JAVA AWT BASED - AMAZON LIKE COMMERCIAL SITE CART DATA MANAGEMENT PLATFORM - 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: **K.KUSUMA<1602-18-737-312>**

UNDER THE GUIDANCE OF B.LEELAVATHY

2020

Department of Information Technology

VASAVI COLLEGE OF ENGINEERING   AUTONOMOUS (AFFLIATED TO O.U) IBRAHIMBAGH, HYDERABAD-500031

BONAFIDE CERTIFICATE

This to Certify that the project report titled "
**AMAZON LIKE COMMERCIAL SITE CART
DATA MANAGEMENT PLATFORM - **" project
work of Miss. K.KUSUMA bearing Roll no: 1602-
18-737-312 who carried out this project under my
supervision in the IV Semester the academic year
2019-2020.

Signature .   **Signature.**

External Examiner   *Internal Examiner*

ABSTRACT

Title:Amazon like commercial site cart data management platform

An web service system that permits a customer to submit online orders for items and/or services from a store that serves both walk-in customers and online customers.In commercial sites any one can buy a product and sell a product this are two users are available .buyer can search for product and customer can place order to buy a product for the site customer can also give rating for product and provided service.Once the customer decides to submit a purchase order, the customer may print the contents of the virtual shopping basket in order to obtain a hard copy record of the transaction. This project has 7 tables this tables describes how the user can select product and places a order and selling products.

a).AIM AND PRIORITY OF THE PROJECT:

To create a GUI based form for the project of **AMAZON LIKE**

COMMERCIAL SITE CART DATA MANAGEMENT PLATFORM

Where the commercial site cart data management platform we can shop on this main objective of the Online Shopping System is to manage the details of Shopping,Internet,Payment, she Bills,Customer. It manages all the information about Shopping, Products, Customer, Shopping.

The values entered (insertion, updation, deletion)by the user for

respective table in **GUI** should be updated in the database using

JDBC.

b) REQUIREMENTS ANALYSIS

List of tables:

- Buyer
- Seller
- Sell
- Product
- Search

Table name	Attribute	Domain
1.Buyer	1.bid 2.bname 3.address 4.phoneno	varchar2(20) char(40) varchar2(50) number(10)
2. Seller	1.Sid 2.Sname 3.Rating	varchar2(20) char (50) number(10)
3.Sell	1.Sid 2.Pid 3.Date of join	varchar2(10) char(20) Date
4.Product	1.Pid 2.Pname 3.product description	varchar2(30) char(30) varchar2(50)
4.Search	1.Pid 2.Bid	Varchar2(30) Varchar2(20)

c)Architecture and Technology used

Java Eclipse, Oracle 11g Database, java SE version 8, SQL *plus ,java AWT

Eclipse: It is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug in system for customizing the environment. The Eclipse software

development kit (SDK), which include java development tools is meant for java developers.

SQL *plus: SQL *plus is a command line tool proprietary to oracle. You can send SQL Queries to the server using the tool.It can also help you format the result of query. SQL is the query language that is used to communicate with the oracle server to access and modify data.

JAVA AWT: Abstract window tool kit is an API to develop GUI or Window based applications in java. Java AWT components are platform dependent i.e components are displayed according to the view of operating system. AWT is heavy weight that is components are using the resources of O.S.

JDBC: Java Database Connectivity is an application programming interface (API) for the programming language java **,**which defines how a client may access database. It is a javabased data access technology used.

d)Data Base Design

Title: Amazon like commercial site cart data management platform

```
Run SQL Command Line
Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> conn kusuma/vasavi
Connected.
SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ -------- ----------
BUYER                          TABLE
PRODUCT                        TABLE
SEARCH                         TABLE
SELL                           TABLE
SELLER                         TABLE

SQL> desc seller;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL VARCHAR2(20)
 SNAME                                              CHAR(50)
 RATING                                             NUMBER(10)

SQL> desc sell;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                                VARCHAR2(20)
 PID                                                VARCHAR2(30)
 DATEOFJOIN                                         DATE

SQL> desc product;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PID                                       NOT NULL VARCHAR2(30)
 PNAME                                              CHAR(30)
 PRODUCTDESCRIPTION                                 VARCHAR2(50)

SQL> desc search;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PID                                                VARCHAR2(30)
 BID                                                VARCHAR2(20)

SQL> desc buyer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BID                                       NOT NULL VARCHAR2(20)
 BNAME                                              CHAR(40)
 ADDRESS                                            VARCHAR2(50)
 PHONENO                                            NUMBER(10)

SQL>
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```
SQL> select * from seller;

SID                      SNAME
------------------- ------------------------------------------------
    RATING
---------
1                        ram
        5

2                        sai
        7

3                        sri
        8


SID                      SNAME
------------------- ------------------------------------------------
    RATING
---------
4                        bob
        9

5                        lack
        9


SQL> select * sell;
select * sell
        *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> select * from sell;

SID                      PID                         DATEOFJOI
------------------- --------------------------- --------
1                        1                           08-AUG-19
2                        5                           06-JAN-00
2                        10                          20-DEC-01

SQL> select * from product;

PID                         PNAME
--------------------------- ----------------------------
PRODUCTDESCRIPTION
-------------------------------------------------
1                             3
```

Type here to search

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```
Select Run SQL Command Line

SQL> select * from product;

PID                             PNAME
------------------------------- -------------------------------
PRODUCTDESCRIPTION
-------------------------------------------------
1                               3
20000

5                               5
600

10                              4
700


PID                             PNAME
------------------------------- -------------------------------
PRODUCTDESCRIPTION
-------------------------------------------------
60                              3
1000


