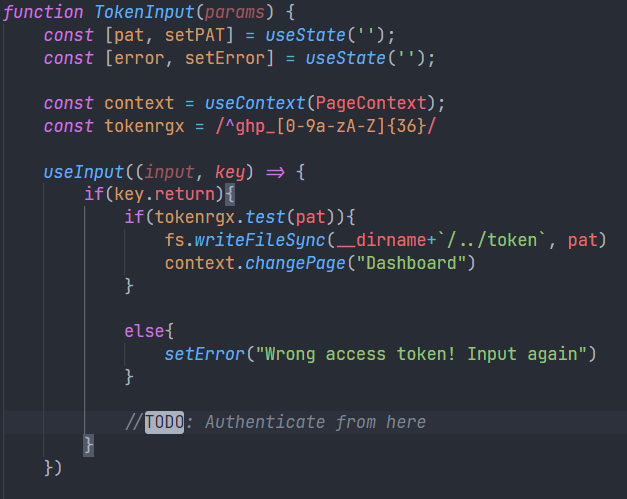
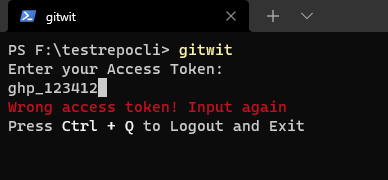
**Functional Testing**

