

Implement a Flask Application(Rest API's) to Shorten URL's

The main objective of this project idea is to shorten URLs. The main task the application will accomplish is to shorten URLs and then redirect users to the original URL when the shortened URL is visited.

In the application, the users will input the original URL, and they will get the new, shortened URL as the result. To do this, you can use a combination of the random and string modules to generate the characters for the shortened URL.

Since users will visit the shortened URL days, months, or even years after, you'll need to save the original and shortened URLs in a database. When a request comes in, the application checks if the URL exists and redirects to the original, or else it redirects to a 404 page.

- The Flask App should have a POST method which accepts a URL and returns a shortened URL
- The URL's should be saved in a local database with timestamps(Your choice of database)
- GET API which takes the shortened URL as path parameter and returns the full URL
- GET API to return ALL URL's available
- The above API should support filtering(date), search, and pagination
- Make sure to have logging enabled and log all actions(function calls, generated string, etc) in the application in log file
- Use combination of original URL and random string to shorten URL
- Write unit tests for the functions you have
- Make sure to use Wrapper functions to log the actions