CITIZEN CARD

SYSTEM



PREFACE

As a students of Bachelor of computer Application according to the syllabus subscribed by Bhakta Kavi Narsinh Mehta University, Junagadh(Gujarat). In 5th semester of B.C.A. to prepare a project on any Particular subject of Student choice. The project should be selected on precise subject that will fulfill the requirements.

The main objective of the project is to develop Awareness regarding Application of Theories in the modern world of Information Technology.

Here the project report Training is under taken at Dr. V. R. Godhaniya Institute of Information technology in porbandar. Project report has been prepared on the base of information collected which is a result of hard work done by us.

We have taken almost care that the information provided in the project is true and project is in smooth running condition and Error free we sincerely apologies if any wrong is given we have tried our level best to provide all the necessary information as per the syllabus.

ACKNOWLEDGEMENT

It is out great pleasure to present our project report on Medical which we conceived during the BCA 5th SEM to Bhakta Kavi Narsinh Mehta University.

We take this opportunity to express our sincere gratitude and we feel immense pleasure to thank our faculty, philosopher and guider **Nirav Dattani** who helped us and gave full support in each and every way to fulfill and accomplish our project of Medical store management system.

We are kindly thank full to **Nirav Dattani** who guide us very well circumstances with his bright knowledge and also suggest us to make this project in **JAVA**.

We are mostly thankful to them, for giving us the inspiration and dedication to successfully complete this project

INDEX

|  |  |  |
| --- | --- | --- |
| Sr No. | Topic Name | Page  No. |
| 1 | Certificate |  |
| 2 | Preface | 2 |
| 3 | Acknowledgement | 3 |
| 4 | Index | 4 |
| 5 | Project Profile | 5 |
| 6 | Abstract of project(Brief system background) | 5 |
| 7 | Feasibility Study(Preliminary investigation)   * Technical * Economical * Operational | 7 |
| 8 | System Analysis and Designing   * Requirement Analysis * Use Case Diagram * Class Diagram(OOP) | 9 |
| 9 | Project Life Cycle   * E-R Diagram * Data Flow Diagram(DFD) * Activity Diagram | 14 |
| 10 | TimeLine Chart | 20 |
| 11 | User Interface Designs(SnapShots) | 22 |

PROJECT PROFILE

|  |  |
| --- | --- |
| **Project Name** | Medical Store Management System |
| **Front End** | Javafx |
| **Back End** | MYSQL |
| **Developed by** | Pooja Jungi, Mahek Sindhav,  Dhruvi Vaghani |
| **Tools** | Netbeans IDE 8.2 |
| **Testing** | Manual |
| **Task** | Analysis, Designing, Coding, Testing |
| **Guided by** | Nirav Dattani |
| **Developed at** | Dr. V. R. Godhaniya IT College Porbandar |

ABSTRACT OF PROJECT(BRIEF SYSTEM BACKGROUND)

This project is used to solve problems of Medical Stores. There are too many products in their Medical. So it is hard to manage stocks of that on Medicines in written. So we make this project to easily manage stock of Medicines. It Also generate PDF of Stock Report which tells the all information related to stock.So shopkeeper can check data any time.

There is one module in this project:

* Stock management :

There is a huge Database of all product. So there are four sub-module:

1. Insert Module
2. Delete Module
3. Update Module
4. View Module

1. Insert Module:

There is one form which provides facility to insert new product or new data into database.

2. Delete Module :

This provides the facility to remove product information from system or from database.

3. Update Module :

This Module the provides the facility to edit existing product into database or system .

4. View Module :

This Module the provides the facility to view all data of medicine.

**Name of Organization** : Shivam Medical Store

**Project Time Duration** : 15 July, 2018 to 1 October, 2018

**Prepared By** : Pooja Jungi BCA 5th Sem, DIV – A, ROLL No – 56

Mahek Sindhav BCA 5th Sem, DIV – A, ROLL No – 152

Dhruvi Vaghani BCA 5th Sem, DIV – A, ROLL No – 174

**Summary :**

Main purpose of making this project that reduces bookwork of shopkeeper. Using this project shopkeeper can easily maintain the all stocks and not need to do any type of bookwork, all things are maintain in this System. This is very easy way to manage a medical.

PRELIMINARY INVASTIGATION

Analysis is conducted with some objectives in mind, such

that identify the customer’s needs, evaluate the system concepts

for feasibility, perform economic and technical analysis, allocate function to hardware and software. People-database and other system elements, established cost and schedule constraints and create system definition that form the foundation for all subsequent engineering work.

We have been analyze about the Medical Store to

examine how the work of case is being carried out and who

are the persons directly associated with the system.

We have seen the Medical Store environment. Work load of the system and the shopkeeper and methods of work.

* Feasibility Studies:

**1.Technical Feasibility Study**

A study of resource availability that my effect the

ability to achieve an acceptable system. This evaluation

determines whether the technology needed for the

proposed system is available or not.

2.**Economical Feasibility Study**

Economic analysis is the most frequently used

method for evaluating the effectiveness of a proposed

system. In this feasibility we decide the proposed

system as on the basis of cost, benefit analysis.

Whether the proposed system is feasible according to

the economic factors

The benefit that this system has are many

including the working speed of the system queries are

sorted in a faster manner, the data are stored in the

database that can be used whenever required

3. **Operational Feasibility Study**

The assessment made id the system will be used or not

once it is developed. According to the project the material

sharing application will use. The operational feasibility if the

system is developed and implemented to measure. How well

our project solution is will work in the organization.

SYSTEM ANALYSIS AND DESIGNING

* Requiremental Analysis
* **Functional Requirement** :

The software must allow input of product data

from administrator & secured access at, and from

the data streaming real-time monitoring equipment.

* **Non-Functional Requirement** :

The software interface must follow design

conventions which allow for familiar

location of menus etc.

Input errors will be returned in red with

appropriate message box.

* **User Requirement**:

The user of system are staff manager.

The use manual installation guide and other

related material should be sufficient to educate

the user how to use and maintains the system.

* **Software Requirement:**

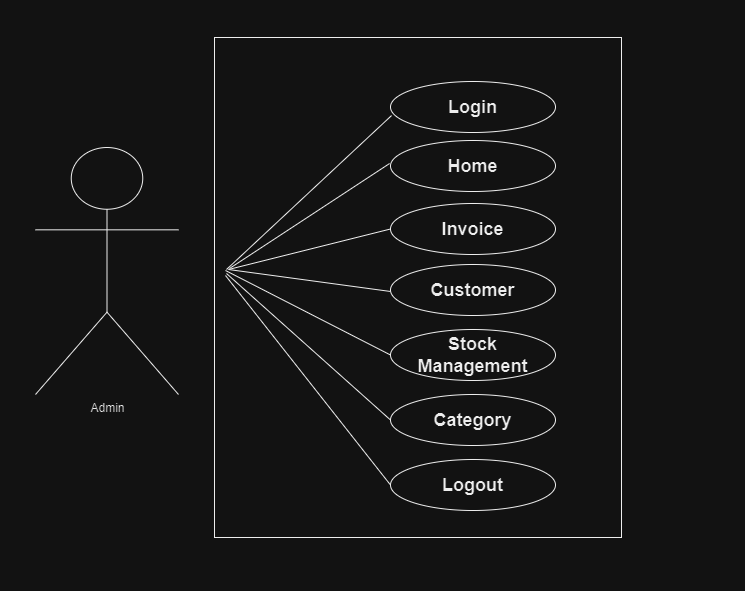
Netbeans IDE 8.2(Wamp Server, Mysql Database)

* **Hardware Requirement:**

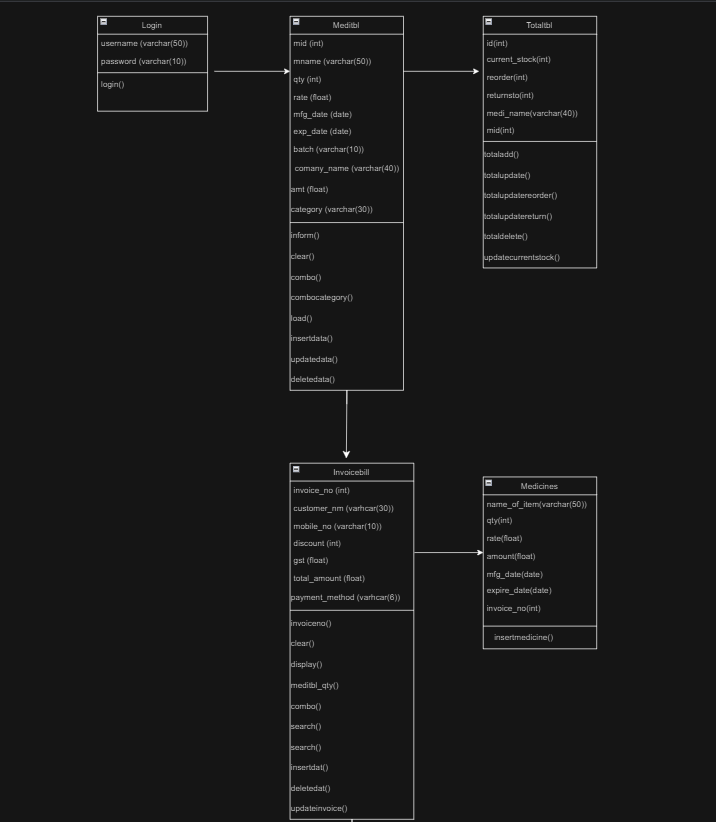
Minimum: 6 GHZ CPU or faster processor, 1 GB

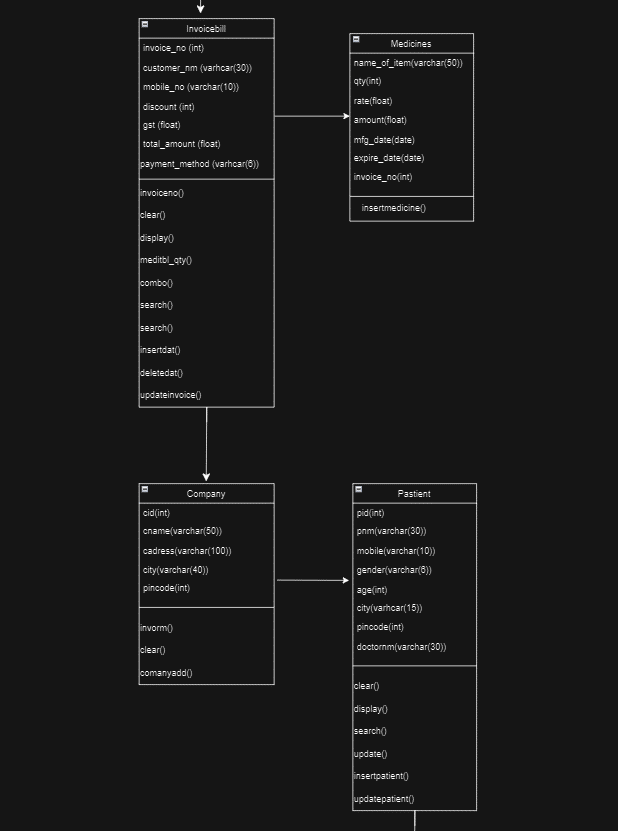
(32 Bit) or 2 GB (64 Bit) RAM

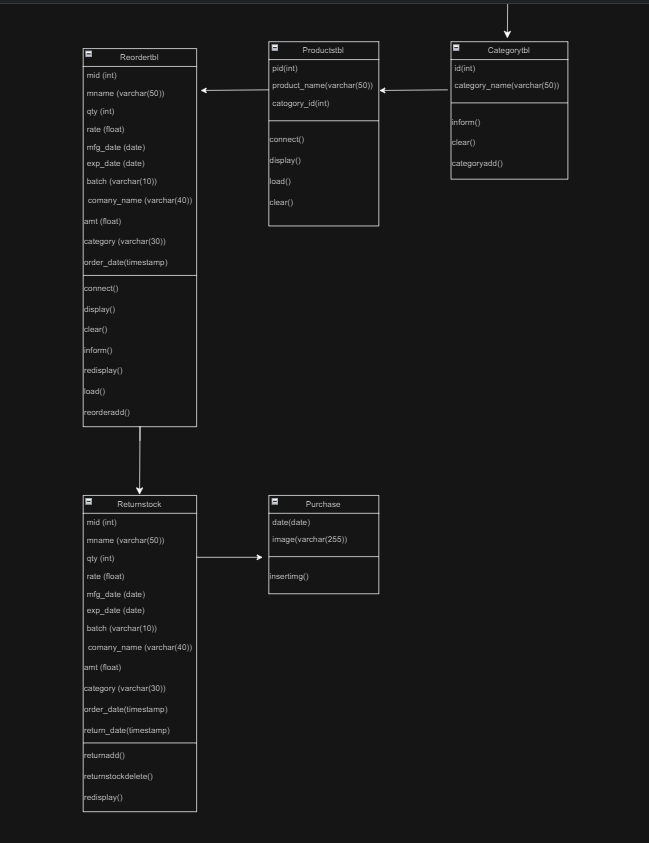
* Use Case Diagram



* Class Diagram(OOP)

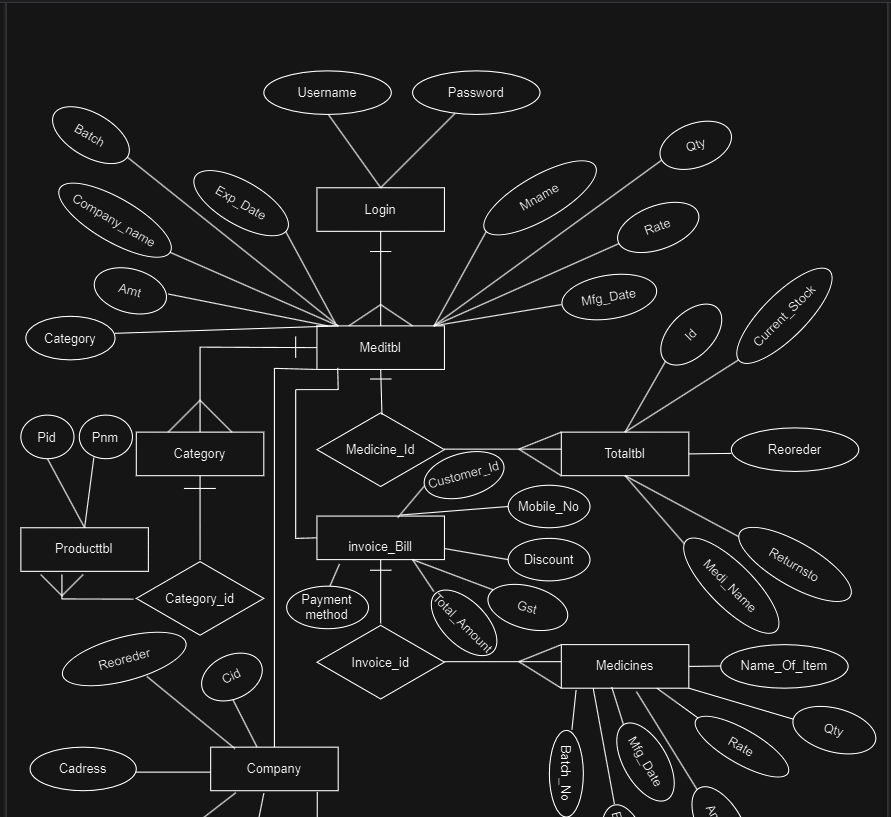


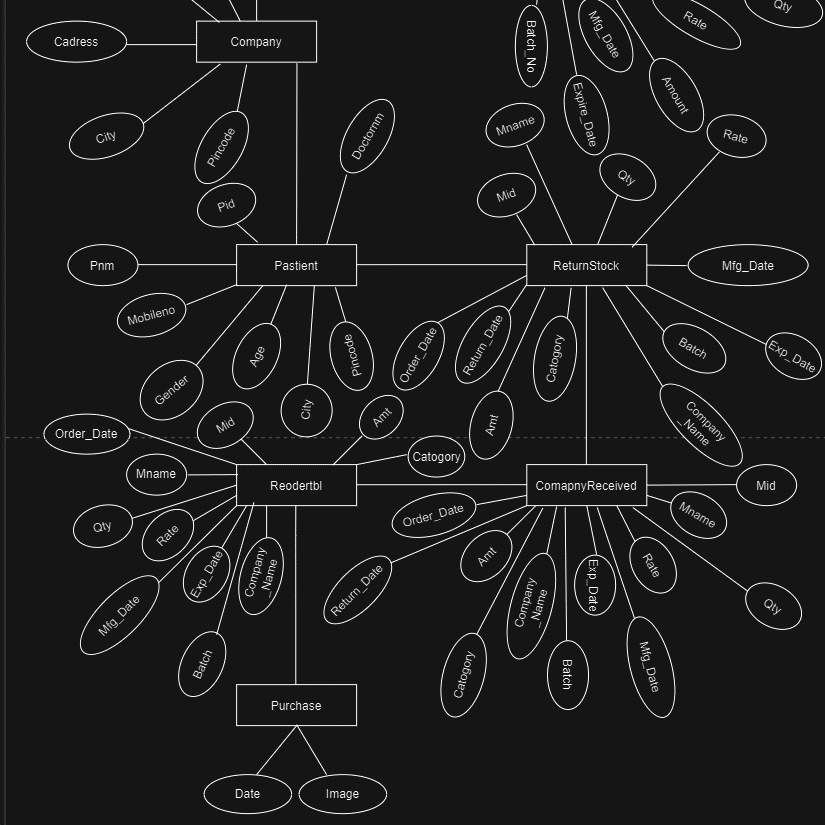




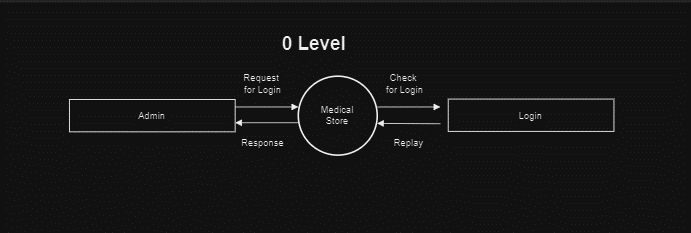
PROJECT LIFE CYCLE

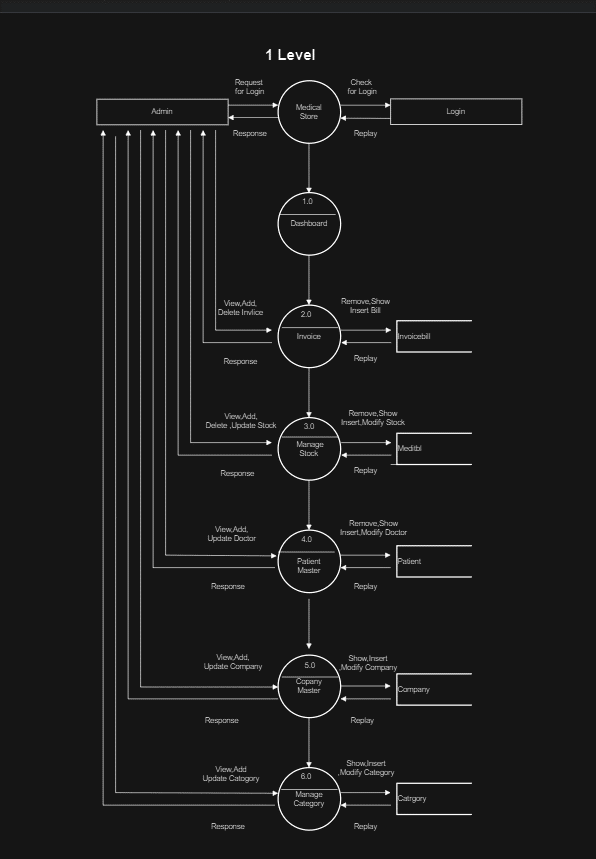
* E-R Diagram



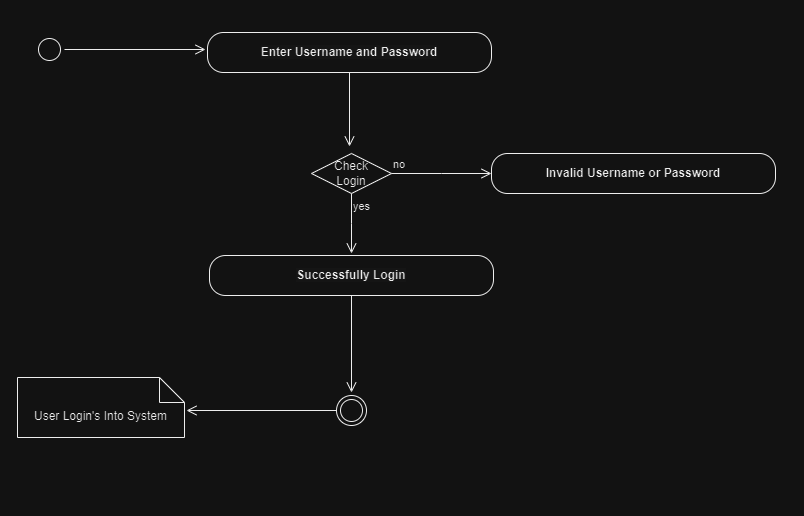


* Data Flow Diagram(DFD)

