

Université Sésame

Prépa 2 C

Rapport de projet

Application web de réservation de services à domicile
(Projet Serve Co)

Étudiant : Nasfi Mohamed Shamseddine et Lina Ben Romdhane

Année universitaire : 2025–2026



12 janvier 2026

Table des matières

Introduction	3
1 Contexte et problématique	4
1.1 Contexte général	4
1.2 Problématique	4
2 Analyse des besoins	5
2.1 Besoins fonctionnels	5
2.2 Besoins non fonctionnels	5
3 Choix technologiques	6
3.1 Technologies front-end	6
3.2 Gestion de la mise en page et du style	6
4 Architecture de l'application	7
4.1 Organisation des pages	7
4.2 Composants transverses	7
5 Conception fonctionnelle	8
5.1 Parcours de réservation	8
5.2 Gestion des services et offres	8
6 Implémentation front-end	9
6.1 Hooks et gestion d'état	9
6.2 Formulaires et feedback utilisateur	9
7 Design et expérience utilisateur	10
7.1 Identité visuelle et mise en page	10
7.2 Valeurs de la marque et contenu	10
8 Page Contact et relation client	11
8.1 Informations de contact	11
8.2 Formulaire de contact	11

9 Discussion et limites	12
9.1 Forces de la solution	12
9.2 Limites et perspectives	12
Conclusion	13

Introduction

Dans le contexte actuel, les services à domicile occupent une place de plus en plus importante dans le quotidien des particuliers, qu'il s'agisse de ménage, de blanchisserie, de garde d'enfants ou encore de lavage de voitures. Pourtant, la mise en relation entre clients et prestataires reste souvent fragmentée, peu digitalisée et peu transparente. Ce projet vise à concevoir et développer une application web moderne permettant de réserver facilement ces services, avec une expérience utilisateur fluide et une gestion structurée des réservations.

L'objectif de ce rapport est de présenter le besoin fonctionnel, les choix techniques, l'architecture logicielle, ainsi que la mise en œuvre de l'application *Serve Co*, réalisée en utilisant le framework Next.js, des composants UI réutilisables et une organisation modulaire des pages.

Chapitre 1

Contexte et problématique

1.1 Contexte général

Les utilisateurs recherchent aujourd’hui des solutions centralisées pour gérer leurs tâches domestiques, avec des critères forts de confiance, de qualité et de rapidité de réservation. Les plateformes classiques restent souvent spécialisées sur un seul type de service (par exemple uniquement le ménage) et ne proposent pas toujours une vision unifiée de l’offre.

L’application *Serve Co* se positionne comme une plateforme de services à domicile couvrant plusieurs catégories : nettoyage de maison, blanchisserie, toilettage d’animaux, lavage de voiture et babysitting. L’utilisateur peut ainsi réserver différents services depuis une interface unique.

1.2 Problématique

La problématique principale peut être formulée comme suit :

Comment concevoir une application web moderne et ergonomique permettant à un utilisateur non technique de réserver, en quelques étapes, un service à domicile tout en garantissant la clarté des informations, la confiance envers les prestataires et la simplicité de la navigation ?

Cette problématique se décline en plusieurs sous-questions :

- Comment structurer le parcours utilisateur pour réduire le nombre d’étapes tout en collectant les informations nécessaires à la réservation ?
- Comment présenter les services, les offres et les tarifs de manière lisible et attractive ?
- Comment organiser le code en composants réutilisables pour faciliter l’évolution du projet ?

Chapitre 2

Analyse des besoins

2.1 Besoins fonctionnels

Les besoins fonctionnels majeurs identifiés pour l'application sont les suivants :

- Consultation d'une page d'accueil mettant en avant l'offre de services et les principaux bénéfices pour l'utilisateur.
- Accès à une page de présentation de l'entreprise, de sa mission, de ses valeurs et de quelques statistiques clés (clients satisfaits, prestataires vérifiés, etc.).
- Possibilité de réserver un service à travers un formulaire en plusieurs étapes : choix du service, des options, de la date, de l'horaire et saisie des informations personnelles.
- Possibilité de contacter l'entreprise via un formulaire dédié permettant d'envoyer un message et des coordonnées de contact.

