```
from flask import Flask, render_template, request
from scipy.misc import imsave, imread, imresize
import numpy as np
import keras.models
import re
import base64

import sys
import os
sys.path.append(os.path.abspath("./model"))
from load import *

app = Flask(__name__)
global model, graph
model, graph = init()

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

@app.route('/predict/', methods=['GET','POST'])
def predict():
    # get data from drawing canvas and save as image
    parseImage(request.get_data())

    # read parsed image back in 8-bit, black and white mode (L)
    x = imread('output.png', mode='L')
    x = np.invert(x)
    x = imresize(x,(28,28))

    # reshape image data for use in neural network
    x = x.reshape(1,28,28,1)
    with graph.as_default():
        out = model.predict(x)
```

```
        print(out)
        print(np.argmax(out, axis=1))
        response = np.array_str(np.argmax(out, axis=1))
        return response

def parseImage(imgData):
    # parse canvas bytes and save as output.png
    imgstr = re.search(b'base64,(.*)', imgData).group(1)
    with open('output.png','wb') as output:
        output.write(base64.decodebytes(imgstr))

if __name__ == '__main__':
    app.debug = True
    port = int(os.environ.get("PORT", 5000))
    app.run(host='0.0.0.0', port=port)
```