UNIVERSITÉ DE MONTPELLIER

POLYTECH MONTPELLIER

Rapport de projet

LifeDash

Réalisé par

Kévin GIORDANI

Sous la direction de

Arnaud CASTELLTORT - Anne LAURENT

Pour l'obtention du diplôme d'ingénieur

Année universitaire 2016- 2017

Remerciements

Je souhaite remercier tout d’abord M Arnaud CASTELLTORT, pour les conseils et le temps consacrés au projet. Il a été d’un très grand conseil.

Je tiens à remercier mon enseignant Mm Anne LAURENT qui m’a appris à utiliser les base de données relationnelles.

Et pour finir, je voudrais remercier tout le personnel administratif, enseignant et camarades, qui m'ont appris et aidé lors de ce projet.

Sommaire

Table des Figures

Introduction

Aujourd'hui l'accès à l'information est facilité par internet et tous les autres vecteurs de communication, cependant elle n'est pas forcément centralisée en un point. Si l'on veut obtenir deux informations distinctes il nous faut généralement accéder à plusieurs lien et ce tous les jours.

Le projet consiste à centraliser l'information en un point permettant ainsi d'accéder au information importante en un instant. Ce site permettra à l'utilisateur d'accéder à du flux d'informations ainsi que de stocker de l'information qui pourras lui être utile dans le futur.

Notre problématique sera de réaliser un site web qui permet d'accéder à de l'information rapidement et clairement en fonction des demandes de l'utilisateur .

Après avoir mis au point la définition du sujet du projet, l'ensemble des besoins fonctionnels et les contraintes technique, je présenterais, dans le rapport technique, le fonctionnement du sit, pour ensuite me concentrais sur toutes les fonctionnalités implémentées dans le programme. Ainsi qu'un rapport d'activité rendant compte des méthodes de travail mises en œuvre dans ce projet.

1 Analyse

1.1 Analyse du sujet et de son contexte

1.1.1 Analyse de l'existant

Il existe déjà des sites permettant de centraliser l'information en un point en fonction de ces besoins, on retrouve notamment **exist.io**. En lui donnant accès à certains de vos comptes sur des réseaux sociaux et autres applications de santé vous permet de centraliser l'information. Il propose aussi des outils d'analyse de la données pour permettre un meilleur traitement. La plateforme est cependant payante. Ce qui est une des raisons pour lesquels j'ai decidé de me lancer dans ce projet et de créer mon propre portail qui repondra a mes besoins journalier en information.

Je me suis donc inspirée de quelqu'une des fonctionnalités implémentées dans **exist.io**. Notamment l'accès à la météo et au prévisions météorologique ou bien le calendrier.

Ce projet est un projet personnel avec pour cible moi, j'en serai le seul utilisateur, l'information seras donc façonnés comme j'en éprouve le besoin. Le résultats final du projet differeras donc de **exist.io** tant par ca forme que par son fond.

1.1.2 Analyse de l'environnement dans lequel le site va être utilisé

Le site seras utilisé de manière journalière et courte, l'accès au site seras bref. L'accès au informations relative à la météo ou bien à l'emploi du temps, seras faites chaque matin. L'information a donc besoin d'être facile d'accès et rapidement assimilable. A côté de cela on a une interface permettant la saisie d'information relatives à la pratique sportive qui seront ensuite analysé par l'individu, cette interface doit être simple et permettre une saisie rapide car lors de la saisie l'utilisateur seras surement en pleine pratique sportive, c'est donc une nécessité qu'il ne perdent pas trop de temps à enregistrer les données.1.2 Analyse des besoins fonctionnels

