Github branching strategie:

The strategy which I’m gonna explain today is **Feature Branch Strategy.**So, what is it? As the name itself states that, it is the way of maintaining the branches separately based on the feature. Let’s get started with it…

Feature Branch Strategy

As a first step create 3 branches from the **main/master**branch, named

1. Prod Stable
2. Prod
3. Staging

Afbeelding met schermopname, diagram, lijn, Perceel

Door AI gegenereerde inhoud is mogelijk onjuist.

**Prod Stable branch:**This is the branch containing the stable code of production code.

**Prod branch:**This branch’s code will be live in production. Before merging any code from the *staging branch,* *prod branch’s* code should be merged in *Prod stable branch*. Dus maak eerst een kopie van Prod branch naar de Prod Stable branch voordat je code vanuit staging branch naar de Prod branch merged.

**Staging branch:**This branch contains all the testing code and it is partial stable.

What’s next?

Before you gonna work on in any new feature, create a new branch with *staging branch*as base branch.

Afbeelding met diagram, Perceel, lijn, schermopname

Door AI gegenereerde inhoud is mogelijk onjuist.

Once the code is committed, run the unit tests. Once all test cases and manual tests are passed then merge it with the *Staging branch* and run the unit tests and manual tests again to check the backward compatibility and ensuring no previous changes were broken. Follow the same step for Feature 2 branch as well.

Afbeelding met diagram, lijn, Perceel, tekst

Door AI gegenereerde inhoud is mogelijk onjuist.

The Above figure illustrates the complete strategy. By following this strategy you can maintain the code in a stable manner and you can roll back to the previous version easily since we have *Prod Stable Branch.*

If you are gonna implement any new feature, take the latest *staging branch*code as base branch. And repeat the cycle.