

1. Jelaskan fungsi dari *session* sesuai dengan pemahaman anda!

Session merupakan sebuah proses disimpannya data pada *web server* yang dapat digunakan untuk berbagai keperluan di sebuah *website*, contohnya *session* digunakan untuk mengautentikasi sebuah *user* pada proses *login*.

2. Berikan penjelasan anda mengenai perbedaan antara *session* dan *cookie*!

Cookie bertujuan untuk menyimpan *info user* pada jangka panjang seperti informasi autentikasi yang digunakan untuk mengotomasi *login user* pada membuka *website* tersebut, dan disimpan pada pihak klien.

Sedangkan, **Session** bertujuan untuk menyimpan info tentang aktivitas terbaru si user. *session* ini disimpan di server daripada di klien.

3. Berikan contoh sederhana dan tambahkan penjelasan mengenai cara membuat *session*, mengubah nilainya dan menghapus *session* pada python!

Source Codes

a. app.py

```
from flask import Flask, render_template, session

app = Flask(__name__)
app.config["SECRET_KEY"] = "123"

@app.route("/")
def index():
    msg = "Session 'user' telah dibuat."
    session['user'] = 'Maidenless Tarnished'
    return render_template("index.html", msg = msg)

@app.route("/ubahsession")
def ubahsession():
    session['user'] = 'Accord'
    msg = "Session 'user' telah dirubah."
    return render_template("index.html", msg = msg)

@app.route("/hapussession")
def hapussession():
    if "user" in session.keys():
        msg = "Session 'user' telah dihapus."
        session.pop("user", None)
    else:
        msg = "session 'user' tidak ditemukan."
    return render_template("index.html", msg = msg)

@app.route("/showsession")
def showsession():
    if "user" in session.keys():
        user = session["user"]
    else:
        user = None
    return render_template("session.html", user = user)

if __name__ == "__main__":
    app.run(debug = True)
```

b. index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Nomor 3</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
  <p style="color: red;">{{ msg }}</p>

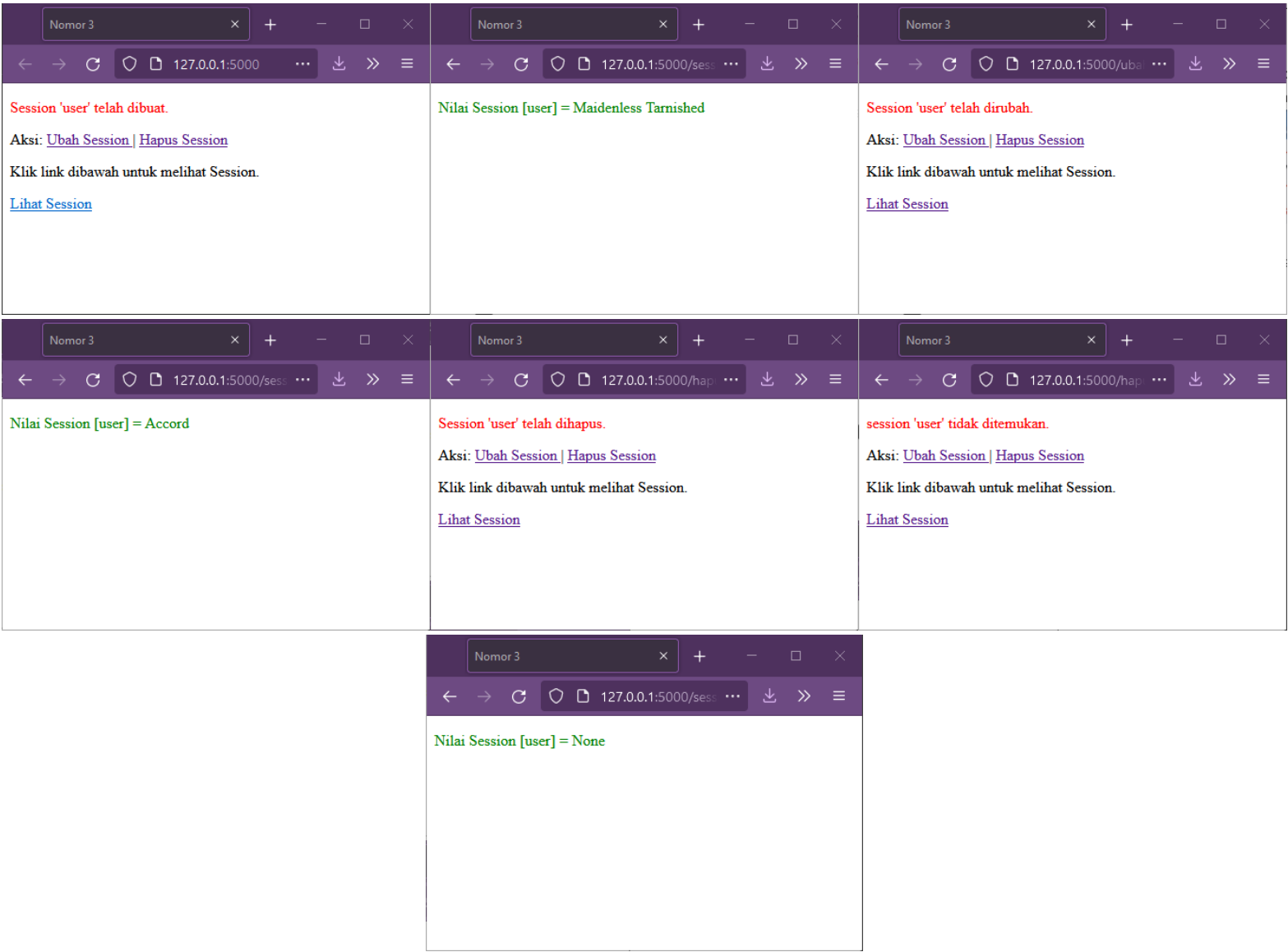
  <p>Aksi:
    <a href="http://127.0.0.1:5000/ubahsession"> Ubah Session </a> |
    <a href="http://127.0.0.1:5000/hapussession"> Hapus Session </a>
  </p>
```

