**Sam Peacock and Jacob Greenberg**

[**GitHub**](https://github.com/sam-peacock1/PokeAPI-PokeSearch)

[**Git Projects**](https://github.com/users/sam-peacock1/projects/2/views/8)

**Overview and Justification:**

After looking for a project for a few hours, our group decided to just make our own project from scratch. Our project is called PokeAPI-PokeSearch, and the link to our github page is [PokeSearch](https://github.com/sam-peacock1/PokeAPI-Pokemon-Finder).Our project is a website where the user enters the name of a Pokemon they want info on. If there is no Pokemon of that name, they will be asked to enter a new name, but if the Pokemon does exist, the Pokemons name, type, height, weight, and abilities will be shown to the user. We were trying to find a project that we could spend a good amount of time on, was something new to us, and seemed fun, but never found anything that fit those criteria. So we threw around a few ideas for our own projects. One of them being a website that would return if an item you were looking for was in stock, at what stores,and what aisle it was in at Target. We found out Sam has access to Target APIs because he works at Target. But we ended up not doing this project after waiting a few days for access tokens and never received one. We also thought of a few ideas to do a project around the video game Rust. but inevitably found the PokeAPI and decided to do this.

**Project Context:**

Our project was created because of a love of Pokemon and a desire to allow people to easily find information about Pokemon. While our current version of the site is not extremely complex, we know in the future there are lots of new features and information that could be added. If we continued to fledge the website, we could even add ads to the site to make a little extra money. But the history of our project is pretty simple. After finding out about Targets APIs we decided to see if there were any other fun APIs we could utilize and eventually found PokeAPI. After reading up on the documentation we knew it would be perfect for our project and started brainstorming ideas of how to use it. Jacob is not super familiar with Pokemon so Sam thought up the idea to create a site where he could find out information about more Pokemon.

**Project Governance:**

The main communication our group used was either in person or Discord. We used Github to collaborate together, both of us creating branches and committing to our branches before merging together to main in the end. We mostly worked on this project together so communication was extremely streamlined and simple, we were always on the same page.

We created separate branches for each issue/ticket. These branches were then merged back into main. We tried to avoid committing directly to main, though towards the end of the project we pushed smaller bits of code and tests directly to main.

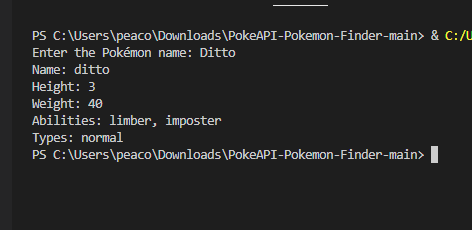
**Task Description:**

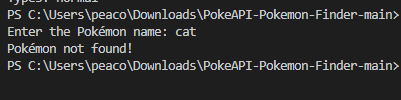
The first task of this project was to connect and get data from the API. After this was done we then needed to decide what data we wanted to pull. Once a decision was made we then created the backend and tested it, making sure all our use cases worked. From there we moved on to the front end and decided to make it in Flask. We then made a rough draft of what we thought our front end could look like. The Next task was to have our front end display the information gathered from the API. Once this was working, and tested, it was all stylization from there.

1. Goals and Objectives
   1. Create working backend
      1. Get connection working
      2. Decide what info to pull
      3. Create final backend
      4. Test
   2. Create working frontend
      1. Rough draft front end look
      2. Get connection between front and backend working
      3. Test
      4. Stylize
2. Success Criteria
   1. If we have a working website with the information we decided to display we will view the project as a success.
3. Deliverables
   1. A Working website
   2. Any and all documentation
4. Scope
   1. Keep scope simple for now, make sure beginning implementation works, can extend in future
5. Resource plan
   1. 24hrs of work per group member
6. Risk Analysis
   1. See Risk analysis sheet
7. Timeline
   1. See Timeline document
8. Plan Milestones
   1. See Github Milestones
9. Team roles
   1. Backend: Sam
   2. Frontend: Jacob
   3. QE Testing: Sam and Jacob
   4. Documentation: Sam and Jacob

