

Examen — React & Node

Karl MARQUES BERNARDO, Amin NAIRI

28/07/2022

Sommaire

1	Introduction	2
2	Fonctionnalités	2
2.1	Inscription, authentification, profile	2
2.2	Gestion de relations et d'amis	2
2.3	Messagerie	2
2.4	Modération	3
2.5	Logs	3
2.6	Analytique	3
3	Contraintes	3
3.1	Client	3
3.1.1	Responsive	3
3.1.2	Hooks	3
3.1.3	Routes	3
3.1.4	Contextes	3
3.2	Serveur	4
3.2.1	Base de données	4
3.2.2	Opérateurs d'aggrégation	4
3.2.3	Hébergement	4
3.3	Général	4
3.3.1	GitHub & GitLab	4
3.3.2	Dépôt unique	4
3.3.3	Livrable et Rendu	4
4	Bonus	4
4.1	Paquet NPM	4
4.2	Temps Réel, WebSocket & Server-Side Events	4
4.3	React Native	4
4.4	PropTypes & Test E2E	4
4.5	Rendu Côté Serveur	5

1 Introduction

Vous êtes en mission pour l'École Supérieure de Génie Informatique et votre objectif est de proposer une plateforme permettant de créer une communauté en ligne en mettant en avant les étudiants.

2 Fonctionnalités

2.1 Inscription, authentification, profile

En tant qu'utilisateur de la plateforme, je dois pouvoir m'inscrire à la plateforme avant de pouvoir y accéder. Je dois pouvoir remplir un listing des technologies qui m'intéressent (javascript, C++, cloud computing, reverse engineering, ...) parmi une liste déjà définie à l'avance.

Une fois inscrit, je dois pouvoir valider mon inscription en utilisant un lien temporaire reçu par mail.

Je dois également pouvoir m'authentifier à l'aide d'un système de jeton d'authentification temporaire.

Si j'ai oublié mes informations, je dois pouvoir réinitialiser mon compte grâce à un lien que je recevrais par mail.

Enfin, si je le souhaite, je dois pouvoir modifier mon profil, notamment mes informations personnelles, mon mot de passe ainsi que ma liste de technologies préférées.

2.2 Gestion de relations et d'amis

En tant qu'utilisateur, je dois pouvoir avoir accès à ma liste d'amis. Je dois également pouvoir avoir accès à la liste des invitations en attente que je peux accepter ou refuser, dans tous les cas, la personne en question doit être informée du statut de l'invitation.

Si je le souhaite, je dois pouvoir ajouter de nouveaux amis en accédant à la liste des étudiants inscrits à la plateforme. Un système de recherche doit me permettre d'affiner ma découverte de nouveaux étudiants sur des critères comme le nom, la filière ou sa liste de technologies préférée.

Si je ne m'entends plus avec quelqu'un, je dois pouvoir supprimer une relation avec un de mes amis. Si je le fais, je ne pourrais plus envoyer de messages à cette personne, par contre je pourrais toujours avoir accès aux messages déjà envoyés. Si je change d'avis, je pourrais toujours envoyer une nouvelle demande d'amis.

Si je le souhaite, je dois également pouvoir reporter une personne pour motif de harcèlement, faux profile, ou pour une autre raison qu'il sera nécessaire de justifier. Si vous le souhaitez, vous pouvez rajouter d'autres motifs par défaut.

2.3 Messagerie

En tant qu'utilisateur, je dois pouvoir avoir accès à la liste des personnes que j'ai contactées par le biais d'un système de messagerie.

Si je le souhaite, je peux contacter une nouvelle personne par message, dans la limite des étudiants faisant partie de mon réseau d'amis.

Il ne doit pas être possible de supprimer les messages d'une conversation. Si je le fais, le message doit être remplacé par un message générique indiquant que ce message a été supprimé.

Il doit être possible de modifier un message, si je le fais, une indication doit me permettre de savoir quels sont les messages modifiés et à quelles heures.

2.4 Modération

En tant qu'administrateur, je dois pouvoir modérer les comptes d'utilisateurs avec la possibilité de modifier leurs informations de compte.

Je dois également pouvoir modérer les utilisateurs qui ont été reportés par d'autres utilisateurs. Je dois notamment pouvoir prendre une décision : donner suite au report (bannir un utilisateur par exemple), envoyer un message d'avertissement sur son compte ou classer sans suite la plainte.

Je dois pouvoir créer ou supprimer des comptes utilisateurs à n'importe quel moment. Dans le cas d'une suppression, toutes les données doivent être conservées sauf les données pouvant identifier de manière personnelle l'étudiant. Ainsi, par exemple, l'email doit être remplacé par un email par défaut.

