Projet : McCrew

1. Analyse des besoins

Introduction

Dans le cadre du cours de développement orienté web, nous allons réaliser un site web qui répond aux besoins d’un client. Nous avons pensé à travailler avec l’équipe du McDonald’s de Nivelles, car la communication au sein de l’équipe n’est pas bonne. Les crew’s ne sont jamais mis au courant des nouveautés, mais aussi des changements. Il est aussi ennuyeux de voir quels sont les horaires de la semaine, car il faut se déplacer jusqu’au restaurant à chaque fois.

L’équipe a accepté de collaborer bénévolement dans le but de créer un site web interne à l’entreprise, pour aider la communication entre les travailleurs (ou Crew’s).

Description globale

Le site web aura différents comptes utilisateurs comprenant des comptes « Managers » et des comptes « Crew’s », ils auront différents droits au niveau du site web. Ces comptes seront tous créés par les managers pour empêcher quelqu’un qui ne travaille pas au McDonald’s de se créer un compte.

Sur la première page se trouveront les annonces pour que les travailleurs ne les ratent pas en arrivant sur le site. La page sera composée de différentes annonces toutes faites par les managers, quand une annonce sera faite, un mail sera envoyé à tous les travailleurs pour qu’ils soient mis au courant des nouveautés. Les annonces seront utiles pour :

             -Annoncer le début des nouvelles campagnes (nouveaux burgers, frites, etc)

             -Annoncer les changements de procédure au sein du McDonald’s

             -…

Il y aura un onglet de champ de discussion qui sera ouvert à tous. Tout le monde pourra y discuter, y laisser un message mais aussi répondre aux messages fait par ses collègues.

Le dernier onglet sera l’onglet des « horaires », il servira quant à lui aux managers pour mettre les horaires en ligne toutes les semaines une fois qu’ils sont prêts, et pour les crew’s à les télécharger et de pouvoir les avoir n’importe quand.

 Tous les membres de l’équipe seront évidements capables de gérer leur compte personnel, de changer leur mot de passe, de mettre à jour leur adresse mail, etc…

Nous devons donc créer un site facile d’utilisation, intuitif et capable d’assurer une meilleure communication au sein du McDonald’s.

User stories

1. En tant qu’utilisateur, je veux pouvoir me connecter pour accéder au site.

* L’utilisateur aura accès à un écran de login possédant deux champs de textes pour son email et son mot de passe ainsi qu’un bouton « se connecter » et « mot de passe oublié ». Comme ceci :

Une image contenant capture d’écran

Description générée automatiquement

* Lorsque l’utilisateur clique sur le bouton se connecter, l’application va vérifier en base de données si ses informations s’y trouvent et sont correctes.
* Pour ce faire, il faut effectuer une requête de type POST à l’url suivant : localhost :8000/api/utilisateurs
* Si les informations sont bonnes, l’utilisateur se connecte au site et accède directement à la page des annonces à l’url suivant : localhost :3000/announcements , sur laquelle se trouve aussi une barre de navigation.
* Si les informations ne sont pas correctes, l’utilisateur revient simplement sur une nouvelle page de login sur laquelle se trouve un message, comme ceci :

Une image contenant capture d’écran

Description générée automatiquement

1. En tant qu’utilisateur, je veux pouvoir accéder à la page des annonces afin de les consulter.

* L’utilisateur aura deux possibilités d’accéder à la page des annonces, soit directement après son login, soit grâce à la barre de navigation qui l’emmènera à l’url suivant localhost :3000/announcements
* Une fois sur la page des annonces, l’utilisateur aura une liste de toutes les annonces qui s’offrira à lui :

Une image contenant capture d’écran

Description générée automatiquement

* Pour charger toutes les annonces, il faut effectuer une requête à la base de données : GET localhost :8000/api/notice
* Pour pouvoir consulter une annonce, l’utilisateur n’aura qu’à cliquer dessus. Cela aura pour effet de l’emmener sur une page qui contient juste l’annonce en question, se trouvant à l’url suivant localhost :3000/announcements/<id>

Une image contenant capture d’écran

Description générée automatiquement

* Pour charger l’annonce, il faut effectuer une requête à la base de donnée de la sorte : GET localhost :8000/api/notice/<id>

1. En tant qu’utilisateur de type manager, je veux pouvoir ajouter des utilisateurs afin d’ajouter tous les membres de mon équipe.

* Pour pouvoir ajouter un utilisateur, l’utilisateur devra se rendre sur la page de gestion de profil localhost :3000/profile sur lequel il trouvera un bouton pour ajouter un utilisateur.
* Une fois que l’utilisateur aura cliquer sur ce bouton, il sera rediriger vers la page d’ajout localhost :3000/register

Une image contenant capture d’écran

Description générée automatiquement

* L’utilisateur devra compléter les informations et ensuite cliquer le bouton envoyer.
* Lorsque l’utilisateur clique sur le bouton créer, une requête POST est effectuée à la base de donnée via l’url suivant : locahost :8000/api/users

1. En tant qu’utilisateur de type manager, je veux pouvoir ajouter une annonce afin de la rendre visible par tout le monde.
2. En tant qu’utilisateur, je veux pouvoir accéder aux horaires afin de les télécharger.
3. En tant qu’utilisateur de type manager, je veux pouvoir ajouter un horaire afin de le rendre accessible à tout le monde.
4. En tant qu’utilisateur, je veux pouvoir accéder à mon profil afin d’y consulter mes données personnelles.
5. En tant qu’utilisateur, je veux pouvoir modifier mes informations personnelles afin d’y effectuer des changements si nécessaire.
6. En tant qu’utilisateur, je veux pouvoir consulter le chat afin d’y lire les messages.
7. En tant qu’utilisateur, je veux pouvoir poster un message dans le chat afin de discuter avec les autres membres.
8. En tant qu’utilisateur, je veux pouvoir cliquer sur un bouton mot de passe oublié lors du login afin de réinitialiser mon mot de passe.
9. En tant qu’utilisateur de type manager, je veux pouvoir supprimer des utilisateurs afin de faire des mise à jour de mon équipe.
10. Choix technologiques

Back-end

En ce qui concerne les technologies niveau serveur nous allons utiliser Django combiné à python. Nous avons fait ce choix car nous pensons que Django est le framework qui a le plus grand avenir devant lui. Certes, beaucoup de site utilisent encore du PHP avec Symphony mais nous pensons que de plus en plus de développeurs vont se tourner vers Django.

Alors pourquoi Django et non Flask ? Alors certes, Flask est beaucoup plus libre que Django mais Django offre plus de possibilités comme un panneau administrateur, une interface de base de données, une architecture dossier pour notre application et notre projet mais surtout un ORM. La principale raison est que nous utilisons une base de données relationnelle et donc nous avons besoin d’un ORM et Flask ne nous en offre pas un directement. Flask est plutôt conseillé dans le cas d’application utilisant une base de données non-relationnelle.

Pour ce qui est de notre base données, nous avons choisi d’utiliser MySQL. Notre justification est assez simple, nous utilisons une DB relationnelle. Pourquoi MySQL et pas un autre qui permet le relationnel ? Simplement par choix car dans un autre cours nous utilisons du MySQL, le soucis de performance n’intervient pas réellement dans notre choix car notre DB restera assez petite.

Front-end

Pour ce qui est des technologies front-end nous allons utiliser React. Car comme pour Django, nous pensons qu’il est préférable d’apprendre une technologie fortement utilisée dans le monde du développement, ce qui est le cas de React.

1. Schéma relationnel de la base de données

