

Rapport individuel SAE 3.01 Partie 3

Synthèse : Pendant la troisième partie de cette SAÉ nous nous sommes occupés de la partie front-end de l'application. Donc nous avons lié tout ce que nous avons fait avant, c'est-à-dire BD et partie algorithmique avec la partie front-end.

Analyse : Le plus dur lors de cette partie de SAE a été de faire le [views.py](#) en réussissant à bien lier l'application avec la BD ainsi que les algorithmes Python que nous avons fait. Par exemple, pour la fonction sequences_adn dans le [views.py](#), il y a un endroit où on peut ajouter des fichiers mais ça les écrasaient tout le temps.

Pour résoudre cela, nous avons utilisé secure_filename.

Voici la fonction finale :

```
@app.route('/sequences_adn/', methods=['GET', 'POST'])
@login_required
@role_required(UserRole.CHERCHEUR, UserRole.ADMIN)
def sequences_adn():
    form = EchantillonForm()
    form.campagne.choices = [(c.idCamp, c.nom) for c in Campagne.query.order_by(Campagne.nom).all()]

    if form.validate_on_submit():
        fichier = form.fichier_sequence.data
        if fichier:
            filename = secure_filename(fichier.filename)
            upload_folder = app.config['UPLOAD_FOLDER']
            if not os.path.exists(upload_folder):
                os.makedirs(upload_folder)
            fichier.save(os.path.join(upload_folder, filename))

            nouvel_echantillon = Echantillon(
                fichier_sequence=filename,
                commentaire=form.commentaire.data,
                idCamp=form.campagne.data
            )
            db.session.add(nouvel_echantillon)
            db.session.commit()
            flash(f'L'échantillon {filename} a été ajouté avec succès.', 'success')
        else:
            flash("Aucun fichier n'a été sélectionné.", 'danger')

    return redirect(url_for('sequences_adn'))
```

Cette partie m'a surtout permis de consolider mes compétences en Flask.

Démonstration de compétences :

AC21.01 | Élaborer et implémenter les spécifications fonctionnelles et non fonctionnelles à partir des exigences

Implémentation conforme : chaque exigence a sa route, son template et son formulaire.

AC21.02 | Appliquer des principes d'accessibilité et d'ergonomie

Utilisation de Bootstrap pour garantir une interface responsive et accessible.

Labels explicites pour chaque champ de formulaire.

Contrastes et tailles de police adaptés dans le CSS.

AC21.03 | Adopter de bonnes pratiques de conception et de programmation

Séparation des responsabilités : modèles, formulaires, vues, templates.

Utilisation de Flask-WTF pour la validation des formulaires.

Respect des conventions de nommage et structuration du projet.

AC22.01 | Choisir des structures de données complexes adaptées au problème

Utilisation de dictionnaires et listes pour manipuler les échantillons.

Modélisation relationnelle avec SQLAlchemy pour les entités

AC23.01 | Concevoir et développer des applications communicantes

Mise en place d'une application web client-serveur avec Flask.

Utilisation de routes pour la communication entre client et serveur .

AC23.02 | Utiliser des serveurs et des services réseaux virtualisés

Déploiement et test de l'application sur un serveur local.

Utilisation de l'environnement virtuel Python.

AC24.03 | Organiser la restitution de données à travers la programmation et la visualisation

Affichage des échantillons dans des tableaux HTML dynamiques.

Utilisation de Jinja2 pour la restitution des données dans les templates.

VIGNON CHAUDEY

Clément

AC24.04 | Manipuler des données hétérogènes

Chargement et adaptation de données YAML pour l'affichage.

Conversion des clés et adaptation des formats pour le front-end.

AC25.02 | Formaliser les besoins du client et de l'utilisateur

Recueil des besoins (consultation, ajout, filtrage, visualisation).

Traduction en fonctionnalités concrètes dans l'application.

AC25.03 | Identifier les critères de faisabilité d'un projet informatique

Analyse de la faisabilité technique (choix de Flask, Bootstrap, SQLAlchemy).

Validation de la capacité à manipuler et afficher les données attendues.

AC25.04 | Définir et mettre en œuvre une démarche de suivi de projet

Organisation du code en modules (models, forms, views).

Utilisation de Git pour le suivi des versions et la collaboration.

AC26.02 | Appliquer une démarche pour intégrer une équipe informatique au sein d'une organisation

Respect des conventions de projet pour faciliter l'intégration de nouveaux membres.

Documentation des routes et des modèles.

AC26.03 | Mobiliser les compétences interpersonnelles pour travailler dans une équipe informatique

Collaboration via Git et partage des tâches.

AC26.04 | Rendre compte de son activité professionnelle

Rédaction de README et documentation technique.

Présentation des fonctionnalités réalisées et des choix techniques.