Perfume Company Database

Yasemin Demirtaş, Talha Ünal, Furkan Örten

171101054, 171101031, 151101059

Abstract

This project is about a perfume company which deals with perfumes business and organizes employees. For representing company’s operation, local web application is designed with PostgreSQL and Python.

* 1. Structure

The database is named “Perfume Database”. Database includes 37 tables with normalized tables. To create database and its connections to web side, Flask and SQLAlchemy is used. To visualize and store the database, pgAdmin4 v4 which is a server is used. Cookies are used.

* 1. Tables

For login operations, 3 tables including admin,customer and employee tables are necessary and for their informations, 3 tables are also in model. For each person’s operations, there are tables in it. Lastly for perfumes and theirs content, database includes tables. All tables with names are represented below.

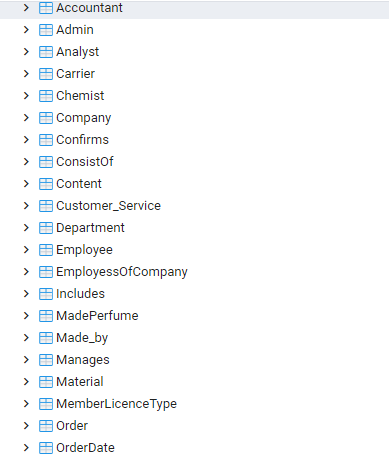
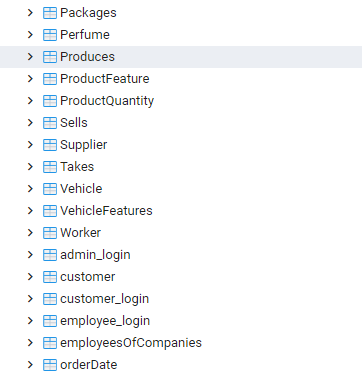
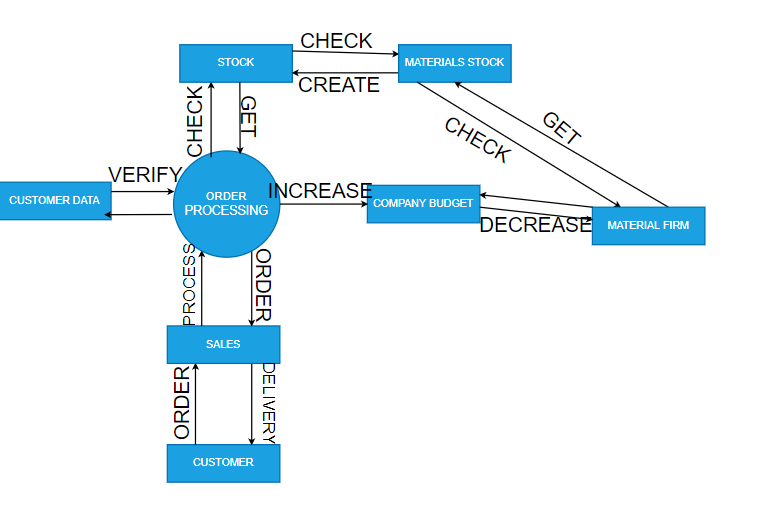
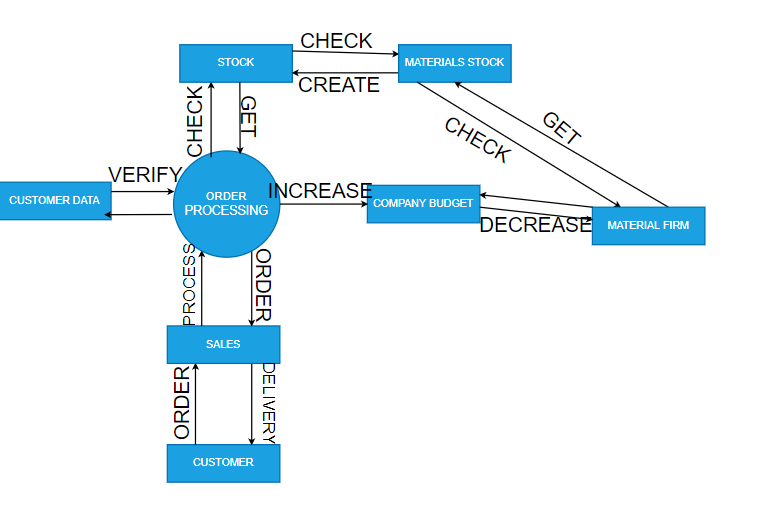


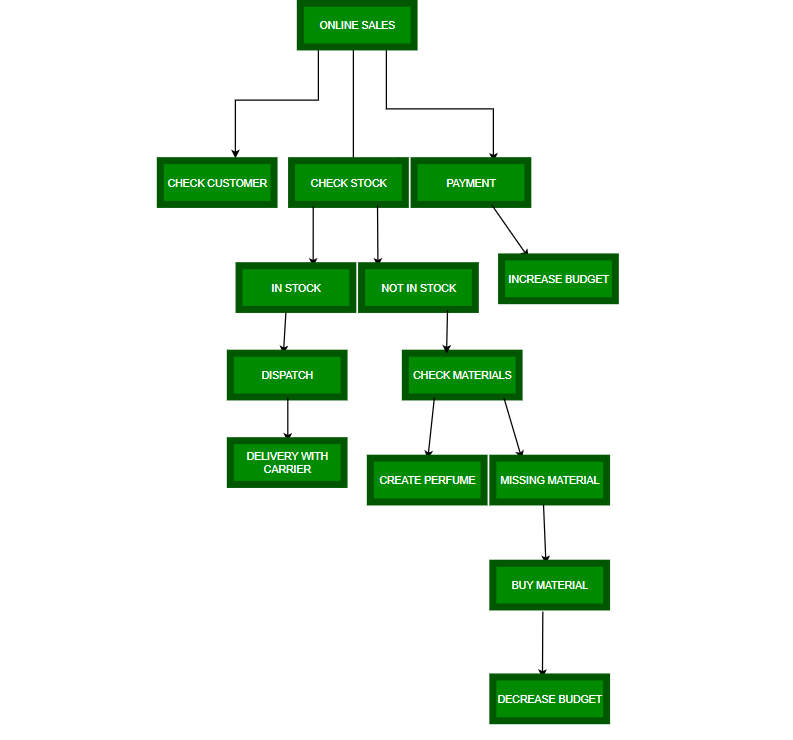
Figure 1.1 :

Names of tables in database.

