

DOSSIER PROJET

Développeur Web et Web mobile



Apprendre le japonais avec Hanasu

« La connaissance s'acquiert par l'expérience, tout le
reste n'est que de l'information. »

« Albert Einstein »

TABLE DES MATIÈRES

I. Introduction.....	5
A) Remerciements :.....	5
B) Glossaire :.....	6
II. Cahier des charges.....	7
A) Contexte :.....	7
B) Présentation du projet :.....	7
C) Présentation de l'entreprise :.....	7
D) Objectifs du site :.....	7
E) Les cibles :.....	8
F) Accessibilité :.....	8
G) Fonctionnalités.....	8
H) Les contenus :.....	9
I) Échéance du projet :.....	9
III. Compétences du référentiel couvertes par le projet.....	10
A) Développer la partie front-end d'une application web ou mobile en intégrant les notions de sécurité.....	10
B) Développer la partie Back-end d'une application web ou mobile en intégrant les notions de sécurité.....	11
C) Compétence Transverse:.....	12
IV. Résumé exécutif.....	13
V. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile.....	14
A) Technologies utilisées.....	14
B) Environnement.....	14
C) Web mobile :.....	15
D) Argumentaire des choix technique :.....	15
E) Organisation.....	15
F) Wireframes.....	16
G) Modèle de données:.....	17
.....	17
VI. Réalisations du candidat comportant les extraits de code les plus significatifs et en les argumentant.....	18
A) Présentation du Framework:.....	18
B) L'architecture:.....	18
C) Le front.....	20
D) Le CSS:.....	21
E) Récupération des informations de la base de données pour le jeu de mémoire:..	22
F) Utilisation des variables dans le contexte:.....	23
A) Fonctionnement du jeu :.....	24

VII. Présentation du jeu d'essai élaboré par le candidat de la fonctionnalité la plus représentative.....	27
VIII. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité.....	28
IX. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet.....	29
A) Recherches :.....	29
B) Recherche anglophone :.....	29
X. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction du candidat (environ 750 caractères).....	30
Annexe 1 (Wireframes).....	31
Annexe 2 (Modele de données).....	32

I. INTRODUCTION

A) Remerciements :

J'adresse tous mes remerciements aux personnes qui ont contribué à la réalisation de ce dossier et tout particulièrement à :

Alain, notre formateur qui a su m'accompagner et me transmettre son savoir tout au long de cette formation.

Mes collègues de formation pour nos échanges et ce partage qui a permis de nous enrichir mutuellement de nos expériences.

L'entreprise Subteno IT, Sébastien LANGE, Vincent COFFIN et l'ensemble des collaborateurs, pour leur disponibilité et leur accompagnement durant notre stage.

Cécilia pour ses conseils avisés et sa relecture.

Ma famille et mes amis avec lesquels j'ai pu échanger, partager et m'enrichir.

Théo notre Mozart de la programmation qui à fait de moi son projet 3 et m'a permis d'avancer dans la formation.

Mes enfants qui ont été très sages durant la formation et m'ont permis d'avoir beaucoup de temps a consacré à celle-ci.

B) Glossaire :

- **Framework web**

Aide au développement d'applications web, ils offrent des outils et un cadre de développement.

- **Wireframes**

Maquette fonctionnelle sous forme de schéma de l'interface utilisateur qui permet de placer les différents composants.

- **CRUD**

Create : créer, Read : lire, Update : mettre à jour, Delete : supprimer

Désigne les opérations pouvant être pratiquées sur des collections de données.

- **Contexte**

Ensemble des données qui se transmettent entre les pages, Django l'utilise sous la forme d'un dictionnaire.

- **Fichiers statiques**

Ensemble des fichiers supplémentaires utilisés par l'application web tels que les images, les scripts JavaScript ou encore les fichiers de style CSS.

II. CAHIER DES CHARGES.

A) Contexte :

Ce projet constitue le troisième et dernier projet de la formation, et répond au sujet suivant proposé par notre formateur : Développer une application web au service du plus grand nombre, c'est-à-dire traiter d'une problématique sociale, associative, humaniste, environnementale ou de santé. Il est réalisé seul.

B) Présentation du projet :

Ce projet est mis en place afin de faciliter l'apprentissage du japonais grâce à ce site.

Il permet aux différents élèves de Mme Thomas d'apprendre les idéogrammes (Hiragana et katakana) par un jeu de mémoire. Le site web est mis en place également pour permettre aux apprenants d'avoir leur propre glossaire mais aussi un blog pour la diffusion d'information lié à activité de Mme Thomas.

C) Présentation de l'entreprise :

Sophie Thomas, 30 ans, donne des cours de japonais depuis 2013. Son entreprise, auparavant à 100 % en présentiel, est passé au digital depuis 4 ans.

D) Objectifs du site :

L'apprentissage du japonais et la mise en avant ou promotion des activités de la professeure sont les principales vocations de ce site.

On doit pouvoir y entrer ses informations personnelles pour s'inscrire. On peut aussi apprendre et tester ses connaissances en termes d'idéogrammes grâce à un jeu, apprendre des faits sur le japon et construire un glossaire personnel à chaque personne inscrite. Ce site a pour vocation d'être utilisé sur les territoires francophones.

E) Les cibles :

Hanasu s'adresse principalement aux étudiants de Mme Sophie Thomas qui souhaiterait apprendre en s'amusant et utilisé un outil pédagogique numérique.

Mais aussi à toutes personnes ayant envie d'apprendre le japonais ou de tester ses connaissances avec des contenus clairs et de sources fiables (ici, articles de Mme Sophie Thomas).

F) Accessibilité :

L'application web est développée comme une application web responsive, ce qui permet aux utilisateurs d'avoir accès depuis un ordinateur, tablettes ou smartphone.

G) Fonctionnalités.

Permissions:

Il existe 3 niveaux de permissions distinct :

- le visiteur,
- le visiteur inscrit,
- l'administrateur.

Les comptes administrateurs sont chargés de créer des articles pour faire vivre le site. Ils sont gérés exclusivement par l'administrateur pour maintenir un contrôle des accès.

Visiteur :

- page d'accueil,
- afficher les articles du blog et les commenter.

Inscrit :

- page d'accueil,
- afficher les articles,
- accès au glossaire,
- accès au jeu,
- commenter un article,
- se connecter/se déconnecter,
- consulter son profil.

