

## Dating Website Project – Advanced Web Programming

Description of my whole project:

Based on the given instructions, I have tried to implement a social media platform where users can create an account, upload a profile picture, write a bio, and like/unlike other users.

Before going into the details of the project, I would like to mention that I have used the following technologies:

1. React for the frontend
2. Express for the backend
3. MongoDB for the database
4. Passport.js for authentication
5. Google OAuth for authentication
6. Multer for file uploads
7. React-Slick for the slider package
8. Materialize CSS for the frontend design (partly)
9. JWT for token generation
10. Bcrypt for password hashing
11. Mui to import some icons (home and google) ...

### Part 1: Backend

In the backend, I have created the following routes:

#### 1.1 Get routes in gets.js:

- /api/user: To get the user details
- /api/all-users: To get all users from the database
- /api/user/like: To get the users whom the current user has liked

- /api/chat: To get the chat messages between two users
- /api/user/image: To get the user's image (if any)
- /api/auth/google: To initiate the Google OAuth process
- /api/auth/google/callback: To handle the Google OAuth callback

## 1.2 Post routes in posts.js:

- /api/user/login: To login a user
- /api/user/bio: To update the user's bio
- /api/user/like: To like/unlike a user
- /api/user/image: To add an image to the user
- /api/chat: To send a chat message

The most time-consuming part from these was making the `updateLikedUsers` function which I directly supply to the `/api/user/like` route. More details about this function can be found in `posts.js`.

## Part 2: Frontend

In the frontend, I have created the following components:

- `App.js`: The main component that holds the routing logic
- `Header.js`: The header component that contains the navigation links
- `Login.js`: The login component
- `Register.js`: The register component
- `ResetPassword.js`: The reset password component
- `Chat.js`: The chat component
- `ChatWindow.js`: where the chat messages are displayed
- `Profile.js`: The profile component
- `AddImage.js`: The add image component

- GoogleAuth.js: The Google OAuth component
- Suggestions.js: The suggestions component
- UsersSlider.js: The users slider component

In the front page you will get 3 buttons: Register, Login, and Google Authenticate. On the top, which stays there all the time, you will see the navigation links: Home, language options and the user's profile picture (if logged in).

Let me start with the [header logic](#):

- You can see everything on the header component. The header component is always visible on the top of the page.
- But, when profile is clicked, there are 4 options: Profile, Add Image, Chat, and Logout.
- Of course, the first three address a logged in user, so even though you can click them unlogged, you will be redirected to the login page.

[Login and Register](#):

- These have basic implementations. The login page has a form with email and password fields.
- The register page has a form with email, password, and name fields.
- Successful login redirects to suggestions page, which contains the sliders of all users.
- Successful register redirects to the login page.

[Dropdown menus on the header](#):

#### 1. My Profile:

- When clicked, the user is redirected to the profile page.
- You can see the user's name, email, bio, and profile picture & date of registration.
- There is an option to update the bio.

## 2. Add Image:

- When clicked, the user is redirected to the add image page.

## 3. Chat:

- When clicked, the user is redirected to the chat page.
- The chat page has a list of users with whom the current user has chatted. And empty chat window.
- the chat window is updated when a user is clicked.
- In the chat window you can see the name of the user you are chatting with, and the messages.
- Additionally, you can search for a text in the messages.

## 4. Logout:

- When clicked, the user is logged out and redirected to the login page.

## Google Authentication:

- When clicked, the user is redirected to the Google OAuth page.
- After successful authentication, the user is redirected to the suggestions page.

## Suggestions:

- The suggestions page contains a slider of all users.
- You can like/unlike a user by clicking the heart icon.
- By clicking expand, you can see the whole bio or name if it was too long and was cut off.

## Additional:

I wanted to add all the node modules and my .env to .gitignore, but I feared it might not work for the evaluator. If you don't get my google client id and secret, I don't think the google authentication would work.

The most time-consuming part of the frontend was the chat window. I had to make sure that the chat window is updated when a user is clicked.

I enjoyed working on this project, but **I am aware that there are many things that can be improved.**

Until 10 days ago, I was following weekly with the course, until I emailed the TA and understood the deadlines.

Time was a constraint, so I had to skip some interesting features I was planning to implement.

All in all, I hope it will work seamlessly on your machine. If you have any questions, please let me know.

Thank you for your time and consideration.

author: "Azarias Galama"