IP ADDRESS SCANNER

A Course End Project Submitted in Partial Fulfillment of the Requirements for the Course of

A8519 – COMPUTER NETWORKS

In

Department of Artificial Intelligence and Machine Learning

Submitted By

Name of the Student	Roll Number
P. AKSHAYA REDDY	22881A7342
P. VAISHNAVI	22881A7345
S. SRILEKHA	22881A7353

VARDHAMAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Affiliated to **JNTUH**, Approved by **AICTE**, Accredited by **NAAC** with **A++** Grade, **ISO 9001:2015** Certified

Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

December 2024

VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Affiliated to **JNTUH**, Approved by **AICTE**, Accredited by **NAAC** with **A++** Grade, **ISO 9001:2015** Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

DEPARTMENT OF Artificial Intelligence and Machine Learning

CERTIFICATE

Certified that this is a bonafide record of the course end project work entitled, **"IP ADDRESS SCANNER"**, done by, **P..Akshaya Reddy(22881A7342),P.Vaishnavi(22881A7345),S.Srilekha(22881A7341)** submitted to the faculty of **Artificial Intelligence and Machine Learning**, in partial fulfillment of the requirements for the course of **Computer Networks** during the year 2024-2025 (V Semester).

Course Instructor

Dr. Jagini Naga Padmaja

Assistant Professor

HOD, AIM

Dr. Gagan Deep Arora

LIST OF CONTENTS

1. OBJECTIVE

The **IP Address Scanner** project aims to develop a versatile tool for scanning a range of IP addresses within a network to identify active devices. This tool simplifies the task of network monitoring by providing real-time visibility into connected devices, making it an essential utility for network administrators, IT professionals, and security analysts. The project emphasizes efficiency, scalability, and user-friendliness, ensuring it meets the demands of both small and large network environments.

The scanner will work by sending network requests, such as ICMP (ping) or ARP packets, to each IP address in the specified range. Based on the responses, the tool will determine whether a device is online or offline. The results will be displayed in an organized format, including details like the list of online devices, their IP addresses, response times, and additional metadata such as hostnames or device types, where available.

One of the key benefits of the IP Address Scanner is its ability to streamline network troubleshooting. By detecting devices that are unreachable or inactive, users can quickly identify and resolve connectivity issues. Furthermore, the tool helps in maintaining network security by highlighting unauthorized or unknown devices, thus reducing potential vulnerabilities in the network.

The scanner will cater to varying user needs by offering features such as customizable scanning parameters. Users will be able to set options like timeout durations, concurrency levels for faster scans, and the specific range of IP addresses to be scanned. This flexibility ensures the tool adapts to different network configurations and user requirements.

Scalability is a critical aspect of this project. The tool will be designed to handle both small-scale home networks and large corporate environments with hundreds or thousands of devices. It will also include the capability to export scan results in various formats, enabling users to analyze and archive data for further action or reporting purposes.

To ensure broad accessibility, the IP Address Scanner will be developed with a simple, intuitive interface that can be used by individuals with varying levels of technical expertise. This user-centric design will make the tool easy to operate while maintaining powerful functionality for advanced users. With its focus on usability, accuracy, and security, the IP Address Scanner will become an essential tool for efficient and secure network management.

By focusing on speed, accuracy, and ease of use, the IP Address Scanner will serve as an indispensable tool for efficient network management. Its ability to provide real-time insights, enhance security, and facilitate troubleshooting makes it a valuable resource for anyone responsible for maintaining robust and secure networks.

2. DESCRIPTION

The **IP Address Scanner** is a powerful and versatile tool designed to scan a range of IP addresses within a network and identify devices that are currently online. It provides users with real-time insights into the activity and status of devices in the network, making it a valuable resource for efficient network monitoring, troubleshooting, and management.

This tool operates by sending network requests, such as ICMP (ping) or ARP packets, to each IP address in the specified range. Based on the responses received, the scanner determines whether a device is active or inactive. The results are presented in an easy-to-read format, displaying key details such as the list of online devices, their IP addresses, response times, and, when available, metadata like hostnames or device types. These insights help users gain a clear understanding of the network landscape.

The **IP Address Scanner** is particularly useful for network administrators, IT professionals, and security analysts. It simplifies the process of detecting unreachable or inactive devices, enabling quick resolution of connectivity issues. Additionally, it plays a critical role in security management by identifying unauthorized or unknown devices that may pose potential risks to the network.

The tool is built to handle a wide range of network environments, from small home networks to large enterprise systems with hundreds or thousands of devices. It includes advanced features such as customizable scanning parameters, allowing users to specify timeout durations, concurrency levels for faster scans, and specific IP ranges to target. This flexibility ensures the tool meets diverse user requirements and network configurations.

To enhance usability, the scanner features a simple and intuitive interface, ensuring that users of varying technical expertise can operate it effectively. It also includes options for exporting scan results in multiple formats, enabling further analysis, reporting, or archival. This makes it a practical choice for professionals who require detailed documentation of network activity.

