DATABASE SYSTEMS LAB MINI PROJECT
Final Report

TITLE:

An E-Commerce portal using Java Swings as front-end and SQL as back-end (database).

Github : https://github.com/anuragpatro/dbs_mini_project

TEAM MEMBERS:

NAME	ROLL NO
1. ANURAG PATRO	59 – D2
2. SURYANSH SHARMA	66 – D2

Problem Statement :

Databases are powerful, well tested and suitable for e-commerce.

The E-Commerce portal will have the following features:

- The Database will consist of all the products/items with their prices and customer details and particular carts will be added on interaction with the GUI. The front-end part of the portal will help you in shopping as it will provide you with the products/items (available in the database). We are going to have an option of ‘add to cart’ so that a customer can add the items/ products in the cart which he/she is willing to buy. The DBMS would be an embodiment of several features, aggregate functions for mathematical, logical operations; Integrity constraints for specifying limitations of various attributes and their characteristics; and nested queries and more to extract information smartly.

- The GUI makes use of several JFrames, connected by Java Swing facilities, making the interface extremely easy to navigate through.

- PL/SQL blocks, particularly a trigger and a procedure have been utilised to facilitate the use of blocks to enhance productivity and reduce effort of coding.

Goal:

We aim to reproduce an e-commerce portal site like Flipkart, Amazon, etc. , replicate their features to a maximal extent, provide marketing to users over various categories, store their details and provide an experience similar to that of the marketing websites with SQL database linked with Java GUI display.

Abstract :

Since our product is a e-marketing application, there is the '**Login**' feature that confirms passwords with entered user details (i.e. email - id and password), and to add a new account, there's the '**Register**' feature.

The '**Register**' feature uses a pl/sql procedure to insert values into the table '**Customer**'.

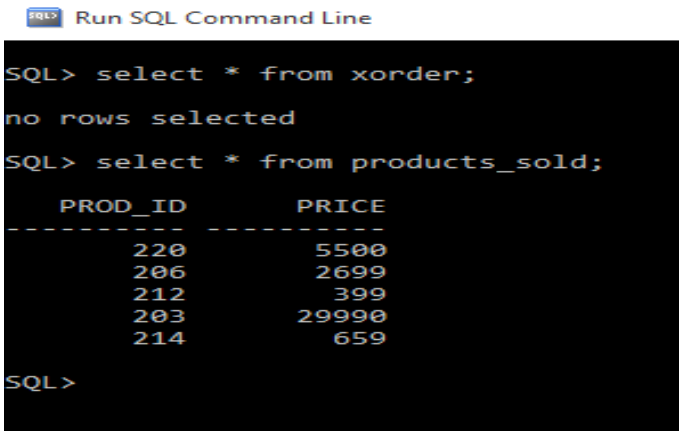
This directs the user to a window containing a variety of categories that a user can select from, which then direct the user to a window for that particular category.

This window displays all products with its requisite details that a shopper can choose from and select the '**Add to Cart**' option that adds the item to the cart table in the database.

Once the user is done with shopping, he/she can proceed to the cart where the application loads the present cart details into the display and calculates the total sum to be paid and asks the User for a preferred '**Shipper**'.

The user can now proceed to the payment page, where a hardcoded QR code is provided that he/she can assume to scan using any UPI app and then notify the application of payment on which they receive details of the Shipper's name, rating and Delivery date as per their selection, retrieved from the table.

This also triggers the notification of an order being placed in the table XOrder, and to signify the stocks, and the amount sold, we use a table '**Products_Sold**' triggered by a pl/sql trigger block to add all products to this table once sold.



DDL COMMANDS

```
create table Customer(
        fname varchar(20) not null,
        mname varchar(20),
        lname varchar(20) not null,
        cust_id number(8) not null,
        postal_id number(8),
        city varchar(20),
        ph_no number(12),
       primary key(cust_id));

create table Shipper(
        ship_id number(8) not null,
        ship_name varchar(20),
        rating number(8),
        ddate date,
        primary key(ship_id));

create table XOrder(
        ddate date ,
        tot_price numeric(10,2),
        ship_id number(8) );


create table Supplier(
        sup_id number(8) not null,
        sup_name varchar(20),
        rating number(8),
        primary key(sup_id));

create table Category(
        cat_id number(8) not null,
        cat_name varchar(20),
        primary key(cat_id));

create table Product(
        prod_id number(8) not null,
```

```
        prod_name varchar(40),
        price numeric(8,2),
        sup_id number(8),
        cat_id number(8),
        primary key(prod_id),
        foreign key(sup_id) references Supplier(sup_id),
        foreign key(cat_id) references Category(cat_id));

create table Cart(
        prod_id number(8),
       ptot_price numeric(10,2) ,
        primary key(prod_id),
        foreign key(prod_id) references Product(prod_id)
        );

create table Account(
          cust_id number(8) not null,
          email_id varchar(320) not null,
          pass varchar(10) not null,
          foreign key(cust_id) references Customer(cust_id));

create table Products_sold(
           prod_id  number(8),
           price    numeric(8,2));
```

