Jamia Millia Islamia

Dept. of Computer Science

Subject: Pattern Matching Using Python
Assignment Topic: Sessions & Cookies in Flask

Submitted By: Wasit Shafi
Roll No: 18MCA054
Date: 18-OCT-2020

Submitted To: Prof.Usman Malik

Q. Create two basic programs that captures the functionality of sessions and cookies.

Sol.

Programm 1.1(Cookies)

```
from flask import Flask
from flask import request
from flask import make_response

# To check presence of cookie in chrome : inspect => Application
app = Flask(__name__)
@app.route('/')
def index():
  return 'Welcome to homepage...!!!' + '<br/> /create_cookie<br> /read_cookie
<br> /delete_cookie'

@app.route('/create_cookie')
def createcookie():
  cok = make_response("<b1>Setting a cookie...</b>")
  name = 'cok1'
  value = '18MCA054'
  max_age = 60 * 60 * 24 # 24-Hours
  #max_age = None # if cookie is not set, then it will expire one the browser is
closed
  cok.set_cookie(name, value, max_age)  # setCookie(<title>, <content>, <expiry
time>)
  return cok

@app.route('/delete_cookie')
def deletecookie():
  cok = make_response("<b1>Deleteing a cookie...</b>")
  name = 'cok1'
  value = '18MCA054'
  max_age = 0 # it will delete the cookie
  cok.set_cookie(name, value, max_age)  # setCookie(<title>, <content>, <expiry
time>)
  return cok

@app.route('/read_cookie')
def readcookie():
  name = 'cok1'
  value = request.cookies.get(name)
  output = 'Reading Cookie Content <br/><br/>' + 'Content of <b>' + str(name) +
'</b> is : <b>' + str(value) + '</b>'
  return output

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

Programm 1.2(Cookies)

```
from flask import Flask
from flask import request
from flask import make_response
from flask import render_template

# To check presence of cookie in chrome : inspect => Application
app = Flask(__name__)
```

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

@app.route('/create_cookie/', methods = ['GET', 'POST'])
def create_cookie():
  cok = make_response("<b1>Setting a Cookie...</b>")
  name = ''
  value = ''
  max_age = 0

  if request.method == 'GET': # then we have to take data from url string
    name = request.args.get('txtb_id')
    value = request.args.get('txtb_content')
    max_age = int(request.args.get('txtb_age'))
  else:
    name = request.form['txtb_id']
    value = request.form['txtb_content']
    max_age = int(request.form['txtb_age'])

  cok.set_cookie(name, value, max_age)
  return cok # returning response

@app.route('/read_cookie/', methods=['GET', 'POST'])
def readcookie():
  name = '102' # assumed
  value = request.cookies.get(name)
  output = 'Reading Cookie Content <br/><br/>' + 'Content of <b>' + str(name) +
'</b> is : <b>' + str(value) + '</b>'
  return output

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

Programm 2(Sessions)

```
from flask import Flask
from flask import request
from flask import make_response
from flask import session
from flask import render_template
from flask import redirect
from flask import session
from flask import url_for


app = Flask(__name__)
app.secret_key = 'ThisIsMySecreteKeyString'

@app.route('/')
def index():
  name = 'wasit'
  if name in session: # check if session contains name as key
    return redirect(url_for('homepage', username = name))
  return render_template('login.html')

@app.route('/login/', methods=['GET', 'POST'])
def login():
  name = ''
```

```
  password = ''

  if request.method == 'GET':
    name = request.args.get('txtb_username')
    password = request.args.get('txtb_password')
  else:
    name = request.form['txtb_username'] # from is a dictionary consisting
key,value pair
    password = request.form['txtb_password']
  session[name] = password
  return redirect(url_for('homepage', username = name))

@app.route('/homepage/<username>')
def homepage(username):
  return 'Welcome ' + username + '<br> Your password was : ' +
session[username];

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