Login

Main.java

package application;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.stage.Stage;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.layout.BorderPane;

public class Main extends Application {

*@Override*

public void start(Stage primaryStage) {

try {

Parent root = FXMLLoader.*load*(getClass().getResource("/application/Login.fxml"));

Scene scene = new Scene(root);

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

primaryStage.setTitle("S'Cool Store");

primaryStage.setScene(scene);

primaryStage.show();

} catch(Exception e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

*launch*(args);

}

}

Login

MainController.java

package application;

import java.io.IOException;

import java.net.URL;

import java.util.ResourceBundle;

import com.sun.glass.ui.Window;

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.Label;

import javafx.scene.control.TextField;

import javafx.stage.Stage;

public class MainController implements Initializable {

*@FXML*

private Label lblStatus;

*@FXML*

private TextField txtUsername;

*@FXML*

private TextField txtPassword;

public void Login(ActionEvent event) throws IOException {

if (txtUsername.getText().equals("admin") && txtPassword.getText().equals("pass")) {

lblStatus.setText("Login Success!");

((Node)event.getSource()).getScene().getWindow().hide();

Parent root = FXMLLoader.*load*(getClass().getResource("/system/Main.fxml"));

Stage primaryStage = new Stage();

Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("/system/application.css").toExternalForm());

primaryStage.setTitle("S'Cool Store");

primaryStage.setScene(scene);

primaryStage.show();

}

else {

lblStatus.setText("Login Failed, Please Try Again.");

}

}

*@Override*

public void initialize(URL arg0, ResourceBundle arg1) {

}

}

Add Product, Transaction, Sales

MainController.java

package system;

import java.io.IOException;

import java.net.URL;

import java.time.LocalDate;

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.TableColumn;

import javafx.scene.control.TextField;

import javafx.scene.control.Alert;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

import java.util.ResourceBundle;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.DatePicker;

import javafx.scene.control.Label;

import javafx.scene.control.TableView;

public class MainController implements Initializable {

public TableView<Product> tableview;

public TableView<Product> tableview2;

public TableView<Transaction> tableview3;

public TableView<Sales> tableview4;

public TableColumn<Product,String> colName;

public TableColumn<Product,Integer> colStock;

public TableColumn<Product,Double> colPrice;

public TableColumn<Product,String> colName1;

public TableColumn<Product,Integer> colStock1;

public TableColumn<Product,Double> colPrice1;

public TextField textfieldName;

public TextField textfieldStock;

public TextField textfieldPrice;

public TableColumn<Transaction,Integer> colcustomerNo;

public TableColumn<Transaction,String> colproductName;

public TableColumn<Transaction,Integer> colproductQuantity;

public TableColumn<Transaction,Double> colproductTotalPrice;

public TextField textfieldName1;

public TextField textfieldStock1;

public TextField textfieldPrice1;

public Label labelcustomerNo;

public Label labelTotalPrice;

public TextField textfieldQuantity;

public DatePicker dp;

public Label showdatalbl;

public Label labelTotalPrice1;

public TableColumn<Sales, Integer> colsalescustomerNo;

public TableColumn<Sales, Double> colsalestotalPrice;

public TableColumn<Sales, LocalDate> colsalesDate;

*@Override*

public void initialize(URL arg0, ResourceBundle arg1) {

colName.setCellValueFactory(new PropertyValueFactory<>("ProductName"));

colStock.setCellValueFactory(new PropertyValueFactory<>("ProductStock"));

colPrice.setCellValueFactory(new PropertyValueFactory<>("ProductPrice"));

colName1.setCellValueFactory(new PropertyValueFactory<>("ProductName"));

colStock1.setCellValueFactory(new PropertyValueFactory<>("ProductStock"));

colPrice1.setCellValueFactory(new PropertyValueFactory<>("ProductPrice"));

colcustomerNo.setCellValueFactory(new PropertyValueFactory<>("transacCustomerNo"));

colproductName.setCellValueFactory(new PropertyValueFactory<>("transacName"));

colproductQuantity.setCellValueFactory(new PropertyValueFactory<>("transacQuantity"));

colproductTotalPrice.setCellValueFactory(new PropertyValueFactory<>("transacTotalPrice"));

colsalescustomerNo.setCellValueFactory(new PropertyValueFactory<>("SalesCustomerNo"));

colsalestotalPrice.setCellValueFactory(new PropertyValueFactory<>("SalesTotalPrice"));

colsalesDate.setCellValueFactory(new PropertyValueFactory<>("SalesDate"));

tableview.setItems(observableList);

setCellValueFromTabletoLabel();

}

ObservableList<Product> observableList = FXCollections.*observableArrayList*(

);

public void About (ActionEvent event) throws IOException {

Parent root = FXMLLoader.*load*(getClass().getResource("/system/About.fxml"));

Stage primaryStage = new Stage();

Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("/system/application.css").toExternalForm());

primaryStage.setTitle("S'Cool Store");

primaryStage.setScene(scene);

primaryStage.show();

