**Overview**

I’ve built project pages using a combination of blogdown (R package), Hugo, and the Wowchemy theme set. Blogdown helps with editing the .md and any .Rmd files for the actual content, Hugo was preferred over Jekyll due to blogdown and better Windows integration, and Wowchemy provided a theme framework with tons of customization options and a highly responsive team on Discord.

**Workflow**

Once a template is locked in for style, the workflow will be:

* Set images and overview text (project name, summary of the brief etc.) in the landing page (\_index.md)
* For each of the three text snippets that appear on the landing page:
  + Create a folder (any name ok) under /projects directory
  + Make a featured.jpg to appear on the landing page
  + Write the content to appear on the landing page in the *summary* section of the YAML header of index.(R)md
  + If embedded R content is needed, write all content in index.Rmd then knit to index.md. If no embedded R then you can work directly with index.md
* Deploy to Netlify

**Customization**

Coarse but easier/quicker customization can be done within the Wowchemy theme framework. Detailed documentation is here: <https://wowchemy.com/docs/>

Custom fonts and coloring required a few steps: <https://wowchemy.com/docs/getting-started/customization/#custom-theme> and <https://www.chrislockard.net/posts/using-local-fonts-hugo-academic-theme/>

At a more granular level everything on the website is supposed to be customizable using custom css code.

**Deployment**

Once set-up we can just update and push content to the repo, then Netlify re-builds the side.

Github pages can also do this. The main advantage of Netlify over Github is it allows any directory structure, so we can keep repos tidy while storing websites together with their corresponding projects. Github pages only accepts a folder structure where you maintain your website in a repo’s root directory or root/docs or root/gh-pages.

Either way, deploy following the blogdown guide: https://bookdown.org/yihui/blogdown/netlify.html

And the Wowchemy theme needs a minor tweak to website configs for all content (mainly images) to deploy correctly: https://wowchemy.com/docs/hugo-tutorials/deployment/

**Examples**

See below for a walkthrough of how different elements of the website relate to the code, as well as where key bits reside in the folder structure.

|  |  |
| --- | --- |
|  | Each section of the landing page corresponds to a ‘block’ in the \_index.md YAML header (coloured sections match website to code blocks).  Because the landing page is a bit funky (all content defined in the header) it doesn’t seem possible to embed R content here.  The ‘portfolio’ block uses tags to fetch content in tagged index.md files with a paired featured.jpg image. These pairs of .md and featured image are stored in separate folders in the projects directory. (See below)  See <https://wowchemy.com/blocks/> for the different blocks you can use with this theme |
|  | The text snippets on the landing page come from individual .md files like the one below. The tag in the header is used to connect the content on the landing page to this content file (and the featured.jpg stored in its directory).  The text snippet on the landing page is in the ‘summary’ section of the YAML header, but doesn’t appear on the content page itself.  Instead, this page displays any content below the YAML header like a normal .(r)md file. |

Text

Description automatically generated

1.

2.

3.

4.

5.

**1. pages-test:** project directory

**2. gh-pages:** hugo site directory

**3. content:** contains the landing page (\_index.md)

**4. project:** contains the images and text for the three snippets on the landing page in three separate folders (one, two, three)

*Project folder names aren’t fixed*

5. Project folder (e.g. ‘**one**’): contains featured.jpg, index.Rmd, and index.md. Work with the .Rmd to embed any R content, then knit into the .md, which is what gets served by the landing page.

*‘featured.jpg’ can be substituted for other images but must be called ‘featured’ to appear with the corresponding text snippet on the landing page*