Flask – Message Flashing

Last Updated : 26 Jun, 2024

In this article, we will discuss Flask – Message Flashing. As we know best Graphical User Interface provides feedback to a user when users interact, as an example, we know that desktop applications use the messages box or JS for an alert purpose. generating like same informative message is easy to display in the Flask web application. The flashing message of the Flask web application makes Flask easier and more useful for users.

What is the Message Flashing

Message Flashing means when we click on any button(any website) and the website immediately any message that can be an error, alert, or any type of message that's call message Flashing and when that happens in a flask or we are creating any website which shows an alert, error type message than its call Flask-Message Flashing.

A Flask contains a ***flash()*** function. which passes a request to the next user. which generally is a template we are using flash methods in the above template in which the **message** parameter is the real message to be flashed, and another one is the optional category parameter **category**, it will happen "error" or "warning".

```
flash(message, category)
```

To understand flashing messages and know about more flashing messages, we start to write code for flashing messages we take an example of a login system so let's write code for flashing message. Create a Virtual Environment.

app.py File

Here, we create an app.py file in which we write our main code of Flask.

In app.py first, we import flask then we initialize the flask function and create a secret key which will help us when we forgot our password then we made two functions one home() for index.html which will help us to call and display the home page when we run flask using flask run command and another one for login login.html we joint both files by using form action and after that, we use if and else condition for flashing message (error or warning) and also for redirect on profile page by filling right password and also right Email ID and after that in if and else condition we create password GFG we can set any password it up to you and also write error flashing message which will show when we fill the wrong password and click on submit button for login and if we fill right password and email id it will redirect us on profile page were also showing one welcome flashing message which we create using python code in app.py file.

That error message which is showing in the above images when we enter the wrong password is called a flashing message also if we fill right password and email id and click on submit button then it will redirect us to the profile page which will also show a welcome flashing message so this way we can display flashing message in our display screen. It will also help to know how to show flashing messages.

[GFGTABS] Python3

```
from flask import *

# Initialize Flask function
app = Flask(__name__)
app.secret_key = "GeeksForGeeks"



# home function for index.html
@app.route("/index")
def home():
    return render_template("index.html")



# row function for profile.html
@app.route("/profile")
```

```
def row():
    return render_template("profile.html")


# write if and else condition if we provide write password then he will redirect
# us in profile page otherwise he will redirect us on same page with
# flashing message Invalid Password
@app.route("/login", methods=['GET', 'POST'])
def login():
    error = None
    if request.method == "POST":
        if request.form['pass'] != "GFG":
            error = "Invalid Password"
        else:
            flash("You are successfully login into the Flask Application")
            return redirect(url_for('row'))

    return render_template("login.html", error=error)


# execute command with debug function
if __name__ == '__main__':
    app.run(debug=True)
```

[/GFGTABS]

Templates File

index.html

In this code, we are writing some flashing (error code) message command and we connect this page to login by using the URL "/login" in the app.py file it means when we click on the URL " login "redirects us to the login page. where we can log in. and we can enter our profile by filling right password which we will set in the below code of the app.py file.

[GFGTABS] HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
</head>
<body>

{% with messages = get_flashed_messages()%}
{% if messages%}
    {% for message in messages%}
    <p>{{message}}</p>
    {%endfor%}
    {%endif%}
    {%endwith%}
    <h3>Welcome to the GeeksForGeeks</h3>
    <a href="{{url_for('login')}}">login</a>

</body>
</html>
```

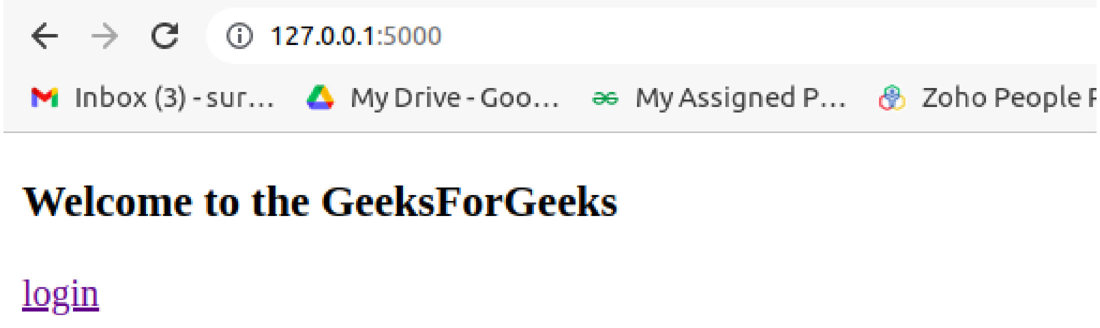
[/GFGTABS]

When we run the following command in our terminal.

```
flask run
```

Output:

This will show on our display screen and it also shows our flask is successfully running.



Welcome to the GeeksForGeeks

login

index.html

login.html

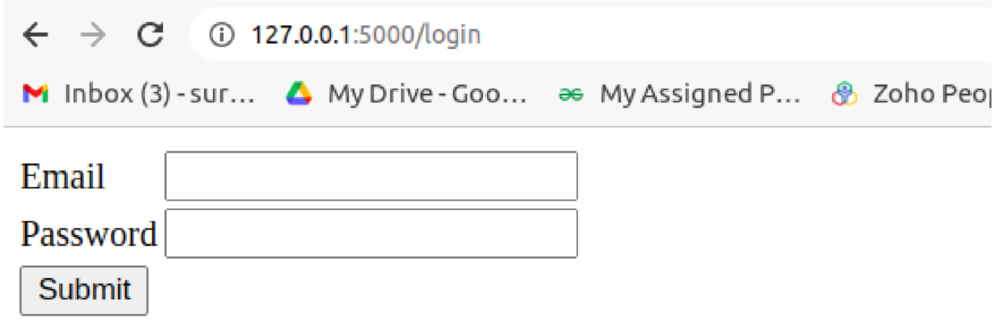
In the login page, we create simple input for the password and email which is required to fill .and also write three lines of python code for showing a flashing message when we enter the wrong password and try to log in and also set both inputs in from tag in which we create one method post and action "/login" which means when we click on previous page login link so it will redirect us on this login form page and we write for one button in which we set the value to submit for submitting our login form after filling Email and password.

[GFGTABS] HTML

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Login</title>
</head>

<body>
    {% if error%}
    <p><strong>{{error}}</strong></p>
    {% endif%}
    <form method="post" action="/login">
    <table>
        <tr>
            <td>Email</td>
            <td><input type="email" name="email"></td>
        </tr>
        <tr>
            <td>Password</td>
            <td><input type="password" name="pass"></td>
        </tr>
        <tr>
            <td><input type="submit" value="Submit"></td>
        </tr>
    </table>
</form>
</body>

</html>
```

[/GFGTABS]

Output:



login.html

profile.html

In this code, we made a simple message that will show after successful login. it will also show us the welcome flashing message which we create in the app.py file by using the flask function.

