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, paid, or incredibly general.

- **DevUtils**
  - Pros: Many functions are included; Can show top X frequently used functions
  - Cons: Has a ton of tools but no categorization or way of saving favorite functions; needs subscription
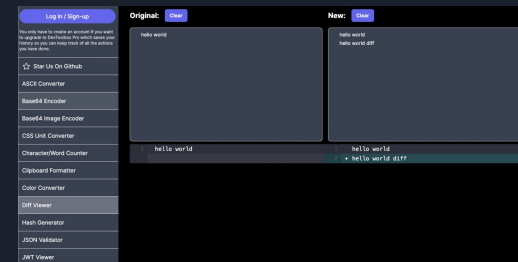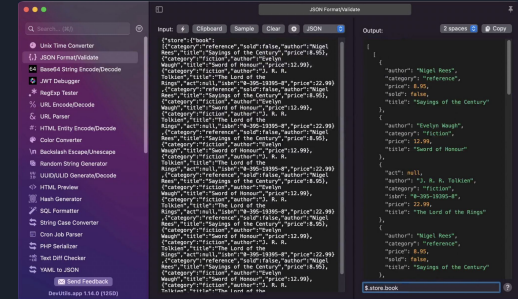- **Developer Toolbox Chrome Extension**
  - Pros: Simple Chrome extension, useful in browser
  - Cons: Deprecated, small user-base, little amount of tools offered, tools are random
- **DevBox**
  - Pros: Many functions included, includes cheat sheets, real-time updates for tool changes
  - Cons: Close-sourced, needs to purchase, needs to download a separate application
- **DeveloperToolbox.app**
  - Pros: Free, plenty of useful tools, simple design
  - Cons: Paid dark mode, too general

User Experience

Home Page → **Tool Loading and Interaction** → **Tool Submission and Results** → **End / Restart**

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

Next Step: User clicks a tool.

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.

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

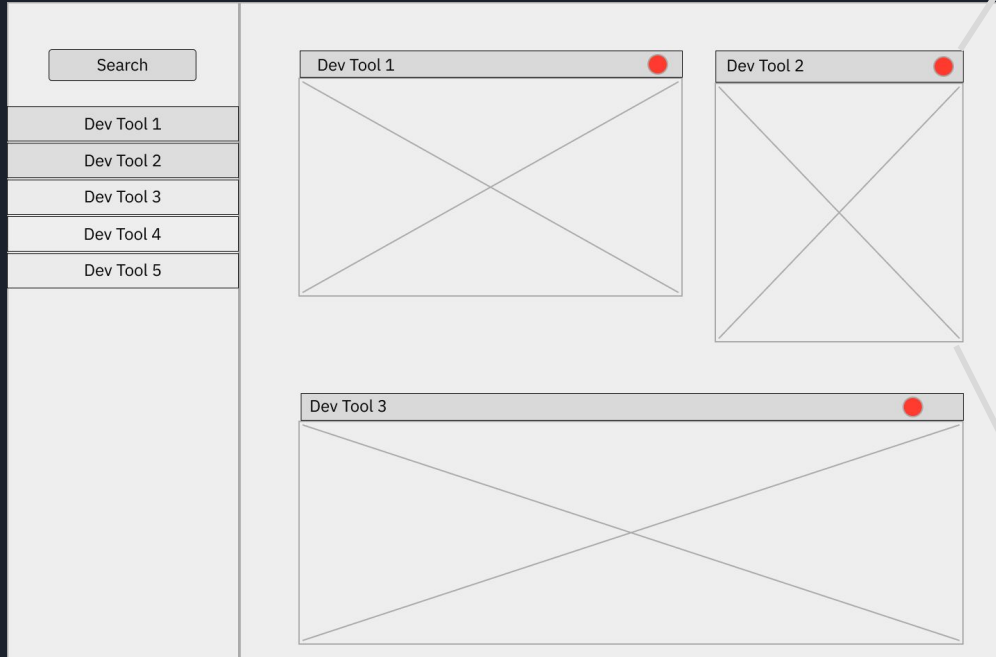**Options:** Clear submission or switch tools from the side panel.

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

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

System Design

JSON Formatter & Validater

{"store":{"book":[{"category":"reference","author":"Nigel Rees","title":"Sayings of the Century","price":8.95},{"category":"fiction","author":"Evelyn Waugh","title":"Sword of Honour","price":12.99},{"category":"fiction","author":"J. R. R. Tolkien","title":"The Lord of the Rings","isbn":"0-395-19395-8","price":22.99}],"bicycle":{"color":"red","price":19.95}}}

Parse

```
{
  "store":{
    "book":[
      {
        "category":"reference",
        "author":"Nigel Rees",
        "title":"Sayings of the Century",
        "price":8.95
      },
      {
        "category":"fiction",
        "author":"Evelyn Waugh",
        "title":"Sword of Honour",
        "price":12.99
      },
      {
        "category":"fiction",
        "author":"J. R. R. Tolkien",
        "title":"The Lord of the Rings",
        "isbn":"0-395-19395-8",
        "price":22.99
```

Copy

Search

Dev Tool 1
Dev Tool 2
Dev Tool 3
Dev Tool 4
Dev Tool 5

Dev Tool 1

Dev Tool 2

Dev Tool 3

Features

The features will be a combination of back-end development tools that are commonly used by developers. More features will be added in a modular fashion if time permitting.

