



# Rapport du projet

## Le magazine en ligne (Tech Horizons)

### Auteurs :

Wissal EL HAMDI

Fouziya CHAKRANE

Mahmoud EL GHARIB

Anas TARYOUS

### Superviseur :

Pr. M'hamed AIT KBIR

Pr. Yasyn EL YUSIFI

# **Contents :**

- 1. Introduction**
  - 1-1. Présentation du projet
  - 1-2. Objectifs principaux
- 2. Outils et technologies utilisés**
  - 2-1. Backend : Laravel (Framework PHP)
  - 2-2. Base de Données : MySQL
  - 2-3. Frontend : HTML5, CSS3 et JavaScript
- 3. Étapes et répartition des tâches**
  - 3-1. Division en deux équipes (Backend et Frontend)
  - 3-2. Responsabilités principales de chaque groupe
- 4. Concept et modélisation**
  - 4-1. Présentation du concept global de l'application
  - 4-2. Modélisation des données (MCD )
- 5. Diagrammes de l'application**
  - 5-1. Diagramme des cas d'utilisation (par type d'utilisateur : invité, abonné, responsable de thème, éditeur)
  - 5-2. Diagrammes de séquence ou d'activité
- 6. Workflow du projet**
  - 6-1. Processus de développement (phases de conception, développement, test, validation)
  - 6-2. Coordination entre le backend et le frontend
- 7. Application et fonctionnalités implémentées**
  - 7-1. Gestion des utilisateurs et des rôles (invité, abonné, responsable de thème, éditeur)
  - 7-2. Consultation des articles liés aux thèmes abonnés
  - 7-3. Gestion des abonnements et de l'historique
  - 7-4. Système de recommandations basé sur l'historique
  - 7-5. Proposition d'articles et suivi des statuts
- 8. Explication des Pages de l'Application**
  - 8-1. Page de sign up
  - 8-2. Page de log in
  - 8-3. Page "admin dashboard"
  - 8-4. Page "responsible dashboard"

- 8-5. Home page
- 8-6. pages des themes et pages des numbers
- 8-7. les autres pages [contact us – a propos]

## **9. Captures d'écran de l'application**

- 9-1. Interfaces principales (accueil, espace abonné, gestion des thèmes, etc.)
- 9-2. Aperçu des fonctionnalités implémentées

## **1. Introduction**

### **1-1. Présentation du projet**

Dans un monde où l'innovation technologique évolue rapidement, **Tech Horizons** est une application web développée avec **PHP/MySQL** et le framework **Laravel**, visant à gérer un magazine en ligne dédié aux nouvelles technologies. L'application permet aux utilisateurs d'accéder à des articles selon leurs abonnements, de gérer leur historique de navigation et de proposer du contenu à publier.

Cette plateforme intuitive et sécurisée offre une expérience utilisateur optimisée grâce à des recommandations d'articles basées sur les préférences et l'historique de navigation.

### **1-2. Objectifs principaux**

L'objectif principal de Tech Horizons est de fournir une solution complète pour la gestion d'un magazine en ligne, en mettant l'accent sur :

- L'accessibilité aux contenus technologiques : Permettre aux utilisateurs de consulter des articles sur des thématiques variées (IA, cybersécurité, IoT, etc.).
- Une gestion efficace des abonnements : Offrir aux abonnés un espace personnalisé pour suivre leurs thèmes favoris.
- L'interaction et l'engagement des utilisateurs : Permettre aux abonnés de proposer des articles, de noter les publications et d'interagir via un système de discussion.
- Un contrôle éditorial structuré : Fournir aux éditeurs et responsables de thèmes des outils de gestion des publications et des statistiques.
- Une application performante et sécurisée : Développer une interface fluide et responsive en utilisant Laravel et un code CSS/JavaScript personnalisé, sans frameworks externes.

## **2. Outil et technologies utilisés**

### **2-1. Backend : Laravel (Framework PHP)**

Laravel est un framework PHP open-source basé sur l'architecture MVC (Modèle-Vue-Contrôleur). Son choix repose sur plusieurs critères :

- Séparation des responsabilités : Laravel adopte une approche modulaire, facilitant la gestion du code et sa maintenance.
- ORM Eloquent : Un puissant ORM permettant une interaction fluide avec la base de données MySQL, simplifiant les requêtes et garantissant une meilleure sécurité grâce aux requêtes préparées.
- Gestion des routes et middleware : Permet un contrôle précis des accès utilisateurs et une gestion efficace des requêtes HTTP.
- Système de migration et seeders : Outils facilitant la gestion et l'évolution de la base de données au fil du développement.
- Sécurité intégrée : Laravel propose des mécanismes natifs de protection contre les attaques courantes (CSRF, XSS, SQL Injection).

## 2-2. Base de Données : MySQL

La base de données MySQL a été choisie pour sa stabilité, sa scalabilité et sa compatibilité avec Laravel. Elle est utilisée pour stocker l'ensemble des informations de l'application :

- Utilisateurs et rôles : Stockage des abonnés, responsables de thèmes, éditeurs et invités.
- Thèmes et articles : Organisation et structuration du contenu en fonction des catégories technologiques proposées.
- Historique de navigation et recommandations : Suivi des consultations et génération de recommandations personnalisées.
- Gestion des numéros du magazine : Association des articles avec leurs publications respectives.

L'optimisation des requêtes MySQL via l'indexation et l'utilisation de relations clés étrangères permet d'améliorer la performance globale de l'application.

## 2-3. Frontend : HTML5, CSS3 et JavaScript

Contrairement à l'utilisation de frameworks comme Bootstrap ou jQuery, le choix a été fait de développer l'interface avec des technologies **natales et personnalisées**, ce qui permet :

- **Un contrôle total sur le design et l'expérience utilisateur.**
- **Un code allégé et optimisé** pour de meilleures performances.
- **Une meilleure compréhension et maintenabilité** du code source.

## **HTML5**

HTML5 est utilisé pour structurer le contenu de l'application en respectant les **normes sémantiques** modernes. Cela améliore le référencement naturel (SEO) et facilite l'accessibilité du site.

## **CSS3**

Le design de l'application repose sur **CSS3 pur**, sans framework CSS, ce qui permet de créer une **interface sur mesure** tout en assurant une compatibilité avec les navigateurs modernes.

- **Utilisation des Flexbox et Grid Layout** pour une disposition fluide et adaptable.
- **Animations et transitions CSS** pour améliorer l'interactivité.
- **Mise en place de media queries** pour assurer un **responsive design**, optimisé pour les écrans desktop, tablettes et mobiles.

## **JavaScript**

L'interactivité et la dynamique de l'application sont assurées par **JavaScript natif (Vanilla JS)**, sans dépendances externes. Ce choix permet de :

- **Optimiser la rapidité d'exécution des scripts** en évitant les surcharges de frameworks.
- **Gérer les interactions utilisateur** (clics, animations, transitions, validation des formulaires).
- **Implémenter des fonctionnalités avancées** comme le chargement asynchrone (AJAX) pour récupérer des données en arrière-plan sans recharger la page.

## **3. Etapes et répartition des tâches**

Pour assurer une gestion efficace du projet Tech Horizons, le travail a été réparti entre deux équipes distinctes : Backend et Frontend. Cette division permet une spécialisation des tâches et une meilleure coordination dans le développement de l'application.

### **3-1. Division en Deux Équipes (Backend et Frontend)**

Le projet a été structuré de manière à séparer les responsabilités techniques en deux pôles :

- L'équipe Backend : Responsable de la logique métier, de la gestion des bases de données et de l'API de communication avec le frontend.
- L'équipe Frontend : Chargée de la conception de l'interface utilisateur et de l'intégration des fonctionnalités interactives.

Cette séparation permet de travailler en parallèle et d'optimiser le développement tout en garantissant une architecture modulaire et évolutive.

