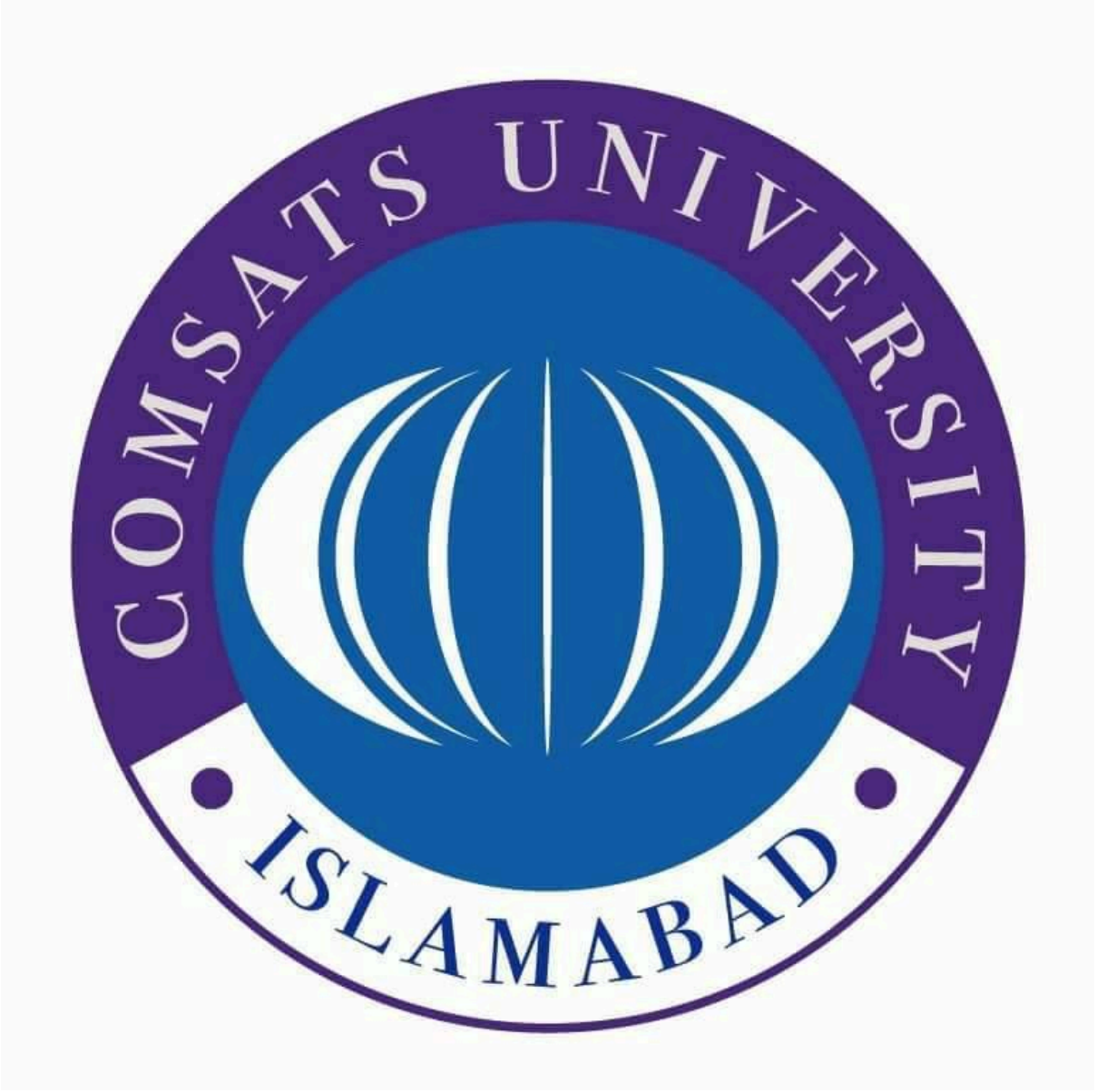


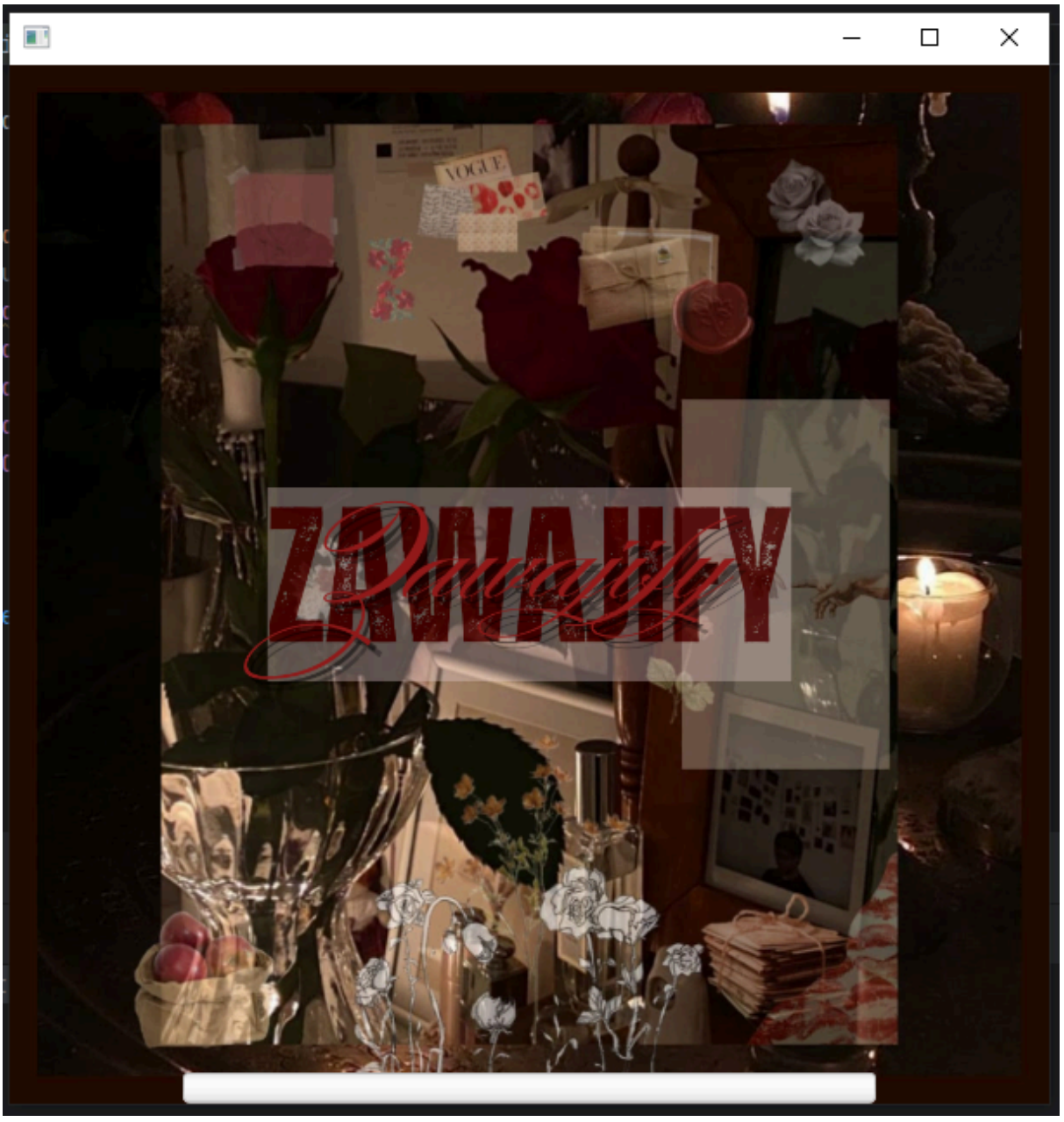
Object Oriented Programming
Lab Assignment 3

Submitted to: Sir Shahid Bhatti
Submitted by: Sheheryar Tariq - SP24-BSE-112 and Khadija Noor SP24-BSE-052

Project:
Marriage Application (Partially Completed)

