

URL Shortener Web Application (Basic)(Flask Application)

Step1 :

Created a python app file, templates folder under Submission main folder

Step2:

In templates folder created, HomePage.html, HistoryPage.html, Layout.html files

Step 3 :

In layout.html page created a layout for the application that is a title of the pages, heading of the pages, a dropdown menu containing the homepage , historypage links for the url application for navigation between the pages and used jinja templating to make title, body parts added features inherit property for the other html pages.

Step 4:

In HomePage.html used bootstrap css and html and created two input text fields one is enter url field and shorten url field ,the url field takes the input url to be shortened and shorten field will display the shorten url after submission through shorten submit button and for the result url to be displayed in the field used jinja templates to get the result from the backend application (app.py) to frontend application (homepage.html). Added copyshortened url button to copy the shorten url from the field shorten url to the clipboard and allows the user to paste the copied shorten url.

Step 5:

In HistoryPage.html displays the user entered url and their resultant shortened urls that are being stored in the sqlite database table .For this operation to be performed used jinja templates to get the result from the backend application that is app.py

Step 6:

1)Imported the required packages.

2)Created a Server

3) Define the routes and binding them to functions like created two routes '/' and '/HistoryPage' which is for homepage and historypage

4) Initially '/' loads which is homepage. This route has methods post, get request that carry the data through url's. Defined shortUrl function, inside this function using post request got the enter url field value from front end page, used pyshorteners function to shorten the url, while performing the operation got two exceptions one is ReadTimeOut which is raised when conversion of an url to short url takes more time i.e due session or response time is exceeded, BadUrlException which is raised when the user does not enter the url and press submit and for this exceptions I have used Exception handling technique, pyautogui module to pop alert messages when these exceptions occur

5) To send the shortened url to the shortened field used render templates to send the values to the front end from the back end.

6) Created Sqlite database to store the entered url and the resultant url

a) ORM object relation mapper connects the backend application the database here ORM used is sql alchemy.

b) As taken the current directory to create the database using os module.

c) Made configurations of the sqlite database, migrated the app, database, passed application into sql Alchemy

d) Created a Model i.e Class Urlshortener, given table name and created id, base url, shorten url columns of type string to store the url values and initialised the init function to run the code

e) In '/' route shorturl function created an object for the class Urlshortener, added the object to the database session method and committed the database session.

f) In command prompt to initialised the sqlite there commands have to run at the current project directory

1) flask db init

2) flask db migrate -m "First Migration"

3) flask db upgrade