



Présentation de l'Application ReactJS et Express

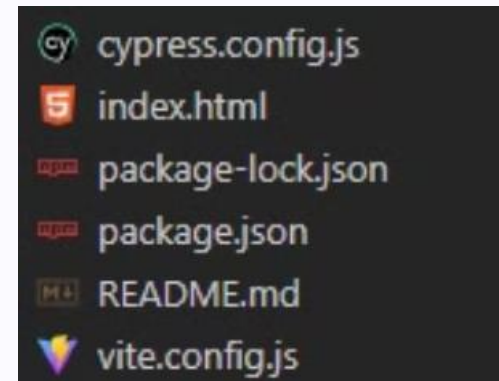
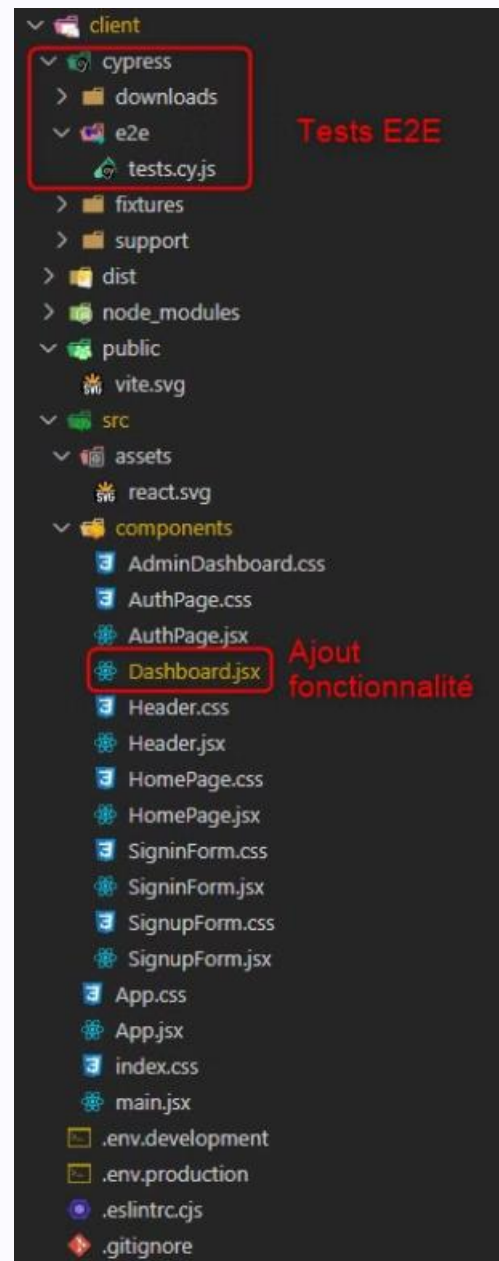
Cette présentation détaillera le développement d'une application d'un site pour garagiste construite à l'aide des technologies ReactJS et Express. Nous explorerons la conception, l'architecture et les fonctionnalités clés de cette solution innovante.



