Une image contenant motif, art, conception

Description générée automatiquement

Stellar

Voyagez atypique

Dossier de synthèse

**Bertuzzi Maxime**

DWWM – Novembre 2023

# Une image contenant motif, art, conception Description générée automatiquementAvant-Propos

Stellar

Voyagez atypique

Dossier de synthèse

La vie est souvent comparée à un voyage, rempli de découvertes et de transformations.

Mes propres voyages, notamment mon périple à pied et en autostop de Stockholm à Colmar et mon aventure en camping-car à travers les États-Unis, ont profondément influencé ma vision du monde et ma carrière.

Ces expériences ont renforcé mon désir de créer des projets qui marient fonctionnalité et esthétique, des valeurs que j'ai d'abord explorées en tant que dessinateur en architecture.

Dans ce domaine, j'ai appris à voir au-delà des simples structures, à comprendre que chaque projet architectural répond à des besoins humains et raconte une histoire.

Aujourd'hui, je transpose ces principes dans le monde du développement web, où je m'efforce de créer des expériences utilisateur significatives.

Le projet Stellar, que je vous présente dans ce dossier, est l'incarnation de cette philosophie.

Il ne s'agit pas simplement d'une plateforme de réservation, mais d'un moyen de connecter les personnes en quête d'aventures avec des logements qui ont du caractère et une histoire à partager.

Ce dossier vous guidera à travers les différentes étapes de la création de Stellar, des technologies employées aux défis surmontés.

Plus qu'un simple compte-rendu, il reflète la vision qui a guidé chaque décision de conception et chaque ligne de code.

Bienvenu dans Stellar !

# Table des matières

[1. Avant-Propos 1](#_Toc151288887)

[2. Table des matières 2](#_Toc151288888)

[3. Introduction au Projet Stellar 5](#_Toc151288889)

[ Contexte du projet 5](#_Toc151288890)

[ Objectif du projet 5](#_Toc151288891)

[4. Cahier des charges 6](#_Toc151288892)

[ Utilisateur final : Ce qu'ils peuvent faire 6](#_Toc151288893)

[ La Map 6](#_Toc151288894)

[ La réservation 6](#_Toc151288895)

[ Pagination des annonces 6](#_Toc151288896)

[ Barre de filtres 7](#_Toc151288897)

[ Propriétaires : Gestion des annonces 8](#_Toc151288898)

[5. Gestion de Projet 8](#_Toc151288899)

[ Méthodologies Agiles et Trello (Annexe. Gestion de Projet) 8](#_Toc151288900)

[ Comprendre la méthode MoSCoW 8](#_Toc151288901)

[ Ma façon d'utiliser Trello avec MoSCoW 8](#_Toc151288902)

[6. Benchmark, Analyse Concurrentielle 10](#_Toc151288903)

[ Qu'est-ce que le Benchmark et comment s'y prendre 10](#_Toc151288904)

[ Benchmark pour Stellar : 10](#_Toc151288905)

[ Airbnb : 10](#_Toc151288906)

[ Autres sites d'hébergements atypiques : 10](#_Toc151288907)

[ Analyse Concurrentielle 11](#_Toc151288908)

[7. Cibles et Personas 12](#_Toc151288909)

[ Comment bien cibler les clients et créer des Personas ? 12](#_Toc151288910)

[ Personas: 12](#_Toc151288911)

[8. Technologies et Compétences 14](#_Toc151288912)

[ Front-End Technologies utilisées et leur approche 14](#_Toc151288913)

[ HTML et TWIG 14](#_Toc151288914)

[ SCSS et Bootstrap 15](#_Toc151288915)

[ JavaScript et jQuery 15](#_Toc151288916)

[ Npm 16](#_Toc151288917)

[ Google Platform pour la Map 16](#_Toc151288918)

[ Bootstrap Datepicker pour le Calendrier 17](#_Toc151288919)

[ Back-End Technologies utilisées et leur approche : 17](#_Toc151288920)

[ Composer 18](#_Toc151288921)

[ MySQL 18](#_Toc151288922)

[9. Architecture et MVC 19](#_Toc151288923)

[ Introduction 19](#_Toc151288924)

[ Qu'est-ce que le Modèle-Vue-Contrôleur (MVC) ? 19](#_Toc151288925)

[ Pourquoi Utiliser MVC ? 20](#_Toc151288926)

[10. Design et Expérience Utilisateur (UX/UI) 22](#_Toc151288927)

[ Wireframes et Maquettage (Annexe. UX/UI) 22](#_Toc151288928)

[ Pourquoi les Wireframes ? 22](#_Toc151288929)

[ Maquettage : Le Prochain Niveau 22](#_Toc151288930)

[ Accessibilité et RGAA (Référentiel Général d'Amélioration de l'Accessibilité) 22](#_Toc151288931)

[ Qu'est-ce que le RGAA ? 23](#_Toc151288932)

[ Pourquoi c'est important ? 23](#_Toc151288933)

[ Accessibilité et UX/UI Design 23](#_Toc151288934)

[ Choix de Couleurs et Typographie 24](#_Toc151288935)

[ Couleurs 24](#_Toc151288936)

[ Typographie 24](#_Toc151288937)

[11. Plan du Site 26](#_Toc151288938)

[12. Conformité RGPD 27](#_Toc151288939)

[ Qu’est-ce que le RGPD ? 27](#_Toc151288940)

[ Liste des Règles à Suivre 27](#_Toc151288941)

[ Mise en Œuvre sur Stellar 28](#_Toc151288942)

[ Conclusion 29](#_Toc151288943)

[13. Fonctionnalités et Interaction 30](#_Toc151288944)

[ Authentification et Gestion de Compte 30](#_Toc151288945)

[ Login : Vue et Contrôleur 30](#_Toc151288946)

[ Signup : Vue et Formulaire 30](#_Toc151288947)

[ Hashage du mot de passe 30](#_Toc151288948)

[ Système de Réservation 31](#_Toc151288949)

[ Interface Utilisateur : Vue de Détail et Calendrier (detail.html.twig, annexe : Calendrier) 32](#_Toc151288950)

[ Map (Annexe. Map) 33](#_Toc151288951)

[ Introduction 33](#_Toc151288952)

[ Google Maps JavaScript API 33](#_Toc151288953)

[ Geocoder 34](#_Toc151288954)

[ Marqueurs et Étiquettes 34](#_Toc151288955)

[ InfoWindow et Interaction Utilisateur 35](#_Toc151288956)

[ Ajout d’annonce (Annexe : Add Advert) 35](#_Toc151288957)

[ Formulaire d'Ajout d'Annonce avec Twig 36](#_Toc151288958)

[ Interaction Utilisateur avec JavaScript 36](#_Toc151288959)

[ Recherche personnalisée par filtres 37](#_Toc151288960)

[ La technique 37](#_Toc151288961)

[14. Fonctionnalité Phare 40](#_Toc151288962)

[ Bootstrap Datepicker 40](#_Toc151288963)

[ Introduction 40](#_Toc151288964)

[ Backend : 40](#_Toc151288965)

[ Frontend : 40](#_Toc151288966)

[15. SEO et Optimisations 43](#_Toc151288967)

[ Introduction au SEO 43](#_Toc151288968)

[ Techniques de SEO sur Stellar et Leur Importance : 43](#_Toc151288969)

[ Mots-clés et Métadonnées : 43](#_Toc151288970)

[ Alt Text : 43](#_Toc151288971)

[ Optimisation du Temps de Chargement: 44](#_Toc151288972)

[ Structure HTML et Balisage Sémantique: 45](#_Toc151288973)

[ Mobile-Friendly et Responsive Design: 45](#_Toc151288974)

[ Contenu de Qualité : 45](#_Toc151288975)

[ Liens interne (et slugs) et Liens Externes : 46](#_Toc151288976)

[16. Sécurité du Site Web 49](#_Toc151288977)

[ Introduction à la Sécurité Web 49](#_Toc151288978)

[ Sécurité utilisée dans Stellar 49](#_Toc151288979)

[ Hashage de mot de passe 49](#_Toc151288980)

[ Le Token CSRF (Cross-Site Request Forgery): 49](#_Toc151288981)

[ Expressions Régulières (REGEX) et Attaque par Dictionnaire 50](#_Toc151288982)

[ Défense contre les attaques par force brute 51](#_Toc151288983)

[ Prévention de la faille XSS (Cross-Site Scripting) : 52](#_Toc151288984)

[ Protection contre les injections SQL : 53](#_Toc151288985)

[ Les Conditions 54](#_Toc151288986)

[ Les autorisations : 55](#_Toc151288987)

[17. Axes d'Amélioration et Futur du Projet 56](#_Toc151288988)

[ Introduction 56](#_Toc151288989)

[ Axes d'Amélioration 56](#_Toc151288990)

[ Futur du Projet 57](#_Toc151288991)

[18. Annexes 58](#_Toc151288992)

[ Fonctionnalités 58](#_Toc151288993)

[ Calendrier et réservation 58](#_Toc151288994)

[ Map 61](#_Toc151288995)

[ Wireframes et Prototypes 62](#_Toc151288996)

[ Schémas de Base de Données (MCD, MLD) 64](#_Toc151288997)

[ Suivi de Projet avec Trello 65](#_Toc151288998)

# Introduction au Projet Stellar

## Contexte du projet

Dans un monde qui **évolue** constamment, marqué par une **urbanisation croissante**, la recherche **d'expériences de voyage uniques et mémorables** est devenue une chose quasiment **indispensable** pour les voyageurs.

C'est dans ce contexte **passionnant et personnel** que j’ai décidé de faire **Stellar**, une plateforme dédiée à la **location de logements atypiques**.

L'essence même de Stellar est de répondre à la demande qui croissante des voyageurs en quête d'hébergements qui **changent du quotidien** en proposant une **alternative** **aux hôtels et aux appartements/maisons** **traditionnels**.

## Objectif du projet

L'objectif premier de ce projet est double : d'une part, **fournir aux propriétaires de logements atypiques une plateforme sécurisée et conviviale** pour partager leurs biens et d’autre part, offrir aux voyageurs une **variété d'options** de logement qui **dépasse les hébergements classiques**.

À long terme (si le projet est mis en production), Stellar aspirera à devenir la référence en matière de location de logements atypiques.

# Cahier des charges

## Utilisateur final : Ce qu'ils peuvent faire

### La Map

À l'ère du numérique, la **navigation visuelle** est plus importante que jamais. C'est pourquoi Stellar intègre une **carte interactive** sur la page d'accueil et sur la page de détails d’annonce.

Cette carte utilise des technologies pour **afficher les annonces** mises en ligne en temps réel.

Chaque emplacement sur la carte est représenté par un **label** avec le prix du logement (**cliquable** pour aller directement sur le détail de l’annonce).

En passant la souris sur ce label, une **fenêtre contextuelle** affiche une image du logement et le titre de l'annonce, donnant à l'utilisateur un **aperçu instantané** sans avoir à quitter la page.

### La réservation

Stellar comprend que **chaque utilisateur est unique**. C'est pourquoi j’ai développé un système de réservation **simplifié** mais robuste.

Après s'être inscrit et connecté, un utilisateur peut cliquer sur une annonce pour accéder à **une page de détails pleine d’informations** (description, photos, prix, commodités, etc).