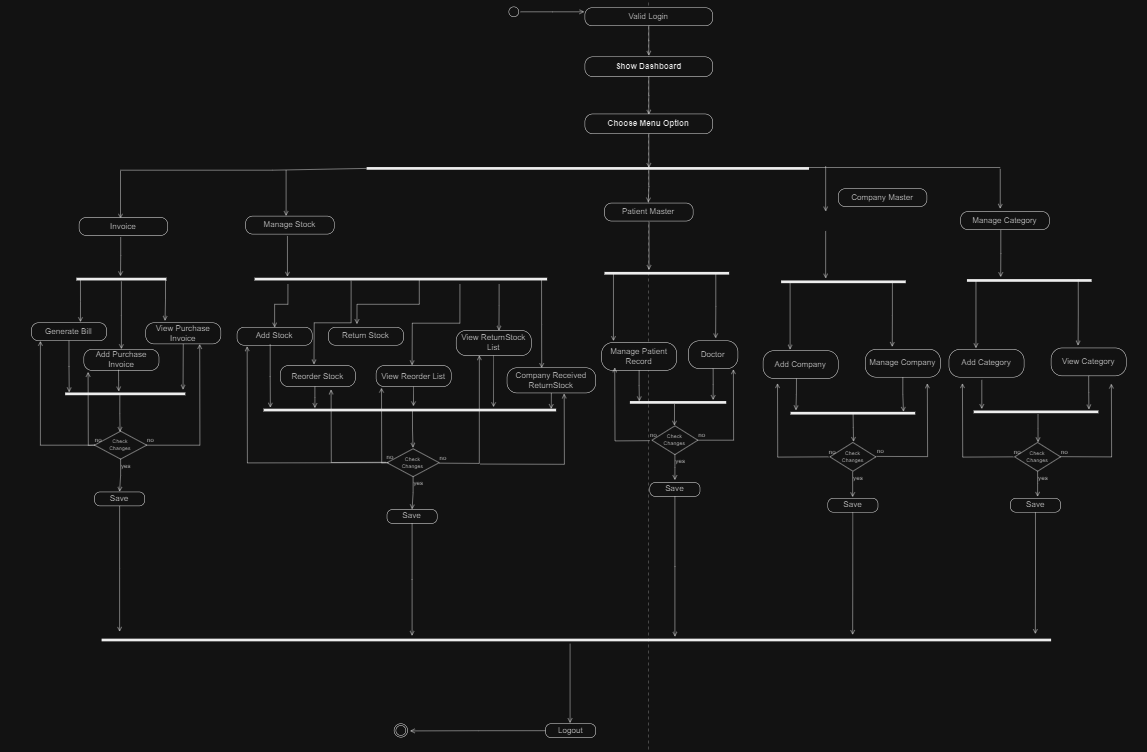




* Activity Diagram
* Login Activity Diagram



* Main System Activity Diagram



* Timeline Chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Activity | Start Date | End Date | Duration |
| 1 | Project Proposal | 11-07-2024 | 11-07-2024 | 1 |
| 2 | Requirement Analysis | 12-07-2024 | 14-07-2024 | 3 |
| 3 | Class Diagram | 15-07-2024 | 17-04-2024 | 3 |
| 4 | E.R.Diagram | 18-07-2024 | 19-07-2024 | 2 |
| 5 | Charts-DFD | 20-07-2024 | 22-07-2024 | 3 |
| 6 | Use Case Diagram | 23-07-2024 | 23-07-2024 | 1 |
| 7 | Activity Diagram | 24-07-2024 | 25-07-2024 | 2 |
| 8 | Coding | 26-07-2024 | 22-09-2024 | 58 |

* User Interface Designs(Snapshots)
* **LoginForm.fxml :**

****

* **LoginController.java :**

package projectfx;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.PasswordField;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import javax.swing.JOptionPane;

public class LoginController implements Initializable {

@FXML

private AnchorPane loginbody;

@FXML

private TextField uname;

@FXML

private PasswordField upassw;

@FXML

private Button btn1;

crud cr = new crud();

@FXML

private TextField textpass;

static int temp = 1;

@FXML

private ImageView img;

@FXML

private ImageView img2;

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

}

@FXML

private void buttonpress(ActionEvent event) {

try {

String un = String.valueOf(uname.getText());

String ps = String.valueOf(upassw.getText());

if (cr.log(un, ps) == true) {

JOptionPane.showMessageDialog(null, "login success", "Login Messages", JOptionPane.INFORMATION\_MESSAGE);

loginbody.getScene().getWindow().hide(); // Hide login window

Stage primaryStage = new Stage();

Parent root = FXMLLoader.load(getClass().getResource("Dashboard.fxml"));

Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("dashboard\_design.css").toExternalForm());

primaryStage.setScene(scene);

primaryStage.setMaximized(true);

primaryStage.show();

} else {

JOptionPane.showMessageDialog(null, "try again", "Login Messages", JOptionPane.WARNING\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "error : " + e, "Login Messages", JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void mouseclick(MouseEvent event) {

img2.setVisible(false);

img.setVisible(true);

upassw.setVisible(false);

textpass.setVisible(true);

textpass.setText(upassw.getText());

}

@FXML

private void mclick(MouseEvent event) {

img2.setVisible(true);

img.setVisible(false);

upassw.setVisible(true);

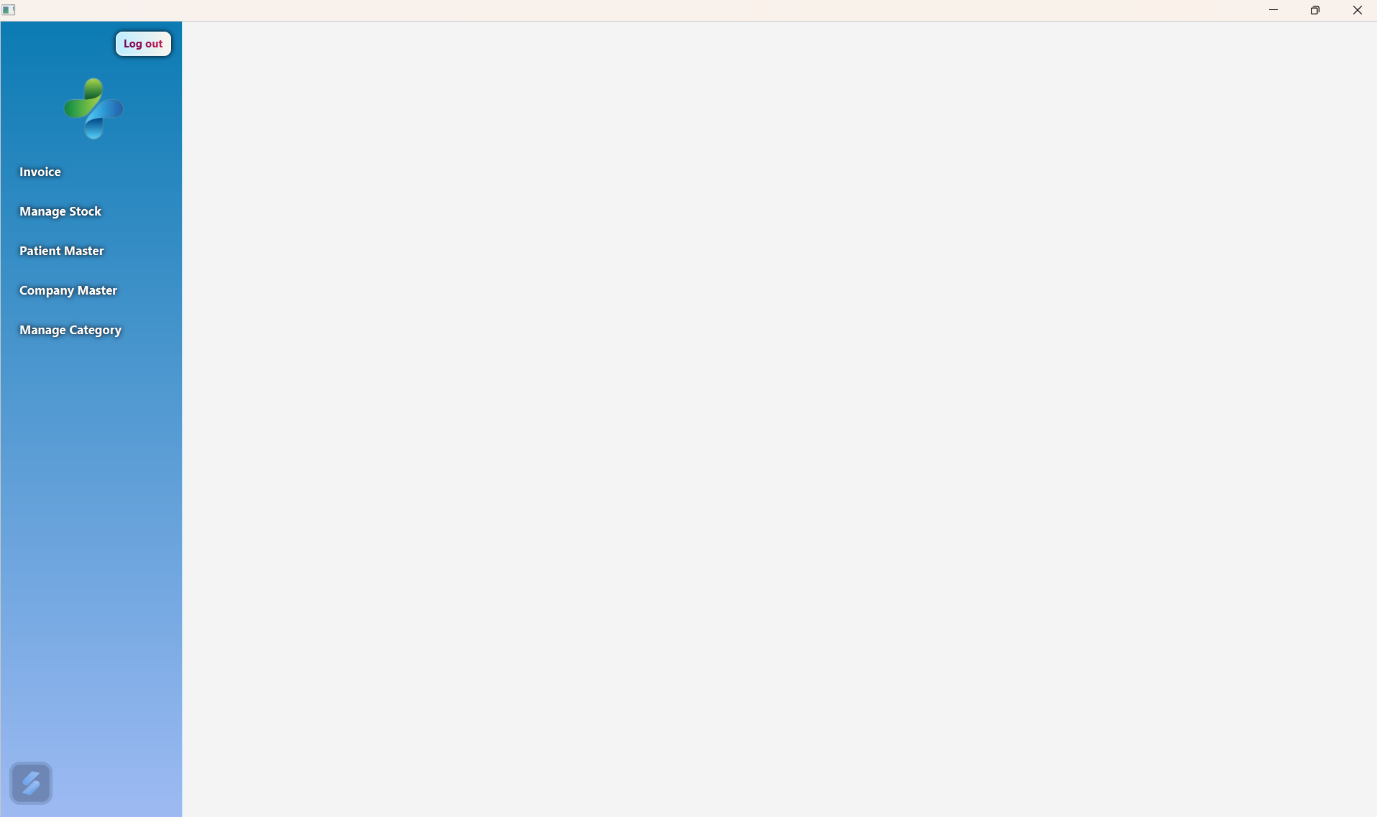
textpass.setVisible(false);

upassw.setText(textpass.getText());

}

}

* **Dashboard.fxml :**

****

* **DashboardController.java :**

package projectfx;

import java.awt.Frame;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.HashMap;

import java.util.Map;

import java.util.ResourceBundle;

import javafx.embed.swing.SwingNode;

import javafx.scene.control.MenuButton;

import javafx.scene.control.MenuItem;

import javafx.scene.control.ScrollPane;

import javafx.scene.control.SplitPane;

import javafx.scene.image.ImageView;

import javafx.scene.layout.AnchorPane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import javax.swing.JOptionPane;

import net.sf.jasperreports.engine.JRBand;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import static net.sf.jasperreports.engine.JasperPrintManager.printReport;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.type.SplitTypeEnum;

import net.sf.jasperreports.engine.util.JRLoader;

import net.sf.jasperreports.swing.JRViewer;

import org.apache.xerces.parsers.SAXParser;

public class DashboardController extends Frame implements Initializable {

@FXML

public SplitPane split;

@FXML

private AnchorPane pane1;

@FXML

private MenuButton menu111;

@FXML

private MenuButton menu11111;

@FXML

private MenuButton menu11112;

AnchorPane ap = new AnchorPane();

@FXML

public ScrollPane pn2;

String unm;

crud cr = new crud();

Connection conn = cr.connect();

private static JasperReport jr;

private static JasperPrint jp;

@Override

public void initialize(URL url, ResourceBundle rb) {

pn2.prefHeightProperty().bind(split.heightProperty());

}

@FXML

private void cateClick(ActionEvent event) {

try {

ap.getChildren().clear();//clear

FXMLLoader loader = new FXMLLoader(getClass().getResource("category.fxml"));

ap = loader.load();

ap.getStylesheets().add(getClass().getResource("category\_design.css").toExternalForm());

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

public void logOutvalue(){

try {

Class.forName("com.mysql.jdbc.Driver");

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb","root","");

PreparedStatement pst = conn.prepareStatement("select username from user");

ResultSet rs = pst.executeQuery();

while (rs.next()) {

unm = rs.getString("username");

}

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void viewCateClick(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("viewcategory.fxml"));

ap = loader.load();

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void AddStockClick(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("medicines.fxml"));

ap = loader.load();

ap.getStylesheets().add(getClass().getResource("medi\_design.css").toExternalForm());

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void reorderclick(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("stock.fxml"));

ap = loader.load();

ap.getStylesheets().add(getClass().getResource("stock\_design.css").toExternalForm());

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void ReturnClick(ActionEvent event) {

returnclic();

}

public void returnclic()

{

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("returnStock.fxml"));

ap = loader.load();

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void ViewReorderClick(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("vieworder.fxml"));

ap = loader.load();

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void ViewReturnStockClick(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("viewReturn.fxml"));

ap = loader.load();

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void CompanyReceived(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("companyReturn.fxml"));

ap = loader.load();

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void companyClick(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("company.fxml"));

ap = loader.load();

ap.getStylesheets().add(getClass().getResource("category\_design.css").toExternalForm());

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void manageCompnay(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("viewCompany.fxml"));

ap = loader.load();

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void invoice(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("invoicefx.fxml"));

ap = loader.load();

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void purchase(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("purchase.fxml"));

ap = loader.load();

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void view(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("displayinvoice.fxml"));

ap = loader.load();

ap.getStylesheets().add(getClass().getResource("purchasecss.css").toExternalForm());

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void details(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("displaymedicines.fxml"));

ap = loader.load();

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void doctorForm(ActionEvent event) {

try {

ap.getChildren().clear();

FXMLLoader loader = new FXMLLoader(getClass().getResource("patient.fxml"));

ap = loader.load();

ap.getStylesheets().add(getClass().getResource("purchasecss.css").toExternalForm());

pn2.setContent(ap);

ap.prefHeightProperty().bind(pn2.heightProperty());

} catch (Exception e) {

System.out.println(e);

}

}

@FXML

private void logOutClick(ActionEvent event) {

try {

logOutvalue();

crud cr = new crud();

if (cr.logout(unm)>0) {

split.getScene().getWindow().hide();

FXMLLoader loader = new FXMLLoader(getClass().getResource("login.fxml"));

Parent root = loader.load();

Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("design.css").toExternalForm());

Stage primaryStage = new Stage();

primaryStage.setScene(scene);

primaryStage.show();

} else {

JOptionPane.showMessageDialog(null,"Try Again.","LogOut Message",JOptionPane.WARNING\_MESSAGE);

}

} catch (Exception e) {

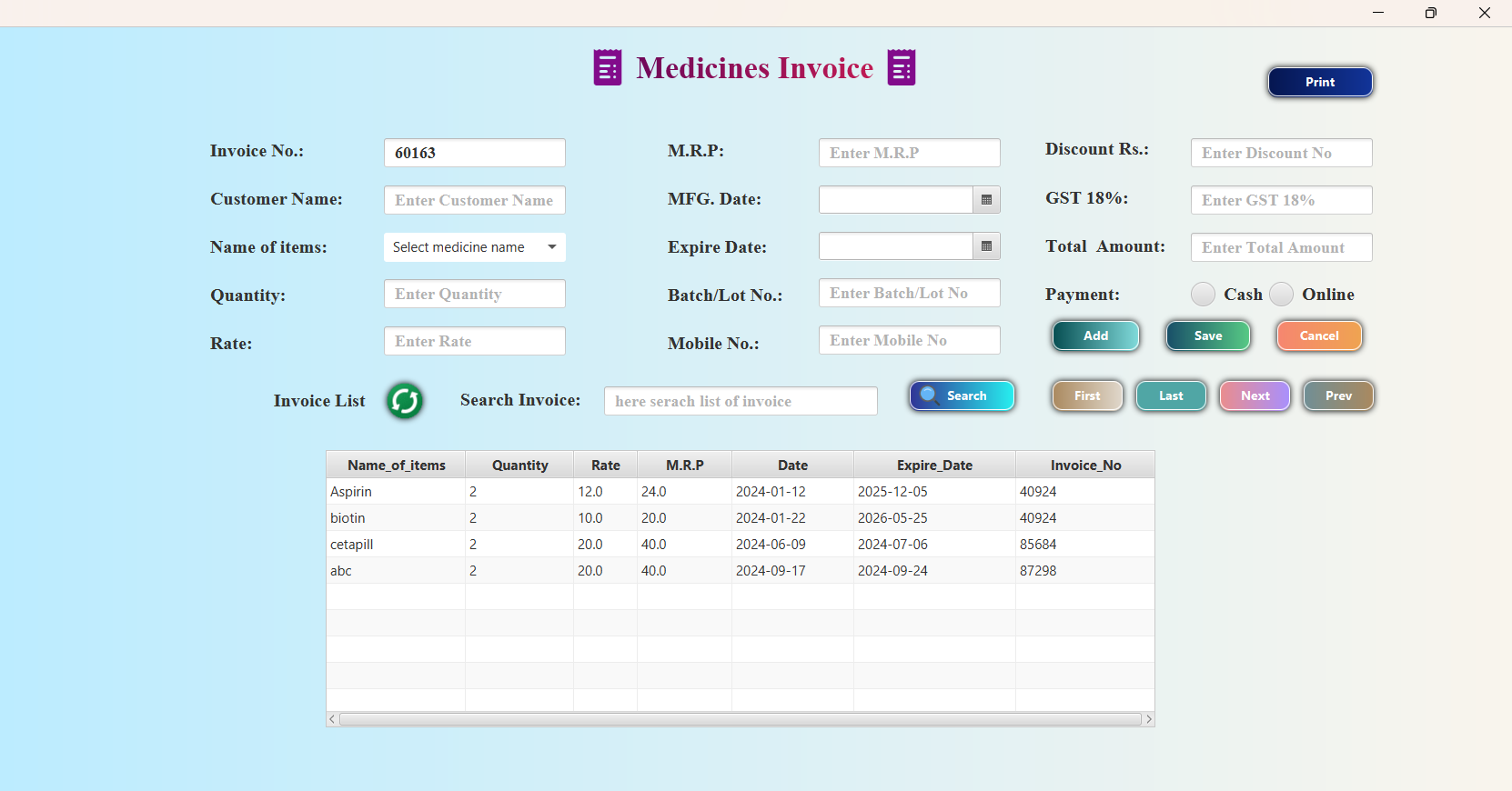
System.out.println(e);

}

}

}

* **invoice.fxml :**



* **invoicefxController.java :**

package projectfx;

import java.awt.Frame;

import java.net.URL;

import java.sql.Connection;

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.HashMap;

import java.util.Map;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.DatePicker;

import javafx.scene.control.RadioButton;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import net.sf.jasperreports.engine.JRException;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.swing.JRViewer;

import java.util.Random;

import java.math.\*;

import java.time.Month;

import java.util.Observable;

import javafx.embed.swing.SwingNode;

import javafx.scene.control.ComboBox;

import javafx.scene.layout.VBox;

/\*\*

\* FXML Controller class

\*

\* @author mahek

\*/

public class InvoicefxController implements Initializable {

@FXML

private TextField txtinvoiceno;

@FXML

private TextField txtcustomernm;

private TextField txtnameofitem;

@FXML

private TextField txtquantity;

@FXML

private TextField txtrate;

@FXML

private TextField txtamount;

@FXML

private DatePicker mfgdate;

@FXML

private DatePicker expiredate;

@FXML

private TextField txtbatchno;

@FXML

private TextField txtmobileno;

@FXML

private TextField txtdiscountno;

@FXML

private TextField txtgst;

@FXML

private TextField txttotalamount;

@FXML

private Button btnfirst;

@FXML

private Button btnlast;

@FXML

private Button btnnext;

@FXML

private Button btnprev;

/\*\*

\* Initializes the controller class.

