



**SOLICODE**

# **RAPPORT PFE**

**Gestion des services**

---

**Réalisé par**  
SOUKAYNA MACHRAA

**Encadrer par**  
IMANE BOUZIANE



## REMERCIEMENTS

Nous tenons avant tout à exprimer notre profonde gratitude envers Dieu Tout-Puissant de nous avoir accordé la santé et la volonté nécessaires pour entreprendre et mener à bien ce projet.

Nous souhaitons adresser nos sincères remerciements à toute l'équipe pédagogique de SolidCode pour leur encadrement précieux et leur dévouement.

Un merci tout particulier à ma famille et à mes proches pour leur amour inconditionnel, leur soutien constant et leurs encouragements indéfectibles tout au long de mes études. Leur confiance en moi a été une source d'inspiration et de motivation essentielle pour l'atteinte de mes objectifs académiques.

Je suis également reconnaissant(e) envers mes camarades de classe et mes ami(e)s qui ont partagé ce parcours scolaire avec moi. Nos échanges, notre soutien mutuel et nos moments de collaboration ont enrichi mon expérience et ont rendu cette étape inoubliable.

Nos remerciements s'adressent spécifiquement à Madame Imane Bouzianne pour ses précieuses remarques et son accompagnement.

Enfin, nous remercions chaleureusement tous nos professeurs pour leur générosité, leur patience remarquable et leur engagement, malgré leurs lourdes charges académiques et professionnelles.



## RÉSUMÉ

Ce rapport présente le développement d'une plateforme web innovante dédiée à la mise en relation de particuliers et de professionnels pour des services à domicile variés. L'objectif principal est de simplifier et de sécuriser la recherche et l'offre de services tels que le jardinage, la plomberie, l'électricité, et d'autres prestations personnalisées. La plateforme offre aux utilisateurs la possibilité de publier des demandes de service détaillées et aux prestataires de présenter leurs compétences et leurs disponibilités. Nous détaillons l'analyse des besoins, la conception de l'architecture fonctionnelle et technique, les choix technologiques adoptés, le processus de développement front-end et back-end, ainsi que les stratégies de test et de déploiement. Ce projet vise à offrir une solution ergonomique et efficace, améliorant l'accès aux services de proximité et facilitant la gestion des activités pour les professionnels indépendants.



# TABLE DES MATIÈRES



# 1 - Introduction générale

Dans un monde de plus en plus connecté, la recherche de prestataires de services fiables est devenue une nécessité quotidienne. Que ce soit pour des travaux de plomberie, de jardinage, d'électricité ou d'autres besoins domestiques, les utilisateurs souhaitent accéder rapidement à des professionnels compétents près de chez eux.

C'est dans ce contexte que s'inscrit mon projet de site web dynamique "Souka.ma", une plateforme en ligne dédiée à la mise en relation entre les prestataires de services et les clients.

L'objectif principal de ce site est double :

Pour les utilisateurs : leur permettre de consulter une liste de services proposés dans différentes catégories (jardinier, plombier, mécanicien, etc.), visualiser les détails de chaque prestataire, et envoyer une demande de service via un formulaire simple et rapide.

Pour les prestataires : leur offrir la possibilité de créer un compte et d'ajouter leur propre service, avec description, image, localisation, et autres informations utiles pour se faire connaître et attirer des clients.



## 1.1 Contexte du Projet et Objectifs Pédagogiques

Dans le cadre de notre formation au sein de SolidCode, nous avons été amenés à réaliser un projet de fin d'études. Ce projet est une opportunité cruciale d'appliquer et d'évaluer les compétences techniques et méthodologiques acquises durant notre parcours de formation. Il vise à nous confronter à un cas concret de développement, de la phase d'analyse des besoins à celle du déploiement, en passant par la conception et la réalisation. Ce travail représente la synthèse de nos acquis et est indispensable pour l'obtention de notre diplôme.

## 1.2 Objectifs généraux

L'objectif principal de ce projet est de créer une plateforme web robuste et conviviale qui connecte directement les particuliers avec des professionnels qualifiés pour une variété de services (jardinage, plomberie, etc.). Nous visons à :

- Faciliter et sécuriser la recherche et l'offre de services à domicile.
- Augmenter la visibilité des professionnels indépendants.
- Simplifier la gestion des demandes de service pour toutes les parties.
- Améliorer globalement l'expérience utilisateur grâce à une interface intuitive

### 1.3 Périmètre du projet

Le périmètre de ce projet inclut le développement d'une application web complète avec des fonctionnalités essentielles telles que :

- Un système d'inscription et de connexion pour clients et prestataires.
- La création et la gestion de profils professionnels détaillés.
- Une catégorisation des services offerts et la possibilité de demandes personnalisées.
- Des outils de recherche et de filtrage avancés.
- Un système de gestion des demandes de service et une messagerie interne.
- Un mécanisme d'avis et de notations pour évaluer les services. Des fonctionnalités plus avancées comme l'intégration de paiement direct ou la géolocalisation en temps réel sont envisagées pour des phases ultérieures. Le déploiement est également une partie du périmètre du projet

### 1.4 Démarche suivie

Nous avons adopté une démarche de développement itérative et incrémentale, inspirée des méthodes agiles, afin d'assurer flexibilité et réactivité. La démarche s'articule autour des phases clés suivantes :

1. Analyse et Conception : Étude des besoins et modélisation du système.
  2. Réalisation : Développement front-end et back-end de l'application en utilisant des technologies pertinentes.
  3. Tests et Validation : Vérification de la qualité et de la conformité de la solution.
  4. Déploiement : Mise en ligne de la plateforme pour les utilisateurs.
- Cette approche nous permet de livrer une solution de haute qualité répondant aux attentes du marché.

## 2 Analyse et conception

L'objectif de cette section est de comprendre le paysage actuel des plateformes de services à domicile et d'identifier les besoins spécifiques des utilisateurs (prestataires de services et clients).

### 2.1 Étude de l'existant / Analyse du besoin

#### **Étude de l'existant :**

Analyse de plateformes existantes de mise en relation de services (ex: Imaalm , AlloVoisins, Helping, etc.) afin d'identifier leurs forces, leurs faiblesses, leurs fonctionnalités clés et leurs modèles économiques.

Identification des lacunes ou des opportunités non couvertes par les solutions actuelles sur le marché cible.

Analyse des retours d'utilisateurs et des avis sur ces plateformes pour cerner les attentes et les frustrations.

#### **Analyse du besoin :**

Pour les prestataires de services (jardinier, plombier, etc.) :

- Besoin d'une plateforme facile d'utilisation pour s'inscrire, créer un profil, présenter leurs services (description, tarifs, portfolio) et gérer leurs disponibilités.
- Nécessité de recevoir des demandes de service ciblées et de pouvoir y répondre efficacement.
- Accès à un système de gestion des rendez-vous et de suivi des interventions.
- Possibilité de collecter des avis et des évaluations pour renforcer leur réputation.
- Un moyen de gérer les paiements et la facturation.



