A VOCATIONAL TRAINING REPORT

ON

"PROGRAMMING WITH JAVA"

A Training Report Submitted to

CHHATTISGARH SWAMI VIVEKANANDA TECHNICAL UNIVERSITY, BHILAI (C.G.), INDIA

For the partial fulfillment of the award of degree

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING

By

MASTER. AKSHAT TIWARI

Under the Training of

Prof. Aditya Tiwari

Training In-Charge

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BHILAI INSTITUTE OF TECHNOLOGY, RAIPUR

Village – Kendri, Near Abhanpur, Atal Nagar, Raipur – 493661 (C.G.) India

BATCH 2019 - 2023

DECLARATION

I the undersigned solemnly declare that the Vocational Training report on **"PROGRAMMING WITH JAVA"** is based on my training work carried out during my vocational duration under the supervision of **MR. SYLVESTER DSOUZA, DIRECTOR** from **DSOUZA COMPUTER EDUCATION.**

I assert that the statements made and conclusions drawn are an outcome of the Vocational Training. I further declare that to the best of my knowledge and belief that this report does not contain any relevant work which has been submitted earlier.

Signature	**:**	
Student's Name	**:**	Master. Akshat Tiwari
Roll No.	**:**	309302219006
Enrollment No.	**:**	**BH1728**

CERTIFICATE

This is to certify that the report of my vocational training on **"PROGRAMMING WITH JAVA"** is the work carried out by **Master. Akshat Tiwari** studying in the 7th semester in **COMPUTER SCIENCE ENGINEERING** branch affiliated to Chhattisgarh Swami Vivekananda Technical University, Bhilai (C.G.), India under the guidance and supervision of **MR**. **SYLVESTER DSOUZA.**

To the best of my knowledge and belief the report

- Embodies the work of the candidate himself/herself.
- Has duly been completed.
- Fulfills the requirement of the ordinance relating to vocational training w.r.t. the university curriculum.

For being referred to the examiners.

Signature
HOD (C.S.E)

Signature
T&P In-Charge (C.S.E)

ACKNOWLEDGEMENT

It is my proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.

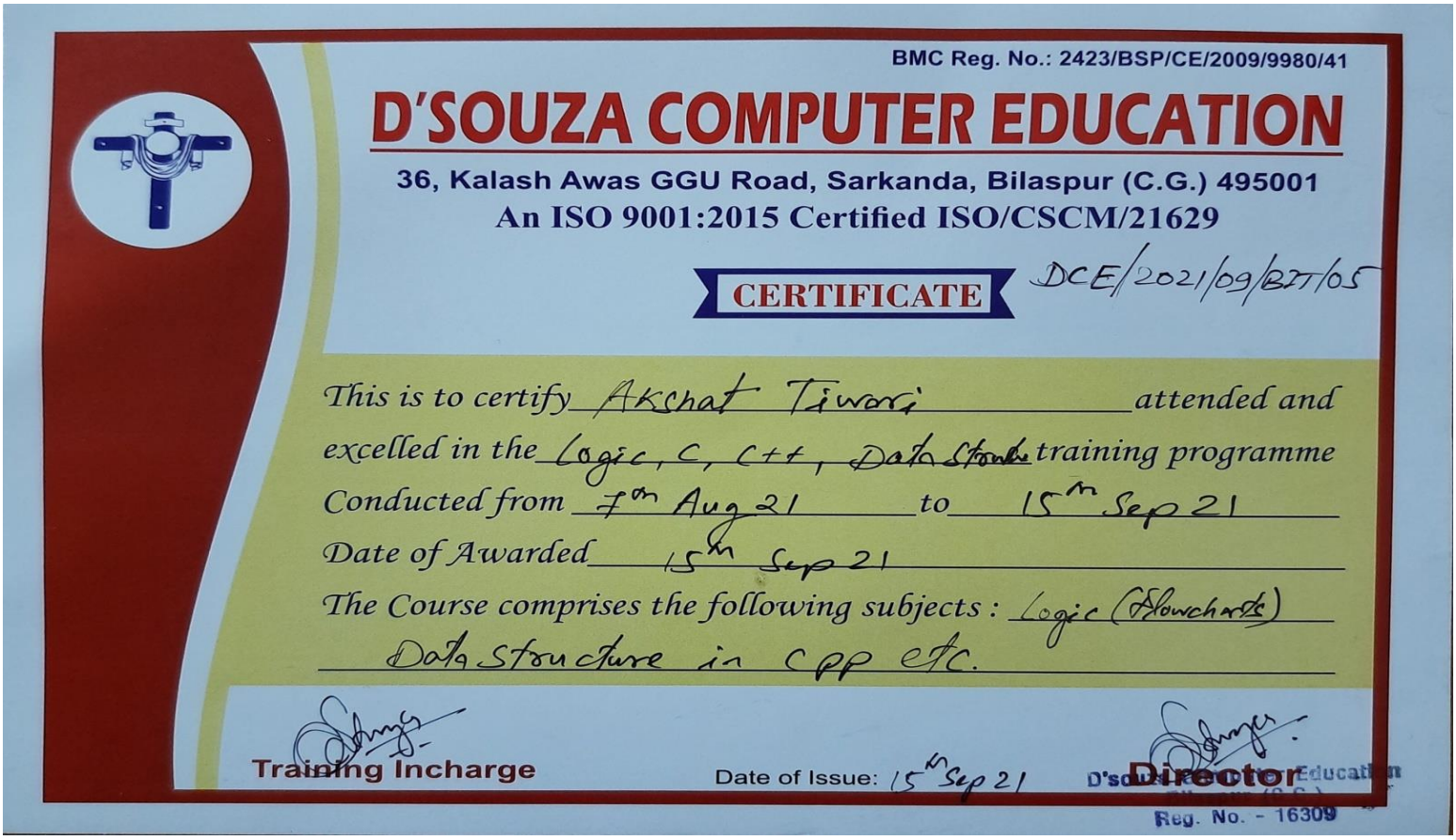
First and foremost, I wish to record our sincere gratitude to DSOUZA COMPUTER EDUCATION Coordinators for their constant support and encouragement in the preparation of this report and for making available videos and interface facilities needed to prepare this report.

The training on “Java” was very helpful to us in giving the necessary background

information and inspiration in choosing this topic for the seminar. Their contributions and technical support in preparing this report are greatly acknowledged.

Last but not the least, we wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

SUMMER TRAINING CERTIFICATE



BMC Reg. No.: 2423/BSP/CE/2009/9980/41

D'SOUZA COMPUTER EDUCATION

36, Kalash Awas GGU Road, Sarkanda, Bilaspur (C.G.) 495001

An ISO 9001:2015 Certified ISO/CSCM/21629

CERTIFICATE DCE/2021/09/BIT/05

This is to certify Akshat Tiwari attended and excelled in the Logic, C, C++, ~~Data Structure~~ training programme

Conducted from 1st Aug 21 to 15th Sep 21

Date of Awarded 15th Sep 21

The Course comprises the following subjects : Logic (Flowcharts)

Data Structure in CPP etc.

Training Incharge

Date of Issue: 15th Sep 21

Director

D'souza Computer Education

Reg. No. - 16309

TABLE OF CONTENTS

1. **Introduction to Programming with JAVA**
2. **Concepts of OOPS**
3. **Introduction to SQL**
4. **Training Contents**
5. **Profile Of the Problem**
6. **Screenshots**
7. **Coding**
8. **Bibliography**

INTRODUCTION TO PROGRAMMING WITH JAVA

Java is a popular programming language, created in 1995.

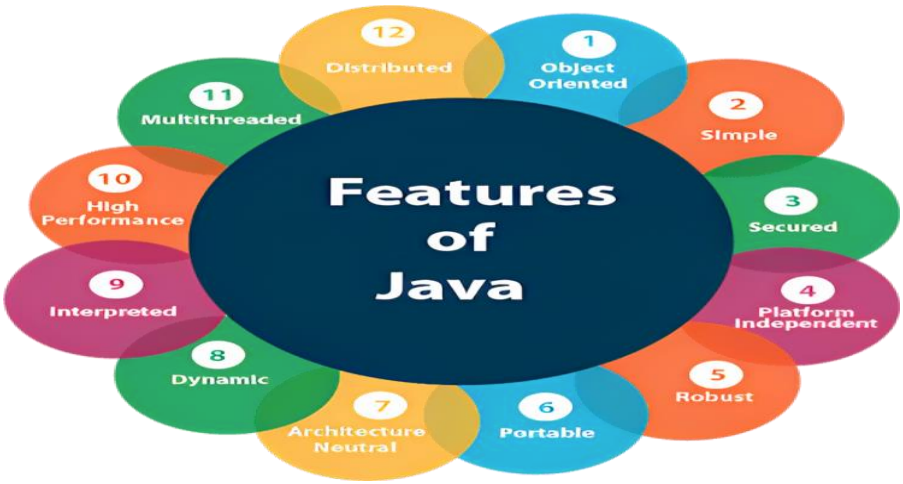
It is owned by Oracle, and more than **3 billion** devices run Java.

It is used for:

- Mobile applications (especially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

Java is a high-level and purely **object-oriented programming language**. It is platform independent, robust, secure, and multithreaded programming language which makes it popular among other OOP languages. It is widely used for software, web, mobile application development, along with this it is also used in big data analytics and server-side technology. Before moving towards features of Java, let us see how Java originated.

Features Of Java



Simple

Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun Microsystem, Java language is a simple programming language because:

Java syntax is based on C++ (so easier for programmers to learn it after C++).

Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.

There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

Object-oriented

Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporate both data and behavior.

Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Basic concepts of OOPs are:

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

Platform Independent

Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

There are two types of platforms software-based and hardware-based. Java provides a software-based platform.

The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on top of other hardware-based platforms. It has two components:

Runtime Environment

API (Application Programming Interface)

Java code can be executed on multiple platforms, for example, Windows, Linux, Sun Solaris, Mac/OS, etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms, i.e., Write Once and Run Anywhere (WORA).

Secured

Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:

No explicit pointer.

Java Programs run inside a virtual machine sandbox.

Classloader: Classloader in Java is a part of the Java Runtime Environment (JRE) which is used to load Java classes into the Java Virtual Machine dynamically. It adds security by separating the package for the classes of the local file system from those that are imported from network sources.

Bytecode Verifier: It checks the code fragments for illegal code that can violate access rights to objects.

Security Manager: It determines what resources a class can access such as reading and writing to the local disk.

Java language provides these securities by default. Some security can also be provided by an application developer explicitly through SSL, JAAS, Cryptography, etc.

Robust

The English mining of Robust is strong. Java is robust because:

It uses strong memory management.

There is a lack of pointers that avoids security problems.

Java provides automatic garbage collection which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.

There are exception handling and the type checking mechanism in Java. All these points make Java robust.

Architecture-neutral

Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.

In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. However, it occupies 4 bytes of memory for both 32 and 64-bit architectures in Java.

Portable

Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

High-performance

Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++). Java is an interpreted language that is why it is slower than compiled languages, e.g., C, C++, etc.

Distributed

Java is distributed because it facilitates users to create distributed applications in Java. RMI and EJB are used for creating distributed applications. This feature of Java makes us able to access files by calling the methods from any machine on the internet.

Multi-threaded

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each

thread. It shares a common memory area. Threads are important for multi-media, Web applications, etc.

Dynamic

Java is a dynamic language. It supports the dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.

Java supports dynamic compilation and automatic memory management (garbage collection).

Concept of OOPS in 'Java'

OOP stands for Object-Oriented Programming.

Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions.

Object-oriented programming has several advantages over procedural programming:

- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP makes it possible to create full reusable applications with less code and shorter development time.

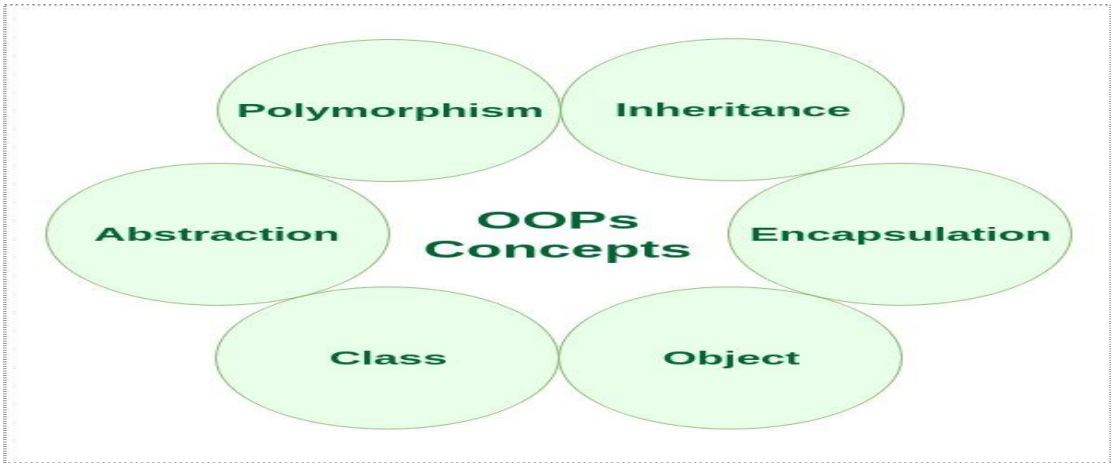


Figure (1) OOPS CONCEPTS

Object

This is the basic unit of object-oriented programming. That is both data and function that

operate on data are bundled as a unit called as object.

Class

When you define a class, you define a blueprint for an object. This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

Abstraction

Data abstraction refers to, providing only essential information to the outside world and hiding their background details, i.e., to represent the needed information in program without presenting the details.

For example, a database system hides certain details of how data is stored and created and maintained. Similar way, Java classes provides different methods to the outside world without giving internal detail about those methods and data.

Encapsulation

Encapsulation is placing the data and the functions that work on that data in the same place. While working with procedural languages, it is not always clear which functions work on which variables but object-oriented programming provides you framework to place the data and the relevant functions together in the same object.

Inheritance

One of the most useful aspects of object-oriented programming is code reusability. As the name suggests Inheritance is the process of forming a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class.

This is a very important concept of object-oriented programming since this feature helps to reduce the code size.

Polymorphism

The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called polymorphism. Poly refers to many. That is a single function or an operator functioning in many ways different upon the usage is called polymorphism.

Introduction to SQL

SQL is a standard language for accessing and manipulating databases.

SQL stands for Structured Query Language.

Features Of SQL

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views.

Some of The Most Important SQL Commands

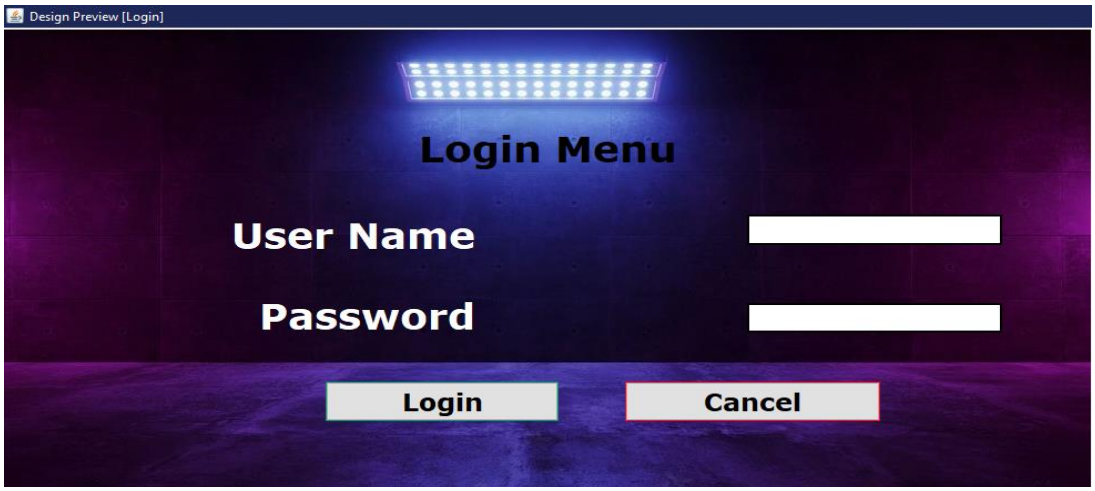
- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

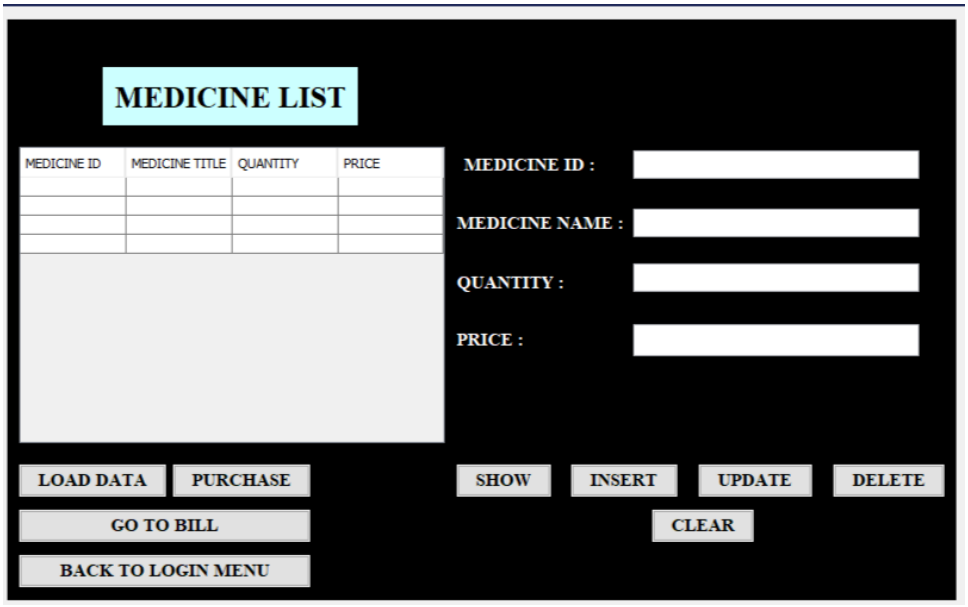
- Different Types of SQL JOINs
- Here are the different types of the JOINs in SQL:
- (INNER) JOIN: Returns records that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

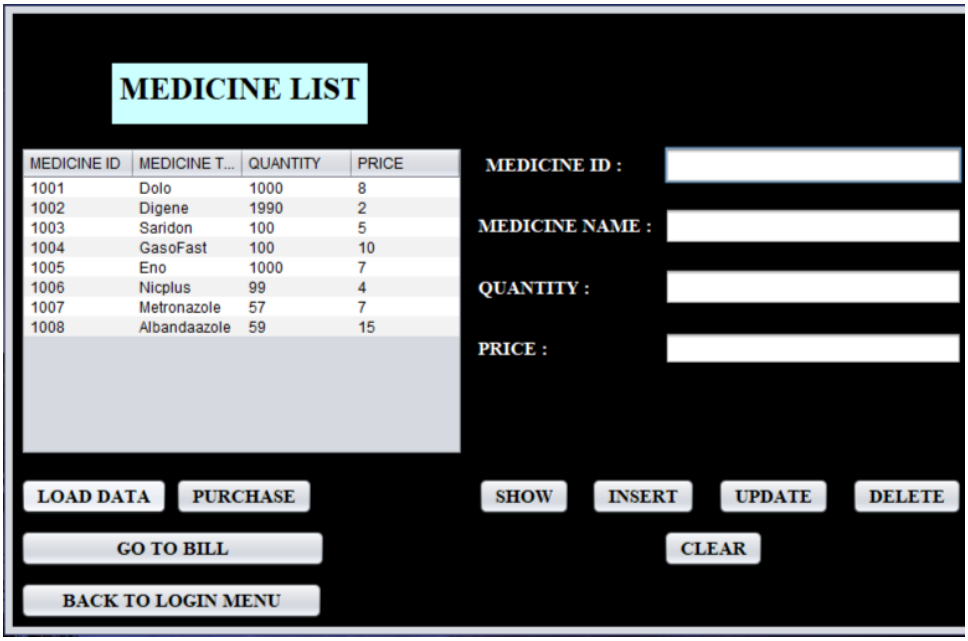
PROFILE OF THE PROBLEM

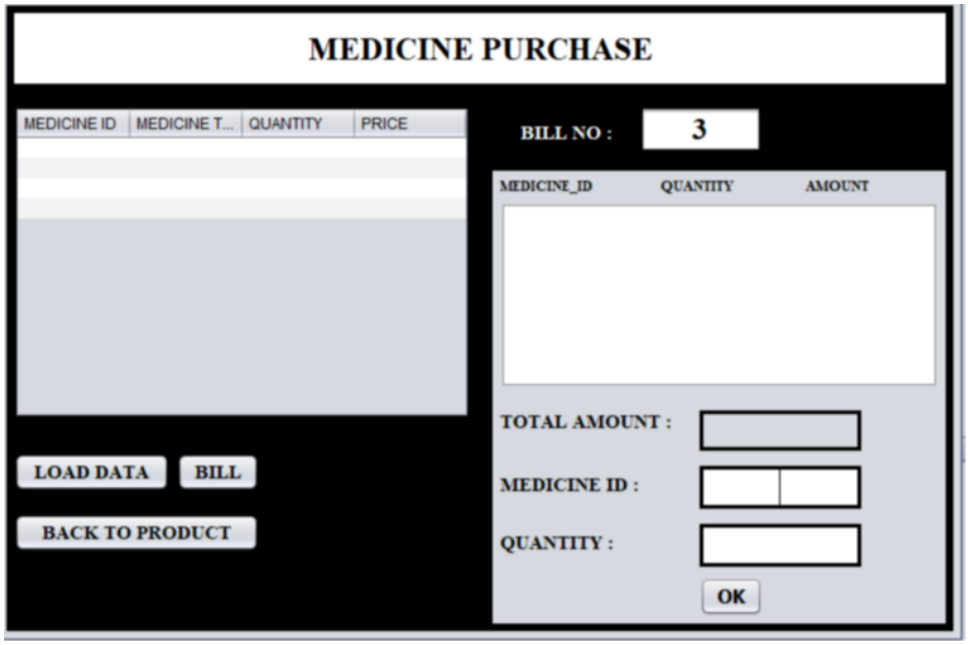
Create a Medical Store Management System Using Java and its Libraries having various kind of features such as Authentication, Stock Pile, Invoice Generation, Report Generation, etc.

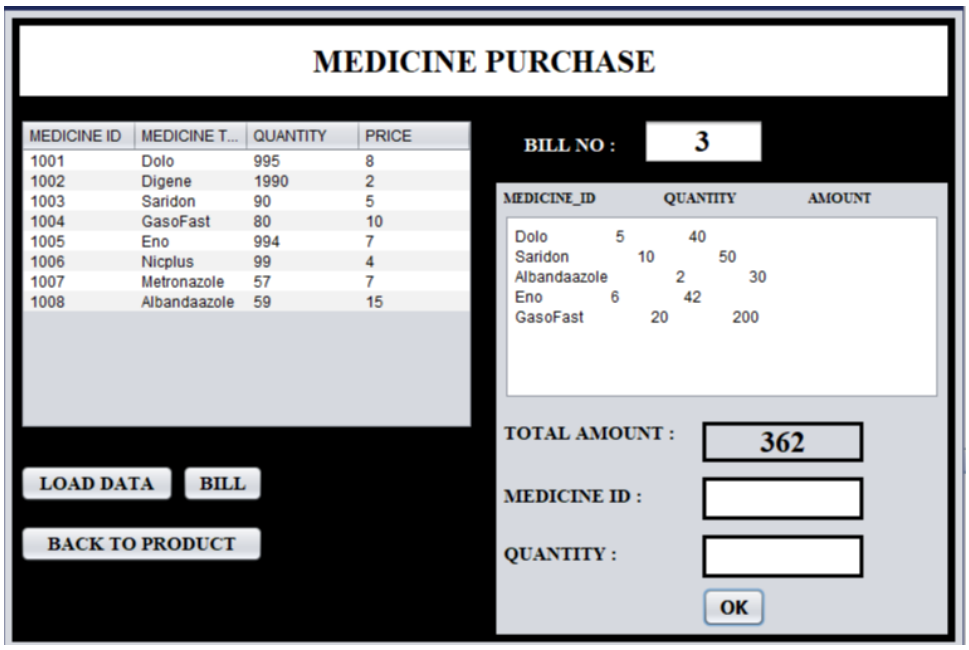
TESTING/DEPLOYMENT

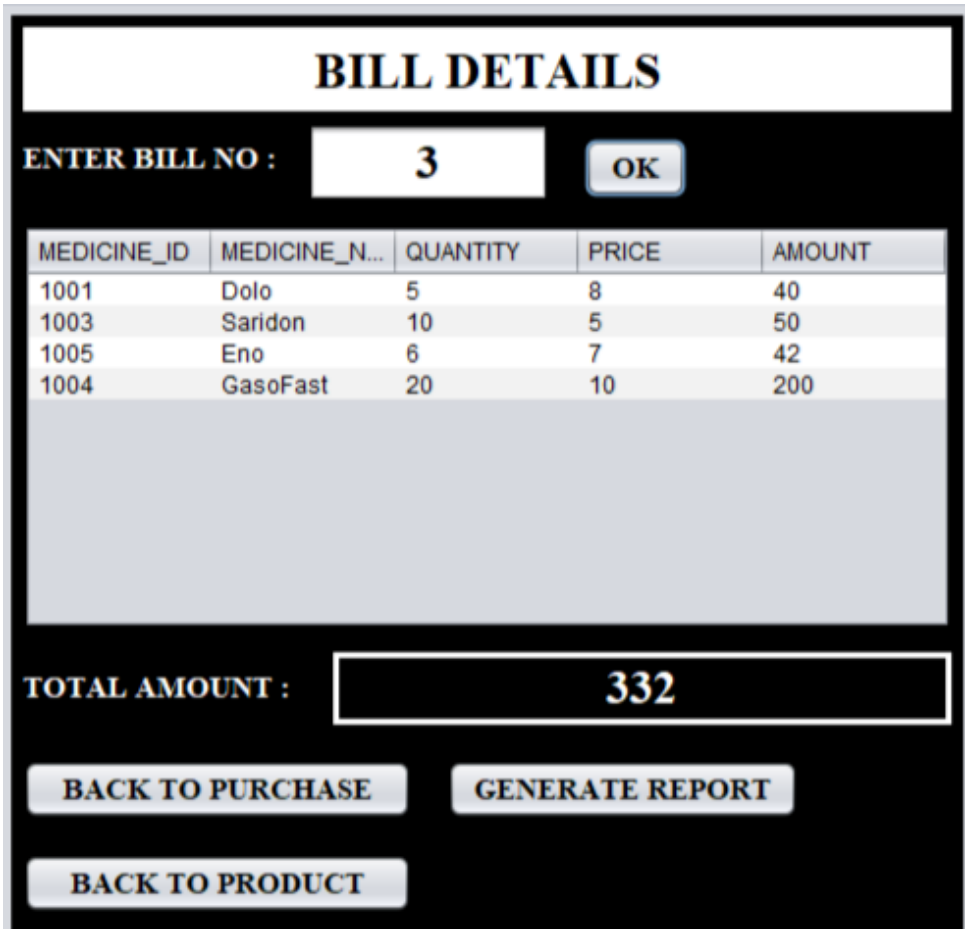


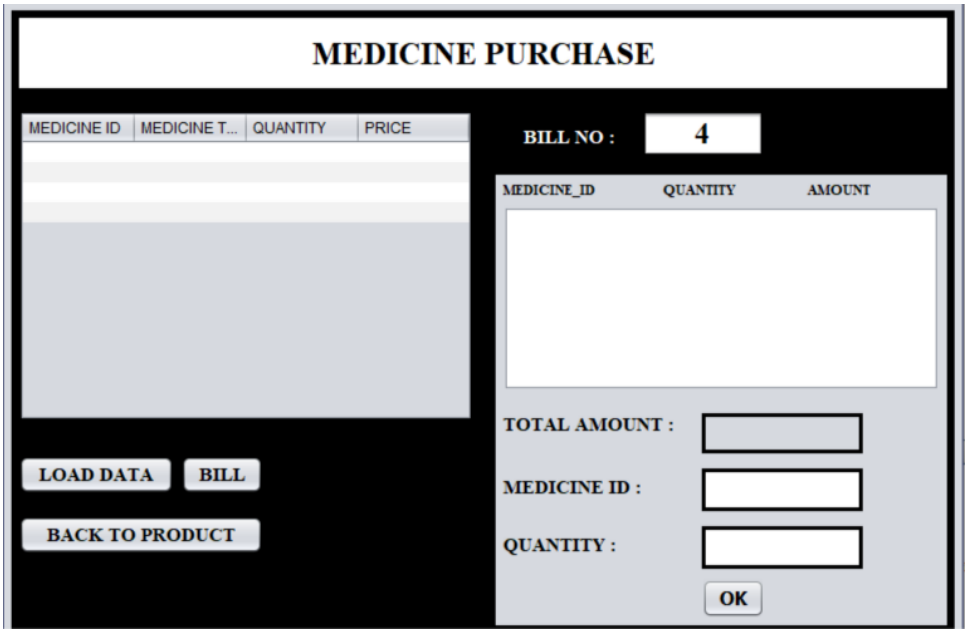

MEDICINE LIST
MEDICINE ID
MEDICINE TITLE
QUANTITY
PRICE
MEDICINE ID :
MEDICINE NAME :
QUANTITY :
PRICE :
LOAD DATA
PURCHASE
SHOW
INSERT
UPDATE
DELETE
GO TO BILL
CLEAR
BACK TO LOGIN MENU


