

Практичне заняття №8

з навчальної дисципліни
Спеціалізовані мови програмування

на тему:

РОЗРОБКА WEB-ДОДАТКІВ

**I. За допомогою CGI створити веб-додаток,
який оброблює дані та cookies веб-
клієнта.**

**а) В поточному каталозі створити HTML-документ
під назвою `index.html` з наступним кодом:**

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Обработка данных формы</title>
</head>
<body>
  <form action="/cgi-bin/form.py">
    <input type="text" name="TEXT_1">
    <input type="text" name="TEXT_2">
    <input type="submit">
  </form>
</body>
</html>
```

**b) В каталозі cgi-bin створимо Python-скрипт
для обробки даних форми під назвою form.py:**

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import cgi, unicodedata, html

form = cgi.FieldStorage()
text1 = form.getfirst("TEXT_1", "did not set")
text2 = form.getfirst("TEXT_2", "did not set")
text1 = html.escape(text1)
text2 = html.escape(text2)

print("Content-type: text/html\n")
print("""<!DOCTYPE HTML>
    <html>
    <head>
        <meta charset="utf-8">
        <title>Form data</title>
    </head>
    <body>""")

print("<h1>Form data!</h1>")
print("<p>TEXT_1: {}</p>".format(text1))
print("<p>TEXT_2: {}</p>".format(text2))

print("""</body>
    </html>""")
```

Тепер, перейшовши за посиланням <http://localhost:8000/> та відіславши на сервер заповнені дані форми, можна побачити їх обробку за допомогою form.py.

с) В каталозі cgi-bin створимо Python-скрипт для обробки cookies користувача під назвою cookie.py:


```
#!/usr/bin/env python3
```

```
import os
```

```
import http.cookies
```

```
cookie = http.cookies.SimpleCookie(os.environ.get('HTTP_COOKIE'))
```

```
name = cookie.get('name')
```

```
if name is None:
```

```
    print("Set-cookie: name=value")
```

```
    print("Content-type: text/html\n")
```

```
    print("Cookies!!!")
```

```
else:
```

```
    print("Content-type: text/html\n")
```

```
    print("Cookies:")
```

```
    print(name.value)
```

Перейшовши за посиланням <http://localhost:8000/cgi-bin/cookie.py> перший раз – отримаємо результат: *Cookies!!!*. Перезавантаживши дану веб-сторінку – результат буде наступним: *Cookies: value*.

2) Створимо прототип веб-додатку “twitter”.

а) В каталозі cgi-bin створимо Python-скрипт для обробки повідомлень користувача під назвою wall.py:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
```

```
import cgi
import html
import http.cookies
import os, unicodedata
```

```
from _wall import Wall
wall = Wall()
```

```
cookie =
http.cookies.SimpleCookie(os.environ.get("HTTP_COOKIE"))
session = cookie.get("session")
if session is not None:
    session = session.value
user = wall.find_cookie(session)
```

```
form = cgi.FieldStorage()
action = form.getfirst("action", "")
```

```
if action == "publish":
    text = form.getfirst("text", "")
    text = html.escape(text)
    if text and user is not None:
        wall.publish(user, text)
elif action == "login":
    login = form.getfirst("login", "")
    login = html.escape(login)
    password = form.getfirst("password", "")
    password = html.escape(password)
    if wall.find(login, password):
        cookie = wall.set_cookie(login)
        print('Set-cookie: session={}'.format(cookie))
    elif wall.find(login):
        pass
    else:
        wall.register(login, password)
        cookie = wall.set_cookie(login)
        print('Set-cookie: session={}'.format(cookie))
```

```
pattern = '''
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title> The wall</title>
</head>
<body>
    Login and registration form. If you enter a nonexistent name, a new user would be registered.
    <form action="/cgi-bin/wall.py">
        Login: <input type="text" name="login">
        Password: <input type="password" name="password">
        <input type="hidden" name="action" value="login">
        <input type="submit">
    </form>

    {posts}

    {publish}
</body>
</html>
'''
```

if user is not None:

pub = ""

<form action="/cgi-bin/wall.py">

<textarea name="text"></textarea>

<input type="hidden" name="action" value="publish">

<input type="submit">

</form>

""

else:

pub = ""

print('Content-type: text/html\n')

print(pattern.format(posts=wall.html_list(), publish=pub))

b) В каталозі cgi-bin створимо Python-скрипт для визначення функцій користувача користувача під назвою _wall.py:

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-
```

```
import json  
import random  
import time
```

```
class Wall:  
    USERS = 'cgi-bin/users.json'  
    WALL = 'cgi-bin/wall.json'  
    COOKIES = 'cgi-bin/cookies.json'
```

```
def __init__(self):
    """Створюємо початкові файли, якщо вони не
створені"""
    try:
        with open(self.USERS, 'r', encoding='utf-8'):
            pass
    except FileNotFoundError:
        with open(self.USERS, 'w', encoding='utf-8') as f:
            json.dump({}, f)

    try:
        with open(self.WALL, 'r', encoding='utf-8'):
            pass
    except FileNotFoundError:
        with open(self.WALL, 'w', encoding='utf-8') as f:
            json.dump({"posts": []}, f)

    try:
        with open(self.COOKIES, 'r', encoding='utf-8'):
            pass
    except FileNotFoundError:
        with open(self.COOKIES, 'w', encoding='utf-8') as f:
            json.dump({}, f)
```

```
def register(self, user, password):
    """Реєструє користувача. Повертає True при успішній реєстрації"""
    if self.find(user):
        return False # Такий користувач існує
    with open(self.USERS, 'r', encoding='utf-8') as f:
        users = json.load(f)
    users[user] = password
    with open(self.USERS, 'w', encoding='utf-8') as f:
        json.dump(users, f)
    return True
```

```
def set_cookie(self, user):
    """Записує куку в файл. Повертає створену куку."""
    with open(self.COOKIES, 'r', encoding='utf-8') as f:
        cookies = json.load(f)
    cookie = str(time.time()) + str(random.randrange(10**14)) # Генеруємо унікальну куку для
користувача
    cookies[cookie] = user
    with open(self.COOKIES, 'w', encoding='utf-8') as f:
        json.dump(cookies, f)
    return cookie
```

```
def find_cookie(self, cookie):  
    """По Куке знаходить своє ім'я користувача"""  
    with open(self.COOKIES, 'r', encoding='utf-8') as f:  
        cookies = json.load(f)  
    return cookies.get(cookie)
```

```
def find(self, user, password=None):  
    """Шукає користувача по імені або по імені і паролю"""  
    with open(self.USERS, 'r', encoding='utf-8') as f:  
        users = json.load(f)  
    if user in users and (password is None or password == users[user]):  
        return True  
    return False
```

```
def publish(self, user, text):  
    """публікує текст"""  
    with open(self.WALL, 'r', encoding='utf-8') as f:  
        wall = json.load(f)  
    wall['posts'].append({'user': user, 'text': text})  
    with open(self.WALL, 'w', encoding='utf-8') as f:  
        json.dump(wall, f)
```

```
def html_list(self):  
    """Список постів для відображення на сторінці"""  
    with open(self.WALL, 'r', encoding='utf-8') as f:  
        wall = json.load(f)  
    posts = []  
    for post in wall['posts']:  
        content = post['user'] + ' : ' + post['text']  
        posts.append(content)  
    return '<br>'.join(posts)
```

Перейшовши за посиланням
<http://localhost:8000/cgi-bin/wall.py> **можна**
побачити в дії створений веб-додаток.

Лекцію закінчено.
Дякую за увагу!