Administrateur :

- page d'accueil,
- afficher les articles,
- accès au glossaire,
- accès au jeu,
- trier/filtrer/rechercher,
- commenter un article,

- se connecter/se déconnecter,
- consulter son profil,
- mettre à jour les données,
- créer les comptes rédacteurs.

H) Les contenus :

Tout le contenu de ce projet (image, photos, logo, textes) sont libre de droits et d'utilisations.

I) Échéance du projet :

Le projet se déroule du 3 janvier 2022 au 21 avril 2022.

III. COMPÉTENCES DU RÉFÉRENTIEL COUVERTES PAR LE PROJET.

A) Développer la partie front-end d'une application web ou mobile en intégrant les notions de sécurité.

Maquetter une application:

Le choix de charte graphique étant intervenu tard dans le projet étant daltonien il est vrai que pour moi cela complique les choses, j'ai réalisé quelques wireframes. Cela m'a permis d'avoir une vision globale de la disposition de mes éléments ainsi les différentes interactions entre mes pages ainsi que le parcours utilisateur.

Je me suis servi pour cela du site Nicepage.

J'ai maqueté les premières pages de mon application. Vous trouverez en annexe (annexe 2) les maquettes que j'ai réalisées.

Réaliser une interface utilisateur web statique et adaptable:

À partir des wireframes et des spécifications fonctionnelles et techniques, j'ai conçu les pages de l'application web avec HTML5 et CSS3. Celle-ci est en fait une grosse interface utilisateur, qui s'adapte en fonction du type d'utilisateur connecté et de son support de connexion. Je vous renvoie sur le cahier des charges pour plus de détails.

Développer une interface utilisateur web dynamique:

Durant la conception du site, l'utilisateur inscrit a un profil personnalisé évolutif par les scores du jeu de mémoire ainsi que par les récompenses gagnées.

Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce:

La solution de gestion de contenu est apparue durant la conception du glossaire individuelle. Une interface pour la gestion du CRUD était indispensable pour l'utilisateur.

Ainsi le visiteur enregistré peut ajouter photos, descriptions et éléments relatif à un mot ou expression(traduction).

B) Développer la partie Back-end d'une application web ou mobile en intégrant les notions de sécurité.

Créer une base de données:

Création de classe en Python permettant la création de table dans la base de données. Ces données peuvent être manipulées par l'interface administrateur de Django ou tout simplement par un système de gestion de base de données (SGBD).

Dans le fichier settings, nous retrouvons Databases: un dictionnaire de dictionnaire permettant d'identifier par défaut la base de donnée.

Plusieurs champs sont nécessaires : Engine, Name, User, Password, Host et Port.

Développer les composants d'accès aux données:

La création de vues (view) permet de récupérer des informations dans la base de données pour générer du contenu qui est ensuite affiché à l'utilisateur via le HTML. Des formulaires permettent de rechercher ou filtrer les données affichées par les différentes pages, notamment des formulaires d'ajout, modification ou suppression.

Développer la partie back-end d'une application web ou web mobile:

Django respecte l'architecture MVT, directement inspirée du très populaire modèle MVC.

Nous créons des modèles de données que nous utilisons dans les fichiers «Views». Nous utilisons les « Views » pour l'affichage de données dans les templates constituer de HTML.

Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce:

La gestion du CRUD est indispensable lors de la création d'un blog

Le CRUD du blog est accessible via l'interface administrateur liée à des tables contenant les champs de type :

- ImageField,
- DateTimeField

- BooleanField.

Une relation avec la table user permet l'identification du créateur de contenu.

C) Compétence Transverse :

Utiliser l'anglais dans son activité professionnelle en informatique

Mon code est rédigé en anglais, la documentation est également rédigée en anglais pour permettre une reprise du projet plus large.

Toutes mes recherches sont effectuées en anglais pour permettre un nombre de réponses plus large et utiliser les termes techniques anglais.

De plus, du peu de mes compétences linguistiques, je suis fier de vous fournir à ce jour ces documents en anglais, de les avoir réussis et compris, et qu'ils aient un sens.

Actualiser et partager ses connaissances en web et web mobile

Au travers de ma veille j'ai réussi à trouver des nouveautés sur les technologies, de nouvelles façons de travailler ou de développer. J'ai pu partager mes connaissances au sein de la formation au travers des mini-cours ou de petites présentations (HTML et CSS).

IV. RÉSUMÉ EXÉCUTIF

Il est reconnu que l'apprentissage des langues étrangères se complique avec l'âge.

Si l'on apprend une nouvelle langue, il vaut mieux le faire dès les premières années de sa vie, quand le cerveau est le plus malléable et capable de s'adapter... Plus tard, il sera toujours possible de devenir bilingue : ce sera juste plus difficile. C'est neurologique !

Pour cela, j'ai créé le site Hanasu ayant pour objectif de faciliter l'apprentissage par le jeu et la gratification personnelle.

Il promeut l'apprentissage du Japonais, pour pouvoir apprendre les Hiragana/ Katakana, avec un système de lexique et un blog. Mais aussi une table des scores qui permet à l'utilisateur de suivre sa progression avec le déblocage de trophées.

J'ai mis en place un formulaire d'inscription ouvrant un espace personnel, un jeu de mémorisation qui regroupe les Hiragana/Katakana. Un onglet « le saviez-vous » affiche des faits réels sur le Japon. Une partie « Lexique », l'utilisateur peut créer son propre Lexique avec ses propres images avec le Romanji (romanisation du mot japonais et sa traduction). Un Blog qui permet à l'administrateur d'ajouter des articles disponibles pour tous les utilisateurs.

La navigation est assurée grâce à un menu présent dans l'entête.

Afin de répondre à la demande, j'ai développé ce site web responsive en utilisant une méthode de travail agile, les langages de programmation Python, JavaScript, HTML et CSS, ainsi que le Framework Django et une Base de données relationnelles qui gère PostgreSQL.

V. SPÉCIFICATIONS TECHNIQUES DU PROJET, ÉLABORÉES PAR LE CANDIDAT, Y COMPRIS POUR LA SÉCURITÉ ET LE WEB MOBILE.

A) Technologies utilisées

J'ai utilisé HTML ⁵, CSS ³ pour la partie front-end en privilégiant les balises sémantiques et les bonnes pratiques. L'adaptabilité mobile est gérée par le CSS ³.

Le projet est principalement développé avec Django, un framework web utilisant le principe du Model View Template (MVT). J'ai choisi d'utiliser Django car il est développé en Python, open-source, rapide, flexible, sécurisé et très utilisé pour développer des applications web, ce qui donne accès à beaucoup de tutoriels, documentations.

