

QUAGLIERI  
LISA

MSPR 1  
**LIVE EVENT**

# Table des matières

1. Introduction générale
2. Audit UX pour le site Live Event
  - 2.1. Introduction à l'audit UX
  - 2.2. Benchmark des plateformes similaires
  - 2.3. Application des 12 heuristiques de Jakob Nielsen
  - 2.4. Recommandations UX pour Live Event
3. Analyse des besoins et spécifications
  - 3.1. Synthèse des besoins exprimés dans le cahier des charges
  - 3.2. Fonctionnalités essentielles
  - 3.3. Innovations proposées
4. Veille technologique sur les technologies front-end et CMS
  - 4.1. Comparatif des technologies front-end
  - 4.2. Avantages de TailwindCSS pour la gestion des styles
  - 4.3. Gestion de contenu avec WordPress et MySQL
  - 4.4. Comparatif des plateformes de déploiement : Vercel vs Netlify
5. Sécurité et conformité réglementaire
  - 5.1. Analyse des résultats SonarQube
  - 5.2. Bonnes pratiques de sécurité implémentées
  - 5.3. Conformité RGPD et protection des données
  - 5.4. Recommandations pour renforcer la sécurité
6. Design Thinking et prototypage
  - 6.1. Phase de réflexion
  - 6.2. Phase de prototypage
7. Recommandations techniques et méthodologiques
  - 7.1. Stratégie de développement front-end et back-end
  - 7.2. Intégration des tests automatisés
  - 7.3. Plan de déploiement et maintenance continue
8. Conclusion générale
9. Annexes

# Introduction général

Dans un monde de plus en plus connecté, où les événements et les expériences culturelles sont au cœur des interactions sociales, le besoin d'applications web et mobiles performantes et immersives est plus pertinent que jamais. Ce projet s'inscrit dans le cadre du développement d'une plateforme digitale pour le festival "Live Event", une initiative visant à offrir une expérience utilisateur fluide, engageante et conforme aux exigences des festivaliers modernes.

La veille documentaire réalisée vise à explorer et analyser les technologies, méthodologies et bonnes pratiques nécessaires pour concevoir une application répondant aux objectifs définis par le cahier des charges. Elle s'articule autour de plusieurs axes clés : l'optimisation de l'expérience utilisateur (UX), la sélection de technologies front-end et back-end adaptées, la gestion de contenu via des CMS robustes comme WordPress, ainsi que l'hébergement et le déploiement sur des plateformes modernes comme Vercel ou Netlify. En parallèle, des aspects fondamentaux tels que la sécurité, l'accessibilité et la conformité aux normes actuelles (comme le RGPD) sont abordés pour garantir une solution professionnelle, pérenne et évolutive.

Cette analyse fournit les outils et recommandations nécessaires pour concrétiser une application moderne, interactive et centrée sur l'utilisateur, tout en respectant les standards techniques et les attentes du projet.

# Audit UX pour le site Live Event en vue de la création d'une application web et web mobile

## 1. Introduction

Cet audit vise à évaluer l'expérience utilisateur (UX) actuelle du site **Live Event** afin de concevoir une **application web et web mobile** immersive et informative pour le festival. L'objectif est d'offrir une expérience fluide et engageante répondant aux attentes des festivaliers, en accord avec les objectifs et spécifications du cahier des charges fourni.

Les recommandations qui en découlent permettront d'améliorer **l'ergonomie, la navigation, l'accessibilité et l'interaction utilisateur**, tout en adoptant les meilleures pratiques du marché.

## 2. Benchmark UX

Pour comparer les bonnes pratiques en matière d'UX, plusieurs plateformes références ont été analysées :

- **Eventbrite** : Référence en billetterie et gestion d'événements.
- **Hopin** : Plateforme spécialisée pour événements virtuels/hybrides.
- **Tomorrowland App** : Application dédiée aux festivals.
- **Live Nation** : Application mobile dédiée aux concerts.

## Comparaison des plateformes

Critères d'analyse	Eventbrite	Hopin	Tomorrowland App	Live Nation
<b>Ergonomie &amp; navigation</b>	Interface intuitive avec navigation fluide.	Organisation bien structurée, mais dense.	Navigation fluide, optimisée pour mobile.	Navigation claire, accès rapide à l'information.
<b>Contenu &amp; accessibilité</b>	Contenus bien hiérarchisés, adaptés à tous.	Informations ciblées pour pros et particuliers.	Informations immersives sur les artistes, horaires et cartes.	Informations riches sur concerts et artistes.
<b>Interaction utilisateur</b>	Interaction fluide avec billetterie intégrée.	Interactivité grâce aux sessions virtuelles.	Expérience immersive via carte interactive et notifications.	Notifications d'événements en temps réel.
<b>Design graphique</b>	Design professionnel et attrayant.	Design moderne mais utilitaire.	Design vibrant et immersif.	Design professionnel et dynamique.
<b>Outils &amp; fonctionnalités</b>	Billetterie intégrée, calendrier.	Vidéoconférences, networking.	Notifications, cartes interactives, playlists.	Géolocalisation, notifications, suivi des artistes.

### 3. Application des 12 heuristiques de Jakob Nielsen

Heuristique	Observation	Recommandation
<b>Visibilité de l'état du système</b>	Retour utilisateur limité (pas d'indicateur de progression ou de confirmation d'action).	Intégrer des barres de progression et des notifications claires lors des actions.
<b>Correspondance entre le système et le monde réel</b>	Termes parfois peu explicites (ex. "informations générales").	Utiliser un vocabulaire clair et orienté utilisateur.
<b>Contrôle utilisateur et liberté</b>	Pas de bouton "retour" ou "annuler" sur certaines pages.	Ajouter des options de navigation explicites.
<b>Cohérence et normes</b>	Incohérence dans les styles d'icônes et typographie.	Harmoniser le design et respecter les normes UI.
<b>Prévention des erreurs</b>	Absence de validation proactive sur les champs de formulaire.	Ajouter des vérifications instantanées.
<b>Reconnaissance plutôt que rappel</b>	Trop de navigation pour retrouver une information.	Ajouter des raccourcis et boutons fixes.
<b>Flexibilité et efficacité</b>	Pas de personnalisation avancée.	Intégrer filtres et notifications personnalisées.
<b>Esthétisme et design minimaliste</b>	Manque de dynamisme visuel.	Intégrer des animations légères et visuels attractifs.
<b>Aide à l'utilisateur</b>	Messages d'erreur peu explicites.	Proposer des messages clairs avec suggestions.

