# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 9

1) Write a program to demonstrate GUI of a Login page using JAVA and networking

LOGIN PAGE

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.OutputStream;

import java.net.HttpURLConnection;

import java.net.URL;

public class LoginPage extends JFrame {

private JLabel labelUsername, labelPassword;

private JTextField textFieldUsername;

private JPasswordField passwordField;

private JButton buttonLogin, buttonReset;

public LoginPage() {

setTitle("Login Page");

setSize(350, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null); // Center the frame

labelUsername = new JLabel("Username:");

labelPassword = new JLabel("Password:");

textFieldUsername = new JTextField(15);

passwordField = new JPasswordField(15);

buttonLogin = new JButton("Login");

buttonReset = new JButton("Reset");

setLayout(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(5, 5, 5, 5);

gbc.gridx = 0;

gbc.gridy = 0;

add(labelUsername, gbc);

gbc.gridx = 1;

add(textFieldUsername, gbc);

gbc.gridx = 0;

gbc.gridy = 1;

add(labelPassword, gbc);

gbc.gridx = 1;

add(passwordField, gbc);

gbc.gridx = 0;

gbc.gridy = 2;

add(buttonLogin, gbc);

gbc.gridx = 1;

add(buttonReset, gbc);

buttonLogin.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String username = textFieldUsername.getText();

String password = new String(passwordField.getPassword());

try {

boolean loginSuccess = sendLoginRequest(username, password);

if (loginSuccess) {

JOptionPane.showMessageDialog(null, "Login Successful!");

} else {

JOptionPane.showMessageDialog(null, "Invalid Username or Password", "Error", JOptionPane.ERROR\_MESSAGE);

}

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, "Network Error: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

});

buttonReset.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

textFieldUsername.setText("");

passwordField.setText("");

}

});

}

private boolean sendLoginRequest(String username, String password) throws Exception {

String urlString = <http://localhost:8080/login>;

URL url = new URL(urlString);

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

connection.setRequestMethod("POST");

connection.setRequestProperty("Content-Type", "application/json");

connection.setDoOutput(true);

String jsonInputString = "{\"username\": \"" + username + "\", \"password\": \"" + password + "\"}";

try (OutputStream os = connection.getOutputStream()) {

byte[] input = jsonInputString.getBytes("utf-8");

os.write(input, 0, input.length);

}

int responseCode = connection.getResponseCode();

return responseCode == 200;

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new LoginPage().setVisible(true);

}

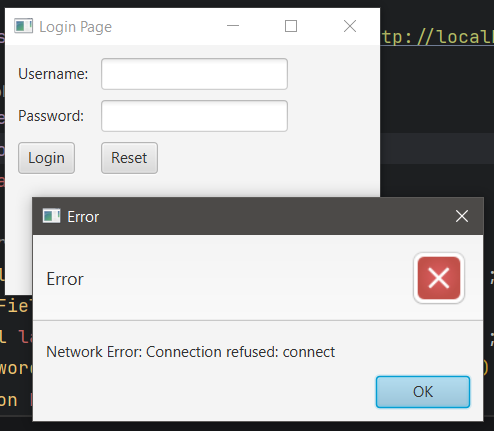
});

