

Cahier des Charges

CHEIKH MBACKE

CANDIDATURE CHEZ AZUR TECH RESEARCH 05/11/2024



SOMMAIRE

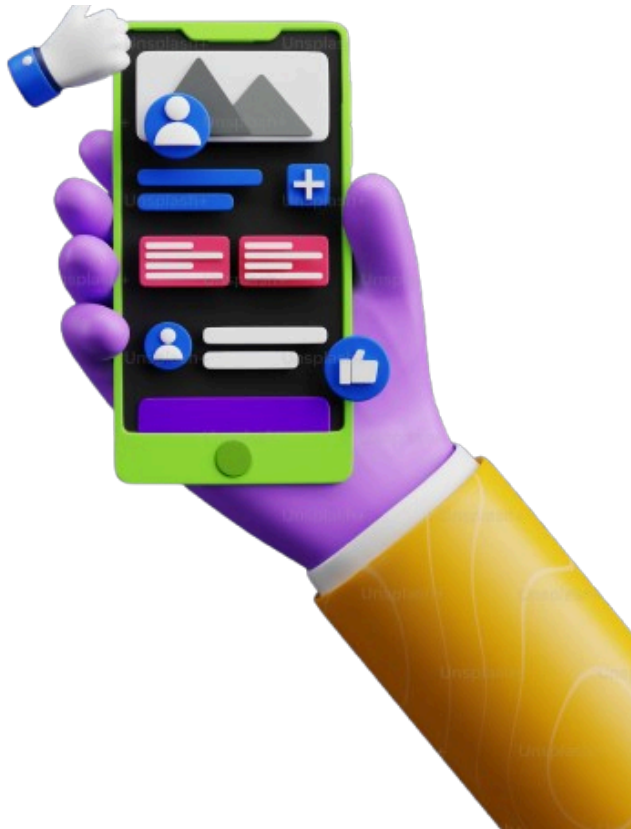
Introduction	1
Organisation du Code et des Fichiers	2
Structure et Fonctionnalités de l'Application	3
Bonnes Pratiques	4



1. Introduction

Ce projet consiste à créer un mini réseau social en local, sans dépendance vers une API externe. Il a pour but d'évaluer **les compétences** en **HTML**, **CSS** et **JavaScript** pur, dans le cadre d'une candidature chez **AZUR TECH RESEARCH**.

L'application devra suivre un style visuel en "**neumorphisme**" tout en restant entièrement responsive, c'est-à-dire capable de s'adapter aux différentes tailles d'écran (desktop, tablette, mobile).



2. Organisation du Code et des Fichiers

L'utilisation de **frameworks** est interdite : tout le projet doit être organisé en HTML, CSS et JavaScript pur.

La structure des fichiers devra être soigneusement pensée pour assurer une bonne lisibilité et une maintenance aisée, tout en respectant les conventions de nommage.

La page de liste d'amis, en particulier, devra inclure une redondance dans les classes CSS, ce qui permettra de juger la gestion des styles sans frameworks.



3. Structure et Fonctionnalités de l'Application

Le mini réseau social comprendra trois pages principales : un feed, une messagerie, et une liste d'amis.

Page de Feed

La page de **feed** affichera une liste de posts dans un format standardisé (auteur, texte, image éventuelle, réactions et commentaires).

Pour chaque post, l'utilisateur pourra réagir en ajoutant un like, un dislike ou un love, chaque réaction déclenchant une animation de particules spécifique pour rendre l'interface plus dynamique.

Les utilisateurs auront également la possibilité de commenter les posts, mais aussi d'ajouter des réponses aux commentaires existants. Enfin, si une photo est incluse dans un post, elle pourra être visualisée en plein écran en cliquant dessus.

Page de Messagerie

La **messagerie** affichera une liste de conversations, dont les données seront aussi extraites d'un fichier JSON. La liste affichera les conversations avec le dernier message de chaque contact.

En cliquant sur une conversation, l'utilisateur pourra accéder à l'historique complet des messages, avec les informations de chaque message comme le nom de l'expéditeur, la photo de profil, le contenu, et un horodatage.

Il sera également possible pour l'utilisateur d'envoyer de nouveaux messages, qui seront ajoutés à la fois au JSON et à l'interface, offrant ainsi une gestion dynamique des conversations.

Page de Liste d'Amis

La **liste d'amis** sera une page plus simple, avec des amis codés en dur dans le HTML (pas de JSON requis ici).

Les utilisateurs pourront filtrer les amis par nom ou prénom et auront accès à un lien direct pour ouvrir une conversation avec chaque ami.

De plus, un système de **drag-and-drop** permettra de réorganiser la liste, rendant la page plus interactive.

5. Bonnes pratiques

Gestion du Code avec Git

La gestion des versions se fera via Git, avec des commits fréquents et bien documentés pour chaque étape ou fonctionnalité majeure. L'objectif est de permettre un suivi clair de l'évolution du projet, grâce à des titres et descriptions de commits détaillés et explicites. L'arborescence des branches devra suivre une logique cohérente pour chaque fonctionnalité ou page.



Documentation et Commentaires

La documentation devra inclure un fichier README avec une description détaillée du projet, les instructions pour le lancer en local, et une présentation des fonctionnalités principales. Ce README expliquera également la structure des fichiers pour faciliter la prise en main. Le code, quant à lui, devra être commenté aux endroits clés pour rendre la logique de chaque fonctionnalité compréhensible et intuitive.



Critères Techniques de Validation

Enfin, en matière de validation, plusieurs critères techniques seront pris en compte. Tout d'abord, le site devra être pleinement responsive pour une expérience utilisateur cohérente sur les différents appareils. Ensuite, le design en neumorphisme devra être respecté dans l'interface utilisateur. La gestion des fichiers CSS, en particulier sur la redondance des classes dans la page d'amis, sera un élément d'évaluation. Pour finir, la manipulation dynamique du DOM et la gestion des données JSON devront être bien structurées, permettant un affichage fluide et efficace des données dans le feed et la messagerie.