by Brian MOREAU

Modélisation de l'Application

Architecture client



Architecture serveur

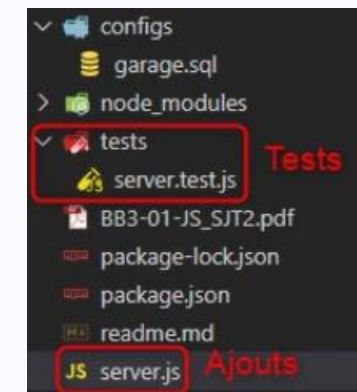
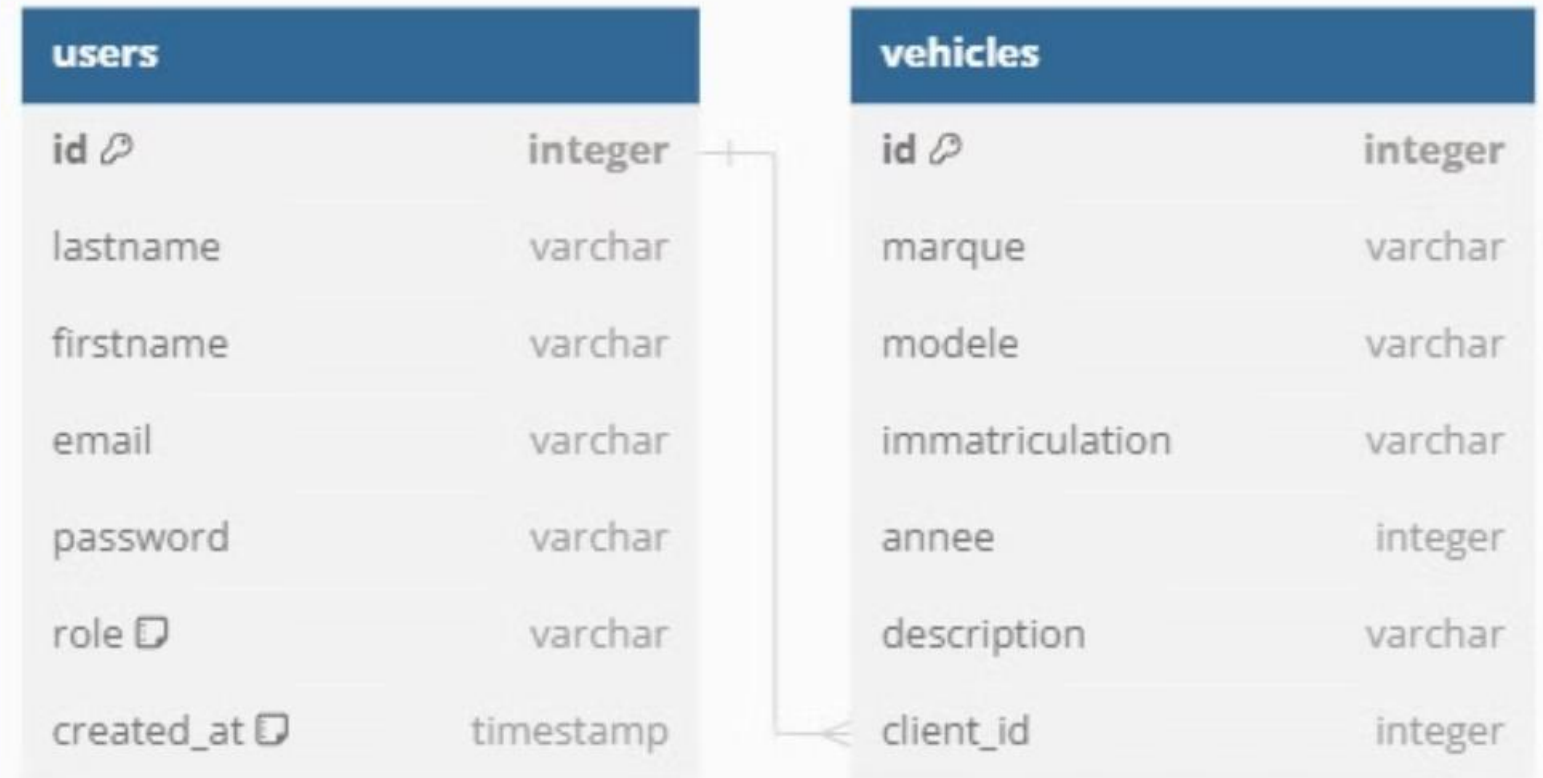
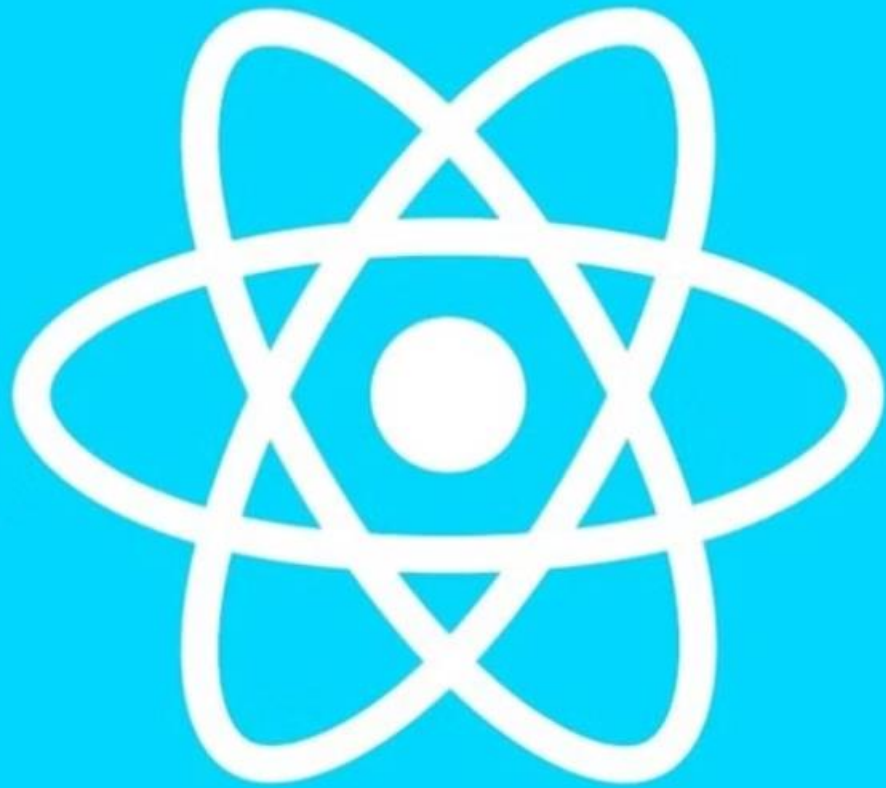


