

**CHANDIGARH**  
**UNIVERSITY**

---

Discover. Learn. Empower.

## **UNIVERSITY INSTITUTE OF ENGINEERING**

### **PROJECT REPORT**

#### **Student Academic Portal**

**Subject Name – Project Based Learning in Java**

**Subject Code – 23CSH-304**

**Submitted To:**

**Er. Deep Prakash Gupta  
(E18557)**

**Submitted By:**

**Name: Sanampreet Singh  
UID: 23BCS13053  
Section: KRG\_2B**

# Student Academic Portal

---

## 1. Introduction

The **Student Academic Portal** is a full-stack web application designed to simplify communication and data management between students and administrators.

It allows students to **register, view announcements, timetables, and results**, while administrators can **manage student approvals, classes, announcements, timetables, and results**.

This project is built using **Spring Boot (Java)** for the backend, **React with Tailwind CSS** for the frontend, and **MongoDB** for data storage.

Additionally, it uses **Spring Mail** to send OTPs for secure user verification during registration.

## 2. Objectives

- To create a centralized portal for students and administrators.
- To automate student registration, class management, and result handling.
- To improve transparency and efficiency in sharing academic updates.
- To ensure secure access using OTP-based email verification.
- To practice full-stack web development using modern technologies.

## 3. System Overview

### Modules:

The system is divided into two major modules:

#### User Module

- User Registration with OTP verification
- Login using UID and password
- Profile view with personal details
- View announcements and timetable
- View semester and regular results

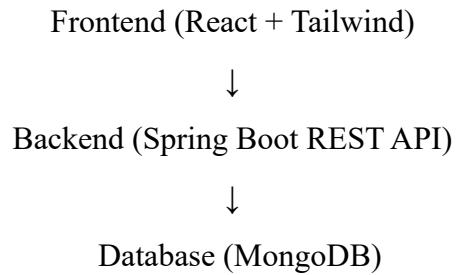
## **Admin Module**

- Approve or reject student registrations
- Create announcements for all users
- Manage timetables for each class
- Create classes and subjects with graded/non-graded options
- Upload internal and external marks for regular and semester results

## **4. Technology Used**

<b>Layer</b>	<b>Technology</b>
<b>Frontend</b>	React.js, Axios, Tailwind CSS
<b>Backend</b>	Spring Boot (Java)
<b>Database</b>	MongoDB
<b>Email Service</b>	Spring Mail
<b>State Management</b>	Redux Toolkit
<b>Security</b>	Spring Security
<b>Build Tools</b>	Maven (for backend), npm (for frontend)

## 5. Data Architecture



- Frontend sends HTTP requests to backend REST APIs.
- Backend handles business logic and interacts with the database.
- MongoDB stores user, class, timetable, and result data.

## 6. Project Structure

### Backend (Spring Boot)

```
src/main/java/com/example/backend  
    |  
    +-- config/  
    |    +-- MongoConnectionChecker.java  
    |    +-- SecurityConfig.java  
    |    +-- WebConfig.java  
    |  
    +-- controller/  
    |    +-- UserController.java  
    |    +-- ResultController.java  
    |    +-- ClassController.java  
    |    +-- WaitingUser.java  
    |    +-- Timetable.java  
    |    +-- Otp.java  
    |    +-- AnnouncementController.java  
    |  
    +-- model/  
    |    +-- User.java  
    |    +-- ClassModel.java  
    |    +-- Otp.java  
    |    +-- WaitingUser.java  
    |    +-- Result.java  
    |    +-- Announcement.java  
    |    +-- Timetable.java  
    |  
    +-- repository/  
    |    +-- UserRepository.java  
    |    +-- ClassRepository.java  
    |    +-- ResultRepository.java  
    |    +-- TimetableRepository.java  
    |    +-- WaitingUserRepository.java  
    |    +-- OtpRepository.java  
    |    +-- AnnouncementRepository.java  
    |  
    +-- service/  
    |    +-- UserService.java  
    |    +-- ResultService.java  
    |    +-- OtpService.java  
    |    +-- ClassService.java  
    |    +-- AnnouncementService.java
```

```
└── util/
    └── EmailUtil.java
```

## Frontend (React + Tailwind+Axios)

```
Frontend/
├── public/
└── src/
    ├── assets/
    ├── components/
    │   ├── admin/
    │   │   ├── AdminNavbar.jsx
    │   │   ├── CreateClass.jsx
    │   │   ├── Footer.jsx
    │   │   ├── Home.jsx
    │   │   ├── Result.jsx
    │   │   └── Timetable.jsx
    |
    │   ├── user/
    │   │   ├── Footer.jsx
    │   │   ├── Home.jsx
    │   │   ├── Navbar.jsx
    │   │   ├── Profile.jsx
    │   │   ├── Result.jsx
    │   │   └── Timetable.jsx
    |
    └── ProtectedRoute.jsx
    ├── feature/ authSlice.jsx
    ├── pages/
    │   ├── auth.jsx
    │   └── Layout.jsx
    ├── redux/Store.jsx
    ├── App.css
    ├── App.jsx
    ├── index.css
    └── main.jsx
└── package.json
```

## 7. Key Features

### User Features

- Secure registration with OTP verification
- Profile management
- View announcements and timetable
- Check semester and regular results

### Admin Features

- Approve or reject new users
- Create classes and subjects
- Upload student marks
- Manage timetables
- Create and view announcements

## 8. Tools and Software Used

- Visual Studio Code (Frontend)
- IntelliJ IDEA / Eclipse (Backend)
- MongoDB Compass / Atlas
- Postman for API testing
- Git & GitHub for version control

## 9. Testing

- **Unit Testing:** For services and repositories
- **Integration Testing:** Between frontend and backend APIs
- **Manual Testing:** To check all user and admin functionalities

## 10. Future Enhancements

- Add notification system for announcements
- Enable role-based dashboards (Teacher, Student, Admin)
- Implement marks graph visualization
- Add dark mode UI

## **11. Conclusion**

The Student Academic Portal successfully integrates all student and admin activities into one system.

It automates academic operations like registration, announcements, timetables, results, and class management, providing an efficient digital academic experience.

## **12. References**

- Spring Boot Documentation
- React.js Documentation
- MongoDB Manual
- Tailwind CSS Docs
- Java Mail API Reference