<b>Documentation utile</b>	Pas de FAQ visible.	Intégrer une section d'aide avec recherche rapide.
<b>Optimisation pour mobile</b>	Fonctionnalité mobile présente mais peu fluide.	Améliorer la navigation tactile et la réactivité.
<b>Performance technique</b>	Temps de chargement perfectibles.	Optimiser les requêtes et réduire les ressources lourdes.

## 4. Analyse des besoins et recommandations

### Besoins clés du cahier des charges

- **Carte interactive** : Géolocalisation des scènes, tri par catégories.
- **Programme des concerts** : Filtres par date, heure, scène.
- **Informations pratiques** : Accès, FAQ.
- **Intégration des réseaux sociaux**.

### Recommandations prioritaires

- **Ergonomie et design** : Un design immersif et attractif pour captiver les utilisateurs.
- **Interaction utilisateur** : Notifications et rappels personnalisés.
- **Accessibilité** : Navigation fluide sur mobile.

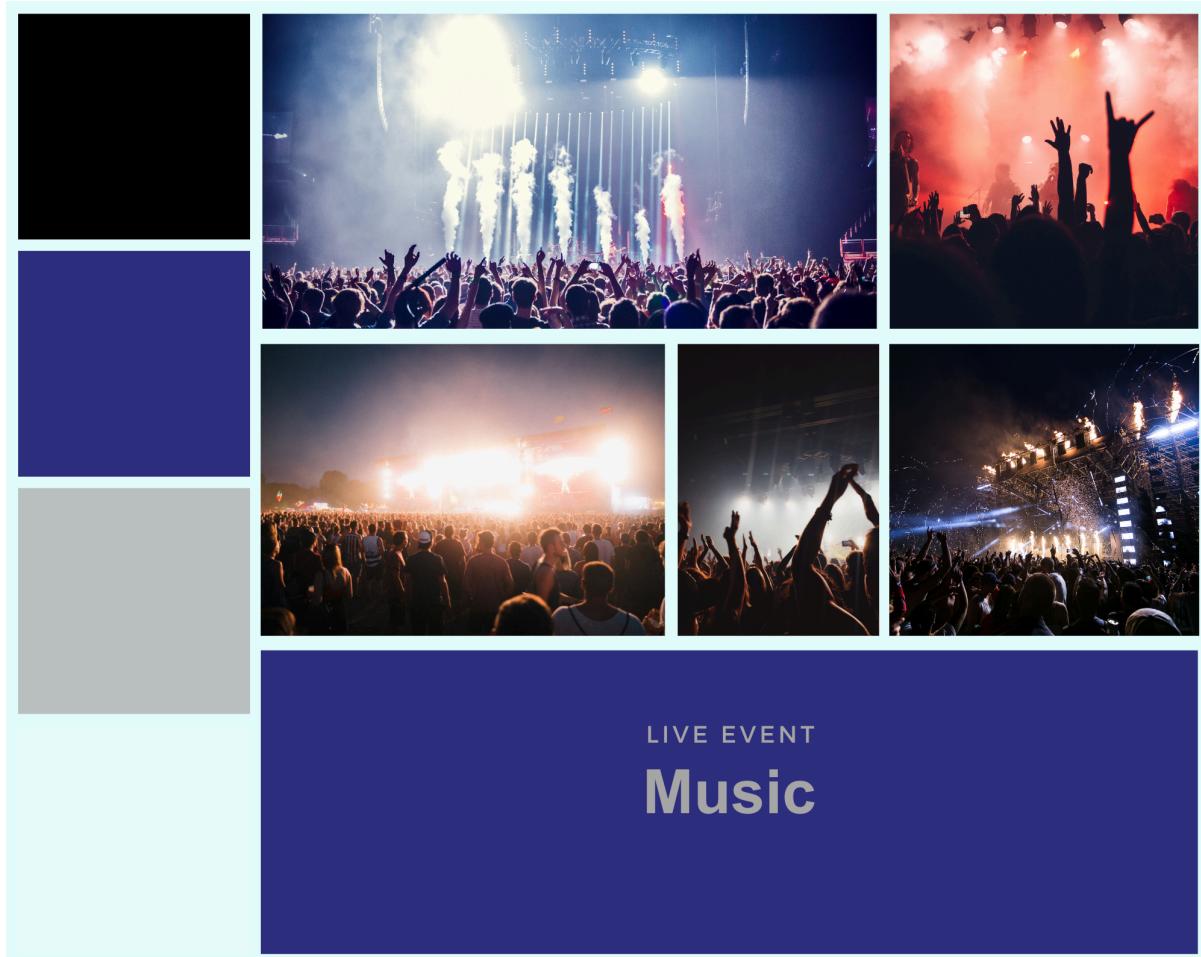
### Innovations possibles

- **Personnalisation** : Favoris (artistes, concerts).
- **Immersion** : Contenus multimédias (vidéos, teasers).
- **Expérience en temps réel** : Notifications push des concerts imminents.

