

Cahier des charges pour l'application "InstaLaid"

Objectif du projet :

Développer une application web de type Instagram, dédiée à la publication d'images humoristiques ou esthétiquement "moches". L'objectif est de proposer une plateforme ludique avec un design volontairement incohérent et excentrique, tout en offrant des fonctionnalités similaires à celles des réseaux sociaux classiques.

1. Fonctionnalités principales :

1.1. Publication d'images :

- Permettre aux utilisateurs de publier des images humoristiques ou décalées sur leur profil.
- Les images doivent pouvoir être likées, dislikées, et recevoir une réaction spéciale "VTFF" (Va Te Faire Foutre - 🍑).

1.2. Interactions sur les publications :

- Intégrer des boutons de like (👍), dislike (👎) et VTFF (🍑) sur chaque publication.
- Permettre aux utilisateurs de commenter les publications. Les commentaires seront non modérés mais filtrés automatiquement pour les insultes, grâce à une base de données des insultes de la langue française.

1.3. Modération :

- Les modérateurs doivent valider ou rejeter les publications, assignées de manière aléatoire.
- Intégrer un système simple de validation initiale avec des boutons "go" ou "nogo", à remplacer par une validation par "swipe" lors de la version finale.

1.4. Rôles utilisateurs :

- **Utilisateur** : Publier des images, commenter et interagir avec les publications.
- **Modérateur** : Valider ou rejeter les publications assignées de façon aléatoire.
- **Administrateur** : Gérer les comptes utilisateurs, modifier les rôles et supprimer des comptes ainsi que assigner les rôles. Aucune modération directe des publications.

1.5. Authentification :

- Implémenter un système d'inscription/connexion via Supabase pour gérer les utilisateurs.

1.6. Mode hors-ligne :

- Permettre la consultation hors-ligne des 20 dernières publications visualisées par l'utilisateur, ou des 20 dernières publications postées par celui-ci.

2. Spécifications techniques :

2.1. Technologies utilisées :

- **Framework** : Utilisation de Next.js avec des templates prédéfinis pour accélérer le développement, si existants.
- **Base de données** : Supabase pour stocker les utilisateurs, les rôles, les publications et les commentaires.
- **Graphismes** : Utilisation de shadcn/ui pour les composants graphiques.
- **Progressive Web App (PWA)** : L'application doit être installable sur mobile et fonctionner en mode hors-ligne (20 dernières publications).
- **Hosting** : L'application sera hébergée via GitHub Pages.

2.2. Base de données :

- Supabase sera utilisé pour gérer les utilisateurs, les rôles, les publications et les interactions.
- Prisma ORM sera utilisé pour une meilleure gestion des données dans la base de données.

2.3. Sécurité :

- La sécurisation de la base de données n'est pas requise. La gestion des droits d'accès et de propriété des données sera assurée côté application.
- Toutes les données seront accessibles par l'ensemble des utilisateurs, mais seulement les administrateurs y auront réellement accès.

3. Design et UX :

3.1. Design global :

- Adopter une esthétique volontairement décalée. L'esthétique doit inclure des couleurs et des styles délibérément mal assortis (ex. vomit, orange sanguine).
- Chaque élément de design doit être beau en soi, mais l'ensemble doit être volontairement incohérent.

3.2. Mobile-first :

- Optimiser l'interface pour les appareils mobiles, avec une expérience desktop secondaire.

3.3. Expérience utilisateur (UX) :

- Malgré l'aspect décalé, l'application doit rester intuitive et simple à utiliser.

4. Livrables attendus :

4.1. Code source et déploiement :

- Repository GitHub contenant tout le code source du projet, avec une gestion versionnée.
- GitHub Pages pour le déploiement de l'application.
- Documentation du projet (ReadMe, guide d'installation et d'utilisation).

4.2. Présentation finale :

- Réaliser une présentation de 10 minutes incluant une démonstration des fonctionnalités principales.
- Inclure une matrice RACI pour détailler les responsabilités de l'équipe.
- Fournir un PowerPoint expliquant les choix techniques et fonctionnels.

5. Équipe de développement :

DENANS Dimitri, ILIOT Thomas, LAURENT Louis et SILVA SÁ Bruno.