Flask Cookies

The cookies are stored in the form of text files on the client's machine. Cookies are used to track the user's activities on the web and reflect some suggestions according to the user's choices to enhance the user's experience.

Cookies are set by the server on the client's machine which will be associated with the client's request to that particular server in all future transactions until the lifetime of the cookie expires or it is deleted by the specific web page on the server.

In flask, the cookies are associated with the Request object as the dictionary object of all the cookie variables and their values transmitted by the client. Flask facilitates us to specify the expiry time, path, and the domain name of the website.

```
response.setCookie(<title>, <content>, <expiry time>)
```

In Flask, the cookies are set on the response object by using the set_cookie() method on the response object. The response object can be formed by using the make_response() method in the view function.

In addition, we can read the cookies stored on the client's machine using the get() method of the cookies attribute associated with the Request object.

```
request.cookies.get(<title>)
```

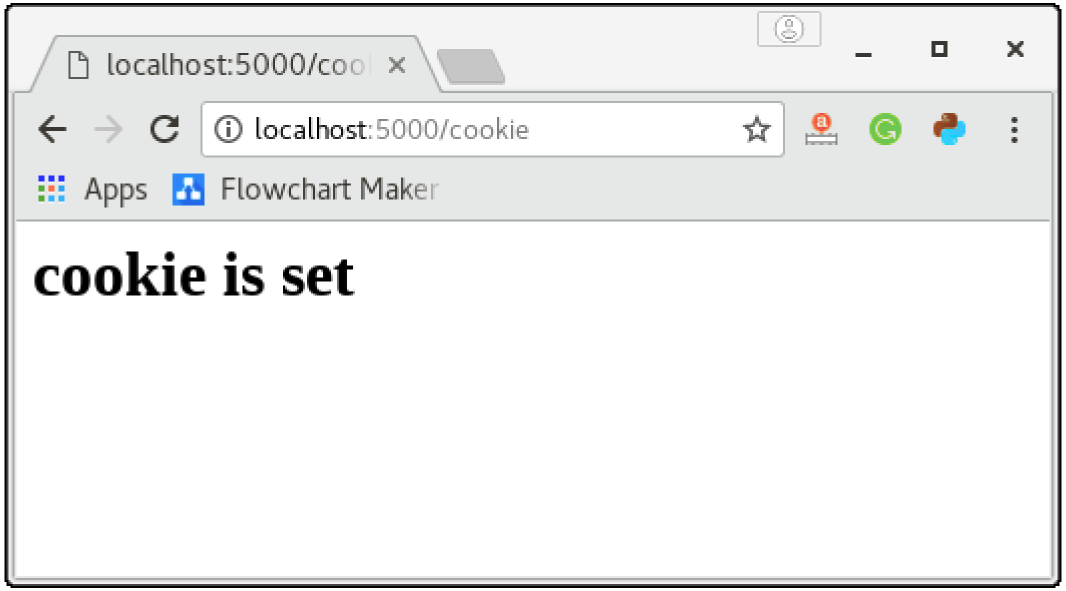
A simple example to set a cookie with the title 'foo' and the content 'bar' is given below.

