GoodDeed: Milestone 1

March 30th, 2025
By the Deeders

Overview

Volunteers often struggle to find relevant, local, or time-flexible opportunities, while smaller organizations lack the resources to reach and coordinate with potential volunteers. So, we decided to create aggregating volunteer opportunities in one place!

Team:

Phong Trinh Ha - IC / Debugging, added Tailwind + shadcn/ui, created UI components

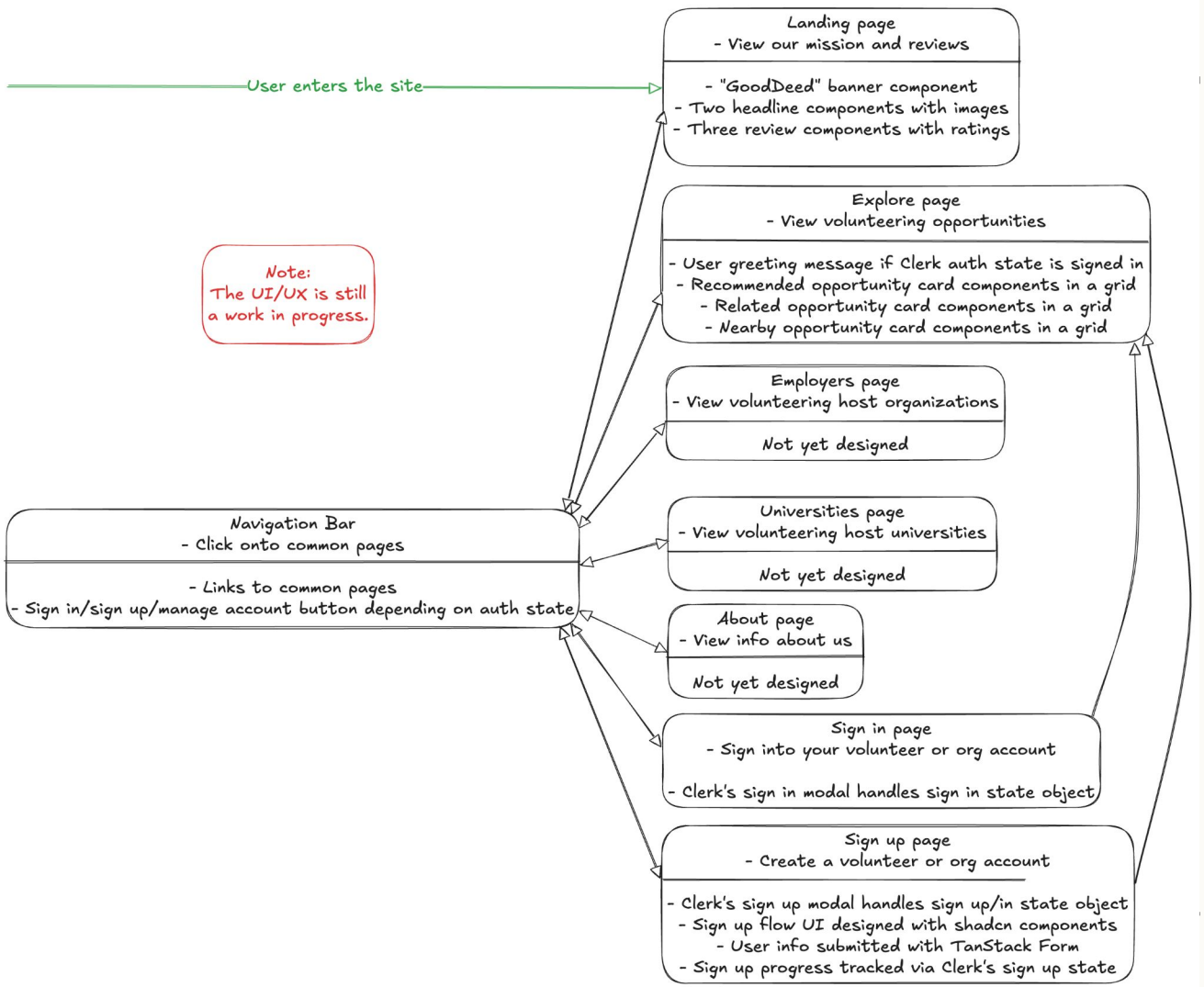
Gabriel Laboy - IC / Debugging, created explore page & UI components

Cameron Proulx - IC / Debugging, created error page

Jacob Beaumont - IC / Added Storybook, created landing page & UI components

IC = Individual Contributor

Architecture



GoodDeed Timeline

Design and Styling Guidelines

[Styling Guidelines Links]

Our design system reflects a dynamic and upbeat tone aimed at a younger audience, while preserving the warmth and welcoming feel that's central to our volunteer-focused mission.

