Assignment #3
(Introduction to Computer Networks)

학번: 2013311659

이름: 곽 창 근

1. 개발 환경

OS : Windows7

사용 언어 : Python 3.6.5 64bit

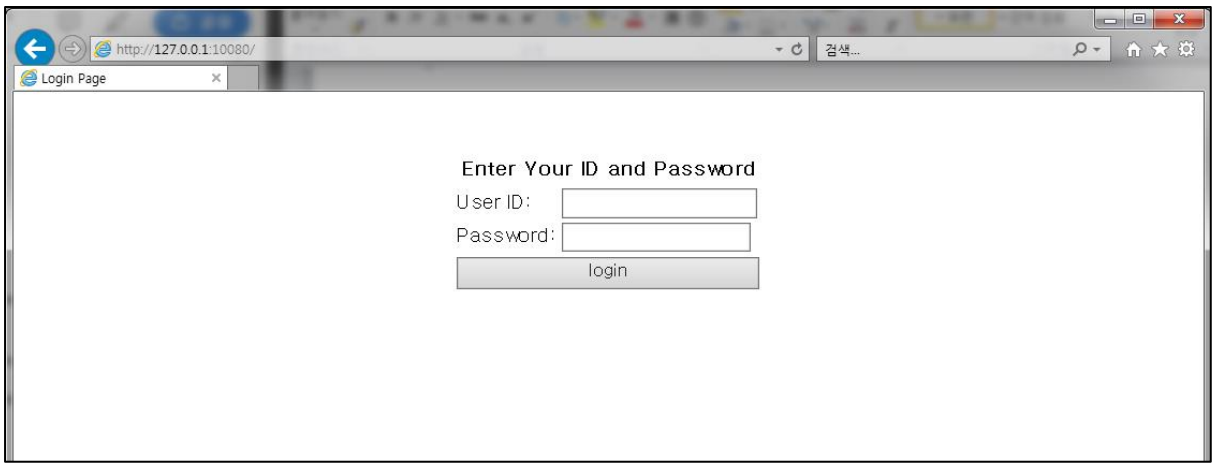
사용 프로그램 : Python 3.6.5 Shell

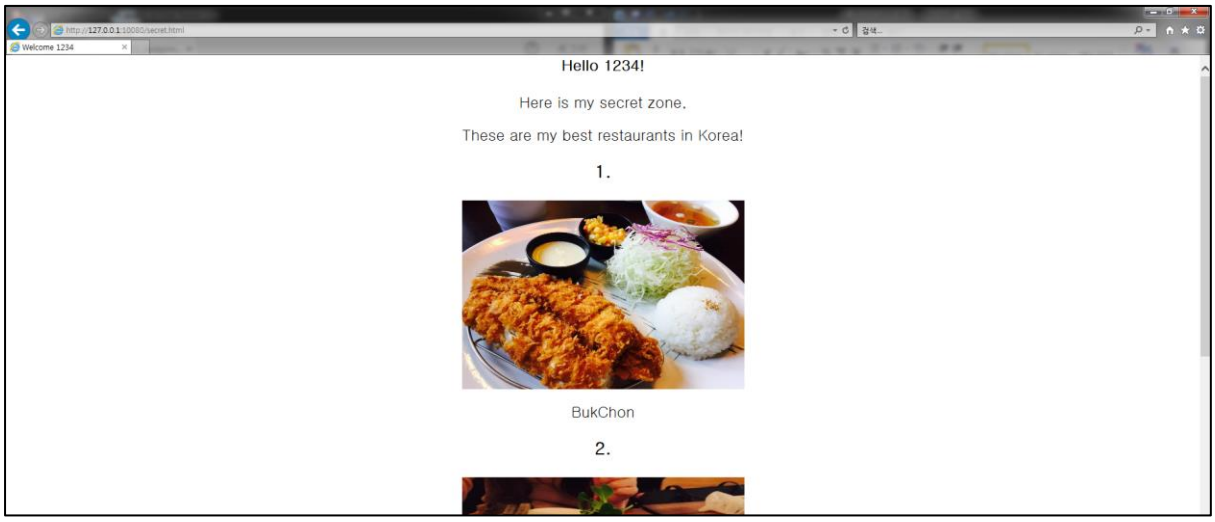
2. 설명 및 작동 방법

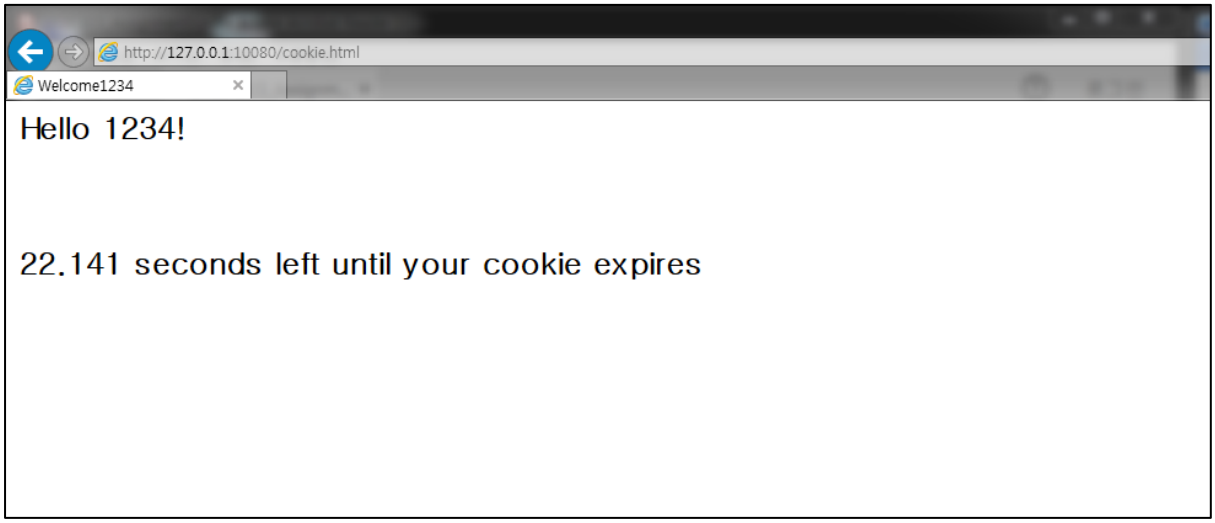
4.1, 4.2, 4.3 모두 구현하였다.

4.1) html 파일은 index.html secret.html cookie.html 이렇게 3 가지로 구성되어있고, 이미지 파일은 zirinsung.jpg myeongdongkyoja.jpg masibasi.jpg bukchon.jpg

이렇게 4 가지로 구성되어있다. 모든 파일은 server.py 파일과 같은 디렉터리 안에 저장되어있다.







위 사진들은 브라우저를 통해 3 개의 html 페이지에 접근한 모습이다.

여러 이미지를 불러오는 것이나 멀티브라우저를 위하여 쓰레딩이 필요하였다. 쓰레딩을 위하여 threading 모듈을 import 하였으며, while 문 안에

```
for x in range(500) :
        t = threading.Thread()
        t.start()
```

이 구문을 통하여 쓰레딩이 된다. 그래서 멀티 브라우저나 이미지 파일을 불러와 볼 수 있다.

웹 서버에서 파일을 불러오는 것은

```
buffer = bytearray()
chunk_size = 4096
```

```
msg = connectionSocket.recv(1024).decode()
filename = msg.split()[1]
if(filename == '/') :
        filename = '/index.html'
f = open(filename[1:], mode='rb')
while True :
        chunk = f.read(chunk_size)
        if len(chunk) < 1:
            break
        connectionSocket.send(chunk)
```

위 코드들로 이루어졌다. buffer 는 파일을 나눠보내기 위한 용도이고, chunk_size 는 나눠보내는 크기이다. msg.split()[1]에 경로를 포함한 요청하는 파일이름이 들어있다. 이를 ‘rb’모드로 파일을 읽어들이는 것이다.

