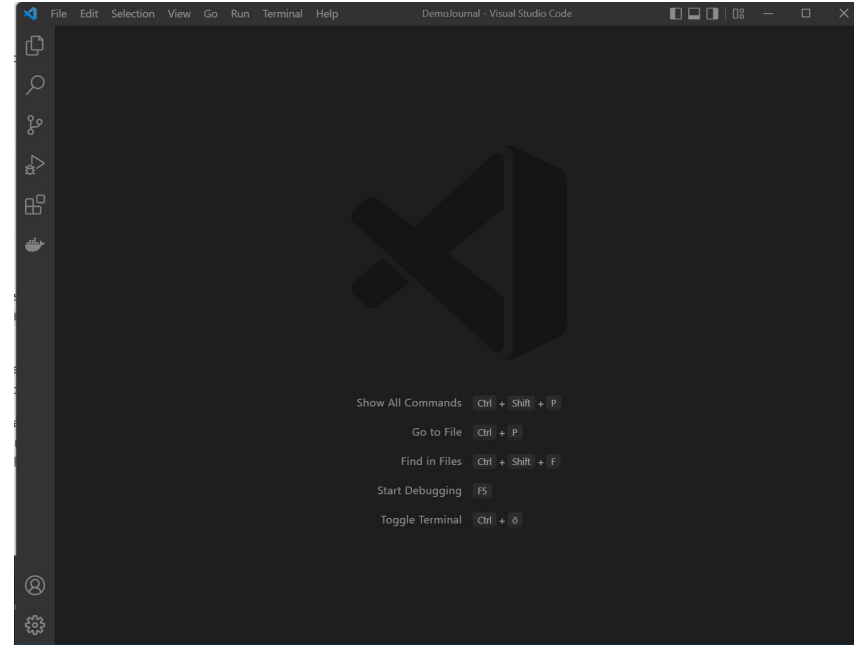# Final Project Pitch

– VScode journal extension

8xEngineers (Team 12)

Statement of Purpose

In the dynamic field of software development, efficiency and organization are paramount. Developers face the challenge of managing multiple projects simultaneously, often feeling overwhelmed by scattered tasks and notes across diverse tools. The VSCode Journal Extension aims to address these challenges by providing a seamless, integrated solution that facilitates the tracking of tasks and documentation directly within the Visual Studio Code environment.
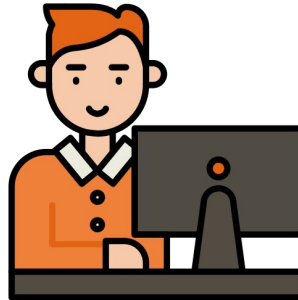
Statement of Purpose

This extension targets individuals who:

- Need an organized way to track their development progress *without leaving their code editor.*
- Want to *quickly* record bugs, code snippets, and ideas that can be easily referenced later.
- Enjoy working with integrated tools and prefer to have *everything they need in one place*.
- Work in a *collaborative environment* and want to document important decisions or issues for team members.
- Needs to streamline their workflow by having the ability to track progress, capture ideas, and document key information without interrupting their coding process .

User Persona and Scenarios

Diego, a mid-level developer working on several projects at once, often feels stressed by having to keep track of tasks and notes across different tools. His main goal is to make his workflow smoother within his coding environment, so he doesn't have to switch between different applications so much. To do this, Diego wants to use a single system where he can easily write daily notes and keep an eye on his project tasks. He organizes his work by categorizing client tasks separately from personal notes, which helps keep things tidy and clear. This setup not only keeps him focused but also makes it simpler to find past work and organize his thoughts. This helps him work more efficiently and reduces stress.

User Persona and Scenarios

Diego, a mid-level developer managing multiple projects, starts his workday by organizing his tasks using the VSCode Journal Extension. He quickly accesses the **Task and Issues view** from the sidebar, where he plans out his day by categorizing tasks under specific project scopes directly within the dashboard. This helps him keep all tasks clearly organized and accessible, **eliminating the need to switch between different tools and allowing him to focus more on coding**.