}

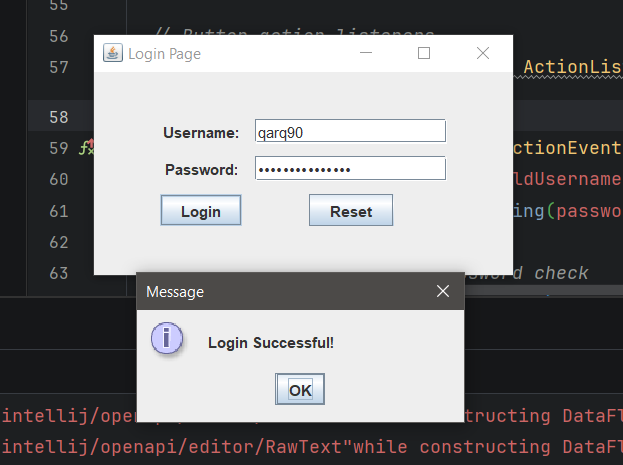
}

OUTPUT

IF NO CONNECTION

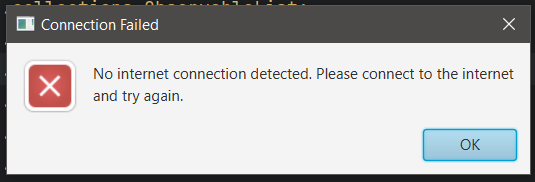


IF CONNECTION PROGRAM EXECUTES AS NEEDED



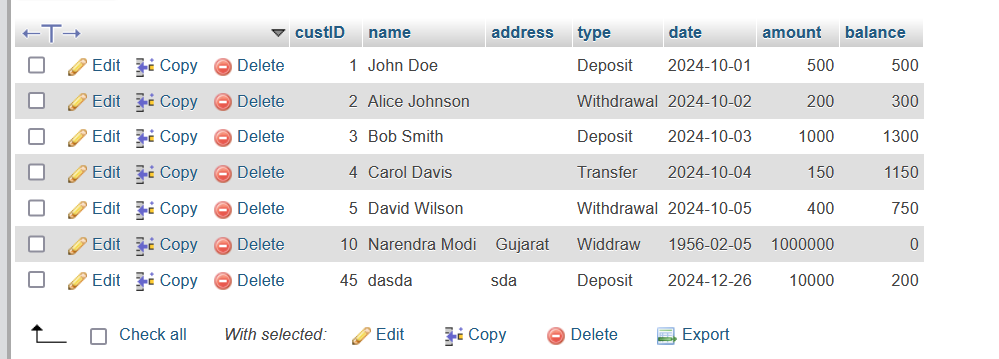
2) Write a program to demonstrate GUI of a banking application using JAVA with networking