1.1.3 Spécifications fonctionnelles

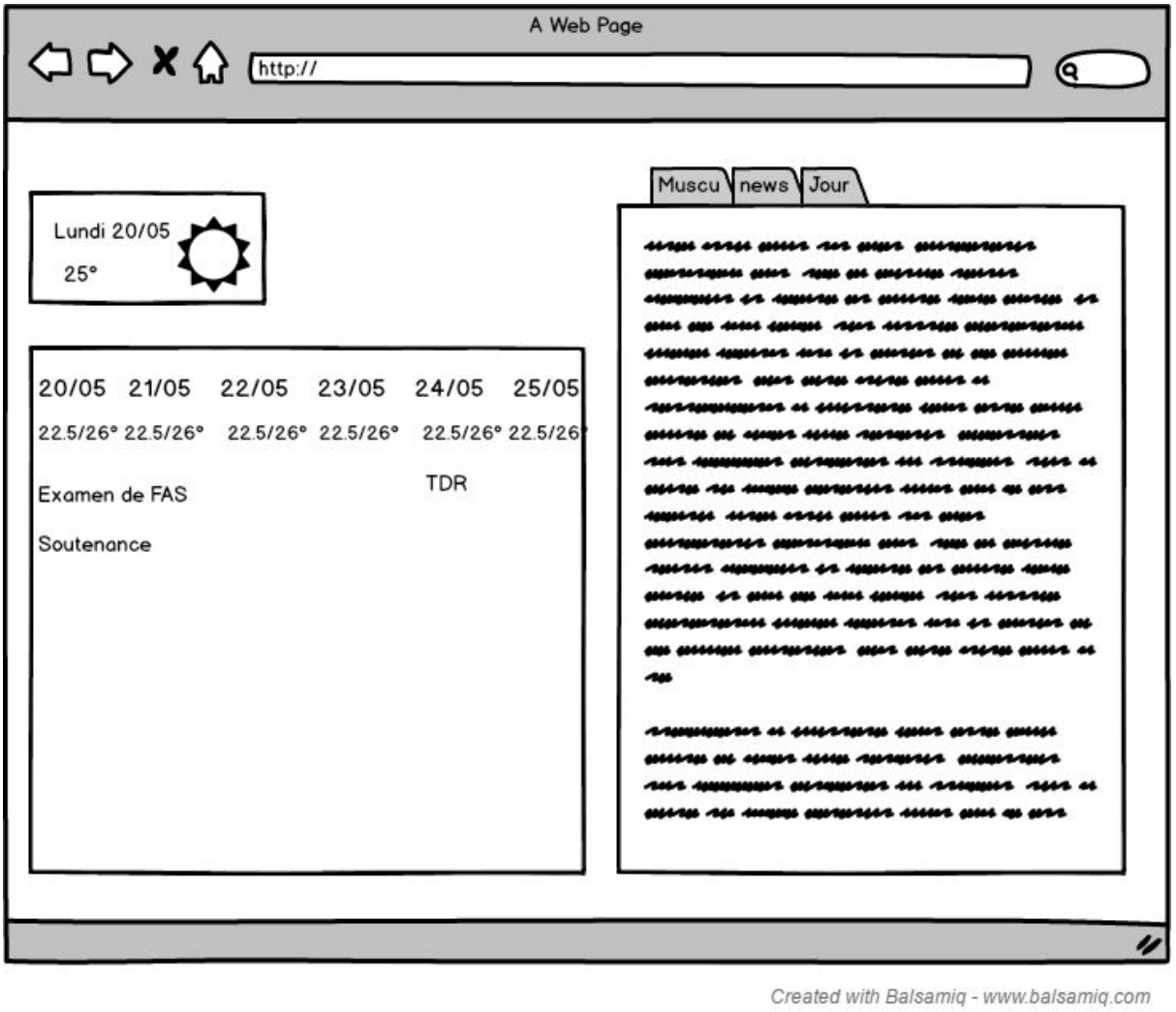

Figure 1 - Maquette de l'interface graphique

La réalisation de la maquette a été faite selon les critères du cahier des charges.

- Il y aura une fenêtre dans laquelle :
    - on retrouveras une interface de saisie pour la musculation.
    - un accès à des news venant du monde en continue.
    - une interface de saisie permettant de noter les journées et de leurs mettre un petit commentaire.
- Voir la météo actuelle ainsi que la date.
- Un bloc permettant de voir l'emploi du temps sur 5 jour avec aussi des informations relatives au température minimale et maximale des jours a venir.

1.2 Analyse des besoins non fonctionnels

La réalisation du site sera faite en PHP du côté serveur, PHP a était le langage choisis car c'est un langage de programmation avec une grande communauté ce qui permet un accès à des tutoriaux ou même de l'aide plus facile que pour d'autres language, aussi le manque de temps a était un sérieux frein à l'apprentissage d'un autre langage. Côté client, le langage choisi a était Angular car il semble parfait pour l'implémentation d'un site en une page. La base de données est une base de données MYSQL car elle ne prendras pas de grande ampleur étant données que j'en serai le seul utilisateur, les données stockées ne dépasseront sans doute jamais les 3000 tuples. Certains élément graphique venant de Bootstrap ont aussi été utilisé permettant un meilleur rendu graphique.

