CodeSnip

Developer Productivity Tool

Team # 13

CSE 210

Nov 6, 2024

Statement Of Purpose

Developers often create reusable code snippets for common programming tasks. These snippets are extremely useful, but can often be disorganized, difficult to locate, or scattered across multiple systems and tools. Building upon an existing code snippet management VSCode extension, our goal is to create a user-friendly platform that allows developers to efficiently organize, access, reuse and share their code snippets with others, enhancing productivity.

Vision

To leverage our academic knowledge and innovative thinking to create impactful, practical software solutions. As a team of passionate learners, we're focused on improving our skills, working together, and delivering projects that make a tangible difference in a real world context.

User Personas

Alex Torres, 28

Frontend Developer at a SaaS company

San Francisco, CA

Needs

- Increased Coding Speed: Wants to accelerate workflow by reducing repetitive tasks.
- Staying Organized: Believes in a clean, well-documented codebase and needs tools that facilitate this.
- Learning and Growth: Interested in improving skills and exploring best practices by keeping organized, optimized snippets.

Goals

- Efficiency: Alex wants to minimize time spent looking up repetitive code snippets or re-writing common utilities.
- Code Reusability: They believe in DRY (Don't Repeat Yourself) principles and want an organized, searchable way to store and reuse their code snippets.
- Seamless Integration: Alex uses VSCode as their main editor and wants tools that integrate smoothly into their existing workflow without added friction.

Pain points

- Lost Productivity: Spending time searching for code snippets impacts productivity, especially when deadlines are tight.
- Inconsistent Snippets: Alex sometimes accidentally re-writes code they've already solved before.
- Lack of Standardization: Alex's current snippet organization lacks standard tagging, making it hard to quickly find what they need.

Jorden Chen, 32

Senior Software Engineer at a large e-commerce company

Seattle, WA

Needs

- Continuous Improvement: Passionate about refining and optimizing code for better performance and readability.
- Team Collaboration: Believes in the power of shared knowledge to elevate the entire team's capabilities
- Work-Life Balance: Seeks to reduce overtime by increasing productivity during regular work hours.

Goals

- Efficiency: Streamline development process by reducing time spent on repetitive coding tasks.
- Knowledge Sharing: Facilitate easy sharing of best practices and optimized solutions within the team.
- Cross-Project Consistency: Maintain consistent coding patterns across different projects and services.

Pain points

- Time Waste: Frequently recreating solutions for common problems that have been solved before.
- Inconsistency: Different team members implementing similar functionalities in varying ways, leading to maintenance issues
- Knowledge Silos: Valuable code solutions often remain isolated within individual projects or developers' personal collections.

Taylor Smith, 30

Full-Stack Developer at a tech startup

Los Angeles, CA

Needs

- Streamlining Workflow: Taylor aims to save time and reduce the need to search for or rewrite code snippets.
- Productivity Anywhere: Maintaining steady output, even without internet access, is essential for Taylor's work.
- Reduced Mental Effort: Having well-organized and clearly documented snippets helps Taylor focus on writing new code.

Goals

- Organized Snippet Storage: Taylor wants an efficient system for saving and organizing both past and new code snippets in the cloud, making them easy to find and use.
- Clear Snippet Documentation: Attaching useful context, like descriptions and use cases, to each snippet is important for future reference.
- Offline Access: Having access to essential code snippets without needing an internet connection is critical for consistent productivity.

Pain points

- Version Control of Snippets: Maintaining different versions for various use cases is cumbersome.
- Snippet Context and Documentation: Understanding the context of older snippets becomes difficult without attached explanations, causing delays in development.
- Offline Access: Productivity suffers when internet connectivity is limited, as Chris cannot retrieve necessary snippets from the cloud.

Jimothy Jorange, 19

Sophomore at San Jose State University studying computer science.

San Jose, California

Needs

- Jimothy often codes on the go, but prefers to code on his computer at home. He needs a place to store and transfer his code between his various devices.
- As Jimothy learns new concept in his CS classes, he often returns to previous code and revises it. He would like to be able to see previous versions of his code to see how far he has progressed. anywhere.

