

STUDENT PROGRESS TRACKER

A Full-Stack Academic Monitoring System

SAPTHAGIRI NPS UNIVERSITY

Submitted By:

Subrat Kumar Jena

(USN: 24SUPMCAGL093)

MCA -AI/ML “B”

Academic Year:

2024 – 2026

JAVA FULL-STACK INTERNSHIP BY TRIPLLAR

Under the Guidance of:

Mr. Gajendra Singh N

CHAPTER NO	DESCRIPTION	PAGE NO
1	Introduction	2
2	Problem Statement	3
3	Proposed Solution	4
4	Technology Stack	5
5	System & Database Design	6-8
6	System Workflow	9-11
7	Module Description/Results	12-15
8	Future Enhancements	16
9	Conclusion	17

Chapter 1

Introduction

In today's rapidly evolving, technology-driven education system, effective academic tracking and performance management have become essential for both students and educational institutions. With the growing volume of academic data and the increasing need for real-time insights, the ability to accurately record, monitor, and analyze student performance has become more important than ever. A well-structured digital system not only enhances transparency but also empowers educators and learners to make informed decisions, identify learning gaps, and improve overall academic outcomes.

Student Progress Tracker is a comprehensive full-stack academic management platform designed to streamline the entire student-performance monitoring process. It enables seamless management of attendance, assignments, test scores, and overall progress through an intuitive and data-rich dashboard. By bringing essential academic workflows together in a single unified system, the application eliminates the inefficiencies of manual record-keeping and fosters a more connected and data-driven learning environment.

As the sole developer of this project, **I, Subrat Kumar Jena**, conceptualized, designed, and built the Student Progress Tracker as part of my academic coursework. Throughout the development process, I focused on creating a secure, scalable, and responsive application that addresses real-world educational challenges. The project leverages a modern full-stack architecture:

- **Frontend:** React (Vite) for a smooth, fast, and interactive UI
- **Backend:** Spring Boot for secure REST APIs and robust business logic
- **Database:** MySQL for efficient and reliable data storage

This technology stack allows the system to deliver high performance while offering strong data handling capabilities and a clean, user-friendly interface.

The primary vision behind the Student Progress Tracker is to bridge the gap between traditional academic management methods and modern digital solutions. By automating key administrative tasks, the system reduces human error, saves time, and ensures high accuracy in student records. It also promotes better communication between students and administrators by providing real-time updates on academic performance, enabling them to take timely action and stay aligned with academic goals.

Overall, the Student Progress Tracker serves as a practical, scalable, and impactful solution that highlights how technology can transform the educational experience. It demonstrates the power of full-stack development in solving real-world problems and reflects my commitment to building meaningful and efficient digital systems.

Chapter 2

Problem Statement

Many educational institutions and students face significant challenges in maintaining an organized, accurate, and transparent record of academic performance. As education continues to evolve, the traditional methods of tracking student progress—often involving manual registers, spreadsheets, or outdated software—are no longer sufficient. These systems tend to be slow, error-prone, and lack the flexibility required to support modern academic needs. Without proper integration across essential modules such as attendance, assignments, internal assessments, and overall performance records, institutions struggle to maintain consistency and efficiency.

This lack of a unified system not only complicates administrative work but also affects the learning experience of students, who often find it difficult to monitor their progress or receive timely updates.

Some of the major challenges commonly faced include:

- **Limited or no real-time insights** into a student's academic performance, preventing early identification of weaknesses
- **Inefficient tracking mechanisms** for attendance and assignment submissions, leading to confusion or discrepancies
- **Absence of a unified digital platform** that connects students, teachers, and administrators in a streamlined manner
- **Dependence on manual data entry**, which increases the chances of human error and consumes unnecessary time
- **Lack of interactive visual analytics**, making it difficult to analyze academic patterns, trends, or overall growth

These recurring issues highlight the need for a modern, intelligent, and user-friendly academic tracking solution. Institutions require a system that not only simplifies record management but also enhances the overall academic experience. This led to the vision for a platform that provides:

- An intuitive and interactive dashboard
- Automated management of attendance, assignments, and results
- Clear visualizations for performance analysis
- Real-time updates and transparency for all stakeholders
- Easy accessibility and seamless navigation for both students and administrators

By addressing the limitations of traditional systems, such a platform ensures improved accuracy, enhanced efficiency, and a more data-driven academic environment—ultimately empowering students and administrators to make informed decisions and stay aligned with academic goals.

