

Université Moulay Ismail
Faculté des sciences et techniques Er-Rachidia
Département d'informatique
Spécialité : Génie logiciel

CONCEPTION ET RÉALISATION D'UNE DAPP POUR LES DESTINATIONS TOURISTIQUES

Projet présenté par :
ASSINI ISSAM
KARDI ABDOSSATTAR

Encadré par :

prof. OUALLA MOHAMED

Soutenu le : 10/07/2024

Jury :
Pr : AIT KHOUYA YOUSSEF
Pr : LAKHAROUT JIHANE
Pr : OUALLA MOHAMED

TABLE DES MATIERES

Dédicace	7
Remerciements	8
Résumé	9
Abstract	10
Liste des abréviations	11
Introduction générale	12
Chapitre 1: Introduction et Présentation du projet	13
1.1 Introduction	13
1.2 Cadre du projet	13
1.2.1 Présentation du projet	13
1.2.2 Planning du projet	13
1.3 Problématique	14
1.4 Objectifs du projet	14
1.5 La démarche suivie	15
1.5.1 Cycle de vie du projet	15
1.5.2 Approche, méthodologie et planning du projet	15
1.6 Explication des Phases du Projet	16
1.6.1 Phase de planification	16
1.6.2 Phase de conception	16
1.6.3 Phase de développement	17
1.6.4 Phase de mise en service	17
1.6.5 Phase d'exploitation	17
1.7 Conclusion	18
Chapitre 2: Web3.0 et dApp	19
2.1 Définition web3.0	19
2.2 Définition dApp	19
2.3 Notions de base	19
2.3.1 Blockchain	19
2.3.2 Smart Contracts	20
2.3.3 Truffle	20
2.3.4 Ganache	20
2.3.5 MetaMask	20
2.4 Avantages de Web 3.0	20
2.5 Avantages de dApp	21
2.6 Conclusion	21
Chapitre 3: Analyse et spécification des besoins	22
3.1 Introduction	22
3.2 Analyse des besoins	22
3.2.1 Identification des acteurs	22
3.2.2 Spécification des besoins fonctionnels	23
3.2.3 Spécification des besoins non fonctionnels	24

3.3	La méthodologie d'analyse	24
3.3.1	Modèle en cascade	24
3.3.2	Étude de faisabilité	24
3.3.3	Analyse des exigences	25
3.3.4	Conception du système	26
3.4	Conclusion	26
Chapitre 4:	Conception de l'application	27
4.1	Introduction	27
4.2	Base de données	27
4.2.1	Diagramme de classes	27
4.3	Interface Graphique	28
4.4	Application	29
4.4.1	Diagramme de cas d'utilisation	29
4.4.2	Description des Cas d'utilisation	30
4.4.3	Diagrammes de Séquences	31
4.5	Conclusion	33
Chapitre 5:	Réalisation de l'application	34
5.1	Introduction	34
5.2	Langages	34
5.2.1	JavaScript	34
5.2.2	Python	34
5.2.3	MySQL	35
5.2.4	Solidity	35
5.3	Frameworks	36
5.3.1	ReactJS	36
5.3.2	Django	36
5.3.3	Rest	37
5.3.4	Truffle	37
5.4	Outils	38
5.4.1	CSS	38
5.4.2	Latex	38
5.4.3	Visual Studio Code	39
5.4.4	MetaMask	39
5.4.5	Ganache	40
5.5	Base de Données	40
5.6	Interfaces Graphiques de l'application	41
5.6.1	Page d'accueil	41
5.6.2	Page d'inscription	41
5.6.3	Page de connexion	42
5.6.4	Planification de voyage	42
5.6.5	Exploration des meilleures destinations	43
5.6.6	Réservation	44
5.6.7	Le paiement	45
5.6.8	Footer	46
5.7	Code	46
5.7.1	Authentification	46
5.7.2	Le paiement	47
5.7.3	Smart Contract	48

5.8	Obstacles et solutions	49
5.8.1	Obstacles techniques	49
5.8.2	Obstacles fonctionnels	50
5.8.3	Obstacles juridiques	51
5.8.4	Obstacles organisationnels	53
5.8.5	Obstacles de Ressources	54
5.9	Conclusion	55
	Références	57
	Documentation	58

Table des figures

1.1	Modèle de cycle de vie en cascade	15
1.2	GANTT	16
3.1	Modèle en cascade	25
4.1	Diagramme de classes	28
4.2	Arbre d'interface graphique du front-office	29
4.3	Diagramme de cas d'utilisation	30
4.4	Diagrammes de Séquences	32
4.5	Diagrammes de séquences de réservation	33
5.1	JavaScript	34
5.2	python	34
5.3	MySQL	35
5.4	Solidity	35
5.5	ReactJS	36
5.6	Django	36
5.7	Rest	37
5.8	Truffle	37
5.9	CSS	38
5.10	Latex	38
5.11	Visual studio code	39
5.12	MetaMask	39
5.13	Ganache	40
5.14	Base de données	40
5.15	Liste des utilisateurs	41
5.16	La page d'accueil	41
5.17	Page d'inscription	42
5.18	Page de connexion	42
5.19	Planification de voyage	43
5.20	Most visited destinations 1	43
5.21	Most visited destinations 2	44
5.22	Most visited destinations 3	44
5.23	Reservation	45
5.24	Reservation	45
5.25	Paiement	46
5.26	Footer	46
5.27	Authentification	47
5.28	PaymentContract	47

5.29 HandlePay	48
5.30 Smart Contract	48
5.31 Gestion de paiement	49

Dédicace :

Nous dédions notre travail à :

- Nos chers parents et nos familles que nous admirons profondément et que nous remercions pour leurs sacrifices, leur compréhension et leur soutien indéfectible.
- Tous nos amis, partisans et collaborateurs Ceux qui ont aidés à réaliser ce projet.
- Tous ceux qui ont aidé à mettre ce petit travail en action, qu'ils soient proches ou éloignés.

Merci a tous.

Remerciements

C'est avec grand plaisir que nous réservons ces quelques lignes en signe de gratitude et de reconnaissance sincères pour tous ceux qui, qu'ils soient proches ou éloignés, ont contribué à l'achèvement et au début de ce travail. Tout d'abord, nous tenons à exprimer notre gratitude à notre campeur, M.OUALLA MOHAMMED, pour ses qualités professionnelles et humaines, son sérieux, sa disponibilité, ses conseils inestimables et son aide tout au long du processus d'écriture. Nous remercions également les membres du jury MAIT KHOUYA YOUSSEF et Mme LAKHAROUI JIHANE d'avoir pris le temps de lire et d'évaluer cet essai. Enfin, nous tenons à exprimer notre sincère gratitude à tout le personnel enseignant de la Faculté des sciences et de la technologie d'Er-Rachidia, en particulier à nos professeurs, pour nous avoir inspirés à travailler et à mettre leurs expériences et leurs connaissances à notre disposition.

Résumé

Le projet de fin d'études consiste en la création d'un site web touristique innovant, basé sur la technologie Web3. L'objectif principal est de tirer parti des capacités décentralisées et sécurisées de Web3 pour offrir aux utilisateurs une expérience unique et enrichissante dans la découverte de destinations touristiques.

Pour ce projet, plusieurs technologies avancées ont été utilisées. Web3 permet l'intégration de la blockchain, garantissant la sécurité des transactions et la transparence des données. Les smart contracts sont employés pour gérer les réservations et les paiements de manière automatisée et sécurisée. Enfin, des frameworks de développement web comme HTML5, CSS3, JavaScript, ainsi que React , ont été utilisés pour créer une interface utilisateur dynamique et réactive.

Le site web propose plusieurs fonctionnalités principales. Les utilisateurs peuvent explorer des destinations touristiques avec des informations détaillées, des photos, des vidéos et des avis. Un système de réservation décentralisé permet de réserver des hébergements et des activités directement via des smart contracts, éliminant ainsi les intermédiaires. Les paiements en crypto-monnaies assurent des transactions rapides et sécurisées.

L'utilisation de la technologie Web3 apporte de nombreux avantages. La sécurité et la transparence des transactions et des données utilisateurs sont assurées par la blockchain. La décentralisation élimine les points de défaillance uniques, rendant le système plus robuste. Les utilisateurs ont également un contrôle total sur leurs données personnelles, renforçant la confiance et la confidentialité.

En conclusion, ce projet démontre le potentiel du Web3 dans le secteur du tourisme, offrant une nouvelle dimension de sécurité, de transparence et de décentralisation. En intégrant des technologies de pointe, le site web vise à révolutionner la manière dont les utilisateurs planifient et réservent leurs voyages, tout en bénéficiant d'une expérience utilisateur améliorée.

Abstract

The final year project involves creating an innovative tourist website based on Web3 technology. The main objective is to leverage the decentralized and secure capabilities of Web3 to provide users with a unique and enriching experience in discovering tourist destinations.

For this project, several advanced technologies have been used. Web3 enables the integration of blockchain, ensuring transaction security and data transparency. Smart contracts are employed to manage bookings and payments in an automated and secure manner. Additionally, web development frameworks like HTML5, CSS3, JavaScript, and React were used to create a dynamic and responsive user interface.

The website offers several key features. Users can explore tourist destinations with detailed information, photos, videos, and reviews. Interactive maps allow for the location of points of interest. A decentralized booking system enables booking accommodations and activities directly through smart contracts, thereby eliminating intermediaries. Cryptocurrency payments ensure fast and secure transactions. Furthermore, user reviews and ratings are stored on the blockchain, guaranteeing their authenticity and preventing fake comments. A loyalty program rewards active users with tokens, which can be used for discounts or exclusive services.

The use of Web3 technology brings numerous benefits. The security and transparency of transactions and user data are ensured by the blockchain. Decentralization eliminates single points of failure, making the system more robust. Users also have full control over their personal data, enhancing trust and privacy.

In conclusion, this project demonstrates the potential of Web3 in the tourism sector, offering a new dimension of security, transparency, and decentralization. By integrating cutting-edge technologies, the website aims to revolutionize the way users plan and book their trips while providing an enhanced user experience.

Liste des abréviations

IA : Intelligence Artificielle

ML : Machine Learning

IoT : Internet of Things

UML : Unified Modeling Language

EVM : Machine virtuelle

CSS : Cascading Style Sheets

HTML : HyperText Markup Language

BAT : Basic Attention Token

Introduction générale

Dans le cadre de la réalisation de notre projet de site web, nous avons choisi d'exploiter les avantages offerts par la technologie Web3. Web3, souvent considéré comme la prochaine évolution du web, permet une décentralisation accrue, une sécurité renforcée grâce à la blockchain, et l'intégration de cryptomonnaies pour les transactions. Ces caractéristiques répondent parfaitement aux besoins d'un site web moderne et sécurisé, et c'est pourquoi nous avons opté pour cette technologie pour développer notre site de réservation touristique.

Notre site de réservation touristique se distingue par son offre de voyages organisés vers des destinations marocaines soigneusement sélectionnées. En plus de garantir une hospitalité exemplaire, notre plateforme propose une variété d'activités pour enrichir l'expérience de nos utilisateurs. L'objectif principal est de créer un site web attrayant et intuitif, capable d'attirer un grand nombre de visiteurs et de faciliter les réservations en ligne via le paiement en cryptomonnaies.

Ce rapport est structuré en plusieurs chapitres pour une présentation claire et détaillée de notre projet. Le premier chapitre est consacré à la présentation globale du projet. Le deuxième chapitre explore en profondeur les concepts de Web3 et des applications décentralisées (DApp). Dans le troisième chapitre, une analyse détaillée et une spécification des besoins seront abordées. Le quatrième chapitre portera sur la conception de l'application, tandis que le cinquième chapitre détaillera la réalisation concrète de celle-ci.

Chapitre 1

Introduction et Présentation du projet

1.1 Introduction

Le Web 3.0 représente une nouvelle ère de l'Internet qui mise sur la décentralisation, la transparence et la sécurité, et qui promet de transformer plusieurs secteurs, y compris l'industrie des sites de réservation touristiques. Contrairement aux évolutions précédentes du Web, le Web 3.0 s'appuie fortement sur des technologies décentralisées comme la blockchain, les applications peer-to-peer (P2P) et les contrats intelligents. Ces innovations apportent des améliorations significatives en matière de transparence, d'efficacité des transactions, de sécurité des données et de confiance des utilisateurs.[1]

1.2 Cadre du projet

1.2.1 Présentation du projet

Le projet entre dans le cadre de préparation du projet de fin d'études pour l'obtention de la licence Sciences et techniques en Génie Logiciel (GL) de la faculté des sciences et techniques Er-Rachidia (FSTE).

1.2.2 Planning du projet

Avant de se pencher sur l'étude détaillée, une phase de planification s'avère primordiale afin d'assurer la bonne conduite du projet puisqu'elle participe au succès de la démarche. De ce fait, la nécessité impérieuse de maîtriser le projet implique la mise en place d'une organisation adaptée en ce qui concerne :

- La définition des tâches nécessaires.
- L'ordonnancement des tâches.
- La définition des rôles des acteurs et intervenants.

Dans notre projet, nous avons estimé de réaliser notre site web dans une durée approximative de 3 mois. Le tableau ci-dessous montre le planning que nous avons adapté pour mener à bien notre réalisation des différents besoins d'estimation qui se font valoir au niveau du projet, au niveau de la phase et au niveau des tâches.

1.3 Problématique

Pour un projet de tourisme en Web3, la problématique pourrait être formulée de différentes manières en fonction des aspects spécifiques que vous souhaitez explorer. Voici quelques suggestions :

"Comment intégrer efficacement la technologie blockchain et les fonctionnalités décentralisées dans l'industrie du tourisme pour améliorer l'expérience des voyageurs tout en garantissant la sécurité et la confidentialité des données ?" "Quels sont les défis et les opportunités de l'adoption de la technologie Web3 dans le secteur du tourisme, et comment pouvons-nous surmonter les obstacles pour créer une plateforme touristique décentralisée réussie et durable ?" "Comment créer un écosystème touristique décentralisé qui encourage la participation active des communautés locales, tout en offrant une expérience enrichissante et authentique aux voyageurs grâce à la technologie Web3 ?"