\*/

static int no;

Connection c;

int i, q, dis;

Date d, ed;

String cm, im, b, m, p;

static float total\_amount = 0;

private static JasperReport jr;

private static JasperPrint jp;

static String NO;

float total, r, a, g;

@FXML

private TextField txtsearch;

@FXML

private Button btnsearch;

@FXML

private Button btncancel;

@FXML

private ImageView ref;

@FXML

private Button btnadd;

@FXML

private Button btnsave1;

@FXML

private AnchorPane invoice\_anchorpane;

@FXML

private TableView<medicine\_variable> tableview;

@FXML

private TableColumn<medicine\_variable, String> nm;

@FXML

private TableColumn<medicine\_variable, Integer> qty;

@FXML

private TableColumn<medicine\_variable, Float> rate;

@FXML

private TableColumn<medicine\_variable, Float> amount;

@FXML

private TableColumn<medicine\_variable, Date> date;

@FXML

private TableColumn<medicine\_variable, Date> expiredate1;

@FXML

private TableColumn<medicine\_variable, Integer> ino;

@FXML

private RadioButton p\_case;

@FXML

private ToggleGroup pay;

@FXML

private RadioButton p\_online;

@FXML

private Button print;

@FXML

private ComboBox<String> combo;

static String combo\_value="";

int q\_t;

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

getdatelist();

display();

invoiceno();

txtinvoiceno.setText(NO);

combo();

// meditbl\_qty();

}

public void invoiceno() {

String[] number = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"};

Random random = new Random();

int randomindex1 = random.nextInt(number.length);

String randomelement1 = number[randomindex1];

int randomindex2 = random.nextInt(number.length);

String randomelement2 = number[randomindex2];

int randomindex3 = random.nextInt(number.length);

String randomelement3 = number[randomindex3];

int randomindex4 = random.nextInt(number.length);

String randomelement4 = number[randomindex4];

int randomindex5 = random.nextInt(number.length);

String randomelement5 = number[randomindex5];

NO = randomelement1 + randomelement2 + randomelement3 + randomelement4 + randomelement5;

// System.out.println(NO);

//

// double n1 =number[Math.floor(Math.random())\*62];

}

public void clear() {

txtinvoiceno.setText(null);

txtinvoiceno.setEditable(true);

txtcustomernm.setText(null);

txtcustomernm.setEditable(true);

txtnameofitem.setText(null);

combo.setValue(null);

txtquantity.setText(null);

txtrate.setText(null);

txtamount.setText(null);

mfgdate.setValue(null);

expiredate.setValue(null);

txtbatchno.setText(null);

txtmobileno.setText(null);

txtmobileno.setEditable(true);

txtdiscountno.setText(null);

txtgst.setText(null);

txttotalamount.setText(null);

total\_amount = 0;

combo.setValue(null);

p\_case.setSelected(false);

p\_online.setSelected(false);

invoiceno();

txtinvoiceno.setText(NO);

}

//display tableview

ObservableList<medicine\_variable> getdatelist() {

ObservableList<medicine\_variable> medicineslist = FXCollections.observableArrayList();

try {

int i = InvoicefxController.no;

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

PreparedStatement p = c.prepareStatement("select \*from medicines");

ResultSet rs = p.executeQuery();

medicine\_variable m;

while (rs.next()) {

m = new medicine\_variable(rs.getString("name\_of\_item"), rs.getInt("qty"), rs.getFloat("rate"), rs.getFloat("amount"), rs.getDate("mfg\_date"), rs.getDate("Expire\_date"), rs.getInt("invoice\_no"));

medicineslist.add(m);

}

} catch (Exception e) {

System.out.println(e);

}

return medicineslist;

}

public void display() {

try {

ObservableList<medicine\_variable> list = getdatelist();

nm.setCellValueFactory(new PropertyValueFactory<medicine\_variable, String>("name\_of\_item"));

qty.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Integer>("qty"));

rate.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Float>("rate"));

amount.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Float>("amount"));

date.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Date>("date"));

expiredate1.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Date>("edate"));

ino.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Integer>("invoice\_no"));

tableview.setItems(list);

} catch (Exception e) {

System.out.println(e + "not display");

}

}

// count the qty and set the current stock

public void meditbl\_qty()

{

String co=combo.getValue();

try {

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

PreparedStatement p = c.prepareStatement("select qty from meditbl where mname like '%"+co+"%'");

ResultSet result = p.executeQuery();

while(result.next())

{

q\_t=result.getInt("qty");

}

int current\_q=Integer.parseInt(txtquantity.getText());

int current\_qty=q\_t-current\_q;

// System.out.println(current\_qty);

iobj o=new iobj();

crud cr = new crud();

if( o.updatecurrnetstock(current\_qty, co)>0){

if (cr.UpdateStockMedicine(current\_qty, co)>0) {

System.out.println("update successfully");

}

}

else

System.out.println("not update");

} catch (Exception e) {

System.out.println(e+" not working");

}

}

//display for comboitems

ObservableList<String> medicinename = FXCollections.observableArrayList();

public void combo() {

try {

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

PreparedStatement p = c.prepareStatement("select mname from meditbl " );

ResultSet rs = p.executeQuery();

// String ms;

while (rs.next()) {

medicinename.add(rs.getString("mname"));

}

ObservableList<String> list = FXCollections.observableArrayList(medicinename);

combo.setItems(list);

} catch (Exception e) {

System.out.println(e);

}

}

ObservableList<medicine\_variable> invoicesearch = FXCollections.observableArrayList();

public void search() {

String se = txtsearch.getText();

try {

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

PreparedStatement p = null;

if (Character.isAlphabetic(se.charAt(0)) || Character.isAlphabetic(4)) {

p = c.prepareStatement("select \*from medicines where name\_of\_item like '%" + se + "%'or batch\_no like'%" + se + "%'");

// or nmi like '%" + se + "%' or blno like '%" + se + "%'");

} else if (Character.isDigit(se.charAt(0))) {

p = c.prepareStatement("select \*from medicines where invoice\_no= " + se);

// + "or qty="+se+" or rate="+se+" or amo="+se+"or mno="+se+" or drs="+se+" or gst="+se+" or tam="+se);

}

ResultSet rs = p.executeQuery();

medicine\_variable m;

while (rs.next()) {

m = new medicine\_variable(rs.getString("name\_of\_item"), rs.getInt("qty"), rs.getFloat("rate"), rs.getFloat("amount"), rs.getDate("mfg\_date"), rs.getDate("Expire\_date"), rs.getInt("invoice\_no"));

invoicesearch.add(m);

}

nm.setCellValueFactory(new PropertyValueFactory<medicine\_variable, String>("name\_of\_item"));

qty.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Integer>("qty"));

rate.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Float>("rate"));

amount.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Float>("amount"));

date.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Date>("date"));

expiredate1.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Date>("edate"));

ino.setCellValueFactory(new PropertyValueFactory<medicine\_variable, Integer>("invoice\_no"));

tableview.setItems(invoicesearch);

} catch (Exception e) {

System.out.println(e);

}

}

//insert code

@FXML

private void btnsave(ActionEvent event) {

i = Integer.parseInt(txtinvoiceno.getText());

cm = txtcustomernm.getText();

im = combo.getValue();

q = Integer.parseInt(txtquantity.getText());

r = Float.parseFloat(txtrate.getText());

a = Float.parseFloat(txtamount.getText());

d = Date.valueOf(mfgdate.getValue());

ed = Date.valueOf(expiredate.getValue());

b = txtbatchno.getText();

m = txtmobileno.getText();

dis = Integer.parseInt(txtdiscountno.getText());

g = Float.parseFloat(txtgst.getText());

total = Float.parseFloat(txttotalamount.getText());

if (p\_case.isSelected()) {

p = p\_case.getText();

} else {

p = p\_online.getText();

}

iobj o = new iobj();

if (txtcustomernm.isEditable()) {

try {

total\_amount = total\_amount + a;

// System.out.println(total\_amount);

if (o.insertdat(i, cm, m, dis, g, total, p) > 0) {

meditbl\_qty();

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, "Generate Bill", "Invoce", JOptionPane.INFORMATION\_MESSAGE);

} else {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, "Somethink wrong", "Try again", JOptionPane.ERROR\_MESSAGE);

}

if (o.insertmedicine(im, q, r, a, d, ed, b, i) > 0) {

System.out.println("insert record in medicines table ");

display();

} else {

System.out.println("not inserted");

}

} catch (Exception e) {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, e, "try", JOptionPane.ERROR\_MESSAGE);

}

} else {

total\_amount = total\_amount + a;

String a\_gst = String.valueOf(total\_amount \* 18 / 100);

//System.out.println(a\_gst);

txtgst.setText(null);

txtgst.setText(a\_gst);

int discount = Integer.parseInt(txtdiscountno.getText());

float dc = total\_amount \* discount / 100;

float a\_g = Float.parseFloat(txtgst.getText());

// System.out.println(a\_g);

String t\_amount = String.valueOf(total\_amount + a\_g - dc);

txttotalamount.setText(t\_amount);

//System.out.println(t\_amount);

float tm = Float.parseFloat(txttotalamount.getText());

if (o.insertmedicine(im, q, r, a, d, ed, b, i) > 0) {

meditbl\_qty();

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, "Generate Bill", "Invoce", JOptionPane.INFORMATION\_MESSAGE);

display();

} else {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, "Somethink wrong", "Try again", JOptionPane.ERROR\_MESSAGE);

}

if (o.updateinvoice(a\_g, tm, i) > 0) {

System.out.println("update record");

} else {

System.out.println("try again");

}

}

}

@FXML

private void txtratekeyreleased(KeyEvent event) {

int qt = Integer.parseInt(txtquantity.getText());

float rt = Float.parseFloat(txtrate.getText());

String answer = String.valueOf(qt \* rt);

txtamount.setText(answer);

}

@FXML

private void txtdiscountkeyreleased(KeyEvent event) {

a = Float.parseFloat(txtamount.getText());

int disa = Integer.parseInt(txtdiscountno.getText());

float d = a \* disa / 100;

String gstc = String.valueOf(a \* 18 / 100);

txtgst.setText(gstc);

float tc = Float.parseFloat(txtgst.getText());

String total = String.valueOf(a + tc - d);

txttotalamount.setText(total);

}

@FXML

private void btnsearch(ActionEvent event) {

search();

}

private void btnref(ActionEvent event) {

display();

}

@FXML

private void btnfirstt(ActionEvent event) {

try {

TableView.TableViewSelectionModel<medicine\_variable> s = tableview.getSelectionModel();

s.selectFirst();

} catch (Exception e) {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, e, "no data", JOptionPane.INFORMATION\_MESSAGE);

}

}

@FXML

private void btnlast(ActionEvent event) {

TableView.TableViewSelectionModel<medicine\_variable> s = tableview.getSelectionModel();

s.selectLast();

}

@FXML

private void btnnext(ActionEvent event) {

try {

TableView.TableViewSelectionModel<medicine\_variable> s = tableview.getSelectionModel();

s.selectNext();

} catch (Exception e) {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, e, "not date", JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void btnprev(ActionEvent event) {

TableView.TableViewSelectionModel<medicine\_variable> s = tableview.getSelectionModel();

s.selectPrevious();

}

@FXML

private void btncancel(ActionEvent event) {

clear();

}

@FXML

private void btnref(MouseEvent event) {

display();

}

@FXML

private void btnadd(ActionEvent event) {

if (txtcustomernm.getText() != null) {

System.out.println("click");

txtinvoiceno.setEditable(false);

txtcustomernm.setEditable(false);

// txtnameofite.setText(null);

txtquantity.setText(null);

txtrate.setText(null);

txtamount.setText(null);

mfgdate.setValue(null);

expiredate.setValue(null);

txtmobileno.setEditable(false);

txtbatchno.setText(null);

txtdiscountno.setEditable(false);

txtgst.setEditable(false);

txttotalamount.setEditable(false);

}

}

public Connection connection() {

try {

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

return c;

} catch (Exception e) {

System.out.println(e);

return null;

}

}

@FXML

private void btnprint(ActionEvent event) throws JRException {

c = connection();

Map<String, Object> m = new HashMap<>();

m.put("invoiceno", Integer.parseInt(txtinvoiceno.getText()));

m.put("customername", txtcustomernm.getText());

m.put("mobileno", txtmobileno.getText());

if (p\_case.isSelected()) {

m.put("payment", p\_case.getText());

} else {

m.put("payment", p\_online.getText());

}

m.put("discount", txtdiscountno.getText());

m.put("gst", txtgst.getText());

m.put("total", txttotalamount.getText());

jr = JasperCompileManager.compileReport("D:\\java\\.net\\projectfx\\src\\projectfx\\report1.jrxml");

jp = JasperFillManager.fillReport(jr, m, c);

SwingNode swingNode = new SwingNode();

JRViewer jrv = new JRViewer(jp);

swingNode.setContent(jrv);

VBox vbox = new VBox();

vbox.getChildren().add(swingNode);

vbox.prefHeightProperty().bind(invoice\_anchorpane.heightProperty());

vbox.prefWidthProperty().bind(invoice\_anchorpane.widthProperty());

invoice\_anchorpane.getChildren().clear();

invoice\_anchorpane.getChildren().add(vbox);

jrv.setOpaque(true);

jrv.setVisible(true);

}

@FXML

private void assign(MouseEvent event) {

// txtrate.setText("null");

String cb=combo.getValue();

try {

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

PreparedStatement p = c.prepareStatement("select \*from meditbl where mname='"+cb+"'" );

ResultSet rs = p.executeQuery();

// current\_variable cv;

// String ms;

while (rs.next()) {

txtrate.setText(String.valueOf(rs.getFloat("rate")));

mfgdate.setValue(rs.getDate("mfg\_date").toLocalDate());

expiredate.setValue(rs.getDate("exp\_date").toLocalDate());

txtbatchno.setText(rs.getString("batch"));

}

} catch (Exception e) {

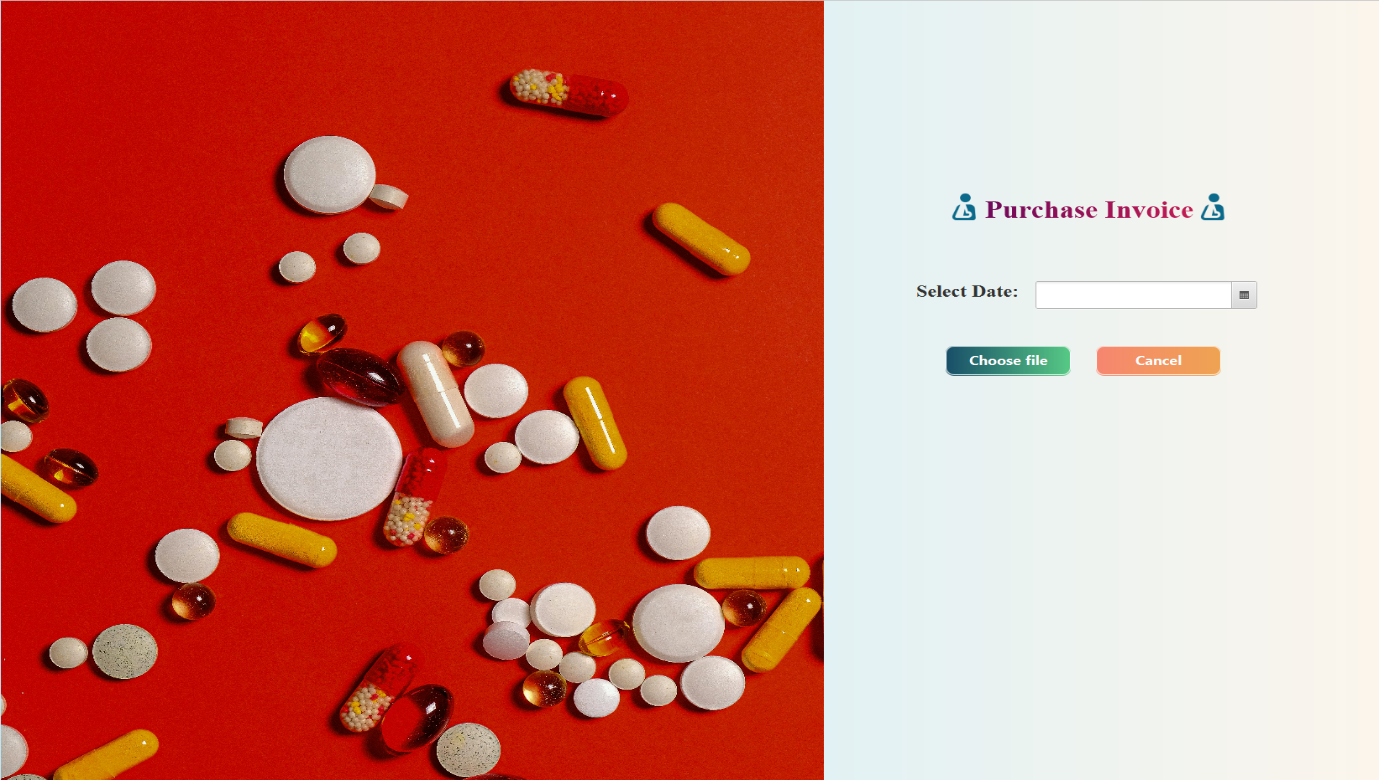
System.out.println(e);

}

}

}

* **Purchase.fxml :**



* **PurchaseController.java :**

package projectfx;

import java.awt.Component;

import java.io.ByteArrayOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.net.URL;

import java.sql.Blob;

import java.sql.Date;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.AnchorPane;

import javafx.scene.layout.VBox;

import javafx.scene.media.Media;

import javafx.scene.media.MediaPlayer;

import javafx.scene.media.MediaView;

import javafx.stage.FileChooser;

import javafx.stage.Stage;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.filechooser.FileNameExtensionFilter;

public class PurchaseController implements Initializable {

@FXML

private Button btnfile;

@FXML

private Button btncancel;

String f;

/\*\*

\* Initializes the controller class.