SQL> select * from search;

PID                             BID
------------------------------- --------------------
1                               1
1                               3
5                               3
10                              1

SQL> select * from buyer;

BID                 BNAME
------------------- ---------------------------------------
ADDRESS                                         PHONENO
------------------------------------------------ ----------
1                   sai
srnagar                                         9908807377

2                   tom
agcolony                                        8686613606

3                   ram
nampally                                        9700744464
```

Entity Relationship Diagram

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

ER Diagram:

user

Phone no

Address

Isa

sid

bname

sname

bid

Payment mode

Buyer

Seller

Display bill

Rating

Order

Place

Search

Sell

Date

Totalamout

Product

Product description

pid

pname

k.kusuma
1602-18-737-312
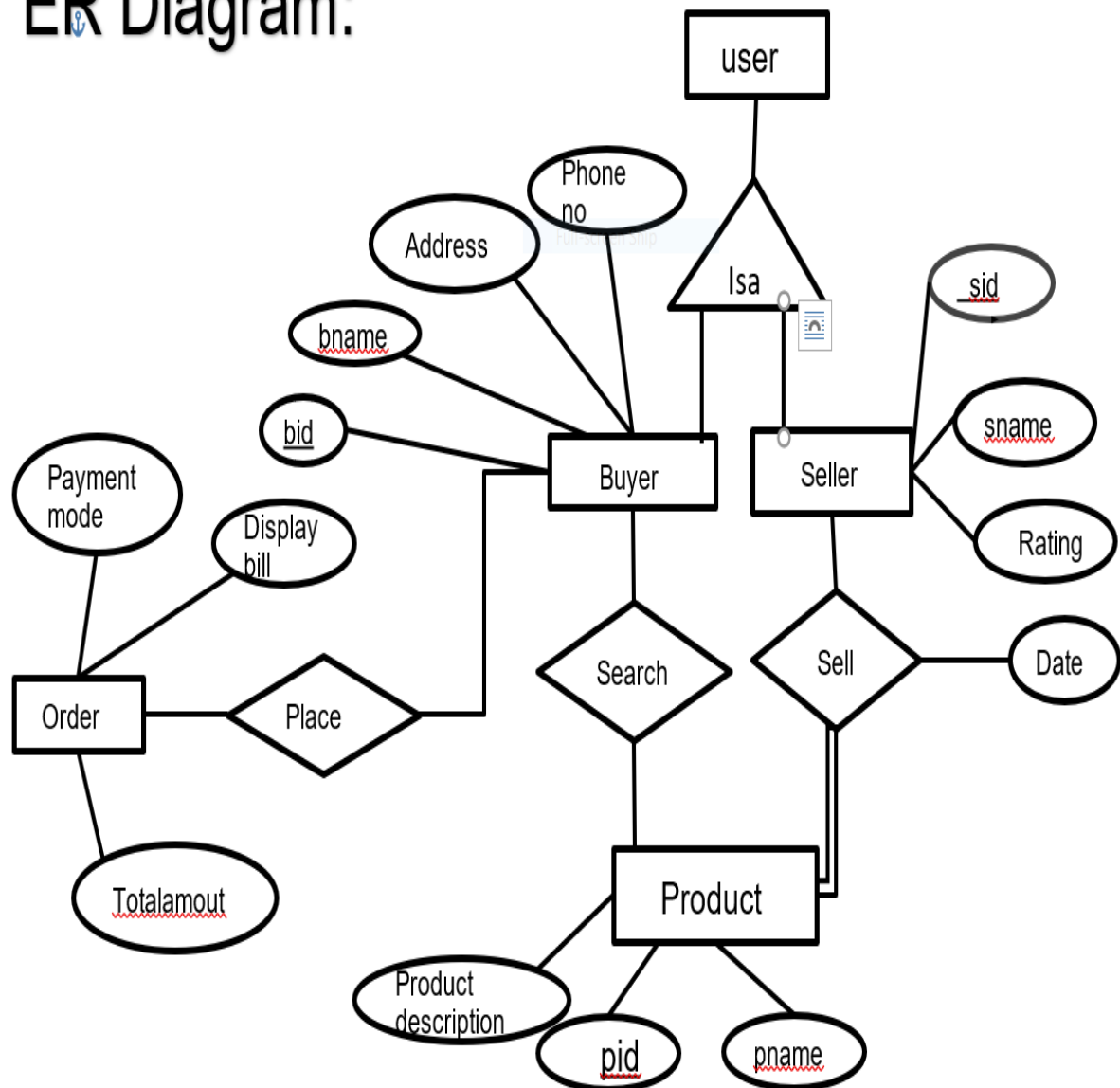
e) Implementation

i) Front end programs and its connectivity.

```
        public void connectToDB()

  {

             try

             {

               connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","kusuma","vasavi");

               statement = connection.createStatement();

               statement.executeUpdate("commit");


             }

             catch (SQLException connectException)

             {

               System.out.println(connectException.getMessage());

               System.out.println(connectException.getSQLState());

               System.out.println(connectException.getErrorCode());

               System.exit(1);

             }

  }
```

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

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

Table Created in SQL for above mentioned purpose is as:

Create table farmer(

bid number(10) primary key, bname varchar(20),phoneno number(11),address varchar2(30));

PROGRAM:

1)AddBuyer

```
package Buyer;

import java.awt.Button;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.Panel;

import java.awt.TextArea;

import java.awt.TextField;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;


public class AddBuyer extends Panel

{


        /**

         *

         */
```

k.kusuma
1602-18-737-312

```java
    private static final long serialVersionUID = 1L;

    Button AddBuyerButton;

    TextField BID,BNAME,ADDRESS,PHONE;

    TextArea errorText;

    Connection connection;

    Statement statement;

    public AddBuyer()

    {

            try

            {

                    Class.forName("oracle.jdbc.driver.OracleDriver");

            }

            catch (Exception e)

            {

                    System.err.println("Unable to find and load driver");

                    System.exit(1);

            }

            connectToDB();


    }


    public void connectToDB()

  {

            try

            {

            connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","kusuma","vasavi");
```

```java
                statement = connection.createStatement();

                statement.executeUpdate("commit");



            }

            catch (SQLException connectException)

            {

             System.out.println(connectException.getMessage());

             System.out.println(connectException.getSQLState());

             System.out.println(connectException.getErrorCode());

             System.exit(1);

            }

    }

        public void buildGUI()

        {

                //Handle Insert Account Button

                AddBuyerButton = new Button("Add Buyer");

                AddBuyerButton.addActionListener(new ActionListener()

                {

                        public void actionPerformed(ActionEvent e)

                        {

                                try

                                {

                                 String query= "INSERT INTO Buyer VALUES(" + BID.getText() + ",
" + BNAME.getText() + ",'" + ADDRESS.getText() + "'," + PHONE.getText() +")";

                                        int i = statement.executeUpdate(query);

                                        statement.executeUpdate("commit");

                                        errorText.append("\nInserted " + i + " rows successfully");
```

```
                    }

                    catch (SQLException insertException)

                    {

                     displaySQLErrors(insertException);

                    }

              }

       });

       BID=new TextField(15);

       BNAME = new TextField(15);

       ADDRESS = new TextField(15);

       PHONE = new TextField(15);

       errorText = new TextArea(10, 40);

       errorText.setEditable(true);


       Panel first = new Panel();

       first.setLayout(new GridLayout(6, 2));

       first.add(new Label("BID:"));

       first.add(BID);

       first.add(new Label("BNAME:"));

       first.add(BNAME);

       first.add(new Label("ADDRESS :"));

       first.add(ADDRESS);

       first.add(new Label("PHONE:"));

       first.add(PHONE);


       first.setBounds(125,80,200,140);
```

k.kusuma
1602-18-737-312

```java
            Panel second = new Panel(new GridLayout(4, 1));

            second.add(AddBuyerButton);

    second.setBounds(125,220,150,100);



            Panel third = new Panel();

            third.add(errorText);

            third.setBounds(125,320,300,200);



            setLayout(null);



            add(first);

            add(second);

            add(third);

            setSize(500, 600);

            setVisible(true);

            System.out.println("hello");

        }




    private void displaySQLErrors(SQLException e)

    {

            errorText.append("\nSQLException: " + e.getMessage() + "\n");

            errorText.append("SQLState:     " + e.getSQLState() + "\n");

            errorText.append("VendorError:  " + e.getErrorCode() + "\n");
```

Title: Amazon like commercial site cart data management platform

```java
        }

}

2)UpdateBuyer

Buyer;

import java.awt.Button;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.Panel;

import java.awt.TextArea;

import java.awt.TextField;

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;


public class UpdateBuyer extends Panel


{
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
/**
 *
 */
private static final long serialVersionUID = 1L;

Button UpdateBuyerButton;

List BuyerIDList;

TextField BID,BNAME,ADDRESS,PHONE;

TextArea errorText;

Connection connection;

Statement statement;

ResultSet rs;


public UpdateBuyer()
{
        try
        {
                Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
                System.err.println("Unable to find and load driver");
                System.exit(1);
        }
        connectToDB();
```

Title: Amazon like commercial site cart data management platform

```java
        }


        public void connectToDB()

    {

                try

                {

                 connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","kusuma","vasavi");

                 statement = connection.createStatement();



                }

                catch (SQLException connectException)

                {

                 System.out.println(connectException.getMessage());

                 System.out.println(connectException.getSQLState());

                 System.out.println(connectException.getErrorCode());

                 System.exit(1);

                }

    }


        public void loadBuyer()

        {

                try

                {

                        BuyerIDList.removeAll();

                rs = statement.executeQuery("SELECT BID FROM Buyer");
```

k.kusuma
1602-18-737-312

```java
                while (rs.next())

                {

                        BuyerIDList.add(rs.getString("BID"));

                }

            }

            catch (SQLException e)

            {

                displaySQLErrors(e);

            }

        }


        public void buildGUI()

        {

            BuyerIDList = new List(10);

            loadBuyer();

            add(BuyerIDList);


            BuyerIDList.addItemListener(new ItemListener()

            {

                    public void itemStateChanged(ItemEvent e)

                    {

                            try

                            {

                                    rs = statement.executeQuery("SELECT * FROM Buyer where BID ="+BuyerIDList.getSelectedItem());

                                    rs.next();
```

```
                              BID.setText(rs.getString("BID"));

                              BNAME.setText(rs.getString("BNAME"));

                              ADDRESS.setText(rs.getString("ADDRESS"));

                              PHONE.setText(rs.getString("PHONE"));


                      }

                      catch (SQLException selectException)

                      {

                              displaySQLErrors(selectException);

                      }

              }

      });



      //Handle Update Sailor Button

      UpdateBuyerButton = new Button("Update Buyer");

      UpdateBuyerButton.addActionListener(new ActionListener()

      {

              public void actionPerformed(ActionEvent e)

              {

                      try

                      {

                              Statement statement = connection.createStatement();

                              int i = statement.executeUpdate("UPDATE Buyer"

                              + " SET BID=" + BID.getText() + ", "
```

```
                            + " BNAME = " + BNAME.getText() + ", "