Goals

- To be able to easily move his code snippets between his desktop and his laptop
- To have a place to store his code for easy access whenever he needs his often-used snippets
- To have a way to view past versions of code snippets in order to see how much his coding has evolved through his studies

Pain points

- Jimothy currently transfers his code using emails and downloading, which is slow and messy
- He often loses track of his various code snippets in his emails and different devices

User Scenarios

Alex Torres, 28

Goal

To enable Alex to utilize previously created code snippets while working on a new project in Visual Studio Code.

Tools

VS Code, Code Snippet Manager

Scenario

Alex starts a new project in Visual Studio Code and needs access to several previously created snippets. He downloads them from his code snippet manager as a package that integrates directly into VS Code. This integration allows him to access and insert his snippets easily within the editor, streamlining his workflow and enhancing productivity as he develops his project.

Jorden Chen, 32

Goal

To enable Jordan to access his code snippets across multiple platforms and to make the snippets shareable across platforms.

Tools

VS Code, Code Snippet Manager, iPad, E-mail client

Scenario

During a lunch break at a restaurant, Jorden receives an urgent email from a manager asking for a Python script that formats JSON responses for an API. Jorden accesses the platform on his iPad, navigates to the "API Formatting" folder, and copies the script. He replies to the email with the script attached, meeting the manager's request without needing to return to his workstation.

Taylor Smith, 30

Goal

To enable Taylor to documentation and annotations to her code snippets

Tools

VS Code, Code Snippet Manager

Scenario

Taylor is creating a collection of utility functions for her projects. For each snippet, she writes detailed descriptions explaining their purpose, parameters, and return values. She also adds comments within the code to clarify complex logic. When she returns to these snippets months later, she finds them well-documented and easy to understand, enabling her to quickly utilize them in new projects.

Jimothy Jorange, 19

Goal

To enable Jimothy to manage versions of his code snippet

Tools

VS Code, Code Snippet Manager

Scenario

Jimothy implements a bubble sort function and saves it in his code snippet manager. A week later, after learning about more efficient sorting algorithms, he revisits the snippet to optimize it. He makes changes and saves the new version while retaining the original. Later, he reviews the version history to compare his initial implementation with the optimized one, allowing him to reflect on his progress and understanding of algorithms.

Proposed Features

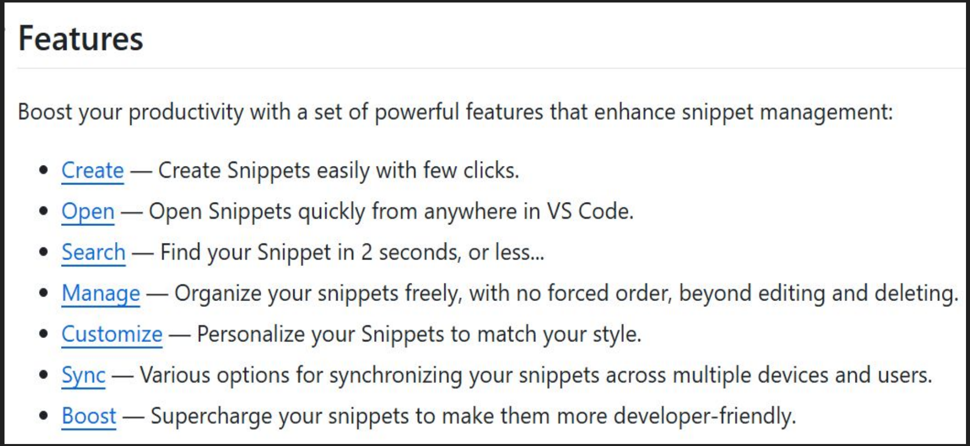
- Sharing the snippets with other developers
- Version control support for snippets
  - Users can access earlier versions of snippets
- Descriptions and comments for snippets
  - Users can add documentation while creating snippets
- Exporting and importing functionality

