# INDEX

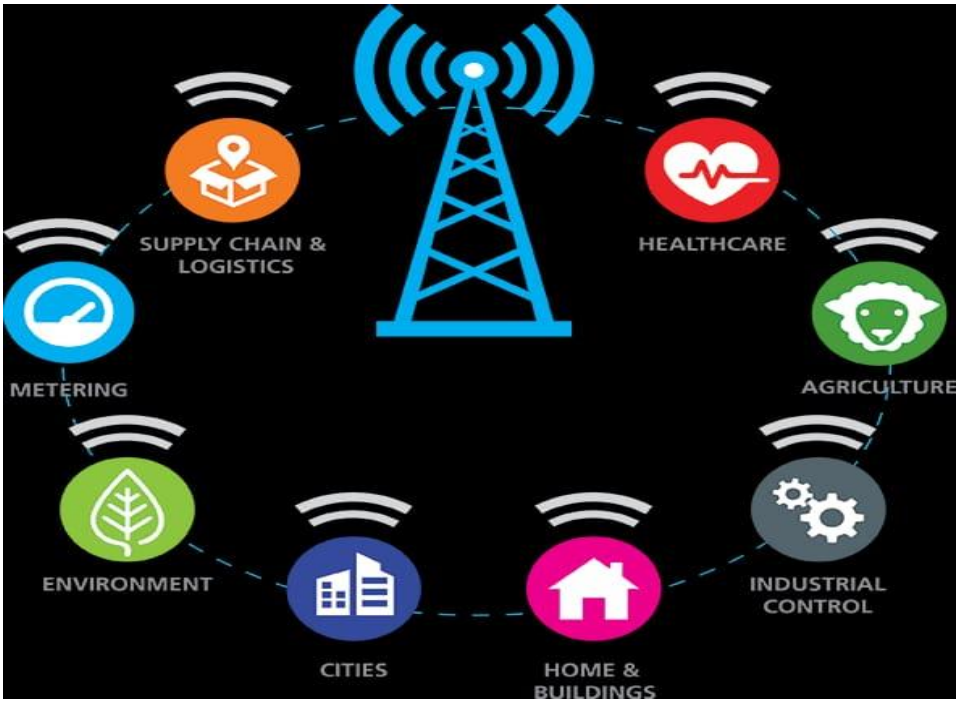
# 1. INTRODUCTION TO JSPYN

# INTRODUCTION

The world has changed after the evolution of term called Automation. In every sector, automation is reducing human labor. Firstly, introduced in industries and now most sectors rely on the automation system. Development of advanced home automation systems has a huge opportunity in the present time. Again Internet of Things (IoT) is going to be a new era of technology in the future. It is expected that 30 billion devices will be connected within 2020. So we decided through our project to prepare for future technologies.

## What is IoT?

IoT is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. IoT evolved from machine-to-machine (M2M) communication, i.e., machines connecting to each other via a network without human interaction. M2M refers to connecting a device to the cloud, managing it and collecting data.



# IoT Based Home Automation

Today automation innovation is making life simpler and less demanding in all spheres. Home automation is a modern technology that modifies your home to perform different sets of task automatically. Automatic frameworks are being favored over manual frameworks. No wonder, home automation in India is already the buzz word, especially as the wave of second generation homeowners grows, they want more than shelter, water, and electricity. Home automation leads to smart homes where comfort and convenience is a major advantage. The smart home/office gadgets interact, seamlessly and securely; control, monitor and improve accessibility, from anywhere across the globe. These smart automation devices happen to have an interface with IoT. With automation, data can be instantly collected and seamlessly passed between devices as its simultaneously analyzed. Home automation is an appealing context for the Internet of Things, by connecting the IP gateway directly to the Internet or through a home/residential gateway; this system can be managed remotely using a PC, Smartphone, Tablet or other devices.

There are three main generations of home automation. First is, different wireless technology with proxy servers, second is Artificial Intelligence (AI) controlled home automation and lastly robots which directly communicate with humans. Our project is the first generation automation. For implementing the first generation of the home automation appliances needs to connect with the internet so users can control the system from any remote place.

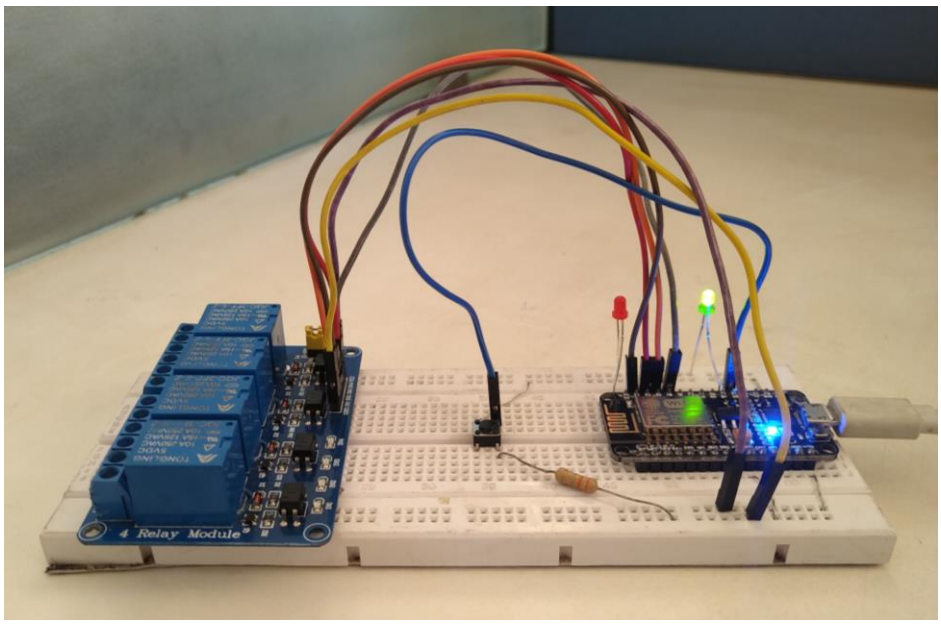
“Through project JSPYN, our aim is to allow users to easily automate their daily electronics appliances and get updates of their current city condition without writing a single line of code”

# 2. PROJECT DESCRIPTION

# DESCRIPTION

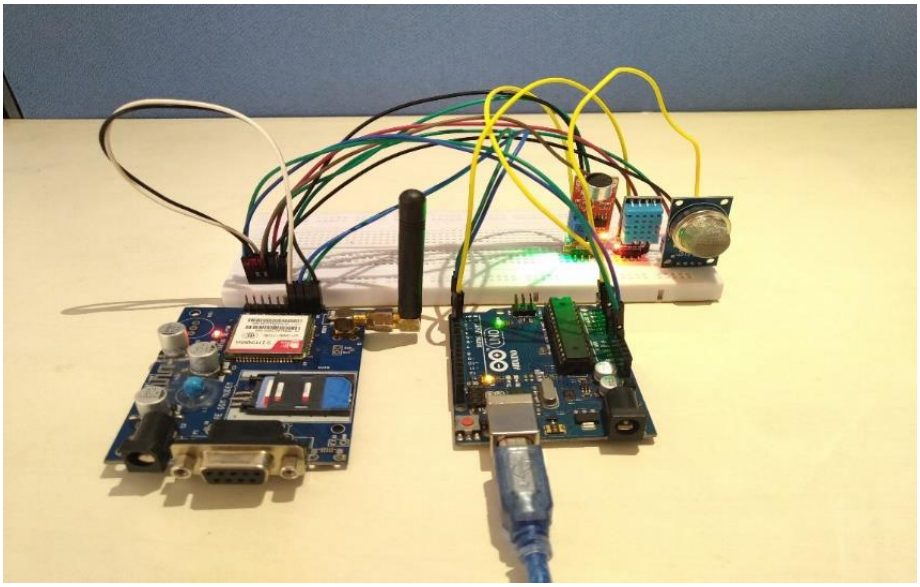
JSPYN is a project designed as an easy solution to home automation, smart home, smart city, smart offices etc. The project is based on collecting real-time data through sensors such as Temperature humidity sensor, Temperature sensor, Air Quality sensor, Flame sensor, soil humidity sensor, Water-flow sensor, Photoresistor and Sound sensor. JSPYN includes JSPYN IoT circuit board and JSPYN city hub circuit board.

- **JSPYN IoT board** includes microcontroller ESP8266 with custom firmware. This controller collects data from the connected sensors and uploads it to our database. It also includes the four channel relay board for controlling AC devices.




- **JSPYN city hub module** includes sensors connected to Arduino board connected to a GSM module to connect to the Internet. These sensors will collect data on temperature, humidity, air quality, sound level of cities. Based on this data users will get notification of his/her city condition on their JSPYN android app. The users can also view the condition of other cities through our Service.

The JSPYN IoT Board will be configured by **JSPYN Android App,** which collects data from the connected sensors. Any user will then be able to visualize the collected information in Graphical format which shows data on temperature, moisture, water flow, etc. The user will also have the functionality to setup “iflets”. Iflets are composed of triggers and actions. Triggers start an iflets and the action ends it.



Currently on JSPYN, iflets performs only three functions such as sending a PUSH notification to JSPYN android app, triggers Specific relay connecting to the JSPYN-IoT board and triggering relays in synchronous with time. When relay board is connected to the JSPYN IoT board the user will be able to set up controls like Switch, Buttons and Variables to manually control the four channel relay board included in our project.

