Flask (Framework for creating Websites in Python Handbook

Raman Deep Singh

Flask is a framework to create websites in Python.

We have PyCharm for this Flask Tutorial.

In Flask we place all html files which are called as templates in templates folder.

We place css files, image files and java script files in static folder.

Use the following command to run your flask project.

flask run

After successful run you can open your flask application by typing following url in browser

http://127.0.0.1:5000/

Following program will print Hello World on the output screen

use following command to run this app.py file

flask run

---

code for app.py is

```
from flask import Flask

app = Flask(__name__)



@app.route('/')
def hello_world():  # put application's code here
    return 'Hello World!'



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

Output

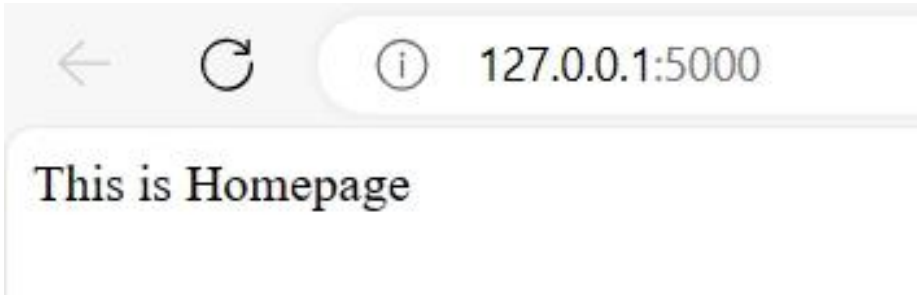
---

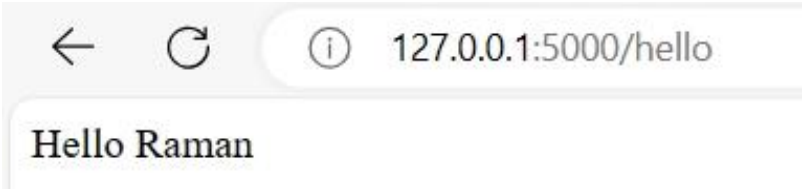
example of app routing in flask

app.py

```
from flask import Flask

app = Flask(__name__)


# Pass the required route to the decorator.
@app.route("/hello")
def hello():
    return "Hello Raman"


@app.route("/")
def index():
    return "This is Homepage"


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

Output





---

another example of app routing

example to demonstrate the dynamic URLs

```
from flask import Flask

app = Flask(__name__)


@app.route('/user/<username>')
def show_user(username):
    # Greet the user
    return f'Hello {username} !'


# Pass the required route to the decorator.
@app.route("/hello")
def hello():
    return "Hello Raman"


@app.route("/")
def index():
    return "Homepage"


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

Output

---

example to demonstrate the converter type

string: It is the default type and it accepts any text without a slash.

int: It accepts positive integers.

float: It accepts positive floating-point values.

path: It is like a string but also accepts slashes.

uuid: It accepts UUID strings.

app.py

```
from flask import Flask

app = Flask(__name__)


@app.route('/post/<int:id>')
def show_post(id):
    # Shows the post with given id.
    return f'This post has the id {id}'


@app.route('/user/<username>')
def show_user(username):
    # Greet the user
    return f'Hello {username} !'


# Pass the required route to the decorator.
@app.route("/hello")
def hello():
    return "Hello, Welcome to cppcourses"


@app.route("/")
def index():
    return "Homepage of cppcourses"


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

Output

The add_url_rule() function – The URL mapping can also be done using the add_url_rule() function. This approach is mainly used in case we are importing the view function from another module. In fact, the app.route calls this function internally.

use the following url to try the following code

http://127.0.0.1:5000/user/raman

app.py

```
from flask import Flask

app = Flask(__name__)


def show_user(username):
    # Greet the user
    return f'Hello {username} !'


app.add_url_rule('/user/<username>', 'show_user', show_user)

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

Output

---

example of url building

app.py

```
#The function accepts the name of a function as first argument, and one or
more keyword arguments, each corresponding to the variable part of URL.

#The following script demonstrates use of url_for() function.

from flask import Flask, redirect, url_for
app = Flask(__name__)

@app.route('/admin')
def hello_admin():
   return 'Hello Admin'

@app.route('/guest/<guest>')
def hello_guest(guest):
   return 'Hello %s as Guest' % guest

@app.route('/user/<name>')
def hello_user(name):
   if name =='admin':
      return redirect(url_for('hello_admin'))
   else:
      return redirect(url_for('hello_guest',guest = name))

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

Output

Following is an example of Flask HTTP Methods

create a index.html file in templates folder

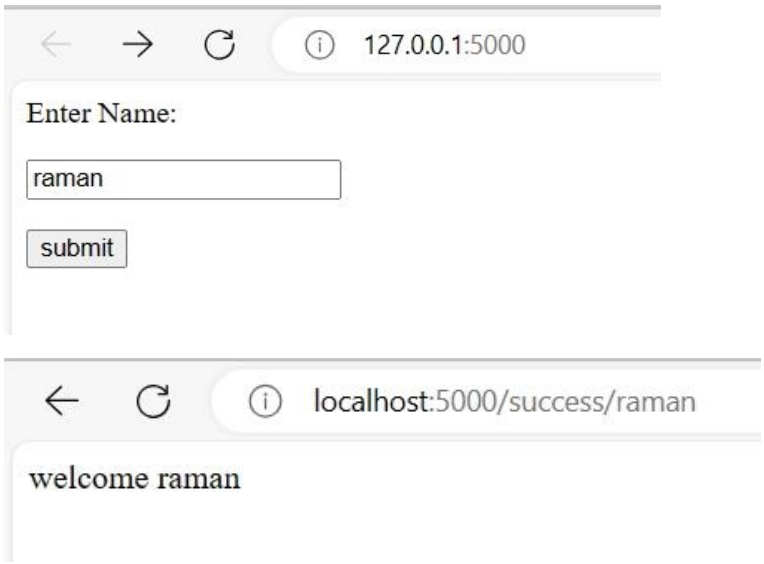
code for index.html is

```
<html>
   <body>
      <form action = "http://localhost:5000/login" method = "post">
         <p>Enter Name:</p>
         <p><input type = "text" name = "nm" /></p>
         <p><input type = "submit" value = "submit" /></p>
      </form>
   </body>
</html>
```

code for app.py

```
from flask import Flask, redirect, url_for, request,render_template
app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')
@app.route('/success/<name>')
def success(name):
   return 'welcome %s' % name

@app.route('/login',methods = ['POST', 'GET'])
def login():
   if request.method == 'POST':
      user = request.form['nm']
      return redirect(url_for('success',name = user))
   else:
      user = request.args.get('nm')
      return redirect(url_for('success',name = user))



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

Output



Example of Flask Templates

create a hello.html file in templates folder

code for hello.html is

```
<!doctype html>
<html>
   <body>

      <h1>Hello {{ name }}!</h1>

   </body>
</html>
```

Code for app.py is

```
from flask import Flask, render_template
app = Flask(__name__)

@app.route('/hello/<user>')
def hello_name(user):
   return render_template('hello.html', name = user)

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

Output

Example of Static files

static files like javascript (.js) files, images are placed in static folder

create file hello.js in static folder

code for hello.js

```
function sayHello() {
   alert("Hello World")
}
```

Create a file sayhello.html in templates folder

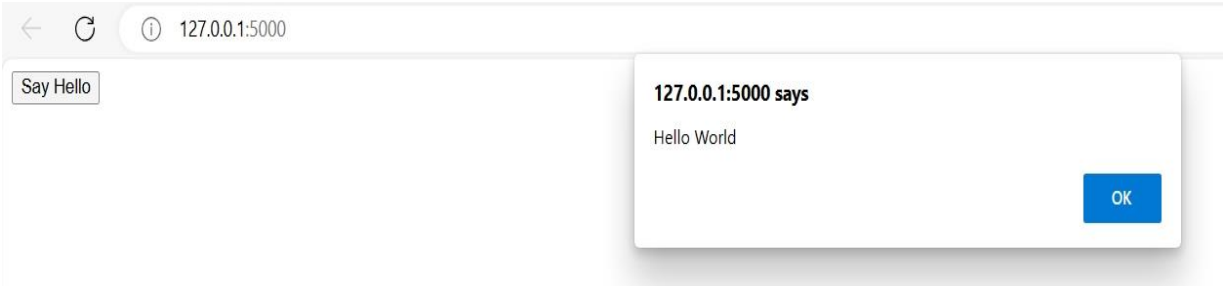
code for sayhello.html

```
<html>
   <head>
      <script type = "text/javascript"
         src = "{{ url_for('static', filename = 'hello.js') }}" ></script>
   </head>

   <body>
      <input type = "button" onclick = "sayHello()" value = "Say Hello" />
   </body>
</html>
```

Code for app.py

```
from flask import Flask, render_template
app = Flask(__name__)

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

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

Output



Example of sending Form Data to Template

create a file student.html in templates folder

code for student.html is

```
<html>
   <body>
      <form action = "http://localhost:5000/result" method = "POST">
         <p>Name <input type = "text" name = "Name" /></p>
         <p>Physics <input type = "text" name = "Physics" /></p>
         <p>Chemistry <input type = "text" name = "chemistry" /></p>
         <p>Maths <input type ="text" name = "Mathematics" /></p>
         <p><input type = "submit" value = "submit" /></p>
      </form>
   </body>
</html>
```

create a file result.html in templates folder

code for result.html is

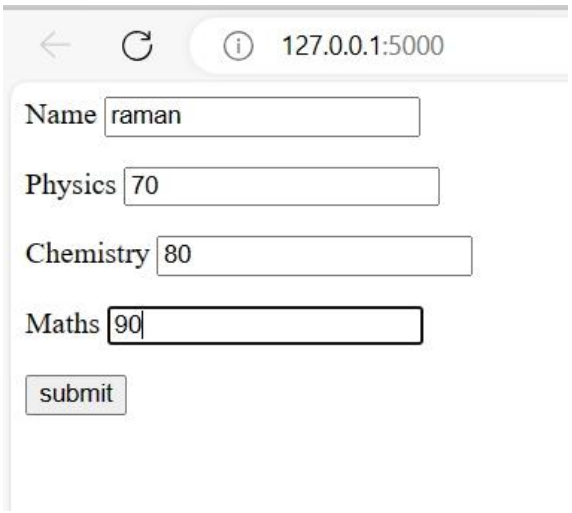
```
<!doctype html>
<html>
   <body>
      <table border = 1>
         {% for key, value in result.items() %}
            <tr>
               <th> {{ key }} </th>
               <td> {{ value }} </td>
            </tr>
         {% endfor %}
      </table>
   </body>
</html>
```

Code for app.py is

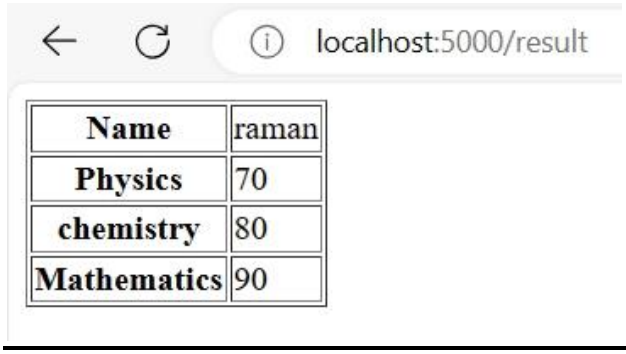
```
from flask import Flask, render_template, request
app = Flask(__name__)

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

@app.route('/result',methods = ['POST', 'GET'])
def result():
   if request.method == 'POST':
      result = request.form
```

```
        return render_template("result.html",result = result)

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

Output



Upon clicking on button Submit Output is



---

example of cookies in flask

create a html file in templates folder with name setcookie.html

Code for setcookie.html

```
<html>
   <body>
      <form action = "/setcookie" method = "POST">
         <p><h3>Enter userID</h3></p>
         <p><input type = 'text' name = 'nm'/></p>
         <p><input type = 'submit' value = 'Login'/></p>
      </form>
   </body>
</html>
```

create readcookie.html in templates folder

code for readcookie.html is

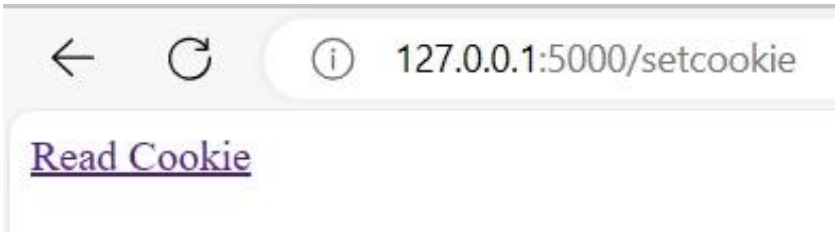
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<a href="/getcookie">Read Cookie</a>
</body>
</html>
```

Code for app.py is

```
from flask import Flask, render_template, request,make_response
app = Flask(__name__)

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


@app.route('/setcookie', methods=['POST', 'GET'])
def setcookie():
    if request.method == 'POST':
        user = request.form['nm']

    resp = make_response(render_template('readcookie.html'))
    resp.set_cookie('userID', user)

    return resp

@app.route('/getcookie')
def getcookie():
   name = request.cookies.get('userID')
   return '<h1>welcome '+name+'</h1>'
```

Output

Click on Login Button and you will see following output



Click on Read Cookie link and you will see following output

Flask Redirects and Errors

create a file log_in.html in templates folder

code for log_in.html is

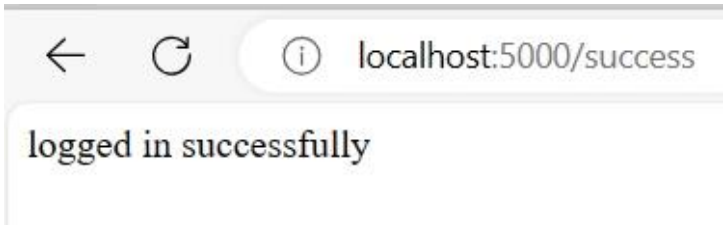
```
<html>
   <body>
      <form action = "http://localhost:5000/login" method = "post">
         <p>Enter Name:</p>
         <p><input type = "text" name = "nm" /></p>
         <p><input type = "submit" value = "submit" /></p>
      </form>
   </body>
</html>
```

Code for app.py

```
from flask import Flask, redirect, url_for, render_template, request, abort
app = Flask(__name__)

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

@app.route('/login',methods = ['POST', 'GET'])
def login():
   if request.method == 'POST':
      if request.form['nm'] == 'admin' :
         return redirect(url_for('success'))
      else:
         abort(401)
   else:
      return redirect(url_for('index'))

@app.route('/success')
def success():
   return 'logged in successfully'

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

Output

Output on clicking on Submit Button



Example of File Uploading in Flask

create a file fileupload.html in templates folder

```
<html>
   <body>
      <form action = "http://localhost:5000/uploader" method = "POST"
         enctype = "multipart/form-data">
         <input type = "file" name = "file" />
         <input type = "submit"/>
      </form>
   </body>
</html>
```

Code for app.py

```
from flask import Flask, render_template, request

app = Flask(__name__)


@app.route('/upload')
def upload_file1():
    return render_template('fileupload.html')


@app.route('/uploader', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        f = request.files['file']
        f.save(f.filename)
        return 'file uploaded successfully'
```

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

Output

Upon clicking on Submit Button