Website blocking code in c language:

**RESEARCH:**

BLOCKING OF WEBSITE

* Purpose : This program is designed to identify and restrict access to certain websites. Also it safety, enhances internet productivity, and helps manage online behaviour.
* Structure: this program handles input from user i.e

It captures URL input from user to block any unwanted websites.

References:

* <https://www.w3schools.com/>
* <https://www.geeksforgeeks.org/how-to-block-websites-in-browsers/>
* <https://www.hostinger.in/tutorials/how-to-block-a-website-on-chrome>

youtube references:

* <https://www.youtube.com/watch?v=a4S6FU5RdR0>
* <https://www.youtube.com/watch?v=xM60R9dWYUI>

pdfs referred:

* <https://people.iith.ac.in/rogers/pds_theory/lect20.pdf>
* <https://www.lenovo.com/in/en/glossary/hosts-file/?orgRef=https%253A%252F%252Fwww.google.com%252F&srsltid=AfmBOooI9ndWz7QF-rpf4qmAwfdpDlO1pmeXkli-IclUHFz15XiMUZpy>

**ANALYZE:**

Definitions of different terms used in code:

Hosts file in windows:

The hosts file is a simple text file on your computer that plays a crucial role in networking. It acts as a local DNS (Domain Name System) resolver, mapping domain names to IP addresses. In other words, it helps your computer find the correct IP address when you enter a website's domain name in your browser. It's like a phonebook for the internet, ensuring smooth communication between your computer and websites you visit.

**Basics of File Handling in C:**

-File handling in C is the process in which we create, open, read, write, and close operations on a file. C language provides different functions such as fopen(), fwrite(), fread(), fseek(), fprintf(), etc. to perform input, output, and many different C file operations in our program.

In C programming, you can specify a file path to access or manipulate files on the host system using functions from the standard library like fopen(), fclose(), fread(), fwrite(), and others. The file path tells the program where to find or create the file on the system.

File functions used here:

**-** fprintf():

In the C programming language, fprintf() sends formatted output to a file stream. The fprintf() function helps print content in a file instead of on the stdout console. The definition of the fprintf() function is included in stdio. h header file.

- fclose():

It is a standard library function used to close a file that was previously opened using functions like fopen(), freopen(), or tmpfile(). Closing a file ensures that all data is properly written to the file and that any system resources associated with the file are released.

-Fgets instead of scanf:

The fgets() reads a line from the specified stream and stores it into the string pointed to by str. It stops when either (n-1) characters are read, the newline character is read, or the end-of-file is reached, whichever comes first.

Buffers are memory storage regions that temporarily hold data while it is being transferred from one location to another. A buffer overflow (or buffer overrun) occurs when the volume of data exceeds the storage capacity of the memory buffer. As a result, the program attempting to write the [data to the buffer overwrites adjacent memory locations](https://www.imperva.com/learn/data-security/structured-and-unstructured-data/).

-stdlib.h:

The stdlib.h header file in C provides a collection of functions for performing various standard library operations, such as memory allocation, process control, conversions, and generating random numbers.

-Strcmp:

C strcmp() is a built-in library function that is used for string comparison. This function takes two strings (array of characters) as arguments,compares these two strings lexicographically, and then returns 0,1, or -1 as the result. It is defined inside <string.h> header file.

