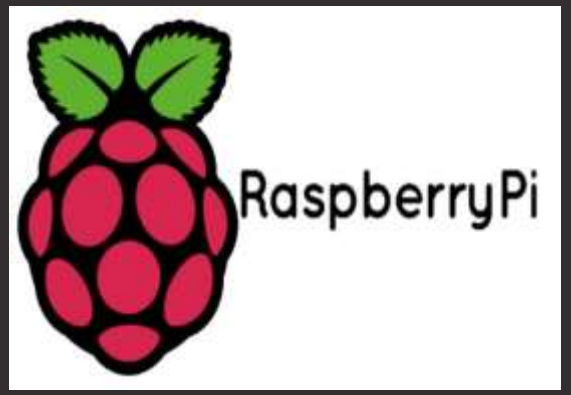
Raspberry Pi

WebServer Project

Kim-HyunSeung/RaspBerry: 라즈베리파이를 이용한 다중통신 및 실습 (github.com)

작성자 : 김현승

Using Tool

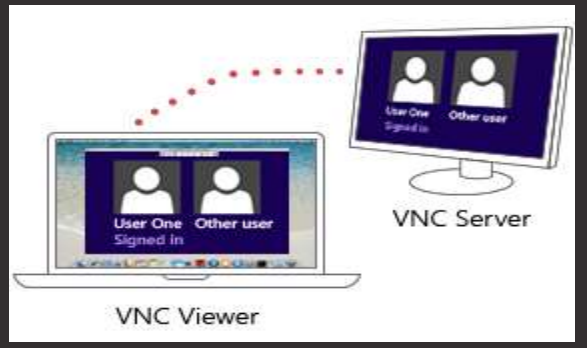


Raspberry Pi

MariaDB

Linux

Flask



VNC Server

INDEX

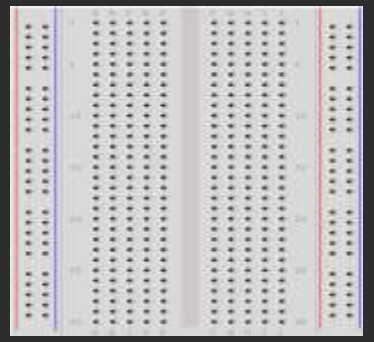
프로젝트 설명

라즈베리파이 내에 있는 마리아 DB에
온습도 센서를 이용한 온도 및 습도값을
저장 후, 웹 플라스크를 이용한 웹서버로
온도 및 습도값을 전달 후, 웹페이지 로그인
여부에따라 실시간 온도 및 습도를 표현.

