URL Shortener Web Application For Basic Users

What will our Web app do (Objectives)?

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 (done with HTML, CSS and Bootstrap)
2. Backend - Flask (Python)
3. Backend - Database ORM (**Click Here** for SQLAlchemy Reference)

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 Web Application For Advance Users

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.
- Users can also save URLS by logging in 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 3 parts:

1) Frontend (done with HTML, CSS and Bootstrap).
2) Database connection. (**Click Here** for SQLAlchemy Reference)
3) Flask (Python) backend.

Front-End Information

The front-end consists of 3 webpages:

1. The first page is Home where the user needs to login or signup if he/she doesn't have an account.
2. The second one is the signup page where the user will enter his/her details regarding username and password.
3. After signup, the page will be redirected to the login page where the user will enter his/her username and password to login.
4. After logging in, 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.

.

Project Workflow

1. During signup, the user has to enter a unique username. If he enters a username which is already taken then an error message will occur displaying "This username already exists…" Then the user needs to enter another username. The length of the username is valid only in the range of 5 and 9. If the username is not of valid length, then an error message will occur: "Username must be between 5 to 9 characters long".
2. After logging in, the main web-application opens which is the URL shortener. Here the user can enter the URL he wants 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.
3. 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 in the forms given below the copy button.

Adding Login Functionality
(NOTE - Use this to understand the Login functionality)

Project Code without Login - sabji_mandi_code_1.zip
Project Code with Login - sabji_mandi_code_2.zip

Follow the below mentioned steps before running the code available in sabji_madi_code_2.zip :

Step 1 - In command prompt run **pip install flask_login**
Step 2 - Go to the directory where **app.py** is present and run below mentioned commands in cmd (we are running these command again because there is the change in overall database because we have added the User table):

1. **flask db migrate -m "Message"**
2. **flask db upgrade**

Step 3 - Run the code and observe the code changes in the Sabji Mandi App