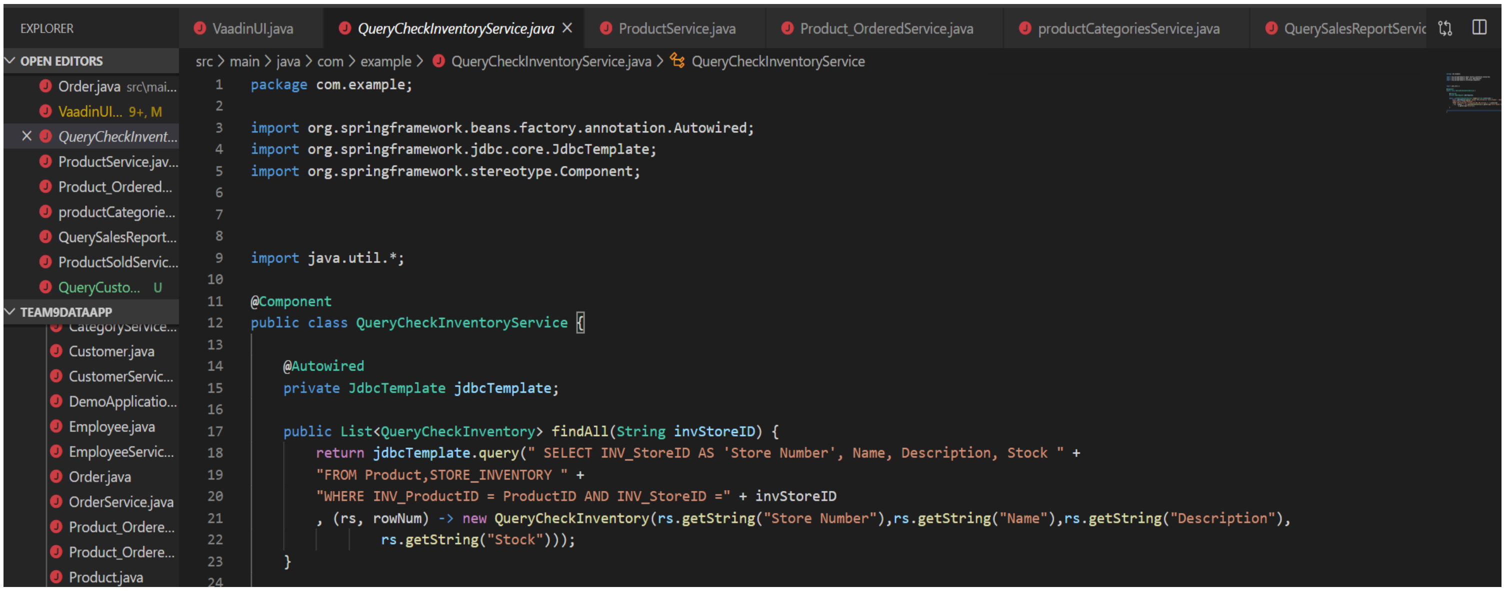
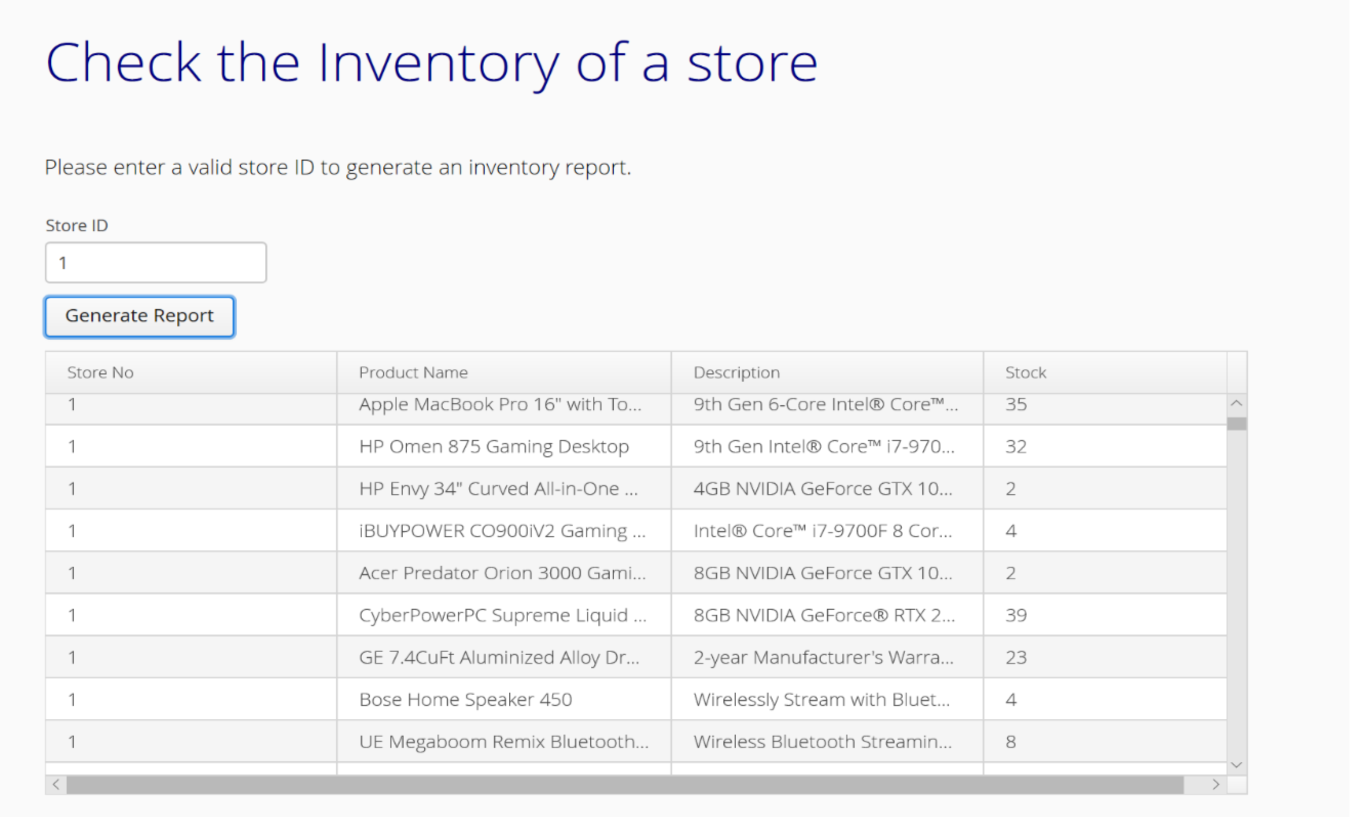
UI Implementation

