Choix technologiques

Projet de fin d’étude

Denuit Maxime, Degrève Olivier, Dubois Corenthin, Tavernier cedric

Table des matières

[1. Introduction : 1](#_Toc470879385)

[2. Choix Technologique : 2](#_Toc470879386)

[2.1. Back-End : 2](#_Toc470879387)

[2.2. Base de Donnée : 3](#_Toc470879388)

[2.3. Front-End : 3](#_Toc470879389)

[3. Diagramme d’architecture : 4](#_Toc470879390)

[4. Outils utilisés : 5](#_Toc470879391)

[4.1. Gît – Github : 5](#_Toc470879392)

[4.2. Trello : 5](#_Toc470879393)

[4.3. Visual Studio Code : 5](#_Toc470879394)

[5. Bibliographie : 6](#_Toc470879395)

# Introduction :

Nous allons vous présenter les différents choix technologiques et les outils choisis pour l’outil de gestion demandé par L’Institut Paul Lambin.

Nous vous expliquerons pourquoi nous avons pris le parti de faire un Back-end en Node.js, un Front-end en JavaScript, HTML 5, CSS3 et JQuery ; ainsi qu’une base de données en PostgreSQL.

Nous avons décidé d’utiliser les outils suivant pour notre projet : Git-GitHub pour le partage de donnée, Trello pour l’organisation du projet et Visual studio Code pour l’implémentation.

# Choix Technologique :

## Back-End :

Nous utiliserons un back-end en Node.js. L’avantage de cette technologie est que nous avons les ressources nécessaires dans notre groupe pour l’utiliser, la documentation est complète et libre d’accès est disponible sur internet.

Une autre chose intéressante à propos de Node.js est sa grande modularité ce qui nous permettra d’adapter le produit tout en maitrisant les couts. Il existe beaucoup de librairies externes permettant d’effectuer des tâches pour un back-end. Que ce soit des librairies serveurs, de requêtes HTTP/HTTPS ou autre. Ce même genre de librairie existe aussi pour le front pour avoir une interface simple à mettre en place (Exemple : AngularJS) ou pour avoir des fonctions temps réel grâce aux WebSockets de Node.js.

Si l’on compare Node.js au Java pour le back-end, il y a clairement plusieurs avantages. D’abord, l’utilisation de Node.js permet le développement d’une application de type « full stack » (c’est-à-dire que l’entièreté du code est en JavaScript) ce qui permet de ne pas avoir de middleware pour convertir d’éventuelles données transmises par le front. Cela permet d’évité de problème de conversion et de complexité du programme (Si nous avions utilisé le Java cela aurait pu causer des problèmes de conversion, il nous aurait fallu convertir le JSON avec Genson).

Ensuite, Node.js a l’avantage d’être plus rapide que java pour faire de l’Input/Output et ne bloque pas d’éventuels requêtes supplémentaires. En effet, il y a une bonne gestion de la concurrence en Node.js ce qui permet de pouvoir effectuer des requêtes Input/Output en parallèle. Bien entendu, il est possible d’implémenter de la concurrence en Java mais cela requiert plus de temps et engendre potentiellement plus de problèmes si mal implémentés.

Node.js a aussi l’avantage d’avoir une très bonne scalabilité. En effet, on peut déployer rapidement une application Node.js pour une entreprise que ce soit une application web ou mobile. De plus, Node.js à l’avantage d’avoir de très bonnes performances notamment de par la rapidité d’exécution des requêtes.

Cependant, Node.js est là principalement pour effectuer des tâches peu complexes. En effet, si l’application requiert des calculs complexes, il serait alors préférable de donner la charge à un autre processus Node.js. Ce problème n’est pas rencontré en Java car les taches complexe peuvent être exécutées en un seul processus.

Niveau structure, Node.js peut utiliser des modules supplémentaires pour intégrer plus de fonctionnalités. Ces modules se situeront dans un dossier « node\_modules » à la racine du dossier contenant le projet. Sinon, il est préférable d’avoir quelques dossiers avec les différentes parties du code et d’éviter les sous-dossiers pour ne pas avoir de chemins trop longs.

## Base de Donnée :

Nous avons choisi de faire une base donnée en SQL et non en NoSQL pour la raison suivante :

Comme dit lors de notre première rencontre avec le client, nous devrons implémenter un outil de gestion. Si l’outil de gestion implique de nombreuse relation entre les différentes données, nous ferons le choix d’utiliser une base de données SQL. Sinon, NoSQL pourrait être envisagé. En effet l’implémentation d’une base de données NoSQL relationnelle est possible mais complexe.

