



Efficient Way Of Web Development Using Python And Flask

Fankar Armash Aslam
Department Of Computer Engineering
Anjuman-I-Islam's Kalsekar Technical Campus
Panvel, India

Hawa Nabeel Mohammed
Jummal Musab Mohd. Munir
Murade Aaraf Gulamgaus
Department Of Computer Engineering
Anjuman-I-Islam's Kalsekar Technical Campus Panvel, India

Prof. P. S. Lokhande
Assistant Professor Department of Computer Engineering
Anjuman-I-Islam's Kalsekar Technical Campus Panvel, India

Abstract: Web is the most frequently used networking aid which satisfies the requirements of all types of users; it provides a solution for any type of problem definition. While developing a web portal the appearance of web portal makes a development more critical. The good appearance of a web can easily attract more number of visitors which is a success of web portal. For designing and developing such well structured and with the good appearance of web we have to choose a proper technology. The technological needs of satisfying a good web portal can be fulfilled by "python" and "flask".

Keywords: Python, Flask, Website, Web Portal, Web Development, Jinja.

I. INTRODUCTION

A. How Python and Flask Supports Web Development Efficiently:

Python is a general purpose, high level programming language that focuses on the code readability, for web development lines of code will be fewer than other languages. It is possible for Python because of large standard libraries which make the web development code simple and short. These libraries have pre-coded functions provided by Python community which can be easily downloaded and can be used as per the development needs. Initially Python was designed for web servers to deal with the incoming traffic on the server.

Flask is a micro framework of Python which provides the basic functionality of web framework and allows more plug-ins to be added so the functionality and feature set can be extended to a new level. Flask is called as micro framework of Python because it makes the core functionality simple but extensible in terms of development.

It can also be used to save time building web applications.

Flask uses Jinja Template Engine and the Werkzeug WSGI Toolkit. Flask structure is categories into two parts "Static files & Template files", template file have all the Jinja templates including Html pages, where as static file have all static codes needed for website such as CSS code, JavaScript code and Image files.[1][2]

B. Technological Advantages:

a. Extensible

Extensibility in web development is a principle rule designed as a system's ability to have new functionality extended, in which the system's internal structure and data flow are minimally or not affected, particularly that recompiling or changing the original source code is unnecessary when changing a system's behavior, Because systems are long lived and will be modified for new features and added functionalities demanded by users.

b. Robust:

Robustness is the ability of system to cope with errors during execution. Robustness is also used as the ability of an algorithm to continue operating despite abnormalities in input, calculations, etc. Robustness can encompass many areas of web development.

c. Open Source:

Python & Flask are an open source languages in which the source code is available to the general public for use and/or modification from its original design. Open-source code is typically a collaborative effort where other developers can improve upon the source code and share the changes within the community so that other members can help improve it further.

II. TECHNICAL REVIEW

A. Introduction to Python:



Figure.1 Python Logo[1]

There are no type declaration of variables, parameters, functions, or methods in source code. This makes the code short and flexible, and you lose the compiler-time type checking of the source code. Python tracks the types of all values at run time and flags code that does not make sense as it runs.

Python is powerful and fast, plays well with others, runs everywhere is friendly and easy to learn. Python source files use the ".py" extension and are called "modules." [1]

a) Features of Python:

- (a). Small Core
- (b). Clear, Concise, and Orthogonal Syntax.
- (c). Self Documenting
- (d). Easy supports for default arguments
- (e). True object oriented and 'First Class' classes and functions
- (f). Classes are used extensively in the standard library
- (g). Multiple Inheritance
- (h). Object-Oriented file handling
- (i). Method Chaining
- (j). Everything is a reference
- (k). 'Del' statement for all data types
- (l). Simple array slicing syntax
- (m). Consistent case sensitivity
- (n). Operator overloading
- (o). Structured exception handling
- (p). Threading
- (q). Lots of high level data types
- (r). Maturity, stability, and upward compatibility
- (s). UNICODE support
- (t). GUI frameworks
- (u). Tends to lead to much scalable applications.

