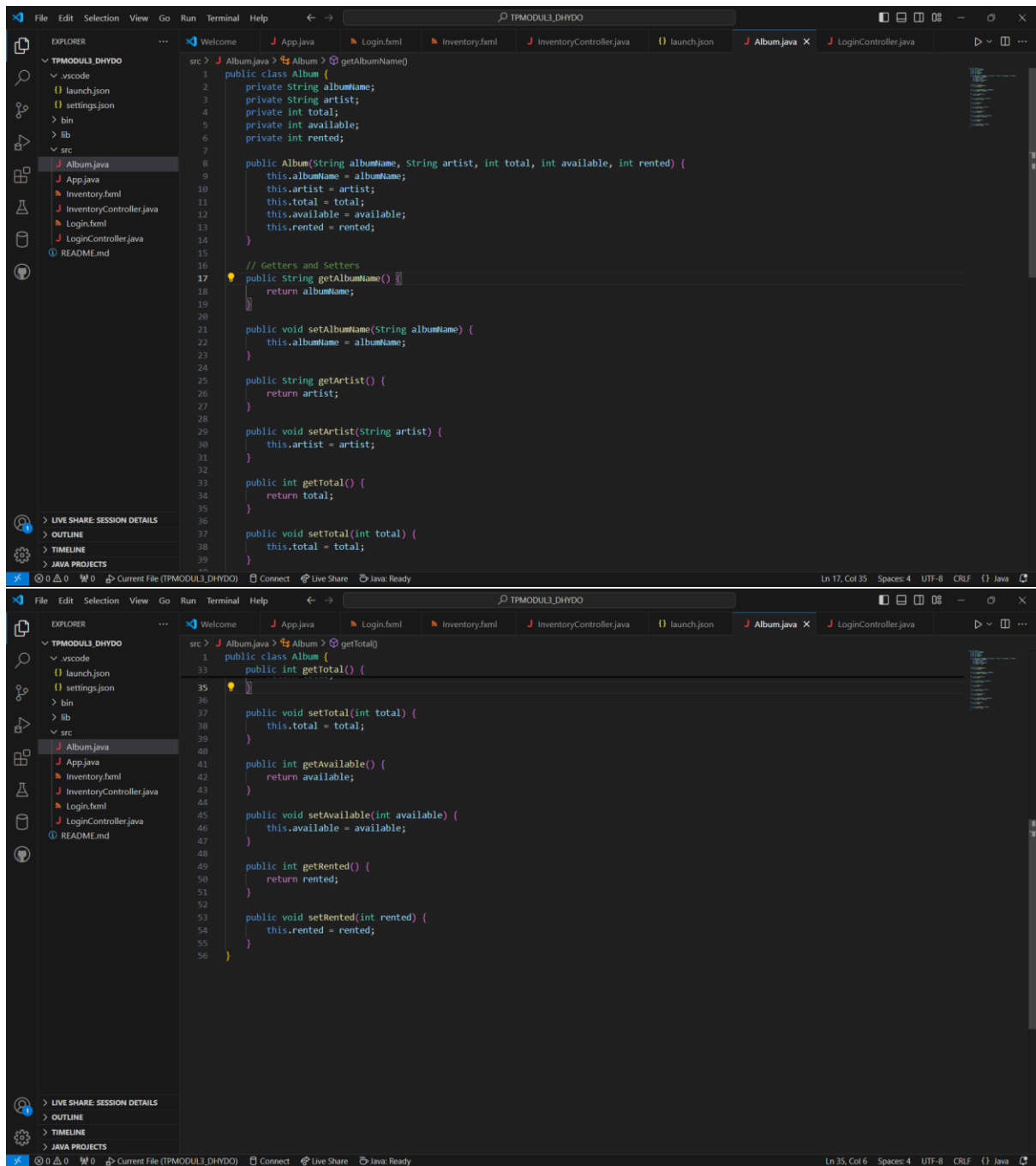
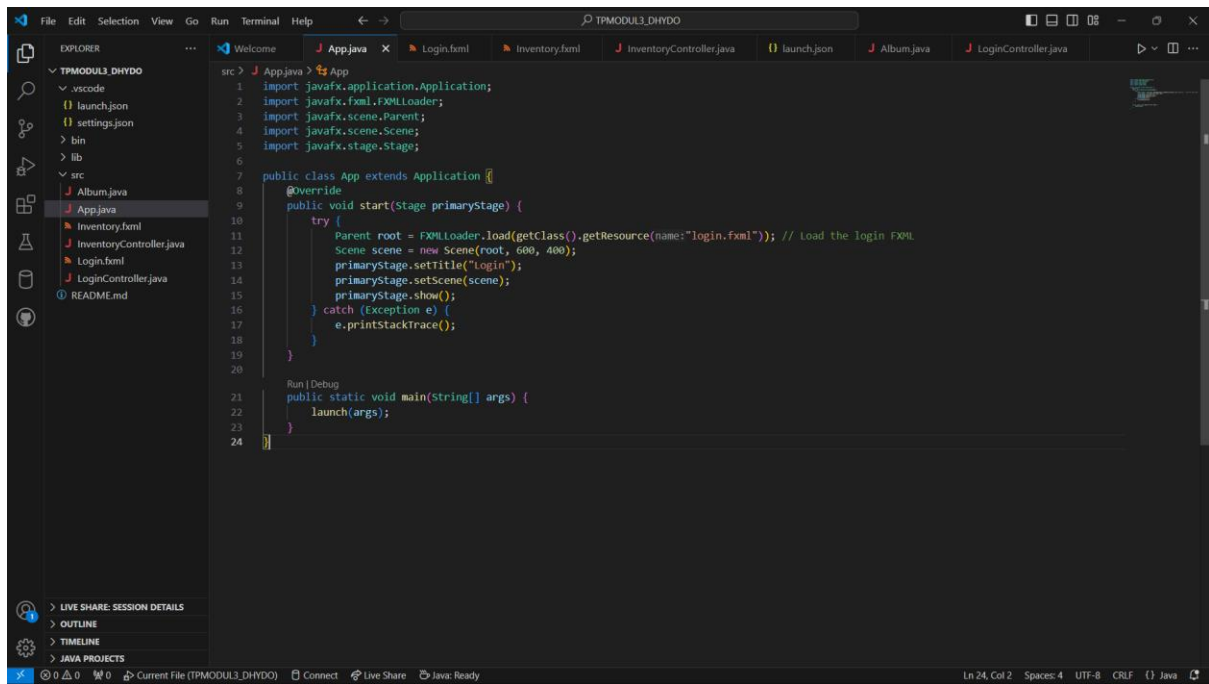
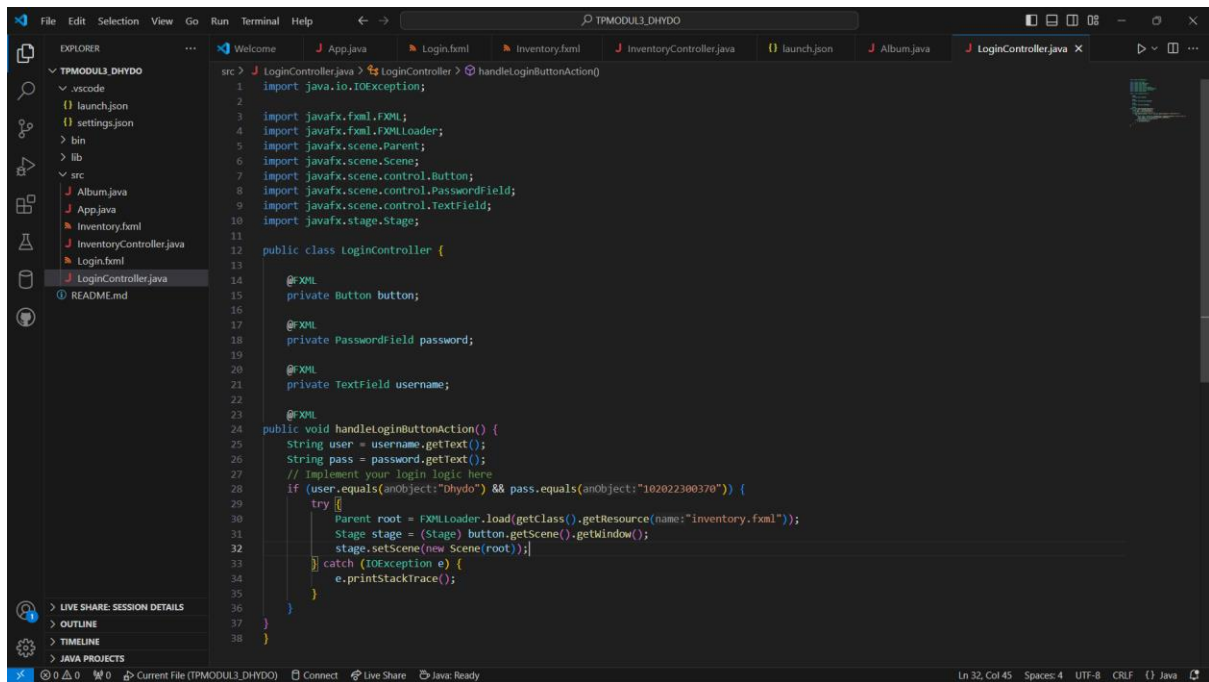


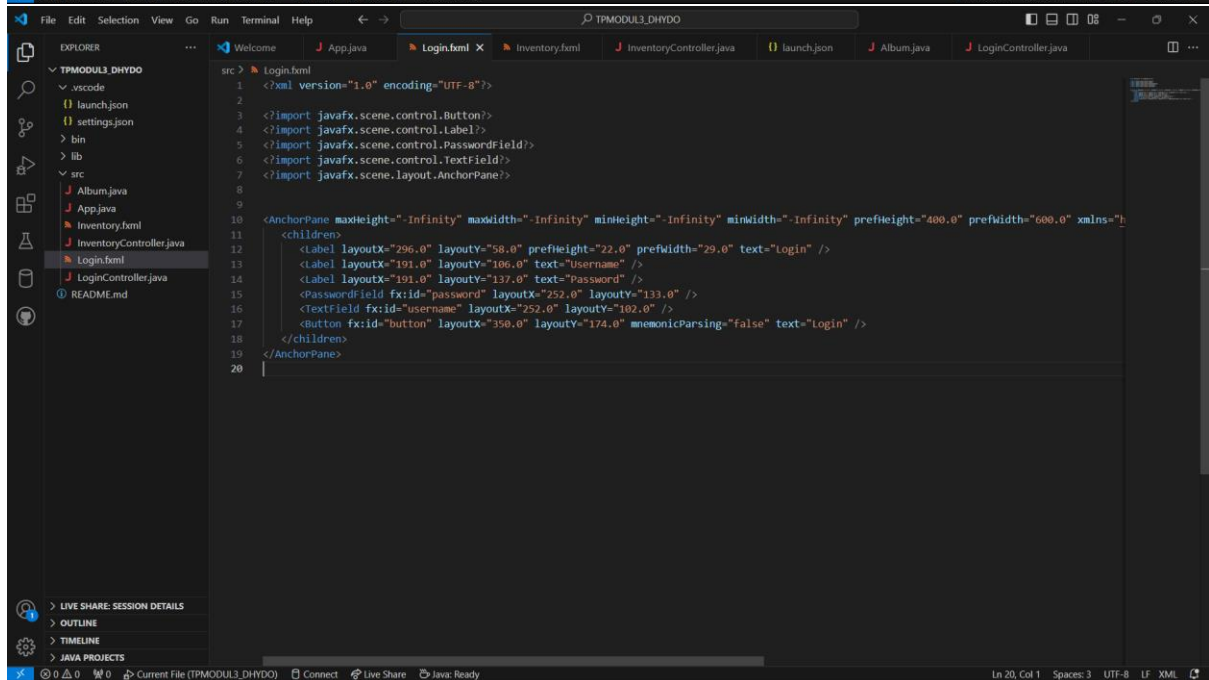
INPUT







```
src > J LoginController.java > LoginController > handleLoginButtonAction()
1  import java.io.IOException;
2
3  import javafx.fxml.FXML;
4  import javafx.fxml.FXMLLoader;
5  import javafx.scene.Parent;
6  import javafx.scene.Scene;
7  import javafx.scene.control.Button;
8  import javafx.scene.control.PasswordField;
9  import javafx.scene.control.TextField;
10 import javafx.stage.Stage;
11
12 public class LoginController {
13
14     @FXML
15     private Button button;
16
17     @FXML
18     private PasswordField password;
19
20     @FXML
21     private TextField username;
22
23     @FXML
24     public void handleLoginButtonAction() {
25         String user = username.getText();
26         String pass = password.getText();
27         // Implement your login logic here
28         if (user.equals("anobject:dhydo") && pass.equals("102022300370")) {
29             try {
30                 Parent root = FXMLLoader.load(getClass().getResource("inventory.fxml"));
31                 Stage stage = (Stage) button.getScene().getWindow();
32                 stage.setScene(new Scene(root));
33             } catch (IOException e) {
34                 e.printStackTrace();
35             }
36         }
37     }
38 }
```



```
src > Login.fxml
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.control.Button?>
4  <?import javafx.scene.control.Label?>
5  <?import javafx.scene.control.PasswordField?>
6  <?import javafx.scene.control.TextField?>
7  <?import javafx.scene.layout.AnchorPane?>
8
9
10 <AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" >
11     <children>
