Objectif du projet:  
Le projet a pour but de mettre en pratique nos connaissances du déploiement continus CI/CD grâce à Git et Github.  
En utilisant github action , nous allons , dans le tp déployé une petite application Node.js qui va générer des citations .  
  
Structure du code:  
Pour mieux clarifier le code et éviter que tout soit dans un même répertoire .  
Dans notre project citation , la fonction qui va générer la citation sera dans le src et le test unitaire sera dans tests.

Public va contenir notre page index.html qu’on va l’envoyer directement dans le render package.json sera les dépendances et la configuration qu”on fait l’initialisation du node dans github/workflow sera le pipeline du CI/CD ci.yml  
  
Le pipeline est une automatisation du processus de développement avec GitHub Actions.

Il se déclenche à chaque push ou pull request sur la branche main.

Le workflow est celui qui va être la partie de vérification et validation pour éviter d'éventuel erreur et se décompose en deux parties :

intégration continue( CI)

Installe Node.js et les dépendances Exécute les tests,Vérifie que le code est fonctionnel avant tout déploiement

déploiement continu (CD) S’exécute uniquement si les tests réussissent Déploie automatiquement le site statique sur Render, où la page public/index.html est hébergée.

Ce pipeline garantit que toute mise à jour du code est testée avant d’être publiée.

Ainsi, il n’y a plus besoin de déployer manuellement : tout se fait automatiquement à partir du code validé.  
  
Difficultés rencontrées  
C’est la première fois que j’entends render et j’ai du voir quelque documentation pour résoudre et appliquer sur le projet en question.  
La partie technique pour lier l’automatisation du CI/CD avec comme ressource Render a été difficile car il y avait bcp d’erreur lors de leur lisaison  
Améliorations possibles  
Il faut que je me document plus sur les nouveautés et augmenter mes connaissances

1-l’intégration continue(CI) c’est la vérification automatique pour voir si le projet ne contient pas d’erreur de build à chaque modification et push tandis que le déploiement condinu ou CD s’il n’y a aucun problème dans le build automatiquement déployé sur le server  
2-les tests unitaires comme son nom l’indique sera un test , une vérification automatique que le projet ne contient pas d’erreur, éviter qu’un bug va dans le projet qui va être déployé,.  
3-Pour sécuriser les workflows CI/CD publics , comme d ans la documentation , on doit crée une clé d’API comme celui du render\_api\_key et render\_service\_id qui va nous sécuriser notre workflows publics.

4-Le déploiement automatisé pour un projet d’équipe pourra apporter un gain de temps énorme grâce aux tests unitaires , le déploiement en continu facilitera les tâches et tout sera en sécurité