JSPYN Android app helps in configuring the individual JSPYN IoT boards, it will also have functionalities like creating switches to control appliances connected to the respective relays to the JSPYN IoT board, viewing collected sensor data in graphical formats and also plays the role of client app to receive push notification from JSPYN. On the other hand JSPYN website will allow users to view their city conditions and even conditions of other cities. It will also allow users to change their current location if needed. Users will get a wide view of the sensor data collected in graphical format. JSPYN Website will also allow users to set iflets for triggering relays and getting push notifications.

# 3.TECHNOLOGIES USED

# INTRODUCTION

JSPYN web application is built on the Express Framework. It is designed with HTML and CSS on the handlebar template to provide a user-friendly interface along with jQuery for DOM manipulation. This application is served by Node.Js, which is a runtime environment for JavaScript. Our entire server side is written in JavaScript scripting language.

Most of the technologies used in JSPYN are provided by Google. At its core backend, JSPYN uses Firebase for the database. Firebase provides realtime database which is suitable for the various JSPYN sensors to collect and send data in real time. Firebase also provides other functions such as Firebase authentication, Firebase cloud messaging, Firebase functions, etc.

The Firebase authentication API is used by JSPYN to authenticate users during signing and logging in into our servise, so that our users can directly take the benefits out our service through their Google account.

Firebase functions are used to trigger the iflets of particular users whenever specific data changes occur in the Realt-ime database specified by the user. They are also used to send push notifications to our users through Firebase cloud messaging.

Due to Firebase cloud messaging our service can only be signed in through our Android app and then can be continued on our web application.

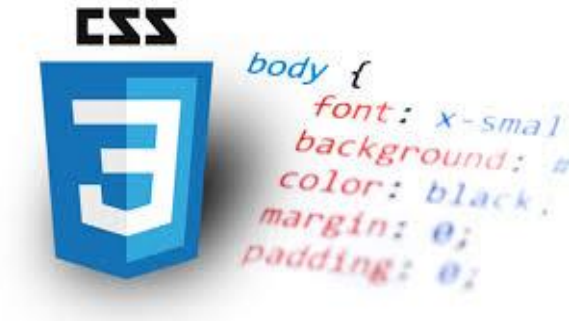
The entire web application is hosted on the Google Cloud Computing engine, which provides easy flexibility in creating user-specific and optimized virtual machine instances. It also acts as MQTT broker to communicate between our Android app and individual sensors.

# WEB TECHNOLOGIES USED

## Hypertext Markup Language


Hypertext markup language (HTML) is the standard markup language for creating web pages and web application. With Cascading style sheets (CSS) and JavaScript. Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus the links are available on a Webpage are called Hypertext.

## Cascading Style Sheets




CSS is a style sheet language used describing the presentation of documents written in a markup language like HTML. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. CSS handles the look and feel part of a web page.

## JQuery


JQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and AJAX.

# Node.js

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009.

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source and a cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

# Express.Js


Express was developed by **TJ Holowaychuk** and is maintained by the Node.js foundation.

**Express.JS** is a web application framework that provides a simple API to build websites, web apps and back ends.

Express provides a minimal interface to build applications. It provides the tools that are required to build an app. It is flexible as there are numerous modules available on **npm**, which can be directly plugged into Express.

# GOOGLE TECHNOLOGIES USED

## FIREBASE

Firebase is a backend platform for building Web, Android, and IOS applications. It offers real-time database, different APIs, multiple authentication types and hosting platform.

### Features of Firebase:

- It is simple and user-friendly. No need for complicated authentication.
- The data is real-time
- Firebase offers simple control dashboard.
- Firebase also provides functionality like analytics, messaging and crash reporting.
- Firebase is built on Google infrastructure and scales automatically, for even the largest apps.
- For building apps, firebase includes services like Cloud firestore, ML kit, Cloud functions, Authentication, Hosting, Cloud storage, and Realtime database.
- For improving app quality firebase provides Crashlytics, Performance monitoring and Test lab.
- To improve the growth of your business it provides services like Analytics, Predictions, Firebase A/B testing, Cloud Messaging, Remote config, Dynamic links, App indexing and Invites.
- Firebase supports app building in iOS, Android, Web, C++ for Android or iOS, SDKs for unity games and provides Admin SDKs for its database.

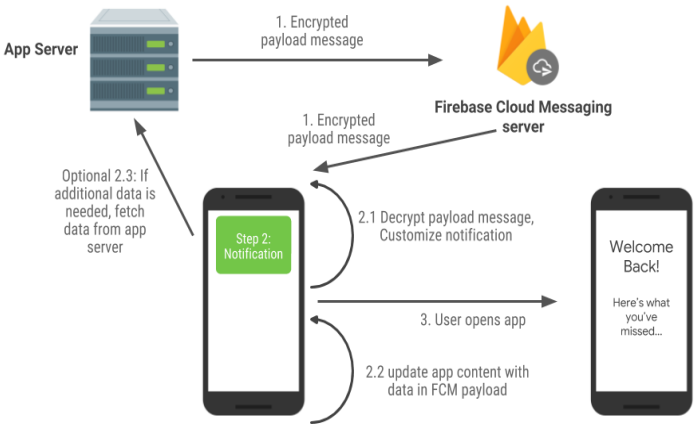
# FIREBASE REAL-TIME DATABASE

The Firebase Real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of the clients share one Real=time Database instance and automatically receive updates with the newest data.

Store and sync data with NoSQL cloud database. Data is synced across all clients in realtime and remains available when the app goes offline.


# FIREBASE CLOUD MESSAGING

Using **FCM**, we can notify a client app that new email or other data is available to sync. We can send notification messages to drive user re-engagement and retention. For use cases such as instant messaging, a message can transfer a payload of up to 4KB to a client app.

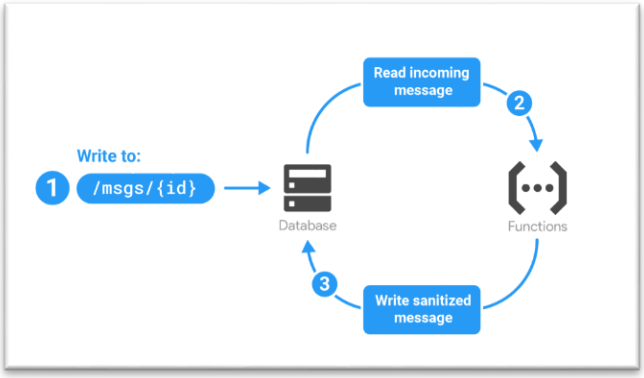


# FIREBASE CLOUD FUNCTIONS

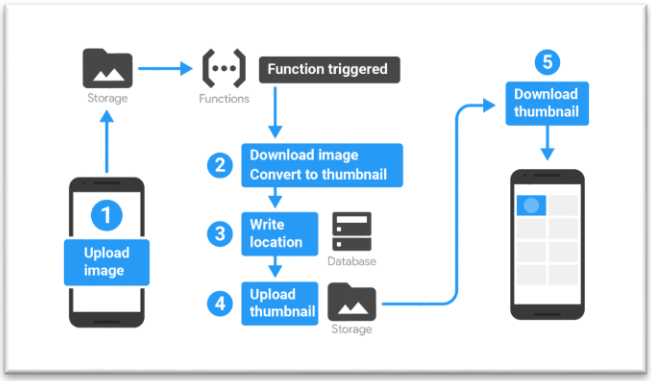
Cloud Functions for Firebase lets you automatically run backend code in response to events triggered by Firebase features and HTTPS requests. Your code is stored in Google's cloud and runs in a managed environment.

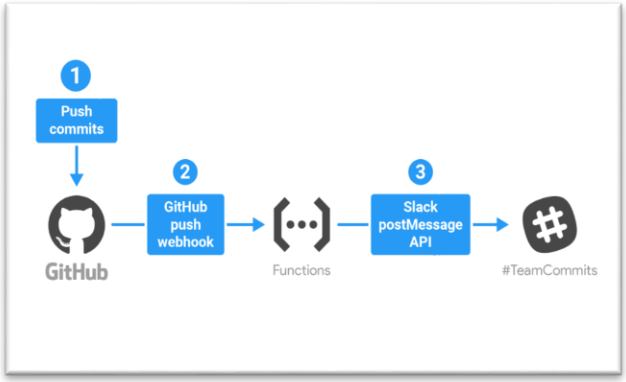
## Features of cloud functions

- Notify users when something interesting happens
- Performs real-time database sanitization and maintenance.


- Execute intensive tasks in the cloud instead of in the app
- Integrates with third-party services and APIs



## FIREBASE AUTHENTICATION

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.
Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.


## ANDROID SDK


A software development kit that enables developers to create applications for the Android platform. The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications. Applications are written using the Java programming language and run on Dalvik, a custom virtual machine designed for embedded use which runs on top of a Linux kernel.

# GOOGLE CLOUD COMPUTING ENGINE

Google Compute Engine delivers virtual machines running in Google's innovative data centres and worldwide fibre network. Compute Engine's tooling and workflow support enables scaling from single instances to global load-balanced cloud computing.


## Features of Google Cloud Platform

- Compute engine's VMs boot quickly.
- Comes with persistent disk storage.
- Creating custom machine types optimized for specific needs.
- Uses Google's global fibre network.
- Resizing clusters, creating machine images, virtualizing network, supports Pre-emptible VM's for batch workloads, creating custom machine types.

# 4. SOFTWARES USED

## Visual Studio Code

Visual Studio Code is a source code editor developed by Microsoft for Windows OS, Linux, and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts and preferences. It is free and open-source, although the official download is under a proprietary license.

## Android Studio

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

## Arduino IDE

Arduino integrated development environment is a cross-platform application for Windows, macOS, and Linux that is written in the programming language Java. It is used to write and upload programs to Arduino board

# 5. PROTOCOLS USED IN JSPYN

# INTRODUCTION

It is a digital language through which we communicate with others on the Internet. Protocol is a set of rules used to implement the exchange of information between the user and the receiver.