MEDICINE LIST
MEDICINE ID
MEDICINE T...
QUANTITY
PRICE
1001 Dolo 1000 8
1002 Digene 1990 2
1003 Saridon 100 5
1004 GasoFast 100 10
1005 Eno 1000 7
1006 Nicplus 99 4
1007 Metronazole 57 7
1008 Albandaazole 59 15
MEDICINE ID :
MEDICINE NAME :
QUANTITY :
PRICE :
LOAD DATA
PURCHASE
SHOW
INSERT
UPDATE
DELETE
GO TO BILL
CLEAR
BACK TO LOGIN MENU


MEDICINE PURCHASE
MEDICINE ID
MEDICINE T...
QUANTITY
PRICE
BILL NO :
3
MEDICINE_ID
QUANTITY
AMOUNT
TOTAL AMOUNT :
MEDICINE ID :
QUANTITY :
OK
LOAD DATA
BILL
BACK TO PRODUCT


MEDICINE PURCHASE
MEDICINE ID
MEDICINE T...
QUANTITY
PRICE
1001 Dolo 995 8
1002 Digene 1990 2
1003 Saridon 90 5
1004 GasoFast 80 10
1005 Eno 994 7
1006 Nicplus 99 4
1007 Metronazole 57 7
1008 Albandaazole 59 15
BILL NO :
3
MEDICINE_ID
QUANTITY
AMOUNT
Dolo 5 40
Saridon 10 50
Albandaazole 2 30
Eno 6 42
GasoFast 20 200
TOTAL AMOUNT :
362
MEDICINE ID :
QUANTITY :
OK
LOAD DATA
BILL
BACK TO PRODUCT


BILL DETAILS
ENTER BILL NO :
3
OK
MEDICINE_ID
MEDICINE_N...
QUANTITY
PRICE
AMOUNT
1001
Dolo
5
8
40
1003
Saridon
10
5
50
1005
Eno
6
7
42
1004
GasoFast
20
10
200
TOTAL AMOUNT :
332
BACK TO PURCHASE
GENERATE REPORT
BACK TO PRODUCT


MEDICINE PURCHASE
MEDICINE ID
MEDICINE T...
QUANTITY
PRICE
BILL NO :
4
MEDICINE_ID
QUANTITY
AMOUNT
TOTAL AMOUNT :
MEDICINE ID :
QUANTITY :
OK
LOAD DATA
BILL
BACK TO PRODUCT

REPORT GENERATION

BILL NO - 3

SHOP INFORMATION

Name : Mahamrityunjaya Medical Agency

Address: Sarkanda, Bilaspur
C.G.

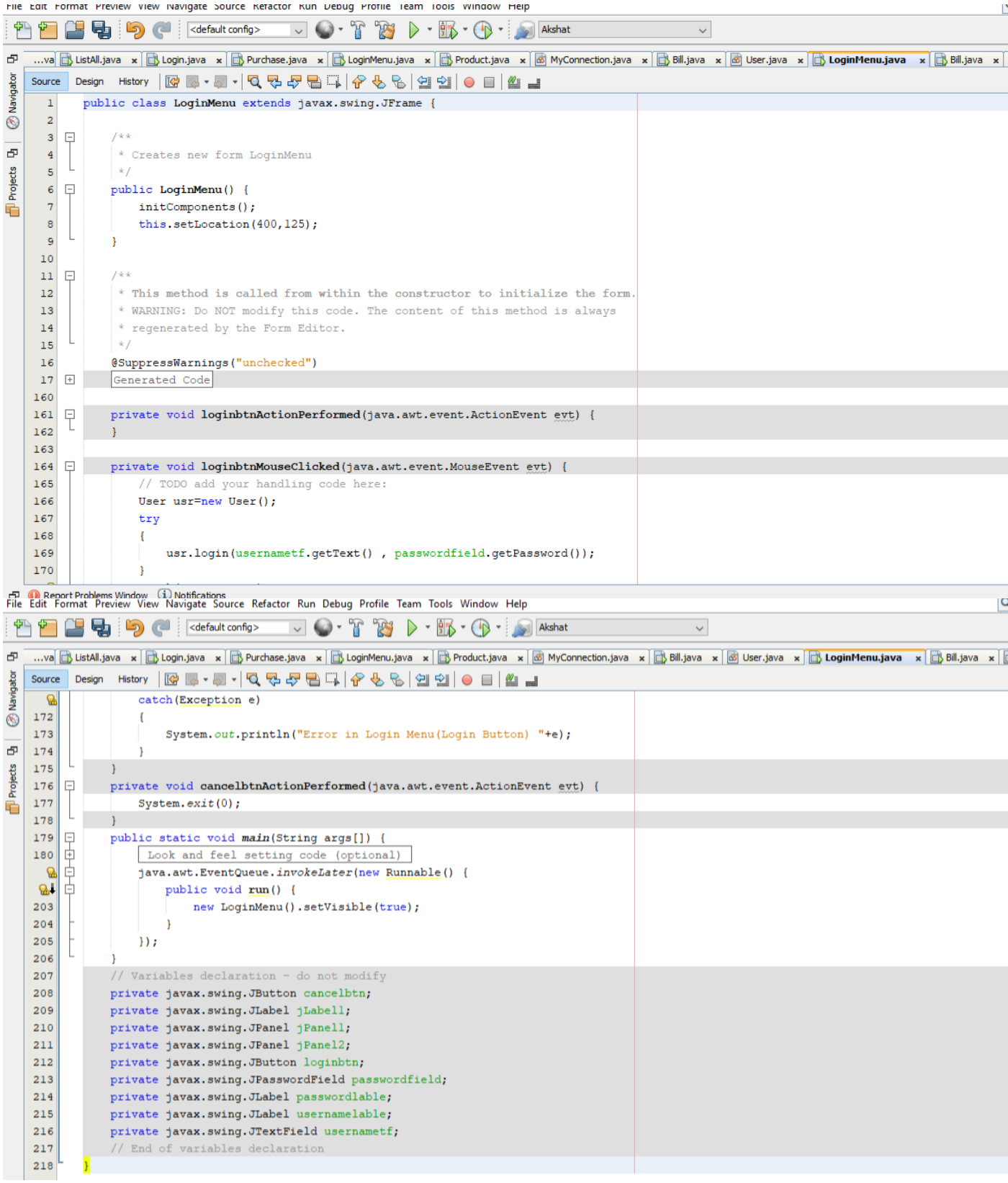
Phone No.: 9827150559, 07752-491733

Medicine	Quantity	Per Unit Price	Price
Dolo	5	8	40.00
Saridon	10	5	50.00
Eno	6	7	42.00
GasoFast	20	10	200.00
		Total :	
			332.00

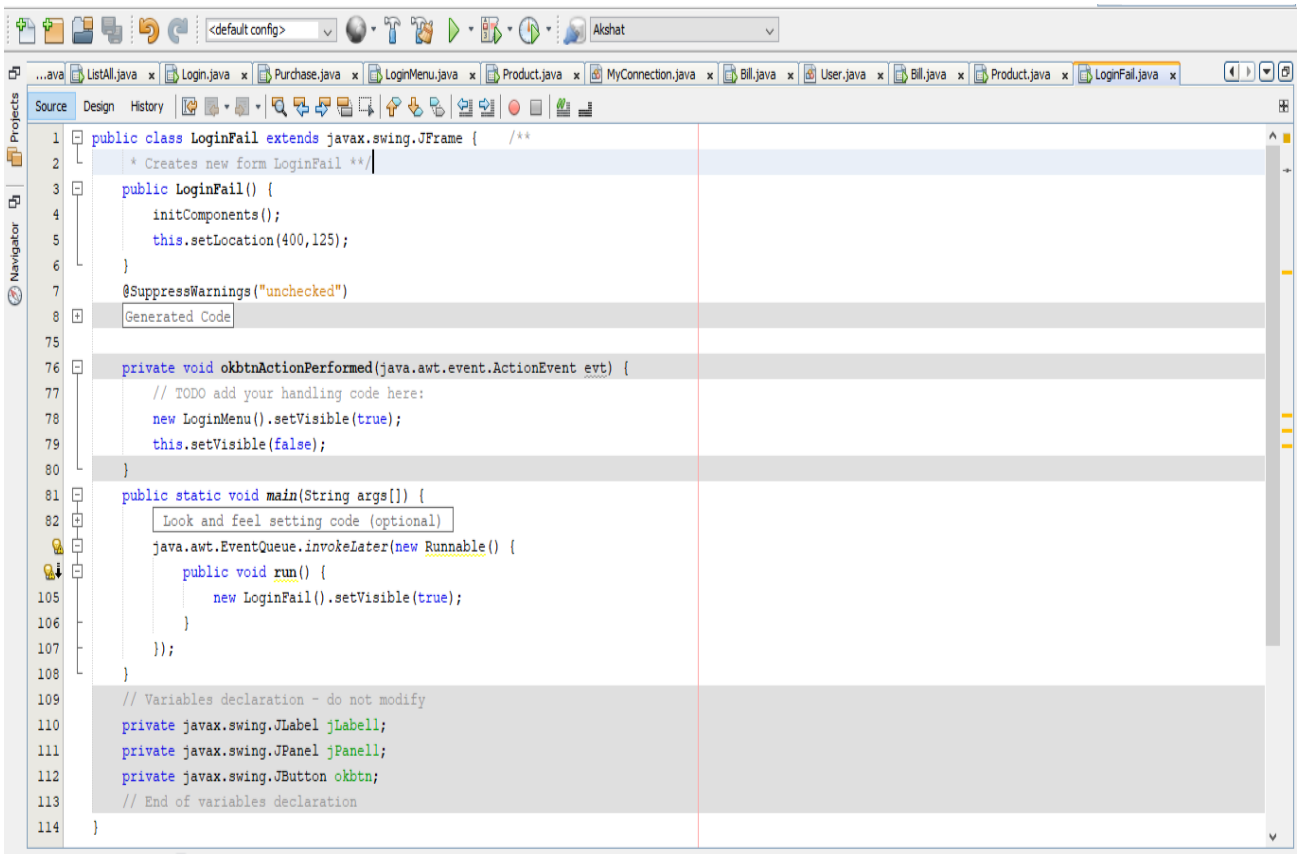
CODING

Now let's see the code of the implement application

1. LoginMenu.java

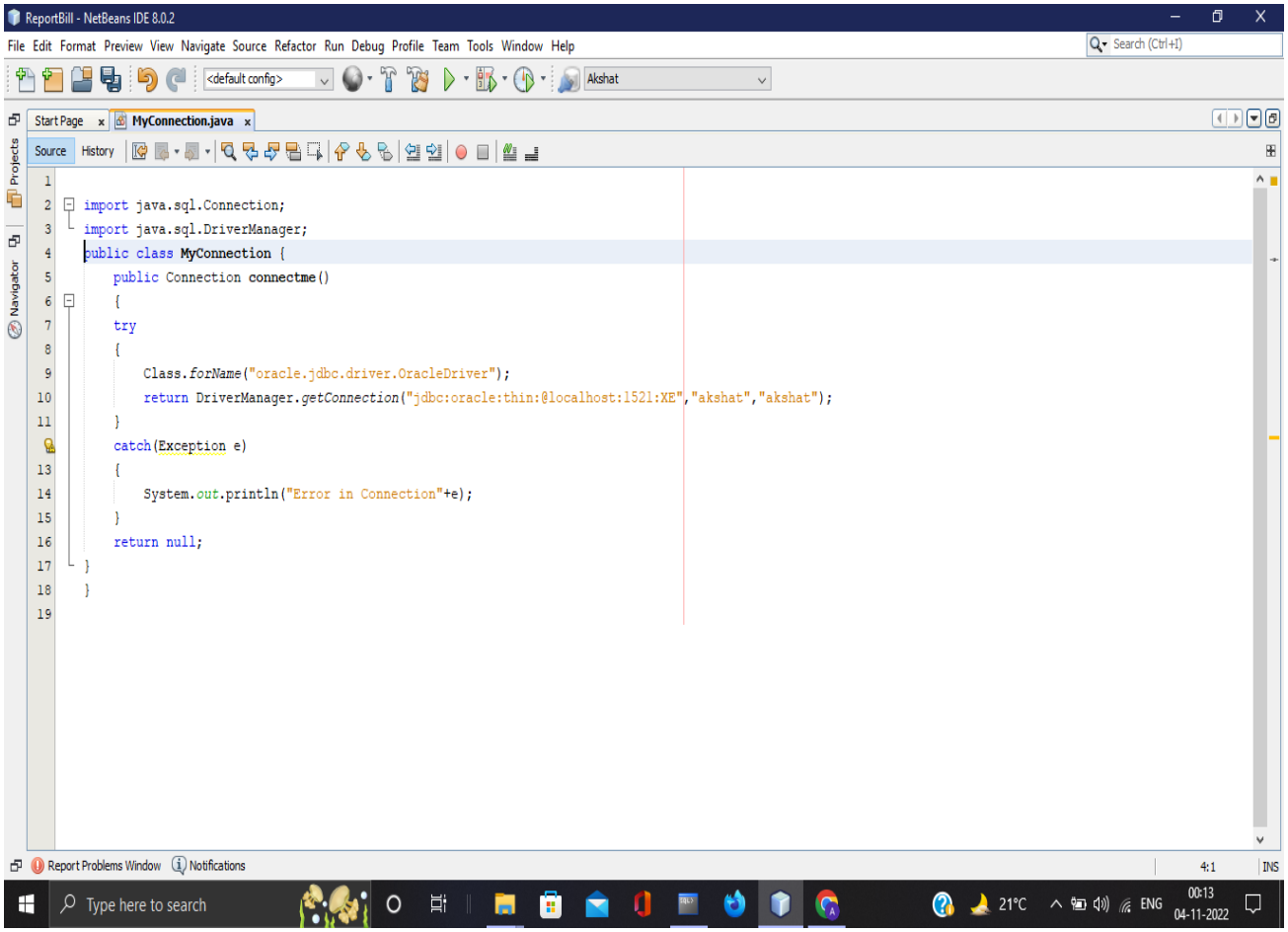

File Edit Format Preview View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Akshat
ListAll.java
Login.java
Purchase.java
LoginMenu.java
Product.java
MyConnection.java
Bill.java
User.java
LoginMenu.java
Bill.java
Source
Design
History
Navigator
Projects
public class LoginMenu extends javax.swing.JFrame {
/**
* Creates new form LoginMenu
*/
public LoginMenu() {
initComponents();
this.setLocation(400,125);
}
/**
* This method is called from within the constructor to initialize the form.
* WARNING: Do NOT modify this code. The content of this method is always
* regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
Generated Code
private void loginbtnActionPerformed(java.awt.event.ActionEvent evt) {
}
private void loginbtnMouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
User usr=new User();
try
{
usr.login(usernametf.getText() , passwordfield.getPassword());
}
Report Problems Window
Notifications
File Edit Format Preview View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Akshat
ListAll.java
Login.java
Purchase.java
LoginMenu.java
Product.java
MyConnection.java
Bill.java
User.java
LoginMenu.java
Bill.java
Source
Design
History
Navigator
Projects
catch(Exception e)
{
System.out.println("Error in Login Menu(Login Button) "+e);
}
}
private void cancelbtnActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0);
}
public static void main(String args[]) {
Look and feel setting code (optional)
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new LoginMenu().setVisible(true);
}
});
}
// Variables declaration - do not modify
private javax.swing.JButton cancelbtn;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JButton loginbtn;
private javax.swing.JPasswordField passwordfield;
private javax.swing.JLabel passwordlable;
private javax.swing.JLabel usernamelable;
private javax.swing.JTextField usernametf;
// End of variables declaration
}

