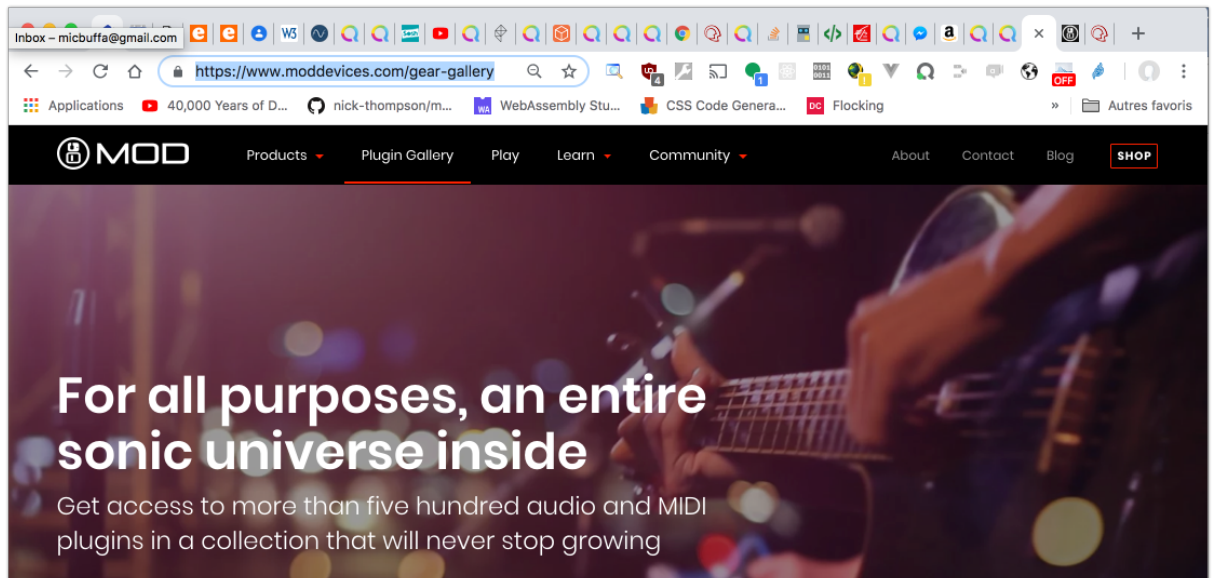


Un magasin de plugins audio

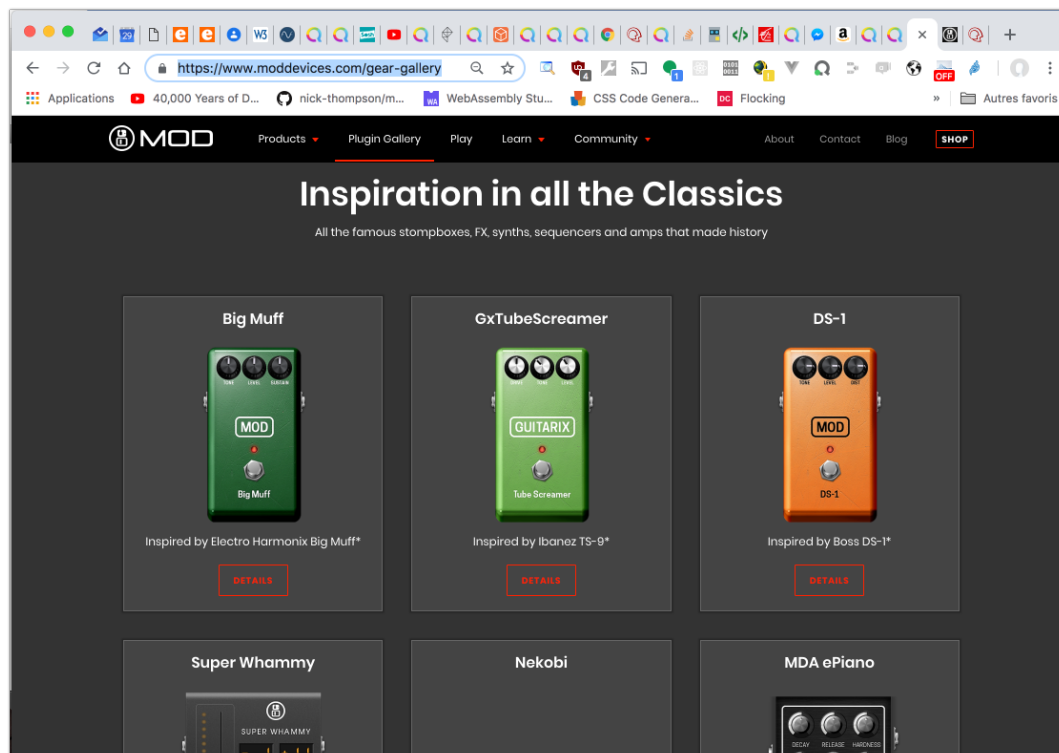
Vous devrez réaliser avec un de ces frameworks : React, VueJS ou Angular et avec un back end NoSQL ou FireBase si vous savez l'utiliser, un magasin de plugins audio.

