

Compte-rendu de mission Télécom Étude

Traitement de données médicales et représentation graphique sur site web

Emile Le Gallic

Telecom Paris, Institut Polytechnique de Paris F-91120, Palaiseau, France

emile.legallic@telecom-paris.fr

Novembre 2023

TABLE DES MATIÈRES

1. <u>Présentation des technologies utilisées</u>	2
1.1 <u>Django</u>	2
1.2 <u>PythonAnywhere</u>	2
2. <u>Description de l'application</u>	4
2.1 <u>Organisation du projet</u>	4
2.2 <u>Développement local</u>	4
3. <u>Utilisation de l'application web</u>	3
3.1 <u>Authentification</u>	5
3.2 <u>Visualisation des données</u>	5
4. <u>Note à propos des fichiers csv à importer</u>	6

1. Présentation des technologies utilisées

1.1 Django

Le choix a été fait de partir sur l'utilisation du framework **Django** qui permet la réalisation d'application web en Python. Ce choix repose sur le fait que l'ensemble des scripts de traitements des données de la mission précédente étaient réalisés en Python. Utiliser une application entièrement développée en Python permet donc d'éviter de séparer la partie application web de celle du traitement de données comme cela pourrait être le cas avec l'utilisation d'un framework de développement web tel que React ou Angular utilisant Javascript.

L'application Django est située dans le répertoire `/dashboard`.

Le code pour la partie web est situé dans le répertoire `/dashboard/client`.

Le code de traitement des données issues des fichiers csv (de la mission précédente) est situé dans le répertoire `/dashboard/client/csv_processing`.

Le code pour les deux pages webs de l'application (authentification et affichage des graphiques) est situé dans le répertoire `/dashboard/client/views`.

Les fichiers `/dashboard/client/constants.py` et `/dashboard/client/metrics.py` définissent respectivement les paramètres statiques utilisés dans l'application et dans les scripts de traitement des données et les différentes métriques utilisées dans la représentation des données.

Pour voir comment lancer l'application locale en Django, voir la section [2.2 Développement local](#)


2.2 PythonAnywhere

PythonAnywhere est un hébergeur spécialisé dans les application Python. Il a été choisi car propose des solutions d'hébergement d'applications Django ce qui est le cas de peu d'hébergeur.

L'accès à la console admin se fait à l'adresse : <https://eu.pythonanywhere.com/> (bien noter le eu. Au début).

Dans l'onglet « *Web* » du menu vous avez accès aux webapps du compte. Ici il en existe une seule, celle de `www.calipnos.com`. Vous avez accès aux métriques de l'application et à sa configuration.

Les noms de domaines `calipnos.com`, `calipnos.fr` et `calipnos.eu` achetés depuis Strato ont été redirigé vers le site hébergé sur PythonAnywhere.



pythonanywhere

by ANACONDA

[Dashboard](#)
[Consoles](#)
[Files](#)
[Web](#)
[Tasks](#)
[Databases](#)

[www.calipnos.com](#)

[Add a new web app](#)

Configuration for [www.calipnos.com](#)

Reload:

[Reload www.calipnos.com](#)

DNS setup:

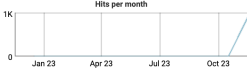
How to point your domain at your website.

CNAME: [webapp-33697.eu.pythonanywhere.com](#)

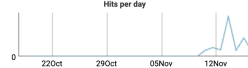
Traffic:

How busy is your site?


Hits per month




Hits per day



Hits per hour



Hits per minute



Code:

What your site is running.

Source code: [/home/calipnos/dashboard](#) [Go to directory](#)

Working directory: [/home/calipnos/](#) [Go to directory](#)

WSGI configuration file: [/var/www/www_calipnos_com_wsgi.py](#)

Python version: 3.9

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[/home/calipnos/virtualenvs/myenv](#)

[Start a console in this virtualenv](#)

Log files:

The first place to look if something goes wrong.

Access log: [www.calipnos.com.access.log](#)


Error log: [www.calipnos.com.error.log](#)

Server log: [www.calipnos.com.server.log](#)

Log files are periodically rotated. You can find old logs here: [/var/log](#)

Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delete
/static/	/home/calipnos/dashboard/static	
Enter URL	Enter path	

Security:

An HTTPS certificate is needed so that people can access your site securely. You can get a free one or configure your site with one you have bought elsewhere below.

HTTPS certificate: [None](#)

You need to **Reload your web app** to activate any changes made below.

Forcing HTTPS means that anyone who goes to your site using the insecure http URL will immediately be redirected to the secure https one. [More information here](#).

Force HTTPS: [Disabled](#)

Password protection is ideal for sites that are under development when you don't want anyone to see them yet.

