SPRINT – 2 PROJECT DOCUMENT

Date	18 November 2022
Team ID	PNT2022TMID32830
Project Name	Flight Delay Prediction Using Machine Learning

DEVELOPMENT PHASE:

SPRINT-2:

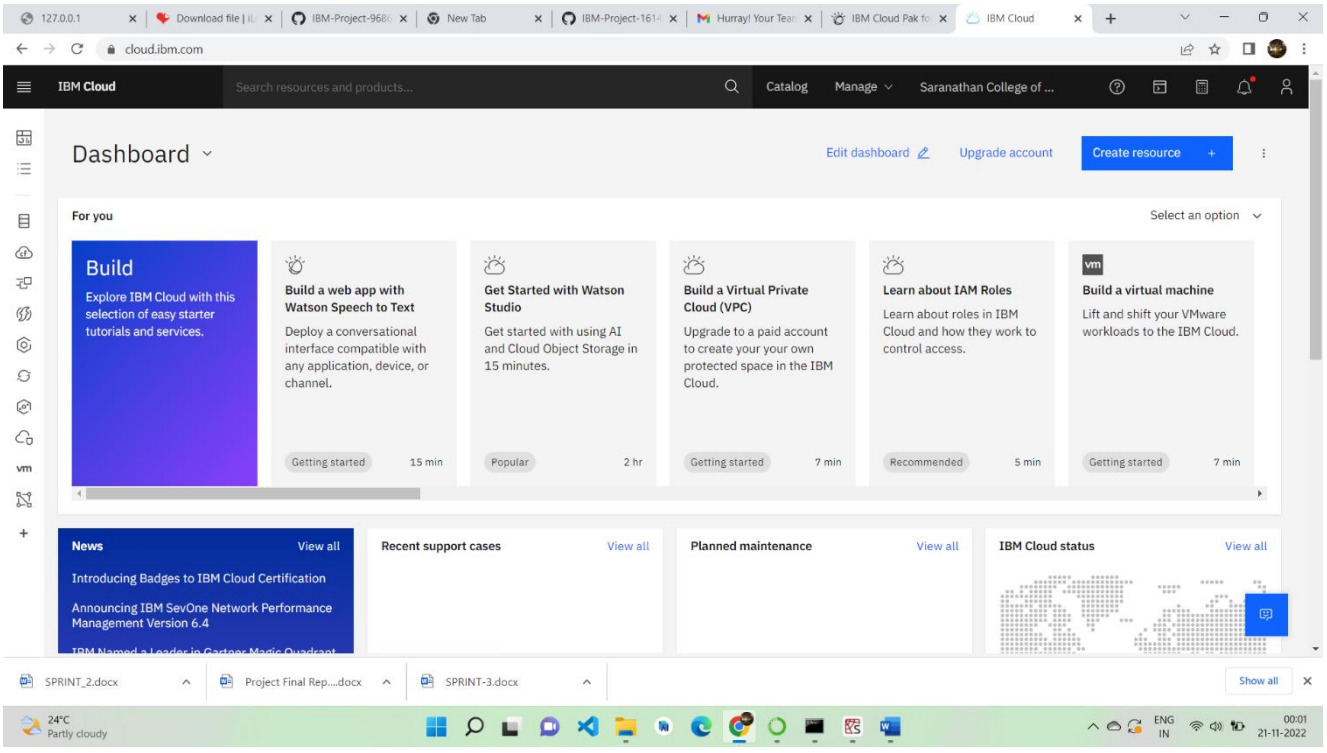
- Creating IBM cloud account & Required Resources
- Deploy our model in IBM Watson
- Creating Dashboard using HTML/CSS
- Create web app and Hosting in falsk
- Testing web app

Creating IBM cloud account & Required Resources:

Creating IBM cloud account:

Frist, need to create IBM Cloud account by using SI Mail Id and SI Password which is provided byIBM in profile.

