

29 Novembre 2021
04 Février 2022

Rapport de stage :

Développeur Django Python dans le cadre du projet de Machine
Learning :
Détection de fraude bancaire en ligne



Auteur : Léo GUIRADO

Enseignant référant : Pierre Lefèbvre

Etablissement : Centre GRETA MTE 77 Lognes – Espace DIDEROT

Tuteur : Mr James KOUTHON

Entreprise : ACENSI Finance

14 Rue du Général Audran, 92400 Courbevoie, France

Sommaire

Remerciements

Introduction

Partie I – L'entreprise et son secteur d'activité : ACENSI

A- Présentation de l'entreprise

1- L'entreprise et son secteur d'activité

2- La concurrence dans ce secteur

B- ACENSI Finance

Partie II – Définition du poste

Partie III – Développement du stage

A- S'auto-former

B- Django

1- Son histoire

2- Son fonctionnement

3- Son ORM

4- Sa Structure

5- Python et ses librairies

Conclusion

Annexes

Remerciements :

Je remercie toutes les personnes qui m'ont aidé de près ou de loin lors de ce stage grâce à qui j'ai pu avoir une vision plus globale du métier de développeur.

En premier lieu, je souhaite remercier Mr Maxime GUIRADO pour sa précieuse aide à la recherche de stage, ses conseils et aides de développeur et son soutien inconditionnel dans la famille.

Je tiens à remercier Mr James KOUTHON, mon Tuteur durant ce stage, pour m'avoir donné cette magnifique opportunité, l'autonomie et la liberté d'un employé, tout en me procurant toutes les ressources et directives dont j'avais besoin.

Je remercie Mr Benjamin MARTY pour son aide sur ce projet, tant au niveau codage et connaissances que pour m'avoir guidé dans l'entreprise.

Je voudrais aussi remercier tout particulièrement ma famille qui m'a aidé, soutenu et inspiré durant tout ce temps.

Introduction :

Dans le contexte actuel, les gens font de plus en plus d'achats en ligne (gain de temps, prix, Covid, etc.), ce qui favorise la fraude bancaire.

Le but de ce projet est de faire évoluer la détection de ces fraudes.

L'intérêt étant d'avoir toujours autant de sécurité pour moins de confirmations à faire avant la validation d'achat.

Pour cela, le Machine Learning a été utilisé.

Partie I – L'entreprise et son secteur d'activité :

A-Présentation de l'entreprise

1- L'entreprise et son secteur d'activité

ACENSI SAS (société par action simplifiée), créée en juin 2003 à Suresnes par Mr David COMBLE et Mr Thibault Blin, est un groupe spécialisé depuis sa création entre le cabinet généraliste et l'ESN spécialisée.

Avec une croissance moyenne annuelle de 20% et 180 Millions de chiffre d'affaires, ACENSI compte 1500 collaborateurs et est implanté dans 8 villes en France et à l'international comme aux USA, au Canada, en Belgique ou à l'Île Maurice pour plus de 90 clients.

ACENSI aujourd'hui, c'est ACENSI SAS, ACENSI Finance, ERES Technologie, ACENSI Sana et ACENSI Technology Services.

Chacun de ces pôles à ses propres missions mais partagent tous le même intérêt pour l'osmose avec un environnement sain et durable, le bien-être de ses salariés, l'égalité femmes-hommes et contribuent à l'insertion des personnes en situation de handicap dans l'entreprise.

2- La concurrence dans ce secteur

Dans le secteur des entreprises de services du numérique, la concurrence est importante. Pour autant, lorsque que l'on regarde le classement annuel des ESN, nous pouvons voir que ACENSI est toujours dans le top 50 des ESN Françaises.

B-ACENSI Finance

Mon stage a eu lieu dans le Service Finance. Son nom fait allusion à la spécialité de l'ESN pour des clients comme les banques.

Au siège, les personnes de ce pôle qui sont sur place sont le directeur technique, les "intermissions" et les stagiaires.

Le travail sur place se trouve dans l'immeuble le Black Pearl, un bâtiment en triangle, recouvert d'une armature noire, son toit est une terrasse laissant un panorama impressionnant entre les immenses buildings de la défense, la Seine ou encore la tour Eiffel, le tout recouvert d'un dôme de cuivre.

