Project 2

Group 99 – agupta85, vbpatel3, btarun

GitHub Repository URL: https://github.com/CSC510SEFall24/URL-Shortener-2.0

Assessment Rubric

CSC510 Group 99 Rubrics		
Notes	**Self Assessment**	**Evidence**
Total of Self Assesment	153	
Workload is spread over the whole team (one team member is often Xtimes more productive than the others…	3	Distribution of features on github
but nevertheless, here is a track record that everyone is contributing a lot)	3	Github Commits
Number of commits	3	Github Commits:
Number of commits: by different people	2	Bhavishya Tarun: 15 Archit Gupta: 20 Vyom Patel: 10
Issues reports: there are many	3	Github Project Board
Issues are being closed	3	Github Project Board
Docs: doco generated, format not ugly	2	Github docs
Docs: what: point descriptions of each class/function (in isolation)	2	Github docs
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	2	Github docs
Docs: why: docs tell a story, motivate the whole thing, deliver a punchline that makes you want to rush out and use the thing	2	Github docs
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	https://github.com/CSC510SEFall24/URL-Shortener-2.0/blob/reimplementation/README.md
Use of version control tools	3	Github Repo
Test cases exist	3	Test Cases Folder
Test cases are routinely executed	3	Github Actions
Issues are discussed before they are closed	3	We used PR Reviews for discussions along with text messages
Chat channel: exists	3	Discussion on github
Test cases: a large proportion of the issues related to handling failing cases.	3	specific issues for testing
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	We all worked on our branches but had a single base "reimplementation" branch as shown

Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	Setups done on same config by everyone. One such example is shown
Evidence that the whole team is using the same tools (e.g. tutor can ask anyone to share screen, they demonstrate the system running on their computer)	3	All screenshots and video were taken by the whole group. Refer github and video for both
Evidence that the members of the team are working across multiple places in the code base	3	Branch history and commits show this
Short release cycles	3	Github Workflows
The file .gitignore lists what files should not be saved to the repo. See [examples]i(https://github.com/github/gitignore)	3	.gitignore file
The file INSTALL.md lists how to install the code	3	INSTALL.md file
The file LICENSE.md lists rules of usage for this repo	3	LICENSE.md
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	CODE_OF_CONDUCT.md
The file CONTRIBUTING.md lists coding standards and lots of tips on how to extend the system without screwing things up; e.g. see example	3	CONTRIBUTING.md
The file README.md contains all the following	3	README.md
Video	3	https://github.com/CSC510SEFall24/URL-Shortener-2.0/blob/reimplementation/video/demo.mov
DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo. DOI badges look like this:	1	Badges can be found in README.md
Badges showing your style checkers	3	Badges can be found in README.md
Badges showing your code formatters.	3	Badges can be found in README.md
Badges showing your syntax checkers.	3	Badges can be found in README.md
Badges showing your code coverage tools	3	Badges can be found in README.md
Badges showing any other Other automated analysis tools	3	Badges can be found in README.md
Does your website and documentation provide a clear, high-level overview of your software?	3	Refer github, and yes
Does your website and documentation clearly describe the type of user who should use your software?	3	yes, the website is concise and clear
Do you publish case studies to show how your software has been used by yourself and others?	1	
Is the name of your project/software unique?	2	URL Shortner
Is your project/software name free from trademark violations?	3	yes, its generic
Is your software available as a package that can be deployed without building it?	1	its not available as a package
Is your software available for free?	3	its on github
Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	3	GitHub
Is your software hosted in an established, third-party repository likeGitHub (https://github.com),	3	GitHub

BitBucket (https://bitbucket.org),LaunchPad (https://launchpad.net) orSourceForge (https://sourceforge.net)?		
Is your documentation clearly available on your website or within your software?	3	GitHub ReadMe
Does your documentation include a "quick start" guide, that provides a short overview of how to use your software with some basic examples of use?	3	GitHub ReadMe
If you provide more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?	3	GitHub ReadMe
Do you provide a comprehensive guide to all your software's commands, functions and options?	2	most of the functions do
Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	1	
If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	1	
Do you store your documentation under revision control with your source code?	3	GitHub ReadMe
Do you publish your release history e.g. release data, version numbers, key features of each release etc. on your web site or in your documentation?	2	
Does your software describe how a user can get help with using your software?	3	The frontend and github, both do the job
Does your website and documentation describe what support, if any, you provide to users and developers?	2	we have given our credentials
Does your project have an e-mail address or forum that is solely for supporting users?	2	we have our mails which we actively respond to
Are e-mails to your support e-mail address received by more than one person?	2	
Does your project have a ticketing system to manage bug reports and feature requests?	3	Projects Board
Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?	3	Projects Board

