

Member Management System for Students

Project Demo in Next.js

September 28, 2024

1 Project Overview

The Member Management System is a web application built using **Next.js**, focused on managing student members within an organization. The users of the system are students who serve as officers responsible for recording detailed information about other students. The system will track the following student data:

- Basic details (Name, Age, Email, etc.)
- Achievements
- Participations in activities
- Loyalty points
- Skills and projects done
- Registrations in events
- Impact to the community
- Badge assignment based on tracked performance

2 System Features

The system offers the following core features:

- **CRUD Operations:** Full Create, Read, Update, and Delete (CRUD) functionality for managing students' information and their tracked data.
- **Activity Statistics:** Generate statistics to track student participation, involvement in activities, and overall activeness.
- **Heat Map:** Visual representation (heat map) showing students' strengths, helping to identify areas where they excel and where they may need improvement.

- **Role-based Access:** Admins have full control, while sub-users (Promotional Team, Record Team, Membership Team) have specific access rights.
- **Student Access:** Students can view their personal data through the system interface.
- **Data Export:** Ability to export students' records as PDFs for individual students and as Excel summaries for officers.

3 Data Storage

The system stores all member data in a local **SQL Server**. The system allows:

- **Admin Access:** Full access to all functionalities and data.
- **Sub-Users:** Limited access based on roles (Promotional Team, Record Team, Membership Team).
- **Student Access:** Read-only access to their individual records.

4 User Interface

The user interface is designed with **Tailwind CSS** for styling, and **Recharts** for rendering interactive visualizations, including:

- Charts and graphs showing student participation in events.
- Heat maps to highlight areas where students excel.

5 Data Handling and API Integration

The system interacts with a backend using a **REST API** for fetching and updating member data. Data is retrieved from the local SQL Server and displayed dynamically on the frontend.

6 Security Measures

To ensure data security, the system implements the following:

- **Bcrypt** for encrypting sensitive data such as passwords.
- **JWT Tokens** for secure authentication and authorization of users.

7 Data Export Options

The system provides the ability to export:

- Individual student records as **PDF** files.
- Summarized reports for officers as **Excel** files.

8 Project Deployment

The application will be deployed on the school's internal servers, ensuring it remains accessible within the school's network.