

# Team Project 11

## Requirements Specifications

### CSCE 5350 (Summer 2022 10W)

## Contents

I.	Project Details .....	2
II.	Overall structure of the system .....	3
1.	ER-Diagram .....	3
2.	Modules.....	3
2.1	User/Customer Module: .....	3
2.2	Admin Module:.....	4
III.	Requirement Specification.....	4
1.	Functional Requirements .....	4
1.1	Components Functionality with respect to each module.....	4
1.2	Database Functionality .....	6
1.3	Database DML Screen shots.....	16
1.4	Website screenshot .....	19
1.4.2	Technology used .....	25
IV.	Project Run Instructions.....	25

## Table of Figures

Figure 1	ER-Diagram of the SOMS.....	3
----------	-----------------------------	---

## I. PROJECT DETAILS

**Project Title:** Store Orders Management System (SOMS)

**Project Description:** Store Orders Management System (SOMS), would be an online e-commerce website for clothing/fashion stores to manage online orders. This website will be an interface between the customer and the store, where customers can purchase the available products from the store's website. This application will also help the store admin to track the products and maintain the product availability information. The admin will also be able to view feedback and queries from customers and also reply to the queries asked by customers.

**Project Number: 11**

**Team Members:**

- i. Neha Goud Baddam
- ii. Reshmi Chowdary Divi
- iii. Purandhara Maharshi Chidurala
- iv. Jaya Bhargav Burugupalli

## II. OVERALL STRUCTURE OF THE SYSTEM

### 1. ER-DIAGRAM

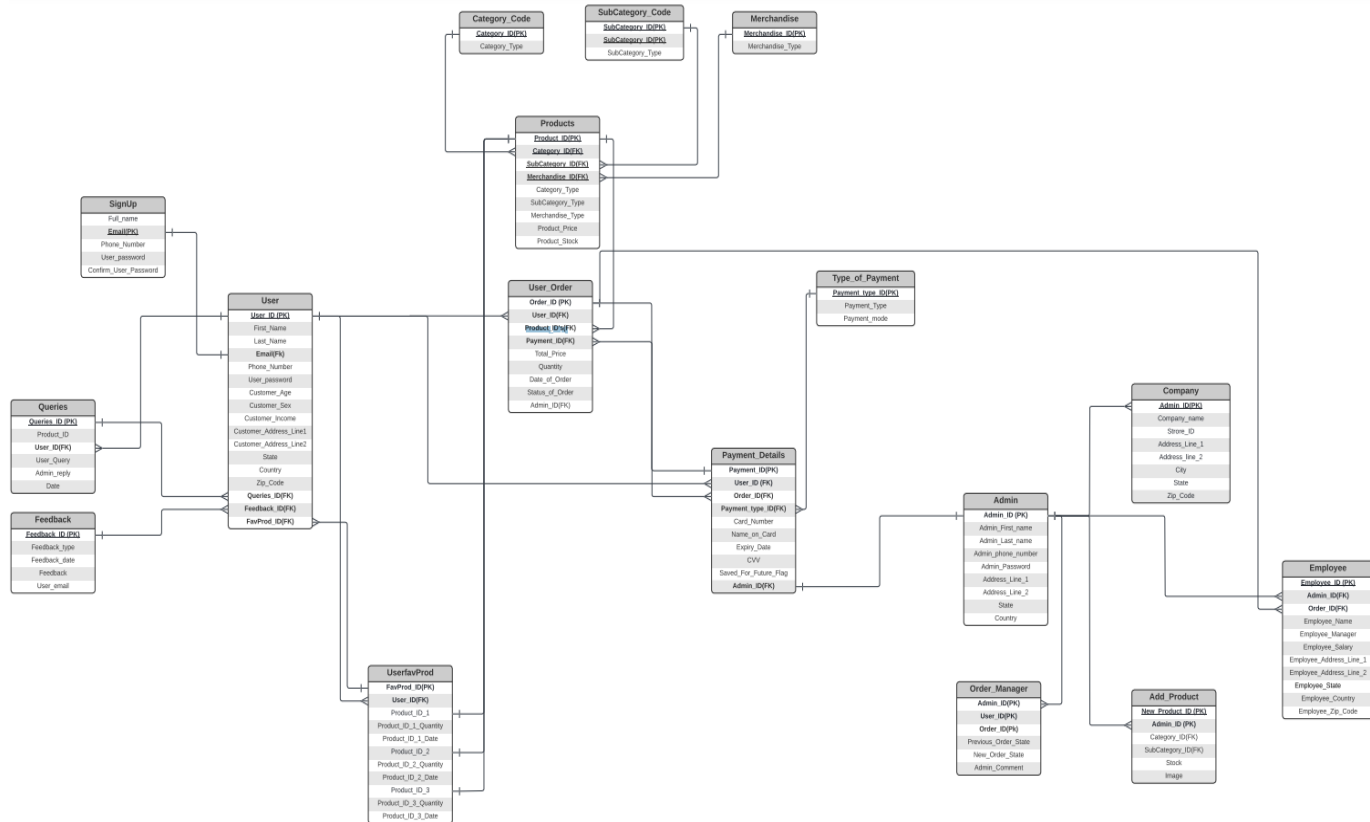


Figure 1 ER-Diagram of the SOMS

### 2. MODULES

Below are the two Main Modules:

#### 2.1 USER/CUSTOMER MODULE:

- Firstly, the customer must open the website and signup by giving the details like email, name, mobile number, and password. Once the customer has signed up, they can login using the email and password provided while sign up.
- After logging in, customer can view the dashboard will all the products available for them.
- Customer can add products they like into the **favorites**, the products will be saved until the customer removes the product.
- Customer can also ask queries about the product and admin would reply to the queries asked.

- Customer can provide feedback about the company and admin may or may not reply to it.
- Customer can add the product they wish to buy in cart and they can modify the quantity required in the cart and can view how much would be the total.
- Customer cannot add more quantity than the stock available for any particular item.
- Finally, customer can order the products the product by going through the payment.
- Customers will choose the payment options (like card details/COD) available and need to give necessary details and the order will be placed successfully.

## 2.2 ADMIN MODULE:

- Firstly, the admin must open the website and login using the existing credentials.
- Once the customer has created an order, admin can manage the order by confirming it or by canceling it. Once the order is confirmed the admin can move the status to confirm, in Transit, Delivered. The product read for delivery will be assigned to an employee.
- The employee once the product is delivered will report to the admin/manager and the admin will move the status to Delivered.
- Admin can update company details.
- Admin can add new products or update existing stock.
- Admin can also manage employee details.
- Admin can view user feedback and queries and can reply to them.

