DNS QUERY TOOL

21CSCT302J /COMPUTER NETWORKS

MINI PROJECT

Submitted by

YUGANDHAR.S :RA2211042020005

SAHIL KADAM   :RA2211042020047

CHARAN KUMAR : RA2211042020058

Under the guidance of

Dr. G. Gangadevi

(Assistant Professor, Department of Computer Science and Engineering)

V SEMESTER/ III YEAR

COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM, CHENNAI

October 2024

1

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University U/S 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project titled **DNS Query Tool** is the bonafide work of **SAHIL KADAM (RA2211042020047), YUGANDHAR (RA2211042020005), CHARAN SURANA (RA2211042020058)** who carried out the project work under my supervision. This project work confirms to 21CSC302J /Computer Networks, V Semester, III year, 2024.

SIGNATURE
Dr. G. Gangadevi, M. Tech, Ph.D.,
Assistant Professor
Computer Science and Engineering,
SRM Institute of Science and Technology,
Ramapuram, Chennai.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM, CHENNAI

DECLARATION

We hereby declare that the entire work contained in this mini project titled "**DNS Query Tool**" has been carried out by **SAHIL KADAM** (REG NO: RA2211042020047), **YUGANDHAR** (REG NO: RA2211042020005), **CHARAN SURANA** (REG NO: RA2211042020058) at SRM Institute of Science and Technology, Ramapuram, Chennai, under the guidance of Dr. G. Gangadevi, Assistant professor, Department of Computer Science and Engineering.

Place: CHENNAI   **YUGANDHR**
DATE   **SAHIL KADAM**
                             CHARAN KUMAR

<u>ABSTRACT</u>

The DNS Query Tool is an innovative desktop application developed to facilitate seamless DNS lookups and reverse DNS queries through a graphical user interface (GUI). This project harnesses the power of Python, specifically utilizing libraries like dnspython, Tkinter, and Pillow, to provide users with a straightforward method for obtaining essential DNS information. Users can query various DNS record types, including A, AAAA, CNAME, MX, TXT, NS, and SOA records, which are crucial for various networking tasks. Furthermore, the tool allows users to perform reverse DNS lookups, converting IP addresses back into domain names.This project aims to bridge the gap between technical complexity and user accessibility, particularly for those who may lack extensive experience with command-line tools. The GUI is designed with user experience in mind, offering a simple and intuitive interface that enhances the process of DNS querying. Additionally, to make the application visually appealing, animated GIFs are employed as background elements, contributing to an engaging user experience.By addressing the usability issues of traditional command-line utilities, this tool not only serves the needs of network administrators and IT professionals but also empowers general users who require quick access to DNS information without having to navigate complicated commands. This report outlines the comprehensive aspects of the DNS Query Tool project, including its objectives, architecture, and potential for future enhancements.

TABLE OF CONTENTS

<u>INTRODUCTION</u>

The Domain Name System (DNS) is a fundamental component of the internet that translates human-friendly domain names into machine-readable IP addresses. It serves as a directory service for the internet, allowing users to access websites and services using easily remembered names rather than complex numerical addresses. However, the underlying complexity of DNS operations often poses challenges, particularly for those unfamiliar with its workings.

The DNS Query Tool is designed to simplify these operations by providing an intuitive graphical interface for performing DNS lookups. It allows users to query various types of DNS records with minimal effort. For example, a user might want to check the MX records for their domain to troubleshoot email routing issues or perform a reverse lookup to identify the domain associated with a specific IP address. In essence, this project serves as a bridge between the technical world of DNS and the everyday user, enabling quick and accurate information retrieval. The development of this tool was motivated by the need to create an accessible solution that combines essential functionalities with an engaging user experience. By integrating a GUI with powerful backend operations, the DNS Query Tool stands to enhance the efficiency and effectiveness of DNS queries.

Problem Statement

Despite the critical role that DNS plays in internet functionality, many users struggle with the complexities of DNS querying due to the dominance of command-line tools. Utilities such as nslookup, dig, and host offer powerful functionalities but require a level of technical understanding that can be daunting for less experienced users. The output of these tools is often raw text, which may not be easily interpretable, particularly for users unfamiliar with DNS terminology or the specifics of different record types.