1. Software Analysis & Design Tools

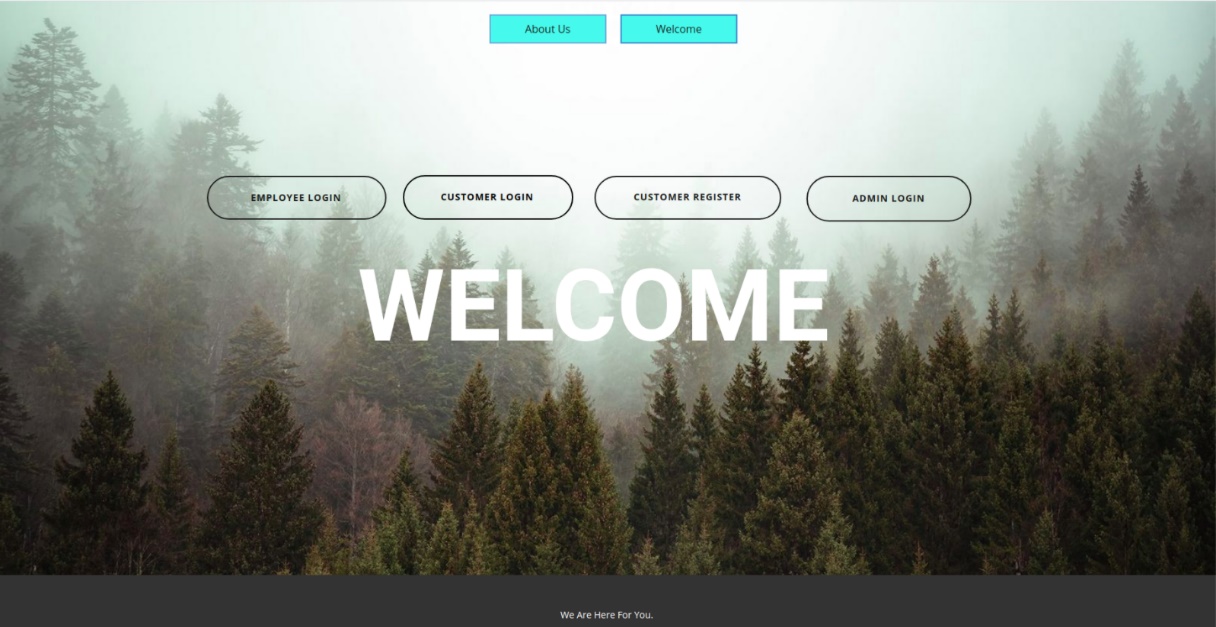
*2.1 Data Flow Diagram*

*Figure 2.1: Level 1 of Data Flow Diagram*

*2.2 Hierarchical Input Process Output*

*Figure 2.2 Hierarchical Input Process Output*

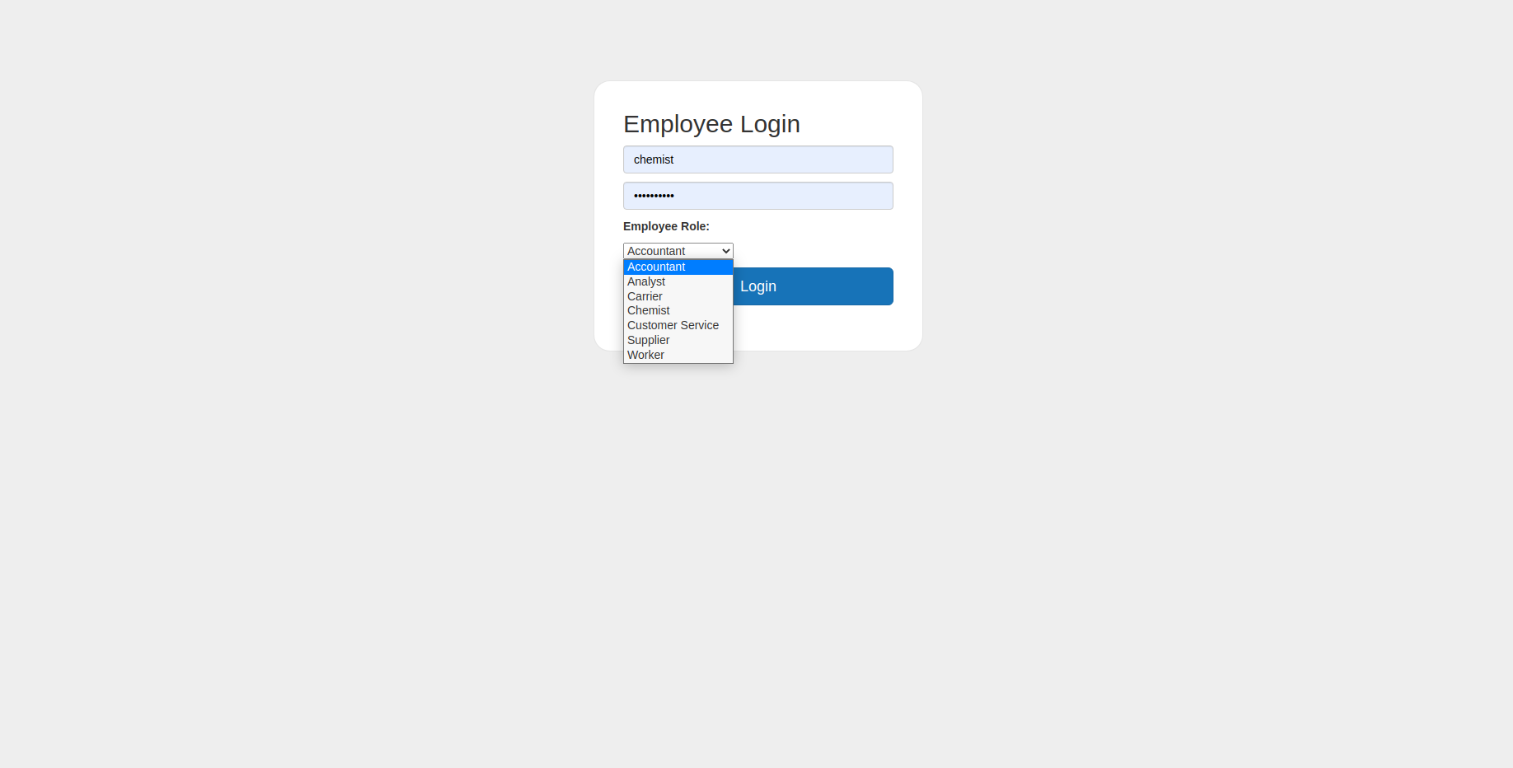
**3) Screenshots of GUI and SQL Codes**



1)

*Figure 3.1: Main page.*

*2)*



*Figure 3.2) Employee Login Page .Employee role must be chosen. SQL Code : select \* from employee\_login where username=? And usertype=?*

*3)* 

Figure 3.3 : Customer Login. SQL Code: SELECT \* FROM customer\_login where Username= …

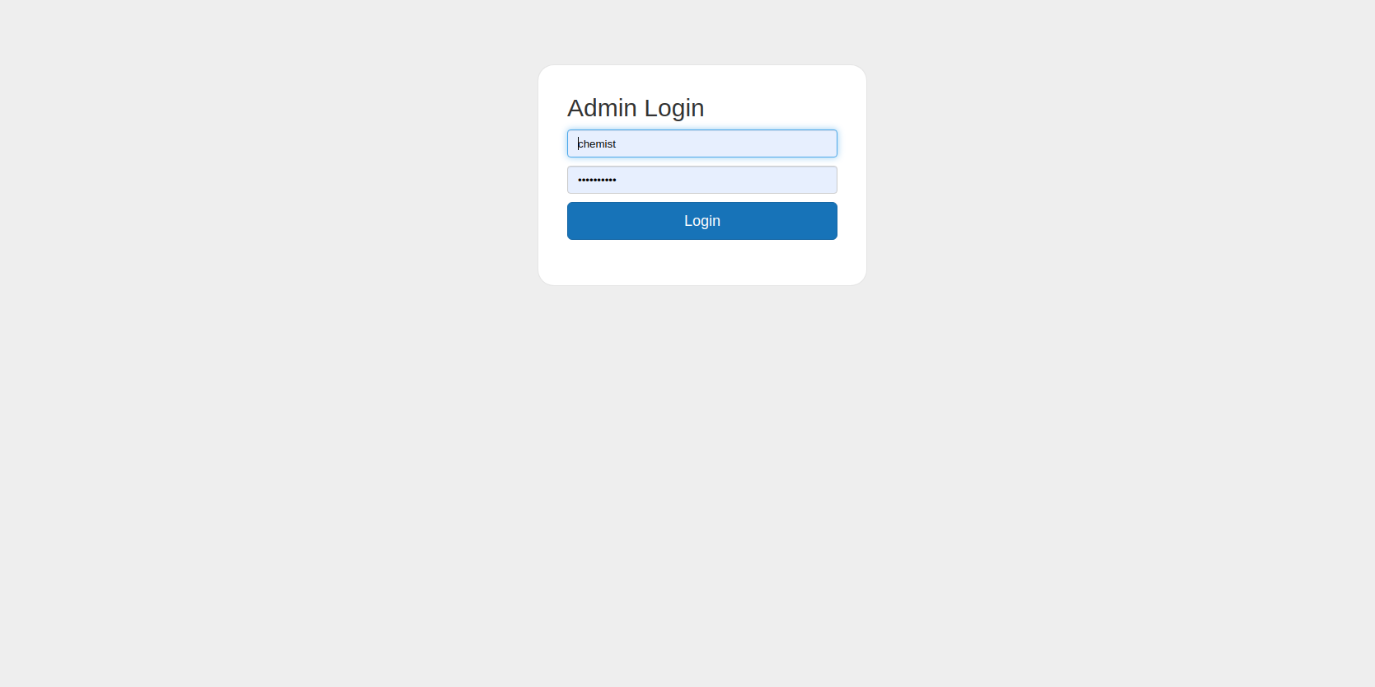
4) 

Figure 3.4: Admin Login. SQL Code: SELECT \* FROM admin\_login

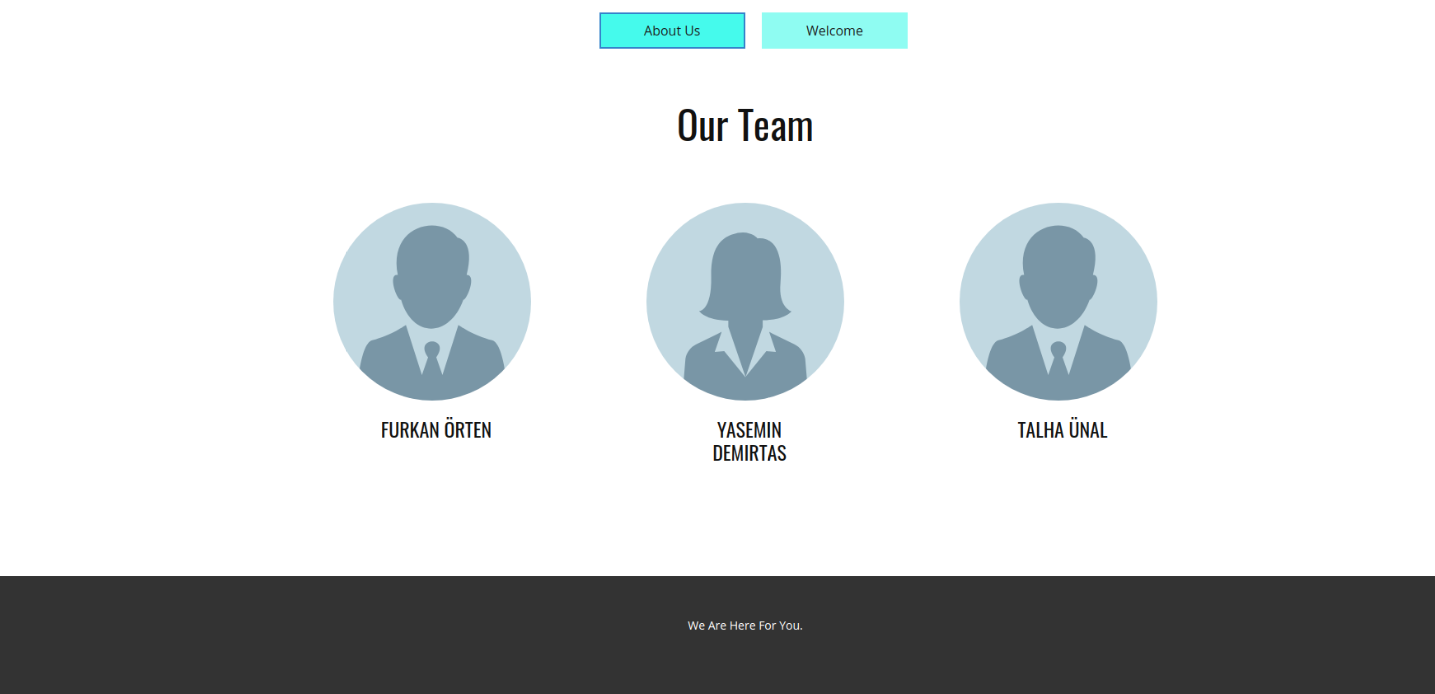
5) 

Figure 3.5 : About us tab at home page.

