
Fiche de veille technologique

BTS SIO SLAM

Sommaire

1. Présentation de la veille technologique.....	3
2. Thème choisi.....	3
3. Objectifs de la veille	3
4. Méthodologie.....	4
5. Sources principales	4
6. Analyse des tendances.....	4
7. Exploitation personnelle.....	5
8. Conclusion.....	5

1. Présentation de la veille technologique

La veille technologique est une démarche proactive qui permet de se tenir informé des innovations techniques, des nouvelles tendances, et des évolutions dans un domaine spécifique. En tant que futur développeur, cette pratique est indispensable pour rester compétitif et adapter ses compétences aux besoins du marché.

Elle consiste à rechercher, trier et organiser les informations utiles, afin de gagner du temps et de ne pas se laisser dépasser par l'évolution rapide des technologies. Cela permet aussi de faire les bons choix d'outils et de langages dans ses projets professionnels ou personnels.

2. Thème choisi

Les frameworks JavaScript modernes, avec un focus sur React.js

React est l'un des frameworks les plus utilisés en développement front-end. Il est développé par Meta (ex-Facebook) et permet de créer des interfaces utilisateur dynamiques et réactives. Cette veille porte sur les évolutions de React, ses concurrents, et les tendances associées (ex : Next.js, Tailwind CSS, etc.).

J'ai choisi ce thème car React est aujourd'hui incontournable dans les offres d'emploi en développement web. Il est souvent utilisé avec d'autres outils comme TypeScript, Vite ou Tailwind CSS, qui apportent des gains en performance et en productivité. Comprendre cet écosystème me permet de mieux construire mon portfolio et d'adopter de bonnes pratiques.

3. Objectifs de la veille

Cette veille a pour but de mieux comprendre les évolutions actuelles du développement front-end, afin d'adapter mes compétences aux pratiques les plus demandées. Les objectifs sont :

- Identifier les tendances autour des frameworks front-end.
- Suivre les mises à jour majeures de React.js.
- Analyser les outils complémentaires (Next.js, Vite, TypeScript...).
- Adapter ma formation en fonction des attentes du marché du travail.

4. Méthodologie

Étape	Détail
Délimitation	Suivi des évolutions de React et de son écosystème dans un cadre professionnel.
Sources utilisées	Daily.dev, YouTube (Micode, Melvynx), GitHub, Sites officiels (reactjs.org)
Outils utilisés	Google Alertes, Feedly, Canva pour la présentation, Daily.dev
Fréquence	Une session de 20 à 30 minutes chaque semaine.
Organisation	Conservation des ressources intéressantes dans mes favoris ou dans un dossier dédié sur mon navigateur ou directement sur Daily.dev

5. Sources principales

- <https://reactjs.org/> — site officiel de React
- <https://vercel.com/> — framework Next.js
- <https://daily.dev> — agrégateur d'articles tech
- YouTube : chaînes **Micode**, **Melvynx** (veille tech francophone)
- YouTube : chaînes **The Net Ninja** / **Fireship** (veille tech américaine)
- GitHub — projets open source en React/Next.js

6. Analyse des tendances

Le monde du développement front-end évolue rapidement, avec des outils et des pratiques qui se renouvellent constamment. Voici les grandes tendances observées durant ma veille :

- **React** reste le framework le plus utilisé dans l'écosystème JavaScript. Il continue d'être largement adopté par les entreprises pour sa flexibilité et sa communauté.
- **Next.js**, basé sur React, s'impose comme la solution idéale pour les applications fullstack, notamment grâce au support du rendu côté serveur (SSR) et de l'hébergement simplifié avec Vercel.
- Le nouveau bundler **Vite**, beaucoup plus rapide que Webpack, devient la nouvelle norme dans les projets modernes.
- L'association **React + Tailwind CSS** permet de créer des interfaces modernes, responsives et cohérentes rapidement, ce qui en fait un combo de plus en plus utilisé.

- **TypeScript** est désormais presque incontournable avec React, car il permet de sécuriser le code, éviter les erreurs et améliorer la lisibilité sur les projets d'équipe.
- Enfin, la **performance** est devenue un critère central : chargement rapide, composants optimisés, lazy loading et bonnes pratiques d'accessibilité sont au cœur des attentes.

Ces tendances influencent directement les compétences attendues chez les développeurs front-end aujourd'hui.

7. Exploitation personnelle

Grâce à cette veille, j'ai pu améliorer ma manière de coder et mieux organiser mes projets:

- J'ai réalisé mon portfolio avec **React**, **Vite** et **Tailwind CSS**, en suivant les tendances actuelles.
- J'ai utilisé **GSAP** pour les animations et j'ai commencé à intégrer un **carrousel** pour mettre en valeur mes projets.
- Je continue à suivre l'actualité grâce à **Daily.dev** et à des créateurs comme **Micode** ou **Melvynx** sur YouTube, ce qui m'aide à progresser régulièrement.

8. Conclusion

Cette veille m'a permis de mieux comprendre les outils que j'utilise, de découvrir de nouvelles pratiques, et surtout d'améliorer concrètement ma façon de coder. Elle m'a aidé à structurer mon apprentissage, à faire évoluer mon portfolio et à mieux me préparer au monde professionnel.