2. LoginFail.java

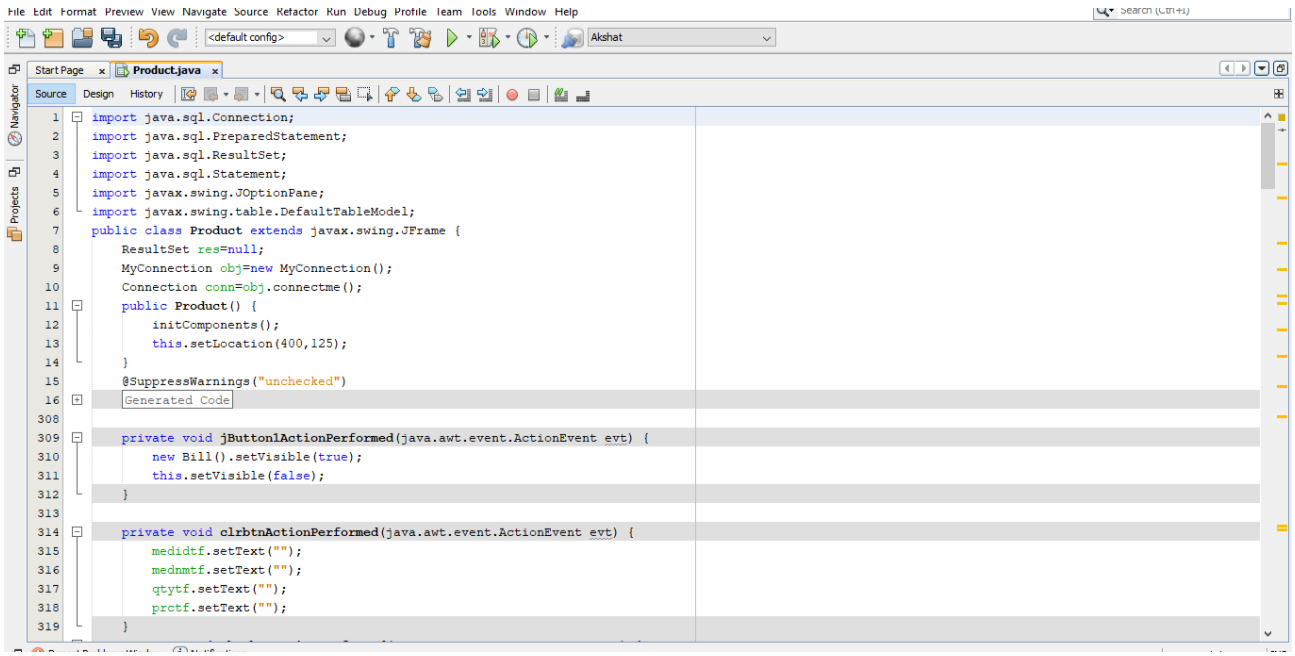


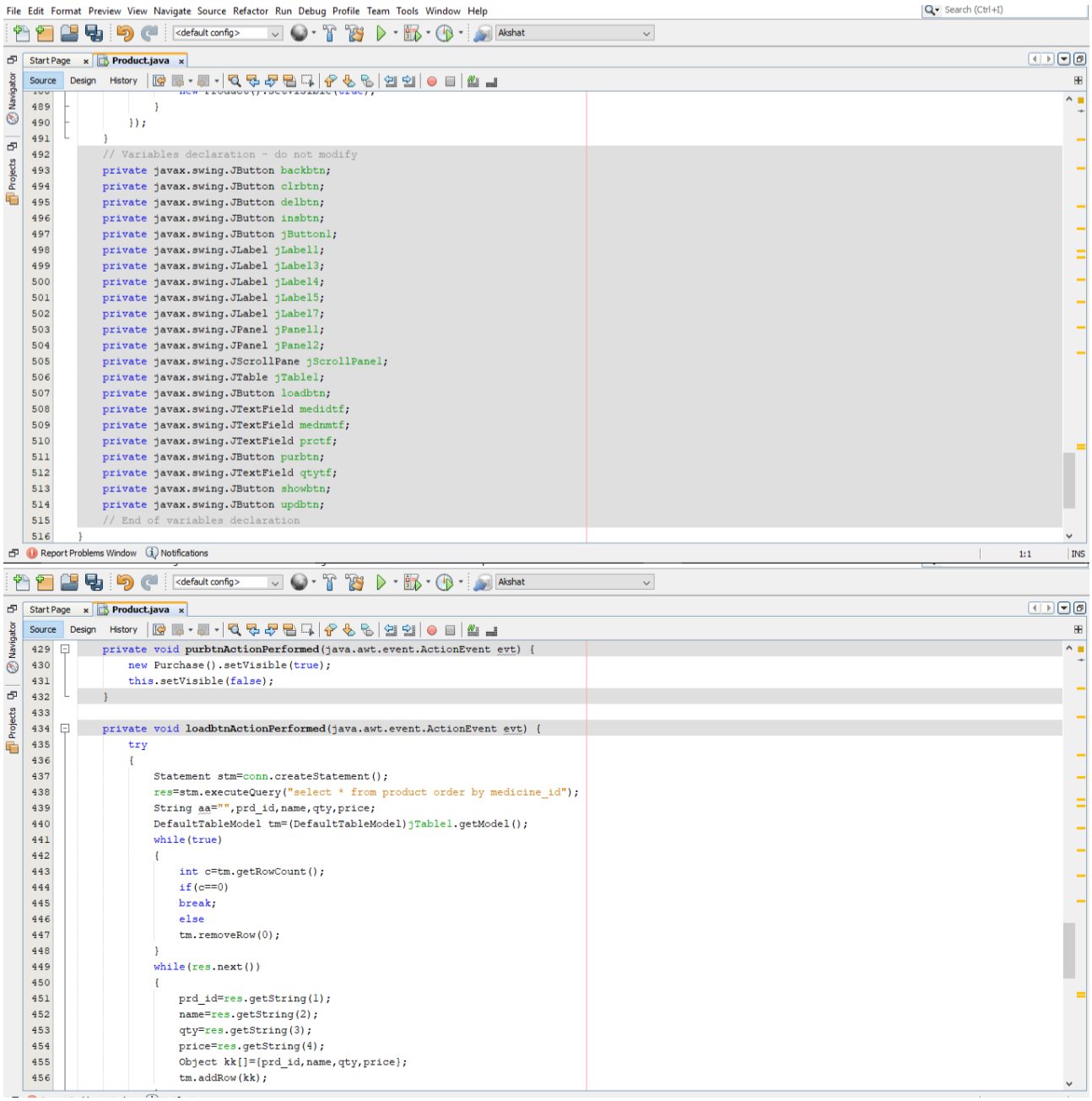
```
public class LoginFail extends javax.swing.JFrame {    /**
     * Creates new form LoginFail **/
    public LoginFail() {
        initComponents();
        this.setLocation(400,125);
    }
    @SuppressWarnings("unchecked")
    Generated Code

    private void okbtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new LoginMenu().setVisible(true);
        this.setVisible(false);
    }
    public static void main(String args[]) {
        Look and feel setting code (optional)
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new LoginFail().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JButton okbtn;
    // End of variables declaration
}
```

3. MyConnection.java

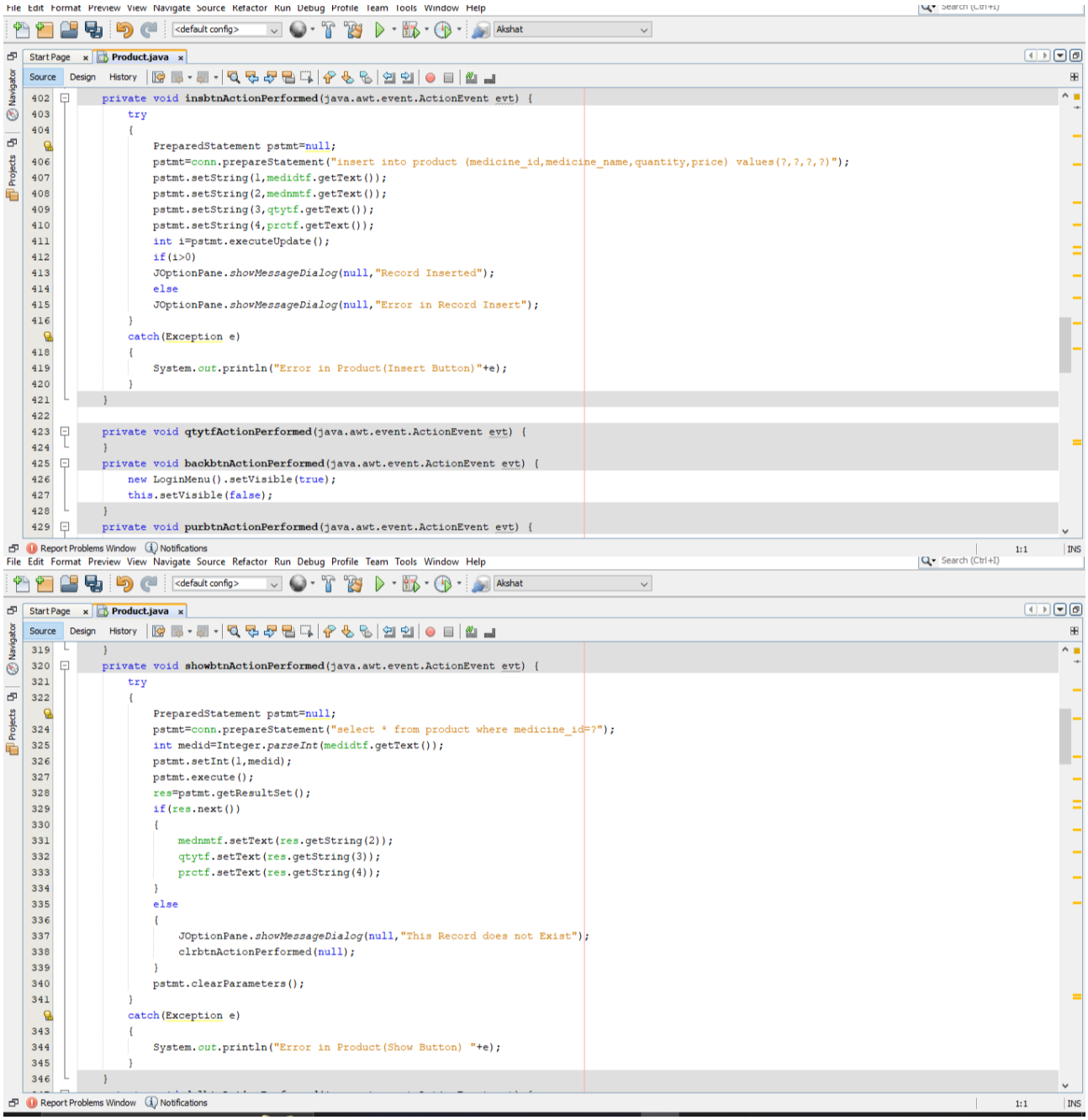


4. Product.java

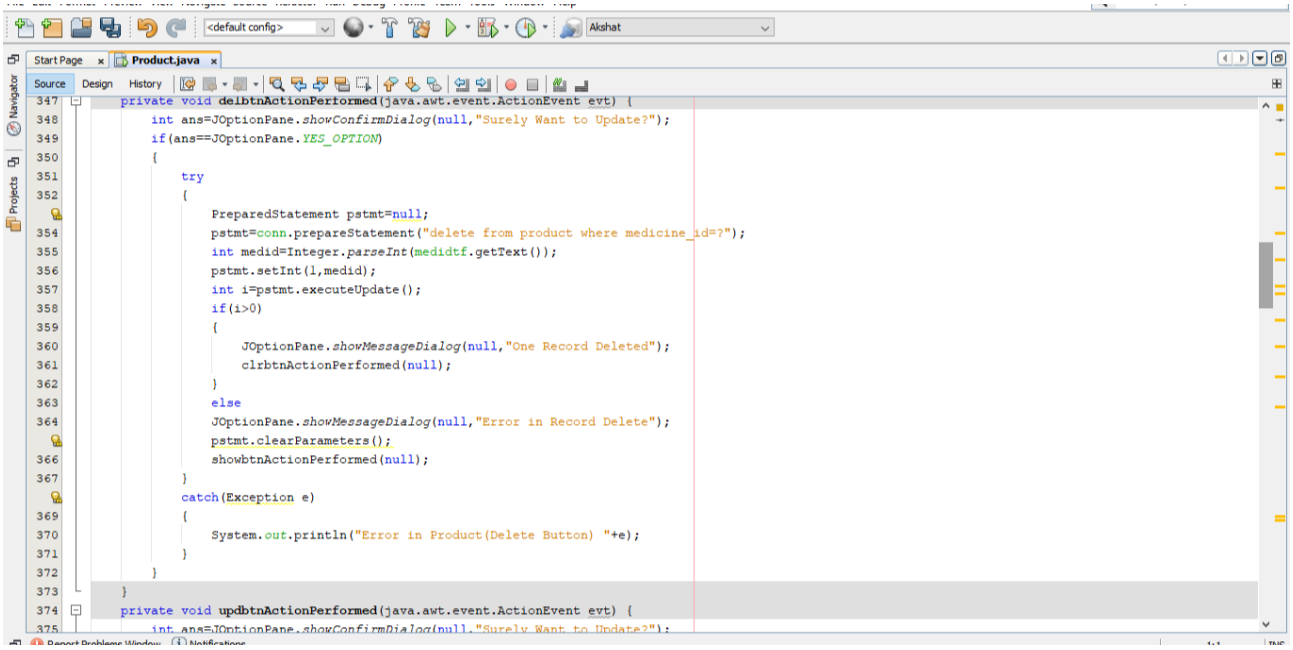



File Edit Format Preview View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Akshat
Start Page
Product.java
Source
Design
History
Navigator
Projects
}
});
}
// Variables declaration - do not modify
private javax.swing.JButton backbtn;
private javax.swing.JButton clrbtn;
private javax.swing.JButton delbtn;
private javax.swing.JButton insbtn;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel7;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JButton loadbtn;
private javax.swing.JTextField mediidtf;
private javax.swing.JTextField mednmtf;
private javax.swing.JTextField prctf;
private javax.swing.JButton purbtn;
private javax.swing.JTextField qtytf;
private javax.swing.JButton showbtn;
private javax.swing.JButton updbtn;
// End of variables declaration
}
Report Problems Window
Notifications
1:1
INS

