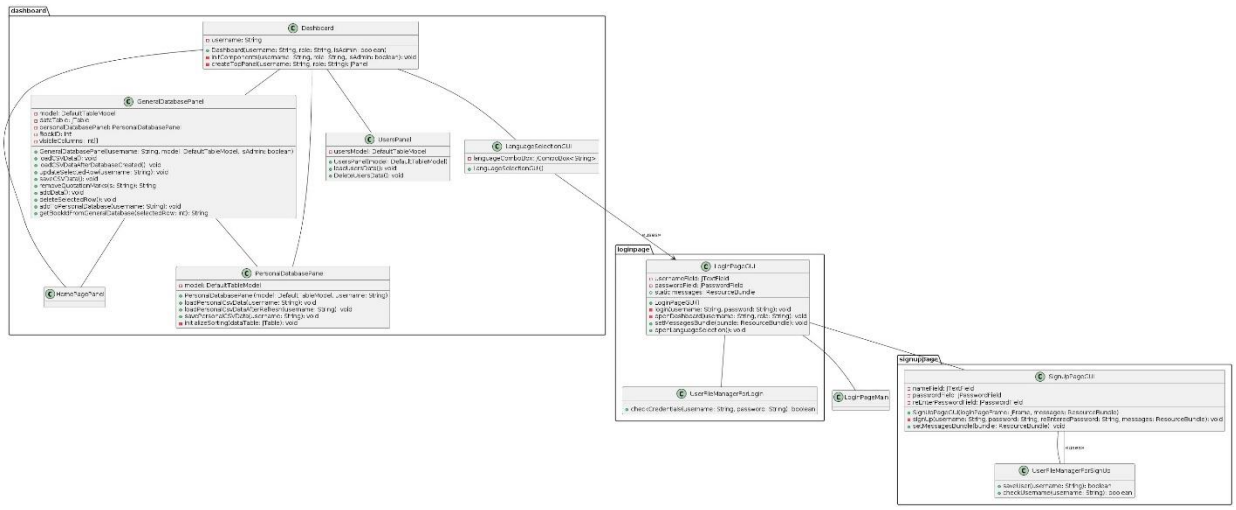
Team Members	Contributions	Contribution Percentage
Farid Karimli	Farid created Dashboard, General database, Personal database, HomePage, UsersPanel, and language selection class.	29%
Ali Hasanli	He helped to create LoginPageGUI, ProgramStart. UserFileManagerForLoginPageClass,	28%
Ismayil Panahov	He helped to create SignUpPageGUI, UserFileManagerForSignUp class and Readme File.	28%
Kanan Abilov	He prepared report of project in pdf and docx format and helped to fix some bugs in code.	15%

Video Presentation Link: [20240510_170619000_iOS.mov]

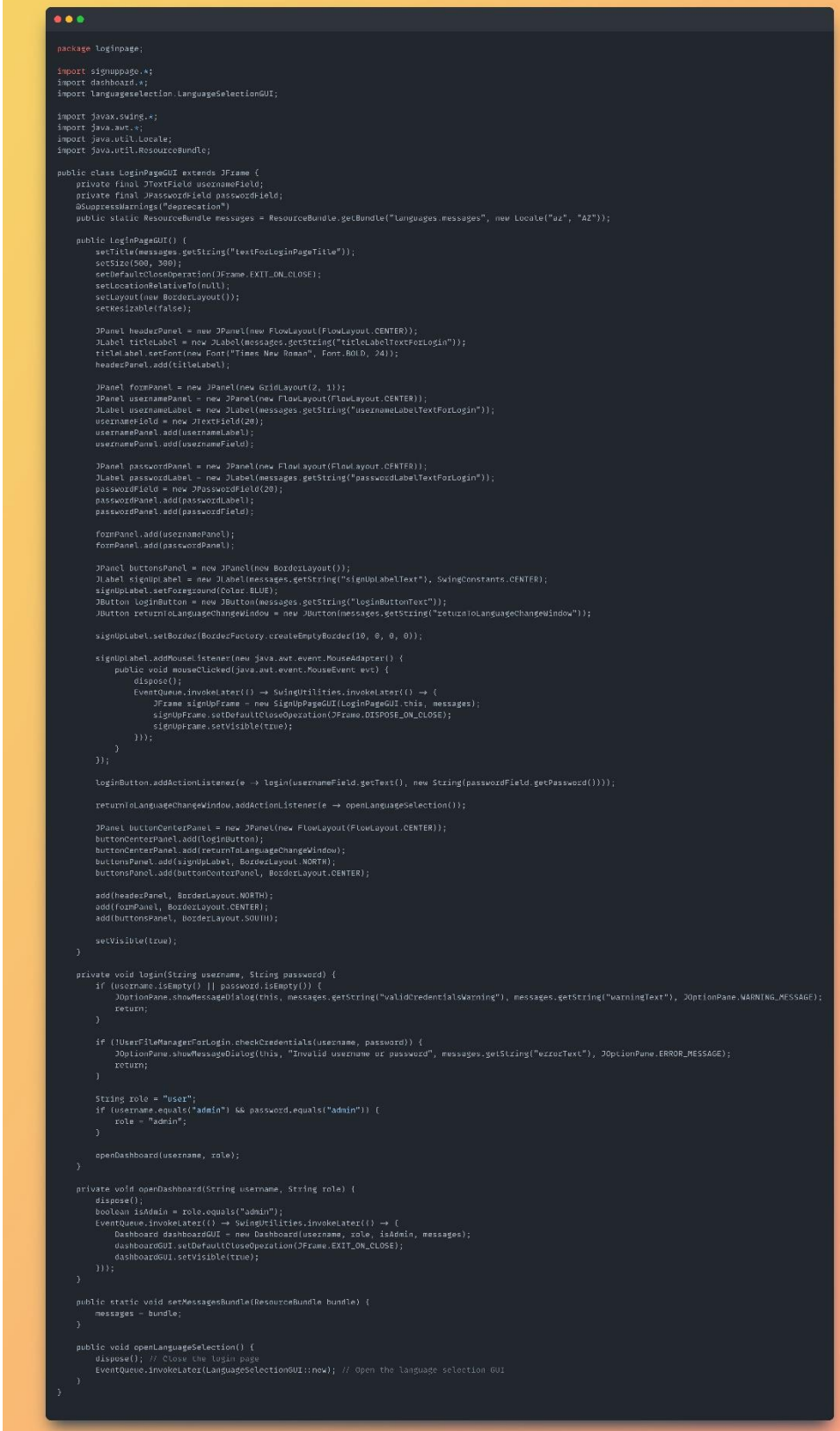
GitHub Link: https://github.com/ADA-SITE-CSCI-1202/team-project-team-44

UML Diagram of our project



Code Snippets:

LoginPageGUI.java Class:

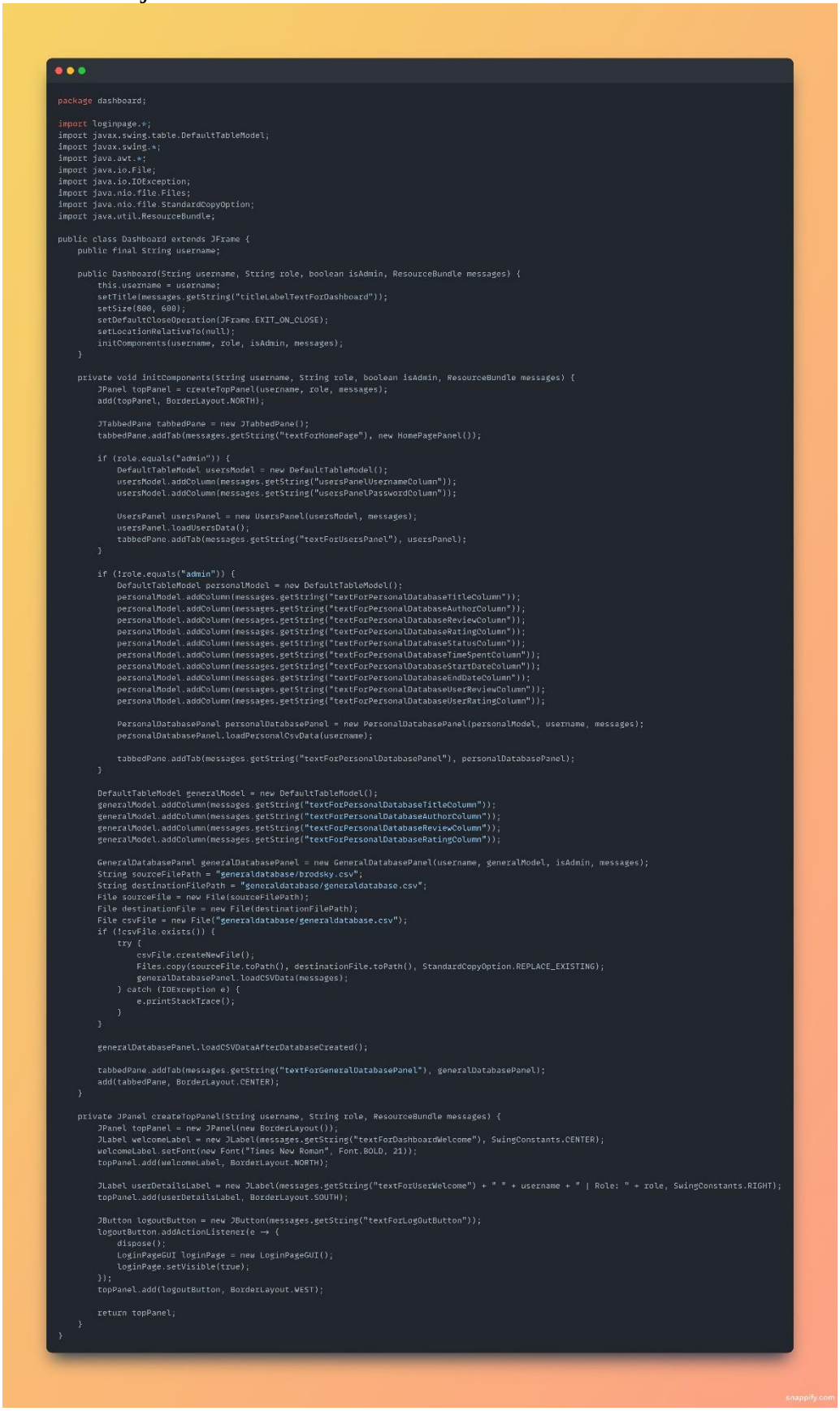


```java
package loginpage;

import signuppage.*;
import dashboard.*;
import languageselection.LanguageSelectionGUI;

import javax.swing.*;
import java.awt.*;
import java.util.Locale;
import java.util.ResourceBundle;

public class LoginPageGUI extends JFrame {
    private final JTextField usernameField;
    private final JPasswordField passwordField;
    @SuppressWarnings("deprecation")
    public static ResourceBundle messages = ResourceBundle.getBundle("languages.messages", new Locale("az", "AZ"));

    public LoginPageGUI() {
        setTitle(messages.getString("textForLoginPageTitle"));
        setSize(500, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());
        setResizable(false);

        JPanel headerPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        JLabel titleLabel = new JLabel(messages.getString("titleLabelTextForLogin"));
        titleLabel.setFont(new Font("Times New Roman", Font.BOLD, 24));
        headerPanel.add(titleLabel);

        JPanel formPanel = new JPanel(new GridLayout(2, 1));
        JPanel usernamePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        JLabel usernameLabel = new JLabel(messages.getString("usernameLabelTextForLogin"));
        usernameField = new JTextField(20);
        usernamePanel.add(usernameLabel);
        usernamePanel.add(usernameField);

        JPanel passwordPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        JLabel passwordLabel = new JLabel(messages.getString("passwordLabelTextForLogin"));
        passwordField = new JPasswordField(20);
        passwordPanel.add(passwordLabel);
        passwordPanel.add(passwordField);

        formPanel.add(usernamePanel);
        formPanel.add(passwordPanel);

        JPanel buttonsPanel = new JPanel(new BorderLayout());
        JLabel signUpLabel = new JLabel(messages.getString("signUpLabelText"), SwingConstants.CENTER);
        signUpLabel.setForeground(Color.BLUE);
        JButton loginButton = new JButton(messages.getString("loginButtonText"));
        JButton returnToLanguageChangeWindow = new JButton(messages.getString("returnToLanguageChangeWindow"));

        signUpLabel.setBorder(BorderFactory.createEmptyBorder(10, 0, 0, 0));

        signUpLabel.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                dispose();
                EventQueue.invokeLater(() -> SwingUtilities.invokeLater(() -> {
                    JFrame signUpFrame = new SignUpPageGUI(LoginPageGUI.this, messages);
                    signUpFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                    signUpFrame.setVisible(true);
                }));
            }
        });

        loginButton.addActionListener(e -> login(usernameField.getText(), new String(passwordField.getPassword())));

        returnToLanguageChangeWindow.addActionListener(e -> openLanguageSelection());

        JPanel buttonCenterPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        buttonCenterPanel.add(loginButton);
        buttonCenterPanel.add(returnToLanguageChangeWindow);
        buttonsPanel.add(signUpLabel, BorderLayout.NORTH);
        buttonsPanel.add(buttonCenterPanel, BorderLayout.CENTER);

        add(headerPanel, BorderLayout.NORTH);
        add(formPanel, BorderLayout.CENTER);
        add(buttonsPanel, BorderLayout.SOUTH);

        setVisible(true);
    }

    private void login(String username, String password) {
        if (username.isEmpty() || password.isEmpty()) {
            JOptionPane.showMessageDialog(this, messages.getString("validCredentialsWarning"), messages.getString("warningText"), JOptionPane.WARNING_MESSAGE);
            return;
        }

        if (!UserFileManagerForLogin.checkCredentials(username, password)) {
            JOptionPane.showMessageDialog(this, "Invalid username or password", messages.getString("errorText"), JOptionPane.ERROR_MESSAGE);
            return;
        }

        String role = "user";
        if (username.equals("admin") && password.equals("admin")) {
            role = "admin";
        }

        openDashboard(username, role);
    }

    private void openDashboard(String username, String role) {
        dispose();
        boolean isAdmin = role.equals("admin");
        EventQueue.invokeLater(() -> SwingUtilities.invokeLater(() -> {
            Dashboard dashboardGUI = new Dashboard(username, role, isAdmin, messages);
            dashboardGUI.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            dashboardGUI.setVisible(true);
        }));
    }

    public static void setMessagesBundle(ResourceBundle bundle) {
        messages = bundle;
    }

    public void openLanguageSelection() {
        dispose(); // Close the login page
        EventQueue.invokeLater(LanguageSelectionGUI::new); // Open the language selection GUI
    }
}
```


LanguageSelectionGUI.java Class:

```java
package languageselection;

import javax.swing.*;

import loginpage.LoginPageGUI;

import java.awt.*;
import java.util.Locale;
import java.util.ResourceBundle;

public class LanguageSelectionGUI extends JFrame {
    private final JComboBox<String> languageComboBox;

    @SuppressWarnings("deprecation")
    public LanguageSelectionGUI() {
