# PROJECT DEVELOPMENT PHASE

**Sprint - IV**

|  |  |
| --- | --- |
| **Date** | 07-Nov-2022 |
| **Team ID** | PNT2022TMID12521 |
| **Project Name** | Developing a Flight Delay Model Using Machine Learning |
| **Maximum Marks** | 8 Marks |

# Integration the Deployed Model with Flask

**Web Application**

**App.py**

from flask import Flask, render\_template, request import requests

import requests

# NOTE: you must manually set API\_KEY below using information retrieved from your IBM Cloud account.

import requests

# NOTE: you must manually set API\_KEY below using information retrieved from your IBM Cloud account.

API\_KEY = "gyOvc0l0Hde4zdTmNc47N4Vh1zmMTFh7FlK8BEcKPADB" token\_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API\_KEY, "grant\_type": 'urn:ibm:params:oauth:grant-type:apikey'})

mltoken = token\_response.json()["access\_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken} import mysql.connector

app = Flask( name )

conn=mysql.connector.connect(host="localhost", user="root", password="", database="login") cursor=conn.cursor()

@app.route('/') def index():

return render\_template('index.html')

@app.route('/login')

def login(): # put application's code here return render\_template('login.html')

@app.route('/register') def register():

return render\_template('register.html')

@app.route('/home') def home():

return render\_template('home.html')

@app.route('/service') def service():

return render\_template('service.html')

@app.route('/about') def about():

return render\_template('about.html')

@app.route('/login\_validation', methods=['POST']) def login\_validation():

email=request.form.get('email') password=request.form.get('password')

cursor.execute("""SELECT \* FROM `users` WHERE `email` LIKE'{}' AND `password` LIKE '{}'""".format(email,password))

users = cursor.fetchall()

if len(users)>0:

return render\_template('home.html') else:

return render\_template('login.html', prediction\_text = "1" )

@app.route('/add\_user', methods=['POST']) def add\_user():

name= request.form.get('name') email = request.form.get('email')

password = request.form.get('password')

cursor.execute("""INSERT INTO `users`(`id`, `name`, `email`, `password`) VALUES (NULL,'{}','{}','{}')""".format(name,email,password))

conn.commit()

return render\_template('login.html', prediction\_text = "0")

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

year = request.form['year'] month = request.form['month'] day = request.form['day'] carrier = request.form['carrier'] origin = request.form['origin'] dest = request.form['dest']

if (carrier=="UA"): carrier=11

if (carrier=="AA"): carrier=1

if (carrier=="B6"): carrier=3

if (carrier=="DL"): carrier=4

if (carrier=="EV"): carrier=8

if (carrier=="MQ"): carrier=9

if (carrier=="US"): carrier=12

if (carrier=="WN"): carrier=14

if (carrier=="VX"): carrier=13

if (carrier=="FL"): carrier=7

if (carrier=="AS"): carrier=2

if (carrier=="9E"): carrier=0

if (carrier=="F9"): carrier=9

if (carrier=="HA"): carrier=4

if (carrier=="OO"): carrier=5

if (carrier=="YV"): carrier=15

if (origin=="EWR"): origin=0

if (origin=="LGA"): origin=2

if (origin=="JFK"): origin=1

if (dest=="ATL"): dest=4

if (dest=="IAH"): dest=43

if (dest=="MIA"): dest=57

if (dest=="BQN"): dest=12

if (dest=="ORD"): dest=68

if (dest=="FLL"): dest=35

if (dest=="IAD"): dest=42

if (dest=="MCO"): dest=53

if (dest=="PBI"): dest=70

if (dest=="TPA"): dest=99

if (dest=="LAX"): dest=49

if (dest=="SFO"): dest=89

if (dest=="DFW"): dest=30

if (dest=="BOS"): dest=11

if (dest=="LAS"):

