ONLINESHOPPING –SQL CONNECTIVITY USING

JDBC

*A*

*Report*

*Submitted in partial fulfillment of the Requirements for the award of the Degree of*

BACHELOR OF ENGINEERING

IN

# INFORMATION TECHNOLOGY

By

ANJAN SAI <1602-20-737-006 > Under the guidance of Ms B. Leelavathy

Department of Information Technology Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31

# BONAFIDE CERTIFICATE

This is to certify that this project report titled ‘**ONLINE SHOPPING**’ is a project work of Anjan Sai bearing roll no. 1602-20-737-006 who carried out the project under my supervision in the IV semester for the academic year 2021- 2022.

Signature Signature

Internal Examine External Examiner

**ABSTRACT**

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform.

This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains.

If shops are providing an online portal where their customers can enjoy

easy shopping from anywhere, the shops won’t be losing any more

customers to the trending online shops such as flipcart or ebay. Since the application is available in the Smartphone it is easily accessible and always available.

**Requirement Analysis**

**List of Tables:**

LOGIN TABLE

USER DETAILS

PRODUCT DETAILS

PRODUCT ORDERS

ARCHITECTURE AND TECHNOLOGY

**Software used:** Java Notepad++, Oracle 11g Database, Java SE version 13, SQL\*Plus.

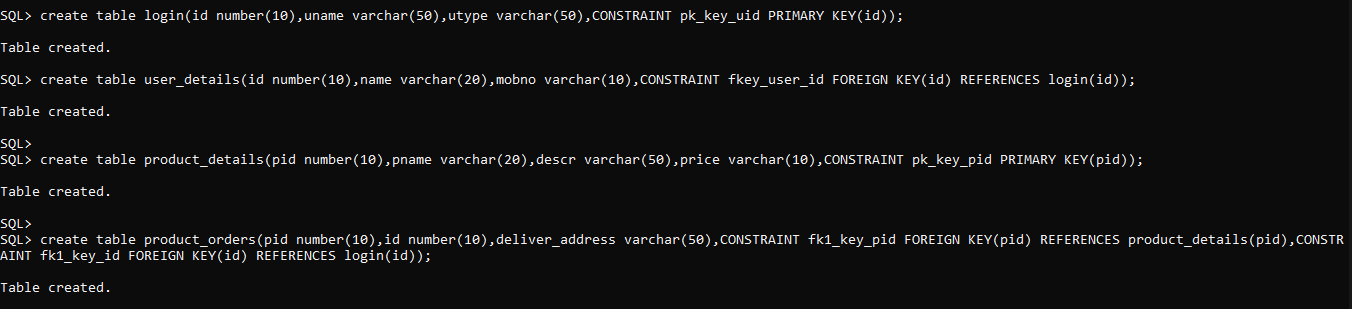
**Java SWINGS:**

**Java SWINGS** is an API to develop GUI or window-based applications injava. Java SWING components are platform-independent. It is lightweight. The javax.swing package provides classes for SWING API such as JTextField, JLabel, JTextArea, JButton, JChoice, JList etc.

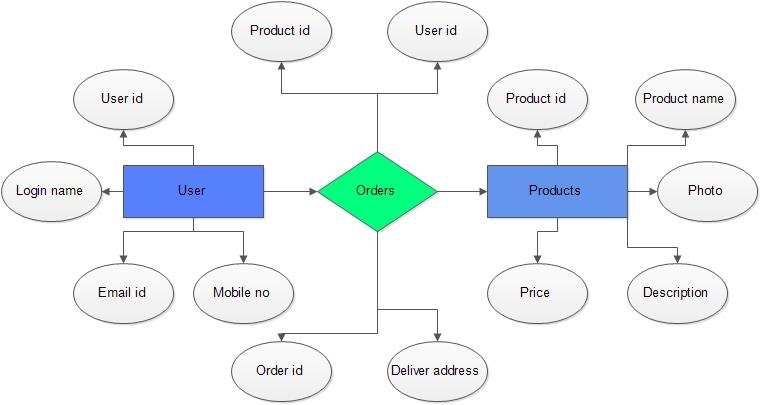
**SQL:**

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

Creation of tables



**E-R DIAGRAM**



IMPLEMENTATION:

## Front end programs and its connectivity

**Java Database Connectivity** (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

**public void** connectToDB(){

**try** {

Connection con = DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:xe",”amruth","vasavi");

}

**catch** (SQLException connectException){

System.***out***.println(connectException.getMessage()); System.***out***.println(connectException.getSQLState()); System.***out***.println(connectException.getErrorCode()); System.*exit*(1);

}

}

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

**PROGRAM:**

import java.awt.BorderLayout;

import java.awt.Font;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import java.awt.BorderLayout;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import java.awt.List;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JButton;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextArea;

importjavax.swing.table.Default TableModel;

public interface CatalogHome extends EJBHome

{

public Catalog create() throws RemoteException,CreateException;

}

public class ConfirmBean implements SessionBean,SessionSynchronization

{

Quotation quotation;

QuotationHome quotationHome;

SessionContext scontext;

Context context;

FileInputStream fis;

ObjectInputStream ois;

Handle myhandle;

Object key;

String value;

int totalPrice;

StringTokenizer tok;

StringBuffer pnames;

Connection con;

String uid;

int crbal;

UserTransaction utx;

Hashtable htab;

public Context getInitialContext() throws Exception

{

Hashtable htab=new Hashtable();

htab.put(Context.INITIAL\_CONTEXT\_FACTORY,"weblogic.jndi.WLInitialContextFactory");

//htab.put(Context.PROVIDER\_URL,"t3://localhost:7001");

return new InitialContext(htab);

}

public void ejbCreate() throws CreateException

{

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

catch(Exception e)

{

e.printStackTrace();

}

System.out.println(" in the create of the ConfirmBean ");

}

public void setSessionContext(SessionContext con)

{

scontext=con;

}

public void ejbActivate()

{

System.out.println(" in the ejb activate of the Confirm bean");

}

public void ejbPassivate()

{

System.out.println(" in the ejb passiavate of the Confirm bean");

}

public void ejbRemove()

{

System.out.println(" in the ejb remove of the Confirm bean");

}

public String doTransaction(String lid,String log,String pas)

{

// context=getInitialContext();

try

{

fis=new FileInputStream("equota.ser");

ois=new ObjectInputStream(fis);

myhandle=(Handle)ois.readObject();

quotation=(project.server.Quotation)myhandle.getEJBObject();

ois.close();

fis.close();

htab=quotation.getAllProducts();

Enumeration keys=htab.keys();

System.out.println(" got the hatab enum the --- size is "+htab.size());

pnames=new StringBuffer();

while(keys.hasMoreElements())

{

key=keys.nextElement();

value=(String)htab.get(key);

tok=new StringTokenizer(value,":");

tok.hasMoreElements();

pnames.append(tok.nextToken());

pnames.append(",");

totalPrice+=Integer.parseInt(tok.nextToken());

}

uid=uniqueID();

System.out.println(" after the uniquid ");

crbal=getACBalance(lid);

System.out.println(" after the acbalance ");

utx=(javax.jts.UserTransaction)context.lookup("javax.jts.UserTransaction");

con=DriverManager.getConnection("jdbc:odbc:ejb","scott","tiger");

//utx.begin();

PreparedStatement pstmt1=con.prepareStatement("update account1 set crbal=crbal-"+totalPrice+" where logid=?");

pstmt1.setString(1,lid);

pstmt1.executeUpdate();

System.out.println(" after stmt 1 ");

PreparedStatement pstmt2=con.prepareStatement("update cash set cashinhand=cashinhand+"+totalPrice);

pstmt2.executeUpdate();

System.out.println(" after stmt 2 ");

PreparedStatement pstmt3=con.prepareStatement("insert into trans\_products values (?,?,?,?,sysdate)");

pstmt3.setString(1,uid);

pstmt3.setString(2,lid);

pstmt3.setString(3,pnames.toString());

pstmt3.setInt(4,totalPrice);

pstmt3.executeUpdate();

System.out.println(" after stmt 3 ");

PreparedStatement pstmt4=con.prepareStatement("insert into trans values(?,sysdate,?)");

pstmt4.setString(1,lid);

pstmt4.setInt(2,totalPrice);

pstmt4.executeUpdate();

