# Consignes pour l'ensemble du projet

* Un maximum de points sera accordé aux applications écrites à l'aide des meilleures pratiques :
  + Utilisation de components (Podcast, Episode, Footer, autres?)
  + Clarté du code
  + Pas de warnings ou erreurs REACT (vérifiez la console du navigateur).
  + Pas d'erreur de lint (utilisez "npm run lint" et "npm run fix"). Ne modifiez pas le fichier eslintrc.json pour désactiver des règles de validation eslint. N'utilisez pas non plus de directives pour désactiver des erreurs eslint sauf pour les react/prop-types, que vous pouvez désactiver.

Une image contenant texte, Police, blanc, typographie

Description générée automatiquement

* N'utilisez pas de composantes externes (par exemple pour la pagination, codez ces composants vous-mêmes).
* Un compte a été créé pour vous: e1234567@site.com/eE1234567! où 1234567 est votre numéro de matricule. Trois podcasts ont été ajoutés à vos abonnements.

**NOTE**

**Un dépôt gitlab a été créé pour vous avec un projet initial. Le projet est réalisé de façon individuelle.**

<https://gitlab.com/cegepmaisonneuve/aut23/5d6/tp2/>

# Réalisation

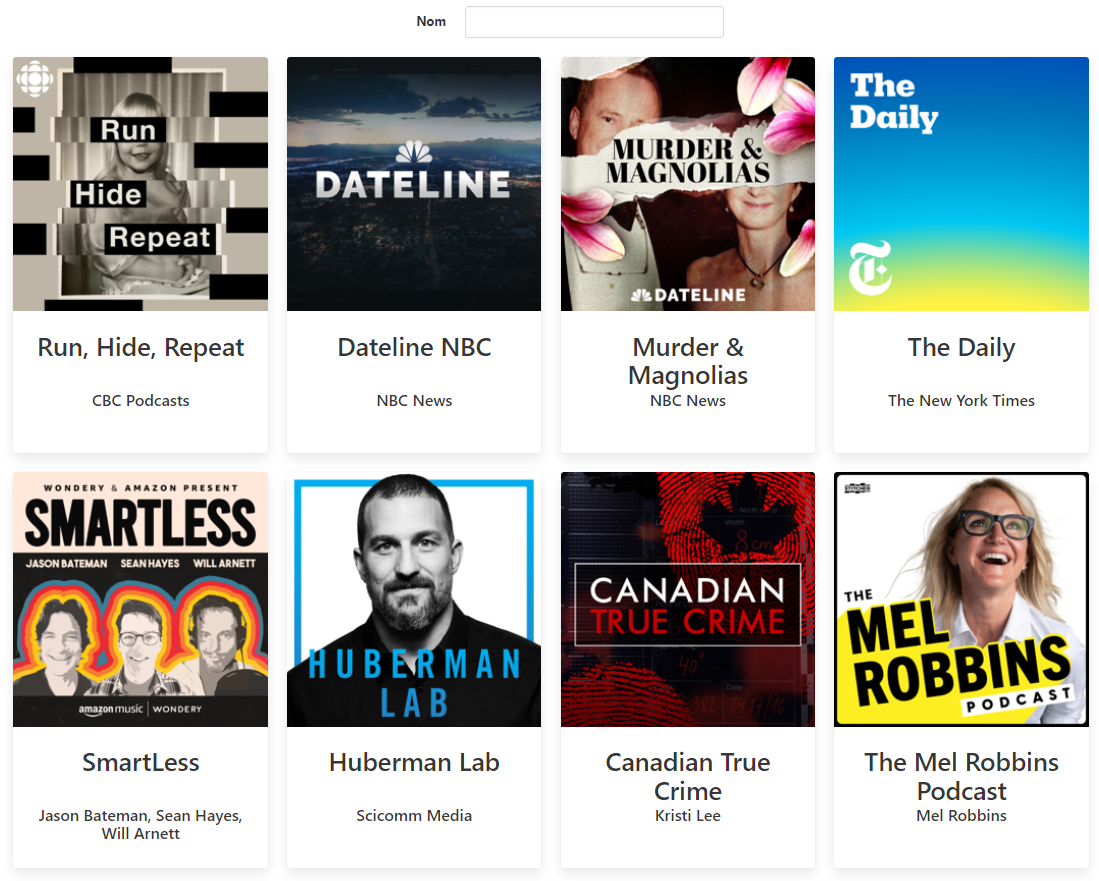
* Consiste à écrire un le code REACT nécessaire pour exploiter l'API <https://podcastsapi.herokuapp.com/>.
* Voir tp2-\*.gif.