Example

```
from flask import *

app = Flask(__name__)

@app.route('/cookie')
def cookie():
        res = make_response("<h1>cookie is set</h1>")
        res.set_cookie('foo','bar')
        return res

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

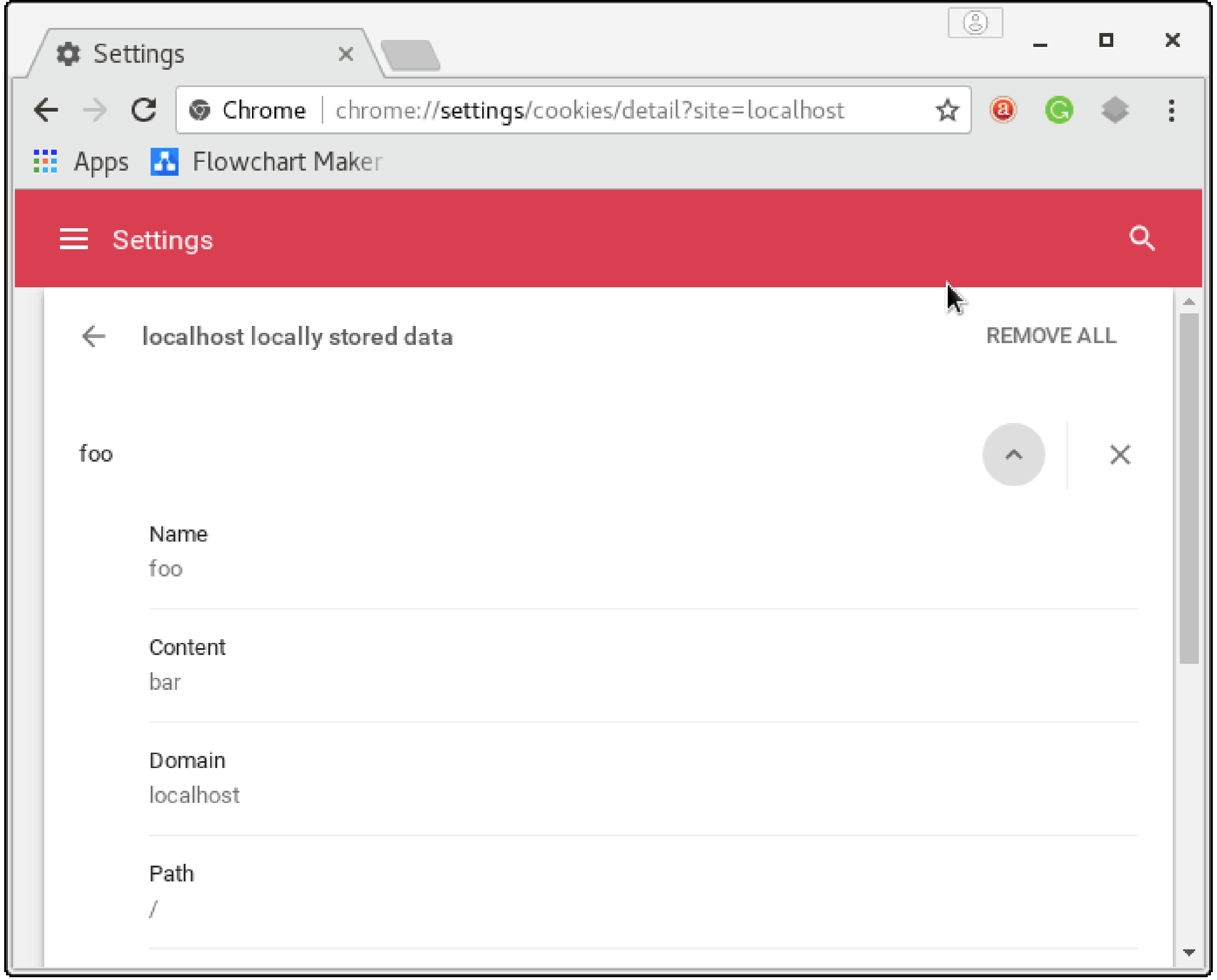
The above python script can be used to set the cookie with the name 'foo' and the content 'bar' on the browser for the website **localhost:5000**.

Run this python script using the command python script.py and check the result on the browser.



cookie is set

We can track the cookie details in the content settings of the browser as given in below image.



localhost locally stored data

foo

Name
foo

Content
bar

Domain
localhost

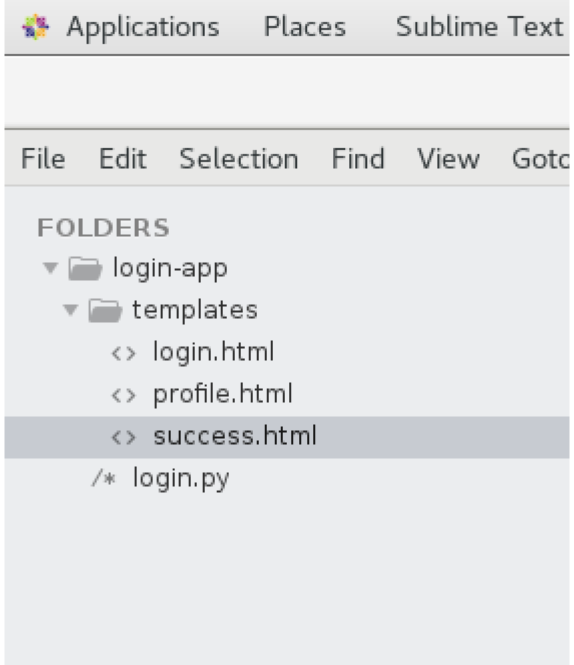
Path
/

Login application in Flask

Here, we will create a login application in the flask where a login page (login.html) is shown to the user which prompts to enter the email and password. If the password is "jtp", then the application will redirect the user to the success page (success.html) where the message and a link to the profile (profile.html) is given otherwise it will redirect the user to the error page.

The controller python flask script (login.py) controls the behaviour of the application. It contains the view functions for the various cases. The email of the user is stored on the browser in the form of the cookie. If the password entered by the user is "jtp", then the application stores the email id of the user on the browser as the cookie which is later read in the profile page to show some message to the user.

The application contains the following python and HTML script. The directory structure of the application is given below.