**Personal Access Token**

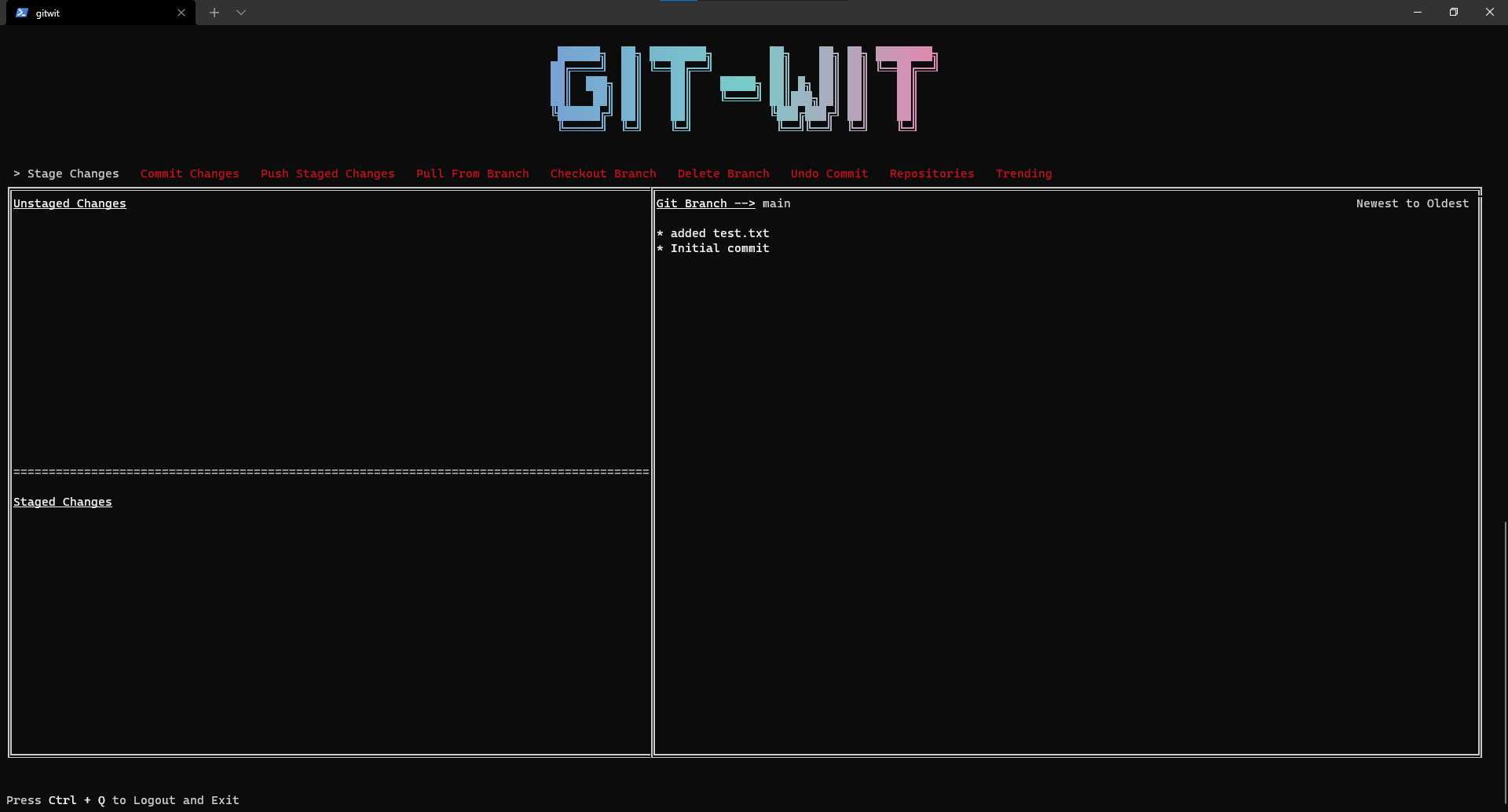


A regex is provided to test the validity of the input token, and only when the regex is tested true, we are saving the token and authenticating later when the components are rendered.

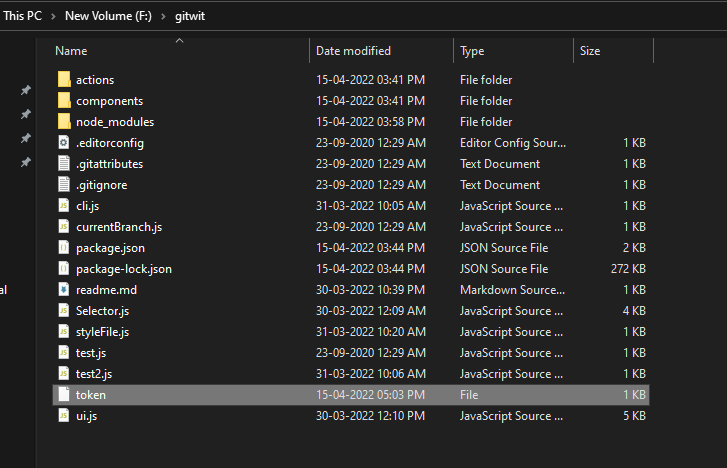
Since there is no way to actually check for the validity of the token without requesting from GitHub, we are checking it earlier here.



If the token is correct, the dashboard renders on the terminal giving the user full access to the program.



