



UNIVERSITÉ DE LORRAINE

Filière : Ingénierie Logicielle (IL)

Module : Technologie Web

Cahier des charges du projet web BuildWealth

Projet académique

Étudiants :

- Kaouthar ASRAR
- Meriem AIT EL ACHKAR

Enseignant :

- Simon Hajek

1^{er} octobre 2025

Table des matières

1 Présentation générale du projet	2
1.1 Contexte	2
2 Objectifs du projet	2
2.1 Objectif général	2
2.2 Objectifs spécifiques	2
3 Public cible	3
3.1 Description générale	3
3.2 Segments de public cible	3
4 Fonctionnalités attendues	3
4.1 Description générale	3
4.2 Fonctionnalités principales	3
4.3 Fonctionnalités techniques complémentaires	4
5 Spécifications techniques	4
5.1 Architecture générale	4
5.2 Sécurité et authentification	4
5.3 Environnements et déploiement	4
5.4 Standards de développement	4
6 Contraintes du projet	5
7 Livrables attendus	5
8 Tests et validation	5
8.1 Objectif	5
8.2 Types de tests	5

1 Présentation générale du projet

1.1 Contexte

Le secteur immobilier évolue rapidement avec la digitalisation. Les investisseurs recherchent des solutions d'investissement accessibles, tandis que les promoteurs souhaitent diversifier leurs sources de financement. Parallèlement, les acheteurs souhaitent pouvoir acquérir des biens de manière simple, rapide et sécurisée, sans intermédiaires traditionnels.

Le crowdfunding immobilier permet de financer collectivement des projets de construction ou de rénovation. La vente en ligne de biens immobiliers, quant à elle, simplifie l'accès à la propriété.

Notre projet combine ces deux aspects en une plateforme web moderne de crowdfunding et de vente immobilière, qui :

- permet aux promoteurs de lever des fonds pour leurs projets,
- donne aux investisseurs la possibilité de financer et suivre leurs investissements,
- propose aux acheteurs d'acheter directement des biens immobiliers livrés ou en construction.

2 Objectifs du projet

2.1 Objectif général

Créer une application web innovante qui combine financement participatif et vente immobilière, offrant une expérience transparente, sécurisée et complète pour tous les acteurs du marché immobilier.

2.2 Objectifs spécifiques

- Offrir une interface intuitive et responsive pour consulter, financer et acheter des biens immobiliers.
- Mettre en place un système d'authentification sécurisé et une gestion des rôles (investisseurs, promoteurs, acheteurs, administrateurs).
- Permettre aux promoteurs de publier et gérer leurs projets, lever des fonds et suivre leur progression
- Permettre aux investisseurs de participer à des projets immobiliers à partir de petits montants, suivre leurs investissements et calculer leurs rendements.
- Proposer un catalogue de biens immobiliers à vendre, avec filtres et moteur de recherche.
- Offrir un tableau de bord financier complet pour suivre les investissements et les rendements.
- Mettre en place un processus d'achat immobilier en ligne, avec simulation de prêt et génération automatique de contrats
- Assurer la sécurité des données, la conformité RGPD, et la protection des transactions.

3 Public cible

3.1 Description générale

La plateforme vise à réunir tous les acteurs du marché immobilier sur une même solution : promoteurs, investisseurs et acheteurs.

3.2 Segments de public cible

- **Promoteurs immobiliers** : lever des fonds, publier des projets, vendre des biens.
- **Investisseurs** : investir dans des projets immobiliers et suivre les rendements.
- **Acheteurs** : acheter des biens immobiliers livrés ou en construction.
- **Administrateurs** : superviser, valider les projets, gérer les utilisateurs et garantir la conformité.