Nice-to-have Features

- Shortcut Keys
- Snippet previews
  - Hovering will show a short preview

Current VSCode extension features

https://github.com/tahabasri/snippets

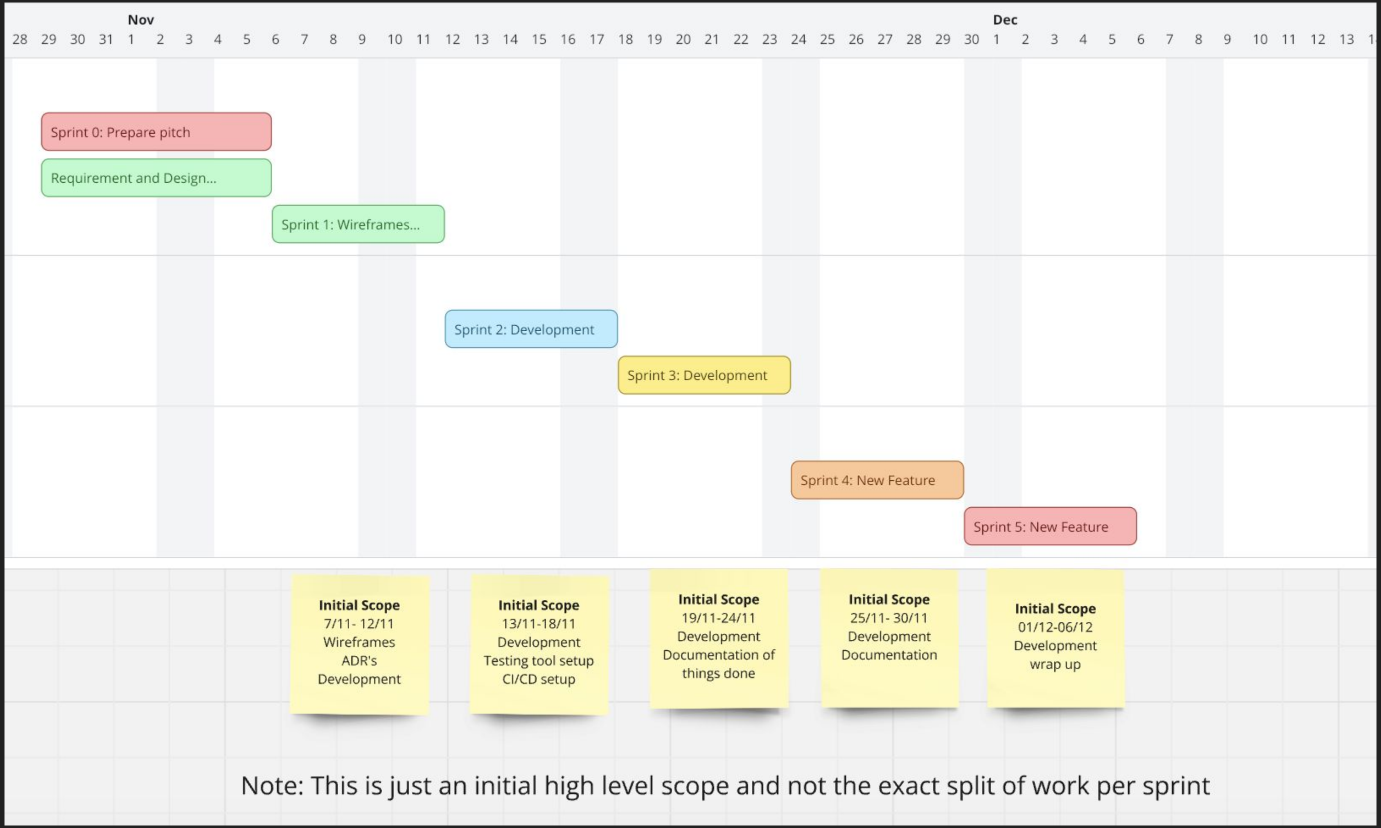


Features

Boost your productivity with a set of powerful features that enhance snippet management:

- Create — Create Snippets easily with few clicks.
- Open — Open Snippets quickly from anywhere in VS Code.
- Search — Find your Snippet in 2 seconds, or less...
- Manage — Organize your snippets freely, with no forced order, beyond editing and deleting.
- Customize — Personalize your Snippets to match your style.
- Sync — Various options for synchronizing your snippets across multiple devices and users.
- Boost — Supercharge your snippets to make them more developer-friendly.

