VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 018.

A PROJECT REPORT

on

"NETWORK MONITOR"

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

Bachelor of Engineering

in

Computer Science and Engineering

by

PAVITHRA K	**:**	**1VI17CS071**
RAJASHREE	**:**	**1VI17CS114**

Under the supervision of

Mr. Noor Basha
Asst. Professor Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VEMANA INSTITUTE OF TECHNOLOGY

BENGALURU – 560 034

2020 - 21

Karnataka ReddyJana Sangha®

VEMANA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

Koramangala, Bengaluru-34.

Department of Computer Science and Engineering

Certificate

Certified that the project work entitled "**NETWORK MONITOR**" carried out jointly by **PAVITHRA K(1VI17CS071), RAJASHREE (1VI17CS114)** are bonafide students **of Vemana Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2020-21. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said degree.

Supervisor	**HOD**	**Principal**
Mr. Noor Basha	Dr. M. Ramakrishna	Dr. Vijayasimha Reddy. B. G.

Submitted for the university examination (viva-voce) held on …….……

External Viva

Internal Examiner **External Examiner**

1.

2.

DECLARATION BY THE CANDIDATES

We the undersigned solemnly declare that the project report "NETWORK MONITOR" is based on our own work carried out during the course of our study under the supervision of 'Mr. Noor Basha'.

We assert the statements made and conclusions drawn are an outcome of our project work. We further certify that,

a. The work contained in the report is original and has been done by us under the general supervision of my supervisor.

b. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

c. We have followed the guidelines provided by the university in writing the report.

d. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and their details are provided in the references.

Date:

Place:

Project Associates:

PAVITHRA K 1VI17CS071

RAJASHREE 1VI17CS114

ACKNOWLEDGEMENT

First we would like to thank our parents for their kind help, encouragement and moral support.

We thank **Dr. Vijayasimha Reddy. B. G,** Principal, Vemana Institute of Technology, Bengaluru for providing the necessary support.

We would like to place on record our regards to **Dr. M. Ramakrishna,** Professor and Head, Department of Computer Science and Engineering for his continued support.

We would like to thank our project coordinators **Mrs. Mary Vidya John,** Assistant Professor and **Mrs. J Brundha Elci,** Assistant Professor, Dept. of CSE for their support and coordination.

We would like to thank our project guide **Mr. Noor Basha,** Designation, Dept. of CSE for his continuous support, valuable guidance and supervision towards successful completion of the project work.

We also thank all the Teaching and Non-teaching staff of Computer Science and Engineering Department, who have helped us to complete the project in time.

Project Associates:

PAVITHRA K (1VI17CS071)
RAJASHREE (1VI17CS114)

CONTENTS

ABSTRACT

Network monitoring has critical significance in latest computer networks due to the rise of new complicated threats every day. Advances in networking hardware technologies enabled administrators to deal networks with large endpoint systems, higher speeds, wireless connectivity, virtual devices, IoT devices, Cloud environments and so on. Network monitoring is a set of mechanisms that allows network administrators to know instantaneous state and long-term trends of a complex computer network. In this project, an effective and automated network monitoring system is presented that is different from existing system in three major aspects – support to wide range of devices, large number of useful features of different categories and overlaying of different protocols to obtain an accurate output. The proposed system supports different types of devices such as routers, switches, access points, endpoint systems, IoT devices, virtual devices, cloud environments and so on. It also has helpful features such as customized alerts configuration, aesthetic presentation of the network, reliability metrics, performance metrics, SSH/Telnet access to the devices and much more. Different networking protocols used in over lay to obtain requisite information from the devices are ICMP, SMTP, LLDP, ARP, and so on.

Keywords: Network Monitoring, IoT, Cloud Computing, ICMP, SMTP, LLDP, ARP, Network Administrators, Computer Networks, Internet.

LIST OF ABBREVIATIONS

Abbreviation	Description
ARP	Address Resolution Protocol
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
GB	Gigabyte
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
ICMP	Internet Control Message Protocol
ID	Identify Devices
IP	Internet Protocol
LLDP	Link Layer Discovery Protocol
LTS	Long Term Support
MAC	Media Access Control Address
OS	Operating System
PDU	Protocol Data Unit
RAM	Random Access Memory
RARP	Reverse Address Resolution Protocol
REGEX	Regular Expression

SMS	Short Message Service
SNMP	Simple Network Management protocol
SSH	Secure Shell
SSLVPN	Secure Sockets Layer Virtual Private Network
SYSLOG	System Logging Protocol
TCP	Transmission Control
TTL	Transistor-Transistor Logic
TV	Television
UML	Unified Modeling Language
VOIP	Voice Over Internet Protocol
VPN	Virtual Private Network

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

A computer network is a group of network nodes that use a set of common communication protocols over digital interconnections for the purpose of sharing resources located on or provided by the network nodes. A network node is either a redistribution point or a communication endpoint such as modem, hub, switch, bridge, routers, network file systems, personal computers and so on. Network monitoring system is a scripted tool that administers a computer network for slow or failing components and that notifies the configured users in case of outages, latencies, upgrades, and other events.

With the advent of internet, network monitors can now perform variating services such as fault analysis, performance management, provisioning of networks and maintain quality of service. Internet technologies have become very important in every person's day to day life and proactive monitoring of interconnected devices has become vital in the internet service provision business. The onset of ecommerce has reduced the prices of IoT to great extent and a typical home in suburban consists of Layer 3 router, a Wi-Fi access point, internet connected TV and remotely controlled lightbulbs. As a result, security and monitoring have become a critical concern for every person with internet connectivity.

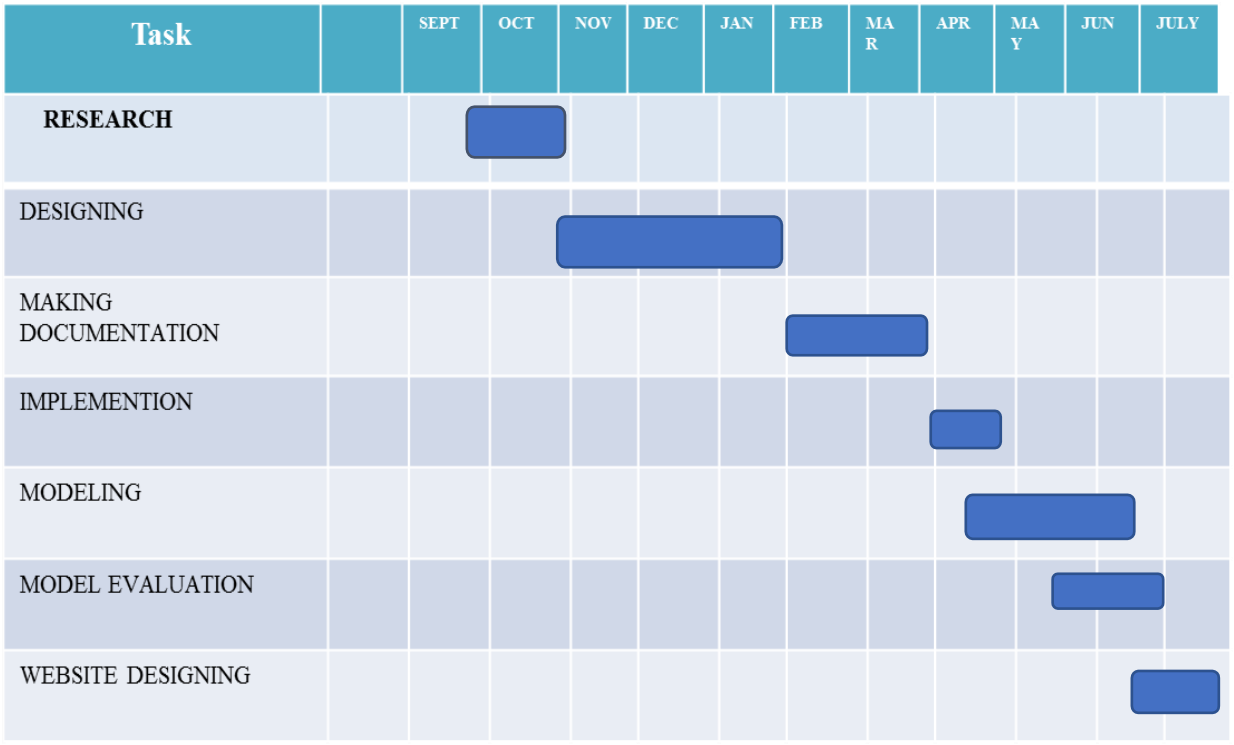
The latest generation of network monitoring systems are designed to support very specific applications. Most of them in general, are devised to monitor enterprise environment nets that generally consist of 100s of similar network nodes such as a network rack, routers, switches, and personal computers. Further, they are designed to support very specific applications such as identifying the device's connectivity or latency or out-of-band analysis and so on. To obtain results they are homogenous in terms of protocol usage. For example, some systems use ICMP alone to obtain performance analysis of a huge network. Also, due to the lack of a buffer Dataset, they are not generally real-time and are quite torpid. A typical topology that consists of less than 50 network nodes such as a startup environment or a home office environment cannot afford such a stack of tools and maintaining a stable and high-performance network is an up- hill task for them. As a result, most of the emerging organizations employ a separate team that act as networkadministrators.

1.1 Scope

The developed network and server monitoring system addresses the problems of the current monitoring tools. It is an all-round application for network monitoring. The system is designed to monitor network information, server resources information and do basic configurations on network devices. The developed network monitoring system is web based. This allows network engineers and network administrators to remotely monitor the network infrastructure. Therefore, with today's technological improvements the system can be monitored by virtually any web enabled mobile device. This feature empowers the network engineers to monitor their systems even on-the-go. Inevitably in a failure of network connectivity or a device the network engineers and administrators can assist the situation while working offsite, whiletraveling, whileat home or even when they are abroad. The configuration facility also vastly improves the flexibility of the system. This reduces the need for another tool to do configurations on network devices. The monitoring system is based on open-source technologies as well as cross platform operability. The developed system is a combination of two applications. One is a web interface for the users to view and monitor network and server and also the terminal for viewing configurations of network devices. The other part is a backend injector application, which gathers information and raw data and processes in order to be displayed via the web interface. The entire system is hosted on a Window based server with web services.

1.2 Objective

1.2.1



CHAPTER 2

LITERATURE SURVEY

The survey of various papers on Network Monitors is presented in the following section:

[1] Network Monitoring System for Network Equipment Availability and Performance Reporting –by Baphumelele Masikisiki, SiyabulelaDyakalashre[2020]

It describes availability and performance of a network. Reporting with UFH being used as a test platform. This system is meant to help network administrators with troubleshooting network devices because it will tell them exactly where to go in order to fix the network. Additionally, the prototype will also have an ability to generate charts that presents the network performance in and untreatable format and also send notifications to the network administrator in cause of any outages.

Advantages and Disadvantages

- It used to test the availability and performances.
- The limitation of memory configurations led to the failure of memory performance

[2] Architecture of a Network Performance Monitor for Application Services on Multi-Clouds – by Young-min Kim, Ki-sung Lee, Jae-cheol Uhm, Si-chang Kim, and Chan-gun Lee 2013

Rudimentary reasons for having a network performance monitor for multi-c loud environments have been illustrated with examples in this research. An architecture for such a network monitor is proposed that explains how external agents are used to connect monitor such an environment. Also, a model to integrate various public clouds is formulated in a flexible manner. In addition, the issues of timely delivery and off-line analysis of measured results are addressed.

Advantages and Disadvantages:

- The model proposed for monitoring multi clouds is flexible and is integrated using external agents. These agents may be network intensive models for logging of network node events.
- This model isn't tested in real-life scenario and the results mentioned are through simulated environments.

- A DBMS approach to store results for offline analysis has been proposed which would be good for non-cloud network monitoring systems as well.

[3] Measurement-Aware Monitor Placement and Routing: A Joint Optimization Approach for Network-Wide Measurements – by Guanyao Huang, Chia-Wei Chang, Chen-Nee Chuah, and Bill Lin 2017

A theoretical framework is proposed in this research that jointly optimizes monitor placement and dynamic routing strategy to achieve maximum measurement utility with limited monitoring resources. Through experiments using real traces and topologies, it has been justified that these heuristic solutions can achieve measurement gains that are quite close to the optimal solutions, while reducing the computation times.

Advantages and Disadvantages:

- There are many implementation issues with the proposed framework that need to be addressed especially determining what routing protocols are being used.
- Also, this framework requires that the traffic reaches it in real-time instead of the framework polling for the same.
- The major takeaway from this research is the procedure involved in forming the base framework for a network monitoring system.

[4] A Web-based monitor and management system architecture for enterprise virtual private network – by Ruey-Shun Chen *, Change-Jen Hsu, Chan-Chine Chang, S.W. Yeh 2017

Monitoring encrypted connection gateways such as enterprise VPNs is a tedious. This paper formulates a feasible system that can monitor VPNs and includes essential elements such as system components, operation flow and much more. Also, it explains how a web GUI can be integrated with back-end for real time analysis.

Advantages and Disadvantages:

- This research is primarily localized on VPNs and encrypted gateways such as SSLVPN, Client-based VPNs and so on. Also, it is more focused on monitoring at a hardware level than at the end-user level.
- It is still dependent on external management tools in order for the enterprises to be rest assured for using the new technologies.

2.1 Comparative Analysis

Reference	Technique	Advantage	Drawback
Network Monitoring System for Network Equipment Availability and Performance Reporting	**SNMP**	**Used to Test the Performance and Availability**	**Memory configura tion**
Architecture of a Network Performance Monitor for Application Services on Multi-Clouds	**Multi-Cloud**	**Flexible and first of its kind**	**Network intensive and simulated results**
Measurement-Aware Monitor Placement and Routing: A Joint Optimization Approach for Network-Wide Measurements	**Routing**	**Method to implement a framework**	**No Real-time analysis**
A Web-based monitor and management system architecture for enterprise virtual private network	**VPN**	**Integration of GUI with back end**	**Covers only VPNs**

CHAPTER 3
SYSTEM ANALYSIS

3.1 Existing System

Networks serve as the backbone for any enterprise. Any network outage during working hours is huge loss for the organizations. As a result, they employ a separate team to look after their labs by constantly logging into several system interfaces and checking their statuses. A typical organization that consists of say 100 employees, the network includes at least 3 routers, 5-6 switches, 100-150 PCs, 3-5 servers, a cloud environment consisting of its 3-5 instances and 1-2 data stores, 20-30 VMs for testing their product. It is a tedious task to login to each of these nodes, check if they are reachable and check their health status constantly. As the enterprise grows, the numbers increase exponentially.

3.1.1 Drawbacks

- Requires separate login and thus the default login authentication system of the operating system is disabled.
- Does not monitor all shared resources.
- The limitation of memory configurations led to the failure of memory performance.
- Also, this framework requires that the traffic reaches it in real-time instead of the framework polling for the same.

3.2 Proposed System

Network Monitor is a web application that reports various status changes and updates of various network nodes in an organization The system basically comprises of 3 parts

Front End: Main functionalities include the following:

- Logging into the system
- Adding a device Editing a device
- Deleting a device
- Displaying the ‘Devices’ table

Database: It comprises of two tables:

- UserLogin is the administer the secure login activity to network monitor.
- Device table consists of the device name, locally generated device id, IP Address and System Parameters such as Reachability, Availability, Latency, Mac Address and so on.

Back End: A Python script that runs every 10 seconds to obtain information of all network nodes using different protocols and updates the same in the Database.

3.2.1 Features of the Proposed System

- It shows root cause Analysis
- It gives Alarm Management and configuration Management

3.3 Feasibility Study

feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that proposed system is not a burden to the organization. For feasibility analysis, some understanding of the major requirement for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

3.3.1 Economic Feasibility

This system is carried out to check the economic impact that the system will have on the organization. The amount of fund that the organization can pour into the research and development of system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirement of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing the system.

3.3.3 Operational Feasibility

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the user solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 Hardware Requirements

The following section describes the various hardware requirements for a network monitor to be developed and used:

- Processor: Intel® Core i5 ™ CPU andabove
- RAM: 8 GB or higher
- Hard Disk: 100 GB orhigher

4.2 Software Requirements

The following section describes the various hardware requirements for a network monitor to be developed and used:

- Operating System: Windows 10/Ubuntu 20.04 LTS
- Architecture: 64-bit OS
- Python 3.8 or higher
- PIP Packages: RegEx, Django
- Database: SQL LITE
- Front End: HTML5, CSS3, Bootstrap4

CHAPTER 5

PROJECT DESCRIPTION

5.1 problem Definition

Employees may feel their privacy has been devalued or violated. It may be difficult to retain employees if monitoring seems intrusive. Monitoring can signal a lack of trust, which can breed resentment and reduce employee morale and productivity. The line between work and home may be blurred, people often use the same devices for both, so where does monitoring cross the line. Extra data means more information could be misused if it lands in the wrong hands. Any monitoring program is only useful if it is actually scrutinized, which takes time and money. Surveillance may create a false sense of security, which can actually be a risk in and of itself. Networks serve as the backbone for any enterprise.

