**Project Proposal: Visual JSON Editor**

1. **Team (Solo)**: Austin Slaughter | [a@slaug.cc](mailto:a@slaug.cc) | @slaugaus on most platforms
2. **Stakeholders**:
   1. Me (see 01)
   2. Peers receptive to me showing them things I made, such as this class and the CSE Drop-In Tutoring Lab I volunteer for
   3. My wife, Melissa, who cares that I succeed but doesn’t have a use for this project
3. **Abstract & Purpose:** Visual JSON Editor is a Visual Studio Code (VS Code) extension that allows you to open JSON (JavaScript Object Notation) files in a WYSIWIG (What You See is What You Get) editor tab. With its heavy use of braces, quotes, and nesting, JSON can often be a pain to read/write by hand. With this extension, I aim to create a useful visualizer and editor for JSON files that is superior to handwriting it, especially for beginners.
4. **Background/Prior Knowledge**: VS Code is my “go-to” editor for code projects that I don’t need a more complex IDE for. I have briefly dabbled in its extension API to add a simple command (action that can be set to a keyboard shortcut), specifically one that removes the word “TODO” from a comment. (I never got around to releasing this.) This extension API is well-documented and allows for the addition of “custom editors” with full control of a file using TypeScript or JavaScript.
   1. **Prior work by others**: [svelte-json-editor library by josdejong](https://github.com/josdejong/svelte-jsoneditor), [vscode-json-editor by sunmorgus](https://github.com/sunmorgus/vscode-json-editor), and [vscode-settings-editor by liriliri](https://github.com/liriliri/vscode-settings-editor) are three notable ones I’ve found, but none are exactly what I want. I pledge not to read their source code.
   2. **Prior work by me**: A web-based prototype, made recently to confirm that HTML and TypeScript were viable for this project, is available at <https://slaug.cc/visual-json-editor-vscode/> ([source code at time of writing](https://github.com/slaugaus/visual-json-editor-vscode/tree/32593ffea4e648ff088142e19d5ab012ea60512a)).
   3. **Foundational courses:** CSE 121B (JavaScript Language) ends with a miniature “capstone” project, in which I dynamically generated the tiers of a basic incremental game. (code [here](https://github.com/slaugaus/scuffed-clicker/blob/main/main.js), the dynamic part being the two for loops.)
5. **Description:** Like the prior work by myself and others, this project will build a GUI based on a JSON file with the purpose of editing the data it represents. I want it to differ from the alternatives by focusing on useful editors for common types of data, such as date/times, and making it look like a program’s settings screen. I will admit that I’m unlikely to surpass or even compete with existing alternatives, but ultimately, I want to have done this for the experience. My target audience is therefore myself, followed by VS Code users searching for “json editor” extensions who find some reason to use this one. (In some sense, the target audience would also be prospective employers.) The project will be “done” in the context of this class when I’m satisfied enough to publish it to the VS Code Marketplace as an extension anyone can download. I then wish to continue developing it after this semester.
6. **Significance:** My priorities with this project are to impress prospective employers first and benefit the developer community second. This project can be used on my resume to demonstrate proficiency in adding on to existing software and self-guided learning, both essential qualities of working on large software systems.
7. **New Concepts:** I am proficient in the fundamentals of JavaScript and TypeScript thanks to their documentation and my proficiency in similar languages, but I have not immersed myself substantially in their ecosystem. VS Code extensions use Node.js, which I have similarly not experienced outside of its fundamentals. I am also new to formally following the UI/UX guidelines of an existing project, i.e. VS Code’s (which I would like to do here). I’d also like to set up a CI/CD pipeline, at the very least automatic unit tests.
8. **Interestingness/Motivation:** There’s a “coolness” to generating UI on the fly that I can’t quite explain. Perhaps it has to do with most of my experience being in languages and frameworks where this would be significantly harder? I am also motivated to uphold my perfect GPA and have a cool project on my resume, of course.
9. **Milestones**
   1. As week 14 is the week everything should be wrapped up by, I have allocated the hours differently from “9 hours a week.”

|  |  |  |  |
| --- | --- | --- | --- |
| **Week #, Special Identifier** | **Hours To Spend** | **Task #1 (Assignments)** | **Task #2 (Development)** |
| 3 (General Conference) | 9 | Brainstorm requirements | Viewer prototype loads as-is in VS Code |
| 4 | 10 | Prepare & submit W04 Initial Requirements | Viewer prototype loads the file it’s supposed to + start implementing editing |
| 5 | 9 |  | Make editing keys and values work |
| 6 | 10 | Outline final requirements | Adding and removing data entries |
| 7 (Proto, Reqs Due) | 10 | Prepare & submit final requirements | Record prototype video |
| 8 | 9 |  | Epic Feature #1: Detect data types inside strings & edit them accordingly (think color, date pickers) |
| 9 | 10 | Prepare SPED Talk |
| 10 (Post SPED Talk) | 10 |  |
| 11 (Thanksgiving) | 9 | Epic Feature #2: Make it look good | |
| 12 | 10 |
| 13 | 10 | Deploy extension to VS Code Marketplace | Bug squashing? |
| 14 (Semester End) | 2 | Final Meeting With Instructor | Finishing touches? |

1. **Resources:** This list lacks specificity as I am very much a “learn as you go” person. I do not expect or intend to use resources or dependencies that cost money.
   1. TypeScript documentation (<https://www.typescriptlang.org/docs/>)
   2. W3Schools and Mozilla HTML documentation
   3. VS Code extension API documentation (<https://code.visualstudio.com/api>)
   4. Compilation and packaging software from the TypeScript, Node.js, and VS Code extension ecosystems
2. **Dependencies:** Again, I expect and intend to use exclusively free things.
   1. Core Languages: TypeScript, HTML, possibly CSS
   2. Other Code: Careful selection of libraries that enhance the project without trivializing my development of it. All will be credited properly no later than the end of the project.
      1. VS Code Extension API
   3. IDE: Visual Studio Code and the extensions I use for it
   4. Platform: Also Visual Studio Code. Target is VS Code for Windows but the project should function on any platform VS Code can be installed to.
3. **Risks:** 
   1. I’d like the project to use or look like the native VS Code interface. Limitations of the extension API may make this impossible.
   2. Internet outages may hinder my progress, if not stop it. The same for power outages.
   3. Any component of the project being much harder/time-consuming than my assumptions
   4. Procrastination
   5. Impostor Syndrome