Assignment Name-

Final Project (Basic Version) - URL Shortener Application

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 a 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 :

1. Frontend (done with HTML, CSS and Bootstrap)
2. Backend - Flask (Python)
3. ORM - SQLAlchemy
4. Database - SQLite

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.

Back-End Information & Database Information are mentioned on next pages.

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.

Approach for creating this URL Shortener web application :

Understanding the requirements: Before starting the coding part in flask(for backend) , I made sure to understand the requirements of the web application. In this case, the requirements were to create a web application that allows users to enter a long URL and get a shortened URL . The web application should also store the original and shortened URLs in a database.

Setting up environment: I installed the necessary tools and libraries required to develop the web application. I used Flask framework for developing URL shortener web app. I also installed 'SQLAlchemy' and 'Flask-Migrate' libraries for managing the database.

First, I have created a virtual environment for flask. Then, created app.py file which is the main file or an application server file. Then imported render_template ,request from flask and also imported os.

Then created "templates" folder which will store all .html files which will render on front end.

SQLALchemy is a popular SQL toolkit and Object Relational Mapper(ORM). It is written in Python and gives full power and flexibility of SQL to an app dev.

To install this, used (pip install Flask-SQLAlchemy) command in my virtual environment and also (pip install flask-migrate).

Then for SQLAlchemy Configuration, these syntax is used. This syntax is always same and used every time to configure.

```
7
8   ######### SQL Alchemy configuration...
9
10  basedir=os.path.abspath(os.path.dirname(__file__))
11  app.config['SQLALCHEMY_DATABASE_URI']='sqlite:///'+os.path.join(basedir,'data.sqllite')
12  app.config['SQLALCHEMY_TRACK_MODIFICATIONS']=False
13
14  db=SQLAlchemy(app)
15  Migrate(app, db)
16
17  ######### Model Creation
```

Then passed app into SQLAlchemy using “db” and “migrate”.

Creating Database: I created a Model and inherited from db.Model. Then I defined the schema of the database for the URL_History table, which has 3 columns: an id column, Original_URL and a Shorten_URL column. I set id column as a Primary key of the Table.

Then configured app to use SQLite database created earlier. And also set up Flask-Migrate extension to manage the database migrations…

flask db init

flask db migrate

fask db upgrade

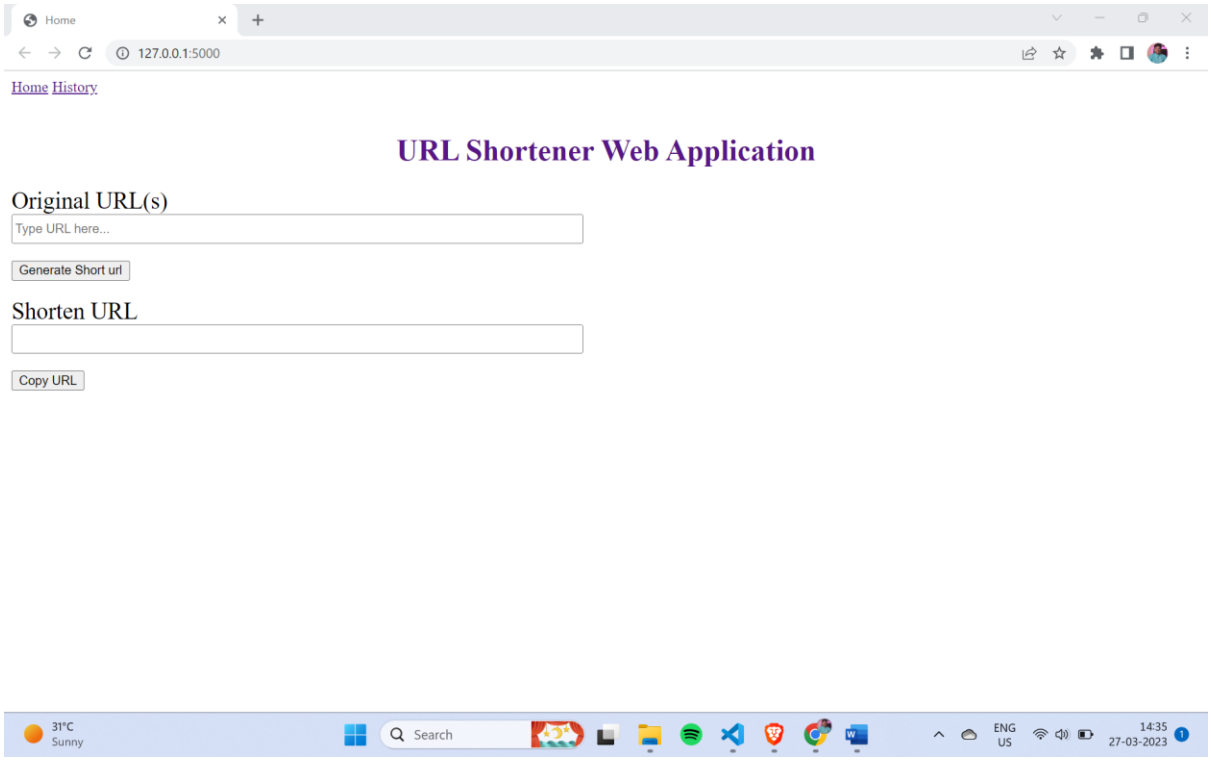
these 3 commands are used..