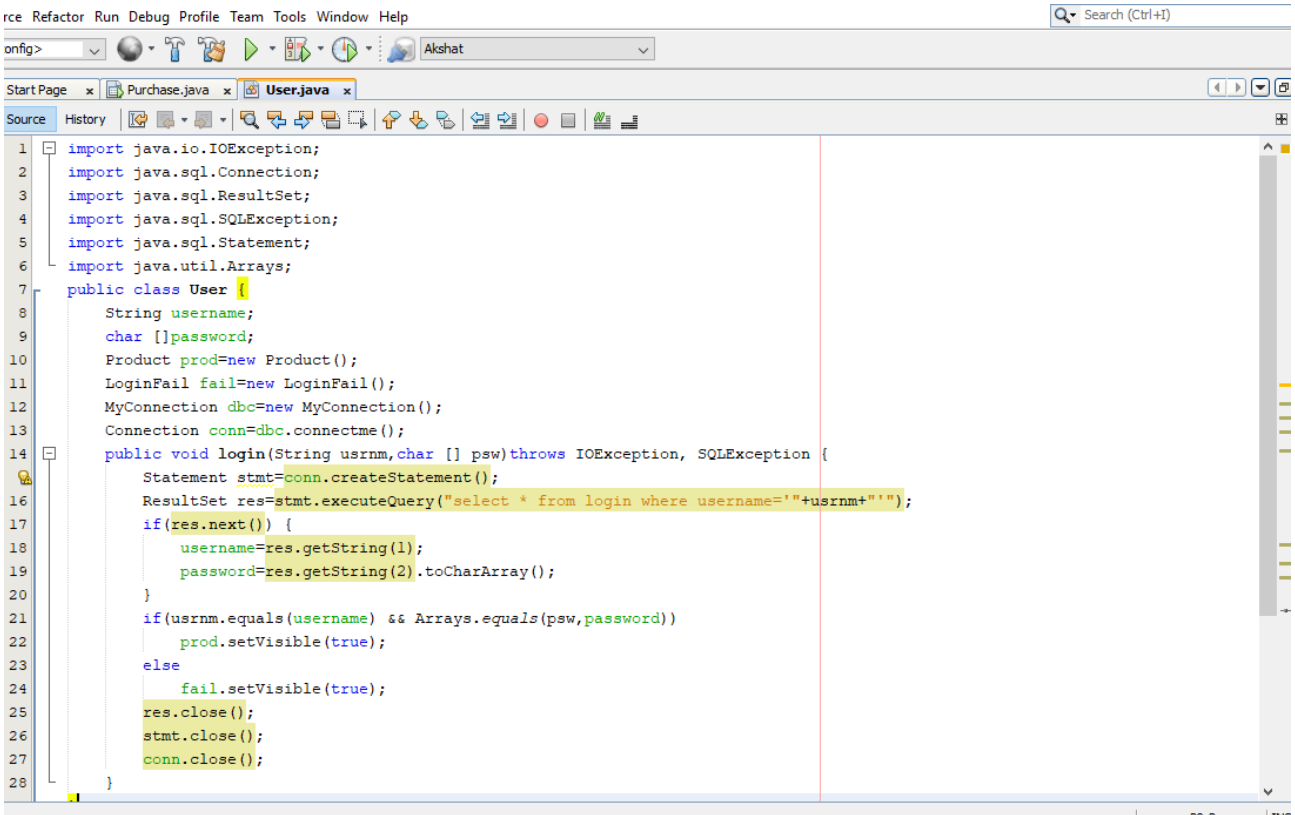
<default config>
Akshat
Start Page
Product.java
Source
Design
History
Navigator
Projects
private void purbtnActionPerformed(java.awt.event.ActionEvent evt) {
new Purchase().setVisible(true);
this.setVisible(false);
}
private void loadbtnActionPerformed(java.awt.event.ActionEvent evt) {
try
{
Statement stm=conn.createStatement();
res=stm.executeQuery("select * from product order by medicine_id");
String aa="",prd_id,name,qty,price;
DefaultTableModel tm=(DefaultTableModel)jTable1.getModel();
while(true)
{
int c=tm.getRowCount();
if(c==0)
break;
else
tm.removeRow(0);
}
while(res.next())
{
prd_id=res.getString(1);
name=res.getString(2);
qty=res.getString(3);
price=res.getString(4);
Object kk[]={prd_id,name,qty,price};
tm.addRow(kk);


File Edit Format Preview View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Akshat
Start Page
Product.java
Source
Design
History
Navigator
Projects
private void insbtnActionPerformed(java.awt.event.ActionEvent evt) {
try
{
PreparedStatement pstmt=null;
pstmt=conn.prepareStatement("insert into product (medicine_id,medicine_name,quantity,price) values(?,?,?,?)");
pstmt.setString(1,mediidtf.getText());
pstmt.setString(2,mednmtf.getText());
pstmt.setString(3,qtytf.getText());
pstmt.setString(4,prctf.getText());
int i=pstmt.executeUpdate();
if(i>0)
JOptionPane.showMessageDialog(null,"Record Inserted");
else
JOptionPane.showMessageDialog(null,"Error in Record Insert");
}
catch(Exception e)
{
System.out.println("Error in Product(Insert Button)"+e);
}
}
private void qtytfActionPerformed(java.awt.event.ActionEvent evt) {
}
private void backbtnActionPerformed(java.awt.event.ActionEvent evt) {
new LoginMenu().setVisible(true);
this.setVisible(false);
}
private void purbtnActionPerformed(java.awt.event.ActionEvent evt) {
Report Problems Window
Notifications
1:1
INS

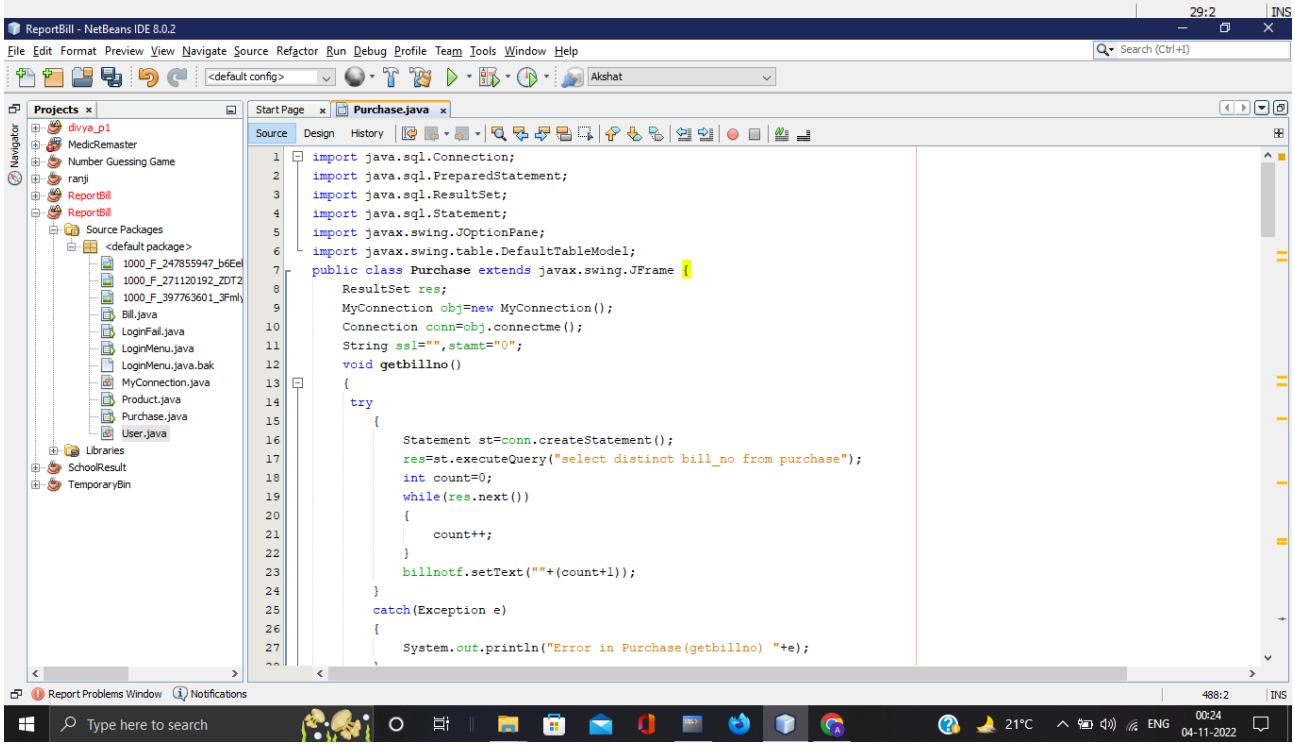
File Edit Format Preview View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Akshat
Start Page
Product.java
Source
Design
History
Navigator
Projects
}
private void showbtnActionPerformed(java.awt.event.ActionEvent evt) {
try
{
PreparedStatement pstmt=null;
pstmt=conn.prepareStatement("select * from product where medicine_id=?");
int medid=Integer.parseInt(mediidtf.getText());
pstmt.setInt(1,medid);
pstmt.execute();
res=pstmt.getResultSet();
if(res.next())
{
mednmtf.setText(res.getString(2));
qtytf.setText(res.getString(3));
prctf.setText(res.getString(4));
}
else
{
JOptionPane.showMessageDialog(null,"This Record does not Exist");
clrbtnActionPerformed(null);
}
pstmt.clearParameters();
}
catch(Exception e)
{
System.out.println("Error in Product(Show Button) "+e);
}
}
Report Problems Window
Notifications
1:1
INS



