**ABDOUT:**

This Java Swing application simulates an Online Book Store, allowing users to log in, register, view available books, purchase books, or sell their own. It uses a CardLayout for seamless navigation between features, ensuring an intuitive user experience while demonstrating object-oriented programming principles like encapsulation and modular design.

**CODE:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.ArrayList;

import java.util.List;

public class Main {

static class Book {

private String title;

private String author;

private double price;

public Book(String title, String author, double price) {

this.title = title;

this.author = author;

this.price = price;

}

@Override

public String toString() {

return "Title: " + title + ", Author: " + author + ", Price: $" + price;

}

}

static class User {

private String username;

private String password;

private List<Book> purchasedBooks;

public User(String username, String password) {

this.username = username;

this.password = password;

this.purchasedBooks = new ArrayList<>();

}

public void addPurchasedBook(Book book) {

purchasedBooks.add(book);

}

public List<Book> getPurchasedBooks() {

return purchasedBooks;

}

}

private JFrame frame;

private JPanel panel;

private CardLayout cardLayout;

private List<Book> booksForSale;

private List<User> users;

private User loggedInUser;

private JTextField usernameField, bookTitleField, bookAuthorField, bookPriceField;

private JPasswordField passwordField;

private JTextArea displayArea;

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new Main().createAndShowGUI());

}

public Main() {

booksForSale = new ArrayList<>();

users = new ArrayList<>();

booksForSale.add(new Book("Java Programming", "John Doe", 29.99));

booksForSale.add(new Book("Effective Java", "Joshua Bloch", 45.00));

booksForSale.add(new Book("Clean Code", "Robert C. Martin", 40.50));

users.add(new User("john\_doe", "password123"));

users.add(new User("jane\_doe", "password456"));

}

public void createAndShowGUI() {

frame = new JFrame("Online Book Store");

panel = new JPanel(new CardLayout());

cardLayout = (CardLayout) panel.getLayout();

panel.add(createLoginPanel(), "Login");

panel.add(createMainMenuPanel(), "Main Menu");

panel.add(createViewBooksPanel(), "View Books");

panel.add(createBuyBookPanel(), "Buy Book");

panel.add(createSellBookPanel(), "Sell Book");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(600, 400);

frame.add(panel);

frame.setVisible(true);

}

private JPanel createLoginPanel() {

JPanel loginPanel = new JPanel(new GridLayout(6, 1));

usernameField = new JTextField();

passwordField = new JPasswordField();

JButton loginButton = new JButton("Login");

loginButton.addActionListener(e -> handleLogin());

JButton registerButton = new JButton("Register");

registerButton.addActionListener(e -> handleRegister());

loginPanel.add(new JLabel("Username: "));

loginPanel.add(usernameField);

loginPanel.add(new JLabel("Password: "));

loginPanel.add(passwordField);

loginPanel.add(loginButton);

loginPanel.add(registerButton);

return loginPanel;

}

private void handleLogin() {

String username = usernameField.getText();

String password = new String(passwordField.getPassword());

for (User user : users) {

if (user.username.equals(username) && user.password.equals(password)) {

loggedInUser = user;

cardLayout.show(panel, "Main Menu");

return;

}

}

JOptionPane.showMessageDialog(frame, "Invalid credentials.");

}

private void handleRegister() {

String username = usernameField.getText();

String password = new String(passwordField.getPassword());

if (username.isEmpty() || password.isEmpty()) {

JOptionPane.showMessageDialog(frame, "Fields cannot be empty.");

return;

}

users.add(new User(username, password));

JOptionPane.showMessageDialog(frame, "Registration successful.");

}

private JPanel createMainMenuPanel() {

JPanel menuPanel = new JPanel(new GridLayout(4, 1));

JButton viewBooksButton = new JButton("View Available Books");

viewBooksButton.addActionListener(e -> cardLayout.show(panel, "View Books"));