2.2 Besoins non fonctionnels

Les principaux besoins non fonctionnels retenus sont :

- **Ergonomie** : interface épurée, formulaires clairs, feedback visuel après une réservation ou l'envoi d'un message.
- **Performance** : chargement rapide des pages grâce à Next.js et au découpage en composants.
- **Réutilisabilité** : composants UI génériques (boutons, cartes, champs de formulaire) pour faciliter la maintenance.
- **Cohérence visuelle** : utilisation d'une feuille de style globale et d'un thème commun à toutes les pages.

Chapitre 3

Choix technologiques

3.1 Technologies front-end

L'interface utilisateur a été développée avec **Next.js**, un framework basé sur React, qui permet le rendu côté serveur ainsi qu'une bonne organisation en pages et en layouts. L'utilisation de la directive `use client` sur certaines pages indique qu'elles sont rendues côté client afin de pouvoir exploiter les hooks React comme `useState` et `useSearchParams`.

Les composants UI (boutons, cartes, champs de texte, sélecteurs, etc.) sont importés depuis un ensemble de composants personnalisés, ce qui contribue à la cohérence et à la réutilisabilité du code. Des icônes provenant de la bibliothèque *lucide-react* sont utilisées pour enrichir l'interface (icônes d'e-mail, téléphone, carte, validation, etc.).

3.2 Gestion de la mise en page et du style

La structure globale du site est définie dans un composant de layout racine, qui intègre les polices, le thème et l'Analytics de Vercel. Le fichier `globals.css` centralise les styles généraux, incluant la typographie, les couleurs et les classes utilitaires communes.

Chaque page (`Home`, `About`, `Book`, `Contact`) réutilise ce layout et applique des classes CSS utilitaires pour gérer les marges, la disposition en grille et le responsive design.

Chapitre 4

Architecture de l'application

4.1 Organisation des pages

L'application s'appuie sur plusieurs pages principales :

- **Page d'accueil** (`HomePage`) : inclut une section héro, un aperçu des services, les avantages, des témoignages et un appel à l'action.
- **Page À propos** (`AboutPage`) : présente la mission, les valeurs, une image de l'équipe et quelques statistiques (clients, prestataires, services, note moyenne).
- **Page Réservation** (`BookPage`) : permet à l'utilisateur de réserver un service via un formulaire en plusieurs étapes avec résumé de réservation.
- **Page Contact** (`ContactPage`) : fournit les coordonnées (e-mail, téléphone, adresse) et contient un formulaire de contact.

Ces pages sont agencées dans une structure de layout globale avec un composant de navigation en haut et un pied de page commun.

4.2 Composants transverses

Plusieurs composants sont partagés entre les pages :

- **Navigation** : barre de navigation contenant le logo et les liens vers les pages principales.
- **Footer** : pied de page avec des liens secondaires et éventuellement des informations légales.
- **Composants UI** : `Button`, `Card`, `Input`, `Textarea`, `Select`, `Calendar`, `Popover`, etc.

Cette architecture en composants favorise la modularité et permet d'étendre le projet en ajoutant facilement de nouvelles pages ou fonctionnalités.

Chapitre 5

Conception fonctionnelle

5.1 Parcours de réservation

La réservation d'un service se fait via un formulaire en trois étapes définies dans la page BookPage :

1. **Choix du service et du package** : l'utilisateur sélectionne une catégorie de service (ménage, blanchisserie, toilettage, etc.) et un package associé avec un prix.
2. **Informations personnelles** : l'utilisateur saisit son nom complet, son numéro de téléphone et l'adresse où le service doit être rendu.
3. **Résumé et confirmation** : un récapitulatif affiche le service choisi, le package, la date, l'horaire et les coordonnées, avant la confirmation de la réservation.

Des validations simples sont en place afin d'empêcher la navigation vers l'étape suivante tant que certains champs obligatoires ne sont pas renseignés.

5.2 Gestion des services et offres

Les services sont décrits sous forme de structure statique contenant :

- un identifiant de service (`value`) et un libellé (`label`) ;
- une liste de packages, chacun défini par un identifiant, un nom et un prix (ex. « Basic Clean », « Deep Clean », etc.).

Cette structure est exploitée pour générer dynamiquement les options de sélection et l'affichage des cartes de packages, ce qui simplifie l'ajout de nouveaux services ou la modification des tarifs.

Chapitre 6

Implémentation front-end

6.1 Hooks et gestion d'état

La logique de formulaire de réservation repose sur le hook `useState` pour stocker :

- l'étape courante du formulaire (`step`) ;
- l'état de soumission (`isSubmitted`) ;
- un objet `formData` contenant les champs `service`, `package`, `date`, `time`, `name`, `phone`, `address` et `notes`.

