

CURRICULUM VITAE

ANDONIAINA Isaia Mohamed

Adresse : Igaga Fianarantsoa 301

Situation matrimoniale : Célibataire

Date de naissance : 21 mai 2003

Téléphone : 032 90 707 29

E-mail : mohamedesaie21@gmail.com



FORMATIONS

2024 – 2025 : Deuxième année en Licence Professionnelle à l'Ecole Nationale d'Informatique. Parcours : Informatique Générale.

2023 – 2024 : Première année en en Licence Professionnelle à l'Ecole Nationale d'Informatique. Parcours : Informatique Générale.

2022-2023 : Formation professionnelle en développement web chez Gasy Tech, Antananarivo

2021 – 2022 : Obtention du diplôme de Baccalauréat de l'enseignement Générale série S, Lycée PHILIBERT TSIRANANA Mahajanga.

COMPETENCES

- **Programmations :**
 - Langage de programmation : C, C++, C#, JAVA, PHP.
 - Web : HTML, CSS, JavaScript.
- **Framework :** React JS, TailwindCss, Laravel, Livewire.
- **Système de Gestion de Base de Donnée :** SQLite, Maria DB, PostgreSQL.
- **Outils :** Git, GitHub.
- **Méthodologie :** Merise.
- **Système d'exploitation :** Linux (Ubuntu, Debian), Windows (7, 10, 11).
- **Bureautique :** Microsoft Office (Word, Excel, PowerPoint)

REALISATIONS

- Développement d'une API REST complète pour une application mobile utilisée par les employés afin de soumettre des données d'évaluation et des scores de performance. Création de l'interface back-office complète avec Laravel Livewire et TailwindCSS, permettant aux équipes RH de gérer les employés, d'analyser les indicateurs de performance, de visualiser des tableaux de bord en temps réel et de générer des rapports d'évaluation structurés.

- Conception et développement d'une API REST Laravel pour un client du secteur aéronautique, permettant la simulation de vols, l'appariement automatique des trajets aller-retour, le calcul des coûts des passagers par classe et la génération de rapports détaillés sur les revenus. Focus sur le développement des endpoints de l'API, l'implémentation de la logique métier, la gestion sécurisée et efficace des données, et la collaboration étroite avec l'équipe frontend pour l'intégration des fonctionnalités.
- Développement d'une application web de gestion des paiements des factures d'eau et d'électricité, utilisant le framework Laravel avec le moteur de templates Blade et une base de données MySQL. Responsable du développement backend et frontend, incluant l'intégration de la base de données, la gestion des utilisateurs et la conception de l'interface.

LANGUES

Langue	Lecture	Parler	Ecriture
Français	Bien	Assez-bien	Assez-bien
Anglais	Bien	Assez-bien	Bien

SOMMAIRE

CURRICULUM VITAE	1
SOMMAIRE	3
REMERCIEMENTS	6
LISTE DES FIGURES	7
LISTE DES TABLEAUX	8
LISTE DES ABREVIATIONS	9
INTRODUCTION GENERALE	11
PARTIE I – PRESENTATIONS	12
Chapitre 1 : PRESENTATION DE L’ECOLE NATIONALE D’INFORMATIQUE	13
1.1. Information d’ordre générale	13
1.2. Missions et historiques	13
1.3. Organigramme institutionnel	15
1.4. Domaine de spécialisation	15
1.5. Architecture des formations pédagogiques	16
1.6. Relation de l’ENI avec les organismes externes	17
1.7. Débouchés professionnels et diplômés	18
1.8. Ressources humaines	20
Chapitre 2 : PRESENTATION DE MASOVIA MADAGASCAR	21
2.1. Introduction	21
2.2. Évolution de la structure organisationnelle	21
2.3. Année 2023 : Création et mise en place de l’entreprise	21
2.3.1 Contexte de création	21
2.3.2 Premières activités	21
2.3.3 Difficultés rencontrées	22
2.4. Année 2024 : Expansion et réalisation de projets majeurs	22
2.4.1 Présentation des projets réalisés	22
2.5. Année 2025 : Consolidation et diversification des activités	22
2.5.1 Axes stratégiques	23
2.5.2 Nouveaux domaines d’intervention	23
2.6. Vision et perspectives d’avenir	23
2.7. Conclusion	23
Chapitre 3 : DESCRIPTION DU PROJET	24
3.1. Formulation	24
3.2. Objectif et besoins de l’utilisateur	24
3.2.1. Objectifs	24

3.2.2. Besoins des utilisateurs	24
3.3. Moyens nécessaires à la réalisation du projet.....	25
PARTIE II – ANALYSE ET CONCEPTION	27
Chapitre 4. ANALYSE PREALABLE	28
4.1 Analyse de l'existant	28
4.2 Critique de l'existant	28
4.3 Conception avant-projet.....	29
Chapitre 5. ANALYSE CONCEPTUELLE	33
5.1 Présentation de la méthode Merise.....	33
5.1.1 Le Cycle de Vie.....	33
5.1.2 Le Cycle de Décision :.....	33
5.2 Dictionnaire des données	33
5.3 Règles de gestion.....	34
5.4 Diagramme de flux.....	35
5.5 Modélisation des données	35
5.5.1. Modèle Conceptuel des Données (MCD).....	35
5.5.2 Modèle Logique des Données (MLD)	36
5.5.3 Modèle Conceptuel des Traitements (MCT)	37
5.5.4 Organisationnel des Traitements (MOT).....	38
PARTIE III – REALISATION	40
Chapitre 6. MISE EN PLACE DE L'ENVIRONNEMENT	41
6.1 Installation et configuration des outils	41
6.1.1 L'environnement de développement (IDE) : Visual Studio Code.....	41
6.1.2 Serveur Web et PHP : XAMPP	42
6.1.3 Système de Gestion de Base de Données : MySQL	43
6.1.4 Gestion de versions : Git.....	43
6.1.5 Installation de PHP et Composer.....	43
6.1.6 Installation de Laravel, Livewire et Vite.....	43
6.2 Architecture de l'application	44
6.2.1. Couche Présentation (Frontend).....	44
6.2.2. Couche Logique Métier (Backend).....	44
6.2.3. Architecture de l'application Couche Données (Base de données MySQL)	44
6.2.4. Interaction entre les couches	44
Chapitre 7. DEVELOPPEMENT DE L'APPLICATION.....	46
7.1. Création de la base de données	46
7.1.1. Création des tables à l'aide des migrations.....	47
7.2. Codage de l'application	48
7.2.1. Codage de la partie Frontend (Laravel Blade & Livewire).....	48

7.2.2. Codage de la partie Backend (Laravel)	49
7.3. Présentation de l'application.....	50
7.3.1. Page d'accueil	50
7.3.2. Inscription et authentification.....	52
7.3.3. Réservation d'une maison d'hôte	52
7.3.4. Tableau de bord Administrateur	53
7.3.4.1. Tableau de bord Administrateur principal (Super Administrateur)	53
7.3.4.2. Tableau de bord Propriétaire de maison d'hôte	54
7.3.5. Autres pages de l'application	55
CONCLUSION GÉNÉRALE	63
BIBLIOGRAPHIE.....	64
WEBOGRAPHIE.....	65
GLOSSAIRE	66
RESUME	67
ABSTRACT	67

REMERCIEMENTS

Premièrement, Je tiens tout d'abord à remercier le Dieu tout-puissant de m'avoir donné la santé et le courage pour l'accomplissement de mon stage. Je souhaite également adresser mes remerciements les plus sincères à toutes les personnes qui m'ont aidé à mener à bien ce stage. Mes sincères remerciements à :

- Monsieur HAJALALAINA Aimé Richard, Docteur HDR, Président de l'Université de Fianarantsoa, pour m'avoir inscrit dans cet établissement
- Monsieur MAHATODY Thomas, Docteur HDR, Directeur de l'École Nationale d'Informatique, qui m'a donné l'opportunité de réaliser ce projet.
- Monsieur NIHANARISON Franck Ulsy, Directeur Général de la société MASOVIA MADAGASCAR, pour m'avoir permis de réaliser mon stage au sein de cette société.
- Monsieur RALAIVAO Jean Christian, Chef de mention Informatique, pour son encadrement académique et son suivi tout au long de notre parcours universitaire.
- Monsieur RANDRIANOMENJANAHARY Lala Ferdinand, mon encadreur pédagogique, merci d'avoir rendu possible le projet que j'ai élaboré et de m'avoir partagé vos compétences.
- Tous les enseignants de l'École Nationale d'Informatique, qui ont partagé leur expérience pour enrichir mes connaissances et consacré leur temps afin d'apporter la meilleure valeur à ce rapport.
- Aux équipes de MASOVIA MADAGASCAR, qui m'ont chaleureusement accueilli(e) et soutenu(e) tout au long de mon stage, en partageant leur expertise et leur expérience professionnelle.
- À ma famille, qui a toujours été à mes côtés et m'a soutenue depuis le début du stage jusqu'à maintenant.
- Et à tous les collègues et amis de l'École Nationale d'Informatique de Fianarantsoa pour leur soutien moral, financier et matériel.

LISTE DES FIGURES

Figure 1: Organigramme de l'Ecole Nationale d'Informatique	15
Figure 2: Architecture des études correspondant au système LMD.....	17
Figure 3: Chronogramme de travail	26
Figure 4: Diagramme de Flux de Données (DFD).....	35
Figure 5: Modèle Conceptuel des Données (MCD).....	36
Figure 6: Modèle Logique des Données (MLD).....	37
Figure 7: Modèle Conceptuel des Traitements (MCT)	38
Figure 8: Modèle Organisationnel des Traitements (MOT).....	39
Figure 9: Interface de VS Code.....	42
Figure 10: L'architecture client–serveur de l'application	45
Figure 11: Configuration du fichier .env du projet Laravel	47
Figure 12: Exécution des migrations dans Laravel	48
Figure 13: Extrait de code du composant Livewire de réservation.....	49
Figure 14: Extrait de code d'un contrôleur Laravel pour la gestion des maisons.....	50
Figure 15: Page d'accueil de l'application Ghetto	51
Figure 16: Page d'inscription utilisateur	52
Figure 17: Interface de réservation d'une maison d'hôte.....	53
Figure 18: Tableau de bord du Super Administrateur.....	54
Figure 19: Interface utilisateur de la page des favoris	55
Figure 20: Tableau de bord Propriétaire de maison d'hôte	55
Figure 21: Interface utilisateur de la page d'historique des réservations.....	56
Figure 22: Interface utilisateur des conditions générales d'utilisation pour les hôtes.....	56
Figure 23: Interface de la page de détails d'une propriété	57
Figure 24: Interface utilisateur de consultation des propriétés	57
Figure 25: Interface de consultation des avis d'une propriété	58
Figure 26: Interface de résultat de recherche sans propriété correspondante.....	59
Figure 27: Interface de gestion des utilisateurs par l'administrateur	59
Figure 28: Interface de gestion des catégories par l'administrateur.....	60
Figure 29: Interface de modération des nouvelles propriétés par l'administrateur	61
Figure 30: Interface de gestion des réservations par le propriétaire	62
Figure 31: Interface de la page FAQ.....	62

LISTE DES TABLEAUX

Tableau 1: organisation du système de formation pédagogique de l'école.....	16
Tableau 2: Mention et parcours au sein de l'ENI	16
Tableau 3: Liste des formations existantes à l'ENI.....	17
Tableau 4: Débouchés éventuels des jeunes diplômés.....	19
Tableau 5 : Liste des personnes impliquées sur le projet.....	25
Tableau 6 : Liste des matériels utilisés	25
Tableau 7 : Les logiciels utilisés	26
Tableau 8 : Comparaison entre merise et uml.....	29
Tableau 9 : Comparaison entre laravel et symfony.....	30
Tableau 10: Comparaison entre les SGBD MySQL, Microsoft SQL Server et SQL Lite.....	31
Tableau 11: Comparaison des outils de modélisation Draw, Win ' design et Power AMC	31
Tableau 12: Dictionnaire de données.....	33
Tableau 13: Synthèse des rôles	55

LISTE DES ABREVIATIONS

API	Application Programming Interface (Interface de programmation d'application)
ASI	Audit des Systèmes d'Information
ASR	Administration des Systèmes et Réseaux
CNH	Commission Nationale d'Habilitation
CRM	Customer Relationship Management (Gestion de la relation client)
CSS	Cascading Style Sheets (Feuilles de style en cascade)
DFD	Diagramme de Flux de Données
ENI	École Nationale d'Informatique
ERP	Enterprise Resource Planning (Progiciel de gestion intégré)
FAQ	Foire Aux Questions
FK	Foreign Key (Clé étrangère)
GB	Génie logiciel et Base de Données
GID	Gouvernance et Ingénierie de Données
HDR	Habilitation à Diriger des Recherches
HTML	HyperText Markup Language
IA	Intelligence Artificielle
IDE	Integrated Development Environment (Environnement de développement intégré)
IG	Informatique Générale
JS	JavaScript
LMD	Licence – Master – Doctorat
MCD	Modèle Conceptuel des Données
MCT	Modèle Conceptuel des Traitements
MDi	Métiers du Digital
MLD	Modèle Logique des Données
MOT	Modèle Organisationnel des Traitements

OCC	Objets connectés et Cybersécurité
ORM	Object-Relational Mapping
PHP	Hypertext Preprocessor
PME	Petites et Moyennes Entreprises
REST	Representational State Transfer
RG	Règle de Gestion
RH	Ressources Humaines
SGBD	Système de Gestion de Base de Donnée
TIC	Technologies de l'Information et de la Communication
UML	Unified Modeling Language
UPST	Université Paul Sabatier de Toulouse

INTRODUCTION GENERALE

Dans le cadre de la formation en Informatique générale, un stage a été réalisé portant sur le développement de la plateforme web « **Ghetto** », conçue pour centraliser la gestion des réservations destinées aux voyageurs et aux propriétaires. Cette plateforme a pour objectif de faciliter la communication entre les utilisateurs et d'améliorer l'organisation des réservations, en proposant un système simple, sécurisé et efficace.

La réalisation de ce projet a permis de mettre en pratique les connaissances théoriques acquises au cours de la formation, tout en développant de nouvelles compétences techniques, notamment en conception et développement d'applications web, en gestion de bases de données, en intégration d'interfaces utilisateurs et en. Par sécurité des systèmes ailleurs, des compétences professionnelles essentielles ont été renforcées, telles que le travail en autonomie, la gestion du temps, la résolution de problèmes complexes et la communication avec les utilisateurs ainsi qu'avec les différentes parties prenantes du projet.

Ce stage a également offert une compréhension concrète des exigences du monde professionnel, contribuant au renforcement de la capacité à travailler sur des projets réels et à améliorer la méthodologie de travail. Cette expérience constitue une étape déterminante dans le parcours de formation, préparant efficacement à relever les futurs défis professionnels dans le domaine de l'informatique.

PARTIE I – PRESENTATIONS

Chapitre 1 : PRESENTATION DE L'ECOLE NATIONALE D'INFORMATIQUE

L'École Nationale d'Informatique (ENI) est reconnue comme un établissement de référence à Madagascar dans le domaine de la formation en informatique.

1.1. Information d'ordre générale

L'Ecole Nationale d'Informatique, en abrégé ENI, est un établissement d'enseignement supérieur rattaché académiquement et administrativement à l'Université de Fianarantsoa. Le siège de l'Ecole se trouve à Tanambao-Antaninarenina à Fianarantsoa. L'adresse pour la prise de contact avec l'Ecole est la suivante : Ecole Nationale d'Informatique (ENI) Tanambao, Fianarantsoa. Le numéro de sa boîte postale est 1487 avec le code postal 301. Téléphone : 038 96 566 96 ou 034 76 591 95. Son adresse électronique est la suivante : **eni@eni.mg**. Il dispose également d'un site web : **www.eni.mg**

1.2. Missions et historiques

L'ENI se positionne sur l'échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd'hui comme la vitrine et la pépinière des élites informaticiennes du pays.

De façon formelle, l'ENI était créée par le décret N° 83- 185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d'Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L'ENI a pour conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

- En fournissant à des étudiants des connaissances de base en informatique ;
- En leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises ;
- En initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l'Information et de la Communication (TIC) ;

La filière de formation d'Analystes Programmeurs a été mise en place à l'Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d'ingénieurs a été ouverte à l'Ecole en 1986.

Une formation de troisième cycle a été ouverte à l'Ecole a été ouverte à l'Ecole depuis l'année 2003 – 2004 grâce à la coopération académique et scientifique entre l'Université de Fianarantsoa pour le compte de l'ENI et l'Université Paul Sabatier de Toulouse (UPST).

Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l'Informatique, et notamment pour préparer la relève des Enseignants-Chercheurs qui étaient en poste.

Pendant l'année 2007-2008, la formation en vue de l'obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l'ENI avec les deux parcours de formation :

- Génie Logiciel et base de Données.
- Administration des Système et réseaux.

La mise en place à l'Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD).

En vue de surmonter les difficultés de limitation de l'effectif des étudiants accueillis à l'Ecole, notamment à cause du manque d'infrastructures, un système de « Formation Hybride » a été mise en place à partir de l'année 2010. Il s'agit en effet d'un système de formation semi présentielle et à distance avec l'utilisation de la visioconférence pour la formation à distance. Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu'Université de Toliara. Cette formation est à l'origine du parcours Informatique Générale.

En 2023, la mention Intelligence Artificielle (IA) a été ouvert au sein de l'Ecole pour répondre les besoins des entreprises. La formation est destinée aux étudiants titulaires du diplôme de licence (Bac +3) en Mathématiques ou en Statistiques ou en Informatique, etc. La mention IA comporte deux parcours :

- Gouvernance et Ingénierie de Données (GID),
- Objets connectés et Cybersécurité (OCC).

Le principe de l'enseignement pour le parcours GID offre aux l'étudiants des compétences scientifiques et techniques spécialisées en Science de données. Pour le parcours OCC, les étudiants octroient la double spécialité premièrement en internet des objets et deuxièmement en cybersécurité. La formation de master est axée sur l'ensemble d'applications de l'Intelligence Artificielle.

Depuis l'année universitaire 2024-2025, la mention Expertise Digitale est officiellement ouverte au grand public. Cette mention propose deux (2) parcours distincts :

- Métiers du Digital (MDi) – disponible aux niveaux Licence et Master. Ce parcours est destiné aux étudiants titulaires du diplôme de Baccalauréat, toutes séries confondues. Il est élaboré pour équiper les étudiants des compétences essentielles dans les domaines du marketing digital, de la communication digitale, et de la gestion de projets digitaux.
- Audit des Systèmes d'Information (ASI) – proposé au niveau Master uniquement. Ce parcours s'adresse aux étudiants titulaires d'une licence (Bac +3) en Mathématiques, en Statistiques ou en Informatique. Il est structuré pour approfondir les connaissances théoriques et pratiques nécessaires à l'audit efficace des systèmes d'information.

Le système d'enseignement adopté pour cette mention est un système en ligne. Ainsi, les cours et ressources pédagogiques sont accessibles en permanence via la plateforme numérique Moodle. Par ailleurs, des regroupements en présentiel ou en ligne seront organisés périodiquement, conformément au planning défini par les responsables pédagogiques.

1.3. Organigramme institutionnel

L'organigramme de l'Ecole est inspiré des dispositions du décret N° 83-185 du 24 Mai 1983. L'ENI est administrée par un Conseil d'Ecole, et dirigée par un directeur nommé par un décret adopté en Conseil des Ministres. Le Collège des enseignants regroupant tous les enseignants-chercheurs permanents de l'Ecole est chargé de résoudre les problèmes liés à l'organisation pédagogique des enseignements. Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l'établissement, en tenant compte notamment de l'évolution du marché de travail et de l'adéquation des formations dispensées par rapport aux besoins des entreprises.

La figure 1 représente l'organigramme actuel de l'Ecole Nationale d'Informatique.

