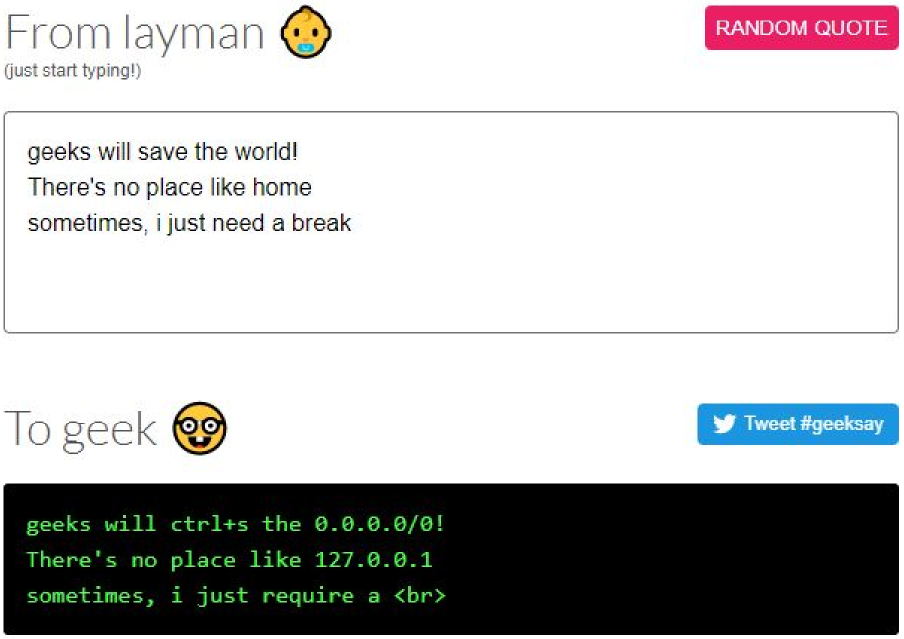
Geeksay Final Portfolio

Initial Project State

Geeksay is a relatively small project that sets out to create a fun keyboard that converts certain words or phrases into the “Geek” language. This is a fun project for people to mess around with but it also has the potential to scale up and be more widely recognized. Their original Github page and website have a CONTRIBUTING page and a README document, however, they do not outline some information that is critical for users and potential contributors to have. The website consists of two main parts: the input text field, where users would insert a word or quotes into; and there is “To Geek”, which outputs the correspondents in the “geek” dictionary. The total number of lines of code for the entire project is close to 4000. It currently has around 130 basic words and 50 quotes in their geek language dictionary.

User Design

Geeksay is a web-based translator that allows users to translate words or quotes in English into the corresponding Geek language. The overall design and functionality are similar to the existing web-based translator like Google Translate. Geeksay is still in its development stage, meaning that there are only a limited amount of words or quotes are available. Geeksay is user friendly, no additional hardware is needed other than a device with web accessibility.



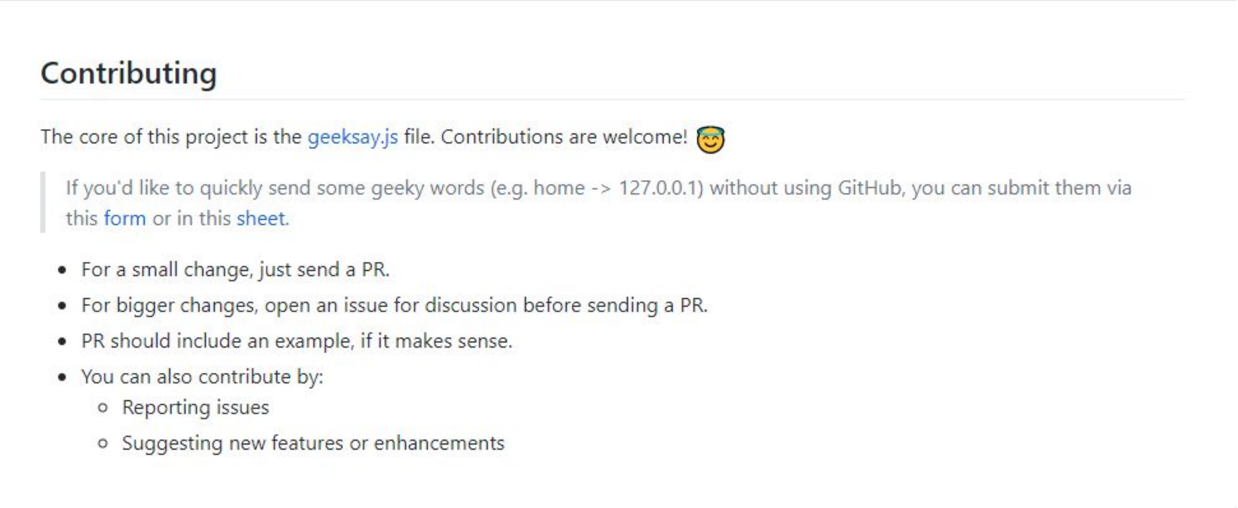
geeks will save the world!
There's no place like home
sometimes, i just need a break

```
geeks will ctrl+s the 0.0.0.0/0!
There's no place like 127.0.0.1
sometimes, i just require a <br>
```

When you get access to the website, Geeksays offers two options: 1. Input words or quotes manually into the textbox below “From layman”, 2. Automatically generates a quote by clicking the “Random Quote” button. If the input matches data in Geeksay’s dictionary, the correspondents will output to the section under “To geek” in the Geek language format. From the screenshot above, “home” translates to “127.0.0.1”

Functionalities

Below, I have shown the original contributing page as well as the original README page. The contributing page is simple as it does not put out any goals for the project or projections into the future versions being developed. Additionally, currently the only way to communicate with the project is through the personal email of the original creator.

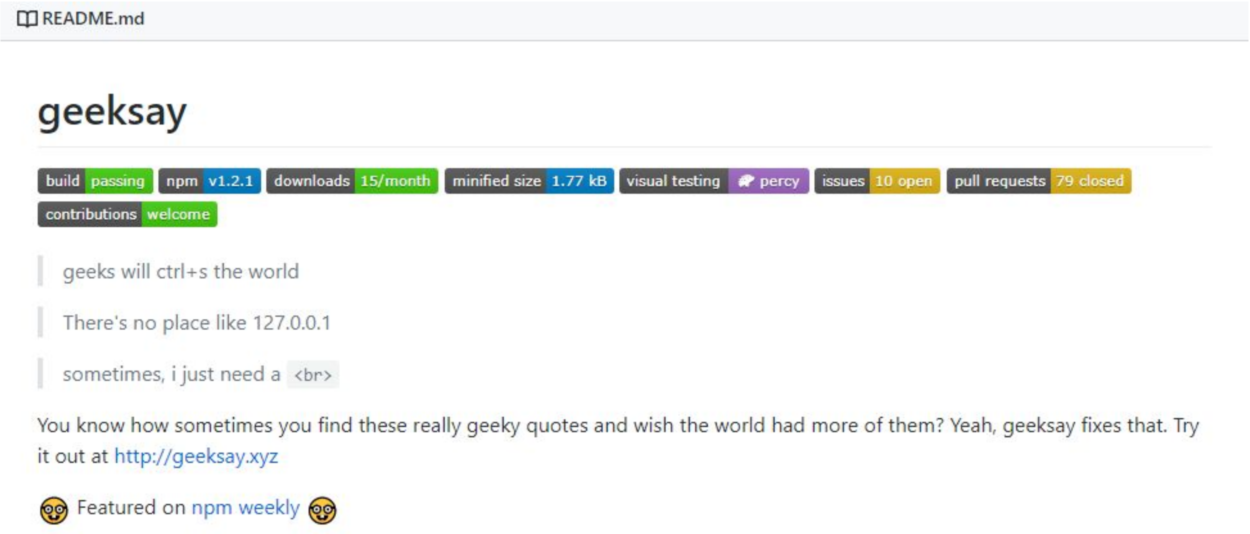


Contributing

The core of this project is the geeksay.js file. Contributions are welcome! 😇

> If you'd like to quickly send some geeky words (e.g. home -> 127.0.0.1) without using GitHub, you can submit them via this form or in this sheet.

- For a small change, just send a PR.
- For bigger changes, open an issue for discussion before sending a PR.
- PR should include an example, if it makes sense.
- You can also contribute by:
  - Reporting issues
  - Suggesting new features or enhancements



README.md

geeksay

build passing | npm v1.2.1 | downloads 15/month | minified size 1.77 kB | visual testing percy | issues 10 open | pull requests 79 closed | contributions welcome

> geeks will ctrl+s the world

> There's no place like 127.0.0.1

> sometimes, i just need a `<br>`

You know how sometimes you find these really geeky quotes and wish the world had more of them? Yeah, geeksay fixes that. Try it out at http://geeksay.xyz

🤓 Featured on npm weekly 🤓

One of the critical aspects of a project like this is outlining the limitations placed on contribution due to licensing. The Licensing page outlines the type of license that the project employs however, a summary of what the license entails could be beneficial.

