DocumentDeSuiviDeProjectFinal : Gestion d'Hôtel (Hotel Management System)

Informations Générales

Nom du Projet	Gestion d'Hôtel (hotelmanagementsystem)
Chef de Projet	Kanmegne Tabouguie Andre Yvan
Équipe	Kanmegne Tabouguie et Marc
GitHub Repo	Lien vers le repo

Objectif et Contexte

But	Développer une application de gestion d'hôtel permettant aux administrateurs de gérer les utilisateurs et les chambres, et aux clients de consulter les disponibilités des chambres.
Motivation	Valider les compétences acquises en développement, infrastructure et réseaux au cours de cette année.
Contexte	Ce projet s'inscrit dans le cadre de notre formation en développement et infrastructure, visant à mettre en pratique les compétences théoriques et pratiques acquises.

Périmètre et Critères de Succès

Dans le Périmètre	Critères de Succès
- Développement d'une API pour la gestion des utilisateurs et des chambres.	- Fonctionnalité complète de l'API (CRUD pour utilisateurs et chambres).
- Développement d'une interface utilisateur pour les administrateurs et les clients.	- Interface utilisateur réactive et accessible.
- Mise en place de l'infrastructure nécessaire (serveurs web, base de données, gestion	- Déploiement réussi de l'application sur un serveur de production.

d'accès).	
- Mise en place de la sécurité (authentification, pare-feu).	- Respect des délais.

Hors du Périmètre

- Fonctionnalités avancées non essentielles à la gestion de base des utilisateurs et des chambres (intégration de systèmes externes, gestion financière, fonctionnalités multilingues et multi-devises).
- Intégration avec des systèmes externes (par exemple, systèmes de paiement, CRM).

Planification

Durée : 6 jours (2 mai au 7 mai)

Activités :

1. **Définir les rôles et responsabilités des membres de l'équipe.**
2. **Établir les objectifs du projet.**
3. **Élaborer un plan de projet détaillé.**
4. **Sélectionner les technologies à utiliser.**

MVP (Minimum Viable Product)

Fonctionnalités Implémentées :

- **User Authentication** : Inscription, connexion et déconnexion des utilisateurs.
- **Authorization** : Protection des routes basées sur les rôles (admin ou guest).
- **Room Management** : Création, mise à jour, suppression et consultation des chambres par les administrateurs.
- **User Management** : Création, mise à jour, suppression et consultation des utilisateurs par les administrateurs.
- **Token Invalidation** : Invalidation des tokens à la déconnexion.
- **Caching** : Mise en cache des données des chambres et des utilisateurs.
- **chatbot to answer user infos**

Livrables

- UML détaillant la base de données.
- Documentation technique complète de l'architecture du projet, du code, et des configurations.
- Documentation détaillant l’objectif du projet.
- README détaillant le lancement de l'application.
- URL d'un dépôt GitHub.

Phase 1: Planification

Activités

1. **Définir les rôles et responsabilités des membres de l'équipe.**
    - **Membre 1(ANdre) (Développeur Backend & Frontend, Documentation et infrastructure)** :
        - Responsable du développement des API backend.
        - Responsable du développement des UX/UI frontend.
        - Conception BDD.
        - Documentation.
    - **Membre 2 (Infrastructure & Documentation)** :
        - Configuration des serveurs et installation des serveurs.
        - Mise en place d’un pare-feu.
        - Gestion des accès.
        - Documentation.
2. **Établir les objectifs du projet.**
    - Développer une application de gestion d'hôtel avec les fonctionnalités suivantes :
        - **User Authentication** : Inscription, connexion et déconnexion des utilisateurs.

- **Authorization** : Protection des routes en fonction des rôles des utilisateurs (admin ou guest).
- **Room Management** : Gestion des chambres par les administrateurs (création, mise à jour, suppression, consultation).
- **User Management** : Gestion des utilisateurs par les administrateurs (création, mise à jour, suppression, consultation).
- **Token Invalidation** : Invalidation des tokens lors de la déconnexion.
- **Caching** : Mise en cache des données des chambres et des utilisateurs.
- **chatbot to answer user infos**

3. **Élaborer un plan de projet détaillé.**

- **Semaine 1 : Planification (3 jours) (2 au 4 mai)**
  - Définir les rôles et responsabilités.
  - Établir les objectifs du projet.
  - Sélectionner les technologies à utiliser.(nodejs with express,html,css,boostrap et javascript,Mysql)
- **Semaine 1 : Conception (3 jours) (5 au 7 mai)**
  - Conception de l'architecture de l'application(mvc architecture monolythic design)
  - Création des diagrammes UML pour la base de données.
  - Rédaction de la documentation technique initiale.
- **Semaine 2-3 : Développement Backend (10 jours) (8 au 17 mai)**
  - Développement des API CRUD pour la gestion des utilisateurs et des chambres.
  - Implémentation de l'authentification des utilisateurs.
  - Tests integration des API.
- **Semaine 4 : Développement Frontend (20-26 mai)**
  - Développement de l'interface utilisateur réactive et accessible.

    - Intégration du frontend avec les API backend.
    - Tests d'interface utilisateur.
- **Semaine 5 : Infrastructure & Réseaux (27 mai - 4juin)**
    - Configuration du frontend servi par Apache
    - Configuration du backend servi par Nginx
    - Configuration du serveur de base de données MySQL
    - Gestion des applications Node.js avec PM2
    - Configuration SSL
    - Règles de pare-feu
    - Gestion des accès
- **Semaine 12 : Tests & Validation(5juin-9juin)**
    - Tests d'intégration.
    - Validation de l'infrastructure.
- **Semaine 13 : Documentation & Livraison(10-13 juin)**
    - Finalisation de la documentation technique et du README.
    - Dépôt du projet sur GitHub.