2 Rapport technique

2.1 Conception

Une fois le sujet bien cernés et les limites exposé, il a fallu élaborer un Modèle Logique de Données, qui permettra d'établir la base de données.

L'interface doit permettre de créer/lire/modifier/supprimer des séances, exercices, d'affecter des exercices à des séances. Elle doit permettre la saisie de performance. Les performances peuvent être différentes selon le type d'exercices : Charge ou temps. Les performance peuvent être liés à une séance ou bien seulement liés à un exercice. Les note des jour doivent être identifiés par la date et contenir une note entre 0 et 10 et un commentaires. Voici donc le MLD :

Seance(IdSeance,Titre,Objectif)
Exercice(IdExercice,Titre,Description,#IdType)
TypeExercice(IdType,Type)
AffiliationSeanceExercice(#IdSeance,#IdExercice)
Performance(IdPerformance,#IdExercice)
PerformanceTemps(#IdPerformance,IdPerformanceTemps,Temps,Jour)
PerformanceCharge(#IdPerformance,IdPerformanceCharge,Serie,Repetition,Charge
,Jour)
AffiliationPerformanceSeance(#IdPerformance,#IdSeance)
NoteJour(Jour,Note,Commentaires)

Le déploiement du site est fait sur ma raspberry personnelle il est disponible a l'adresse **http://rifhice.com/LifeDash/v1**. Ce choix se justifie par la proximité du serveur qui me permet de le configurer entièrement comme je le souhaite. De plus le site n'auras pas d'expansion tant au niveau de la BDD que du trafic, la raspberry étant déjà installé chez moi possède donc une facilité d'utilisation qui ma poussé a choisir ce mode de déploiement.

Il fallait ensuite définir les flux d'information que l'on voulait afficher, il y avait donc la Météo et les prévisions, les News et l'emploi du temps. Je me suis donc tourner vers des API me permettant d'accéder à ces information, pour ce qui est de l'emploi du temps, l'université donne accès à une API permettant de récupérer l'emploi du temps de Polytech, je m'en suis donc servi. Ensuite pour la météo j'ai choisi **OpenWeatherMap** qui dans sa version gratuite donne accès aux températures toutes les 3h sur 5 jours, j'ai donc dû traiter l'information pour la changer en

prévisions journalière. Enfin pour les news j'ai utilisé l'API de **newsapi** qui permet l'accès à plusieurs source d'information, parmis le vaste catalogue, j'ai choisi d'utiliser Google news comme source d'informations.

Une fois les technologies choisis et les source d'information décidé, il a fallu que j'apprenes AngularJS notamment en suivant le cour de **codeschool** dédiés à AngularJS.

La première étapes a était d'interfacer avec les API choisies pour récupérer les informations qui seront ensuite afficher. Une fois les clés d'accès récupéré et la connexion établie le traitement et tris des données obtenu est fait du côté du serveur. Il ne me reste donc plus qu'a appeler les différentes ressources depuis le côté client et afficher à l'utilisateur les informations du JSON récupérer. Le client redemandes les ressources toutes les heures, pour garder l'information actualisé.

Les interfaces de saisies ont ensuite dû être élaborée, une fois la connexion a la BDD faites, les accès au tables on était divisé en fichier PHP, chaque fichiers permet la connexion à une table et agit sur cette table en fonction du verbe HTTP. Le côté client va donc faire des appels asynchrone au ressource dont il a besoins, ces appels seront ensuite redirigé par le fichier .htaccess qui va appelé le fichier PHP concernés qui renverra la données en JSON.

L'arborescence des fichiers se présente comme suit :

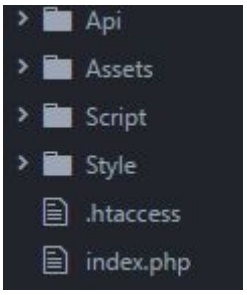

Figure 7 - Arborescence des fichiers

Dans le dossier Api, on retrouves l'ensemble des scripts PHP permettant d'effectuer la connexion et les actions sur la BDD, le dossier Assets on retrouve l'ensemble des images et videos nécessaire. Le dossier Script contient le script Angular pour le client, Le dossier Style contient les fichiers css.

Le fichier index.php est le point d'ancrage du site et la seule page. Le fichier .Htaccess permet la redirection des appels au ressources vers les fichiers PHP correspondant.