Check Inventory Query



```java
package com.example;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Component;



import java.util.*;

@Component
public class QueryCheckInventoryService {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    public List<QueryCheckInventory> findAll(String invStoreID) {
        return jdbcTemplate.query(" SELECT INV_StoreID AS 'Store Number', Name, Description, Stock " +
        "FROM Product,STORE_INVENTORY " +
        "WHERE INV_ProductID = ProductID AND INV_StoreID =" + invStoreID
        , (rs, rowNum) -> new QueryCheckInventory(rs.getString("Store Number"),rs.getString("Name"),rs.getString("Description"),
                rs.getString("Stock")));
    }
```



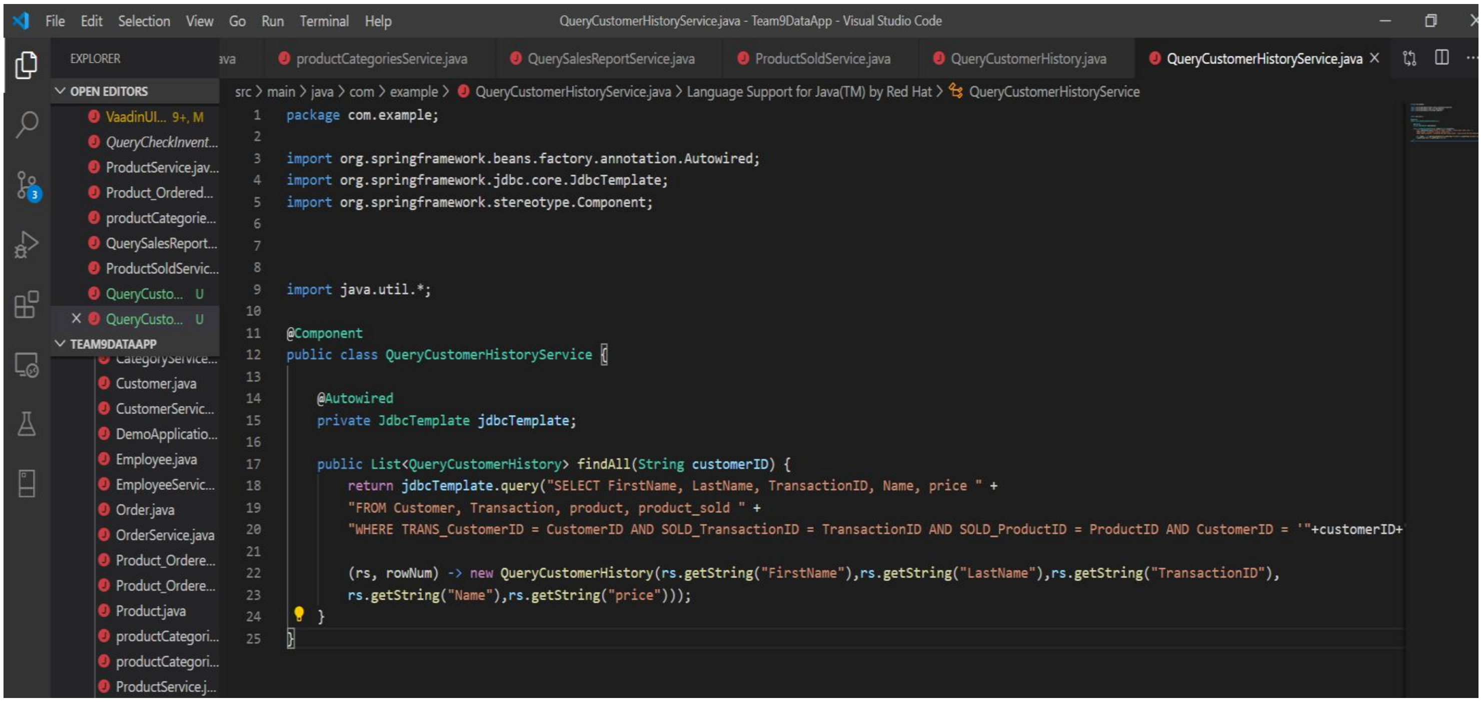
Check the Inventory of a store

Please enter a valid store ID to generate an inventory report.

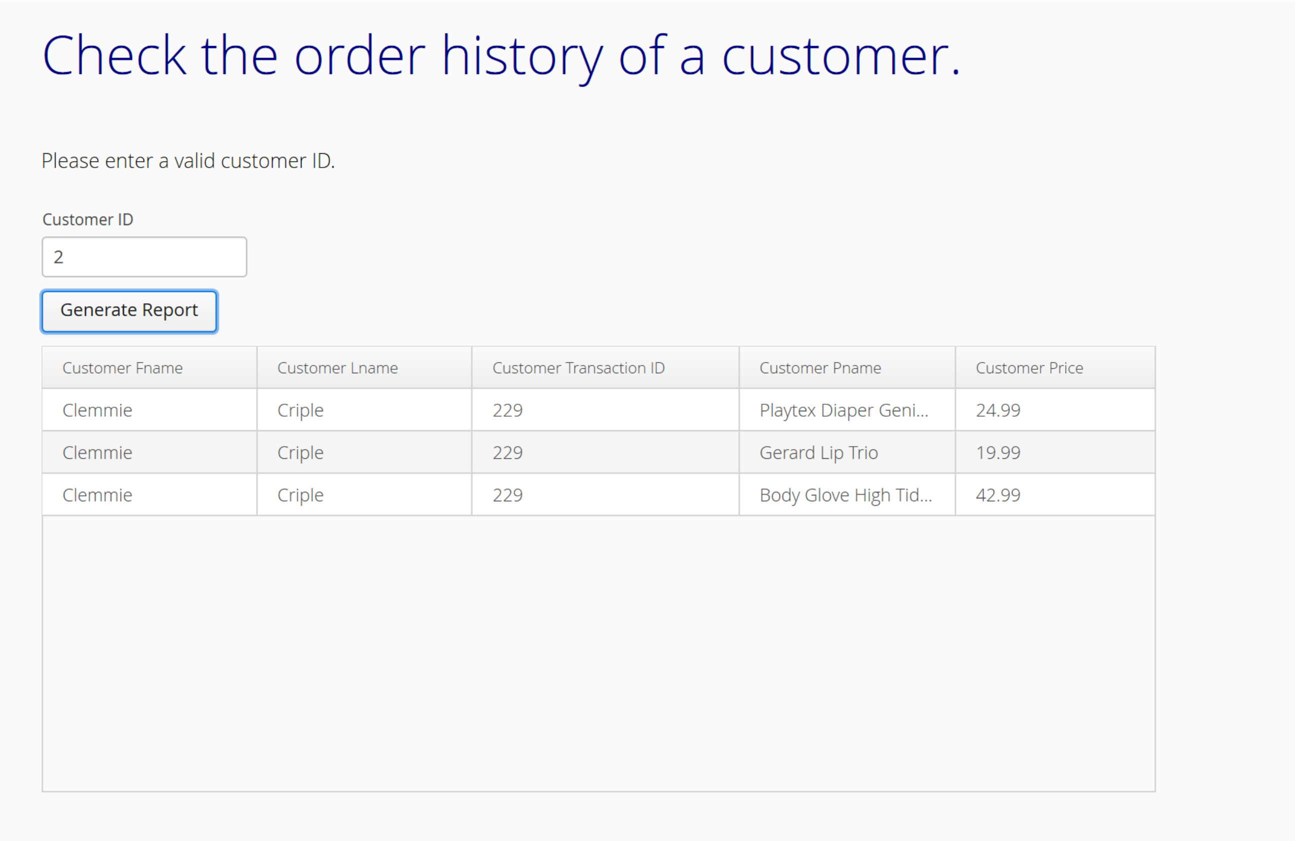
Store ID: 1

Store No	Product Name	Description	Stock
1	Apple MacBook Pro 16" with To...	9th Gen 6-Core Intel® Core™...	35
1	HP Omen 875 Gaming Desktop	9th Gen Intel® Core™ i7-970...	32
1	HP Envy 34" Curved All-in-One ...	4GB NVIDIA GeForce GTX 10...	2
1	iBUYPOWER CO900iV2 Gaming ...	Intel® Core™ i7-9700F 8 Cor...	4
1	Acer Predator Orion 3000 Gami...	8GB NVIDIA GeForce GTX 10...	2
1	CyberPowerPC Supreme Liquid ...	8GB NVIDIA GeForce® RTX 2...	39
1	GE 7.4CuFt Aluminized Alloy Dr...	2-year Manufacturer's Warra...	23
1	Bose Home Speaker 450	Wirelessly Stream with Bluet...	4
1	UE Megaboom Remix Bluetooth...	Wireless Bluetooth Streamin...	8