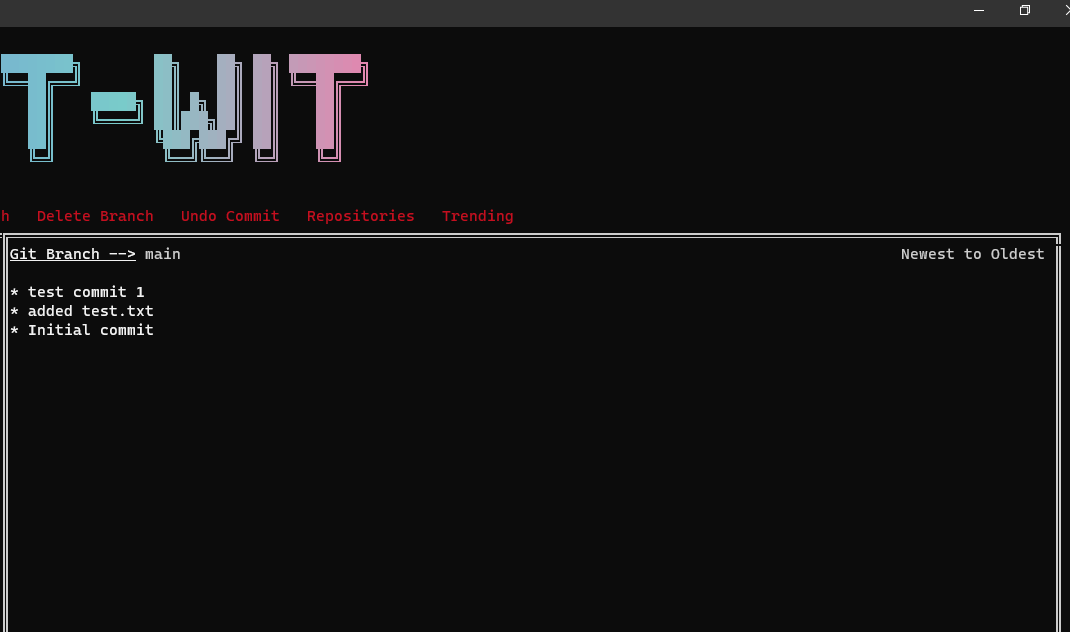
If the token is verified by the regex test, it is then saved as a file called “token” which caches the token till the program is logged-out explicitly using “Ctrl + Q”. If the user doesn’t logout, the token is kept and the user can login anytime without using the token again.



**git commit**



This component shows the staged/unstaged changes in the repository. So that the staged files can be committed and we can see the files in the component above.



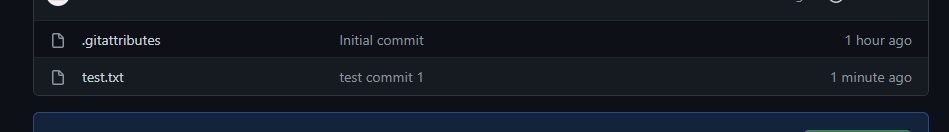
Then we can see the committed message in the list of commits.

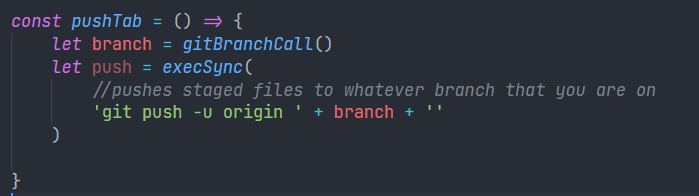


Commit is done through this function. execSync is used to execute any terminal command through JavaScript. The commit message is stored in a React State object and then while executing the command, it is passed through in the parameter.

**git push**

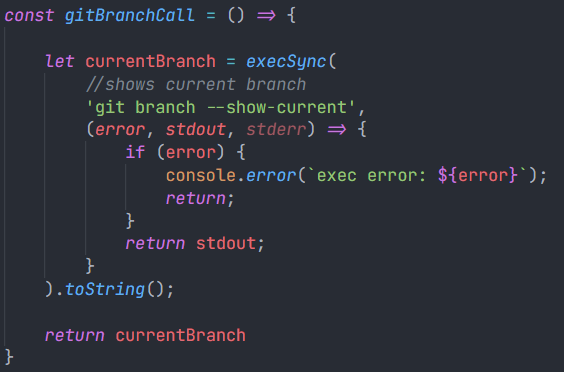
After pushing the changes to the repository the commit made can be seen in the GitHub repository.





Push is done through this function. The only other thing that this function takes as a non-changeable(by the user) parameter is the branch. The current branch is taken from the function gitBranchCall( ).

