WEB Programming report

Username: tim | Password: colin

Username: allan | Password: pau

Username: armand | Password: sylvain

Username: ugo | Password: tagnati

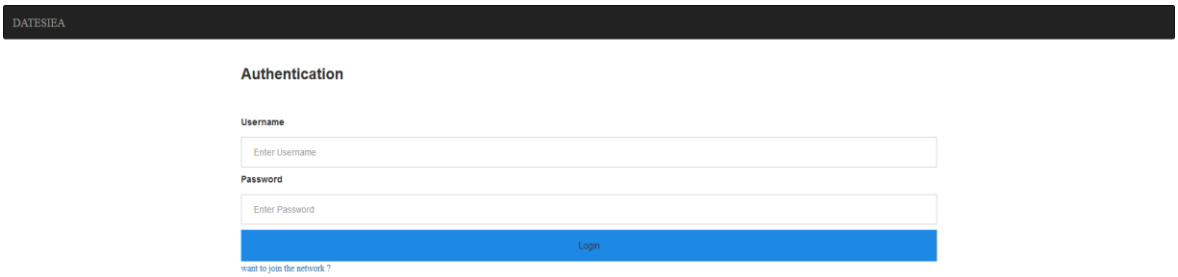
My project is a meeting/dating web application which gives people the possibility to get information on one another.

The thing I want to emphasis is the fact that I choose not to give the possibility to anyone to get in touch through the internet, all I want is to give them the possibility for an IRL discussion.

The website is constituted of three html pages: the first one for the login, the second one for the main gallery (where each profile is displayed) and the third on is for the user to have access to his personal information.

I decided not to use any JavaScript files because even though the website worked perfectly there where some errors due to the use of Vue.js. (simply used <script></<script>)

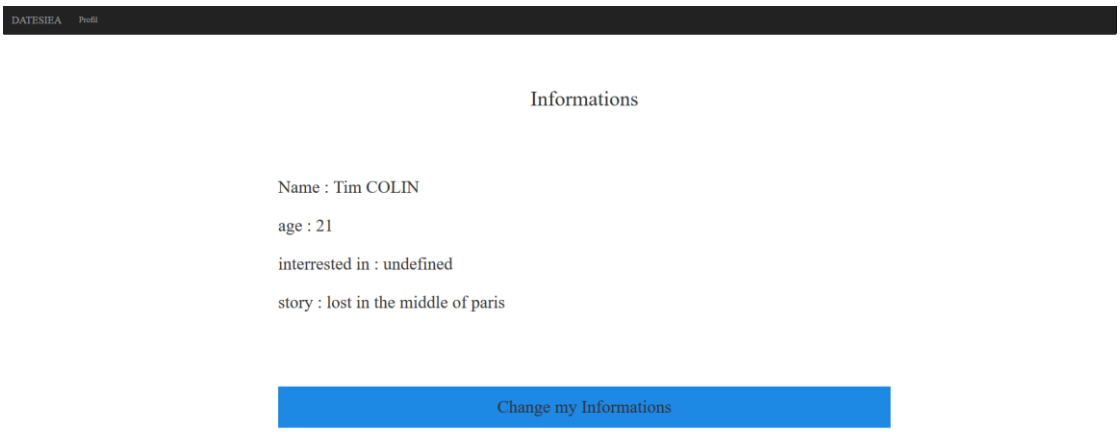
I used node.js and express to deal with all the server requests. The only requests I implemented are POST and GET requests. Most of my request are used to contact the JSON file registering all the users' information.



Conception:

I started to “draw” my website, putting as much ideas as I could so that in the first days I could figure out what was possible and what could not be done in only two weeks.

That is the reason why the first few days I choose to only read documentation, picking up the numerous elements I would need later during the realisation of my website.



Realisation:

At first, I thought I'd have the possibility to use only an html page for the entire website, using Vue.js to navigate through the layers. But I realised fast enough that it wasn't as easy as it sounded so I decided to work on several html pages rather than one.

I started by creating the structure of the website, working on forms, navigation bars and more. When I thought the structure was as I wanted it to be, I added some CSS to it to make it more enjoyable to watch. I took the liberty of using some bootstrap CSS libraries for that process.

After that I gave all my attention to the back-end, focusing at first on all the GET requests used to inject in my html pages all the information I stocked into my JSON file, then later I work on the POST requests.

The GET request are used for the gallery, it displays all the useful information of the different people: name, age, interest and story!

But those requests are also used at the time of the authentication. It is a simple JavaScript script that make a POST request, it later uses the respond to compare the values given in the form and the ones contained in the users' JSON file.

The role of the POST request for my case is only to give the client the ability to be recognize by the website even when moving through the different layers.

I created a simple text file with the username of the connected client to stock this information in memory.

After that the website was pretty much working. I gave it a little more suitable interface by rearranging some of the first CSS used and it was it!

Problems:

So, at the start I did an html page for every layer needed in the website, then I tried to concatenate every on of them onto one page using Vue.js' transitions.

It all went smoothly until I tried to use my login: because when I did nothing happened, the form was not sent and so the JavaScript script could not even run.

So, I decided to leave the login apart, so that it would not be mixed with Vue.js.

A different problem I encountered was the use of the POST request for a JSON, I tried to make an adhesion form, but even though the form's information was received and used I could not find a way to integrate it to the JSON file without errors or worst without erasing the content of the file itself.

The lack of time prevented me from correcting it so for the time being the from will be sent but no response is be expected from it.

Glitch worked perfectly, it is easy to use and clean. The deployment was problem less.

That is all the information I can give you about how I conducted my project, how I conceived it, how I realised it and why it ended-up being what it is.