

LAMPIRAN

1) *Main Script*

```
#run.py
import sqlite3
from sqlite3.dbapi2 import version_info
from flask import Flask, request, jsonify, render_template, redirect, flash
from flask.helpers import url_for
import paramiko
from paramiko.client import SSHClient
from librouteros.query import Key
from librouteros import connect
from datetime import datetime
import time
import os
import psutil

app = Flask(__name__)

@app.route('/base')
def base():
    return render_template('base.html')

@app.route('/')
def index():
    sqliteConnection = sqlite3.connect('routerDB.db')
    cursor = sqliteConnection.cursor()
    cursor.execute("select * from router_info")
    datatest=cursor.fetchall()
    sqliteConnection.commit()
    cursor.close()
    result_len2=len(datatest)
    return render_template('index.html',datatest=datatest,result_len2=result_len2)

@app.route('/tables', methods=["POST", "GET"])
def table():
    sqliteConnection = sqlite3.connect('routerDB.db')
    cursor = sqliteConnection.cursor()
    cursor.execute("select * from router_info")
    data=cursor.fetchall()
    result_len=len(data)
    return render_template('tables.html', data=data, result=result_len)

@app.route('/view/<no>', methods=["POST", "GET"])
def view(no):

    sqliteConnection = sqlite3.connect('routerDB.db')
    cursor = sqliteConnection.cursor()
    cursor.execute("select * from device_conf where serial_number='{ }'.format(no))
```

```

result = cursor.fetchall()
cursor.execute("select * from user_hotspot where serial_number='{ }'".format(no))
total_user = cursor.fetchall()
total_user = len(total_user)
return render_template('view.html',result=result, total_user=total_user)

@app.route('/update', methods=['POST','GET'])
def update():

    tstart = time.time()

    hostname = request.form['hostname']
    ssid = request.form['ssid']
    landingpage = request.form['landingpage']
    serial_number = request.form['sn']
    dns_name = request.form['dnsname']

    sqliteConnection = sqlite3.connect('routerDB.db')
    cursor = sqliteConnection.cursor()
    cursor.execute("update device_conf set hostname='{ }', ssid='{ }', hotspot_dns='{ }',
landing_page='{ }' where
serial_number='{ }'".format(hostname,ssid,dns_name,landingpage,serial_number))
    sqliteConnection.commit()

    cursor.execute("select * from router_info where
serial_number='{ }'".format(serial_number))
    login = cursor.fetchone()

    ssh_client = paramiko.SSHClient()
    ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh_client.connect(hostname=login[3], username=login[4], password=login[5],
allow_agent=False, look_for_keys=False, banner_timeout=200)

    config_list = [
        hostname,
        ssid,
        landingpage,
        dns_name
    ]
    config_param = [
        "system identity set name={ }",
        "interface wireless set ssid={ } wlan1",
        "ip hotspot profile set html-directory={ } hotspot1",
        "ip hotspot profile set dns-name={ } hotspot1"
    ]
    for x in range(len(config_param)):
        ssh_client.exec_command(config_param[x].format(config_list[x]))

    ssh_client.close()
    cursor.close()

```

```

tend = time.time()
ttime = tend - tstart

print("Processing Time : {0:.2f} second".format(ttime))
print("CPU Usage : { }%".format(psutil.cpu_percent(interval=1)))

return redirect(url_for('table'))

@app.route('/device/user/delete/<id>')
def delete_user(id):
    x = request.args['serial']
    y = request.args['user']
    sqliteConnection = sqlite3.connect('routerDB.db')
    cursor = sqliteConnection.cursor()
    cursor.execute("delete from user_hotspot where rowid={}".format(id))
    sqliteConnection.commit()
    cursor.execute("select * from router_info where serial_number='{}'.format(x)")
    login = cursor.fetchone()

    ssh_client = paramiko.SSHClient()
    ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh_client.connect(hostname=login[3], username=login[4], password=login[5],
allow_agent=False, look_for_keys=False, banner_timeout=200)

    config_param = "ip hotspot user remove {}".format(y)

    ssh_client.exec_command(config_param)

    ssh_client.close()
    cursor.close()
    flash("Employee Updated Successfully")

return redirect(url_for('device_user', no=x))

@app.route('/device/user/update/', methods=['POST','GET'])
def update_user():
    if request.method == 'POST':
        tstart = time.time()
        nomor = request.form['no']
        user = request.form['user']
        password = request.form['password']

        sqliteConnection = sqlite3.connect('routerDB.db')
        cursor = sqliteConnection.cursor()
        cursor.execute("update user_hotspot set user='{}', password='{}' where
nomor='{}'.format(user,password,nomor))
        sqliteConnection.commit()

