GitHub Workflow

The point of this document is to provide an easy-to-understand index of all the things we do on GitHub. This is meant both for designers and developers. If there's anything missing from this guide, feel free to contact paul@dentalrevenue.com to request an update!

What is GitHub?

GitHub is simply a collection of folders and files. The site provides various tools for teams to see exactly who has worked on certain files, and what changes have been made. Our organization has a GitHub account, which you should have access to if you're reading this document. The Dental Revenue GitHub page can be found at: https://github.com/Dental-Revenue. If you navigate to this page, you'll see we have several different repositories. Each of these represents different theme templates that we've used in the past. At the time of this writing, the only one in active development is codebase-2. This is a private repository, so only people who have been granted access will be able to see this page. Most of what you do will be with this repo, so for all the following actions you should be in the codebase-2 repository.

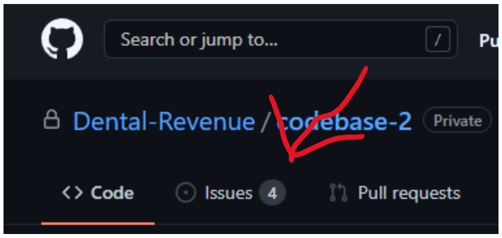
It's convenient to think of a repo like a river; as code is added, the river gets longer, branches off into smaller streams, and those streams eventually come back to the main branch of the river. So, for every repo you see, there will be a main, or master branch. This is the code that is currently in production on our sites! Any time a developer needs to make changes, they will create a new branch on their local machine, which will allow them to make changes without breaking the master branch. Once they've made the necessary changes, they upload that branch to GitHub. At this point, another developer will review the code to confirm that it looks good. If it does, that developer will approve the changes, and that branch is ready to be merged into the master branch. Once we have enough branches to merge back in, we do that and make a new release! It might seem complicated, but this allows us to track and review all the code changes that are being made.

GitHub Issues

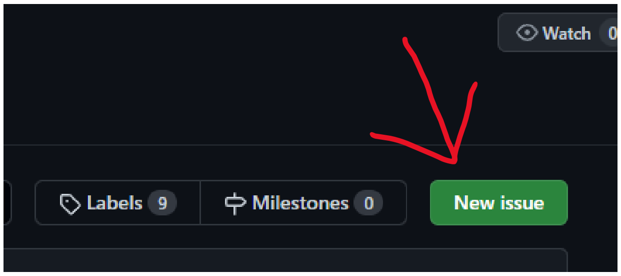
There are two types of issues that we'll mainly be dealing with, and those are bugs or features. A bug is when we expect the website to do one thing, but it does another. Examples would be unexpected behavior on mouse hover, or the nav menu looking wrong on mobile, or an animation jumping around on the screen. A feature (also known as an "enhancement") is when the site doesn't do something useful that we would like it to. An example would be any kind of module feature that doesn't exist that we would like to have. Bugs will be addressed by the development team before features and may trigger an unscheduled patch release to fix the bug. Feature releases are usually on a schedule and will take more time to develop and test.

Steps To Add an Issue

1) Log in to GitHub, and navigate to the codebase-2 repo.
2) Click on the Issues tab:



3) Click on the New Issue button:



4) Give the issue a title. The title should be brief and general. Examples of good titles include "Fix Static Image Split Bug", or "Add Color Picker to Smart Gallery". Examples of bad titles would be "Remove the Button On The ATF And Add It To The Footer", or "Do The Thing We Talked About In The Meeting". These are both too long, while the first one is too detailed, and the second one is not detailed enough.

5) In the main text box, you can add comments. This is where you should put all the information you have on this issue, including which modules are affected, the list of features that should be added, and all relevant details about the issue.

6) To the right of the text box, there are a number of options which can be useful. For all of these, you can set them by clicking the gear icon next to the name.

   a) Assignees – You can assign this issue to a specific developer, including yourself. If left unassigned, either the development team lead or one of the developers will assign it.
   b) Labels – There are many options here, but as of this writing, we haven't used any except Bug or Enhancement. If you feel a different label would be more appropriate, feel free to use it.
   c) Projects – You can assign this issue to a specific project.
   d) Milestones – This can be ignored in most cases, this is more meant for much more complicated issues than we use.
   e) Development – This shows any linked branches or pull requests. You can't set this until after you create the issue, but it is recommended. Once the issue is created, you should create a new branch for this issue. The developer will then do all their work in this branch. The branch name should be one or two words, and if it's more than one word, separate the words with a hyphen (e.g., static-image-fix).

7) Create the issue by clicking the Submit New Issue button. That's it, the development team will take it from here.