Pour les clients (demandeurs de service) :

- Besoin de trouver rapidement et facilement des professionnels qualifiés pour des services variés (jardinage, plomberie, électricité, ménage, etc.).
- Possibilité de comparer les profils des prestataires, leurs compétences, leurs tarifs et les avis d'autres utilisateurs.
- Un processus simple pour demander un service spécifique, décrire leurs besoins et fixer un rendez-vous.
- Accès à un suivi de leur demande et à une communication aisée avec le prestataire.
- Un système d'évaluation post-service pour partager leur expérience.
- La possibilité de soumettre une demande pour un service non listé, ouvrant ainsi la porte à une flexibilité accrue.


## 2.2 Cahier des charges

Le cahier des charges formalise l'ensemble des exigences fonctionnelles et non fonctionnelles du projet

Votre texte de paragraphe Fonctionnalités principales :

- Gestion des utilisateurs (Particuliers et Professionnels) : Inscription et connexion.
- Gestion des profils utilisateurs et modification des informations personnelles et professionnelles.
- Réinitialisation de mot de passe.

### **Fonctionnalités principales :**

- 
- Création et gestion de profils professionnels (description des services, tarifs, zone d'intervention, portfolio photos/vidéos).
  - Upload de documents justificatifs (licences, assurances) avec validation par l'administrateur.
  - Gestion des catégories de services proposées.
  - Gestion des disponibilités et du calendrier (bloquer créneaux, définir heures d'ouverture/fermeture).
  - Recevoir et gérer les demandes de services (notifications, accepter/refuser, proposer nouvelle date/heure, envoyer devis).
  - Suivre les paiements et revenus (tableau de bord financier).
  - Accéder aux avis clients et y répondre (modéré par l'administrateur).

### **Gestion des services :**

- Catégorisation des services (plomberie, jardinage, etc.).
- Déclaration détaillée du service, tarif, localisation des services, et gestion des créneaux de disponibilité par les professionnels

### **Gestion des demandes de service :**

- Recherche de services par catégorie, localisation, date et heure souhaitée.
- Possibilité d'effectuer une demande de réservation.
- Suivi du statut des réservations (en attente, acceptée, refusée, terminée) pour les particuliers

### **Messagerie interne :**

Communication entre clients et prestataires (implicite via gestion des demandes et réponses aux avis).

### **Système de recherche et de filtrage :**

Recherche par catégorie, localisation, disponibilité et tarifs.

### **Gestion des avis et notations :**

- Consultation des notes et commentaires laissés par les clients.
- Examen et validation des avis par l'administrateur avant publication

### **Tableau de bord :**

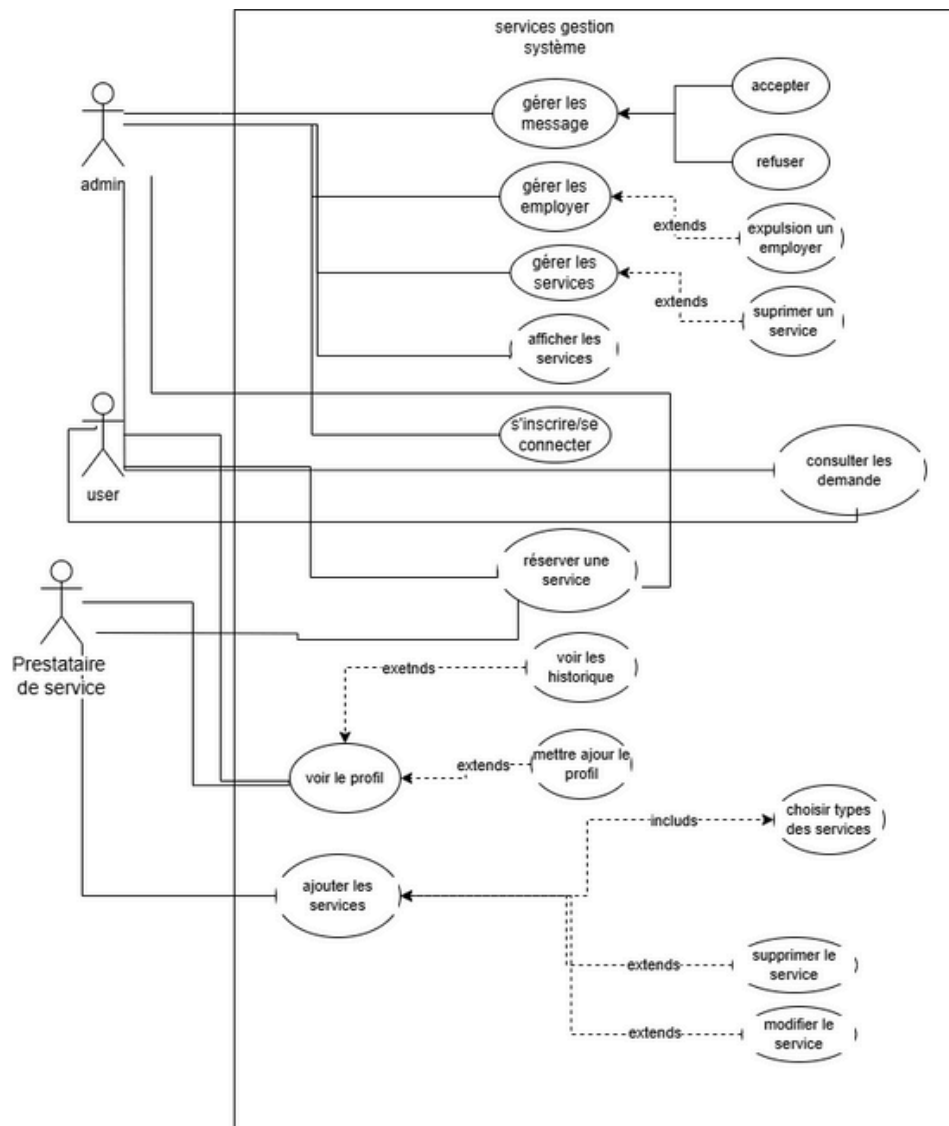
- Tableau de bord financier pour les prestataires.
- Tableau de bord (Dashboard) sur l'interface Back office pour les administrateurs avec des graphes et indicateurs clés.

### **Gestion Back-office (interface administrateur) :**

- Gestion des profils utilisateurs (création, modification, suppression, rôles et permissions).
- Validation des inscriptions et profils prestataires (vérification documents, activation/désactivation de comptes).
- Gestion des services et catégories (ajout, modification, suppression, attributs spécifiques).
- Gestion des demandes et rendez-vous (supervision, intervention en cas de litige).
- Modération des avis et commentaires, gestion des signalements d'abus.
- Gestion des tarifs et commissions (paramétrage global, taux de commission).
- Traçabilité (historique des actions).
- Gestion de contenu de la page d'accueil et informations (pages statiques, nouveautés, promotions).