JButton buyBookButton = new JButton("Buy a Book");

buyBookButton.addActionListener(e -> cardLayout.show(panel, "Buy Book"));

JButton sellBookButton = new JButton("Sell a Book");

sellBookButton.addActionListener(e -> cardLayout.show(panel, "Sell Book"));

JButton logoutButton = new JButton("Logout");

logoutButton.addActionListener(e -> cardLayout.show(panel, "Login"));

menuPanel.add(viewBooksButton);

menuPanel.add(buyBookButton);

menuPanel.add(sellBookButton);

menuPanel.add(logoutButton);

return menuPanel;

}

private JPanel createViewBooksPanel() {

JPanel viewBooksPanel = new JPanel();

viewBooksPanel.setLayout(new BoxLayout(viewBooksPanel, BoxLayout.Y\_AXIS));

displayArea = new JTextArea(10, 40);

displayArea.setEditable(false);

refreshBooksDisplay();

JButton backButton = new JButton("Back to Main Menu");

backButton.addActionListener(e -> cardLayout.show(panel, "Main Menu"));

viewBooksPanel.add(new JScrollPane(displayArea));

viewBooksPanel.add(backButton);

return viewBooksPanel;

}

private void refreshBooksDisplay() {

displayArea.setText("Available Books:\n");

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

displayArea.append((i + 1) + ". " + booksForSale.get(i) + "\n");

}

}

private JPanel createBuyBookPanel() {

JPanel buyBookPanel = new JPanel();

buyBookPanel.setLayout(new BoxLayout(buyBookPanel, BoxLayout.Y\_AXIS));

JTextField bookSelectionField = new JTextField(10);

JButton buyButton = new JButton("Buy Book");

buyButton.addActionListener(e -> {

try {

int bookNo = Integer.parseInt(bookSelectionField.getText());

if (bookNo < 1 || bookNo > booksForSale.size()) {

JOptionPane.showMessageDialog(frame, "Invalid book number.");

return;

}

Book selectedBook = booksForSale.get(bookNo - 1);

loggedInUser.addPurchasedBook(selectedBook);

JOptionPane.showMessageDialog(frame, "You bought: " + selectedBook);

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(frame, "Enter a valid number.");

}

});

JButton backButton = new JButton("Back to Main Menu");

backButton.addActionListener(e -> cardLayout.show(panel, "Main Menu"));

buyBookPanel.add(new JLabel("Enter the book number:"));

buyBookPanel.add(bookSelectionField);

buyBookPanel.add(buyButton);

buyBookPanel.add(backButton);

return buyBookPanel;

}

private JPanel createSellBookPanel() {

JPanel sellBookPanel = new JPanel();

sellBookPanel.setLayout(new BoxLayout(sellBookPanel, BoxLayout.Y\_AXIS));

bookTitleField = new JTextField(20);

bookAuthorField = new JTextField(20);

bookPriceField = new JTextField(20);

JButton sellButton = new JButton("Sell Book");

sellButton.addActionListener(e -> {

try {

String title = bookTitleField.getText();

String author = bookAuthorField.getText();

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

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

JOptionPane.showMessageDialog(frame, "All fields must be filled.");

return;

}

booksForSale.add(new Book(title, author, price));

refreshBooksDisplay();

JOptionPane.showMessageDialog(frame, "Book added.");

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(frame, "Invalid price.");

}

});

JButton backButton = new JButton("Back to Main Menu");

backButton.addActionListener(e -> cardLayout.show(panel, "Main Menu"));

sellBookPanel.add(new JLabel("Book Title:"));

sellBookPanel.add(bookTitleField);

sellBookPanel.add(new JLabel("Book Author:"));

sellBookPanel.add(bookAuthorField);

sellBookPanel.add(new JLabel("Book Price:"));

sellBookPanel.add(bookPriceField);

sellBookPanel.add(sellButton);

sellBookPanel.add(backButton);

return sellBookPanel;

}

}