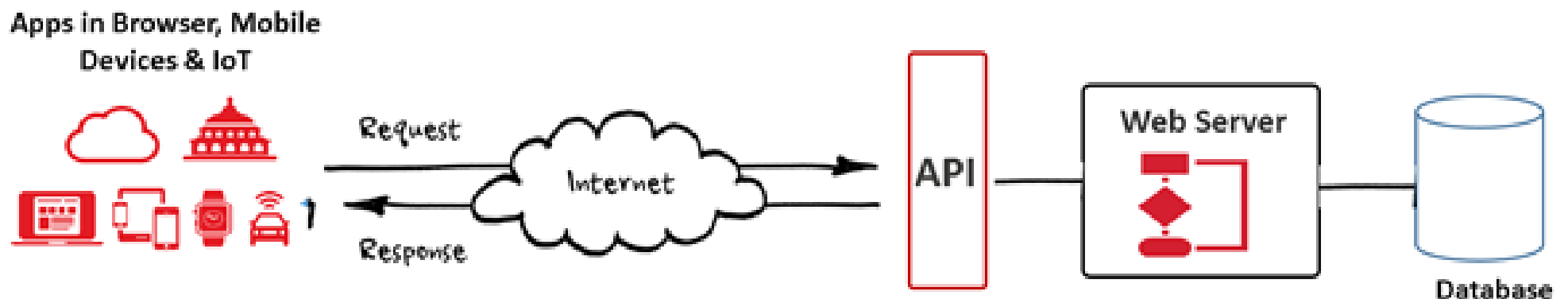


WHAT IS API?

API is the acronym for Application Programming Interface, which is a **software intermediary that allows two applications to talk to each other**. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.



WHAT IS FLASK?

Flask is a **lightweight** WSGI **web application framework**. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.

According to me, Flask is the easiest way to get started with APIs





WHAT A FLASK APPLICATION LOOKS LIKE

Let's start by reviewing the structure of a small Flask application

```
#importing flask
from flask import Flask

#creating flask object
app = Flask(__name__)

#creating route
@app.route('/', methods=['GET', 'POST'])

def main_route():
    pass

#run flask app
if __name__ == '__main__':
    app.run()
```

import flask



create flask object



**create app routes
following with
function associated
with that rout**



run app()



arjun-panwar

WHAT IS ROUTE?

You must be seeing following links while using linkedin

- **`https://www.linkedin.com/feed`**
- **`https://www.linkedin.com/messaging`**

Here /feed and /messaging are routes, you can also recall it as a path to a certain code in our program.

`app.route('/feed')` will run the function following it.

GET VS POST

These are methods of requesting data

**when you search for anything on Google, you can easily see your search query in the resulting URL
but**

When you log in to your Gmail account, the resulting URL doesn't show your Gmail id or password

**Google search use Get Method-For less secure data
Gmail log in use Post Method-for secured fata**

LETS GET STARTED WITH OUR FIRST FLASK API

```
from flask import Flask, render_template, request, jsonify

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST']) # To render Homepage
def home_page():
    return render_template('index.html')

@app.route('/math', methods=['POST']) # This will be called from UI
def math_operation():
    if request.method == 'POST':
        operation = request.form['operation']
        num1 = int(request.form['num1'])
        num2 = int(request.form['num2'])
        if operation == 'add':
            r = num1 + num2
            result = 'the sum of ' + str(num1) + ' and ' + str(num2) + ' is ' + str(r)
        if operation == 'subtract':
            r = num1 - num2
            result = 'the difference of ' + str(num1) + ' and ' + str(num2) + ' is ' + str(r)
        if operation == 'multiply':
            r = num1 * num2
            result = 'the product of ' + str(num1) + ' and ' + str(num2) + ' is ' + str(r)
        if operation == 'divide':
            r = num1 / num2
            result = 'the quotient when ' + str(num1) + ' is divided by ' + str(num2) + ' is ' + str(r)
    return render_template('results.html', result=result)
```