Customer History Query



```java
package com.example;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Component;



import java.util.*;

@Component
public class QueryCustomerHistoryService {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    public List<QueryCustomerHistory> findAll(String customerID) {
        return jdbcTemplate.query("SELECT FirstName, LastName, TransactionID, Name, price " +
        "FROM Customer, Transaction, product, product_sold " +
        "WHERE TRANS_CustomerID = CustomerID AND SOLD_TransactionID = TransactionID AND SOLD_ProductID = ProductID AND CustomerID = '"+customerID+

        (rs, rowNum) -> new QueryCustomerHistory(rs.getString("FirstName"),rs.getString("LastName"),rs.getString("TransactionID"),
        rs.getString("Name"),rs.getString("price")));
    }
}
```



Check the order history of a customer.

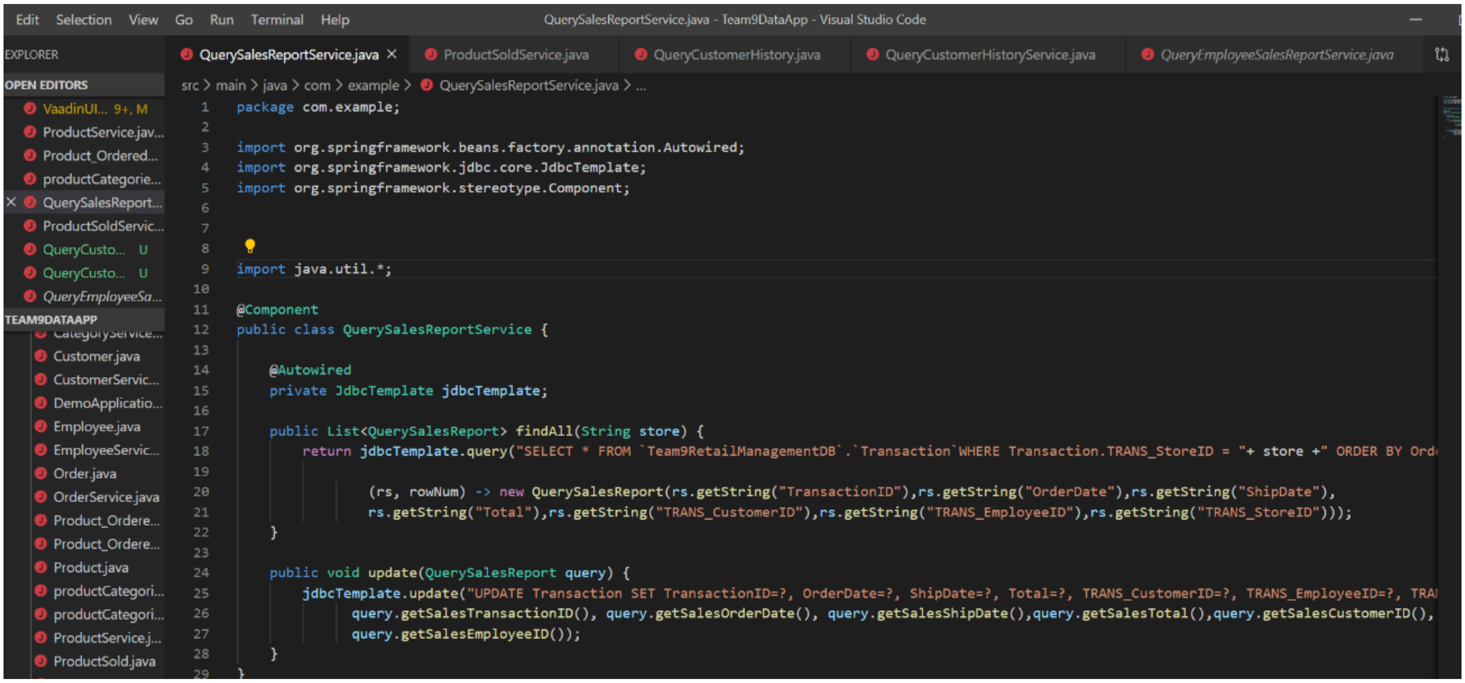
Please enter a valid customer ID.