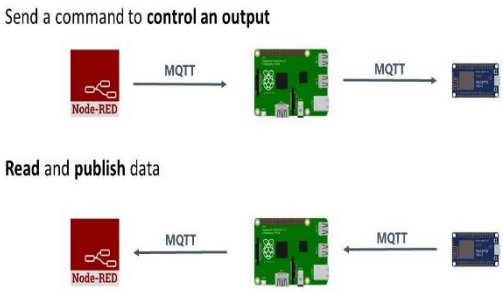
In our project JSPYN we use four main protocols:

1. **HTTP** for web application and Android application to make requests to the server and database.
2. **MQTT** for communication between the Android app and individual sensors.
3. **IEEE 802.11** which is the protocol used by ESP8266 microcontroller to connect to the internet.
4. **NETWORK TIME PROTOCOL** is used in the JSPYN city hub and the JSPYN IoT board to send the sensors data respective to the current time.

## Hyper Text Transfer Protocol


The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP is a TCP/IP based communication protocol that is used to deliver data on the World Wide Web.

## MQTT




MQTT (Message Queuing Telemetry Transport) is an ISO standard publish-subscribe based messaging protocol. It works on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker.

## IEEE 802.11 Protocol

It defines an over-the-air interface between a wireless client and a base station or between two wireless clients. Wi-Fi stands for Wireless Fidelity Wi-Fi. It is based on the IEEE 802.11 family of standards and is primarily a local area networking technology designed to provide in-building broadband coverage.

## NETWORK TIME PROTOCOL

Network Time Protocol is a protocol used to synchronize computer clock times in a network. It belongs to and is one of the oldest parts of the TCP/IP protocol suite. NTP uses Coordinated Universal Time to synchronize computer clock times with extreme precision, offering greater accuracy on smaller networks down to a single millisecond in a local area network and within tens of milliseconds over the internet.

# 6. HARDWARE

# INTRODUCTION

JSPYN includes two main hardware devices JSPYN IoT board and JSPYN city hub.

JSPYN IOT board includes microcontroller ESP8266 to communicate between the different sensors and firebase. It includes sensors such as temperature sensor, flame sensor, photoresistor, soil humidity sensor, water flow sensor. The microcontroller collects the data from the individual sensors and sends it directly to the firebase real-time database.

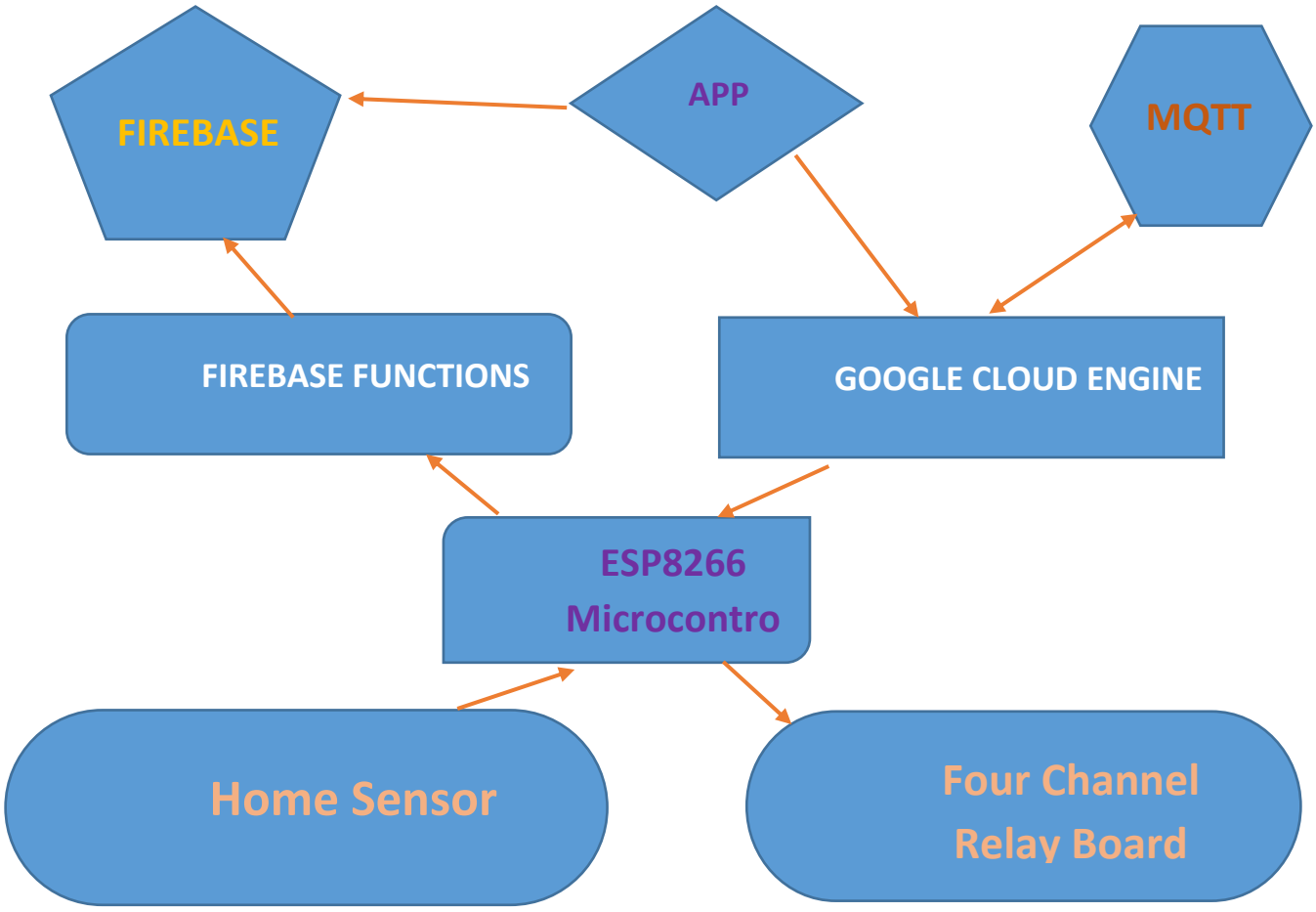
JSPYN city hub includes Arduino microcontroller to collect data from the different sensors like the temperature and humidity sensor, air quality sensor and sound sensor. The collected data is sent to the firebase real-time database with the help of GSM module SIM900A. The entire setup will be battery powered and will be placed in different cities.

# JSPYN HARDWARE

## JSPYN IoT board

JSPYN IoT device is esp8266 microcontroller with custom firmware and detached capability to reprogram. Its firmware is programmed using C and tested under harsh conditions. In case of power loss or failure, there is no need for the user to reconfigure the device as it stores all the configuration in its ROM.

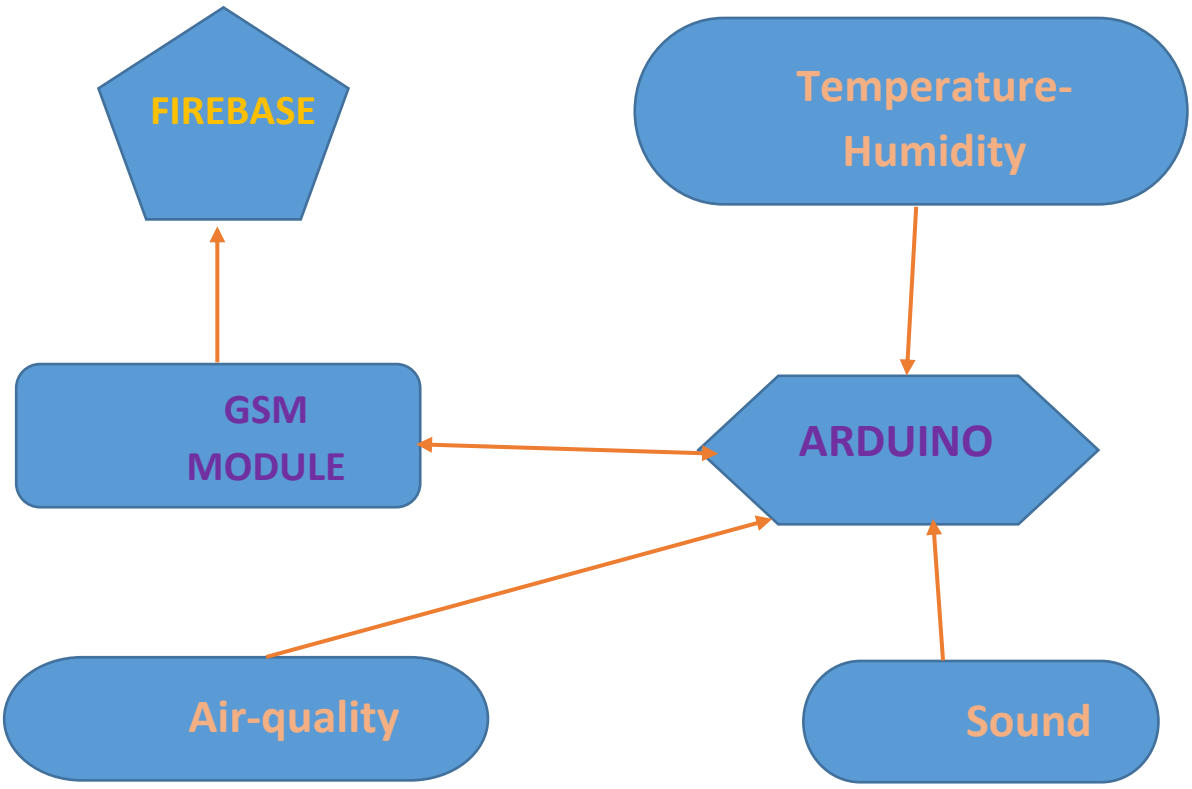
The JSPYN IoT board has four channel relays build on it, along with extra pins to attach different sensors and PWM output (rotary mechanism). The device is supported for one sensor and two PWM output. Four channel relays are marked as Relay_1 - Relay_4 accordingly. The device could be powered using a normal USB phone charger. Along with this, it has two leds for different indicators and two buttons for resetting device and to push the device on setup mode.




