

Compte-rendu hebdomadaire - stage SIO

BESNIER Alexandre

5e semaine

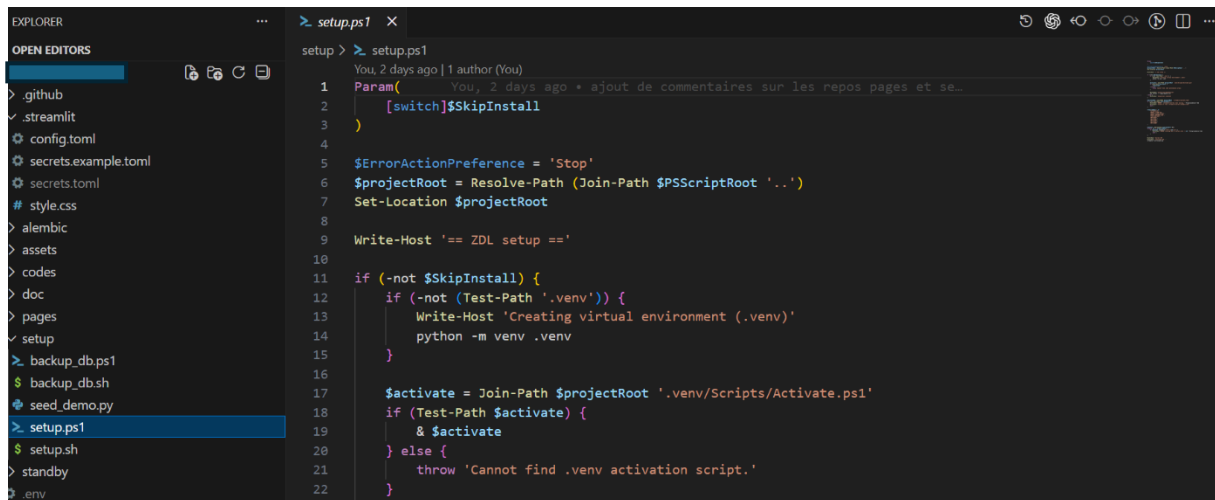
Description de mission :

Mission principale :

Réaliser la **documentation de référence** du projet pour une **passation** : permettre à un repreneur (technique et non technique) de comprendre rapidement l'application, la maintenir, la déployer et l'exploiter, en sécurisant particulièrement les zones sensibles (auth, RBAC, RGPD, migrations, modules stocks/finance).

Travail réalisé :

Développement de Scripts Bash pour le premier lancement de l'application (avec une connexion à un compte configuré la création des tables et leur remplissage avec des données par défaut) :