NO CONNECTION

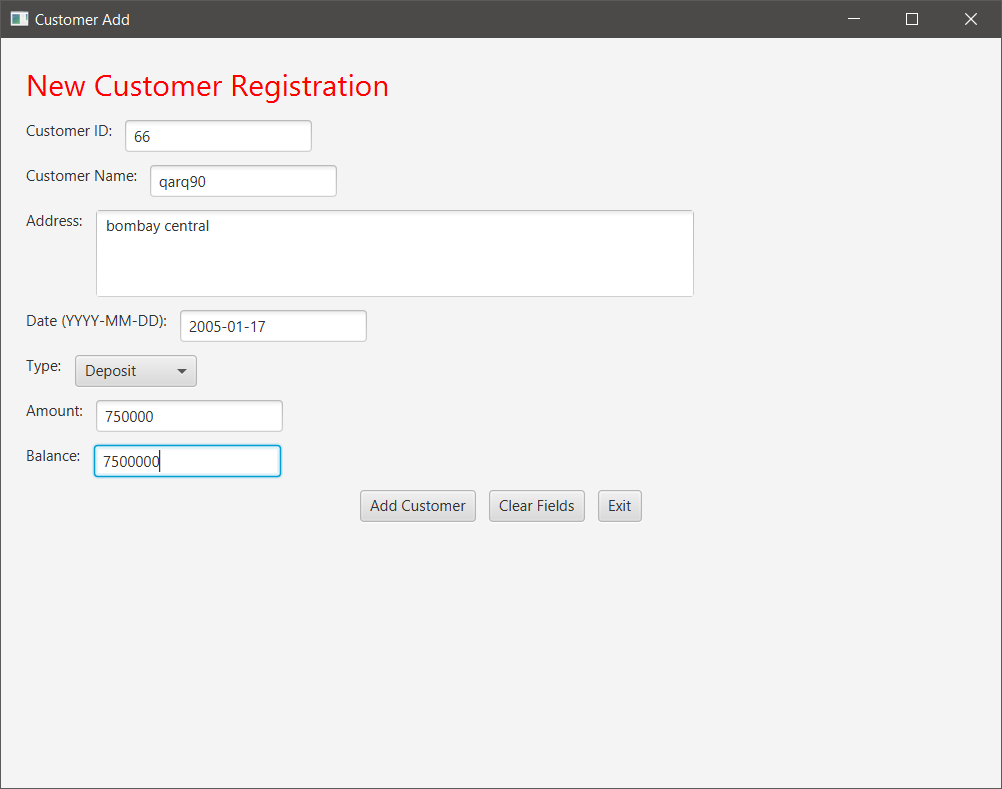


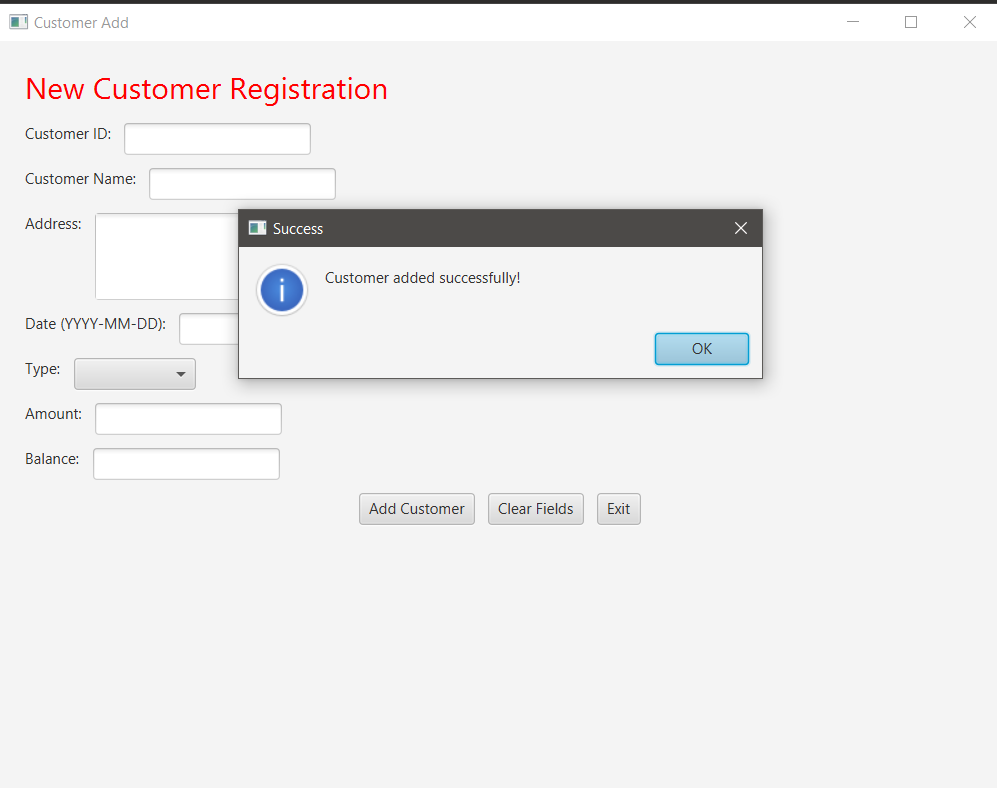
IF CONNECTION PROGRAM EXECUTES AS NEEDED