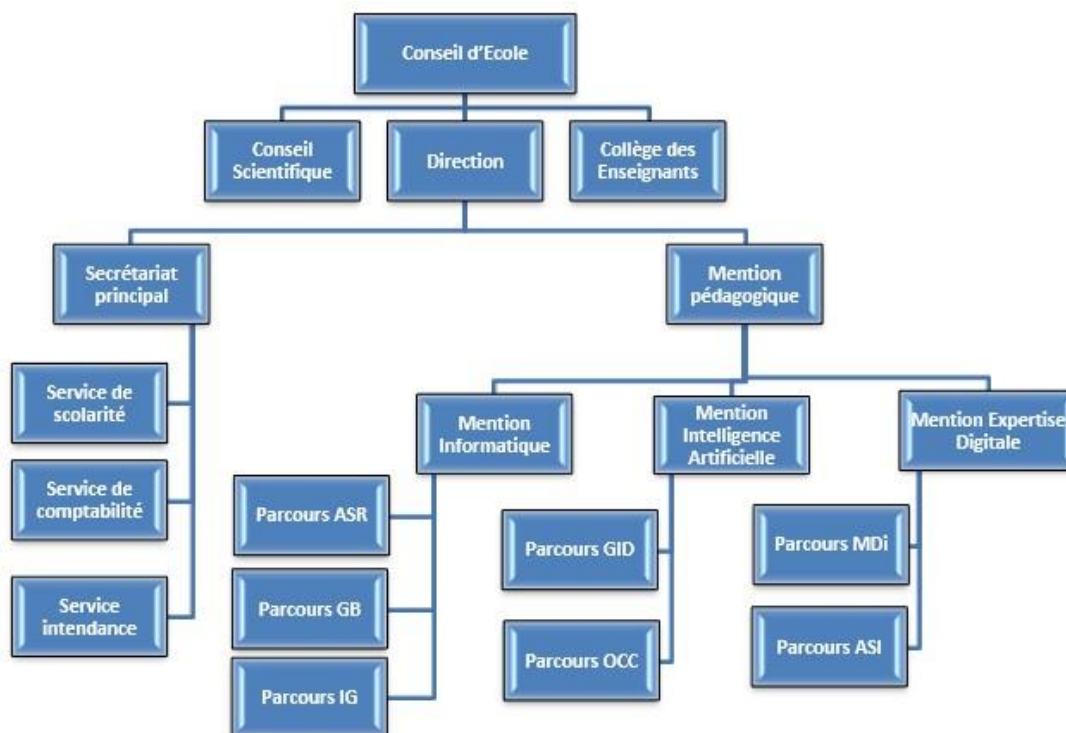


FIGURE 1: ORGANIGRAMME DE L'ECOLE NATIONALE D'INFORMATIQUE

1.4. Domaine de spécialisation

Les activités de formation et de recherche organisées à l'ENI portent sur les domaines suivants :

- Génie logiciel et Base de Données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale ;
- Modélisation informatique et mathématique des Systèmes complexes ;

- Intelligence artificielle.

Le tableau 1 décrit l'organisation du système de formation pédagogique de l'Ecole.

TABEAU 1: ORGANISATION DU SYSTEME DE FORMATION PEDAGOGIQUE DE L'ECOLE

Formation Théorique	Formation Pratique
<ul style="list-style-type: none"> ➤ Enseignement théorique ➤ Travaux dirigés ➤ Travaux pratiques ➤ Conférences 	<ul style="list-style-type: none"> ➤ Etude de cas ➤ Travaux de réalisation ➤ Projets/ Projets tutorés ➤ Voyages d'Etudes ➤ Stages en entreprise

1.5. Architecture des formations pédagogiques

Le recrutement des étudiants à l'ENI se fait uniquement par voie de concours d'envergure nationale en première année. Les offres de formation organisées à l'Ecole ont été validées par la Commission Nationale d'Habilitation (CNH). Au sein de l'ENI, il existe deux mentions et cinq parcours. Le tableau 2 récapitule les mentions et les parcours au sein de l'Ecole :

TABEAU 2: MENTION ET PARCOURS AU SEIN DE L'ENI

Mention	Parcours
Informatique	Génie logiciel et Base de Données (GB)
	Administration des Systèmes et Réseaux (ASR)
	Informatique Générale (IG)
Intelligence Artificielle	Gouvernance et Ingénierie de Données (GID)
	Objets Connectés et Cyber sécurités (OCC)
Expertise Digitale	Métiers du Digital (MDi)
	Audit des Systèmes d'Information (ASI)

La figure 2 représente l'architecture des études correspondant au système LMD.

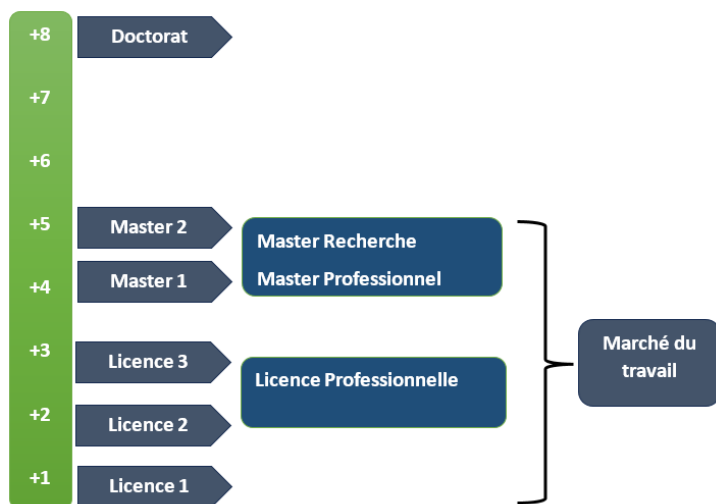


FIGURE 2: ARCHITECTURE DES ETUDES CORRESPONDANT AU SYSTEME LMD

La licence peut avoir une vocation générale ou professionnelle. Le master peut avoir une vocation professionnelle ou de recherche. L'accès en première année de MASTER se fait automatiquement pour les étudiants de l'Ecole qui ont obtenu le diplôme de Licence Professionnelle.

Le tableau 3 illustre la liste des formations existantes à l'ENI.

TABLEAU 3: LISTE DES FORMATIONS EXISTANTES A L'ENI

	FORMATION	
	LICENCE PROFESSIONNELLE	MASTER
Condition admission	Par voie de concours	Par voie de concours pour la mention IA
Condition d'accès	Bac de série C, D ou Technique	Être titulaire de licence professionnelle
Durée de Formation	3 ans	2 ans
Diplôme délivré	Diplôme de Licence Professionnelle	Diplôme de Master Professionnel Diplôme de Master Recherche

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s'inscrire directement dans une Ecole Doctorale.

Les étudiants diplômés de l'Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (Canada, Suisse, France, ...)

1.6. Relation de l'ENI avec les organismes externes

Les stages effectués chaque année par les étudiants mettent l'Ecole en rapport permanent avec plus de 400 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux. L'Ecole

dispose ainsi d'un réseau d'entreprises, de sociétés et d'organismes publics et privés qui sont des partenaires par l'accueil en stage de ses étudiants, et éventuellement pour le recrutement après l'obtention des diplômes par ces derniers. Les compétences que l'Ecole cherche à développer chez ses étudiants sont l'adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l'expérimentation et l'innovation.

En effet, la vocation de l'ENI est de former des licenciés et des ingénieurs de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d'évoluer professionnellement dans des secteurs d'activité variés intégrant l'informatique. Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l'Ecole et les besoins évolutifs du marché de l'emploi.

Parmi les sociétés, les entreprises et les organismes partenaires de l'Ecole, on peut citer : ACCENTURE Mauritius, AKATA Goavana, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie (AUF), AXIAN, B2B, Banque Centrale, , BIANCO, BlueLine, CNaPS, Bureau National de Gestion des Risques et des Catastrophes (BNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, COLAS, Direction Générale des Douanes, DLC, E-Tech Consulting, , FID, FIHARY Soft, FTM, GNOSYS, GENIUS AT WORK, Hello Tana, IBONIA, INGENOSIA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MANAO, MEF, MEN, MESupRES, MFB, , MININTER, Min des Postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SG Madagasikara SMMC, SMMEC, SNEDADRS Antsirabe, Sénat, Société d'Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN'I BETSILEO, WWF, UGD, ARATO, MANAO, MNDPT, NG ACADEMY.NG, Relia, Spoon Consulting ...

1.7. Débouchés professionnels et diplômés

Les formations proposées par l'Ecole permettent aux diplômés d'être immédiatement opérationnels sur le marché du travail avec la connaissance d'un métier complet lié à l'informatique aux TIC.

L'Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante. Les diplômés en LICENCE et en MASTER issus de l'ENI peuvent faire carrière dans différents secteurs.

L'Ecole bénéficie aujourd'hui de 40 années d'expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C'est une Ecole Supérieure de référence en matière informatique.

D'une manière générale, les diplômés de l'ENI n'éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l'ENI recommande à ses diplômés de promouvoir l'entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d'études.

Le tableau 4 représente les débouchés éventuels selon la mention.

TABEAU 4: DEBOUCHES EVENTUELS DES JEUNES DIPLOMES

Mention	Débouchés professionnels
INFORMATIQUE	<p>GB : Développeur logiciel / développeur full-stack, Ingénieur en développement d'applications (desktop, web, mobile), Architecte logiciel, Intégrateur d'applications, Développeur d'API, Administrateur de bases de données (DBA), Ingénieur en conception de bases de données, Développeur SQL / PL-SQL, Spécialiste en migration et optimisation de bases de données, Data engineer (ingénieur en traitement de données), Analyste programmeur, Analyste fonctionnel, Chef de projet informatique, Scrum master / Product owner (en méthodologie Agile), Ingénieur QA (Quality Assurance), Testeur logiciel / testeur automatisé, Spécialiste maintenance applicative, Expert en refactoring et optimisation de code, Analyste de données (Data analyst), Concepteur d'outils décisionnels (BI), Ingénieur en Big Data, Machine Learning Engineer, Consultant fonctionnel ou freelance</p> <p>ASR : Administrateur systèmes et réseaux, Ingénieur systèmes et réseaux, Administrateur systèmes Linux / Windows, Administrateur cloud (AWS, Azure, Google Cloud), Ingénieur virtualisation (VMware, Hyper-V, Proxmox), Administrateur sécurité (Security Administrator), Ingénieur cybersécurité, Analyste SOC (Security Operations Center), Spécialiste firewall / VPN, Responsable sécurité informatique (RSSI – avec expérience), Technicien support informatique (N2/N3), Technicien réseaux et télécoms, Ingénieur exploitation, Administrateur messagerie (Exchange, Microsoft 365), Technicien datacenter, Architecte réseaux, Architecte systèmes, Consultant infrastructure, Chef de projet systèmes et réseaux, Ingénieur DevOps, Administrateur Kubernetes / Docker, Cloud engineer, Spécialiste automatisation (Ansible, Terraform)</p> <p>IG : GB et SR en même temps</p>
INTELLIGENCE ARTIFICIELLE	<p>GID : Développeur IA, Data Scientist, Entrepreneur IA, Consultant IA, Responsable marketing IA, Business Developer IA, Product Owner, Consultant en cybersécurité, Directeur de l'Innovation ; OCC : Analyste en cybersécurité, Ingénieur en cybersécurité, Spécialiste en réponse aux incidents, Architecte de sécurité, Consultant en cybersécurité, Auditeur en cybersécurité, Expert en sécurité des données, Responsable de la sécurité des informations, Chercheur en cybersécurité, Formateur en cybersécurité, Ingénieur IoT, Développeur IoT, Architecte IoT, Spécialiste en sécurité IoT, Analyste de données</p>

EXPERTISE DIGITALE	<p>MDi : Spécialiste en Marketing Digital, Gestionnaire de Contenu Web, Analyste UX/UI, Product Owner, Chef de Projet Digital, Stratège en Médias Sociaux, Analyste de Performance Digitale, Chef de Projet Digital, Stratège en Médias Sociaux, Concepteur-Rédacteur Web Analyste de Performance Digitale, Expert en Commerce Électronique, Expert en Commerce Électronique, Spécialiste SEO/SEM, Manager de Communauté (Community Manager), Conseiller en Stratégie Digitale, Responsable CRM (Customer Relationship Management), Spécialiste en Analytique et Intelligence d’Affaires (Business Intelligence), Consultant en Transformation Digitale, Responsable des Partenariats Digitaux, Spécialiste en Cybersécurité Marketing, Directeur de l’Innovation Digitale, Manager de l'E-réputation</p> <p>ASI : Auditeur des Systèmes d'Information, Consultant en Sécurité des Systèmes d'Information, Analyste de Risques IT, Chef de Projet SI, Responsable de la Conformité IT, Spécialiste en Cyber Forensics, Directeur de l'Audit et de la Sécurité des Systèmes d'Information, Expert en Gouvernance IT, Formateur ou Éducateur en Sécurité des Chercheur en Sécurité IT</p>
-----------------------	---

1.8. Ressources humaines

Les ressources humaines sont citées ci-dessous selon leurs responsabilités :

- Directeur de l’Ecole : Monsieur MAHATODY Thomas, Professeur
- Responsable de la Mention « Informatique » : Monsieur RALAIVAO Jean Christian, Maître de Conférences
- Responsable de la Mention « Intelligence Artificielle » : Monsieur DIMBISOA William Germain, Maître de Conférences

L’ENI compte douze (12) enseignants permanents dont un (01) Professeur Titulaire, deux (02) Professeurs, Sept (07) Maîtres de Conférences, Deux (02) Assistants d’Enseignement Supérieur et de Recherche, dix (10) enseignants vacataires, quarante un (41) personnel administratif.

Chapitre 2 : PRESENTATION DE MASOVIA MADAGASCAR

2.1. Introduction

Masovia Madagascar est une entreprise œuvrant dans le domaine des technologies de l'information et de la communication. Fondée en 2023, elle s'inscrit dans une dynamique de transformation numérique visant à accompagner les organisations publiques et privées dans l'amélioration de leurs performances à travers des solutions informatiques adaptées.

Ce chapitre a pour objectif de présenter l'historique de Masovia Madagascar, depuis sa création jusqu'à son évolution récente, en mettant en évidence le contexte de sa naissance, les différentes étapes de son développement, les projets réalisés ainsi que sa vision stratégique.

2.2. Évolution de la structure organisationnelle

Depuis sa création, Masovia Madagascar a connu une croissance progressive de ses ressources humaines.

- **En 2024**, l'entreprise comptait six (6) membres, répartis comme suit : un commercial, deux développeurs backend, deux développeurs frontend, ainsi que le Directeur Général, également fondateur.
- **En 2025**, l'effectif est passé à onze (11) collaborateurs, comprenant : un commercial, deux développeurs backend, deux développeurs frontend, le Directeur Général et fondateur, quatre stagiaires et un comptable.

Cette évolution témoigne de la structuration progressive de l'entreprise et de l'augmentation de ses activités.

2.3. Année 2023 : Création et mise en place de l'entreprise

2.3.1 Contexte de création

L'année 2023 marque la création officielle de Masovia Madagascar. Son fondateur, ingénieur en informatique, disposait de plus de quatre années d'expérience professionnelle dans le développement web et mobile, ainsi que dans l'administration des systèmes et réseaux informatiques. Cette expérience lui a permis d'identifier les besoins croissants des entreprises malgaches en matière de digitalisation.

Le contexte national présentait une situation contrastée : une jeunesse fortement connectée et familière des outils numériques, face à des entreprises encore peu engagées dans la transformation digitale. Ce constat a motivé la création de Masovia Madagascar, avec pour objectif de proposer des solutions informatiques accessibles, fiables et adaptées au contexte local.

2.3.2 Premières activités

Au cours de l'année 2023, l'entreprise a concentré ses efforts sur la mise en place de ses bases fondamentales, notamment :

- La création officielle de la structure juridique ;
- La définition des principaux services proposés, à savoir le développement web, le développement d'applications mobiles et l'administration des systèmes et réseaux ;
- La constitution d'une première équipe partageant une vision commune.

Les premiers projets réalisés concernaient principalement la création de sites vitrines, le développement de solutions de gestion simplifiées pour les petites et moyennes entreprises, ainsi que des prestations de maintenance informatique.

2.3.3 Difficultés rencontrées

Comme toute jeune entreprise, Masovia Madagascar a été confrontée à plusieurs difficultés, notamment :

- Le manque de notoriété auprès des clients potentiels ;
- Des contraintes budgétaires limitant les investissements initiaux ;
- Une forte concurrence sur le marché des services numériques.

Malgré ces obstacles, l'engagement et le professionnalisme de l'équipe ont permis à l'entreprise de surmonter cette phase initiale.

2.4. Année 2024 : Expansion et réalisation de projets majeurs

L'année 2024 a été marquée par une phase d'expansion et par la réalisation de projets plus structurés. Masovia Madagascar a renforcé sa crédibilité et élargi son portefeuille clients, tant au niveau national qu'international.

2.4.1 Présentation des projets réalisés

a- Plateforme e-commerce de matériel de boxe

Ce projet visait à répondre au manque de visibilité en ligne des acteurs du secteur des arts martiaux à Madagascar. La solution mise en place consistait en une plateforme e-commerce intégrant la gestion des stocks, des variantes de produits et des promotions. Les résultats ont permis une augmentation de la clientèle et une ouverture vers le marché international.

b- Application web de mise en relation de masseurs en France

Le projet avait pour objectif de centraliser les offres des masseurs indépendants. La plateforme développée comprend des profils professionnels, un système de réservation en ligne, la géolocalisation et une messagerie intégrée. Cette solution a favorisé la création d'une communauté active d'utilisateurs.

c- Application de gestion scolaire

Cette application répondait aux difficultés rencontrées par les établissements scolaires dans la gestion administrative et pédagogique. Elle permet la gestion des élèves, le suivi pédagogique, l'accès des parents aux informations scolaires et la planification des emplois du temps. Les résultats observés montrent une amélioration notable de l'organisation interne.

2.5. Année 2025 : Consolidation et diversification des activités

En 2025, Masovia Madagascar est entrée dans une phase de consolidation de ses acquis et de diversification de ses services.

2.5.1 Axes stratégiques

Les principales orientations stratégiques de l'entreprise sont les suivantes :

- Renforcement de la relation client et fidélisation des partenaires ;
- Diversification des services, notamment dans le cloud computing, l'hébergement sécurisé et la cybersécurité ;
- Mise en place de programmes de formation et de stages destinés aux étudiants malgaches.

2.5.2 Nouveaux domaines d'intervention

L'entreprise s'est orientée vers des secteurs jugés stratégiques, tels que la santé, l'éducation et la gestion d'entreprise, à travers le développement d'applications médicales, de plateformes e-learning et de solutions CRM et ERP adaptées aux PME.

2.6. Vision et perspectives d'avenir

Masovia Madagascar ambitionne de devenir un acteur majeur de la transformation digitale en Afrique. Sa vision repose sur l'innovation continue, la qualité des solutions proposées et la proximité avec les clients, afin d'accompagner durablement les organisations dans leur modernisation.

2.7. Conclusion

L'évolution de Masovia Madagascar illustre le parcours d'une jeune entreprise technologique ayant su s'adapter aux réalités du marché et répondre aux besoins croissants en matière de digitalisation. En quelques années, elle est passée d'un projet entrepreneurial à une structure organisée, capable de mener des projets d'envergure et de contribuer au développement numérique à Madagascar.

Chapitre 3 : DESCRIPTION DU PROJET

3.1. Formulation

Le projet consiste à concevoir et développer une plateforme web de réservation de maisons d'hôtes, permettant aux utilisateurs de rechercher et réserver des hébergements et aux propriétaires de gérer leurs annonces.

3.2. Objectif et besoins de l'utilisateur

Pour la concrétisation du projet **Ghetto**, il est indispensable de définir clairement les objectifs généraux de l'application ainsi que les besoins spécifiques des différents types d'utilisateurs. Cette étape permet d'assurer que la solution développée répond efficacement aux attentes fonctionnelles et organisationnelles du système de réservation de maisons d'hôtes.

3.2.1. Objectifs

L'application web **Ghetto** a pour objectif principal de mettre en relation les voyageurs et les propriétaires de maisons d'hôtes à travers une plateforme numérique moderne, intuitive et sécurisée.

Elle vise notamment à :

- Faciliter la recherche et la réservation de maisons d'hôtes ;
- Offrir aux hôtes un espace dédié pour gérer leurs propriétés et leurs réservations ;
- Permettre à l'administrateur principal de superviser, modérer et sécuriser l'ensemble de la plateforme ;
- Améliorer la visibilité des maisons d'hôtes locales grâce à une solution digitale centralisée.

3.2.2. Besoins des utilisateurs

➤ Le futur système doit permettre au Super Administrateur de :

- Créer et gérer les comptes administrateurs et hôtes ;
- Valider ou refuser les nouvelles maisons d'hôtes proposées par les hôtes (modération) ;
- Gérer les catégories de propriétés ;
- Superviser l'ensemble des utilisateurs de la plateforme ;
- Modérer les contenus (propriétés, avis, descriptions) afin de garantir la qualité et la conformité ;
- Consulter les statistiques globales (nombre de propriétés, réservations, utilisateurs, avis) ;
- Assurer le bon fonctionnement général et la sécurité de l'application.

➤ Le futur système doit permettre aux Hôtes (propriétaires) de :

- Créer et gérer leur compte hôte ;
- Ajouter, modifier et supprimer leurs maisons d'hôtes ;
- Renseigner les informations détaillées des propriétés (description, images, prix, disponibilité, localisation, catégories) ;
- Consulter et gérer les réservations effectuées par les utilisateurs ;
- Accepter ou refuser les demandes de réservation ;

- Consulter l'historique des réservations de leurs propriétés ;
- Accéder aux conditions générales d'utilisation spécifiques aux hôtes.