Si nous faisons le choix du SQL, nous prendrons une base de données PostgreSQL. Ce qui a comme avantage de pouvoir gérer un grand nombre de données. En outre, un support important existe et PostgreSQL étant une technologie libre, les couts pour la société seront plus faibles (pas de licence commercial). Nous avons écarté le choix de PgSQL car cela ne convient que pour des petits sites.

Si finalement le choix se porte sur le NoSQL, nous ferons le choix d’aller vers du MangoDb qui a comme avantage de pouvoir stocker d’importante donné et d’avoir une performance élevée.

## Front-End :

Comme langage de programmation pour le front-end nous avons décidé d’utiliser le JavaScript.

Premièrement nous avons décidé d’utiliser le JavaScript car il permet d’agir directement et ne dois pas attendre que les serveurs envoient une réponse. Ceci accélère l’ouverture des sites web sur les navigateurs des clients. Le JavaScript ne nécessite pas un programme d’interprétation comme par exemple Flash player. En outre, le JavaScript n’occupe pas une grande place sur disques des sites web. Nous utiliserons aussi une bibliothèque JavaScript, JQuery qui nous assure une compatibilité multi-navigateurs, une simplification et normalisation des scripts.

Deuxièmement comme nous avons décidé d’utiliser le node.js comme langage back-end, cela nous permet de travailler en « full-stack ». C’est à dire, de pouvoir programmer toute l’application dans un même langage et ne pas devoir utiliser un langage intermédiaire pour rendre compatibles nos données entre les différents langages.

Pour la mise en forme du site nous avons décidé d’utiliser l’HTML5 (HyperText Markup Language 5) et le CSS3(Cascading Style Sheet 3).

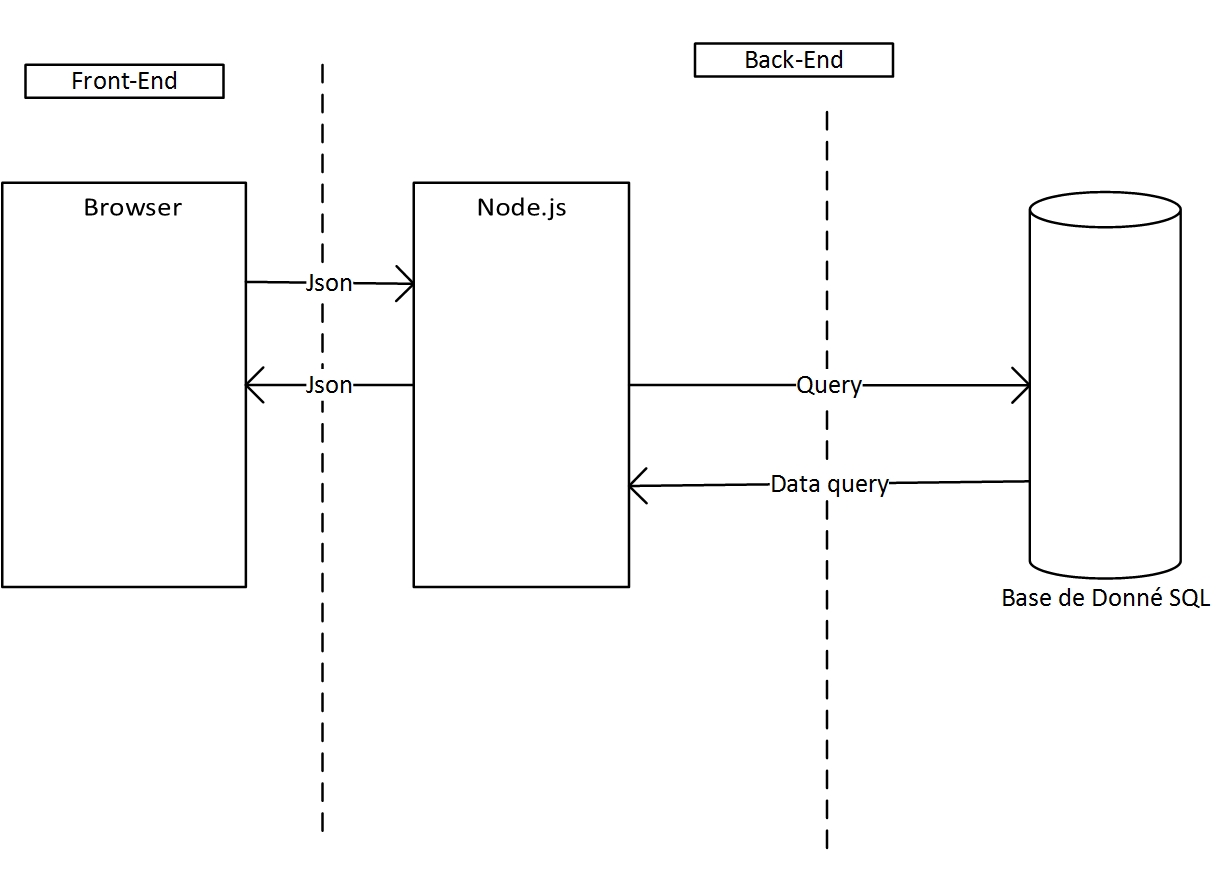
Nous avons décidé d’utiliser la dernière version de ces langages car ceux-ci nous permettent d’utiliser toutes les dernières fonctionnalités.