12         <Label layoutX="296.0" layoutY="58.0" prefHeight="22.0" prefWidth="29.0" text="Login" />
13         <Label layoutX="191.0" layoutY="106.0" text="Username" />
14         <Label layoutX="191.0" layoutY="137.0" text="Password" />
15         <PasswordField fx:id="password" layoutX="252.0" layoutY="133.0" />
16         <TextField fx:id="username" layoutX="252.0" layoutY="102.0" />
17         <Button fx:id="button" layoutX="350.0" layoutY="174.0" mnemonicParsing="false" text="Login" />
18     </children>
19 </AnchorPane>
20
```

```
File Edit Selection View Go Run Terminal Help
TPMODUL3_DHYDO

src > J InventoryController.java > InventoryController
11 public class InventoryController {
12     // Observable list to store albums
13     private ObservableList<Album> albumList = FXCollections.observableArrayList();
14
15     @FXML
16     public void initialize() {
17         // Initialize TableView columns
18         albumNameColumn.setCellValueFactory(new PropertyValueFactory<>(<property>:"albumName"));
19         artistColumn.setCellValueFactory(new PropertyValueFactory<>(<property>:"artist"));
20         availableColumn.setCellValueFactory(new PropertyValueFactory<>(<property>:"available"));
21         totalColumn.setCellValueFactory(new PropertyValueFactory<>(<property>:"total"));
22
23         // Set the items to the TableView
24         albumTableView.setItems(albumList);
25     }
26
27     @FXML
28     public void handleTambahButtonAction() {
29         try {
30             // Validate input fields
31             if (JudulAlbum.getText().isEmpty() || NamaArtist.getText().isEmpty() ||
32                 JumlahTotal.getText().isEmpty() || JumlahRented.getText().isEmpty()) {
33                 showAlert(message:"Please fill in all fields");
34                 return;
35             }
36
37             String albumName = JudulAlbum.getText();