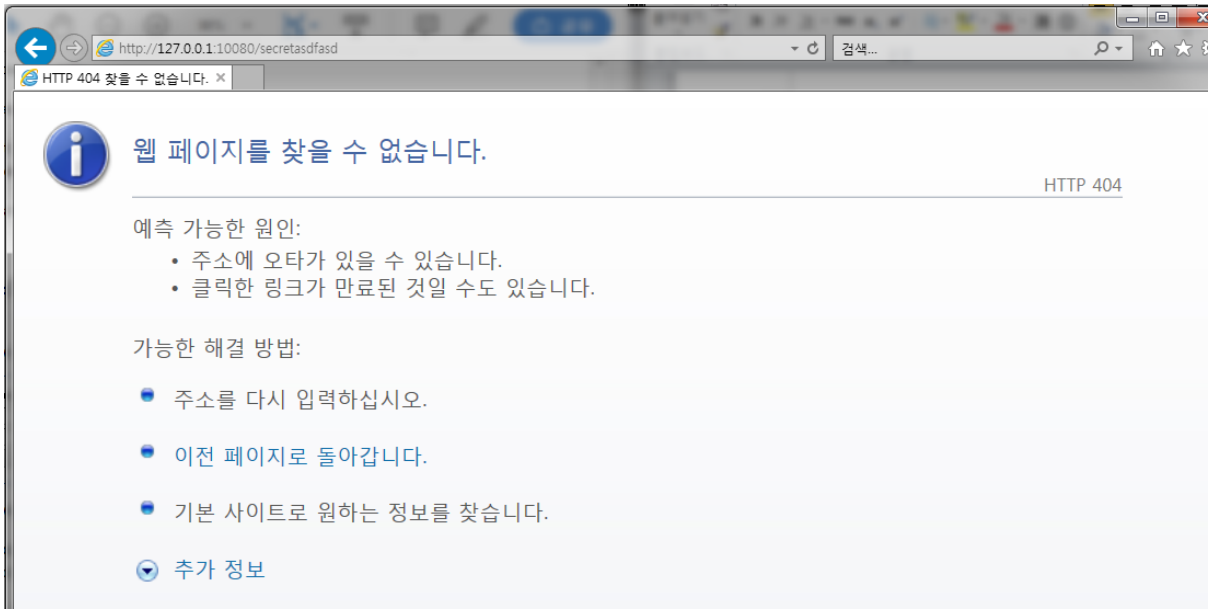
만약 웹서버에 있지 않은 파일을 요청할 경우,

```
filename = msg.split()[1]
        if(filename == '/') :
            filename = '/index.html'
            f = open(filename[1:], mode='rb')
```

위 구문에서 현재 존재하지 않은 파일을 open 하려고 시도하게 되므로 IOError 가 발생하게 된다. 이를

```
except IOError:
        connectionSocket.send("HTTP/1.1 404 OK\r\n\r\n".encode())
        connectionSocket.close()
```

위 구문을 통하여 에러를 캐치하고, 404 http response 를 보내도록 하였다. 아래 사진은 없는 파일을 요청했을 경우이다.



4.2)

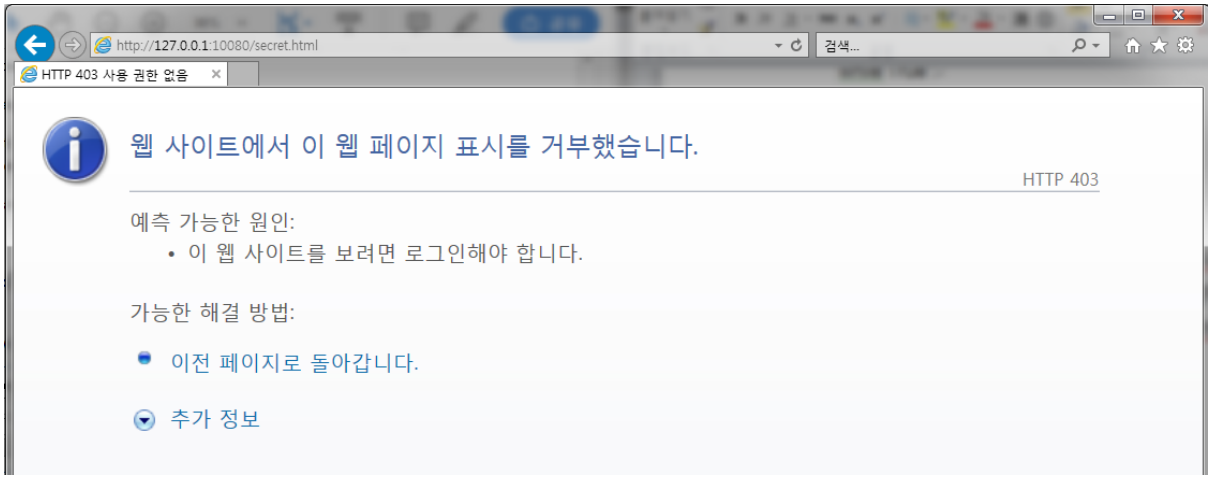
```
if(filename == '/') :
    filename = '/index.html'
```

