



CSE 210: TEAM 4

Web-based Developer Toolbox

- Samyak Mehta
- Nikhil Gautam
- Yunhao Jiang
- Hailey Li

- Laura Tian
- Jake Norbie
- Adrian Layer
- Tim kraemer
- Anusha Ravichandran



Statement Of Purpose

The Problem:

1. Inefficiencies in building and optimizing common utility functions.
2. Redundancy and repetitive coding tasks.
3. Need for reliable, performance-focused solutions without additional costs.

The Solution:

1. Reduces repetitive coding, allowing focus on unique project needs.
2. Standardizes utility functions for consistency and reliability.
3. Curated for performance and best practices, eliminating inefficiencies.
4. Ensures seamless compatibility across various environments.

Relevant Research On Similar Products

Similar Developer Toolkits exist, however the majority found are either deprecated or paid options.

- DevUtils

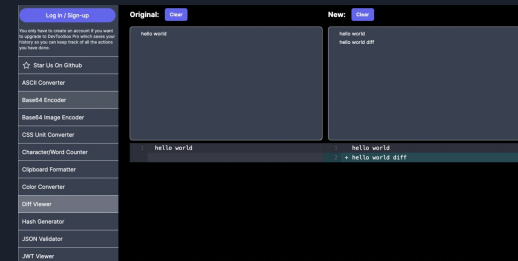
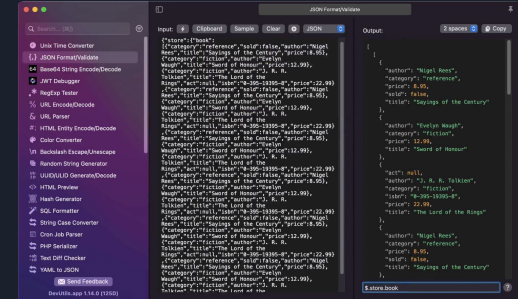
- Pros: Many functions are included; Can show top X frequently used functions
- Cons: Not open-sourced; does not categorize well; no way of saving favorite functions; needs subscription

- Developer Toolbox Chrome Extension

- Pros: Simple Chrome extension, useful in browser
- Cons: Deprecated, small user-base, little amount of tools offered, tools are random

- DevBox

- Pros: Many functions included, includes cheat sheets, real-time updates for tool changes
- Cons: Close-sourced, needs to purchase, needs to download a separate application



User Experience

Home Page

Action: User lands on the dashboard with all developer tools listed.

Next Step: User clicks a tool.

Tool Loading and Interaction

Action: Selected tool loads on the same page, allowing immediate interaction.

Options: Side panel remains accessible for switching between tools.

Next Step: User performs actions within the tool.

Tool Submission and Results

Action: User submits data within the tool and sees results displayed on the same page.

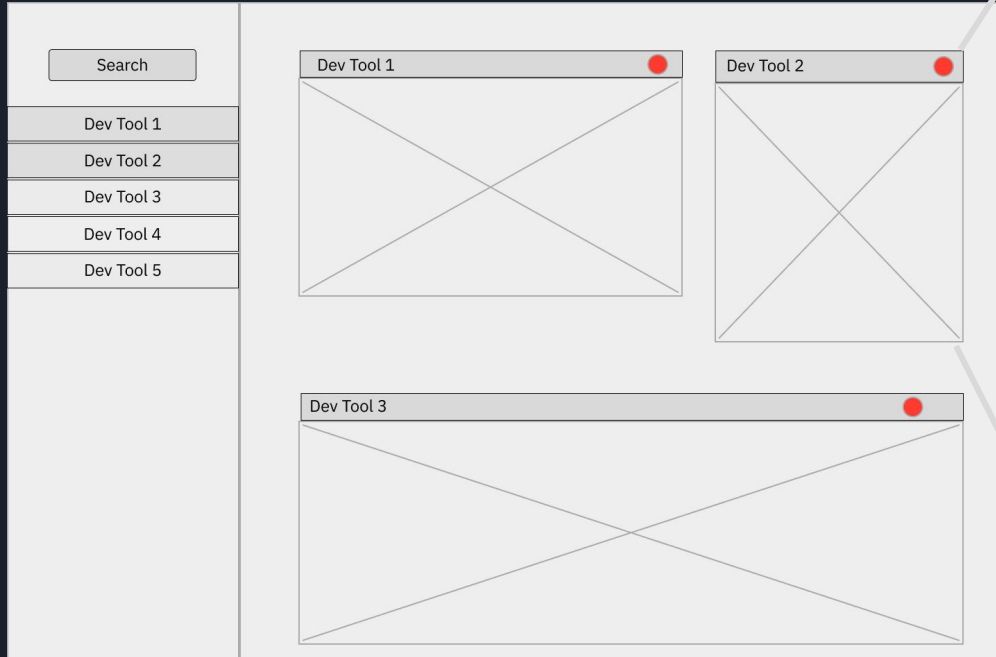
Options: Clear submission or switch tools from the side panel.

Next Step: User may clear results and retry, choose another tool, or exit.

End / Restart

Option: User exits or returns to tool selection on the dashboard.