```
FOLDERS
▼ login-app
  ▼ templates
      login.html
      profile.html
      success.html
    login.py
```

login.py

```
from flask import *

app = Flask(__name__)

@app.route('/error')
def error():
        return "<p><strong>Enter correct password</strong></p>"

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

@app.route('/success',methods = ['POST'])
def success():
        if request.method == "POST":
                email = request.form['email']
                password = request.form['pass']

        if password=="jtp":
                resp = make_response(render_template('success.html'))
                resp.set_cookie('email',email)
                return resp
        else:
                return redirect(url_for('error'))

@app.route('/viewprofile')
def profile():
        email = request.cookies.get('email')
        resp = make_response(render_template('profile.html',name = email))
        return resp

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

login.html

```
<html>
<head>
        <title>login</title>
</head>
<body>
        <form method = "post" action = "http://localhost:5000/success">
                <table>
                        <tr><td>Email</td><td><input type = 'email' name = 'email'></td></tr>
                        <tr><td>Password</td><td><input type = 'password' name = 'pass'></td></tr>
                    <tr><td><input type = "submit" value = "Submit"></td></tr>
                </table>
        </form>
</body>
</html>
```

success.html

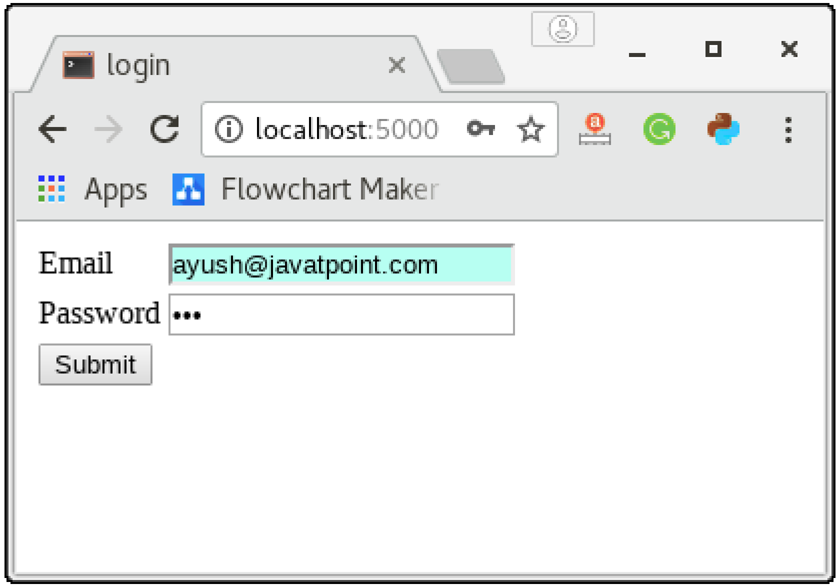
```
<html>
<head>
<title>success</title>
</head>
<body>
        <h2>Login successful</h2>
        <a href="/viewprofile">View Profile</a>
</body>
</html>
```

profile.html

```
<html>
<head>
        <title>profile</title>
</head>
<body>
        <h3>Hi, {{name}}</h3>
</body>
</html>
```

