**CHAPTER 1**

**INTRODUCTION**

**1.1 PROJECT OVERVIEW:**

* The website blocker/unblocker application project is developed under the category of utility software.
* Utility software helps manage, maintain ,control a computer and support it’s infrastructure.
* Every functionality of the computer uses a utility software.
* Utility software is set of tools that helps configure, analyse and optimize computer resources to help users perform multiple tasks efficiently.
* Most of the operating systems have input build-in utility tools, but additional software further ensure improved functionality.
* Utility software focuses on computer components such a hardware, software operating system, network configuration and storage parts.
* Example : data generators, hex editors, html checkers and merge programs
* The website blocker and unblocker application uses URL filtering technique that merged with simple GUI as a utility application.
* This application runs only with administrator right

**1.2 EXISTING SYSTEM WITH DISADVANTAGES**

**Existing System:**

* The local administrator needs to manually enter to the website name with localhost IP address in the host file to block that particular website.
* Also the local administrator can write a script for blocking the websites but requires to modify the script if new websites are needed to block.
* The process to change the script will be tedious if the script is automated with OS tasks

**Disadvantages:**

1. Requires manual operation to add or remove website in host file
2. The script needs to be modified when new websites are added
3. Requires more effort if the script is automated with the OS tasks
4. Once the script is automated websites cannot be unlocked /blocked at any time as per local administrator wishes.

**1.3 PROBLEM DEFINITION**

**User Interface:**

* Since no UI is present , the websites are that required to block is achieved through writing script and executing them

**Period of Time:**

* Even though the script is executed and automated using windows automation utility the websites will be blocked for a certain period of time

**1.4 PROPOSED SYSTEM WITH ADVANTAGES**

**Proposed System:**

* Using this application project the local administrator just needs to enter the website name to block/unblock using GUI
* Once blocked/unblocked the application effect will be permanent in the system
* Local administrator can use this application with ease as it provides with simple GUI layout
* The local administrator does not need to setup a schedule task to block or unblock the website in the system

**Advantages:**

1. Simple to use
2. Scheduled blocking is not required
3. Application effect will be permanent
4. Websites can be blocked/unblocked at any time as per local administrator wishes

**1.5 OBJECTIVES**

* To block the social media websites during the working hours
* To block the webistes without writing a script and automating it
* To increase the effectiveness of work during working hours
* To Regulate access to web content by users connected to the networks of organizations
* To aim at not to cause any unintended consequences to third parties.

**1.6 SCOPE**

* Administrator can block the websites which are distractful during working hours
* This application can be used on OS like windows , mac and linux
* Mainly deployed as a service for parental control purposes, and for regulating the access to Web content by users connected to the networks of enterprises, libraries, schools, etc.
* Very effective at identifying content that may be on different servers or services because the URL doesn’t change even if the server changes IP addresses.
* Does not cause unintended consequences to third parties

**CHAPTER 2**

**LITERATURE REVIEW**

**1. Base Paper Name :** Internet Society Perspectives on Internet Content Blocking : An Overview

**Author name / Published by :** ISOC

**Published year :** 24 March 2017

**Abstract :**

* Understanding the different blocking techniques, their effects and side effects, is important both for policy makers considering the use of such measures and for Internet advocates and others wishing to influence content blocking practices.
* All blocking techniques are prone to two main drawbacks:

1. **They do not solve the problem**
2. **They inflict collateral damage**

* The Internet Society believes the most appropriate way to counteract illegal content and activities on the Internet is to attack them at their source.
* Using filters to block access to online content is inefficient, likely to be ineffective, and is prone to generate collateral damage affecting innocent Internet users.
* two main strategies to counteract the illegal content on the Internet:

1. **Attack the issue at the source**
2. **Prioritize and use alternative approaches**

**2. Base Paper Name :** The Effectiveness of Internet Content Filters

**Author name / Published by :** Philip B Stark

**Published year :** January 2008

**Abstract :**

* As part of its defense of the Child Online Protection Act, which seeks to prevent minors from viewing commercially published harmful-to-minors material on the World Wide Web, the U.S. Department of Justice commissioned a study of the prevalence of “unwanted" materials and the effectiveness of Internet content filters in blocking them.
* As of 2005–2006, about 1.1% of webpages indexed by Google and MSN were unwanted—hundreds of millions of pages and about 6% of a set of 1.3 billion searches executed on AOL, MSN and Yahoo! in summer 2005 .
* It retrieved at least one unwanted webpage among the first ten results, and about 1.7% of those results are unwanted webpages.
* These estimates are based on both simple random samples of webpages indexed by search engines and on a stratified random sample of searches.
* Webpages intended with unwanted content were used to test a variety of Internet content filters for underblocking—failing to block webpages that they are intended to block.
* A random sample of "clean" webpages with no unwanted content or reference to unwanted content was used to test the filters for overblocking—blocking webpages they are not intended to block.
* Webpages retrieved by the most popular searches according to Wordtracker were also categorized and used to test filters.
* Generally, filters with lower rates of underblocking had higher rates of overblocking.
* If the filter most effective at blocking adult materials were applied to search indexes, typical query results, or the results of popular queries, the number of clean pages blocked in error would exceed the number of adult pages blocked correctly.

**3. Base Paper Name :** Comparative Study on Blocking , Filtering and Take down of Illegal Internet Content

**Author name / Published by :** Swiss Institute of Comparative Law

**Published year :** 20 December 2015

**Abstract :**

* The study consists, essentially, of two main parts.
* The first part represents a compilation of country reports for each of the Council of Europe member states.
* It presents a more detailed analysis with to respect filtering, blocking and takedown of illegal content on the internet in each member state.
* For ease of reading and comparison, each country report follows a similar structure.
* The second part contains comparative considerations on the practices in the member states in respect of filtering, blocking and takedown of illegal online content.
* The purpose is to identify and to attempt to explain possible convergences and divergences between the member states’ approaches to the issues included in the scope of the study.
* Blocking achieved through a number of techniques, including the blocking of the DNS or the URL.
* Takedown or removal of internet content, on the other hand, will instead broadly refer to demands or measures aimed at the website operator (or “host”) to remove or delete the offending website content or webpages.
* Both internet access providers and website hosts may be referred to as “internet intermediaries”.

**4. Base Paper Name :** Internet Content Filtering Guideline – Department of Education and Training (Title : Guidelines for Managing Access to the Content on the Internet)

**Author name/Published by :** Victoria State Government

**Published year :** 22 October 2018

**Abstract :**

* It outlines the guidelines for managing access to internet content by all users on the Department's corporate network.
* Access to internet content is controlled using an internet filtering service
* It provides outlines for managing the categories of content that are allowed and blocked by the internet filtering service.
* It also provides outlines for managing exceptions.
* The internet filtering service uses categories to allow or block access to websites or web pages on internet sites.
* The filtering service uses the content of information on the internet site to determine the category for the site.
* The filtering service will allow access to a site whose category is allowed, and block access to a site in a category that is blocked.
* Access to sites may also be blocked on advice of the government and non-government security specialists to protect the information assets of the Department.
* Blocking access to internet sites to protect Department information assets will be managed by ICT security specialists in the Department.
* Blocking may occur both on the individual computer, or in a network device between the computer and the rest of the Internet.

**5. Base Paper Name :** RFC 7754-Technical Consideration for Internet service Blocking and Filtering

**Author name/Published by :**

1.Richard barnes

2.Alissa cooper

3.Olaf kolkman

4.Dave thaler

5.Erik nordmark

**Published year :** March 2016

**Abstract :**

* The Internet is structured to be an open communications medium.
* This openness is one of the key underpinnings of Internet innovation, but it can also allow communications that may be viewed as undesirable by certain parties.
* Thus, as the Internet has grown, so have mechanisms to limit the extent and impact of abusive or objectionable communications.
* There has been an increasing emphasis on "blocking" and "filtering", the active prevention of such communications.
* Several technical approaches to Internet blocking and filtering in terms of their alignment with the overall Internet architecture is examined .
* When it is possible to do so, the approach to blocking and filtering that is most coherent with the Internet architecture is to inform endpoints about potentially undesirable services, so that the communicants can avoid engaging in abusive or objectionable communications.
* It is observed that certain filtering and blocking approaches can cause unintended consequences to third parties.

**6. Base Paper Name :** Research on interactive technology of Webpage block localization based on JavaScript

**Author name/Published by:** Peng Nie , Ruixuan Wang , Jialin Dong , Yuchen Zheng

**Abstract :**

* With the development of web applications, the main way for people to obtain information is gradually transferred to the network.
* XPath can be used to uniquely represent a certain position in a web page for those who are familiar with web applications.
* However, it is difficult to interact with a computer to locate the content of a web page for most ordinary people who benefit from the web.
* Thus, a webpage block localization technology based on JS scripts which is convenient for human-computer interaction is proposed.
* The experiments prove that it shows advanced performance both in interaction mode and block mode.
* Also, the scheme can be integrated with crawler technology to monitor a webpage block through the interaction of webpage block localization, and it can be applied to the problem feedback module of webpage to facilitate users' feedback and suggestions.

**7. Base Paper Name :** Intelligent web content filtering system using MAS

**Author name/Published by :** Abbas M. Al-Bakry , Talib T. Al-Fatlawii

**Abstract :**

* The WWW has become infinite repository of information over the world because of the vast amount of information that they can gain.
* Also the access to the web enables any user easily to get information anytime and anywhere.
* But with these advantages, there is also some disadvantages such as the some websites contains harmful material like terrorism, hate, violence, and pornographic.
* Therefore developing a system that can filter these contents is become necessary.
* This paper introduces a web content filtering using multi agents, in this system FCM algorithm and some webpage feature in classification process is used.
* Experimental results based on (2000) different websites are given us an efficient results.
* On the testing set, the system was able to achieve correct acceptance and reject of 97.5%.

**8. Base Paper Name :** Centralized content-based Web filtering and blocking: how far can it go ?

**Author name/Published by :** Chen Ding , Chi-Hung Chi , Jing Deng , Chun-Lei Dong

**Abstract :**

* To an organisation, centralized Internet filtering and blocking is very important.
* Educators and parents would like to block offensive materials from children.
* Companies also want to reduce the amount of work time that employees spend on non-productive Web surfing.
* Current blocking and filtering mechanisms can roughly be classified into two approaches: URL-based and content filtering.
* In the URL-based approach, a requested URL address is blocked if a match is found in the blocked list.
* However, keeping the list up-to-date is very difficult.
* In the content filtering approach, keyword matching is often used.
* Its main problem is misblocking.
* Many desirable Web sites are blocked because some predefined keywords appear in their Web pages, though in different meaning or context.