System Design





Potential Risks and Rabbit Holes

- **Become too focused on individual tools/components and fail to consider the bigger picture**
 - Danger to cohesiveness
 - Components and tools might not fit one vision
 - All our tools should cater to a set of skills that play off each other
 - It would be better to market our toolbox to a specialized group
- **Be too ambitious in what tools we want to implement**
 - Although certain tools can be cool to implement, they might be too complex with the time that we have - it's a real possibility with the modular structure of this tool that we can spread ourselves too thin if we choose the wrong tools.
 - Our group has limited front end experience
 - Cannot understate that the design aspect of our site and making it feel good will probably take longer than expected.
- **Oversimplifying the modularity of the top level web-app architecture**
 - The top level web-app seems simple to implement due to tool modularity, which might inadvertently cause us to overlook important aspects



Tools

Front-end

- Markup
- CSS
- JavaScript

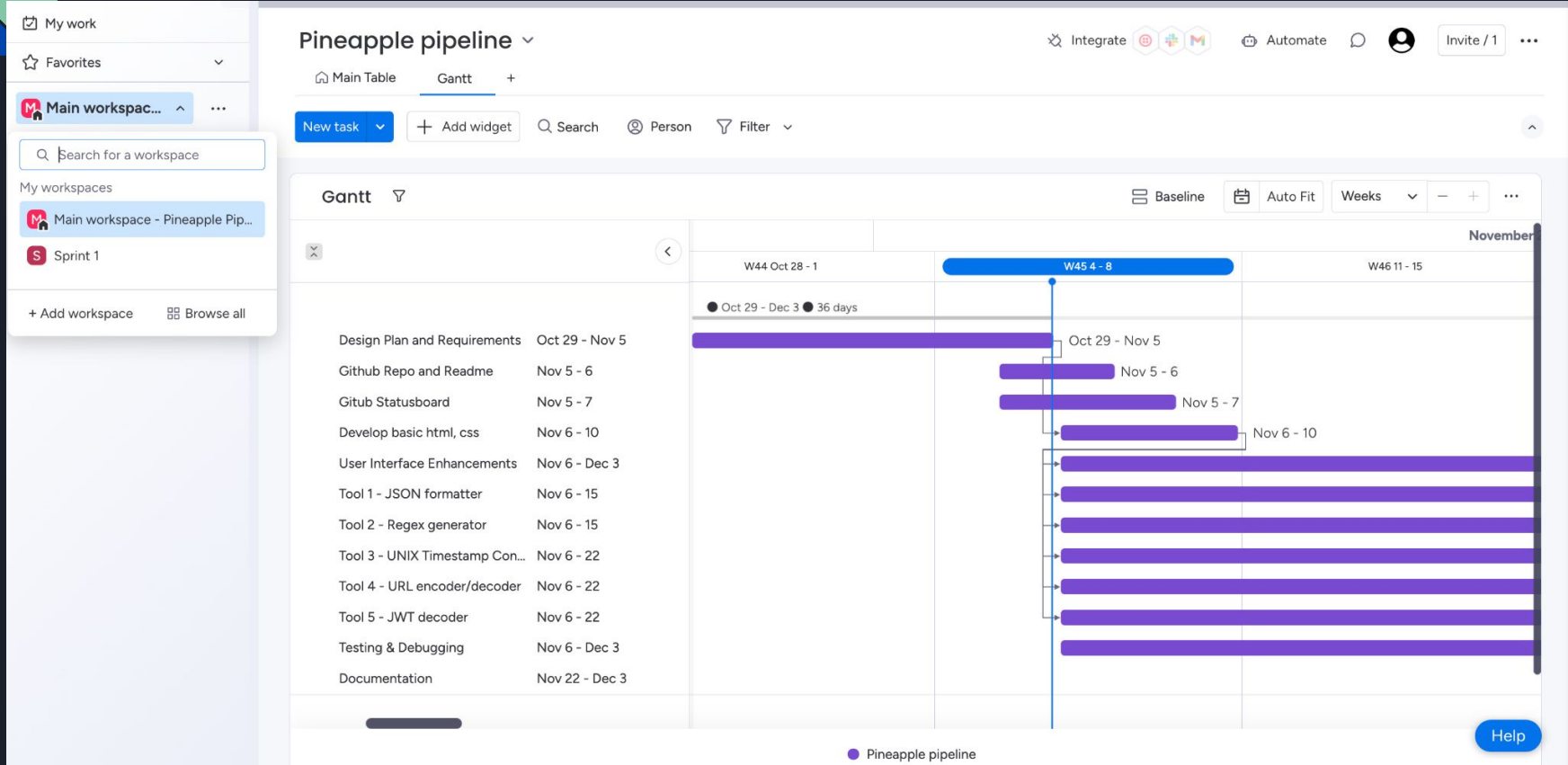
Backend

- Javascript
- NodeJS
- Not storing any user information, so no database needed

Design

- Figma for prototyping
- Miro for brainstorming, design ideas, etc.
- Monday.com for project and workflow management. We'll be migrating to github's status board and roadmap soon.

Proposed Project Timeline





ANY QUESTIONS?