## III. REQUIREMENT SPECIFICATION

### 1. FUNCTIONAL REQUIREMENTS

#### 1.1 COMPONENTS FUNCTIONALITY WITH RESPECT TO EACH MODULE

##### 1.1.1 USER/CUSTOMER MODULE REQUIREMENTS:

##### i. **SIGN-UP:**

To Sign-up, user will have to enter Full name, Email Id, Phone number, and password fields.

After capturing above all fields. Up on Clicking “Sign up”, account will be created to login.

##### ii. **LOGIN VALIDATION:**

To Login, user must enter Email Id/phone number and password. Using user Email and password account is validated and Logged in.

iii. **DISPLAY DASHBOARD:**

The Customer can view the products available for in the company to buy.

iv. **FAVORITE PRODUCTS:**

After logged in to the account, User can select products as favorite items which are in his/her wish list to buy in future or liked items. These favorite items are stored in account favorites tab and user can buy products from there directly.

v. **ADD/REMOVE PRODUCT:**

User can “add” products to cart, which he/she is looking for purchase. Later if user is not interested in purchasing the product, “remove” option is given to delete the product from cart and proceed with other items to purchase.

vi. **PLACE ORDER:**

Once all items are selected for purchase, user can click on “Place order” to buy the products.

vii. **PAYMENT:**

Clicking on place order will re-direct the user to “Payment” option. User can choose different payment modes such as Cash on delivery or using Credit/debit cards. Once the payment is successful, an order will be placed.

viii. **QUERIES:**

The customer can ask queries about the product and post them, and the admin would reply to the queries.

ix. **FEEDBACK:**

The user will provide feedback about the company.

1.1.2 **ADMIN MODULE REQUIREMENTS:**

i. **LOGIN VALIDATION:**

For admin credentials are provided by the company. Admin credentials are validated, and admin is logged in to the account.

ii. **DISPLAY DASHBOARD:**

Once the admin logs in, dashboard is displayed. That contains options to view all orders placed by the customers, option to add new products, to manage employees and reply to queries by users.

iii. **MANAGE ORDERS:**

The admin can manage orders by either confirming the order or by updating the status of the order. The status can be moved to Packaged/Ready for delivery/Delivered.

iv. **MANAGE PRODUCTS:**

Admin will manage the existing stock/products and can also add new stock/products.

v. **MANAGE EMPLOYEES:**

Admin will manage Employee details and assign work to them.

vi. **REPLY TO QUERY:**

Admin can reply to the queries asked by the users.

## 1.2 DATABASE FUNCTIONALITY

Below are the tables that are being used.

### 1.2.2 SIGNUP

This table is specific to customer/user. The signup table has customer details like name, email, phone number, user password and confirm password fields.

**Database Fields:**

Full\_name  
Email(PK)  
Phone\_Number  
User\_password  
Confirm\_User\_Password

**SQL Query: -**

```
CREATE TABLE `SignUp` (
  `Full_name` VARCHAR(255),
  `Email` VARCHAR(255),
  `Phone_Number` VARCHAR(255),
  `User_password` VARCHAR(255),
  `Confirm_User_Password` VARCHAR(255),
  PRIMARY KEY(Email)
);
```

### 1.2.3 USER

This table is specific to customer/user. The user table has details of user details like name, email, phone number, user password and confirm password, customer age, customer gender, Queries, feedbacks, and list of favorite products fields.

**Database Fields:**

User\_ID (PK)  
First\_Name  
Last\_Name  
Email (FK)  
Phone\_Number  
User\_password  
Customer\_Age  
Customer\_Sex  
Customer\_Income  
Queries\_ID(FK)  
Feedback\_ID(FK)  
FavProd\_ID(FK)

**SQL Query: -**

```
CREATE TABLE `User` (
  `User_ID` VARCHAR(255),
  `First_Name` VARCHAR(255),
  `Last_Name` VARCHAR(255),
  `Email` VARCHAR(255),
  `Phone_Number` VARCHAR(255),
  `User_password` VARCHAR(255),
  `Customer_Age` VARCHAR(255),
  `Customer_Sex` VARCHAR(255),
  `Customer_Income` VARCHAR(255),
  `Customer_Address_Line1` VARCHAR(255),
  `Customer_Address_Line2` VARCHAR(255),
  `State` VARCHAR(255),
  `Country` VARCHAR(255),
  `Zip_Code` VARCHAR(255),
  `Queries_ID` VARCHAR(255),
  `Feedback_ID` VARCHAR(255),
  `FavProd_ID` VARCHAR(255),
  PRIMARY KEY(User_ID),
  FOREIGN KEY(Email) REFERENCES SignUp(Email),
  FOREIGN KEY(Queries_ID) REFERENCES Queries(Queries_ID),
  FOREIGN KEY(Feedback_ID) REFERENCES Feedback(Feedback_ID),
  FOREIGN KEY(FavProd_ID) REFERENCES UserfavProd(FavProd_ID)
);
```

**1.2.4 USERFAVPROD**

This table has users' favorite products.

**Database Fields:**

FavProd\_ID(PK) User\_ID(FK)

Product\_ID\_1

Product\_ID\_1\_Quantity

Product\_ID\_1\_Date

Product\_ID\_2

Product\_ID\_2\_Quantity

Product\_ID\_2\_Date

Product\_ID\_3

Product\_ID\_3\_Quantity

Product\_ID\_3\_Date

**SQL Query: -**

```
CREATE TABLE `UserfavProd` (
  `FavProd_ID` VARCHAR(255),
  `User_ID` VARCHAR(255),
  `Product_ID_1` VARCHAR(255),
```

```

`Product_ID_1_Quantity` VARCHAR(255),
`Product_ID_1_Date` VARCHAR(255),
`Product_ID_2` VARCHAR(255),
`Product_ID_2_Quantity` VARCHAR(255),
`Product_ID_2_Date` VARCHAR(255),
`Product_ID_3` VARCHAR(255),
`Product_ID_3_Quantity` VARCHAR(255),
`Product_ID_3_Date` VARCHAR(255),
PRIMARY KEY(FavProd_ID)
);

```

### 1.2.5 USER ORDER

User order table has details of order placed by the user including the payment details.

