URL Shortner App Report

- **Dhruv Dixit**
@Linkedin

A. What will our Web app do (Objectives)?

As the name suggests, it shortens URLs.

Users can also save URLS by coming to the web app.

B. Why do we need URL Shortener?

Sometimes we need to share or send links and this can be tiresome and annoying to copy and paste long URLs. That is where URL shorteners come in. Not only it helps in shortening the URL but it also allows the user to copy the shortened URL with a click of a button.

C. The project consists of following parts:

Frontend (done with HTML, CSS, JavaScript and Bootstrap)

Backend - Flask (Python)

Backend - Database ORM (SQLAlchemy)

Backend - Database (SQLite)

D. Front-End Information:

-> The front-end consists of 2 web pages:

Home Page - A page will be shown where the user can enter the URL he/she wants to shorten. After the 'shorten' button is clicked, the shortened URL is displayed in the text-field which the user can copy using the copy button.

History Page - Containing all the Original URLs along with the Shortened URLs.

E. Project Workflow

Users can enter the URL they want to shorten. After entering a URL, click on the 'Shorten' URL button to display the shortened URL in the following text-field which can be copied by clicking on the copy button.

After the 'Shorten' button is clicked, the URL that is entered is saved in our database with the shortened URL. It is saved in the database so that the user can look into the previous URLs he/she entered in our web-app with their shortened URL.

The app also verifies that whether the URL entered by the user is correct or not.

F. Approach

To build the app following packages were used:

i.) flask

ii.) flask_sqlalchemy

iii.) flask_migrate

iv.) os

v.) validators

vi.) random,string

Steps used were as follows:

1. Build a basic app.py and import the required modules from the given packages and code the basic flask template

2. Create a folder inside the project folder,i.e. the root directory of app.py and name it “templates”.

3. Create 3 HTML files inside templates folder and name them “layout.html”,“home.html” and “history.html”.

4. Go to root directory of app.py and create a new sqlite database using command prompt with admin access with 3 fields as “id” (Primary Key), “full_url” and “shorten_url”.

5. “layout.html” should contain all the basic template code with the navigation bar and should containg blocks for inheritance.

6. Build “home.html” and “history.html” using the concept of inheritance from “layout.html” and should contain relevant requests, forms and display content.

7. Give a CSS styling to both the HTML files.

8. In “history.html” add a button to copy the shorten URLs using JavaScript code snippets.

9. In app.py first locate the root directory (base folder) of app.py using “os” module and store it in a variable

10. Configure the SQLAlchemy ORM configurations and use the variable used to store path using os module to give the path of SQLite database.

11. Build a class URL and create the table with relevant columns as mentioned in step 4.

12. Create 3 routes “/”, “/history”, “/<finalurl>” with functions home(), history() and redirection() respectively.

13. “home()” - takes input URL in form of POST request and verifies the validity of URL using validators module and if URL is incorrect then it displays error otherwise checks that whether the entered URL is present in database or not. If present then returns its already created shorten URL otherwise creates a random string of 5 characters and attach it to the base “127.0.0.1:5000” URL (if not otherwise) and stores this link in a variable.

It then checks that the link in variable is already present in the database if present then generates new URL until a unique URL is found and attaches the Original URL and Shorten URL together and adds as an object to the database. It then finally displays the Shorten URL below the Original URL and this can copied using the Copy URL button.

14. “history()” - this fetches all the rows of database and finally displays each original URL and Shorten URL ever created row-by-row in a table in Frontend.

15. “redirection()” - this is used to provide the functionality of fetching the original URL from the database when a short URL is entered and open the relevant page associated with that Shorten URL.

Linkedin : https://www.linkedin.com/in/dhruv-dixit/

Github: https://github.com/dhruvdixit06/