À partir de là, il peut facilement **choisir des dates et réserver** le logement sur une page où j’ai essayé d’être le plus intuitive possible.

### Pagination des annonces

Pour offrir une navigation **plus rapide** et **moins encombrée**, le système **pagine automatiquement** les annonces au-delà des 14 premières visibles sur la page d'annonces.

Ce n'est pas qu’une question d’esthétique, mais un choix influencé par des études d'UX qui montrent qu'un trop grand nombre d'options peut rendre l'expérience utilisateur **écrasante** et **inefficace**.

* Le code en **back-end** utilise des algorithmes de pagination pour **récupérer seulement les annonces nécessaires de la base de données, réduisant ainsi la charge sur le serveur et accélérant le temps de chargement de la page**.
* Côté **front-end**, une barre de navigation de pagination est **dynamiquement générée** en bas de la liste d'annonces, permettant aux utilisateurs de **naviguer facilement** entre les différentes pages.

### Barre de filtres

L'une des fonctionnalités les plus « puissantes » de Stellar est sa **barre de filtres**, située en haut de la page d'annonces.

Ça permet à l'utilisateur de **trier** les annonces selon **plusieurs critères**, offrant une **personnalisation** qui va au-delà de la recherche textuelle basique.

#### Par Prix

Les utilisateurs peuvent **définir une fourchette de prix** en utilisant deux curseurs interactifs.

Ce n'est pas juste une requête de base de données, le système **ajuste dynamiquement** les annonces visibles **en temps réel** grâce à des appels **AJAX**, sans avoir besoin de recharger la page.

#### Par Ville ou Pays

Grâce à un input, les utilisateurs peuvent **filtrer les annonces par ville ou pays.**

Ils peuvent appuyer sur entrée pour ajouter chaque filtre qui s’affichera en petites étiquettes en dessous de leur input respectif avec la possibilité de supprimer chaque **étiquettes** unes à unes.

#### Par Disponibilité

La **recherche de disponibilité** est une autre fonction de filtrage.

Les utilisateurs peuvent **sélectionner** des **dates d'arrivée et de départ** deux inputs et le système filtre alors les annonces disponibles pendant cette période.

Cela nécessite une logique de back-end qui **vérifie les dates de disponibilité de chaque** logement et prend également en compte les **réservations existantes**.

Cette barre de filtres est une intégration plus compliquée qui comprend du **front-end etdu back-end**, utilisant des technologies comme **JavaScript, AJAX et PHP** pour créer une **expérience utilisateur fluide**, **réactive** et **sans soucis**.

Les requêtes de filtrage sont optimisées pour **être aussi légères que possible**, afin de **minimiser le temps de latence** et d'assurer que **l'interface reste rapide et réactive** même sur les connexions Internet plus lentes ou les appareils moins puissants.

## Propriétaires : Gestion des annonces

Les propriétaires peuvent **facilement créer des annonces** en utilisant un **formulaire intuitif** qui guide à travers chaque étape, du titre et de la description jusqu'aux équipements avec une interface faites pour que la page de création d’annonce ressemble quasi **identiquement à la page de détail d’annonce**.

De plus, le **service d'upload** d'images intégré permet d'ajouter des photos en **glissant-déposant** simplement les fichiers ou bien en cliquant simplement dessus, rendant le **processus plus efficace.**

# Gestion de Projet

## Méthodologies Agiles et Trello (Annexe. Gestion de Projet)

Dans mon parcours personnel de gestion de projet, j'ai adopté une approche Agile, valorisant la flexibilité et une capacité d'adaptation rapide. Une composante clé de ma méthode est la technique de priorisation MoSCoW, qui m'a permis de trier efficacement les exigences de mes projets.

### Comprendre la méthode MoSCoW

La méthode MoSCoW m'a aidé à classifier mes tâches en fonction de leur importance :

**- Must have :**

Ce sont les tâches essentielles, celles sans lesquelles mon projet ne peut être considéré comme complet.

**- Should have :**

Ce sont les tâches importantes qui améliorent le projet mais ne sont pas cruciales pour son lancement.

**- Could have :**

Ce sont les tâches que je pourrais faire si j'ai suffisamment de temps et de ressources, elles ne sont pas prioritaires.

**- Won’t have :**

Ce sont les tâches que j'ai décidé de ne pas traiter dans le cadre actuel du projet.

### Ma façon d'utiliser Trello avec MoSCoW

J'utilise Trello pour appliquer la méthode MoSCoW à mes projets solo. Mon tableau Trello est organisé avec des listes qui reflètent les catégories MoSCoW, me fournissant une visualisation claire de là où je dois concentrer mes efforts :

**- Planification :**

Je définis mes tâches et les place dans la liste appropriée, ce qui me donne un plan d'action clair.

**- Exécution :**

Je m'attaque d'abord aux "Must have", pour m'assurer que les fondations de mon projet sont solides.

**- Évaluation :**

Je réévalue régulièrement mes priorités pour m'adapter à l'évolution du projet, garantissant ainsi la meilleure allocation de mes ressources.

**- Adaptation :**

La flexibilité de Trello me permet de repositionner facilement les tâches en fonction des changements et des nouvelles informations.

**Conclusion**

Intégrer la méthode **MoSCoW** dans ma gestion de projet m'a offert une structure qui guide mes décisions et assure que je traite les aspects les plus **importants** **en premier**. Cela a été essentiel pour atteindre mes objectifs efficacement et maintenir une qualité de travail élevée.

# Benchmark, Analyse Concurrentielle

## Qu'est-ce que le Benchmark et comment s'y prendre

Le benchmarking est une analyse comparative entre une entreprise et les leaders du secteur. L'objectif est d'identifier des axes d'amélioration et de fixer des objectifs réalisables.

Il faut analyser ces données pour repérer les forces et faiblesses de l’idée du siteet créer des plans d'action pour aborder les domaines qui nécessitent une amélioration.

Il ne faut pas oublier que le benchmarking est un processus continu, nécessitant des ajustements réguliers la stratégie en fonction des évolutions du marché.

Le benchmarking et les personas se complètent. Utilisés ensemble, ils fournissent une feuille de route solide pour ajuster les produits et services aux besoins du marché, favorisant ainsi la croissance d’une entreprise.

**Liens vers les sites qui ont étés analysés :**

* [**https://www.airbnb.fr/**](https://www.airbnb.fr/)
* [**https://www.espaces-atypiques.com/**](https://www.espaces-atypiques.com/)
* [**https://www.abracadaroom.com/fr/**](https://www.abracadaroom.com/fr/)
* [**https://www.cabanes-de-france.com/hebergements-insolites-regions/**](https://www.cabanes-de-france.com/hebergements-insolites-regions/)

## **Benchmark pour Stellar :**

### Airbnb :

**Points forts :**

Grande variété d'hébergements, interface utilisateur conviviale, processus de réservation simple.

**Points faibles :** Pas vraiment de points faibles, ce sont les leaders du marché.

### Autres sites d'hébergements atypiques :

**Points forts :**

Offrent des options d'hébergement uniques qui se démarquent d'Airbnb.

**Points faibles :**

Les sites peuvent être moins conviviaux et plus difficiles à naviguer.

Souvent, ils n'ont pas l'air aussi modernes ou ne sont pas aussi optimisés pour le mobile.

### Analyse Concurrentielle

#### Expérience utilisateur :

Mon objectif avec Stellar est de surpasser les concurrents en termes d'expérience utilisateur, notamment sur le front-end.

Mon expérience en design pourrait me donner la capacité de créer une interface à la fois visuellement attrayante et intuitive.

#### Unicité des hébergements :

Je cible un créneau plus spécifique en mettant en avant des hébergements vraiment uniques.

Cela devient mon argument de vente unique (USP, (Unique Selling Proposition)).

#### Technologie :

L'utilisation de technologies front-end et back-end modernes me donne un avantage en termes de performances du site, de SEO et d'expérience utilisateur globale.

#### Communauté et Confiance :

En intégrant des fonctionnalités sociales et un système de revues robuste, je souhaite instaurer un sentiment de communauté et de confiance qui pourrait devenir un autre avantage concurrentiel.

Pour finir, mon analyse concurrentielle et ma compréhension des personas me placent dans une bonne position pour créer un produit qui répond aux besoins des voyageurs modernes.

Je compte sur mon désir d'exceller en design front-end comme un réel différenciateur sur le marché.

# Cibles et Personas

## Comment bien cibler les clients et créer des Personas ?

Avant commencer le benchmarking, j'ai défini mes cibles et personas pour Stellar.

Comprendre mes clients passe par la création de personas, des profils fictifs mais basés sur des données réelles.

Le but ?

Comprendre leurs besoins et comportements pour mieux répondre à leurs attentes.

Ces personas sont ensuite devenus le plus important de ma stratégie, influençant le développement du site, le marketing et le service client.

Il est important de noter que les personas ne sont pas fixes. Les besoins des clients évoluent, tout comme le marché.

## Personas:

Pour mieux comprendre mon public cible et afin de créer une expérience utilisateur plus personnalisée, j'ai développé plusieurs personas :

#### L'Aventurier Urbain :

Cette persona représente le voyageur qui aime la découverte urbaine mais veut s'éloigner des options d'hébergement génériques.

Elle est à la recherche de lieux qui ont une âme et qui reflètent la culture et l'ambiance unique de la ville qu'elle visite.

Les détails comme une décoration locale authentique ou des conseils personnalisés sur des spots méconnus en ville font toute la différence pour elle.

#### Le Globe-Trotter Éco-conscient :

Cette persona incarne l'individu soucieux de l'environnement, cherchant à minimiser son empreinte écologique même lorsqu'il voyage.

Il apprécie les initiatives durables telles que le recyclage, l'usage d'énergies renouvelables dans le logement ou encore des options de transport éco-responsable à proximité.

#### Le Nomade Numérique :

Ce profil est celui du professionnel moderne qui a la liberté de travailler à distance et veut combiner travail et découverte du monde.

Pour lui, des facteurs comme un Wi-Fi fiable, un espace de travail ergonomique et des équipements modernes sont cruciaux.

Il apprécie également la flexibilité dans les conditions de réservation, étant donné que ses plans peuvent changer rapidement.

**Conclusion**

Dans ce chapitre, on a exploré comment définir des personas pour Stellar.

En développant des profils comme l'Aventurier Urbain, le Globe-Trotter Éco-conscient et le Nomade Numérique, on peut mieux comprendre les différents types de clients potentiels. Cette méthode a permis d'imaginer des services adaptés à chaque groupe.

C'est une bonne base pour un projet comme le celui-ci, où comprendre le client est clé. En gardant un œil sur l'évolution des tendances, on peut ajuster nos idées pour qu'elles restent pertinentes.

# Technologies et Compétences

## Front-End Technologies utilisées et leur approche

Pour la structure et le design de Stellar nous avons utilisé les bases du développement web, tel qu’HTML, CSS et SCSS, mais pour plus de dynamisme et de modernité, javascript et jQuery ont aussi été d’une grande utilité pour une expérience utilisateur plus agréable et immersive.

### HTML et TWIG

#### HTML

Le HTML (HyperText Markup Language) est fondamental pour n'importe quel site web.