***Block diagram showing the working of JSPYN IOT Board***

## JSPYN city hub

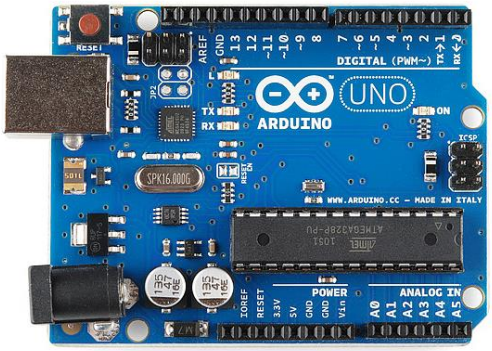
JSPYN city hub module includes sensors connected on an Arduino board using the GSM module to connect to the Internet. These sensors will collect data on temperature, humidity, air quality and the sound level of the city. Based on this data users will get notifications of his/her city condition on their JSPYN android app. The users can also view the condition of other cities through our Service.




***Block diagram showing the working of JSPYN city hub***

# BRIEF ABOUT MICROCONTROLLERS AND SENSORS USED

## Arduino

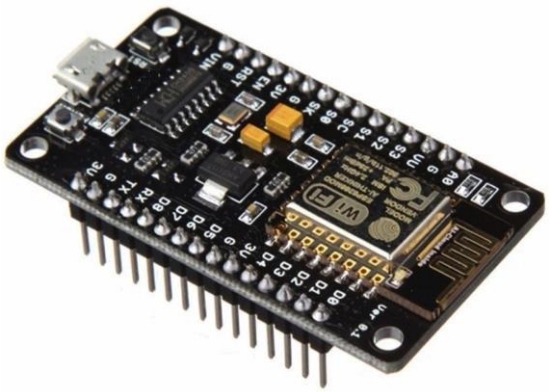



Arduino is an open-source platform based on easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE which is used to write and upload the computer code to the physical board.

## GSM Module SIM900A


The SIM900A is a complete Dual-band GSM/GPRS solution in an SMT module. Featuring an industry-standard interface, the SIM900A delivers GSM/GPRS 900/1800MHz performance for voice, SMS, Data, and Fax in a small form factor and with low power consumption.

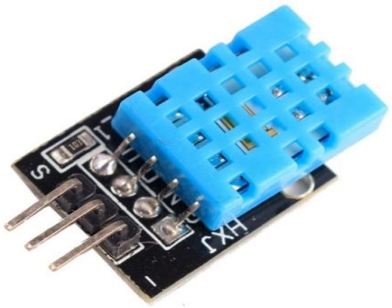
## NodeMCU ESP8266



NodeMCU is an open source firmware and development kit that helps to prototype or build IoT product. It includes firmware which runs on the esp8266 Wi-Fi SoC from EspressIf Systems, and hardware which is based on the ESP-12 module. The ESP8266 itself is a self-contained Wi-Fi networking solution offering as a bridge from existing micro controller to Wi-Fi and is also capable of running self-contained applications

# SMART CITY SENSORS

## Temperature Humidity Sensor (DHT11)



This DHT11 Temperature and Humidity Sensor features a calibrated digital signal output with the temperature and humidity sensor capability. It is integrated with a high-performance 8-bit microcontroller. It has excellent quality, fast response, anti-interference ability, and high performance

## Air quality Sensor (MQ135)

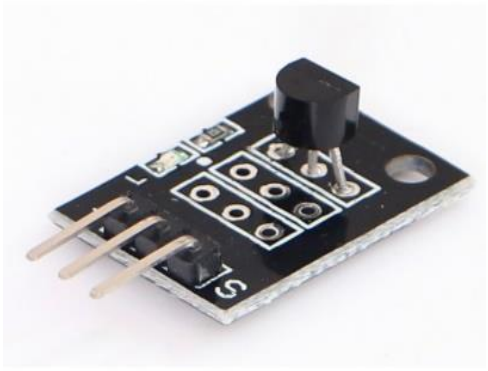
The air quality sensor is also an MQ-135 sensor for detecting venomous gases that are present in the air in homes and offices. The air quality sensor detects ammonia, nitrogen oxide, smoke, $CO_2$ and other harmful gases. The air quality sensor has a small potentiometer that permits the adjustment of the load resistance of the sensor circuit.

## Sound Sensor (Big Sound Sensor)

The sound sensor module provides an easy way to detect sound and is generally used for detecting sound intensity. This module can be used for security, switch, and monitoring applications. Its accuracy can be easily adjusted for the convenience of usage.

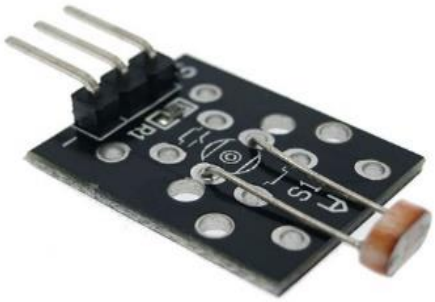
# SMART HOME SENSORS

## Temperature Sensor (18B20)



The **DS18B20** is a 1-wire programmable Temperature sensor from maxim integrated. It is widely used to measure temperature in hard environments like in chemical solutions, mines or soil etc. It can measure a wide range of temperature from **-55°C to +125°** with a decent accuracy of **±5°C**.

## Photoresistor



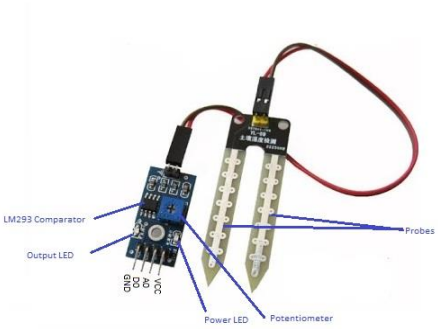
Photoresistors, also known as light dependent resistors (LDR), are light sensitive devices most often used to indicate the presence or absence of light or to measure the light intensity. In the dark, their resistance is very high, sometimes up to 1MΩ, but when the LDR sensor is exposed to light, the resistance drops dramatically, even down to a few ohms, depending on the light intensity.

## Flame Sensor

This flame sensor can be used to detect fire source or other light source of the wavelength in the range of 760nm – 1100nm. It is based on the YG1006 sensor which is a high speed and high sensitive NPN silicon phototransistor. When the sensor detects flame the signal LED will light up and the DO (Digital output) pin goes LOW.

## Soil Humidity Sensor

This sensor measures the volumetric content of water inside the soil and gives us the moisture level as an output. The sensor is equipped with both analog and digital output, so it can be used in both analog and digital mode.




## Water Flow Sensor (YF-S201)

This sensor sits in line with your water line and contains a pinwheel sensor to measure how much liquid has moved through it. There's an integrated magnetic Hall Effect sensor that outputs an electrical pulse with every revolution. The Hall Effect sensor is sealed from the water pipe and allows the sensor to stay safe and dry.

# 7. LIBRARIES AND PACKAGES USED IN JSPYN

# LIBRARIES USED IN JSPYN HARDWARE

## JSPYN IoT BOARD

### DHT.h

It is a Temperature and Humidity Sensor Libraries for Arduino. It has the function to get the Humidity and Temperature reading from the Sensor.

### ESP8266WIFI.h

It's all about Wi-Fi to connect new ESP8266 module to Wi-Fi Network to start Sending and Receiving data.

### FirebaseArduino.h

Firebase Arduino is a library to simplify connecting to the Firebase Database from Arduino clients.

### EzTime.h

EzTime is very easy to use Arduino time and date library that provides NETWORK TIME PROTOCOL network time lookups, extensive timezone support, formatted time and date strings, user events, milliseconds precision and more.

## JSPYN CITY HUB

### DallasTemperature.h & OneWire.h

OneWire lets you access 1-wire devices made by Dallas, such as temperature sensors and button secure memory for Temperature sensors, The Dallas Temperature library can be used with OneWire library. OneWire communicates with 1-wire devices.

### MQ135.h

This library is used for MQ-135 gas sensor to:

- Get the sensor resistance in kOhm.
- Get the ppm of CO2 sensed (assuming only CO2 in the air).
- Get the resistance RZero of the sensor for calibration purposes.

# PACKAGES USED IN ANDROID STUDIO

### Okhttp

OkHttp is a third-party library developed by Square for sending and receive HTTP-based network requests.

### FileTree

A FileTree represents a hierarchy of files. It is used to represent files to copy or the contents of an archive

### Firebase Core

It provides services to firebase analytics.

### Firebase database

Provides services to implement firebase real-time database

### Firebase Messaging

It provides services to send firebase cloud messages.

### Firebase Auth

Includes dependencies to set user authentication.

### GraphView

Android Graph Library for creating zoom able and scrollable line and bar graphs.

### SpinnerView(by jaredrummler)

A spinner view for Android

### AnimationEasingFunctions(by daimajia)

Android animation functions.

### AndroidAnimations(by daimajia)

Android animation collection.

### AndroidSwipeLayout(by daimajia)

Swipe layout for android

### Crystal-Preloaders(by syedowaisali)

Preloaders for android.