C'est donc à cet endroit que, d'un œil expert, le directeur technique passe, chaque jour de présence, de collaborateur en collaborateur, mettant en évidence des détails qui poussent à la réflexion et donnant des directives toujours pleines de lumière, toujours à l'écoute d'une quelconque explication ou incompréhension du code ou du projet mis en place dans le but de former ces personnes.

Pour ces collaborateurs comme pour les ceux en poste dans d'autres entreprises, le travail a subi quelques changements dû au covid et aux restrictions. Le télétravail à pris une place dominante dans l'entreprise, ce qui n'à pas dû être facile pour tout le monde puisque personne n'y était préparé.

C'est dans ce contexte que mon stage s'est déroulé, un mélange de travail en entreprise avec le contact visuel et de télétravail où je me retrouvais faussement livré à moi-même.

Partie II – Définition du poste

L'objectif des 10 semaines qui ont constituées mon stage a été d'occuper le poste de Développeur Django Python pour le projet « Détection de la fraude bancaire avec apprentissage automatique et la théorie des jeux ».

Ce dernier est un projet sur le temps afin de former des personnes au Machine Learning et au Big Data.

Pour aider les développeurs du projet à faire leurs présentations aux directeurs responsables du projet, un site de présentation a été mis en place.

Ce site contient des images, des tableaux, des graphiques et permet de lancer les algorithmes de détection.

La mission du poste est la réécriture d'un outil de présentation des résultats de recherche sur la détection de fraude bancaire.

Pour cela, il était entendu qu'il faudrait :

- Une prise en main du code source existant
- Une refactorisation du code existant
- Un ajout de nouvelles fonctionnalités concernant la théorie des jeux

Et si le temps le permettait, une modification de l'apparence serait un plus.

Partie III – Développement du stage

A- Se former

Pendant les 10 premiers jours, j'ai commencé par m'auto-former sur plusieurs sujets dont mon Tuteur m'a fait part :

- Python, pour bien assimiler la POO.
- Django, un Framework du langage Python dont le slogan est : "le Framework pour les perfectionnistes avec des deadlines".
- Le Virtual Environnement de python, pour isoler la version de python, les bibliothèques et les scripts installés.
- Le CRUD, désignant les quatre opérations de base des bases de données.
- HTML et CSS, les "langages" indispensable aux sites web.
- Anaconda, en particulier Jupyter Notebook, une application web qui permet de coder d'une façon totalement différente d'un IDE normal comme PyCharm ou Visual Studio Code.
- Git et GitLab, solution de versioning qui permet de stocker vos projets sur place ou à distance. La version communautaire de GitLab se nomme GitHub.

La liste a continué de s'allonger au fur et à mesure des conversations tout au long du stage sans se limiter à la programmation en elle-même ou à celle de mon projet :

- Teams, outil de communication, indispensable lors du travail à la maison.
- Intracens : le site intranet pour valider ses jours de travail et contacter les différents services afin d'obtenir différentes choses : une carte d'accès à l'immeuble, un ordinateur portable pour le télétravail, l'accès au wifi, etc...
- JavaScript, un langage de programmation web utilisé pour du FrontEnd, même s'il est possible de l'utiliser en BackEnd aussi.
- Le Design Pattern, une bonne pratique de la POO, pour faire face aux problèmes récurrents en programmation.
- SOLID, cinq principes de conception pour coder en POO de manière plus compréhensible, flexible et maintenable.
- Les Cheats Sheet, ou anti-sèche pour les langages de programmation. Evite de perdre trop de temps à chercher partout sur le web alors que tout est dessus.

B- Django

"Le framework pour les perfectionnistes avec des deadlines."



Django est un cadre de développement (ou Framework) web open source Python. Son but est de rendre simple et rapide la création de sites web.