Schéma de la Base de Données





React JS

Choix des Composants client

1

ReactJS

Bibliothèque pour construire l'interface utilisateur de façon dynamique et réactives

2

React Router

Bibliothèque permettant la gestion des routes et des vues dans une application React

3

Vite

Outil de construction moderne pour les applications front-end conçu pour un développement rapide et une construction optimisée pour la production

Choix des Technologies serveur



Express

Express a été choisi pour créer une API RESTful performante et évolutive.



Base de données MySQL

Système de gestion de base de données (SGBD) relationnelle open-source. Stocke les données sous forme de tables.



Bcrypt

Permet de hacher les mots de passe de manière sécurisée en utilisant l'algorithme bcrypt



Json Web Token (JWT)

Bibliothèque permettant de créer et vérifier des JWT, utilisés pour la gestion des sessions

xpress J



Implémentation de la sécurité

1

Gestion des mots de passe

Hashage avec bcrypt pour sécuriser les mots de passe avant stockage en base de données

2

Authentification

- Mise en place d'un JWT pour la gestion des sessions et la protection des routes.
- Middleware pour vérifier la présence et la validité du JWT

3

Cross-Origin Resource Sharing

Configuration des CORS avec la librairie "cors" d'Express

- Accepte uniquement les requêtes provenant du site



Implémentation de la sécurité

4

Content Security Policy

La CSP aide à détecter certains types d'attaques comme les XSS. Elle permet de spécifier les sources de contenu autorisées pour une page Web

5

Protection contre les failles XSS

Utilisation des fonctions d'échappements pour éviter l'exécution de scripts malveillants dans les pages Web

6

Protection contre les attaques CSRF

Ajout de token CSRF dans les formulaires pour éviter les attaques CSRF

7

Protection contre l'injection SQL

Utilisation des requêtes paramétrées pour se protéger des injections en séparant les données des commandes SQL

Phase de tests fonctionnels

Tests fonctionnels

Test le comportement complet d'un composant avec JEST

- Récupération liste des véhicules :

```
Run | Debug
it("devrait récupérer la liste des véhicules", async () => {
  const res = await request(app)
    .get("/api/vehicles")
    .set("x-access-token", token);
  expect(res.statusCode).toEqual(200);
  expect(Array.isArray(res.body)).toBe(true);
}, 10000);
```

✓ devrait récupérer la liste des véhicules (68 ms)

- Récupération liste des clients

```
Run | Debug
it("devrait récupérer la liste des clients", async () => {
  const res = await request(app)
    .get("/api/clients")
    .set("x-access-token", token);
  expect(res.statusCode).toEqual(200);
  expect(Array.isArray(res.body)).toBe(true);
}, 10000);
```

✓ devrait récupérer la liste des clients (6 ms)

- Suppression d'un véhicule

- Création d'un véhicule

```
it("devrait créer un véhicule", async () => {
  const vehicle = {
    marque: "Honda",
    modele: "Civic",
    immatriculation: "ABC-123",
    annee: 2021,
    client_id: 1,
  };

  const res = await request(app)
    .post("/api/vehicles")
    .set("x-access-token", token)
    .send(vehicle);
  expect(res.statusCode).toEqual(201);
}, 10000);
```

✓ devrait créer un véhicule (18 ms)

- Mise à jour d'un véhicule

```
it("devrait mettre un jour un véhicule", async () => {
  const updatedVehicle = {
    marque: "Honda",
    modele: "Accord",
    immatriculation: "XYZ-789",
    annee: 2022,
    client_id: 1,
  };

  const res = await request(app)
    .put("/api/vehicles/12")
    .set("x-access-token", token)
    .send(updatedVehicle);
  expect(res.statusCode).toEqual(200);
  expect(res.text).toBe("Véhicule mis à jour");
}, 10000);
```

