

```

1 from tensorflow.keras.models import load_model
2 from tensorflow.keras.preprocessing import image
3 import numpy as np
4 import os
5 from flask import Flask,render_template,request,jsonify,redirect
6 import pyrebase
7 firebaseConfig = {
8     "apiKey": "AIzaSyA6DB5LsOP3SZtmwZtELAvS067bepWv8Hg",
9     "authDomain": "ibm-natural-disaster.firebaseio.com",
10    "databaseURL": "https://ibm-natural-disaster-default-rtdb.firebaseio.com",
11    "projectId": "ibm-natural-disaster",
12    "storageBucket": "ibm-natural-disaster.appspot.com",
13    "messagingSenderId": "289608247253",
14    "appId": "1:289608247253:web:3781d7e4fed3cc9e947f5c",
15    "measurementId": "G-TMHEYWJF1B"
16 }
17 firebase=pyrebase.initialize_app(firebaseConfig)
18 auth=firebase.auth()
19
20 app=Flask(__name__,template_folder="templates")
21
22 model=load_model("Disasters.h5")
23
24 @app.route("/")
25 def home():
26     return render_template("home.html")
27
28 @app.route("/loginnn")
29 def beginn():
30     return render_template("index.html")
31
32 @app.route("/uploading",methods=["GET","POST"])
33 def up():
34     if request.method=="POST":
35         f=request.files["image"]
36         basepath=os.path.dirname(__file__)
37         filepath=os.path.join(basepath,"uploads",f.filename)
38         f.save(filepath)
39         img=image.load_img(filepath,target_size=(64,64))
40         print(img)
41         x=image.img_to_array(img)
42         x=np.expand_dims(x,axis=0)
43         op=['Cyclone', 'Earthquake', 'Flood', 'Wildfire']
44         pred=np.argmax(model.predict(x))
45         return render_template("uploading.html",result=op[pred])
46         # return send_file(op[pred], mimetype='image/gif')
47         # return op[pred]
48
49     if request.method=="GET":
50         return render_template("uploading.html")
51
52 @app.route("/signup",methods=["get","post"])
53 def gg():
54     if request.method=="GET":
55         return render_template("signup.html")
56     if request.method=="POST":
57         emailid=request.form["emailid"]
58         password=request.form["password"]
59         conf_password=request.form["conpassword"]
60         if password ==conf_password:
61             user=auth.create_user_with_email_and_password(emailid,password)
62             return render_template("index.html", msg="successfully registered")
63         else:
64             return render_template("signup.html",error="password and conform
password not match")
65
66
67
68
69
70 @app.route("/login",methods=["post"])
71 def loginn():
72
73     emailid=request.form["emailid"]
74     password=request.form["password"]
75     # user=auth.create_user_with_email_and_password(emailid,password)
76     try:
77         user=auth.sign_in_with_email_and_password(emailid,password)
78     except:
79         return render_template("index.html",msg="failed to login")
80     return redirect("/uploading")
81

```

```
82
83
84
85 # port = os.getenv('VCAP_APP_PORT', '8080')
86
87 if __name__ == "__main__":
88     # app.secret_key = os.urandom(12)
89     app.run(debug=True, host='0.0.0.0', port=5000 )
90
```