**Database Fields:**

Order\_ID (PK)  
 User\_ID(FK)  
 Product\_ID's(FK)  
 Payment\_ID(FK)  
 Total\_Price  
 Quantity  
 Date\_of\_Order  
 Status\_of\_Order  
 Admin\_ID(FK)

**SQL Query: -**

```

CREATE TABLE `User_Order` (
  `Order_ID` VARCHAR(255),
  `User_ID` VARCHAR(255),
  `Product_ID` VARCHAR(255),
  `Payment_ID` VARCHAR(255),
  `Total_Price` VARCHAR(255),
  `Quantity` VARCHAR(255),
  `Date_of_Order` VARCHAR(255),
  `Status_of_Order` VARCHAR(255),
  `Admin_ID` VARCHAR(255),
  PRIMARY KEY(Order_ID),
  FOREIGN KEY (Admin_ID) REFERENCES ADMIN(Admin_ID),
  FOREIGN KEY (Product_ID) REFERENCES Products(Product_ID),
  FOREIGN KEY (User_ID) REFERENCES USER(User_ID)
);

```

### 1.2.6 QUERIES

Queries table stores all the queries asked by the user and admin replies to that query or about the specific product.

**Database Fields:**

Queries\_ID (PK)



Product\_ID  
 User\_ID(FK)  
 User\_Query  
 Admin\_reply  
 Date  
 Admin\_ID(FK)

**SQL Query: -**

```
CREATE TABLE `Queries` (
  `Queries_ID` VARCHAR(255),
  `Product_ID` VARCHAR(255),
  `User_ID` VARCHAR(255),
  `User_Query` VARCHAR(255),
  `Admin_ID` VARCHAR(255),
  `Admin_reply` VARCHAR(255),
  `Date` VARCHAR(255),
  PRIMARY KEY(Queries_ID),
  FOREIGN KEY(Admin_ID) REFERENCES Admin(Admin_ID)
);
```

**1.2.7 FEEDBACK**

Feedback table stores all the feedbacks provided by the user about the company.

**Database Fields:**

Feedback\_ID (PK)  
 Feedback\_type  
 Feedback\_date  
 Feedback  
 User\_email

**SQL Query: -**

```
CREATE TABLE `Feedback` (
  `Feedback_ID` VARCHAR(255),
  `Feedback_type` VARCHAR(255),
  `Feedback_date` VARCHAR(255),
  `Feedback` VARCHAR(255),
  `User_email` VARCHAR(255),
  PRIMARY KEY(Feedback_ID)
);
```

**1.2.8 PRODUCT**

Product table stores all the available product details.

**Database Fields:**

Product\_ID(PK)  
 Category\_ID(FK)

SubCategory\_ID(FK)  
 Merchandise\_ID(FK)  
 Category\_Type  
 SubCategory\_Type  
 Merchandise\_Type  
 Product\_Price  
 Product\_Stock

**SQL Query: -**

```
CREATE TABLE `Products` (
  `Product_ID` VARCHAR(255),
  `Category_ID` VARCHAR(255),
  `SubCategory_ID` VARCHAR(255),
  `Merchandise_ID` VARCHAR(255),
  `Category_Type` VARCHAR(255),
  `SubCategory_Type` VARCHAR(255),
  `Merchandise_Type` VARCHAR(255),
  `Product_Price` VARCHAR(255),
  `Product_Stock` VARCHAR(255),
  PRIMARY KEY(Product_ID),
  FOREIGN KEY(Category_ID) REFERENCES
Category_Code(Category_ID),
  FOREIGN KEY(SubCategory_ID) REFERENCES
SubCategory_Code(SubCategory_ID),
  FOREIGN KEY(Merchandise_ID) REFERENCES
Merchandise(Merchandise_ID)
);
```

**1.2.9 MERCHANDISE**

Products are added to the products table based on the type of merchandise it is.

**Database Fields:**

Merchandise\_ID(PK)  
 Merchandise\_Type

**SQL Query: -**

```
CREATE TABLE `Merchandise` (
  `Merchandise_ID` VARCHAR(255),
  `Merchandise_Type` VARCHAR(255),
  PRIMARY KEY(Merchandise_ID)
);
```

### 1.2.10 CATEGORY\_CODE

Products are added to the products table based on the category code.

**Database Fields:**

Category\_ID(PK)

Category\_Type

**SQL Query: -**

```
CREATE TABLE `Category_Code` (  
  `Category_ID` VARCHAR(255),  
  `Category_Type` VARCHAR(255),  
  PRIMARY KEY(Category_ID)  
);
```

### 1.2.11 SUBCATEGORY\_CODE

Products are added to the products table based on the sub-category code.

**Database Fields:**

SubCategory\_ID(PK)

SubCategory\_ID(PK)

SubCategory\_Type

**SQL Query: -**

```
CREATE TABLE `SubCategory_Code` (  
  `SubCategory_ID` VARCHAR(255),  
  `Category_ID` VARCHAR(255),  
  `SubCategory_Type` VARCHAR(255),  
  PRIMARY KEY(SubCategory_ID),  
  FOREIGN KEY(Category_ID) REFERENCES  
  Category_Code(Category_ID)  
);
```

### 1.2.12 PAYMENT\_DETAILS

Once the order is placed and payment ID is created, this table stores all the payment details of the order with respect to the user ID and order ID.

**Database Fields:**

Payment\_ID(PK)

User\_ID (FK)

Order\_ID(FK)

Payment\_type\_ID(FK)

Card\_Number

Name\_on\_Card

Expiry\_Date

CVV

Saved\_For\_Future\_Flag

Admin\_ID(FK)

**SQL Query: -**

```
CREATE TABLE `Payment_Details` (
  `Payment_ID` VARCHAR(255),
  `User_ID` VARCHAR(255),
  `Order_ID` VARCHAR(255),
  `Payment_type_ID` VARCHAR(255),
  `Card_Number` VARCHAR(255),
  `Name_on_Card` VARCHAR(255),
  `Expiry_Date` VARCHAR(255),
  `CVV` VARCHAR(255),
  `Saved_For_Future_Flag` VARCHAR(255),
  `Admin_ID` VARCHAR(255),
  PRIMARY KEY(Payment_ID),
  FOREIGN KEY(User_ID) REFERENCES User(User_ID),
  FOREIGN KEY(Payment_type_ID) REFERENCES
Type_of_Payment(Payment_type_ID),
  FOREIGN KEY(Admin_ID) REFERENCES Admin(Admin_ID),
  FOREIGN KEY(Order_ID) REFERENCES User_Order(Order_ID)
);
```

**1.2.13 TYPE OF PAYMENT**

This table stores different payment times like COD/Visa/Debit/Credit etc.

**Database Fields:**

Payment\_type\_ID(PK)  
 Payment\_Type  
 Payment\_mode

**SQL Query: -**

```
CREATE TABLE `Type_of_Payment` (
  `Payment_type_ID` VARCHAR(255),
  `Payment_Type` VARCHAR(255),
  `Payment_mode` VARCHAR(255),
  PRIMARY KEY(Payment_type_ID)
);
```

**1.2.14 ADMIN**

This table stores login details of admin and personal details of admin.