Le hook `useSearchParams` est utilisé pour pré-sélectionner un service en fonction des paramètres de l'URL, ce qui permet de créer des liens de réservation ciblés depuis d'autres pages.

6.2 Formulaires et feedback utilisateur

Les formulaires utilisent des composants `Input`, `Textarea`, `Select` et un `Calendar` pour la sélection de la date. L'utilisateur bénéficie d'un feedback visuel lors de la soumission :

- en cas de succès, un message de confirmation est affiché avec un symbole de validation (`CheckCircle2`).
- l'utilisateur peut ensuite réinitialiser le formulaire pour effectuer une nouvelle réservation ou envoyer un nouveau message.

Un composant de chargement est également défini pour indiquer à l'utilisateur que le formulaire de réservation se charge lorsqu'il est encapsulé dans un `Suspense`.

Chapitre 7

Design et expérience utilisateur

7.1 Identité visuelle et mise en page

Le design privilégie une mise en page centrée, avec des sections de largeur maximale contrôlée et des marges généreuses, afin d'améliorer la lisibilité. La typographie utilise une combinaison de polices importées (par exemple Inter et Playfair Display) pour distinguer les titres du corps de texte.

Les cartes, boutons et icônes sont employés pour structurer l'information et guider le regard de l'utilisateur vers les actions principales (réserver un service, envoyer un message, etc.).

7.2 Valeurs de la marque et contenu

La page « À propos » décrit la mission de la plateforme, qui est de rendre les services à domicile plus simples et plus fiables. Elle met en avant des valeurs clés telles que le *care*, l'excellence, la confiance et l'intégrité, chacune illustrée par une icône et un descriptif.

Des statistiques mises en forme (par exemple nombre de clients satisfaits, prestataires vérifiés, services complétés et note moyenne) renforcent la crédibilité et la perception de fiabilité de la plateforme.

Chapitre 8

Page Contact et relation client

8.1 Informations de contact

La page de contact affiche plusieurs modes de communication :

- une adresse e-mail dédiée aux demandes des clients ;
- un numéro de téléphone pour un contact direct ;
- une adresse géographique à Ariana, avec un lien vers Google Maps.

Chaque information est présentée dans une carte avec une icône correspondante (mail, téléphone, localisation), ce qui facilite l'identification visuelle.

8.2 Formulaire de contact

Un formulaire permet à l'utilisateur de saisir :

- son nom ;
- son e-mail ;
- éventuellement son numéro de téléphone ;
- le message à adresser à l'équipe.

Après soumission, un message de confirmation informe l'utilisateur que son message a été reçu et qu'une réponse sera envoyée dans un délai déterminé. Là encore, un bouton permet de réinitialiser le formulaire pour envoyer un nouveau message.

Chapitre 9

Discussion et limites

9.1 Forces de la solution

L'application *Serve Co* présente plusieurs points forts :

- un parcours de réservation clairement découpé en étapes ;
- une architecture modulaire basée sur des composants réutilisables ;
- une identité visuelle cohérente grâce à une feuille de style globale et un layout partagé ;
- la mise en avant de la confiance et de la transparence via la page À propos et les statistiques.

9.2 Limites et perspectives

Certaines limites peuvent toutefois être relevées :

- les données de services et tarifs sont actuellement statiques dans le code, ce qui limite l'évolutivité sans back-end dédié ;
- la gestion réelle des réservations (stockage en base, gestion des comptes utilisateurs, notifications) n'est pas encore implémentée ;
- l'absence d'un système d'authentification empêche pour l'instant la gestion de profils clients et prestataires.

Parmi les perspectives d'évolution, on peut envisager :

- l'intégration d'une API et d'une base de données pour stocker les réservations et les utilisateurs ;
- la mise en place d'un espace client permettant de suivre l'historique des services ;
- l'ajout de systèmes de notation et d'avis sur les prestataires.

Conclusion

Ce projet a permis de concevoir et de réaliser une application web de réservation de services à domicile reposant sur des technologies modernes et une architecture front-end claire. La solution proposée offre une interface ergonomique pour la sélection de services, la saisie d'informations et la confirmation des réservations, tout en valorisant la confiance et la transparence auprès des utilisateurs.

Le travail effectué constitue une base solide pour de futures évolutions, notamment l'intégration d'un back-end complet, l'ajout de fonctionnalités avancées et le déploiement en production dans un contexte réel d'utilisation.