# TLE CodePlus: Documentation

**Ashutosh Jaiswal**
*CSE: Data Science and AI*
*IIIT-NR*

## 1. Product Overview

TLE CodePlus is a comprehensive competitive programming education management system designed to track student progress on Codeforces. It provides detailed analytics, progress tracking, and automated management features for educational institutions.

### 1.1. Live Application Links

- **Frontend Application:** https://tle-code-plus.vercel.app/
- **Backend API:** https://tle-codeplus.onrender.com/
- **GitHub Repository:** https://github.com/ashut0shj/TLE-CodePlus

### 1.2. Key Features

- Student progress tracking and analytics
- Contest history visualization
- Problem-solving statistics
- Activity heatmaps
- Automated data synchronization with Codeforces
- CSV export functionality

## 2. Technology Stack

### 2.1. Frontend Technologies

| Technology | Purpose |
|---|---|
| React.js | Primary UI framework with component-based architecture |
| Tailwind CSS | Utility-first CSS framework for responsive design |
| Recharts | Data visualization library for interactive charts |
| Calendar Heatmap | Activity pattern visualization over time |

TABLE I
FRONTEND TECHNOLOGY STACK

### 2.2. Backend Technologies

| Technology | Purpose |
|---|---|
| Node.js | JavaScript runtime environment |
| Express.js | Web application framework for RESTful APIs |
| MongoDB | NoSQL database for flexible data storage |
| Mongoose | Object Data Modeling (ODM) library |

TABLE II
BACKEND TECHNOLOGY STACK

## 3. User Interfaces

### 3.1. Student Management Interface

- **Student Table:** Complete CRUD operations with advanced filtering
- **Search & Sort:** Efficient data navigation
- **Data Export:** CSV export for external analysis
- **Visual Indicators:** Color-coded rating systems and inactivity highlighting

### 3.2. Student Profile Dashboard

- **Contest History:** Interactive rating progression charts
- **Performance Analytics:** Problem-solving patterns and statistics
- **Activity Visualization:** Daily coding activity heatmaps
- **Achievement Tracking:** Personal milestones and hardest problems solved

### 3.3. BONUS FEATURES

- **Mobile & Tablet Responsive:** Optimized for all screen sizes
- **Dark/Light Mode Toggle:** System-wide theme switching
- **Well-Documented Code:** Comprehensive code documentation

## 4. Automation & Performance Features

### 4.1. Mailing Service

- Integrated mailing service for student notifications
- Configurable inactivity period for reminder triggers
- Automatic and manual reminder sending capabilities
- Email reminder tracking and counting per student

### 4.2. Automatic Data Refresh

- Scheduled job runs daily at 2 AM server time
- Fetches latest Codeforces data for all students
- Updates ratings, contests, and problem data automatically
- Refreshes inactivity status and reminder tracking

### 4.3. Manual Data Refresh

- User-triggered refresh for specific students
- Immediate profile and activity stats update
- Available from student profile page
- Fetches real-time data from Codeforces API

### 4.4. Caching & Rate Limiting

- Database caching prevents excessive API calls
- Data refresh only at scheduled times or manual triggers
- Fast UI performance with cached data
- Protection against external API rate limits

## 5. API Endpoints

### 5.1. Base URL

`https://tle-codeplus.onrender.com/api`

### 5.2. Student Management Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | `/students` | Retrieve all student records |
| GET | `/students/:id` | Fetch specific student details |
| POST | `/students` | Create new student profile |
| PUT | `/students/:id` | Update student information |
| DELETE | `/students/:id` | Remove student record |
| GET | `/students/export` | Download student data as CSV |

TABLE III: Student Management API Endpoints

### 5.3. Data Synchronization Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | `/students/:studentId/contests` | Get contest participation history |
| GET | `/students/:studentId/problems` | Fetch problem-solving statistics |
| POST | `/students/:id/refresh` | Trigger manual data sync |

| | | |
|---|---|---|
| POST | `/students/:id/send-reminder` | Send reminder notifications |

TABLE IV: Data Synchronization API Endpoints

### 5.4. Analytics Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | `/students/:id/heatmap` | Generate activity heatmap data |