Below dashboard of an account after created,



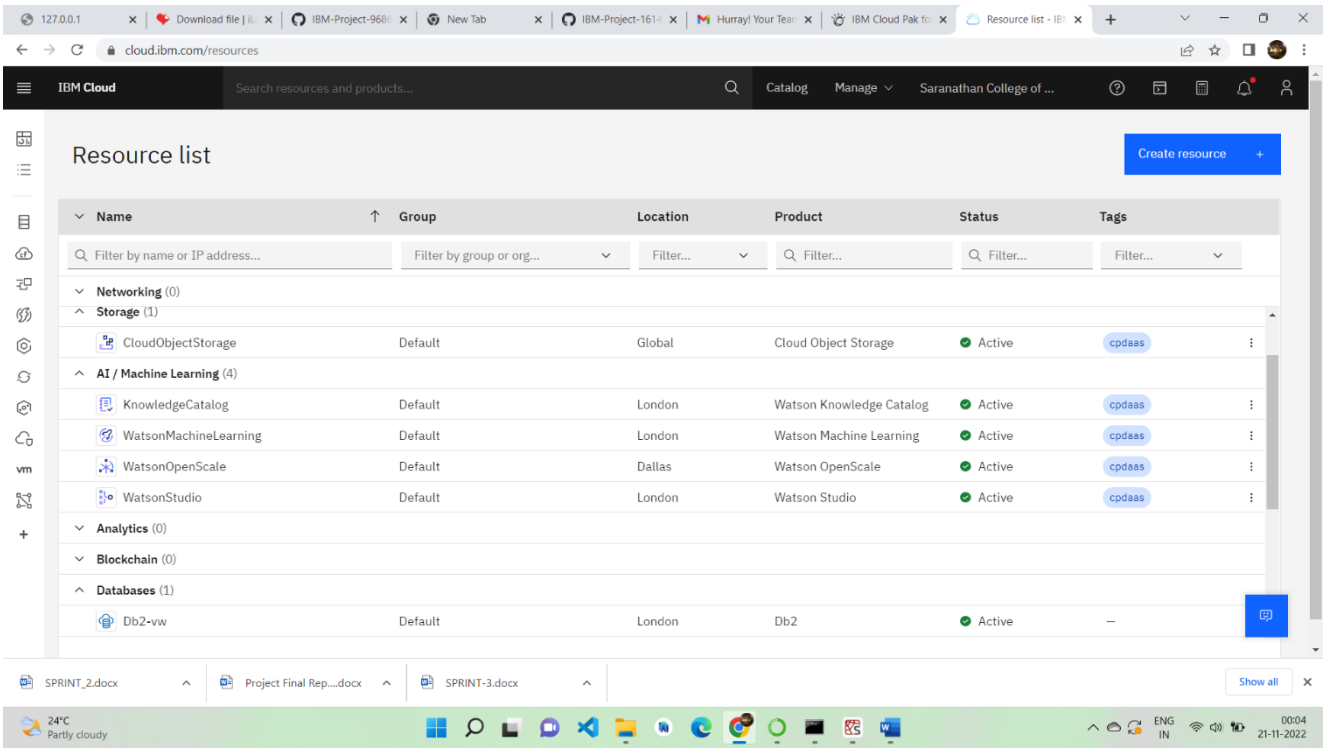
Creating IBM Cloud Required Resources:

After creating IBM cloud account, to deploy ML model, need to create following resources such as,

Cloud Object Storage

Watson Machine LearningWatson Studio

After created above resources Resource List of an account is displayed as follow,



All the resource are in active state.

All the required cloud resources are created successfully.

Deploy our model in IBM Watson:

To deploy ML model in IBM cloud, need to create project in IBM Watson. After successful creation of project import .ipynb file of sprint-1 which ML models are build in Jupyter notebook.

Upload required datasets and import it.Deploy model using following code,

```
!pip install -U ibm-watson-machine-learning
from ibm_watson_machine_learning import APIClient
import json
import numpy as npwml_cred={  "apikey":"IF-dWY9BFLrCPxwHr1kyXWh9AVRL4G4aDElRzF9TsYu6",
 "url":"https://eu-gb.ml.cloud.ibm.com"}
wml_clients=APIClient(wml_cred)wml_clients.spaces.list()
space_id="0e35a1c8-fc26-4a0f-a7ba-2096e25022cf"
```

```
wml_clients.set.default_space(space_id) wml_clients.software_specifications.list(500)
MODEL_NAME="randomforest"
DEPLOYMENT_NAME="rf_deployment"
DEMO_MODEL=rf
soft_sepc_id=wml_clients.software_specifications.get_id_by_name("runtime-22.1-
py3.9")
```