        setTitle("Language Selection");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());
        setResizable(false);

        JPanel languagePanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        JLabel languageLabel = new JLabel("Select Language:");
        languageComboBox = new JComboBox<>(new String[] { "English", "Azerbaijani" });
        languagePanel.add(languageLabel);
        languagePanel.add(languageComboBox);

        JButton confirmButton = new JButton("Confirm");
        confirmButton.addActionListener(e -> {
            String selectedLanguage = (String) languageComboBox.getSelectedItem();
            Locale newLocale = selectedLanguage.equals("English") ? new Locale("en", "US") : new Locale("az", "AZ");
            LoginPageGUI.setMessagesBundle(ResourceBundle.getBundle("languages.messages", newLocale));
            dispose();
            EventQueue.invokeLater(LoginPageGUI::new);
        });

        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        buttonPanel.add(confirmButton);

        add(languagePanel, BorderLayout.CENTER);
        add(buttonPanel, BorderLayout.SOUTH);

        setVisible(true);
    }
}
```

Dashboard.java Class:



```java
package dashboard;

import loginpage.*;
import javax.swing.table.DefaultTableModel;
import javax.swing.*;
import java.awt.*;
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.StandardCopyOption;
import java.util.ResourceBundle;

public class Dashboard extends JFrame {
    public final String username;

    public Dashboard(String username, String role, boolean isAdmin, ResourceBundle messages) {
        this.username = username;
        setTitle(messages.getString("titleLabelTextForDashboard"));
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        initComponents(username, role, isAdmin, messages);
    }

    private void initComponents(String username, String role, boolean isAdmin, ResourceBundle messages) {
        JPanel topPanel = createTopPanel(username, role, messages);
        add(topPanel, BorderLayout.NORTH);

        JTabbedPane tabbedPane = new JTabbedPane();
        tabbedPane.addTab(messages.getString("textForHomePage"), new HomePagePanel());

        if (role.equals("admin")) {
            DefaultTableModel usersModel = new DefaultTableModel();
            usersModel.addColumn(messages.getString("usersPanelUsernameColumn"));
            usersModel.addColumn(messages.getString("usersPanelPasswordColumn"));

            UsersPanel usersPanel = new UsersPanel(usersModel, messages);
            usersPanel.loadUsersData();
            tabbedPane.addTab(messages.getString("textForUsersPanel"), usersPanel);
        }

        if (!role.equals("admin")) {
            DefaultTableModel personalModel = new DefaultTableModel();
            personalModel.addColumn(messages.getString("textForPersonalDatabaseTitleColumn"));
            personalModel.addColumn(messages.getString("textForPersonalDatabaseAuthorColumn"));