**-**Strcspn(complementary span):

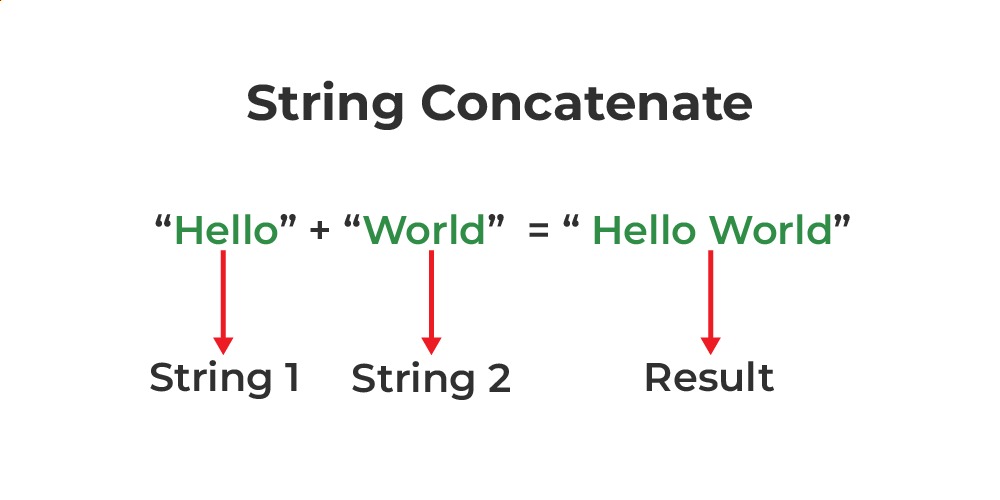
The C library function strcspn()calculates the length of the number of characters before the 1st occurrence of character present in both the string.

-Strcpy:

strcpy is a C standard library function that copies a string from one location to another. It is defined in the string.h header file.

-Strcat(string concatenate)**:**

It is a function that appends the string pointed to by src to the end of the string pointed to by dest. It will append a copy of the source string in the destination string. plus a terminating Null character. The initial character of the string(src) overwrites the Null-character present at the end of the string(dest).



-Strlen:

The strlen() function in C calculates the length of a given string. The strlen() function is defined in string.h header file. It doesn’t count the null character ‘\0’.

-The define() function defines a constant:

Constants are much like variables, except for the following differences:

* A constant's value cannot be changed after it is set
* Constant names do not need a leading dollar sign ($)
* Constants can be accessed regardless of scope
* Constant values can only be strings and numbers

Syntax

define(*name,value,case\_insensitive*)

- The **stdlib.h** header file in C provides a collection of functions for performing various standard library operations, such as memory allocation, process control, conversions, and generating random numbers.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

**IDEATE :**

This C program prompts the user to input a website URL and performs several string operations on the input. It first asks for the URL and checks if the input is valid using strcmp to ensure it's not empty. The program then copies the URL using strcpy to another variable for safekeeping and displays it. Afterward, it concatenates the website URL to a predefined message using strcat and prints the full message. Additionally, it calculates and prints the length of the website URL using strlen.

-Dynamic addition of blocked sites through user input.

* Logging functionality to keep track of accessed URLs.
* Categorization of sites (e.g., social media, gaming) for more targeted blocking.

**BUILD:**

**Code for blocking website:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define Max\_URL\_Length 256

#define Hosts\_File\_Path "C:\\Windows\\System32\\drivers\\etc\\hosts"

void blockWebsite(const char \*website) {

FILE \*hostsFile = fopen(Hosts\_File\_Path, "a");

if (hostsFile == NULL) {

printf("Failed to open the hosts file. Make sure to run as administrator.\n");

return;

}

fprintf(hostsFile, "127.0.0.1 %s\n", website);

fclose(hostsFile);

printf("Website %s is now blocked.\n", website);

}

int main() {

char website[Max\_URL\_Length];

char copiedWebsite[Max\_URL\_Length];

char message[Max\_URL\_Length + 50] = "Attempting to block website: ";

printf("Enter a website URL to block: ");

fgets(website, sizeof(website), stdin);

website[strcspn(website, "\n")] = 0;

if (strcmp(website, "") == 0) {

printf("No website entered. Please provide a valid URL.\n");

return 1;

}

strcpy(copiedWebsite, website);

printf("You entered: %s\n", copiedWebsite);

strcat(message, copiedWebsite);

printf("%s\n", message);

int urlLength = strlen(website);

printf("The length of the website URL is: %d characters.\n", urlLength);

blockWebsite(website);

return 0;

