

# **1. Cahier de Charges**

## *1. Objectifs du Projet YTG*

Le projet YTG vise à développer une plateforme technologique en ligne pour la gestion de clients et de services proposés par YAVE Technologies Group. Cette plateforme accessible depuis un ordinateur, une tablette ou un smartphone, offrant une expérience utilisateur optimale à ses utilisateurs.

## *2. Fonctionnalités à Implémenter*

La plateforme inclure les fonctionnalités suivantes :

- **Accueil** : Page principale présentant une vue d'ensemble du site avec des informations sur les services offerts et une navigation claire vers les différentes sections.
- **À Propos** : Section détaillant la vision, la mission, l'équipe, la feuille de route et les partenaires de YTG.
- **Consultation** : Services de consultation incluant l'IA, la Blockchain, la sécurité informatique, les bases de données et les réseaux informatiques.
- **Formation** : Offre de formations continues et personnalisées dans des domaines comme la Data Science, l'IA, la Blockchain, la Cybersecurity, le Génie Logiciel, et les Réseaux Informatiques.
- **Tutoriat** : Listing des cours universitaires disponibles à l'UPB et dans une autre université, avec un formulaire d'inscription intégré.
- **Cryptomonnaie** : Guide pour la création de portefeuilles (Wallet) Metamask et modules d'envoi (SEND) et de conversion (SWAP) de cryptomonnaie.
- **Contact** : Coordonnées de YTG et informations de localisation géographique.
- **Connexion** : Page permettant la création de compte membre avec gestion des profils utilisateurs et stockage sécurisé des données personnelles.

## *3. Technologies Utilisées*

Pour la réalisation de ce projet, les technologies suivantes seront utilisées :

- **Frontend** : React.js pour le développement de l'interface utilisateur interactive et dynamique.
- **Backend** : Spring boot pour la création d'une API robuste et sécurisée.
- **Base de données** : MySQL pour le stockage des données utilisateur et transactionnelles.
- **Blockchain** : Intégration de Metamask pour les fonctionnalités de cryptomonnaie.
- **Sécurité** : Utilisation de JWT (JSON Web Tokens) pour l'authentification et l'autorisation des utilisateurs.

#### *4. Rôles et Responsabilités des Membres de l'Équipe*

L'équipe est composée de développeurs frontend, backend, de designers UI/UX, et de testeurs. Chaque membre a des responsabilités spécifiques dans le développement, le design, les tests, et la gestion de projet selon les méthodologies SCRUM , nous étions au total 2.

- **Product Owner** : Responsable de la définition des besoins métier et de la priorisation des fonctionnalités.
- **Scrum Master** : Garant de l'application des pratiques SCRUM et de la gestion des sprints.
- **Développeurs** : Implémentation des fonctionnalités frontend et backend, intégration de l'API et gestion des bases de données.
- **Designers** : Conception des interfaces utilisateur intuitives et attrayantes.
- **Testeurs** : Mise en œuvre des tests unitaires, d'intégration et d'acceptation pour assurer la qualité du logiciel.

#### *5. Délais et Calendrier de Réalisation*

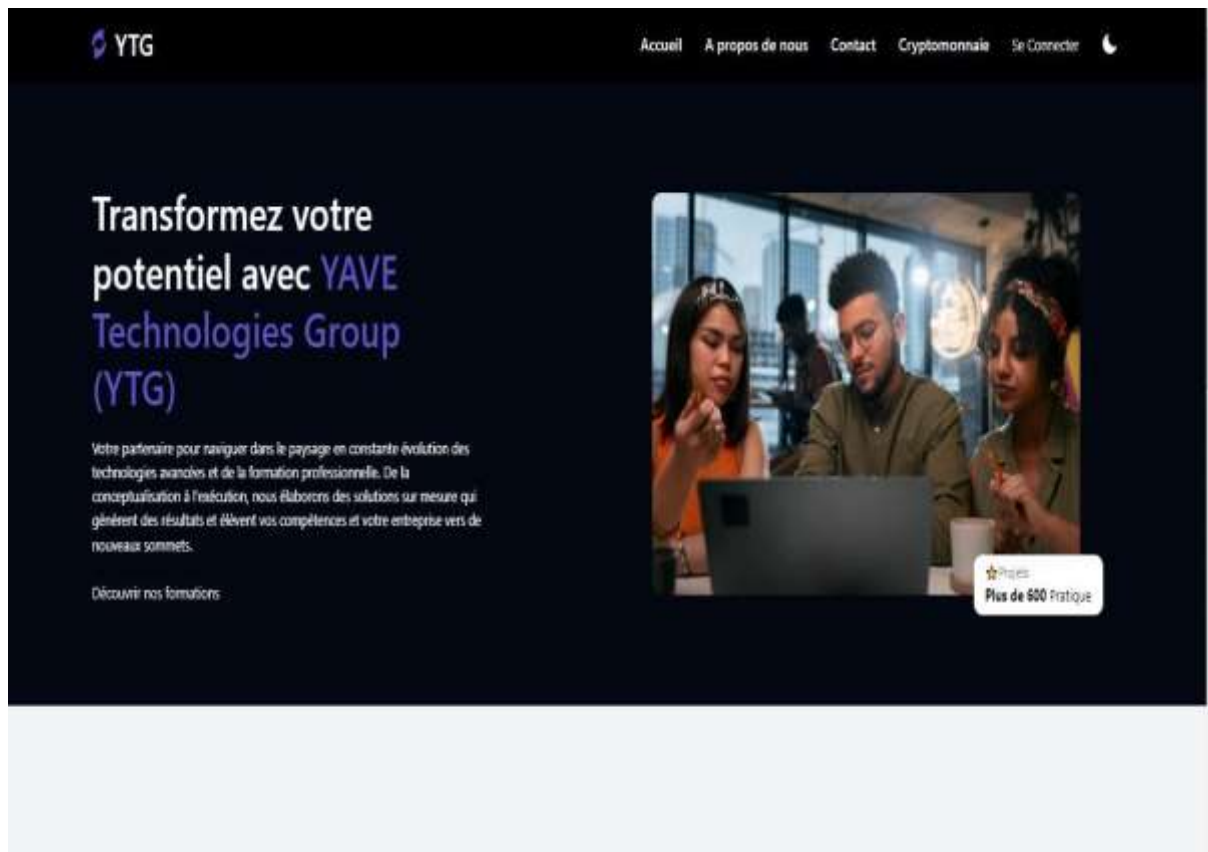
Le projet sera développé sur une période de 3 semaines, découpée en sprints de 1 semaine chacun. Chaque sprint comprendra la planification, le développement, les tests et la revue. Les livraisons incrémentales permettront de suivre l'évolution du projet et de répondre aux feedbacks des utilisateurs.

