PROJECT
REPORT

SUBMITTED BY

MD. SHAHARIAR HASSAN RONOK
ROLL : 1710046
DEPT. : ELECTRICAL & COMPUTER ENGINEERING
RAJSHAHI UNIVERSITY OF ENGINEERING AND TECHNOLOGY

INDEX

Key Words: IoT(Internet of Things), Kodular, Firebase, ThingSpeak, Adafruit I/O, Android application, Raspberry pi, Rest API, Postman, Mosquito, MQTT, Encryption, serial communication, Publisher and subscriber.

Mobile Application Based Industry Sub Unit Automation System and Auto Reporting System Using IOT Devices

Abstract:
As the technology is advancing, we are witnessing automation in each and every field. A fully autonomous office is going to be the future. People often wish to have automatic control over various electrical appliances in office like fan, light, computer and microwave oven. This project presents a solution which helps in accomplishing the task successfully. A universal switch has been realized using ESP 8266 12E, Arduino Uno Atmel Microcontroller, Android application and GSM modem along with fire and human sensors the whole of which constitutes Office Automation System. The main objective is to design and implement an Office Automation System using IoT(Internet of Things) that is capable of controlling and automating most of the office appliances through an easily manageable android application.

.

Software requirements for app and database:

I. Kodular

II. Firebase

III. ThingSpeak

IV. Adafruit I/O

V. Excel sheet

VI. Flutter

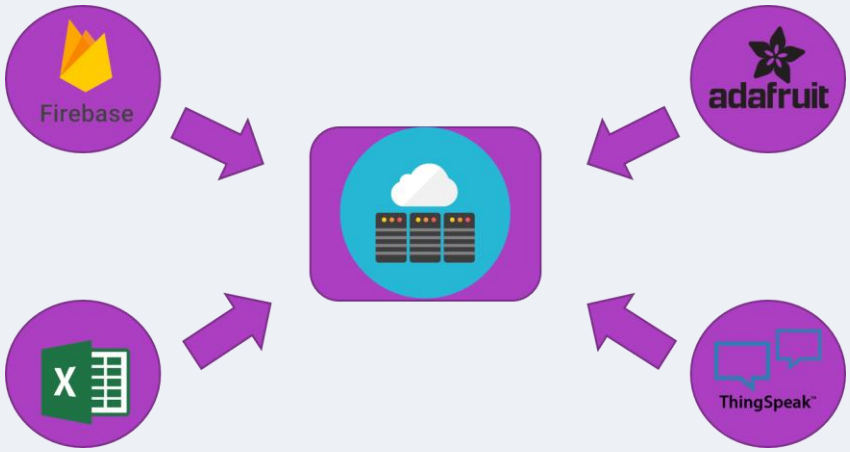
Features:

1. Switch of the devices (must login first)
2. Show status of the devices (numerical value/ logical condition/GUI)
3. Show graph of the production rate
4. Chat and file share (Testing phase)

Extra features in future:

1. Strong security
2. Chat bot
3. Voice assistant
4. Connect camera
5. Auto report creation

Connection of the databases:



Here Firebase, Adafruit, Thingspeak are runtime databases. Firebase is used for user's identity and personal information collection. Adafruit is used for controlling online based IoT devices. Thingspeak is used for showing the graphical representation of the collected data. And lastly MS Excel is used for collecting all the numerical data with time and date for local database as report.

Hardware and software requirements for IoT devices:

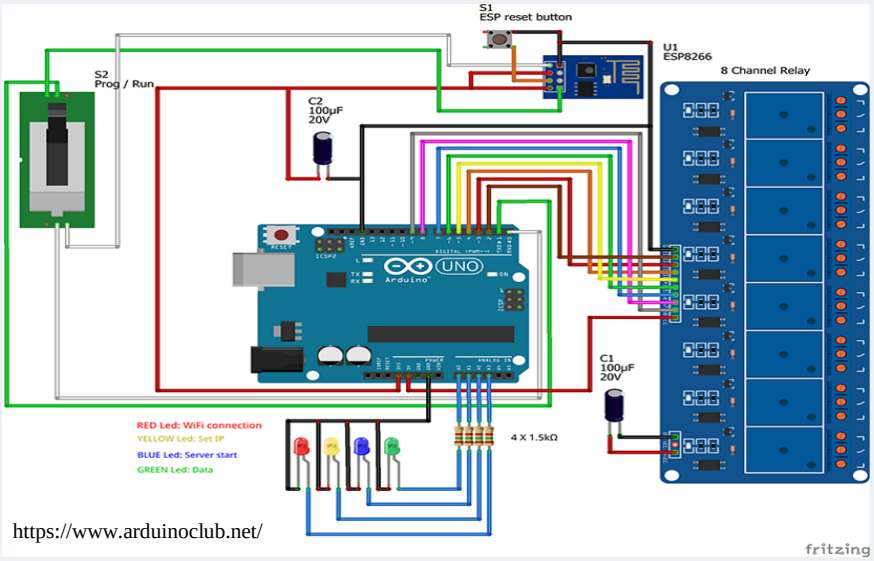
I. ESP8266-12E

II. Arduino uno

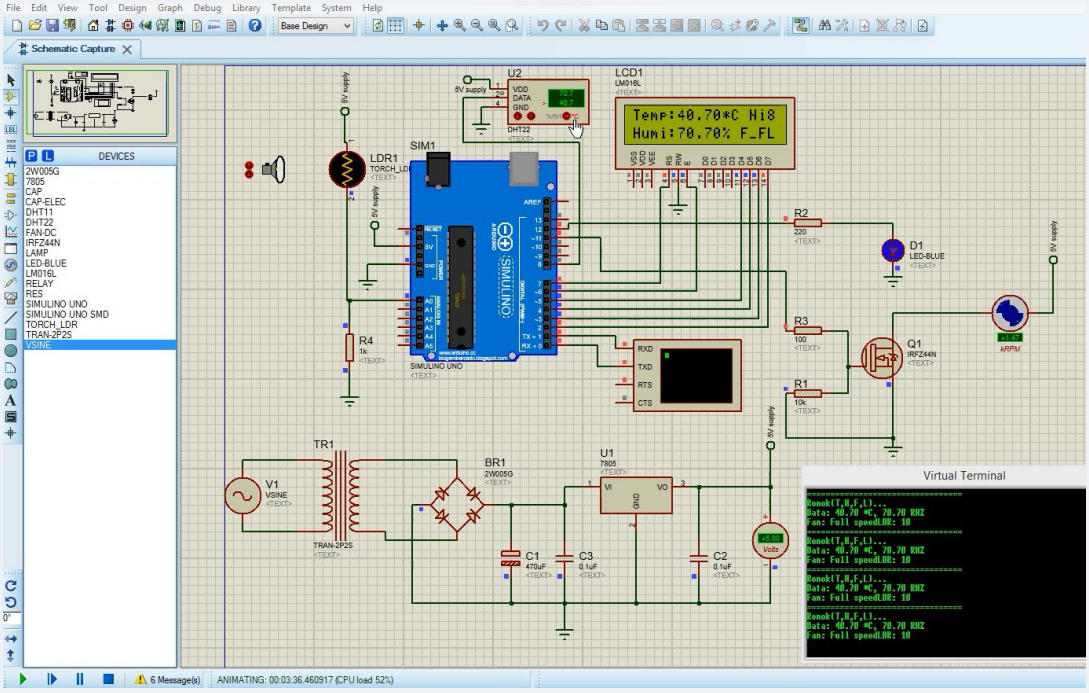
III. Relay module (12V)

