WelcomePage.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class WelcomePage extends JFrame {

public WelcomePage() {

initComponents(); // Initialize components

}

private void initComponents() {

// Create components

JPanel panel = new JPanel(); // Background panel

panel.setBackground(new Color(54, 57, 63)); // Dark background

JLabel welcomeLabel = new JLabel("Welcome to the Library", SwingConstants.CENTER);

welcomeLabel.setFont(new Font("Serif", Font.BOLD, 36));

welcomeLabel.setForeground(Color.WHITE);

JButton enterButton = new JButton("Enter Library");

enterButton.setFont(new Font("Arial", Font.PLAIN, 20));

enterButton.setBackground(new Color(0, 128, 255)); // Light blue button

enterButton.setForeground(Color.WHITE);

enterButton.setFocusPainted(false);

enterButton.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));

// Add action listener to the Enter button

enterButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

openMainPage();

}

});

// Set layout and add components

panel.setLayout(new BorderLayout(20, 20));

panel.add(welcomeLabel, BorderLayout.CENTER);

panel.add(enterButton, BorderLayout.SOUTH);

panel.setBorder(BorderFactory.createEmptyBorder(50, 50, 50, 50));

// Frame settings

add(panel);

setTitle("Welcome");

setSize(500, 400);

setLocationRelativeTo(null); // Center the frame

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

// Open Main Page and close Welcome Page

private void openMainPage() {

new Main().setVisible(true);

dispose(); // Close Welcome Page

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new WelcomePage().setVisible(true));

}

}

Main.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Main extends JFrame {

private JButton addBookButton, issueBookButton, returnBookButton, viewBooksButton, deleteBookButton;

public Main() {

initComponents();

}

private void initComponents() {

// Initialize buttons

addBookButton = createButton("Add Book");

issueBookButton = createButton("Issue Book");

returnBookButton = createButton("Return Book");

viewBooksButton = createButton("View Books");

deleteBookButton = createButton("Delete Book");

// Add action listeners for navigation

addBookButton.addActionListener(e -> new AddBookPage().setVisible(true));

issueBookButton.addActionListener(e -> new IssueBookPage().setVisible(true));

returnBookButton.addActionListener(e -> new ReturnBookPage().setVisible(true));

viewBooksButton.addActionListener(e -> new ViewBooksPage().setVisible(true));

deleteBookButton.addActionListener(e -> new DeleteBookPage().setVisible(true));

// Button panel layout

JPanel buttonPanel = new JPanel(new GridLayout(5, 1, 10, 10));

buttonPanel.setBackground(new Color(47, 79, 79));

buttonPanel.setBorder(BorderFactory.createEmptyBorder(30, 30, 30, 30));

// Add buttons to the panel

buttonPanel.add(addBookButton);

buttonPanel.add(issueBookButton);

buttonPanel.add(returnBookButton);

buttonPanel.add(viewBooksButton);

buttonPanel.add(deleteBookButton);

// Frame settings

add(buttonPanel);

setTitle("Library Dashboard");

setSize(400, 500);

setLocationRelativeTo(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

private JButton createButton(String text) {

JButton button = new JButton(text);

button.setFont(new Font("Arial", Font.PLAIN, 18));

button.setBackground(new Color(0, 128, 255));

button.setForeground(Color.WHITE);

button.setBorder(BorderFactory.createLineBorder(Color.WHITE, 2));

return button;

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new Main().setVisible(true));

}

}

AddBookPage.java

import javax.swing.\*;

import java.awt.\*;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class AddBookPage extends JFrame {

private JTextField titleField, authorField, userIdField;

private JButton addButton, backButton;

public AddBookPage() {

initComponents();

}

private void initComponents() {

JLabel titleLabel = new JLabel("Book Title:");

JLabel authorLabel = new JLabel("Author:");

JLabel userIdLabel = new JLabel("User ID:");

titleField = new JTextField(20);

authorField = new JTextField(20);

userIdField = new JTextField(20);

addButton = new JButton("Add Book");

backButton = new JButton("Back");

addButton.addActionListener(e -> addBook());

backButton.addActionListener(e -> {

new Main().setVisible(true);

dispose();

});