```

```

flash("Employee Updated Successfully")
tend = time.time()
ttotal = tend - tstart

print("Processing Time : {0:.2f} second".format(ttime))
print("CPU Usage : { }%".format(psutil.cpu_percent(interval=1)))

return redirect(url_for('device_user', no=request.form['serial']))

@app.route('/device/user/<no>', methods=['POST','GET'])
def device_user(no):

    sqliteConnection = sqlite3.connect('routerDB.db')
    cursor = sqliteConnection.cursor()
    cursor.execute("select * from user_hotspot where serial_number='{ }'".format(no))
    datatest=cursor.fetchall()

    return render_template('user.html',data=datatest,no=no)

@app.route('/device/user/add', methods=['POST','GET'])
def adduser():

    tstart = time.time()

    nomor = request.form['nomor']
    user = request.form['user']
    password = request.form['pass']

    sqliteConnection = sqlite3.connect('routerDB.db')
    cursor = sqliteConnection.cursor()
    cursor.execute("insert into user_hotspot (serial_number,username,password) values
    ('{ }','{ }','{ }')".format(nomor,user,password))
    sqliteConnection.commit()

    cursor.execute("select * from router_info where serial_number='{ }'".format(nomor))
    login = cursor.fetchone()

    ssh_client = paramiko.SSHClient()
    ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh_client.connect(hostname=login[3], username=login[4], password=login[5],
    allow_agent=False, look_for_keys=False, banner_timeout=200)

    config_param = "ip hotspot user add name={ } password={ }"

    ssh_client.exec_command(config_param.format(user,password))

    ssh_client.close()
    cursor.close()

    tend = time.time()

```

```

ttime = tend - tstart

print("Processing Time : {0:.2f} second".format(ttime))
print("CPU Usage : { }%".format(psutil.cpu_percent(interval=1)))
return redirect(url_for('device_user', no=nomor))

@app.route('/test', methods=['POST','GET'])
def test():
    sqliteConnection = sqlite3.connect('na.db')
    cursor = sqliteConnection.cursor()

    if request.method == 'GET':
        cursor.execute("select * from coba")
        datatest=cursor.fetchall()
        return render_template('test.html',data=datatest)

    else:
        landingpage = request.form['landingpage']
        nomor = request.form['no']
        cursor.execute("update coba set landingpage='{}' where
no={}".format(landingpage, nomor))
        cursor.execute("select * from coba")
        datatest=cursor.fetchall()
        sqliteConnection.commit()
        cursor.close()
        return render_template('test.html',data=datatest)

@app.route('/testconfig', methods=['GET','POST'])
def test_config():
    dats = request.get_json()
    ip = dats['ip_router']
    username = 'admin'
    password = "

    api = connect(
        username='admin',
        password="",
        host=ip,
    )

    router_board_info = api.path('system/routerboard')
    tuple(router_board_info)
    time.sleep(0.2)

    for rbi in router_board_info:
        serial_number=rbi['serial-number']
        model=rbi['model']
        version=rbi['upgrade-firmware']
        time.sleep(0.2)

```

```

identity_info = api.path('system/identity')
tuple(identity_info)
time.sleep(0.2)

for ii in identity_info:
    identity=ii['name']
    time.sleep(0.2)

try:
    sqliteConnection = sqlite3.connect('routerDB.db')
    cursor = sqliteConnection.cursor()
    cursor.execute("INSERT INTO router_info values
('{}','{}','{}','{}','{}','{}')".format(serial_number,model,version,ip,username,password))
    sqliteConnection.commit()
    cursor.close()
except sqliteConnection.Error as error:
    print("OK")
finally:
    print("Sukses bro")
    time.sleep(0.2)

# connect to router using ssh
ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(hostname=ip, username=username, password=password,
allow_agent=False, look_for_keys=False, banner_timeout=200)

config_list = [
    'system identity set name=Hotspot-Client',
    'interface wireless enable wlan1',
    'interface wireless set mode=ap-bridge numbers=wlan1',
    'interface wireless set wlan1 ssid=Company-Hotspot',
    'ip firewall nat add chain=srcnat src-address=172.168.1.0/24 action=masquerade',
    'ip firewall nat add chain=srcnat out-interface=ether2 action=masquerade',
    'ip address add address=172.168.1.1/24 interface=wlan1',
    'ip pool add name=dhcp_hotspot ranges=172.168.1.2-172.168.1.254',
    'ip dhcp-server add address-pool=dhcp_hotspot disabled=no interface=wlan1
name=hotspot',
    'ip dhcp-server network add address=172.168.1.0/24 gateway=172.168.1.1',
    'ip hotspot profile add dns-name=default.net hotspot-address=172.168.1.1
name=hotspot1',
    'ip hotspot add address-pool=dhcp_hotspot disabled=no interface=wlan1
name=hotspot1 profile=hotspot1',
    'ip hotspot user add name=admin password=admin'
]

for config in config_list:
    ssh_client.exec_command(config)
    time.sleep(0.2)