1.4 Objectifs du projet

L'application web permet d'atteindre les objectifs suivants :

- Développer une interface web 3.0 intuitive offrant une gamme complète de services touristiques, allant de la réservation d'hébergement à la découverte de lieux hors des sentiers battus. Par exemple, permettre aux utilisateurs de réserver des hébergements dans des riads traditionnels ou des séjours chez l'habitant.
- Utiliser les contrats intelligents pour automatiser les processus de réservation, garantissant ainsi des transactions sécurisées et transparentes pour les utilisateurs. Par exemple, implémenter des contrats intelligents pour la réservation d'activités touristiques et la gestion des paiements.
- Mettre en place un système de stockage décentralisé des données, assurant ainsi la sécurité et la confidentialité des informations personnelles des utilisateurs. Par exemple, utiliser la technologie IPFS pour stocker des données de manière décentralisée et sécurisée.
- Intégrer une identité numérique décentralisée pour permettre aux utilisateurs de contrôler leurs données personnelles et leur historique de voyages de manière sécurisée. Par exemple, mettre en œuvre des solutions d'authentification décentralisée basées sur la blockchain.
- Créer un système de réputation décentralisé basé sur la blockchain pour garantir l'authenticité des avis laissés par les utilisateurs. Par exemple, utiliser des tokens

non fongibles (NFT) pour attribuer des récompenses aux utilisateurs les plus actifs et les plus fiables.

1.5 La démarche suivie

1.5.1 Cycle de vie du projet

Le cycle de vie d'un projet est la succession des différentes étapes de son existence depuis sa conception jusqu'à son terme. Il va de soi que chaque projet est différent mais les phases par lesquelles il passe sont généralement communes. Et la réussite de chacune d'entre elles garantit sa réalisation et l'atteinte des objectifs fixés. Le cycle de vie d'un projet peut être mis en place pour le lancement d'un produit par exemple. Ce modèle entre dans ce que l'on appelle la gestion du projet.

1.5.2 Approche, méthodologie et planning du projet

Le modèle de cycle de vie en cascade comporte 6 phases : spécification des besoins, analyse, conception, implémentation, enfin mise en place et exploitation. Chacune de ces phases doit produire un ou plusieurs livrables définis à l'avance et à une date d'échéance fixée. On ne peut passer d'une étape à l'autre que lorsque les livrables de l'étape en cours sont jugés satisfaisants. Si tout se passe bien on passe à la phase suivante.

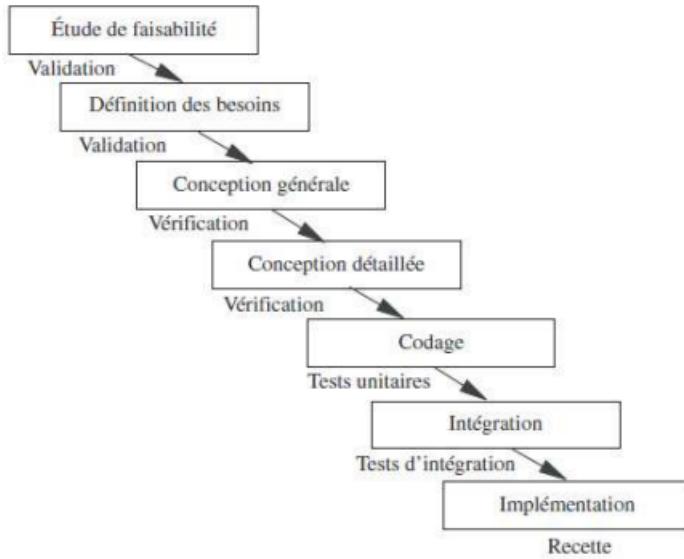


FIGURE 1.1 – Modèle de cycle de vie en cascade.

Le principe de ce modèle est qu'avec toute décomposition doit être décrite la recomposition et que toute description d'un composant est accompagnée de tests qui permettent de s'assurer qu'il correspond à sa description. Ceci rend explicite la préparation des dernières phases (validation-vérification) par les premières (construction du logiciel), et permet ainsi d'éviter un écueil bien connu de la spécification du logiciel : énoncer une

propriété qu'il est impossible de vérifier objectivement après la réalisation.

L'ensemble des activités de ce modèle suivant le temps est :

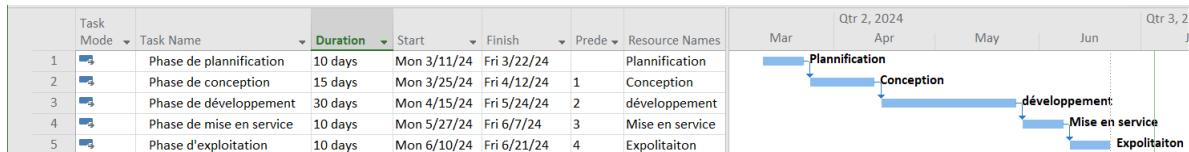


FIGURE 1.2 – GANTT

1.6 Explication des Phases du Projet

1.6.1 Phase de planification

► Objectif

Définir les grandes lignes du projet et établir une feuille de route détaillée.

► Activités

- **Établissement des objectifs** : Identifier les buts principaux du projet, comme les fonctionnalités du site, les attentes des utilisateurs, et les résultats attendus.
- **Évaluation des ressources** : Déterminer les ressources nécessaires, y compris les ressources humaines, financières, et technologiques.
- **Planification du calendrier** : Développer un calendrier de projet avec des jalons et des dates clés.
- **Risques et contingences** : Identifier les risques potentiels et planifier des mesures de contingence pour les atténuer.

1.6.2 Phase de conception

► Objectif

Créer une conception détaillée de l'application et définir les spécifications techniques.

► Activités

- **Architecture système** : Développer l'architecture globale de l'application, y compris la structure de la base de données et les composants principaux du système.
- **Wireframes et maquettes** : Créer des esquisses visuelles de l'interface utilisateur pour visualiser la disposition et l'apparence du site.
- **Spécifications techniques** : Rédiger des documents détaillant les spécifications techniques, y compris les exigences fonctionnelles et non fonctionnelles.
- **Sécurité et conformité** : Planifier les mesures de sécurité et garantir que le site sera conforme aux régulations pertinentes.

1.6.3 Phase de développement

► Objectif

Construire et développer les fonctionnalités de l'application conformément aux spécifications définies.

► Activités

- **Développement front-end** : Programmer l'interface utilisateur en utilisant des technologies web modernes.
- **Développement back-end** : Créer la logique serveur et les API nécessaires pour le fonctionnement de l'application.
- **Intégration de la blockchain** : Mettre en œuvre les aspects blockchain et cryptomonnaie pour les transactions et la sécurité.
- **Tests unitaires et intégration continue** : Tester les composants individuels et intégrer des tests automatisés pour assurer la qualité du code.

1.6.4 Phase de mise en service

► Objectif

Préparer l'application pour le lancement public et assurer une transition en douceur.

► Activités

- **Tests bêta** : Lancer une version bêta du site pour un groupe limité d'utilisateurs afin de recueillir des feedbacks et identifier les bogues.
- **Optimisation** : Affiner les performances et l'efficacité de l'application basée sur les résultats des tests.
- **Documentation** : Finaliser la documentation utilisateur et technique pour soutenir le déploiement.
- **Préparation du déploiement** : Mettre en place les environnements de production et de pré-production et effectuer les tests finaux.

1.6.5 Phase d'exploitation

► Objectif

Assurer le suivi et le maintien de l'application après son lancement.

► Activités

- **Surveillance et maintenance** : Surveiller les performances du site et effectuer les mises à jour et la maintenance nécessaires.
- **Support utilisateur** : Offrir un support technique aux utilisateurs pour résoudre les problèmes et répondre aux questions.
- **Analyse des performances** : Collecter et analyser les données d'utilisation pour identifier les améliorations possibles.
- **Améliorations continues** : Mettre en œuvre des améliorations basées sur les retours des utilisateurs et les analyses de performance.

1.7 Conclusion

Dans ce chapitre, nous avons cité les différents systèmes d'exploitations mobiles dans un premier temps. Ensuite, nous avons décrit le contexte général dans lequel s'inscrit le projet. A la lumière des problématiques dégagées. Après on a présenté la démarche que nous avons suivie pour arriver à notre but et cela par la présentation du processus de développement ou cycle de vie du projet que nous avons adopté en l'occurrence le modèle en cascades, et le pilotage de l'ensemble de ces activités représentées ici par le diagramme de GANTT.

Dans le chapitre suivant nous préciserons les besoins fonctionnels et non fonctionnels.

Chapitre 2

Web3.0 et dApp

2.1 Définition web3.0

L'objectif du Web 3.0 est de décentraliser ces données en passant par un système de « Peer to Peer » : les données ne transitent plus par des serveurs qui appartiennent aux marques et entités, mais sont partagées entre tous les utilisateurs qui naviguent sur internet, et chacun stocke des bouts d'informations. Ces données sont ensuite reconstituées avec une chaîne de caractères cryptés.

2.2 Définition dApp

Une DApp est une application décentralisée. On l'aura donc compris, Les DApps s'opposent aux applications centralisées que l'on connaît et utilise majoritairement aujourd'hui. Pour fonctionner, elles sont constituées de [2] :

- ▶ Un (ou plusieurs) contrat autonome (plus connu sous le nom de Smart Contract) : c'est la structure de la DApp. Les contrats sont construits sur le protocole Peer to Peer qui ne dépend pas d'un tiers de confiance
- ▶ Une interface utilisateur
- ▶ Un modèle distribué de stockage de données

2.3 Notions de base

2.3.1 Blockchain

Une blockchain est un registre, une grande base de données qui a la particularité d'être partagée simultanément avec tous ses utilisateurs, tous également détenteurs de ce registre, et qui ont également toute la capacité d'y inscrire des données, selon des règles spécifiques fixées par un protocole informatique très bien sécurisé grâce à la cryptographie.

À quoi cela sert-il ?

Les blockchains se sont historiquement développées pour soutenir des transactions réalisées sous une nouvelle forme de moyens de paiement, appelées cryptomonnaies et qui ont comme caractéristique principale de n'être gérées par aucun organisme centralisateur

(comme une banque centrale) et d'être internationales : les bitcoins ou les ethers sont les plus connues.[3]

2.3.2 Smart Contracts

Sont des codes ou des applications écrits pour interagir avec l'écosystème Ethereum. Ces contrats contiennent des conditions prédéterminées que les parties concluant le contrat doivent remplir. Le contrat entre en vigueur uniquement lorsque les conditions définies dans le contrat intelligent sont remplies. Lancé en 2015 sur ETH, les contrats intelligents favorisent les calculs complexes dans un mode sans confiance, garantissant la réalisation des termes des deux parties. Aujourd'hui, le marché est inondé de centaines de DAPP dans divers domaines tels que les jeux, le commerce de détail, le défi, etc.[4]

2.3.3 Truffle

Truffle est un environnement de développement, un cadre de test et un pipeline d'actifs pour Ethereum. Il permet aux développeurs d'écrire, de déployer et de tester les contrats intelligents de manière simplifiée. Truffle fournit des outils intégrés pour gérer des projets de contrats intelligents complexes, ce qui permet aux développeurs de déboguer, tester et déployer leur code. Avec TruffLe, les développeurs peuvent automatiser les flux de travail de développement et réduire le temps et les efforts nécessaires pour créer des applications décentralisées.

2.3.4 Ganache

Ganache, quant à elle, est une blockchain personnelle pour le développement d'Ethereum. Il permet aux développeurs de créer un environnement blockchain simulé pour tester leurs contrats intelligents et leurs applications décentralisées. Ganache fournit une interface utilisateur graphique qui affiche les informations sur les transactions, les événements de contrat et les sorties de journaux, permettant aux développeurs de surveiller et de déboguer plus facilement leur code. En utilisant Ganache, les développeurs peuvent simuler diverses conditions de réseau et tester leurs applications dans un environnement contrôlé avant de les déployer sur le réseau principal Ethereum.

2.3.5 MetaMask

Metamask est une extension de navigateur populaire et établie qui fonctionne comme un portefeuille de crypto-monnaie qui se connecte à la blockchain Ethereum. Metamask permet aux utilisateurs d'interagir avec l'écosystème Ethereum, qui héberge un vaste univers d'applications décentralisées (DAPP), sans avoir à télécharger l'intégralité de la blockchain sur leur appareil. En tant que tel, c'est l'une des meilleures solutions de portefeuille Ethereum pour un accès facile à des échanges décentralisés (DEX), des plateformes de jeu, des sites de jeu et de nombreuses autres applications.[5]

2.4 Avantages de Web 3.0

- ❶ Aucun point de contrôle central

-
- ② Interconnexion accrue des informations
 - ③ Ergonomie et facilité de navigation
 - ④ Création de nouvelles opportunités commerciales
 - ⑤ Gestion de la propriété intellectuelle

2.5 Avantages de dApp

Une application décentralisée s'exécute au sein d'une chaîne de blocs. Chaque nœud de la blockchain est traité par des utilisateurs et des développeurs indépendants. Ainsi réparties sur l'ensemble d'une chaîne de blocs décentralisée, les dApps bénéficient de leur sécurité et de leur résilience. Ainsi, l'application devient inarrêtable une fois activé.

2.6 Conclusion

Le rôle des DApp dans l'expansion du Web3 est indéniablement important car le premier contribue à établir la décentralisation dans l'écosystème Web 3.0. L'utilisation de la technologie des contrats intelligents a le potentiel d'étendre les capacités de travail des domaines industriels. C'est pourquoi les experts du secteur font de leur mieux pour concevoir des DApp qui renforcent le cluster décentralisé, ouvrant ainsi des opportunités de croissance pour les utilisateurs. Le secteur DApp se développe lentement avec la participation active des développeurs concevant des produits interopérables, accessibles et sécurisés. L'avenir des DApps semble prometteur, nous pourrons donc bientôt les voir rejoindre l'industrie grand public.

