LALAJI MEMORIAL
OMEGA INTERNATIONAL SCHOOL

BONA-FIDE CERTIFICATE

SUBJECT: INFORMATICS PRACTICES

REGISTRATION NO.:20606804

Certified to be the *bona-fide* record of work done by

Muhil Raghavandhar A

of **Class XII** as **PROJECT** in the topic **Online Shopping Management** at **Lalaji Memorial Omega International School, Chennai 600128** during the academic year **2019-2020**. Submitted for **AISSCE Practical Examination** held on 23-01-2020.

LALAJI MEMORIAL OMEGA INTERNATIONAL SCHOOL
KOLAPAKKAM, CHENNAI - 600 128
Seal

PRINCIPAL
LALAJI MEMORIAL
OMEGA INTERNATIONAL SCHOOL
KOLAPAKKAM, CHENNAI - 600 128
Principal

Date: 23.01.2020

Internal Examiner

TN2605015

External Examiner

ACKNOWLEDGEMENT

The satisfaction and euphoria of the successful completion of any task would be incomplete without the mention of the people who made it possible. The constant guidance of these people and encouragement provided by them crowned my efforts with success and glory. I consider it as a privilege to express my gratitude to all those who led and guided me in the course of this project.

First and foremost I would like to express my gratitude to the **MANAGEMENT** of my school for making it possible for me to be a part of this project.

I would like to convey my sincere regards to **Mrs.G.MAHALAKSHMI**, my internal guide for her valuable suggestion and necessary guidance during the course of the project.

I also take esteem privilege to thank **Ms.JAIPRIYA**, who has been the very backbone and a catalyst for the success behind the completion of this project.

Last but not the least I thank my **PARENTS AND FRIENDS** who had been constant source of inspiration for the completion of this project.

Above all, I thank the **Almighty and My Master** for the blessings to complete this project successfully.

TABLE OF CONTENTS

1. Introduction

1.1 Front End – Java

Java is a programming language and a computing platform for application development. It was first released by Sun Microsystem in 1995 and later acquired by Oracle Corporation. It is one of the most used programming languages.

Java platform is a collection of programs that help to develop and run programs written in the Java programming language. Java platform includes an execution engine, a compiler, and a set of libraries. JAVA is platform-independent language. It is not specific to any processor or operating system.

Java is a widely used programming language expressly designed for use in the distributed environment of the internet. It is the most popular programming language for Android smartphone applications and is also among the most favored for the development of edge devices and the internet of things.

Java was designed to have the look and feel of the C++ programming language, but is simpler to use and enforces an object-oriented programming model. Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build a small application module or applet for use as part of a webpage.

Java is object oriented. An object is made up of data as fields or attributes and code as procedures or methods. An object can be a part of a class of objects to inherit code common to the class. Objects can be thought of as "nouns" that a user can relate to "verbs." A method is the object's capabilities or behaviors. Because Java's design was influenced by C++, Java was mainly built as an object-orientated language. Java also uses an automatic garbage collector to manage object lifecycles. A programmer will create objects, but the automatic garbage collector will recover memory once the object is no longer in use. However, memory leaks may occur when an object which is no longer being used is stored in a container.

Unlike C++, Java does not use pointers, which can be unsecured. Data converted to bytecode by Java is also not readable to humans. Additionally, Java will run programs inside a sandbox to prevent changes from unknown sources.

1.2 Back End - MySQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all

platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.

MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into seperate applications.

MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer.

MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.

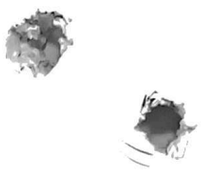
MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability. MySQL users aren't required to learn new commands; they can access their data using standard SQL commands.

MySQL is written in C and C++ and accessible and available across over 20 platforms, including Mac, Windows, Linux and Unix. The RDBMS supports large databases with millions records and supports many data types including signed or unsigned integers 1, 2, 3, 4, and 8 bytes long; FLOAT; DOUBLE; CHAR; VARCHAR; BINARY; VARBINARY; TEXT; BLOB; DATE; TIME; DATETIME; TIMESTAMP; YEAR; SET; ENUM; and OpenGIS spatial types. Fixed- and variable-length string types are also supported.

For security, MySQL uses an access privilege and encrypted password system that enables host-based verification. MySQL clients can connect to MySQL Server using several protocols, including TCP/IP sockets on any platform. MySQL also supports a number of client and utility programs, command-line programs and administration tools such as MySQL Workbench.

1.3 Organization of the Report

Organization of the report is the short summary of the forth coming chapters. Chapter 1 deals with the introduction, front end - Java, back end - MySQL and the organization of the report. Chapter 2 deals with the System analysis, proposed system and enhanced system. Chapter 3 describes the hardware and software requirements of the system. Chapter 4 describes the problem definition, overview of the project, flow chart and module wise description. Chapter 5 describes the conclusion of the project.



2. SYSTEM ANALYSIS

2.1 PROPOSED SYSTEM

The following program was developed to smoothen the process of online shopping management system. Lots of people use Online shopping these days which makes it difficult for the owners to manage a huge customer base. Hence the program we have designed helps the managers to manage all their customers and their details and their items and their details.

Advantages

1. Modify customer details easily.
2. Alter item details with ease.

2.2 Further Scope:

This project is only a basic version of what can be done with a online shopping management system. If a good online shopping management system is built with sufficient time and other requirements, there is immense potential for it. Some of the suggested improvements to this are:

- Update stock of every product
- Maintain a record of all the transactions

These and many other features could be added to create a very powerful library management system.

2. SYSTEM ANALYSIS

2.1 PROPOSED SYSTEM

The following program was developed to smoothen the process of online shopping management system. Lots of people use Online shopping these days which makes it difficult for the owners to manage a huge customer base. Hence the program we have designed helps the managers to manage all their customers and their details and their items and their details.

Advantages

1. Modify customer details easily.
2. Alter item details with ease.

2.2 Further Scope:

This project is only a basic version of what can be done with a online shopping management system. If a good online shopping management system is built with sufficient time and other requirements, there is immense potential for it. Some of the suggested improvements to this are:

- Update stock of every product
- Maintain a record of all the transactions

These and many other features could be added to create a very powerful library management system.