```

```

hostname = "Hotspot-Client"
interface = "wlan1"
interface_mode = "ap-bridge"
ssid = "Company-Hotspot"
router_ip = "172.168.1.1/24"
dhcp_pool_name = "dhcp_hotspot"
dhcp_ip_range = "172.168.1.2-172.168.1.254"
dhcp_name = "hotspotX"
dhcp_net_add = "172.168.1.0/24"
hotspot_name = "hotspot1"
hotspot_dns = "default.net"
landing_page = "hotspot"
try:
    sqliteConnection = sqlite3.connect('routerDB.db')
    cursor = sqliteConnection.cursor()
    cursor.execute("INSERT INTO device_conf values
('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}').format(serial_number,interface,in
terface_mode,hostname,ssid,router_ip,dhcp_name,dhcp_pool_name,dhcp_ip_range,dh
cp_net_add,hotspot_name,hotspot_dns,landing_page))
    sqliteConnection.commit()
    cursor.close()
except:
    print("Already There")

scp = ssh_client.open_sftp()
data_loc = '/home/ihza/Desktop/coba/network-automation/static/landpage'
init_dir = ['/Style1','/Style2']
try:
    for init_d in init_dir:
        scp.mkdir(init_d)
        local_path = data_loc + init_d
        os.chdir(local_path)
        for file in os.listdir(local_path):
            temp1_loc_path = local_path
            temp1_rem_path = init_d
            os.chdir(temp1_loc_path)
            scp.chdir(temp1_rem_path)
            try:
                scp.put(file, file)
            except:
                folder = file
                scp.mkdir(folder)
                temp2_loc_path = os.path.join(local_path, folder)
                temp2_rem_path = os.path.join(init_d, folder)
                os.chdir(temp2_loc_path)
                scp.chdir(temp2_rem_path)
                for file1 in os.listdir(temp2_loc_path):
                    temp3_rem_path = temp2_rem_path
                    temp3_loc_path = temp2_loc_path

```

```

os.chdir(temp3_loc_path)
scp.chdir(temp3_rem_path)
try:
    scp.put(file1,file1)
except:
    folder1 = file1
    scp.mkdir(folder1)
    temp4_loc_path = os.path.join(temp3_loc_path,folder1)
    temp4_rem_path = os.path.join(temp3_rem_path,folder1)
    os.chdir(temp4_loc_path)
    scp.chdir(temp4_rem_path)
    for file2 in os.listdir(temp4_loc_path):
        try:
            scp.put(file2,file2)
        except:
            folder2 = file2
            scp.mkdir(folder2)
            temp5_rem_path = os.path.join(temp4_rem_path, folder2)
            temp5_loc_path = os.path.join(temp4_loc_path, folder2)
            os.chdir(temp5_loc_path)
            scp.chdir(temp5_rem_path)
            for file3 in os.listdir(temp5_loc_path):
                scp.put(file3,file3)

time.sleep(0.5)

scp.close()
except:
    print("Already Exist")

data = {'status': 'ok'}

ssh_client.close()

return jsonify(data)

#host dibawah adalah ip server ZTP
if __name__ == '__main__':
    app.secret_key = 'super secret key'
    app.run(host='172.16.10.1', debug=True)

```

2) Script Base Template (base.html)

```

<!DOCTYPE html>
<html lang="en">

<head>

```



```
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
<meta name="description" content="">
<meta name="author" content="">

<title>[- ZTP -]</title>

<!-- Custom fonts for this template-->
<link href="{{ url_for('static', filename='vendor/fontawesome-
free/css/all.min.css') }}" rel="stylesheet" type="text/css">
<link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600
,600i,700,700i,800,800i,900,900i"
rel="stylesheet">

<!-- Custom styles for this template-->
<link href="{{ url_for('static', filename='css/sb-admin-2.min.css') }}"
rel="stylesheet">

<!-- Modal CRUD -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-
Vko08x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh
" crossorigin="anonymous">

<!-- Custom styles for dataTable-->
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.22/css/jquery.dataTables.css">
```

```

<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.22/css/jquery.dataTables.min.css">
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.22/css/dataTables.bootstrap4.min.css">
<link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.5.2/css/bootstrap.css">

</head>

<body id="page-top">

<!-- Page Wrapper -->
<div id="wrapper">

<!-- Sidebar -->
<ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion"
id="accordionSidebar">

<!-- Sidebar - Brand -->
<a class="sidebar-brand d-flex align-items-center justify-content-center"
href="{{ url_for('index') }}">
<div class="sidebar-brand-icon rotate-n-15">
<i class="fas fa-angle-up"></i>
</div>
<div class="sidebar-brand-text mx-3">Cloud WiFi Dashboard</div>
</a>

<!-- Divider -->
<hr class="sidebar-divider my-0">

<!-- Nav Item - Dashboard -->
<li class="nav-item">
<a class="nav-link" href="{{ url_for('index') }}">

