

```

from flask import Flask, request, render_template, render_template_string,
jsonify
import json, os, binascii
import urllib
import urllib.request

app = Flask(__name__)
app.config['DEBUG'] = True
app.config['SECRET_KEY'] = binascii.hexlify(os.urandom(32)).decode()
role = "user_default"

def transfer(index, dest):
    for k, v in index.items():
        if hasattr(dest, '__getitem__'):
            if dest.get(k) and type(v) == dict:
                transfer(v, dest.get(k))
            else:
                dest[k] = v
        elif hasattr(dest, k) and type(v) == dict:
            transfer(v, getattr(dest, k))
        else:
            setattr(dest, k, v)

class HiddenTaskRunner(object):
    def __init__(self):
        pass
    def varlify(self):
        return vars(self)

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

@app.route("/secret-task-manager", methods=['GET', 'POST'])
def runtask():
    if request.method == 'GET':
        return render_template_string("<h1>Nothing's here. Go away!</h1>")
    else:
        req = json.loads(request.data)
        task = HiddenTaskRunner()
        transfer(req, task)
        res = task.varlify()
        #LKS Judge : Do not modify this role! This role is crucial to
determine the service uptime!
        if role == "lks-winner":
            if req['task']:
                # TL;DR developer will update this later
                os.system(req['task'])

```

```

        tpl = "Request granted."
    else:
        tpl = "No action defined."
    else:
        tpl = "You're only {}".format(role)

    #LKS Judge : Do not delete the message!
    res['message'] = tpl
    return jsonify(res)

@app.route("/welcome",methods=['GET'])
def exp():
    name = request.args.get('name')
    tpl = "<h1>Welcome to LKSN 2023, {}</h1>".format(name)
    if name != None:
        return render_template_string(tpl)
    else:
        return render_template_string("<h1>Welcome to LKSN 2023, User!</h1>")

@app.route("/safedomainopener",methods=['GET'])
def safe():
    url = request.args.get('url')
    if url != None:
        uri = urllib.parse.unquote(url)
        return urllib.request.urlopen(uri).read().decode('utf-8')
    else:
        return render_template_string("<h1>Please enter the GET['url']
parameter to continue</h1>")

@app.errorhandler(404)
def error(e):
    return render_template("404.html")

if __name__ == '__main__':
    app.run()

```