

8주차

리눅스 시스템

2024.동계계절학기

CONTENTS

1. 파이썬 플라스크 - 회원 가입 및 로그인

요구사항

- **개요**
 - 회원 가입과 로그인을 할 수 있는 사이트이다.
- **로그인 페이지**
 - 아이디가 없으면 회원 가입 페이지로 이동할 수 있다.
 - 가입한 아이디와 암호를 입력하여 로그인을 할 수 있다.
 - 로그인에 실패하면 실패 메시지가 뜬다.
 - 로그인에 성공하면 home 페이지로 이동한다.
- **회원가입 페이지**
 - 사이트에 접속하여 회원 가입 페이지로 이동한다.
 - 아이디와 암호를 입력한 후 회원 가입을 한다.
 - 회원 정보는 서버의 json 파일에 저장한다.
- **Home 페이지**
 - 로그인에 성공하면 Home 화면으로 이동한다.
 - Home 화면에서는 로그아웃을 할 수 있다.
 - 로그아웃을 하면 로그인 페이지로 이동한다.
 - 로그인 하지 않고 Home 페이지로 오면 로그인 페이지로 이동한다.

- JSON

- JavaScript Object Notation은 데이터 교환을 위해 개발된 경량 데이터 형식
- JSON은 언어 독립적이며, 대부분의 프로그래밍 언어에서 사용이 가능함
- 웹 개발에서 클라이언트<->서버 간의 데이터를 전송하는 표준 방식으로 많이 사용됨
- {}로 감싸져 있으면 key,value 쌍이 나와야 함. 순서는 상관없음
- []는 배열로 어떤 형태라도 들어갈 수 있으며 순서가 있음

```
{
  "name": "Park Jonggyu",
  "age": 19,
  "isStudent": false,
  "address": {
    "street": "Chungryeol-ro",
    "city": "Busan"
  },
  "phoneNumbers": ["010-1234-5678", "070-0000-9999"]
}
```

```
# 객체를 JSON 문자열로 변환
json_obj = {"name": "Jong", "age": 19}
json_string = json.dumps(json_obj)
print(json_string)

# JSON 형태의 문자열을 객체로 변환
json_string = '{"name": "Jong", "age": 19}'
json_obj = json.loads(json_string)
print(json_obj["name"])
```

프로젝트 구조

```
my_flask_project/
├── templates/
│   ├── login.html      # 로그인 및 첫 번째 화면
│   ├── register.html   # 회원가입 화면
│   └── home.html        # 로그인 후 홈 화면
├── static/
│   └── style.css        # CSS 파일
├── users.json           # 사용자 데이터를 저장하는 JSON 파일
└── app.py
```

app.py - 구조

• 기본 및 라우트 구조

- **load_users()**
 - 로그인 시에 저장된 회원정보를 불러오는 기능
- **save_users(users)**
 - 회원 가입시에 회원 정보를 저장하는 기능
- **login()**
 - 사용자가 입력한 이름과 아이디가 맞는지 확인
 - 성공하면 세션을 저장하고 home으로 이동
- **register()**
 - 사용자가 입력한 이름과 아이디를 저장
- **home()**
 - 로그인하지 않은 사용자가 접근하면 로그인 페이지로 보냄
 - 로그인에 성공하여 세션이 있으면 홈화면을 보여줌
- **logout()**
 - 세션을 제거하고 로그인 화면으로 보냄

```
import json
from flask import Flask, render_template, request, redirect, url_for, session

app = Flask(__name__)

def load_users():
    return

def save_users(users):
    return

@app.route("/", methods=["GET", "POST"])
def login():
    message = None
    return render_template("login.html", message=message)

@app.route("/register", methods=["GET", "POST"])
def register():
    return render_template("register.html")

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

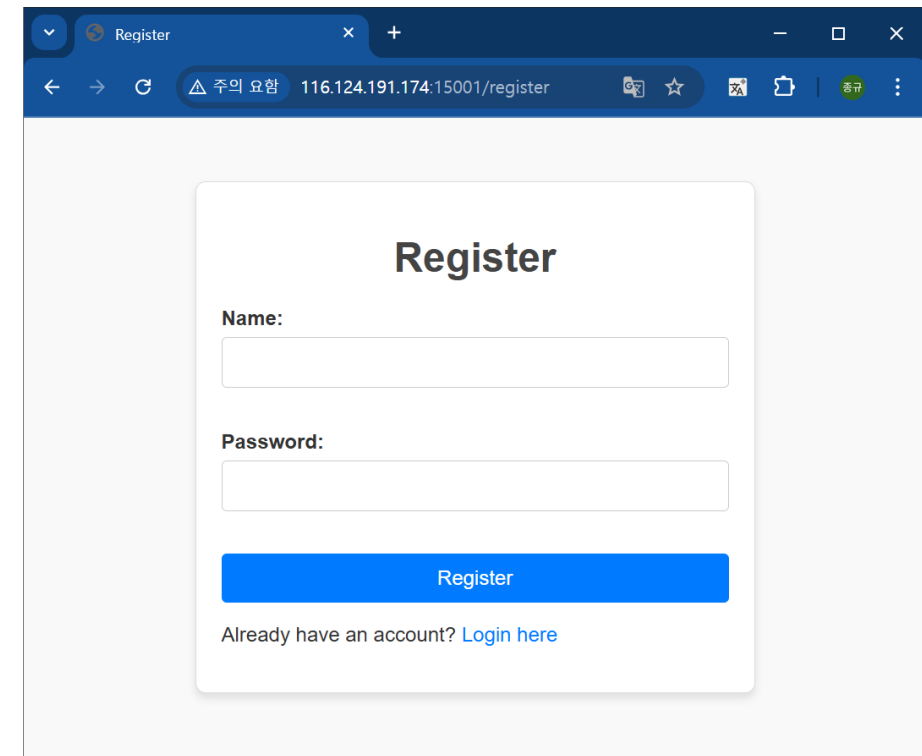
@app.route("/logout", methods=["POST"])
def logout():
    return redirect(url_for("login"))

if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=15001)
```

register.html

- register.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Register</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
  <div class="container">
    <h1>Register</h1>
    <form method="post">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name" required>
      <br>
      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required>
      <br>
      <button type="submit">Register</button>
    </form>
    <p>Already have an account? <a href="/">Login here</a></p>
  </div>
</body>
</html>
```



The screenshot shows a web browser window with the title 'Register'. The address bar displays the URL '116.124.191.174:15001/register'. The main content area features a registration form with the following elements:

- Title:** Register
- Name:** A text input field.
- Password:** A password input field.
- Register Button:** A blue button labeled 'Register'.
- Footer:** A link that says 'Already have an account? [Login here](#)'.

app.py - register

- **register()**

```
@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        name = request.form.get("name")
        password = request.form.get("password")
        users = {}
        users[name] = password
        save_users(users)
        return redirect(url_for("login"))

    return render_template("register.html")
```


app.py - register

- **save_users(users)**

```
USER_FILE = "users.json"

