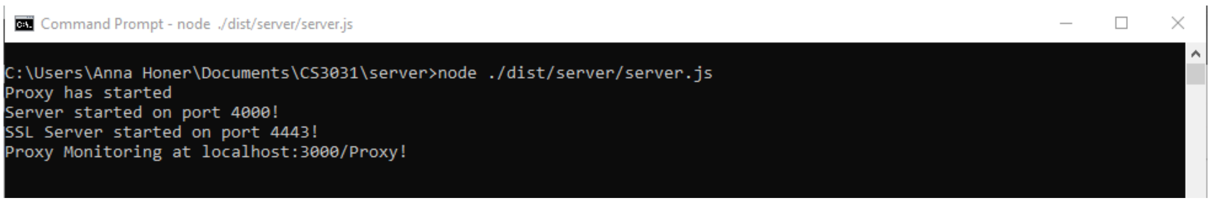
PROJECT ONE

April 1, 2019

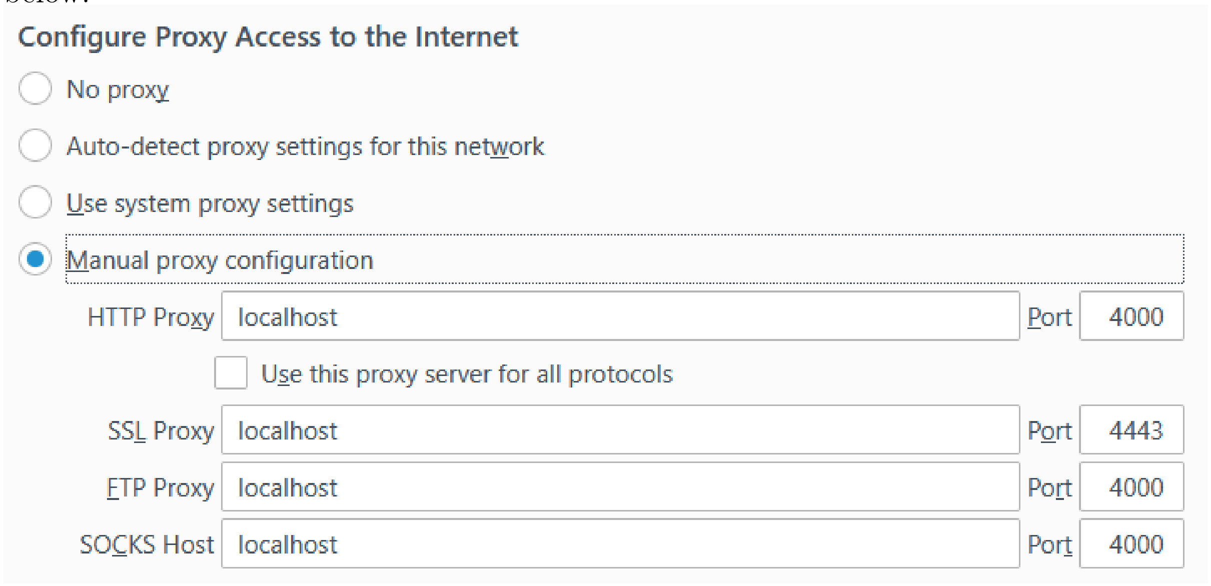
Anna-Zorina Honer
15324090
honera@tcd.ie

0.1 Walkthrough



```
C:\Users\Anna Honer\Documents\CS3031\server>node ./dist/server/server.js
Proxy has started
Server started on port 4000!
SSL Server started on port 4443!
Proxy Monitoring at localhost:3000/Proxy!
```

We being by running our server via console. We start the server on ports 4000, 4443 for the SSL server, and we monitor on port 3000. We have firefox to redirect via ports listed below.

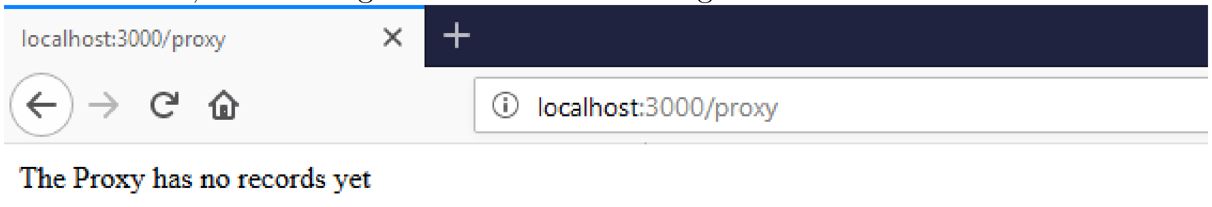


Configure Proxy Access to the Internet

- ( ) No proxy
- ( ) Auto-detect proxy settings for this network
- ( ) Use system proxy settings
- (●) Manual proxy configuration

