# Modèle de copie :
# Évaluation en cours de formation

**Développeur Web et Web Mobile**

**Ceci est un modèle de copie. N'oubliez pas de renseigner vos prénom/nom, ainsi que le nom et le lien vers le projet.**

**Vous pouvez bien sûr agrandir les cadres pour répondre aux questions sur la description du projet si nécessaire.**

**Prénom : Emmanuel**
**Nom : SCELLES**

**ATTENTION ! PENSEZ À RENSEIGNER VOS NOM ET PRÉNOM DANS LE TITRE DE VOS FICHIERS / PROJETS !**

Nom du projet : Hypnos

Lien Github du projet : https://github.com/Thanos974/ECF-Hypnos contient le code source et les pdf des mockup , charte graphique, documentation technique et livret d'utilisation
Lien Wireframes Whimsical : https://whimsical.com/ecf-Bq182Xp7k3E8KKc82BGLz7
Lien Mockup Figma: https://www.figma.com/file/Jhu9uSi3kWtd2Nmh92bHYl/ECF-Hypnos
Lien Trello: https://trello.com/b/V9Ym2YcW/ecf-hypnos
**Attention ! Merci de bien classer vos documents dans votre Github ou votre drive.**
URL du site (si vous avez mis votre projet en ligne) : https://thanos.pythonanywhere.com

## Description du projet

1. Liste des compétences du référentiel qui sont couvertes par le projet

FRONT :
1. Maquetter une application
2. Réaliser une interface utilisateur web statique et adaptable
3. Développer une interface utilisateur web dynamique
4. Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.

BACK :
1. Créer une base de données
2. Développer les composants d'accès aux données
3. Développer la partie backend d’une application web ou web mobile
4. Élaborer et mettre en œuvre des composants dans une application de gestion de contenue ou e-commerce

2. Résumé du projet en français d'une longueur d'environ 20 lignes soit 200 à 250 mots, ou environ 1200 caractères espaces non compris

Dans le cadre de ma formation chez Studi j'ai eu à exécuter le développement d'une application complète avec un cahier des charges bien défini et ceux dans un délai imparti qui est du 15 mars au 21 avril 202. Pour cela je vais devoir démontrer des compétences au niveau frontend et back-end.

Hypnos est un groupe hôtelier fondé en 2004. Propriétaire de 7 établissements dans les quatre coins de l'hexagone, chacun de ces hôtels s'avère être une destination idéale pour les couples en quête d'un séjour romantique à deux.

Chaque suite au design luxueux et romantique inclut des services hauts de gamme (un spa privatif notamment, un room service, un service de conciergerie), de quoi plonger pleinement dans une atmosphère chic-romantique.
Hypnos souhaiterait ne pas dépendre uniquement de sites tiers comme Booking.com pour la réservation de ses chambres.

C'est pourquoi le groupe hôtelier Hypnos aimerait être pourvu de son propre système de réservation sur son propre site web.
Pour ce projet, il est question de mettre en place certaines user stories qui permettent à l'administrateur de gérer chacun de ses établissements ainsi que les gérants de chaque établissement du groupe.
Pour cela un formulaire de connexion avec nom, prénom, email et mot de passe sécurisé sont attendu.
Le gérant aura la possibilité de se connecter en tant que gérant d'un établissement mais également les réservations des clients.

Les clients pourront consulter les pages de différents hôtels du groupe Hypnos ainsi que leurs suites mais ne pourront réserver que s'ils sont inscrits et connecté.
Les clients non authentifiés ne pourront pas réserver de suites. Nous donnerons donc des accès spécifiques pour chacun d'entre eux pour qu'ils puissent interagir sur l'application selon ses besoins et ses droits administrés.
Enfin les clients pourront également réserver sur le site booking.com comme c'était le cas jusqu'à maintenant.

3. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet

Dans ce projet livré, le client a exigé certains points sur les user stories à développer qui sont :
- La gestion des établissements
- La gestion des Suites
- Un catalogue des établissements
- La réservation d'une suite
- Voir ses réservations
- Accélérer les réservations d'une suite
- Pouvoir contacter le groupe hôtelier

Le client ne souhaite pas avoir de fonctionnalité avec moyen de paiement car il se fera obligatoirement sur place même pour les réservations en ligne.

4. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

Les spécifications techniques du projet sont les suivantes :

Il nous faut un admin qui puisse gérer les gérants de chaque établissement avec un panel d'administration. Que l'hébergeur soit solide car il va traiter les connexion utilisateur des clients qui auront créé un compte ainsi que les comptes gérant et admin.

C'est pourquoi je choisi de partir sur python avec Django car c'est un Framework web pour les perfectionnistes qui ont des délais à respecter tout en satisfaisant aux exigences strictes des développeurs. Pour la partie Front je vais partir sur un Framework similaire qui est Bootstrap 5.

