Projet de Fin de module

Intitulé :

Conception d’une application de gestion des evenements

Réalisé par :

-Radi Laakel Aya

-Lakbiri Fatima Zohra

-Bakkali Sara

Sous la Direction de :

-M. Elaachak Lotfi

Résumé

Ce projet s’inscrit dans un contexte pédagogique, mais répond à un besoin réel : fournir une plateforme simple, rapide et intuitive qui permet de créer, publier et gérer des événements, tout en permettant aux utilisateurs de consulter les événements disponibles et de s’y inscrire facilement.
Le projet est structuré selon une architecture client-serveur :

- **Le backend** est développé avec FastAPI, un framework Python moderne, rapide et asynchrone, qui expose une API REST pour gérer les différentes entités du système (utilisateurs, événements, inscriptions, etc.). **Le frontend** est réalisé avec
- React, une bibliothèque JavaScript permettant de construire une interface utilisateur interactive et réactive. **La base de données** utilisée est MySQL, un système relationnel qui permet de
- structurer les données efficacement (relations entre utilisateurs, événements, inscriptions...).

Ce projet a été l’occasion de mettre en pratique des compétences techniques variées : conception d’API REST, développement asynchrone, structuration d’une base de données, interactions entre client et serveur, gestion de l’authentification, et mise en place d’une interface utilisateur moderne.

Choix techniques

◆ FastAPI (Backend) :

FastAPI est un framework web moderne basé sur Python, connu pour sa rapidité et sa simplicité. Nos raisons principales pour le choisir sont :

- Performance élevée : Grâce à son support natif d'async/await, FastAPI offre des performances comparables à Node.js ou Go, ce qui est idéal pour construire des APIs rapides.
- Validation des données avec Pydantic : La validation des entrées (requêtes, formulaires...) est faite automatiquement, ce qui réduit les erreurs et simplifie le code.

◆ MySQL (Base de données)

Nous avons choisi MySQL pour sa robustesse et sa simplicité :

- Système relationnel mature : MySQL est un SGBD open-source très répandu, qui offre de bonnes performances et une fiabilité reconnue.
- Bon support d'intégration avec SQLAlchemy : Nous avons utilisé l'ORM SQLAlchemy pour interagir avec la base de données de manière orientée objet, tout en gardant la possibilité d'écrire des requêtes SQL en cas de besoin.

◆ React.js (Frontend)

Pour la partie interface utilisateur, nous avons opté pour React.js, une bibliothèque JavaScript orientée composants :

- Réactivité de l'interface : React offre une mise à jour rapide de l'interface utilisateur grâce à son Virtual DOM.
- Composants réutilisables : La logique de l'interface a été décomposée en composants réutilisables, ce qui facilite l'organisation du code et sa maintenance.

Introduction générale

1. Sujet du projet et problème résolu

2. Fonctionnalités implémentées

3.Activités Realisé

Chapitre I : Sujet du projet et problème résolu

Ce projet consiste à développer une application web de gestion des événements, visant à faciliter l'organisation d'événements (conférences, ateliers, séminaires, etc.) ainsi que l'inscription des participants. L'application s'adresse à deux types d'utilisateurs : les utilisateurs classiques (clients) qui peuvent consulter et s'inscrire à des événements, et les administrateurs qui peuvent gérer les événements et suivre les inscriptions.

Problème identifié :

Dans de nombreuses structures, la gestion des événements est encore réalisée manuellement ou avec des outils non adaptés (fichiers Excel, formulaires, mails), ce qui engendre :

- Un manque de centralisation des données.
- Une mauvaise traçabilité des inscriptions.
- Une absence de gestion des rôles (utilisateur/admin).
- Une perte de temps dans la mise à jour ou la diffusion des informations.

Objectif de la solution :

Nous avons conçu une solution numérique moderne reposant sur :

- Un backend en FastAPI, exposant une API REST sécurisée.
- Une interface utilisateur en React, simple et réactive.
- Une base de données MySQL, pour la persistance des événements, des utilisateurs et des inscriptions.

