LOG ANALYSER

Description -:
Log Analyser analyzes logs and gets useful data. Here we analyze task4.log and get data like IP, IP country, timestamp, and request for the different protocol(HTTP, FTP, Modbus)

Language used-:
Python,HTML,JavaScript,css

Library Used-:
Geolite2(for IP country)
Datetime(for time)

How to download-:
Python- https://www.python.org/downloads/windows/ (for window) Sudo apt-get install python3(Linux)

Pypnut- pip install maxminddb-geolite2
Datetime- installed by default

Algorithm:-

Python-
calculate_Timestamp (time)-It takes time as a string in the format of (2019-08-27 11:58:07,112) and returns its timestamp
ipInfo(ip= "")-it takes IP as a string and returns its country if possible using geolite2 and Null if it does not get country
program-It read task4.log file line by line separated by '\n' using readline() function and get data like IP, IP Country, Request, Timestamp from it
and store it in respective CSV file for the protocol (HTTP, FTP, and Modbus)

CSS-
Beautify table display

JavaScript-
Up()-It read a CSV file display table (each row is separated by '\n' each cell is separated by ',' and data store format of "data"

HTML-
It get the CSV file and displays it in tabular format

Code:-

Python-
```
from datetime import datetime
from geolite2 import geolite2

geo = geolite2.reader()

def calculate_Timestamp(time):
    time2 = time1
    element = datetime.strptime(time2, "%Y-%m-%d %H:%M:%S,%f")
    Timestamp = datetime.timestamp(element)
    return Timestamp

def ipInfo(ip=""):
    try:
        x = geo.get(ip)
    except ValueError:
        # Faulty IP value
        return "NULL"
    try:
        return x["country"]["names"]["en"]
    except:
        # fault in getting country name
        return "NULL"

file1 = open("task4.log", "r")
Lines = file1.readlines()
```

```
#f2mhf,f2mh,f2mf,f2m use to store modbus's rest,IP,etc in a CSV file.

#f2mhf,f2mh,f2hf,f2h use to store HTTP's rest,IP,etc in a CSV file.

#f2mhf,f2hf,f2mf,f2f use to store FTP's rest,IP,etc in a CSV file.
try:
    f = open("modbus http ftp.csv")
    f2 = open("modbus http ftp.csv", "a")

    fm = open("modbus.csv")
    f2m = open("modbus.csv", "a")

    fh = open("http.csv")
    f2h = open("http.csv", "a")

    ff = open("ftp.csv")
    f2f = open("ftp.csv", "a")

    fmf = open("modbus ftp.csv")
    f2mf = open("modbus ftp.csv", "a")

    fmh = open("modbus http.csv")
    f2mh = open("modbus http.csv", "a")

    fhf = open("http ftp.csv")
    f2hf = open("http ftp.csv", "a")

except IOError:
    f = open("modbus http ftp.csv", "w")
    f.write('"Timestamp","IP","IP Country","Request"\n')
    f.close()
    f2 = open("modbus http ftp.csv", "a")

    fh = open("http.csv", "w")
    fh.write('"Timestamp","IP","IP Country","Request"\n')
    fh.close()
    f2h = open("http.csv", "a")

    fm = open("modbus.csv", "w")
    fm.write('"Timestamp","IP","IP Country","Request"\n')
    fm.close()
    f2m = open("modbus.csv", "a")

    ff = open("ftp.csv", "w")
    ff.write('"Timestamp","IP","IP Country","Request"\n')
    ff.close()
    f2f = open("ftp.csv", "a")

    fmh = open("modbus http.csv", "w")
    fmh.write('"Timestamp","IP","IP Country","Request"\n')
    fmh.close()
    f2mh = open("modbus http.csv", "a")

    fmf = open("modbus ftp.csv", "w")
    fmf.write('"Timestamp","IP","IP Country","Request"\n')
    fmf.close()
    f2mf = open("modbus ftp.csv", "a")

    fhf = open("http ftp.csv", "w")
    fhf.write('"Timestamp","IP","IP Country","Request"\n')
    fhf.close()
    f2hf = open("http ftp.csv", "a")
#countt sore line number of file is reading can use to analyze specific portion of file
countt = 0
for line in Lines:
    time1 = line[0:23]
    Request = ""
    IP = ""
    IPCounty = ""
#space seperate every line with space and store it to find IP
    space = line.split(" ")

    d = line[24:43]
    if d == "Modbus traffic from":
        IP += space[5]
        IP = IP[:-1]

        IPCounty += ipInfo(IP)

        updated_line = " ".join(line.split(" ")[:-1])
        Request += updated_line
        Request = Request[24:]

        f2.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
        f2m.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
        f2mh.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
        f2mf.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )

    dd = line[24:49]
    if dd == "HTTP/1.1 GET request from":

        updated_line = " ".join(line.split(" ")[:-1])
        Request += updated_line
        Request = Request[24:]

        IP += space[6]
        IP = IP[:-2]
        IP = IP[2:]

        IPCounty += ipInfo(IP)

        f2.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
        f2h.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
        f2mh.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
        f2hf.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )

    ddd = line[24:38]
    if ddd == "FTP traffic to":

        updated_line = " ".join(line.split(" ")[:-1])
        Request += updated_line
        Request = Request[24:]

        IP += space[5]
        IP = IP[:-2]
        IP = IP[2:]

        IPCounty += ipInfo(IP)

        f2.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
        f2f.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
        f2mf.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
        f2hf.write(
            '"{}","{}","{}","{}"\n'.format(calculate_Timestamp(time1), IP, IPCounty, Request)
        )
    countt += 1
```