Il sert de charpente à toute la structure, fournissant la base pour le contenu, la mise en page et l'interactivité.

Assez à première vue, un HTML bien structuré est crucial pour l'accessibilité et le SEO (Search Engine Optimization).

Il utilise une série de balises pour organiser le contenu, et chaque balise a son propre rôle et sa propre importance.

#### Une image contenant plante d’intérieur, plante, pot de fleurs, vase Description générée automatiquementTwig

Twig est un moteur de templates pour PHP. Il agrandit les capacités du HTML en y intégrant une logique côté serveur, ce qui permet de créer des vues dynamiques.

Ainsi, plutôt que de devoir coder à la main chaque variante possible d'une page en HTML pur, on peut utiliser des variables, des boucles et des conditions pour rendre le contenu plus dynamique.

Avec Twig, le code est plus propre, plus facile à maintenir et plus modulable.

**Spécificités de leur utilisation dans Stellar**

Dans Stellar, HTML sert de base pour toutes les pages, tandis que Twig est utilisé pour injecter des données dynamiques comme la liste des annonces ou les informations utilisateur.

En combinant HTML et Twig, il est possible de créer une interface utilisateur cohérente qui s'adapte aux besoins de chaque visiteur.

### SCSS et Bootstrap

#### SCSS

SCSS (Sassy CSS) est un préprocesseur qui étend les capacités du CSS.

Il ajoute des fonctionnalités comme les variables et les fonctions, ce qui rend le code plus propre, plus réutilisable, et plus facile à maintenir.

Le SCSS est ensuite compil é en CSS standard qui peut être interprété par les navigateurs web.

#### Bootstrap

Bootstrap est un framework CSS open-source qui permet de développer des interfaces utilisateurs modernes et réactives avec une base de code cohérente.

Il fournit une multitude de composants pré-stylisés, ce qui accélère grandement le processus de développement.

**Spécificités de leur utilisation dans Stellar :**

Dans Stellar, SCSS est utilisé pour une grande partie du projet et pour modifier certaines parties du bootstrap.

Il a été très utile pour pouvoir ajouter par exemple des couleurs en variables et entre ensuite réutilisables tout au long du projet sans se tromper dans une teinte.

Bootstrap, quant à lui, n’a finalement pas été utilisé mise à part pour le Datepicker précisé plus bas.

### JavaScript et jQuery

#### Une image contenant Police, Graphique, logo, conception Description générée automatiquementJavaScript

JavaScript est le langage de programmation qui anime le web. Il est responsable de l'interactivité et de la dynamique des pages.

Par exemple, lorsque vous cliquez sur un bouton et qu'un pop-up apparaît, c'est probablement grâce à JavaScript.

Il fonctionne côté client, ce qui signifie que le code est exécuté sur l'ordinateur de l'utilisateur, rendant les interactions rapides et fluides.

#### jQuery

jQuery est une bibliothèque JavaScript qui simplifie la manipulation du Document Object Model (DOM), la gestion des événements, et les appels AJAX.

Elle a été créée pour écrire moins de code tout en faisant plus de choses.

Alors que JavaScript natif peut être plus complexe pour certaines tâches, jQuery les rend plus simples.

**Spécificités de leur utilisation dans Stellar**

Dans Stellar, JavaScript et jQuery sont utilisés pour une multitude de fonctionnalités, y compris le filtrage des annonces, la pagination et même les interactions avec la carte.

La combinaison de ces technologies permet de créer une expérience utilisateur riche et réactive. jQuery, en particulier, accélère le développement en offrant des solutions simples pour des tâches complexes comme les requêtes AJAX pour la mise à jour en temps réel des données affichées.

### Une image contenant Graphique, rouge, capture d’écran, Police Description générée automatiquementNpm

Npm (Node Package Manager) est un gestionnaire de paquets pour JavaScript.

Semblable à Composer, mais pour le monde JS, il permet de télécharger, de mettre à jour et de gérer les bibliothèques JavaScript que votre projet utilise.

**Spécificités de son utilisation dans Stellar :**

Dans Stellar, npm est utilisé pour gérer les packages JavaScript, comme jQuery ou même des outils de build comme Webpack.

Cela facilite la gestion des ressources front-end et fait que toutes les dépendances sont à jour.

### Google Platform pour la Map

Google Platform offre une suite d'outils pour la géolocalisation et la cartographie, dont Google Maps API.

Cette API permet d'intégrer facilement des cartes Google dans des applications web et d'y ajouter des fonctionnalités comme des marqueurs, des infobulles, etc.

**Spécificités de son utilisation dans Stellar**

Dans Stellar, Google Maps est utilisé pour afficher la localisation de chaque annonce sur une carte interactive.

Un utilisateur peut donc avoir une idée instantanée de la situation géographique des logements disponibles.

Chaque annonce est représentée par un marqueur et au survol de la souris, une infobulle apparaît avec des informations supplémentaires sur le logement.

Cela enrichit l'expérience utilisateur en fournissant des informations visuelles à chaque annonce.

### Bootstrap Datepicker pour le Calendrier

Bootstrap Datepicker est un plugin jQuery qui ajoute un sélecteur de date au formulaire.

Il est utilisé pour faciliter la sélection des dates dans des champs de formulaire.

**Spécificités de son utilisation dans Stellar**

Dans le système de réservation de Stellar, le Datepicker est indispensable pour la sélection des dates d'arrivée et de départ.

Il simplifie ce processus en fournissant une interface utilisateur intuitive.

Les dates déjà réservées peuvent même être désactivées, évitant ainsi des conflits de réservation.

## Back-End Technologies utilisées et leur approche :

Pour une base solide et pour plus de rapidité j’ai choisi d’utiliser le framework Symfony accompagné de Composer pour gérer les dépendances PHP, puis MySQL pour la gestion de base de données, qui est indispensable pour ce type de projet !

#### Symfony

Symfony est un framework PHP robuste et flexible qui est utilisé pour construire des applications web complexes et évolutives.

Il fournit un ensemble d'outils et de composants réutilisables qui accélèrent le développement.

Ses gros avantages sont : son système de routage, sa facilité d'intégration avec des bases de données et son moteur de template Twig que j’ai défini plus haut.

**Spécificités de son utilisation dans Stellar**

Dans le cas de Stellar, Symfony sert de fondation au backend.

Le framework gère la logique métier, les interactions avec la base de données.

Il permet une organisation claire du code, ce qui rend le projet plus facile à maintenir et à évoluer.

### The Composer Official Logo: a male orchestra conductor with both arms in the air and his head tilted down, reading music sheetsComposer

Composer est un gestionnaire de dépendances pour PHP.

Il permet de gérer les bibliothèques et les outils que le projet utilise, de sorte qu’on n’ait pas à les télécharger manuellement ou à nous soucier des versions.

**Spécificités de son utilisation dans Stellar :**

Dans Stellar, Composer est utilisé pour gérer toutes les dépendances PHP du projet, y compris Symfony.

Il facilite l'ajout de nouvelles bibliothèques et assure que tout le monde travaille avec les mêmes versions, ce qui est crucial pour la cohérence et la stabilité du projet.

### Une image contenant Graphique, Police, graphisme, logo Description générée automatiquementMySQL

MySQL est un système de gestion de base de données relationnelle.

Il est populaire, facile à utiliser et efficace pour stocker, récupérer et gérer les données.

**Spécificités de son utilisation dans Stellar :**

Pour Stellar, il héberge toutes les informations relatives aux annonces, aux utilisateurs, aux réservations, etc.

La facilité d'intégration avec Symfony rend les interactions avec la base de données presque transparentes, grâce à l'ORM Doctrine, par exemple.

# Architecture et MVC

### Introduction

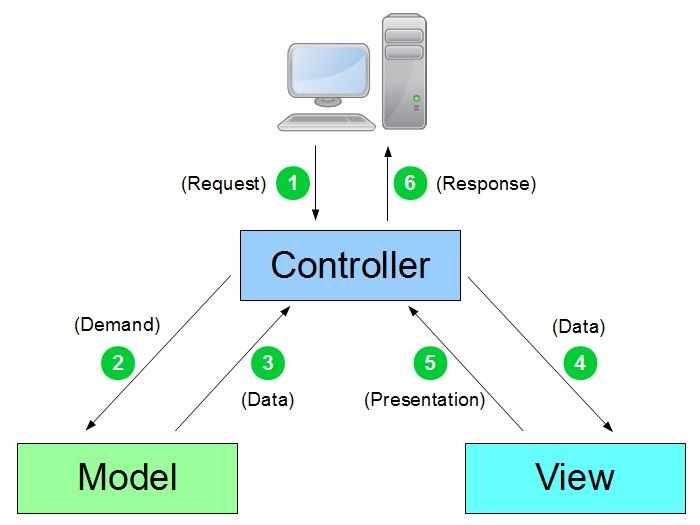
Dans le monde du développement logiciel, l'architecture logicielle sert de fondation à tout projet.

Il s'agit du plan initial qui oriente la structure, le comportement et du système à construire. Une bonne architecture peut faire la différence entre un projet réussi et un projet chaotique qui part dans tout les sens.

L'un des modèles d'architecture les plus populaires, surtout dans le développement web, est l'architecture Modèle-Vue-Contrôleur (MVC).

### Qu'est-ce que le Modèle-Vue-Contrôleur (MVC) ?

MVC est un modèle architectural qui sépare une application en trois composants interconnectés mais indépendants :



#### Modèle (Model) :

Le Modèle représente la logique métier et les données. Il est responsable de l'accès à la base de données, des requêtes, de l'insertion et de la mise à jour des données.

Le modèle est également chargé de notifier la Vue chaque fois que les données sont modifiées, pour que les changements puissent être affichés à l'utilisateur.

#### Vue (View) :

La Vue est essentiellement l'interface utilisateur. Elle affiche les données provenant du Modèle à l'utilisateur et envoie les commandes de l'utilisateur au Contrôleur.

#### Contrôleur (Controller) :

Le Contrôleur agit comme une interface entre le Modèle et la Vue.

Il prend les entrées de l'utilisateur via la Vue, les transforme en commandes pour le Modèle ou la Vue.

En d'autres termes, il gère les interactions entre le Modèle et la Vue.

### Pourquoi Utiliser MVC ?

L'utilisation du modèle MVC offre plusieurs avantages qui en font un choix populaire :

#### Séparation des préoccupations:

Le MVC permet une séparation claire des préoccupations, où la logique métier, l'interface utilisateur et l'interaction utilisateur sont séparées en différentes sections du code base.

Ca facilite la maintenance et le débogage, car les développeurs peuvent travailler sur un aspect du code sans avoir à toucher aux autres.

#### Réutilisabilité du code et modularité:

Les composants du Modèle sont souvent réutilisables et indépendants de l'interface utilisateur.

Cela signifie que vous pouvez réutiliser ces composants dans différents projets ou même au sein de différentes parties du même projet, ce qui peut réduire considérablement le temps et les efforts de développement.

#### Développement parallèle et scalabilité:

La séparation des préoccupations permet également de travailler sur différents composants en parallèle.

Par exemple, tandis que les développeurs front-end peuvent se concentrer sur la Vue, les développeurs back-end peuvent travailler sur le Modèle, chacun sans attendre l'autre.