3. SYSTEM SPECIFICATION

3.1 SOFTWARE REQUIREMENTS

- Operating system : Windows
- Front End : Java
- Back End : MySQL
- Tool : NetBeans IDE 7.4 and MySQL Server 5.0

3.2 HARDWARE REQUIREMENTS

- System : Intel Core 2 Duo 2.20 GHz.
- Hard Disk : 320 GB.
- Monitor : 15 VGA Colour.
- Ram : 2 Gb.

4. PROJECT DESCRIPTION

4.1 PROBLEM DEFINITION

A shopping website consists of many products and customers. When the details of these customers and products are made easily accessible to the manager their stress is reduced and makes it easier for them to run the store.

As software applications usage is increasing in our daily life developing applications like this will be an easy task and saves a lot of time for online stores to manage their customers and products through computerized systems. In present scenario details of the customers and items are maintained in manual records which is not an efficient method.

Much of the functioning of the online stores can be automated using the available technologies. Instead of dealing with files, the use of a good database together with a proper front end can reduce operating cost. Technology allows for efficient, accurate and organized storage for retrieval of required data. It can also make the task of a manager easier.

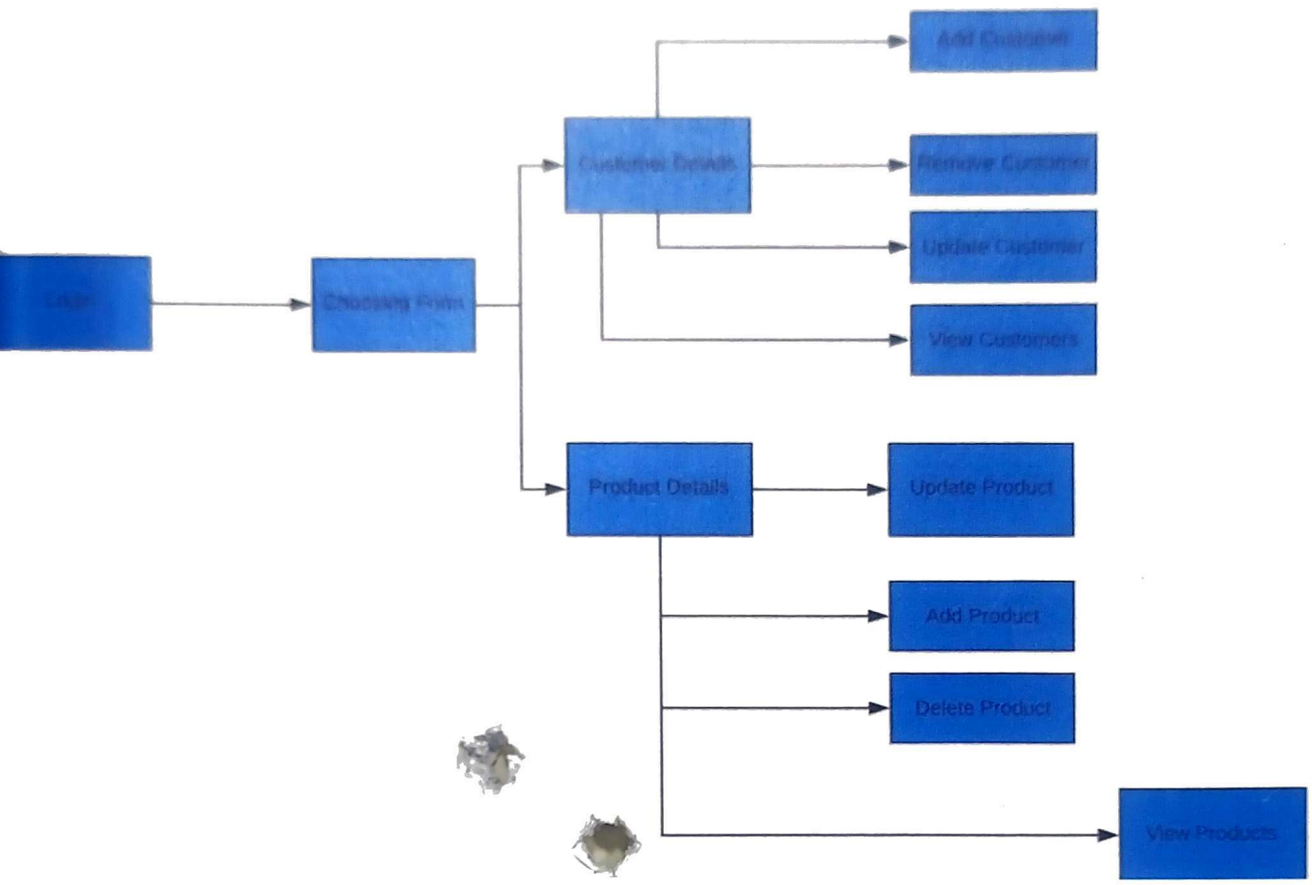
The aim of this project is to develop a software which would automate the functioning of an Online Store. The main objective of the Online Shopping management system project is to make the management more efficient.

4.2 OVERVIEW OF THE PROJECT

This project is designed to create software that can be used in an Online Store. The software has three main uses:

1. Management of Customers.
2. Management of Products,

3.3 FLOW CHART

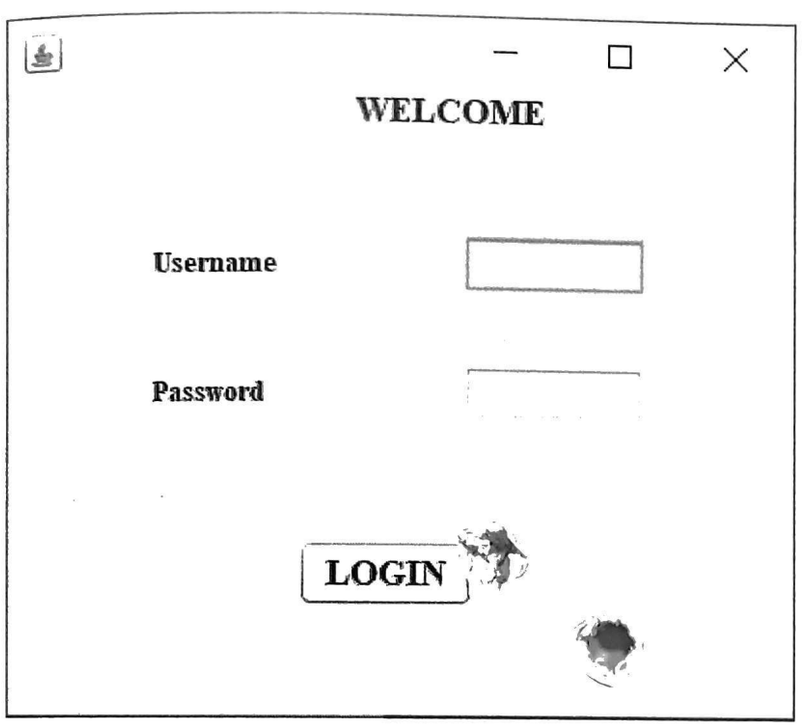


4.4 MODULE DESCRIPTION

4.4.1 Log In

This form accepts a username and password from the user. It then checks if the user is the administrator (the librarian) and if yes, it checks the password and opens the choosing form.