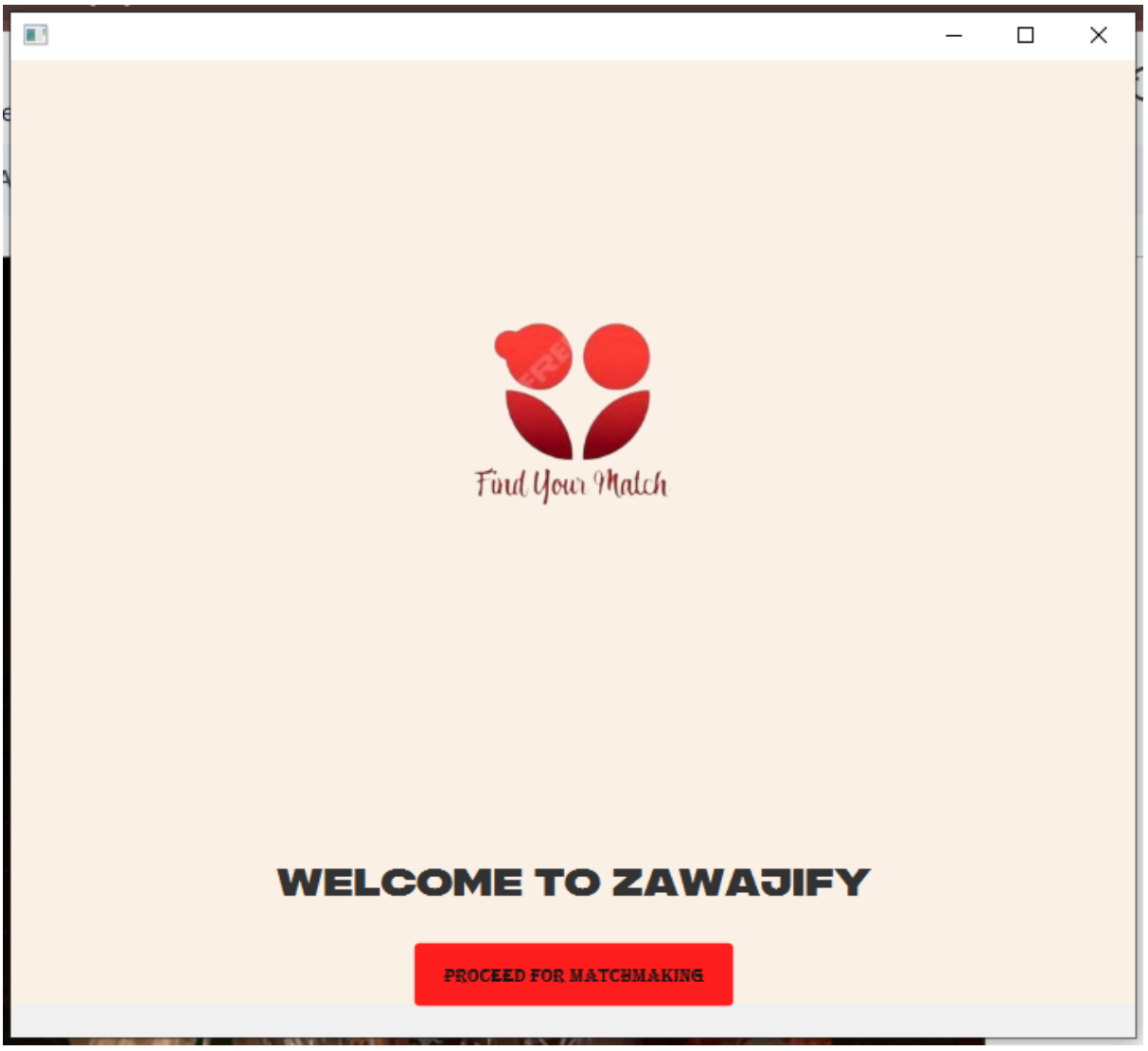
Overview:
The application works in the following sequence. It has the folliwng screens.

Screen no.1:
This screen stays for a few seconds for as long as the progress bar loads.



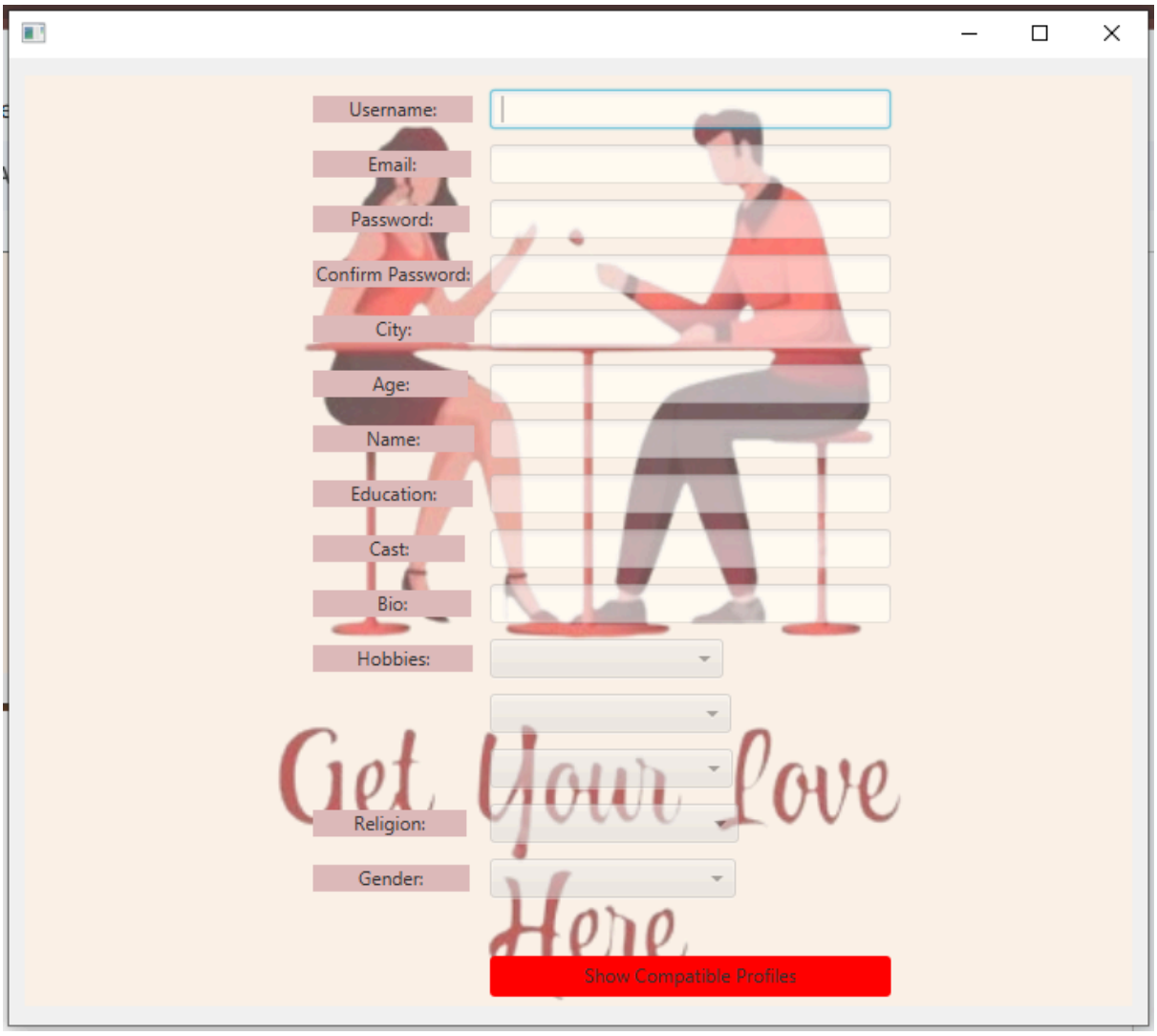
Screen no.2:

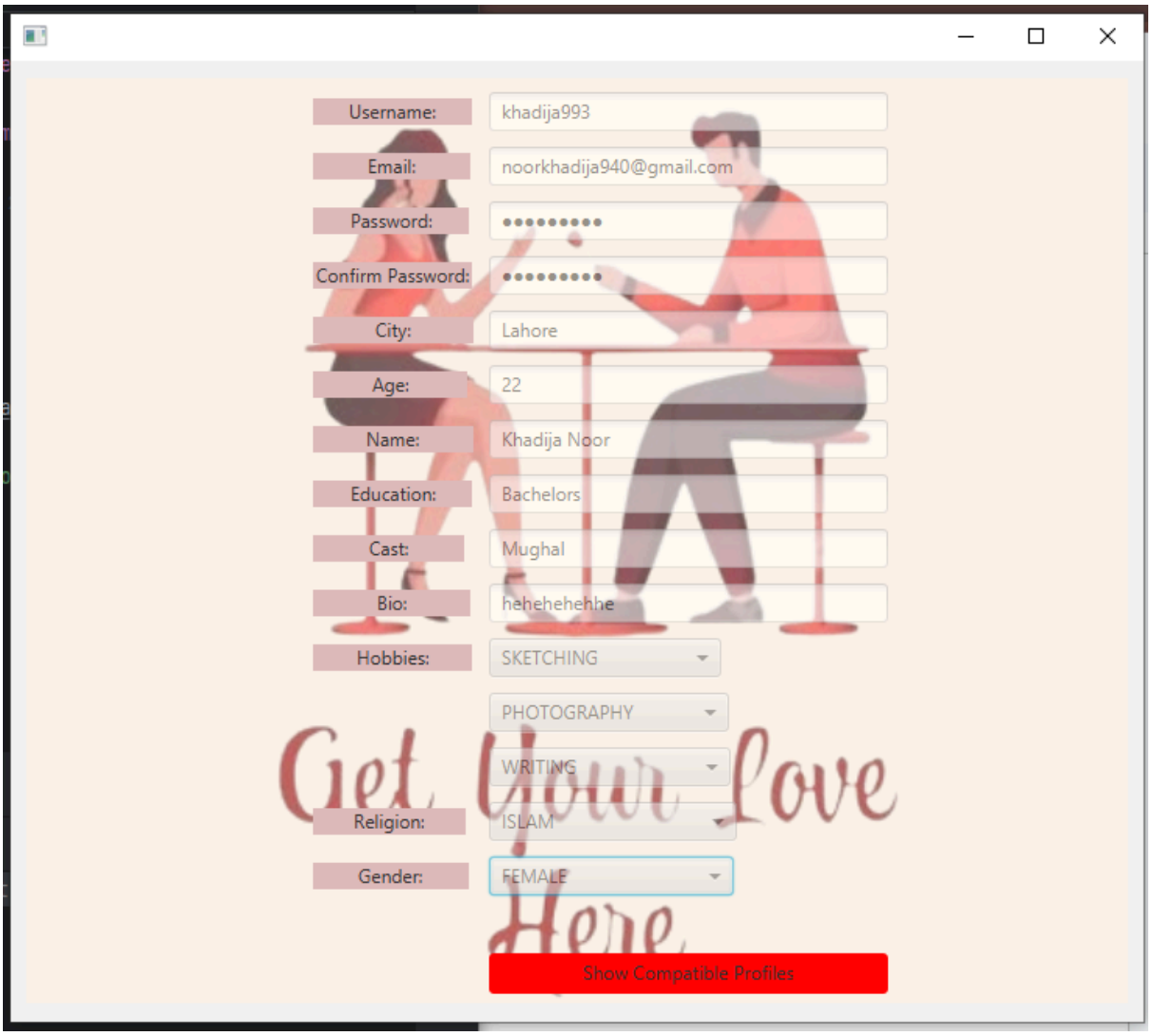
This screen has one button that takes you to the next screen when clicked upon.