➤ Le futur système doit permettre aux Utilisateurs (clients) de :

- Créer un compte utilisateur et s'authentifier sur la plateforme ;
- Rechercher des maisons d'hôtes selon différents critères (localisation, prix, catégorie, disponibilité) ;
- Consulter les détails complets d'une propriété ;
- Effectuer une réservation en ligne ;
- Consulter l'historique de leurs réservations ;
- Ajouter des maisons d'hôtes à leurs favoris ;
- Laisser des avis et des notes sur les propriétés réservées ;
- Consulter la foire aux questions (FAQ) et les conditions générales d'utilisation.

3.3. Moyens nécessaires à la réalisation du projet

Afin d'assurer la réussite de ce projet de **conception et de développement d'une application de réservation de maisons d'hôtes**, diverses ressources humaines, matérielles et logicielles ont été mobilisées. Les tableaux présentés ci-après décrivent de manière détaillée l'ensemble des moyens nécessaires à la réalisation, au déploiement et au bon fonctionnement de l'application.

TABEAU 5 : LISTE DES PERSONNES IMPLIQUEES SUR LE PROJET

Nom et prénoms	Sexe	Rôle dans le projet	Titre
Monsieur RANDRIANOMENJANAHARY Lala Ferdinand	Homme	Encadreur pédagogique	Enseignant à L'ENI
Monsieur NIHANARISON Franck Ulsy	Homme	Encadreur professionnel	DG de MASOVIA MADAGASCAR

TABEAU 6 : LISTE DES MATERIELS UTILISES

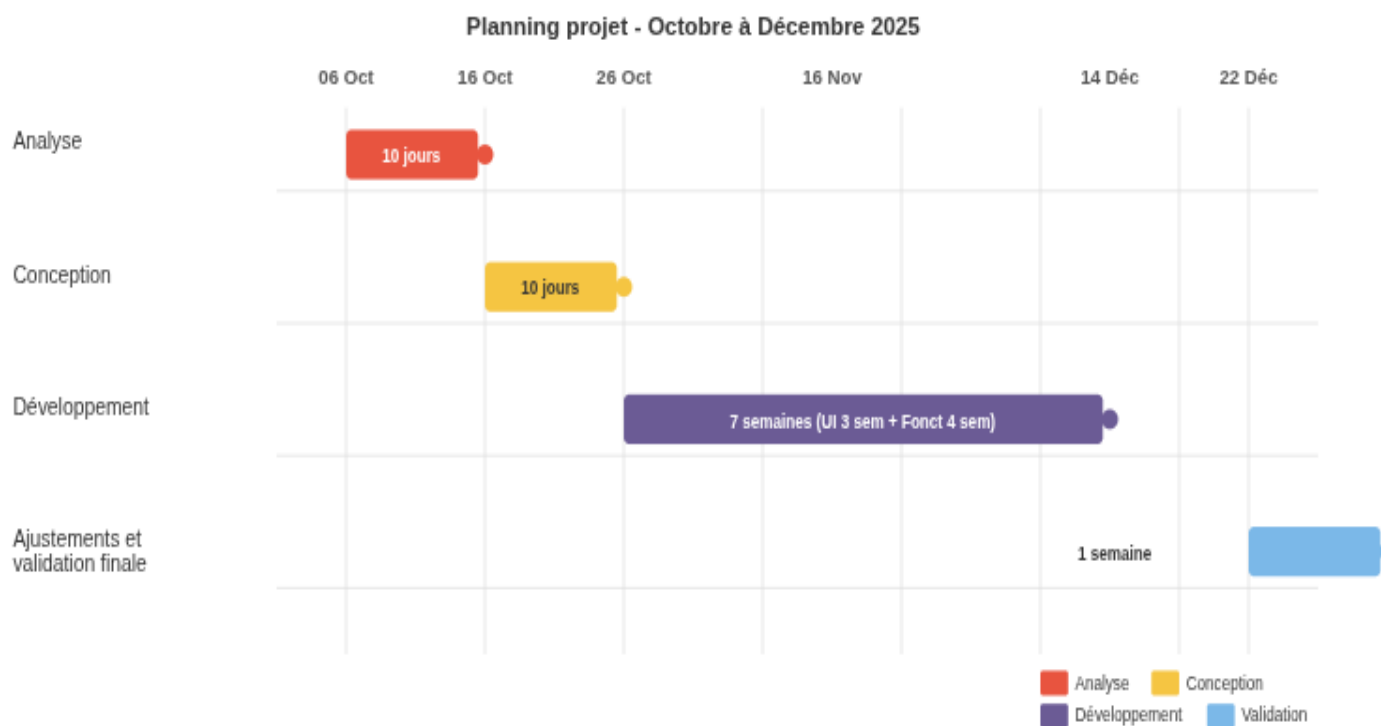
Nom	Caractéristiques
PC portable	12 Go RAM, 500 Go HDD, SE Linux Manjaro GNOME, Serveur Simafri.
Connexion internet stable	Wifiber Go orange

TABEAU 7 : LES LOGICIELS UTILISES

Logiciel	Version	Langage de programmation	Fabriquant
VS code	1.95.1	PHP, Laravel, Livewire, Tailwind CSS,	Microsoft
XAMPP	1.95.1	SQL	Apache Friends
Git	2.51.0		Linus Torvalds

3.4 Chronogramme de travail

Le tableau ci-dessus présente la planification prévisionnelle des différentes phases du projet, depuis l'analyse jusqu'aux ajustements et à la validation finale. Chaque activité est définie par une date de début et une durée estimée, permettant d'assurer une organisation cohérente et un suivi efficace de l'avancement des travaux

**FIGURE 3: CHRONOGRAMME DE TRAVAIL**

PARTIE II – ANALYSE ET CONCEPTION

Chapitre 4. ANALYSE PREALABLE

4.1 Analyse de l'existant

La présente étude a été menée auprès d'une maison d'hôtes (ou d'un ensemble de maisons d'hôtes) afin d'analyser les pratiques actuelles de gestion des réservations. Cette analyse de l'existant consiste à examiner les méthodes actuellement utilisées pour la prise de réservations, la gestion des disponibilités, ainsi que le suivi des clients. Elle vise à mettre en évidence les limites du système en place, notamment en termes d'efficacité, de fiabilité et d'accessibilité, et à identifier les opportunités d'amélioration pouvant être apportées par la mise en œuvre d'une application informatisée de réservation.

4.1.1 Organisation actuelle

L'organisation actuelle de la gestion des réservations dans la maison d'hôtes repose sur un processus principalement manuel. Les demandes de réservation sont généralement effectuées par les clients via des formulaires papier ou des appels téléphoniques, puis traitées par le personnel administratif.

Le personnel concerné comprend les réceptionnistes, les responsables des réservations et, dans certains cas, les gestionnaires des hébergements. Les réceptionnistes sont chargés de l'enregistrement des demandes et de la tenue des registres, tandis que les responsables des réservations assurent la validation et la confirmation des séjours. Les gestionnaires des hébergements participent également au processus pour coordonner les disponibilités et préparer l'accueil des clients.

Cependant, la communication entre ces différents acteurs manque parfois de fluidité, ce qui peut entraîner des erreurs ou des retards dans le traitement des réservations et affecter la satisfaction des clients.

4.1.2 Inventaire des moyens matériels et logiciels

En termes de moyens matériels, la maison d'hôtes dispose d'ordinateurs, d'imprimantes et d'outils bureautiques tels qu'Excel. Il est cependant nécessaire d'évaluer l'état et la performance de ces équipements afin de s'assurer qu'ils répondent aux besoins opérationnels liés à la gestion des réservations.

Pour ce qui est des moyens logiciels, l'évaluation des outils informatiques existants est essentielle pour déterminer leur capacité à gérer efficacement les réservations, les disponibilités et les informations clients, tout en assurant une communication fluide entre les différents services impliqués dans le processus.

4.2 Critique de l'existant

L'analyse de l'existant a permis d'identifier plusieurs **points forts** et **points faibles** dans le système actuel de gestion des réservations au sein de la maison d'hôtes.

Points forts :

- **Engagement du personnel :** Le personnel en charge des réservations et de l'accueil est motivé et impliqué dans le traitement des demandes des clients, ce qui contribue à une certaine efficacité dans la gestion des réservations.

- **Respect des procédures :** Les méthodes actuellement utilisées respectent les règles internes de la maison d'hôtes pour l'enregistrement des clients et la gestion des disponibilités, ce qui assure une certaine fiabilité du service.

Points faibles :

- **Manque d'automatisation :** Le processus de gestion des réservations reste principalement manuel, ce qui peut générer des retards, des erreurs ou des doubles réservations.
- **Communication fragmentée :** La communication entre les différents services (réception, gestion des hébergements, service administratif) est parfois peu fluide, ce qui complique le suivi des réservations et peut nuire à l'expérience client.

4.3 Conception avant-projet

Cette section présente les propositions de solutions visant à améliorer le système actuel de **gestion des réservations** dans la maison d'hôtes, ainsi que les méthodes de conception et les outils envisagés pour la mise en œuvre de ce projet.

4.3.1 Proposition des solutions

Pour optimiser la gestion des réservations, plusieurs solutions sont envisagées :

- **Plateforme intégrée :** Développer une application unique qui centralise toutes les étapes du processus de réservation, depuis la demande initiale jusqu'à la confirmation et le suivi des séjours. Cette approche permet de fluidifier la communication entre les différents services et de réduire les risques d'erreurs.
- **Formation du personnel :** Organiser des sessions de formation pour le personnel en charge des réservations et de l'accueil afin d'améliorer leur maîtrise du système et de renforcer la coordination entre les équipes.

4.3.2. Méthodes de conception et outils utilisés

Les méthodes de modélisation sont utilisées afin de donner une description la plus approximative du contenu fonctionnel d'un système d'information. Plusieurs méthodes existent, mais les plus reconnues sont la méthode MERISE et les méthodes orientées objet comme UML.

TABEAU 8 : COMPARAISON ENTRE MERISE ET UML

	AVANTAGES	INCONVENIENTS
MERISE	<ul style="list-style-type: none"> - Rigueur et structuration : Approche structurée et reconnue pour la conception de systèmes d'information - Séparation claire des données et des traitements - Vision globale du système - Méthode éprouvée depuis les années 1970 	<ul style="list-style-type: none"> - Rigidité face aux changements - Lourdeur de la documentation - Non orientée objet - Principalement connue dans les pays francophones

	- Particulièrement adaptée aux bases de données relationnelles	
UML	<ul style="list-style-type: none"> - Standard international reconnu - Approche orientée objet moderne - Grande flexibilité - Multiples diagrammes pour différents points de vue - Adapté aux méthodes agiles 	<ul style="list-style-type: none"> - Courbe d'apprentissage importante - Peut être complexe pour des projets simples - Nécessite une bonne maîtrise de l'orienté objet - Risque de sur-modélisation

Le choix s'est porté sur la méthode MERISE, car elle est bien adaptée à la gestion des projets internes aux organisations, en se limitant à un domaine spécifique. MERISE est une méthode de développement de projets informatiques de gestion. Elle convient à toutes les applications, qu'elles soient sur micro-ordinateurs, mini-ordinateurs ou grands systèmes informatiques.

Passons maintenant aux différents outils qui peuvent être utilisés pour réaliser ce projet. Bien qu'il en existe plusieurs, nous nous concentrerons uniquement sur les plus connus dans ce domaine et ferons une comparaison entre eux. Examinons d'abord les frameworks pouvant être utilisés pour développer l'application.

TABLEAU 9 : COMPARAISON ENTRE LARAVEL ET SYMFONY

Framework	Avantages	Inconvénients
Laravel	<ul style="list-style-type: none"> - Syntaxe élégante et expressive - Eloquent ORM puissant - Migration et seeding intégrés - Système d'authentification complet - Documentation excellente - Large communauté - Écosystème riche (Passport, Sanctum) - Idéal pour les API REST 	<ul style="list-style-type: none"> - Performances légèrement inférieures à Symfony - Mise à jour majeure annuelle - Peut être "magique" (convention over configuration)
Symfony	<ul style="list-style-type: none"> - Très performant - Modularité par composants - Stabilité à long terme - Documentation professionnelle - Utilisé par de grandes entreprises 	<ul style="list-style-type: none"> - Courbe d'apprentissage plus raide - Configuration plus complexe - Verbose du code - Délai de mise sur le marché plus long

Choix retenu : On a choisi **Laravel** comme framework Backend. Ce choix se justifie par :

- Sa syntaxe claire et expressive, idéale pour un développement rapide
- Son ORM Eloquent qui facilite les interactions avec MySQL
- Son système d'authentification JWT intégré (Laravel Passport/Sanctum)
- Son excellent support pour les API RESTful
- Sa documentation complète et sa large communauté
- Sa parfaite adéquation avec les besoins du projet

Autour des outils de gestion des bases de données, nous pouvons distinguer trois SGBD principaux : Microsoft SQL Server, MySQL et SQLite.

TABEAU 10: COMPARAISON ENTRE LES SGBD MYSQL, MICROSOFT SQL SERVER ET SQL LITE

	Avantages	Inconvénients
Microsoft SQL Server	Augmente la sécurité des données. Facilité de configuration. Stockage.	Les fonctionnalités plus avancées sont payantes. Restriction matérielle.
MySQL	Multi-utilisateur SGBD Open Source Pas besoin d'avoir une grande RAM pour fonctionner Outils d'administration très complets	Support technique médiocre. Difficile de postuler avec une grande base de données
SQL Lite	Léger Meilleures performances Fiable Portable Réduit la complexité	Limité à un accès mono-utilisateur La quantité de mémoire requise augmente à mesure que la base de données s'agrandit

Le choix de MySQL s'impose pour sa capacité à gérer efficacement les données et sa flexibilité à répondre aux besoins de multiples applications. Il permet de structurer et de manipuler les données de manière optimale. De plus, les systèmes de gestion de bases de données tels que MySQL, Microsoft SQL Server et SQLite offrent des avantages spécifiques, chacun adapté à différents environnements et exigences de projet.

Passons maintenant aux outils de modélisation. Dans ce cas, la comparaison se focalisera entre Draw.io, Win 'design et Power AMC Designer

TABEAU 11: COMPARAISON DES OUTILS DE MODELISATION DRAW, WIN ' DESIGN ET POWER AMC

	Avantages	Inconvénients
Draw.io (diagrams.net)	Permet une modélisation structurée des processus, des données et des architectures, offrant une vue claire et organisée de la structure de l'entreprise.	Fonctionnalités moins avancées que certains logiciels payants pour la modélisation complexe des données. Pas de support client dédié ;

	<p>Gratuit et accessible en ligne, compatible avec tous les systèmes d'exploitation. Interface intuitive et simple à utiliser, même pour les débutants.</p> <p>Collaboration possible via stockage cloud (Google Drive, OneDrive, etc.), permettant à plusieurs utilisateurs de travailler simultanément sur les diagrammes.</p>	<p>dépend des ressources en ligne et tutoriels. Peut nécessiter un peu de temps pour gérer des projets très larges et complexes.</p>
Win ' design	<p>Permet une modélisation structurée des processus, des données et des architectures, offrant une vue claire et organisée de la structure de l'entreprise. Cela facilite la compréhension et la communication au sein de l'organisation</p> <p>Il offre des fonctionnalités de collaboration, permettant à plusieurs membres de l'équipe de travailler sur des modèles simultanément, favorisant ainsi la coopération et la cohérence dans la conception</p>	<p>Comme tout logiciel de conception complexe, Win' Design peut avoir une courbe d'apprentissage abrupte. La prise en main du logiciel peut nécessiter du temps et de la formation, ce qui peut être un défi pour les utilisateurs novices.</p> <p>Win' Design peut ne pas être compatible avec certains systèmes d'exploitation ou logiciels, ce qui peut limiter sa portée et sa flexibilité</p>
Power AMC Designer	<p>Le logiciel offre des fonctionnalités avancées pour la modélisation des données, la gestion de projet et la collaboration entre équipes</p> <p>Propose une interface conviviale qui facilite la création et la modification de modèles de données de processus</p>	<p>Le coût de licence de Power AMC Designer peut être élevé pour les petites entreprises ou les individus</p> <p>Certains utilisateurs ont noté un support client limité de l'éditeur, ce qui peut poser des problèmes en cas de difficultés techniques</p>

Draw.io est idéal pour les étudiants et les projets académiques, car il est **gratuit, multiplateforme et très flexible**. Il couvre la majorité des besoins pour la modélisation MERISE (MCD, MLD, MCT, MOT), UML, et diagrammes de flux, tout en restant simple à prendre en main.

Chapitre 5. ANALYSE CONCEPTUELLE

Afin de maîtriser l'architecture et la conception de notre système d'information, nous allons explorer les piliers de cette étude : nous commencerons par la Présentation de la méthode Merise, avant d'analyser le Dictionnaire des données, les Règles de gestion, le Diagramme de flux, et enfin la Modélisation des données.

5.1 Présentation de la méthode Merise

La méthode MERISE est la méthodologie retenue pour l'analyse et la conception du futur système d'information. Son choix est justifié par sa capacité à séparer clairement les préoccupations fonctionnelles des préoccupations techniques. La méthode s'articule autour de deux cycles majeurs :

5.1.1 Le Cycle de Vie

Il structure la réalisation du projet en distinguant les niveaux d'abstraction, allant du niveau métier au niveau technique :

- **Niveau Conceptuel** : Modélisation du Quoi (les traitements et les données) indépendamment de toute contrainte d'organisation ou de choix technique. C'est l'objectif du Chapitre 5.
- **Niveau Logique (ou Organisationnel)** : Modélisation du Qui et du Quand (l'organisation, les acteurs, le temps) sans encore se soucier du matériel.
- **Niveau Physique** : Modélisation du Comment (la technologie, le matériel, le logiciel – choix du SGBD et du Framework).

5.1.2 Le Cycle de Décision :

Il garantit la validation du projet à des étapes clés (Étude Préalable, Étude Détaillée, Réalisation) avant de passer à la phase suivante.

5.2 Dictionnaire des données

TABLEAU 12: DICTIONNAIRE DE DONNEES

Nom	Description	Type	Taille	Observation
id	identifiant	N	4	Auto-incrémente
email	e-mail de l'utilisateur.	AN	40	Unique
name	Nom ou nom complet	A	255	
password	Mot de passe haché	AN	255	
phone	Numéro de téléphone	AN	255	
role	Rôle (admin, client, etc.)	A	255	Par défaut 'client'
active	État du compte	N	1	Booléen
can_host	Autorisation à héberger	N	1	Booléen

address	Adresse physique ou complète	AN	255	
category_id	Type de catégorie	N	20	FK
hotel_id	Hôtel rattaché	N	20	FK, Optionnel
owner_id	Propriétaire / Gérant	N	20	FK
description	Détails ou description	AN	Text	
pricePerNight	Prix par nuit	N	20	
guest	Capacité maximale (personnes)	N	10	
room	Nombre de chambres	N	10	
coverImage	Image principale	AN	255	Chemin du fichier
isAvailable	Disponibilité actuelle	N	1	Booléen
property_id	Bien / Propriété concernée	N	20	FK
user_id	Utilisateur concerné	N	20	FK
check_in	Date de début	D		
check_out	Date de fin	D		
status	État ou statut	A	Enum	Défaut 'pending'
total_price	Montant total	N	20	
city	Ville	A	255	
rating	Note attribuée	N	11	
comment	Avis ou commentaire	AN	Text	
data	Données supplémentaires	AN	Text	
read_at	Date de lecture	D		

A : Alpha AN : Alphanumérique N : Numérique D : Date

5.3 Règles de gestion

Les règles de gestion sont des informations qui définissent la suite des opérations effectuées pour passer des données de base au résultat attendu. Elles indiquent les liens existants entre les données, les actions effectuées par les utilisateurs et les contraintes sur le système. Pour le domaine d'étude Ghetto, les règles de gestion sont :

- **RG1** : L'utilisateur crée un compte avec le rôle **Client** par défaut.
- **RG2** : L'utilisateur peut s'authentifier pour accéder aux fonctionnalités selon son rôle.
- **RG3** : Un propriétaire peut ajouter un ou plusieurs biens (propriétés/hôtels) à la plateforme.
- **RG4** : Un client peut rechercher et consulter les biens disponibles.
- **RG5** : Un client peut réserver un ou plusieurs biens, chaque réservation étant associée à un bien et à un client unique.
- **RG6** : Lorsqu'une réservation est effectuée, le propriétaire concerné est notifié et doit valider la réservation.
- **RG7** : Une réservation inclut les dates d'arrivée et de départ, le prix total, et son statut (pending, confirmed, etc.).
- **RG8** : Les clients peuvent laisser un avis et une note sur un bien après leur séjour ; un avis appartient à un utilisateur et à un bien unique.
- **RG9** : Les propriétaires ou gérants d'hôtels peuvent gérer leurs établissements et biens associés.
- **RG10** : Un bien peut appartenir à une catégorie et peut être lié à un hôtel (optionnel).

