Краткий гайд о работе в команде на github

1) создаем репозитори и добавляем в него нужные файлы.

git init

git romoute add origin <site>(site – ссылка на ваш гитхаб)

git checout -b <ветка> (в нашем случае ветка разработки<development>)

git add .

git commit -m “first commit”

git push origin <ветка>(в нашем случае ветка разработки<development>)

2)РАБОТА В КОМАНДЕ

вот, у вас или у вашего сокомандника уже есть ветка на гитхаб, перед каждым разом когда вы садитесь выполнять задание нужно прописать:

git pull origin <ветка>(в нашем случае ветка разработки<development>)

в первый раз, когда вы будете только начинать работать с кодом нужно будет выбрать папку кда будете сохранять проект, затем вызвать git bush here, и

git clone <site> если вам нужно импортировать ветку не main, то git clone -b <ветка> <site>

Далее создайте на своем гитхабе такой же репозиторий что и тот, с которого вы клонировали и залейте туда то, что вы с клонировали.

Для этого перейдите в папку с проектом, там должна быть папка «,git» не заходя в нее открываем git bush и пишем:

git romoute add origin <site>(site – ссылка на ваш гитхаб)

git checout -b <ветка> (в нашем случае ветка разработки<development>)

git add .

git commit -m “first commit”

git push origin <ветка>(в нашем случае ветка разработки<development>)

Поздравляю, теперь у вас есть такой же репозиторий что и у всех.

Т.к. разрабатывать код мы будем не в ветке development, а в Fork, то меняем ее.

git checout -b <ветка> (в нашем случае ветка разработки<Fork>)

Теперь у вас есть ветка, куда вы будете заливать код, который вы сами пишите и есть ветку где весь актуальный код, написанный вашими сокомандниками.

Теперь можете смело работать и заливать свой код в <Fork>

git checout -b <ветка> (в нашем случае ветка разработки<Fork>)

git add .

git commit -m “first commit to Fork”

git push origin <ветка>(в нашем случае ветка разработки<Fork>)

дальше работаем в своем локальном репозитории. Когда закончите создайте на своём гитхабе новый репозиторий с любым именем