Looking for a date ?

Uday Darade

28 April 2024

1 Project Documentation

1.1 Introduction

This documentation provides an overview of a dating website that allows users to login using their userid and password, retrieve their password if forgotten , fill their interests and hobbies along with other details to find a perfect match with an option to contact them and scroll or swipe through the profiles of other users on the website..

2 Website Navigation and Functionality

After unzipping the folder please open the file index.html with live server on your local device. The website begins with the homepage, `index.html`, designed to confirm that the visitor is a human. This is achieved through a CAPTHA code that the visitor must enter. Once verified, the user is directed to `login.html`, the login page, where they are required to enter their username and password. If a user forgets their password, they can retrieve it by answering a secret question provided on the same page.

After successfully logging in, the user reaches `dating.html`, a form that requests the user to provide personal details. This page offers four main functionalities:

1. **Find a Match:** The user must complete all form fields and submit the information. This action redirects the user to `match.html`, which displays the most compatible match based on the user's input.

2. **Scroll or Swipe through Profiles:** By clicking this option, the user is taken to `Scroll_or_swipe.html`. Here, users can browse through other profiles in a swipeable format, similar to popular dating apps.

3. **Find Percentage Match with All Profiles:** This option leads to `percentageall.html`. On this page, the user enters their details, and upon submission, is redirected to `matchall.html`. This final page visually presents the compatibility percentage between the user and all other profiles on the website.

4. **Logout:** This option will logout the user out of the website and land on `login.html`

All of the pages, `scroll_or_swipe.html`, `match.html`, `matchall.html`, and `percentageall.html` feature consistent navigation options at the top of the page, including a "Logout" button to exit the session and a "Back" button to return to the previous page. These options enhance user navigation and improve the overall user experience on the website.

3 Basic Tasks

3.1 Login Page

This page has option for the user to input his username and password which are verified with the data from login.json using javascript. If the data is wrong, user is asked to try again. The option Forgot password redirects on new page where he can get back his password.

The styling of this page has various options like dynamic background, dynamic heading effects , fade in animations and hover effects.

- **Changing Background:** The background of the page changes every 20 seconds, making it look cool and lively. This is done using special CSS codes that switch between different background images over time. The animation is defined in the CSS file with the @keyframes backgroundchange
- **Dynamic Heading Effects:** The big title "The Moon is beautiful, isn't it?" is eye-catching! It has a colorful rainbow look and a gentle shaking motion to make it more interesting. These effects are achieved using CSS. The rainbow effect is created using a linear gradient background, and the shake animation is applied to the text.
- **Interactive Button Animations:** When you hover your mouse over the buttons, they enlarge, creating a cool effect that invites you to click them. This is achieved with CSS hover effects that modify the buttons' appearance when hovered over.
- **Click Sound Effects:** Clicking on input boxes to type your username or password triggers a pleasant click sound. This adds an interactive and fun element to the project. JavaScript is used to add this sound effect by playing an audio file when the input boxes are clicked.
- **Entrance Animation:** When the page loads, everything fades in slowly, creating a stylish entrance that captures your attention. This effect is implemented using CSS animation to gradually increase the opacity of the elements, making the project look smoother and more stylish.

3.2 Forgot Password Page

This page has similar styling as the login page . When user inputs his username and clicks on submit, his name is checked in the login.json file and the corresponding question is shown. If the user answers the questions correctly, the password is shown on the screen and there is option to go back to the login page after that.

3.3 Dating form Page

The HTML code collects various details from the user, including their IITB Roll Number, name, year of study, age, email, gender, interests, and hobbies. Upon form submission, JavaScript functions are triggered to process the data and find the best match based on the provided criteria.
The styling of the page is similar to the login page with animations and sound effects. Some extra buttons added are :

- **Scroll through Profiles**: Redirects users to scroll or swipe through profiles.
- **Logout**: Logs the user out and redirects to the login page.
- **Find Percentage Match with All Profiles**: Finds the percentage match with all available profiles.

3.4 Match Found page

The Match Found page is designed to display information about a potential match and provide options to interact with the match.

The background of the page displays a profile background image, creating an attractive visual appeal. The page layout and styling are defined using internal CSS, including font styles, button colors, and animation effects. The profile container features a subtle shaking animation, giving it a dynamic and engaging appearance. The animation is implemented using CSS keyframes, creating a continuous shaking effect on the profile container. There is a percentage bar showing the percentage match, logic of which is implemented using javascript functions.

Three interactive buttons are provided: "Want to contact ??", "Ask for a date ??", and "Try again ??", each triggering different actions when clicked. JavaScript functions are assigned to these buttons to handle user interactions, such as sending emails or navigating to other pages.

3.5 Swipe through profiles page

This page serves as the central hub for users to browse through potential matches, facilitating the discovery of compatible individuals within the platform.

The CSS styles applied to the page elements enhance its aesthetic appeal and usability. Various properties such as font styles, background colors, and box shadows are meticulously crafted to create a cohesive and visually engaging layout. It has buttons to go to next profile or the previous profile and the buttons have sound effects on clicking. The page includes interactive elements such as navigation buttons and logout options to enhance user experience and functionality.

JavaScript functionality is seamlessly integrated into the page to handle user interactions and dynamic content loading.

3.6 Finding the Right match

The formula is designed to optimize user compatibility based on several key factors, ensuring high user satisfaction and meaningful connections. This is the code relevant to that part.

```
for (const interest of user.Interests) {
    if (profile.Interests.includes(interest)) {
        interestscore++;
    }
}
for (const hobby of user.Hobbies) {
    if (profile.Hobbies.includes(hobby)) {
        hobbiescore++;
    }
}
const userinterests = user.Interests.length;
const userhobbies = user.Hobbies.length;
const bestmatchinterest = profile.Interests.length;
const bestmatchhobbies = profile.Hobbies.length;

interestscore = Math.max(0.5, 2 * interestscore);
hobbiescore = Math.max(0.5, 2 * hobbiescore);

const studentYearOfStudy = parseInt(profile['Year of Study'].slice
    (0, -1), 10) || 0;
const yearDiff = Math.abs(user['year-of-study'] -
    studentYearOfStudy);
const ageDiff = Math.abs(user['age'] - profile['Age']);
const percentagematch = Math.max(10, (100 - (yearDiff) - (ageDiff /
     2) - 2 * (userinterests - (interestscore / 2)) - 4 * (
    bestmatchinterest - (interestscore / 2)) - 3 * (
    bestmatchhobbies - (hobbiescore / 2)) - (userhobbies - (
    hobbiescore / 2))));
```

The formula has been developed with the following considerations:

1. **Interests vs. Hobbies:** Having more interests in common is prioritized over hobbies because sharing similar interests is crucial for compatibility. Interests are weighted more heavily than hobbies in the compatibility calculation.

2. **Year of Study and Age:** Being in the same year of study typically indicates a higher probability of shared experiences and perspectives, which

fosters a stronger connection. Although age is considered to a lesser extent, it is also factored into the compatibility to enhance match relevance.

The matching algorithm is designed to pair users based on their gender preferences:

- Females are matched with males, and vice versa, based on the data provided.
- Users who identify as 'Other' can be matched with any user, regardless of gender, enhancing inclusivity and providing equal opportunities for meaningful relationships.

To enhance user experience and satisfaction, the system ensures that every user is matched with someone as long as data is available. This approach not only increases engagement but also ensures that everyone gets a fair chance at finding a connection.
This is my personal opinion and i think this is the best way to find the right match.

4 Customisations

4.1 Animations

- **Changing Background:** The background of the page changes every 20 seconds, making it look cool and lively. This is done using special CSS codes that switch between different background images over time. The animation is defined in the CSS file with the @keyframes backgroundchange
- **Dynamic Heading Effects:** The big title "The Moon is beautiful, isn't it?" is eye-catching! It has a colorful rainbow look and a gentle shaking motion to make it more interesting. These effects are achieved using CSS. The rainbow effect is created using a linear gradient background, and the shake animation is applied to the text.
- **Interactive Button Animations:** When you hover your mouse over the buttons, they enlarge, creating a cool effect that invites you to click them. This is achieved with CSS hover effects that modify the buttons' appearance when hovered over.
- **Entrance Animation:** When the page loads, everything fades in slowly, creating a stylish entrance that captures your attention. This effect is implemented using CSS animation to gradually increase the opacity of the elements, making the project look smoother and more stylish.
- **Visually Appealing percentage meter:** This feature is used on matchall and bestmatch page. It is a circle which fills up according to the percentage provided.

- **Background Images:** The background images are designed by Microsoft Designer specially for this website to make it look visually appealing. On login, dating and few pages, there are images of a boy and girl standing on two sides of the main content and they look like a lovely couple , demonstrating that the user might find such partner through this website. For the match page, a special image was created with the couple and a white board between them.

- **Container Styling:** The '.container' class styles the central content container of the page. It sets a margin from the top to create space, and padding to provide internal spacing within the container. The 'background-color' property is set to 'rgba(255, 255, 255, 0)' to make the background transparent.

- **Navigation Buttons:** The navigation buttons of logout and back are two vibrant images, which on clicked perform javascript functions.

4.2 Sound Effects

Sound Effects are added to enhance user experience. When user clicks on any input box a click sound is played. In the swipe profile page, if user clicks on the next or back button, corresponding back and next sounds are played. Also, on loading the match profile, a nice sound is played.

This can be done by two ways , and i have used both the ways

- **Using HTML tag:** Audio elements are embedded within the HTML markup and triggered using JavaScript event listeners

- **Declaring in Javascript:** Audio elements are declared and assigned the corresponding sound files using the new Audio('filename.mp3') syntax.

4.3 Percentage Display

The percentage display is a visual representation of how well two profiles match each other. It's a dynamic element that provides users with a quick understanding of the compatibility between profiles.

- **Appearance**: It is represented by a circular progress ring. The ring starts empty and gradually fills up as the percentage match increases.

- **Primary Functionality**: The percentage match is calculated based on various factors such as interests, hobbies, age, and other profile information.

- **Implementation**: The percentage match display is implemented using SVG (Scalable Vector Graphics) in HTML. The circle's stroke-dasharray and stroke-dashoffset properties are manipulated using JavaScript to create the filling effect based on the percentage match.

- **Interactive Feedback**: As the percentage match is updated, the progress ring visually reflects these changes, providing users with immediate feedback on the compatibility level.

This function is used mainly in two pages, match.html to show percentage match with the best match profile and matchall.html to show percentage match with every profile. This function enhances the user experience and is really interactive customization.

4.4 Emailing the right match:

After the right match is found, there are two options to email them.

- **Contact them :** This sends a casual email to the match, politely asking them further modes of communication and that i am interested in you.
- **Ask for a date :** This sends a email asking the match for a if they are interested in a casual date, and asking for the date and time which they will be free to schedule the date.

4.5 Find percentage match with every profile:

This is done using the percentageall.html and matchall.html pages. The first page is just like dating.html and takes input values from user. The formula that I have used here based on experimentation on various matchings is :

```
for (const interest of user.Interests) {
    if (student.Interests.includes(interest)) {
        interestscore++;
    }
}

// Calculate the score for hobbies
for (const hobby of user.Hobbies) {
    if (student.Hobbies.includes(hobby)) {
        hobbiescore++;
    }
}

userinterests = user.Interests.length
userhobbies = user.Hobbies.length

interestscore = Math.max(0.5, 2 * interestscore);
hobbiescore = Math.max(0.5, 2 * hobbiescore);

const studentYearOfStudy = parseInt(student['Year of Study'].slice
    (0, -1), 10) || 0;
score = interestscore * interestscore * hobbiescore;
const yearDiff = Math.abs(user['year-of-study'] -
    studentYearOfStudy);
const ageDiff = Math.abs(user['age'] - student['Age']);
const adjustment = Math.max(0.5, 4 - (yearDiff) - (ageDiff / 4));
score = score * adjustment;
```

Basically I calculated the number of interests and hobbies which are not common and then giving priority to the interests of match, i have made this function which calculates the percentage match.

5 Conclusion

In wrapping up, creating this dating website has been quite the journey! We've covered a lot, from making sure the site looks good and works smoothly, to helping users find their perfect match. We've made sure the site is easy to use and looks great. Those cool animations and sound effects? They're there to make the experience more fun and engaging. The matching system is all about connecting people who share similar interests and hobbies. I want users to find someone they really click with. Overall, this website is all about helping people connect and find love in the digital age.I've put a lot of thought and efforts into making it user-friendly and enjoyable to use. After all, love is always in style!