Team 7
Team Name: Akatsuki 7
GitHub Repo: https://github.com/cse110-sp24-group7/warmup-exercise
Youtube Link: https://www.youtube.com/watch?v=7UkRRGBkEec
SWOT Analysis:
https://docs.google.com/document/d/1N82GoM1R-T9xykzWgvNZo0JEvexnpx2wjImGjumuT_0/edit?usp=sharing

Akatsuki 7 - SWOT Analysis

Please take a look below for our motivations/analysis

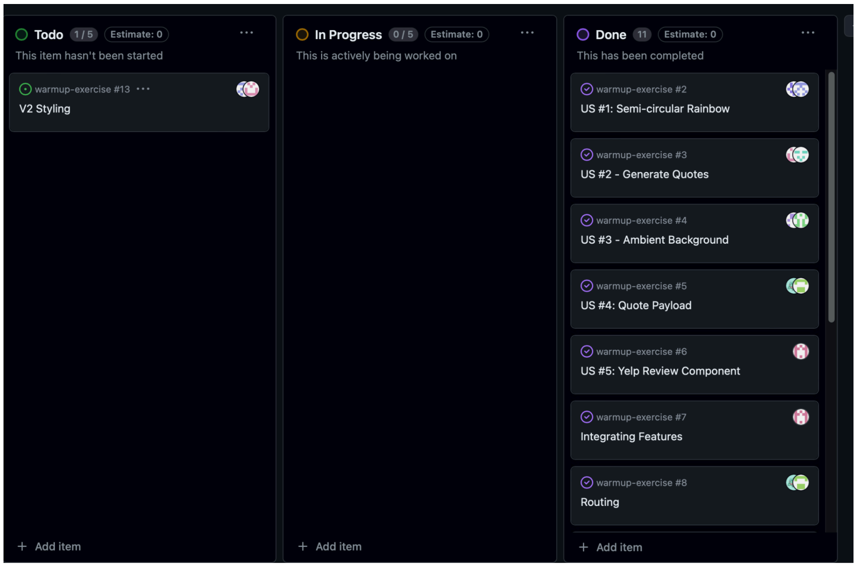
Strengths - Team of Dynamic Individuals - Heavy focus on documentation and software engineering processes - Specialization in animations - Focus on GitHub Workflows - Emphasis on cleanliness of developer workspace	Weaknesses - Clashing Schedules - Lack of parallel development - Lack of formal design experience
Opportunities - GitHub Actions - Better Repo Structure - More Parallelized Tasks - Better usage of GitHub Projects - Segmented Development - Testing	Threats - Increased academic workload leading to more scheduling conflicts - Deadlock due to lack of parallelization - Suboptimal development due to lack of UX familiarity

Strengths:

We find ourselves having a lot of *dynamic individuals*, able to adapt to new situations and new techstacks. None of us have formally experimented with Javascript or vanilla HTML and CSS → However in the span of **4 days**, we were able to come up with a fully client-side rendering app, with UX considerations and routing from different pages.

Many of our developers specialize in *animations*, we were able to animate pop-ups, a vibrant background, and render in **Yelp Reviews** and bubbled them up.

Our team leads also focused heavily on **documentation** and the **software engineering process**. All Github Issues were tied together with Assigned Pull Requests, reviewers were manually assigned by team leads, and a consistent CI process was always employed. After each feature, **2 developers** would review each PR that was tied with a specific issue. This was tethered to a Github Project that focused on implementing different components of the widget, that was *templated,* and *well-documented.* (See Image for reference)



Todo 1 / 5 Estimate: 0
This item hasn't been started

warmup-exercise #13
V2 Styling

Add item

In Progress 0 / 5 Estimate: 0
This is actively being worked on

Add item

Done 11 Estimate: 0
This has been completed

warmup-exercise #2
US #1: Semi-circular Rainbow

warmup-exercise #3
US #2 - Generate Quotes

warmup-exercise #4
US #3 - Ambient Background

warmup-exercise #5
US #4: Quote Payload

warmup-exercise #6
US #5: Yelp Review Component

warmup-exercise #7
Integrating Features

warmup-exercise #8
Routing

Add item

Our team also heavily focused on **Github Workflows** and we were able to successfully set up Linting without cluttering too much on our Github Repository. As such, all node-dependencies were installed Developer side and we were able to successfully run jobs on the cloud. PR Requests had a ruleset where **2 reviewers were** required to review each PR(or Codeowners) and Pull Requests must satisfy all code quality requirements before being merged with main. We also lended to *squash merging* branches to lessen the commit-log on the main branch. Coupled with all of this yielded a **clean and effective** workspace for development. (Developers also had access to a Togglable workflow that they could run on their branch, that would run *prettier* checks for them).

Weaknesses:

Though our team worked exceptionally well in creating this widget, seamlessly combining unique skill sets to create a great product, we are not without our weaknesses.

One issue that we'd like to work on going forward is **scheduling tasks better** throughout the project's timeline. While we had many issues hashed out from the get-go and a great template on how to go about creating the widget, the developers' **schedules frequently clashed** which hindered our progress substantially. Additionally, most work was done sequentially, leading to a **lack of parallel development**.

Another setback we ran into was the **lack of formal design experience** among the team. When building a feature that was heavily frontend-based, we found that no one on the team was well-versed with Figma or building apps. However, we took this into stride and used the best tools we had available to us within the timeframe requested. Many of our most creative minds worked together to bring out a design template for the widget that we think looks beautiful from a user perspective.

Opportunities:

Our team lacked experience in designing responsive and effective web applications, however we see this as an **opportunity for growth**. We made basic wireframes using MIRO, as many of us were not familiar with particular design systems, wireframes, and reusable components. This *ad hoc* approach provided a natural growth, but also slightly disorganized flow of development. At Akatsuki 7, we see this as an opportunity for us to become more familiar with eye-pleasing web apps. We aim to use resources provided by our TA, Akshay Prabhu ,such as **Dribble, Figma**, and **Pinterest** to gain inspiration for user-friendly web apps, as a medium for more **User Centric Development**(UCD).

We also aim to timeline and document more of our processes. Currently we lended more into the **Development Process** without focusing too much on sustainability. While we continuously integrated over this iteration, we didn't focus too much on testing, developing unit tests using **Jest** and other frameworks would have been incredibly valuable for the development process, as a way to both sanity check our code, and provide us with the confidence to continue adding to the codebase. Leveraging our experience from writing Github Actions, we already have the experience to write **workflows on the cloud**, and we can surely use another workflow as a status check before a PR is made or modifications to the `dev` branch.

Another opportunity is to segment our development. Also mentioned by our TA, we did not lend to having a *build branch* and a `dev` branch for any checks. This would have been immensely similar to working in industry. Often times work is segmented into:

1. Prod
2. QA
3. Dev

We will see this Warmup as an opportunity to refine our branching methodology, and while each branch was well documented with an issue. Currently, a lot of the integration was handled on a developer's branch, and then merged with main, status checks and developer designed tests should be handled on a **Dev/QA** branch before being merged with main(action to be taken by the code-owners).

An opportunity that we can also focus on is **integration**. Our code was quite disorganized in the beginning, with developers making components that were not modular, with their own independent styling and independent folders. This led to **redundant code** that was eventually consolidated in an **integration** issue(a label we defined). In future, we will lend to having common code styling, a common repo structure and aim to produce modular, reusable assets/components that can be used across different web pages and flows.

Threats:

One of our core values as a team is dedication, and as such we will work to combat any potential threats to efficiency and progress on the horizon. Based on our weaknesses witnessed so far, there are certain potential issues that need to be addressed.

With the quarter progressing as it often does, we recognize that our schedules will get a lot busier in the coming weeks. **Increased academic workload** could lead to more scheduling conflicts, making it harder to develop products to their maximum potential within a tight time frame. As such, we aim to get ahead of the scheduling issue by *hashing tasks out as soon as deliverables are requested*.

Additionally, efficiency can be significantly hindered by **lack of parallel work**, as developers are put in deadlock waiting for updates from one another. To combat this threat, we aim to encourage *more work on parallel branches* on issues that do not affect one another. This will allow us to seamlessly develop within the most optimal timeline possible.

Finally, we are concerned about the team's **lack of UX familiarity** leading to a suboptimal product being designed. This is a serious threat to our work, as our ability to develop in a user-centered manner is hindered drastically by unfamiliarity with frontend technology. During time between assignments, our designers and developers aim to *become more acquainted* with tools such as Figma and Miro, as well as *communicate ideas better internally* for maximal efficiency.