Chapitre 3

Analyse et spécification des besoins

3.1 Introduction

L'analyse et la spécification des exigences sont des étapes importantes dans tout projet de développement de logiciel ou de conception de système.

Il est important de comprendre les besoins et exigences des utilisateurs finaux, des parties prenantes et autres parties prenantes impliquées dans le projet pour définir les objectifs, les limites et les résultats attendus et identifier les risques et défis potentiels.

L'analyse et la spécification des exigences permettent de garantir que les ressources sont utilisées efficacement, que les délais sont respectés et que les résultats répondent aux attentes de toutes les parties prenantes.

3.2 Analyse des besoins

Cette section servira à préciser toutes les exigences liées à notre application. Afin de pouvoir clarifier les besoins des utilisateurs, nous proposerons des exigences fonctionnelles et des exigences non fonctionnelles.

3.2.1 Identification des acteurs

les Utilisateurs, Les fonctionnalités de l'acteur seront décrites dans le diagramme de cas d'utilisation :

► L'Utilisateur

Est un utilisateur ayant un compte du site web.

Il bénéficie de toutes les fonctionnalités de site web en mode connecté, il se considère aussi comme un utilisateur de site web qui trouvera des endroits à visiter. Il peut rechercher et réserver.

► L'administrateur

est responsable de la gestion globale du système de réservation de voyage.

► Metamask agit

comme un acteur secondaire où le paiement des utilisateurs est traité.

3.2.2 Spécification des besoins fonctionnels

Les besoins fonctionnels sont des exigences ou spécifications qui définissent ce que le système, doit faire pour satisfaire aux besoins des utilisateurs ou des clients.

❖ Créer un Compte

- ① Permet à l'utilisateur de créer un compte en fournissant les informations nécessaires.
- ② Vérification de l'unicité de l'adresse email et gestion des erreurs de création de compte.

❖ Se connecter

- ① Authentification sécurisée de l'utilisateur via identifiant et mot de passe.
- ② Gestion des sessions utilisateur pour accéder aux fonctionnalités personnalisées.

❖ Rechercher destination

- ① Interface de recherche permettant à l'utilisateur de trouver des destinations de voyage.
- ② Prise en charge de la recherche par mots-clés et filtres par catégorie.

❖ Filtrer par Catégorie

- ① Extension de la fonctionnalité de recherche pour permettre à l'utilisateur de filtrer les résultats par catégories spécifiques telles que lieu, type de voyage, attractions, etc.

❖ Afficher destination

- ① Affichage détaillé des informations sur une destination sélectionnée par l'utilisateur après une recherche

❖ Planifier un voyage

- ① Fonctionnalité principale permettant à l'utilisateur de créer et gérer un itinéraire de voyage.
- ② Intégration des options de recherche et d'affichage de destination dans le processus de planification.

❖ Ajouter une destination

- ① Gestion de l'ajout, de la modification et de la suppression de destinations dans l'itinéraire en cours de planification.
- ② Inclusion de destinations sélectionnées dans l'itinéraire de voyage de l'utilisateur.

❖ Effectuer une Réservation

- ① Processus permettant à l'utilisateur de réserver des services ou des hébergements pour les destinations de son itinéraire.

❖ Suppression Réservation

- ① Annulation d'une réservation existante selon les politiques de gestion des annulations.

② Gestion des notifications et des mises à jour de l'itinéraire suite à l'annulation.

❖ Effectuer une Réservation

① Processus permettant à l'utilisateur de réserver des services ou des hébergements pour les destinations de son itinéraire.

② Support des paiements traditionnels ainsi que des paiements par crypto-monnaies.

3.2.3 Spécification des besoins non fonctionnels

Les besoins non fonctionnels concernent les contraintes à prendre en considération pour mettre en place une solution adéquate aux attentes des utilisateurs.

Notre application doit nécessairement assurer ces besoins :

- **la rapidité de traitement** : En effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.
- **La performance** : Un site web doit être avant tout performant c'est-à-dire à travers ses fonctionnalités, répondre à toutes les exigences des usagers d'une manière optimale.
- **Fiabilité** : Disponibilité élevée du système avec un minimum de temps d'arrêt planifié pour les mises à jour et la maintenance.
- **La convivialité** : La future application doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.
- **Efficacité** : Le site Web doit être fonctionnel indépendamment de toutes circonstances pouvant entourer l'utilisateur.
- **Sécurité** : Authentification sécurisée avec gestion des sessions et chiffrement des données sensibles.
- **Documentation** : Documentation complète des fonctionnalités et des procédures pour les administrateurs système, les développeurs et les utilisateurs finaux.

3.3 La méthodologie d'analyse

3.3.1 Modèle en cascade

C'est le modèle le plus traditionnel de développement de SI. Il est basé sur une approche séquentielle et linéaire, où chaque étape doit être terminée avant de passer à la suivante. Les étapes du modèle en cascade sont : analyse des besoins, conception, développement, tests, mise en production et maintenance.[6]

3.3.2 Étude de faisabilité

❖ Analyse des besoins

① Fonctionnalités principales .

- Exploration des destinations et réservation en ligne.
- Services complets de A à Z incluant hébergement, restauration, et activités.

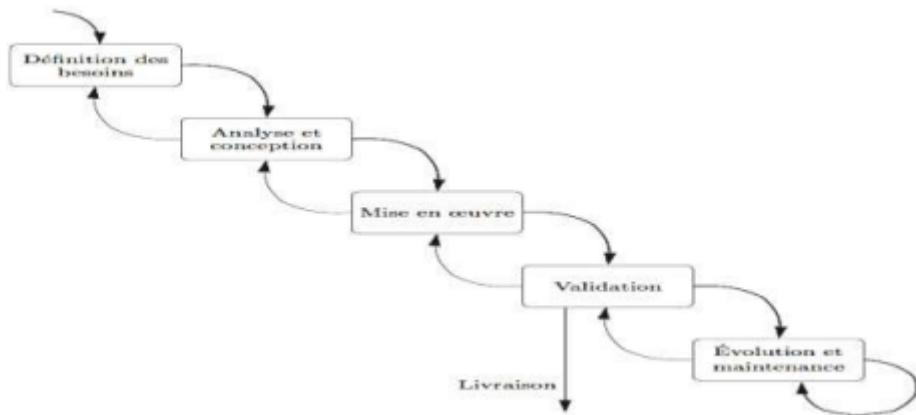


FIGURE 3.1 – Modèle en cascade

- Paiement par cryptomonnaies pour les réservations.
- ② Avantages attendus
 - Site moderne et sécurisé grâce à la technologie Web3.
 - Expérience utilisateur enrichie par une variété d'activités et de services.
 - Attirer un large public intéressé par les paiements en cryptomonnaies.
- ❖ Évaluation des technologies
 - ① Avantages de Web3 et de la blockchain
 - Décentralisation accrue, réduisant les risques de point unique de défaillance.
 - Sécurité renforcée grâce à la nature immuable de la blockchain.
 - Facilitation des transactions transfrontalières avec les cryptomonnaies.
 - ② Inconvénients potentiels
 - Complexité accrue du développement et des opérations.
 - Possible résistance des utilisateurs non familiers avec les cryptomonnaies.
 - Régulations légales et incertitudes autour des cryptomonnaies.
- ❖ Analyse des coûts
 - ① Budget : 0 DH pour le moment, car le projet est à l'état de test.
- ❖ Étude de marché
 - ① Concurrence : Analyse de tous les sites touristiques actuels.
 - ② Tendances et opportunités :
 - Croissance de l'intérêt pour les paiements en cryptomonnaies.
 - Demande accrue pour des expériences de voyage personnalisées et complètes.

3.3.3 Analyse des exigences

- ❖ Réunions avec les parties prenantes
 - ① Parties prenantes principales : Utilisateurs finaux, développeurs, et investisseurs potentiels.
 - ② Exigences : Attentes en termes de fonctionnalité, sécurité, performance, et convivialité.

-
- ❖ Rédaction des exigences fonctionnelles
 - ① Fonctionnalités du site
 - Moteur de recherche pour explorer des destinations marocaines.
 - Système de réservation incluant l'hébergement, la restauration, et les activités.
 - Intégration de paiements en cryptomonnaies.
 - Gestion des profils utilisateurs et historique des réservations.
 - ❖ édaction des exigences non fonctionnelles
 - ① Performance : Temps de chargement rapide et haute disponibilité.
 - ② Sécurité : Protection contre les attaques et sécurité des transactions en cryptomonnaies.
 - ③ Évolutivité : Capacité à gérer un nombre croissant d'utilisateurs et de transactions.
 - ④ Convivialité : Interface intuitive et facile à utiliser.

3.3.4 Conception du système

- ❖ Architecture globale
 - ① Front-end : Utilisation de framework moderne React pour une interface utilisateur réactive et dynamique.
 - ② Back-end : Utilisation de django rest framework pour la logique serveur et la gestion des API.
 - ③ Base de données : Utilisation de MySql pour sa flexibilité et sa capacité à gérer de grandes quantités de données non structurées.
- ❖ Conception de la base de données
 - ① Données stockées : Informations sur les destinations, les utilisateurs, les réservations, les transactions, etc.
 - ② Organisation : Schéma de base de données NoSQL avec des collections pour chaque type de données.
- ❖ Design de l'interface utilisateur
 - ① Pages principales : Page d'accueil, pages de recherche, pages de destination, page de réservation, profil utilisateur.
 - ② Expérience utilisateur : Navigation intuitive, design responsive, et accessibilité optimisée.

3.4 Conclusion

Dans ce chapitre, nous avons énuméré les différents besoins fonctionnels et non fonctionnels de notre application. Ensuite, nous avons fait une étude des différents cas d'utilisation de notre solution.

Ce chapitre a été d'une importance cruciale surtout pour la compréhension des besoins et attentes du client.

Chapitre 4

Conception de l'application

4.1 Introduction

La phase d'analyse et conception de l'application représente une étape primordiale dans le cycle de développement d'un projet. En effet, elle permet de mieux comprendre le travail demandé en dégageant les besoins des différents utilisateurs que le système doit accomplir.

4.2 Base de données

4.2.1 Diagramme de classes

Le diagramme de classe est une représentation abstraite des objets du système qui vont interagir ensemble pour la réalisation des cas d'utilisation. Nous pouvons distinguer entre quatre classes primordiales : " Utilisateur ", "Destination" et "Réservation", et la classe "Paiement".

- **Utilisateur** : Personne qui interagit avec l'application sans pour autant être connectée.
- **Destination** : Représente une destination touristique que les visiteurs peuvent consulter et réserver.
- **Réservation** : Représente la réservation d'une destination par un visiteur.

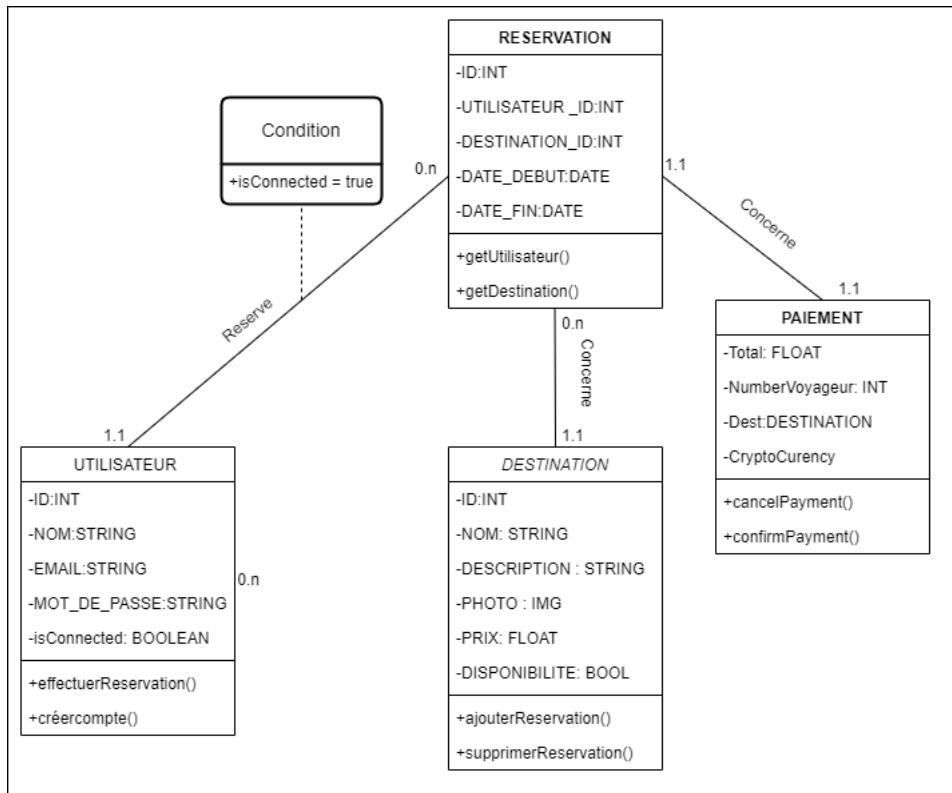


FIGURE 4.1 – Diagramme de classes

4.3 Interface Graphique

Le plan du site ou l’arborescence est le schéma qui organise les pages du site en rubriques et définit les différents niveaux de navigation. L’arbre ci-dessous montre le schéma suivi lors de l’exécution de notre application.

La figure 4.2 représentent l’arbre hiérarchique qui illustre notre interface graphique et les rubriques que comporte notre site web du côté client, ou ce que nous appelons front-office.