### **3-2. Responsabilités Principales de Chaque Groupe**

#### **Équipe Backend**

Technologies utilisées : Laravel (PHP), MySQL

Les principales responsabilités de l'équipe Backend incluent :

- Création et gestion de la base de données avec MySQL.
- Développement des modèles et contrôleurs en utilisant Laravel.
- Sécurisation de l'application (authentification, gestion des rôles et permissions).
- Optimisation des requêtes et gestion des performances.
- Gestion des différentes entités : utilisateurs, abonnements, articles, historiques de navigation, etc.
- Mise en place des fonctionnalités avancées comme les recommandations d'articles.

#### **Équipe Frontend**

Technologies utilisées : HTML5, CSS3, JavaScript

Les principales missions de l'équipe Frontend sont :

- Conception et développement de l'interface utilisateur selon les maquettes définies.
- Développement de styles CSS personnalisés pour une identité visuelle unique.
- Intégration des fonctionnalités dynamiques en JavaScript natif.
- Mise en place d'un design responsive pour une compatibilité avec tous les appareils (desktop, mobile, tablette).
- Gestion des interactions utilisateur (animations, transitions, navigation intuitive).
- Implémentation des requêtes AJAX pour interagir avec le backend sans recharger la page.
- Tests et optimisation des performances frontend.

## 4. Concept et modélisation

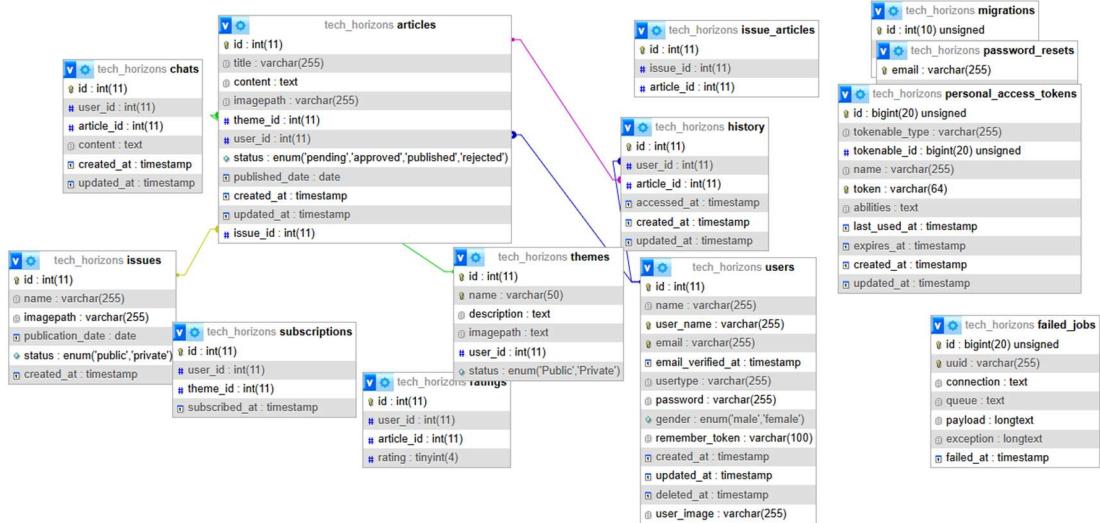
### 4-1. Présentation du concept global de l'application

L'application "Tech Horizons VR" est une plateforme immersive de gestion et de publication d'articles en réalité virtuelle. Elle permet aux utilisateurs de créer, publier et consulter du contenu interactif en lien avec divers thèmes. L'objectif est d'offrir une expérience enrichissante et dynamique.

Les utilisateurs peuvent soumettre des articles qui sont d'abord modérés avant publication. Chaque article est associé à un thème et à un utilisateur. Une gestion des rôles permet de différencier les auteurs, modérateurs et administrateurs.

### 4-2. Modélisation des données (MCD)

La base de données repose sur un modèle relationnel structuré afin d'assurer une gestion efficace des articles, des utilisateurs et des thèmes.



## 5. Diagrammes de l'application

### 5-1. Diagramme des cas d'utilisation

Le **diagramme des cas d'utilisation** permet de décrire les différentes interactions entre les utilisateurs de l'application Tech Horizons et le système. Il identifie les fonctionnalités offertes par l'application, en les reliant aux rôles des utilisateurs qui y accèdent. Ce diagramme est essentiel pour mieux comprendre les actions possibles pour chaque type d'utilisateur et la manière dont ils interagissent avec l'application.

Pour chaque rôle, les cas d'utilisation sont spécifiés comme suit :

- **Invité** : Un utilisateur invité peut uniquement consulter les informations publiques disponibles sur les thèmes technologiques et déposer une demande d'inscription au magazine. Il peut également visualiser les numéros publics des magazines sans pouvoir interagir davantage avec le contenu.
- **Abonné** : Un abonné dispose d'un espace personnalisé où il peut consulter l'ensemble des numéros du magazine, gérer ses abonnements aux différents thèmes (tels que l'intelligence artificielle, la cybersécurité, etc.), consulter l'historique de ses navigations, et proposer des articles à publier. Il peut également participer aux discussions en ligne, attribuer des notes aux articles, et suivre l'état de ses propositions (Refus, En cours, Retenu, Publié).
- **Responsable de thème** : Ce rôle a un accès étendu aux fonctionnalités liées à la gestion des abonnements pour son thème spécifique. Le responsable peut gérer les abonnements, consulter les articles proposés par les abonnés, les soumettre pour publication, modérer les conversations relatives aux articles et consulter les statistiques des abonnés et des articles de son thème.
- **Éditeur** : L'éditeur a des priviléges de gestion avancée. Il peut gérer les numéros du magazine, publier ou rendre public un numéro, et gérer les utilisateurs (abonnés, responsables de thème). L'éditeur peut également modifier ou supprimer des articles et des utilisateurs, ainsi qu'activer ou désactiver certains contenus.

Chaque rôle est relié à des cas d'utilisation spécifiques dans le diagramme, permettant ainsi de visualiser clairement les interactions entre les utilisateurs et les fonctionnalités du système.

## 5-2. Diagramme de séquence ou d'activité

Les diagrammes de séquence et d'activité sont utilisés pour décrire le flux d'interactions dans le système, notamment pour chaque processus clé ou fonction importante de l'application.

- **Diagramme de séquence** : Ce diagramme montre comment les objets du système interagissent au fil du temps, en mettant en évidence les messages échangés entre eux. Par exemple, pour un abonné qui propose un article, le diagramme de séquence pourrait illustrer les étapes suivantes :
  1. L'abonné soumet un article à travers l'interface utilisateur.
  2. Le système vérifie les données de l'utilisateur et l'enregistre dans la base de données.

- 3. Un message de confirmation est envoyé à l'abonné pour lui indiquer que l'article a été soumis.
- 4. Si l'article est accepté, un responsable de thème reçoit une notification pour examen. Ce flux décrit les interactions entre l'utilisateur, l'application, et la base de données dans le cadre d'une proposition d'article.
  
  
- **Diagramme d'activité** : Ce diagramme permet de décrire le flux de travail d'une activité spécifique dans le système, en mettant en évidence les différentes étapes de cette activité. Par exemple, pour le processus de gestion des abonnements, le diagramme d'activité pourrait détailler :
  1. Un abonné se connecte à son espace.
  2. L'abonné consulte ses thèmes d'abonnement et choisit de s'abonner à un nouveau thème.
  3. Le système met à jour la base de données pour ajouter l'abonnement.
  4. L'abonné peut voir ce changement dans son interface personnalisée.

Ces diagrammes sont cruciaux pour comprendre les détails des processus métiers dans l'application et permettent de mieux définir les flux d'interactions entre les utilisateurs et le système. Ils servent également de base pour la programmation des différentes fonctionnalités.

## 5. Workflow du projet

### 6-1. Processus de développement

Le **processus de développement** du projet Tech Horizons suit une approche structurée, en plusieurs phases, permettant d'assurer une gestion efficace et une livraison de qualité. Ce processus est divisé en quatre grandes étapes : conception, développement, test, et validation.

