



**Learner's Academy**

*DELL FSD - Learner's Academy Project*

*By*

*Prasanth Chandrasekar*

## *TABLE OF CONTENTS*

<b>S.No</b>	<b>Content</b>	<b>Page Number</b>
<b>1</b>	<b>Problem Statement</b>	<b>3</b>
<b>2</b>	<b>Sprints and the User Stories</b>	<b>3</b>
<b>3</b>	<b>Jira User Story</b>	<b>5</b>
<b>4</b>	<b>Jira Sprint</b>	<b>5</b>
<b>5</b>	<b>Tools used in the project</b>	<b>6</b>
<b>6</b>	<b>Hierarchy of Codes</b>	<b>6</b>
<b>7</b>	<b>Application Architecture / Flow Chart</b>	<b>7</b>
<b>8</b>	<b>Exception Handling</b>	<b>8</b>
<b>9</b>	<b>Entity Relationship Diagram</b>	<b>8</b>
<b>10</b>	<b>Entity and Tables</b>	<b>9</b>
<b>11</b>	<b>Tables Created in Database</b>	<b>9</b>
<b>12</b>	<b>Project requirement and it's verification links in POSTMAN</b>	<b>10</b>
<b>13</b>	<b>Screenshots showing completion of Project requirement</b>	<b>10</b>
<b>14</b>	<b>Git Screenshots</b>	<b>14</b>
<b>15</b>	<b>GitHub Screenshots</b>	<b>15</b>
<b>16</b>	<b>GitHub Repository Link for source Code and other Project related Files</b>	<b>16</b>

### **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.

### **Sprints and the User Stories:**

#### **Sprint-1: (One week)**

- **User Story-1 :** As an Administrator, I want to have an option to create Students, so that i can set up a master list of Students
- **User Story-2 :** As an Administrator, I want to have an option to create Subjects, so that i can set up a master list of Subjects
- **User Story-3 :** As an Administrator, I want to have an option to create Classes, so that i can set up a master list of Classes
- **User Story-4 :** As an Administrator, I want to have an option to create Teachers, so that i can set up a master list of Teachers
- **User Story-5 :** As an Administrator, I want to have an option to view Students, so that i can view a Student by Id and also a list of Students.
- **User Story-6 :** As an Administrator, I want to have an option to view Subjects, so that i can view a Subject by Id and also a list of Subjects.
- **User Story-7 :** As an Administrator, I want to have an option to view Classes, so that i can view a Class by Id and also a list of Classes.
- **User Story-8 :** As an Administrator, I want to have an option to view Teachers, so that i can view a Teacher by Id and also a list of Teacher.
- **User Story-9 :** As an Administrator, I want to have an option to update Students, so that i can modify a Student by Id.



- **User Story-10** : As an Administrator, I want to have an option to update Subjects, so that i can modify a Subject by Id.
- **User Story-11** : As an Administrator, I want to have an option to update Classes, so that i can modify a Class by Id.
- **User Story-12** : As an Administrator, I want to have an option to update Teachers, so that i can modify a Teacher by Id.

### **Sprint-2:(one week)**

- **User Story-1** : As an Administrator, I want to have an option to delete Students, so that i can remove Students whenever they are off-boarded.
- **User Story-2** : As an Administrator, I want to have an option to delete Subjects, so that i can remove Subjects whenever they are not required.
- **User Story-3** : As an Administrator, I want to have an option to delete Classes, so that i can remove Classes whenever they are not required.
- **User Story-4** : As an Administrator, I want to have an option to delete Teachers, so that i can remove Teachers whenever they are off-boarded.
- **User Story-5** : As an Administrator, I want to have an option to assign Students to Classes, so that i can assign Students to Classes whenever the students are promoted or on-boarded.
- **User Story-6** : As an Administrator, I want to have an option to assign Subjects to Classes, so that i can have subjects assigned to Class.
- **User Story-7** : As an Administrator, I want to have an option to assign Subjects to Teachers, so that i can assign Teachers to Subjects.
- **User Story-8** : As an Administrator, I want to have an option to assign Teachers to Classes, so that i can assign Teachers to Classes.:

## DELL FSD PROJECT - Learner's Academy – Prasanth Chandrasekar

### Jira User Story – Agile Scrum:

✎ Add epic / 🟢 DFLA-13

👁 1 🔗 ⋮ ✕

As an Administrator, I want to have an option to delete Students, so that i can remove Students whenever they are off-boarded.

📎 Attach 🧩 Add a child issue 🔗 Link issue ⌵ ⋮

#### Description

Add a description...

#### Activity

Show: **Comments** History



Add a comment...

Pro tip: press **M** to comment

To Do ▾

IN PROGRESS

DONE

Labels

None

Sprint

DFLA Sprint 2

Story point estimate

None

Reporter



Prasanth Chandrasekar

### Jira Sprint– Agile Scrum:

**Jira Software** Your work Projects ▾ Filters ▾ Dashboards ▾ People Apps ▾ **Create**

DELL FSD Learner's...  
Next-gen software project

- Roadmap
- Backlog
- Board**
- Pages
- Add item
- Project settings

You're in a next-gen project  
Give feedback Learn more

Projects / DELL FSD Learner's Academy

### DFLA Sprint 2

to fulfill the requirement from the administrator

TO DO 7

As an Administrator, I want to have an option to delete Classes, so that i can remove Classes whenever they are not required.

🟢 DFLA-15

As an Administrator, I want to have an option to delete Teachers, so that i can remove Teachers whenever they are off-boarded.

🟢 DFLA-16

As an Administrator, I want to have an option to assign Students to Classes, so that i can assign Students to Classes whenever the students are promoted or on-boarded.

🟢 DFLA-17

As an Administrator, I want to have an option to assign Subjects to Classes, so that i can have subjects assigned to Class.

🟢 DFLA-18

As an Administrator, I want to have an option to assign Subjects to Teachers, so that i can assign Teachers to Subjects.

🟢 DFLA-19

IN PROGRESS

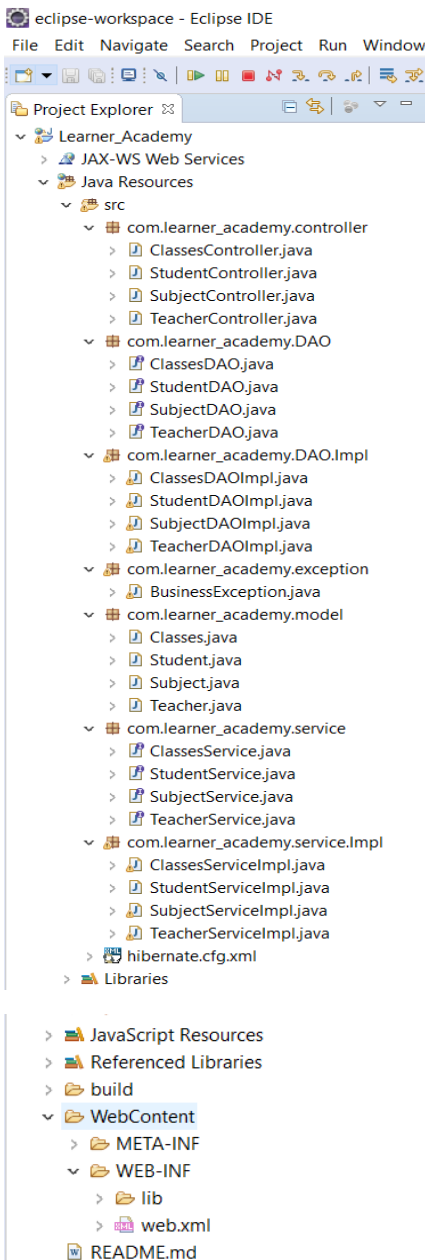
DONE 1 ✓



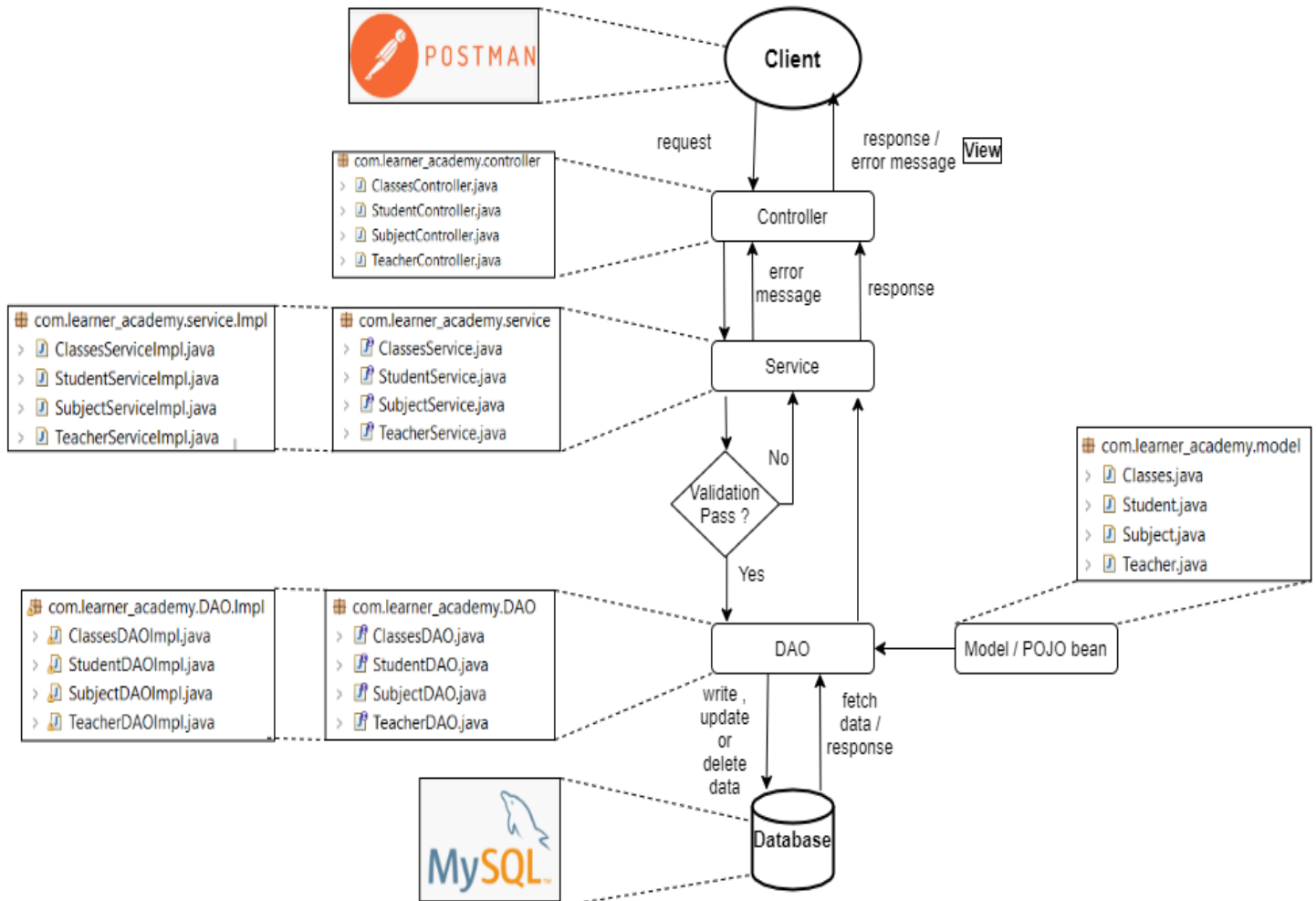
## Tools used in the Project:

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

## Hierarchy of codes developed:



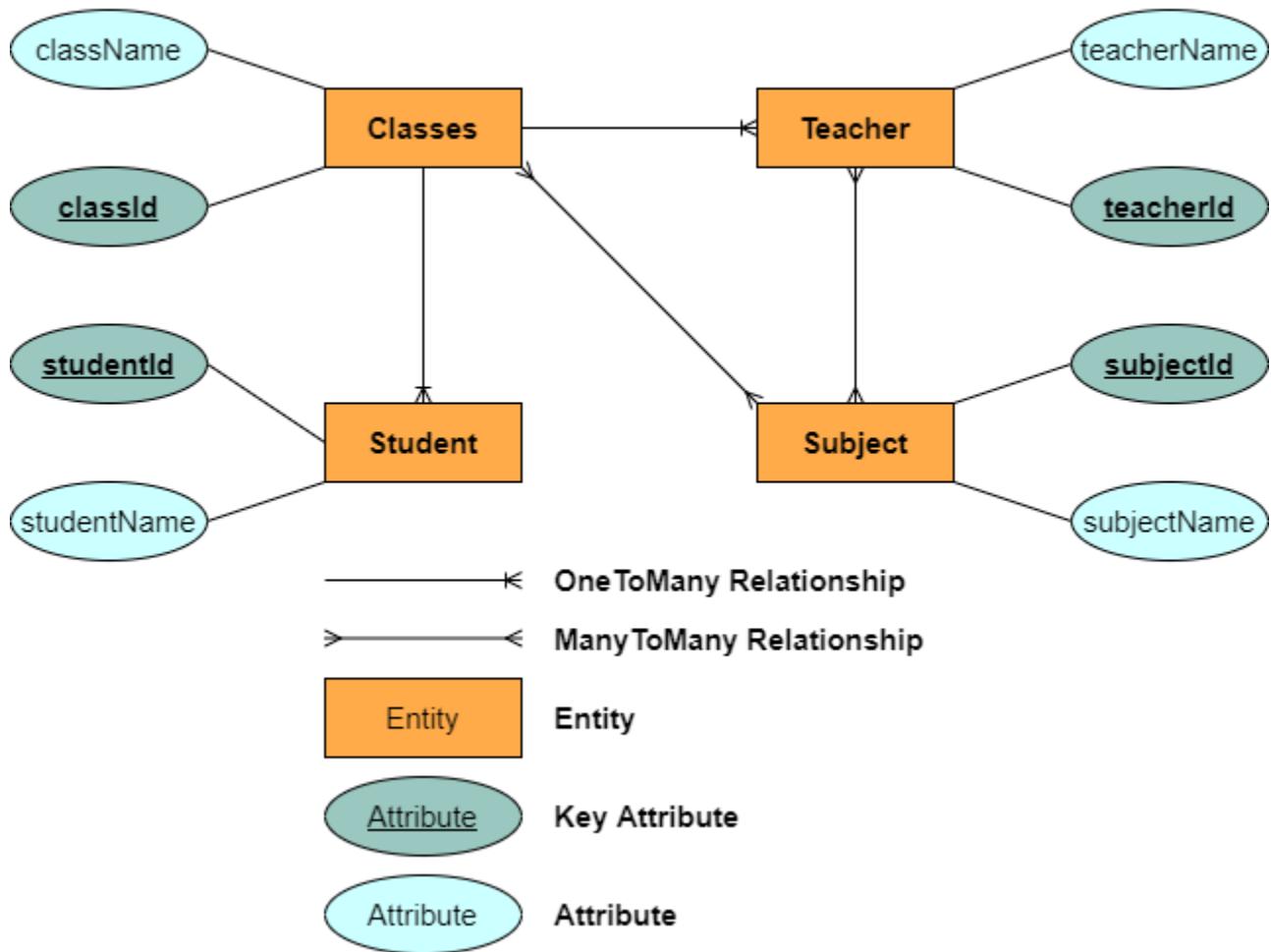
## Application Architecture and Flow Chart:



## Exception Handling:

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

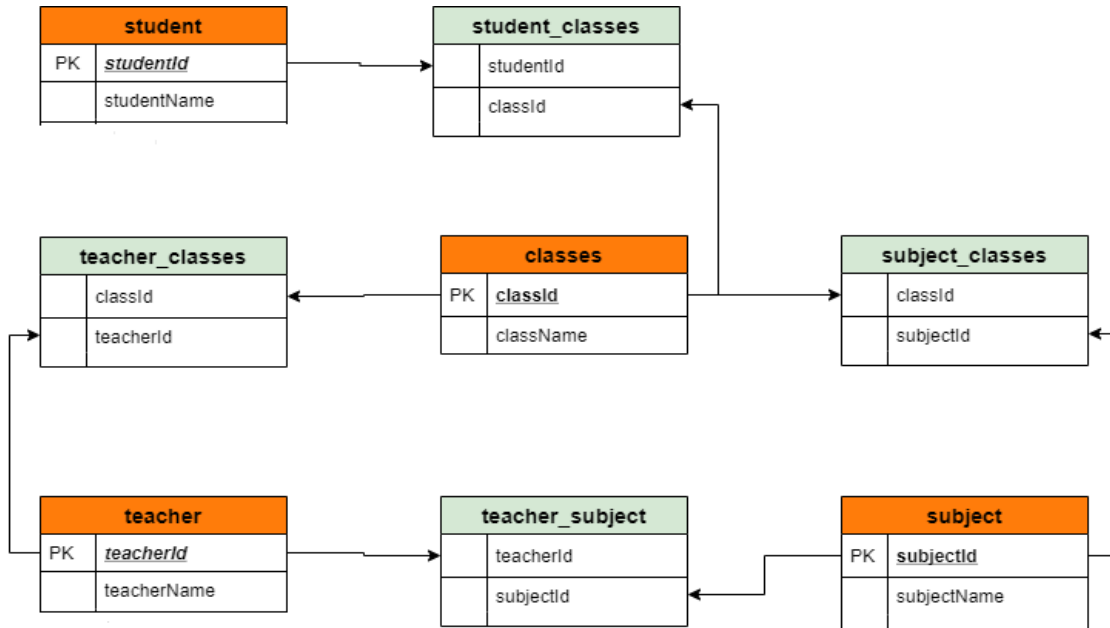
### Entity Relationship Diagram









## 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	<a href="http://localhost:8080/Learner_Academy/Subject">http://localhost:8080/Learner_Academy/Subject</a>
2	Set up a master list of all the teachers	GET	<a href="http://localhost:8080/Learner_Academy/Teacher">http://localhost:8080/Learner_Academy/Teacher</a>
3	Set up a master list of all the classes	GET	<a href="http://localhost:8080/Learner_Academy/Classes">http://localhost:8080/Learner_Academy/Classes</a>
4	Assign subject for class	GET	<a href="http://localhost:8080/Learner_Academy/Classes">http://localhost:8080/Learner_Academy/Classes</a>
5	Assign teacher to subject	GET	<a href="http://localhost:8080/Learner_Academy/Classes">http://localhost:8080/Learner_Academy/Classes</a>
6	Get a master list of students	GET	<a href="http://localhost:8080/Learner_Academy/Student">http://localhost:8080/Learner_Academy/Student</a>
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	<a href="http://localhost:8080/Learner_Academy/Classes">http://localhost:8080/Learner_Academy/Classes</a>

### Screenshots taken while executing project requirements and their verification:

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

The image displays two screenshots of the Postman application. The left screenshot shows the 'Body' tab of a POST request to `http://localhost:8080/Learner_Academy/Subject`. The request body is a JSON array with one object: `{ "subjectName": "sub4" }`. The right screenshot shows the 'Body' tab of a GET request to the same URL. The response is a JSON array of four objects, each representing a subject with a unique ID and name: `[ { "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 ▼

```

1 {
2   "teacherName": "tea4"
3 }

```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```

1 {
2   "subject": [],
3   "teacherId": 4,
4   "teacherName": "tea4"
5 }

```


GET ▼ http://localhost:8080/Learner\_Academy/Teacher

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```

1 [
2   {
3     "subject": [],
4     "teacherId": 1,
5     "teacherName": "tea1"
6   },
7   {
8     "subject": [],
9     "teacherId": 2,
10    "teacherName": "tea2"
11  },
12  {
13    "subject": [],
14    "teacherId": 3,
15    "teacherName": "tea3"
16  },
17  {
18    "subject": [],
19    "teacherId": 4,
20    "teacherName": "tea4"
21  }
22 ]

```

## 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 ▼

```

1 {
2   "className": "clas4"
3 }

```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 


```

1 {
2   "classId": 4,
3   "className": "clas4",
4   "student": [],
5   "subject": [],
6   "teacher": []
7 }

```

GET ▼ http://localhost:8080/Learner\_Academy/Classes

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```

1 [
2   {
3     "classId": 1,
4     "className": "clas1",
5     "student": [],
6     "subject": [],
7     "teacher": []
8   },
9   {
10    "classId": 2,
11    "className": "clas2",
12    "student": [],
13    "subject": [],
14    "teacher": []
15  },
16  {
17    "classId": 3,
18    "className": "clas3",
19    "student": [],
20    "subject": [],
21    "teacher": []
22  },
23  {
24    "classId": 4,
25    "className": "clas4",
26    "student": [],
27    "subject": [],
28    "teacher": []
29  }
30 ]

```

#### 4) Assign subject for class:

GET ▼ http://localhost:8080/Learner\_Academy/Classes/1

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ≡

```

1 {
2   "classId": 1,
3   "className": "clas1",
4   "student": [],
5   "subject": [],
6   "teacher": []
7 }
```

PATCH ▼ 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** ▼

```

1 {
2   "classId": 1,
3   "className": "clas1",
4   "student": [],
5   "subject": [
6     {
7       "subjectId": 1,
8       "subjectName": "sub1"
9     }
10  ],
11  "teacher": []
12 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ≡

```

1 {
2   "classId": 1,
3   "className": "clas1",
4   "student": [],
5   "subject": [
6     {
7       "subjectId": 1,
8       "subjectName": "sub1"
9     }
10  ],
11  "teacher": []
12 }
```

#### 5) Assign teacher to subject:

GET ▼ http://localhost:8080/Learner\_Academy/Teacher/1

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ≡

```

1 {
2   "subject": [],
3   "teacherId": 1,
4   "teacherName": "teal"
5 }
```

PATCH ▼ 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** ▼

```

1 {
2   "subject": [
3     {
4       "subjectId": 1,
5       "subjectName": "sub1"
6     }
7   ],
8   "teacherId": 1,
9   "teacherName": "teal"
10 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ≡

```

1 {
2   "subject": [
3     {
4       "subjectId": 1,
5       "subjectName": "sub1"
6     }
7   ],
8   "teacherId": 1,
9   "teacherName": "teal"
10 }
```

## 6) Get a master list of students

POST
http://localhost:8080/Learner\_Academy/Student

Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings

none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON

```

1 {
2   "studentName": "stu4"
3 }

```

Body
Cookies
Headers (5)
Test Results

Pretty
Raw
Preview
Visualize
JSON

```

1 {
2   "studentId": 5,
3   "studentName": "stu4"
4 }

```

GET
http://localhost:8080/Learner\_Academy/Student

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings

none
form-data
x-www-form-urlencoded
raw
binary
GraphQL

This request

Body
Cookies
Headers (5)
Test Results

Pretty
Raw
Preview
Visualize
JSON

```

1 {
2   {
3     "studentId": 1,
4     "studentName": "stu1"
5   },
6   {
7     "studentId": 2,
8     "studentName": "stu2"
9   },
10  {
11    "studentId": 3,
12    "studentName": "stu3"
13  },
14  {
15    "studentId": 5,
16    "studentName": "stu4"
17  }
18 }

```

## 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

Body
Cookies
Headers (5)
Test Results

Pretty
Raw
Preview
Visualize
JSON

```

1 {
2   {
3     "classId": 1,
4     "className": "clas1",
5     "student": [
6       {
7         "studentId": 1,
8         "studentName": "stu1"
9       }
10    ],
11    "subject": [
12      {
13        "subjectId": 1,
14        "subjectName": "sub1"
15      }
16    ],
17    "teacher": [
18      {
19        "subject": [],
20        "teacherId": 1,
21        "teacherName": "tea1"
22      }
23    ]
24  },
25  {
26    "classId": 2,
27    "className": "clas2",
28    "student": [
29      {
30        "studentId": 2,
31        "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

## Git Screenshots:

```
chandp23@INGSCHANDP23L1C MINGW64 /c:/users/chandp23/eclipse-workspace/Learner_Academy (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

```
chandp23@INGSCHANDP23L1C MINGW64 /c:/users/chandp23/eclipse-workspace/Learner_Academy (master)
$ git log --oneline
fed62ef (HEAD -> master, origin/master) sets nullable to false for attributes in all entities
e8230c3 handles exceptions for all four getById methods
248b89c handles exceptions for getStudentById and getClassesById methods
5f5d2b0 handles exceptions for getStudentById method
5f328c2 adds @Override annotation to all the methods of classes implemented from interface
f051874 changes classes subject relationship as ManyToMany
ac2bfd8 changes teacher subject relationship as ManyToMany
3cebb29 all mapping and CRUD works - need to work on validation and exception handling
d9dab4c throws constraint violation exception while assigning subject to classes (patching) subject and classes, subject classes mapping needs to be checked
82a4e6c removes 5 unwanted commented old files
3111dc0 adds teacher_subject table and mapping - class report needs to be created
a3b1aa5 all CRUD and mappings working - need to work class report
4f086e6 update on 491f5a9 - creates teacher, subject, teacher_subject tables and mapping setup between teacher and subject
491f5a9 changes the jointable from classes_student to student_classes
450fe4c creates student,classes,classes_student, subject and teacher tables - CRUD working - classes/student mapping done - fixes lazy loading exception in get method - changed fetch type to eager for student list
54b8e7f creates student,classes,classes_student, subject and teacher tables - CRUD working - classes/student mapping done
003fe00 creates student,classes,classes_student, subject tables - CRUD working - need to work on adding teacher
b3ea499 creates student,classes and classes_student tables CRUD working - need to check on JSON format on body
034431b after zoom session - no CRUD works
8da64e8 all CRUD working - mapping not done
5477533 throws mapping error
657459b removes constraints and unwanted columns in student and classes tables
547baeb all CRUD methods working for student and classes - mapping not done
22bd958 working on getById method
4a88e31 working - student and classes - getAll,update and create - without mapping
9e9e1b1 align subject class
8879a8e merge origin master
6b5f9a8 remove .classpath file
089dedc remove .project file
1c71db5 removed mapping for student and classes
a85bd32 removed mapping for student and classes
70e198f added .classpath and .project files
13c5cad second commit
dee3def first commit
```



## GitHub Screenshot:

PrasanthChand / LearnerAcademy

Unwatch 1 Star 0 Fork 0

<> Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

Dell FSD Learner's Academy Project 

Edit

Manage topics

34 commits

2 branches

0 packages

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

Prasanth sets nullable to false for attributes in all entities

Latest commit fed62ef 2 minutes ago

WebContent	second commit	7 days ago
build/classes	sets nullable to false for attributes in all entities	2 minutes ago
src	sets nullable to false for attributes in all entities	2 minutes ago
README.md	first commit	7 days ago

README.md

LearnerAcademy

## GitHub Repository:

All the project related files are uploaded to the below GitHub repository.

**Note:** The data in the database might be different from the screenshots, as few data were modified for testing purpose after taking screenshot. This project specification document is uploaded to below GitHub Repository in both .doc and .pdf extensions.

<https://github.com/PrasanthChand/LearnerAcademy>