2.2 Résultats

Le site répond au cahier des charge et propose bien une centralisation de la données. Le site est divisé en trois bloc. Le premier bloc en haut à gauche permet d'afficher la température extérieure courant à montpellier ainsi que la Date. Le second bloc à gauche permet quant à lui d'afficher l'emploi du temps d'IG3 de Polytech montpellier sur 5 jours, en parallèle il affiche les températures minimale et maximale attendu pour les prochain jours. Enfin le troisième bloc sur la droite est divisé en 3 onglets, l'onglet "Muscu" qui propose une interface de saisie et de lecture pour les séances, les exercices et les performances, le deuxième onglet "News" permet l'affichage des nouvelles de Google News, en cliquant sur une d'elles on est redirigé vers l'article en rapport, finalement le troisième onglet "Note jour" propose une interface pour noter les jour de 0 a 10 et d'y écrire un commentaire.

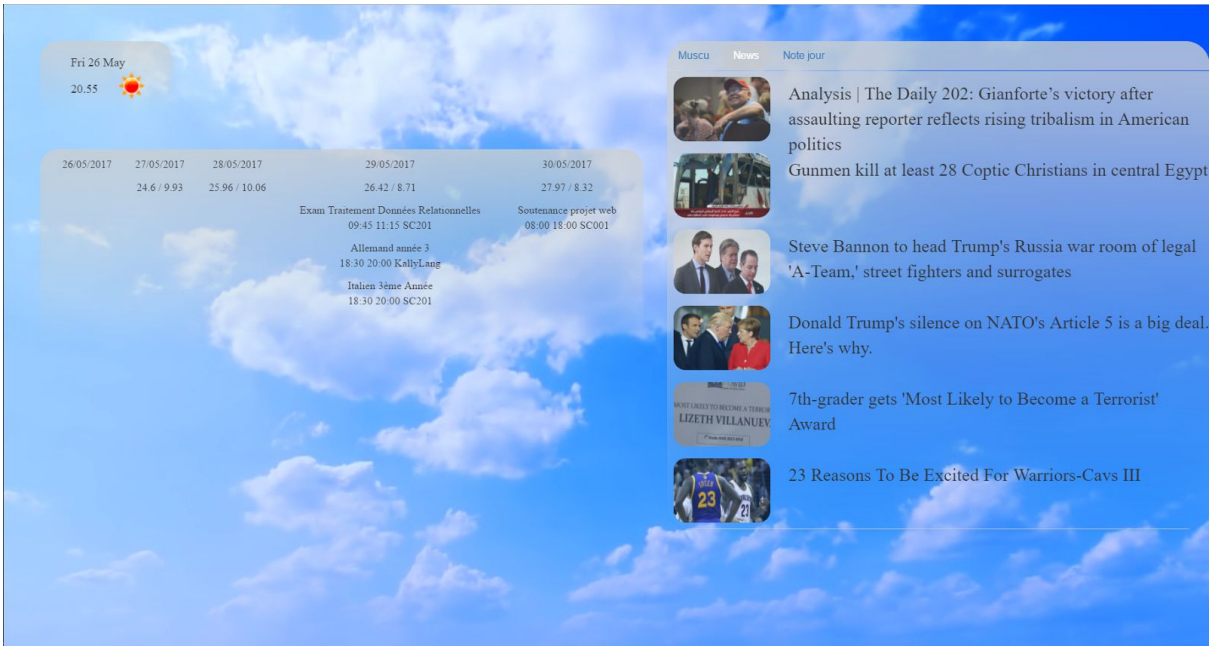

Figure 8 - Capture d'écran du site

2.3 Perspectives

Le site est fonctionnel, il reste néanmoins des imperfections et des points à améliorer. Graphiquement, l'interface pourrait être plus attrayante et ergonomique. En effet, les champs de saisies ne sont pas très agréable à l'oeil, même l'agencement général de l'information n'est pas très bonne.

Des problèmes ont été rencontrés, notamment avec le fait que les appels aux API était des appels asynchrone. Je n'ai pas réussi à faire en sorte que lorsqu'une données est modifié dans la base de données elle soit directement modifié sur la page, d'où l'implémentation d'un bouton "Refresh" permettant de rafraîchir les informations spécifique sans recharger la page.

Avec plus de temps j'aurais aimé me pencher sur NodeJs pour le côté serveur, ainsi que peut-être utilisé les Routes sur AngularJs qui aurait sans doute permis une meilleure réactivité du site.