* There are suggestions for image, audio and video understanding in real-time content filtering.
* The delay time is also of great concern.
* This paper, investigates how far multimedia content analysis should go for Internet filtering and blocking.
* A set of guidelines for defining the heuristics used in real-time Web content analysis is also given.
* These heuristics not only have higher filtering accuracy than most multimedia retrieval techniques do, but they also have a comparable runtime overhead to that of keyword matching.
* The experience of deploying a adult content filtering system in high schools is also described.
* Experience from the system's implementation and deployment is found to give a very good direction to the centralized filtering and blocking of Web content.

**9. Base Paper Name :** Block Me If You Can: A Large-Scale Study of Tracker-Blocking Tools

**Author name/Published by :** Chen Ding , Chi-Hung Chi , Jing Deng , Chun-Lei Dong

**Abstract :**

* This paper, quantifies the effectiveness of third-party tracker blockers on a large scale.
* First, the architecture of various state-of-the-art blocking solutions is analyzed and the advantages and disadvantages of each method are discussed.
* Second, a two-part measurement study on the effectiveness of popular tracker-blocking tools is performed.
* The analysis quantifies the protection offered against trackers present on more than 100,000 popular websites and 10,000 popular Android applications.
* This provide novel insights into the ongoing arms race between trackers and developers of blocking tools as well as which tools achieve the best results under what circumstances.
* Among others, it is discovered that rule-based browser extensions outperform learning-based ones
* Trackers with smaller footprints are more successful at avoiding being blocked, and CDNs pose a major threat towards the future of tracker-blocking tools.
* Overall, the contributions of this paper advance the field of web privacy by providing not only the largest study to date on the effectiveness of tracker-blocking tools
* Also they highlight the most pressing challenges and privacy issues of third-party tracking.
* Blocking usually is effective at high-level URLs, such as a particular web page,
* But is not as effective when deep links (such as individual bits of content within a web page) are considered.

# 10. Base Paper Name : A General Framework for Web Content Filtering

**Author name/Published by :** Elisa Bertino

**Abstract :**

* Web content filtering is a means to make end-users aware of the ‘quality’ of Web resources by evaluating their contents and/or characteristics against users’ preferences.
* Although they can be used for a variety of purposes, Web content filtering tools are mainly deployed as a service for parental control purposes, and for regulating the access to Web content by users connected to the networks of enterprises, libraries, schools, etc.
* Current Web filtering tools are based on well established techniques, such as data mining and firewall blocking, and they typically cater to the filtering requirements of very specific end-user categories.
* Therefore, what is lacking is a unified filtering framework able to support all the possible application domains, and making it possible to enforce interoperability among the different filtering approaches and the systems based on them.
* In this paper, a multi-strategy approach is described, which integrates the available techniques and focuses on the use of metadata for rating and filtering Web information.
* Such an approach consists of a filtering meta-model, referred to as MFM, which provides a general representation of the Web content filtering domain, independently from its possible applications.
* Two of its prototype implementations, is partially carried out in the framework of the EU projects EUFORBIA and QUATRO, and designed for different application domains:

1. user protection and
2. Web quality assurance, respectively.

* Blocking requires the blocking party (such as the user’s ISP) to have the ability to intercept and control traffic between the end-user and the Internet.

**CHAPTER 3**

**REQUIREMENTS SPECIFICATIONS**

**3.1 INTRODUCTION**

* The requirements specification is a technical specification of requirements for the software products.
* It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements.
* The requirements also provide usage scenarios from a user, an operational and an administrative perspective.
* The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals.
* This describes the project target audience and its user interface, hardware and software requirements.
* It defines how the client, team and audience see the project and its functionality.

**3.2 HARDWARE AND SOFTWARE SPECIFICATIONS**

**3.2.1 Software used**

* Operating System : Windows 10 Pro
* OS Version : 21H2
* OS build : 19044.1645
* Language : Python 3.10.3
* UML Software : Rational Rose
* Documentation : MS-Office
* Shader Version : 5.1
* Microsoft DirectX Version : 12
* Vulkan Version : 1.2.177
* Graphics Output Protocol Version : 9.0.1085

**3.2.2 Hardware used**

* Processor : Intel ® Core™ i5-8265U
* Clock rate : 1.60GHz
* Maximum clock rate : 1.80GHZ
* System type : 64-bit operating system, x64- based processor
* RAM used : 16GB (15.9 GB usable)
* Hard drive : 1 TB
* Monitor type : 1366x768 color display
* GPU : Intel® UHD Graphics 620
* Graphics Driver Version : 27.20.100.9664

**3.3 TECHNOLOGIES USED**

* This chapter is about the software language and the tools used in the development of the project.
* The platform used here is Python**.**
* The software in built modules used are Tkinter and Platform
* Python provides a rich application framework that allows you to build innovative things.

**3.3.1 Python**

* **Python** is a high-level, interpreted, general-purpose programming language.
* Its design philosophy emphasizes code readability 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.
* Python consistently ranks as one of the most popular programming languages.
* Python offers multiple options for developing GUI .

**3.3.2 Tkinter**

* Out of all the GUI methods, tkinter is the most commonly used method.
* It is a standard Python interface to the Tk GUI toolkit shipped with Python.
* Python with tkinter is the fastest and easiest way to create the GUI applications.
* Creating a GUI using tkinter is an easy task.
* The [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit.
* Both Tk and [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) are available on most Unix platforms, including macOS, as well as on Windows systems.
* Tcl/Tk is not a single library but rather consists of a few distinct modules, each with separate functionality and its own official documentation.
* Python’s binary releases also ship an add-on module together with it.

**3.3.2.1 Tcl**

* Tcl is a dynamic interpreted programming language, just like Python.
* Though it can be used on its own as a general-purpose programming language, it is most commonly embedded into C applications as a scripting engine or an interface to the Tk toolkit.
* The Tcl library has a C interface to create and manage one or more instances of a Tcl interpreter, run Tcl commands and scripts in those instances, and add custom commands implemented in either Tcl or C.
* Each interpreter has an event queue, and there are facilities to send events to it and process them.
* Unlike Python, Tcl’s execution model is designed around cooperative multitasking, and Tkinter bridges this difference.

**3.3.2.2 Tk**

* Tk is a Tcl package implemented in C that adds custom commands to create and manipulate GUI widgets.
* Each [Tk](https://docs.python.org/3/library/tkinter.html#tkinter.Tk) object embeds its own Tcl interpreter instance with Tk loaded into it.
* Tk’s widgets are very customizable, though at the cost of a dated appearance.
* Tk uses Tcl’s event queue to generate and process GUI events.

**3.3.2.3 Ttk**

* Themed Tk (Ttk) is a newer family of Tk widgets that provide a much better appearance on different platforms than many of the classic Tk widgets.
* Ttk is distributed as part of Tk, starting with Tk version 8.5.
* Python bindings are provided in a separate module, [tkinter.ttk](https://docs.python.org/3/library/tkinter.ttk.html#module-tkinter.ttk).
* Internally, Tk and Ttk use facilities of the underlying operating system, i.e., Xlib on Unix/X11, Cocoa on macOS, GDI on Windows.
* When your Python application uses a class in Tkinter, e.g., to create a widget, the [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) module first assembles a Tcl/Tk command string.
* It passes that Tcl command string to an internal \_tkinter binary module, which then calls the Tcl interpreter to evaluate it.
* The Tcl interpreter will then call into the Tk and/or Ttk packages, which will in turn make calls to Xlib, Cocoa, or GDI.

**3.3.3 Platform**

* Python defines an in-built module **platform** that provides system information.
* The **Platform module** is used to retrieve as much possible information about the platform on which the program is being currently executed.
* Now by platform info, it means information about the device, it’s OS, node, OS version, Python version, etc.
* This module plays a crucial role when you want to check whether your program is compatible with the python version installed on a particular system or whether the hardware specifications meet the requirements of your program.
* This module already exists in the python library and does not require any installation using **pip**.
* It is used to access the underlying platform’s data,  
  such as, hardware, operating system, and interpreter version information.
* The platform module includes tools to see the platform’s hardware, operating  
  system, and interpreter version information where the program is running.
* This module is used to perform compatibility checks.
* When we have a Python program that requires certain conditions to be met, e.g., the architecture of the processor, the operating system in use, or the Python version the system has, then this module can be used.

