

Réflexions initiale technologique

Pour le projet Arcadia, j'ai choisi d'utiliser plusieurs technologies qui, selon moi, répondent parfaitement aux besoins variés de notre application.

Bases de données : PostgreSQL et MongoDB

Pour la gestion des données, j'ai opté pour PostgreSQL et MongoDB. PostgreSQL est un système de gestion de base de données relationnelle open-source, connu pour sa robustesse et ses performances. Il est particulièrement utile pour les données structurées qui nécessitent des relations complexes, comme les comptes utilisateurs et les transactions. Ce qui me plaît avec PostgreSQL, c'est sa capacité à gérer des requêtes complexes et à maintenir l'intégrité des données grâce à sa conformité ACID.

En parallèle, MongoDB sera utilisé pour les données non structurées ou semi-structurées. MongoDB, en tant que base de données NoSQL, est extrêmement flexible et scalable. Il est parfait pour stocker des données sous forme de documents JSON, ce qui facilite la gestion des données variées et évolutives que nous pourrions rencontrer dans notre projet.

Langages de requêtes : SQL

Pour interagir avec PostgreSQL, j'utiliserai SQL. C'est un langage de requête bien établi et puissant, idéal pour tirer parti des fonctionnalités avancées de PostgreSQL.

Serveur : Express.js

Pour la partie serveur, mon choix s'est porté sur Express.js. Express.js est un framework pour Node.js qui est à la fois minimaliste et flexible. Il facilite la création de notre serveur web en gérant les routes, les middlewares et les requêtes HTTP de manière efficace. Cela rend notre application plus performante et plus facile à maintenir.

Frontend : EJS, TailwindCSS et JavaScript

Côté frontend, j'utiliserai EJS pour générer du HTML dynamique. EJS est pratique pour créer des templates HTML avec des données dynamiques, ce qui simplifie le rendu côté serveur.

Pour le style, TailwindCSS sera utilisé. C'est un framework CSS basé sur des classes utilitaires qui permet de concevoir rapidement des interfaces utilisateur élégantes et cohérentes. TailwindCSS rend le processus de stylisation plus flexible et maintenable.

Enfin, pour rendre le site interactif, j'utiliserai JavaScript côté client. JavaScript est indispensable pour manipuler le DOM, gérer les événements utilisateur et communiquer avec le serveur de manière asynchrone via AJAX. Cela permettra d'améliorer l'expérience utilisateur en rendant l'application plus réactive et dynamique.