**Student Progress Management System**

A full-stack MERN project to manage and monitor student progress using **Codeforces API**, featuring inactivity tracking and scheduled emails with **SMTP integration**.

**Features**

**Student Table View**

* View all students in a searchable, paginated table
* Columns: Name, Email, Phone, Codeforces Handle, Current Rating, Max Rating, Last Synced
* Actions:
  + ➕ Add new student
  + ✏️ Edit existing student
  + ❌ Delete student
  + 📥 Export student data to CSV
  + 🔄 Sync Codeforces data manually
  + 👁️ View detailed profile

**Student Profile View**

**Split into two sections:**

**Contest History**

* Filter: last 30 / 90 / 365 days
* Shows:
  + Contest list with rating change, rank
  + Rating line chart

**Problem Solving Data**

* Filter: last 7 / 30 / 90 days
* Shows:
  + 📊 Most difficult problem solved
  + 🧮 Total problems solved
  + ⭐ Average rating
  + 📆 Average problems/day
  + 📉 Bar chart of problems per rating bucket
  + 🔥 Submission heat map

**Data Sync (Codeforces API)**

* Automatically fetches updated student data using **Codeforces API** once per day via a **cron job**
* Real-time re-fetch occurs when a student’s handle is changed
* Admin can manually trigger full sync or per-student sync
* Sync time & frequency configurable via backend settings

**Email Reminders ( SMTP)**

* Students inactive for more than 7 days receive a motivational email
* Email sent using **SMTP (Nodemailer)** based on saved credentials
* Tracks number of reminder emails sent
* Admins can disable email reminders for individual students

**Tech Stack**

| **Layer** | **Tech** |
| --- | --- |
| Frontend | React, Tailwind CSS |
| Backend | Node.js, Express.js |
| Database | MongoDB + Mongoose |
| Charts | ApexCharts, Chart.js |
| Emails | Nodemailer (SMTP) |
| Cron Jobs | node-cron |
| Others | Joi (validation), CSV export, Toast notifications |

**API Endpoints**

**Students**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/students | Get all students |
| GET | /api/students/:id | Get full student data |
| POST | /api/students | Add a new student |
| PUT | /api/students/:id | Update student |
| DELETE | /api/students/:id | Delete student |
| POST | /api/students/:id/sync | Sync Codeforces data for student |

**Codeforces**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/codeforces/contests/:studentId | Contest data (30/90/365 days) |
| GET | /api/codeforces/problems/:studentId | Problem solving data |
| POST | /api/codeforces/sync-all | Manually sync all students |

**Settings (Cron/Email)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/settings | Get current cron/email settings |
| PUT | /api/settings | Update cron schedule/timezone or SMTP config |

**Emails**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/test/send-reminders | Manually trigger inactivity check |

**Folder Structure**

student-progress-system/

├── backend/

│ ├── models/

│ ├── routes/

│ ├── services/

│ ├── utils/

│ ├── .env

│ └── server.js

├── frontend/

│ ├── components/

│ ├── pages/

│ └── App.jsx

├── public/

│ └── assets/

│ └── demo.mp4 <- Video walkthrough

├── docs/

│ └── document.docx <- This documentation

└── README.md

**Setup Instructions**

**1. Clone and Install**

git clone https://github.com/your-username/student-progress-system.git

cd backend

npm install

cd ../frontend

npm install

**2. Setup .env**

# backend/.env

MONGODB\_URI=<your Mongo URI>

FRONTEND\_LOCAL\_URL=http://localhost:5173

**3. Run Locally**

# Start backend

cd backend

npm run dev

# Start frontend

cd ../frontend

npm run dev

**Video Walkthrough**

[Demo Video Link](hhttps://github.com/yshagit31/Student-Progress-Management/blob/main/frontend/public/assets/Demo.mp4)  
(Add your Google Drive / YouTube / Loom link)

**GitHub Repo**

[GitHub Repo Link](https://github.com/yshagit31/Student-Progress-Management)

**🤝 Acknowledgements**

* [Codeforces API](https://codeforces.com/apiHelp)
* [Nodemailer](https://nodemailer.com/)
* [ApexCharts](https://apexcharts.com/)
* [react-hot-toast](https://react-hot-toast.com/)