Using HardWare

Raspberry Pi

DHT-11

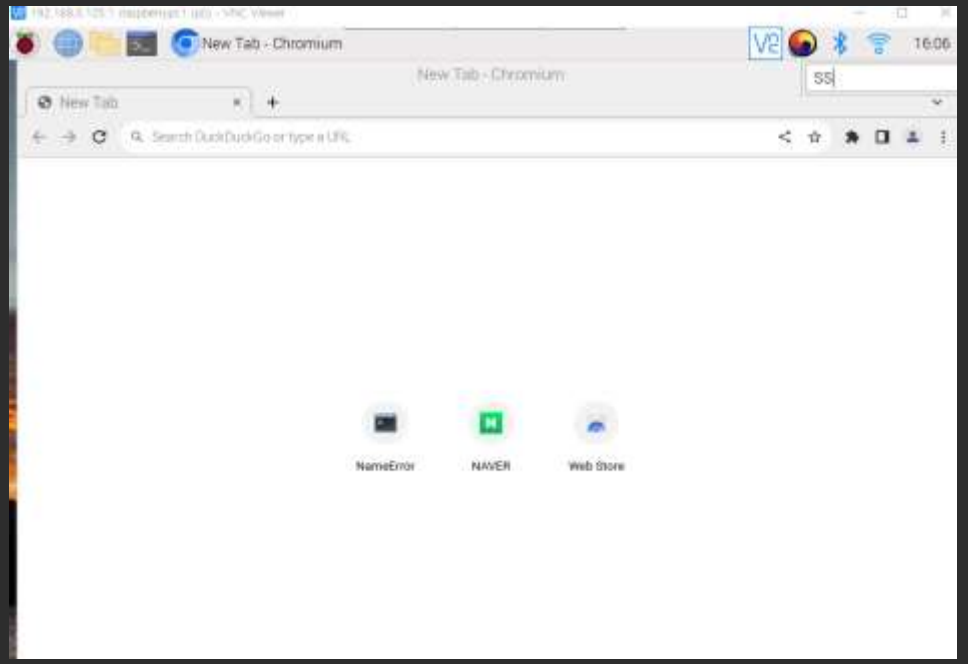


브레드보드

Using SoftWare

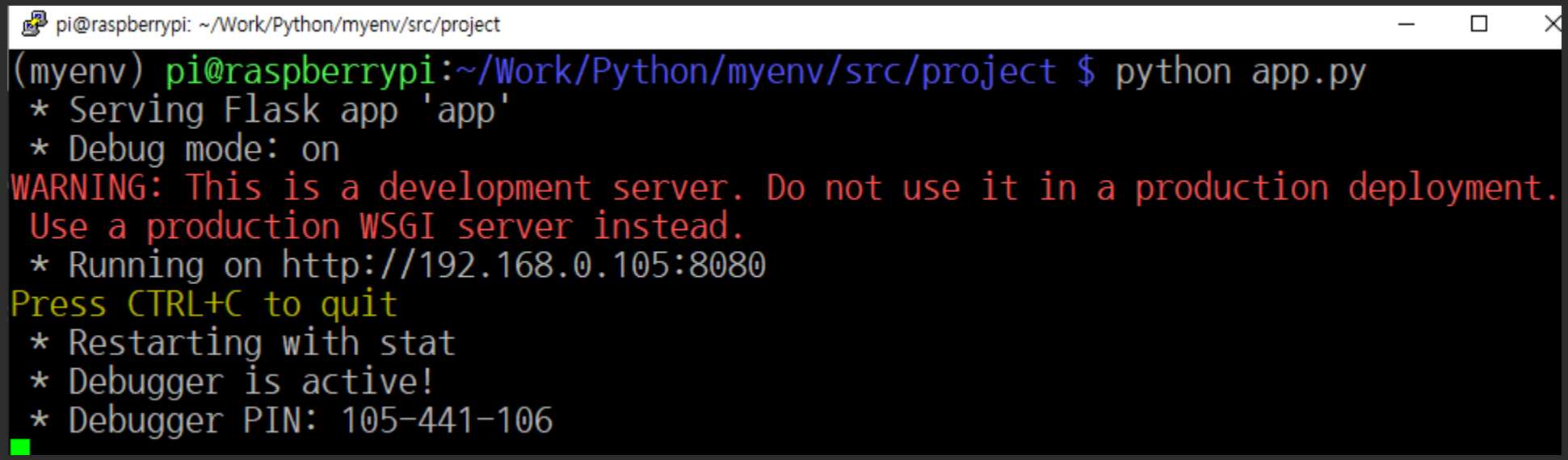
```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 168
Server version: 10.5.15-MariaDB-0+deb11u1 Raspbian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use P01
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [P01]> show tables;
+-----------------+
| Tables_in_P01   |
+-----------------+
| P02             |
| login           |
| loginflag       |
| loginflag2      |
| loginflag3      |
| loginflag4      |
| temp            |
| temp2           |
| temp3           |
| temp4           |
| temp5           |
+-----------------+
11 rows in set (0.001 sec)
```

Linux



VNC Server

주요 실행 화면



App.py 파일을 실행시켜 웹서버를 실행.

실행 주요 코드

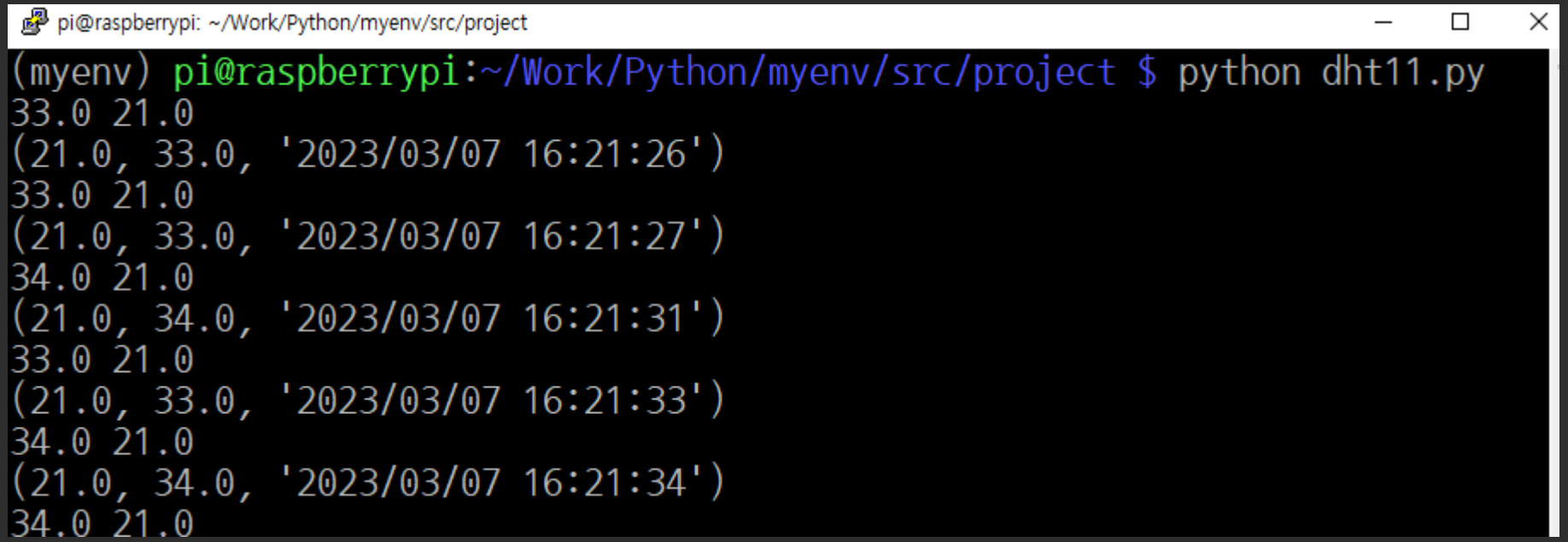
```
from flask import request
import time
import pymysql as ps
from flask import Flask,render_template,url_for,session,request,redirect

conn = ps.connect(host = 'localhost',user='hyun',password='1234',db='P01',charset='utf8')
curs = conn.cursor()

app = Flask(__name__)
app.secret_key = "hyun1111"
```

```
@app.route("/temp")
def temp():
    while True:
        sql = """SELECT * from temp5 order by date desc"""
        curs.execute(sql)
        sql_all=curs.fetchall()
        conn.commit()
        return render_template('index.html',list=sql_all)
```

```
if __name__ == '__main__':
    app.run(host='192.168.0.105',port = 8080,debug='True')
```

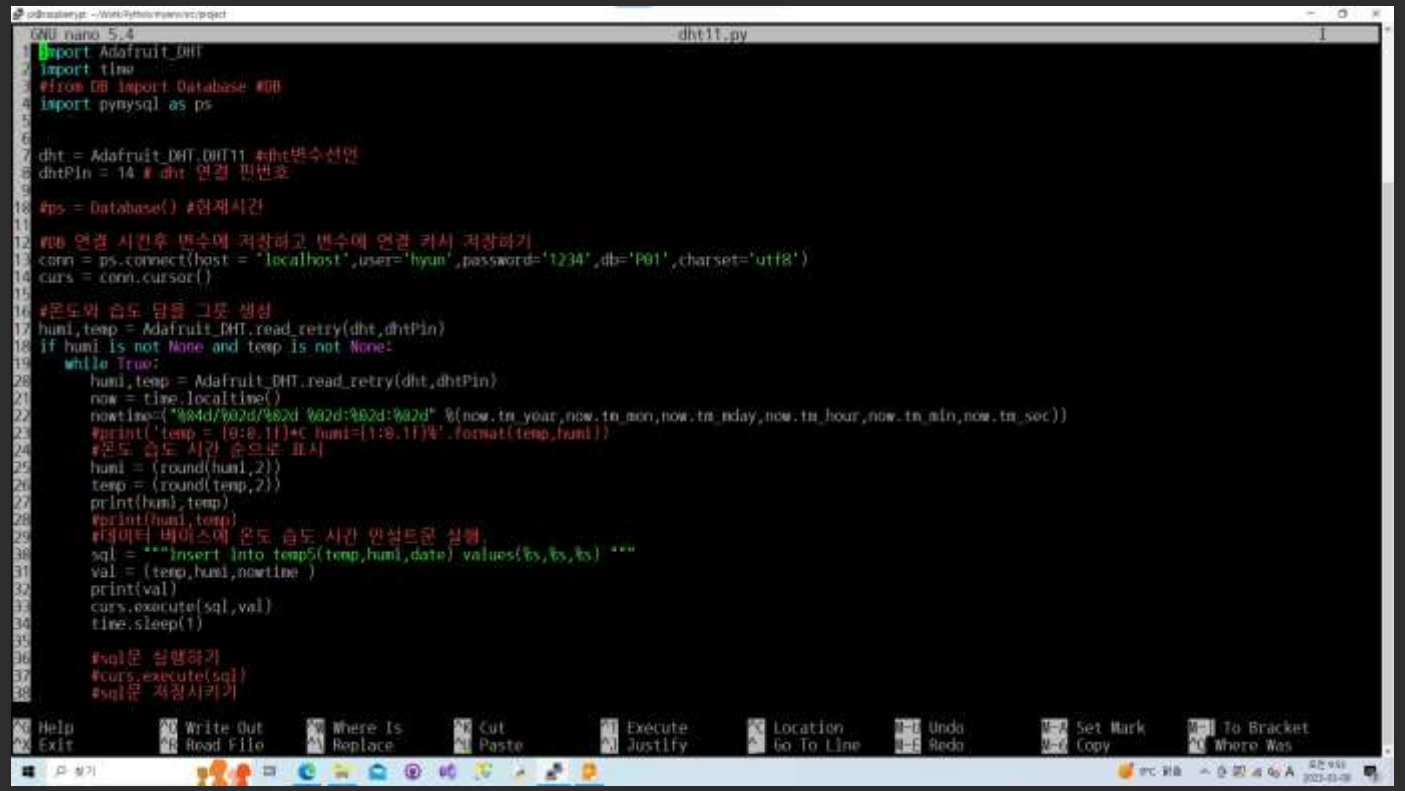
App.py 실행시켰을때, DB랑 연동도 같이 시키고, 웹페이지에 보여줄 html 파일을 불러옴.

주요 실행 화면



1. Dht11.py 파일을 실행시켜 온도, 습도, 날짜가 정상적으로 출력되는지 확인.
2. 출력된 값들이 정상적으로 DB에 들어가는지 확인

실행 주요 코드



온도,습도,날짜를
1초마다 출력하여
DB에 값을 저장시킨다.