- **RG11** : Les utilisateurs peuvent ajouter des biens à leurs favoris pour consultation future.
- **RG12** : Le système peut envoyer des notifications aux utilisateurs concernant les réservations, messages ou changements de statut.
- **RG13** : Les informations d'utilisateur, de réservation, de bien et d'hôtel doivent être cohérentes et respecter les contraintes d'intégrité référentielle (FK, unicité, etc.).
- **RG14** : Les données sensibles comme le mot de passe doivent être stockées de manière sécurisée (hachage).

5.4 Diagramme de flux

Pour visualiser la circulation des données et des informations à travers les processus du système, nous avons élaboré le Diagramme de Flux de Données (DFD). Ce diagramme est présenté ci-dessous.

La Figure 4 ci-dessous illustre la représentation graphique du Diagramme de Flux de Données (DFD).

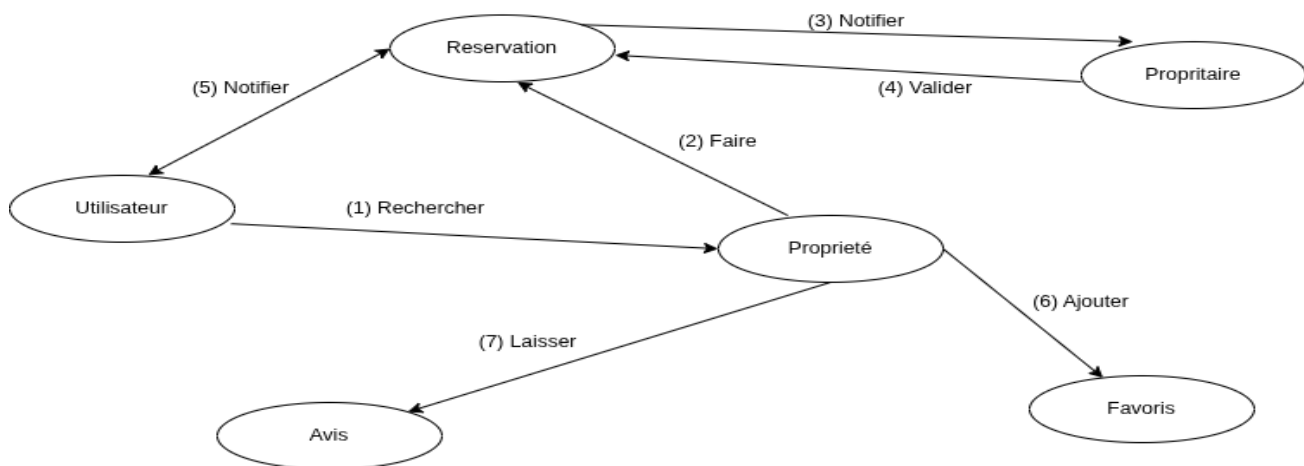


FIGURE 4: DIAGRAMME DE FLUX DE DONNEES (DFD).

5.5 Modélisation des données

Cette section est dédiée à la Modélisation des données de notre système d'information, un passage fondamental pour structurer logiquement et physiquement l'information

5.5.1. Modèle Conceptuel des Données (MCD)

Le Modèle Conceptuel des Données (MCD) permet de décrire, sans contrainte technique, l'ensemble des données manipulées par le système ainsi que les relations entre elles. Sa représentation est la suivante.

La Figure 5 illustre la représentation graphique du Modèle Conceptuel des Données (MCD).

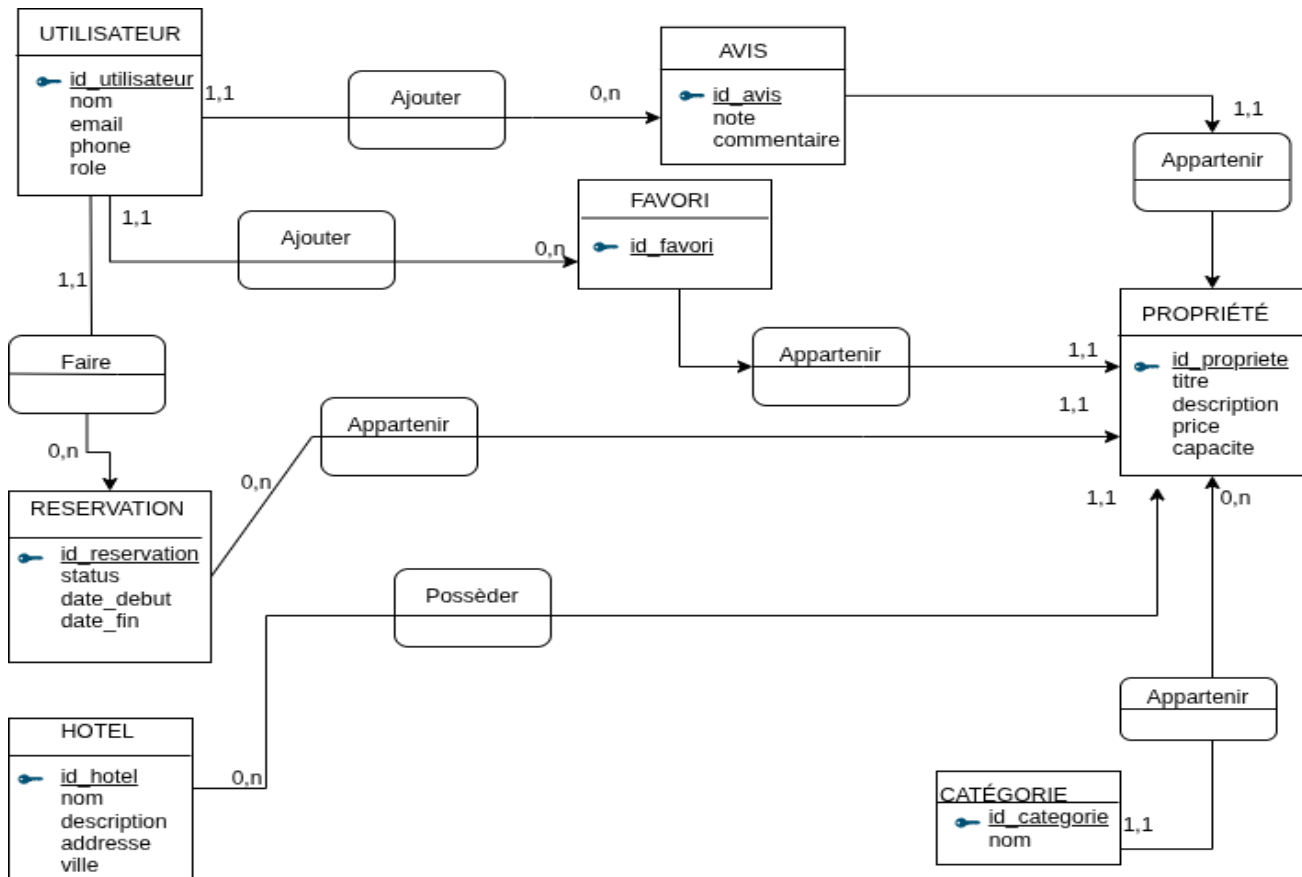


FIGURE 5: MODELE CONCEPTUEL DES DONNEES (MCD).

5.5.2 Modèle Logique des Données (MLD)

Le passage du conceptuel au physique nécessite une structuration adaptée : voici le Modèle Logique des Données (MLD) qui traduit le MCD dans un formalisme orienté base de données.

La Figure 6 ci-dessous illustre la représentation graphique du Modèle Logiques des Données (MLD).

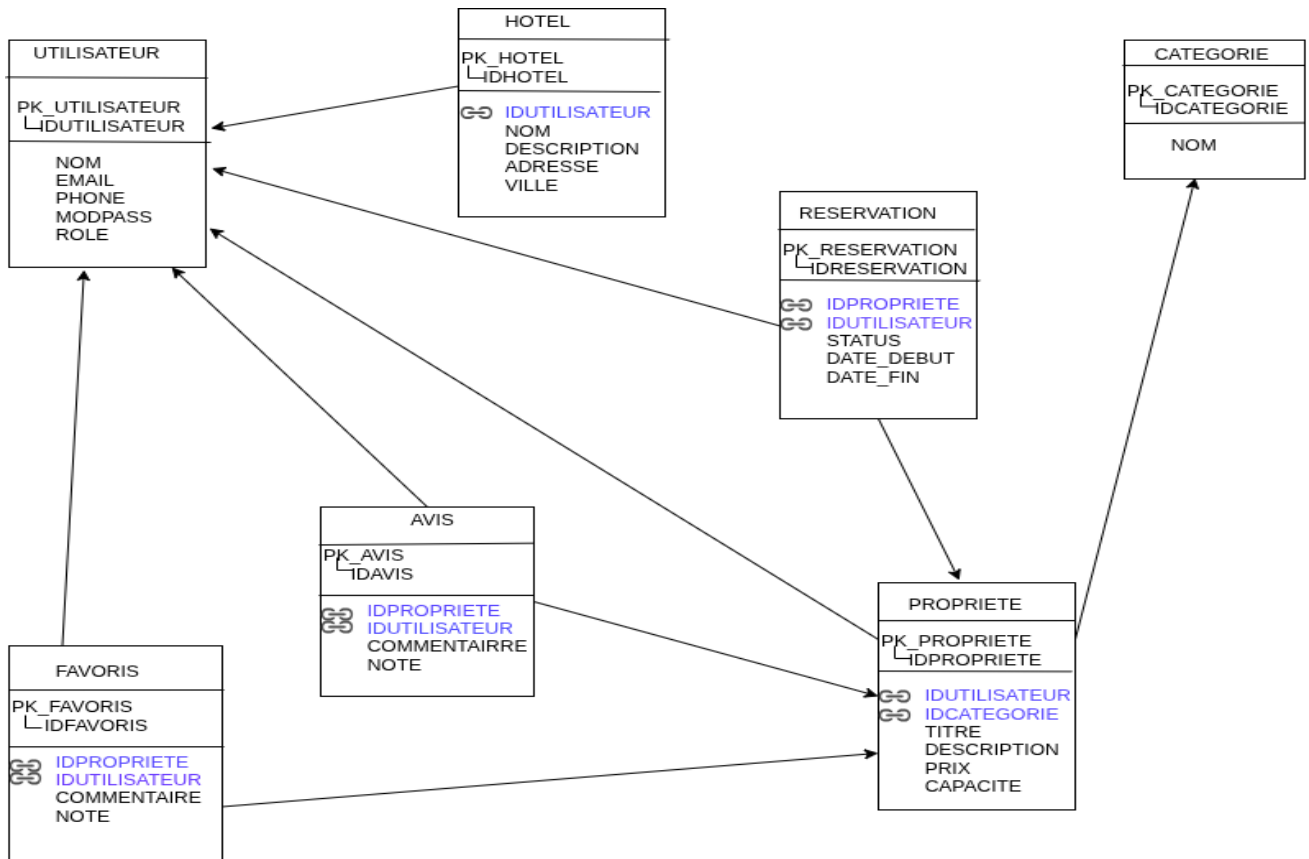


FIGURE 6: MODELE LOGIQUE DES DONNEES (MLD)

5.5.3 Modèle Conceptuel des Traitements (MCT)

Pour définir la dynamique de notre système et les enchaînements logiques des actions, nous avons élaboré le Modèle Conceptuel des Traitements (MCT).

La Figure 7 ci-dessous illustre la représentation graphique du Modèle Conceptuel des Traitements (MCT).

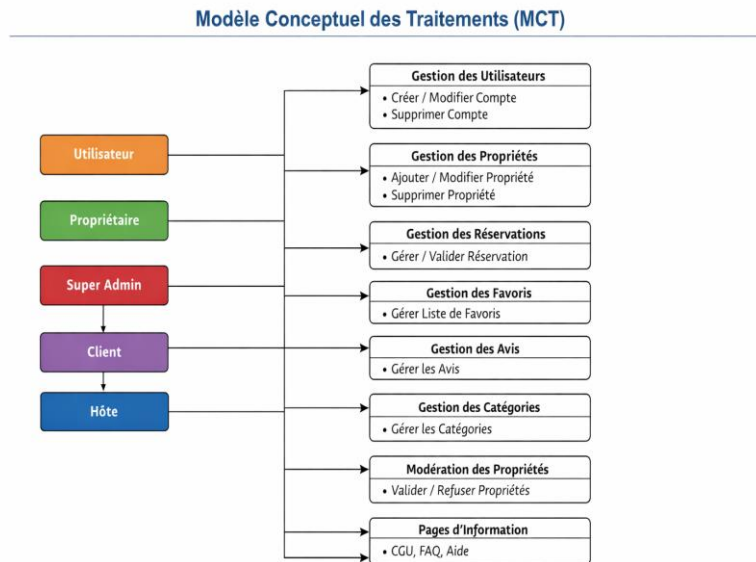


FIGURE 7: MODELE CONCEPTUEL DES TRAITEMENTS (MCT)

5.5.4 Organisationnel des Traitements (MOT)

Le Modèle Organisationnel des Traitements (MOT) apporte la dimension concrète en précisant qui, quand et où chaque traitement est exécuté. Pour compléter l'analyse conceptuelle, voici le MOT qui détaille la répartition des tâches et des traitements dans le temps et l'espace au sein de l'organisation.

La figure 8 présente le Modèle Organisationnel des Traitements (MOT).

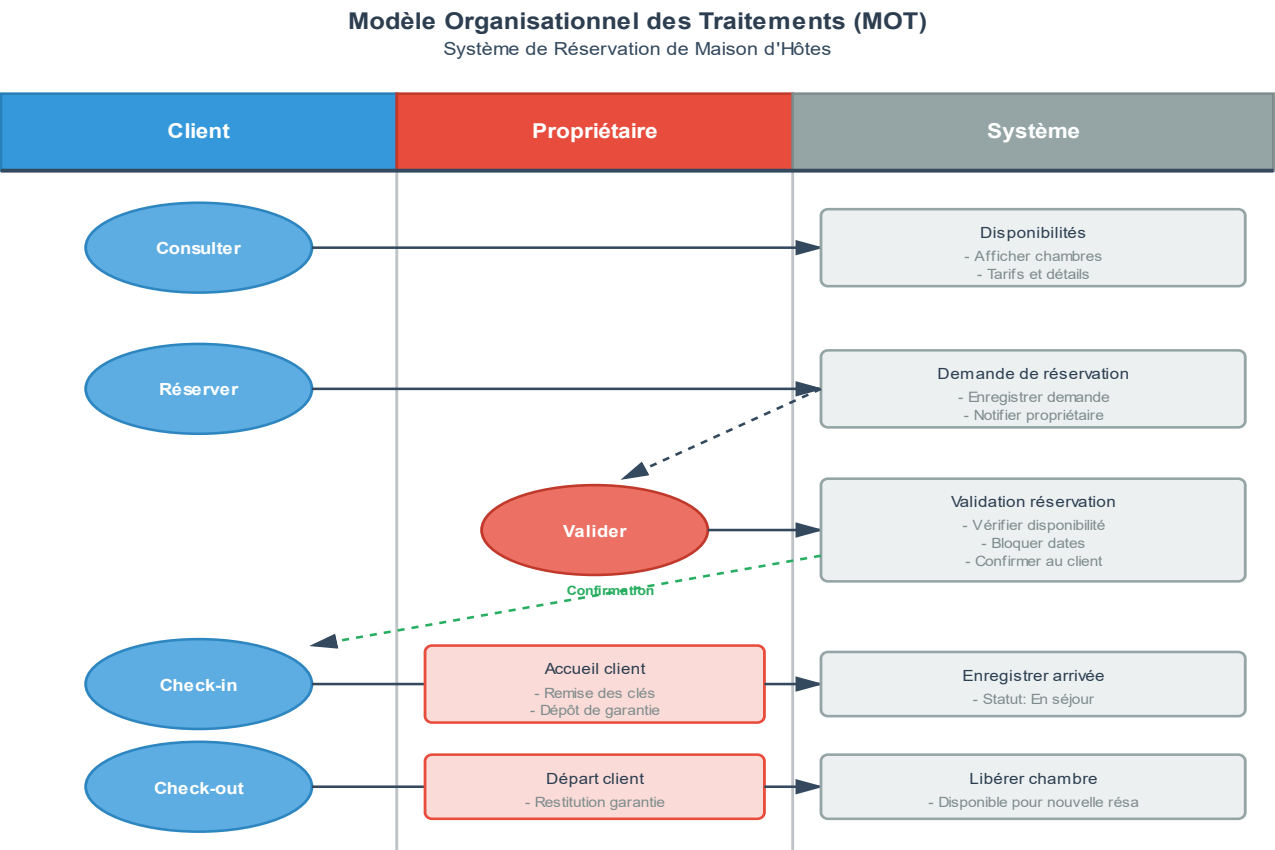


FIGURE 8: MODELE ORGANISATIONNEL DES TRAITEMENTS (MOT)

PARTIE III – REALISATION

Chapitre 6. MISE EN PLACE DE L'ENVIRONNEMENT

Dans cette partie, nous présentons la mise en place de l'environnement de développement de l'application **Ghetto Réservation**, une application web dédiée à la réservation de maisons d'hôtes. Cette étape comprend l'installation et la configuration des outils nécessaires ainsi que la description de l'architecture globale de l'application.

6.1 Installation et configuration des outils

Le développement de cette application a été réalisé dans un environnement de travail stable et adapté aux technologies web modernes.

Nous avons opté pour **Linux Manjaro**, une distribution Linux performante et flexible, particulièrement appréciée pour le développement logiciel.

Les outils utilisés sont :

- Visual Studio Code
- XAMPP (Apache, PHP, MySQL)
- Laravel avec Livewire
- Vite
- Git

6.1.1 L'environnement de développement (IDE) : Visual Studio Code

Pour l'édition, l'organisation et le débogage du code source, **Visual Studio Code (VS Code)** a été choisi comme environnement de développement intégré (IDE).

VS Code offre :

- Une interface légère et intuitive ;
- Une excellente compatibilité avec Linux Manjaro ;
- De nombreuses extensions pour PHP, Laravel, Blade et Livewire.

Sous Linux Manjaro, l'installation de VS Code peut se faire via le gestionnaire de paquets pacman ou via l'AUR.

Vérification de l'installation :

➤ `code --version`

La Figure 9 ci-dessous illustre l'interface de Visual Studio Code utilisée lors du développement.

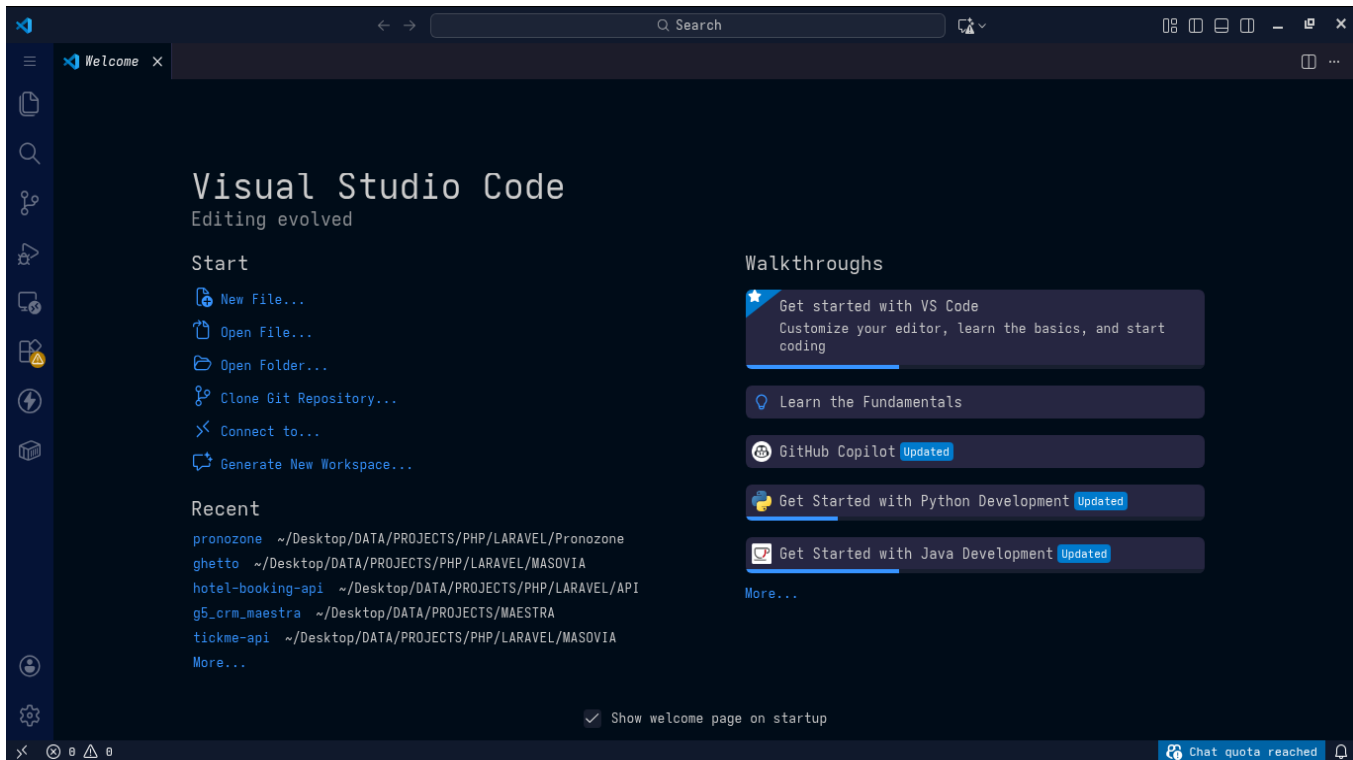


FIGURE 9: INTERFACE DE VS CODE.