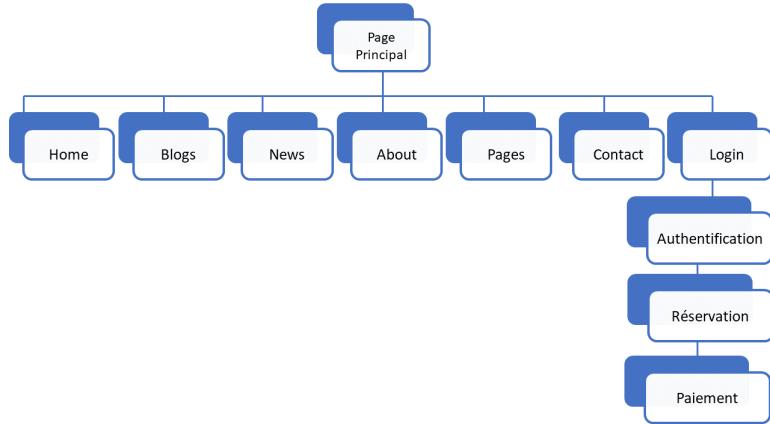


FIGURE 4.2 – Arbre d'interface graphique du front-office

Home : ceci est la première page qui s'ouvre une fois sur l'application.

Blogs : Descriptions des villes marocaines

News : Actualités sur le Maroc et points touristiques

About : Unified Modeling Language

Pages : Pages en relation avec celle-ci

Contact : Pour communiquer avec le staff

Login : La page qui contient le formulaire de connexion.

Réservation : Pour réserver

Paiement : Pour payer en crypto

4.4 Application

Après avoir présenté les entités du système, à travers le diagramme de classe, et déterminé le plan de notre site, nous allons procéder à la présentation des fonctions du système selon les besoins de l'utilisateur. Nous allons modéliser cela en utilisant les diagrammes UML suivants : le diagramme de cas d'utilisation et le diagramme de séquence.

4.4.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente ce qu'un système est supposé faire. Un cas d'utilisation représente une interaction entre un utilisateur (humain ou machine) et un système, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation.

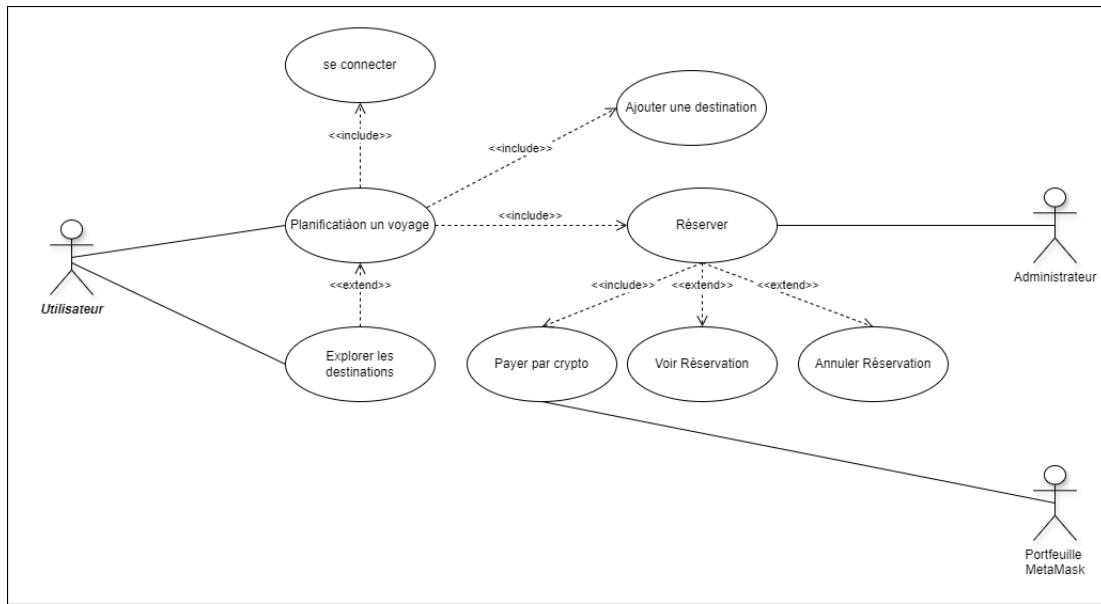


FIGURE 4.3 – Diagramme de cas d'utilisation

4.4.2 Description des Cas d'utilisation

- ▶ Créer un Compte
 - Ce cas d'utilisation étend le cas d'utilisation " se connecter".
 - Permet à l'utilisateur de créer un compte sur la plateforme.
- ▶ Se connecter
 - Permet à l'utilisateur de se connecter à son compte.
- ▶ Planification un voyage
 - Permet à l'utilisateur de planifier un voyage.
 - Ce cas d'utilisation inclut les cas d'utilisation "se connecter" et "Ajouter une destination".
 - Ce cas d'utilisation peut être étendu par les cas d'utilisation "Explorer les destinations" et "Réserver".
- ▶ Explorer les destinations
 - Permet à l'utilisateur de parcourir les différentes destinations disponibles.
- ▶ Ajouter une destination
 - Permet à l'administrateur d'ajouter une nouvelle destination à la plateforme.
- ▶ Réserver
 - Permet à l'utilisateur de réserver une destination.
 - Ce cas d'utilisation inclut les cas d'utilisation "Payer par crypto" et "Voir Réservation".
 - Ce cas d'utilisation peut être étendu par les cas d'utilisation "Annuler Réservation".
- ▶ Payer par crypto

-
- Permet à l'utilisateur de payer une réservation en utilisant des cryptomonnaies via un portefeuille MetaMask.
 - ▶ Voir Réservation
 - Permet à l'utilisateur de visualiser ses réservations.
 - ▶ Annuler Réservation
 - Permet à l'utilisateur d'annuler une réservation.

4.4.3 Diagrammes de Séquences

Le diagramme de séquence décrit l'aspect dynamique du système. Il modélise les interactions entre les objets ou entre utilisateur et objet, en mettant l'accent sur la chronologie des messages échangés. Dans ce qui suit, nous allons dresser les diagrammes de séquences des cas d'utilisation majeurs du diagramme de cas d'utilisation.

▶ Diagramme de séquence pour le Cas d'utilisation "s'authentifier"

Pour pouvoir exécuter une quelconque tâche (exploitation/ réservation), l'utilisateur doit saisir ses paramètres de connexion. Le système vérifie la disponibilité du login et du mot de passe, pour donner ensuite l'accès aux fonctionnalités de l'application. La figure 4.4 représente le diagramme de séquence pour le cas d'utilisation "s'authentifier".

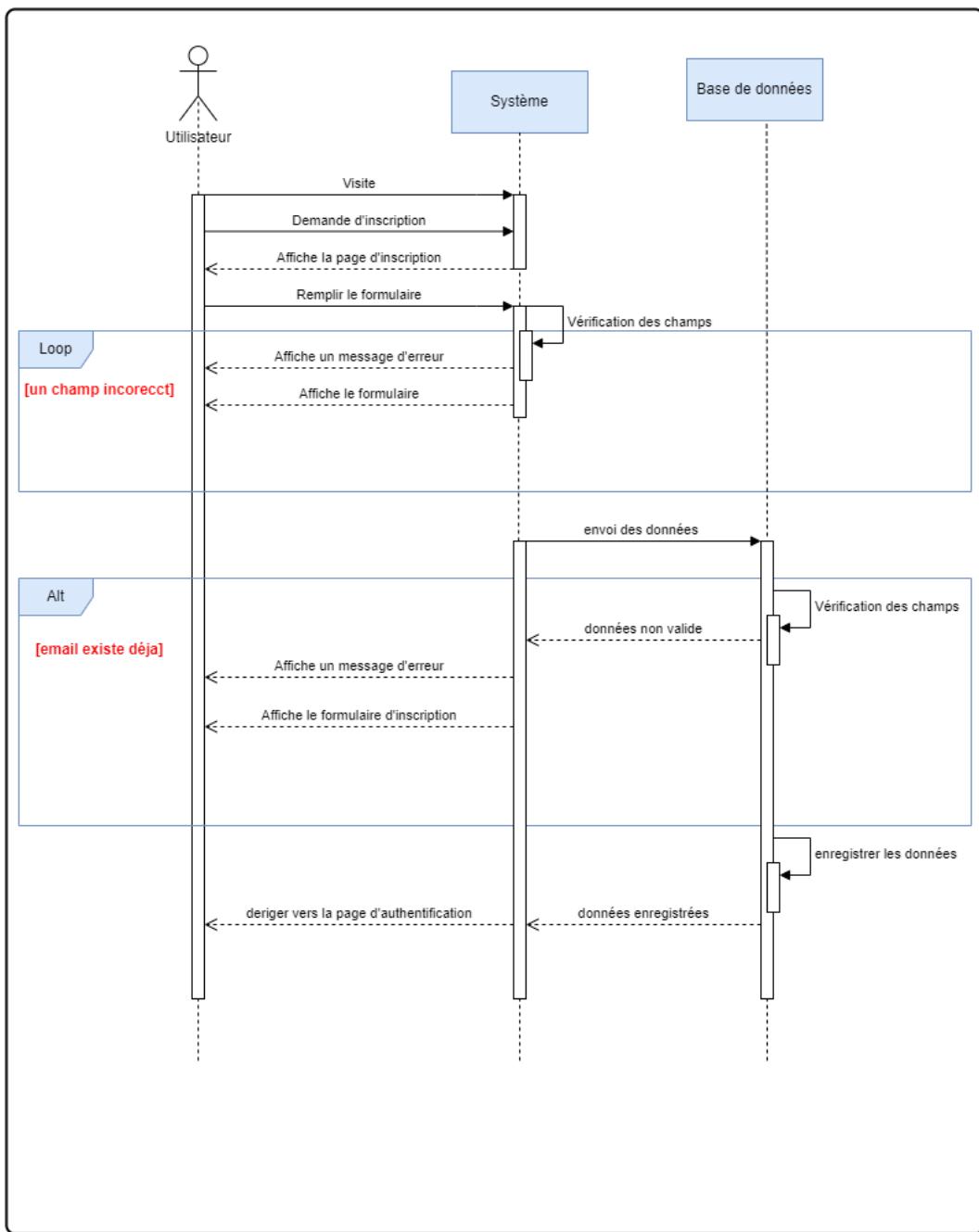


FIGURE 4.4 – Diagrammes de Séquences

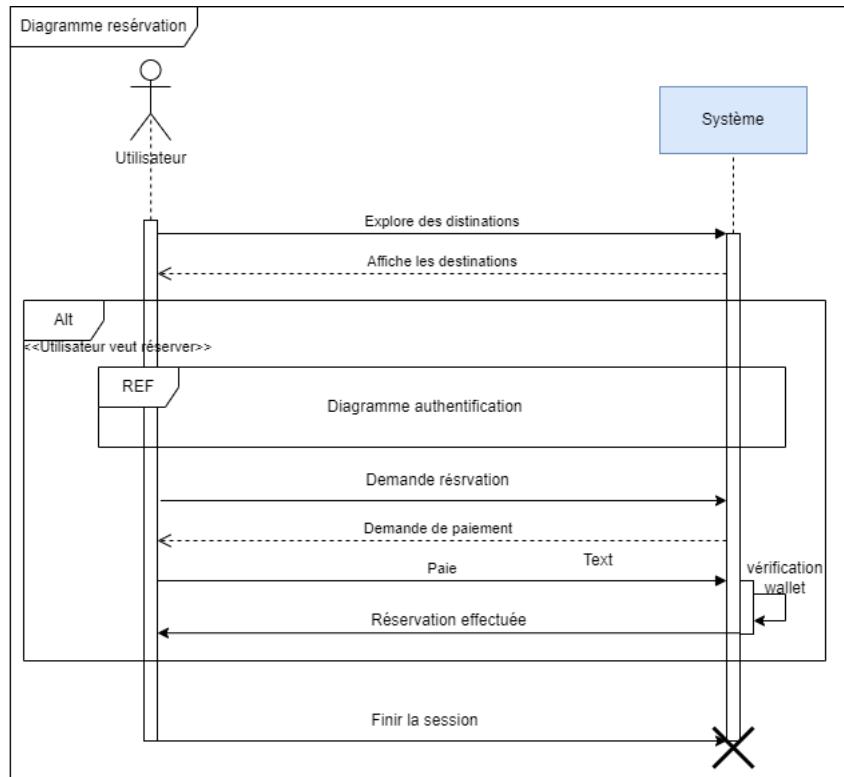


FIGURE 4.5 – Diagrammes de séquences de réservation

4.5 Conclusion

Dans ce chapitre, nous avons représenté la méthodologie de travail ainsi que les différents diagrammes UML avec lesquels nous avons modélisé l'application

Chapitre 5

Réalisation de l'application

5.1 Introduction

Dans ce chapitre nous allons présenter le choix des solutions techniques (langages et outils) adoptées pour la réalisation de ce site web.

5.2 Langages

5.2.1 JavaScript



FIGURE 5.1 – JavaScript

Le JavaScript est un langage de programmation web côté client qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du Navigateur et non du serveur web.

5.2.2 Python



FIGURE 5.2 – python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. Ainsi, développer du code avec python est plus rapide qu'avec d'autres langages.

5.2.3 MySQL



FIGURE 5.3 – MySQL

MySQL est système de gestion de bases de données ainsi qu'un serveur de bases de données relationnel où les données sont stockées dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. MySQL est très rapide, fiable et facile à utiliser. Ses possibilités de connexions, sa rapidité et sa sécurité font de lui un serveur hautement adapté à internet.

5.2.4 Solidity



FIGURE 5.4 – Solidity

Comme langage de programmation pour les contrats intelligents, Solidity permet aux développeurs de créer la logique métier derrière leurs dApps. En écrivant des contrats intelligents en Solidity, les développeurs peuvent définir les règles de fonctionnement de l'application, créer des transactions et interagir avec d'autres contrats ou utilisateurs sur la blockchain Ethereum.