def save_users(users):
    with open(USER_FILE, "w") as file:
        json.dump(users, file)
```

- **users.json**

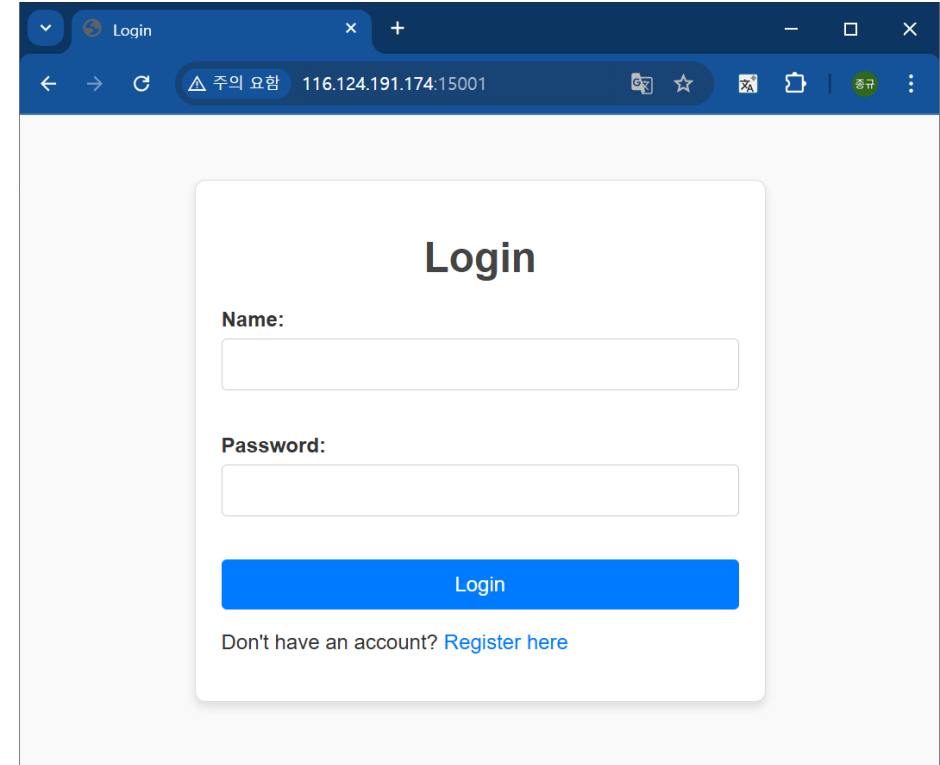
```
{}
```

login.html

- login.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Login</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
  <div class="container">
    <h1>Login</h1>
    <form method="post">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name" required>
      <br>
      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required>
      <br>
      <button type="submit">Login</button>
    </form>
    <p>Don't have an account? <a href="/register">Register here</a></p>

    {% if message %}
      <p class="message">{{ message }}</p>
    {% endif %}
  </div>
</body>
</html>
```

A screenshot of a web browser window displaying a login page. The browser's address bar shows the URL "116.124.191.174:15001". The page has a dark blue header with the title "Login". The main content area is white and contains a login form. The form has a title "Login" in bold. Below the title, there are two input fields: "Name:" and "Password:". The "Name:" field is a simple text input, and the "Password:" field is a password input with a small eye icon to toggle visibility. Below the input fields is a blue "Login" button. At the bottom of the form, there is a link: "Don't have an account? Register here".

app.py - login

- login()

```
app.secret_key = "secret_key" // 임의값

@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        name = request.form.get("name")
        password = request.form.get("password")
        users = {}
        users[name] = password
        save_users(users)
        return redirect(url_for("login"))

    return render_template("register.html")
```

app.py - login

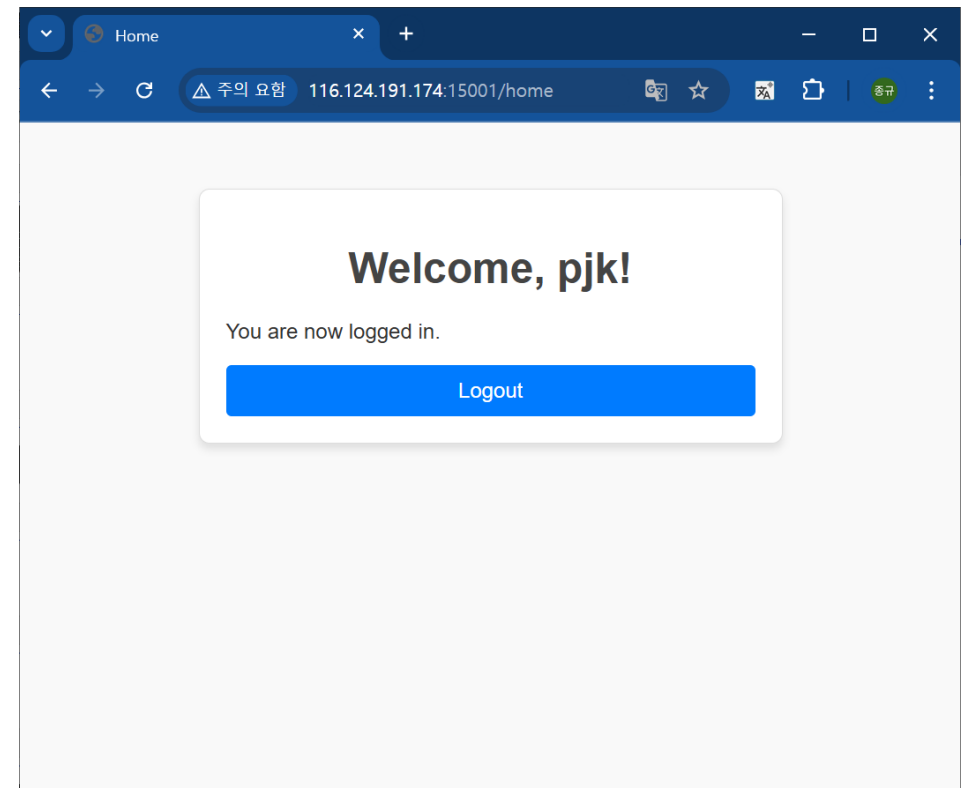
- **load_users()**

```
def load_users():  
    try:  
        with open(USER_FILE, "r") as file:  
            return json.load(file)  
    except FileNotFoundError:  
        return {}
```