6.1.2 Serveur Web et PHP : XAMPP

Le serveur web et l'environnement PHP ont été mis en place grâce à **XAMPP**, une solution tout-en-un intégrant :

- Apache HTTP Server ;
- PHP ;
- MySQL.

XAMPP a été choisi pour :

- Sa simplicité d'installation ;
- Sa facilité de configuration ;
- Sa rapidité de mise en œuvre en environnement local.

Après l'installation, les services Apache et MySQL sont démarrés depuis le panneau de contrôle XAMPP ou via le terminal.

Vérification du serveur Apache :

➤ `sudo /opt/lampp/lampp status`

6.1.3 Système de Gestion de Base de Données : MySQL

La gestion des données de l'application est assurée par **MySQL**, intégré dans XAMPP.

MySQL a été retenu pour :

- Sa fiabilité ;
- Son intégration native avec Laravel via l'ORM Eloquent ;
- Sa capacité à gérer efficacement les données relationnelles.

La base de données stocke notamment :

- Les utilisateurs ;
- Les maisons d'hôtes ;
- Les réservations ;

6.1.4 Gestion de versions : Git

Le système de gestion de versions **Git** est utilisé pour assurer le suivi du code source et faciliter l'évolution du projet.

Vérification de l'installation :

- `git --version`

Configuration initiale :

- `git config --global user.name "Isaia-mhd"`
- `git config --global user.email "mohamedesaie21@gmail.com"`

6.1.5 Installation de PHP et Composer

Laravel nécessite PHP 8.x ainsi que **Composer**, le gestionnaire de dépendances PHP.

Dans notre environnement XAMPP, PHP est déjà inclus. Composer a été installé séparément afin de gérer les dépendances du projet Laravel.

Vérification :

- `php -v`
- `composer --version`

6.1.6 Installation de Laravel, Livewire et Vite

La création du projet Laravel s'effectue à l'aide de Composer :

- `composer create-project laravel/laravel ghetto`

Installation de Livewire

- `composer require livewire/livewire`

Installation des dépendances frontend avec Vite

- npm install
- npm run dev

6.2 Architecture de l'application

L'application Ghetto Réservation repose sur une architecture en trois couches (3-tiers), garantissant une meilleure organisation, une maintenance facilitée et une évolutivité accrue.

6.2.1. Couche Présentation (Frontend)

La couche présentation est assurée par :

- Blade (moteur de templates Laravel) ;
- Livewire pour les interactions dynamiques.

Cette couche permet :

- L'affichage des maisons d'hôtes ;
- La gestion des formulaires de réservation ;
- Les interactions utilisateur sans rechargement de page.

6.2.2. Couche Logique Métier (Backend)

La logique métier est implémenté avec **Laravel**.

Elle assure :

- Le traitement des réservations ;
- La gestion des utilisateurs ;
- La validation des paiements ;
- La communication avec la base de données.

Les composants Livewire jouent un rôle central dans la gestion des actions utilisateur.

6.2.3. Architecture de l'application Couche Données (Base de données MySQL)

La couche données repose sur **MySQL**, accessible via **Eloquent ORM**.

Elle permet :

- L'enregistrement des données ;
- La mise à jour des informations ;
- La récupération des données nécessaires à l'affichage.

6.2.4. Interaction entre les couches

Le fonctionnement global de l'application suit les étapes suivantes :

- L'utilisateur interagit avec l'interface web.
- Livewire intercepte l'action et envoie la requête au serveur Laravel.
- Laravel applique la logique métier et interagit avec MySQL.
- Les résultats sont renvoyés à Livewire.
- L'interface utilisateur est mise à jour dynamiquement.

La figure 10 suivante illustre l'architecture client–serveur de l'application.

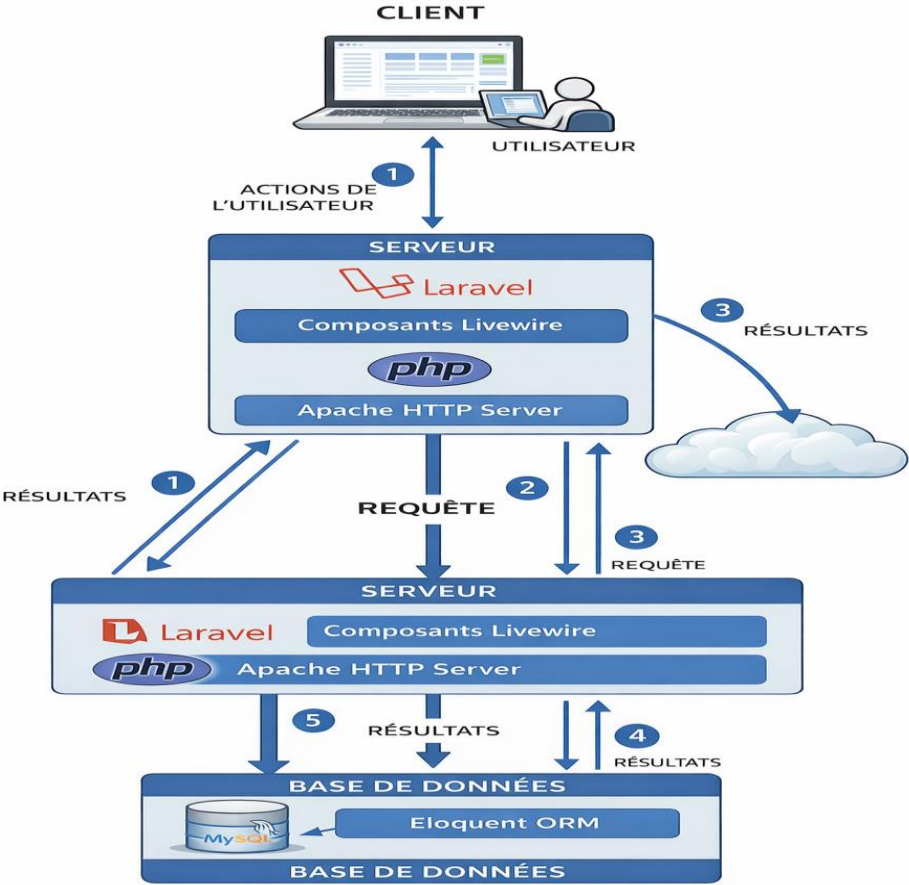


FIGURE 10: L'ARCHITECTURE CLIENT–SERVEUR DE L'APPLICATION

Chapitre 7. DEVELOPPEMENT DE L'APPLICATION

Dans cette section, nous passons de la phase de conception à la phase de réalisation concrète de l'application **Ghetto** **Réservation**. Elle présente successivement la **création de la base de données**, le **codage de l'application** ainsi que la **présentation de l'interface finale**.

7.1. Création de la base de données

La création de la base de données a été réalisée à l'aide du framework **Laravel**, qui fournit des outils intégrés puissants tels que **Eloquent ORM** et le système de **migrations**, facilitant la gestion et l'évolution de la structure de la base de données.

Avant toute création, les paramètres de connexion à la base de données doivent être configurés dans le fichier **.env** du projet Laravel.

Exemple de configuration utilisée :

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=reservation
DB_USERNAME=root
DB_PASSWORD=
```

Où :

- **root** représente le nom d'utilisateur MySQL ;
- **DB_PASSWORD** correspond au mot de passe MySQL (vide par défaut sous XAMPP) ;
- **127.0.0.1** est l'adresse du serveur local ;
- **3306** est le port par défaut de MySQL ;
- **reservation** est le nom de la base de données.

La figure 11 suivante illustre la configuration du fichier .env du projet Laravel.

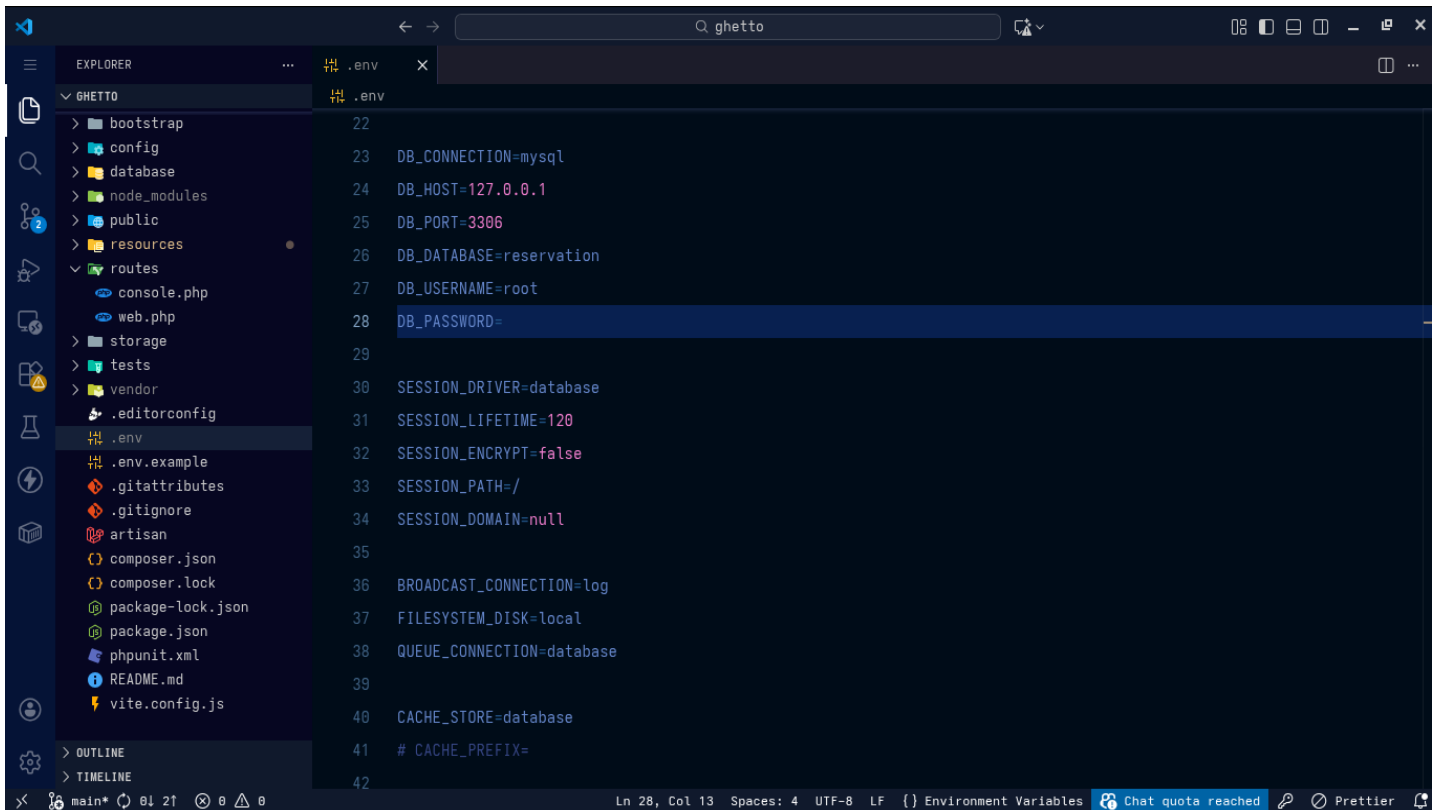


FIGURE 11: CONFIGURATION DU FICHIER .ENV DU PROJET LARAVEL

7.1.1. Création des tables à l'aide des migrations

Laravel permet de créer les tables de la base de données à travers des **migrations** générées automatiquement via Artisan.

Exemple de commande pour créer une migration :

- `php artisan make:migration create_maisons_table`

Les entités principales créées dans la base de données sont :

- Utilisateur
- Propriete
- Réservation
- Avis
- Favoris

Une fois les migrations définies, la commande suivante permet de créer physiquement les tables dans la base de données :

- `php artisan migrate`

La figure 12 suivante montre l'exécution des migrations dans Laravel.

```
mohamed@mohamed:~/Desktop/DATA/PROJECTS/PHP/LARAVEL/MASOVIA/ghetto
php artisan migrate

WARN The database 'reservation' does not exist on the 'mysql' connection.

Would you like to create it?
Yes

INFO Preparing database.
Creating migration table ..... 186.01ms DONE

INFO Running migrations.

0001_01_01_000000_create_users_table ..... 1 s DONE
0001_01_01_000001_create_cache_table ..... 524.39ms DONE
0001_01_01_000002_create_jobs_table ..... 1 s DONE
2025_10_09_140834_add_phone_to_users ..... 134.44ms DONE
2025_10_13_082853_create_hotels_table ..... 1 s DONE
2025_10_13_083716_create_categories_table ..... 167.38ms DONE
2025_10_13_083744_create_properties_table ..... 3 s DONE
2025_10_17_083608_create_favorites_table ..... 2 s DONE
2025_10_22_065702_create_bookings_table ..... 1 s DONE
2025_10_23_094925_create_tarifications_table ..... 211.56ms DONE
2025_11_09_091325_create_notifications_table ..... 402.15ms DONE
2025_11_29_125925_add_owner_id_to_bookings ..... 1 s DONE
2025_12_13_095249_create_reviews_table ..... 2 s DONE
2025_12_16_082248_add_can_host_to_users ..... 100.26ms DONE
2025_12_16_141050_create_subscribers_table ..... 1 s DONE
```

FIGURE 12: EXECUTION DES MIGRATIONS DANS LARAVEL

7.2. Codage de l'application

Après la mise en place de la base de données, l'étape suivante consiste à développer l'ensemble des fonctionnalités de l'application. Cette phase de codage couvre à la fois la **partie Frontend**, responsable de l'interface utilisateur, et la **partie Backend**, chargée de la logique métier et de l'accès aux données.

7.2.1. Codage de la partie Frontend (Laravel Blade & Livewire)

La partie Frontend de l'application est développée à l'aide de :

- **Blade**, moteur de templates de Laravel ;
- **Livewire**, permettant de créer des interfaces dynamiques sans recourir à JavaScript complexe ;
- **Vite**, pour la gestion des fichiers CSS et JavaScript.

Les composants Livewire constituent le cœur de l'interface utilisateur. Chaque fonctionnalité importante (recherche de maison, réservation, gestion des comptes) est associée à un composant Livewire dédié.

Par exemple :

- Composant de recherche de maisons d'hôtes ;
- Composant de réservation ;
- Composant de gestion du profil utilisateur ;

La figure 13 suivante montre un extrait de code d'un composant Livewire utilisé pour la réservation.

```

1 <div x-data="{ open: false }" class="relative">
10 <div>
15
16 <div @click.outside="open = false"
17 class="bg-white rounded-xl shadow-lg p-6 w-xl z-10 transform transition-transform scale-95 animate-zoomIn">
18
19 <h2 class="text-lg font-semibold text-gray-800 mb-4">Réserver cette maison</h2>
20 @include(view: "components.error")
21 {{-- booking content --}}
22 <div x-data="bookingComponent()" x-init="init()" class="space-y-4">
23
24 {{-- Guest --}}
25 <div>
26 <label class="block text-xs font-medium text-gray-700 mb-1">Nombre de personnes</label>
27 <input type="number" wire:model="guest" placeholder="Ex: 2"
28 class="w-full rounded-md px-3 py-2 text-sm border border-gray-200 shadow-md outline-none">
29 @error('guest')
30 <p class="text-red-500 text-xs">{{ $message }}</p>
31 @enderror
32 </div>
33

```

FIGURE 13: EXTRAIT DE CODE DU COMPOSANT LIVERWIRE DE RESERVATION

7.2.2. Codage de la partie Backend (Laravel)

La partie Backend est entièrement développée avec **Laravel**.

Elle assure :

- Le traitement des requêtes utilisateurs ;
- La validation des données ;
- La gestion des réservations ;
- La communication avec la base de données via **Eloquent ORM**.

La logique métier est réparti entre :

- Les **Controllers** ;
- Les **Models** ;
- Les **Livewire Components**.

La figure 14 suivante illustre un extrait de code d'un contrôleur Laravel gérant les maisons d'hôtes.

```
app > Livewire > Property.php > Property > mount()
9 class Property extends Component
    0 references | 0 overrides
    public function render(): View
    {
        $categories = Category::all();

        $properties = Apartment::query()
            ->where(column: 'active', operator: true)

            ->when(value: $this->selectedCategory, callback: function (Builder<Property> $query): void {
                $query->where(column: 'category_id', operator: $this->selectedCategory);
            })

            ->when(value: $this->minPrice, callback: function (Builder<Property> $query): void {
                $query->where(column: 'pricePerNight', operator: '>=', value: $this->minPrice);
            })

