This document provides detailed project ideas for the final assignment. Each project includes the core concept, required microservices, optional extensions, and expected technology stack.

**Blog Platform**

**Core Concept:** A multi-user blogging platform where users can publish and interact with content.

**Required Microservices:**

**Database Needs:**

* Mongo DB
* User DB (authors, readers)
* Content DB (posts, categories, tags)
* Interaction DB (comments, reactions)

Travel blog with location integration

**Core Requirements**

Every final project must include the following components:

**1. Architecture & Services**

* **Microservices Architecture**: Implement at least 2 separate backend services plus a frontend
* **Service Boundaries**: Each service should have clear responsibilities and API contracts
* **Independent Deployment**: Services must be containerized separately

**2. Technology Stack**

* **Frontend**: Single-page application using a modern framework (React)
* **Backend Services**: RESTful APIs using appropriate frameworks (Node.js)
* **Database**: Mongo DB
* **API Communication**: Well-defined interfaces between services
* **Configuration**: Environment variables for configuration

**3. Infrastructure & Deployment**

* **Docker**: Each service must have a Dockerfile
* **Docker Compose**: Complete docker-compose.yml for local deployment
* **Environment Isolation**: Proper separation of development and production concerns

**4. Security Fundamentals**

* **Authentication**: User registration and login functionality
* **Authorization**: Basic role-based access control
* **Input Validation**: Sanitization of user inputs
* **Secure Configurations**: No hardcoded credentials

In the end, the whole app should be able to run with docker compose up --build

**Required Microservices:**

**User Service (Auth & Profiles). Handles user registration, login, and profile management.**

Functions: POST /register, POST /login, GET /profile

**Content Service (Posts, Media, Notifications, Comments & Likes)**

Manages user-generated content including posts, comments, likes, and media uploads.

Includes: Creating posts, adding comments, and liking/unliking posts. Uploading and deleting photos and videos (POST /media, DELETE /media). Sending notifications when someone likes or comments on a post (GET /notifications, PUT /notifications/read)

**Location Service**

Handles location data and helps users discover places.

Functions: POST /location, GET /location

FrontEnd : React js

BackEnd : Node js

Database: Mongo DB

How the app should work:

* A user **creates a post** → Content Service stores it
* Another user **likes or comments** → Notifications Service alerts the original author.
* Users receive **notifications in real time**.

If u need my mongo db uri and my jwt token ill send it to u.

1. [{\_id: "p-1749932574633", userId: "684c10cda9558bd2653327bb",…},…]
   1. 0: {\_id: "p-1749932574633", userId: "684c10cda9558bd2653327bb",…}
   2. 1: {\_id: "p-1749834800344", userId: "u1",…}
   3. 2: {\_id: "p-1749830516882", userId: "u1",…}