Screen no.3:

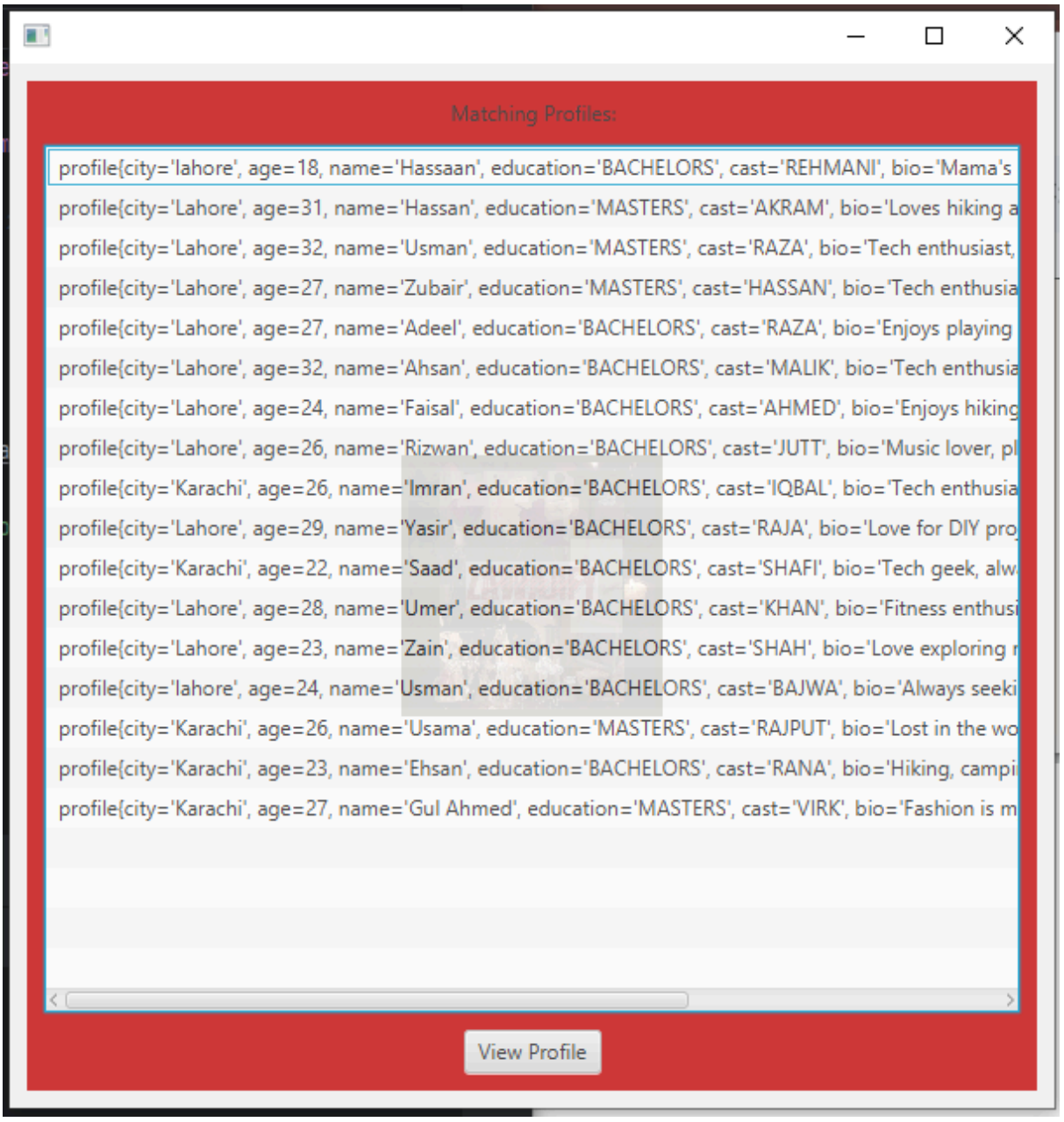
This screen shows a menu with text input boxes. It allows the user to add their data in them with certain conditions e.g. using special characters in password, etc.


