BASAVARAJESWARI GROUP OF INSTITUTIONS

Ballari Institute of Technology & Management

AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA,
BELAGAVI 590018

INTERNSHIP

Report On

NETWORK LATENCY TESTER

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Engineering

ARTIFICIAL INTTELIGENCE AND MACHINE LEARNING

Submitted by

S VEERENDRA SWAMY ,SATHEESH,SUBHOOD,VALMIKI UDAY KIRAN

3BR22AI141,3BR22AI148,3BR22AI157,3BR22AI181

Internship Carried Out
By

EZ TRAININGS & TECHNOLOGIES PVT.LTD
HYDERABAD

Internal Guide	External Guide
Rachel Evangeline Chirstian	Raj Vardhan
Asst.prof,AIML	Technical Trainer
Hosmani Manikeshwari	
Asst. prof,AIML	

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT
NACC Accredited Institution*
(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to
Visvesvaraya Technological University, Belagavi)
"Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road,
Allipur, Ballar1-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197.

2023-2024

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

Autonomous institute under VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA,

BELAGAVI 590018

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to
Visvesvaraya Technological University, Belagavi)

"JnanaGangotri"Campus,No.873/2,Ballari-HospetRoad,Allipur,
Ballar1-583 104 (Karnataka)(India)
Ph: 08392 – 237100 / 237190, Fax: 08392 –237197

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

This is to certify that the Internship entitled **" NETWORK LATENCY TESTER "** has been successfully completed by S VEERENDRA SWAMY ,SATHEESH,SUBHOOD,VALMIKI UDAY KIRAN bearing USN **3BR22AI141,3BR22AI148,3BR22AI157,3BR22AI181** a bonafide student of Ballari Institute of Technology and Management, Ballari. For the partial fulfillment of the requirementsfor the **Bachelor's Degree in Artificial Intelligence and Machine Learning** of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belagavi during the academic year 2023-2024.

Signature of Internship
Co-ordinator

Rachel Evangeline Chirstian
Asst.prof,AIML
Hosmani Manikeshwari
Asst. prof,AIML

Signature of HOD

DR. B M VIDYAVATHI
Prof. and HOD(AIML)

DECLARATION

I, S VEERENDRA SWAMY ,SATHEESH,SUBHOOD,VALMIKI UDAY KIRAN **,** second year student of Artificial Intelligence and Machine Learning, Ballari Institute of Technology, Ballari, declare that Internship entitled **NETWORK LATENCY TESTER (POC)** is a part of Internship Training successfully carried out by **EZ TECHNOLOGIES & TRAININGS PVT.LTD ,Hyderabad** at "**BITM,BALLARI".** This report is submitted in partial fulfillment of the requirements for the award of the degree, Bachelor of Engineering in Artificial Intelligence and Machine Learning of the Visvesvaraya Technological University, Belagavi.

Date : 4/04/1 **Signature of the Student**

Place :

ACKNOWLEDGEMENT

The satisfactions that a company the successful completion of my internship on **“NETWORK LATENCY TESTER ”** would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is my privilege to express my gratitude and respect to all those who inspired me in the completion of my internship.

I am grateful to our respective coordinator **“Rachel Evangeline Christian (Asst.prof,AIML) , Hosmani Manikeshwari (Asst.prof,AIML)”** for his noble gesture, support co-ordination and valuable suggestions given to me in the completion of Internship.

I also thank **Dr. B M Vidyavathi,** H.O.D. Department of **Artificial intelligence and machine learning** for extending all his valuable support and encouragement.

Index

Chapter no	Topic	Page No
1	Company Profile	1-2
2	Abstract	2-3
3	Introduction to project	3-5
4	Daily Work Progress	5-6
5	Module Description	7-8
6	Algorithm	9-12
7	Outputs	13-16
8	Conclusion	16-17
9	Reference	17-18

CHAPTER-1

COMPANY PROFILE

Company Name: EZ Trainings and Technologies Pvt. Ltd.

Introduction:

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

Mission:

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.

Services:

College Trainings:

- Tailored training programs designed to enhance the employability of students.
- Industry-aligned curriculum covering technical and soft skills.
- Placement assistance and career guidance.