```

```

        <i class="fas fa-fw fa-tachometer-alt"></i>
        <span>Dashboard</span></a>
    </li>

    <!-- Divider -->
    <hr class="sidebar-divider">

    <!-- Heading -->
    <!-- <div class="sidebar-heading">
    Interface
    </div -->

    <!-- Nav Item - Pages Collapse Menu -->
    <li class="nav-item">
        <a class="nav-link" href="{{ url_for('table') }}">
            <i class="fa fa-history" aria-hidden="true"></i>
            <span>Provisioning List</span></a>
        </li>

    <!-- Divider -->
    <!-- <hr class="sidebar-divider"> -->

    <!-- Divider -->
    <!-- <hr class="sidebar-divider d-none d-md-block"> -->

    <!-- Sidebar Toggler (Sidebar) -->
</ul>

<!-- End of Sidebar -->

<!-- Content Wrapper -->
<div id="content-wrapper" class="d-flex flex-column">

```

```

<!-- Main Content -->
<div id="content">

    <!-- Topbar -->
    <nav class="navbar navbar-expand navbar-light bg-white topbar mb-4 static-
top shadow">

        <!-- Sidebar Toggle (Topbar) -->
        <button id="sidebarToggleTop" class="btn btn-link d-md-none rounded-
circle mr-3">
            <i class="fa fa-bars"></i>
        </button>

        <!-- Topbar Search -->
        <form
            class="d-none d-sm-inline-block form-inline mr-auto ml-md-3 my-2
my-md-0 mw-100 navbar-search">
            <div class="input-group">
                <input type="text" class="form-control bg-light border-0 small"
placeholder="Search for..."
                    aria-label="Search" aria-describedby="basic-addon2">
                <div class="input-group-append">
                    <button class="btn btn-primary" type="button">
                        <i class="fas fa-search fa-sm"></i>
                    </button>
                </div>
            </div>
        </form>

        <!-- Topbar Navbar -->
        <ul class="navbar-nav ml-auto">

            <!-- Nav Item - Search Dropdown (Visible Only XS) -->

```

```

<li class="nav-item dropdown no-arrow d-sm-none">
  <a class="nav-link dropdown-toggle" href="#" id="searchDropdown"
role="button"
    data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
    <i class="fas fa-search fa-fw"></i>
  </a>
  <!-- Dropdown - Messages -->
  <div class="dropdown-menu dropdown-menu-right p-3 shadow
animated--grow-in"
    aria-labelledby="searchDropdown">
    <form class="form-inline mr-auto w-100 navbar-search">
      <div class="input-group">
        <input type="text" class="form-control bg-light border-0
small"
          placeholder="Search for..." aria-label="Search"
          aria-describedby="basic-addon2">
        <div class="input-group-append">
          <button class="btn btn-primary" type="button">
            <i class="fas fa-search fa-sm"></i>
          </button>
        </div>
      </div>
    </form>
  </div>
</li>
<div class="topbar-divider d-none d-sm-block"></div>

<!-- Nav Item - User Information -->
<li class="nav-item dropdown no-arrow">
  <a class="nav-link dropdown-toggle" href="#" id="userDropdown"
role="button"

```

```

        data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
        <span class="mr-2 d-none d-lg-inline text-gray-600
small">Admin</span>
        
        </a>

    </li>
</ul>

</nav>
<!-- End of Topbar -->

<!-- Begin Page Content -->
{% block index %}{% endblock% }
{% block tables%}{% endblock% }
<!-- /.container-fluid -->

</div>
<!-- End of Main Content -->

<!-- Footer -->
<footer class="sticky-footer bg-white">
    <div class="container my-auto">
        <div class="copyright text-center my-auto">
            <span>Copyright &copy; Fariz, Ihza 2021</span>
        </div>
    </div>
</footer>
<!-- End of Footer -->

```

```

</div>
<!-- End of Content Wrapper -->

</div>
<!-- End of Page Wrapper -->

<!-- Scroll to Top Button-->
<a class="scroll-to-top rounded" href="#page-top">
  <i class="fas fa-angle-up"></i>
</a>

<!-- Bootstrap core JavaScript-->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript-->
<script src="vendor/jquery-easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->
<script src="js/sb-admin-2.min.js"></script>

<!-- Page level plugins -->
<script src="{{ url_for('static',
filename='vendor/chart.js/Chart.min.js') }}"></script>

<!-- Page level custom scripts -->
<script src="{{ url_for('static', filename='/demo/chart-area-demo.js') }}"></script>
<script src="{{ url_for('static', filename='js/demo/chart-pie-demo.js') }}"></script>

<!-- DataTable scripts -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script type="text/javascript" charset="utf8"
src="https://cdn.datatables.net/1.10.22/js/jquery.dataTables.js"></script>

```

```

<!-- Modal User Hotspot Update, Delete, Add -->
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-
J6qa4849bIE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script>

<script>
    var months = ["January", "February", "March", "April", "May", "June", "July",
    "August", "September", "October", "November", "December"];

    $(document).ready(function(){
        displayClock();
    });

    function displayClock(){
        var d=new Date();
        var zone=d.toString().match(/([\+-][0-9]+)s/)[1];
        $("#zone").text(('+zone+'));
        var tgl=months[d.getMonth()+1] +d.getDate()+", "+d.getFullYear();
        $("#tgl").text(tgl);
        var jam=d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();
        $("#jam").text(jam);

        setTimeout(displayClock,1000);
    }