\*/

byte[] photo = null;

String filename = null;

static File selectedfile;

@FXML

private DatePicker dt;

@FXML

private Label l;

String fis;

@FXML

private AnchorPane anchor;

static Date date;

@FXML

private VBox box;

AnchorPane apn = new AnchorPane();

@Override

public void initialize(URL url, ResourceBundle rb) {

}

@FXML

private void btncancel(ActionEvent event) {

}

@FXML

private void filechoose(ActionEvent event) {

System.out.println("click");

FileChooser chooser = new FileChooser();

chooser.getExtensionFilters().add(new FileChooser.ExtensionFilter("Image Files", "\*.png", "\*.jpg", "\*.gif"));

selectedfile = chooser.showOpenDialog(new Stage());

date = Date.valueOf(dt.getValue());

String f = selectedfile.getPath();

try {

if (selectedfile != null) {

FXMLLoader.load(getClass().getResource("purchase.fxml"));

FXMLLoader.load(getClass().getResource("image.fxml"));

scene.getStylesheets().add(getClass().getResource("medi\_design.css").toExternalForm());

anchor.getScene().getWindow().hide();

FXMLLoader loader = new FXMLLoader(getClass().getResource("Dashboard.fxml"));

Parent root = loader.load();

DashboardController d = loader.getController();

FXMLLoader loading = new FXMLLoader(getClass().getResource("image.fxml"));

apn = loading.load();

d.pn2.setContent(apn);

apn.prefHeightProperty().bind(d.pn2.heightProperty());

Stage primaryStage = new Stage();

Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("dashboard\_design.css").toExternalForm());

primaryStage.setScene(scene);

primaryStage.setMaximized(true);

primaryStage.show();

} else {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, "no file selected");

}

} catch (Exception e) {

Component rootPane = null;

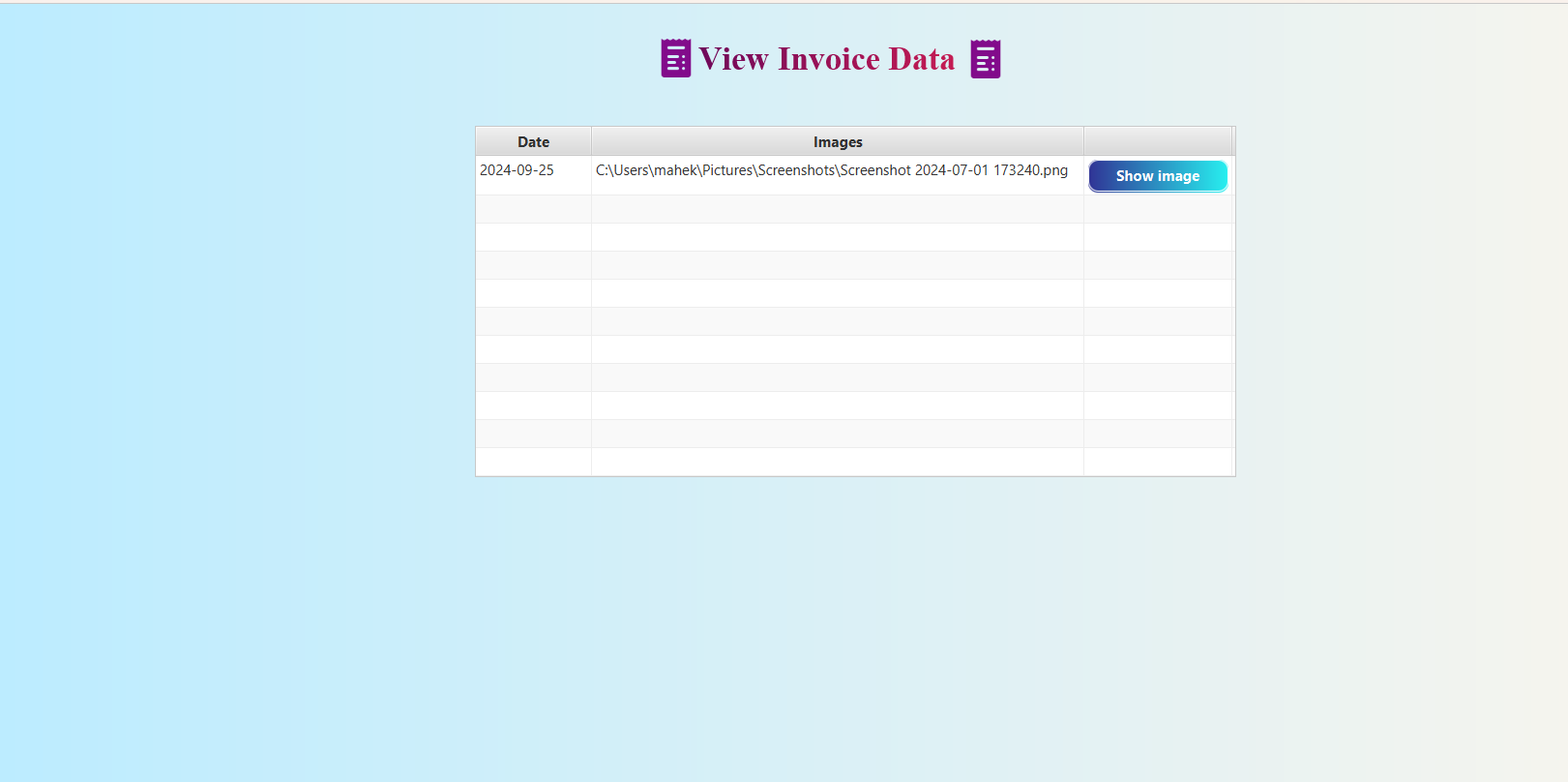
JOptionPane.showMessageDialog(rootPane, e, "try again", JOptionPane.ERROR\_MESSAGE);

}

}

}

* **Displayinvoice.fxml :**



* **DisplayinvoiceController.java :**

package projectfx;

import java.net.URL;

import java.sql.Connection;

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.geometry.Pos;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.TableCell;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

public class DisplayinvoiceController implements Initializable {

@FXML

private TableView<setvar> tableview;

@FXML

private TableColumn<setvar, Date> date;

@FXML

private TableColumn<setvar, String> img;

Connection c;

static String path;

@FXML

private TableColumn<String, Button> btn;

@FXML

private AnchorPane thisanchor;

AnchorPane apn = new AnchorPane();

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

getdatelist();

display();

}

ObservableList<setvar> getdatelist() {

ObservableList<setvar> invoicelist = FXCollections.observableArrayList();

try {

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

PreparedStatement p = c.prepareStatement("select \*from purchase");

ResultSet rs = p.executeQuery();

setvar s;

while (rs.next()) {

s = new setvar(rs.getDate("date"), rs.getString("images"));

//System.out.println(s);

invoicelist.add(s);

}

} catch (Exception e) {

System.out.println(e);

}

return invoicelist;

}

public void display() {

try {

ObservableList<setvar> list = getdatelist();

date.setCellValueFactory(new PropertyValueFactory<setvar, Date>("date"));

img.setCellValueFactory(new PropertyValueFactory<setvar, String>("images"));

btn.setCellFactory(col -> new TableCell<String, Button>() {

@Override

protected void updateItem(Button obj, boolean b) {

super.updateItem(obj, b);

if (b) {

setGraphic(null);

} else {

Button B = new Button("Show image");

B.setPrefSize(150, 30);

B.setId("button");

setGraphic(B);

B.setOnAction((event) -> {

System.out.println("hello");

path = tableview.getSelectionModel().getSelectedItem().images;

System.out.println(path);

try {

thisanchor.getScene().getWindow().hide();

FXMLLoader loader = new FXMLLoader(getClass().getResource("Dashboard.fxml"));

Parent root = loader.load();

DashboardController d = loader.getController();

FXMLLoader loading = new FXMLLoader(getClass().getResource("viewinvoice.fxml"));

apn = loading.load();

d.pn2.setContent(apn);

apn.prefHeightProperty().bind(d.pn2.heightProperty());

Stage primaryStage = new Stage();

Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("dashboard\_design.css").toExternalForm());

primaryStage.setScene(scene);

primaryStage.setMaximized(true);

primaryStage.show();

} catch (Exception e) {

System.out.println(e);

}

});

}

}

});

tableview.setItems(list);

} catch (Exception e) {

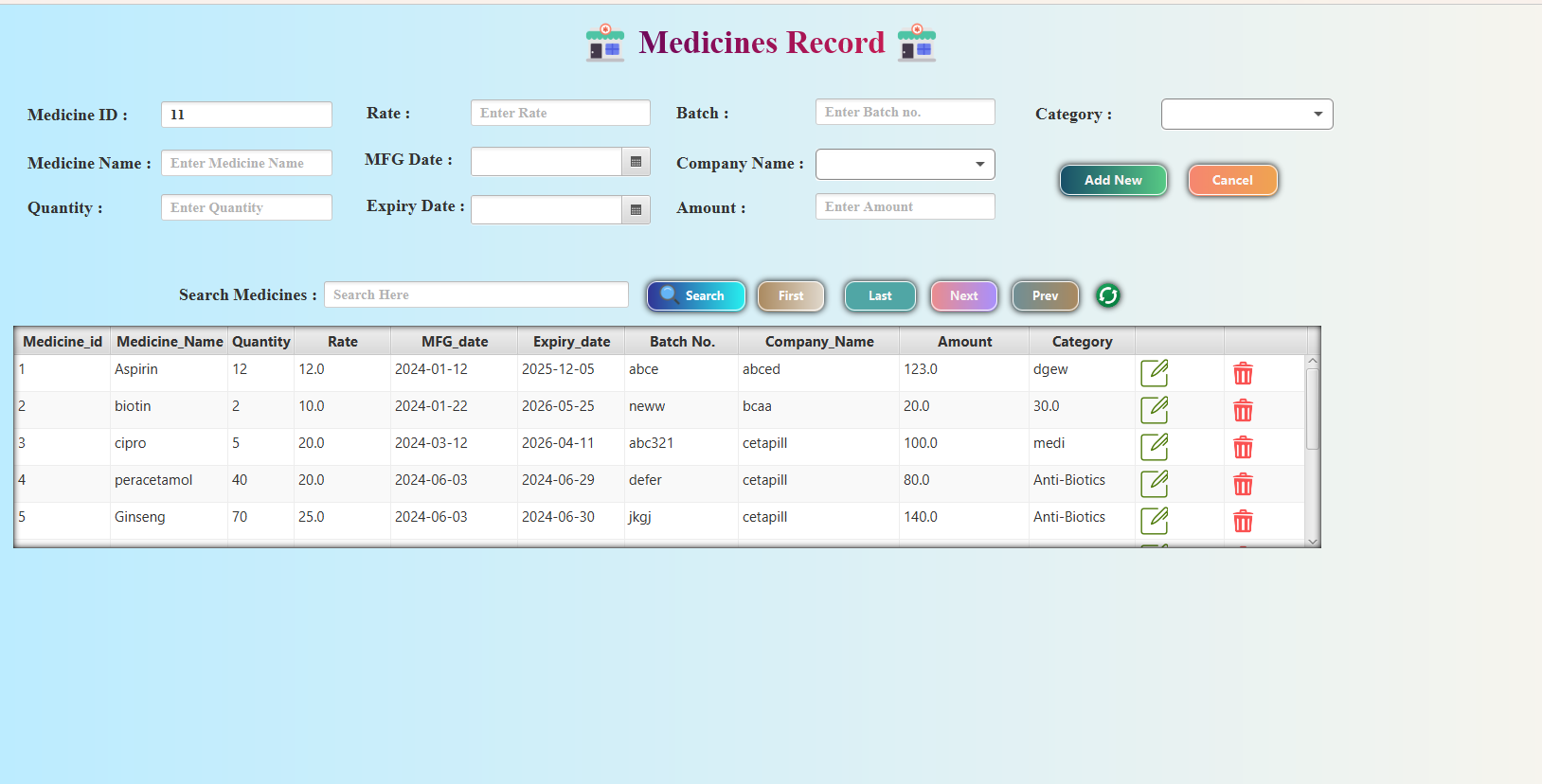
System.out.println(e + "not display");

}

}

}

* **Medicines.fxml :**



* **MedicinesController.java :**

import java.net.URL;

import java.sql.\*;

import java.time.LocalDate;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.scene.Cursor;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.TableCell;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.scene.input.KeyEvent;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.AnchorPane;

import javafx.scene.layout.HBox;

import javax.swing.JOptionPane;

public class MedicinesController implements Initializable {

@FXML

private AnchorPane body;

@FXML

private TableView<tbldata> tbl;

@FXML

private TextField id;

@FXML

private TextField nm;

@FXML

private TextField Qt;

@FXML

private TextField rat;

@FXML

private TextField amot;

@FXML

private DatePicker mf;

@FXML

private DatePicker ex;

@FXML

private ComboBox<String> cat;

@FXML

private TextField batc;

crud cr = new crud();

ObservableList<tbldata> listM = FXCollections.observableArrayList();

ObservableList<String> data = FXCollections.observableArrayList();

ObservableList<String> list = FXCollections.observableArrayList();

int index = -1;

Connection conn = null;

ResultSet r = null;

PreparedStatement pst = null;

tbldata tbldata = null;

@FXML

private TableColumn<String, Button> action;

int c;

int newid, qtyy,medi\_id;

String cate, name, b,comp;

float rt, amount, gstt, totamt;

Date m, expiry;

String trans;

@FXML

private ImageView reload;

@FXML

private TableColumn<tbldata, Integer> mid;

@FXML

private TableColumn<tbldata, String> mname;

@FXML

private TableColumn<tbldata, Integer> qty;

@FXML

private TableColumn<tbldata, Float> rate;

@FXML

private TableColumn<tbldata, Date> mfg\_date;

@FXML

private TableColumn<tbldata, Date> exp\_date;

@FXML

private TableColumn<tbldata, Float> amt;

@FXML

private TableColumn<tbldata, String> batch;

@FXML

private TableColumn<tbldata, String> category;

@FXML

private Button search;

@FXML

private TextField searchText;

@FXML

private TableColumn<String, Button> editcol;

@FXML

private Label title;

@FXML

private TableColumn<tbldata, String> company\_name;

@FXML

private ComboBox<String> company;

@Override

public void initialize(URL url, ResourceBundle rb) {

conn = connect();

display();

load();

combo();

comboCategory();

medi\_id++;

id.setText(String.valueOf(medi\_id));

}

public void inform() {

newid = Integer.parseInt(id.getText());

name = nm.getText();

qtyy = Integer.parseInt(Qt.getText());

rt = Float.parseFloat(rat.getText());

m = Date.valueOf(mf.getValue());

expiry = Date.valueOf(ex.getValue());

b = batc.getText();

comp=company.getSelectionModel().getSelectedItem();

amount = Float.parseFloat(amot.getText());

cate = cat.getSelectionModel().getSelectedItem();

}

public void clear()

{

id.setText(null);

nm.setText(null);

Qt.setText(null);

rat.setText(null);

mf.setValue(null);

ex.setValue(null);

batc.setText(null);

company.getSelectionModel().clearSelection();

company.setValue(null);

cat.getSelectionModel().clearSelection();

cat.setValue(null);

amot.setText(null);

searchText.setText(null);

medi\_id++;

id.setText(String.valueOf(medi\_id));

}

@FXML

private void btnnew(ActionEvent event) {

inform();

try {

if (cr.insertdata(newid, name, qtyy, rt, m, expiry, b,comp, cate, amount) > 0 && cr.CategoryProduct(name,cate)>0) {

if (cr.totalAdd(qtyy,name,newid)>0) {

JOptionPane.showMessageDialog(null, "Medicines Added Successfully", "Adding ", JOptionPane.INFORMATION\_MESSAGE);

}

else{

System.out.println("try again");

}

display();

clear();

} else {

JOptionPane.showMessageDialog(null, "Try Again.","adding",JOptionPane.WARNING\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"adding",JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void multiply(KeyEvent event) {

try {

if (!Qt.getText().isEmpty() && !rat.getText().isEmpty()) {

int a = Integer.parseInt(Qt.getText());

float b = Float.parseFloat(rat.getText());

amot.setText(String.valueOf(a \* b));

}

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(null, ex,"Total",JOptionPane.ERROR\_MESSAGE);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Total",JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void click(MouseEvent event) {

id.setText(String.valueOf(tbl.getSelectionModel().getSelectedItem().getMid()));

nm.setText(tbl.getSelectionModel().getSelectedItem().getMname());

Qt.setText(String.valueOf(tbl.getSelectionModel().getSelectedItem().getQty()));

rat.setText(String.valueOf(tbl.getSelectionModel().getSelectedItem().getRate()));

mf.setValue(tbl.getSelectionModel().getSelectedItem().getMfg\_date().toLocalDate());

ex.setValue(tbl.getSelectionModel().getSelectedItem().getExp\_date().toLocalDate());

batc.setText(tbl.getSelectionModel().getSelectedItem().getBatch());

company.setValue(tbl.getSelectionModel().getSelectedItem().getCompany\_name());

amot.setText(String.valueOf(tbl.getSelectionModel().getSelectedItem().getAmt()));

cat.setValue(tbl.getSelectionModel().getSelectedItem().getCategory());

// gs.setText(String.valueOf(tbl.getSelectionModel().getSelectedItem().getGst()));

// tamot.setText(String.valueOf(tbl.getSelectionModel().getSelectedItem().getTot\_amt()));

c=tbl.getSelectionModel().getSelectedItem().getMid();

}

public void combo()

{

try {

conn = connect();

PreparedStatement pst = conn.prepareStatement("select cname from company");

r = pst.executeQuery();

while (r.next()) {

data.add(r.getString("cname"));

}

company.setItems(data);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Company Fetching",JOptionPane.ERROR\_MESSAGE);

}

}

public void comboCategory()

{

try {

conn = connect();

pst = conn.prepareStatement("select category\_name from categorytbl");

r = pst.executeQuery();

while (r.next()) {

list.add(r.getString("category\_name"));

}

cat.setItems(list);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Company Fetching",JOptionPane.ERROR\_MESSAGE);

}

}

public void display() {

try {

conn = connect();

listM.clear();

PreparedStatement pst = conn.prepareStatement("select \*from meditbl order by mid");

r = pst.executeQuery();

while (r.next()) {

listM.add(new tbldata(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category")));

medi\_id=r.getInt("mid");

tbl.setItems(listM);

}

} catch (Exception e) {

System.out.println(e);

}

}

public Connection connect() {

try {

Class.forName("com.mysql.jdbc.Driver");

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Connection Alert",JOptionPane.WARNING\_MESSAGE);

}

return conn;

}

@FXML

private void btnRefresh(MouseEvent event) {

clear();

display();

}

@FXML

private void btnSearch(ActionEvent event) {

String s = searchText.getText();

try {

conn = connect();

listM.clear();

PreparedStatement pst = null;

if (Character.isAlphabetic(s.charAt(0)) || Character.isAlphabetic(4)) {

pst = conn.prepareStatement("select \*from meditbl where mname like '%" + s + "%' or mfg\_date like '%" + s + "%' or exp\_date like '%" + s + "%' or batch like '%" + s + "%' or company\_name like '%" + s + "%' or category like '%" + s + "%' ");

} else if (Character.isDigit(s.charAt(0))) {

pst = conn.prepareStatement("select \*from meditbl where mid=" + s + " or qty=" + s + " or rate=" + s + " or amt=" + s );

}

r = pst.executeQuery();

while (r.next()) {

listM.add(new tbldata(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"),r.getFloat("amt"),

r.getString("category")));

tbl.setItems(listM);

load();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Search Alert",JOptionPane.WARNING\_MESSAGE);

}

}

@FXML

private void btnFirst(ActionEvent event) {

try {

listM.clear();

pst = conn.prepareStatement("select \*from meditbl where mid=1");

r = pst.executeQuery();

while (r.next()) {

listM.add(new tbldata(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category")));

tbl.setItems(listM);

load();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"First Alert",JOptionPane.WARNING\_MESSAGE);

}

}

public void load()

{

mid.setCellValueFactory(new PropertyValueFactory<>("mid"));

mname.setCellValueFactory(new PropertyValueFactory<>("mname"));

qty.setCellValueFactory(new PropertyValueFactory<>("qty"));

rate.setCellValueFactory(new PropertyValueFactory<>("rate"));

mfg\_date.setCellValueFactory(new PropertyValueFactory<>("mfg\_date"));

exp\_date.setCellValueFactory(new PropertyValueFactory<>("exp\_date"));

batch.setCellValueFactory(new PropertyValueFactory<>("batch"));

company\_name.setCellValueFactory(new PropertyValueFactory<>("company\_name"));

amt.setCellValueFactory(new PropertyValueFactory<>("amt"));

category.setCellValueFactory(new PropertyValueFactory<>("category"));