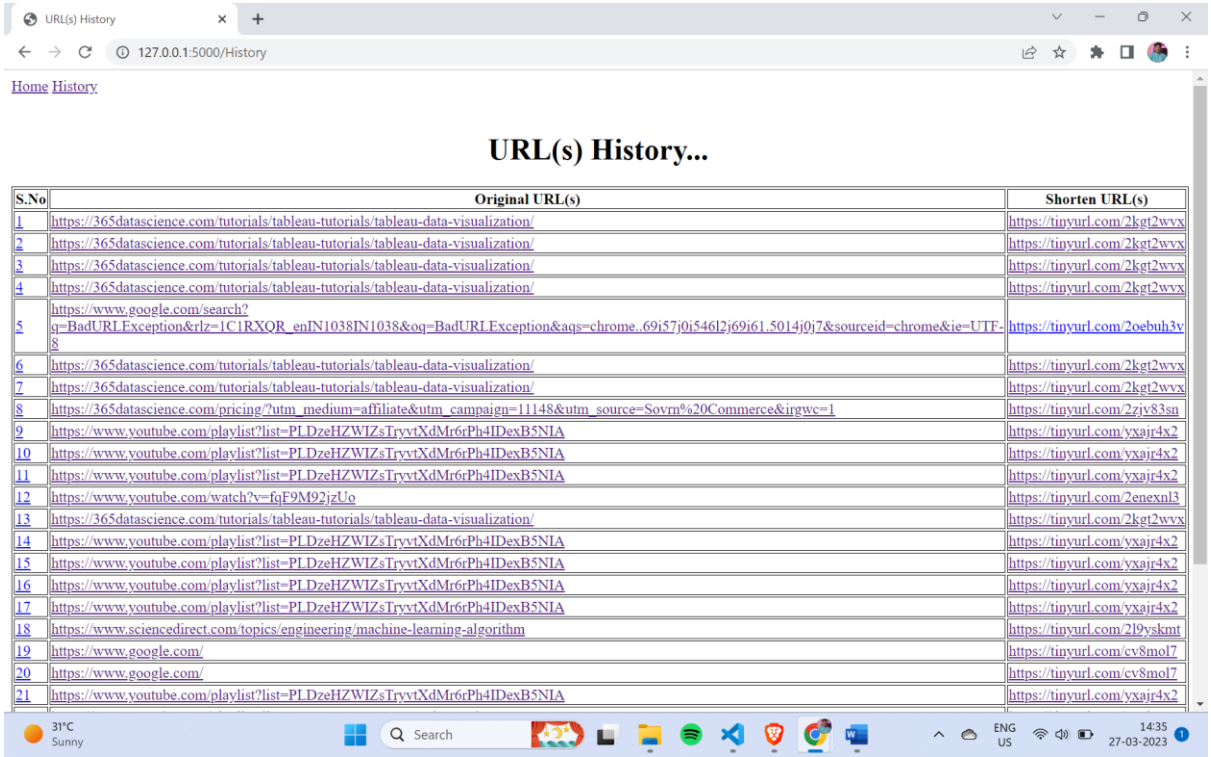
Developing views: i created 2 views for app. First view is for the home page , where users can enter Original URL(s) which are generally long url and generate a short url. The second is for the History web page, which display a list of all the URLs that have been shortened.

