

Projet Panza

Voici le contexte de la mission :

Pour le besoin d'une troupe de théâtre bordelaise d'une quinzaine de personnes, vous devez concevoir une application. La partie front sera gérée en interne par les gens de la troupe, donc à vous de concevoir surtout le backend ! Après un début de structure HTML avec JS serait un plus pour la troupe avant d'intégrer tout ça.

Cette application permet de gérer tous les **membres** de la troupe mais surtout tous les **ateliers** et les **spectacles** des comédiens ! Elle sera vitale pour la troupe !

Il y aura une première page pour **se connecter** au site, le membre devra saisir son login et password.

Une fois connecté, il arrivera sur un tableau de bord avec : le calendrier du mois en cours, les sélections des prochains spectacles et les dates des prochains ateliers avec leurs thèmes, etc.

Il y a aura donc une page de gestion des sélections, où un membre lambda **peut valider ou non sa présence et la préférence de son rôle**, il peut également **mettre un petit commentaire** pour prévenir d'un retard éventuellement ou autre.

Car sur chaque **spectacle** il y a besoin d'avoir des **comédiens** mais aussi des **bénévoles** pour **prendre des photos ou préparer à manger**.

Un spectacle est caractérisé par **un nom, un lieu, une date, une heure, une description, un nombre de comédiens, une catégorie (comme Shakespeare, Molière, Théâtre de Boulevard, Contemporain, ...) etc et une sélection !**

Un membre lui possède **un nom, prénom, mail, photo**. Le membre pourra **modifier ses informations** si nécessaire.

Parmi les **membres** il y aura quelques personnes qui seront **admin** comme le président ou le vice-président de la troupe de théâtre.

Ils seront en charge de la **création des spectacles, des membres et des ateliers**.

Une fois un spectacle lancé sur la plateforme, alors tous les membres **reçoivent un mail pour postuler**. Et J-14 avant le spectacle, **les admins** sont **en charge de choisir** parmi les personnes

présentes qui aura le privilège de jouer le spectacle ! Une fois la sélection validée, alors tous les membres reçoivent la sélection par mail !

Pour aider à faire cette sélection, les admins possèdent plusieurs outils comme : des statistiques de réponses des membres, savoir combien de fois ils ont été sélectionné et le nombre de tentatives etc... combien de fois ils ont eu le rôle de comédien ou de bénévoles etc...

Bien évidemment les admins auront à disposition l'historique de toutes les sélections de l'année ! Sachant que la troupe espère jouer 1 à 2 fois par mois dans l'année au minimum !

De plus, les admin souhaitent via le site, éditer les ateliers afin de les planifier et savoir le taux de présence des membres. Une alerte sera par exemple envoyée la veille ou l'avant veille de l'atelier.

L'atelier possède lui aussi un nom avec un coach, une date, un lieu et une description ou remarques en plus si besoin !

Bien évidemment la présence en atelier sera un facteur pour être plus sélectionnable qu'une autre personne !

Votre mission sera de créer le backend par rapport à ce besoin !

Pour cela, il faut y aller étape par étape !

Ne pas hésiter à venir me voir afin d'avoir plus d'informations ou pour confirmer certaines hypothèses, comme dans une relation client/prestataire.

Également dans la même idée, vous pouvez proposer des idées aux clients et/ou confronter les besoins avec lui.

Voici les étapes à suivre :

1. Bien lire le texte et essayer d'entourer les entités et souligner les fonctionnalités
2. Faire une todolist et s'organiser
3. Débuter une analyse UML
 - a. Faire un schéma de UC pour bien définir le besoin
 - b. Faire un diagramme de classe pour définir les classes
 - c. Si besoin des wireframes rapide pour bien comprendre le besoin
4. Débuter l'analyse de l'API en prévoyant les endpoints et les retours possibles
5. Créer un repo github pour le projet
6. Commencer à développer l'API en commençant par les Schéma, si besoin, voici des ressources en plus pour gérer les relations entre schéma :
 - a. <https://dev.to/alexmercedcoder/mongodb-relationships-using-mongoose-in-nodejs-54cc>
 - b. <https://gabrielromanato.name/nodejs-model-relationships-in-mongoose>
7. Créer les différents endpoints et tester avec des données factices via Postman
8. Pour les prochaines fois
 - a. Gestion de l'authentification

- b. Gestion des fichiers
- c. Passer de mongoose à Sequelize (ORM SQL)