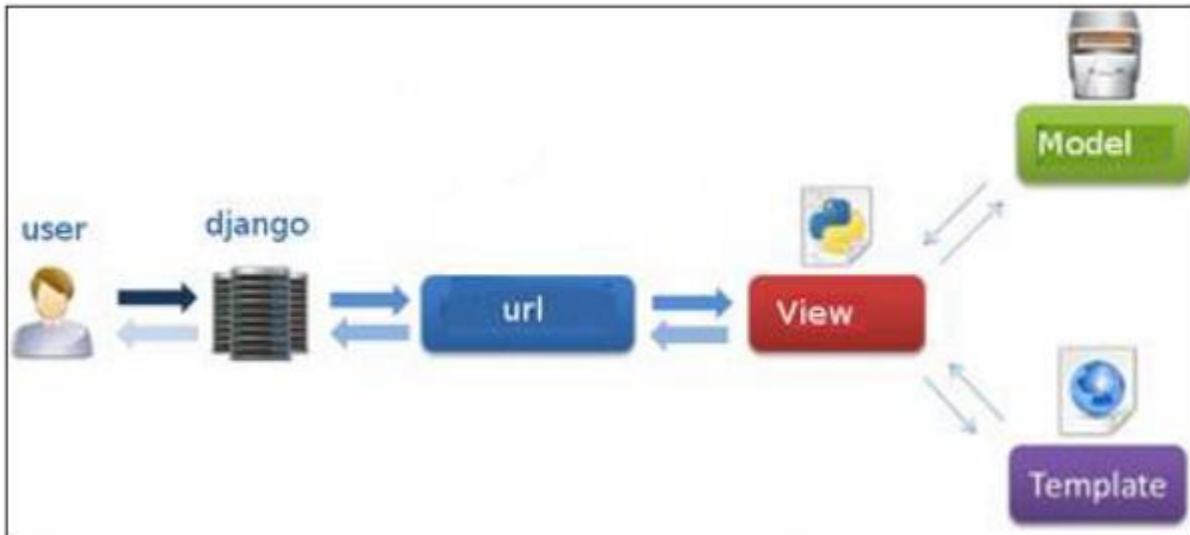
1- Son histoire

Ce cadre a été développé pour un journal local de la ville de Lawrence, dans le Kansas en 2003.

Ses créateurs , Adrian Holovaty et Simon Willison, ont eu pour idée de permettre le développement de sites web complets, interfacés avec une base de données, dans un laps de temps très court.

2- Son Fonctionnement

Django fonctionne avec un MVT (Model,View,Template) qui n'est d'autre qu'un MVC.

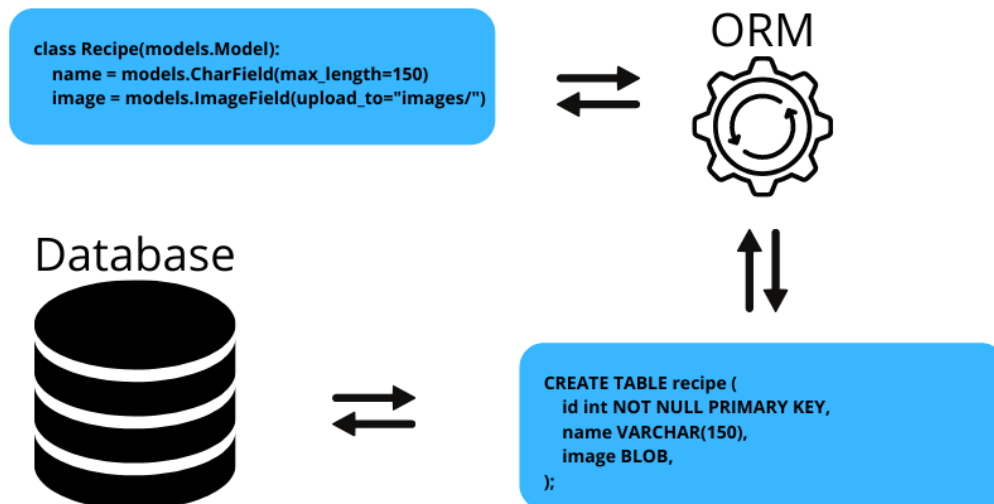


- L'utilisateur envoie une action.
- L'action arrive à View demandée via le routeur "urls.py".
- Ce dernier, selon la demande, pourra faire une requête à Model qui lui renverra des informations et les enverra à Template pour modéliser la page.
- Ensuite, cette page est renvoyée à View pour être transmise au client.

3- Son ORM

Object-Relational Mapping.

L'ORM est un ensemble de classes permettant de manipuler les tables d'une base de données.

