CENTRAL MANUFACTURING TECHNOLOGY INSTITUTE

APPERENTICE REPORT ON

“Network IP Based Data Visualization”

Submitted by:

Lalit Simariya

Apperentice (CMTI), Sl No.: 70040

Information Technology (C-SVT)

Central Manufacturing Technology Institute,

Bengaluru–560022, Karnataka.

Under the mentorship of:

Mr. Vinodh Venkatesh Gumaste

Scientist – C and Group Head– IT,

Centre for Sensor & Vision Technology (C-SVT).

Central Manufacturing Technology Institute,

Bengaluru – 560 022, Karnataka

In the duration of

12TH JULY,2023 to 12th JULY,2024

C-SVT DEPARTMENT

<u>APPRENTICESHIP COMPLETION LETTER</u>

Date:12-06-2024

Subject: Request for the Internship completion certificate.

Respected Sir/Madam,
I am Lalit Simariya, working here as a Apprentice in the department of C-SVT. I'm thankful to CMTI for providing me the opportunity to carry out my training on the topic "Network IP Based Data Visualization", under the guidance of Mr. Vinod Venkatesh Gumaste**,** Scientist- C and Group Head IT, Centre for Sensor & Vision Technology (C-SVT).

I am grateful for all the experience and knowledge I have received during my Apprentice from January 12th, 2023 to June 12th, 2024

I hereby request that you kindly issue the Apprenticeship completion certificate.

Thanking you,

Yours Sincerely,
Lalit Simariya
IT - Apprentice
C-SVT

Mr. Vinodh Venkatesh Gumaste
Scientist–C and Group head - IT
Centre for Sensor & Vision Technology
(C-SVT)
Central Manufacturing Technology Institute

DECLARATION

I**,** Lalit Simariya, hereby declare that the presented Apprentice report is uniquely prepared by me after the completion of Apprenticeship for aduration of 12 months at the Centre for Sensor & Vision Technology (C-SVT) department of Central Manufacturing Technology Institute (CMTI), Bangalore.

I also confirm that the report is only prepared for my academic requirement. It will not be used with the interest of any other organization without written permission of the Director of CMTI.

Lalit Simariya
(Sl No.: 70040)

ACKNOWLEDGEMENT

I am deeply thankful to the Head of the Academy of Excellence for Advanced Manufacturing Technology (AEAMT), CMTI, Mrs. Asha R Upadhyaya, for providing me with the opportunity to be a part of CMTI as a Apprentice. This Apprenticeship experience at CMTI has been an exceptional chance for my learning and professional development, and I am truly grateful for the exposure it has provided me.

I express my sincere gratitude to my supervisor, Mr. Vinodh Venkatesh Gumaste, Scientist-C and Group Head (IT), Centre for Sensor & Vision Technology
(C-SVT) whose invaluable guidance, unwavering support, and boundless encouragement have been instrumental in shaping the outcome of this report.

I am immensely grateful for the support and guidance provided to me throughout the course of this Apprentice, and I extend my heartfelt appreciation to all those who have contributed to its successful completion.

I am confident that the practical experience I have gained at Central Manufacturing Technology Institute (CMTI)will greatly contribute to my future success in the industry. I am excited to apply the valuable lessons learnt from this Training to future projects and continue growing as a professional.