5.3 Frameworks

5.3.1 ReactJS

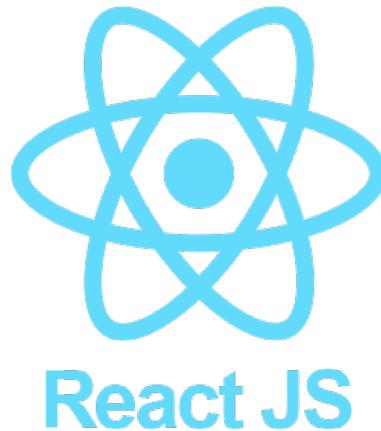


FIGURE 5.5 – ReactJS

est une bibliothèque JavaScript open-source qui est utilisée pour construire des interfaces utilisateur spécifiquement pour des applications d'une seule page. Elle est utilisée pour gérer la couche d'affichage des applications web et mobiles. React nous permet également de créer des composants d'interface utilisateur réutilisables. React a été créé par Jordan Walke, un ingénieur logiciel travaillant pour Facebook. React a été déployé pour la première fois sur le flux d'informations de Facebook en 2011 et sur Instagram en 2012.

5.3.2 Django



FIGURE 5.6 – Django

est un logiciel que vous pouvez utiliser pour développer des applications web rapidement et efficacement. La plupart des applications web ont plusieurs fonctions communes, telles que l'authentification, la récupération d'informations à partir d'une base de données et la gestion des cookies. Les développeurs doivent coder des fonctionnalités similaires dans chaque application web qu'ils écrivent. Django leur facilite la tâche en regroupant les différentes fonctions dans une vaste collection de modules réutilisables, appelés cadres d'applications web. Les développeurs utilisent le cadre web de Django pour organiser et écrire leur code de manière plus efficace et réduire considérablement le temps de développement web.

5.3.3 Rest



FIGURE 5.7 – Rest

est une extension puissante et flexible de Django, un framework web Python populaire. Il permet de créer rapidement et efficacement des API web RESTful (Representational State Transfer) robustes et bien structurées.

5.3.4 Truffle



FIGURE 5.8 – Truffle

Le composant majeur de Truffle Suite est Truffle. Il s'agit d'un environnement de développement spécialisé dans le développement de contrats intelligents. Avec son large éventail de fonctionnalités précieuses, le développement de dApps est beaucoup plus facile pour les développeurs, cette facilité résonne également avec son slogan « Smart Contracts Made Sweeter ».

Truffle automatise la compilation et l'empaquetage du code Solidity en bytecode qui est envoyé à Ethereum. Machine virtuelle (EVM). Cela simplifie également le processus d'intégration de votre dApp avec des outils de développement front-end.

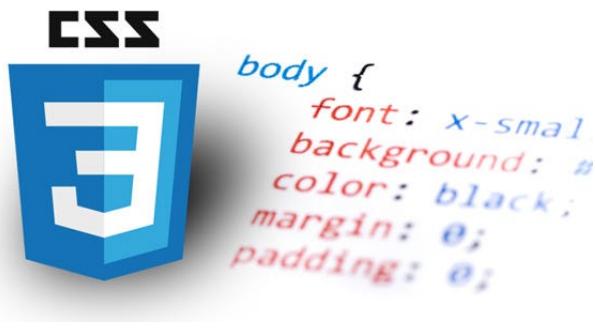


FIGURE 5.9 – CSS

5.4 Outils

5.4.1 CSS

c'est un autre langage qui vient compléter le HTML. C'est le langage qui vous permet de choisir la couleur de votre texte. Il permet de sélectionner la police utilisée sur votre site. Il permet aussi de définir la taille du texte, les bordures, le fond... Et aussi, Il permet de faire la mise en page de votre site. Vous pourrez dire : je veux que mon menu soit à gauche et occupe telle largeur, que l'en-tête de mon site soit calé en haut et qu'il soit toujours visible, etc.

5.4.2 Latex

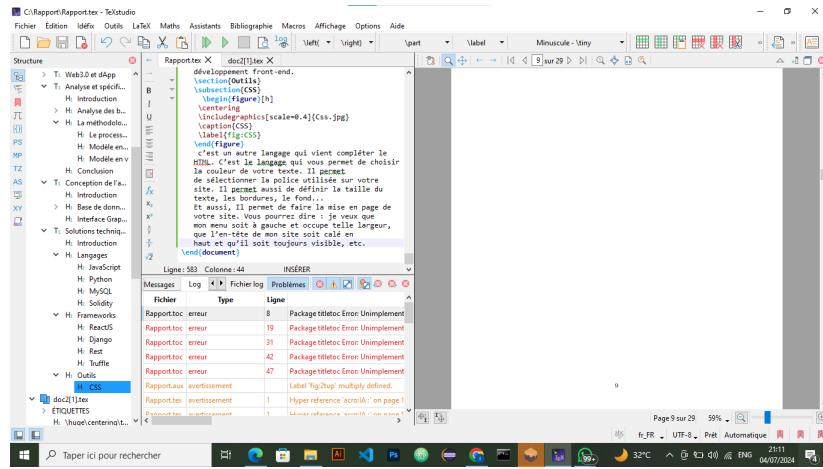


FIGURE 5.10 – Latex

est un langage de mise en page permettant de produire des documents d'une grande qualité typographique et rigoureusement homogènes dans leur présentation. Œuvre de Leslie Lamport, LaTeX constitue une surcouche du langage TeX, créé quant à lui par Donald Knuth. Ce langage est à comparer au (X)HTML, en ceci qu'il permet de baliser un texte, alors découpé proprement en sections, sous-sections, paragraphes, titres ou notes de bas de page. Le texte accompagné de ses balises est ensuite compilé, pour être mis en forme selon les spécifications de maquette choisies par l'utilisateur.

5.4.3 Visual Studio Code

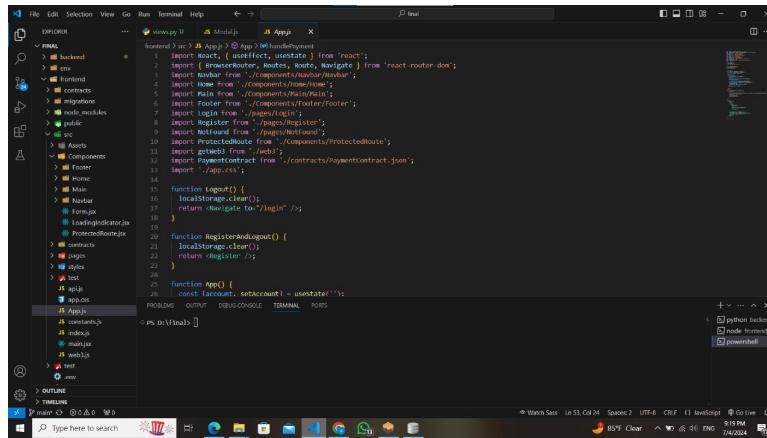


FIGURE 5.11 – Visual studio code

est un éditeur de code simplifié, qui est gratuit et développé en open source par Microsoft. Il fonctionne sous Windows, mac OS et Linux. Il fournit aux développeurs à la fois un environnement de développement intégré avec des outils permettant de faire avancer les projets techniques, de l'édition, à la construction, jusqu'à débogage.

5.4.4 MetaMask

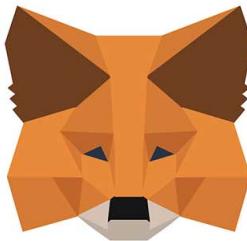


FIGURE 5.12 – MetaMask

est un wallet (portefeuille) de cryptodevises basé sur la machine virtuelle Ethereum (Ethereum Virtual Machine). Il est donc compatible avec n'importe quelle Blockchain s'appuyant sur EVM à l'instar du BAT (Basic Attention Token) et du BNB (Binance Coin).

Metamask est également connu comme étant une extension de navigateur permettant d'interagir avec des Dapp (applications décentralisées). Il est compatible avec Google Chrome, Firefox et Brave, sachant que les risques d'hameçonnage sont réduits grâce à un anti-phishing intégré.

5.4.5 Ganache



FIGURE 5.13 – Ganache

Bien que Ganache soit communément appelée Blockchain personnelle, elle devrait être considérée davantage comme un simulateur Bitcoin/Ethereum. Il s'agit d'un composant de Truffle Suite qui a été créé pour tester des applications en simulant la blockchain. Cela leur permet de s'assurer que leur code fonctionnera correctement avec de véritables transactions blockchain.

La meilleure façon de considérer Ganache est comme un grand livre bancaire (base de données) avec quelques fonctionnalités de base comme la création de comptes, la création et l'envoi de transactions, etc. C'est un excellent moyen de connaître l'efficacité de votre code et de tester vos contrats intelligents dans un environnement local avant de les déployer sur la blockchain.

5.5 Base de Données

SQLite comme base de données par défaut. Lorsque on crée un nouveau projet Django, il génère automatiquement un fichier de base de données SQLite (généralement nommé db.sqlite3) dans le répertoire du projet. SQLite est une base de données légère, sans serveur et facile à configurer, ce qui la rend idéale pour le développement et les tests.

Name	Type	Schema
Tables (13)		<pre> CREATE TABLE "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime NOT NULL, "is_superuser" bool NOT NULL, "is_staff" bool NOT NULL, "is_active" bool NOT NULL, "date_joined" datetime NOT NULL, "username" varchar(150) NOT NULL UNIQUE) CREATE TABLE "auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE) CREATE TABLE "auth_group_permissions" ("group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED) CREATE TABLE "auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "codename" varchar(100) NOT NULL, "name" varchar(50) NOT NULL) CREATE TABLE "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime NULL, "is_superuser" bool NOT NULL, "username" varchar(150) NOT NULL, "first_name" varchar(30) NOT NULL, "last_name" varchar(30) NOT NULL, "email" varchar(254) NOT NULL, "is_staff" bool NOT NULL, "is_active" bool NOT NULL, "date_joined" datetime NOT NULL) CREATE TABLE "auth_user_groups" ("user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED) CREATE TABLE "auth_user_permissions" ("user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED) CREATE TABLE "django_admin_log" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "object_id" text NOT NULL, "object_repr" varchar(200) NOT NULL, "action_flag" smallint unsigned NOT NULL, "change_message" longtext NOT NULL, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "timestamp" datetime NOT NULL) CREATE TABLE "django_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varchar(255) NOT NULL, "applied" datetime NOT NULL) CREATE TABLE "django_session" ("session_key" varchar(40) NOT NULL PRIMARY KEY, "session_data" text NOT NULL, "expire_date" datetime NOT NULL) CREATE TABLE "sqlite_sequence" ("name" seq) CREATE TABLE "trips_trip" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "destination" varchar(100) NOT NULL, "price_paid" decimal NOT NULL, "number_of_travelers" integer unsigned NOT NULL) </pre>
Indices (17)		<pre> CREATE INDEX "auth_user_author_id_b6d78e8c" ON "auth_user" ("author_id") CREATE INDEX "auth_group_group_name_a7172643" ON "auth_group" ("group_name") CREATE INDEX "auth_group_permissions_group_id_76f8c14b" ON "auth_group_permissions" ("group_id") CREATE INDEX "auth_group_permissions_group_id_92369c92" ON "auth_group_permissions" ("group_id", "permission_id") CREATE INDEX "auth_group_permissions_permission_id_84c492c9" ON "auth_group_permissions" ("permission_id") CREATE INDEX "auth_permission_content_type_id_2f47e7d0" ON "auth_permission" ("content_type_id") CREATE INDEX "auth_permission_content_type_id_c0ab375a" ON "auth_permission" ("content_type_id", "codename") CREATE INDEX "auth_user_groups_group_id_97595944" ON "auth_user_groups" ("group_id") CREATE INDEX "auth_user_groups_user_id_a1126a39" ON "auth_user_groups" ("user_id") CREATE INDEX "auth_user_permissions_permission_id_943940c8" ON "auth_user_permissions" ("permission_id") CREATE INDEX "auth_user_permissions_permission_id_f11fb52d" ON "auth_user_permissions" ("permission_id") CREATE INDEX "auth_user_permissions_user_id_a20ead1b" ON "auth_user_permissions" ("user_id") CREATE INDEX "auth_user_permissions_user_id_d14a60632" ON "auth_user_permissions" ("user_id", "permission_id") CREATE INDEX "django_admin_log_content_type_id_a1c0b8e0" ON "django_admin_log" ("content_type_id") CREATE INDEX "django_admin_log_user_id_c564fbaf" ON "django_admin_log" ("user_id") CREATE INDEX "django_content_type_app_label_76bd3d3b" ON "django_content_type" ("app_label", "model") CREATE INDEX "django_session_expire_date_a5c62663" ON "django_session" ("expire_date") </pre>

FIGURE 5.14 – Base de données

Un système d'authentification dans Django REST Framework (DRF) permet de sécuriser les API en vérifiant l'identité des utilisateurs qui tentent d'accéder aux ressources.

	Id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	pbkdf2_sha256\$720000\$1qPwFxr1ut...	2024-07-04 18:12:51.456942	0	len		abdossattar.kardii@gmail.com	1	1	2024-05-14 17:03:40.783222	
2	10	pbkdf2_sha256\$720000\$UazE8MyBS...	2024-07-03 10:10:30.4569	1	assini	issam	issamAssini@gmail.com	1	1	2024-07-04 18:10:07.651481	
3	12	pbkdf2_sha256\$720000\$UazE8MyBS...	2024-07-03 10:10:30.4569	0				1	1	2024-07-03 08:08:20.4869	Assini

FIGURE 5.15 – Liste des utilisateurs

5.6 Interfaces Graphiques de l'application

5.6.1 Page d'accueil

Afin de répondre aux besoins et attentes de nos clients, nous avons conçu une interface simple à utiliser et accessible à tous les utilisateurs. Celle-ci offre une visibilité claire sur la page d'accueil et les différentes sections du site.