Le projet utilise une base de données relationnelles PostgreSQL consultée en utilisant DBeaver.

B) Environnement

J'ai utilisé les outils suivants :

– VS Code : éditeur de code extensible, <https://code.visualstudio.com/>

– Environnement virtuel Python : permet de définir les modules nécessaires pour que l'application web fonctionne. Cette liste est regroupée dans un fichier requirements.txt, <https://docs.python.org/3/library/venv.html>

– Dbeaver : système de gestion de base de données, <https://dbeaver.io/>

– Photoshop : éditeur d'image de la suite Adobe , <https://www.adobe.com/>

– Git/Github : comme outil de versioning
<https://git-scm.com/> | <https://github.com/>

– Chrome et Firefox : pour consulter l'application web, tester le responsive et surtout l'application web progressive.

C) Web mobile :

Le site doit être adaptable sur différents appareils et navigateur. En 2022, 98 % des utilisateurs utilisent un appareil mobile.

D) Argumentaire des choix technique :

Sécurité :

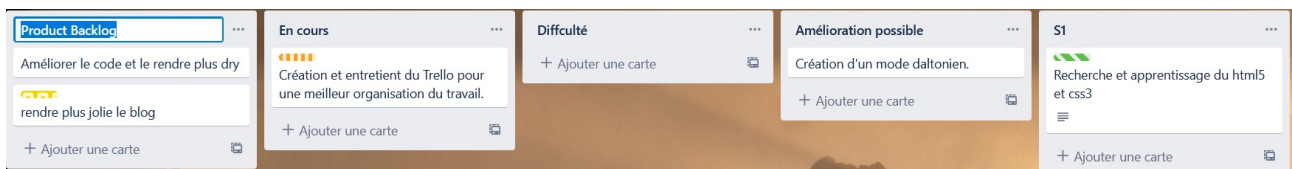
Lors de la veille sur les vulnérabilités de sécurité, je me suis informé sur les différentes mesures de protection. Dans le cadre de ce projet, je me suis principalement renseigné sur les failles XSS (Cross-Site-Scripting). Cette faille dangereuse, permet l'injection de code HTML ou JavaScript dans une base de données s'il n'y a pas de vérification du côté client et serveur.

Afin de réduire les risques sur ma page web, j'utilise le CRSF token (mot de passe crypté) permettant de vérifier l'origine des données entrant dans ma base de données.

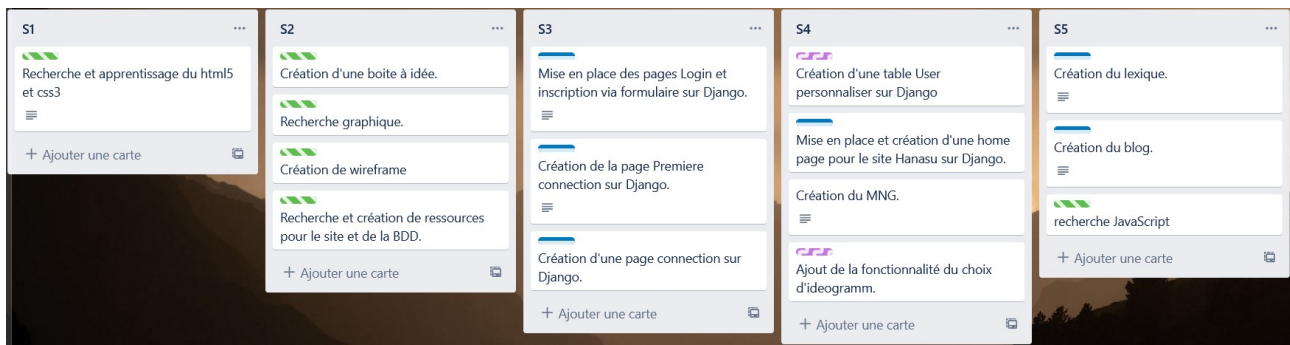
E) Organisation

Approche Agile:

J'ai travaillé en m'inspirant de la méthode SCRUM, en m'organisant en sprint d'une semaine alimenté par le product backlog regroupant toutes les tâches du projet.



Un cycle étant composé de réunion journalière (daily meeting), d'une présentation au client de l'avancée (sprint review) et d'une rétrospective (sprint retrospective) réalisés en fin de semaine avec notre formateur.



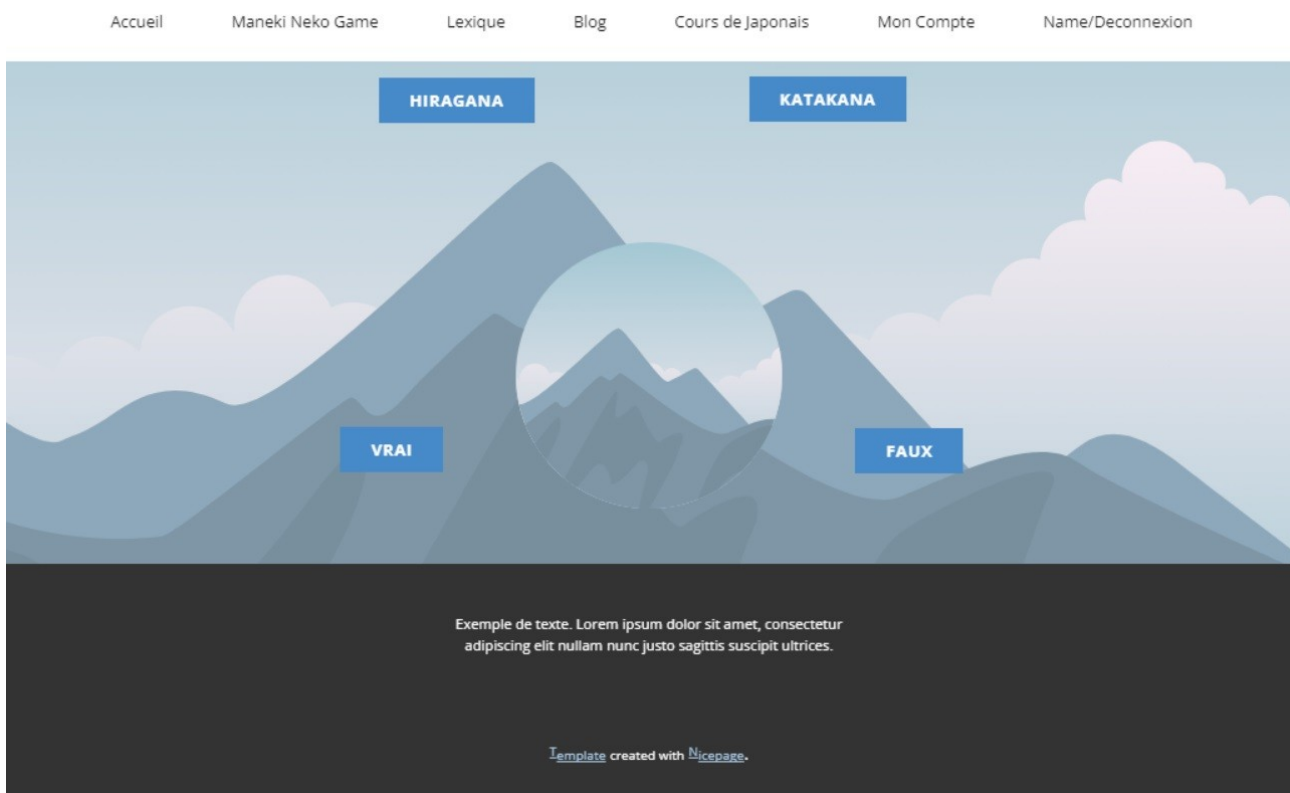
Cette organisation permet de prévoir les étapes nécessaires pour l'aboutissement du projet sans être inscrite dans le marbre ce qui permet de s'adapter aux changements ou nouveautés apportées par le développement ou le client.