1. **Phase de conception** : La première étape est celle de la conception, qui commence par l'analyse des besoins du projet et la rédaction du cahier des charges. Durant cette phase, l'architecture de l'application est définie, ainsi que les rôles des utilisateurs, les fonctionnalités attendues, et la structure de la base de données. Cette phase implique également la création des diagrammes nécessaires (diagrammes de cas d'utilisation, de séquence, d'activité, etc.) pour une meilleure compréhension des interactions du système. Des wireframes peuvent aussi être créés pour visualiser l'interface utilisateur.

Une fois la conception validée, l'équipe détermine les technologies à utiliser, en l'occurrence Laravel pour le backend et du CSS/JavaScript personnalisé pour le frontend. Les différentes contraintes techniques sont également prises en compte, telles que la gestion des utilisateurs, la sécurité de l'application, et l'intégration de la base de données.

2. **Phase de développement :** La phase de développement se divise en deux parties principales : le backend et le frontend. Le backend est principalement développé avec Laravel, qui permet de gérer les utilisateurs, la logique métier, et l'interaction avec la base de données MySQL. Cela inclut la mise en place des modèles, des contrôleurs, des migrations de la base de données, et des vues.

En parallèle, le développement du frontend se concentre sur la création des interfaces utilisateurs à l'aide de HTML, CSS et JavaScript, en suivant les spécifications du design. L'interface est conçue pour être intuitive et responsive, tout en étant adaptée aux différents types d'utilisateurs.

Une attention particulière est accordée à la gestion des rôles (invité, abonné, responsable de thème, éditeur) dans le backend, avec l'utilisation de middleware dans Laravel pour sécuriser l'accès aux différentes ressources en fonction des permissions de l'utilisateur.

3. **Phase de test :** Une fois les fonctionnalités développées, des tests sont réalisés pour vérifier le bon fonctionnement de l'application. Cela inclut des tests unitaires pour le backend, permettant de valider les logiques métier (par exemple, la gestion des abonnements ou la proposition d'articles). Des tests fonctionnels sont aussi réalisés pour valider les flux utilisateur, tels que la consultation des articles, l'inscription ou la gestion de l'historique de navigation.

En parallèle, des tests d'intégration sont menés pour s'assurer que le backend et le frontend fonctionnent correctement ensemble, notamment lors des interactions avec la base de données (par exemple, l'ajout d'un article ou la gestion des abonnements). Des tests de sécurité sont également effectués pour vérifier la robustesse de l'application contre les attaques courantes (comme l'injection SQL ou la falsification de requêtes intersites).

4. **Phase de validation :** La dernière étape consiste à valider le projet en s'assurant que toutes les exigences du cahier des charges ont été respectées. Un audit complet de l'application est réalisé pour valider la conformité des fonctionnalités et de la sécurité. Les utilisateurs finaux (ou une équipe de test) peuvent également être impliqués dans cette phase pour donner un retour sur

l'ergonomie et la performance de l'application. Si tout est validé, l'application est prête pour la mise en production.

## 6-2. Coordination entre le backend et le frontend

La **coordination entre le backend et le frontend** est essentielle pour assurer la fluidité de l'application et garantir que l'interface utilisateur fonctionne correctement avec les données stockées dans la base de données. Voici les principales étapes de coordination:

1. **Web.php et gestion des données** : Le backend, développé avec Laravel, est responsable de la gestion de la logique métier, de l'accès à la base de données et de la manipulation des données. Le frontend, quant à lui, est chargé d'afficher ces données et d'interagir avec l'utilisateur. La communication entre ces deux parties se fait principalement via des web.php.

Par exemple, lorsque l'utilisateur se connecte à son espace abonné, une requête API est envoyée au backend pour récupérer les données associées à l'utilisateur (comme ses abonnements, son historique de navigation, etc.). Le backend traite la requête, interroge la base de données et renvoie les résultats au frontend sous forme de JSON, qui sont ensuite affichés à l'utilisateur dans l'interface.

2. **Validation des interactions** : La coordination entre le frontend et le backend s'effectue également lors des actions utilisateur. Par exemple, lorsqu'un abonné propose un article, le frontend collecte les données (comme le titre, le contenu, etc.) et les envoie au backend via une API. Le backend valide ces données, les enregistre dans la base de données, et renvoie une confirmation au frontend, qui affiche un message à l'utilisateur (par exemple, "Article soumis avec succès").
3. **Gestion des erreurs et retours utilisateur** : Il est également crucial que le backend renvoie des messages d'erreur appropriés en cas de problème (par exemple, si les données envoyées par le frontend sont invalides). Ces erreurs sont renvoyées sous forme de messages d'erreur dans la réponse JSON, que le frontend peut traiter et afficher de manière conviviale pour l'utilisateur.
4. **Synchronisation des mises à jour** : Lorsque des modifications sont apportées aux données (comme l'ajout d'un article ou la mise à jour de l'historique de navigation), le frontend doit être mis à jour en temps réel pour refléter ces changements. Cela peut être réalisé par le biais de requêtes AJAX, permettant une mise à jour dynamique sans nécessiter de recharge complet de la page.

En résumé, une communication fluide entre le backend et le frontend est essentielle pour assurer que l'application fonctionne de manière cohérente et performante. Les développeurs doivent veiller à maintenir une séparation claire des responsabilités, tout

en veillant à ce que les données circulent efficacement entre les différentes parties du système.

## 6. Application et fonctionnalités implémentées

### 7-1. Gestion des utilisateurs et des rôles

L'application **Tech Horizons** gère les utilisateurs et leurs rôles via Laravel pour une navigation sécurisée.

- **Invité** : Accède aux informations publiques et peut se connecter via un formulaire sécurisé.
- **Abonné** : Dispose d'un espace personnel pour gérer ses abonnements et proposer des articles.
- **Responsable de thème** : Modère les articles et gère les abonnements liés à son thème.
- **Éditeur** : Administre les utilisateurs, articles et numéros avec des permissions avancées.

Les accès sont contrôlés via des rôles définis en base de données et sécurisés par des middlewares.

### 7-2. Consultation des articles liés aux thèmes abonnés

Une des fonctionnalités majeures de l'application est la possibilité pour un abonné de consulter des articles en fonction des thèmes auxquels il est abonné. Lorsqu'un abonné se connecte à son espace personnalisé, il peut accéder à une liste d'articles qui correspondent aux thèmes qu'il suit, tels que l'intelligence artificielle, la cybersécurité ou la réalité augmentée. Cette fonctionnalité est rendue possible grâce à une relation entre les utilisateurs et les thèmes dans la base de données, qui permet d'afficher uniquement les articles pertinents pour chaque abonné.

Les articles sont présentés de manière interactive, avec des images correspondant à chaque numéro du magazine. En cliquant sur une image, l'utilisateur est redirigé vers l'article correspondant. Cette approche permet une navigation fluide et intuitive dans le contenu du magazine, tout en garantissant que chaque abonné bénéficie d'une expérience personnalisée en fonction de ses centres d'intérêt.

### **7-3. Gestion des abonnements et de l'historique**

La gestion des abonnements et de l'historique est une autre fonctionnalité clé de l'application. Un abonné peut facilement gérer ses abonnements aux différents thèmes du magazine, en ajoutant ou supprimant des abonnements à partir de son tableau de bord personnalisé. Ces informations sont stockées dans la base de données et peuvent être modifiées à tout moment, offrant ainsi une grande flexibilité aux utilisateurs.

L'historique de navigation permet également à l'abonné de suivre les articles qu'il a déjà consultés. Ce système est particulièrement utile pour retrouver des articles précédemment lus, en utilisant des filtres pour affiner la recherche par thème, titre, ou date de consultation. Cette fonctionnalité améliore l'expérience utilisateur en permettant un accès rapide à des contenus déjà explorés.

### **7-4. Système de recommandations basé sur l'historique**

Pour enrichir l'expérience de l'abonné, un système de recommandations a été implémenté, basé sur l'historique de navigation et les abonnements de l'utilisateur. Lorsqu'un abonné consulte plusieurs articles dans un thème spécifique, le système analyse son comportement et lui propose des articles similaires ou pertinents, en fonction de ses intérêts. Cette fonctionnalité utilise un algorithme simple, qui peut être amélioré par la suite avec des techniques plus avancées de filtrage collaboratif ou de machine learning pour rendre les recommandations plus personnalisées.