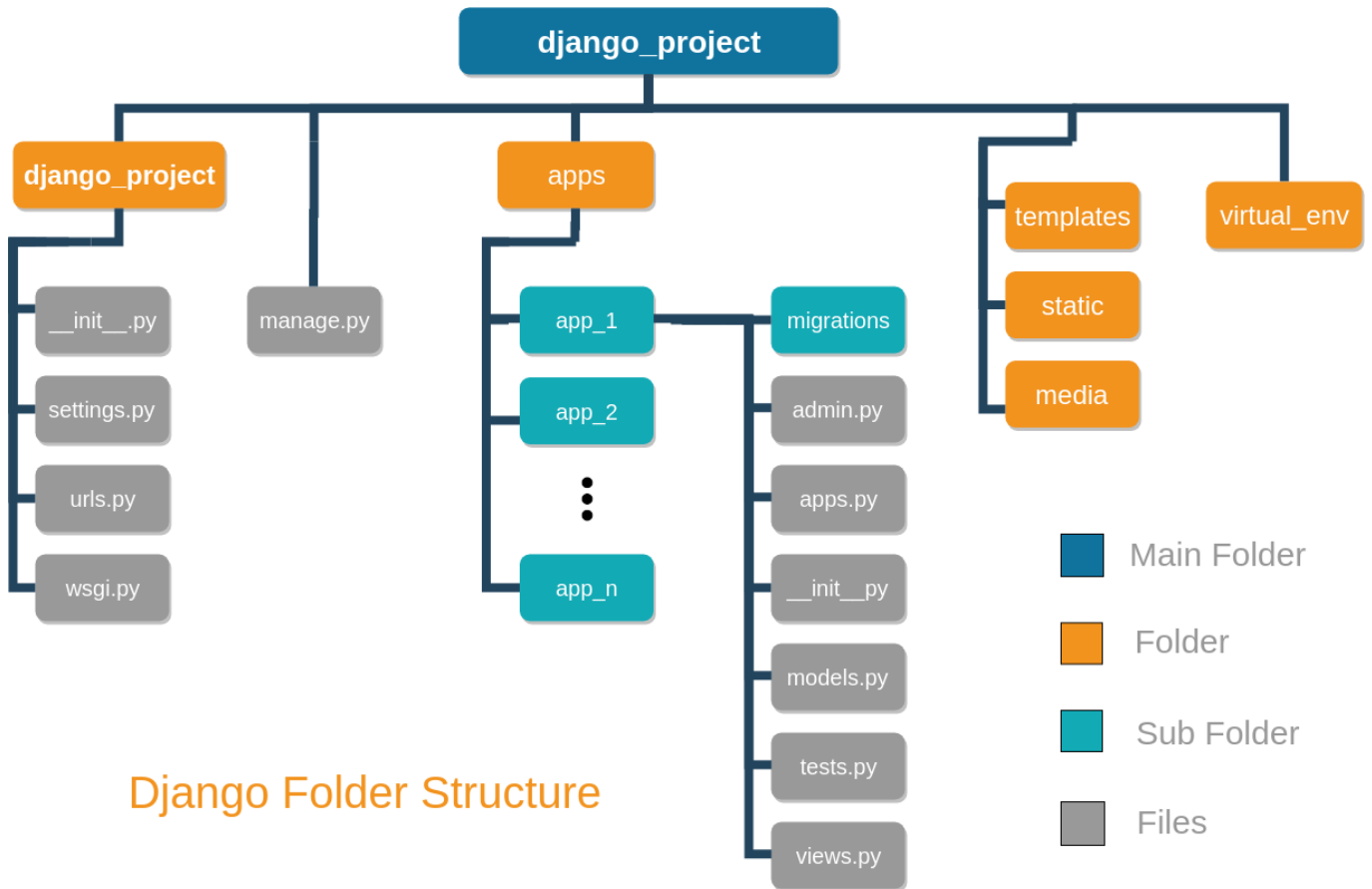


Il va se placer entre le programme ("Models.py") et la base de données utilisé, ce qui nous permet de coder en python.

La structure générale du code reste la même bien que nous n'ayons plus à utiliser de requête, nous pouvons nous atteler directement au code en lui même.

4- Sa Structuration

Cette structure peut paraître effrayante pour un débutant mais avec le temps



Django Folder Structure

- Virtual env ou `.env` : isolement de python, des librairies et scripts utilisé pour chaque projet.
- Templates : les pages web dans lesquels l'utilisateur naviguera.
- Static : les langages annexes (CSS, JS) et les images dont le développeur se servira dans l'application.
- media : les fichiers uploadés par l'utilisateur

(ces 3 derniers, bien codés, peuvent être réutilisables à volonté.)

