OPTIBRAIN Web application

Tasks **completed**

* Setup of the front-end
* Setup of the server (used flask to create an API with PostgreSQL database)

Tasks **to be done**

* Restructure the sql request file from the CHUSJ database to match the front-end
* Tests (front and back end)
* Manage cache with React-Query
* Dynamic Re-Routing

Insights

* Access to the database can take up to 2 months due to administration documents
* Access to cathydb (CHUSJ db), ask Michael Sauthier : [michael.Sauthier.med@ssss.gouv.qc.ca](mailto:michael.Sauthier.med@ssss.gouv.qc.ca)
* Proper environment to launch the application within the server, ask Edem : [edem.tiassou.hsj@ssss.gouv.qc.ca](mailto:edem.tiassou.hsj@ssss.gouv.qc.ca)
* Front-end: React, Typescript, MUI material, Recharts, Axios
* Back-end: PostgreSQL, Flask (create a local cache if desired)
* Create your own configuration file for your SQL database (see Figure 1)
* Look for the TODOs in the code

Link to pertinent ressources

* a) Figma : <https://www.figma.com/file/cytmnd4MxKCq0k6B677wNR/SIAD-USIP> (done by Sandrine Laliberté)
* b) Already done project by CHUSJ research center: https://github.com/sauthiem/axolotl.git
* c) Flask-server + Postgres: https://www.youtube.com/watch?v=DlNIXC9SaF4&t=865s
* d) Dashboard inspiration: <https://www.youtube.com/watch?v=uoJ0Tv-BFcQ&t=10581s>
* e) React-Query: <https://www.youtube.com/watch?v=r8Dg0KVnfMA>

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Figure 1**. Edem will give you your access (add this file in /postgresql directory)

Une image contenant Bleu électrique, texte, Police, capture d’écran

Description générée automatiquement

**Figure 2**. Environment provided by Edem on my distant CHUSJ work environment