Ainsi, lorsque l'utilisateur se connecte à son espace, la page d'accueil affiche une sélection d'articles recommandés en fonction de ses précédentes interactions. Ce système offre une expérience plus dynamique, en incitant les utilisateurs à découvrir de nouveaux articles qui correspondent à leurs intérêts.

### **7-5. Proposition d'articles et suivi des statuts**

Les abonnés peuvent proposer des articles à publier dans le magazine. Cette fonctionnalité permet aux utilisateurs de participer activement à la création du contenu. Lorsqu'un abonné soumet un article, celui-ci est stocké dans la base de données avec un statut initial de "En cours", indiquant qu'il est en attente d'examen par un responsable de thème ou un éditeur. Le suivi de l'état de chaque article proposé est essentiel, car il permet à l'abonné de savoir si son article a été refusé, accepté, ou publié.

Une fois qu'un responsable de thème évalue l'article, le statut est mis à jour (par exemple, "Retenu" ou "Refusé"). Cette gestion des statuts assure une traçabilité des propositions et permet de suivre l'évolution de chaque article. De plus, les abonnés peuvent être notifiés du changement de statut de leurs articles, garantissant une communication transparente entre les auteurs et les responsables.

## 7. Explication des pages de l'application

### 8-1. Page de sign up

La page de Sign Up dans l'application Laravel permet aux nouveaux utilisateurs de s'inscrire en fournissant des informations personnelles essentielles. Voici la logique de création et de fonctionnement de cette page :

#### a. Interface utilisateur (Frontend) :

La page contient un formulaire d'inscription avec plusieurs champs :

- Nom d'utilisateur : Identifiant unique pour l'utilisateur.
- Email : Adresse électronique pour la connexion et la récupération de compte.
- Mot de passe : Champ sécurisé pour protéger l'accès au compte.
- Confirmation du mot de passe : Assure que l'utilisateur a bien saisi le bon mot de passe.
- sexe

#### b. Traitement des données (Backend) :

Lorsqu'un utilisateur soumet le formulaire, les données sont envoyées à une route Laravel dédiée à l'inscription, définie dans routes/auth.php. Cette route est associée à une méthode du RegisteredUserController.

Le contrôleur effectue plusieurs étapes :

1. Validation des données : Vérifie que tous les champs requis sont remplis et que l'email est unique.
2. Hashing du mot de passe : Sécurisation des mots de passe avec bcrypt avant stockage.
3. Création de l'utilisateur : Un nouvel enregistrement est ajouté à la table users dans la base de données.
4. Authentification automatique : Après une inscription réussie, Laravel connecte directement l'utilisateur avec Auth::login().
5. Redirection vers une page de sélection de thèmes pour personnaliser l'expérience utilisateur.

### c. Gestion des erreurs et retours utilisateur :

- Si la validation échoue, Laravel renvoie l'utilisateur sur la page d'inscription avec des messages d'erreur pour corriger les informations incorrectes.
- Si l'inscription est réussie, un message de confirmation s'affiche, et l'utilisateur est redirigé vers la prochaine étape de l'expérience.

Cette logique permet d'assurer une inscription sécurisée et fluide pour l'utilisateur tout en respectant les bonnes pratiques de Laravel.

## 8-2. Page de log in

La page de connexion (Login) de l'application Laravel permet aux utilisateurs déjà enregistrés de se connecter à leur compte en toute sécurité.

### a. Interface utilisateur (Frontend) :

Le formulaire de connexion comprend deux champs essentiels : l'email et le mot de passe. Lorsqu'un utilisateur soumet ces informations, elles sont envoyées à une route dédiée dans le fichier routes/auth.php, associée à la méthode login du AuthController.

### b. Traitement des données (Backend) :

Le contrôleur commence par valider les informations d'identification fournies, en vérifiant l'existence de l'utilisateur dans la base de données et en comparant le mot de passe saisi avec celui stocké dans la base de données en utilisant la méthode Hash::check. Si les identifiants sont corrects, Laravel authentifie automatiquement l'utilisateur avec Auth::attempt() et le redirige vers la page souhaitée, généralement le tableau de bord ou la dernière page consultée. En cas d'erreur, comme des identifiants incorrects ou un utilisateur non trouvé, l'utilisateur est renvoyé à la page de connexion avec des messages d'erreur explicites pour corriger la situation.

Ce processus garantit une connexion fluide, sécurisée et bien gérée tout en offrant une expérience utilisateur intuitive.

## 8-3. Page “admin dashboard”

# Rapport du Projet Laravel : Admin Dashboard

## 1

### Structure du Tableau de Bord

Le menu de navigation comprend les sections suivantes :

- **Dashboard** (Vue d'ensemble)
- **Manage Articles** (Gestion des articles)
- **Create Article** (Ajout d'un article)
- **Manage Users** (Gestion des utilisateurs)
- **Manage Themes** (Gestion des thèmes)
- **Manage Numbers** (Gestion des numéros)
- **Settings** (Paramètres de l'admin)

#### 2.1. Dashboard (Vue d'ensemble)

**Front-end :**

- Page affichant un récapitulatif des articles publiés, du nombre d'utilisateurs, et des autres ressources.
- Utilisation d'HTML pour la structure et CSS pour le style.
- JavaScript pour les interactions dynamiques (ex. graphiques de statistiques).

**Back-end :**

- Route : `Route::get('admin/adminhome', [AdminController::class, 'index'])->name('admin.adminhome');`
- Contrôleur : `AdminController@index()` récupère les données et les envoie à la vue.
- Modèles utilisés : `Article, User, Theme, Issue` pour compter les éléments et générer des statistiques.

#### 2.2. Manage Articles (Gestion des articles)

**Front-end :**

- Affichage d'une liste des articles sous forme de tableau avec options d'édition/suppression.
- Boutons d'action en JavaScript (confirmations de suppression, pagination dynamique, etc.).

**Back-end :**

- Routes :

- Route::get('admin/articles', [AdminArticleController::class, 'index'])->name('admin.articles.index');
  - Route::get('/admin/articles/create', [AdminCreateArticleController::class, 'create'])->name('admin.articles.create');
  - Route::post('/admin/articles/store', [AdminCreateArticleController::class, 'store'])->name('admin.articles.store');
  - Route::post('/admin/articles/update', [AdminArticleController::class, 'updateStatus'])->name('admin.articles.updateStatus');
  - Route::post('/admin/articles/{id}', [AdminArticleController::class, 'destroy'])->name('admin.articles.delete');
- Contrôleurs : AdminArticleController et AdminCreateArticleController gèrent l'affichage, la création et la suppression des articles.
- Modèle : Article avec récupération des données via Article::all();

## 2.3. Manage Users (Gestion des utilisateurs)

### Front-end :

- Tableau listant les utilisateurs avec options de modification et suppression.
- JavaScript pour gérer la confirmation de suppression.

### Back-end :

- Routes :
  - Route::get('/users', [AdminController::class, 'index'])->name('admin.users.index');
  - Route::get('/users/{id}/edit', [AdminController::class, 'edit'])->name('admin.users.edit');
  - Route::put('/users/{id}', [AdminController::class, 'update'])->name('admin.users.update');
  - Route::delete('/users/{id}', [AdminController::class, 'destroy'])->name('admin.users.destroy');
- AdminUserController gère l'affichage et la modification des utilisateurs.
- Modèle : User

## 2.4. Manage Themes (Gestion des thèmes)

### Front-end :

- Interface similaire à la gestion des articles, avec affichage des thèmes disponibles.

### Back-end :

- Routes :
  - Route::get('themes', [AdminThemeController::class, 'index'])->name('admin.themes.index');
  - Route::post('themes', [AdminThemeController::class, 'store'])->name('admin.themes.store');
  - Route::put('themes/{id}', [AdminThemeController::class, 'update'])->name('admin.themes.update');

- o Route::delete('themes/{id}', [AdminController::class, 'destroy'])->name('admin.themes.destroy');
- Modèle : Theme

## 2.5. Manage Numbers (Gestion des numéros)

**Front-end :**

- Page affichant une liste des numéros publiés.

