

Projet Mady

Veille technologique

Groupe Indigoes

Projet de synthèse 2020-2021

Maîtrise d'ouvrage (client) :		LAPIETRA Lorenzo
Maître d'oeuvre (enseignant tuteur) :		ROIRAND Xavier
Maîtrise d'œuvre	Chef de projet :	JOSSE Samuel
	Responsable de communication :	COMBRISSEON Ewen
	Responsable de la documentation :	QUINIQUO Christophe
	Responsable des tests :	GARREC Meven

Date :	23/10/2020
Version :	1.0.0

TABLE DES MATIERE

I. Front	3
1. Les bases du site	3
2. L'animation du site	3
3. La responsabilité du site	3
4. Framework JavaScript	4
II. Back	4
1. Base de données PostgreSQL	4
2. NodeJS	5
III. Documentation	6
1. Documentation du code source	6
2. Documentation du projet	6
IV. Tests	7
1. Tests d'interfaces	7
2. Tests fonctionnels	7
3. Tests de la base de données	7

I. Front

1. Les bases du site

Pour la mise en place du site web du côté front, nous allons nous servir de différentes technologies : dans un premier temps pour le contenu du site web nous utiliserons le HTML.

Le HTML permettra de mettre en place les différents éléments du site web tels que les images, les textes, les liens ou encore le formulaire. HTML sera accompagné du CSS permettant la mise en forme du site web pour le rendre plus attractif. Cela permettra de mettre de la couleur, de dimensionner les contenus du HTML comme les images ou encore les différentes polices. Il permettra également de structurer la page. Nous avons déjà utilisé ces deux technologies dans d'autres projets.



2. L'animation du site

Afin de rendre plus dynamique notre site, nous allons utiliser JS accompagné de sa librairie JQuery pour permettre une animation du site et le rendre plus agréable. Nous avons des bases mais nous ne connaissons pas encore très bien JQuery mais cela va nous permettre de l'expérimenter



3. La responsabilité du site

Comme le site que nous développons devra surtout être utilisé sur mobile et que notre client nous a recommandé de ne pas développer une application, nous ferons en mobile-first. C'est-à-dire qu'on développe à la taille du mobile pour ensuite l'adapter à l'ordinateur.

Il nous faut donc une technologie permettant le responsive et pour cela nous utiliserons Bootstrap. C'est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.



4. Framework JavaScript

Nous avons également discuté avec le client d'utiliser un framework JS. Ils nous ont déconseillé d'utiliser react JS car il est assez difficile à prendre en main surtout si on s'en est jamais servis. Il nous ont plus plutôt conseillé d'utiliser Vue JS donc nous allons nous y intéresser.



II. Back

Dans un site web, il y a le côté visible des utilisateurs (le front), et le côté invisible et pourtant indispensable. Cette partie du site web est appelée back-end. Elle a pour objectif d'administrer le site web en mettant en place les fonctionnalités désirées.

Pour créer cette partie, nous allons utiliser différentes technologies qui présentent des avantages et des inconvénients.

1. Base de données PostgreSQL

Concernant la base de données, nous allons la créer en PostgreSQL. Il s'agit d'un système de base de données libres. Ce système présente de nombreux avantages. Tout d'abord, comme je l'ai déjà dit, il s'agit d'un outil libre fondé sur une communauté mondiale de développeurs et d'entreprises. Ensuite son atout principal est qu'il utilise des types de données modernes ce qui signifie qu'on peut stocker plus de types de données que les types simples comme les entiers, les caractères....Il est possible de stocker des fonctions, utiliser l'héritage de type....Il est aussi possible d'utiliser des interfaces graphiques pour gérer les tables, cela permet de gagner du temps. Enfin, le dernier avantage est qu'il existe de nombreuses bibliothèques pour de nombreux langages comme Java, C++, JS...

Si l'on veut trouver des inconvénients à cette technologie, c'est que nous ne la connaissons pas encore, ce qui signifie qu'on va devoir prendre en compte un temps d'assimilation. Cependant, il s'agit aussi d'un avantage car cela nous permettra d'enrichir nos connaissances sur ce sujet.



2. NodeJS

Node JS est également un logiciel libre. On utilise le langage appelé JavaScript utilisé très souvent dans la conception de sites web. NodeJS est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.

Parlons des avantages de NodeJS.

Tout d'abord, NodeJS est un système non bloquant. Par exemple, si un client fait une requête d'un fichier au serveur alors il lance cette requête sans attendre le résultat. Il n'attend pas car précisément il n'est pas bloquant. Si un autre client vient faire une autre requête, il est tout de suite capable de traiter cette requête également.

Ensuite, NodeJS est très light comme plateforme et n'a pas beaucoup de fonctionnalités déjà intégrées. C'est à nous de choisir quels sont les modules que nous voulons lui greffer. Nous avons déjà prévu d'utiliser le module Express. Il s'agit d'un framework d'application web utilisé comme framework serveur pour NodeJS, qui permet de générer des applications web. express nous permet de choisir différentes infrastructures frontales pour créer une interface utilisateur...