**Database Fields:**

Admin\_ID (PK)  
 Admin\_First\_name  
 Admin\_Last\_name  
 Admin\_phone\_number  
 Admin\_Password  
 Address\_Line\_1  
 Address\_Line\_2  
 State

Country

**SQL Query: -**

```
CREATE TABLE `Admin` (  
  `Admin_ID` VARCHAR(255),  
  `Admin_First_name` VARCHAR(255),  
  `Admin_Last_name` VARCHAR(255),  
  `Admin_phone_number` VARCHAR(255),  
  `Admin_Password` VARCHAR(255),  
  `Address_Line_1` VARCHAR(255),  
  `Address_Line_2` VARCHAR(255),  
  `State` VARCHAR(255),  
  `Country` VARCHAR(255),  
  PRIMARY KEY(Admin_ID)  
);
```

#### 1.2.15 COMPANY

This table stores the company/store details.

**Database Fields:**

Admin\_ID(PK)  
Company\_name  
Store\_ID  
Address\_Line\_1  
Address\_line\_2  
City  
State  
Zip\_Code

**SQL Query: -**

```
CREATE TABLE `Company` (  
  `Admin_ID` VARCHAR(255),  
  `Company_name` VARCHAR(255),  
  `Store_ID` VARCHAR(255),  
  `Address_Line_1` VARCHAR(255),  
  `Address_line_2` VARCHAR(255),  
  `City` VARCHAR(255),  
  `State` VARCHAR(255),  
  `Zip_Code` VARCHAR(255)  
);
```

#### 1.2.16 ORDER MANAGER

Order manager table stores all the order that have been placed by all the customers. The admin would confirm the order and change the status of the order. All these details are stored in this table.

**Database Fields:**

Admin\_ID(PK)  
 User\_ID(PK)  
 Order\_ID(Pk)  
 Previous\_Order\_State  
 New\_Order\_State  
 Admin\_Comment

**SQL Query: -**

```
CREATE TABLE `Order_Manager` (
  `Admin_ID` VARCHAR(255),
  `User_ID` VARCHAR(255),
  `Order_ID` VARCHAR(255),
  `Previous_Order_State` VARCHAR(255),
  `New_Order_State` VARCHAR(255),
  `Admin_Comment` VARCHAR(255),
  PRIMARY KEY(Admin_ID,User_ID,Order_ID)
);
```

**1.2.17 ADD\_PRODUCT**

The admin can add new products based on the merchandise type, category code, and sub-category code. All these details are stored in this table and then added to the Product table.

**Database Fields:**

New\_Product\_ID (PK)  
 Admin\_ID (PK)  
 Category\_ID(FK)  
 SubCategory\_ID(FK)  
 Stock Image

**SQL Query: -**

```
CREATE TABLE `Add_Product` (
  `New_Product_ID` VARCHAR(255),
  `Admin_ID` VARCHAR(255),
  `Category_ID` VARCHAR(255),
  `SubCategory_ID` VARCHAR(255),
  `Stock` VARCHAR(255),
  `Image` VARCHAR(255),
  PRIMARY KEY(New_Product_ID,Admin_ID),
  FOREIGN KEY(Category_ID) REFERENCES
  Category_Code(Category_ID),
  FOREIGN KEY(SubCategory_ID) REFERENCES
  SubCategory_Code(SubCategory_ID)
);
```

**1.2.18 EMPLOYEE**

The employee table stores all the details of employees and the admin under

which the employee works. This table also stores Order Id of the products delivered by the employee.

**Database Fields:**

Employee\_ID (PK)

Admin\_ID(FK)

Employee\_Name

Employee\_Manager

Employee\_Salary

Employee\_Address\_Line\_1

Employee\_Address\_Line\_2

Employee\_State

Employee\_Country

Employee\_Zip\_Code

**SQL Query: -**

```
CREATE TABLE `Employee` (  
  `Employee_ID` VARCHAR(255),  
  `Admin_ID` VARCHAR(255),  
  `Order_ID` VARCHAR(255),  
  `Employee_Name` VARCHAR(255),  
  `Employee_Manager` VARCHAR(255),  
  `Employee_Salary` VARCHAR(255),  
  `Employee_Address_Line_1` VARCHAR(255),  
  `Employee_Address_Line_2` VARCHAR(255),  
  `Employee_State` VARCHAR(255),  
  `Employee_Country` VARCHAR(255),  
  `Employee_Zip_Code` VARCHAR(255),  
  PRIMARY KEY(Employee_ID),  
  FOREIGN KEY(Admin_ID) REFERENCES ADMIN(Admin_ID),  
  FOREIGN KEY(Order_ID) REFERENCES user_order(Order_ID)  
);
```

All the DMLs can be found in below attachments.



Project DML.sql

## 1.3 DATABASE DML SCREEN SHOTS

### 1.3.2 SIGNUP

10 • `SELECT * FROM SIGNUP;`

	Full_name	Email	Phone_Number	User_password	Confirm_User_Password
▶	Neha Goud Baddam	testemail1@gmail.com	1111111111	test_pass1	test_pass1
	Reshmi Chowdary Divi	testemail2@gmail.com	2222222222	test_pass2	test_pass2
	Purandhara Maharshi Chidurala	testemail3@gmail.com	3333333333	test_pass3	test_pass3
*	NULL	NULL	NULL	NULL	NULL

### 1.3.3 USER

10 • `SELECT * FROM USER;`

	User_ID	First_Name	Last_Name	Email	Phone_Number	User_password	Customer_Age	Customer_Sex	Customer_Income	Customer_Address
▶	99101	Louise	Montgomery	testemail1@gmail.com	512-925-8738	test_pass_1	25	M	100000\$	2797 Christie Way
	99102	Darlene	Hughes	testemail2@gmail.com	773-717-9539	test_pass_2	32	F	150000\$	4105 Colony Street
	99103	Rosie	Cannon	testemail3@gmail.com	619-925-6125	test_pass_3	27	M	1750000\$	1563 Webster Street
	99104	Maryann	Powell	testemail1@gmail.com	214-454-5951	test_pass_4	42	F	200000\$	4809 Hickory Street
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 1.3.4 USERFAVPROD

10 • `SELECT * FROM USERFAVPROD;`

	FavProd_ID	User_ID	Product_ID_1	Product_ID_1_Quantity	Product_ID_1_Date	Product_ID_2	Product_ID_2_Quantity	Product_ID_2_Date	Product_ID_3	Product_ID_3_Quantity
▶	FP103	99103	P101	3	15-07-2022	P102	4	21-07-2022	P103	3
	FP102	99102	P104	1	20-07-2022	P105	1	18-07-2022	P106	4
	FP101	99101	P101	2	25-07-2022	P102	2	10-07-2022	P103	1
		99103	P101	3	15-07-2022	P102	4	21-07-2022	P103	3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 1.3.5 USER\_ORDER

10 • `SELECT * FROM USER_ORDER;`

	Order_ID	User_ID	Product_ID	Payment_ID	Total_Price	Quantity	Date_of_Order	Status_of_Order	Admin_ID
▶	O101	99101	P101	5510101	200\$	10	25-07-2022	In-Progress	admin101
	O102	99102	P102	5510202	250\$	5	25-07-2022	In-Transit	admin102
	O103	99103	P103	5510303	300\$	20	25-07-2022	Order	admin102
	O104	99104	P104	5510404	350\$	25	25-07-2022	Out for Delivery	admin101
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 1.3.6 QUERIES

10 • `SELECT * FROM QUERIES;`

	Queries_ID	Product_ID	User_ID	User_Query	Admin_ID	Admin_reply	Date
▶	Q102	P102	99102	How many days will the refund take?	admin102	By next week	24-07-2022
	Q101	P101	99101	When will the sale start?	admin101	In december	24-07-2022
		P102	99102	How many days will the refund take?	admin102	By next week	24-07-2022
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL



### 1.3.7 FEEDBACK

10 • `SELECT * FROM FEEDBACK;`

Feedback_ID	Feedback_type	Feedback_date	Feedback	User_email
F102	Postive	24-07-2022	I like you products.	testemail2@gmail.com
F101	Postive	24-07-2022	The color and fabric of my dress is awesome.	testemail1@gmail.com
	Postive	24-07-2022	I like you products.	testemail2@gmail.com
NULL	NULL	NULL	NULL	NULL

### 1.3.8 PRODUCTS

10 • `SELECT * FROM PRODUCTS;`

Product_ID	Category_ID	SubCategory_ID	Merchandise_ID	Category_Type	SubCategory_Type	Merchandise_Type	Product_Price	Product_Stock
P101	C101	SC101		Men	Men Shirts		100\$	15
P102	C102	SC102		Women	Women wear		150\$	25
P103	C103	SC103		Children	Children dress		75\$	30
P104	C104	SC104	M101	Merchandise	Watch	Watch	250\$	10
P105	C101	SC103		Men	Men Shirts		200\$	35
P106	C102	SC102		Women	Women wear		125\$	20

### 1.3.9 MERCHANDISE

10 • `SELECT * FROM MERCHANDISE;`

Merchandise_ID	Merchandise_Type
M103	Handbag
M102	Purse
M101	Watch
	Handbag
NULL	NULL

### 1.3.10 CATEGORY CODE

10 • `SELECT * FROM CATEGORY_CODE;`

Category_ID	Category_Type
C104	Merchandise
C103	Children
C102	Women
C101	Men
	Merchandise
NULL	NULL

### 1.3.11 SUBCATEGORY\_CODE

10 • `SELECT * FROM SUBCATEGORY_CODE;`

SubCategory_ID	Category_ID	SubCategory_Type
SC104	C104	Watch
SC103	C103	Children dress
SC102	C102	Women wear
SC101	C101	Men Shirts
	C104	Watch
NULL	NULL	NULL

### 1.3.12 PAYMENT\_DETAILS

10 • `SELECT * FROM PAYMENT_DETAILS;`

Payment_ID	User_ID	Order_ID	Payment_type_ID	Card_Number	Name_on_Card	Expiry_Date	CVV	Saved_For_Future_Flag	Admin_ID
99121011	99101	O101	5510101	1230-4560-7890	Louise	Jul-27	123	Y	admin101
99121012	99102	O102	5510202	1111-1111-1111	Darlene	Aug-26	123	Y	admin102
99121013	99103	O103	5510303	2222-2222-2222	Rosie	Sep-24	123	N	admin101
99121014	99104	O104	5510404	3333-3333-3333	Maryann	Sep-25	123	N	admin102
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 1.3.13 TYPE\_OF\_PAYMENT

10 • `SELECT * FROM TYPE_OF_PAYMENT;`

Payment_type_ID	Payment_Type	Payment_mode
5510101	Credit card	Online
5510202	Debit card	Online
5510303	COD	COD
5510404	Amazon Pay	Online
NULL	NULL	NULL

### 1.3.14 ADMIN

10 • `SELECT * FROM ADMIN;`

Admin_ID	Admin_First_name	Admin_Last_name	Admin_phone_number	Admin_Password	Address_Line_1	Address_Line_2	State	Country
admin101	Jaya Bhargav	Burugupalli	4444444444	testadmin_pass1	2659 Wescam Court		Texas	USA
admin102	Purandhara Maharshi	Chidurula	5555555555	testadmin_pass2	4041 Elsie Drive		Texas	USA
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 1.3.15 COMPANY

10 • `SELECT * FROM COMPANY;`

Admin_ID	Company_name	Store_ID	Address_Line_1	Address_line_2	City	State	Zip_Code
admin101	Store Order Management System	101	8711 Wilson Road		Fort Dodge	IA	50501
admin102	Store Order Management System	102	9 Brickell Avenue		Williamsport	PA	17701

### 1.3.16 ORDER\_MANAGER

10 • `SELECT * FROM ORDER_MANAGER;`

Admin_ID	User_ID	Order_ID	Previous_Order_State	New_Order_State	Admin_Comment
admin101	99103	O103	Delivered		
admin102	99102	O102	In-Transit	Will be out for delivery	
admin102	99104	O104	Out for Delivery	Delivered	
admin102	99104	O105	Out for Delivery	Delivered	
NULL	NULL	NULL	NULL	NULL	NULL

### 1.3.17 ADD\_PRODUCT

10 • `SELECT * FROM ADD_PRODUCT;`

Result Grid

Filter Rows:

Edit:

Export/Im

	New_Product_ID	Admin_ID	Category_ID	SubCategory_ID	Stock	Image
	P108	admin101	C102	SC102	20	IMG2
	P108	admin102	C102	SC102	20	IMG2
	NULL	NULL	NULL	NULL	NULL	NULL

### 1.3.18 EMPLOYEE

10 • `SELECT * FROM EMPLOYEE;`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Employee_ID	Admin_ID	Order_ID	Employee_Name	Employee_Manager	Employee_Salary	Employee_Address_Line_1	Employee_Address_Line_2	Employee_State	Employee_Country
E102	admin102	O102	Noah Day	Geneva Steele	1250000\$	7628 Cambridge St.		FL	USA
E103	admin101	O103	Jean Banks	Geneva Steele	2500000\$	134 Oxford Street		GA	USA
E104	admin102	O104	Wendell Brown	Geneva Steele	3500000\$	3 Thorne St.		MD	USA
E105	admin101	O101	Martha Gordon	Geneva Steele	4000000\$	9 Brickell Avenue		PA	USA
E106	admin101	O101	Martha Gordon	Geneva Steele	4000000\$	9 Brickell Avenue		PA	USA

All the DML's can be found below:

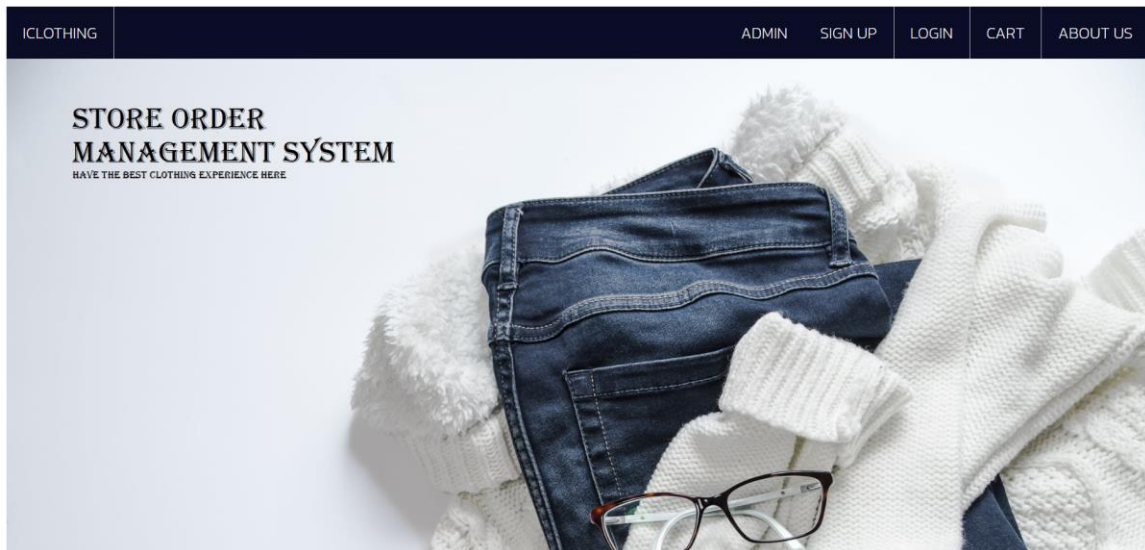


Project DML.sql

## 1.4 WEBSITE SCREENSHOT

### 1.4.2 CUSTOMER DASHBOARD

The customer can go to LOGIN page or can go to Sign up page from here. The admin can move to admin page fot login.



### 1.4.3 ADMIN/CUSTOMER LOGIN PAGE

Where customer and admin can give their credentials to login

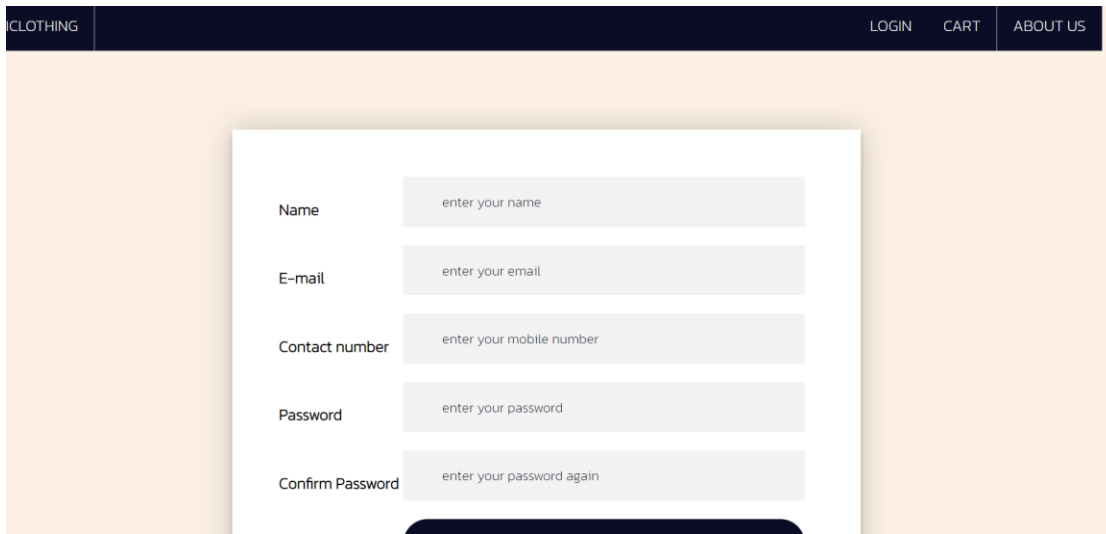
Admin page:

Customer page: Customer can go to SIGNUP page if not an existing customer from here.

Sign up'."/>

#### 1.4.4 CUSTOMER SIGNUP PAGE

New Customers can sign up from here.