Design:



Coding:

```
try

{

    Class.forName("java.sql.DriverManager");

    Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://local
host:3306/PROJECT","root","aishwarya");

    Statement stat=(Statement)con.createStatement();

    String user=TxtUser.getText();

    String check="";

    String pass=PwF.getText();
```

```
    if(user.equalsIgnoreCase("Admin"))
{
        if(pass.equals("omega"))
        {
            ChoosingForm choose=new ChoosingForm();
            choose.setVisible(true);
            this.setVisible(false);
        }
        else
            JOptionPane.showMessageDialog(this,"Incorrect
Username or Password");
    }
    else
    {
        JOptionPane.showMessageDialog(this,"Try Again");
    }
}
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(this,e.getMessage());
        }
}
```

4.4.2 Choosing Form

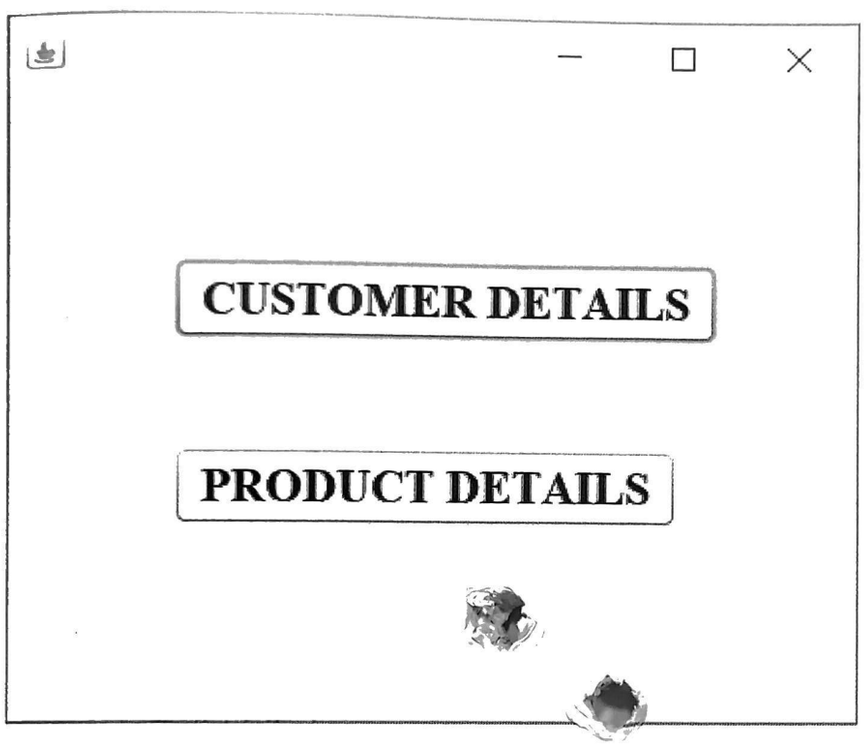
This form opens after logging in.It contains of two buttons:-

i. Customer Details

ii. Product Details

Depending on the button clicked, it opens the appropriate form.

Design:



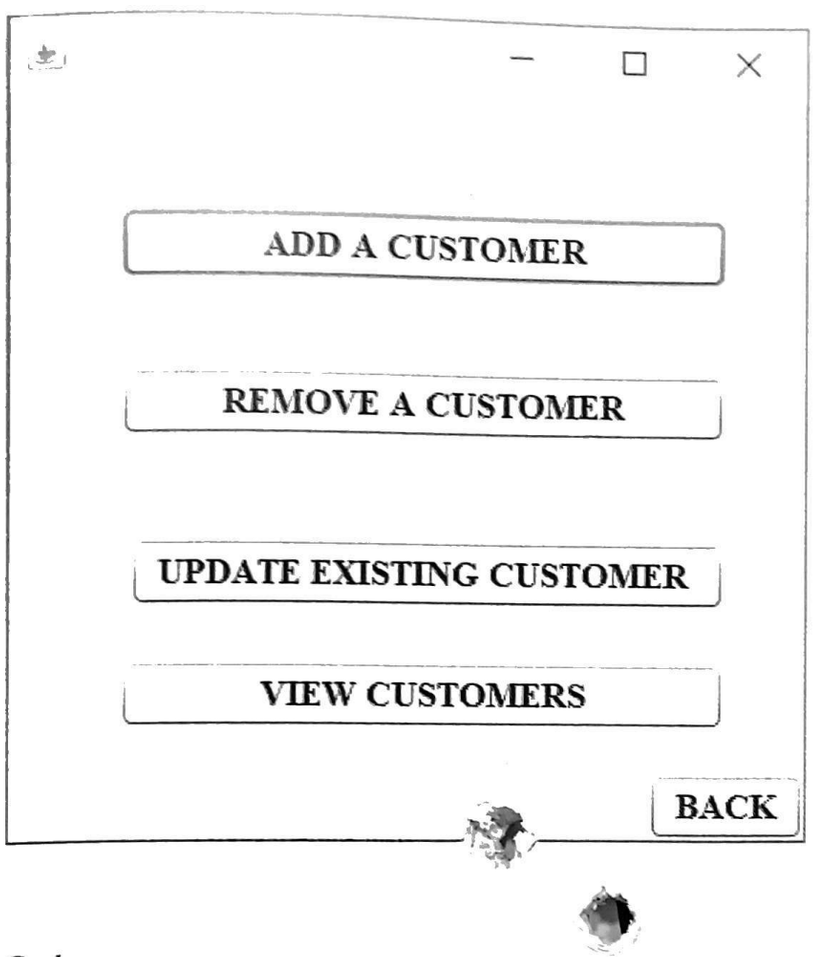
Code:

```
{CustomerDetails cust=new CustomerDetails();
        cust.setVisible(true);
         this.dispose();}
{ ProductDetails prod=new ProductDetails();
        prod.setVisible(true);
        this.dispose();}
```

4.4.3 Customer Details Form:-

This form allows the manager to select the desired process i.e to update or add or delete or view all the customers and takes the manager to the desired form.

