

Introduction to Micro Web Framework Flask

Understand the concept of micro web framework Flask in Python.

Introduction

Web Development in today's time is very important for any business. Many big companies and startups rely on a great website to attract many customers.

Web Development for a Data Scientist is essential today because to a customer or user, demonstrating the code and showing the output in a terminal form wouldn't make any sense if he is not from a programming background. This may lead to rejections.

To show the functionality of the machine learning or deep learning model, deploying it as a website or a web app makes sense as it is easy to show users the core functionality in a graphical form which tends to attract more users.

Web Development has become quite the way for a Data Scientist or any AI enthusiast to show his algorithm, a project that he has worked on. This is why most machine learning and deep learning engineers use some web development to deploy their models. But, how can a Data Scientist quickly deploy his/her AI model? Well, to answer that question we have something called **Flask** which helps us achieve it.

Overview

Let's take a look at the contents that we will cover in this article.

- What is Flask?
- Why use Flask?
- Merits of using Flask
- Prerequisites
- Installing Flask
- Implementation of Flask
- Conclusion

What is Flask?

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries

provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies, and several common framework-related tools.



Flask

Flask is very easy to learn and quick to implement. It is preferred by many Data Scientists as it is quick to implement and easy to show a demo of their models. As Flask is written in Python, Developers who are familiar with Python find it not difficult to learn Flask. Though Flask has no additional tools or libraries that a normal

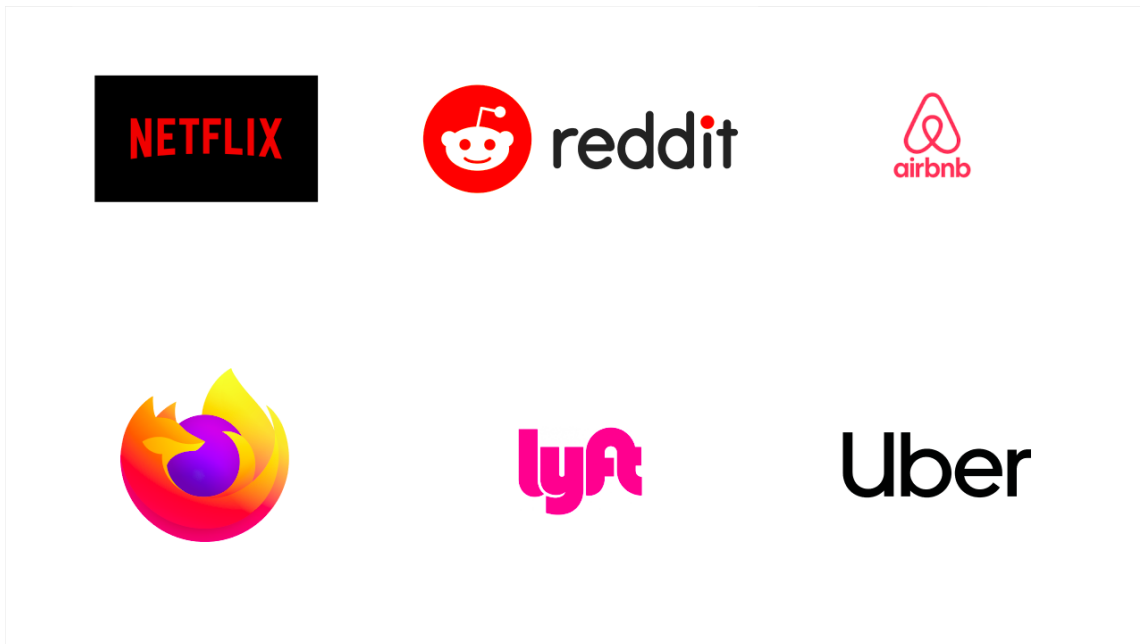
framework would have, it makes it easy to implement other services like databases.

Why use Flask?

Flask is a lightweight Web Server Gateway Interface Web Application (WSGI) framework. It gives the developer varieties of choice when developing web applications, it provides you with the necessary tools to build and deploy a web application. It does not enforce any dependencies or give a fixed project structure like other Python Web Frameworks like Django offers. This adds to the advantage side by making it useable by many developers. The website can be on anything, but still, it allows the developers the opportunity to use some extensions provided by the community that allows you to add more functionality to the web application.

Because of its simplicity, Flask is used in top tech companies like

- Netflix
- Reddit
- Airbnb
- Lyft
- Mozilla
- Uber

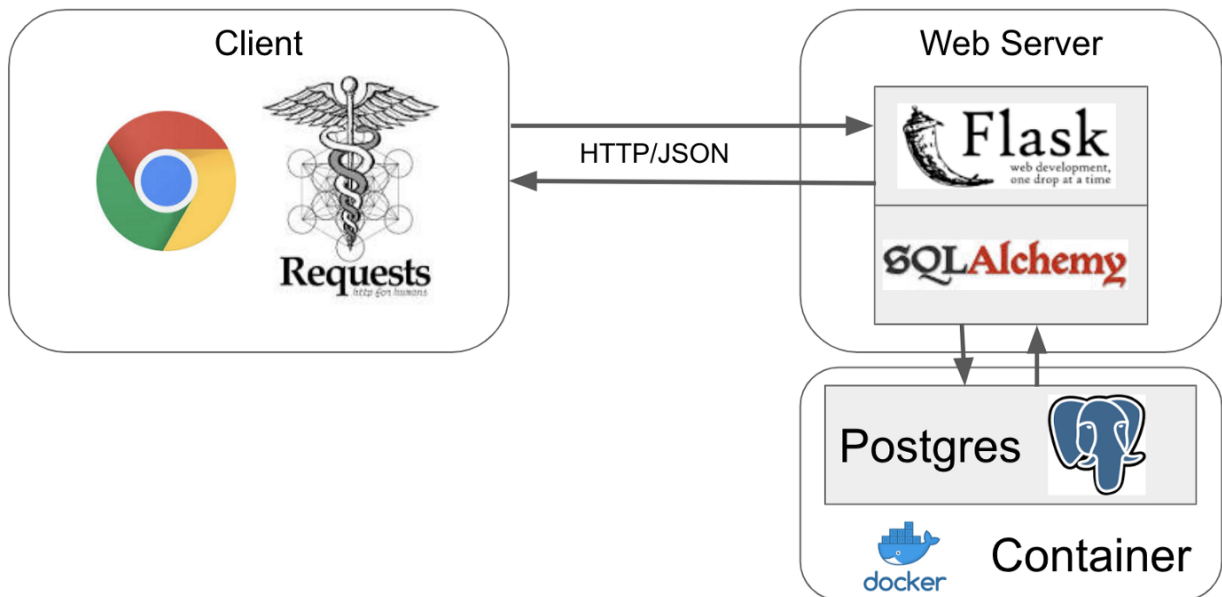


Merits of using Flask

As said above Flask is a lightweight Web Server Gateway Interface (WSGI) web application framework. It has minimal or no external libraries. It is used to create a simple website and then scale it up to complex applications. Below are some of the merits of using Flask.

- **Easy to Use:** The Flask framework is easy to understand, which makes it perfect for beginners. The simplicity in the flask framework enables the developer to navigate around and create the application easily.
- **Very Flexible:** Most of the Flask components can be altered. It allows users to customize the website.
- **Testing:** It allows unit testing through its integrated support, built-in development server, fast

debugger, and restful request dispatching. Makes testing much faster.



Prerequisites

You'll need the following to run Flask.

- Python (latest version recommended)
- Flask (latest version recommended)
- virtualenv (latest version recommended)

That's it we are all set to install Flask and create a simple website.

Installing Flask

Installing Flask is pretty easy. All you have to do is just type one command. That's it.

Note: It is advised to use a virtual environment for any Python project you start.

First, install the virtualenv package using the pip command.

- `pip install virtualenv`

Go ahead and create your virtual environment using the following command.

- `virtualenv flask_env`

Note: Here flask_env is the name of the environment created.

Now activate your virtualenv and install the requirements.

For Linux based systems

- `source flask_env/bin/activate`

For Windows,

- `flask_env\Scripts\activate`

After activation install Flask.

- `pip install flask`

This should install all the necessary Flask modules.

Implementation of Flask

Before implementing Flask, we will just take a look at our project structure.

.

```
|— app.py
|— templates
|   |— index.html
1 directory, 2 files
```

All we did was to create an app.py and then a folder called templates and inside templates we have index.html.

Note: You should also have a folder called flask_env which is our flask environment.

Now jump into your code editor and create a new file called app.py and import Flask and render_template to render HTML onto the webpage.

```
from flask import Flask, render_template
# Now use Flask to create an endpoint and
# render a template there.
app = Flask(__name__)
@app.route('/')
def home():
    return render_template('index.html')
if __name__ == '__main__':
    app.run(debug=True)
```

that's it our app.py is done.