                            + " ADDRESS = '"+ ADDRESS.getText()

                            + "  WHERE PHONE = " + BuyerIDList.getSelectedItem());

                            errorText.append("\nUpdated " + i + " rows successfully");

                            i = statement.executeUpdate("commit");


                            loadBuyer();

                        }

                    catch (SQLException insertException)

                    {

                            displaySQLErrors(insertException);

                    }

                }

        });

        BID = new TextField(15);

        BNAME = new TextField(15);

        ADDRESS= new TextField(15);

        PHONE = new TextField(15);

        PHONE.setEditable(false);


        errorText = new TextArea(10, 40);

        errorText.setEditable(false);


        Panel first = new Panel();

        first.setLayout(new GridLayout(6, 2));
```

k.kusuma
1602-18-737-312

```java
        first.add(new Label("BID:"));

        first.add(BID);

        first.add(new Label("BNAME:"));

        first.add(BNAME);

        first.add(new Label("ADDRESS:"));

        first.add(ADDRESS);

        first.add(new Label("PHONE"));

        first.add(PHONE);


        Panel second = new Panel(new GridLayout(4, 1));

        second.add(UpdateBuyerButton);


        Panel third = new Panel();

        third.add(errorText);


        add(first);

        add(second);

        add(third);


        setSize(500, 600);

        setLayout(new FlowLayout());

        setVisible(true);


    }
```

k.kusuma
1602-18-737-312

```java
        private void displaySQLErrors(SQLException e)

        {

                errorText.append("\nSQLException: " + e.getMessage() + "\n");

                errorText.append("SQLState:     " + e.getSQLState() + "\n");

                errorText.append("VendorError:  " + e.getErrorCode() + "\n");

        }


}
```

3)DeleteBuyer

package Buyer;

import java.awt.Button;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import java.awt.Label;

import java.awt.List;

import java.awt.Panel;

import java.awt.TextArea;

import java.awt.TextField;

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;

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
import java.sql.SQLException;

import java.sql.Statement;


public class DeleteBuyer extends Panel
{
        /**
         *
         */
        private static final long serialVersionUID = 1L;

        Button deleteBuyerButton;

        List BuyerIDList=null;

        TextField BID,BNAME,ADDRESS,PHONE;

        TextArea errorText;

        Connection connection;

        Statement statement;

        ResultSet rs;


        public DeleteBuyer()
        {
                try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                }
                catch (Exception e)
                {
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
                    System.err.println("Unable to find and load driver");

                    System.exit(1);

            }

            connectToDB();

        }


        public void connectToDB()

    {

            try

            {

             connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","kusuma","vasavi");

             statement = connection.createStatement();


            }

            catch (SQLException connectException)

            {

              System.out.println(connectException.getMessage());

              System.out.println(connectException.getSQLState());

              System.out.println(connectException.getErrorCode());

              System.exit(1);

            }

    }


        public void loadBuyer()

        {
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
        try
        {

            BuyerIDList.removeAll();

            rs = statement.executeQuery("SELECT * FROM Buyer");

            while (rs.next())

            {

                    BuyerIDList.add(rs.getString("BID"));

            }

        }

        catch (SQLException e)

        {

            displaySQLErrors(e);

        }

    }


    public void buildGUI()

    {

        BuyerIDList = new List(10);

            loadBuyer();

            add(BuyerIDList);


            //When a list item is selected populate the text fields

            BuyerIDList.addItemListener(new ItemListener()

            {
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
public void itemStateChanged(ItemEvent e)
{
    try
    {
        rs = statement.executeQuery("SELECT * FROM Buyer");

        while (rs.next())
        {
            if (rs.getString("BID").equals(BuyerIDList.getSelectedItem()))
                break;
        }
        if (!rs.isAfterLast())
        {
            BID.setText(rs.getString("BID"));

            BNAME.setText(rs.getString("BNAME"));

            ADDRESS.setText(rs.getString("ADDRESS"));

            PHONE.setText(rs.getString("PHONE"));
        }
    }
    catch (SQLException selectException)
    {
        displaySQLErrors(selectException);
    }
});
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
                deleteBuyerButton = new Button("Delete Buyer");

                deleteBuyerButton.addActionListener(new ActionListener()

                {

                        public void actionPerformed(ActionEvent e)

                        {

                                try

                                {

                                        Statement statement = connection.createStatement();

                                        int i = statement.executeUpdate("DELETE FROM Buyer WHERE
BID = "

                                                        + BuyerIDList.getSelectedItem());

                                        errorText.append("\nDeleted " + i + " rows successfully");

                                        BID.setText(null);

                                        BNAME.setText(null);

                                        ADDRESS.setText(null);

                                        PHONE.setText(null);

                                        statement.executeUpdate("commit");

                                        loadBuyer();

                                }

                                catch (SQLException insertException)

                                {

                                        displaySQLErrors(insertException);

                                }

                        }

                });
```

k.kusuma
1602-18-737-312

```java
BID = new TextField(15);

BNAME = new TextField(15);

ADDRESS = new TextField(15);

PHONE = new TextField(15);

errorText = new TextArea(10, 40);

errorText.setEditable(false);


Panel first = new Panel();

first.setLayout(new GridLayout(6, 2));

first.add(new Label("BID:"));

first.add(BID);

first.add(new Label("BNAME:"));

first.add(BNAME);

first.add(new Label("ADDRESS:"));

first.add(ADDRESS);

first.add(new Label("PHONE:"));

first.add(PHONE);

Panel second = new Panel(new GridLayout(4, 1));

second.add(deleteBuyerButton);


Panel third = new Panel();

third.add(errorText);


add(first);

add(second);
```

k.kusuma
1602-18-737-312

```java
                add(third);


                setSize(450, 600);

                setLayout(new FlowLayout());

                setVisible(true);


        }



        private void displaySQLErrors(SQLException e)

        {

                errorText.append("\nSQLException: " + e.getMessage() + "\n");

                errorText.append("SQLState:     " + e.getSQLState() + "\n");

                errorText.append("VendorError:  " + e.getErrorCode() + "\n");

        }


}
```