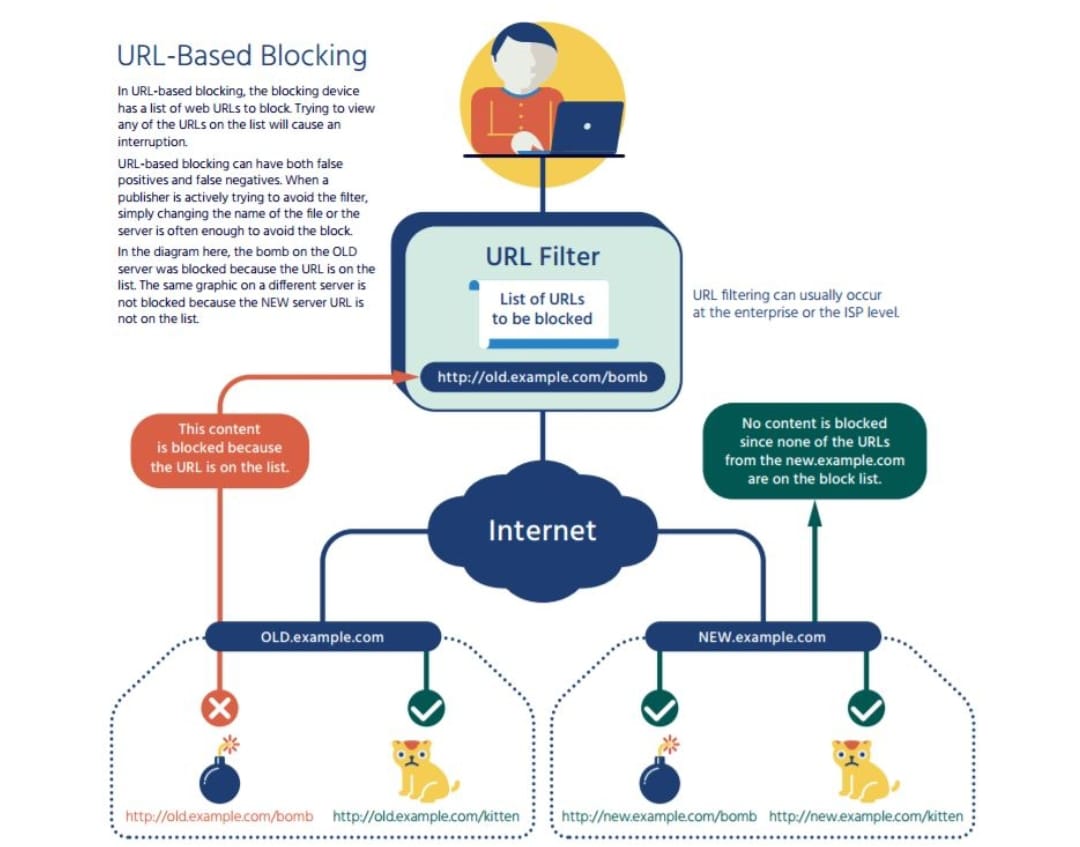
**CHAPTER 4**

**SYSTEM DESIGN AND ARCHITECTURE**

**4.1 GENERAL**

* Design Engineering deals with the various UML diagrams for the implementation of project.
* Design is the place where quality is rendered in software engineering.
* Design is the means to accurately translate customer requirements into finished product.
* Design is a meaningful engineering representation of a thing that is to be built.
* Software design is a process through which the requirements are translated into representation of the software.
* Design specification describes how a system performs the requirements outlined in the Functional Requirements.
* Depending on the system, this can include instructions on performing specific requirements, configuration settings or review of functions or code.
* Design is a meaningful engineering of something that is to be built.
* System design is therefore the process of defining and developing systems to satisfy specified requirements of the user
* System design is the process of art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.
* System design contains logical design and physical Design.
* Logical Designing describes the structures& characteristics or features, like input, output, files, database & procedures.

**4.2 ARCHITECTURE DIAGRAM**



***Fig 4.2: Architecture Diagram***

**4.3 DESCRIPTION OF SYSTEM ARCHITECTURE**

* This project uses the concept of URL Based Filtering Algorithm as a System Architecture

**4.3.1 URL BASED FILTERING SYSTEM ARCHITECTURE**

* URL-based blocking is a very popular blocking method, and may occur both on the individual computer, or in a network device between the computer and the rest of the Internet.
* URL blocking works with web-based applications, and is not used for blocking non-web applications (such as VoIP).
* With URL blocking, a filter intercepts the flow of web (HTTP) traffic and checks the URL, which appears in the HTTP request, against a local database or on-line service.
* Based on the response, the URL filter will allow or block the connection to the web server requested.
* Generally, URLs are managed by category (such as “sports sites”) and an entire category is blocked, throttled, or allowed.
* In the case of a national policy requiring URL blocking, the on-line service and blocking policy would likely be managed by the government.
* The URL filter can simply stop the traffic, or it can redirect the user to another web page, showing a policy statement or noting that the traffic was blocked.
* URL blocking in the network can be enforced by proxies, as well as firewalls and routers.
* URL blocking requires the blocking party (such as the user’s ISP) to have the ability to intercept and control traffic between the end-user and the Internet.
* URL blocking is usually expensive, because the filtering device generally has to be in-line between the user and the Internet, and thus requires a high level of resources to give acceptable performance.
* URL blocking is generally considered to be very effective at identifying content that may be on different servers or services because the URL doesn’t change even if the server changes IP addresses.
* In a few cases, URL blocking may fail to fully block the traffic when the URLs are very complicated or change frequently.
* This can happen because an information publisher has deliberately decided to actively evade URL filter blocking, or it can be a side effect of some advanced publishing systems such as those used for large on-line publications.
* URL blocking usually is effective at high-level URLs, such as a particular web page, but is not as effective when deep links (such as individual bits of content within a web page) are considered.
* Depending on how the user navigated to the particular content, URL blocking may or may not be able to block all access—if the user has a “deep link” not covered by the URL filter, the content will be allowed.
* For example, the Playboy web site includes both playboy.com URLs, but also embedded content using the “playboy.tv” domain name.
* A URL filter that didn’t also include “playboy.tv” URLs would not block the video content.
* All types of URL blocking are highly dependent on the quality of the filter, and a poorly designed or overly broad filter may block unintended traffic or have other negative effects on the user experience, such as affecting the loading or formatting of web pages when some component is being blocked.
* URL blocking requires some type of proxy to see the full URL when traffic is encrypted with HTTPS (TLS/SSL).
* For encrypted traffic, URL blocking can only see the IP address of the server, and not the full URL, resulting in a much higher level of unintended blocking.
* Because proxies are expensive and intrusive to the user experience, URL blocking does not work well as a tool for policybased blocking.
* In URL-based blocking, the blocking device has a list of web URLs to block.
* Trying to view any of the URLs on the list will cause an interruption
* URL-based blocking can have both false positives and false negatives.
* When a publisher is actively trying to avoid the filter , simply changing the name of the file or the server is often enough to avoid the block.
* URL filtering can usually occur at the enterprise or the ISP level.
* Example: the bomb on the OLD server was blocked because the URL is on the list.
* The same graphic on a different server is not blocked because the NEW server URL is not on the list.
* URL filtering is enabled through local database lookups, or by querying a master cloud­-based database.
* Local lookups on a limited, but frequently accessed, number of websites­ ensure maximum in-line performance and minimal latency for the most frequently­ accessed URLs, while cloud lookups provide coverage for the latest sites.
* To account for firms’ unique traffic patterns, on-device caches store the most recently accessed URLs, with the ability to also query a master database in the cloud for URL category­ information when an on-device URL is not found.

**4.4 UNIFIED MODELLING LANGUAGE**

* The UML is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.
* The UML offers a way to visualize a system‘s architectural blueprints in a diagram.
* UML provides a comprehensive notation for the full life cycle of object oriented design documentation, the UML has been extended to cover a larger set of design documentation, and been found useful in many contexts

**ADVANTAGES**

* To represent complete systems using object oriented concepts.
* To establish an explicit coupling between concepts and executable code.
* To creating a modeling language usable by both human and critical system.
* The class model captures the static structure.
* The state model expresses the dynamic behavior of objects.
* The use case model describes the requirements of the user.
* The implementation model shows the work units.

**4.4.1 Use Case Diagram:**

* A use case diagram is a type of behavioral diagram created from a Use-case analysis.
* The purpose of use case is to present overview of the functionality provided by the system in terms of actors, their goals and any dependencies between those use cases.
* A use case is a set of scenarios that describing an interaction between a user and a system.
* A use case diagram displays the relationship among actors and use cases.
* The two main components of a use case diagram are use cases and actors.
* An actor is represents a user or another system that will interact with the system modeled.
* A use case is an external view of the system that represents some action the user might perform in order to complete a task.

****

***Fig 4.4.1 : Use Case Diagram***

**4.4.2 Class Diagram:**

* Class diagram is UML structure diagram which shows structure of the designed system at the level of classes and interfaces, shows their features constraints and relationships -associations, generalizations, dependencies, etc .

****

***Fig 4.4.2 : Class Diagram***

**4.4.3 Sequence Diagram:**

* A sequence diagram shows, as parallel vertical lines (“lifelines”), different processes or objects that live simultaneously, and as the horizontal arrows, the messages exchanged between them, in the order in which they occur.
* This allows the specification of simple run time scenarios in a graphical manner.

****

***Fig 4.4.3 Sequence Diagram***

**4.4.4 Collaboration Diagram:**

* A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time.
* Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined.
* The relationships between the objects are shown as lines connecting the rectangles.
* The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.
* Collaboration diagrams are best suited to the portrayal of simple interactions among relatively small numbers of objects.
* These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

****

***Fig 4.4.4: Collaboration Diagram***

**4.4.5 Activity Diagram:**

* Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.
* In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflow of components in a system.
* An activity diagram shows the overall flow of control.
* It captures the dynamic behavior of the system.
* The flow can be sequential, branched, or concurrent.

****

***Fig 4.4.5 : Activity Diagram***

**4.4.6 State Machine Diagram:**

* State machine diagram is a behavior diagram which shows discrete behavior of a part of designed system through finite state transitions.
* State machine diagrams can also be used to express the usage protocol of part of a system.
* It is used to give the different states in the sequence order.
* Whenever the event occur, the object transfer one state to another state.



***Fig 4.4.6 : State Machine Diagram***

**4.4.7 Component Diagram:**

* A component diagram depicts how components are wired together to form larger components and or software systems.
* They are used to illustrate the structure of arbitrarily complex systems.
* It is used to identify the sub components of the system.
* It is a statistical view diagram.
* It represents modules part of system that encapsulates its content and whose manifestation is replaceable within its network.



***Fig 4.4.7 : Component Diagram***

**4.4.8 Deployment Diagram:**

* Deployment diagram shows execution architecture of systems that represent the assignment (deployment) of software [artifacts](http://www.uml-diagrams.org/deployment-diagrams.html#artifact) to deployment targets (usually [nodes](http://www.uml-diagrams.org/deployment-diagrams.html#node)).
* There are two types of Nodes:

1. Device Node
2. Execution Environment Node

* It is used to give the interaction between the software components to static containers.
* It is assignment of concrete software artifacts to computational nodes.
* It is deployment of software elements to physical architecture and communication between physical elements.



***Fig 4.4.8 : Deployment Diagram***

**CHAPTER 5**

**ALGORITHM AND METHODOLOGY**

**5.1 ALGORITHM**

* The Website blocker/unblocker uses URL Based Filtering Algorithm

**5.1.1 Algorithm for website blocker application:**

1. Local administrator enters multiple websites to block
2. Then clicking on the block button it will check the condition that if the website is already blocked
3. If already blocked then it displays ‘already blocked’
4. Else if blocks all the mentioned websites and displays ‘blocked’
5. To verify if the given website is blocked open the host file of the local system used and check for the website name ;
6. If present the website is blocked successfully else the website is not yet blocked

**5.1.2 Algorithm for website unblocker application:**

1. Local administrator enters one website at a time to unblock
2. Then clicking on the unblock button will check the condition that if the website is already unblocked
3. If already unblocked then it displays ‘already unblocked’
4. Else if unblocks the mentioned website and displays ‘unblocked’
5. To verify if the given website is unblocked open the host file of the local system used and check for the website name
6. If absent the website is unblocked successfully else the website is not yet unblocked

**5.2 METHODOLOGY**

**5.2.1. Host file:**

* Every system has host file whether it is mac, windows or linux.
* Host file location in mac and linux :

/etc/hosts

* Host file location in windows:

C:\windows \system32\drivers\etc\hosts

**5.2.2. Working :**

* Host is an operating system file which maps hostnames of websites to our localhost address .
* Using python file handling manipulation the hostname will be written in or removed from hosts.txt of the respective os
* After the host file is saved the mentioned websites in the file will be blocked and the websites that aren’t in the file remains unblocked.
* Local administrator enters multiple websites to block or enters one website at a time to unblock
* With URL blocking, a filter intercepts the flow of web (HTTP) traffic and checks the URL, which appears in the HTTP request, against a local database or on-line service.
* Based on the response, the URL filter will allow or block the connection to the web server requested.
* The URL filter can simply stop the traffic, or it can redirect the user to another web page, showing a policy statement or noting that the traffic was blocked.

