

Documentation Technique

Aperçu

Ce document fournit une vue d'ensemble technique du projet d'application web Garage V. Parrot, y compris l'architecture, les technologies utilisées, et les procédures de déploiement.

Architecture de l'Application

L'application est structurée en suivant une architecture modèle-vue-contrôleur (MVC) pour séparer la logique métier de l'interface utilisateur, facilitant ainsi la maintenance et l'évolutivité.

Backend

- **Langage de programmation** : Python 3.8
- **Framework web** : Flask
- **Base de données** : PostgreSQL
- **Authentification** : Flask-Login pour la gestion des sessions utilisateurs

Frontend

- **HTML5, CSS3** : pour la structure et le style de base
- **Bootstrap** : pour un design réactif et mobile-first
- **JavaScript** : pour l'interactivité côté client

Sécurité

Des mesures de sécurité ont été implémentées pour protéger l'application contre les vulnérabilités courantes :

- **SQL Injection** : Utilisation d'ORM SQLAlchemy pour toutes les requêtes de base de données
- **Cross-Site Scripting (XSS)** : Échappement automatique des entrées utilisateur dans les templates Flask
- **Cross-Site Request Forgery (CSRF)** : Protection CSRF activée par défaut dans Flask-WTF

Déploiement

L'application est déployée sur Heroku, une plateforme cloud PaaS qui facilite la mise en ligne rapide d'applications web :

1. **Configuration d'Heroku** : Création d'une application Heroku et configuration des variables d'environnement.
2. **Base de données** : Utilisation de l'addon Heroku PostgreSQL.
3. **Déploiement** : Utilisation de Git pour pousser le code sur Heroku.