            personalModel.addColumn(messages.getString("textForPersonalDatabaseReviewColumn"));
            personalModel.addColumn(messages.getString("textForPersonalDatabaseRatingColumn"));
            personalModel.addColumn(messages.getString("textForPersonalDatabaseStatusColumn"));
            personalModel.addColumn(messages.getString("textForPersonalDatabaseTimeSpentColumn"));
            personalModel.addColumn(messages.getString("textForPersonalDatabaseStartDateColumn"));
            personalModel.addColumn(messages.getString("textForPersonalDatabaseEndDateColumn"));
            personalModel.addColumn(messages.getString("textForPersonalDatabaseUserReviewColumn"));
            personalModel.addColumn(messages.getString("textForPersonalDatabaseUserRatingColumn"));

            PersonalDatabasePanel personalDatabasePanel = new PersonalDatabasePanel(personalModel, username, messages);
            personalDatabasePanel.loadPersonalCsvData(username);

            tabbedPane.addTab(messages.getString("textForPersonalDatabasePanel"), personalDatabasePanel);
        }

        DefaultTableModel generalModel = new DefaultTableModel();
        generalModel.addColumn(messages.getString("textForPersonalDatabaseTitleColumn"));
        generalModel.addColumn(messages.getString("textForPersonalDatabaseAuthorColumn"));
        generalModel.addColumn(messages.getString("textForPersonalDatabaseReviewColumn"));
        generalModel.addColumn(messages.getString("textForPersonalDatabaseRatingColumn"));

        GeneralDatabasePanel generalDatabasePanel = new GeneralDatabasePanel(username, generalModel, isAdmin, messages);
        String sourceFilePath = "generaldatabase/brodsky.csv";
        String destinationFilePath = "generaldatabase/generaldatabase.csv";
        File sourceFile = new File(sourceFilePath);
        File destinationFile = new File(destinationFilePath);
        File csvFile = new File("generaldatabase/generaldatabase.csv");
        if (!csvFile.exists()) {
            try {
                csvFile.createNewFile();
                Files.copy(sourceFile.toPath(), destinationFile.toPath(), StandardCopyOption.REPLACE_EXISTING);
                generalDatabasePanel.loadCSVData(messages);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        generalDatabasePanel.loadCSVDataAfterDatabaseCreated();

        tabbedPane.addTab(messages.getString("textForGeneralDatabasePanel"), generalDatabasePanel);
        add(tabbedPane, BorderLayout.CENTER);
    }

    private JPanel createTopPanel(String username, String role, ResourceBundle messages) {
        JPanel topPanel = new JPanel(new BorderLayout());
        JLabel welcomeLabel = new JLabel(messages.getString("textForDashboardWelcome"), SwingConstants.CENTER);
        welcomeLabel.setFont(new Font("Times New Roman", Font.BOLD, 21));
        topPanel.add(welcomeLabel, BorderLayout.NORTH);

        JLabel userDetailsLabel = new JLabel(messages.getString("textForUserWelcome") + " " + username + " | Role: " + role, SwingConstants.RIGHT);