Development Projects:

- End-to-end development services, from ideation to execution.
- Expertise in diverse technologies and frameworks.
- Custom solutions to meet specific business needs.

Locations: Hyderabad | Delhi NCR

At EZ Trainings and Technologies Pvt. Ltd., we believe in transforming potential into excellence

CHAPTER-2

ABSTRACT

The network latency tester is a tool designed to measure and analyze network latency, providing valuable insights into the performance and reliability of network connections. By sending test packets between two endpoints and measuring the round-trip time, the tool assesses latency, helping network administrators identify potential bottlenecks and optimize network performance. This abstract provides an overview of the network latency tester, its functionality, and its importance in ensuring efficient network operations.

Tool allows users to send test packets between two endpoints, measuring the round-trip time and providing valuable insights into network performance. By analyzing latency, the Network Latency Tester helps network administrators identify bottlenecks and optimize network efficiency. This abstract outlines the functionality and significance of the Network Latency Tester as a Python project for assessing and enhancing network performance.

The Network Latency Tester is a Python-based tool designed to measure and analyze network latency effectively. By sending ICMP echo request packets between two endpoints, it calculates the round-trip time, providing valuable insights into network performance. This abstract outlines the functionality and significance of the Network Latency Tester in assessing and optimizing network efficiency. With features such as customizable packet size and frequency, support for both IPv4 and IPv6 networks, and detailed reporting capabilities, the Network Latency Tester is an invaluable tool for network administrators and developers. Its user-friendly interface and powerful functionality make it an essential component for ensuring optimal network performance and reliability

CHAPTER-3

INTRODUCTION OF THE PROJECT

Network latency is crucial. The Network Latency Tester Proof of Concept (POC) is a Python-based project developed to facilitate this process. One of its key features is the implementation of CRUD (Create, Read, Update, Delete) operations for managing test results. This functionality allows network administrators and developers to effectively organize and manipulate latency test data. With the ability to create, read, update, and delete test results, users can easily track network performance over time and make informed decisions to improve efficiency and reliability. This section will detail the implementation of CRUD operations within the Network Latency Tester POC, demonstrating its importance in network latency management.

Key Features

1. CRUD Operations for Test Results:

- Create, Read, Update, and Delete operations for managing test results.

2. perform_latency_test(network_details):

- This function tests network latency between two points by sending ICMP echo request packets and measuring the round-trip time.

3. record_test_results(result_id):

- Records and displays the latency test results.

Project Objectives:

- To provide a prototype for a network latency testing tool using Python.
- To demonstrate the functionality of testing network latency between two points.
- To create a simple interface for recording and displaying test results.

Target Audience:

- Network administrators
- Developers

Technologies Used:

- Python
- ICMP (Internet Control Message Protocol)

In today's digital landscape, network performance is a critical aspect of ensuring smooth operations. The Network Latency Tester Proof of Concept (POC) is a Python-based project developed to address this need effectively. With its primary functionalities including CRUD (Create, Read, Update, Delete) operations for managing test results, performing network latency tests between two points, and recording and displaying the latency test results, this project aims to provide network administrators and developers with a comprehensive tool for optimizing network performance. This introduction provides an overview of the Network Latency Tester POC, highlighting its significance in assessing and improving network latency effectively.

CHAPTER-5

MODULE DESCRIPTION

This Python module provides a Network latency tester. Here's a detailed description of its components:

Import socket: The socket module in Python is a core module used for low-level networking operations. In the Latency Tester module, the socket module is imported to enable network communication functionalities required for measuring network latency. Here's how the socket module is utilized in the Latency Tester module

Import os : In the Latency Tester module, the os module is imported to facilitate file management operations. This module is crucial for creating, deleting, and checking the existence of files in the local file system. Thcreate_data_file() method utilizes the open() function from the os module to create a data file named "veer.txt". This file is used to store the host and port information required for latency tests. Similarly, the create_result_file() method also utilizes the open() function to create a result file named "file1.txt". This file is intended to store the test results along with their corresponding IDs.On the other hand, the delete_data_file() and delete_result_file() methods use the os.path.exists() function to check if the data and result files exist. If the files exist, the os.remove() function is used to delete them. This ensures that the data and result files are properly managed within the module.