### MaterialTextField(by florent37)

Material design edit text for android.

### ExplosionField(by tyrantgit)

Explosive dust effect libraries for views in android.

### SweetAlertDialogue(by FORIS)

Alert dialogues for android.

### DescreteSeekbar(by ANderWeb)

Material design Seekbar.

# PACKAGES USED IN JSPYN WEBSITE

## Server-Side Packages

### Body-Parser

- It is a Node.js body parsing middleware.
- It parses incoming request bodies in a middleware before your handlers.
- The body is available under the “req.body” property.

### Consolidate

- It is a template engine consolidation library.
- It includes the handlebar view template which is used in the JSPYN Website.

### Express

- Express.JS is a web application framework that provides a simple API to build websites, web apps and back ends.
- It’s a Fast, un-opinionated, minimalist web framework for node.

### Express-Handlebars

- Handlebars let you build semantic templates effectively.
- Handlebars allow templates to be precompiled and included as JavaScript code rather than the handlebars template allowing for faster start up time.

### Express-Session

- Express session stores user data between HTTP requests.
- It does this by creating user sessions on the server side

### Passport

- Passport is authentication middleware for Node.js
- It serves one purpose, authenticates requests
- It can be also be integrated with express-sessions to serialize and de-serialize users in sessions.

## Firebase-Admin

The Admin SDK lets you interact with Firebase from privileged environments to perform actions like:

- Read and write Real-time Database data with full admin privileges.
- Programmatically send Firebase Cloud Messaging messages using a simple, alternative approach to the FCM server protocols.
- Generate and verify Firebase auth tokens.
- Access Google Cloud Platform resources like Cloud Storage buckets and Firestore databases associated with your Firebase projects.
- Create your own simplified admin console to do things like look up user data or change a user's email address for authentication.

## Path

- The path module provides utilities for working with file and directory paths.

# Client-Side Packages

## Fontawesome

Provides vector icons and social logos for websites.

## Materializecss

A modern responsive front-end framework based on Material Design

## Modernizer

- Modernizr tells you what HTML, CSS, and JavaScript feature the user's browser has to offer.
- It's a collection of superfast tests, which runs as your web page loads, then the results can be used to tailor the experience to the user.

## Styleshout

This site offers a collection of beautifully handcrafted hyml5 and css3 website templates.

## CDN

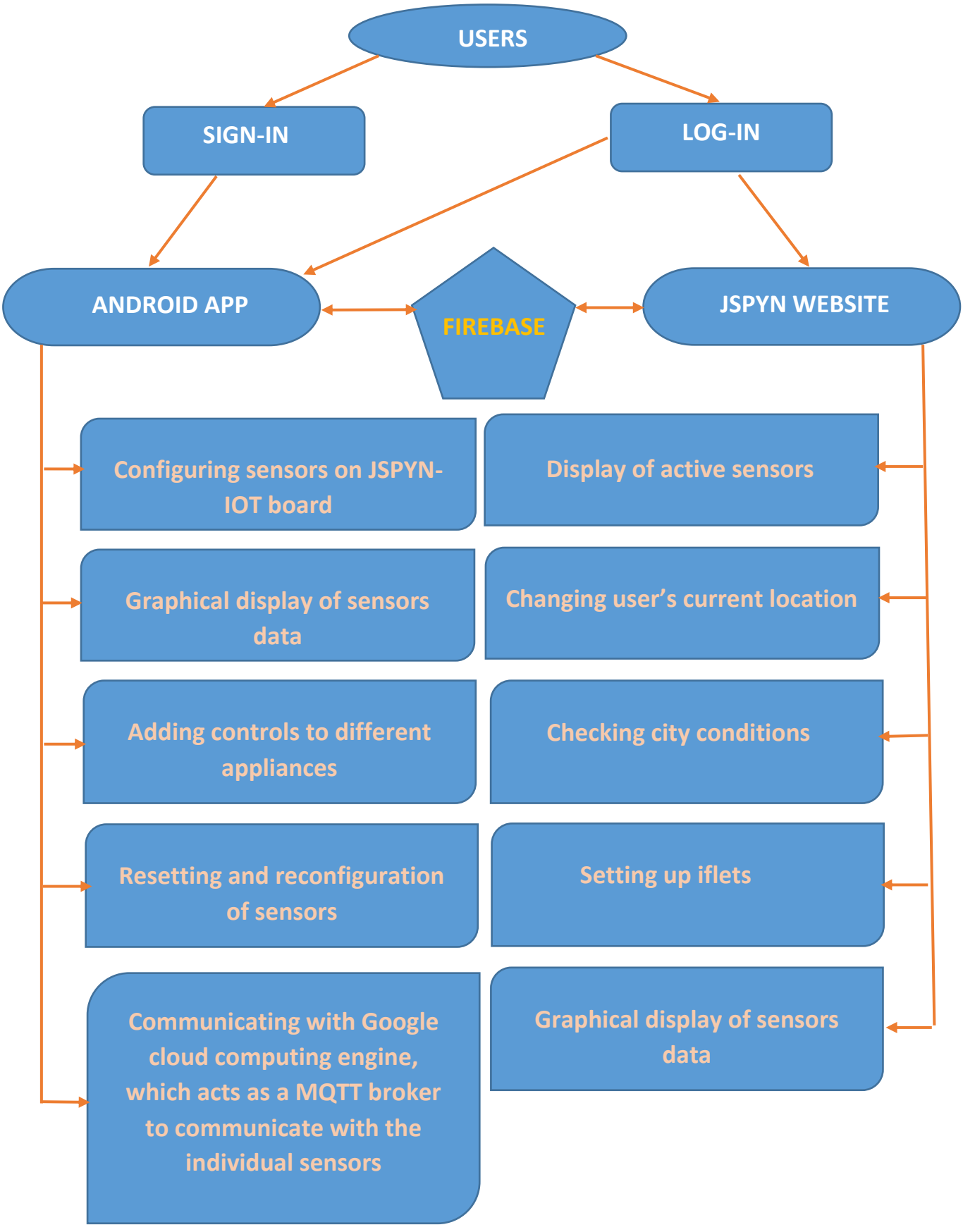
**CDN** is short for content delivery network. A content delivery network is a system of distributed servers that deliver pages and other Web content to a user, based on the geographic locations of the user, the origin of the webpage and the content delivery server.

## ApexCharts

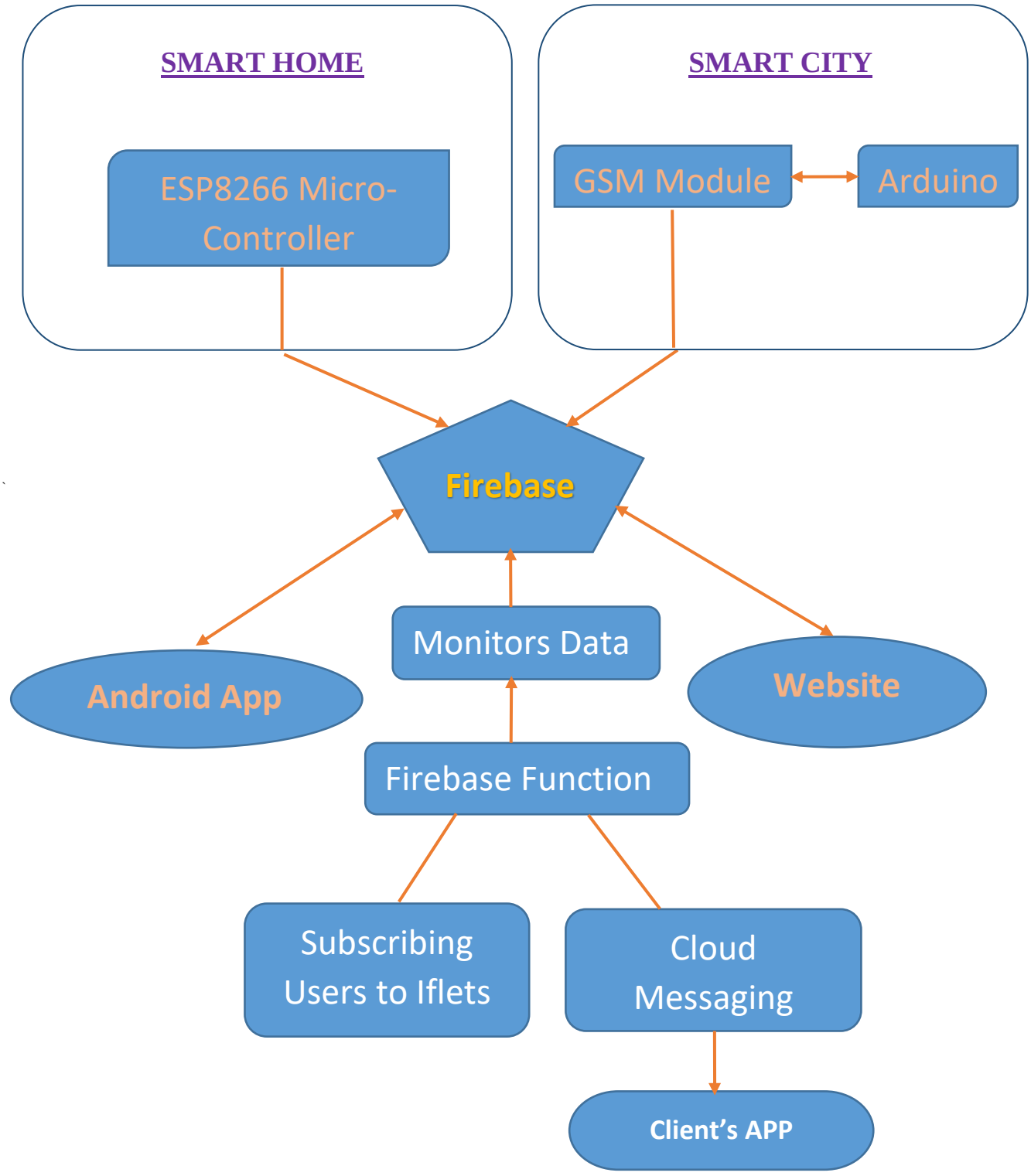
Provides Modern & Interactive JavaScript Charts to create a beautiful representation of data.

# 8. DATA HANDLING

# USE CASE DIAGRAM



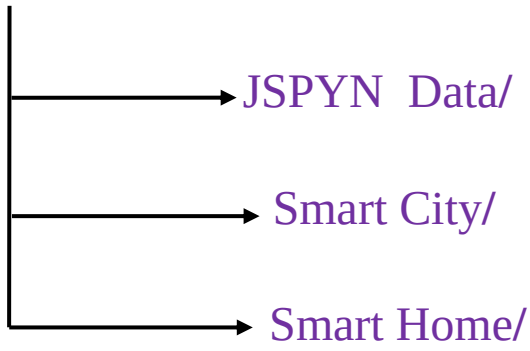
# DATA FLOW


SMART HOME
ESP8266 Micro-Controller
SMART CITY
GSM Module
Arduino
Firebase
Monitors Data
Android App
Website
Firebase Function
Subscribing Users to Iflets
Cloud Messaging
Client's APP

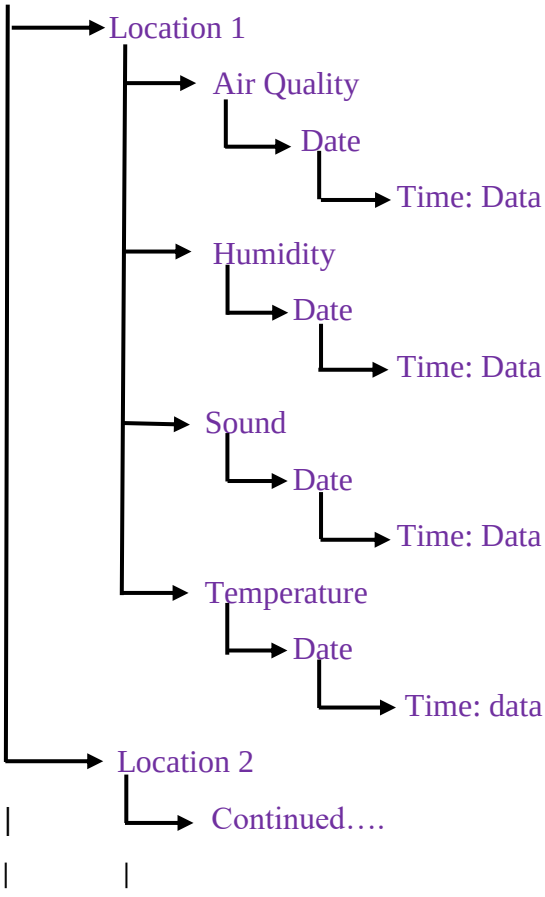
# DATA STRUCTURE

## FIREBASE NODES (NoSQL database)

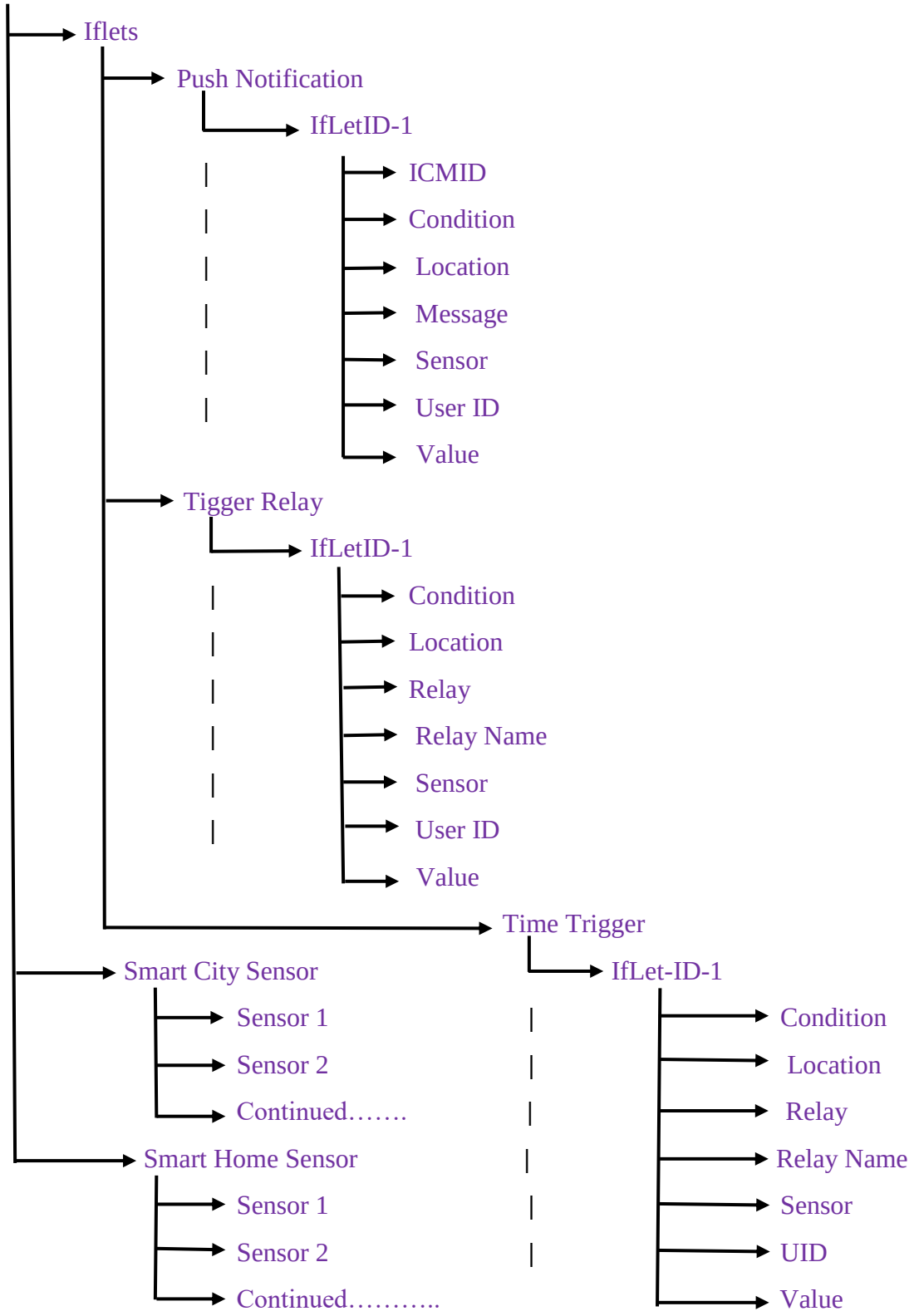