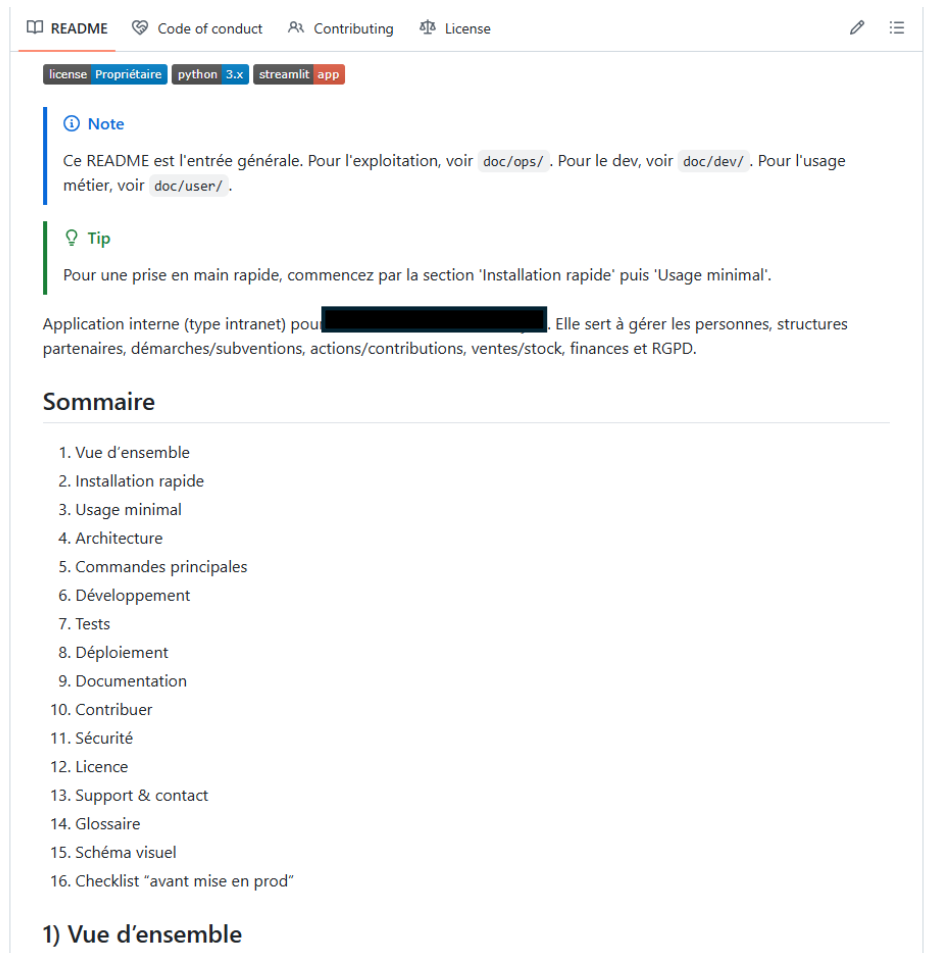


```
1 Param(
2     [switch]$SkipInstall
3 )
4
5 $ErrorActionPreference = 'Stop'
6 $projectRoot = Resolve-Path (Join-Path $PSScriptRoot '..')
7 Set-Location $projectRoot
8
9 Write-Host '== ZDL setup =='
10
11 if (-not $SkipInstall) {
12     if (-not (Test-Path '.venv')) {
13         Write-Host 'Creating virtual environment (.venv)'
14         python -m venv .venv
15     }
16
17     $activate = Join-Path $projectRoot '.venv/Scripts/Activate.ps1'
18     if (Test-Path $activate) {
19         & $activate
20     } else {
21         throw 'Cannot find .venv activation script.'
22     }
23 }
```

Documentation utilisateur pour l'application Streamlit afin de réaliser une passation de connaissance au client.

- Mise en place d'un **point d'entrée unique** : `doc/README.md`
→ orientant vers 3 blocs : **Produit (users) / Engineering (dev) / Exploitation (ops)**

Affichage dans GitHub :



license Propriétaire python 3.x streamlit app

Note

Ce README est l'entrée générale. Pour l'exploitation, voir `doc/ops/`. Pour le dev, voir `doc/dev/`. Pour l'usage métier, voir `doc/user/`.

Tip

Pour une prise en main rapide, commencez par la section 'Installation rapide' puis 'Usage minimal'.

Application interne (type intranet) pour [REDACTED]. Elle sert à gérer les personnes, structures partenaires, démarches/subventions, actions/contributions, ventes/stock, finances et RGPD.

Sommaire

1. Vue d'ensemble
2. Installation rapide
3. Usage minimal
4. Architecture
5. Commandes principales
6. Développement
7. Tests
8. Déploiement
9. Documentation
10. Contribuer
11. Sécurité
12. Licence
13. Support & contact
14. Glossaire
15. Schéma visuel
16. Checklist "avant mise en prod"

1) Vue d'ensemble

Création de dossiers de documentation dans le projet GitHub :

2) Documentation “Utilisateur” (doc/user/...)

- Guide utilisateur synthétique : doc/user/guide_utilisateur.md (parcours principaux)
- FAQ : doc/user/faq.md
- Glossaire : doc/user/glossaire.md
- Documentation par module : doc/user/modules/... avec un format homogène (objectif, accès/roles, pages, actions, règles/erreurs)

3) Documentation “Développeur” (doc/dev/...)

- architecture.md : vue en couches + schéma système (Streamlit Cloud ↔ Supabase) + points d’attention (secrets, RGPD, migrations)
- development.md : description concernant : l’onboarding, le lancement local, la migrations Alembic, la seed démo et les scripts de setup
- configuration.md : liste complète des secrets (OAuth, DB, SMTP, chiffrement) + comment les générer
- data_model.md : résumé des entités et relations métier clés
- testing.md : checklist de tests manuels minimaux (login, invitation, RGPD, exports...)

4) Documentation “Exploitation / Prod” (doc/ops/...)