## 2.3 Modélisation fonctionnelle (Diagrammes de cas d'utilisation)

Les diagrammes de cas d'utilisation permettent de visualiser les interactions entre les acteurs du système et les fonctionnalités offertes par la plateforme.



### Acteurs :

- Particulier (Client) : Utilisateur qui recherche et demande des services.
- Prestataire de services : Utilisateur qui propose et réalise des services.
- Administrateur : Gère la plateforme (validation des profils, gestion des catégories, etc.).

### Exemples de Cas d'utilisation (basé sur le diagramme fourni) :

- Client (User) :S'inscrire/Se connecter.
- Consulter les demandes.
- Réserver un service.
- Voir l'historique (étend de Réserver une service).
- Voir le profil.

### Prestataire de service :S

- 'inscrire/Se connecter.
- Réserver une service.
- Voir l'historique (étend de Réserver une service).
- Voir le profil.
- Mettre à jour le profil (étend de Voir le profil).
- Ajouter les services.
- Supprimer le service (étend de Ajouter les services).
- Modifier le service (étend de Ajouter les services).
- Choisir types des services (inclus dans Mettre à jour le profil).

### Admin :Gérer les messages.

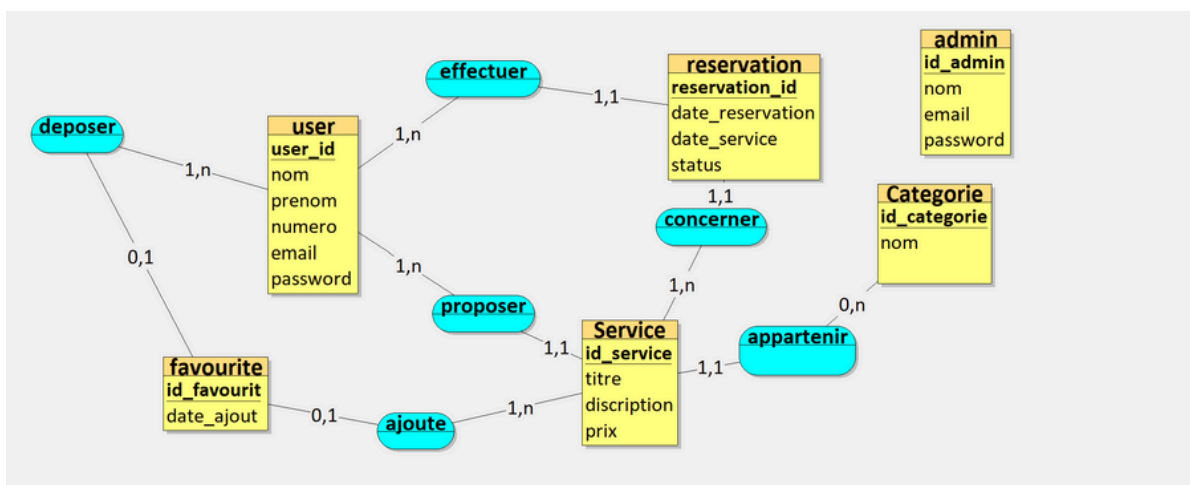
- Accepter (étend de Gérer les messages).
- Refuser (étend de Gérer les messages).
- Gérer les employés.
- Expulser un employé (étend de Gérer les employés).
- Gérer les services.
- Supprimer un service (étend de Gérer les services).
- Afficher les services.

## 2.4 Modélisation des données (MCD / MLD / MPD)

### Modèle Conceptuel de Données (MCD) :

La modélisation des données est cruciale pour structurer la base de données de manière efficace.

Modèle Conceptuel de Données (MCD) : Le MCD représente les entités principales du système et les relations entre elles, indépendamment de toute considération technique ou de la future base de données. Il permet de définir clairement les informations à stocker et la manière dont elles sont liées.

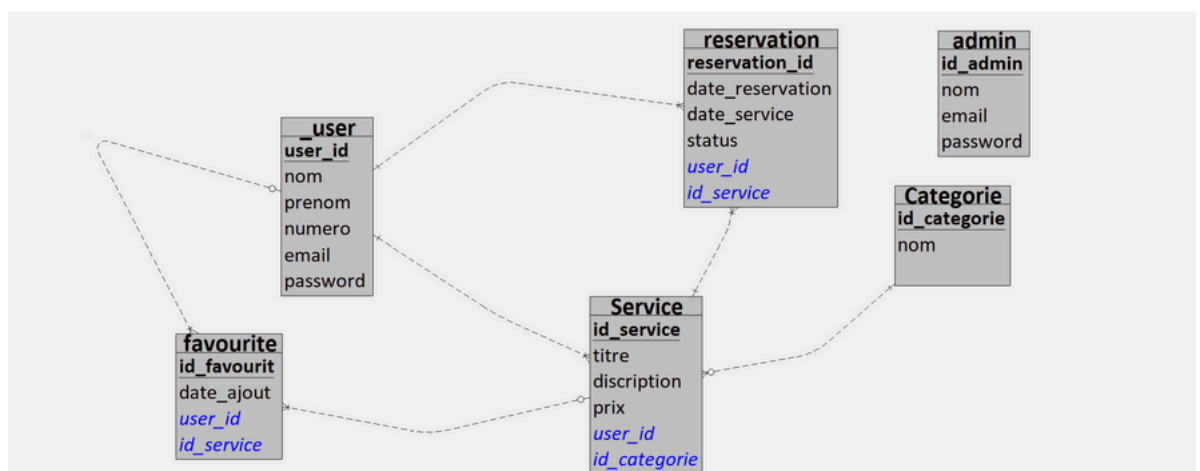


- Entités :user (Utilisateur) : Représente les utilisateurs de la plateforme, qu'ils soient particuliers ou prestataires. Il contient les attributs user\_id (identifiant unique), nom, prenom, numero, email, et password.
- reservation : Enregistre les détails de chaque réservation. Attributs : reservation\_id, date\_reservation, date\_service, et status.
- Service : Décrit les services spécifiques offerts. Attributs : id\_service, titre, et discription.
- admin : Représente les administrateurs de la plateforme. Attributs : id\_admin, nom, email, et password.
- Categorie : Catégorise les différents types de services. Attributs : id\_categorie et nom.
- pestataire (Prestataire) : Représente les professionnels offrant des services. Il contient l'attribut id\_pestataire.

