

Cahier des charges

Site applicatif

Local Link Immo

Site d'autogestion de biens immobiliers

Date du 18 mars 2024

Version 1.3

Session septembre2023

SOMMAIRE

1. Présentation de mon Site applicatif

- a) Pourquoi ce site ?
- b) Objet du site
- c) Public cible
- d) Notre Equipe dédiée
- e) Budget
- f) Organisation
- g) Les tâches à accomplir
- h) Roadmap
- i) Les outils collaboratifs choisis

2. Les fonctionnalités clés de mon projet

- a) Les besoins fonctionnels de mon site
- b) A l'avenir quelles améliorations

3. Conception et structure

- a) Mise en place des bonnes pratiques en éco-conception
- b) A l'avenir, quelles améliorations ?
- c) Structure du site

3. Les contraintes techniques

4. Technologies utilisées

- a) Front-end
- b) Back-end
- c) Architecture logicielle
- d) Veilles techniques

Lexique

1.Présentation de mon site applicatif

a) Pourquoi ce site ?

L'investissement immobilier est en constante progression, que ce soit :

- de petits propriétaires avec de petits parcs immobiliers
- ou de gros promoteurs.

Le rêve est le même : devenir prospère à moindre coût !

En effet, déléguer sa gestion locative implique des frais qui viennent réduire la rentabilité de l'investissement. Les agences immobilières avec mandat de gestion, facturent généralement entre 6% et 10% du loyer hors charges et davantage si vous disposez de garanties.

De plus, certaines agences peuvent manquer de réactivité en cas de problème, ce qui peut nuire à la relation entretenue avec les locataires. C'est pourquoi de nombreux propriétaires aujourd'hui choisissent la gestion locative en autonomie.

En conclusion, la gestion locative est un aspect essentiel pour assurer la rentabilité et la durabilité de l'investissement immobilier. Le choix entre une agence immobilière, un site internet pour une gestion en solo dépendra des besoins, du budget mais aussi et surtout de la disponibilité du propriétaire.



Je suis abonnée à [] depuis plusieurs années et j'aurais du mal à m'en passer. C'est un outil performant, simple d'utilisation, complet. Dès que j'ai besoin d'un conseil ou d'une intervention sur mon compte, j'ai un interlocuteur-trice réactif et compétent en face de moi (par chat). En somme, tout est fluide et répond à mes attentes.

V []



Bonjour , je suis utilisatrice de cette plateforme depuis 2018 et je peux dire qu'elle est très performante, facile d'utilisation, les personnes sont très à l'écoute des demandes, apportent une réponse assez rapidement, quelques point a améliorer mais ils sont sur le bon chemin , merci a []

[].....continuer comme ça vous êtes top !!

B []



[] est très bien et l'assistance téléphonique et tchat aussi vraiment très bien et disponible, aimable et professionnel. Toutefois, j'aurais aimé avoir reçu une formation sur le logiciel, même une formation courte d'une journée et même si cette formation est payante ! Car l'optimisation du logiciel est vraiment nécessaire pour faire une gestion correct et sans erreur ou faute. Merci à toute l'équipe !

B []

b) Objet du site

La solution la plus profitable serait de se tourner vers un site de gestion locative, qui permet au propriétaire de garder la main sur sa gestion immobilière, tout en bénéficiant d'une expertise dans différents domaines pour gagner du temps au quotidien !

Notre objectif est de proposer un site de gestion locative qui offre une alternative aux agences immobilières traditionnelles en fournissant des services similaires à moindre coût.

La digitalisation permet véritablement de simplifier certaines démarches chronophages et d'automatiser une partie de la gestion administrative.

Ce site se divisera en plusieurs parties :

- la page d'accueil,
- la page de connexion,
- un tableau de bord différent en fonction des rôles,
- formulaire de contact.

Il y aura une partie monétisation à implémenter. Nous souhaitons faire un site payant avec abonnement selon les besoins (soit uniquement du stockage avec un nombre de 2 baux max pour un abonnement gratuit, ensuite celui de base et enfin premium qui peut aller jusqu'à l'automatisation d'envois des courriers etc).

c) Public cible

Ce site sera entièrement dédié aux particuliers qui se situeront entre 20 et 99 ans.

Il permettra aux petits investisseurs, grâce à son tableau de bord intuitif, de se familiariser avec la gestion de leur bien et de garder un lien avec leurs locataires.

