Part 1: Creating a Repository
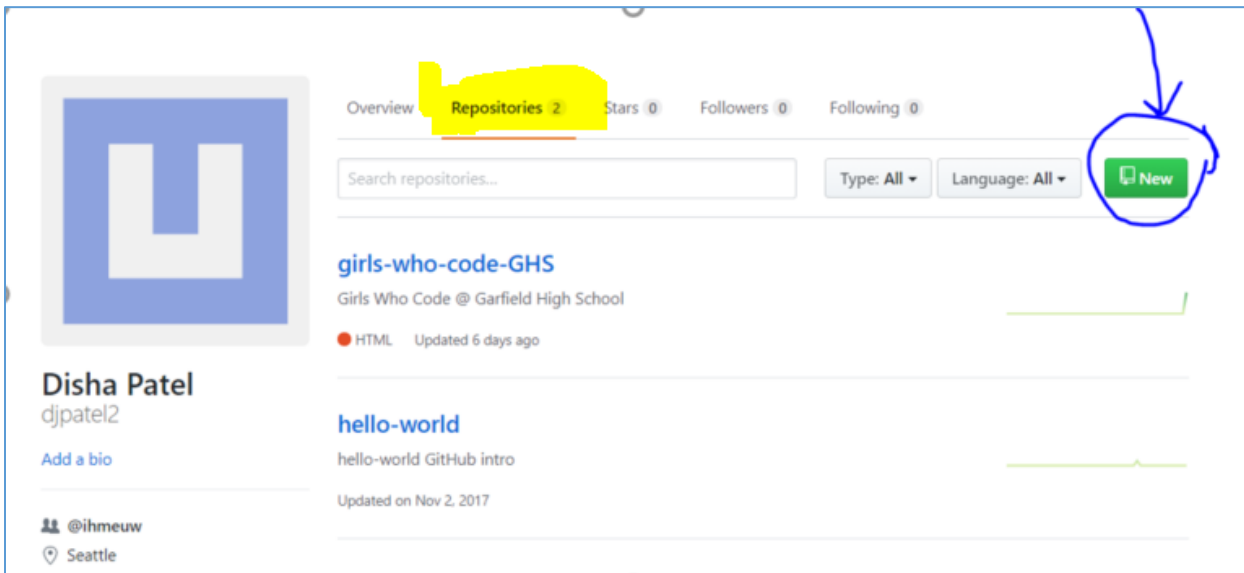
Once you created your GitHub account, create a repository for GWC. This is where you will save your Glitch code:

A Git repository must have at least one file in it, before any code can be exported to it. We will create a README file to fulfill this. Create a README file, use this link to create the README (should auto populate after you create a repository).

djpatel2 / **GWC**

👁 Watch ▾ 0   ★ Star 0   ⑂ Fork 0

<> Code   ⓘ Issues 0   Pull requests 0   Projects 0   Wiki   Insights   Settings

Quick setup — if you've done this kind of thing before

⬇ Set up in Desktop   or   HTTPS   SSH   https://github.com/djpatel2/GWC.git

We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# GWC" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/djpatel2/GWC.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/djpatel2/GWC.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

💡 **ProTip!** Use the URL for this page when adding GitHub as a remote.

Add your name or anything else you want to the file, and then commit the new file

GWC / README.md   or cancel

<> Edit new file   ⊙ Preview   Spaces ⇕   2 ⇕   No wrap ⇕

```
1   # GWC
2   Girls Who Code for Garfield High School
3   Disha Patel
4
```