- Relations :posseder : Une relation (1,1) entre user et prestataire, indiquant qu'un utilisateur peut être un prestataire. Cette relation semble suggérer qu'un prestataire est un type d'utilisateur.
- effectuer : Une relation (1,n) entre user et reservation, indiquant qu'un utilisateur peut effectuer plusieurs réservations.
- concerner : Une relation (1,1) entre reservation et Service, indiquant qu'une réservation concerne un seul service.
- appartenir : Une relation (0,n) entre Service et Catégorie, indiquant qu'un service appartient à une catégorie et une catégorie peut avoir plusieurs services.
- fournir : Une relation (1,1) entre prestataire et Service, indiquant qu'un prestataire fournit un service. Il semble que cette relation devrait être (1,n) car un prestataire peut fournir plusieurs types de services.
- L'administrateur (admin) n'a pas de relation directe représentée avec les autres entités, ce qui est typique pour les entités de gestion qui interagissent avec le système globalement plutôt qu'avec des instances spécifiques.

## Modèle Logique de Données (MLD) :

Le MLD est une représentation des données structurée en tables, colonnes, clés primaires et clés étrangères, qui est directement mappable à un système de gestion de base de données relationnelle. Il dérive du MCD en traduisant les entités en tables et les relations en clés étrangères.



## 3 Réalisation

Cette section détaille les choix techniques effectués pour le développement de la plateforme ainsi que les aspects liés au développement des interfaces utilisateur (frontend) et de la logique métier (backend).

### 3.1 Choix techniques

Les choix techniques ont été guidés par des considérations de performance, de scalabilité, de maintenabilité, de sécurité, et de la rapidité de développement, tout en s'alignant sur les compétences de l'équipe et les tendances du marché.

Les choix techniques ont été guidés par des considérations de performance, de scalabilité, de maintenabilité, de sécurité, et de la rapidité de développement, tout en s'alignant sur les compétences de l'équipe et les tendances du marché.

- Langages de programmation :
  - Backend : PHP pour sa flexibilité, sa large adoption dans le développement web, et sa capacité à interagir efficacement avec les bases de données.
  - Frontend : HTML5, CSS3, et JavaScript pour la structure, le style et l'interactivité des pages web.
- Frameworks et bibliothèques :
  - Frontend : L'utilisation de JavaScript pur ou d'une bibliothèque légère comme jQuery peut être envisagée pour la gestion des interactions dynamiques et des requêtes asynchrones, complétant HTML et CSS pour une interface interactive. Un framework CSS comme Bootstrap sera utilisé pour assurer le responsive design et accélérer le stylisme.
  - Backend : Un framework PHP tel que Laravel ou Symfony sera utilisé pour structurer l'application, faciliter le développement d'APIs RESTful et gérer l'interaction avec la base de données de manière sécurisée et efficace.



- Base de données :

MySQL : Ce système de gestion de base de données relationnelle (SGBDR) est choisi pour sa robustesse, sa performance, sa facilité d'utilisation et sa compatibilité étendue avec PHP. Il est idéal pour gérer les volumes de données structurées de la plateforme.

- Gestion de version :

Git avec un dépôt distant sur GitHub pour le contrôle de version du code source, facilitant la collaboration de l'équipe et le suivi des modifications.

- Environnement de développement :

Docker peut être utilisé pour la conteneurisation des différentes parties de l'application (base de données, backend, frontend), assurant un environnement de développement cohérent et facilitant le déploiement.

## 2.3 Développement frontend

Le développement frontend concerne la création de l'interface utilisateur et de l'expérience interactive pour les clients, les prestataires et les administrateurs. L'objectif est d'assurer une interface graphique conviviale et responsive.

### **Architecture :**

Le projet repose sur une architecture de type MPA (Multi-Page Application), avec une séparation claire entre la logique, la présentation et les données. Le développement s'appuie sur les technologies suivantes : HTML, CSS, JavaScript, PHP et MySQL

## Composants de l'interface utilisateur :

- Pages d'accueil : Présentation des services, moteur de recherche principal, mise en avant des prestataires populaires.
- Pages d'inscription et de connexion : Formulaires sécurisés pour les différents types d'utilisateurs.
- Profils utilisateurs (clients et prestataires) : Affichage et modification des informations personnelles, des services proposés, des portfolios, et des avis.
- Pages de recherche de services : Interface avec filtres (catégorie, localisation, date, tarif) et affichage des résultats.
- Formulaires de demande de service/réservation : Saisie détaillée des besoins du client.

## Tableaux de bord :

- Client : Suivi des demandes, historique des réservations, gestion des avis.
- Prestataire : Gestion des demandes, calendrier des disponibilités, suivi des revenus, gestion du profil et des services.
- Système de messagerie : Interface pour la communication entre clients et prestataires.
- Pages d'administration (Back-office) : Interfaces pour la gestion des utilisateurs, des services, des catégories, la modération des avis, etc..

**Responsive Design :** Utilisation de CSS Media Queries et d'un framework CSS (ex: Bootstrap ou Tailwind CSS) pour garantir une adaptation parfaite de l'interface à toutes les résolutions d'écran (ordinateurs, smartphones, tablettes).

**Interactivité :** Mise en œuvre de JavaScript pour les validations de formulaires en temps réel, les chargements asynchrones de données, et les animations fluides, améliorant l'expérience utilisateur.

### 3.3 Développement backend

Le développement backend, réalisé avec PHP et SQL, sur un environnement XAMPP, est le cœur de la logique métier de la plateforme. Il gère la persistance des données, les traitements complexes, la sécurité et la communication avec le frontend via des APIs.

#### **Environnement de Développement : XAMPP**

XAMPP est utilisé comme environnement de développement local, intégrant Apache (serveur web), MySQL (base de données SQL), PHP (langage de script serveur) et Perl. Cette suite logicielle facilite la configuration rapide d'un environnement de développement web complet et portable.

#### **Architecture :**

Mise en place d'une architecture orientée API RESTful pour permettre une communication standardisée et efficace entre le frontend (HTML, CSS, JavaScript) et le backend (PHP).

#### **Gestion des utilisateurs et authentification :**

Inscription et Connexion : Développement des scripts PHP pour gérer l'inscription des utilisateurs (particuliers et prestataires) et leur connexion sécurisée. Cela inclut la validation des données d'entrée, le hachage des mots de passe avant stockage dans la base de données MySQL, et la gestion des sessions utilisateur ou l'utilisation de tokens pour maintenir l'état de connexion. Le processus d'inscription professionnelle inclura une pré-validation et l'envoi de documents nécessaires pour la validation finale par l'administrateur. Un utilisateur (user) peut également devenir un prestataire (prestataire).

### **Gestion de Profils :**

Gestion de Profils : Implémentation des fonctions PHP pour permettre aux utilisateurs de consulter et de modifier leurs informations personnelles et professionnelles. Les modifications de la situation professionnelle des prestataires (services, tarifs, adresse, contacts, cessation/reprise d'activité) nécessiteront une validation par l'administrateur.

### **Gestion des Rôles :**

Définition et gestion des rôles (Particulier, Prestataire, Administrateur) et des permissions associées, assurant que chaque utilisateur n'accède qu'aux fonctionnalités autorisées.

