

# **DENGUARD ANTI-VIRUS**

## **A PROJECT REPORT**

Submitted by

**D.S.J. NANDINI (20MEI10049)**

**SHRADDHA PANDEY (20MEI10029)**

**CHANCHAL SHARMA (20MEI10031)**

**MANSI BAKHSI (20MEI10015)**

*in partial fulfillment for the award of the degree  
of*

**INTEGRATED MASTER OF TECHNOLOGY**

*in*

**CYBER SECURITY AND DIGITAL FORENSICS**



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRIKALAN, SEHORE**

**MADHYA PRADESH - 466114**

APRIL, 2022

**VIT BHOPAL UNIVERSITY, KOTRIKALAN, SEHORE  
MADHYA PRADESH – 466114**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **“DENGUARD ANTI-VIRUS”** is the bonafide work of **“CHANCHAL SHARMA (20MEI10031), D.S.J. NANDINI (20MEI10049), SHRADDHA PANDEY (20MEI10029), MANSHI BAKHSI (20MEI10015)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROGRAM CHAIR**

Dr. H. Azath

School of Computer Science and Engineering  
VIT BHOPAL UNIVERSITY

**PROJECT GUIDE**

**Supervisor :** Dr Thangavel M

**Reviewer:** Dr Asdaque Hussain

School of Computer Science and Engineering  
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on 23 April, 2022

## ACKNOWLEDGEMENT

First and foremost, we would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

We wish to express our heartfelt gratitude to Dr. H. Azath Head of the Department, School of Computer Science for much of his valuable support encouragement in carrying out this work.

We would like to thank our internal guides Dr. Asdaque Hussain and Dr. Thangavel M, for continually guiding and actively participating in our project, giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Computer Science, who extended directly or indirectly all support.

Last, but not least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

## LIST OF ABBREVIATIONS

S. No.	ABBREVIATIONS	
1.	DOS	Denial of service attack

<b>2.</b>	<b>DDOs</b>	<b>Distributed denial of service</b>
<b>3.</b>	<b>GUI</b>	<b>Graphical User Interface</b>
<b>3.</b>	<b>SYN</b>	<b>Synchronize</b>
<b>4.</b>	<b>AV</b>	<b>Antivirus</b>
<b>5</b>	<b>VSC</b>	<b>Visual Studio Code</b>
<b>6.</b>	<b>SHA256</b>	<b>Secure Hash Algorithm</b>
<b>8.</b>	<b>OS</b>	<b>Operating System</b>

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>3.3.4</b>	<b>LOOK AND FEEL REQUIREMENTS</b>	<b>16</b>
<b>4.3</b>	<b>SOFTWARE ARCHITECTURAL DESIGN</b>	<b>19</b>
<b>5.2</b>	<b>TECHNICAL CODING AND CODE SOLUTIONS</b>	<b>20</b>
<b>5.2</b>	<b>TECHNICAL CODING AND CODE SOLUTIONS</b>	<b>21</b>
<b>5.2</b>	<b>TECHNICAL CODING AND CODE SOLUTIONS</b>	<b>22</b>
<b>5.2</b>	<b>TECHNICAL CODING AND CODE SOLUTIONS</b>	<b>22</b>
<b>5.2</b>	<b>TECHNICAL CODING AND CODE SOLUTIONS</b>	<b>23</b>
<b>5.3</b>	<b>PROTOTYPE SUBMISSION</b>	<b>24</b>
<b>5.3</b>	<b>PROTOTYPE SUBMISSION</b>	<b>24</b>
<b>5.4</b>	<b>TEST AND VALIDATION</b>	<b>25</b>

## **ABSTRACT**

Antivirus software also known as anti-malware. Antivirus is a kind of software used to prevent, scan, detect and delete viruses from a computer. And prevent from multiple attacks like DDOS, DOS Comprehensive virus protection programs help protect your files and hardware from malware such as worms, Trojan horses and spyware, and may also offer additional protection such as customizable firewalls and website blocking. Antivirus programs and computer protection software are designed to evaluate data such as web pages, files, software and applications to help find and eradicate malware as quickly as possible.

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	List of Abbreviations	iii
	List of Figures	iv
	Abstract	v
		vi
1	<b>CHAPTER-1:</b> <b>PROJECT DESCRIPTION AND OUTLINE</b> 1.1 Introduction 1.2 Motivation for the work 1.3 [About Introduction to the project including techniques] 1.4 Problem Statement 1.5 Objective of the work 1.6 Organization of the project 1.7 Summary	1   . . .
2	<b>CHAPTER-2:</b> <b>RELATED WORK INVESTIGATION</b> 2.1 Introduction 2.2 Core area of the project 2.3 Existing Approaches/Methods 2.3.1 Approach/Method -1 2.3.2 Approach/Method -2 2.3.3 Approach/Method -3 2.4 Pros and cons of the existing Approaches/Methods 2.5 Issues/observations from investigation	
		6   Page

	2.6 Summary	
3	<p style="text-align: center;"><b>CHAPTER-3:</b></p> <p style="text-align: center;"><b>REQUIREMENT ARTIFACTS</b></p> <p>3.1 Introduction</p> <p>3.2 Hardware and Software requirements</p> <p>3.3 Specific Project requirements</p> <p>3.3.1 Data requirement</p> <p>3.3.2 Functions requirement</p> <p>3.3.3 Performance and security requirement</p> <p>3.3.4 Look and Feel Requirements</p> <p>3.4 Summary</p>	
4	<p style="text-align: center;"><b>CHAPTER-4:</b></p> <p style="text-align: center;"><b>DESIGN METHODOLOGY AND ITS NOVELTY</b></p> <p>4.1 Methodology and goal</p> <p>4.2 Functional modules design and analysis</p> <p>4.3 Software Architectural designs</p> <p>4.4 Subsystem services</p> <p>4.5 User Interface design</p> <p>4.6 Summary</p>	
5	<p style="text-align: center;"><b>CHAPTER-5:</b></p> <p style="text-align: center;"><b>TECHNICAL IMPLEMENTATION &amp; ANALYSIS</b></p> <p>5.1 Outline</p> <p>5.2 Technical coding and code solutions</p> <p>5.3 Prototype submission</p> <p>5.4 Test and validation</p> <p>5.5 Performance Analysis (Graphs/Charts)</p> <p>5.6 Summary</p>	
6	<p style="text-align: center;"><b>CHAPTER-6:</b></p>	

	<p><b>PROJECT OUTCOME AND APPLICABILITY</b></p> <p>6.1 Outline</p> <p>6.2 key implementation outlines of the System</p> <p>6.3 Significant project outcomes</p> <p>6.4 Project applicability on Real-world applications</p> <p>6.5 Inference</p>	
7	<p><b>CHAPTER-7:</b></p> <p><b>CONCLUSIONS AND RECOMMENDATION</b></p> <p>7.1 Outline</p> <p>7.2 Limitation/Constraints of the System</p> <p>7.3 Future Enhancements</p> <p>7.4 Inference</p>	
	<p>Appendix A</p> <p>References</p> <p><i>Note: List of References should be written as per IEEE/Springer reference format. (Specimen attached)</i></p>	



## **CHAPTER – 1**

### **PROJECT DESCRIPTION AND OUTLINE**

#### **1.1 INTRODUCTION**

Antivirus software, or **antivirus software** (abbreviated to **AV software**), also known as **anti-malware**, is a computer program used to prevent, detect, and remove malware. Antivirus software was originally developed to detect and remove computer viruses, hence the name. However, with the proliferation of other malware, antivirus software started to protect from other computer threats. In particular, modern antivirus software can protect users from malicious browser helper objects (BHOs), browser hijackers, ransomware, keyloggers, backdoors, rootkits, trojan horses, worms, malicious LSPs, dialers, fraud tools, adware, and spyware. Some products also include protection from other computer threats, such as infected and malicious URLs, spam, scam and phishing attacks, online identity (privacy), online banking attacks, social Engineering techniques, advanced persistent threat (APT), and botnet DDoS attacks.

#### **1.2 MOTIVATION FOR THE WORK**

The motivation behind developing this antivirus tool according to its use in various organizations to scan their system in order to keep their system safe, as well as, it can be used for finding the malicious virus worms and trojan in their host so that they can protect them. The main goal of using the antivirus software is to scan the system or files to know if there are any loopholes in the system or files or folder. If there are any loopholes in their system so the antivirus can detect them, and protect their system against any malicious attack.

#### **1.3 INTRODUCTION TO THE PROJECT INCLUDING TECHNIQUES**

Antivirus software begins operating by checking computer programs and files against a database of known types of malwares. Since new viruses are constantly created and distributed by hackers, it will also scan computers for the possibility of new or unknown type of malware

threats. Typically, most programs will use three different detection devices: specific detection, which identifies known malware; generic detection, which looks for known parts or types of malware or patterns that are related by a common codebase. When the program finds a file that contains a virus, it will usually quarantine it and/or mark it for deletion, making it inaccessible and removing the risk to your device.

