URL Shortener (Basic Version)

Module and Technology use: -

-Flask: -

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

-Jinja (web template engine): -

Jinja is a fast, expressive, extensible templating engine. Special placeholders in the template allow writing code similar to Python syntax. Then the template is passed data to render the final document.

-pyshorteners: -

A Python lib to wrap and consume the most used shorteners APIs.

-SQLite: -

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

-HTML: -

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser.

-JavaScript: -

JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. JavaScript on the client side for webpage behavior, often incorporating third-party libraries.

URL Shortener Working*:* -

Basically, the web-app accept the long URL from user and convert it into short URL and save the shortened URL in Database as well as the full URL.

The main long URL is accepted through the main home page and send in to back-end server which is design using Flask.

URL is sent through 'Message-Body' called POST Method.

In app-server, the URL is pass to main-route,

-Firstly, the URL is captured in Variable.

-Then the variable is passed to the pyshortener in which the pyshortener convert the long URL to short URL.

https://pyshorteners.readthedocs.io/en/latest/apis.html

-First, I had used 'Os.db' API, but this API contain Advertisement after opening the website which has Shortened, then I switch to 'Tiny URL' API which work goods without any advertisement.

-Then the Long URL and Short URL sent to the database to keep track and to display on the history.html page.

-Apart the Shortened URL is sent to front end page with render.template.

-After that I assign another route-history.

-In history route, the function fetch all the data from database and saving it in variable after that the variable is through render.template to history.html.

I have created 3 html page,

-layout.html

-home.html

-history.html

Layout.html contain the layout which is going to same on every html page. In this project the navbar is same so created navbar in layout which will be common through all html page.

Home.html contain home and history tab, and the form which have two text field one for accepting the long URL

and another for displaying the shortened URL and two button one for submitting the long URL and another is for copying the short URL.

For copy button I used JavaScript.

History.html display all the previous long URL and Shortened URL in Tabular Format.

I really enjoyed completing this project, little bit challenging one under guidance of Kanav Bansal sir.