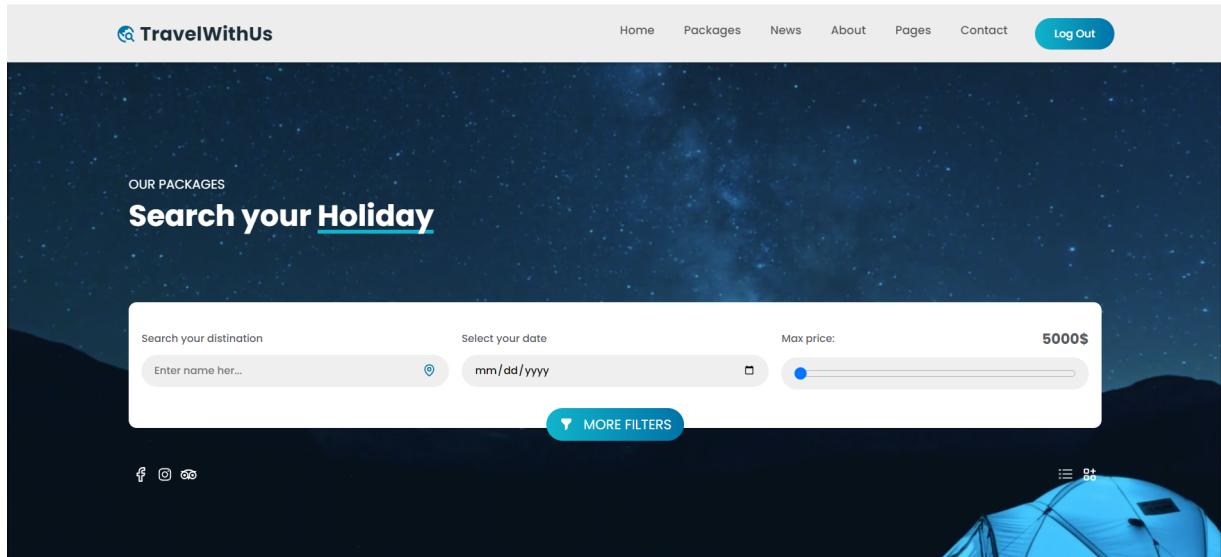


FIGURE 5.16 – La page d'accueil

5.6.2 Page d'inscription

Depuis la page d'accueil, l'utilisateur a la possibilité de s'inscrire sur notre site pour accéder à toutes les fonctionnalités de l'application en tant que client. La figure ci-dessous illustre l'interface d'inscription de l'utilisateur.

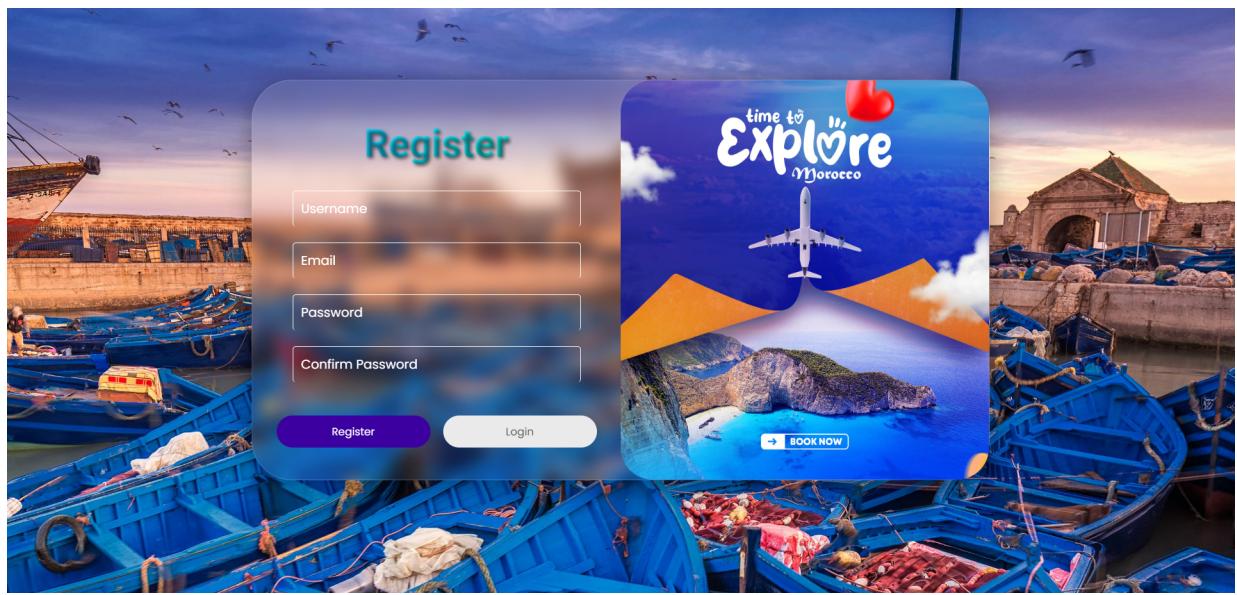


FIGURE 5.17 – Page d'inscription

5.6.3 Page de connexion

Une fois inscrit, l'utilisateur peut se connecter à son compte en entrant son nom d'utilisateur et son mot de passe, comme illustré dans la figure suivante :

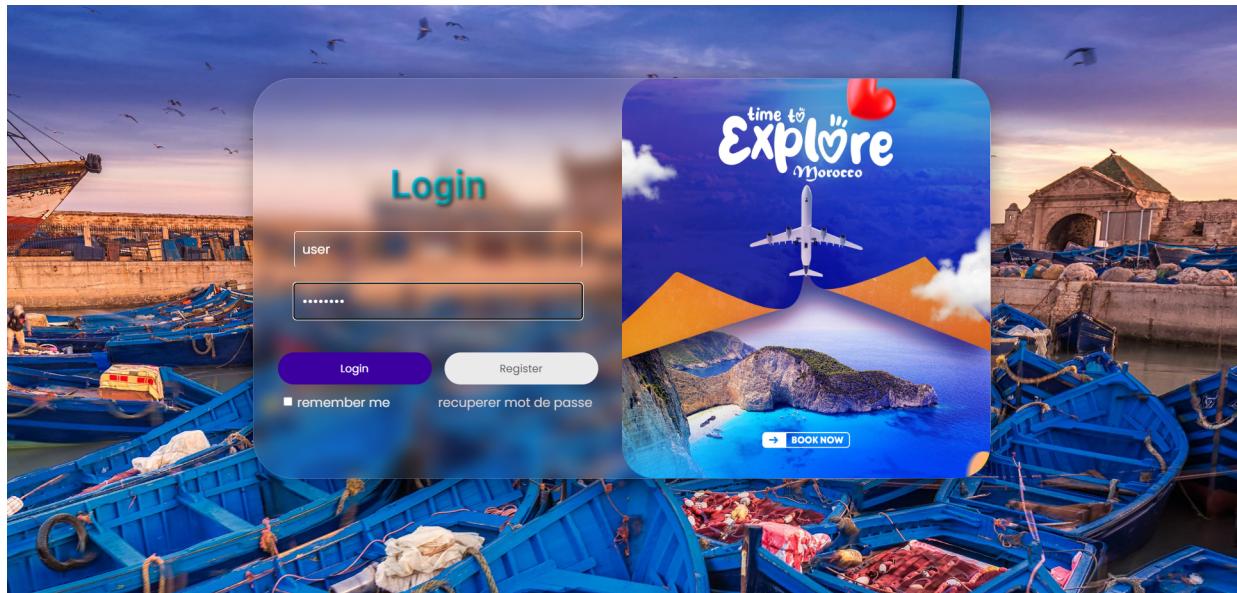


FIGURE 5.18 – Page de connexion

5.6.4 Planification de voyage

La page d'accueil vous offre la possibilité de rechercher les meilleures conditions pour votre voyage en fonction de la destination, de la durée, de la date et du prix.

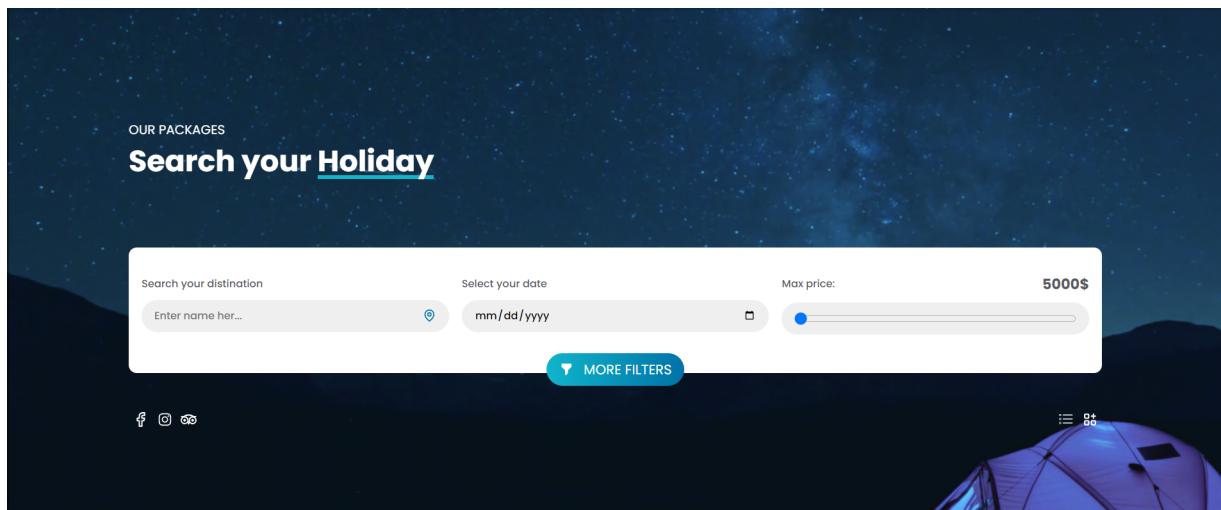


FIGURE 5.19 – Planification de voyage

5.6.5 Exploration des meilleures destinations

Explorez les meilleures destinations sur notre site, avec une opportunité unique d'explorer différentes villes du Maroc à travers leurs descriptions historiques détaillées et de découvrir les prix moyens des voyages organisés vers ces destinations.

The screenshot displays a section titled 'Most visited destinations' on a travel website. It features three cards, each representing a destination:

- Marrakech**: Located in Morocco. Type: CULTURAL RELAX. Price: \$545. Description: Welcome to Marrakesh! This former imperial city is a hub of history and culture. Explore the walled medina's alleys, filled with bustling souks selling traditional treasures. Don't miss the iconic Koutoubia Mosque's Moorish minaret, a symbol of the city. Button: DETAILS
- Chefchaouen**: Located in Morocco. Type: CULTURAL RELAX. Price: \$432. Description: Discover Chefchaouen, nestled in the Rif Mountains. Known for its blue-washed buildings, the city offers a serene atmosphere. Explore its cobbled lanes, lined with workshops and historic sites. Admire the octagonal minaret of the Great Mosque. Button: DETAILS
- Essaouira**: Located in Morocco. Type: CULTURAL RELAX. Price: \$390. Description: Experience Essaouira, a charming port city on the Atlantic coast. Wander the medina's seafront ramparts and admire the ocean views. The city's crescent beach is a paradise for water sports enthusiasts. Feel the Alizée winds as you explore. Button: DETAILS

FIGURE 5.20 – Most visited destinations 1

voilà la suite des destinations :

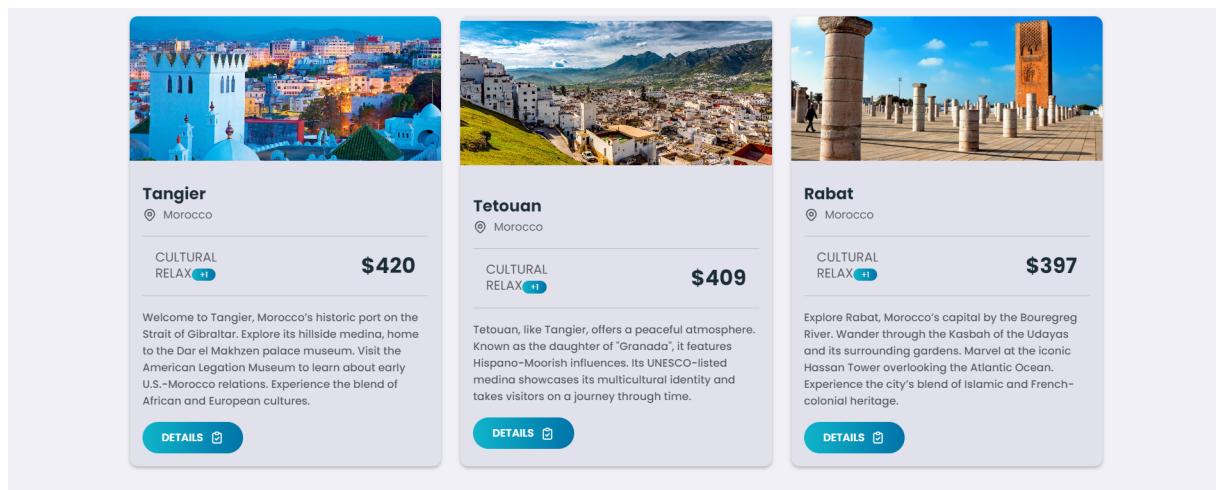


FIGURE 5.21 – Most visited destinations 2

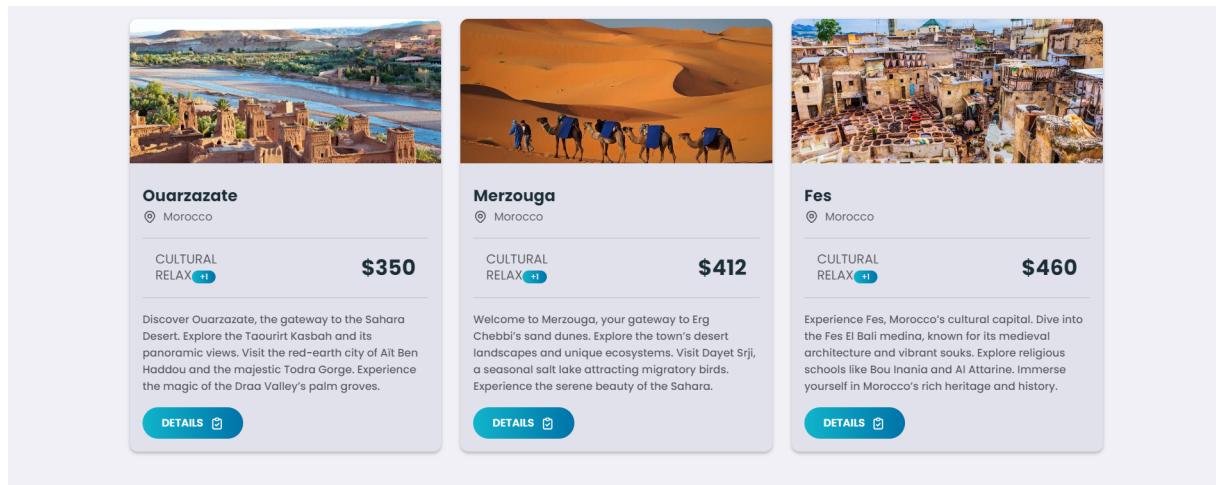


FIGURE 5.22 – Most visited destinations 3

5.6.6 Réservation

Lorsque vous cliquez sur "Détails" lors de la réservation, une fenêtre s'ouvre avec un diaporama d'images des destinations et des activités disponibles dans cette ville. Vous trouverez également un paragraphe détaillant les informations sur le voyage organisé, incluant les dates, les activités prévues et d'autres détails importants. En dessous, vous pouvez sélectionner le nombre de voyageurs et consulter le prix total avant de procéder au paiement en cliquant sur le bouton dédié.