// gst.setCellValueFactory(new PropertyValueFactory<>("gst"));

// tot\_amt.setCellValueFactory(new PropertyValueFactory<>("tot\_amt"));

action.setCellFactory(col -> new TableCell<String, Button>() {

@Override

protected void updateItem(Button button, boolean empty) {

super.updateItem(button, empty);

if (empty) {

setGraphic(null);

} else {

Button deleteButton = new Button();

deleteButton.setId("btn");

deleteButton.setPrefWidth(27);

setGraphic(deleteButton);

deleteButton.setOnAction(event -> {

int i = getIndex();

int n = tbl.getItems().get(i).getMid();

trans = tbl.getItems().get(i).getMname();

try{

if (cr.deletedata(n)>0 && cr.CategoryProductDelete(trans)>0) {

if (cr.totalDelete(n)>0) {

JOptionPane.showMessageDialog(null, "Medicine Deleted SuccessFully","Medicines Deletion",JOptionPane.INFORMATION\_MESSAGE);

}

display();

clear();

} else {

JOptionPane.showMessageDialog(null, "Try Again.","Medicines Deletion",JOptionPane.WARNING\_MESSAGE);

}

}catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

});

}

}

});

editcol.setCellFactory(col -> new TableCell<String, Button>() {

@Override

protected void updateItem(Button button, boolean empty) {

super.updateItem(button, empty);

if (empty) {

setGraphic(null);

} else {

Button editButton = new Button();

editButton.setId("btnedit");

editButton.setPrefWidth(30);

setGraphic(editButton);

editButton.setOnAction(event -> {

inform();

try{

if (cr.updatedata(newid, name, qtyy, rt, m, expiry, b,comp, cate, amount) >0) {

if (cr.totalUpdate(name, newid)>0) {

JOptionPane.showMessageDialog(null, "Changes are saved SuccessFully","Medicine Changes",JOptionPane.INFORMATION\_MESSAGE);

}

display();

clear();

} else {

JOptionPane.showMessageDialog(null, "Try Again.","Medicine Changes",JOptionPane.WARNING\_MESSAGE);

}

}catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

display();

});

}

}

});

}

@FXML

private void btnLast(ActionEvent event) {

try {

int n = listM.size();

listM.clear();

pst = conn.prepareStatement("select \*from meditbl where mid="+n);

ResultSet r = pst.executeQuery();

while (r.next()) {

listM.add(new tbldata(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category")));

tbl.setItems(listM);

load();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Last Alert",JOptionPane.WARNING\_MESSAGE);

}

}

@FXML

private void btnNext(ActionEvent event) {

try {

listM.clear();

pst = conn.prepareStatement("select \*from meditbl where mid="+(c+1));

r = pst.executeQuery();

while (r.next()) {

listM.add(new tbldata(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category")));

tbl.setItems(listM);

load();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Next Alert",JOptionPane.WARNING\_MESSAGE);

}

}

@FXML

private void btnPrev(ActionEvent event) {

try {

listM.clear();

pst = conn.prepareStatement("select \*from meditbl where mid="+(c-1));

r = pst.executeQuery();

while (r.next()) {

listM.add(new tbldata(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category")));

tbl.setItems(listM);

load();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Prev Alert",JOptionPane.WARNING\_MESSAGE);

}

}

private void btnExit(ActionEvent event) {

System.exit(0);

}

@FXML

private void btnSearch(MouseEvent event) {

String s = searchText.getText();

try {

conn = connect();

listM.clear();

// System.out.println("select \*from meditbl where mid=" + s + " or mname='" + s + "' or qty=" + s + " or rate=" + s + " or mfg\_date=" + s + " or exp\_date=" + s + " or batch='" + s + "' or amt=" + s + " or category='" + s + "' or gst=" + s + " or tot\_amt=" + s);

PreparedStatement pst = null;

if (Character.isAlphabetic(s.charAt(0)) || Character.isAlphabetic(4)) {

pst = conn.prepareStatement("select \*from meditbl where mname like '%" + s + "%' or mfg\_date like '%" + s + "%' or exp\_date like '%" + s + "%' or batch like '%" + s + "%' or category like '%" + s + "%' ");

} else if (Character.isDigit(s.charAt(0))) {

pst = conn.prepareStatement("select \*from meditbl where mid=" + s + " or qty=" + s + " or rate=" + s + " or amt=" + s );

}

r = pst.executeQuery();

while (r.next()) {

listM.add(new tbldata(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category")));

tbl.setItems(listM);

load();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Search Alert",JOptionPane.WARNING\_MESSAGE);

}

}

@FXML

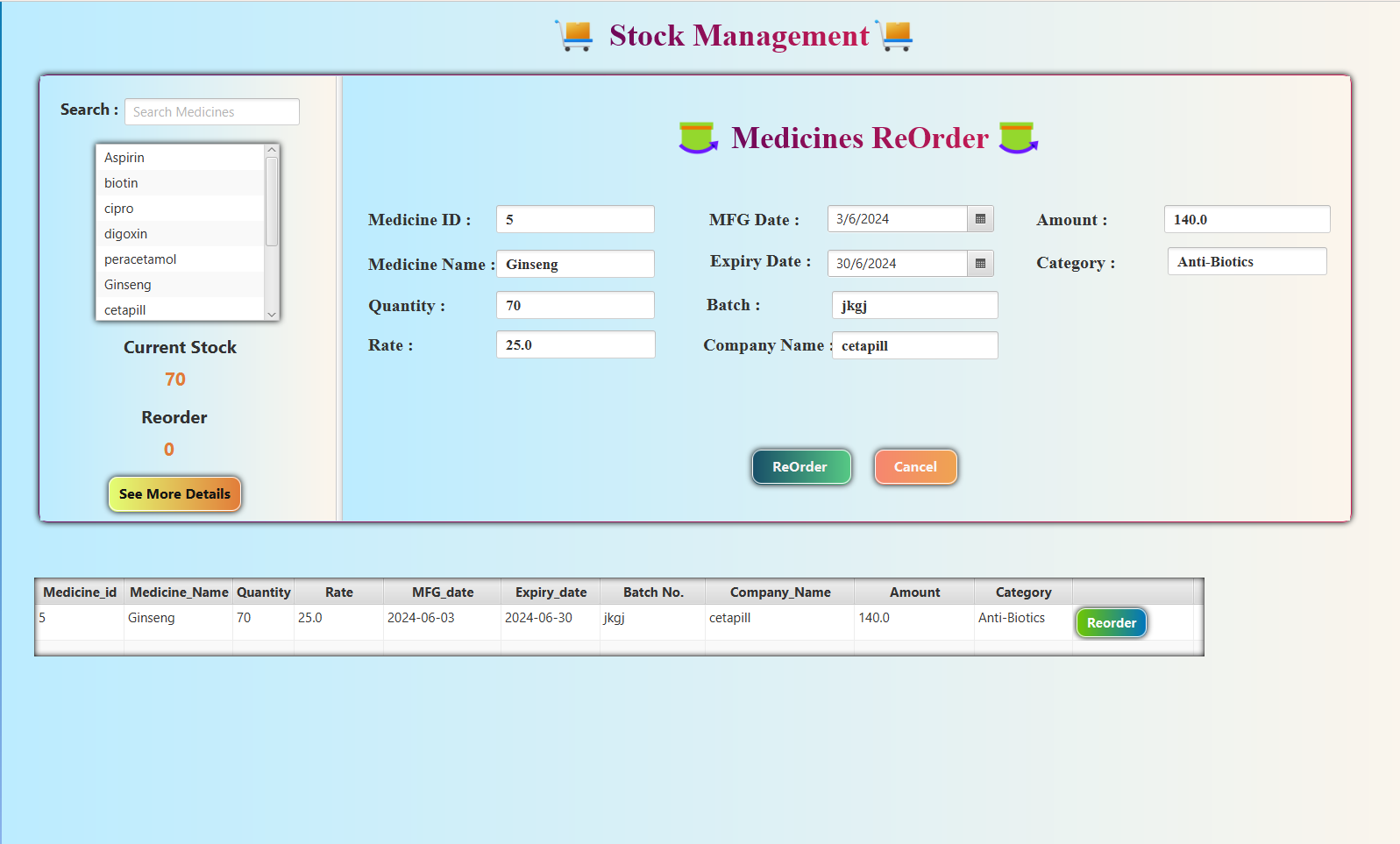
private void btnCancel(ActionEvent event) {

clear();

}

}

* **Stock.fxml :**



* **StockController.java :**

package projectfx;

import java.net.URL;

import java.sql.\*;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Button;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.ListView;

import javafx.scene.control.TableCell;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.effect.DropShadow;

import javafx.scene.input.KeyEvent;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.AnchorPane;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class StockController implements Initializable {

@FXML

private ListView<String> listview;

Connection conn;

private TextField search;

ObservableList<String> list = FXCollections.observableArrayList();

@FXML

private Label qtyText;

@FXML

private TableView<tbldata> tbl;

@FXML

private TableColumn<tbldata, Integer> mid;

@FXML

private TableColumn<tbldata, String> mname;

@FXML

private TableColumn<tbldata, Integer> qty;

@FXML

private TableColumn<tbldata, Float> rate;

@FXML

private TableColumn<tbldata, Date> mfg\_date;

@FXML

private TableColumn<tbldata, Date> exp\_date;

@FXML

private TableColumn<tbldata, String> batch;

@FXML

private TableColumn<tbldata, String> company\_name;

@FXML

private TableColumn<tbldata, Float> amt;

@FXML

private TableColumn<tbldata, String> category;

ObservableList<tbldata> listM = FXCollections.observableArrayList();

@FXML

private TableColumn<String, Button> reorder;

@FXML

private TextField nm;

@FXML

private TextField Qt;

@FXML

private TextField rat;

@FXML

private DatePicker mf;

@FXML

private DatePicker ex;

@FXML

private TextField batc;

@FXML

private TextField amot;

@FXML

private Label title;

@FXML

private TextField id;

@FXML

private TextField company;

@FXML

private TextField cat;

@FXML

private TextField searchText;

@FXML

private Label reorderCount;

crud cr = new crud();

MedicinesController mc = new MedicinesController();

int newid, qtyy;

String cate, name, b,comp;

float rt, amount, gstt, totamt;

Date m, expiry;

@FXML

private AnchorPane panel;

@Override

public void initialize(URL url, ResourceBundle rb) {

display();

load();

}

public Connection connect()

{

try {

Class.forName("com.mysql.jdbc.Driver");

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb","root","");

} catch (Exception e) {

System.out.println(e);

}

return conn;

}

public void display()

{

try {

conn = connect();

PreparedStatement pst = conn.prepareStatement("select mname from meditbl");

ResultSet r = pst.executeQuery();

while (r.next()) {

list.add(r.getString("mname"));

}

listview.setItems(list);

listview.requestLayout();

} catch (Exception e) {

System.out.println(e);

}

}

public void inform() {

newid = Integer.parseInt(id.getText());

name = nm.getText();

qtyy = Integer.parseInt(Qt.getText());

rt = Float.parseFloat(rat.getText());

m = Date.valueOf(mf.getValue());

expiry = Date.valueOf(ex.getValue());

b = batc.getText();

comp=company.getText();

amount = Float.parseFloat(amot.getText());

cate = cat.getText();

}

String data,test;

int c = 0;

public boolean Redisplay()

{

try {

conn = connect();

PreparedStatement pst = conn.prepareStatement("select mname from reordertbl where mname like '%"+test+"%'");

c=0;

ResultSet r = pst.executeQuery();

while (r.next()) {

data = r.getString("mname") + " ";

c++;

}

return true;

} catch (Exception e) {

System.out.println(e);

return false;

}

}

public void clear()

{

id.setText(null);

nm.setText(null);

Qt.setText(null);

rat.setText(null);

mf.setValue(null);

ex.setValue(null);

batc.setText(null);

company.setText(null);

cat.setText(null);

// gs.setText(null);

amot.setText(null);

}

@FXML

private void TextKey(KeyEvent event) {

if (!searchText.getText().isEmpty()) {

conn = connect();

String s = searchText.getText();

if (Character.isAlphabetic(s.charAt(0))) {

try {

list.clear();

PreparedStatement pst = conn.prepareStatement("select mname from meditbl where mname like '%"+s+"%'");

ResultSet r = pst.executeQuery();

while (r.next()) {

list.add(r.getString("mname"));

}

listview.setItems(list);

} catch (Exception e) {

System.out.println(e);

}

}

}

}

@FXML

private void listviewMouseClick(MouseEvent event) {

test = listview.getSelectionModel().getSelectedItem();

try {

PreparedStatement pst = conn.prepareStatement("select qty from meditbl where mname like '%"+test+"%'");

ResultSet r = pst.executeQuery();

while (r.next()) {

qtyText.setText(String.valueOf(r.getInt("qty")));

}

if (Redisplay()==true) {

reorderCount.setText(String.valueOf(c));

} else {

reorderCount.setText("0");

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"quantity",JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void btnDetails(ActionEvent event) {

String test = listview.getSelectionModel().getSelectedItem();

listM.clear();

if (listview.getSelectionModel().getSelectedItem()==null) {

String imagepath = "C:\\Users\\poojajungi\\Downloads\\hand.png";

ImageIcon icon = new ImageIcon(imagepath);

JOptionPane.showMessageDialog(null, "Please Select the Medicine.","Medicines message",JOptionPane.PLAIN\_MESSAGE,icon);

}

else{

try {

tbl.setVisible(true);

PreparedStatement pst = conn.prepareStatement("select \* from meditbl where mname like '%"+test+"%'");

ResultSet r = pst.executeQuery();

while (r.next()) {

listM.add(new tbldata(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category")));

tbl.setItems(listM);

}

listview.getSelectionModel().clearSelection();

// c=0;

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"See more details ",JOptionPane.ERROR\_MESSAGE);

}

}

}

public void load()

{

mid.setCellValueFactory(new PropertyValueFactory<>("mid"));

mname.setCellValueFactory(new PropertyValueFactory<>("mname"));

qty.setCellValueFactory(new PropertyValueFactory<>("qty"));

rate.setCellValueFactory(new PropertyValueFactory<>("rate"));

mfg\_date.setCellValueFactory(new PropertyValueFactory<>("mfg\_date"));

exp\_date.setCellValueFactory(new PropertyValueFactory<>("exp\_date"));

batch.setCellValueFactory(new PropertyValueFactory<>("batch"));

company\_name.setCellValueFactory(new PropertyValueFactory<>("company\_name"));

amt.setCellValueFactory(new PropertyValueFactory<>("amt"));

category.setCellValueFactory(new PropertyValueFactory<>("category"));

reorder.setCellFactory(col -> new TableCell<String,Button>(){

@Override

protected void updateItem(Button button, boolean empty) {

super.updateItem(button, empty);

if (empty) {

setGraphic(null);

} else {

Button reorderbtn = new Button("Reorder");

reorderbtn.setPrefSize(80,15);

reorderbtn.setId("btnReorder");

DropShadow ds = new DropShadow();

reorderbtn.setEffect(ds);

setGraphic(reorderbtn);

reorderbtn.setOnAction(event -> {

panel.setVisible(true);

id.setText(mid.getCellData(0).toString());

nm.setText(mname.getCellData(0));

Qt.setText(qty.getCellData(0).toString());

rat.setText(rate.getCellData(0).toString());

mf.setValue(mfg\_date.getCellData(0).toLocalDate());

ex.setValue(exp\_date.getCellData(0).toLocalDate());

batc.setText(batch.getCellData(0));

company.setText(company\_name.getCellData(0));

amot.setText(amt.getCellData(0).toString());

cat.setText(category.getCellData(0));

// gs.setText(gst.getCellData(0).toString());

// tamot.setText(tot\_amt.getCellData(0).toString());

});

}

}

});

}

@FXML

private void btnnew(ActionEvent event) {

try {

String imagepath = "C:\\Users\\poojajungi\\Downloads\\check.png";

ImageIcon icon = new ImageIcon(imagepath);

inform();

if (cr.reOrderAdd( name, qtyy, rt, m, expiry, b, comp, cate, amount)>0) {

if (cr.totalUpdateReOrder(name)>0) {

JOptionPane.showMessageDialog(null, "Reorder Successfully.", "Reorder ", JOptionPane.PLAIN\_MESSAGE,icon);

}

try {

conn = connect();

PreparedStatement pst = conn.prepareStatement("select mname from reordertbl where mname like '%"+name+"%' ");

c=0;

ResultSet r = pst.executeQuery();

while (r.next()) {

data = r.getString("mname") + " ";

c++;

}

reorderCount.setText(String.valueOf(c));

listview.getSelectionModel().clearSelection();

} catch (Exception e) {

System.out.println(e);

}

clear();

listM.clear();

panel.setVisible(false);

tbl.setVisible(false);

} else {

JOptionPane.showMessageDialog(null, "Try Again.","Reorder",JOptionPane.WARNING\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Reorder",JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void btnCancel(ActionEvent event) {

clear();

}

}

* **returnstock.fxml :**



* **returnstockController.java :**

package projectfx;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.Date;

import java.sql.Timestamp;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class ReturnStockController implements Initializable {

Connection conn;

ObservableList<tblreorder> listM = FXCollections.observableArrayList() ;

@FXML

private TableView<tblreorder> newtbl;

@FXML

private TableColumn<tblreorder, Integer> mid1;

@FXML

private TableColumn<tblreorder, String> mname1;

@FXML

private TableColumn<tblreorder, Integer> qty1;

@FXML

private TableColumn<tblreorder, Float> rate1;

@FXML

private TableColumn<tblreorder,Date> mfg\_date1;

@FXML

private TableColumn<tblreorder,Date> exp\_date1;

@FXML

private TableColumn<tblreorder, String> batch1;

@FXML

private TableColumn<tblreorder, String> company\_name1;

@FXML

private TableColumn<tblreorder, Float> amt1;

@FXML

private TableColumn<tblreorder, String> category1;

@FXML

private TableColumn<tblreorder, Timestamp> order\_date1;

@FXML

private AnchorPane body;

@FXML

private Label total;

AnchorPane apn = new AnchorPane();

int newid;

int qtyy;

String cate, name, b,comp;

float rt, amount, gstt, totamt;

Date m, expiry;

Timestamp ord;

String a = "";

crud cr = new crud();

ObservableList<tblreorder> d = FXCollections.observableArrayList();

@Override

public void initialize(URL url, ResourceBundle rb) {

}

public void redisplayed(ObservableList<tblreorder> list)

{

newtbl.getItems().addAll(list);

mid1.setCellValueFactory(new PropertyValueFactory<>("mid"));

mname1.setCellValueFactory(new PropertyValueFactory<>("mname"));

qty1.setCellValueFactory(new PropertyValueFactory<>("qty"));

rate1.setCellValueFactory(new PropertyValueFactory<>("rate"));

mfg\_date1.setCellValueFactory(new PropertyValueFactory<>("mfg\_date"));

exp\_date1.setCellValueFactory(new PropertyValueFactory<>("exp\_date"));

batch1.setCellValueFactory(new PropertyValueFactory<>("batch"));

company\_name1.setCellValueFactory(new PropertyValueFactory<>("company\_name"));

amt1.setCellValueFactory(new PropertyValueFactory<>("amt"));

category1.setCellValueFactory(new PropertyValueFactory<>("category"));

order\_date1.setCellValueFactory(new PropertyValueFactory<>("order\_date"));

total.setText(String.valueOf(newtbl.getItems().size()));