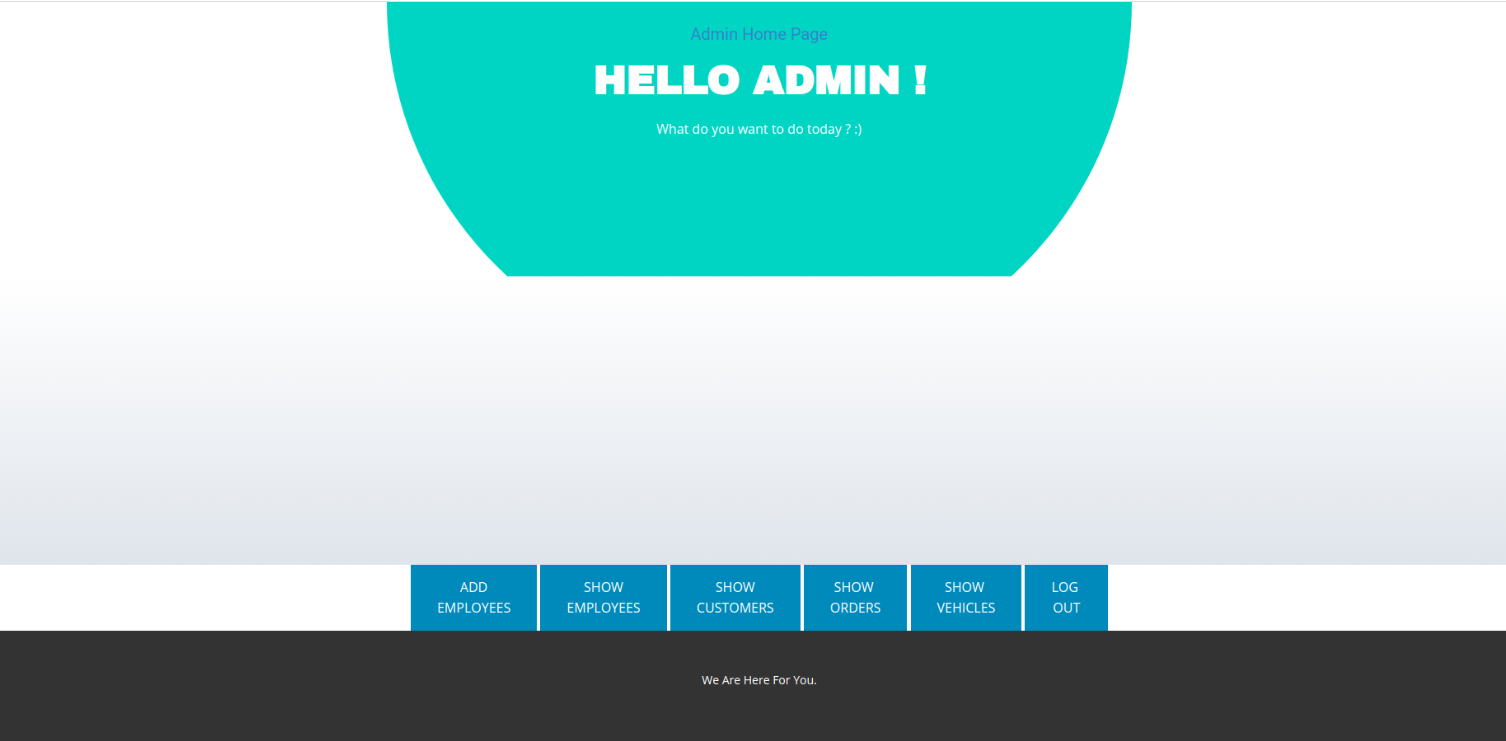
6) 

Figure 3.6: Admin home page. 6 buttons are for operations of company.

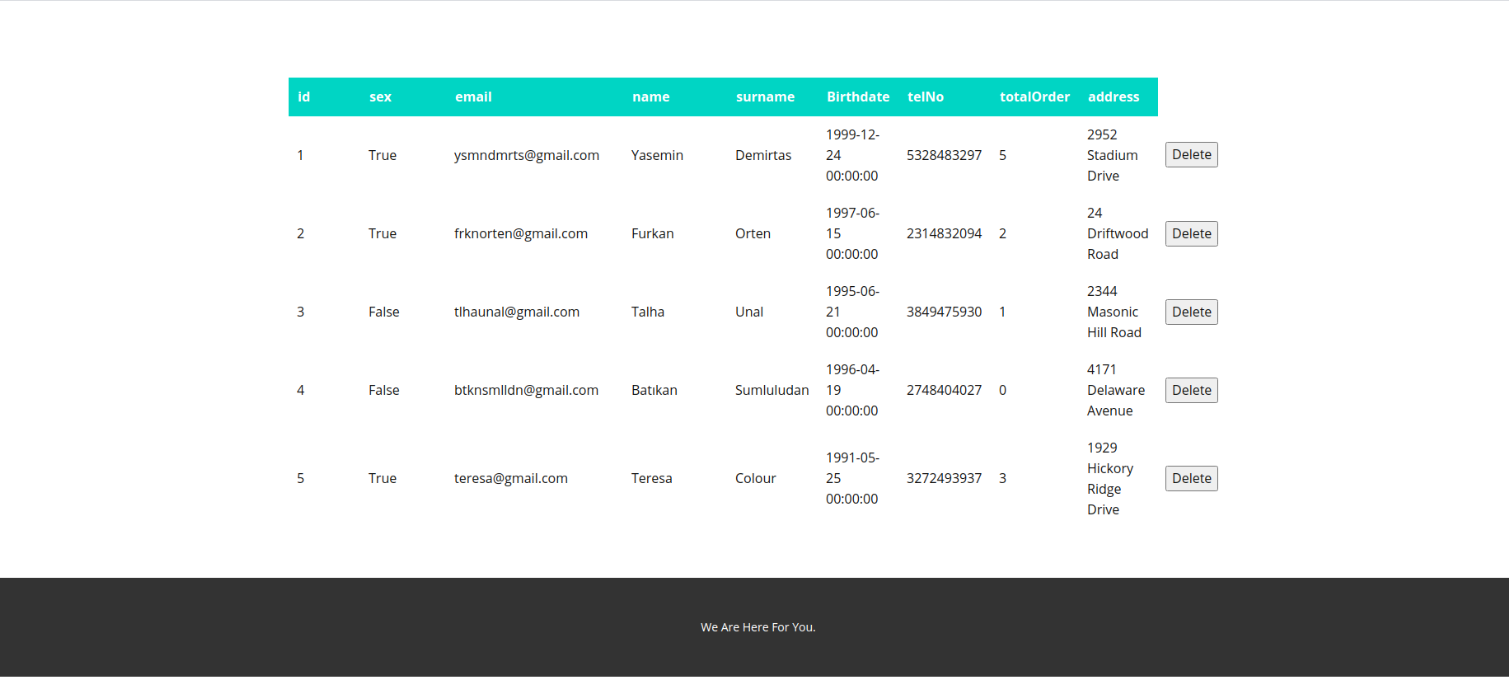
7)

Figure 3.7: List Customers at Admin’s Page. Admin can delete customer but can not edit customer’s information. SQL Code: SELECT \* FROM CUSTOMER // DELETE FROM Customer where id=?

8) 

Figure 3.8: List Employees at Admin’s Page. Admin can edit employee’s information. SQL Code : SELECT \* FROM Employee

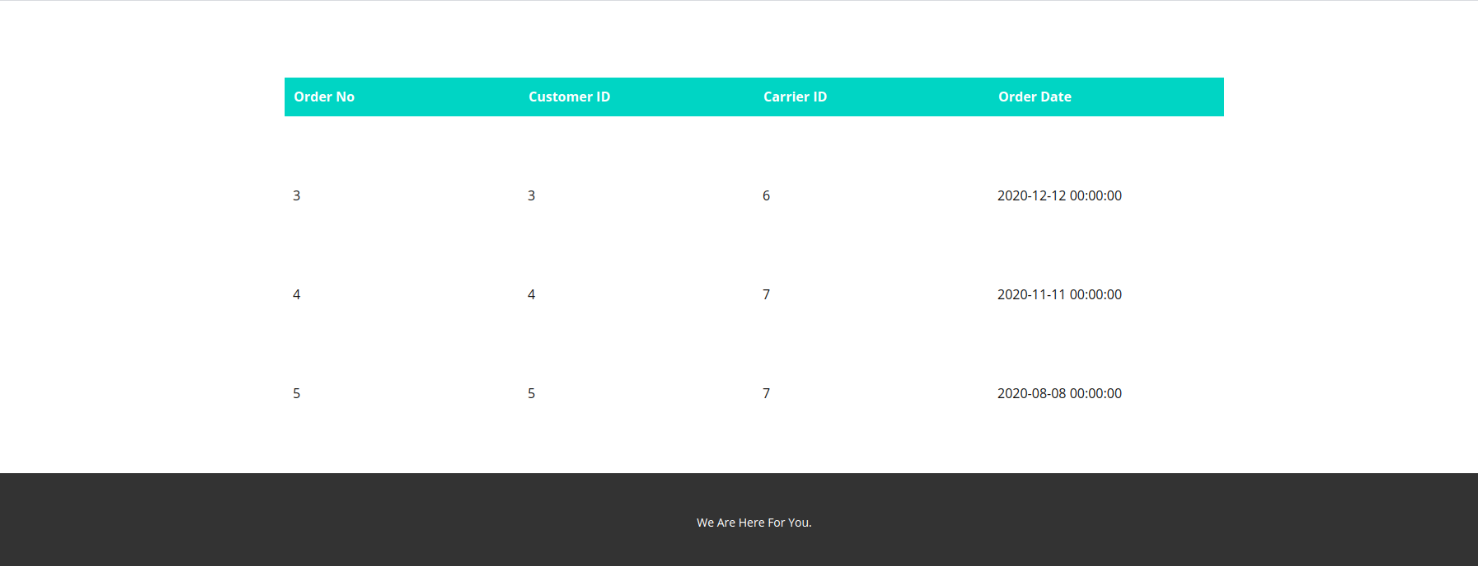
9) 

Figure 3.9: List Orders at Admin’s Page. SQL Code: Select \* From Order

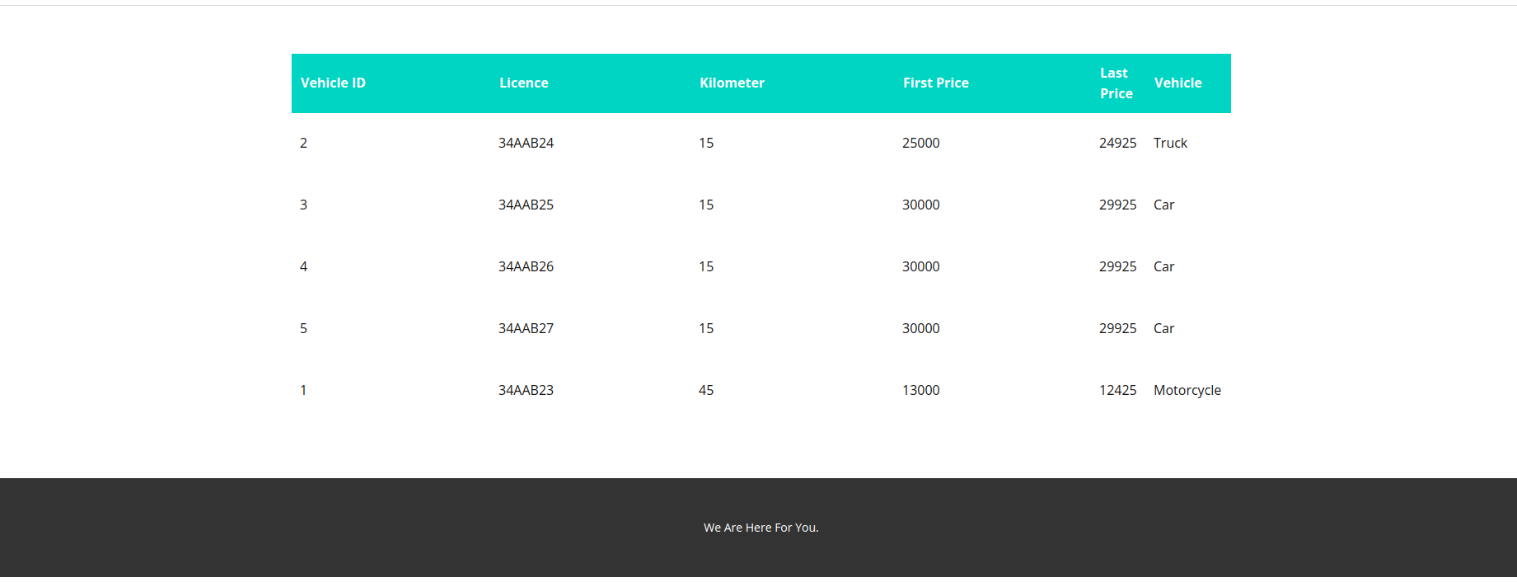
10) 

