

Mini-Projet : Utilisation de Git & GitHub pour le Suivi de vos Cours

Période : du 21 au 28 mai 2025

La **gestion des risques informatiques** est l'ensemble des processus visant à **identifier, évaluer, prévenir et atténuer** les menaces pesant sur les systèmes d'information (pannes, cyberattaques, erreurs humaines, pertes de données, etc.).

Dans ce cadre, **Git** (système de gestion de versions) et **GitHub** (plateforme collaborative) jouent un rôle clé en :

- **Assurant la traçabilité** des modifications, ce qui permet de **revenir rapidement à une version stable** en cas d'incident ;
- **Réduisant les erreurs humaines** grâce à la validation du code ;
- **Sécurisant le code source** grâce à des sauvegardes dans le cloud, minimisant les risques de perte ou de corruption ;
- **Automatisant les tests** via l'intégration continue, pour prévenir les défaillances avant la mise en production ;
- **Facilitant les audits** grâce à un historique horodaté de toutes les actions.

Ainsi, Git et GitHub sont des **leviers puissants de maîtrise des risques** dans tout environnement informatique.

Objectif général

Ce mini-projet a pour but de vous familiariser davantage avec les outils de **gestion de version** et de **travail collaboratif** que sont **Git** et **GitHub**, en les appliquant concrètement à l'organisation et à la sauvegarde progressive de vos contenus de cours personnels.

Objectifs spécifiques

En réalisant ce projet, vous serez capables de :

- **Sauvegarder progressivement** vos documents de cours sur GitHub ;
- **Suivre l'historique** de vos modifications ;
- **Revenir à une version antérieure** de vos fichiers à tout moment ;
- **Collaborer efficacement** sans conflit de version ;
- **Réduire les risques** de perte de données (erreurs humaines, pannes, etc.) ;
- **Organiser vos ressources** de manière claire, durable et professionnelle.

Consignes de travail

1. Créez un **dépôt public GitHub** intitulé par exemple `mes-cours-2025`.
2. Pour chaque cours, créez un **sous-répertoire dédié** :

- Exemple : architecture-réseaux/ gestion-des-projets-informatique etc.
- 3. **Ajoutez régulièrement** les contenus suivants :
 - Plans de cours
 - Diapositives
 - Notes personnelles
 - Exercices, TPs, projets, etc.
- 4. **Effectuez des commits fréquents** avec des **messages clairs et explicites** :
 - Exemples : "Ajout du plan de cours gestion des projets informatique ", "Mise à jour du TP3 architecture réseaux "...
- 5. **Poussez (push) systématiquement vos modifications** sur GitHub.
- 6. **Simulez au moins une fois une opération de retour à une version précédente** (rollback ou checkout d'un commit antérieur).

Livrable attendu (rapport final)

À rendre au plus tard le 28 mai 2025 à 23h59.

Le rapport doit contenir :

- ✓ **Lien vers votre dépôt GitHub** (en accès public)
- ✓ **Captures d'écran commentées** illustrant :
 - l'organisation des dossiers
 - les commits effectués
 - un exemple de retour à une version antérieure
- ✓ Une **courte analyse des perspectives** :
 - Quelle utilité pour vos futurs projets académiques ou professionnels ?
 - Qu'avez-vous appris grâce à cet exercice ?
 - Quelles fonctionnalités Git souhaitez-vous approfondir ?