ORIGINAL DATABASE TABLE

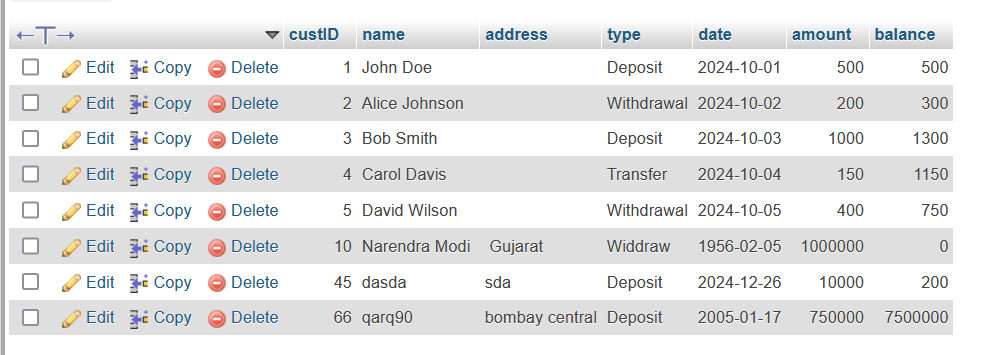


ADDING A CUSTOMER





AFTER ADDING - ORIGINAL DATABASE TABLE



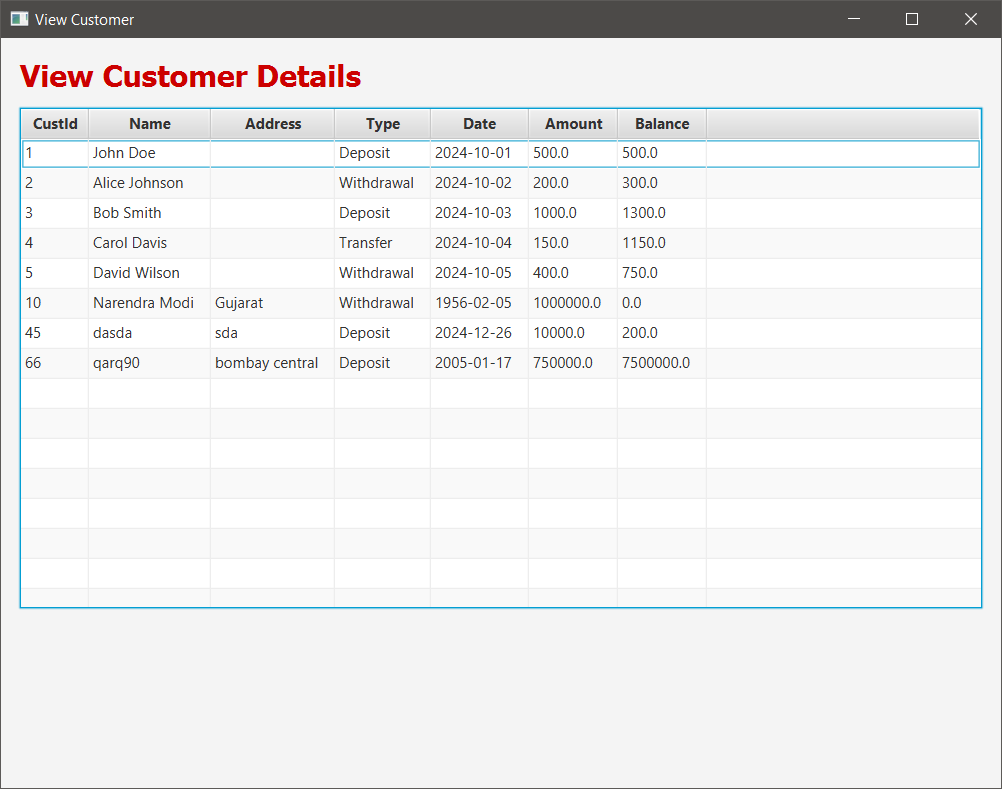
ADDING A CUSTOMER (CODE)

private void showCustomerAdd(Stage primaryStage) {  
 BorderPane customerAddPane = new BorderPane();  
  
 Label jLabel1 = new Label("Customer ID:");  
 TextField txtid = new TextField();  
  
 Label jLabel2 = new Label("Customer Name:");  
 TextField txtname = new TextField();  
  
 Label jLabel3 = new Label("Address:");  
 TextArea jTextArea1 = new TextArea();  
 jTextArea1.setPrefRowCount(3);  
 jTextArea1.setWrapText(true);  
  