### **Gestion des services et catégories :**

#### **CRUD des Catégories et Services :**

Scripts PHP pour l'ajout, la modification, la suppression et la récupération des catégories de services (ex: Plomberie, Électricité) et des services spécifiques proposés par les prestataires (ex: Installation de robinet, Taille de haie).

#### **Détails des Services :**

Chaque service pourra inclure une description détaillée, un tarif, la localisation des services et la gestion des créneaux de disponibilité.

#### **Portfolio et Documents :**

Gestion du téléchargement, du stockage sécurisé et de l'association des images/vidéos de portfolio et des documents justificatifs (licences, assurances) aux profils des prestataires. Ces documents nécessiteront une validation par l'administrateur.

- **Gestion des demandes de service et réservations :**

**Création de Demandes :**

Fonctions PHP pour enregistrer les demandes de service faites par les particuliers, incluant les détails du besoin, la date souhaitée et la localisation.

**Workflow Réactif :**

Mise en place d'une logique backend pour un workflow réactif, informant le client de l'avancement du processus et des étapes clés

**Devis :**

Fonctionnalité pour que les prestataires puissent envoyer des devis détaillés en réponse aux demandes des clients.

**Gestion des avis et notations :**

- **Soumission et Récupération :**

API en PHP pour permettre aux clients de soumettre des avis et des notes sur les services rendus, et pour récupérer ces avis.

**Statistiques et Reporting :**

- **Collecte de Données :**

implémentation de requêtes SQL et de scripts PHP pour collecter et agréger les données pertinentes (nombre de demandes par catégorie, revenus générés, prestataires les plus populaires, taux de satisfaction client).

**Sécurité des données**

- **Hachage de Mots de Passe :**

Utilisation de fonctions de hachage robustes (ex: password\_hash() en PHP) pour stocker les mots de passe de manière sécurisée.

## 4 .Tests et validation

Cette section décrit les différentes phases de tests et de validation menées tout au long du cycle de vie du projet pour garantir la qualité, la fiabilité et la conformité de la plateforme aux exigences fonctionnelles et non fonctionnelles définies dans le cahier des charges.

### 4.1 Stratégie de tests

Nous avons adopté une stratégie de tests complète pour assurer la robustesse de l'application :

- Tests Unitaires : Vérification du bon fonctionnement de chaque composant de code de manière isolée.
- Tests d'Intégration : Validation des interactions entre les différents modules de l'application (frontend, backend, base de données).
- Tests Fonctionnels : Vérification que toutes les fonctionnalités métier répondent aux exigences du cahier des charges, en simulant les scénarios d'utilisation par les Particuliers, Prestataires et Administrateurs.
- Tests de Performance : Évaluation de la réactivité et de la stabilité de l'application sous différentes charges.
- Tests de Sécurité : Identification des vulnérabilités potentielles pour protéger les données (ex: injections SQL, XSS, CSRF).
- Tests de Compatibilité/Responsivité : Assurer un affichage et un fonctionnement optimaux sur divers navigateurs et appareils (ordinateurs, mobiles, tablettes).

## 4.2 Cas de tests et scénarios

Des cas de tests détaillés ont été définis pour couvrir les fonctionnalités critiques. Voici quelques exemples :

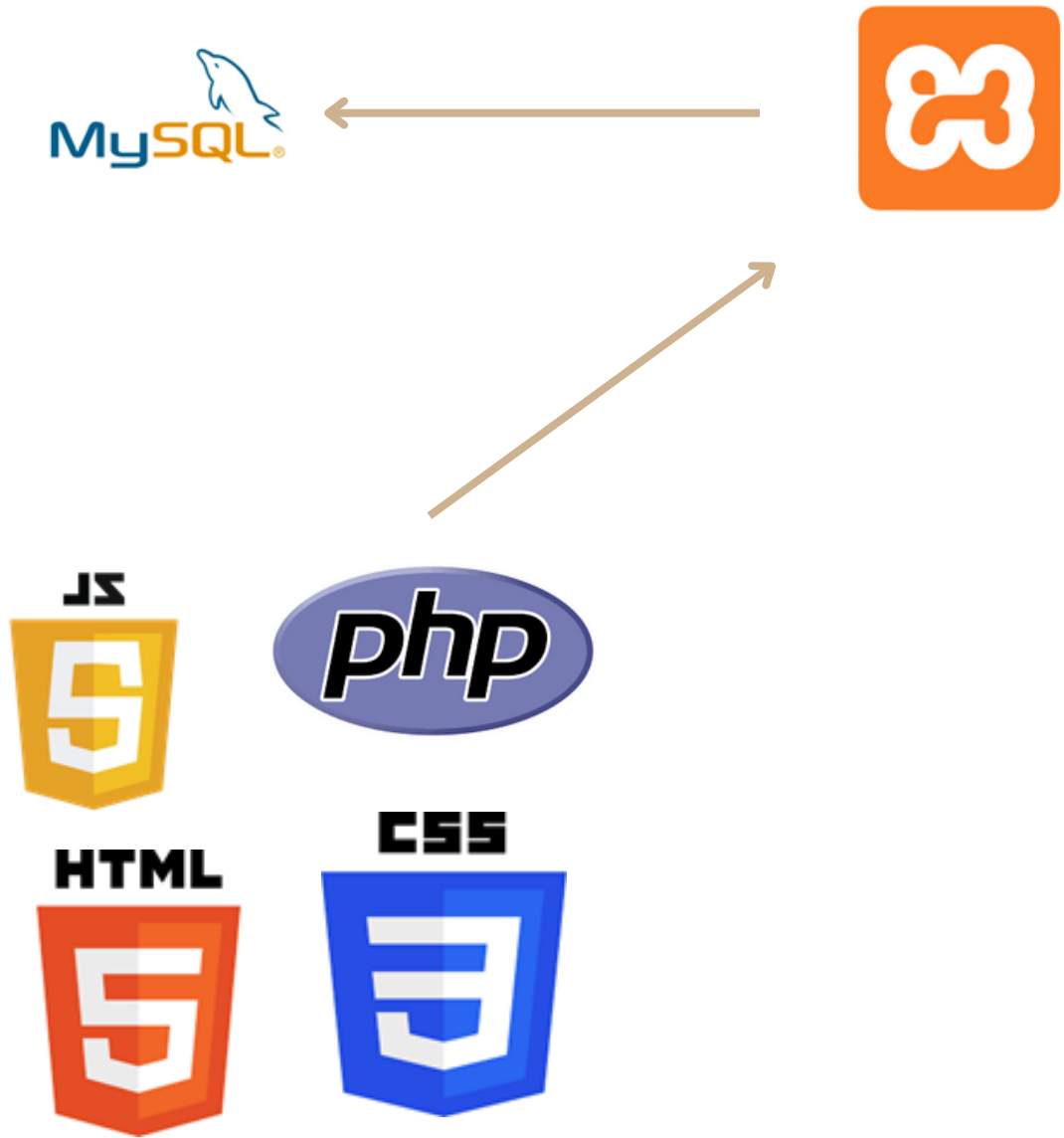
- Inscription d'un nouveau Particulier : Vérifier la création complète du compte et l'enregistrement en base de données.
- Demande de service par un Particulier : S'assurer que le processus de soumission de demande fonctionne et que le prestataire reçoit la notification.
- Acceptation d'une demande par un Prestataire : Confirmer la mise à jour du statut de la demande et la notification du client.
- Validation d'un Prestataire par l'Administrateur : Vérifier que l'administrateur peut activer le profil d'un prestataire après revue.

