

RAPPORT DE PROJET : APPLICATION DE GESTION DE MÉDICAMENTS

Imane Hadi

18 janvier 2025

Table des matières

1	Introduction	3
2	Objectifs du projet	3
3	Étapes de développement	3
3.1	Analyse des besoins	3
3.2	Conception de la base de données	3
3.3	Développement de l'application	4
3.3.1	Ajout de médicaments	4
3.3.2	Affichage des médicaments	4
3.3.3	Recherche des médicaments	4
3.3.4	Graphique des variations	4
4	Difficultés rencontrées	4
4.1	Gestion des erreurs SQL	4
4.2	Affichage des graphiques	5
4.3	Sécurisation de l'application	5
5	Solutions apportées	5
5.1	Gestion des erreurs SQL	5
5.2	Affichage des graphiques	5
5.3	Sécurisation de l'application	5
6	Conclusion	5

1 Introduction

Le but de ce projet était de concevoir une application web permettant la gestion de médicaments dans une pharmacie. L'application permet à un utilisateur de gérer les informations sur les médicaments, telles que leur nom, leur description, et leur prix. De plus, l'application permet la visualisation graphique des variations de certains paramètres des médicaments au fil du temps.

2 Objectifs du projet

Les objectifs principaux de ce projet étaient les suivants :

- Permettre l'ajout de nouveaux médicaments à la base de données.
- Permettre l'affichage des médicaments stockés dans la base de données.
- Fournir une fonctionnalité de recherche pour les médicaments.
- Offrir une fonctionnalité de visualisation graphique des variations des médicaments.

3 Étapes de développement

3.1 Analyse des besoins

La première étape du projet a consisté à analyser les besoins de l'application. Nous avons identifié les fonctionnalités suivantes comme étant essentielles :

- Une page d'accueil permettant de visualiser la liste des médicaments.
- Un formulaire d'ajout de médicament.
- Une barre de recherche pour trouver des médicaments en fonction de critères.
- Un graphique pour afficher les variations des prix ou d'autres paramètres des médicaments.

3.2 Conception de la base de données

La base de données a été conçue en utilisant MySQL. La table principale est la table `medicaments`, qui contient les informations suivantes :

- `id` (clé primaire)
- `nom`
- `description`
- `categorie`
- `dosage`

- prix
- date ajout
- quantite
- date de peremption

3.3 Développement de l'application

Le développement a été effectué en PHP pour la logique backend, avec HTML et CSS pour la présentation. Le code PHP est utilisé pour gérer les actions comme l'ajout, l'affichage, et la recherche des médicaments. La fonctionnalité graphique est réalisée à l'aide de la bibliothèque JavaScript `Chart.js` pour générer les graphiques dynamiques.

3.3.1 Ajout de médicaments

Le formulaire d'ajout permet à l'utilisateur de saisir le nom, la description, et le prix d'un médicament. Une fois soumis, ces informations sont envoyées à la base de données via une requête SQL `INSERT`

3.3.2 Affichage des médicaments

Les médicaments stockés dans la base de données sont récupérés à l'aide d'une requête SQL `SELECT` et affichés sur la page d'accueil sous forme de liste.

3.3.3 Recherche des médicaments

Une barre de recherche permet de filtrer les médicaments en fonction du nom ou d'autres critères. Cette fonctionnalité est réalisée en envoyant une requête de recherche à la base de données.

3.3.4 Graphique des variations

Un graphique dynamique montre l'évolution des prix des médicaments au fil du temps. Les données sont extraites de la base de données et affichées dans un graphique interactif en utilisant `Chart.js`.

4 Difficultés rencontrées

4.1 Gestion des erreurs SQL

Une des premières difficultés a été la gestion des erreurs SQL, notamment lors de l'insertion ou de la sélection des données dans la base de données. Les erreurs

liées à des requêtes incorrectes ou des conflits de données ont été courantes au début du projet.

4.2 Affichage des graphiques

L'intégration de graphiques dynamiques à partir des données SQL a été un défi. L'utilisation de `Chart.js` a nécessité une certaine courbe d'apprentissage, surtout en ce qui concerne la récupération et le formatage correct des données.

4.3 Sécurisation de l'application

Une autre difficulté a été de garantir la sécurité de l'application, en particulier pour éviter les injections SQL. Nous avons résolu ce problème en utilisant des requêtes préparées et en validant les entrées utilisateur.

5 Solutions apportées

5.1 Gestion des erreurs SQL

Nous avons utilisé des blocs `try-catch` pour intercepter les erreurs SQL et afficher des messages d'erreur appropriés. De plus, l'utilisation des requêtes préparées a permis de réduire les risques d'erreurs liées à des requêtes mal formées.

5.2 Affichage des graphiques

Pour résoudre le problème des graphiques, nous avons suivi des tutoriels détaillés sur `Chart.js` et avons veillé à ce que les données extraites de la base de données soient bien formatées pour être utilisées par la bibliothèque JavaScript.

5.3 Sécurisation de l'application

Pour sécuriser l'application, nous avons utilisé des requêtes SQL préparées avec des paramètres liés afin de prévenir les injections SQL. De plus, nous avons mis en place une validation côté serveur des entrées utilisateur pour garantir la qualité des données.

6 Conclusion

Ce projet a permis de mettre en pratique des compétences en développement web, en gestion de bases de données, et en utilisation de bibliothèques JavaScript

pour des visualisations interactives. Bien que des difficultés aient été rencontrées au cours du développement, les solutions mises en place ont permis de surmonter ces obstacles, aboutissant à une application fonctionnelle et sécurisée.