Username:
Email:
Password:
Confirm Password:
City:
Age:
Name:
Education:
Cast:
Bio:
Hobbies:
Religion:
Gender:
Get Your Love Here
Show Compatible Profiles



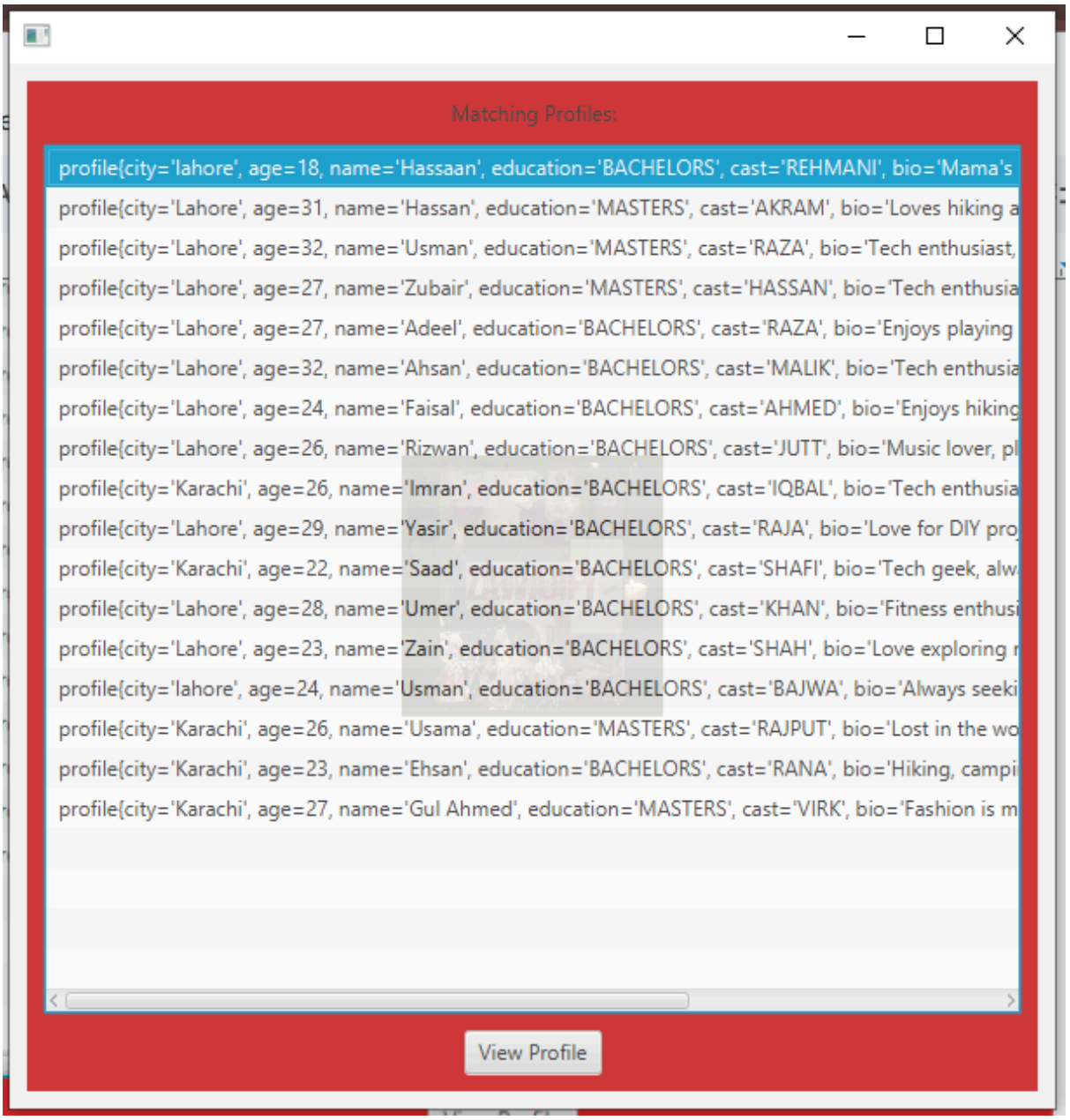
Screen no.4:

When clicked on the button ‘Show Compatible Profiles’, it shows a list of profiles that match with the details added.