38             String artist = NamaArtist.getText();
39             int total = Integer.parseInt(JumlahTotal.getText());
40             int rented = Integer.parseInt(JumlahRented.getText());
41
42             // Validate total and rented numbers
43             if (rented > total) {
44                 showAlert(message:"Rented albums cannot be more than total albums");
45                 return;
46             }
47
48             // Create new album
49             Album newAlbum = new Album(albumName, artist, total, total - rented, rented);
50         }
51     }
52 }
```

```
File Edit Selection View Go Run Terminal Help
TPMODUL3_DHYDO

src > J InventoryController.java > InventoryController > handleTambahButtonAction()
68 public void handleTambahButtonAction() {
69
70     // Create new album
71     Album newAlbum = new Album(albumName, artist, total, total - rented, rented);
72
73     // Add to list
74     albumList.add(newAlbum);
75
76     // Clear input fields
77     clearInputFields();
78 } catch (NumberFormatException e) {
79     showAlert(message:"Please enter valid numbers for Total and Rented");
80 }
81
82 @FXML
83 public void handleHapusButtonAction() {
84     // Get selected album
85     Album selectedAlbum = albumTableView.getSelectionModel().getSelectedItem();
86
87     if (selectedAlbum != null) {
88         albumList.remove(selectedAlbum);
89         clearInputFields();
90     } else {
91         showAlert(message:"Please select an album to delete");
92     }
93 }
94
95 @FXML
96 public void handleRentButtonAction() {
97     Album selectedAlbum = albumTableView.getSelectionModel().getSelectedItem();
98
99     if (selectedAlbum != null) {
100         if (selectedAlbum.getAvailable() > 0) {
101             // Update album details