JPanel panel = new JPanel(new GridLayout(4, 2, 10, 10));

panel.setBorder(BorderFactory.createEmptyBorder(30, 30, 30, 30));

panel.add(titleLabel);

panel.add(titleField);

panel.add(authorLabel);

panel.add(authorField);

panel.add(userIdLabel);

panel.add(userIdField);

panel.add(addButton);

panel.add(backButton);

add(panel);

setTitle("Add Book");

setSize(400, 200);

setLocationRelativeTo(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

private void addBook() {

String title = titleField.getText().trim();

String author = authorField.getText().trim();

String userId = userIdField.getText().trim();

if (title.isEmpty() || author.isEmpty() || userId.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please fill in all fields.");

return;

}

try (Connection conn = DatabaseConnection.getConnection()) {

String query = "INSERT INTO books (title, author, user\_id) VALUES (?, ?, ?)";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, title);

stmt.setString(2, author);

stmt.setString(3, userId);

int rowsInserted = stmt.executeUpdate();

if (rowsInserted > 0) {

JOptionPane.showMessageDialog(this, "Book added successfully!");

titleField.setText("");

authorField.setText("");

userIdField.setText("");

} else {

JOptionPane.showMessageDialog(this, "Failed to add the book.");

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Error adding book: " + e.getMessage());

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

AddBookPage app = new AddBookPage();

app.setVisible(true);

});

}

}

ViewBookPage.java

import javax.swing.\*;

import java.awt.\*;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class ViewBooksPage extends JFrame {

private JTextArea booksTextArea;

private JButton backButton;

public ViewBooksPage() {

initComponents();

}

private void initComponents() {

booksTextArea = new JTextArea(20, 40);

booksTextArea.setEditable(false);

JScrollPane scrollPane = new JScrollPane(booksTextArea);

backButton = new JButton("Back");

backButton.addActionListener(e -> {

new Main().setVisible(true);

dispose();

});

JPanel panel = new JPanel(new BorderLayout());

panel.setBorder(BorderFactory.createEmptyBorder(30, 30, 30, 30));

panel.add(scrollPane, BorderLayout.CENTER);

panel.add(backButton, BorderLayout.SOUTH);

add(panel);

setTitle("View Books");

setSize(500, 400);

setLocationRelativeTo(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

loadBooks();

}

private void loadBooks() {

try (Connection conn = DatabaseConnection.getConnection()) {

String query = "SELECT \* FROM books";

PreparedStatement stmt = conn.prepareStatement(query);

ResultSet rs = stmt.executeQuery();

while (rs.next()) {

booksTextArea.append(

"ID: " + rs.getInt("book\_id") +

", Title: " + rs.getString("title") +

", Author: " + rs.getString("author") +

", Issued: " + (rs.getBoolean("is\_issued") ? "Yes" : "No") +

"\n"

);

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Error loading books.");

}

}

}

ReturnBookPage.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class ReturnBookPage extends JFrame {

private JTextField bookIdField;

private JButton returnButton, backButton;

public ReturnBookPage() {

initComponents();

}

private void initComponents() {

JLabel bookIdLabel = new JLabel("Enter Book ID:");

bookIdField = new JTextField(10);

returnButton = new JButton("Return Book");

backButton = new JButton("Back");

returnButton.addActionListener(e -> returnBook());

backButton.addActionListener(e -> {

new Main().setVisible(true);

dispose();

});

JPanel panel = new JPanel(new GridLayout(3, 1, 10, 10));

panel.setBorder(BorderFactory.createEmptyBorder(30, 30, 30, 30));

panel.add(bookIdLabel);

panel.add(bookIdField);

panel.add(returnButton);

panel.add(backButton);

add(panel);

setTitle("Return Book");

setSize(400, 200);

setLocationRelativeTo(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

private void returnBook() {

int bookId = Integer.parseInt(bookIdField.getText());

try (Connection conn = DatabaseConnection.getConnection()) {

String query = "UPDATE books SET is\_issued = 0 WHERE book\_id = ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setInt(1, bookId);

int rowsUpdated = stmt.executeUpdate();

if (rowsUpdated > 0) {

JOptionPane.showMessageDialog(this, "Book returned successfully!");