C'est une copie d'un site réel : <https://www.moddevices.com/gear-gallery> n'hésitez pas à jouer avec pour comprendre ce que vous devez faire.

Page présentant tous les plugins audio disponibles :

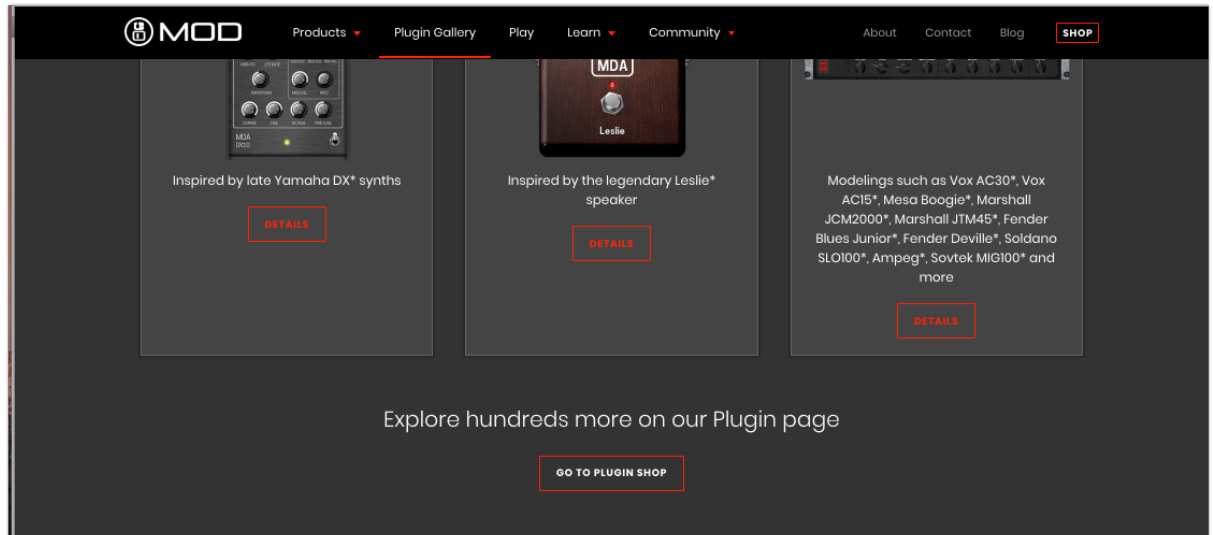


Si on scrolle on voit le catalogue des plugins disponibles :

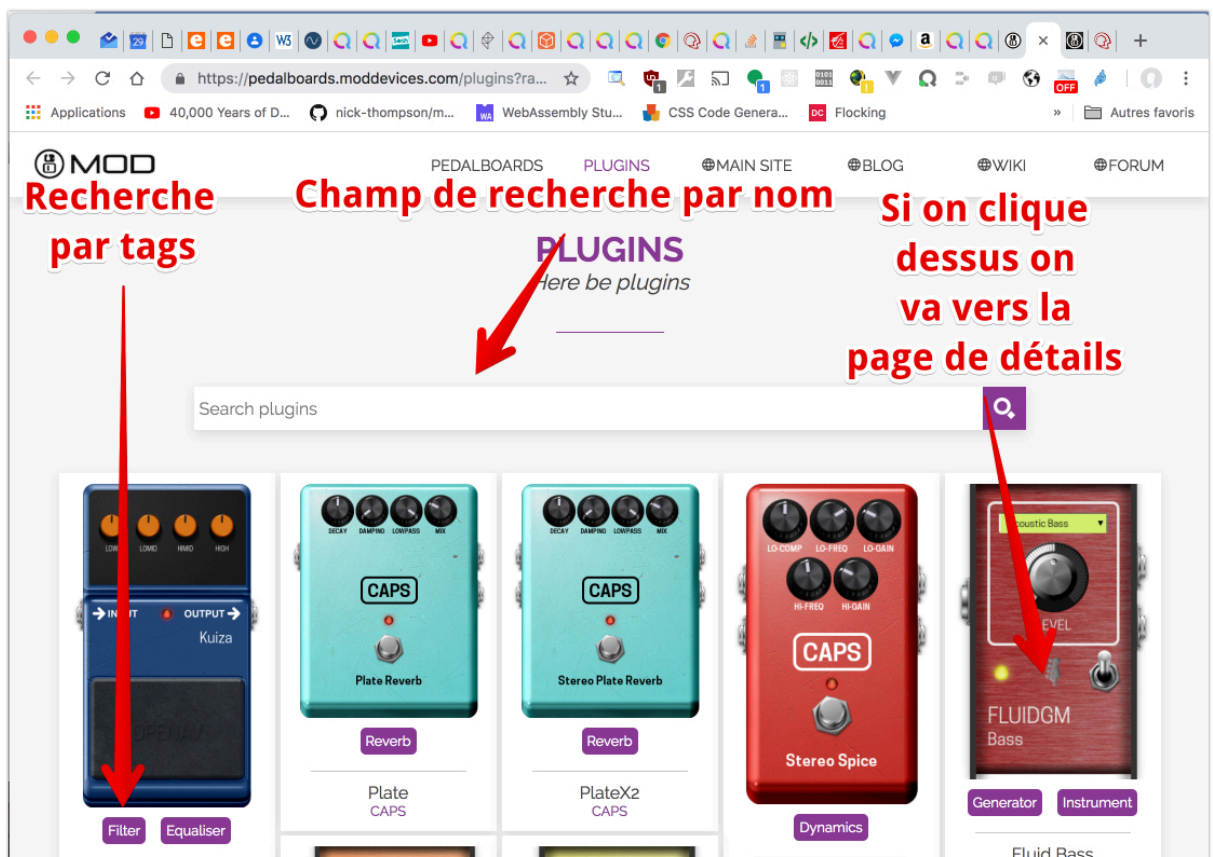


Dans cette vue si on clique sur la zone « détails » on va vers la page de détails d'un plugin (détaillée plus loin).

Au début on ne voit que les derniers ajouts, mais au bout d'un moment, on peut aller au « plugin shop » :



Page d'accueil du « plugin shop » :



Page de détail :

Page du créateur

OpenAV
http://openavproductions.com

Kuiza

Détails des paramètres (boutons etc.)

Nb de pedalboards qui l'utilisent

Description

A 4 band equalizer with preset bands.

Control	Default	Min	Max
Gain	0.50	0.00	1.00
Low Gain	0.50	0.00	1.00
LoMid Gain	0.50	0.00	1.00

Là aussi si on clique sur un tag on a tous les plugins qui s'affichent par tag...

TRAVAIL A FAIRE PARTIE OBLIGATOIRE :

- Aller sur le site <https://pedalboards.moddevices.com> et tester la véritable application (oui, c'est un vrai site)
- Implémenter dans un premier temps la partie « consultation » du site des plugins, avec le moteur de recherche, la pagination, la recherche par tags etc...
- Je vous conseille de décrire un ensemble de pédales avec un fichier JSON fait à la main dans un premier temps. Vous le mettez sur dans votre base de données, et vous faite l'affichage, la recherche etc. Rappelez-vous qu'on a vu en TP un moyen simple de faire une BD en JSON avec le site

<https://jsonplaceholder.typicode.com>

- Un plugin c'est :
 - Un vendeur (un URL du site du créateur, <http://openavproductions.com> dans l'exemple précédent)
 - Le nom du créateur (Kuiza dans l'image de la pédale bleue)
 - Une image
 - Un ou plusieurs tags (mots clés qui serviront d'indexs)
 - Une description textuelle
 - Une table avec le détail des paramètres (en gros, pour chaque bouton : nom, valeur par défaut, min et max)
- Dans un second temps vous implémenterez le formulaire de publication d'un nouveau plugin
- Un auteur doit être authentifié sur le site (gestion login/password), il a un nom, l'URL de sa home page externe. Si vous utilisez FireBase, optionnellement : regardez comment faire une authentification par exemple avec un compte gmail ou github.
- On demandera à l'auteur de remplir un formulaire classique en décrivant les propriétés du plugin qu'il propose
- Vous devrez traiter le cas de l'upload de l'image sur le serveur.
- Ensuite vous vous attaquerez à la suppression et à la modification des plugins (seul un auteur peut supprimer ses propres plugins, ou éventuellement un admin (compte admin = optionnel))

MODALITES DE RENDU

- Travail à faire en groupe de 2 personnes
- Framework au choix
- Obligatoire : utiliser des trucs vus en cours. PHP ou autre interdit. Autre framework interdit (pas de angularJS ou de SpringBoot ou que sais-je). Sont

autorisés : les trois frameworks étudiés, back end node + MongoDB ou cloud (firebase).