dest=48

if (dest=="MSP"): dest=60

if (dest=="DTW"): dest=32

if (dest=="RSW"): dest=82

if (dest=="SJU"): dest=91

if (dest=="PHX"): dest=73

if (dest=="BWI"): dest=16

if (dest=="CLT"): dest=23

if (dest=="BOS"): dest=11

if (dest=="BUF"): dest=14

if (dest=="DEN"): dest=29

if (dest=="SNA"): dest=94

if (dest=="LAS"): dest=48

if (dest=="MSY"): dest=61

if (dest=="SLC"): dest=92

if (dest=="SEA"): dest=88

if (dest=="ROC"): dest=99

if (dest=="ATL"): dest=4

if (dest=="DCA"): dest=33

if (dest=="RDU"): dest=4

if (dest=="BNA"): dest=4

if (dest=="CLE"): dest=88

if (dest=="STL"): dest=82

if (dest=="MDW"): dest=99

if (dest=="CVG"): dest=68

if (dest=="CMH"): dest=68

if (dest=="CHS"): dest=99

if (dest=="PIT"): dest=1

if (dest=="SAN"): dest=82

if (dest=="MKE"): dest=11

if (dest=="JAX"): dest=88

if (dest=="BTV"): dest=4

if (dest=="AUS"): dest=23

if (dest=="RIC"): dest=64

if (dest=="PWM"): dest=83

if (dest=="HOU"): dest=89

if (dest=="IND"): dest=47

if (dest=="MCI"): dest=80

if (dest=="SYR"): dest=78

if (dest=="BWI"): dest=4

if (dest=="MEM"): dest=23

if (dest=="PHL"): dest=14

if (dest=="GSO"): dest=96

if (dest=="ORF"): dest=23

if (dest=="DAY"): dest=57

if (dest=="PDX"):

dest=83

if (dest=="SRQ"): dest=91

if (dest=="SDF"): dest=29

if (dest=="XNA"): dest=88

if (dest=="MHT"): dest=43

if (dest=="BDL"): dest=23

if (dest=="OMA"): dest=4

if (dest=="GSP"): dest=57

if (dest=="SAV"): dest=28

if (dest=="GRR"): dest=16

if (dest=="HNL"): dest=24

if (dest=="SAT"): dest=30

if (dest=="TYS"): dest=99

if (dest=="MSN"): dest=55

if (dest=="DSM"): dest=23

if (dest=="STT"): dest=23

if (dest=="ALB"): dest=99

if (dest=="BUR"): dest=41

if (dest=="PVD"): dest=32

if (dest=="PSE"): dest=96

if (dest=="OKC"): dest=61

if (dest=="TUL"): dest=60

if (dest=="SMF"): dest=88

if (dest=="ACK"): dest=11

if (dest=="AVL"): dest=10

if (dest=="ABQ"): dest=30

if (dest=="MVY"): dest=68

if (dest=="EGE"): dest=32

if (dest=="CRW"): dest=4

if (dest=="ILM"): dest=79

if (dest=="CAE"): dest=69

t=[[int(year),int(month),int(day),int(carrier),int(origin),int(dest)]]

payload\_scoring = {"input\_data": [{"fields": [["f0","f1","f2","f3","f4","f5"]], "values":t }]} #payload\_scoring = {"input\_data": [{"fields": [array\_of\_input\_fields], "values":

[array\_of\_values\_to\_be\_scored, another\_array\_of\_values\_to\_be\_scored]}]}

response\_scoring = requests.post('https://us- south.ml.cloud.ibm.com/ml/v4/deployments/f4014f53-d84e-4c2a-9dd2- e36cd70e6b22/predictions?version=2022-11-04', json=payload\_scoring, headers={'Authorization': 'Bearer ' + mltoken})

print("Scoring response")

payload\_scoring = {"input\_data": [{"fields": [["f0","f1","f2","f3","f4","f5"]], "values":t }]} pred= response\_scoring.json()

output=pred['predictions'][0]['values'][0][0] print(output)

return render\_template('home.html', prediction\_text = output)

if name == ' main ': app.run(debug=True)

# For mac, make 'app.run(debug=True)'