Furthermore, existing graphical DNS tools tend to either be overly complicated, featuring a myriad of options that can overwhelm users, or they may lack essential functionalities needed for basic queries. This disconnect between user needs and tool capabilities creates an opportunity for a new solution: a straightforward, efficient DNS querying tool that offers the necessary features in an intuitive format.

OBJECTIVE

The primary objective of the DNS Query Tool is to develop an application that makes DNS querying accessible to users of all technical backgrounds. This objective can be broken down into several specific goals:

User-Friendly Interface: Create an intuitive GUI that enables users to perform DNS lookups with minimal effort. The design should be straightforward, allowing for easy navigation and interaction. Comprehensive DNS Query Capabilities: Support various DNS record types, including A (IPv4 address), AAAA (IPv6 address), MX (Mail Exchange), CNAME (Canonical Name), TXT (Text), NS (Name Server), and SOA (Start of Authority). This comprehensive support ensures users can gather the information they need for various purposes. Reverse DNS Lookup Functionality: Implement a feature that allows users to input an IP address and retrieve the corresponding domain name, facilitating troubleshooting and identification tasks. Real-Time Results Display: Ensure that the results of DNS queries are displayed immediately and clearly in the GUI, allowing users to understand the information presented without delay. Enhanced User Experience: Integrate dynamic visual elements, such as animated GIFs, to make the application visually engaging and to improve user satisfaction during interactions. Clear Error Handling: Implement robust error handling to provide users with meaningful messages in case of invalid input or issues with DNS resolution, thus enhancing the overall user experience. By achieving these objectives, the DNS Query Tool aims to serve as a reliable and efficient resource for DNS information retrieval, meeting the needs of both technical and non-technical users.

SCOPE OF PROJECT

The DNS Query Tool project is designed to provide a comprehensive set of functionalities aimed at various user demographics. The target users include network administrators, web developers, IT support staff, and general users. Network administrators, responsible for maintaining the integrity and performance of networks, can utilize the tool to regularly check DNS records for troubleshooting purposes. Web developers, who are involved in website development and deployment, can benefit from quick access to DNS records for domain verification and troubleshooting. IT support staff, tasked with assisting users with technical issues, can use the tool's DNS lookup capabilities to diagnose problems efficiently. Additionally, general users can look up domain or IP information without needing to resort to complex command-line tools. The core features of the DNS Query Tool include DNS record lookup, allowing users to query for various types of records such as A, AAAA, MX, CNAME, TXT, NS, and SOA, which serve a wide range of networking tasks. The tool also offers reverse DNS lookup, enabling users to enter an IP address to find the corresponding domain name, assisting in network diagnostics and identification. A graphical user interface (GUI) designed using Tkinter ensures accessibility for users with different technical expertise, providing a clean and intuitive experience. To further enhance user engagement, the interface includes dynamic background animations with GIFs, adding a visually appealing element to the tool. Lastly, the results of DNS queries are displayed clearly and in a format that is easy to read and understand, ensuring that users can quickly interpret and use the information provided.

Existing System

In the realm of DNS querying, several command-line tools have established themselves as the go-to solutions for network administrators and developers alike. Utilities such as nslookup, dig, and host are known for their robust functionality and ability to provide comprehensive DNS information. However, these tools also present challenges, particularly for non-technical users.

nslookup is a widely used utility that allows users to retrieve DNS records by querying DNS servers. While it is effective, its command syntax can be a hurdle for casual users or those not familiar with the command-line interface. Similarly, dig is a powerful tool that provides in-depth details on DNS queries and responses, often making it the tool of choice for professionals looking for a granular analysis of DNS. However, its complexity and the verbosity of its output make it less user-friendly, especially for those who are not well-versed in DNS terminology. On the simpler side, host offers more straightforward DNS lookups, providing a cleaner, easier-to-understand output compared to dig, but it still relies on a command-line interface. For non-technical users, the requirement to understand and use command-line commands can be a significant barrier.