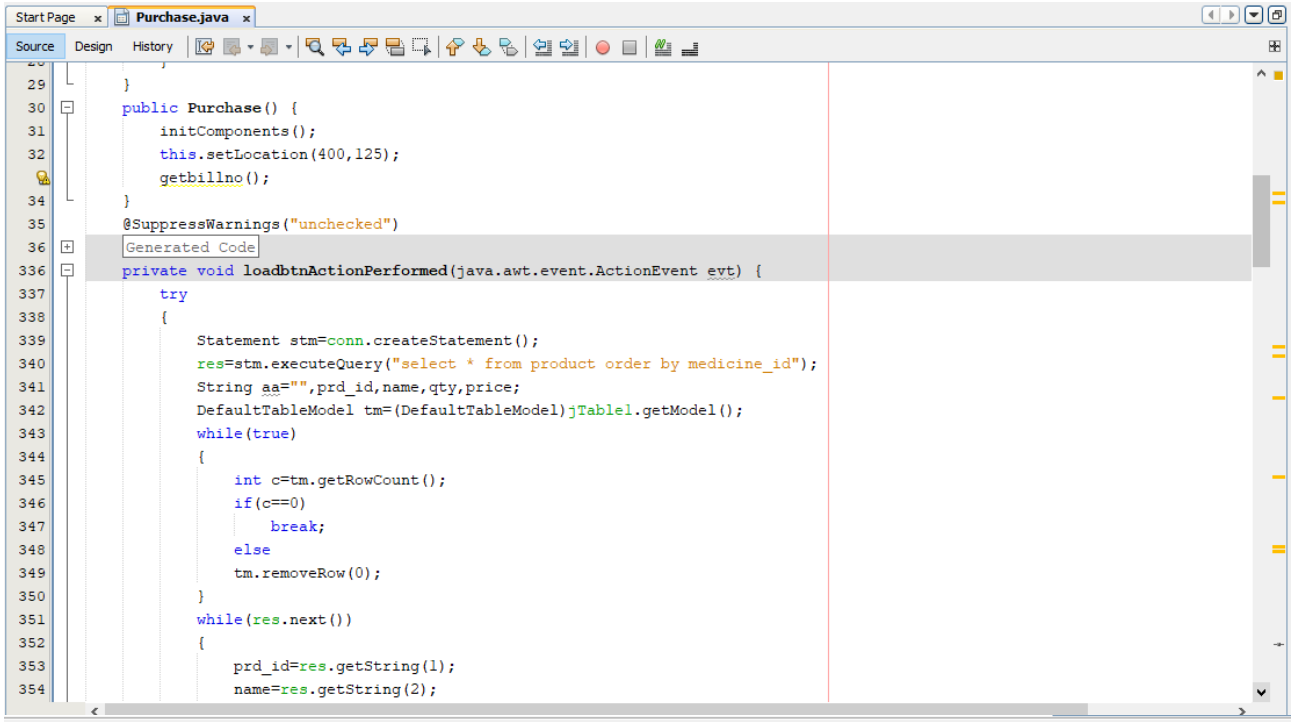
5. Purchase.java

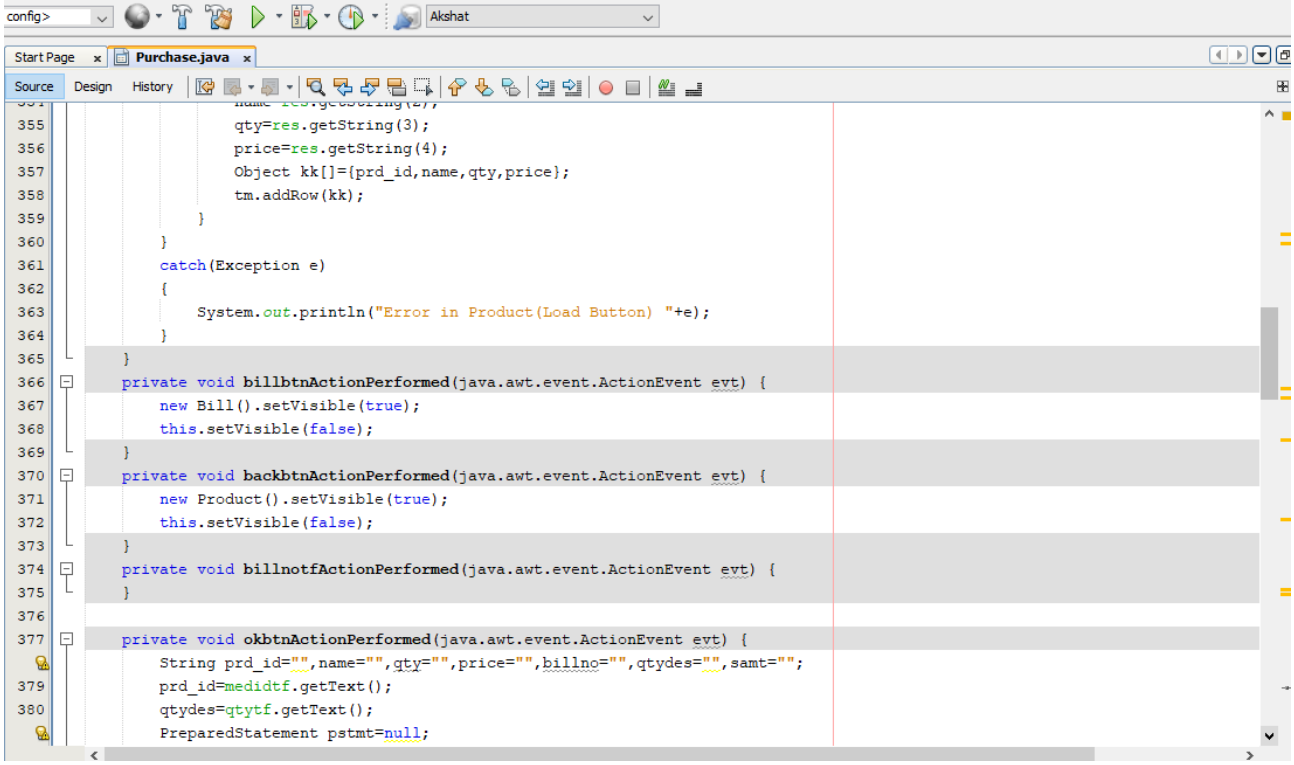


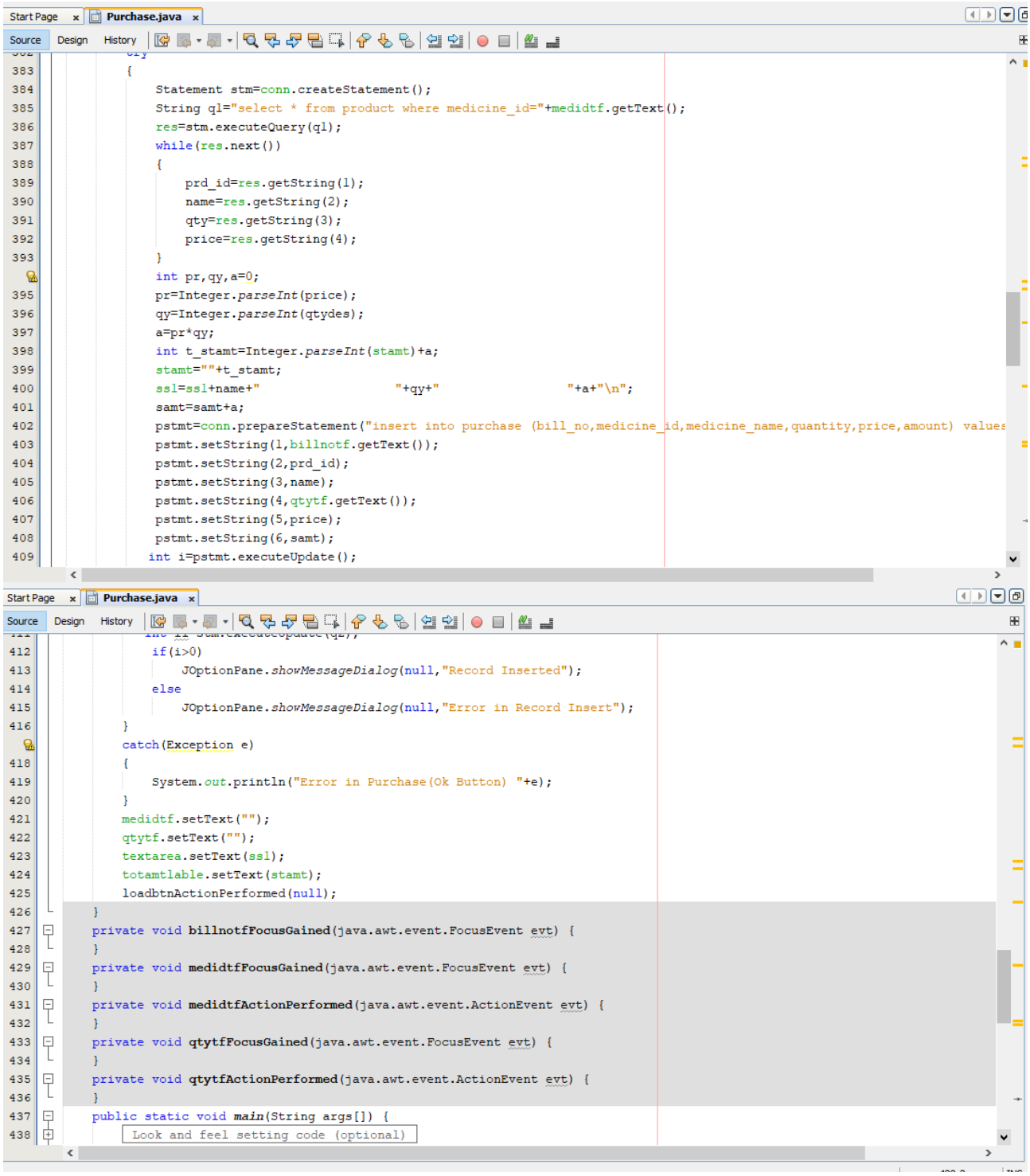
```
import java.io.IOException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Arrays;
public class User {
    String username;
    char []password;
    Product prod=new Product();
    LoginFail fail=new LoginFail();
    MyConnection dbc=new MyConnection();
    Connection conn=dbc.connectme();
    public void login(String usrnm,char [] psw)throws IOException, SQLException {
        Statement stmt=conn.createStatement();
        ResultSet res=stmt.executeQuery("select * from login where username='"+usrnm+"'");
        if(res.next()) {
            username=res.getString(1);
            password=res.getString(2).toCharArray();
        }
        if(usrnm.equals(username) && Arrays.equals(psw,password))
            prod.setVisible(true);
        else
            fail.setVisible(true);
        res.close();
        stmt.close();
        conn.close();
    }
```

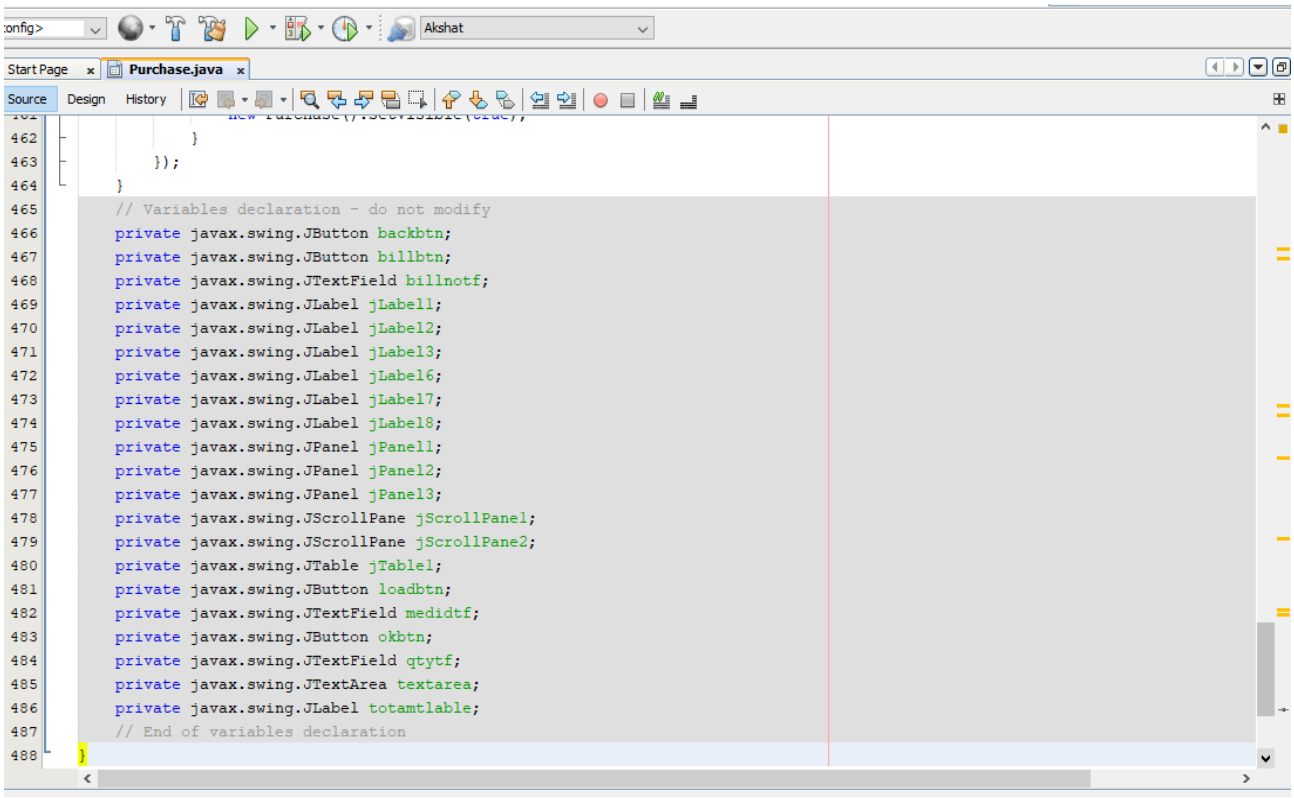


```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class Purchase extends javax.swing.JFrame {
    ResultSet res;
    MyConnection obj=new MyConnection();
    Connection conn=obj.connectme();
    String ssl="",stamt="0";
    void getbillno()
    {
     try
        {
            Statement st=conn.createStatement();
            res=st.executeQuery("select distinct bill_no from purchase");
            int count=0;
            while(res.next())
            {
                count++;
            }
            billnotf.setText(""+(count+1));
        }
        catch(Exception e)
        {
            System.out.println("Error in Purchase(getbillno) "+e);
```

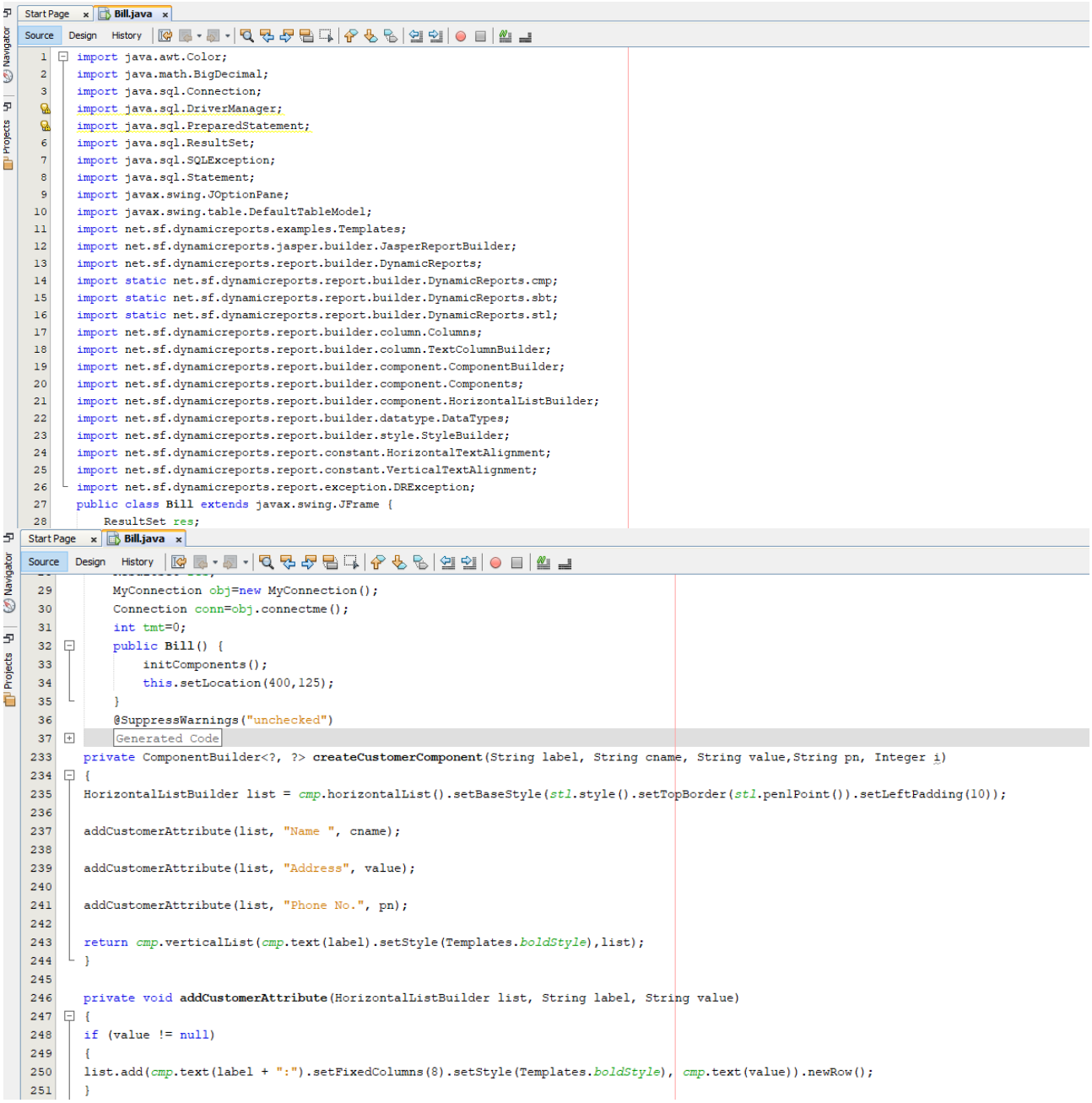

Start Page
Purchase.java
Source
Design
History
}
public Purchase() {
initComponents();
this.setLocation(400,125);
getbillno();
}
@SuppressWarnings("unchecked")
Generated Code
private void loadbtnActionPerformed(java.awt.event.ActionEvent evt) {
try
{
Statement stm=conn.createStatement();
res=stm.executeQuery("select * from product order by medicine_id");
String aa="",prd_id,name,qty,price;
DefaultTableModel tm=(DefaultTableModel)jTable1.getModel();
while(true)
{
int c=tm.getRowCount();
if(c==0)
break;
else
tm.removeRow(0);
}
while(res.next())
{
prd_id=res.getString(1);
name=res.getString(2);