**CHAPTER 6**

**MODULES**

**6.1 MODULES USED**

The Website Blocker/Unblocker contains the following modules

1. website\_blocker.py
2. website\_unblocker.py
3. hostfilePath
4. runapp1
5. Blocker
6. Unblocker
7. runapp2

**6.2 MODULE DESCRIPTION**

**Blocker:**

* This module gets the website to be blocked from the user by matching the host website ip address to local host ip address
* After the host website ip address is matched to local host ip address it is written to the host file of the respective OS used
* The absolute path of host file is retrieved from the module ‘ hostfilePath ‘ .

**Unblocker:**

* This module gets the website to be unblocked from the user by matching the website name needed to be unblocked to the websites present in the host file.
* After the website name is matched to host file websites it is removed from the host file of the respective OS used
* The absolute path of host file is retrieved from the module ‘ hostfilePath ‘ .

**Runapp1:**

* This module builds up the GUI for the website\_blocker.py module and website\_unblocker.py module.
* After retrieving the absolute path of host file , the runapp module starts to build a simple GUI window using the python in-built module ‘tkinter’ .
* The GUI window contains tkinter elements like textarea , window title name , label and a button.
* This GUI window also displays messagebox using the in-built python module ‘tkinter.messagebox’ to display the website is already blocked/unblocked (or) website is blocked/unblocked successfully.
* This module is used for blocking websites in a particular system or workstation

**Runapp2:**

* This module builds up the GUI for the website\_blocker.py module and website\_unblocker.py module.
* After retrieving the absolute path of host file , the runapp module starts to build a simple GUI window using the python in-built module ‘tkinter’ .
* The GUI window contains tkinter elements like textarea , window title name , label and a button.
* This GUI window also displays messagebox using the in-built python module ‘tkinter.messagebox’ to display the website is already blocked/unblocked (or) website is blocked/unblocked successfully.
* This module is used for unblocking websites in a particular system or workstation

**HostfilePath:**

* This module returns the absolute path location of host file in the currently working system.
* First it finds the OS of the working system using in-built python module ‘ platform ’ and based on the OS , host file path is returned to the application
* If the OS of the working system is Windows then it returns the host file path as C:\windows \system32\drivers\etc\hosts to run
* If the OS of the working system is Linux (or) Mac then it returns the host file path as /etc/hosts
* If the OS of the working system is not identified then it displays ‘ Operating system cannot be recognized in this system ’ .

**Website\_unblocker.py:**

* This module is used to run the website unblocker application to unblock the websites as the local administrator wishes.
* This module is responsible for finding the hostfile path for the OS used in the system , building a simple GUI window for application and removing the website name to the hostfile.
* Sub modules used in website\_unblocker.py:

1. hostfilePath

2. runapp2

3. Unblocker

**Website\_blocker.py:**

* This module is used to run the website blocker application to block the websites as the local administrator wishes.
* This module is responsible for finding the hostfile path for the OS used in the system , building a simple GUI window for application and writing the website name to the hostfile.
* Sub modules used in website\_blocker.py:

1. hostfilePath

2. runapp1

3. Blocker

**CHAPTER 7**

**TESTING AND CODING**

**7.1 CODING STANDARDS**

* Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program.
* They make the code easier to read, understand and maintain.
* This phase of the system actually implements the blueprint developed during the design phase.
* The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary.
* Some of the standard needed to achieve the above-mentioned objectives are as follows:
  + - Program should be simple, clear and easy to understand.
    - Naming conventions
    - Value conventions
    - Script and comment procedure
    - Message box format
    - Exception and error handling

**7.1.1 Naming Conventions**

* Naming conventions of classes, data member, member functions, procedures etc., should be self-descriptive.
* One should even get the meaning and scope of the variable by its name.
* The conventions are adopted for easy understanding of the intended message by the user.
* So it is customary to follow the conventions.
* These conventions are as follows:
* **Class names**
* Class names are problem domain equivalence and begin with capital letter and have mixed cases.
* **Member Function and Data Member name**
* Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

## 7.1.2 Value Conventions

* Value conventions ensure values for variable at any point of time. This involves the following:
  + - * Proper default values for the variables.
      * Proper validation of values in the field.
      * Proper documentation of flag values.

## 7.1.3 Script Writing and Commenting Standards

* Script writing is an art in which indentation is utmost important.
* Conditional and looping statements are to be properly aligned to facilitate easy understanding.
* Comments are included to minimize the number of surprises that could occur when going through the code.

## 7.1.4 Message Box Format

* When something has to be prompted to the user, he must be able to understand it properly.
* To achieve this, a specific format has been adopted in displaying messages to the user.
* They are as follows:
* X – User has performed illegal operation.
* ! – Information to the user.

**7.2 TEST PROCEDURE**

**7.2.1 System Testing**

* Testing is performed to identify errors.
* It is used for quality assurance.
* Testing is an integral part of the entire development and maintenance process.
* The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself.
* For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected.
* Detection of design faults can be achieved by means of inspection as well as walkthrough.
* Testing is one of the important steps in the software development phase.
* Testing checks for the errors, as a whole of the project testing involves the following test cases:
* Static analysis is used to investigate the structural properties of the Source code.
* Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

**7.3 TEST DATA AND OUTPUT**

**7.3.1 Unit Testing**

* Unit testing is conducted to verify the functional performance of each modular component of the software.
* Unit testing focuses on the smallest unit of the software design (i.e.), the module.
* The white-box testing techniques were heavily employed for unit testing.

## 7.3.2 Functional Testing

* Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.
* Three types of tests in Functional test:
  + - * Performance Test
      * Stress Test
      * Structure Test

## 7.3.3 Performance Test

* It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

## 7.3.4 Stress Test

* Stress Test is those test designed to intentionally break the unit.
* A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.
* It determines the robustness of software by testing beyond the limits of normal operation.
* Stress testing is particularly important for critical software but is used for all types of software.

## 7.3.5 Integration Testing

* Integration test is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product.
* The objective is to take untested modules and build a program structure tester should identify critical modules.
* Critical modules should be tested as early as possible.
* One approach is to wait until all the units have passed testing, and then combine them and then tested.
* This approach is evolved from unstructured testing of small programs.
* Another strategy is to construct the product in increments of tested units.
* A small set of modules are integrated together and tested, to which another module is added and tested in combination and so on.
* The advantages of this approach are that, interface dispenses can be easily found and corrected.
* The major error that was faced during the project is linking error.
* When all the modules are combined the link is not set properly with all support files.
* Then we checked out for interconnection and the links.
* Errors are localized to the new module and its intercommunications
* The product development can be staged, and modules integrated in as they complete unit testing.
* Testing is completed when the last module is integrated and tested.

**7.3.6 Testing Techniques/Testing Strategies**

1. **Testing**

* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that has a high probability of finding an as-yet – undiscovered error.
* A successful test is one that uncovers an as-yet- undiscovered error.
* System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences.
* It verifies that the whole set of programs hang together.
* System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system.
* This is the last chance to detect and correct errors before the system is installed for user acceptance testing.
* The software testing process commences once the program is created and the documentation and related data structures are designed.
* Software testing is essential for correcting errors.
* Otherwise the program or the project is not said to be complete.
* Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding.
* A successful test is one that uncovers an yet undiscovered error.
* **Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is[Defect](https://www.guru99.com/defect-management-process.html)free.
* It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest.
* The purpose of testing is to identify errors, gaps or missing requirements in contrast to actual requirements.
* Some prefer saying testing definition as a White Box and Black Box Testing.
* In simple terms,Testing means the Verification of AUT.
* Any engineering product can be tested in one of the two ways:

## White Box Testing

* This testing is also called as Glass box testing.
* In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function.
* It is a test case design method that uses the control structure of the procedural design to derive test cases.
* Basis path testing is a white box testing.

Basis path testing:

* + Flow graph notation
  + Cyclometric complexity
  + Deriving test cases
  + Graph matrices Control

## Black Box Testing

* In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised.
* It fundamentally focuses on the functional requirements of the software.
* The steps involved in black box test case design are:
  + Graph based testing methods
  + Equivalence partitioning
  + Boundary value analysis
  + Comparison testing

## Software Testing Strategies

* A software testing strategy provides a road map for the software developer.
* Testing is a set activity that can be planned in advance and conducted systematically.
* For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:
  + Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
  + Different testing techniques are appropriate at different points in time.
  + The developer of the software and an independent test group conducts testing.
  + Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

## Integration Testing Methodologies

* Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with.
* Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together.
* The problem of course, is “putting them together”- interfacing.
* There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

## Program Testing

* The logical and syntax errors have been pointed out by program testing.
* A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written.
* An improperly defined field dimension or omitted keywords are common syntax error.
* These errors are shown through error messages generated by the computer.
* A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations.
* Since the compiler s will not deduct logical error, the programmer must examine the output.
* Condition testing exercises the logical conditions contained in a module.
* The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses , a relational operator or on arithmetic expression.
* Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

## Security Testing

* Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration.
* The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack.
* During security, the tester places the role of individual who desires to penetrate system.

## Validation Testing

* At the culmination of integration testing, software is completely assembled as a package.
* Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins.
* Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer.
* Software validation is achieved through a series of black box tests that demonstrate conformity with requirement.
* After validation test has been conducted, one of two conditions exists.
  + The function or performance characteristics confirm to specifications and are accepted.
  + A validation from specification is uncovered and a deficiency created.
  + Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies.