}

public void ShowDate(ActionEvent event) {

LocalDate Id = dp.getValue();

showdatalbl.setText(Id.toString());

}

public boolean Save () {

if (textfieldName.getText().isEmpty() | textfieldStock.getText().isEmpty() | textfieldPrice.getText().isEmpty()) {

Alert alert = new Alert(*AlertType*.***WARNING***);

alert.setTitle("Error!");

alert.setHeaderText(null);

alert.setContentText("Please Enter Into the Fields!");

alert.showAndWait();

return false;

}

Product product = new Product(textfieldName.getText(),Integer.*parseInt*(textfieldStock.getText()),Double.*parseDouble*(textfieldPrice.getText()));

tableview.getItems().add(product);

tableview2.getItems().add(product);

textfieldName.clear();

textfieldStock.clear();

textfieldPrice.clear();

return true;

}

public void Clear() {

textfieldName.clear();

textfieldStock.clear();

textfieldPrice.clear();

}

public void Edit() {

Product pl = tableview2.getSelectionModel().getSelectedItem();

textfieldName1.setText(pl.getProductName());

textfieldStock1.setText(pl.getProductStock().toString());

textfieldPrice1.setText(pl.getProductPrice().toString());

textfieldName1.setEditable(true);

textfieldStock1.setEditable(true);

textfieldPrice1.setEditable(true);

}

public void Update(ActionEvent event) {

Product product = new Product(textfieldName1.getText(),Integer.*parseInt*(textfieldStock1.getText()),Double.*parseDouble*(textfieldPrice1.getText()));

tableview2.getItems().set(tableview2.getSelectionModel().getSelectedIndex(), product);

textfieldName1.setText(null);

textfieldStock1.setText(null);

textfieldPrice1.setText(null);

}

public boolean Delete(ActionEvent actionEvent) {

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

Alert alert = new Alert(*AlertType*.***WARNING***);

alert.setTitle("Error!");

alert.setHeaderText(null);

alert.setContentText("Please select the item to delete");

alert.showAndWait();

return false;

}

ObservableList<Product> allProduct,SingleProduct;

allProduct = tableview.getItems();

SingleProduct = tableview.getSelectionModel().getSelectedItems();

SingleProduct.forEach(allProduct::remove);

return true;

}

public void Delete1(ActionEvent actionEvent) {

ObservableList<Product> allProduct,SingleProduct;

allProduct = tableview2.getItems();

SingleProduct = tableview2.getSelectionModel().getSelectedItems();

SingleProduct.forEach(allProduct::remove);

}

//transaction

public void setCellValueFromTabletoLabel() {

tableview2.setOnMouseClicked(e -> {

Product pl = tableview2.getItems().get(tableview2.getSelectionModel().getSelectedIndex());

textfieldName1.setText(pl.getProductName());

textfieldStock1.setText(Integer.*valueOf*(pl.getProductStock()).toString());

textfieldPrice1.setText(pl.getProductPrice().toString());

textfieldName1.setEditable(false);

textfieldStock1.setEditable(false);

textfieldPrice1.setEditable(false);

});

}

ObservableList <Transaction> observableList1 = FXCollections.*observableArrayList*(

);

public boolean Total() {

//if (textfieldQuantity.getText().compareTo(textfieldStock1.getText())>0) {

if (Integer.*parseInt*(textfieldQuantity.getText()) > Integer.*parseInt*(textfieldStock1.getText())) {

Alert alert = new Alert(*AlertType*.***WARNING***);

alert.setTitle("Error!");

alert.setHeaderText(null);

alert.setContentText("Quantity is greater than Stock, Please add the Stock");

alert.showAndWait();

textfieldQuantity.clear();

return false;

}

int a = Integer.*parseInt*(textfieldStock1.getText());

double b = Double.*parseDouble*(textfieldPrice1.getText());

int c = Integer.*parseInt*(textfieldQuantity.getText());

int d = a - c;

double e = b \* c;

textfieldStock1.setText(Integer.*valueOf*(d).toString());

labelTotalPrice.setText(Double.*valueOf*(e).toString());

return true;

}

public void AddTransaction(ActionEvent actionEvent) {

Product product = new Product(textfieldName1.getText(),Integer.*parseInt*(textfieldStock1.getText()),Double.*parseDouble*(textfieldPrice1.getText()));

tableview2.getItems().set(tableview2.getSelectionModel().getSelectedIndex(), product);

Transaction transaction = new Transaction(Integer.*parseInt*(labelcustomerNo.getText()),textfieldName1.getText(),Integer.*parseInt*(textfieldQuantity.getText()), Double.*parseDouble*(labelTotalPrice.getText()) );

tableview3.getItems().add(transaction);

textfieldName1.setText(null);

textfieldStock1.setText(null);

textfieldPrice1.setText(null);

textfieldQuantity.setText(null);

double sum = 0.00;

for (Transaction item : tableview3.getItems()) {

sum = sum + item.getTransacTotalPrice();

labelTotalPrice1.setText(Double.*valueOf*(sum).toString());