Chapter 3

Proposed Solution

The **Student Progress Tracker** is designed as a comprehensive academic monitoring system that modernizes and simplifies how students and administrators manage academic records. Acting as a centralized digital platform, it seamlessly integrates all essential academic information—attendance, assignments, assessments, and results—into a single, intuitive interface. This integration eliminates fragmentation across different systems and provides a smooth, transparent, and user-friendly experience for all users.

Developed with a strong emphasis on accuracy, accessibility, and real-time insights, the system empowers stakeholders to stay consistently informed about academic progress. Its clean design, coupled with advanced visual analytics, enables users to easily interpret performance patterns, identify areas for improvement, and take timely corrective actions when needed.

The system delivers a wide range of key capabilities, including:

- **Secure user authentication (JWT-based)** to ensure privacy, data protection, and controlled access
- **Assignment tracking** with clear status indicators such as *Pending*, *Submitted*, and *Late*, allowing students to stay organized and informed
- **Automated attendance analysis**, complete with percentage calculations and interactive visual representations for better understanding
- **Result visualization tools** that display subject-wise performance, trends, and score comparisons through dynamic charts and graphs
- **Real-time notifications** for upcoming deadlines, newly added results, and warnings related to low attendance
- **A powerful admin dashboard** that allows administrators to manage student records, attendance logs, results, assignments, and overall academic data efficiently
- **A visually enhanced student dashboard** that presents key insights using pie charts, bar charts, and line graphs, making complex data simple and meaningful

Overall, the Student Progress Tracker delivers a structured, intelligent, and data-driven approach to academic monitoring. By organizing information effectively and presenting it through visually engaging analytics, the system enhances transparency, boosts efficiency, and ensures that academic progress is always clear and easy to understand for both students and administrators.

Chapter 4

Technology Stack and Decision Rationale

Layer	Technology	Rationale
Frontend	React (Vite)	Modular and high-performance component-based framework
Backend	Spring Boot	Secure and scalable REST API framework
Database	MySQL	Reliable and easy to integrate with Spring
Authentication	JWT	Ensures secure session management
Charts	Recharts	Interactive and responsive visualization library
Notifications	React Toastify	For rounded, real-time notifications
Styling	CSS3 + Responsive Layout	Ensures modern and adaptive design

These technologies were chosen to create a responsive, efficient, and user-focused application that supports scalability and performance.

Chapter 5

Database Design

The **Student Progress Tracker** uses a **MySQL database** with a normalized structure that ensures reliability, scalability, and efficient academic data management.

The schema stores information related to **students, assignments, attendance, and results**, each linked through proper relational mapping.

Database Schema Overview

Table	Purpose
users	Stores student/admin details and login credentials
assignments	Manages assignment records for each student
results	Stores exam results, marks, and subject details
attendance	Tracks student attendance with date-wise entries

Schema Structure (SQL)

```
CREATE DATABASE IF NOT EXISTS student_tracker;  
  
USE student_tracker;
```

```
-- USERS
```

```
CREATE TABLE IF NOT EXISTS users (  
  id BIGINT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(150) NOT NULL,  
  email VARCHAR(150) NOT NULL UNIQUE,  
  roll_no VARCHAR(50),  
  password VARCHAR(255) NOT NULL,  
  role VARCHAR(20) DEFAULT 'STUDENT'  
);
```

```
-- ASSIGNMENTS
```

```
CREATE TABLE IF NOT EXISTS assignments (  

```

```
id BIGINT AUTO_INCREMENT PRIMARY KEY,  
student_id BIGINT,  
title VARCHAR(200),  
description TEXT,  
due_date DATE,  
status VARCHAR(20) DEFAULT 'PENDING',  
FOREIGN KEY (student_id) REFERENCES users(id)  
);
```

-- RESULTS

```
CREATE TABLE IF NOT EXISTS results (  
id BIGINT AUTO_INCREMENT PRIMARY KEY,  
student_id BIGINT,  
subject VARCHAR(100),  
marks DOUBLE,  
max_marks DOUBLE,  
exam_date DATE,  
FOREIGN KEY (student_id) REFERENCES users(id)  
);
```

-- ATTENDANCE

```
CREATE TABLE IF NOT EXISTS attendance (  
id BIGINT AUTO_INCREMENT PRIMARY KEY,  
student_id BIGINT,  
date DATE,  
status VARCHAR(20),  
notes VARCHAR(255),  
FOREIGN KEY (student_id) REFERENCES users(id)  
);
```

Relationships

- **One User → Many Assignments**
- **One User → Many Results**
- **One User → Many Attendance Records**

Each Assignment, Result, and Attendance entry belongs to one Student (user)

This relational design ensures proper data integrity and smooth retrieval for analytics and dashboards.

