Python code

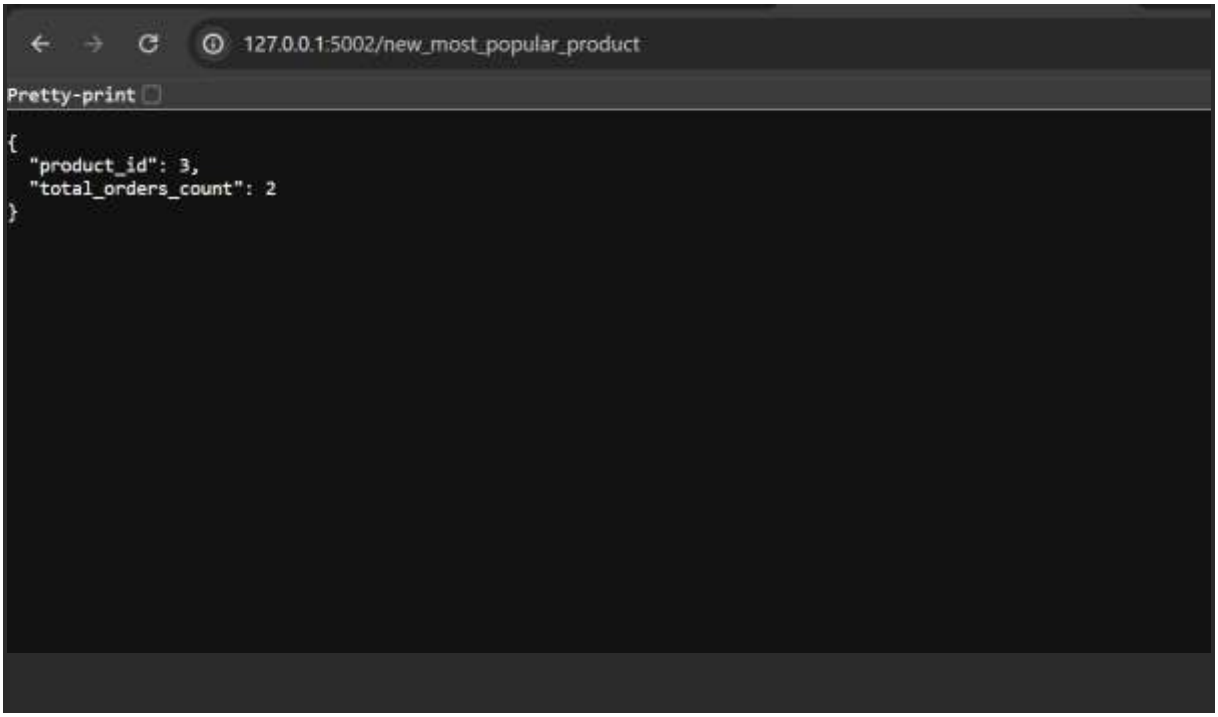
2100031993

```
from flask import Flask, jsonify
from pymongo import MongoClient

app = Flask(__name__)

# Connect to MongoDB
client = MongoClient("mongodb://localhost:27017/")
db = client["new_database"]

# Sample data for Customers table
customers_data = [
    {"customer_id": 1, "first_name": "John", "last_name": "Doe", "email":
"john@example.com", "date_of_birth": "1985-01-15"},
    {"customer_id": 2, "first_name": "Jane", "last_name": "Smith", "email":
"jane@example.com", "date_of_birth": "1990-06-20"}
]

# Sample data for Products table
products_data = [
    {"product_id": 1, "product_name": "Laptop", "price": 1000},
    {"product_id": 2, "product_name": "Smartphone", "price": 600},
    {"product_id": 3, "product_name": "Headphones", "price": 100}
]

# Sample data for Orders table
orders_data = [
    {"order_id": 1, "customer_id": 1, "order_date": "2023-01-10"},
    {"order_id": 2, "customer_id": 2, "order_date": "2023-01-12"}
]

# Sample data for OrderItems table
order_items_data = [
    {"order_item_id": 1, "order_id": 1, "product_id": 1, "quantity": 1},
    {"order_item_id": 2, "order_id": 1, "product_id": 3, "quantity": 2},
    {"order_item_id": 3, "order_id": 2, "product_id": 2, "quantity": 1},
    {"order_item_id": 4, "order_id": 2, "product_id": 3, "quantity": 1}
]

# Insert data into Customers collection
db.Customers.insert_many(customers_data)

# Insert data into Products collection
db.Products.insert_many(products_data)

# Insert data into Orders collection
db.Orders.insert_many(orders_data)

# Insert data into OrderItems collection
db.OrderItems.insert_many(order_items_data)

# Task 1: List all customers
@app.route('/new_customers', methods=['GET'])
def list_customers():
```

```
    customers = list(db.Customers.find({}, {"_id":
0}).distinct("customer_id"))
    return jsonify(customers)

# Task 2: Find all orders placed in January 2023
@app.route('/new_orders/january_2023', methods=['GET'])
def january_2023_orders():
    january_2023_orders = list(db.Orders.find({"order_date": {"$regex":
"^2023-01"}}, {"_id": 0}).distinct("order_id"))
    return jsonify(january_2023_orders)

# Task 3: Get the details of each order, including the customer name and
email
@app.route('/new_order_details', methods=['GET'])
def order_details():
    order_details = db.Orders.aggregate([
        {
            "$lookup": {
                "from": "Customers",
                "localField": "customer_id",
                "foreignField": "customer_id",
                "as": "customer_info"
            }
        },
        {
            "$unwind": "$customer_info"
        },
        {
            "$project": {
                "_id": 0,
                "order_id": 1,
                "order_date": 1,
                "customer_name": {"$concat": ["$customer_info.first_name",
" ", "$customer_info.last_name"]},
                "customer_email": "$customer_info.email"
            }
        }
    ])
    return jsonify(list(order_details))

# Task 4: List the products purchased in a specific order (e.g., OrderID =
1)
@app.route('/new_order_products/<int:order_id>', methods=['GET'])
def order_products(order_id):
    order_products = list(db.OrderItems.find({"order_id": order_id},
{"_id": 0, "order_id": 0}).distinct("product_id"))
    return jsonify(order_products)

# Task 5: Calculate the total amount spent by each customer
@app.route('/new_total_spent_by_customer', methods=['GET'])
def total_spent_by_customer():
    total_spent = db.OrderItems.aggregate([
        {
            "$lookup": {
                "from": "Products",