# Remise

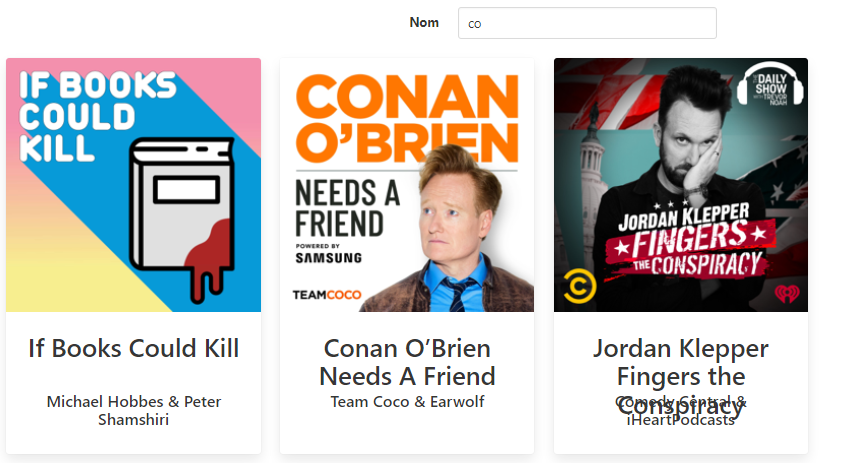
* Faire un git/push.

# Page d'accueil

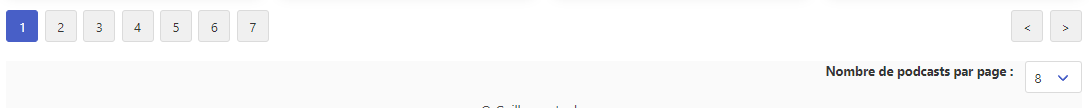
* Afficher les 30 podcasts les plus populaires (voir méthode [/podcasts/top](https://podcastsapi.herokuapp.com/#/default/get_podcasts_top) de l'api).
* On affiche les podcasts en 4 colonnes en mode desktop, 3 colonnes en mode tablette et deux colonnes en mode mobile.
* Indiquer votre matricule (e1234567) au bas de la page ([footer](https://bulma.io/documentation/layout/footer/)).
* Pour chacun des podcasts, le nom (name), l'auteur (artist) et l'image (artworkUrl) du podcast sont affichées.
* Un clic sur un podcast permet de passer à la page detailspodcast.



* Une zone de filtre permet de filtrer les podcasts à l'aide de leur nom (name).



* Au bas de la page, un système de pagination permet la navigation. Il y a un lien permettant de se déplacer vers le numéro de page directement. Aussi, un bouton page suivante et page précédente sont présents. Le nombre de pages varie en fonction du filtre de recherche par nom et du nombre de podcasts par page.



* Le système de pagination indique la page courante.
* Les liens de pagination suivant et précédent sont désactivés aux moments opportuns.
* Le nombre de podcasts par page est soit 8, 12 ou 24.

Une image contenant texte

Description générée automatiquement

* Le nombre de podcasts par page persiste entre les lancements de l’application.
* Les données retournées par la méthode [/podcasts/top](https://podcastsapi.herokuapp.com/#/default/get_podcasts_top) ont le format suivant:

[

{

"podcastId": "1606242135",

"name": "Welcome to Paradise",

"artist": "CBC Podcasts",

"genres": [

"Society & Culture"

],

"artworkUrl": "https://is5-ssl.mzstatic.com/.../250x250bb.png"

{

"podcastId": "1464919521",

"name": "Dateline NBC",

"artist": "NBC News",

"genres": [

"True Crime",

"News"

],

"artworkUrl": "https://is1-ssl.mzstatic.com/.../250x250bb.png"

},

…

]

# Menu

* Ajouter un menu au projet.
* Le menu contient les choix Home, Login et SignUp lorsqu'aucun utilisateur est connecté.



* Le menu contient les choix Home, Profile, Subscriptions et Logout lorsqu'un utilisateur est connecté.



* Le bouton Logout supprime le jeton d'authentification du contexte global et l'utilisateur est redirigé vers la page racine.

# Footer

* Le component Footer est affiché au bas des pages. Il indique votre numéro de matricule.