## User Acceptance Testing

* User acceptance of the system is key factor for the success of any system.
* The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required via input and output screen.
* **UAT** is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment.
* UAT is done in the final phase of testing after functional, integration and system testing is done.
* The main **Purpose of UAT** is to validate end to end business flow.
* It does not focus on cosmetic errors, spelling mistakes or system testing.
* User Acceptance Testing is carried out in a separate testing environment with production-like data setup.
* It is kind of black box testing where two or more end-users will be involved.

**CHAPTER 8**

**CONCLUSION**

* This project helps to block and unblock n websites in the system that uses windows/linux/mac OS.
* The local users cannot access the websites that are blocked by the local administrator of the respective system.
* The administrator can block and unblock at his/her own will that doesn’t require scheduling.
* Misleading and distractive websites that reduce the productivity of the users can blocked during working hours.
* If the users need to access the blocked website they need to request the local administrator of the system to unblock that website.
* This application is based on URL-based blocking , can be used both on the individual computer, or in a network device between the computer and the rest of the Internet.
* It is very effective at identifying content that may be on different servers or services because the URL doesn’t change even if the server changes IP addresses.
* It can simply stop the traffic, or it can redirect the user to another web page, showing a policy statement or noting that the traffic was blocked.
* The local administrator does not need to setup a schedule task to block or unblock the website in the system.

**CHAPTER 9**

**APPENDIX**

**9.1 Appendix 1 : Code**

**hostfilePath.py:**

def hostfilePath():

system\_os = platform.system()

if system\_os == 'Linux':

path = '/etc/host'

elif system\_os == 'Windows':

path = 'C:\Windows\System32\drivers\etc\hosts'

else:

showinfo('Unidentified Operating System', message = 'Operating System in this device cannot be recognized')

return path

if \_\_name\_\_ == '\_\_main\_\_':

hostfilePath()

**Blocker.py:**

def Blocker():

websiteList = website.get(1.0,END)

blocked\_website\_list = list(websiteList.split(","))

with open (path,'r+') as host\_file:

file\_content = host\_file.read()

for web in blocked\_website\_list:

if web in file\_content:

showinfo('Website already blocked',message='The entered website(s) are already blocked ')

pass

else:

host\_file.write(ip+ " " + web + '\n')

showinfo('Website blocked',message='The entered website(s) are blocked successfully')

if \_\_name\_\_ == '\_\_main\_\_':

Blocker()

**Unblocker.py:**

def Unblocker():

unblock\_website\_temp = blocked\_website.get(1.0,END)

unblock\_website = unblock\_website\_temp.split(",")

for i in range(len(unblock\_website)):

unblock\_website[i] = ip + " " + unblock\_website[i]

with open (path,'r+') as host\_file:

content = host\_file.readlines()

host\_file.seek(0)

if unblock\_website[0] not in content:

showinfo('Website not found',message='Website not yet blocked')

else:

for line in content:

if not any(website in line for website in unblock\_website):

host\_file.write(line)

host\_file.truncate()

showinfo('Website unblocked',message='Website has been unblocked successfully')

if \_\_name\_\_ == '\_\_main\_\_':

Unblocker()

**runapp1.py**

import Blocker as block

import hostfilePath as pathfinder

root = Tk()

root.geometry('500x300')

root.resizable(0,0)

root.title("Website blocker")

Label(root,text='Website blocker',font='arial 20 bold').pack()

ip = '127.0.0.1'

Label(root,text='Enter Website :',font='arial 13 bold').place(x=5,y=60)

website = Text(root,font='arial 10',height='2',width='40')

website.place(x=140,y=60)

block=Button(root,text='Block',font='arial 12 bold',pady=5,command=block.Blocker,width=6,bg='#5DADE2',activebackground = 'sky blue')

block.place(x=230,y=150)

def runapp1(path):

root.mainloop()

if \_\_name\_\_ == '\_\_main\_\_':

path = pathfinder.hostfilePath()

runapp1(path)

**website\_blocker.py**

import runapp1 as ra1

import hostfilePath as hfp

if \_\_name\_\_ == ‘\_\_main\_\_’:

path = hfp.hostfilePath()

ra1.runapp(path)

**runapp2.py:**

import Unblocker as unblock

import hostfilePath as pathfinder

root = Tk()

root.geometry('500x300')

root.resizable(0,0)

root.title("Website blocker")

Label(root,text='Website unblocker',font='arial 20 bold').pack()

ip = '127.0.0.1'

Label(root,text='Enter One Website :',font='arial 13 bold').place(x=5,y=60)

website = Text(root,font='arial 10',height='2',width='40')

website.place(x=140,y=60)

unblock=Button(root,text='Block',font='arial 12 bold',pady=5,command=unblock.Unblocker,width=6,bg='#5DADE2',activebackground = 'sky blue')

unblock.place(x=230,y=150)

def runapp2(path):

root.mainloop()

if \_\_name\_\_ == '\_\_main\_\_':

path = pathfinder.hostfilePath()

runapp2(path)

**website\_unblocker.py**

import runapp2 as ra2

import hostfilePath as hfp

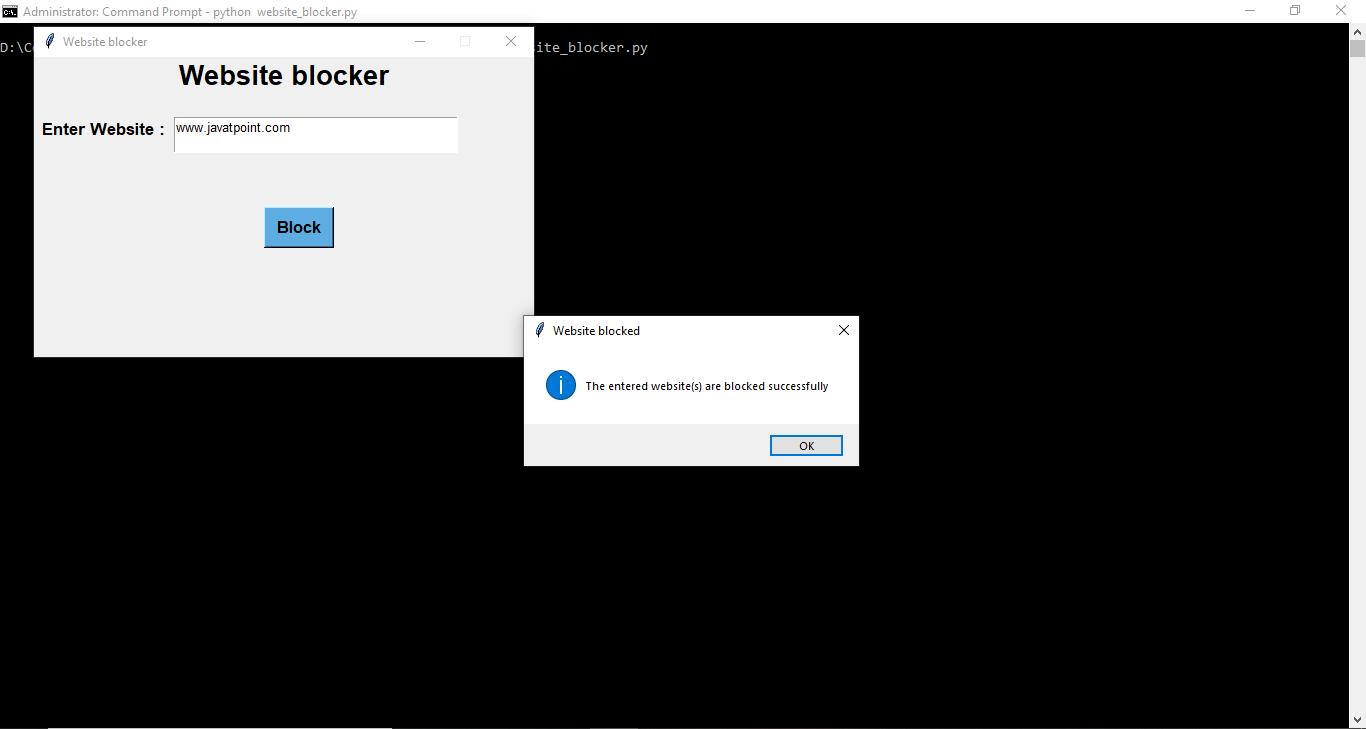
if \_\_name\_\_ == ‘\_\_main\_\_’:

path = hfp.hostfilePath()

ra2.runapp(path)

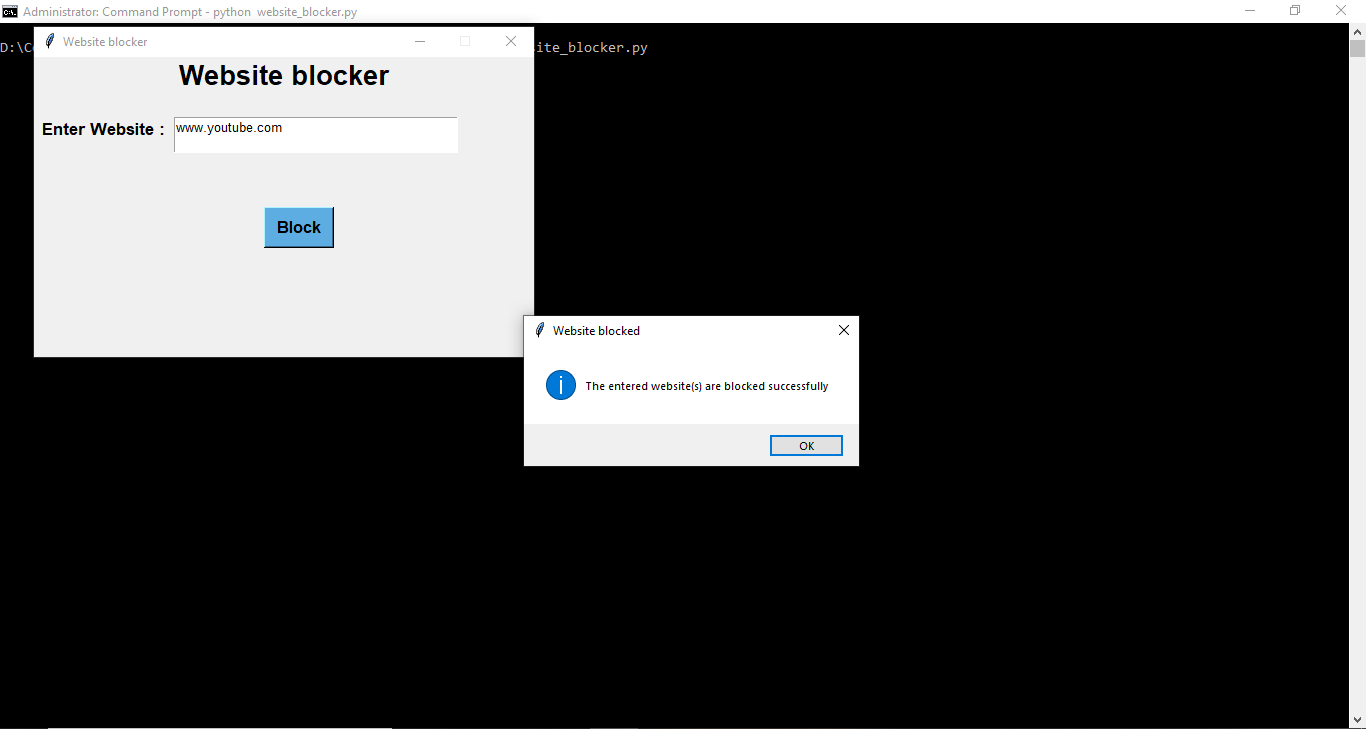
**9.2 Appendix 2 : Snapshots of Output**