#### **1.4 PROBLEM STATEMENT**

As we know hackers, by using different kind of tools and techniques they try to enter different type of malware in the system so that they can get in to the system or they can gain unauthorized access. So, first step of any attacker is that they first trying to enter the malicious code in to the system. And once they successful in their work then they can perform their malicious activity. how will this that before entering any malware into the system we can detect and remove from the system. Obviously, there is lot of tools and algorithm is available that scanning the all files which is coming from the Internet no, matter it is malicious or not. but they are quite slow. so, to avoid this problem we have developed an antivirus tools or algorithm that is quite fast comparing to others algorithm.

#### **1.5 OBJECTIVE OF THE WORK**

Our objective is to make an antivirus that can scan all files, which are coming from the network or which we are downloading from the network and can also scan web application very efficient manner and less amount of time and get you know is there any files which are containing any malware. If there are any files which containing any malware so we need to detect and remove those files from the system also secure the organization or system to any type of futuristic attack. There are many tools in this field but each tool is having some cons along with pros. Our aim is to develop a flawless tool that contains solution for most of the cons in existing tools. The main objective is that it could scan the any files or folder in very short period so it can secure the time of the user. And in less time the user can find out the malicious files into their system and can remove them.

#### **1.6 SUMMARY**

The basic idea is that scan any files or folder in very fast manner and save the time of the user. Our objective is to develop the field of security. Our tool can finish scanning 15 files in less than

1 second! Anti-Virus proves to be useful in many cases, an authorized user can use this tool to see if there are any files in their system which are containing any kind of malware (virus, worms, trojan) so, that it can detect and remove from the system.

## **CHAPTER – 2**

### **RELATED WORK INVESTIGATION**

#### **2.1 INTRODUCTION**

At the investigation phase of our project, we planned to analyze various antivirus tools available online and offline also. In this process we have tested below mentioned tools and software individually

1. Bitdefender Internet Security.
2. Norton Security.
3. McAfee.
4. Comodo Internet Security.
5. Malwarebytes.
6. Adware
7. Kaspersky
8. VirusTotal

#### **2.2 CORE AREA OF THE PROJECT**

Core area of the project is Malware scanning, and our objective is to develop a malware scanning tool with extra added features like –graphical user interface and efficiency (took less time). To enable the quick and effective scanning of the file/folder. Our aim is to develop a flawless algorithm for our tool which contains solutions to the most problems in existing tools.

#### **2.3 EXISTING APPROACHES/METHODS**

As we are developing a scanning tool, so the existing tools which use different methods like when we scan any host through VirusTotal, VirusTotal inspects items with over 70 antivirus scanners and URL/domain block listing services, in addition to a myriad of tools to extract signals from the studied content. Any user can select a file from their computer using their

browser and send it to VirusTotal. VirusTotal offers a number of file submission methods, including the primary public web interface, desktop uploaders, browser extensions and a programmatic API. The web interface has the highest scanning priority among the publicly available submission methods. Submissions may be scripted in any programming language using the HTTP-based public API.

## **2.4 PROS AND CONS OF THE EXISTING APPROACHES/METHODS**

### **1. Bitdefender Internet Security.**

Pros: -

- Fast and Secure VPN
- Best Firewall security
- Hardly interfere with other software
- Quick installation
- Value for Money
- Top-class security rating
- Malware detection
- Secure system mode for every situation

Cons: -

- High price rate
- Weak offline security
- Continually ask to upgrade your plans
- Not Unlimited access to VPN
- Take more time than usual to scan

### **2. Norton Security.**

Pros: -

- Industry-leading protection and performance.
- Powered by one of the world's largest 24x7 threat monitoring networks.
- Patented protection against viruses, scams, phishing attempts, zero-day exploits and more.
- Automatic "silent" updates ensure you have the latest protection.

Cons: -

- Demanding on Resources
- Costly
- Slow First Scan
- Technical Support Issues

- Uninstalling Problems.

### 3.McAfee.

Pros: -

- Unbeatable malware detection and removal
- Proactive protection from zero-day attacks
- Leverage McAfee Global Threat Intelligence™ (McAfee GTI™)
- Manage your security
- Manage quarantined files

Cons: -

- System slowdown
- Full system scan takes more than an hour to complete
- Detection limits
- Ties up system memory
- Updates and uninstallation
- Costly

### 4.Adware

Pros: -

- Simple and intuitive interface
- Affordable prices
- Free baseline real-time protection is available
- Doesn't overload the system
- Compatible with other antivirus programs

Cons: -

- Mixed lab test scores
- Doesn't detect malware well based on its behavior
- Limited configuration
- Isn't compatible with macOS
- Doesn't offer protection for mobile devices

### 5.MalwareByte

Pros: -

- Free version
- Good virus detection rates
- Intuitive user interface

- Minimal CPU usage

Cons: -

- The free version has no real-time protection
- No live chat supports
- No firewall features

## **2.5 ISSUES/OBSERVATIONS FROM INVESTIGATION**

As we analyzed different types of tools, that too large in number, common findings were -

- Most of the tools required compatible devices or software.
- They were not attack resilient but vulnerable.
- Very large number of tools were available online, but to use most of them we have to first download them then we were able to use them.
- And also, they were very slow

## **2.6 SUMMARY**

Very large number of tools were available online, but to use most of them we have to first download them then we were able to use them. And also, they were quite slow. We are dedicated to remove these flaws from our tool and have made very efficient and fast.

# **CHAPTER – 3**

## **REQUIREMENT ARTIFACTS**

### **3.1 INTRODUCTION**

As our idea is to develop a malware scanning tool with GUI which is user-friendly which provides the graphics-based interface that uses icons, menus and a mouse (to click on the icon or pull down

the menus) to manage interaction with the system, for that we need some hardware and software. By using our tool, the users would be able to access/download this tool and they came to know that how our (tool) algorithm works.

## **3.2 HARDWARE AND SOFTWARE REQUIREMENTS**

We need to have -

- \* Laptop/Desktop/Tablet
- \* Any python Id
- \* It does not require any Internet Connection
- \* Windows 7 and above version
- \* Windows 7 & Windows 8:
- \* 32-bit: 1GB RAM minimum, 2GB recommended
- \* 64-bit: 2GB RAM minimum, 4GB recommended

## **3.3 SPECIFIC PROJECT REQUIREMENTS**

### **3.3.2 FUNCTIONS REQUIREMENT**

As we have used visual studio framework in our project, it is very popular IDE for writing python. It combines the best developer experience with an obsessive focus on end-user performance. Our platform enables frontend teams to do their best work. Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

### **3.3.3 PERFORMANCE AND SECURITY REQUIREMENT**

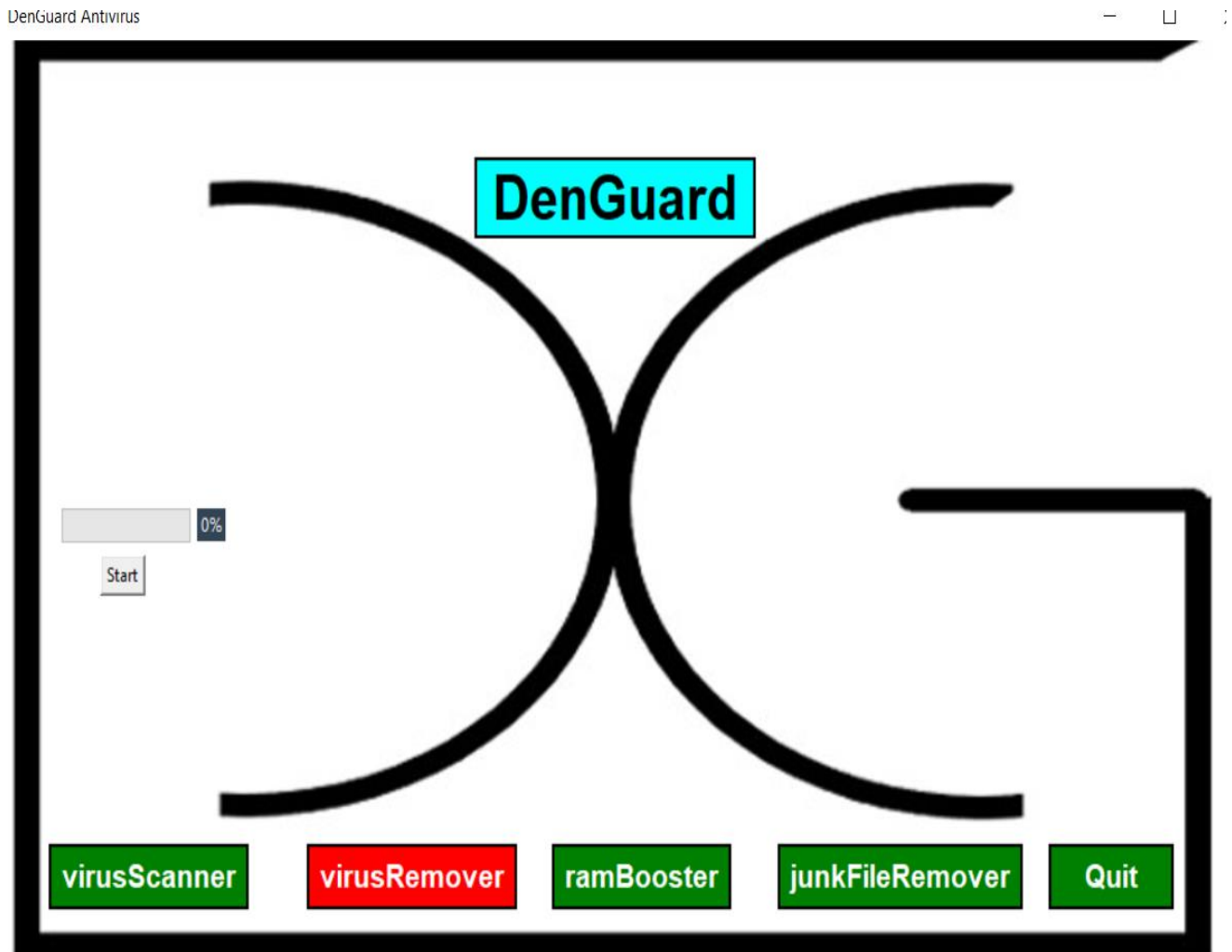
It scans any folder/file within minimum amount of time. It can scan 100 files within a minute. Before downloading any software files from the internet first we must scan the files carefully, and check for the malware. So that we can avoid the malicious activity of malious users. Otherwise, if we are not doing this then the malicious user

can get unauthorized access of our system and he/she can perform their activity whatever he/she want. So to avoid this kind of problem we required this kind of tool.

### 3.3.4 LOOK AND FEEL REQUIREMENTS

We have designed our Tool in such a way that user find it attractive.

Here we have used TkInter module to make it look better when display the output.



### 3.4 SUMMARY



By using our tool (Algorithm), the users would be able to access/download this tool. And Then They will come to know that how it works and what is uniqueness of our tool.

## **CHAPTER – 4**

### **DESIGN METHODOLOGY AND ITS NOVELTY**

#### **4.1 METHODOLOGY AND GOAL**

We are using Heuristic-based detection technique for scanning the malware. Heuristic-based detection is considered the most common form of virus detection that uses an algorithm to differentiate the signature of known viruses against a potential threat. It can unearth viruses that have not yet been discovered, as well as known viruses that have been modified or disguised, and released into the wild again.

Our goal is to develop an algorithm in the field of malware detection with new features like GUI (The Graphical User Interface displays objects that convey information, and represent actions that can be taken by the user). Flawless algorithm for the new tool that contains solution for most of the cons in existing tools. A tool which is compatible with window and which have more efficiency than existing one.

#### **4.2 FUNCTIONAL MODULES DESIGN AND ANALYSIS**

##### **OS**

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The `*os*` and `*os.path*` modules include many functions to interact with the file system.

##### **SYS**

The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter. It lets us access system-specific parameters and functions. `import sys`- First, we have to import the sys module in our program before running any functions.

### **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. To creating a GUI using TkInter is easy, for that we need to follow these steps-

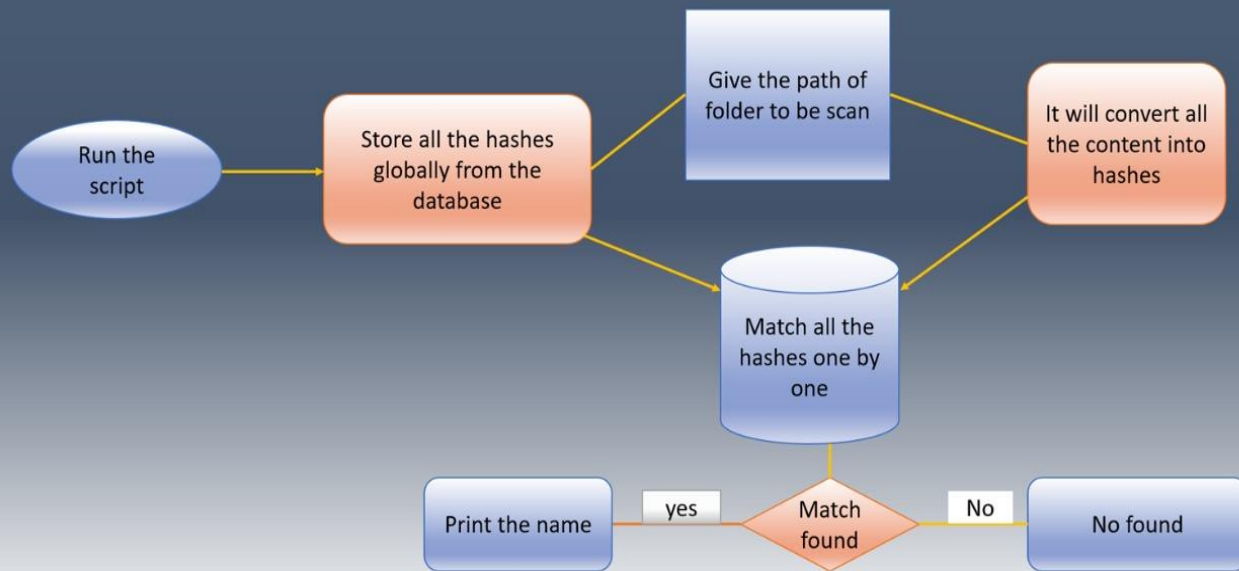
- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

### **Hashlib**

The Python hashlib module is an interface for hashing messages easily. This contains numerous methods which will handle hashing any raw message in an encrypted format. The core purpose of this module is to use a hash function on a string, and encrypt it so that it is very difficult to decrypt it. Using Python hashlib to Implement SHA256. Python has a built-in library, hashlib, that is designed to provide a common interface to different secure hashing algorithms. The module provides constructor methods for each type of hash. For example, the `sha256()` constructor is used to create a SHA256 hash.

## **4.3 SOFTWARE ARCHITECTURAL DESIGN**

## Overall system architecture diagram



### 4.4 SUBSYSTEM SERVICES

We can scan any files/folder within a minute and user can know which file is containing the malware and which are not. It is very fast can scan 100 files within 5 second. And save the time of the user.

It is very simple to use. If a person which is not having knowledge about the antivirus can also scan the files/folder. Means it is age friendly. It also removes the junk file from the system. It also kills all the process which is running in the background of the system. It first detects the malware files then if it finds any malware files then it removes from the system then exit from the system.

### 4.5 SUMMARY

Flawless tool for the malware scanning that contains solution for most of the cons in existing tools. A tool which is compatible with windows and which doesn't require any type internet connection. The purpose of malware scanning (Antivirus) is to acquire information from the system to which the files are attached. antivirus software helps to protect the file system against unwanted programs.

## CHAPTER – 5

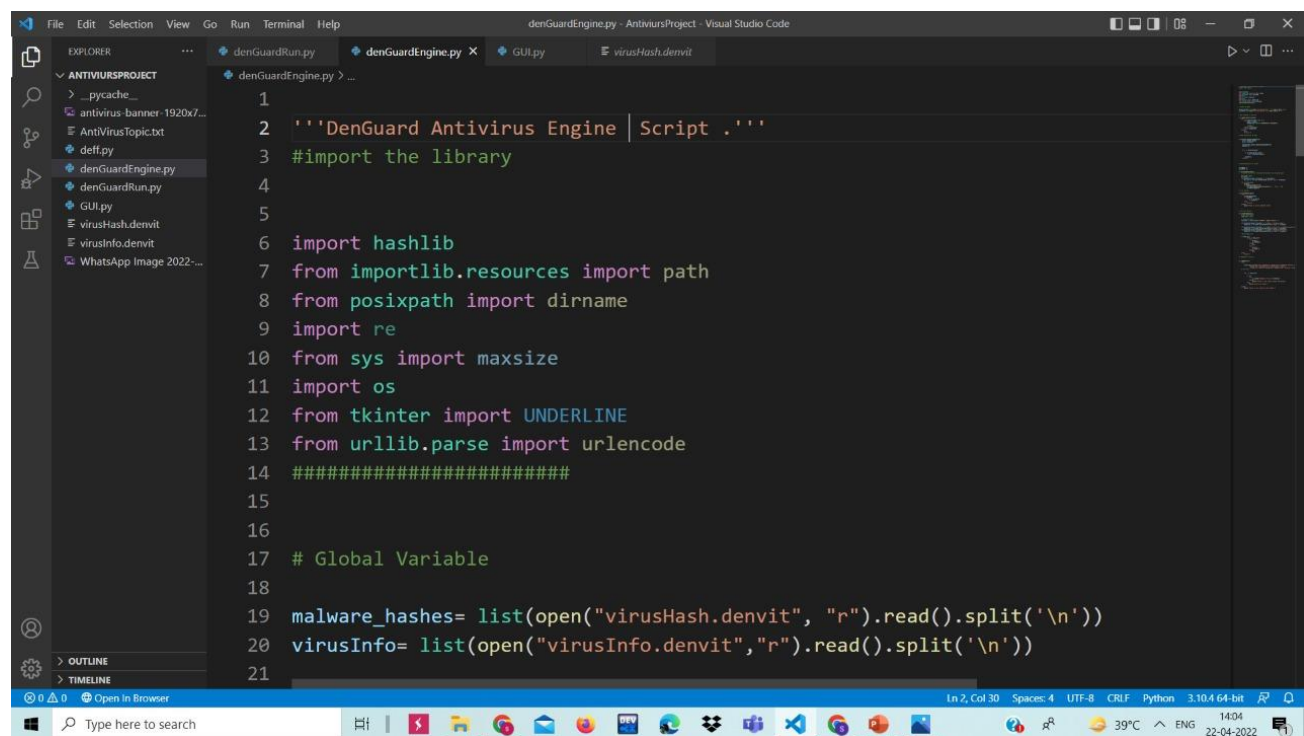
### TECHNICAL IMPLEMENTATION & ANALYSIS

#### 5.1 OUTLINE

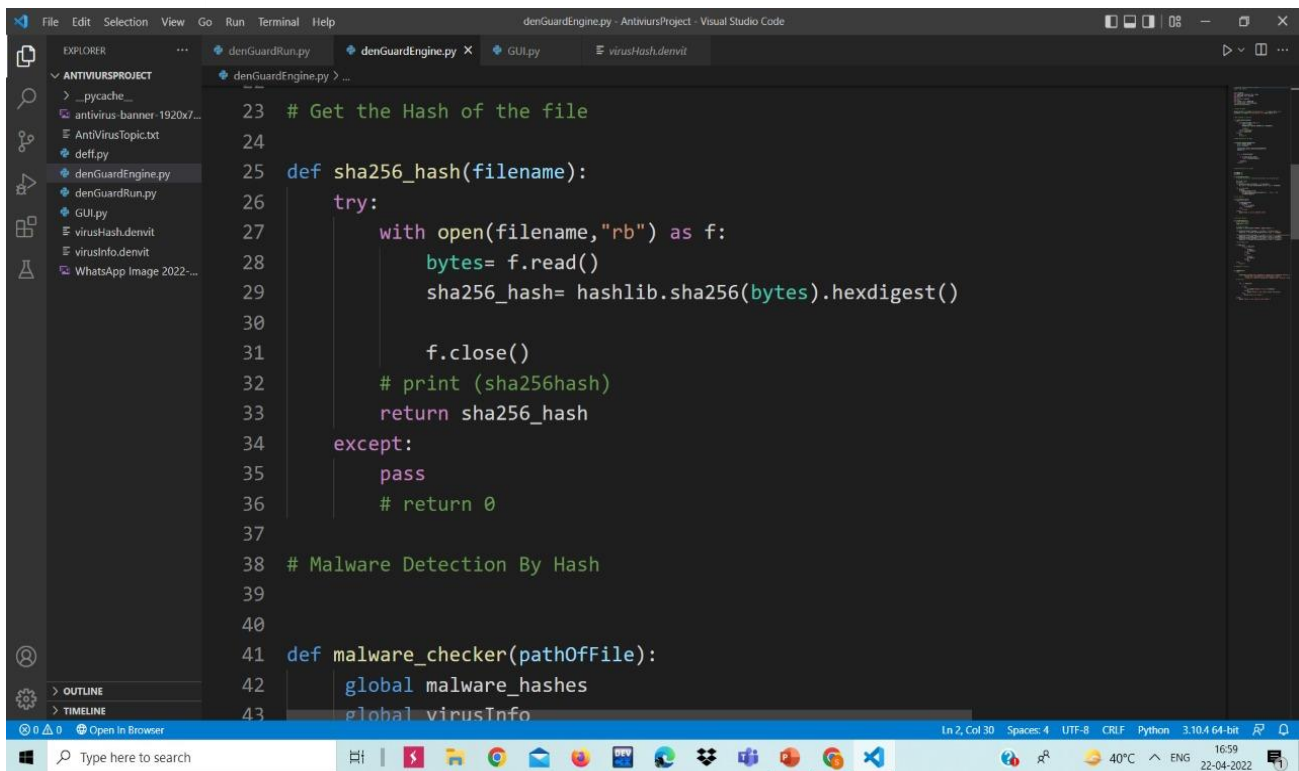
Our aim is to develop a flawless algorithm/tool which is:

- User-friendly
- Invulnerable or resilient
- Compatible with windows OS
- Comfortable to view
- Age friendly
- Simple to use

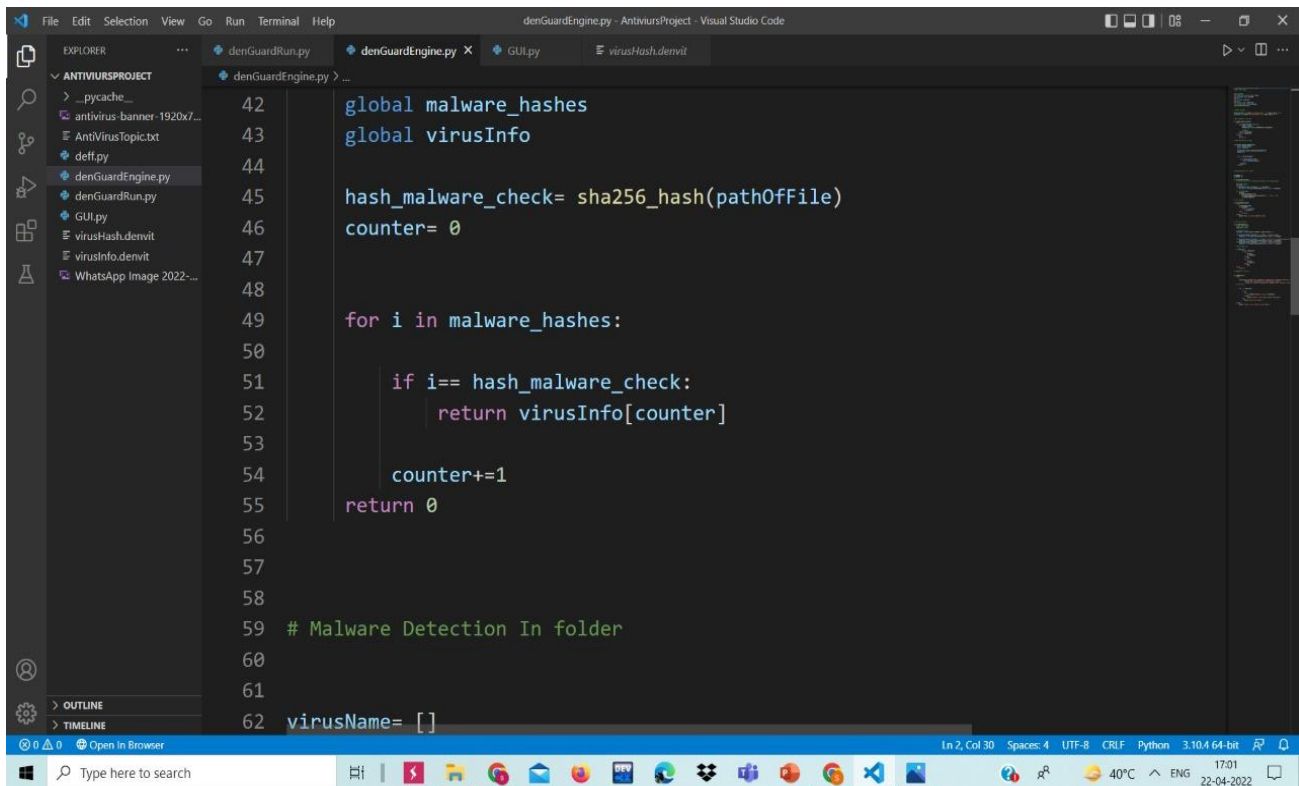
#### 5.2 TECHNICAL CODING AND CODE SOLUTIONS