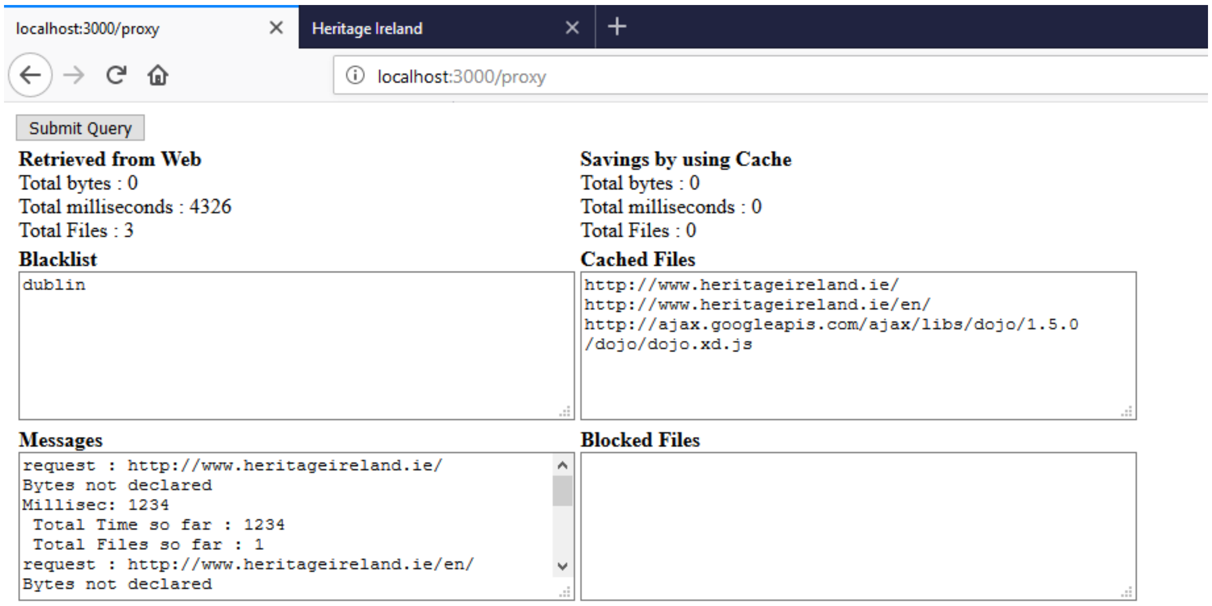
HTTP Proxy	localhost	Port	4000
	☐ Use this proxy server for all protocols		
SSL Proxy	localhost	Port	4443
FTP Proxy	localhost	Port	4000
SOCKS Host	localhost	Port	4000

We now move to localhost:3000/proxy to view our proxy information. Before any sites have been loaded, we will be greeted with the following screen:



The Proxy has no records yet

Once a site has been loaded, it will be moved to the cache. Information regarding the cache will be displayed on our Cached Files section including the total btyes saved, time saved, and the sites cached. However, as our first time loading any site is a cache miss, no savings will have been made.