Other features of Python...

- (a). Python code has to be strictly indented but the indentation helps in much cleaner code
- (b). High level data structures are list, dictionary are well suited in python.
- (c). Easy to write, easy to read and easy to understand.

b) Advantages:

- (a). Python developer offer far better service than PHP
- (b). The syntax of python are simple as compared to PHP.
- (c). In PHP, you required to have several tools; were as in python, you don't really need to have many tool with you. You can do well just the debugging tools available in the downloader package of python.

c) Disadvantages:

- (a). Python isn't the best for memory intensive tasks.
- (b). Python not a great choice for a high-graphic 3d game that takes up a lot of CPU.

B. Introduction to Flask:



Figure.2 Flask Logo[2]

Flask is a lightweight web application framework written in python and based on the WSGI toolkit and jinja2 template engine. Flask takes the flexible python programming language and provides a simple template for web development. Once imported into python, Flask can be used to save time building web applications. It keeps the core simple but extensible. It has no database abstraction layer, form validation, or any other components. Flask supports extensions. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and more. [2]

a) Features of Flask:

- (a). Integrated supports for unit testing
- (b). Uses Jinja2 templating
- (c). Support for secure cookies
- (d). Extensive documentation
- (e). Google app engine compatibility
- (f). Restful request dispatching
- (g). Unicode based

C. Introduction to Jinja2:



Figure.3 Jinja2 Logo[6]

Jinja2 is a library for python that is designed to be flexible, fast and secure. Jinja2 is a modern and designer-friendly templating language for python, modeled after Django's templates. It is fast, widely used and secure with the optional sandboxed template execution environment. Jinja2 is more readable because its syntax is easy to visually distinguish from HTML code. [6]

a) Features of Jinja2:

- (a). Sandbox execution
- (b). Template inheritance
- (c). Compiles down to the optimal
- (d). Python code just in time
- (e). Easy to debug

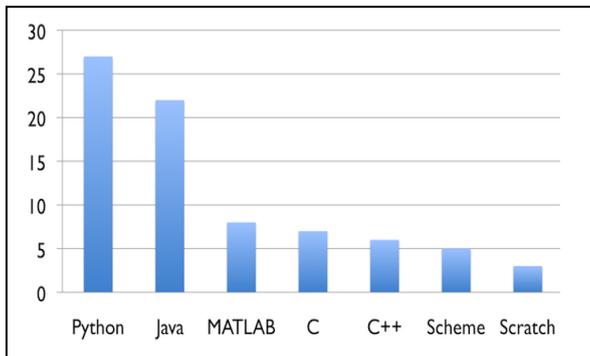


Figure.4 Statistics of Python [7]

III. IMPLEMENTATION

A. The Support From Creation Of Decorative Template by using Flask:

a) Hello World Program [3]:

Hello World program in Flask is a basic example of Flask where we are importing Flask class using import function, then we are defining hello world function and then returning 'Hello World!'.

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello World!'
if __name__ == '__main__':
    app.run()
```

b) Template Inheritance [4]:

Template Inheritance in Flask permits developers to build a base template which can be overridden by child template. It uses `{% block %}` and it tells base template that child template may override the functionality defined in `{% block %}`. `{% extends %}` section defines that this template can "extend" another template. When the template system evaluates this template, first it locates the parent. The extends tag must be the first tag in the template. To render the contents of a block defined in the parent template, use `{{ super() }}`.