}

@FXML

private void btnReturn(ActionEvent event){

int n = newtbl.getItems().size();

if (n==0) {

JOptionPane.showMessageDialog(null, "Please Select the Stocks.","Null Selection",JOptionPane.WARNING\_MESSAGE);

}

try {

int res=0;

for (int i = 0; i < n; i++) {

newid = newtbl.getItems().get(i).mid;

name = newtbl.getItems().get(i).mname;

qtyy = newtbl.getItems().get(i).qty;

rt = newtbl.getItems().get(i).rate;

m = newtbl.getItems().get(i).mfg\_date;

expiry = newtbl.getItems().get(i).exp\_date;

b = newtbl.getItems().get(i).batch;

comp = newtbl.getItems().get(i).company\_name;

amount = newtbl.getItems().get(i).amt;

cate = newtbl.getItems().get(i).category;

ord = newtbl.getItems().get(i).order\_date;

a = String.valueOf(newid) ;

if (cr.returnAdd(name, qtyy, rt, m, expiry, b, comp, cate, amount, ord)>0) {

if (cr.totalUpdateReturn(name)>0) {

res++;

}

FXMLLoader loader = new FXMLLoader(getClass().getResource("vieworder.fxml"));

Parent root = loader.load();

VieworderController r = loader.getController();

r.remove(a);

newtbl.getItems().remove(i);

total.setText("-");

} else {

JOptionPane.showMessageDialog(null, "Try Again","ReturnStock Message",JOptionPane.WARNING\_MESSAGE);

}

}

if(res>0)

{

JOptionPane.showMessageDialog(null, "Stock Return Successfully.","ReturnStock Message",JOptionPane.INFORMATION\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"ReturnStock Message",JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void btnSelect(ActionEvent event) throws IOException {

body.getScene().getWindow().hide();

FXMLLoader loader = new FXMLLoader(getClass().getResource("Dashboard.fxml"));

Parent root = loader.load();

DashboardController d = loader.getController();

FXMLLoader loading = new FXMLLoader(getClass().getResource("vieworder.fxml"));

apn = loading.load();

d.pn2.setContent(apn);

apn.prefHeightProperty().bind(d.pn2.heightProperty());

Stage primaryStage = new Stage();

Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("dashboard\_design.css").toExternalForm());

primaryStage.setScene(scene);

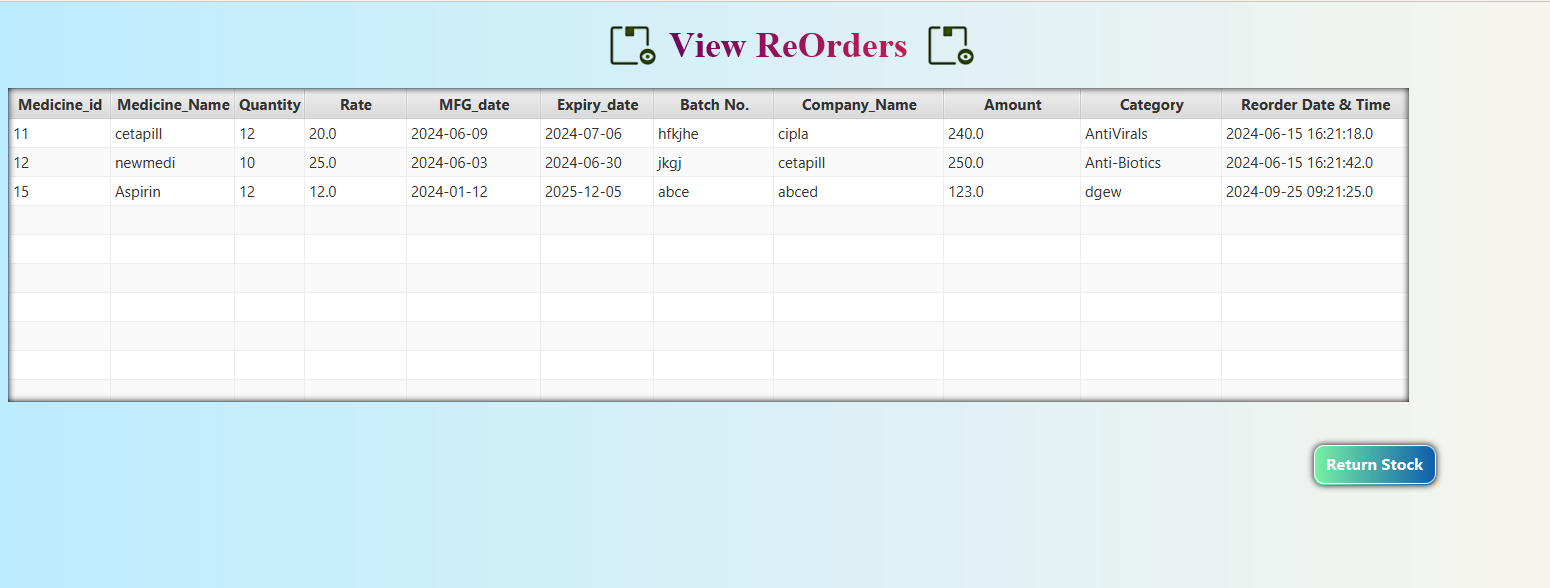
primaryStage.setMaximized(true);

primaryStage.show();

}

}

* **viewOrder.fxml :**



* **viewOrderController.java :**

package projectfx;

import java.sql.\*;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.SelectionMode;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class VieworderController implements Initializable {

@FXML

public TableView<tblreorder> tbl;

@FXML

private TableColumn<tblreorder, Integer> mid;

@FXML

private TableColumn<tblreorder, String> mname;

@FXML

private TableColumn<tblreorder, Integer> qty;

@FXML

private TableColumn<tblreorder, Float> rate;

@FXML

private TableColumn<tblreorder,Date> mfg\_date;

@FXML

private TableColumn<tblreorder,Date> exp\_date;

@FXML

private TableColumn<tblreorder, String> batch;

@FXML

private TableColumn<tblreorder, String> company\_name;

@FXML

private TableColumn<tblreorder, Float> amt;

@FXML

private TableColumn<tblreorder, String> category;

@FXML

private TableColumn<tblreorder, Timestamp> order\_date;

crud cr = new crud();

Connection conn;

ObservableList<tblreorder> listM = FXCollections.observableArrayList() ;

static ObservableList<tblreorder> da = FXCollections.observableArrayList();

AnchorPane apn = new AnchorPane();

@FXML

private AnchorPane Viewbody;

@FXML

private Button btnsubmit;

@Override

public void initialize(URL url, ResourceBundle rb) {

tbl.getSelectionModel().setSelectionMode(SelectionMode.MULTIPLE);

display();

load();

tbl.getSelectionModel().clearSelection();

}

public Connection connect()

{

try {

Class.forName("com.mysql.jdbc.Driver");

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb","root","");

} catch (Exception e) {

System.out.println(e);

}

return conn;

}

public void display() {

try {

conn = connect();

listM.clear();

PreparedStatement pst = conn.prepareStatement("select \*from reordertbl");

ResultSet r = pst.executeQuery();

while (r.next()) {

listM.add(new tblreorder(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category"),r.getTimestamp("order\_date")));

tbl.setItems(listM);

}

} catch (Exception e) {

System.out.println(e);

}

}

public void load()

{

mid.setCellValueFactory(new PropertyValueFactory<>("mid"));

mname.setCellValueFactory(new PropertyValueFactory<>("mname"));

qty.setCellValueFactory(new PropertyValueFactory<>("qty"));

rate.setCellValueFactory(new PropertyValueFactory<>("rate"));

mfg\_date.setCellValueFactory(new PropertyValueFactory<>("mfg\_date"));

exp\_date.setCellValueFactory(new PropertyValueFactory<>("exp\_date"));

batch.setCellValueFactory(new PropertyValueFactory<>("batch"));

company\_name.setCellValueFactory(new PropertyValueFactory<>("company\_name"));

amt.setCellValueFactory(new PropertyValueFactory<>("amt"));

category.setCellValueFactory(new PropertyValueFactory<>("category"));

order\_date.setCellValueFactory(new PropertyValueFactory<>("order\_date"));

}

public void remove(String b)

{

int p = Integer.parseInt(b);

try {

if (cr.reOrderDelete(p)>0) {

} else {

System.out.println("Try Again");

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

@FXML

private void btnReturn(ActionEvent event) {

if (tbl.getSelectionModel().getSelectedItems().size()==0) {

JOptionPane.showMessageDialog(null, "Please Select the Items.","Null Selection",JOptionPane.WARNING\_MESSAGE);

}

else{

try{

da = tbl.getSelectionModel().getSelectedItems();

Viewbody.getScene().getWindow().hide();

FXMLLoader loader = new FXMLLoader(getClass().getResource("Dashboard.fxml"));

Parent root = loader.load();

DashboardController d = loader.getController();

FXMLLoader loading = new FXMLLoader(getClass().getResource("returnStock.fxml"));

Parent returnContent = loading.load();

ReturnStockController r = loading.getController();

if (r == null) {

System.out.println("null");

} else {

r.redisplayed(da);

}

apn.getChildren().add(returnContent);

apn.prefHeightProperty().bind(d.pn2.heightProperty());

d.pn2.setContent(apn);

Stage primaryStage = new Stage();

Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("dashboard\_design.css").toExternalForm());

primaryStage.setScene(scene);

primaryStage.setMaximized(true);

primaryStage.show();

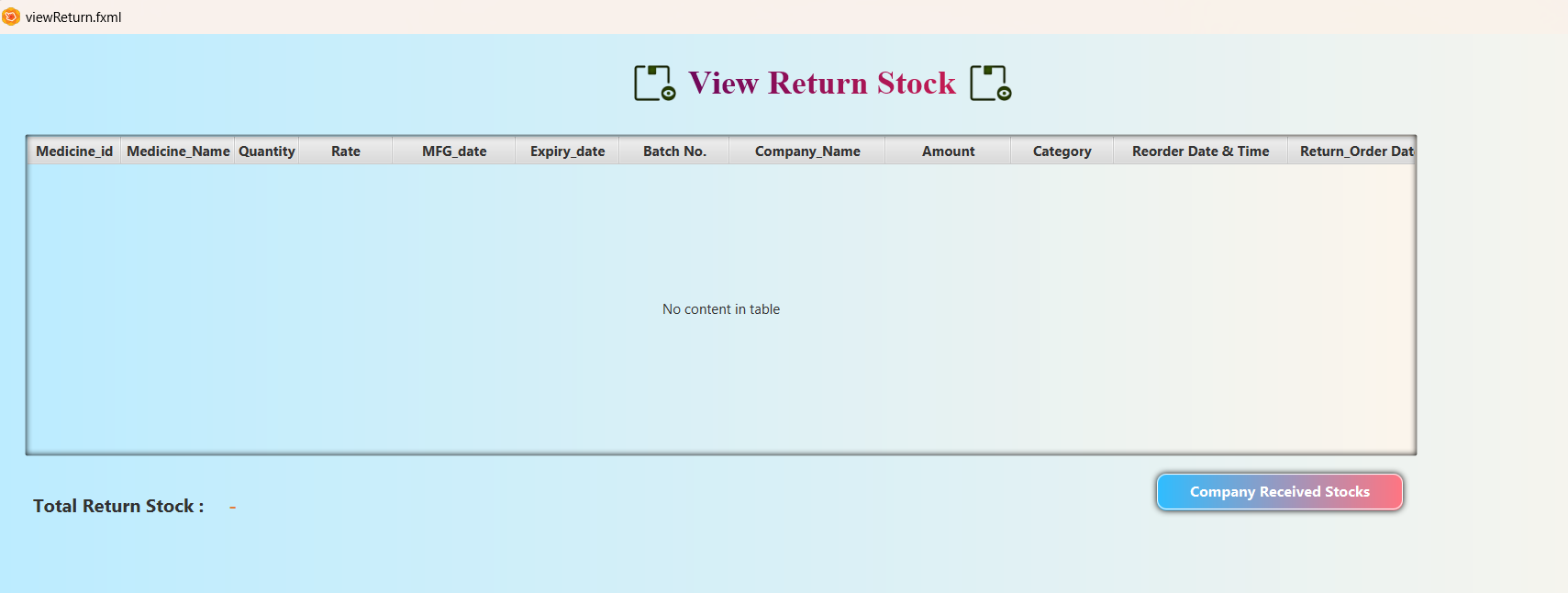
}catch(Exception e)

{System.out.println(e); }

}

}

* **viewreturn.fxml :**



* **viewreturncontroller.java :**

package projectfx;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Label;

import javafx.scene.control.SelectionMode;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TablePosition;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class ViewReturnController implements Initializable {

@FXML

private AnchorPane body;

@FXML

private TableView<tblreturn>tbl;

@FXML

private TableColumn<tblreturn,Integer> mid;

@FXML

private TableColumn<tblreturn,String> mname;

@FXML

private TableColumn<tblreturn,Integer> qty;

@FXML

private TableColumn<tblreturn,Float> rate;

@FXML

private TableColumn<tblreturn,Date> mfg\_date;

@FXML

private TableColumn<tblreturn,Date> exp\_date;

@FXML

private TableColumn<tblreturn,String> batch;

@FXML

private TableColumn<tblreturn,String> company\_name;

@FXML

private TableColumn<tblreturn,Float> amt;

@FXML

private TableColumn<tblreturn,String> category;

@FXML

private TableColumn<tblreorder,Timestamp> order\_date;

@FXML

private TableColumn<tblreorder,Timestamp> return\_date;

@FXML

private Label total;

crud cr = new crud();

Connection conn;

ObservableList<tblreturn> listM = FXCollections.observableArrayList();

ObservableList<tblreturn> data = FXCollections.observableArrayList();

ArrayList id = new ArrayList();

ArrayList name = new ArrayList();

ArrayList bat = new ArrayList();

ArrayList comp = new ArrayList();

ArrayList cate = new ArrayList();

ArrayList rat = new ArrayList();

ArrayList am = new ArrayList();

ArrayList qt = new ArrayList();

ArrayList or = new ArrayList();

ArrayList retu = new ArrayList();

ArrayList mfg = new ArrayList();

ArrayList exp = new ArrayList();

String nm,b,com,cat;

float t,a,g,r;

int q,idd;

Timestamp order,retun;

Date mfgd,expd;

int res = 0,ty = 0;

@Override

public void initialize(URL url, ResourceBundle rb) {

display();

load();

total.setText(String.valueOf(tbl.getItems().size()));

tbl.getSelectionModel().setSelectionMode(SelectionMode.MULTIPLE);

}

public Connection connect()

{

try {

Class.forName("com.mysql.jdbc.Driver");

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb","root","");

} catch (Exception e) {

System.out.println(e);

}

return conn;

}

public void inform(int i)

{

idd = Integer.parseInt(id.get(i).toString());

nm = name.get(i).toString();

q =Integer.parseInt(qt.get(i).toString());

r = Float.parseFloat(rat.get(i).toString());

mfgd = Date.valueOf(mfg.get(i).toString());

expd = Date.valueOf(exp.get(i).toString());

b = bat.get(i).toString();

com = comp.get(i).toString();

a = Float.parseFloat(am.get(i).toString());

cat = cate.get(i).toString();

order = Timestamp.valueOf(or.get(i).toString());

retun = Timestamp.valueOf(retu.get(i).toString());

}

public void selected()

{

try {

String path = "E:\\javaprograms\\medicalshop\\src\\images\\stockicon.png";

ImageIcon icon = new ImageIcon(path);

int n = name.size();

for (int i = 0; i < n; i++) {

inform(i);

if (cr.CompnayReturnStock(nm, q, r, mfgd, expd, b, com, cat, a, order, retun)>0) {

res++;

}

}

if (res>0) {

JOptionPane.showMessageDialog(null, "Company Successfully Received the Stock","ReturnStock Messages",JOptionPane.PLAIN\_MESSAGE,icon);

removeItems(n);

display();

total.setText(String.valueOf(tbl.getItems().size()));

tbl.getSelectionModel().clearSelection();

}

else{

JOptionPane.showMessageDialog(null, "Try Again","ReturnStock Messages",JOptionPane.WARNING\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"ReturnStock Messages",JOptionPane.ERROR\_MESSAGE);

}

}

public void removeItems(int n)

{

try {

for (int i = 0; i < n; i++) {

inform(i);

if (cr.ReturnStockDelete(idd)>0) {

} else {

System.out.println("Delete is not Done.");

}

}

} catch (Exception e) {

System.out.println(e);

}

}

public void display() {

try {

conn = connect();

listM.clear();

PreparedStatement pst = conn.prepareStatement("select \*from returnstock");

ResultSet r = pst.executeQuery();

while (r.next()) {

listM.add(new tblreturn(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category"),r.getTimestamp("order\_date"),r.getTimestamp("return\_date")));

tbl.setItems(listM);

}

} catch (Exception e) {

System.out.println(e);

}

}

public void load()

{

mid.setCellValueFactory(new PropertyValueFactory<>("mid"));

mname.setCellValueFactory(new PropertyValueFactory<>("mname"));

qty.setCellValueFactory(new PropertyValueFactory<>("qty"));

rate.setCellValueFactory(new PropertyValueFactory<>("rate"));

mfg\_date.setCellValueFactory(new PropertyValueFactory<>("mfg\_date"));

exp\_date.setCellValueFactory(new PropertyValueFactory<>("exp\_date"));

batch.setCellValueFactory(new PropertyValueFactory<>("batch"));

company\_name.setCellValueFactory(new PropertyValueFactory<>("company\_name"));

amt.setCellValueFactory(new PropertyValueFactory<>("amt"));

category.setCellValueFactory(new PropertyValueFactory<>("category"));

order\_date.setCellValueFactory(new PropertyValueFactory<>("order\_date"));

return\_date.setCellValueFactory(new PropertyValueFactory<>("return\_date"));

}

@FXML

private void btnDetails(ActionEvent event) {

if (tbl.getSelectionModel().getSelectedItems().size()==0) {

JOptionPane.showMessageDialog(null, "Please Select the Items.","Null Selection",JOptionPane.WARNING\_MESSAGE);

}

else{

try {

data = tbl.getSelectionModel().getSelectedItems();

selected();

} catch (Exception e) {

System.out.println(e);

}

}

}

@FXML

private void mouseClick(MouseEvent event) {

id.add(tbl.getSelectionModel().getSelectedItem().getMid());

name.add(tbl.getSelectionModel().getSelectedItem().getMname());

bat.add(tbl.getSelectionModel().getSelectedItem().getBatch());

comp.add(tbl.getSelectionModel().getSelectedItem().getCompany\_name());

cate.add(tbl.getSelectionModel().getSelectedItem().getCategory());

qt.add(tbl.getSelectionModel().getSelectedItem().getQty());

rat.add(tbl.getSelectionModel().getSelectedItem().getRate());

am.add(tbl.getSelectionModel().getSelectedItem().getAmt());

mfg.add(tbl.getSelectionModel().getSelectedItem().getMfg\_date());

exp.add(tbl.getSelectionModel().getSelectedItem().getExp\_date());

or.add(tbl.getSelectionModel().getSelectedItem().getOrder\_date());

retu.add(tbl.getSelectionModel().getSelectedItem().getReturn\_date());