While these tools are excellent for technical users needing detailed DNS data, they are not designed with user experience in mind. The plain text format of the output can be confusing or overwhelming, especially for users unfamiliar with DNS records and query results. Thus, despite their technical capabilities, these command-line tools do little to bridge the gap for users who need a more intuitive, accessible interface for DNS querying.

Hardware and Software Requirements

To run the DNS Query Tool effectively, the following hardware and software specifications are recommended:

Hardware Requirements:

Processor: Intel i3 or higher

RAM: 4 GB or more

Storage: 100 MB of free disk space

Display: 1280x720 resolution or higher

Software Requirements:

Operating System: Windows 10, macOS, or Linux

Python Version: Python 3.7 or later

Required Libraries:

dnspython for DNS resolution

Tkinter for GUI development

Pillow for image processing (including GIFs)

ttk for advanced widgets like combobox

Additional Tools: Any IDE (e.g., VS Code, PyCharm) for code development and debugging

PROPOSED SYSTEM

The proposed system brings a significant advancement in how DNS queries are handled, emphasizing accessibility and functionality. Its design focuses on making DNS data available to a broader range of users, especially those who may not be familiar with the technicalities of command-line tools like `dig` or `nslookup`. By offering a graphical user interface (GUI), the tool removes the need for users to memorize complex commands or interpret raw outputs, replacing it with an intuitive and visually driven approach.

The graphical interface is built to accommodate various types of DNS records, such as A, AAAA, MX, CNAME, TXT, NS, and SOA. This means users can perform everything from basic IP address lookups to more sophisticated tasks, like retrieving email server details or verifying domain aliasing, all from a simple, easy-to-use platform. The system also ensures a fluid experience by incorporating error handling mechanisms that provide clear and informative feedback if a query fails due to non-existent domains, missing records, or incorrect formats.

In addition to simplifying the querying process, the proposed system introduces a new layer of automation. For example, when users search for CNAME records, the tool automatically adjusts for common domain variations, like whether a domain includes the "www" prefix or not. This ensures that users get the most accurate information without having to manually adjust their inputs or re-query domains. Such automation saves time and effort, making the tool more efficient for both novice users and experienced IT professionals.

The system's modular design ensures that it is not only effective today but adaptable for the future. It is built in a way that allows additional features to be integrated without overhauling the core structure. For instance, future updates could include enhanced support for DNSSEC (Domain Name System Security Extensions), providing verification that DNS queries are authentic and have not been tampered with, making the tool more secure. Other enhancements could involve incorporating real-time DNS monitoring, giving users the ability to track changes in DNS records over time, which can be useful for domain management and security monitoring.

The ability to perform reverse DNS lookups is another notable feature of the proposed system. This functionality allows users to input an IP address and retrieve the associated domain name, which is particularly useful for tasks like troubleshooting network issues or verifying server identities. It's a feature that enhances the overall utility of the tool by providing more comprehensive DNS insights, further reducing the need for users to rely on multiple separate utilities.

Moreover, the system is designed with performance optimization in mind. As the user base grows or queries become more complex, the tool can be extended to handle multiple DNS queries concurrently using multi-threading, which boosts performance. Such an enhancement would be particularly beneficial for IT professionals managing large networks, as it would allow them to conduct bulk lookups and analyze results in a fraction of the time required by traditional tools.

Furthermore, the system's user interface is customizable, offering the flexibility to adjust aesthetics or incorporate branding. This could make the tool adaptable for enterprise environments, where an organization's IT team could use it in internal documentation or as part of a larger network management system. Similarly, visual elements like animated backgrounds or personalized themes could be added to improve the user experience, making the tool engaging without sacrificing performance.

To ensure the system remains effective, future enhancements could also introduce support for mobile devices, allowing users to perform DNS queries from smartphones or tablets. This would extend the tool's reach, making it available to users who are not always at their desktops but still need access to DNS data while on the go. Mobile compatibility would also be beneficial for IT administrators,

The proposed system offers a comprehensive solution that enhances both the usability and functionality of DNS querying. By focusing on ease of use, the system lowers the barrier to entry for casual users while still offering the powerful features and flexibility needed by professionals.
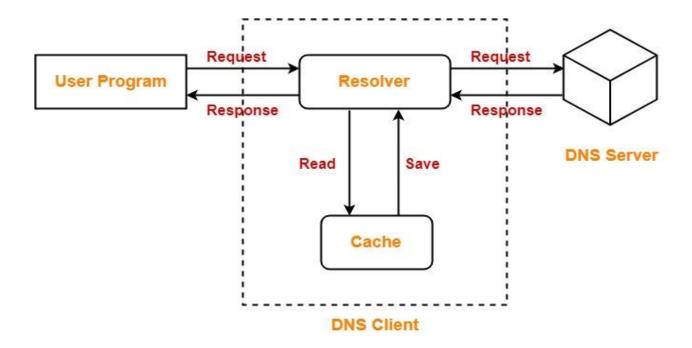
Architecture Diagram

Figure 1: Block Diagram of DNS Query Tool
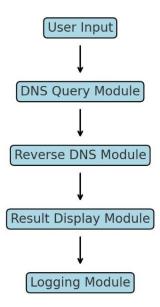
MODULES DESCRIPTION

The DNS Query Tool is a Python application that allows users to perform DNS lookups and reverse DNS queries through a user-friendly graphical interface. Built using the `tkinter` library for the GUI and the `dnspython` library for DNS resolution, this tool is versatile and capable of retrieving various types of DNS records.

Key Modules

The DNS Record Lookup tool allows users to query various types of DNS records, including A (Address Record) for mapping domain names to IPv4 addresses, AAAA (IPv6 Address Record) for IPv6 mapping, CNAME (Canonical Name Record) for domain name aliases, MX (Mail Exchange Record) for specifying mail servers for email delivery, TXT (Text Record) for holding text-based information, NS (Name Server Record) for identifying authoritative DNS servers, and SOA (Start of Authority Record) for administrative zone details. Additionally, the tool includes a Reverse DNS Lookup feature, enabling users to input an IP address to retrieve the corresponding domain name, which is useful for verifying a server's identity.

The application comes with a user-friendly graphical interface (GUI) built using `tkinter`, making it accessible to users without programming experience. The layout includes input fields for domain names and IP addresses, dropdown menus for selecting record types, and buttons to execute queries. Enhancing the visual appeal, the application features an animated background using a GIF, providing an engaging user experience. Robust error handling is also integrated, ensuring that the tool gracefully manages errors, such as non-existent domains or failed DNS queries, while offering informative feedback to users.

Figure 2: Module Interaction in DNS Query Tool

Usage Instructions

To launch the application, run the Python script containing the provided code, ensuring that all required libraries, such as `dnspython`, `tkinter`, and `PIL`, are installed in your Python environment. To perform a DNS query, simply enter a valid domain name in the input field, select the desired record type from the dropdown menu, and click on the "Query DNS" button to retrieve and display the corresponding DNS records. For a reverse DNS lookup, input a valid IP address into the designated field and click the "Reverse DNS Lookup" button to obtain and display the associated domain name. To clear all input fields and reset previous results, click on the "Clear" button.

Dependencies

The `dnspython` library plays a crucial role in the functionality of the application, as it is responsible for performing DNS queries and reverse lookups, enabling the retrieval of various DNS records such as A, AAAA, MX, CNAME, and others. This library allows the application to interact with DNS servers and obtain the necessary information for the user.

In addition to `dnspython`, the application utilizes `tkinter`, which is the standard GUI toolkit for Python. This toolkit is fundamental in building the user interface, providing interactive elements such as input fields, dropdown menus, and buttons. Through `tkinter`, the application offers an intuitive and visually accessible interface, making it easy for users to perform DNS queries and reverse lookups without needing any programming knowledge. To enhance the visual appeal of the application, the Python Imaging Library (PIL) is employed for image processing tasks. Specifically, PIL enables the application to display animated GIFs in the background, adding a dynamic and engaging visual element to the user interface. The combination of these libraries ensures that the application is not only functional and user-friendly but also visually engaging, providing a seamless experience for users conducting DNS lookups and reverse queries.
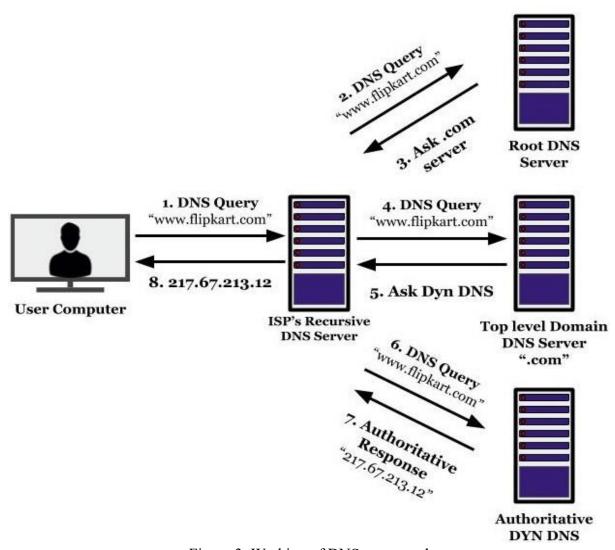

