

Compte rendu

GAUNET Aline et SAKHO Oumy

Réalisation du projet :

Tabnine est un outil d'aide à l'écriture de code, intégré dans mon éditeur (VSCode). Pendant le développement de ce projet React, il m'a proposé des suggestions pour compléter le code en fonction de ce que j'étais en train de faire. Cela m'a aidé à écrire des composants, des fonctions, et des styles plus rapidement. Cela a accéléré le développement en proposant des morceaux de code bien adaptés au contexte.

Par exemple, en écrivant `addToCart`, Tabnine m'a proposé un modèle pour ajouter un produit au panier avec le bon usage des états (`useState`). Quand j'ai écrit la fonction `fetchProduits`, Tabnine m'a suggéré le bon format pour utiliser `async/await` avec `fetch`, ce qui m'a évité des erreurs. En manipulant les produits dans le panier, Tabnine m'a aidé à structurer les fonctions avec `map`, `filter`, et d'autres méthodes.

Tabnine m'a aidé à coder plus rapidement en complétant automatiquement des parties courantes du code, comme les états et les boucles. Il a aussi réduit les erreurs en suggérant les bons noms de variables et en évitant des oublis dans les crochets de `useEffect`. De plus, ses suggestions m'ont appris des méthodes plus efficaces, comme l'utilisation du `spread operator` pour mettre à jour le panier.

Optimisation de la qualité du code :

Dans le cadre de l'optimisation de la qualité du code, nous avons sélectionné deux exercices parmi les quatre proposés : Défis de codage avec l'IA et Sécurité et Refactoring, respectivement réalisés en Python et en JavaScript.

Pour le premier exercice, nous avons utilisé l'intelligence artificielle pour générer des propositions de code adaptées, accélérer le processus de développement et renforcer les aspects de sécurité du projet. L'outil Tabnine s'est révélé particulièrement utile pour documenter et expliquer les lignes de code générées, facilitant ainsi la compréhension et la collaboration au sein de l'équipe. L'IA a démontré son efficacité notamment lors de l'édition de code.

Par exemple, pour implémenter une fonction de tri des utilisateurs, il nous a suffi de définir une structure initiale, comme `def tri_utilisateur()`, et d'utiliser les suggestions d'édition de l'IA. En précisant nos besoins — par exemple, trier les utilisateurs par âge ou par nom — l'outil a généré une solution fonctionnelle, optimisée et adaptée à nos attentes. Son impact principal réside dans la réduction du temps de développement et l'amélioration de la qualité du code. Elle nous a permis de nous concentrer davantage sur la logique métier et la conception globale, tout en assurant une documentation claire et une sécurité renforcée grâce à des vérifications automatiques et des suggestions adaptées.