        topPanel.add(userDetailsLabel, BorderLayout.SOUTH);

        JButton logoutButton = new JButton(messages.getString("textForLogOutButton"));
        logoutButton.addActionListener(e -> {
            dispose();
            LoginPageGUI loginPage = new LoginPageGUI();
            loginPage.setVisible(true);
        });
        topPanel.add(logoutButton, BorderLayout.WEST);

        return topPanel;
    }
}
```


GeneralDatabasePanel.java Class:

```java
package dashboard;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;
import javax.swing.table.TableModel;
import java.awt.*;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ResourceBundle;

import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;

public class GeneralDatabasePanel extends JPanel {
    public final DefaultTableModel model;
    public final JTable dataTable;
    private PersonalDatabasePanel personalDatabasePanel;
    private int BookID = 1;
    private final int[] visibleColumns = { 1, 2, 3, 4 };

    public GeneralDatabasePanel(String username, DefaultTableModel model, boolean isAdmin, ResourceBundle messages) {
        this.model = model;
        setLayout(new BorderLayout());

        dataTable = new JTable(model);
        JScrollPane scrollPane = new JScrollPane(dataTable);
        add(scrollPane, BorderLayout.CENTER);
        initializeSorting();

        JPanel buttonPanel = new JPanel();
        JTextField searchField = new JTextField(20);
        JButton searchButton = new JButton(messages.getString("textForSearchButton"));

        if (isAdmin) {
            JButton addButton = new JButton(messages.getString("textForAddButton"));
            buttonPanel.add(addButton);
            addButton.addActionListener(e -> addData(messages));

            JButton deleteButton = new JButton(messages.getString("textForDeleteButton"));
            buttonPanel.add(deleteButton);
            deleteButton.addActionListener(e -> deleteSelectedRow(messages));

            JButton updateButton = new JButton(messages.getString("textForUpdateButton"));
            buttonPanel.add(updateButton);
            updateButton.addActionListener(e -> updateSelectedRow(username, messages));
        } else {
            JButton addToPersonalDatabaseButton = new JButton(messages.getString("textForAddToPersonalDatabaseButton"));
            buttonPanel.add(addToPersonalDatabaseButton);
            addToPersonalDatabaseButton.addActionListener(e -> addToPersonalDatabase(username, messages));
        }

        buttonPanel.add(searchField);
        buttonPanel.add(searchButton);
        add(buttonPanel, BorderLayout.NORTH);

        searchButton.addActionListener(e -> {
            String searchText = searchField.getText().trim().toLowerCase();
            if (!searchText.isEmpty()) {
                TableRowSorter<TableModel> sorter = new TableRowSorter<>(model);
                dataTable.setRowSorter(sorter);
                sorter.setRowFilter(RowFilter.regexFilter("(?i)" + searchText));