Figure 3.10: List Vehicles at Admin’s Page. SQL Code: SELECT \* FROM Vehicle

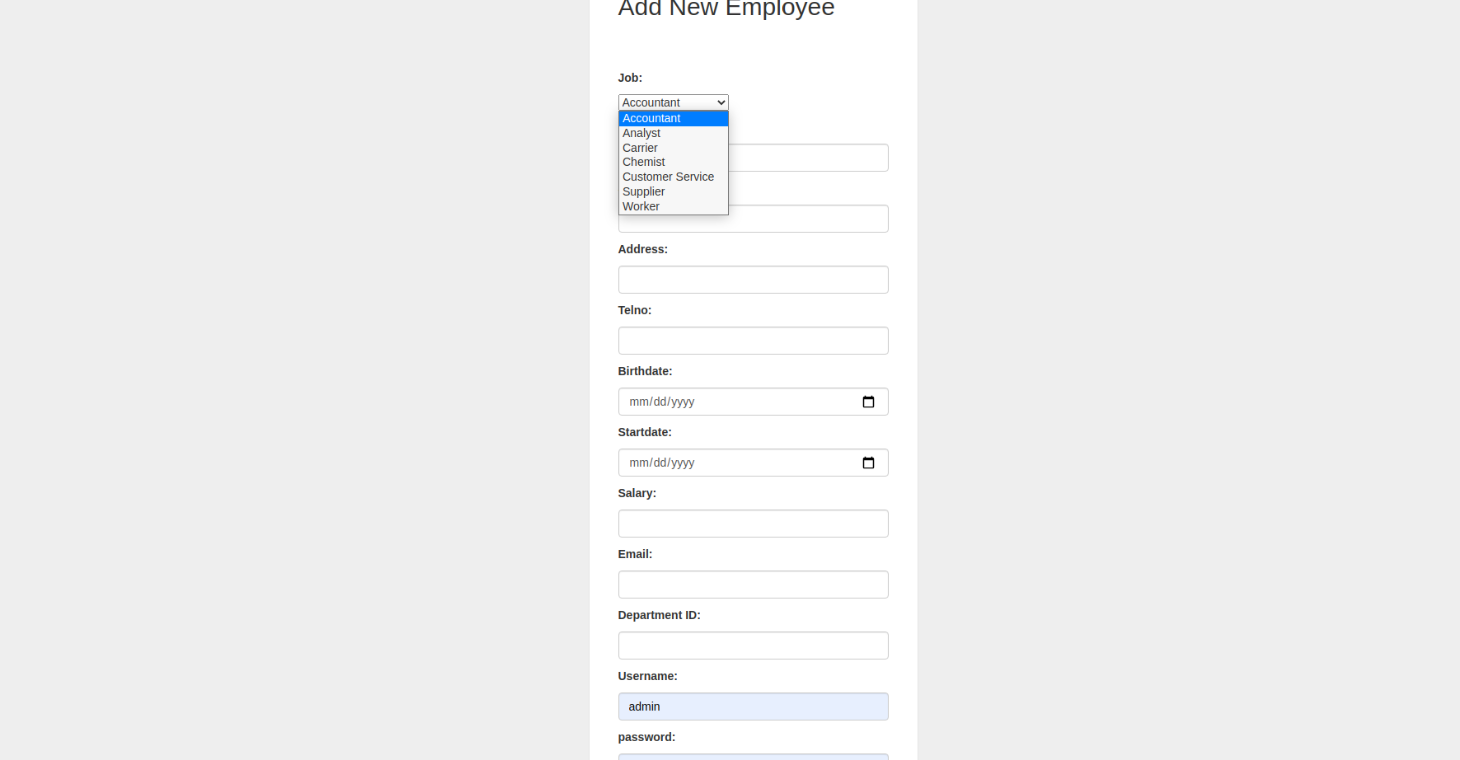
11) 

Figure 3.11: Adding new Employee at Admin’s Page. SQL Code:

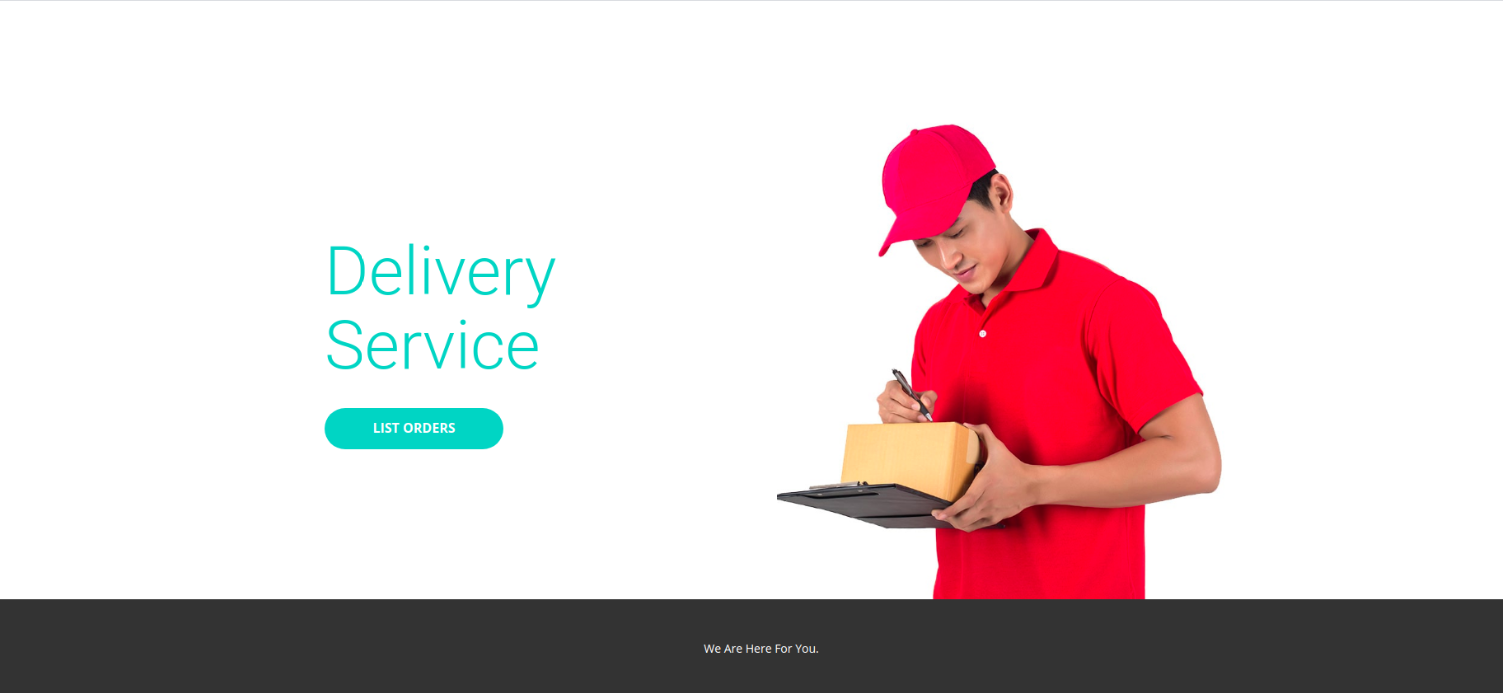
12) 

Figure 3.12: Main Page of Carrier.

13)

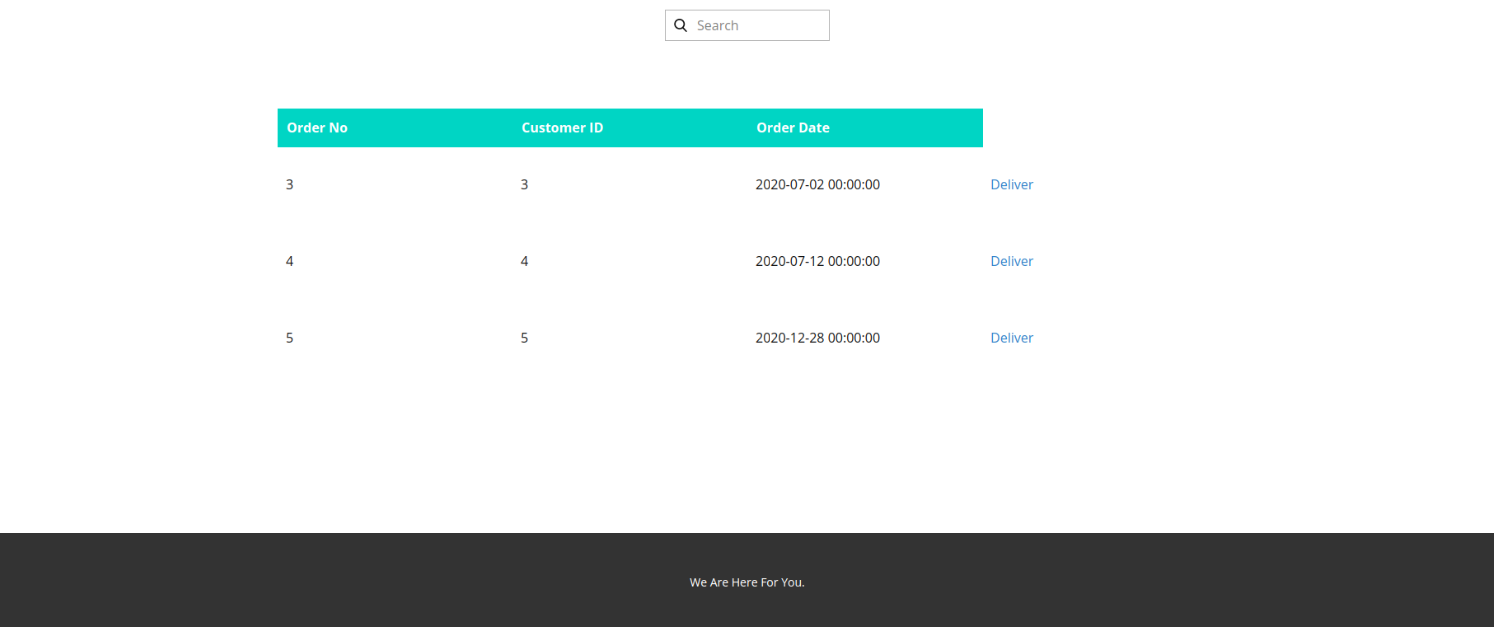


Figure 3.13: Orders for carrier. SQL Code:

SELECT \*

FROM Order ord, OrderDate ord\_d, Carrier c

WHERE ord.order\_no=ord\_d.order\_no AND c.memberID=ord.carrier\_id;

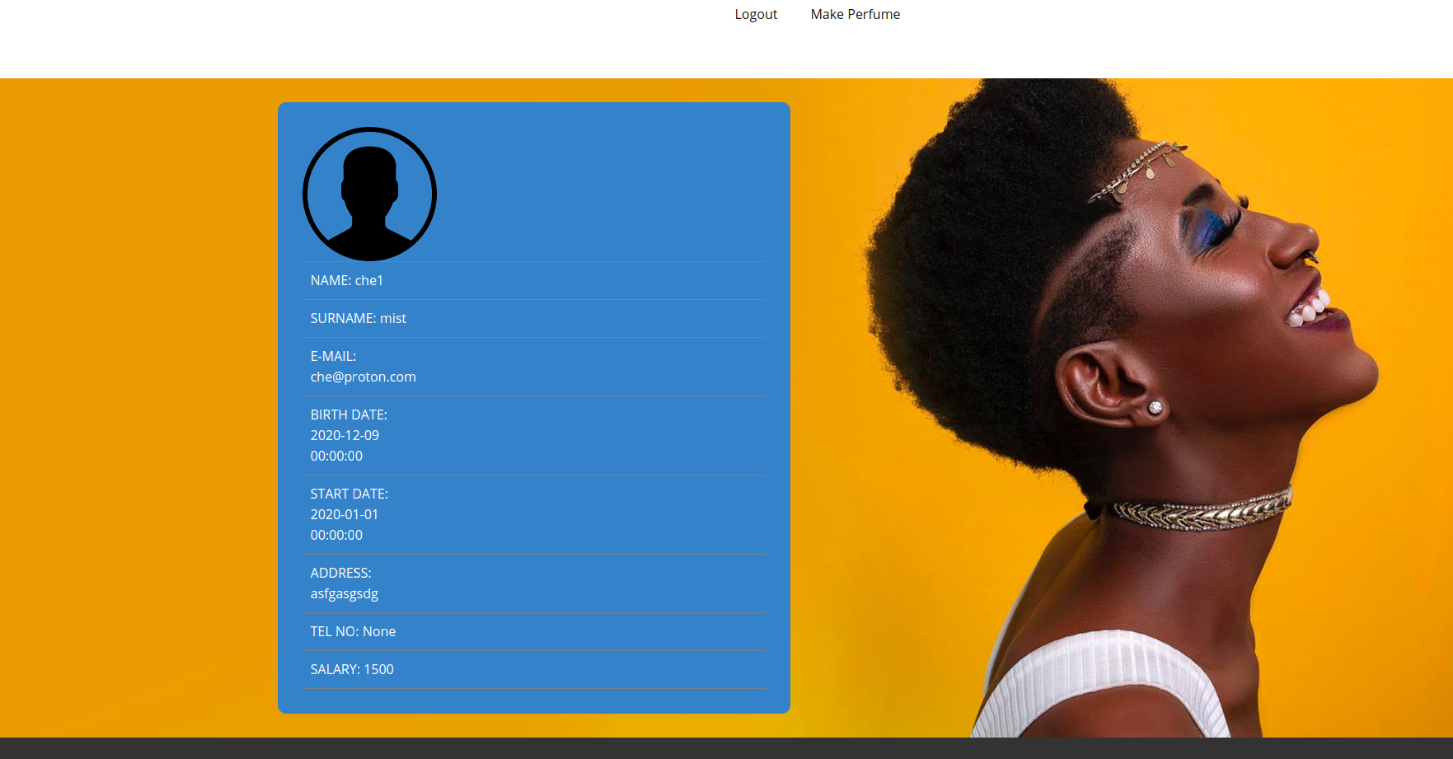
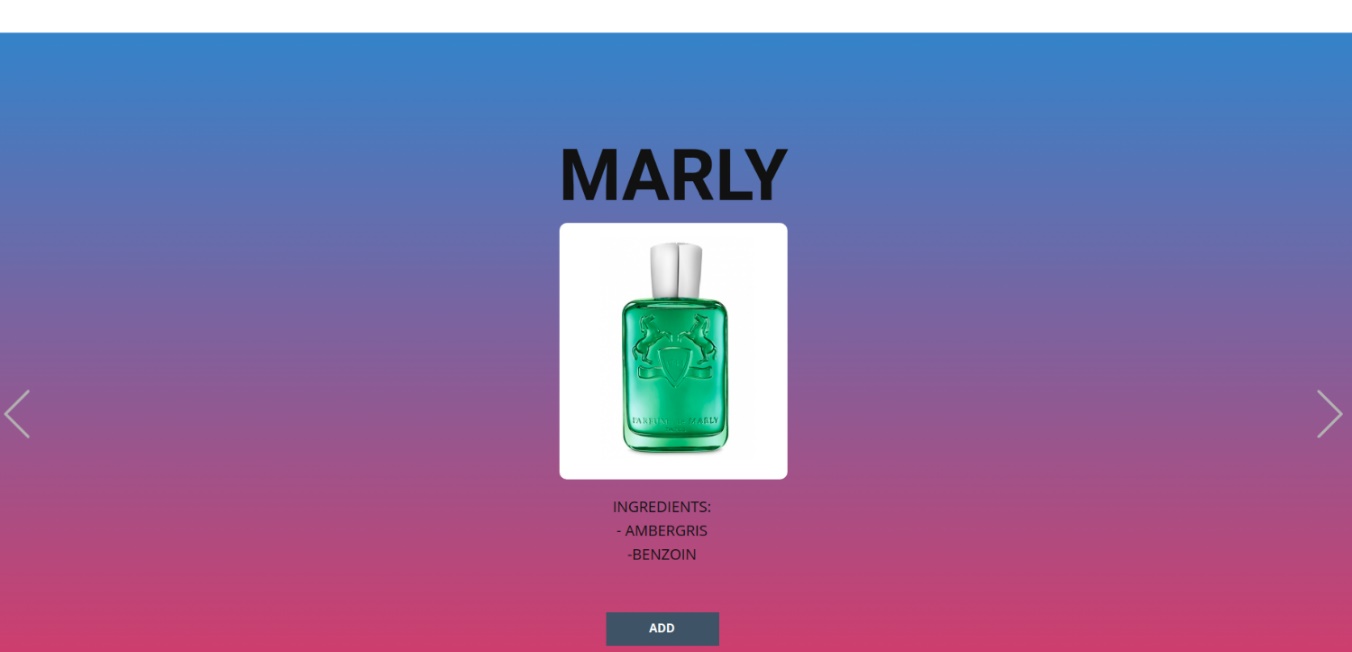
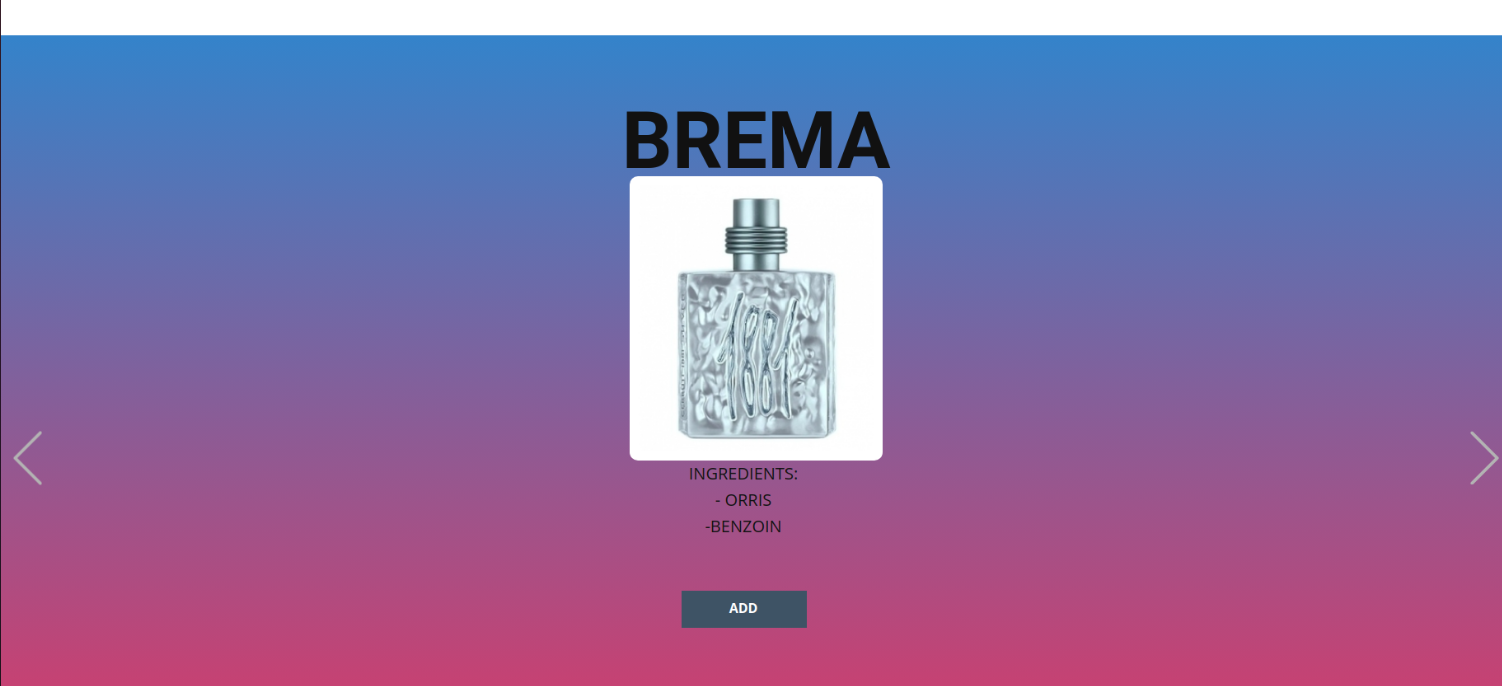
14) 

