

Compte-rendu hebdomadaire - stage SIO

BESNIER Alexandre

2e semaine

Description de mission :

Mission principale :

Travail réalisé en équipe de deux.

Le travail principal de la semaine a consisté à migrer l'outil de gestion développé initialement en VBA/Excel vers une application web développée en Python avec Streamlit afin de répondre aux besoins du client recensés pendant la réunion de la semaine 1.

Cette migration a nécessité : la compréhension du fonctionnement de l'outil VBA existant, la transposition des formulaires Excel en formulaires web, l'adaptation des traitements métiers vers une architecture plus modulaire

Streamlit : un Framework Python qui permet de créer rapidement des applications web interactives pour saisir, consulter et analyser des données, sans avoir à développer un site web complexe. Accessible via navigateur, il offre une interface simple divisée en pages qui comprennent des formulaires, tableaux et graphiques. Dans ce projet, Streamlit a été choisi pour remplacer VBA, pour proposer une solution plus moderne, facile à évoluer et utiliser.

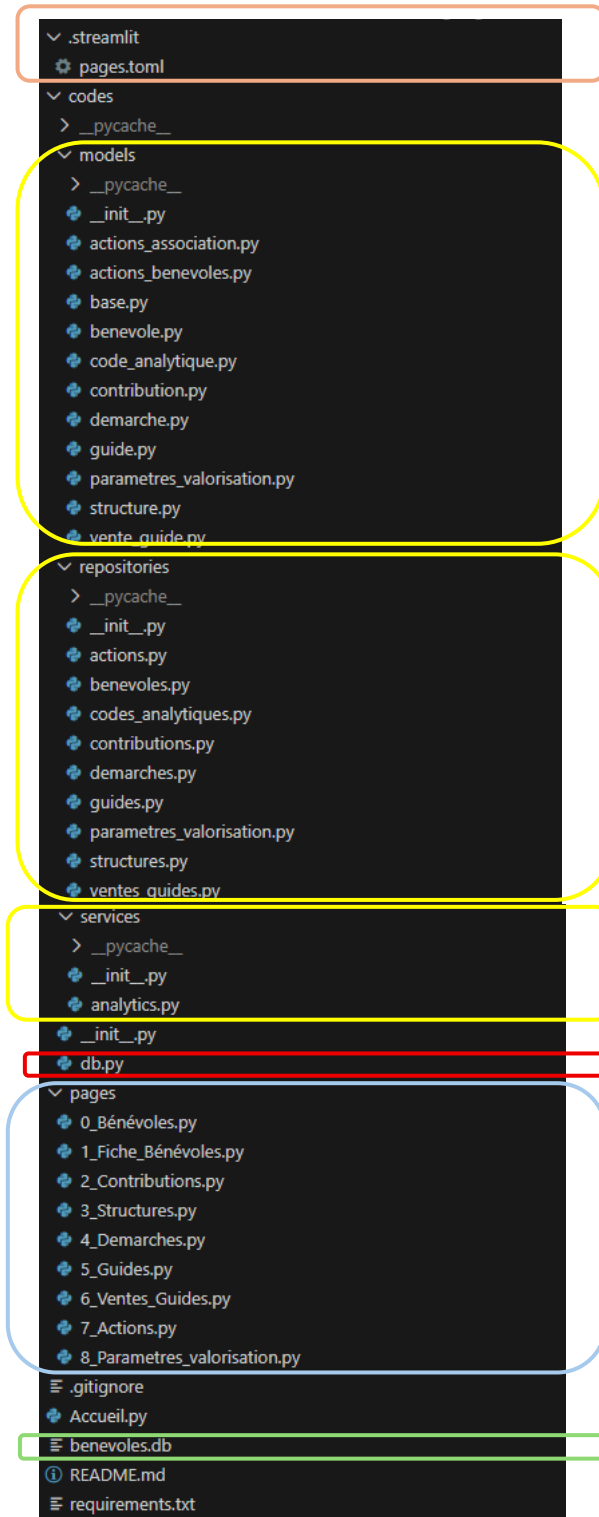
Travail réalisé :

- Création d'un projet Streamlit
- Analyse et redéfinition de la structure de la base de données.
- Création des différents modèles et fonctions de connexions et pages Streamlit.
- Portage des fonctionnalités existantes en VBA vers Streamlit.
- Création de fonctionnalités supplémentaires : Dashboard, gestion avancée des contributions des bénévoles.

