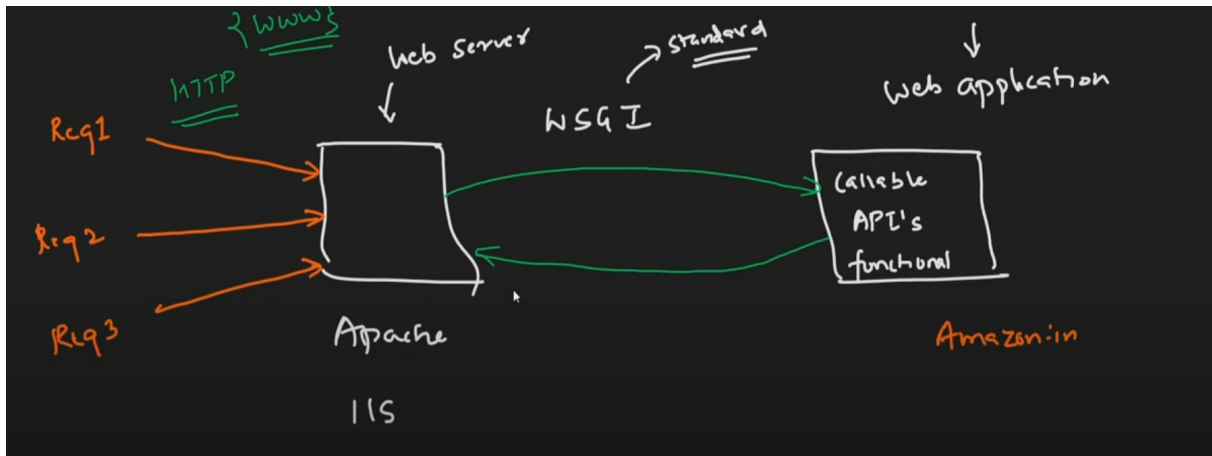


## #Introduction

- Flask is a very important web application framework written in python. You can use it to develop web applications , web api's , M.L applications and other end-to-end project.
- **Armin Ronacher developed it.**
- It is based on WSGI and Jinja2 template Engine concept.
- WSGI stands for web server Gateway Interface . It is a standard or protocol . Whenever you host an application you host it on a web server such as apache , The webserver requires standards or protocol in order to communicate with web application and for this we use WSGI.



- Jinja 2 template engine is a popular template engine for python . It basically combines a web template along with certain datasource to render dynamic web-pages. For example creating a web template for detecting cat or dog face and then combining it with a model such as pickle or sql or any other datasource.

## #Setting up Environment in VScode

- go to the project directory and open terminal inside vscode.
- Set up virtual environment

```
python -m venv myenv
```

- Activate Virtual Environment

```
myenv\Scripts\activate
```

- Install flask

```
pip install Flask
```

- create simple `app.py` file

```

from flask import Flask
# WSGI APPLICATION create an instance of the Flask application.
app = Flask(__name__)
#DECORATOR defines a route for your web application.
@app.route('/')
def home():
    return "Hello, Flask!"

if __name__ == '__main__':
    app.run(debug=True)

#If Debug is not set to true you need to restart entire application
#after any changes

```

- To run the file type "python app.py"
- The line `if __name__ == '__main__':` is used in Python to check if a script is being run directly or being imported as a module in another script. `__name__` : This is a special built-in variable in Python. It represents the name of the current module. If the module is being run directly, `__name__` is set to `'__main__'`.

