



[Flask Templates](#) [Jinja2](#) [Flask-REST API](#) [Python SQLAlchemy](#) [Flask Bcrypt](#) [Flask Cookies](#) [Json](#) [Postman](#)

Create A Bookmark Organizer For Website Using Flask

Last Updated : 06 May, 2024

This article explains how to make a tool called a Bookmark Organizer with Flask for a website. With this tool, we can add, create, update, delete, and edit bookmarks easily. We just need to input the title and link, and we can save them. Then, we can click on the link anytime to visit the website.

What is a Bookmark Organizer?

A Bookmark Organizer is a tool or application that helps users manage and keep track of their bookmarks or favorite web links. It allows users to add, create, update, delete, and edit bookmarks in an organized manner. Typically, users can save the title and link of a website, making it convenient to revisit the websites they find interesting or frequently visit. The Bookmark Organizer aims to simplify and enhance the user's experience.

Create A Bookmark Organizer For Website Using Flask

Below, are the step-by-step guide to Create A Bookmark Organizer For Website Using [Flask](#) in Python

Create a Virtual Environment

First, create the [virtual environment](#) using the below commands

```
python -m venv env
.\env\Scripts\activate.ps1
```

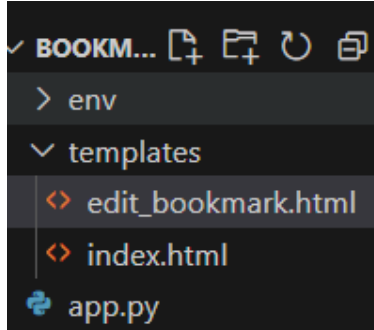
Install Neccassary Library

To properly create a Bookmark Organizer for a website using Flask, it is essential to install the necessary library. To install Flask, use the following

command:

```
pip install flask
```

File Structure :



File Structure

Setting Necessary Files

app.py : In below code, the Flask application creates a simple Bookmark Organizer. It includes routes for displaying bookmarks, adding new bookmarks, deleting bookmarks by ID, and editing bookmark details. The bookmarks are stored in a list, and the application utilizes Flask's functionality for rendering templates and handling form submissions. The program runs as a Flask web server, allowing users to manage their bookmarks through the provided routes.

Python

```
from flask import Flask, render_template, request, redirect, url_for

app = Flask(__name__)

# List to store bookmarks
bookmarks = []

# Route to display bookmarks on the index page

@app.route("/")
def index():
    return render_template("index.html", bookmarks=bookmarks)

# Route to add a new bookmark
```

```

@app.route("/add_bookmark", methods=["POST"])
def add_bookmark():
    # Get the title and URL from the form
    title = request.form.get("title")
    url = request.form.get("url")

    # Check if both title and URL are provided
    if title and url:
        # Create a new bookmark dictionary
        bookmark = {"id": len(bookmarks) + 1, "title": title, "url": url}
        # Add the new bookmark to the list
        bookmarks.append(bookmark)

    # Redirect to the index page after adding a bookmark
    return redirect(url_for("index"))

# Route to delete a bookmark based on its ID

@app.route("/delete_bookmark/<int:bookmark_id>")
def delete_bookmark(bookmark_id):
    global bookmarks
    # Create a new list without the specified bookmark ID
    bookmarks = [
        bookmark for bookmark in bookmarks if bookmark["id"] !=
bookmark_id]
    # Redirect to the index page after deleting a bookmark
    return redirect(url_for("index"))

# Route for editing a bookmark

@app.route("/edit_bookmark/<int:bookmark_id>", methods=["GET", "POST"])
def edit_bookmark(bookmark_id):
    # Find the bookmark with the specified ID
    bookmark = next((bm for bm in bookmarks if bm["id"] == bookmark_id),
None)

    if bookmark:
        if request.method == "POST":
            # Update bookmark details based on the form submission
            bookmark["title"] = request.form.get("title")
            bookmark["url"] = request.form.get("url")
            return redirect(url_for("index"))
        else:
            # Render the edit bookmark form
            return render_template("edit_bookmark.html",
bookmark=bookmark)
    else:
        # Redirect to the index page if the bookmark is not found
        return redirect(url_for("index"))

# Run the Flask application
if __name__ == "__main__":
    app.run(debug=True)

```

Creating GUI

index.html: This Your HTML template is designed for a Bookmark Organizer web page using Bootstrap. It includes a form for adding bookmarks and a list for displaying existing bookmarks, each with options to visit, edit, and delete. The layout is clean and responsive. Integrate it with your Flask app for dynamic functionality.