**Back-end :**

- Routes :
  - o Route::get('issues', [AdminController::class, 'index'])->name('admin.issues.index');
  - o Route::post('issues', [AdminController::class, 'store'])->name('admin.issues.store');
  - o Route::patch('issues/{id}/status', [AdminController::class, 'updateStatus'])->name('admin.issues.updateStatus');
  - o Route::delete('issues/{id}', [AdminController::class, 'destroy'])->name('admin.issues.destroy');
- Modèle : Issue

## 2.6. Settings (Paramètres d'administration)

**Front-end :**

- Formulaire permettant la modification des informations administratives (nom, mot de passe, etc.).

**Back-end :**

- Routes :
  - o Route::get('/settings', [AdminController::class, 'index'])->name('admin.settings');
  - o Route::post('/settings/update', [AdminController::class, 'updateSettings'])->name('admin.updateSettings');
- AdminSettingsController gère l'affichage et la mise à jour des paramètres administratifs.

## 8-4. Page “responsible dashboard“

**Structure du Tableau de Bord**

Le tableau de bord est constitué d'un menu latéral (`sidebar`) avec plusieurs sections, chacune correspondant à une fonctionnalité spécifique de gestion. Voici les principales sections du tableau de bord :

1. **Dashboard** : La page d'accueil du tableau de bord, offrant un aperçu des statistiques et des informations générales.
2. **Articles** : Permet de gérer les articles publiés, de les consulter et de les supprimer.
3. **Abonnés** : Section dédiée à la gestion des abonnés, avec la possibilité de voir leurs informations et de les supprimer si nécessaire.
4. **Propositions de Modification** : Permet de consulter et de gérer les propositions soumises par les abonnés pour modifier du contenu.
5. **Conversations** : Cette section permet de gérer les conversations entre les abonnés et les modérateurs.
6. **Paramètres du Profil** : Permet au responsable de mettre à jour ses informations personnelles telles que le nom d'utilisateur et le mot de passe.

## HTML du Menu Latéral

```
html
CopyEdit
<ul class="sidebar-menu">
    <li><a href="{{ route('moderatorhome') }}" data-section="dashboard" class="menu-link active">Dashboard</a></li>
    <li><a href="{{ route('moderator.articles.index') }}" data-section="articles" class="menu-link">Articles</a></li>
    <li><a href="{{ route('moderator.subscribers.index') }}" data-section="subscribers" class="menu-link">Subscribers</a></li>
    <li><a href="{{ route('moderator.proposals.index') }}" data-section="subscriber-proposals" class="menu-link">Subscriber Proposals</a></li>
    <li><a href="{{ route('moderator.conversations.index') }}" data-section="conversations" class="menu-link">Conversations</a></li>
    <li><a href="{{ route('moderator.settings') }}" data-section="profile-settings" class="menu-link">Profile Settings</a></li>
</ul>
```

## Routes Laravel

Les routes définies dans Laravel permettent d'associer chaque section du tableau de bord à une méthode dans un contrôleur correspondant. Ces routes sont protégées par un middleware `auth` et `moderator`, ce qui garantit que seul un utilisateur authentifié et ayant le rôle de modérateur peut accéder à ces fonctionnalités.

Voici la liste des routes associées au tableau de bord responsable :

```
php
CopyEdit
// Routes pour le modérateur
Route::middleware(['auth', 'moderator'])->group(function () {
    // Dashboard
    Route::get('/moderator/dashboard',
    [ModeratorDashboardController::class, 'dashboard'])->name('moderatorhome');

    // Gestion des articles
    Route::resource('articles', ArticleController::class);
    Route::resource('subscribers', SubscriberController::class);
    Route::resource('proposals', ProposalController::class);
    Route::resource('conversations', ConversationController::class);
    Route::resource('settings', ModeratorSettingController::class);
});
```

```

    Route::get('/moderator/articles', [ModeratorArticlesController::class,
'index'])->name('moderator.articles.index');
    Route::get('/moderator/articles/{id}', [ModeratorArticlesController::class, 'show'])-
>name('moderator.articles.show');
    Route::delete('/moderator/articles/{id}', [ModeratorArticlesController::class, 'destroy'])-
>name('moderator.articles.destroy');
    Route::post('/moderator/articles/{id}/publish', [ModeratorArticlesController::class, 'publish'])-
>name('moderator.articles.publish');

    // Gestion des abonnés
    Route::get('/moderator/subscribers',
[ModeratorSubscribersController::class, 'index'])-
>name('moderator.subscribers.index');
    Route::get('/moderator/subscribers/{id}', [ModeratorSubscribersController::class, 'show'])-
>name('moderator.subscribers.show');
    Route::delete('/moderator/subscribers/{id}', [ModeratorSubscribersController::class, 'destroy'])-
>name('moderator.subscribers.destroy');

    // Gestion des conversations
    Route::get('/moderator/conversations', [ModeratorChatController::class,
'index'])->name('moderator.conversations.index');
    Route::delete('/moderator/conversations/{id}', [ModeratorChatController::class, 'destroy'])-
>name('moderator.conversations.destroy');

    // Gestion des propositions
    Route::get('/moderator/proposals', [ModeratorProposalController::class,
'index'])->name('moderator.proposals.index');
    Route::delete('/moderator/proposals/{id}', [ModeratorProposalController::class, 'destroy'])-
>name('moderator.proposals.destroy');
    Route::post('/moderator/proposals/{id}/propose-edit', [ModeratorProposalController::class, 'proposeEdit'])-
>name('moderator.proposals.proposeEdit');

    // Mise à jour du profil
    Route::put('/moderator/profile/update',
[ModeratorSettingsController::class, 'update'])-
>name('moderator.profile.update');
    Route::get('/moderator/settings', [ModeratorSettingsController::class,
'settings'])->name('moderator.settings');
});


```

## Fonctionnalités et Description des Routes

### 1. Dashboard

- **Route :** /moderator/dashboard
- **Méthode :** GET
- Cette route affiche la page d'accueil du tableau de bord, avec un aperçu des statistiques et des informations clés sur la plateforme.

### 2. Articles

- **Route** : /moderator/articles
- **Méthode** : GET
- Permet d'afficher une liste des articles, avec la possibilité de les visualiser et de les supprimer.
- **Route de Publication** : POST /moderator/articles/{id}/publish
- **Route de Suppression** : DELETE /moderator/articles/{id}

### 3. Abonnés

- **Route** : /moderator/subscribers
- **Méthode** : GET
- Affiche une liste des abonnés et permet de consulter leurs détails ou de les supprimer.

### 4. Propositions de Modification

- **Route** : /moderator/proposals
- **Méthode** : GET
- Permet de consulter et de gérer les propositions soumises par les abonnés.

### 5. Conversations

- **Route** : /moderator/conversations
- **Méthode** : GET
- Affiche la liste des conversations en cours entre abonnés et modérateurs.

### 6. Paramètres du Profil

- **Route** : /moderator/settings
- **Méthode** : GET
- Permet de consulter et de mettre à jour les informations du profil du modérateur.

## 8-5. user dashboard

### 1. Structure du Menu de Navigation

Le menu de navigation du dashboard utilisateur est généré par une liste de liens dans un fichier Blade. Chaque lien pointe vers une route spécifique dans Laravel. Les sections incluent :

- **Dashboard**
- **Subscription**
- **My Articles**
- **Browsing History**

- **Propose Article**
- **Settings**

Chaque élément <li> utilise un lien <a> avec une route Laravel (par exemple, {{ route('user.dashboarduser') }}). Le paramètre `data-section` semble indiquer la section active du menu, ce qui peut être utilisé par JavaScript pour gérer l'interface utilisateur, comme la mise en surbrillance de la section active.