```



```

</script>
<script>
    $(document).ready(function() {
        $('#c_info').DataTable();
    });
</script>
<script>
    $(document).ready(function() {
        $('#d_info').DataTable( {
            "lengthMenu": [[5, 10, 25, -1], [5, 10, 25, "All"]]
        });
    });
</script>
<script>
var x = document.getElementById("hostname").readOnly;
function myFunctionHostname(){
    if (x == true){
        document.getElementById("hostname").readOnly = false;
        x = false
    }else{
        document.getElementById("hostname").readOnly = true;
        x = true
    }
}
</script>
<script>
var x = document.getElementById("ssid").readOnly;
function myFunctionSSID(){
    if (x == true){
        document.getElementById("ssid").readOnly = false;
        x = false
    }else{
        document.getElementById("ssid").readOnly = true;
    }
}

```

```

        x = true
    }
}
</script>
<script>
    var x = document.getElementById("dnsname").readOnly;
    function myFunctionDNS(){
        if (x == true){
            document.getElementById("dnsname").readOnly = false;
            x = false
        }else{
            document.getElementById("dnsname").readOnly = true;
            x = true
        }
    }
</script>
<script>
    function changeImage() {
        var landingpage = document.getElementById('landingpage');
        var value = landingpage.options[landingpage.selectedIndex].value;
        var text = landingpage.options[landingpage.selectedIndex].text;
        var image = document.getElementById('myImage');
        if (value == 'hotspot') {
            image.src = "{{ url_for('static', filename='img/Hotspot.PNG') }}";
        }else if (value == 'Style1') {
            image.src = "{{ url_for('static', filename='img/Style1.PNG') }}";
        }else if (value == 'Style2') {
            image.src = "{{ url_for('static', filename='img/Style2.PNG') }}";
        }
    }
</script>
</body>
</html>

```

3) Script Index / Homepage (index.html)

```

{% extends "base.html" % }
{% block index % }
<div class="container-fluid">

  <!-- Page Heading -->
  <div class="d-sm-flex align-items-center justify-content-between mb-4">
    <h1 class="h3 mb-0 text-gray-800">Dashboard</h1>
  </div>

  <!-- Content Row -->
  <div class="row">
    <!-- Earnings (Monthly) Card Example -->
    <div class="col-xl-3 col-md-6 mb-4" style="margin: auto; width: 50%;">
      <div class="card border-left-primary shadow h-100 py-2">
        <div class="card-body">
          <div class="row no-gutters align-items-center">
            <div class="col mr-2">
              <div class="text-xs font-weight-bold text-primary text-uppercase mb-
1">
                Total Provisioning</div>
              <div class="h2 mb-0 font-weight-bold text-gray-
800">{{ result_len2 }}</div>
            </div>
            <div class="col-auto">
              <i class="fas fa-sitemap fa-2x text-gray-300"></i>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="row">
      {% for i in tmp % }

        <div class="col-xl-3 col-md-6 mb-4" style="margin: auto; width: 50%;">
          <div class="card border-left-primary shadow h-100 py-2">
            <div class="card-body">
              <div class="row no-gutters align-items-center">
                <div class="col mr-2">
                  <div class="text-xs font-weight-bold text-primary text-uppercase
mb-1">
                    {{ i[0] }} | <br>
                    User Hotspot</div>
                  <div class="h2 mb-0 font-weight-bold text-gray-
800">{{ i[1] }}</div>
                </div>
                <div class="col-auto">

```

```

        <i class="fas fa-sitemap fa-2x text-gray-300"></i>
    </div>
</div>
</div>
</div>
    <div>
        { % endfor % }
    </div>

<!-- Content Row -->

<!-- Content Row Start -->
<!-- Content Chart Start -->
<div class="row">

<div style="margin: auto; width: 50%;">
    <div class="card border-left-primary shadow h-100 py-2">
        <div class="card-body">
            <div class="row no-gutters align-items-center">
                <div class="col mr-2">
                    <div class="text-xs font-weight-bold text-primary text-uppercase mb-
1">
                        ACTIVE USER</div>
                    <canvas id="pieChart" style="max-width: 750px;"></canvas>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
<!-- Content Chart End -->

<!-- Area Chart -->
<div>
    <div class="card shadow mb-4 col-xl-12 col-lg-7">
        <!-- Card Header - Dropdown -->
        <div
            class="card-header py-3 d-flex flex-row align-items-center justify-content-
between">
            <h6 class="m-0 font-weight-bold text-primary">Recent Devices</h6>
        </div>
        <!-- Card Body -->
        <div class="table-responsive">
            <table class="table table-bordered" id="d_info" width="100%"
cellspacing="0">
                <thead>
                    <tr>
                        <th>Serial Number</th>
                        <th>Model</th>

