**Q: Employee payroll system - Design a class to manage employee payroll, including calculating salaries, taxes, and benefits.**

**Approach to solve:**

1.Understand the problem statement clearly and find out the all the end points need to be created .

2.Figure out the proper datasturctures to manipulate the data in the api

3.Figure out the database design with proper list of tables and appropriate constraints to solve the problem.

**Pseudocode:**

**GET\_EMPLOYEES (HTTP GET method):**

Step 1: Create a database connection using the set\_connection() method. If it is created successfully, execute the SELECT query to fetch all the employee records from the employees table.

Step 2: Fetch the data using fetchall() method.

Step 3: Close all the connections and return the fetched data as a JSON response along with the status code 200.

**GET\_EMPLOYEE\_DETAILS (HTTP GET method):**

Step 1: Extract the required data from the json payload.

Step 2: Create a database connection using the set\_connection() method. If it is created successfully, execute the SELECT query to fetch the employee record with the provided employee\_id.

Step 3: Fetch the data using fetchone() method.

Step 4: If the employee data is not found, raise an EmployeeNotFound exception.

Step 5: Close all the connections and return the fetched data as a JSON response along with the status code 200.

**CREATE\_EMPLOYEE (HTTP POST method):**

Step 1: Extract the required data from the JSON payload.

Step 2: If the data format is not correct or any required field is missing, display a message saying "Missing required field(s)".

Step 3: Create a database connection using the set\_connection() method. If it is created successfully, execute the INSERT query to insert the employee data into the employees table.

Step 4: Commit the transaction and close all the connections.

Step 5: Return a message saying "Employee created successfully" along with the status code 200.

**UPDATE\_EMPLOYEE (HTTP PUT method):**

Step 1: Extract the required data from the JSON payload.

Step 2: If the data format is not correct or any required field is missing, display a message saying "Missing required field(s)".

Step 3: Create a database connection using the set\_connection() method. If it is created successfully, execute the UPDATE query to update the employee record with the provided employee\_id.

Step 4: If the employee data is not found, raise an EmployeeNotFound exception.

Step 5: Commit the transaction and close all the connections.

Step 6: Return a message saying "Employee updated successfully" along with the status code 200.

**DELETE\_EMPLOYEE (HTTP DELETE method):**

Step 1: Create a database connection using the set\_connection() method. If it is created successfully, execute the SELECT query to fetch the employee record with the provided employee\_id.

Step 2: If the employee data is not found, raise an EmployeeNotFound exception.

Step 3: Execute the DELETE query to delete the employee record with the provided employee\_id.

Step 4: Commit the transaction and close all the connections.

Step 5: Return a message saying "Employee deleted successfully" along with the status code 200.