TABLE V
ANALYTICS API ENDPOINTS

## 6. Database Models

### 6.1. Student Model

| Field | Type | Description |
|---|---|---|
| name | String | Student full name |
| email | String | Email address |
| phoneNumber | String | Contact number |
| codeforcesHandle | String | Codeforces username |
| currentRating | Number | Current Codeforces rating |
| maxRating | Number | Maximum rating achieved |
| enrollmentDate | Date | Registration date |
| isActive | Boolean | Activity status |
| lastUpdated | Date | Last profile update |
| lastSubmissionDate | Date | Last problem submission |
| lastDataSync | Date | Last synchronization with Codeforces |

TABLE VI: Student Model Structure

### 6.2. Contest Model

| Field | Type | Description |
|---|---|---|
| studentId | ObjectId | Reference to student |
| contestId | String | Codeforces contest ID |
| contestName | String | Contest title |
| contestDate | Date | Contest date |
| oldRating | Number | Rating before contest |
| newRating | Number | Rating after contest |
| rank | Number | Contest rank |
| problemsSolved | Number | Problems solved in contest |
| totalProblems | Number | Total problems in contest |
| ratingChange | Number | Rating change (+/-) |
| contestType | String | Type of contest |

TABLE VII: Contest Model Structure

## 6.3. Problem Model

| Field | Type | Description |
|---|---|---|
| studentId | ObjectId | Reference to student |
| problemId | String | Unique problem identifier |
| problemName | String | Problem title |
| contestId | String | Associated contest |
| problemIndex | String | Problem index in contest |
| rating | Number | Problem difficulty rating |
| tags | Array[String] | Problem categories/topics |
| solvedDate | Date | Solution date |

| | | |
|---|---|---|
| submissionId | String | Submission identifier |
| verdict | String | Submission result |
| programmingLanguage | String | Language used |
| timeConsumed | Number | Execution time |
| memoryConsumed | Number | Memory usage |
| points | Number | Points scored |