```

```

        <th>Firmware</th>
        <th colspan="2" style="text-align: center;">Action</th>
    </tr>
</thead>
<tbody>
    {% for item in datatest %}
    <tr>
        <td>{{ item[0] }}</td>
        <td>{{ item[1] }}</td>
        <td>{{ item[2] }}</td>
        <td style="text-align: center;"><a class="null"
href="{{ url_for('view',no=item.0) }}">Detail</td>
        <td style="text-align: center;"><a class="null"
href="{{ url_for('device_user',no=item.0) }}">User Hotspot</td>
    </tr>
    {% endfor %}
</tbody>
</table>
</div>
</div>
</div>
</div>
{% endblock %}

{% block script %}
<script>
    // const serialnumber = ['{{ datuse[0] }}'];
    const temp = [{{ datuse|safe }}];
    const temp2 = [{{ users|safe }}];
    const dats = [];
    const serialnumber = [];

    for (let i = 0; i < temp[0].length; i++){
        serialnumber.push(temp[0][i]);
        dats.push(temp2[0][i])
    }

    var ctxP = document.getElementById("pieChart").getContext('2d');
    var myPieChart = new Chart(ctxP, {
        type: 'pie',
        data: {
            labels: serialnumber,
            datasets: [{
                data: dats,
                backgroundColor: ["#F7464A", "#46BFBD", "#FDB45C", "#949FB1",
"#4D5360"],
                hoverBackgroundColor: ["#FF5A5E", "#5AD3D1", "#FFC870",
"#A8B3C5", "#616774"]
            }]
        }
    });
</script>
</div>
</div>
</div>
</div>

```

```

    },
    options: {
        responsive: true
    }
});
</script>
{% endblock%}

```

4) *Script Provisioning List Device* (tables.html)

```

{% extends "base.html" %}
{% block tables %}
<div class="container-fluid">
    <div class="d-sm-flex align-items-center justify-content-between mb-4">
        <h1 class="h3 mb-0 text-gray-800"><b><i>Provisioning Devices</i></b></h1>
    </div>
</div>
<div class="card shadow mb-4 col-xl-12 col-lg-7">
    <div class="card-header py-3">
        <!-- <h6 class="m-0 font-weight-bold text-primary">Master Data</h6> -->
    </div>
    <div class="card-body">
        <div class="table-responsive">
            <table class="table table-bordered" id="c_info" width="100%"
cellspacing="0">
                <thead>
                    <tr>
                        <th>Serial Number</th>
                        <th>Model</th>
                        <th>Firmware</th>
                        <th colspan="2" style="text-align: center;">Action</th>
                    </tr>
                </thead>
                <tbody>
                    {% for item in data %}
                    <tr>
                        <td>{{ item.0 }}</td>
                        <td>{{ item[1] }}</td>
                        <td>{{ item[2] }}</td>
                        <td style="text-align: center;"><a class="null" href="{{ url_for('view',
no=item.0) }}">Detail</td>
                        <td style="text-align: center;"><a class="null"
href="{{ url_for('device_user',no=item.0) }}">User Hotspot</td>
                    </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
    </div>
</div>

```

```
</div>
{% endblock % }
```

5) *Script User Hotspot Page (user.html)*

```
{% extends 'base.html' % }

{% block index % }
<div class="container-fluid">
  <div class="d-sm-flex align-items-center justify-content-between mb-4">
    <h1 class="h3 mb-0 text-gray-800"><b><i>User Hotspot</i></b></h1>
  </div>
</div>
<div class="container">
  <div class="row">
    <div class="col md-12">

      <div class="card shadow mb-4 col-xl-12 col-lg-7">

        <!-- <div class="card-header py-3">
          <h6 class="m-0 font-weight-bold text-primary">User Hotspot</h6>
        </div> -->
        <div style="margin-top:10px; margin-bottom: 2px;">
          <h1><button type="button" class="btn btn-success float-right" data-
toggle="modal" data-target="#myModal">Add User</button></h1>

        </div>

        {% with messages = get_flashed_messages() % }

        {% if messages % }

        {% for message in messages % }

          <div class="alert alert-success alert-dismissible" role="alert">
            <button type="button" class="close" data-dismiss="alert" aria-
label="close">
              <span aria-hidden="true">x</span>
            </button>
            {{ message }}
          </div>

          {% endfor % }
          {% endif % }
          {% endwith % }

          <table class="table table-hover table-bordered">

            <tr>
```

```

<!-- <th>ID</th> -->
<th>Username</th>
<th>Password</th>
<th colspan="2">Action</th>
</tr>

{% for row in data %}
<tr>
<!-- <td>{{ row.0 }}</td> -->
<td>{{ row.2 }}</td>
<td>{{ row.3 }}</td>
<td>
<a href="/device/user/update/{{ row.0 }}" class="btn btn-warning btn-
xs" data-toggle="modal" data-target="#modaledit{{ row.id }}">Edit</a>
<!-- <a href="/device/user/delete/{{ row.0 }}" class="btn btn-danger
btn-xs" onclick="return confirm('Are You Sure To Delete ?')>Delete</a> -->
</td>
<td>
<a href="{{ url_for('delete_user', id=row.0, serial=row.1,
user=row.2) }}" class="btn btn-danger btn-xs" onclick="return confirm('Are You Sure
To Delete ?')>Delete</a>
<!-- <a href="{{ url_for('delete_user', id=row.0) }}" class="btn btn-
danger btn-xs" onclick="return confirm('Are You Sure To Delete ?')>Delete</a> -->
</td>
</tr>

<!-- Modal Edit Employee-->
<div id="modaledit{{ row.id }}" class="modal fade" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h4 class="modal-title">Update Information</h4>
</div>
<div class="modal-body">
<form action="{{ url_for('update_user') }}" method="POST">
<div class="form-group">
<label>Name:</label>
<input type="hidden" name="no" value="{{ row.0 }}">
<input type="hidden" name="serial" value="{{ row.1 }}">
<input type="text" class="form-control" name="user" value="{{ row.2 }}">
</div>
<div class="form-group">
<label>Email:</label>
<input type="text" class="form-control" name="password"
value="{{ row.3 }}">
</div>
<!-- <div class="form-group">
<label>Phone:</label>
<input type="text" class="form-control" name="phone"
value="{{ row.phone }}">