In [115]:

```
model_props={ wml_clients.repository.ModelMetaNames.NAME:MODEL_NAME,
wml_clients.repository.ModelMetaNames.TYPE:"scikit-learn_1.0",
wml_clients.repository.ModelMetaNames.SOFTWARE_SPEC_UID: soft_sepc_id

}
```

In [116]:

```
model_details=wml_clients.repository.store_model(model=DEMO_MODEL,meta_props=model_props,trai
ning_data=x_train,
                                    training_target=y_train.values.ravel())
```

In [117]:

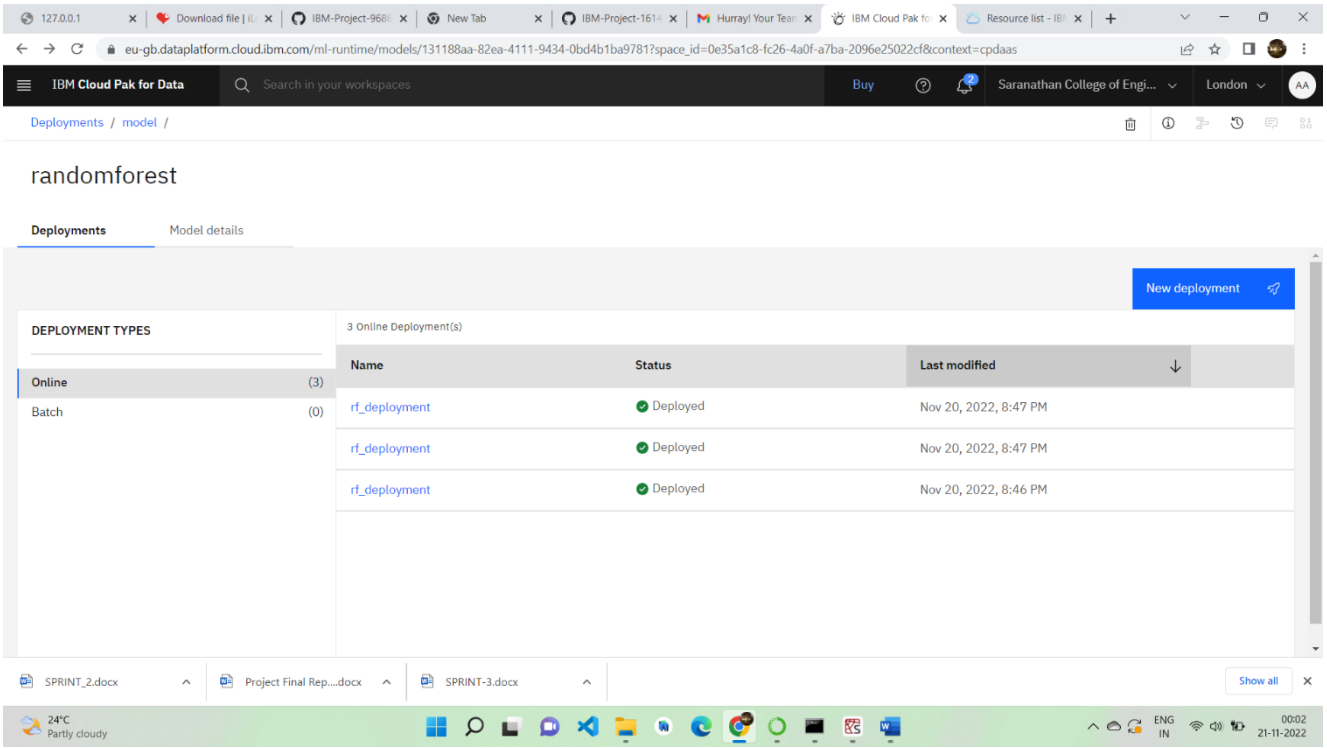
```
model_details

model_id=wml_clients.repository.get_model_id(model_details)dep_props={
wml_clients.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
wml_clients.deployments.ConfigurationMetaNames.ONLINE:{}
}
```

In [125]:

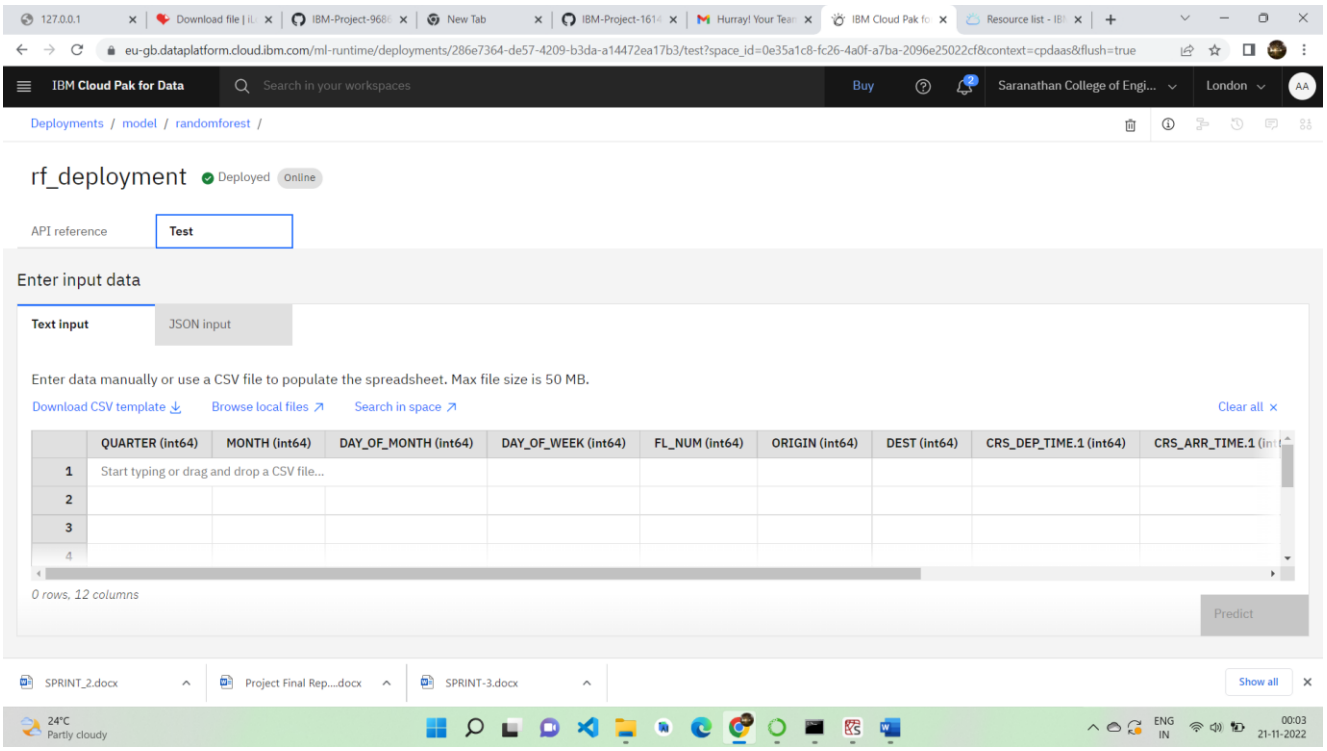
```
deployment=wml_clients.deployments.create(artifact_uid=model_id,meta_props=dep_props
)
```

NOTE: APIKey must need to create to deploy and connect API

After successful of deployment, deployed is appeared in Deployment section as follow,



