Dashboard project

Deadline front: 29 Novembre
Deadline finale: 10 Décembre

Les dashboards ou tableaux de bord permettent d'avoir une vision claire et précise sur l'état d'un business, d'un service…

Aujourd'hui énormément d'entreprises utilise des dashboards. La demande est tellement forte, que des startups créé leur service de dashboard sur mesure à l'instar de Forest ou Toucan Toco. Ces entreprises ont aujourd'hui des clients qui appartiennent au CAC40 et réussissent à lever de l'argent (3M € pour forest), preuve de l'engouement de cet outil formidable qu'est le dashboard.

Ces startup insistent sur le mot data visualisation qui englobe des techniques utilisées dans les tableaux de bord. Ces techniques permettent de faciliter grandement la lecture de données hétérogènes et d'en comprendre le sens très rapidement. L'objectif est de faciliter la vie des collaborateurs et les aider à comprendre la donnée.

Objectifs:

L'objectif du projet est de créer votre propre dashboard. Ce projet est séparé en deux parties bien distinctes.

Première Partie:

La première partie partie concerne le front-end, la partie visible par l'utilisateur:

Le front-end sera réalisé grâce à la librairie React qui nous permettra de créer des composants visuels appelés widget. Chacun des ces widgets aura un rôle spécifique.

L'objectif de cette première partie est donc de réaliser au moins **6 widgets différents** (inspiration dans les liens en bas de page) libre à vous d'en faire plus (Ce sera pris en compte dans la notation uniquement si les éléments obligatoires sont réalisés).
Au moins un widget doit utiliser la librairie recharts (ou une librairie similaire de création de graphique en react), par exemple un widget qui affiche un graphique linéaire. Chaque widgets devra être utilisé et testé dans l'environnement storybook initialisé dans le repo du projet.

En accord avec la philosophie de React, chaque widget devra être **réutilisable.**

Votre dashboard comportera deux pages. La première page sera l'accueil de votre site, elle comportera au moins les six différents widgets sur lesquels vous avez travaillé.
La seconde page sera une page administrative qui comportera dans un premier temps un formulaire simple qui vous permettra d'injecter de la donnée dans votre API.

Avant de vous lancer dans le code tête baissé, déterminez avec votre binôme les différentes tâches à effectuer et utiliser un outil de type airtable ou trello pour pouvoir suivre votre avancement.
Vous devrez donc réfléchir à un découpage en petites tâches de votre projet, assigner les tâches aux bonnes personnes et leur donner une date de fin. Vous mettrez le lien public de l'outil que vous avez sélectionné dans le readme de votre repo.

Prenez le temps d'effectuer un mock-up de votre site sur un outil comme mockflow ou figma. Ce mock up vous permettra de réfléchir à la fois à l'expérience utilisateur (UX) et à l'interface utilisateur (UI). Vous mettrez l'image extraite de votre mockup dans le repo.

Objectifs obligatoires partie 1:

- ❏ Utiliser des outils de planification/gestion de projet
- ❏ Créer un wireframe de votre site
- ❏ Utiliser storybook pour tester et documenter vos widgets.
- ❏ Créer 6 widgets différents dont au moins un avec une librairie de création de graphiques
- ❏ Le dashboard doit être responsive et être affichable sur mobile, tablette et PC.

Technos:

Front: HTML/CSS, React, Bootstrap

Inspiration widgets design:

https://www.uplabs.com/search?q=dashboard
https://dribbble.com/search?q=dashboard
https://artists.spotify.com/blog/introducing-spotify-fan-insights

Deuxième partie:

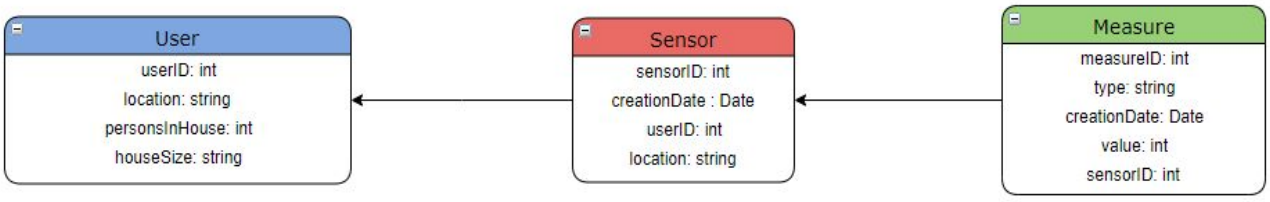
La deuxième partie concerne le back-end de votre projet.

Mise en situation

Félicitations vous avez créé un objet connecté destiné au grand public, vous avez quelques exemplaires en circulation et ils remontent déjà de la donnée ! Ces objets permettent entre autre de capturer de la donnée sur l'environnement qui les entourent. Il peuvent relever la température, l'humidité et la pollution de l'air.

Lors de notre dernière rencontre nous avons réfléchi tous ensemble à une structure de donnée pour pouvoir accéder et modifier la donnée de la meilleure des manières. Après avoir synthétisé tous nos échanges, je vous transmets la première version de cette structure de données. Elle vous aidera surement à travailler sur la premiere version de votre API qui vous permettre de consommer cette donnée en toute sécurité.

Base de donnée:



User:

- userID: Identifiant de l'utilisateur
- Country: pays de l'utilisateur
- personsInHouse: nombre de personnes dans la maison de l'utilisateur
- houseSize: taille de la maison (small, medium, big)

Sensor:

- sensorID: Identifiant du capteur
- creationDate: date de mise en place du capteur
- userID: identifiant de l'utilisateur à qui appartient le capteur
- location: endroit où est situé le capteur (kitchen, livingRoom, bedroom, bathroom, entrance)

Measure:

- measureID : identifiant de la mesure
- Type: type de mesure (humidity, temperature, airPollution)
- creationDate : date de creation de la mesure
- Value: valeur chiffrée de la mesure
- sensorID: identifiant du sensor à qui appartient cette mesure

API

Pour chacun des éléments de la base (User, Sensor, Measure) l'utilisateur de votre API doit être capable d'effectuer au minimum les quatres opérations CRUD (création, recherche, suppression, mise à jour). Sur ces éléments de la base vous devrez effectuer d'autres opérations qui vont permettront de récupérer des éléments statistiques par exemple :

- Le nombre total de capteurs en circulation
- Le nombre total d'utilisateurs
- La moyenne de température sur tout le parc des capteurs
- Les maximales/minimales des 3 derniers mois par pays
- ...

Cette liste est non exhaustive, à vous de trouver des données statistiques intéressantes à afficher dans votre dashboard. Ces éléments statistiques seront consommés par votre application de dashboard et vous aurez à les afficher dans les composants que vous avez créé dans la première partie. Il est bien sur possible de créer de nouveaux widgets afin d'avoir une analyse plus percutante des données.

Administration

Dans votre front end, créer une route "/admin" (à l'aide de react-router, comme vu en cours). Dans cette vue, vous aurez un formulaire qui permet de rajouter un utilisateur dans votre base et une liste de tous les utilisateurs présents dans votre base.

Objectifs obligatoires partie 2:

- ❏ Créer une API à l'aide de express qui permettra de consommer les données
- ❏ Créer une route dans votre frontend qui permettra une administration simple des utilisateurs.

Technos:

Back: Node.js, Express, mongodb, mongoose

Bonus

Vous pouvez si vous souhaitez aller plus loin, mettre en place l'une ou toutes ces fonctionnalités:

- Pouvoir ajouter des capteurs à un utilisateur
- Pouvoir modifier un utilisateur déjà existant
- Pouvoir supprimer un utilisateur (supprimer un utilisateur implique de supprimer aussi ses capteurs et les mesures de ces capteurs)