

# System Design Document

## Student Academic Portal

### 1. Introduction

The Student Academic Portal is a full-stack web application designed to act as a one-stop portal for academic information. It centralizes results, timetables, and announcements into a single structured user interface. The system uses a React frontend, Spring Boot backend, and PostgreSQL database.

### 2. System Requirements

#### 2.1 Functional Requirements

**Secure Login:** Students can register and log in securely.

**Academic Results:** Students can view grades and performance history.

**Timetable Management:** Students can view daily and weekly schedules.

**Notice Board:** Admin-controlled notices are visible to all students.

#### 2.2 Non-Functional Requirements

**Scalability:** Supports multiple concurrent users.

**Data Integrity:** PostgreSQL ensures consistent academic records.

**Security:** JWT-based authentication for secure sessions.

### 3. Module Division (8 Modules)

#### Frontend Modules (React)

Module 1: Authentication & Profile UI – Login, registration, profiles.

Module 2: Academic Dashboard UI – View results and transcripts.

Module 3: Info & Announcements UI – Timetables and notices.

#### Backend Modules (Spring Boot)

Module 4: Identity Management – Registration and JWT login APIs.

Module 5: Result Management Service – Manage student grades.

Module 6: Timetable Service – Class schedule logic.

Module 7: Announcement Service – Admin notices and history.

Module 8: Database & Persistence Layer – PostgreSQL schema and indexing.

## 4. High-Level Design (HLD)

**Architecture:** Client-Server model.

**Communication:** REST APIs between React and Spring Boot.

**Storage:** PostgreSQL database for persistence.

## 5. Low-Level Design (LLD)

### 5.1 Database Design (PostgreSQL)

**Student Table:** userId, name, email, password, department.

**Result Table:** resultId, userId, courseCode, grade, semester.

**Timetable Table:** scheduleId, courseName, timeSlot, room.

### 5.2 API Design

POST /api/auth/login – User authentication.

GET /api/student/results – Retrieve grades.

GET /api/announcements/summary – Fetch notices.

## 6. Future Enhancements

Export reports as PDF/Excel.

AI-based academic recommendations.

Native mobile application support.