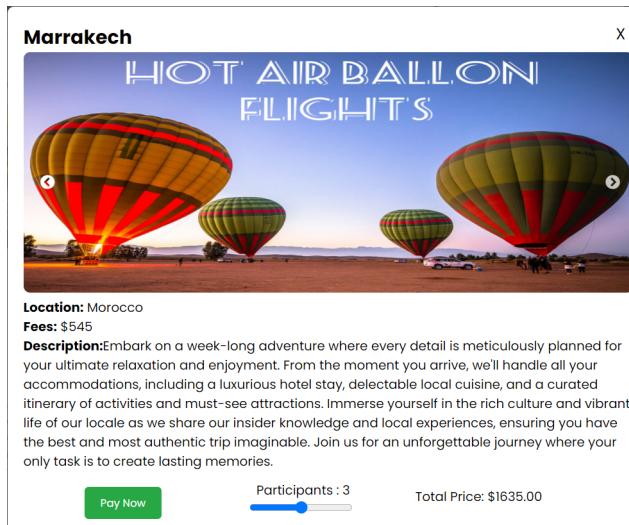


FIGURE 5.23 – Reservation

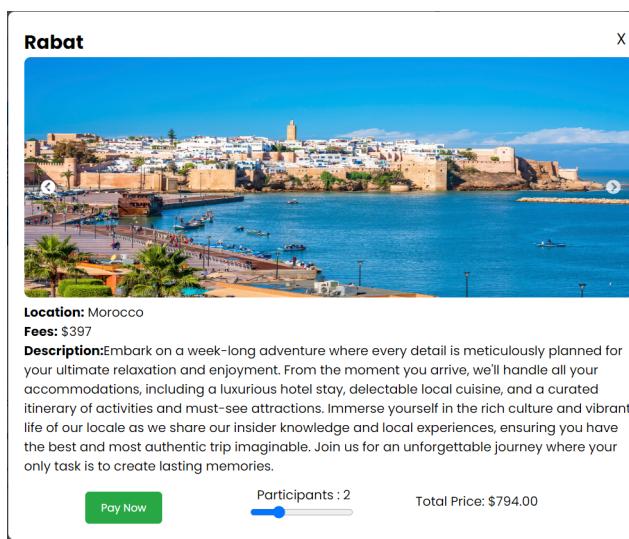


FIGURE 5.24 – Reservation

5.6.7 Le paiement

Le processus de paiement vous dirige vers votre portefeuille MetaMask, où vous pouvez vérifier les frais de transaction et de paiement. Après confirmation, votre voyage est enregistré avec tous ses détails, et un e-mail contenant toutes les informations nécessaires est envoyé à votre boîte de réception.

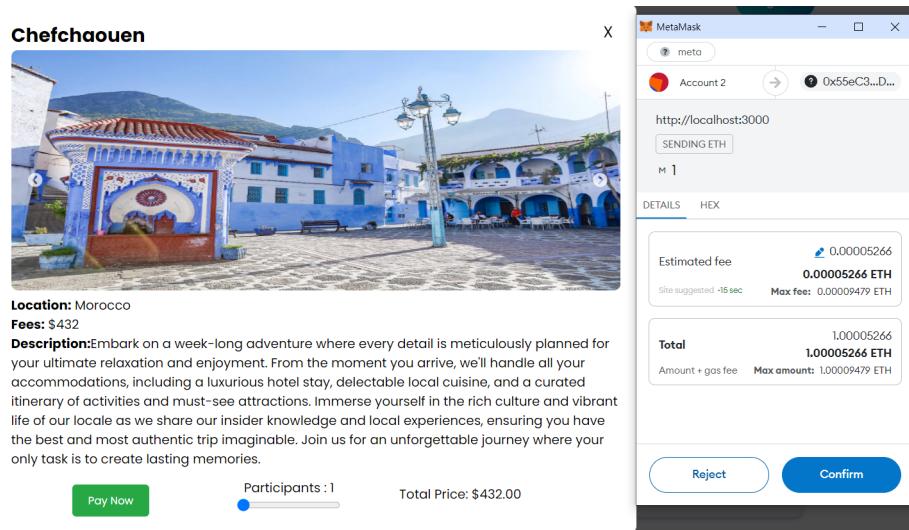


FIGURE 5.25 – Paiement

5.6.8 Footer

Le footer comprend un sommaire de notre site, incluant les pages de contact, nos partenaires, ainsi que des suggestions pour d'autres agences de tourisme étrangères.

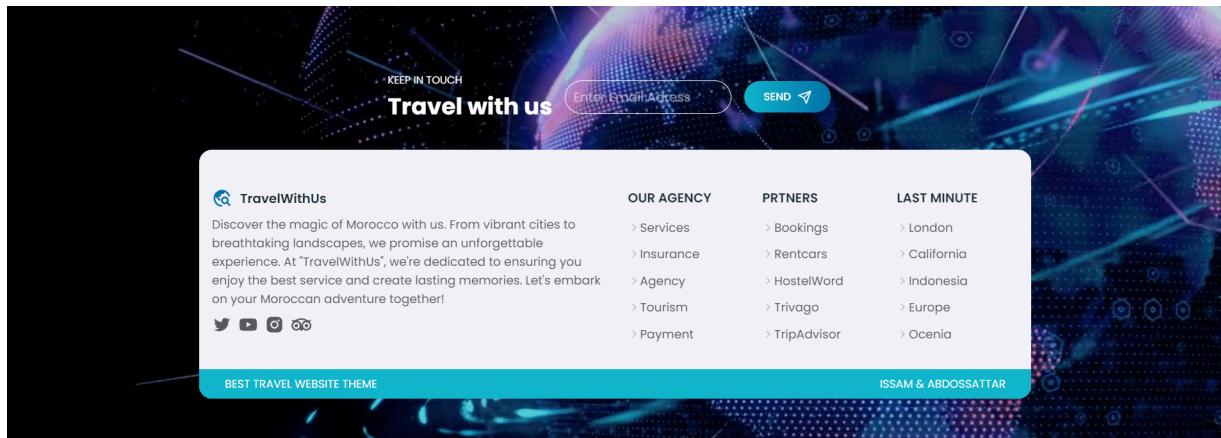


FIGURE 5.26 – Footer

5.7 Code

Dans cette partie, nous allons présenter les codes qui permettent le fonctionnement de l'application.

5.7.1 Authentification

Quand le client saisie son email et mot de passe, le système vérifie son existence dans la base de données. En cas de présence, une session est ouverte et le client est redirigé vers son profil.

```
1 function Form({ route, method }) {
2   const [username, setUsername] = useState("");
3   const [email, setEmail] = useState(""); // New state for email
4   const [password, setPassword] = useState("");
5   const [confirmPassword, setConfirmPassword] = useState(""); // New state for retype password
6   const [loading, setLoading] = useState(false);
7   const navigate = useNavigate();
8
9   const name = method === "login" ? "Login" : "Register";
10
11   const handleSubmit = async (e) => {
12     setLoading(true);
13     e.preventDefault();
14
15     try {
16       if (method === "register" && password !== confirmPassword) {
17         throw new Error("Passwords do not match");
18       }
19
20       const requestData = method === "register"
21         ? { username, email, password } // If registering, include email
22         : { username, password }; // Otherwise, only username and password
23
24       const res = await api.post(route, requestData);
25
26       if (method === "login") {
27         localStorage.setItem(ACCESS_TOKEN, res.data.access);
28         localStorage.setItem(REFRESH_TOKEN, res.data.refresh);
29         navigate("/");
30       } else {
31         navigate("/login");
32       }
33     } catch (error) {
34       alert(error.message);
35     } finally {
36       setLoading(false);
37     }
38   };
39
40   const handleOtherButtonClick = () => {
41     if (method === "login") {
42       navigate("register");
43     } else {
44       navigate("/login");
45     }
46   };
47
48 }
49
50
51 const handleOtherButtonClick = () => {
52   if (method === "login") {
53     navigate("register");
54   } else {
55     navigate("/login");
56   }
57 }
```

FIGURE 5.27 – Authentification

5.7.2 Le paiement

Pour mettre en place un processus de paiement utilisant la bibliothèque Web3 et Smart Contracts.

FIGURE 5.28 – PaymentContract

```
50 | const handlePayment = async (amount) => {
51 |   if (contract) {
52 |     try {
53 |       await contract.methods.pay().send({ from: account, value: web3.utils.toWei(amount.toString(), 'ether') });
54 |       alert('Payment Successful!');
55 |     } catch (error) {
56 |       console.error(error);
57 |       alert('Payment Failed!');
58 |     }
59 |   }
60 | };


```

FIGURE 5.29 – HandlePay

5.7.3 Smart Contract

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract PaymentContract {
5   address public owner;
6
7   constructor() {
8     owner = msg.sender;
9   }
10
11  function pay(address payable recipient) public payable {
12    require(msg.value > 0, "Send some Ether");
13    recipient.transfer(msg.value);
14  }
15 }
```

FIGURE 5.30 – Smart Contract