            } else {
                dataTable.setRowSorter(null);
            }
        });

        searchField.getDocument().addDocumentListener(new DocumentListener() {
            @Override
            public void insertUpdate(DocumentEvent e) {
                filterTable();
            }

            @Override
            public void removeUpdate(DocumentEvent e) {
                filterTable();
            }

            @Override
            public void changedUpdate(DocumentEvent e) {
                filterTable();
            }

            private void filterTable() {
                String searchText = searchField.getText().trim().toLowerCase();
                TableRowSorter<TableModel> sorter = new TableRowSorter<>(model);
                dataTable.setRowSorter(sorter);
                if (!searchText.isEmpty()) {
                    sorter.setRowFilter(RowFilter.regexFilter("(?i)" + searchText));
                } else {
                    dataTable.setRowSorter(null);
                }
            }
        });
    }
}
```


PersonalDatabasePanel.java Class:

```java
package dashboard;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;
import javax.swing.table.TableModel;
import java.awt.*;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ResourceBundle;

import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;

public class PersonalDatabasePanel extends JPanel {
    public final DefaultTableModel model;

    public PersonalDatabasePanel(DefaultTableModel model, String username, ResourceBundle messages) {
        this.model = model;
        setLayout(new BorderLayout());

        JTable dataTable = new JTable(model);
        JScrollPane scrollPane = new JScrollPane(dataTable);
        add(scrollPane, BorderLayout.CENTER);

        JPanel buttonPanel = new JPanel();
        JButton startButton = new JButton(messages.getString("textForPersonalDatabaseStartButton"));
        JButton endButton = new JButton(messages.getString("textForPersonalDatabaseEndButton"));
        JButton deleteButton = new JButton(messages.getString("textForPersonalDatabaseDeleteButton"));
        JButton refreshButton = new JButton("Refresh");
        JTextField searchField = new JTextField(20);
        JButton searchButton = new JButton(messages.getString("textForPersonalDatabaseSearchButton"));

        buttonPanel.add(startButton);
        buttonPanel.add(endButton);
        buttonPanel.add(deleteButton);
        buttonPanel.add(refreshButton);
        buttonPanel.add(searchField);
        buttonPanel.add(searchButton);

        add(buttonPanel, BorderLayout.NORTH);

        refreshButton.addActionListener(e -> {
            loadPersonalCsvDataAfterRefresh(username);
        });

        deleteButton.addActionListener(e -> {
            int selectedRow = dataTable.getSelectedRow();
            if (selectedRow == -1) {
                JOptionPane.showMessageDialog(this, messages.getString("textForUserDeleteError"), messages.getString("errorText"), JOptionPane.ERROR_MESSAGE);
