Web Development Project

Kabir S Prakash

April 28 2024

1 Introduction

This documentation provides overview of my HTML, CSS, and JavaScript code which combined form my dating website **SoulSync**. This website has a login page, forgot password page, dating page, matching page and a scrolling page. I have made my website interactive and responsive as best I could.

2 Launching the Website

Follow the steps given below to launch and navigate through the website :-

1. Unzip the submission.zip file.
2. Host the login.html page on a local server using a browser.
3. Enter valid username and password.
4. If you don't know the password click on forgot password button.
5. On forgot.html page enter username again and click on submit to see the secret question.
6. Answer the secret question in given text box and click on Submit. If your answer is correct, password will be shown which you can remember and go back to the login page by clicking on 'Go back to login' button.
7. After entering valid username and correct password on login.html click on Login button to go to the dating.html page.
8. To see all profiles click on the 'See All Profiles' button visible on front section.
9. On scroll.html page you can see all the profiles and also use filters by clicking on the filter icon.
10. To go back to the dating.html page you can click on the custom built go back icon.

11. On scrolling down the dating.html page you see a form to fill your profile information to be used for matching with a student. Fill all the sections and click on 'Find Match' button.

12. On match.html page you can see the profile of matched student.

13. On dating.html page, scroll.html page and match.html page you can find a Logout button on top right corner of the page which you can use to go back to login.html page.

3 Basic Tasks

1. **Login Page**
   On the login page I have provided a login box which takes username and password as input, validates it based on some regex I formulated and if the password is correct corresponding to the username, **Login** button redirects to the dating page. Any error during credentials validation is shown as a pop up using alert. Login page also has a **Forgot Password** button which redirects to the forgot page.

2. **Forgot Page**
   'Forgot' page has a input box which first asks for username. After clicking on **Submit** button with the help of DOM Manipulation a secret question and text input box for secret answer is displayed inside the same input box. After validating the secret answer with JS script, if the input answer is wrong then a error message is shown as a alert while if the answer is correct, password is displayed inside the same input box.

3. **Dating Page**

   - Dating page has a form for user to enter his information. **Find Match** button inside the form box first checks for empty input fields and displays error accordingly. Find Match button also validates the form based on some regex which I have formulated. If all credentials are valid then a student object is created for the user and passed into the **findMatch(student)** function call. Any error during form validation is displayed as text just below the form.
   - **Matching Algorithm**

     – I have created a students array which stores all student objects. I have also created an array of valid indices based on the profile of the user which is filtered sequentially based on interests and hobbies match with the existing students data.
     – I have used an array named **flags** which stores objects. These objects store information about every student in the database based on criteria which I have made. The data member age_flag

sets true if the student's age gap with user is ± 2. yos_flag sets true if year of study is same. interest_flag stores number of interests that match with the user. hobby_flag stores number of hobbies that match with the user.

- Firstly the valid indices is filled with indices of opposite gender which also ensures that the algorithm can never match with the same person.
- Then after checking every time the length of the valid_indices array I sequentially apply age filter, year of study filter, interests filter and hobbies filter. For filtering I have created function for each filter which take valid_indices and flags array as input.
- ageFilter() keeps all profiles with age gap ± 2. Amongst all profiles in valid_indices yosFilter() i.e year of study filter keeps all profiles with same year of study. interestFilter() keeps all the profiles in valid_indices which have maximum intersection with interests of the user. hobbyFilter() keeps all the profiles in valid_indices which have maximum intersection with hobbies of the user.
- If the final valid_indices array has more than one element then a random function gives profile of matched student while if valid_indices has only one element then that profile is stored as a student object **matched_student**.

- The **matched_student** object is converted to a string using JSON.stringify() and then encoded into a URI string. This encoded student is passed into match.html page.

4. **Match Page**
   The encoded student object is retrieved as a Query Parameter and then decoded back to object **final_student** using **decodeURIComponent()** function. The profile of this final student is displayed on match page.

5. **Scroll Page**
   See All Profiles button on dating page and match page redirect to the scroll page. Here all students profiles are fetched from students.json and displayed.

4 Customizations

4.1 General Customizations

- Firstly I have made the **UI** as beautiful as I could with use of vast CSS attributes. All the web pages have a footer section which have dummy links which are responsive on hovering of mouse over them.

- I have used many icons like instagram icon, facebook icon, language icon etc. to improve the UI. **All the buttons** on my website are interactive which change style on hovering of mouse giving the feel of a button.
- I have changed passwords of login.json template file which was provided to use my keypad feature explained below. I have also changed format of roll numbers in students.json file from **2023XXX** to **23BXXXX or 23bXXXX**. But validation of user works for both formats.
- I have added many **more** interests and hobbies and hence included 48 more profiles in students.json with these new interests and hobbies. I have provided photos for all the students in photos directory.

4.2 Navigation Bar

I have added a navigation bar on dating, match and scroll page which has the website logo, some dummy links and a Logout button.
Logout button redirects directly to the login.html page.

4.3 Keypad Authentication

I have implemented two types of security authentication on the login page. If in the login.json file passwords of all the users are numeric string then **onscreen keypad** is displayed with interactive buttons to enter the passcode. Otherwise a text box is displayed to enter the input password. Use the login.json file provided by me to explore the keypad feature :).

4.4 Filters

I have added a **filter icon** on scroll page which on clicking displays a **filter box**. Filter box has options to register for gender and year of study. To apply filters click on the Apply button.
Algorithm

- I created an valid_indices array which first stores the indices of profiles with gender selected as filter.
- valid_indices array is then filtered based on year of study selected.
- Finally profiles of students corresponding to valid_indices is displayed in the profiles container after clearing the container.

4.5 Interactive Images

Instead of boring checkboxes to register interests and hobbies I have added clickable images describing interest and hobby.

4.6 Go Back Button

On match and scroll page I have added a custom built go back icon which on clicking redirects to the previous web page.

5 Conclusion

SoulSync is the best dating website to find your special partner and make friends. Our tagline is "Where Sparks Ignite and Love Unfolds!". This is my first major project and I hope you enjoy exploring the website. :)