# Social Media Dashboard

## Project Overview

The Social Media Dashboard is a centralized platform designed to help users effectively track, analyze, and manage their social media presence across multiple platforms such as Twitter and Instagram. It provides real-time data visualization, analytics, and a content scheduler for efficient post management.

## Features

* User Authentication and Profiles: Secure registration and login for managing private dashboards.
* Data Visualization (Charts/Graphs): Display follower growth, engagement rates, and content performance with interactive charts.
* Analytics: Provides audience demographics, best posting times, and top-performing content insights.
* Content Scheduler: Compose, schedule, and queue posts for automatic publication.
* Multi-Platform Integration: Connects with external social media REST APIs (simulated in the front-end).

## Tech Stack

|  |  |  |
| --- | --- | --- |
| Component | Technology | Description |
| Frontend | React.js | Single Page Application (SPA) with Tailwind CSS and Recharts for visualization. |
| Backend | Node.js (Express) | RESTful API server handling business logic and authentication. |
| Database | PostgreSQL | Stores user profiles, metrics, and scheduled content. |
| APIs | Twitter / Instagram | External APIs used for metrics fetching and posting content. |

## Getting Started

Follow these steps to set up and run the Social Media Dashboard locally.

### Prerequisites

Ensure the following are installed:

* Node.js (LTS recommended)
* npm or yarn
* PostgreSQL database instance

### 1. Backend Setup (Node.js/Express)

Clone the repository and install dependencies:

git clone [Your Repository URL]  
cd social-media-dashboard/backend  
npm install

Create a .env file in the backend directory with configuration:

PORT=3001  
DATABASE\_URL=postgres://user:password@host:port/database\_name  
JWT\_SECRET=your\_strong\_secret\_key  
TWITTER\_API\_KEY=your\_twitter\_api\_key  
INSTAGRAM\_CLIENT\_ID=your\_instagram\_client\_id

Run migrations and start the server:

npm run migrate  
npm start

### 2. Frontend Setup (React.js)

Navigate to the frontend directory and start the app:

cd ../frontend  
npm install  
npm start

The application should open at http://localhost:3000

## Usage

Login with simulated credentials:

Email: user@socialdash.com  
Password: password

## Pages

* Dashboard: Displays KPIs and engagement charts.
* Analytics: Shows detailed platform-specific insights.
* Schedule: Allows post composition and queue management.