            ->when(value: $this->maxPrice, callback: function (Builder<Property> $query): void {
                $query->where(column: 'pricePerNight', operator: '<=', value: $this->maxPrice);
            })
    }
    61
```

FIGURE 14: EXTRAIT DE CODE D'UN CONTROLEUR LARAVEL POUR LA GESTION DES MAISONS

7.3. Présentation de l'application

Cette partie présente les principales interfaces de l'application **Ghetto**.

7.3.1. Page d'accueil

La page d'accueil est accessible à tous les visiteurs sans authentification. Elle permet :

- De consulter les maisons d'hôtes disponibles ;
- D'effectuer des recherches selon différents critères (localisation, prix, disponibilité)

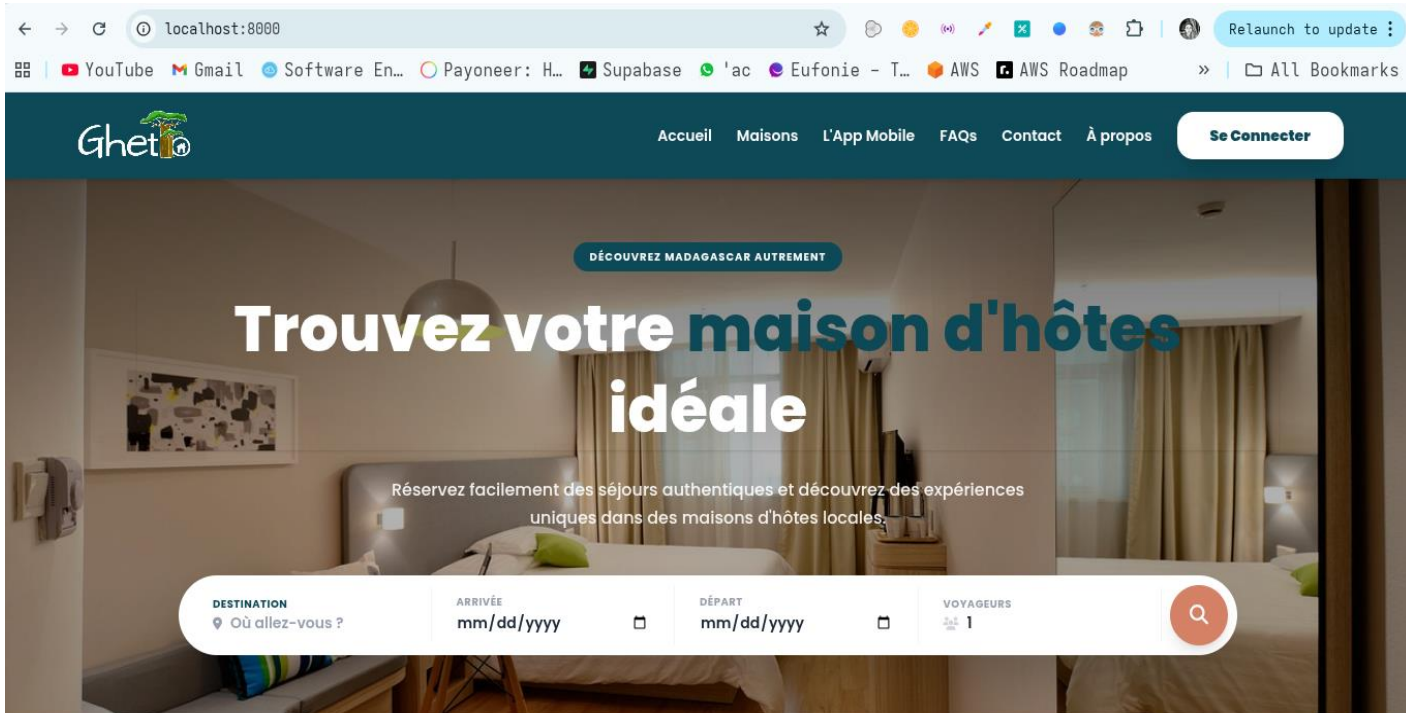


FIGURE 15: PAGE D'ACCUEIL DE L'APPLICATION GHETTO

7.3.2. Inscription et authentification

L'application propose un système d'inscription et de connexion sécurisé permettant aux utilisateurs de créer un compte afin d'effectuer des réservations.

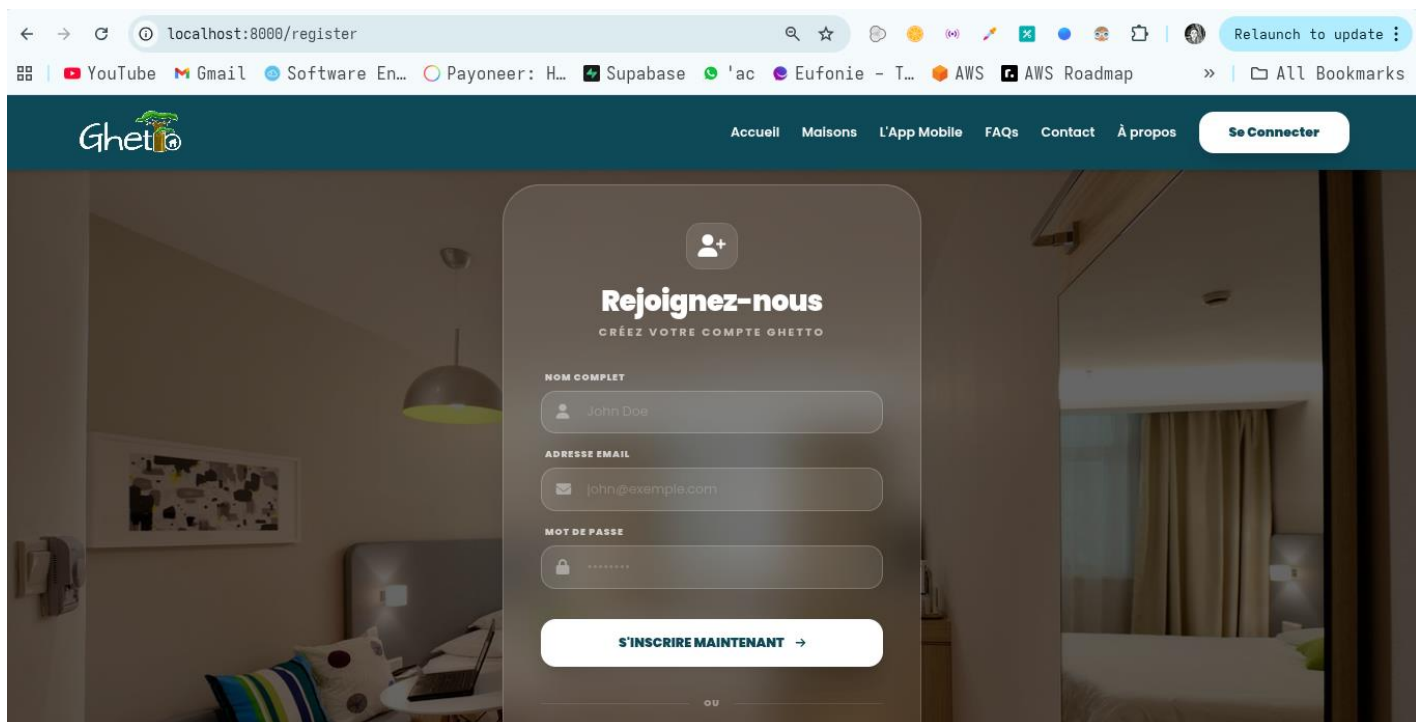


FIGURE 16: PAGE D'INSCRIPTION UTILISATEUR

7.3.3. Réservation d'une maison d'hôte

Un utilisateur connecté peut :

- Consulter les détails d'une maison ;
- Sélectionner les dates de séjour ;
- Effectuer une réservation.

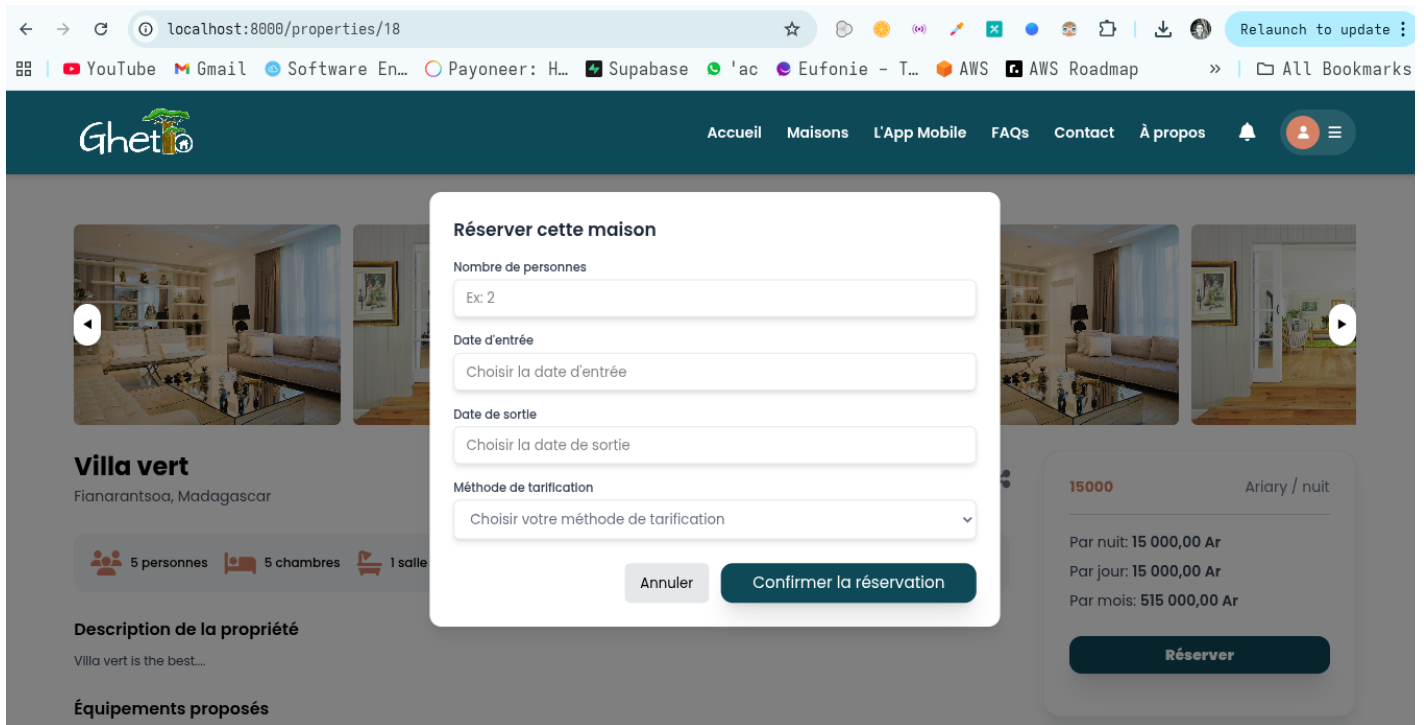


FIGURE 17: INTERFACE DE RESERVATION D'UNE MAISON D'HOTE

7.3.4. Tableau de bord Administrateur

L'application **Ghetto Réservation** distingue **deux types de tableaux de bord administratifs**, chacun répondant à des rôles et responsabilités spécifiques au sein du système.

7.3.4.1. Tableau de bord Administrateur principal (Super Administrateur)

Le **Super Administrateur** représente l'administrateur principal de l'application. Il dispose d'un accès global lui permettant de superviser l'ensemble de la plateforme.

À partir de son tableau de bord, le Super Administrateur peut :

- Valider ou refuser l'inscription des nouveaux propriétaires de maisons d'hôtes ;
- Approuver ou rejeter les nouvelles maisons d'hôtes proposées sur la plateforme ;
- Superviser toutes les réservations effectuées par les utilisateurs ;
- Suivre l'activité globale de l'application ;
- Gérer les comptes utilisateurs et propriétaires.

Ce tableau de bord joue un rôle essentiel dans la **sécurité**, la **qualité des annonces** et le **bon fonctionnement général** de la plateforme.

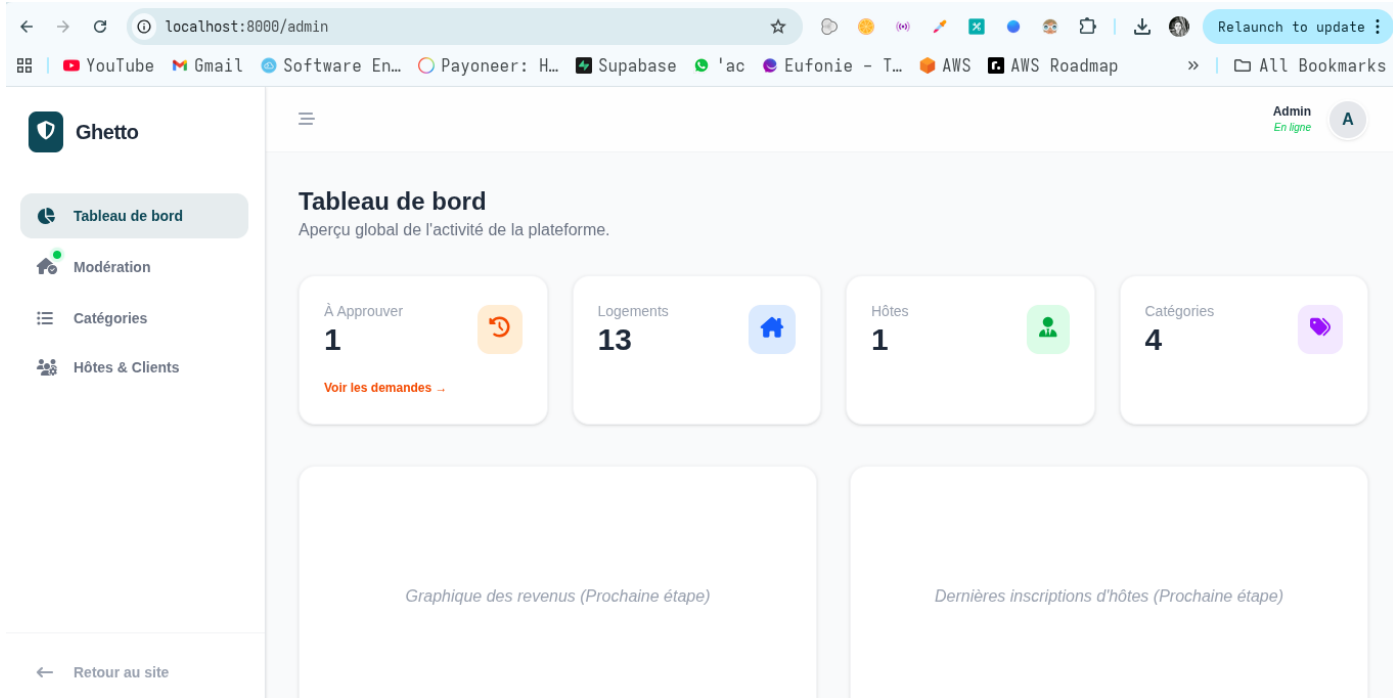


FIGURE 18: TABLEAU DE BORD DU SUPER ADMINISTRATEUR

7.3.4.2. Tableau de bord Propriétaire de maison d'hôte

Le **Propriétaire de maison d'hôte** dispose d'un tableau de bord spécifique, limité à la gestion de ses propres biens et réservations.

Depuis ce tableau de bord, le propriétaire peut :

- Ajouter de nouvelles maisons d'hôtes ;
- Modifier les informations de ses propriétés (description, prix, disponibilité, photos) ;
- Consulter la liste des réservations concernant ses maisons ;
- Valider ou refuser les demandes de réservation ;
- Suivre l'historique des réservations ;

Ce tableau de bord permet au propriétaire de gérer efficacement son activité tout en respectant les règles définies par le Super Administrateur.

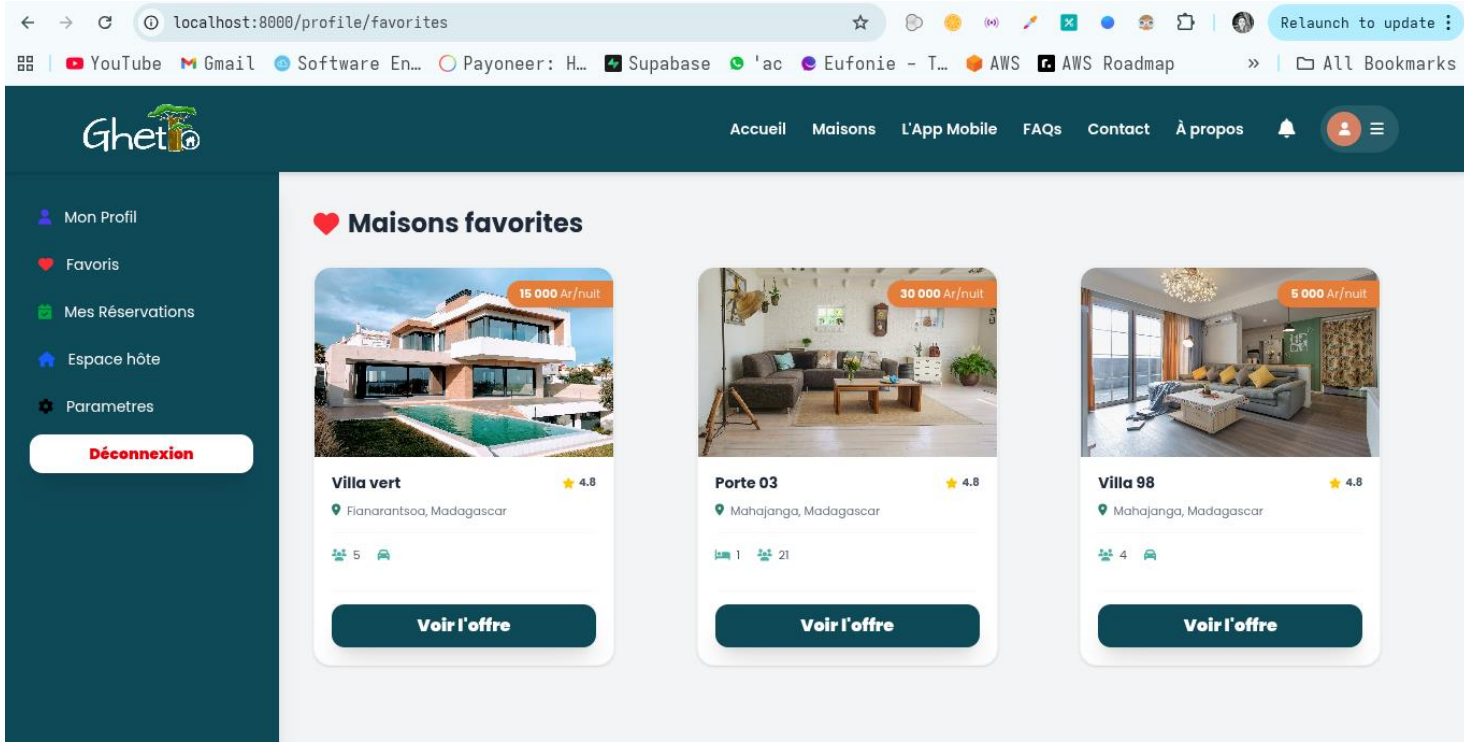


FIGURE 19: INTERFACE UTILISATEUR DE LA PAGE DES FAVORIS

Synthèse des rôles

TABLEAU 13: SYNTHESE DES ROLES

Rôle	Responsabilités principales
Super Administrateur	Validation des propriétaires et maisons, supervision globale
Propriétaire	Gestion des propriétés et des réservations

7.3.5. Autres pages de l’application

Cette section regroupe les interfaces secondaires de l’application. Les figures suivantes présentent les différentes pages permettant aux utilisateurs et aux propriétaires d’interagir efficacement avec la plateforme.

La figure 20 montre la page permettant à l’utilisateur de consulter et gérer les maisons d’hôtes ajoutées à ses favoris.

La figure 21 montre la page permettant à l’utilisateur de consulter l’historique de ses réservations, incluant les détails et le statut de chaque réservation.

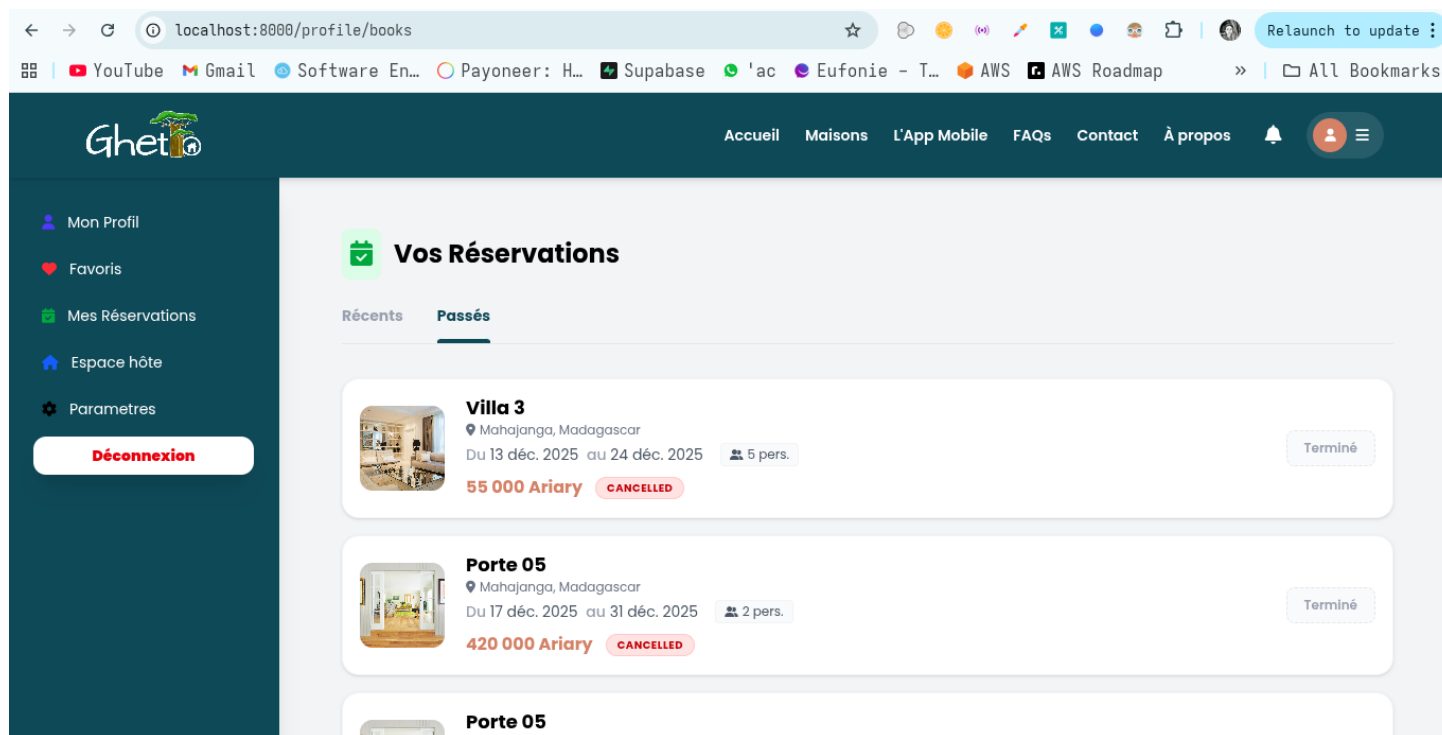


FIGURE 21: INTERFACE UTILISATEUR DE LA PAGE D’HISTORIQUE DES RESERVATIONS

La figure 22 illustre la page des conditions générales d’utilisation destinées aux hôtes, permettant leur consultation au sein de l’application.

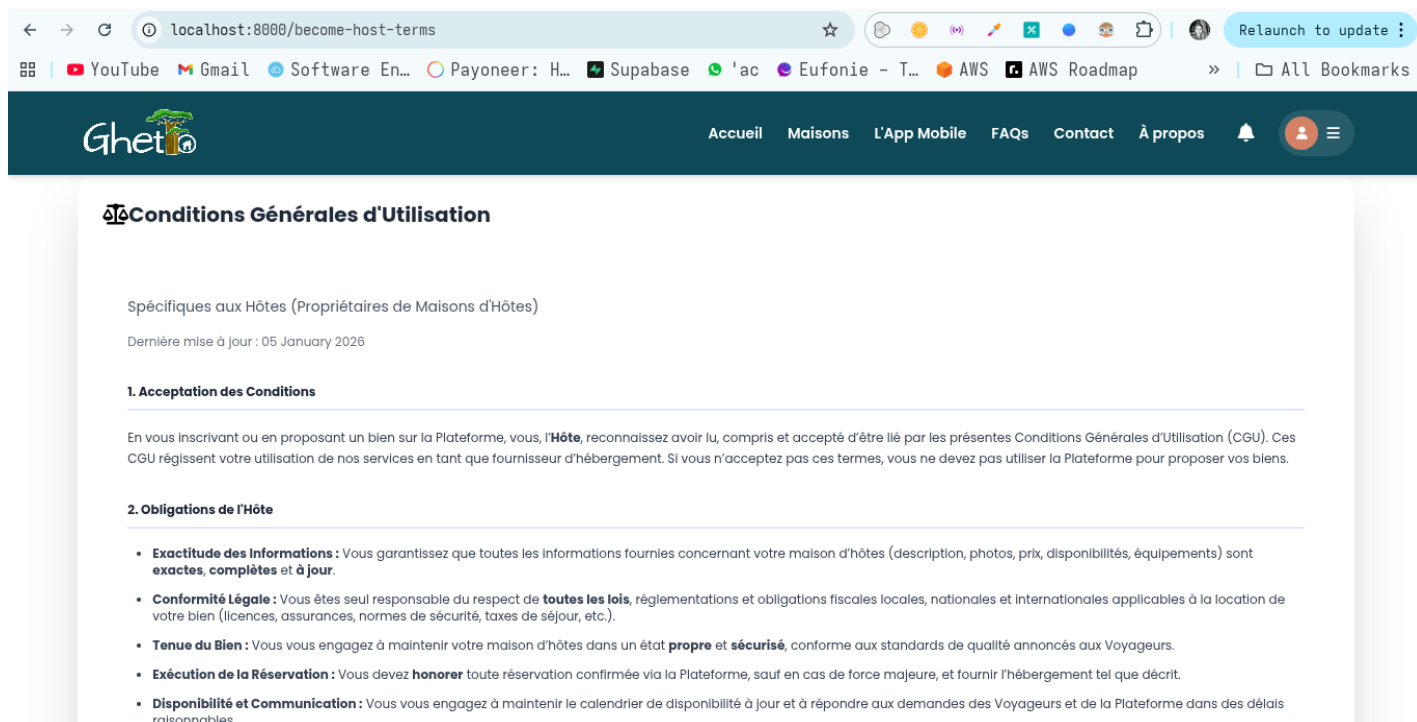


FIGURE 22: INTERFACE UTILISATEUR DES CONDITIONS GENERALES D’UTILISATION POUR LES HOTES

La figure 23 présente la page de détails d’une propriété, affichant les informations essentielles, les images et les caractéristiques de la maison d’hôte.

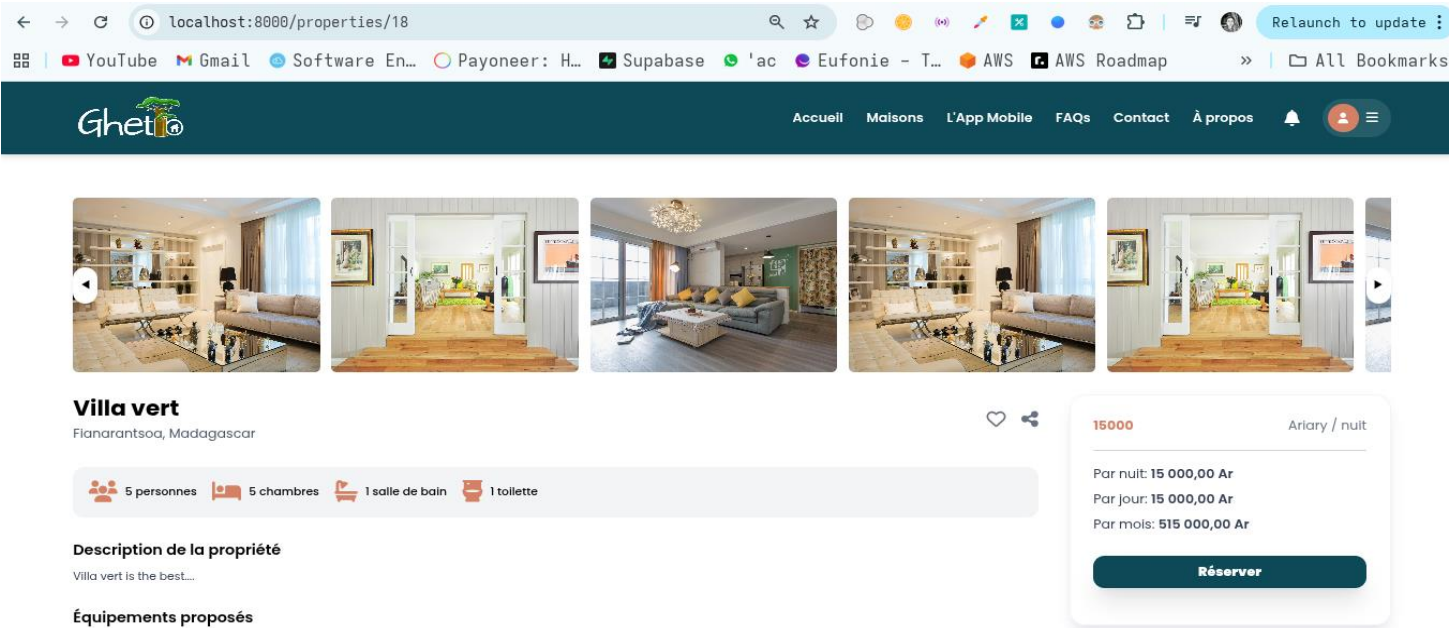


FIGURE 23: INTERFACE DE LA PAGE DE DETAILS D’UNE PROPRIETE

La figure 24 illustre la page d’affichage des propriétés disponibles