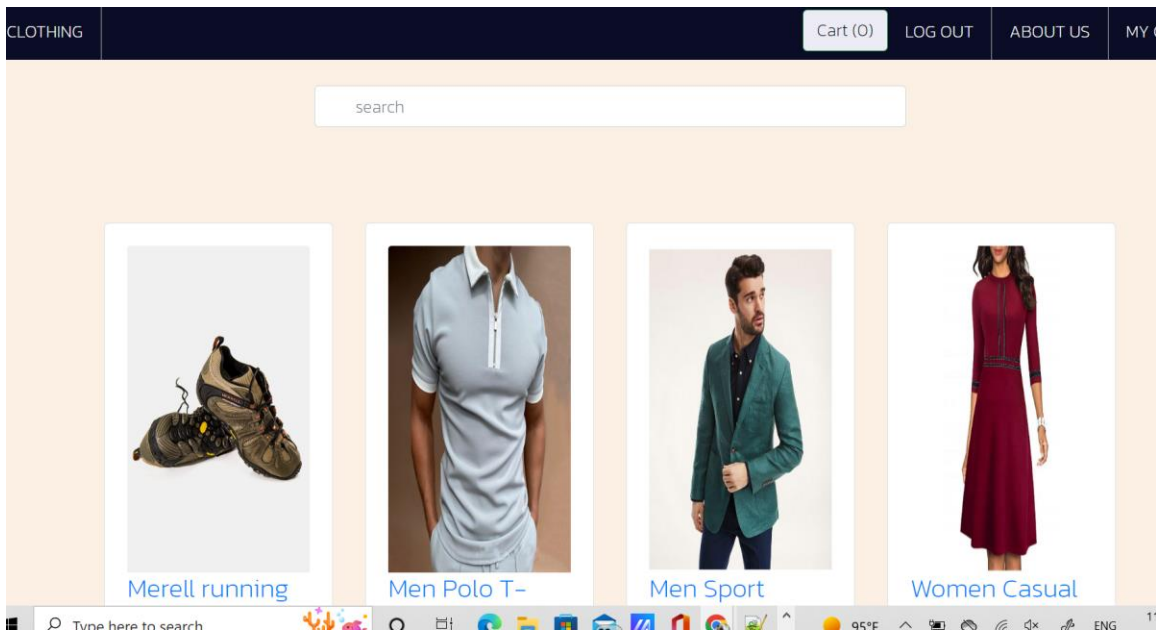
The screenshot shows a registration form on a website with a dark blue header. The header contains the text 'CLOTHING' on the left and 'LOGIN', 'CART', and 'ABOUT US' on the right. The form itself is a white box with a light orange background. It contains five input fields, each with a label on the left and a placeholder text inside the field:

- Name**: enter your name
- E-mail**: enter your email
- Contact number**: enter your mobile number
- Password**: enter your password
- Confirm Password**: enter your password again

Below the input fields is a dark blue button.

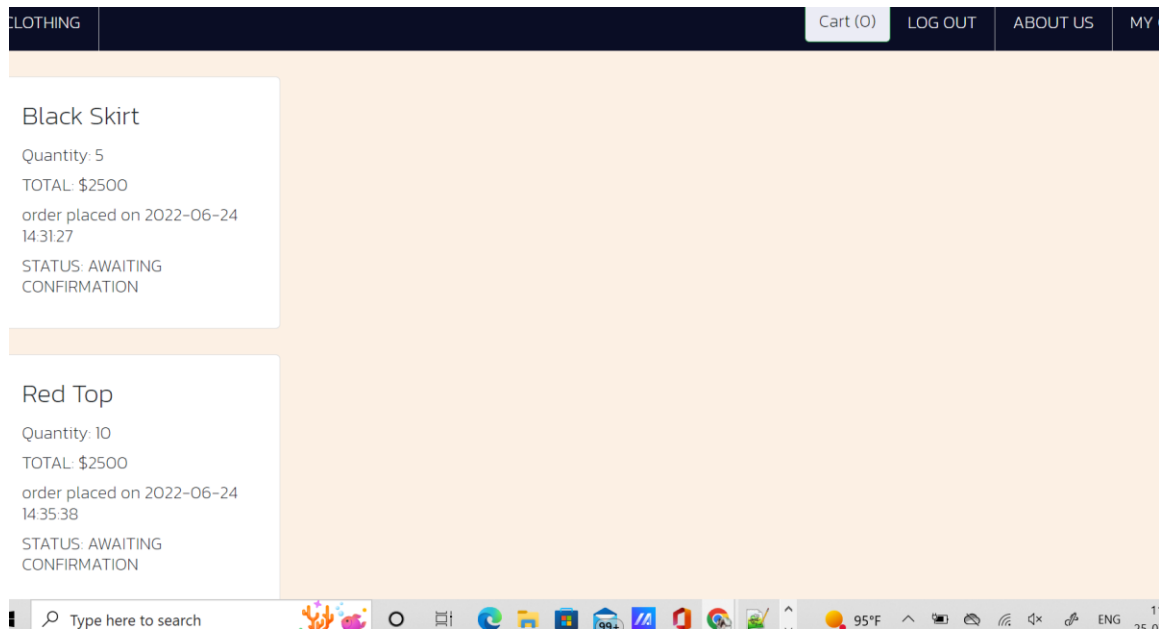
#### 1.4.5 CUSTOMER DASHBOARD

After logging in customer can view dashboard for available products.



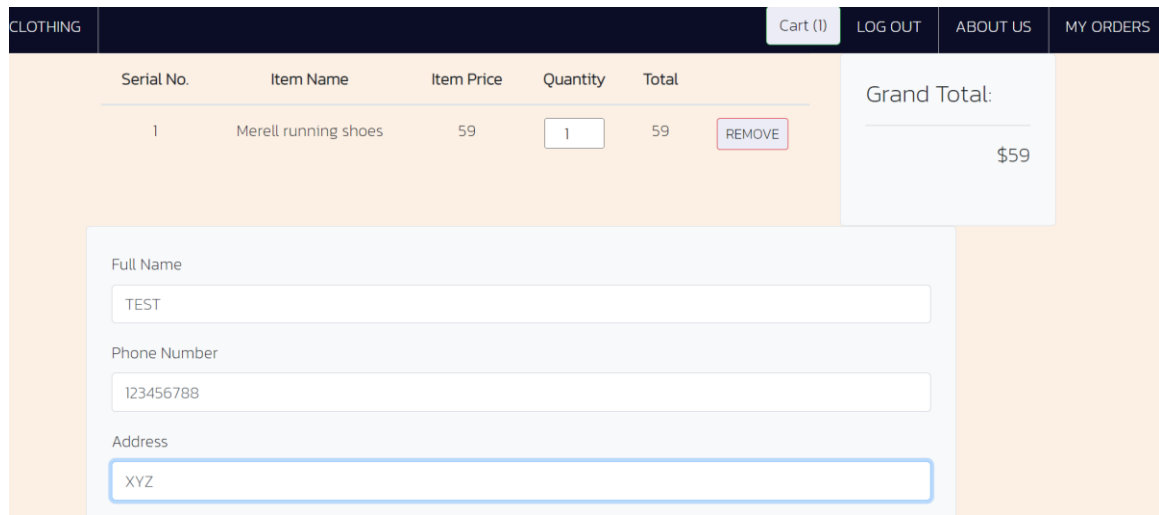
#### 1.4.6 CUSTOMER MYORDER PAG

Customer can view their previous orders.



#### 1.4.7 [CART PAGE](#)

The products added by the customer can be viewed here and the customer can increase the quantity here and can view the total amount. They can also give their details to place an order.



#### 1.4.8 [PAYMENT PAGE](#)

Customer should give their payment option and required details to place an order.