Design:



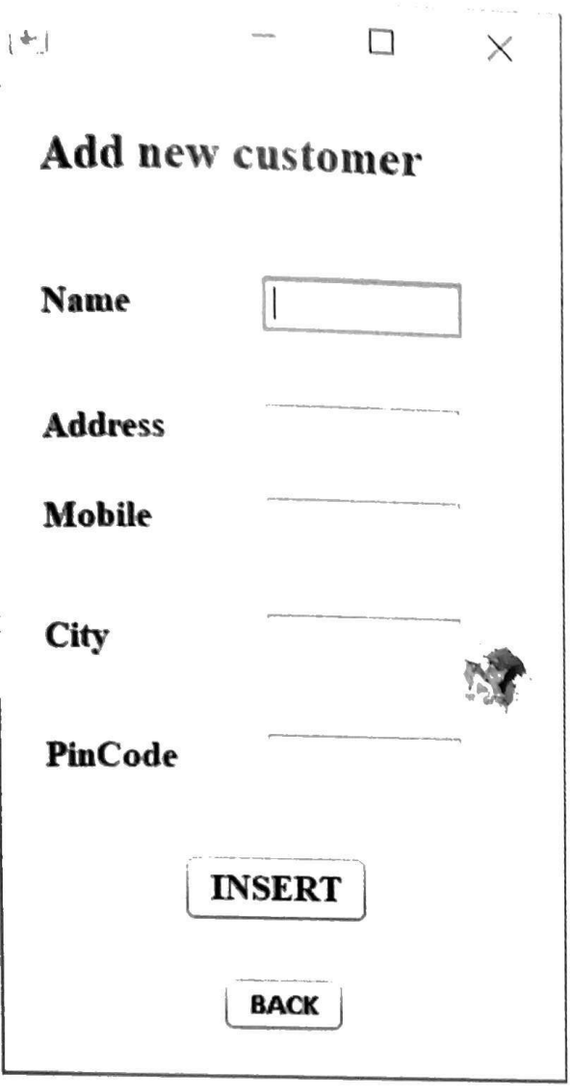
Code:

```
{DeleteCust del=new DeleteCust();
        del.setVisible(true);
        this.dispose();}
{AddCust add=new AddCust();
        add.setVisible(true);
        this.dispose();}
{
View vw=new View();
        vw.setVisible(true);
        this.dispose();}
{ UpdateCust up=new UpdateCust();
        up.setVisible(true);
        this.dispose();}
{ChoosingForm choose=new ChoosingForm();
        choose.setVisible(true);
        this.dispose();}
```

4.4 Add Customer

This form allows the manager to add a customer into the database.

Design:



Code:

To add a new member:

```
String Name=TxtName.getText();
        String Mobile=TxtMob.getText();
        String Address=TxtAdd.getText();
    String PinCode=TxtPC.getText();
        String City=TxtCity.getText();
```

```
        try
        {
            Class.forName("java.sql.DriverManager");
            Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://local
host:3306/project","root","root");
            Statement stat=(Statement)con.createStatement();
            String query="INSERT INTO CUSTOMER
VALUES('"+Name+"','"+Address+"','"+Mobile+"','"+City+"','"+PinC
ode+"');";
            stat.executeUpdate(query);
            JOptionPane.showMessageDialog(this,"Customer
Added");
        }
        catch(Exception e)
        {

JOptionPane.showMessageDialog(this,e.getMessage());}
      catch (Exception e)
      {
          JOptionPane.showMessageDialog(this,e.getMessage());

}
```

4.4.5 Delete Customer

Design:

Code:

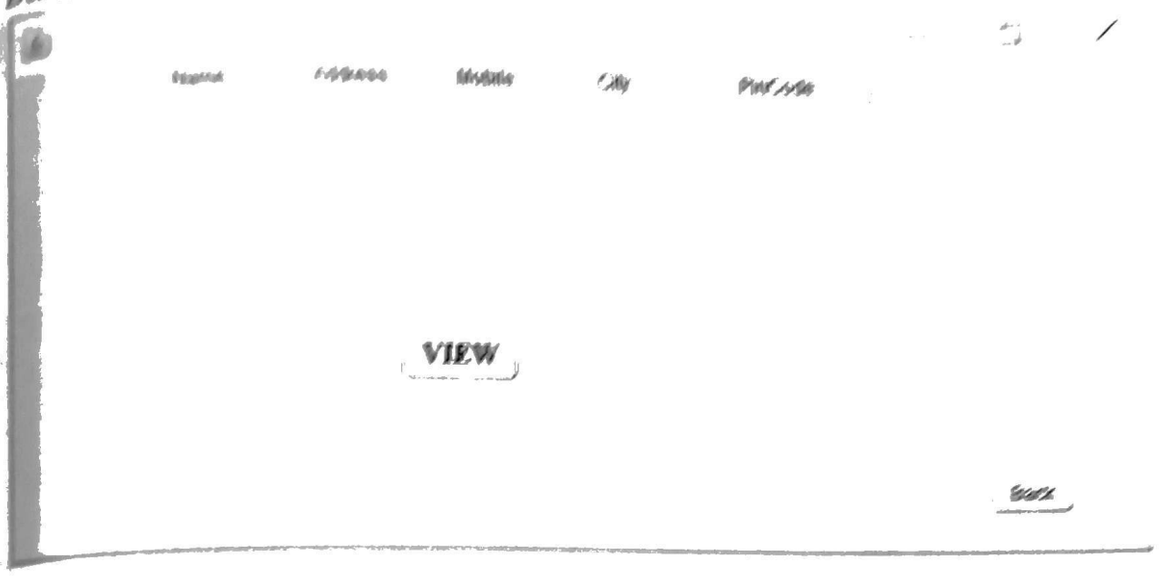
```
String Mobile=TxtID.getText();
        if(Mobile.isEmpty())
        {
            JOptionPane.showMessageDialog(this,"Enter ID of the
customer");

        }
        else
     {try
        {
```

```
            Class.forName("java.sql.DriverManager");
   Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://local
host:3306/project","root","root");
            Statement stat=(Statement)con.createStatement();
            String message="Are you sure to delete?";
            String title="Confirm Deletion";
            int
reply=JOptionPane.showConfirmDialog(null,message,title,JOptionP
ane.YES_NO_OPTION);
            if(reply==JOptionPane.YES_OPTION)
            {
                String query="DELETE FROM CUSTOMER WHERE
Mobile='"+Mobile+"'";
                stat.executeUpdate(query);
                JOptionPane.showMessageDialog(this,"Record
Deleted");
            }
            else
                JOptionPane.showMessageDialog(this,"Record not
deleted");
        }

        catch(Exception e)
        {

JOptionPane.showMessageDialog(this,e.getMessage());}
```

