

```

1  from flask import Flask, render_template, redirect, url_for, request
2  import requests
3
4  app = Flask(__name__)
5
6  @app.route("/", methods = ['POST', 'GET'])
7  def index():
8      if request.method == 'POST':
9          arr = []
10         for i in request.form:
11             val = request.form[i]
12             if val == '':
13                 return redirect(url_for("demo2"))
14             arr.append(float(val))
15
16         # deepcode ignore HardcodedNonCryptoSecret: <please specify a reason of ignoring this>
17         API_KEY = "wf8mge_QQdwVO8ao2kmlCtfx0FLWl8442SH44V85v2Ls"
18         token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={
19             "apikey": API_KEY,
20             "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'
21         })
22         mltoken = token_response.json()["access_token"]
23         header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
24         payload_scoring = {
25             "input_data": [{"fields": [ 'GRE Score',
26                                     'TOEFL Score',
27                                     'University Rating',
28                                     'SOP',
29                                     'LOR ',
30                                     'CGPA',
31                                     'Research'],
32                             "values": [arr]
33             }
34         }
35
36         response_scoring = requests.post(
37             'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/8308fd4c-24a5-46ab-96fa-263657ae4ad0/predictions?version=2022-10-18',
38             json=payload_scoring,
39             headers=header
40         ).json()
41

```

```
36     response_scoring = requests.post(
37         'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/8308fd4c-24a5-46ab-96fa-263657ae4ad0/predictions?version=2022-10-18',
38         json=payload_scoring,
39         headers=header
40     ).json()
41
42     result = response_scoring['predictions'][0]['values']
43
44     if result[0][0] > 0.5:
45         return redirect(url_for('chance', percent=result[0][0]*100))
46     else:
47         return redirect(url_for('no_chance', percent=result[0][0]*100))
48 else:
49     return redirect(url_for("demo2"))
50
51 @app.route("/home")
52 def demo2():
53     return render_template("demo2.html")
54
55 @app.route("/chance/<percent>")
56 def chance(percent):
57     return render_template("chance.html", content=[percent])
58
59 @app.route("/nochance/<percent>")
60 def no_chance(percent):
61     return render_template("noChance.html", content=[percent])
62
63 @app.route('/<path:path>')
64 def catch_all():
65     return redirect(url_for("demo2"))
66
67 if __name__ == "__main__":
68     app.run()
```