Main program

```java
package main;


import java.awt.*;

import java.awt.event.*;


import javax.swing.*;
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
import javax.swing.JOptionPane;


import Seller.AddSeller;

import Seller.UpdateSeller;

import Seller.DeleteSeller;

import Buyer.DeleteBuyer;

import Buyer.UpdateBuyer;

import Search.*;

import Sell.AddSell;

import Sell.UpdateSell;

import Sell.DeleteSell;

import Buyer.AddBuyer;

import Product.AddProduct;

import Product.UpdateProduct;

import Product.DeleteProduct;

class FrontPage extends JFrame implements ActionListener

{

        private static final long serialVersionUID = 1L;

        String msg = "";

          Label ll;

         CardLayout cardLO;



        //Create Panels for each of the menu items, welcome screen panel and home screen panel
with CardLayout

        AddBuyer add;

        UpdateBuyer upb;
```

k.kusuma
1602-18-737-312

```java
        DeleteBuyer delb;

        AddSeller addr;

        UpdateSeller upr;

        DeleteSeller delr;

        AddSell adds;

        DeleteSell dels;

        UpdateSell ups;

        AddSearch adde;

        UpdateSearch upe;

        DeleteSearch dele;

        AddProduct addp;

        UpdateProduct upp;

        DeleteProduct delp;

        Panel home,welcome;


            public FrontPage()

        {

                cardLO = new CardLayout();



                //Create an empty home panel and set its layout to card layout

                home = new Panel();

                home.setLayout(cardLO);



                Label ll = new Label();
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
ll.setAlignment(Label.CENTER);

ll.setText("Welcome to web site System database");


//Create welcome panel and add the label to it

welcome = new Panel();

welcome.add(ll);

//create panels for each of our menu items and build them with
respective components

add=new AddBuyer();add.buildGUI();

upb = new UpdateBuyer();  upb.buildGUI();

delb = new DeleteBuyer();        delb.buildGUI();

addr= new AddSeller();addr.buildGUI();

upr = new UpdateSeller();upr.buildGUI();

delr=new DeleteSeller();delr.buildGUI();

adds=new AddSell();adds.buildGUI();

dels=new DeleteSell();dels.buildGUI();

ups=new UpdateSell();ups.buildGUI();

adde=new AddSearch();adde.buildGUI();

upe=new UpdateSearch();upe.buildGUI();

dele=new DeleteSearch();dele.buildGUI();

addp=new AddProduct();adde.buildGUI();

upp=new UpdateProduct();upe.buildGUI();

delp=new DeleteProduct();dele.buildGUI();



home.add(welcome, "Welcome");
```

k.kusuma
1602-18-737-312

```java
home.add(add, "Add buyer");

home.add(upb, "Update buyer");

home.add(delb, "Delete buyer");

home.add(addr, "Add seller");

home.add(upr,"Update seller");

home.add(delr,"Delete seller");

home.add(adds,"Add sell");

home.add(dels,"Delete sell");

home.add(ups,"Update sell");

home.add(adde,"Add search");

home.add(upe,"Update search");

home.add(dele,"Delete search");

home.add(addp,"Add Product");

home.add(upp,"Update Product");

home.add(delp,"Delete Product");




// add home panel to main frame

add(home);



// create menu bar and add it to frame

MenuBar mbar = new MenuBar();

setMenuBar(mbar);



// create the menu items and add it to Menu
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
Menu Buyer = new Menu("Buyer ");

MenuItem item1, item2,item3;

Buyer.add(item1 = new MenuItem("Add Buyer"));

Buyer.add(item2 = new MenuItem("View Buyer"));

Buyer.add(item3 = new MenuItem("Delete Buyer"));

mbar.add(Buyer);


Menu Seller = new Menu("Seller");

MenuItem item4, item5, item6;

Seller.add(item4 = new MenuItem("Add Seller"));

Seller.add(item5 = new MenuItem("View seller"));

Seller.add(item6 = new MenuItem("Delete seller"));

mbar.add(Seller);


Menu Sell = new Menu("Sell");

MenuItem item7, item8,item9;

Sell.add(item7 = new MenuItem("Add sell"));

Sell.add(item8 = new MenuItem("View sell"));

Sell.add(item9 = new MenuItem("Delete sell"));

mbar.add(Sell);


Menu product = new Menu("product");

MenuItem item10, item11, item12;

product.add(item10 = new MenuItem("Add product"));

product.add(item11 = new MenuItem("View product"));
```

k.kusuma
1602-18-737-312

```
product.add(item12 = new MenuItem("Delete product"));

mbar.add(product);


Menu Search = new Menu("Search");

MenuItem item13, item14, item15;

Search.add(item13 = new MenuItem("Add search"));

Search.add(item14 = new MenuItem("View search"));

Search.add(item15 = new MenuItem("Delete search"));

mbar.add(Search);



// register listeners

item1.addActionListener(this);

item2.addActionListener(this);

item3.addActionListener(this);

item4.addActionListener(this);

item5.addActionListener(this);

item6.addActionListener(this);

item7.addActionListener(this);

item8.addActionListener(this);

item9.addActionListener(this);

item10.addActionListener(this);

item11.addActionListener(this);

item12.addActionListener(this);

item13.addActionListener(this);
```