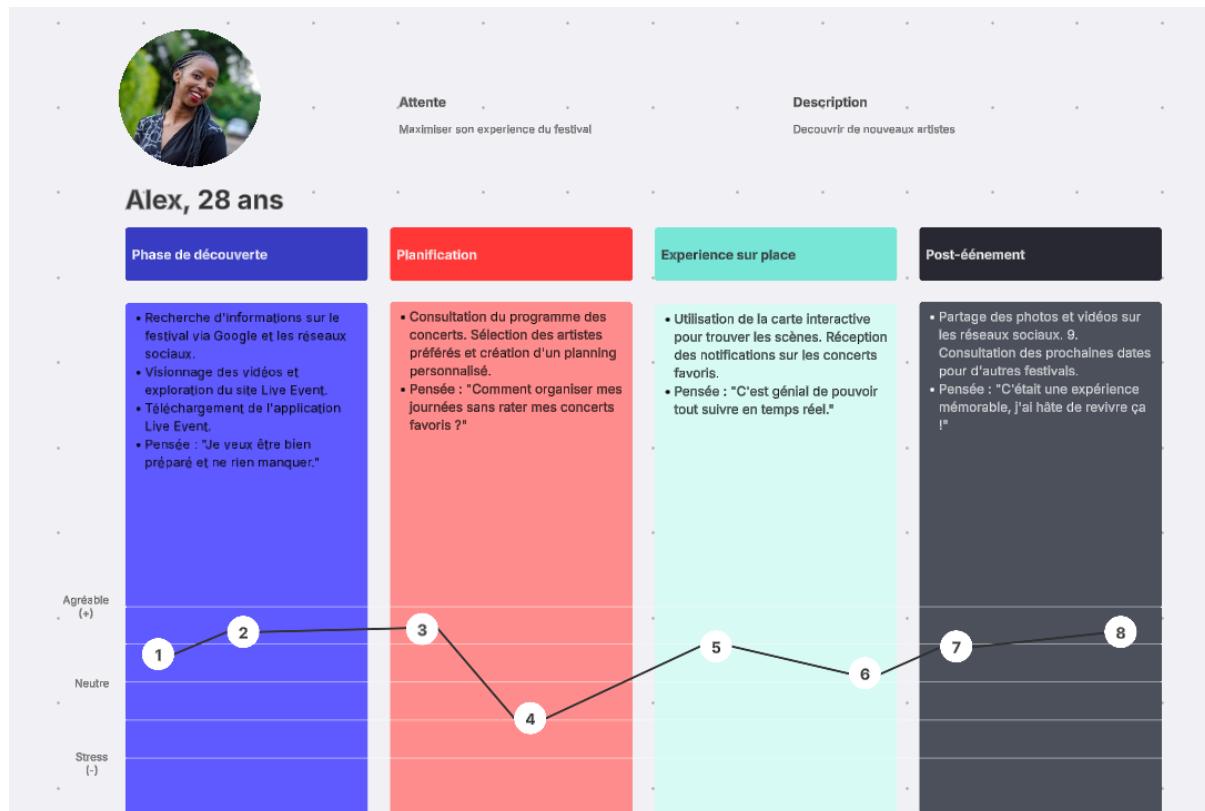
## 5. Design Thinking pour Live Event

- **Moodboard** : Recueil d'inspirations visuelles pour définir l'identité graphique de l'application.
- **Parcours utilisateur** : Cartographie des différentes étapes d'interaction avec l'application.
- **Arborescence du site**
- **Wireframing et storyboarding** : Création de premiers schémas conceptuels.

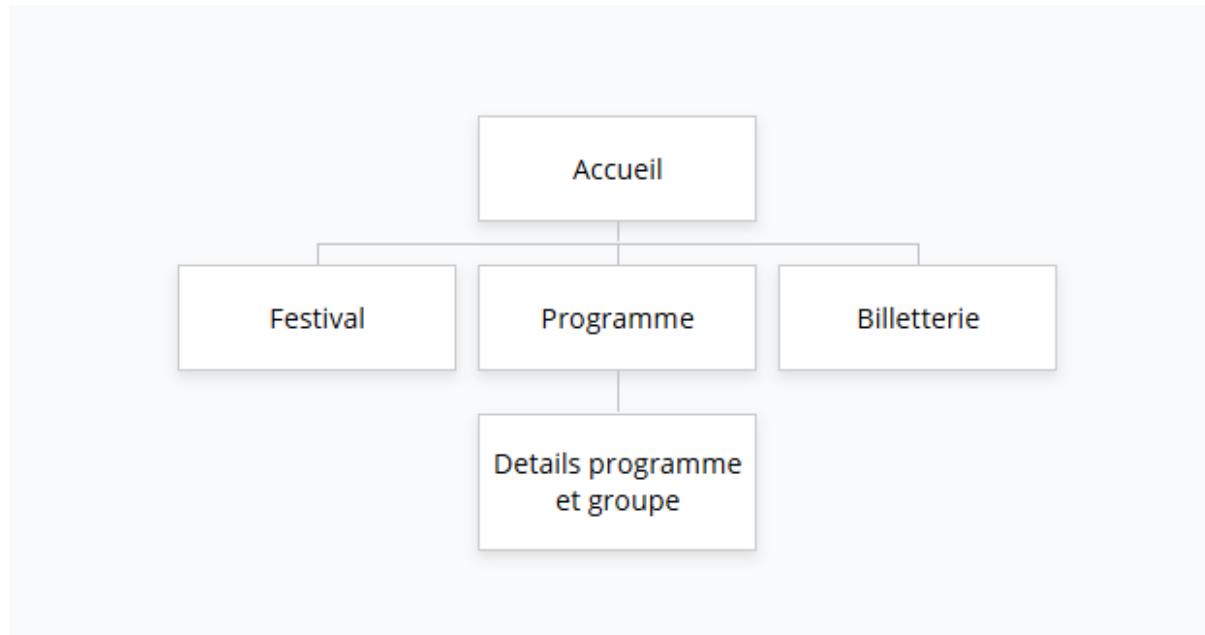
**Moodboard :**



## customer Journey Map :



<https://www.gloomaps.com/2GdMtmjYTs>



🔗 Wireframe (Annexe)

## Prototypage

- Création des prototypes interactifs permettant de tester les fonctionnalités et le design avant développement.
- Tests utilisateurs pour ajustements et itérations.

*(Voir prototype en annexe)*

## 6. Conclusion

Cet audit met en avant les axes d'amélioration UX pour **Live Event**. En appliquant ces recommandations et en intégrant un processus de **Design Thinking**, l'application offrira une expérience immersive et engageante, renforçant la satisfaction des festivaliers.

# **Veille Documentaire sur les Technologies Front-End et CMS**

## **Introduction générale**

Dans un monde numérique en constante évolution, le choix des technologies de développement web joue un rôle crucial dans la réussite des projets. Les frameworks front-end, les outils de gestion de bases de données et les plateformes de déploiement constituent autant d'éléments stratégiques pour optimiser la création et la gestion d'applications modernes. Cette veille documentaire présente une analyse approfondie de quatre technologies incontournables : **ReactJS** et **TailwindCSS** pour le développement front-end, **WordPress** associé à **MySQL** et **phpMyAdmin** pour la gestion de contenu, et enfin **Vercel** et **Netlify** pour le déploiement d'applications web. L'objectif est d'offrir une vision claire et comparative de ces solutions afin d'aider les développeurs à faire des choix technologiques pertinents et adaptés à leurs besoins.

### **1. Introduction aux Technologies Front-End**

Le développement front-end est un domaine en constante évolution, avec des frameworks et bibliothèques toujours plus performants. Aujourd'hui, trois technologies majeures dominent le marché : **ReactJS**, **Angular** et **Vue.js**. Cette veille documentaire met en avant les avantages de **ReactJS** et son écosystème, ainsi que l'utilisation de **TailwindCSS** comme solution optimisée pour la gestion des styles CSS.

## 2. Comparatif des Technologies Front-End

### 2.1. Pourquoi choisir ReactJS ?

**ReactJS** est une bibliothèque JavaScript développée par Facebook qui s'est imposée comme une référence incontournable pour la création d'interfaces utilisateur dynamiques. Comparé aux autres frameworks comme Angular et Vue.js, ReactJS offre une approche plus flexible et optimisée.

Critère	ReactJS	Angular	Vue.js
<b>Performance</b>	Optimisée grâce au Virtual DOM	Moins performant en raison du Two-Way Binding	Bon compromis, mais moins optimisé que React
<b>Courbe d'apprentissage</b>	Relativement simple pour les développeurs familiers de JavaScript	Complexe, nécessite d'apprendre TypeScript et d'autres concepts	Facile à apprendre, mais peut devenir complexe dans de grands projets
<b>Flexibilité</b>	Très flexible, sans structure imposée	Très structuré, suivant une approche MVC stricte	Flexible mais structuré, hybride
<b>Communauté et écosystème</b>	Très large, nombreuses bibliothèques et support continu	Grande communauté, principalement orientée entreprise	Bonne communauté, plus restreinte
<b>Modularité</b>	Basé sur des composants réutilisables	Modulaire, mais plus rigide	Très modulaire, avec quelques dépendances

## 2.2. Architecture de ReactJS

ReactJS repose sur une architecture déclarative et modulaire qui s'appuie sur :

- **Composants réutilisables** : Chaque partie de l'interface utilisateur est un composant indépendant, facilitant la réutilisation et la maintenance.
- **Virtual DOM** : Optimisation des mises à jour en ne modifiant que les parties nécessaires.
- **Unidirectional Data Flow** : React impose un flux de données descendant, rendant l'état plus prévisible.
- **Hooks et gestion d'état** : Simplifie la gestion d'état avec des hooks tels que `useState`, `useContext` et `useReducer`, et s'appuie sur des bibliothèques comme Redux ou Recoil.

## 2.3. L'écosystème React

L'écosystème de React est riche et comprend de nombreux outils facilitant le développement front-end :

Domaine	Outils et Packages
Routage	react-router-dom
Gestion de l'état	redux, recoil, zustand
Requêtes HTTP	axios, react-query
Tests et QA	jest, react-testing-library, cypress
Animations	framer-motion, react-spring
UI Frameworks	material-ui, chakra-ui, antd
Optimisation	react-virtualized, react-window

## 2.4. Différences entre JavaScript et JSX

JSX (JavaScript XML) est une extension de syntaxe pour JavaScript utilisée dans React. Elle permet d'écrire des balises HTML directement dans le code JavaScript, rendant le développement plus intuitif.

Caractéristique	JavaScript	JSX
<b>Syntaxe</b>	Basée sur <code>document.createElement()</code> ou <code>innerHTML</code>	Syntaxe similaire à HTML intégrée dans JavaScript
<b>Lisibilité</b>	Plus verbeux, moins intuitif pour le rendu d'UI	Plus clair et proche du HTML standard
<b>Performance</b>	Exécuté nativement par le moteur JavaScript	Transformé en <code>React.createElement()</code> par Babel
<b>Sécurité</b>	Nécessite une gestion stricte de l'injection	Sécurisé par défaut contre les attaques XSS

## 2.5. Pourquoi choisir TailwindCSS ?

**TailwindCSS** est un framework CSS utilitaire qui permet de concevoir des interfaces modernes avec une grande flexibilité.

Critère	TailwindCSS	Bootstrap
<b>Personnalisation</b>	Très élevé, aucune contrainte de design	Structuré, mais limité par les classes prédéfinies
<b>Poids du CSS</b>	Optimisé grâce à PurgeCSS, ne charge que les styles utilisés	Plus lourd en raison des styles par défaut
<b>Facilité d'apprentissage</b>	Nécessite d'apprendre les classes utilitaires	Facile grâce aux composants préconçus

## 2.6. Conclusion (Front-End)

En combinant ReactJS et TailwindCSS, les développeurs disposent d'un écosystème puissant et performant pour construire des applications modernes, scalables et optimisées.

## 4. WordPress avec MySQL et phpMyAdmin

**WordPress** reste le CMS le plus utilisé au monde, offrant une infrastructure de gestion de contenu à la fois puissante et flexible. Associé à **MySQL** pour la gestion des bases de données, il permet de stocker les informations de manière structurée. L'utilisation de **phpMyAdmin** offre quant à elle une interface graphique conviviale pour administrer la base de données.

### Statistiques sur les CMS en 2024

Selon W3Techs, voici la répartition des parts de marché :

- **WordPress** : 63,5 %
- **Shopify** : 5,8
- **Wix** : 3,8 %
- **Squarespace** : 3,0 %
- **Joomla!** : 2,5 %
- **Drupal** : 1,8 %
- **Autres CMS** : 19,6 %

WordPress domine largement le marché, représentant plus de 43 % de tous les sites web.

## Pourquoi choisir WordPress ?

- **Fonctionnalités** : Utilisé par plus de 60 % des sites CMS, WordPress offre une interface intuitive même pour les non-développeurs, une extensibilité via plus de 58 000 plugins et une compatibilité SEO grâce à des outils intégrés comme Yoast SEO.
- **Communauté et support** : Une large communauté et une documentation abondante facilitent l'apprentissage et la résolution de problèmes.
- **Adaptabilité** : Convient aussi bien aux blogs qu'aux sites e-commerce, portfolios, etc.

## Installation et configuration dans un environnement containerisé

Dans notre configuration, WordPress est déployé dans un conteneur Docker, et son accès se fait via l'adresse :

<http://158.69.54.81:84/wp-admin>

Cette approche permet de bénéficier d'un environnement isolé et cohérent, où Apache et les autres services web sont encapsulés dans l'image Docker, sans installation directe sur le serveur hôte.

La configuration suit ces étapes :

1. **Téléchargement et préparation** : Téléchargez la dernière version de WordPress depuis wordpress.org et intégrez-la dans une image Docker.
2. **Installation de MySQL** : Déployez un conteneur MySQL pour gérer la base de données.
3. **Utilisation de phpMyAdmin** : (optionnel) pour une gestion graphique de la base de données.
4. **Orchestration via Docker Compose** : Utilisez un fichier Docker Compose pour démarrer automatiquement WordPress, MySQL et phpMyAdmin, assurant une interconnexion fluide entre les services.

## 5. Comparatif Vercel vs Netlify

**Vercel** et **Netlify** sont deux plateformes populaires pour le déploiement des applications front-end modernes, notamment pour des projets basés sur React, Next.js et d'autres frameworks JavaScript.

Fonctionnalité	Vercel	Netlify
<b>Performance</b>	Optimisé pour le rendu statique et les API serverless	Optimisé pour le rendu statique et les API serverless
<b>Simplicité d'utilisation</b>	Déploiement automatique via GitHub, GitLab et Bitbucket	Déploiement automatique via GitHub, GitLab et Bitbucket
<b>Scalabilité</b>	Infrastructure serverless gérant automatiquement la montée en charge	Infrastructure serverless avec gestion automatique des CDN
<b>Prise en charge des frameworks</b>	Support natif de Next.js, React, Vue, Angular et Svelte	Support de Next.js, React, Vue, Gatsby, etc.
<b>CDN et caching</b>	Mise en cache intelligente et Edge Functions	Mise en cache CDN avancée et Edge Functions
<b>HTTPS et domaines</b>	Configuration automatique du SSL et gestion des domaines	Configuration automatique du SSL et gestion des domaines
<b>Fonctionnalités supplémentaires</b>	Mode preview dynamique pour Next.js	Netlify Functions pour API serverless

## Pourquoi choisir Vercel ?

- Intégration native avec Next.js, optimisée pour le SSR et le Static Site Generation.
- Interface simple et rapide pour le déploiement.

## Pourquoi choisir Netlify ?

- Idéal pour les sites statiques et les applications Jamstack.
- Facilité d'ajouter des fonctionnalités backend avec Netlify Functions.

## Docker et Infrastructure du Back-End

Afin d'assurer une cohérence entre les environnements de développement, de test et de production, nous avons opté pour l'utilisation de Docker. La containerisation offre plusieurs avantages majeurs pour la gestion du back-end :

- **Isolation des composants** : Chaque service (serveur web, base de données, API) est isolé dans un conteneur, garantissant une indépendance totale entre les composants et une gestion simplifiée des dépendances.
- **Portabilité** : Les conteneurs Docker assurent que l'application fonctionne de manière identique sur tout environnement supportant Docker, facilitant grandement le déploiement sur le serveur de production.
- **Scalabilité et maintenance** : La gestion des mises à jour et le déploiement de nouvelles versions sont simplifiés grâce aux images Docker, permettant une évolution progressive de l'infrastructure sans impact majeur pour l'utilisateur final.
- **Gestion simplifiée des configurations** : L'utilisation de fichiers de configuration via Docker Compose permet d'orchestrer plusieurs conteneurs et de définir clairement les interactions entre les différents services.

## Déploiement sur un Serveur Personnel sous Debian avec Docker Compose

Pour l'hébergement du back-end, nous avons choisi un serveur personnel basé sur Debian, offrant ainsi :

- **Stabilité et Sécurité** : Debian est reconnu pour sa robustesse, faisant de lui un système d'exploitation idéal pour un environnement serveur.
- **Contrôle Total** : L'hébergement sur un serveur personnel permet une configuration fine et personnalisée en fonction des exigences spécifiques du projet.
- **Optimisation des Ressources** : Le déploiement de conteneurs Docker sur Debian permet une gestion efficace des ressources matérielles, assurant des performances optimales.

Dans notre configuration, aucun Apache n'est installé directement sur le serveur hôte. Tous les services, y compris WordPress (accessible via <http://158.69.54.81:84/wp-admin>) et ses dépendances, sont exécutés dans des conteneurs Docker. Grâce à Docker Compose, tous les services se démarrent et s'interconnectent automatiquement, assurant ainsi une orchestration fluide et homogène de l'infrastructure.

