PROJECT NAME : News Tracker Application

Team ID: PNT2022TMID05148

SPRINT-3

# Using flask to make an api

# import necessary libraries and functions from flask import Flask, jsonify, request

# creating a Flask app app

= Flask( name )

# on the terminal type: curl http://127.0.0.1:5000/ # returns hello world when we use GET.

# returns the data that we send when we use POST. @app.route('/', methods = ['GET', 'POST']) def home(): if(request.method == 'GET'):

data = "hello world" return jsonify({'data': data})

# A simple function to calculate the square of a number

# the number to be squared is sent in the URL when we use GET # on the terminal type: curl http://127.0.0.1:5000 / home / 10

# this returns 100 (square of 10) @app.route('/home/<int:num>', methods = ['GET']) def disp(num): return jsonify({'data': num\*\*2})

# driver function if

name == ' main ':

app.run(debug = Tru# import necessary libraries and functions from flask import Flask, jsonify, request

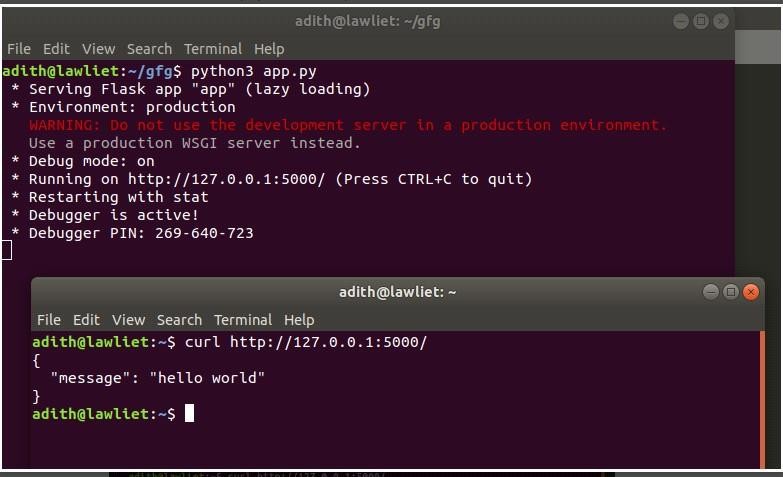
# creating a Flask app app

= Flask( name )

# on the terminal type: curl http://127.0.0.1:5000/ # returns hello world when we use GET.

# returns the data that we send when we use POST.

# OUTPUT :



**Method :**

# using flask\_restful from flask import Flask, jsonify, request from flask\_restful import Resource, Api

# creating the flask app app = Flask( name ) # creating an API object api = Api(app)

# making a class for a particular resource

# the get, post methods correspond to get and post requests # they are automatically mapped by flask\_restful.

# other methods include put, delete, etc.

class Hello(Resource):

# corresponds to the GET request.

# this function is called whenever there # is a GET request for this resource def get(self):

return jsonify({'message': 'hello world'})

# Corresponds to POST request def post(self):

data = request.get\_json() # status code return jsonify({'data': data}), 201

# another resource to calculate the square of a number class Square(Resource):

def get(self, num):

return jsonify({'square': num\*\*2})

# adding the defined resources along with their corresponding urls api.add\_resource(Hello, '/')

api.add\_resource(Square, '/square/<int:num>')

# driver function if

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

# Output:

