

# Gestionnaire de voyage

## I- Concept

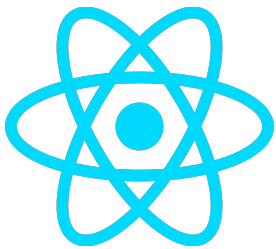
Le but de ce projet est de réaliser un gestionnaire de voyage qui permet à l'utilisateur de pouvoir organiser ses vacances en 3 clics.

Une fois inscrit sur le site web du service, l'utilisateur devra répondre un à questionnaire pour que l'algorithme connaisse ces goûts.

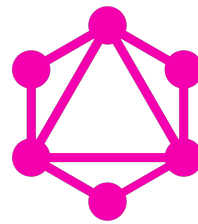
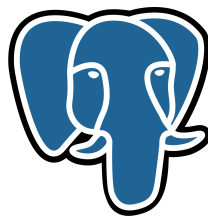
Suite à cela, l'utilisateur aura la possibilité de réserver son premier voyage. Ce dernier n'aura qu'à renseigner sa destination ainsi que les dates auxquelles il souhaite partir en vacances, ensuite l'algorithme se charge d'organiser vos journées en fonctions de vos goûts.

## II- Stack technique

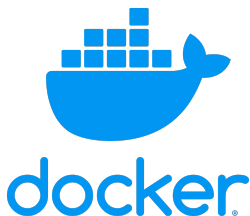
### - Frontend :



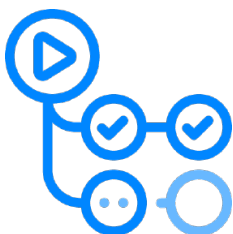
### - Backend :



### - Devops :



### - CI-CD :



- Tests :



J'ai choisi de travailler avec une stack en full TypeScript que j'ai divisé en 5 parties distinctes.

Pour le frontend, j'ai choisi de travailler avec React en TypeScript pour avoir une application facile à coder, la manière d'écrire du code React et selon moi assez proche de la manière dont est écrit le vanilla JavaScript. La librairie React possède une grosse communauté de développeur et une bonne documentation, ce qui permet d'accueillir facilement plus de développeurs sur le projet, la marche de progression pour apprendre le fonctionnement de la librairie reste abordable. De plus, la librairie nous laisse une certaine liberté au niveau de l'architecture du projet et de l'utilisation de certain outil. Pour ce projet, je vais utiliser la librairie sass pour styliser l'interface utilisateur de l'application. Enfin, cela va me permettre de pouvoir avec une version mobile du projet grâce à React Native. Cette transition va pouvoir se faire assez rapidement puisqu'il y a déjà tout en base de code commune entre React et React Native.

Pour ce projet, j'ai choisi aussi bien côté frontend que backend d'utiliser TypeScript. Cet outil permet de combler les lacunes du vanilla JavaScript notamment au niveau du typage et d'avoir une meilleure compréhension du code. TypeScript possède une grosse communauté et une très bonne documentation. De plus cet outil est maintenant utilisé dans plus de 70% des projets en JavaScript pour son typage plus strict. Un typage strict et une bonne organisation sont selon moi deux points très important dans un projet pour garantir sa scalabilité.

Pour styliser ce projet j'ai choisi d'utiliser la librairie sass. Elle possède une bonne documentation avec une grosse communauté de développeurs. Sass offre plus de fonctionnalités par rapport au vanilla CSS, ce qui permet d'avoir un code plus propre, plus organisé et plus réutilisable.

Pour m'aider dans l'UI de ce projet, j'ai choisi d'utiliser les design tokens, c'est une technologie plutôt récente que j'utilise tout les jours depuis plus de 6 mois, elle offre une souplesse au niveau de style de notre application notamment au niveau de la charte graphique. Un changement effectué sur Figma est directement répercuté sur notre projet. Avec les designs tokens, il est possible de créer une véritable design API pour garantir la scalabilité du projet.

Pour la partie backend du projet, j'ai choisi d'utiliser Nest.js, un framework Node.js écrit 100% en TypeScript qui offre une bonne architecture de projet. Le framework possède une grosse communauté de développeurs avec une bonne documentation.