Key Characteristics

- **Referential integrity** using foreign keys
- **Normalized tables** to avoid redundancy
- **Supports secure multi-user operations**
- **Optimized for analytics dashboards** (attendance %, grade trends, assignment status)
- **Scalable** for multiple students and administrators

Chapter 6

System Workflow

The **Student Progress Tracker** follows a smooth and secure full-stack workflow, ensuring accurate data flow between the frontend, backend, and database.

Each layer—UI, API, and database—works together to provide a seamless academic management experience for both students and administrators.

Frontend Layer (React)

The frontend is developed using **React**, providing an interactive and responsive user interface.

Key responsibilities:

- Rendering pages such as **Dashboard, Assignments, Results, Attendance, Profile, and Admin Panels**.
- Handling interactions such as:
 - Submitting assignments
 - Viewing results and attendance
 - Admin adding assignments, results, or attendance
- All requests are sent to the backend using **Axios**, a promise-based HTTP client.

Example: When an admin adds attendance:

- The admin form triggers a **POST request using Axios** to `/api/admin/attendance`.

Backend Layer (Spring Boot)

The backend consists of **Spring Boot controllers, services (if used), and JPA repositories**.

Key responsibilities:

- Receiving and processing API requests from the frontend.
- Validating inputs and applying academic logic (assignment status, result calculation, etc.).
- Communicating with the MySQL database via **Spring Data JPA**.

Important API routes include:

- `/api/auth/login` – User authentication
- `/api/admin/students` – Manage students
- `/api/admin/assignments` – Manage assignments
- `/api/admin/results` – Manage results

- /api/admin/attendance – Manage attendance
- /api/assignments/student/{id} – Student assignment retrieval
- /api/results/student/{id} – Fetch exam results for a student
- /api/attendance/student/{id} – Fetch attendance records

Example flow for adding an expense:

1. **Frontend:**
Student clicks the “Submit” or “Completed” button on an assignment.
Axios sends a **PUT** request to:
/api/assignments/{id}/status
2. **Controller Layer:**
AssignmentController receives the request and extracts the new status (SUBMITTED / LATE / etc.).
3. **Service / Repository Layer:**
AssignmentRepository.save() updates the record.
4. **Database (MySQL):**
MySQL executes an UPDATE query on the assignments table.
5. **Backend Response:**
Returns updated assignment data as JSON.
6. **Frontend Update:**
Dashboard and Assignment page refresh the data visually.

Data Flow Summary

User → React (Forms, Dashboard, Actions)



Axios Request (JSON Payload)



Spring Boot Controllers

(AssignmentController, ResultController, AttendanceController, AuthController)



Service Layer / Repository Layer

(UserRepository, AssignmentRepository, ResultRepository, AttendanceRepository)



MySQL Database

(Tables: users, assignments, results, attendance)



Response → React (Updated UI, Dashboard Refresh, Notifications)

Authentication and Security

The Student Progress Tracker application uses JWT (JSON Web Tokens) to ensure secure and role-based access across the system.

How Authentication Works

- When a student or admin logs in, the backend verifies the credentials and issues a signed JWT token.
- The token is stored securely in the browser's localStorage on the frontend.
- All future API requests (e.g., fetching assignments, results, attendance) automatically include this token in the Authorization header.
- The backend has a JWT filter that intercepts requests, validates the token, and ensures the user is authenticated before accessing any resource.

Role-Based Access Control (RBAC)

- **ADMIN users can:**
 - Add/manage students
 - Add/update assignments
 - Add/update results
 - Add/update attendance
 - View student dashboards
- **STUDENT users can:**
 - View their assignments
 - View results and progress
 - View attendance
 - Update their profile

Security Benefits

- Prevents unauthorized access to student data.
- Ensures that only valid users with proper roles can perform actions.
- Protects all API requests using signed tokens.
- Eliminates session hijacking issues found in traditional login systems.

