Backend Developer Challenge: Django + PostgreSQL

Overview

This challenge focuses on backend fundamentals and decision-making. You will design a small system with two tables and expose them through a RESTful API. The goal is to evaluate how you design, not to follow strict instructions.

Database Structure

1️⃣ User Table

- **name** (Required)
- **email** (Required)
- *Other fields decided by developer*

2️⃣ Subscription Table

- **status** (Required)
- **user_id** (Required - FK to User)
- **plan_cost** (Required)
- **start_date** (Required)
- *Other fields decided by developer*

3️⃣ Feature Table

- **name** (Required - e.g., "SMS Notifications", "Priority Support")
- **description** (Optional)
- *Other fields decided by developer*

Relationship Requirement

- **One User → Many Subscriptions**

Relationship: Subscription ↔ Feature (Many-to-Many)

- A subscription can include multiple features
- A feature can belong to multiple subscriptions
- Requires a junction/pivot table (e.g., SubscriptionFeature)

Database Requirements

- PostgreSQL version 16
- Utilize generate_series or similar logic to rapidly construct a dataset of 500,000 randomized rows

RESTful API

- Full CRUD for all tables
- OpenAPI / Swagger documentation
- Developer-led design decisions following REST API best practices with focus on performance and business scalability

Analytics & Performance Endpoint

Create a dedicated analytics endpoint (e.g., /api/analytics/) designed to deliver high-performance insights from large datasets. This endpoint must demonstrate efficient query optimization techniques to ensure sub-second response times even when processing 500,000+ records, showcasing your ability to handle real-world production scenarios where database performance is critical.

Required Metrics in Response:

- Total Recurring Revenue: The sum of plan_cost for all Active subscriptions.
- Average Subscription Cost: The average plan_cost across all subscriptions.
- Monthly Revenue History: A list showing the total revenue generated per month for the last 12 months (grouped by start_date).
- Top Users: The top 5 users who have the highest total subscription value.

Evaluation Focus

- Database design choices
- Relationship modeling
- API structure and clarity
- Overall engineering quality