IV. Arduino IDE (Software)

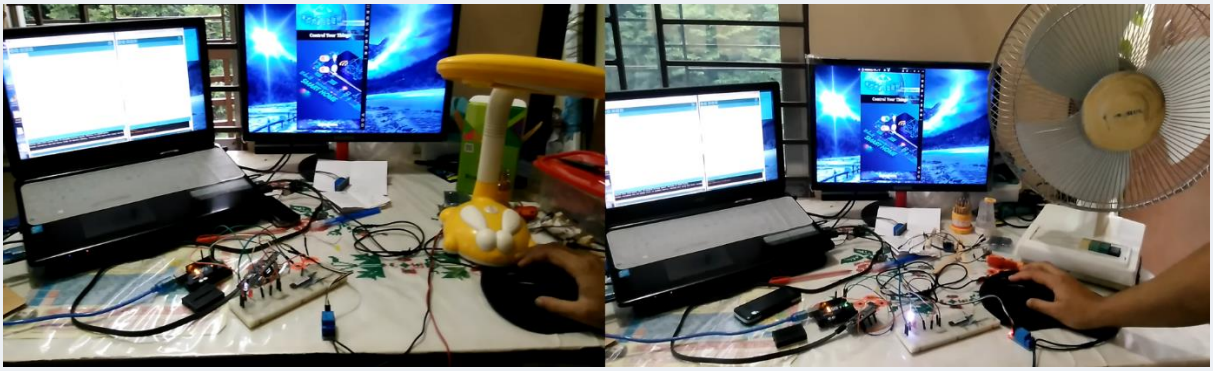
Demo diagram connection of IoT devices



Hardware Simulation:



System demo:



Main working of this App:

I. record data of the machineries and environment

(like temperature, pressure, air condition, Molecule (O2, CO2...) amount in the environment)

II. control devices using the mobile application

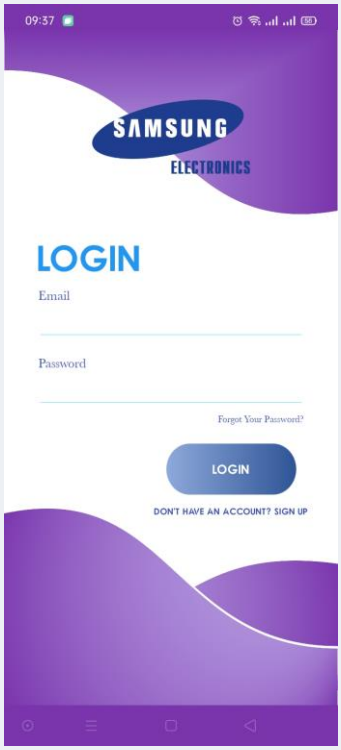
III. Show status of the devices and provide notification when there emergency warning occur.

