

**CENTENNIAL COLLEGE
COMP 228: JAVA PROGRAMMING**

FINAL PROJECT

NAME: JOKKU SABU KATTAKKAYAM

STUDENT ID NO.: 301013603

Screenshots

Login



The screenshot shows a login interface for 'JAS Book Store'. The title 'JAS Book Store' is displayed in yellow text on a dark teal background. Below the title, there are two input fields: 'Email' with the value 'admin@gmail.com' and 'Password' with masked characters '.....'. At the bottom, there are two buttons: a green 'Login' button and a red 'Cancel' button.

Main Window

JAS BOOK STORE (INVENTORY MANAGEMENT)

JAS Book Store
Best Seller: LOTR

Search:

LOTR

Add To Inventory:

Book Id:

Book Name:

Author:

Price:

Publisher:

Available Quantity:

Sold Quantity:

Available Books:

PRICE	NAME
150	LOTR
100	Harry Potter
12	Kalikkudukka
100	Balarama
135	Comic Book
67	Spider Man
100	Jack Reacher

Code Window

```
import javax.swing.JOptionPane;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;
import javafx.scene.text.Text;
import javafx.stage.StageStyle;
//For Database
import java.sql.*;

public class ApplicationGUI extends Application
{
    Text bestSellerText = new Text("Best Seller: ");
    Label title= new Label("JAS Book Store");

    //For Login Window
    //creating label email
    Label emailLbl= new Label("Email");

    //creating label password
    Label passwordLbl= new Label("Password");

    //Creating Text Filed for email
    TextField email = new TextField();
```

```

//Creating Text Filed for password
PasswordField password = new PasswordField();

//Creating Buttons
Button b1 = new Button("Login");
Button b2 = new Button("Cancel");

//For Main Window

Label bookIdLbl = new Label ("Book Id: ");
TextField bookIdTextField = new TextField();

Label bookNameLbl = new Label ("Book Name: ");
TextField bookNameTextField = new TextField();

Label bookAuthorLbl = new Label ("Author: ");
TextField bookAuthorTextField = new TextField();

Label bookPriceLbl = new Label ("Price: ");
TextField bookPriceTextField = new TextField();

Label bookPublisherLbl = new Label ("Publisher: ");
TextField bookPublisherTextField = new TextField();

Label quantityAvailableLbl = new Label ("Available Quantity: ");
TextField quantityAvailableTextField = new TextField();

Label quantitySoldLbl = new Label ("Sold Quantity: ");
TextField quantitySoldTextField = new TextField();

Label addToInventoryLbl = new Label ("Add To Inventory: ");
TextField addToInventoryTextField = new TextField();

Label searchFromInventoryLbl = new Label ("Search: ");
TextField searchFromInventoryTextField = new TextField();

Label availableBooksLbl = new Label ("Available Books: ");
TextArea availableBooksTextArea = new TextArea();

//*****
//Label TemporaryLbl = new Label("THIS IS A TEMPORARY LABEL.");

Button searchBtn = new Button("Search");
Button createBtn = new Button("Add Book");
Button deleteBtn = new Button("Delete Book");
Button editBtn = new Button("Edit Book");
Button saveBtn = new Button("Save Book info");
Button viewAllBtn = new Button("View Existing Books");
Button addQuantityBtn = new Button("Add Quantity");
Button removeQuantityBtn = new Button("Delete Quantity");

Button showAvailableBooksBtn = new Button("Show Available");
Button soldBtn = new Button("SOLD");

Stage window;           //Taking instance of a stage as Global stage variable.

```

```

@Override
public void start(Stage primaryStage) throws Exception
{
    Stage popupwindow=new Stage();

    popupwindow.setTitle("Login window");

    b1.setOnAction(e -> LoginBtnClicked(popupwindow));

    b2.setOnAction(e -> System.exit(0));

    //Creating a Grid Pane
    GridPane gridPane = new GridPane();

    //Setting size for the pane
    gridPane.setMinSize(400, 300);

    //Setting the padding
    gridPane.setPadding(new Insets(10, 10, 10, 10));

    //Setting the vertical and horizontal gaps between the columns
    gridPane.setVgap(20);
    gridPane.setHgap(10);

    //Setting the Grid alignment
    gridPane.setAlignment(Pos.CENTER);

    //Arranging all the nodes in the grid
    gridPane.add(emailLbl, 0, 1);
    gridPane.add(email, 1, 1);
    gridPane.add(passwordLbl, 0, 2);
    gridPane.add(password, 1, 2);
    gridPane.add(b1, 0, 3);
    gridPane.add(b2, 1, 3);
    gridPane.add(title, 0, 0);

    //Styling nodes
    b1.setStyle("-fx-background-color: green; -fx-text-fill: white;");
    b2.setStyle("-fx-background-color: red; -fx-text-fill: white;");

    emailLbl.setStyle(" -fx-text-fill: white; -fx-font: normal bold 15px
'serif'");
    passwordLbl.setStyle(" -fx-text-fill: white; -fx-font: normal bold 15px
'serif' ");
    title.setStyle(" -fx-text-fill: gold; -fx-font: normal bold 25px
'calibri' ");
    gridPane.setStyle("-fx-background-color: darkslategrey;");

    // Creating a scene object
    Scene scenel = new Scene(gridPane);

    popupwindow.setScene(scenel);

    popupwindow.initStyle(StageStyle.UNDECORATED);
    popupwindow.showAndWait();

    popupwindow.setOnCloseRequest(e->System.exit(0));

    popupwindow.close();
}