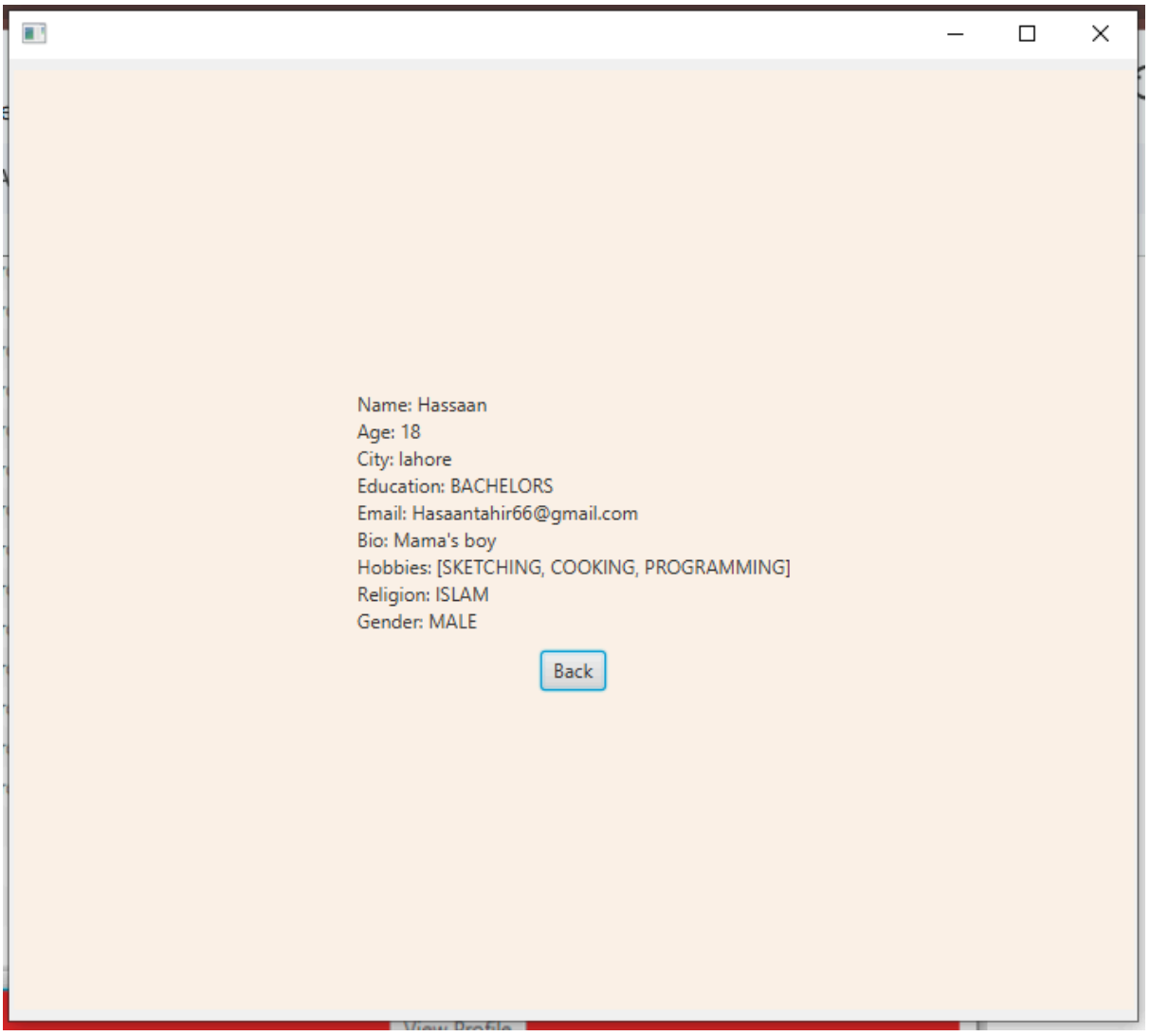
Screen no.5:

When clicked on any profile, you can view it by then clicking on the ‘View Profile’ button.



Screen no. 6:

The profile details are shown. The back button takes you back to the previous screen of the list of profiles.



Code and functioning of the Screens:

Following are the code snippets and functioning of the screens.

HelloApplication Class:

This class basically starts the program and sets the stage of the application.

```java
package com.example.connect;

import ...

public class HelloApplication extends Application {
    private Stage primaryStage;  no usages
    private Scene startupScene, intermediateScene, signupScene, matchingProfilesScene, detailedProfileScene;  no usages
    private ArrayList<Profile> profiles;  no usages

    @Override
    public void start(Stage stage) throws IOException {
        try {
            FXMLLoader loader = new FXMLLoader(HelloApplication.class.getResource( name: "startupScene.fxml"));
            Scene scene = new Scene(loader.load());
            stage.setScene(scene);
            stage.show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) { launch(args); }
}
```

IntermediateController Class:

This class creates a link between the GUI and the functioning code.
It loads the FXML files from the code and also inputs the font used in the application.

```java
package com.example.connect;

import ...

public class IntermediateController {

    @FXML
    private Label welcomeLabel; // Add this declaration to link to the FXML's fx:id

    @FXML
    void handleProceed(ActionEvent event) {
        try {
            FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "signupScene.fxml"));
            Parent root = loader.load();
            Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
            stage.setScene(new Scene(root));
            stage.show();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void initialize() {
        // Load the Riffton font
        Font rifftonFont = Font.loadFont(getClass().getResourceAsStream( name: "/com/example/connect/rifton-regular.otf"),  v: 24);
        if (rifftonFont != null) {
            welcomeLabel.setFont(rifftonFont); // Apply the font
        } else {
            System.out.println("Font could not be loaded!");
        }
    }
}
```

SignupUser Class:

Takes input from the user.

```
public class Signup_user {
    private static String city;  2 usages
    private static int age;  2 usages
    private static  String name;  2 usages
    private static String education;  2 usages
    private static String cast;  2 usages
    private static String bio;  2 usages
    private static ArrayList<Hobbies> hobby = new ArrayList<~>();  2 usages
    private static Religion religion;  2 usages
    private static Gender gender;  2 usages
    private static int userId;  2 usages
    private static String username;  2 usages
    private static String email;  2 usages
    private static String password;  2 usages
```

