# **CHAPITRE 3: PRESENTATION DE LA SOLUTION**

Dans ce chapitre 3, nous allons présenter la solution que nous avons proposée pour résoudre le problème posé. Nous détaillerons les différentes interfaces de la solution. Avant cela, nous introduirons les outils utilisés.

* 1. **PRESENTATION DES LOGICIELS, LANGAGE DE PROGRAMMATION**

### **I.1.** **OUTILS UTILISÉ**

Afin de mener à bien notre travail, nous avons utilisé plusieurs outils qui nous ont assistés dans la modélisation, la programmation

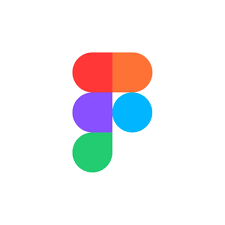


* **Visual Studio Code (VS Code) :**

VS Code est un éditeur de code source polyvalent utilisé pour écrire, modifier et déboguer du code. Il offre une multitude de fonctionnalités et d'extensions pour faciliter le développement logiciel dans différents langages de programmation. VS Code est largement utilisé par les développeurs en raison de sa légèreté, de sa grande flexibilité et de sa riche sélection d'extensions. Il facilite la productivité des développeurs en offrant des fonctionnalités telles que la complétion automatique du code, le débogage intégré, le contrôle de version Git et l'intégration avec d'autres outils de développement.

* **Draw.io :**

Draw.io est un outil de modélisation graphique utilisé pour créer des diagrammes UML, des organigrammes, des schémas de réseau et d'autres types de diagrammes. Il permet de concevoir visuellement l'architecture et les composants du système. Draw.io est particulièrement utile pour représenter les structures de données, les flux de travail et les interactions entre les différents éléments du système. Il aide à clarifier et à communiquer les concepts de conception avec l'équipe de développement.

* **Figma :**

### **Figma** est un outil de **conception d’interface utilisateur (UI)** et de **prototypage collaboratif**. Il permet aux designers, développeurs et autres parties prenantes de créer des maquettes interactives directement depuis un navigateur, sans nécessiter d’installation locale.

Figma facilite la **création, la modification et le partage** de maquettes graphiques, tout en intégrant des fonctionnalités de **collaboration en temps réel**. Grâce à une interface intuitive et des outils puissants de design vectoriel, il permet de concevoir des interfaces modernes et réactives pour les applications web et mobiles.

### **I.2. TECHNOLOGIES UTILISÉS**



* **Angular**

**Angular** est un framework JavaScript robuste utilisé pour structurer l’application et gérer les composants. Grâce à son architecture basée sur les composants, Angular permet de développer des interfaces utilisateur dynamiques, modulaires et faciles à maintenir. Il facilite également la gestion des routes, des formulaires et de la communication avec les services backend, tout en assurant une organisation claire du code.



* **Firebase**

**Firebase** a été choisi comme solution de backend pour sa base de données en temps réel et son système d’authentification intégré. Grâce à Firebase, l’application bénéficie d’une gestion instantanée des données (synchronisation en temps réel entre utilisateurs) et d’un système d’authentification sécurisé, simplifiant la gestion des comptes utilisateurs et des accès.



* **Capacitor**

**Capacitor** est un outil qui permet de transformer l’application web Angular en application mobile native. Il facilite l’accès aux fonctionnalités des appareils mobiles (caméra, GPS, stockage local, etc.) et permet de déployer l’application sur Android et iOS, tout en conservant une base de code unique.



* **TailwindCSS**

**TailwindCSS** est un framework CSS utilitaire utilisé pour le design et la mise en page responsive. Il permet de concevoir rapidement des interfaces modernes et adaptatives, en appliquant des classes utilitaires directement dans le code HTML. Cela accélère le développement, garantit la cohérence visuelle et facilite la personnalisation du design.



* **HTML et SCSS**

**HTML** sert à structurer les différentes pages et composants de l’interface utilisateur, tandis que **SCSS** (Sass) permet d’écrire des styles CSS avancés, plus organisés et réutilisables. L’utilisation de SCSS améliore la gestion des styles grâce à la possibilité d’utiliser des variables, des mixins et des imbrications.



* **TypeScript**

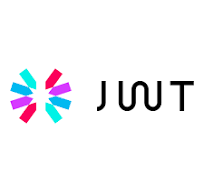
**TypeScript** est un sur-ensemble de JavaScript qui ajoute le typage statique. Son utilisation dans le projet permet de sécuriser le code, de détecter les erreurs à la compilation, d’améliorer l’autocomplétion et de faciliter la maintenance du projet, notamment dans une application Angular où la structure des données est complexe.

Pour assurer la performance, la modularité et la robustesse du projet Wandawork, plusieurs technologies ont été choisies, couvrant à la fois le backend, l’authentification, la gestion des données, l’asynchrone, le stockage, le déploiement et l’intégration continue.



* **Backend: Django 5 + Django REST Framework (DRF)**

Le backend est développé avec **Django 5**, un framework Python puissant et mature, qui facilite la création d’applications web sécurisées et évolutives.  
**Django REST Framework (DRF)** est utilisé pour construire une API RESTful robuste, permettant une communication efficace entre le frontend et le backend.



* **Authentification: JWT + OTP (email)**

Pour sécuriser l’accès des utilisateurs, le projet utilises une authentification basée sur **JWT (JSON Web Tokens)**, garantissant des sessions stateless et sécurisées.  
Un système **OTP (One-Time Password) par email** est intégré pour renforcer la sécurité lors de la connexion ou des actions sensibles.