}

}

public void Buy (ActionEvent actionEvent) {

Sales sales = new Sales(Integer.*parseInt*(labelcustomerNo.getText()), Double.*parseDouble*(labelTotalPrice1.getText()), LocalDate.*parse*(showdatalbl.getText())) ;

tableview4.getItems().add(sales);

textfieldName1.clear();

textfieldStock1.clear();

textfieldPrice1.clear();

textfieldQuantity.clear();

labelTotalPrice.setText(null);

labelTotalPrice1.setText(null);

String s = labelcustomerNo.getText();

int IdNumber = Integer.*valueOf*(s);

s = Integer.*toString*(++IdNumber);

labelcustomerNo.setText(s);

tableview3.getItems().clear();

}

}

Product.java

package system;

import javafx.beans.property.SimpleDoubleProperty;

import javafx.beans.property.SimpleIntegerProperty;

import javafx.beans.property.SimpleStringProperty;

public class Product {

private SimpleStringProperty productName;

private SimpleIntegerProperty productStock;

private SimpleDoubleProperty productPrice;

public Product(String productName,Integer productStock, Double productPrice) {

this.productName = new SimpleStringProperty (productName);

this.productStock = new SimpleIntegerProperty (productStock);

this.productPrice = new SimpleDoubleProperty (productPrice);

}

public String getProductName() {

return productName.get();

}

public void setProductName(String productName) {

this.productName =new SimpleStringProperty (productName);

}

public Integer getProductStock() {

return productStock.get();

}

public void setProductStock(Integer productStock) {

this.productStock = new SimpleIntegerProperty(productStock);

}

public Double getProductPrice() {

return productPrice.get();

}

public void setProductPrice(Double productPrice) {

this.productPrice = new SimpleDoubleProperty(productPrice);

}

}

Transaction.java

package system;

import javafx.beans.property.SimpleDoubleProperty;

import javafx.beans.property.SimpleIntegerProperty;

import javafx.beans.property.SimpleStringProperty;

public class Transaction {

private SimpleIntegerProperty transacCustomerNo;

private SimpleStringProperty transacName;

private SimpleIntegerProperty transacQuantity;

private SimpleDoubleProperty transacTotalPrice;

public Transaction(Integer transacCustomerNo, String transacName,Integer transacQuantity, Double transacTotalPrice) {

this.transacCustomerNo = new SimpleIntegerProperty(transacCustomerNo);

this.transacName = new SimpleStringProperty (transacName);

this.transacQuantity = new SimpleIntegerProperty (transacQuantity);

this.transacTotalPrice = new SimpleDoubleProperty (transacTotalPrice);

}

public Integer getTransacCustomerNo() {

return transacCustomerNo.get();

}

public void setTransacCustomerNo(Integer transacCustomerNo) {

this.transacCustomerNo =new SimpleIntegerProperty (transacCustomerNo);

}

public String getTransacName() {

return transacName.get();

}

public void setTransacName(String transacName) {

this.transacName =new SimpleStringProperty (transacName);

}

public Integer getTransacQuantity() {

return transacQuantity.get();

}

public void setTransacQuantity(Integer transacQuantity) {

this.transacQuantity = new SimpleIntegerProperty(transacQuantity);

}

public Double getTransacTotalPrice() {

return transacTotalPrice.get();

}

public void setTransacTotalPrice(Double transacTotalPrice) {

this.transacTotalPrice = new SimpleDoubleProperty(transacTotalPrice);

}

}

Sales.java

package system;

import java.time.LocalDate;

import java.util.Date;

import javafx.beans.property.ObjectProperty;

import javafx.beans.property.SimpleDoubleProperty;

import javafx.beans.property.SimpleIntegerProperty;

import javafx.beans.property.SimpleObjectProperty;

import javafx.beans.property.SimpleStringProperty;

public class Sales {

private SimpleIntegerProperty salesCustomerNo;

private SimpleDoubleProperty salesTotalPrice;

private SimpleObjectProperty<LocalDate> salesDate;

public Sales(Integer salesCustomerNo, Double salesTotalPrice, LocalDate salesDate) {

this.salesCustomerNo = new SimpleIntegerProperty(salesCustomerNo);

this.salesTotalPrice = new SimpleDoubleProperty (salesTotalPrice);

this.salesDate = new SimpleObjectProperty<LocalDate>(salesDate);

}

public Integer getSalesCustomerNo() {

return salesCustomerNo.get();

}

public void setSalesCustomerNo(Integer salesCustomerNo) {

this.salesCustomerNo =new SimpleIntegerProperty (salesCustomerNo);

}

public Double getSalesTotalPrice() {

return salesTotalPrice.get();

}

public void setSalesTotalPrice(Double salesTotalPrice) {

this.salesTotalPrice = new SimpleDoubleProperty(salesTotalPrice);

}

public LocalDate getSalesDate() {

return salesDate.get();

}

public void setSalesDate(LocalDate salesDate) {

this.salesDate = new SimpleObjectProperty<LocalDate>(salesDate);

}

}