                "localField": "product_id",
                "foreignField": "product_id",
                "as": "product_info"
            }
        },
        {
```

```
            "$unwind": "$product_info"
        },
        {
            "$group": {
                "_id": "$customer_id",
                "total_spent": {"$sum": {"$multiply":
["$product_info.price", "$quantity"]}}
            }
        }
    ])
    return jsonify(list(total_spent))

# Task 6: Find the most popular product (the one that has been ordered the
most)
@app.route('/new_most_popular_product', methods=['GET'])
def most_popular_product():
    most_popular_product = db.OrderItems.aggregate([
        {
            "$group": {
                "_id": "$product_id",
                "total_orders": {"$addToSet": "$order_id"}
            }
        },
        {
            "$project": {
                "_id": 1,
                "total_orders_count": {"$size": "$total_orders"}
            }
        },
        {
            "$sort": {"total_orders_count": -1}
        },
        {
            "$limit": 1
        }
    ])

    most_popular_product_info = list(most_popular_product)[0]
    most_popular_product_id = most_popular_product_info["_id"]
    total_orders_count = most_popular_product_info["total_orders_count"]

    return jsonify({"product_id": most_popular_product_id,
"total_orders_count": total_orders_count})
# Task 7: Get the total number of orders and the total sales amount for
each month in 2023
@app.route('/new_monthly_sales_2023', methods=['GET'])
def monthly_sales_2023():
    monthly_sales_2023 = db.Orders.aggregate([
        {
            "$match": {
                "order_date": {"$regex": "^2023"}
            }
        },
        {
            "$lookup": {
                "from": "OrderItems",
                "localField": "order_id",
                "foreignField": "order_id",
                "as": "order_items"
            }
        },
```

```
        {
            "$unwind": "$order_items"
        },
        {
            "$lookup": {
                "from": "Products",
                "localField": "order_items.product_id",
                "foreignField": "product_id",
                "as": "product_info"
            }
        },
        {
            "$unwind": "$product_info"
        },
        {
            "$group": {
                "_id": {"$substr": ["$order_date", 0, 7]},
                "total_orders": {"$sum": 1},
                "total_sales": {"$sum": {"$multiply":
["$order_items.quantity", "$product_info.price"]}}
            }
        }
    ])
    return jsonify(list(monthly_sales_2023))

# Task 8: Find customers who have spent more than $1000
@app.route('/new_big_spender_customers', methods=['GET'])
def big_spender_customers():
    big_spender_customers = db.OrderItems.aggregate([
        {
            "$lookup": {
                "from": "Products",
                "localField": "product_id",
                "foreignField": "product_id",
                "as": "product_info"
            }
        },
        {
            "$unwind": "$product_info"
        },
        {
            "$group": {
                "_id": "$customer_id",
                "total_spent": {"$sum": {"$multiply":
["$product_info.price", "$quantity"]}}
            }
        },
        {
            "$match": {
                "total_spent": {"$gt": 1000}
            }
        }
    ])
    return jsonify(list(big_spender_customers))


if __name__ == '__main__':
    app.run(debug=True, port=5002)
```


127.0.0.1:5002/new_most_popular_product
Pretty-print
{
"product_id": 3,
"total_orders_count": 2
}