```



```

        </div> -->
        <div class="form-group">
            <button class="btn btn-primary" type="submit">Update</button>
        </div>
    </form>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
</div>
</div>
</div>
</div>
        {% endfor %}
    </table>
</div>

    <!-- Modal Add Employee-->
<div id="mymodal" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h4 class="modal-title">Add User Hotspot</h4>
            </div>
            <div class="modal-body">
                <form action="{ {url_for('adduser')}}" method="POST">
                    <div class="form-group">
                        <input type="hidden" class="form-control" name="nomor" required="1"
value={ {no}} >
                    </div>
                    <div class="form-group">
                        <label>User:</label>
                        <input type="text" class="form-control" name="user" required="1">
                    </div>
                    <div class="form-group">
                        <label>Password:</label>
                        <input type="password" class="form-control" name="pass" required="1">
                    </div>
                    <div class="form-group">
                        <button class="btn btn-primary" type="submit">Add</button>
                    </div>
                </form>
            </div>
            <div class="modal-footer">

                <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>

            </div>
        </div>
    </div>

```

```

    </div>
  </div>
</div>
  </div>
</div>
</div>
{% endblock %}

```

6) Script View / Detail Page

```

{% extends 'base.html' %}

{% block index %}

<div class="container-fluid">
  <div class="d-sm-flex align-items-center justify-content-between mb-4">
    {% for item in result %}
      <h1 class="h3 mb-0 text-gray-800">Device Details | <b><i>SN :
    {{ item.0 }}</i></b></h1>
    </div>
  </div>

<div>
  <!-- Area Chart -->
  <!-- <div class="col-xl-8 col-lg-7"> -->
  <div>
    <div class="card shadow mb-4 col-xl-12 col-lg-7">
      <!-- Card Header - Dropdown -->
      <div
        class="card-header py-3 d-flex flex-row align-items-center justify-content-
        between">
        <!-- <h6 class="m-0 font-weight-bold text-primary">Custom
        Configuration</h6> -->
      </div>
      <!-- Card Body -->
      <div style="margin-left: 25px; padding: 10px;">
        <form action="{{ url_for('update') }}" , method="POST">
          <div class="form-group">
            <h4>Router Information</h4>
            <div class="form-group row">
              <label for="inputEmail3" class="col-sm-2 col-form-label">Wireless
            Interface</label>
              <div class="col-sm-10">
                <input class="hidden" id="" name="int" value="{{ item.1 }}"
            readonly>
                <input type="hidden" id="" name="sn" value="{{ item.0 }}">
              </div>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>

```

```

<div class="form-group row">
  <label for="inputEmail3" class="col-sm-2 col-form-label">Wireless
Mode</label>
  <div class="col-sm-10">
    <input class="hidden" id="" name="intMode" value="{{item.2}}"
readonly>
  </div>
</div>
<div class="form-group row">
  <label for="inputEmail3" class="col-sm-2 col-form-label">Router
Hostname</label>
  <div class="col-sm-10">
    <input class="hidden" id="hostname" name="hostname"
value="{{item.3}}" readonly>
    <!-- <input class="form-control hidden" id="inputEmail3"
name="ssid" value="{{item.2}}" readonly> -->
    <i type="button" onclick="myFunctionHostname()" class="fas
fa-pen"></i>
    <!-- <button type="button" onclick="myFunction()" i class="fas
fa-pen"></button> -->
  </div>
</div>
<div class="form-group row">
  <label for="inputEmail3" class="col-sm-2 col-form-
label">SSID</label>
  <div class="col-sm-10">
    <input class="hidden" id="ssid" name="ssid" value="{{item.4}}"
readonly>
    <i type="button" onclick="myFunctionSSID()" class="fas fa-
pen"></i>
  </div>
</div>
<div class="form-group row">
  <label for="inputEmail3" class="col-sm-2 col-form-label">Router
IP Address</label>
  <div class="col-sm-10">
    <input class="hidden" id="ipadd" name="ipadd"
value="{{item.5}}" readonly>
  </div>
</div>
</div>
<div class="form-group">
  <h4>DHCP Server</h4>
  <!-- <div class="form-group row">
    <label for="inputEmail3" class="col-sm-2 col-form-label">DHCP
Server Name</label>
    <div class="col-sm-10">
      <input class="hidden" id="" name="" value="{{item.6}}"
readonly>
    </div>
  </div>

