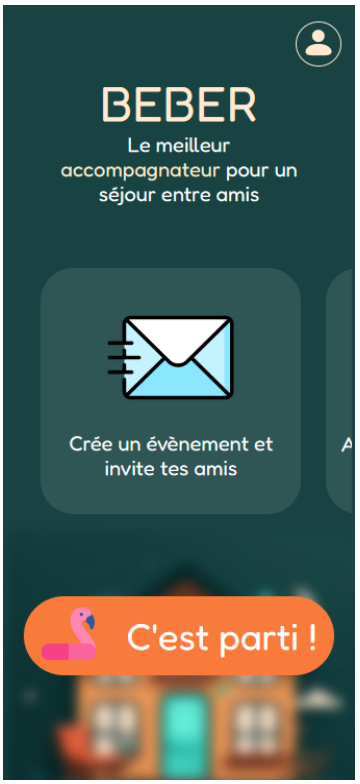
Béber : Note sur l’application

Contexte

Béber est né d’une expérience personnelle. Nous organisions un séjour entre amis venant des 4 coins de la France et de la Belgique. Arrive le moment de choisir un lieu satisfaisant pour tous, en prenant en compte les situations géographiques et financières de chacun. Et c’est le début des drive, des excel et des mails dans tous les sens, avant de se décider sans trop de concertation pour un lieu qui s'avèrera être une arnaque.

Cette situation aurait pu être évitée avec une organisation plus efficace et collective. L’idée d’une application accompagnant le groupe pour une décision plus éclairée commença à germer. J’ai alors commencé à échanger avec d’autres personnes pour en apprendre plus sur les éléments pris en compte dans le choix d’un lieu pour un séjour. Le site de réservation, le prix, la localisation, le nombre de couchages, les photos, etc.. sont autant d’indicateurs permettant une base de débat décisionnel pertinente. L’objectif d’avoir une application regroupant toutes ces informations et permettant ensuite de décider en groupe était alors fixé.



Page d’accueil de Béber

Public cible

Étant avant tout un projet réalisé pour mes amis et moi, le public cible initial n'était pas très large. Mais à force de discussions, je me suis rendu compte que nous rencontrions les mêmes problématiques que beaucoup de personnes. Celles-ci étaient issues de différentes situations socio-économiques allant du couple qui réfléchit aux différentes options pour un week-end low-cost en amoureux, à la vingtaine d'amis aisés débattant de quel château est le plus intéressant à louer, en passant par la famille qui s'intéresse aux activités proposées aux alentours du lieu. L'application tend à s'adresser à des profils très larges.

Béber doit donc être un outil intuitif, rapide et simple à prendre en main. Il doit être clair et chaleureux, mettre en valeur les informations importantes pour accompagner la décision du groupe.

Organisation fonctionnelle

Avant de coder et de déployer un premier outil testable par les potentiels utilisateurs, il fallait cibler un ensemble de fonctions essentielles au squelette de l'application. Il y avait alors deux axes principaux à prendre en compte :

- Une dimension fonctionnelle individuelle : la possibilité pour un utilisateur de créer un compte, créer un événement, inviter d'autres utilisateurs à l'évènement, ajouter une annonce de logement, noter les logements proposés
- Une dimension fonctionnelle collective : la possibilité pour les utilisateurs participant à un événement de voir les moyennes de chaque lieu, voir le prix par personne en fonction du nombre de participants

La première version de Béber testable devait nécessairement avoir ces éléments. Les autres fonctionnalités telles que la distance de chaque logement à la position de chaque participant, ou la possibilité d'écrire des commentaires pour chaque logement proposé viendront par la suite.

Concernant l'ajout d'une annonce, l'utilisateur renseigne un lien qu'il a vu sur airbnb ou booking par exemple, et l'application devra aller chercher les informations à partir de ce lien. Pour cette première version, Béber ne concerne que les annonces Airbnb.

