Looking For A Date Project

VENNELA YASASVI NAIDU

April 28, 2024

1 Introduction

This document provides an overview of the HTML,CSS,Javascript web languages used for finding a partner based on some criterias such as hobbies ,interests ,gender.We also need to include some customizations and should provide only one perfect match if possible.Otherwise we will provide respective error messages.

2 Launching Website

1. Unzip submission.zip file.
2. Open a terminal at the directory where your ”login.html” file is located.
3. Type the following in the terminal:- python3 -m http.server
4. Press Enter and Don't interrupt/close this command while you are viewing your .html file.
5. Go to a web browser and type:-”http://localhost:8000/login.html”.

3 Basic Tasks

3.1 Login Page

To access the input interface, there can be a login screen before that,which will require you to enter username and password of some sort.If you are “registered” and you fill correct entries, then you login, otherwise give appropriate error messages.We are using login.html file for this login page.

This code creates a login page for a dating website. It includes HTML elements for username and password input fields, a login button, links for forgot password and sign-up, and a section to display error messages. The styling is done using CSS, and JavaScript is used to handle the login functionality by fetching data from a JSON file.I also used background image

I used some external links for styling the HTML page at :

- Font Awesome for icons.
- Google Fonts for custom font styles.

```javascript
fetch('login.json')
    .then(response => {
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        return response.json();
    })
    .then(data => {
        // Handle the fetched data
    })
    .catch(error => {
        console.error('Error fetching login data:', error);
    });
```

This code snippet fetches data from a file named login.json, parses it as JSON, and then handles the fetched data. The then() method is used to chain promises, allowing you to perform sequential operations.

3.2 Forgot Password

We need to provide some kind of a password recovery system if a previously registered user forgets their password. A secret question-answer phrase will be there for every username-password in login.json.I used some external links for styling the HTML page.

Users are prompted to enter their username, upon which the JavaScript function forgotpassword() is invoked. This function retrieves user data from a JSON file named ”login.json”, cross-references the entered username to find the corresponding secret question, and prompts the user to answer it.

After submitting their answer, the pass() function validates it against the stored secret answer. If the answer matches, the user’s password is displayed; otherwise, an error message is shown. This process guides users through recovering their password securely, providing clear feedback at each step.

3.3 Input Interface

We need to create a decent input interface with proper labels and input boxes for a person to fill in his/her/their personal details.Those details should correspond to the fields in students.json, i.e. the person's "IITB Roll Number", "Name", "Year of Study", "Age", "Gender", "Interests", "Hobbies", "Email" and "Photo".

The HTML document provided constructs a web page titled "Matching" for a dating website. Users are prompted to input various personal details such as their roll number, name, year of study, age, gender, email, interests, and hobbies. The form includes input fields for textual data, radio buttons for gender selection, and checkboxes for selecting interests and hobbies.

Upon submission, the form data is validated using JavaScript to ensure all required fields are filled. If validation passes, the form data is stored in the browser's local storage for future reference, and the user is redirected to a page titled "match.html" to find a partner. Additionally, there are options to scroll through profiles or log out, each linked to their respective pages.

```
localStorage.setItem('gen', gender1);
localStorage.setItem('int', JSON.stringify(interests1));
localStorage.setItem('hob', JSON.stringify(hobbies1));
localStorage.setItem('rol', roll1);
localStorage.setItem('email', email1);
```

I am storing the values in localstorage and using them in match.html code to obtain a perfect match.On again giving the inputs the values stored in the local storage are updated to the newly assigned values.

3.4 Scrolling

We need to provide a feature in the input interface (dating.html) to allow the person to scroll/swipe through all the students (their details and photo) present in students.json file.

This HTML document displays profiles of students fetched from a JSON file (students.json). It dynamically creates fieldsets for each student, containing their photo, name, roll number, year of study, age, gender, interests, hobbies, and email.