Retrieved from Web
Total bytes : 0
Total milliseconds : 4326
Total Files : 3

Savings by using Cache
Total bytes : 0
Total milliseconds : 0
Total Files : 0

Blacklist

```
dublin
```

Cached Files

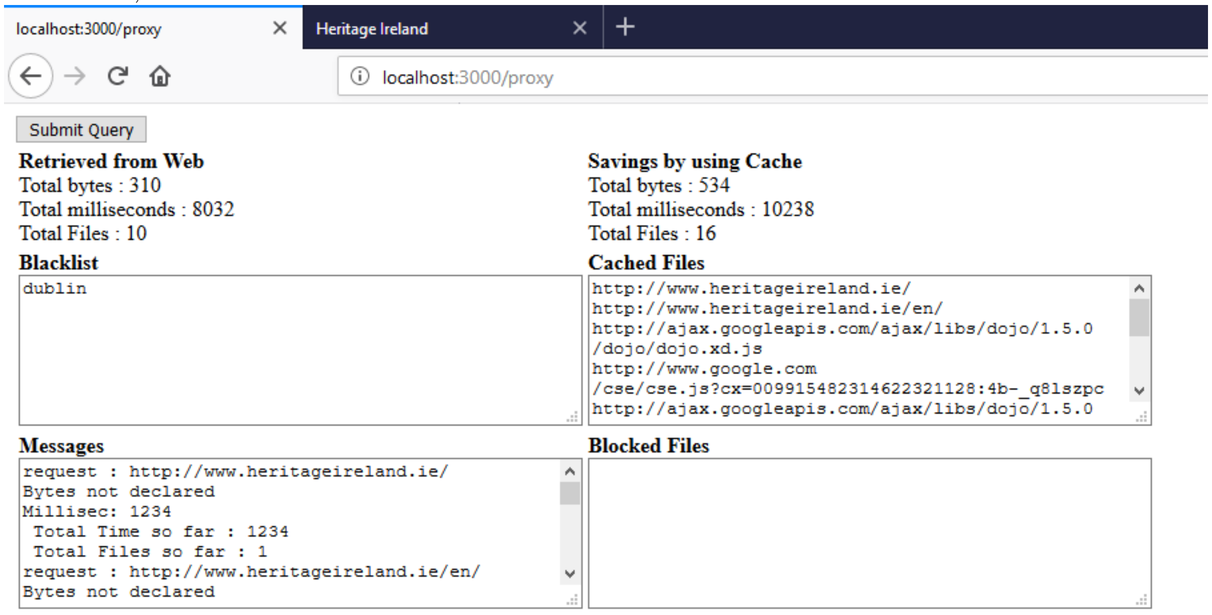
```
http://www.heritageireland.ie/
http://www.heritageireland.ie/en/
http://ajax.googleapis.com/ajax/libs/dojo/1.5.0
/dojo/dojo.xd.js
```

Messages

```
request : http://www.heritageireland.ie/
Bytes not declared
Millisec: 1234
 Total Time so far : 1234
 Total Files so far : 1
request : http://www.heritageireland.ie/en/
Bytes not declared
```

Blocked Files

For reference, I include an image of the cache after refreshing http://www.heritageireland.ie several times, to show the cache in action.



Retrieved from Web
Total bytes : 310
Total milliseconds : 8032
Total Files : 10

Savings by using Cache
Total bytes : 534
Total milliseconds : 10238
Total Files : 16

Blacklist

```
dublin
```

Cached Files

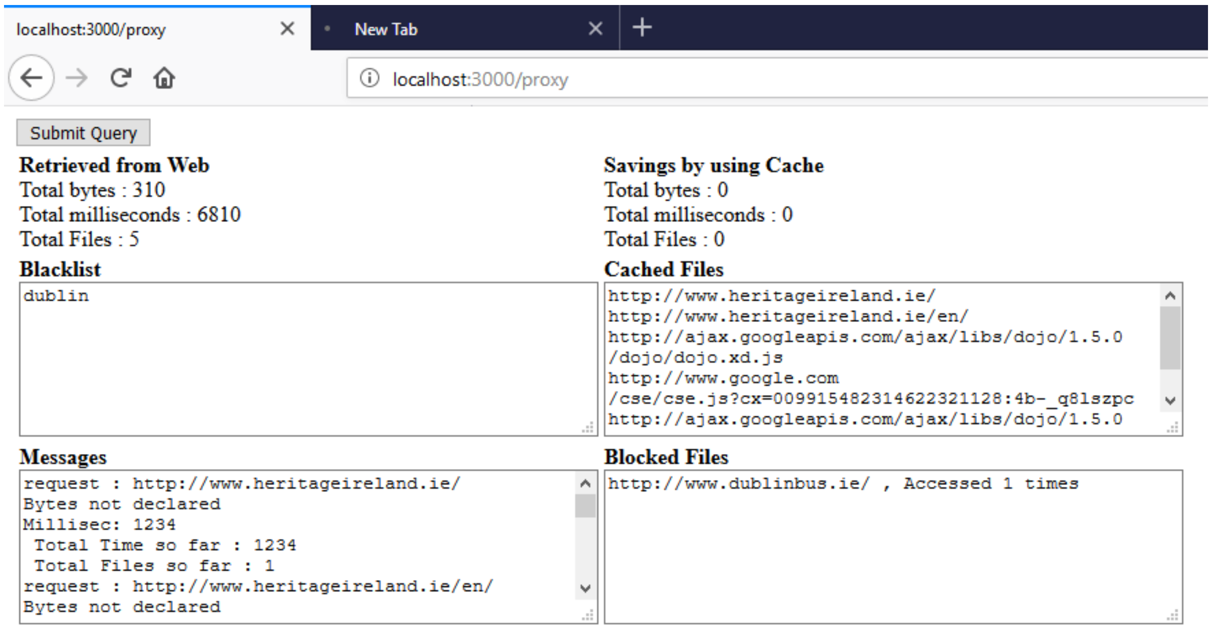
```
http://www.heritageireland.ie/
http://www.heritageireland.ie/en/
http://ajax.googleapis.com/ajax/libs/dojo/1.5.0
/dojo/dojo.xd.js
http://www.google.com
/cse/cse.js?cx=009915482314622321128:4b-_q8lszpc
http://ajax.googleapis.com/ajax/libs/dojo/1.5.0
```