INSERTIONS:

```
INSERT INTO Shipper VALUES(500,'EVER_MARINE',4,'18 APRIL 2019');
INSERT INTO Shipper VALUES(501,'GOLD-FISH',3.5,'19 APRIL 2019');
INSERT INTO Shipper VALUES(502,'BLUE-WHALE',4.5,'17 APRIL 2019');
INSERT INTO Shipper VALUES(503,'BLUE_SEA',3,'19 APRIL 2019');
INSERT INTO Shipper VALUES(504,'OCEAN-BLUE',5,'17 APRIL 2019');
INSERT INTO Shipper VALUES(505,'STORMY',3,'20 APRIL 2019');

INSERT INTO SUPPLIER VALUES(2000,'NIKE',5);
INSERT INTO SUPPLIER VALUES(2001,'PHILIPS',4);
INSERT INTO SUPPLIER VALUES(2002,'GADA ELECTRONICS',4.5);
INSERT INTO SUPPLIER VALUES(2003,'RAMESH WATCHES',4.5);
```

```
INSERT INTO SUPPLIER VALUES(2004,'FBB',4);
INSERT INTO SUPPLIER VALUES(2005,'MORE MEGASTORE',4.5);
INSERT INTO SUPPLIER VALUES(2006,'WALDEN',4);
INSERT INTO SUPPLIER VALUES(2007,'UNIVERSAL PUBLISHER',5);
INSERT INTO SUPPLIER VALUES(2008,'OMNITECHRETAIL',5);
INSERT INTO SUPPLIER VALUES(2009,'PUMA',5);
INSERT INTO SUPPLIER VALUES(2011,'BOAT',5);

INSERT INTO CATEGORY VALUES(3000,'ELECTRONICS');
INSERT INTO CATEGORY VALUES(3001,'FASHION-MEN');
INSERT INTO CATEGORY VALUES(3002,'FASHION-WOMEN');
INSERT INTO CATEGORY VALUES(3003,'SPORTS');
INSERT INTO CATEGORY VALUES(3004,'BOOKS');

INSERT INTO PRODUCT VALUES(201,'NIKE MERCURIAL
CLEATS',5350,2000,3003);
INSERT INTO PRODUCT VALUES(202,'Philips Speakers',3999,2001,3000);
INSERT INTO PRODUCT VALUES(203,'Lenovo Ideapad
330',29990,2002,3000);
INSERT INTO PRODUCT VALUES(204,'JBL T110BT',2199,2002,3000);
INSERT INTO PRODUCT VALUES(205,'FOSSIL WATCH',7995,2003,3001);
INSERT INTO PRODUCT VALUES(206,'ADIDAS NEO',2699,2004,3001);
INSERT INTO PRODUCT VALUES(207,'LEVIS JEANS',1049,2005,3001);
INSERT INTO PRODUCT VALUES(208,'DUCATI MEN T-
SHIRT',1199,2004,3001);
INSERT INTO PRODUCT VALUES(209,'CARLTON LONGDON
WEDGES',2795,2004,3002);

INSERT INTO PRODUCT VALUES(210,'Ray-Ban Aviator
Sunglasses',7800,2004,3002);
INSERT INTO PRODUCT VALUES(211,'Nivea Bath and Body
set',1249,2005,3002);
INSERT INTO PRODUCT VALUES(212,'DRESS BERRY SOLID
TOP',399,2005,3002);
INSERT INTO PRODUCT VALUES(213,'BOAT Headphones',1099,2011,3000);
INSERT INTO PRODUCT VALUES(214,'Database Systems -
Korth',659,2006,3004);
INSERT INTO PRODUCT VALUES(215,'The Alchemist',249,2006,3004);
INSERT INTO PRODUCT VALUES(216,'S. Brown - Digital
Systems',799,2007,3004);
```

```
INSERT INTO PRODUCT VALUES(217,'Kimberly - Circuit
Analysis',489,2007,3004);
INSERT INTO PRODUCT VALUES(218,'KRX HOME GYM
KIT',2125,2008,3003);
```

USE OF PROCEDURES:

```
CREATE OR REPLACE PROCEDURE "INSERTS"
(FNAME IN VARCHAR2,
 MNAME IN VARCHAR2,
 LNAME IN VARCHAR2,
 CUST_ID IN NUMBER,
 POSTAL_ID IN NUMBER,
 CITY IN VARCHAR2,
 PH_NO IN NUMBER)
 IS
 BEGIN
INSERT INTO CUSTOMER
VALUES(FNAME,,MNAME,LNAME,CUST_ID,POSTAL_ID,CITY,PH_NO);
END;
 /
```

The above procedure is used to insert data into customer table through REGISTER GUI.
The procedure is invoked in the following way:

CallableStatement smt=con.prepareCall("{call INSERTS(?,?,?,?,?,?,?)}");

USE OF TRIGGER:

```
CREATE OR REPLACE TRIGGER INS_DEL
BEFORE DELETE ON CART
FOR EACH ROW
BEGIN
INSERT INTO PRODUCTS_SOLD VALUES (:OLD.prod_id,:OLD.PTOT_PRICE);
END;
/
```

The above trigger is used to populate the table PRODUCTS_SOLD soon after a customer logs out after shopping and as his cart gets empty.