102             selectedAlbum.setAvailable(selectedAlbum.getAvailable() - 1);
103             selectedAlbum.setRented(selectedAlbum.getRented() + 1);
104
105             // Refresh TableView
106         }
107     }
108 }
```

```
src > J InventoryController.java > InventoryController > handleRentButtonAction()
11 public class InventoryController {
115     public void handleRentButtonAction() {
122         selectedAlbum.setRented(selectedAlbum.getRented() + 1);
123
124         // Refresh TableView
125         albumTableView.refresh();
126     } else {
127         showAlert(message:"No available albums to rent");
128     }
129 } else {
130     showAlert(message:"Please select an album to rent");
131 }
132
133 @FXML
134 public void handleUpdateButtonAction() {
135     Album selectedAlbum = albumTableView.getSelectionModel().getSelectedItem();
136
137     if (selectedAlbum != null) {
138         try {
139             // Update album
140             void Album.setAlbumName(String albumName)
141             selectedAlbum.setAlbumName(JudulAlbum.getText());
142             selectedAlbum.setArtist>NamaArtist.getText());
143
144             int total = Integer.parseInt(JumlahTotal.getText());
145             int rented = Integer.parseInt(JumlahRented.getText());
146
147             // Validate total and rented
148             if (rented > total) {
149                 showAlert(message:"Rented albums cannot be more than total albums");
150                 return;
151             }
152
153             selectedAlbum.setTotal(total);
154             selectedAlbum.setAvailable(total - rented);
155             selectedAlbum.setRented(rented);
156
157             // Refresh TableView
158             albumTableView.refresh();
159         } catch (NumberFormatException e) {