Customer ID

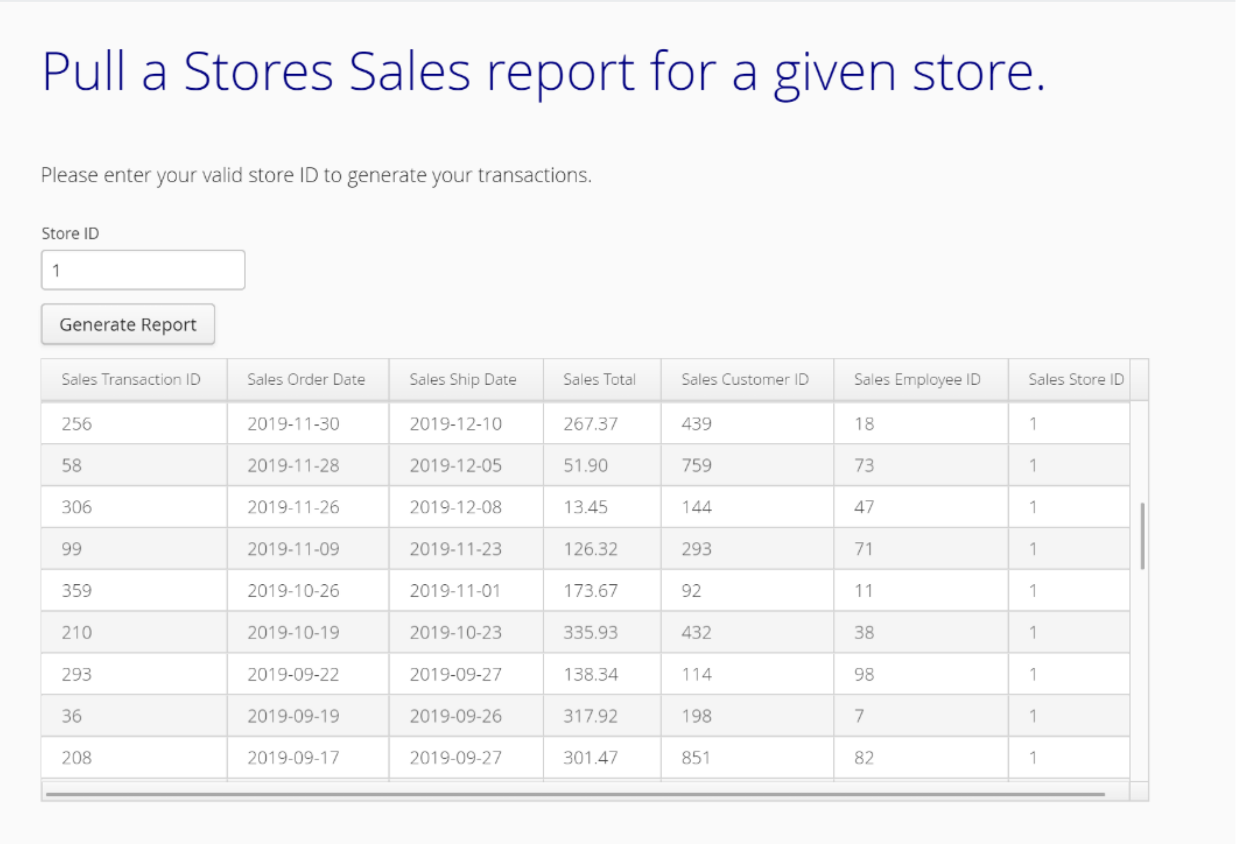
2

Customer Fname	Customer Lname	Customer Transaction ID	Customer Pname	Customer Price
Clemmie	Criple	229	Playtex Diaper Geni...	24.99
Clemmie	Criple	229	Gerard Lip Trio	19.99
Clemmie	Criple	229	Body Glove High Tid...	42.99

Store Transactions Query



```java
package com.example;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Component;



import java.util.*;

@Component
public class QuerySalesReportService {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    public List<QuerySalesReport> findAll(String store) {
        return jdbcTemplate.query("SELECT * FROM `Team9RetailManagementDB`.`Transaction`WHERE Transaction.TRANS_StoreID = "+ store +" ORDER BY Orde

                (rs, rowNum) -> new QuerySalesReport(rs.getString("TransactionID"),rs.getString("OrderDate"),rs.getString("ShipDate"),
                rs.getString("Total"),rs.getString("TRANS_CustomerID"),rs.getString("TRANS_EmployeeID"),rs.getString("TRANS_StoreID")));
    }

    public void update(QuerySalesReport query) {
        jdbcTemplate.update("UPDATE Transaction SET TransactionID=?, OrderDate=?, ShipDate=?, Total=?, TRANS_CustomerID=?, TRANS_EmployeeID=?, TRA
               query.getSalesTransactionID(), query.getSalesOrderDate(), query.getSalesShipDate(),query.getSalesTotal(),query.getSalesCustomerID(),
               query.getSalesEmployeeID());
    }
}
```



Pull a Stores Sales report for a given store.