SIMPLE AND COMPEX QUERIES USED

LOGIN PAGE:

select pass from account where email_id= +email_id;

ELECTRONICS JTABLE DISPLAY:

select prod_id,prod_name,price,sup_name from product natural join supplier
where cat_id in (select cat_id from category where cat_name = 'ELECTRONICS';

SPORTS JTABLE DISPLAY:

select prod_id,prod_name,price,sup_name from product natural join supplier
where cat_id in (select cat_id from category where cat_name = 'SPORTS’;

MEN’S FASHION JTABLE

select prod_id,prod_name,price,sup_name from product natural join supplier
where cat_id in (select cat_id from category where cat_name = 'FASHION-MEN’;

WOMEN’S FASHION JTABLE

select prod_id,prod_name,price,sup_name from product natural join supplier
where cat_id in (select cat_id from category where cat_name = 'FASHION-
WOMEN’;

BOOKS’S FASHION JTABLE

select prod_id,prod_name,price,sup_name from product natural join supplier
where cat_id in (select cat_id from category where cat_name = 'BOOKS’;

TO DISPLAY PRODUCTS IN CART:

select prod_id,prod_name,price from product where prod_id in (select prod_id from cart);

SHIPPER DETAILS

select ddate as dddate,rating,ship_name from Shipper NATURAL JOIN XOrder;

TO INSERT SELECTED PRODUCT IN CART

insert into cart values(prod_id,price);

TO CALCULATE THE FINAL AMOUNT TO BE PAID

select sum(ptot_price) from cart;

TO CLEAR THE CART AFTER CUSTOMER LOGS OUT

DELETE FROM CART;

DELETE FROM XOrder;

JAVA CODE FOR FUNCTIONAL DESIGN (JDBC AND DB ACCESS)

TO ESTABLISH CONNECTION BETWEEN JAVA AND DATABASE:

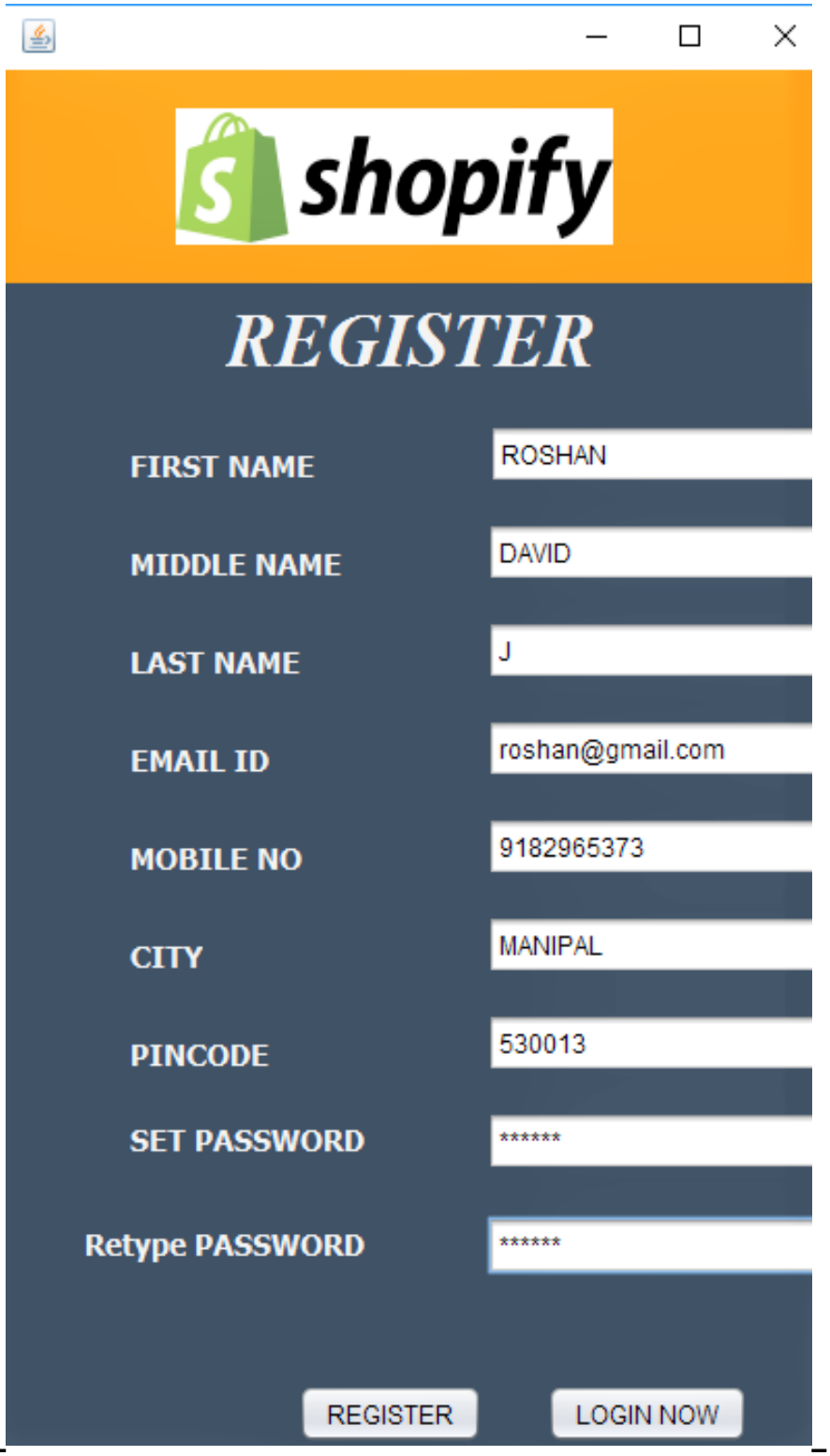
```
try {
        // step1 load the driver class
          Class.forName("oracle.jdbc.driver.OracleDriver");
           // step2 create the connection object
            Connection con;
          con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE",
"suryansh",
                    "computers");
            // step3 create the statement object
             Statement stmt = con.createStatement();
             // step4 execute query
             ResultSet rs = stmt.executeQuery(q); //q is your query
             System.out.println("connected");
               while (rs.next())
                        {
                           lblDetails.setText(lblDetails.getText() + " " + rs.getInt(1) + "  "
+ rs.getString(2));
                        }
                        // step5 close the connection object

                    con.close();

} catch (SQLException | ClassNotFoundException ex) {

                        System.out.println(ex);
                }
```