Password protection: [Disabled](#)

Username: [Enter a username](#)

Password: [Enter a password](#)

Disable:

You can temporarily turn off your web site by disabling it. You can always re-enable it later.

[Disable webapp](#)

Delete:

Deleting this web app will remove your web app from the Internet, but will not remove your code. The WSGI configuration file will be backed up.

[Delete www.calipnos.com](#)

Copyright © 2011-2023 PythonAnywhere LLP — [Terms](#) — [Privacy & Cookies](#)

Page d'administration de la webapp sur pythonanywhere

Page 3

2. Description de l'application

2.1 Organisation du projet

L'application possède deux pages principales :

- La page authentification qui permet à un utilisateur de s'enregistrer ou de se connecter
- La page d'affiche des résultat sur laquelle l'utilisateur peut visualiser ses données issues de ses fichiers csv.

Les codes Python de ces deux pages sont respectivement situés dans les fichiers `/dashboard/client/views/auth_view.py` et `/dashboard/client/views/results_view.py`.

Les codes HTML de ces deux pages sont situés dans le répertoire `/dashboard/client/template`.

2.2 Développement local

L'application est compatible Python 3.8 et Python 3.9. La version sur PythonAnywhere est 3.9.

Pour développer en local, il est nécessaire d'installer les librairies sur lesquelles dépend le projet. Ces dernières sont listées dans le fichier `/dashboard/requirements.txt`.

Pour les installer, vous pouvez soit le faire directement sur votre poste avec pip :

```
$ pip install -r requirements.txt
```

ou alors en utilisant un environnement python virtuel (venv) situé à la racine

```
$ pip install virtualenv
$ python -m venv myenv
$ source myenv/bin/activate
```

Ensuite il suffit de se rendre dans le répertoire `/dashboard` et de faire :

```
$ python manage.py runserver
```

L'application locale sera disponible à l'adresse <http://127.0.0.1:8000/>

3. Utilisation de l'application web

3.1 Authentification

Il faut s'identifier pour accéder à la page de visualisation. Il est possible pour un nouvel utilisateur de s'enregistrer en cliquant sur « *S'enregistrer ici* ».

3.2 Visualisation des données

La page de visualisation des données permet de déposer son fichier csv et ensuite afficher les graphiques issus de son traitement.

Pour déposer un nouveau fichier, cliquez sur « *Déposer* » puis sélectionner le fichier csv.

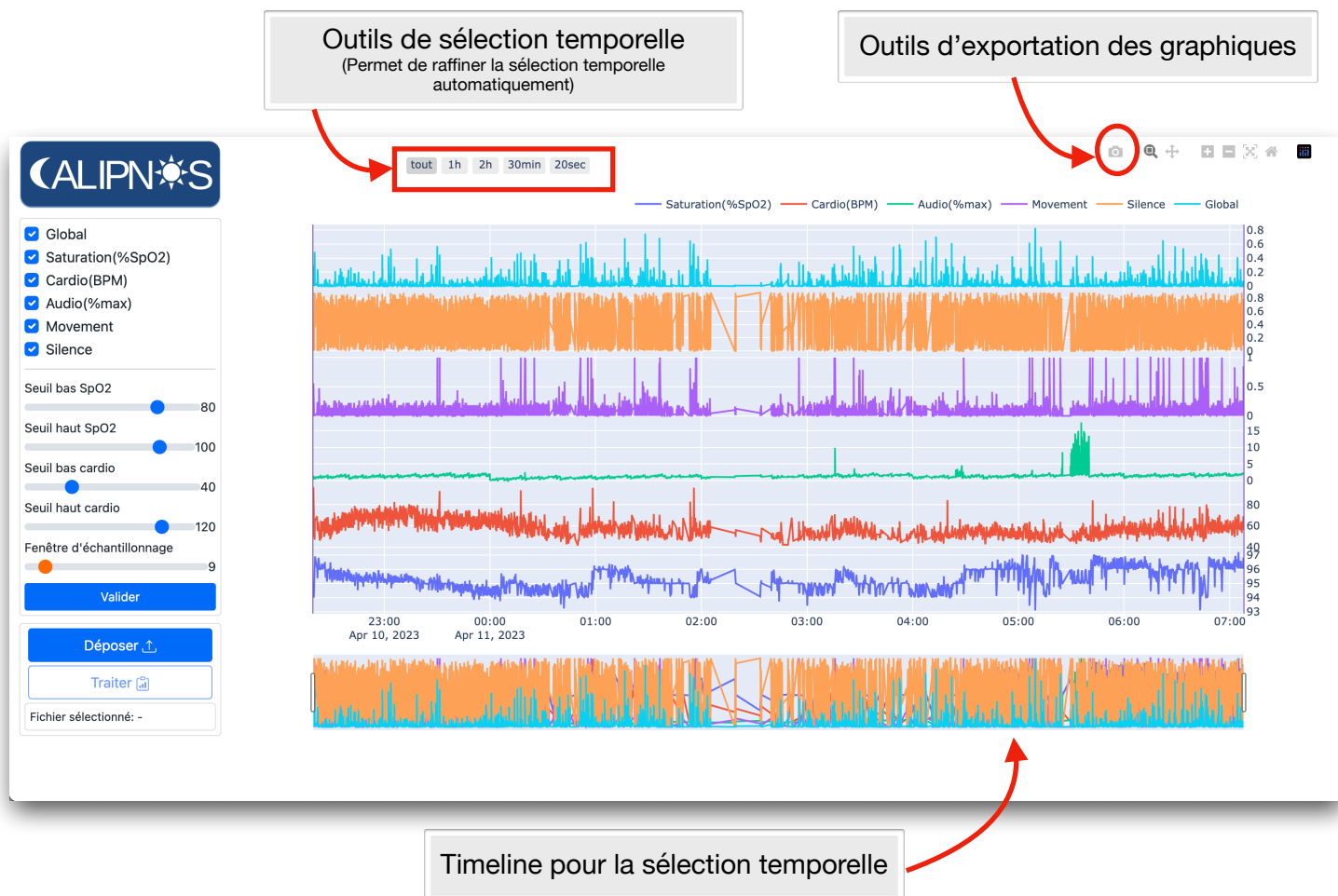
Pour traiter le fichier et afficher les graphiques, cliquez sur « *Traiter* ».