Figure 3.14: Chemist Profile Information. SQL Code: Home Page for Employees (with employee informations, Example is for Chemistry Department(department\_id=1))

SELECT e.name, e.surname, e.email, e.Birthdate, e.startDate, e.address, e.telNo, e.salary

FROM Employee e, Department d WHERE e.department\_id=d.department\_id;

15)



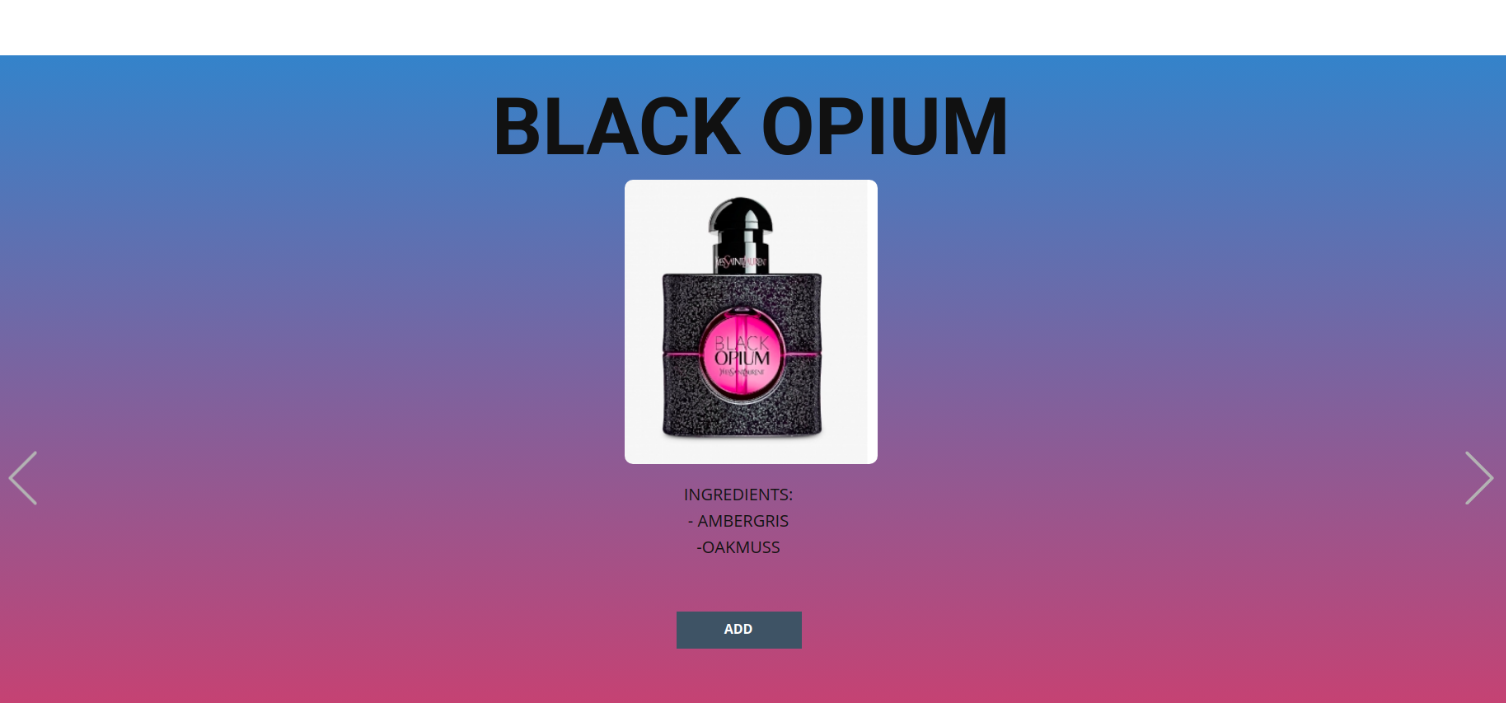


Figure 3.15: Slider of adding new perfume for chemist.The company has 15 perfumes. With sliding, chemist can choose perfume.

UPDATE Perfume

SET NumberOfStock=NumberOfStock+1

Because we know materials of each perfume, we can decrease materials with each add button.

16)

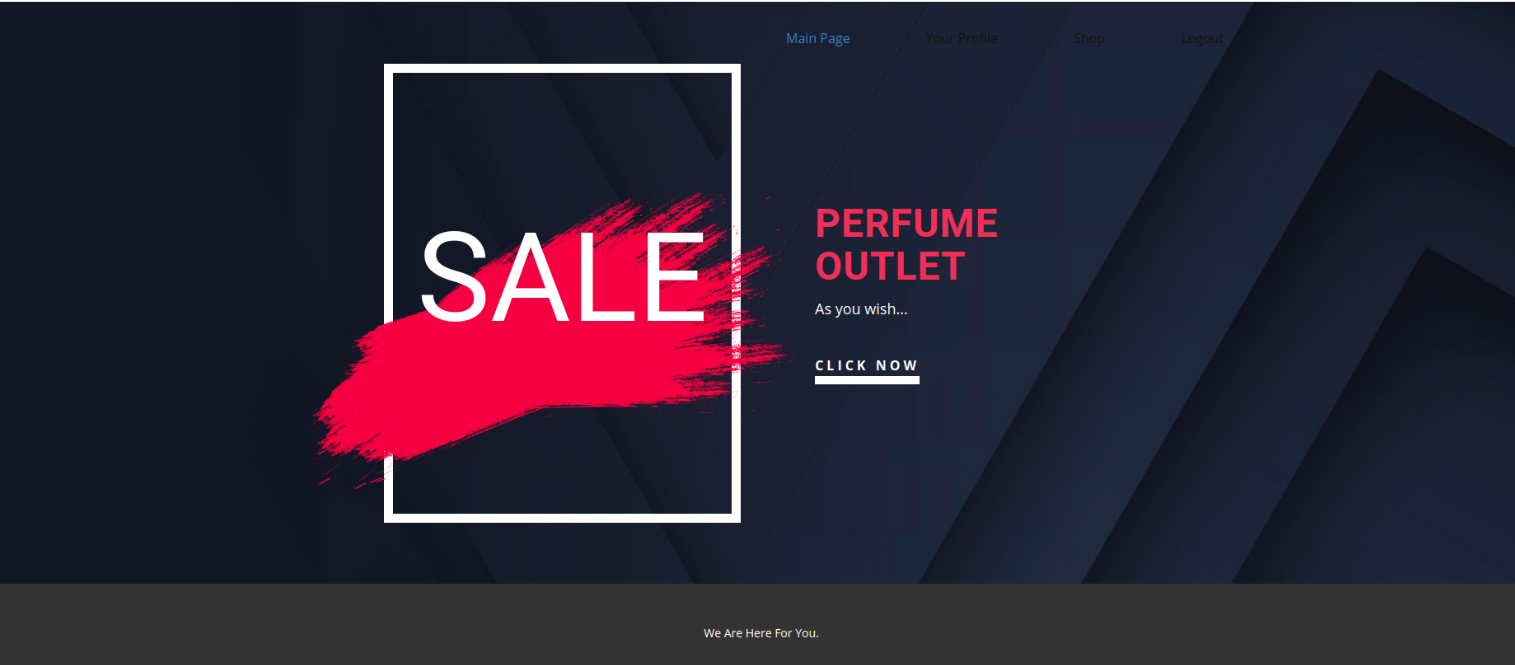


Figure 3.16: Main page of Customer. When clicking “click now”, perfumes list are represented.

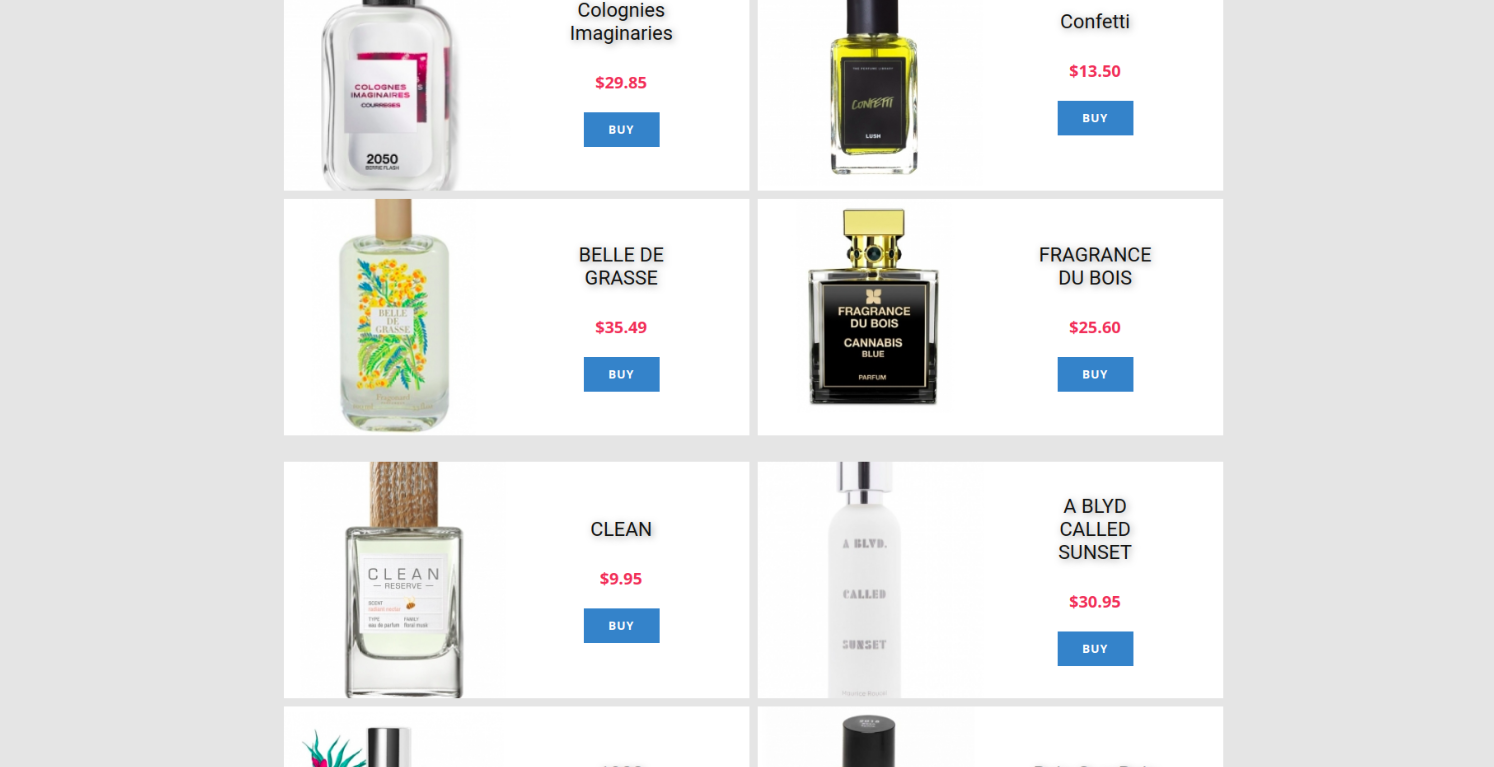
17) 