Messages

```
request : http://www.heritageireland.ie/
Bytes not declared
Millisec: 1234
 Total Time so far : 1234
 Total Files so far : 1
request : http://www.heritageireland.ie/en/
Bytes not declared
```

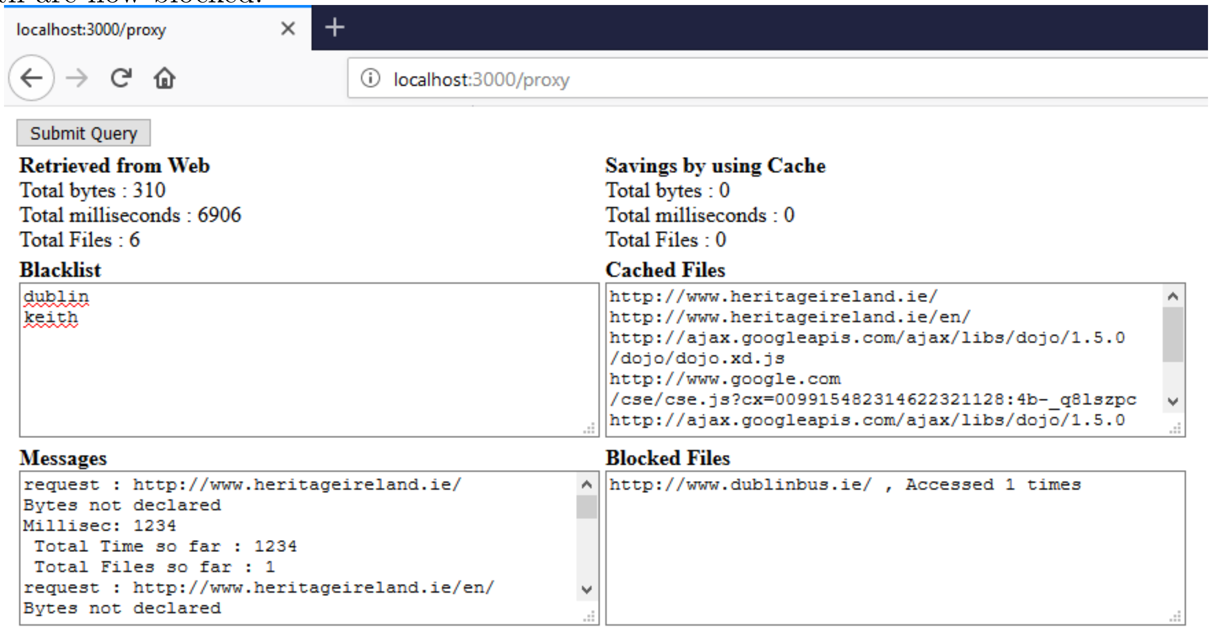
Blocked Files

From here, we can continues to access other sites, however the first one we will try is http://www.dublinbus.ie, which is blocked as "dublin" is a key word included under our blacklist. On the second tab, we can see http://www.dublinbus,ie attempt to load, however it will stall indefinitely as it is blocked.



