Coding Task: Backend Development for a Food Delivery App

Objective:

Develop a REST API backend for a food delivery app using Node.js. The primary focus is on a dynamic pricing module to calculate the total cost of food delivery based on various factors.

Key Features:

Dynamic Pricing Module with REST API: Create an API to calculate delivery costs for different types of food items across various zones based on the distance and item type.

- The API should dynamically determine pricing based on:
  - Base Distance and Price: For example, a base distance of 5 km with a base price of 10 euros.
  - Per Km Price: For distances beyond the base, e.g., 1.5 EUR/km for perishable items and 1 EUR/km for non-perishable items.
- The API response should include the total price for the delivery of the specified food items in the given zone for the particular organization.

Database Relations:

- Organization: May have multiple pricing structures based on the zone and item type.
- Item: Identified by type (perishable, non-perishable) and description.
- Pricing: Linked to an organization and item, includes zone-specific base pricing and per km rates.

Database Schema: Use PostgreSQL. Design the schema with validations for API input payloads.

Technical Specifications:

- API Request and Response Format:
  - Request: `{ zone: "central", organization_id: "005", total_distance: 12, item_type: "perishable" }`
  - Response: `{ total_price: 20.5 }`
- Database Models:
  - Organization: `{ id, name }`
  - Item: `{ id, type, description }`
  - Pricing: `{ organization_id, item_id, zone, base_distance_in_km: 5, km_price: 1.5/1, fix_price: 10 }`
- Price Calculation: Implement as a service object. Prices should be in cents to avoid decimal issues.
- Coding Standards: Follow a recognized linting standard (e.g., Airbnb's style guide).

Deliverables:

Deploy application on render.com with a proper swagger page.
Source Code: Complete with service objects, controllers, and models.
API Documentation: Detailing endpoints, request/response formats, and error handling.
Test Suite: Covering major functionalities and edge cases.
Setup Guide: Instructions for setting up the project and database locally.

Evaluation Criteria:

- Correctness and efficiency of the delivery pricing calculation logic.
- Code quality, organization, and adherence to the specified linting standard.
- Robustness of error handling and input validation.
- Completeness and clarity of documentation and tests.

This task is designed to assess the candidate's ability to create a robust and scalable backend system, with a focus on dynamic pricing calculation, database design, and adherence to coding standards.

Submission:
Fill in the assignment submission form mentioned below with appropriate details.
https://forms.office.com/r/FNcY9nLj6q

In case of any query please reach us at:
siddesh@vigaet.com
chinmay.p@vigaet.com
priyadarshini.nataraj@vigaet.com