```
<p>Klik link dibawah untuk melihat Session.</p>
<a href="http://127.0.0.1:5000/showsession"> Lihat Session </a>
</body>
</html>
```

c. session.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Nomor 3</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
  <p style="color: green;">Nilai Session [user] = {{ user }}</p>
</body>
</html>
```

ScreenShot



Penjelasan

Cara membuat sebuah session :

```
from flask import Flask, render_template, session

app = Flask(__name__)
```

└ Pertama kita *import* dulu *Flask* dan fungsi-fungsinya termasuk *session*.

```
app.config["SECRET_KEY"] = "123"
```

└ Tambahkan “**Secret Key**”, karena *Session* membutuhkan sebuah *Security* agar bisa berjalan.

```
@app.route("/")
def index():
    msg = "Session 'user' telah dibuat."
    session['user'] = 'Maidenless Tarnished'
    return render_template("index.html", msg = msg)
```

└ Buatlah fungsi *index* untuk mengakses *homepage session*, di dalam *index* ini terdapat *variable* pesan dan *session* yang dapat diisi dengan bebas. Untuk menampilkan halaman web, bisa dengan menggunakan fungsi *render_template*.

```
@app.route("/ubahsession")
def ubahsession():
    session['user'] = 'Accord'
    msg = "Session 'user' telah dirubah."
    return render_template("index.html", msg = msg)
```

└ Buatlah fungsi untuk mengubah nilai pada *variable session*, nilai pada *variable session* pada fungsi ini bisa dibedakan dengan nilai *session* yang ada pada fungsi *index*, supaya dapat diketahui apakah programnya bekerja atau tidak.

```
@app.route("/hapusession")
def hapusession():
    if "user" in session.keys():
        msg = "Session 'user' telah dihapus."
        session.pop("user", None)
    else:
        msg = "session 'user' tidak ditemukan."
    return render_template("index.html", msg = msg)
```

└ Buatlah fungsi untuk menghapus nilai pada *session*, kita perlu mengecek apakah *session* memang terdapat sebuah nilai atau tidak agar bisa menghindari error dengan menggunakan *if statement*. Untuk menghapus nilai dari *session* kita bisa menggunakan perintah *pop* pada *variable session*.

```
@app.route("/showsession")
def showsession():
    if "user" in session.keys():
        user = session["user"]
    else:
        user = None
    return render_template("session.html", user = user)
```

└ Buatlah fungsi untuk menampilkan nilai *session* dengan menggunakan *if statement* supaya bisa mengetahui apakah *session* terdapat sebuah nilai atau tidak, lalu *program* tersebut akan menampilkan nilai tersebut ke dalam halaman yang berbeda.