Je dois également modérer les messages instantanés, notamment les messages jugés injurieux ou portant atteinte à l'intégrité d'une personne. Lorsque je le fais, le message doit apparaître comme modéré et visuellement différents des autres messages. Il ne doit pas être possible d'ajouter ou modifier des messages.

2.5 Logs

Je dois pouvoir avoir accès à un système de logs serveur me permettant de voir en temps réel quelles sont les erreurs qui sont apparues sur mon serveur. Je dois pouvoir ainsi rechercher et filter les logs par sévérité (au sens de la RFC5424), par date, par application (serveur ou client), par contenu et tout autre filtre que vous jugerez utile.

2.6 Analytique

Enfin, je dois pouvoir avoir accès à toutes les données d'analytique du site client, notamment lorsqu'un utilisateur clique sur un bouton d'action. Par exemple, affichez le nombre de conversion (inscriptions confirmées), les durées de sessions moyennes, les appareils utilisés pour accéder au site client, le nombre de visiteurs en temps réel, et toute autre information analytique que vous jugerez utile.

3 Contraintes

3.1 Client

3.1.1 Responsive

Le site doit être utilisable et testable par les membres du jury lors de la soutenance sur appareils mobiles.

3.1.2 Hooks

Vous devez obligatoirement utiliser des composants fonctionnels avec l'API des Hooks de React. Utiliser tous les hooks que vous jugerez utiles afin d'adopter les bonnes pratiques d'écritures de composants et optimiser le runtime.

3.1.3 Routes

Vous devez faire en sorte que le site soit le plus rapide possible, notamment grâce à l'utilisation de routes dynamiques, mais également d'autres bonnes pratiques (chargement des images le plus tard possible, chargement conditionnel, etc...).

3.1.4 Contextes

Vous devez avoir au moins un contexte vous permettant de gérer les informations de l'utilisateur au travers de toutes les pages de votre site. Vous pouvez bien entendu avoir d'autres contextes pour d'autres parties de votre application.

3.2 Serveur

3.2.1 Base de données

Vous devez obligatoirement utiliser deux bases de données : une relationnelle (MariaDB, Postgres, ...) et une autre NoSQL (MongoDB), notamment pour toute la partie recherche Full-Text.

3.2.2 Opérateurs d'agrégations

Vous devez montrer que vous avez compris comment fonctionnent les opérateurs d'agrégations et comment les utiliser avec MongoDB.

3.2.3 Hébergement

Vous devez obligatoirement héberger votre solution en ligne. Vous pouvez utiliser des PaaS en ligne pour héberger votre serveur et vos bases de données, ou utiliser une solution type VPS. Votre site doit être accessible au moins 1 jour avant la soutenance, et pendant une semaine après la fin de la soutenance.

3.3 Général

3.3.1 GitHub & GitLab

Votre code source doit être historisé sur GitHub ou GitLab. Vous devez vous organiser en issues et pull request. Chaque membre du groupe doit avoir un nombre d'issues et de pull request équitable avec le reste des membres de son groupe. Toutes les personnes doivent participer au projet et sur toutes les parties du projet.

Votre projet doit être disponible en ligne 1 semaine après votre soutenance.

3.3.2 Dépôt unique

Votre projet doit être hébergé sur un seul dépôt et non pas deux dépôts séparés pour la partie client et serveur. Si vous le souhaitez, vous pouvez vous utiliser des outils de gestion Workspace comme les Workspaces de NPM, Yarn, Learn, etc... afin de vous organiser.

3.3.3 Livrable et Rendu

Votre livrable doit obligatoirement inclure le dossier .git qui contient l'historique des commits du projet en entier.

4 Bonus

4.1 Paquet NPM

Vous pouvez utiliser des paquets NPM pour héberger certaines parties de votre code source. Si c'est le cas, vous devrez rendre ces paquets publiques et les ajouter à votre liste de dépendances.

4.2 Temps Réel, WebSocket & Server-Side Events

Vous pouvez utiliser des technologies temps réel pour votre application, notamment pour la partie messagerie pour la rendre instantanée ou pour l'analytique afin d'avoir des données instantanées d'utilisation du site.

4.3 React Native

Vous pouvez construire une application mobile permettant de parcourir votre application sur un appareil iOS ou Android. Si c'est le cas, vous devrez présenter la version mobile & navigateur Web.

4.4 PropTypes & Test E2E

Vous pouvez utiliser la librairie intégrée à ReactDOM afin de tester vos composants fonctionnels. Également, l'utilisation de PropTypes est appréciée afin de valider que vos composants reçoivent les bons paramètres.

4.5 Rendu Côté Serveur

Vous pouvez utiliser la librairie intégrée à ReactDOM afin de construire le marquage HTML côté serveur et l'hydrater côté client afin d'améliorer la réactivité du site.