- Date de limite de rendu : le 13/12/2018
- Vous m'enverrez par mail le GitHub de votre projet
 - Faites un Readme sur GitHub avec toutes les explications pour que je puisse faire tourner votre projet
- **GROS PLUS SI VOTRE PROJET EST EN LIGNE QUELQUE PART**
 - Vous êtes en Master 2, démerdez-vous ☺
- Je peux débarquer à tout moment entre le 13 et les vacances pour vous demander une démo.
- Projet pompé = vous assumez les risques.
- Vous avez le droit et vous êtes encouragés à utiliser des modules pour de belles UIs (material UI, etc.)

TRAVAIL OPTIONNEL : GESTION DES PEDALBOARDS

Ce travail est facultatif et vise les étudiants qui sont « à l'aise » avec les technologies utilisées.

Les plugins peuvent être assemblés en « pedalboards ». En réalité sur le vrai site, on peut à la fois publier des plugins et des pédalboards qui sont des assemblages de plugins.

On ne peut construire un pedalboard qu'à partir de plugins qui existent déjà dans la base de données.

Si on clique sur la page du plugin du dernier exemple de la page précédente, on verra tous les pedalboards qui l'utilisent (page suivante, mais le mieux est de tester le vrai site).

Un pedalboard est constitué :

- Un screenshot
- Un nom
- Un auteur
- Une date de publication.
- Optionnellement : un fichier audio mp3 de « preview » des sons qu'on peut obtenir
- Un bouton « Try now » qui envoie la configuration dans le hardware que le site modduo vend (on ignorera cette option, ou on mettra un bouton inactif dans la GUI)
- Un bouton « View more » qui donne le détail des plugins utilisés (voir figure deux page plus loin)

Dans le travail à faire, il faudra faire l'affichage et la publication de pedalboards. On utilisera dans un premier temps des images fictives et on prendra des ids de plugins existants pour composer un pedalboard.

PEDALBOARD FEED

Here be pedalboards

Filtering by Kuiza

[Clear](#)

Search pedalboards



FILTERS

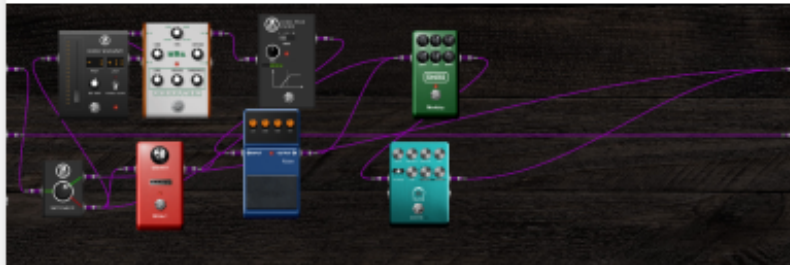
Delay
Distortion
Dynamics
Filter
Generator
Modulator
Reverb
Simulator
Spatial
Spectral
Utility

Beta



BBSvein
4 months ago

Guitar Fuzz Synth

[Try Now](#)[View More](#)

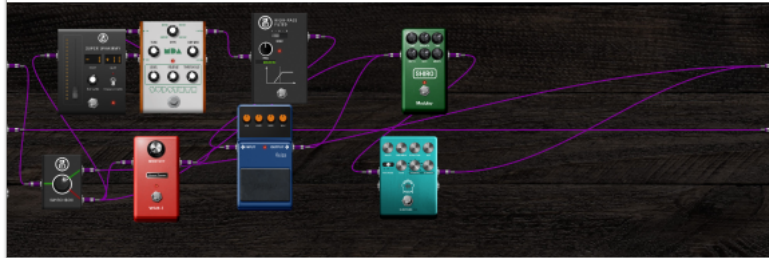
Rogério Couto
a year ago

Love is all you need #2 [rogeriocouto.com.br]



Same pedalboard as the previous one but now with a dry/wet chorus mix control, assigned to knob #1
To obtain a correct mix I used the "stereo x-fade" plugin with a gain plugin set to -3db in front of it to control the dry signal and another gain plugin after it set to +3db to compensate the losses of

Guitar Fuzz Synth



Plugins



Spectral

SubSynth
MDA



Filter

HighPassFilter
MOD



Spectral

Super Whammy
MOD



Détails d'un pedalboard...