Compte rendu du brief projet

Sur ce projet on a pu travailler en groupe, organiser une équiper de 5 personne et partitionner les taches à la méthode SCRUM chaqu’un des membres de groupe a faut une tache différente au cour du brief projet

On a pu s’adapter au changement en effectuent les taches de maniéré rotatif pour que chaqu’un d’entre nous puise effectuer toutes les tâches du projet d’une manière individuelle, chercher les informations nécessaires pour effectuer sa tâche et aider à l’avancement du projet, on a pu aussi organiser notre temps d’une manière à finaliser le projet dans les délais.

Les 3 parties du projet  :

1ere partie :

* Organiser l’équipe et a partitionner les taches
* Appliquer la méthode sacrum pour partitionner les taches
* Savoir ce que c’est que le git et Se familiariser avec les commande git
* Création et enregistrement des version et d’avancement de notre projet avec des commit
* Ajout de branche et merging des branches pour tester les modifications avant de les mètre en place
* Création des alias des commandes git pour simplifier l’utilisation du git bash
* Utiliser le fichier .gitignore
* Comment résoudre les conflits de merg de branche avec L’outil P4MERE

2eme partie :

* Création et utilisation des tags
* Comment enregistres les modifications avec le git statsh et les récupérer
* Création et utilisation du d’un compte GitHub
* Push du projet du git local vers GitHub avec les tags
* La différence entre l’authentification https et ssh et comment crée la clé ssh ou cas d’utilisation
* Comment ajouter de nouveau document a notre ripo git a partir des comment git bash
* Comment on push ces nouveau document
* Conflit de push des nouveaux documents et comment gérer ce conflit
* L’utilité des commandes git fetch et git pull

3eme partie

* Effectuer des changements sur le GitHub et les visualiser sur le git local
* Etre en mesure de manipules les changement sur le ripo local et GitHub au même temps
* Le fonctionnement de branching et du menu INSIGHTS
* Avoir une vue graphique des commit et du merge des branche

Voici quelque commande utiliser dans le projet

git config --global u

git config --global user.name

git config --global user.email

cd pour accédé au dossier

git init initialisation du dossier

touch création fichier

ls pour listé les fichier

ls -a tt les dossier

git status status du répertoire

git ls-files --others --exclude-standard pour vérifier les fichier traqué

git commit

git log historique des commit

cat index.html verrifier le contenu

vim index.html ajouter text

i = insert text

ctrl +c :qa! ( :x) pour enregistrer et quiter

cat index .html pour afficher le contenu

git commit -a -m pour garder

pour coppier un dossier sur un autre on doit être déjà sur desktop mv (nom du nouveau dossier /\*) (nom du projet/)

apré avoir copier les fichier on add . Et on commit tt les nouveau fichier

start index.html ouvrier les fichier

mv (1er nom du fichier ) (nouveau nom) changer le nom du fichier

rm (nom du fichier) pour supprimé -r pour forcer

git reset -- hard annuler le dernier commit ca affiche id du commit

git reset --hard 'id du 1er commit pour aller sur le 1er commit si on effectu une modification et on commit impossible d'allé sur les autre commit

git log --oneline afficher le lot en 1seul ligne

git log -1 historique du dernier commit

git log -a pour auteur

git log --format= '%an %ad ' pour afficher en ordre auteur et datte

git log -- author "nom d'auteur "

rm style.css pour supprimer un fichier

git diff parcourir les différence dans l'historique

"git branche"

"git checkout "

"git checkout -d" pour supprimer

sit la branche et tjr pas merger on peut forcer la suppression avec "git checkout -D "

"git merge "

Conflit de merge

Voir ou ce trouve le conflit choisir un des 2changement et refaire un"" add commit ""

"git remote add origine + lien " pour crée une remote

"git remote -v "pour voir les remote disponible

git push origin (par default ) master (nom de la branche )

"git pull" pour recuperer les fichier depuit github remote (origin) (branch) master

git fetch

git pull origin master

git push origine master

Lien du rapport de l’équipe : https://github.com/ucef8233/CR.git