Project Timelines



Initial Scope
7/11- 12/11
Wireframes
ADR's
Development

Initial Scope
13/11-18/11
Development
Testing tool setup
CI/CD setup

Initial Scope
19/11-24/11
Development
Documentation of things done

Initial Scope
25/11- 30/11
Development
Documentation

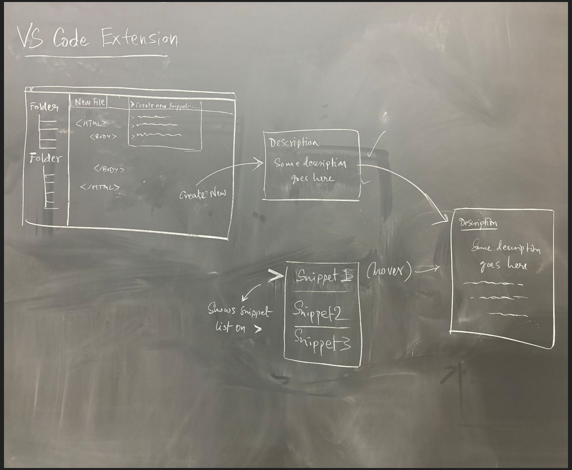
Initial Scope
01/12-06/12
Development
wrap up

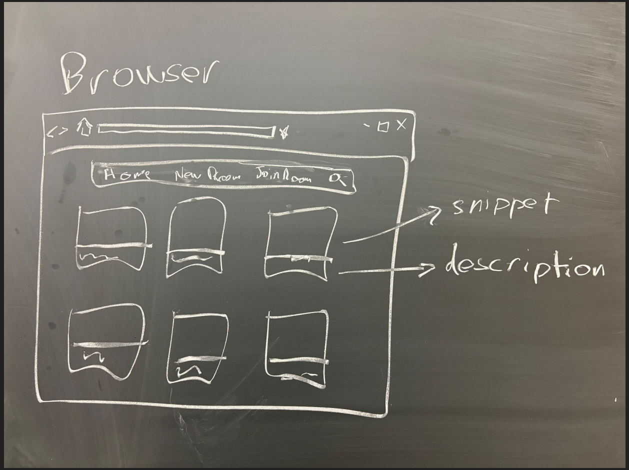
Note: This is just an initial high level scope and not the exact split of work per sprint

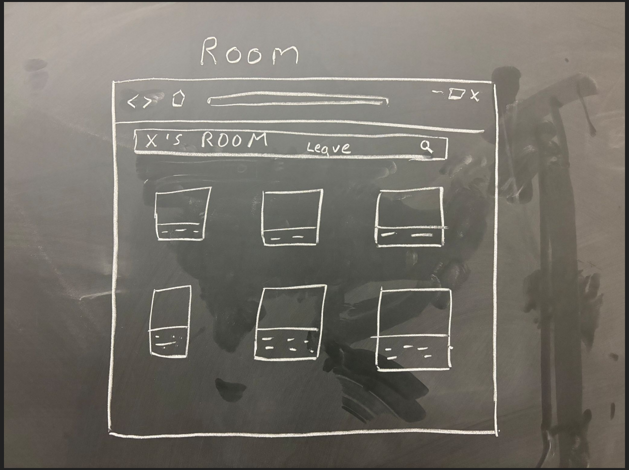
5 Weeks
Development Cycle

7 Devs + 2 Designer

Fat-Marker Sketches







VSCode extension to support:

- Adding descriptions and comments
- View the description on hover
- Version control of snippets

A local browser app to support:

- Management of all the snippets
- Visualization of snippets

Web Application to support:

- Sharing of snippets with users
- Export & import of snippets

MVP v1 → MVP v1.1 → v2

Risks

- Team is unfamiliar with making VS Code extensions
  - The proposed features may not be compatible
- Limited Extendibility of existing codebase
- Cross-Device Compatibility
  - Our team has limited experience, but this is a necessary feature

Rabbit Holes

- Focusing on the UX in the early stages of development
  - Focus on getting the core functionality of the project ready first
- Snippet sharing details (permissions, etc.)
  - First implement an application with no user permissions. Focus on the granular details of authorization in the next version.
- Authentication via OAuth
  - Authentication involves many considerations. Alternatives that were discussed include having a room with a shared passcodes and/or a common data storage point for snippets sharing.

🚫 No-Go’s

Real Time Updates & Collaboration

This feature is outside the scope of this project

- Introduces complexities of synchronization of data between multiple users
- Involves handling concurrent edits, resolving conflicts due to changes, managing access to different versions of same snippet by various users
- We need to implement proper authorization-level permissions and ensure that only authorized users can access and edit snippets
- We haven’t been able to identify a strong enough use-case to justify the development efforts required for this complex feature

Any Questions?