Now open our index.html and type some content that you want to display onto the webpage.


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
    <title>First Flask App</title>
  </head>
  <body>
    <h1>First Flask Web App</h1>
  </body>
</html>
```

Now we are almost done. Just open your terminal and type

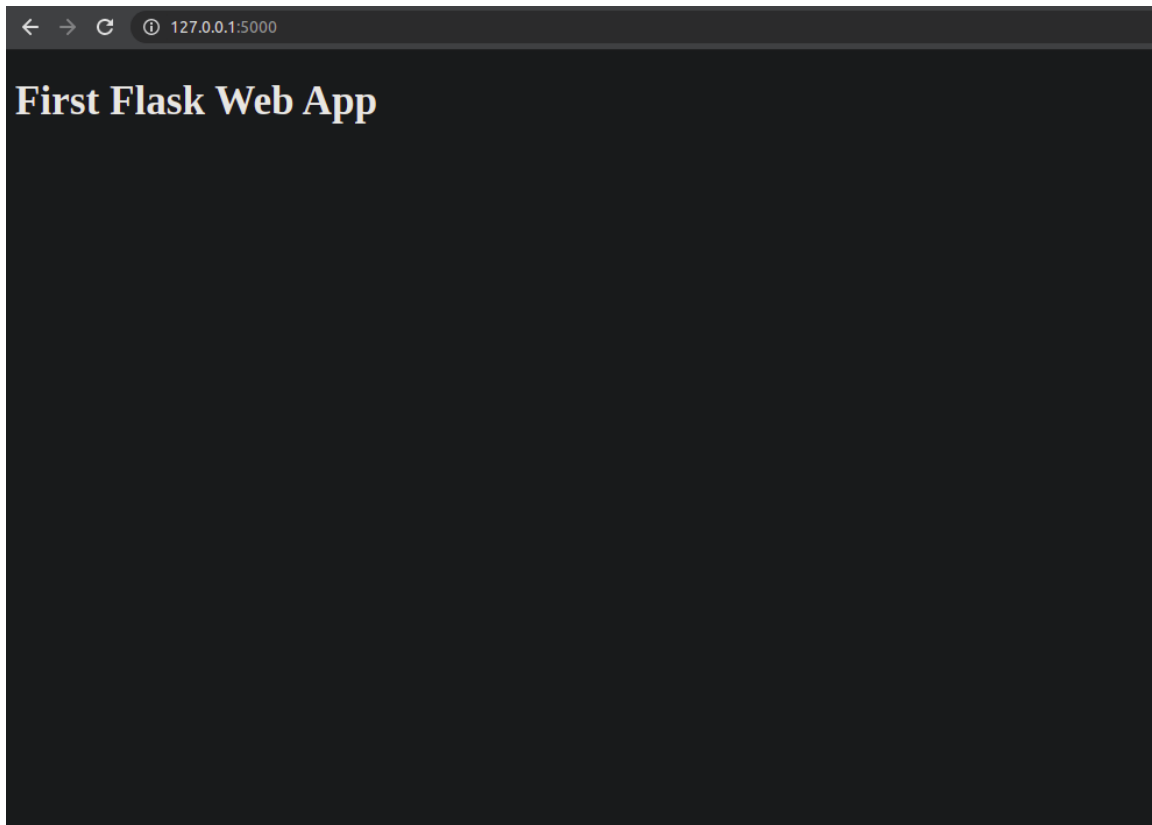
- `python app.py`

This should run the flask app and you should see something like this.

```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 321-259-564
```

Now open your localhost browser that is `http://localhost:5000/` to see your flask web app render

your HTML there. You should see an output similar to this.



That's it we installed Flask and implemented a very simple Flask application. Now it is your time to create some cool web apps and dynamic content and amaze everyone. If you are an AI enthusiast then feel free to deploy your model as a Flask application and share it with others. For a simple demonstration of how to machine learning models using Flask check out this [GitHub repository](#).

Conclusion

In this article, we have seen Flask and its importance in today's world. We even saw some of the major companies that use Flask. A simple implementation on how to install Flask and use it is demonstrated.

Flask opens a huge set of possibilities and if you want to create awesome applications quickly, Flask is your go-to. They are perfect to deploy your AI models. Explore more about Flask. I am sure you will be amazed at how simple yet powerful it is. I hope this article was useful to you all. Will see you in my next article until then, as always code learn repeat ☕

Prepared By [Dev Mehta](#)
Founder and CEO at [ZeroDay](#)