```

52  | const handlePayment = async (amount, recipientAddress) => {
53  |   if (contract) {
54  |     try {
55  |       await contract.methods.pay(recipientAddress).send({
56  |         from: account,
57  |         value: web3.utils.toWei(amount.toString(), 'ether')
58  |       });
59  |       alert('Payment Successful!');
60  |     } catch (error) {
61  |       console.error(error);
62  |       alert('Payment Failed!');
63  |     }
64  |   }
65  | };

```

FIGURE 5.31 – Gestion de paiement

5.8 Obstacles et solutions

5.8.1 Obstacles techniques

❖ Enregistrement des Données sur Web3

► Obstacle

Utilisation de technologies de blockchain pour le stockage des données, nécessitant une approche différente de la conception et de l'implémentation par rapport aux bases de données traditionnelles.

► Solution

Utiliser des solutions adaptées au stockage décentralisé, comme IPFS (InterPlanetary File System) pour le stockage de fichiers et des bases de données décentralisées pour des données structurées. Adopter des modèles de conception adaptés à la nature immuable et distribuée de la blockchain.

❖ Coût et Complexité des Frameworks Web3

► Obstacle

Difficultés et coûts associés à l'utilisation de frameworks spécifiques pour le développement, le test et le déploiement des DApps.

► Solution

Explorer des alternatives open-source ou des frameworks moins complexes selon les besoins spécifiques du projet. Utiliser des environnements de développement décentralisés (DIDE) qui simplifient le processus de développement et de déploiement. Intégrer des outils de gestion de versions adaptés aux contrats intelligents et à la blockchain.

❖ Intégration Complète et Transparence

► Obstacle

Défis pour atteindre une transparence totale en exploitant toutes les fonctionnalités de la blockchain tout en maintenant une interface utilisateur conviviale.

► Solution

Concevoir une architecture flexible qui combine la transparence offerte par la blockchain avec une expérience utilisateur intuitive. Utiliser des bibliothèques JavaScript et des API pour faciliter l'interaction avec la blockchain tout en cachant la complexité technique. Mettre en place des mécanismes robustes de vérification et d'authentification des transactions pour garantir la sécurité et la confiance des utilisateurs.

❖ Évolutivité et Performance

► Obstacle

Contraintes liées à l'évolutivité des DApps sur la blockchain, notamment en termes de vitesse et de capacité à gérer un grand nombre de transactions.

► Solution

Explorer des blockchains qui offrent des solutions de mise à l'échelle comme les sidechains, les sharding ou le proof-of-stake pour améliorer la performance. Optimiser les contrats intelligents et les interactions pour réduire les frais de transaction et améliorer la vitesse de traitement.

❖ Sécurité et Auditabilité

► Obstacle

Risques de sécurité spécifiques aux contrats intelligents et à l'environnement décentralisé, nécessitant une attention particulière aux failles de sécurité et à l'auditabilité des transactions.

► Solution

Effectuer des audits de sécurité réguliers, utiliser des modèles de développement sécurisés et des frameworks qui facilitent l'écriture de contrats intelligents sûrs. Mettre en œuvre des mécanismes de gouvernance et de consensus robustes pour minimiser les risques d'exploitation.

5.8.2 Obstacles fonctionnels

❖ Ergonomie et UX (Expérience Utilisateur)

► Obstacle

Concevoir une interface utilisateur intuitive et agréable peut être complexe, surtout dans le contexte de sites Web axés sur le tourisme, qui doivent offrir une navigation fluide et conviviale.

► Solution

- **Recherche Utilisateur** Effectuer des études approfondies sur les utilisateurs cibles (touristes) pour comprendre leurs besoins, leurs préférences et leurs comportements en ligne. Cela permet de concevoir une interface qui répond spécifiquement à leurs attentes.
- **Prototypage et Tests** Utiliser des outils de prototypage pour créer des maquettes interactives du site et les tester auprès d'un groupe d'utilisateurs représentatif. Les retours des utilisateurs peuvent fournir des insights précieux pour affiner et améliorer l'interface.

-
- **Conception Adaptative** Adopter une approche de conception responsive, assurant que le site s'adapte de manière optimale à divers appareils (ordinateurs de bureau, tablettes, smartphones) et résolutions d'écran. Cela garantit une expérience utilisateur cohérente et agréable sur tous les dispositifs.

❖ **Fonctionnalités Spécifiques aux Besoins des Utilisateurs Touristiques**

► **Obstacle**

Développer et intégrer des fonctionnalités qui répondent aux besoins uniques des utilisateurs touristiques peut nécessiter une compréhension approfondie des attentes et des comportements spécifiques de cette audience.

► **Solution**

- **Analyse des Besoins** Identifier les fonctionnalités essentielles pour les touristes tels que la recherche d'hébergements, la planification d'itinéraires, la réservation d'activités, les guides touristiques, etc.
- **Partenariats et Intégrations** Collaborer avec des fournisseurs de services locaux (hôtels, restaurants, agences de voyage) pour intégrer leurs services directement sur le site. Cela peut améliorer l'expérience utilisateur en offrant des options de réservation directe et en temps réel.
- **Personnalisation** Mettre en œuvre des fonctionnalités de personnalisation basées sur les préférences et le comportement des utilisateurs, comme la recommandation d'activités basée sur les intérêts précédemment exprimés ou les recherches effectuées.

❖ **Performances : Optimisation des Temps de Chargement et de la Réactivité du Site**

► **Obstacle**

Les sites Web de tourisme doivent être rapides et réactifs pour offrir une expérience utilisateur fluide, même dans des conditions de réseau variables.

► **Solution**

- **Optimisation des Images et des Médias** Réduire la taille des images et des vidéos sans compromettre leur qualité pour minimiser les temps de chargement.
- **Mise en Cache** Utiliser des techniques de mise en cache pour stocker temporairement les données fréquemment demandées côté client, réduisant ainsi les requêtes au serveur et accélérant la réponse du site.
- **CDN (Content Delivery Network)** Utiliser un CDN pour distribuer le contenu du site à partir de serveurs géographiquement dispersés, réduisant ainsi la latence et améliorant la vitesse de chargement pour les utilisateurs dans différentes régions.

5.8.3 Obstacles juridiques

❖ **Conformité RGPD (Règlement Général sur la Protection des Données)**

► **Obstacle**

Le RGPD impose des obligations strictes sur la collecte, le traitement et la gestion des données personnelles des utilisateurs résidant dans l'Union Européenne (UE). Cela inclut la nécessité d'obtenir un consentement explicite pour la collecte de données, le respect des droits des utilisateurs (comme le droit à l'effacement et le droit à la portabilité des données), et l'obligation de sécuriser les données personnelles.

► Solution

- **Audit de Conformité** Effectuer un audit pour évaluer la conformité actuelle du site par rapport aux exigences du RGPD. Cela inclut l'examen des politiques de confidentialité, des mécanismes de consentement et des mesures de sécurité des données.
- **Mise en Place de Mesures** Mettre en œuvre des mesures techniques et organisationnelles appropriées pour protéger les données personnelles. Cela peut inclure le cryptage des données, la pseudonymisation, et la limitation de l'accès aux données sensibles.
- **Consentement Clair** Assurer que les utilisateurs donnent leur consentement explicite et libre pour la collecte et le traitement de leurs données personnelles, en fournissant des options claires et faciles pour le retrait du consentement.

❖ Propriété Intellectuelle

► Obstacle

Gérer les droits d'auteur et les licences pour les contenus utilisés sur le site, y compris les images, les vidéos, les textes et autres médias, afin de respecter les droits des créateurs et des détenteurs de droits.

► Solution

- **Obtention de Licences** S'assurer d'obtenir les licences appropriées pour utiliser les contenus protégés par des droits d'auteur. Cela peut impliquer l'achat de licences auprès de fournisseurs de médias ou la négociation directe avec les détenteurs de droits.
- **Utilisation de Contenus Libres de Droits** Privilégier l'utilisation de contenus sous licence libre, tels que ceux distribués sous des licences Creative Commons, qui permettent une utilisation spécifiée sans nécessiter de permission directe.
- **Mention des Crédits** Inclure des crédits appropriés et des attributions pour les contenus tiers utilisés sur le site, conformément aux termes des licences et aux exigences des détenteurs de droits.

❖ Réglementations Locales

► Obstacle

Respecter les lois et régulations spécifiques aux différentes régions touristiques couvertes par le site, ce qui peut inclure des exigences variées en matière de protection des consommateurs, de taxes locales, de normes de sécurité, etc.

► Solution

- **Recherche Juridique** Effectuer une recherche approfondie sur les réglementations locales dans chaque région touristique couverte par le site. Cela peut nécessiter des consultations avec des experts juridiques locaux pour s'assurer de la conformité aux exigences spécifiques.
- **Adaptation des Politiques** Adapter les politiques du site, telles que les conditions d'utilisation et les politiques de confidentialité, pour refléter les exigences légales spécifiques de chaque région.

❖ Problème de Scraping des Données

► Obstacle

Le scraping des données peut poser des défis en termes de respect des droits de propriété intellectuelle, de violation des conditions d'utilisation des sites sources, et de conformité avec les lois sur la protection des données lorsque des informations personnelles sont collectées.

► Solution

- **Conditions d'Utilisation Claires** Élaborer des conditions d'utilisation claires et restrictives interdisant le scraping des données à partir du site. Inclure des mesures techniques pour détecter et bloquer les activités de scraping automatisé.
- **Permission Préalable** Obtenir la permission préalable des propriétaires des sites sources avant de collecter et d'utiliser des données à partir de leurs pages web, en respectant leurs conditions d'utilisation et leurs politiques de confidentialité.
- **Respect des Lois sur la Protection des Données** Assurer que toutes les données personnelles collectées à travers le scraping sont traitées conformément aux réglementations sur la protection des données en vigueur, en obtenant le consentement approprié lorsque nécessaire.

5.8.4 Obstacles organisationnels

❖ Gestion de Projet

► Obstacle

Assurer une coordination efficace des tâches et le respect des délais dans le cadre du développement du site web.

► Solution

- **Planification Structurée** Élaborer un plan de projet détaillé incluant les étapes clés, les livrables attendus, les responsabilités des membres de l'équipe, et les échéances spécifiques.
- **Utilisation d'Outils de Gestion de Projet** Adopter des outils collaboratifs tels que Trello, Asana ou Jira pour attribuer des tâches, suivre les progrès, et gérer les dépendances entre les différentes phases du projet.
- **Réunions Régulières** Organiser des réunions périodiques pour faire le point sur l'avancement du projet, résoudre les problèmes rencontrés et ajuster les plans si nécessaire.
- **Gestion Agile** Appliquer des méthodologies agiles comme Scrum ou Kanban pour favoriser une flexibilité et une réactivité accrues aux changements tout au long du cycle de développement.

❖ Communication

► Obstacle

Assurer une communication claire et efficace entre les membres de l'équipe de développement, ainsi qu'avec les parties prenantes externes.

► Solution

-
- **Canal de Communication Dédié** Utiliser des plateformes de communication comme Slack, Microsoft Teams, ou Discord pour faciliter les échanges rapides et la collaboration entre les membres de l'équipe.
 - **Documentation** Maintenir une documentation claire et accessible sur les objectifs du projet, les décisions prises, et les défis rencontrés, afin que toutes les parties prenantes soient informées et alignées.
 - **Transparence** Favoriser une culture de transparence en partageant régulièrement les progrès du projet, les obstacles rencontrés, et les plans d'action pour les surmonter.
 - **Réunions Structurées** Organiser des réunions régulières avec les parties prenantes pour obtenir des retours d'information, clarifier les exigences et assurer une compréhension mutuelle des objectifs du projet.

❖ **Formation**

► **Obstacle**

Besoin de former l'équipe aux nouvelles technologies utilisées dans le cadre du développement du site web touristique.

► **Solution**

- **Sessions de Formation** Planifier des sessions de formation initiales pour familiariser l'équipe avec les technologies spécifiques utilisées, telles que les frameworks de développement web, les bibliothèques de blockchain, ou les outils de gestion de contenu.
- **Formation Continue** Encourager la formation continue en soutenant la participation à des cours en ligne, des webinaires, des conférences, ou des ateliers pertinents pour développer les compétences nécessaires.
- **Mentorat Interne** Faciliter le mentorat entre les membres de l'équipe expérimentés et les nouveaux arrivants pour partager les connaissances et les bonnes pratiques.
- **Documentation et Ressources** Fournir un accès facile à une documentation détaillée, des tutoriels et des ressources en ligne pour permettre à l'équipe d'approfondir ses connaissances de manière autonome.

5.8.5 Obstacles de Ressources

❖ **Ressources Humaines**

► **Obstacle**

Besoin de développeurs spécialisés en technologies Web3 et autres compétences spécifiques.

► **Solution**

- **Recrutement ciblé** Identifier et recruter des développeurs expérimentés dans les technologies Web3 telles que Solidity pour les contrats intelligents, ainsi que des compétences complémentaires comme la blockchain et les applications décentralisées (DApps).

-
- **Formation interne** Investir dans la formation interne pour développer les compétences nécessaires au sein de l'équipe existante. Cela peut inclure des cours en ligne, des workshops, et des projets pilotes pour acquérir de l'expérience pratique.
 - **Partenariats stratégiques** Établir des partenariats avec des experts externes ou des consultants spécialisés en technologie blockchain pour apporter un soutien supplémentaire et des conseils stratégiques.

❖ **Ressources Financières**

► **Obstacle**

Budget nécessaire pour le développement et le maintien du site web touristique

► **Solution**

- **Élaboration d'un budget détaillé** Estimer les coûts initiaux et récurrents pour le développement du site, y compris les frais de développement logiciel, les licences, l'hébergement, la maintenance et la sécurité.
- **Recherche de financements** Explorer différentes options de financement telles que les investisseurs, les subventions gouvernementales, les concours de startup, ou le crowdfunding, en fonction des besoins spécifiques du projet et de sa portée.
- **Optimisation des coûts** Évaluer les possibilités d'optimisation des coûts en utilisant des solutions open-source, en choisissant des services d'hébergement économiques, et en planifiant soigneusement l'infrastructure technologique pour éviter les dépenses excessives.

❖ **Ressources Matérielles**

► **Obstacle**

Infrastructure nécessaire pour héberger et faire fonctionner le site de manière optimale.

► **Solution**

- **Choix d'une infrastructure scalable** Sélectionner une plateforme d'hébergement cloud comme AWS (Amazon Web Services), Google Cloud Platform, ou Azure, qui offre des solutions évolutives adaptées aux besoins de trafic variables d'un site touristique.
- **Optimisation des performances** Configurer l'infrastructure pour maximiser la disponibilité et la vitesse du site, en utilisant des services de mise en cache, de répartition de charge (load balancing), et en optimisant les bases de données pour des temps de réponse rapides.
- **Sécurité et sauvegardes** Mettre en place des mesures de sécurité robustes pour protéger les données des utilisateurs et assurer des sauvegardes régulières des données critiques afin de minimiser les risques de pertes.

5.9 Conclusion

Nous avons soigneusement sélectionné nos outils techniques pour offrir la meilleure expérience à un large éventail de visiteurs. L'utilisation de React a permis de créer un site compatible avec toutes les résolutions, que ce soit sur mobile ou sur PC. En intégrant Django, nous avons simplifié la connexion à la base de données, facilitant ainsi la gestion

des données essentielles du site. De plus, l'incorporation des fonctionnalités de Web3 a assuré une transparence totale des transactions, renforçant la confiance des utilisateurs dans nos services. Cette combinaison stratégique d'outils nous a permis non seulement de répondre aux besoins techniques complexes mais aussi de garantir une expérience utilisateur fluide et sécurisée.

Références

- [1]Rapport de la mission d'information commune sur la blockchain (chaîne de blocs) et ses usages :un enjeu de souveraineté
- [2].<https://www.advency.fr/expertises/creation-de-projet-numerique/creation-de-dapp-web3>(Dernière Consultation 13/07/2024)
- [3] Rapport de la mission d'information commune sur la blockchain (chaîne de blocs) et ses usages,un enjeu de souveraineté 2018
- [4].MDP .Web 3.0 and Decentralized Applications
- [5].<https://www.comparateurbanque.com/guides/metamask-cest-quoi/> (Dernière Consultation 13/07/2024)
- [6].Cours GL Prof Brahim Ouchao
- [7].Cours GL Prof Brahim Ouchao

Documentation

<https://legacy.reactjs.org/docs/getting-started.htm> (Dernière Consultation 13/07/2024)

<https://docs.djangoproject.com/en/5.0/> (Dernière Consultation 13/07/2024)

<https://forum.freecodecamp.org/t/documentation-of-html-and-css/551354> (Dernière Consultation 13/07/2024)

<https://developer.mozilla.org/en-US/docs/Web/JavaScript> (Dernière Consultation 13/07/2024)

<https://www.sqlite.org/docs.html> (Dernière Consultation 13/07/2024)