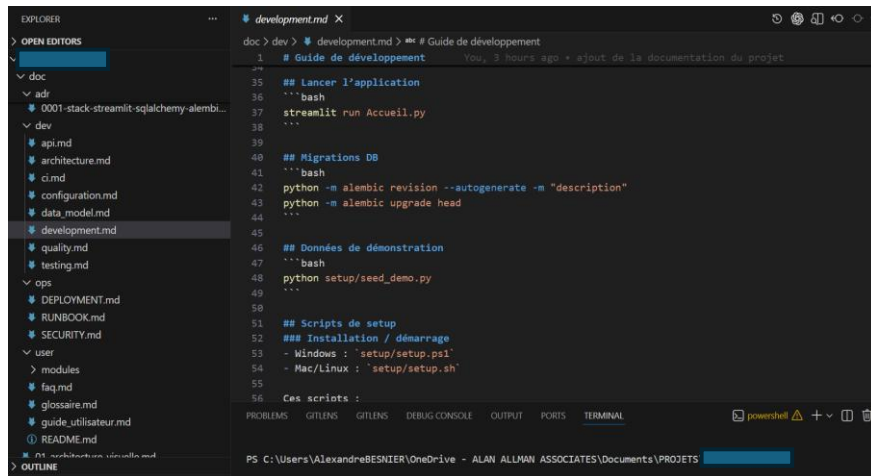
- DEPLOYMENT.md : process de déploiement, migrations, rollback
- RUNBOOK.md : vérifs post-déploiement, sauvegardes/restauration, gestion d’incident, escalade
- SECURITY.md : politique de signalement vulnérabilité + règles de prudence (données sensibles / RGPD)

5) Références “métier + support” (racine /doc)

- 01_architecture_visuelle.md : schémas rapides + flux “Démarche → Subvention”
- 02_roles_et_droits.md : tableau rôle → pages (RBAC) + méthode de modification
- 03_checklist_processus_metier.md : synthèse des processus par module (quoi / qui / ce qui est enregistré)
- 06_incidents_frequents.md : pannes types (secrets, OAuth redirect, Alembic...) + correctifs rapides

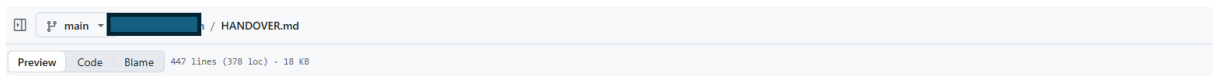
Livrables principaux (Semaine 5)

- Dossier doc/ à la racine du projet complet, structuré, navigable et actionnable constitué de fichiers textes en .md (avec des sous dossiers users/dev/ops + leurs références) :



```
1 # Guide de développement
27
35 ## Lancer l'application
36 ```bash
37 streamlit run Accueil.py
38 ```
39
40 ## Migrations DB
41 ```bash
42 python -m alembic revision --autogenerate -m "description"
43 python -m alembic upgrade head
44 ```
45
46 ## Données de démonstration
47 ```bash
48 python setup/seed_demo.py
49 ```
50
51 ## Scripts de setup
52 ### Installation / démarrage
53 - Windows : 'setup/setup.ps1'
54 - Mac/Linux : 'setup/setup.sh'
55
56 Ces scripts :
```

- fichier 'HANDOVER.md', document général de passation avec méthode de lecture/modification et points de vigilance pour les fichiers du projet (notamment via le guide de reprise et les conventions de développement) :



0.13 Guide reprise technique (résumé)

- Démarrage local : `streamlit run Accueil.py`
- Démarrage prod : déploiement via Streamlit Community Cloud (voir `doc/ops/DEPLOYMENT.md`)
- Scripts de démarrage : `setup/setup.ps1`, `setup/setup.sh`
- Secrets : `.streamlit/secrets.toml` et Streamlit Cloud Secrets
- DB : Supabase (PostgreSQL), vérifier `DATABASE_URL`
- Migrations : `python -m alembic upgrade head`
- Sauvegardes : `setup/backup_db.ps1`, `setup/backup_db.sh`
- Données de démonstration : `python setup/seed_demo.py`

1) Lire les commentaires "contrats" (méthode)

L'application utilise des commentaires structurés pour rendre le code lisible par des non-dev et pour sécuriser les changements. On les trouve surtout dans `pages/`, `codes/repositories/` et `codes/services/`.