✓ devrait mettre un jour un véhicule (6 ms)

- Suppression d'un véhicule

```
it("devrait supprimer un véhicule", async () => {
  const res = await request(app)
    .delete("/api/vehicles/12")
    .set("x-access-token", token);
  expect(res.statusCode).toEqual(200);
  expect(res.text).toBe("Véhicule supprimé");
}, 10000);
```

✓ devrait supprimer un véhicule (5 ms)

Résultat global :

```
PASS tests/server.test.js
Tests sur les appels API
  ✓ devrait récupérer la liste des véhicules (68 ms)
  ✓ devrait récupérer la liste des clients (6 ms)
  ✓ devrait créer un véhicule (18 ms)
  ✓ devrait mettre un jour un véhicule (6 ms)
  ✓ devrait supprimer un véhicule (5 ms)

Test Suites: 1 passed, 1 total
Tests: 5 passed, 5 total
Snapshots: 0 total
Time: 1.114 s, estimated 2 s
Ran all test suites.
```


Phase de tests End-to-End (e2e)

Tests end to end

Tests avec Cypress

- Test connexion:

```
it("login", function () {  
  cy.visit("http://localhost:5173");  
  cy.get(".login-button").click();  
  cy.get(".auth-toggle > :nth-child(2)").click();  
  cy.get('[type="email"]').clear("e");  
  cy.get('[type="email"]').type("email@email.fr");  
  cy.get('[type="password"]').clear();  
  cy.get('[type="password"]').type("Password");  
  cy.get(".signin-form > button").click();  
});
```

- Test de création d'un véhicule

```
it("créer un véhicule", function () {  
  cy.visit("http://localhost:5173");  
  cy.get(".login-button").click();  
  cy.get(".auth-toggle > :nth-child(2)").click();  
  cy.get('[type="email"]').clear("e");  
  cy.get('[type="email"]').type("email@email.fr");  
  cy.get('[type="password"]').clear();  
  cy.get('[type="password"]').type("Password");  
  cy.get(".signin-form > button").click();  
  
  cy.get('[placeholder="Marque"]').clear("R");  
  cy.get('[placeholder="Marque"]').type("Renault");  
  cy.get('[placeholder="Modèle"]').clear();  
  cy.get('[placeholder="Modèle"]').type("Clio 4");  
  cy.get('[placeholder="Immatriculation"]').clear();  
  cy.get('[placeholder="Immatriculation"]').type("AA-BBB-CC");  
  cy.get('[type="number"]').clear();  
  cy.get('[type="number"]').type("2018");  
  cy.get('[placeholder="Description"]').clear();  
  cy.get('[placeholder="Description"]').type("Ma description");  
  cy.get("select").select("2");  
  cy.get(":nth-child(3) > button").click();  
});
```

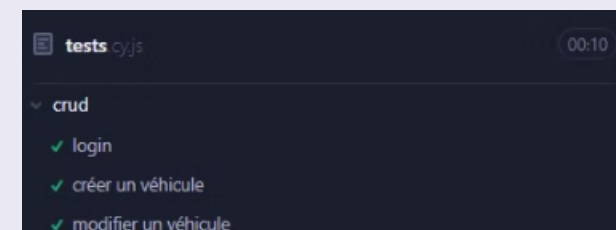
- Test de modification d'un véhicule

```
it("modifier un véhicule", function () {  
  cy.visit("http://localhost:5173");  
  cy.get(".login-button").click();  
  cy.get(".auth-toggle > :nth-child(2)").click();  
  cy.get('[type="email"]').clear("e");  
  cy.get('[type="email"]').type("email@email.fr");  
  cy.get('[type="password"]').clear();  
  cy.get('[type="password"]').type("Password");  
  cy.get(".signin-form > button").click();  
  
  cy.get(":nth-child(12) > :nth-child(7) > :nth-child(1)").click();  
  cy.get("tbody > :nth-child(12)").click();  
  cy.get(":nth-child(1) > input").clear();  
  cy.get(":nth-child(1) > input").type("Renault modifié");  
  cy.get(":nth-child(2) > input").clear("Clio ");  
  cy.get(":nth-child(2) > input").type("Clio 5");  
  cy.get(":nth-child(12) > :nth-child(3) > input").clear();  
  cy.get(":nth-child(12) > :nth-child(3) > input").type("AA-BBB-CC");  
  cy.get(":nth-child(4) > input").clear();  
  cy.get(":nth-child(4) > input").type("2023");  
  cy.get(":nth-child(12) > :nth-child(5)").click();  
  cy.get(":nth-child(5) > input").clear("de");  
  cy.get(":nth-child(5) > input").type("description");  
  cy.get(":nth-child(12) > :nth-child(7) > :nth-child(1)").click();  
});
```

