**PROJET DJANGO**

**THEME:** Système de gestion des infrastructures publics

|  |  |
| --- | --- |
| **Binôme1** | GOUBA Georgette |
| **Biôme2** | SAWADOGO Nathalie |

**1. Introduction**

**Présentation générale du projet**

Le **Système de suivi des infrastructures publiques** est une application web conçue pour améliorer la gestion et la maintenance des infrastructures publiques au Burkina Faso. Ce système permet aux citoyens de signaler des pannes et aux autorités locales de gérer ces signalements et de planifier les interventions nécessaires.

**Importance de la gestion des infrastructures au Burkina Faso**

Le Burkina Faso fait face à des défis croissants en matière de gestion des infrastructures publiques, tels que les routes, les bâtiments, et les installations publiques. Un système de suivi des pannes et de gestion des interventions est crucial pour assurer un développement durable et une utilisation optimale des infrastructures existantes.

**2. Contexte du projet**

**Problématiques dans la gestion des infrastructures**

Les infrastructures publiques souffrent souvent de pannes et de dégradations non signalées ou mal suivies. Cela peut entraîner une inefficacité dans la maintenance, des coûts supplémentaires et une dégradation rapide des infrastructures.

**Justification de la nécessité d’un tel système**

Un **système centralisé** de gestion des pannes permettrait d'améliorer la réactivité des municipalités, d'optimiser la maintenance et de réduire les coûts. Ce système permet également aux citoyens de participer activement à la gestion des infrastructures en signalant les pannes rapidement et facilement.

**3. Objectifs du projet**

**Objectifs généraux**

Le projet a pour objectif principal **d'améliorer la gestion des infrastructures publiques** en fournissant un outil de suivi des pannes et de planification des interventions de maintenance.

**Objectifs spécifiques**

* **Suivi des pannes** : Permettre aux citoyens de signaler les pannes des infrastructures via un formulaire simple.
* **Planification des interventions** : Permettre aux municipalités de gérer les pannes signalées et de planifier les interventions de manière efficace.
* **Amélioration de la réactivité** : Réduire le délai entre le signalement d'une panne et son traitement par les autorités compétentes.

**4. Analyse des besoins**

**Identification des besoins des municipalités et des gestionnaires**

Les municipalités ont besoin d'un système leur permettant de suivre en temps réel les pannes signalées sur les infrastructures sous leur responsabilité. Elles doivent également pouvoir planifier et prioriser les interventions en fonction de la gravité des pannes.

**Analyse des utilisateurs cibles**

Les **administrations publiques** et les **citoyens** sont les principaux utilisateurs du système. Les citoyens peuvent signaler des pannes, tandis que les administrations sont responsables de gérer ces signalements et d'organiser les interventions.

**Étude de l’environnement actuel**

Les infrastructures publiques sont souvent gérées manuellement ou avec des systèmes décentralisés, ce qui rend difficile la gestion des pannes. Il n'existe pas encore de système efficace de suivi en temps réel qui inclut la participation des citoyens.

**5. Conception et architecture**

**Choix des technologies utilisées**

* **Langage de programmation** : Python.
* **Framework web** : Django.
* **Base de données** : PostgreSQL.
* **Interface utilisateur** : HTML, CSS, et Bootstrap pour une interface moderne et réactive.

**Architecture du système**

Le système est basé sur une architecture **client-serveur**. Le client (citoyen ou administrateur) interagit avec l'application via une interface web, tandis que le serveur gère les signalements, les interventions, et la base de données.

**Modélisation de la base de données**

* **Table Infrastructure** : Contient des informations sur les infrastructures publiques.
* **Table Panne** : Enregistre les pannes signalées pour chaque infrastructure.

**6. Implémentation**

**Étapes de développement**

Le développement du projet s'est déroulé en plusieurs étapes :

1. **Création des modèles** pour gérer les infrastructures et les pannes.
2. **Développement des vues** permettant de signaler les pannes et de gérer les interventions.
3. **Mise en place des formulaires** pour les utilisateurs (citoyens) et les administrateurs.

**Présentation des fonctionnalités développées**

* **Signalement des pannes** : Un formulaire permet aux citoyens de signaler des pannes en précisant l'infrastructure concernée et les détails du problème.
* **Suivi des interventions** : Les administrateurs peuvent consulter la liste des pannes signalées et planifier des interventions.

**7. Interface utilisateur**

**Conception de l’interface utilisateur**

L'interface est divisée en deux parties :

* **Interface pour citoyens** : Un formulaire simple pour signaler une panne.
* **Interface pour administrateurs** : Une interface de gestion des infrastructures et des pannes.