k.kusuma
1602-18-737-312

```java
                item14.addActionListener(this);

                item15.addActionListener(this);


                 // Anonymous inner class which extends WindowAdaptor to handle the
Window event: windowClosing

                addWindowListener(new WindowAdapter(){

                        public void windowClosing(WindowEvent we)

                        {

                                quitApp();

                        }

                });


                //Frame properties

                setTitle("web site System");

                setSize(500, 600);

                setVisible(true);


        }


        public void actionPerformed(ActionEvent ae)

        {

                String arg = ae.getActionCommand();

                if(arg.equals("Add buyer"))

                {

                cardLO.show(home, "Add buyer");

    }
```

k.kusuma
1602-18-737-312

```java
            else if(arg.equals("View buyer"))

            {

                    cardLO.show(home, "Update buyer ");

                    upb.loadBuyer();

            }


            else if(arg.equals("Delete buyer"))

            {

                    cardLO.show(home, "Delete buyer");

                    delb.loadBuyer();

            }
    else if(arg.equals("Add seller"))

            {

                    cardLO.show(home, "Add seller");

            }

            else if(arg.equals("View seller"))

            {

                    cardLO.show(home, "Update seller");

                    upr.loadSeller();

            }

            else if(arg.equals("Delete seller"))

            {

            cardLO.show(home, "Delete seller");

                    delr.loadSeller();
```

k.kusuma
1602-18-737-312

```
			}

		else if(arg.equals("Add sell"))

			{

				cardLO.show(home, "Add sell");

			}

			else if(arg.equals("Update sell"))

			{

				cardLO.show(home, "Update sell");

				ups.loadSell();

			}

		else if(arg.equals("Delete sell"))

		{


				cardLO.show(home, "Delete sell");

				dels.loadSell();

			}

		else if(arg.equals("Add search"))

			{

				cardLO.show(home, "Add search");

			}

			else if(arg.equals("View search"))

			{

				cardLO.show(home, "Update search");

				upe.loadSearch();

			}
```

k.kusuma
1602-18-737-312

```java
            else if(arg.equals("Delete search"))

            {

                    cardLO.show(home, "Delete search");

                    dele.loadSearch();



            }

             else if(arg.equals("Add Product"))

            {

                    cardLO.show(home, "Add Product");

            }

            else if(arg.equals("View Product"))

            {

                    cardLO.show(home, "Update Product");

                    upp.loadProduct();

            }

            else if(arg.equals("Delete Product"))

            {

                    cardLO.show(home, "Delete Product");

                    delp.loadProduct();



            }



    }
        private void quitApp () {
```

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```java
                try {

                        //Show a Confirmation Dialog.

                        int reply = JOptionPane.showConfirmDialog (this,

                                "Are you really want to exit\nFrom web site System?",

                                "web site SYSTEm - Exit",
JOptionPane.YES_NO_OPTION, JOptionPane.PLAIN_MESSAGE);

                        //Check the User Selection.

                        if (reply == JOptionPane.YES_OPTION) {

                                setVisible (false);        //Hide the Frame.

                                dispose();                //Free the System Resources.

                                System.out.println ("Thanks for Using web site  System\nAuthor
- kusuma");

                                System.exit (0);      //Close the Application.

                        }
                        else if (reply == JOptionPane.NO_OPTION) {

                                setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);

                        }
                }


                catch (Exception e) {}



        }
        public static void main(String ... args)