Chapter 7

Modules Implemented

1. Authentication Module

The Authentication Module manages secure user login and registration using Spring Security integrated with JWT (JSON Web Token) authentication. It validates user credentials, generates secure tokens, and ensures that only verified users can access the system's protected features. This module provides role-based access control, prevents unauthorized entry, and maintains data privacy across the entire application.

The screenshot shows the 'Student Login' page of the ProgressTracker application. The page has a light blue background. In the top left corner is the 'ProgressTracker' logo. In the top right corner are two buttons: 'Login' (light blue) and 'Sign Up' (blue). The main content is a white card with a blue header 'Student Login'. Below the header are three input fields: 'Email', 'Password', and 'Login as Student' (a dropdown menu). Below these fields is a blue 'Login' button. At the bottom of the card, there is a link: 'Don't have an account? Sign up'.

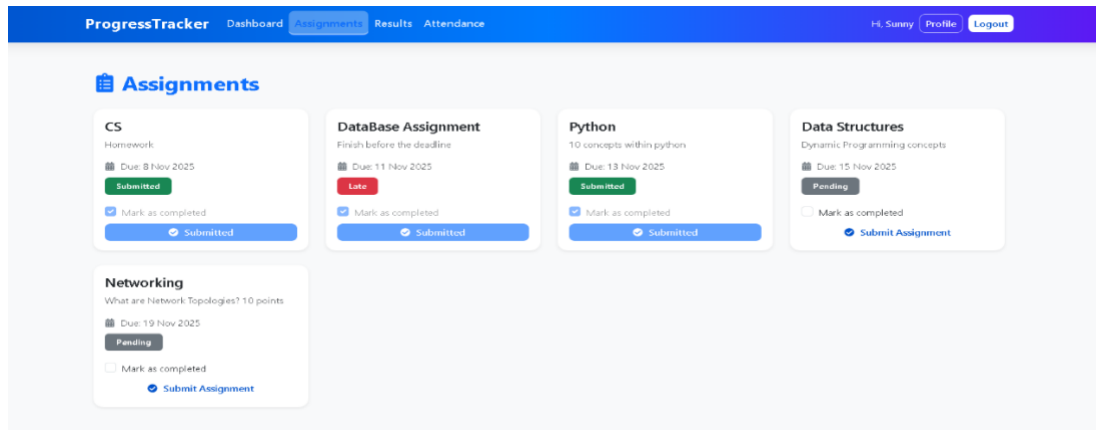
The screenshot shows the 'Student Sign Up' page of the ProgressTracker application. The page has a light blue background. In the top left corner is the 'ProgressTracker' logo. In the top right corner are two buttons: 'Login' (light blue) and 'Sign Up' (blue). The main content is a white card with a blue header 'Student Sign Up'. Below the header are four input fields: 'Full Name', 'Email', 'Roll No', and 'Password'. Below these fields is a blue 'Sign Up' button. At the bottom of the card, there is a link: 'Already have an account? Login'.

2. Assignment Management Module

Allows the admin to add, edit, and delete assignments easily.

Students can view assignment details such as due dates and status.

Smart UI cards and sorting help in quickly tracking pending and completed tasks.