Figure 3.17: Perfume page for Customer to buy. SQL Code: UPDATE Perfume

SET NumberOfStock=NumberOfStock-1;

INSERT INTO Order(customer\_id,carrier\_id)

VALUES(customer\_id\_cookie,carrier\_id\_cookie);

18) 

Figure 3.18:Customer profile. Customers can edit their own information. SQL Code:

(for cus\_id=1)

SELECT \*

FROM customer

WHERE cus\_id=1;

19)

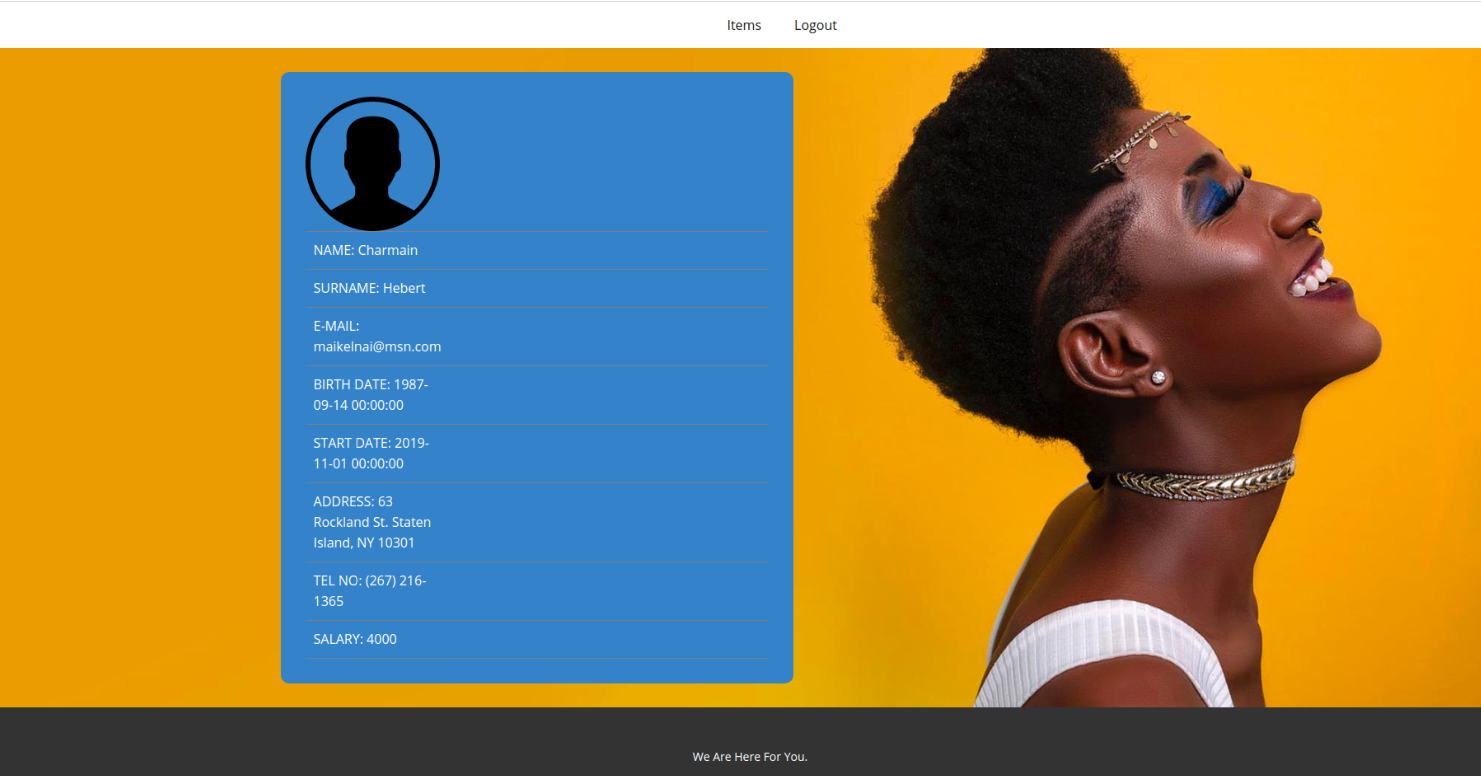


Figure 3.18: Supplier profile. SQL Code: : Home Page for Employees (with employee informations, Example is for Chemistry Department(department\_id=1))

SELECT e.name, e.surname, e.email, e.Birthdate, e.startDate, e.address, e.telNo, e.salary

FROM Employee e, Department d WHERE e.department\_id=d.department\_id;