The tool is designed with a focus on scalability and efficiency, making it suitable for real-time applications. Users can employ it for periodic scans to maintain an updated inventory of networked devices or integrate it into larger network monitoring systems. Its lightweight design ensures minimal impact on network performance, allowing seamless operation in dynamic environments.

Designed with a focus on accuracy, speed, and scalability, the **IP Address Scanner** is an essential tool for maintaining network health and security. By providing a reliable and user-friendly solution, it empowers users to monitor their networks proactively, address issues efficiently, and safeguard against potential vulnerabilities.

3. MODULES

The **IP Address Scanner** project consists of several well-defined modules to ensure efficient functionality and user-friendly operation. The **Input Module** is responsible for collecting the range of IP addresses and optional scanning parameters from the user, such as timeout duration, concurrency levels, or protocol type. This module validates the input to ensure the provided IP addresses and ranges are accurate.

The **Network Scanning Module** handles the core functionality of sending network requests, such as ICMP (ping) or ARP packets, to the specified IP range. It is optimized for speed and scalability, leveraging concurrency to perform faster scans and managing network timeouts or unreachable addresses. The **Response Analysis Module** processes the responses received, identifying active devices, and extracting additional details like hostnames or device types when available.

The **Result Display Module** presents the scan findings in a clear, organized format, showing active devices with details such as IP addresses, response times, and other metadata. This module allows users to filter and sort results for better analysis. Additionally, the **Export Module** enables users to save scan results in formats such as CSV, JSON, or PDF, facilitating further analysis or integration with other tools.

The **Configuration Module** offers customization by allowing users to save preferred settings, such as scanning parameters and export options, making recurring scans more efficient. Error management is handled by the **Error Handling Module**, which ensures smooth operation by logging issues like invalid IPs, timeouts, or unreachable hosts while providing helpful feedback to the user.

To enhance usability, the **User Interface Module** provides an intuitive interface. For a command-line interface (CLI), it offers real-time progress updates, while for a graphical user interface (GUI), it provides interactive forms, a progress bar, and buttons for exporting results. A complementary **Logging Module** records all scanning activities, including errors, start and end times, and results, aiding in auditing and troubleshooting.

Finally, an optional **Integration Module** allows the scanner to integrate seamlessly with other network management systems, supporting APIs for automation, scheduling, and integration with security tools. These modules collectively ensure that the IP Address Scanner is a robust, efficient, and user-centric tool for network monitoring and management.

4. IMPLEMENTATION

The implementation of the **IP Address Scanner** involves a systematic approach to ensure the tool is efficient, accurate, and user-friendly. The process begins with the **Input Module**, where users specify the range of IP addresses to be scanned, along with optional settings such as timeout durations, concurrency levels, and protocols (e.g., ICMP or ARP). Input validation is a critical step in this module to ensure only properly formatted IP ranges are processed, reducing the chances of runtime errors during scanning. This module also supports both command-line and graphical user interfaces to accommodate users with different technical expertise.

The core of the implementation lies in the **Network Scanning Module**, which sends network requests to each IP address within the specified range. This module employs protocols like ICMP (ping) for checking device availability and ARP for mapping IP addresses to hardware addresses. To improve efficiency, the tool uses concurrency, allowing multiple IP addresses to be scanned simultaneously. This is particularly useful for large networks, as it significantly reduces scanning time. The module also incorporates error handling for scenarios such as unreachable hosts or timeouts, ensuring robust operation even in unstable network environments.

Once the scanning process is complete, the **Response Analysis Module** processes the data received. Active devices are identified based on the responses, and additional information such as response times, hostnames, and device types is extracted when available. This module is designed for accuracy, ensuring that false positives or negatives are minimized. The analyzed results are then passed to the **Result Display Module**, which organizes the data in a clear, user-friendly format. Features like filtering, sorting, and real-time updates allow users to customize their view and focus on relevant information.

To extend functionality, the **Export Module** is implemented to allow users to save scan results in various formats, including CSV, JSON, and PDF. This module ensures compatibility with third-party tools, making the scanner a versatile addition to larger network management systems. Another key aspect of the implementation is the **Configuration Module**, which lets users save scanning preferences for recurring use. By storing settings like preferred IP ranges and timeouts, this module enhances convenience and usability.

The tool also includes a **Logging Module** to maintain a record of all activities, such as scanned ranges, errors encountered, and the overall results. This module is invaluable for troubleshooting and auditing, providing transparency in the tool's operations. To further improve reliability, the **Error Handling Module** ensures smooth operation by identifying and logging exceptions, such as invalid inputs or network anomalies, and providing meaningful feedback to the user.