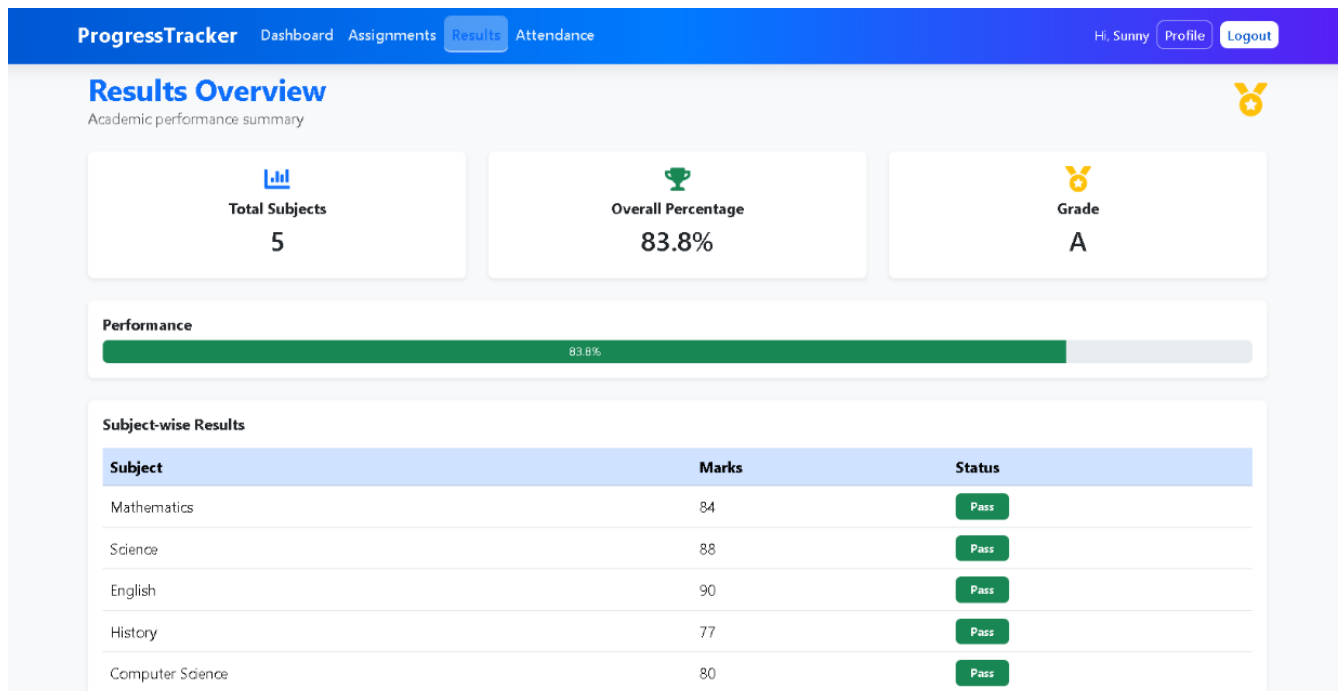


3. Result Management Module

Enables admins to upload and update students' exam results.

Displays subject-wise marks, total percentage, and performance trends.

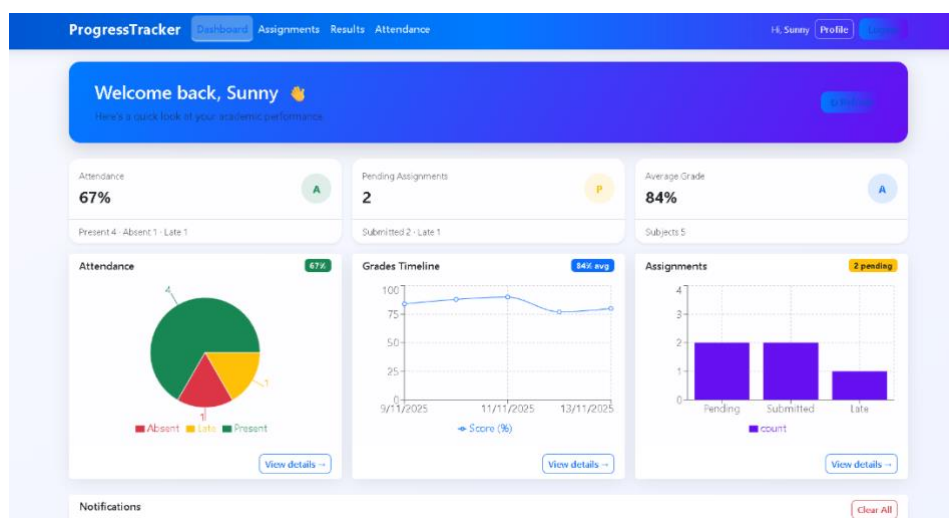
Students can view their complete grade history in a clean, organized format.