System.out.println(" after stmt 4 ");

con.close();

}

catch(Exception e)

{e.printStackTrace();

if (utx!=null)

utx.rollback();

}

return uid;

}

public String uniqueID()

{

String strid=null;

try

{

con=DriverManager.getConnection("jdbc:odbc:ejb","scott","tiger");

PreparedStatement stmt=con.prepareStatement("select count(\*)+100 from trans");

ResultSet res=stmt.executeQuery();

res.next();

int id=res.getInt(1);

strid="qt"+String.valueOf(id);

con.close();

}

catch(Exception e)

{

e.printStackTrace();

}

return strid;

}

public int getACBalance(String lid)

{

int bal=0;

try

{

System.out.println(" in the get acbal ");

context=getInitialContext();

IAccountHome iah=(IAccountHome)context.lookup("server.IAccountHome");

if ( iah==null)

{

System.out.println(" the home is nulll --- in the get acbal ");

}

IAccount ia=(IAccount)iah.findByPrimaryKey(lid);

if ( ia==null)

{

System.out.println(" the remote is nulll --- in the get acbal ");

}

bal=ia.getCrBal();

System.out.println(" after getCrBal --- in the get acbal ");

}

catch(Exception ex)

{

ex.printStackTrace();

}

return bal;

}

public void afterBegin()

{

System.out.println(" after begin of the transactions ");

}

public void beforeCompletion()

{

System.out.println(" before compeletion of the transaction in the confirm bean");

}

public void afterCompletion(boolean flag)

{

if (flag==true)

System.out.println(" after compeletion -- the transaction is successful ");

else

System.out.println(" after compeletion -- the transaction is not successful ");

}

}

public class IProductBean implements EntityBean

{

EntityContext context;

public String pid,pname,capacity,um,producer;

public int pprice;

public void setEntityContext (EntityContext con)

{

context=con;

}

public void unsetEntityContext ()

{

context=null;

}

public String ejbCreate(String pid,String pname,int pprice,String capacity,String UM,String producer) throws CreateException

{

System.out.println(" in ejb create of the iproduct ");

return null;

}

public void ejbRemove()

{

System.out.println(" in the ejb remove of the iproduct ");

}

public void ejbLoad()

{

System.out.println(" in the ejb load of the iproduct ");

}

public void ejbStore()

{

System.out.println(" in the ejb store of the iproduct ");

}

public void ejbActivate()

{

System.out.println(" in the ejb activate of the iproduct ");

}

public void ejbPassivate()

{

System.out.println(" in the ejb passivate of the iproduct ");

}

public void ejbPostCreate(String pid,String pname,int pprice,String capacity,String UM,String producer)

{

System.out.println(" in the ejb post create of the iproduct ");

}

public String getID()

{

return pid;

}

public String getName()

{

return pname;

}

public int getPrice()

{

return pprice;

}

public String getCapacity()

{

return capacity;

}

public String getUM()

{

return um;

}

public String getProducer()

{

return producer;

}

}

public class QuotationBean implements SessionBean

{

public Hashtable products;

private SessionContext context;

public void addProduct(String spid,int price,String pname) throws NameAlreadyExistException

{ Object pid=(Object)spid;

String nampri;

nampri=pname+":"+String.valueOf(price);

System.out.println(" in the addition of the product - "+pid +" having"+nampri);

if(products.containsKey(pid))

{

throw new NameAlreadyExistException("produt you want to buy is Already in quotation ");

}

else

{

products.put(pid,(Object)nampri);

System.out.println(" added the item to the quotation < -- > no of products are"+products.size());

}

}

public void removeProduct(String pid) throws NameNotFoundException

{

Object flag=null;

flag=products.remove((Object)pid);

if(flag==null)

{

throw new NameNotFoundException("Name of the product Not Found");

}

}

public Hashtable getAllProducts()

{

return products;

}

public void ejbCreate() throws CreateException

{

products=new Hashtable();

}

public void setSessionContext(SessionContext sc)

{

context=sc;

}

public void ejbActivate()

{

System.out.println("in ejb activate of the quotation.....");

}

public void ejbPassivate()

{

System.out.println("in ejb passivate of the quotation....");

}

public void ejbRemove()