home.html

- **home()**

```
<!DOCTYPE html>
<html>
<head>
  <title>Home</title>
  <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
</head>
<body>
  <div class="container">
    <h1>Welcome, {{ user }}!</h1>
    <p>You are now logged in.</p>
    <form action="/logout" method="post">
      <button type="submit">Logout</button>
    </form>
  </div>
</body>
</html>
```



app.py - home

- **home()**

```
@app.route("/home")
def home():
    if "user" not in session:
        return redirect(url_for("login"))
    return render_template("home.html", user=session["user"])
```

- **logout()**

```
@app.route("/logout", methods=["POST"])
def logout():
    session.pop("user", None)
    return redirect(url_for("login"))
```

app.py - 전체

```
import json
from flask import Flask, render_template, request, redirect, url_for, session

app = Flask(__name__)
app.secret_key = "secret_key"
USER_FILE = "users.json"

def load_users():
    try:
        with open(USER_FILE, "r") as file:
            return json.load(file)
    except FileNotFoundError:
        return {}

def save_users(users):
    with open(USER_FILE, "w") as file:
        json.dump(users, file)

@app.route("/", methods=["GET", "POST"])
def login():
    message = None
    if request.method == "POST":
        name = request.form.get("name")
        password = request.form.get("password")

        users = load_users()
        if users.get(name) == password:
            session["user"] = name
            return redirect(url_for("home"))
        else:
            message = "Login Fail"

    return render_template("login.html", message=message)
```

<1/2>

```
@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        name = request.form.get("name")
        password = request.form.get("password")
        users = {}
        users[name] = password
        save_users(users)
        return redirect(url_for("login"))

    return render_template("register.html")

@app.route("/home")
def home():
    if "user" not in session:
        return redirect(url_for("login"))
    return render_template("home.html", user=session["user"])

@app.route("/logout", methods=["POST"])
def logout():
    session.pop("user", None)
    return redirect(url_for("login"))

if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=15001)
```

<2/2>

style.css

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f9f9f9;
  color: #333;
}

.container {
  max-width: 400px;
  margin: 50px auto;
  padding: 20px;
  background: #fff;
  border: 1px solid #ddd;
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

h1 {
  text-align: center;
  margin-bottom: 20px;
  color: #444;
}

form {
  display: flex;
  flex-direction: column;
}
```

<1/3>

```
label {
  margin-bottom: 5px;
  font-weight: bold;
}

input {
  margin-bottom: 15px;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

button {
  padding: 10px;
  font-size: 16px;
  background-color: #007BFF;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}
```

<2/3>

```
a {
  color: #007BFF;
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

.message {
  margin-top: 15px;
  padding: 10px;
  background-color: #f8d7da;
  color: #721c24;
  border: 1px solid #f5c6cb;
  border-radius: 4px;
}
```

<3/3>

- 어떤 기능을 추가할 수 있을까?
 - 회원 가입시 이름/암호 외에 추가정보
 - 회원 가입 시 암호 한번 더 확인하는 기능
 - 회원 가입 시 기존에 있는 이름인지 확인하는 기능
 - 회원 가입 시 이름이나 암호를 하나라도 안 넣으면 에러가 나게 하는 기능
 - 회원 가입 시...
 - 로그인 시...