                return;
            }

            int confirmDelete = JOptionPane.showConfirmDialog(this, "Are you sure you want to delete this row?",
                    "Confirm Deletion", JOptionPane.YES_NO_OPTION);
            if (confirmDelete == JOptionPane.YES_OPTION) {
                model.removeRow(selectedRow);
                savePersonalCSVData(username, messages);
            }
        });

        startButton.addActionListener(e -> {

        });

        searchButton.addActionListener(e -> {
            String searchText = searchField.getText().trim().toLowerCase();
            if (!searchText.isEmpty()) {
                TableRowSorter<TableModel> sorter = new TableRowSorter<>(model);
                dataTable.setRowSorter(sorter);
                sorter.setRowFilter(RowFilter.regexFilter("(?i)" + searchText));
            } else {
                dataTable.setRowSorter(null);
            }
        });

        searchField.getDocument().addDocumentListener(new DocumentListener() {
            @Override
            public void insertUpdate(DocumentEvent e) {
                filterTable();
            }

            @Override
            public void removeUpdate(DocumentEvent e) {
                filterTable();
            }

            @Override
            public void changedUpdate(DocumentEvent e) {
                filterTable();
            }

            private void filterTable() {
                String searchText = searchField.getText().trim().toLowerCase();
                TableRowSorter<TableModel> sorter = new TableRowSorter<>(model);
                dataTable.setRowSorter(sorter);
                if (!searchText.isEmpty()) {
                    sorter.setRowFilter(RowFilter.regexFilter("(?i)" + searchText));
                } else {
                    dataTable.setRowSorter(null);
                }
            }
        });
        initializeSorting(dataTable);
    }
```

UserPanel.java Class:

```java
package dashboard;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ResourceBundle;

public class UsersPanel extends JPanel {
    private DefaultTableModel usersModel;

    public UsersPanel(DefaultTableModel model, ResourceBundle messages) {
        this.usersModel = model;
        JTable usersTable = new JTable(usersModel);
        JScrollPane scrollPane = new JScrollPane(usersTable);
        add(scrollPane);

        JPanel buttonPanel = new JPanel();
        JButton deleteButton = new JButton(messages.getString("textForDeleteButton"));
        buttonPanel.add(deleteButton);

        add(buttonPanel, BorderLayout.NORTH);

        deleteButton.addActionListener(e -> {
            int selectedRow = usersTable.getSelectedRow(); // Get selected row index
            if (selectedRow != -1) {
                model.removeRow(selectedRow); // Remove selected row from personal database
                DeleteUsersData();
            } else {
                JOptionPane.showMessageDialog(this, messages.getString("textForUserDeleteError"), messages.getString("errorText"), JOptionPane.ERROR_MESSAGE);
            }
        });

    }

    public void loadUsersData() {
        try (BufferedReader br = new BufferedReader(new FileReader("users.csv"))) {
            String line;
            while ((line = br.readLine()) != null) {
                String[] data = line.split(",");
                if (data.length >= 2) {
                    String username = data[0].trim();
                    String role = data[1].trim();
                    usersModel.addRow(new Object[]{username, role});
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void DeleteUsersData() {
        try (FileWriter fw = new FileWriter("users.csv")) {
            for (int i = 0; i < usersModel.getRowCount(); i++) {
                String username = usersModel.getValueAt(i, 0).toString();
                String password = usersModel.getValueAt(i, 1).toString();
                fw.write(username + "," + password + "\n");
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

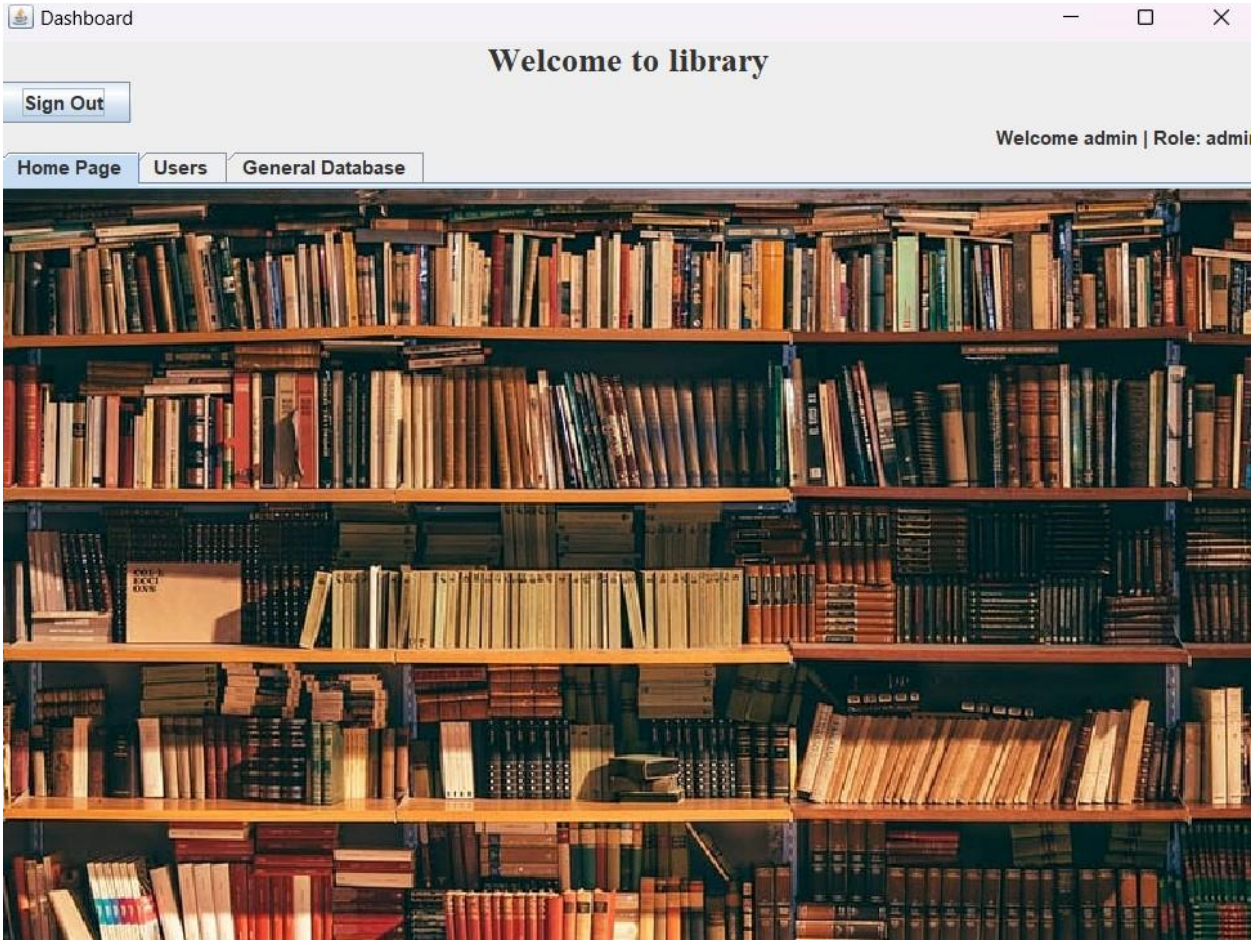
UI Snippets:

Language Selection:

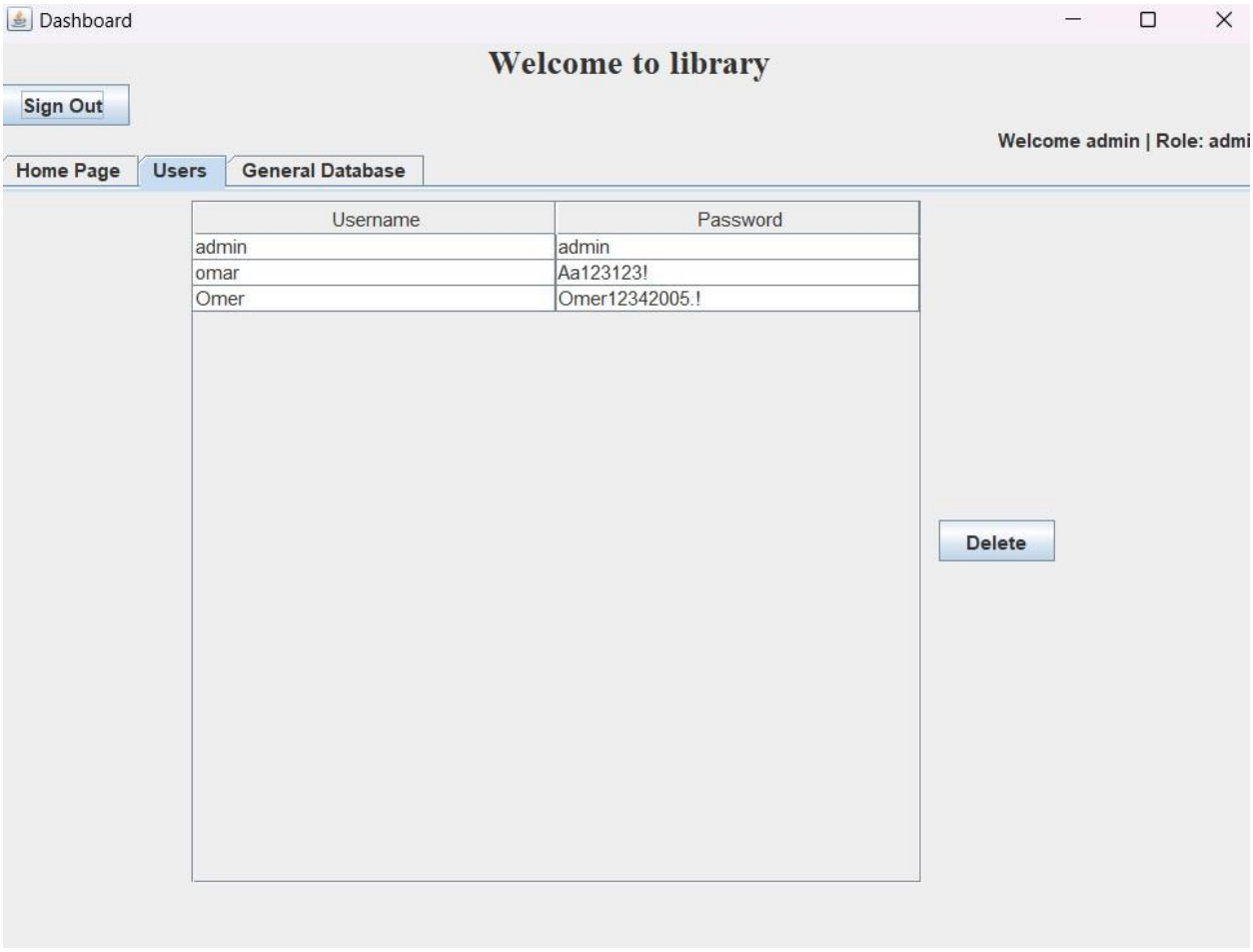
Login Page:

Sign Up Page:

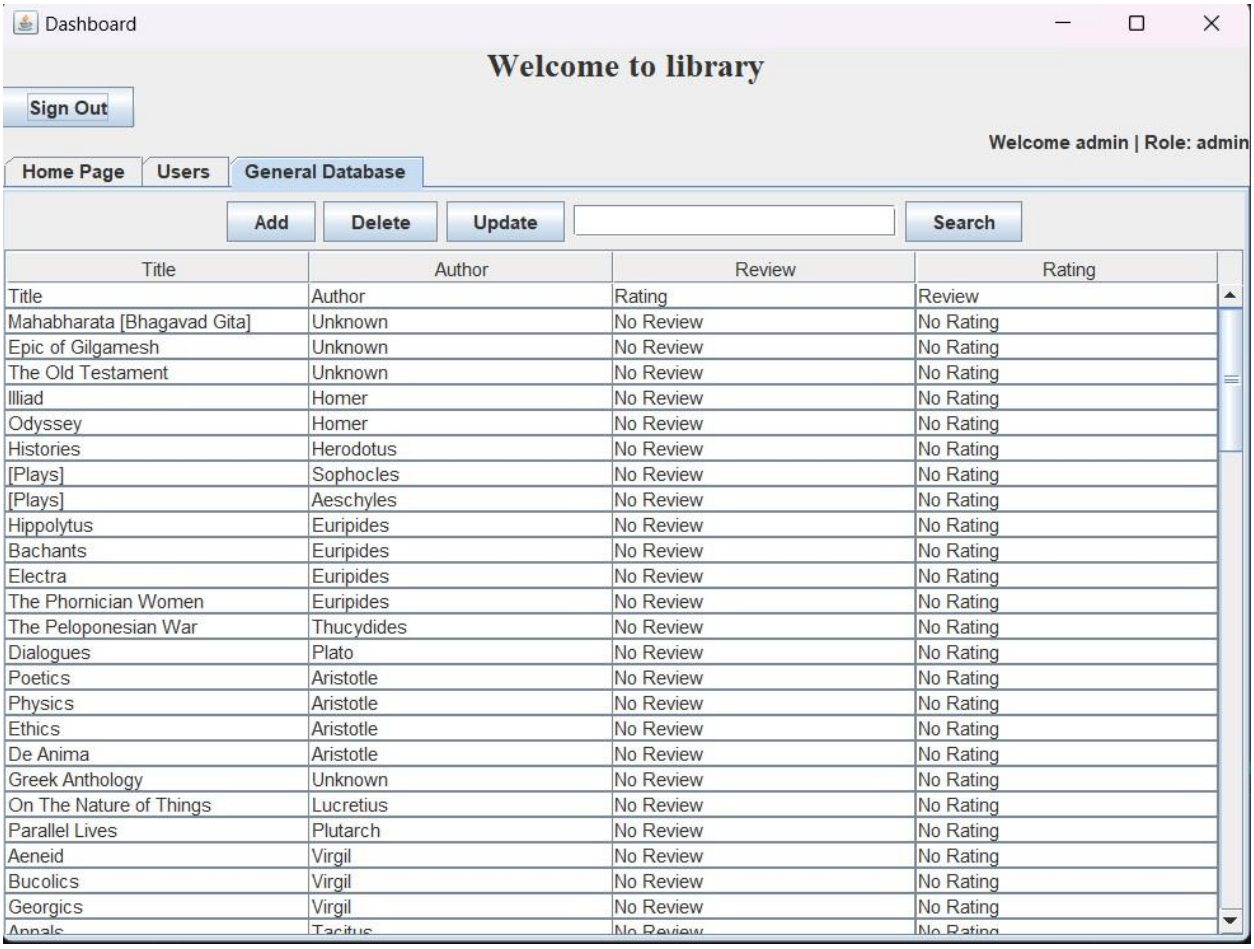
Home Page:



Users Panel:

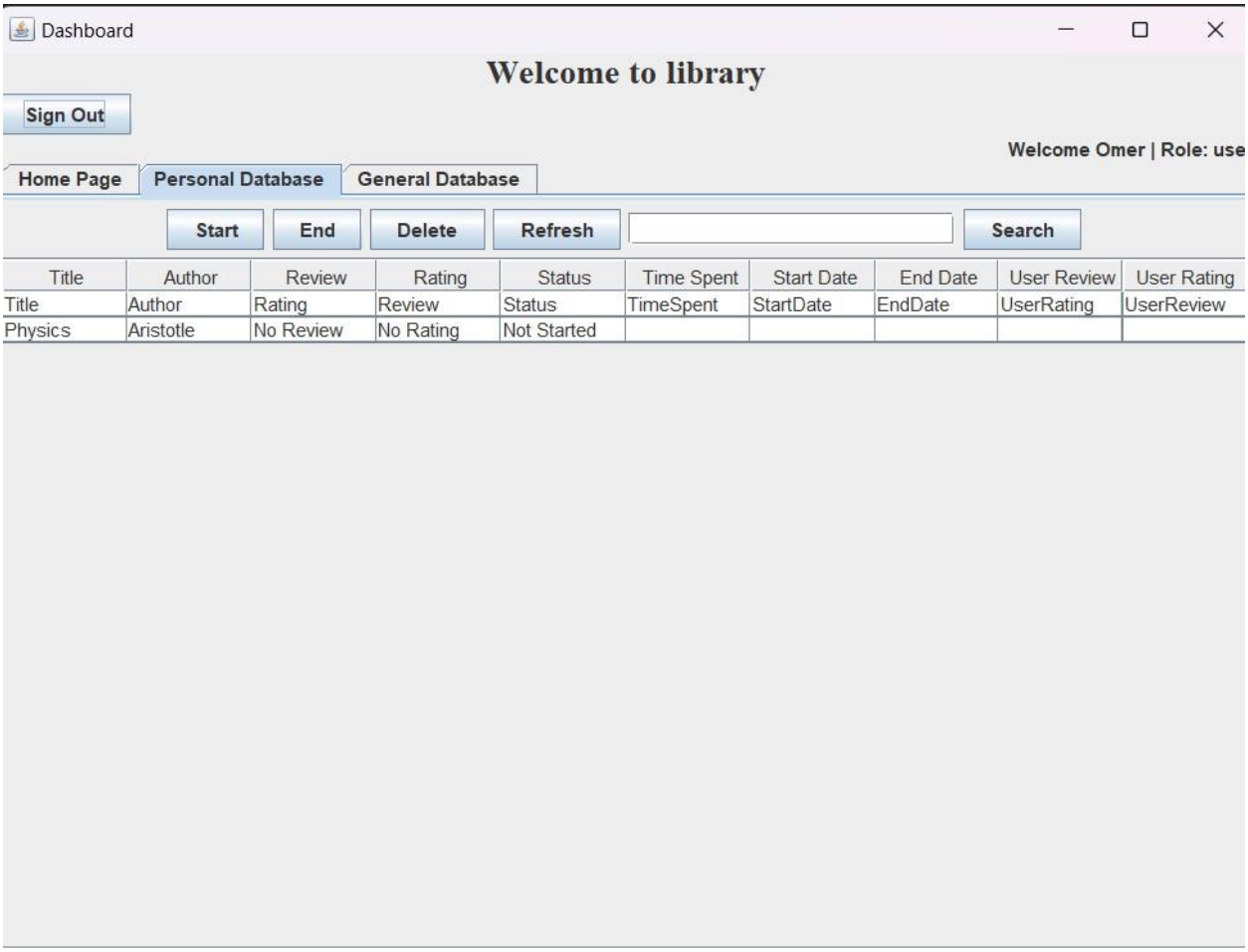

Dashboard
Welcome to library
Sign Out
Welcome admin | Role: admi
Home Page
Users
General Database
Username
Password
admin
admin
omar
Aa123123!
Omer
Omer12342005.!
Delete

General Database Panel:



Title	Author	Review	Rating
Title	Author	Rating	Review
Mahabharata [Bhagavad Gita]	Unknown	No Review	No Rating
Epic of Gilgamesh	Unknown	No Review	No Rating
The Old Testament	Unknown	No Review	No Rating
Illiad	Homer	No Review	No Rating
Odyssey	Homer	No Review	No Rating
Histories	Herodotus	No Review	No Rating
[Plays]	Sophocles	No Review	No Rating
[Plays]	Aeschyles	No Review	No Rating
Hippolytus	Euripides	No Review	No Rating
Bachants	Euripides	No Review	No Rating
Electra	Euripides	No Review	No Rating
The Phornician Women	Euripides	No Review	No Rating
The Peloponesian War	Thucydides	No Review	No Rating
Dialogues	Plato	No Review	No Rating
Poetics	Aristotle	No Review	No Rating
Physics	Aristotle	No Review	No Rating
Ethics	Aristotle	No Review	No Rating
De Anima	Aristotle	No Review	No Rating
Greek Anthology	Unknown	No Review	No Rating
On The Nature of Things	Lucretius	No Review	No Rating
Parallel Lives	Plutarch	No Review	No Rating
Aeneid	Virgil	No Review	No Rating
Bucolics	Virgil	No Review	No Rating
Georgics	Virgil	No Review	No Rating
Annals	Tacitus	No Review	No Rating

Personal Database Panel:


Dashboard
Welcome to library
Sign Out
Welcome Omer | Role: use
Home Page
Personal Database
General Database
Start
End
Delete
Refresh
Search
Title
Author
Review
Rating
Status
Time Spent
Start Date
End Date
User Review
User Rating
Title
Author
Rating
Review
Status
TimeSpent
StartDate
EndDate
UserRating
UserReview
Physics
Aristotle
No Review
No Rating
Not Started