**Conclusion sur l'Architecture et MVC**

L'architecture logicielle est plus qu'un simple plan pour construire une application, c'est la fondation sur laquelle repose tout le projet.

Le choix d'une bonne architecture peut simplifier le développement, faciliter la maintenance et même améliorer les performances.

L'architecture MVC s'est imposée comme un standard dans le développement web moderne grace à sa modularité, de sa réutilisabilité et de sa facilité d'adaptation à divers projets et équipes.

Sa séparation claire des préoccupations entre la logique métier (Modèle), l'interface utilisateur (Vue) et la logique de contrôle (Contrôleur) en fait un choix robuste pour tout projet web un peu complèxe.

# Design et Expérience Utilisateur (UX/UI)

## Wireframes et Maquettage (Annexe. UX/UI)

Avant de plonger dans le code et le design visuel d'un projet web, il est important de commencer par une phase de planification.

Cette étape peut prendre la forme de wireframes et de maquettes, qui servent de guides visuels pour l'expérience utilisateur (UX) et l'interface utilisateur (UI) du site web.

### Pourquoi les Wireframes ?

Les wireframes sont essentiellement les "squelettes" d'un site web.

Ils aident à visualiser où les éléments seront placés sur chaque page sans se soucier du design visuel.

C'est un peu comme l'équivalent en architecture de vos plans d'étage.

Le but est de s'assurer que tous les éléments essentiels sont présents et disposés de manière à faciliter l'interaction de l'utilisateur.

### Maquettage : Le Prochain Niveau

Le maquettage est l'étape suivante.

Alors que les wireframes se concentrent sur la fonctionnalité, les maquettes ajoutent une couche de design visuel.

C'est là que vous commencez à voir la "chair sur les os" que vous avez créée avec vos wireframes.

Le maquettage vous permet de tester vos choix de couleurs, vos polices et autres éléments de design dans un contexte qui ressemble beaucoup à votre produit final.

Les outils de maquettage comme Adobe XD, Sketch ou Figma vous permettent de créer des maquettes interactives.

On peut les utiliser pour tester le flux de navigation et même pour collecter des commentaires des utilisateurs avant de commencer le développement.

Pour « Stellar » uniquement Figma a été utilisé.

## Accessibilité et RGAA (Référentiel Général d'Amélioration de l'Accessibilité)

L'accessibilité web est un sujet qui devient de plus en plus important à mesure que le monde en ligne devient plus intégré dans notre quotidien.

Faire en sorte que notre site soit accessible signifie qu'il peut être utilisé par le plus grand nombre de personnes possible, y compris celles qui ont des handicaps.

En France, le RGAA sert de cadre pour mesurer l'accessibilité d'un site web.

### Qu'est-ce que le RGAA ?

Le RGAA, ou Référentiel Général d'Amélioration de l'Accessibilité, est un ensemble de directives conçu pour aider les développeurs à créer des sites web et des applications plus accessibles.

Ce référentiel est aligné avec les normes internationales d'accessibilité du Web, notamment les WCAG (Web Content Accessibility Guidelines).

Il est obligatoire pour les services publics en France de se conformer à ces directives.

### Pourquoi c'est important ?

Le RGAA n'est pas seulement une contrainte légale, il a aussi des avantages pratiques.

Un site accessible peut toucher un public plus large, améliorer l'expérience utilisateur pour tous et même améliorer le référencement (SEO) de votre site.

### Accessibilité et UX/UI Design

L'accessibilité est vraiment liée à un bon design UX/UI. Voici quelques points à soulignés :

#### Contraste des couleurs :

S’assurer que le texte est clairement lisible sur l'arrière-plan.

Des outils comme le Color Contrast Analyser (Utilisé pour ce projet) peuvent aider à vérifier cela. **(Annexe. UX/UI)**

#### Taille du texte :

Le texte devrait être redimensionnable sans perte de contenu ou de fonctionnalité. Une taille de police minimale de 16px est généralement recommandée pour le texte du corps.

#### Navigation au clavier :

Toutes les fonctionnalités doivent être accessibles via le clavier pour les utilisateurs qui ne peuvent pas utiliser une souris.

#### Étiquettes et rôles ARIA :

L'utilisation d'attributs ARIA (Accessible Rich Internet Applications) peut aider à rendre les éléments interactifs plus accessibles aux lecteurs d'écran.

#### Multimédia et Images :

Fournir des transcriptions pour l'audio et des sous-titres pour la vidéo ainsi que mettre des « alt » sur chaque image pour qu’une personne mal voyante puisse se référer à ça.

#### Formulaires :

S’assurer que les champs de formulaire sont correctement étiquetés et que les messages d'erreur sont clairs et faciles à comprendre.

**Conclusion**

Le respect des directives du RGAA et l'incorporation des principes d'accessibilité dans votre design UX/UI ne sont pas seulement des "bonnes pratiques" mais aussi des étapes importantes pour créer un environnement en ligne inclusif.

En considérant l'accessibilité dès le début, on construit un site qui est plus utilisable et agréable pour tous et on se met en conformité avec les réglementations en vigueur.

## Choix de Couleurs et Typographie

Le choix des couleurs et de la typographie n'est pas seulement une décision esthétique, il a également un impact significatif sur l'expérience utilisateur.

### Couleurs

Dans le cas du site "Stellar", l'utilisation de couleurs inspirées de l'espace évoque une atmosphère cosmique qui renforce le thème du site.

Le bleu profond et le violet, par exemple, peuvent évoquer un sentiment de mystère et d'infini.

Il est également crucial de penser à l'accessibilité lors du choix des couleurs.

### Typographie

La typographie est un autre élément qui peut faire ou défaire l'expérience utilisateur.

Des polices lisibles et bien espacées peuvent vraiment améliorer la lisibilité et la compréhension.

Dans "Stellar", l'utilisation de polices comme Poppins, Roboto et Inter ajoute de la modernité tout en conservant une excellente lisibilité.

* **Pacifico** : Utilisée pour le titre, c’est un font plutôt créative et enfantine
* **Poppins** : Police principale de Stellar pour une avoir une uniformité elle a été ajoutée directement au body pour qu’elle soit initialisée, c’est une font assez moderne.
* **Inter** : Elle a été alternée avec Poppins pour certaines parties où je souhaitais une police plus ronde et douce, ainsi qu’aux endroits où la taille de police était assez petite, particulièrement sur la page de détails d’annonces.

La cohérence est également importante.

Utiliser la même police ou une palette de polices cohérente à travers le site peut renforcer l'identité de la marque et améliorer l'expérience utilisateur.

En résumé, un bon design UX/UI est bien plus que de simples choix esthétiques.

C'est une combinaison de fonctionnalité, d'accessibilité et d'esthétique qui, lorsqu'elle est bien exécutée, crée une expérience utilisateur harmonieuse et engageante, qui ramènera donc plus de clientèle.

# Plan du Site

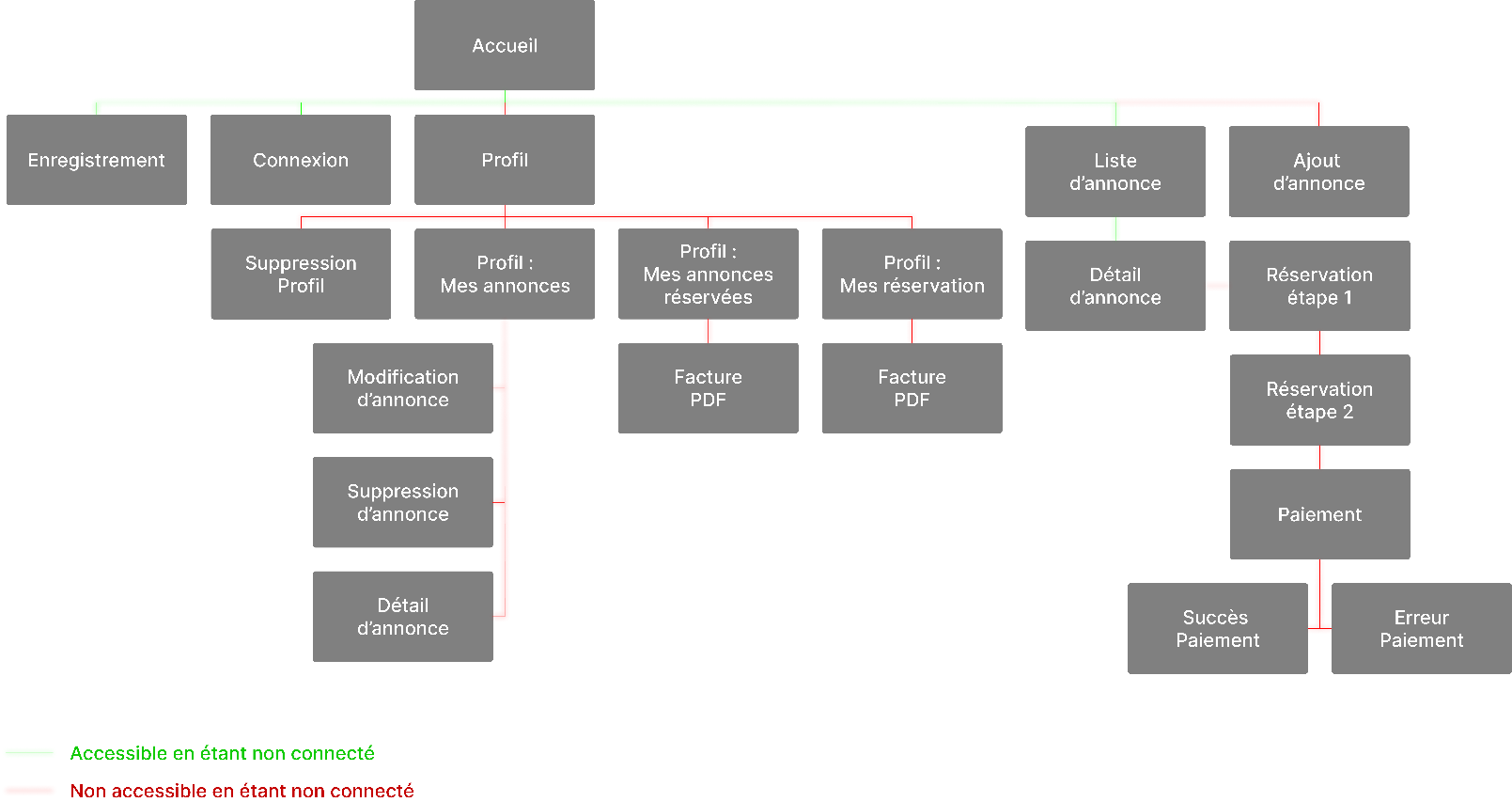
Stellar avait plusieurs possibilités d’accès :

**- Bloquer l’accès aux personnes non connectées à la page d’accueil :**

Au début la réservation se faisait immédiatement sur la page de détail, ce qui aurait donc donné la possibilité à des personnes non connectées de faire des réservations, ce qui n’est pas possible.

**-** **Autoriser l’accès à la page d’accueil et de détail :**

Finalement, trouvant ça assez réducteur j’ai autorisé l’accès aux personnes non connectées à la page de détail, avec une logique qui fasse en sorte que lorsqu’un user n’est pas connecté, un lien vers la page de connexion apparaisse.



