

Take-Home Assignment – Full-Stack Developer

Goal

We want to evaluate both **system design & architecture skills** and the ability to **implement a small MVP**. The goal is not perfect code, but to see structured thinking, prioritization, and the ability to build something end-to-end.

Choose One Scenario

1. **Real-Time Group Chat System**
system that allows multiple users to join a chat room and exchange messages in real time
 2. **Real-time Chat with an AI Assistant**
A user sends chat messages and receives responses in real time from an “AI” engine (you can simulate responses).
-

Assignment Requirements

Part 1 – Design

- High-level architecture diagram (Frontend → Backend → Infra → DB/Queue/Model).
- Short Design Doc (max 2 pages) covering:
 - MVP definition.
 - Core APIs/Endpoints.
 - Sync vs Async flow.

- Scalability bottlenecks.
- Reliability (Retries, DLQ, Idempotency).
- Observability (metrics/logs).
- Security & Privacy.
- Cost awareness.
- Sequence Diagram for one critical path (e.g. Upload or Chat).
- Short Roadmap: MVP → 3 days → 1 week → Production.

Part 2 – Small MVP Code

- **Backend:**
 - Endpoint to create a job (POST).
 - Endpoint to fetch job status (GET).
 - A dummy worker/service that returns a fake “result” after a short delay.
- **Frontend:**
 - A single page to upload a file *or* send a chat message.
 - Displays job/chat status and the final result.

The goal is to demonstrate connecting Frontend ↔ Backend ↔ “Processing”.
Complex features are not required.

Deliverables

- GitHub repo or ZIP with code.
- Design Doc (PDF/Markdown).