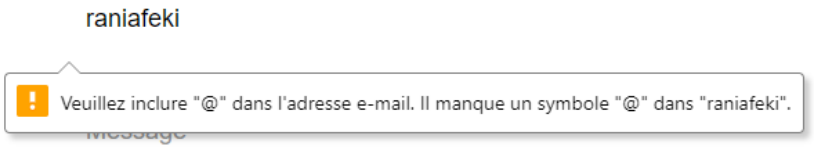
Rendu Individuel
- Rania Fekih -

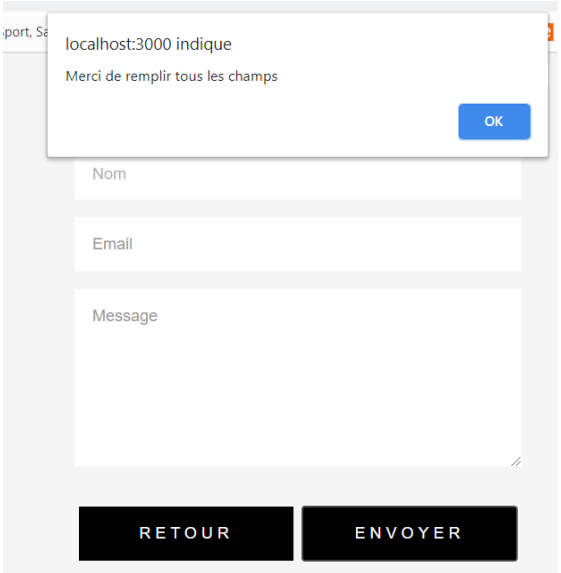
Identifiant github :
raniafe / rania.fekih@etu.unice.fr

Tâches effectuées :

- Développer le composant graph qui utilise le composant chart pour afficher les Line chart de nombre de cas et de nombre de santé mental par région :
- Intégrer le changement du thème dans le composant graph et chart
- Modifier les composants List et Map pour ajouter la redirection vers le graph de la région sélectionnée
- Modifier App.js pour ajouter la route vers le graph + regionId.
  → Branche : #27-Front-End-Graph
- Formulaire de contact, sauvegarder le message dans la BD
- Intégrer le changement du thème dans le composant contact Form
- Ajouter les boutons de redirection vers le contact et le bouton back dans les composants RegionStats et Graph.
- Vérifier l'adresse mail :



- Vérifier que les champs ne sont pas vides



→ Branche : #35-contact-form

Stratégie employée pour la gestion des versions avec Git

Pour assurer que nous sommes tous au courant de l'avancement du projet on fait chaque semaine deux points pour mettre à jour les feature et tâches dans le tableau du projet dans la colonne ToDo et on assigne les tâches à chacun d'entre nous et à partir de là on suit ces règles :

- On transforme le ticket en issue et on assigne les labels correspondant
- On crée une branche pour ce feature
- Au cours du développement on merge toujours la branche main pour s'assurer qu'on est à jour des modifications de tout le monde.
- A la fin du développement on merge la branche main, on s'assure qu'il n' y a pas de conflit et on crée un Pull request. A la création du pull request on assigne une personne et on lui notifie sur slack pour vérifier le code et on discute les différents points.
- S'il y a des modifications à faire on les fait et on notifie la personne concernée
- Quand tout est réglé le code reviewer accepte le merge request

Solutions choisies

- **Préciser pourquoi**
    - J'ai choisi google chart comme libraire pour afficher nos données sous un format de line chart parce que c'est plus facile à manipuler ses attributs et ses options. En revanche, le plugin "Chart" permet d'utiliser la bibliothèque Google Charts à tout moment sans avoir à télécharger une bibliothèque autonome: il télécharge uniquement la bibliothèque lors de la demande de création du premier graphique.
    - Pour la vérification de l'email j'ai trouvé que c'est suffisant d'utiliser un input de type email qui affiche un message si l'adresse rentrée n'est pas vraiment une adresse mail
    - Pour l'envoie des requêtes j'ai travaillé avec fetch et axios pour ces raisons :
        - Pour les requêtes de type get j'ai trouvé fetch une librairie qui fait l'affaire.
        - Pour les requêtes de type post j'ai utilisé axios vu qu'il convertit automatiquement le body en json et il y a aucun traitement à faire.
    - Pour la redirection vers les composants graph et Contact, il y avait deux possibilités Link ou history.push(). Avec Link le

composant s’affiche dans la même page ce qui n’est pas ce qu’on souhaite faire. Donc avec l’aide de mes collègues j’ai modifié Link avec history.push()

Difficultés rencontrées

- Au début nous voulons afficher le nombre de cas et les nombres de personnes qui ont de troubles de santé mentale dans le même graph qui n’était pas facile à réaliser parce qu'il nous fait un tableau qui contient les deux avec un même échelle j’ai essayé de faire ce traitement dans le backend pour donner un seul tableau qui contient les deux avec les mêmes dates et donc j’ai décidé d’assembler les chiffre par moi et non plus par jour
- Une autre difficulté que j’ai rencontré c’est au niveau de redirection comme j’ai cité précédemment la fonctionnalité link nous affiche le composant dans la même page et il ne fait pas une redirection donc pour faire ça j’ai utilisé history.push()
- Les autres difficultés étaient plutôt coté backend vu que j’ai travaillé plus sur le back que sur le front

Temps de développement / tâche

- Les deux premières semaines j’étais plutôt concentré sur le back vu qu’on peut rien afficher sans les données traitées et stockées dans la BD.
- La 3ème semaine j’ai fait les graphs
- Cette semaine j’ai finalisé quelques éléments dans le graph (thème, redirection) et j’ai fait le contact form + le rapport.

Code

- Dans le composant Graph au final nous avons décidé d'afficher un graph par données vu que ce n’était pas vraiment intuitives le fait de mettre les deux dans le même graph. Donc pour éviter que le code soit redondant, dans le composant graph je fetch et je prépare les données après je passe ces données vers le composant chart pour le display.
  Je suis satisfaite de ce composant mais je pense qu’on peut l’améliorer en ajoutant d'autres formes de graph qui peuvent être plus convaincantes. Par exemple histogramme par jour pour le nombre de cas .
- Je suis fière de mon composant contact parce qu’il est clean et “straight to the point”. C’est lisible et facile à comprendre et ça a été développé en utilisant les hooks.