Phone Number  
123456788

Address  
XYZ

CASH ON DELIVERY ☐ CREDIT/DEBIT CARD NUMBER ☐

SECURE SSL ENCRYPTION GUARANTEED SAFE CHECKOUT  
PayPal VISA Mastercard AMERICAN EXPRESS

8881 2545 2545 2245

Expiry MM / YY CVV /CVC 123

Cancel Make Purchase

#### 1.4.9 FEEDBACK/QUERIES PAGE

Customers can ask their queries and give their feedback here by entering their email first.

feedback/Query

email Enter Your email id..

OPTIONS SELECT OPTIONS

SELECT OPTIONS  
feedback  
query

Submit

Type here to search 95°F 12:01 25-07-2022

#### 1.4.10 ADMIN ORDER VIEW PAGE

Admin can view the order placed by customer can change the status of the order.

NG ORDERS					FEEDBACKS		LOGOUT	QUERIES	ADD PRODUCT
Order ID	Customer Name	Phone No	Address	Pay Mode	Orders				
31	sa	3	a	cod	Item Name	Price	Quantity	STATUS	
					Black Skirt	500	5	AWAITING CONFIRMATION <a href="#">update</a>	
32	SA	23	3	cod	Item Name	Price	Quantity	STATUS	
					Red Top	250	10	AWAITING CONFIRMATION <a href="#">update</a>	
39	TEST	123456789	XYZ		Item Name	Price	Quantity	STATUS	
					Merell running shoes	59	1	AWAITING CONFIRMATION <a href="#">update</a>	

#### 1.4.11 [ADMIN FEEDBACK VIEW PAGE](#)

Admin can view feedback given by customer and can view them and may be reply.

date	TYPE	feedback	EMAIL
2022-07-25 12:01:18			<a href="#">reply</a>
2022-07-24 15:54:50	feedback	Hi	jsaidevprakash101@gmail.com <a href="#">reply</a>
2022-07-24 15:54:36			<a href="#">reply</a>
2022-06-24 00:34:32	query	when will new collection be added?	srijachagamreddy@gmail.com <a href="#">reply</a>
2022-06-23 14:42:46	query	how long does it take to ship orders?	chundur.sandeep2@gmail.com <a href="#">reply</a>
2022-06-22 10:08:25	feedback	This is my very first order through site, and I am totally and completely satisfied! The fit is great and so are the prices. I will definitely return again and again...	no email <a href="#">reply</a>

#### 1.4.12 [ADMIN QUERIES VIEW PAGE](#)

Admin can view the queries asked by customer and can reply to them.

ORDERS					FEEDBACKS	LOGOUT	ADD PRODUCT
	id	date	PRODUCT ID	USER ID	QUERY		
	8	2022-06-24 17:14:55	1	1	Is there any discount? <a href="#">reply</a>		
	6	2022-06-24 16:16:15	1	1	is Medium size available? <a href="#">reply</a>		



### 1.4.13 ADMIN ADD PRODUCT PAGE

Admin can the products from this page and they will be visible to customer dashboard.

ORDERS FEEDBACKS LOGOUT ORDERS

CATEGORY accessories add

SUB CATEGORY belts add

NAME OF APPAREL

COST

UPLOAD IMAGE Choose File No file chosen

### 1.4.2 TECHNOLOGY USED

Front End: - HTML, CSS

Database: - MYSQL

Back-End: - PHP

## IV. PROJECT RUN INSTRUCTIONS

1. Install Xampp on your system - <https://www.apachefriends.org/download.html>
2. Download the below Zip file and extract it in the path [C:\xampp\htdocs](#).



iclothing.zip

3. Open the Xampp control panel and start the Apache and MySQL Modules.

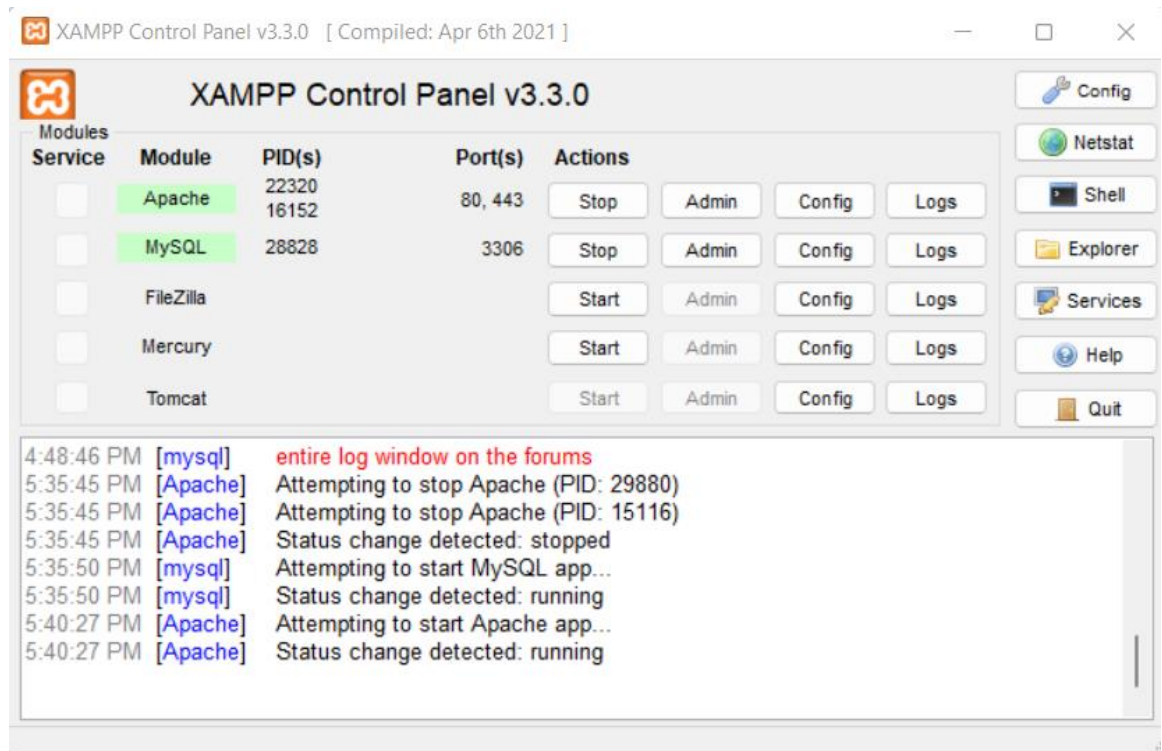


Figure 2: Xampp Control panel

4. Open MySQL Workbench, open “Project DDL.sql” followed by “Project DML.sql” and execute the files one after the other. Make sure the database is created and has data as per the DML and DDL provided.
5. Type the URL <http://localhost/fdb>, Main page of the project will be displayed