}

}

* **Companyreturn.fxml :**



* **CompanyreturnController.java :**

package projectfx;

import java.sql.\*;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

public class CompanyReturnController implements Initializable {

@FXML

private TableView<tblreturn>tbl;

@FXML

private TableColumn<tblreturn,Integer> mid;

@FXML

private TableColumn<tblreturn,String> mname;

@FXML

private TableColumn<tblreturn,Integer> qty;

@FXML

private TableColumn<tblreturn,Float> rate;

@FXML

private TableColumn<tblreturn,Date> mfg\_date;

@FXML

private TableColumn<tblreturn,Date> exp\_date;

@FXML

private TableColumn<tblreturn,String> batch;

@FXML

private TableColumn<tblreturn,String> company\_name;

@FXML

private TableColumn<tblreturn,Float> amt;

@FXML

private TableColumn<tblreturn,String> category;

@FXML

private TableColumn<tblreorder,Timestamp> order\_date;

@FXML

private TableColumn<tblreorder,Timestamp> return\_date;

Connection conn;

ObservableList<tblreturn> listM = FXCollections.observableArrayList();

@Override

public void initialize(URL url, ResourceBundle rb) {

display();

load();

}

public void load()

{

mid.setCellValueFactory(new PropertyValueFactory<>("mid"));

mname.setCellValueFactory(new PropertyValueFactory<>("mname"));

qty.setCellValueFactory(new PropertyValueFactory<>("qty"));

rate.setCellValueFactory(new PropertyValueFactory<>("rate"));

mfg\_date.setCellValueFactory(new PropertyValueFactory<>("mfg\_date"));

exp\_date.setCellValueFactory(new PropertyValueFactory<>("exp\_date"));

batch.setCellValueFactory(new PropertyValueFactory<>("batch"));

company\_name.setCellValueFactory(new PropertyValueFactory<>("company\_name"));

amt.setCellValueFactory(new PropertyValueFactory<>("amt"));

category.setCellValueFactory(new PropertyValueFactory<>("category"));

order\_date.setCellValueFactory(new PropertyValueFactory<>("order\_date"));

return\_date.setCellValueFactory(new PropertyValueFactory<>("return\_date"));

}

public Connection connect()

{

try {

Class.forName("com.mysql.jdbc.Driver");

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb","root","");

} catch (Exception e) {

System.out.println(e);

}

return conn;

}

public void display() {

try {

conn = connect();

listM.clear();

PreparedStatement pst = conn.prepareStatement("select \*from companyreceived");

ResultSet r = pst.executeQuery();

while (r.next()) {

listM.add(new tblreturn(

r.getInt("mid"), r.getString("mname"),

r.getInt("qty"), r.getFloat("rate"),

r.getDate("mfg\_date"), r.getDate("exp\_date"),

r.getString("batch"), r.getString("company\_name"), r.getFloat("amt"),

r.getString("category"),r.getTimestamp("order\_date"),r.getTimestamp("return\_date")));

tbl.setItems(listM);

}

} catch (Exception e) {

System.out.println(e);

}

}

}

* **displaymedicines.fxml :**

****

* **displaymedicinesController.java :**

package projectfx;

import java.net.URL;

import java.sql.Connection;

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.TableCell;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import static projectfx.InvoicefxController.no;

public class DisplaymedicinesController implements Initializable {

@FXML

private AnchorPane medicine\_panel;

Connection c;

@FXML

private TableView<setvariables> tableview1;

@FXML

private TableColumn<setvariables, String> custnm;

@FXML

private TableColumn<setvariables, String> mno;

@FXML

private TableColumn<setvariables, Integer> drs;

@FXML

private TableColumn<setvariables, Float> gsti;

@FXML

private TableColumn<setvariables, Float> tam;

@FXML

private TableColumn<setvariables, Integer> ino;

@FXML

private TableColumn<setvariables, String> payment;

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

getdatelist();

display();

}

//display tableview

ObservableList<setvariables> getdatelist() {

ObservableList<setvariables> invoicelist = FXCollections.observableArrayList();

try {

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

PreparedStatement p = c.prepareStatement("select \*from invoicebill");

ResultSet rs = p.executeQuery();

setvariables s;

while (rs.next()) {

s = new setvariables(rs.getInt("invoice\_no"), rs.getString("customer\_nm"), rs.getString("mobile\_no"), rs.getInt("discount"), rs.getFloat("gst"), rs.getFloat("total\_amount"),rs.getString("payment\_method"));

invoicelist.add(s);

}

} catch (Exception e) {

System.out.println(e);

}

return invoicelist;

}

public void display() {

try {

ObservableList<setvariables> list = getdatelist();

ino.setCellValueFactory(new PropertyValueFactory<setvariables, Integer>("ino"));

custnm.setCellValueFactory(new PropertyValueFactory<setvariables, String>("custnm"));

mno.setCellValueFactory(new PropertyValueFactory<setvariables, String>("mno"));

drs.setCellValueFactory(new PropertyValueFactory<setvariables, Integer>("drs"));

gsti.setCellValueFactory(new PropertyValueFactory<setvariables, Float>("gst"));

tam.setCellValueFactory(new PropertyValueFactory<setvariables, Float>("tam"));

payment.setCellValueFactory(new PropertyValueFactory<setvariables, String>("payment"));

tableview1.setItems(list);

} catch (Exception e) {

System.out.println(e + "not display");

}

}

private void btnback(ActionEvent event) {

try {

AnchorPane pane = FXMLLoader.load(getClass().getResource("displaymedicines.fxml"));

medicine\_panel.getScene().getWindow().hide();

Stage stage = new Stage();

Parent p = FXMLLoader.load(getClass().getResource("invoicefx.fxml"));

Scene s = new Scene(p);

s.getStylesheets().add(getClass().getResource("purchasecss.css").toExternalForm());

stage.setScene(s);

stage.show();

System.out.println("click");

} catch (Exception e) {

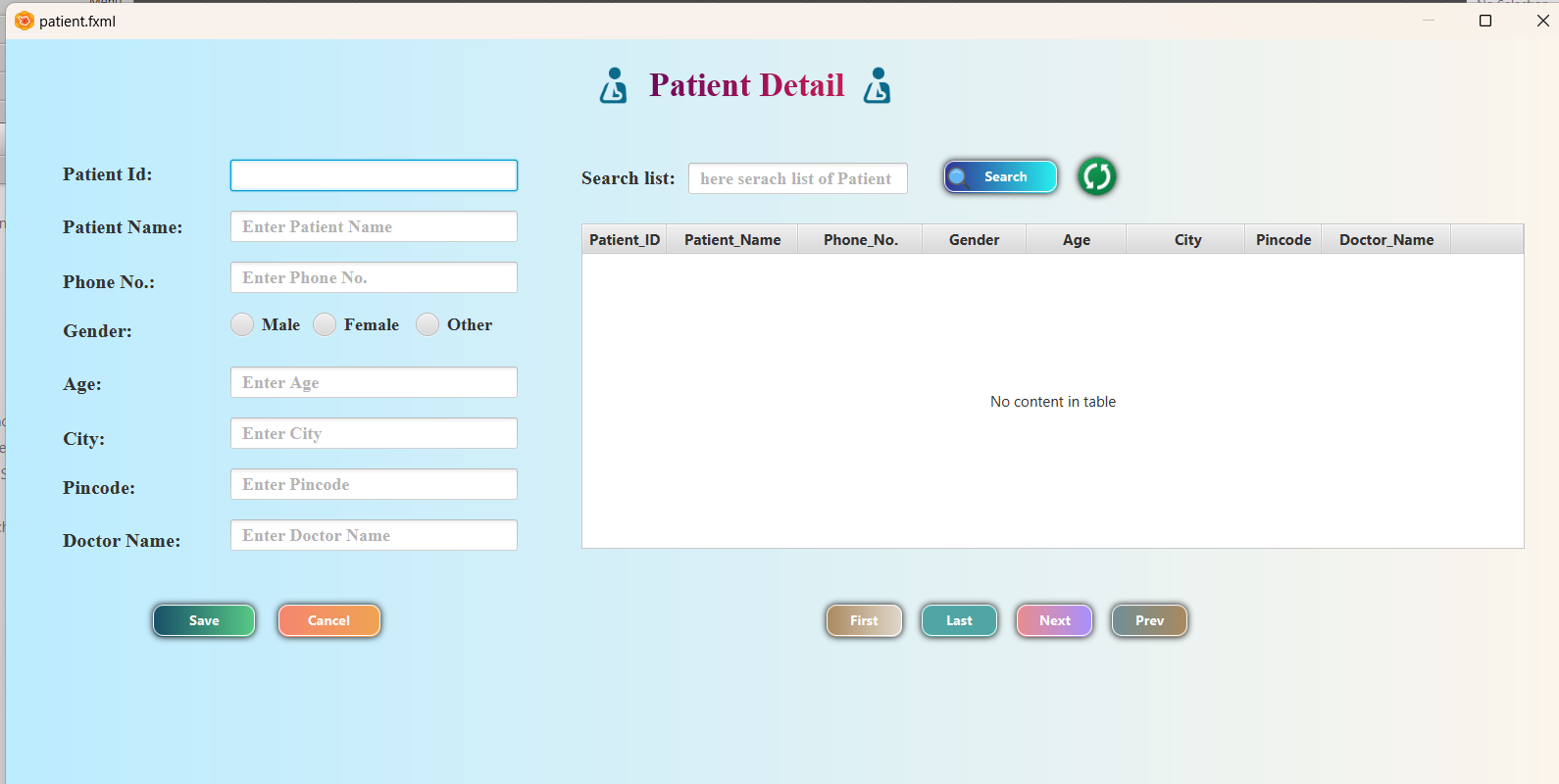
System.out.println(e);

}

}

}

* **patient.fxml :**



* **patientController.java :**

package projectfx;

import java.sql.\*;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.image.ImageView;

import javafx.scene.input.KeyEvent;

import javafx.scene.input.MouseDragEvent;

import javafx.scene.input.MouseEvent;

import javax.swing.JOptionPane;

public class PatientController implements Initializable {

@FXML

private Button btnsave;

@FXML

private Button btncancel;

@FXML

private TextField txtpatientid;

@FXML

private TextField txtpatientnm;

@FXML

private TextField txtphoneno;

@FXML

private RadioButton radiomale;

@FXML

private RadioButton radiofemale;

@FXML

private RadioButton radioother;

@FXML

private TextField txtage;

@FXML

private TextField txtcity;

@FXML

private TextField txtpincode;

@FXML

private TextField txtdoctornm;

@FXML

private TextField txtsearch;

@FXML

private Button btnsearch;

@FXML

private Button btnfirst;

@FXML

private Button btnlast;

@FXML

private Button btnnext;

@FXML

private Button btnprev;

@FXML

private TableView<setpatient> tableview;

@FXML

private TableColumn<setpatient, Integer> pid;

@FXML

private TableColumn<setpatient, String> pnm;

@FXML

private TableColumn<setpatient, String> pno;

@FXML

private TableColumn<setpatient, String> gender;

@FXML

private TableColumn<setpatient, Integer> age;

@FXML

private TableColumn<setpatient, String> city;

@FXML

private TableColumn<setpatient, Integer> pincode;

@FXML

private TableColumn<setpatient, String> dnm;

int id,ag,pin;

String name,phoneno,g,ct,doctor;

Connection c;

@FXML

private ImageView ref;

@FXML

private ToggleGroup gen;

@FXML

private TableColumn<String, Button> edit;

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

getdatelist();

display();

}

public void clear() {

txtpatientid.setText(null);

txtpatientnm.setText(null);

txtphoneno.setText(null);

txtage.setText(null);

txtcity.setText(null);

txtpincode.setText(null);

txtdoctornm.setText(null);

}

ObservableList<setpatient> getdatelist() {

ObservableList<setpatient> addlist = FXCollections.observableArrayList();

try {

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

PreparedStatement p = c.prepareStatement("select \*from pastient");

ResultSet rs = p.executeQuery();

setpatient s;

while (rs.next()) {

s = new setpatient(rs.getInt("pid"), rs.getString("pnm"), rs.getString("mobileno"), rs.getString("gender"),

rs.getInt("age"), rs.getString("city"), rs.getInt("pincode"), rs.getString("doctornm"));

addlist.add(s);

}

} catch (Exception e) {

System.out.println(e);

}

return addlist;

}

public void display() {

try {

ObservableList<setpatient> list = getdatelist();

pid.setCellValueFactory(new PropertyValueFactory<setpatient, Integer>("pid"));

pnm.setCellValueFactory(new PropertyValueFactory<setpatient, String>("pnm"));

pno.setCellValueFactory(new PropertyValueFactory<setpatient, String>("mobileno"));

gender.setCellValueFactory(new PropertyValueFactory<setpatient, String>("gender"));

age.setCellValueFactory(new PropertyValueFactory<setpatient, Integer>("age"));

city.setCellValueFactory(new PropertyValueFactory<setpatient, String>("city"));

pincode.setCellValueFactory(new PropertyValueFactory<setpatient, Integer>("pincode"));

dnm.setCellValueFactory(new PropertyValueFactory<setpatient, String>("doctonm"));

edit.setCellFactory(col -> new TableCell<String, Button>() {

@Override

protected void updateItem(Button obj, boolean b) {

super.updateItem(obj, b);

if (b) {

setGraphic(null);

} else {

Button btn = new Button();

btn.setPrefSize(30, 30);

btn.setId("buttonedit");

setGraphic(btn);

btn.setOnAction((event) -> {

try {

update();

} catch (Exception e) {

System.out.println(e);

}

});

}

}

});

tableview.setItems(list);

} catch (Exception e) {

System.out.println(e + "not display");

}

}

ObservableList<setpatient> getdatasearch() {

ObservableList<setpatient> search = FXCollections.observableArrayList();

String se = txtsearch.getText();

try {

Class.forName("com.mysql.jdbc.Driver");

c = DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb", "root", "");

PreparedStatement p = null;

if (Character.isAlphabetic(se.charAt(0)) || Character.isAlphabetic(4)) {

p = c.prepareStatement("select \*from pastient where pnm like '%" + se + "%' or gender like '%" + se + "%' or city like '%" + se + "%' or doctornm like '%" + se + "%'");

} else if (Character.isDigit(se.charAt(0))) {

p = c.prepareStatement("select \*from pastient where pid=" + se);

// + "or qty="+se+" or rate="+se+" or amo="+se+"or mno="+se+" or drs="+se+" or gst="+se+" or tam="+se);

}

ResultSet rs = p.executeQuery();

setpatient s;

while (rs.next()) {

s = new setpatient(rs.getInt("pid"), rs.getString("pnm"), rs.getString("mobileno"), rs.getString("gender"),

rs.getInt("age"), rs.getString("city"), rs.getInt("pincode"), rs.getString("doctornm"));

search.add(s);

}

} catch (Exception e) {

System.out.println(e);

}

return search;

}

public void search() {

try {

ObservableList<setpatient> l = getdatasearch();

pid.setCellValueFactory(new PropertyValueFactory<setpatient, Integer>("pid"));

pnm.setCellValueFactory(new PropertyValueFactory<setpatient, String>("pnm"));

pno.setCellValueFactory(new PropertyValueFactory<setpatient, String>("mobileno"));

gender.setCellValueFactory(new PropertyValueFactory<setpatient, String>("gender"));

age.setCellValueFactory(new PropertyValueFactory<setpatient, Integer>("age"));

city.setCellValueFactory(new PropertyValueFactory<setpatient, String>("city"));

pincode.setCellValueFactory(new PropertyValueFactory<setpatient, Integer>("pincode"));

dnm.setCellValueFactory(new PropertyValueFactory<setpatient, String>("doctonm"));

tableview.setItems(l);

} catch (Exception e) {

System.out.println(e + "not display");

}

}

public void update()

{

try {

id = Integer.parseInt(txtpatientid.getText());

name = txtpatientnm.getText();

phoneno = txtphoneno.getText();

if (radiomale.isSelected()) {

g = radiomale.getText();

} else if (radiofemale.isSelected()) {

g = radiofemale.getText();

} else {

g = radioother.getText();

}

ag = Integer.parseInt(txtage.getText());

ct = txtcity.getText();

pin = Integer.parseInt(txtpincode.getText());

doctor = txtdoctornm.getText();

iobj obj = new iobj();

System.out.println(id+" "+name);

if (obj.updatepatient(id, name, phoneno, g ,ag, ct, pin, doctor) > 0) {

System.out.println(id+" "+name);

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, "update record", "update", JOptionPane.INFORMATION\_MESSAGE);

display();

// clear();

} else {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, "try again", "try", JOptionPane.ERROR\_MESSAGE);

}

} catch (Exception e) {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, e, "try", JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void btnsave(ActionEvent event) {

try {

id = Integer.parseInt(txtpatientid.getText());

name = txtpatientnm.getText();

phoneno = txtphoneno.getText();

if (radiomale.isSelected()) {

g = radiomale.getText();

} else if (radiofemale.isSelected()) {

g = radiofemale.getText();

} else {

g = radioother.getText();

}

ag = Integer.parseInt(txtage.getText());

ct = txtcity.getText();

pin = Integer.parseInt(txtpincode.getText());

doctor = txtdoctornm.getText();

iobj obj = new iobj();

if (obj.insertpatientdata(id, name, phoneno, g, ag, ct, pin, doctor) > 0) {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, "insert record", "insert", JOptionPane.INFORMATION\_MESSAGE);

display();

// clear();

} else {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, "try again", "try", JOptionPane.ERROR\_MESSAGE);

}

} catch (Exception e) {

Component rootPane = null;

JOptionPane.showMessageDialog(rootPane, e, "try", JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void btncancel(ActionEvent event) {

clear();

}

@FXML

private void btnsearch(ActionEvent event) {

search();

}

@FXML

private void btnfirstt(ActionEvent event) {

TableView.TableViewSelectionModel<setpatient> s = tableview.getSelectionModel();

s.selectFirst();

}

@FXML

private void btnlast(ActionEvent event) {

TableView.TableViewSelectionModel<setpatient> s = tableview.getSelectionModel();

s.selectLast();

}

@FXML

private void btnnext(ActionEvent event) {

TableView.TableViewSelectionModel<setpatient> s = tableview.getSelectionModel();

s.selectNext();

}

@FXML

private void btnprev(ActionEvent event) {

TableView.TableViewSelectionModel<setpatient> s = tableview.getSelectionModel();

s.selectPrevious();

}

@FXML

private void btnref(MouseEvent event) {

display();

}

@FXML

private void mouseClick(MouseEvent event) {

txtpatientid.setText(String.valueOf(tableview.getSelectionModel().getSelectedItem().getPid()));

txtpatientnm.setText(tableview.getSelectionModel().getSelectedItem().getPnm());

txtphoneno.setText(tableview.getSelectionModel().getSelectedItem().getMobileno());

if(radiomale.getText().equals(tableview.getSelectionModel().getSelectedItem().getGender()))

{

radiomale.setSelected(true);

}

else if(radiofemale.getText().equals(tableview.getSelectionModel().getSelectedItem().getGender()))

{

radiofemale.setSelected(true);

}

else

{

radioother.setSelected(true);

}

txtage.setText(String.valueOf(tableview.getSelectionModel().getSelectedItem().getAge()));

txtcity.setText(tableview.getSelectionModel().getSelectedItem().getCity());

txtpincode.setText(String.valueOf(tableview.getSelectionModel().getSelectedItem().getPincode()));

txtdoctornm.setText(tableview.getSelectionModel().getSelectedItem().getDoctonm());

}

}

* **Company.fxml :**



* **CompanyController.java :**

package projectfx;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.Label;

import javafx.scene.control.TextArea;

import javafx.scene.control.TextField;

import javafx.scene.layout.VBox;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class CompanyController implements Initializable {