4. Kerjakan setiap poin berikut ini!

a. Buatlah sebuah *database* dengan nama sia_3nimanda dan dengan struktur tabel sebagai berikut!

Nama Atribut	Tipe Data	Panjang Data	Konfigurasi Tambahan
id_user_3nimanda	INT	3	PRIMARY KEY
nama_user_3nimanda	VARCHAR	30	NOT NULL
username_3nimanda	VARCHAR	30	NOT NULL
password_3nimanda	VARCHAR	50	NOT NULL

**3nimanda merupakan 3 digit terakhir nim anda (harusnya setiap mahasiswa tidak ada yang sama)*

```
CREATE TABLE `user` (
  `id_user_383` INT(3) NOT NULL AUTO_INCREMENT,
  `nama_user_383` VARCHAR(30) NOT NULL,
  `username_383` VARCHAR(30) NOT NULL,
  `password_383` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`id_user_383`)
);
```

#	Name	Datatype	Length/Set	Unsign...	Allow N...
1	id_user_383	INT	3	<input type="checkbox"/>	<input type="checkbox"/>
2	nama_user_383	VARCHAR	30	<input type="checkbox"/>	<input type="checkbox"/>
3	username_383	VARCHAR	30	<input type="checkbox"/>	<input type="checkbox"/>
4	password_383	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>

sia_383
 user_383

b. Tambahkan minimal 3 data pada tabel yang anda buat pada soal nomor 4!

```
INSERT INTO user_383 (nama_user_383, username_383, password_383)
VALUES ('Tarnished', 'tarnished96', 'toomanydeath');

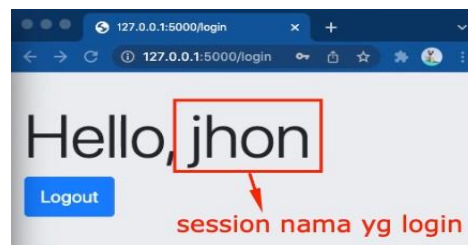
INSERT INTO user_383 (nama_user_383, username_383, password_383)
VALUES ('The Ashen One', 'ashenone4ever', 'hollowed');

INSERT INTO user_383 (nama_user_383, username_383, password_383)
VALUES ('Sekiro', 'wolf4kuro', 'shadowdiettwice');
```

*	id_user_383	nama_user_383	username_383	password_383
1	1	Tarnished	tarnished96	toomanydeath
2	2	The Ashen One	ashenone4ever	hollowed
3	3	Sekiro	wolf4kuro	shadowdiettwice

c. Kembangkan proses **login** dan **logout** dengan memanfaatkan flask, *database* mysql (soal a dan b) dan bootstrap!

d. Implementasikan penggunaan *session* pada proses login pada soal c sehingga nama *user* yang login ditampilkan pada halaman home seperti gambar berikut ini!



- Jawaban soal **C** dan **D**.

Source Codes

a. app.py

```
from flask import Flask, render_template, request, session
import mysql.connector

app = Flask(__name__)
app.config["SECRET_KEY"] = "123"
db = mysql.connector.connect (
    host = "localhost",
    user = "root",
    password = "",
    database = "sia_383"
)

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

@app.route("/login", methods=["GET", "POST"])
def login():
    if (request.method == "POST"):
        uname = request.form['uname']
        pwd = request.form['pwd']

        cur = db.cursor()
        cur.execute("select * from user_383 where username_383=%s", (uname,))
        user = cur.fetchone()
        cur.close()

        if (len(user) > 0):
            if (pwd == user[3]):
                session["uname"] = user[2]
                session['pwd'] = user[3]
                session['rname'] = user[1]
                return render_template("home.html")
            else:
                return "Username and Password are NOT Match"
        else:
            return "User with that Username are NOT Found"
    return render_template("index.html")

@app.route("/logout")
def logout():
    session.clear()
    return render_template("index.html")

if __name__ == "__main__":
    app.run(debug=True)
```