주요 실행 화면

```
MariaDB [P01]> SELECT * FROM temp5;
+-------+-------+---------------------+
| temp  | humi  | date                |
+-------+-------+---------------------+
| 24.00 |  9.00 | 2023/03/03 13:25:21 |
| 24.00 |  9.00 | 2023/03/03 13:25:23 |
| 24.00 | 10.00 | 2023/03/03 13:25:24 |
| 24.00 |  9.00 | 2023/03/03 13:25:26 |
| 24.00 |  9.00 | 2023/03/03 13:25:27 |
| 24.00 | 10.00 | 2023/03/03 13:25:29 |
| 24.00 |  9.00 | 2023/03/03 13:25:30 |
| 24.00 |  9.00 | 2023/03/03 13:25:32 |
| 24.00 |  9.00 | 2023/03/03 13:25:33 |
| 24.00 |  9.00 | 2023/03/03 13:25:35 |
| 24.00 |  9.00 | 2023/03/03 13:25:36 |
| 24.00 |  9.00 | 2023/03/03 13:25:38 |
```

Maria DB 창을 실행시킨 후, 조회문을 통해 테이블에
온도,습도,날짜 값이 들어가는지 확인

주요 실행 화면

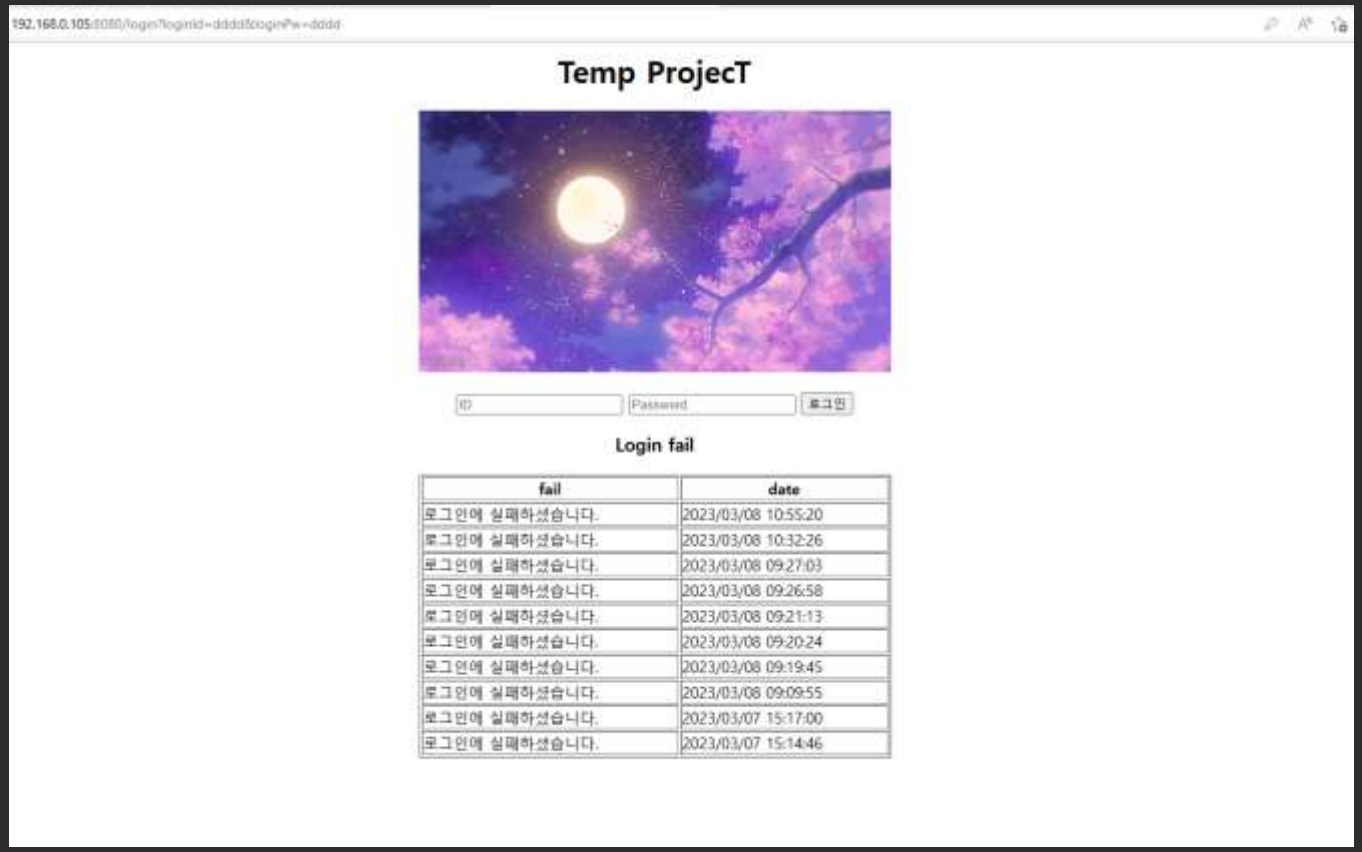
지정한 IP와 Port 주소로 웹서버와 연결 및 데이터 베이스에 저장된 값들이 정상적으로 출력되는지 확인.

