

PROGRAMMATION WEB

IMAC2

— 2019-2020 —

RAPPORT PROJET

Andréa Guillot — Baptiste Ory — Brice Mangeat

Solène Mary-Vallée — Victor Julien

[Github](#) — [Dashboard](#)

I.	Présentation du projet	2
II.	Présentation de nos choix	2
1.	Choix techniques	
2.	Choix graphiques	
III.	Gestion de projet	5
IV.	Satisfaction du projet	6
V.	Prochaines MàJ	7
	Annexes	8
	Sitographie	10



I. Présentation du projet

Notre projet est une application web et s'appelle « IMAC League : superhero dashboard ». Il s'agit d'un tableau de bord pour superhéros. Le héros fictif peut ainsi utiliser fictivement cette application pour voir son identité, consulter la météo du crime, traiter ses tâches et prendre connaissance de ses capacités. Il y a six tableaux pour six superhéros différents, avec toujours les mêmes modules, mais un thème et des informations différentes. Notre projet utilise les outils et techniques modernes du web : la plateforme logicielle Node.js, un empaqueteur d'application, un framework avec un modèle vue-contrôleur, des APIs, des bibliothèques JavaScript et ES6.

Notre but était d'avoir un tableau de bord fonctionnel, simple, facilement compréhensible et avec sa propre personnalité. Nous voulions aussi en profiter pour mettre en pratique ce que nous avons vu en cours ce semestre (ou même les précédents) et découvrir des technologies web. C'était aussi l'occasion d'apprendre à mieux gérer le travail de groupe, d'autant plus que nous devons exceptionnellement faire tout cela à distance.

II. Présentation de nos choix

1. Choix techniques

Parcel est idéal pour des projets définis sur le court terme avec des technologies standards. Parcel est également **plus performant** sur le temps de compilation, en général deux fois plus rapide que Webpack. Alors que de nombreux empaqueteurs demandent d'installer et de configurer des plugins pour transformer les ressources, Parcel a un support prêt à l'emploi pour de nombreuses transformations telle que la **conversion en ES6**. Tout de même, nous étions contraint d'ajouter Babel en tant que dépendance pour permettre la conversion en JSX.

La contrainte de temps nous demandait de réévaluer les compétences que nous avons déjà, car l'apprentissage de nouvelles technologies n'aurait pas été viable. C'est pourquoi, **Hyperapp** semble être une évidence pour ce type de projet : **intuitif et rapide à prendre en main**. De plus, Hyperapp utilise la même logique de component que les frameworks les plus populaires (React, Vue).

Pour améliorer la **lisibilité et la cohérence du code**, nous avons également mis un **Linting**, avec les règles standards, ainsi que des règles plus spécifiques à notre façon de développer. Nous avons également établi des standards concernant le nommage des variables et les commentaires (cf Standards). Concernant le CSS, nous hésitions entre du Less ou du SASS. Les



différences sont très minimes, beaucoup de subtilité dans la syntaxe. **SASS** étant le plus utilisé, nous avons choisi ce dernier.

Flexboxgrid, un système de grille sous flex très pratique pour la **mise en forme du dashboard**. Cependant, nous avons dû beaucoup l'adapter en fonction de nos besoins. À refaire, nous ne le prendrons peut-être pas. Chart.js ou D3.js : les critères étaient le temps, mais aussi un graphique de type radar. D3.js est une librairie où tout est faisable, mais le **temps d'apprentissage** est peu plus long. **Chart.js** a une prise en main très **rapide et répondait à nos besoins**.

Au départ, nous n'avons pas installé le plugin de transformation pour le **JSX** : il n'était pas nécessaire. En cours de route, nous l'avons ajouté, car le besoin s'est fait ressentir lorsque nous avons ajouté des SVG. Pour l'animation de certains composants, nous avons utilisé la librairie **GSAP** : complète, bien documentée et avec une communauté importante.

Deux APIs publiques sont utilisées **SuperHero API** et **OpenWeatherMap**. L'API des super-héros nous apportent toute la **donnée nécessaire** (GET) à notre Dashboard. L'API météo était utilisée pour rendre dynamique la météo du crime (GET). Concernant la liste des tâches, nous voulions aller plus loin que le TP. Les requêtes **POST**, **DELETE** et **PUT** sont fonctionnelles car nous avons créé notre **propre API** contenant les tâches à faire. Le backend est conçu en Node.js grâce au CMS **Strapi** et la base de données est hébergée chez **MongoDB**. Strapi rend la création d'une API très rapide. Le déploiement de l'API s'est fait grâce à **Heroku**.