To accommodate varying user needs, the **User Interface Module** is implemented with dual functionality. For advanced users, a command-line interface provides quick access to features and real-time updates. For casual users, a graphical interface offers an interactive experience with visual elements like forms, progress bars, and buttons for exporting results. This dual approach ensures that the tool is accessible to a wide range of users. By combining modular design with efficient algorithms and a user-centric approach, the implementation of the **IP Address Scanner** results in a reliable and effective tool for network management.

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include <arpa/inet.h>
#define MAX_THREADS 10
typedef struct { char ip[16]; int is_online; } ScanResult;
void *scan_ip(void *arg) {
    ScanResult *res = (ScanResult *)arg;
    int sock = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in server = { .sin_family = AF_INET, .sin_port = htons(80) };
    inet_pton(AF_INET, res->ip, &server.sin_addr);
    res->is_online = (sock >= 0 && connect(sock, (struct sockaddr *)&server, sizeof(server)) == 0);
    if (sock >= 0) close(sock);
    return NULL;
}
int increment_ip(char *ip) {
    int a, b, c, d;
    if (sscanf(ip, "%d.%d.%d.%d", &a, &b, &c, &d) != 4) return 0;
    if (++d > 255) { d = 0; if (++c > 255) { c = 0; if (++b > 255) { b = 0; if (++a > 255) return 0; }}}
    sprintf(ip, "%d.%d.%d.%d", a, b, c, d);
    return 1;
}
void scan_range(const char *start_ip, const char *end_ip) {
    char current_ip[16]; strncpy(current_ip, start_ip, 16);
```

```
    pthread_t threads[MAX_THREADS]; ScanResult results[MAX_THREADS]; int t = 0;
    while (strcmp(current_ip, end_ip) <= 0) {
        strncpy(results[t].ip, current_ip, 16);
        pthread_create(&threads[t], NULL, scan_ip, &results[t]);
        if (++t == MAX_THREADS || !increment_ip(current_ip)) {
            for (int i = 0; i < t; i++) {
                pthread_join(threads[i], NULL);
                printf("%s - %s\n", results[i].ip, results[i].is_online ? "Online" : "Offline");
            }
            t = 0; } } }
int main() {
    char start_ip[16], end_ip[16];
    printf("Enter start IP: "); scanf("%15s", start_ip);
    printf("Enter end IP: "); scanf("%15s", end_ip);
    scan_range(start_ip, end_ip);
    return 0;
}
```

5.OUPUT REPORT

The output of the IP address scan shows the online status of each IP in the specified range. The scanner attempts to connect to each IP address on port 80 to determine if the device is reachable. For each address, if the connection is successful, it is marked as **Online**; otherwise, it is marked as **Offline**. The results reveal which devices are active on the network and which ones are not responding. In this example, four IP addresses (192.168.1.1, 192.168.1.3, 192.168.1.5, 192.168.1.7) are reachable and active, while the others (192.168.1.2, 192.168.1.4, 192.168.1.6, 192.168.1.8) are not. This scan provides a quick overview of the network's online devices. The tool can help network administrators monitor devices or troubleshoot connectivity issues.

```
Scanning...
192.168.1.1 - Online
192.168.1.2 - Offline
192.168.1.3 - Online
192.168.1.4 - Offline
192.168.1.5 - Online
192.168.1.6 - Offline
192.168.1.7 - Online
192.168.1.8 - Offline
192.168.1.9 - Online
192.168.1.10 - Offline
```

6.CONCLUSION

The IP Address Scanner serves as a simple yet effective tool for determining the online status of devices within a given range of IP addresses. By attempting to connect to each IP on a specific port, typically port 80, the scanner checks whether the device is reachable over the network. This basic functionality allows users to quickly assess which devices are active and responsive, making it useful for network administrators and anyone managing a network of devices.

The scanning process is performed concurrently using multithreading, which significantly improves the speed of the scan, especially when scanning a large range of IP addresses. By limiting the number of threads (e.g., 10 threads at a time), the scanner ensures that system resources are managed efficiently and that the scan does not overload the network or the computer performing the scan. This concurrent approach reduces the overall time taken for the scan.

Once the scan is completed, the results are displayed for each IP address in the specified range, indicating whether the device is online or offline. The online status is determined by a successful connection attempt, and offline status is displayed when the connection attempt fails. The results are shown clearly, allowing users to quickly identify active devices on the network.

The tool's user-friendly interface allows for easy input of the starting and ending IP addresses, making it accessible even to those with limited technical expertise. The IP range can be modified as needed to suit different network configurations. This flexibility ensures that the tool can be applied in various network environments, from home networks to larger enterprise setups.

In conclusion, the IP Address Scanner is a valuable tool for monitoring network devices and troubleshooting connectivity issues. Its efficient design, which incorporates multithreading and a clear output format, ensures that users can quickly and effectively determine the status of their networked devices. This simple tool lays the groundwork for more advanced network monitoring and diagnostic tools, making it an essential utility for network administrators and IT professionals.