Concernant les inconvénients de cette technologie, c'est qu'il s'agit d'une technologie complète et complexe. En effet, elle comporte de nombreuses fonctionnalités ce qui augmente les difficultés d'utilisation. Une fois de plus, bien que l'on possède des bases sur ce sujet, nous apprendrons en même temps que nous avancerons dans le projet.

Nous allons également utiliser une autre technologie qui est en réalité également un autre module de NodeJS. Il s'agit de node-postgre.

Nous allons utiliser ce module pour connecter nos bases de données et effectuer les requêtes. Voici un exemple de connection entre NodeJS et une base de données

```
const { Pool, Client } = require('pg')

// pools will use environment variables
// for connection information
const pool = new Pool()

pool.query('SELECT NOW()', (err, res) => {
  console.log(err, res)
  pool.end()
})

// you can also use async/await
const res = await pool.query('SELECT NOW()')
await pool.end()

// clients will also use environment variables
// for connection information
const client = new Client()
await client.connect()

const res = await client.query('SELECT NOW()')
await client.end()
```

III. Documentation

Un projet informatique est en général assez complexe, il est donc nécessaire pour l'équipe qui s'occupe du développement de documenter tout au long du développement chaque avancée dans le projet pour que tout le monde soit capable de rejoindre une autre partie du projet à n'importe quel moment. Une fois le livrable donné au client, il est nécessaire pour lui de pouvoir comprendre comment fonctionne les fonctions qui lui sont destinées et aussi pouvoir fournir à ces techniciens des éléments pour les éventuelles améliorations et les maintenances.

1. Documentation du code source

Pour la documentation de nos codes sources j'ai choisi d'utiliser JSDoc pour nos codes fait en javascript qui représente une grande majorité de la partie technique du projet. JSDoc est un outil qui est gratuit mais qui n'est pas libre mais sur la licence creative commun qui est un peu plus restrictive mais qui nous permet quand même de modifier le code et d'en faire une version modifiée pour laquelle nous devons citer l'auteur.

@use JSDoc

Quant aux autres langages utilisés dans le projet comme le HTML, CSS et les requêtes SQL nous utiliserons les commentaires natif de chacun des langages pour faire la documentation.

Commentaire html

```
<!--This is a comment.-->
```

Commentaire en css

```
/* Commentaire */
```

Commentaire en postgresql

```
<!-- Web.Config Configuration File -->
```

2. Documentation du projet

Pour la documentation du projet en générale nous avons choisi d'utiliser BookStack comparé à MkDocs car c'est un outil qui n'est pas libre de droit et dont les pages de documentation générées ont une esthétique moins travailler que sur BookStack. Le déploiement et la maintenance d'une documentation chez BookStack est simplifié.



IV. Tests

Tester son travail, son application est une étape essentielle pour le déploiement d'une application. Sans test, un projet ne peut pas être considéré comme valide ni fonctionnel.

1. Tests d'interfaces

Afin de tester si les interfaces du front répondent correctement aux interactions entre elles et l'utilisateur, nous allons utiliser Selenium IDE.

Selenium IDE est une extension Chrome/Firefox, qui va me permettre de créer des scénarios directement grâce au site web.

Les scénarios vont être faits sur une utilisation normale de l'interface, par exemple si nous voulons tester l'apparition d'un bouton après une inscription. Nous allons enregistrer les étapes à faire pour faire apparaître le bouton. Ainsi, si ce n'est pas le même bouton qui s'affiche, le test donnera un refus.

2. Tests fonctionnels

Pour les tests fonctionnels, cela va être surtout testé le côté back-end mais aussi sur la relation entre le front-end et le back-end.

Pour cela, je vais utiliser Mocha. Mocha est un framework s'utilisant avec NodeJS qui a été créée pour réaliser des tests fonctionnels. Mocha utilise Karma, ce qui va nous permettre d'avoir une plus grande marge de manœuvre sur les tests.

Pour chaque fonctionnalité, des méthodes des tests seront créées grâce aux indications données par les collègues. Puis des scénarios avec des cas nominaux, cas d'erreurs, seront mis en place.

3. Tests de la base de données

Pour tous les tests fait avec Mocha ou Selenium, les résultats seront stockés dans une base de données ou bien dans des fichiers pouvant stocker les infos tels que la fonctionnalité testée, le résultat etc...

Une base de données de test sera mise en place, elle est une copie de la bdd faite pour le back-end. Des scripts de remplissage de tables seront créés afin de démontrer la sécurité et la fiabilité de la base de données.

On ne pense pas utiliser de technologie particulière pour les scripts de tests. Nous utiliserons PostgreSQL.