

## Assignment Summary

- Tech Stack: MERN (MongoDB, Express, React, Node.js)
  - Additional Tools: TypeScript, Node Cron, Nodemailer, Codeforces API
  - Submission Includes: Code on GitHub, final demo video, and this documentation.
- 

## Project Overview

The Student Progress Management System is a web application designed to track competitive programming progress of students based on their Codeforces activity. The application provides features to manage student information and display Codeforces metrics in an intuitive and interactive way.

---

## Features

### Student Table View

- Displays student records: Name, Email, Phone Number, Codeforces Handle, Current Rating, Max Rating
- Add, Edit, and Delete student entries
- Download full dataset as CSV
- Click on a row to view detailed profile of the student

### Student Profile View

Divided into two sections:

#### Contest History

- Filter by last 30, 90, or 365 days
- Displays:
  - Rating graph
  - List of contests with name, rating change, rank, and count of unsolved problems

### Problem Solving Data

- Filter by last 7, 30, or 90 days

- Displays:
    - Most difficult problem solved (by rating)
    - Total problems solved
    - Average rating of solved problems
    - Average problems per day
    - Bar chart by rating buckets
    - Submission heatmap
- 

## **Codeforces Data Sync**

### **Node Cron Jobs**

- Library Used: node-cron
  - Sync runs every day at 2:00 AM
  - Fetches:
    - Contest data
    - Submission history
    - Rating updates
  - Admin can configure sync time and frequency
  - If a student's Codeforces handle is updated, data is fetched in real-time
- 

## **Inactivity Detection and Email Alerts**

### **Nodemailer Integration**

- Library Used: nodemailer
- After each sync, system:
  - Detects students with no submissions in the last 7 days
  - Sends automatic email reminders to them
- Tracks how many reminder emails each student has received

- Option to disable email alerts per student
- 

### **Interface and UX Enhancements**

- Responsive UI for mobile, tablet, and desktop
  - Light and dark theme toggle
  - Modular, clean, and well-documented codebase
- 

### **REST API Summary**

- GET /students - Get all students
  - GET /students/:id - Get student by ID
  - POST /students - Add a new student
  - PUT /students/:id - Update student data
  - DELETE /students/:id - Delete student
  - GET /students/:id/contests - Get contest history of a student
  - GET /students/:id/problems - Get problem-solving stats
  - PUT /students/:id/notifications - Enable or disable email reminders
- 

### **Submission Links**

- GitHub Repository: [GitHub Link Here]
  - Demo Video: [Video Link Here]
- 

### **Developed With**

- Frontend: React + TypeScript, TailwindCSS
- Backend: Express.js + Node.js
- Database: MongoDB (with Mongoose)
- External APIs: Codeforces API

- Email Service: Nodemailer (SMTP with Gmail)
- Scheduler: node-cron

---

## Acknowledgements

- [Codeforces Public API](#)
  - [Node Cron](#)
  - [Nodemailer](#)
-