**9.2.1 Website Blocker Output:**



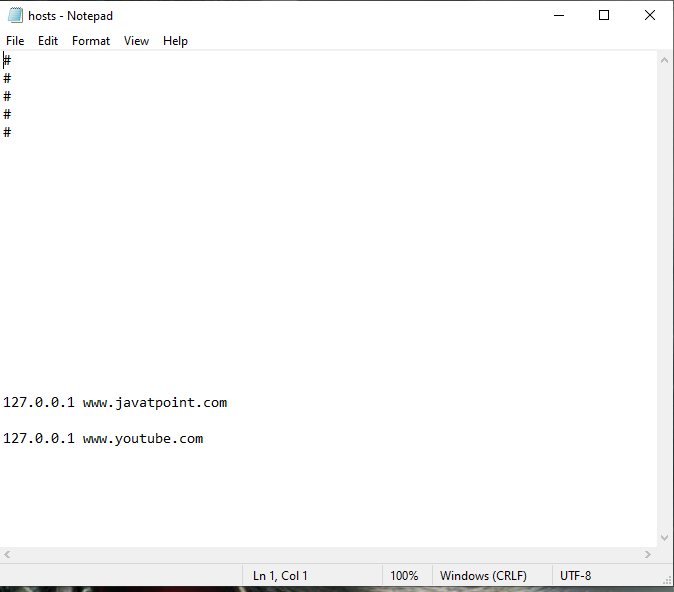
***Fig 9.2.1.1 : Blocking*** [***www.javatpoint.com***](http://www.javatpoint.com) ***website***

* The website [www.javatpoint.com](http://www.javatpoint.com) is blocked.



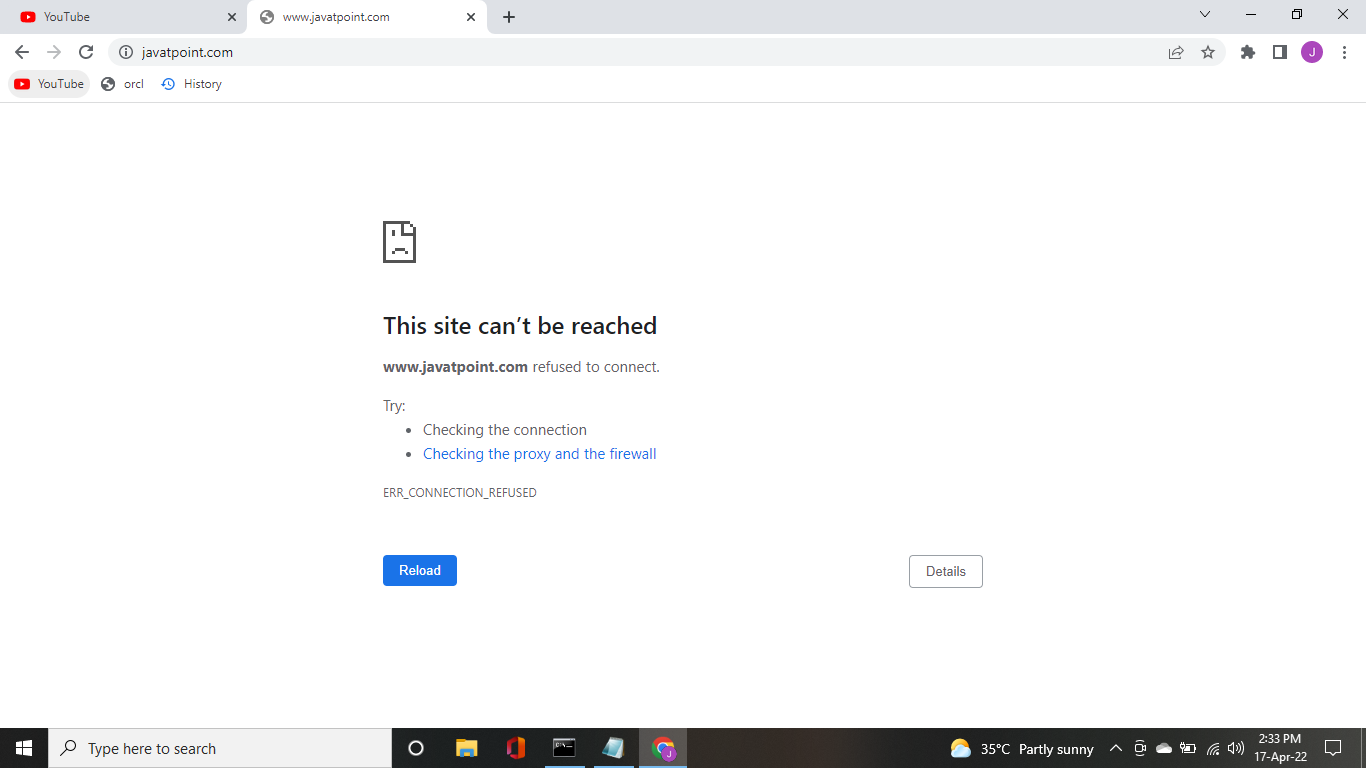
***Fig 9.2.1.2 : Blocking*** [***www.youtube.com***](http://www.youtube.com) ***website***

* The website [www.youtube.com](http://www.youtube.com) is blocked.



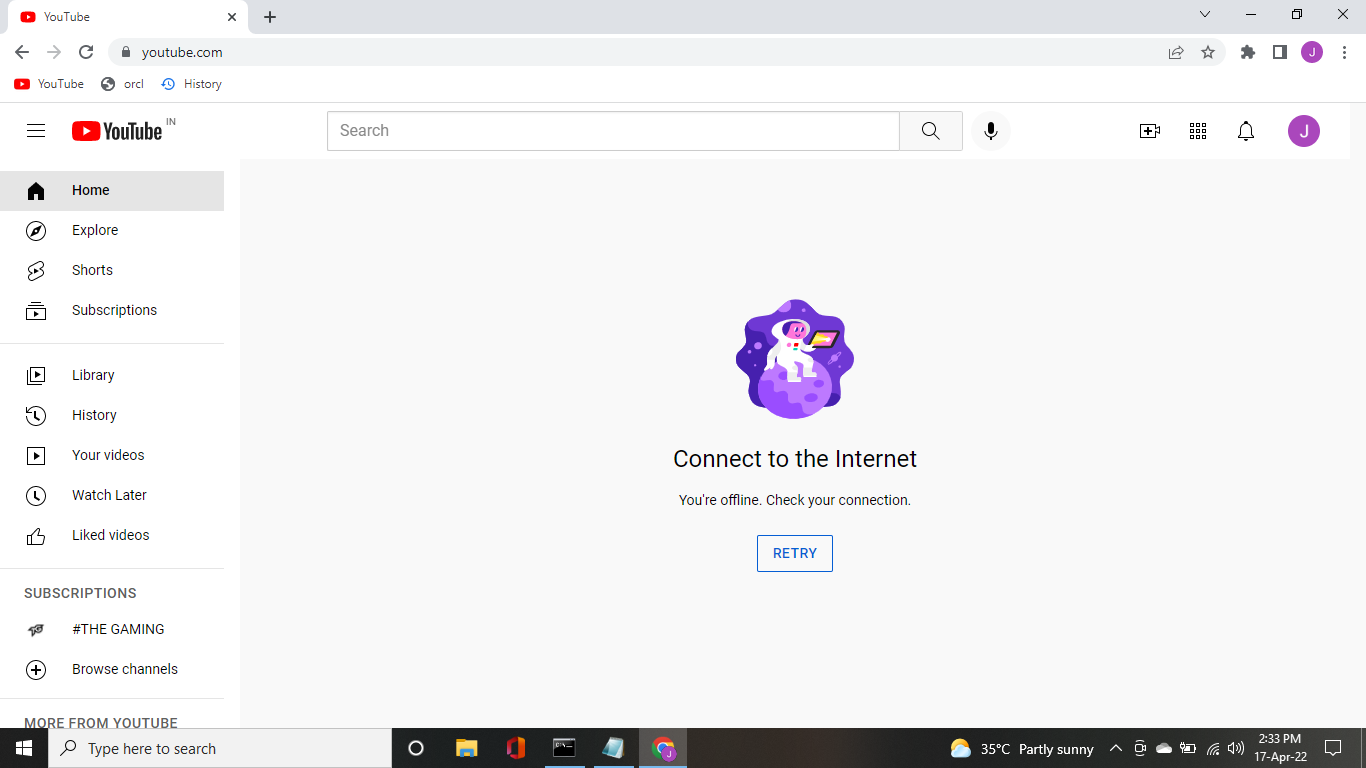
***Fig 9.2.1.3 : Host file after blocking the given website names***

* The websites [www.javatpoint.com](http://www.javatpoint.com) and [www.youtube.com](http://www.youtube.com) are present in the hosts file.
* This shows that the websites present in the hosts file are blocked in the system and is inaccessible