NAVIGATION BETWEEN ONE JFRAME TO ANOTHER:

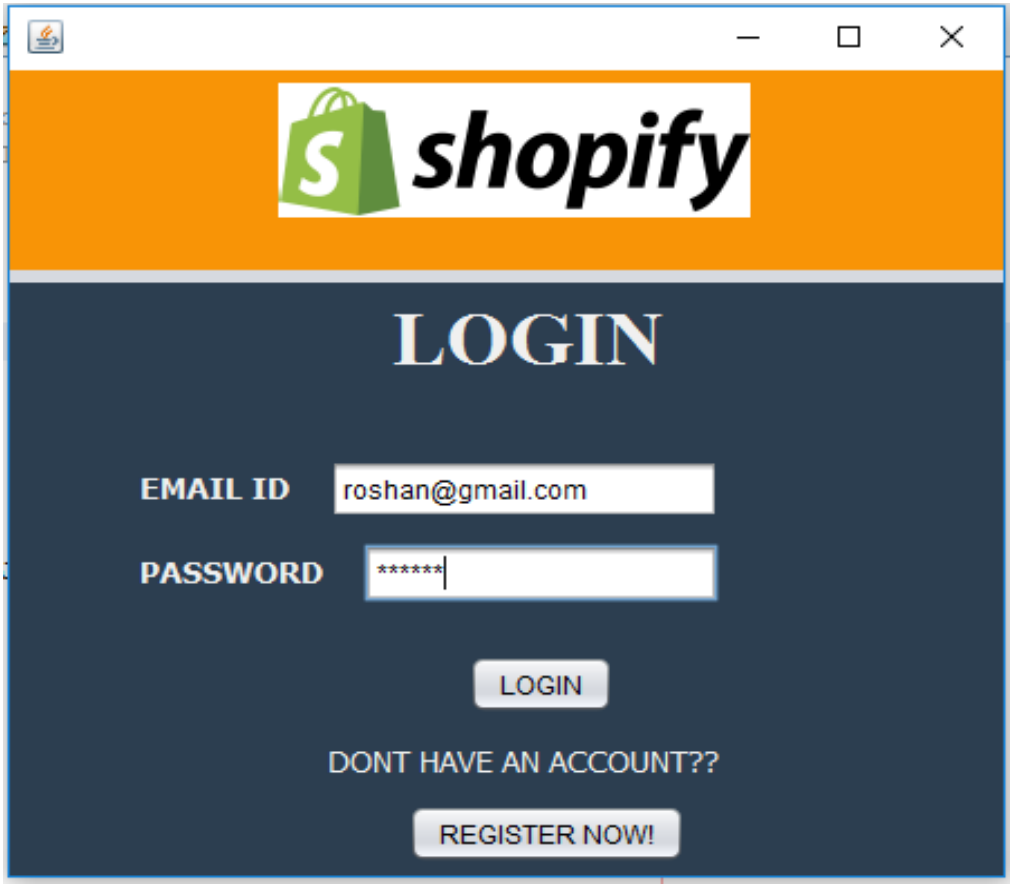
Eg: NAVIGATION TO SHOPPING PORTAL FROM INDIVIDUAL CATEGORIES-

```
private void GoToCat4ActionPerformed(java.awt.event.ActionEvent
evt) {

    Shopping shop = new Shopping();
    shop.setVisible(true);
    shop.pack();
    shop.setLocationRelativeTo(null);
    //b.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.dispose();

}
```