Start Page
Purchase.java
Source
Design
History
qty=res.getString(3);
price=res.getString(4);
Object kk[]={prd_id,name,qty,price};
tm.addRow(kk);
}
}
catch(Exception e)
{
System.out.println("Error in Product(Load Button) "+e);
}
}
private void billbtnActionPerformed(java.awt.event.ActionEvent evt) {
new Bill().setVisible(true);
this.setVisible(false);
}
private void backbtnActionPerformed(java.awt.event.ActionEvent evt) {
new Product().setVisible(true);
this.setVisible(false);
}
private void billnotfActionPerformed(java.awt.event.ActionEvent evt) {
}
private void okbtnActionPerformed(java.awt.event.ActionEvent evt) {
String prd_id="",name="",qty="",price="",billno="",qtydes="",samt="";
prd_id=medidtf.getText();
qtydes=qtytf.getText();
PreparedStatement pstmt=null;


Start Page
Purchase.java
Source
Design
History
try
{
Statement stm=conn.createStatement();
String ql="select * from product where medicine_id="+meditf.getText();
res=stm.executeQuery(ql);
while(res.next())
{
prd_id=res.getString(1);
name=res.getString(2);
qty=res.getString(3);
price=res.getString(4);
}
int pr,qy,a=0;
pr=Integer.parseInt(price);
qy=Integer.parseInt(qtydes);
a=pr*qy;
int t_stamt=Integer.parseInt(stamt)+a;
stamt=""+t_stamt;
ssl=ssl+name+" "+qy+" "+a+"\n";
samt=samt+a;
pstmt=conn.prepareStatement("insert into purchase (bill_no,medicine_id,medicine_name,quantity,price,amount) values
pstmt.setString(1,billnotf.getText());
pstmt.setString(2,prd_id);
pstmt.setString(3,name);
pstmt.setString(4,qtytf.getText());
pstmt.setString(5,price);
pstmt.setString(6,samt);
int i=pstmt.executeUpdate();
Start Page
Purchase.java
Source
Design
History
if(i>0)
JOptionPane.showMessageDialog(null,"Record Inserted");
else
JOptionPane.showMessageDialog(null,"Error in Record Insert");
}
catch(Exception e)
{
System.out.println("Error in Purchase(Ok Button) "+e);
}
meditf.setText("");
qtytf.setText("");
textarea.setText(ssl);
totamtlable.setText(stamt);
loadbtnActionPerformed(null);
}
private void billnotfFocusGained(java.awt.event.FocusEvent evt) {
}
private void meditfFocusGained(java.awt.event.FocusEvent evt) {
}
private void meditfActionPerformed(java.awt.event.ActionEvent evt) {
}
private void qtytfFocusGained(java.awt.event.FocusEvent evt) {
}
private void qtytfActionPerformed(java.awt.event.ActionEvent evt) {
}
public static void main(String args[]) {
Look and feel setting code (optional)


Akshat
Start Page
Purchase.java
Source
Design
History
}
});
}
// Variables declaration - do not modify
private javax.swing.JButton backbtn;
private javax.swing.JButton billbtn;
private javax.swing.JTextField billnotf;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPanel;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTable1;
private javax.swing.JButton loadbtn;
private javax.swing.JTextField medidtf;
private javax.swing.JButton okbtn;
private javax.swing.JTextField qtytf;
private javax.swing.JTextArea textarea;
private javax.swing.JLabel totamtlable;
// End of variables declaration
}

5. Bill.java



Start Page | Bill.java

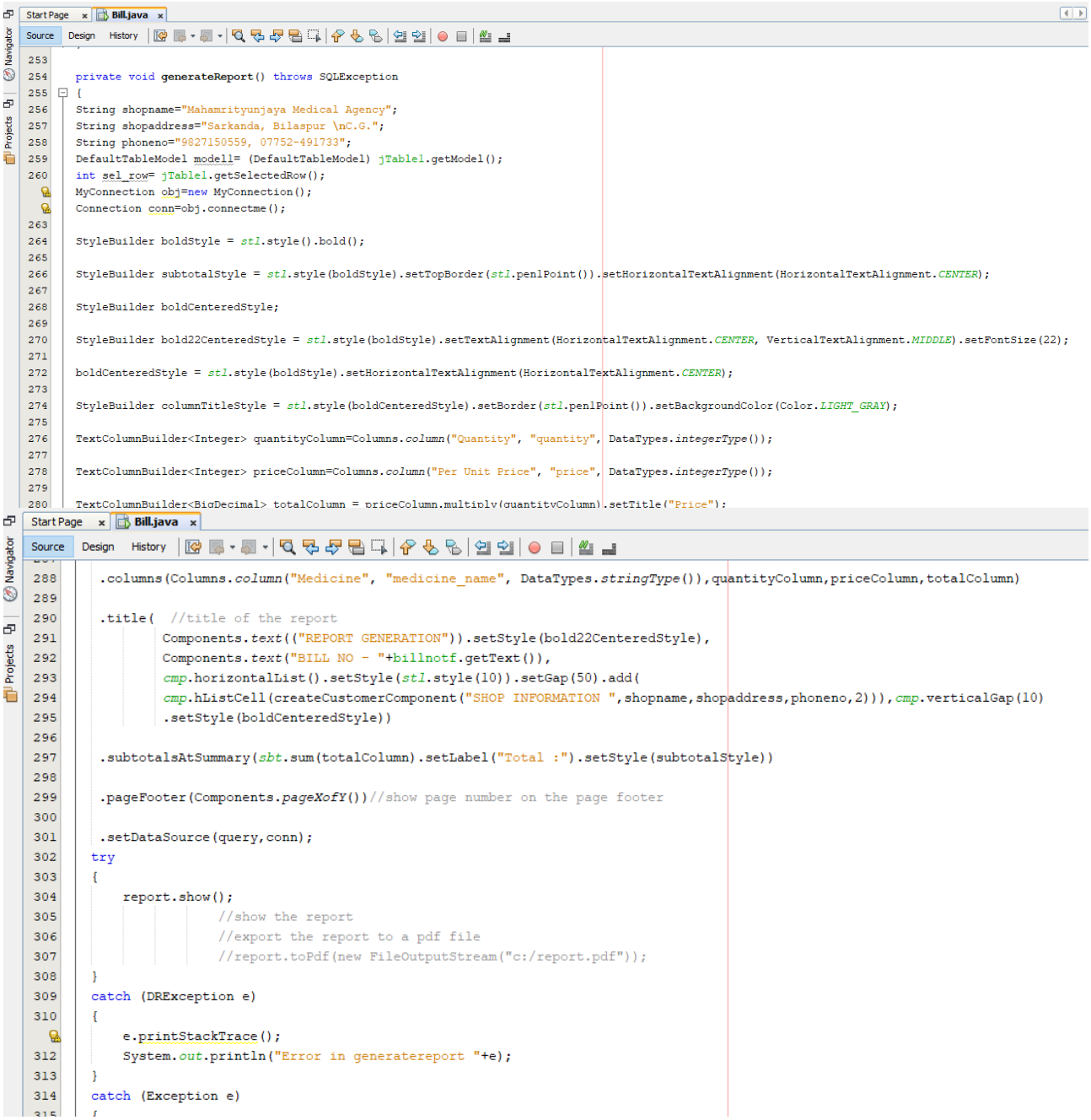
Source | Design | History

```
1  import java.awt.Color;
2  import java.math.BigDecimal;
3  import java.sql.Connection;
4  import java.sql.DriverManager;
5  import java.sql.PreparedStatement;
6  import java.sql.ResultSet;
7  import java.sql.SQLException;
8  import java.sql.Statement;
9  import javax.swing.JOptionPane;
10 import javax.swing.table.DefaultTableModel;
11 import net.sf.dynamicreports.examples.Templates;
12 import net.sf.dynamicreports.jasper.builder.JasperReportBuilder;
13 import net.sf.dynamicreports.report.builder.DynamicReports;
14 import static net.sf.dynamicreports.report.builder.DynamicReports.cmp;
15 import static net.sf.dynamicreports.report.builder.DynamicReports.sbt;
16 import static net.sf.dynamicreports.report.builder.DynamicReports.stl;
17 import net.sf.dynamicreports.report.builder.column.Columns;
18 import net.sf.dynamicreports.report.builder.column.TextColumnBuilder;
19 import net.sf.dynamicreports.report.builder.component.ComponentBuilder;
20 import net.sf.dynamicreports.report.builder.component.Components;
21 import net.sf.dynamicreports.report.builder.component.HorizontalListBuilder;
22 import net.sf.dynamicreports.report.builder.datatype.DataTypes;
23 import net.sf.dynamicreports.report.builder.style.StyleBuilder;
24 import net.sf.dynamicreports.report.constant.HorizontalTextAlignment;
25 import net.sf.dynamicreports.report.constant.VerticalTextAlignment;
26 import net.sf.dynamicreports.report.exception.DRException;
27 public class Bill extends javax.swing.JFrame {
28     ResultSet res;
```

Start Page | Bill.java

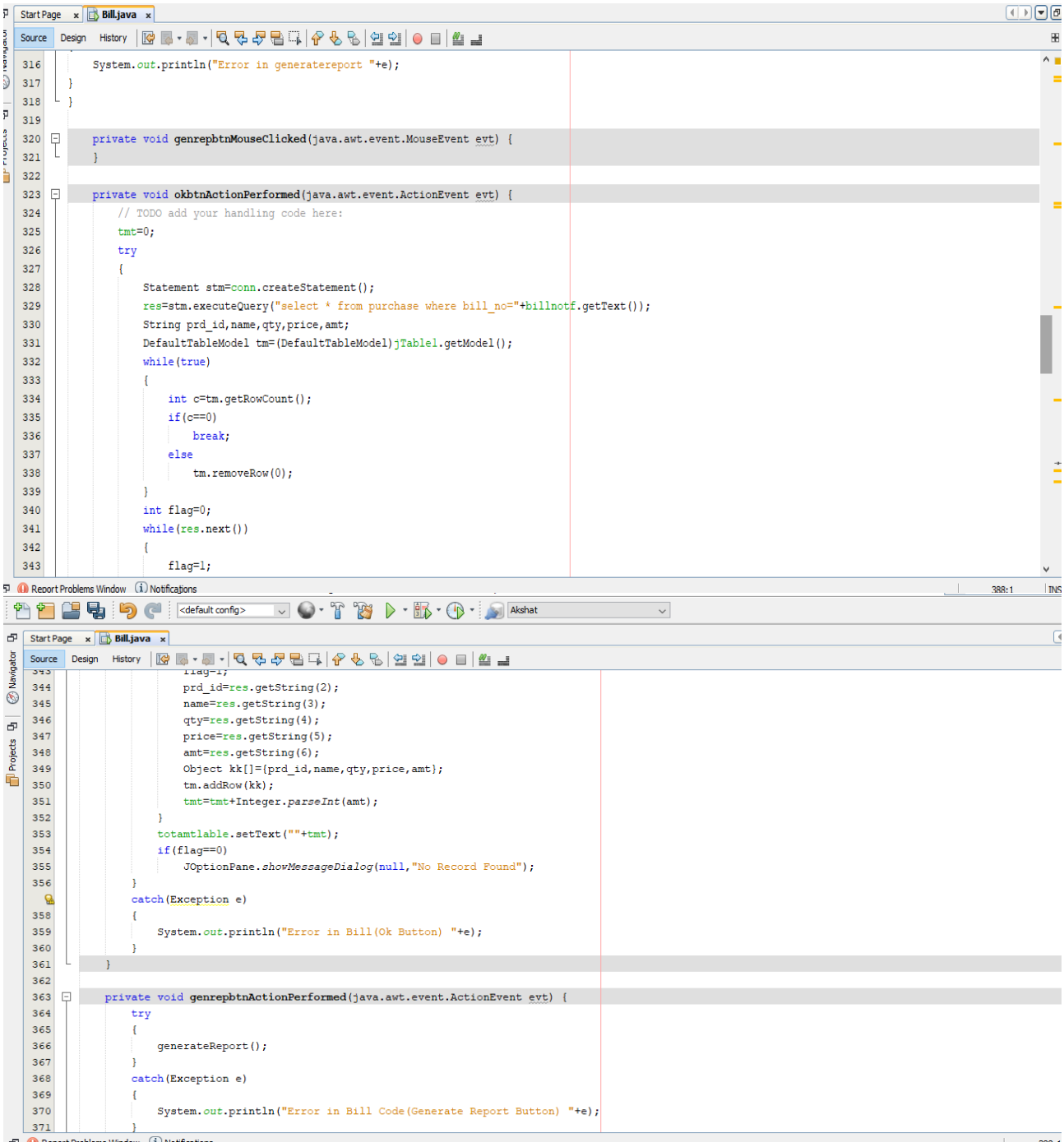
Source | Design | History

```
29     MyConnection obj=new MyConnection();
30     Connection conn=obj.connectme();
31     int tmt=0;
32     public Bill() {
33         initComponents();
34         this.setLocation(400,125);
35     }
36     @SuppressWarnings("unchecked")
37     Generated Code
233 private ComponentBuilder<?, ?> createCustomerComponent(String label, String cname, String value,String pn, Integer i)
234 {
235 HorizontalListBuilder list = cmp.horizontalList().setBaseStyle(stl.style().setTopBorder(stl.penlPoint()).setLeftPadding(10));
236
237 addCustomerAttribute(list, "Name ", cname);
238
239 addCustomerAttribute(list, "Address", value);
240
241 addCustomerAttribute(list, "Phone No.", pn);
242
243 return cmp.verticalList(cmp.text(label).setStyle(Templates.boldStyle),list);
244 }
245
246 private void addCustomerAttribute(HorizontalListBuilder list, String label, String value)
247 {
248 if (value != null)
249 {
250 list.add(cmp.text(label + ":").setFixedColumns(8).setStyle(Templates.boldStyle), cmp.text(value)).newRow();
251 }
```