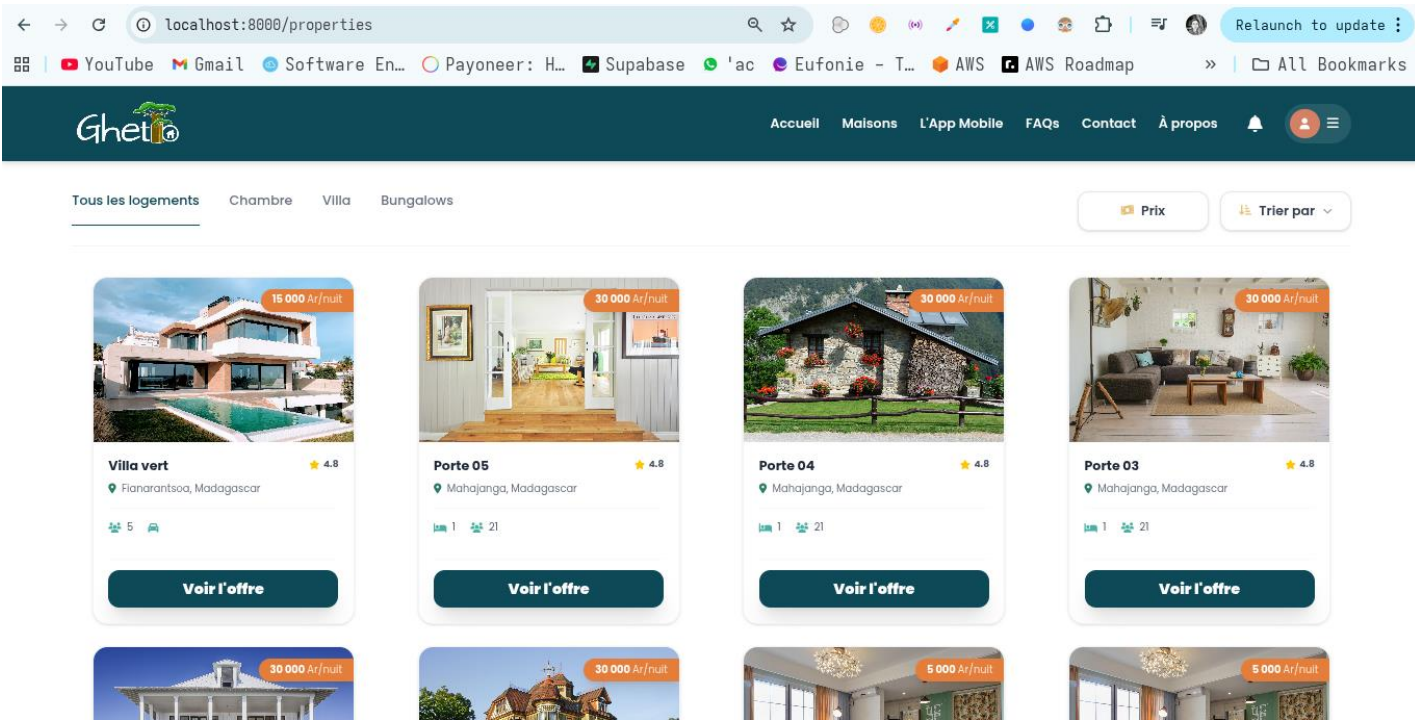


FIGURE 24: INTERFACE UTILISATEUR DE CONSULTATION DES PROPRIETES

La figure 25 illustre la page des avis d’une propriété, permettant aux utilisateurs de consulter les évaluations et commentaires laissés par les clients précédents.

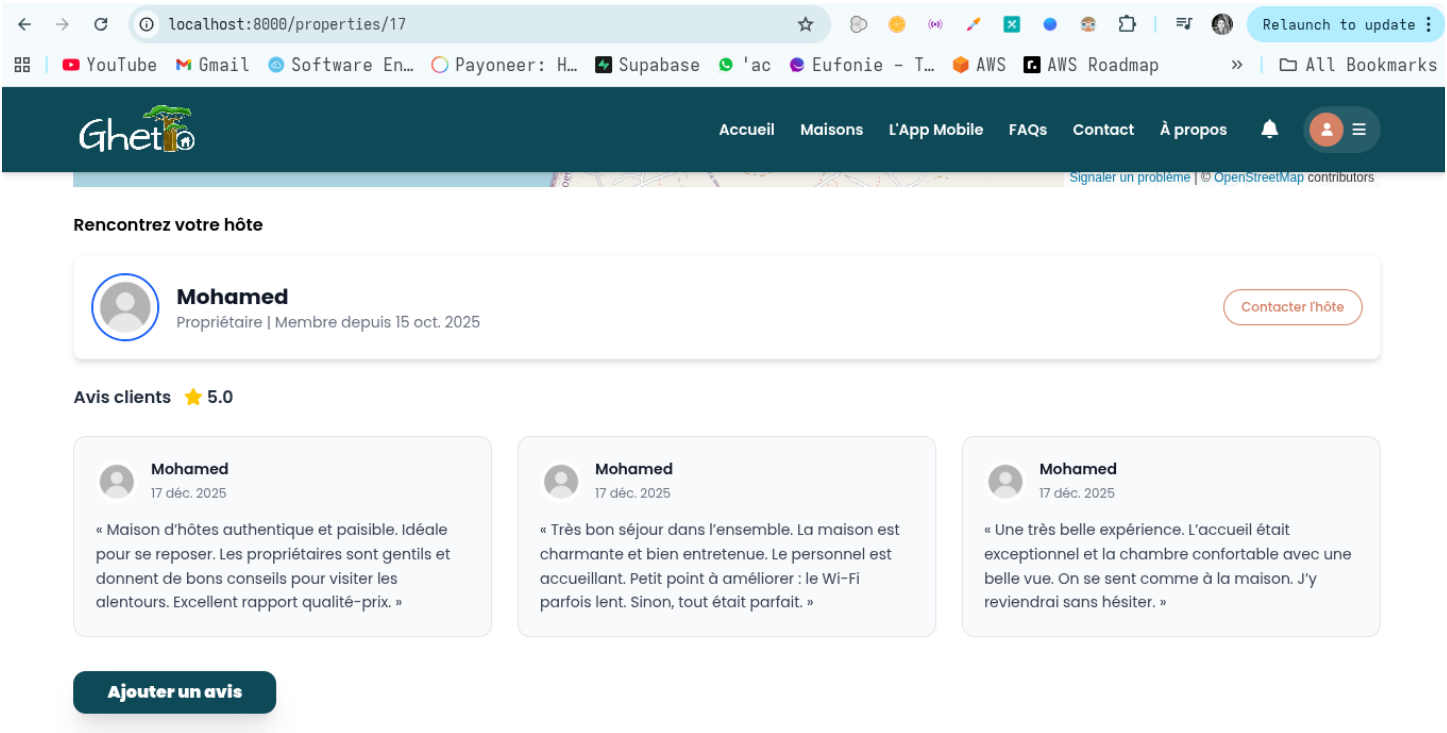


FIGURE 25: INTERFACE DE CONSULTATION DES AVIS D’UNE PROPRIETE

La figure 26 illustre la page affichée lorsque les critères de recherche saisis par l'utilisateur ne correspondent à aucune propriété disponible.

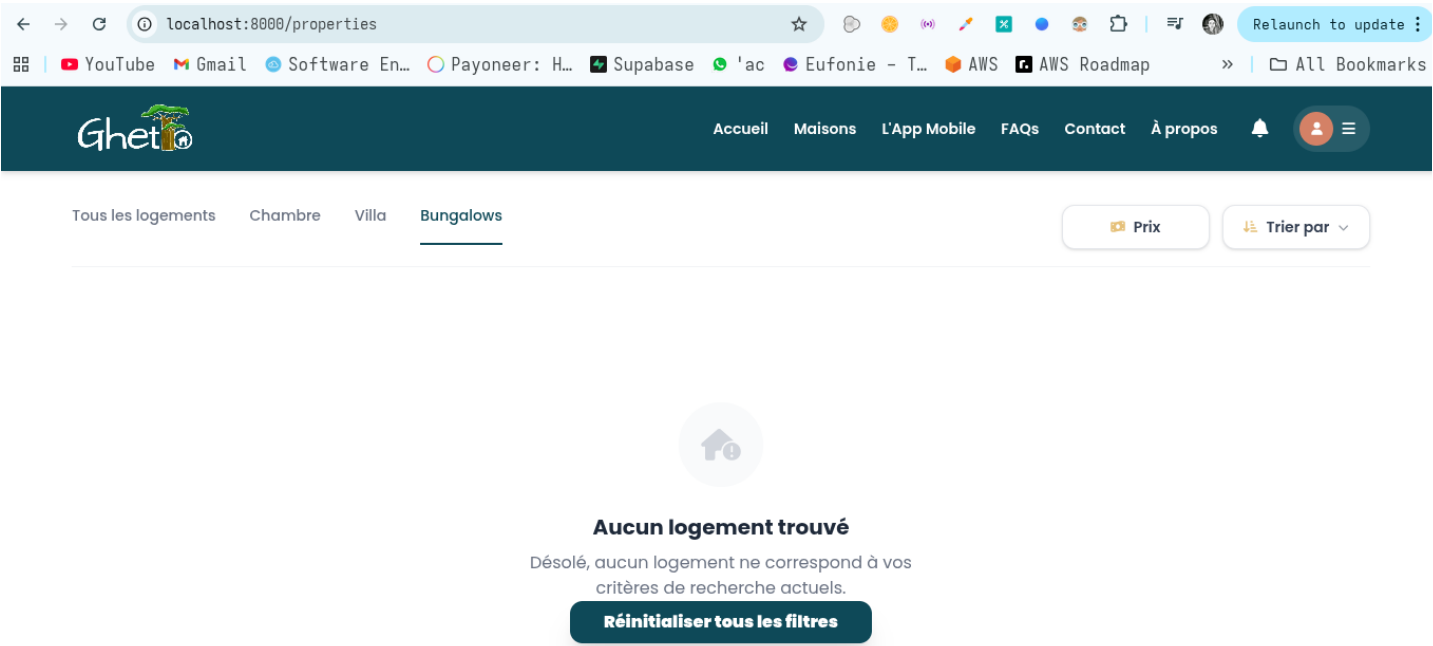


FIGURE 26: INTERFACE DE RESULTAT DE RECHERCHE SANS PROPRIETE CORRESPONDANTE

La figure 27 présente la page de gestion des utilisateurs, permettant à l'administrateur de consulter, modifier les comptes des utilisateurs de l'application.

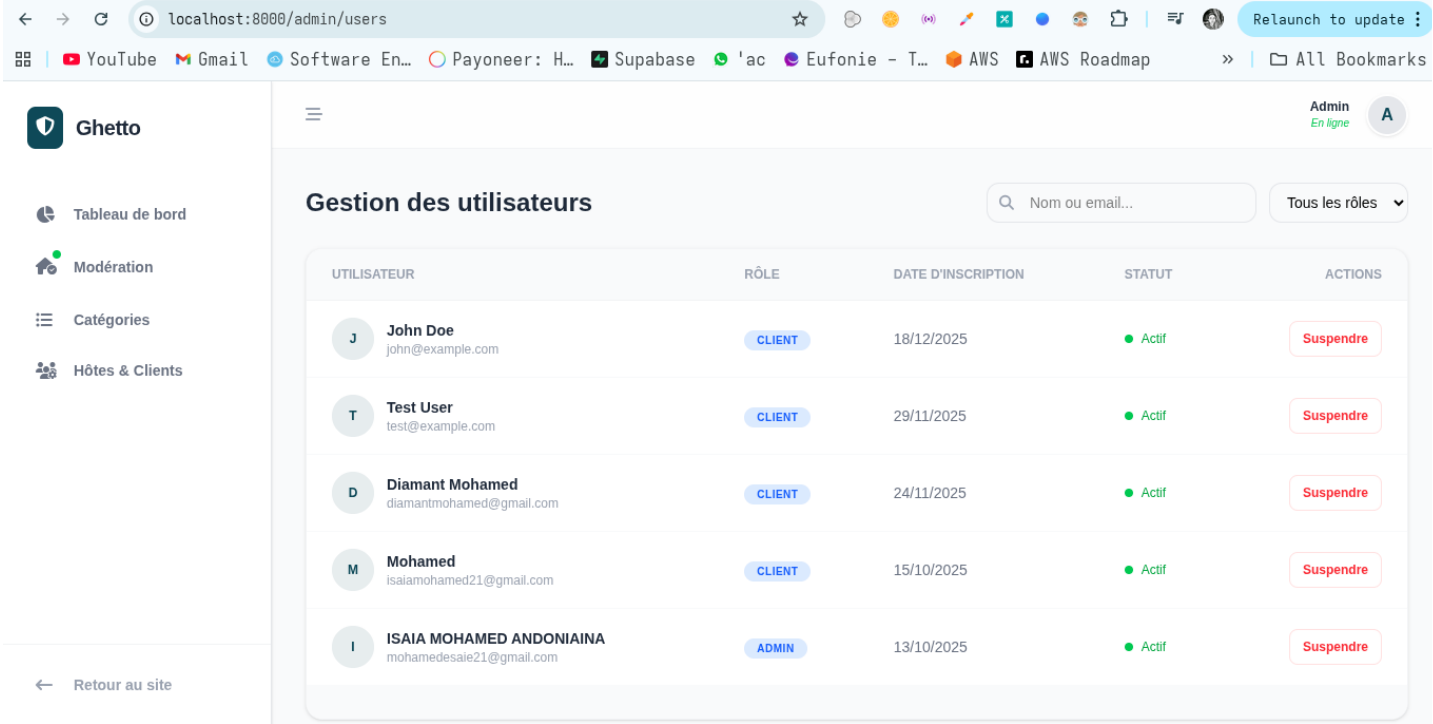


FIGURE 27: INTERFACE DE GESTION DES UTILISATEURS PAR L'ADMINISTRATEUR

La figure 28 présente la page de gestion des catégories, permettant à l’administrateur d’ajouter, modifier ou supprimer les catégories de propriétés disponibles dans l’application.

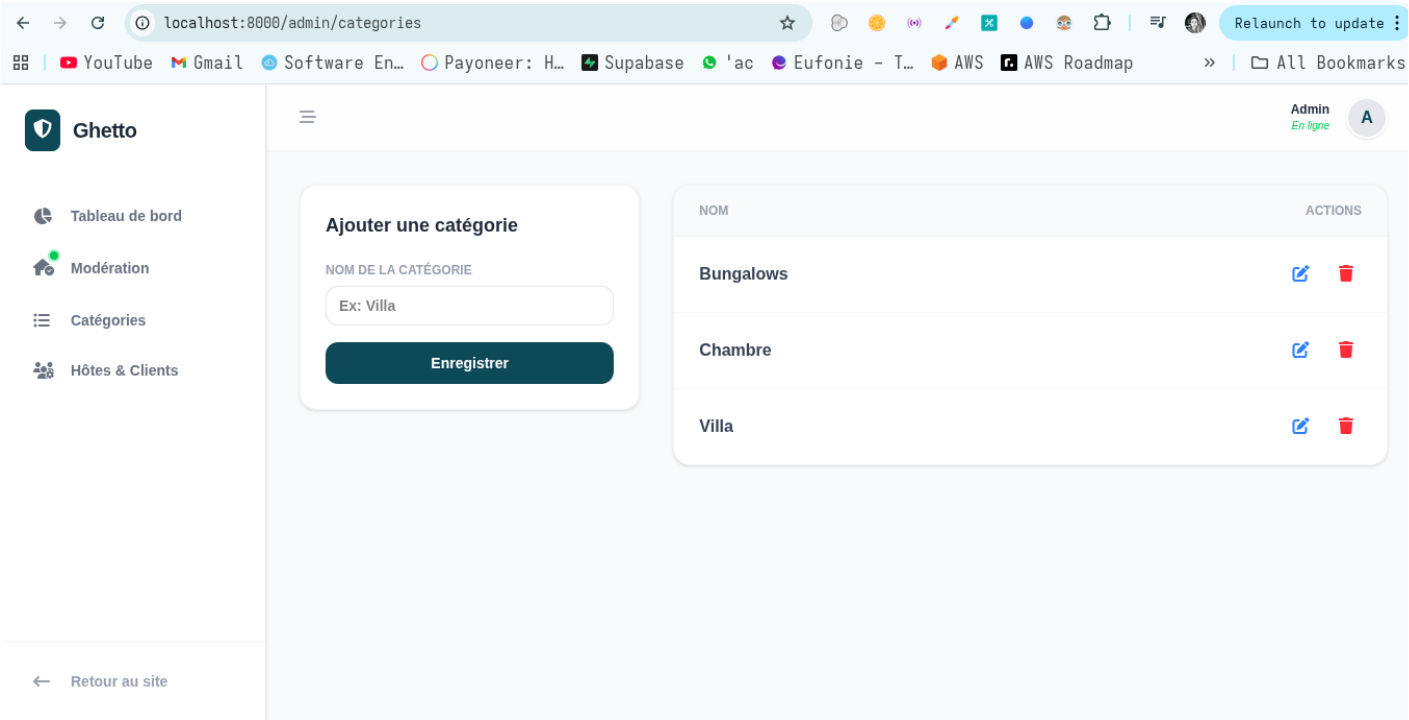


FIGURE 28: INTERFACE DE GESTION DES CATEGORIES PAR L’ADMINISTRATEUR

La figure 29 illustre la page de modération, permettant à l’administrateur de valider ou de refuser les nouvelles **propriétés soumises par les propriétaires.**

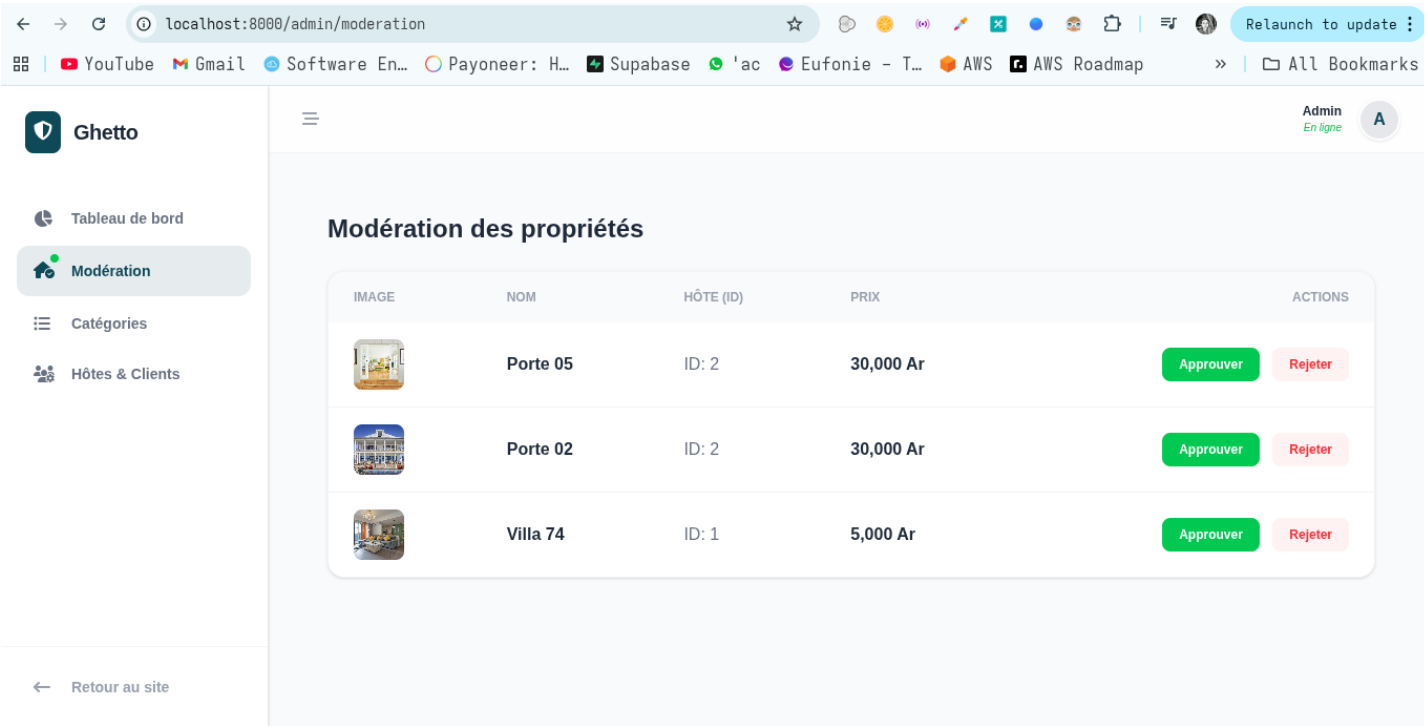


FIGURE 29: INTERFACE DE MODERATION DES NOUVELLES PROPRIETES PAR L’ADMINISTRATEUR

La figure 30 présente la page permettant au propriétaire de gérer les réservations de ses propriétés, avec la possibilité de consulter, accepter ou refuser chaque réservation.

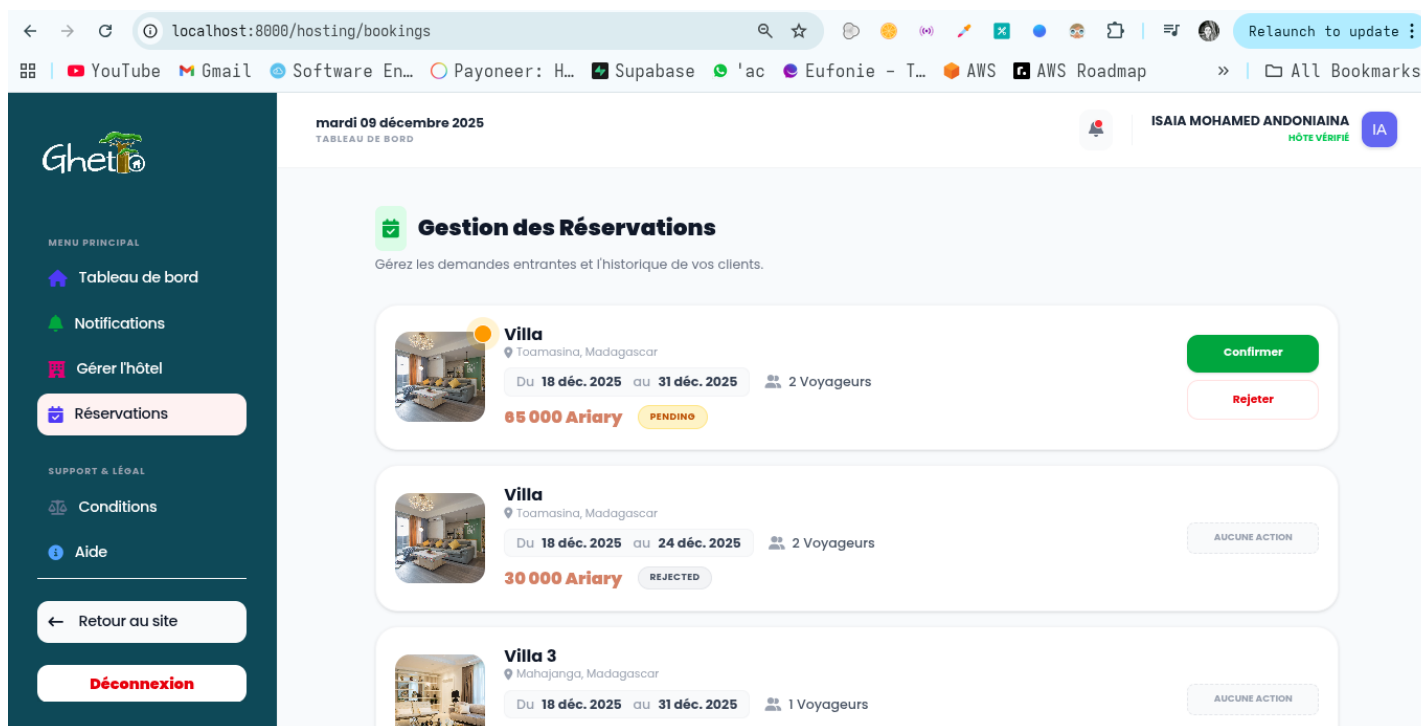


FIGURE 30: INTERFACE DE GESTION DES RESERVATIONS PAR LE PROPRIETAIRE

La figure 31 illustre la page FAQ, permettant aux utilisateurs de consulter les réponses aux questions les plus fréquentes concernant l'utilisation de l'application et les services proposés.

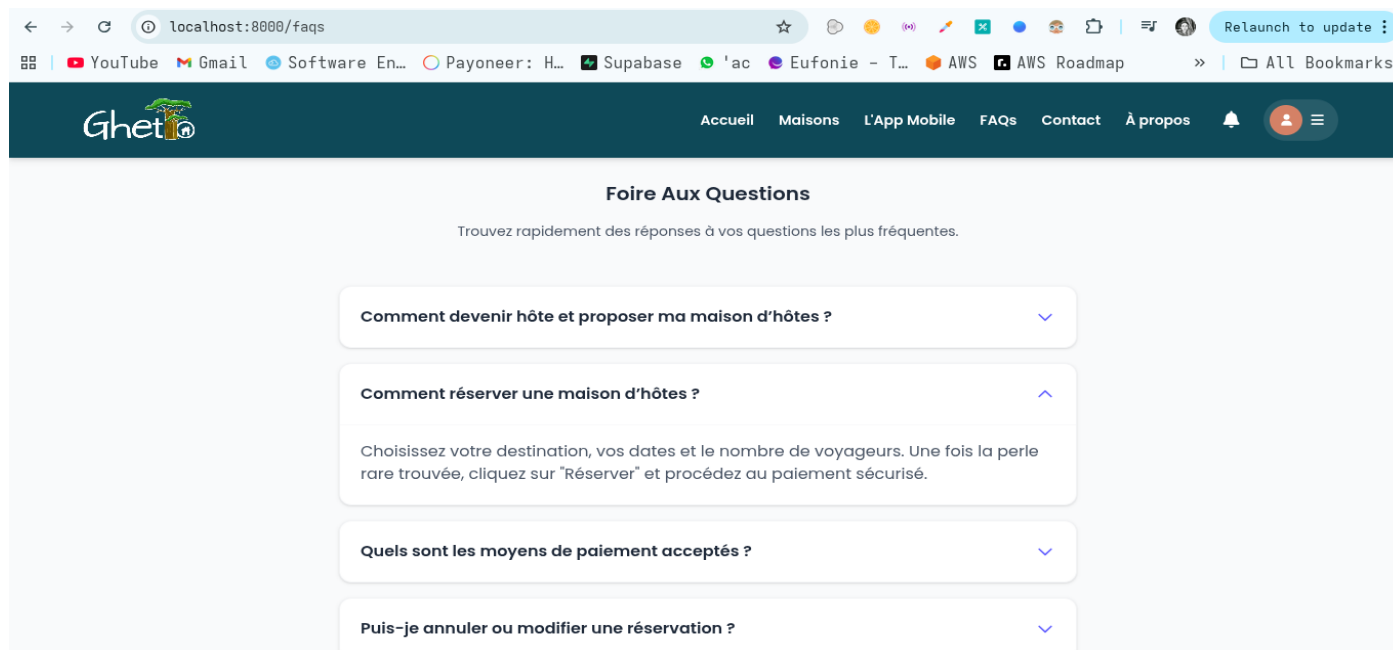


FIGURE 31: INTERFACE DE LA PAGE FAQ

CONCLUSION GÉNÉRALE

Au terme de ce travail, on a conçu et réalisé l'application web **Ghetto**, une plateforme dédiée à la réservation de maisons d'hôtes, répondant aux besoins des utilisateurs, des propriétaires et de l'administration de la plateforme.

L'approche adoptée s'est appuyée sur la **méthode Merise**, permettant une modélisation rigoureuse du système à travers les modèles conceptuels et logiques, ainsi qu'une bonne compréhension des flux d'informations et des traitements. Cette démarche a facilité la structuration des données et la définition claire des rôles et responsabilités au sein de l'application.

L'architecture de l'application repose sur un modèle **client-serveur en trois couches**, garantissant une séparation efficace entre la couche présentation, la logique métier et la couche données. Le choix des technologies, notamment **Laravel**, **Livewire**, **MySQL** et **XAMPP** sous **Linux Manjaro**, a permis de développer une application performante, sécurisée et facilement maintenable.

Un point central du système est la mise en place de **deux tableaux de bord distincts**, adaptés aux différents profils d'administration :

- Le **Super Administrateur**, qui assure la supervision globale de la plateforme, la validation des nouveaux propriétaires et des maisons d'hôtes, ainsi que le contrôle de la qualité et de la sécurité du système ;
- Le **Propriétaire de maison d'hôte**, qui gère exclusivement ses propriétés, ses disponibilités et les réservations associées, tout en respectant les règles définies par l'administration centrale.

Cette séparation des rôles permet de renforcer la sécurité, d'améliorer l'organisation du système et d'offrir une expérience adaptée à chaque type d'utilisateur. Elle garantit également un meilleur contrôle des contenus publiés et une gestion efficace des réservations.

En perspective, plusieurs améliorations peuvent être envisagées, telles que l'intégration de moyens de paiement en ligne, l'ajout d'un système d'évaluation et de commentaires, ainsi que l'optimisation des performances et de la sécurité. Ces évolutions permettront de renforcer la fiabilité et l'attractivité de la plateforme.

Ainsi, l'application **Ghetto** constitue une solution complète, évolutive et adaptée à la gestion moderne des réservations de maisons d'hôtes.

BIBLIOGRAPHIE

- [1] TARDIEU, Hubert et al. **La Méthode MERISE : Principes et Outils**. Eyrolles, 1983, 250 pages.
[Consulté le : 1 Septembre 2025].
- [2] GARDARIN, Georges. **Bases de données : Les systèmes et leurs langages**. 6e édition. Eyrolles, 2010.
[Consulté le : 20 Septembre 2025].