render_template() is used for rendering HTML page

```

@app.route('/via_postman', methods=['POST']) # for calling the API from Postman/SOAPUI
def math_operation_via_postman():
    if (request.method=='POST'):
        operation=request.json['operation']
        num1=int(request.json['num1'])
        num2 = int(request.json['num2'])
        if(operation=='add'):
            r=num1+num2
            result= 'the sum of '+str(num1)+' and '+str(num2) +' is '+str(r)
        if (operation == 'subtract'):
            r = num1 - num2
            result = 'the difference of ' + str(num1) + ' and ' + str(num2) + ' is ' + str(r)
        if (operation == 'multiply'):
            r = num1 * num2
            result = 'the product of ' + str(num1) + ' and ' + str(num2) + ' is ' + str(r)
        if (operation == 'divide'):
            r = num1 / num2
            result = 'the quotient when ' + str(num1) + ' is divided by ' + str(num2) + ' is ' + str(r)
        return jsonify(result)

if __name__ == '__main__':
    app.run()

```

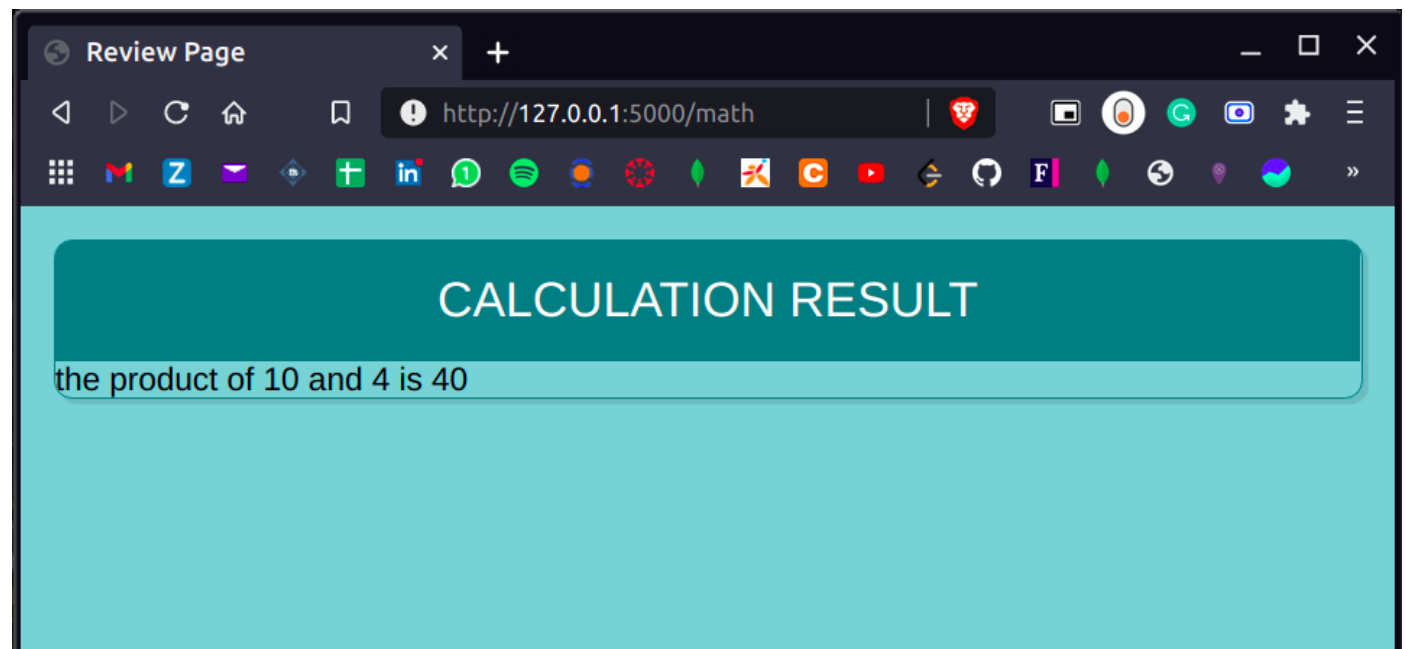
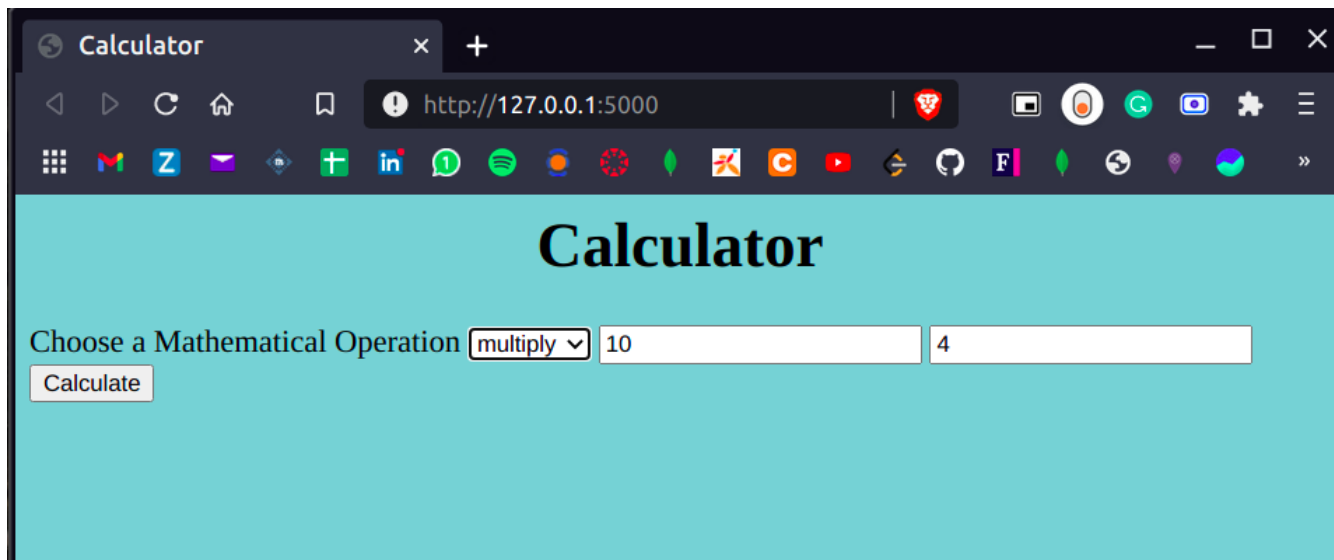
**This is a simple calculator API. Here I have made 3 routes,
The first 2 for using this API in browser using HTML
Third will take json input, which we will pass using Postman**