# Conformité RGPD

## Qu’est-ce que le RGPD ?

Le Règlement Général sur la Protection des Données (RGPD) est un ensemble de lois établies par l'Union européenne en 2018.

Il vise à donner aux citoyens un meilleur contrôle sur leurs données personnelles et à garantir que les entreprises respectent ce droit.

Le non-respect du RGPD peut entraîner des sanctions financières sévères, d'où l'importance d'une mise en conformité stricte.

### Liste des Règles à Suivre

#### 1. Minimisation des Données

Il s'agit de collecter uniquement les données nécessaires à l’utilisation finale du projet. Par exemple, lors d'une réservation, on ne demandera que les informations essentielles comme l’identité, l'adresse e-mail, numéro de téléphoen et les dates de réservation.

#### 2. Finalité des Données

Chaque donnée collectée doit avoir une raison d'être. Par exemple, l'adresse e-mail est nécessaire pour la confirmation de réservation, tandis que le numéro de téléphone peut être utilisé en cas d'urgence ou de changement de dernière minute.

#### 3. Cookies

Pour la gestion des cookies, l'application utilise "Tarte au Citron", un gestionnaire de consentement aux cookies, qui permet aux utilisateurs de choisir les cookies qu'ils souhaitent activer ou désactiver.

#### 4. Consentement

Une checkbox « Agree to terms » est présente dans le formulaire d'inscription, obligeant les utilisateurs à donner leur consentement explicite pour le traitement de leurs données.

#### 5. Droit d'Information

Les liens vers les **Conditions Générales d'Utilisation** (CGU) sont clairement visibles dans le pied de page du site.

Ces documents détaillent comment les données sont utilisées et stockées.

#### 6. Temporalité des Données

Les données sont conservées pendant une période définie en fonction de leur utilité.

Par exemple, les données de réservation peuvent être conservées pendant une période de deux ans pour des raisons comptables.

#### 7. Droit à l'Oubli

Les utilisateurs ont la possibilité de supprimer leur compte. Dans ce cas, toutes les données associées à ce compte sont anonymisées ou supprimées, à l'exception des transactions qui sont conservées pour des raisons légales.

#### 8. Droit de Rectification

Les utilisateurs peuvent accéder à une page « Éditer le profil » où ils peuvent mettre à jour ou corriger leurs informations personnelles.

#### 9. Sécurité des Données

L'application utilise des algorithmes de hachage forts comme bcrypt pour sécuriser les mots de passe.

De plus, des mesures sont en place pour prévenir les vulnérabilités telles que les injections SQL et les attaques XSS.

**Liens vers les sites qui ont étés analysés :**

* [**https://www.rgpd-experts.com/cgv-cgu/**](https://www.rgpd-experts.com/cgv-cgu/)

### Mise en Œuvre sur Stellar

#### Anonymisation des Données:

Lorsqu'il y aura la possibilité de laisser des commentaire et qu’un utilisateur supprimera son compte, les commentaires qu'il aura laissé ne seront pas supprimés mais deviendront anonymes.

#### Suppression des Données:

Lorsqu'un utilisateur supprime son compte, les annonces qu'il a laissés sont supprimées uniquement s’il n’y a pas de réservations en cours.

#### Minimisation et Finalité:

Lors de l'inscription, seules des données minimales comme l'adresse e-mail sont collectées. Pour une réservation, des données supplémentaires seront nécessaires.

#### Droit à l'Oubli et de Rectification:

Les utilisateurs ont la possibilité de supprimer ou de modifier leur compte et leurs données.

#### Consentement et Informations Légales:

Une checkbox est présente dans le formulaire d'inscription, et des liens vers les documents légaux sont facilement accessibles.

### Conclusion

La mise en conformité avec le RGPD est une étape cruciale dans le développement de toute application en ligne.

Elle assure la protection des utilisateurs et la conformité légale de l'application.

# Fonctionnalités et Interaction

## Authentification et Gestion de Compte

L'authentification est un élément important de quasiment toutes les applications web modernes. Sur Stellar, j'utilise **Symfony** et ses composants de sécurité pour gérer cette partie.

Le système doit être solide pour protéger les données des utilisateurs et assurer que seules les personnes autorisées aient accès aux comptes.

### Login : Vue et Contrôleur

* **La vue Twig** pour la page de connexion (login.html.twig) étend un layout de base et implémente un formulaire simple avec les champs email et mot de passe.

Le **système de sécurité** **Symfony** est déjà assez bien configuré pour gérer les erreurs et les sessions. Le formulaire envoie les données via POST et si une erreur se produit, elle s'affiche directement sur la page.

* **Dans le contrôleur (SecurityController),** Symfony injecte une dépendance AuthenticationUtils qui contient des méthodes pour obtenir la dernière erreur d'authentification et le dernier nom d'utilisateur entré.

### Signup : Vue et Formulaire

* **La vue Twig** pour la page d'inscription (signup.html.twig) est similaire à celle de la connexion.

La principale différence est qu'elle utilise form\_row pour rendre les champs de formulaire, qui sont définis dans le fichier RegistrationFormType.

* **Le formulaire d'inscription** est créé comme une classe dans Symfony, où je définis les champs et leurs validations.

Par exemple, le champ plainPassword utilise RepeatedType::class pour s'assurer que l'utilisateur entre le même mot de passe deux fois.

J'ai également ajouté un champ **agreeTerms** qui utilise la contrainte **IsTrue** pour s'assurer que l'utilisateur accepte les termes et conditions.

### Hashage du mot de passe

Je n'ai pas inclus le code moi-même pour le **hashage** du **mot** **de** **passe** ici, mais il est essentiel de le mentionner.

Le **hashage** est une manière de sécuriser les mots de passe en les transformant en une série de caractères qui ne peuvent pas être inversés pour obtenir le mot de passe original.

Symfony fournit des encodeurs de mots de passe qui peuvent utiliser différentes méthodes de hashage, comme **bcrypt**, **argon2i**, etc.

#### Hashage en PHP natif

En PHP natif, on utilise principalement la fonction password\_hash() pour le hashage. Cette fonction prend le mot de passe et un algorithme de hashage comme arguments.

Voici un exemple basique :



Ici, PASSWORD\_DEFAULT est une constante qui utilise l'algorithme de hashage par défaut, qui est bcrypt à l'heure actuelle mais peut être modifié dans les futures versions de PHP.

* **Bcrypt (Utilisé par default sur symfony) :**

**Très répandu et soutenu** par de nombreuses plates-formes.

**Résistant aux attaques** par force brute grâce au coût (un facteur qui ralentit l'opération de hashage).

**Le** **choix** **de** **l'algorithme** peut dépendre de plusieurs choses comme la compatibilité, le temps de traitement et les exigences de sécurité.

Pour des projets sensibles, **Argon2** est généralement la meilleure option grâce à sa résistance contre divers types d'attaques.

**Bcrypt** est également une bonne option si **Argon2** n'est pas disponible. Les autres méthodes comme **SHA-256** et **MD5** sont généralement déconseillées pour le hashage des mots de passe.

## Système de Réservation

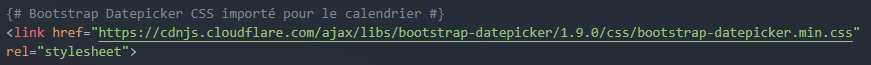
**Le système de réservation** est une de **fonctionnalité les plus importante** de l'application Stellar, conçu pour offrir une expérience utilisateur fluide en s'appuyant sur une architecture **back-end solde**.

Le tout est bâti sur **Symfony, jQuery et Bootstrap Datepicker**, qui assure à la fois **flexibilité et sécurité**.

### Interface Utilisateur : Vue de Détail et Calendrier (detail.html.twig, annexe : Calendrier)

#### Calendrier de Réservation

La vue de détail inclue un calendrier Bootstrap Datepicker pour permettre une sélection intuitive des dates.

Les dates non disponibles sont visuellement désactivées pour orienter l'utilisateur.

#### Logique Côté Client : jQuery et JavaScript (annexe : Calendrier)

Le comportement dynamique du calendrier est géré par jQuery et JavaScript.

Ces scripts permettent de griser les dates déjà réservées, de calculer dynamiquement le coût de la réservation, de récupérer les dates choisies, etc.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

#### Contrôleur : Logique Serveur (AdvertController.php, annexe : Réservation)

Le contrôleur AdvertController est le cerveau du côté serveur. Il gère la récupération des données des annonces, les dates déjà prises, le prix de l’annonce, ...

#### Formulaire de Réservation (ReservationType.php, annexe : Réservation)

Le formulaire, crucial pour l'interaction utilisateur, est défini dans ReservationType. Il comporte des champs cachés pour les dates qui sont obligatoires.

Champs cachés pour les dates : Ces champs de type HiddenType stockent les dates d'arrivée et de départ sélectionnées dans le calendrier.

#### Modèle de Données : Entité de Réservation (Reservation.php, annexe : Réservation)

L'entité Reservation est le modèle qui stocke les données de chaque réservation.

Elle contient des champs obligatoires pour les dates d'arrivée et de départ.

#### Calcul des Prix et Sécurité

La logique de calcul du prix est implantée côté serveur pour assurer l'intégrité des données.

De plus, l'accès à la fonction de réservation est limité aux utilisateurs authentifiés.

Il est impossible d’avoir une date de départ antérieur à la date d’arrivée, ce qui évite des prix négatifs et des situations impossibles.

## Map (Annexe. Map)

### Introduction

Dans ce projet, j’ai utilisé la Google Maps JavaScript API en combinaison avec le service Geocoder pour créer une carte interactive qui affiche différentes annonces.

### Google Maps JavaScript API

#### Initialisation de la Carte :

Après avoir chargé la bibliothèque Google Maps via le script, la fonction initMap() est appelée pour initialiser la carte.

Cette fonction crée une nouvelle instance de l'objet Map en spécifiant les options, telles que le centre de la carte et le niveau de zoom.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

#### Bibliothèques Importées :

La bibliothèque AdvancedMarkerView est importée pour personnaliser davantage les marqueurs sur la carte.

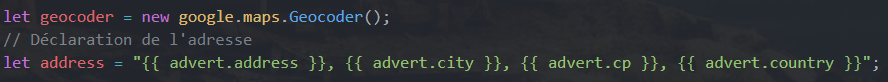


### Geocoder

#### Géocodage des Adresses :

Geocoder est utilisé pour convertir les adresses textuelles en coordonnées géographiques.

C’est particulièrement utile pour les annonces, car elles sont uniquement entrées en format texte.





### Marqueurs et Étiquettes

#### Ajout de Marqueurs :

Les coordonnées géographiques obtenues par Geocoder sont utilisées pour placer des marqueurs sur la carte.

Chaque marqueur représente une annonce individuelle.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

#### Étiquettes de Prix :

Les étiquettes sont ajoutées aux marqueurs pour afficher le prix des annonces.

Les étiquettes sont personnalisées grâce à la classe CSS custom-price-label.

