Redis

```python
# Writes the top 50 most popular products based on average rating to redis
def update_popular_products_in_redis():
    query = """
    MATCH (p:Product)
    WITH p, p.`Average Rating` AS averageRating
    ORDER BY averageRating DESC
    RETURN p
    LIMIT 50
    """

    with driver.session() as session:
        result = session.run(query)
        items = [dict(record["p"]) for record in result]

    # Store items in Redis as a hashset
    redis_key = "popular_products"

    # Delete the existing hashtable
    r.delete(redis_key)

    # Update Redis with the new data
    redis_data = {str(i): json.dumps(items[i]) for i in range(len(items))}
    r.hmset(redis_key, redis_data)

    return jsonify({"Products": items})
```

```python
# Returns top 50 popular products
def get_products_from_redis():
    redis_key = "popular_products"
    data = get_parsed_hashtable_from_redis(redis_key)
    return jsonify(data)
```

```python
# Add product to cart
def add_cart_product():
    user_id = request.json['user_id']
    product_id = request.json['product_id']
    product_data = request.json['product_data']

    # Convert product_data dictionary to JSON string
    product_data_json = json.dumps(product_data)

    # Set the Hash Set in Redis
    r.hset('Cart:' + user_id, product_id, product_data_json)

    return 'Hash Set created successfully'
```

```python
# Show cart based on userID
def get_all_cart_products(userID):
    cart_data = r.hgetall('Cart:' + str(userID))

    order_lines = []
    for key, value in cart_data.items():
        product_id = int(key.decode())
        product_data = json.loads(value.decode())
        order_lines.append({
            "product_id": product_id,
            "product_name": product_data["name"],
            "quantity": product_data["quantity"],
            "price": product_data["price"]
        })

    response_data = order_lines

    return jsonify(response_data)
```