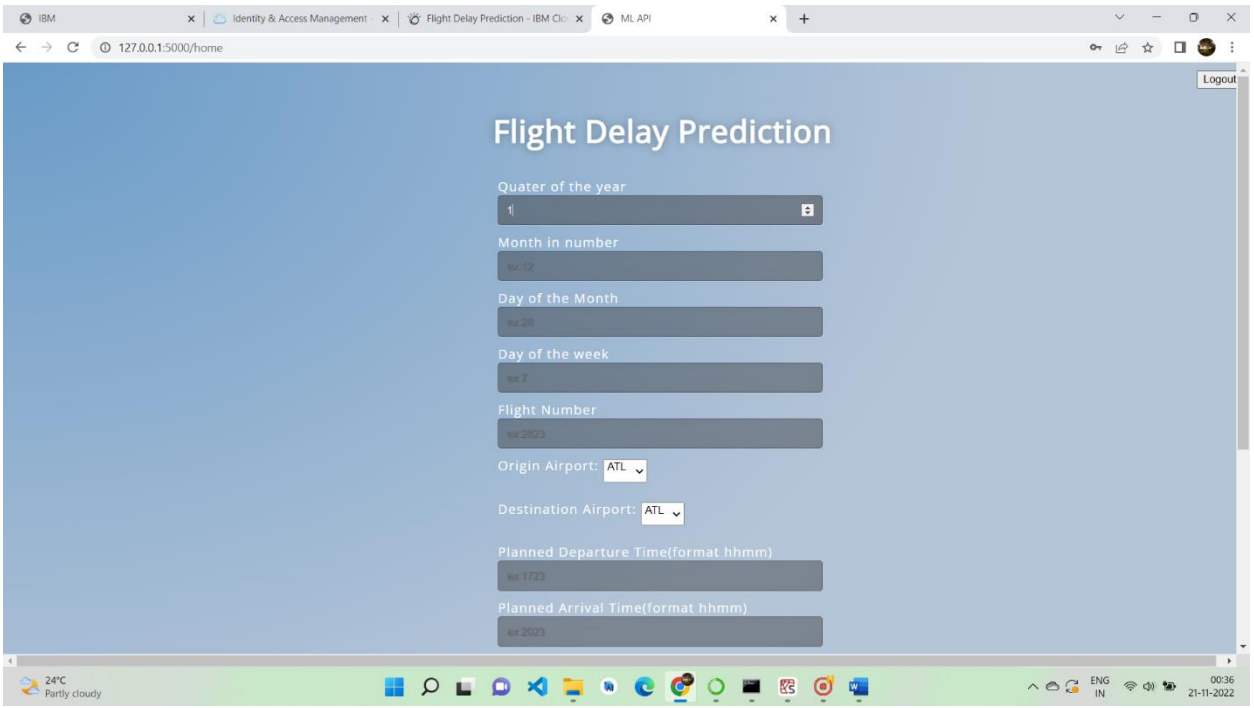
Testing of deployed model as follow, by giving values of all the features and it gives prediction



After these, need to copy API requesting codes on required language(python).

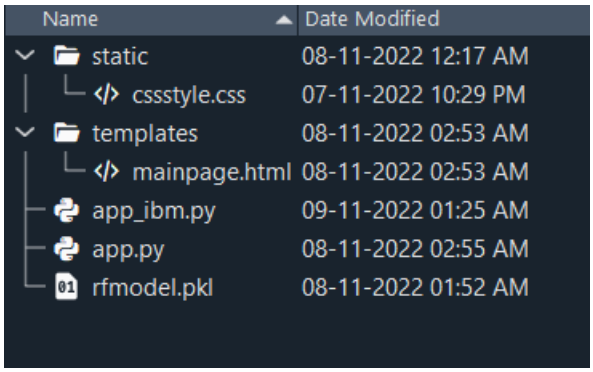
Creating Dashboard using HTML/CSS:

Frontend Dashboard is created using HTML/CSS,Result as web page like,

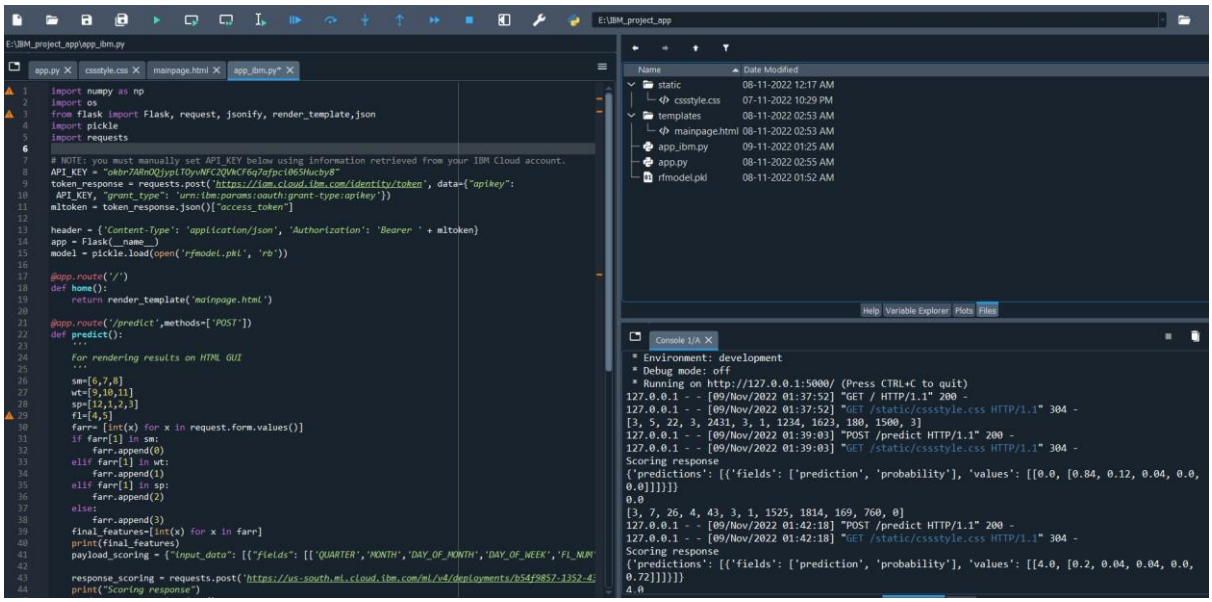


Create web app and Hosting in falsk:

First thing, need to create directory as follow,



Then, code the required logic in app.py file with API connection , request and response code.Spyder IDE looks like,



```
import numpy as np
import os
from flask import Flask, request, jsonify, render_template,json
import pickle
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "okbr7ARnOQjypLTOyvNFC2QVkCF6q7afpc1065Hucby8"
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}
app = Flask(__name__)
model = pickle.load(open('rfmodel.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('mainpage.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    sm=[6,7,8]
    wt=[9,10,11]
    sp=[12,1,2,3]
    fl=[4,5]
    farr= [int(x) for x in request.form.values()]
    if farr[1] in sm:
        farr.append(0)
    elif farr[1] in wt:
        farr.append(1)
    elif farr[1] in sp:
        farr.append(2)
    else:
        farr.append(3)
    final_features=[int(x) for x in farr]
    print(final_features)
    payload_scoring = {"input_data": [{"fields": [['QUARTER','MONTH','DAY_OF_MONTH','DAY_OF_WEEK','FL_NUM
    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/b54f9857-1352-4
    print("Scoring response")
```

```
 * Environment: development
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [09/Nov/2022 01:37:52] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2022 01:37:52] "GET /static/cssstyle.css HTTP/1.1" 304 -
[3, 5, 22, 3, 2431, 3, 1, 1234, 1623, 180, 1500, 3]
127.0.0.1 - - [09/Nov/2022 01:39:03] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2022 01:39:03] "GET /static/cssstyle.css HTTP/1.1" 304 -
Scoring response
{'predictions': [{'fields': ['prediction', 'probability'], 'values': [[0.0, [0.84, 0.12, 0.04, 0.0,
0.0]]]}]}
0.0
[3, 7, 26, 4, 43, 3, 1, 1525, 1814, 169, 760, 0]
127.0.0.1 - - [09/Nov/2022 01:42:18] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2022 01:42:18] "GET /static/cssstyle.css HTTP/1.1" 304 -
Scoring response
{'predictions': [{'fields': ['prediction', 'probability'], 'values': [[4.0, [0.2, 0.04, 0.04, 0.0,
0.72]]]}]}
4.0
```

Run the app.py file.