Lalit Simariya

Table of Figures

Figure 1- Introduction..6

Figure 2.1- Uses & Features 6

Figure 3.1- Python 3.11 7

Figure 2.2 -Project Flowchart 8

Figure 2.3 - Coding Languages and Library..8

Figure 4.1- Django 4.2.4 9

Figure 3.2- MySql 8.0..10

Figure 5.1- Python Backend Functions..10

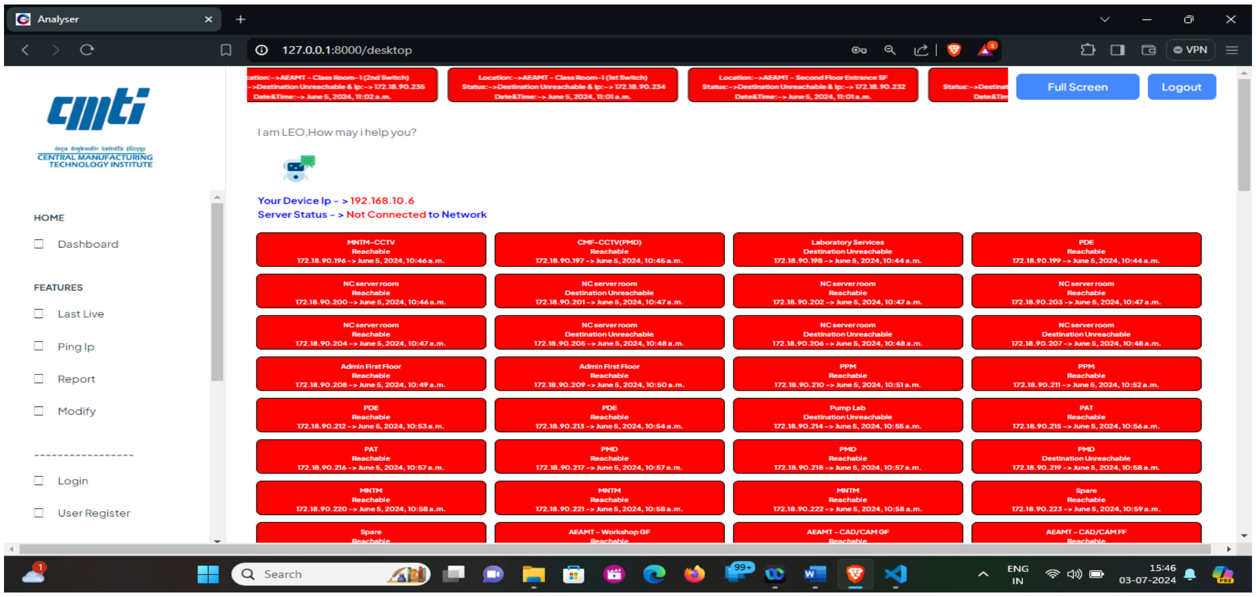
Figure 3.4- CMTI Network Connection 16

Figure 3.4- LEO chatbot 17

IP ping data visualization involves representing and analyzing data related to the results of Internet Control Message Protocol (ICMP) ping requests. Ping is a network utility tool that sends a message (ICMP echo request) to a target host and waits for a response (ICMP echo reply). Visualizing ping data can provide insights into network latency, packet loss, and the overall health of a network. Here are some aspects and tools related to IP ping data visualization

Uses & Features

- Centrally monitoring of network based IP devices, implemented using the ping command of IP protocol
- To visualize the data, created web page . It is helpful for IT team.
- The web page including the live data of all the networking devices status like Reachable , Unreachable.
- This website also ping IP to submit manually and showing status stored in database.
- For further information, use the LEO chatbot.
- Fullscreen Feature also added.



cation:-->AEAMT - Class Room-1 (2nd Switch) ->Destination Unreachable & Ip:--> 172.18.90.235 Date&Time:--> June 5, 2024, 11:02 a.m.

Location:-->AEAMT - Class Room-1 (1st Switch) Status:-->Destination Unreachable & Ip:--> 172.18.90.234 Date&Time:--> June 5, 2024, 11:01 a.m.

Location:-->AEAMT - Second Floor Entrance SF Status:-->Destination Unreachable & Ip:--> 172.18.90.232 Date&Time:--> June 5, 2024, 11:01 a.m.

Status:-->Destinat Date&Tim

I am LEO,How may i help you?

