INNOMATİCS
RESEARCH LABS

INTERNSHIP PROJECT REPORT

ON

WEB BASED URL SHORTENER

SUBMITTED BY

DIVYA MANOJ

BATCH

OCT – 21

GUIDED BY

CHIEF DATA SCIENTIST “KANAV BANSAL”

Oct 28, 2021

Web-based URL Shortener

Summary

URL shortening is a technique on the World Wide Web in which a URL (Uniform Resource Locator) made substantially shorter and still redirect a user to a required webpage. This is achieved by using the redirect() which link webpages to each other for redirection and often the redirect name is shorter than the original one. A friendly URL may be desired for messaging technologies that limits the number of characters in a message (for example SMS), for reducing the amount of typing required is the reader is copying the URL from a print source, also for making it easier for a person to remember.

Purpose

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. Thus, short URLs may be more convenient for a websites or hard copy publications.

Project Structure

The project consists of 2 parts:
1. Frontend (done with HTML and CSS)
2. Backend - Flask (Python)
3. Backend - Database ORM

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.

Technology Stack

1. Python
   Python is an interpreted high-level general-purpose programming language. It supports multiple programming paradigms,
   including structured (particularly, procedural), object-oriented and functional programming.

2. Flask
   Flask is a Python web framework built with a small core and easy-to-extend philosophy. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

3. HTML 5
   HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content.

4. CSS
   CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces.

5. SQLite3
   SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

6. Flask-SQLAlchemy
   Flask-SQLAlchemy is an extension for Flask that adds support for SQLAlchemy to your application. It aims to simplify using SQLAlchemy with Flask by providing useful defaults and extra helpers that make it easier to accomplish common tasks.

Tool Stack

1. Visual Studio Code
   Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS.[9] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Working Phases

The system working is dived into 4 phases -

1. URL Input – The goal of the URL input is to accept the valid URL from auser and pass it to the URL shortening module to shorten the URL.

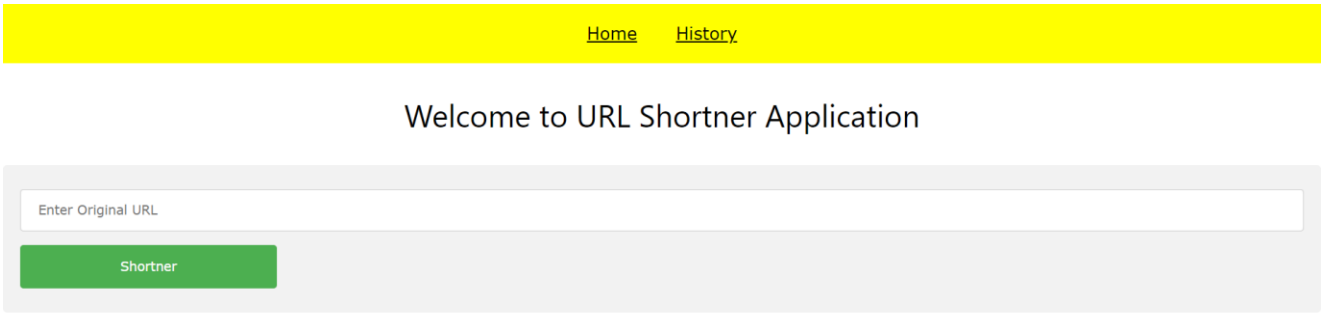
2. URL Shortening -- The goal of this phase is to map the given URL with max of 4 unique alphanumeric character. hashids library was used to generate unique ids.

3. URL Mapping – The goal of this phase is to map the short and long URL in the database so that the URL redirection phase can use this information for future use.

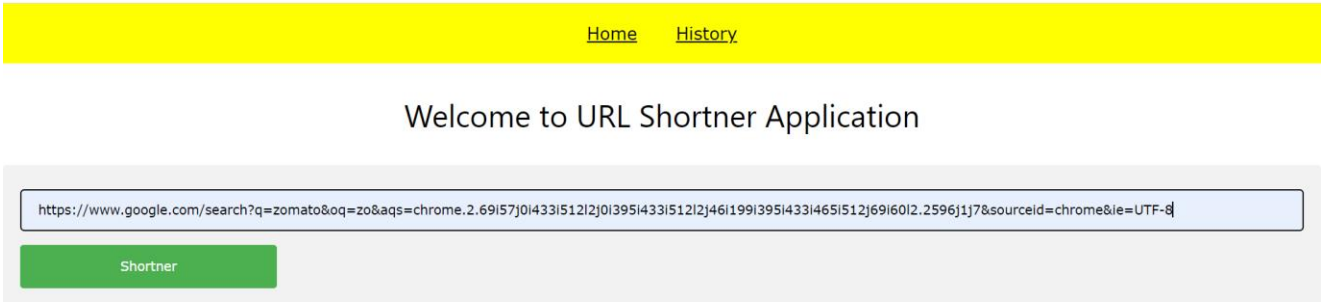
4. URL Redirection – The goal of this phase is to enable shorten URL to redirect to the original URL web page.

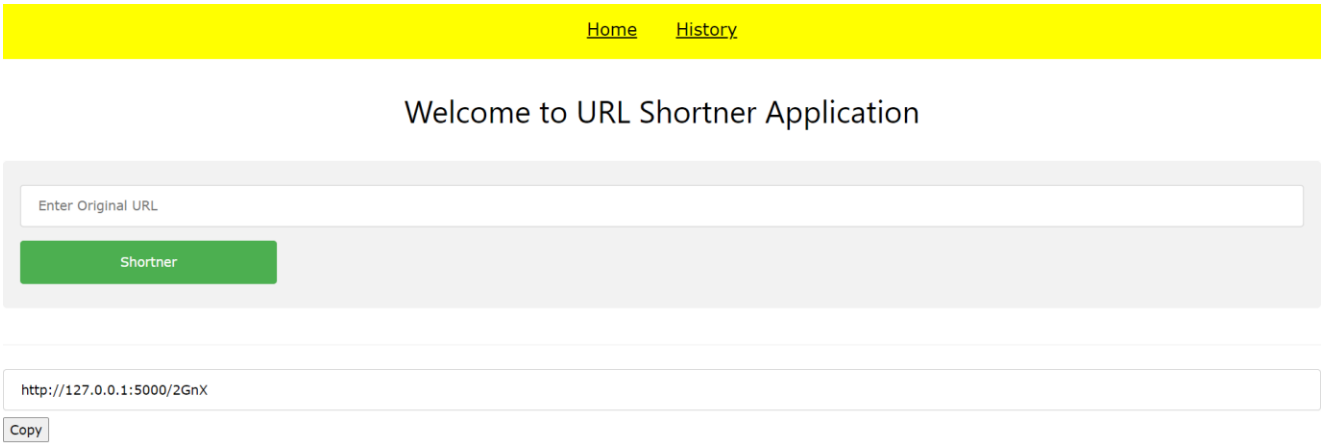
Output

Home Page



When user enters url and clicks ‘Shortner button’





History Page