```

```

</div> -->
<div class="form-group row">
  <label for="inputEmail3" class="col-sm-2 col-form-label">DHCP
Pool Name</label>
  <div class="col-sm-10">
    <input class="hidden" id="" name="" value="{{ item.7 }}"
readonly>
  </div>
</div>
<div class="form-group row">
  <label for="inputEmail3" class="col-sm-2 col-form-label">IP
Range</label>
  <div class="col-sm-10">
    <input class="hidden" id="iprange" name="iprange"
value="{{ item.8 }}" readonly>
  </div>
</div>
<div class="form-group row">
  <label for="inputEmail3" class="col-sm-2 col-form-
label">Network Address</label>
  <div class="col-sm-10">
    <input class="hidden" id="netadd" name="netadd"
value="{{ item.9 }}" readonly>
  </div>
</div>
</div>
<div class="form-group">
  <h4>Hotspot</h4>
  <div class="form-group row">
    <label for="inputEmail3" class="col-sm-2 col-form-label">Profile
Name</label>
    <div class="col-sm-10">
      <input class="hidden" id="profilename" name="profilename"
value="{{ item.10 }}" readonly>
    </div>
  </div>
  <div class="form-group row">
    <label for="inputEmail3" class="col-sm-2 col-form-label">DNS
Name</label>
    <div class="col-sm-10">
      <input class="hidden" id="dnsname" name="dnsname"
value="{{ item.11 }}" readonly>
      <i type="button" onclick="myFunctionDNS()" class="fas fa-
pen"></i>
    </div>
  </div>
  <div class="form-group row">
    <label for="inputEmail3" class="col-sm-2 col-form-label">Active
User</label>
    <div class="col-sm-10">

```

```

        <input class="hidden" id="activeuser" name="activeuser"
value="{{ total_user }}" readonly>
    </div>
</div>
<!-- <div class="form-group row">
    <label for="inputEmail3" class="col-sm-2 col-form-
label">Hotspot User</label>
    <div class="col-sm-10">
        <input class="hidden" id="inputEmail3" placeholder="Email">
    </div>
</div -->
<!-- <div class="form-group row">
    <label for="inputEmail3" class="col-sm-2 col-form-label">IP
Address</label>
    <div class="col-sm-10">
        <input class="hidden" id="hs_ipadd" placeholder="Email">
    </div>
</div -->
</div>
<div class="form-group">
    <h4>Landing Page</h4>
    <div style="margin: 5px;">
        {% if item.12 == 'hotspot' %}
        

        {% elif item.12 == 'Style1' %}
        

        {% elif item.12 == 'Style2' %}
        

        {% endif %}
    </div>
    <label for="">Pilih :</label>
    <select id="landingpage" name="landingpage"
onclick="changeImage()">
        <option value="{{ item.12 }}">{{ item.12 }}</option>
        <option value="hotspot">hotspot</option>
        <option value="Style1">Style1</option>
        <option value="Style2">Style2</option>
    </select>
</div>
{% endfor %}
<input type="submit", value="Confirm" class="btn btn-cancel btn-xs">

```

```

        <a href="{{url_for('table')}}" class="btn btn-test btn-xs">Cancel</a>
    </form>
</div>
</div>
<div class="card shadow mb-4 col-xl-12 col-lg-7">
    <!-- Card Header - Dropdown -->
    <!-- Card Body -->
    <div>
        <form action="POST">
            </form>
        </div>
    </div>
</div>
</div>
</div>
{% endblock% }

```

7) *Script VPN Trap*

```

import os
import time
import requests

website = "http://172.16.10.1:5000/testconfig"
old_ip_vpn = ""
while True:
    try:
        recent = os.popen("last -1 | grep ppp")
        recent_ip_vpn = recent.read()
        data_ip_vpn = recent_ip_vpn.split()
        parameter = data_ip_vpn[9]
        ip_vpn = data_ip_vpn[2]

        if ip_vpn != old_ip_vpn and parameter == 'no':
            ip_router = {"ip_router" : ip_vpn}
            data_send = requests.post(url = website, json = ip_router)
            old_ip_vpn = ip_vpn
            print("DONE")
        else:
            print("Jika tidak ada Client baru")
            print("Client Lama : " + old_ip_vpn)
    except:
        print("Kosong")

    time.sleep(1)

```