URL Shortener with Flask and SQLite

13 August 2022
Chaitanya Dubal

Flask is a framework for building web applications using Python and SQLite is a database engine that you can use with Python to store application data.

Here we have used Hashids which is a python library that generates unique ID from intergers.For example , we can use to convert a number like 15 from a unique string 1BxVd .Hence we will use Hashids to generate the unique strings for URL ID.

Basic Requirements:

- We will have to install the Flask and the Hashids library by using the pip package installer.Then we have to create a programming background environment and then install the flask and hashids library in there.Further we need to create a database for storing the data that will be given to the website by user hence we will name this as schema.swl file while contains the SQL commands to create the table if there are any new URLs otherwise to drop the table if one exists.Thus the table create will contain different parameters like of ID, timestamp and original-url given by user and the clicks, which displays the number of times the URL has been clicked.
- We will also form a init_db.py file to create the urls table and then executing the schema.sql file . While in this file the connection to the database is been made and hence multiple SQL statements are executed at once and forming of url tables and then we commit the changes and close the connection.

Main program(app.py):

- Here we will start by importing the sqlite3 , hashids and flask libraries . After this now as this is the main application hence we will setup a database connection to this file by using the function get_db_connection().Here as the data of one user shouldn't be given to another user hence we use here sessions. And to secure the sessions we create a secret key , since the key is random string and we will also use it for hash which also keeps on changing also with the hashes(we also call the random string as salt which is hashing function from hashids).

- Index function : First we will set the route to ("/") along with the method get and post.If the request is GET request, then it will skips the code and will return the render_template named as index.html,which contain the form for the users to enter the url.If the request is POST request then condition being true will now take the value of url from the Form and store the url into the URL(variable).Then the value of the url is been passed into the database and pass the tuple containing url to database and hence commit the changes and closes the connection.We also have stored the url-id(variable) to store the id of the url we inserted into the database.

- We have constructed the hash using the hashids.encode() method and we passed the url-id to it and saved the result into hashid and hence same way the different hashes are produced for given url on based of the url-id.

- The main function we used here to construct the short URL is request.host_url which is an attribute to access the url of application host , hence for any url the domain is been extracted and along with that the salt we use the unique hash will be appended to it and stored into the variable as short_url(variable).

- For redirect the route: Here we will add a new route that takes the short has the application generate and decodes it into the integer value which result into original url's id.And will result into increment of the click value.Inside the view function, you first open a database connection. Then you use the decode() method of the hashids object to convert the hash to its original integer value and store it in the original-id variable. You check that the original-id has a value—meaning decoding the hash was successful. If it has a value, you extract the ID from it. As the decode() method returns a tuple, you fetch the first value in the tuple with original-id(), which is the original ID.

- Then we select the SQL statement to fetch the original url and its number of clicks from the table where ID is been matched.Further we increment the number of clicks of the url.

Adding the Statistics page: we will add a new route that displays how many times each URL has been clicked.Here in this we use view function and we open a database connection.Then we fetch the ID and other data and we use fetchall() method to get all

the rows.Hence we will save this data in the db_urls(variable) and then close the connection.We have use the dict() function to convert the sqlite3.row objects to a dictionary to allow the assignment and storage.After we have called short_url to the dictionary and with value reuqest.host_url and hashids.encode(url)is been used to construct the short URLs in index function,we then will append this to the dictionary to the urls list.In this route we will render the template called as stats.html and now close the file.In thhis html file we have extended giving different columns as short and originla and no.of clicks and the timestamp is been given to show the old urls given by the users.