- Apps : pour chaque application sa gestion de :
 - Migrations (cf : annexe)
migration du code utilisé dans `Models.py` afin de créer une/des table(s) de données.
 - Admin référence les tables.
 - Apps : registre des applications installées.
Je ne l'ai pas utilisé.

- __init__.py : fichiers nécessaires pour que Python considère les dossiers comme contenant des paquets.

Je ne l'ai pas utilisé.

- models contient toutes les classes de table de données.
- tests est utilisé pour les tests unitaires.
- views contient la logique nécessaire pour renvoyer une réponse à la demande de l'utilisateur.

- Manage est un utilitaire pour les tâches administratives.

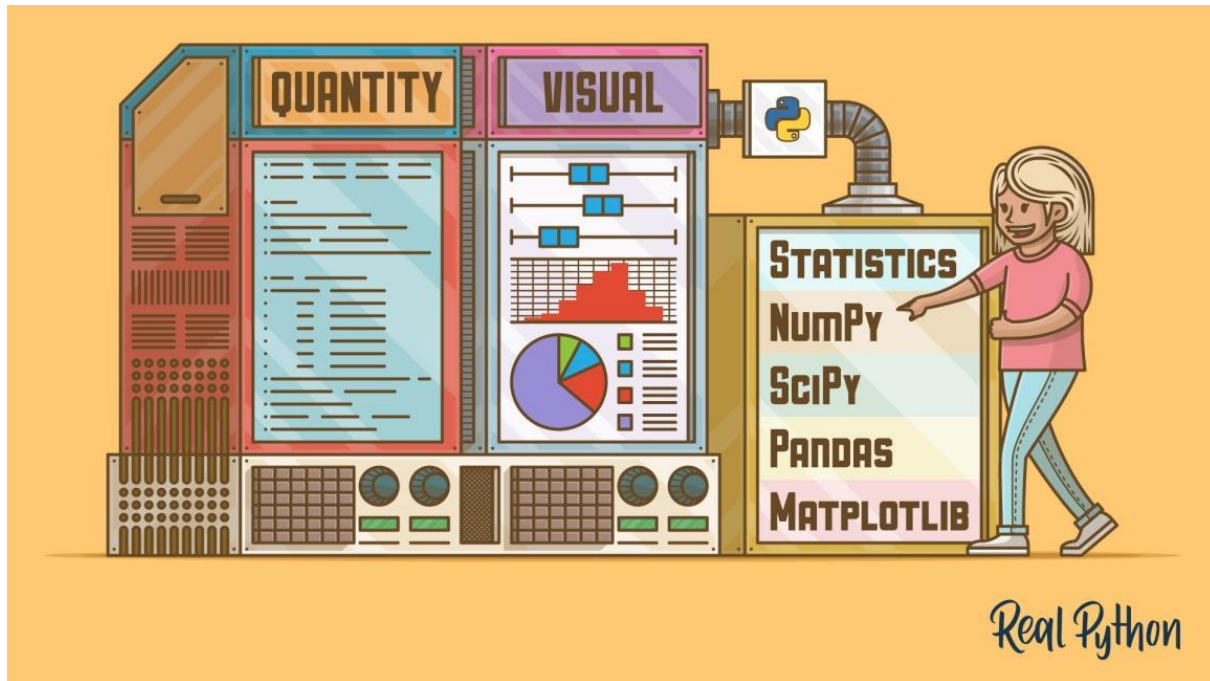
Je ne l'ai pas utilisé.

- Django project (ou fraude dans le cadre de mon stage)

- Settings contient toutes les configurations de Django.
- urls défini les chemins d'accès.
- wsgi dialogue avec le serveur pour accueillir la requête HTTP.

5- Python et ses Librairies

Comme tous les langages, python peut utiliser des libraires (ou bibliothèques) pour rendre son code plus simple et ne pas avoir à réinventer la roue.



- NumPy et SciPy pour le calcul scientifique.
- Pandas, Matplotlib, Seaborn pour l'exploration et la visualisation des données.
- Scikit learn, Keras, TensorFlow pour le machine learning.
- Pillow pour le traitement d'image.

En ML (Machine Learning), python est un langage incontournable; facile à coder, orienté objet et des librairies pour les mathématiques.

CONCLUSION

Lors de ce stage de 10 semaines en tant que développeur Python - Django, j'ai pu mettre en pratique mes connaissances acquises avant et pendant ma formation de BTS SIO option SLAM au Greta de Lognes, tout en étant confronté à la réalité du monde du travail.