TABLE VIII: Problem Model Structure

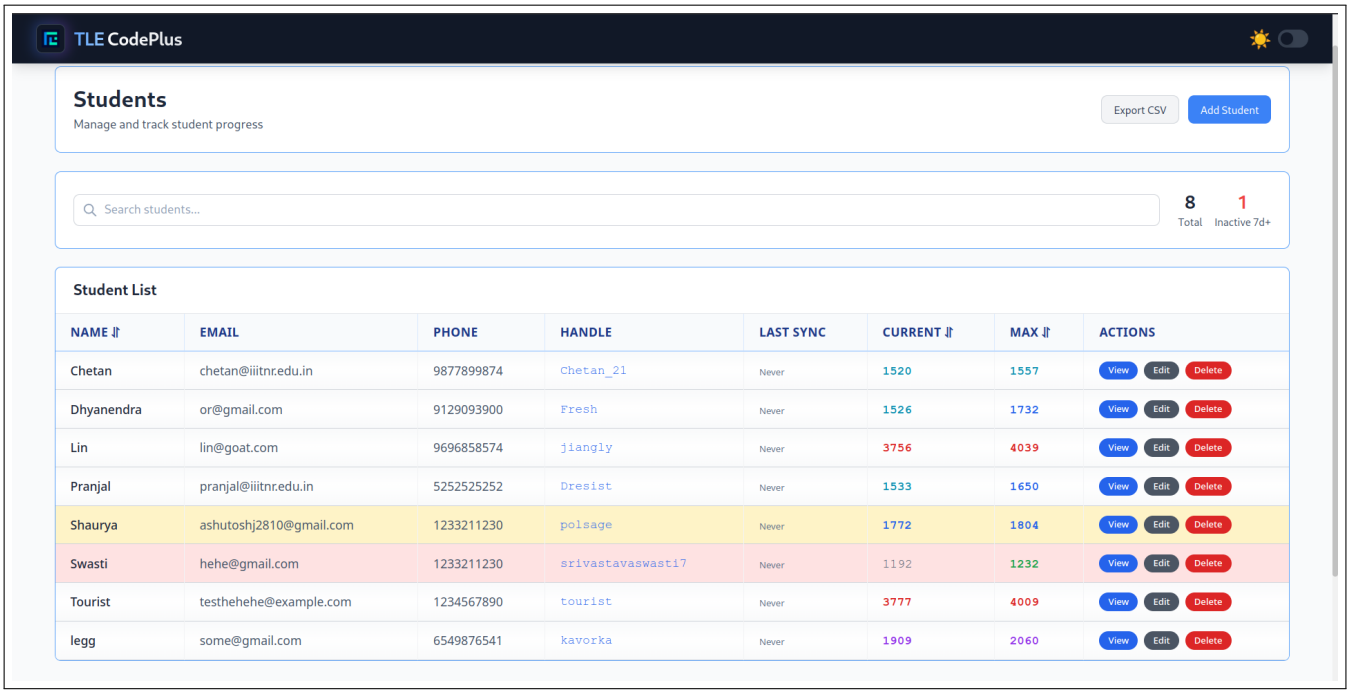


Fig. 1. Main Page with Student Table View

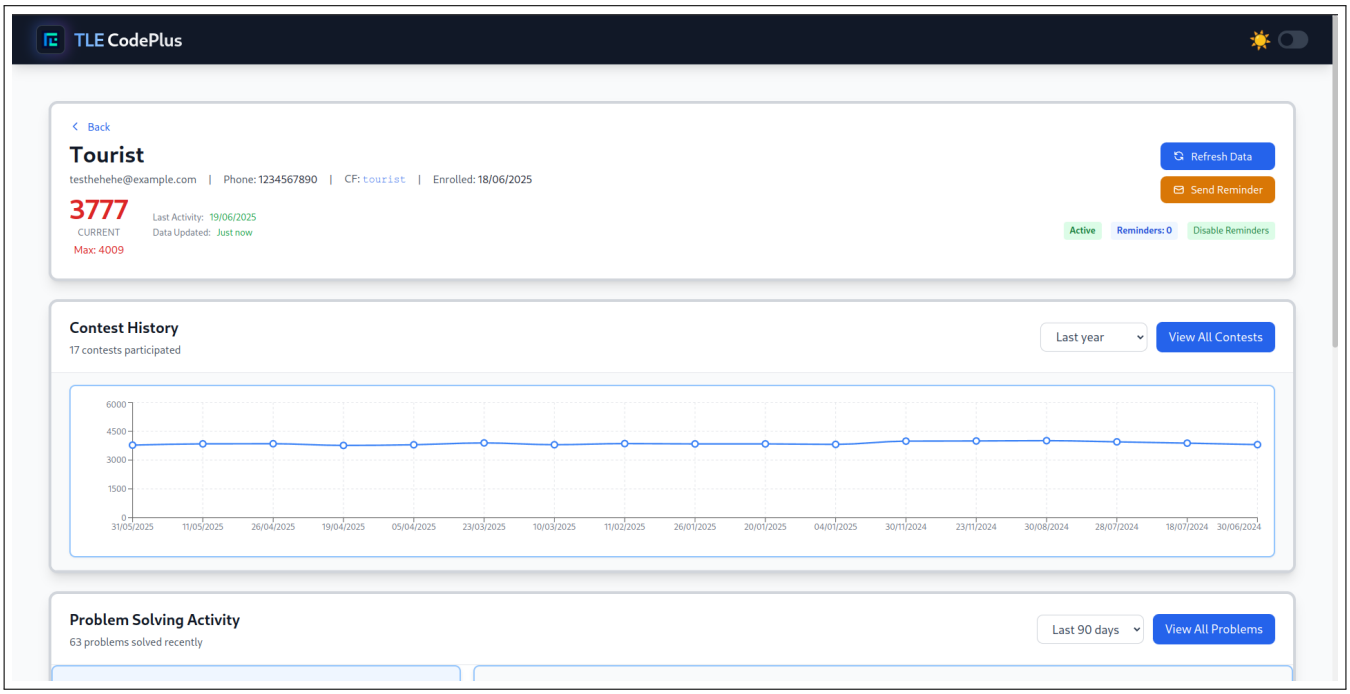


