

Projet : Site vitrine et système de réservation pour une société de nettoyage automobile

1. Objectif du projet

Créer un **site vitrine moderne** permettant aux clients de : - Consulter les formules et services proposés. - Prendre un rendez-vous en ligne. - (Optionnel) Créer un compte client.

L'administrateur (le gérant ou ses employés) pourra : - Gérer les rendez-vous. - Modifier les formules ou les tarifs.

L'objectif secondaire est d'utiliser ce projet comme **exercice technique** pour monter en compétence sur une **stack JavaScript moderne**.

2. Choix techniques envisagés



Stack moderne (JavaScript / TypeScript)

Front-end : React

- Créer une interface moderne, responsive et fluide.
- Utilisation possible de **Tailwind CSS** ou **Material UI** pour le design.
- Possibilité d'évolution vers **Next.js** pour un meilleur SEO.

Back-end : NestJS

- Framework Node.js structuré, inspiré d'Angular.
- Idéal pour développer une API REST.
- Gestion de l'authentification via JWT.
- ORM recommandé : **TypeORM** (compatible MySQL).

Base de données : MySQL

- Bonne compatibilité avec TypeORM.
- Structure simple : tables `users`, `appointments`, `services`.

Hébergement

Composant	Service recommandé	Gratuit ?	Description
Front React	Netlify	✓	Déploiement simple, rapide, CDN inclus
Back NestJS	Render ou Railway	✓	Héberge API Node.js
Base MySQL	PlanetScale / Railway DB	✓	Base de données cloud fiable

Schéma d'architecture :

React (Netlify) <----> NestJS API (Render) <----> MySQL (PlanetScale)

Avantages :

- Stack moderne et professionnelle (courante dans les entreprises).
- Très bon projet d'apprentissage.
- Gratuite à déployer.

Inconvénients :

- Mise en place plus complexe (CORS, connexions API, ORM, etc.).
- Overkill pour un simple site vitrine sans logique complexe.

Stack alternative (PHP + Twig + MySQL)

Front + Back : PHP (ex : Symfony, Slim, ou Laravel)

- Site vitrine généré côté serveur avec **Twig**.
- Gestion des rendez-vous via formulaires PHP.
- Authentification et CRUD simples.

Base de données : MySQL (hébergée sur le même serveur)

Hébergement : OVH / Ionos / Infomaniak

- Offre mutualisée (~2-3 €/mois) ou essai gratuit.

Avantages :

- Déploiement plus simple (tout sur un seul hébergement).
- Idéal pour un site purement vitrine ou avec peu d'interactivité.

Inconvénients :

- Moins moderne.
- Moins formateur si l'objectif est de progresser sur JS/TS.

3. Alternatives gratuites pour commencer

Option légère sans backend : Netlify Forms + Google Sheets

- Pour un formulaire de contact simple (pas de comptes clients).
- Données stockées via **Netlify Forms**.
- Synchronisation possible avec Google Sheets via **Zapier** ou **Make.com**.

Idéal pour : un site vitrine simple, sans gestion interne.

4. Recommandation finale

Si ton objectif principal est **de progresser sur les technologies modernes**, alors :

Pars sur la stack **React + NestJS + MySQL**.

Commence petit : 1. Page vitrine avec React (hébergée sur Netlify). 2. API NestJS minimale pour recevoir les réservations (hébergée sur Render). 3. Base MySQL sur PlanetScale.

Puis ajoute progressivement : - Authentification. - Panneau admin. - Dashboard des rendez-vous.

Tu apprendras énormément (frontend, backend, API, BDD, déploiement cloud), tout en créant un vrai projet réutilisable dans ton portfolio.

Étapes suggérées

1. Créer le repo GitHub (front + back séparés).
2. Développer le front React avec un design simple.
3. Créer une API NestJS avec endpoints `/services`, `/appointments`, `/users`.
4. Connecter MySQL via TypeORM.
5. Déployer front sur Netlify et back sur Render.
6. Tester la communication front/back.
7. Ajouter les fonctionnalités bonus (authentification, interface admin, etc.).