**Book Shop**

package bookshop;

/\*\*

\* Programmer: Abhishek Jassal

\* @version 2.1

\* @author Abhishek

\*/

public class BookShop extends Application {

/\*\*

\* This main method which calls the launch method

\* @param args the command line arguments

\*/

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage stage) throws Exception {

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

Scene scene=new Scene(root);

stage.setTitle("Book Shop");

stage.setScene(scene);

stage.show();

}

}

**Book Inventory**

package bookshop;

private String bookname,author;

private int quantity;

private double price;

public BooksInventory() {

}

public BooksInventory(String bookname, String author, int quantity, double price) {

this.bookname = bookname;

this.author = author;

this.quantity = quantity;

this.price = price;

}

public String getBookname() {

return bookname;

}

public String getAuthor() {

return author;

}

public int getQuantity() {

return quantity;

}

public double getPrice() {

return price;

}

public void setBookname(String bookname) {

this.bookname = bookname;

}

public void setAuthor(String author) {

this.author = author;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

public void setPrice(double price) {

this.price = price;

}

}

**Inventory Controller**

package bookshop;

public class InventoryController implements Initializable {

static int record=0;

PrintWriter output;

Scanner input;

File newfile;

@FXML private TextField txtBkname,txtauthname,txtprcname,txtnumbk,txtsrchname;

@FXML private Button btnsearch,btnadd,btnsave,btnexit,btnwrite,btnnext,btnprevious,btnedit,btndisplay;

@FXML private ListView lstview;

ArrayList<BooksInventory> books = new ArrayList<BooksInventory>();

ArrayList<String> options = new ArrayList<>();

ObservableList<String> items;

@FXML

private void add(ActionEvent event) //adding the new record

{

record =books.size();

System.out.println("In Add, Record = "+record+" Size = "+books.size());

txtBkname.clear();

txtauthname.clear();

txtprcname.clear();

txtnumbk.clear();

}

@FXML

private void save(ActionEvent event)

{

BooksInventory bks = new BooksInventory();

bks.setBookname(txtBkname.getText());

bks.setAuthor(txtauthname.getText());

bks.setPrice(Double.parseDouble(txtprcname.getText()));

bks.setQuantity(Integer.parseInt(txtnumbk.getText()));

if(record==books.size())

{

books.add(bks);

System.out.println("In Save, if Record = "+record+" Size = "+books.size());

}

else

{

books.set(record, bks);

System.out.println("In Save, if Record = "+record+" Size = "+books.size());

}

}

@FXML

private void exit(ActionEvent event)

{

{

Alert box=new Alert(Alert.AlertType.CONFIRMATION);

box.setTitle("Exit message");

box.setHeaderText("Confirmation");

box.setContentText("Do you want to Exit?");

box.showAndWait().ifPresent(response -> {

if(response == ButtonType.OK)

{

System.exit(0);

}

});

}

}

@FXML

private void write(ActionEvent event)

{

try

{

PrintWriter pwr = new PrintWriter(newfile);

System.out.println("In PrintWriter, Your File is Empty");

pwr.print("");

pwr.close();

FileWriter fwr = new FileWriter(newfile,true);

output = new PrintWriter(fwr);

System.out.println("PrintWriter Output Append Mode");

for(int i=0;i<books.size();i++)

{

output.println(books.get(i).getBookname()+","+books.get(i).getAuthor()+","+books.get(i).getPrice()+","+books.get(i).getQuantity());

}

output.close();

}

catch(Exception e)

{

System.out.println("In Writer, There is Error in creation of File") ;

}

}

@FXML

private void next(ActionEvent event)

{

try

{

record++;

txtBkname.setText(books.get(record).getBookname());

txtauthname.setText(books.get(record).getAuthor());

txtprcname.setText(String.valueOf(books.get(record).getPrice()));

txtnumbk.setText(String.valueOf(books.get(record).getQuantity()));

if(record==0)

btnprevious.setDisable(true);

else

btnprevious.setDisable(false);

if(record == books.size()-1)

btnnext.setDisable(true);

else

btnnext.setDisable(false);

}

catch(IndexOutOfBoundsException e)

{

System.out.println("Next"+e.getMessage());

}

}

@FXML

private void previous(ActionEvent event)

{

try

{

record--;

txtBkname.setText(books.get(record).getBookname());

txtauthname.setText(books.get(record).getAuthor());

txtprcname.setText(String.valueOf(books.get(record).getPrice()));

txtnumbk.setText(String.valueOf(books.get(record).getQuantity()));

if(record==0)

btnprevious.setDisable(true);

else

btnprevious.setDisable(false);

if(record==books.size()-1)

btnnext.setDisable(true);

else

btnnext.setDisable(false);

}

catch(IndexOutOfBoundsException e)