J'ai réussi à maintenir cette organisation jusqu'à l'arrivée en stage. J'ai utilisé un tableau kanban pour avoir une vision globale de l'avancée du projet. Les outils Meet et Slack m'ont permis de réaliser les réunions journalières et de garder la liaison malgré la situation sanitaire et géographique de ma cliente.

F) Wireframes

J'ai développé l'interface du site pour qu'il soit accessible sur tout support, le projet étant destiné à une utilisation sur smartphone ou tablette et ordinateur, au vu des utilisateurs visés.

En voici un exemple :



J'ai uniquement utilisé les wireframes pour illustrer ma vision des éléments du jeu. Mes entretiens et discussion avec Mme Thomas m'a guidé tout au long du projet. La charte graphique ayant été choisie bien plus tard à cause de mon problème de daltonisme.

L'identité graphique finale du site n'est arrivée que pendant les dernières semaines de développement.

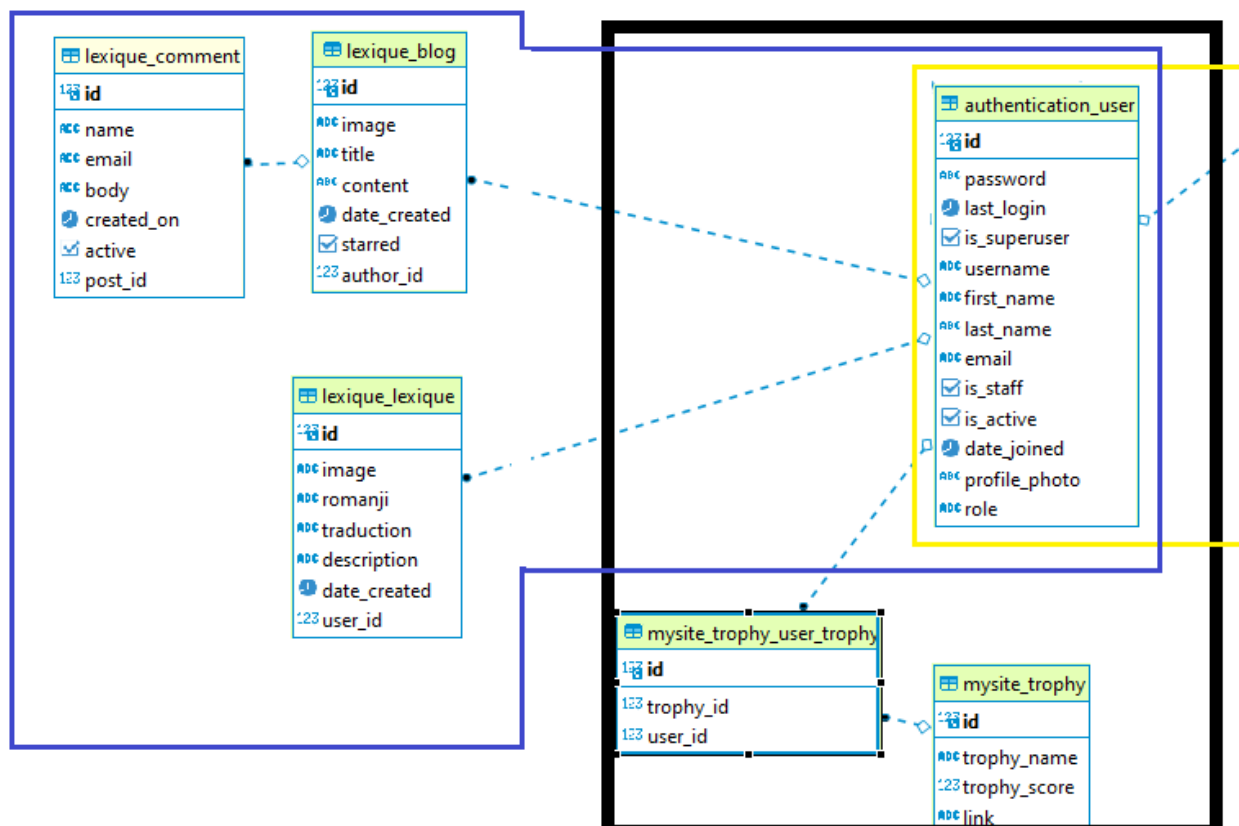
(Autres wireframes en annexe1)

G) Modèle de données:

La base des tables nécessaires au bon fonctionnement de Django qui ont été générées automatiquement ne sont pas représentées ici, la table (authentication_user) est utilisée dans le modèle des articles du blog pour en identifier l'auteur. Les autres tables sont créées pendant la commande makemigrations par les classes présentent dans le fichier python models héritant de la classe models.Model de Django.

Django permet de créer, lire, mettre à jour ou encore supprimer (CRUD) sans utiliser de requête SQL (en réalité Django les réalisent pour nous) et ainsi limiter le risque d'injection SQL par l'utilisateur.

Pour voir la totalité des tables présentes créé par mes soins dans la base de données, se référer à l'annexe 2.



