Projet final

Les étudiants ont **liberté totale** dans le choix de la stack technologique, en respectant les principes de **modularité, maintenabilité et qualité logicielle**.

Quelques exemples de stacks possibles (non limitatif) :

- **Full JavaScript** : Backend en **Node.js (Express, Fastify, etc.)**, Frontend en **Vanilla JS, React, Vue, ou autre**.
- **PHP & Frontend au choix** : Backend en **PHP (Laravel, Symfony, ou natif)**, Frontend en **HTML/CSS/JS, Vue, React, etc.**
- **Python & Frontend au choix** : Backend en **Flask ou Django**, Frontend libre.
- **Java/Kotlin & Frontend au choix** : Backend en **Spring Boot, Quarkus, etc.**.
- **C# & Frontend au choix** : Backend en **ASP.NET**, Frontend libre.
- **Autres langages/frameworks** : Ruby on Rails, Go, Rust, etc.

Fonctionnalités attendues

Interface utilisateur ergonomique permettant :

- La saisie de deux nombres et le choix d'une opération.
- L'affichage du résultat du calcul.
- L'affichage d'un **historique des calculs**.

Backend permettant :

- La gestion des opérations +, -, ×, /, **% (modulo)**, **exponentiation (x^y)**.
- Une gestion **des erreurs** (ex : division par zéro, entrées invalides).

Qualité et tests :

- **Tests unitaires** sur les fonctions de calcul.
- **Tests E2E** simulant une interaction utilisateur complète.
- **Tests de performance** sur le backend.
- **CI/CD** : automatisation des tests (GitHub Actions, GitLab CI, Jenkins…).
- **Reproductibilité** de l'environnement (Docker, gestion des dépendances…).
- **Linting & Documentation** pour assurer un code propre et maintenable.

Déroulement du projet

1. Développement de l'application

- Conception de l'interface utilisateur et du backend.
- Gestion des calculs et des erreurs.
- Connexion backend-frontend si applicable.

2. Implémentation des tests et validation

- Mise en place des tests unitaires et d'intégration.
- Exécution des tests E2E et validation des fonctionnalités.
- Analyse de la performance et optimisation si nécessaire.

3. Mise en place des bonnes pratiques

- Configuration du CI/CD et exécution des tests automatisés.
- Documentation du projet (README, API, guide d'installation).
- Ajout d'outils de qualité logicielle (linting, formatage automatique).