Pour la base de données j'ai choisi de travailler avec PostgreSQL. Cette technologie est idéale pour mon projet qui ne nécessite pas d'avoir énormément de requêtes. Cet outil possède une grosse communauté de développeurs avec une bonne documentation. L'utilisation de cette base de donnée avec Nest.js est facilitée grâce à l'utilisation de TypeORM, un ORM qui permet de créer des tables et colonnes SQL facilement et rapidement. L'utilisation de cet ORM nous permet d'automatiser certaines tâches notamment l'écriture en base de donnée.

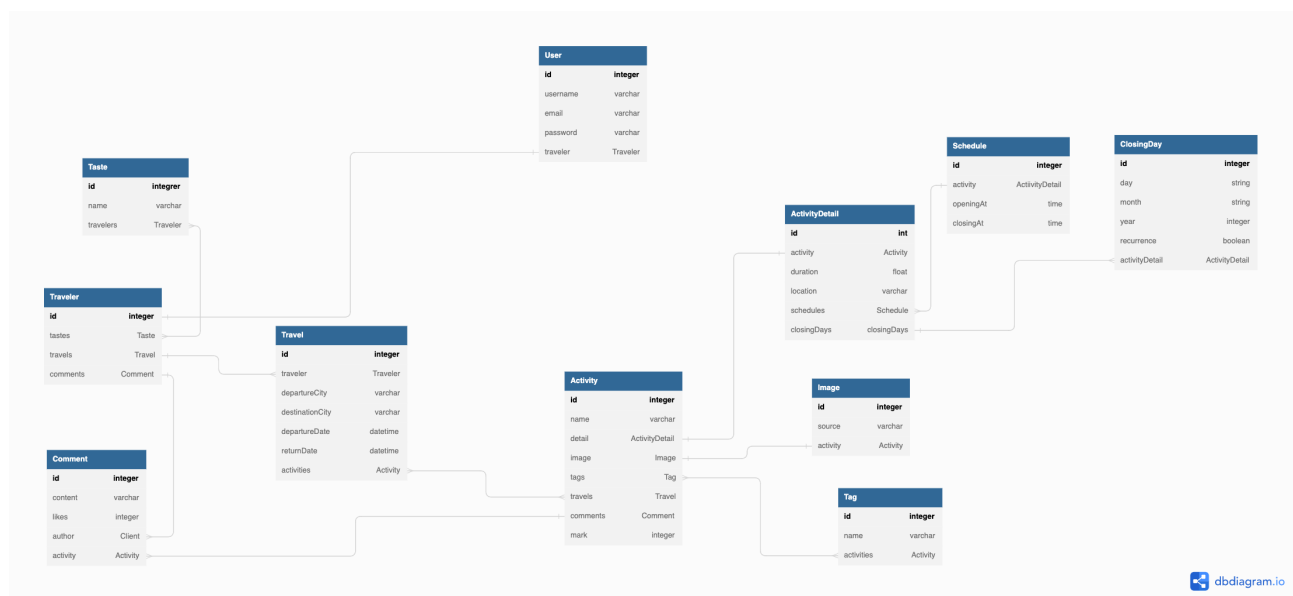
Pour l'API de ce projet, j'ai choisi d'utiliser une API GraphQL qui permet d'offrir un typage strict des données. De plus, cette technologie a une grosse communauté de développeurs et une bonne documentation. Le fait d'utiliser cet outil permet d'appeler que ce dont nous avons besoin, pour pouvoir consommer ou envoyer que les données que nous voulons à un instant T. L'utilisation de cette technologie est selon moi plus écologique, puisque nous recevons des requêtes avec uniquement les choses que nous avons demandées.

Pour le devops, j'ai décidé d'utiliser Docker, un outil très utilisé par de nombreux développeurs, il possède une bonne documentation. Dans mon cas, je vais me servir de cette technologie dans la phase de développement de mon application pour que celui-ci soit plus rapide. Docker offre un gros avantage est qu'il y a besoin que de fichiers docker-compose et Dockerfile, ensuite Docker s'occupe du reste avec son système de conteneurs. La configuration est donc facilement partageable.