4.4.6 View Customers

This form allows the manager to see all the customers that have registered with the store.

Design



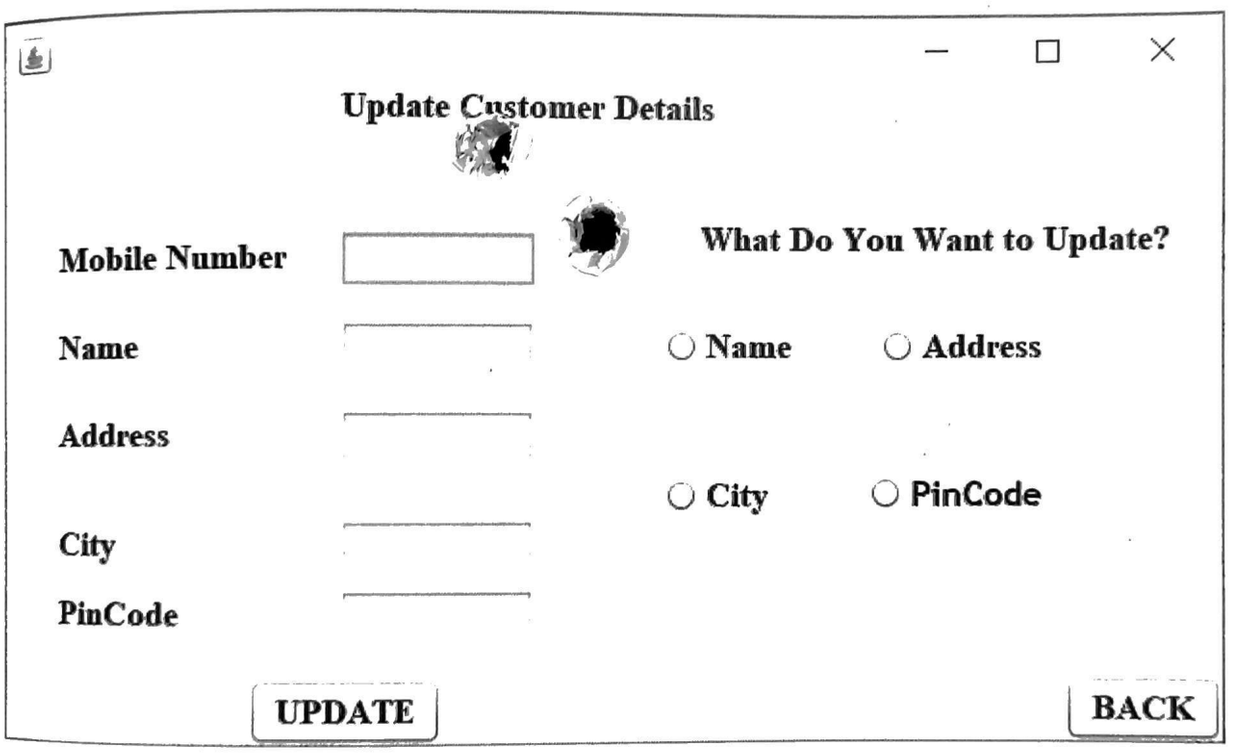
Code:

```
try{
Class.forName("java.sql.DriverManager");
 Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://local
host:3306/project","root","root");
Statement stat=(Statement)con.createStatement();
DefaultTableModel model=(DefaultTableModel)TblCust.getModel();
int rows=model.getRowCount();
while(rows>0)
            {
                model.removeRow(0);
                rows--;}

            String query="SELECT * from customer";
            ResultSet rs=(ResultSet)stat.executeQuery(query);
            while(rs.next())
            {
                String Name=rs.getString("Name");
                String Address=rs.getString("Address");
                String Mobile=rs.getString("Mobile");
                String City=rs.getString("City");
```

```
                String PinCode=rs.getString("PinCode");
                model.addRow(new
Object[]{Name,Address,Mobile,City,PinCode});
            }
            rs.close();
            stat.close();
            con.close();
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(this,e.getMessage());
        }
```

4.4.7 Update Customer

Design :



Code:

```
String Mobile=TxtMob.getText();

        if(Mobile.isEmpty())

        {
```

```
            JOptionPane.showMessageDialog(this,"Enter mobile
number of the customer");
        }
        else
        {
            try
            {
                Class.forName("java.sql.DriverManager");
                Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://local
host:3306/project","root","root");
                Statement
stat=(Statement)con.createStatement();
                String query="SELECT Name,Address,City,PinCode
from customer WHERE Mo[redacted]e='"+Mobile+"'";
                ResultSet r[redacted]tat.executeQuery(query);
                if(rs.next())
                { String Name=rs.getString("Name");
                    String Address=rs.getString("Address");
                    String City=rs.getString("City");
                    String PinCode=rs.getString("PinCode");
                    TxtAdd.setText(Address);
                    TxtCity.setText(City);
                    TxtPC.setText(PinCode);
                    TxtName.setText(Name);
                    if(jRadioButton1.isSelected())
                    {
```