4 Fonctionnalités attendues

4.1 Description générale

La plateforme couvrira l'ensemble du cycle immobilier : financement, suivi de projet, vente et post-achat

4.2 Fonctionnalités principales

- **Gestion des utilisateurs** : Création de comptes, authentification via JWT, gestion des rôles (promoteur, investisseur, acheteur, admin).
- **Gestion des projets de crowdfunding** :
 - Publication, validation, suivi et mise à jour de projets immobiliers.
 - Participation aux projets avec historique des investissements.
 - Calcul automatique du rendement et échéancier de remboursement.
- **Vente directe de biens immobiliers** :
 - Catalogue d'annonces immobilières consultable avec filtres (prix, localisation, type, surface, etc.).
 - Système de réservation et de suivi d'achat.
 - Simulation de prêt immobilier et calcul des mensualités.
- **Messagerie interne** : Échanges sécurisés entre investisseurs, promoteurs et acheteurs.
- **Transparence et confiance** : Statuts détaillés des projets, reporting d'avancement, avis et retours d'expérience..
- **Dossier locataire numérique** : Mise à disposition d'un espace permettant aux locataires/acheteurs de constituer un dossier numérique sécurisé et partageable avec les propriétaires.

4.3 Fonctionnalités techniques complémentaires

- Respect des bonnes pratiques de sécurité et RGPD.
- API REST pour une architecture évolutive.
- Documentation technique complète.

5 Spécifications techniques

5.1 Architecture générale

L'application sera conçue en architecture *Full-Stack* avec une séparation claire entre le **frontend** et le **backend**. Le frontend communiquera avec le backend via une API REST sécurisée, et le backend interagira avec une base de données NoSQL.

- **Frontend** : Développé en React, avec hooks et React Router pour le routage. Gestion d'état cohérente via Context API ou Redux.
- **Backend** : Développé en NestJS avec architecture modulaire, services, contrôleurs et validation des données.
- **Base de données** : MongoDB avec Mongoose pour la gestion des schémas et des relations.

5.2 Sécurité et authentification

L'application doit assurer la sécurité des données et des échanges utilisateurs :

- **Authentification et autorisation** : Utilisation de JWT pour protéger les routes sensibles.
- **Gestion des droits** : Différenciation des permissions selon les rôles (propriétaires, locataires/acheteurs, administrateurs).
- **Protection des données personnelles** : Conformité au RGPD (chiffrement, anonymisation si nécessaire).

5.3 Environnements et déploiement

Le projet doit être configuré pour fonctionner sur différents environnements (développement, test, production) :

- **Variables d'environnement** : Utilisation de fichiers .env pour les clés et paramètres sensibles.
- **Déploiement** : Hébergement sur une plateforme cloud (Heroku, Vercel, Render ou équivalent) pour rendre l'application accessible en ligne.
- **Documentation technique** : Fourniture d'un README détaillé et d'une documentation d'installation.

5.4 Standards de développement

Le code doit respecter les standards de qualité professionnelle :

- **Clean code** : Respect des principes DRY et SOLID, code commenté et structuré.
- **Gestion des erreurs** : Mise en place de middlewares pour la gestion globale des erreurs dans le backend.
- **Versioning** : Utilisation cohérente de Git avec des commits explicites.

6 Contraintes du projet

- **Projets parallèles** : L'équipe travaille sur plusieurs projets issus d'autres modules en parallèle, ce qui limite le temps disponible pour ce projet.
- **Délais de livraison** : La présentation du projet est prévue la dernière séance de cours, à 20h dans la salle de classe.
- **Planification serrée** : Les différentes étapes (analyse, conception, développement, tests) doivent être planifiées pour respecter ce délai.
- **Ressources limitées** : Les environnements de développement et de déploiement devront être configurés avec des outils gratuits ou académiques.

7 Livrables attendus

- Code source complet
- Repository Git
- Documentation technique et fonctionnelle
- Présentation finale
 - Démonstration fonctionnelle de l'application.
 - Support visuel (slides) expliquant le concept et la mise en œuvre.
- Déploiement en ligne (bonus)

8 Tests et validation

8.1 Objectif

Vérifier que l'application respecte les spécifications fonctionnelles et techniques et qu'elle fonctionne de manière fiable et sécurisée.

8.2 Types de tests

- **Unitaires** : Vérifier chaque composant ou service.
- **Intégration** : Vérifier l'interaction frontend-backend-base de données.
- **Fonctionnels** : Tester les principales fonctionnalités (authentification, annonces, messagerie...).
- **Sécurité** : Contrôler la protection des données et des accès.