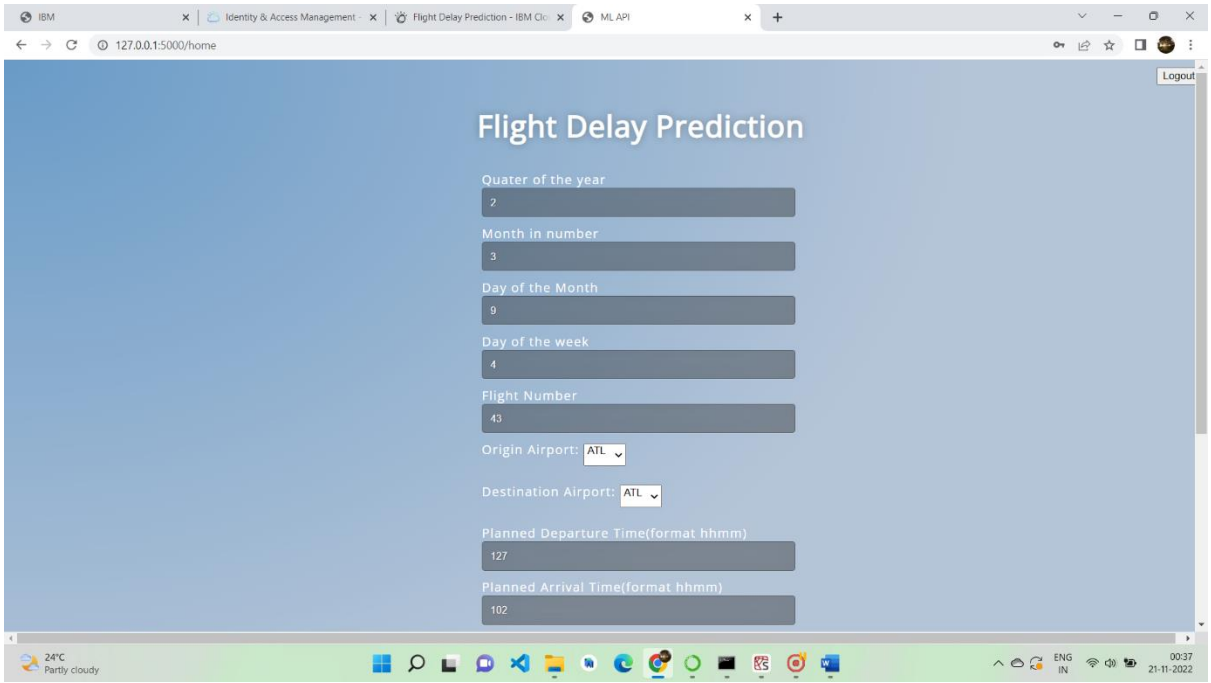
Localhost url is displayed in console, copy and paste in browser then search it , frond end HTML?CSSpage is displayed. Successfully created and hosted web app in flask.

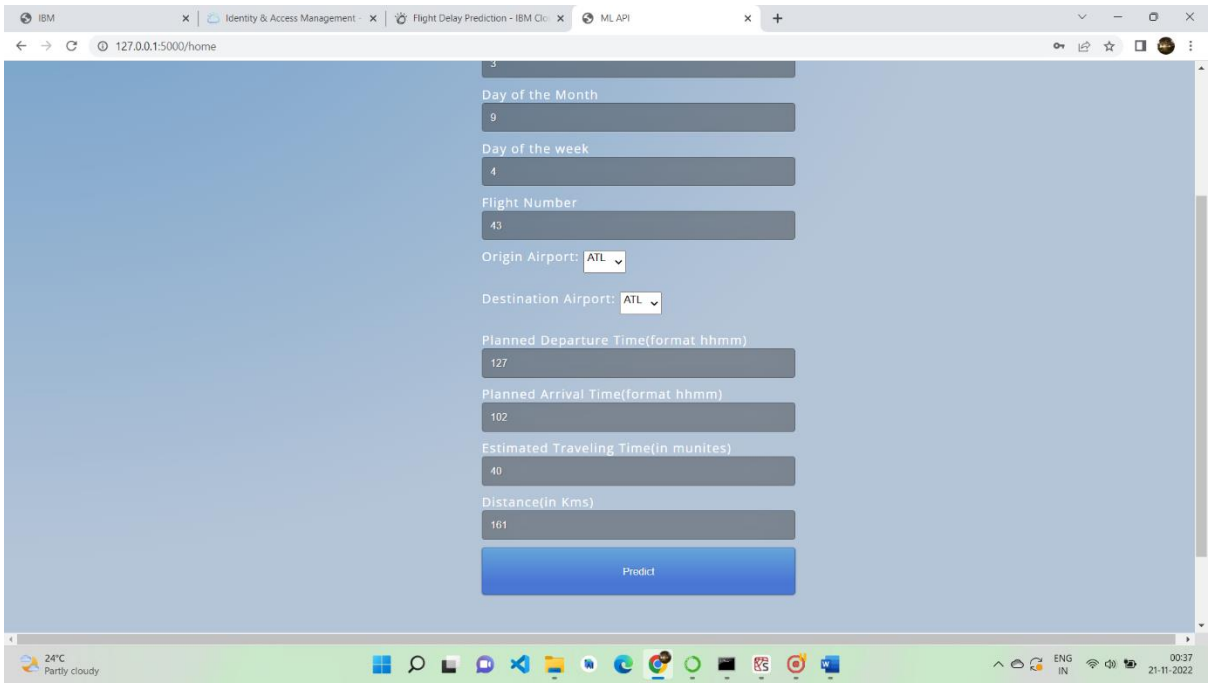
If any error caused as flask in production mode, thenSet FLASK_ENV=Development,