**gitBranchCall**

****

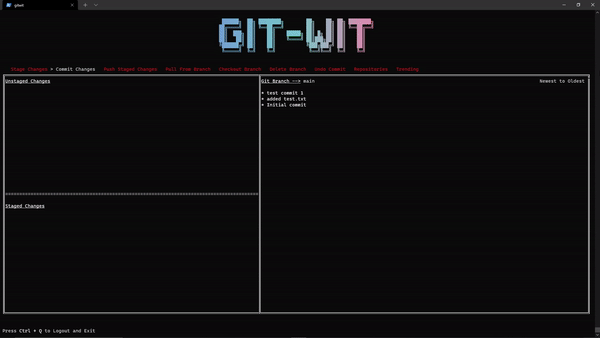
This function returns the current branch of the git directory we are working on. This is helpful for the push function as well as to show the user which branch the person is working on currently.

(Current branch displayed in UI)



**GitHub functionalities**

**View own repositories**

****

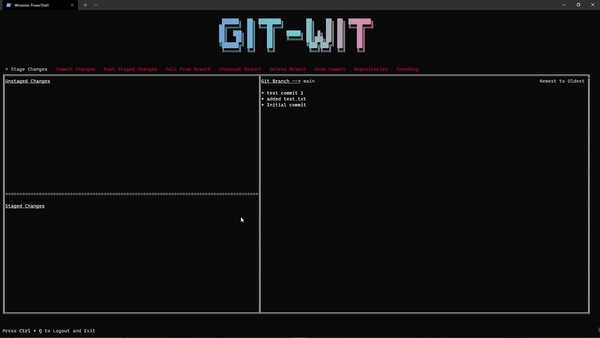
****

****

This API call returns the user repositories that are created by the user. This includes public as well as private repositories. The private repositories are fetched through providing the personal access token to the API as the config parameter.

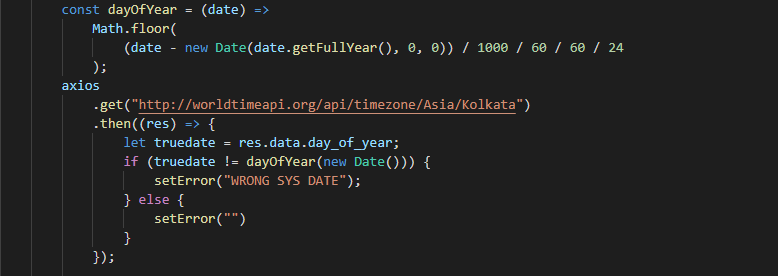
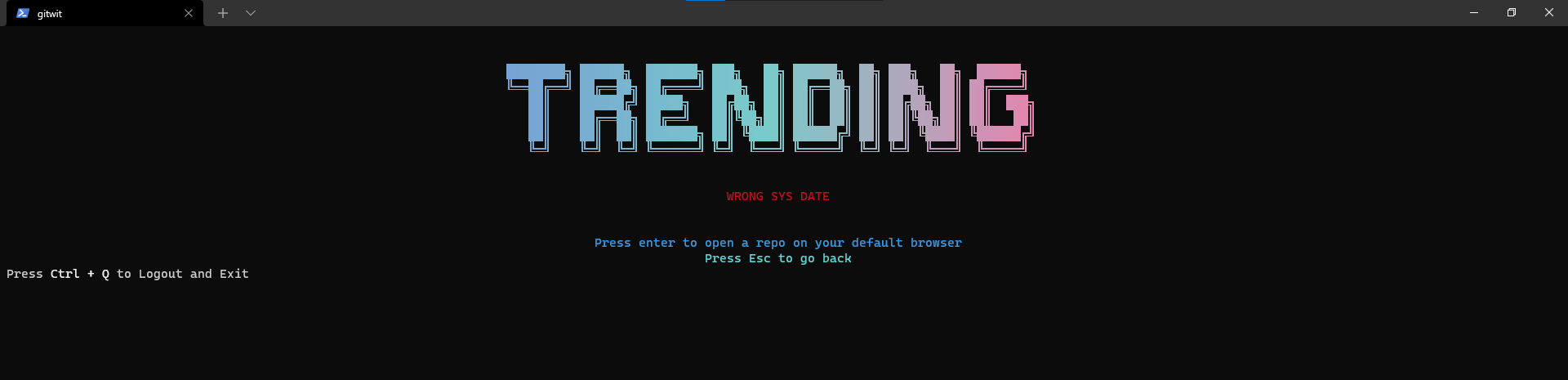
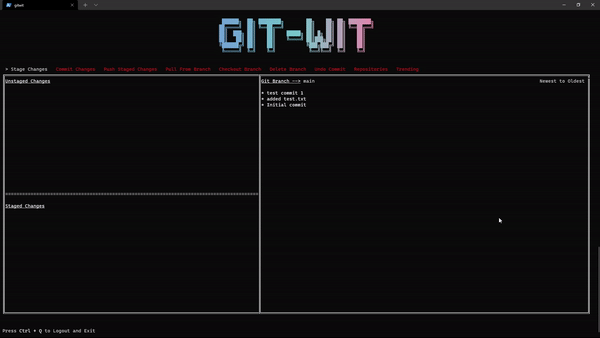
**View trending repositories**





