Url Shortner

D.V.S.RajKiran (s160848@rguktsklm.ac.in)

October 31, 2021

1 Introduction

These days we usually prefer the things to be more pleasant to eyes, for that we develop so many things during the course of time. One of the things is shortening the length of the URL.

The sharing of web content is done by sharing the respected URLs only, but some URLs are way too long to be shared and visually bit stretchy also.So, the necessity of URL shortening is arrived. .

2 Process of shortening

2.1 What are the requirements

The shortening of URL is done by copying some random URL and pasting in the URL shortener application. The application basically, required both front-end and back-end

- Front-end consists of (HTML,CSS,Bootstrap)
- Back-end consists of Flask(python) and Database ORM

We integrate both the application front end and back end through Object Oriented Mapping(ORM). Because, it provides the flexibility to use desired database.

2.2 Creating Flask and database

The application can be accessed through the routes, so firstly we need to build the required routes for the application(like history,home,index...).After setting up the basic environment of flask to get started, we move on to establishing the databases in our system. For creating the database we need to follows certain rules they are as follows:

- **flask db init**—Initializes migration support for the application. Creates an empty revision script.
- **flask db migrate**—provides a convenient command-line interface for performing database migrations
- **flask db upgrade**— To sync the database in another system just refresh the migrations folder from source control and run the upgrade command.

After the completion of above 3 steps the database is created accordingly to the code written in the file.After laying the foundation we need to build floors on it likewise, we now needed to create the front end part part.

2.3 Creation of HTML files

The application will start of displaying the home first and asking the user to register if he is firstly doing the shortening after the successful registration he will get to login and will be access of shortening his desired URLs, this will be the workflow of the application. The first page we need to create is a layout html page. Because, for all the routes of the application the layout is common.It can be created once and inherited wherever it is required, which reduces the multiple times of writing the same thing.

Index.html After the creation of layout.html file, then we will create a home file which is going to be displayed firstly when user access the application.It will just greet like *Welcome to Url Shortner.*

Register.html When the user visits first time for the application for shortening the URLs he needs to have an account , if he didn't have one here we will make him one and it will redirect to login page where he has to enter the registered details.The username has been restricted to 5-9 characters only and already used username is not allowed to use again.

Login.html The login page allows the user to enter the fields like *Username* and *password* after successful login of the user will get the access of URL shortening application and view the history of the entries in the application.

URL typer.html Here the user will be given a text field to enter his URL and submit the button then he will be getting short version of the URL he entered and in addition he will get a copy to clipboard option, which will provide the user to copy the shortned URL to their clipboard.

History.html Here the entries will be displayed to the user, which he used the application and what URLs he entered.This is simply retrieving from database and displaying the data.

Home Shortner History Logout

History

Original Url	Shorten Url
https://colab.research.google.com/drive/143mmoOqlJ7kO5HLdPtOOEX2ispWa2oGy#scrollTo=c9tpMzO5HWHm	127.0.0.1:5000/4BD4TM7N
https://play.google.com/store/apps/details?id=us.zoom.videomeetings&hl=en_IN&gl=US	127.0.0.1:5000/AWE6K2QA
https://www.getmyuni.com/college/gandhi-institute-of-technology-management-gitam-hyderabad	127.0.0.1:5000/3XCJYBTT
https://twitter.com/search?q=indian+cricket+team&ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Esearch	127.0.0.1:5000/L4WR6DMB
https://indianexpress.com/article/sports/sport-others/neeraj-chopra-personalised-mahindra-xuv700-87-58-sticker-7600151/	127.0.0.1:5000/QP96KGXN
https://sports.ndtv.com/cricket/on-this-day-in-2005-dhoni-registered-his-highest-odi-score-2594379	127.0.0.1:5000/BLWUJZ5L
https://www.aljazeera.com/sports/2021/10/31/india-kohli-spineless-critics-after-shami-abuse	127.0.0.1:5000/BZU3PA55
https://en.wikipedia.org/wiki/Cornell_University_College_of_Architecture,_Art,_and_Planning	127.0.0.1:5000/0O3W6WHR
https://play.google.com/store/apps/details?id=org.telegram.messenger&hl=en_IN&gl=US	127.0.0.1:5000/1P0T90DK

Figure 1: History of URL shortener

2.4 How shortening happens

It is assigning a unique to key to the long and original url and the generation of key happens random and unique simultaneously to provide distinct keys to the given urls. The randomness is created by including uppercase, lowercase and digits inside the key making the key generation has so many number of unique keys.

2.5 Integrating both

Until now we separately created both html and database but we didn't integrate both the things.So, we use ***render template*** to bridge both the front-end and back-end parts and making the application to run successfully.

2.6 How to work without database?

The application can be run without creating the database thing. We need create a dictionary and push the original link as value into the dictionary and shortened version of the url is assigned as key.So,

the when access key which means when we call the shortened url the value (original) will be retrieved by the application. But the problem is dictionary is variable which is volatile and the data will be vanished after the system gets power disconnected.Due this problem we tend to go for the database.

3 Conclusion and future work

The URL shortener is created and implemented successfully with limited features and functionalities. This is just basic version of Url shortener application which consists of very limited features and very basic designing. where the designing can be improved and many more features can be added to application to make the application beautiful and elegant.