Après une période d'adaptation et d'auto-formation, j'ai pu m'intégrer à l'équipe et au projet et j'ai eu l'occasion d'améliorer ma sociabilité auprès de mes collaborateurs.

Ce stage a été très enrichissant pour moi, il m'a permis de découvrir le domaine du développement informatique qui se distingue, sur plusieurs aspects, de mes précédents emplois. J'ai pu participer à un réel projet. Les connaissances, la cohésion d'équipe, la prise en compte de la deadline, sont des attributs importants sur lesquels j'ai du travailler.

Si ma mission s'apparentait à du Full Stack (aussi bien le décor que l'envers), j'ai tout de même préféré la partie Back-End (l'envers du décor) sur laquelle j'ai passé la majorité du stage. La partie Front-End (coté utilisateur) avait tout de même son intérêt. Le placement, la couleur, la présentation sont des points essentiels qui feront que l'application sera "user-friendly" (intuitive) ou inefficace.

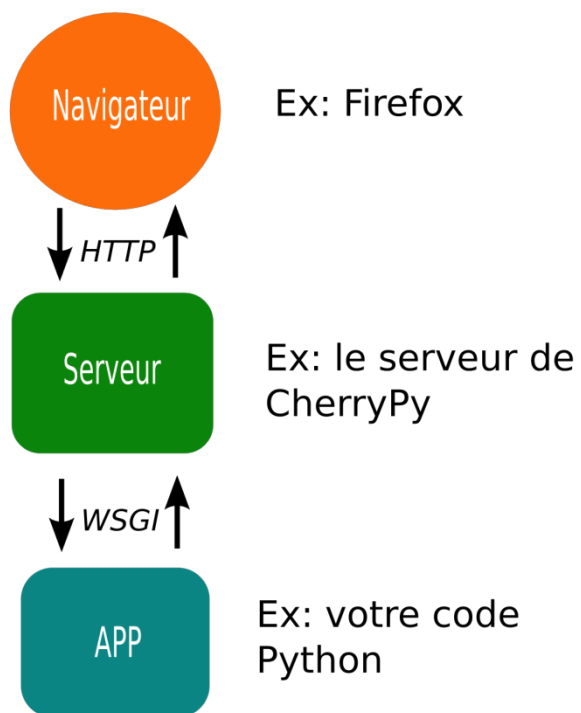
Avec mes maigres connaissances et ma motivation à toutes épreuves, j'ai pu remettre le site en route, modifier quelques parties de la vue et implémenter de nouvelles fonctionnalités. Pour autant, je ne me suis pas arrêté là. J'ai voulu aller plus loin dans l'intégration et j'ai essayé de créer une ambiance pérenne entre collaborateurs (ceux en inter-mission et le seconde stagiaire) alliant la détente, la demande de connaissances et une ligne de conduite au travail que j'ai pu acquérir avec mes différentes expériences passées.

Mon niveau d'études n'étant pas celui requis, je ne peux malheureusement pas continuer avec eux. Cela n'impacte en rien mon regard sur cette expérience que je referais avec joie. J'ai beaucoup apprécié ce stage et mon envie de travailler dans ce monde est d'autant plus forte depuis.

Il me reste maintenant à savoir où me spécialiser et jusqu'à quel diplôme aller. Ce qui est un sujet bien difficile.

Annexes

Dessin explicatif du WSGI :



les vues :

Vue Admin créée par Django

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

Change

Users

+ Add

Change

HOME

Algorithmes

+ Add

Change

Clients

+ Add

Change

Excels

+ Add

Change

Explorations

+ Add

Change

Images

+ Add

Change

Marchands

+ Add

Change

Theory games

+ Add

Change

Transactions

+ Add


Change

Type games

+ Add

Change

Exemple de vue crée

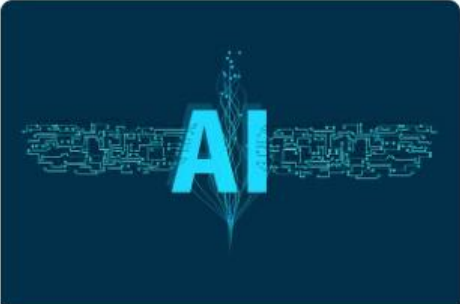


Entrainement Facile 2D

Le jeu de données sera restreint et nous connaissons les résultats à l'avance. Ceci nous permettant de vérifier que le code fonctionne.

