# Projects for 2021

I want to target three languages and tools for learning in 2021: Vue, Go and Electron.

These projects are the ones I've thought about so far. I may have additional ones come up as the year progresses.

This will also enable to create a monorepo for each of these projects that needs one and use Bazel as the build system.

## WordPress in Vue

I've always liked Vue better than React (regardless of the later's popularity and usage) so I started working on a combination of Vue and the WordPress REST API.

See the following blog entries for the work I've already done:

- WordPress + Vue: Building a blog
- Adding JWT to WordPress

Some things to do moving forward

- Technical
  - Upgrade to Nuxt 3.x whenever they update the code to use Vue 3.x
- Components and Features
  - Create a pagination component
  - Create a page component
  - Create a framework to handle custom content types
- Editor
  - Is it possible to create an editor to publish content as a set of Vue components
  - Is it worth the effort
  - Does it makes more sense to use the WordPress backend for publishing content?

## Ebook Manifest generator (Go)

One of the earlier projects I did in Python (and the reason why I deeply dislike

Stack Overflow) was an epub ebook generator.

What would it take to make the same tool in Go? The steps are as follow:

- Read in a JSON file with the required document metadata
- Read a recursive list of files in the directory
- Match each file to its mime type, either with a map or a built-in library
- Write the file pulling from the JSON file and the list of file and mime types

The idea is that the resulting file should pass validation.

We can also compile this to WASM to run in the browser or in an Electron applicaation.

# Content Editor (Electron?)

I think web technologies are at the point where we can create applications that do most of what we want.

Starting with a basic Electron application the idea is to follow these steps in roughly this order:

1. Work with Typescript from the start
2. User the Monaco text editor as the core editor for the following formats
    1. Markdown
    2. CSS
    3. SCSS
    4. HTML
    5. XML
    6. Javascript
3. Handle file management tasks with the File System Access API
4. Export Markdown to HTML
5. Allow packaging of content into zipped bundles, epub3, and web bundles
6. Research what it would take to convert the web bundler tool and the ebook manifest generator into WASM so it can be used directly in the app

This is just the basics. It needs more thorough analysis and a more rigurous design document.

# xAPI Profile and Library (Typescript)

I wrote a series of posts about building an xAPI profile.

The project seeks to understand what would it take to build these xAPI statement to send to an LRS? as new tools and articles from Rustici Software in the coming year. The specific stems may include:

1. Setup an LRS for development or work with SCORM cloud as the development system
2. Build a profile for developers
    1. Take into account the most common features that we expect them to do
3. Build statements to send to the LRS
    1. Group these statements into a reusable library
4. Implement the library on a sample static site
5. Implement the library as a WordPress plugin or directly on a theme

# Vue.js Tutorial for the absolute beginner

Frameworks in general and the big three frameworks (Angular, React and Vue) in particular are hard to break into and assume knowledge that may or may not be there.

The idea is to write a project-based series of articles on how to work with Vue, assuming no prior knowledge and some basic HTML, CSS, and Javascript.

The only thing that this needs is a project to work for these posts. This is different than the blog project.