**Cahier des Charges pour une Plateforme de Réseau Social pour Développeurs**

**Introduction**

La plateforme sera un réseau social destiné aux développeurs, offrant des fonctionnalités de blog, de portfolio, et de formation. Les utilisateurs pourront créer des articles de blog, des portfolios personnalisés, et des formations payantes. La plateforme générera des revenus grâce aux commissions sur les ventes de formations.

**Objectifs**

1. Créer un réseau social interactif pour les développeurs.
2. Permettre aux utilisateurs de publier des articles de blog et de créer des portfolios.
3. Offrir une section de formation où les utilisateurs peuvent vendre des cours.
4. Générer des revenus par le biais de commissions sur les ventes de formations.

**Fonctionnalités Principales**

**Utilisateurs**

* **Inscription et Authentification**
  + Inscription par email, Google, GitHub.
  + Authentification avec système de récupération de mot de passe.
* **Profils Utilisateurs**
  + Profils avec photo, bio, compétences, réseaux sociaux.
  + Possibilité de suivre d'autres utilisateurs.

**Blog**

* **Publication d'Articles**
  + Création, modification et suppression d'articles.
  + Catégorisation et taggage des articles.
  + Système de commentaires.
* **Interaction Sociale**
  + Like et partage des articles.
  + Notifications pour les interactions (commentaires, likes).

**Portfolio**

* **Création de Portfolio**
  + Section pour les projets, expériences professionnelles, compétences.
  + Personnalisation avec thèmes et templates.

**Formation**

* **Espace Formateur**
  + Inscription en tant que formateur avec vérification.
  + Création, modification et suppression de cours.
  + Mise en place d'un système de prix pour les cours.
* **Espace Apprenant**
  + Achat de cours.
  + Suivi de la progression des cours.
* **Commission**
  + Mise en place d'un pourcentage de commission sur chaque vente de formation.

**Technologies Utilisées**

**Frontend**

* **React**
* **Redux** pour la gestion des états
* **Vite** pour le bundling
* **TypeScript** pour le typage

**Backend**

* **Django**
  + Django REST Framework pour l'API
  + Django ORM pour la gestion de la base de données

**Base de Données**

* **PostgreSQL** pour la gestion des données relationnelles

**Authentification et Autorisation**

* **JWT** (JSON Web Token) pour les sessions utilisateur

**Stockage**

* **Amazon S3** pour le stockage des fichiers (images, vidéos de cours)

**Déploiement**

* **Docker** pour la containerisation
* **Kubernetes** pour l'orchestration des conteneurs
* **CI/CD** avec GitHub Actions

**Structure de l'Application**

**Frontend**

1. **Pages Publiques**
   * Page d'accueil
   * Page de blog
   * Page de portfolio
   * Page de formation
2. **Pages Authentifiées**
   * Tableau de bord utilisateur
   * Création/Modification de blog
   * Création/Modification de portfolio
   * Création/Modification de cours

**Backend**

1. **Modèles**
   * Utilisateur
   * Blog
   * Portfolio
   * Formation
   * Transaction
2. **API REST**
   * Endpoints pour l'inscription/authentification
   * Endpoints CRUD pour blog, portfolio, formation
   * Endpoints pour la gestion des transactions

**Stratégies de Monétisation**

1. **Commissions sur les Cours**
   * Pourcentage de chaque vente de cours.
2. **Abonnement Premium**
   * Offrir des fonctionnalités supplémentaires pour les utilisateurs payants (analyse avancée, thèmes premium, plus d'espace de stockage).
3. **Publicité**
   * Intégrer des annonces ciblées sur la plateforme.
4. **Partenariats**
   * Collaboration avec entreprises pour des formations sponsorisées.

**Plan de Développement**

**Étape 1 : Planification**

* Définir les exigences détaillées.
* Établir le plan de projet et le calendrier.

**Étape 2 : Conception**

* Conception des maquettes et wireframes.
* Définir l'architecture de la base de données.

**Étape 3 : Développement**

* Mise en place de l'environnement de développement.
* Développement du frontend et du backend.
* Intégration des API.

**Étape 4 : Tests**

* Tests unitaires et intégration.
* Tests utilisateurs bêta.

**Étape 5 : Déploiement**

* Configuration des serveurs de production.
* Mise en place de CI/CD.
* Déploiement initial.

**Étape 6 : Maintenance**

* Support et correction des bugs.
* Mise à jour régulière des fonctionnalités.

**Conclusion**

Ce cahier des charges fournit une base détaillée pour la création d'une plateforme de réseau social pour développeurs. En suivant ce plan, vous pourrez concrétiser votre idée en une application fonctionnelle et rentable.

**Plan de Sprints pour le Développement de la Plateforme de Réseau Social pour Développeurs**

**Introduction**

Le développement de la plateforme sera divisé en sprints, chaque sprint ayant une durée de 2 semaines. Chaque sprint aura des objectifs spécifiques et des livrables clairement définis. Voici un plan détaillé des sprints, étape par étape, du début du projet jusqu'à la fin.

**Sprint 1: Initialisation et Planification**

**Objectifs :**

* Mise en place de l'environnement de développement.
* Définir les exigences détaillées et le plan de projet.