b. index.html

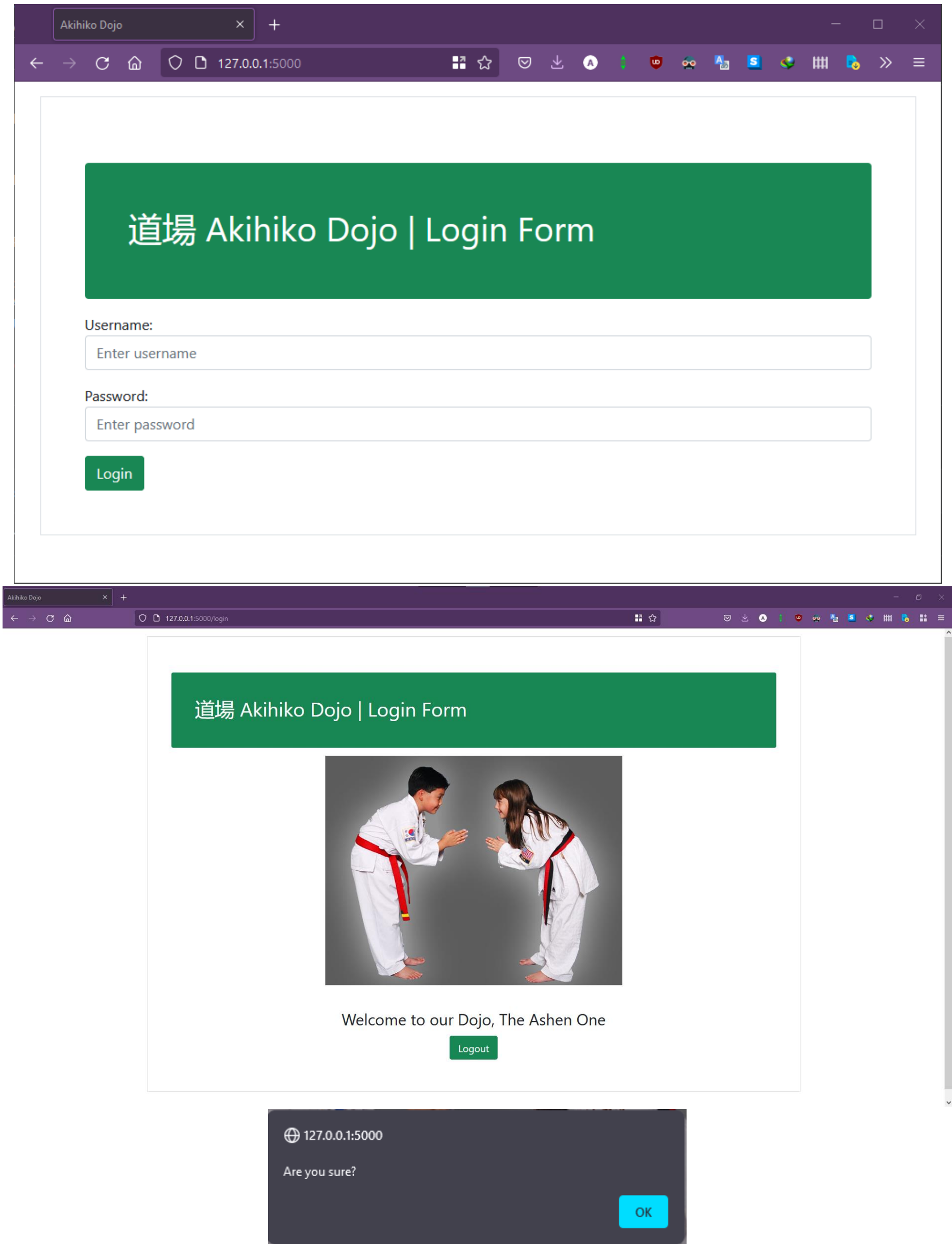
```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Akihiko Dojo</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
  <div class="container border p-5 my-5 mt-3">
    <div class="mt-4 p-5 bg-success text-white rounded">
      <h1>道場 Akihiko Dojo | Login Form</h1>
    </div>
    <form action="/login" method="post">
      <div class="mb-3 mt-3">
        <label for="username">Username:</label>
        <input type="text" class="form-control" id="username" placeholder="Enter username" name="uname">
      </div>
      <div class="mb-3">
        <label for="pwd">Password:</label>
        <input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pwd">
      </div>
      <button type="submit" class="btn btn-success">Login</button>
    </form>
  </div>
</body>
</html>
```

