**Projet : Système de Gestion d'Hôpital**  
Ce projet est une application web , permettant de gérer efficacement les opérations d'un hôpital. Le système est divisé en trois panneaux principaux : **Admin**, **Patient**, et **Médecin**, chacun ayant des fonctionnalités spécifiques.

**Cahier des charges : Système de Gestion d'Hôpital**

**1. Contexte et objectifs**

Le système de gestion d'hôpital vise à automatiser et centraliser les opérations administratives et médicales. Il permettra de gérer les patients, les rendez-vous, les médecins, les traitements, et les facturations dans un hôpital.  
**Objectifs principaux :**

* Faciliter la gestion des patients et des médecins.
* Optimiser le suivi des dossiers médicaux.
* Réduire les erreurs humaines dans les processus administratifs.

**2. Description générale**

Le projet sera une application web divisée en trois catégories principales d'utilisateurs :

1. **Administrateur** : Gère l’ensemble des données de l’hôpital.
2. **Médecin** : Gère les patients, approuve les rendez-vous, et fournit les prescriptions et traitements.
3. **Patient** : Prend des rendez-vous, consulte son dossier médical et effectue des paiements.

**3. Fonctionnalités**

**A. Fonctionnalités pour les patients**

* Création d'un compte personnel.
* Prise de rendez-vous avec un médecin.
* Consultation du statut des rendez-vous (en attente, approuvé, rejeté).
* Accès au dossier médical (traitements, prescriptions, factures).
* Paiement des factures et consultation des reçus.

**B. Fonctionnalités pour les médecins**

* Accès aux informations des patients et leur historique médical.
* Gestion des rendez-vous : approbation ou rejet.
* Ajout de traitements et prescriptions (nom des médicaments, dosage, etc.).
* Définition des horaires de consultation.
* Suivi des revenus personnels.

**C. Fonctionnalités pour les administrateurs**

* Gestion des utilisateurs (patients et médecins).
* Gestion des départements, traitements, et médicaments.
* Visualisation des rapports financiers (revenus par médecin, factures).
* Suivi des statistiques : nombre de patients, consultations, etc.

**4. Contraintes techniques**

* **Langages utilisés** : PHP (Backend), HTML, CSS, JavaScript (Frontend).
* **Base de données** : MySQL.
* **Compatibilité** : Navigateurs modernes (Google Chrome, Firefox, etc.).
* **Design** : Utilisation d’un framework CSS (Materialize, Bootstrap ou Tailwind).

**5. Structure des utilisateurs**

**Rôles et accès :**

1. **Patient**
   * Accès limité à son propre compte et ses données.
   * Interaction avec les fonctionnalités liées à son profil (rendez-vous, paiements).
2. **Médecin**
   * Gestion des rendez-vous des patients.
   * Ajout et mise à jour des données médicales des patients.
3. **Administrateur**
   * Accès complet à toutes les données de l’hôpital.
   * Supervision des patients, médecins et ressources de l’hôpital.

**6. Livrables**

* Application web fonctionnelle.
* Documentation technique (structure de la base de données, code source commenté).
* Rapport utilisateur (manuel pour comprendre les fonctionnalités principales).

**7. Planification**

**Phases du projet :**

1. **Analyse des besoins** : 1 semaine.
2. **Conception (maquettes, base de données)** : 1 semaine.
3. **Développement Backend** : 2 semaines.
4. **Développement Frontend** : 2 semaines.
5. **Test et validation** : 1 semaine.
6. **Livraison** : 1 jour.

**8. Évaluation des risques**

* Risques de sécurité : Implémentation de protections (hachage des mots de passe, validation des données).
* Complexité des fonctionnalités : Répartition du projet en modules indépendants.

**9. Budget et ressources**

**Ressources nécessaires :**

* Logiciels : Serveur local (XAMPP/WAMP), éditeur de code (VS Code, Sublime Text).
* Hébergement : Optionnel (pour déploiement en ligne).

**Présentation du Projet : Système de Gestion d'Hôpital**

**1. Analyse des besoins**

L’objectif principal est de centraliser et automatiser la gestion des opérations hospitalières.

* **Utilisateurs cibles** : Administrateurs, médecins, et patients.
* **Besoins identifiés** :
  + Optimisation des rendez-vous et du suivi des dossiers médicaux.
  + Amélioration de la gestion des factures et des paiements.
  + Réduction des erreurs humaines dans les processus administratifs.

**2. Problématique**

Les hôpitaux traditionnels rencontrent souvent des défis tels que :

* Gestion désorganisée des dossiers médicaux.
* Difficultés dans la coordination des rendez-vous.
* Manque d’outils centralisés pour visualiser les statistiques et les rapports financiers.

**3. Objectifs du projet**

* Automatiser les processus administratifs pour réduire les tâches répétitives.
* Permettre un accès rapide et sécurisé aux dossiers médicaux pour les médecins et les patients.
* Centraliser la gestion des données hospitalières dans une seule application web.

**4. Conception**

* **Maquettes :** Création d’interface utilisateur (UI) pour les trois rôles (Admin, Médecin, Patient).
* **Base de données :** Conception d’un schéma relationnel pour gérer les entités clés (patients, médecins, rendez-vous, traitements, paiements).

**5. Développement**

* **Backend :**
  + Langage : PHP (ou un framework comme Laravel).
  + Gestion des données via MySQL.
* **Frontend :**
  + Langages : HTML, CSS, JavaScript.
  + Framework : Bootstrap ou Tailwind pour le design responsive.
* **Fonctionnalités clés :**
  + Patients : Prise de rendez-vous, paiement en ligne.
  + Médecins : Gestion des consultations, prescriptions.
  + Admin : Gestion des utilisateurs, statistiques et rapports.

**6. Sécurité**

* Mise en œuvre du hachage des mots de passe.
* Validation des entrées utilisateur pour éviter les attaques (SQL Injection, XSS).
* HTTPS pour la sécurité des données échangées.

**7. Tests et validation**

* Test unitaire pour vérifier chaque module.
* Test d’intégration pour assurer l’interaction correcte entre les modules.
* Test utilisateur pour valider l’ergonomie et les fonctionnalités.

**8. Déploiement**

* Hébergement sur un serveur web (optionnel : Cloud ou hébergement local).
* Installation et configuration sur les ordinateurs/serveurs de l’hôpital.

**9. Maintenance**

* Suivi des bugs signalés.
* Mise à jour régulière pour améliorer la performance et ajouter de nouvelles fonctionnalités.

**10. Conclusion**

Le système permettra de transformer la gestion hospitalière en offrant une solution moderne, rapide et sécurisée, adaptée aux besoins actuels des hôpitaux.