# Documentation Github

## 1. Accès au GitHub :

Tous les membres de l'équipe auront accès au projet GitHub.

Les membres de l'équipe auront la possibilité de cloner le repository dans lequel ils souhaitent travailler.

## 2. Branches principales :

Nous utilisons trois branches principales dans notre projet :

- La branche "prod" (production) : C'est la branche où la version la plus récente et stable de notre code est déployée et utilisée en production. Elle ne contient que le code prêt à être utilisé par les utilisateurs finaux.

- La branche "master" : C'est la branche principale de développement. Elle contient le code en cours de développement et qui sera fusionné dans la branche "prod" après avoir été testé et validé..

- La branche "dev" (développement) : C'est la branche utilisée pour les tests et les intégrations continues. Les développeurs fusionnent leurs branches de fonctionnalités ou de corrections de bugs dans cette branche pour assurer une intégration et des tests réguliers avant de les fusionner dans la branche "master".

## 3. Branches liées aux tickets :

Nous utilisons un système de tickets pour suivre les bugs et les fonctionnalités à développer dans notre projet. Voici comment ces branches sont organisées :

- Bugs : Lorsqu'un bug est identifié, un ticket est créé pour le décrire en détail. Ce ticket est ensuite attribué à un développeur qui se charge de résoudre le bug. Le développeur crée une branche distincte à partir de la branche "dev" pour travailler sur la résolution du bug.

- Fonctionnalités : Lorsqu'une nouvelle fonctionnalité est proposée, un ticket est créé pour la décrire en détail. Ce ticket est ensuite assigné à un développeur qui sera responsable de son implémentation. Le développeur crée une branche distincte à partir de la branche "dev" pour travailler sur la fonctionnalité.

## 4. Droits de pull requests :

Les membres de l'équipe pourront proposer des pull-requests pour les modifications qu'ils souhaitent apporter au projet.

Les demandes de pull doivent être soumises en utilisant un modèle standardisé pour garantir la qualité et la cohérence.

Les commentaires et suggestions concernant les demandes de pull seront fournis via GitHub.

## 5. Fusion de pull requests :

La capacité de fusionner les demandes de pull sera limitée à une autre personne en dehors du créateur de la pull request.

Cette personne s'assurera que toutes les modifications ont été correctement testées et validées.

Les commentaires et suggestions concernant les demandes de fusion de pull seront fournis via GitHub.

## 6. Nomenclature des Commits :

Pour maintenir une structure cohérente et compréhensible de l'historique des commits, nous utilisons une nomenclature spécifique pour chaque commit.  
  
Préfixe du type de commit entre crochets : Chaque commit commence par un préfixe entre crochets pour identifier le type de modification effectuée.

Description succincte : Après le préfixe du type de commit, une description courte mais informative des changements effectués est fournie.

Les commits sont faits en anglais

[Feat] : Pour les nouvelles fonctionnalités ajoutées.

Ex: *“[Feat] Add Google Sign-In functionality”*

[fix] : Pour les corrections de bugs.

Ex: *“[Fix] Resolve image display bug”*

[docs] : Modifications de la documentation.

Ex: *“[Docs] Update API documentation”*

[refactor] : Refactoring du code existant.

Ex: *“[Refactor] change code structure”*

[style] : Modifications de l'apparence ou du formatage du code.

Ex: *“[Style] Update UI font and colors”*