 Label jLabel4 = new Label("Date (YYYY-MM-DD):");  
 TextField txtdate = new TextField();  
  
 Label jLabel5 = new Label("Type:");  
 ComboBox<String> lsttype = new ComboBox<>();  
 lsttype.getItems().addAll("Deposit", "Withdraw");  
  
 Label jLabel6 = new Label("Amount:");  
 TextField txtamount = new TextField();  
  
 Label jLabel7 = new Label("Balance:");  
 TextField txtbalance = new TextField();  
  
 Button cmdadd = new Button("Add Customer");  
 Button cmdclear = new Button("Clear Fields");  
 Button cmdexit = new Button("Exit");  
  
 cmdadd.setOnAction(e -> {  
 addCustomer(txtid, txtname, jTextArea1, txtdate, lsttype, txtamount, txtbalance);  
 });  
 cmdclear.setOnAction(e -> clearFields(txtid, txtname, jTextArea1, txtdate, lsttype, txtamount, txtbalance));  
 cmdexit.setOnAction(e -> primaryStage.close());  
  
 VBox inputPane = new VBox(10);  
 inputPane.setPadding(new Insets(20));  
 inputPane.getChildren().addAll(  
 new Label("New Customer Registration") {{  
 setStyle("-fx-font-size: 24; -fx-text-fill: red;");  
 }},  
 new HBox(10, jLabel1, txtid),  
 new HBox(10, jLabel2, txtname),  
 new HBox(10, jLabel3, jTextArea1),  
 new HBox(10, jLabel4, txtdate),  
 new HBox(10, jLabel5, lsttype),  
 new HBox(10, jLabel6, txtamount),  
 new HBox(10, jLabel7, txtbalance),  
 new HBox(10, cmdadd, cmdclear, cmdexit) {{  
 setAlignment(javafx.geometry.Pos.CENTER);  
 }}  
 );  
  
 customerAddPane.setCenter(inputPane);  
  
 Scene customerAddScene = new Scene(customerAddPane, 800, 600);  
 Stage customerAddStage = new Stage();  
 customerAddStage.setTitle("Customer Add");  
 customerAddStage.setScene(customerAddScene);  
 customerAddStage.show();  
}  
  
private void addCustomer(TextField txtid, TextField txtname, TextArea jTextArea1, TextField txtdate,  
 ComboBox<String> lsttype, TextField txtamount, TextField txtbalance) {  
 try {  
 Class.forName("com.mysql.cj.jdbc.Driver");  
 Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/dbms", "root", "");  
 PreparedStatement ps = con.prepareStatement("INSERT INTO bank VALUES (?, ?, ?, ?, ?, ?, ?)");  
 ps.setInt(1, Integer.parseInt(txtid.getText()));  
 ps.setString(2, txtname.getText());  
 ps.setString(3, jTextArea1.getText());  
 ps.setString(4, txtdate.getText());  
 ps.setString(5, lsttype.getValue());  
 ps.setDouble(6, Double.parseDouble(txtamount.getText()));  
 ps.setDouble(7, Double.parseDouble(txtbalance.getText()));  
 ps.executeUpdate();  
 clearFields(txtid, txtname, jTextArea1, txtdate, lsttype, txtamount, txtbalance);  
 showAlert("Success", "Customer added successfully!");  
 } catch (Exception e) {  
 showAlert("Error", "Could not add customer: " + e.getMessage());  
 e.printStackTrace();  
 }  
}  
  
private void clearFields(TextField txtid, TextField txtname, TextArea jTextArea1, TextField txtdate,  
 ComboBox<String> lsttype, TextField txtamount, TextField txtbalance) {  
 txtid.clear();  
 txtname.clear();  
 jTextArea1.clear();  
 txtdate.clear();  
 lsttype.setValue(null);  
 txtamount.clear();  
 txtbalance.clear();  
}  
  
private void showAlert(String title, String message) {  
 Alert alert = new Alert(Alert.AlertType.INFORMATION);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
}