Commit new file

Create README.md

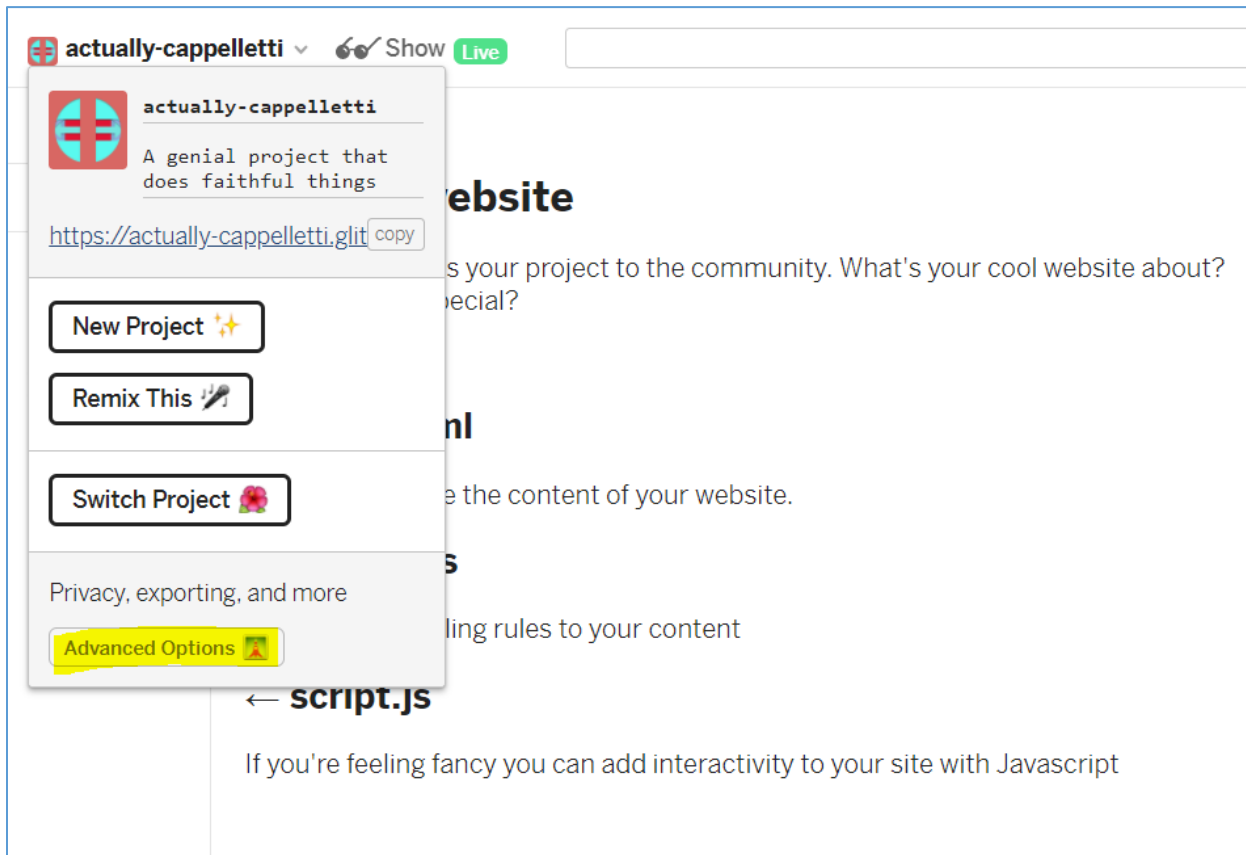Add an optional extended description...

Commit new file   Cancel

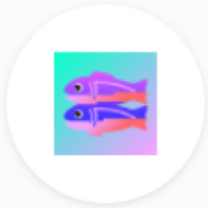You should now have one file (README.md) in your GWC repository.

Part 2: Adding your code from Glitch to your Git repository:

Make sure you are signed into Glitch with your new GitHub account.

Click on Advanced Options (click on your project icon in the top left and you will see this option):

After you click on Advanced Options there should be an option that says 'Grant Access', click that and then Authorize FogCreek

