

# Project report

## URL

Glitch : not working

<https://ephemeral25-careshop.glitch.me>

Github : <https://github.com/Ephemeral25/careshop>

## Description

Ce projet reprend les même grandes lignes que le précédent mais dans un contexte totalement différent.

Il propose à l'utilisateur une interface web construite avec un serveur utilisant Vue.js. Cette interface interagit avec un serveur « backend » express.js qui nous permet d'interagir avec notre base de données.

En prenant de la hauteur par rapport au projet, nous avons essayé de nous détacher du perfectionnisme perpétuel qui nous à énormément ralentie lors du premier projet. Ajouter toutes les dépendances d'un coup, mettre en place une vue agréable AVANT de s'attarder sur la logique de données n'était certainement pas la bonne approche. Nous avons cherché cette fois ci à mettre en place une base évolutive, réfléchir à la data logic et au data binding entre nos deux serveurs avant de s'attarder sur le visuel ou à la mise en place de dépendance « superflux ».

## Conception schema

La première étape a été de choisir les technologies à employer. Au vu du premier projet nous avions déjà une idée, un front end basique et un backend simple qui ne nous rajouterons pas de contraintes au moment du déploiement. Nous nous sommes donc orienté dans un premier temps sur un front end Vue.js puis un backend express servant une API sur une base de données SQL. Cette même base de données utilisant le module sqlite nous permettant de stocker l'information au sein d'un fichier texte. Ce sont donc les grandes lignes directrices qui ont été prises dès le début.

Par la suite après la mise en place du système de Login/Enregistrement/Distribution des vues, nous sommes partie sur la mise en place de vuetify pour embellir notre application et créer la seconde grosse partie de notre application, le dashboard avec l'intégration de la contrainte CRUD.

La module Axios nous a permis de mettre en place un client http pour transformer notre server express en API.

Différentes routes sont appelées et exécute des requêtes SQL qui gère l'authentification. Le contrôle des données formulaires est faites depuis le front.

Une fois connecté c'est le module JWT qui entre en jeu et nous permet de générer un token d'authentification qui sera utilisé dans le reste du site.

## Difficulties

Les principales difficultés ont été d'adapter les différences ressources trouvées aux besoins et contraintes de notre application.

Problème au niveau de la mise en production (No PostCSS config found).

**Clément :** *Mais je dirais que la principale difficulté a été de reprendre un projet ayant initialement abouti à un échec avec exactement les mêmes contraintes. C'est aussi se remémorer les conflits passés mais cela a été surtout une remise en question de la conclusion que nous avons tiré après ce projet à savoir ne plus travailler ensemble car amitié ne fait pas forcément bon ménage avec travail. Un manque de motivation énorme à été présent et la précipitation de dernière minute à eu effet du coup manque de temps pour implémenter la partie CRUD. C'est un peu le même schéma qui s'est déroulé mais j'ai pris le temps de mettre en place la structure du projet et gérer toute l'authentification.*