My Report

What will our Web app do?

1. As the name suggests, it shortens URLs.
2. Users can also save URLS by coming to the web app.

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.

The project consists of 2 parts:

1. Frontend - HTML and Bootstrap
2. Backend - Flask (Python)
3. Backend - Database ORM (SQLAlchemy)

Front-End Information

The front-end consists of 2 web pages:

1. **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.
2. **History Page** - Containing all the Original URLs along with the Shortened URLs.

Project Workflow :

1. 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.

2. 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 entered in our web-app with their shortened URL.

3. Try to verify the URL entered by the user is correct or not. (Do some googling to find out how to make it possible)

URL Shortener Project :

1. Import Necessary Libraries:

Import Flask, render template, request, redirect, Flask SQL alchemy, Flask migrate, string and os,Requests.

2. Create a Flask App:

Create app using the code: app = Flask(__name__)

3. SQL_Alchemy Configuration:

Flask app config is the SQLALCHEMY_DATABASE_URI key. Pass the application into SQLALCHEMY class

4. Define Models and create table Schema:

Subclass db.Model to define a model class. The db object makes the names in SQL alchemy and sqlalchemy.orm available for convenience, it creates the schema of the database table. With the column names as we required.

5.Checking weather it is valid url or not:

By using the Requests library, using the method Requests.status,If the value returned equals to 200 then it is a valid one.

6. Create a Shorten URL Characters:

After validating the url,I used here seven characters with uppercase, lowercase and digits.

7. Create endpoints:

Basically, the "endpoint" is an identifier that is used in determining what logical unit of our code should handle the request. Normally, an endpoint is just the name of a view function. I created 2 endpoints 1.'/'->homepage 2.'/history'->list of changed url

8. Run the application:

Finally run the app in localhost with port no :5000 along with debug=True.

Issues faced during the project:

1. Not able to copy the url using the button ,so i used anchor tag directly so that u could directly go to the original url page,and the deadline is almost there.

2. Redirecting to original url :
I did ask many people about this on discord but couldnt resolve it, I checked in few references and checked my code again and again but i couldn’t ,I even posted my doubts on discord group as well,i hope u would help me to solve it at the end, but i created an alternative one when you use the shortened url you will get the original url that is the actual one.

thankyou