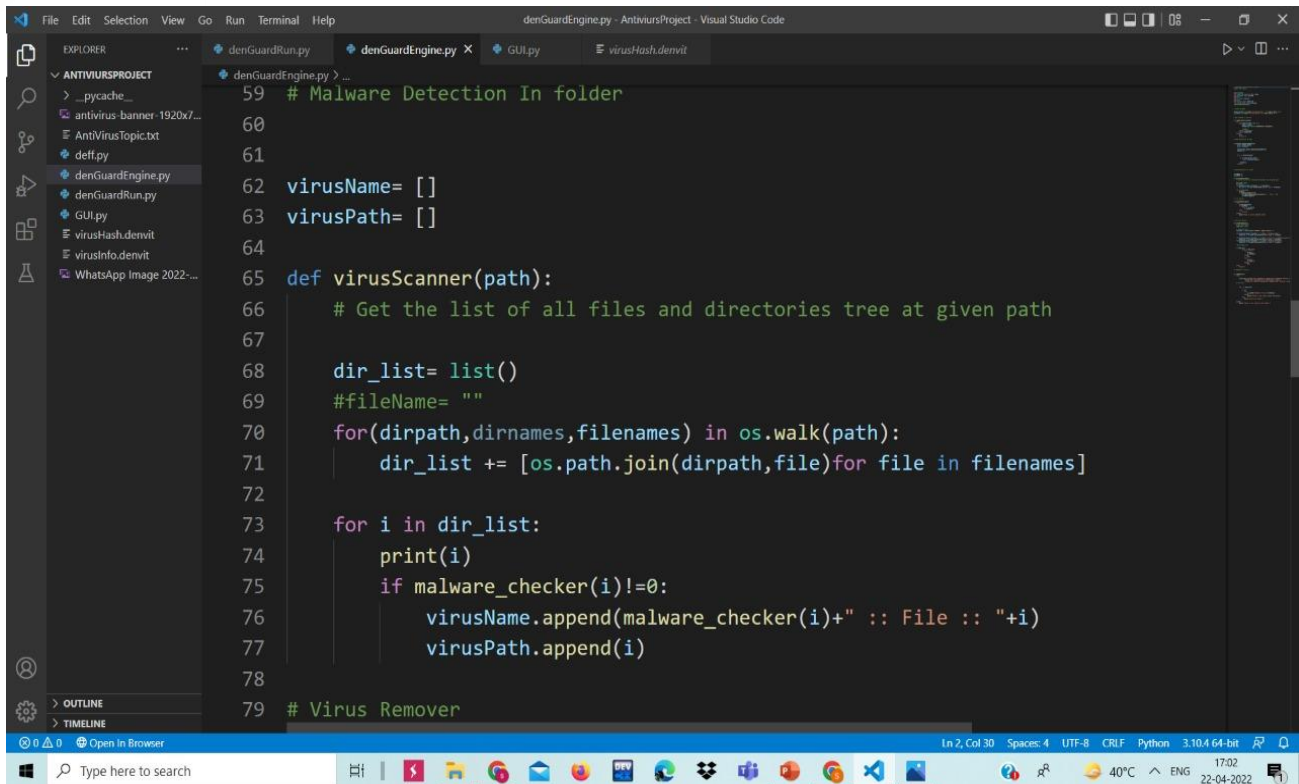
```
1
2 '''DenGuard Antivirus Engine | Script .'''
3 #import the library
4
5
6 import hashlib
7 from pathlib import Path
8 from pathlib import Path
9 import re
10 from sys import maxsize
11 import os
12 from tkinter import UNDERLINE
13 from urllib.parse import urlencode
14 #####
15
16
17 # Global Variable
18
19 malware_hashes= list(open("virusHash.denvit", "r").read().split('\n'))
20 virusInfo= list(open("virusInfo.denvit", "r").read().split('\n'))
21
```



```
23 # Get the Hash of the file
24
25 def sha256_hash(filename):
26     try:
27         with open(filename,"rb") as f:
28             bytes= f.read()
29             sha256_hash= hashlib.sha256(bytes).hexdigest()
30
31             f.close()
32             # print (sha256hash)
33             return sha256_hash
34     except:
35         pass
36         # return 0
37
38 # Malware Detection By Hash
39
40
41 def malware_checker(pathOfFile):
42     global malware_hashes
43     global virusInfo
```



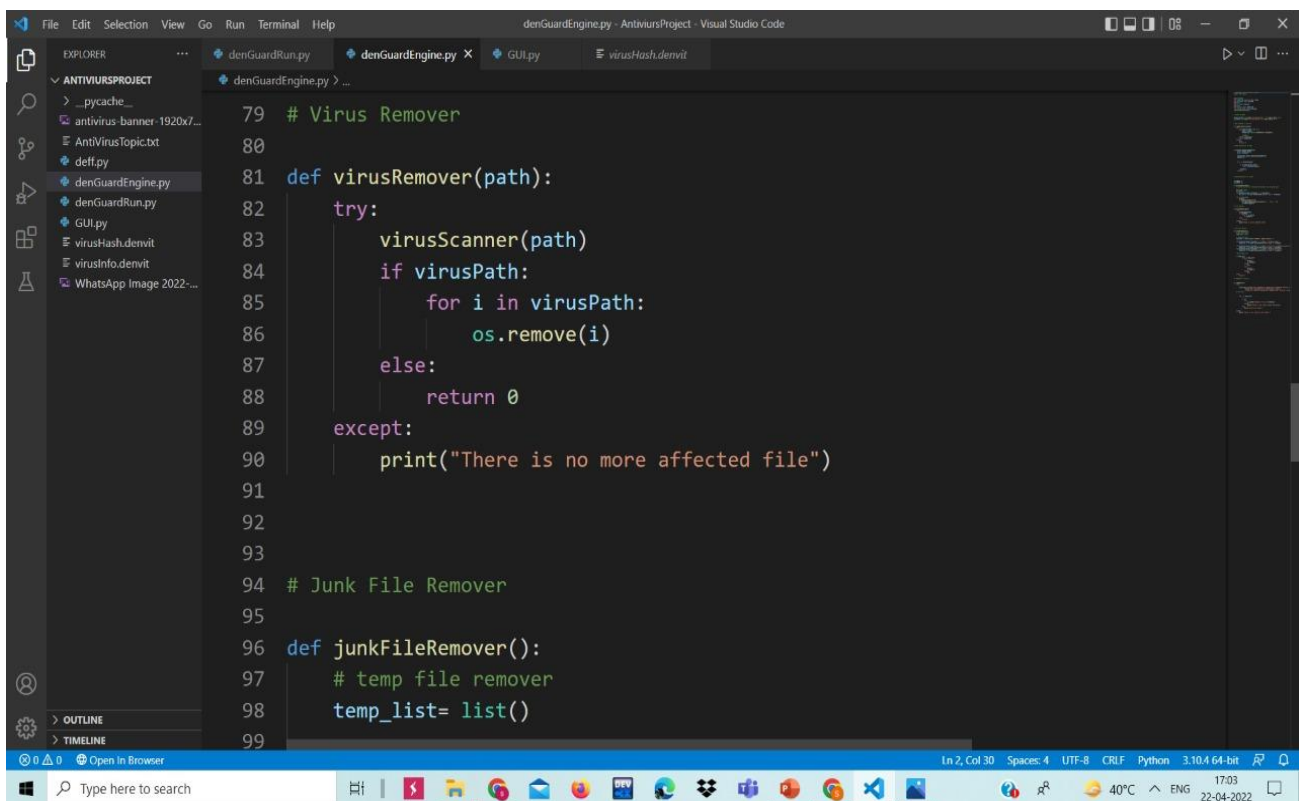
```
42     global malware_hashes
43     global virusInfo
44
45     hash_malware_check= sha256_hash(pathOfFile)
46     counter= 0
47
48
49     for i in malware_hashes:
50
51         if i== hash_malware_check:
52             return virusInfo[counter]
53
54         counter+=1
55     return 0
56
57
58
59 # Malware Detection In folder
60
61
62 virusName= []
```



```
File Edit Selection View Go Run Terminal Help
denGuardEngine.py - AntivirusProject - Visual Studio Code

EXPLORER
ANTIVIRUSPROJECT
  __pycache__
  antivirus-banner-1920x7...
  AntiVirusTopic.txt
  deff.py
  denGuardEngine.py
  denGuardRun.py
  GUI.py
  virusHash.denvit
  virusInfo.denvit
  WhatsApp Image 2022-...

denGuardEngine.py
59 # Malware Detection In folder
60
61
62 virusName= []
63 virusPath= []
64
65 def virusScanner(path):
66     # Get the list of all files and directories tree at given path
67
68     dir_list= list()
69     #fileName= ""
70     for(dirpath,dirnames,filenames) in os.walk(path):
71         dir_list += [os.path.join(dirpath,file)for file in filenames]
72
73     for i in dir_list:
74         print(i)
75         if malware_checker(i)!=0:
76             virusName.append(malware_checker(i)+" :: File :: "+i)
77             virusPath.append(i)
78
79 # Virus Remover
```

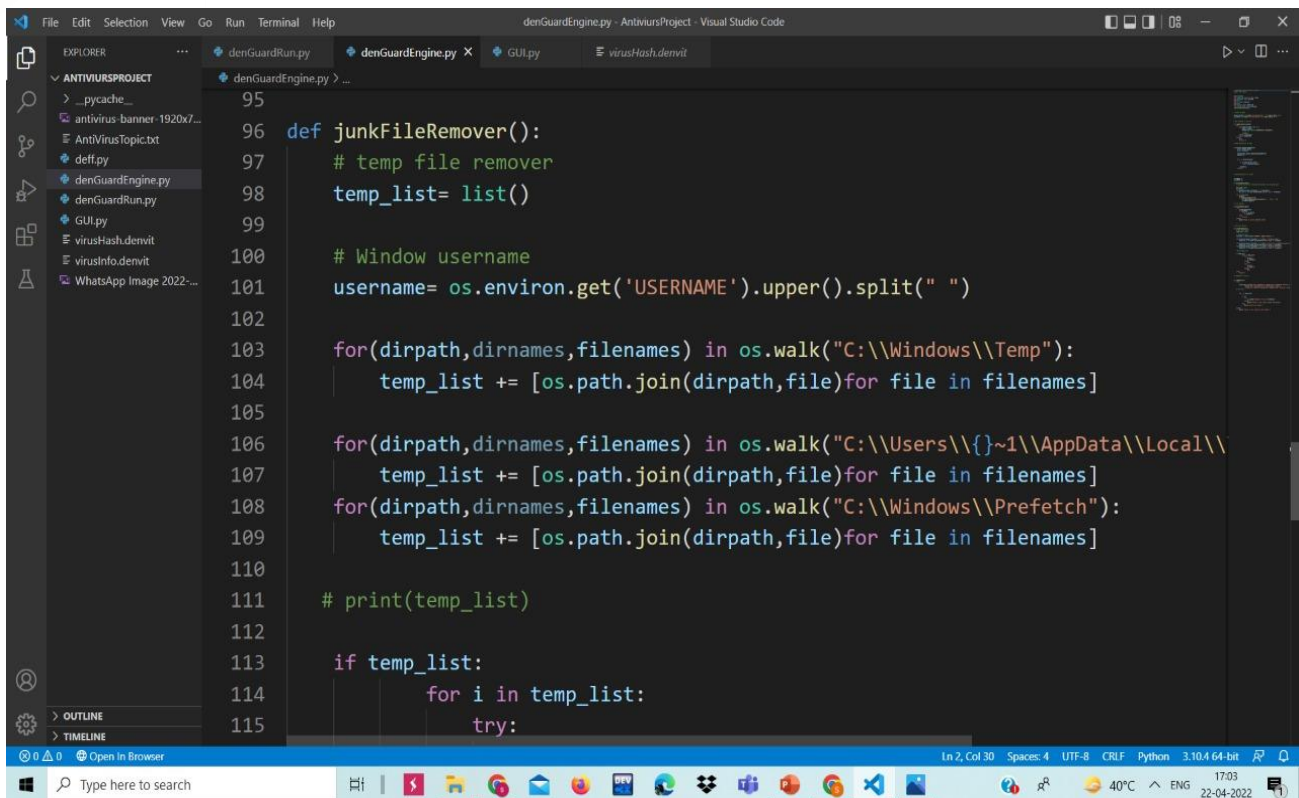


```
File Edit Selection View Go Run Terminal Help
denGuardEngine.py - AntivirusProject - Visual Studio Code

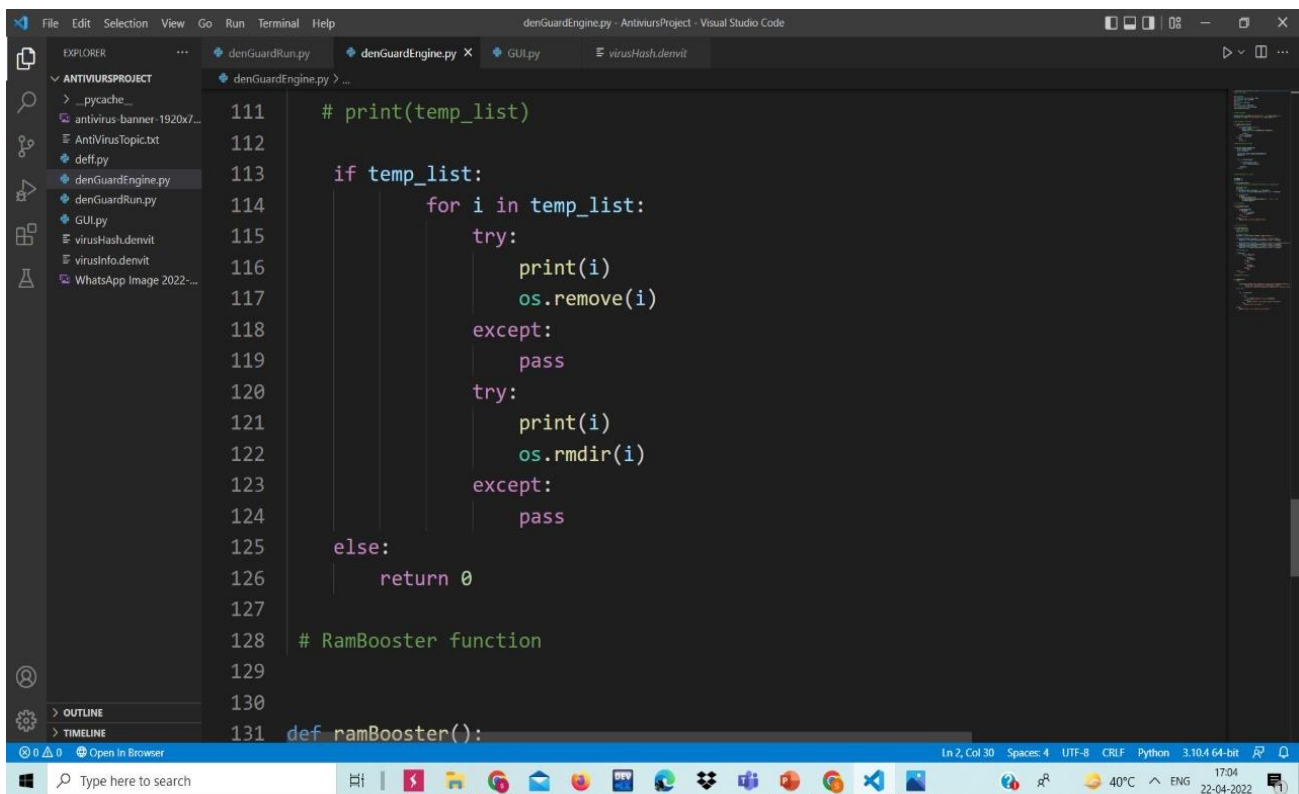
EXPLORER
ANTIVIRUSPROJECT
  __pycache__
  antivirus-banner-1920x7...
  AntiVirusTopic.txt
  deff.py
  denGuardEngine.py
  denGuardRun.py
  GUI.py
  virusHash.denvit
  virusInfo.denvit
  WhatsApp Image 2022-...

denGuardEngine.py
79 # Virus Remover
80
81 def virusRemover(path):
82     try:
83         virusScanner(path)
84         if virusPath:
85             for i in virusPath:
86                 os.remove(i)
87         else:
88             return 0
89     except:
90         print("There is no more affected file")
91
92
93
94 # Junk File Remover
95
96 def junkFileRemover():
97     # temp file remover
98     temp_list= list()
99
```