We achieve this through a vibrant color palette, accessible contrasts, and a modern layout system that feels both friendly and intuitive.

Component Documentation

NavBar:

- Key sections (Explore, Employers, Universities, About)
- Clear **CTAs**

Opportunity Card:

- Heading with requested **role and organization**
- Concise description of primary task
- Clicking opens details

Performance Considerations

Package Management (pnpm):

- Utilize pnpm’s caching mechanism to speed up CI/CD pipelines.
- Consider using pnpm’s offline mirror capabilities to further enhance build reliability.
- Use hoist mechanism to bolster monorepo set up

Bundle Optimization:

- Use Vite to optimize bundle size for both monorepo and assets such as video and images
- Use tree-shaking and lazy loading to eliminate dead code and only utilize what is needed

Monorepo & Build System:

- Leverage turborepo’s caching to avoid rebuilding unchanged packages.
- Optimize pipeline configuration to reduce unnecessary work across projects.

Gabriel: Work Summary

✅ Create NavBar Skeleton (#38)

✅ Create Footer (#8)

✅ Create Opportunity Card (#3)

✅ Create Explore Page (Feat Dashboard #2)

Pull Requests:

- Navbar skeleton
- feat/Explore Page

Commits: 4

Gabriel: Code & UI Explanation

Key Code Contributions

- navbar.tsx/css
- opportunitycard.tsx/css
- footer.tsx/css
- explorepage.tsx/css
- Infocard.tsx/css
- exploresection.tsx/css

For Milestone 1, I worked on several key parts of the website, including the Navbar, **OpportunityCard, Footer, ExplorePage, InfoCard,** and ExploreSection, along with their CSS. These components helped shape the main layout and look of the site, making it easier for users to explore and understand the platform.

By building these sections, I helped create a clean and consistent design that gives users a clear first impression. This foundation will also make it easier for our team to keep adding new features and improving the site moving forward.

Gabriel: Challenges & Insights

One of the main challenges I faced was trying to use gluestack-ui for our UI components. We ran into compatibility issues since it doesn’t work well with React in our setup. After some troubleshooting, we decided to switch to TailwindCSS and shadcn/ui instead. This change made development smoother and gave us more flexibility with styling and component customization. It was a helpful learning experience in choosing the right tools for the tech stack we’re using.

Jacob: Work Summary

Assigned Issues:

✅ Create Landing Page (#17)

✅ Create Landing Page Card Component (#18)

✅ Create Banner w/ Tagline (#19)

✅ Create Review Cards (#20)

✅ Install Storybook & Initialize Storybook Components (#46)

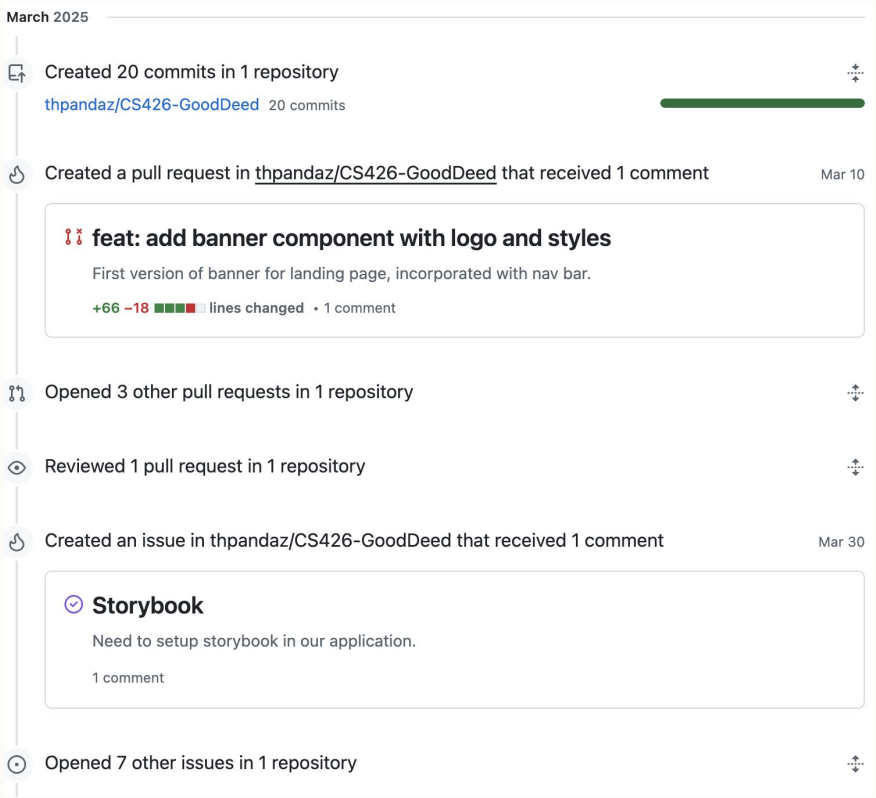
Pull Requests Opened:

- feat/landing page

Pull Requests Closed:

- feat/storybook

Commits: 20



Jacob: Code & UI Explanation

Key Code Contributions

- banner.tsx
- landingcard.tsx
- reviewcarousel.tsx
- landingPage.tsx
- Storybook

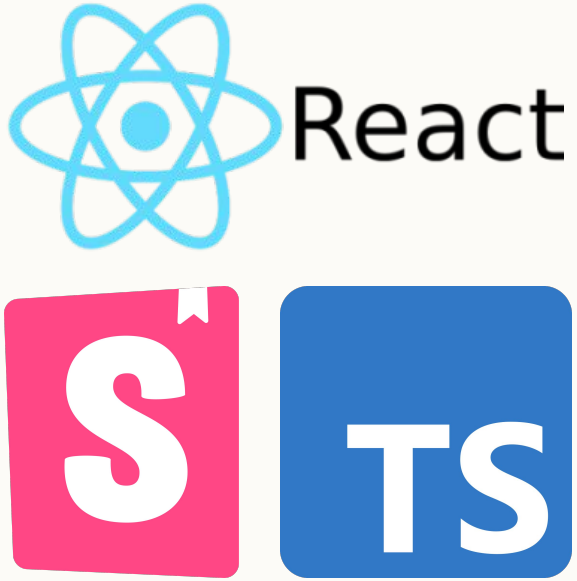
The creation of these components and corresponding Landing Page resulted in an engaging first look at our web page, showcasing a clean, user-friendly interface. This initial impression is key in keeping users engaged and encouraging them to explore further volunteer opportunities on the platform.

Integration with Storybook will allow us to develop and test UI components in isolation, ensuring consistency and reusability across the app. This will streamline our development process and enhance collaboration between design and development teams, leading to a more polished user experience.

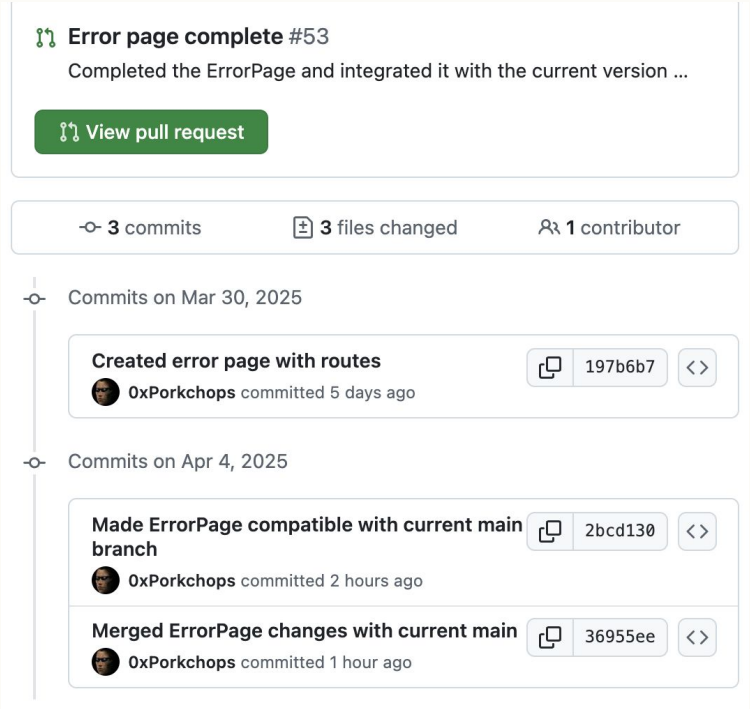
Jacob: Challenges & Insights

This milestone gave me a great first look into not only React, but web development as a whole. I learned how to break down a project into reusable components and gain an understanding of how static UIs are structured and styled. This was a great starting point for understanding the foundations of React and how it fits into the larger picture of building modern web applications.

