**Piyush Piyush(22050267)**

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.\*;

import javafx.stage.Stage;

import javafx.geometry.\*;

import java.sql.\*;

import javafx.collections.\*;

public class Main extends Application {

TableView<Record> table;

TextField titleInput, formatInput, priceInput;

Connection conn;

ObservableList<Record> data = FXCollections.observableArrayList();

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage stage) {

connectToDatabase();

TableColumn<Record, Integer> idCol = new TableColumn<>("ID");

idCol.setCellValueFactory(new PropertyValueFactory<>("id"));

TableColumn<Record, String> titleCol = new TableColumn<>("Title");

titleCol.setCellValueFactory(new PropertyValueFactory<>("title"));

TableColumn<Record, String> formatCol = new TableColumn<>("Format");

formatCol.setCellValueFactory(new PropertyValueFactory<>("format"));

TableColumn<Record, Double> priceCol = new TableColumn<>("Price");

priceCol.setCellValueFactory(new PropertyValueFactory<>("price"));

table = new TableView<>();

table.getColumns().addAll(idCol, titleCol, formatCol, priceCol);

titleInput = new TextField();

titleInput.setPromptText("Title");

formatInput = new TextField();

formatInput.setPromptText("Format");

priceInput = new TextField();

priceInput.setPromptText("Price");

Button insertBtn = new Button("Insert");

insertBtn.setOnAction(e -> {

try {

insertRecord();

} catch (NumberFormatException ex) {

showAlert("Invalid Input", "Please enter a valid number for Price.");

}

});

Button updateBtn = new Button("Update");

updateBtn.setOnAction(e -> {

try {

updateRecord();

} catch (Exception ex) {

showAlert("Update Failed", "Please select a record and enter valid data.");

}

});

Button deleteBtn = new Button("Delete");

deleteBtn.setOnAction(e -> {

try {

deleteRecord();

} catch (Exception ex) {

showAlert("Delete Failed", "Please select a record to delete.");

}

});

Button viewBtn = new Button("View Data");

viewBtn.setOnAction(e -> loadData());

HBox inputBox = new HBox(10, titleInput, formatInput, priceInput, insertBtn, updateBtn, deleteBtn, viewBtn);

inputBox.setPadding(new Insets(10));

Label footer = new Label("Name: Piyush Piyush | Student ID: 22050267 | Date: 2025-06-05");

VBox layout = new VBox(10, table, inputBox, footer);

layout.setPadding(new Insets(10));

stage.setTitle("Record Store Inventory");

stage.setScene(new Scene(layout, 800, 400));

stage.show();

loadData(); // Automatically load data on start

}

private void connectToDatabase() {

try {

conn = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/records",

"sharmapiyush17",

"Piyush@123"

);

System.out.println("Connected to database.");

} catch (Exception e) {

e.printStackTrace();

showAlert("Database Error", "Unable to connect to the database.");

}

}

private void loadData() {

data.clear();

try {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM Records");

while (rs.next()) {

data.add(new Record(

rs.getInt("record\_id"),

rs.getString("title"),

rs.getString("format"),

rs.getDouble("purchase\_price")

));

}

table.setItems(data);

} catch (SQLException e) {

e.printStackTrace();

}

}

private void insertRecord() {

String title = titleInput.getText();

String format = formatInput.getText();

double price = Double.parseDouble(priceInput.getText());

if (title.isEmpty() || format.isEmpty()) {

showAlert("Missing Fields", "Title and Format cannot be empty.");

return;

}

try {

PreparedStatement ps = conn.prepareStatement("INSERT INTO Records (title, format, purchase\_price) VALUES (?, ?, ?)");

ps.setString(1, title);

ps.setString(2, format);

ps.setDouble(3, price);

ps.executeUpdate();

clearInputs();

loadData();

} catch (SQLException e) {

e.printStackTrace();

showAlert("Insert Failed", "Could not insert the record.");

}

}

private void updateRecord() {

Record selected = table.getSelectionModel().getSelectedItem();

if (selected == null) {

showAlert("No Selection", "Please select a record to update.");

return;

}

String title = titleInput.getText();

String format = formatInput.getText();

double price = Double.parseDouble(priceInput.getText());

if (title.isEmpty() || format.isEmpty()) {

showAlert("Missing Fields", "Title and Format cannot be empty.");

return;

}

try {

PreparedStatement ps = conn.prepareStatement("UPDATE Records SET title=?, format=?, purchase\_price=? WHERE record\_id=?");

ps.setString(1, title);

ps.setString(2, format);

ps.setDouble(3, price);

ps.setInt(4, selected.getId());

ps.executeUpdate();

clearInputs();

loadData();

} catch (SQLException e) {

e.printStackTrace();

showAlert("Update Failed", "Could not update the record.");

}

}

private void deleteRecord() {

Record selected = table.getSelectionModel().getSelectedItem();

if (selected == null) {

showAlert("No Selection", "Please select a record to delete.");

return;

}

try {

PreparedStatement ps = conn.prepareStatement("DELETE FROM Records WHERE record\_id=?");

ps.setInt(1, selected.getId());

ps.executeUpdate();

loadData();

} catch (SQLException e) {

e.printStackTrace();

showAlert("Delete Failed", "Could not delete the record.");

}

}

private void clearInputs() {

titleInput.clear();

formatInput.clear();

priceInput.clear();

}

private void showAlert(String title, String message) {

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

alert.setTitle(title);

alert.setHeaderText(null);

alert.setContentText(message);

alert.showAndWait();

}

}A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

2. DATABASE CONNECTION

CODE :

private void connectToDatabase() {

try {

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/recordstore", "sharmapiyush17", "Piyush@123");

} catch (Exception e) {

e.printStackTrace();

}

}

……EXTRACTION OF MYSQL CONNECTOR

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

NOTES:

As we are getting problem in connecting javafx to database. So I am just adding screenshots of the first section and the database files I made.