Import time: In the Latency Tester module, the os module is imported and used for file management operations. The os module provides functions for interacting with the operating system, including file operations such as file creation, deletion, and checking file existence.The create_data_file() method uses the open() function from the os module to create a data file named "veer.txt". This file is created to store the host and port information required for latency tests. Inside the with statement, the open() function is called with the file name "veer.txt" and the mode 'w' to open the file in write mode. This method writes the host and port information to the file using the write() method

The `LatencyTester` module provides a comprehensive solution for measuring and managing network latency. Below is a brief description of each function within the module:

1. perform_latency_test(host, port)

- This function measures network latency between two points by establishing a TCP connection to the specified host and port. It calculates the round-trip time in milliseconds and returns the latency value.

2. `record_test_result(result_id, latency)

- Records the latency test result along with a unique result ID in a dictionary called `results`. It prints a confirmation message after recording the test result.

3. display_test_results()`

- Displays all recorded test results stored in the `results` dictionary.

4. create_data_file(host, port)`

- Creates a data file named "veer.txt" to store the host and port information for latency tests.

5. update_data_file(host, port)`

- Updates the data file with new host and port information.

6. `read_data_file()`

- Reads the data file to retrieve host and port information for latency tests.

7. `delete_data_file()`

- Deletes the data file.

8. create_result_file(result_id, latency)

- Creates a result file named "file1.txt" to store test results along with their corresponding IDs.

9. update_result_file(result_id, latency)`

- Updates the result file with new test results.

10. read_result_file()`

- Reads the result file to retrieve test results.

11. delete_result_file()`

- Deletes the result file.

12. menu()`:

- Displays a menu of options for the user to interact with the tool.

- Allows the user to perform latency tests, record and display test results, manage data and result files, and exit the program.

CHAPTER-6

ALGORITHM

Algorithm: Network Latency Tester

1. Import required modules:

```
  - import socket
  - import time
  - import os
```

2. Define the LatencyTester class:

```
    Class LatencyTester:
  2.1 Initialize the class:
      - Define the __init__() method:
        - Initialize an empty dictionary to store test results.
 2.2 Define a method to perform latency test:
    Method perform_latency_test(host, port):
        - Try the following:
          - Create a socket object using socket.socket() method.
          - Set a timeout for the connection attempt using settimeout() method.
          - Get the start time using time.time() method.
          - Connect to the server using connect() method.
          - Get the end time using time.time() method.
          - Calculate the latency (in milliseconds).
          - Return the latency.
        - Except socket.timeout:
          - Return 'inf' if the connection attempt timed out.
        - Except Exception as e:
          - Print the error message.
          - Return None.
        - Finally:
          - Close the socket.\
```

2.3 Define a method to record test result:

```
Method record_test_result(result_id, latency):
  - Record the latency test result in the results dictionary.
  - Print a confirmation message.
```

2.4 Define a method to display test results:

```
Method display_test_results():
  - If no test results are recorded, print a message.
  - Else, print all recorded test results.
```

2.5 Define a method to create a data file:

```
Method create_data_file(host, port):
  - Create a data file named "veer.txt".
  - Write host and port information to the file.
```

2.6 Define a method to update the data file:

```
Method update_data_file(host, port):
  - Call create_data_file() method to update the data file with new host and port information.
```

2.7 Define a method to read data from the data file:

```
Method read_data_file():
  - Try the following:
    - Read host and port information from the data file.
    - Return the host and port.
  - Except FileNotFoundError:
    - Print "Data file not found".
    - Return None, None.
  - Except IndexError:
    - Print "Data file is corrupted".
    - Return None, None.
```

2.8 Define a method to delete the data file:

```
Method delete_data_file():
  - If the data file exists, delete it.
  - Print a message indicating whether the file was deleted or not.
```

2.9 Define a method to create a result file:

```
Method create_result_file(result_id, latency):
  - Create a result file named "file1.txt".
  - Write result ID and latency to the file.
```

2.10 Define a method to update the result file:

```
Method update_result_file(result_id, latency):
  - Call create_result_file() method to update the result file with new result.
```