VI. RÉALISATIONS DU CANDIDAT COMPORTANT LES EXTRAITS DE CODE LES PLUS SIGNIFICATIFS ET EN LES ARGUMENTANT.

A) Présentation du Framework:

Django suit le principe « Model-View -Template», dérivé du très connu « Modèle-Vue-Contrôleur » .

Le contrôleur est Django lui-même, le modèle est transformé en tables SQL via l'ORM (*Object Relational Mapping*) intégré et la vue est une page HTML enrichie de variables propres à Django.



B) L'architecture:

Un site web consiste en une ou plusieurs sections, par exemple un site principal, un blog, un wiki...

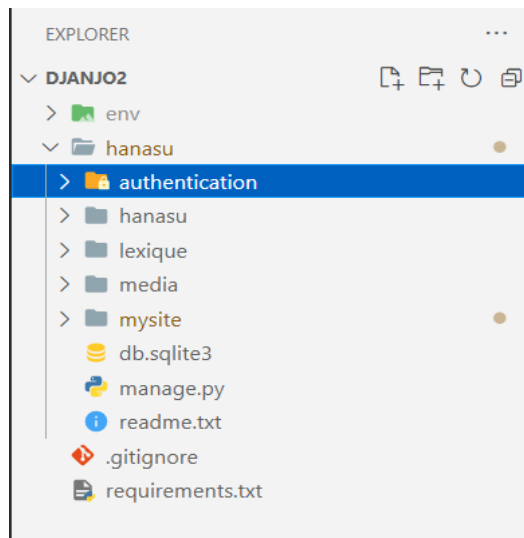
La bonne pratique avec Django est de réaliser chacun des composants comme des applications séparées qui pourront éventuellement être réutilisées dans d'autres projets.

Nous devons utiliser la commande «django-admin» pour créer le dossier du projet, les sous-dossiers et fichiers de base ainsi que le script de gestion du projet (manage.py).

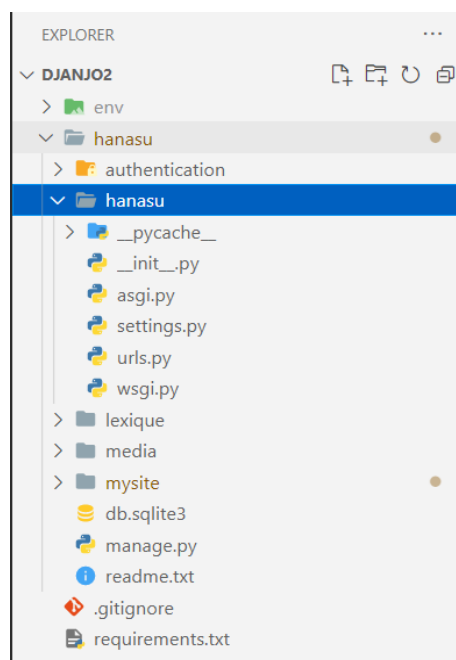
(ici Hanasu)

Puis nous utilisons la commande «manage.py» pour créer une ou plusieurs *applications* du projet.

(ici authentication, lexique et mysite)



Enregistrez la nouvelle application dans le projet puis liez les URL et chemins pour chaque application.



Une application est composée de plusieurs dossiers et fichiers dont:

Static: dossier regroupant par exemple les images, fichiers JavaScript ou encore les fichiers CSS.

Template: dossier contenant les fichiers HTML (_base et HTML).

Admin: permet d'enregistrer les models (Tables) utilisé dans l'application.

Models: Permet de créer les class, représentation python des tables de la base de données.

Views: permet de faire les opérations en python, comme par exemple la récupération du contenu de la BDD.

C) Le front.

Le HTML dans Django est séparé en deux parties toutes deux se trouvant dans les templates:

La première partie appelée « _base.html » :

Nous commençons par l'import (du ou) des fichiers CSS et le {%Block title%} dans le head contenant les balises Meta du fichier. Puis dans le « header » j'ai choisi d'intégrer le menu du site pour une question de praticité celui-ci revenant à presque chaque page.

Dans le « body » J'appelle le contenu de la page grâce aux balises {%block content%}

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="{% static css %}">
  <link rel="stylesheet" href="{% static css2 %}">
  <title>{% block title %}{% endblock title %}</title>
</head>
<body>
  <header>
    <nav>
      <label for="toggle">≡</label>
      <input type="checkbox" id="toggle">
      <div class="main_pages">
        <a href="{% url 'home' %}">Accueil</a>
        <a href="{% url 'maneki_neko' %}">Maneki Neko</a>
        <a href="{% url 'lexique' %}">Lexique</a>
        <a href="{% url 'blog' %}">Blog</a>
        <a href="#">Cours de japonais</a>
        <a href="{% url 'user_page' %}">Mon compte</a>
        <a href="{% url 'logout' %}">{{user.username}} / se déconnecter</a>
      </div>
    </nav>
  </header>
  {% block content %}{% endblock content %}
</body>
</html>
```

La seconde partie:

Elles portent le nom de la page désirée et dans un premier temps nous lui indiquons qu'elle est l'extension du fichier `_base` via la balise `{%extend%}` suivie par le chargement du fichier static.

Nous retrouvons le `{%block title%}` puis le `{%block content%}` contenant respectivement le titre de la page et le contenu de celle-ci (hormis le menu)

Le `{%block content%}` est constitué de html.

```
1  {% extends 'mysite/_base.html' %}
2  {% load static %}
3  {% block title %}Maneki Neko Game{% endblock title %}
4  {% block content %}
5
6  > <section>...
25 </section>
26 
27 > <span class="container">      You, 2 months ago • ajout de fonction pagination et blog ... ...
55 </span>
56
57 > <section>...
61 </section>
62
63 {% endblock content %}
```

D) Le CSS:

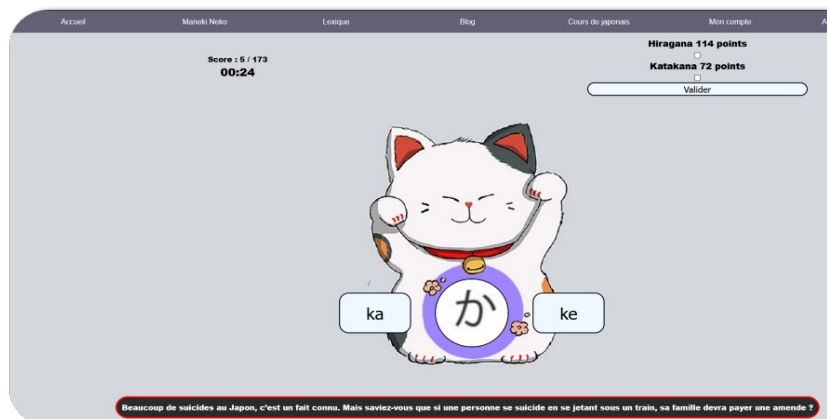
De la même façon que le HTML, le CSS n'est pas vraiment un langage de programmation. C'est un *langage de feuille de style*, c'est-à-dire qu'il permet d'appliquer des styles sur différents éléments sélectionnés dans un document HTML.

Par exemple, on peut sélectionner tous les éléments d'une page HTML qui sont des paragraphes et afficher leurs textes en rouge avec ce code CSS.

Ici la partie responsive de mon CSS:

Les requêtes média (*media queries*) permettent de modifier l'apparence d'un site ou d'une application en fonction du type d'appareil (impression ou écran par exemple) et de ses caractéristiques (la résolution d'écran ou la largeur de la zone d'affichage (*viewport*) par exemple).