Your Device Ip - > 192.168.10.6
Server Status - > Not Connected to Network

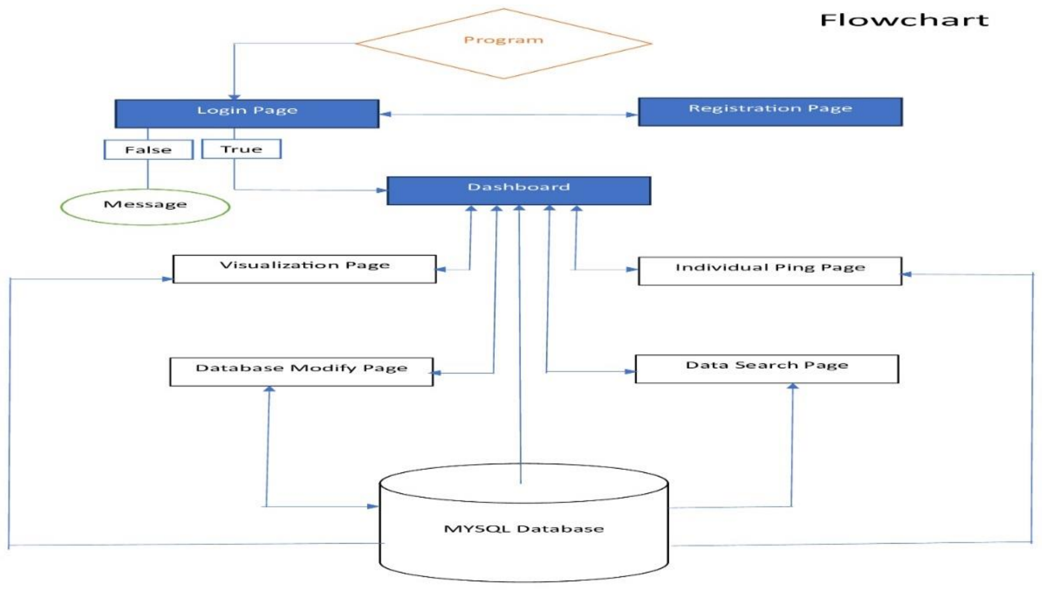
MNTM-CCTV Reachable 172.18.90.196 -> June 5, 2024, 10:46 a.m.	CMF-CCTV(PMD) Reachable 172.18.90.197 -> June 5, 2024, 10:45 a.m.	Laboratory Services Destination Unreachable 172.18.90.198 -> June 5, 2024, 10:44 a.m.	PDE Reachable 172.18.90.199 -> June 5, 2024, 10:44 a.m.
NC server room Reachable 172.18.90.200 -> June 5, 2024, 10:46 a.m.	NC server room Destination Unreachable 172.18.90.201 -> June 5, 2024, 10:47 a.m.	NC server room Reachable 172.18.90.202 -> June 5, 2024, 10:47 a.m.	NC server room Reachable 172.18.90.203 -> June 5, 2024, 10:47 a.m.
NC server room Reachable 172.18.90.204 -> June 5, 2024, 10:47 a.m.	NC server room Destination Unreachable 172.18.90.205 -> June 5, 2024, 10:48 a.m.	NC server room Destination Unreachable 172.18.90.206 -> June 5, 2024, 10:48 a.m.	NC server room Destination Unreachable 172.18.90.207 -> June 5, 2024, 10:48 a.m.
Admin First Floor Reachable 172.18.90.208 -> June 5, 2024, 10:49 a.m.	Admin First Floor Reachable 172.18.90.209 -> June 5, 2024, 10:50 a.m.	PPM Reachable 172.18.90.210 -> June 5, 2024, 10:51 a.m.	PPM Reachable 172.18.90.211 -> June 5, 2024, 10:52 a.m.
PDE Reachable 172.18.90.212 -> June 5, 2024, 10:53 a.m.	PDE Reachable 172.18.90.213 -> June 5, 2024, 10:54 a.m.	Pump Lab Destination Unreachable 172.18.90.214 -> June 5, 2024, 10:55 a.m.	PAT Reachable 172.18.90.215 -> June 5, 2024, 10:56 a.m.
PAT Reachable 172.18.90.216 -> June 5, 2024, 10:57 a.m.	PMD Reachable 172.18.90.217 -> June 5, 2024, 10:57 a.m.	PMD Reachable 172.18.90.218 -> June 5, 2024, 10:57 a.m.	PMD Destination Unreachable 172.18.90.219 -> June 5, 2024, 10:58 a.m.
MNTM Reachable 172.18.90.220 -> June 5, 2024, 10:58 a.m.	MNTM Reachable 172.18.90.221 -> June 5, 2024, 10:58 a.m.	MNTM Reachable 172.18.90.222 -> June 5, 2024, 10:58 a.m.	Spare Reachable 172.18.90.223 -> June 5, 2024, 10:59 a.m.
Spare Reachable	AEAMT - Workshop GF Reachable	AEAMT - CAD/CAM GF Reachable	AEAMT - CAD/CAM FF Reachable

Python 3.11 includes several new features and improvements over previous versions. Here are some of the key changes and enhancements introduced in Python 3.11:

1. **Performance Improvements**: Python 3.11 aims to be up to 10-60% faster than Python 3.10, thanks to various optimizations in the interpreter.
2. **Exception Groups and except***: This new syntax allows handling multiple exceptions simultaneously, which is especially useful for asynchronous and parallel code.
3. **Task and Exception Groups**: New classes taskgroup and exceptiongroup help manage multiple tasks and their exceptions more effectively in asynchronous programming.
4. **Variadic Generics**: This enhancement in type hinting allows defining generic types that can accept a variable number of type parameters.
5. **Enhanced Error Messages**: Error messages in Python 3.11 are more informative, with better context and suggestions for fixing errors.
6. **Fine-Grained Error Locations**: Python 3.11 provides more precise error locations, helping developers pinpoint the exact cause of issues in their code.
7. **New tomllib Module**: This module enables reading TOML configuration files, making it easier to work with this format in Python.
8. **Self Type in Class Bodies**: The Self type hint can be used to indicate that a method returns an instance of its class, improving type hinting in class definitions.
9. **New Syntax Features**: Python 3.11 introduces some minor syntax enhancements, such as pattern matching improvements and new syntax for type annotations.
10. **Deprecations and Removals**: As with any new Python release, some older features and modules are deprecated or removed to streamline the language and encourage best practices.

FLOWCHART

This flowchart illustrates the visualization of network IP-based data and the connection of many pages.



Coding Languages and Library

Back End:- Python:- Django Framework, Datetime, Pythonping.

Front-End:- HTML5, CSS3, JavaScript.

