**4Proj**

Sommaire

**Introduction**

* Résume de Project
* Fonctionnalités principales

**Conception**

* Logo et couleurs
* Diagramme UML
* Charte Graphique web
* Charte graphique mobile

**Architecture Micro-Service**

* Description
* Composants Clés
* Avantage

**Conclusion**

* Conclusion
* **Introduction**

1. **Résume de Project**

Le projet vise à concevoir une **application de navigation en temps réel** pour les usagers de la route en France métropolitaine. Cette application permettra :

* **D’optimiser les déplacements :** grâce à des itinéraires calculés en fonction des conditions de circulation en temps réel (trafic, accidents, routes fermées, etc.).
* **De renforcer l’entraide communautaire :** les utilisateurs pourront signaler des incidents comme des embouteillages, accidents, obstacles, ou contrôles policiers, tout en validant ou infirmant les signalements d'autres utilisateurs.
* **D’informer en temps réel :** des alertes seront envoyées pour prévenir des obstacles à l’approche.
* **De personnaliser les trajets :** avec des options pour éviter les péages ou privilégier certains itinéraires en fonction des préférences de l’utilisateur.
* **De favoriser une interaction mobile-web fluide :** les itinéraires préparés sur l’interface web pourront être partagés facilement sur l’application mobile via un QR code ou un système de synchronisation directe.

Cette solution innovante entend améliorer la sécurité, le confort, et l'efficacité des déplacements quotidiens en s'appuyant sur les contributions de la communauté et sur une gestion intelligente des données de trafic.

1. **Fonctionnalités principales**

**Navigation en temps réel** :

* + Itinéraires optimisés selon le trafic.
  + Recalcul automatique en cas d’incidents.
  + Affichage des trajets sur une carte interactive.

**Signalements d’incidents** :

* Accidents, bouchons, contrôles policiers, routes fermées, obstacles.
* Validation ou infirmation des signalements par la communauté.

**Alertes et notifications** :

* Infos en temps réel sur les conditions de circulation.

**Partage d’itinéraires** :

* Transfert d’itinéraires entre web et mobile via QR code.

**Personnalisation des trajets** :

* Options pour éviter les péages ou choisir l’itinéraire le plus rapide.

Plus

* Calcule le prix de l’itinéraire (prix de péage e prix gasoilea)
* **Conception**

1. **Logo et colleurs**

Image

Motive

Colleurs

Motive

**1. Navigation en temps réel**

* **Leaflet** lui-même ne gère pas la logique d'itinéraire ou le calcul des itinéraires en temps réel. Pour cela, vous pouvez l'intégrer avec des services tiers comme :
  + **OpenRouteService** (basé sur OpenStreetMap) : Offre des itinéraires optimisés avec des options comme éviter les péages, les autoroutes, etc.
  + **Mapbox Directions API** : Permet de calculer des itinéraires avec des options de trafic en temps réel.
  + **Google Maps API** : Fournit également des itinéraires optimisés selon le trafic, mais cela peut être payant.

Ces services vous permettront de calculer l'itinéraire, d'ajuster les trajets en temps réel selon le trafic et les incidents, et d'afficher ces informations sur une carte Leaflet.

**2. Signalements d'incidents**

* Pour afficher les **signalements d'incidents** (accidents, bouchons, routes fermées, etc.), vous pouvez utiliser un service externe ou créer une API qui collecte ces informations en temps réel. Vous pouvez afficher ces incidents sous forme de **marqueurs** sur la carte avec **Leaflet**.
  + Par exemple, des services comme **Waze**, **TomTom**, ou même **OpenStreetMap** peuvent fournir des données de trafic en temps réel, mais souvent, ces informations sont disponibles via des API tierces.
  + Vous pourriez aussi permettre à vos utilisateurs de soumettre des incidents via une interface utilisateur, qui serait ensuite validée par la communauté. Cela pourrait être géré par un backend que vous développez vous-même (par exemple, avec **Node.js** et **MongoDB**).

**3. Alertes et notifications**

* **Leaflet** ne gère pas directement les alertes ou notifications en temps réel. Vous pouvez utiliser un service tiers ou une bibliothèque pour envoyer des notifications aux utilisateurs, telles que :
  + **Firebase Cloud Messaging** (FCM) : Pour envoyer des notifications push sur les appareils mobiles.
  + **WebSockets** : Pour envoyer des notifications en temps réel aux utilisateurs sur une interface web.
  + **Backend** : Implémenter un système de notifications qui se base sur les événements de trafic, tels que des bouchons, des accidents, ou des fermetures de routes.

**4. Partage d'itinéraires via QR Code**

* Pour permettre le partage d'itinéraires, vous pouvez générer un **QR code** contenant l'URL de l'itinéraire. Lorsque l'utilisateur scanne ce code, il est redirigé vers l'itinéraire sur le site web ou l'application mobile.
  + **QRCode.js** est une bibliothèque simple pour générer des QR codes côté client.
  + Vous pouvez également intégrer cela avec votre backend, où vous stockez les itinéraires générés et permettez aux utilisateurs de partager les liens vers ces itinéraires.

**5. Personnalisation des trajets (ex. éviter les péages, choisir l'itinéraire le plus rapide)**

* Pour la **personnalisation des trajets** (ex. éviter les péages ou choisir l'itinéraire le plus rapide), vous pouvez également vous appuyer sur des API externes comme :
  + **OpenRouteService** et **Mapbox Directions** qui offrent des options pour personnaliser les itinéraires (par exemple, éviter les péages, éviter les autoroutes, choisir l'itinéraire le plus rapide).
  + **Google Maps Directions API** fournit des options de personnalisation similaires.