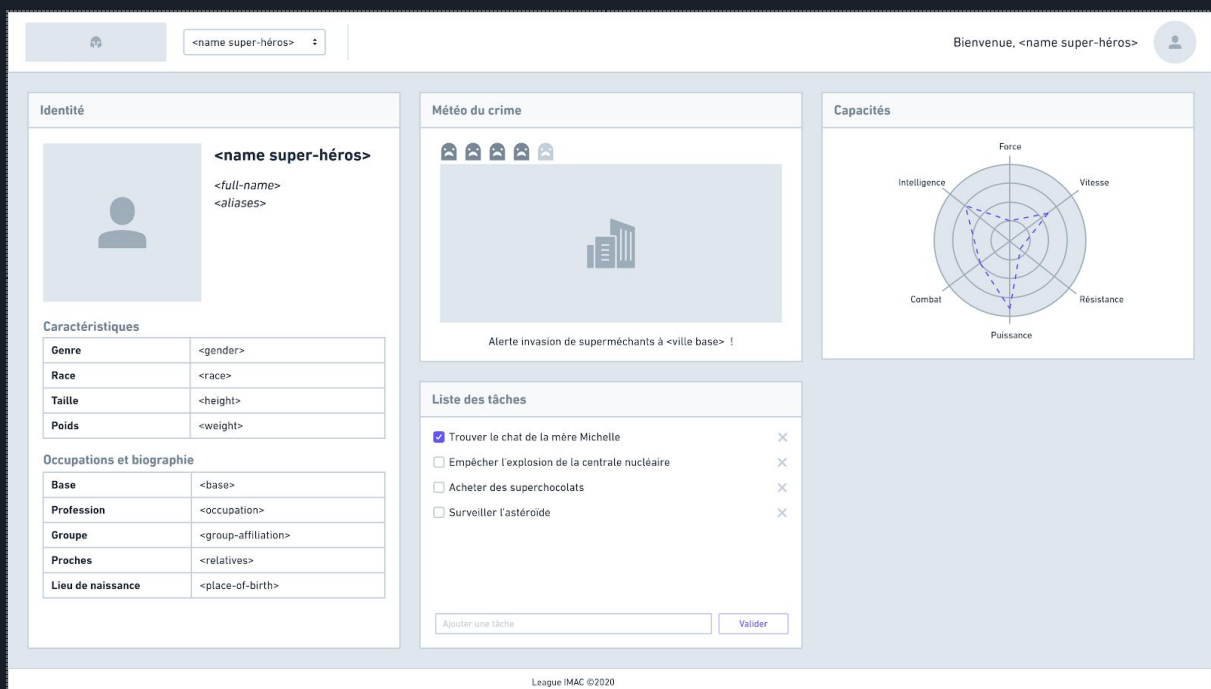
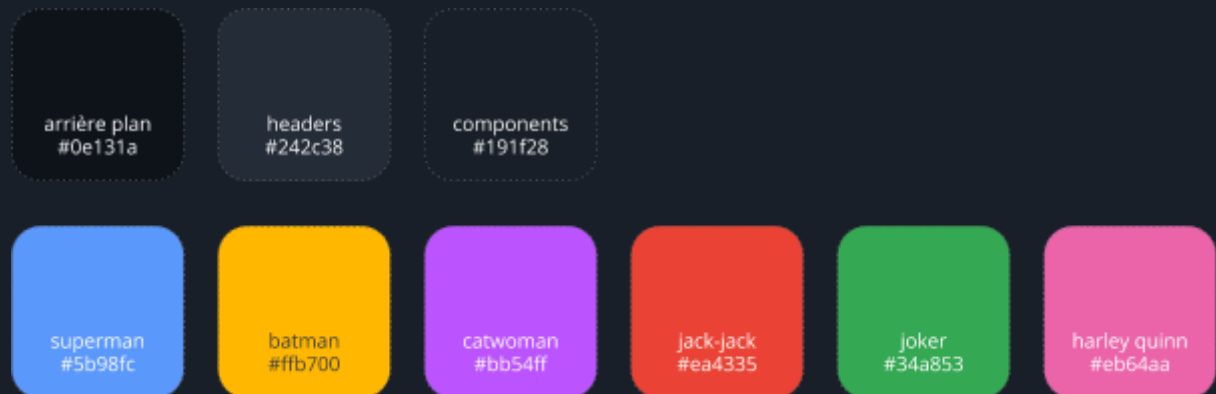
2. Choix graphiques