Au niveau des fonctionnalités, le site peut encore s'améliorer, on peut imaginer d'autres interfaces de saisies d'information comme par exemple la saisie de données relatives au sommeil ou bien à l'alimentation. Au niveau des notes de jours on pourrait changer les couleurs du texte en fonction de la note. Par exemple si un jour a une note supérieure à 5 la couleur serait verte, et rouge dans le cas contraire. L'API de météo aussi semble renvoyer des données par forcément très exacte, il faudrait peut-être envisagé à en trouver une autre ou bien se penché vers les API payantes pour une meilleure précision. Le traitement des données relatives à la météo serait un plus cela permettrait d'avoir des alertes lorsque de la pluie ou bien de grosse chaleur sont annoncés. Une interface de saisies d'événements qui pourrait être intégrée dans l'emploi du temps et enfin une interface de prise de notes pour écrire des idées ou rappels.

Toutes les fonctionnalités cités plus haut sont des idées qui avec plus de temps pourrait prendre vie et ajouter de la valeur au site.

3 Rapport d’activité

3.1 Méthode de développement

Le cahier des charges a été établie au début du projet. Il a permis tout au long du projet de savoir vers où aller. Les objectifs était clairement, l'établissement d’une maquette tôt a permis aussi de bien guider les choix visuels.

3.2 Planification

Figure 20 - Planning

3.3 Méthodes et outils de travail

Le site étant déployé sur ma raspberry, j’ai ouvert les port pour le FTP et le SSH, ce qui me permettait de travailler directement dessus depuis n’importe ou, j’ai aussi télécharger un plugin “RemoteFtp” sur le logiciel de traitement de texte Atom. Ce plug-in me permettait que des que je sauvegarder en local un fichier présent sur la raspberry, il envoyé les modification via FTP directement.

Conclusion

Les objectifs de ce projet été de développer un site web à partir d'un cahier des charges. Il avait pour but d'éveiller notre curiosité quant aux web et à ces mécaniques.

Les résultats obtenus au regard des attentes initiales ne sont pas si différents. Ce projet a permis une autoformation à AngularJS, il a aussi permis une ouverture au URI et au verbes HTTP, des notions qui m'étaient étrangères au début de ce projet. La maquette faite lors de la phase de conception a été respectée. Ce projet as aussi permis de mieux comprendre comment gérer un projet pour lequel je suis le client et le développeur, cela m'a appris sur ma capacité de travail et d'apprentissage. Bien que du point de vue du cahier des charges initial le projet est finis je suis tout de même amener à penser qu'il reste des perspectives d'évolution comme l'amélioration de l'interface graphique ou encore l'ajout de fonctionnalités.

Je conclurais sur le fait que le projet a été intéressant non seulement sur la manière dont j'ai pris en main le sujet mais aussi sur l'aspect de la programmation d'un site web.

Annexes

Lien vers le site : http://rifhice.com/LifeDash/v1/
Lien vers le dépôt GitHub : https://github.com/Rifhice/LifeDash/

Quatrième de couverture

LifeDash contraction de “Life Dashboard” est un projet qui implique d’utiliser le web pour centraliser les informations qui peuvent s’averer utile au jour le jour, comme la météo ou encore l’emploi du temps. Il permet aussi des interfaces de saisies pour le stockage de données relatives à la pratique du sport ou au bien être. Ce projet est destinés à une utilisation personnelle seulement, il a été réalisé dans le cadre d’un projet universitaire pour Polytech Montpellier. Les technologies utilisé sont PHP, AngularJS and MYSQL. Le site est actuellement en ligne a l’adresse : http://rifhice.com/LifeDash/v1/ .Il est hebergé sur une raspberry pi 3 personnelle.

Mots clés : Flux Informations, Web, Centralisation, AngularJS, PHP, MYSQL, Raspberry

LifeDash is the short term for “Life Dashboard” it’s a project which uses the web to centralise information that are useful on a daily basis, such as weather forecast or time schedule. It also allows the user to input data for it to be saved in a database so he can analyse it later, those data are related to sport and wellbeing. This project is strictly designed for a personal use, it has been made for a project at Polytech Montpellier, France. The technologies used were PHP, AngularJs and MYSQL. The Website is currently available at : http://rifhice.com/LifeDash/v1/ . It’s hosted on a personal Raspberry Pi 3.

Keywords : Information flow, Web, Centralizing, AngularJS, PHP, MYSQL, Raspberry