Une image contenant texte, Police, capture d’écran, conception

Description générée automatiquement

### InfoWindow et Interaction Utilisateur

#### InfoWindows au Survol (Annexe. Maps, InfoWindows) :

Les InfoWindows sont utilisées pour afficher des informations détaillées sur chaque annonce lorsque l'utilisateur passe la souris sur un marqueur.

Les InfoWindows sont créées dynamiquement à partir du titre et de l'image de l'annonce.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

#### Marqueurs Cliquables :

Un écouteur d'événements est ajouté à chaque marqueur pour rendre le marqueur cliquable.

En cliquant sur le marqueur, l'utilisateur est redirigé vers la page de détails de l'annonce.

Une image contenant texte, Police, écriture manuscrite, tableau

Description générée automatiquement

**Conclusion**

L'utilisation de la Google Maps JavaScript API en combinaison avec le service Geocoder permet une intégration transparente et efficace de fonctionnalités de cartographie.

Les marqueurs personnalisés et les InfoWindows ajoutent une dimension supplémentaire à l'expérience utilisateur, rendant la navigation interactive sur le site.

## Ajout d’annonce (Annexe : Add Advert)

Dans la page d'ajout d'annonces, le code Twig et le JavaScript sont utilisés ensembles pour une expérience utilisateur optimale.

Voyons cela en détail :

### Formulaire d'Ajout d'Annonce avec Twig

#### Structure du Formulaire :

Le formulaire d'ajout d'annonce est créé en utilisant le framework Symfony côté serveur.

Les différents champs sont injectés dans le HTML via la syntaxe Twig.

Une image contenant texte, Police, tableau, écriture manuscrite

Description générée automatiquement

#### Catégories et Types d'Annonce :

Le formulaire utilise des boucles pour générer des cases à cocher pour les catégories et les types d'annonces.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

### Interaction Utilisateur avec JavaScript

#### Glisser-Déposer des Images :

L'interface utilisateur permet de télécharger des images soit par un glisser-déposer, soit en cliquant sur un bouton.

Le JavaScript est utilisé pour gérer ces interactions.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

#### Gestion des Scrolling dans les Cases à Cocher :

Le JavaScript est également utilisé pour permettre le défilement horizontal et vertical dans les sections de cases à cocher.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

**Conclusion sur l'Ajout d'Annonces**

La page d'ajout d'annonces est un excellent exemple d'intégration entre le back-end et le front-end.

Twig facilite la création de formulaires dynamiques et personnalisés, tandis que le JavaScript améliore l'expérience utilisateur avec des fonctionnalités comme le glisser-déposer et le défilement personnalisé.

## Recherche personnalisée par filtres

La barre de filtres est utile pour chercher des annonces spécifiques.

Sur celle de Stellar nous pouvons y trouver des filtres de :

* Lieux tels que le pays où la ville.
* Une tranche de prix avec un prix minimum et maximum pour viser directement les annonces sans dépasser son budget.
* Une période, avec une date d’arrivée et de départ pour trouver les annonces uniquement disponibles sur les dates choisies.

### La technique

Cette fonctionnalité contient absolument tout ce qu’on ait pu apprendre durant le cours de l’année, voyons-en un peu plus :

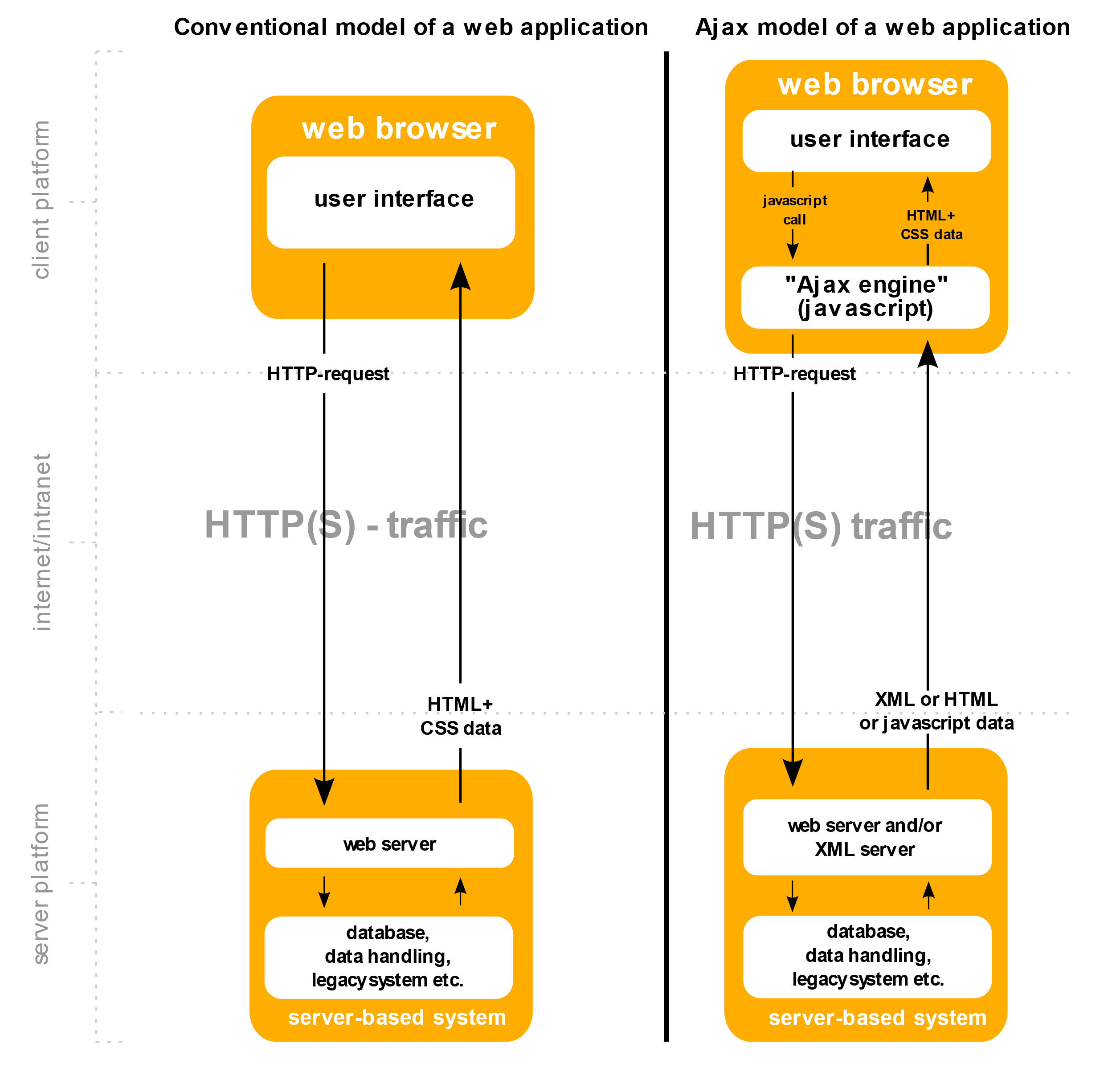
#### Front-end :

* **HTML (Exemple filtre de prix) :**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

* **Javascript** (Utilisation de la méthode fetch pour une requête en AJAX, avec shema) :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

L’AJAX (Asynchronous JavaScript and XML) est très utile pour éviter un passage sur une autre page lors de changement de données qui nécessiterait un chargement de celle-ci et modifie partiellement les données d’une page pour une expérience utilisateur plus dynamique.

#### Back-end :

* PHP et DQL :

Une image contenant texte, Appareils électroniques, capture d’écran, ordinateur

Description générée automatiquement

# Fonctionnalité Phare

## Bootstrap Datepicker

### Introduction

L'une des fonctionnalités les plus importante de ce projet est le système de réservation basé sur un calendrier.

Le calendrier est implémenté en utilisant Bootstrap Datepicker, une bibliothèque de choix de date pour Bootstrap.

Cette fonctionnalité permet aux utilisateurs de sélectionner des dates d'arrivée et de départ pour leur réservation, tout en bloquant les dates déjà réservées.

### Backend :

#### PHP et Symfony

Dans le contrôleur AdvertController, la méthode showDetailAdvert() s'occupe de récupérer les données de l'annonce et de gérer la logique de réservation.

Elle utilise plusieurs repositories pour récupérer les données nécessaires et les passer au template.

Un objet de réservation est créé et associé à l'annonce et à l'utilisateur en cours.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

### Frontend :

#### Twig et JavaScript

Le fichier detail.html.twig contient la mise en page HTML pour cette fonctionnalité.

Les deux champs de date d'arrivée et de départ sont désactivés par défaut et seront remplis automatiquement lorsque l'utilisateur sélectionne des dates sur le calendrier.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

#### Bootstrap Datepicker

L'élément de calendrier est initialisé avec plusieurs options.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

#### Gestion des Dates Réservées

Le code JavaScript utilise la méthode beforeShowDay pour parcourir les dates déjà réservées et les rendre non cliquables.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

#### Validation et Alertes

Le système vérifie également que la date de départ est ultérieure à la date d'arrivée. Si ce n'est pas le cas, un popup d'alerte est affiché.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

#### Calcul des prix

Les prix sont automatiquement calculés par rapport aux dates choisies par l’utilisateur.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

#### Résérvation « multiple-step »

La réservation ne peut se faire uniquement avec un bouton cliquable quand l’utilisateur a choisi des dates conformes.

Ensuite la réservation se fait sur plusieurs étapes en utilisant du jQuery pour avancer ou reculer dans les étapes :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement Une image contenant texte, capture d’écran, Police

Description générée automatiquement

# SEO et Optimisations

## Introduction au SEO

Le Search Engine Optimization (SEO), ou optimisation pour les moteurs de recherche, est une techniqie déstinée à améliorer la visibilité d'un site web dans les résultats des moteurs de recherche.

Bien plus qu'une simple tactique de marketing, le SEO est devenu un élément essentiel pour toute entreprise ou individu cherchant à établir une présence en ligne visible.

## Techniques de SEO sur Stellar et Leur Importance :

### Mots-clés et Métadonnées :

La recherche et l'incorporation de **mots-clés pertinents** sont **essentielles** pour le SEO.

Les mots-clés doivent être placés dans les **titres**, les **sous-titres**, les **méta-descriptions** et même les **URL**. Les métadonnées, bien qu'invisibles pour l'utilisateur moyen, fournissent des informations cruciales aux moteurs de recherche sur le contenu de la page.

La **méta-description** par exemple est un **court texte descriptif** qui apparaît sous le lien dans **les résultats de recherche**.

Elle doit être **concise** et **attrayante** pour **inciter les utilisateurs à cliquer** sur le lien.

Il faut s’assurer d'inclure des **mots-clés pertinents** dans la méta-description, car cela peut **améliorer le classement SEO**.

Cependant, il faut éviter le bourrage de mots-clés, le texte doit rester naturel et informatif.

**Exemple :**

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

### Alt Text :

J'utilise **l'attribut alt (texte alternatif)** pour **décrire chaque image** de mon site **aux moteurs de recherche**.

Chaque balise alt est **descriptive** et **décrit clairement le contenu de l'image**, en utilisant des **mots-clés pertinents** si possible.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

Nous avous par exemple ici un **text alternatif « image de l’annonce : titre de l’annonce »,**  qui **précise** bien ca à quoi l’image correspond.