This function returns the current trending repositories provided by GitHub as we can see on the explore tab of GitHub. The present moment is defined by providing the system date as provided by the Date( ) API of JavaScript.

**Trending page error handling**



This function checks if the system date is matching with the actual date provided by the worldtimeapi. We are checking this to ensure that the system time is correct and we are not providing a wrong date. In this case a wrong date can be a future date or a date not recognized by the GitHub API.

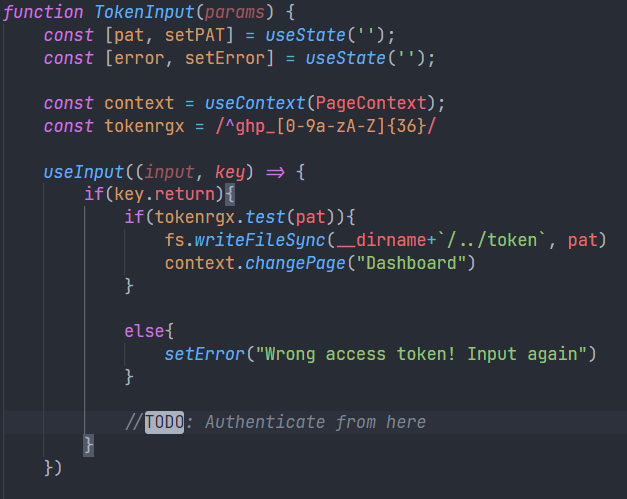
——————————————————————————

**Black Box Testing**

For Black Box testing we performed the following tests.