URL BIT
url shortener

Group 99

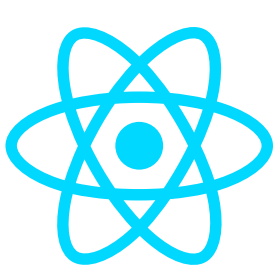
Archit, Bhavishya, Vyom

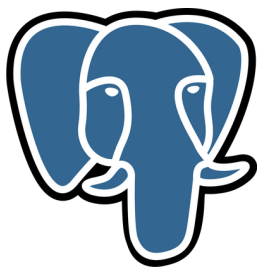
Introduction

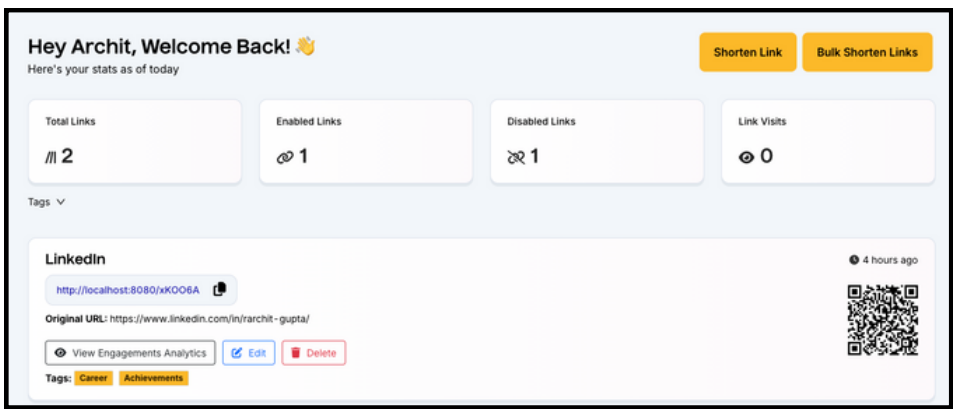
URL Bit is a URL shortener service that allows user to shorten URL's and organize them. It also allows them to view statistics related to URL engagement and enables users to filter them by tags

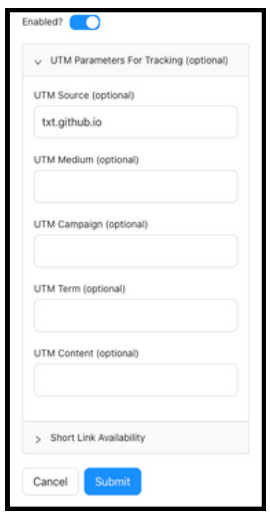
Tech Stack and Architecture

The project uses **Flask** and **Python** on the backend with an exposed REST API. This is connected to a PostgreSQL relational database. This is consumed by a frontend client written in React









QR Code

Github Repository

Testing Stats

65+ Test Cases

82% Code Coverage

Notable Improvements

Extensive Refactoring:
The software was extensively rewritten to allow for a much more modular codebase. The previous version included API calls and other functionality within side effects which would cause frequent re-renders. This has been addressed and fixed

URL Tagging:
Each shortened URL can now have tags associated with it that allow a user to filter out URL's which could be based on a particular category, campaign or product

Bulk Link Generation:
With a simple process, the application now allows users to generate multiple shortened URL's. These URL's can be uploaded as a .txt file with the links and relevant titles and can be processed all at once

Pre-determine link expiry
The system now includes a way to make sure shortened links expire after a certain number of engagements. This allows users to prevent unintended use and organize URL's efficiently

Possible Future Enhancements

Added Link Analytics

Although the system currently allows users to view link engagement and provides a sense of how many users viewed a URL, further analytics like device type, country, referrer data can improve usefulness of this service amongst campaign managers
Implementation: The frontend client can be modified to accept query parameters, and use device permissions to collect information

A/B Testing Options

A feature could be implemented to provide multiple redirect URL's in case of an A/B testing campaign which randomly redirects users to collect appropriate data
Implementation: The backend could be modified to accept multiple long URL's instead of a single URL. Then redirection could be handled on a random or specific information basis for successful A/B testing

Custom Short URLs and Domains

The system could also be enhanced by offering users to integrate custom domains and specifying what is generated as part of the shortened URL
Implementation: Allow third party domain registration to redirect to their domain, and allow custom text in the frontend client to make the URLs more customizable

Enhanced Security and Authentication

Upon inspection of the code, we found that the software currently uses unauthenticated backend requests. This could allow unintended users to access sensitive resources
Implementation: Secure JWT based authentication for all API endpoints