SignController Class:

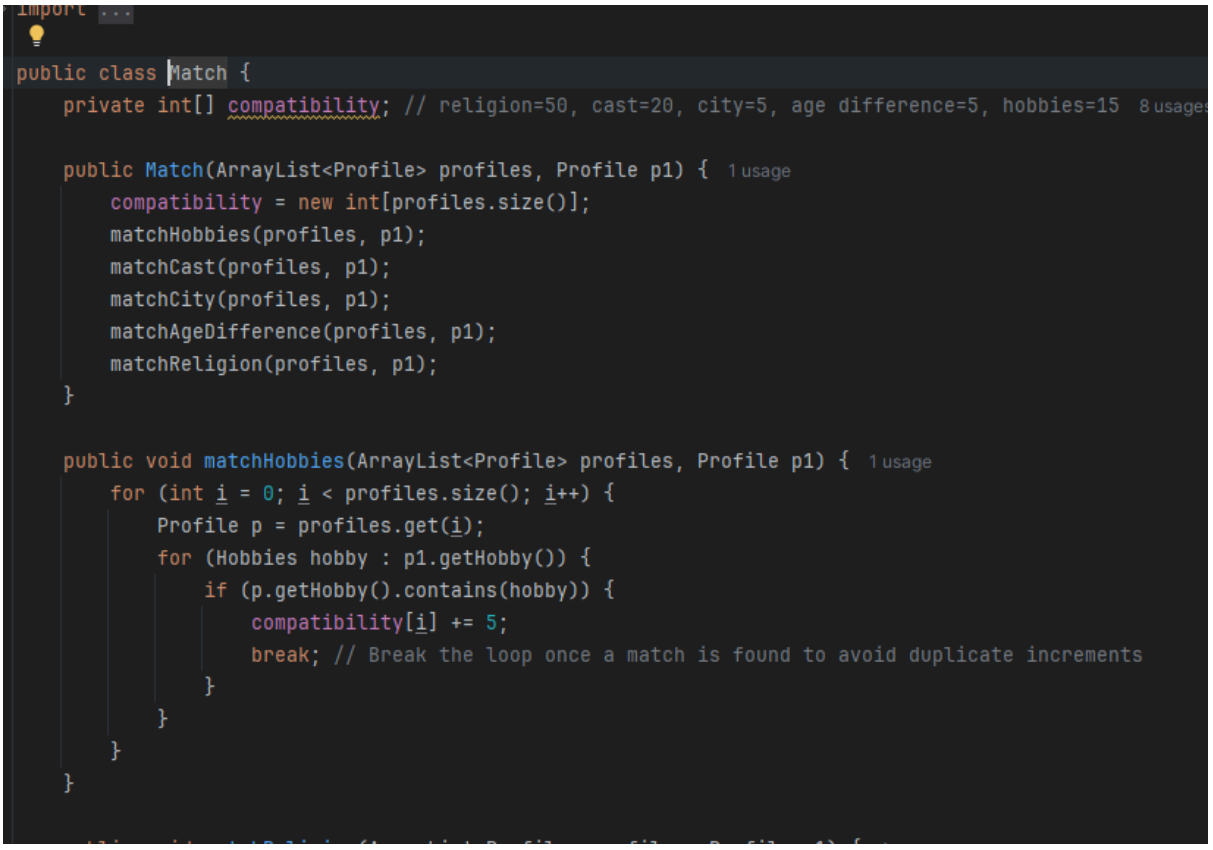
Stores data.

```
public class SignController {
    void handleSignup(ActionEvent event) {

        // Create a profile for the student
        p1 = new Profile(
                Signup_user.getUsername(), Signup_user.getEmail(), Signup_user.getPassword(),
                Signup_user.getCity(), Signup_user.getAge(), Signup_user.getName(),
                Signup_user.getEducation(), Signup_user.getCast(), Signup_user.getBio(),
                Signup_user.getHobby(), Signup_user.getReligion(), Signup_user.getGender()
        );
```

Match Class:

Checks for matches.



```
import ...

public class Match {
    private int[] compatibility; // religion=50, cast=20, city=5, age difference=5, hobbies=15  8 usages

    public Match(ArrayList<Profile> profiles, Profile p1) {  1 usage
        compatibility = new int[profiles.size()];
        matchHobbies(profiles, p1);
        matchCast(profiles, p1);
        matchCity(profiles, p1);
        matchAgeDifference(profiles, p1);
        matchReligion(profiles, p1);
    }

    public void matchHobbies(ArrayList<Profile> profiles, Profile p1) {  1 usage
        for (int i = 0; i < profiles.size(); i++) {
            Profile p = profiles.get(i);
            for (Hobbies hobby : p1.getHobby()) {
                if (p.getHobby().contains(hobby)) {
                    compatibility[i] += 5;
                    break; // Break the loop once a match is found to avoid duplicate increments
                }
            }
        }
    }
```

FXML Files:

They run the gui.