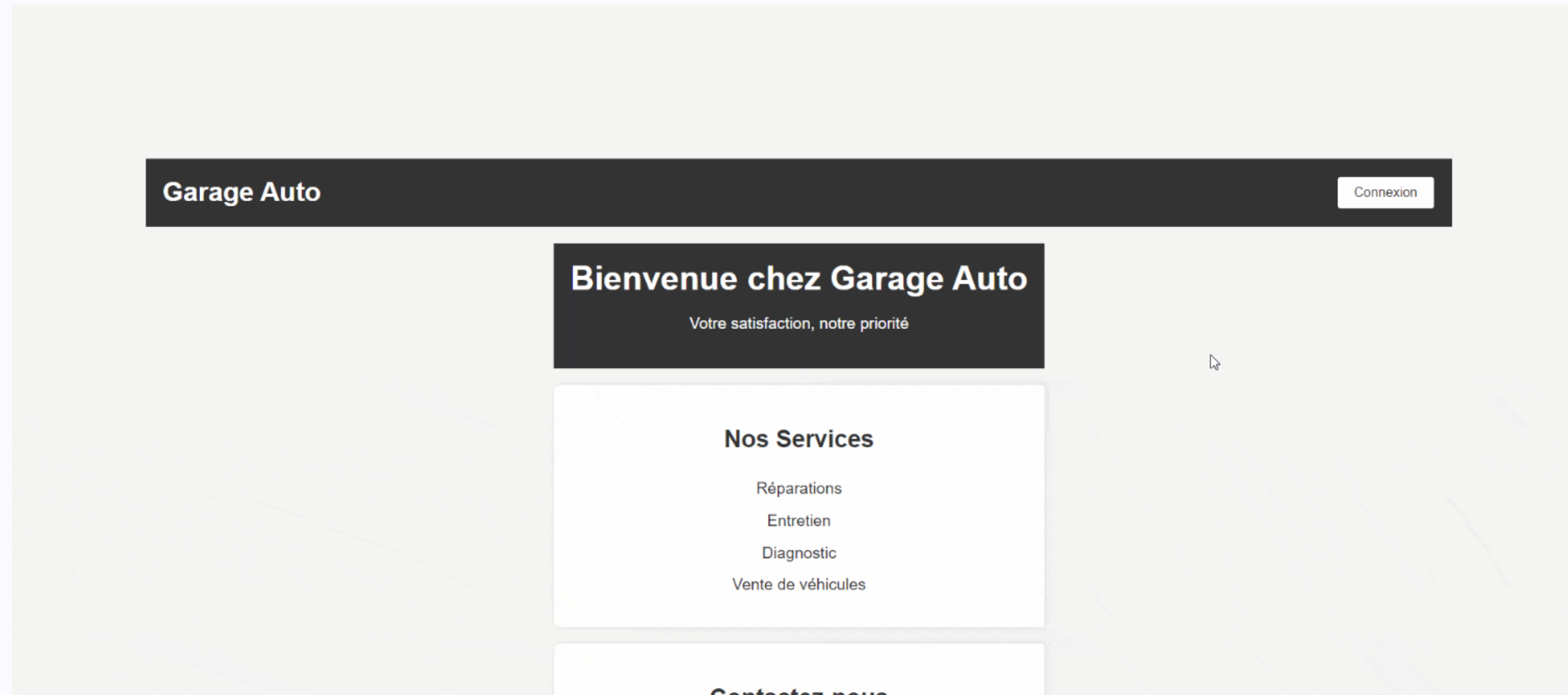
- Test de suppression d'un véhicule

```
it("supprimer un véhicule", function () {  
  cy.visit("http://localhost:5173");  
  cy.get(".login-button").click();  
  cy.get(".auth-toggle > :nth-child(2)").click();  
  cy.get('[type="email"]').clear("e");  
  cy.get('[type="email"]').type("email@email.fr");  
  cy.get('[type="password"]').clear();  
  cy.get('[type="password"]').type("Password");  
  cy.get(".signin-form > button").click();  
  
  cy.get(":nth-child(13) > :nth-child(7) > :nth-child(2)").click();  
});
```

