

ADVANCED WEB PROGRAMMING

Project Report : PostQuote



CHEN Kevin
LIONET Myriam

https://github.com/Hevilio/web_prog_4AI
<https://glitch.com/edit/#!/fanatic?path=server.js:1:0>
<https://fanatic.glitch.me/>

TABLE

I) REMINDER.....	2
A) Context of the project.....	2
B) What is our project ?.....	2
C) Product Backlog.....	2
II) BUILDING OUR FIRST SPA FROM SCRATCH.....	2
III) HOW DOES IT REPLY TO THE REQUIREMENTS ?.....	3
IV) DIFFICULTIES ENCOUNTERED.....	4

I) REMINDER

A) Context of the project

We had to use HTML 5, CSS 3 (or bootstrap), Vue.js and Node.js in the aim of doing a website with a few functionalities such as login or CRUD. The website could be whatever we want.

B) What is our project ?

Our project is a web site named PostQuote on which people can submit their own quotes once they are logged in (after being registered). We can also read quotes other people submitted.

C) Product Backlog

Our web-site should include :

- a. Register system
- b. Login/logout system
- c. Display all quotes
- d. Add new quotes
- e. Delete quotes
- f. Update quotes

II) BUILDING OUR FIRST SPA FROM SCRATCH

At first, our main idea was to create a website where people could post their own novels and fictions. We wanted to allow users to interact a lot with novels of other people (comments, likes, searching by categories and trending novels/authors...), and to add some account features like a profile picture and a biography of each user... That's what we tried to achieve firstly.

We started to do all the aesthetic aspect of the project so we focused on how to display information on the website with Vue.js, HTML and CSS. We all also wanted to try Bootstrap, however we encounter some difficulties for the implementation with Vue.js with multiple html pages. Indeed, we had underestimated the amount of work. It was a huge project to realise with our new and weak web-programming knowledges. After more than a week of learning and trying to make this website work the way we conceived it, Kevin and I realised it was time to review our expectations.

“Is it better to make a website with multiple functionalities but also lot of bugs whether a simple and well-done application ?”

The answer seemed obvious at that point. We had to privilege quality over quantity. We dropped our old website and decided to remake all from scratch. Maybe it would be less impressive, but at least we would understand every single row of code we write, and we wouldn't cut and paste code segments from internet examples we barely understand.

This is why we preferred to simplify our website to a single page application with all functionalities and a code entirely written by ourselves. This page would offer its users to share long stories as we planned first, but instead little quotes they have in mind. They would have to register or login before posting a quote, and they could see people's quotes too. A “refresh” button would be used to actualize the submissions so a user can see the last quotations of other users without having to refresh the whole page.

We focused on the more difficult part this time, which was the back-end, And succeeded to have a website working as we expected.

The last step was the deployment of the project. We first tried on Google Cloud, but even with its documentation, Google Cloud proposes a lot a features and you have to know exactly what you are doing. As we were not used to work with it, we decided to use Glitch which has a more friendly interface. Glitch also proposes to upload your project thanks to GitHub, which is a good point too.

When we finally deployed our website, we checked if everything was fine and corrected all deployment problems between the website and Glitch.

III) HOW DOES IT REPLY TO THE REQUIREMENTS ?

Our project is responding to the requirements since it uses **HTML ,CSS, Bootstrap, Vue.js, Node.js** and **Express**. With those languages we managed to build a full **User authentication** on our Website. We managed to **Create** and **Read** however we didn't managed to implement the **Update and Delete** sides of CRUD on time.

We didn't managed to have a good **data persistence** : data is persistent only on a session. It means that if we restart the server data is lost. All this project is deployed on **Glitch**. This is why our project replies to the specifications of the project.

IV) DIFFICULTIES ENCOUNTERED

Diving into new languages and discovering aspects of the web programming is a challenge for everyone willing to create a website in a few weeks. We faced a lot of bugs and errors we didn't know how to deal with. We finally managed to resolve all of them. However, this wasn't the main issue we had during the realization of the project. The anticipation of the difficulty and the amount of work was probably an aspect we messed up in the project. At least, first. Sometimes we want to make great things, ignoring how to proceed and how long it will take. That's why we didn't have enough time to implement the UPDATE and DELETE options. Next time, we will focus on a simple and reachable project, and then we will add step by step new functionalities to make it better, depending on the time we have left. If we had thought this way first, our website would have been much more complete..!