Soutenance Git

Bonjour et bienvenue à ma soutenance sur Git ! Aujourd'hui, nous allons parler de Git, un système de contrôle de version distribué. J'espère que vous êtes prêt à apprendre et à comprendre comment Git fonctionne.

Qualité de la soutenance (3 points) : Pour cette soutenance, je vais vous donner une introduction sur Git, vous expliquer les différents concepts clés de Git et vous montrer comment utiliser Git pour travailler sur un projet. Je vais également vous montrer comment configurer et personnaliser Git, ainsi que quelques astuces pour vous aider à travailler plus efficacement avec Git.

Commits signés de tous les membres et sur tous les commits (1 point) : Avant de parler des fonctionnalités de Git, je tiens à souligner l'importance de signer les commits. Chaque membre du projet doit signer ses commits, ce qui permet d'authentifier l'auteur du commit. Pour signer un commit, vous devez utiliser la commande "git commit -s" ou "git commit --gpg-sign".

Issue + issue templates (1 point) : Les issues sont des fonctionnalités clés de Git qui permettent de suivre les problèmes et les tâches liées à un projet. Il est important de créer des issues claires et détaillées pour chaque tâche ou problème rencontré. Nous avons également utilisé un modèle d'issue pour garantir que toutes les informations nécessaires sont incluses.

Merge Request + Merge Request template (1 point) : Les Merge Requests sont une autre fonctionnalité clé de Git, qui permettent de proposer des modifications à une branche principale. Nous avons utilisé un modèle de Merge Request pour garantir que toutes les informations nécessaires sont incluses dans la demande de fusion.

Github project (1 point) : Nous avons également utilisé la fonctionnalité de projet Github pour suivre les tâches et les problèmes liés à notre projet. Nous avons organisé les tâches en fonction de leur priorité et de leur état pour faciliter le suivi du travail à effectuer.

Un readme, contributing, code of conduct correctement rédigés (1 point) : Nous avons pris soin de rédiger des fichiers README, CONTRIBUTING et CODE OF CONDUCT clairs et concis pour aider les nouveaux contributeurs à comprendre notre projet et notre processus de contribution.

Un .gitignore cohérent (1 point) : Nous avons également ajouté un fichier .gitignore cohérent pour éviter que des fichiers inutiles ou sensibles soient inclus dans notre dépôt Git.

Respect du Git flow (les bonnes branches et les bonnes protections de branche) (2 points) : Nous avons suivi le Git flow, qui est une méthode de gestion de branches pour les projets Git. Nous avons utilisé les branches de fonctionnalités, les branches de développement et les branches de version pour organiser notre travail et garantir la qualité de notre code. Nous avons également protégé nos branches de version pour empêcher les modifications non autorisées.

Un hooks qui vérifie un code js (linter) (2 points) : Nous avons également utilisé un hook Git pour vérifier la qualité de notre code JavaScript. Le hook utilise un linter pour détecter les erreurs et les avertissements dans notre code, ce qui nous aide à garantir la qualité de notre code.

Question de cours individuel sur 3 points : Je suis maintenant prêt à répondre à vos questions. N'hésitez pas à poser toutes les questions que vous avez.

"git commit -s" ou "git commit --gpg-sign".

3/ Créez une nouvelle branche de fonctionnalité à partir de la branche principale de votre référentiel Git en utilisant la commande "git checkout -b nouvelle-branche".

Effectuez les modifications souhaitées dans cette nouvelle branche.

Une fois que vous avez terminé vos modifications, confirmez vos changements en exécutant la commande "git commit".

Poussez votre nouvelle branche sur le référentiel Git distant à l'aide de la commande "git push".

Accédez à la page de votre référentiel Git sur le web et créez un nouveau Merge Request en sélectionnant la nouvelle branche que vous venez de créer.

Ajoutez une description claire et concise de vos modifications et mentionnez les personnes concernées pour les informer du Merge Request.

Attendez que les membres de l'équipe examinent votre Merge Request, fournissent des commentaires et approuvent la fusion de votre branche dans la branche principale du projet.

Git hub Project:

Accédez à votre référentiel Git sur GitHub.

Cliquez sur l'onglet "Projects" dans le menu supérieur.

Cliquez sur le bouton "New project" pour créer un nouveau projet.

Donnez un nom à votre projet et choisissez un modèle de tableau Kanban adapté à vos besoins.

Ajoutez des colonnes à votre tableau pour représenter les différentes étapes de votre processus de travail (par exemple, "À faire", "En cours", "Terminé").

Ajoutez des cartes à chaque colonne pour représenter les différentes tâches à effectuer dans le cadre de votre projet.

Déplacez les cartes entre les colonnes pour refléter l'état d'avancement de chaque tâche.

Invitez les membres de votre équipe à collaborer sur le projet en leur accordant des autorisations de lecture et d'écriture.