Outils :

Git(projet séparé en branches), Github(kanban et user stories), Framework Streamlit(sur lequel l'application est développée)

Structure du Projet :



Le dossier **.Streamlit** est le dossier de configuration du projet, dedans les fichiers de configurations vont avoir des effets sur la mise en page de l'application web

Le dossier `/codes/models` contient les modèles de données de l'application correspondant chacun à une table de la base de données Les modèles assurent la liaison entre le code python et la base de données SQL grâce à l'ORM SQLAlchemy.

Le dossier `/codes/repositories` regroupe l'ensemble des fonctions permettant d'accéder à la base de données (lire, insérer, mettre à jour, supprimer). Il sert d'intermédiaire entre les modèles de données et le reste de l'application.

Les fonctions utilisent :

- les modèles SQLAlchemy définis dans le dossier `models`
- des sessions SQLAlchemy pour exécuter les requêtes
- des requêtes construites avec l'API Python de SQLAlchemy (select, where, etc.)

Le dossier `/codes/services` est une couche intermédiaire entre les **repositories** et les **pages** Streamlit. Il contient des fonctions d'analyse qui servent pour la mise en forme de Dashboard.

Appelé dans la page **Accueil.py** (page d'accueil affichée au démarrage de l'application), **db.py** contient la fonction de création et connexion de la base de données au lancement de l'application.

Le dossier `/pages` contient les différentes pages de l'application, elles sont codées en python et appellent les différentes fonctions du dossier `/codes/repositories` pour traiter et afficher les données de la bdd.

La base de données du projet

Les fichiers `__init__.py` sont les fichiers de configuration des fonctions, ils permettent d'importer les fonctions d'un dossier vers un autre

Cette architecture en couches (`models / repositories / services / pages`) permet de séparer clairement : la logique métier, l'accès aux données, l'affichage dans l'interface utilisateur. Ce choix améliore : la maintenabilité du code, la lisibilité du projet, la facilité d'ajout de nouvelles fonctionnalités.

```

02_Contributions.py M X
pages > 02_Contributions.py > ...
296
297 st.divider()
298 st.subheader("📄 Contributions enregistrées")
299
300 contribs = select_contributions(engine)
301
302 labels = []
303 for c in contribs:
304     perso = is_valorisation_personnalisee(engine, c)
305
306     labels.append(
307         f"{c.date:%d/%m/%Y} 📅 "
308         f"{c.benevole.name} {c.benevole.suriname} 🧑 "
309         f"{c.activite} - "
310         f"{c.code_analytique} "
311     )
312
313 selected = st.selectbox(
314     "Sélectionner une contribution",
315     labels,
316     index=None
317 )

```

Cet extrait de code correspond à la partie « consultation et sélection des contributions » de l'application web. La fonction `select_contributions()` (stockée dans le dossier `repositories`) permet de récupérer **l'ensemble des contributions enregistrées**.

Chaque contribution est ensuite transformée en une étiquette textuelle lisible par l'utilisateur. Ces étiquettes sont construites dynamiquement à partir de plusieurs champs :

- la date de la contribution,
- le nom et le prénom du bénévole concerné,
- l'activité réalisée,
- le code analytique associé.

Le composant Streamlit `selectbox` est utilisé afin de proposer une **liste déroulante** permettant à l'utilisateur de sélectionner une contribution précise.

Aucune sélection n'est imposée par défaut (`index=None`), ce qui évite toute modification involontaire.

Cela permet une navigation simple et intuitive dans les données tout en maintenant un lien direct entre l'interface utilisateur et les objets métiers issus de la base de données.

Visuel sur la page :

The screenshot shows the application's user interface. On the left is a sidebar with a 'censuré' header and several menu items: 'Gestion', 'Gestion des bénévoles', 'Gestion des contributions', 'Gestion des structures', 'Démarchage des structures', 'Gestion des guides', and 'Ventés des guides'. The main content area has a title 'Contributions enregistrées' with a subtitle 'Le titre'. Below the title is a dropdown menu labeled 'Sélectionner une contribution' containing three items: '15/11/2024 - ALEXANDRE TEST - dev en b - Animation', '15/11/2024 - ALEXANDRE TEST - dev en b - Animation', and '30/01/2020 - ALEXANDRE TEST - test - Fonctionnement'. Below the dropdown is a form for the selected contribution, showing 'Heures' (1,50), 'Type valorisation' (Base), and 'Code analytique' (Animation). A blue box displays 'Valorisation : 17.80 €'. At the bottom, there are buttons for 'Enregistrer les modifications' and 'Supprimer la contribution', and a breadcrumb 'Données Contributions'.