2.11 Define a method to read data from the result file:

```
Method read_result_file():
  - Try the following:
    - Read result ID and latency from the result file.
```

```
      - Return the result ID and latency.
    - Except FileNotFoundError:
      - Print "Result file not found".
      - Return None, None.
    - Except IndexError:
      - Print "Result file is corrupted".
      - Return None, None.
```

2.12 Define a method to delete the result file:

```
  Method delete_result_file():
    - If the result file exists, delete it.
    - Print a message indicating whether the file was deleted or not.
```

2.13 Define a method to display the menu:

```
  Method menu():
    - Display menu options to the user.
    - Take user input.
    - Perform actions based on user input.
```

3. Example usage:

- Instantiate the LatencyTester class.
- Call the menu() method to display the menu and perform actions.

4. End

CHAPTER-7

OUTPUTS

```
Menu:
1. Perform Latency Test
2. Display Test Results
3. Read Data File
4. Update Data File
5. Read Result File
6. Update Result File
7. Delete Data File
8. Delete Result File
0. Exit
Enter your choice:  1
Enter result ID:  1
Test result recorded - ID: 1, Latency: 99.98226165771484 ms

Menu:
1. Perform Latency Test
2. Display Test Results
3. Read Data File
4. Update Data File
5. Read Result File
6. Update Result File
7. Delete Data File
8. Delete Result File
0. Exit
Enter your choice:  2
Recorded Test Results:
ID: 1, Latency: 99.98226165771484 ms
```

```
Menu:
1. Perform Latency Test

2. Display Test Results
3. Read Data File
4. Update Data File
5. Read Result File
6. Update Result File
7. Delete Data File
8. Delete Result File
0. Exit
Enter your choice:  3
Data Host: www.google.com, Data Port: 80

Menu:
1. Perform Latency Test
2. Display Test Results
3. Read Data File
4. Update Data File
5. Read Result File
6. Update Result File
7. Delete Data File
8. Delete Result File
0. Exit
Enter your choice:  4
Enter new IP address:  www.youtube.com
Enter new port number:  443

Menu:
1. Perform Latency Test
```

```
2. Display Test Results

3. Read Data File

4. Update Data File

5. Read Result File

6. Update Result File

7. Delete Data File

8. Delete Result File

0. Exit

Enter your choice:  5

Result ID: 1, Result Latency: 99.98226165771484 ms


Menu:

1. Perform Latency Test

2. Display Test Results

3. Read Data File

4. Update Data File

5. Read Result File

6. Update Result File

7. Delete Data File

8. Delete Result File

0. Exit

Enter your choice:  6

Enter result ID:  1

Enter result latency:  99.98


Menu:

1. Perform Latency Test

2. Display Test Results

3. Read Data File

4. Update Data File

5. Read Result File
```

```
6. Update Result File
7. Delete Data File
8. Delete Result File
0. Exit
Enter your choice:  8
Result file deleted.

Menu:
1. Perform Latency Test
2. Display Test Results
3. Read Data File
4. Update Data File
5. Read Result File
6. Update Result File
7. Delete Data File
8. Delete Result File
0. Exit
Enter your choice:  0
```

CHAPTER-8

CONCLUSION

The Network Latency Tester POC is a simple yet effective Python project for testing network latency between two points. This Proof of Concept (POC) demonstrates the basic functionality of measuring network latency and storing the test results. Throughout the project, we have implemented various functionalities including performing latency tests, recording test results, displaying test results, and managing data and result files.

The use of sockets, time module, and file management operations provided by the os module ensures the smooth execution of network latency tests and accurate storage of test results. Although this POC provides basic functionalities, it can be further expanded and enhanced to include more advanced features such as graphical user interface (GUI), multi-threading for concurrent tests, and support for different types of network protocols.

Overall, this Network Latency Tester POC serves as a foundation for building a more comprehensive and robust network latency testing tool.

CHAPTER-9

REFERENCES

- https://chat.openai.com/c/fd7b734f-d486-4848-9fe2-1e3b8045facc
- https://www.netskope.com/blog/how-to-measure-network-latency-the-5-best-tool
- https://www.dnsstuff.com/network-latency