Formulaire d'ajout d'une annonce

Choix techniques

L'application est organisée en trois blocs : front, back, et database. Le front est réalisé en React/Redux, en utilisant le préprocesseur Sass pour le CSS. Pour cette première version, aucune librairie CSS n'a été utilisée. Le back est réalisé en Node.JS/Express. Pour la database, j'ai voulu écrire les requêtes SQL à la main et j'ai préféré mysql à TypeORM.

Pour la récupération des données d'un lien d'annonce, cette première version est réalisée à l'aide de techniques de scraping, l'API d'Airbnb n'étant pas accessible gratuitement. Après différents tests avec cheerio, axios et selenium, c'est finalement avec puppeteer que les informations de l'annonce sont récupérées, enregistrées et rendues accessibles aux utilisateurs.

UI/UX

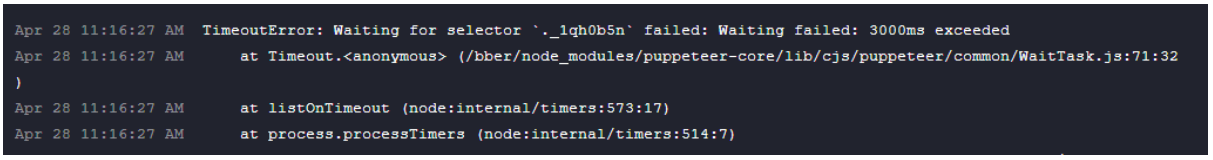
Un UX designer m'a proposé une maquette Figma de la page d'accueil à partir de croquis que je lui avais fournis. Je me suis basé sur cette page d'accueil pour définir un premier brouillon global du design de l'application.

Première croquis de page d'accueil réalisée à partir de Balsamiq

L’important étant d’avoir une navigation fluide et intuitive, en conservant un fil d'ariane pour aider l’utilisateur à se situer dans l’application.
Par la suite, le design global est amené à évoluer, notamment par l’utilisation de librairies CSS/Javascript dédiées. Le but à terme étant d’avoir une application chaleureuse et agréable d’utilisation, rapide et efficace.

Déploiement

Je me suis formé sur Docker afin de pouvoir configurer mon déploiement sur le serveur qu’un Devops me proposait. Initialement déployé avec trois containers(image du front, du back, et d’une database MariaDB), l’application est maintenant hébergée pour des raisons économiques via des service tiers : Render pour le back, Vercel pour le front, et Tidb pour la database. Toutes les erreurs sont récupérées, affichées et enregistrées dans les logs de l’application (via un middleware d’erreur affecté à toutes les routes). Cela permet d’une part d’avoir une première version testable stable, et d’autre part de pouvoir consulter les situations amenant aux erreurs afin de les corriger plus efficacement.



Erreur récupérée dans les logs

Cependant, ces solutions sont temporaires, car la gratuité à un coût : la vitesse des serveurs. Béber nécessite notamment via l’utilisation de puppeteer de ressources rapides, et la version testable est donc particulièrement longue à ajouter une annonce.
Par la suite, l’objectif sera de faire reposer l’application sur des infrastructures plus robustes, pour une utilisation plus fluide et rapide.

Sécurité

Le premier élément de sécurité majeur de Béber est l’authentification. Lorsque l’utilisateur se connecte, il reçoit un token valable un certain temps. Si le token expire, Redux déconnecte l’utilisateur de l’application front et il est invité à se reconnecter. Toute requête doit être authentifiée avant d’accéder aux controllers.
Les données utilisateurs sont vérifiées en front avant de pouvoir valider un formulaire, et en back avant d’être traitées. Le mot de passe est hashé et n’est jamais visible. L’utilisation d’Helmet pour le back a également été mise en place. Les dépendances sont régulièrement mises à jour pour bénéficier du meilleur niveau de sécurité de chacun des éléments qui composent l’application.
Concernant les hébergeurs actuels de l’application, des mots de passe sécurisés ont été utilisés pour chaque service.

CGU/RGPD

Ces éléments ne sont pas encore établis au sein de l'application pour sa version testable. Ils seront cependant rédigés pour la prochaine version.