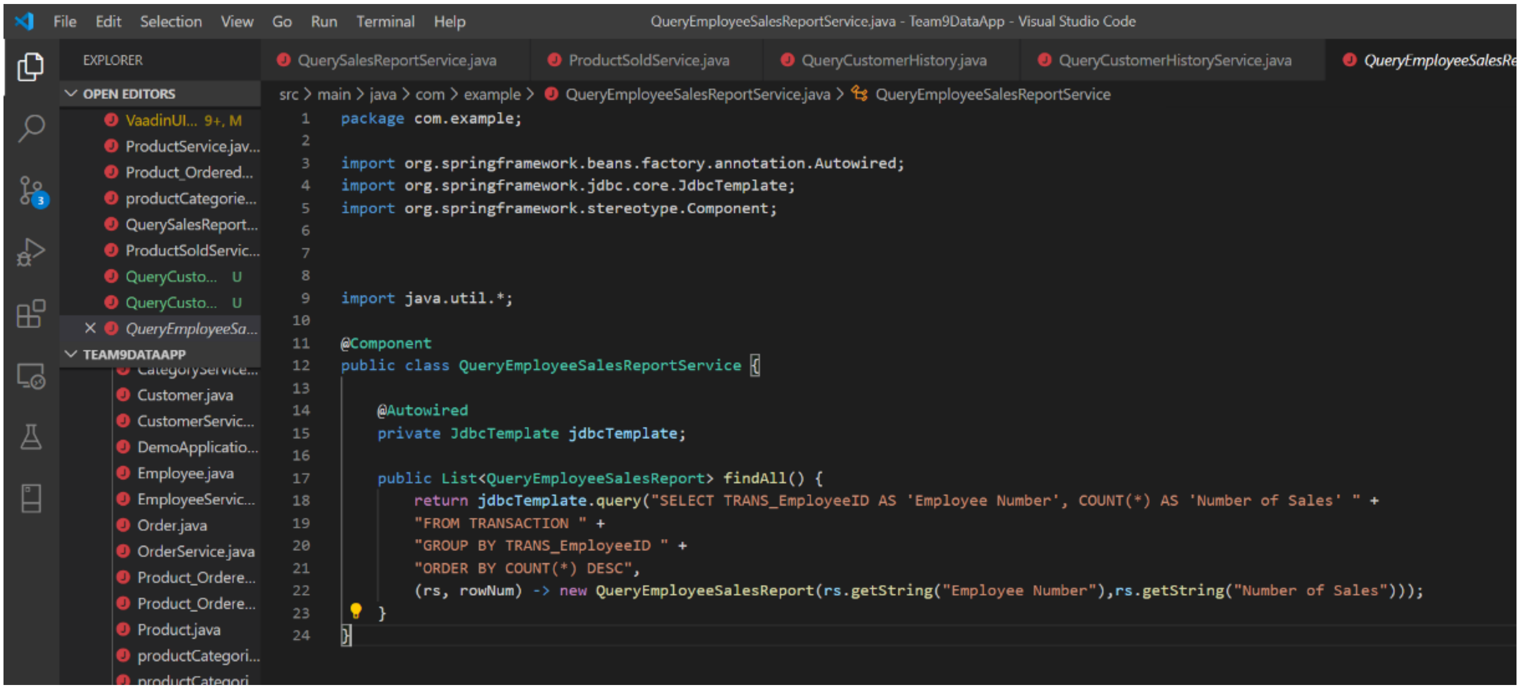
Please enter your valid store ID to generate your transactions.

