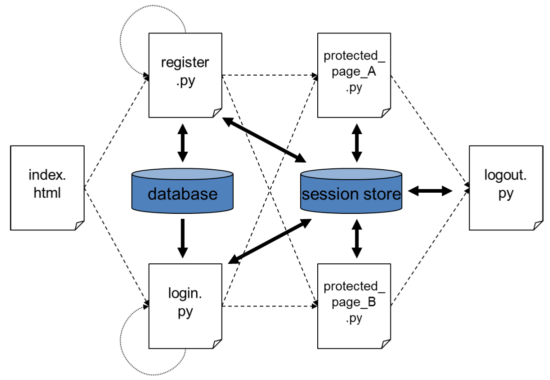
CS1116/CS5018

Web Development 2

Dr Derek Bridge

School of Computer Science & Information Technology
University College Cork

Overview of the programs



MySQL database for user authentication

```
CREATE TABLE users
(
    username VARCHAR(20) NOT NULL,
    password VARCHAR(64) NOT NULL,
    PRIMARY KEY (username)
);
```

Before being stored in the database, passwords will be encrypted using a non-reversible, one-way encryption algorithm (SHA256)

`index.html`

- `index.html` is just a welcome page
- The page displays hyperlinks to `register.py` (for new users) and `login.py` (for existing users)

```
<a href="login.py">Login</a>
<a href="register.py">Register new user</a>
```

register.py: pseudocode

A (partly) sticky, self-processing page

```
result = ''
if the user is sending us data:
    if username blank or passwords blank or passwords not equal:
        result = error message
    else:
        if username already in database:
            result = error message
        else:
            encrypt password
            insert new user details in database
            create a cookie containing a session id
            store data about this user in session store
            result = the 'protected' content
            output the cookie
output Web page containing form and 'result'
```

register.py: the Python

```
#!/usr/local/bin/python3

from cgitb import enable
enable()

from cgi import FieldStorage
from html import escape
from hashlib import sha256
from time import time
from shelve import open
from http.cookies import SimpleCookie
import pymysql as db

form_data = FieldStorage()
username = ''
result = ''
if len(form_data) != 0:
    username = escape(form_data.getfirst('username', '').strip())
    password1 = escape(form_data.getfirst('password1', '').strip())
    password2 = escape(form_data.getfirst('password2', '').strip())
    if not username or not password1 or not password2:
        result = '<p>Error: user name and passwords are required</p>'
    elif password1 != password2:
        result = '<p>Error: passwords must be equal</p>'
    else:
        try:
            connection = db.connect('localhost', 'userid', 'password', 'database_name')
            cursor = connection.cursor(db.cursors.DictCursor)
            cursor.execute("""SELECT * FROM users
                              WHERE username = %s""", (username))
            if cursor.rowcount > 0:
                result = '<p>Error: user name already taken</p>'
            else:
                sha256_password = sha256(password1.encode()).hexdigest()
                cursor.execute("""INSERT INTO users (username, password)
                                  VALUES (%s, %s)""", (username, sha256_password))
                connection.commit()
                cursor.close()
                connection.close()
                cookie = SimpleCookie()
                sid = sha256(repr(time()).encode()).hexdigest()
                cookie['sid'] = sid
                session_store = open('sess_' + sid, writeback=True)
                session_store['authenticated'] = True
                session_store['username'] = username
                session_store.close()
                result = """
                   <p>Succesfully inserted!</p>
                   <p>Thanks for joining Web Dev 2.</p>
                   <ul>
                       <li><a href="protected_page_A.py">Web Dev 2 - Members Only A</a></li>
                       <li><a href="protected_page_B.py">Web Dev 2 - Members Only B</a></li>
                       <li><a href="logout.py">Logout</a></li>
                   </ul>"""
                print(cookie)
        except (db.Error, IOError):
            result = '<p>Sorry! We are experiencing problems at the moment. Please call back later.<

print('Content-Type: text/html')
print()
print("""
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <meta charset="utf-8" />
            <title>Web Dev 2</title>
        </head>
        <body>
            <form action="register.py" method="post">
                <label for="username">User name: </label>
                <input type="text" name="username" id="username" value="%s" />
                <label for="password1">Password: </label>
                <input type="password" name="password1" id="password1" />
                <label for="passwords2">Re-enter password: </label>
                <input type="password" name="password2" id="password2" />
                <input type="submit" value="Register" />
            </form>
            %s
        </body>
    </html>""" % (username, result))
```

`login.py`: pseudocode

Also a (partly) sticky, self-processing page, and very similar to `register.py`

```
result = ''
if the user is sending us data:
    if username blank or password blank:
        resut = error message
    else:
        encrypt password
        search for username and encrypted password in database
        if they aren't there:
            result = error message
        else:
            create a cookie containing a session id
            store data about this user in session store
            result = the 'protected' content
            output the cookie
output Web page containing form and 'result'
```