Il permettra également de comprendre les enjeux qui se profilent derrière une bonne gestion immobilière.

d) Notre équipe dédiée

1. **Chef de projet** : Reponsable de la gestion globale du projet et de la communication avec le client,
2. **Designer Web** : il crée l'interface utilisateur (UI*) et l'expérience utilisateur (UX*) pour assurer un design attrayant et fonctionnel,
3. **Développeur Front-end/intégrateur Web** : Reponsable de la programmation des éléments visibles du site, s'assure de son bon fonctionnement du côté client et convertit les maquettes de design en code web fonctionnel,
4. **Développeur Back-end** : il gère la logique serveur, la base de données et les fonctionnalités qui ne sont pas directement visibles par l'utilisateur,
5. **Spécialiste SEO/ rédacteur de contenu** : convertit les maquettes de design en code fonctionnel et optimise le site pour les moteurs de recherche afin d'améliorer sa visibilité en ligne,
6. **Testeur de qualité** : il effectue des tests approfondis pour garantir la qualité et la fonctionnalité du site,
7. **Administrateur Système** : il gère l'hébergement, la sécurité et la performance du site.

e) Budget

Pour une base d'environ 6 mois :

Frais de développement technique comprenant :

- Salaire du Responsable de projet : au taux horaire de 45€/heure, pour 6 mois de travail, à raison de 25 jours par mois, 9h/jour : 60 750€
- Salaire d'un développeur back senior : au taux horaire de 35€/heure, pour 6 mois de travail, à raison de 25 jours par mois, 8h/jour : 42000€,
- Salaire d'un développeur front junior : au taux horaire de 28€/heure, pour 6 mois de travail, à raison de 25 jours par mois, 7h/jour: 29 400€,
- Salaire testeur et administrateur système : au taux horaire de 25€/heure, pour 4 mois de travail à raison de 20 jours par mois, 8h/jour : 32 000€

Frais opérationnels :

- Frais des locaux : 12 000€/an
- Coût d'hébergement et de maintenance : 3 000€/an (appelé à augmenter selon le trafic utilisateur)

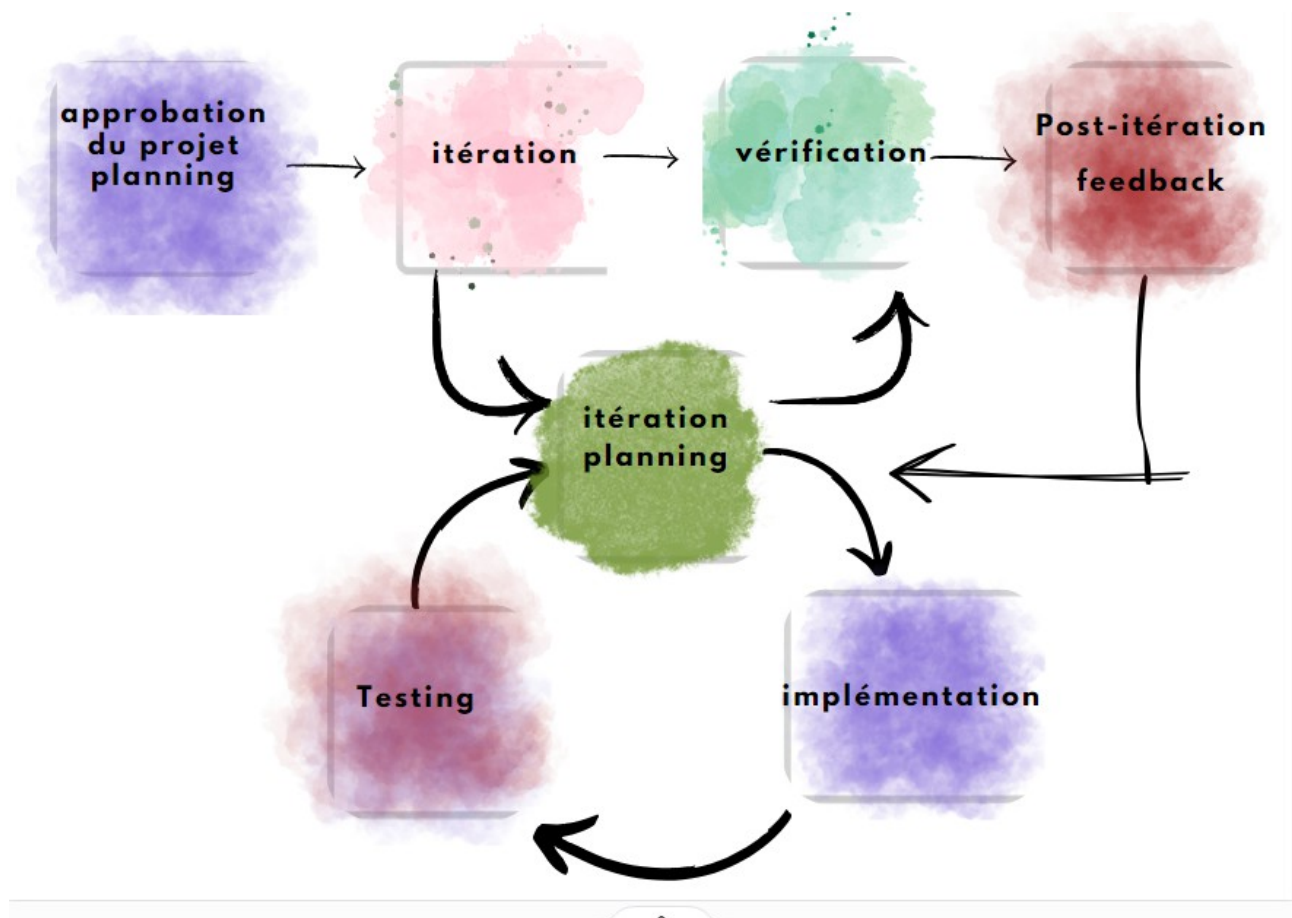
Soit un total de la première année : 179 150€

Considérons une marge d'erreur : 20 % => 214 980€

f) Organisation

L'équipe utilisera la méthode Agile, ce qui permettra de planifier la conception du site par couches et favorisera une vision plus globale de chacune des nouvelles itérations (sprints) et incrémentations, ainsi le produit peut être testé et amélioré en continu tout au long du projet plutôt qu'à la fin.

Notre équipe de développeurs va traduire les maquettes dans les langages informatiques sélectionnés, tels qu'ils seront documentés dans ce cahier des charges. Ces maquettes qui étaient alors des images statiques vont prendre vie et devenir un site à proprement parler. On parle "d'intégration" des maquettes et de programmation des fonctionnalités du site.



g) Les tâches à accomplir

1. Planification du sprint:

- Tenir une réunion de planification de sprint pour définir les objectifs
- Identifier les fonctionnalités prioritaires à implémenter

2. Liste de tâches(Backlog*) du sprint :

- Créer un backlog de sprint avec des user stories* décrivant les fonctionnalités à développer
- Estimer la complexité de chaque user story

3. Sélection des user stories (1 action = 1 fonctionnalité) :

- Choisir les user stories à inclure dans le sprint en fonction des priorités et des estimations

4. Développement :

- Implémenter* les fonctionnalités définies dans les user stories sélectionnées
- Collaborer étroitement avec l'équipe et assurer la communication quotidienne

5. Tests unitaires :

- Ecrire des tests unitaires pour chaque fonctionnalité développée
- S'assurer que chaque user story est couverte par des tests

6. Intégration continue :

- Mettre en place un processus d'intégration continue pour tester le code régulièrement

7. Démo du sprint :

- Organiser une démonstration à la fin du sprint pour présenter les fonctionnalités développées

8. Rétrospective du sprint:

- Tenir une réunion de rétrospective pour évaluer ce qui a bien fonctionné et ce qui peut être amélioré
- Mettre en place des actions d'amélioration pour le prochain sprint

9. Planification du sprint suivant :

- Définir les objectifs pour le prochain sprint en se basant sur les retours de la rétrospective

10. Refonte du backlog produit

- Mettre à jour le backlog produit en fonction des nouvelles informations et des changements

11. Communication continue:

- Assurer une communication régulière avec les parties prenantes pour obtenir des retours et des ajustements

12. Tests utilisateurs continus:

- Encourager les tests utilisateurs fréquents pour obtenir des retours utilisateurs rapides

13. Documentation continue :

- Mettre à jour la documentation en temps réel au fur et à mesure du développement

14. Livraison continue :

- Chercher à délivrer des versions fonctionnelles à la fin de chaque sprint

h) Roadmap

Cette chronologie est une évaluation qui peut varier selon les influences dûes à plusieurs paramètres différents. La souplesse de la méthodologie Agile permet d'adapter le plan sur une base de 6 mois, en fonction des évolutions du projet.

Mois 1-2 : Sprints 1-4

Sprint 1 : Planification initiale, définition des grandes lignes du projet

Sprints 2-4 : Développement initial des fonctionnalités de base, mise en place de l'architecture

Mois 3-4 : Sprints 5-10

Sprints 5-6 : Développement des fonctionnalités de gestion immobilière de base

Sprints 7-8 : Intégration des fonctionnalités complémentaires

Sprints 9-10 : Tests intensifs, réglages et préparations pour la première démo

Mois 5: Sprints 11-12

Sprints 11-12 : Finalisation des fonctionnalités restantes, préparations pour le déploiement

Mois 6 :Sprints 13-14

Sprints 13-14 : Déploiement en production, suivi post-lancement, ajustements en fonction du retour des utilisateurs

Continuité : Itérations ultérieures avec des sprints réguliers pour les améliorations, correctifs et nouvelles fonctionnalités.

i) Les outils collaboratifs choisis

Dans un contexte de responsabilité environnementale et de durabilité nous mettrons l'accent sur les outils collaboratifs éco-responsables autant que faire se peut :

1. Confluence et Jira :

La société Altassian, possédant ces outils, a un engagement dans le cadre de la durabilité. Confluence est utilisé pour la création de contenu collaboratif comme la documentation, la création de bases de connaissances et la collaboration générale au sein d'une équipe alors que Jira est axé sur la gestion de projet et le suivi des problèmes. Il est utilisé pour planifier, suivre, gérer les tâches, les bogues et les projets,

2. Notion :

Notion est une plateforme de collaboration tout-en-un permettant de créer, partager et de coopérer sur des documents, des bases de connaissances et des projets. Il a une interface conviviale qui permet également d'optimiser votre travail grâce à l'IA intégrée et peut être une alternative éco-responsable pour la documentation collaborative,

3. Trello :

Fait également partie de la société Altassian, cet outil utilise une interface de style tableau Kanban (permet de visualiser le travail en cours et maximise l'efficacité) qui est simple et intuitif pour la gestion de projet, sa simplicité peut conduire à une utilisation plus efficace des ressources,

4. Figma :

Figma est un outil de conception collaboratif qui fonctionne dans le navigateur. Bien que principalement axé sur les conceptions d'interfaces utilisateur, il peut également être utilisé pour collaborer sur des prototypes, réduisant ainsi le besoin d'impression papier,

5. Microsoft Team :

Microsoft a pris des engagements en matière de durabilité et Team est son outil de communication et de coopération. Il offre des fonctionnalités telles que le partage de documents, la messagerie instantanée et les réunions en vidéo,

6. Gitlab :

GitLab, en tant que plateforme de gestion de versions et de développement collaboratif, propose des fonctionnalités pour gérer l'intégration continue et le déploiement. Cela peut contribuer à une utilisation plus efficace des ressources,

7. GreenHost :

Pour l'hébergement de sites web, GreenHost est une entreprise d'hébergement qui met l'accent sur l'énergie renouvelable et la durabilité environnementale.

2. Les fonctionnalités clés de mon projet

a) Les besoins fonctionnels de mon site

En restant très succincte, listons les fonctionnalités actuelles de mon site :

1. La gestion des propriétés :

- Ajouter un nouveau logement à la plateforme avec l'adresse,
- La modification qui permet de pouvoir changer les documents liés au logement,
- La suppression autorise de supprimer les biens qui n'existent plus dans le parc du propriétaire,
- La gestion des documents fournit un espace qui permet le dépôt du bail notamment et tout autre document pertinent lié à la propriété,
- Affiche les biens qui sont disponibles.

2. La gestion des locataires :

- Gestion des locataires : permet au propriétaire de lier un logement au locataire,
- Gestion des contrats de location : facilite l'intégration et la lecture des baux,
- Suivi de l'historique des locataires : permet de savoir si le locataire est toujours en place ou si le logement est disponible.

3. La gestion des contrats de location :

- Stockage sécurisé des baux en garantissant la confidentialité des informations,
- L'historique des baux y compris les versions précédentes avec les modifications apportées.

4. Accessibilité et authentification:

- Interface conviviale et permettant un accès facile aux informations et fonctionnalités pertinentes,
- Permet de se connecter grâce à son mail et un mot de passe, et de définir le rôle de la personne accédant au site, soit propriétaire soit locataire.

b) A l'avenir, quelles améliorations ?

Un site est en constante évolution, selon les demandes des clients, il y aura donc des améliorations à apporter pour un site optimal :

- Savoir si le loyer a été payé en temps et en heure, avec identification du logement, du montant du loyer hors charge et le montant des charges,
- L'envoi de la quittance de manière automatisée à réception du règlement,
- L'envoi de courriers automatisés de diverses relances,
- Un mémo des dates d'entrée dans les lieux avec une alerte sur les dates de fin, notamment pour les contrats de courte durée,
- L'intégration des états des lieux, de divers échanges pertinents liés au logement, entre locataires et propriétaires,
- L'intégration des montants des impôts, des crédits en cours, avec le nom des banques par logement.

3. Conception et structure

a) Mise en place des bonnes pratiques en éco-conception

En intégrant ces pratiques d'éco-conception dès la phase de conception, nous pourrons contribuer à réduire l'empreinte carbone tout en offrant un site de qualité:

1. Architecture :

Ici nous adopterons une architecture légère et modulaire pour minimiser la consommation de ressource de système pour éviter de surcharger inutilement la plateforme ce qui augmenterait la consommation d'énergie.

2. Langages de programmation :

Nous allons utiliser des langages de programmation efficaces en optimisant le code pour réduire la consommation énergétique.

3. Optimisation des médias et scripts :

Nous nous engageons à optimiser à la fois les médias, tels que les documents et photos utilisés sur le site. Cela garantit que les fichiers sont légers et que les scripts sont efficaces, réduisant ainsi la demande d'énergie pour leur exécution.

4. Responsive* :

Notre approche du responsive design est également centrée sur l'efficacité énergétique et l'accessibilité. En optimisant le site pour qu'il s'adapte parfaitement à différentes tailles d'écrans, nous réduisons la nécessité de charger des ressources supplémentaires ou inadaptées.

Cela se traduit par une consommation d'énergie moindre, tant du côté des appareils utilisateurs que de nos serveurs. De plus, un design responsive améliore l'accessibilité du site, en garantissant que les utilisateurs bénéficient d'une expérience fluide et intuitive, quelle que soit la taille de l'écran.

5. Hébergements responsables :

Nous utiliserons des serveurs éco-énergétiques. Nous privilégierons ceux utilisant des sources d'énergie renouvelable ou mettant en place des pratiques de réduction de leur empreinte carbone. Cela contribuera à minimiser l'impact environnemental de notre infrastructure, même lorsque nous utiliserons des scripts.

6. Accessibilité :

Nous maintenons nos engagements pour l'accessibilité en veillant à ce que l'utilisation de scripts ne soit pas discriminatoire envers les utilisateurs ayant des besoins spécifiques en matière d'accessibilité. Nous nous assurerons que les balises sémantiques soient utilisées correctement et que les attributs liés à l'accessibilité spécifique soient mis en place. Le site sera conforme aux directives WCAG¹ (Web Content Accessibility Guidelines) pour assurer une expérience aux utilisateurs en situation de handicap, notamment en proposant des alternatives avec une structure logique et une bonne lisibilité.

7. L'impact environnemental :

Nous évaluons régulièrement l'empreinte carbone de notre plateforme, y compris celle liée à l'utilisation de scripts.

1: <https://www.w3.org/WAI/standards-guidelines/wcag/fr>

LocalLinkImmo s'efforce de concevoir et de développer son site de manière responsable, en tenant compte de l'utilisation de scripts pour améliorer l'expérience des utilisateurs tout en minimisant l'impact environnemental

e) Structure du site

Les pages principales de l'application sont au nombre de 4 (voir annexes). Certaines ne sont accessibles qu'après avoir rempli la page précédente ou après s'être connecté.

Accueil

La page d'accueil sera composée d'une barre de navigation permettant la navigation entre les différentes sections du site, du logo et de la description du site.

Section userLog

Cette page permettra la création d'un compte (First connexion) avec l'email qui liera le rôle soit du propriétaire (owner) soit du locataire (tenant), le mot de passe et sa vérification.

Section userLog connexion

Cette section permettra la connexion et la direction du tableau de bord (dashboard) qui sera doté des commandes de création de logement (home), de document selon le rôle attribué.

Section Contact

Cette dernière page permettra le contact avec de l'aide via un formulaire.

4. Les contraintes techniques

La sécurité des données :

Assurer la sécurité des données utilisateurs est notre priorité y compris les informations personnelles. Il est essentiel de mettre en place des protocoles de sécurité robustes et de prévenir les attaques potentielles.

Côté serveur Node.JS et API :

- La validation des entrées : Nous prioriserons la validation et le filtrage de toutes les entrées utilisateurs pour prévenir les attaques par injection comme celles XSS (framework express-validator) et SQL injection,
- L'utilisation des packages sécurisés : Nous choisirons des modules npm (Node Package Manager) réputés et maintenus pour éviter les vulnérabilités connues. Des mises à jour doivent être faites régulièrement, afin d'éviter les risques liés à ces faiblesses,
- Authentification : Il y aura une mise en place de mécanisme d'authentification robuste, tel que les JWT (JSON Web Tokens) dans le but de vérifier l'identité de l'utilisateur à chaque requête.

Côté client React.JS :

- Performance : Nous nous assurerons que le site soit réactif et fluide en évitant les opérations coûteuses en ressources et en optimisant le code des composants,
- Politique du mot de passe fort : Exiger des mots de passe complexes (longueur minimale),
- Hachage et salage : Stocker les mots de passe en utilisant un hachage robuste (comme bcrypt) avec salage pour prévenir les attaques,

Protection contre les attaques spécifiques :

- Injection Sql : Utiliser des requêtes préparées et l'ORM* pour interagir avec la base de données,
- Overlaying* : Ou overlay attack sera de gérer correctement les permissions pour prévenir l'utilisateur si une autre application tente de se superposer à la nôtre.

Sensibilisation et transparence envers les utilisateurs :

- Politique de confidentialité claire : Expliquer comment les données sont utilisées, protégées et stockées. Rendre cette politique accessible et facile à comprendre,
- Formation des utilisateurs : Informer les utilisateurs sur les bonnes pratiques de sécurité, comme choisir un mot de passe fort ou reconnaître les emails de phishing.

Tests et audits de sécurité :

- Tests d'intrusion : Faire régulièrement des tests d'intrusion afin d'identifier et de corriger les vulnérabilités,
- Audit de code : Faire auditer notre code par des experts en sécurité à l'effet de détecter les failles potentielles.

Performance et scalabilité :

Nous garantirons que le site puisse gérer un grand nombre d'utilisateurs simultanément. Pour améliorer la scalabilité* et la performance, nous optimiserons donc les requêtes avec un code simple et facilement scalable pour les mises à jour.

Base de données :

Concevoir une base de données robuste pour stocker les différentes informations. La gestion efficace des bases de données est essentielle pour garantir des temps de réponses rapides.

Mise à jour et maintenance :

Prévoir des procédures de mise à jour régulières pour maintenir le site, corriger les bugs et ajouter de nouvelles fonctionnalités.

Hébergement :

Nous sélectionnerons un service d'hébergement web fiable et évolutif pour garantir que le site reste en ligne et soit performant même en cas de montée en charge.

Sauvegarde des données :

Mettre en place des procédures de sauvegardes régulières pour éviter la perte de données en cas de problème technique.

Législation sur la protection de la vie privée :

Respecter les lois et réglementations locales et internationales concernant la protection de la vie privée des utilisateurs, telle que la RGPD en Europe.

Coûts

Evaluer les coûts associés à la création, à l'hébergement et à la maintenance du site, en tenant compte du budget disponible.

Evolutivité :

Penser à l'avenir à la possibilité d'ajouter de nouvelles fonctionnalités à mesure que le site se développe.

Expérience d'utilisation :

Chercher à offrir une expérience utilisateur fluide et intuitive. Cela inclut une interface conviviale, une navigation facile à comprendre, des temps de chargement rapides, des images optimisées et une ergonomie de navigation adéquate.

SEO

Le SEO (Search Engine Optimization) sera optimisé afin d'améliorer la visibilité et la pertinence d'un site aux yeux des moteurs de recherche, ce qui entraîne une augmentation du trafic organique et une meilleure expérience utilisateur.

Pour atteindre cet objectif, le SEO implique plusieurs actions :

- Optimisation des mots-clés : l'intégration de mots-clés stratégiques liés au thème de mon site tels que « autogestion », « biens immobiliers », « location », « propriétaire », « locataire », « gestion », « immobilier », « logements », « administration », est importante et nous prêterons également une attention particulière aux mots-clés secondaires comme « contrat », « propriétés », « plateforme », « patrimoine » pour attirer plus d'utilisateurs.
- Optimisation des balises : L'utilisation appropriée des balises HTML (titre, méta-description, balises H1-H6, attributs ALT pour les images etc.) pour aider les moteurs de recherches à comprendre le contenu de la page,

- Choisir le bon nom: Local Link Immo à été choisi pour définir le lien local du propriétaire avec son patrimoine immobilier qu'il va pouvoir gérer en toute autonomie,
- Analyse et suivi: l'utilisation d'outils d'analyse pour surveiller les performances du site, identifier les opportunités d'amélioration et ajuster la stratégie SEO en conséquence.

5. Technologies utilisées

a) Front-end

Ici nous utiliserons la bibliothèque la plus connue qui se trouve être React.js, qui a été développée par des ingénieurs de Facebook (meta) pour la création d'interfaces dynamiques et interactives. C'est donc tout naturellement que nous nous tournerons vers celle-ci.

React.JS

React est une bibliothèque Javascript pour la construction d'interfaces utilisateurs que l'on pourra diviser en petits composants réutilisables et modulaires rendant l'architecture du code plus efficace et maintenable.

React utilise un concept appelé le « DOM* (Document Object Model) virtuel » pour améliorer les performances en minimisant les manipulations du DOM réel. Plutôt que de manipuler directement le DOM réel à chaque mise à jour, React compare les versions précédentes et actuelles du DOM virtuel et met à jour le DOM réel uniquement s'il y a eu changement. Cette action rend les mises à jour de l'interface utilisateur plus rapides et efficaces. Il ne sera pas nécessaire de redémarrer l'application pour voir les modifications car elles se font instantanément et cela améliore considérablement l'expérience de développement et accélère le processus de test et de débogage.

Avec la croissance rapide de React et de sa communauté d'utilisateurs, il existe une gamme croissante de bibliothèques de supports, en particulier les bibliothèques de composants UI. Ces bibliothèques nous font gagner beaucoup de temps et d'efforts lors de la création d'applications React graphiques et modernes.

Elles fournissent de nombreux composants prêts à l'emploi tels que : des jeux d'icônes, de boutons, de sélecteurs, d'entrées de formulaires, de calendriers, de menus, de paginations, de cartes, et bien d'autres encore.

JSX

JSX est une extension syntaxique pour Javascript utilisée en React. Elle permet d'écrire du code HTML de manière déclarative à l'intérieur de Javascript, en les combinant en un seul fichier. Cela permet aux développeurs de définir facilement l'interface utilisateurs de leurs composants React en utilisant une syntaxe familière et expressive, tout en bénéficiant de la puissance de Javascript pour la logique dynamique et les manipulations de données.

SCSS

SCSS est une extension de syntaxe de CSS qui offre des fonctionnalités avancées pour la création de feuilles de style plus puissantes et plus organisées. La syntaxe est améliorée et offre des fonctionnalités supplémentaires et une meilleure organisation du code par rapport au CSS standard. Le SCSS permet l'utilisation de variables, de fonctions, de boucles et de directives conditionnelles, ce qui facilite la gestion et la maintenance des feuilles de style complexes.

Il y a la possibilité d'utiliser plusieurs approches pour styliser un site en React, mais nous utiliserons le styled-components qui permet de définir des styles de manière plus dynamique et réactive, puisque cette bibliothèque permet d'écrire des styles directement dans les composants React à l'aide de modèles de chaînes. Ce qui rend la maintenabilité plus facile.

b) **Back-end**

Node. JS

Node. JS sera la solution que nous utiliserons. C'est un environnement open source pour le Javascript qui permet l'exécution de code Javascript côté serveur. Il est construit sur le moteur Javascript V8 de Google Chrome et permet de développer des applications réseau rapidement et efficacement. Node.JS fonctionne sur un modèle d'entrées/sorties non bloquant et orienté événement, ce qui le rend particulièrement adapté pour des applications nécessitant des performances élevées en matière de traitement des données en temps réel. Node.JS est également reconnu pour son écosystème riche en modules, géré via le Node Package Manager (NPM), qui offre une vaste gamme de bibliothèques et d'outils facilitant le développement d'applications.