## 2. Routes Web (web.php)

Les routes sont définies dans le fichier `web.php` de Laravel. Chaque route est associée à un contrôleur et à une méthode qui gère l'affichage ou la logique de la section du dashboard. Voici les différentes routes et leurs actions :

- **Dashboard** : Route qui charge la vue du tableau de bord via la méthode `index` du `DashboardUserController`.

```
php
CopyEdit
Route::get('/user/dashboarduser', [DashboardUserController::class,
    'index'])->name('user.dashboarduser');
```

- **Subscription** : Route qui charge la page des abonnements et permet de gérer les abonnements via la méthode `index` du `UserSubscriptionController`.

```
php
CopyEdit
Route::get('/user/subscription', [UserSubscriptionController::class,
    'index'])->name('user.subscription');
```

- **My Articles** : Route qui charge la liste des articles de l'utilisateur via la méthode `myArticles` du `UserArticleController`.

```
php
CopyEdit
Route::get('/user/my-articles', [UserArticleController::class,
    'myArticles'])->name('user.myarticles');
```

- **Browsing History** : Route pour afficher l'historique de navigation de l'utilisateur via la méthode `index` du `UserHistoryController`.

```
php
CopyEdit
Route::get('/user/browsing-history', [UserHistoryController::class,
    'index'])->name('user.browsing-history');
```

- **Propose Article** : Route pour proposer un article via la méthode `showProposalForm` du `UserProposalController`.

```
php
CopyEdit
Route::get('/user/proposearticle', [UserProposalController::class,
    'showProposalForm'])->name('user.proposearticle');
```

- **Settings** : Route pour afficher et mettre à jour les paramètres utilisateur via les méthodes `settings` du `UserController` et `update` du `UserSettingsController`.

```
php
CopyEdit
Route::get('/user/settings', [UserController::class, 'settings'])->name('user.settings');
Route::put('/settings/update', [UserSettingsController::class, 'update'])->name('user.settings.update');
```

### 3. Contrôleurs

Chaque route pointe vers un contrôleur qui est responsable de la logique associée à cette section. Par exemple :

- `DashboardUserController` : Gère la page d'accueil du tableau de bord utilisateur.
- `UserSubscriptionController` : Gère l'affichage et la logique des abonnements utilisateur.
- `UserArticleController` : Gère les articles de l'utilisateur.
- `UserHistoryController` : Gère l'historique de navigation de l'utilisateur.
- `UserProposalController` : Permet à l'utilisateur de proposer des articles.

### 8-5. Home page

Cette page web constitue l'interface principale de Tech Horizons, permettant aux utilisateurs d'explorer un magazine en ligne spécialisé dans les innovations technologiques

#### Analyse et conception de la page :

L'interface web est structurée en plusieurs sections essentielles :

- Une barre de navigation comprenant un logo, une barre de recherche et des liens vers les pages de connexion, d'inscription et de consultation des thèmes du magazine.
- Une section principale qui introduit Tech Horizons avec un message de bienvenue et un appel à l'action encourageant les utilisateurs à explorer le contenu.
- Une section dédiée aux éditions du magazine, affichant des cartes représentant les différentes publications disponibles, avec un bouton permettant d'accéder aux articles correspondants.
- Un pied de page qui fournit des liens vers des ressources importantes, ainsi que des icônes de réseaux sociaux pour améliorer l'engagement des utilisateurs.

## **8-6. Page thèmes**

La page des thèmes offre une vue d'ensemble des différentes catégories de contenu disponibles sur Tech Horizons. Elle présente quatre cartes thématiques principales : Intelligence Artificielle, Internet des Objets, Cybersécurité et Réalité Virtuelle. Chaque carte comporte une image représentative, un titre, une brève description, le nombre d'articles publiés et un score moyen affiché sous forme d'étoiles. Le design utilise une grille responsive (grid-template-columns avec auto-fit) qui s'adapte automatiquement à la taille de l'écran. Les cartes sont interactives, avec des effets de survol et une animation de translation vers le haut. Un système de notation sophistiqué affiche des étoiles pleines, demi-pleines ou vides selon les scores attribués. La page inclut également une version mobile optimisée avec des media queries pour réorganiser le contenu sur les petits écrans.

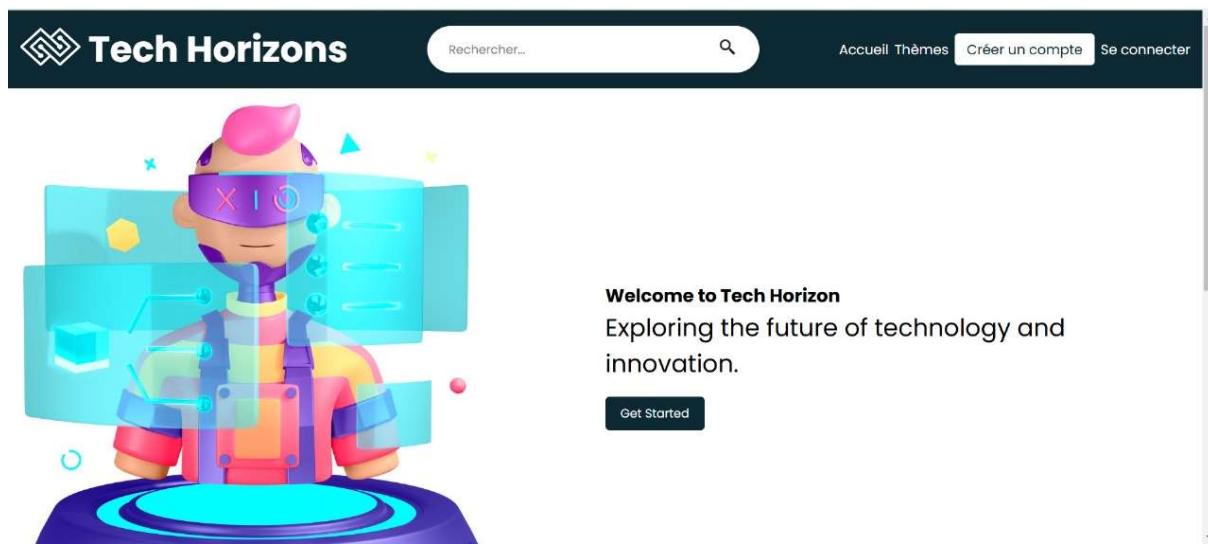
## **8-7. Page Articles**

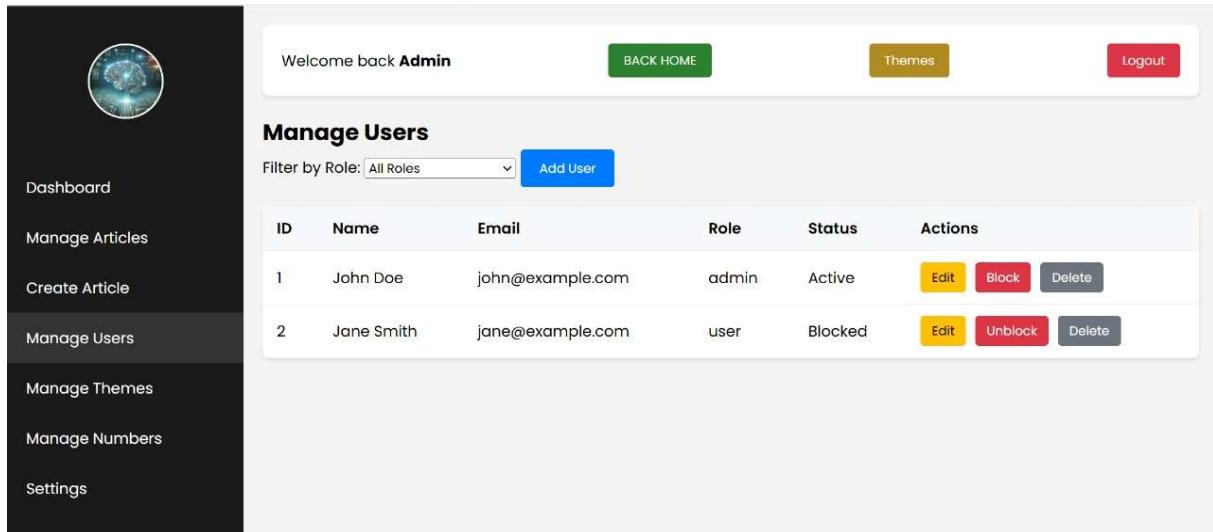
La page article présente une structure dédiée à la lecture approfondie d'un contenu spécifique. L'article commence par un grand titre, suivi de la date de publication et du nom de l'auteur. Une image principale illustre le sujet, suivie du corps de l'article. La page intègre plusieurs fonctionnalités interactives : un système de notation avec des étoiles, un bouton pour recommander l'article, et une section de commentaires complète. Cette dernière permet aux lecteurs d'ajouter leurs commentaires et de répondre aux commentaires existants. Le design est épuré, centré sur le contenu, avec une largeur maximale de 800 pixels pour garantir une lecture confortable. Les interactions utilisateur sont gérées par JavaScript, notamment pour la gestion des étoiles et des commentaires.

