

1. Template = HTML with some python code embedded
2. Python code embedding is done by a template generator = **JINJA**---Python through html generator
3. Template Generator === Template Engine
<https://www.fullstackpython.com/template-engines.html>

Context folder = Project folder

templates - folder
*.html

server.py

Requirement -

Simple HTML flow

1. GET that shows a login page ---login (submit)
2. On click of Submit --- POST request ---- verify login and return Welcome Username on success or return the Login page with error message

JINJA uses following template output syntax

{{}} =====> put a variable or a func call or an expression that evaluates to a single value
This output is added to HTML

{% -----python like code to generate HTML ----- %}

Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h5>{{error}}</h5>
  <form action="verify" method="post">
    <p>Uname :<input type="text" name="uname" />
    <p>Pass :<input type="text" name="pass" />
    <p><input type="submit" value="login" />
  </form>
</body>
</html>
```

First.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <p>Hello from Python {{username}}</p>
</body>
</html>

```

Server.py

```

from flask import Flask,request,render_template
app=Flask(__name__)
@app.get("/")
def first():
    #return "<h1>this is the test</h1>"
    return render_template("first.html")
@app.get("/login")
def login():
    return render_template("login.html")

@app.post("/verify")
def abcd():
    formdata = request.form
    user= formdata.get("uname")
    password= formdata.get("pass")
    if(user=="prachi" and password=="123"):
        return render_template("first.html",username=user)
    else:
        return render_template("login.html",error="Invalid Login ,Try Again...")

if __name__ == "__main__":
    app.run()

```