Figure 3: Working of DNS query tool

17

CODE FOR DNS QUERY TOOL

```python
import dns.resolver
import dns.reversename
from tkinter import Tk, Label, Button, Entry, Frame, StringVar, ttk
from PIL import Image, ImageTk

# Function to perform DNS lookup for A, AAAA, CNAME, MX, TXT, NS, and SOA records
def dns_query(domain, record_type='A'):
    try:
        answers = dns.resolver.resolve(domain, record_type)
        return [str(answer) for answer in answers]
    except dns.resolver.NoAnswer:
        return [f"No {record_type} records found for {domain}."]
    except dns.resolver.NXDOMAIN:
        return [f"The domain {domain} does not exist."]
    except Exception as e:
        return [f"Error occurred: {e}"]

# Function to perform reverse DNS lookup
def reverse_dns_lookup(ip_address):
    try:
        rev_name = dns.reversename.from_address(ip_address)
        domain_name = dns.resolver.resolve(rev_name, "PTR")[0]
        return f"Reverse DNS for {ip_address}: {domain_name}"
    except Exception as e:
        return f"Error occurred: {e}"

# Function to initialize the GUI
def init_gui():
    global domain_var, record_type_var, result_label, ip_var, gif_label

    # Create the main window
    root = Tk()
    root.title("DNS Query Tool")
    root.geometry("800x600")  # Set window size

    # Create a label to hold the GIF frames
    gif_label = Label(root)
    gif_label.place(x=0, y=0, relwidth=1, relheight=1)  # Make label cover the entire window

    # Start the GIF playback
    play_gif()

    # Domain input
    domain_var = StringVar()
    domain_label = Label(root, text="Enter Domain Name:", bg="lightblue", font=("Arial",
12))
    domain_label.pack(pady=5)
```
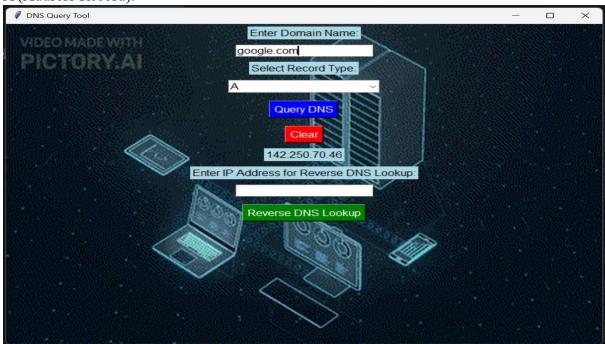
```python
    domain_entry = Entry(root, textvariable=domain_var, font=("Arial", 12))
    domain_entry.pack(pady=5)

    # Record type selection (Combobox)
    record_type_var = StringVar()
    record_type_label = Label(root, text="Select Record Type:", bg="lightblue", font=("Arial",
12))
    record_type_label.pack(pady=5)

    record_types = ['A', 'AAAA', 'CNAME', 'MX', 'TXT', 'NS', 'SOA']
    record_type_combobox    =    ttk.Combobox(root,    textvariable=record_type_var,
values=record_types, font=("Arial", 12))
    record_type_combobox.pack(pady=5)
    record_type_combobox.current(0)  # Set default selection to the first item

    # Query button
    query_button = Button(root, text="Query DNS", command=perform_dns_query, bg="blue",
fg="white", font=("Arial", 12))
    query_button.pack(pady=10)

    # Clear button
    clear_button = Button(root, text="Clear", command=clear_fields, bg="red", fg="white",
font=("Arial", 12))
    clear_button.pack(pady=5)

    # Result label
    result_label = Label(root, text="", bg="lightblue", wraplength=500, font=("Arial", 12))
    result_label.pack(pady=5)

    # Reverse DNS lookup input
    ip_var = StringVar()  # Initialize ip_var here
    ip_label = Label(root, text="Enter IP Address for Reverse DNS Lookup:", bg="lightblue",
font=("Arial", 12))
    ip_label.pack(pady=5)
    ip_entry = Entry(root, textvariable=ip_var, font=("Arial", 12))
    ip_entry.pack(pady=5)

    # Reverse DNS button
    reverse_button    =    Button(root,    text="Reverse    DNS    Lookup",
command=perform_reverse_dns_lookup, bg="green", fg="white", font=("Arial", 12))
    reverse_button.pack(pady=10)

    # Start the GUI loop
    root.mainloop()

# Function to play GIF frames and fit them to the window
def play_gif():
    gif_image = Image.open("cn.gif")  # Update with your GIF path

    def update_frame(frame_num):
```

```python
        gif_image.seek(frame_num)

        # Resize the frame to fit the label before converting to PhotoImage
        frame = gif_image.copy()  # Copy the current frame
        frame_resized = frame.resize((800, 600), Image.LANCZOS)  # Resize to window size
        photo_frame = ImageTk.PhotoImage(frame_resized)

        gif_label.configure(image=photo_frame)
        gif_label.image = photo_frame  # Keep a reference to avoid garbage collection
        frame_num += 1

        # Loop the GIF frames
        if frame_num >= gif_image.n_frames:
            frame_num = 0
        gif_label.after(100, update_frame, frame_num)

    update_frame(0)

# Function to perform DNS query and update the result label
def perform_dns_query():
    domain = domain_var.get()
    record_type = record_type_var.get().upper()
    result = dns_query(domain, record_type)
    result_label.config(text="\n".join(result))

# Function to perform reverse DNS lookup and update the result label
def perform_reverse_dns_lookup():
    ip_address = ip_var.get()
    result = reverse_dns_lookup(ip_address)
    result_label.config(text=result)

# Function to clear the input fields and result label
def clear_fields():
    domain_var.set("")
    ip_var.set("")
    result_label.config(text="")

# Main execution
if _name_ == "_main_":
    init_gui()
```
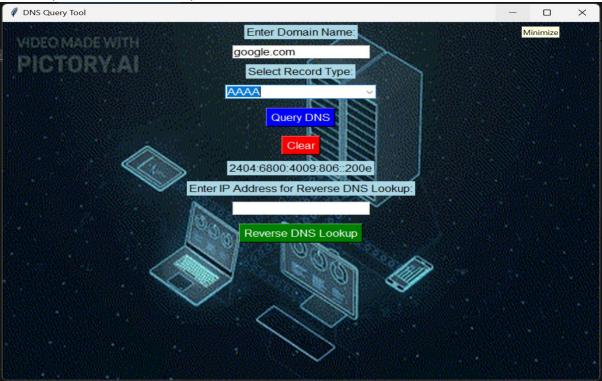
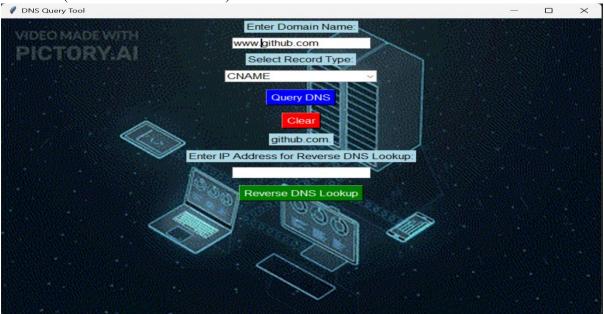# **Output**