Store ID

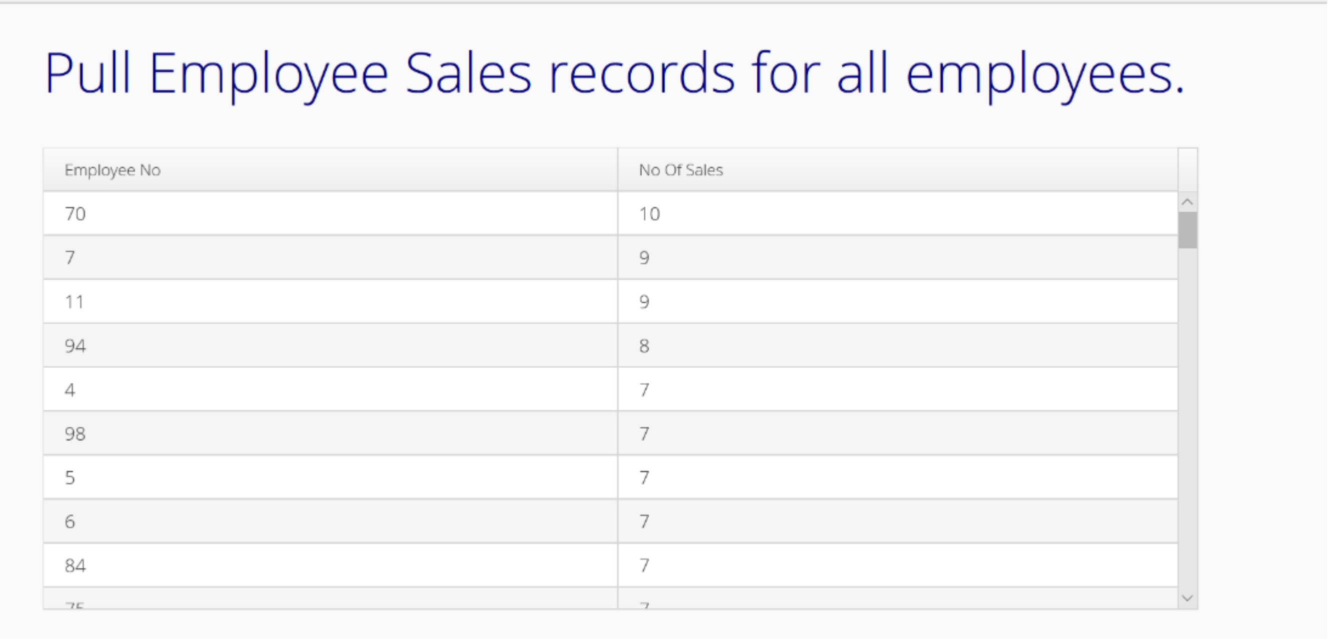
1

Sales Transaction ID	Sales Order Date	Sales Ship Date	Sales Total	Sales Customer ID	Sales Employee ID	Sales Store ID
256	2019-11-30	2019-12-10	267.37	439	18	1
58	2019-11-28	2019-12-05	51.90	759	73	1
306	2019-11-26	2019-12-08	13.45	144	47	1
99	2019-11-09	2019-11-23	126.32	293	71	1
359	2019-10-26	2019-11-01	173.67	92	11	1
210	2019-10-19	2019-10-23	335.93	432	38	1
293	2019-09-22	2019-09-27	138.34	114	98	1
36	2019-09-19	2019-09-26	317.92	198	7	1
208	2019-09-17	2019-09-27	301.47	851	82	1

All Employees Sales Query



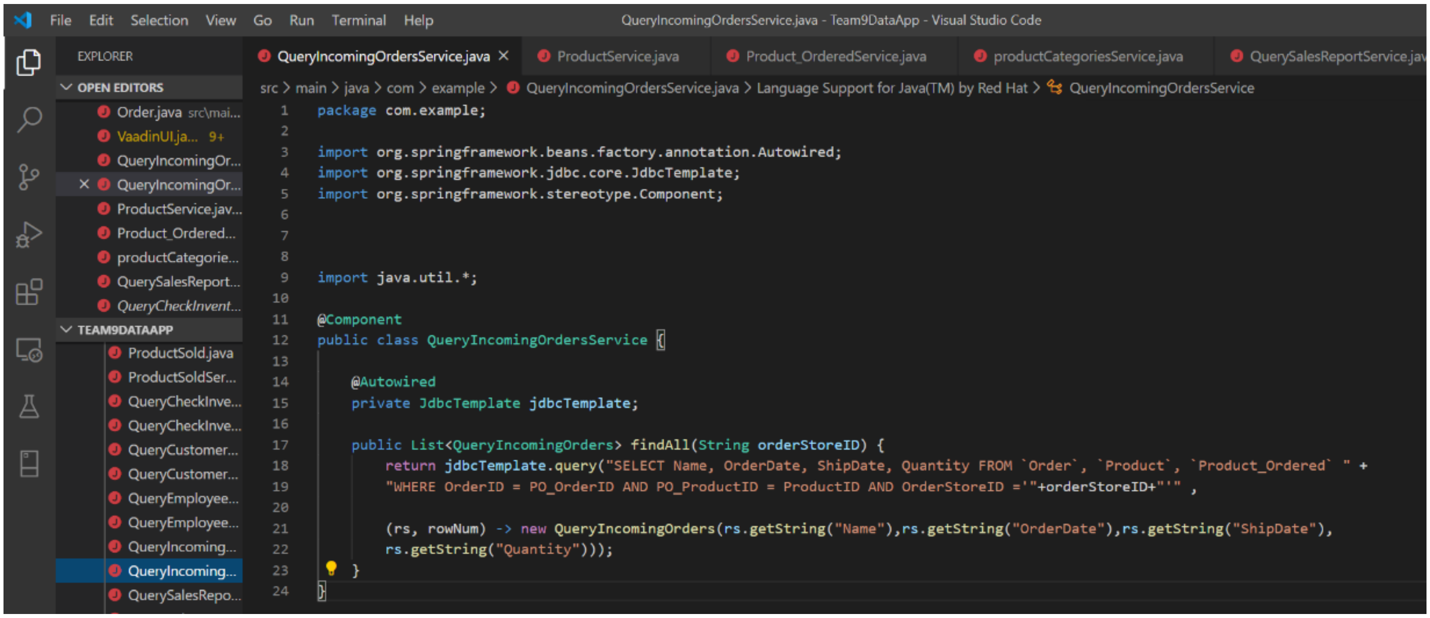
```java
package com.example;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Component;



import java.util.*;

@Component
public class QueryEmployeeSalesReportService {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    public List<QueryEmployeeSalesReport> findAll() {
        return jdbcTemplate.query("SELECT TRANS_EmployeeID AS 'Employee Number', COUNT(*) AS 'Number of Sales' " +
        "FROM TRANSACTION " +
        "GROUP BY TRANS_EmployeeID " +
        "ORDER BY COUNT(*) DESC",
        (rs, rowNum) -> new QueryEmployeeSalesReport(rs.getString("Employee Number"),rs.getString("Number of Sales")));
    }
}
```



