

SELLECT

Documentation technique

Informations

- Nom du projet : **Sellect**
- Type de document : documentation technique
- Date : 27/07/2022
- Auteurs (élèves à la Wild Code School) :
 - Audren FLOCH
 - Christophe BOINET
 - Gloria BESAMBA

Description

Sellect est un site web à usage des particuliers, l'association a pour objectif d'optimiser ou renégocier les coûts de contrats (assurance, mutuelle, téléphone, etc) de ses clients afin d'optimiser leur pouvoir d'achat.

Technologies utilisées

- *Langages* : **Javascript, Node.js, CSS, SQL**
- *Frameworks* : **React, Express, MySQL**
- *Modules frontend* : **DownloadJS, Moment, Qrcode, Calendly**
- *Modules backend* : **Brcrypt, Jwt, Formidable, Cookie-parser, Cors, Nodemailer**

Fonctionnalités

Générales :

- Présentation de l'association (objectifs, FAQ, statistiques, etc)
- Création d'un compte utilisateur, connexion, déconnexion
- Mise à jour de compte (email, téléphone, mot de passe)
- Envoi d'emails automatiques

Utilisateur :

- Prise de rendez-vous pour un audit (via un module Calendly)
- Envoi de contrats avec le coût actuel par mois
- Parrainage d'un autre utilisateur

Administrateur :

- Gestion des utilisateurs
- Envoi de compte-rendus d'audit
- Mise à jour des contrats utilisateurs (nouveau coût de contrat, calcul des économies réalisées par rapport à l'ancien contrat)
- Suppression éventuelle d'utilisateurs ou de fichiers

Base de données

Table User

id int AUTO-INCREMENT NOT NULL,
firstname varchar(255) NOT NULL,
lastname varchar(255) NOT NULL,
phoneNumber varchar(20) NOT NULL,
email varchar(190) NOT NULL,
password varchar(255) NOT NULL,
sponsorCode varchar(20) NULL,
referralCode varchar(255) NOT NULL,
role varchar(20) NOT NULL,
signupDate varchar(45) NOT NULL

Table File

id int NOT NULL AUTO-INCREMENT,
userId int NOT NULL FOREIGN-KEY,
name varchar(255) NOT NULL,
content varchar(255) NOT NULL,
category varchar(255) NOT NULL,
sendDate varchar(255) NOT NULL,
newSendDate varchar(255) NULL,
initialCost varchar(255) NULL,
newCost varchar(255) NULL,
gain int NULL

Routes backend

La table *Item* sert de modèle et n'est pas utilisée. Détails des routes (voir le fichier *router.js* dans le backend) :

Routes utilisateurs :

- 1 – Inscription des utilisateurs
- 2 – Obtention des données de tous les utilisateurs
- 3 – Connexion
- 4 – Déconnexion
- 5 – Obtention des données d'un utilisateur unique
- 6 – Suppression d'un utilisateur
- 7 – Mise à jour de l'email et du téléphone
- 8 – Mise à jour du mot de passe
- 9 – Obtention du parrain d'un utilisateur à partir du sponsorCode
- 10 – Obtention du nombre d'utilisateurs

Routes fichiers :

- 1 – Envoi de contrat
- 2 – Envoi de compte-rendu d'audit
- 3 – Obtention des fichiers de tous les utilisateurs
- 4 – Obtention des fichiers d'un utilisateur
- 5 – Téléchargement de fichier
- 6 – Visualisation de fichier
- 7 – Suppression de fichier
- 8 – Suppression de tous les fichiers d'un utilisateur
- 9 – Mise à jour d'un contrat
- 10 – Obtention des gains réalisés par l'utilisateur
- 11 – Obtention du nombre total de compte-rendus d'audit
- 12 – Obtention du nombre total d'économies réalisées par mois par tous les utilisateurs

Bugs Connus

1 - Message d'erreur apparaissant dans le terminal lors de la connexion dans le dashboard administrateur :

```
{
  code: 'ER_BAD_FIELD_ERROR',
  errno: 1054,
  sql: 'SELECT * FROM user WHERE id = NaN',
  sqlState: '42S22',
  sqlMessage: "Unknown column 'NaN' in 'where clause'"
}
```

Ce message n'affecte pas le bon fonctionnement du site.

2 - Lorsque l'on supprime les fichiers d'un utilisateur, le coût total des économies réalisées par an se met correctement à jour en temps réel sauf lorsqu'il ne reste plus qu'un seul fichier, l'ancien montant reste affiché en front mais disparaît si on actualise la page.

3 – La conversion du code de parrainage en qrcode a été commentée car le qrcode créé ne contenait pas le code de parrainage.

4 – La fonctionnalité Nodemailer a été implémentée mais a été commentée car elle fonctionnait de façon aléatoire.

Bon courage !