Management App – Project Overview

# Backend

Repository: https://github.com/gsaravanakumar932/mgmt-app-server

* Google Login: Implements secure OAuth2 authentication via Google Sign-In, enabling users to log in using their Google accounts.
* Microservices Architecture: The backend is built on a modular microservices architecture, promoting scalability, maintainability, and independent service deployment.
* Dockerized Application: All services are containerized using Docker to ensure consistent development and deployment across environments.
* Redis for Cache: Integrated Redis to cache frequently accessed data and reduce load on primary databases, improving application performance.
* Prometheus and Grafana for Monitoring and Logging: Prometheus is used for collecting metrics from services, and Grafana provides real-time dashboards and visualizations for monitoring system health.
* API Gateway, Rate Limiter, JWT Authentication: A centralized API gateway manages request routing, enforces rate limits, and secures endpoints using JWT-based authentication.
* Consul for Service Discovery: Integrated HashiCorp Consul to enable automatic service discovery and health checks for better communication among microservices.
* Facebook Integration: Facebook login functionality is implemented but currently pending activation due to a missing client ID. The integration code is already in place.
* MongoDB with Replica Set for Posts and PostgreSQL for User Auth: The system uses MongoDB (with replica set configuration) for storing posts and PostgreSQL for managing user authentication and authorization.

# Frontend

Repository: https://github.com/gsaravanakumar932/mgmt-app-client-ui

* Google Login: Supports Google Sign-In for a seamless and secure login experience on the client side.
* NgRx for State Management: Uses NgRx to manage application state predictably and efficiently, based on Redux principles.
* Nginx Reverse Proxy for Backend API Calls: Configured Nginx to act as a reverse proxy, forwarding frontend API calls securely to the backend.
* Unit Test Cases and Cypress E2E Testing: Achieved 88% unit test coverage and implemented end-to-end tests using Cypress to ensure robustness and reliability.

# Deployment & Infrastructure

Docker Compose is used for infrastructure management and deployment, making it easy to set up the entire stack with one command. This approach ensures all services, including databases, backend services, and monitoring tools, are orchestrated and run consistently.