{

System.out.println("Previous"+e.getMessage());

}

}

@FXML

private void edit(ActionEvent event)

{

btnsave.setDisable(false);

btnadd.setDisable(true);

if(record==0)

btnprevious.setDisable(true);

else

btnprevious.setDisable(false);

if(record==books.size()-1)

btnnext.setDisable(true);

else

btnnext.setDisable(false);

}

@FXML

private void display(ActionEvent event)

{

try

{

Scanner input=new Scanner(newfile); //open read mode

if(newfile.exists())

{

options.add("Name"+" "+"Author"+" "+"Price"+" "+"Quantity"+" "+"Total Price");

while(input.hasNext())

{

String s=input.next();

System.out.println(s);

Scanner in = new Scanner(s);

in.useDelimiter(",");

String name = in.next();

String author = in.next();

double price = in.nextDouble();

int quantity= in.nextInt();

double totalPrice;

totalPrice=price\*quantity;

options.add(name+" "+author+" "+price+" "+quantity+" "+totalPrice);

lstview.getItems().addAll();

}

}

items = FXCollections.observableArrayList(options);

lstview.setItems(items);

options.removeAll(items);

System.out.println("listview size="+items.size());

input.close();

}

catch(IOException e)

{

System.out.println("Error in File Display");

}

catch (InputMismatchException ex)

{

System.out.println("Invalid Input");

}

}

@FXML

private void search(ActionEvent event)

{lstview.getItems().clear();

try

{

Scanner input=new Scanner(newfile); //open read mode

if(newfile.exists())

{

options.add("Name"+" "+"Author"+" "+"Price"+" "+"Quantity"+" "+"Total Price");

while(input.hasNext())

{

String s=input.next();

System.out.println(s);

Scanner in = new Scanner(s);

in.useDelimiter(",");

String name = in.next();

String author = in.next();

double price = in.nextDouble();

int quantity= in.nextInt();

double totalPrice;

totalPrice=price\*quantity;

if(name.equalsIgnoreCase(txtsrchname.getText()) ){

options.add(name+" "+author+" "+price+" "+quantity+" "+totalPrice);

lstview.getItems().addAll();

}

}

}

items = FXCollections.observableArrayList(options);

lstview.setItems(items);

options.removeAll(items);

System.out.println("listview size="+items.size());

input.close();

}

catch(IOException e)

{

System.out.println("Error in File Display");

}

catch (InputMismatchException ex)

{

System.out.println("Invalid Input");

}

}

@Override

public void initialize(URL url, ResourceBundle rb) {

lstview.setPrefHeight(700);

newfile=new File("Books.txt");

try

{

if(newfile.exists())

{

input=new Scanner(newfile);

while(input.hasNext())

{

String s=input.next();

Scanner in=new Scanner(s);

in.useDelimiter(",");

BooksInventory bki=new BooksInventory();

bki.setBookname(in.next());

bki.setAuthor(in.next());

bki.setQuantity(in.nextInt());

bki.setPrice(in.nextDouble());

books.add(bki);

txtBkname.setText(books.get(record).getBookname());

txtauthname.setText(books.get(record).getAuthor());

txtprcname.setText(String.valueOf(books.get(record).getQuantity()));

txtnumbk.setText(String.valueOf(books.get(record).getPrice()));

}

}

else

{

System.err.println("File Not Exists Add Record First");

}

input.close();

}

catch(Exception e)

{

System.out.println("Error in Initialize");

}

}

}

**CSS File**

.root{

-fx-background-color:lavender;

}

.button{

-fx-border-style:solid;

-fx-border-color:lightpink;

-fx-border-width: 2px;

-fx-text-fill:palevioletred;

-fx-font-family: 'serif';

-fx-font-size:10pt;

-fx-font-style: italic;

-fx-font-weight: bold;

}

.label{

-fx-text-fill:darkslategray;

-fx-font-weight: bold;

-fx-font-size:10pt;

}

**Inventory FXML**

<?xml version="1.0" encoding="UTF-8"?>

<GridPane id="AnchorPane" prefHeight="400.0" prefWidth="1000.0" styleClass="mainFxmlClass" xmlns:fx="http://javafx.com/fxml/1" fx:controller="bookshop.InventoryController">

<stylesheets>

<URL value="@ShopStyle.css"/>

</stylesheets>

<columnConstraints>

<ColumnConstraints percentWidth="50.0" />

<ColumnConstraints percentWidth="50.0" />

</columnConstraints>

<padding>

<Insets top="30" bottom="30" right="30" left="30" />

</padding>

<children>

<Label text="Book Name" GridPane.rowIndex="0" GridPane.columnIndex="0" GridPane.halignment="LEFT" />

<Label text="Enter Book Name to Search" GridPane.rowIndex="0" GridPane.columnIndex="1" GridPane.halignment="LEFT" />

