**Q:Online shopping cart - Design a class to manage an online shopping cart, including adding items, removing items, and calculating the total price.**

**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:**

**ADD\_TO\_CART (HTTP POST method):**

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

Step 2: If the data format is not correct, display a message saying "Invalid payload".

Step 3: Create a database connection using the set\_connection() method. If it is created successfully, insert the extracted data into the cart table and save changes. If any exception is raised, roll back the transaction and display a message saying "Failed to add item to cart".

Step 4: Update the cart dictionary with the new item and its quantity and price. If the item is already in the cart, increment its quantity and price accordingly. If not, add the item with its quantity and price.

Step 5: Close all the connections and return the added item details in JSON format.

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

Step 1: Create a database connection using the set\_connection() method. If it is created successfully, fetch all the rows from the cart table. If any exception is raised, display a message saying "Failed to get cart contents".

Step 2: Loop through all the rows and update the temp\_cart dictionary with the items, their quantity, and price. If the item is already in the temp\_cart, increment its quantity and price accordingly. If not, add the item with its quantity and price.

Step 3: Close all the connections and return the temp\_cart details in JSON format.

**REMOVE\_FROM\_CART (HTTP DELETE method):**

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

Step 2: If the data format is not correct, display a message saying "Invalid payload".

Step 3: Create a database connection using the set\_connection() method. If it is created successfully, update the quantity of the item in the cart table and save changes. If any exception is raised, roll back the transaction and display a message saying "Failed to remove item from cart".

Step 4: Update the cart dictionary with the new item quantity. If the item quantity becomes zero, remove it from the cart.

Step 5: Close all the connections and return a message saying "Item deleted successfully" or an error message if the item is not found or the quantity is not enough.

**GET\_TOTAL\_CART (HTTP GET method):**

Step 1: Create a database connection using the set\_connection() method.

Step 2: Execute the SQL query to get the total price of all items in the cart table.

Step 3: Fetch the result of the query and extract the total price.

Step 4: Close the database connection. Step 5: If there was an error executing the SQL query, display a message saying "Failed to get cart total". Otherwise, return a JSON response with the total price.