```

[illegible]

[illegible]

[illegible]

[illegible]


```

        int numberOfColumns = metaData.getColumnCount();

        //To print the first book
        while (resultSet.next()) {
            displayDetails(resultSet);
        }

        setEditableFalse();
    }
    catch(ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
        e.printStackTrace();
    }
    catch(SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
        e.printStackTrace();
    }
}

//To add a new book
public void CreateBtnClicked()
{
    //To get the new ID
    int nextId=0;
    //loads the driver in memory
    try {

        Class.forName("oracle.jdbc.OracleDriver");
        //Connection connection =
        DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:1521:SQLD", "COMP228SY_F18_41", "password");//connection details
        Connection connection =
        DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
        "COMP228SY_F18_41", "password");//connection details
        //String selected= _cmbID.getSelectionModel().getSelectedItem();
        Statement showBooksStatement = connection.createStatement();
        ResultSet resultSet = showBooksStatement.executeQuery("select * from
        BOOKS where BOOKID = (select max(BOOKID) from BOOKS)");

        // fetch all rows from the table
        // get ResultSet's meta data

        ResultSetMetaData metaData = resultSet.getMetaData();

        // display query results
        while (resultSet.next())
        {
            nextId = Integer.parseInt(resultSet.getString("BOOKID"));
        }
    }
    catch(ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
        e.printStackTrace();
    }
    catch(SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
        e.printStackTrace();
    }
}

```

```

    }

    if(createBtn.getText().equals("Save"))
    {
        setEditableTrue();
        nextId = nextId +1;
        addToInventoryLbl.setText("ItsWorking");
        //loads the driver in memory
        try {

            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
            DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:1521:SQLD", "COMP228SY_F18_41", "password");//connection details
            Connection connection =
            DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
            "COMP228SY_F18_41", "password");//connection details
            //String selected= _cmbID.getSelectionModel().getSelectedItem();
            Statement showBooksStatement = connection.createStatement();

            showBooksStatement.executeQuery("insert into Books(BOOKID, NAME,
            AUTHOR, PRICE, PUBLISHER, AVAILABLEQUANTITY, SOLDQUANTITY) values(
            "+nextId+", '"+ bookNameTextField.getText()
            +"', '"+bookAuthorTextField.getText()+"',
            "+Integer.parseInt(bookPriceTextField.getText())+",
            '"+bookPublisherTextField.getText()+"',"+
            Integer.parseInt(quantityAvailableTextField.getText())+",
            "+Integer.parseInt(quantitySoldTextField.getText())+")");

            // fetch all rows from the table
            // get ResultSet's meta data

            //ResultSetMetaData metaData = resultSet.getMetaData();

        }
        catch(ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
        }
        catch(SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
        }
        clearTextFields();
        createBtn.setText("Add Book");
        quantitySoldTextField.setDisable(false);
        bookIdTextField.setDisable(false);
    }
    else
    {
        clearTextFields();
        setEditableTrue();
        bookIdTextField.setDisable(true);
        bookNameTextField.requestFocus();
        createBtn.setText("Save");
        quantitySoldTextField.setText("0");
        quantitySoldTextField.setDisable(true);
    }
}

```

```

}

//To delete a book
public void DeleteBtnClicked()
{
    //For the deletion of the selected book

    if(bookIdTextField.getText().isEmpty())
    {
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("No Books selected!");
        alert.showAndWait();
    }
    else
    {
        try {

            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
            DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY_F18_41", "password");//connection details
            Connection connection =
            DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY_F18_41", "password");//connection details
            //String selected=_cmbID.getSelectionModel().getSelectedItem();
            Statement showBooksStatement = connection.createStatement();
            ResultSet resultSet = showBooksStatement.executeQuery("delete from
BOOKS where BOOKID = "+Integer.parseInt(bookIdTextField.getText()));

        }
        catch(ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
        }
        catch(SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
        }
        clearTextFields();
        ShowBtnClicked();
    }
}

//To Edit the details of the book
public void EditBtnClicked()
{
    if(bookIdTextField.getText().isEmpty())
    {
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("No Books selected!");
        alert.showAndWait();
    }
    else
    {
        if(editBtn.getText().equals("Save"))