UI SCREENSHOTS




shopify
LOGIN
EMAIL ID
andaman@gmail.com
PASS
Message
Email ID not in database
OK
REGISTER NOW!


shopify
LOGIN
EMAIL ID
roshan@gmail.com
PASSWORD
******
LOGIN
DONT HAVE AN ACCOUNT??
REGISTER NOW!

shopify

SHOPPING PORTAL

ELECTRONICS

BOOKS

SPORTS

FASHION (MEN)

FASHION (WOMEN)

shopify

ELECTRONICS

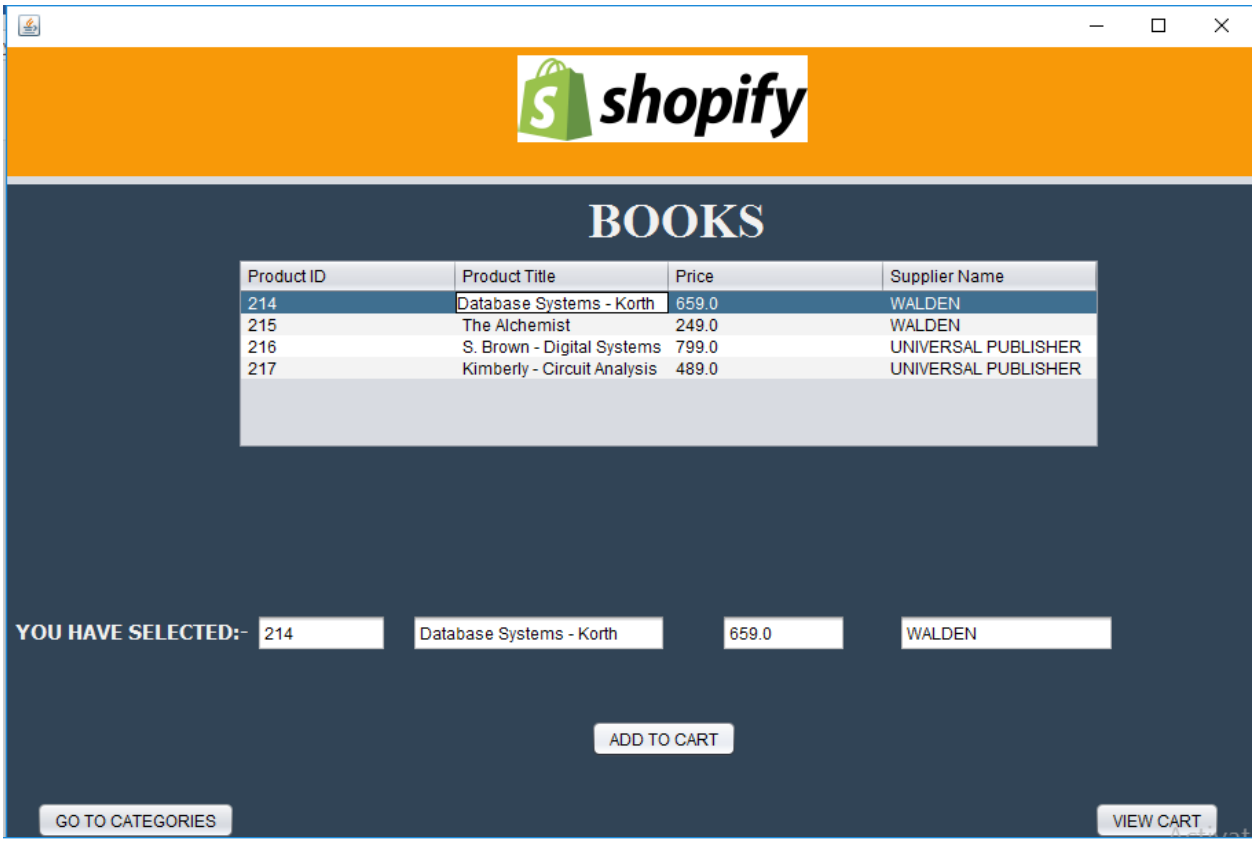
Product ID	Product Title	Price	Supplier Name
202	Philips Speakers	3999.0	PHILIPS
203	Lenovo Ideapad 330	29990.0	GADA ELECTRONICS
204	JBL T110BT	2199.0	GADA ELECTRONICS
213	BOAT Headphones	1099.0	BOAT

