

## *Projet Base de donnée/Flask*

### Sommaire :

#### 1.Le site:

- 1a: Explication du projet*
- 1b: Les différentes pages/Possibilités*
- 1c: Remarque*

#### 2.La base de données:

- 2a: Schémas relationnelle*
- 2b: Les différentes Tables et leurs fonctionnalités*

#### 3.Le serveur:

- 3a: Fonctionnement*
- 3b: Fonction importante*
- 3c: Requête SQL importante*

#### 4.Conclusion:

- 4a: Difficulté*
- 4b: Amélioration possible*
- 4c: Répartition du travail*
- 4d: Bilan*

## 1. Le site:

### *1a: Explication du projet*

Notre site consiste en une sorte de base de données de Pokémon. Celle-ci sera équipée de plusieurs fonction permettant l'upload de pokémon, capacités, talents et de pouvoir aussi assigner des évolutions. Il permet aussi de visionner tout les talents/capacités enregistré ainsi que les pokémon qui les possèdent.

### *1b: Les différentes pages/Possibilités*

Toutes les pages ont en commun 2 choses :

- Le logo Pocket Monster, qui renvoie automatiquement sur l'index du site.
- Une barre contenant 4 boutons, ceux-ci renvoient sur 4 pages : le pokédex, le menu d'upload, les capacités et les talents

Donc parmi les pages nous avons :

- '/' L'index, à chaque lancement de cette page un nouveau pokémon aléatoire apparaît, celui-ci est cliquable.

- '/PKM?id=' avec l'id du pokémon choisis derrière. Affiche les infos du pokémon. Parmi les choses cliquables, il y a les images des évolutions qui renvoient sur la page du pokémon en question, le talent qui

affichera la page du talent avec tout les pokémon le possèdent et les capacités , qui comme pour les talents,affichera la page de la capacité.

- '/capacite' , affiche un tableau avec toute les capacités, toutes sont cliquables et renvoie une page dédié à celle ci.

- '/capacitePKM?id=' suivi de l'id de la capacité, affiche un tableau de tout les pokémon la possèdent, bien évidemment tout les pokémon sont cliquable et renvoie à leur page.

- '/talent', affiche la liste de tous les talents , cliquer sur ceux ci vous renvoie à la page du talent.

- '/talentPKM?id=' avec le nom du talent , renvoie la liste de tout les pokémon ayant ce talent.

- '/upload' menu de l'upload , propose 7 type d'interaction.

- '/uploadtalent' et 'uploadcapacite', ces deux pages propose plusieurs champs pour ajouter un(e) talent/capacité , si l'un des champs est vide, renvoie la même page avec un message d'erreur.

- '/uploadPKM' , permet d'ajouter un pokémon, petit différence avec les deux pages d'avant, il est nécessaire d'avoir une image 256x256. Et lui aussit renvoie la même page avec une erreur si un des champs est vide.

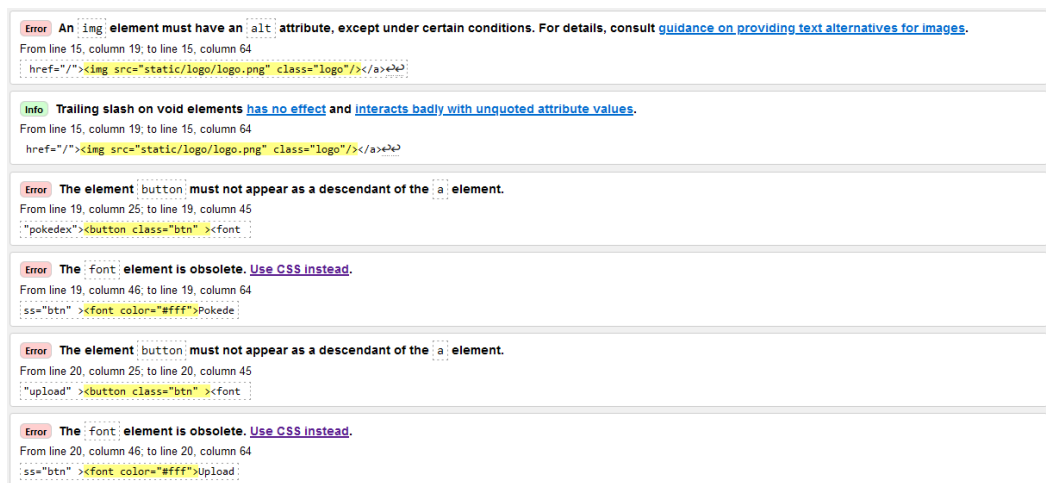
-'/assignerTalent' et '/assignerCapacite', dans la première case choisissez un pokémon puis dans la seconde un(e) capacité/talent. Une fois le tout ajoutée , la page redirige vers l'index