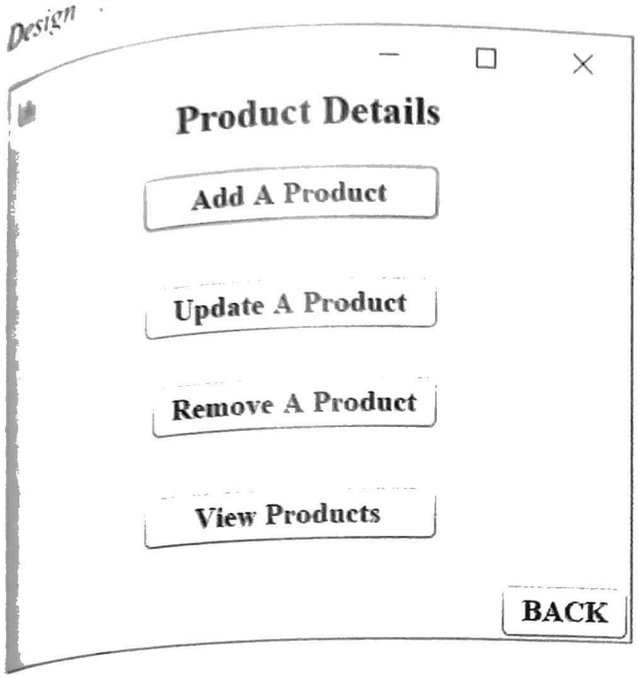
```
                    String
NewName=JOptionPane.showInputDialog(this,"Enter New Name:");
                    String query1="UPDATE CUSTOMER SET
NAME='"+NewName+"' WHERE Mobile='"+Mobile+"';";
                    stat.executeUpdate(query1);

JOptionPane.showMessageDialog(this,"Name updated succesfully");

                }
                else if(jRadioButton2.isSelected())
                {
                    String
NewAdd=JOptionPane.showInputDialog(this,"Enter New Address:");
                    String query1="UPDATE CUSTOMER SET
Address='"+NewAdd+"' WHERE Mobile='"+Mobile+"';";
                    stat.executeUpdate(query1);

JOptionPane.showMessageDialog(this,"Address updated
succesfully");
                }
                else if(jRadioButton3.isSelected())
                {
                    String
NewCity=JOptionPane.showInputDialog(this,"Enter New City:");
                    String query1="UPDATE CUSTOMER SET
City='"+NewCity+"' WHERE Mobile='"+Mobile+"';";
                    stat.executeUpdate(query1);

JOptionPane.showMessageDialog(this,"City updated succesfully");
```

```
                    }
                    else if(jRadioButton4.isSelected())
                    {
                        String
NewPc=JOptionPane.showInputDialog(this,"Enter New PinCode:");
                    String query1="UPDATE CUSTOMER SET
NAME='"+NewPc+"' WHERE Mobile='"+Mobile+"';";
                    stat.executeUpdate(query1);

JOptionPane.showMessageDialog(this,"PinCode updated
succesfully");
                    }

                }
                else
                    JOptionPane.showMessageDialog(this,"No Such
Mobile Number Found");
            }
            catch(Exception e)
            {

JOptionPane.showMessageDialog(this,e.getMessage());
            }
        }
```

4.4.8 Product Details Form

This form allows you to select the various processes that can be done with the product details and takes you to the desired form.

Design :



:

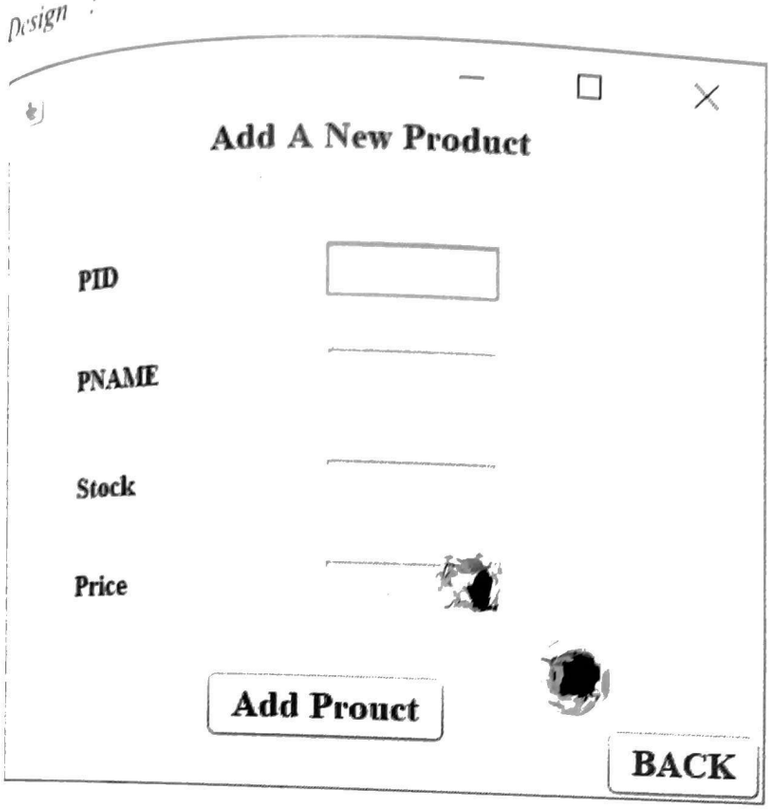
```
iewProductsForm view= ViewProductsForm();
      view.setVisible(true);
      this.dispose();}
ChoosingForm choose=new ChoosingForm();
      choose.setVisible(true);
      this.dispose();}
 dProduct add=new AddProduct();
      add.setVisible(true);
      this.dispose();}
DeleteProd del=new DeleteProd();
      del.setVisible(true);
  this.dispose();}
{UpdateProd pro=new UpdateProd();
      pro.setVisible(true);
```

```
this.dispose();}
```

4.4.9 Add Product

This form allows you to add a product into the database.