## 5 . L'architecture d'application

C'est un site qui suit la structure back-end, c'est-à-dire qu'il sépare la couche d'accès aux données de la couche de présentation.

La base de données est facile à trouver phpmyadmin.

Ce site est facile à utiliser et possède des icônes simples et simples.



architecture de souka.ma

L'architecture adoptée pour notre application web souka.ma est l'architecture 3-tiers, une approche structurée qui divise le système en trois couches distinctes pour optimiser la modularité, la gestion et l'évolutivité.



- **Couche Présentation (Frontend) :** Cette couche représente la façade visible de souka.ma, l'interface interactive avec laquelle les particuliers, les professionnels et les administrateurs interagissent directement via leurs navigateurs web. Elle est responsable de l'affichage des services, des profils, des formulaires de demande et de réservation, et de la gestion de l'expérience utilisateur globale. Elle est développée en HTML, CSS et JavaScript, assurant une interface conviviale et réactive sur tous les types d'appareils.
- **Couche Traitement (Backend - Logique Métier) :** C'est le cerveau de souka.ma, où réside toute la logique métier de l'application. Elle reçoit les requêtes des utilisateurs depuis la couche de présentation, les traite, applique les règles métier (gestion des inscriptions, des profils, des demandes de service, des validations, des paiements, etc.), et orchestre les interactions avec la base de données. Cette couche est principalement développée en PHP, et expose une API RESTful pour communiquer efficacement avec le frontend via le serveur web Apache.
- **Couche Accès aux Données :** Cette couche est dédiée à la gestion et à la persistance de toutes les informations de souka.ma. Elle stocke de manière organisée les données relatives aux utilisateurs (particuliers et prestataires), aux services, aux catégories, aux réservations, aux avis, et aux transactions. L'accès, la manipulation et la sécurisation de ces données sont gérés par un système de gestion de base de données MySQL.



architecture client database

## 6 . Annexes.



The screenshot shows the login interface for Souka.ma. At the top is the Souka.ma logo. Below it, a link reads "Vous n'avez pas de compte ? Créez votre compte". The main heading is "Connexion à votre compte". There are two radio buttons: "Ou" (selected) and "Ou" (unselected). Below are three input fields: "Adresse e-mail" (with a sub-label "Adresse e-mail"), "Mot de passe" (with a sub-label "Mot de passe"), and a "Se connecter" button.

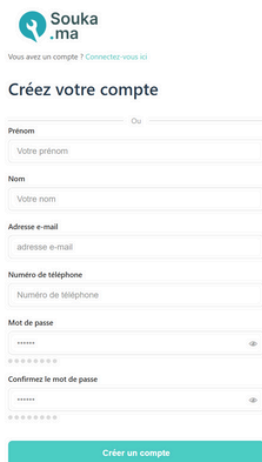


page se connecter

Page de connexion à un compte utilisateur (client ou prestataire) sur le site Souka.ma.

Elle permet à un utilisateur :

- de saisir son e-mail et mot de passe,
- de se connecter à son compte existant,
- ou d'aller vers la page Créer un compte s'il n'a pas encore de compte

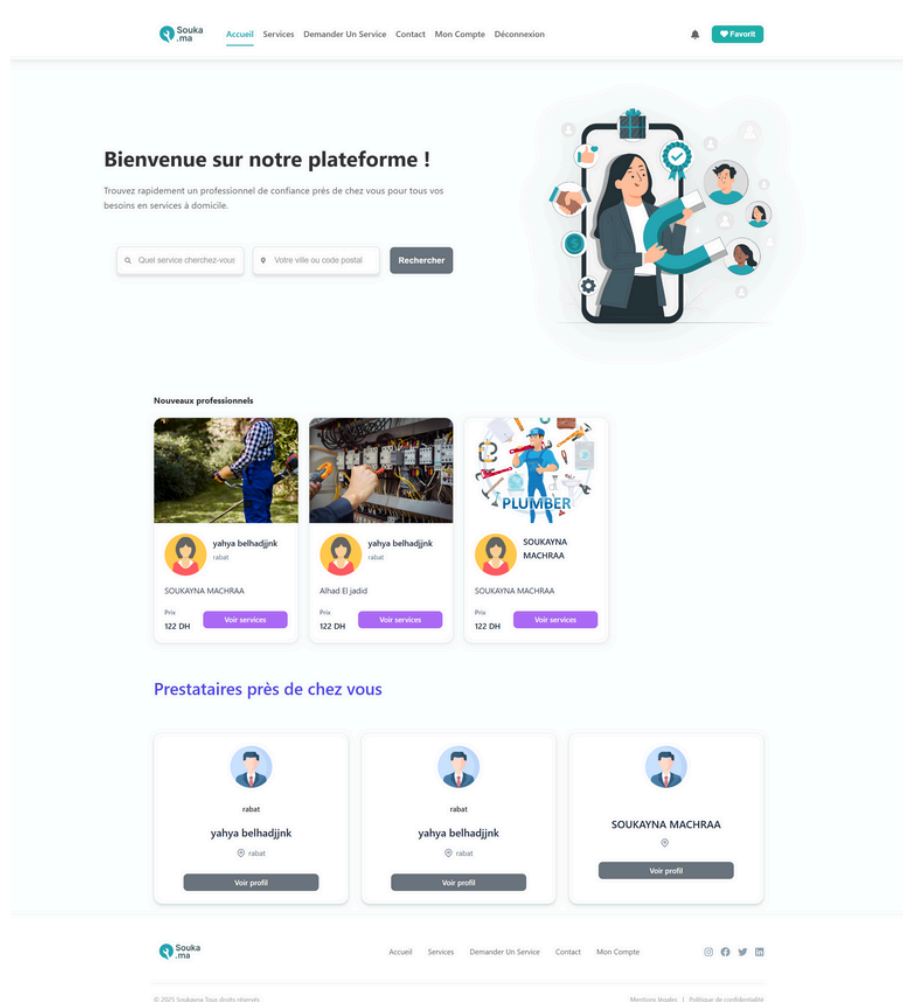


The screenshot shows the registration interface for Souka.ma. At the top is the Souka.ma logo. Below it, a link reads "Vous avez un compte ? Connectez-vous ici". The main heading is "Créez votre compte". There are two radio buttons: "Ou" (selected) and "Ou" (unselected). Below are several input fields: "Prénom" (with a sub-label "Votre prénom"), "Nom" (with a sub-label "Votre nom"), "Adresse e-mail" (with a sub-label "adresse e-mail"), "Numéro de téléphone" (with a sub-label "Numéro de téléphone"), "Mot de passe" (with a sub-label "Mot de passe"), and "Confirmez le mot de passe" (with a sub-label "Confirmez le mot de passe"). At the bottom is a "Créer un compte" button.