```
index.html:

{% extends "layout.html" %}
{% block content %}
<h1>Say something</h1>
<form method="post" action="{{ url_for('signup') }}">
<p><label>Username:</label>
<input type="text" name="username" required></p>
<p><label>Message:</label>
<textarea name="message"></textarea></p>
<p><button type="submit">Send
</button></p>
</form>
{% endblock %}
```

```
message.html:
{% extends "layout.html" %}
{% block content %}
<h1>{{ username }} said:</h1>
<p>
{{ message }}
</p>
<a href="{{ url_for('home') }}">Say something else</a>
{% endblock %}
```

```
layout.html:
<!doctype html>
<html lang="en">
<head>
<title>Say something</title>
<meta http-equiv="content-type" content="text/html;
charset=utf-8">
<link rel="shortcut icon" href="{{ url_for('static',
filename='favicon.ico') }}">
</head>
<body>
{% block content %}{% endblock %}
</body>
</html>
```

c) File Organization[5]:

File Organization is used to define the way in which the files are arranged in the directory and it shows the way in which they can be called. Example "/" is used to define the index page.

```
example/
__init__.py
static/
favicon.ico
templates/
index.html
hello.html
```

B. How Python Will Support To Design A Dynamic Web Portal:

a) Login Code Using Python [1]:

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form.get('username', None)
        password = request.form.get('password', None)
        profile = Profile.authenticate(email, password)
        if profile and profile.id:
            set_session_and_login(profile)
            flash('Successfully logged in.', category='success')
            return redirect('/stream/me')
        return render_template('/generic/main/login.html',
            menu=MenuView(None))
    def set_session_and_login(profile):
        session['user'] = str(profile.id)
        event=LoginEvent(user=session['user'],
            url=request.url,
            ip_address=request.remote_addr)
        event.save()
    @app.route('/logout', methods=['GET', 'POST'])
    @login_required
    def logout():
        if hasattr(g, 'user'):
            g.user = None
            session.clear()
            return redirect(url_for('home'))
```

b) Updating Profile Code Using Python[1]:

```
class ProfileEdit(Form):
    id = HiddenField()
    name = TextField('Name', validators=[Required()])
    email=TextField('Email address', validators =
        [Required('Please provide a valid email address'),
        Length(min=6, message=(u'Email address too short')),
        Email(message=(u'That\'s not a valid email address.))])
    username = TextField('Username', validators=[Required()])
    phone = TextField('Phone', validators=[])
    address = TextAreaField('Address', validators=[])
    is_social_login = BooleanField('Is Social Login',
        default = False)
    facebook = TextField('Facebook Link', validators=[])
    linkedin = TextField('LinkedIN Link', validators=[])
    website = TextField('Website Link', validators=[])
    text = TextAreaField("About Me")
```

IV. FUTURE SCOPE

Great thing about Python and Flask is that it is portable and interactive language for web development including speedy prototyping and dynamic semantics potential. In python it is also possible to bind new modules to Python to extend its core functionality. Many of great websites are moving to Python because its robustness. Python provides easy way to use standard libraries. Python works with almost all databases, powerful text processing and document processing facilities, and plays well with other web technologies.

V. CONCLUSION

As we all know that web development is a complex process of structuring content with dynamic data transactions. For maintaining such complexity technologies such as python Jinja Flask are more useful. Such technologies are also helps to create more user friendly interface for data fetch from WWW.

This paper can be conclude as Python can be used for making web more powerful, fast and efficient with the help of Flask Template Engine.

VI. REFERENCES

- [1]. The Python Software Foundation is a non-profit corporation, "The Python Tutorial", <https://docs.python.org/3/tutorial/index.html>, March 16th 2015.
- [2]. By Armin Ronacher, "Flask Web Development, One Drop At A Time", <http://flask.pocoo.org/>, March 20th 2015.
- [3]. By Armin Ronacher, "The Templates", <http://flask.pocoo.org/docs/0.10/tutorial/templates/>, March 20th 2015
- [4]. By Armin Ronacher, "The Templates", <http://flask.pocoo.org/docs/0.10/tutorial/templates/>, March 20th 2015
- [5]. By Armin Ronacher, "Larger Applications", <http://flask.pocoo.org/docs/0.10/patterns/packages/>, March 20th 2015
- [6]. By Armin Ronacher, "Jinja is Beautiful", <http://jinja.pocoo.org/>, March 21th 2015
- [7]. By Philip Guo, <http://www.pgbovine.net/>, July 2014