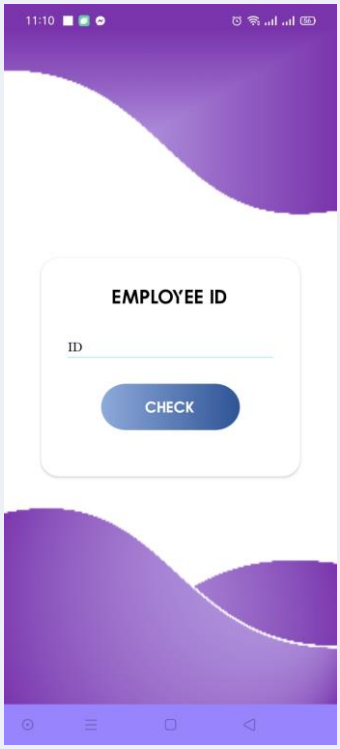
IV. Chatting between team members

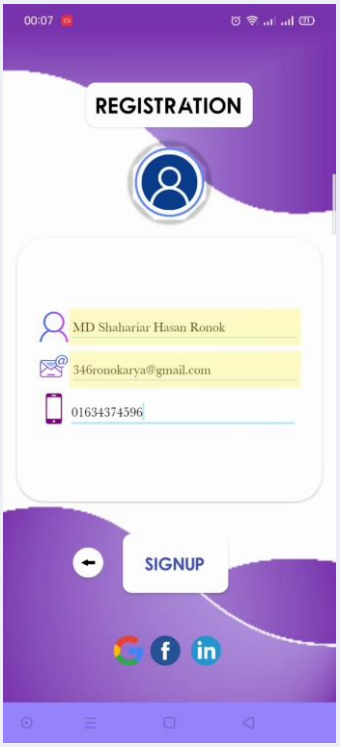
V. File sharing and report making

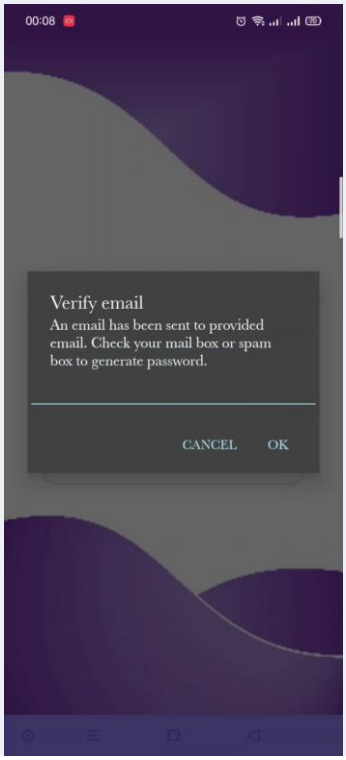
RIoT(app) Review:

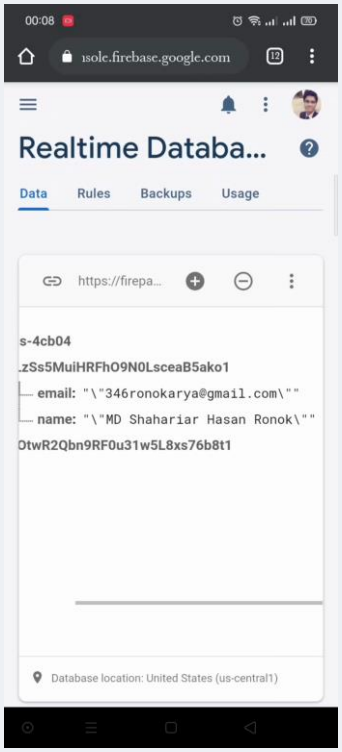
I. Signup and Login:

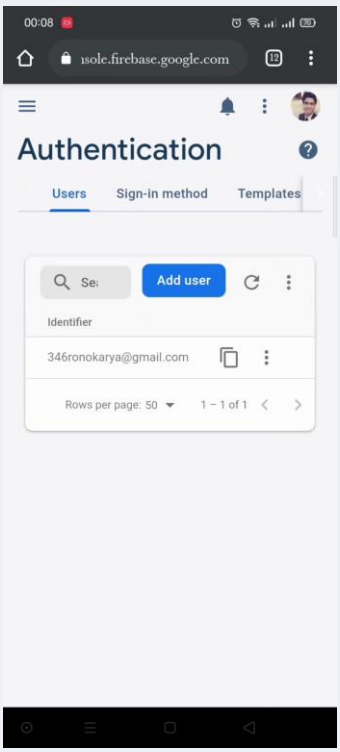




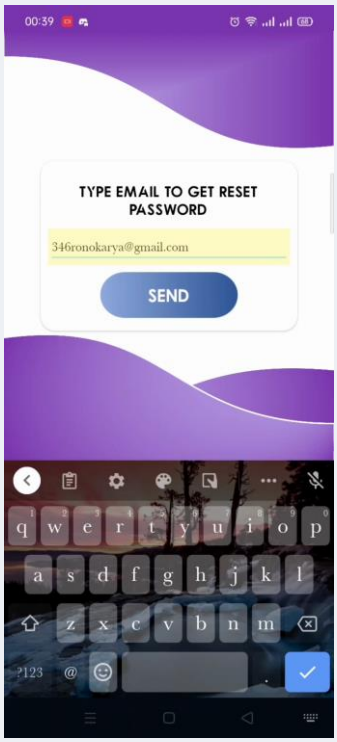


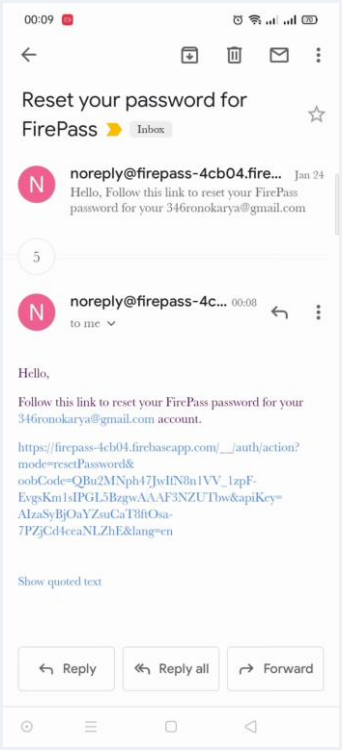


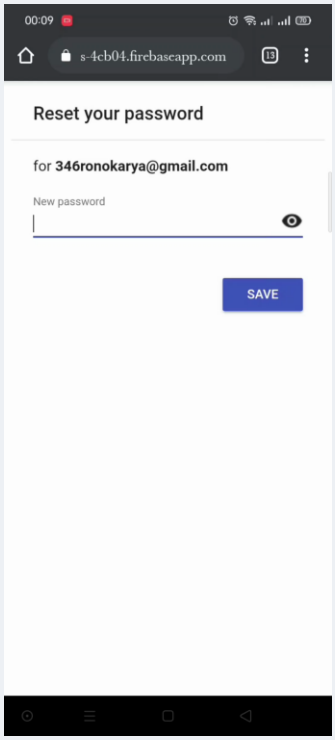




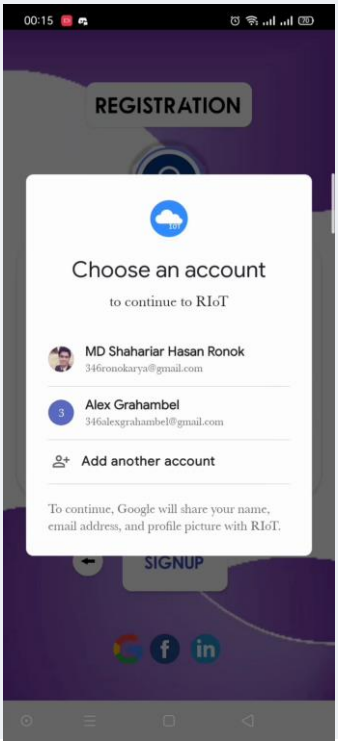
II. Forget password:



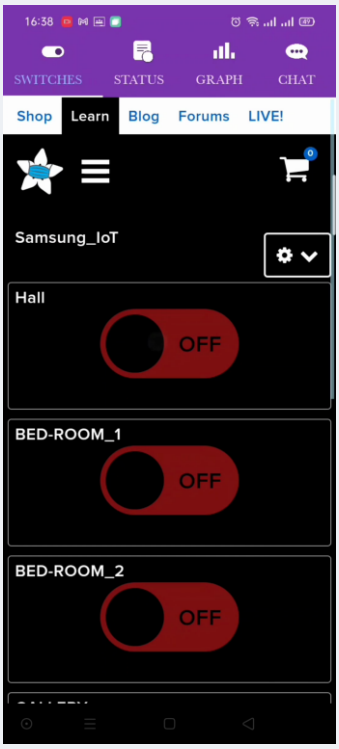


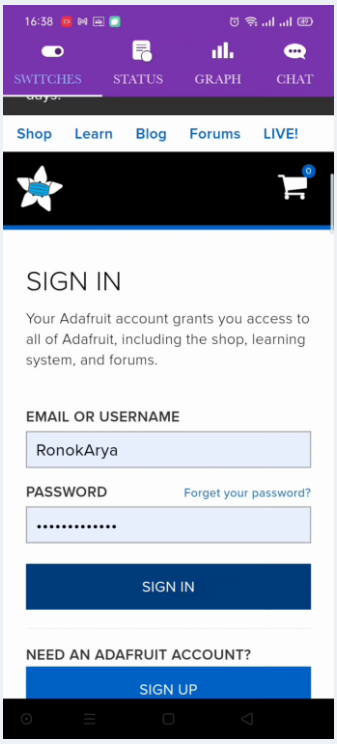


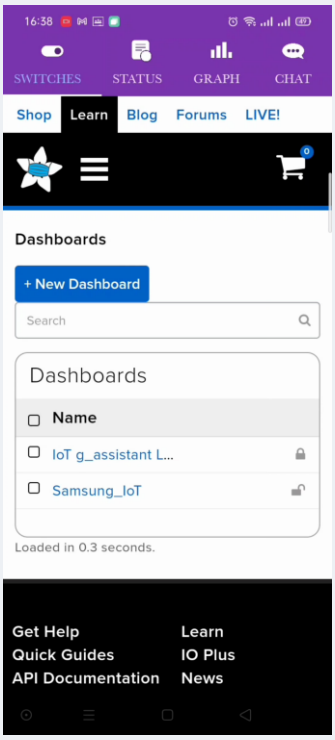
III. Google signup:

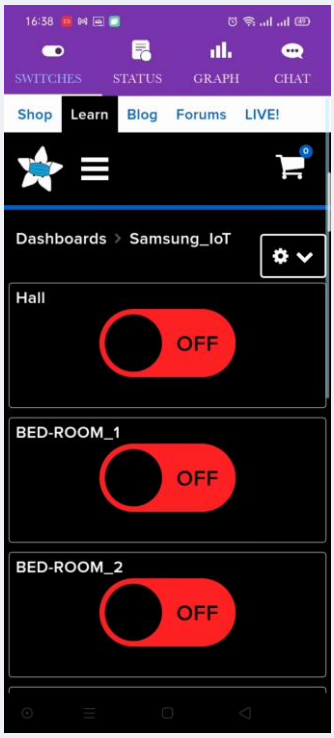


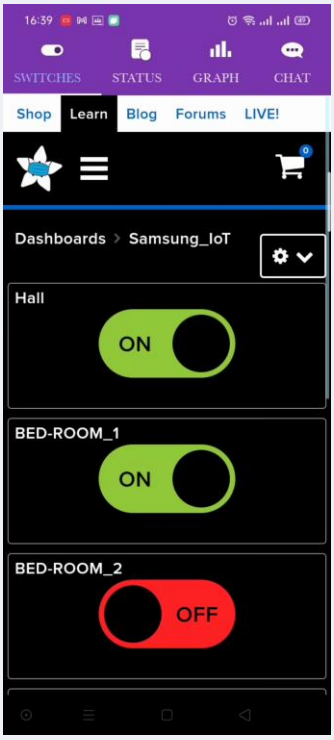
IV. Secure button:



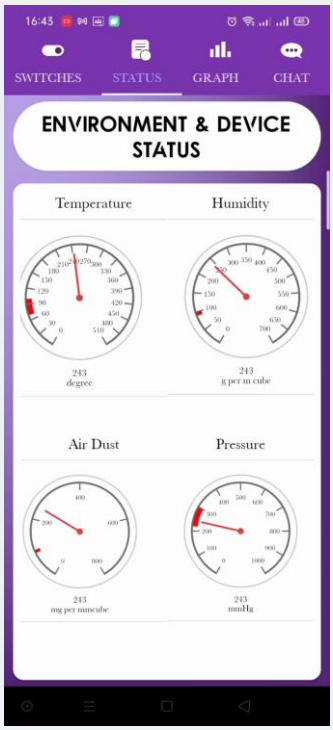


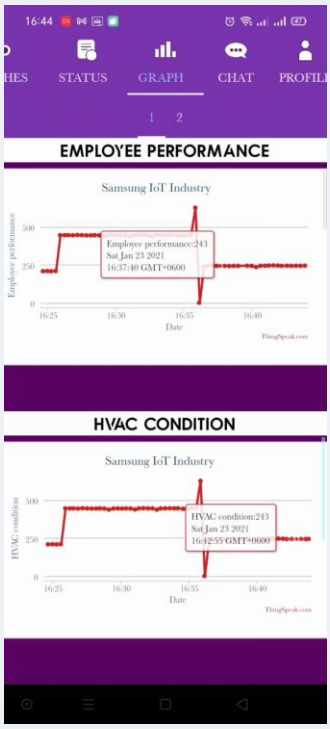




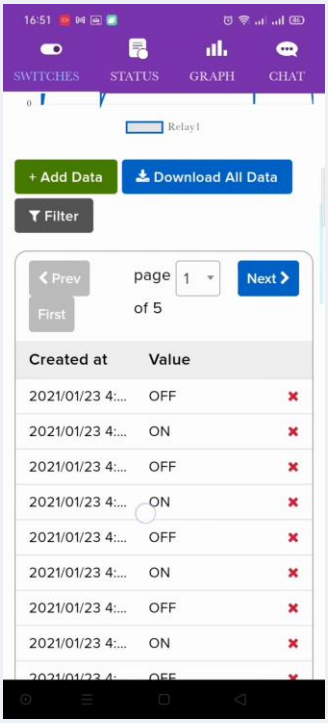


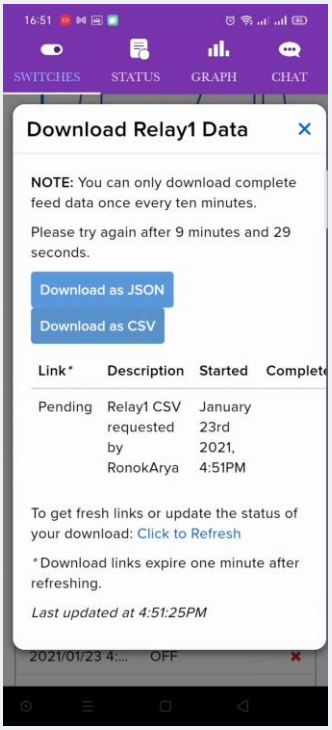
V. Graphical view and time and date data storage record view:





VI. Database records:

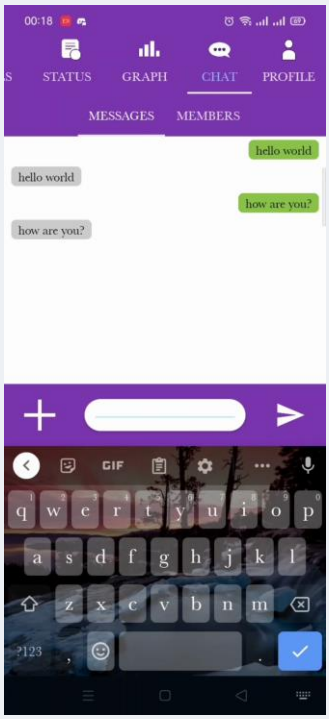


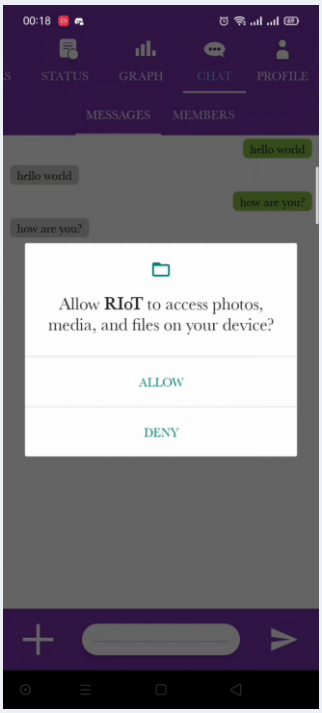


	A	B	C	D	E	F
184	0EMDQYQ	OFF	1441984	2021-01-01	14:40:18	UTC
185	0EMDQYR	ON	1441984	2021-01-01	14:40:21	UTC
186	0EMDQZX	OFF	1441984	2021-01-01	14:42:22	UTC
187	0EMZ5VV	ON	1441984	2021-01-23	06:40:52	UTC
188	0EMZ5VV	OFF	1441984	2021-01-23	06:40:54	UTC
189	0EMZ5VZ	ON	1441984	2021-01-23	06:41:07	UTC
190	0EMZ5W0	OFF	1441984	2021-01-23	06:41:09	UTC
191	0EMZ5W6	ON	1441984	2021-01-23	06:41:29	UTC
192	0EMZ5XN	OFF	1441984	2021-01-23	06:44:04	UTC
193	0EMZ5XQ	ON	1441984	2021-01-23	06:44:11	UTC
194	0EMZ5XSF	OFF	1441984	2021-01-23	06:44:16	UTC
195	0EMZ5XYH	ON	1441984	2021-01-23	06:44:33	UTC
196	0EMZ5XYX	OFF	1441984	2021-01-23	06:44:34	UTC
197	0EMZ5YRZ	ON	1441984	2021-01-23	06:46:00	UTC
198	0EMZ5YSJ	OFF	1441984	2021-01-23	06:46:02	UTC
199	0EMZ9T8K	ON	1441984	2021-01-23	10:21:48	UTC
200	0EMZ9T8K	ON	1441984	2021-01-23	10:21:48	UTC
201	0EMZ9TAL	OFF	1441984	2021-01-23	10:21:54	UTC
202	0EMZA42M	ON	1441984	2021-01-23	10:38:57	UTC
203	0EMZA435	OFF	1441984	2021-01-23	10:38:59	UTC
204	0EMZA43M	ON	1441984	2021-01-23	10:39:01	UTC
205	0EMZA45E	OFF	1441984	2021-01-23	10:39:06	UTC
206	0EMZA48F	ON	1441984	2021-01-23	10:39:16	UTC
207	0EMZA4AS	OFF	1441984	2021-01-23	10:39:22	UTC
208	0EMZA4AI	ON	1441984	2021-01-23	10:39:24	UTC
209	0EMZA4DS	OFF	1441984	2021-01-23	10:39:32	UTC
210	0EMZA79M	ON	1441984	2021-01-23	10:44:35	UTC
211	0EMZA7A:	OFF	1441984	2021-01-23	10:44:36	UTC
212	0EMZA7F1	ON	1441984	2021-01-23	10:44:52	UTC
213	0EMZA7FC	OFF	1441984	2021-01-23	10:44:54	UTC
214	0EMZA7F2	ON	1441984	2021-01-23	10:44:56	UTC

Relay1-20210123-1051

VII. Chat between teammates and file sharing:





VIII. Profile:

Benefits from the project:

I. Safety for the industry
II. Easy for monitoring
III. Increase production rate with low cost
IV. Easy for employee in large company to communicate with each group members
V. Fault detect and early notify.

Drawback of This App:

- Slow
- Can't hold the initial value of the device at switching of the devices
- Resize of the graphical meter and graph
- Only picture can be sent no other files
- Again code to add another system
- Store in cloud database so data can be stored for few months

Work done & work not done:

Done	Not Done
1. Login & Signup(password must be greater than 6 digits, must contain numbers and email should have '@') 2. Forget password 3. Email verification 4. Google Signup	1. Sign up value automatically show in profile 2. Edit profile 3. Testing chatting between members 4. Show stored file in profile 5. Notification while the parameter excess the limit 6. Using same data(as my DHT sensor burn out)

Software improvement

- Using Rest API and testing it using 'Postman'
- Using Mosquito and creating own server in Linux for MQTT protocol
- Using flutter and React for more GUI and user friendly environment
- Creating own dynamic database like firebase, Adafruit IO, Thingspeak
- Encryption of the company data for better security

Structure Improvement

- Using Raspberry pi to create own server and fast communications
- Using serial communication for add more systems
- Using portable system to connect more devices without re-coding
- If Publish and subscriber face problem then use indicator light for notifications for that system

Project outcome:

It will be much beneficial for developing country like Bangladesh to step in automation system and increase the production. Some small company or entrepreneur company will also find it beneficial from starting.

RIoT app will bring a great help for the automation industry to monitor and maintenance of the devices via IoT.