Any network outage during working hours is huge loss for the organizations. As a result, they employ a separate team to look after their labs by constantly logging into several system interfaces and checking their statuses. A typical organization that consists of say 100 employees, the network includes at least 3 routers, 5-6 switches, 100-150 PCs, 3-5 servers, a cloud environment consisting of its 3-5 instances and 1-2 data stores, 20-30 VMs for testing their product. It is a tedious task to login to each of these nodes, check if they are reachable and check their health status constantly. As the enterprise grows, the numbers increase exponentially. There are many open source and commercial products which provide network monitoring facilities. These applications provide network engineers and administrators with many features and tools to identify and examine network infrastructure. Many of the monitoring tools available in the industry provide specific monitoring tasks which are limited for one feature. The user web interface for the users to view and monitor network and server and also the terminal for viewing configurations of network devices.

5.2 Overview of the Project

Network Monitor is a scripted tool that alerts the User about various status changes and updates such as Reachability, Firmware version changes, Issues, Resource usage, Device Information and so on in network nodes such as Servers, Endpoint PCs, Routers, Switches, Virtual Machines, Data stores and so on. Alerts can be sent through various modes such as GUI, Email, SMS and so on. Various Protocols are used for obtaining this information such as ICMP, ARP, LLDP, SNMP and so on. A wide range and variety of devices are supported by the system such as Servers, Routers, Switches, Virtual Machines, IoT Devices, Cloud Instances, Data Stores, Wireless Access Points, Endpoint PCs, Printers, Mobiles and so on. A fully functional network monitoring system that can overlay various monitoring protocols such as ICMP, SNMP, SYSLOG, ARP, LLDP and so on to obtain all information about various network nodes. Development of easy installation script setup documentation. A lucid, user-friendly User Interface for the application that can display information with right intensity Consistent triggering for alerts in any inconsistent conditions that found in a network. Alerts should be sent to the users configured as per their customizations. allow your IT team to work behind the scenes fixing problems you didn't even know you had to keep the business network running smoothly. The benefits to network monitoring are not in simply knowing that an issue occurred, but in correcting the issue before it creates downtime for the end user or entire business.

Network Monitor is a web application that reports various status changes and updates of various network nodes in an organization The system basically comprises of three parts i.e., Front End, Database, Back End. A Python script that runs every 10 seconds to obtain information of all network nodes using different protocols and updates the same in the Database. Network monitoring notifies your system administrator anytime the network or segments of the network (server, firewall, switch, etc.) goes down for any reason. Even if the internet goes out, your system administrator is quickly notified so she can begin to take corrective action. This is especially useful for outages that may occur outside of normal business hours.

5.3 System Architecture

The system architecture of the proposed system is illustrated in the Figure 5.1

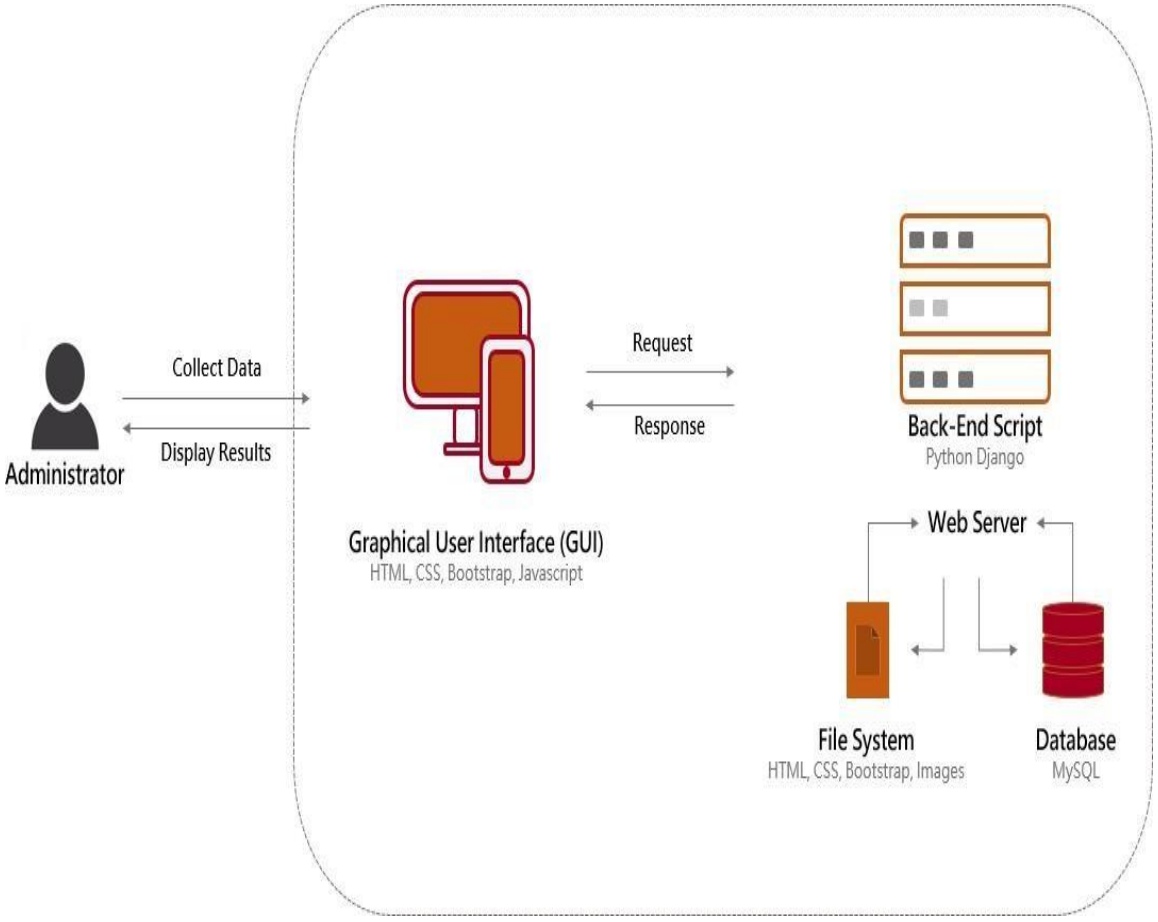


Fig.5.1: System Architecture

Administrator

Administrator class contains various attributes such as ID, Name and so on which are needed to login to the Network Monitor System to perform various functionalities such as Adding a device, deleting a device, configuring custom alerts, configuring contact details such an email and so on and taking access to the device.

Graphical User Interface:

This facilitates the user to login into the system, initiate the utility, whereby detail of the entire network is present in a table. The Traffic Status will show up eventually which are obtained from the device using ICMP protocol. The user can also see all the performance attributes such as reachability, latency and so on present on any network node by switching between tabs. Coding Languages used for this Component: HTML, CSS, Bootstrap Templates, JavaScript.

Back-end Processing:

This module handles all functionality from finding the network nodes, attributes, querying with the network nodes and updating the database with information obtained through back-end processing. Coding Languages used for this Component: Python, Django Framework, JavaScript, Pymysql framework**.**

Database:

This module handles acts a buffer between the backend functionality and the graphical user interface. It provides synchronization between the activities of the user and the backend. MySQL Tables used:

User Authentication: Deals with user login, logout, and session keys.

Devices: Deals with devices and their attributes at real time.

Languages used for this Component: MySQL Database, MySQL query Language, MySQL Workbench.

5.4 Module Description

Different Modules involved in a Network Monitor are explained in detail below:

Graphical User Interface:

- This facilitates the user to login into the system, initiate the utility, whereby detail of the entire network is present in a table.
- This table is automatically filled with details of the network node such as MAC Address, OS and so on using LLDP protocol
- The Traffic Stats will show up eventually which are obtained from the device using ICMP protocol

- The user can also see all the performance attributes such as reachability, latency and so on present on any network node by switching between tabs.
- The user can take SSH/Web/Telnet access to the network node by clicking on Access button in the table.
- The user can update, modify, change, or configure the network nodes. This triggers a background process of history keeping. It also provides the ability to caution any change of an attribute which may not be compatible with the resource at hand.
- The GUI also shows latest alerts on the top with various color codes assigned based on the criticality of the event.
- Types of Alerts and their frequency can be configured under the Profile option.
- Coding Languages used for this Component: HTML, CSS, Bootstrap Templates, JavaScript.