{

System.out.println("in ejb remove of the quotation.....");

}

}

public class IAccountBean implements EntityBean

{

EntityContext context;

public String logid,login,passwd;

public int crbal,dbbal;

public Date doj;

public String ejbCreate(String logid,String login,String passwd,int crbal,int dbbal,Date doj)throws CreateException

{

this.logid=logid;

this.login=login;

this.passwd=passwd;

this.crbal=crbal;

this.dbbal=dbbal;

this.doj = doj;

System.out.println(" in the ejb create of the Individual account");

return null;

}

public void setEntityContext (EntityContext con)

{

context=con;

}

public void unsetEntityContext ()

{

context=null;

}

public void ejbRemove()

{

System.out.println(" in the ejb remove of the iaccount ");

}

public void ejbLoad()

{

System.out.println(" in the ejb load of the iaccount ");

}

public void ejbStore()

{

System.out.println(" in the ejb store of the iaccount ");

}

public void ejbActivate()

{

System.out.println(" in the ejb activate of the iaccount ");

}

public void ejbPassivate()

{

System.out.println(" in the ejb passivate of the iaccount ");

}

public void ejbPostCreate(String logid,String login,String passwd,int crbal,int dbbal,Date doj)

{

System.out.println(" in the ejb post create of the iaccount ");

}

public int getCrBal()

{

return crbal;

}

public int getDbBal()

{

return dbbal;

}

public Date getDOJ()

{

return doj;

}

}

public interface IAccountHome extends EJBHome

{

public IAccount create(String logid,String login,String passwd,int crbal,int dbbal,Date doj) throws RemoteException,CreateException;

public IAccount findByPrimaryKey(String loginid) throws RemoteException,FinderException;

//public java.util.Enumeration findNullId() throws RemoteException,FinderException;

}

public interface IProductHome extends EJBHome

{

public IProduct create(String pid,String pname,int pprice,String capacity,String UM,String producer) throws RemoteException,CreateException;

public IProduct findByPrimaryKey(String pid) throws RemoteException,FinderException;

//public java.util.Enumeration findNullId() throws RemoteException,FinderException;

}

public class NameAlreadyExistException extends Exception

{

public NameAlreadyExistException() { }

public NameAlreadyExistException(String msg)

{

super(msg);

}

}

public class NameNotFoundException extends Exception

{

public NameNotFoundException() { }

public NameNotFoundException(String msg)

{

super(msg);

}

}

public interface Catalog extends EJBObject

{

public ResultSet getProducts() throws RemoteException;

}

public interface Quotation extends EJBObject

{

public void addProduct(String pid,int price,String pname) throws NameAlreadyExistException,RemoteException;

public void removeProduct(String pid) throws RemoteException,NameNotFoundException;

public Hashtable getAllProducts() throws RemoteException;

}

public interface QuotationHome extends EJBHome

{

public Quotation create() throws RemoteException,CreateException;

}

public class testing

{

public static void main(String str[])

{

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:raja","rajendra","sankara");

Statement stmt=con.createStatement();

ResultSet res=stmt.executeQuery("select hiredate from emp where empno=7788");

res.next();

System.out.println(" the firedate is "+res.getDate(1));

}

catch(Exception ed)

{

ed.printStackTrace();

}

}

}

public interface IProduct extends EJBObject

{

public String getID() throws RemoteException;

public String getName() throws RemoteException;

public int getPrice()throws RemoteException;

public String getCapacity()throws RemoteException;

public String getUM()throws RemoteException;

public String getProducer()throws RemoteException;

}

public interface ConfirmHome extends EJBHome

{

public Confirm create() throws RemoteException,CreateException;

}

public interface IAccount extends EJBObject

{

public int getCrBal() throws RemoteException;

public int getDbBal() throws RemoteException;

public Date getDOJ()throws RemoteException;

}

public class CatalogBean implements SessionBean

{

SessionContext context;

public void ejbCreate() throws CreateException

{

System.out.println(" in the create of CatalogBean ");

}

public ResultSet getProducts()

{

ResultSet res=null;

try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:ejb","scott","tiger");

Statement stmt=con.createStatement();

res=stmt.executeQuery("select \* from product");

}

catch(Exception e)