### Optimisation du Temps de Chargement:

Un **site web lent** peut gravement **affecter le taux de rebond** et par conséquent, le classement SEO.

La **réduction de la taille des images**, la **minification des fichiers CSS** et **JavaScript** et l'utilisation de la **mise en cache** sont quelques techniques qui peuvent **accélérer le temps de chargement**.

Nous avons par exemple la possibilité d’utiliser de **l’AJAX (Asynchronous JavaScript and XML),** pour **modifier** des parties de page **sans forcément avoir besoin d’avoir un rafraichissement** de la page, comme je l’utilise ici pour ma **barre de filtres :**

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

### Structure HTML et Balisage Sémantique:

Utiliser des **balises HTML sémantiques** comme h1, h2, strong, etc., **aide les moteurs de recherche** à comprendre le **contexte** et la **structure** du contenu. Cela peut **améliorer l'indexation** et le classement du site.

**Exemple d’une section sur la page d’accueil :**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

### Mobile-Friendly et Responsive Design:

Avec l'augmentation de l'utilisation des appareils mobiles pour accéder au web, avoir un **design réactif** et **mobile-friendly** est devenu essentiel.

**Google pénalise les sites** qui ne sont pas **optimisés pour les mobiles**, ce qui en fait une considération de SEO incontournable.

Nous faisons ça en utilisant des **« media queries »** en **CSS** pour **adapter** les pages web en fonction de la **taille d’écran** :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

### Contenu de Qualité :

En fin de compte, **le contenu est roi**.

Un contenu de **haute qualité**, **informatif** et **engageant** attire non seulement **plus de visiteurs** mais est aussi plus susceptible d'obtenir des **backlinks** (liens d’autres site vers Stellar) de **qualité**, ce qui est un autre facteur de classement important.

### Liens interne (et slugs) et Liens Externes :

**- Liens internes :**

Je crée **des liens internes** vers d'autres pages de mon site **pour aider les moteurs de recherche** à comprendre sa **structure** et à indexer le contenu de manière efficace.

Les ancres des liens contiennent des **mots-clés** pertinents pour la **page cible**.

**- Liens externes :**

En ce qui concerne **les liens externes**, je m'assure qu'ils pointent vers des **sources fiables et de qualité** pour renforcer la crédibilité de mon contenu.

Le seul exemple que j’ai dans Stellar est mon **« Easter egg »** sur la page d’accueil où nous pouvons cliquer sur le titre « Stellar » pour **viser une page** de définition du mot **« Ataraxie »** :

**Slug :**

Le **slug** est un **élément essentiel** de l'URL d'une page web. Il s'agit de la partie de **l'adresse web** qui identifie spécifiquement la page en question.

Le **rôle principal du slug** est de rendre l'URL plus **conviviale** pour **les utilisateurs et les moteurs de recherche**.

Un **bon slug** doit être **court**, **descriptif** et contenir des **mots-clés** **pertinents** liés au contenu de la page. Il **permet** aux visiteurs **de comprendre** de quoi parle la page rien qu'en **regardant l'URL**.

**J'évite les caractères spéciaux** et les **espaces dans les slugs**, utilisant plutôt des **tirets pour séparer les mots**.

**- Vue :**



**- Controller :**



**- Navigateur :**



**Par exemple**, dans mon URLmis en capture d’écran au-dessus, **"http://127.0.0.1:8000/detail-annonce/le-dome-des-cimes/52"**, **"le-dome-des-cimes"** est le slug.

**Traduction d’une documentation Symfony :**

**- Anglais :**

« Generating a slug that reads well in a URL (where anything besides ASCII characters should be encoded) is a challenging task, especially for languages other than English. How do you convert é to e for instance?

Instead of reinventing the wheel, let's use the Symfony String component, which eases the manipulation of strings and provides a slugger.

…

The computeSlug() method only computes a slug when the current slug is empty or set to the special - value. Why do we need the - special value?

Because when adding a conference in the backend, the slug is required. So, we need a non-empty value that tells the application that we want the slug to be automatically generated. »

**- Français :**

« Générer un slug qui se lit bien dans une URL (où tout ce qui n'est pas des caractères ASCII doit être encodé) est une tâche difficile, en particulier pour les langues autres que l'anglais. Comment convertir par exemple é en e ?

Au lieu de réinventer la roue, utilisons le composant String de Symfony, qui facilite la manipulation des chaînes de caractères et fournit un générateur de slug.

…

La méthode computeSlug() ne calcule un slug que lorsque le slug actuel est vide ou défini sur la valeur spéciale -. Pourquoi avons-nous besoin de la valeur spéciale - ?

Parce que lors de l'ajout d'une conférence dans le backend, le slug est requis. Ainsi, nous avons besoin d'une valeur non vide qui indique à l'application que nous voulons que le slug soit généré automatiquement. »

**Liens vers la page de documentation :** <https://symfony.com/doc/current/the-fast-track/en/13-lifecycle.html#using-slugs-in-the-application>

**Conclusion sur le SEO et Optimisations**

Le SEO est une façon de faire à plusieurs faces qui va au-delà de l'insertion de mots-clés ou de l'obtention de backlinks.

C'est une combinaison de techniques de qualité, de performances du site et de fourniture d'un contenu qui répond aux besoins et aux questions des utilisateurs.

Dans un monde numérique en constante évolution, le SEO n'est pas un "one-off", mais un processus continu qui nécessite une attention et une adaptation constantes.

Ignorer le SEO, c'est risquer de se perdre dans l'océan infini du web, tandis qu'un bon SEO peut être votre bouée de sauvetage vers la visibilité et le succès en ligne.

# Sécurité du Site Web

## Introduction à la Sécurité Web

La sécurité du site web est souvent l'un des aspects les plus négligés dans le développement web, mais elle est vitale pour protéger à la fois les données de l'entreprise et celles des utilisateurs.

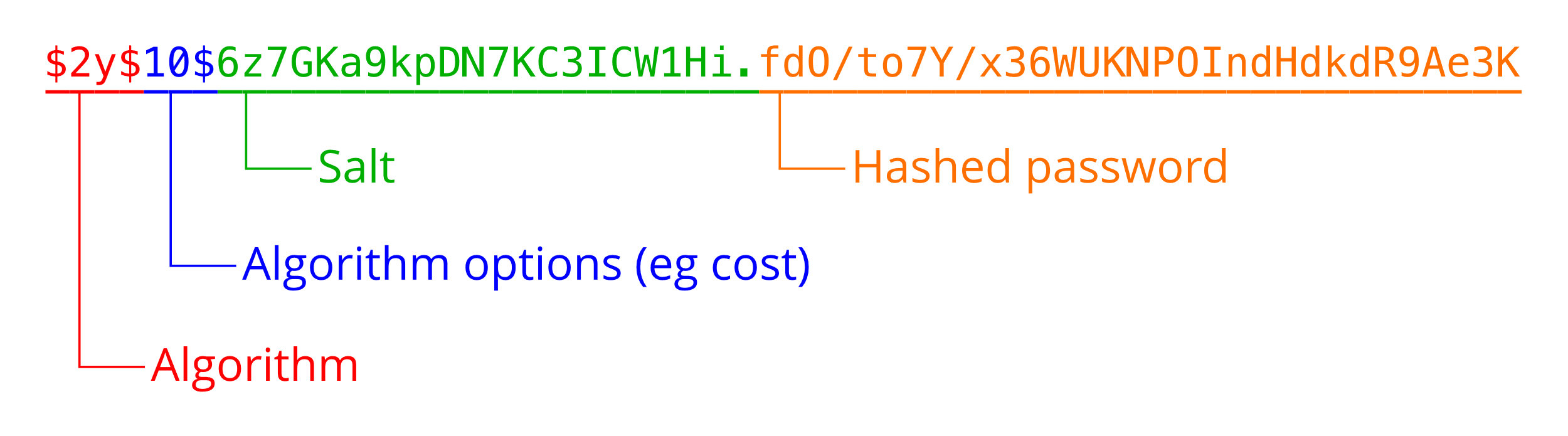
Une seule faille de sécurité peut entraîner des conséquences désastreuses, allant de la perte de données à des dommages financiers et à une mauvaise réputation.

## Sécurité utilisée dans Stellar

### Hashage de mot de passe

Pour les **données personnelles** nous aurons besoin d’éviter toute type d’**intrusion** par un potentiel **malfaiteur**, pour cela nous nous devons de faire en sorte d’avoir **aucune possibilité d’accès** à un compte qui n’est pas celui de l’utilisateur lui-même, ce qui passe par une sécurité à l’entrée de la connexion.

Symfony gère automatiquement ceci avec un **hashage** de mot de passe avec la méthode de hashage **Bcrypt (une des meilleures fonctions de hashage)** qui est une méthode lente à exécuter et qui rend donc l’attaque par force brut plus pénible ainsi qu’un salt automatique :



Le **Salt** fait en sorte que chaque mot de passes, deviennent finalement **uniques**, **indéchiffrables** et **irréversibles** si quelqu’un accède par quelconque techniques à la base de données.

Il agit comme une **"emprunte numérique"**, ce qui signifie que même si deux utilisateurs ont le **même mot de passe**, leur **version hashée** sera **différente** en raison de l'utilisation du salt.

Cela **renforce encore la sécurité**, rendant les **attaques par dictionnaire** ou par **tables arc-en-ciel** extrêmement difficiles à réussir.

### Le Token CSRF (Cross-Site Request Forgery):

La protection contre les **attaques CSRF** est cruciale. Dans Stellar, nous utilisons un jeton CSRF pour **protéger nos formulaires**.

Ce jeton est stocké **en session et hashé**.

Lorsqu'un **formulaire est soumis**, le jeton CSRF est validé pour s'assurer qu'il correspond au **jeton stocké sur le serveur**.

Si ce n'est pas le cas, la **requête** est **refusée**.

**Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement- Le code où il est appliqué :**

**- Son apparition dans le formulaire de connexion :**

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

**- Génération du Token CSRF :**

Quand le formulaire de **connexion est rendu**, **un "token CSRF"** est **généré** et **inséré** dans le formulaire comme un **champ caché**.

**- Validation du Token CSRF :**

Lorsque le **formulaire est soumis**, le **CsrfTokenBadge** reçoit le **token CSRF soumis** et le **compare** au **token CSRF stocké** sur le serveur (typiquement dans la **session** **de l'utilisateur**).

**- Sécurité renforcée :**

Si le **token CSRF** soumis ne **correspond pas** à celui du serveur, cela indique que la requête pourrait être **une attaque CSRF** et la **connexion** est alors **refusée**.

### Expressions Régulières (REGEX) et Attaque par Dictionnaire

Pour de **sécuriser** les systèmes contre les **intrusions malveillantes**, notamment les **injections de code via les inputs des formulaires,** l'emploi des **expressions régulières (REGEX)** est un outil vraiment nécessaire.

Prenons l'exemple de la **configuration des mots de passe dans Stellar**, où un **REGEX** spécifique est utilisé pour imposer une complexité minimale :

Une image contenant texte, capture d’écran

Description générée automatiquement

Cette stratégie est non seulement **un barrage contre les attaques par injection**, mais elle protège également contre **les attaques par dictionnaire**, où les **attaquants** exploitent des listes de **mots de passe courants** pour accéder aux comptes.