CSS-

```
<style>
 table{
  border-collapse: collapse;

}

 td,  th {
  padding: 8px;
}

 tr:nth-child(even){background-color: #f2f2f2;}

 tr:hover {background-color: #ddd;}

 th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4CAF50;
  color: white;
}
</style>
```

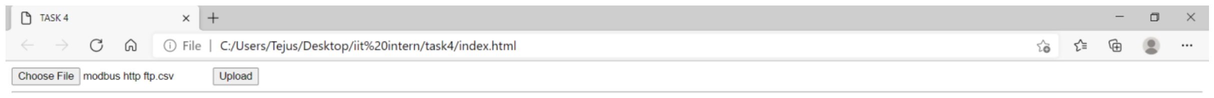
JavaScript-

```
   <script type="text/javascript">
        function Up() {
            var file = document.getElementById("fileUp");
            var r = /^([a-zA-Z0-9\s_\\.\-:])+(.csv)$/;
            if (r.test(file.value.toLowerCase())) {
                if (typeof (FileReader) != "undefined") {
                    var read = new FileReader();
                    read.onload = function (e) {
                        var t = document.createElement("table");
                        var rs = e.target.result.split("\n");

                        for (var i = 0; i < rs.length; i++) {
                                rs[i]=rs[i].substring(1, rs[i].length - 2);
                            var cs = rs[i].split("\",\"");
                            var dd = new Date(0);
                            dd.setUTCSeconds(cs[0]);
                            if(i>0)
                            {cs[0]=dd;}
                            else
                            {cs[0]="Time in IST"}
                            if (cs.length > 1) {
                                var rw = t.insertRow(-1);
                                for (var j = 0; j < cs.length; j++) {
                                    var c = rw.insertCell(-1);
                                    c.innerHTML = cs[j];
                                }
                            }
                        }
                        var CSV = document.getElementById("CSV");
                        CSV.innerHTML = "";
                        CSV.appendChild(t);
                    }
                    read.readAsText(file.files[0]);
                } else {
                    alert("This browser does not support HTML5.");
                }
            } else {
                alert("Please upload a valid CSV file.");
            }
        }
    </script>
```

HTML-

```
<html>
<head>
   <title>TASK 4</title>
<style>
 table{
  border-collapse: collapse;
}

 td,  th {
  padding: 8px;
}

 tr:nth-child(even){background-color: #f2f2f2;}

 tr:hover {background-color: #ddd;}

 th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4CAF50;
  color: white;
}
</style>
</head>
<body>
   <script type="text/javascript">
       function Up() {
           var file = document.getElementById("fileUp");
           var r = /^([a-zA-Z0-9\s_\\.\-:])+(.csv)$/;
           if (r.test(file.value.toLowerCase())) {
               if (typeof (FileReader) != "undefined") {
                   var read = new FileReader();
                   read.onload = function (e) {
                       var t = document.createElement("table");
                       var rs = e.target.result.split("\n");

                       for (var i = 0; i < rs.length; i++) {
                               rs[i]=rs[i].substring(1, rs[i].length - 2);
                           var cs = rs[i].split("\",\"");
                           var dd = new Date(0);
                           dd.setUTCSeconds(cs[0]);
                           if(i>0)
                           {cs[0]=dd;}
                           else
                           {cs[0]="Time in IST"}
                           if (cs.length > 1) {
                               var rw = t.insertRow(-1);
                               for (var j = 0; j < cs.length; j++) {
                                   var c = rw.insertCell(-1);
                                   c.innerHTML = cs[j];
                               }
                           }
                       }
                       var CSV = document.getElementById("CSV");
                       CSV.innerHTML = "";
                       CSV.appendChild(t);
                   }
                   read.readAsText(file.files[0]);
               } else {
                   alert("This browser does not support HTML5.");
               }
           } else {
               alert("Please upload a valid CSV file.");
           }
       }
   </script>
   <input type="file" id="fileUp"/>
   <input type="button" id="upload" value="Upload" onclick="Up()" />
   <hr />
   <div id="CSV">
   </div>
</body>
</html>
```