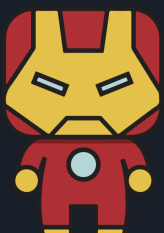
Avatars super-héro et logo IMAC League

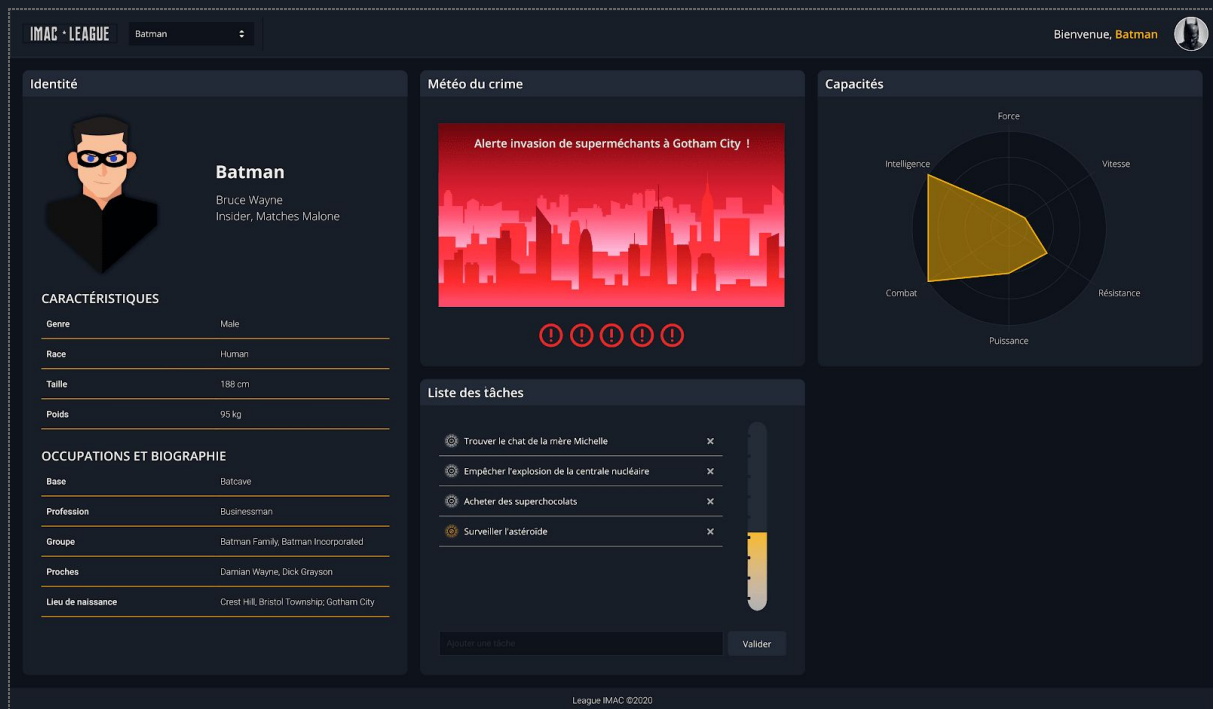
Les **avatars** sont stylisés en fonction des **caractéristiques** (couleur de cheveux et couleur des yeux) du héros choisi.

Après deux proposition de maquette, l'une avec un thème clair, l'autre avec un thème sombre, nous avons opté pour le **thème sombre** pour notre dashboard. À cela s'ajoute une touche de **couleur**, que l'on retrouve à divers endroits, propre au héro sélectionné.



Wireframe





Maquette

III. Gestion de projet

Tout au long de ce projet, nous avons consigné notre évolution dans un **Trello** (cf. annexe), ce qui nous a permis de nous repérer dans le temps et de répartir la charge de travail.

Dans un premier temps, nous avons effectué un travail dit « préalable ». Celui-ci consistait à faire des recherches sur les outils technologiques dont nous aurions besoin, organiser le setup du projet, choisir quelle direction nous voulions prendre et, enfin, concevoir le wireframe et la maquette du dashboard.