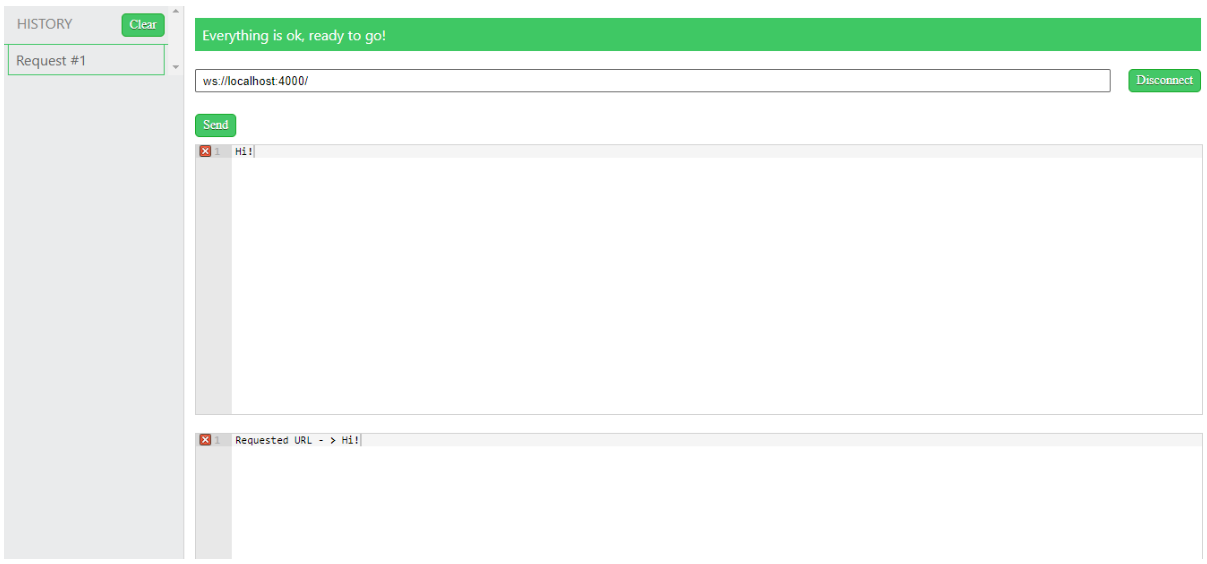
```
Retrieved from Web                    Savings by using Cache
Total bytes : 310                     Total bytes : 0
Total milliseconds : 6810             Total milliseconds : 0
Total Files : 5                       Total Files : 0
Blacklist                             Cached Files
dublin                                http://www.heritageireland.ie/
                                      http://www.heritageireland.ie/en/
                                      http://ajax.googleapis.com/ajax/libs/dojo/1.5.0
                                      /dojo/dojo.xd.js
                                      http://www.google.com
                                      /cse/cse.js?cx=009915482314622321128:4b-_q8lszpc
                                      http://ajax.googleapis.com/ajax/libs/dojo/1.5.0
Messages                              Blocked Files
request : http://www.heritageireland.ie/    http://www.dublinbus.ie/ , Accessed 1 times
Bytes not declared
Millisec: 1234
 Total Time so far : 1234
 Total Files so far : 1
request : http://www.heritageireland.ie/en/
Bytes not declared
```

We can also add and subtract items from our blacklist, via the keyword system. We can be as specific as we wish with the updated list, and urls containing the phrases dublin or keith are now blocked.



```
Retrieved from Web                    Savings by using Cache
Total bytes : 310                     Total bytes : 0
Total milliseconds : 6906             Total milliseconds : 0
Total Files : 6                       Total Files : 0
Blacklist                             Cached Files
dublin                                http://www.heritageireland.ie/
keith                                 http://www.heritageireland.ie/en/
                                      http://ajax.googleapis.com/ajax/libs/dojo/1.5.0
                                      /dojo/dojo.xd.js
                                      http://www.google.com
                                      /cse/cse.js?cx=009915482314622321128:4b-_q8lszpc
                                      http://ajax.googleapis.com/ajax/libs/dojo/1.5.0
Messages                              Blocked Files
request : http://www.heritageireland.ie/    http://www.dublinbus.ie/ , Accessed 1 times
Bytes not declared
Millisec: 1234
 Total Time so far : 1234
 Total Files so far : 1
request : http://www.heritageireland.ie/en/
Bytes not declared
```

Finally, I include the Websocket server, which allows the user to pass messages through the websocket. It allows for multiple requests to come through and threads them appropriately.



Everything is ok, ready to go!

ws://localhost:4000/

```
Hi!
```

```
Requested URL - > Hi!
```

0.2 Code Description

The Code is split into a few distinct sections. From lines 1 to 52 we set up our foundation for the program - imports, constants, interfaces, lets. Imports worth noting include express, which we use to aid the creation of the web application, fs, which is used to navigate our file system - will be explained further under blacklist, and body-parser, which is included to aid taking in and reading the body of a site we intend to visit.

Under constants, we have options - used when generation our keys and certs, app for our web interface, server for creating a http server, sserver for a https server, and finally wss, for our websocket server.

Under Interfaces we create types which will be used in the cache creation.Under Lets we have misc variables.

Our http and https "blocks" carry out in structurally similar ways. In the code, we begin with a site which has already been cached, but for the purposes of this report we will begin with creating the cache web interface, including writing our header and setting up our cache Files, Time and Bytes variables. We see options in use here when parsing the requested url. "var connector" retrieves the response from the url and various processes are paused while the header is creates. We record the servers response and status code in said header. Following this, the processes are resumes and any retrieved data is stored as a chunk. On the serverResponse's end, ie when the page is finished loading, we publish the results of the url request. Our cache variables are set up and reset to 0.

In the case that the site has already been cached, we can retrieve the requested url from said cache, any further requested are paused while this is retrieved and the cache stats are retrieved. Requests are resumed and the url is loaded in from the cache.

An alternative exists to both these scenarios. In the case that a blacklisted keyword matches any part of the url we prevent the url from being loaded. We push news of this to both the console and our web interface.

The https section for the most part is what has been listed above, however we must use the ssl key generated by our options constant. Unfortunately, I could not get the cache to work correctly for https sites. This includes images or videos embedded in a http site.

The Page formatting section of our code is the html design of the web interface. It allows for us to read in and update the blacklist file we have stored, while displaying all the information regarding our cache and blocked sites.

The semi-final section of code deals with the websocket. The websocket allows us to transfer messages but not urls, so much of the worry from the http and https sections are not relevant here. We do however still publish results regarding how long it took to send these messages etc.

At the very end of our code we have the creation of our apps and servers. Our proxy app receives any url requests. Our http and https servers are running on ports 4000 and 4443 respectively. Our websocket is set to run on port 3000.

Below you can find all the code listed.

0.3 Code

```
########## IMPORTS    ###############
import * as express from 'express';
import * as http from 'http';
import * as https from 'https';
import * as request from 'request';
import * as WebSocket from 'ws';
import * as url from 'url';
import * as fs from 'fs';
import * as bodyParser from 'body-parser';



########## CONSTS ###############
const options = {
  key: fs.readFileSync('./src/key.pem'),
  cert: fs.readFileSync('./src/cert.pem')
};
const app= express();
//http as opposed to
const server = http.createServer();
//https
const sserver = https.createServer(options);
const wss = new WebSocket.Server({ server });

########## TYPES and INTERFACES ###############
type PCache ={
    Body: any;
    Header: any;
    Status: any;
    Bytes: any;
    Time:any;
}
```

```
type PBlocked ={
    Freq:any;
}
interface CacheObject {
    [key: string]: PCache;
}
interface BlockedObject {
    [key: string]: PBlocked;
}

 ######### LETS #############
let black_list = JSON.parse(fs.readFileSync('./src/black_list.json',
    'utf8'));
//read fs for blist
let wBytes=0,wTime=0, wFiles=0, tBytes=0;
let cBytes=0,cTime=0, cFiles=0;
let bkey=''
let mng ="The Proxy has no records yet"
let hData: any
let hHeader:any
let hStatus:any
let tstart=Date.now(), tend=Date.now();
let cache: CacheObject = {};
let blocked: BlockedObject = {};
let consoleLog = new Array;


 ######### HTTP PROXY ########
server.on("request",(request, response)=>{
    request.pause();
    let showurl=true;
    black_list.forEach((v:any)=>{
        if (request.url.includes(v)){
                bkey=v;
                showurl=false;
```

```
            return false;
        }
    })
    if (showurl) {
        if (typeof cache[request.url] !="undefined") {
            //set up cache in case of new site
                consoleLog.push("Cached page : " + request.url)
                //write cache header
                response.writeHeader(cache[request.url].Status,
                    cache[request.url].Header);
                response.write(cache[request.url].Body);
                cFiles+=1;
                cTime+=Number(cache[request.url].Time);
                cBytes+=Number(cache[request.url].Bytes);
                consoleLog.push("Saved Bandwidth : " +
                    cache[request.url].Bytes);
                consoleLog.push("Saved Time : " + cache[request.url].Time);
                        request.pipe(response);
                request.resume();
        } else {
                tstart=Date.now()
                var options = url.parse(request.url);
                var connector = http.request(options,
                    function(serverResponse) {
                        serverResponse.pause();
                    response.writeHeader(serverResponse.statusCode,
                        serverResponse.headers);
                    serverResponse.pipe(response);
                        hStatus=serverResponse.statusCode;
                    hHeader=serverResponse.header;s
                    serverResponse.resume();
                    hData='';
                    serverResponse.on('data', chunk => {
                                hData+=chunk;
                                });
                serverResponse.on('end',() => {
                    wFiles+=1;
                    consoleLog.push('request : ' + request.url);
                    if (typeof serverResponse.headers
                        ['content-length'] =="undefined") {
                            wBytes=wBytes+0;
                            tBytes=0;
                            consoleLog.push("Bytes not declared");
                    } else {
                            tBytes=Number(serverResponse.headers
                                ['content-length'])
                                    consoleLog.push
                                        ("Bytes: " + tBytes)
                                    wBytes=wBytes+tBytes
                                    consoleLog.push(" Total" +
                                        "Bytes so far : " + wBytes)
                            }
                            tend=Date.now()
                            consoleLog.push("Millisec: " +
                                (tend-tstart))
                            wTime=wTime+(tend-tstart);
                            cache[request.url]={Body:hData,Header:
                                hHeader,Status:hStatus, Bytes:
                                    tBytes,Time:(tend-tstart)};
                            consoleLog.push(" Total Time so far : "
                                + wTime);
                            consoleLog.push(" Total Files so far : "
                                + wFiles);
                })
            });
            request.pipe(connector);
            request.resume();
    }
    } else {
    if (typeof blocked[request.url]=="undefined") {
            blocked[request.url]={Freq:1}
        } else {
            blocked[request.url]={Freq:blocked[request.url].Freq+1}
        }
                consoleLog.push(request.url +
                    ' was blocked as it is containing the keyword ' + bkey)
                app.get('/', (req, res) => res.send(request.url +
                    ' was blocked as it is containing the keyword '
                        + bkey));
                request.resume();
        }


######### PAGE FORMATTING #########
        mng="<html><body><form action='blacklist' method='post'>
            <input type='submit'>"
    mng+="<table><tr><td><strong>Retrieved from Web
        </strong><div>Total bytes : " + wBytes + "</div>"
        mng+="<div>Total milliseconds : " + wTime + "</div><div>
            Total Files : " + wFiles + "</div></td>"
    mng+="<td><strong>Savings by using Cache</strong><div>Total bytes : "
        + cBytes + "</div>"
        mng+="<div>Total milliseconds : " + cTime + "</div><div>
            Total Files : " + cFiles + "</div>"
        mng+="</td></tr>"
    mng+="<tr><td><strong>Blacklist</strong><br><textarea id='bl'
        rows='6' cols='50'>"
        black_list.forEach((v:any)=>{
            mng+=v+"\n"
        })
    mng+="</textarea></td>"
    mng+="<td><strong>Cached Files</strong><br><textarea id='cf'
        rows='6' cols='50'>"
        for (let key in cache) {
            mng+=key + "\n";
        }
    mng+="</textarea></td></tr>"
    mng+="<tr><td><strong>Messages</strong><br><textarea id='msg'
        rows='6' cols='50'>"
        for (let entry of consoleLog) {
            mng+=entry+ "\n";
        }
    mng+="</textarea></td>"
    mng+="<td><strong>Blocked Files</strong><br><textarea id=
        'blocked' rows='6' cols='50'>"
        for (let key in blocked) {
            mng+=key + " , Accessed " + Number(blocked[key].Freq) +
                " times\n";
                        consoleLog.push("Client attempted to access "
                            + key);
        }
    mng+="</textarea></td></tr></table></form></body><html>"
});

 ######### HTTPS PROXY #########
sserver.on("request",(request, response)=>{
consoleLog.push("    #########    SSL ######### ")
    request.pause();
    let showurl=true;
    black_list.forEach((v:any)=>{
        if (request.url.includes(v)){
            bkey=v;
            showurl=false;
            return false;
        }
    })
    if (showurl) {
        if (typeof cache[request.url] !="undefined") {
            consoleLog.push("Cached page : " + request.url);
            response.writeHeader(cache[request.url].Status,
                cache[request.url].Header);
        response.write(cache[request.url].Body);
        cFiles+=1;
        cTime+=Number(cache[request.url].Time);
        cBytes+=Number(cache[request.url].Bytes);
        consoleLog.push("Saved Bandwidth : " + cache[request.url]
            .Bytes);
        consoleLog.push("Saved Time : " + cache[request.url].Time);
        request.pipe(response);
        request.resume();
    } else {
        tstart=Date.now();
        var options = url.parse(request.url);
        var connector = https.request(options,
            function(serverResponse) {
            serverResponse.pause();
            response.writeHeader(serverResponse.statusCode,
                serverResponse.headers);
            serverResponse.pipe(response);
            hStatus=serverResponse.statusCode
            hHeader=serverResponse.headers
            serverResponse.resume();
            hData=''
            serverResponse.on('data', chunk => {
                hData+=chunk;
            });
            serverResponse.on('end',() => {
                wFiles+=1;
                consoleLog.push('request : ' + request.url);
                if (typeof serverResponse.headers['content-length']
                    =="undefined") {
                    wBytes=wBytes+0
                    tBytes=0;
                    consoleLog.push("Bytes not declared")
                } else {
                    tBytes=Number(serverResponse.headers[
                            'content-length'])
                        consoleLog.push("Bytes: " + tBytes)
                        wBytes=wBytes+tBytes
                        consoleLog.push(" Total Bytes so far : " + wBytes)
                    }
                    tend=Date.now()
                    consoleLog.push("Millisec: " + (tend-tstart))
                    wTime=wTime+(tend-tstart);
                    cache[request.url]={Body:hData,Header:hHeader,
                        Status:hStatus, Bytes:tBytes,Time:(tend-tstart)};
                    consoleLog.push(" Total Time so far : " + wTime);
                    consoleLog.push(" Total Files so far : " + wFiles);
                })
            });
            request.pipe(connector);
            request.resume();
        }
    } else {
        if (typeof blocked[request.url]=="undefined") {
            blocked[request.url]={Freq:1}
        } else {
            blocked[request.url]={Freq:blocked[request.url].Freq+1}
        }
        consoleLog.push(request.url +
            ' was blocked as it is containing the keyword ' + bkey);
        app.get('/', (req, res) => res.send(request.url
            + ' was blocked as it is containing the keyword ' + bkey))
        request.resume();
    }



########## PAGE FORMATTING #########
    mng="<html><body><form action='blacklist' method=
        'post'><input type='submit'>"
    mng+="<table><tr><td><strong>Retrieved from Web</strong><div>
```

```
        Total bytes : " + wBytes + "</div>"
        mng+="<div>Total milliseconds : " + wTime + "</div><div>
            Total Files : " + wFiles + "</div></td>"
    mng+="<td><strong>Savings by using Cache</strong><div>Total bytes :
        " + cBytes + "</div>"
        mng+="<div>Total milliseconds : " + cTime + "</div><div>
            Total Files : " + cFiles + "</div>"
        mng+="</td></tr>"
    mng+="<tr><td><strong>Blacklist</strong><br><textarea id='bl'
        rows='6' cols='50'>"
        black_list.forEach((v:any)=>{
            mng+=v+"\n"
        })
    mng+="</textarea></td>"
    mng+="<td><strong>Cached Files</strong><br><textarea id='cf'
        rows='6' cols='50'>"
        for (let key in cache) {
            mng+=key + "\n";
        }
    mng+="</textarea></td></tr>"
    mng+="<tr><td><strong>Messages</strong><br><textarea id='msg'
        rows='6' cols='50'>"
        for (let entry of consoleLog) {
            mng+=entry+ "\n";
        }
    mng+="</textarea></td>"
    mng+="<td><strong>Blocked Files</strong><br><textarea id='blocked'
        rows='6' cols='50'>"
        for (let key in blocked) {
            mng+=key + " , Accessed " +
                Number(blocked[key].Freq) + " times\n";
        }
    mng+="</textarea></td></tr></table></form></body><html>"
});
```

```
 ######### WEBSOCKET PROXY #########
wss.on('connection', (ws: WebSocket) => {
    ws.on('message', (message: string) => {
        if (message!=bkey) {
                var tstart=Date.now()
                        ws.send(`Requested URL -> ${message}`)
                        consoleLog.push("Bytes: " + message.length)
                        wBytes=wBytes+message.length
                        wFiles+=1;
                        var tend=Date.now()
                        consoleLog.push("Millisec: " + (tend-tstart))
                wTime=wTime+tend-tstart
                        consoleLog.push(`Requested URL -> ${message}`)
                } else {
                        var data="The URL " + message + " was blocked"
                        consoleLog.push(data)
                        consoleLog.push(`Blocked URL -> ${message}`)
                }
    });
    ws.send('Hi there, I am a WebSocket server');
});



 ######### LISTENER #########
app.get('/Proxy', function(req, res){
        res.send(mng);
});
app.post('/', function(req, res){
        console.log(res)
});
server.listen(4000, () => {
    console.log(`Server started on port 4000!`);
});
```

```
sserver.listen(4443, () => {
    console.log(`SSL Server started on port 4443!`);
});
app.listen(3000,() => {
        console.log(`Proxy Monitoring at localhost:3000/Proxy!`)
})
console.log("Proxy has started")
```