

Full Stack Experiment 7

Name: Abhay Mathur

SAPID: 60017210016

Batch: AIML A1

Theory:

Abhay Mathur
60017210016

Full Stack Experiment - 7

Ans: Using templates in Flask. Use Jinja 2 templates to create dynamic content in the Flask application.

Theory: Templating with Jinja 2 in Flask. Create another template called 'welcome.html' inside the template folder. Now, we need to a way to actually inherit some templates instead of reusing them. We can do that by creating the block in Jinja. They allow us to create a template block and we can use them in other templates with the name given to the block.

So let us reuse one ~~index.html~~ ~~index.html~~ and create a block in there.

To do that we use '`{% block name %}`' where name = "body". To start the block thus will take everything above it and store it in a virtual block of template, to end the block you simply use '`{% endblock %}`'. This will copy everything below it.

This looks like a two-liner but will also extend the index.html.

~~This is~~ This is by using the '`{% extend "file.html" %}`' tags. They parse the work into the mentioned template.

After this, we ~~can~~ can add the things we want.

If you use the include tags it will not put the replacement program in the correct place or index.html page. It will create an invalid HTML file, but some of the browsers are very forgiving you will not continue unless you look at the source generated. The Body must be properly nested.

Finally, the piece left here is the ~~route~~ route to home.html, so just create that as well.

Jinja in the conditions are started with a curly bracket and a % symbol like `{% if condition %}` you can add `{% elif %}`

For loop
`{% for i in numbers %}`
`{% end for %}`
if condition
`{% if i % 2 == 0 %}`
`{% end if %}`

Conclusion: Therefore templates in flask were created using Jinja2 templater to create dynamic content in the flask app.

Code:

Server.py:

```
from flask import Flask
app = Flask(__name__)
print("Initialized the app! With a name of: ", __name__)
```

```

@app.route('/string/<variable>')
def string(variable):
    return f'''
    <h1>Hi {variable}!</h1>
    <h1>Hi %s!</h1>
    ''' % variable

@app.route('/int/<int:id>')
def int(id):
    return f'''
    <h1>This POST request has the id: {id}!</h1>
    <h1>This POST request has the id: %d!</h1>
    ''' % id

@app.route('/float/<float:balance>')
def float(balance):
    return f'''
    <h1>This POST request has the id: {balance}!</h1>
    <h1>This POST request has the id: %f!</h1>
    ''' % balance

def extra(username):
    return f'''
    <h1>Hi {username}!</h1>
    This is to show how to add a url route to a function using the
    add_url_rule method.
    <br><br>
    This approach is mainly used in case we are importing the view function
    from another module.
    '''

app.add_url_rule('/extra/<username>', 'extra', extra) #'extra' is the name
assigned to the url route '/extra/<username>' and is used to call the view
function extra

from flask import render_template

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

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

@app.route('/home2')
def home2():
    return render_template('home2.html')

```

```

@app.route('/about')
def about():
    sites =
['https://www.google.com', 'https://www.youtube.com', 'https://www.facebook.com'
]
    return render_template('about.html', sites=sites)

@app.route('/contact/<role>')
def contact(role):
    return render_template('contact.html', person=role)

if __name__ == '__main__':
    print("App is starting...I think?")
    app.run(debug=True, use_reloader=True) #this will autoreload the server on
reloading website when you start app using flask run. IF you run each
individual .py file with python only then in that case the server
automatically reloads every time a change is made, no need for using auto
reloader
    print("Umm...no idea what/how this message will be seen")

```

welcome.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>FlaskTest</title>
</head>
<body>
    <h1>Flask Rendering Templates</h1>
    <a href="{{ url_for('index') }}">Index</a>
    {% block body %}
    <a href="{{ url_for('home2') }}">Home2</a>
    <p>This is a flask application</p>
    {% endblock %}
</body>
</html>

```

index.html:

```

<!DOCTYPE html>
<html>

<head>
    <title>FlaskTest</title>
</head>

```

```

<body>
  <h2>Welcome To GFG</h2>
  <h4>Flask: Rendering Templates</h4>
  <a href="{{ url_for('welcome') }}">Welcome</a>
</body>

</html>

```

home2.html:

```

{% extends 'welcome.html' %}
{% block body %}
<h4>Flask Rendering Template Extended From <a href="{{ url_for('welcome') }}">'welcome.html'</a></h4>
<p>This is a flask application</p>
{% endblock %}

```

contact.html:

```

{% extends 'welcome.html' %}
{% block body %}
  {% if person == 'admin' %}
  <p>Admin Section</p>
  {% elif person == 'maintainer' %}
  <p>App Source for page maintainer</p>
  {% elif person == 'member' %}
  <p>Hope you are enjoying our service</p>
  {% else %}
  <p>Hello {{ person }} </p>
  {% endif %}
{% endblock %}

```

answer.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Answer Page!</title>
</head>
<body>
  <h1>Keep Learning Maths!</h1>
  <h2>Square of number {{num}} is :{{squareofnum}}</h2>
</body>
</html>

```

about.html:

```
{% extends 'welcome.html' %}
{% block body %}
<ul>
    {% for site in sites %}
    <a href="{{ site }}">{{ site }}</a>
    <br>
    {% endfor %}
</ul>
{% endblock %}
```

squarenum.py:

```
from flask import Flask, render_template, request

app = Flask(__name__)

@app.route('/sqGET', methods=['GET'])
def squarenumberGET(): #Changing the url and then getting the data out of that url
    # If method is GET, check if number is entered or user has just requested the page.
    # Calculate the square of number and pass it to answermaths method
    if request.method == 'GET':
        # If 'num' is None, the user has requested page the first time
        if(request.args.get('num') == None):
            return render_template('squarenumGET.html')
        # If user clicks on Submit button without entering number, display error
        elif(request.args.get('num') == ''):
            return "<html><body> <h1>Invalid number</h1></body></html>"
        else:
            # User has entered a number
            # Fetch the number from args attribute of request accessing its 'id'
            from HTML
            number = (request.args.get('num'))

            print(number)
            sq = int(number) * int(number)

            # pass the result to the answer HTML
            # page using Jinja2 template
            return render_template('answer.html',
                                   squareofnum=sq, num=number)

@app.route('/sqPOST', methods=['GET', 'POST'])
```

```

def squarenumberPOST():#No change in url, data is gotten from from POST
request sent
    # If method is POST, get the number entered by user
    # Calculate the square of number and pass it to answermaths
    if request.method == 'POST':
        if(request.form['num'] == ''):
            return "<html><body> <h1>Invalid number</h1></body></html>"
        else:
            number = request.form['num']
            sq = int(number) * int(number)
            return render_template('answer.html',
                                   squareofnum=sq, num=number)
    # If the method is GET,render the HTML page to the user
    if request.method == 'GET':
        return render_template("squarenumPOST.html")

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

```

squarenumGET.html:

```

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

<head>
    <meta charset="UTF-8">
    <title>Square Of Number!</title>
</head>

<body>
    <h1><i> Welcome to the Maths page!</i></h1>
    <p>Logic shapes every choice of our daily lives.<br>
        Logical thinking enables someone to learn and
        make decisions that affect their way of life. !</p>
    <form method="GET" action="/sqGET">
        Enter a number (GET) :
        <input type="text" name="num" id="num">
        <input type="submit" name="btnnum" id="btnnum">
    </form>
</body>

</html>

```

squarenumPOST.html:

```

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

```

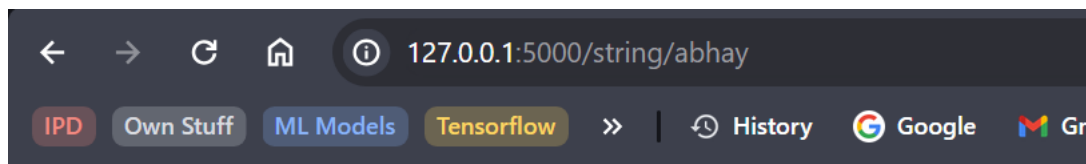
```
<head>
  <meta charset="UTF-8">
  <title>Square Of Number!</title>
</head>

<body>
  <h1><i> Welcome to the Maths page!</i></h1>
  <p>Logic shapes every choice of our daily lives.<br>
    Logical thinking enables someone to learn and
    make decisions that affect their way of life. !</p>
  <form method="POST" action="/sqPOST">
    Enter a number (POST) :
    <input type="text" name="num" id="num">
    <input type="submit" name="btnnum" id="btnnum">
  </form>

</body>

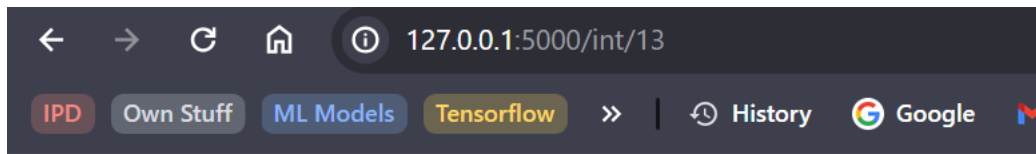
</html>
```

Output:



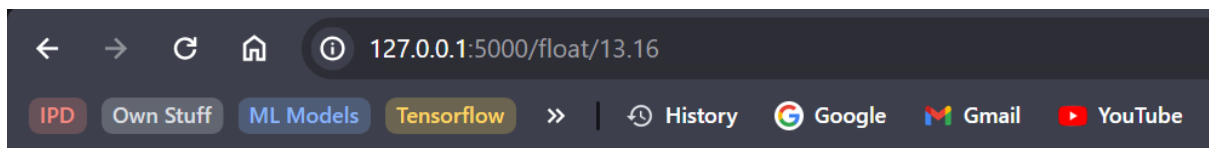
Hi abhay!

Hi abhay!



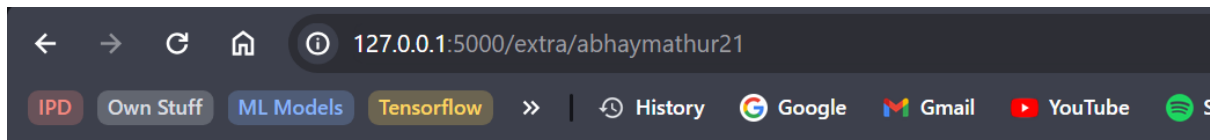
This POST request has the id: 13!

This POST request has the id: 13!



This POST request has the id: 13.16!

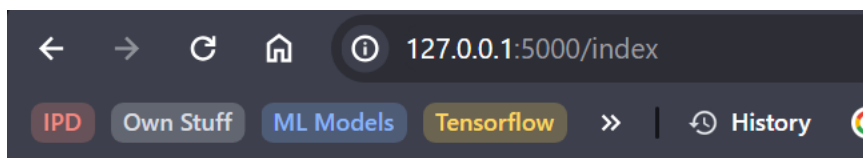
This POST request has the id: 13.160000!



Hi abhaymathur21!

This is to show how to add a url route to a function using the `add_url_rule` method.

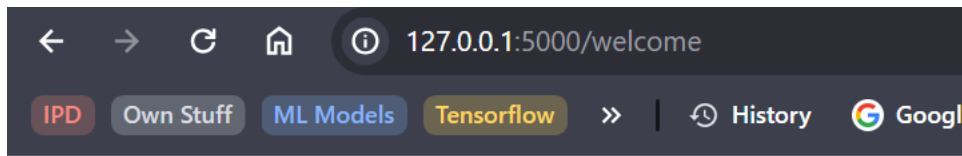
This approach is mainly used in case we are importing the view function from another module.



Welcome To GFG

Flask: Rendering Templates

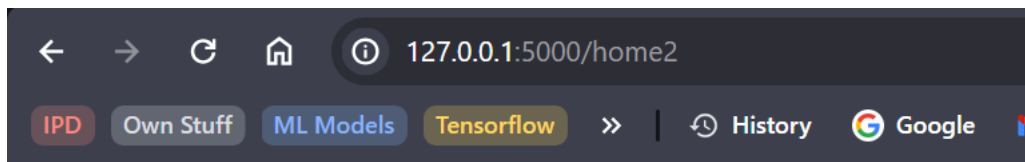
[Welcome](#)



Flask Rendering Templates

[Index](#) [Home2](#)

This is a flask application

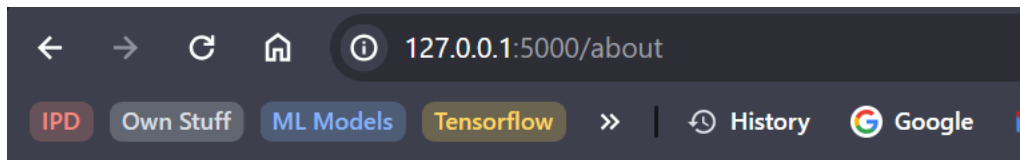


Flask Rendering Templates

[Index](#)

Flask Rendering Template Extended From ['welcome.html'](#)

This is a flask application



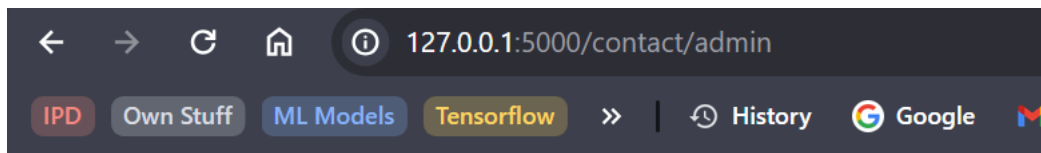
Flask Rendering Templates

[Index](#)

<https://www.google.com>

<https://www.youtube.com>

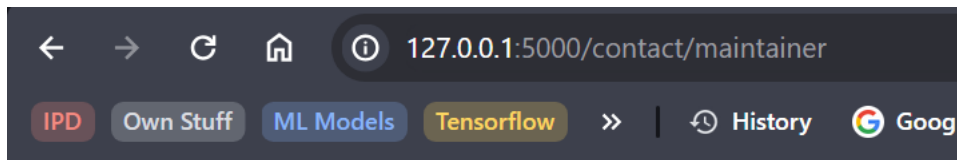
<https://www.facebook.com>



Flask Rendering Templates

[Index](#)

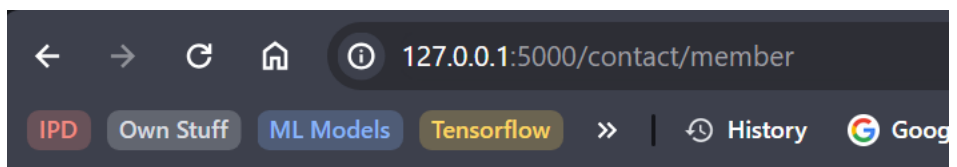
Admin Section



Flask Rendering Templates

[Index](#)

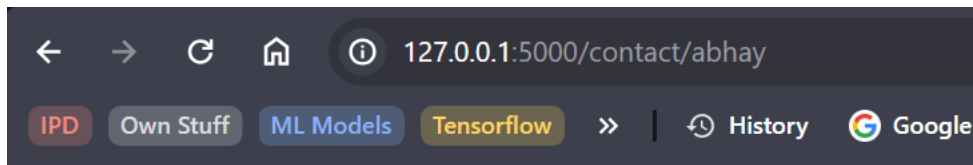
App Source for page maintainer



Flask Rendering Templates

[Index](#)

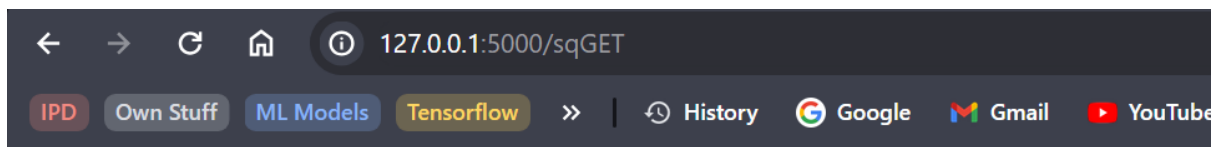
Hope you are enjoying our service



Flask Rendering Templates

[Index](#)

Hello abhay

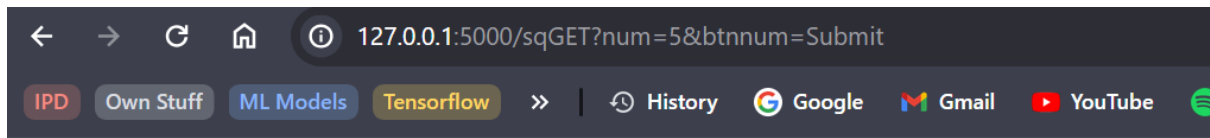


Welcome to the Maths page!

Logic shapes every choice of our daily lives.

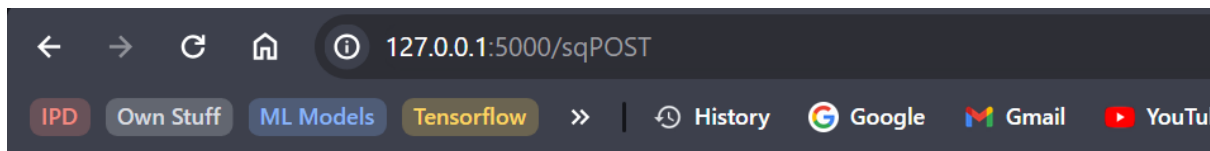
Logical thinking enables someone to learn and make decisions that affect their way of life. !

Enter a nurnber (GET) :



Keep Learning Maths!

Square of number 5 is :25

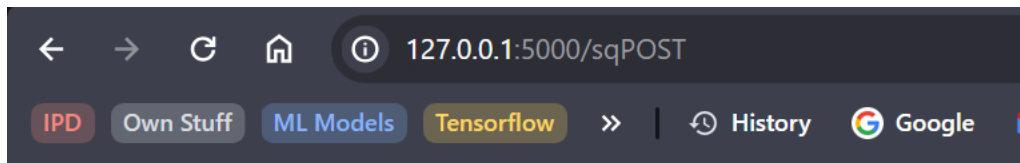


Welcome to the Maths page!

Logic shapes every choice of our daily lives.

Logical thinking enables someone to learn and make decisions that affect their way of life. !

Enter a number (POST) :



Keep Learning Maths!

Square of number 5 is :25

Conclusion: Therefore templates in flask were created using Jinja2 templating syntax to create dynamic context in the flask application.