Ensuite est venue la phase de « développement ». Pour le développement du projet, nous avons fragmenté le travail et chaque membre du groupe s'est occupé d'une feature de A à Z, de telle sorte que nous puissions tous toucher à tout dans le code (JS et CSS). Bien entendu, cela ne nous a pas empêchés de nous entre-aider, si par exemple l'un de nous rencontrait des difficultés sur la sienne. Finalement, une fois le code achevé, nous avons attaqué les étapes de « vérification » du code ainsi que la rédaction de ce rapport, en prenant soin, une fois de plus, de nous répartir le travail.

Régulièrement, nous avons organisé des réunions sur **Discord** afin de nous assurer du bon déroulement du projet. Ainsi, tous les 3 - 4 jours, nous nous sommes réunis afin de discuter de ce qui avait été fait, de ce qui était en train d'être fait et de ce qu'il nous restait à faire. À la fin de celles-ci, tout le monde repartait en sachant ce qu'il devait faire.



Au cours de ces réunions, nos prises de décision se sont le plus souvent déroulées sous la forme d'un vote. Que ce soit sur [Google Sheet](#) ou bien sur [Straw Poll](#), chaque membre du groupe récoltait de son côté une certaine quantité d'idées que nous mettions par la suite en commun. Puis, nous choisissons individuellement et selon nos préférences lesquelles nous semblaient les meilleures. Finalement, nous avons retenu les idées qui intéressaient la majorité.

Quelques exemples de sujets soumis à un vote : le thème du dashboard, les features à coder, les héros/vilains à représenter, les couleurs du dashboard

Concernant la gestion de [Git](#), nous avons instauré certains standards tels que la création de branches pour chaque nouvelle feature. Les messages des commit sont également normalisés. (cf [Standard](#)). Nous avons également mis un système d'archivage des branches avec des Tags.

IV. Satisfaction du projet

Solène

Travailler sur ce projet m'a permis de mieux comprendre le fonctionnement d'Hyperapp et d'essayer de consolider mes bases en JavaScript et en web de manière générale. La répartition de notre travail m'a fait toucher un peu à tout : JS, Hyperapp, CSS, manipulation d'une API. De plus, j'ai pu découvrir la librairie GSAP pour réaliser des animations, ce que j'aimerais éventuellement approfondir davantage. J'ai rencontré de nombreuses difficultés durant ce projet, néanmoins j'ai pu les surmonter, parfois seule, parfois grâce aux autres membres de l'équipe. Dans l'ensemble, j'ai apprécié travailler sur ce projet. C'était satisfaisant de traverser toutes les différentes étapes, de la réflexion, la conception (recherche graphique, maquette...) à la réalisation, pour atteindre le résultat final.

Andréa

De manière générale, j'ai beaucoup apprécié travailler sur ce projet. Un projet de programmation ou de développement web est pour moi l'occasion de progresser plus rapidement dans le domaine. En effet, il me laisse le temps de tester, de me tromper et de recommencer. De plus, travailler sur un sujet concret me stimule et me motive. J'ai éprouvé quelques difficultés dans la syntaxe du code, car j'ai tendance à me mélanger avec les différents langages que je connais. Pour pallier ce problème, j'ai repris les TP vus en cours, fait des recherches sur Internet ou bien demandé de l'aide aux autres membres du groupe.

Brice

Plutôt satisfait du projet dans son ensemble. Nous avons su bien se répartir le travail et nous avons bien défini notre MVP. C'était hyper intéressant de créer un projet from scratch, ce qui m'a



permis de mieux comprendre l'utilité de Babel ou Webpack par exemple. Hyperapp permet de facilement comprendre la gestion des composants. Je me sens prêt à essayer de plus gros framework tel que React ou Vue qui possèdent la même logique.

Baptiste

J'ai beaucoup aimé ce projet, car il m'a permis de travailler sur plein d'aspects différents du web, et même plus ! J'ai ainsi pu faire de la recherche d'idées, du design, de l'UX de l'intégration et de la programmation. C'était aussi l'occasion de découvrir ou mieux comprendre les outils du web (HyperApp, ES6, Parcel, Babel, ESLint, etc.). Cependant, l'intégration m'a donné un peu de fil à retordre : il était compliqué d'être fidèle à la maquette tout en utilisant les fonctions à ma disposition et en rendant notre site « responsive ». C'était aussi difficile d'appréhender le framework pour rendre un code fonctionnel, tout en étant simple et cohérent.