4. Analytics Module

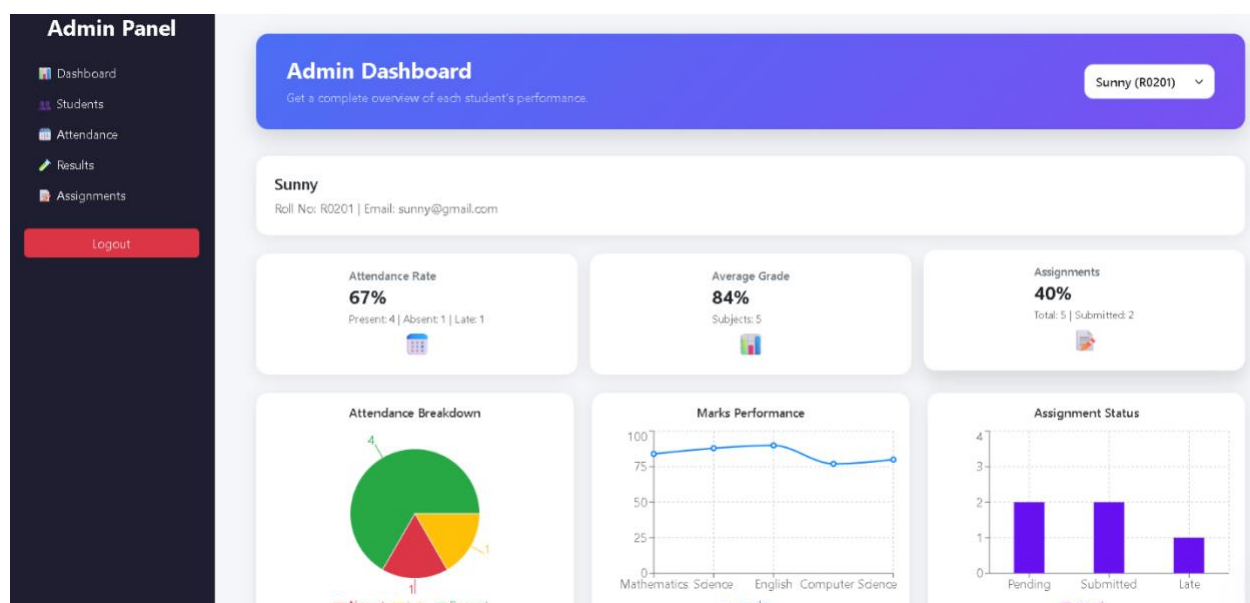
Provides visual insights into academic performance using:

- Pie Charts for attendance distribution
 - Line Charts for marks progression
 - Bar Charts for assignment status
- Helps students understand their strengths and weaknesses effectively.



5. User Interface and Theme Module

Implements a clean, responsive layout with easy navigation between dashboard sections. Ensures consistent visual design with enhanced UI components.

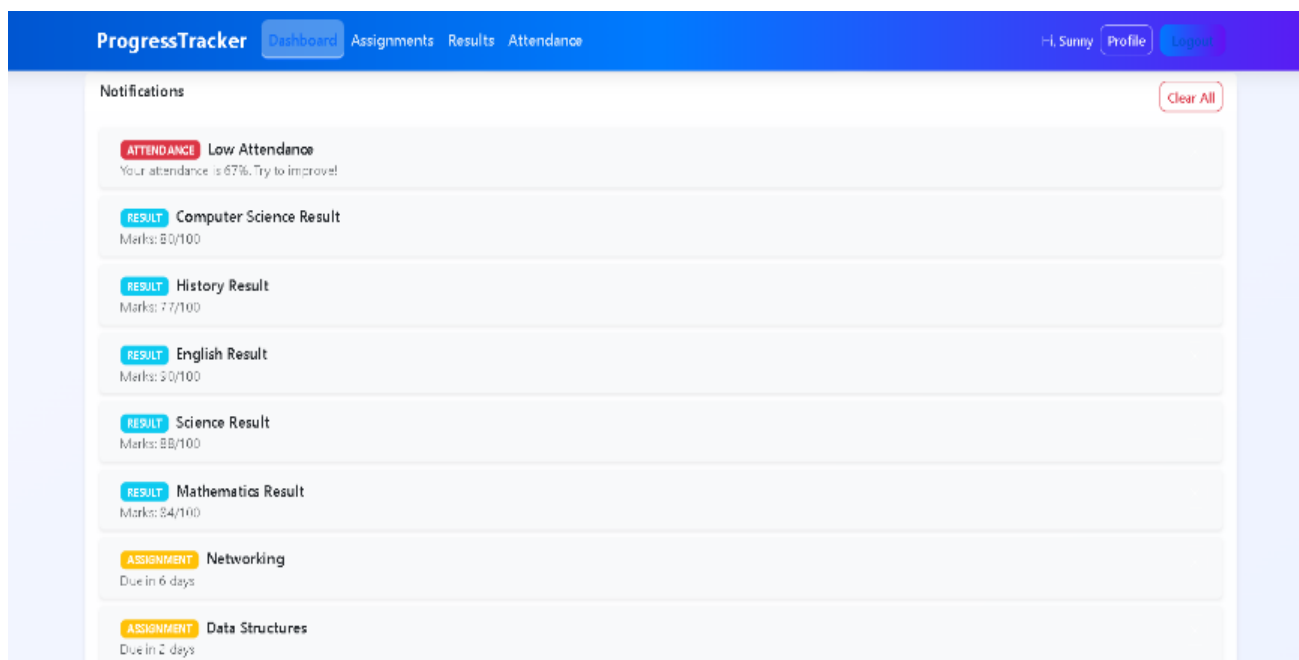


6. Notification Module

Displays real-time alerts for key events such as:

- Assignment updates
- Result uploads
- Low attendance warnings
- Upcoming deadlines

Notifications appear in a clean, modern panel, improving clarity and user experience.



Chapter 8

Future Enhancements

- **AI-Based Performance Prediction:**

Integrate machine learning algorithms to analyze attendance records, assignment scores, class participation, and previous exam results. The system can predict a student's future performance, highlight at-risk learners, and provide personalized recommendations to improve academic outcomes.

- **Automated Report Generation:**

Enable automatic creation of detailed student progress reports that include attendance summaries, assignment status, exam performance, and analytics charts. These reports can be exported in PDF, Excel, or print-ready formats for use by teachers, administrators, and parents.

- **Parent/Guardian Access Portal:**

Introduce a secure portal where parents or guardians can log in to view their child's academic data. They can track attendance, monitor assignment completion, review test results, and stay updated on overall academic progress.

- **Real-Time Teacher Notifications:**

Provide instant alerts to teachers and administrators whenever a student's attendance falls below a defined threshold or when their grades start to decline. This helps educators intervene early and offer necessary guidance or support.

- **Timetable & Events Module:**

Add features for managing class schedules, exam timetables, and institution-wide events. Students receive timely reminders, ensuring they stay informed about important dates, upcoming assessments, and academic activities.

- **Mobile App Integration:**

Develop an Android and iOS mobile application that syncs perfectly with the web system via REST APIs. This ensures students, teachers, and parents can access academic information anytime, from anywhere.

- **Chat Support Between Students & Teachers:**

Implement a secure in-app messaging feature that enables direct communication between students and teachers. This can be used for clearing doubts, discussing assignments, sharing feedback, and enhancing overall engagement.

Chapter 9

Conclusion

The development of the **Student Progress Tracker** showcases the end-to-end creation of a modern, scalable, and efficient academic monitoring system tailored to the needs of today's educational environment. This project demonstrates how digital solutions can transform the traditional academic management process by providing structured, real-time insights into student performance. Through its integrated modules and user-friendly design, the system effectively bridges the gap between manual record-keeping and intelligent academic analytics.

Working on this project significantly strengthened my technical capabilities across various domains. On the frontend, I gained in-depth expertise in **React and Vite**, building dynamic and responsive interfaces that prioritize clarity and usability. On the backend, I developed secure and efficient REST APIs using **Spring Boot**, implemented JWT-based authentication, and structured a robust database using **MySQL**. The project also deepened my understanding of **system architecture, API integration, state management, UI/UX principles, and data visualization**, enabling me to create a well-rounded, full-stack solution.

Beyond technical proficiency, this project enhanced my ability to think critically about real-world academic challenges. By analyzing the pain points of students, teachers, and administrators, I designed a system that consolidates attendance, assignments, and performance data into one centralized platform. The Student Progress Tracker delivers detailed performance insights through interactive charts, automated analytics, and personalized dashboards—making academic data easier to understand and more actionable.

The system successfully streamlines academic monitoring by offering transparency, reducing manual workload, and helping users make informed decisions based on clear data trends. Students can track their progress effortlessly, teachers can manage academic records more efficiently, and administrators benefit from improved accuracy and smoother workflows. The combination of intuitive UI elements, reliable backend processes, and real-time data updates makes the platform both powerful and easy to use.

Overall, the Student Progress Tracker stands as a comprehensive and impactful academic management solution. It not only highlights my technical and problem-solving abilities but also sets a strong foundation for future enhancements. Features such as AI-powered performance prediction, mobile app integration, real-time communication tools, and automated reporting can be seamlessly added, making the system even more advanced and future-ready.

This project represents a significant milestone in my journey as a full-stack developer—demonstrating my capability to conceptualize, design, and implement complete software solutions that address practical needs and deliver meaningful value.