## 8-8. Page Annonces

La page d'annonces sert de vitrine aux articles publiés sur Tech Horizons. Son en-tête comporte le logo du site accompagné du titre "Tech Horizons", une barre de recherche centrale et un menu de navigation. La section principale affiche trois articles majeurs sous forme de cartes, chacune contenant une image, un titre, un résumé et des statistiques de lecture. Ces articles couvrent des sujets d'actualité comme l'Intelligence Artificielle, la Cybersécurité et la Réalité Virtuelle. Chaque carte d'article propose deux boutons : "Lire plus" et "Partager". Le design utilise une disposition flexible avec des conteneurs article-container qui organisent le contenu en deux colonnes : le texte à gauche et l'image à droite. Cette mise en page assure une lecture fluide et une présentation visuelle attrayante des informations.

## 8. Captures d'écran de l'application





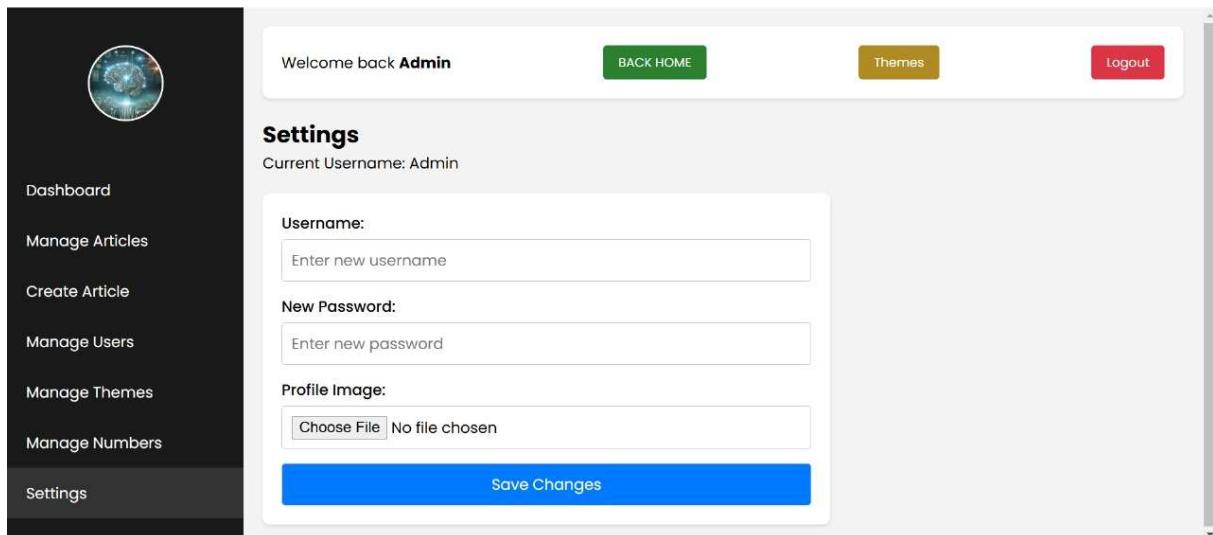
Welcome back **Admin**

BACK HOME Themes Logout

## Manage Users

Filter by Role: All Roles Add User

ID	Name	Email	Role	Status	Actions
1	John Doe	john@example.com	admin	Active	<button>Edit</button> <button>Block</button> <button>Delete</button>
2	Jane Smith	jane@example.com	user	Blocked	<button>Edit</button> <button>Unblock</button> <button>Delete</button>



Welcome back **Admin**

BACK HOME Themes Logout

## Settings

Current Username: Admin

Username:

New Password:

Profile Image:  No file chosen

**Save Changes**

Welcome back **Admin**

BACK HOME Themes Logout

### Manage Magazine Numbers

Filter by Status: All Statuses

Number	Title	Publication Date	Status	Actions
1	Issue 1	2023-01-01	Public	<button>Make Public</button> <button>Make Private</button> <button>Remove</button>
2	Issue 2	2023-02-01	Private	<button>Make Public</button> <button>Make Private</button> <button>Remove</button>

[Add Number](#)

Dashboard Manage Articles Create Article Manage Users Manage Themes **Manage Numbers** Settings

Welcome back **Admin**

BACK HOME Themes Logout

### Manage Themes

Filter by Status: All Statuses

Theme	Responsible User	Articles Count	Status	Actions
Technology	John Doe	15	Public	<button>Change Responsible</button> <button>Remove Responsible</button> <button>Toggle Status</button>
Science	Jane Smith	10	Private	<button>Change Responsible</button> <button>Remove Responsible</button> <button>Toggle Status</button>
Health	Unassigned	5	Public	<button>Change Responsible</button> <button>Remove Responsible</button> <button>Toggle Status</button>
Business	Unassigned	8	Private	<button>Change Responsible</button> <button>Remove Responsible</button> <button>Toggle Status</button>

Dashboard Manage Articles Create Article Manage Users **Manage Themes** Manage Numbers Settings



Welcome back **Admin**

[BACK HOME](#) [Themes](#) [Logout](#)

## Create Article

**Dashboard**

**Manage Articles**

**Create Article**

**Manage Users**

**Manage Themes**

**Manage Numbers**

**Settings**

**Title:**  
Enter the article title

**Theme:**  
Technology

**Status:**  
Published

**Cover Image:**  
 No file chosen

**Content:**  
Enter the article content

[Create Article](#)



Welcome back **Admin**

[BACK HOME](#)

[Themes](#)

[Logout](#)

## Latest Statistics

Subscribers		Themes		Numbers	
Total Subscribers	1,500	Total Themes	5	Total Numbers	10
Active Subscribers	1,200	Active Themes	0	Published Numbers	8
Articles					
Total Articles	50	Published Articles	40	Archived Articles	10



# Tech Horizons

Search icon

[Accueil](#) [Thèmes](#) [Créer un compte](#) [Se connecter](#)

## Explorez nos thèmes



**Intelligence Artificielle**

Découvertes et applications de l'IA.

Articles publiés : 42

★★★★★



**Internet des Objets**

Connectivité et innovations IoT.

Articles publiés : 30

★★★★★



**Cybersécurité**

Protéger vos données et systèmes.

Articles publiés : 25

★★★★★



**Réalité Virtuelle**

Expériences immersives et technologies AR/VR.

Articles publiés : 18

★★★★★

**Tech Horizons**

Rechercher...

Accueil Thèmes Mes articles Se déconnecter

### Annonces des Articles Publié

Découvrez les derniers articles publiés sur Tech Horizons

**Les dernières innovations en Intelligence Artificielle**  
Explorez comment l'IA transforme les différents secteurs industriels et son impact sur notre quotidien.  
Publié le : 2025-01-10 | Vues : 120.

**La Cybersécurité : Enjeux et Perspectives**  
Analyse des menaces cybernétiques actuelles et des solutions pour les contrer efficacement.  
Publié le : 2025-01-08 | Vues : 98.

**Réalité Virtuelle et Augmentée : Une immersion sans limite**  
Découvrez comment les technologies immersives redéfinissent le divertissement et la formation.  
Publié le : 2025-01-05 | Vues : 75.