{

System.out.println(" an error while executing the business mehtod of catalog bean");

e.printStackTrace();

}

return res;

}

public void setSessionContext(SessionContext ctx)

{

context=ctx;

}

public void ejbActivate()

{

System.out.println(" in the catalog bean's Activate");

}

public void ejbPassivate()

{

System.out.println(" in the catalog bean's passivate");

}

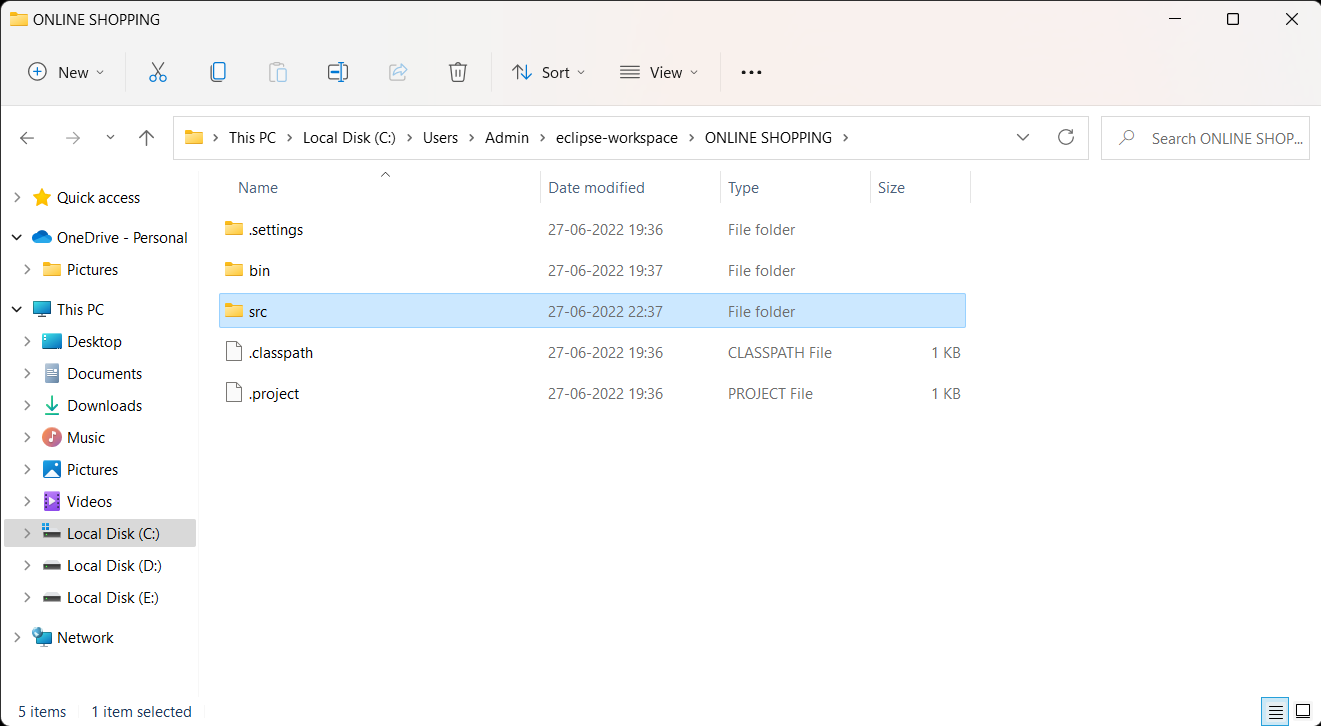
public void ejbRemove()