Fig. 2. Student Profile Overview and Contest History

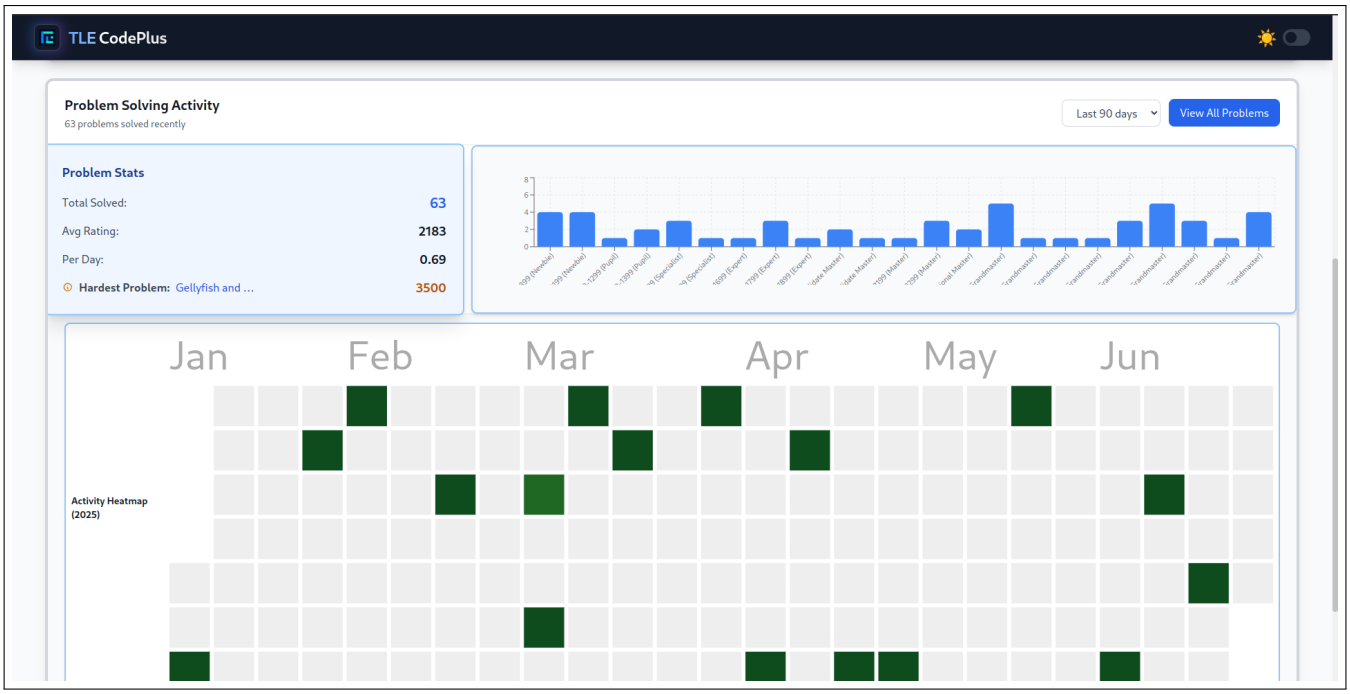


Fig. 3. Problem Solving History and Submission Heatmap

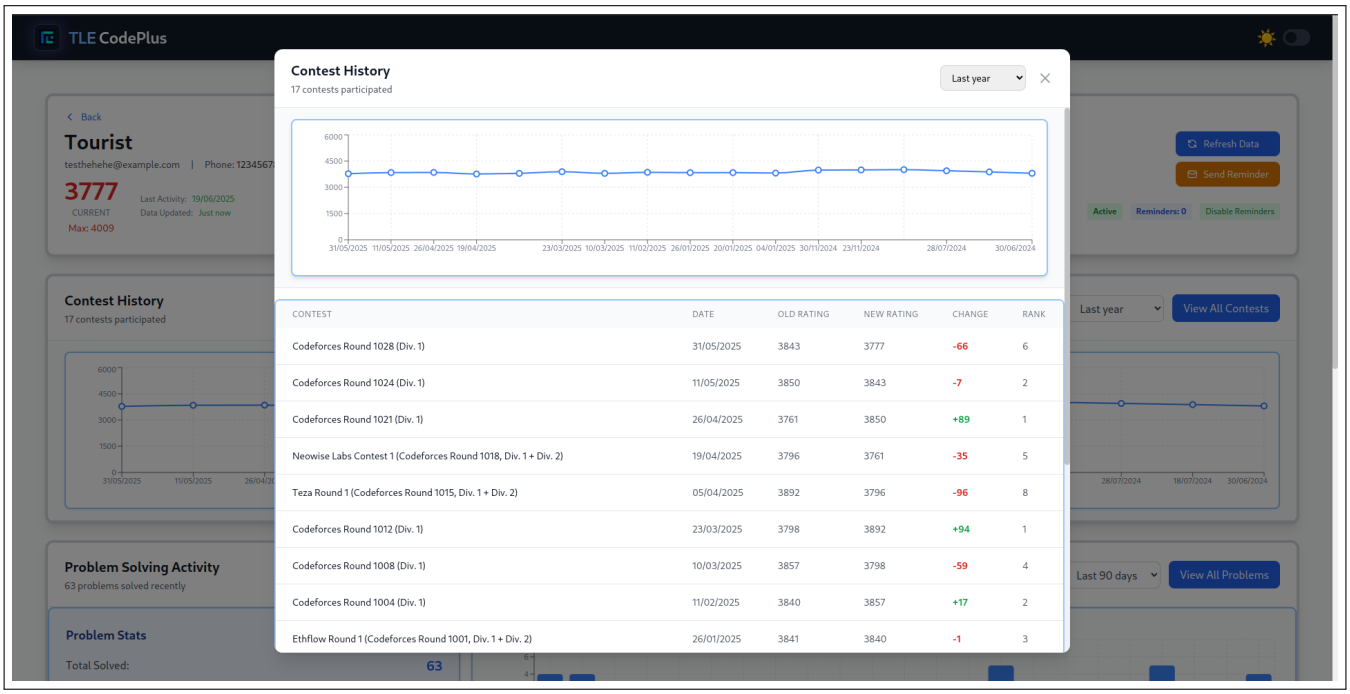


