

Cours : Comprendre le Processus Métier de LabGenius

Le projet LabGenius n'est pas qu'une simple application web ; c'est un **simulateur de laboratoire**. Pour le réussir, il faut comprendre le cheminement d'une manipulation génétique, de l'entrée des données jusqu'au résultat final.

1. Le Cœur du Sujet : La Séquence d'ADN

Tout commence par l'ADN. En informatique, nous allons traiter l'ADN comme une **chaîne de caractères (String)** composée uniquement de quatre lettres (les bases azotées) : **A, T, C, G**.

- **A** (Adénine) se lie toujours à **T** (Thymine).
- **C** (Cytosine) se lie toujours à **G** (Guanine).

Le Processus Métier n°1 : La Transcription

Dans un laboratoire, on ne manipule pas toujours l'ADN directement, on crée de l'ARN messager (ARNm).

- **Règle logique** : Le processus remplace le **T** par un **U** (Uracile).
- *Exemple de code* : ATGC devient AUGC.
- L'ADN est une molécule qui contient l'information génétique. Elle est composée d'une séquence de ****4 bases nucléotidiques**** :
 - A | Adénine - S'associe toujours avec T
 - T | Thymine - S'associe toujours avec A
 - G | Guanine - S'associe toujours avec C
 - C | Cytosine - S'associe toujours avec GRègle fondamentale : A s'associe avec T, G s'associe avec C (complémentarité des bases).

2. Le Flux de Travail de l'Utilisateur (Workflow)

Pour que votre application soit "limpide", elle doit suivre les étapes réelles d'un chercheur :

Étape A : La Saisie (Input)

L'utilisateur doit pouvoir entrer une séquence.

- **Contrainte technique** : Vous devez vérifier (valider) que l'utilisateur n'a pas tapé d'autres lettres que A, T, C, G. C'est la **gestion des erreurs**.

Étape B : La Manipulation (Logique Métier)

C'est ici que le PHP intervient. L'utilisateur choisit une action :

1. **Synthèse** : Créer le brin complémentaire.
2. **Mutation** : Modifier aléatoirement une lettre de la séquence.
3. **Transcription** : Transformer l'ADN en ARN.

Étape C : La Visualisation (Output)

Le résultat ne doit pas être juste du texte brut. Pour une meilleure expérience (UX) :

- Colorer chaque base (ex: A en rouge, T en bleu).
- Afficher des statistiques (ex: % de GC dans la séquence).

3. Architecture du Projet (Le "Comment Faire")

Pour que votre code soit professionnel (critère majeur du cahier des charges), utilisez une approche modulaire :

1. **Le Moteur (PHP)** : Créez des classes ou des fonctions dédiées. Une fonction `calculerMutation($sequence)` ne doit faire que cela.
2. **L'Interface (HTML/CSS)** : Utilisez un design "Laboratoire futuriste" (couleurs sombres, accents néons ou blanc médical épuré).

4. Les Points Clés pour Réussir l'Évaluation

- **La Robustesse** : Que se passe-t-il si l'utilisateur laisse le champ vide ? Votre programme doit afficher un message d'erreur clair ("Veuillez entrer une séquence") au lieu de planter.
- **Le Carnet de Laboratoire (Bonus)** : C'est une excellente idée pour la persistance des données. Permettez à l'utilisateur de sauvegarder ses séquences favorites dans une session ou une petite base de données.
- **Modularité** : Séparez bien votre CSS, votre JS et votre PHP dans des fichiers distincts.

Résumé du Processus à Coder :

1. **Entrée** : ATGC (Utilisateur)
2. **Traitement** : Fonction PHP complémentaire() -> TACG
3. **Sortie** : Affichage stylisé sur l'interface avec des couleurs.