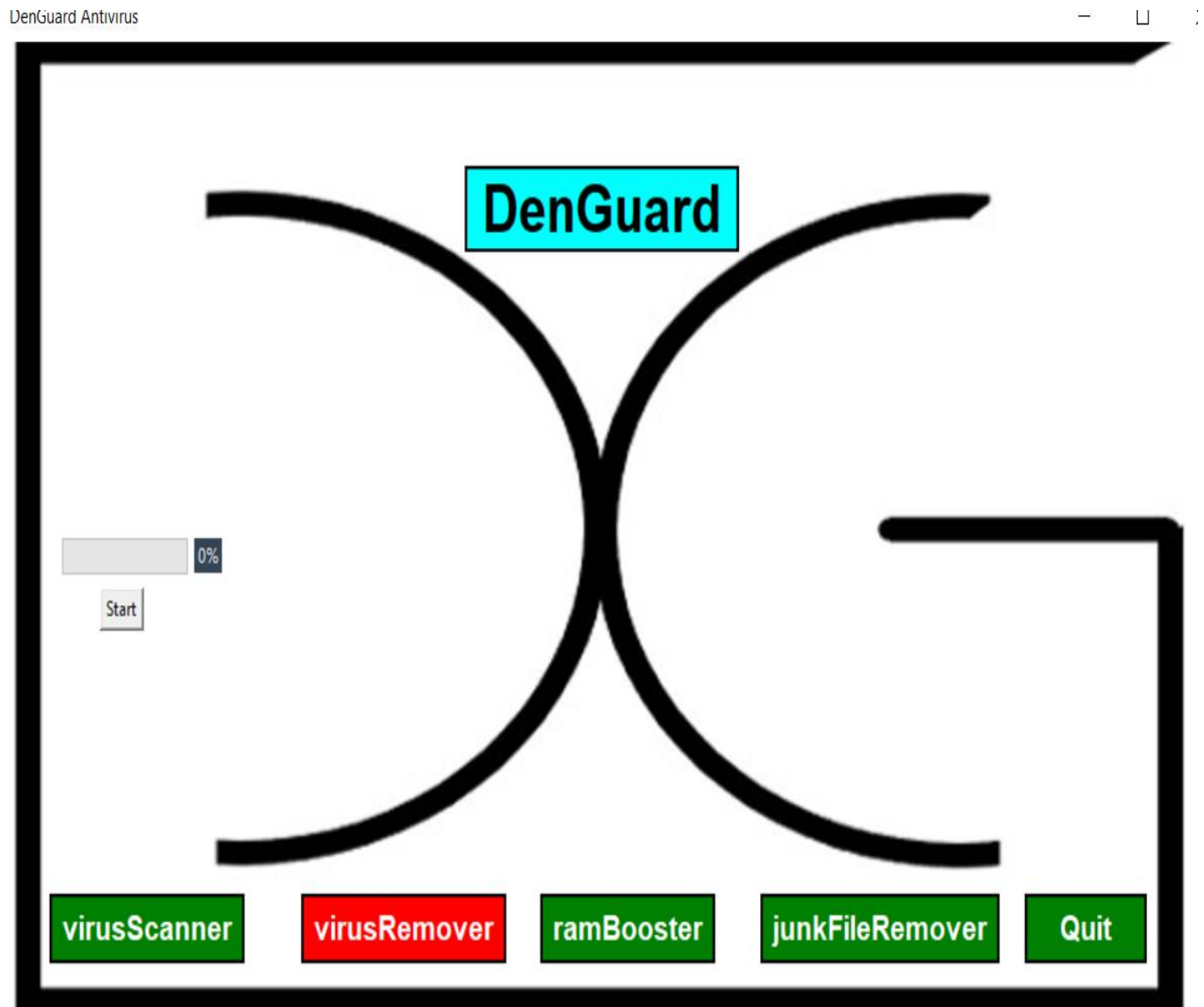
```
95
96 def junkFileRemover():
97     # temp file remover
98     temp_list= list()
99
100     # Window username
101     username= os.environ.get('USERNAME').upper().split(" ")
102
103     for(dirpath,dirnames,filenames) in os.walk("C:\\Windows\\Temp"):
104         temp_list += [os.path.join(dirpath,file)for file in filenames]
105
106     for(dirpath,dirnames,filenames) in os.walk("C:\\Users\\{}~1\\AppData\\Local\\
107         temp_list += [os.path.join(dirpath,file)for file in filenames]
108     for(dirpath,dirnames,filenames) in os.walk("C:\\Windows\\Prefetch"):
109         temp_list += [os.path.join(dirpath,file)for file in filenames]
110
111     # print(temp_list)
112
113     if temp_list:
114         for i in temp_list:
115             try:
```



```
111     # print(temp_list)
112
113     if temp_list:
114         for i in temp_list:
115             try:
116                 print(i)
117                 os.remove(i)
118             except:
119                 pass
120             try:
121                 print(i)
122                 os.rmdir(i)
123             except:
124                 pass
125         else:
126             return 0
127
128     # RamBooster function
129
130
131 def ramBooster():
```

This is the snapshot of our code, which is written in visual studio code. Visual Studio is an IDE made by Microsoft and used for different types of software development such as computer programs, websites, web apps, web services, and mobile apps.

