

Full Stack Project Description Report

Name: Tanay Manish Nesari

UID: 23BCS13761

Section: KRG-1B

Student Academic Portal

Tech Stack:

- **Frontend:** React
- **Backend:** Spring Boot
- **Database:** PostgreSQL

Use Case Overview

The Student Academic Portal serves as a centralized platform for students to access all essential academic information. It provides a secure login system for students to view their results, check class timetables, and stay updated with the latest announcements from the institution.

Administrators can manage and publish content, ensuring information is accurate and up to date.

Core Modules

1. Student Login

- **Authentication:** Secure login for registered students using unique credentials.
- **Profile Management:** View and update personal details such as name, contact info, and profile picture.
- **Password Management:** Option to reset or change passwords with security validation.

2. Results

- **Result Viewing:** Students can view semester-wise results, grades, and subject-wise marks.
- **Result History:** Access to past academic records for multiple terms.

3. Announcements

- **Notice Board:** Admin can publish academic announcements, event schedules, exam notifications, and policy updates.

4. Timetable

- **Class Schedule:** View subject-wise schedules with date, time, and venue information.
- **Exam Timetable:** Display upcoming exam schedules in a separate section.

Key Evaluation Aspects

1. Structured UI

- Clean, intuitive design with easy navigation between results, announcements, and timetable.
- Highlight urgent or time-sensitive updates for better visibility.

2. Admin-Controlled Notices

- Role-based access ensuring only admins can publish or edit announcements.
- Ability to schedule notices for future publishing dates.
- Maintain a log of edits and removals for transparency.

3. Role-Based Access Control

- **Student Role:** Access to personal results, timetables, and announcements.
- **Admin Role:** Full control over results upload, timetable creation, and announcement posting.
- Prevent unauthorized data access between different student accounts.

4. Version Control & Deployment

- Git-based versioning for tracking changes to both frontend and backend code.
- Continuous deployment setup for pushing updates without downtime.

How the Tech Stack Supports This

- **React:**
 - Dynamic and responsive UI for students and admins.
 - Modular components for results table, announcements feed, and timetable view.
 - API-driven data fetching to keep information updated in real-time.
- **Spring Boot:**
 - REST API for user authentication, results fetching, timetable management, and announcements CRUD.
 - Security layer with Spring Security for enforcing role-based access control.
 - Business logic for filtering data per student and scheduling updates.
- **PostgreSQL:**
 - Relational database for storing students, results, timetables, and announcements.
 - Support for filtering, ordering, and searching data efficiently.
 - Transaction management to ensure data consistency during updates by admins.