**Syntax Driven Testing**

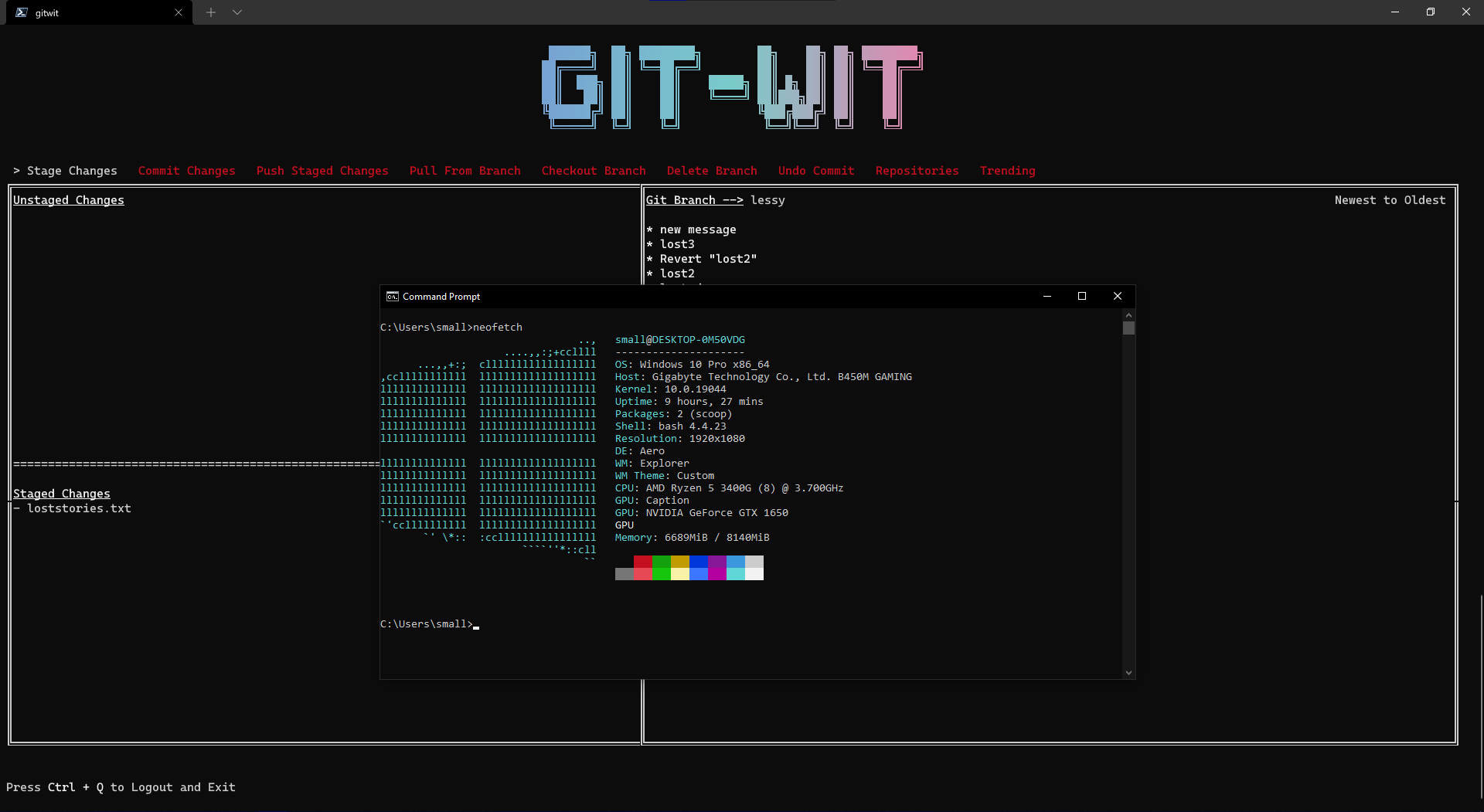
For Syntax Driven Testing, we matched the input token using regex.