        {

                new FrontPage();
```

k.kusuma
1602-18-737-312

       }

       }

ii) GitHub Link and folder structure:

    a) Link- https://github.com/kusumakoduru/amazon-like-commercial-site
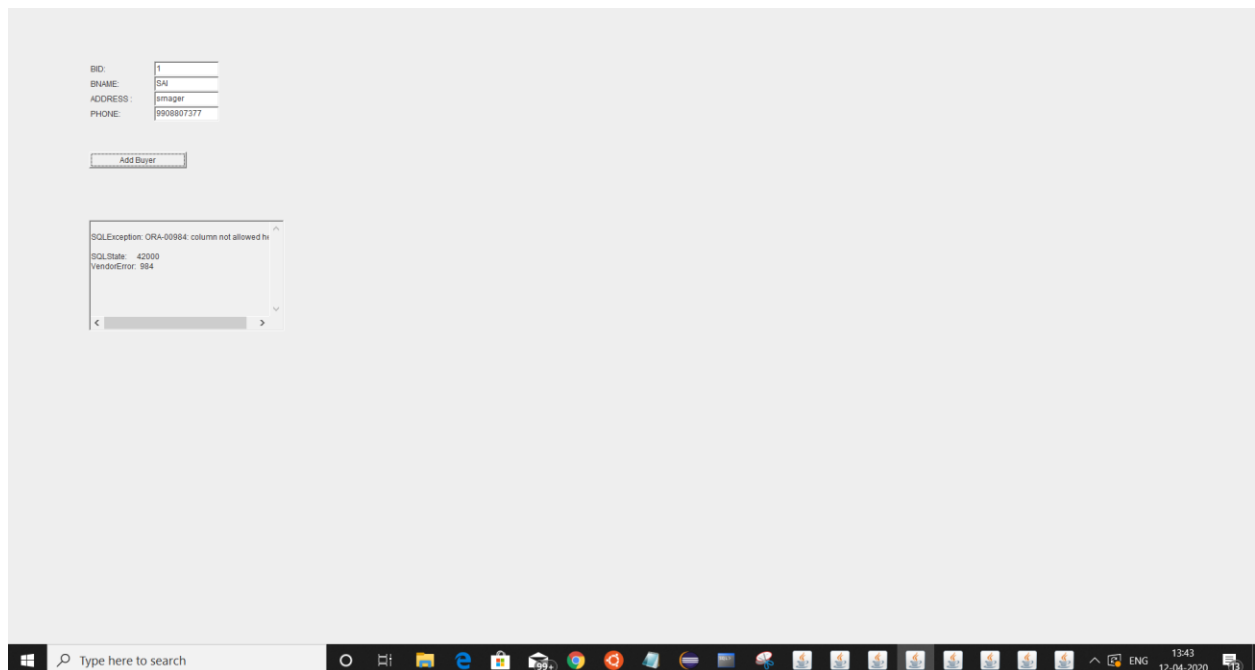   b) Folder Structure:
            1.Data Base Design
            2.Mini Project Report
            3.Front End Programs (JAVA CODE)

Testing:

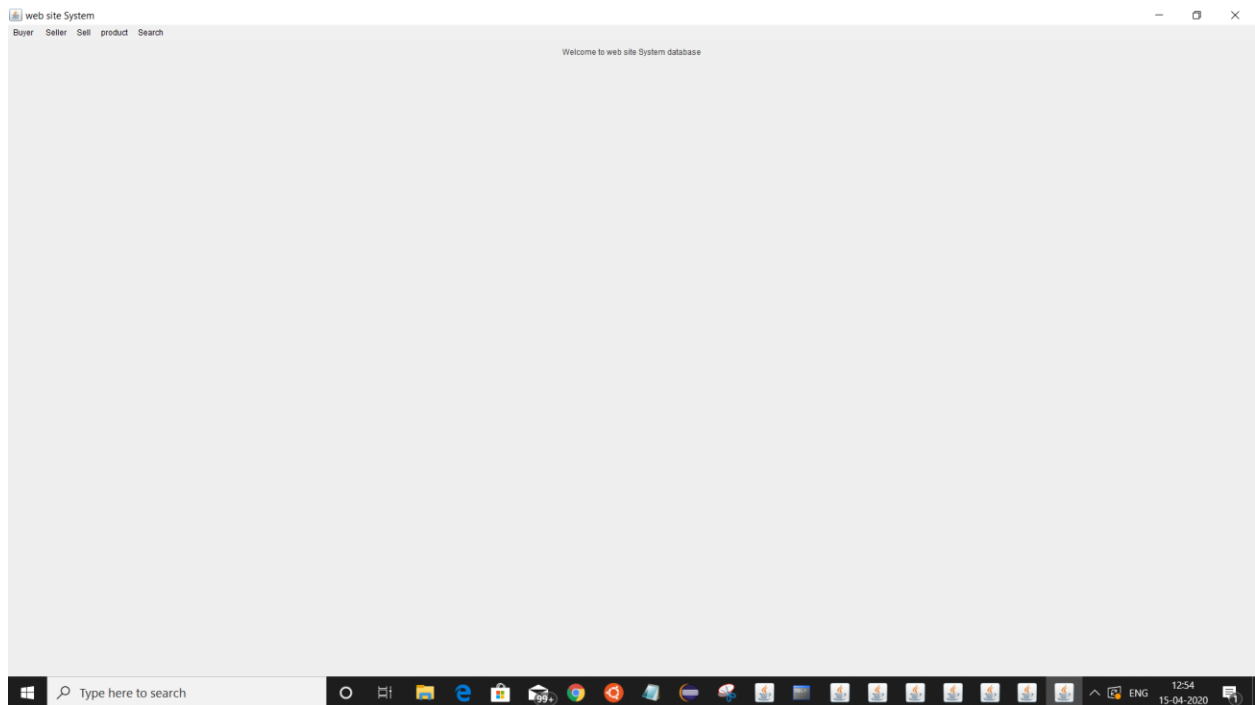If incorrect values are entered which mismatch data types it won't allow to insert.

RESULT:

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

The process of entering information into the frame created by java code so that the data is reflected in the database using **JDBC connectivity** is done successfully.

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

Buyer   Seller   Sell   product   Search

BID:
1
PID:
3

Add Buyer

Inserted 1 rows successfully

Type here to search

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

```
SQL> select * from search;

PID                            BID
------------------------------ --------------------
1                              1
1                              3
5                              3
10                             1
1                              2

SQL>
```

web site System

Buyer   Seller   Sell   product   Search

```
1
3
3
1
2
```

BID:
```
3
```
Delete Buyer

PID:
```
5
```

Deleted 1 rows successfully
Deleted 1 rows successfully

k.kusuma
1602-18-737-312

Title: Amazon like commercial site cart data management platform

web site System

Buyer Seller Sell product Search