V. Prochaines MàJ

Nous sommes tous d'accord pour dire que nous pourrions ajouter davantage de fonctionnalités pour compléter le site, bien qu'il constitue déjà un bon appui !

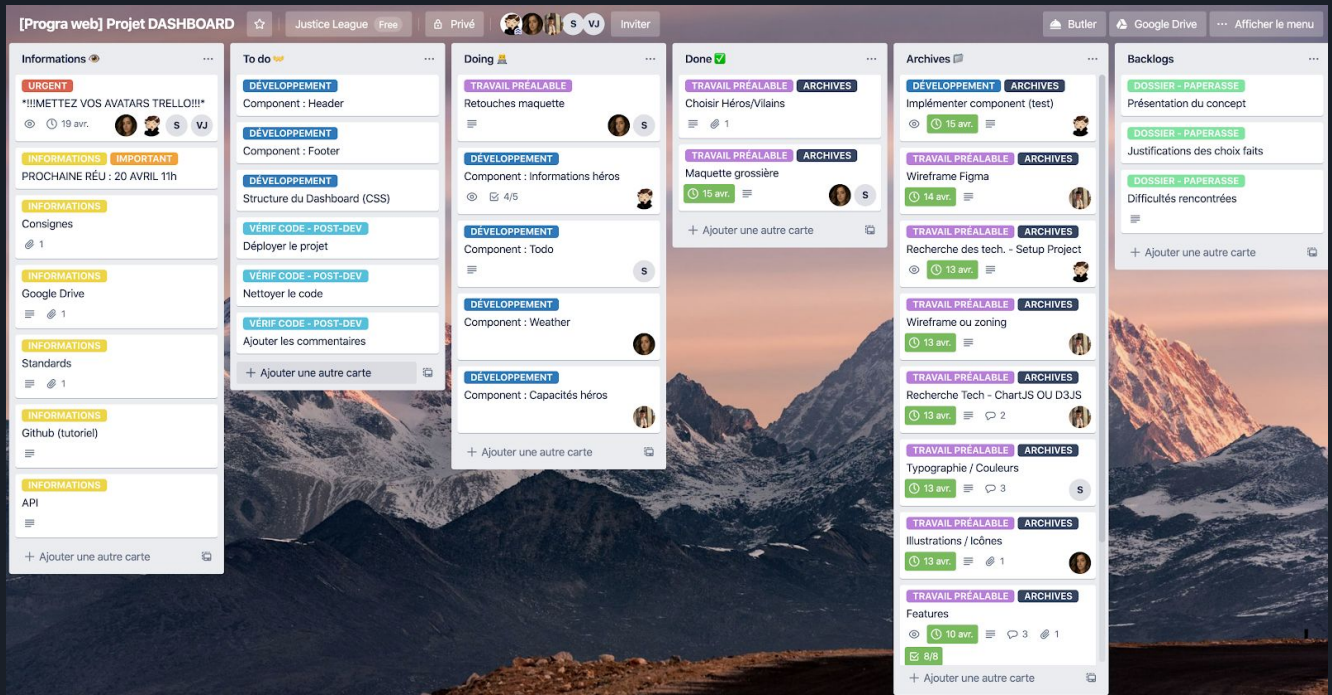
En réfléchissant aux modules que nous aurions pu intégrer à notre application, beaucoup d'idées semblaient intéressantes, mais auraient nécessité un temps de développement trop important. Ainsi, à l'avenir, on pourrait envisager d'intégrer ces modules : pourcentage de popularité du superhéros, affichage d'un ennemi aléatoire à combattre, évolution du taux de criminalité ou encore carte pour situer le lieu de naissance du superhéro. Comme notre projet constitue déjà une bonne base, ces fonctionnalités seraient simples à ajouter dans la structure du tableau.

Pour aller plus loin, on pourrait envisager de pouvoir laisser l'utilisateur choisir entre un thème clair et notre thème sombre déjà existant, changer la langue du site ou même ranger ses modules comme il le souhaite. On pourrait aussi ajouter de nouvelles animations.