En exigeant un minimum de **12 caractères**, dont **une** **lettre majuscule**, **un caractère spécial** et **un chiffre**, le **REGEX** rend les **mots de passe** **moins susceptibles d'être devinés ou craqués par des méthodes automatisées**.

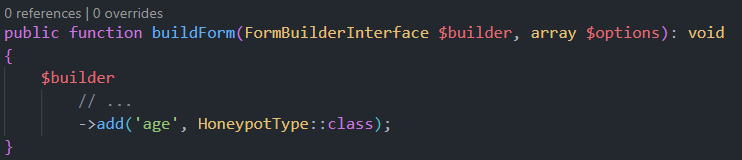
### Défense contre les attaques par force brute

Les **attaques par force brute** sont une **menace commune et persistante** pour la sécurité en ligne, où les attaquants tentent de **deviner les mots de passe en testant de nombreuses combinaisons**. Pour y faire face, Stellar intègre des **mesures préventives** telles que les honeypots dans ses formulaires.

#### Intégration d'un Honeypot dans les formulaires :

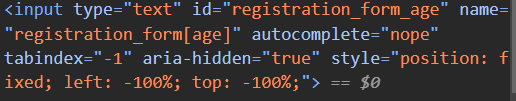
Nous ajoutons des **champs honeypots aux formulaires**, qui sont **invisibles aux utilisateurs** mais **détectables par les scripts automatisés**.

Voici comment nous implémentons un **honeypot** :



#### Apparition du Honeypot dans le code HTML :

**Le champ honeypot** est rendu **invisible par des styles CSS**, **évitant** ainsi d'être **rempli** par les **utilisateurs humains**, mais **restant une cible pour les bots** :



#### Validation et Sécurité renforcée :

Lorsque le champ **honeypot** est **rempli**, cela **signale** une **tentative d'accès automatisée**. Cela nous permet **d'identifier** et de **bloquer** les **requêtes suspectes**, **renforçant** la **sécurité** contre les **tentatives de connexion automatisées et les attaques par force brute**.

### Prévention de la faille XSS (Cross-Site Scripting) :

La **faille XSS** est une **vulnérabilité courante** qui permet aux attaquants **d'injecter des scripts côté client** dans les pages web vues par d'autres utilisateurs.

Pour la contrer, il est essentiel d'assurer **l'échappement adéquat des données** saisies par les utilisateurs **avant leur affichage**.

#### Échappement des données en Twig :

**Twig**, le moteur de template par défaut de Symfony, **échappe automatiquement toutes les variables** pour la sécurité.

Lorsqu'on affiche une variable, **Twig traite les données** pour **éviter l'exécution de code HTML ou JavaScript** malveillant.

Voici comment une variable email est traitée dans Twig :

Image téléchargée

Cette instruction utilise **form\_row** pour rendre le champ de l'email et **Twig** **s'occupe de l'échappement** des valeurs de l'email pour prévenir l'injection de scripts.

#### Utilisation d'EmailType dans les formulaires Symfony :

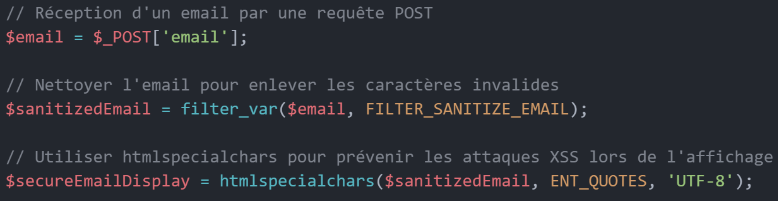
Dans **les formulaires Symfony**, lorsqu'on **définit un champ** avec **EmailType**, Symfony s'assure que la **valeur saisie** est une **adresse email valide** :

Une image contenant texte, Police, capture d’écran, écriture manuscrite

Description générée automatiquement

#### Explication en PHP natif :

En **PHP** pur, le **nettoyage des données** reçues est une étape **cruciale** pour **éviter les failles XSS**. La fonction **filter\_var** avec **FILTER\_SANITIZE\_EMAIL** est utilisée pour **nettoyer l'email**, ce qui **enlève les caractères non valides** pour une adresse email :



Après ce nettoyage, **htmlspecialchars transforme tout caractère spécial en entités HTML,** **empêchant** toute **tentative d'injection de code malveillant**. Cette combinaison de **nettoyage et d'échappement** est essentielle pour **renforcer la sécurité** contre les **attaques XSS**.

### Protection contre les injections SQL :

**Une injection SQL** est un type de faille de sécurité où des **requêtes non autorisées sont introduites** dans une **base de données**.

Cela peut arriver si les entrées des utilisateurs ne sont **pas correctement vérifiées** avant d'être intégrées dans une requête SQL.

C'est un **risque majeur** car cela peut **permettre** à un **attaquant de lire**, **modifier**, ou **supprimer** des **données sensibles**.

#### L'efficacité des requêtes préparées :

Les **requêtes préparées** sont une **technique standard** pour **bloquer** ces **injections**.

Leur fonctionnement est simple : elles **séparent** les **données des utilisateurs du code SQL de la requête**.

Cela signifie que, **peu importe** ce que **l'utilisateur entre**, cela ne sera **pas interprété comme une commande par la base de données**.

En utilisant **Doctrine** avec **Symfony**, on construit des **requêtes SQL sûres automatiquement** grâce au **QueryBuilder**.

Par exemple, pour filtrer les annonces par ville, on écrirait :

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

#### Pourquoi utiliser des requêtes préparées ?

Les **requêtes préparées** sont **essentielles** pour plusieurs raisons.

Elles **empêchent** les **utilisateurs** de **briser la structure de la requête SQL** avec des **entrées** **malveillante**, elles sont **plus faciles à lire** et **à maintenir**, et elles peuvent aussi **améliorer les performances** des **requêtes répétées**.

En somme, les requêtes préparées sont une **partie critique** de la **sécurisation des applications web**.

En les utilisant systématiquement, on **s'assure** que seules les **instructions prévues** par les développeurs sont exécutées, ce qui **maintient l'intégrité** de la **base de données** et **la confiance des utilisateurs**.

### Les Conditions

Avec **twig** nous pouvons, avec des **conditions**, **sécuriser** certaines parties de **notre site**. Pour avoir un **exemple concret**, dans la page de détail de Stellar non avons aussi les étapes de réservation, sachant qu’un **utilisateur non connecté n’a pas les droits pour effectuer une réservation**, il a été **obligatoire** de mettre une **condition** pour **vérifier** si l’utilisateur est **connecté ou non** :

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

**« isLoggedIn »** a été récupérée dans la fonction **showDetailAdvert()** dans **AdvertController** qui là, est une variable contenant une **condition** **condensée** **gérée** en **PHP** en utilisant **l’opérateur** **ternaire** :

Une image contenant texte, Police, écriture manuscrite, capture d’écran

Description générée automatiquement

### Les autorisations :

Mais nous devons aussi **éviter** qu’un user **non connecté** ne puisse en **aucun cas** **se trouver sur une page dont il n’a pas le droit ou encore pire, une page qui ne lui appartient pas**, c’est pour cela que nous devons faire en sorte de restreindre les fonctions faites dans le contrôleur.

Ici Symfony nous aide beaucoup avec un contrôle d’accès fais en plusieurs étapes :

**- Définition des rôles dans le fichier security.yaml**

Une image contenant texte, Police, capture d’écran, typographie

Description générée automatiquement

**- Ajout du path spécifique pour chacune**

****

**- Utiliser isGranted pour restreindre les rôles autorisés**



**Conclusion sur la Sécurité du Site Web**

La **sécurité** du site web est un **impératif non négociable** dans le développement web. **Ignorer** cet aspect peut entraîner **des conséquences graves**, notamment la **perte de données et de réputation**.

Par conséquent, il est **crucial d'intégrer des pratiques** de **sécurité robustes** tout au long du cycle de développement.

# Axes d'Amélioration et Futur du Projet

## Introduction

Alors que la plateforme de réservation en ligne fonctionne bien en l'état actuel, il y a toujours de la place pour l'amélioration et l'innovation. Dans cette section, nous allons discuter des axes potentiels d'amélioration et de ce que l'avenir pourrait réserver au projet.

## Axes d'Amélioration

#### Optimisation des Performances:

Le code pourrait être optimisé pour réduire les temps de chargement des pages et améliorer l'expérience utilisateur.

#### Fonctionnalités de Recherche Avancée:

Des filtres supplémentaires basés sur les préférences de l'utilisateur pourraient être ajoutés.

#### Système de Notation et Commentaires:

Un système de notation pourrait aider les utilisateurs à choisir plus facilement un hébergement qui leur convient.

#### Messagerie entre contacts

Un système de messagerie interne entre utilisateurs pourrait être mis en place pour potentiellement demander des informations supplémentaires, pour éviter de passer par l’adresse e-mail.

#### Paiement en Ligne:

L'ajout de plusieurs options de paiement et la mise en œuvre d'un système de paiement sécurisé pourraient attirer un plus grand nombre d'utilisateurs.

#### Internationalisation:

Le site pourrait être traduit en plusieurs langues pour attirer un public plus large.

## Futur du Projet

#### Intégration de l'Intelligence Artificielle:

L'utilisation de l'IA pour des recommandations personnalisées pourrait être une étape logique.

#### Expérience Utilisateur Augmentée:

L'intégration de technologies comme la réalité virtuelle pour les visites d'hébergement pourrait être envisagée.

#### Partenariats et Collaborations:

Établir des partenariats avec des entreprises liées au voyage pourrait offrir des opportunités de croissance.

#### Évolution vers une Application Mobile:

Développer une application mobile dédiée pourrait être une étape logique pour rendre le service plus accessible.

#### Éco-responsabilité:

Le site pourrait mettre en avant des hébergements écologiques et encourager des pratiques durables.

**Conclusion**

En tenant compte de ces axes d'amélioration et en envisageant les différentes directions que pourrait prendre le futur du projet, on peut s'attendre à une plateforme de réservation en ligne encore plus robuste, flexible et orientée utilisateur.

# Annexes

## Fonctionnalités

### Calendrier et réservation

#### Calendrier

* **Visuel :**

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Une image contenant texte, conception, capture d’écran

Description générée automatiquementUne image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

**- Snippets**

Html et Twig :

**Une image contenant texte, capture d’écran, Police

Description générée automatiquement**

Jquery et Javascript :

Une image contenant texte, capture d’écran

Description générée automatiquement

Une image contenant texte, capture d’écran, nuit

Description générée automatiquement

#### Réservation

Une image contenant texte, capture d’écran

Description générée automatiquement

### Map

Une image contenant carte, atlas, texte

Description générée automatiquement

## Wireframes et Prototypes

Une image contenant texte, capture d’écran, Rectangle, carré

Description générée automatiquement

Une image contenant capture d’écran, texte, conception

Description générée automatiquement

Une image contenant capture d’écran, texte, distributeur automatique

Description générée automatiquement

## Schémas de Base de Données (MCD, MLD)

Une image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquement

Une image contenant texte, diagramme, capture d’écran, Plan

Description générée automatiquement

## Suivi de Projet avec Trello

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement