

Alec Howard

Kate Sharp

Samuel Lipscomb

Sec Elec Commerce

3/9/2023

Project 2 Report

New Pages for Login Process

1 - Signup

Complete SignupHandler python code that allows the creation of a new account

```
#Create a new class for the signup handler FINISHED
class SignupHandler(BaseHandler):
    def get(self):
        authorized = self.get_cookie("loggedin")
        self.render("signup.html", auth=authorized)

    def post(self):
        username = self.get_argument("username")
        phone_num = self.get_argument("phone_num")
        email = self.get_argument("email")
        password = self.get_argument("password")
        confirm = self.get_argument("confirm")

        #Confirm that the password and confirm password fields match
        if password != confirm:
            self.redirect("/signup")
            return

        #Intermediary step to salt & hash the password
        hashed_passwordb = bcrypt.hashpw(password.encode('utf8'), salt)
        hashed_password = str(hashed_passwordb, 'utf-8')

        #If match, insert into database
        c = self.db.cursor()
        c.execute("INSERT INTO users (username, phone_num, email, password) VALUES (%s, %s, %s, %s)", (username, phone_num, email, hashed_password))
        self.db.commit()
        self.redirect("/login")
        return
```

HTML for the Signup page

```

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Basic Page Needs
  ----- -->
  <meta charset="utf-8">
  <title>Sign Up</title>
  <meta name="description" content="Create an account.">
  <meta name="author" content="Bill Brasky">

  <!-- FONT
  ----- -->
  <link href="//fonts.googleapis.com/css?family=Roboto:400,300,600" rel="stylesheet" type="text/css">
  <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css" rel="stylesheet">

  <!-- CSS
  ----- -->
  <link rel="stylesheet" href="{ { static url("imports.min.css") } }" type="text/css">
  <link rel="stylesheet" href="{ { static url("simpleStore.min.css") } }" type="text/css">

  <!-- Favicon
  ----- -->
  <link rel="icon" type="image/png" href="{ { static url("favicon.png") } }">

  <!-- Scripts
  ----- -->
  <script src="//code.jquery.com/jquery-2.1.4.min.js"></script>

</head>
<body>

  <!-- Primary Page Layout
  ----- -->
  <div class="container simpleStore">
    <div class="row">
      { { block header % } ( % and % ) }
    </div>

    { { block content % } }

    <form action="{ { request.path } }" method="post">
      <div>{ { _("Username") } } <input type="text" name="username"/></div>
      <div>{ { _("Phone Number") } } <input type="text" name="phone_num"/></div>
      <div>{ { _("Email") } } <input type="text" name="email"/></div>
      <div>{ { _("Password") } } <input type="password" name="password"/></div>
      <div>{ { _("Confirm Password") } } <input type="password" name="confirm"/></div>
      <div><input type="submit" value="{ { _("Sign Up") } }" /></div>
      { { module csrf_form_html() % } }
    </form>

    { { and % } }

    <div class="footer">
      { { block Footer % } ( % and % ) }
    </div>
  </div>

  <!-- End Document
  ----- -->
</body>
</html>

```

Webpage

harvey.utulsa.edu Content BB Lecture, Exam, and Group p2_report_sharp_howard Sign Up

10.10.1.13:8095/signup

Username

Phone Number

Email

Password

Confirm Password

Before filling out signup:

```

35 | sam | 1 | 1 | $2b$12$1xo8Fq158e878jggtsm18OGddIA1xiIsEXCVh8Geh5nz8s071z8oK
36 | sam | 1234567890 | sam@sam.com | $2b$12$W4Y2kFA1HUBEi4TnBpGRuKs7yg2k5vAm48ZprnjExqA6Ki3Mx085
37 | sam | 1234567890 | sam@sam.com | $2b$12$NAuf8qJuil5Qri02ybU50ux8LgFc3VfMUmAvwNCHa5P48mW6GXC
38 | sam | 1234567890 | sam@sam.com | $2b$12$.Qh6795ZRPAP8uqlyFLfY0j3ru2Nxx001Q2eaFEY/.H5dydkSKZ86
39 | sam | 1234567890 | sam@sam.com | $2b$12$.Qh6795ZRPAP8uqlyFLfY0j3ru2Nxx001Q2eaFEY/.H5dydkSKZ86
40 | sam | 1234567890 | sam@sam.com | $2b$12$f1wuau6Dz6xPPN6SPNpRK./oPA81buc/67bA81T16RD5TxREsEi36
41 | sam | 1234567890 | sam@sam.com | $2b$12$f8kP11JffXqcmMbPug1e8.ekglZYA8Mty2xu.x0d0sk8PM3b9ZUbe
42 | sam | 1234567890 | sam@sam.com | $2b$12$ndHLrse3o5wY8lwT1iqSe2n6nLQomfAesU.M.vWncjMSUcUQqhm
43 | sam | 1234567890 | sam@sam.com | $2b$12$He5y5j362fMbAQcm3ddJenSd5hv3/tkT0e1XT/7CQ4msrhd.rkqi
44 | joe | 1234567890 | joe@joe.com | $2b$12$He5y5j362fMbAQcm3ddJenXieY17nnbl.855eNqq311c1plq5qv2
45 | troy | 1234567890 | troy@farmersonly.com | troy
46 | troy | 1234567890 | troy@farmersonly.com | $2b$12$He5y5j362fMbAQcm3ddJegqMUiGwScH8c92I3peT0fkYIYHmz2
47 | alec | 1234567890 | alecmhoward@outlook.com | $2b$12$He5y5j362fMbAQcm3ddJekUvcfP27vnmde/uR.Y81Ro3w65exkG
47 rows in set (0.00 sec)

```

After filling out signup:

```

34 | sam | 1 | 1 | $2b$12$M41YkE7X66Lmk/SpGLE..06iR68unynB0DctvGdY3Mc0G8709wGcBa
35 | sam | 1 | 1 | $2b$12$1xo8Fq158e878jggtsm18OGddIA1xiIsEXCVh8Geh5nz8s071z8oK
36 | sam | 1234567890 | sam@sam.com | $2b$12$W4Y2kFA1HUBEi4TnBpGRuKs7yg2k5vAm48ZprnjExqA6Ki3Mx085
37 | sam | 1234567890 | sam@sam.com | $2b$12$NAuf8qJuil5Qri02ybU50ux8LgFc3VfMUmAvwNCHa5P48mW6GXC
38 | sam | 1234567890 | sam@sam.com | $2b$12$.Qh6795ZRPAP8uqlyFLfY0j3ru2Nxx001Q2eaFEY/.H5dydkSKZ86
39 | sam | 1234567890 | sam@sam.com | $2b$12$.Qh6795ZRPAP8uqlyFLfY0j3ru2Nxx001Q2eaFEY/.H5dydkSKZ86
40 | sam | 1234567890 | sam@sam.com | $2b$12$f1wuau6Dz6xPPN6SPNpRK./oPA81buc/67bA81T16RD5TxREsEi36
41 | sam | 1234567890 | sam@sam.com | $2b$12$f8kP11JffXqcmMbPug1e8.ekglZYA8Mty2xu.x0d0sk8PM3b9ZUbe
42 | sam | 1234567890 | sam@sam.com | $2b$12$ndHLrse3o5wY8lwT1iqSe2n6nLQomfAesU.M.vWncjMSUcUQqhm
43 | sam | 1234567890 | sam@sam.com | $2b$12$He5y5j362fMbAQcm3ddJenSd5hv3/tkT0e1XT/7CQ4msrhd.rkqi
44 | joe | 1234567890 | joe@joe.com | $2b$12$He5y5j362fMbAQcm3ddJenXieY17nnbl.855eNqq311c1plq5qv2
45 | troy | 1234567890 | troy@farmersonly.com | troy
46 | troy | 1234567890 | troy@farmersonly.com | $2b$12$He5y5j362fMbAQcm3ddJegqMUiGwScH8c92I3peT0fkYIYHmz2
47 | alec | 9186076401 | alecmhoward@outlook.com | $2b$12$He5y5j362fMbAQcm3ddJekUvcfP27vnmde/uR.Y81Ro3w65exkG
48 | Codi | 1111111111 | codi.west@cool.com | $2b$12$He5y5j362fMbAQcm3ddJes8l0BkF4z7AFuNUdniGkQZj0WQV0vS
48 rows in set (0.00 sec)

mysql>

```

2 - Login

Try testing the signins:

User: alec | Pass: beans

User Codi | Pass: abc123

Python for LoginHandler

```
#Create a new class for the login handler FINISHED
class LoginHandler(BaseHandler):
    def get(self):
        authorized = self.get_cookie("loggedin")
        self.render("login.html", auth=authorized)

    def post(self):
        username = self.get_argument("username")
        password = self.get_argument("password")
        hashed_passwordb = bcrypt.hashpw(password.encode('utf8'), salt)
        hashed_password = str(hashed_passwordb, 'utf-8')

        c = self.db.cursor()
        c.execute("SELECT * FROM users WHERE username = %s AND password = %s", (username, hashed_password))
        user = c.fetchone()
        if user:
            # Set cookie fields for later authentication
            self.set_cookie("loggedin", "true")
            self.set_cookie("username", username)
            self.redirect("/account")
        else:
            self.redirect("/login")
```

HTML

```

<!-- Type HTML -->
<html lang="en">
<head>
({autoescape None %})
    <!-- Basic Page Needs -->
    <meta charset="utf-8">
    <title>Classic Store</title>
    <meta name="description" content="A store for things you never knew you didn't need.">
    <meta name="author" content="Bill Brasky">

    <!-- FONT -->
    <link href="//fonts.googleapis.com/css?family=Roboto:400,300,600" rel="stylesheet" type="text/css">
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css" rel="stylesheet">

    <!-- CSS -->
    <link rel="stylesheet" href="{% static url('imports.min.css') %}" type="text/css">
    <link rel="stylesheet" href="{% static url('simpleStore.min.css') %}" type="text/css">

    <!-- Favicon -->
    <link rel="icon" type="image/png" href="{% static url('favicon.png') %}">

    <!-- Scripts -->
    <script src="//code.jquery.com/jquery-2.1.4.min.js"></script>

</head>
<body>

    <!-- Primary Page Layout -->
    <div class="container simpleStore">
        <div class="row">
            {% block header %}{% end %}
        </div>

        {% block content %}

            <form action="{% request.path %}" method="post">
                <div>{{ _("Username") }} <input type="text" name="username"/></div>
                <div>{{ _("Password") }} <input type="password" name="password"/></div>
                <div><input type="submit" value="{{ _("Sign in") }}" /></div>
                {%- module xsrf_form_html() %}
            </form>

            {%- end %}

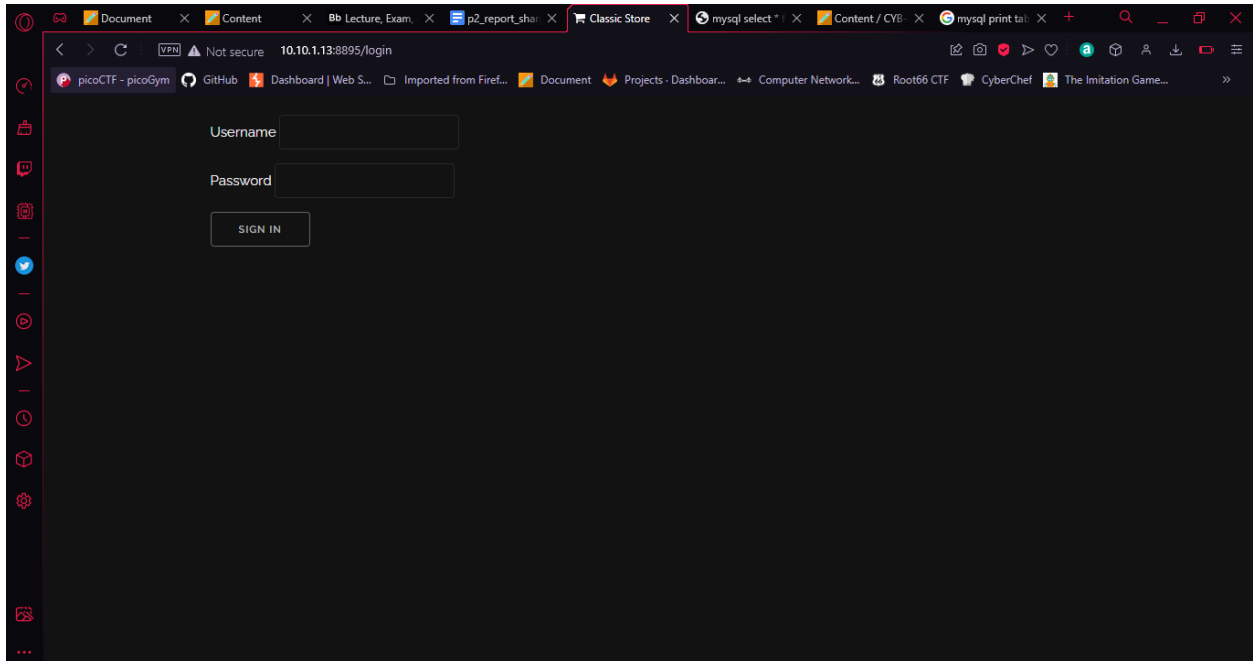
            <div class="footer">
                {%- block footer %}{% end %}
            </div>

        </div>

    <!-- End Document -->
</body>
</html>