### JSPYN



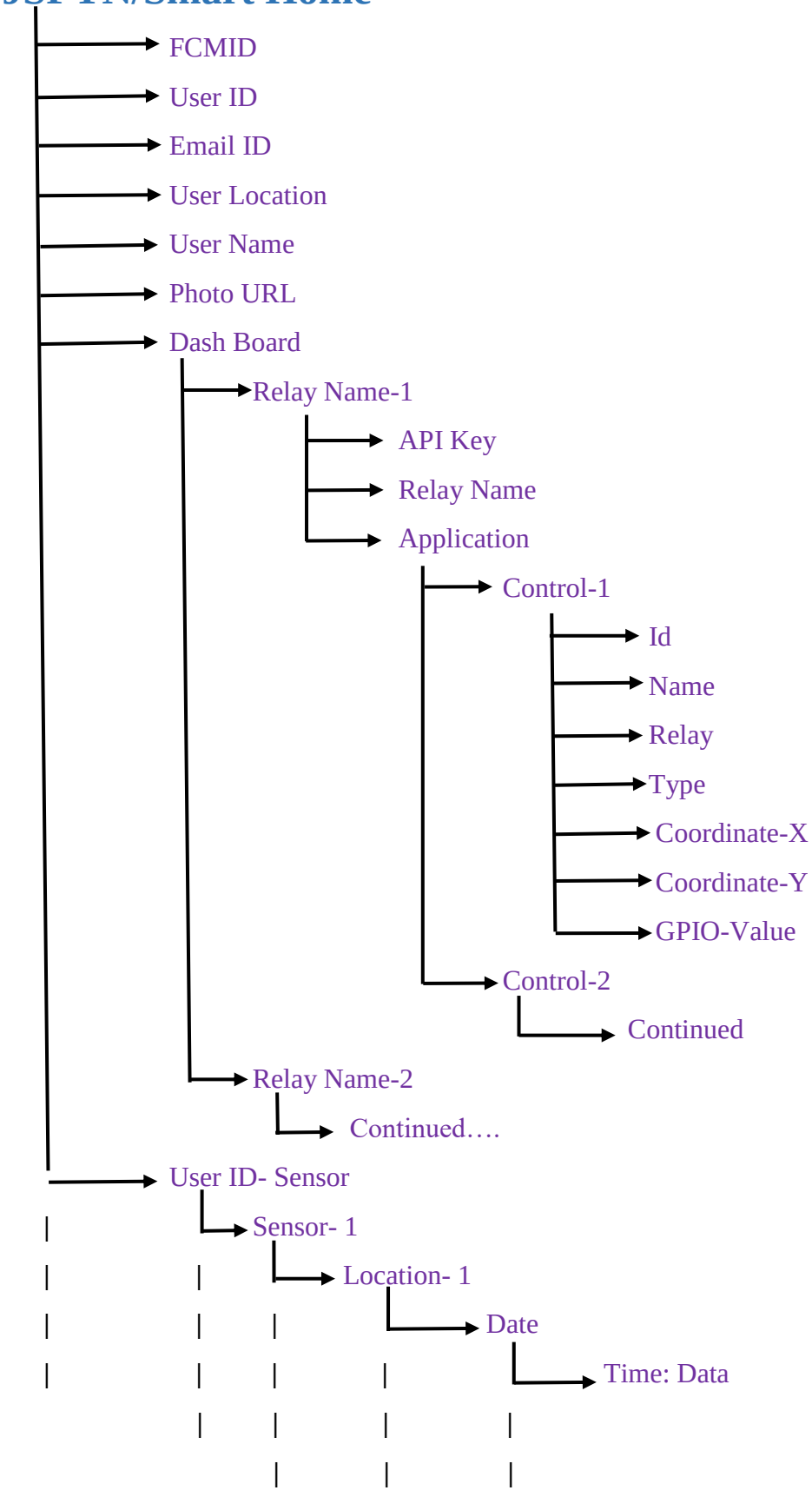

### JSPYN/Smart City



# JSPYN/JSPYN Data



# JSPYN/Smart Home

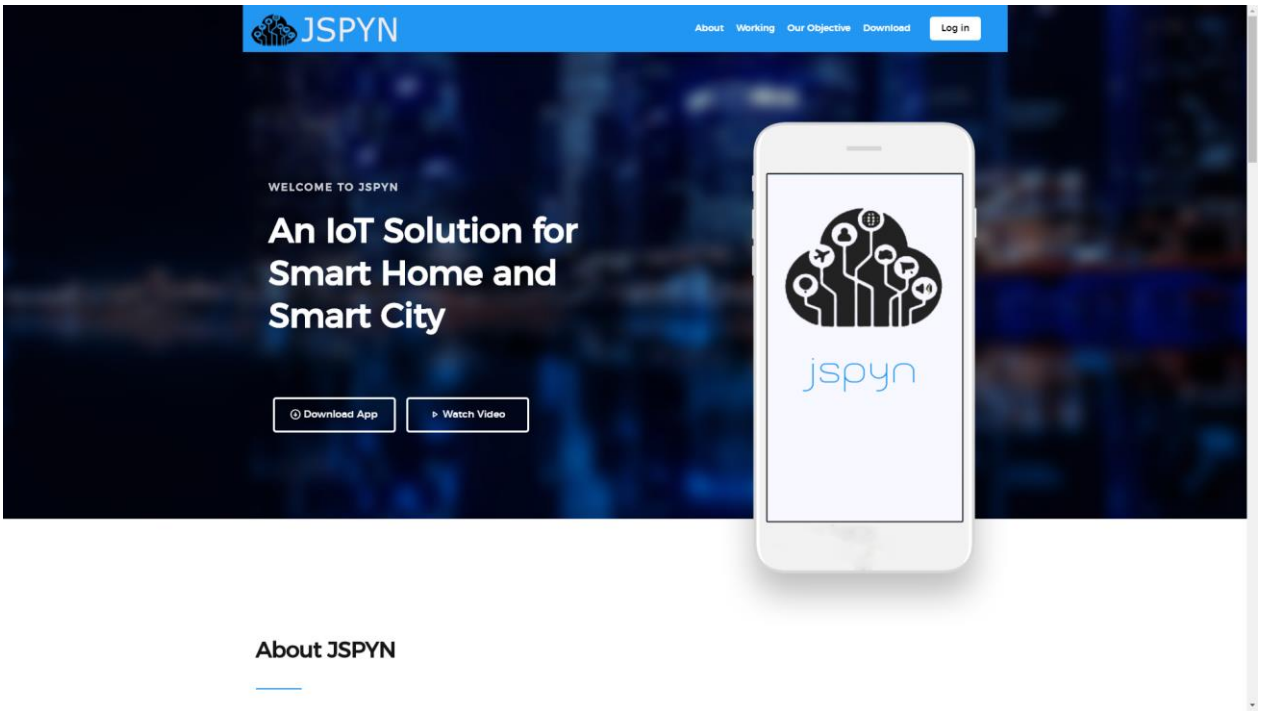


# 9. USER MANUAL

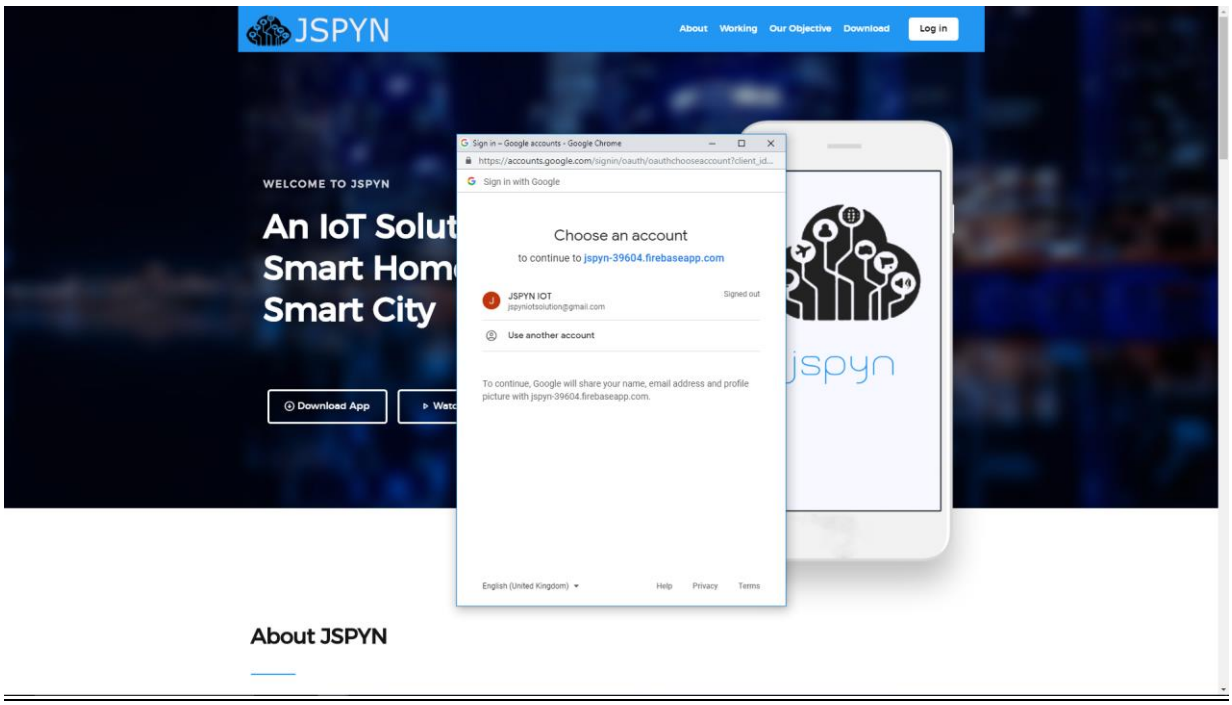
# JSPYN WEBSITE

## Log-in/ Sign-in Page

1. For Sign-In/Log-in we are using fire-base authentication, from which we are using the Google authentication providers to Sign-In/Log-in our users.