Un inconvénient mineur est que l’on devra tenir en compte du fait que certains navigateurs ne prendront pas en charge la dernière version de l’HTML5. Pour régler ce problème, nous devrons donc coder en fonction du navigateur le moins « up to date ».

Pourquoi nous n’allons pas utiliser Bootstrap ?

Il nous semble intéressent de pouvoir créer de l’HTML / CSS « fait maison », ce qui nous permet d’utiliser notre HTML/CSS dans nos codes JAVASCRIPT facilement, avec les balises/identifiants qui sont claire et lisible. De ce fait, il nous semble intéressant de créer nous même un site responsif et non d’utiliser un Template tout fait, comme proposé par Bootstrap.

# Diagramme d’architecture :



# 

# Outils utilisés :

## Gît – Github :

Pour mener à bien ce projet, trois outils distincts vont être utilisés.

Git-GitHub en est le premier. D’une part, l’hébergement de projets et de dépôts, sont pris en charge par le site web GitHub. Cela facilite la collaboration sur des projets entre plusieurs intervenants. D’autre part, nous avons un logiciel qui permet aux utilisateurs de sauvegarder différentes versions de fichiers durant le cycle de vie d’un projet sur leur pc, à savoir : Git. Cette application permet également aux utilisateurs d’envoyer leurs différentes versions sur leur compte GitHub ou sur un projet partagé. On pourra ensuite « merger » les fichiers de tous les collaborateurs du projet en un travail unique.

Pourquoi s’être tourné vers Git-GitHub ?

Parce que GitHub est considérée comme étant la plateforme d’hébergement de référence. Elle réunit plus de 5,8 millions de développeurs actifs et la société lève en permanence des fonds importants pour améliorer leur service mais aussi pour y ajouter de nouveaux outils.

GitHub va également nous permettre de communiquer directement avec Trello, notre second outil.

## Trello :

Trello est un logiciel permettant d’organiser nos différentes tâches, de les assigner à chaque développeur et de visualiser l’évolution de notre projet. Grâce à la fonctionnalité GitHub Power-Up de Trello, il nous est permis de lier notre compte GitHub à Trello. On pourra ainsi joindre un commit, une pull-request et en voir son état d’avancement.

Mais pourquoi Trello et pas un autre ?

Parce que Trello comptabilise plus de 10 millions d’utilisateurs dans le monde et est utilisé par des entreprises renommées, telles que PayPal, Google, Adobe, … Le logiciel est adapté pour les mobiles, point important pour visualiser à n’importe quel moment les différentes tâches qu’il nous reste à réaliser.

## Visual Studio Code :

Nous avons pris comme décision de développer sur Visual Studio Code car Il est présent sur tous les system d’exploitation, est gratuit, a un system de terminal intégré et est disponible chez le client. C’est un outil extensible et personnalisable.

# Bibliographie :

1. Documentation de la base de donnée :
   1. <http://blog.postgresql.fr/index.php?post/drupal/216>
   2. <https://www.mongodb.com/fr>
2. Documentation du Back-End :
   1. <https://nodejs.org/en/about/>
3. Documentation d’outils utilisés :
   1. <https://trello.com/business-class?utm_source=blog&utm_medium=sidebar_link_v2&utm_campaign=bc3>
   2. <https://guides.github.com/activities/hello-world/#what>