`login.py`: the Python

```
#!/usr/local/bin/python3

from cgitb import enable
enable()

from cgi import FieldStorage
from html import escape
from hashlib import sha256
from time import time
from shelve import open
from http.cookies import SimpleCookie
import pymysql as db

form_data = FieldStorage()
username = ''
result = ''
if len(form_data) != 0:
    username = escape(form_data.getfirst('username', '').strip())
    password = escape(form_data.getfirst('password', '').strip())
    if not username or not password:
        result = '<p>Error: user name and password are required</p>'
    else:
        sha256_password = sha256(password.encode()).hexdigest()
        try:
            connection = db.connect('localhost', 'userid', 'password', 'database_name')
            cursor = connection.cursor(db.cursors.DictCursor)
            cursor.execute("""SELECT * FROM users
                              WHERE username = %s
                              AND password = %s""", (username, sha256_password))
            if cursor.rowcount == 0:
                result = '<p>Error: incorrect user name or password</p>'
            else:
                cookie = SimpleCookie()
                sid = sha256(repr(time()).encode()).hexdigest()
                cookie['sid'] = sid
                session_store = open('sess_' + sid, writeback=True)
                session_store['authenticated'] = True
                session_store['username'] = username
                session_store.close()
                result = """
                   <p>Succesfully logged in!</p>
                   <p>Welcome back to Web Dev 2.</p>
                   <ul>
                       <li><a href="protected_page_A.py">Web Dev 2 - Members Only A</a></li>
                       <li><a href="protected_page_B.py">Web Dev 2 - Members Only B</a></li>
                       <li><a href="logout.py">Logout</a></li>
                   </ul>"""
                print(cookie)
            cursor.close()
            connection.close()
        except (db.Error, IOError):
            result = '<p>Sorry! We are experiencing problems at the moment. Please call back later.<

print('Content-Type: text/html')
print()
print("""
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <meta charset="utf-8" />
            <title>Web Dev 2</title>
        </head>
        <body>
            <form action="login.py" method="post">
                <label for="username">User name: </label>
                <input type="text" name="username" id="username" value="%s" />
                <label for="password">Password: </label>
                <input type="password" name="password" id="password" />
                <input type="submit" value="Login" />
            </form>
            %s
        </body>
    </html>""" % (username, result))
```

`protected_page_A.py/protected_page_B`: the 'protected' content

- These programs output either a 'permission denied' page or the 'protected' content
- Here the 'protected' content is just an image, but in general it could be something more interesting, e.g. your shopping cart system
- These pages also contain a hyperlink to `logout.py`

`protected_page_A.py`: the 'protected' content

```
#!/usr/local/bin/python3

from cgitb import enable
enable()

from os import environ
from shelve import open
from http.cookies import SimpleCookie

print('Content-Type: text/html')
print()

result = """
   <p>You do not have permission to access this page.</p>
   <ul>
       <li><a href="register.py">Register</a></li>
       <li><a href="login.py">Login</a></li>
   </ul>"""

try:
    cookie = SimpleCookie()
    http_cookie_header = environ.get('HTTP_COOKIE')
    if http_cookie_header:
        cookie.load(http_cookie_header)
        if 'sid' in cookie:
            sid = cookie['sid'].value
            session_store = open('sess_' + sid, writeback=False)
            if session_store.get('authenticated'):
                result = """
                    <p>
                        Hey, %s. We hope you enjoy this photo!
                    </p>
                    <img src="photo1.jpg">
                    <ul>
                        <li><a href="protected_page_A.py">Web Dev 2 - Members Only A</a></li>
                        <li><a href="logout.py">Logout</a></li>
                    </ul>""" % session_store.get('username')
            session_store.close()
except IOError:
    result = '<p>Sorry! We are experiencing problems at the moment. Please call back later.</p>'

print("""
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <meta charset="utf-8" />
            <title>Web Dev 2</title>
        </head>
        <body>
            %s
        </body>
    </html>""" % (result))
```

logout.py

```
#!/usr/local/bin/python3

from cgitb import enable
enable()

from os import environ
from shelve import open
from http.cookies import SimpleCookie

print('Content-Type: text/html')
print()

result = '<p>You are already logged out</p>'
try:
    cookie = SimpleCookie()
    http_cookie_header = environ.get('HTTP_COOKIE')
    if http_cookie_header:
        cookie.load(http_cookie_header)
        if 'sid' in cookie:
            sid = cookie['sid'].value
            session_store = open('sess_' + sid, writeback=True)
            session_store['authenticated'] = False
            session_store.close()
            result = """
                <p>You are now logged out. Thanks for using Web Dev 2</p>
                <p><a href="login.py">Login again</a></p>"""
except IOError:
    result = '<p>Sorry! We are experiencing problems at the moment. Please call back later.</p>'

print("""
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <meta charset="utf-8" />
            <title>Web Dev 2</title>
        </head>
        <body>
            %s
        </body>
    </html>""" % (result))
```