Back-end Processing:

- This module handles all functionality from finding the network nodes, attributes, querying with the network nodes and updating the database with information obtained through back-end processing.
- It also maintains a dynamically updated database which contains all the attributes present on every resource along with compatibility problems. As a result, it can take decisions on issues occurred and report errors accordingly.
- Various protocols are used to obtain different types of data for example LLDP for device information, SNMP for performance statistics, SSH/Telnet/HTTP for device access, ICMP for traffic statistics and so on.
- Raw IP sockets are formed and sent in various ways to determine the operating system the network nodes are using, packet filters in use and several other attributes. Coding Languages used for this Component: Python, Django/Flask Framework, JavaScript, Pymysql framework.

Database:

- This module handles acts a buffer between the backend functionality and the graphical user interface. It provides synchronization between the activities of the user and the backend.

- To avoid top-down triggering and querying of the network node data every time the web page is loaded or meta-refreshed, the backend functionality runs infinitely to fill various attributes of the network node by querying it using various network protocols at regular intervals.
- Simultaneously, GUI fetches data from database which results in real-time transmission at any given second.
- MySQL Tables used:
  - UserAuthentication: Deals with user login, logout, and session keys
  - Devices: Deals with devices and their attributes at real time

Languages used for this Component: MySQL Database, MySQL query Language, MySQL Workbench / MySQL CLI for diagnostics.

Agent: This is a software that is installed on a device that is being managed by SNMP.

Management System: This is software is installed on Manger.

SYSLOG: It is a standard for system logging. In short, we can say anything we configure to log from a system will be forwarded and will store like a typical log book. SYSLOG use UDP port 514 for communication between SYSLOG servers and clients. As its UDP we know that it doesn't do reliable communication. SYSLOG packet contains information about the facility Severity, Hostname, Timestamp, and Message. SYSLOG messages are categorized widely basis of the source generate them. For example, it could be a router, switch, an operating system, process etc. These categories are called by the Facility. Let see how SYSLOG works and it can help us in monitoring. SYSLOG has three Components by which it performs logging. There are three SYSLOG components described below.

- **Log device**: devices which create log messages such as router, firewall, server or any device which can create an event message for SYSLOG.
- **Log collector**: These components collect SYSLOG data. SYSLOG server would be a typical log collector in this case.
- **Log relay**: Sometime message needs to forward through a log collector. These component forward messages which they receive from log devices.

Typical applications and OS are created in such a way that they are capable to give clues of their condition or any change of state. These clues help us to understand what is the application is doing? Individual applications and OS deliver message differently but SYSLOG is one of such application which is capable of accepting messages from different vendors at the same time. Logging is a very important tool for network admin to identify unusual activity in the network. This can be used for debugging any problems also. SYSLOG can be used as intrusion detection system. If any system is compromised, intruder tries to cover the footprint but with SYSLOG it's really hard because it gives real time logging. So, network admin actually can find out instantly or later how intruder compromised the system. If any system is compromised, a proper investigation will be started in order to determine of severity and compromise. At that time SYSLOG played a great role in creating a picture of the system of that time. Let say we monitoring one of the routers. What kind of info we can get by SYSLOG. Nay port failure will be logged. So, we can instantly fix the problem by watching log message. Some time we need to see who logged remotely in our system. We can get info also form SYSLOG. SYSLOG provides info of different severity level which helps us to understand how the system is healthy and what steps we need to take based on that severity level.

ICMP: Ping is a basic Internet program that uses ICMP and allows a user to verify that a particular IP address exists and can accept requests. Ping is used diagnostically to ensure that a host computer the user is trying to reach is actually operating. Ping works by sending an Internet Control Message Protocol (ICMP) Echo Request to a specified interface on the network and waiting for a reply. Ping can be used for troubleshooting to test connectivity and determine response time Ping command is very useful in debugging and testing network and internet connection.

ARP: The Address Resolution Protocol (ARP) is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address. The Address Resolution Protocol uses a simple message format containing one address resolution request or response. The message header specifies the types of networks in use at each layer as well as the size of addresses of each.

LLDP: The Link Layer Discovery Protocol (LLDP) is a vendor-neutral link layer protocol used by network devices for advertising their identity, capabilities, and neighbors on a local area network based on IEEE 802 technology, principally wired Ethernet. Media Endpoint Discovery is an enhancement of LLDP, known as LLDP-MED, that provides:

- Auto-discovery of LAN devices
- Device location discovery to allow creation of location databases and, in the case of Voice over Internet Protocol (VoIP)
- Inventory management, allowing network administrators to track their network devices, and determine their characteristics (manufacturer, software and hardware versions, serial or asset number).

Web Interface: Data which is collected by the core software and the syslog is stored in the database. The database is read by a web application which is written using Python. Main objective here is to display the important information which is collected by the software in an attractive manner. Technologies like JavaScript, CSS, and jQuery are used in the client side and Python is used as the server-side programming. One of the most important objectives we accomplished in this was that we could be able to run basic configuration commands using the web interface. The administrator can run simple commands like show and debug, then observe the output of those commands in the same web interface. And the most important thing is this process can be done from anywhere in the world.

5.5 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

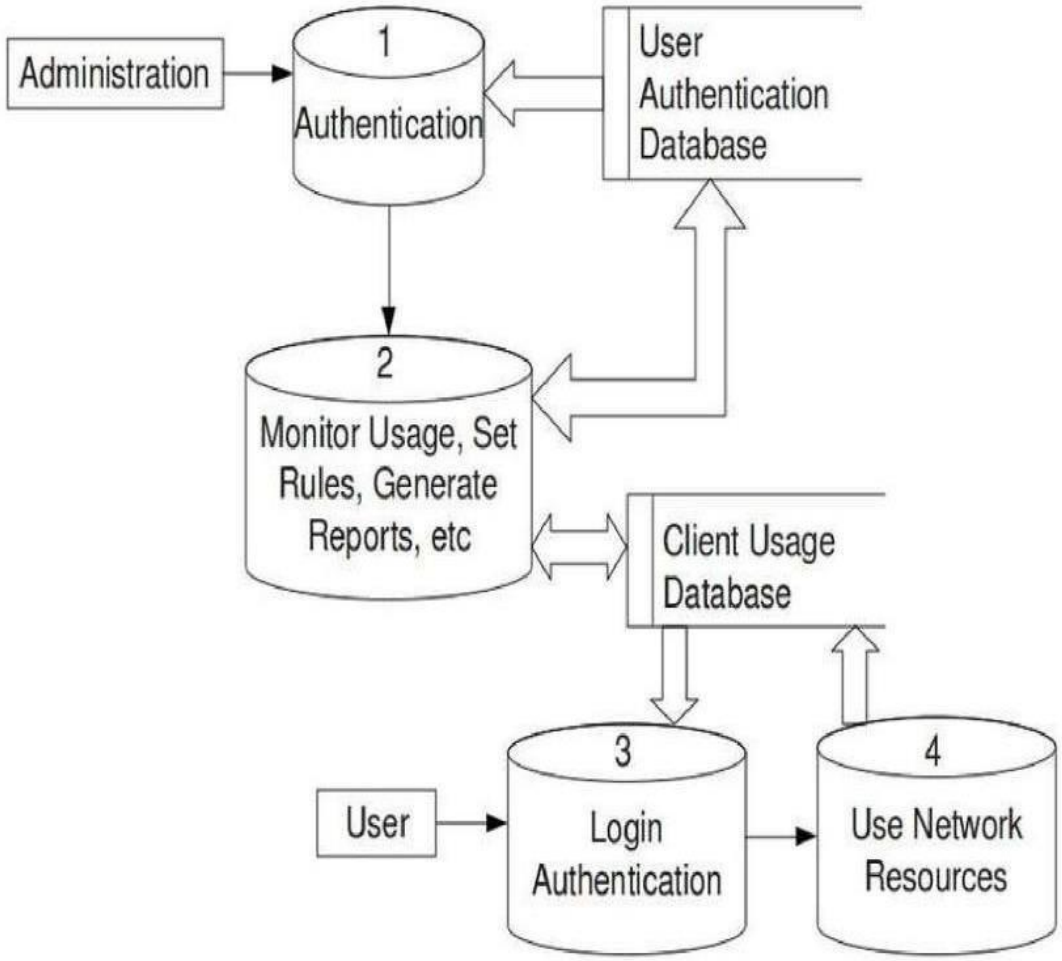


Fig:5.2 Data Flow Diagram

The Figure 4.5 shows the various data items flowing for Network Monitor. The various data items involved in this design are: Authentication, addition/deletion of devices, alerts, database updates, query retrievals, device monitoring and device data updates. Authentication involves registration and login modules of the network monitor. When the user enters credentials into the GUI and clicks on Login, the credentials are encrypted by backend processing and sent to the Database through SQL Query exchange.