**Explication des choix de design**

Le design a été pensé pour être **simple** et **intuitif**, avec des boutons clairs et des formulaires faciles à remplir. L'utilisation de Bootstrap permet d'assurer une mise en page responsive, adaptée aux ordinateurs et aux mobiles.

**8. Suivi et maintenance**

**Gestion des signalements et priorisation des interventions**

Le système permet aux administrateurs de **prioriser les interventions** en fonction de la gravité des pannes signalées. Les signalements sont classés par ordre de priorité pour une meilleure gestion des ressources.

**Outils pour planifier les interventions**

Un tableau de bord est intégré pour suivre les interventions en cours, les signalements résolus, et ceux en attente de traitement.

**9. Évaluation des performances**

**Tests de performance et sécurité**

Des tests ont été effectués pour vérifier la capacité du système à gérer un grand nombre de signalements simultanés. Des mécanismes de sécurité ont également été intégrés pour protéger les données sensibles.

**Résultats des tests et corrections apportées**

Les tests ont révélé une bonne performance sous charge modérée. Quelques ajustements ont été apportés pour améliorer la gestion des données et l'optimisation des requêtes.

**10. Conclusion**

**Résumé des réalisations**

Le projet a permis de créer un système de suivi des infrastructures publiques, capable de gérer les signalements de pannes par les citoyens et de faciliter la planification des interventions par les administrations.

**Suggestions d'améliorations futures**

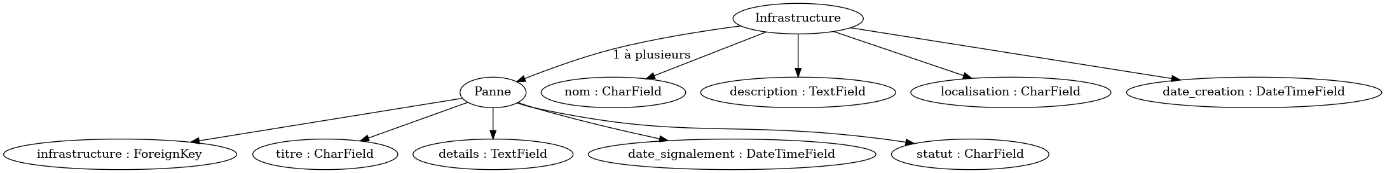
Pour les futures versions, il serait intéressant d'intégrer des **notifications en temps réel** pour informer les citoyens de l'état des interventions. De plus, l'ajout d'une **carte interactive** pourrait permettre de visualiser les infrastructures et les pannes signalées.

**Annexe**

**1. Diagrammes de modélisation**

**Diagramme UML des Modèles**

Voici un exemple d'un diagramme UML représentant les relations entre les modèles **Infrastructure** et **Panne**. Ce diagramme montre comment les pannes sont liées aux infrastructures.

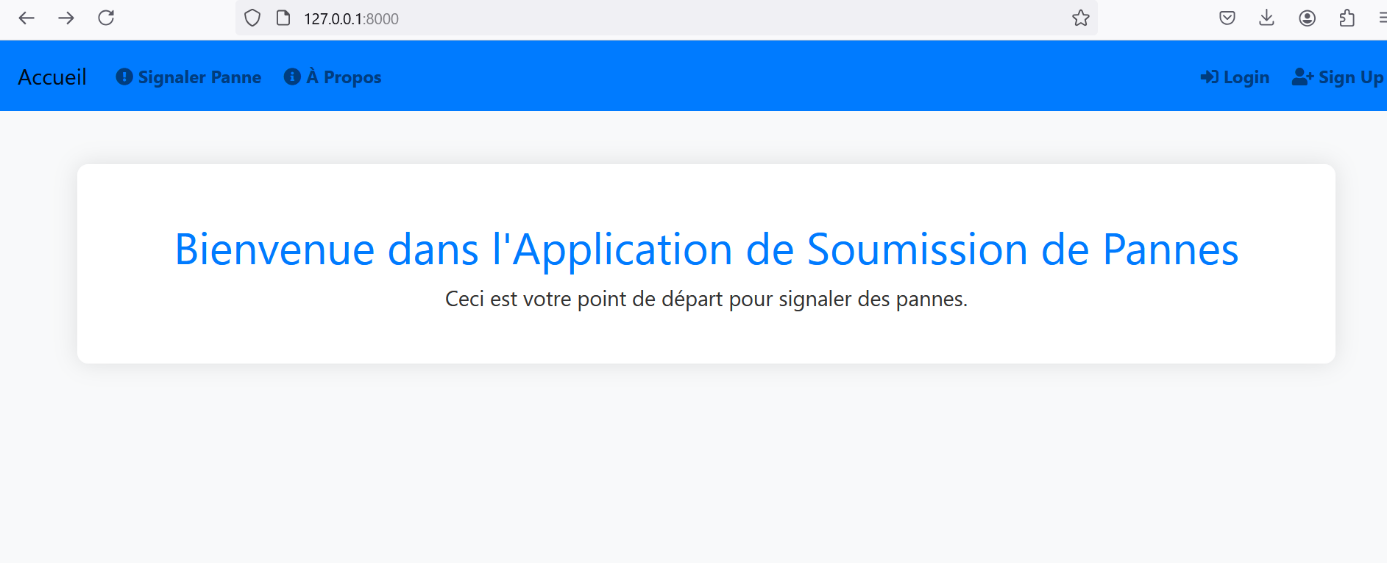
* **Infrastructure**
  + nom : CharField
  + description : TextField
  + localisation : CharField
  + date\_creation : DateTimeField
* **Panne**
  + infrastructure : ForeignKey vers Infrastructure
  + titre : CharField
  + details : TextField
  + date\_signalement : DateTimeField
  + statut : CharField
* 