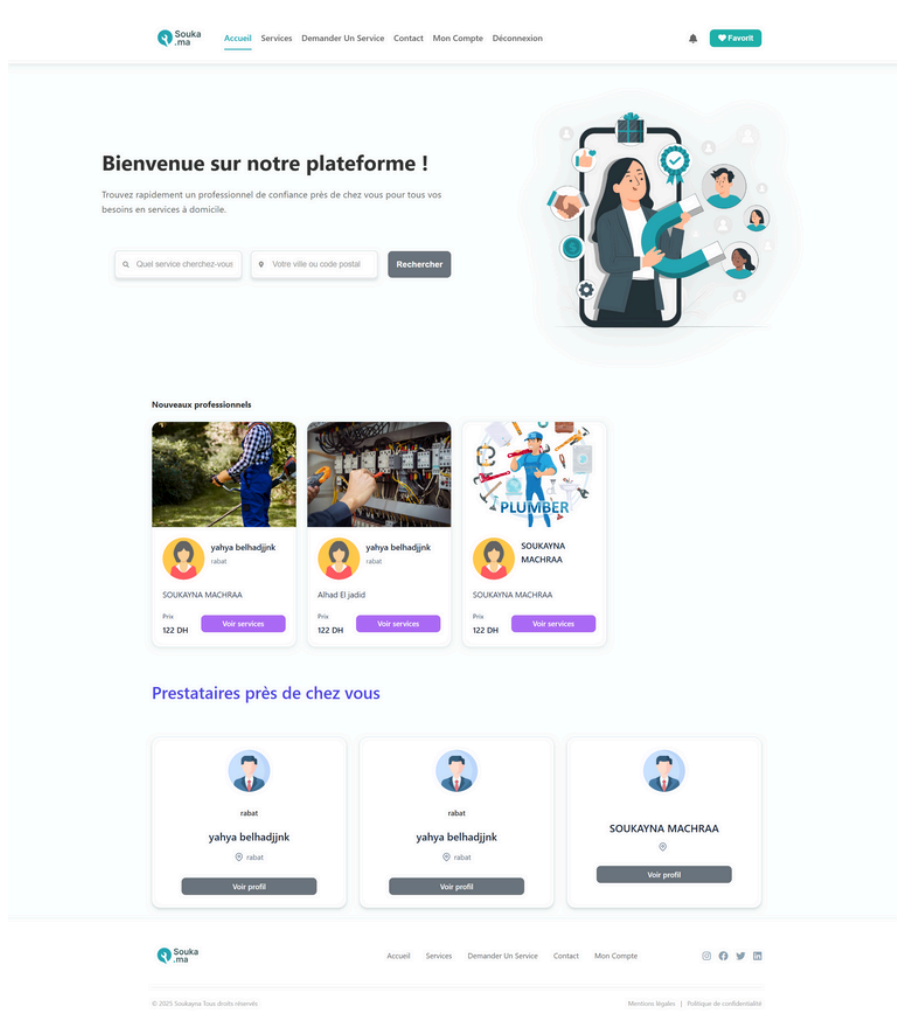
page s'inscrire

La page affichée est une interface de création de compte sur le site Souka.ma Elle permet à un utilisateur de créer un compte en saisissant son prénom, nom, adresse e-mail, numéro de téléphone, mot de passe, et en confirmant ce dernier. Une option de connexion via Google est également disponible pour simplifier l'inscription. À droite, une illustration montre un décor traditionnel marocain avec une fontaine, des plantes et des personnes en tenue de travail, suggérant un lien avec des services locaux, peut-être lié à l'entretien ou à la gestion. Le bouton "Créer un compte" facilite l'accès pour les nouveaux utilisateurs.



page accueil

La page d'accueil de « Souka.ma » est une plateforme conviviale conçue pour mettre en relation les particuliers avec des professionnels de services locaux. Elle dispose d'une barre de recherche bien visible pour trouver des services par type et par localisation, présente les nouveaux prestataires et les prestataires à proximité avec des informations essentielles comme le nom, la ville, le service et le prix, et propose une navigation claire et des appels à l'action, le tout dans une interface claire et attrayante.



page service

Sur la page des services de souka.ma, les utilisateurs peuvent facilement découvrir l'étendue des prestations proposées. Cette section clé de l'interface frontend met en avant une organisation claire grâce à des catégories de services populaires (comme "Plomberie", "Jardinage", "Mécanique", "Ménage", "Électricité").

Service demandé  
Ménage

Catégorie  
Ménage

Date du service  
jj/mm/aaaa

Heure du service  
--:--

Numéro de téléphone  
+212076252104

Ville  
Sélectionner une ville

Statut  
Sélectionner le statut

Valider

page demander

page demander de service

Bienvenue, faty faty!