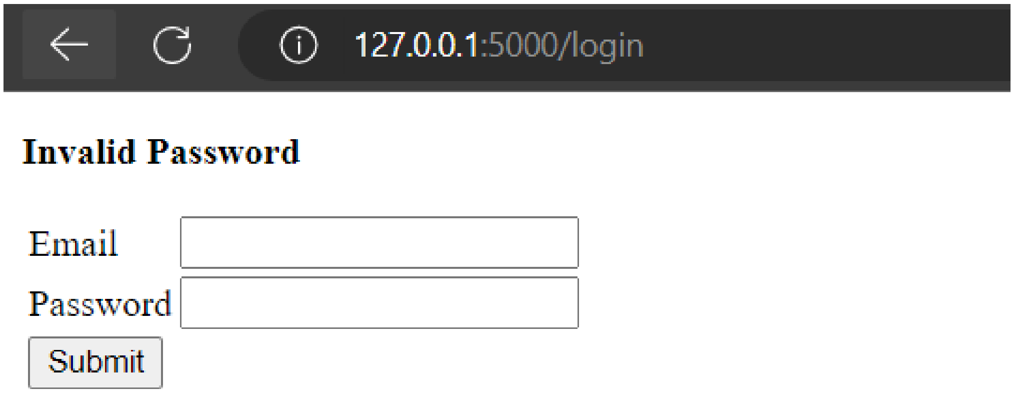
[GFGTABS] HTML

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Profile</title>
</head>
<body>
    <div class="container">
        <p><strong>Welcome to your GFG Profile Page</strong></p>
    </div>
</body>

</html>
```

[/GFGTABS]

Final Output of unsuccessful Flashing Message

If we write the wrong Password and press submit button so it will show a flashing message (error or warning) like the below output image **Invalid Password** is called **Flashing Message.** and if we fill right password and email it will redirect us to the profile page which will also show one flashing welcome message on our display screen.



127.0.0.1:5000/login

Invalid Password

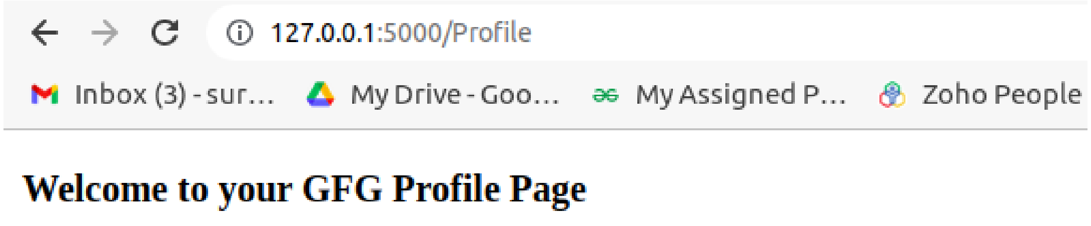
Email

Password

Submit

profile.html

The output of successful Flashing Message



Welcome to your GFG Profile Page

profile.html

Looking to dive into the world of programming or sharpen your Python skills? Our [Master Python: Complete Beginner to Advanced Course]() is your ultimate guide to becoming proficient in Python. This course covers everything you need to build a solid foundation from fundamental programming concepts to advanced techniques. With **hands-on projects**, real-world examples, and expert guidance, you'll gain the confidence to tackle complex **coding challenges**. Whether you're starting from scratch or aiming to enhance your skills, this course is the perfect fit. Enroll now and master Python, the language of the future!

prath... 3

Previous Article

Upload Multiple files with Flask

Next Article

Create Contact Us using WTForms in Flask

Similar Reads

Python Pyramid - Message Flashing

In this article, we will explore the concept of Python flashing through a practical example. We'll be creating a registration page where, upon entering the correct...

5 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

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

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

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

Python Flask - ImmutableMultiDict

MultiDict is a sub-class of Dictionary that can contain multiple values for the same key, unlike normal Dictionaries. It is used because some form elements ha...

2 min read

Handling 404 Error in Flask

Prerequisite: Creating simple application in Flask A 404 Error is showed whenever a page is not found. Maybe the owner changed its URL and forgot to...

4 min read

Python | Using for loop in Flask

Prerequisite: HTML Basics, Python Basics, Flask It is not possible to write front-end course every time user make changes in his/her profile. We use a template...

3 min read

How to Build a Simple Android App with Flask Backend?

Flask is an API of Python that allows us to build up web applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's...

8 min read

Article Tags : Python Technical Scripter Technical Scripter 2022

Practice Tags : python

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

Engineering Maths

Software Development

Software Testing

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

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

Azure

GCP

DevOps Roadmap

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