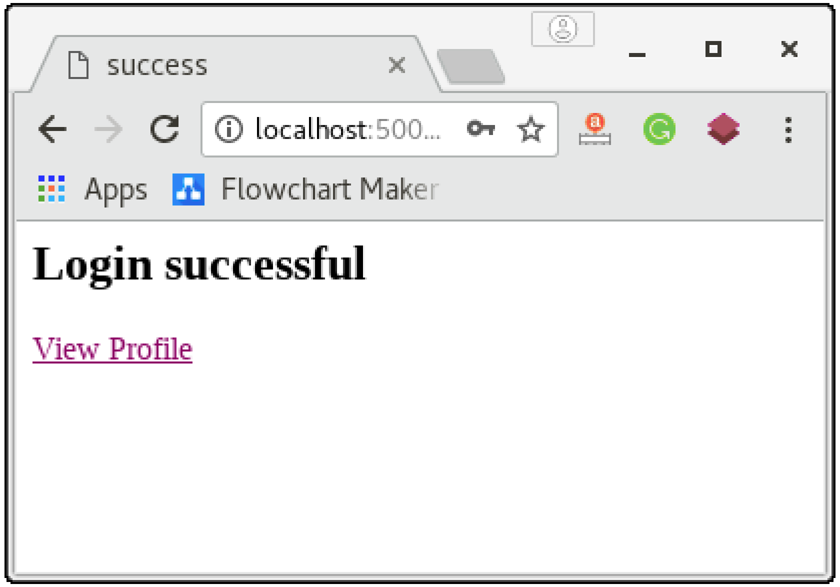
Execution

Run the python script by using the command python login.py and visit localhost:5000/ on the browser as given in the following snapshots.



Email ayush@javatpoint.com
Password •••

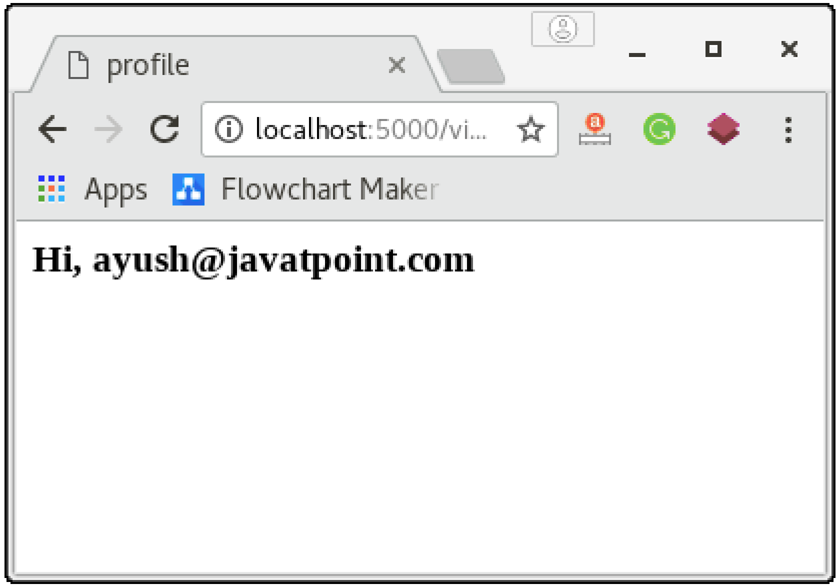
Click Submit. It will show the success message and provide a link to the profile.html.



Login successful

View Profile

Click on view profile. It will read the cookie set as a response from the browser and display the following message.



Hi, ayush@javatpoint.com

← Prev

Next →

Feedback

- Send your Feedback to [email protected]

Help Others, Please Share