[Lire plus](#) [Partager](#)

[Lire plus](#) [Partager](#)

[Lire plus](#) [Partager](#)

**Tech Horizons**

Tech Horizons est votre guide pour comprendre les transformations technologiques  
et leur impact sur notre quotidien.

Rechercher...

Accueil Thèmes Mes articles Se déconnecter

## L'Intelligence Artificielle : Une Révolution en Marche

Publié le 01/01/2025 | Par Alexis Dupont



L'intelligence artificielle (IA) est aujourd'hui au cœur des transformations technologiques majeures, redéfinissant les frontières de ce que les machines peuvent accomplir. Mais qu'est-ce que l'IA, et pourquoi suscite-t-elle autant d'intérêt et de débats ?

**Une technologie au service de l'humain**

L'IA désigne des systèmes informatiques capables de simuler l'intelligence humaine. Qu'il s'agisse de reconnaître des visages, de résoudre des tâches complexes ou encore d'améliorer la précision des prévisions météorologiques, l'IA joue un rôle croissant dans notre quotidien.

**Tech Horizons**

Rechercher...

Accueil Thèmes [Créer un compte](#) [Se connecter](#)

### Contact Information

Si vous avez des questions, des commentaires ou des demandes de renseignements, n'hésitez pas à nous contacter via ce qui suit :

Email: [contact@techhorizons.ma](mailto:contact@techhorizons.ma)  
Téléphone: +212 500 000 000  
Adresse: Tech Horizons Maroc  
Avenue des Technologies,  
Quartier Innovation,  
Casablanca 20200,  
Maroc

### Envoyez-nous un message

Votre nom:

## À propos de l'application Tech Horizons

Tech Horizons est une application web novatrice, spécifiquement conçue pour répondre aux besoins d'un public avide de découvertes technologiques. Axée sur la gestion d'un magazine en ligne, elle couvre un large éventail de thématiques liées aux innovations majeures de notre époque, tout en offrant une plateforme collaborative et interactive.

Cette application a été pensée pour simplifier l'accès aux contenus technologiques, tout en apportant une touche de personnalisation et d'intelligence. Elle permet aux utilisateurs, qu'ils soient simples curieux, amateurs passionnés ou professionnels du domaine, de s'informer, de participer activement et de contribuer à la création de contenus pertinents.

Tech Horizons ne se limite pas à une simple plateforme de lecture. Elle se veut être un espace de référence, un lieu où la technologie et ses enjeux éthiques, sociaux et culturels peuvent être explorés, discutés et partagés. En mettant l'accent sur l'innovation, la personnalisation et la collaboration, l'application offre à ses utilisateurs la possibilité non seulement de consommer du contenu, mais aussi de devenir acteurs des transformations technologiques de demain. Avec une approche centrée sur l'utilisateur et une technologie de pointe, Tech Horizons ambitionne de redéfinir la manière dont les passionnés de technologie explorent et interagissent avec le savoir numérique.



**Tech Horizons**

Tech Horizons est votre guide pour comprendre les transformations technologiques majeures et leurs impacts sur l'avenir.

**NAVIGATION**

- Accueil
- Thèmes

**SUPPORT**

- Contact Us
- FAQ

**SUIVEZ-NOUS**

- [Facebook](#)
- [Twitter](#)
- [Instagram](#)
- [LinkedIn](#)

Welcome back, **Responsible**

[Logout](#)

**Manage Subscriber Proposals**

**Subscriber Article Proposals**

Title	Theme	Status	Actions
Building a Positive Mindset	Motivation	Proposed	<a href="#">Delete</a> <a href="#">Propose for Editing</a>
Time Management Tips	Productivity	Proposed	<a href="#">Delete</a> <a href="#">Propose for Editing</a>

**Dashboard**

**Articles**

**Subscribers**

**Subscriber Proposals**

**Conversations**

**Statistics**

**Profile Settings**



Welcome back, **Responsible** [Logout](#)

## Statistics

<b>Total Articles</b> <b>0</b>	<b>Total Subscribers</b> <b>0</b>	<b>Article Views</b> <b>0</b>
-----------------------------------	--------------------------------------	----------------------------------

- Dashboard
- Articles
- Subscribers
- Subscriber Proposals
- Conversations
- Statistics**
- Profile Settings



Welcome back, **Responsable**

[Logout](#)

### Dashboard Overview

Articles	Subscribers	Conversations
0	0	0

[Dashboard](#)

[Articles](#)

[Subscribers](#)

[Subscriber Proposals](#)

[Conversations](#)

[Statistics](#)

[Profile Settings](#)



Welcome back, **Responsable**

[Logout](#)

### Manage Subscribers

Username	Email	Status	Actions
JohnDoe	johndoe@example.com	Subscribed	<a href="#">Unsubscribe</a>
JaneSmith	janesmith@example.com	Not Subscribed	<a href="#">Subscribe</a>

[Dashboard](#)

[Articles](#)

[Subscribers](#)

[Subscriber Proposals](#)

[Conversations](#)

[Statistics](#)

[Profile Settings](#)



Welcome back, **Admin**

[Themes](#)

[Logout](#)

### Propose un Article

**Titre de l'article**

**Thèmes de l'article**

- Thème 1
- Thème 2
- Thème 3

**Image de couverture**

**Description de l'article**

[Proposer l'article](#)

[Dashboard](#)

[Subscription](#)

[My Article](#)

[Browsing History](#)

[Propose Article](#)

[Settings](#)



- [Dashboard](#)
- [Subscription](#)
- [My Article](#)
- [Browsing History](#)
- [Propose Article](#)
- [Settings](#)

Welcome back, Admin

Themes
Logout

### Historique de Navigation

Filtrer par titre:

Filtrer par Thème:

Tous les thèmes

Filtrer par Statut:

Tous les statuts

Filtrer par Date:

Toutes les Dates

Titre de l'Article	Thèmes	Statut	Date
Article 1	theme1	Publié	28 janvier 2025
Article 2	theme2	En cours	27 janvier 2025
Article 3	theme1, theme3	Retenu	21 janvier 2025
Article 4	theme3	Refus	5 janvier 2025



- [Dashboard](#)
- [Subscription](#)
- [My Article](#)
- [Browsing History](#)
- [Propose Article](#)
- [Settings](#)

Welcome back, Admin

Themes
Logout

### Tableau de Bord

#### Articles Recommandés

**Article 1**

**Thèmes:** theme1

**Status:** Publié

**Date:** 28/01/2025



**Article 2**

**Thèmes:** theme2

**Status:** En cours

**Date:** 27/01/2025



#### Numéros du Magazine



**Numéro 1 - Janvier 2025**

Le premier numéro du mois de janvier.



**Numéro 2 - Février 2025**



- [Dashboard](#)
- [Subscription](#)
- [My Article](#)
- [Browsing History](#)
- [Propose Article](#)
- [Settings](#)

Welcome back, Admin

Themes
Logout

### Subscription

Manage your subscription and billing details.

Theme Name	Status	Action
Theme 1	Not Subscribed	<button>Subscribe</button>
Theme 2	Not Subscribed	<button>Subscribe</button>
Theme 3	Not Subscribed	<button>Subscribe</button>

**Save Changes**



Welcome back, Admin

Themes

Logout

### Settings

Username:

New Password:

Profile Image:

Save Changes

This screenshot shows the 'Settings' page for an administrator. It includes fields for changing the username and password, and a file input for updating the profile image. A green 'Save Changes' button is at the bottom.



Welcome back, Admin

Themes

Logout

### Mes Articles

Titre de l'Article	Thèmes	Statut

This screenshot shows the 'Mes Articles' (My Articles) section. It displays a table header with columns for 'Titre de l'Article', 'Thèmes', and 'Statut'. There are currently no articles listed in the table body.



### Sign Up

Full Name

Username

Email

Password

Confirm Password

Gender

Male  Female

Sign Up

This screenshot shows a 'Sign Up' form. It includes fields for 'Full Name', 'Username', 'Email', 'Password', 'Confirm Password', and gender selection ('Male' or 'Female'). A large green 'Sign Up' button is at the bottom.

## Login

Email 

Password 

Remember me

**Login**

Don't have an account? [Register](#)