YOU HAVE SELECTED:- 203 Lenovo Ideapad 330 Rs 29990.0 GADA ELECTRONICS

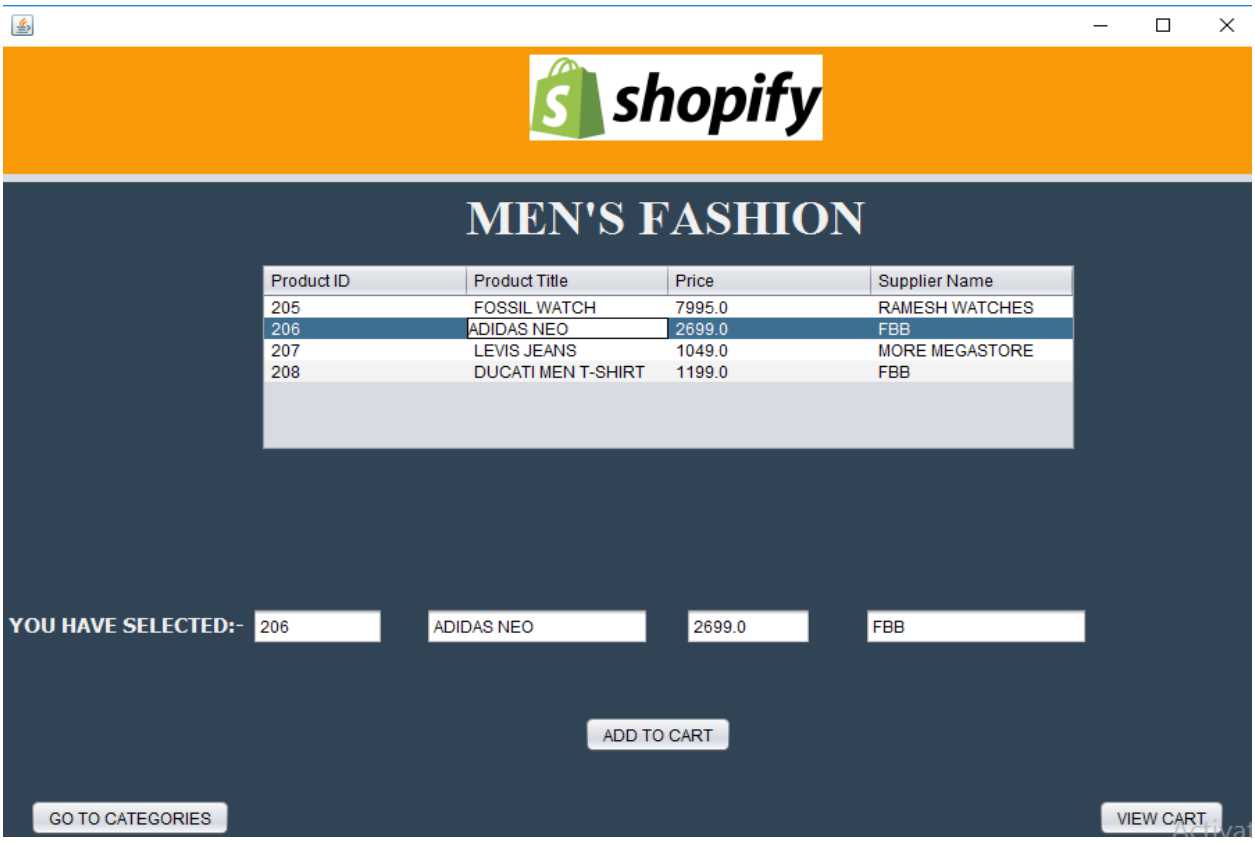
ADD TO CART

GO TO CATEGORIES

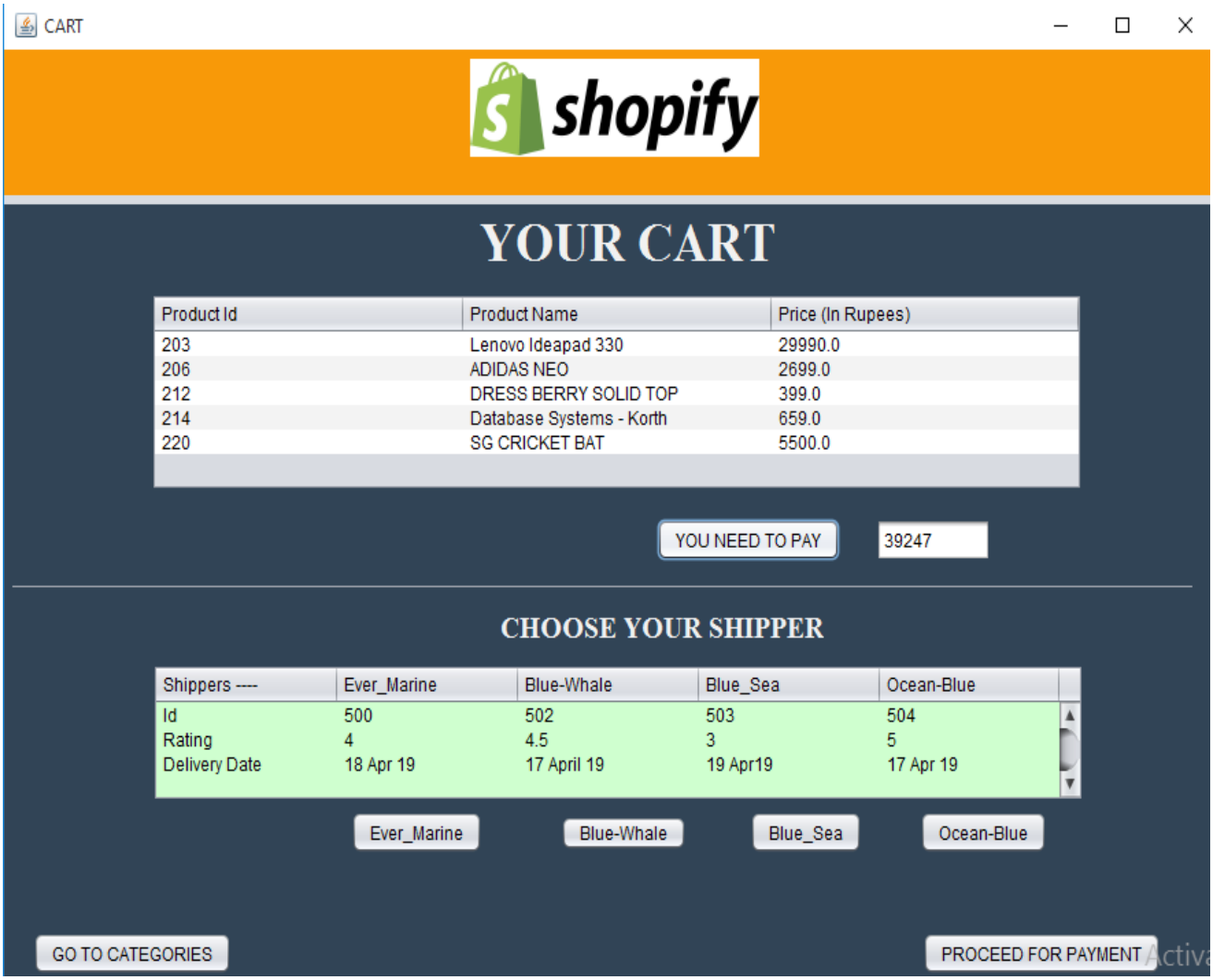
VIEW CART


shopify
BOOKS
Product ID
Product Title
Price
Supplier Name
214
Database Systems - Korth
659.0
WALDEN
215
The Alchemist
249.0
WALDEN
216
S. Brown - Digital Systems
799.0
UNIVERSAL PUBLISHER
217
Kimberly - Circuit Analysis
489.0
UNIVERSAL PUBLISHER
YOU HAVE SELECTED:-
214
Database Systems - Korth
659.0
WALDEN
ADD TO CART
GO TO CATEGORIES
VIEW CART