DNS Record Lookup:
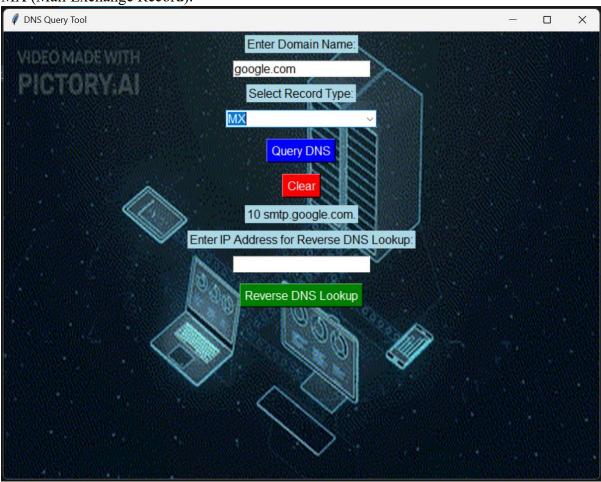
A (Address Record):
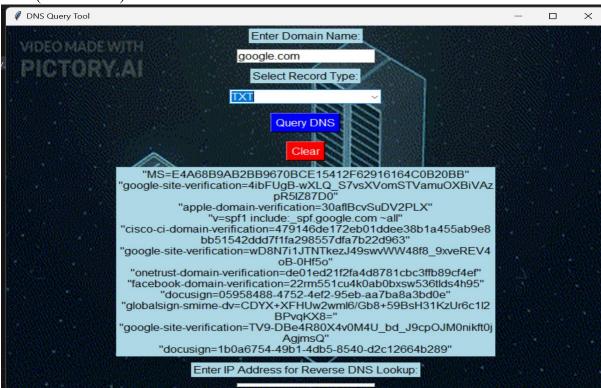

AAAA (IPv6 Address Record):
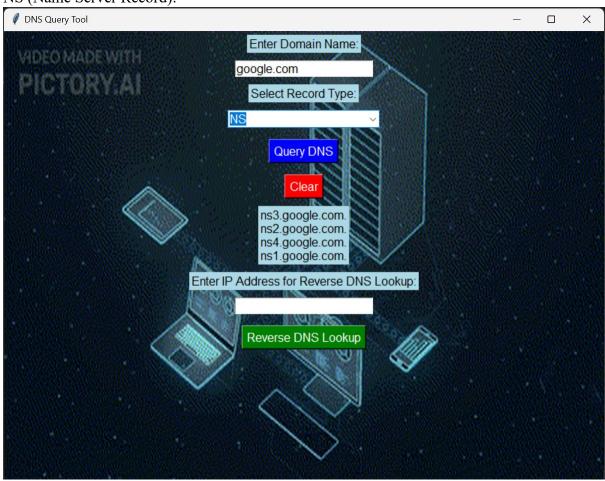
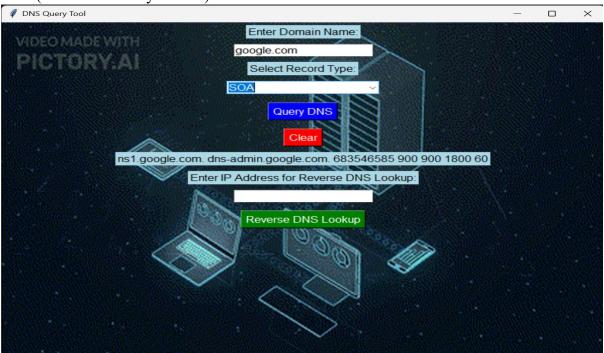CNAME (Canonical Name Record):