Priority

- JSON Formatter
- Regex Generator
- URL encoder/decoder
- UNIX Timestamp Convertor

Time Permitting

- JWT Decoder
- Hash Generator
- SQL Converter

Potential Risks and Rabbit Holes

- **Become too focused on individual tools/components and fail to consider the bigger picture**
  - Cohesiveness in tools
  - One vision

- **Be too ambitious in what tools we want to implement**
  - Overly complex tools
  - Our group has limited front end experience
  - Too many tools

- **Underestimating how much time it will take to put everything together at the top level**

Tools

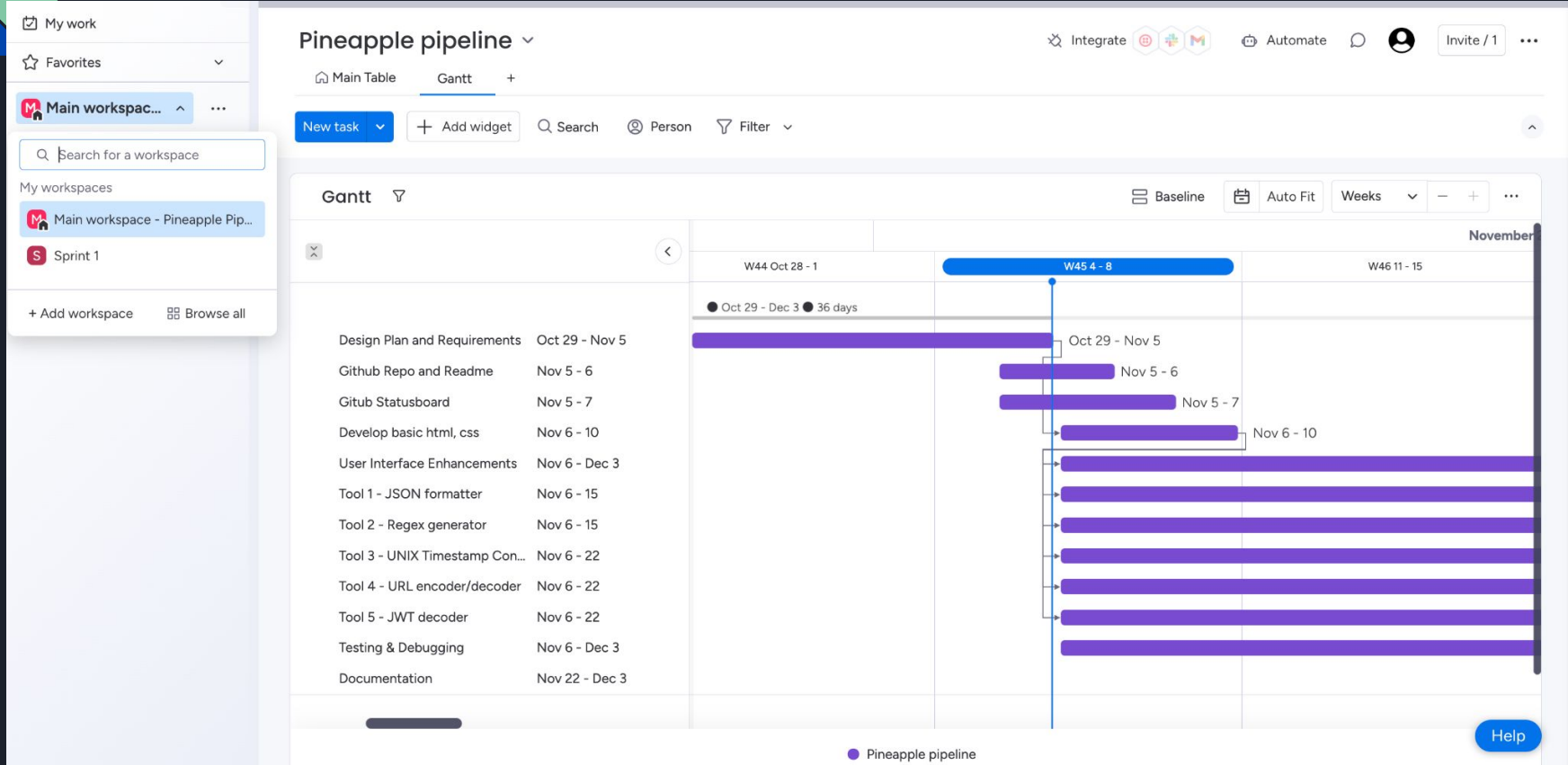| Front-end	Backend	Testing	Design
- Markup - CSS - JavaScript	- Javascript - NodeJS - Not storing any user information, so no database needed	- Jest	- Figma for prototyping - Miro for brainstorming, design ideas, etc. - Monday.com for project and workflow management.

Proposed Project Timeline

ANY QUESTIONS?

[Link to our Miro Dashboard](#)

Proposed Tech Stack

Front-end

- Markup, CSS, JavaScript

Back-end

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

Design

- Figma for prototyping
- Miro for brainstorming, design ideas, etc.

User Experience

- **Fully accessible via browser**
    - Save them the storage space and hassle of an app
    - Low commitment
    - Aids accessibility
- **No login required**
    - Ensures user security and privacy
    - Does not store user information
- **Allows for Customization**
    - tools/components are draggable and can be moved for ease of use
    - Remove and add tools on dashboard
        - Remove useless clutter and prioritize most important tools at the moment
- **Utility**
    - Made for backend developers by backend developers
        - Specialized tooling and features catered specifically for this group

Statement Of Purpose

The purpose of this dev toolbox is to address the inefficiencies and redundancy backend developers face when implementing and optimizing common utility functions. By providing a curated set of optimized, reliable, and high-quality functions **tailored specifically for backend development needs**, the toolbox enables developers to save valuable time and maintain consistency across projects.

Focusing on performance and best practices, the toolbox serves as a central resource for high-use utilities, reducing the need for repetitive coding while ensuring cross-platform compatibility. This solution **prioritizes accessibility**, allowing developers to concentrate on the unique aspects of their projects rather than paying for unnecessary subscriptions, downloading apps, or reinventing foundational tools.