Design :



Code:

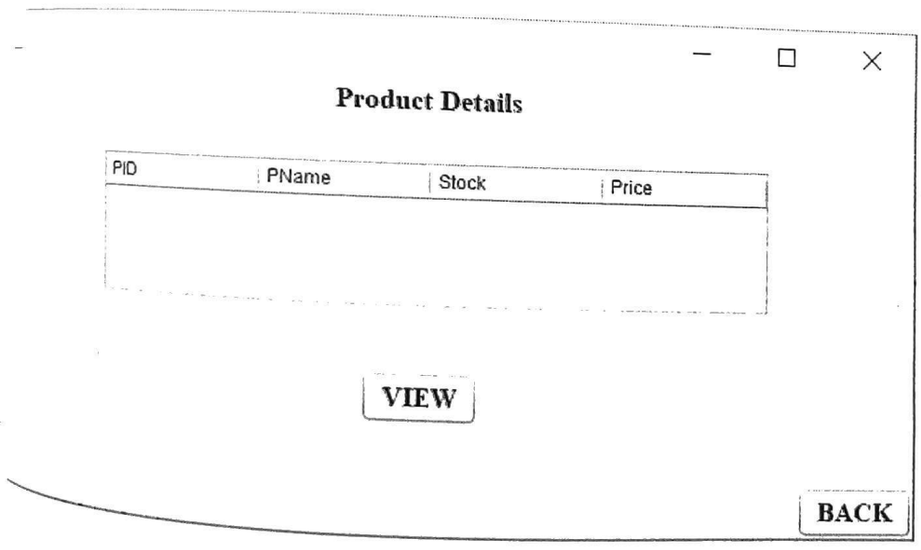
```
String PID=TxtPID.getText();
      String PName=TxtPName.getText();
      String Stock=TxtStock.getText();
      String Price=TxtPrice.getText();
      try
      {
        Class.forName("java.sql.DriverManager");
        Connection
Con=(Connection)DriverManager.getConnection("jdbc:mysql://local
host:3306/project","root","root");
```

```
        Statement stat=(Statement)con.createStatement();
        String query="INSERT INTO PRODUCT
VES('"+PID+"','"+PName+"','"+Stock+"','"+Price+"');";
        stat.executeUpdate(query);
        JOptionPane.showMessageDialog(this,"Product Added
cessfully");


    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(this,e.getMessage());
     }
```

4.4.10 View Products

This form allows the manager to view all the products present in the database.

Design :



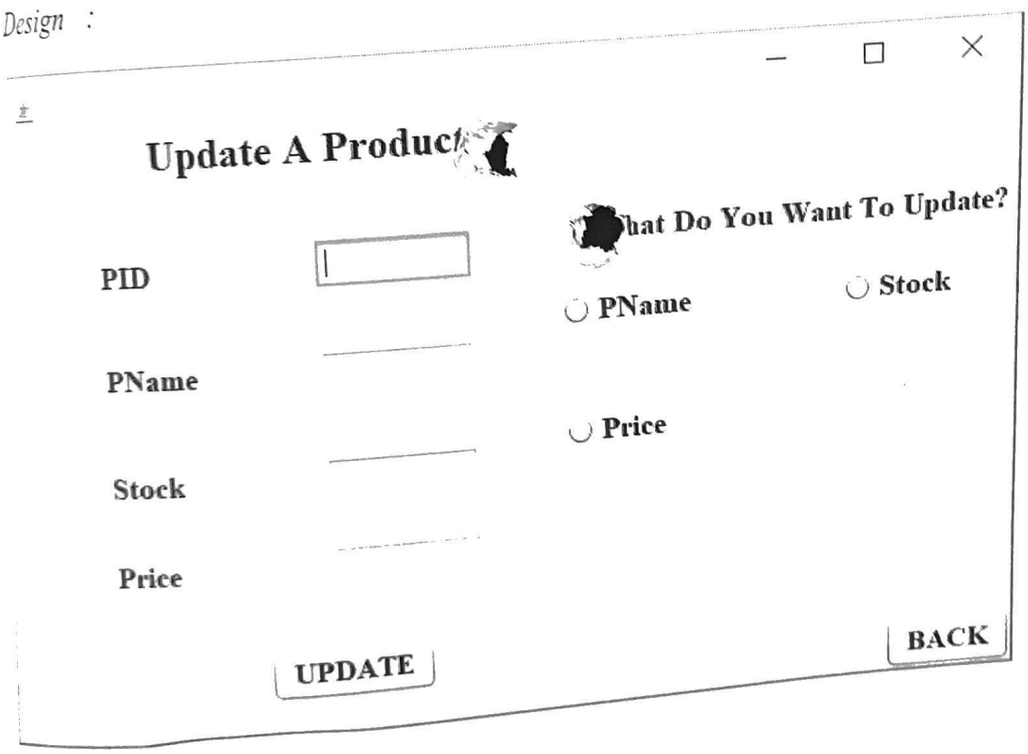
```
{
    Class.forName("java.sql.DriverManager");
    Connection
Connection)DriverManager.getConnection("jdbc:mysql://local
3306/project","root","root");
    Statement stat=(Statement)con.createStatement();
    DefaultTableModel
=(DefaultTableModel)jTable1.getModel();
    int rows=model.getRowCount();
    while(rows>0)
    {
        model.removeRow(0);
        rows--;
    }
    String query="Select * from product;";
    ResultSet rs=(ResultSet)stat.executeQuery(query);
    while(rs.next())
    {
        String PID=rs.getString("PID");
        String PName=rs.getString("PName");
        String Stock=rs.getString("Stock");
        String Price=rs.getString("Price");
        model.addRow(new Object[]
,PName,Stock,Price});
}

    rs.close();
```

```
        stat.close();
        con.close();
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(this,e.getMessage());
    }
```

4.4.11 Update Product Form

This form allows the manager to update the product details in the database.

Design :



Code:

```
String PID=TxtPID.getText();
        if(PID.isEmpty())
        {
```

```
            JOptionPane.showMessageDialog(this,"Enter the
product ID");
        }
        else
        {
            try
            {
                Class.forName("java.sql.DriverManager");
                Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://local
host:3306/project","root","root");
                Statement
stat=(Statement)con.createStatement();
                String query="SELECT PName,Stock,price from
product WHERE PID='"+PI[illegible]"'";
                ResultSet r[illegible]tat.executeQuery(query);
                if(rs.next())
                { String PName=rs.getString("PName");
                    String Stock=rs.getString("Stock");
                    String Price=rs.getString("Price");

                    TxtPName.setText(PName);
                    TxtStock.setText(Stock);
                    TxtPrice.setText(Price);

                    if(jRadioButton1.isSelected())
                    {
```