Pour changer les paramètres d'affichage vous pouvez cocher les métriques à afficher et changer les seuils de certains des paramètres de traitement. Une fois ces derniers ajustés, cliquez sur « *Valider* » pour relancer la visualisation avec les nouveaux paramètres.

La fenêtre d'échantillonnage correspond au nombre de point utilisés dans la fenêtre glissante de traitement des différents signaux.

Pour se déplacer dans les graphiques vous pouvez utiliser la ligne temporelle en bas pour se déplacer dans le temps. Vous pouvez définir une fenêtre de visualisation en la sélectionnant directement sur les graphiques. Un double clic permet de réinitialiser la fenêtre de visualisation. Il est également possible d'ajuster directement la fenêtre de visualisation avec des valeurs pré-définies de 1h, 2h, 30min, 20sec en haut à gauche.

Pour exporter les graphiques, cliquez sur l'icône photo en haut à gauche en passant la souris sur le graphique.



4. Note à propos des fichiers csv à importer

Les fichiers csv déposés doivent répondre à certains critères que j'ai pu identifier en utilisant le jeu de données initialement fourni :

1. Les noms des colonnes ne doivent pas se répéter dans le fichier csv (i.e il doit y avoir une seule en tête et elle doit être au début du fichier)

Par exemple ce fichier csv ne convient pas car le nom des colonnes se répète :

```
Index;Date;Time;Temperature (°C) ;Cardio (BPM) ;Saturation (%SpO2) ;Audio (%max)
;Rotation (rad/s) ;Acceleration (m/s2)
1;16/03/2023;22:05:58;22.60;0.00;0;0.00;0.19;10.23
2;16/03/2023;22:06:00;22.75;11.24;0;3.07;0.55;11.79
3;16/03/2023;22:06:02;22.88;76.86;112;3.06;0.66;10.36
4;16/03/2023;22:06:04;22.98;60.19;112;3.12;0.80;10.99
5;16/03/2023;22:06:06;23.02;41.98;93;2.91;0.80;11.01
6;16/03/2023;22:06:08;23.07;75.55;94;3.19;0.58;12.16
7;16/03/2023;22:06:10;23.15;63.51;94;3.02;0.20;10.39
Index;Date;Time;Temperature (°C) ;Cardio (BPM) ;Saturation (%SpO2) ;Audio (%max)
;Rotation (rad/s) ;Acceleration (m/s2)
8;16/03/2023;22:06:12;23.20;40.67;93;3.11;0.23;10.63
9;16/03/2023;22:06:14;23.23;33.60;93;3.44;0.05;10.62
10;16/03/2023;22:06:16;23.27;56.32;93;3.26;0.06;10.62
11;16/03/2023;22:06:18;23.31;30.25;0;2.77;0.07;10.62
12;16/03/2023;22:06:20;23.36;0.00;0;3.23;0.51;10.68
13;16/03/2023;22:06:22;23.41;64.94;0;2.89;0.15;10.34
```

2. Les colonnes des valeurs de Temperature, Cardio, Saturation, Audio, Rotation, et d'Acceleration seront reconnues tant que les noms des colonnes comportent comme trois premières lettres celles des métriques citées. Par exemple : « Car » suffira pour la colonne Cardio ou bien « Sat(SpO2) » suffira pour la colonne Saturation.