Other Features

Features	Icon	Result
The “Git” button redirects to the Geeksay’s Github repository.		Passed.
The “RANDOM QUOTE” button automatically generates random quotes in the input text box.		Passed.
The “Tweet #geeksay” button redirects a popup that allows users to share a link of Geeksay’s website to Twitter.		Passed.
The “Need an API?” button redirects to one of the open issues on Geeksay’s Github repository.		Can be improved.
The “ONE MORE!” button automatically generates a random translation (words only) that gives users a general idea of how translation works.		Passed.

Overall User Experience

README:

Overall the page outlines the procedure to download it and initialize the project. However, it does fall short in its attempt to introduce the project. For a person who first finds the project on Github, it could be confusing to understand the purpose and goals of the project without visiting the website in addition to the Github page.

Contributing:

The contributing page needs to have some organization so that contributors that have questions about what needs to be contributed or even just an outline of previous contributions. Currently, the work required to get to know the project deters developers from trying to find a way to contribute. Additionally, it does not outline any credit that would be allotted to contributors.

Initial additions and response:

Based on the user experience with the contribution page, I decided to reformat it and added two sections. The first being contributions that could be made by the average person visiting the project. These contributions can come in the form of additional translations for the project that could be suggested using a google form that I linked. This form would then add those entries into an excel sheet in which duplicates would be counted allowing the developers to see the popularity of certain terms and translations and take that into consideration when choosing whether to add them.

The second section is more directed toward those that have a computer science background. This outlines the openness to contributors and the different avenues that they have to communicate with the creator and fellow contributors. I considered adding reporting bugs to the more “layman's” section however, it took away from the overall organization and flow of the document. I also added some emojis to both sections which reflect those that are used in other places in the project.

I also chose to change the README file to be more concise and show more relevant information first. They created an amazing demo of how the project works in the form of a GIF. However, originally this was placed in the middle of the initialization of the project and installation. Instead, I chose to show this in at the start and leave all of setting up the project last. I also added more summary and general information to start the document off.

Overall, the user experience was good, the program does not require any pre-knowledge and is easy to use. The translator functions well, however, due to the limited size of its dictionary, I would normally get an output in English instead of the Geek language. Since expanding the dictionary might take a long period of time, I suggest adding a list of words or quotes to the webpage that allows users to learn and find out its availability. In addition, most of the buttons worked well and I did not find them unnecessary. My only concern was that the redirection of the “Need an API?” button felt a bit unprofessional. It links to an issue on its Github repository, I suggest creating a new file on Github or a new page on the website which should include the necessary steps to get an API as well as the methods to communicate.

Testing of new capabilities:

The new additions allowed for a person who just wanted to add some suggestions to do so in a way where they feel like it has meaning. Before the user had access directly to the excel sheet in which they could see other suggestions that did not get accepted. I observed that this could encourage new users to feel that their opinions at least were being taken into consideration instead of being put into an excel sheet to never be seen again.

In regards to the README file, the additions helped those that wanted to get insight into the direction and purpose of the project, to get a better idea. This differed from before in which users that found the page through Github would need to find their way to the website to understand the purpose or even uses of the project.

Manual Capability Testing

The table below shows different test cases including normal inputs and different types of abnormal inputs.

Test Case	Input	Expected Output
#1	NULL	NULL
#2	hello world	ping 0.0.0.0/0
#3	helloworld	helloworld
#4	HeLLo WORLD	ping 0.0.0.0/0
#5	-hello world-	-ping 0.0.0.0/0-
#6	0.0.0.0/0	0
#7	0	0
#8	10	1010
#9	go home	go 127.0.0.1
#10	!!!not	!!!!
#11	copy-cut	copy-cut

#12	copy cut	ctrl+c ctrl+x

Manual Capability Testing Outcomes

The table below shows the actual outputs from different test cases and the result of the outputs.

Test Case	Actual Output	Result
#1	NULL	Passed
#2	ping 0.0.0.0/0	Passed
#3	helloworld	Passed
#4	ping 0.0.0.0/0	Passed
#5	-ping 0.0.0.0/0-	Passed
#6	0	Passed
#7	0	Passed
#8	1010	Passed
#9	go 127.0.0.1	Passed
#10	!!!!	Passed
#11	copy-cut	Passed
#12	ctrl+c ctrl+x	Passed

All tests passed and the results came out as expected, the program works well with normal inputs and different types of inputs (case sensitivity, ‘space’ detection, decimal to binary conversion, interaction with symbols).

Final additions and suggestions

Lastly, I think that there still needs to have quite a bit of work done on the licensing page in order for contributors to get a good idea of the limitations they may face. Additionally, different avenues of communication with the developer and fellow contributors. These additional avenues would help contributors have a network to work with one another and understand what is currently being worked on. There could also be more changes done to the website however, to speculate what these changes would be is difficult because many of the options such as communication avenues and development goals depend on the developer.