Pour le CI-CD j'ai décidé d'utiliser les Github actions pour avoir un déploiement rapide et automatisé. Mais aussi avoir du code propre au moment du merge des différentes branches.

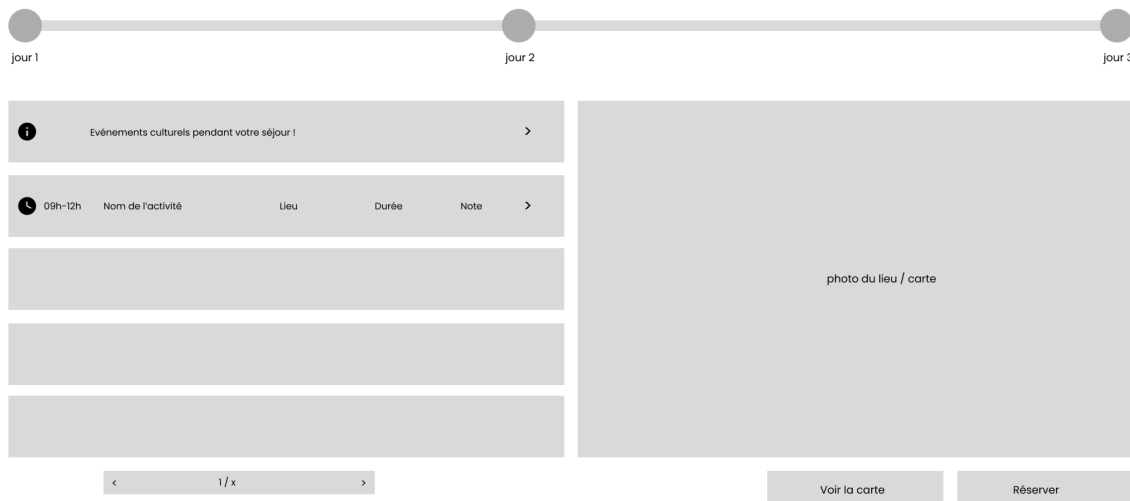
Pour les tests, j'ai choisi d'utiliser Jest, une librairie de testing qui permet de tester son code JavaScript rapidement et simplement. Cet outil possède une grande communauté de développeur avec bonne documentation.

### III - Schéma de base de données



## IV- Wireframes & specs fonctionnels

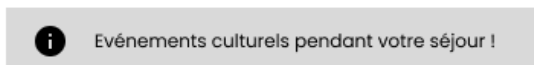
### Mon voyage



9:41



### Mon voyage



Jour 1

09h-12h	Nom de l'activité	
durée	Lieu	Note



La carte

Réserver

Sur cette page, vous pouvez voir quelles activités l'algorithme a sélectionné pour votre séjour. Sur Mobile, l'accès au planning des différentes journées du séjour de l'utilisateur se fera grâce à un menu en accordéon. Sur Desktop la navigation pour connaître le planning des différentes journées de l'utilisateur se fera via un fil d'ariane en haut de la page.

Si il y a des événements locaux (fêtes, manifestation, marathon, ...) pendant le séjour de l'utilisateur, une card événement sera affichée. En cliquant sur la card "événements", l'utilisateur sera redirigé vers une nouvelle page avec plus d'information sur les événements locaux prévus durant le séjour de l'utilisateur.

En cliquant sur la card d'une activité, l'utilisateur sera redirigé vers une nouvelle page avec plus d'informations concernant l'activité (photos, description, commentaires, ...).

En cliquant sur le bouton “carte” l'utilisateur sera redirigé vers une nouvelle page avec une map qui lui indiquera les emplacements des différentes activités pendant le séjour sous forme de points de localisation.

En cliquant sur le bouton “réserver”, l'utilisateur sera redirigé vers une nouvelle page avec le récapitulatif de sa réservation (destination, dates, lieu, prix).