MySQL

C'est système de gestion de bases de données relationnelle (SGBDR) en open source largement utilisé. C'est l'un des outils les plus populaires pour l'administration de bases de données MySQL en raison de sa simplicité d'utilisation et de sa grande variété de fonctionnalités.

Voici quelques-unes de ces fonctionnalités principales :

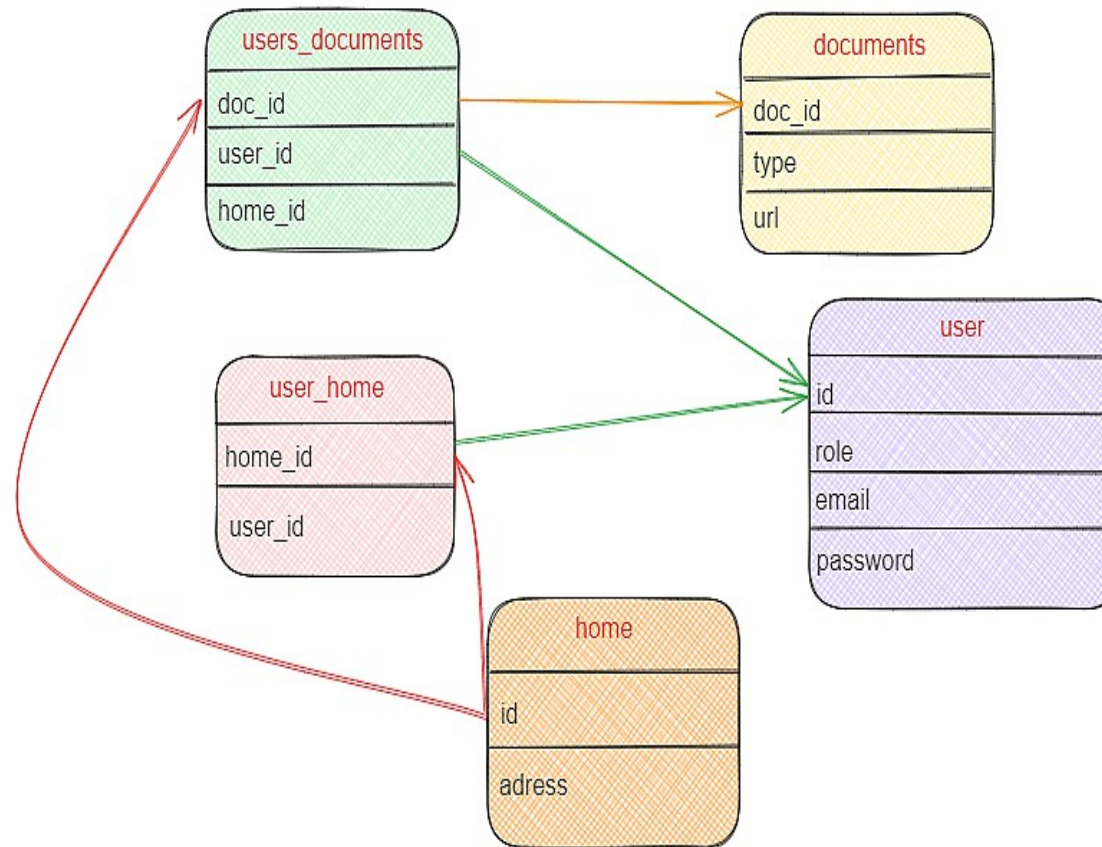
- Gestion des bases de données : permet de créer, modifier et supprimer des bases de données MySQL, ainsi que de gérer leurs paramètres et propriétés.
- Gestion des tables : il offre des fonctionnalités pour créer, modifier et supprimer des tables MySQL, ainsi que définir des clés primaires, des index et des contraintes.

- L'édition des données: permet d'insérer, mettre à jour et supprimer des données dans les tables MySQL à l'aide d'une interface conviviale.
- Importation et exportation de données : phpmyadmin prend en charge l'importation et l'exportation de données à partir de différents formats de fichiers, tels que CSV, SQL, XML, etc
- Gestion des utilisateurs et des privilèges : il permet de gérer les utilisateurs MySQL et leurs privilèges, y compris la création, la modification et la suppression des comptes utilisateurs.
- Visualisation des statistiques et des informations sur le serveur : phpmyadmin fournit des informations détaillées sur le serveur MySQL, telles que les statistiques d'utilisation des ressources, les processus en cours d'exécution etc.