```
1
3
3
1
2
3
```

BID:
1   Update Search

PID:
3

SQLException: ORA-01747: invalid user.table.colu

SQLState: 42000
VendorError: 1747

---

web site System

Buyer Seller Sell product Search

Welcome to web site System database

web site SYSTEm - Exit   X

**Are you really want to exit
From web site System?**

Yes   No

---

k.kusuma
1602-18-737-312
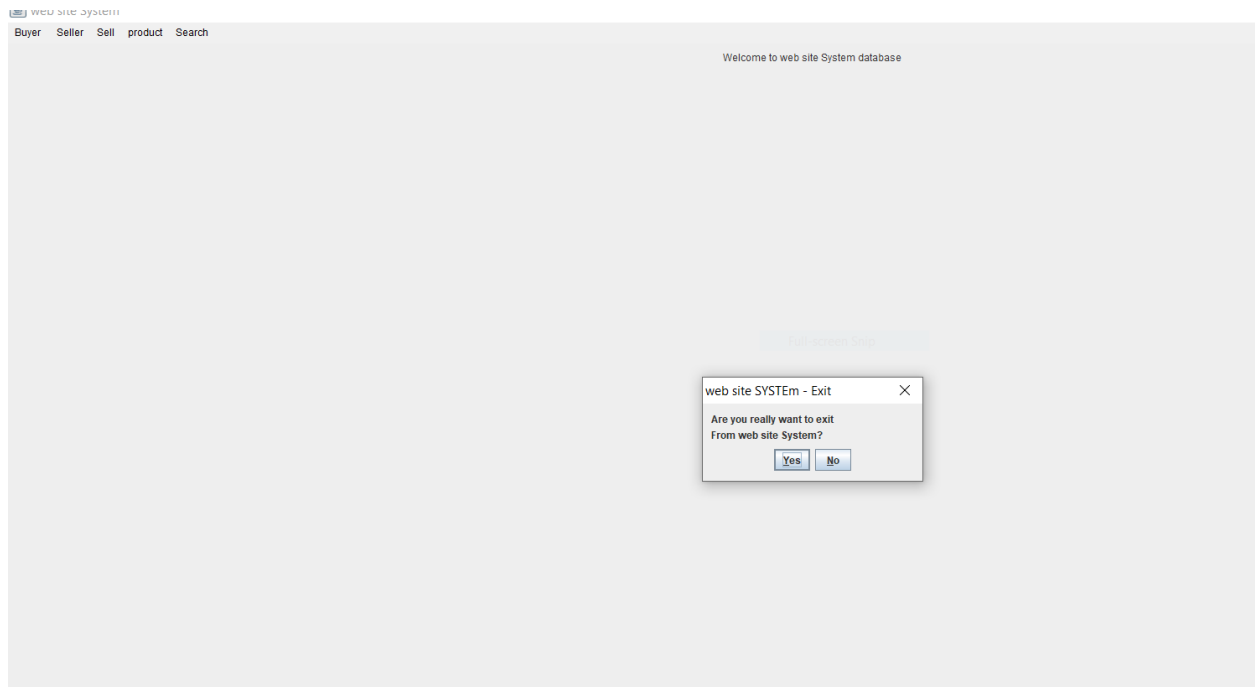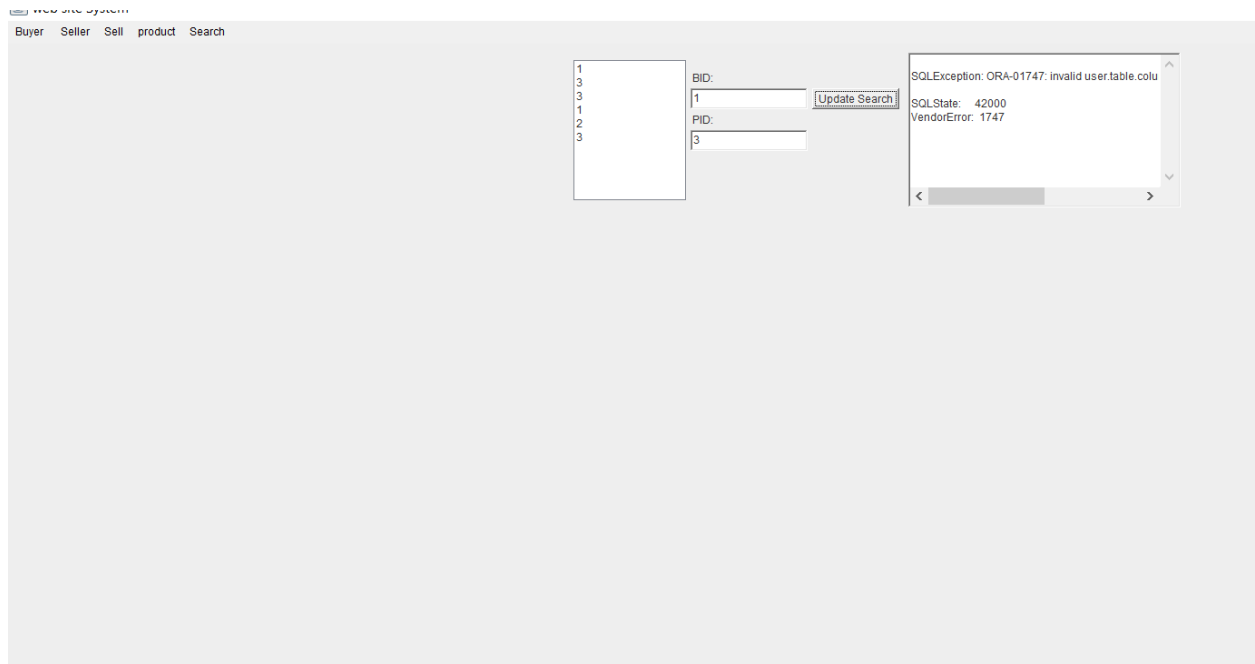
DISCUSSION & FUTURE WORK :

The application done till now is basically to store the details of commercial site which means through this site any one can place the order and few people can sell there product through this site buyer can search for different product and they can place order they can buy the products for this number of entities and relationship between them will come into picture, which can be converted into first tables using SQL commands and then into GUI program using java code.

REFERENCES:

https://docs.oracle.com/javase/8/docs/api/

https://www.geeksforgeeks.org/establishing-jdbc-connection-in-java/