© e1234567

# Login

* Permet à un/une utilisateur/utilisatrice de s'authentifier.
* Utilisez e1234567@site.com/eE1234567! comme nom d'utilisateur et mot de passe où 1234567 est votre numéro de matricule.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

#### Authentification au serveur podcastapi à l’aide d’un compte de site

* Utilisez la méthode [/auth/token](https://podcastapi5d6.herokuapp.com/#/default/post_auth_token) de l'api pour obtenir un jeton.
* Vous devez envoyer, dans le BODY de la requête, les données d'authentification

{

"email": "e1234567@site.com",

"password": "eE1234567!"

}

* Si l’utilisateur existe, vous recevrez le résultat sous le format suivant:

{

"token": "eyJhbGciOiJI[…]ww7GWyJL0"

}

* Si l'authentification échoue, affichez le message provenant du serveur (propriété message). La réponse contient la raison de l’échec.

{

"message": "Le message d'erreur"

}

* Après une authentification réussie, l'utilisateur est redirigé vers la page des abonnements.
* Ajouter le code nécessaire de façon à que le rechargement de l’application (F5) ne perde pas le jeton d’authentification (utilisation de localStorage).

#### Authentification au serveur podcastapi à l’aide d’un compte Google

* Le serveur supporte aussi l’authentification Google. Par conséquent, ajoutez un bouton « Se connecter avec Google » à l’interface.
* Utilisez la méthode [/auth/tokenFromGoogleLogin](https://podcastapi5d6.herokuapp.com/#/default/post_auth_tokenFromGoogleLogin) de l'api pour obtenir un jeton en échange d’un credential Google.
* Vous devez envoyer, dans le BODY de la requête, le champs credential qui est retourné par Google au moment de l’authentification

{

"credential": "ABCDEFGHIJ123…123",

}

* Si l’utilisateur existe, vous recevrez le résultat sous le format suivant:

{

"token": "eyJhbGciOiJI[…]ww7GWyJL0"

}

* Si l'authentification échoue, affichez le message provenant du serveur (propriété message). La réponse contient la raison de l’échec.

{

"message": "Le message d'erreur"

}

* Le client\_id du service d’authentification est :

client\_id: "660418432193-q2b1ig5la4og1pk2m3i2m9t0ttu858mo.apps.googleusercontent.com",

* Activez aussi le *One Tap Sign In* dans la page de Login.
* Après une authentification réussie, l'utilisateur est redirigé vers la page des abonnements.
* Ajouter le code nécessaire de façon à que le rechargement de l’application (F5) ne perde pas le jeton d’authentification (utilisation de localStorage).

# Sign Up

* La page Sign Up permet à un utilisateur de créer un compte utilisateur.

Une image contenant texte

Description générée automatiquement

## Validations

* Le courriel doit contenir contient le caractère "@".
* Le mot de passe contient au moins 8 caractères.
* Le mot de passe contient au moins un symbole parmi les suivants: !@#$%&\*()[].
* Le mot de passe contient au moins une lettre en majuscule.
* Le mot de passe contient au moins une lettre en minuscule.
* Si le champ confirmation de mot de passe contient du texte, alors la confirmation de mot de passe est identique au mot de passe.
* Lorsque la règle de validation est valide, le même message est affiché mais en vert. Ces indications changent lors de la frappe, en continu (voir gif).

Une image contenant texte

Description générée automatiquement

* Le bouton Sign Up est désactivé ([disabled](https://www.w3schools.com/tags/att_input_disabled.asp)) et inutilisable si des règles de validations ne sont pas respectées.

## Bouton SignUp

* S'il n'y a pas d'erreur de validation, envoyez une requête au serveur en utilisant l'API [/auth/register](https://podcastsapi.herokuapp.com/#/default/post_auth_register_) (méthode POST) avec, dans le corps du message (body), le JSON {"email":"X", "password":"Y"} où X est le courriel désiré et Y est le mot de passe désiré.
* Affichez un message d'erreur si le serveur retourne une erreur. La réponse contient la raison de l’échec.

{

"message": "Le message d'erreur"

}

* Si l'enregistrement fonctionne, redirigez l'utilisateur vers la page login.

# Profile

* La page Profile permet à l'utilisateur de modifier son profile (email et mot de passe).

Une image contenant texte

Description générée automatiquement

* Si l'utilisateur n'est pas authentifié, cette page n'est pas disponible.
* Au chargement de la page, utilisez l'API [/user](https://podcastsapi.herokuapp.com/#/default/get_user) (méthode GET) pour obtenir et afficher le courriel de l'utilisateur présentement connecté.