```
@media(max-width: 991px)
{
  .child
  {
    width: 160px;
    height: 160px;
  }
}
```



E) Récupération des informations de la base de données pour le jeu de mémoire:

Pour la création du jeu de mémoire, j'ai fait le choix de faire des request,Get :

Ici, l'utilisateur, n'envoie aucune information dans la base de données. Il n'y a aucun besoin de sécuriser la request avec un CSRF Token, cette méthode est visible dans l'URL du site.

```
@login_required
def hanasugame(request):
    documentary = Documentary.objects.all()
    # hiragana = "on" ou "off"
    hiragana = request.GET.get("hiragana", "off")
    # katakana = "on" ou "off"
    katakana = request.GET.get("katakana", "off")
```

« Hiragana », une fois activé (via un bouton en front) va filtrer l'affichage des idéogrammes pour n'afficher que les Hiragana.

« Katakana » aura le même effet, mais pour sa propre catégorie d'idéogrammes.

Si les deux boutons sont actionnés ou désélectionnés, alors les deux catégories sont actives.

Hiragana ☐ Valider

Score : 50 / 50

Katakana ☐ Valider

```
# si hiragana est "on" et katakana est "off" filtre pour avoir que les hiragana.
if hiragana == "on" and katakana == "off":
    ideogramms = Ideogramm.objects.filter(ideotype_id = Ideotype.objects.get(Name = "Hiragana"))
# si katakana est "on" et hiragana est "off" alors on filtre pour avoir que les katakana.
elif katakana == "on" and hiragana == "off":
    ideogramms = Ideogramm.objects.filter(ideotype_id = Ideotype.objects.get(Name = "Katakana"))
# sinon prendre toute la table ideogramm.
else:
    ideogramms = Ideogramm.objects.all()
```

F) Utilisation des variables dans le contexte:

Ici nous déterminons les variables que nous allons avoir besoin dans le contexte:

random_ideogramms: Variable contenant les deux idéogrammes qui sera affiché sur la page web du jeu.

random_documentary: Variable contenant un fait réel sur le Japon affiché sur la page.

Les deux CSS.

score: permettant de récupérer les informations du joueur.

now: Variable contenant la date du jour avec la time zone «utc»

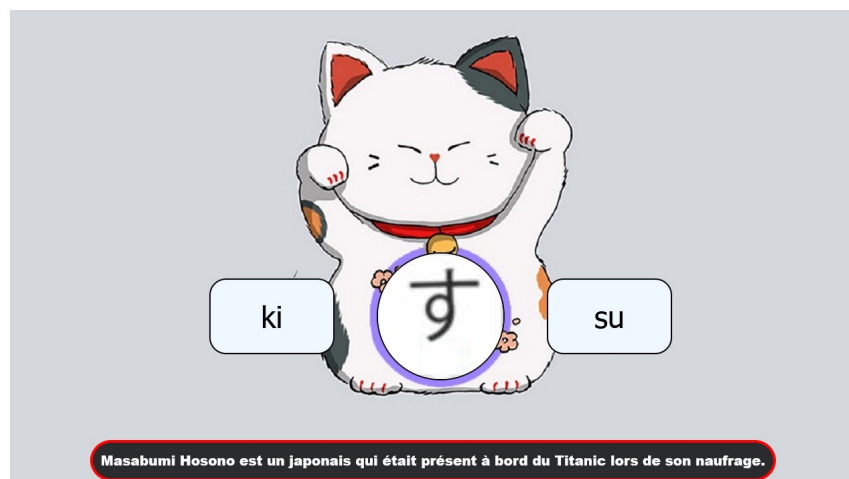
correct_ideogramm: Variable contenant la bonne réponse du jeu de mémoire.


```

random_ideograms = random.choices(ideograms, k=2)
random_documentary = random.choice(documentary)
css = "mysite/maneki_style.css"
css2 = "mysite/menu.css"
score = Score.objects.get(user_id=request.user.id)
now = datetime.datetime.now().replace(tzinfo=datetime.timezone.utc)
correct_ideogram = random.choice(random_ideograms)

context = {
    "scores" : score,
    "correct_ideogram" : correct_ideogram,
    "ideograms": random_ideograms,
    "random_documentary": random_documentary,
    "css":css,
    "css2": css2,
    "hiragana": hiragana,
    "katakana": katakana,

```



A) Fonctionnement du jeu :

Mise à jour du score journalier :

Pour le bon fonctionnement du jeu et surtout la mise en place d'une gratification, j'ai mis en place la variable «score.last_game» me permettant de vérifier une condition.

Si la différence entre «now» et «score.last_game» est supérieur ou égal à 1 jour, alors le current_score du joueur retombe à zéro, cela permet d'actualiser le score du joueur de façon journalière.

```

if request.method == "POST":
    # ID de la réponse qui vient d'être cliqué.
    current = request.POST['current']

    # L'ID de la bonne réponse.
    correct = request.POST['correct']

    # Création de la variable contenant la date d'aujourd'hui avec
    # l'instruction utc pour le créneau horaire.
    score.last_game = datetime.datetime.now(datetime.timezone.utc)
    You, 1 second ago • Uncommitted changes
    # Création d'une variable pour connaître la différence entre last_game et maintenant.
    difference = now - score.last_game

    # Condition si la différence est sup à 1 jour.
    if difference.days >= 1:
        score.current_score = 0

```

La mise en place des réponses:

Plus haut dans le code, vous aurez constaté la mise en place de « current » et « correct ». Les deux variables, me permettent de faire une comparaison.

Si « current » et « correct » sont égaux alors:

Vérifier quelle catégorie d'idéogrammes sont actives, attribuer les points dans le bon champs de score puis sauvegarder celui-ci. Cela permet aussi de tenir le compte du nombre de question posé et de mettre à jour le champs concerné.

Sinon :

Tenir compte de la question et sauvegarder.