- Résultats :

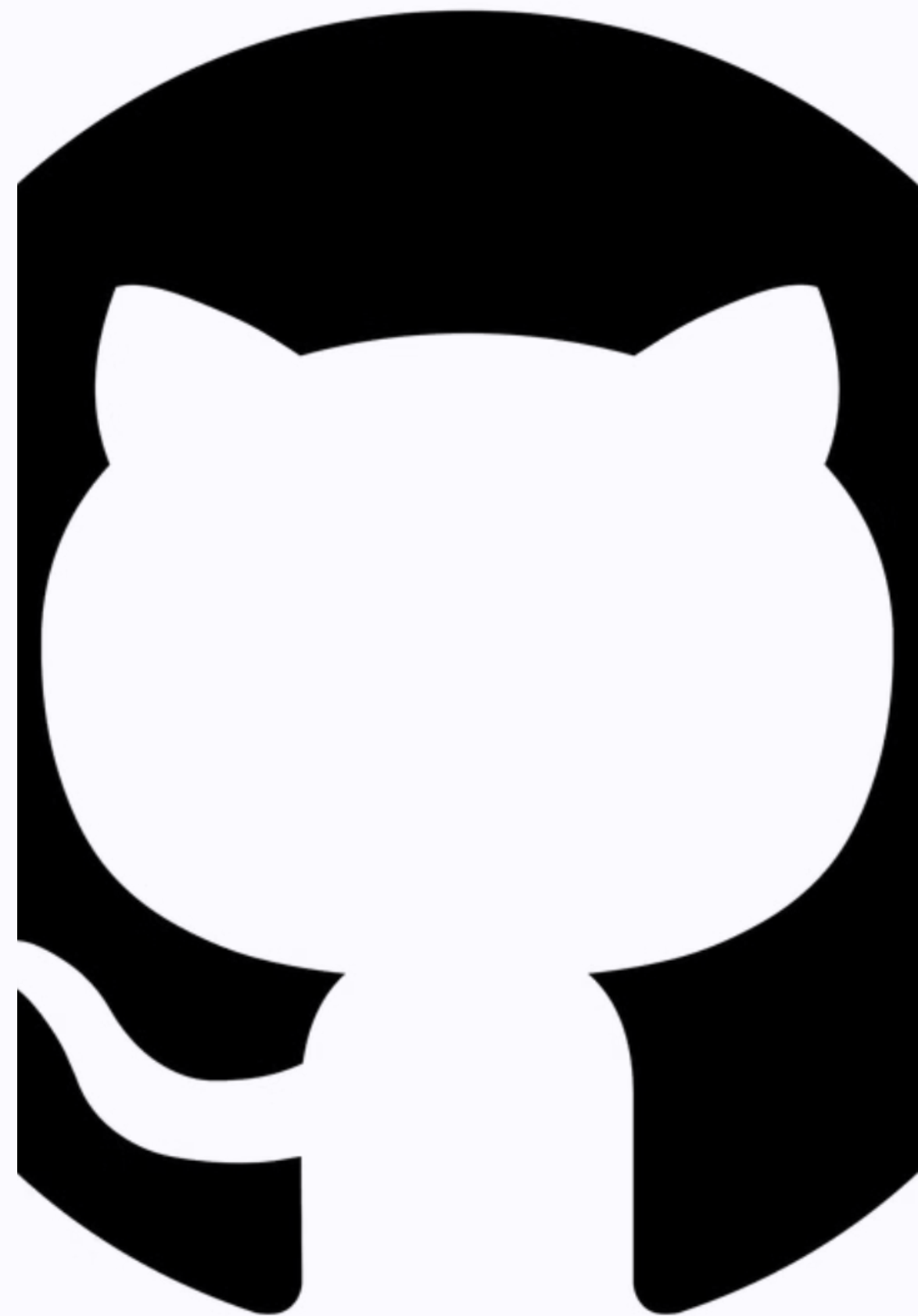


Démonstration de l'Application



Fichiers Sources

Les fichiers sources de l'application sont disponibles sur le [dépôt GitHub](#). Vous y trouverez le code de l'application ReactJS et l'API Express





Conclusion et Perspectives

Cette application ReactJS et Express offre une solution sécurisée et interactif. Nous avons mis en place des mesures de sécurité, comme les tokens les CSP, les CORS ou encore les jetons CSRF .. assurant la protection des données. Le système est testé et prêt pour une utilisation fiable et évolutive de la création de véhicules.