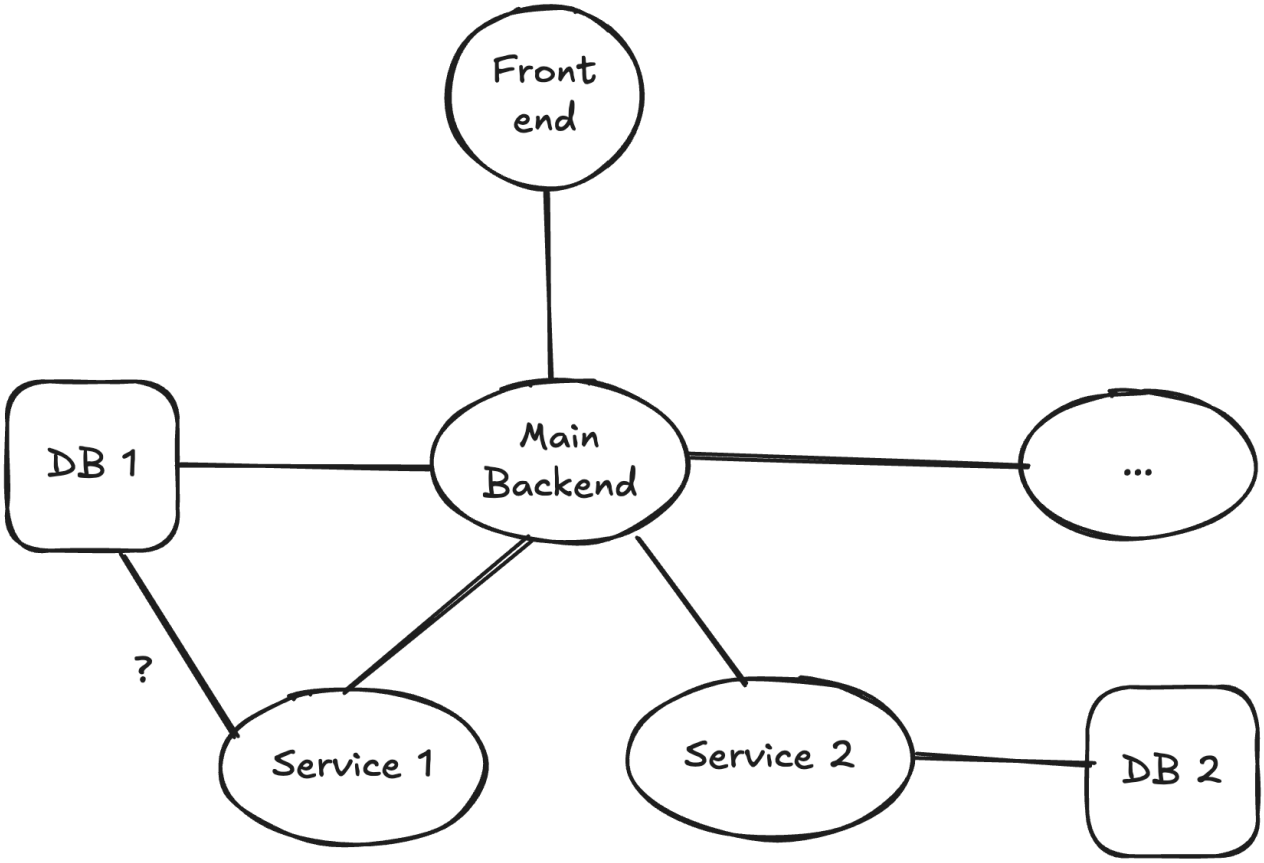
Final Project: Building a Production-Ready Full Stack JS App

Project Overview

This project involves creating a comprehensive microservice architecture web application that encompasses both client-side and server-side functionalities. The project topic is flexible, allowing you to explore areas of personal interest or market demand. The aim is to develop a full stack application that is ready for production, emphasizing the integration of frontend, backend, and database components.

Key Requirements

- **Topic Flexibility**: you can choose any topic for your application. Make sure it's a topic that interests you, and that it might be an idea you're willing to develop further in the future.
- **Technology Stack**: Any JavaScript framework or runtime (e.g., React, Angular, Vue.js for frontend; Node.js, Deno for backend; npm, bun for runtime) can be utilized. Choice of database (SQL or NoSQL) is also flexible.



- **Architecture Components**:
  - **Frontend**: A user interface that is interactive and user-friendly.
  - **Backend**: Server-side logic that handles business processes. The backend must include at least 3 projects: one main backend, and at least 2 microservice projects.

    - **Database**: Persistent storage for application data is needed. The project should include at least 2 databases.

- **Communication**:
    - The frontend and backend must communicate using the backend's API to fulfill the application's functionality.
    - Communication between frontend and backend should be authenticated to ensure security.
- **Authentication**:
    - Implement login functionality with Google for user authentication.
    - Ensure that access to certain URLs is restricted to logged-in or authorized users only. Unauthorized access should redirect users to an appropriate screen, such as the login page.
- **API as a Service**:
    - Provide an API (added to the backend) that external users can interact with by passing an authentication token (JWT). The API should return responses based on the services offered by the application. Have a demo ready to present showing an API call (using a tool like Postman) to that API with a JWT, and another call without a JWT to show an "access denied" response.
    - Implement proper error handling for unauthenticated requests.
    - Bonus points to integrating an external API service.
- **API Paradigms**:
    - The application should utilize at least two API paradigms.
- **Team Collaboration**:
    - you may work in groups of up to 3 members. Using a Task manager (clickup is a really nice tool for example, and this is purely optional) might help make sure that everyone has a clear path for what needs to be developed.

Presentation Day

During your presentation, all team members must participate. You will have 12 minutes to present your work. Make sure to focus on the key project points:

- An app demonstration
- tech stack used
- the different services created, with their databases
- The APIs and their paradigms (a quick view over key points of the code)
- the authentication process

This project is an opportunity for you to demonstrate your ability to create a comprehensive, real-world application using modern JavaScript technologies. Best of luck!