```

# si la réponse est la bonne: ajout du point dans les colonnes score
# et ajoute 1 dans le nombre de question posé.
if correct == current:

    score.total_questions += 1
    score.current_score += 1
    score.scores_max += 1

    # condition pour savoir que faire pour les boutons actifs.
    if hiragana == "on" and katakana == "off":
        score.score_hiragana += 1
    elif katakana == "on" and hiragana == "off":
        score.score_katakana += 1
    elif katakana == "on" and hiragana == "on":
        score.score_katakana += 1
        score.score_hiragana += 1
    score.save()

# si réponse n'est pas la bonne, ajoute 1 dans le nombre de
# question posé et sauvegarde dans la table score.
else:
    score.total_questions += 1
    score.save()

# permet de rajouter quelque chose dans le context.

context['scores'] = score

return render(request, "mysite/maneki.html", context)

```

Mise en place des variables dans le HTML :

Ici vous pouvez retrouver deux élément avec la class « child », presque identique mais contenant chacune un idéogramme différent, l'un étant la bonne réponse et l'autre la mauvaise. Entre celle-ci un élément de class « child2 » qui est l'image du bon idéogramme.

```
<!-- A form that is being submitted when the button is clicked. -->
<span class="child">
  <form method="post">
    {% csrf_token %}
    <input type="hidden" name="current" value="{{ideogramms.0.id}}">
    <input type="hidden" name="correct" value="{{correct_ideogramm.id}}">

    <button type="submit">{{ideogramms.0.romanji }}</button>
  </form>
</span>

<!-- The image that is being displayed. -->
<span class="child2"> You, last month • ajout de fonction pagination et blog ...
  
</span>

<!-- A form that is being submitted when the button is clicked. -->
<span class="child">
  <form method="post">
    {% csrf_token %}
    <input type="hidden" name="current" value="{{ideogramms.1.id}}">
    <input type="hidden" name="correct" value="{{correct_ideogramm.id}}">

