Web X Mini Project Report

Topic: Student Performance Tracker

Name:- Ronit Santwani   Class:- D15A   Roll No.:- 48

Contents

1. Introduction

In today’s educational environment, continuous monitoring and tracking of student performance is crucial for both academic institutions and students. A digital solution for tracking performance can provide insights into individual progress, highlight areas for improvement, and support better decision-making by educators and students. This project presents a full-stack web application named **Student Performance Tracker**, which allows users to manage student profiles, record their academic grades, and visualize their academic performance trends using charts.

---

2. Problem Statement

Traditional methods of tracking student performance involve manual entry in registers or spreadsheets, which are not only time-consuming but also prone to human error. Moreover, analyzing data and trends is difficult without proper visualization tools. There is a need for a digital system that can streamline the data management process and provide analytical insights in real-time.

---

3. Objectives of the Project

- To develop an intuitive platform for managing student academic records.
- To allow users to add, edit, and delete student profiles.
- To record grades/marks per subject.
- To display grade distribution and performance analysis using charts.
- To create a responsive and user-friendly web application for teachers and academic administrators.

---

4. Requirements & Technology Stack

Frontend:

- React (Vite)
- TailwindCSS
- Recharts (for graphs)

Backend:

- Flask (Python)
- Flask-CORS

Database:

- MongoDB (with cloud-hosted MongoDB Atlas)

Other Tools:

- Axios (for HTTP requests)
- Node.js (for package management)

---

5. Proposed System

The proposed system is a full-stack web application that allows users to:

- Create and manage student profiles.
- Record each student's grades based on subjects.
- Update or delete student records.
- View a real-time pie chart displaying grade distribution.

The application provides a clean and interactive dashboard, simplifying the performance management process.

---

6. Proposed Design

The system follows a client-server architecture:

- The **client side** (React) offers a form for data entry and dynamically displays the student list and pie chart.
- The **server side** (Flask) handles HTTP requests, processes data, and communicates with MongoDB to store and retrieve student records.

The design is based on component-based architecture with separate UI elements for form inputs, student list display, and chart rendering.

---

7. Implementation

- **Form Submission:** The form captures `name`, `subject`, and `grade` inputs. On submission, Axios sends the data to the Flask API.
- **API Endpoints:** Flask provides RESTful endpoints (`GET`, `POST`, `PUT`, `DELETE`) for student management.
- **Database Operations:** MongoDB stores each student as a document containing their profile and academic data.
- **Chart Visualization:** Recharts renders a pie chart to show the distribution of grades.
- **Styling & Layout:** TailwindCSS is used to create a responsive and modern UI.

---

8. Result Analysis

The application was tested with multiple student entries.

Functionalities like adding, editing, and deleting records worked as expected. The pie chart updates dynamically based on the student list, providing an immediate visual summary of grade distribution. The application efficiently displays trends and supports quick decision-making.

9. Conclusion

The Student Performance Tracker provides an effective solution for managing and analyzing student performance in real-time. Its interactive and intuitive interface enables educational stakeholders to gain insights into academic trends. By combining a modern tech stack with a user-centric design, the application serves as a valuable tool for academic data management and analysis.

Future enhancements may include login authentication, exporting reports, and line charts for tracking individual performance trends over time.