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster.

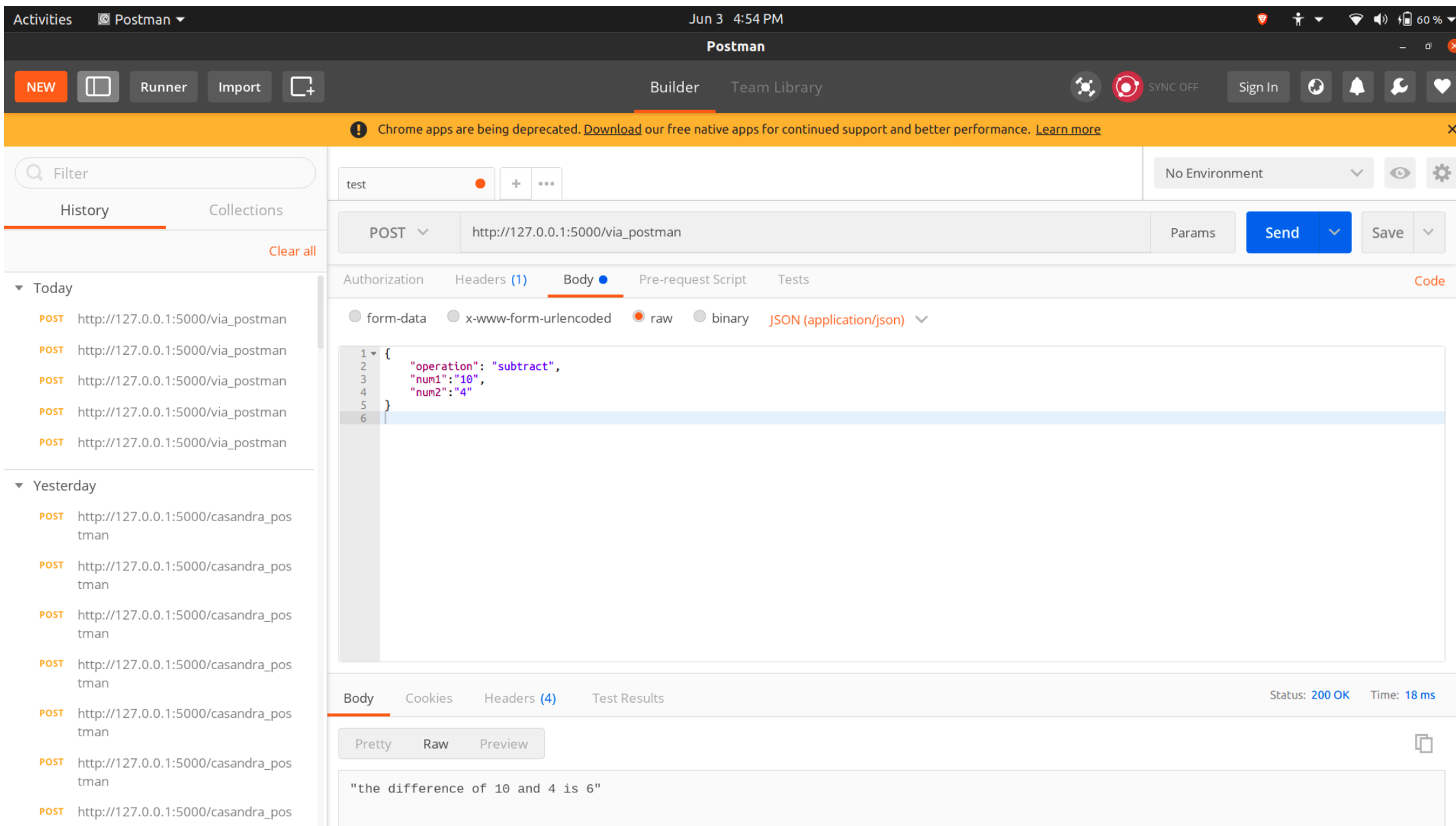


arjun-panwar

BROWSER-USING HTML



USING POSTMAN



The screenshot displays the Postman application interface. The top bar shows the time as 4:54 PM on Jun 3. The main toolbar includes buttons for 'NEW', 'Runner', 'Import', 'Builder', and 'Team Library'. A notification banner at the top states: "Chrome apps are being deprecated. Download our free native apps for continued support and better performance. Learn more".

The left sidebar shows a 'History' tab with a search filter and a 'Collections' tab. Under 'Today', there are five POST requests to 'http://127.0.0.1:5000/via_postman'. Under 'Yesterday', there are five POST requests to 'http://127.0.0.1:5000/cassandra_pos tman'.

The main workspace shows a POST request to 'http://127.0.0.1:5000/via_postman'. The 'Body' tab is selected, showing a JSON payload:

```
1 {
2   "operation": "subtract",
3   "num1": "10",
4   "num2": "4"
5 }
6
```

The bottom section shows the response, which is a string: "the difference of 10 and 4 is 6". The status is 200 OK and the time is 18 ms.