Input

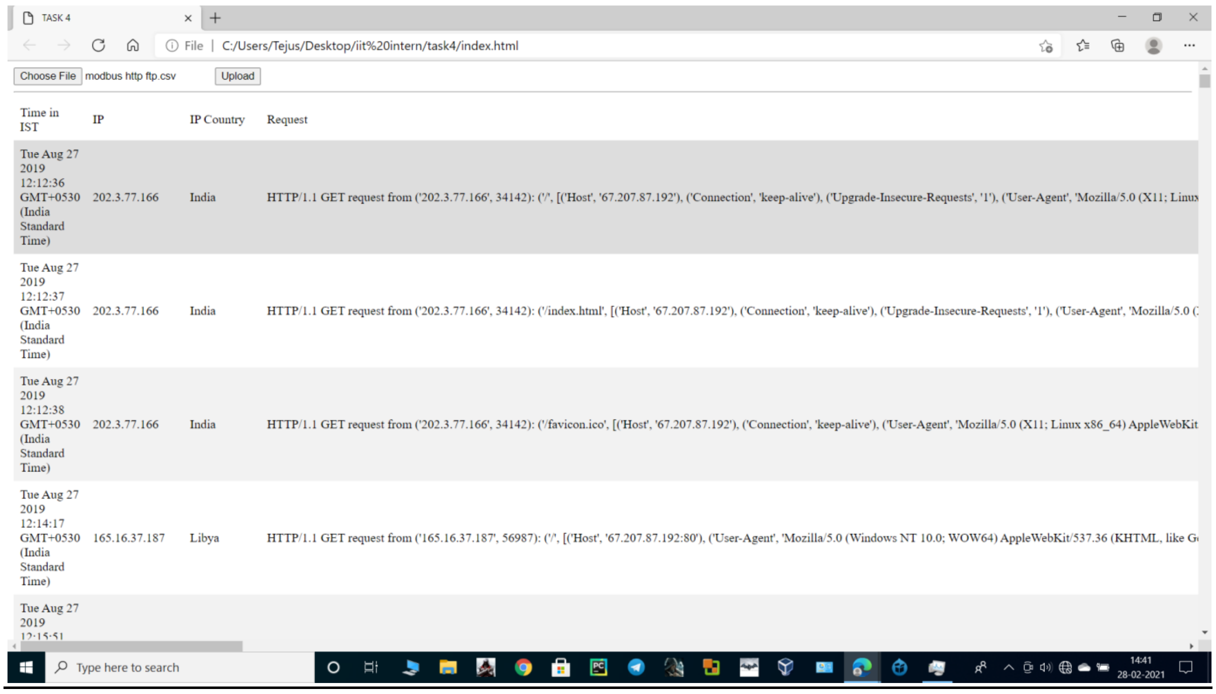


TASK 4

C:/Users/Tejus/Desktop/iit%20intern/task4/index.html

Choose File modbus http ftp.csv Upload

Output



TASK 4

C:/Users/Tejus/Desktop/iit%20intern/task4/index.html

Choose File modbus http ftp.csv Upload

Time in IST	IP	IP Country	Request
Tue Aug 27 2019 12:12:36 GMT+0530 (India Standard Time)	202.3.77.166	India	HTTP/1.1 GET request from ('202.3.77.166', 34142): ('/', [('Host', '67.207.87.192'), ('Connection', 'keep-alive'), ('Upgrade-Insecure-Requests', '1'), ('User-Agent', 'Mozilla/5.0 (X11; Linux
Tue Aug 27 2019 12:12:37 GMT+0530 (India Standard Time)	202.3.77.166	India	HTTP/1.1 GET request from ('202.3.77.166', 34142): ('/index.html', [('Host', '67.207.87.192'), ('Connection', 'keep-alive'), ('Upgrade-Insecure-Requests', '1'), ('User-Agent', 'Mozilla/5.0 (
Tue Aug 27 2019 12:12:38 GMT+0530 (India Standard Time)	202.3.77.166	India	HTTP/1.1 GET request from ('202.3.77.166', 34142): ('/favicon.ico', [('Host', '67.207.87.192'), ('Connection', 'keep-alive'), ('User-Agent', 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit
Tue Aug 27 2019 12:14:17 GMT+0530 (India Standard Time)	165.16.37.187	Libya	HTTP/1.1 GET request from ('165.16.37.187', 56987): ('/', [('Host', '67.207.87.192:80'), ('User-Agent', 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like G
Tue Aug 27 2019 12:15:51			