Devenir Prestataire

Nom  
faty faty

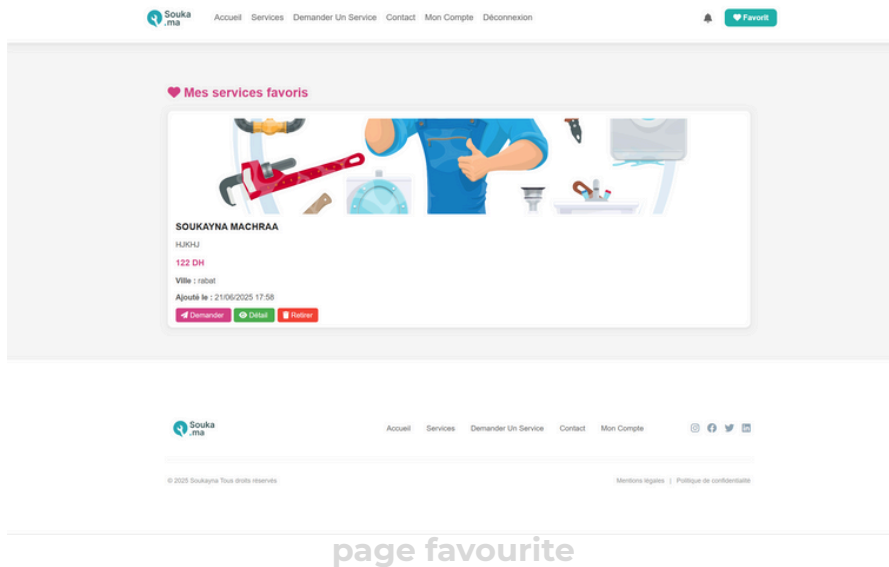
Numéro de téléphone  
604773687

Ville  
Taounate

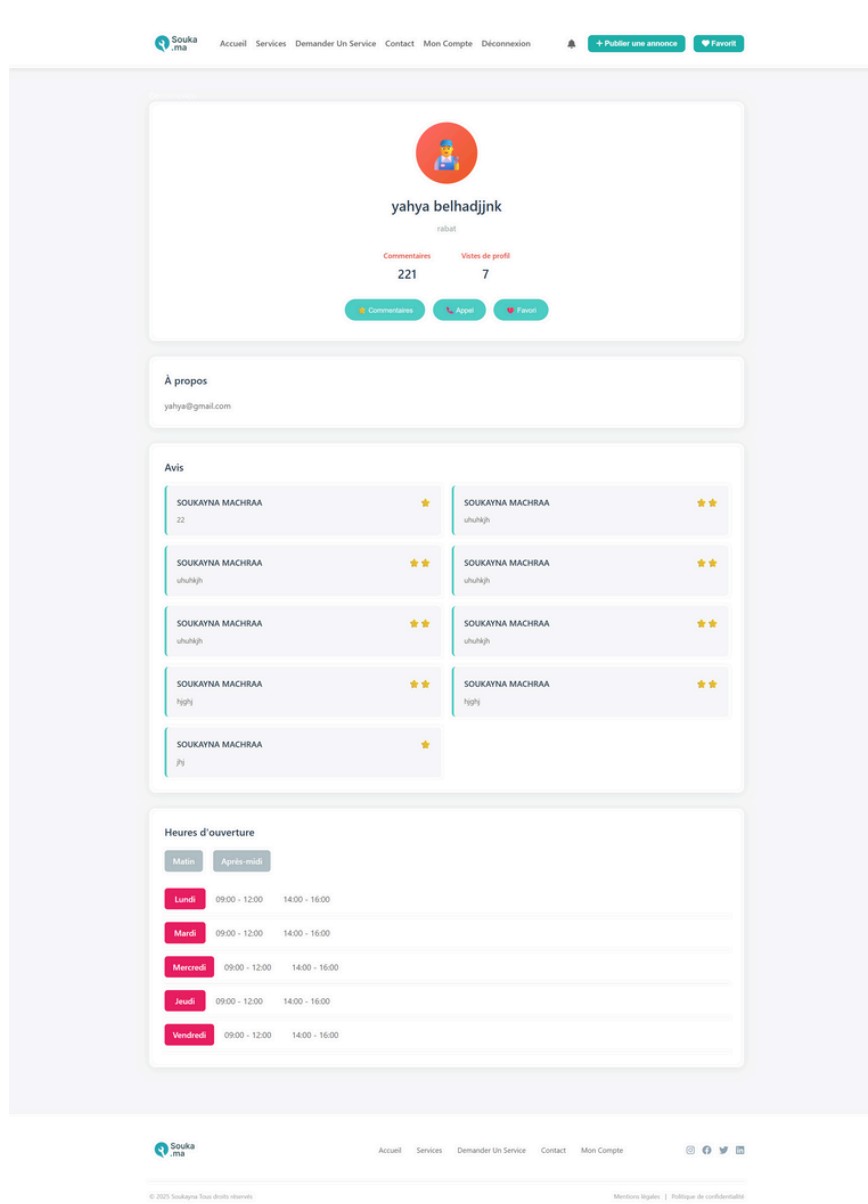
Mettre à jour

mon compte user

## VOTRE TEXTE DE PARAGRAPHE



La page "Mes favoris" est conçue pour vous offrir un accès rapide et personnalisé à vos services et prestataires préférés. Ici, vous retrouvez en un coup d'œil les profils que vous avez épinglés pour leur qualité ou leur pertinence



## page prestataire

page de compte de prestLe Compte Prestataire : Gérez Votre Présence et Vos Services

## 7 .Résultats et Bilan

Cette section est cruciale pour présenter de manière synthétique et objective les aboutissements de votre projet. Il s'agit de montrer ce qui a été réalisé concrètement et d'en évaluer la pertinence par rapport aux objectifs fixés.

- Présentation des livrables : Décrivez et listez les principaux éléments produits. Dans le cas d'un projet web comme le vôtre, cela inclurait :
  - La plateforme "Souka.ma" elle-même (site web fonctionnel).
  - Les différentes fonctionnalités implémentées (recherche de services, affichage des professionnels, gestion de compte utilisateur/prestataire, etc.).
  - La base de données (schéma, tables \_user, service, etc.).
  - Tout autre document ou outil développé (par exemple, un cahier des charges, des maquettes, des scripts spécifiques).
- Analyse des performances : Si applicable, présentez des métriques ou des indicateurs de performance. Pour votre plateforme, cela pourrait être :
- Technique : Temps de chargement des pages, réactivité de la recherche, gestion des erreurs (comme les erreurs de connexion à la base de données).
- Fonctionnelle : Le nombre de professionnels inscrits, le nombre de services listés (si ces données sont mockées ou réelles à ce stade du projet), la facilité d'utilisation perçue.
- Conformité aux objectifs : Reprenez les objectifs initialement définis (voir votre introduction ou la section sur les objectifs) et évaluez dans quelle mesure ils ont été atteints.
- Soulignez les objectifs pleinement réalisés.
- Expliquez les éventuels objectifs partiellement atteints ou non atteints, en justifiant les raisons (contraintes techniques, temps, complexité inattendue, etc.).
- Points forts du projet : Mettez en avant les aspects les plus réussis de votre développement. Cela peut être :
  - La qualité du design (UI/UX).
  - La robustesse du code.
  - L'efficacité d'une fonctionnalité particulière.
  - La clarté de l'interface.
  - La pertinence de la solution apportée au problème initial.



## 8. Conclusion

En conclusion, ce Projet de Fin d'Études, la plateforme "Souka.ma", a permis de développer une solution web fonctionnelle et intuitive visant à simplifier la mise en relation entre les particuliers et les professionnels des services à domicile. En dépit des défis techniques rencontrés, la rigueur de la conception et la persévérance dans le développement ont abouti à une application robuste et réactive, intégrant des fonctionnalités essentielles de recherche, de présentation des prestataires et de gestion des utilisateurs. Ce projet a non seulement consolidé nos compétences en développement web (PHP, SQL, HTML/CSS, Git), mais a également renforcé notre approche de la résolution de problèmes et notre capacité à transformer une idée en une solution concrète et potentiellement utile pour la communauté. Les perspectives d'évolution sont nombreuses, et "Souka.ma" jette les bases d'une plateforme qui pourrait grandir pour devenir un acteur significatif dans le domaine des services locaux.