실행 주요 코드

```
GNU nano 5.4                                  index.html                                  1
<!DOCTYPE html>
<html lang ="en">
    <head>
    <meta charset ="UTF-8">
        <p><h1><center><span style="border-radius: 15px 15px 15px 0; border: 3px solid #FFA05B; padding:0.5em 0.6em;color: #FF8000;">TEMP HUMI TIME</span></
    </head>

    <body>
    <!--  background-image:href("https://p4.wallpaperbetter.com/wallpaper/466/485/262/hi-def-nature-pictures-2560x1440-wallpaper-preview.jpg");-->
        <body bgColor = "pink">
        <center><h2> 시간에따른 온도,습도 표현 </h2></center>
<center>
    <button type="button" onclick="document.bgColor='white'">흰색</button>
    <button type="button" onclick="document.bgColor='pink'">핑크색</button>
    <button type="button" onclick="document.bgColor='lightGreen'">연녹색</button>
    <br>
    <label for="start">Start date:</label>

    <br>
    <input type="date" id="start" name="trip-start"
           value="2018-07-22" min="2020-01-01" max="2023-12-30">


</center>

    <div><center>
        <table border = "1" width="500" height="200">
        <th>온도</th>
        <th>습도</th>
        <th>날짜</th>
        {% for i in list %}
    <tr>

    <td> {{i[0]}}C</td>
    <td> {{i[1]}}%</td>
    <td> {{i[2]}}</td>
    </tr>
        {% endfor %}
```

출력된 웹페이지에CSS 및 받아온 값들이 정상적으로 출력되는지 확인.

주요 실행 화면



App.py를 실행키시면 처음으로 나타나는 웹의 로그인 화면.

로그인이 실패하면 로그인 실패내역을 DB에서 받아와서 웹에 보여지게함.

실행 주요 코드

```python
@app.route("/")
def home():
    if "userID" in session:
        return render_template("home.html",username = session.get("userID"),login = True)
    else:
        return render_template("home.html",login = False)
```

```python
@app.route("/login", methods = ["GET"])
def login():
    global ID,PW
    _id_ = request.args.get("loginId")
    _password_ = request.args.get("loginPw")

    if ID == _id_ and _password_ == PW:
        session["userID"] = _id_
```

```python
    sql = """SELECT * from loginflag3 order by date desc limit 10"""
    curs.execute(sql)
    sql_all=curs.fetchall()
    conn.commit()

    fail = "로그인에 실패하셨습니다."
    if fail is not None:
        now = time.localtime()
        nowtime=("%04d/%02d/%02d %02d:%02d:%02d" %(now.tm_year,now.tm_mon,now.tm_mday, now.tm_hour,now.tm_min,now.tm_sec))
        sql = """insert into loginflag3(fail,date) values(%s,%s) """
        val = (fail,nowtime)
        curs.execute(sql,val)
        time.sleep(1)
        conn.commit()
    return render_template("home.html",list=sql_all,login=False)
```

입력한 ID 와 PW가 일치하지 않아, 로그인이 실패하였을 경우, 리턴값으로 기존의 home.html로 보냄.