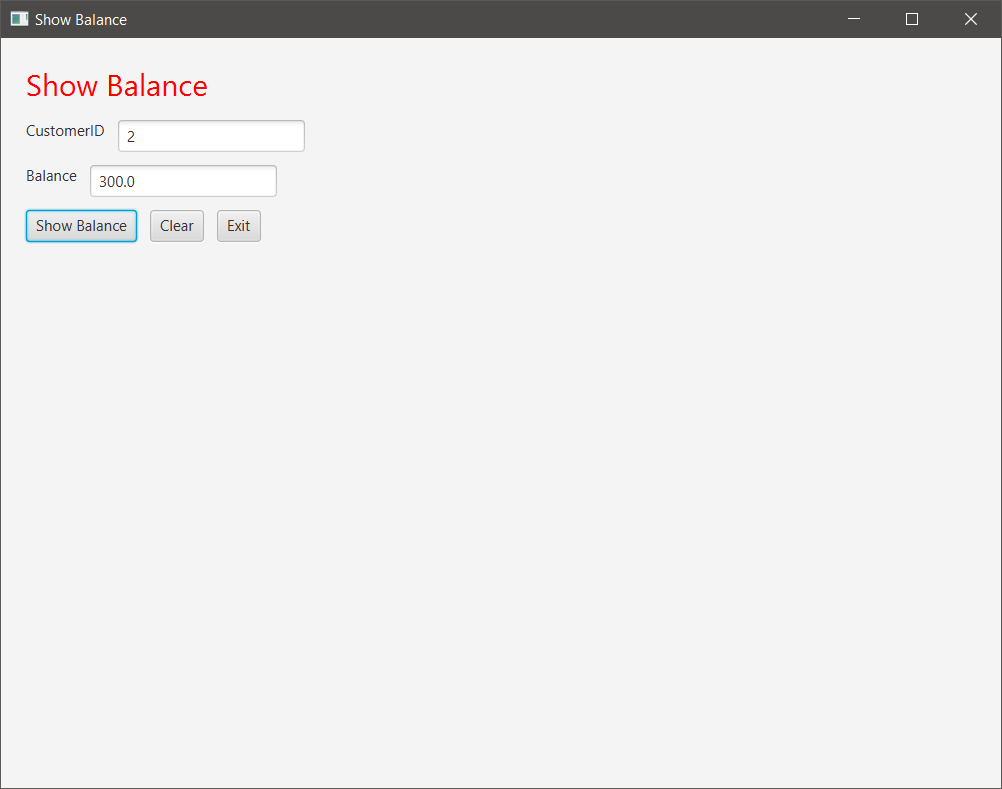
VIEWING CUSTOMERS (CODE)

private void showViewCustomer(Stage primaryStage) {  
 Label lblTitle = new Label("View Customer Details");  
 lblTitle.setFont(Font.*font*("Tahoma", FontWeight.*BOLD*, 24));  
 lblTitle.setTextFill(Color.*rgb*(204, 0, 0));  
  
 TableView<Customer> tableView = new TableView<>();  
  
 TableColumn<Customer, Integer> colCustId = new TableColumn<>("CustId");  
 colCustId.setCellValueFactory(cellData -> new SimpleIntegerProperty(cellData.getValue().getCustId()).asObject());  
  
 TableColumn<Customer, String> colName = new TableColumn<>("Name");  
 colName.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue().getName()));  
  
 TableColumn<Customer, String> colAddress = new TableColumn<>("Address");  
 colAddress.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue().getAddress()));  
  
 TableColumn<Customer, String> colType = new TableColumn<>("Type");  
 colType.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue().getType()));  
  
 TableColumn<Customer, String> colDate = new TableColumn<>("Date");  
 colDate.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue().getDate()));  
  
