

EduTrack — Classroom & Student Management Portal

Technology Stack

- **Backend:** Java, Spring Boot
 - **Database:** MongoDB (NoSQL)
 - **Frontend / Client:** Java-based client application
-

Project Overview

The **EduTrack Portal** is a full-stack Java-based web + client application designed to manage and streamline academic activities.

The portal provides **CRUD operations for managing batches** along with detailed student records within each batch.

It leverages **Spring Boot** for building a RESTful backend, **MongoDB** for scalable data storage, and a **Java client application** for user interaction.

The system simplifies batch creation, student management, and record updates, making it a centralized classroom management solution.

Key Features

Batch Management

- Create, update, delete, and view batches.

- Store batch details such as batch name, schedule, duration, and trainer information.

Student Records

- Maintain student details (name, contact, enrollment date, assigned batch).
- Link students dynamically with their respective batch.

CRUD Operations

- REST APIs for **Create, Read, Update, Delete** functionalities (Spring Boot).

Database Integration

- Uses **MongoDB** for flexible and scalable data storage.

Client-Side Application

- Java-based client with a user-friendly menu to interact with backend APIs.

Scalable & Modular Design

- Built on Spring Boot architecture, allowing easy integration with future features (attendance, performance tracking, fee management).

Learning Outcomes

- Hands-on experience with the **Spring Boot** framework for backend development.
 - Practical understanding of **MongoDB** and NoSQL data modeling.
 - Exposure to **REST API design** and client-server architecture.
 - Strong application of **full-stack Java** development.
 - Experience in **modular project design** suitable for real-world systems.
-

GitHub Repository

<https://github.com/Yash-Pagar1/EduTrack-Portal>

Application Usage

Client Application Menu

===== EduTrack Portal =====

1. Create New Batch
2. View All Batches
3. Update Batch Information
4. Delete Batch
5. Add Student to Batch
6. View Students in a Batch
7. Exit

Enter Choice: 1

Sample REST API (Spring Boot) -

//Example - Create Batch endpoint

```
@PostMapping("/batch")
```

```
public Batch createBatch(@RequestBody Batch batch) {
```

```
    return batchRepository.save(batch);
```

```
}
```

MongoDB Sample Document

```
{
```

```
"batchId": "101",  
"batchName": "Pre-Placement Activity",  
"trainer": "Piyush Khairnar",  
"students": [  
  { "name": "Amit", "contact": "9876543210" },  
  { "name": "Sneha", "contact": "9123456789" }  
]  
}
```