Fig. 4. Contest History List and Performance Statistics

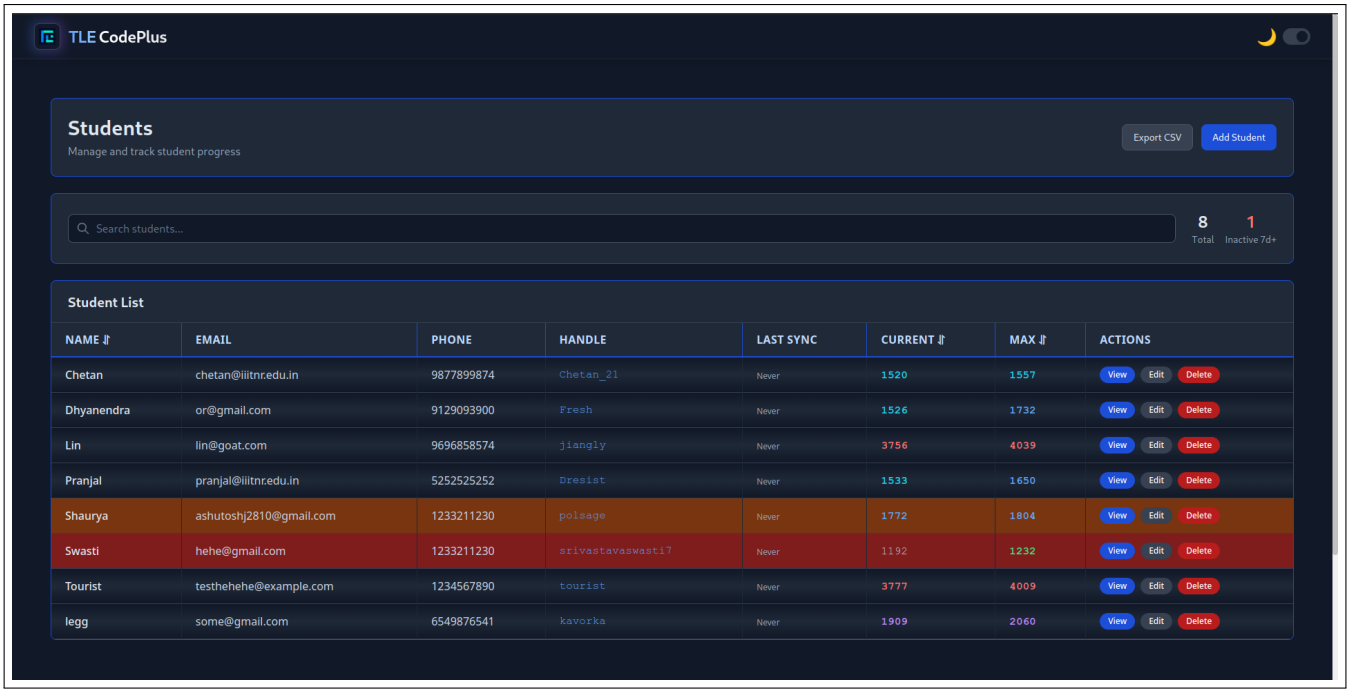


Fig. 5. Dark Mode UI Preview of the Student Dashboard