* **Base de données: PostgreSQL**

La base de données relationnelle choisie est **PostgreSQL**, reconnue pour sa fiabilité, ses performances et ses fonctionnalités avancées (gestion des transactions, intégrité des données, extensibilité).



* **Stockage des fichiers : Azure CDN**

Les fichiers statiques et médias sont stockés et distribués via **Azure CDN**, un réseau de diffusion de contenu performant qui améliore la rapidité d’accès aux ressources pour les utilisateurs partout dans le monde.



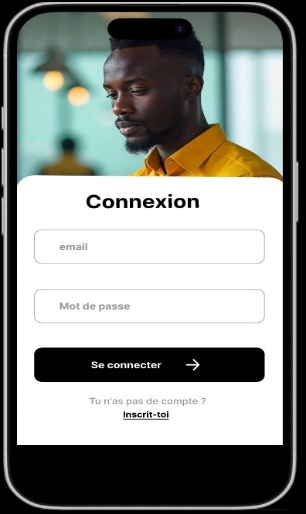
* **Déploiement : Docker + Nginx + Gunicorn**

Le projet est conteneurisé avec **Docker**, assurant portabilité et cohérence entre les environnements de développement et de production.  
**Nginx** est utilisé comme serveur web et reverse proxy, optimisant la gestion des requêtes HTTP.  
**Gunicorn** est le serveur d’application WSGI qui exécute le code Django de manière performante.

* 1. **PRESENTATION DES INTERFACES**

Notre solution wandawork possède une version web et mobile, mais ici nous présenterons les différentes interfaces de la version mobile de notre application.

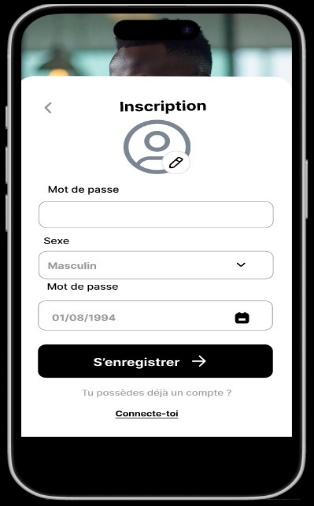
### **Interface de connexion**



Sur cette interface mobile, les utilisateurs possédant déjà un compte peuvent se connecter directement en utilisant leur identifiant utilisateur et leur mot de passe. Les utilisateurs n'ayant pas encore de compte peuvent également accéder à cette interface pour naviguer vers l'écran de création de compte.

**Figure 1 : interface de connexion**

### **Interface de creation de compte**

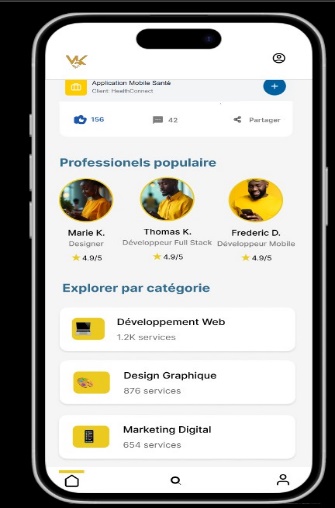
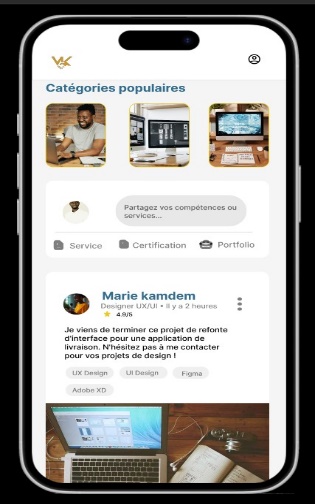


Sur cette interface mobile, les utilisateurs peuvent créer leurs comptes en toute sécurité. Les utilisateurs ayant déjà un compte peuvent également accéder à cette interface pour naviguer vers l'écran de connexion.

**Figure 2: interface de création de compte**

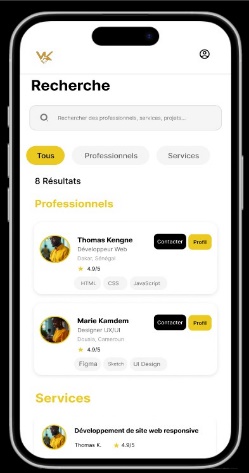
### **Interface principale ou page d’accueil**

Sur cette interface mobile, les utilisateurs peuvent visualiser les les prestataires ayant eu de bonne notations (les plus populaires), les différentes catégories de professionnels et les différentes annonces des prestataires par catégories.

****

**Figure 3 : interface principale ou page d’accueil**

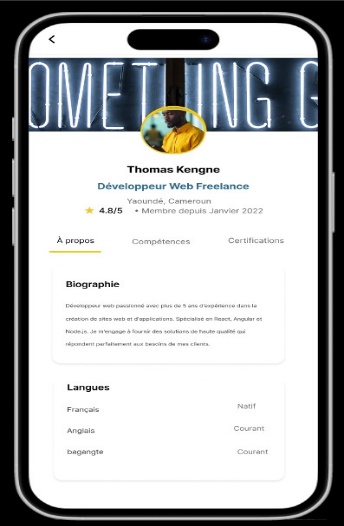
### **Interface de visualisation des recherches par categories**

****

Sur cette interface mobile, les utilisateurs peuvent visualiser le résultat de recherche de professionnels disponibles sur notre plateforme.

**Figure 4 : interface visualisation des categories et mention spéciale**

### **Interface de presentation du profil d’un prestataire**

****

Sur cette interface mobile, on peut voir la biographie de ce prestataire, les différentes langues qu’il parles, ses compétences et même c’est certifications.

### **Figure 5 : interface de presentation du profil d’un prestataire**