When the database validates the authentication request, the backend creates a new session along with their tokens. Then appropriate message is displayed on GUI. Adding/Deleting a device basically involves addition/removal of rows in Device Table of the database which are identified using the device id, name, and IP address as parameters. An added device is monitored using different protocols, namely, ICMP, SNMP, LLDP, ARP and SYSLOG in the initial phase. Each protocol is used to obtain a particular data about various parameters attributing to the device. For example, ICMP provides us the reachability

status and TTL values for required to reach the device. Similarly, LLDP is used to obtain Serial number, chassis number, and firmware version and so on from the LLDP RX packets from the device. Administrators are facilitated with various options to configure alerts such as allowing only critical alerts to be sent at certain times. Based on this configuration and criticality of the events, various logs are accumulated and sent in email to the user. Backend processing is basically one constantly running python scripts that updates a database using Pymysql framework, which acts as a buffer for GUI to load quickly, every time an update comes up from added devices.

5.6 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams.

The Figure 4.2 shows the Use Case Diagram for Network Monitor that illustrates the relationship between the user and the system in different use cases. Various actors involved in the product are Administrator – user who controls the network monitor system, Back-end – involves a python script running the functions, Database – to store and relay the obtained data from devices and Devices that are monitored as per administrator's instruction.

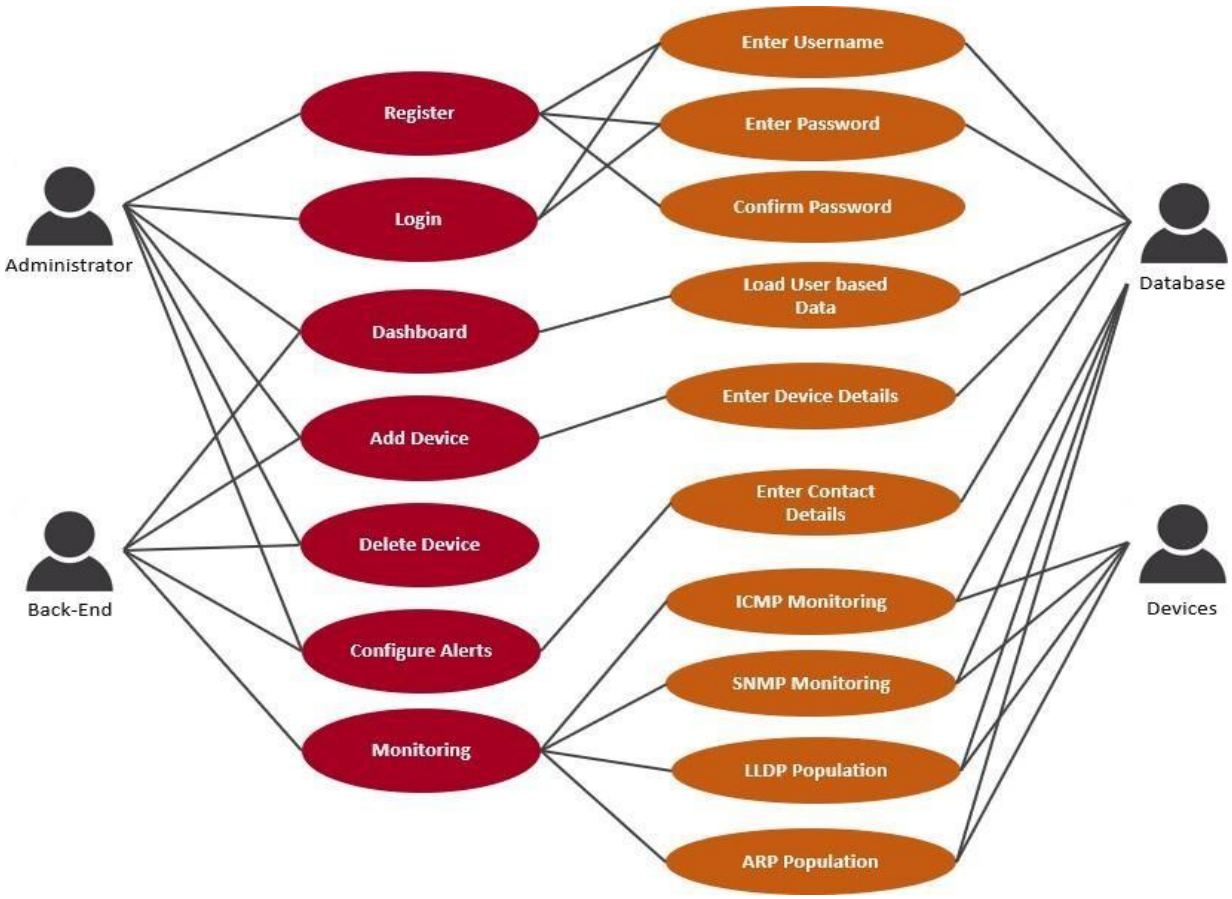


Fig.5.3 use case Diagram

5.7 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. Figure 4.4 explains how an administrator logs into the Network Monitor System. Various steps performed for this case are explained below:

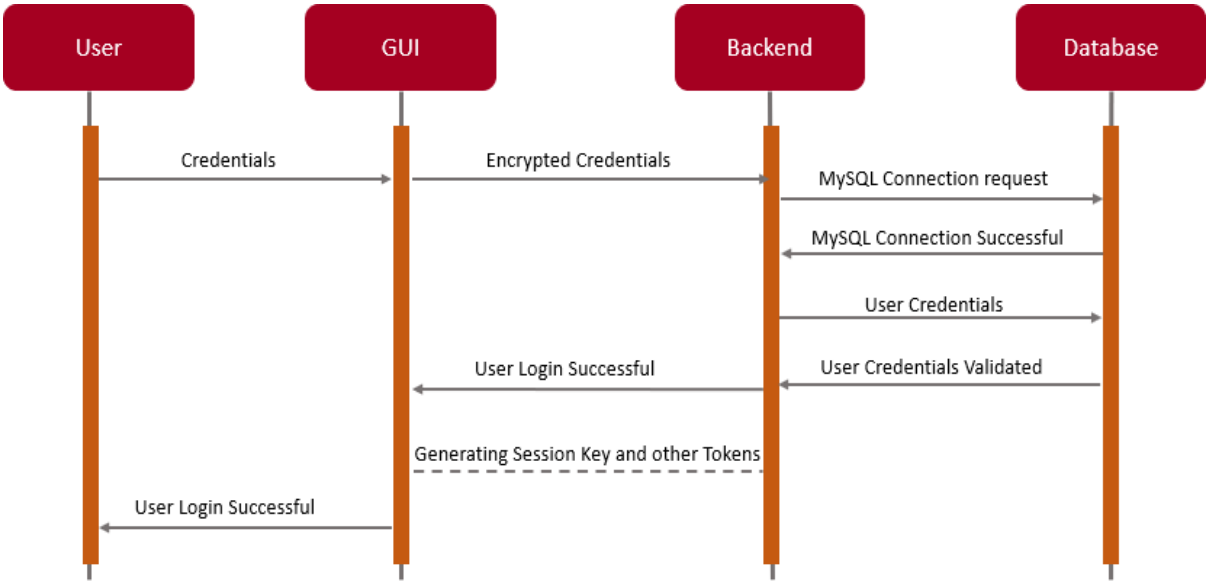


Fig.5.4: Sequence Diagram

Firstly, the user clicks on the "Login" button and enters the credentials namely username and password in the graphical user interface and clicks on "Submit" button. These credentials are then encrypted by the Django framework's session manager and sent to the Backend python script. The Backend control makes connection request to the Database and sends the user credentials upon connection establishment. The Database, MySQL in this case, would verify the credentials and sends a "OK" message if valid as a query response. The Framework then generates session key and other tokens to manage the user session. Upon successful creation of the session, a message is displayed on the GUI that "User Login is Successful" and loads the customer specific dashboard.

CHAPTER 6

SYSTEM TESTING

6.1 Introduction

Software testing is the process used to help identify the correctness, completeness, security and quality of developed computer software. With that in mind, testing can never completely establish the correctness of arbitrary computer software. In computability theory, a field of computer science, an elegant mathematical proof concludes that it is impossible to solve the halting problem, the question of whether an arbitrary program will enter an infinite loop, or halt and produce output. In other words, testing is criticism or comparison that is comparing the actual value with an expected one. There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation, not merely a matter of creating and following rote procedure. One definition of testing is "the process of questioning a product in order to evaluate it", where the "questions" are things, the tester tries to do with the product, and the product answers with its behavior in reaction to the probing of the tester. Although most of the intellectual processes of testing are nearly identical to that of review or inspection, the word testing is connoted to mean the dynamic analysis of the product, putting the product through its paces. The quality of the application can, and normally does, vary widely from system to system but some of the common quality attributes includes reliability, stability, portability, maintainability and usability

6.1 Testing Strategies

Designing effective test cases is important but so is the strategy to use them to execute them. If it is conducted in haphazard manner time is wasted and unnecessary effort is expended. Thus, it seems reasonable to establish a systematic strategy for testing software

6.1.1 Unit Testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated.

It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2 Integration Testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.1.3 Functional Testing:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input: identified classes of valid input must be accepted.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.1.4 System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1.5 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

6.1.6 Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.1.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Test Cases:

Test cases can be divided in to two types. First one is Positive test cases and second one is negative test cases. In positive test cases are conducted by the developer intention is to get the output. In negative test cases are conducted by the developer intention is to don't get the output.