La mise en place s'effectue en plusieurs étapes :

1. **Installation de Docker et Docker Compose** sur le serveur Debian.
2. **Création d'images personnalisées** pour chaque service du back-end, intégrant les dépendances et configurations nécessaires.  
**Déploiement des conteneurs** via Docker Compose, garantissant le démarrage et l'interconnexion automatique des services.
3. **Mise à jour et maintenance** grâce à des stratégies de déploiement continu, assurant la pérennité et la sécurité de l'infrastructure.

En résumé, l'intégration de Docker, orchestrée via Docker Compose sur un serveur personnel Debian, permet de disposer d'une infrastructure flexible, sécurisée et facilement maintenable, parfaitement adaptée aux besoins évolutifs du projet "Live Event".

## 6. Conclusion générale

L'essor des technologies web modernes permet aux développeurs d'accéder à des solutions toujours plus performantes et flexibles. **ReactJS**, associé à **TailwindCSS**, offre une approche modulaire et efficace pour la création d'interfaces dynamiques. **WordPress**, couplé à **MySQL** et **phpMyAdmin**, reste un choix incontournable pour la gestion de contenu — et, dans notre cas, est déployé de manière containerisée pour offrir une accessibilité centralisée via <http://158.69.54.81:84/wp-admin>. Enfin, les plateformes de déploiement comme **Vercel** et **Netlify** permettent de simplifier et d'accélérer la mise en production des applications.

En combinant ces outils, il est possible de concevoir des solutions web modernes, scalables et adaptées aux exigences actuelles du marché. Cette veille technologique constitue ainsi un guide essentiel pour les développeurs souhaitant rester à la pointe des innovations et optimiser leurs choix stratégiques en matière de développement web.

# Sécurité : Analyse et recommandations basées sur SonarQube

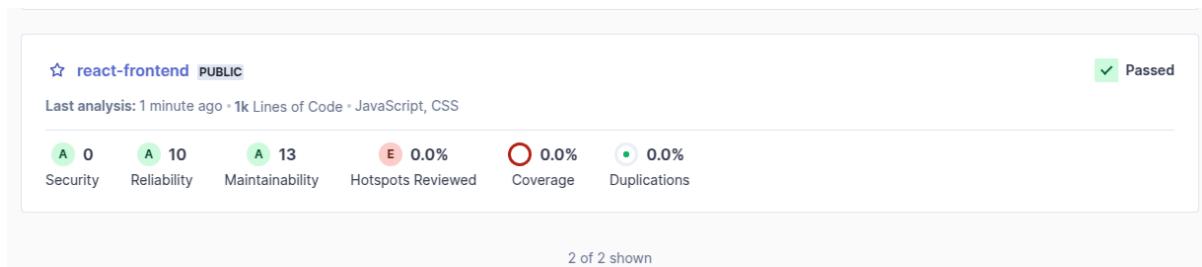
La sécurité constitue un aspect fondamental du développement d'applications web modernes, en particulier lorsque des données sensibles ou des interactions utilisateurs sont impliquées. Une analyse détaillée des résultats SonarQube permet d'évaluer les bonnes pratiques mises en œuvre et d'identifier les axes d'amélioration pour maintenir un niveau de sécurité élevé.

## 1. Analyse des résultats SonarQube

Le score global obtenu est A (0 issues), reflétant une excellente gestion de la sécurité. Aucun problème critique ou majeur n'a été détecté dans le code par SonarQube. Ces résultats mettent en évidence plusieurs points :

- Le respect des pratiques de codage sécurisées.
- L'absence de vulnérabilités connues dans les dépendances externes utilisées dans le projet.

La mise en œuvre de contrôles efficaces, comme la validation des données entrantes et le stockage sécurisé des informations.



## 2. Mesures identifiées pour atteindre ce score

Les résultats obtenus suggèrent que plusieurs bonnes pratiques ont été respectées :

- Utilisation de HTTPS : Les communications entre le client et le serveur sont protégées par un chiffrement TLS.
- Validation des entrées utilisateur : Les données saisies par les utilisateurs sont correctement validées et/ou échappées, ce qui limite les risques d'injection SQL ou XSS.
- Gestion des dépendances : Les bibliothèques utilisées, telles que React et TailwindCSS, sont à jour et exemptes de vulnérabilités connues.

### **3. Recommandations pour renforcer et maintenir la sécurité**

Bien que les résultats soient satisfaisants, certaines actions peuvent être mises en œuvre pour garantir la sécurité à long terme.

#### **3.1. Analyse et gestion des hotspots**

Les hotspots détectés n'ont pas encore été analysés (0 % actuellement). Ces zones du code peuvent contenir des risques potentiels.

Il est recommandé d'identifier ces hotspots via SonarQube et de vérifier les zones sensibles, notamment les appels aux API, le stockage des données et les dépendances tierces.

#### **3.2. Surveillance continue des dépendances**

Les vulnérabilités des bibliothèques tierces peuvent émerger au fil du temps. L'intégration d'outils comme Dependabot ou Snyk permet de surveiller les failles et d'alerter sur les mises à jour critiques.

#### **3.3. Sécurisation des API**

Les appels API doivent être protégés par des limites de taux (rate limiting) pour éviter les abus.

Les données provenant des API externes doivent être systématiquement validées avant d'être exploitées dans l'application.

#### **3.4. Tests de sécurité automatisés**

Des outils tels que OWASP ZAP ou Burp Suite peuvent être intégrés dans le pipeline CI/CD afin de détecter automatiquement les vulnérabilités courantes, telles que les injections ou les attaques XSS.

#### **3.5. Surveillance en production**

Un système de journalisation (logs) permet de capturer et d'analyser les événements suspects en production.

Des outils comme Sentry ou Winston peuvent être utilisés pour surveiller les anomalies.

### **3.6. Conformité RGPD**

Toutes les données sensibles, telles que les mots de passe et tokens, doivent être stockées de manière sécurisée à l'aide de techniques comme le hashage bcrypt ou le chiffrement AES.

Les utilisateurs doivent être informés sur la gestion de leurs données par le biais d'une politique de confidentialité claire.

## **4. Objectifs de suivi et amélioration continue**

Pour garantir une sécurité optimale, les actions suivantes sont recommandées :

Effectuer des audits réguliers du code à l'aide de SonarQube pour identifier et corriger les nouvelles vulnérabilités.

Former les équipes de développement aux bonnes pratiques de sécurité, en s'appuyant notamment sur les principes de l'OWASP Top 10.

Documenter les processus de sécurité afin de faciliter leur mise en œuvre et leur compréhension par tous les membres de l'équipe.

## **Conclusion**

Les résultats obtenus via SonarQube confirment une gestion sécurisée du projet, symbolisée par un score A sans aucune issue critique ou majeure. Cependant, une attention continue reste nécessaire pour anticiper les évolutions technologiques et répondre aux menaces émergentes. En mettant en œuvre ces recommandations, il est possible de maintenir un haut niveau de sécurité tout en renforçant la résilience de l'application face aux risques.

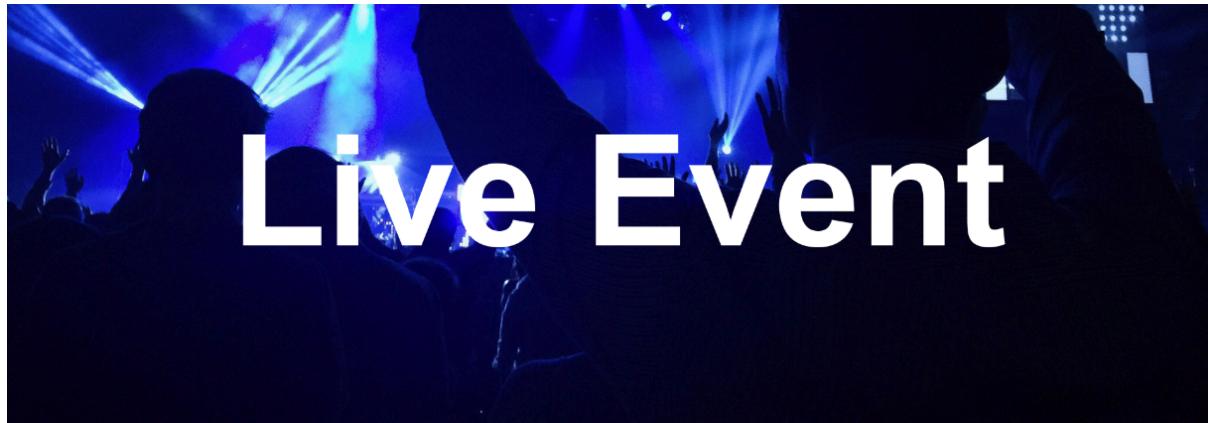
## **Conclusion générale**

La réalisation de ce projet met en lumière les défis et les opportunités liés au développement d'une application web et mobile pour un événement de grande envergure comme "Live Event". Les conclusions de la veille documentaire soulignent l'importance d'un design immersif, d'une navigation intuitive et d'une gestion efficace des contenus pour offrir une expérience optimale aux utilisateurs.

En combinant des technologies modernes telles que ReactJS et TailwindCSS pour le développement front-end, WordPress pour la gestion de contenu et des plateformes comme Vercel ou Netlify pour le déploiement, le projet bénéficie d'une infrastructure technique solide et évolutive. De plus, l'intégration de fonctionnalités clés comme la géolocalisation, les notifications push personnalisées et un programme interactif répond parfaitement aux attentes des festivaliers.

Enfin, en prenant en compte des éléments critiques comme la sécurité, l'accessibilité et la conformité réglementaire, ce projet s'aligne avec les meilleures pratiques de l'industrie, garantissant une solution robuste et durable. Cette démarche proactive et bien documentée permet non seulement de répondre aux besoins actuels, mais aussi d'anticiper les évolutions futures, consolidant ainsi la réussite et l'impact de l'application développée pour "Live Event".

# **Annexes**



Live Event

Live Event

Aa

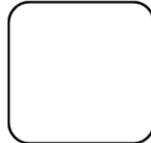
Roboto

Aa

Arizl sans serif



#5D5D5D



#FFFFFF



#010101



#312E81



**Live Event**

Festival Programme Billetterie

## WELCOME TO THE Live Event Festival !

Bienvenue sur notre site web. Découvrez les événements à venir et réservez vos billets dès aujourd'hui.

[Get Started](#)

## Billetterie

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quos ea neque sunt, accusantium soluta minus veniam tempora deserunt? Molestiae eius quidem quam repellat.

[Get Started Today](#)

## Nos partenaires



Partenaire 1

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quos ea neque sunt, accusantium soluta minus veniam tempora deserunt? Molestiae eius quidem quam repellat.



Partenaire 1

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quos ea neque sunt, accusantium soluta minus veniam tempora deserunt? Molestiae eius quidem quam repellat.



Partenaire 1

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quos ea neque sunt, accusantium soluta minus veniam tempora deserunt? Molestiae eius quidem quam repellat.

## Programme

groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## Carte des Lieux

■ Toilettes ■ Bouvettes ■ Concert



[Live Event](#)

**RESSOURCES**

- [Festival](#)
- [Programme](#)

**SUVEZ NOUS**

- [GitHub](#)
- [Discord](#)

**LEGAL**

- [Politique de confidentialité](#)
- [Mentions Légales](#)
- [Contactez-nous](#)

Développé par le studio web XYZ

## Le Festival

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quos ea neque sunt, accusantium soluta minus veniam tempora deserunt? Mole stiae eius quidem quam repella.