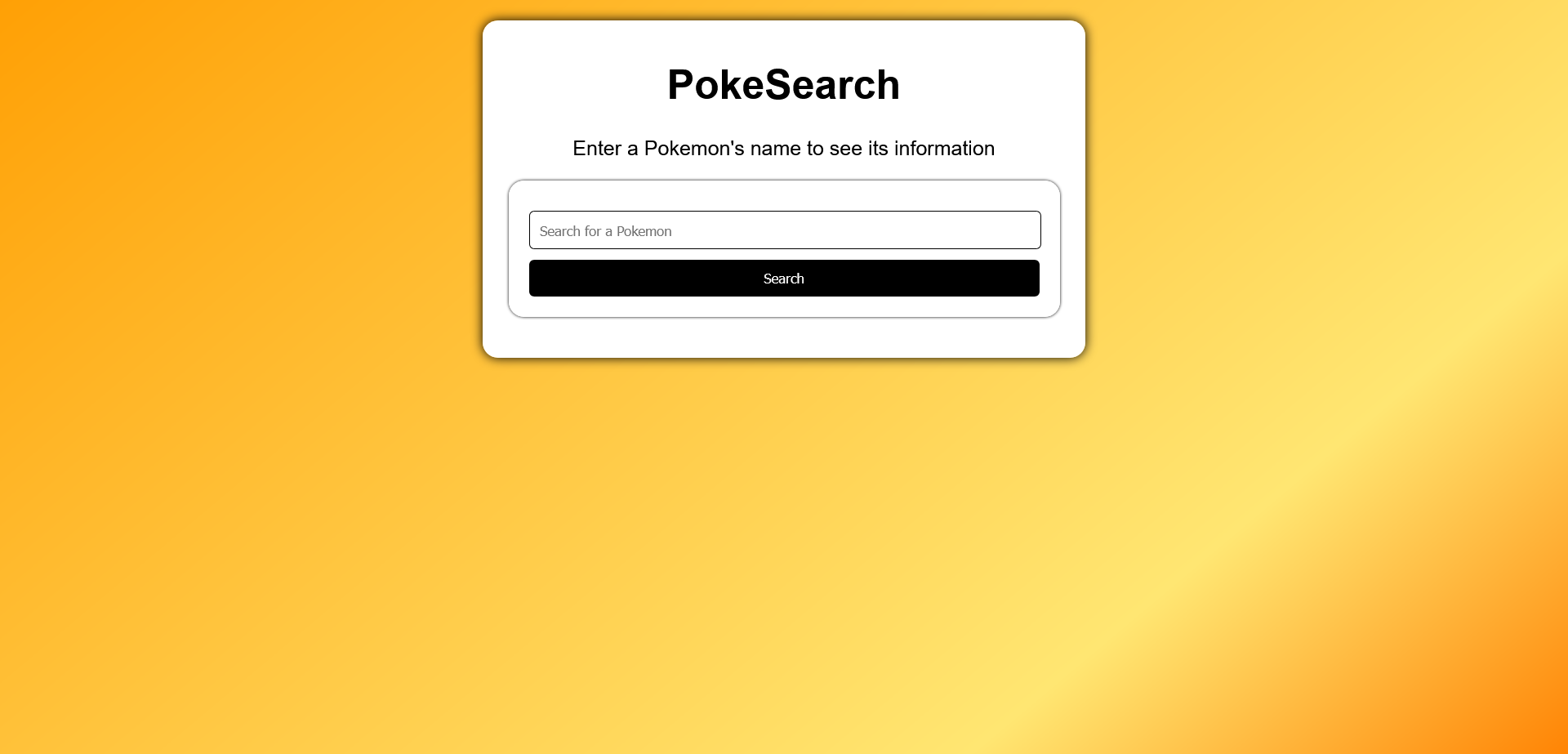
**Build Evidence:**

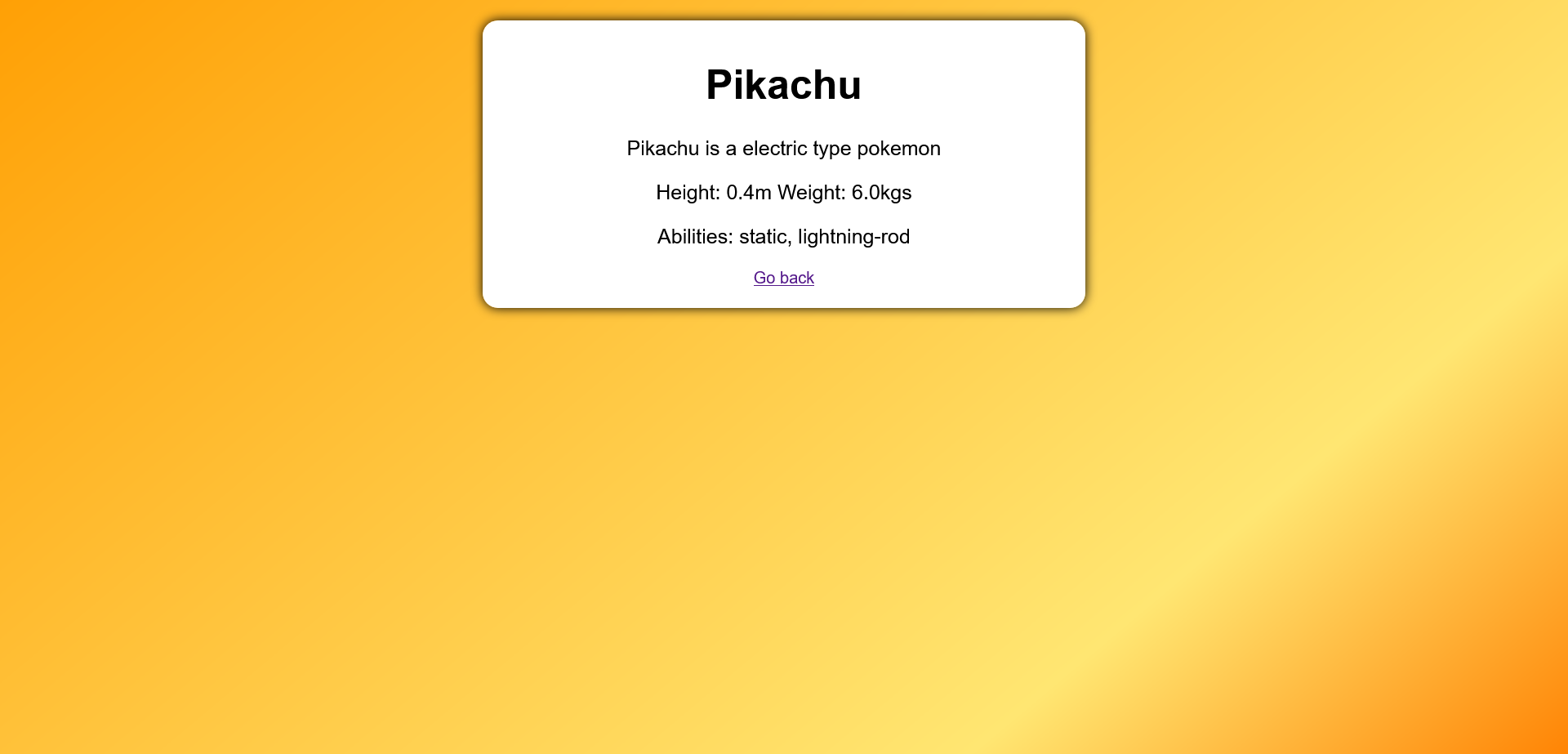
Back End:

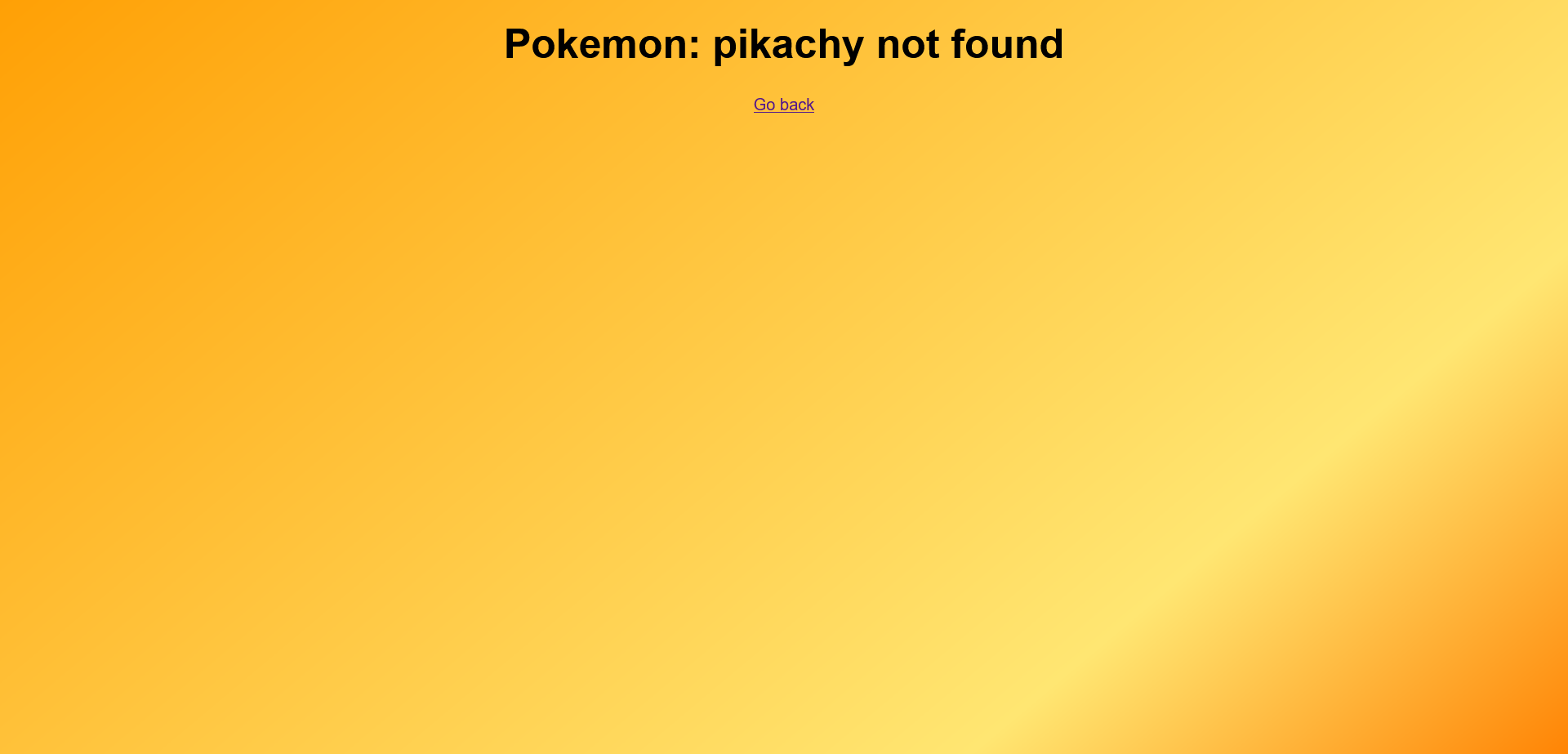




Front End:







**Submitted Artifacts:**

All Artifacts can be found on the Github Page: [PokeSearch](https://github.com/sam-peacock1/PokeAPI-PokeSearch)

**QA Strategy and Evidence:**

PokeSearch has a number of testing and QA strategies.

* Automated Testing

PokeSearch’s gitlab repository runs an automated test to ensure backend API calls are functioning properly

* Manual Testing

PokeSearch contains two tests which are conducted on an already running web server. The tests verify pokemon can be searched over the web and that invalid pokemon show the correct ‘not found’ page.

* Third Party Verification

We conducted some short third party tests. We ask friends to try using the application and provide feedback. This gave us insight into what the average user might be thinking when they use the search engine, compared to someone involved in the project like us.

**Plan Updates:**

Our plan really didn't change at all, we spent the time during our discovery phase to make sure everything we wanted to do was possible. We also built this into our initial schedule when deciding what info to pull. Our scope, work allocation, and schedule stayed consistent.

**Experience and Recommendations:**

We had an interesting experience with trying to interact with the open source community as we had a difficult time finding a project or issues for us to work on. We spent hours looking for something that would fit our criteria and weren't able to find anything, which is why we decided to create our own open source project. We learned a lot doing this project, from correctly using source control to the troubles of creating an aesthetically pleasing website. But the most difficult part was definitely choosing a project, then making the whole plan to create the project, but I think we benefited from setting time aside to map it all out. For future projects, if we could get access tokens to the Target API discussed earlier, that could be very cool. A suggestion for finding good issues to work on for first time open source users, is make sure to only look at issues that are tagged with beginner friendly, that's how we found options that were the most plausible, even if we didn't decide to go with them.All and all, we learned a lot doing this project and are glad we got the experience to work on this.