**2. Documentation complémentaire**

* **Installation de l'application** :
  1. Cloner le projet : git clone https://github.com/SG\_infrastructure
  2. Installer les dépendances : pip install -r requirements.txt
  3. Configurer la base de données PostgreSQL.
  4. Appliquer les migrations : python manage.py migrate
  5. Lancer le serveur de développement : python manage.py runserver
* **Prérequis** :
  1. Python 3.9+
  2. Django 5.x
  3. PostgreSQL 13+

**Instructions pour ajouter des captures d'écran :**

**Page d’accueil**

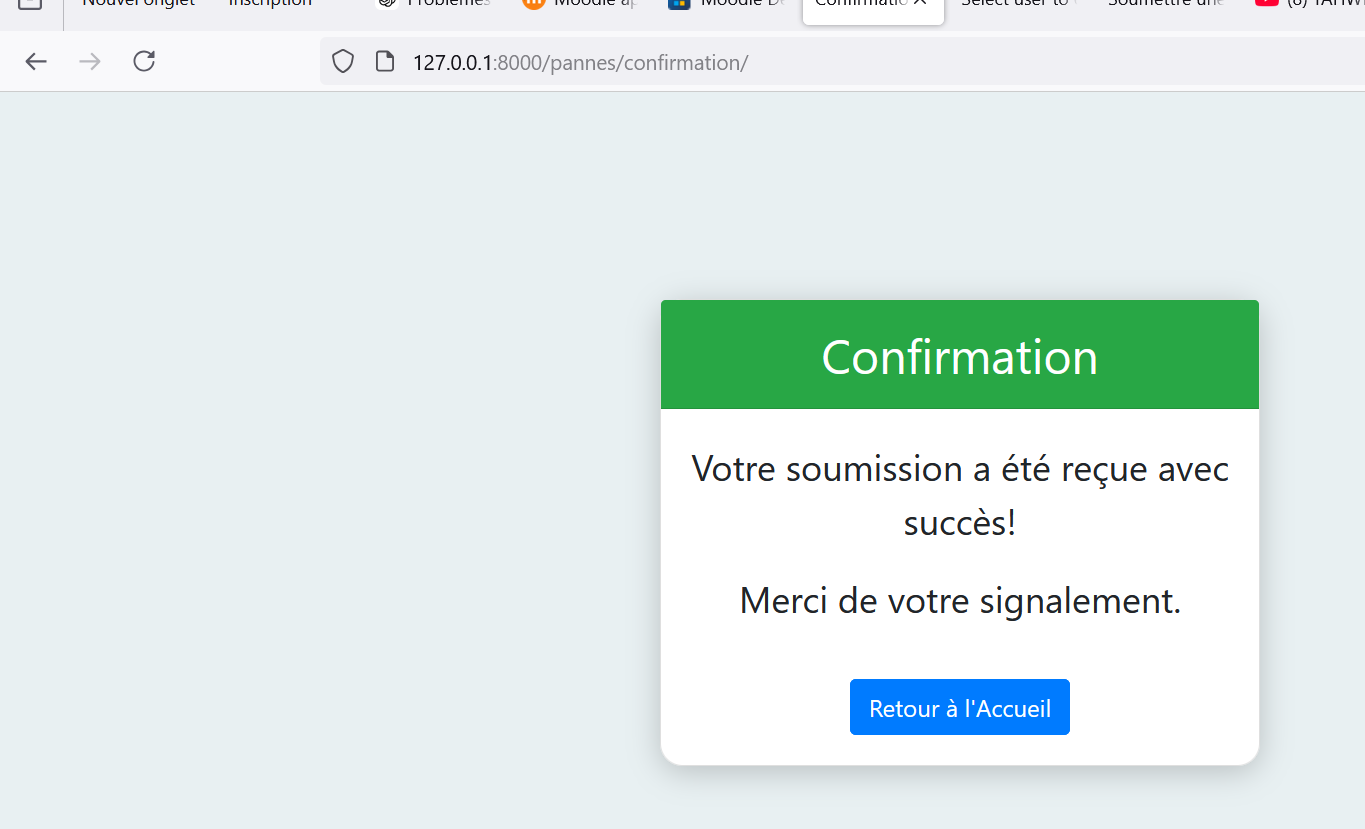


**Interface utilisateur (Signalement de Panne)**

L'interface utilisateur pour signaler une panne est conçue de manière simple et intuitive.

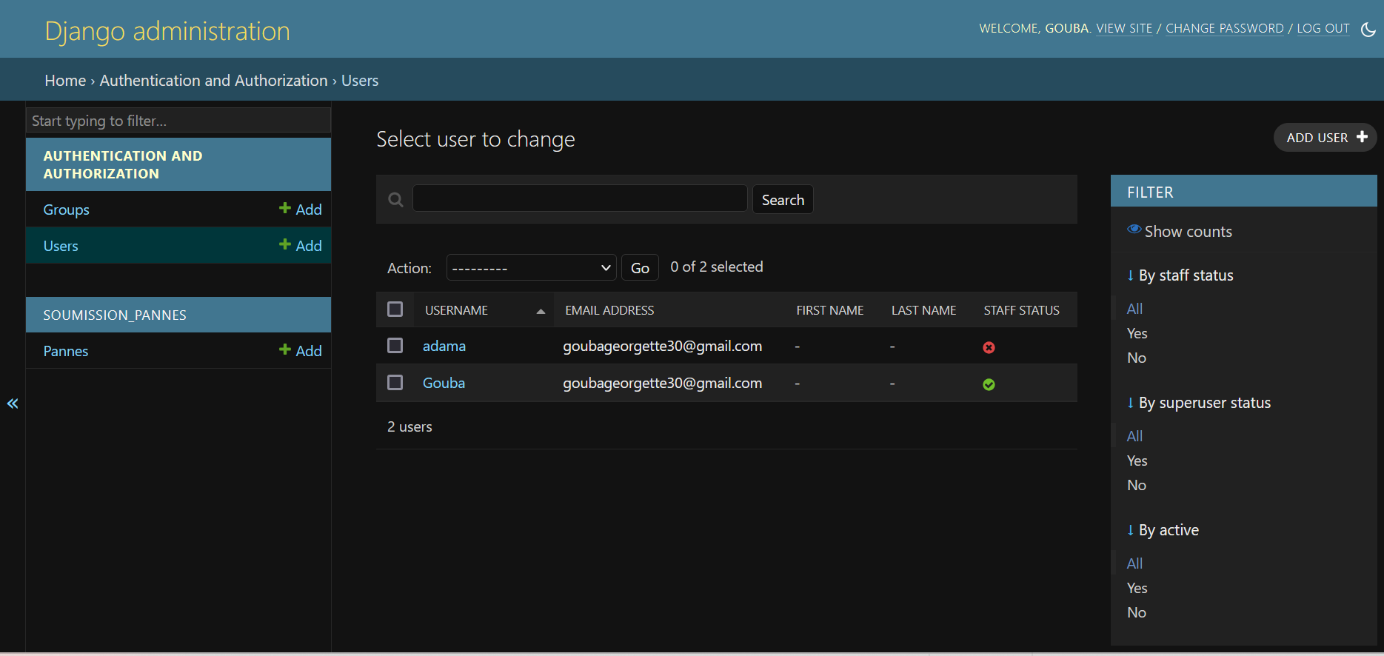


**Confirmation**



**Gestion des Pannes (Interface Administrateur)**

L'interface administrateur permet de visualiser les pannes signalées et de gérer les interventions.



Voici notre lien GitHub