

# Assistant Urbanisme POC

Un assistant IA pour répondre aux questions sur les réglementations d'urbanisme (PLU, zonage, hauteurs autorisées, etc.)

## Démarrage rapide

### Prérequis

- Python 3.8+
- Node.js (pour Vercel)
- Redis (optionnel)

### Installation

#### 1. Cloner le repository

bash

```
git clone https://github.com/yourusername/urbanisme-poc.git
cd urbanisme-poc
```

#### 2. Backend

bash

```
cd backend
python -m venv venv
source venv/bin/activate # Windows: venv\Scripts\activate
pip install -r requirements.txt
```

#### 3. Variables d'environnement

bash

```
cp .env.example .env
# Éditer .env avec vos clés API
```

### Lancement local

#### Backend:

bash

```
cd backend
uvicorn main:app --reload
```

## Frontend:

```
bash
```

```
cd frontend
```

```
python -m http.server 3000
```

Ouvrir <http://localhost:3000>



## Structure du projet

```
urbanisme-poc/
├── backend/
│   ├── main.py          # API FastAPI
│   ├── requirements.txt # Dépendances Python
│   ├── .env.example     # Template variables d'env
│   └── railway.json     # Config Railway
├── frontend/
│   ├── index.html       # Interface utilisateur
│   ├── app.js           # Logique frontend
│   └── vercel.json       # Config Vercel
├── tests/
│   └── test_load.py     # Tests de charge
└── README.md
```



## Déploiement

### Backend (Railway)

1. Push sur GitHub
2. Connecter Railway au repo
3. Ajouter les variables d'environnement
4. Deploy!

### Frontend (Vercel)

1. Installer Vercel CLI: `npm i -g vercel`
2. Dans `/frontend`: `vercel`
3. Suivre les instructions



## Tests

```
bash
```

```
cd tests
```

```
python test_load.py
```



## Roadmap

- ☒ POC basique
- ☒ Cache Redis
- ☒ Déploiement cloud
- ☐ Intégration vraie API GPU
- ☐ Ajout IA/RAG
- ☐ Authentification
- ☐ Dashboard analytics



## Contribution

Les PRs sont les bienvenues !



## License

MIT