This project is a full-stack web application for college event management.The backend and frontend is divided into two distinct services.

Website : https://admin-portal-3f7t.onrender.com

The college-event-management which is the backend handles all the business logic and API routes while the admin-portal which is the frontend manages the user interface for admins and admin features.

Backend:College Event Management:

Using Node.js and Express.js the college-event-management is implemented to serve as RESTful API.

It uses Supabase(Postgre) as its database,which makes it easy to authenticate and store data.The directory structure is organized with specific folders for various components and features.

server.js:This is considered as application's entry point which initializes and starts the Express server.

src/app.js:The middleware,CORS configuration,the API routes declared in the routes directory is handled by this file.

src/config/supabaseClient.js:This is the point where the connection to the Supabase database is set up and initialized and also where an object client is passed to other parts of the application.

src/controllers:This is the point where the majority of the logic for every API endpoint resides.For example,authController.js processes user registration and login on the other hand eventController.js processes event creation and updating

src/middleware/verifyAdmin.js:This is applied to routes such as POST/admin/events to verify that only authorized and logged-in administrators can access them and also it is an important security middleware function.

src/routes:For specifying the API endpoints and mapping them to the respective controller functions this folder is used.

The primary functions are API endpoints/auth,/admin,/students,and /reports:authentication,admin activities such as creating events,student actions and report generations.

Frontend:Admin Portal:

The admin-portal serves as the administrator user interface and it is a React+Vite application

It an up-to-date development toolchain featuring Vite and Tailwind CSS and it is a single-page application

src/App.jsx:This is the top-level component whivh manages the application usually responsible for routing and state management.

src/pages:This is the folder containing various views of the application,each mapping to a particular route.

Login.jsx and Register.jsx:this is a top-level landing pagewhich probably shows a summary of event metrics utilizing components such as ChartCard.jsx

Events.jsx:This is an admin page which handles events,talking to the /admin/events endpoint.

Reports.jsx:This page provides insights into events and student activity and graphs data from the backend’s reporting endpoints.

Students.jsx:This is an admin page for handling student-related data,such as attendance and feedback.

src/components:This directory includes reusable UI components like Navbar.jsx and Sidebar.jsx which provides a consistent navigation experience across the application.

Tailwind.config.jsx:This is the file which enables quick styling in the JSX files directly and configures Tailwind CSS,a utility-first framework for designing the UI.

Frontend:React.js,Vite,Tailwind CSS which is deployed on Render.Deployment:Both frontend and backend are deployed individually on Render,which is the component practice for web applications that supports independent scaling and upkeep of the two services.the frontend sends API calls to the deployed URL of the backend to retrieve and send data

Technologies and Deployment:

Backend:Node.js,Express.js,Supabase,deployed on Render

Frontend:React.js,Vite,Tailwind CSS,deployed on Render.