API Integration

The purpose of this test is to check your knowhow using an API service. You must use the following routes to get the data. Create the integration with Python, PHP or Javascript:

Route	Method	Type	Full Route	Description
/employee	GET	JSON	http://dummy.restapiexample.com/api/v1/employees	Get all employee data
/employee/{id}	GET	JSON	http://dummy.restapiexample.com/api/v1/employee/{id}	Get a single employee data
/create	POST	JSON	http://dummy.restapiexample.com/api/v1/create	Create new record in database

1. How many employees earn more than $300,000.00?
2. Create a record with your name. You can use fake data for the other attributes.
3. What's your user id?

Practical exercise

Task 1: Product Search

Write a function or algorithm to search for a product by name in a list of products. Each product is represented by a dictionary with 'product_id', 'product_name', and 'price'.

Task 2: Cart Total

Write a function or algorithm to calculate the total price of items in a customer's shopping cart. The cart is represented as a list of dictionaries, where each dictionary contains 'product_id' and 'quantity' fields.

Task 3: Discount Calculation

Write a function or algorithm to calculate the total price of items in a shopping cart after applying discounts. Each product in the cart has a 'price' and a 'discount_percentage' field.

Task 4: Top Selling Products

Write a function or algorithm to find the top N best-selling products based on the quantity sold. Given a list of orders (each order containing 'product_id' and 'quantity'), return the list of top N products.

SQL Test

Task 1: Table Creation

Create the following tables for an e-commerce system:

1. Customers
    - Columns: customer_id, first_name, last_name, email, phone
2. Products
    - Columns: product_id, product_name, price, stock_quantity
3. Orders
    - Columns: order_id, customer_id , order_date
4. OrderItems
    - Columns: order_item_id, order_id, product_id, quantity, subtotal

Task 2: CRUD Operations

Write SQL queries for the following operations:

1. Insert a new product with the following details: Product Name: "Laptop", Price: 1000.00, Stock Quantity: 50.
2. Update the stock quantity of the product with product_id = 3 to 75.
3. Delete the order with order_id = 10 and its associated order items.
4. Retrieve the customer's first and last names who placed the order with order_id = 5.
5. Calculate the total revenue generated by each product (sum of subtotals from order items).

Task 3: Stored Procedures

Create a stored procedure that calculates and returns the total revenue generated by a specific customer.

Task 4: Integration

Imagine you have an application with the following requirements:

- Display a list of products and their stock quantities.
- Allow the user to search for products by their name.
- Allow the user to place an order by selecting a customer and adding order items.

For each exercise create a branch on github.Share your own repo when finishing the test for review of the code and the branches. Take in consideration the best practices of gitflow.