***Fig 9.2.1.4: Result of blocking*** [***www.javatpoint.com***](http://www.javatpoint.com) ***website***

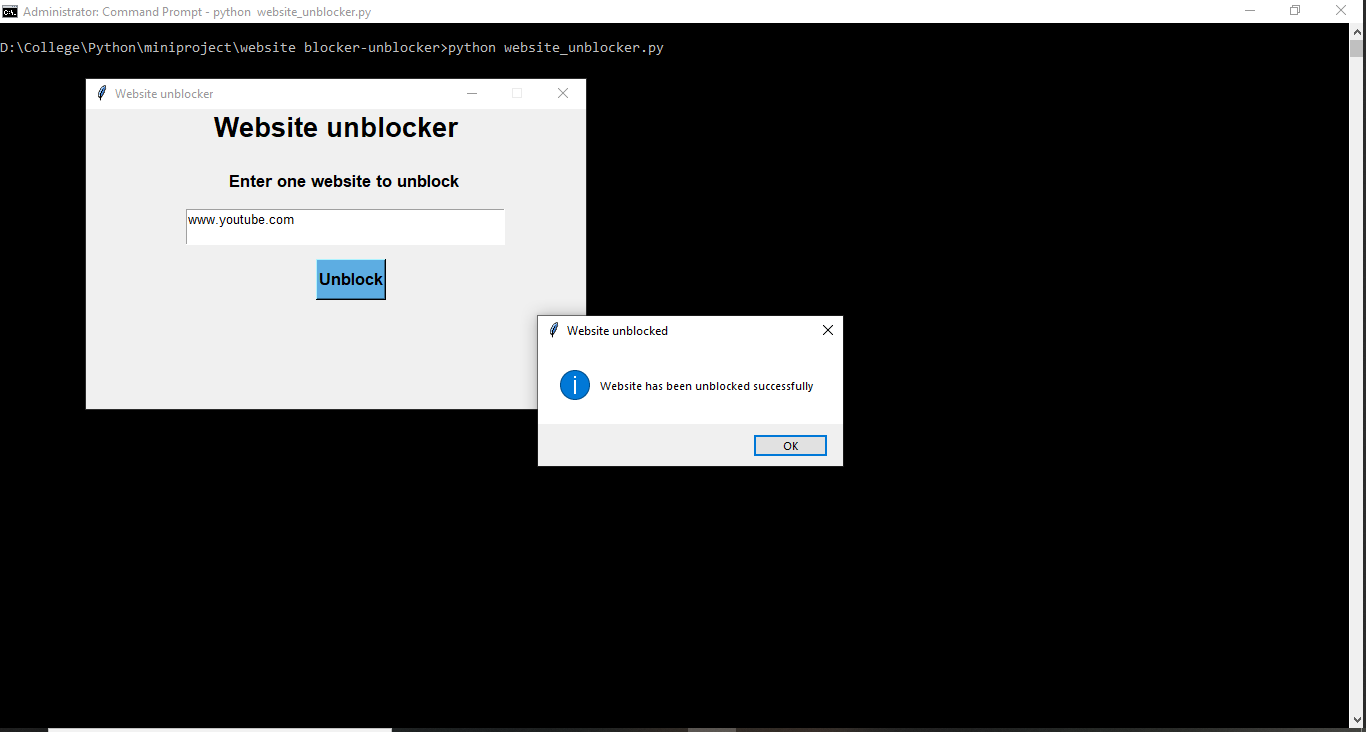
* The website [www.javatpoint.com](http://www.javatpoint.com) is blocked and inaccessible.



***Fig 9.2.1.5 : Result of blocking*** [***www.youtube.com***](http://www.youtube.com) ***website***

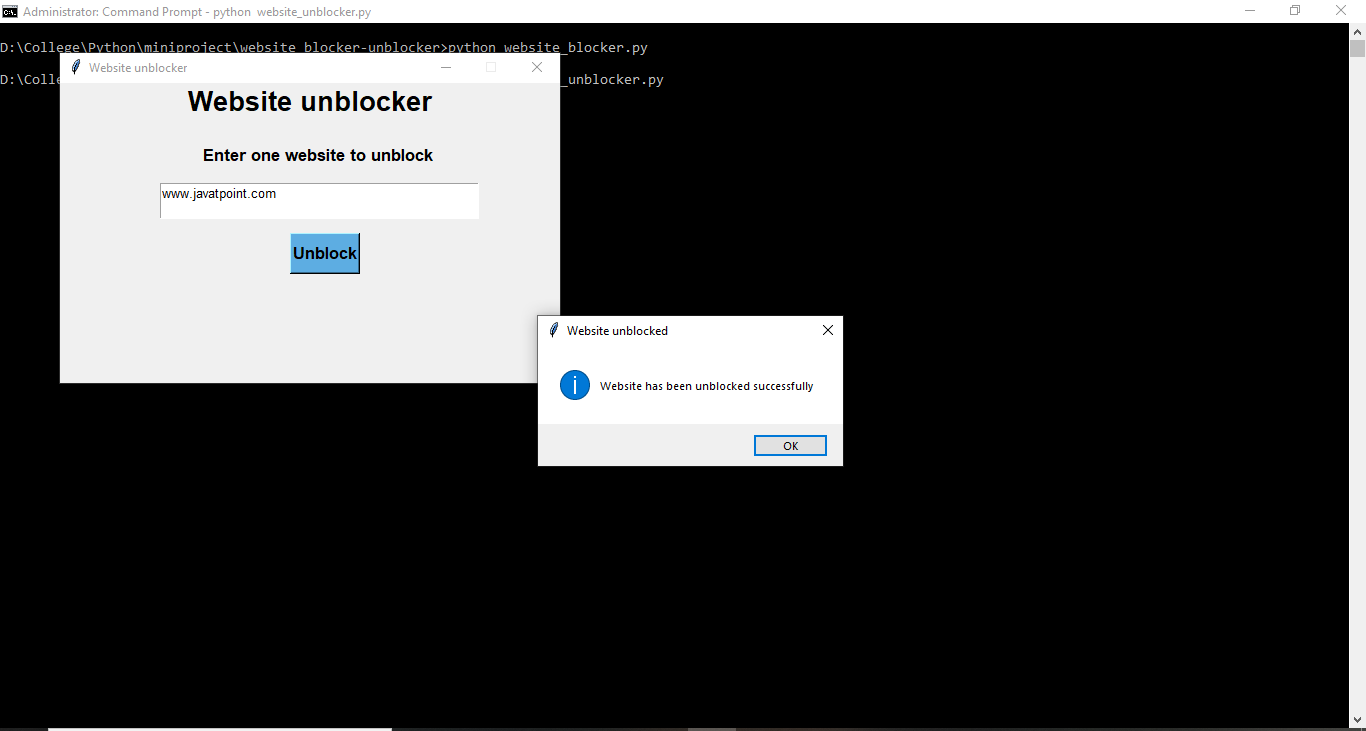
* The website [www.youtube.com](http://www.youtube.com) is blocked and is inaccessible.

**9.2.2 Website Unblocker Output:**



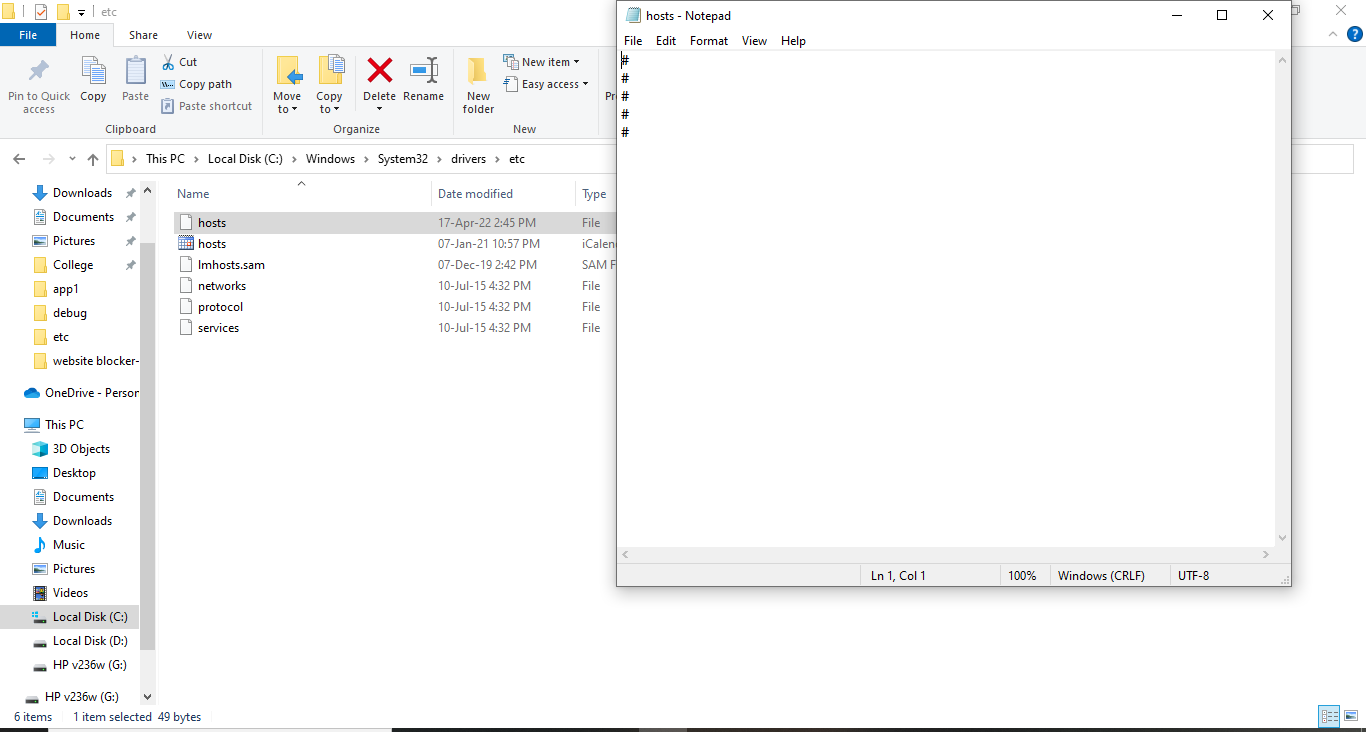
***Fig 9.2.2.1 : Unblocking*** [***www.youtube.com***](http://www.youtube.com) ***website***

* The website [www.youtube.com](http://www.youtube.com) is unblocked if the website name is present in the hosts file