Autres pages réalisée (front et back end) :

Création de formulaires CRUD pour chaque table de la BDD et développement de sécurités et de vérifications pour chaque champ des formulaires.

Exemple du formulaire d'ajout d'une contribution :

Contributions bénévoles

+ Nouvelle contribution

Bénévole *

Nom Prénom

Date *

2026/01/16

Type de valorisation *

Choose an option

Code analytique

Animation

Activité bénévole *

Nombre d'heures *

0,00

Coût horaire (€)

0,00

Valorisation financière : 0.00 €

Paramètres de valorisation

Enregistrer

Création de pages de dashboards dynamiques répondants aux attentes du client (données de test):

Fiche bénévole

Sélectionner un bénévole

ALEXANDRE TEST

Période d'analyse

Date de début

2020/01/01

Date de fin

2026/01/16

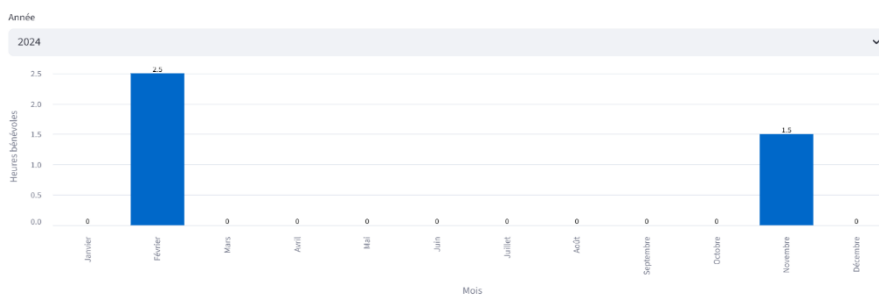
Informations

Nom : ALEXANDRE Prénom : TEST Statut : Actif.ve Ancienneté : 0 mois

Synthèse d'activité

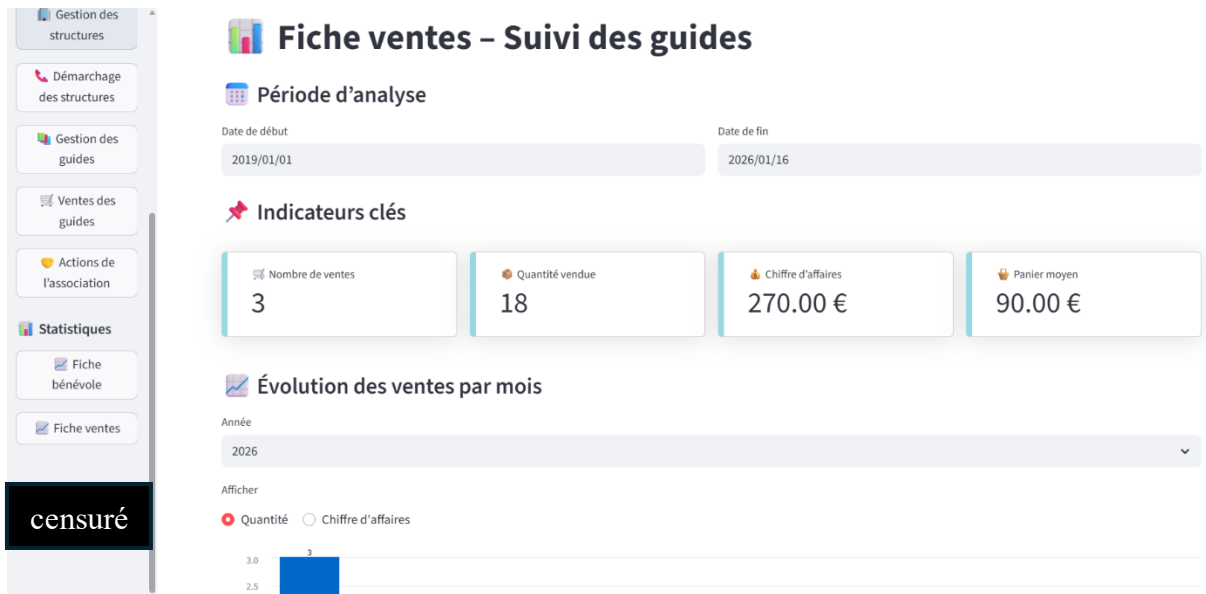
Heures	Actions	Moyenne / action
6.0 h	4	1.5 h