```

```

        {
            try {

                Class.forName("oracle.jdbc.OracleDriver");
                //Connection connection =
                DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:1521:SQLD", "COMP228SY_F18_41", "password");//connection details
                Connection connection =
                DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
                "COMP228SY_F18_41", "password");//connection details
                //String selected= _cmbID.getSelectionModel().getSelectedItem();
                Statement showBooksStatement = connection.createStatement();
                ResultSet resultSet = showBooksStatement.executeQuery("update
                BOOKS set NAME = '"+bookNameTextField.getText()+"',
                AUTHOR='"+bookAuthorTextField.getText()+"', " +
                " PRICE =
                "+Integer.parseInt(bookPriceTextField.getText())+",
                PUBLISHER='"+bookPublisherTextField.getText()+"',AVAILABLEQUANTITY="+Integer.pa
                rseInt(quantityAvailableTextField.getText())
                +",
                SOLDQUANTITY="+Integer.parseInt(quantitySoldTextField.getText())+" where
                BOOKID="+Integer.parseInt(bookIdTextField.getText()));

            }
            catch(ClassNotFoundException e) {
                JOptionPane.showMessageDialog(null, e.getMessage());
                e.printStackTrace();
            }
            catch(SQLException e) {
                JOptionPane.showMessageDialog(null, e.getMessage());
                e.printStackTrace();
            }
            clearTextFields();
            ShowBtnClicked();
            clearTextFields();
            editBtn.setText("Edit Book");
            quantitySoldTextField.setDisable(false);
            bookIdTextField.setDisable(false);
        }
        else
        {
            setEditableTrue();
            bookIdTextField.setDisable(true);
            bookNameTextField.requestFocus();
            editBtn.setText("Save");
        }
    }

}

//For login
public void LoginBtnClicked(Stage s)
{
    if(email.getText().isEmpty() || password.getText().isEmpty())
    {
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("Please enter your Email ID and Password!");
        alert.showAndWait();
    }
}

```

```

else
{
    if(email.getText().equals("admin@gmail.com") &&
password.getText().equals("admin@books"))
    {
        s.close();
    }
    else
    {
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("Invalid Email ID or Password!");
        alert.showAndWait();
    }
}

//To Add quantity
public void AddQuantityBtnClicked()
{
    if(bookIdTextField.getText().isEmpty())
    {
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("No Books selected!");
        alert.showAndWait();
    }
    else
    {
        try {

            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
            DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY_F18_41", "password");//connection details
            Connection connection =
            DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY_F18_41", "password");//connection details
            //String selected= _cmbID.getSelectionModel().getSelectedItem();
            Statement showBooksStatement = connection.createStatement();
            ResultSet resultSet = showBooksStatement.executeQuery("update BOOKS
set AVAILABLEQUANTITY = AVAILABLEQUANTITY +
"+Integer.parseInt(addToInventoryTextField.getText())+"where
BOOKID="+Integer.parseInt(bookIdTextField.getText()));

        }
        catch(ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
        }
        catch(SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
        }
        clearTextFields();
        ShowBtnClicked();
    }
}

```

```

//To remove quantity of a book
public void RemoveQuantityBtnClicked()
{
    if(bookIdTextField.getText().isEmpty())
    {
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("No Books selected!");
        alert.showAndWait();
    }
    else
    {
        try {

            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
            DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
            1:SQLD", "COMP228SY_F18_41", "password");//connection details
            Connection connection =
            DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
            "COMP228SY_F18_41", "password");//connection details
            //String selected= _cmbID.getSelectionModel().getSelectedItem();
            Statement showBooksStatement = connection.createStatement();
            ResultSet resultSet = showBooksStatement.executeQuery("update BOOKS
            set AVAILABLEQUANTITY = AVAILABLEQUANTITY -
            "+Integer.parseInt(addToInventoryTextField.getText())+"where
            BOOKID="+Integer.parseInt(bookIdTextField.getText()));

        }
        catch(ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
        }
        catch(SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
            e.printStackTrace();
        }
        clearTextFields();
        ShowBtnClicked();
    }
}

//To sell one book
public void SoldBtnClicked()
{
    if(bookIdTextField.getText().isEmpty())
    {
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error Dialog");
        alert.setHeaderText("Look, an Error Dialog");
        alert.setContentText("No Books selected!");
        alert.showAndWait();
    }
    else
    {
        try {

            Class.forName("oracle.jdbc.OracleDriver");
            //Connection connection =
            DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
            1:SQLD", "COMP228SY_F18_41", "password");//connection details

