Fix Dead links in GitHub Repository

Team09 Zhengdong He Tianming Zhang
Supervisor: Christoph Treude

Introduction

Our web-based application is currently designed for GitHub user to fix the dead-link in their GitHub repository. the tool will monitors the links that inserted to the source code, and creates a backup copy of these target websites, then generates new links with the backup copy for user to access. Therefore, once the original link is not available anymore, this tool will make it easier for GitHub user to keep their source code documentation up to date and to avoid dead links in their source code comments.

Used Software & Tools

Heroku
A cloud platform basted on a managed container system for developing and running the application

Deployed on this online platform for user to use

GitHub API
A programming interface between GitHub and our program. The information from GitHub can be accessed through the AIP.

MySQL
An open-source database management system .

- Cross-platform support.
- Stored procedures,
- It is connected with the Heroku ClearDB

- jQuery
- JavaScript
- PHP
- CSS

Milestone

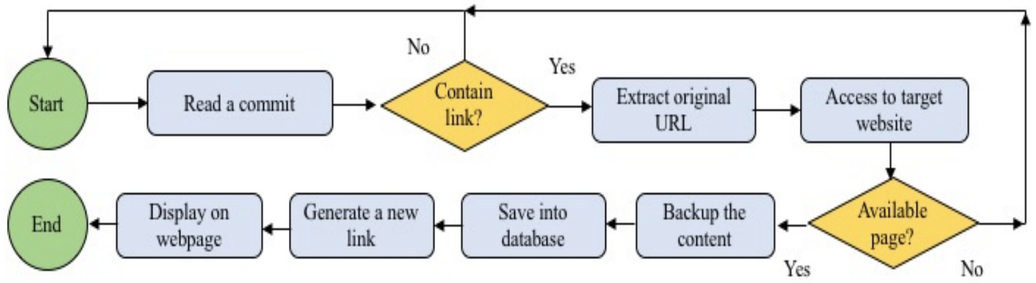
Milestone 1

- Set up the database on local server
- Use GitHub API to extract the information
- Insert the value into the local database
- Build the front-end webpage
- Display the value on font-end webpage.

Milestone 2

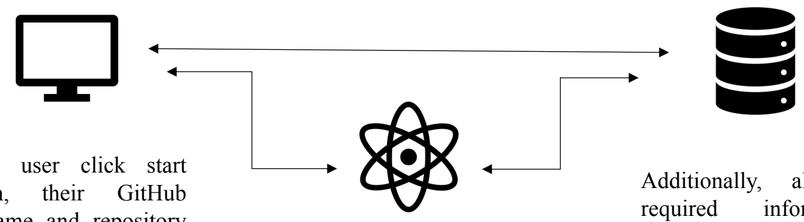
- Deploy the application on Heroku platform
- Backup the source code of target website
- Generate a new link with the backup copy
- Apply framework to re-design the front-end webpage

Key Concept



- Our program start by reading the commit from GitHub repository via GitHub API
- If the commit contains any inserted links, the program will extract the original URL by regular expression function and access to target website, if there are no links, the program will check the next commit circularly
- If the target website is available, the program will download the source code of webpage and save it into our Heroku database
- A new link with the backup will be generated, and all required information such as SHA(Secure Hash Algorithm), original URL and new link will be displayed on our webpage
- Lastly, user can access to the target website via click the new link from the webpage

System Architecture



When user click start button, their GitHub username and repository name will be passed from HTML textboxes to JavaScript variables. Theses information will sent to GitHub API to get corresponding JSON (JavaScript Object Notation) files to extract the SHA and original link

All the information will be extracted through JS and the original links will be sent to PHP page, which is to download the source code of the target website. The download source code will be saved into database and transfer to another new page with a new link that generated by PHP file

Additionally, all the required information includes the new link with the backup file will be saved into the Heroku database. Then, these information can be read by JS and then shown on the HTML pages (front-end webpage) for user to access

Achievement

Planed outcome	Actual outcome
Automatically monitor the GitHub commits through regularly checking the changes	Manually monitor the GitHub commits by clicking the button
Access all the GitHub commits and abstract links	Access all the GitHub commits and abstract links but some websites are not available
Download the contents from links and save them into the database	Same as planed
Replace the original links with the new links on GitHub	Show original links and new links through a table.
The whole program runs on a public server	Same as planed

Extension

- The regular expression will be improved and solutions of the redirected pages will also be found in the future.
- Plan to apply our program for other software users such as Bitbucket or Google Code.
- Heroku Background Job is to transfer the time of our application from the web layer to background process outside the user request lifecycle, but it was failed due to one of a required add-on is missed. However, we plan to fix the problem and apply this background job in future.

Conclusion

The program have achieved the most part of basic requirements, It is operated in Heroku platform and enable to generate a new link with the backup copy of the target website. GitHub user then can access the target website by click the new links and avoid the dead link in their repository. However, in the future, we plan to fix the problem of Heroku background job and improve our regular expression function