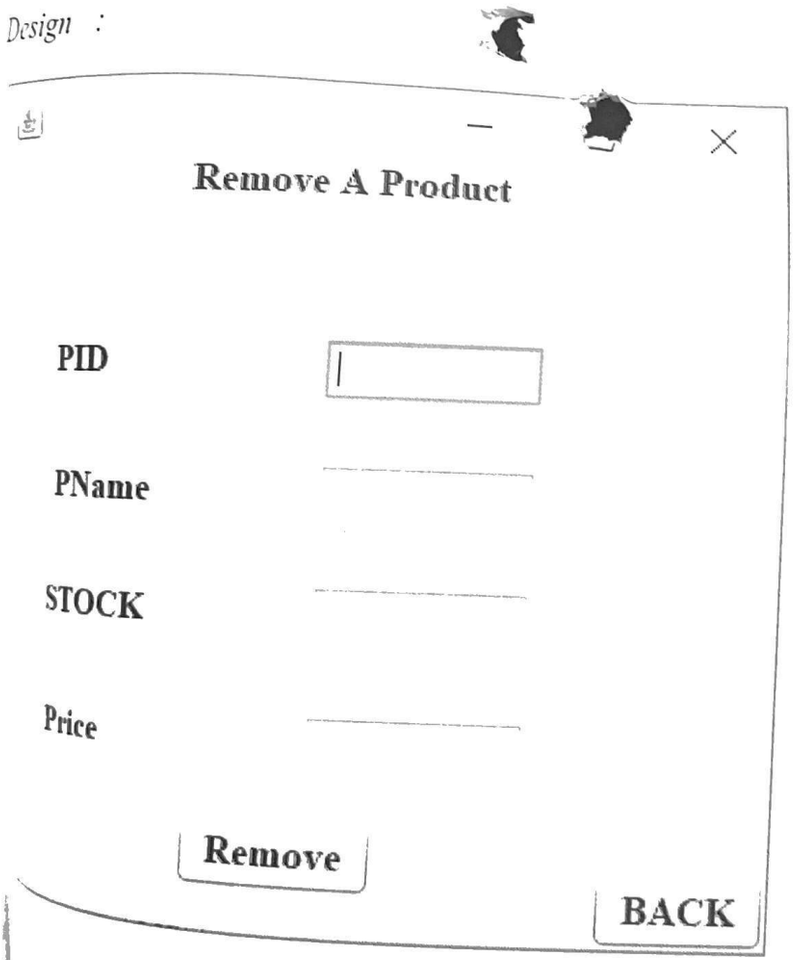
```
                String
NewName=JOptionPane.showInputDialog(this,"Enter New Name:");

pNAME='"+NewName+"' String query1="UPDATE product SET
                    WHERE PID='"+PID+"';";

                stat.executeUpdate(query1);

JOptionPane.showMessageDialog(this,"Name updated succesfully");

                }

                else if(jRadioButton2.isSelected())

                {

                    String
NewStock=JOptionPane.showInputDialog(this,"Enter Stock:");

                    String query1="UPDATE product set
stock='"+NewStock+"' WHERE PID='"+PID+"';";

                    stat.executeUpdate(query1);

JOptionPane.showMessageDialog(this,"Stock updated
succesfully");

                }

                else if(jRadioButton3.isSelected())

                {

                    String
NewPrice=JOptionPane.showInputDialog(this,"Enter New Price:");

                    String query1="UPDATE product set
Price='"+NewPrice+"' WHERE PID='"+PID+"';";

                stat.executeUpdate(query1);

OptionPane.showMessageDialog(this,"Price updated
ccesfully");
                }}
```

```
                else
Product ID Found");
                    JOptionPane.showMessageDialog(this,"No Such
                }
                catch(Exception e)
                {
JOptionPane.showMessageDialog(this,e.getMessage());
                }
        }
```

4.4.12 Delete Product

This form allows the manager to delete a product from the database.

Design :



```
ring PID=TxtID.getText();
        if(PID.isEmpty())
        {
            JOptionPane.showMessageDialog(this,"Enter the
oduct ID");

        }
        else
        {
            try
            {
                Class.forName("java.sql.DriverManager");
                Connection
=(Connection)DriverManager.getConnection("jdbc:mysql://local
st:3306/project","root","root");
                Statement
=(Statement)con.createStatement();
                String query="SELECT PID,PName,STOCK,Price FROM
CT WHERE PID='"+PID+"';";
                ResultSet rs=stat.executeQuery(query);
                if(rs.next())
                {

                    String PName=rs.getString("PName");
                    String Stock=rs.getString("Stock");
                    String Price=rs.getString("Price");
                    TxtPName.setText(PName);
```

```
                    TxtStock.setText(Stock);
                    TxtPrice.setText(Price);
                    String message="Are you sure to delete?";
                    String title="Confirm Deletion";
                    int
y=JOptionPane.showConfirmDialog(null,message,title,JOptionP
.YES_NO_OPTION);
                    if(reply==JOptionPane.YES_OPTION)
                    {
                        String query1="DELETE FROM PRODUCT
RE PID='"+PID+"';";
                        stat.executeUpdate(query1);
ionPane.showMessageDialog(this,"Record Deleted");
                    }
                    else
ionPane.showMessageDialog(this,"Record not deleted");
                }
                else
                    JOptionPane.showMessageDialog(this,"No Such
t ID Found");
            }
            catch(Exception e)
            {
```

```
OptionPane.showMessageDialog(this,e.getMessage());
            }

        }
```

Database Structure

1 Login:

Field	Type	Null	Key	Default	Extra
ername	varchar(25)	YES		NULL	
ssword	varchar(25)	YES		NULL	

Customers:

d	Type	Null	Key	Default	Extra
	varchar(25)	YES		NULL	
ss	varchar(25)	YES		NULL	
e	int(11)	YES		NULL	
	varchar(20)	YES		NULL	
e	int(11)	YES		NULL	

oducts:

Type	Null	Key	Default	Extra
int(11)	NO		NULL	
varchar(25)	NO		NULL	
int(11)	YES		NULL	
int(11)	NO		NULL	

5. CONCLUSION

Doing this project has helped us understand how online stores work and how certain programs can help make the work of online stores much easier. The speed and memory capacity of a computer combined with the creativity and innovation of human beings can create efficient and effective systems for the betterment of society.

We've learnt many things about Java programming and about the importance of designing the program and database according to the situational requirements. This has been a very interesting learning experience and we have benefitted greatly from it.

REFERENCES

https://searchoracle.techtarget.com/definition/MySQL - MySQL Content.

https://www.edureka.co/blog/what-is-java -Java Content.

Class 12 Informatics Practices NCERT

JAVA HANDBOOK

OOP- OBJECT ORIENTED PROGRAMMING

DATABSE SYSTEMS- A PRACTICAL APPROACH.