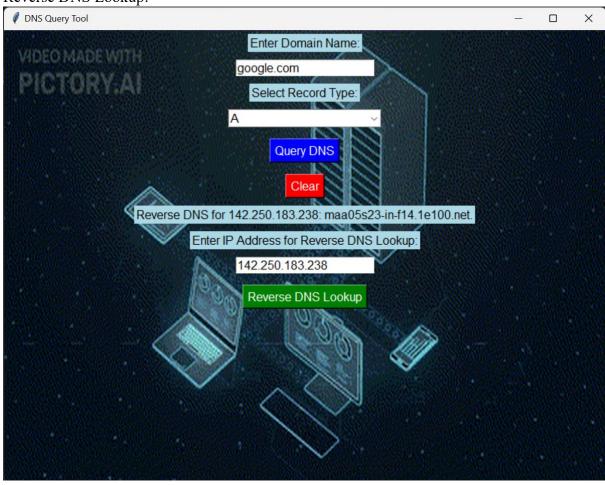MX (Mail Exchange Record):

TXT (Text Record):

NS (Name Server Record):

SOA (Start of Authority Record):

Reverse DNS Lookup:

CONCLUSION

The DNS Query Tool represents a significant advancement in how users interact with domain name systems, providing an intuitive graphical user interface that simplifies complex DNS queries and makes the technology accessible to non-technical users. By consolidating various DNS record lookups—such as A, AAAA, CNAME, MX, TXT, NS, and SOA—into a single application, it enhances usability and efficiency while integrating features like reverse DNS lookup to retrieve relevant information effortlessly. This tool not only benefits everyday users but also serves as a valuable resource for IT professionals and network administrators who require quick access to DNS data. Prioritizing user experience, the tool allows queries to be executed with minimal friction, and the implementation of a GIF background adds a dynamic element that engages users during wait times. As demand for efficient and user-friendly networking tools continues to grow, the DNS Query Tool stands out as a practical solution that addresses the limitations of traditional command-line utilities, ultimately meeting the current needs of its users and setting the foundation for future enhancements that could further expand its capabilities in the evolving digital landscape.

Future Enhancements

Future enhancements for the DNS Query Tool aim to significantly expand its capabilities and improve user experience further. One proposed enhancement is the integration of a caching mechanism to store previously queried DNS records, thereby speeding up subsequent queries for the same information and reducing the load on DNS servers. Additionally, incorporating multi-threading will enable the tool to perform multiple queries simultaneously, enhancing responsiveness, particularly during peak usage times. Another exciting enhancement could involve adding real-time monitoring features that display DNS server status and response times, helping users diagnose connectivity issues more effectively. The tool could also benefit from an expanded graphical interface, introducing more visual elements, such as interactive charts to represent DNS query results, making the information more digestible. Lastly, allowing users to customize their experience with themes or settings tailored to their preferences would further elevate user engagement, making the DNS Query Tool not only a practical utility but also a personalized networking solution that adapts to the needs of diverse user groups.

References

[1]. DNS-O-Matic. (n.d.). DNS Records Explained. Retrieved from https://dnsomatic.com/dns-records-explained

[2]. Internet Engineering Task Force (IETF). (2005). RFC 1035: Domain Names - Implementation and Specification. Retrieved from https://tools.ietf.org/html/rfc1035

[3]. Bind9. (n.d.). BIND 9 Administrator Reference Manual. Retrieved from https://bind9.readthedocs.io/en/latest/

[4]. Python Software Foundation. (n.d.). Python DNS library. Retrieved from https://pypi.org/project/dnspython/

[5]. GeeksforGeeks. (2021). Understanding DNS Queries in Python. Retrieved from https://www.geeksforgeeks.org/understanding-dns-queries-in-python/

[6]. Real Python. (2021). Building a DNS Client with Python. Retrieved from https://realpython.com/python-dns-client/

[7]. Tutorialspoint. (n.d.). Python - GUI Programming with Tkinter. Retrieved from https://www.tutorialspoint.com/python/python_gui_programming.htm