Database:- MySQL.

Django 4.2.4

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It was designed to help developers take applications from concept to completion as quickly as possible. Here are some key features and components of Django:

1. **MVC Framework**: Django follows the Model-View-Controller (MVC) architectural pattern, though it refers to it as Model-View-Template (MVT).
2. **ORM (Object-Relational Mapping)**: Django includes a powerful ORM that allows developers to interact with their database using Python code instead of SQL.
3. **Admin Interface**: Django provides an automatically generated admin interface for managing application data, which is highly customizable.
4. **Form Handling**: Django offers a robust form handling system, including form validation and rendering.
5. **Authentication**: Built-in authentication system with user management, password validation, and support for integrating third-party authentication systems.
6. **URL Routing**: Django uses a URL dispatcher based on regular expressions, making it easy to design clean, readable URLs.
7. **Templating Engine**: Django includes a templating engine that supports template inheritance and allows for separating design from business logic.
8. **Security**: Django provides various security features out of the box, including protection against common web vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
9. **Internationalization**: Django supports internationalization, allowing developers to create applications that support multiple languages and locales.
10. **Testing Framework**: Django comes with a comprehensive testing framework that makes it easy to write and run tests for your applications.
11. **Scalability**: Django is designed to handle high-traffic sites and can be scaled by using caching, database replication, and load balancing.

12. **Extensive Documentation**: Django has thorough and well-maintained documentation, making it easier for developers to learn and use the framework.

MySQL 8.0 includes numerous features and enhancements designed to improve performance, reliability, and usability. Here are some of the key features introduced in MySQL 8.0:

1. **Transactional Data Dictionary**: MySQL 8.0 uses a transactional data dictionary, which means that DDL operations are now atomic, crash-safe, and more efficient.
2. **UTF8MB4 as Default Charset**: The default character set is now utf8mb4, which supports a wider range of Unicode characters, including emojis.
3. **Window Functions**: MySQL 8.0 supports window functions (also known as analytic functions), which allow you to perform calculations across a set of table rows related to the current row.
4. **JSON Enhancements**: Improved JSON support with new functions for manipulating JSON data, allowing more complex and efficient operations.
5. **Security Improvements**: Enhanced security features, including roles for simplifying privilege management, better password management policies, and more secure defaults.

Stored procedures are not required in Django, but they can be useful in certain scenarios. Django primarily uses its ORM (Object-Relational Mapping) to interact with the database, which provides a high-level, Pythonic interface for querying and manipulating data. However, there are cases where stored procedures might be beneficial:

Python Backend Functions

I created several functions for the backend in views.py:-

@csrf_exempt:

This decorator is used to exempt a view from the Cross-Site Request Forgery (CSRF) protection that Django provides.

CSRF protection is important for preventing malicious users from performing actions on behalf of a logged-in user without their consent.

Exempting a view from CSRF protection can expose the application to certain types of attacks, so it should be used with caution.

@login_required:

This decorator ensures that only authenticated users can access the view.

If a user is not authenticated, they will be redirected to the login page.

It is important for protecting views that should not be accessible to anonymous users.

Functions

alert:

Handles POST requests to record the status of an IP address (whether it is reachable or not).

It uses the ping function to check the IP address and saves the result in the database.

Renders ui-alerts.html with the data and status information.

For GET requests, it filters events based on optional start_date and end_date parameters and renders them.

time:

Handles GET requests to filter events based on start_date and end_date and count the status of the events.

Renders ui-card.html with the event data and counts of different statuses.

Log:

Handles POST requests for user login.

Checks user credentials against the custom user model.

If valid, logs in the user and redirects to the 'desktop' page.

If invalid, renders the login page with an error message.

Regiss:

Handles POST requests for user registration.

Validates the uniqueness of the username and email.

Creates a new user in the custom user model and responds with a success message.

check_network_connection:

Pings a specific IP address to check the network connection.

Returns "Connected" if the ping is successful, otherwise "Connected".

desktop:

Handles various operations, including fetching and displaying data for the authenticated user.

Checks the network connection and pings IP addresses from a list.

Renders index2 - Copy.html with various data and status information.

Logt:

Logs out the user and redirects to the login page.

pie2:

Handles GET requests to display the latest IP addresses and their statuses.

Renders ui-buttons.html with the data.

alert2:

Similar to the alert function, but with slight differences in handling the status and rendering the data.

Renders ui-alerts.html with the updated data and status information.

ip_save:

Handles POST requests to add, update, or delete IP address records.

Validates the IP address and performs the corresponding database operations.

Renders record.html with the updated data and status messages.

read_csv_data:

Reads input patterns and responses from a CSV file to create a list of pairs for a chatbot.

chat:

Handles user messages and responds with either IP address information or a chatbot response.