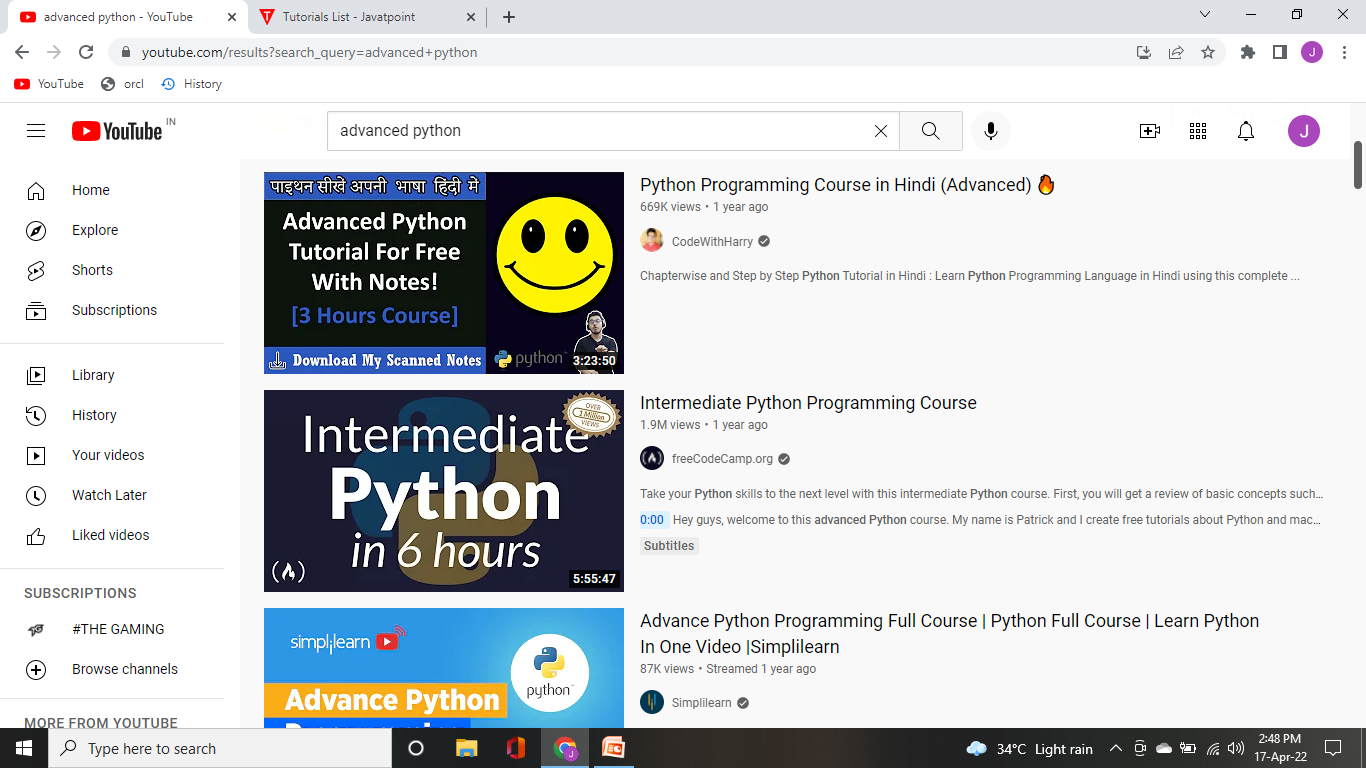
***Fig 9.2.2.2 : Unblocking*** [***www.javatpoint.com***](http://www.javatpoint.com) ***website***

* The website [www.javatpoint.com](http://www.javatpoint.com) is unblocked if the website name is present in the hosts file



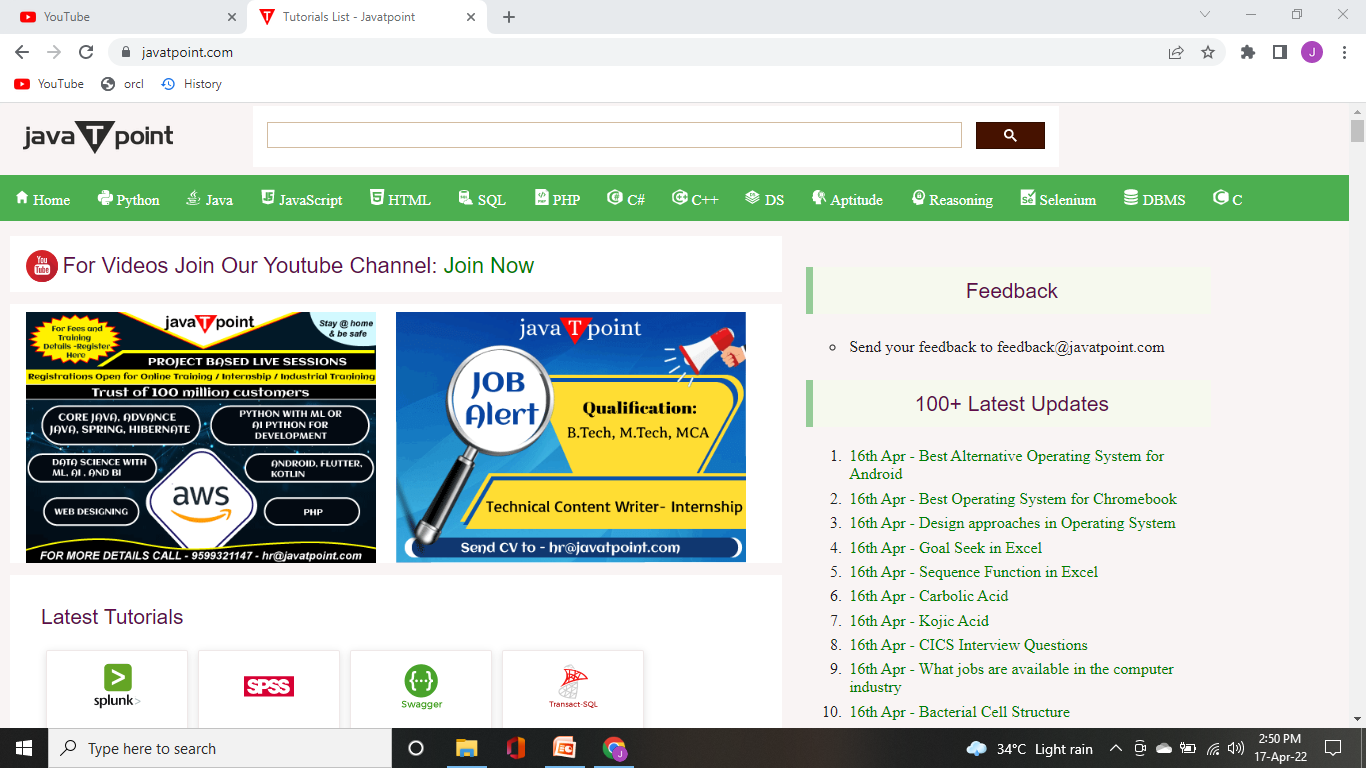
***Fig 9.2.2.3 : Host file after unblocking the given website names***

* Both the website names are absent in the hosts file
* This indicates that the websites are unblocked and can be accessed.



***Fig 9.2.2.4 : Result of unblocking*** [***www.youtube.com***](http://www.youtube.com) ***website***

* The website [www.youtube.com](http://www.youtube.com) is unblocked and is accessible



***Fig 9.2.2.5 : Result of unblocking*** [***www.javatpoint.com***](http://www.javatpoint.com) ***website***

* The website [www.javatpoint.com](http://www.javatpoint.com) is unblocked and is accessible.

**CHAPTER 10**

**REFERENCES**

[1] Philip B.stark , “The Effectiveness of Internet Content Filters” ,January 2008

[2] Richard barnes , Alissa cooper , Olaf kolkman , Dave thaler and Erik

nordmark , “RFC 7754 – Technical Consideration for Internet Service Blocking and Filtering” , March 2016

[3] Polpinji, J., Sibunrang, C., Paungpronpitag, S., Chamchong, R. and Chotthanom, A (2008). A web pornography patrol system by content-based analysis: In particular text and image. In Proceedings of IEEE International Conf. Systems, Man and Cybernetics. Singapore: IEEE Xplore, p. 500-505

[4] Anuradha Mathrani , Massoud Alipour , “Website Blocking Across Ten Countries: A Snapshot ” , January 2010

[5] George Merzdovnik , Markus Huber , Damjan Buhov , Nick Nikiforakis , Sebastian Neuner , Martin Schmiedecker , Edgar Weippl , “Block Me If You Can: A Large-Scale Study of Tracker-Blocking Tools” , IEEE European Symposium on Security and Privacy , April 2017

[6] D. New and N. Borenstein, “KidCode: Naming Conventions for Protecting Children on the World Wide Web and Elsewhere on the Internet Without Censorship” , June 1995 , [online] Available : Google Scholar

[7] Chen Ding , Chi-Hung Chi , Jing Deng , Chun-Lei Dong , “Centralized content-based Web filtering and blocking: how far can it go ?”, Singapore: IEEE Xplore, 06 August 2002

[8] “Manual: Obtaining Data from Query Results”, PEAR-PHP Extension and Application Repository. The PHP Group. Web, May 2016 , [online] Available : <http://pear.php.net/manuallen/package.database.mdb2.intro-fetch.php>.

[9] Maurice Reed , Zizhong John Wang , “Securing Webpages with EasyIPBlocker” , IEEE Xplore , 20 March 2017

[10] Elisa Bertino et al., “A General Framework for Web Content Filtering” , World Wide Web, vol. 13, no. 3, pp. 215-249, September 2010

[11] Abbas M. Al-Bakry , Talib T. Al-Fatlawii , “Intelligent web content filtering system using MAS” , IEEE Xplore , 17 March 2014

[12] Ali Ahmad et al., “Intelligent classification of webpages using contextual and visual features” , Applied Soft Computing Elsevier, vol. 11, no. 2, pp. 1638-1647, March 2017

[13] Ertugrul Akbag , “Next Generation Filtering: Offline Filtering Enhanced Proxy Architecture For Web Content Filtering” , 23rd International Symposium on Computer and Information Sciences(ISCIS’08) IEEE, pp. 1-4 , 2008

[14] Mohammed Hammami et al., “WebGuard: Web Based Adult Content Detection and Filtering System”, Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI’03), pp. 574-578 , 2003

[15] Peng Nie , Ruixuan Wang , Jialin Dong , Yuchen Zheng , “Research on interactive technology of Webpage block localization based on JavaScript”, IEEE Xplore , 07 September 2021