Get Started

## Carte des Lieux

Toilettes  Buvettes  Concert



# Programme

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## Live Event

### RESSOURCES

[Festival](#)

[Programme](#)

### SUIVEZ-NOUS

[Github](#)

[Discord](#)

### LEGAL

[Politique de confidentialité](#)

[Mentions Légales](#)

[Contactez-nous](#)

© 2025 Live Event™. Tous droits réservés.



**Groupe 1**

Description : Lorem ipsum dolor sit amet consectetur adipisciing elit. Aut qui hic atque tenetur quis eius quo ea neque sunt, accusantium

Coût: 200\$  
Lieu: Concert B  
Organisateur: Live Event

Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00

[Commander son billet](#)

[Get Started Today](#)

**Live Event****RESSOURCES**[Festival](#)[Programme](#)**SUIVEZ-NOUS**[Github](#)[Discord](#)**LEGAL**[Politique de confidentialité](#)[Mentions Légales](#)  
[Contactez-nous](#)

© 2025 Live Event™. Tous droits réservés.



[Live Event](#)

Festival Programme Billetterie

# WELCOME TO THE Live Event Festival !

Bienvenue sur notre site web. Découvrez les événements à venir et réservez vos billets dès aujourd'hui.

[Get Started](#)

## Billetterie

Lore ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quo es neque sunt, accusantium soluta veniam tempora deserunt? Molestiae eius quidem quam repellat.

[Get Started Today](#)

## Nos partenaires

**MUS**  
Partener 1  
Lore ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quo es neque sunt, accusantium soluta

**MUS**  
Partener 1  
Lore ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quo es neque sunt, accusantium soluta

**MUS**  
Partener 1  
Lore ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quo es neque sunt, accusantium soluta

## Programme

**groupe 1**  
Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00  
[Voir plus sur ce groupe](#)

**groupe 1**  
Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00  
[Voir plus sur ce groupe](#)

**groupe 1**  
Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00  
[Voir plus sur ce groupe](#)

**groupe 1**  
Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00  
[Voir plus sur ce groupe](#)

**groupe 1**  
Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00  
[Voir plus sur ce groupe](#)

**groupe 1**  
Date de début : 12/02/2024 08:00:00  
Date de fin : 21/02/2024 17:00:00  
[Voir plus sur ce groupe](#)

## Carte des Lieux

Toilettes Duvettes Concert

**Live Event**

RESSOURCES  
Festival  
Programme

SUIVEZ NOUS  
GitHub  
Discord

LEGAL  
Politique de confidentialité  
Mentions légales  
Contactez-nous

© 2024 Live Event. Tous droits réservés.

30

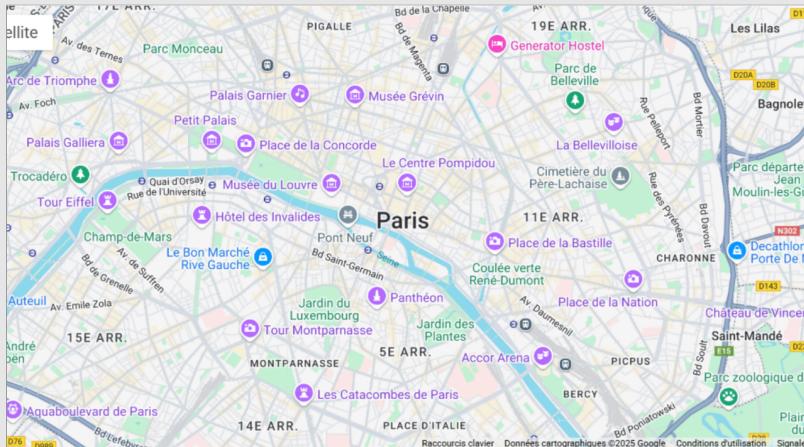
## Le Festival

Lore ipsum dolor, sit amet consectetur adipisicing elit. Aut qui hic atque tenetur quis eius quos ea neque sunt, accusantium soluta minus veniam tempora deserunt? Mole stiae eius quidem quam repella.

[Get Started](#)


## Carte des Lieux

Toilettes Buvettes Concert



## Live Event

### RESSOURCES

Festival  
Programme

### SUIVEZ-NOUS

Github  
Discord

### LEGAL

Politique de confidentialité  
Mentions Légales  
Contactez-nous

© 2023 Live Event\*. Tous droits réservés.



# Programme

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## groupe 1

Date de début : 12/02/2024 08:00:00  
Date de fin: 21/02/2024 17:00:00

[Voir plus sur ce groupe](#)

## Live Event

### RESSOURCES

Festival

Programme

### SUIVEZ-NOUS

Github

Discord

### LEGAL

Politique de confidentialité

Mentions Légales

Contactez-nous

© 2025 Live Event™. Tous droits réservés.



# **Sources du projet**

## **1. Documentation principale**

ReactJS : Documentation officielle

TailwindCSS : Guide officiel

Docker : Tutoriels sur Docker (ex. Docker Documentation)

WordPress : Site officiel de WordPress

MySQL et phpMyAdmin : Tutoriels et documentation (ex. MySQL Documentation et phpMyAdmin)

## **2. Sources supplémentaires**

MDN Web Docs : Documentation MDN

State of CSS Report : Rapport officiel

State of JavaScript Report : Rapport officiel

## **3. Documentation liée aux technologies**

Vercel : Vercel Official Website

Netlify : Netlify Official Website

Next.js : Next.js Documentation

React : React Official Documentation

## **4. CMS et gestion de contenu**

WordPress : WordPress Documentation

MySQL : MySQL Official Documentation

phpMyAdmin : phpMyAdmin Official Website

CMS Market Share : W3Techs - CMS Market Share

State of CMS Report : State of CMS Report

## **5. Comparaisons et benchmarks**

Smashing Magazine : Frontend Framework Comparison

Eventbrite : Eventbrite

Hopin : Hopin

Tomorrowland App : Tomorrowland

Live Nation : Live Nation