NO ATTACK

UNDER ATTACK



Entrainement Moyen 2D



La suite de l'entrainement aura les mêmes algorithmes avec le dataset complet.

NO ATTACK

UNDER ATTACK

Vue GitLab :

Création d'un Readme.md pour expliquer le projet, les librairies utilisées, l'installation requise, etc.

 Readme.md 

Projet Prévention Fraude CB

Aujourd'hui, nous avons beaucoup de demande de confirmation lors d'un achat en ligne, le but étant d'assurer un maximum de sécurité, de confort et de sérénité lors de nos achats en ligne.

Avec ce projet, nous désirons obtenir les mêmes effets tout en diminuant drastiquement ces confirmations.

Pour ce faire, nous avons mis en place un site web pour présenter plusieurs approches de prévention, une détection et nous allons plus loin avec la théorie des jeux pour devancer les futures menaces.

Lien GitLab : <https://gitlab.com/finance.acensi/fraude>

acensi

Table of Contents

- 1. General Info
- 2. Technologies
- 3. Installation
- 4. collaboration
- 5. FAQs

General Info

Code Source :

admin.py (tout ce qui apparaîtra dans la vue admin du site)

```
1  from django.contrib import admin
2  from .models import Algorithme, Exploration
3
4  admin.site.register(Algorithme)
5  admin.site.register(Exploration)
6  admin.site.register(Transaction)
7  admin.site.register(Marchand)
8  admin.site.register(Client)
9  admin.site.register(TheoryGame)
10 admin.site.register(Excel)
11 admin.site.register(Image)
12 admin.site.register(TypeGame)
```

models.py (exemple de classe qui créera une table de données)

```
39 class Excel(models.Model):
40     title = models.CharField(max_length=50)
41     titleOfTheory = models.ForeignKey(TheoryGame, null=True, on_delete=models.CASCADE)
42     excel = models.CharField(max_length=128)
43     def __str__(self):
44         return self.title
```

views.py (exemple de fonction)

```
19 def game(request): #Contrôleur page théorie des jeux
20     #Get list of Theories
21     list_theories = TheoryGame.objects.filter(typeGame__title="preview").all()
22     print(list_theories)
23     #Get list of Images
24     previewImage = []
25     for theories in list_theories:
26         previewImage.append(Image.objects.get(titleOfTheory=theories))
27     print(previewImage)
28     #Return lists in game.html
29     data = {'games': list_theories, 'images': previewImage}
30     return render(request, 'game.html', data)
```

partie d'un "template" avec le langage Gabarit de Django

```
1 {% extends 'bases/base.html' %}
2 {% block content %}
3 {% load static %}
4
5 <div class="container mt-5 mb-8">
6     <h2 class="text-center"> Théorie des jeux </h2>
7     <h3 class="text-center"> ici nous aurons 3 versions de la théorie des jeux :<br>
8     entraînement facile 2D, entraînement moyen 2D, données réelles.</h3>
9 </div>
10
11 <div class="container">
12     <div class="row">
13         
14         {% for game in games %}
15
16             <div class="col-12 col-md-6 col-lg-3 col-xl-3" style="...">
17
```

Exemple de migration :

```
1 # Generated by Django 3.2.4 on 2021-07-05 13:28
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     initial = True
9
10     dependencies = [
11     ]
12
13     operations = [
14         migrations.CreateModel(
15             name='Algorithme',
16             fields=[
17                 ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
18                 ('title', models.CharField(max_length=50)),
19                 ('desc', models.TextField()),
20                 ('image', models.ImageField(upload_to='')),
21                 ('excel', models.CharField(max_length=50)),
22             ],
23         ),
24     ]
```

partie du fichier urls.py qui sert de routeur :

```
29 urlpatterns = [  
30     path('admin/', admin.site.urls),  
31     path('', home, name='home'),  
32     path('home/', home, name='home'),  
33     path('algo/<int:id_algo>', detail, name="detail"),  
34     path('gameTheory/', game, name="TheoryGame"),  
35     path('gameT/Def<int:id_NumberOfTheory>', noAtk, name="noAtk"),  
36     path('gameT/Atk<int:id_NumberOfTheory>', underAtk, name="underAtk"),  
37     path('article/recherche', search, name="search"),
```