Futur de l'application

L'objectif est d'avoir à terme, un outil optimisé pour l'organisation d'un séjour entre amis. Prendre en compte le prix de trajet de chacun, mettre en place des covoiturages, donner la possibilité d'organiser une bourse pour permettre aux plus aisés d'aider ceux qui ont moins de moyen à participer, et toutes les idées qui émergeront des discussions avec les utilisateurs seront autant d'axe de travail pour l'évolution de Béber.

N'ayant alors pas de cahier des charges fixe, j'ai choisi de travailler de manière agile pour la suite du projet. J'ai mis en place un tableau Kanban à l'aide de Jira où je définis mes tickets en fonction des bugs constatés, des retours utilisateurs, et des nouvelles fonctionnalités à ajouter. Un développeur m'ayant demandé à rejoindre le projet, ce tableau me permet également d'avoir une vision sur le niveau de difficulté des tâches et la manière de déléguer ou non certains tickets. Plusieurs branches sont établies sur Github pour que chacun puisse avancer sur des points précis.

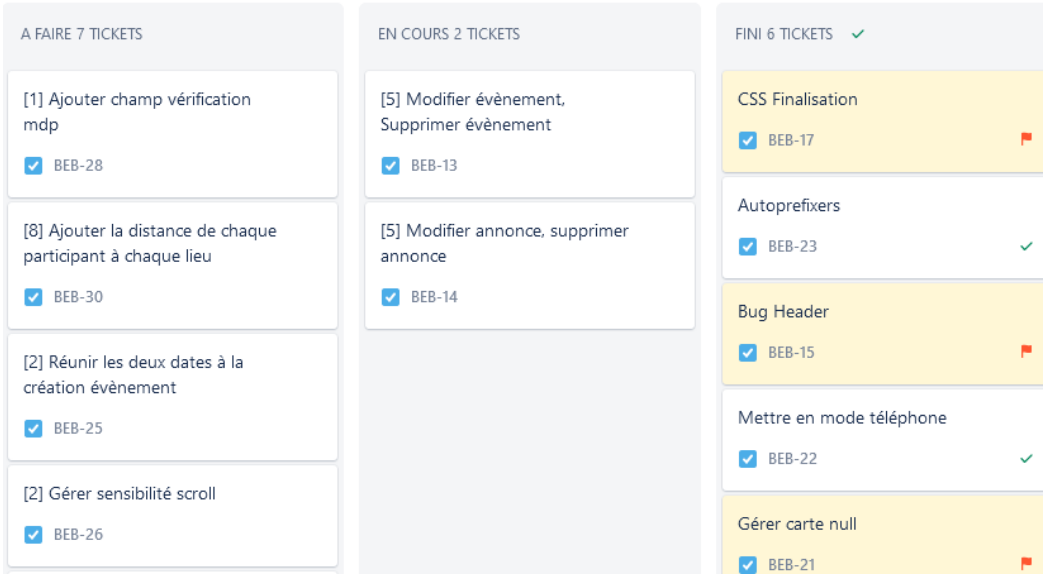


Tableau Kanban du sprint à venir

Dans l'immédiat, la prochaine étape sera de déployer une application mobile et une application web responsive prenant en compte tous les retours utilisateurs. L'ajout des commentaires et la mise en place du système de localisation sont les fonctionnalités prioritaires qui seront développées très prochainement.

Conclusion

Béber est à l'heure actuelle une esquisse de ce qu'il tend à devenir : un accompagnateur d'organisation de séjour. La phase de tests en cours permettra de repenser l'application en fonction des retours utilisateurs, et de savoir quelles sont les directions pertinentes pour la suite.

La dimension économique devra être sérieusement questionnée. Le scraping en tant que solution gratuite n'est pas viable et un investissement pour un accès direct aux API des hébergeurs d'annonce serait un gain de ressources non négligeable. Des services d'hébergement plus puissants représentent également un coût. Il sera nécessaire d'étudier quels seront les coûts nécessaires à la mise en place d'une application complète rapide et efficace.