SUCCESS TEST CASE

Table 6.1 Success test case

S. No	Test Case Description	Expected Result	Actual Result	Results
1	Enter Registered Username	Registered Username Is valid	Successful Login	Pass
2	Enter Valid Passwo rd	Password entered is Valid	Successful Login	Pass
3	Enter the devices name	Displaying the Total Devices Information	Successful updated the devices Information	Pass
4	Status of the devices	Live/not Reachable	Alert to user if not Reachable	Pass

S.No	Test Case Description	Expected Result	Actual Result	Result
1	Enter Registered Username	Registered username is Invalid	Login failed	Fail
2	Enter Valid Password	Password entered is Invalid	Login failed	Fail
3	Status of the devices	Live/not Reachable	Alert to user if not Reachable	Fail

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 Introduction

Graphical User Interface:

This facilitates the user to login into the system, initiate the utility, whereby detail of the entire network is present in a table. The Traffic Status will show up eventually which are obtained from the device using ICMP protocol. The user can also see all the performance attributes such as reachability, latency and so on present on any network node by switching between tabs. Coding Languages used for this Component: HTML, CSS, Bootstrap Templates, JavaScript.

HTML

HTML (Hyper Text Markup Language): The Hyper Text Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript. Web receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.In between these tags' web designers can add text, tags, comments and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML markup. Another important component is the HTML document type declaration, which triggers standards mode rendering.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML andJavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications. CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display.

The web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified. Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents. The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file which reduces complexity and repetition in the structural content as well as enabling the .CSS file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

Bootstrap

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is the second most-starred project on GitHub, with more than 107,000 stars and 48,000 forks. Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden. According to twitter developer Mark Otto: "A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within the company. After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a hackathon-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap, and released as a Bootstrap is a HTML, CSS & JS Library that focuses on simplifying the development of informative web pages. The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking.

Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-coloured tables, page headings, more prominent pull quotes, and text with a highlight. Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto- complete function for input fields. The most prominent components of Bootstrap are its layout components, as they affect an entire web page.

Back-end Processing:

This module handles all functionality from finding the network nodes, attributes, querying with the network nodes and updating the database with information obtained through back-end processing.

Coding Languages used for this Component: Python, Django Framework

Python

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and collected. It Supports Multiple Programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included language" due to its comprehensive library. Python is a multi-paradigm programming language. Object- oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution. It supports functional and structured programming methods as well as OOP. It can be used as a scripting language or can be compiled to byte-code for building large applications. Python can be used on a server to create web applications. Python can be used alongside software to create workflows. Python can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production-ready software development. Python is a popular programming language.

It was created by Guido van Rossum, and released in 1991.It is used for: web development (server-side), software development System scripting. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-oriented way or a functional way. Its ease of use. For those who are new to coding and programming, Python can be an excellent first step. It's relatively easy to learn, making it a great way to start building your programming knowledge. Its simple syntax. Python is relatively easy to read and understand, as its syntax is more like English. Its straightforward layout means that you can work out what each line of code is doing. Its thriving community. As it's an open-source language, anyone can use Python to code. What's more, there is a community that supports and develops the ecosystem, adding their own contributions and libraries. Its versatility. As we'll explore in more detail, there are many uses for Python. Whether you're interested in data visualisation, artificial intelligence or web development, you can find a use for the language.

Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Ridiculously fast. Django was designed to help developers take applications from concept to completion as quickly as possible. Reassuringly secure. Django takes security seriously and helps developers avoid many common security mistakes. Exceedingly scalable. Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support. Versatile Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites.

It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django Internally, while it provides choices for almost any functionality you might want (e.g., several popular databases, templating engines, etc.), it can also be extended to use other components if needed. Secure Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hashDjango enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking Maintainable Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules Portable Django is written in Python, which runs on many platforms.

Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template.

Database:

This module handles acts a buffer between the backend functionality and the graphical user interface. It provides synchronization between the activities of the user and the backend MySQL Tables used:

UserAuthentication: Deals with user login, logout, and session keys

Devices: Deals with devices and their attributes at real time

Languages used for this Component: MySQL Database, MySQL query Language, MySQL Workbench

SQLLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly. SQLite does not require a separate server process or system to operate (serverless). A complete SQLite database is stored in a single cross-platform disk file. SQLite is very small and light weight SQLite is self-contained, which means no external dependencies. SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads. SQLite supports most of the query language features found in SQL92 (SQL2) standard. SQLite is written in ANSI-C and provides simple and easy-to-use API. SQLite is an in-process library that implements a self- contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several projects. SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format. SQLite database files are a recommended storage format by the US Library of Congress. Think of SQLite not as a replacement for Oracle but as a replacement for fopen()SQLite is a compact library. With all features enabled, the library size can be less than 600KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function in lining and loop unrolling can cause the object code to be much larger.) There is a trade-off between memory usage and speed. SQLite generally runs faster the more memory you give it.

Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be SQLite is very carefully tested prior to every release and has a reputation for being very reliable. Most of the SQLite source code is devoted purely to testing and verification. An automated test suite runs millions and millions of test cases involving hundreds of millions of individual SQL statements and achieves 100% branch test coverage. SQLite responds gracefully to memory allocation failures and disk I/O errors. Transactions are ACID even if interrupted by system crashes or power failures. All of this is verified by the automated tests using special test harnesses which simulate system failures. Of course, even with all this testing, there are still bugs. But unlike some similar projects (especially commercial competitors) SQLite is open and honest about all bugs and provides bugs list and minute-by-minute chronologies of code changes. The SQLite code base is supported by an international team of developers who work on SQLite full-time.

7.2 Screen Shorts

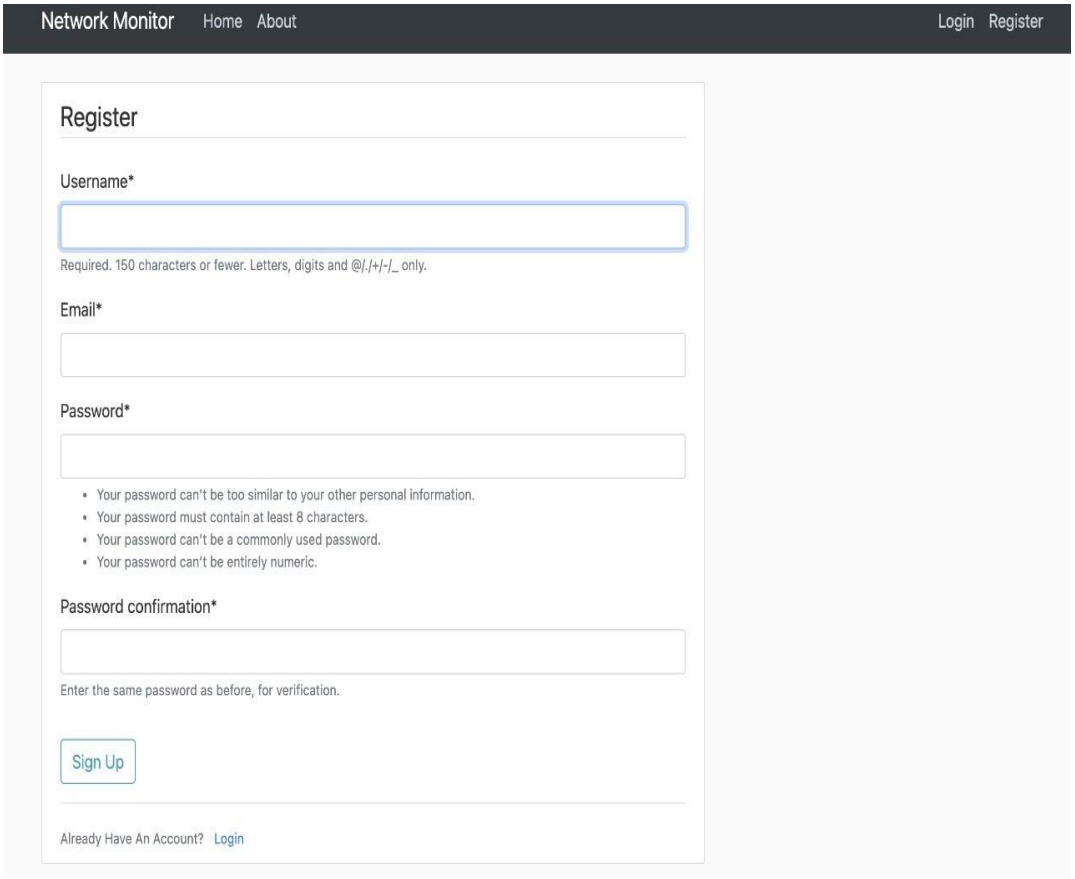


Fig.7.1: Register Page

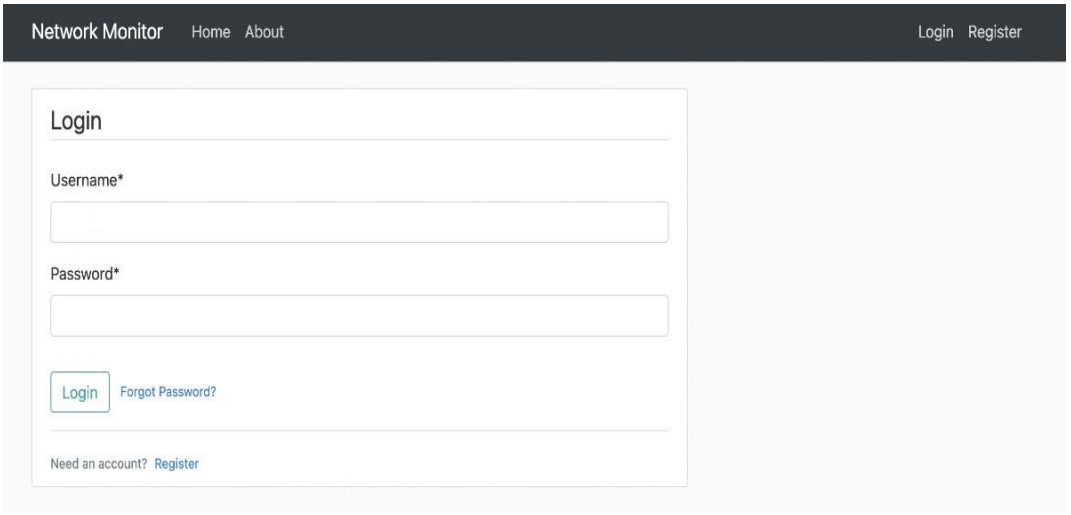


Fig.7.2: Login Page

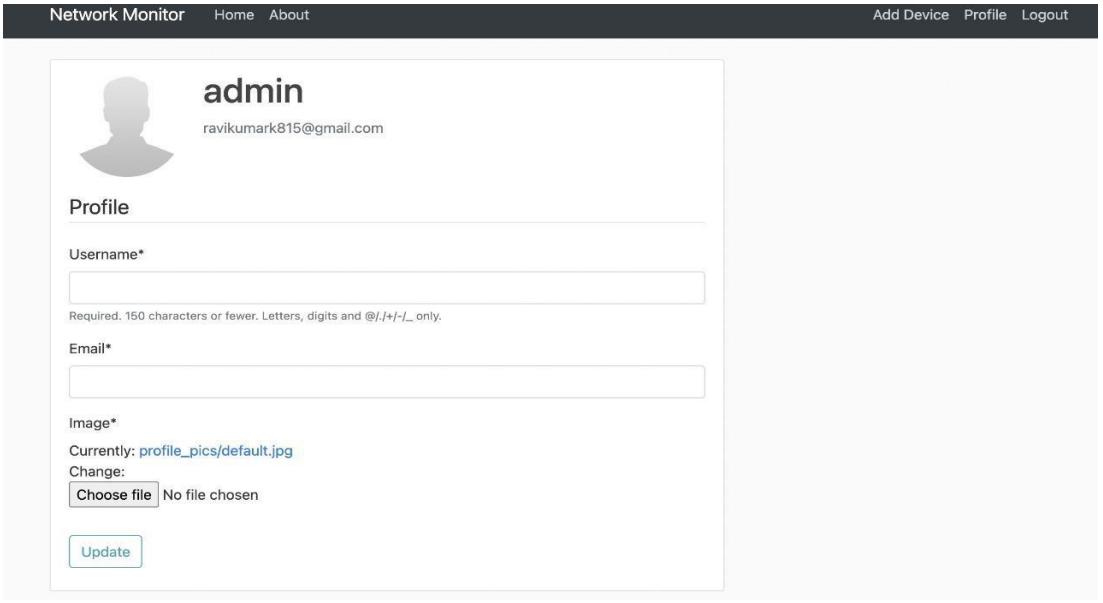


Fig.7.3: User Profile Page

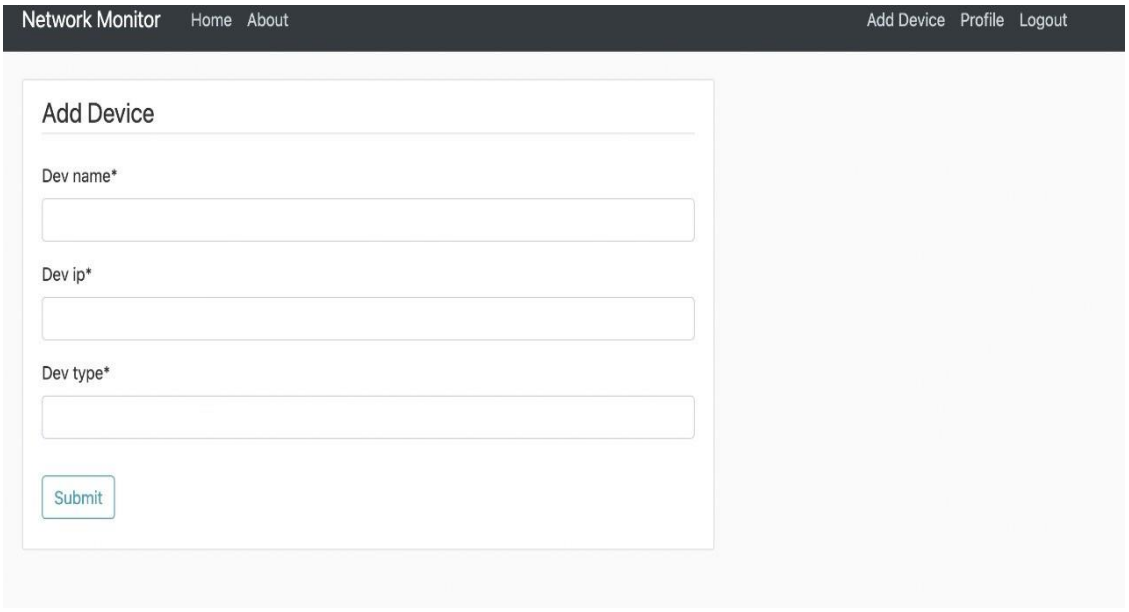


Fig.7.4: Adding Device

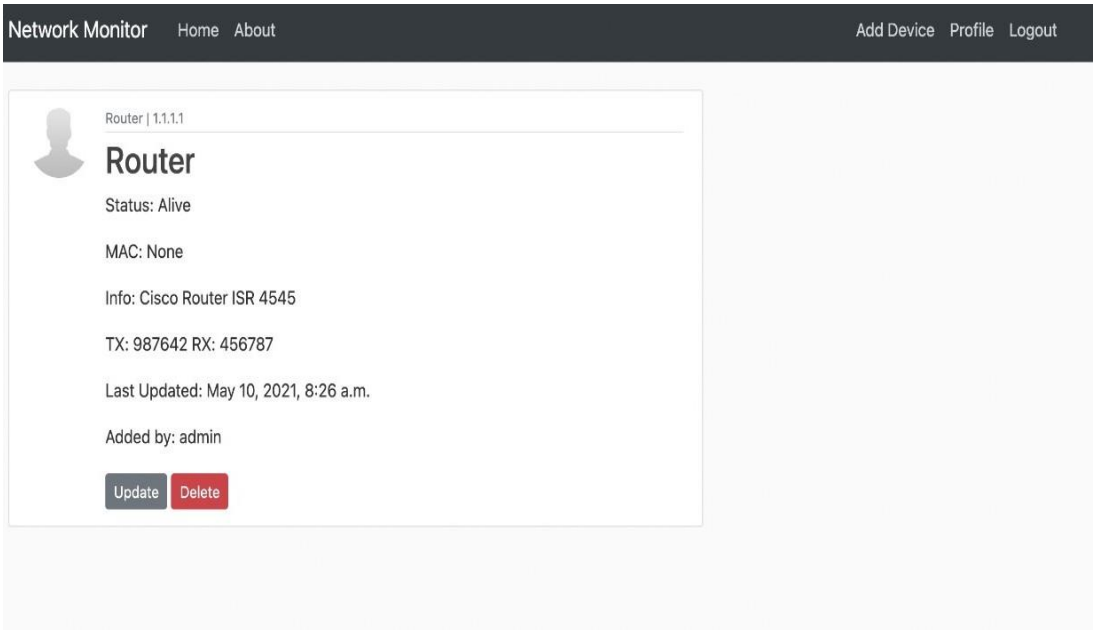


Fig.7.5: Total Information about each device

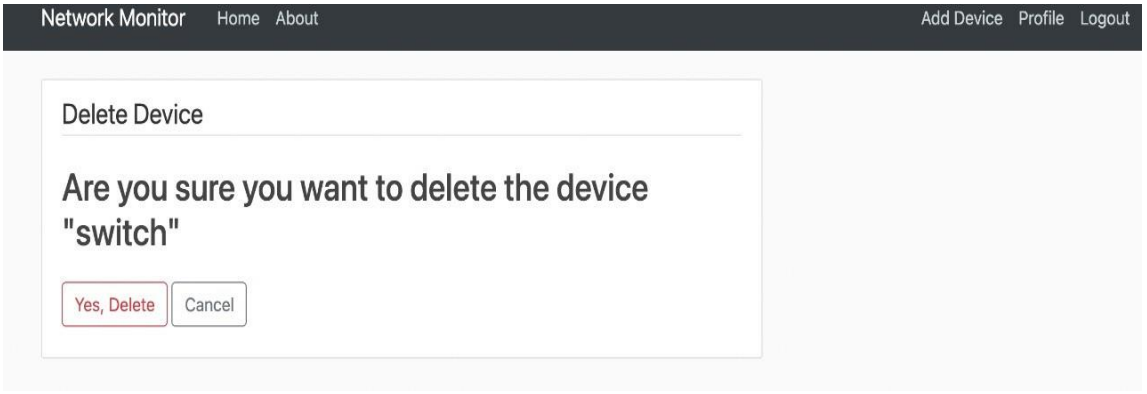


Fig.7.6: Delete device

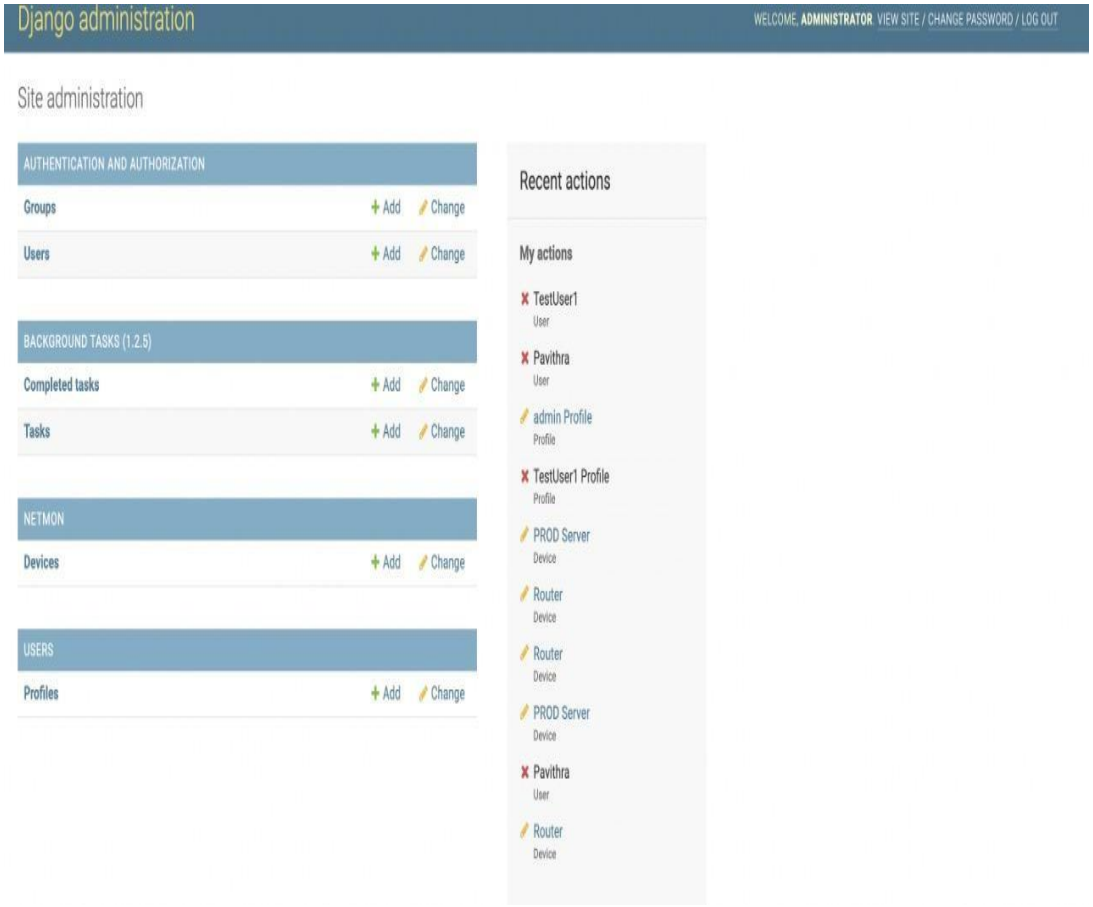


Fig.7.7: Admin Page

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

This report illustrates how very basic tools can be used to monitor the network. Knowing that simple tools can be so useful in monitoring our network, it gives details about all those tools work which most network administrators don't really take note, also with the details giving in this report reader can easily see which type of network monitoring to use and what it can monitor. And depends on those tools, companies are creating tailor made monitoring tools. These monitoring tools giveus a great flexibility to monitor our network in order to tune the performance measurement in a great extent. Moreover, they tell us what to monitor deeper, why should we monitor and what can be done with the result. Monitoring a network is very essential, intelligent management of the business. It helps the business grow and prevent them from un- necessary downtime due to inappropriate use of resources.

8.2 Future Enhancement

This tool can be further used in areas such as remote monitoring, where we can monitor branch network of a company from the main office. On the other hand, we could implement in mobile devices such as the iPhone or Android phone for monitoring people so that they can monitor their network on mobility. SNMP could have been implemented in order to get more info from servers, routers and network printers. Servers with virtual machine could be monitored. Cloud monitoring is also possible to implement by thistool.