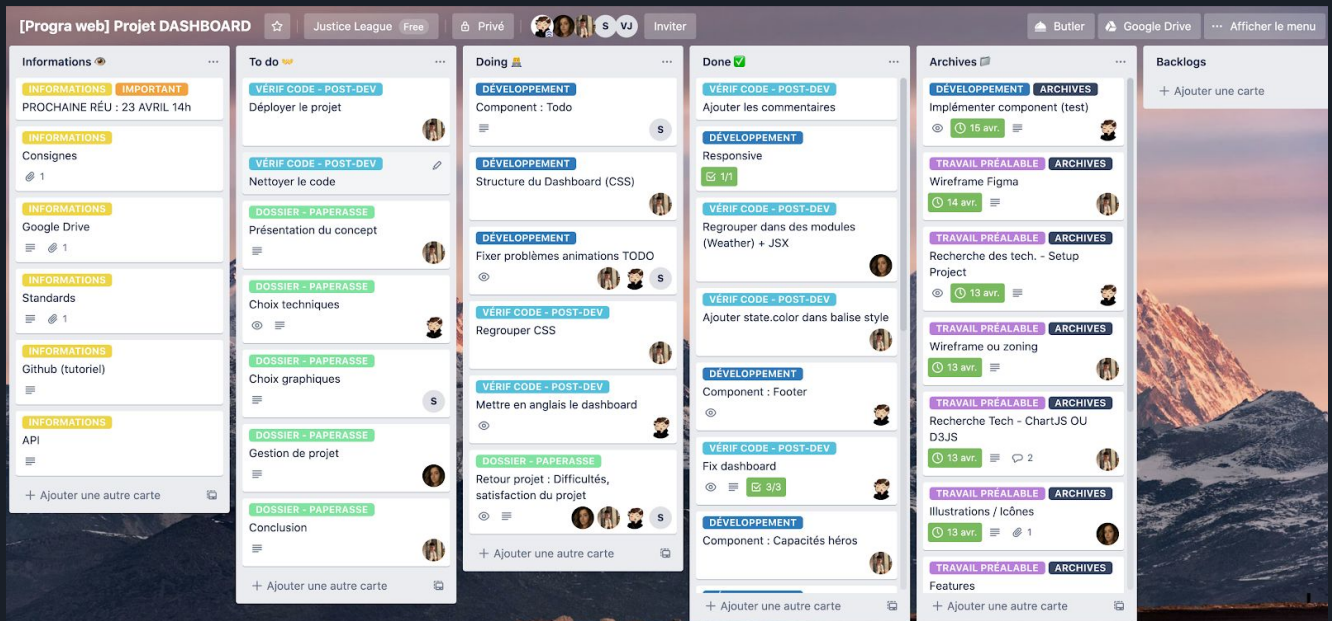


Annexes

Évolution du Trello au cours de la réalisation du projet



16-04-2020



23-04-2020

[Progra web] Projet DASHBOARD ☆ Justice League Free Privé S VJ Inviter Butler Google Drive Afficher le menu

Informations ⓘ

INFORMATIONS IMPORTANT

PROCHAINE RÉU : -----

INFORMATIONS

Consignes

1

INFORMATIONS

Google Drive

1

INFORMATIONS

Standards

1

INFORMATIONS

Github (tutoriel)

INFORMATIONS

API

+ Ajouter une autre carte

To do 📌

VÉRIF CODE - POST-DEV

Déployer le projet

DOSSIER - PAPERASSE

Présentation du concept

DOSSIER - PAPERASSE

Choix graphiques

DOSSIER - PAPERASSE

Gestion de projet

DOSSIER - PAPERASSE

Conclusion

+ Ajouter une autre carte

Doing 📌

DÉVELOPPEMENT

Structure du Dashboard (CSS)

VÉRIF CODE - POST-DEV

Nettoyer le code

DOSSIER - PAPERASSE

Choix techniques

VÉRIF CODE - POST-DEV

Regrouper CSS

DOSSIER - PAPERASSE

Retour projet : Difficultés, satisfaction du projet

+ Ajouter une autre carte

Done ✔

VÉRIF CODE - POST-DEV

Mettre en anglais le dashboard

DÉVELOPPEMENT

Composant : Todo

VÉRIF CODE - POST-DEV

Fixer problèmes animations TODO

VÉRIF CODE - POST-DEV

Ajouter les commentaires

DÉVELOPPEMENT

Responsive

VÉRIF CODE - POST-DEV

Regrouper dans des modules (Weather) + JSX

VÉRIF CODE - POST-DEV

Ajouter state.color dans balise style

DÉVELOPPEMENT

+ Ajouter une autre carte

Archives 📌

DÉVELOPPEMENT ARCHIVES

Implémenter component (test)

15 avr.