- [3] DATE, C. J. **Introduction aux systèmes de bases de données**. 8e édition. Pearson Education, 2004.
[Consulté le : 18 Septembre 2025].
- [4] LARMAN, Craig. **Applying UML and Patterns**. 3rd Edition. Pearson Education, 2004.
[Consulté le : 22 Septembre 2025].

WEBOGRAPHIE

[5] LARAVEL.COM. **Documentation officielle Laravel.**

[en ligne]. Disponible sur : <https://laravel.com/docs>

[Consulté le : 1 Septembre 2025].

[6] LIVEWIRE.LARAVEL.COM. **Documentation officielle Laravel Livewire.**

[en ligne]. Disponible sur : <https://livewire.laravel.com>

[Consulté le : 5 Septembre 2025].

[7] MYSQL.COM. **MySQL Documentation.**

[en ligne]. Disponible sur : <https://dev.mysql.com/doc/>

[Consulté le : 10 Septembre 2025].

[8] APACHEFRIENDS.ORG. **XAMPP Documentation.**

[en ligne]. Disponible sur : <https://www.apachefriends.org>

[Consulté le : 12 Septembre 2025].

[9] W3SCHOOLS.COM. **PHP & MySQL Tutorial.**

[en ligne]. Disponible sur : <https://www.w3schools.com/php/>

[Consulté le : 15 Septembre 2025].

[10] GIT-SCM.COM. **Documentation officielle Git.**

[en ligne]. Disponible sur : <https://git-scm.com/doc>

[Consulté le : 18 Septembre 2025].

GLOSSAIRE

Architecture client–serveur : modèle informatique dans lequel les traitements sont répartis entre un client (navigateur) et un serveur (application et base de données).

Base de données : ensemble structuré de données permettant leur stockage, leur gestion et leur exploitation.

Framework : environnement logiciel fournissant des outils et des conventions facilitant le développement d'applications.

Laravel : framework PHP open source basé sur l'architecture MVC, utilisé pour le développement d'applications web modernes.

Livewire : bibliothèque Laravel permettant de créer des interfaces dynamiques sans utiliser directement JavaScript.

Méthode MERISE : méthode d'analyse et de conception des systèmes d'information basée sur la séparation des données et des traitements.

Modèle Conceptuel de Données (MCD) : représentation des données indépendamment des contraintes techniques.

Réservation : action par laquelle un utilisateur demande la disponibilité d'une maison d'hôte pour une période donnée.

SGBD : Système de Gestion de Base de Données permettant la manipulation et la gestion des données.

RESUME

Dans le cadre de mon projet académique en Informatique à l'École Nationale d'Informatique (ENI) de Fianarantsoa, j'ai conçu et développé une application web de réservation de maisons d'hôtes dénommée **Ghetto**.

L'objectif principal du projet est de faciliter la mise en relation entre les voyageurs et les propriétaires de maisons d'hôtes, tout en assurant une gestion efficace des réservations. L'analyse et la conception du système ont été réalisées en s'appuyant sur la **méthode MERISE**, permettant une modélisation rigoureuse des données et des traitements.

La solution technique repose sur une architecture **client-serveur**, utilisant le framework **Laravel** avec **Livewire** pour le développement de l'application, et **MySQL** comme système de gestion de base de données. L'environnement de développement est basé sur **Linux Manjaro**, **XAMPP** et **Visual Studio Code**.

L'application distingue clairement les rôles du **Super Administrateur** et du **Propriétaire**, garantissant une gestion sécurisée et structurée des maisons d'hôtes et des réservations. Ce projet a permis de consolider les compétences acquises en analyse, conception et développement d'applications web.

Mots-clés : Réservation, Maison d'hôte, MERISE, Laravel, Livewire, MySQL.

ABSTRACT

As part of an academic project in Computer Science at the École Nationale d'Informatique (ENI) of Fianarantsoa, a web application for guest house reservations called **Ghetto** was designed and developed.

The main objective of the project is to facilitate the connection between travelers and guest house owners while ensuring efficient reservation management. The system analysis and design were carried out using the **MERISE methodology**, allowing a rigorous modeling of data and processes.

The technical solution is based on a **client-server architecture**, using the **Laravel** framework with **Livewire** for application development, and **MySQL** as the database management system. The development environment relies on **Linux Manjaro**, **XAMPP**, and **Visual Studio Code**.

The application clearly distinguishes the roles of **Super Administrator** and **Property Owner**, ensuring secure and structured management of guest houses and reservations. This project strengthened practical skills in analysis, design, and web application development.

Keywords: Reservation, Guest house, MERISE, Laravel, Livewire, MySQL.