}

**Code for unblocking website:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define Max\_URL\_Length 256

#define Hosts\_File\_Path "C:\\Windows\\System32\\drivers\\etc\\hosts"

void unblockWebsite(const char \*website) {

FILE \*hostsFile = fopen(Hosts\_File\_Path, "r");

if (hostsFile == NULL) {

printf("Failed to open the hosts file. Make sure to run as administrator.\n");

return;

}

FILE \*tempFile = fopen("temp\_hosts.txt", "w");

if (tempFile == NULL) {

printf("Failed to create temporary file.\n");

fclose(hostsFile);

return;

}

char line[Max\_URL\_Length];

while (fgets(line, sizeof(line), hostsFile)) {

if (strstr(line, website) == NULL) {

fputs(line, tempFile);

}

}

fclose(hostsFile);

fclose(tempFile);

remove(Hosts\_File\_Path);

rename("temp\_hosts.txt", Hosts\_File\_Path);

printf("Website %s is now unblocked.\n", website);

}

int main() {

char website[Max\_URL\_Length];

printf("Enter a website URL to unblock: ");

fgets(website, sizeof(website), stdin);

website[strcspn(website, "\n")] = 0;

if (strcmp(website, "") == 0) {

printf("No website entered. Please provide a valid URL.\n");

return 1;

}

unblockWebsite(website);

return 0;

}

**TEST:**

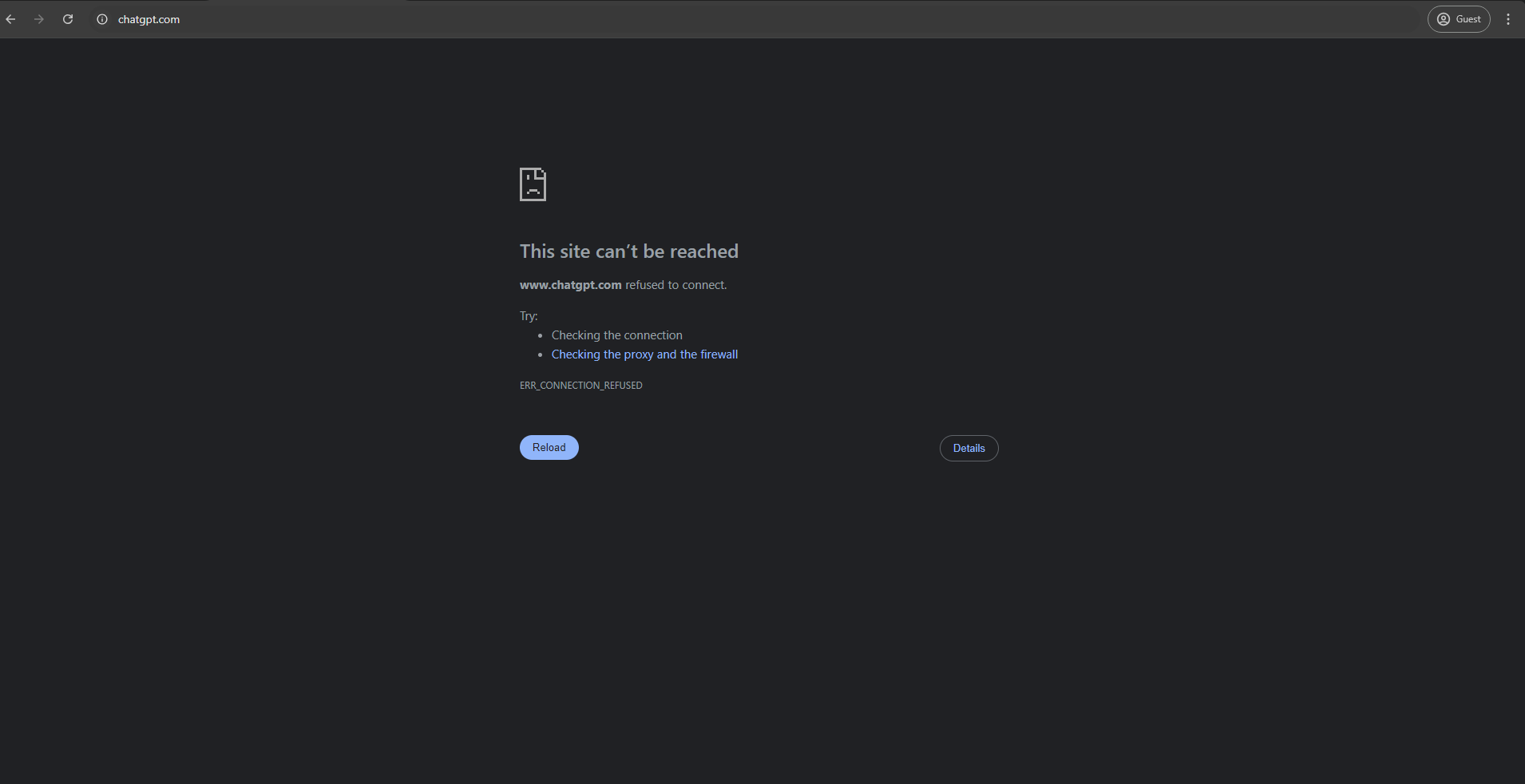
Enter a website URL to block: www.chatgpt.com

You entered: www.chatgpt.com

You are attempting to block the following website: www.chatgpt.com

The length of the website URL is: 15 characters.

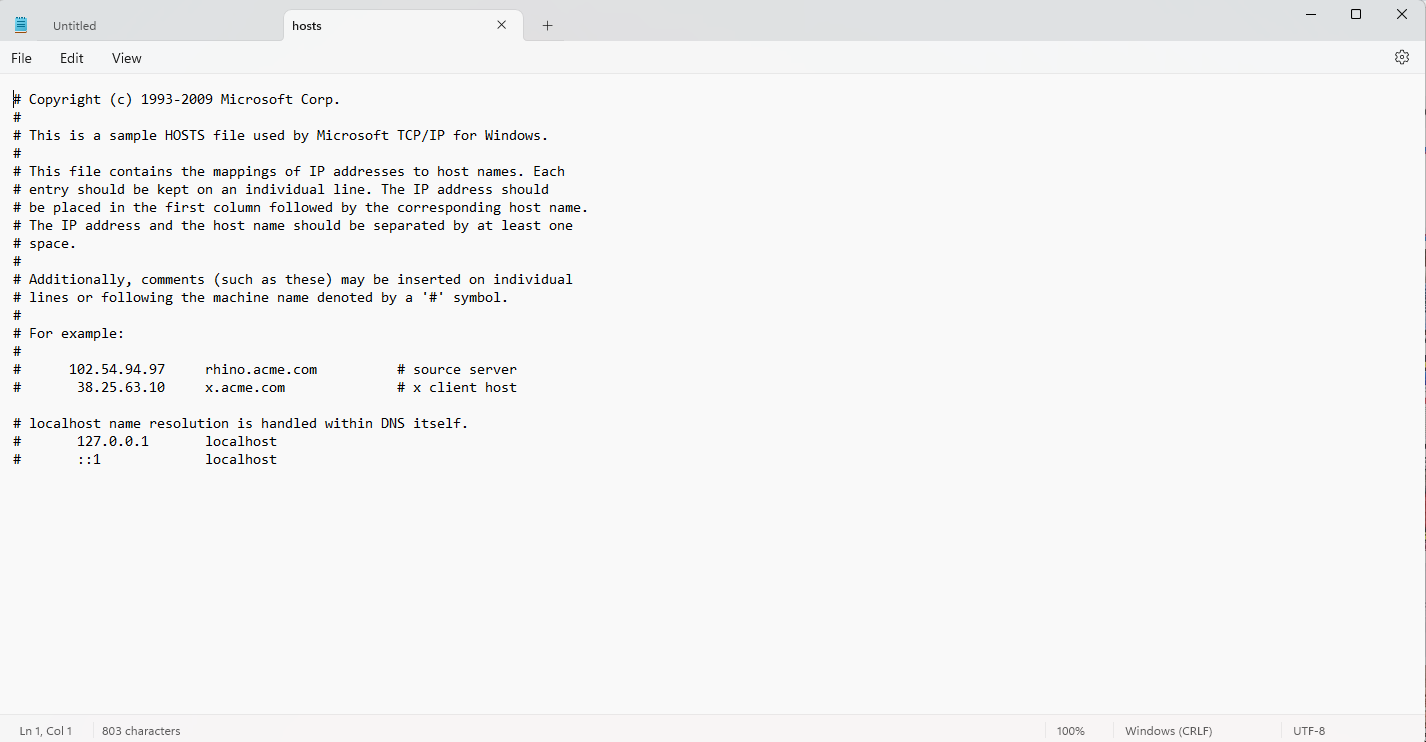
Website www.chatgpt.com is now blocked.



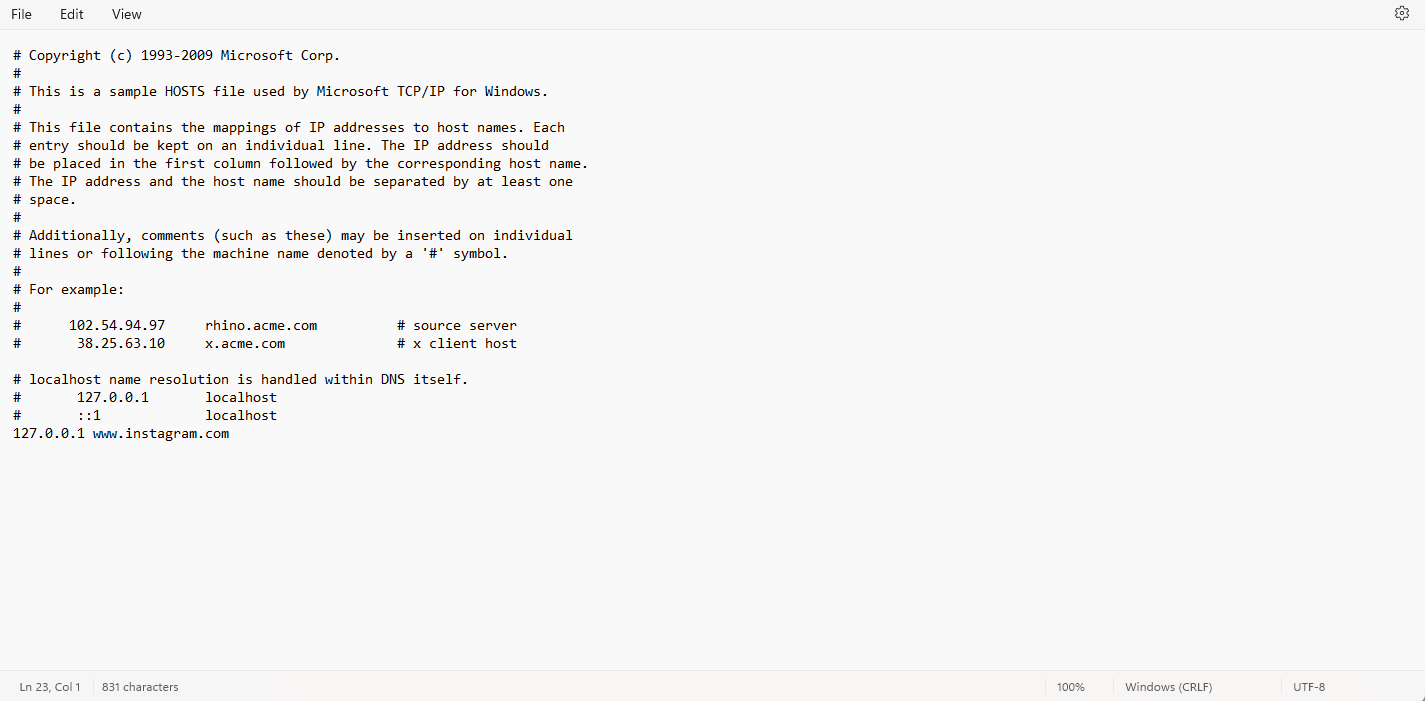
The program includes functionality to block websites by modifying the system's hosts file.

In general, the hosts file is a local configuration file used by the operating system to map domain names (like www.example.com) to IP addresses. By adding entries in the format 127.0.0.1 [website] to the file, the program effectively redirects any request for the specified website to the local machine (which is represented by the IP address 127.0.0.1). This makes the website inaccessible because the browser cannot reach the actual server. To modify the hosts file, the program would open it in append mode, add the blocking entry for the specified website, and then close the file. This approach requires administrative privileges since modifying system files like the hosts file is restricted to prevent unauthorized changes:

Hosts file before blocking website:



Hosts file after blocking website:



**IMPLEMENTATION:**

This code serves as a practical tool for managing local network settings by unblocking websites. Users should exercise caution when modifying system files and ensure they understand the implications of their changes.

This C program provides a straightforward method for unblocking websites by editing the hosts file. By understanding and utilizing this code, users can manage their local network settings effectively.

Also in many software companies and universities this code can be used to block particular websites to avoid malpractice and other cheating purposes.

DIFFERENT AREAS WHERE THIS CODE CAN BE IMPLEMENTED:

**1. Parental Control**

* **Protecting children**: Parents may block websites with inappropriate or harmful content, such as those related to violence, adult content, or unsafe social platforms.
* **Limiting distractions**: Some parents block certain websites to reduce distractions and encourage children to focus on studying or other productive activities.

**2. Workplace Productivity**

* **Preventing distractions**: Employers often block access to social media, gaming, or video streaming sites during work hours to maintain productivity.
* **Security reasons**: Blocking potentially harmful or unsecure websites prevents employees from accidentally downloading malware or phishing content that could harm the company's network.

**3. Security and Privacy**

* **Avoiding malicious sites**: Blocking websites known for hosting malware, phishing schemes, or other cyber threats helps secure personal or organizational devices and data.
* **Restricting unwanted content**: Blocking ad-heavy or tracking-heavy websites can improve privacy and reduce unwanted intrusions, such as pop-up ads or unauthorized tracking.

**4. Network Management**

* **Reducing bandwidth usage**: Schools, businesses, or other organizations may block streaming services, large file-sharing platforms, or other bandwidth-heavy sites to optimize internet speed for essential services.
* **Content filtering in public spaces**: Libraries, schools, or cafes might block websites to ensure appropriate usage of public Wi-Fi networks.

**5. Personal Productivity**

* **Avoiding distractions**: Individuals may block social media, video streaming, or gaming sites during study or work time to stay focused.
* **Time management**: Self-imposed website blocking tools, like productivity apps, help people control their online habits and stick to productive routines.

**6. Censorship or Content Control**

* **Government control**: Governments may block websites that share illegal content or content that violates national laws, such as websites promoting hate speech or illegal activities.
* **Political or social control**: In some cases, websites may be blocked to control the flow of information, such as during political unrest or to restrict access to dissenting views.

**7. Educational Purposes**

* **Focus on educational content**: Schools may block access to entertainment sites or social media to ensure that students use school networks primarily for educational purposes.

GITHUB LINK:

<https://github.com/swaradadeshpande/WEBSHIELD>