TRAVAIL PRÉALABLE ARCHIVES

Wireframe Figma

14 avr.

TRAVAIL PRÉALABLE ARCHIVES

Recherche des tech. - Setup Project

13 avr.

TRAVAIL PRÉALABLE ARCHIVES

Wireframe ou zoning

13 avr.

TRAVAIL PRÉALABLE ARCHIVES

Recherche Tech - ChartJS OU D3JS

13 avr.

TRAVAIL PRÉALABLE ARCHIVES

Illustrations / Icônes

13 avr.

TRAVAIL PRÉALABLE ARCHIVES

Features

14 avr.

+ Ajouter une autre carte

Backlogs

+ Ajouter une carte

25-04-2020

[Progra web] Projet DASHBOARD ☆ Justice League Free Privé S VJ Inviter Butler Google Drive Afficher le menu

Informations ⓘ

INFORMATIONS IMPORTANT

PROCHAINE RÉU : -----

INFORMATIONS

Consignes

1

INFORMATIONS

Google Drive

1

INFORMATIONS

Standards

1

INFORMATIONS

Github (tutoriel)

INFORMATIONS

API

+ Ajouter une autre carte

To do 📌

DOSSIER - PAPERASSE

Présentation du concept

DOSSIER - PAPERASSE

Gestion de projet

DOSSIER - PAPERASSE

Conclusion

+ Ajouter une autre carte

Doing 📌

DOSSIER - PAPERASSE

Choix techniques

DOSSIER - PAPERASSE

Choix graphiques

DOSSIER - PAPERASSE

Retour projet : Difficultés, satisfaction du projet

+ Ajouter une autre carte

Done ✔

VÉRIF CODE - POST-DEV

Déployer le projet

VÉRIF CODE - POST-DEV

Nettoyer le code

VÉRIF CODE - POST-DEV

Regrouper CSS

DÉVELOPPEMENT

Structure du Dashboard (CSS)

VÉRIF CODE - POST-DEV

Mettre en anglais le dashboard

DÉVELOPPEMENT

Composant : Todo

VÉRIF CODE - POST-DEV

Fixer problèmes animations TODO

+ Ajouter une autre carte

Archives 📌

DÉVELOPPEMENT ARCHIVES

Implémenter component (test)

15 avr.

TRAVAIL PRÉALABLE ARCHIVES

Wireframe Figma

14 avr.

TRAVAIL PRÉALABLE ARCHIVES

Recherche des tech. - Setup Project

13 avr.

TRAVAIL PRÉALABLE ARCHIVES

Wireframe ou zoning

13 avr.

TRAVAIL PRÉALABLE ARCHIVES

Recherche Tech - ChartJS OU D3JS

13 avr.

TRAVAIL PRÉALABLE ARCHIVES

Illustrations / Icônes

13 avr.

TRAVAIL PRÉALABLE ARCHIVES

+ Ajouter une autre carte

Backlogs

+ Ajouter une carte

26-04-2020

Sitographie

- Général :
 - <https://github.com/okwolf/hyperapp-logger>
 - <https://parceljs.org/>
 - <https://babeljs.io/docs/en/>
 - <https://eslint.org/docs/rules/>
 - <https://www.synbioz.com/blog/tech/un-code-js-impeccable-grace-a-eslint>
 - <https://blog.daftcode.pl/hyperapp-parcel-71823bd93f1c>
 - <https://nodesource.com/blog/the-basics-of-package-json-in-node-js-and-npm/>
- SASS :
 - <https://blog.webnet.fr/less-versus-sass/>
 - <https://sass-lang.com/guide>
 - <https://parceljs.org/scss.htm>
- Git :
 - [//mache po](https://launchschool.com/books/git/read/github)
 - <https://riptutorial.com/fr/git/example/885/ignorer-les-fichiers-et-les-repertoires-avec-un-fichier--gitignore>
 - <https://github.com/github/gitignore/blob/master/Node.gitignore>
- Librairies JS et CSS :
 - <http://flexboxgrid.com/>
 - <https://www.chartjs.org/docs/master/charts/radar.html>
 - <https://blog.bitsrc.io/customizing-chart-js-in-react-2199fa81530a>
 - <https://www.d3-graph-gallery.com/spider.html>
 - <https://greensock.com/docs/v2/NPMUsage>



Projet réalisé dans le cadre du
cours de web du semestre 2 à l'IMAC