} else {

JOptionPane.showMessageDialog(this, "Book ID not found.");

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Error returning book.");

}

}

}

IssueBookPage.java

import javax.swing.\*;

import java.awt.\*;

public class IssueBookPage extends JFrame {

private JTextField bookIdField;

private JTextField userIdField;

private JButton issueButton;

public IssueBookPage() {

initComponents();

}

private void initComponents() {

setTitle("Issue Book");

setSize(400, 300);

setLocationRelativeTo(null);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

JPanel panel = new JPanel(new GridBagLayout());

panel.setBackground(new Color(48, 51, 107));

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(10, 10, 10, 10);

JLabel bookIdLabel = createLabel("Book ID:");

bookIdField = new JTextField(15);

JLabel userIdLabel = createLabel("User ID:");

userIdField = new JTextField(15);

issueButton = createButton("Issue Book", new Color(33, 150, 243));

issueButton.addActionListener(evt -> issueBook());

gbc.gridx = 0; gbc.gridy = 0;

panel.add(bookIdLabel, gbc);

gbc.gridx = 1;

panel.add(bookIdField, gbc);

gbc.gridx = 0; gbc.gridy = 1;

panel.add(userIdLabel, gbc);

gbc.gridx = 1;

panel.add(userIdField, gbc);

gbc.gridwidth = 2; gbc.gridy = 2;

panel.add(issueButton, gbc);

add(panel);

}

private JLabel createLabel(String text) {

JLabel label = new JLabel(text);

label.setFont(new Font("Arial", Font.BOLD, 14));

label.setForeground(Color.WHITE);

return label;

}

private JButton createButton(String text, Color color) {

JButton button = new JButton(text);

button.setBackground(color);

button.setForeground(Color.WHITE);

return button;

}

private void issueBook() {

JOptionPane.showMessageDialog(this, "Book issued successfully!");

}

public static void main(String[] args) {

SwingUtilities.invokeLater(IssueBookPage::new);

}

}

DeleteBookPage.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class DeleteBookPage extends JFrame {

private JTextField bookIdField;

private JButton deleteButton, backButton;

public DeleteBookPage() {

initComponents();

}

private void initComponents() {

JLabel bookIdLabel = new JLabel("Enter Book ID:");

bookIdField = new JTextField(10);

deleteButton = new JButton("Delete Book");

backButton = new JButton("Back");

deleteButton.addActionListener(e -> deleteBook());

backButton.addActionListener(e -> {

new Main().setVisible(true);

dispose();

});

JPanel panel = new JPanel(new GridLayout(3, 1, 10, 10));

panel.setBorder(BorderFactory.createEmptyBorder(30, 30, 30, 30));

panel.add(bookIdLabel);

panel.add(bookIdField);

panel.add(deleteButton);

panel.add(backButton);

add(panel);

setTitle("Delete Book");

setSize(400, 200);

setLocationRelativeTo(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

private void deleteBook() {

int bookId = Integer.parseInt(bookIdField.getText());

try (Connection conn = DatabaseConnection.getConnection()) {

String query = "DELETE FROM books WHERE book\_id = ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setInt(1, bookId);

int rowsDeleted = stmt.executeUpdate();

if (rowsDeleted > 0) {

JOptionPane.showMessageDialog(this, "Book deleted successfully!");

} else {

JOptionPane.showMessageDialog(this, "Book ID not found.");

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Error deleting book.");

}

}

}

DatabaseConnection.java

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

// Simple credentials and URL setup

private static final String URL = "jdbc:mysql://localhost:3306/librarymanagement";

private static final String USER = "root"; // Replace with your MySQL username

private static final String PASSWORD = ""; // Replace with your MySQL password

// Method to establish a database connection

public static Connection getConnection() {

Connection conn = null;

try {

Class.forName("com.mysql.cj.jdbc.Driver");

conn = DriverManager.getConnection(URL, USER, PASSWORD);

} catch (SQLException | ClassNotFoundException e) {

e.printStackTrace(); // or use a logger to log exceptions

}

return conn;

}

public static void main(String[] args) {

Connection conn = getConnection();

if (conn != null) {

System.out.println("Connection established successfully!");

} else {

System.out.println("Failed to establish connection.");

}

}

}