
Projet 2

Site internet Flask

Auteurs :

Adrien MONTARIOL

Giannis MATROZOS

Arthur WERY

Groupe C.6

Référent :

Ayham KASSAB

Table des matières

1	Introduction	1
2	Python	1
3	SQL	1
4	HTML - CSS	1
5	Javascript	2
6	Gestion et cohésion d'équipe	3
7	Conclusion	3

1 Introduction

Dans ce rapport vous retrouverez toutes les étapes par lesquelles nous sommes passés pour la réalisation de notre site permettant des visualisations de statistiques Inginious. Ce rapport est divisé en 4 grandes parties : Python, SQL, HTML et Javascript. On vous expliquera également la manière dont on a travaillé en groupe, comment nous nous sommes organisés.

2 Python

Nous avons utilisé la partie python pour faire interagir Flask avec la base de données, ainsi, avec le module sqlite3, flask et random, nous pouvons donner l'instruction de faire apparaître les graphiques de notre choix. Quand l'utilisateur sélectionne un graphique dans un drop-down menu, par exemple, l'instruction est donnée au programme python qui renvoie le ou les graphiques demandés. Pour ce faire, nous avons séparés tous les types de graphiques en fonctions différentes, et pour chaque graphique, on appelle une certaine fonction avec un certains paramètres.

Cet exemple de code ci dessous permet de récupérer la base de données en python et d'en faire des dictionnaires.

```
1 def get_db():
    db = getattr(g, '_database', None)
3     if db is None:
        db = g._database = sqlite3.connect(DATABASE)
5     db.row_factory = make_dicts
    return db
```

3 SQL

Nous avons utilisé de nombreuses requêtes SQL afin de tirer les données de la DB.

Par exemple dans le code ci-dessous, nous demandons à la DB de nous renvoyer les tâches pour le cours "x" où la valeur de "succeeded" vaut True. Nous demandons aussi de formater le renvoi d'une certaine manière facilitant l'affichage pour la suite.

```
@app.route('/percent')
2 def percent():
    return jsonify({'percent': query_db(
4         "select task, count(*) as val from user_tasks where task in (" + request.args.get('
            tasks', '') + ") and succeeded='true' group by task order by 2 DESC;"
    )})
```

4 HTML - CSS

Pour créer notre site, nous avons bien sûr utilisé l'HTML :5 avec un framework réputé intitulé "Bootstrap" ainsi que le CSS. La plus grande partie de notre HTML est utile pour la page d'accueil, le reste des pages est fait principalement en JS. Nous avons également fait le footer en HTML. Le CSS nous a permis de donner ce style très épuré au site afin qu'il soit le plus clair possible pour l'utilisateur.

```

1 <div id="menu1" class="tab-pane fade">
    <div class="chart-container" >
3       <canvas id="myChart1"></canvas>
    </div>
5 </div>

```

```

1 body {
    background: #FFFFFF;
3    font-family: 'Work Sans', sans-serif;
    margin: 2rem;
5 }

```

5 Javascript

Une grande partie du projet est faite avec JavaScript. Quand l'utilisateur interagit avec le site, le JavaScript envoie à python des instructions, python interagit, avec ces instructions, avec la base de données, et ensuite renvoie au JavaScript les données des graphiques. Le menu déroulant est fait en JavaScript, cela permet d'avoir des modifications sur le site en temps réel quand on modifie la base de donnée.

Avec chartJS, nous faisons différents types de graphiques. Premièrement les graphiques en double-bâtonnet (chart1), des graphiques en toile d'araignée (chart2), et des graphiques en bâtonnet (chart3)

Dans l'exemple ci dessous, nous voyons comment ChartJS est implémenté. Ce graphique sera un graphique en batonnets horizontal. Nous voyons aussi

```

1 function chart3() {
    var ctx3 = document.getElementById('myChart3').getContext('2d');
3
    var getData1 = $.get('/try');
    getData1.done(function (avgtries){
5        var myChart3 = new Chart(ctx3, {
7            type: 'horizontalBar',
            title: 'Inginious',
9            data: {
                labels: avgtries.avgtries.map(r => r.lbl),
11               datasets: [{
                    label: 'Average number of tries',
13                    data: avgtries.avgtries.map(r => r.val),
                    backgroundColor: '#50b1e4',
15                    borderColor: '#50b1e4'
17                }
                ]
19            }
21        });
23    })
}

```

6 Gestion et cohésion d'équipe

Pour le travail d'équipe, nous avons un planning pour nous organiser, nous communiquons via différents canaux de discussion tel que Discord (Vocal) et Messenger (Message). On s'entend tous les 3 biens et nous somme travailleurs donc la cohésion d'équipe était notre point fort. On se partage beaucoup d'idées entre nous, et on les écrit quelque part si elles sont intéressantes. On s'est souvent divisés le travail et réparti les tâches afin d'aller plus vites, mais on passait à chaque fois derrière les autres pour être tous les 3 en accord sur nos avancées sur le projet.

7 Conclusion

Ce travail nous a permis de nous familiariser encore plus avec les modules Flask et chart.js, effectivement, ça peut nous être très utile par la suite. Ça nous a permis de nous rendre vraiment compte de ce qu'est un projet de plus grande ampleur. Ce projet nous a également beaucoup forgé au niveau de la cohésion de groupe et comment travailler ensemble, c'est une très bonne expérience que l'on a passé en tant que programmeur peu expérimenté!