HINTRO

BACKEND ENGINEER ASSIGNMENT

Problem Statement

Build a Smart Airport Ride Pooling Backend System that groups passengers into shared cabs while optimizing routes and pricing.

Functional Requirements

- Group passengers into shared cabs.
- Respect luggage and seat constraints.
- Minimize total travel deviation.
- Ensure no passenger exceeds detour tolerance.
- Handle real-time cancellations.
- Support 10,000 concurrent users.
- Handle 100 requests per second.
- Maintain latency under 300ms.

Expected Deliverables

- DSA approach with complexity analysis.
- Low Level Design (class diagram + patterns used).
- High Level Architecture diagram.
- Concurrency handling strategy.
- Database schema and indexing strategy.
- Dynamic pricing formula design.

EXECUTION GUIDE

Mandatory Implementation Requirement

• You MUST implement working backend code. Design-only submissions will not be accepted.
• The system must be runnable locally.
• All required APIs must be fully implemented.
• Concurrency handling must be demonstrated in code.
• Database schema must be implemented with migrations or setup scripts.

Submission Instructions

• Push complete working code to a Git repository.
• Include a detailed README with setup and run instructions.
• Provide API documentation (Swagger/OpenAPI/Postman).
• Mention tech stack and assumptions clearly.
• Include sample test data.
• Document algorithm complexity where applicable.

Evaluation Focus

• Correctness of implementation.
• Database modeling and indexing.
• Concurrency safety.
• Performance considerations.
• Clean architecture and modularity.
• Testability and maintainability.