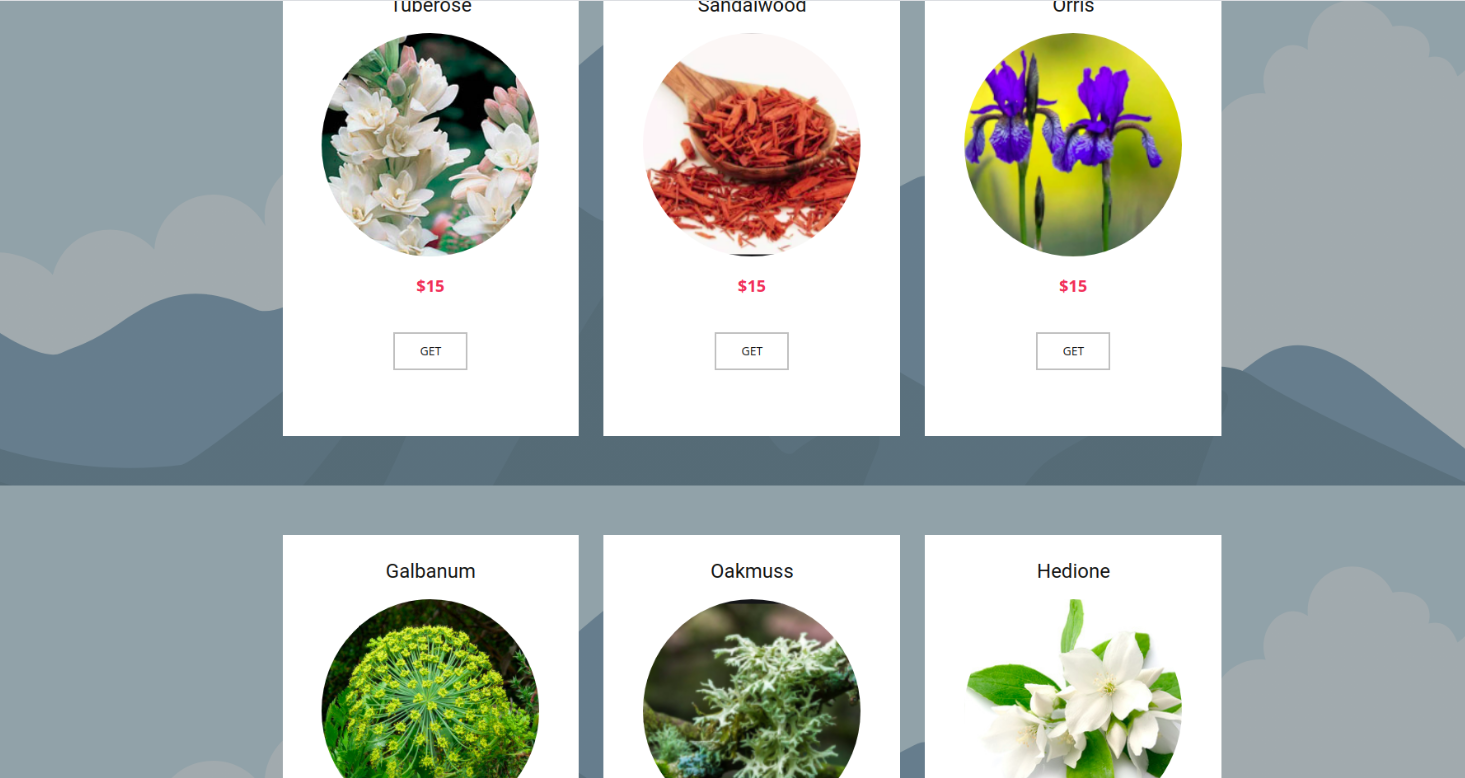
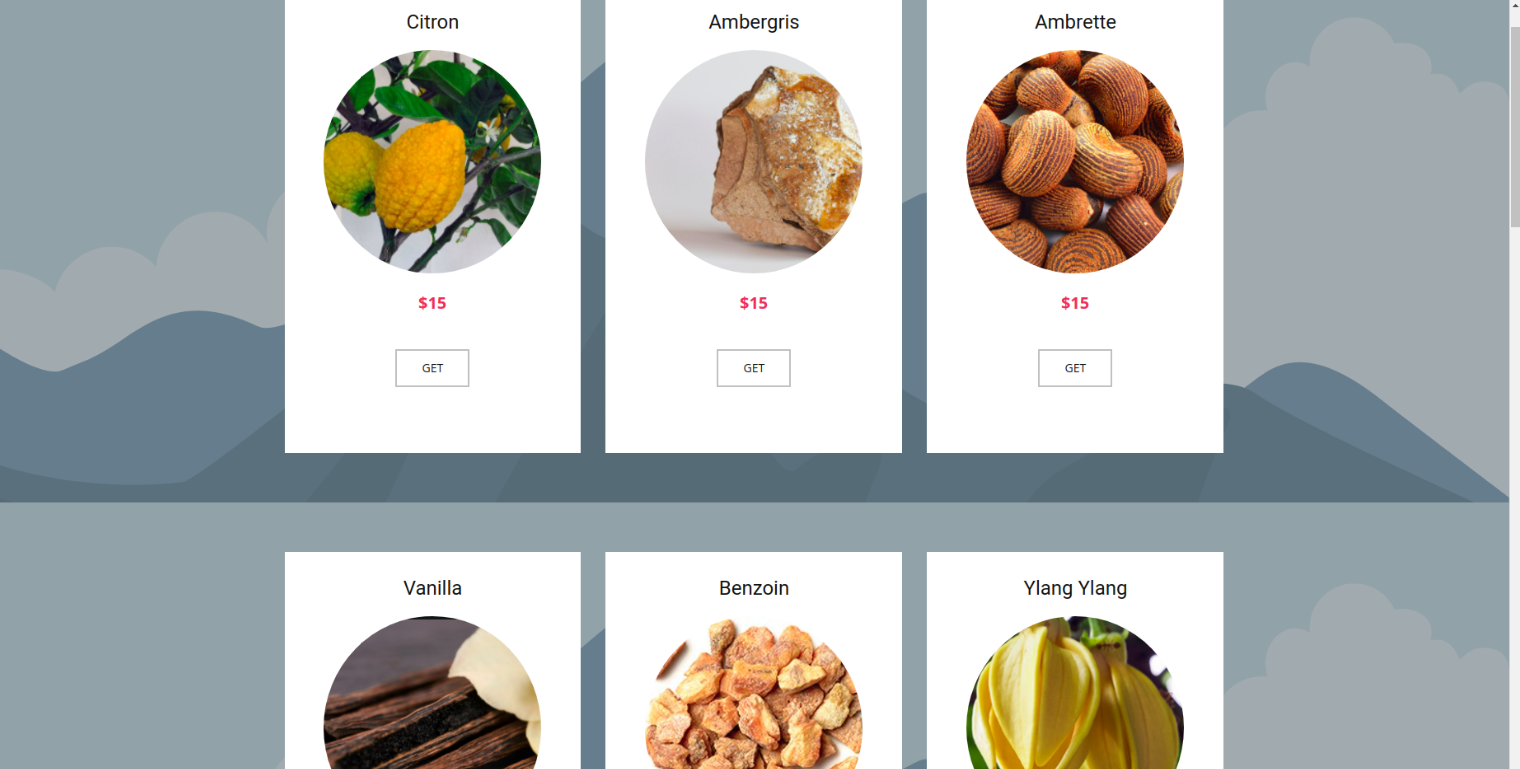
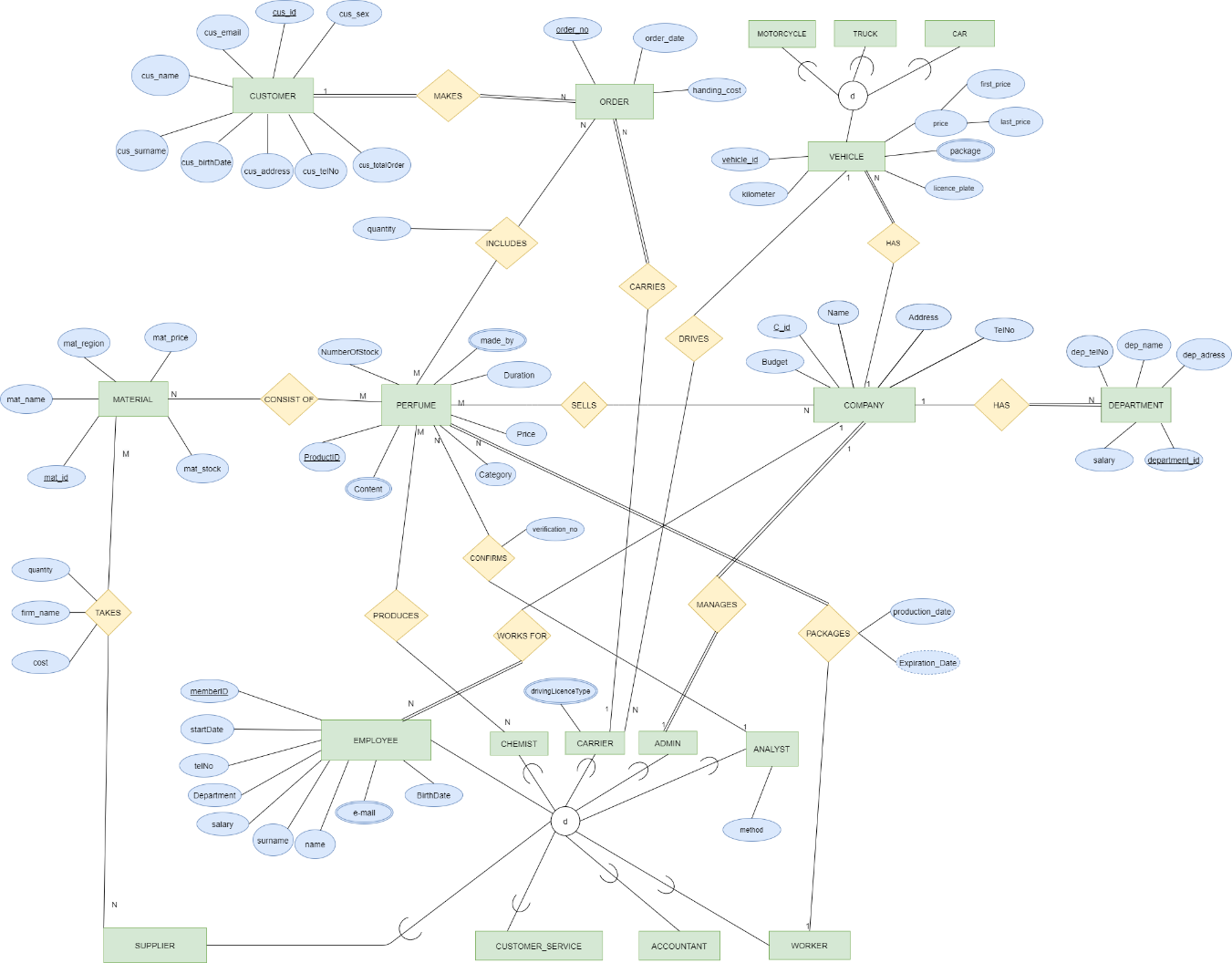
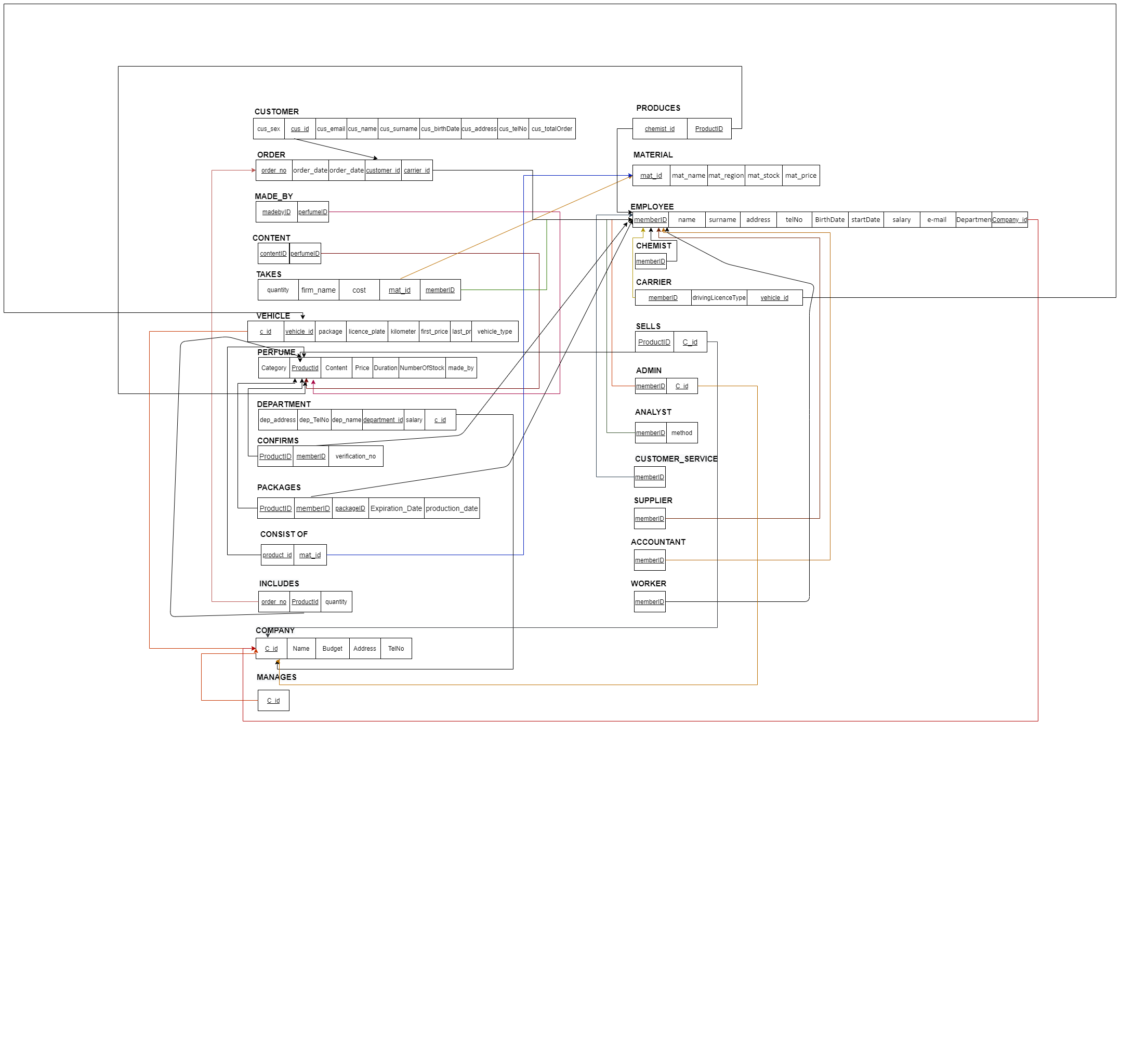
20)  

Figure 3.20: Items to buy for supplier. SQL Code: Buy Item(for mat\_id=1) 🡪 UPDATE Material SET mat\_stock=mat\_stock+1 WHERE mat\_id=1;

**Appendix A:**

****ER Diagram of Perfume Database.

**Appendix B:**

****

You can zoom or you can find relational map in ‘Dökümanlar’ files