c. home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Akihiko Dojo</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

  <div class="container border p-5 my-5 mt-3">
    <div class="mt-4 p-5 bg-success text-white rounded">
      <h1>道場 Akihiko Dojo | Login Form</h1>
    </div>
    
    {% if session["rname"] %}
      <h2 class="text-center">Welcome to our Dojo, {{ session["rname"] }}</h2>
    {% else %}
      Not login
    {% endif %}
    <p class="lead text-center mt-3">
      <a class="btn btn-success btn-lg" href="/logout" role="button" onclick="alert('Are you sure?')">Logout</a>
    </p>
  </div>
</body>
</html>
```

Screenshot



e. Berikan penjelasan anda mengenai proses login dan logout tersebut, sehingga anda memahami semua alur program anda!

Cara kerja proses *login* dan *logout* aplikasi tersebut.

```
db = mysql.connector.connect (  
    host = "localhost",  
    user = "root",  
    password = "",  
    database = "sia_383"  
)
```

└ Pertama *Python* akan membuka koneksi ke database *MySQL*, yaitu *database* *sia_383*.

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

└ Pada saat aplikasi dijalankan, *Python* akan menampilkan halaman *login*, yaitu *index.html*

```
@app.route("/login", methods=["GET", "POST"])  
def login():
```

└ Ketika user menekan tombol pada halaman *index.html*, maka *Python* akan *me-routing* aplikasi tersebut untuk menjalankan fungsi *login*.

```
if (request.method == "POST"):  
    uname = request.form['uname']  
    pwd = request.form['pwd']  
  
    cur = db.cursor()  
    cur.execute("select * from user_383 where username_383=%s", (uname,))  
    user = cur.fetchone()  
    cur.close()  
  
    if (len(user) > 0):  
        . . .  
    else:  
        return "User with that Username are NOT Found"  
return render_template("index.html")
```

└ Di dalam fungsi tersebut terdapat percabangan, jika tombol yang ditekan mengirimkan method berupa *POST*, maka aplikasi akan mengambil data dari halaman tersebut yang berupa *username* dan *password*. Setelah itu aplikasi akan mengambil data dari *table* *user_383* dimana *username* yang berada di *database* sama dengan *username* yang dari halaman *html*.

```
if (len(user) > 0):  
    if (pwd == user[3]):  
        session["uname"] = user[2]  
        session['pwd'] = user[3]  
        session['rname'] = user[1]  
        return render_template("home.html")  
    else:  
        return "Username and Password are NOT Match"  
else:  
    return "User with that Username are NOT Found"
```

└ Jika *username* yang berada di *database* sama dengan *username* yang dari halaman *html* memang sama (*variable* *user* lebih dari 0), maka aplikasi akan membuat *session* yang berupa data-data dari *database*, yang akan ditampilkan ke halaman *home*, sedangkan jika *if statements* tersebut tidak sesuai, maka yang akan ditampilkan merupakan pesan *error*.

```
@app.route("/logout")  
def logout():  
    session.clear()  
    return render_template("index.html")
```

└ Setelah memasuki halaman *home*, user bisa menekan tombol *Log Out* untuk kembali ke halaman *login*. Di dalam fungsi aplikasi akan membersihkan data-data yang ada pada *session*, dan akan menampilkan halaman *index*.