-'/assignerSOUS' et 'assignerSUR' , dans le premier champs choisissez le pokémon puis dans le second , son évolution ( précédent ou suivant ) et redirige vers l'index

### 1c: Remarque

*Comme demandé nous avons passer nos pages html dans le w3c validator. Il nous a renvoyé plusieurs erreurs que nous avons corrigé , malgré tout il nous à renvoyer des erreurs qui , si on les corrigeai , allait rendre le site beaucoup moins agréables parcourir.*

*Les voici :*



**Error** The element `button` must not appear as a descendant of the `a` element.  
 From line 21, column 27; to line 21, column 46  

```
apacite"><button class="btn"><font
```

**Error** The `font` element is obsolete. [Use CSS instead.](#)  
 From line 21, column 47; to line 21, column 65  

```
ass="btn"><font color="#fff">Capaci
```

**Error** The element `button` must not appear as a descendant of the `a` element.  
 From line 22, column 25; to line 22, column 44  

```
"talent"><button class="btn"><font
```

**Error** The `font` element is obsolete. [Use CSS instead.](#)  
 From line 22, column 45; to line 22, column 63  

```
ass="btn"><font color="#fff">Talent
```

**Error** Misplaced non-space characters inside a table.  
 From line 41, column 9; to line 43, column 39  

```
</tr><tr style="border: 3px solid black;">
```

**Fatal Error** Cannot recover after last error. Any further errors will be ignored.  
 From line 41, column 9; to line 43, column 39  

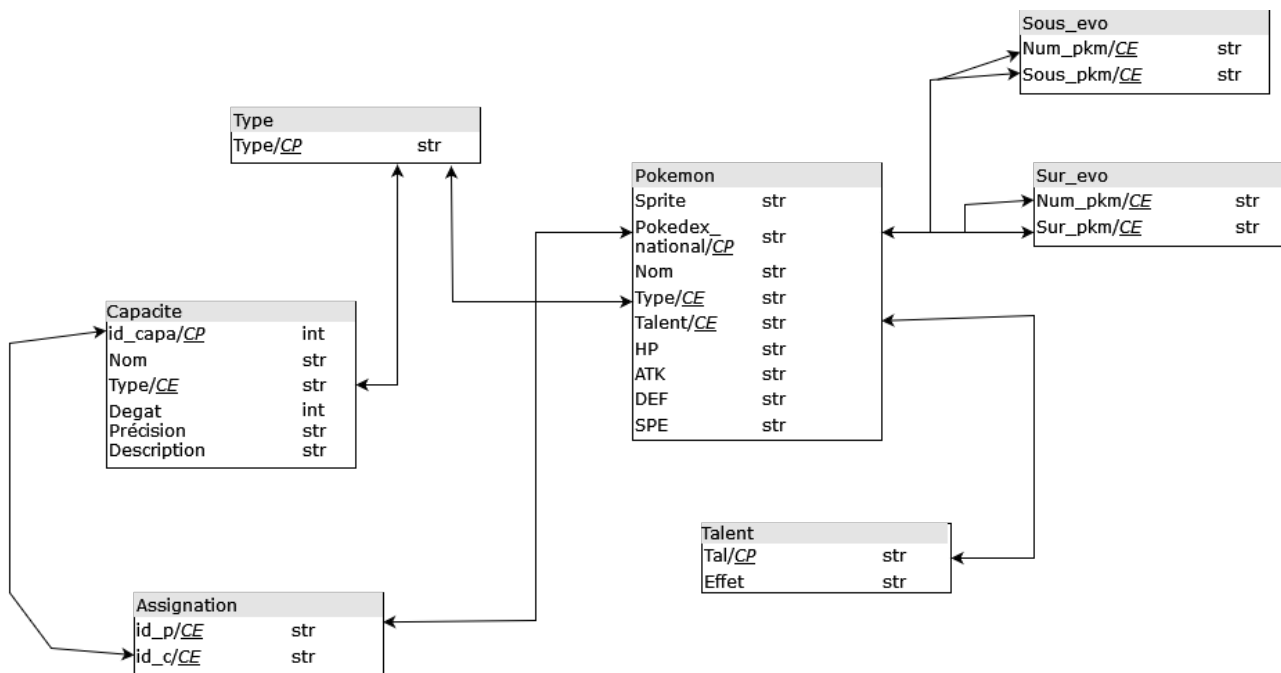
```
</tr><tr style="border: 3px solid black;">
```

Nous avons donc choisi de garder les balises `<font>`, de mettre les `<buttons>` à l'intérieur des `<a>` car si nous les avons mise dans le sens inverse , seul l'écriture et non le bouton serait cliquable.

