

Cahier des charges du Projet ft_trascendance

Ce projet consiste a créer une application web complet de notre choix.

Notre choix: une application web de support basée sur un système de tickets, permettant aux utilisateurs de signaler des problèmes et de suivre leur traitement.

1. Présentation générale du projet

1.1 Contexte

Dans de nombreuses organisations, la gestion des demandes de support se fait de manière désorganisée (emails, messages privés, discussions orales). Cela entraîne une perte d'informations, des retards de traitement et un manque de visibilité pour les utilisateurs.

1.2 Objectif du projet

Le projet vise à mettre en place une application web de support basée sur un système de tickets, permettant de centraliser les demandes, d'assurer un suivi clair et d'améliorer la communication entre les utilisateurs et l'équipe technique.

1.3 Public cible

- Utilisateurs ayant besoins d'assistance (clients, étudiants, employés)
- Equipe technique (développeurs, agents de supports)
- Administrateurs du systeme (les entreprises)

2. Présentation générale du projet

2.1 Fonctionnalités incluses

- Authentification

- Création et gestion des tickets
- Suivi de l'état des tickets
- Système de commentaires
- Notifications en temps réel
- Tableau de bord et statistiques
- Gestion des utilisateurs et des rôles

2.2 Fonctionnalités exclues

- Paiement en ligne
- Support téléphonique
- Application mobile native

3. Gestion des utilisateurs et des rôles

3.1 Types d'utilisateurs

*** Client:**

- Créer un ticket
- Consulter ses tickets (son état)
- Suivre l'avancement
- Ajouter des commentaires

*** Agent (Développeur ou autre):**

- Consulter tous les tickets
- Modifier le statut des tickets
- Répondre aux tickets
- Prendre en charge un ticket

*** Administrateur:**

- Supprimer les tickets
- Créer et gérer les catégories
- Gérer les utilisateurs et leurs rôles
- Accéder aux statistiques globales

4. Système de tickets

4.1 Création d'un ticket

Un ticket contient:

- Titre
- Description
- Priorité (Basse, Haute, Critique)
- Catégorie

- Date de création
- Auteur

4.2 Cycle de vie d'un ticket

Un ticket suit les états suivants:

1. Ouvert
2. En cours
3. Résolu

Chaque changement d'état est enregistré dans l'historique.

4.3 Assignation

Un ticket peut être assigné à un agent spécifique (par l'admin) pour assurer le traitement.

5. Commentaires et historique

- Chaque ticket dispose d'un fil de discussion dédié
- Les clients et agents peuvent échanger des messages
- Le système conserve un historique des actions importantes (changement de statut, assignation, fermeture)

6. Notifications en temps réel

- Notification instantanée lors de la création d'un ticket
- Notification lors d'une création ou d'un changement de statut
- Indicateur visuel pour les nouveaux messages

7. Tableau de bord et statistiques

Le tableau de bord permet de visualiser:

- Nombre de tickets ouverts, en cours et résolus
- Temps moyen de résolution
- Répartition des tickets par priorité
- Activité globale du support

8. Exigences techniques

8.1 Technologies envisagées

Frontend: React (ou Next.js)

Backend: NestJs

Base de données: PostgreSQL

ORM: Prisma

Librairie css: TailwindCSS

8.2 Compatibilité

- Navigateurs (Chrome, Firefox)
- Interface responsive (ordinateur, tablette, mobile)

8.3 Sécurité

- Authentification sécurisée
- Gestion des droits selon les rôles
- Protection des données utilisateurs

9. Contraintes

- Respect des délais du projet
- Simplicité d'utilisation (l'UX doit être le plus simple possible)
- Performance acceptable même avec une connexion internet limitée
- Interface claire et intuitive

10. Livrables attendus

- Application web fonctionnelle
- Code source du projet
- Documentation utilisateur
- Documentation technique

11. Critères de réussite

- Toutes les fonctionnalités prévues sont opérationnelles
- Les utilisateurs peuvent suivre facilement leurs tickets
- Le système est stable et sécurisé
- Les statistiques sont fiables et lisibles

Planning prévisionnel

Analyse et Conception -> 3 jours

Développement -> 30 jours

Tests et corrections -> 5 jours

Les modules a faire

Web:

- Major: Use a framework for both frontend (Next.js) and backend (NestJs). Monorepo (A rechercher) -> 2 points
- Major: Implement real-time features using WebSockets or similar technology. Firebase message (A rechercher) -> 2 points
- Major: Allow users to interact with other user. -> 2 points
- Minor: Use an ORM for the database (prisma) -> 1 point
- Minor: A complete notification system for all creation, update, and deletion actions. -> 1 point
- Minor: Server-Side Rendering (SSR) for improved performance and SEO. -> 1 point

Accessibility and Internationalization:

- Minor: Support for multiple languages (fr, en, esp) -> 1 point

User Management:

- Minor: Implement remote auth with OAuth (42, Google) -> 1 point
- Major: Advanced permissions system (Role based access, RBA) -> 2 points

Date and Analytics

- Major: Advanced analytics dashboard with data visualisation -> 2 points

Total: 16 points

Note: Ce cahier des charges peut/doit etre modifier selon les besoins de l'equipe. par exemple, on pourra choisir plus de modules pour avoir plus de points, supprimer certains modules jugés pas trop intéressants, etc ...