## # Building URL Dynamically using variable rules and URL Building

```

from flask import Flask, redirect, url_for

app = Flask(__name__)

@app.route('/')
def home():
    return "Hello, Flask!"

@app.route('/success/<int:score>')
def success(score):
    return "<html><body><h1>The person is passed , score and marks is </h1></body></html>"+

@app.route('/fail/<int:score>')
def fail(score):
    return "The person score and marks is "+str(score)

@app.route('/results/<int:marks>')
def results(marks):
    result = ""
    if marks<50:
        result='fail'
    else:
        result="success"
    return redirect(url_for(result,score=marks)) #displaying different pages

```

```
if __name__ == '__main__':  
    app.run(debug=True)
```

## # Integrating HTML and handling "get" and "post"

```
from flask import Flask, redirect, url_for, render_template, request  
  
app = Flask(__name__)  
  
@app.route('/')  
def home():  
    return render_template('index.html')  
  
@app.route('/success/<int:score>')  
def success(score):  
    res=""  
    if score >=50:  
        res="PASS"  
    else:  
        res="FAIL"  
    return render_template('result.html', result=res)  
  
@app.route('/fail/<int:score>')  
def fail(score):  
    return "The person has failed , score and marks is "+str(score)  
  
# ///////////////////////////////////  
@app.route('/results/<int:marks>')  
def results(marks):  
    result =""  
    if marks<50:  
        result='fail'  
    else:  
        result="success"  
    return redirect(url_for(result, score=marks)) #displaying different pages  
  
@app.route('/submit', methods=['POST', 'GET'])  
def submit():  
    total_score=0  
    if request.method=='POST':  
        science=float(request.form['science'])  
        maths=float(request.form['math'])  
        english=float(request.form['english'])  
        history=float(request.form['history'])  
        total_score = (science+maths+english+history)/4  
    res=""
```

```

    if total_score >= 50:
        res = "success"
    else:
        res = "fail"
    return redirect(url_for(res, score=total_score))

if __name__ == '__main__':
    app.run(debug=True)

```

```

<!DOCTYPE html>
<html>
  <body>
    <h2>HTML Forms!</h2>

    <form action="/submit" method="post">
      <label for="Science">Science:</label><br />
      <input type="text" id="science" name="science" value="0" /><br />
      <label for="Math">Math:</label><br />
      <input type="text" id="math" name="math" value="0" /><br />
      <label for="English">English:</label><br />
      <input type="text" id="english" name="english" value="0" /><br />
      <label for="History">History:</label><br />
      <input type="text" id="history" name="history" value="0" /><br />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>

```

```

<html>
  <body>
    <h2>Final Results</h2>
    The output is {{result}}
  </body>
</html>

```

## # Jinja2 Template Engine

- Integrating HTML with some data source.

```

'''
{%...%} for statements
{{  }} expressions to print output
{#...#} this is for comments
'''

```

```
{% if result>=50 %}
<h1>Result: Pass</h1>
{% else %}
<h1>Result: Failed</h1>
{% endif %}
```

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

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/success/<int:score>')
def success(score):
    res=""
    if score>=50:
        res="PASS"
    else:
        res="FAIL"
    exp={'score':score,'res':res}
    return render_template('result.html',result=exp)

@app.route('/fail/<int:score>')
def fail(score):
    return "The person has failed , score and marks is "+str(score)

# //////////////////////////////////////
@app.route('/results/<int:marks>')
def results(marks):
    result =""
    if marks<50:
        result='fail'
    else:
        result="success"
    return redirect(url_for(result,score=marks)) #displaying different pages

@app.route('/submit',methods=['POST','GET'])
def submit():
    total_score=0
    if request.method=='POST':
        science=float(request.form['science'])
        maths=float(request.form['math'])
        english=float(request.form['english'])
        history=float(request.form['history'])
        total_score = (science+maths+english+history)/4
    res=""
    if total_score>=50:
```

```

        res="success"
    else:
        res="fail"
    return redirect(url_for(res,score=total_score))

if __name__ == '__main__':
    app.run(debug=True)

```

```

<html>
<h2>Final Results</h2>
<body>
<table border="2">
    {% for key,value in result.items() %}
    <tr>
        {# This a the comment sections #}
        <th>{{key}}</th>
        <th>{{value}}</th>
    </tr>
    {% endfor %}
</table>
</body>
</html>

```

## # Integrating JS and CSS

```

<link
    rel="stylesheet"
    href="{{ url_for('static', filename='css/style.css')}}"
/>
<script
    type="text/javascript"
    src="{{url_for('static', filename='script/script.js')}}"
></script>

```