#### *6. Critères de Réussite du Projet*

- Déploiement complet de toutes les fonctionnalités décrites dans le cahier des charges.
- Validation par les utilisateurs finaux des fonctionnalités de cryptomonnaie (SEND et SWAP).
- Stabilité et performance de la plateforme sous charge.

## **2. Les 7 pages web de la plateforme**

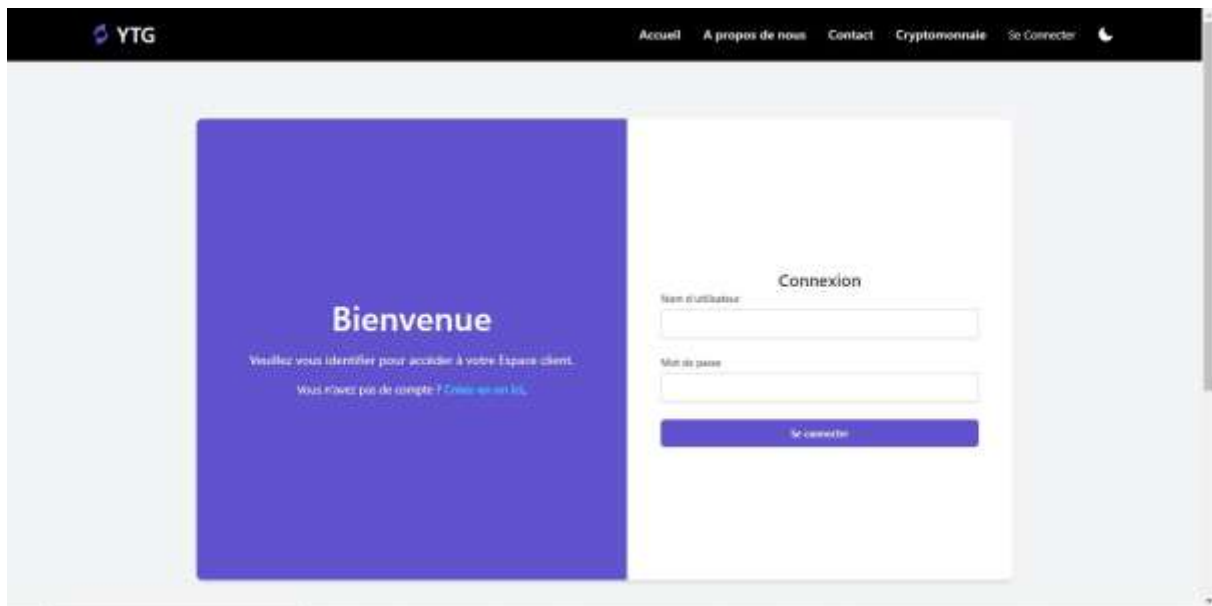
La page d'accueil est la porte d'entrée principale de notre site, conçue pour accueillir les visiteurs et leur offrir une vue d'ensemble claire et engageante des services ou produits proposés.