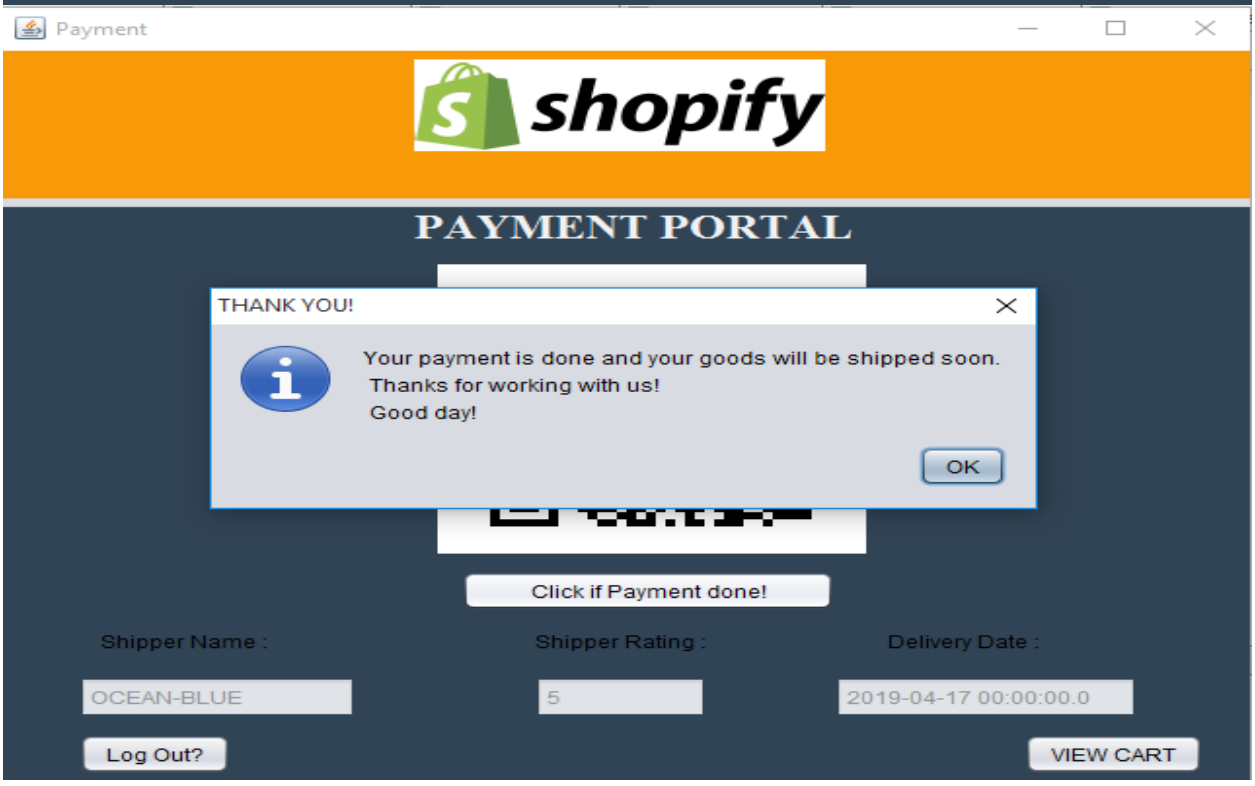

shopify
SPORTS
Product ID
Product Title
Price
Supplier Name
201
NIKE MERCURIAL CL...
5350.0
NIKE
218
KRX HOME GYM KIT
2125.0
OMNITECHRETAIL
219
Puma 19.1 Bowling S...
5999.0
PUMA
220
SG CRICKET BAT
5500.0
OMNITECHRETAIL
YOU HAVE SELECTED:-
220
SG CRICKET BAT
5500.0
OMNITECHRETAIL
ADD TO CART
GO TO CATEGORIES
VIEW CART


shopify
MEN'S FASHION
Product ID
Product Title
Price
Supplier Name
205
FOSSIL WATCH
7995.0
RAMESH WATCHES
206
ADIDAS NEO
2699.0
FBB
207
LEVIS JEANS
1049.0
MORE MEGASTORE
208
DUCATI MEN T-SHIRT
1199.0
FBB
YOU HAVE SELECTED:-
206
ADIDAS NEO
2699.0
FBB
ADD TO CART
GO TO CATEGORIES
VIEW CART


shopify
WOMEN'S FASHION
Product ID
Product Title
Price
Supplier Name
209
CARLTON LONGDON WED...
2795.0
FBB
210
Ray-Ban Aviator Sunglasses
7800.0
FBB
211
Nivea Bath and Body set
1249.0
MORE MEGASTORE
212
DRESS BERRY SOLID TOP
399.0
MORE MEGASTORE
YOU HAVE SELECTED:-
212
DRESS BERRY SOLID TOP
399.0
MORE MEGASTORE
ADD TO CART
GO TO CATEGORIES
VIEW CART


CART
shopify
YOUR CART
Product Id
Product Name
Price (In Rupees)
203
Lenovo Ideapad 330
29990.0
206
ADIDAS NEO
2699.0
212
DRESS BERRY SOLID TOP
399.0
214
Database Systems - Korth
659.0
220
SG CRICKET BAT
5500.0
YOU NEED TO PAY
39247
CHOOSE YOUR SHIPPER
Shippers ----
Ever_Marine
Blue-Whale
Blue_Sea
Ocean-Blue
Id
500
502
503
504
Rating
4
4.5
3
5
Delivery Date
18 Apr 19
17 April 19
19 Apr19
17 Apr 19
Ever_Marine
Blue-Whale
Blue_Sea
Ocean-Blue
GO TO CATEGORIES
PROCEED FOR PAYMENT


Payment
shopify
PAYMENT PORTAL
Click if Payment done!
Shipper Name :
Shipper Rating :
Delivery Date :
Log Out?
VIEW CART


Payment
shopify
PAYMENT PORTAL
THANK YOU!
Your payment is done and your goods will be shipped soon.
Thanks for working with us!
Good day!
OK
Click if Payment done!
Shipper Name :
Shipper Rating :
Delivery Date :
OCEAN-BLUE
5
2019-04-17 00:00:00.0
Log Out?
VIEW CART

References :

https://www.youtube.com/playlist?list=PLJ5y5njLJrx-PYJfONhCV4nNRmj7vJ-A4

Database Systems Manual 2018-19

TutorialsPoint