De plus , nous avons fourni dans un dossier test tout ce qui est nécessaire pour faire des test avec chacun des types d'upload.

## 2.La base de données:

### 2a: Schémas relationnelle



### 2b: Les différentes Tables et leurs fonctionnalités

Voici un résumé des tables les plus importantes :

- Pokemon, cette table contient toute les info importantes du pokémon. L'attribut sprite consiste au chemin de l'image.

- Les tables Assigination, Sous\_evo et Sur\_evo fonctionne de la même manière, dans la première ligne il y a l'id du pokémon et dans la seconde l'id de la capacité/évolution

### 3.Le serveur:

#### *3a: Fonctionnement*

Avant de lancer le serveur , il est important d'avoir entrée les 6 commandes suivantes dans le shell :

- pip install flask
- pip install flask\_uploads
- pip install werkzeug==0.16.1
- pip install Flask==1.1.4
- pip uninstall markupsafe
- pip install markupsafe==2.0.1

Le module flask\_uploads utilise des anciennes versions de certains modules, donc pour le bon fonctionnement du serveur il est nécessaire de les importées.

Une fois le serveur lancée , il sera présent sous l'ip suivante : <http://127.0.0.1:5000>

#### *3b: Fonction importante*

Dans ce projet , il y a deux fichier python. Le premier bd.py s'occupe des interactions avec la base de donnée . Il est possible via le shell de faire tout les fonction présent dans le menu upload. Parmi les plus importante , je pense à infoPKM(id) et evolution(id) qui permette d'avoir plein d'information, il y aussi ajoutpkm(...), à noté que pour l'utiliser depuis le shell il

faut rentrer pour le sprite le nom du fichier plus ce chemin 'static/pkm/'.

Et le second fichier python est serv.py qui s'occupe uniquement de la partie serv en appelant les fonctions de l'autre fichier. Les fonction reste plutôt simple d'un point de vue algorithmique à pars peut être uploadPKM().

### *3c: Requête SQL importante*

```
-'SELECTCapacite.id_capa,Capacite.Nom,Capacite
.Type,Capacite.Degat,Capacite.Précision,Capacite.
Description FROM Pokemon JOIN Assigation ON
Pokemon.Pokedex_national = Assigation.id_p JOIN
Capacite ON Assigation.id_c = Capacite.id_capa WHERE
Pokedex_national= ? ;'
```

Par exemple cette requête permet d'obtenir toute les capacité d'un pokémon en fonction de son id.

```
-"INSERT INTO Pokemon (sprite,Pokedex_national,
Nom,Type,HP,ATK,DEF,SPE) VALUES (?, ?, ?, ?, ?, ?, ?, ?)"
```

Celle ci par exemple permet d'insérer un pokémon.



## 4. Conclusion:

### *4a: Difficulté*

La première difficulté qui me vient en tête est la mise en place du l'upload d'image. Trouvé les bonnes versions de chaque modules a été plutôt long , sa va aussi pour sa compréhension malgré que cela représente que quelque ligne.

La partie interactions avec le base de donnée n'a pas été très compliqué, même chose pour la mise en place du serveur flask.

### *4b: Amélioration possible*

Parmi les idées qu'on a eu mais qu'on a pas pu exécuter par manque de temps , je pense à toute la partie suppression. On avait comme idée de pouvoir faire une versions inverse de chaque fonction du menu upload. J'aurais aussi voulu pourvoir arranger plusieurs fonction pour éviter qu'il y est des actions imprévus. J'aurais aussi aimé intégrer plus de jinja dans le html, à mon avis il aurait été possible de rendre les pages plus dynamique avec ce langage. J'aurai aussi aimer utiliser du javascript mais encore une fois un manque de temps.

### *4c: Répartition du travail*

*Arthur → Toute la partie html/css*

*Kyllian → Serveur/python et gestion de liste dans le html*

#### *4d: Bilan*

Un projet réellement intéressant , comme durant chaque projet , on apprend énormément. La première fois qu'on s'intéresse autant à la partie serveur. Pas mal de regret , on aurait aimé aller plus loin avec certaines parties. Notamment la partie visuelle qui reste peu convaincante.