Uses the Chat class to generate responses based on predefined patterns and responses from a CSV file.

DESKTOP:

The provided code is a Django view function named desktop, which is intended to be executed upon accessing a particular endpoint in a web application. Here's a breakdown of the function, explaining its purpose and workflow:

@login_required decorator:

This decorator ensures that the view can only be accessed by authenticated users. If a user is not logged in, they will be redirected to the login page.

Initialization and Data Fetching:

now = datetime.now() and now = date.today(): These lines fetch the current date and time.

user_id = request.user.id: This retrieves the ID of the currently logged-in user.

data = details.objects.filter(id_cust_id=user_id): This fetches all records from the details table where the id_cust_id matches the logged-in user's ID.

Various other queries are executed to fetch data from the details model ordered by their id in descending order, and to get the latest and specific records.

Data Processing:

type_counts = dict(Counter(fruit.Ip for fruit in data)): This creates a dictionary that counts occurrences of each IP address in the data queryset.

labels = [str(Ip) for Ip in last_5_addresses] and dat = [1] * len(last_5_addresses): These lines create lists for labels and data points, respectively, for the most recent IP addresses.

Network and Server Operations:

The user's IP address is fetched using request.META.get('REMOTE_ADDR').

The function attempts to ping a specific IP address (172.18.90.200) and sets the state based on the success of the ping.

The server's IP address is determined using socket.gethostbyname(socket.gethostname()).

Handling POST Requests:

If the request method is POST, the function processes form data. If specific inputs are present, it records the current time; otherwise, it asks the user to type a specific input.

Depending on the network state, it checks the network connection and processes ping requests for IP addresses from the csv_1 model.

Database Updates:

If the network is connected, it iterates through IP addresses from the csv_1 model, pings them, and records the results in the details model.

The request.session['current_index'] is used to keep track of the current IP address being processed to allow for pagination or iterative checks across requests.

Context Preparation:

A context dictionary is prepared with all the fetched and processed data to be passed to the template for rendering.

This context includes various data points like user-specific details, network status, and the results of IP address pings.

Rendering the Template:

The context dictionary is passed to the template index2 - Copy.html using the render function to generate the HTML response for the user.

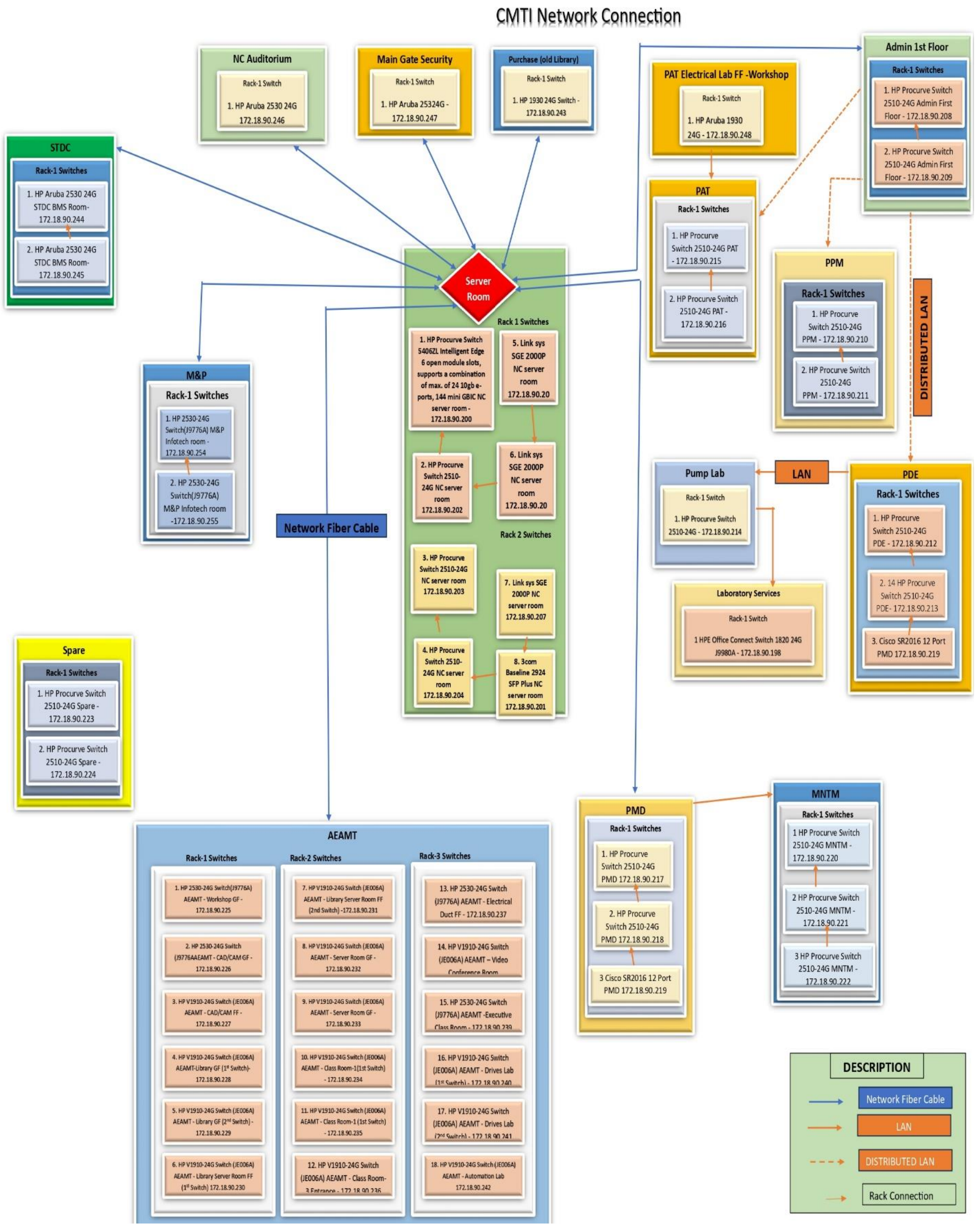
Usage and Considerations

The use of @csrf_exempt should be reviewed carefully to ensure it is necessary and does not introduce security vulnerabilities.

The @login_required decorator ensures that sensitive operations are protected and accessible only to authenticated users.

Proper error handling and validation are essential to prevent unexpected issues and enhance user experience.

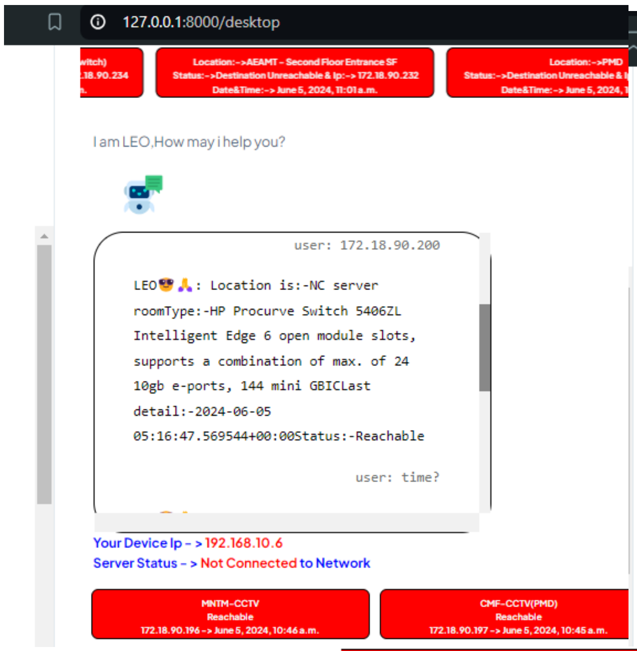
CMTI NETWORK Connection



CMTI Network Connection
NC Auditorium
Rack-1 Switch
1. HP Aruba 2530 24G 172.18.90.246
Main Gate Security
Rack-1 Switch
1. HP Aruba 25324G - 172.18.90.247
Purchase (old Library)
Rack-1 Switch
1. HP 1930 24G Switch - 172.18.90.243
PAT Electrical Lab FF -Workshop
Rack-1 Switch
1. HP Aruba 1930 24G - 172.18.90.248
Admin 1st Floor
Rack-1 Switches
1. HP Procurve Switch 2510-24G Admin First Floor - 172.18.90.208
2. HP Procurve Switch 2510-24G Admin First Floor - 172.18.90.209
STDC
Rack-1 Switches
1. HP Aruba 2530 24G STDC BMS Room- 172.18.90.244
2. HP Aruba 2530 24G STDC BMS Room- 172.18.90.245
PAT
Rack-1 Switches
1. HP Procurve Switch 2510-24G PAT - 172.18.90.215
2. HP Procurve Switch 2510-24G PAT - 172.18.90.216
PPM
Rack-1 Switches
1. HP Procurve Switch 2510-24G PPM - 172.18.90.210
2. HP Procurve Switch 2510-24G PPM - 172.18.90.211
DISTRIBUTED LAN
Server Room
Rack 1 Switches
1. HP Procurve Switch 5406ZL Intelligent Edge 6 open module slots, supports a combination of max. of 24 10gb e-ports, 144 mini GBIC NC server room - 172.18.90.200
5. Link sys SGE 2000P NC server room 172.18.90.20
2. HP Procurve Switch 2510-24G NC server room 172.18.90.202
6. Link sys SGE 2000P NC server room 172.18.90.20
Rack 2 Switches
3. HP Procurve Switch 2510-24G NC server room 172.18.90.203
7. Link sys SGE 2000P NC server room 172.18.90.207
4. HP Procurve Switch 2510-24G NC server room 172.18.90.204
8. 3com Baseline 2924 SFP Plus NC server room 172.18.90.201
M&P
Rack-1 Switches
1. HP 2530-24G Switch(J9776A) M&P Infotech room - 172.18.90.254
2. HP 2530-24G Switch(J9776A) M&P Infotech room -172.18.90.255
Network Fiber Cable
Pump Lab
Rack-1 Switch
1. HP Procurve Switch 2510-24G - 172.18.90.214
LAN
PDE
Rack-1 Switches
1. HP Procurve Switch 2510-24G PDE - 172.18.90.212
2. 14 HP Procurve Switch 2510-24G PDE- 172.18.90.213
3. Cisco SR2016 12 Port PMD 172.18.90.219
Laboratory Services
Rack-1 Switch
1 HPE Office Connect Switch 1820 24G J9980A - 172.18.90.198
Spare
Rack-1 Switches
1. HP Procurve Switch 2510-24G Spare - 172.18.90.223
2. HP Procurve Switch 2510-24G Spare - 172.18.90.224
PMD
Rack-1 Switches
1. HP Procurve Switch 2510-24G PMD 172.18.90.217
2. HP Procurve Switch 2510-24G PMD 172.18.90.218
3 Cisco SR2016 12 Port PMD 172.18.90.219
MNTM
Rack-1 Switches
1 HP Procurve Switch 2510-24G MNTM - 172.18.90.220
2 HP Procurve Switch 2510-24G MNTM - 172.18.90.221
3 HP Procurve Switch 2510-24G MNTM - 172.18.90.222
AEAMT
Rack-1 Switches
Rack-2 Switches
Rack-3 Switches
1. HP 2530-24G Switch(J9776A) AEAMT - Workshop GF - 172.18.90.225
2. HP 2530-24G Switch (J9776AAEAMT - CAD/CAM GF - 172.18.90.226
3. HP V1910-24G Switch (JE006A) AEAMT - CAD/CAM FF - 172.18.90.227
4. HP V1910-24G Switch (JE006A) AEAMT-Library GF (1st Switch)- 172.18.90.228
5. HP V1910-24G Switch (JE006A) AEAMT - Library GF (2nd Switch) - 172.18.90.229
6. HP V1910-24G Switch (JE006A) AEAMT - Library Server Room FF (1st Switch) 172.18.90.230
7. HP V1910-24G Switch (JE006A) AEAMT - Library Server Room FF (2nd Switch) -172.18.90.231
8. HP V1910-24G Switch (JE006A) AEAMT - Server Room GF - 172.18.90.232
9. HP V1910-24G Switch (JE006A) AEAMT - Server Room GF - 172.18.90.233
10. HP V1910-24G Switch (JE006A) AEAMT - Class Room-1(1st Switch) - 172.18.90.234
11. HP V1910-24G Switch (JE006A) AEAMT - Class Room-1 (1st Switch) - 172.18.90.235
12. HP V1910-24G Switch (JE006A) AEAMT - Class Room-3 Entrance - 172.18.90.236
13. HP 2530-24G Switch (J9776A) AEAMT - Electrical Duct FF - 172.18.90.237
14. HP V1910-24G Switch (JE006A) AEAMT – Video Conference Room
15. HP 2530-24G Switch (J9776A) AEAMT -Executive Class Room - 172.18.90.239
16. HP V1910-24G Switch (JE006A) AEAMT - Drives Lab (1st Switch) - 172.18.90.240
17. HP V1910-24G Switch (JE006A) AEAMT - Drives Lab (2nd Switch) - 172.18.90.241
18. HP V1910-24G Switch (JE006A) AEAMT - Automation Lab 172.18.90.242
DESCRIPTION
Network Fiber Cable
LAN
DISTRIBUTED LAN
Rack Connection

LEO CHATBOT

LEO chatbot that is intended to communicate with users via text over a network is known as a network chatbot. These chatbots can be used for a number of tasks, including providing technical help, and IP information. The following are some essential attributes and elements of network chatbots:



127.0.0.1:8000/desktop

witch)
.18.90.234
n.

Location:-->AEAMT - Second Floor Entrance SF
Status:-->Destination Unreachable & Ip:-> 172.18.90.232
Date&Time:-> June 5, 2024, 11:01 a.m.

Location:-->PMD
Status:-->Destination Unreachable & I
Date&Time:-> June 5, 2024, 1

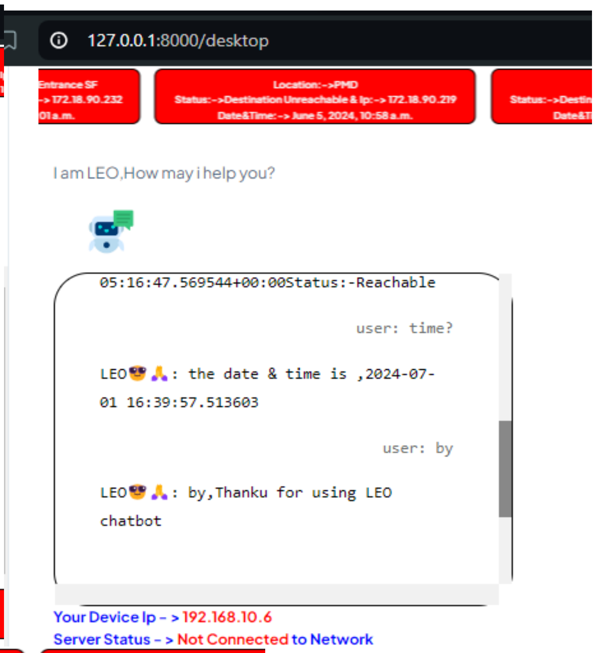
I am LEO,How may i help you?

```
                         user: 172.18.90.200

LEO😎🙏: Location is:-NC server
roomType:-HP Procurve Switch 5406ZL
Intelligent Edge 6 open module slots,
supports a combination of max. of 24
10gb e-ports, 144 mini GBICLast
detail:-2024-06-05
05:16:47.569544+00:00Status:-Reachable

                              user: time?
```

Your Device Ip - > 192.168.10.6
Server Status - > Not Connected to Network

MNTM-CCTV
Reachable
172.18.90.196 -> June 5, 2024, 10:46 a.m.

CMF-CCTV(PMD)
Reachable
172.18.90.197 -> June 5, 2024, 10:45 a.m.



127.0.0.1:8000/desktop

Entrance SF
-> 172.18.90.232
01 a.m.

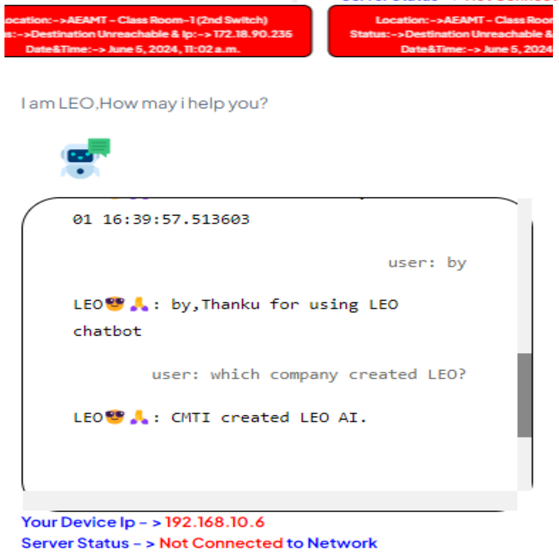
Location:-->PMD
Status:-->Destination Unreachable & Ip:-> 172.18.90.219
Date&Time:-> June 5, 2024, 10:58 a.m.

Status:-->Destin
Date&Ti

I am LEO,How may i help you?

```
05:16:47.569544+00:00Status:-Reachable

                              user: time?

LEO😎🙏: the date & time is ,2024-07-
01 16:39:57.513603

                                user: by

LEO😎🙏: by,Thanku for using LEO
chatbot
```

Your Device Ip - > 192.168.10.6
Server Status - > Not Connected to Network



ocation:-->AEAMT - Class Room-1 (2nd Switch)
s:-->Destination Unreachable & Ip:-> 172.18.90.235
Date&Time:-> June 5, 2024, 11:02 a.m.

Location:-->AEAMT - Class Roo
Status:-->Destination Unreachable &
Date&Time:-> June 5, 2024

I am LEO,How may i help you?

```
01 16:39:57.513603

                                user: by

LEO😎🙏: by,Thanku for using LEO
chatbot

          user: which company created LEO?

LEO😎🙏: CMTI created LEO AI.
```

Your Device Ip - > 192.168.10.6
Server Status - > Not Connected to Network

Context Management:

- Maintains context across a conversation to provide coherent and relevant responses.
- Can handle single conversations effectively.

Multimodal Interaction:

- Supports multiple forms of input and output, such as text.
- Enhances user experience by allowing flexibility in how users interact.

Components

1. **User Interface (UI):**
   - The front-end interface where users interact with the chatbot.
   - Could be a chat window on a website.
2. **Backend Services:**
   - Processes user inputs and generates responses.
   - Includes NLP engines, Static integration layers.
3. **Database:**
   - Stores user IP data, and fetching other relevant information.
   - Used for context management and personalization.
4. **API Layer:**
   - Connects the chatbot with external services and databases.
   - Facilitates tasks like fetching data, or general conversation.