The logic or core functionality: In Home, I used pyshorteners library to generate the shortened URL. Then created a new URL_History object and added it to the database session. And then Finally, I committed a session to save the new URL to our database.

In templates folder, home.html file, history.html file and the last one layout.html are created. Layout file contains forms, and “url_for(“home”)” and “url_for(“history”)”. This layout file will be used for inheriting home and history files.

In Home file “navigator.clipboard.writeText” method is used to copy the text on clipboard (Shortened Url). And History page will show a table containing Serial.No. , Original URL(s) and Shorten URL(s).


Home
127.0.0.1:5000
Home History
URL Shortener Web Application
Original URL(s)
Type URL here...
Generate Short url
Shorten URL
Copy URL
31°C
Sunny
Search
ENG
US
14:35
27-03-2023


URL(s) History
127.0.0.1:5000/History
Home History
URL(s) History...
S.No
Original URL(s)
Shorten URL(s)
1 https://365datascience.com/tutorials/tableau-tutorials/tableau-data-visualization/ https://tinyurl.com/2kgt2wvx
2 https://365datascience.com/tutorials/tableau-tutorials/tableau-data-visualization/ https://tinyurl.com/2kgt2wvx
3 https://365datascience.com/tutorials/tableau-tutorials/tableau-data-visualization/ https://tinyurl.com/2kgt2wvx
4 https://365datascience.com/tutorials/tableau-tutorials/tableau-data-visualization/ https://tinyurl.com/2kgt2wvx
5 https://www.google.com/search?q=BadURLException&rlz=1C1RXQR_enIN1038IN1038&oq=BadURLException&aqs=chrome..69i57j0i546l2j69i61.5014j0j7&sourceid=chrome&ie=UTF-8 https://tinyurl.com/2oebuh3v
6 https://365datascience.com/tutorials/tableau-tutorials/tableau-data-visualization/ https://tinyurl.com/2kgt2wvx
7 https://365datascience.com/tutorials/tableau-tutorials/tableau-data-visualization/ https://tinyurl.com/2kgt2wvx
8 https://365datascience.com/pricing/?utm_medium=affiliate&utm_campaign=11148&utm_source=Sovrn%20Commerce&irgwc=1 https://tinyurl.com/2zjv83sn
9 https://www.youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA https://tinyurl.com/yxajr4x2
10 https://www.youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA https://tinyurl.com/yxajr4x2
11 https://www.youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA https://tinyurl.com/yxajr4x2
12 https://www.youtube.com/watch?v=fqF9M92jzUo https://tinyurl.com/2enexnl3
13 https://365datascience.com/tutorials/tableau-tutorials/tableau-data-visualization/ https://tinyurl.com/2kgt2wvx
14 https://www.youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA https://tinyurl.com/yxajr4x2
15 https://www.youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA https://tinyurl.com/yxajr4x2
16 https://www.youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA https://tinyurl.com/yxajr4x2
17 https://www.youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA https://tinyurl.com/yxajr4x2
18 https://www.sciencedirect.com/topics/engineering/machine-learning-algorithm https://tinyurl.com/2l9yskmt
19 https://www.google.com/ https://tinyurl.com/cv8mol7
20 https://www.google.com/ https://tinyurl.com/cv8mol7
21 https://www.youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA https://tinyurl.com/yxajr4x2
31°C
Sunny
Search
ENG
US
14:35
27-03-2023