### 5.3 PROTOTYPE SUBMISSION





## 5.4 TEST AND VALIDATION

```
1 import denGuardEngine
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

D:\VS code\view folder\AntivirusProject\WhatsApp Image 2022-01-26 at 21.38.12.jpeg  
D:\VS code\view folder\AntivirusProject\denGuardEngine.py  
D:\VS code\view folder\AntivirusProject\\_pycache\\_dendron.cpython-39.pyc  
D:\VS code\view folder\AntivirusProject\\_pycache\\_tkinter.cpython-39.pyc  
D:\VS code\view folder\AntivirusProject\\_pycache\\_denGuard.cpython-39.pyc  
D:\VS code\view folder\AntivirusProject\\_pycache\\_denGuardEngine.cpython-310.pyc  
The virus is: []

D:\VS code\view folder\AntivirusProject\GUI.py  
D:\VS code\view folder\AntivirusProject\virusHash.denvit  
D:\VS code\view folder\AntivirusProject\virusInfo.denvit  
D:\VS code\view folder\AntivirusProject\deff.py  
D:\VS code\view folder\AntivirusProject\denGuardRun.py  
D:\VS code\view folder\AntivirusProject\AntiVirusTopic.txt  
D:\VS code\view folder\AntivirusProject\antivirus-banner-1920x744.jpg  
D:\VS code\view folder\AntivirusProject\WhatsApp Image 2022-01-26 at 21.38.12.jpeg  
D:\VS code\view folder\AntivirusProject\denGuardEngine.py  
D:\VS code\view folder\AntivirusProject\\_pycache\\_dendron.cpython-39.pyc  
D:\VS code\view folder\AntivirusProject\\_pycache\\_tkinter.cpython-39.pyc  
D:\VS code\view folder\AntivirusProject\\_pycache\\_denGuard.cpython-39.pyc  
D:\VS code\view folder\AntivirusProject\\_pycache\\_denGuardEngine.cpython-310.pyc

C:\Windows\Temp\CMC1ploader.log  
C:\Windows\Temp\CMC1ploader.log  
C:\Windows\Temp\LAPTOP-FHMRJULI-20220419-0709.log  
C:\Windows\Temp\officeclicktorun.exe\_streamserver(20220419070909478).log  
C:\Windows\Temp\officeclicktorun.exe\_streamserver(20220419070909478).log  
C:\Windows\Temp\ELAN MBF LOG\ELan FP-Driver Log-0.htm  
C:\Windows\Temp\ELAN MBF LOG\ELan FP-Driver Log-0.htm  
C:\Windows\Temp\ELAN MBF LOG\ELan FP-Engine Log-2.htm  
C:\Windows\Temp\ELAN MBF LOG\ELan FP-Engine Log-2.htm  
C:\Windows\Temp\ELAN MBF LOG\ELan FP-Sensor Log-2.htm  
C:\Windows\Temp\ELAN MBF LOG\ELan FP-Sensor Log-2.htm  
C:\Windows\Temp\ELAN MBF LOG\ELan FP-Storage Log-2.htm  
C:\Windows\Temp\ELAN MBF LOG\ELan FP-Storage Log-2.htm

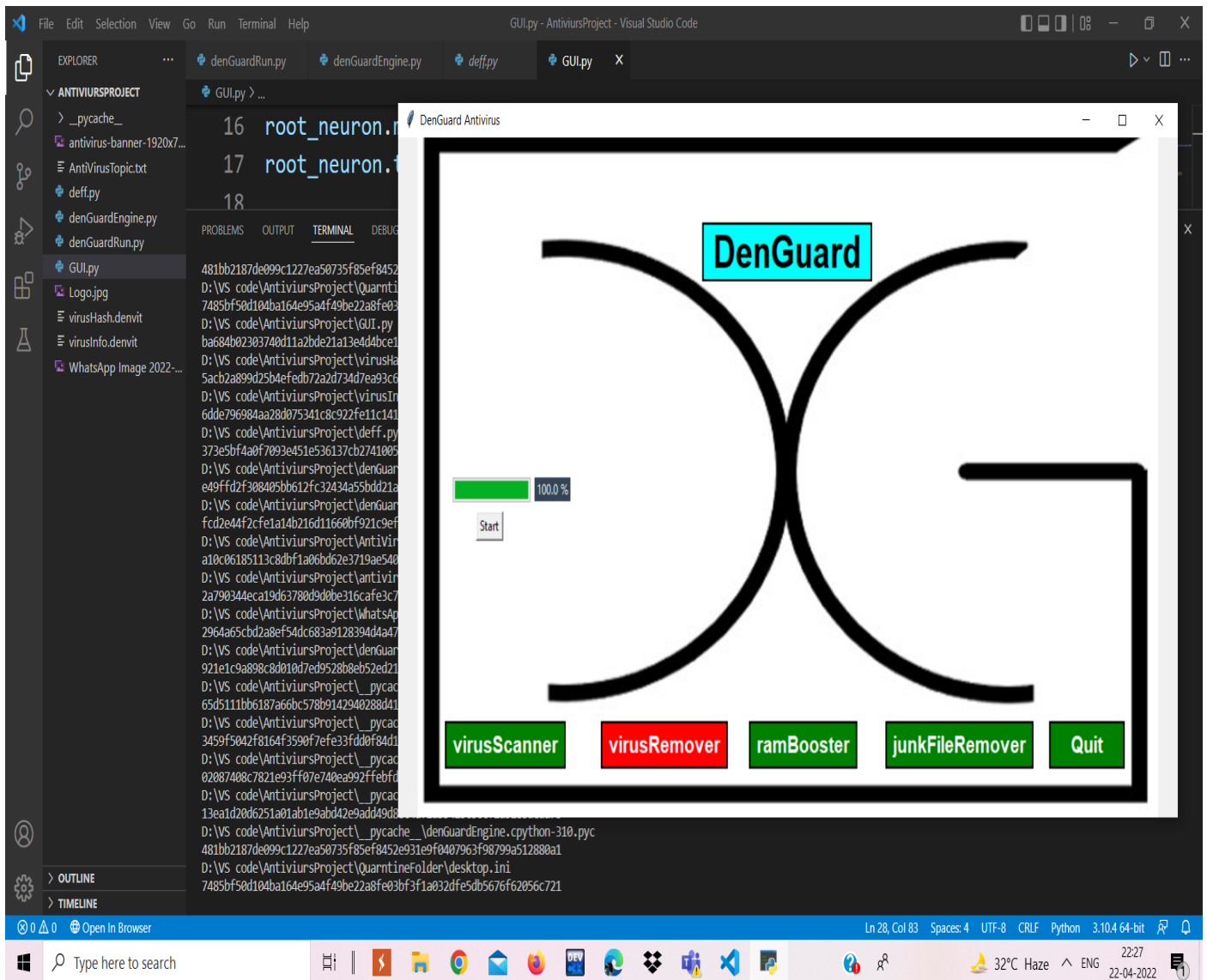
=====

PS D:\VS code\view folder\AntivirusProject>

Open in Browser

Type here to search

Ln 18, Col 2 Spaces: 4 UTF-8 CRLF Python 3.10.4 64-bit 15:55 22-04-2022



As we can see from above images our tool is working properly.

## 5.5 Performance Analysis

# COMPARISON TABLE

Sr. No.	Folder/file Path	Time taken by tool	Time taken by Virus Total
1.	"D:\\VS code\\AntiviursProject"	3.45 s	10.04 s
2.	"D:\\VS code\\AntiviursProject\\WhatsApp Image 2022-01-26 at 21.38.12.jpeg"	1.1 s	3.44 s
3.	"D:\\VS code\\New folder\\AntiviursProject\\antivirus-banner-1920x744.jpg"	1.2 s	3.56 s

## 5.6 SUMMARY

As we can see It scans more than 20 files containing a folder within a second. Detected the virus from those files with name. So it is very fast tool. The basic Idea behind this tool is to detect the virus and remove it from the system. In the database we have stored we stored the hashes of the virus, so when we run the program, it immediately checks the hashes of each files containing the particular folder with the hash which is stored in the database if any of the hash is matches with the hashes stored in the folder means the folder contains the virus so it will detect the name of the virus and remove it from the folder.

## CHAPTER – 6

### PROJECT OUTCOME AND APPLICABILITY

#### 6.1 OUTLINE

An antivirus software, as the name indicates, is a program that works against a virus. It detects or recognizes the virus, and then after detecting the presence of the virus, it works on removing it from

the computer system. Antivirus software works as a prophylactic so that it not only eliminates a virus but also prevents any potential virus from infecting your computer in the future.

## **6.2 SIGNIFICANT PROJECT OUTCOMES**

- Malware and virus protection.
- Defence against data thieves.
- Increases your computer's lifetime.
- Comprehensive threat protection.
- Available at low cost.

## **6.3 PROJECT APPLICABILITY ON REAL-WORLD APPLICATIONS**

Antivirus software typically runs as a background process, scanning computers, servers or mobile devices to detect and restrict the spread of malware. Many antivirus software programs include real-time threat detection and protection to guard against potential vulnerabilities as they happen, as well as system scans that monitor device and system files looking for possible risks.

Antivirus software usually performs these basic functions:

- Scanning directories or specific files for known malicious patterns indicating the presence of malicious software;
- Allowing users to schedule scans so they run automatically;
- Allowing users to initiate new scans at any time; and
- Removing any malicious software, it detects. Some antivirus software programs do this automatically in the background, while others notify users of infections and ask them if they want to clean the files.

## **6.4 INFERENCE**

This chapter provides a high-level introduction to methods and tools of antivirus. Antivirus is a kind of software used to prevent, scan, detect and delete viruses from a computer. Once installed, most antivirus software runs automatically in the background to provide real-time protection against virus

attacks. Comprehensive virus protection programs help protect your files and hardware from malware such as worms, Trojan horses and spyware, and may also offer additional protection such as customizable firewalls and website blocking.

## **CHAPTER – 7**

### **CONCLUSIONS AND RECOMMENDATION**

#### **7.1 OUTLINE**

Antivirus software acts as the final line of defense for your PC(S) and other devices. Which means it can protect – or at least mitigate threats – your devices when every other security software fails. Therefore, do not underestimate its potential and never leave your devices lying around without antivirus in them. Install antivirus, stay protected against the modern-day malware!

#### **7.2 LIMITATION/CONSTRAINTS OF THE SYSTEM**

In this software there are some limitations such as-

- It lacks the internet protection.
- It lacks the real-time protection.
- Absence of dynamic features.
- Instruction is needed for every step.
- User interaction is mandatory while it is executing.

#### **7.3 FUTURE ENHANCEMENTS**

In future we will be adding following features:

- We will to add real time protection feature.
- We will try to minimize the above-mentioned limitations.

#### **7.4 INFERENCE**

The basic idea is that scan any files or folder in very fast manner and save the time of the user. Our objective is to develop the field of security. Our tool can finish scanning 15 files in less than 1 second! Anti-Virus proves to be useful in many cases, an authorized user can use this tool to

see is there any files in their system which is containing the any kind of malware (virus, worms, trojan) so, that it can detect and remove from the system.

The basic Idea behind this tool is to detect the virus and remove it from the system. In the database we have stored we stored the hashes of the virus, so when we run the program, it immediately checks the hashes of each files containing the particular folder with the hash which is stored in the database if any of the hash is matches with the hashes stored in the folder means the folder contains the virus so it will detect the name of the virus and remove it from the folder.

## **APPENDIX A**

### **Virus**

A computer virus is a type of computer program that, when executed, replicates itself by modifying other computer programs and inserting its own code. If this replication succeeds, the affected areas are then said to be "infected" with a computer virus, a metaphor derived from biological viruses. Computer viruses generally require a host program. The virus writes its own code into the host program. When the program runs, the written virus program is executed first, causing infection and damage. A computer worm does not need a host program, as it is an independent program or code chunk. Therefore, it is not restricted by the host program, but can run independently and actively carry out attacks.

### **Worms**

A computer worm is a standalone malware computer program that replicates itself in order to spread to other computers. It often uses a computer network to spread itself, relying on security failures on the target computer to access it. It will use this machine as a host to scan and infect other computers. Computer worms use recursive methods to copy themselves without host programs and distribute themselves based on the law of exponential growth, thus controlling and infecting more and more computers in a short time.

## **Trojan Horse**

A Trojan Horse Virus is a type of malware that downloads onto a computer disguised as a legitimate program. The delivery method typically sees an attacker use social engineering to hide malicious code within legitimate software to try and gain users' system access with their software.

## **GUI (Graphical User Interface)**

A graphical user interface (GUI) is a type of user interface through which users interact with electronic devices via visual indicator representations. Graphical user interfaces would become the standard of user-centered design in software application programming, providing users the capability to intuitively operate computers and other electronic devices through the direct manipulation of graphical icons such as buttons, scroll bars, windows, tabs, menus, cursors, and the mouse pointing device. Many modern graphical user interfaces feature touchscreen and voice-command interaction capabilities.

## **RELATED WORK INVESTIGATION**

As we analyzed different types of software, that too large in number, common findings were -

- Most of the tools required compatible devices or software.
- They were not attack resilient but vulnerable.
- Very large number of tools were available online, but to use most of them we have to first download them then we were able to use them.
- And also, they were very slow

## **REFERENCES**

1. <https://docs.python.org/3/library/hashlib.html>
2. <https://docs.python.org/3/library/os.html#module-os>
3. <https://docs.python.org/3/library/sys.html#module-sys>

4. <https://www.verizon.com>
5. <https://www.codeproject.com/Questions/5290494/Python-can-you->
6. <https://pythontutorialsolveissue.blogspot.com/2020/11/solved-attributeerror-type-object-image.html>
7. <https://pythonguides.com/python-tkinter-progress-bar/>