<Label text="Author Name" GridPane.rowIndex="1" GridPane.columnIndex="0" GridPane.halignment="LEFT" >

<GridPane.margin>

<Insets top="5"/>

</GridPane.margin>

</Label>

<Label text="Price of Book" GridPane.rowIndex="2" GridPane.columnIndex="0" GridPane.halignment="LEFT" >

<GridPane.margin>

<Insets top="5"/>

</GridPane.margin>

</Label>

<Label text="Number of books in stock" GridPane.rowIndex="3" GridPane.columnIndex="0" GridPane.halignment="LEFT" >

<GridPane.margin>

<Insets right="200"/>

</GridPane.margin>

</Label>

<TextField fx:id="txtBkname" GridPane.rowIndex="0" GridPane.columnIndex="0" GridPane.halignment="RIGHT">

<GridPane.margin>

<Insets left="180" right="80" />

</GridPane.margin>

</TextField>

<TextField fx:id="txtsrchname" GridPane.rowIndex="0" GridPane.columnIndex="1" GridPane.halignment="CENTER">

<GridPane.margin>

<Insets left="200" right="80" />

</GridPane.margin>

</TextField>

<Button fx:id="search" text="SEARCH" onAction="#search" GridPane.rowIndex="0" GridPane.columnIndex="1" GridPane.halignment="RIGHT"/>

<TextField fx:id="txtauthname" GridPane.rowIndex="1" GridPane.columnIndex="0" GridPane.halignment="RIGHT">

<GridPane.margin>

<Insets left="180" right="80" top="10"/>

</GridPane.margin>

</TextField>

<TextField fx:id="txtprcname" GridPane.rowIndex="2" GridPane.columnIndex="0" GridPane.halignment="RIGHT">

<GridPane.margin>

<Insets left="180" right="80" top="10"/>

</GridPane.margin>

</TextField>

<TextField fx:id="txtnumbk" GridPane.rowIndex="3" GridPane.columnIndex="0" GridPane.halignment="RIGHT">

<GridPane.margin>

<Insets left="180" right="80" top="10"/>

</GridPane.margin>

</TextField>

<!-- buttons-->

<Button fx:id="btnadd" text="ADD" onAction="#add" GridPane.rowIndex="4" GridPane.columnIndex="0" GridPane.halignment="LEFT">

<GridPane.margin>

<Insets top="30" left="20"/>

</GridPane.margin>

</Button>

<!-- -->

<Button fx:id="btnsave" text="SAVE" onAction="#save" GridPane.rowIndex="4" GridPane.columnIndex="0" GridPane.halignment="LEFT">

<GridPane.margin>

<Insets top="30" left="100"/>

</GridPane.margin>

</Button>

<!-- -->

<Button fx:id="btnexit" text="EXIT" onAction="#exit" GridPane.rowIndex="4" GridPane.columnIndex="0" GridPane.halignment="LEFT">

<GridPane.margin>

<Insets top="30" left="190" />

</GridPane.margin>

</Button>

<!-- -->

<Button fx:id="btnwrite" text="WRITE" onAction="#write" GridPane.rowIndex="4" GridPane.columnIndex="0" GridPane.halignment="RIGHT">

<GridPane.margin>

<Insets top="30" right="140" />

</GridPane.margin>

</Button>

<!-- -->

<Button fx:id="btnnext" text="NEXT" onAction="#next" GridPane.rowIndex="5" GridPane.columnIndex="0" GridPane.halignment="LEFT">

<GridPane.margin>

<Insets top="30" left="20"/>

</GridPane.margin>

</Button>

<!-- -->

<Button fx:id="btnprevious" text="PREVIOUS" onAction="#previous" GridPane.rowIndex="5" GridPane.columnIndex="0" GridPane.halignment="LEFT">

<GridPane.margin>

<Insets top="30" left="90" />

</GridPane.margin>

</Button>

<!-- -->

<Button fx:id="btnedit" text="EDIT" onAction="#edit" GridPane.rowIndex="5" GridPane.columnIndex="0" GridPane.halignment="LEFT">

<GridPane.margin>

<Insets top="30" left="190"/>

</GridPane.margin>

</Button>

<!-- -->

<Button fx:id="btndisplay" text="DISPLAY" onAction="#display" GridPane.rowIndex="5" GridPane.columnIndex="0" GridPane.halignment="RIGHT">

<GridPane.margin>

<Insets top="30" right="140"/>

</GridPane.margin>

</Button>

<!-- -->

<ListView fx:id="lstview" GridPane.rowIndex="1" GridPane.columnIndex="1" GridPane.rowSpan="6">

<GridPane.margin>

<Insets top="30"/>

</GridPane.margin>

</ListView>

</children>

</GridPane>