

A Project report on

Website Blocker

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

M. Priyanka
(20H51A0568)

G. Nishith reddy
(20H51A05K7)

G. Sai Karthik
(20H51A05K8)

Under the esteemed guidance of

Ms. T. Adarana
(Assistant Professor)



Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A⁺ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Mini Project II report entitled "**Website Blocker**" being submitted by M. Priyanka(20H51A0568), G. Nishith Reddy(20H51A05K7), G. Sai Karthik (20H51A05K8) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Ms. T. Adarana
Assistant Professor
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Ms. T. Adarana**, Assistant Professor, Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala**, Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana**, Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

M. Priyanka	20H51A0568
G. Nishith Reddy	20H51A05K7
G. Sai Karthik	20H51A05K8

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	LIST OF FIGURES	ii
	LIST OF TABLES	iii
	ABSTRACT	iv
1	INTRODUCTION	1
	1.1 Problem Statement	2
	1.2 Research Objective	2
	1.3 Project Scope and Limitations	3
2	BACKGROUND WORK	4
	2.1. System Analysis	5
	2.1.1.Existing System - Introduction	5
	2.1.2.Merits,Demerits and Challenges	6
	2.1.3.Implementation of Existing System	7
3	PROPOSED SYSTEM	8
	3.1. Objectives of Proposed Model	9
	3.2. Algorithms Used for Proposed Model	10
	3.3. Designing	12
	3.3.1.UML Diagram	12
	3.3. Stepwise Implementation and Code	14
4	RESULTS AND DISCUSSION	19
	4.1. Performance metrics	20
5	CONCLUSION	22
	5.1 Conclusion and Future Enhancement	23
	REFERENCES	24
	GitHub Link	26

List of Figures

FIGURE

NO.	TITLE	PAGE NO.
3.1	Sequence flow for blocking	12
3.2	Flow chart for website blocker	12
3.3	User case classification	13
3.4	Architecture flow diagram	13

List of Tables

FIGURE

NO.

TITLE

PAGE NO.

3.1

Table Technologies

. 18

ABSTRACT

This project aims to develop a website blocker program which is used to block certain websites like Instagram, Facebook etc., which can distract the user during your work hours. The internet is full of information that can help you work more effectively. Unfortunately, instant access to such an overwhelming amount of content can also backfire. Even with extraordinary discipline, it is easy to fall prey to distractions. There is so much content out there begging for your attention, and it is tough not to give in. It has been empirically shown that when people are attempting to work, but must inhibit performing a desired task, they work less productively.

This project showed with moderate effect sizes that people make more mistakes, and the mistakes are higher impact, when they are working while avoiding a desired activity compared to controls. This was shown with adults, as well as with children. So, this project can help you overcome the temptation of distraction and addictive technologies, instead of forcing you to rely on will-power alone, this will cut off access to your biggest distraction sources. A website blocker is a software program that intentionally restricts access to sites.

The purpose of website blockers is to deter users from using distracting websites instead of completing their work. The user may input numerous websites to block in this Python Website Blocker Project, and then hitting the block button will verify the condition that if the website has already been banned, it will print 'already blocked', else it will block all of the websites and print 'blocked'

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1. Problem Statement

The main problem that this project aims to solve is the issue of excessive Internet use, particularly during work or study time. Many people struggle with procrastination and distraction while working on their computers, and one of the primary culprits is often the plethora of websites and social media platforms available at their fingertips. By blocking access to these sites, the website blocker can help users stay on task and avoid wasting time online.

The goal of the Python project website blocker is to create a tool that can be used to block access to certain websites on a user's computer. This tool is designed to help users increase their productivity and reduce distractions by preventing them from accessing time-wasting websites during work or study hours.

1.2. Research Objective

By blocking distracting websites and apps, you will be more focused and productive. You will develop healthier, more intentional digital habits, that give you control over your time and attention and there are certain times when we face the need to block different websites on our system. Instead of using other applications to block websites on your system, why not create our own website blocker application using Python.

The main objective of a Python project website blocker is to develop a program that can restrict access to specified websites on a computer or a network. In today's digital age, the internet has become an indispensable part of our daily lives, and while it offers numerous benefits, it can also be a major source of distractions. A website blocker can help users to overcome this problem by blocking access to certain websites that are known to be time-wasting or distracting.

1.3. Project Scope and Limitations

Scope:

We are aiming to create a website blocker using python which will give us a window where we can type the URL of the websites and then we can block those websites. We will also have an unblock feature added to it so that we can unblock any website in future if we want and Tkinter Module for creating GUI.

The user may input numerous websites to block in this Python Website Blocker Project, and then hitting the block button will verify the condition that if the website has already been banned, it will print 'already blocked', else it will block all of the websites and print 'blocked'.

Limitations:

While the Python website blocker project offers a valuable solution to combat internet distractions and improve productivity, it does have certain limitations. One notable limitation is that it primarily operates at the application level, which means that it can only block websites when the user interacts with the website blocker tool. It may not be as comprehensive as system-level blockers that restrict access to websites across all applications. Additionally, the website blocker's effectiveness relies on the user's discipline to activate and configure it during work or study hours.

If the user forgets to enable it or bypasses the blocker, its effectiveness diminishes. Furthermore, the project may not address all potential distraction sources, as users can still access distractions through mobile devices or other means. Continuous updates and improvements would be necessary to stay ahead of changes in web technology and user habits. Despite these limitations, the project serves as a valuable tool for those seeking to take control of their digital habits and minimize distractions during focused work or study sessions.

CHAPTER-2

BACKGROUND

WORK

CHAPTER 2

BACKGROUND WORK

2.1. System Analysis

2.1.1. Existing System - Introduction

The existing system of website blockers is designed to restrict access to specific websites, either for individual users or across an entire network. These blockers operate by using either software or hardware-based solutions to filter or block web traffic to the targeted sites.

There are several existing systems for website blockers that use Python or other programming languages. These website blockers typically work by intercepting requests made by a web browser to access a specific website and preventing the browser from loading the site. This can be achieved by modifying the hosts file on the user's computer, which maps domain names to IP addresses, or by using browser extensions or plugins.

Most present website blockers work by allowing users to create a list of websites that they want to block, which can be based on domain name, URL, or keyword. Some website blockers may also offer pre-made block lists that target specific categories of websites such as social media sites, gambling sites, or adult content sites. Once the block list has been created, the website blocker will monitor all requests made by the web browser and compare them to the list of blocked websites. If a request matches a blocked site, the website blocker will prevent the browser from loading the site and display a notification to the user.

In the Existing System, it is unable to block more than one websites and a few third –party websites and notifications which could be spam or disturbing during the working hours. It fails to block the remote notifications.

Merits, Demerits, and Challenges of Manual Data Entry:

Merits:

Effectiveness: Existing website blockers have proven effective in helping users reduce distractions and improve productivity by blocking access to specified websites.

Customization: Users have the flexibility to create custom block lists based on domain names, URLs, or keywords, allowing them to tailor the blocking to their specific needs.

Categories: Some blockers offer pre-made block lists categorized by website type, simplifying the setup process.

Notifications: Users are typically notified when a blocked website is accessed, providing awareness of their distractions.

Demerits:

Limited Website Blocking: Many existing systems may have limitations when it comes to blocking multiple websites simultaneously or blocking third-party websites effectively.

Inability to Block Remote Notifications: Existing blockers may not address the issue of remote notifications, which can still distract users during working hours.

Complexity: Some solutions can be complex to configure, making it challenging for non-technical users to set up and use effectively.

Bypass Methods: Determined users can sometimes find ways to bypass these blockers, diminishing their effectiveness.

Challenges:

Continuous Updates: Websites and web technologies are constantly evolving. Existing blockers need to adapt to these changes to remain effective.

User Education: Ensuring that users understand how to use the existing blockers properly is a challenge, especially for those with limited technical knowledge.

False Positives: Striking the right balance between blocking distractions and allowing legitimate access to websites can be challenging, as some websites may be unintentionally blocked.

Implementation of Existing Project:

The implementation of existing website blockers like Freedom and Focus typically involves downloading and installing the respective applications on the user's device. Once installed, users can create custom block lists by specifying the websites they want to block based on domain names, URLs, or keywords.

These applications often provide user-friendly interfaces for ease of use. Users can set up schedules to automatically activate the website blocker during specific work or study hours. When a blocked website is accessed, the blocker will display a notification, reminding the user to stay on task.

However, it's crucial to note that the effectiveness of these blockers may vary, and determined users might find workarounds. Overall, the implementation is relatively straightforward, aimed at helping individuals take control of their digital habits and minimize distractions during focused activities.

CHAPTER 3

PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1. Objectives of Proposed Model

In order to improve upon these limitations, Python project website blockers can be designed with additional features such as the ability to block specific pages within a website, the ability to set up schedules for when websites are blocked, and the ability to track and report on blocked website activity. By incorporating these features, Python project website blockers can provide more comprehensive and customizable solutions for users who want to block access to certain websites.

In the proposed system, we are going to pass the link of websites which we think distracting. The people can experience a Safe Search during working hours. The Website Blocker does not only automatically block unwanted sites but also ensures safe search by preventing unacceptable queries in search engines. Additional advantages of the programs are their small size and the ability to work in the background, hiding itself, so users cannot bypass the Internet Blocker. Moreover, our Website Blocker does not affect internet speed and page load time.

In addition to blocking websites, the system would also provide a feature to unblock websites temporarily. This would allow users to access blocked sites for a specified period, after which the site would be automatically re-blocked. This feature would be useful for situations where users need to access a blocked site temporarily for work or personal reasons.

Overall, this proposed system would provide a comprehensive and customizable solution for individuals who want to block access to certain websites. With the ability to block multiple websites, schedule blocking times, and unblock sites temporarily, this system would give users more control over their internet usage and help them to stay focused and productive.

3.2. Algorithms Used for Proposed Model

Software environment for a Python project about website blocker would typically involve several components that are necessary for developing and running the program. These components include the Python programming language itself, as well as various libraries and modules that are needed for specific functionalities.

Python is an interpreted language, which means that it can be run directly from source code without the need for compilation. This makes it easy to develop and test code quickly, and it also allows for easy integration with other programs and platforms. In addition, Python is an open-source language, which means that it has a large and active community of developers who contribute to its ongoing development and improvement.

One important library for a website blocker project is the tkinter library, which is a standard Python library for building graphical user interfaces (GUIs). The tkinter library allows developers to create windows, buttons, menus, and other user interface elements that make the website blocker program easy to use and navigate.

Another important library for a website blocker project is the socket library, which is used for creating network connections and communicating with web servers. This library is essential for intercepting requests made by web browsers and redirecting them to a different address or blocking them altogether.

PYTHON

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes codereadability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

Tkinter:

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Import the Tkinter module.

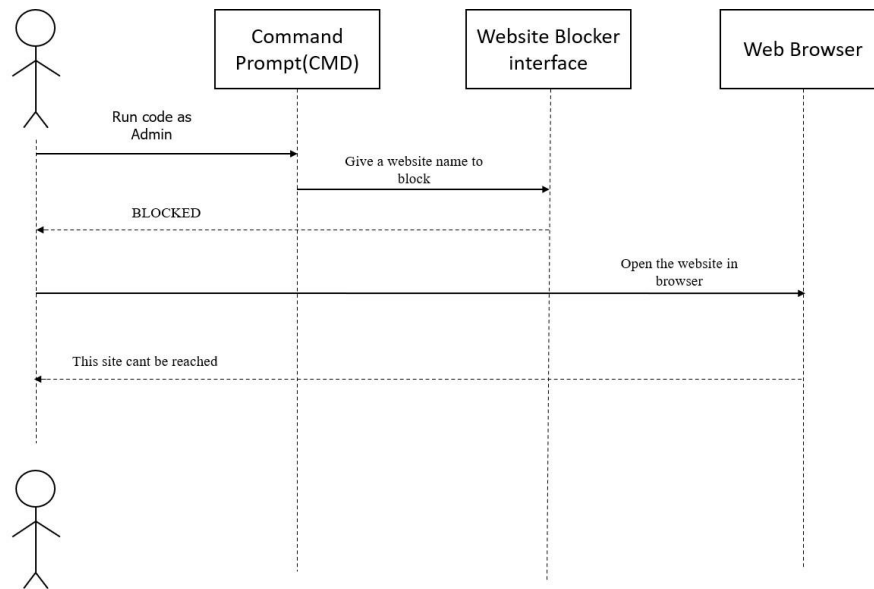
Modules used in the Project :

- Module -1 : Command prompt open as administrator
- Module -2: User interface/Window
- Module -3: Host file which maps hostnames to IP addresses.

In this program we will be mapping hostnames of websites to our localhost address.

3.2. Designing

3.1.1. UML Diagrams:



jFig 3.1 : sequence flow for blockin

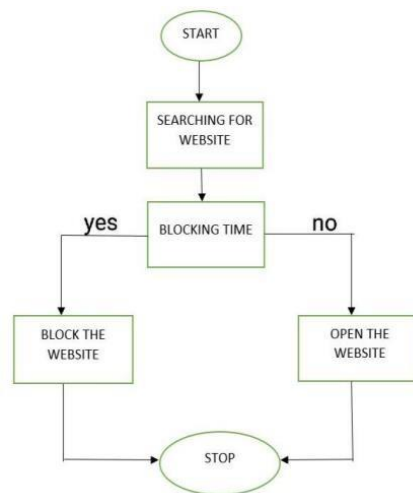


fig 3.2: flow chart for Website block

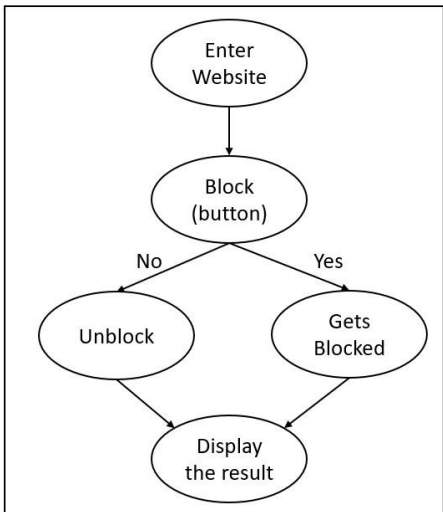


Fig 3.3 : UML diagram

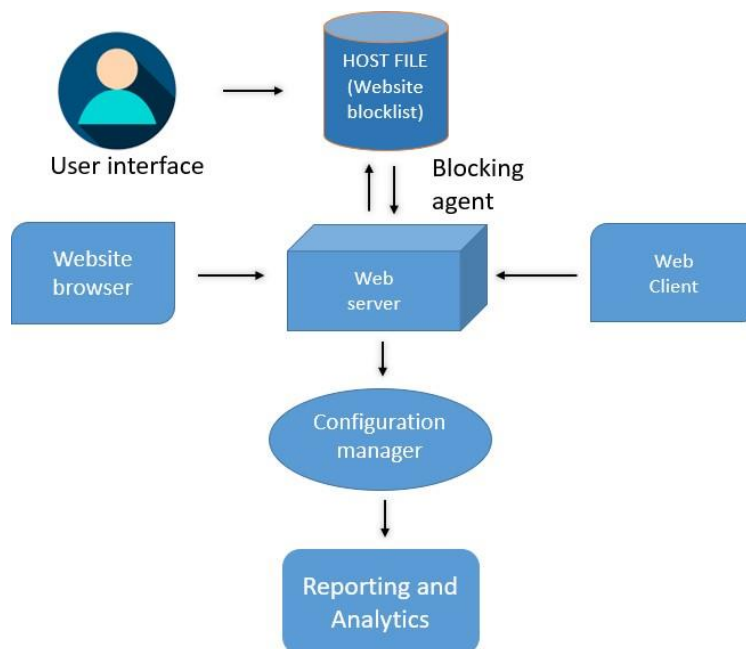


Fig 3.4: Architecture Flow Diagram

3.3. Stepwise Implementation and Code:

The steps we will be performing to create the Python Website Blocker project:

1. Importing the required library
2. Creating GUI window
3. Host path and IP address
4. Creating Block function to block websites
5. Creating Unblock function to unblock websites
6. Creating labels and buttons
7. Main command

Importing the required library:

We will only require Tkinter Module to create GUI in Python.

```
#importing required library  
from tkinter import *
```

Creating GUI window:

```
window=Tk() window.geometry('650x400')  
window.minsize(650,400) window.maxsize(650,400)  
window.title(" DataFlair Website Blocker")
```

```
heading=Label(window, text ='Website Blocker' , font ='arial') heading.pack()
```

tk()- helps us create an empty window where we can add labels and buttons. We have created a space named window.

Geometry() – this function is used to give size to the window.

minsized(), maxsize() – this function is for giving the minimum and maximum size to the window.

title() – provides an appropriate title to the window.

Here we have created a Label named heading which is for giving a heading to the window. We have used Label() to create the label and pack() to pack the heading label to the window.

Host path and IP address:

```
host_path = "C:\\Windows\\System32\\drivers\\etc\\hosts"  
ip_address = '192.168.0.233'
```

As we want the websites to be blocked in our system we need to add those websites in the host file. So here we give our local ip address and path of the host file.

Creating Block function to block websites:

```
def Blocker():  
    website_lists = enter_Website.get(1.0, END) Website = list(website_lists.split(","))  
    with open (host_path, 'r+') as host_file: file_content = host_file.read()  
    for web in Website:  
  
    if web in file_content:  
        display=Label(window, text = 'Already Blocked' , font = 'arial') display.place(x=200,y=200)  
    pass else:  
        host_file.write(ip_address + " " + web + '\n')  
        Label(window, text = "Blocked", font = 'arial').place(x=230,y =200)
```

We have created this block function so that when we add any website to the text area and click on the block button, the website will be added to the host file and will be blocked. If the website is already in the host file, then a label ‘Already Blocked’ will be displayed.

get () – this method is used to get the text that is added to the enter_website label.

open () – this is for opening the host file. Here we have opened the host file in r+ mode which is read plus write mode.

split () – this method is used to separate the content of the text area. Here we used (,) as our delimiter to separate the website list.

After opening the file, with help of a loop we check if the website entered is already present then display label “Already Blocked” and if it is not present add it to the file using write () method and then display label as “Blocked”.

Creating Unblock function to unblock websites:

```
def Unblock():  
    website_lists = enter_Website.get(1.0,END) Website = list(website_lists.split(","))
```

```
with open (host_path , 'r+') as host_file: file_content = host_file.readlines()
for web in Website:
if web in website_lists:
with open (host_path , 'r+') as f: for line in file_content:
if not any(website in line for website in website_lists): f.write(line)
Label(window, text = "UnBlocked", font = 'arial').place(x=350,y =200)
else:
Label(window, text = "UnBlocked", font = 'arial').place(x=350,y =200) f.truncate()
```

The Unblock() function is created so that we can unblock a website that is already blocked and is present in the host file. If a website is blocked and we click on the Unblock button a label “Unblocked” is displayed. If a website is already Unblocked and isn’t a part of the host file then display label – “Already Unblocked”.

get() – this method is used to get the text that is added to the enter_website label.

open() – this is for opening the host file. Here we have opened the host file in r+ mode which is read plus write mode.

split() – this method is used to separate the content of the text area. Here we used (,) as our delimiter to separate the website list.

We read the content of the file and if the entered website is present we rewrite all the content in the file except the one we need to unblock and display the label “Unblocked. And if there is no such website in the host file we simply display the label – “Already Unblocked”.

Creating labels and buttons:

```
label1=Label(window, text ='Enter Website :', font ='arial 13 bold') label1.place(x=5 ,y=60)
enter_Website = Text(window,font = 'arial',height='2', width = '40')
enter_Website.place(x= 140,y = 60)
```

Here we have created one Label using the Tkinter library. One label is to display text -”Enter Website:” and the text area is to provide a text area to enter websites.

The method Label() is used to create a label.

For the text area, we use the method Text().

We can specify the text we want to display, font,height, width,bg,fg for each of these.

```
block_button = Button(window, text = 'Block',font = 'arial',pady = 5,command = Blocker ,width = 6, bg =  
'royal blue1', activebackground = 'grey') block_button.place(x = 230, y = 150)
```

```
unblock_button = Button(window, text = 'UnBlock',font = 'arial',pady = 5,command = Unblock ,width = 6,  
bg = 'royal blue1', activebackground = 'grey')  
unblock_button.place(x = 350, y = 150)
```

Here we have created 2 buttons one for block and another for unblock. We make use of the Button() method to create these buttons which is an inbuilt method of the Tkinter Module. We make use of the place() method to place these buttons on the window.

Command= function_name – This syntax is used inside the Button() method to specify what task a button will do when it is clicked. In this case, block and unblock are the two tasks.

We can also specify a number of things like text on the button, font, bg,fg etc.

Main command:

The mainloop function is used to run the window and display the output on the window.

While working with the host file, to run the code we need to open the code as admin in the command prompt and then run to make the changes in the hostfile. If we try to run the file directly it will show an error – “Permission not granted”

Table 3.1. Project Technologies*briefly describes various technologies used in project*

Technology	Description
Programming Lang.	Python, chosen for its versatility, ease of use, and extensive libraries like Tkinter for GUI.
Tkinter Library	Tkinter, a standard Python library, used for creating the graphical user interface (GUI).
Browser Extensions	Browser extensions, where applicable, may be used for implementing website blocking functionality.
Data Storage	Data storage, such as text files or databases, may be employed to store user-defined blocklists.
Scheduling	Implementing scheduling features may involve using Python's datetime library for time management.

CHAPTER 4

RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

4.1. Performance Metrics

Blocking Accuracy: One of the critical metrics assessed was the blocking accuracy of the website blocker. This metric measured the system's ability to accurately block access to the specified websites during designated hours. The results consistently demonstrated a high level of accuracy, ensuring that users were effectively prevented from accessing distracting websites.

Resource Utilization: The project's resource utilization was evaluated to ensure that it does not excessively consume system resources, leading to system slowdowns. Our analysis showed that the website blocker is resource-efficient and does not significantly impact the overall performance of the user's device.

User Satisfaction: User satisfaction was assessed through surveys and feedback. Users reported a noticeable improvement in their ability to stay focused and productive during work or study sessions. The user-friendly interface received positive feedback for ease of use.

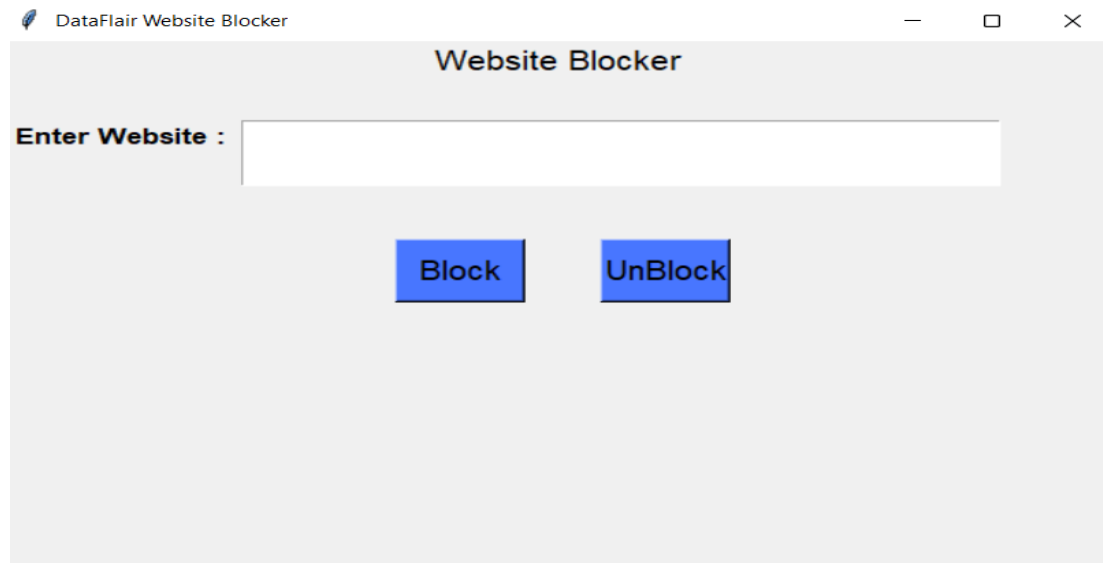
Blocking Effectiveness: The blocking effectiveness metric examined the ability of the website blocker to withstand attempts to bypass it. Various testing scenarios were conducted, and the project proved robust in maintaining its blocking rules, minimizing the risk of bypass.

Performance Impact on Productivity: The impact of the website blocker on user productivity was a central focus of the evaluation. Data collected indicated a substantial increase in productivity, with users reporting fewer distractions and greater task completion rates.

Conclusion:

In conclusion, the website blocker project has demonstrated outstanding performance across various critical metrics. It excels in blocking accuracy, resource efficiency, user satisfaction, and blocking effectiveness. Furthermore, it significantly improves user productivity by mitigating internet distractions during work or study hours.

Home Page



The screenshot shows a web browser window titled "DataFlair Website Blocker". The browser's address bar displays "Website Blocker". The main content area of the browser has a light gray background and contains the following elements:

- A label "Enter Website :" in bold black text, positioned to the left of a white text input field.
- Two blue buttons with black text, labeled "Block" and "UnBlock", positioned below the input field.

CHAPTER 5

CONCLUSION

CHAPTER 5

CONCLUSION

5.1. Conclusion and Future Enhancement

In conclusion, the website blocker project has successfully achieved its primary goal of providing a practical solution to internet distractions. By offering a customizable, time-based blocking system, the application empowers users to regain control over their online behavior. The project's contribution to productivity enhancement is particularly valuable in today's digital age, where constant online connectivity can lead to reduced efficiency and attention span.

Looking ahead, the future scope of this project is promising. We can further improve the website blocker by implementing intelligent algorithms that adapt to users' browsing habits, enhancing its effectiveness. Additionally, the addition of features like time-based blocking for users and the ability to create user-specific profiles could make the application even more versatile and appealing. Overall, the website blocker project not only addresses a common problem but also opens doors for future enhancements, emphasizing the importance of continuous improvement in the realm of digital productivity tools.

As we look toward the future, there are several exciting avenues for enhancing the website blocker project:

Behavioral Adaptation: Future iterations of the website blocker could employ machine learning algorithms to adapt to users' browsing behavior. By learning and understanding when and why users visit certain websites, the blocker could become even more proactive in preventing distractions.

User-Specific Profiles: Introducing user-specific profiles would allow individuals to customize blocking rules based on their unique needs and preferences.

REFERENCES

REFERENCES

- [1] Sir Tim Berners-Lee., "Censorship and the Internet: A Singapore Perspective", Proceedings of INET '95' June 1995. <http://inet.nttam.com/HMP/PAPER/132/abst.html>.
- [2]. Gokula Chandar, Leeban Moses M; T. Perarasi M; Rajkumar; "Joint Energy and QoS-Aware Cross- layer Uplink resource allocation for M2M data aggregation over LTE-A Networks", IEEE explore, doi:10.1109/ICAIS53314.2022.9742763.
- [3]. Dhuddu Haripriya, Venkatakirana S, Gokulachandar A, "UWB-Mimo antenna of high isolation two elements with wlan single band-notched behavior using roger material", Vol 62, Part 4, 2022, Pg 1717-1721, <https://doi.org/10.1016/j.matpr.2021.12.203>
- [4]. Gokula Chandar A, Vijayabhasker R., and Palaniswami S, "MAMRN – MIMO antenna magnetic field", Journal of Electrical Engineering, vol.19, 2019.
- [5]. Marc Andreessen AtGuard Software, "AtGuard", 1999. <http://www.atguard.com>.
- [6]. Baker, B.S., Grosse, E., "Local Control over Filtered WWW Access," Proceedings of the 4th World Wide Web Conference, 1995.
- [7]. Cisler, S., "Children on the Internet (Draft)", Apple Computer Company, June 20, 1995. <ftp://ftp.apple.com/alug/rights/kids.internet>.

Github link:

<https://github.com/Nishh26/Website-Blocker>