As he progresses through his coding sessions, Diego encounters new ideas or issues that need attention. **With the file tagging feature** of the extension, he easily tags files in the navigation bar under relevant categories and drops quick notes into the integrated task and issues view. This setup maintains a clutter-free workspace, with all related notes and files neatly organized. Diego enjoys the convenience of **having all essential information retrievable from one place**, which significantly boosts his efficiency.
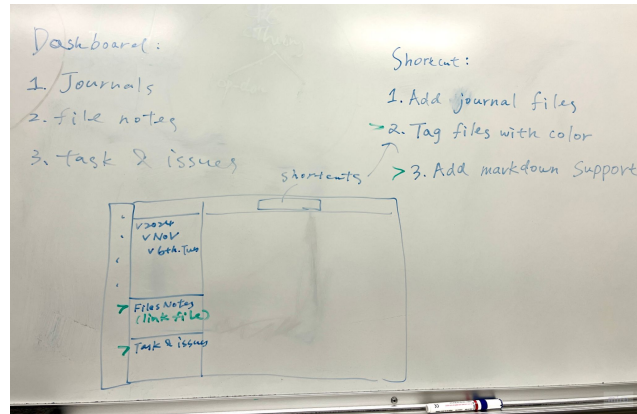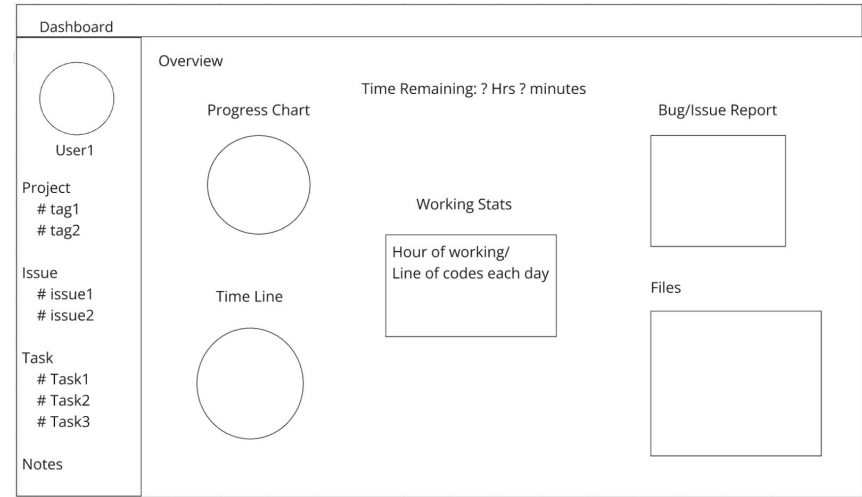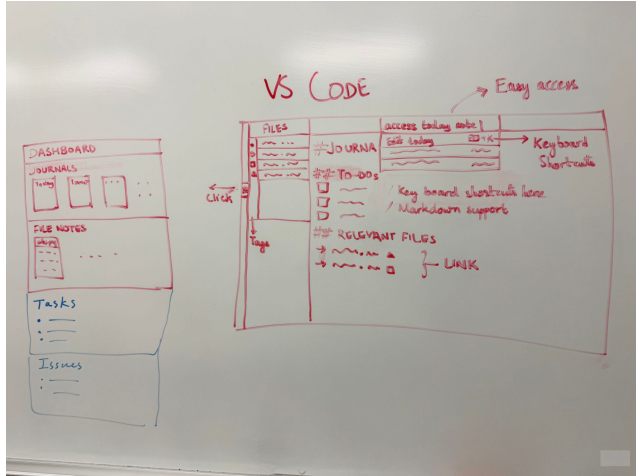
At the end of each week, Diego uses the same extension to review and compile his work. He filters the view to show only the information relevant to each project, allowing him to **navigate through tagged files and consolidated notes effortlessly**. With all the data well-organized, he efficiently prepares for client meetings and team updates. This streamlined process not only makes these meetings more productive but also ensures that no detail is overlooked. As a result, Diego's preparation time is significantly reduced, freeing up more hours for actual development work.

What are we doing then?

Key target useful features from the personas

- **File tagging** - Users click beside file names in the navigation bar to tag the files in categories

- **Task and Issues view** - Append task and issues list in the extension dashboard (one click

  from the sidebar) where users can see their daily notes as well.

- **A one-place solution** - A plugin where the user can find all the features in one place.

Sketches and Wireframe

Topics of Risks - User problems

- May not be easy to use for users just starting out with VS code

- Takes some time to get used to navigating around the plugin

- Does it have more value over documentation and other tools? Is it just over-complicating the solution to a simple problem?

Topics of Risks - A developer perspective

- Merging two plugins could be arduous.

- May be aiming for a lot more than can be done in the remaining time
  - May not finish all the planned features

- Unfamiliarities with VSCode plugin development.

Plan Ahead

- Spend one week digging into the codebase
  - Understand the files and functionalities
  - Start merging the codebases - BLOCKER
- During the next week, split tasks pairwise
  - Build the functionalities - markdown, task view and file tagging
  - Build unit tests to test the functionalities
- Follow CI/CD to review and compile everything together