@FXML

private Label title;

@FXML

private TextField name;

@FXML

private TextArea addr;

@FXML

private TextField city;

@FXML

private TextField pin;

int p;

String nm,address,cityy;

crud cr = new crud();

@FXML

private VBox vb;

@Override

public void initialize(URL url, ResourceBundle rb) {

// TODO

}

public void inform()

{

nm = name.getText();

address = addr.getText();

cityy = city.getText();

p = Integer.parseInt(pin.getText());

}

public void clear()

{

name.setText(null);

addr.setText(null);

city.setText(null);

pin.setText(null);

}

@FXML

private void btnCancel(ActionEvent event) {

clear();

}

@FXML

private void btnAdd(ActionEvent event) {

String imagepath = "C:\\Users\\poojajungi\\Downloads\\check.png";

ImageIcon icon = new ImageIcon(imagepath);

try {

inform();

if (cr.companyAdd(nm, address, cityy, p)>0) {

JOptionPane.showMessageDialog(null, "Company Add Successfully.","Company Details Message",JOptionPane.PLAIN\_MESSAGE,icon);

clear();

} else {

JOptionPane.showMessageDialog(null, "Try Again.","Company Details Message",JOptionPane.WARNING\_MESSAGE);

}

} catch (Exception e) {

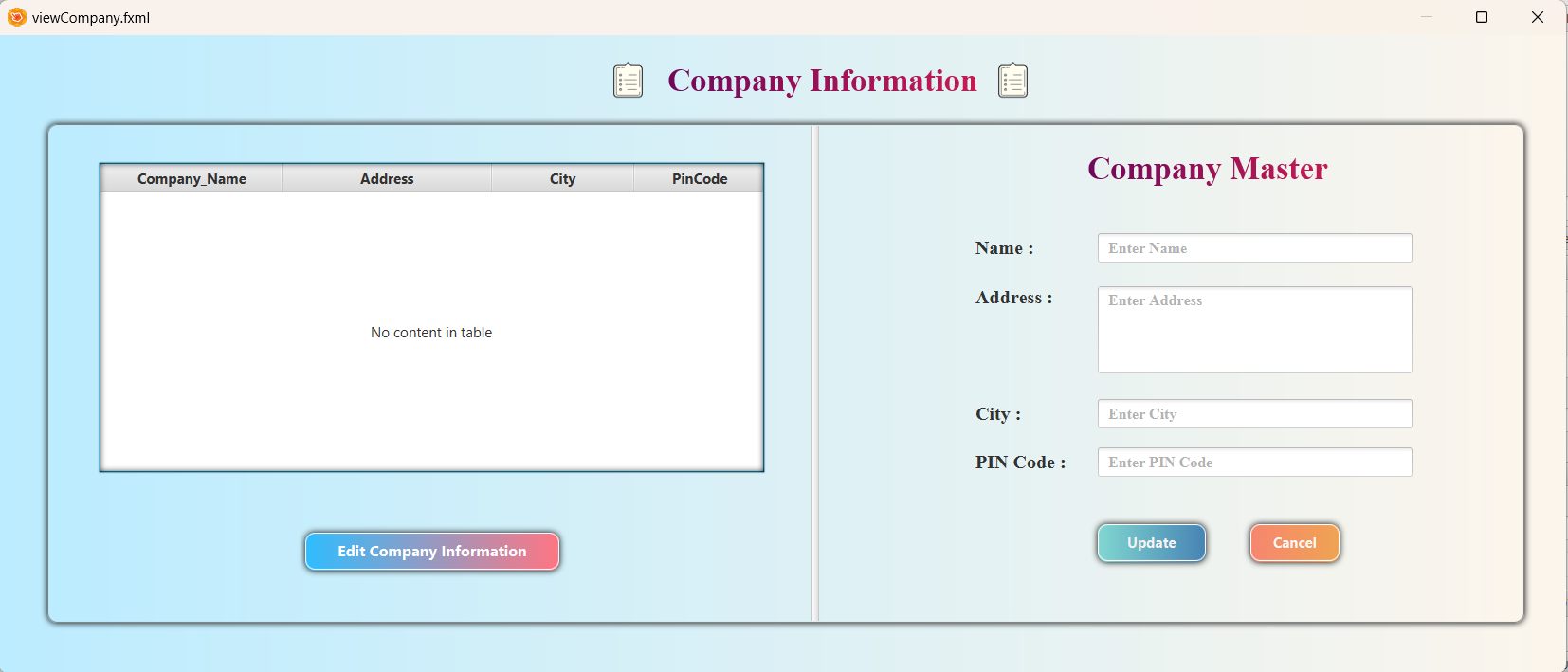
JOptionPane.showMessageDialog(null, e,"Company Details Message",JOptionPane.ERROR\_MESSAGE);

}

}

}

* **ViewCompany.fxml:**



* **ViewCompanyController.java:**

package projectfx;

import java.net.URL;

import java.sql.\*;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.AnchorPane;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class ViewCompanyController implements Initializable {

@FXML

private Label title;

@FXML

private Label title1;

@FXML

private TextField name;

@FXML

private TextArea addr;

@FXML

private TextField pin;

@FXML

private TextField citytxt;

@FXML

private TableView<tblcompany> tbl;

@FXML

private TableColumn<tblcompany,String> city;

@FXML

private TableColumn<tblcompany,String> cname;

@FXML

private TableColumn<tblcompany,String> caddress;

@FXML

private TableColumn<tblcompany,Integer> pincode;

crud cr = new crud();

Connection conn;

ObservableList<tblcompany> list = FXCollections.observableArrayList();

int p;

String nm,addre,ci;

@FXML

private AnchorPane panel2;

@Override

public void initialize(URL url, ResourceBundle rb) {

display();

load();

}

@FXML

private void btnDetails(ActionEvent event) {

panel2.setVisible(true);

name.setText(tbl.getSelectionModel().getSelectedItem().getCname());

addr.setText(tbl.getSelectionModel().getSelectedItem().getCaddress());

citytxt.setText(tbl.getSelectionModel().getSelectedItem().getCity());

pin.setText(String.valueOf(tbl.getSelectionModel().getSelectedItem().getPincode()));

tbl.getSelectionModel().clearSelection();

}

@FXML

private void btnAdd(ActionEvent event) {

try {

inform();

String imagepath = "C:\\Users\\poojajungi\\Downloads\\check.png";

ImageIcon icon = new ImageIcon(imagepath);

if (cr.companyUpdate(nm, addre, ci, p)>0) {

JOptionPane.showMessageDialog(null, "Company Updated Successfully.","View Company Message",JOptionPane.PLAIN\_MESSAGE,icon);

clear();

display();

panel2.setVisible(false);

} else {

JOptionPane.showMessageDialog(null, "Try Again.","View Company Message",JOptionPane.WARNING\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"View Company Message",JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void btnCancel(ActionEvent event) {

clear();

}

public void display()

{

try {

list.clear();

conn = connect();

PreparedStatement pst =conn.prepareStatement("select \*from company");

ResultSet r = pst.executeQuery();

while (r.next()) {

list.add(new tblcompany(r.getString("cname"),

r.getString("caddress"),

r.getString("city"),

r.getInt("pincode")));

}

tbl.setItems(list);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Display Messages",JOptionPane.ERROR\_MESSAGE);

}

}

public void load()

{

cname.setCellValueFactory(new PropertyValueFactory<>("cname"));

caddress.setCellValueFactory(new PropertyValueFactory<>("caddress"));

city.setCellValueFactory(new PropertyValueFactory<>("city"));

pincode.setCellValueFactory(new PropertyValueFactory<>("pincode"));

}

public Connection connect()

{

try {

Class.forName("com.mysql.jdbc.Driver");

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb","root","");

} catch (Exception e) {

System.out.println(e);

}

return conn;

}

public void inform()

{

nm = name.getText();

addre = addr.getText();

ci = citytxt.getText();

p = Integer.parseInt(pin.getText());

}

public void clear()

{

name.setText(null);

addr.setText(null);

citytxt.setText(null);

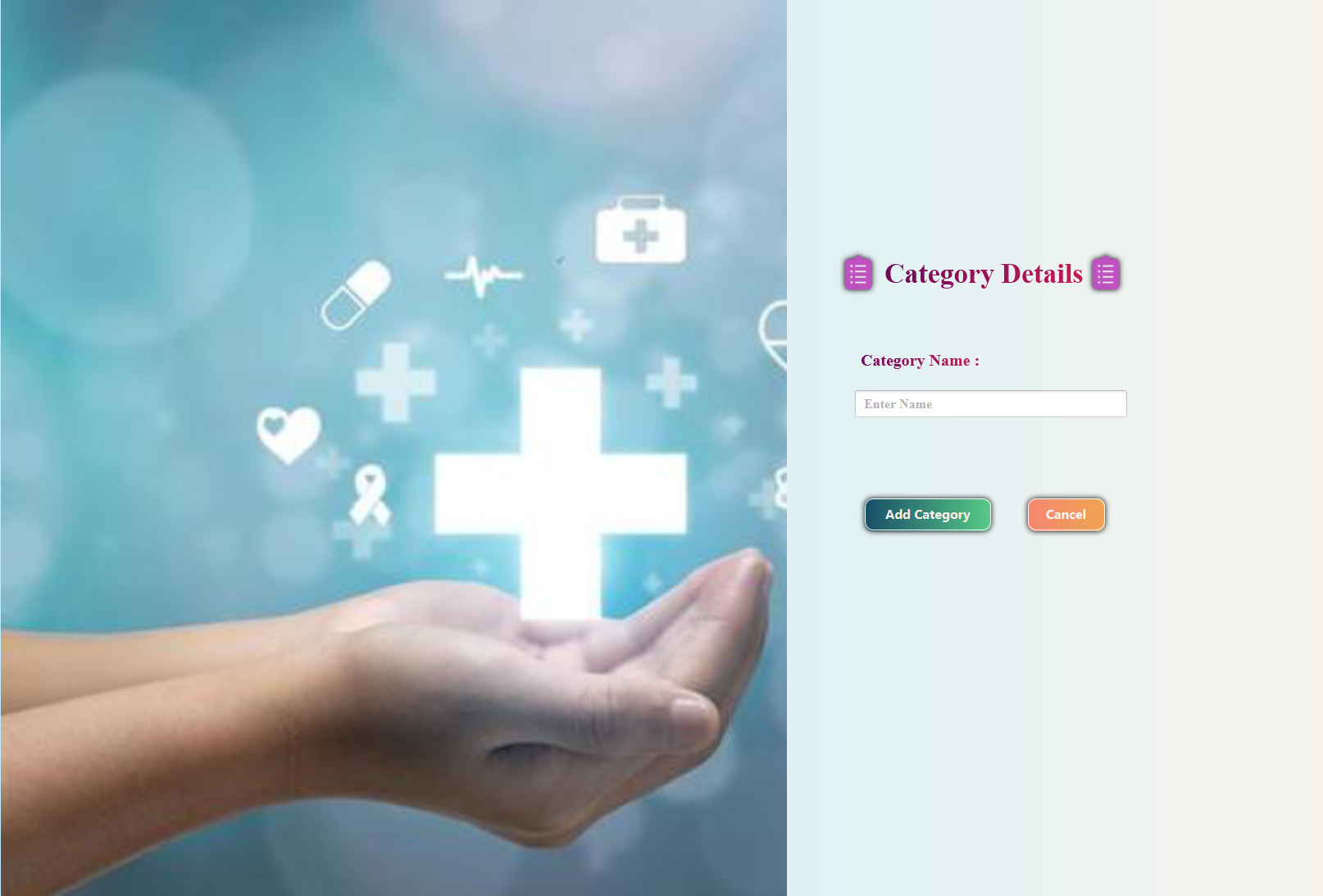
pin.setText(null);

tbl.getSelectionModel().clearSelection();

}

}

* **Category.fxml:**



* **CategoryController.java:**

package projectfx;

import java.net.URL;

import java.util.ResourceBundle;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.\*;

import javafx.scene.layout.VBox;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class CategoryController implements Initializable {

@FXML

private Label title;

@FXML

private TextField name;

@FXML

private VBox vb;

String nm,pro;

crud cr = new crud();

public void inform()

{

nm = name.getText();

}

public void clear()

{

name.setText(null);

}

@Override

public void initialize(URL url, ResourceBundle rb) {

}

@FXML

private void btnAdd(ActionEvent event) {

String imagepath = "C:\\Users\\poojajungi\\Downloads\\check.png";

ImageIcon icon = new ImageIcon(imagepath);

try {

inform();

if (cr.categoryAdd(nm)>0) {

JOptionPane.showMessageDialog(null, "New Category Added Successfully.","Category Messages",JOptionPane.PLAIN\_MESSAGE,icon);

clear();

} else {

JOptionPane.showMessageDialog(null, "Try Again.","Category Message",JOptionPane.WARNING\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"Company Details Message",JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

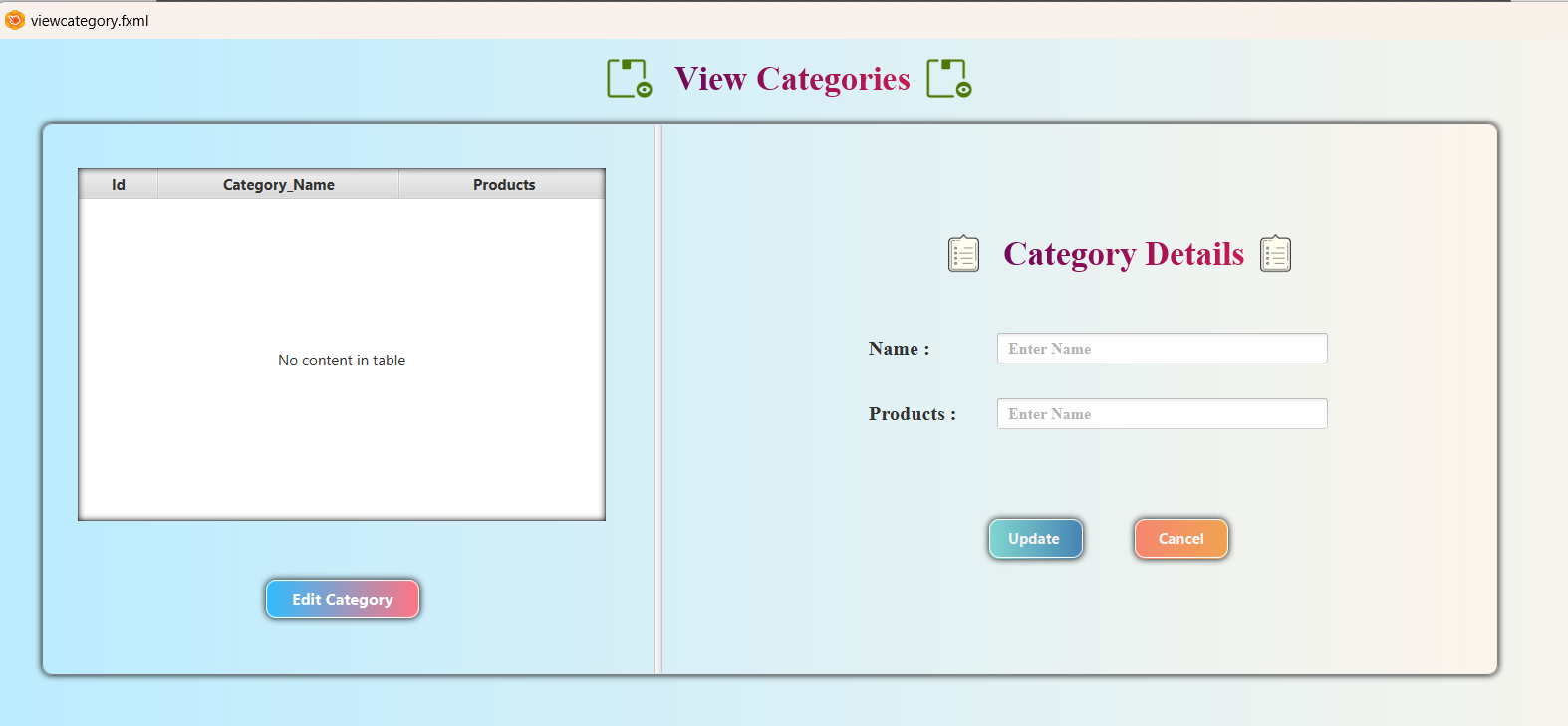
private void btnCancel(ActionEvent event) {

clear();

}

}

* **ViewCategory.fxml:**



* **ViewCategoryController.java :**

package projectfx;

import java.io.IOException;

import java.net.URL;

import java.sql.\*;

import java.util.ResourceBundle;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.input.MouseEvent;

import javafx.scene.layout.AnchorPane;

import javafx.stage.Stage;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class ViewcategoryController implements Initializable {

@FXML

private Label title;

Connection conn;

ObservableList<tblcategory>list = FXCollections.observableArrayList();

int c=0;

String nm ,pro;

crud cr = new crud();

private Label total;

@FXML

private AnchorPane panel2;

@FXML

private TableView<tblcategory> tbl;

@FXML

private TableColumn<tblcategory, Integer> pid;

@FXML

private TextField name;

@FXML

private TextField product;

@FXML

private TableColumn<tblcategory,String> category\_name;

@FXML

private TableColumn<tblcategory,String> product\_name;

@Override

public void initialize(URL url, ResourceBundle rb) {

display();

load();

}

public Connection connect()

{

try {

Class.forName("com.mysql.jdbc.Driver");

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/jtb","root","");

} catch (Exception e) {

System.out.println(e);

}

return conn;

}

public void display()

{

try {

list.clear();

conn = connect();

PreparedStatement pst = conn.prepareStatement("select p.pid,c.category\_name,p.product\_name from categorytbl c,productstbl p where c.id=p.category\_id");

ResultSet r = pst.executeQuery();

while (r.next()) {

list.add(new tblcategory(r.getInt("pid"),

r.getString("category\_name"),

r.getString("product\_name")));

c++;

}

tbl.setItems(list);

} catch (Exception e) {

System.out.println(e);

}

}

public void load()

{

pid.setCellValueFactory(new PropertyValueFactory<>("pid"));

category\_name.setCellValueFactory(new PropertyValueFactory<>("category\_name"));

product\_name.setCellValueFactory(new PropertyValueFactory<>("product\_name"));

}

public void inform()

{

nm = name.getText();

pro = product.getText();

}

public void clear()

{

name.setText(null);

product.setText(null);

}

@FXML

private void btnAdd(ActionEvent event) {

String imagepath = "C:\\Users\\poojajungi\\Downloads\\check.png";

ImageIcon icon = new ImageIcon(imagepath);

try {

inform();

if (cr.CategoryUpdate(nm, pro)>0) {

JOptionPane.showMessageDialog(null, "Category Updated Successfully.","View Category Message",JOptionPane.PLAIN\_MESSAGE,icon);

clear();

display();

panel2.setVisible(false);

} else {

JOptionPane.showMessageDialog(null, "Try Again.","View Category Message",JOptionPane.WARNING\_MESSAGE);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e,"View Category Message",JOptionPane.ERROR\_MESSAGE);

}

}

@FXML

private void btnCancel(ActionEvent event) {

clear();

}

@FXML

private void btnDetails(ActionEvent event) throws IOException {

panel2.setVisible(true);

name.setText(tbl.getSelectionModel().getSelectedItem().getCategory\_name());

product.setText(tbl.getSelectionModel().getSelectedItem().getProduct\_name());

}

}