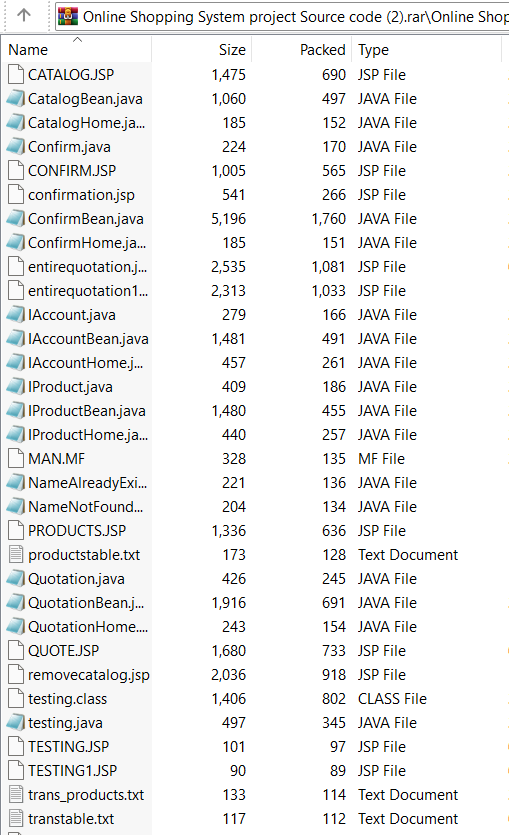
{

System.out.println(" in the catalog bean's remove");

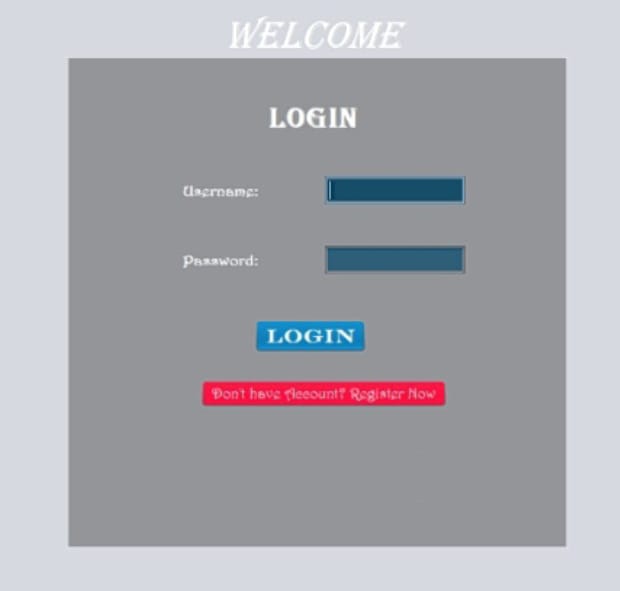
}

}

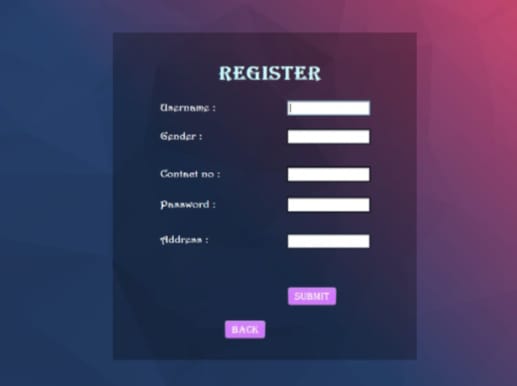




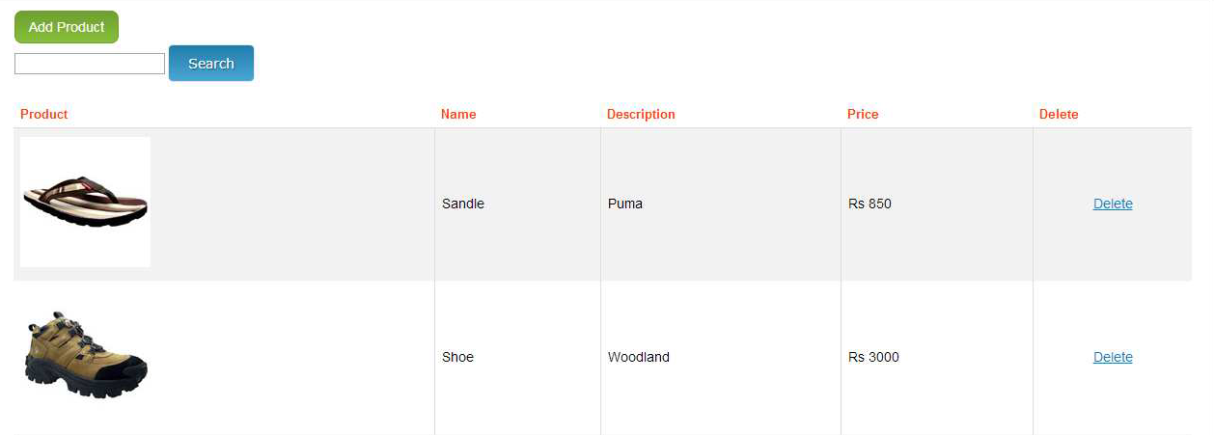
LOGIN:



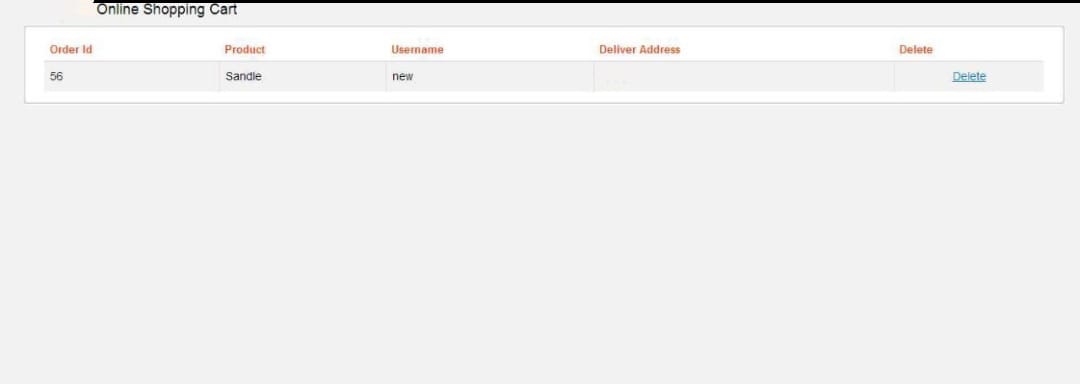
REGISTRATION



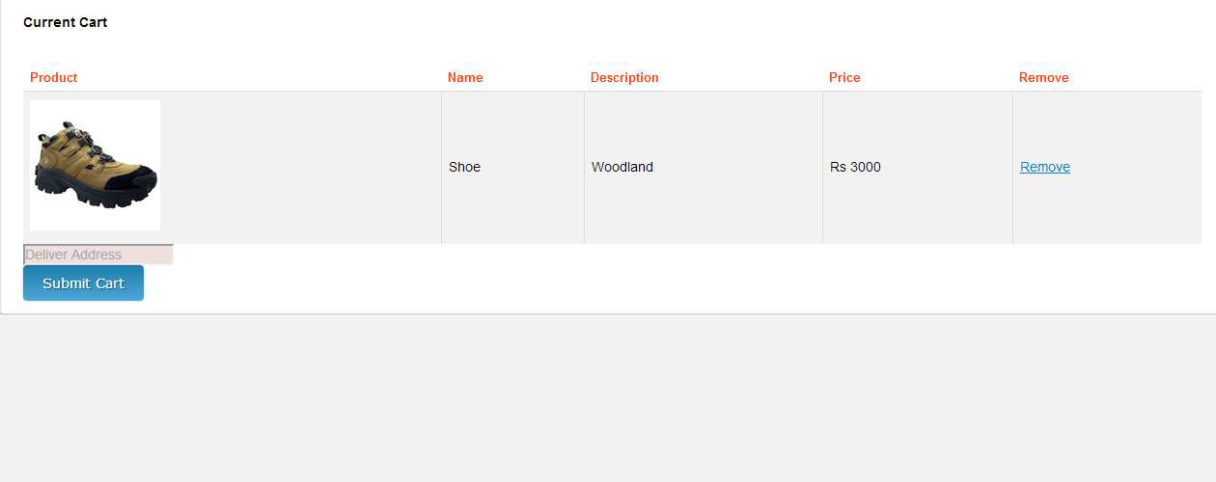
PRODUCTS

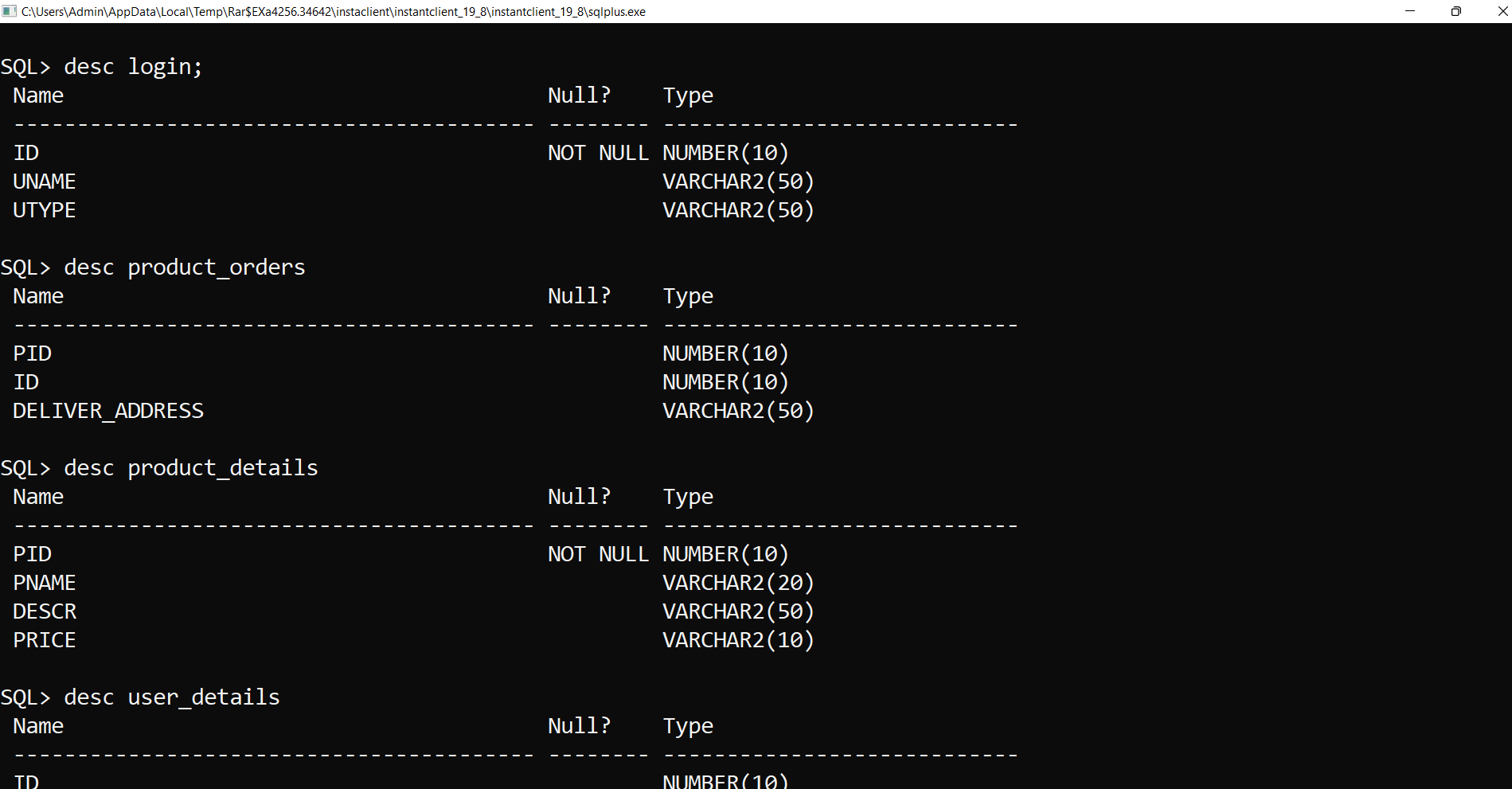


ORDER



USER CART





**Result:**

I successfully completed this DBMS PROJECT “ONLINE SHOPPING”.

**Discussion and Future work**

While doing this project I got new ideas I understood how to work on projects. Now to further extend this project I want to create a android app by which I can control my project on my hand and connect to it.

**References:**

https://www.academia.edu/36893248/Ramakrishnan\_\_Database\_Management\_Systems\_3rd\_Edition

https://docs.oracle.com/javase/7/docs/index.html

https://www.javatpoint.com/dbms-tutorial http://www.sqlines.com/articles/java/sql\_server\_jdbc\_connection https://netbeans.apache.org/