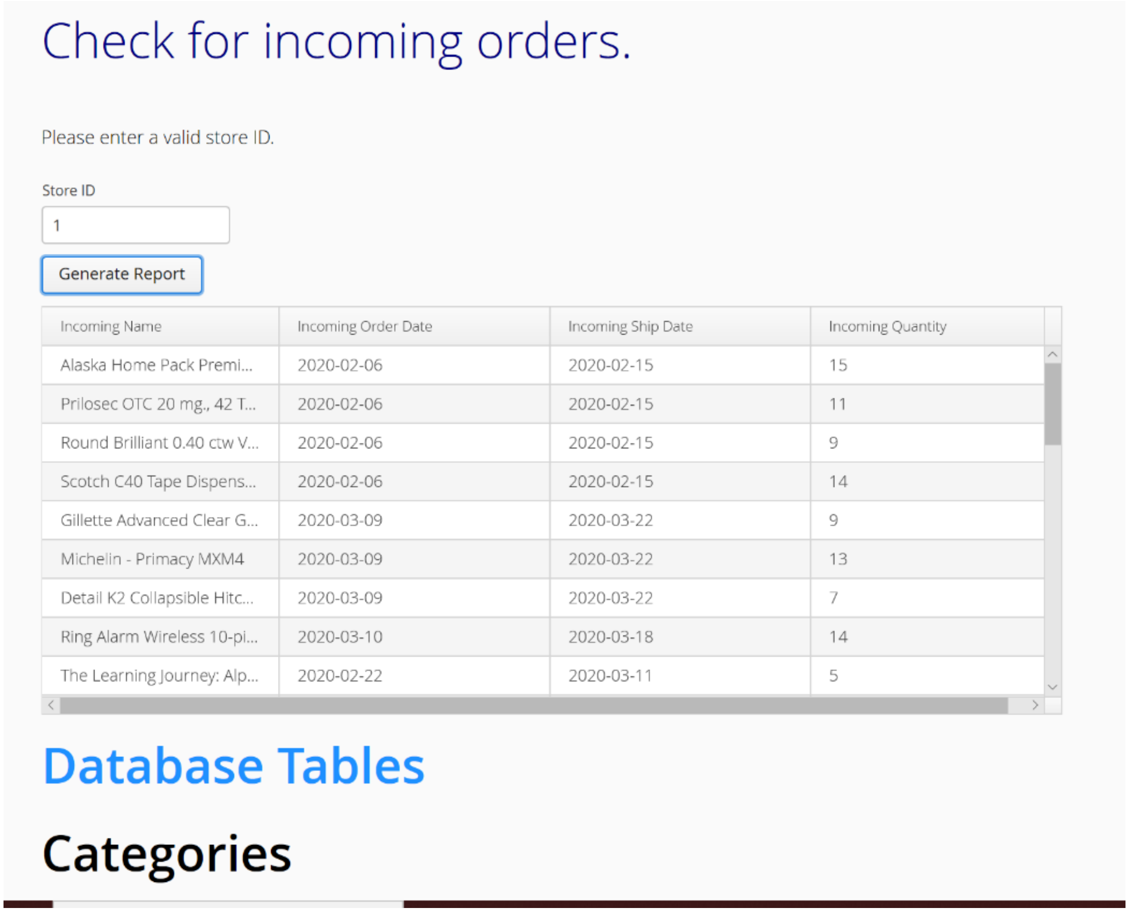
Pull Employee Sales records for all employees.

Employee No	No Of Sales
70	10
7	9
11	9
94	8
4	7
98	7
5	7
6	7
84	7
75	7

Incoming Orders For A Store Query



```java
package com.example;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Component;



import java.util.*;

@Component
public class QueryIncomingOrdersService {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    public List<QueryIncomingOrders> findAll(String orderStoreID) {
        return jdbcTemplate.query("SELECT Name, OrderDate, ShipDate, Quantity FROM `Order`, `Product`, `Product_Ordered` " +
        "WHERE OrderID = PO_OrderID AND PO_ProductID = ProductID AND OrderStoreID ='"+orderStoreID+"'" ,

        (rs, rowNum) -> new QueryIncomingOrders(rs.getString("Name"),rs.getString("OrderDate"),rs.getString("ShipDate"),
        rs.getString("Quantity")));
    }
}
```



Check for incoming orders.

Please enter a valid store ID.

Store ID

1

Incoming Name	Incoming Order Date	Incoming Ship Date	Incoming Quantity
Alaska Home Pack Premi...	2020-02-06	2020-02-15	15
Prilosec OTC 20 mg., 42 T...	2020-02-06	2020-02-15	11
Round Brilliant 0.40 ctw V...	2020-02-06	2020-02-15	9
Scotch C40 Tape Dispens...	2020-02-06	2020-02-15	14
Gillette Advanced Clear G...	2020-03-09	2020-03-22	9
Michelin - Primacy MXM4	2020-03-09	2020-03-22	13
Detail K2 Collapsible Hitc...	2020-03-09	2020-03-22	7
Ring Alarm Wireless 10-pi...	2020-03-10	2020-03-18	14
The Learning Journey: Alp...	2020-02-22	2020-03-11	5

Database Tables

Categories