Cette application permet :

- Aux utilisateurs : de consulter les événements, voir les détails, s’inscrire et modifier leurs informations personnelles.
- Aux administrateurs : d’ajouter/modifier des événements, de gérer les utilisateurs et de suivre les inscriptions.

Chapitre II : Fonctionnalités implémentées

Authentification , Coté client , Coté Admin :

L’application propose des fonctionnalités adaptées à deux types d’utilisateurs : les clients et les administrateurs. Les clients peuvent créer un compte, se connecter, consulter la liste des événements, accéder aux détails, s’y inscrire et modifier leurs informations personnelles. Les administrateurs disposent de droits supplémentaires : ajouter d’autres admins, créer, modifier ou supprimer des événements, et consulter la liste des participants. L’authentification est sécurisée, avec une gestion des rôles bien définie. Les échanges frontend/backend se font via des appels API, garantissant une interaction fluide et dynamique entre React et FastAPI.

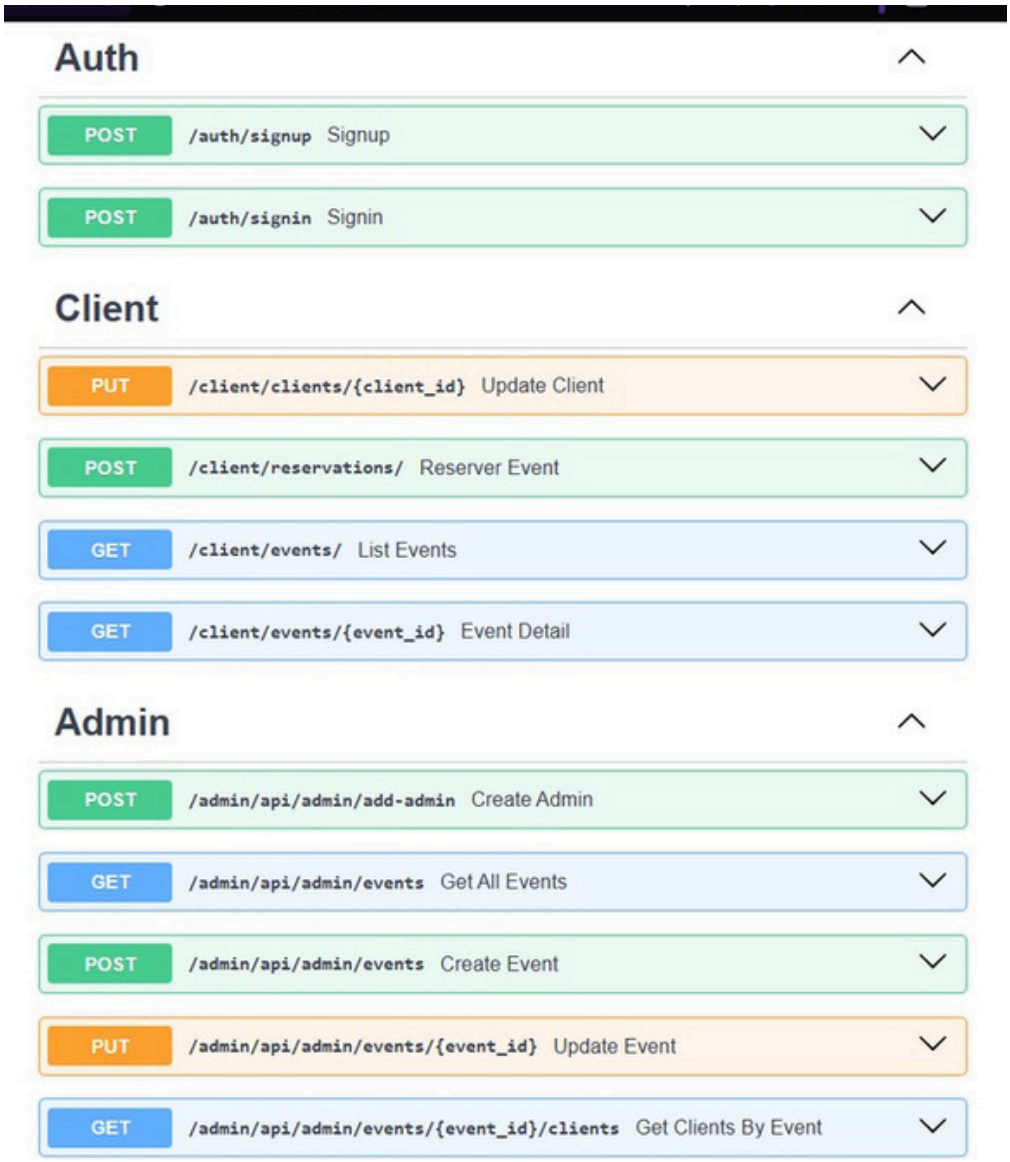
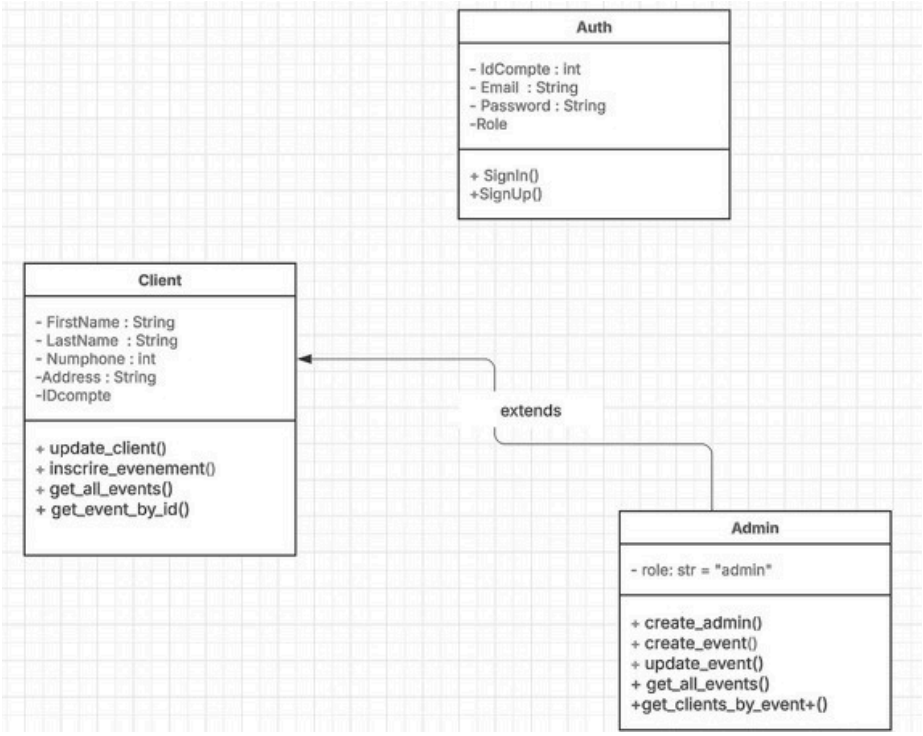
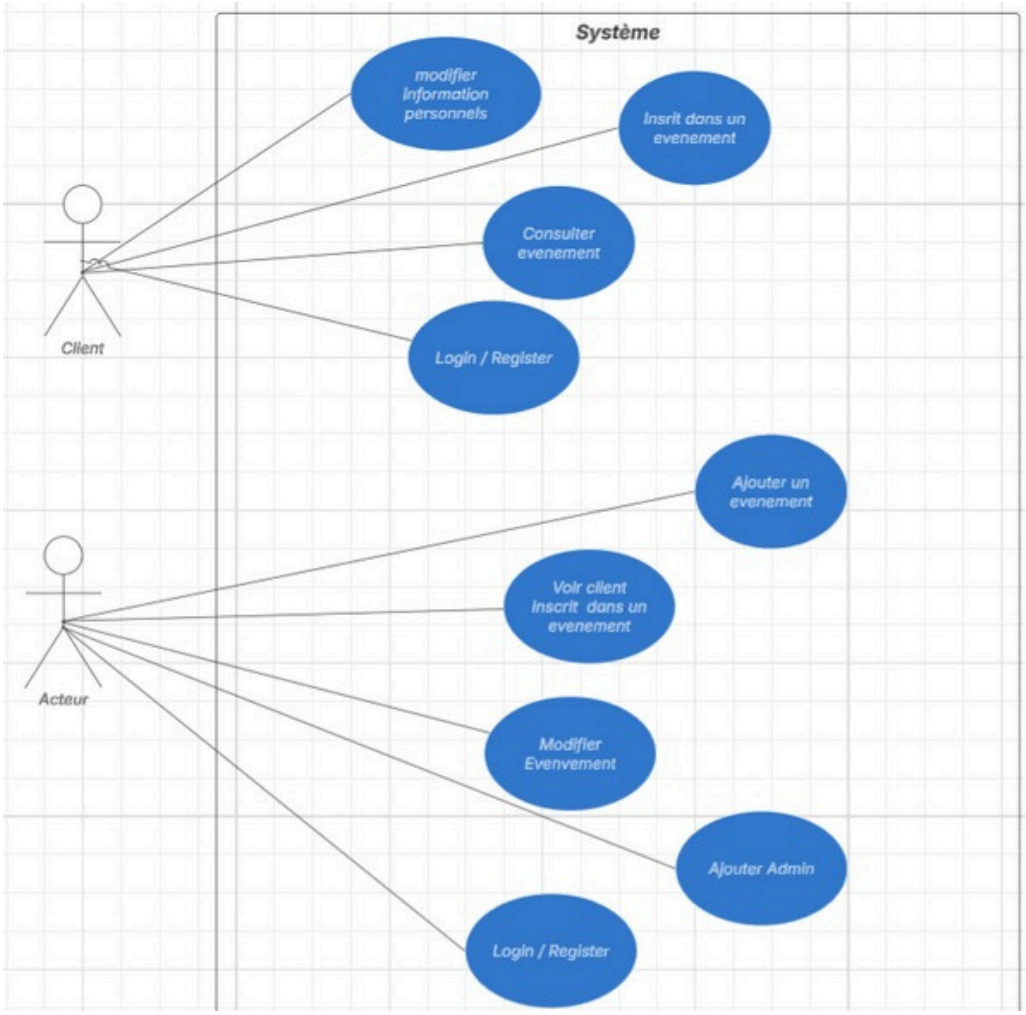