Now, go to Advanced Options again and you should be able to click on 'Export to GitHub'.
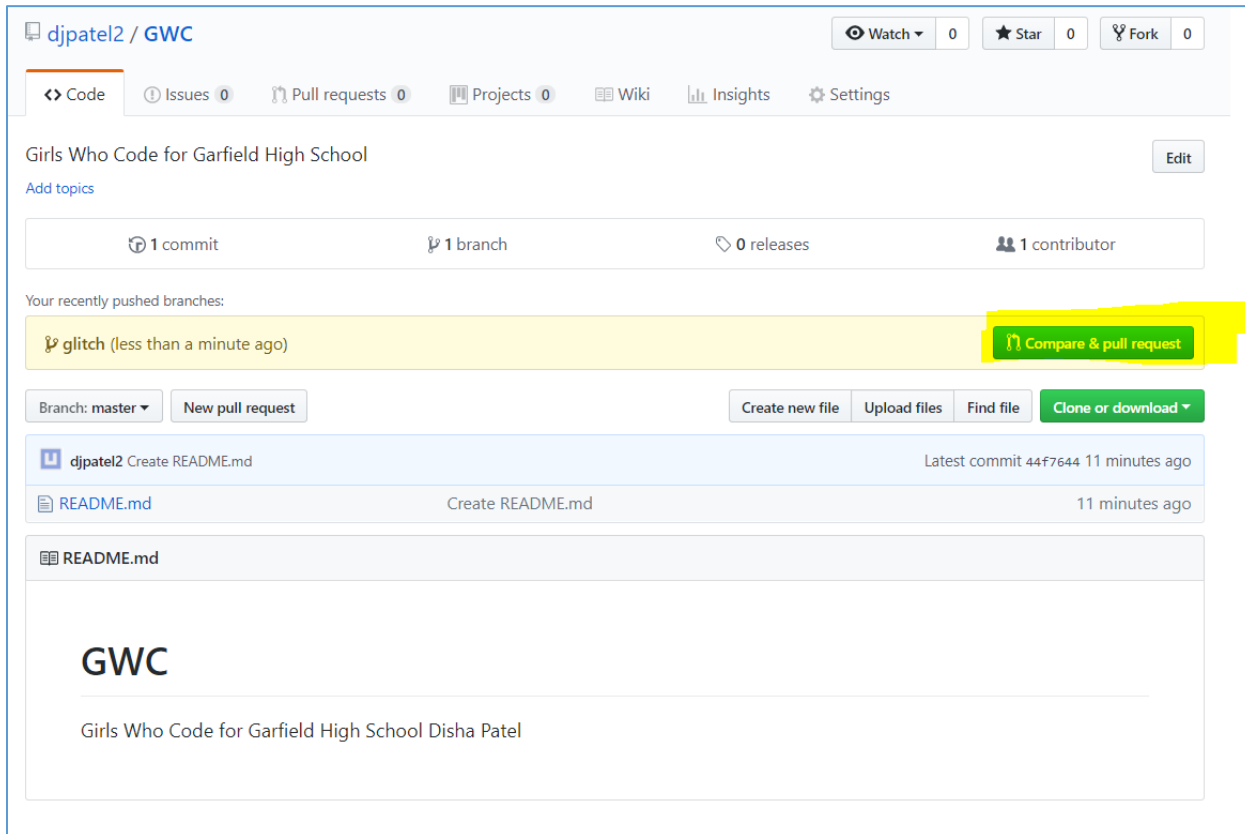
Fill in the repository path so glitch knows where to drop your files:
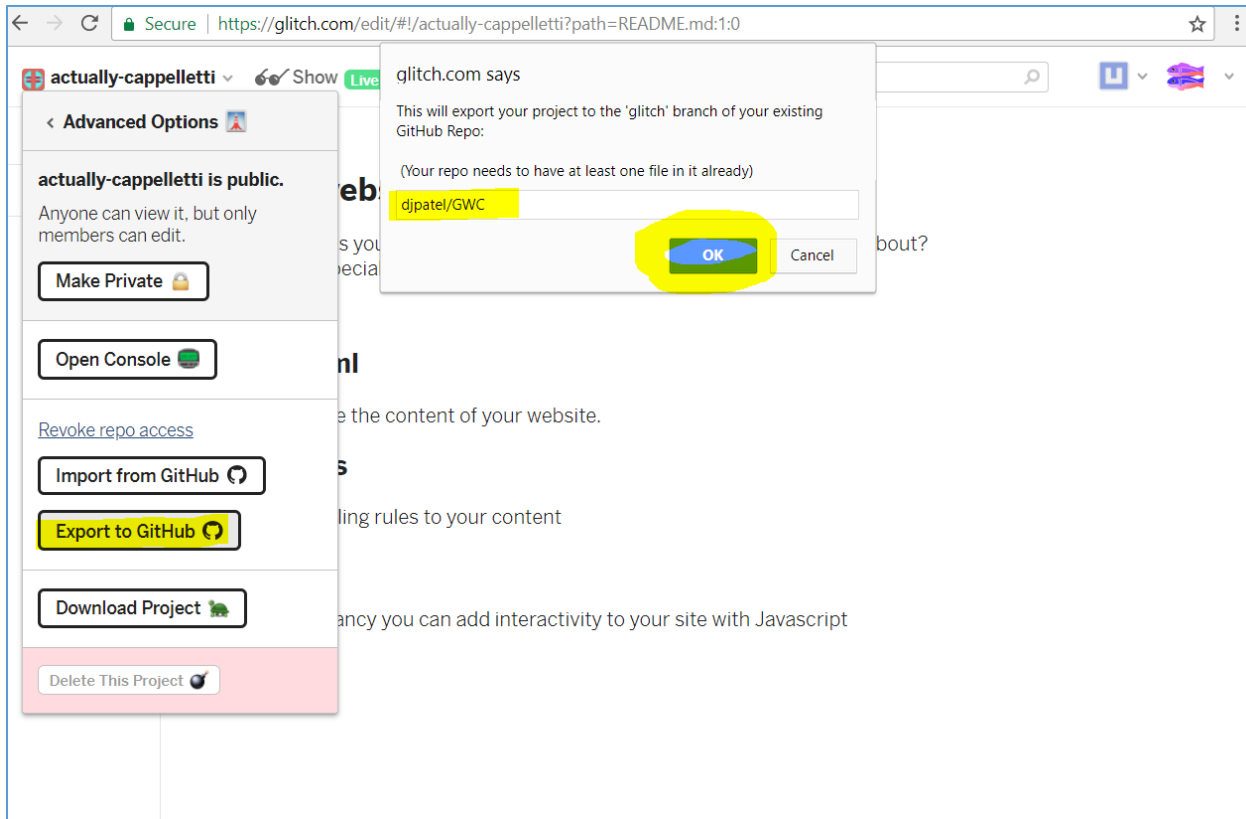
You will see an update with Glitch message automatically pop up, and then a green bar in the right corner that says 'exported successfully'

Part 3: Creating a Pull Request

Add the new code in by creating a Pull Request:

You can add a comment describing the code if you want to, then click 'Create Pull Request':

Merge the Pull Request you just created by clicking 'Merge Pull Request' then 'Confirm Merge':

Double check that the code exported successfully by navigating to the GWC folder. You should see all the new files you have been working on from Glitch:

Congratulations! You have successfully completed your first code export and Pull Request!