Puisque c'est un site web il nous faut utiliser pour la partie Front:
=> HTML5, CSS et JAVASCRIPT (c'est les seules langages compris par les navigateurs)
=> Bootsrap 5
Il faut également que le site soit responsif et mobile first c'est à dire conçu et pensé d'abord pour un usage mobile.
Pour le back je vais utiliser le framework django avec une base de donnée en SQL

5. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité

Etant donnée que je suis parti sur le Framework Django pour la réalisation de mon projet "Hypnos", j'ai consulté quelques sites spécialisés comme Docstring.fr pour recueillir de nouvelles informations sur l'utilisation des systèmes d'authentification que propose Django.
J'ai également cherché sur la documentation officielle de Django (https://www.djangoproject.com/) pour savoir qu'elle était les éléments permettant la sécurisation des donnés comme l'authentification des utilisateurs.

J'ai découvert qu'il était proposé par Django le "CSRF_TOKEN" qui permet de sécuriser les données afin d'assurer que ces derniers sont authentiques et non pas été intercepté ou modifié par un tiers pendant la requête.
Cela permet d'éviter les attaques de type CROSS SITE, Ces types d'attaques consiste à injecter du code malicieux à travers le bouton d'un formulaire ou du javascript.
Ce token est également utilisé sur tous les formulaires car obligatoire par Django sinon la requête ne peut aboutir.

Utilisation de SQLite3 pour la partie développement puis une migration vers PostgreSQL pour la production sera effectué cependant le site hébergeur gratuit que j'utilise ne le permet pas.

6. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone

J'ai dû effectuer pas mal de recherche aussi sur différent site comme stackoverflow mais aussi d'autre comme docstring.fr, la documentation officielle de Django et YouTube. Aujourd'hui les possibilités de trouver des ressources sont multiples.
Pour ma part mes recherches ont le plus souvent était orienté lors des bugs que je rencontrais sur mon IDE lors de sa phase de développement.

Par exemple lors de la création d'un formulaire de contact ou de connexion, je n'arrivais pas à réaliser ma requête car je n'avais pas utilisé le " CSRF_TOKEN" ou selon le type de formulaire utilisé il était rendu obligatoire d'utiliser le Field **username** sinon la requête retourné une erreur.

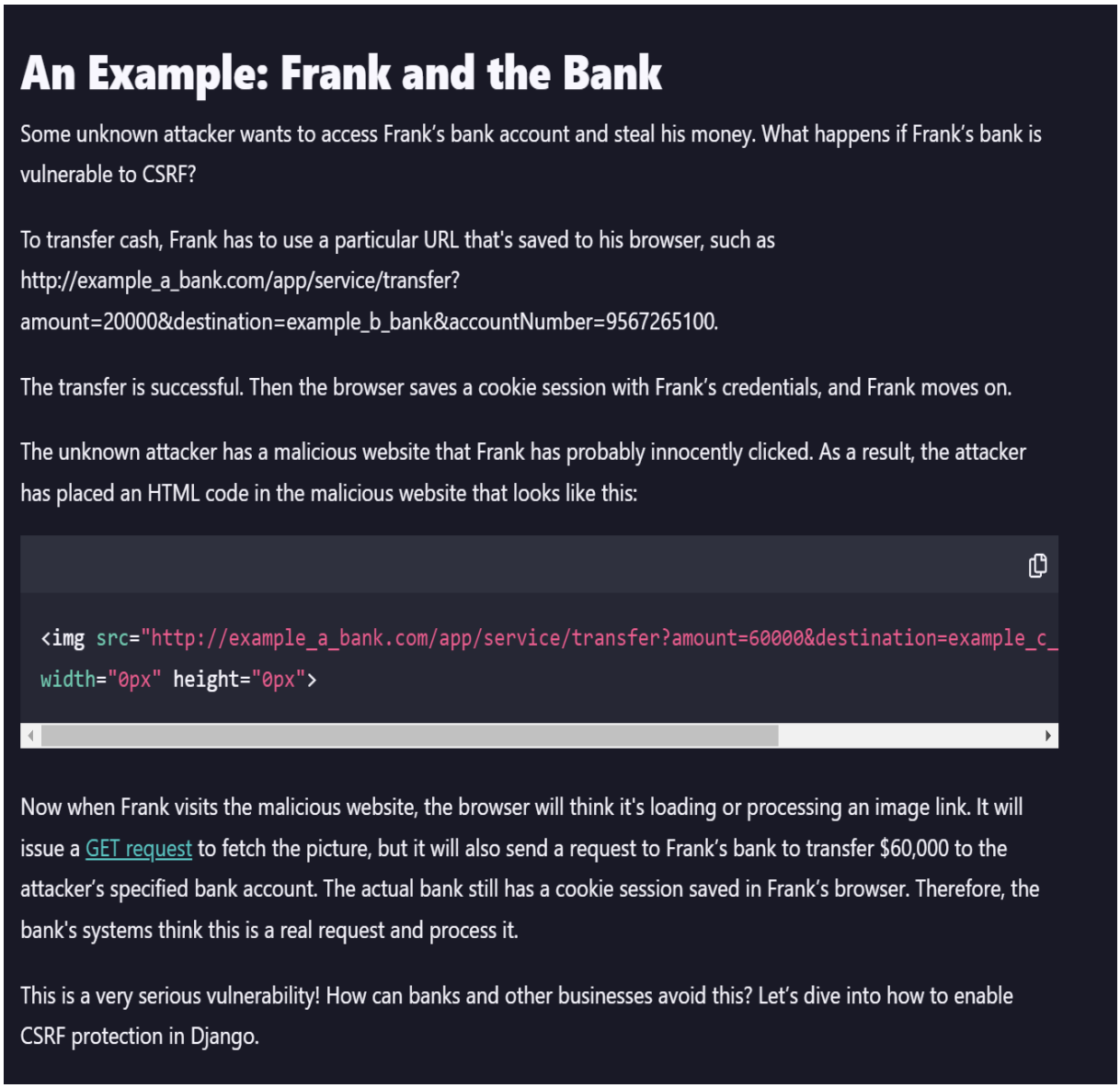
7. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique (environ 750 signes).