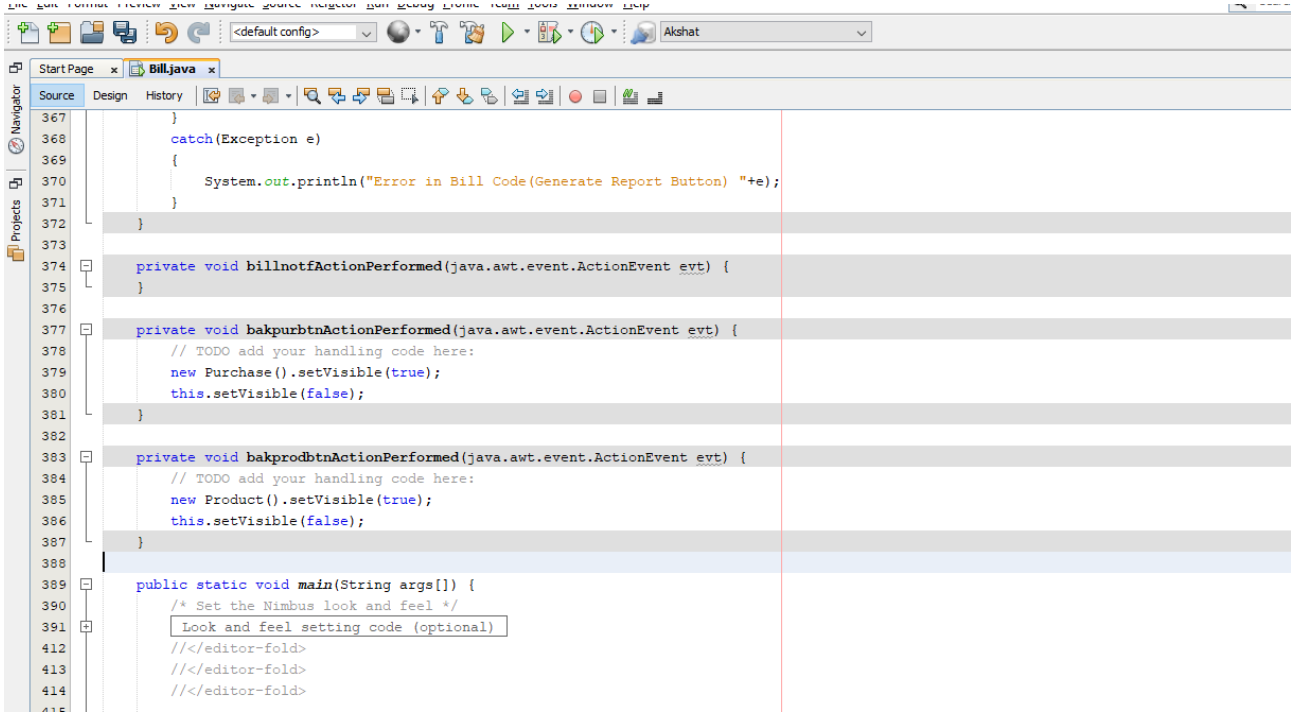


```
253
254  private void generateReport() throws SQLException
255  {
256  String shopname="Mahamrityunjaya Medical Agency";
257  String shopaddress="Sarkanda, Bilaspur \nC.G.";
258  String phoneno="9827150559, 07752-491733";
259  DefaultTableModel modell= (DefaultTableModel) jTable1.getModel();
260  int sel_row= jTable1.getSelectedRow();
     MyConnection obj=new MyConnection();
     Connection conn=obj.connectme();
263
264  StyleBuilder boldStyle = stl.style().bold();
265
266  StyleBuilder subtotalStyle = stl.style(boldStyle).setTopBorder(stl.penlPoint()).setHorizontalTextAlignment(HorizontalTextAlignment.CENTER);
267
268  StyleBuilder boldCenteredStyle;
269
270  StyleBuilder bold22CenteredStyle = stl.style(boldStyle).setTextAlignment(HorizontalTextAlignment.CENTER, VerticalTextAlignment.MIDDLE).setFontSize(22);
271
272  boldCenteredStyle = stl.style(boldStyle).setHorizontalTextAlignment(HorizontalTextAlignment.CENTER);
273
274  StyleBuilder columnTitleStyle = stl.style(boldCenteredStyle).setBorder(stl.penlPoint()).setBackgroundColor(Color.LIGHT_GRAY);
275
276  TextColumnBuilder<Integer> quantityColumn=Columns.column("Quantity", "quantity", DataTypes.integerType());
277
278  TextColumnBuilder<Integer> priceColumn=Columns.column("Per Unit Price", "price", DataTypes.integerType());
279
280  TextColumnBuilder<BigDecimal> totalColumn = priceColumn.multiply(quantityColumn).setTitle("Price");
```

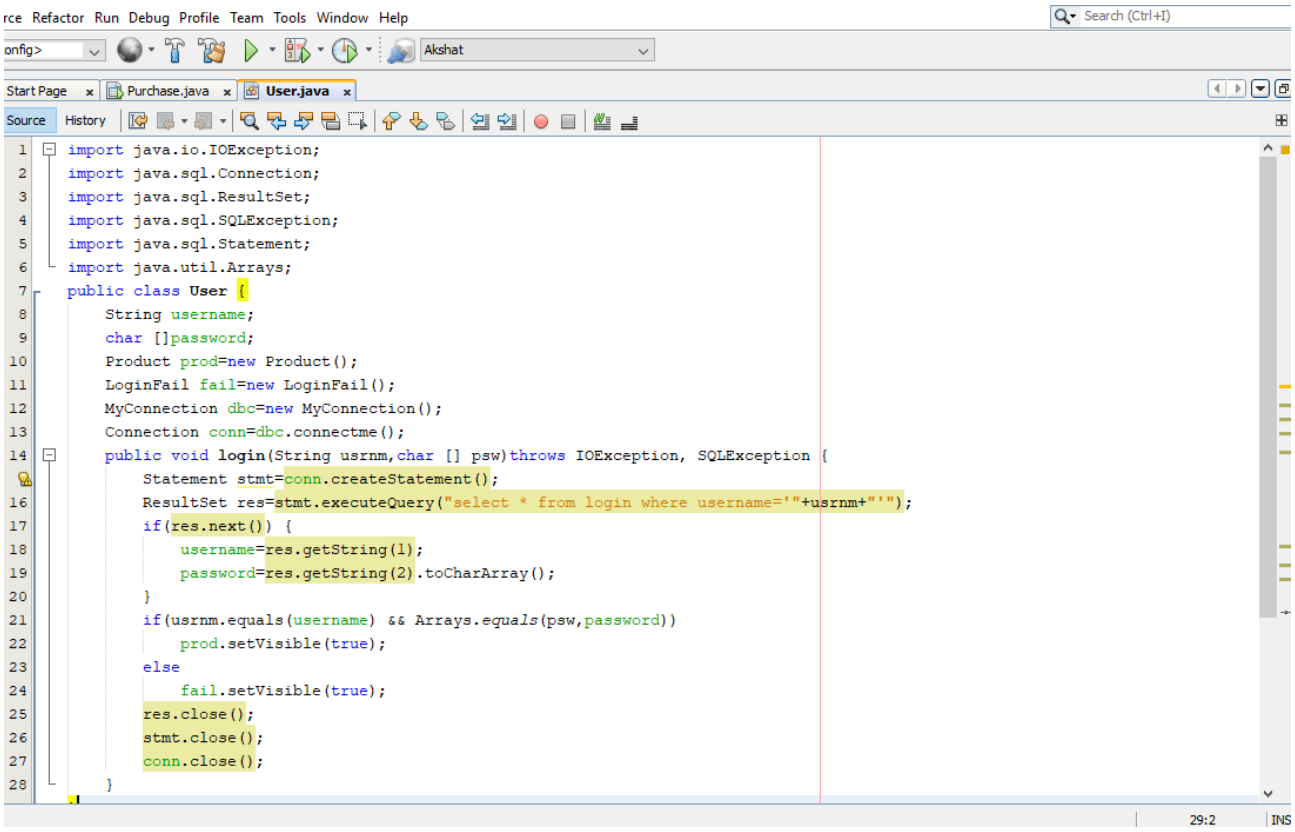
```
288    .columns(Columns.column("Medicine", "medicine_name", DataTypes.stringType()),quantityColumn,priceColumn,totalColumn)
289
290    .title(  //title of the report
291            Components.text(("REPORT GENERATION")).setStyle(bold22CenteredStyle),
292            Components.text("BILL NO - "+billnotf.getText()),
293            cmp.horizontalList().setStyle(stl.style(10)).setGap(50).add(
294            cmp.hListCell(createCustomerComponent("SHOP INFORMATION ",shopname,shopaddress,phoneno,2))),cmp.verticalGap(10)
295            .setStyle(boldCenteredStyle))
296
297    .subtotalsAtSummary(sbt.sum(totalColumn).setLabel("Total :").setStyle(subtotalStyle))
298
299    .pageFooter(Components.pageXofY())//show page number on the page footer
300
301    .setDataSource(query,conn);
302   try
303   {
304       report.show();
305                   //show the report
306                   //export the report to a pdf file
307                   //report.toPdf(new FileOutputStream("c:/report.pdf"));
308   }
309   catch (DRException e)
310   {
312       e.printStackTrace();
313       System.out.println("Error in generatereport "+e);
314   }
315   catch (Exception e)
```



```
316         System.out.println("Error in generatereport "+e);
317     }
318     }
319
320     private void genrepbtnMouseClicked(java.awt.event.MouseEvent evt) {
321     }
322
323     private void okbtnActionPerformed(java.awt.event.ActionEvent evt) {
324         // TODO add your handling code here:
325         tmt=0;
326         try
327         {
328             Statement stm=conn.createStatement();
329             res=stm.executeQuery("select * from purchase where bill_no="+billnotf.getText());
330             String prd_id,name,qty,price,amt;
331             DefaultTableModel tm=(DefaultTableModel)jTable1.getModel();
332             while(true)
333             {
334                 int c=tm.getRowCount();
335                 if(c==0)
336                     break;
337                 else
338                     tm.removeRow(0);
339             }
340             int flag=0;
341             while(res.next())
342             {
343                 flag=1;
344                 prd_id=res.getString(2);
345                 name=res.getString(3);
346                 qty=res.getString(4);
347                 price=res.getString(5);
348                 amt=res.getString(6);
349                 Object kk[]={prd_id,name,qty,price,amt};
350                 tm.addRow(kk);
351                 tmt=tmt+Integer.parseInt(amt);
352             }
353             totamtlable.setText(""+tmt);
354             if(flag==0)
355                 JOptionPane.showMessageDialog(null,"No Record Found");
356         }
358         catch(Exception e)
359         {
360             System.out.println("Error in Bill(Ok Button) "+e);
361         }
362     }
363
364     private void genrepbtnActionPerformed(java.awt.event.ActionEvent evt) {
365         try
366         {
367             generateReport();
368         }
369         catch(Exception e)
370         {
371             System.out.println("Error in Bill Code(Generate Report Button) "+e);
```


Start Page
Bill.java
Source
Design
History
Navigator
Projects
Akshat
<default config>
}
catch(Exception e)
{
System.out.println("Error in Bill Code(Generate Report Button) "+e);
}
}
private void billnotfActionPerformed(java.awt.event.ActionEvent evt) {
}
private void bakpurbtnActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
new Purchase().setVisible(true);
this.setVisible(false);
}
private void bakprodbtnActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
new Product().setVisible(true);
this.setVisible(false);
}
public static void main(String args[]) {
/* Set the Nimbus look and feel */
Look and feel setting code (optional)
//</editor-fold>
//</editor-fold>
//</editor-fold>

6. User.java


rce Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
onfig>
Akshat
Start Page
Purchase.java
User.java
Source
History
import java.io.IOException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Arrays;
public class User {
String username;
char []password;
Product prod=new Product();
LoginFail fail=new LoginFail();
MyConnection dbc=new MyConnection();
Connection conn=dbc.connectme();
public void login(String usrnm,char [] psw)throws IOException, SQLException {
Statement stmt=conn.createStatement();
ResultSet res=stmt.executeQuery("select * from login where username='"+usrnm+"'");
if(res.next()) {
username=res.getString(1);
password=res.getString(2).toCharArray();
}
if(usrnm.equals(username) && Arrays.equals(psw,password))
prod.setVisible(true);
else
fail.setVisible(true);
res.close();
stmt.close();
conn.close();
}
29:2
INS

CONCLUSION

I believe that test and implementation has shown conclusively that is both possible and desirable to use Java as the principal teaching language.

- It is free.
- It is trivial to install on a windows pc allowing students to take their interest further. Many hurdles to installing a Java compiler on a windows.
- It is a flexible tool that allows both the teaching of traditional of traditional procedural programming and modern oop; it can be used to teach many transferable skills;
- It is quicker to learn and in combination with its many liberals, this offers the possibility of more rapid student development allowing the course to made more challenging and varied.

BIBLIOGRAPHY

- https://www.w3schools.com/java/
- https://www.oracle.com/in/java/technologies/
- https://www.java.com/en/
- https://www.geeksforgeeks.org/java/
- https://www.javatpoint.com/java-tutorial
- https://www.codecademy.com/learn/learn-java
- https://www.oracle.com/in/java/