It's easy to get stuck during the build out of a large project. You can use these workflow tips to get back on track. Save this URL for when you need to refer to the code in the final project: [Upskill Github](https://github.com/UpskillCourses/upskill_saas_tutorial)

If you're experiencing an error that you can't seem to get past, always search the error message in Google. Look through StackOverflow posts for tips. This is a critical developer workflow skill that will make you better at debugging.

If you've tried everything you can, then, since you're using Git, you can simply backtrack to a point in the application that you know was working properly. Remember Git works like a time machine for your code.

**Go back to a previous branch.**

First stash away any uncommitted changes otherwise Git will complain:

git stash

Then, simply check out an older branch that you know has the correct, working code. It could be the master branch, or some other branch like contact\_form, etc. (replace previous\_branch\_name below with whichever branch you want to switch to):

git checkout previous\_branch\_name

Delete the branch with the broken code to keep things organized:

git branch -D broken\_branch

Then checkout a new branch, and name it whatever is relevant:

git checkout -b new\_branch\_name

Now, you're basically back to the point where the app was working before in a previous branch's state.

**Reset to a working branch.**

Another, more direct workflow is to use Git's reset feature.

git stash

The following command will reset your current branch to whichever branch you specify after the "/". You can put any other branch name in there, but in this example, your code would end up matching whatever was most recently pushed to the master branch:

git reset --hard origin/master