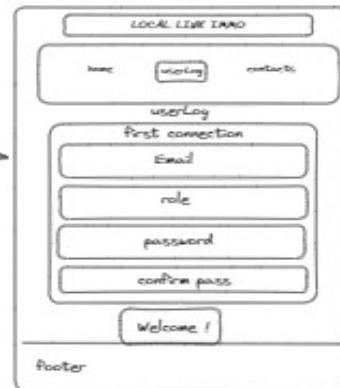
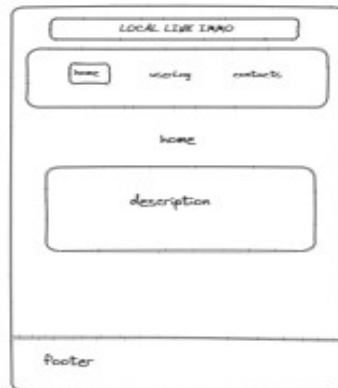
Il est possible d'accéder aux fonctionnalités de gestion de bases de données en utilisant des méthodes non interactives telles que l'envoi de requêtes directement aux scripts PHP sous-jacents de phpmyadmin. Cependant cela n'est pas recommandé car phpmyadmin n'est pas à utiliser de cette manière, nous utiliserons donc une solution plus appropriée comme la bibliothèque « mysql » qui permet l'accès aux bases de données via des API RESTful* (Representational State Transfer) tels que GET, POST, PUT ou DELETE pour effectuer des opérations CRUD (Create, Read, Update et Delete).

c) Architecture logicielle

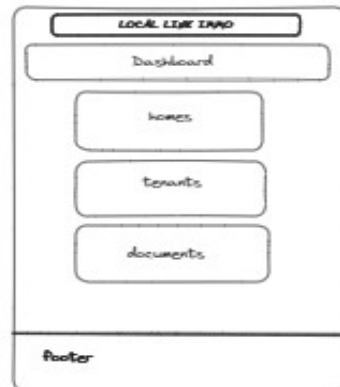
Schéma de mes tables



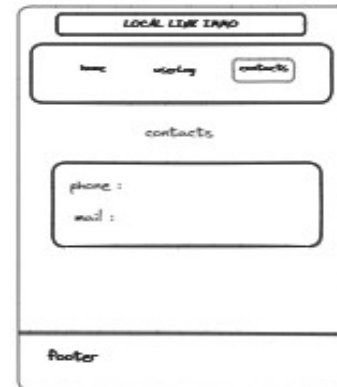
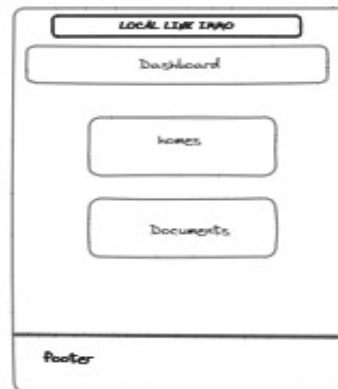
Wireframe site applicatif Local Link Inno



OWNER



Tenant



d) Veilles techniques

Le projet va s'étaler sur plus de 6 mois, or on le sait, les langages, frameworks*, librairies* et autres sont en constante évolution. Il est essentiel de faire de la veille quotidienne pour se tenir informé des changements qui peuvent impacter notre site.

Nous avons la chance de faire partie d'une communauté très portée sur le partage d'informations et les moyens de faire de la veille efficace sont nombreux.

Nous avons sélectionné deux canaux différents :

- daily.dev¹ ou Feedly sont des agrégateurs d'articles sous forme d'extension de navigateur permettant, par un système extrêmement bien fait de filtres, de proposer un contenu pertinent,
- X (anciennement Twitter), BlueSky et LinkedIn comptent de nombreux développeurs passionnés qui partagent leurs astuces et leurs découvertes. Il existe une communauté de développeurs sur chaque réseau social et ils sont souvent réactifs et bienveillants pour répondre aux questions.

1. <https://daily.dev/>

Lexique

UI (p.6) :

Ou User Interface concerne l'environnement graphique du site dans lequel évolue l'utilisateur.

UX(p.6) :

Ou User Experience c'est la conception d'interface accessible facile à prendre en main pour tout type de support.

Backlog(p.9) :

Liste ordonnée de tâches de fonctionnalités ou de problèmes à résoudre.

User stories (p.9) :

Illustrent la façon dont les équipes produit envisagent les interactions avec l'application.

Implémenter(p.9) :

Mettre en œuvre un plan en le rendant fonctionnel dans un programme informatique.

Responsive (p.16) :

Site conçu de façon à pouvoir s'adapter à toutes les différentes résolutions d'écran.

ORM (p.20) :

Ou Object Relationnal Mapping est un type de programme informatique qui convertit les données entre les tables pour générer un code SQL d'une base de données relationnelle avec des CRUD.

Overlay attack(p.20) :

Est une technique permettant à un hacker de superposer une interface utilisateur malveillante ou trompeuse sur une application ou un site légitime afin de capturer des informations sensibles.

Scalabilité(p.20) :

Capacité de l'application à se comporter correctement malgré un grand nombre d'utilisateurs

DOM (p.24) :

Document Object Model est une interface de programmation qui représente une structure en arbre d'un document, permettant aux développeurs d'accéder et de manipuler le contenu et le style de la page via du code Javascript.

API RESTful (p.27) :

Ensemble de règles pour construire des services web qui utilise le protocole HTTP pour communiquer et échanger des données de manière simple et cohérente.

Librairies (p.30) :

Ensemble de codes pré-écrits et compilés, regroupés dans des modules ou des packages, qui offrent une fonctionnalité spécifique.

Framework (p.30) :

Ensemble de bibliothèques ou d'outils utilisés offrant des fonctionnalités préconçues pour accélérer le processus de développement.