Jérémy BARON
Carla RENAUX
4A-I

Project report
Web Programming

2018-2019

Jérémy BARON
Carla RENAUX
4A-I

Table des matières

Jérémy BARON
Carla RENAUX
4A-I

I. Description

This year's project consists in building a web application with different technologies. For the front-end we were told to use HTML5, CSS3, Javascript with vue.js and for the back-end, node.js with the framework express. The project has to be CRUD: Create, Read, Update, Delete.

To respond to the project requirements, we decided to do a Support Application were the user will be able to tell the support his problem. He will also be able to see and to delete his message.

The user first signs up if he hasn't an account yet. Then he logs in with his username and password. There, a contact form will ask him some information and what kind of problems he has. Once the request sent, he will see in the "Request" page his problem.

II. Project

a. Conception

The conception started by analyzing the problem and to find a web application that correspond to the need. After doing a Brain storming, we found two different ideas but decided to go with our support application because of its originality. Once our idea concretized, the idea was to decide what pages and what design we were going to make. Our project would be composed of 5 different pages: Home page, sign up, login in, contact and requests.

Our team is composed of two members: Jérémy Baron and Carla Renaux. We had to divide the work in two: front-end and back-end.

b. Making

Front-end

For the design of our web application, we used bootstrap library to build a responsive web app. By doing that we have a full responsive html interface.

To only have one html page, we used the vue.js framework. The idea that when the user clicks on contact for example, only the content of the page changes and not the entire page. To do that, we used "v-on:click" and "v-if" so that when the user clicks on "Contact" a variable n, initialized at 0, because 3 and the code corresponding executes and shows the contact form. That is how we proceeded for each of buttons. Thanks to that, we only have one html file.

The server

For the server, we use node.js with express.js to fit the requirements. We followed a lot of tutorials for this part as we think it was the hardest one.

Sign Up

When the user creates an account, the username and the password go directly into a json file. Our functions in the server.js does all the work. Whenever people sing up, their datas are stocked in a .json file.

Sign In

Once we have the information of the user, to log in, we compare the information written in the log in form with the ones in the json file using a “for” loop to run through the file.

The contact form

Based on the same principle as the sign up, we capture the information in the input and store it in a json file.

Request

This is the place the user is supposed to see his requests and delete it. Unfortunately, we didn’t manage to just display the data stocked in our json file, even after working on it for several days.

c. Deployment

To deploy our web application, we are using glitch. At the start we tried to use google cloud platform, but we had a lot of issues with it. We switched to glitch because its convenient and very fast by logging with github and just import our project from there.

III. Difficulties

Through this project we met a lot of problems. First of all, we did not have much time to learn the new frameworks and language. We had to do a lot a research to understand everything, so we lost a lot of time.

The main problem was to capture the information of the user and put it in the json file. We tried different ways doing it but most of them were not going through the server, so they were going against the rules. We finally decided to do small functions in the server.js. By doing that we managed to store the datas on .json files. Then we had another problem: how to display the content of our files without using javascript or jquery. This was the main thing we tried to do for the last days remaining. We even tried to display it with basic javascript even if it was forbidden but it didn’t work because of view.js ruling the different pages of our apps.

We think it was very difficult to assimilate express/node.js and vue.js. We tried our best to be as fast as possible. We had issues with vue.js but we managed to fix them.

IV. Conclusion

In conclusion, we didn't manage to cover all the instructions. We have a deployed app on internet. The user can create data, we can see those data manually in the json files, but we didn't manage to display it on our html page. Because of that, the user can no delete his messages. We also tried to send the messages directly by mail after validating the contact form by using nodemailer but we didn't have time to make it work either. Even if we didn't manage to fulfill all the requirements, we did learn a lot of things during this project and facing difficulties is the best way to learn. Maybe with few more days we could have finish our app.