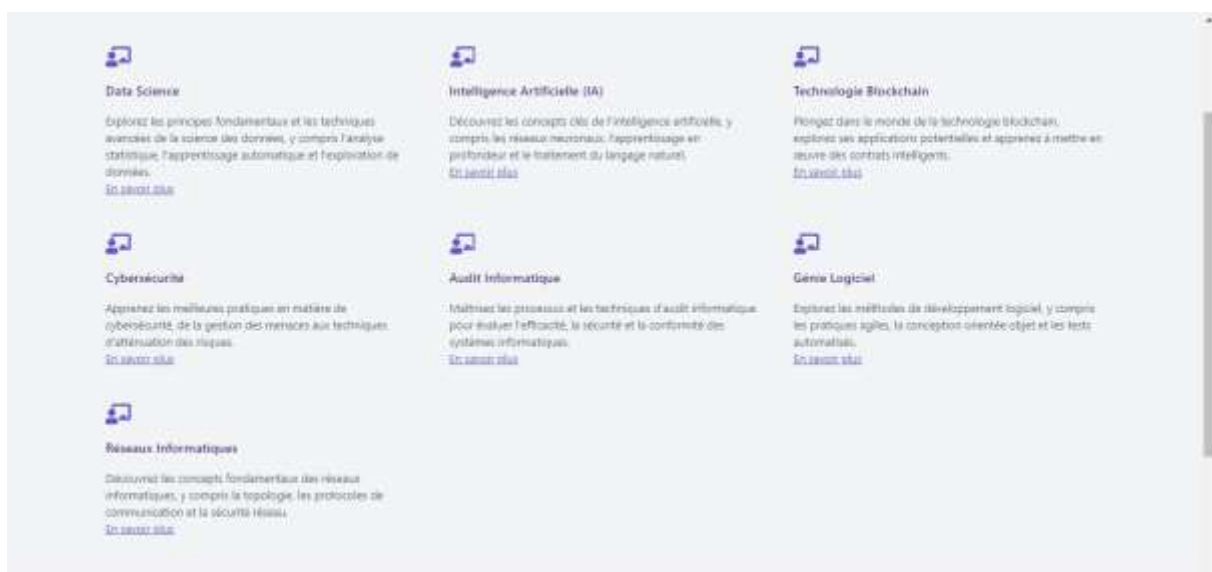
La page dédiée à l'inscription permet à toute entité de s'enregistrer de manière structurée et sécurisée. Sur cette page, les utilisateurs peuvent saisir les informations requises pour créer un compte et accéder aux fonctionnalités du système.

The image shows a registration form titled 'Inscription'. The form has the following fields: 'Nom d'utilisateur', 'Prénom', 'Nom', 'Téléphone', 'Adresse e-mail', 'Université', 'Société', 'Pays', 'Mot de passe', and 'Confirmez le mot de passe'. There is a 'Choisir un fichier' button next to the 'Pays' field. At the bottom of the form, there is a 'Se connecter' button and a link 'Vous n'avez pas de compte ? Cliquez ici'.

La page de connexion est conçue pour permettre aux utilisateurs enregistrés de se connecter de manière sécurisée. En saisissant leurs identifiants, ils peuvent accéder à leur compte et profiter pleinement des fonctionnalités offertes par le système.



La page dédiée aux formations est spécialement conçue pour offrir aux utilisateurs une vue complète et détaillée des différentes opportunités de formation disponibles. Sur cette page, les utilisateurs peuvent explorer une variété de programmes éducatifs et de développement professionnel.



La page "À propos de nous" est dédiée à fournir aux visiteurs une compréhension approfondie de l'organisation. Cette section présente l'histoire de l'entreprise, sa mission, ses valeurs fondamentales et sa vision pour l'avenir.



L'interface d'échange des crypto:



### 3. Documentations utilisateur et développeur

**Guide d'Installation et de Configuration pour un Projet React et Spring Boot**

## Prérequis

Assurez-vous d'avoir les logiciels suivants installés sur votre machine :

1. **Node.js et npm** : Téléchargez Node.js [ici](#).
2. **Java Development Kit (JDK) 17 ou supérieur** : Téléchargez le JDK [ici](#).
3. **Maven** : Téléchargez Maven [ici](#).

## Configuration des Variables d'Environnement

1. Créez un fichier `application.properties` dans le répertoire `src/main/resources`.
2. Ajoutez-y les configurations nécessaires, telles que les détails de la base de données :

```
properties Copier
le code
# Exemple de configuration de la base de données
spring.datasource.url=jdbc:mysql://localhost:3306/votre_base_de_donnees
spring.datasource.username=votre_nom_utilisateur
spring.datasource.password=votre_mot_de_passe

spring.jpa.hibernate.ddl-auto=update spring.jpa.show-sql=true

# Autres configurations
```

## Compilation et Exécution de l'Application Spring Boot

1. Utilisez Maven pour compiler l'application :

```
bash
Copier le code mvn
clean install
```

2. Puis, exécutez l'application Spring Boot :

```
bash
Copier le code mvn
spring-boot:run
```

## Configuration du Frontend (React)

1. Accédez au répertoire du frontend.
2. Installez les dépendances nécessaires en utilisant `yarn` :

```
bash
Copier le code yarn
install
```

3. Démarrez l'application React avec la commande suivante :

```
bash
Copier le code yarn
start
```

## Accéder à l'Application

Le frontend React sera accessible à l'adresse suivante : <http://localhost:3000>.