Some challenges I faced were mostly due to the steep learning curve for this class, especially with understanding and implementing components alongside CSS. The React framework can also be challenging at times, but most questions can be answered with the provided documentation.

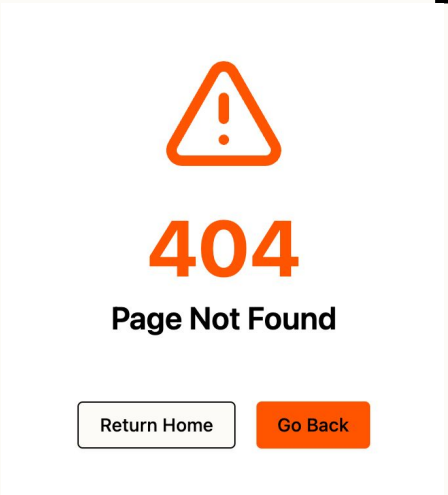


Cameron: Work Summary



✅ I completed issue #22 which was to create a default error page. Here are the commits. Here is the pull request. I also spent a lot of time debugging dependencies.

Cameron: Code & UI Explanation

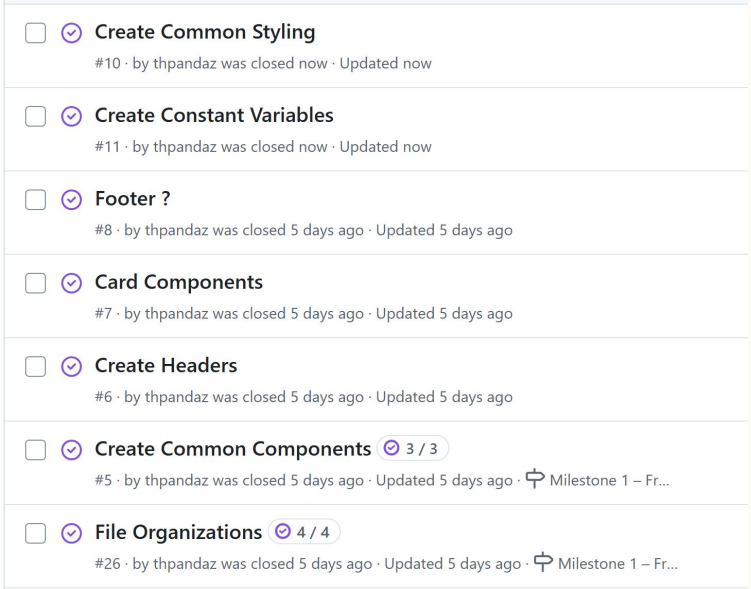


I created errorPage.tsx which is the default page to serve when a route isn't available. It takes the status of the request from the request params, such as 404 or 500, and shows an appropriate message to the user. The user can then return to the previous page or to the home page.

Cameron: Challenges & Insights

I had issues installing TailwindCSS and shadcn/ui into our project structure. To get around this, I continued working in a basic React project with TailwindCSS and shadcn/ui installed and then transplanted the page I was working on into our project. When Phong resolved the installation (it turned out to be a problem with bun), I made adjustments and merged my changes. Working on this issue opened my eyes to the nuances of adapting tools to different project structures.

Phong: Work Summary



✅ My primary role has involved extensive debugging and code maintenance. By the end of this milestone, I believe I've delivered a stable and sustainable version that sets a solid foundation for future development.

Total commits: 14 commits this sprints

Phong: Code & UI Explanation

Key contributions:

- Added shadcn/ui
- Integrate tailwind with the monorepo system
- Create a standardize styling methods
- Create a standardize error handling system
- Created User Context and Constant

Phong: Challenges & Insights

✖️ The biggest challenge I faced was understanding package management. Many newer libraries and package managers (like Bun and Deno) rely heavily on caching methods, which made our monorepo setup with Turborepo incredibly difficult to configure and maintain. Although I had to downgrade several components, the experience taught me a lot—not only about how impressive these modern tools are, but also where there’s still room for improvement.

Future Improvements & Next Steps

Now that a lot of the synchronous aspects of setting up the project are done, we can make faster progress asynchronously. Our next steps are to continue working on completing our GitHub Issues to build out more core functionality of our app. This includes refactoring pages to use shadcn/ui components, adding user authentication, designing more pages, creating a backend, creating a database, etc.