

CS108 Project
Report

HTML CSS Javascript Webdev

Bachelor of Technology
in
Computer Science and Engineering

Submitted by
23B1068, Vaibhav Singh



Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
Powai, Maharashtra, India – 400 076
Spring Semester 2024

Contents

1	Problem Definition	1
2	Basic Tasks	1
2.1	Login Page	1
2.2	“Forgot Password?” Button	1
2.3	Input Interface	1
2.4	Swiping	2
2.5	Finding the “right” match	2
2.6	Output Interface	3

1 Problem Definition

The goal here is to create a basic dating website. We're using HTML, CSS, and JavaScript to build it. Simply put, we want to make a place where you can enter some info about yourself and then find someone who might be a good match for you among a bunch of students. It's all about making a website that's easy to use and helps people connect with each other.

2 Basic Tasks

Following are all the basic tasks that needed to be done.

2.1 Login Page

The project includes a login page feature, which acts as a gateway to access the input interface. Users are prompted to enter a username and password combination. If the entered credentials match those of a registered user stored in the "login.json" file, access is granted. Otherwise, appropriate error messages are displayed. This ensures that only registered users can access the website, even at a later time. The login page is implemented in a ".html" file named "login.html".

2.2 "Forgot Password?" Button

Additionally, the project incorporates a "Forgot Password?" feature to cater to instances where registered users forget their passwords. Upon clicking the designated "Forgot Password?" button on the login page, users are directed to a new page named "forgot.html" or, if integrated, remain on the login page itself. Here, users input their username, triggering the display of the corresponding secret question associated with the username from the "login.json" file. If the user correctly answers the secret question, the password is revealed on the screen. Otherwise, appropriate error messages are displayed, ensuring a streamlined password recovery process for registered users.

2.3 Input Interface

The task involves crafting an input interface using HTML and CSS for users to fill in their personal details. These details correspond to the fields in the "students.json" file, including "IITB Roll Number", "Name", "Year of Study", "Age", "Gender", "Interests", "Hobbies", "Email", and "Photo".

As customisation, I have also included 'Interested in' question to know the orientation of the user and display results accordingly.

For the "Gender" field, radio buttons are implemented to allow users to choose a single option. For "Interests" and "Hobbies", checkboxes are provided for users to select multiple options for each category.

Additionally, a "Submit" button is included, which, upon clicking, initiates the search for the person's ideal match. A "Logout" button is also provided to return users to the login screen for added convenience.

All of this functionality is designed and implemented within the "dating.html" file for the HTML structure and "style.css" file for the visual styling.

2.4 Swiping

A feature is added to the input interface to enable users to browse through all the students' profiles and photos stored in the "students.json" file. This page can be accessed through the navbar.

Upon clicking the button, users are directed to a new page named "scroll_or_swipe.html". This page allows users to freely scroll or swipe through profiles, viewing details and photos of each student in the dataset.

This page filters out the profiles according to the 'interested in' input provided by the user on the dating.html.

2.5 Finding the "right" match

The task involves improving the "Submit" button on the input interface ("dating.html") to find a suitable match for the user among the profiles in "students.json" using JavaScript. The match is based on shared interests and custom criteria. If no suitable match is found, an appropriate message is displayed. If there are multiple matches, one is chosen arbitrarily. The matchmaking process is flexible, adapting to individual preferences.

I used Jaccard's matching algorithm to generate the 'right' or we can say best match. This algorithm basically gives a number which is basically number of common interests/hobbies divided by total number of hobbies/interests. In other words size of intersection of their hobbies/interests divided by size of union of their hobbies/interests. greater the number of things common between two user... greater the score. To scale up the score, I have returned $\sqrt{\text{originalscore}}$ and then multiplied it by 100 to convert it in percentage.

2.6 Output Interface

After clicking "Submit," a new tab will display the "right match" for the person, including their details if a match is found. The interface for this is structured in a separate ".html" file named "match.html", utilizing the same CSS ("style.css") and JavaScript ("script.js") files. On clicking submit in dating.html, first foundMatch.html is displayed. On hovering over the card in foundMatch.html, user can see the match. In case there is no match... The script leads the user to another webpage where an appropriate message is displayed.