REFERENCES

[1] **Dynamic Network Control with QoS and Resource Priority Monitor Based on Active "eM" for Commercial VoIP** – by Melanie Grah and Dr. Peter Radcliffe 2018

[2] **Architecture of a Network Performance Monitor for Application Services on Multi-Clouds** – by Young-min Kim, Ki-sung Lee, Jae-cheol Uhm, Si-chang Kim, and Chan-gun Lee 2016

[3] **Measurement-Aware Monitor Placement and Routing: A Joint Optimization Approach for Network-Wide Measurements** – by Guanyao Huang, Chia-Wei Chang, Chen-Nee Chuah, and Bill Lin 2017

[4] **Study on monitor and control of POL transport by road in IOT** – by Yang Chen, Qidong Yong, Dong Xiang 2017

[5] **Distributed Interplanetary Delay/Disruption Tolerant Network (DTN) Monitor and Control System** – by Shin-Ywan Wang 2012

[6] **Overhead Contact System On-line Monitor Technology Based on Wireless Sensor Network** – by Jiangjian Xie and Yi Wang, Tingting Lu 2011

[7] **The Design and implementation of a UPS Monitor and Control System** – by Lidong Fu and Bin Zhang 2011

[8] **Fault-tolerant Schemes for NoC with a Network Monitor** – by Zhang Ying , Wu Ning, Wan Yu Peng , Ge Fen , Zhou Fang 2010

[9] **Using activity sensitivity and network topology information to monitor project time performance** – by Mario Vanhoucke1. 2010

[10] A transparent virtual machine monitor level packet compression network service – by Ali Hamidi, Hadi Salimi and Mohsen Sharifi. [2010]

[11] **On evaluating the differences of TCP and ICMP in network measurement** – by Li Wenwei, Zhange Dafang, Yang Jinmin and Xie Gaogang [2017]