 TableColumn<Customer, Double> colAmount = new TableColumn<>("Amount");  
 colAmount.setCellValueFactory(cellData -> new SimpleDoubleProperty(cellData.getValue().getAmount()).asObject());  
  
 TableColumn<Customer, Double> colBalance = new TableColumn<>("Balance");  
 colBalance.setCellValueFactory(cellData -> new SimpleDoubleProperty(cellData.getValue().getBalance()).asObject());  
  
 tableView.getColumns().addAll(colCustId, colName, colAddress, colType, colDate, colAmount, colBalance);  
  
 ObservableList<Customer> data = FXCollections.*observableArrayList*(  
 new Customer(1, "John Doe", "", "Deposit", "2024-10-01", 500.0, 500.0),  
 new Customer(2, "Alice Johnson", "", "Withdrawal", "2024-10-02", 200.0, 300.0),  
 new Customer(3, "Bob Smith", "", "Deposit", "2024-10-03", 1000.0, 1300.0),  
 new Customer(4, "Carol Davis", "", "Transfer", "2024-10-04", 150.0, 1150.0),  
 new Customer(5, "David Wilson", "", "Withdrawal", "2024-10-05", 400.0, 750.0),  
 new Customer(10, "Narendra Modi", "Gujarat", "Withdrawal", "1956-02-05", 1000000.0, 0.0),  
 new Customer(45, "dasda", "sda", "Deposit", "2024-12-26", 10000.0, 200.0),  
 new Customer(66, "qarq90", "bombay central", "Deposit", "2005-01-17", 750000.0, 7500000.0)  
 );  
  
 tableView.setItems(data);  
  
 VBox vbox = new VBox(10, lblTitle, tableView);  
 vbox.setPadding(new Insets(15));  
  
 Scene scene = new Scene(vbox, 800, 600);  
 primaryStage.setScene(scene);  
 primaryStage.setTitle("View Customer");  
 primaryStage.show();  
}

VIEWING CUSTOMERS



VIEWING CUSTOMERS’S BALANCE



VIEWING CUSTOMERS’S BALANCE (CODE)

private void showShowBalance(Stage primaryStage) {  
 VBox showBalanceLayout = new VBox(10);  
 showBalanceLayout.setPadding(new Insets(20));  
  
 Label titleLabel = new Label("Show Balance");  
 titleLabel.setStyle("-fx-font-size: 24; -fx-text-fill: red;");  
  
 Label jLabel1 = new Label("CustomerID");  
 TextField txtcuid = new TextField();  
  
 Label jLabel2 = new Label("Balance");  
 TextField txtbal = new TextField();  
 txtbal.setEditable(false);  
  
 Button jButton1 = new Button("Show Balance");  
 Button jButton2 = new Button("Clear");  
 Button jButton3 = new Button("Exit");  
  
 jButton1.setOnAction(e -> {  
 try {  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/dbms", "root", "");  
 Statement stmt = con.createStatement();  
 ResultSet rs = stmt.executeQuery("SELECT \* FROM bank WHERE custid=" + *txtcuid*.getText());  
 double cbalance = 0;  
 while (rs.next()) {  
 cbalance = rs.getDouble(7);  
 }  
 *txtbal*.setText("" + cbalance);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 });  
  
 jButton2.setOnAction(e -> {  
 *txtcuid*.setText("");  
 *txtbal*.setText("");  
 });  
  
 jButton3.setOnAction(e -> *primaryStage*.close());  
  
 showBalanceLayout.getChildren().addAll(titleLabel,  
 new HBox(10, jLabel1, txtcuid),  
 new HBox(10, jLabel2, txtbal),  
 new HBox(10, jButton1, jButton2, jButton3));  
  
 Scene showBalanceScene = new Scene(showBalanceLayout, 800, 600);  
 Stage showBalanceStage = new Stage();  
 showBalanceStage.setTitle("Show Balance");  
 showBalanceStage.setScene(showBalanceScene);  
 showBalanceStage.show();  
}