위 코드로 요청하는 파일이름이 없을 경우에도 index.html 파일이 불러오게 하였다.

로그인을 하지 않았을 때 다른 곳으로의 접근을 막는 것은

```
if((user_id == '') and filename != '/index.html' and filename != '/') :
        connectionSocket.send("HTTP/1.1 403 Forbidden\r\n\r\n".encode())
        connectionSocket.close()
        continue
```

위 코드를 통해 user_id 가 입력되지 않아 비어있다면 403Forbidden 을 내보내도록 하였다. 아래는 로그인 하지 않고 secret.html 에 접근하려고 할 때 403 forbidden 이 출력되었다.



user_id 에 대한 쿠키가 있는지 매번 확인하는 것은

```
#쿠키 체크
user_id = pw = ''
for s in msg.split() :
        if(s[0:7] == "User_ID") :
            #id, pw 저장
            value = s.split('&')
            user_id = value[0].split('=')[1]
            pw = value[1].split('=')[1]
            break
        elif(s[0:7] == "user_id") :
            user_id = s[8:]
            break;
```

위 코드를 통해서 확인하도록 하였다. 수신된 http response 에서 User_ID 를 찾아서 읽은 방식이다.

cookie.html 은 위에 스크린샷으로 찍었던 것처럼 초단위까지 잘 출력된다. 남은 시간을 계산하는건 자바스크립트를 통해 구현하였다. cookie.html 파일에

```
var getCookie = function(name) {
    var value = document.cookie.match('(^|;) ?' + name + '=([^;]*)(;|$)');
    return value? value[2] : null;
```

```
        };

    function getSecond() {
        var d_date = new Date(getCookie('date'));
        var date = new Date();
        date.setTime(date.getTime());
        return (d_date-date)/1000;
    }
```

위 두 함수를 만들어서 사용하였다. getCookie 는 저장된 쿠키값을 불러오는 함수다. 위를 통하여 user_id 값도 불러와서 “Welcome [ID]!” 의 환영문도 출력하도록 하였다. index.html 에서 로그인을 하고 들어올 때 user_id, pw 뿐만 아니라 로그인 시간도 ‘date’라는 쿠키 이름에 저장하였다. 이를 통하여

```
        var d_date = new Date(getCookie('date'));
```

문장을 통해 저장된 쿠키의 시간을 불러오고, var date 에 현재 시간을 입력하여서 return (d_date-date)/1000; 을 통해 남은 시간을 불러오도록 구현하였다.

4.3) 브라우저에서 요청한 파일을 불러올 때 불러오기 성공하면 http response 를 보내줄 때

```
connectionSocket.send("HTTP/1.1 200 OK\r\nConnection: Keep-Alive\r\nKeep-Alive: timeout=5, max=1000\r\n\r\n".encode())
```

보내주는 메시지에 Connection:Keep-Alive 와 Keep-Alive:timeout=5, max=1000 이라는 문구를 추가해서 보냈다. 이는 통신 한 번 주고받고 끝나지 않고 연결을 유지하는데, timeout=5 동안, 그리고 최대 1000 개의 연결을 유지할 수 있도록 하라는 메시지이다. 그래서 http response 를 보면,

위와 같이 persistent http 모드로 잘 설정된 것을 볼 수 있다.