```

Webpage



3 - Logout

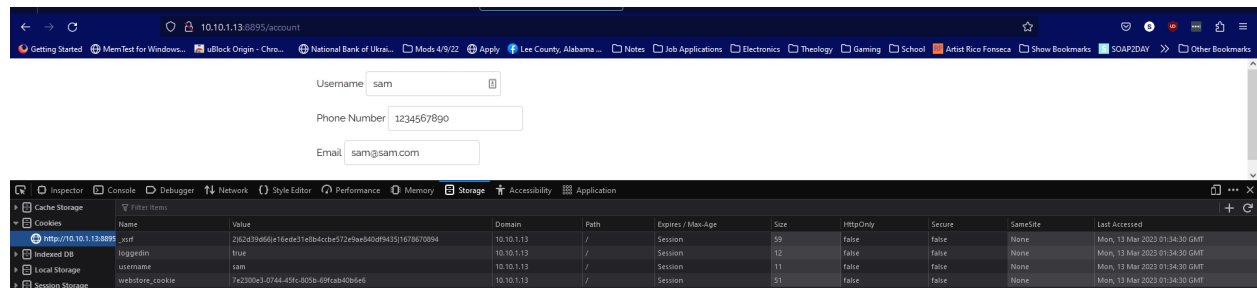
Complete LogoutHandler python code that, well, logs out the user

```
121
122 #Create a new class for the logout handler
123 class LogoutHandler(BaseHandler):
124     def get(self):
125         self.clear_cookie("loggedin")
126         self.clear_cookie("username")
127         self.redirect("/")
128         return
129
```

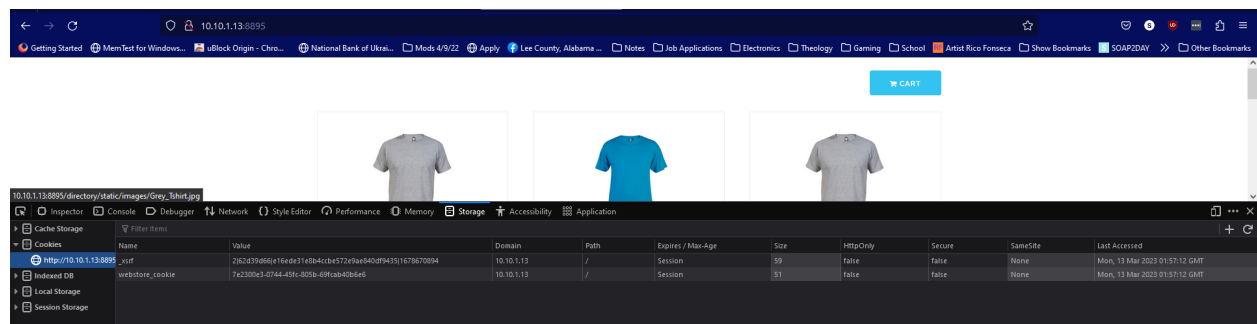
HTML page for logout that acts as a placeholder

```
class_website.py  logout.html X  account.html
project_website > templates > <> logout.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  {%autoescape None %}
5      <!-- Basic Page Needs
6      ----- -->
7      <meta charset="utf-8">
8      <title>Classic Store</title>
9      <meta name="description" content="A store for things you never knew you didn't need.">
10     <meta name="author" content="Bill Brasky">
11
12     <!-- FONT
13     ----- -->
14     <link href="//fonts.googleapis.com/css?family=Raleway:400,300,600" rel="stylesheet" type="text/css">
15     <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css" rel="stylesheet">
16
17     <!-- CSS
18     ----- -->
19     <link rel="stylesheet" href="{{ static_url('imports.min.css') }}" type="text/css">
20     <link rel="stylesheet" href="{{ static_url('simpleStore.min.css') }}" type="text/css">
21
22     <!-- Favicon
23     ----- -->
24     <link rel="icon" type="image/png" href="{{ static_url('favicon.png') }}">
25
26     <!-- Scripts
27     ----- -->
28     <script src="//code.jquery.com/jquery-2.1.4.min.js"></script>
29
30 </head>
31 <body>
32
33     <!-- Primary Page Layout
34     ----- -->
35     <div class="container simpleStore">
36         <div class="row">
37             {% block header %}{% end %}
38         </div>
39
40         {% block content %}
41
42
43
44
45         {% end %}
46
47         <div class="footer">
48             {% block footer %}{% end %}
49         </div>
50     </div>
51
52
53 <!-- End Document
54 ----- -->
55 </body>
56
57 </html>
58
59
```

Before logging out, with cookie still populated with loggedin and username flags



After logging out, with cookie cleared out and user redirected to main page



4 - Account

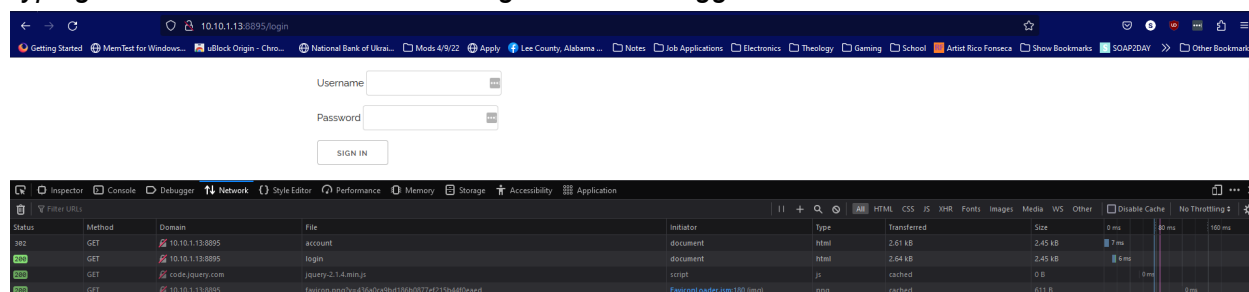
Complete AccountHandler python code that renders the account page

```
159 #Create a new class for the account handler
160 class AccountHandler(BaseHandler):
161     def get(self):
162         authorized = self.get_cookie("loggedin")
163         #Return the user to the login page if they are not logged in
164         if not authorized:
165             self.redirect("/login")
166             return
167         #If logged in, render the account page
168         else:
169             # Grab username from cookie
170             username = self.get_cookie("username")
171
172             # Get the user's username, phone number, email, and password from the database
173             c = self.db.cursor()
174
175             # Grab the username, phone number, email, and password using parameterized query
176             c.execute("SELECT * FROM users WHERE username = %s", (username,))
177             user = c.fetchone()
178
179             # Render the user's data on the page
180             self.render("account.html", user=user)
181
```

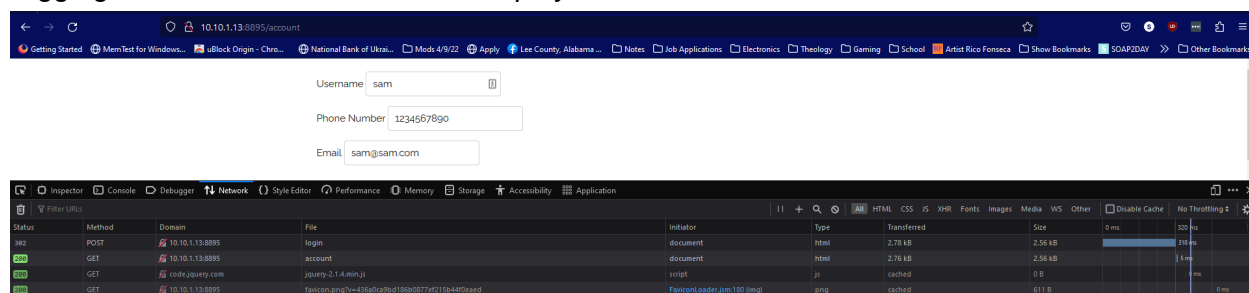

HTML page for account that displays user's username, phone number, and email

```
class_website.py  account.html  signup.html
project_website > templates > > account.html > > html > > body > > div.container.simpleStore > > div.row
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  {%autoescape None %}
5      <!-- Basic Page Needs
6      ----- -->
7      <meta charset="utf-8">
8      <title>Classic Store</title>
9      <meta name="description" content="A store for things you never knew you didn't need.">
10     <meta name="author" content="Bill Brasky">
11
12     <!-- FONT
13     ----- -->
14     <link href="//fonts.googleapis.com/css?family=Raleway:400,300,600" rel="stylesheet" type="text/css">
15     <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css" rel="stylesheet">
16
17     <!-- CSS
18     ----- -->
19     <link rel="stylesheet" href="{{ static_url('imports.min.css') }}" type="text/css">
20     <link rel="stylesheet" href="{{ static_url('simpleStore.min.css') }}" type="text/css">
21
22     <!-- Favicon
23     ----- -->
24     <link rel="icon" type="image/png" href="{{ static_url('favicon.png') }}">
25
26     <!-- Scripts
27     ----- -->
28     <script src="//code.jquery.com/jquery-2.1.4.min.js"></script>
29
30 </head>
31 <body>
32
33     <!-- Primary Page Layout
34     ----- -->
35     <div class="container simpleStore">
36         <div class="row">
37             {% block header %}{% end %}
38         </div>
39
40         {% block content %}
41         <!--Display the user info-->
42         <form action="{{ request.path }}" method="post">
43             <!--Display the user's info-->
44             <div>{{ _("Username") }} <input type="text" name="username" value="{{user[1]}}"/></div>
45             <div>{{ _("Phone Number") }} <input type="text" name="username" value="{{user[2]}}"/></div>
46             <div>{{ _("Email") }} <input type="text" name="username" value="{{user[3]}}"/></div>
47
48             {% module xsrf_form_html() %}
49         </form>
50
51
52         {% end %}
53
54         <div class="footer">
55             {% block footer %}{% end %}
56         </div>
57     </div>
58
59
60 <!-- End Document
61 ----- -->
62 </body>
63
64 </html>
65
66
67
```

Typing /account into url redirects to /login when not logged in



Logging in redirects to /account and displays the user's data



Setting Up SSL/TLS

1 - Create a self-signed certificate

Commands to create the self-signed certificate

```
group8@secomproject:~/project_website/certs$ openssl genrsa 2048 > host.key
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
group8@secomproject:~/project_website/certs$ chmod 400 host.key
group8@secomproject:~/project_website/certs$ openssl req -new -x509 -nodes -sha256 -days 365 -key host.key -out host.cert
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:OK
Locality Name (eg, city) []:Tulsa
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Codi's Class
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:group8
Email Address []:
group8@secomproject:~/project_website/certs$ ls
host.cert host.key
```

2 - Add certificate to Tornado

Add `host.cert` and `host.key` to `class_website.py`

```
def main():  
    http_server = tornado.httpserver.HTTPServer(Application(), ssl_options={"certfile": os.path.join("certs/host.c  
    http_server.listen(options.port)  
    print(f"Web server started. In your browser, go to 10.10.0.13:{options.port}")  
    tornado.ioloop.IOLoop.current().start()
```

Final Result when navigating to `https://10.10.1.13:8895`