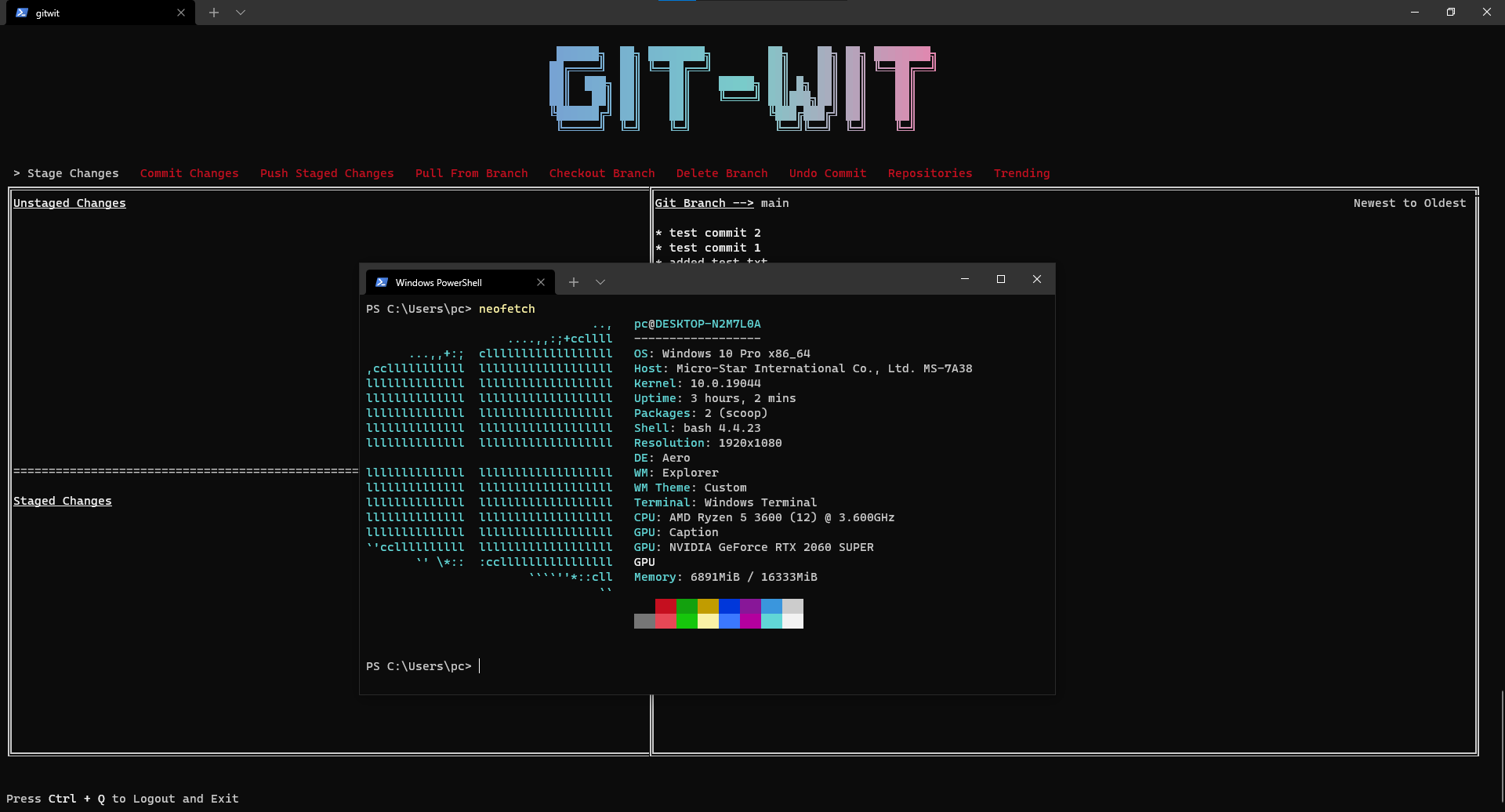


**Cause Effect Graphing**

|  | Case1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 | Case 7 | Case 8 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Token | ⨯ | ⨯ | ⨯ | ✔ | ✔ | ⨯ | ✔ | ✔ |
| Folder | ⨯ | ⨯ | ✔ | ⨯ | ✔ | ✔ | ⨯ | ✔ |
| Date | ⨯ | ✔ | ⨯ | ⨯ | ⨯ | ✔ | ✔ | ✔ |
| Error | ✔ | ✔ | ✔ | ✔ | ‒ | ✔ | ✔ | ‒ |
| Dashboard | ‒ | ‒ | ‒ | ‒ | ✔ | ‒ | ‒ | ✔ |
| Trending | ‒ | ‒ | ‒ | ‒ | ‒ | ‒ | ‒ | ✔ |

**Compatibility testing**

****





We tested our app on three systems and all the systems ran our program flawlessly. These systems varied in processor, RAM and GPU. Thus we can say that almost any system can handle our program efficiently.

**Requirement based testing**

The functionalities we mentioned in the SRS documentation are validated as shown above.