**Livrables :**

* Cahier des charges finalisé.
* Environnement de développement configuré (frontend et backend).
* Backlog de produit créé avec toutes les user stories initiales.

**Sprint 2: Conception et Architecture**

**Objectifs :**

* Conception des maquettes et wireframes.
* Définir l'architecture de la base de données et de l'application.

**Livrables :**

* Wireframes pour les principales pages (authentification, tableau de bord, blog, portfolio, formation).
* Modèle de base de données défini.
* Diagramme de l'architecture de l'application.

**Sprint 3: Authentification et Autorisation**

**Objectifs :**

* Implémentation de l'inscription et de l'authentification des utilisateurs.
* Mise en place de JWT pour les sessions utilisateur.

**Livrables :**

* API d'inscription et d'authentification.
* Interface utilisateur pour l'inscription et la connexion.
* Authentification sécurisée avec JWT.

**Sprint 4: Gestion des Profils Utilisateurs**

**Objectifs :**

* Développer les fonctionnalités de gestion des profils utilisateurs.
* Permettre la personnalisation des profils (photo, bio, compétences).

**Livrables :**

* API pour la gestion des profils.
* Interface utilisateur pour l'édition des profils.
* Stockage des images de profil sur Amazon S3.

**Sprint 5: Module de Blog - Partie 1**

**Objectifs :**

* Implémenter la création, modification et suppression d'articles de blog.
* Développer l'interface utilisateur pour les articles de blog.

**Livrables :**

* API CRUD pour les articles de blog.
* Interface utilisateur pour la gestion des articles de blog.

**Sprint 6: Module de Blog - Partie 2**

**Objectifs :**

* Ajouter des fonctionnalités de catégorisation et de taggage des articles.
* Développer le système de commentaires et d'interactions sociales (like, partage).

**Livrables :**

* API pour la gestion des catégories et tags.
* Interface utilisateur pour la gestion des commentaires.
* Notifications pour les interactions sur les articles.

**Sprint 7: Module de Portfolio - Partie 1**

**Objectifs :**

* Développer les fonctionnalités de création et modification de portfolios.
* Permettre la personnalisation avec thèmes et templates.

**Livrables :**

* API pour la gestion des portfolios.
* Interface utilisateur pour la création et la modification des portfolios.

**Sprint 8: Module de Portfolio - Partie 2**

**Objectifs :**

* Ajouter des fonctionnalités avancées de personnalisation (projets, expériences, compétences).
* Intégrer le stockage des fichiers (images de projet) sur Amazon S3.

**Livrables :**

* API pour la gestion des projets, expériences et compétences.
* Interface utilisateur améliorée pour les portfolios.
* Stockage des fichiers sur Amazon S3.

**Sprint 9: Module de Formation - Partie 1**

**Objectifs :**

* Implémenter les fonctionnalités de création et gestion de cours pour les formateurs.
* Développer l'interface utilisateur pour les formateurs.

**Livrables :**

* API CRUD pour les cours de formation.
* Interface utilisateur pour la gestion des cours par les formateurs.

**Sprint 10: Module de Formation - Partie 2**

**Objectifs :**

* Ajouter les fonctionnalités de paiement et de suivi des cours pour les apprenants.
* Implémenter le système de commission sur les ventes de cours.

**Livrables :**

* Intégration des API de paiement.
* Interface utilisateur pour l'achat et le suivi des cours.
* Système de commission mis en place.

**Sprint 11: Tests et Débogage**

**Objectifs :**

* Effectuer des tests unitaires et d'intégration sur toutes les fonctionnalités développées.
* Corriger les bugs identifiés pendant les tests.

**Livrables :**

* Suite de tests automatisés.
* Rapport de tests et corrections de bugs.

**Sprint 12: Optimisation et Sécurité**

**Objectifs :**

* Optimiser les performances de l'application.
* Assurer la sécurité de l'application (protection contre les attaques XSS, CSRF, etc.).

**Livrables :**

* Améliorations de performance (caching, optimisation des requêtes).
* Mesures de sécurité mises en place.

**Sprint 13: Préparation au Déploiement**

**Objectifs :**

* Préparer l'application pour le déploiement en production.
* Mettre en place CI/CD avec GitHub Actions.

**Livrables :**

* Configuration de Docker et Kubernetes.
* Pipelines CI/CD configurés.

**Sprint 14: Déploiement et Lancement**

**Objectifs :**

* Déployer l'application en production.
* Lancer officiellement la plateforme.

**Livrables :**

* Application déployée sur des serveurs de production.
* Lancement de la plateforme et communication marketing.

**Sprint 15: Feedback et Améliorations**

**Objectifs :**

* Recueillir les feedbacks des premiers utilisateurs.
* Apporter des améliorations et corrections basées sur les feedbacks.

**Livrables :**

* Rapport de feedback utilisateur.
* Améliorations et corrections apportées.

**Conclusion**

En suivant ce plan de sprints, vous pouvez développer la plateforme de manière structurée et itérative, en vous assurant que chaque composant est soigneusement planifié, développé et testé avant de passer à la phase suivante.