## Validations

* Le courriel doit contenir contient le caractère "@".
* Le mot de passe contient au moins 8 caractères.
* Le mot de passe contient au moins un symbole parmi les suivants: !@#$%&\*()[].
* Le mot de passe contient au moins une lettre en majuscule.
* Le mot de passe contient au moins une lettre en minuscule.
* Si le champ confirmation de mot de passe contient du texte, alors la confirmation de mot de passe est identique au mot de passe.
* Lorsque la règle de validation est valide, le même message est affiché mais en vert. Ces indications changent lors de la frappe, en continue (voir gif).

Une image contenant texte

Description générée automatiquement

* Le bouton Update Profile ne fonctionne que si les règles de validations sont respectées. Il est [disabled](https://www.w3schools.com/tags/att_input_disabled.asp) dans le cas contraire.
* Le bouton Delete account demande affiche une fenêtre de [confirmation modale](https://bulma.io/documentation/components/modal/) et ensuite supprime l'utilisateur ([/user](https://podcastsapi.herokuapp.com/#/default/delete_user) avec la méthode DELETE). Ensuite, le jeton est supprimé du contexte global et l'utilisateur est redirigé vers la page d'accueil.

Une image contenant texte

Description générée automatiquement

# Page détails

* Affiche les informations détaillées du podcast ainsi que les épisodes (voir méthode [/podcast?podcastId=X](https://podcastsapi.herokuapp.com/podcast?podcastId=1439621628) de l'api).
* En haut de page, afficher le nom (name), l'artiste (artist), les genres séparés d'une virgule (genres) et la description du podcast.
* Ensuite, pour chacun des épisodes, affichez le titre (title), le descriptif de l'épisode (content) et la date de l'épisode (isoDate) ainsi que le contrôle audio permettant d'écouter l'épisode.

Une image contenant texte

Description générée automatiquement

* Pour le contrôle audio, utilisez preload="metadata" pour éviter le chargement complet de tous les épisodes sur la page. Par exemple: <audio id="audioUrl" controls="controls" preload="metadata" src="…" />.
* Utilisez un component pour l’affichage des épisodes.

# Abonnement/Désabonnement à un podcast

* Pour un utilisateur authentifié, un bouton d’abonnement (ou de désabonnement) s’affiche au chargement de la page. Il permet à l’utilisateur présentement authentifié de s’abonner ou se désabonner du podcast.
* Pour modifier les abonnements de l'utilisateur présentement authentifié, on utilise l'api [/subscription?podcastId=X](https://podcastsapi.herokuapp.com/#/default/get_subscription) où X est le podcastId du podcast.
* Ceci est une api authentifiée, elle nécessite un jeton d'authentification bearer token pour fonctionner.
* L’appel à [/subscription?podcastId=X](https://podcastsapi.herokuapp.com/#/default/get_subscription) avec la méthode GET retourne un json contenant la propriété isSubscribed. C'est une propriété booléenne. Utilisez cette propriété pour déterminer si vous devez afficher le bouton pour s’abonner ou se désabonner du podcast.

{"isSubscribed":true} ou {"isSubscribed":false}

* Un appel à [/subscription?podcastId=X](https://podcastsapi.herokuapp.com/#/default/post_subscription) avec la méthode POST permet d'ajouter le podcast dont le id est spécifié en paramètre aux abonnements de l'utilisateur.
* Un appel à [/subscription?podcastId=X](https://podcastsapi.herokuapp.com/#/default/delete_subscription) avec la méthode DELETE permet de supprimer le podcast dont le id est spécifié en paramètre aux abonnements de l'utilisateur.
* Lorsque l'utilisateur s'abonne ou se désabonne, la page courante ne change pas, mais le bouton approprié (abonnement ou désabonnement) est affiché.

# Subscriptions

* Affiche les abonnements de l'utilisateur.
* On affiche les podcasts en 4 colonnes en mode desktop, 3 colonnes en mode tablette et deux colonnes en mode mobile.
* Les données proviennent de la méthode [/suscriptions](https://podcastsapi.herokuapp.com/#/default/get_subscriptions) de l'api.
* Si l'utilisateur n'est pas authentifié, retournez l’utilisateur à la page d’accueil.

Une image contenant texte, Visage humain, capture d’écran, personne

Description générée automatiquement