CSRF (ou XSRF) est également connu sous le nom de cross-site request forgery. Comme son nom l'indique, CSRF est un type d'attaque contre des sites, principalement menée par d'autres sites (malveillants), ou parfois par un utilisateur (malveillant) sur le site.

En général, il existe de nombreux cas où un site demande à un utilisateur de remplir des données provenant d'un autre site Web au nom de cet utilisateur. Par exemple, de nombreux blogs utilisent Disqus pour alimenter leur système de commentaires. Pour commenter dans ce blog particulier, le blog exige que vous vous connectiez d'abord à Disqus. Il s'agit d'une utilisation de base d'un CDN (réseau de diffusion de contenu), et cet exemple est une demande légitime.

Les attaques CSRF reposent généralement sur l'identité de l'utilisateur. Que se passe-t-il donc lorsqu'un utilisateur visite un site web malveillant ? Ce site envoie des formes cachées d'une requête JavaScript XMLHttpRequest. Cette requête utilise les informations d'identification d'un utilisateur (celui qui a visité le site malveillant) pour effectuer certaines actions sur un autre site Web qui fait confiance au navigateur ou à l'identité de l'utilisateur.

Un site identifie généralement les utilisateurs authentifiés en enregistrant dans leur navigateur des cookies dont les en-têtes et le contenu représentent cet utilisateur particulier. Les attaquants s'en servent pour accéder aux informations d'identification de l'utilisateur afin de mener leurs attaques.



## An Example: Frank and the Bank

Some unknown attacker wants to access Frank's bank account and steal his money. What happens if Frank's bank is vulnerable to CSRF?

To transfer cash, Frank has to use a particular URL that's saved to his browser, such as http://example_a_bank.com/app/service/transfer?amount=20000&destination=example_b_bank&accountNumber=9567265100.

The transfer is successful. Then the browser saves a cookie session with Frank's credentials, and Frank moves on.

The unknown attacker has a malicious website that Frank has probably innocently clicked. As a result, the attacker has placed an HTML code in the malicious website that looks like this:

```
<img src="http://example_a_bank.com/app/service/transfer?amount=60000&destination=example_c_
width="0px" height="0px">
```

Now when Frank visits the malicious website, the browser will think it's loading or processing an image link. It will issue a GET request to fetch the picture, but it will also send a request to Frank's bank to transfer $60,000 to the attacker's specified bank account. The actual bank still has a cookie session saved in Frank's browser. Therefore, the bank's systems think this is a real request and process it.

This is a very serious vulnerability! How can banks and other businesses avoid this? Let's dive into how to enable CSRF protection in Django.

Un attaquant inconnu veut accéder au compte bancaire de Frank et lui voler son argent. Que se passe-t-il si la banque de Frank est vulnérable à CSRF ?

Pour transférer de l'argent, Frank doit utiliser une URL particulière qui est enregistrée dans son navigateur, comme : http://example_a_bank.com/app/service/transfer?amount=20000&destination=example_b_bank&accountNumber=9567265100.

Le transfert est réussi. Le navigateur enregistre alors une session de cookies avec les informations d'identification de Frank, et ce dernier peut continuer son chemin.

```
<img src="http://example_a_bank.com/app/service/transfer?amount=60000&destination=example_c
width="0px" height="0px">
```

Maintenant quand Frank visitera le site web malveillant, le navigateur pensera qu'il est en train de charger ou de traiter un lien image. Il émettra une requête GET pour récupérer l'image, mais il enverra également une requête à la banque de Frank pour transférer 60 000 dollars sur le compte bancaire spécifié par l'attaquant. La banque a toujours une session de cookies enregistrée dans le navigateur de Frank. Par conséquent, les systèmes de la banque pensent qu'il s'agit d'une demande réelle et la traitent.

Il s'agit d'une vulnérabilité très grave ! Comment les banques et autres entreprises peuvent-elles éviter cela ? Voyons comment activer la protection CSRF dans Django.

Lien : https://www.stackhawk.com/blog/django-csrf-protection-guide/

8. Autres ressources

9. Informations complémentaires