Diagramme UML :

1.Diagramme de classe



2. Diagramme de cas d'utilisation



Chapitre III : Défis rencontrés et solutions apportées:

2.Page LogIn:

Tout au long du développement du projet, plusieurs défis techniques et organisationnels se sont présentés. Le premier défi majeur a été l'intégration de FastAPI avec MySQL en mode asynchrone. Pour y remédier, nous avons utilisé SQLAlchemy avec le moteur asyncmy, ce qui nous a permis de tirer parti des performances asynchrones tout en gardant une structure ORM claire. Ensuite, la gestion des rôles et des autorisations a nécessité une attention particulière. Nous avons opté pour une séparation des routes dans des fichiers distincts (auth.py, user.py, admin.py), et chaque endpoint est protégé selon le rôle de l'utilisateur. Concernant l'authentification, l'utilisation du hashage des mots de passe via bcrypt a été mise en place pour garantir la sécurité des comptes. Côté frontend, l'un des défis a été la communication entre React et FastAPI : nous avons utilisé Axios pour gérer les requêtes HTTP et structuré les appels de manière cohérente pour respecter les réponses de l'API. Enfin, le suivi des inscriptions aux événements (relation plusieurs- à-plusieurs entre utilisateurs et événements) a demandé la création d'une table intermédiaire (Registration), que nous avons modélisée dans la base de données et manipulée via des fonctions dédiées. Ces solutions nous ont permis de maintenir une architecture claire, modulaire et évolutive tout en répondant aux exigences du projet.