Then run the app

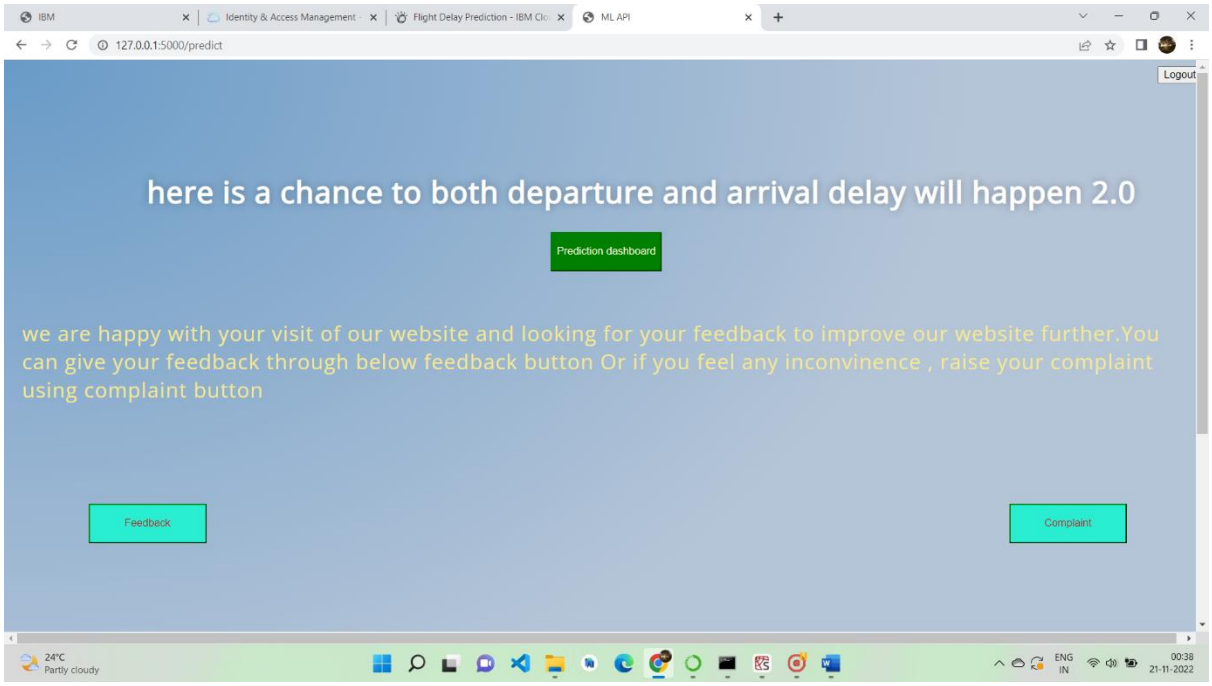
Testing web app:

Enter the data on the required fields,





Output is predicted by ML model successfully.


IBM
Identity & Access Management
Flight Delay Prediction - IBM Clo
ML API
127.0.0.1:5000/predict
Logout
here is a chance to both departure and arrival delay will happen 2.0
Prediction dashboard
we are happy with your visit of our website and looking for your feedback to improve our website further.You can give your feedback through below feedback button Or if you feel any inconvinence , raise your complaint using complaint button
Feedback
Complaint
24°C
Partly cloudy
ENG
IN
00:38
21-11-2022