



Introduction

1. Objectif du projet : Créer une plateforme en ligne permettant aux élèves de trouver des cours de soutien et aux professeurs de proposer leurs services, tout en offrant une expérience utilisateur conviviale et intuitive.
2. But de la plateforme : Faciliter l'accès aux cours de soutien pour les élèves, offrir aux professeurs une plateforme pour proposer leurs services, et encourager l'apprentissage continu.
3. Public cible : Élèves à la recherche de cours de soutien, professeurs souhaitant dispenser des cours particuliers, parents cherchant des ressources éducatives supplémentaires pour leurs enfants.

Fonctionnalité

a. Interface Administrateur :

- Connexion sécurisée à un tableau de bord administratif.
- Gestion des comptes utilisateurs, des profils des enseignants et des annonces des élèves.
- Suivi des cours réservés, des paiements effectués et des activités sur la plateforme.
- Fonctionnalités de modération et de gestion de contenu.

b. Interface Enseignant :

- Crédit et gestion de profils enseignants avec des informations détaillées sur leurs compétences, leur expérience et leurs disponibilités.
- Ajout, édition et suppression d'annonces de cours dans le catalogue.
- Suivi des réservations de cours et des paiements reçus.
- Communication avec les élèves via messagerie interne.

c. Interface Élève :

- Inscription et création de compte élève.
- Navigation intuitive à travers les annonces de cours disponibles.
- Réservation de cours et paiement en ligne sécurisé via PayPal.
- Suivi des cours réservés et de l'historique des paiements.
- Possibilité de laisser des commentaires et des évaluations sur les enseignants et les cours suivis.

d. Catalogue de cours :

- Mise en place d'un catalogue en ligne regroupant les annonces de cours proposées par les élèves.
- Système de recherche avancée permettant aux élèves de trouver facilement des cours spécifiques.
- Présentation détaillée des annonces de cours avec des descriptions, des niveaux d'étude concernés et les prix proposés.

e. Processus de réservation :

- Sélection d'un cours et ajout au panier de réservation.
- Choix des options de paiement via PayPal.
- Confirmation de la réservation par courriel.

f. Paiement via PayPal :

- Intégration du système de paiement en ligne sécurisé PayPal.
- Possibilité pour les élèves de régler les frais de réservation de cours de manière sécurisée.

g. Système de commentaires et d'évaluations :

- Fonctionnalité permettant aux élèves de laisser des commentaires et des évaluations sur les enseignants et les cours suivis.
- Modération des commentaires pour garantir des retours constructifs et respectueux.

Design et expérience utilisateur

- **1.** Conception d'interfaces utilisateur conviviales et intuitives pour les administrateurs, les associations et les clients.
- **2.** Utilisation de principes de design UX pour garantir une expérience utilisateur optimale.

Tests et déploiement

- **1.** Plan de tests exhaustif pour vérifier le bon fonctionnement de la plateforme avant son déploiement.
- **2.** Déploiement sur un serveur sécurisé et fiable.

Maintenance et évolutivité

- **1.** Prévoir des mises à jour régulières pour améliorer les fonctionnalités existantes et ajouter de nouvelles fonctionnalités en fonction des retours des utilisateurs et des besoins émergents.
- **2.** Garantir la maintenance continue de la plateforme pour assurer sa stabilité et sa performance.

Plan Marketing

- Voici un plan marketing adapté pour une plateforme de cours en ligne :
 1. Identification du marché cible :
 - Analyser les groupes démographiques intéressés par l'apprentissage en ligne : cela peut inclure les étudiants universitaires, les professionnels en reconversion, les parents cherchant des ressources éducatives pour leurs enfants, etc.
 - Identifier les personnes sensibles aux causes sociales liées à l'éducation : celles qui soutiennent l'accès à l'éducation pour tous, l'alphabétisation, ou le développement des compétences dans les régions défavorisées.

- Utilisation des réseaux sociaux (Facebook, Instagram, Twitter, etc.) pour promouvoir la plateforme, partager des histoires d'associations et des produits uniques, et interagir avec la communauté.
- Mise en place d'une stratégie de contenu engageant, comprenant des articles de blog, des vidéos et des témoignages d'éducation et de clients.
- **4. Partenariats et collaborations :** Établissement de partenariats avec des influenceurs, des blogueurs et des médias spécialisés dans les produits traditionnels et le soutien aux associations
- **5. Campagnes publicitaires :** Utilisation de publicités ciblées en ligne pour accroître la notoriété de la plateforme et encourager les achats et les dons
- **6. Suivi des performances :** Utilisation d'outils d'analyse pour suivre les performances de la plateforme, mesurer le trafic, les conversions et l'engagement des utilisateurs, et ajuster les stratégies marketing en conséquence.

Technologies

a. Back-end :

- Framework Spring Boot pour le développement de l'application back-end. Spring Boot offre une configuration automatique simplifiée et une intégration transparente avec d'autres frameworks.
- Spring Security pour la gestion de l'authentification, de l'autorisation et la sécurisation des endpoints de l'application.
- Base de données PostgreSQL pour le stockage des données. PostgreSQL est une base de données relationnelle robuste et largement utilisée, offrant des fonctionnalités avancées de gestion des données.
- Utilisation de JPA (Java Persistence API) pour la gestion de la couche d'accès aux données et le mapping objet-relationnel.

b. Front-end :

- Framework Angular pour le développement de l'interface utilisateur côté client. Angular est un framework JavaScript open-source développé par Google, offrant une structure robuste et des fonctionnalités avancées pour la création d'applications web interactives.

□ Utilisation de Bootstrap pour le design et la mise en page responsives. Bootstrap est un framework CSS open-source qui facilite la création d'interfaces utilisateur attrayantes et adaptatives.

□ Utilisation de JavaScript et jQuery pour la manipulation du DOM et l'interaction avec les éléments de l'interface utilisateur.

c. Déploiement :

□ Utilisation d'un serveur d'application tel qu'Apache Tomcat pour le déploiement de l'application back-end.

□ Utilisation de services d'hébergement cloud comme AWS (Amazon Web Services) ou Google Cloud Platform pour le déploiement de l'application front-end et de la base de données PostgreSQL.

□ Utilisation de Docker pour la conteneurisation des applications et la gestion des environnements de développement, de test et de production.

d. Sécurité :

□ Intégration de JWT (JSON Web Tokens) avec Spring Security pour la gestion des sessions et l'authentification basée sur token.

□ Configuration de mécanismes de sécurités avancés au niveau de la couche frontale et de la couche back-end pour prévenir les attaques et protéger les données sensibles.

Contrainte de Durée :

- **1.** Le développement de la plateforme devra être complété avant la date limite du 24 mars 2024.
- **2.** Le délai total de développement est donc de 45 jours à compter de la date de début des travaux de développement.
- **3.** Un planning détaillé sera établi dès le début du projet, décomposant les différentes phases de développement, y compris la conception, la programmation, les tests, la mise en production, et la phase de correction des bugs.
- **4.** Une gestion de projet efficace sera mise en place pour s'assurer que les délais sont respectés et que les objectifs de développement sont atteints dans les temps impartis.

- 5.** Des réunions régulières seront organisées pour suivre l'avancement du projet, identifier les éventuels retards ou obstacles, et prendre les mesures correctives nécessaires pour respecter la date limite du 24 mars 2024