Évolution des heures par mois



Répartition par code analytique





Création du front et du back end d'une barre de navigation :

La barre de navigation est une fonction qui est appelée dans chaque pages :



```

You, 2 hours ago | 1 author (You)
1 import streamlit as st
2 from datetime import date
3 from codes.repositories import (
4     insert_contribution,
5     update_contribution,
6     select_contributions,
7     delete_contribution,
8     get_cout_horaire,
9     is_valorisation_personnalisee,
10    get_all_codes_analytiques,
11    insert_code_analytique,
12    select_all_benevoles_for_select,
13    is_code_analytique_used,
14    delete_code_analytique
15 )
16 from codes.db import engine
17 from codes.services import sidebar_nav
18 st.set_page_config("Contributions", layout="wide")
19 sidebar_nav()
  
```

Création du front et du back end d'une vue de gestion de la base de données sur le site du style phpMyAdmin (données d'entraînement) :

Avec un système dynamique de sélection de la table, de filtres, d'édition, d'insertion et d'ajout de données :

The screenshot shows a web interface for managing a database table. At the top, there is a dropdown menu to 'Sélectionner une table' (Select a table) with 'Démarches' selected. Below this, the 'Table SQL' is identified as 'démarches'. A 'Filtres' (Filters) section allows adding filters, with one active filter: 'Date_Demarche' with the operator 'plage' (range) and the date range '2026/01/15 - 2026/01/24'. Below the filters, the 'Résultats (10 ligne(s))' (Results) section shows a table with 10 rows. The table has columns: 'Nouveau', 'Supprimer', 'ID_DEMARCHE', 'Date_Demarche', 'Relance', 'ID_STRUCTURE', 'ID_BENEVOLE', 'Type_action', 'Commentaire', 'Pre_vente_guide', and 'Partenariat_long_terme'. The visible rows are:

Nouveau	Supprimer	ID_DEMARCHE	Date_Demarche	Relance	ID_STRUCTURE	ID_BENEVOLE	Type_action	Commentaire	Pre_vente_guide	Partenariat_long_terme
<input type="checkbox"/>	<input type="checkbox"/>	2	2026-01-16	None	5	1	Appel		<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	6	2026-01-16	None	5	1	Appel		<input type="checkbox"/>	<input type="checkbox"/>

Et une fonction d'exportation des données de la table en Excel :

The screenshot shows the 'Export Excel' section of the web application. It displays a table with 4 rows of data. The table has columns: 'ID_VENTE', 'Date_vente', 'ID_STRUCTURE', 'ID_GUIDE', 'ID_BENEVOLE', 'Quantite', 'Prix_unitaire', 'Montant_total', and 'Commentaire'. The visible rows are:

ID_VENTE	Date_vente	ID_STRUCTURE	ID_GUIDE	ID_BENEVOLE	Quantite	Prix_unitaire	Montant_total	Commentaire
1	2027-01-21		5	9	2	18	36	
6	2024-05-16		5	6	1	15	165	
7	2024-04-18		5	6	None	4	60	
8	2026-01-16		5	5	None	3	45	

Below the table, there is an 'Export Excel' section with a dropdown menu 'Exporter les données' (Export data) and two radio button options for the export mode: 'Vue affichée (filtres appliqués)' (Selected) and 'Table complète (page de lignes)'. A 'Générer l'export' (Generate export) button is also present.

Compétences mobilisées :

Mission 1 :

1.6 : Organisation de son développement professionnel

Mission 2 :

1.2 : Réponse aux incidents et aux demandes d'assistance et d'évolution

1.4 : Travail en mode projet

1.5 : Mise à disposition des utilisateurs d'un service informatique