실행 주요 코드

```
{%else %}
<center><h1>Temp ProjecT</h1></center>
<center><form methods = "get" id = "login" action = {{url_for("login")}}>
<img src=https://mblogthumb-phinf.pstatic.net/MjAxOTA4MDdfMTEw/MDAxNTY1MTg1MDk1OTUz.3fjmL-ODlTYUGNz88uVmGvwLT7JXWB4ZJtweP7I8gdcg.R4z5A8mfmCfsqGKS9pw
   <br>
   <br>
   <input type="id" id="loginId" name="loginId" placeholder="ID">
   <input type="password" id="loginPw" name ="loginPw" placeholder="Password">
   <button type="submit" class="login">로그인</button>
   <!--<body bgColor = "aqua">-->


<h3>Login fail</h3>
<div><center>
   <table border = "1" width="500" height="200">
   <th>fail</th>
   <th>date</th>
   {% for i in list %}
<tr>

<td>{{i[0]}}</td>
<td>{{i[1]}}</td>
</tr>
   {% endfor %}
   </table>
</div></center>
</form></center>
{%endif%}
```

html을 이용한 로그인 페이지 화면 구성.

주요 실행 화면

로그인이 성공적으로 됐을때, 이 페이지로 접속후, 로그인 성공 내역을 DB에서 받아 보여준다.

온습도 화면 버튼을 이용하여 온습도 웹페이지로 이동할수 있다.

실행 주요 코드

```
{% if login == True %}
<center><h3>로그인에 성공했습니다. hyen 님 어서오세요</h3></center>
<center><button type="button" onclick = "location.href='/temp'">온습도 확인</button>
<br>
<br>
<center><a href = {{url_for("logout")}}>로그아웃</a></center>
   <body bgColor = "beige">
<h3>Login success</h3>
<div><center>
   <table border = "1" width="500" height="200">
   <th>success</th>
   <th>date</th>
   {% for i in list %}
<tr>

<td>{{i[0]}}</td>
<td>{{i[1]}}</td>
</tr>
   {% endfor %}
   </table>
</div></center>
```

로그인이 됐을때,
화면을 꾸며주기 위한
html코드

실행 주요 코드

```
    sql = """SELECT * from loginflag4 order by date desc limit 10"""
#SELECT COUNT(*) FORM 로그인정보테이블 WHERE 아이디 = 입력한 아이디 AND 비밀번호 = 입력한 비밀번호
    curs.execute(sql)
    sql_all=curs.fetchall()
    conn.commit()

    success = "로그인에 성공하셨습니다."
#   if success is not None and flag2 ==0:
    if success is not None and ID ==_id_ and _password_ == PW:
       session["userID"] = _id_
       now = time.localtime()
       nowtime=("%04d/%02d/%02d %02d:%02d:%02d" %(now.tm_year,now.tm_mon,now.tm_mday,now.tm_hour,now.tm_min,now.tm_sec))
       #sql 쿼리문 작성
       sql = """insert into loginflag4(success,date) values(%s,%s) """
       val = (success,nowtime)
       #sql query 실행
       curs.execute(sql,val)
       time.sleep(1)
       conn.commit()
    return render_template('home.html',list=sql_all,login =True)
```

로그인이 되었을때, 성공한 내역과 페이지를 뛰우는 코드

실행 주요 코드

```
@app.route("/logout")
def logout():
    session.pop("userID")
    return redirect(url_for("home"))
```

로그아웃 활성화를 시키기 위한 코드

Thank You