

Rapport 3

LACOUME
CORENTIN

Synthèse:

Dans la troisième partie de cette SAÉ, nous nous sommes consacrés au développement de la partie front-end de l'application. Cette étape a consisté à mettre en place l'interface utilisateur et à assurer la cohérence entre l'affichage, la base de données et la logique métier conçue lors des phases précédentes.

Nous avons donc intégré l'ensemble des fonctionnalités implémentées auparavant et veillé à ce que l'interface permette d'interagir correctement avec les données et les traitements. Ce travail nous a conduits à réfléchir à l'ergonomie, à la clarté de la navigation, ainsi qu'à la fluidité de l'expérience utilisateur.

Cette phase a également permis d'unifier tous les modules du projet pour obtenir une application fonctionnelle, utilisable et visuellement cohérente.

Analyse:

L'une des difficultés que j'ai rencontrées concernait la gestion des mutations dans la séquence ADN. Au début, je pensais que ce serait assez simple, mais je me suis vite rendu compte que ce n'était pas le cas. J'ai eu du mal à comprendre comment structurer correctement la fonction pour récupérer les données du formulaire, valider le taux de mutation et surtout gérer les erreurs de conversion.

Ce qui m'a vraiment compliqué la tâche, c'est de devoir anticiper tous les cas possibles : une séquence vide, un taux non valide, un utilisateur non autorisé, etc. J'avais l'impression que plus j'avançais, plus je découvrais de nouveaux scénarios à prendre en compte. Cela m'a obligé à revoir plusieurs fois ma logique, et à faire des tests que je n'avais pas prévus au départ.

En plus de ça, l'intégration avec la fonction muter_complet a ajouté une autre couche de complexité. Je devais être sûr que les paramètres passés à la fonction étaient cohérents et que la séquence était toujours dans un format acceptable pour l'algorithme.

Au final, cette partie m'a paru difficile parce qu'elle demandait de penser en même temps à la logique métier, à la validation des données et à la sécurité des routes. Cela faisait beaucoup à gérer d'un coup, et j'ai dû m'y reprendre à plusieurs reprises avant d'obtenir un résultat propre et fonctionnel.

Compétence:

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.

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.