

# **Backend Development Internship**

### **Problem statement**

An online billing system is essential for businesses to manage their invoicing, payments, and financial transactions efficiently. Your task is to develop a Node.js server for a billing system that provides seamless functionality and a user-friendly experience.

## Instructions

Create a Node.js server to handle billing operations for a company. The server should expose the features via APIs based on REST principles and handle different scenarios that may arise during billing processes.

#### Context

- The company offers various products and services, each with its own pricing.
- A user can create his account, add/remove items to/from their cart, and view his total bill during checkout.
- Integrate tax calculation based on the price range of the product using the following rules:
  - Apply Tax <u>PA</u> if the price range of the product is greater than 1000 and less than or equal to 5000. The tax percentage should be 12% of the price.
  - Apply Tax <u>PB</u> if the price of the product is above 5000. The tax percentage should be 18% of the price.
  - Apply Tax **PC** to all products with a flat tax amount of 200.
- Integrate tax calculation based on the price range of the services using the following rules:
  - Apply Tax <u>SA</u> if the price range of the service is greater than 1000 and less than or equal to 8000. The tax percentage should be 10% of the price.
  - Apply Tax <u>SB</u> if the price of the service is above 8000. The tax percentage should be 15% of the price.
  - Apply Tax <u>SC</u> to all services with a flat tax amount of 100.
- The tax should be applied to each product/service individually, not on the entire bill.

#### Must Haves

The server should have endpoints to enable the following functionalities for users:

- Create an account.
- Fetch all products and services information with their prices.
- Add a product or service to the cart.
- Remove a product or service from the cart.
- Clear the cart.
- View total bill (should include price, quantity, and tax on each item as well as total value of selected items)
- Confirm the order

#### Good to have

- Additional API for admin to see all the orders.
- Appropriate test cases to simulate practical scenarios that you would want to test the system for.

**Note:** Use an appropriate database to handle the problem.

# **Submission** (send it to geetika@plotline.so)

- A Postman collection for the APIs and relevant credentials to test out
- Link to the code hosted in a private repository on Github/Gitlab/Bitbucket with (Please provide access to amit@plotline.so and rajat@plotline.so for evaluation)

#### Your submission will be evaluated on:

- 1. Functionality and robustness
- 2. Project structure
- 3. Code quality (Clarity, Readability, Best practices, Efficiency, etc.)
- 4. Code maintenance through the course of development on Version Control System (commits etc.)

Write back to us on <a href="mailto:qeetika@plotline.so">qeetika@plotline.so</a> in case of any questions/ clarifications.