160             showAlert(message:"Please enter valid numbers for Total and Rented");
161         }
162     } else {
163         showAlert(message:"Please select an album to update");
164     }
165 }
166
167 // Helper method to show alerts
168 private void showAlert(String message) {
169     Alert alert = new Alert(Alert.AlertType.ERROR);
170     alert.setTitle(title:"Error");
171     alert.setHeaderText(headerText:null);
172     alert.setContentText(message);
173     alert.showAndWait();
174 }
175
176 // Clear input fields
177 private void clearInputFields() {
178     JudulAlbum.clear();
179    >NamaArtist.clear();
180     JumlahTotal.clear();
181     JumlahRented.clear();
182 }
183
184 }
```

```
src > J InventoryController.java > InventoryController > clearInputFields()
11 public class InventoryController {
135     public void handleUpdateButtonAction() {
153         selectedAlbum.setTotal(total);
154         selectedAlbum.setAvailable(total - rented);
155         selectedAlbum.setRented(rented);
156
157         // Refresh TableView
158         albumTableView.refresh();
159         clearInputFields();
160     } catch (NumberFormatException e) {
161         showAlert(message:"Please enter valid numbers for Total and Rented");
162     }
163 } else {
164     showAlert(message:"Please select an album to update");
165 }
166
167 // Helper method to show alerts
168 private void showAlert(String message) {
169     Alert alert = new Alert(Alert.AlertType.ERROR);
170     alert.setTitle(title:"Error");
171     alert.setHeaderText(headerText:null);
172     alert.setContentText(message);
173     alert.showAndWait();
174 }
175
176 // Clear input fields
177 private void clearInputFields() {
178     JudulAlbum.clear();
179    >NamaArtist.clear();
180     JumlahTotal.clear();
181     JumlahRented.clear();
182 }
183
184 }
```