    <button type="submit">{{ideogramms.1.romanji }}</button>
  </form>
</span>
```

VII. PRÉSENTATION DU JEU D'ESSAI ÉLABORÉ PAR LE CANDIDAT DE LA FONCTIONNALITÉ LA PLUS REPRÉSENTATIVE.

Dans cette partie, je teste l'application Lexique et vérifie qu'il est possible :

- d'ajouter des mots ou expressions sur la page Lexique,
- d'afficher les mots créés,
- de modifier les mots ou expressions,
- de supprimer les mots ou expressions.

Préparation du jeu d'essai:

J'ai choisi d'ajouter 5 mots et 5 expressions, avec pour chaque: une image, une traduction et leurs idéogrammes.

Résultats:

Les mots et expressions sont bien ajoutés à la base de données et ils s'affichent bien dans la page Lexique. Je peux consulter les pages Lexique concernées.

Je peux modifier et supprimer les mots ou expressions.

VIII. DESCRIPTION DE LA VEILLE, EFFECTUÉE PAR LE CANDIDAT DURANT LE PROJET, SUR LES VULNÉRABILITÉS DE SÉCURITÉ.

Lors de la veille sur les vulnérabilités de sécurité, je me suis informé sur les différentes mesures de protection. Dans le cadre de ce projet, je me suis principalement renseigné sur les failles XSS (Cross-Site-Scripting). Cette faille dangereuse, permet l'injection de code HTML ou JavaScript dans une base de données s'il n'y a pas de vérification du côté client et serveur.

Afin de réduire les risques sur ma page web, je vérifie via l'envoi d'un csrf token chaque donnée saisie avant de les envoyer.

IX. DESCRIPTION D'UNE SITUATION DE TRAVAIL AYANT NÉCESSITÉ UNE RECHERCHE, EFFECTUÉE PAR LE CANDIDAT DURANT LE PROJET

Dans mon projet, j'utilise les deux méthodes, Get et Post.

La méthode GET (visible dans l'URL) pour la récupération de données pour le jeu de mémoire.

La méthode POST pour par exemple l'envoi de donnée lors de la création d'un compte utilisateur inscrit ou pour la création du glossaire dans le Lexique.

A) Recherches :

Qu'est-ce qu'un jeton CSRF ?

Un jeton CSRF est un jeton aléatoire sécurisé utilisé pour empêcher les attaques CSRF.

Celui-ci doit avoir une grande valeur aléatoire pour le rendre difficile à deviner. Une application sécurisée CSRF attribue un jeton unique pour chaque session utilisatrice.

B) Recherche anglophone :

Mes connaissances en anglais étant limitées, mes recherches ont été orientées par les mots clefs : CSRF , Token , Django, Security.

- <https://www.w3schools.com>
- <https://stackoverflow.com>
- <https://fr.barracuda.com/glossary/csrf>
- <https://www.stackhawk.com/blog/django-csrf-protection-guide/>

X. EXTRAIT DU SITE ANGLOPHONE, UTILISÉ DANS LE CADRE DE LA RECHERCHE DÉCRITE PRÉCÉDEMMENT, ACCOMPAGNÉ DE LA TRADUCTION DU CANDIDAT (ENVIRON 750 CARACTÈRES)

HTTP Request: GET vs. POST

Two commonly used methods for a request-response between a client and server are: GET and POST.

- GET - Requests data from a specified resource
- POST - Submits data to be processed to a specified resource

GET is basically used for just getting (retrieving) some data from the server. Note: The GET method may return cached data.

POST can also be used to get some data from the server. However, the POST method NEVER caches data, and is often used to send data along with the request.

To learn more about GET and POST, and the differences between the two methods, please read our HTTP Methods GET vs POST chapter

SOURCE: <https://www.w3schools.com>

Demande HTTP : GET vs POST

Les deux méthodes couramment utilisées pour une requête ou réponse entre un client et un serveur sont : GET et POST.

- GET – Demande des données à partir d'une ressource spécifiée.
- POST – Soumet les données à traiter à une ressource spécifiée.

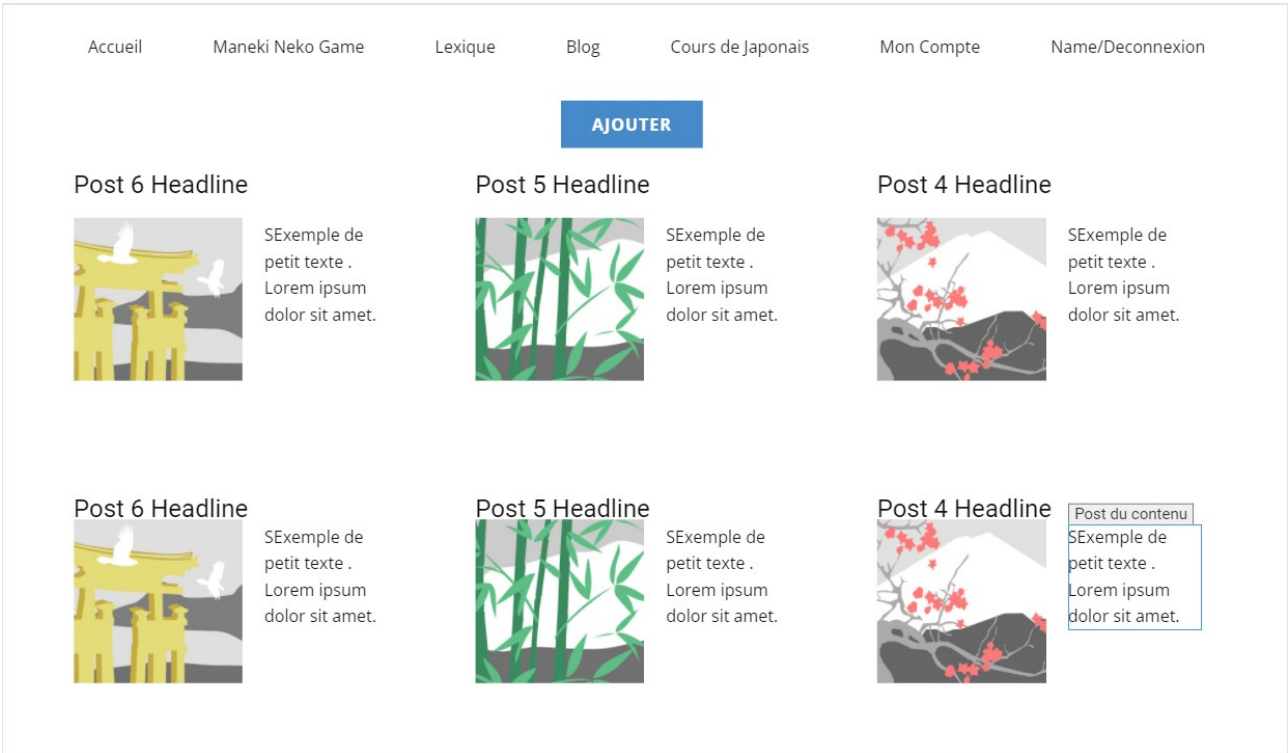
GET est fondamentalement utilisé pour simplement récupérer quelques données du serveur.
Remarque : La méthode GET peut renvoyer des données mises en cache.

POST peut également être utilisé pour obtenir des données du serveur. Cependant, la méthode POST ne cache JAMAIS de données, et est souvent utilisée pour envoyer des données avec une requête.

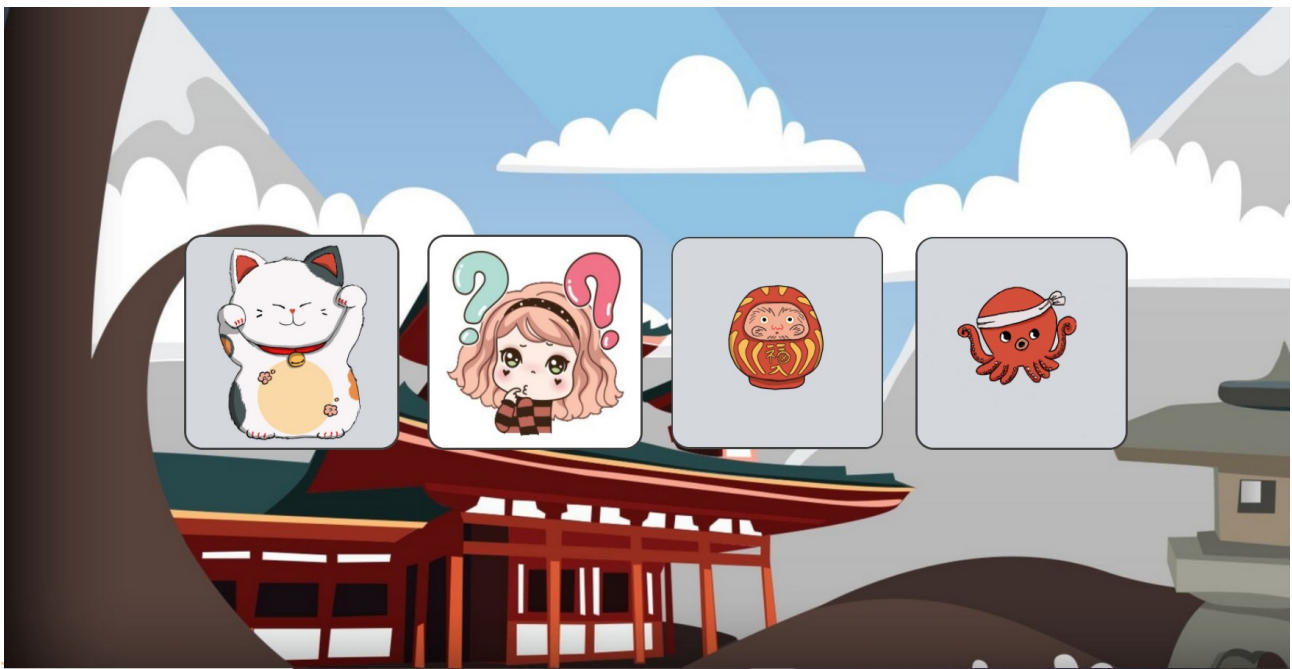
Pour en savoir plus sur GET et POST, et les différences entre les deux méthodes, veuillez lire notre chapitre Méthodes HTTP GET vs POST.

ANNEXE I (WIREFRAMES)

Le Lexique



La page d'accueil.



ANNEXE 2 (MODELE DE DONNÉES)