HTML



```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bookmark Organizer</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.
css" rel="stylesheet"
  integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
</head>

<body class="container mt-5">

  <h2 class="mb-4">Bookmark Organizer</h2>

  <form action="/add_bookmark" method="post" class="mb-4">
    <div class="form-group">
      <label for="title">Title:</label>
      <input type="text" class="form-control" id="title"
name="title" required>
    </div>

    <div class="form-group">
      <label for="url">URL:</label>
      <input type="url" class="form-control" id="url" name="url"
required>
    </div>

    <button type="submit" class="btn btn-primary mt-4">Add
Bookmark</button>
  </form>

  <ul class="list-group">
    {% for bookmark in bookmarks %}
    <li class="list-group-item d-flex justify-content-between align-
items-center mb-2">
      <div>
        <b>{{ bookmark.title }}</b>
```

```

</div>
<div>
  <a href="{{ bookmark.url }}" target="_blank" class="btn
btn-success btn-md mr-1">Visit</a>
  <a href="/edit_bookmark/{{ bookmark.id }}" class="btn btn-
warning btn-md mr-1">Edit</a>
  <a href="/delete_bookmark/{{ bookmark.id }}" class="btn
btn-danger btn-md">Delete</a>
</div>
</li>
{% endfor %}
</ul>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle
.min.js"
  integrity="sha384-
C6RzsynM9kWDrmMNeT87bh950GNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
  crossorigin="anonymous"></script>
</body>

</html>

```

edit_bookmark.html: This HTML template is designed for editing a bookmark using Bootstrap. It includes a form with fields to modify the title and URL of the bookmark. The form is pre-filled with the existing values, allowing users to update the bookmark details. The template also has buttons for submitting the update and canceling the operation.

HTML



```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Edit Bookmark</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.
css" rel="stylesheet"
  integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
  crossorigin="anonymous">
</head>

<body class="container mt-5">

  <h2 class="mb-4">Edit Bookmark</h2>

  <form action="/edit_bookmark/{{ bookmark.id }}" method="post"
class="mb-4">
    <div class="form-group">

```

```
<label for="title">Title:</label>
<input type="text" class="form-control" id="title"
name="title" value="{{ bookmark.title }}" required>
</div>

<div class="form-group">
<label for="url">URL:</label>
<input type="url" class="form-control" id="url" name="url"
value="{{ bookmark.url }}" required>
</div>

<button type="submit" class="btn btn-primary mt-4">Update
Bookmark</button>
</form>

<a href="{{ url_for('index') }}" class="btn btn-secondary">Cancel</a>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle
.min.js"
integrity="sha384-
C6RzsynM9kWDrmNeT87bh950GNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
</body>

</html>
```

Run the server with the help of following command:

```
python app.py
```

Output

Before adding Bookmark

Bookmark Organizer

Title:

URL:

Add Bookmark

After Adding GeeksforGeeks as a Bookmark

Editing the Saved Bookmark



goura...



Previous Article

Form Submission API with Swagger Editor
and Python Flask

Next Article

Similar Reads

Automated File Organizer using Tkinter in Python

Prerequisites: Tkinter File organization is helpful when you are trying to find a specific one in a pool of files and directories. It seems time-consuming and...

2 min read

Junk File Organizer in Python

Basically, as a lazy programmer, my desktop is full of files (Junk Files). Due to a large number of files, it is a daunting task to sit and organize each file. To make...

3 min read

Documenting Flask Endpoint using Flask-Autodoc

Documentation of endpoints is an essential task in web development and being able to apply it in different frameworks is always a utility. This article discusses...

4 min read

Minify HTML in Flask using Flask-Minify

Flask offers HTML rendering as output, it is usually desired that the output HTML should be concise and it serves the purpose as well. In this article, we would...

12 min read

Create URL Bookmark Manager Using Django

This article explains how to make a tool called a Bookmark Organizer with Django for a website. With this tool, we can add, create, update, delete, and edit...

5 min read

How to use Flask-Session in Python Flask ?

Flask Session - Flask-Session is an extension for Flask that supports Server-side Session to your application. The Session is the time between the client logs in to...

4 min read

How to Integrate Flask-Admin and Flask-Login

In order to merge the admin and login pages, we can utilize a short form or any other login method that only requires the username and password. This is know...

8 min read

Flask URL Helper Function - Flask url_for()

In this article, we are going to learn about the flask url_for() function of the flask URL helper in Python. Flask is a straightforward, speedy, scalable library, used f...

11 min read

Build Blog website using Flask

In this article, we'll explore how to build a dynamic blog website using Flask, a lightweight and versatile Python web framework. Flask provides developers wit...

15+ min read

Create a Weather app using Flask | Python

Prerequisite : Flask installation Flask is a lightweight framework written in Python. It is lightweight because it does not require particular tools or libraries...

2 min read

Article Tags : [Python](#) [Python Flask](#) [Flask Projects](#)

Practice Tags : [python](#)



Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,

Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

Company

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

DevOps

Git
Linux
AWS
Docker
Kubernetes

Engineering Maths
Software Development
Software Testing

Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved