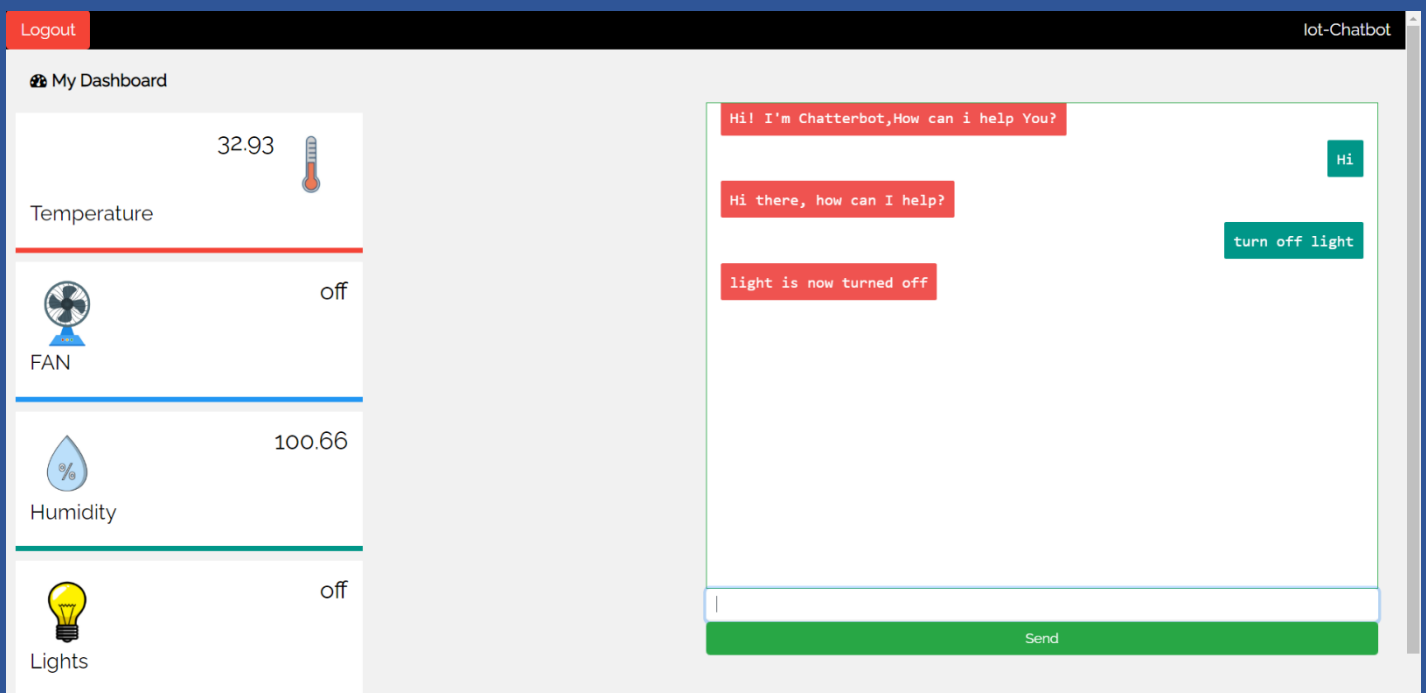




IoT CHATBOT

PROJECT REPORT



CONTENTS

- Abstract
- Motivation
- Networking
- Workflow
- Applications
- Limitations
- Future Improvements
- Installation
- Team Members
- Mentors
- References and Links

I. Abstract

Internet of Things or **IoT** refers to a network of devices that are interconnected and facilitate exchange of data with each other and other systems. The project presents a chatbot integrated with an IoT (Internet of Things) network of devices in context of home automation. The chatbot acts as an interface for the user to view real time data and live status of home appliances as well as control them by giving commands.

II. Motivation

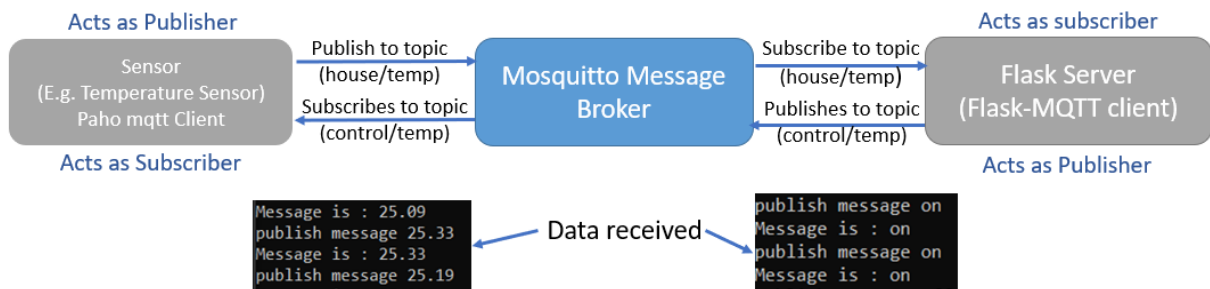
Using IoT to allow communication between various home appliances, security systems, lighting, entertainment systems and sensors will enhance lives of people and make homes smarter. Controlling home device via chatbot remotely is convenient, easy to use and time saving.

III. Networking

Creation of IoT Network Using MQTT (Message Queuing Telemetry Transport)

- Adding multiple clients requires threading and has overhead if we have many IoT applications. So, we used MQTT protocol which is apt for IoT network. It involves publishers which publish a message to a topic to message broker. The subscriber is program which retrieves this data from message broker by subscribing.
- The IoT devices act as publishers and send their real time data to respective topics. For e.g. Temperature sensor publishes to a topic house/temp. We used Paho MQTT module in python library to create these IoT clients (sensors and IoT devices). They continuously send their data to their respective topic to the broker in small time intervals.
- It uses Mosquitto as local message broker whereas uses HiveMQ and Azure IoT hub for online message broker services act as a broker between publisher and subscriber.
- This data is then received by flask server which acts as subscriber in this case. MQTT has been implemented in flask with the help of flask-mqtt module in python.

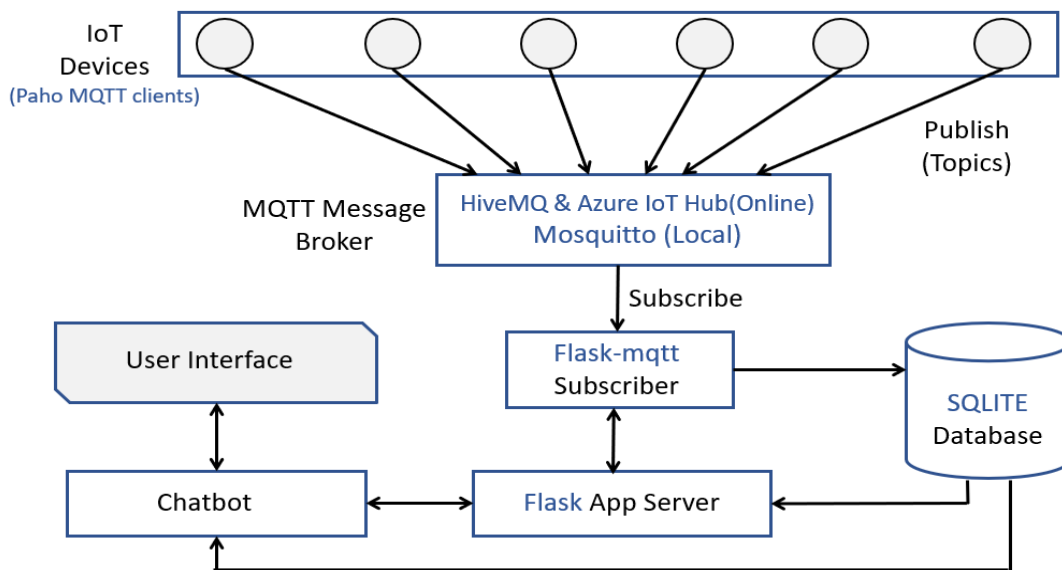
- In the communication from chatbot to IoT devices, the flask server act as publisher, publishing to topics (e.g. control/temp) and IoT devices act as subscriber.



Chatbot

- We used Flask-python Framework for creating Chatbot and Dashboard webapp.
- API calls are made to transfer data to chatbot from python app.
- We also added auth login system for security measures
- Sqlite3 as database to store the data as it is stored for very short interval of time in broker. This data can be used for analysis and making ML models.
- Microsoft Azure App services act as Cloud based hosting platform**

IV. Workflow



IoT-CHATBOT ARCHITECTURE

V. Applications

- Healthcare Sector
- Home Automation
- Automotive IoT
- Industrial Automation
- Wearables

VI. Limitations

- Currently our broker/server can handle only upto 8 clients/IoT devices.

VII. Future Improvements

- Incorporation of ML model for better user experience which can give best suggestions to user by analysing previous user input and automate the system.
- Allowing connection of more clients and IoT devices. Work on increasing the capacity of a server to handle more clients at the same time
- Use Raspberry Pi and Sensors so that we can deal with real-time data.

VIII. Installation

1) By Using Hosted Website

- Download the all files from folder sensors
- Install Python
- Install Paho-mqtt
 - `pip install paho-mqtt`
- Run each file in a separate terminal
 - `python pub_temp.py`
 - `python pub_humidity.py`
 - `python pub_fan.py`
 - `python pub_light.py`

- Open the link lotChatbot (<http://iotchatbot.azurewebsites.net/>)
- Use following credentials
 - Username: test
 - Password : 1234

2) By Using Local Host

- Download the code as a zip file or clone the repo by using following command in git bash
 - `git clone https://github.com/marsiitr/loT-ChatBot.git`
- create venv (optional)
- Install required python-packages
 - `pip install -r requirements.txt`
- Go to lotchatbot directory in command line and run flask app
 - `python app.py`
- Go to sensors directory and run each file in a separate terminal
 - `python pub_temp.py`
 - `python pub_humidity.py`
 - `python pub_fan.py`
 - `python pub_light.py`
- Open the link in browser
 - `http://127.0.0.1:5000/`
- Use following credentials
 - Username: test
 - Password : 1234

IX. Team Members

1. Khushi Kumawat
2. Kushagra Agarwal
3. Nagesh Bansal
4. Vaishnavi Gupta

X. Mentors

1. Pradnesh Chavan
2. Sanjeev Krishnan R.

XI. Hosted Chatbot Link

<http://iotchatbot.azurewebsites.net/>

XII. References

- MQTT Basics
<https://medium.com/python-point/mqtt-basics-with-python-examples-7c758e605d4>
- Flask Documentation
<https://flask.palletsprojects.com/en/2.0.x/>
- IoT Architecture
<https://www.avsystem.com/blog/what-is-iot-architecture/>
- Azure IoT Hub
<https://docs.microsoft.com/en-in/azure/iot-hub/>