```

```

        Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY_F18_41", "password");//connection details
        //String selected= _cmbID.getSelectionModel().getSelectedItem();
        Statement showBooksStatement = connection.createStatement();
        ResultSet resultSet = showBooksStatement.executeQuery("update BOOKS
set AVAILABLEQUANTITY = AVAILABLEQUANTITY - 1 where
BOOKID="+Integer.parseInt(bookIdTextField.getText()));

    }
    catch(ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
        e.printStackTrace();
    }
    catch(SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
        e.printStackTrace();
    }
    clearTextFields();
    ShowBtnClicked();
}

}

//Method to show all the available books from database
public void ShowBtnClicked()
{
    availableBooksTextArea.clear();//clear text area before showing result
    bestSellerText.setText("Best Seller: ");
    //loads the driver in memory
    try {

        Class.forName("oracle.jdbc.OracleDriver");
        //Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@oracle1.centennialcollege.ca:152
1:SQLD", "COMP228SY_F18_41", "password");//connection details
        Connection connection =
DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521:SQLD",
"COMP228SY_F18_41", "password");//connection details
        //String selected= _cmbID.getSelectionModel().getSelectedItem();
        Statement showBooksStatement = connection.createStatement();
        //ResultSet resultSet = showBooksStatement.executeQuery("select
NAME,AVAILABLEQUANTITY from Books where AVAILABLEQUANTITY > 0 ");
        ResultSet resultSet = showBooksStatement.executeQuery("select PRICE,
NAME from Books order by SOLDQUANTITY desc ");
        // fetch all rows from the table
        // get ResultSet's meta data

        ResultSetMetaData metaData = resultSet.getMetaData();
        int numberOfColumns = metaData.getColumnCount();

        // display the names of the columns in the ResultSet
        for (int i = 1; i <= numberOfColumns; i++){

            availableBooksTextArea.appendText(String.format("%-
10s\t\t",metaData.getColumnName(i)));

        }
        availableBooksTextArea.appendText("\n\n");
    }
}

```

```

        Label tempLabel= new Label("");
        //VBox vbox = new VBox();
        //vbox.setPadding(new Insets(10));
        //vbox.setSpacing(8);

        // display query results
        int j=1;
        while (resultSet.next())
        {
            if(j==1)
            {
                bestSellerText.setText(bestSellerText.getText() +
resultSet.getString("NAME"));
            }

            for (int i = 1; i <= numberOfColumns; i++)
            {

                availableBooksTextArea.appendText(String.format("%s\t\t\t",
resultSet.getObject(i)));
            }
            availableBooksTextArea.appendText("\n");
            j++;
        }
    }

    catch(ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
        e.printStackTrace();
    }
    catch(SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
        e.printStackTrace();
    }
}

//Method to clear all the text fields
public void clearTextFields()
{
    bookIdTextField.clear();
    bookAuthorTextField.clear();
    bookNameTextField.clear();
    bookPriceTextField.clear();
    bookPublisherTextField.clear();
    quantityAvailableTextField.clear();
    quantitySoldTextField.clear();
}

//Method to print details
public void displayDetails(ResultSet resultSet)
{
    try {
        bookIdTextField.appendText(resultSet.getString("BOOKID"));
        bookAuthorTextField.appendText(resultSet.getString("AUTHOR"));
        bookNameTextField.appendText(resultSet.getString("NAME"));
        bookPriceTextField.appendText(resultSet.getString("PRICE"));
        bookPublisherTextField.appendText(resultSet.getString("PUBLISHER"));
        if(Integer.parseInt(resultSet.getString("AVAILABLEQUANTITY"))<0)
        {
            quantityAvailableTextField.appendText("0");
        }
    }
}

```



```

    }
    else
    {

quantityAvailableTextField.appendText(resultSet.getString("AVAILABLEQUANTITY"))
;
    }

    quantitySoldTextField.appendText(resultSet.getString("SOLDQUANTITY"));
}
catch(SQLException e) {
    JOptionPane.showMessageDialog(null, e.getMessage());
    e.printStackTrace();
}
}

//To set the text boxes as read only
public void setEditableFalse()
{
    bookIdTextField.setEditable(false);
    bookNameTextField.setEditable(false);
    bookAuthorTextField.setEditable(false);
    bookPriceTextField.setEditable(false);
    bookPublisherTextField.setEditable(false);
    quantitySoldTextField.setEditable(false);
    quantityAvailableTextField.setEditable(false);
}

//To set the text fields as editable
public void setEditableTrue()
{
    bookIdTextField.setEditable(true);
    bookNameTextField.setEditable(true);
    bookAuthorTextField.setEditable(true);
    bookPriceTextField.setEditable(true);
    bookPublisherTextField.setEditable(true);
    quantitySoldTextField.setEditable(true);
    quantityAvailableTextField.setEditable(true);
}
}

```