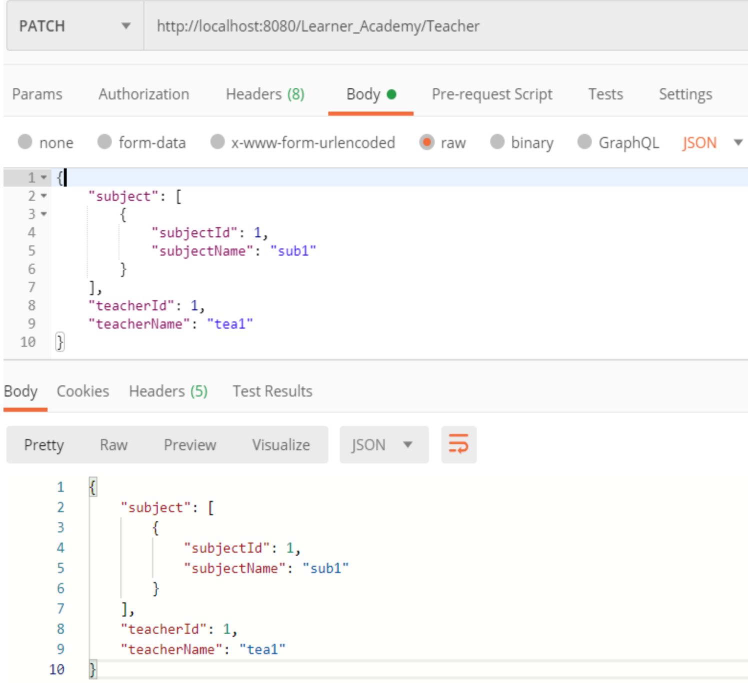
```
{
    "classId": 1,
    "className": "clas1",
    "student": [],
    "subject": [
        {
            "subjectId": 1,
            "subjectName": "sub1"
        }
    ],
    "teacher": []
}
```

5) ***Assign teacher to subject:***



GET http://localhost:8080/Learner_Academy/Teacher/1

```
{
    "subject": [],
    "teacherId": 1,
    "teacherName": "tea1"
}
```

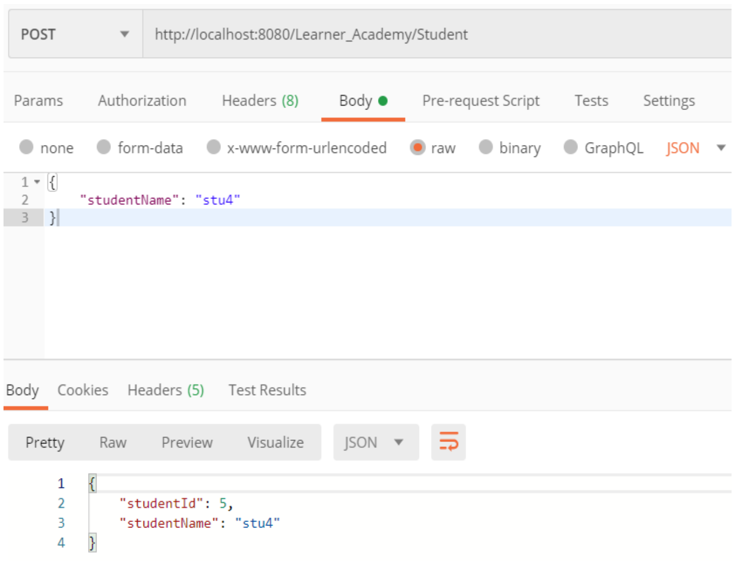


PATCH http://localhost:8080/Learner_Academy/Teacher

```
{
    "subject": [
        {
            "subjectId": 1,
            "subjectName": "sub1"
        }
    ],
    "teacherId": 1,
    "teacherName": "tea1"
}
```

```
{
    "subject": [
        {
            "subjectId": 1,
            "subjectName": "sub1"
        }
    ],
    "teacherId": 1,
    "teacherName": "tea1"
}
```

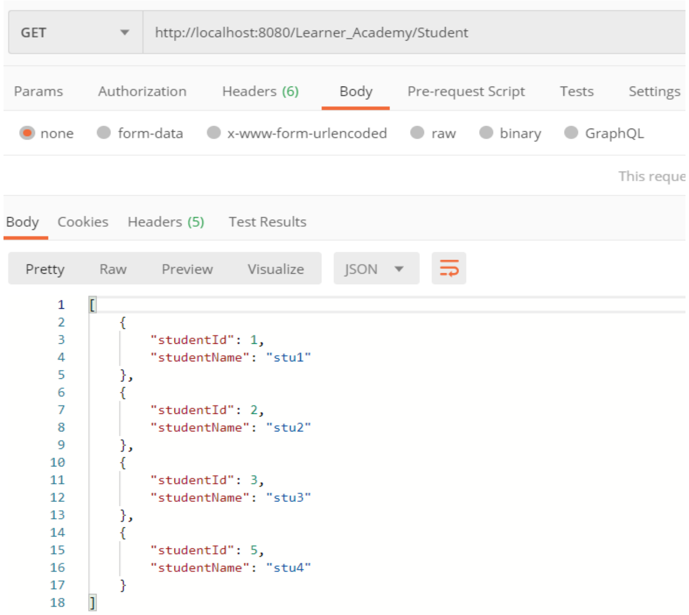
6) ***Get a master list of students***



POST http://localhost:8080/Learner_Academy/Student

```
{
    "studentName": "stu4"
}
```

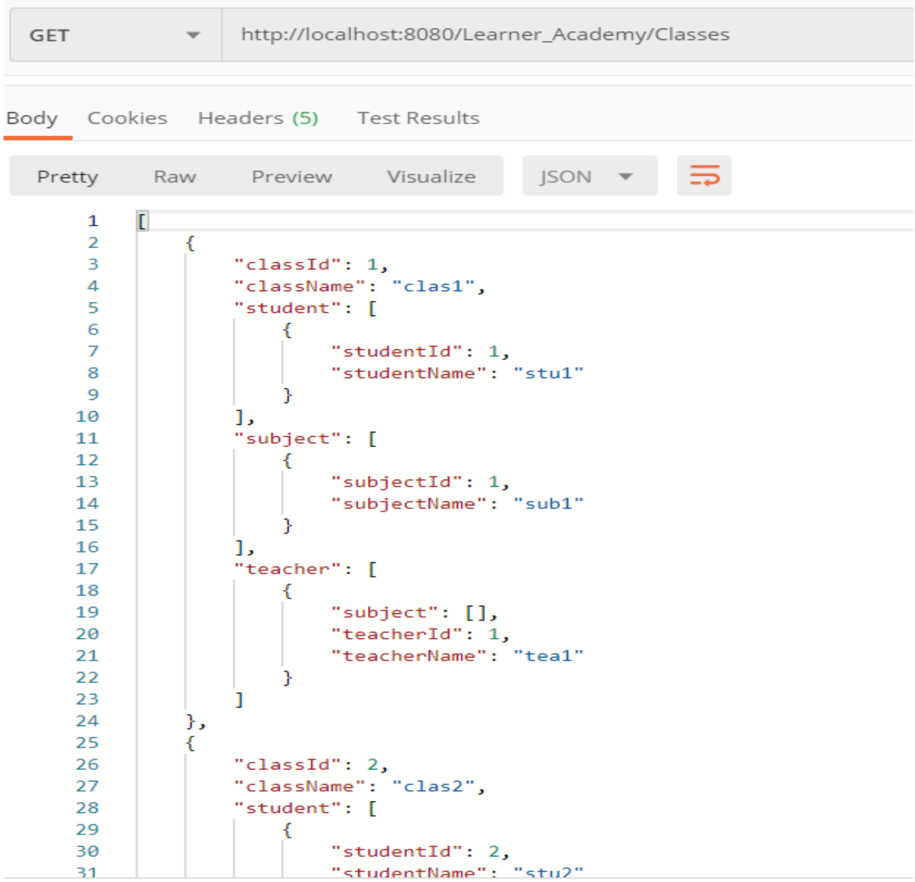
```
{
    "studentId": 5,
    "studentName": "stu4"
}
```



GET http://localhost:8080/Learner_Academy/Student

```
[
    {
        "studentId": 1,
        "studentName": "stu1"
    },
    {
        "studentId": 2,
        "studentName": "stu2"
    },
    {
        "studentId": 3,
        "studentName": "stu3"
    },
    {
        "studentId": 5,
        "studentName": "stu4"
    }
]
```

7) ***An option to view a Class Report which will show all the information about the class, such as the list of students, subjects, and teachers:***



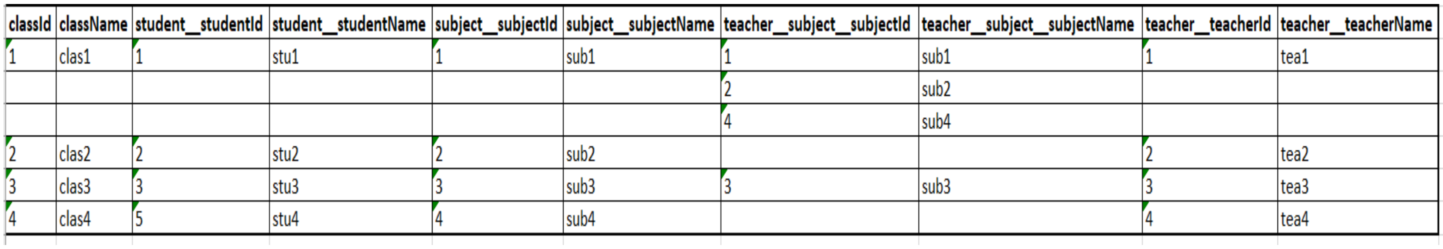
GET http://localhost:8080/Learner_Academy/Classes

```
[
    {
        "classId": 1,
        "className": "clas1",
        "student": [
            {
                "studentId": 1,
                "studentName": "stu1"
            }
        ],
        "subject": [
            {
                "subjectId": 1,
                "subjectName": "sub1"
            }
        ],
        "teacher": [
            {
                "subject": [],
                "teacherId": 1,
                "teacherName": "tea1"
            }
        ]
    },
    {
        "classId": 2,
        "className": "clas2",
        "student": [
            {
                "studentId": 2,
                "studentName": "stu2"
```

Since the response Body for above requirement is lengthier, the output is copied to notepad and embedded here (also uploaded to the project's GitHub Repository)

Class Report
response JSON.txt

Converted the above JSON response of Class Master Report to CSV for ease of viewing



classId	className	student__studentId	student__studentName	subject__subjectId	subject__subjectName	teacher__subject__subjectId	teacher__subject__subjectName	teacher__teacherId	teacher__teacherName
1	clas1	1	stu1	1	sub1	1	sub1	1	tea1
						2	sub2		
						4	sub4		
2	clas2	2	stu2	2	sub2			2	tea2
3	clas3	3	stu3	3	sub3	3	sub3	3	tea3
4	clas4	5	stu4	4	sub4			4	tea4

GITHUB URL:

Project file uploaded here

https://github.com/VIGNESH2803/Leaners-Academy