Additionally, it provides a rating system for each profile, allowing users to rate students on a scale of 1 to 5 stars. Users can click on stars to select a rating, and then submit the rating. The UI updates to display the selected rating, and users can also clear their rating selection if needed.Used 3 functions for rating,submit,clear.

The styling includes a background image, font adjustments, and layout settings for a visually appealing presentation of the profiles.

3.5 Perfect Match

This JavaScript function profile(student) is designed to populate a profile card for a given student. It takes a student object as input and dynamically creates HTML elements to display various details about the student, such as their name, roll number, year of study, age, gender, interests, hobbies, and email. It first clears any existing content within the specified HTML container (.a), then creates a fieldset (fs) to hold the profile information. Inside the fieldset, it creates a division (div2) for the student's photo and another division (div3) for textual information. Within div2, it adds an image element (image) with the source set to the student's photo URL. Inside div3, it creates paragraphs (p) for each piece of information, appending them with appropriate text content obtained from the student object.
The perfect match of the user is known based on the following criterias:

1. If the gender of user is male then should match with female,if the gender of user is female then should match with male and if the gender of user is other then should match with other respectively.
2. The "right match" of a person is not the person itself. So I used that the IITB Roll Number of user should not be equal to that of students.
3. After this the user interests and hobbies are checked with students interests and hobbies.The students with maximum no.of intersections between interests and hobbies will be selected.
4. If the no.of students selected are greater than or equal to one then the first student in that is a perfect match.
5. If the no.of students selected are zero then respective error messages are given.

3.6 Output Interference

When the person clicks "Submit", a new tab will appear showing the "right match" for that person along with their details (if you succeed in finding the "right match" for that person).If didn't find any match then given respective error message.

The details given are photo with the information present in students.json.I used a background video for the output details.I also used different font styles.

4 Customisations

4.1 UI

Made the UI more interactive by adding the animations,graphics,background photos,background images,gifs,styled fonts.I extracted these images,fonts,gifs,.. from internet AI.

4.2 Rating of Profiles

While scrolling through students profiles we can rate the students from 1 to 5 and we need to submit if we wan't clear the rating we can click the clear button.If we will again change the rating and click submit the previous rating is replaced by the present rating.

1. Create Rating System: The createRatingSystem function dynamically generates a rating system for each student profile. It creates a container div (ratingContainer) with a span containing the text "Give Rating:". Then, it creates five spans (star) representing the rating stars. Each star is assigned a data-rollno attribute to identify the associated student and a data-rating attribute to represent its rating value. Event listeners are added to each star to handle clicks (handleStarClick).

2. Handle Star Click: When a star is clicked, the handleStarClick function is triggered. It retrieves the roll number and rating value of the clicked star, then updates the visual appearance of the stars. Stars with a rating value less than or equal to the clicked star's rating are marked as active (colored).

3. Submit Rating: Upon clicking the "Submit" button, the handleSubmitClick function is called. It retrieves the roll number and the selected rating value from the clicked button's attributes. If no rating is selected, an alert message is shown. Otherwise, it adds a div (ratedValue) displaying the selected rating out of 5.

4. Clear Rating: Clicking the "Clear" button triggers the handleClearClick function. It removes the active state from all stars and clears any existing rated value displayed.

Appended createRatingSystem(data[i]["IITB Roll Number"] to div3 present in scroll.html file.

4.3 Mailing the right match

This sendEmail function is designed to send an email to a specified recipient using the SMTP.js library. It first retrieves the user's email address from the local storage. Then, it constructs an email message object with the necessary parameters such as the host (in this case, Gmail's SMTP server), the sender's email address, the recipient's email address, the email subject, and the email body.

After constructing the email message, it attempts to send the email using the Email.send() function. If the email is sent successfully, it displays an alert confirming the successful delivery. However, if an error occurs during the sending process, it displays an alert indicating the failure and logs the error message to the console for debugging purposes. It's important to note that for the Password field, users need to enter their Gmail password for authentication, which should be handled securely.