Q1. Explain GET and POST methods.

answer: GET and POST method are used to send data to the server. the best example of GET is the GOOGLE SEARCH URL .

but GET method is send data theough the url. and POST method send data through the body. the best example of POST is the LOGIN PAGE.

Q2. Why is request used in Flask?

Answer: Request in flask is used to access the data that is sent from client to the server.

Q3. Why is redirect() used in Flask?
Answer : Redirect() is used in flask to redirect the user to another url.

Q4. What are templates in Flask? Why is the render_template() function used?
Answer: A Template is a file that contains HTML code. The render_template() function is used to render HTML code in a template file.

Q5. Create a simple API. Use Postman to test it. Attach the screenshot of the output in the Jupyter Notebook.

```
@app.route('/Postman_data',methods=['POST'])
def math_ops():
   if(request.method =='POST'):
     ops = request.json['operation']
     num1 = request.json['num1']
     num2 = request.json['num2']
     if ops == 'add':
        res = num1 + num2
        result = "The sum of    "  + str(num1) + '  and    ' + str(num2) + "  is    "  + str(r)
     if ops == 'sub':
        res = num1 - num2
        result = "The sub of    "  + str(num1) + '  and    ' + str(num2) + "  is    "  + str(r)
     if ops == 'mul':
        res = num1 * num2
        result = "The mul of    "  + str(num1) + '  and    ' + str(num2) + "  is    "  + str(r)
     if ops == 'div':
        res = num1 / num2
        result = "The div of    "  + str(num1) + '  and    ' + str(num2) + "  is    "  + str(r)
     return jsonify(result)

if __name__=="__main__":
   app.run(host="0.0.0.0")
```

```python
@app.route('/Postman_data',methods=['POST'])
def math_ops():
    if(request.method == 'POST'):
        ops = request.json['operation']
        num1 = int(request.json['num1'])
        num2 = int(request.json['num2'])
        if ops == 'add':
            r = num1+num2
            result = "The sum of     " + str(num1) + ' and    ' + str(num2) + " is    " + str(r)
        if ops == 'subtract':
            r = num1-num2
            result = "The subtract of    " + str(num1) + ' and    ' + str(num2) + " is    " + str(r)
        if ops == 'multiply':
            r = num1*num2
            result = "The multiply of     " + str(num1) + ' and   ' + str(num2) + " is    " + str(r)
        if ops == 'divide':
            r = num1/num2
            result = "The divide of    " + str(num1) + ' and   ' + str(num2) + " is    " + str(r)
        if ops == 'Greater then':
            r = num1>num2
            result = "num1 is Greater then num2    " + str(num1) + ' and  ' + str(num2) + " which   is    " + str(r)
        if ops == 'Less then':
            r = num1<num2
            result = " num1 is Less then num2     " + str(num1) + ' and   ' + str(num2) + " which   is     " + str(r)

        return jsonify(result)

if __name__=="__main__":
    app.run(host="0.0.0.0")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
[Running] python -u "/config/workspace/app.py"
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.18.0.14:5000
Press CTRL+C to quit
```

---

https://red-intern-inhhk.pwskills.app:5000//Postman_data

Save

POST | https://red-intern-inhhk.pwskills.app:5000//Postman_data | Send

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings   Cookies

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON ⌄   Beautify

```
1  {
2      "operation":"multiply",
3      "num1": 22,
4      "num2":2
5  }
```

Body Cookies Headers (6) Test Results   Status: 200 OK Time: 252 ms Size: 234 B   Save as Example

Pretty Raw Preview Visualize JSON ⌄

```
1   "The multiply of     22   and    2   is    44"
```