1.1 Les mots-clés et leur sens

- **Objectif** : intention métier de la fonction ou du bloc (le "pourquoi").
- **Contexte** : rappel de l'usage ou du périmètre.
- **Flow** : un flux fonctionnel (enchaînement logique d'actions). Souvent suivi de préconditions, effets et postconditions.
- **Contrat** : règles "formelles" de la fonction. Ce qui doit être vrai avant, pendant, après.
- **Précondition** : ce qui doit être vrai avant d'appeler la fonction ou de déclencher l'action UI.
- **Invariant** : règle qui doit rester vraie, même après modification (si vous la cassez, vous cassez le métier).
- **Postcondition** : ce qui doit être vrai après exécution (DB/UI/Audit).
- **Effet (DB/UI/Audit)** : impact attendu sur la base de données, l'interface ou les logs.
- **Risque** : scénario dangereux ou fragile.
- **Barrière** : protection mise en place contre ce risque (UI, service, repo, contrôle d'accès).
- **Note** : point d'attention (performance, compatibilité, dette technique).
- **Règle** : règle métier ou UI qui doit rester stable.
- **TODO(...)** : amélioration prévue ou dette technique. Traiter avec prudence.

Seconde mission :

Réaliser l'audit et l'amélioration du site web d'un client en utilisant les comptes administrateurs du CMS et de l'outil de mesure d'audience du dit client.

- Audit du SEO du site du client B (rapport de 9 pages comprenant des indications techniques, des graphiques, réalisés via scripts Python, d'analyse des données de visites du site). extrait:



- Pages du site du client B modifiées et/ou complétées pour répondre à ses attendus de SEO (gestion des pages et des modules du site via le CMS Drupal) :

Retour au site | Gérer | Raccourcis | Besnier | Admin Toolbar Search (Alt+a)

Contenu | Structure | Apparence | Extension | Configuration | Personnes | Rapports | Aide

Choisir quelle boîte à outils image utiliser si vous avez installé des boîtes à outils optionnelles.

Recherche et metadonnées

- Alias d'URL**
Ajouter des URL personnalisées aux chemins existants.
- Metatag**
Configurer les balises meta par défaut.
- URL redirigées**
Redirigez les utilisateurs d'une URL vers une autre.

Services Web

- Google Tag**
Configure the website integration with Google Tag and Google Tag Manager.
- Publication RSS**
Configurer la description du site, le nombre d'éléments par flux et si les flux devraient être des titres / accroches / textes complets.
- Menu Linkset Settings**
Enable or disable the menu linkset endpoint.

Region and language

- Paramètres régionaux**
Configurer les paramètres régionaux et fuseaux horaires.
- Formats de date et d'heure**
Configurer l'affichage des dates et heures.
- Langues**
Configurer les langues utilisées pour le contenu, l'interface, et la configuration.
- Langue du contenu et traduction**
Configurer la prise en charge de la langue et de la traduction pour le contenu.
- Traduction de l'interface utilisateur**
Configurer l'importation des fichiers de traduction, et ajouter ou personnaliser les traductions de l'interface utilisateur.
- Traduction de la configuration**
Traduire la configuration.

Sécurité

Outils utilisés

- Git/GitHub (documentation)
- Dépendances du projet (Streamlit, SQLAlchemy/Alembic, Python, Bash)
- Google Cloud Authentication (Plateforme de gestion du service de reconnaissance par compte Google)
- Supabase (Gestion de base de donnée PostgreSQL en ligne)
- Drupal (CMS)
- Matomo (plateforme de mesure d'audience pour site internet concurrente à Google Analytics)

Compétences mobilisées

- **1.2** Réponse aux demandes d'assistance et d'évolution (évolution de l'application à la suite de demandes du client A sur la page Dashboard des activités des bénévoles)
- **1.3** Développer la présence de l'organisation (analyse de l'activité d'un site client B, audit du SEO du site et refonte de pages pour améliorer le SEO)
- **1.4** Travail en mode projet (planifications d'étapes à suivre dans le sprint au cours du projet de modification du SEO)
- **1.5** Mise à disposition des utilisateurs d'un service informatique (documentation d'un projet GitHub, support utilisateur de l'application et réalisation de test techniques)