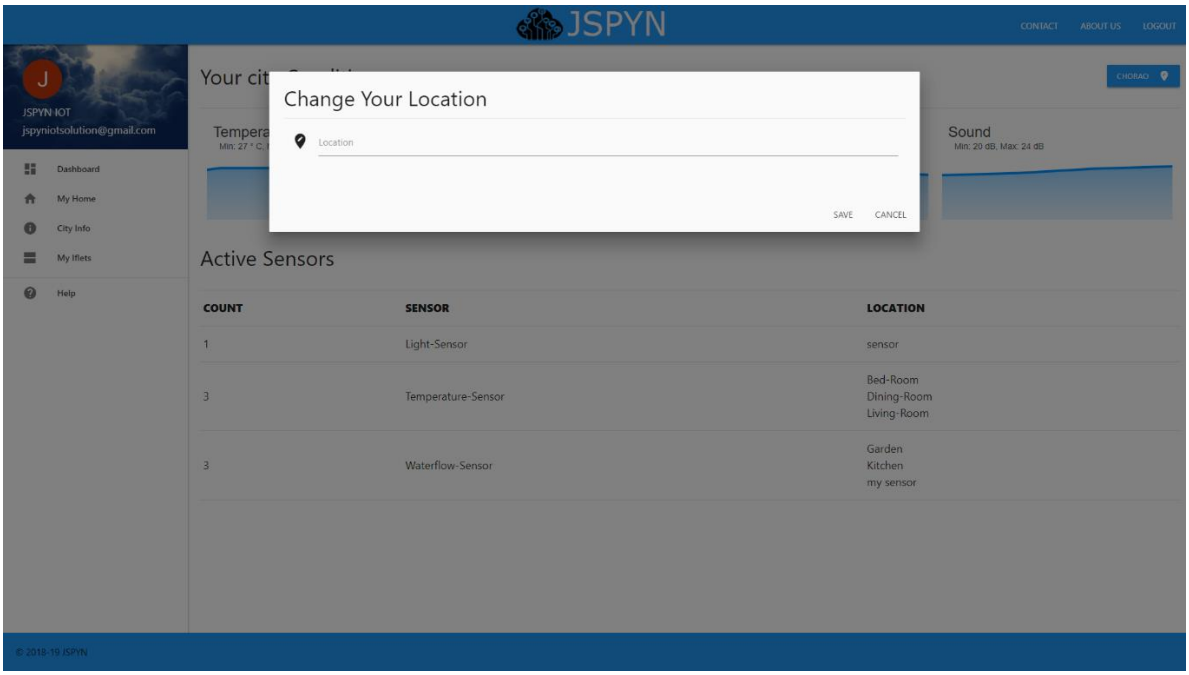



2. Our service can only be signed in through our Android app not from our website, due to fire-base messaging FCMID.



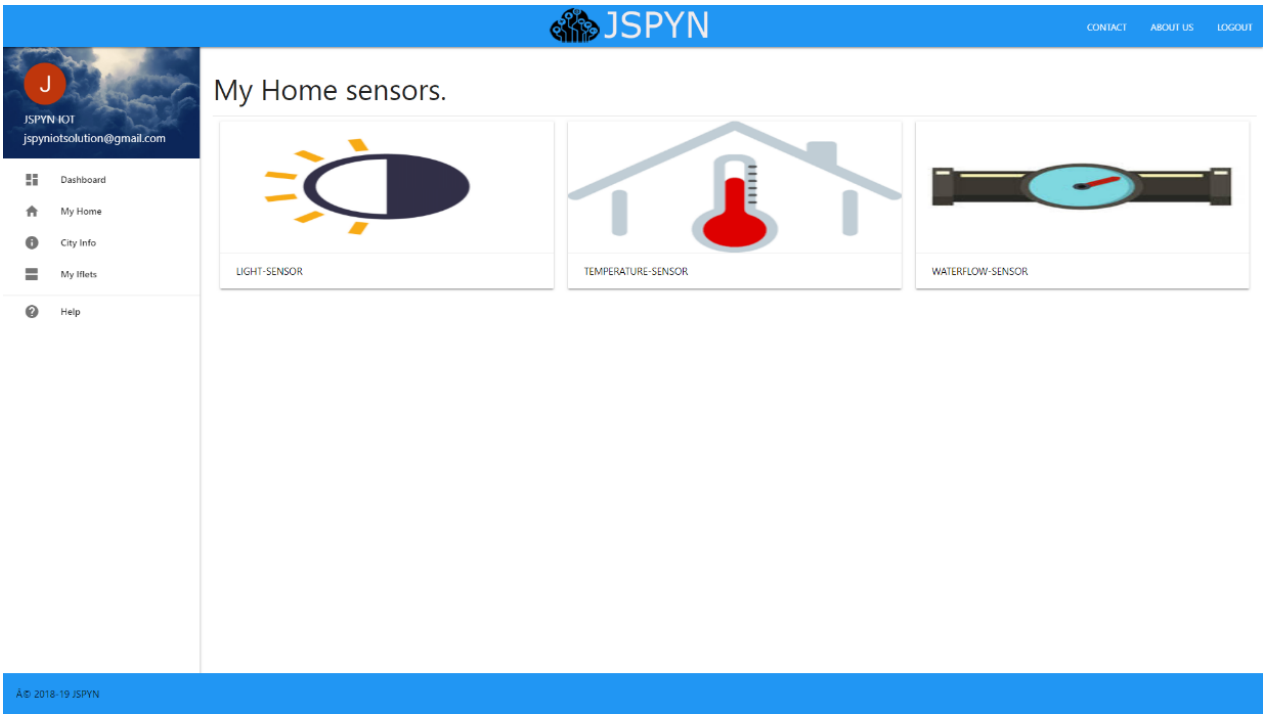
## Dashboard


1. The dashboard shows the current condition of the user's location (temperature, humidity, sound, Air Quality) based on his/her location.
2. It shows the number of active sensors at the users home in tabular form.
3. Top right button allows the user to change his/her current location, the input also provides auto-complete functionality based on the cities covered by JSPYN.

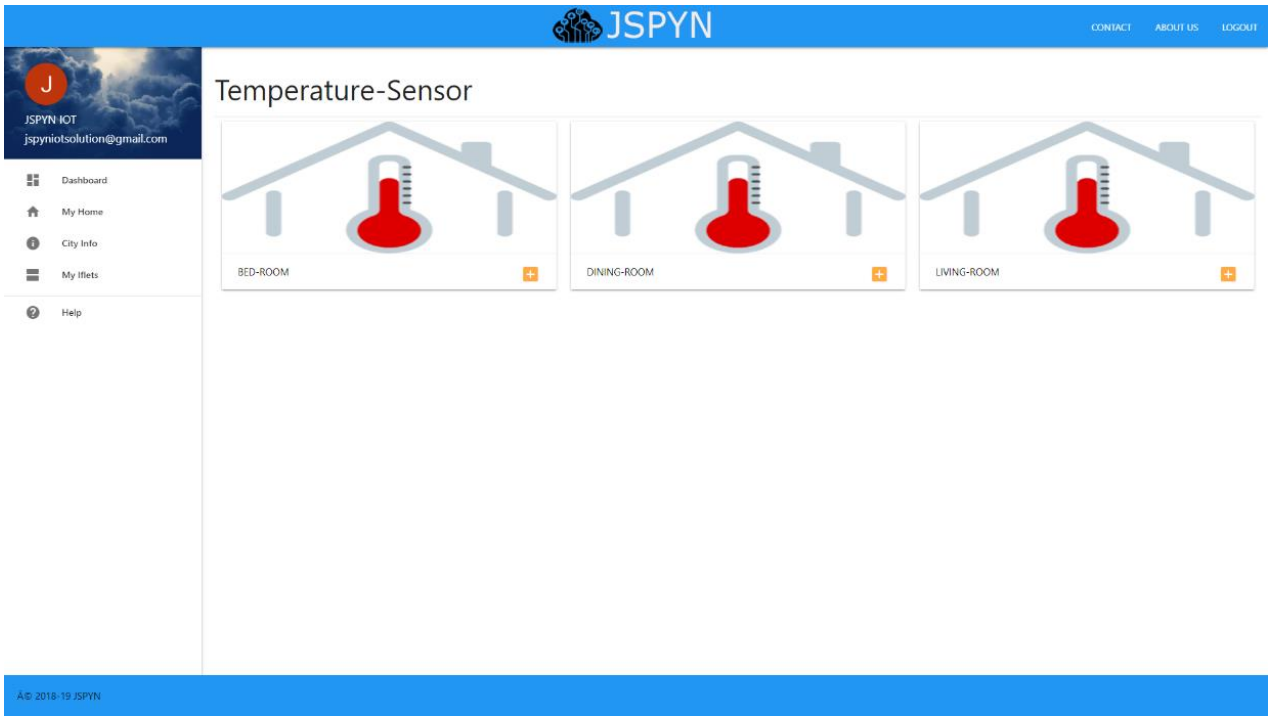


## Smart Home

1. The smart home section provides a list of all sensors configured by the user at his/her home.

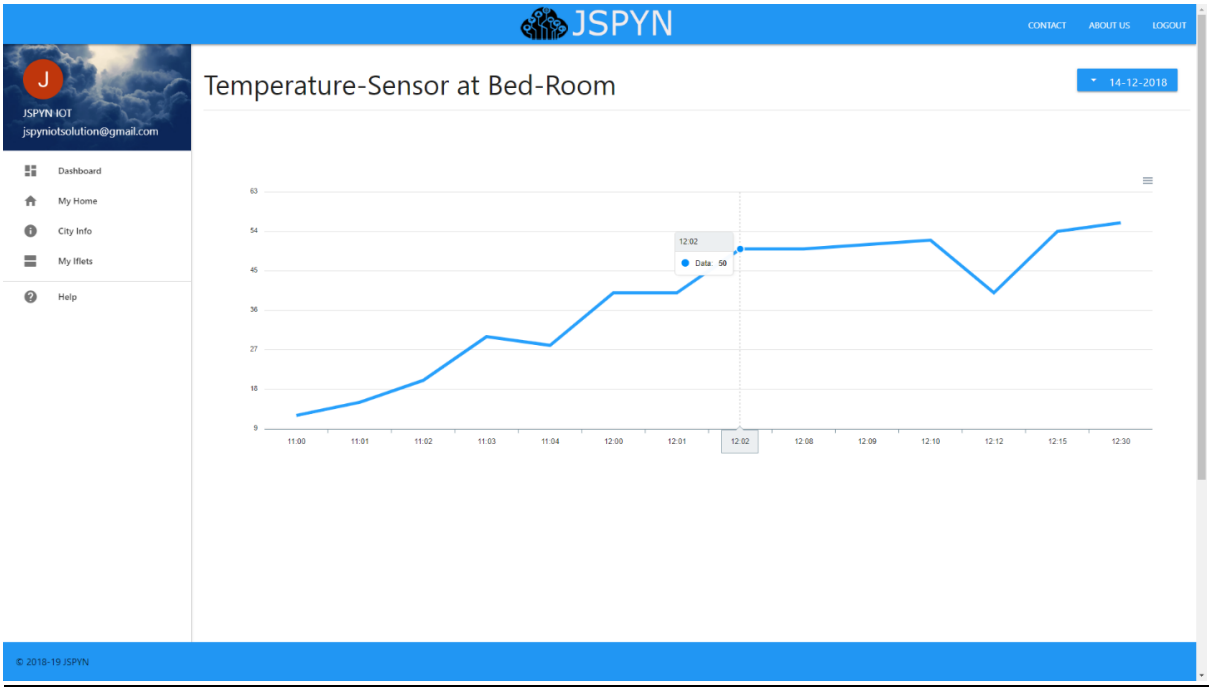



2. Each sensor then gives a list of the location at which the sensor is installed;



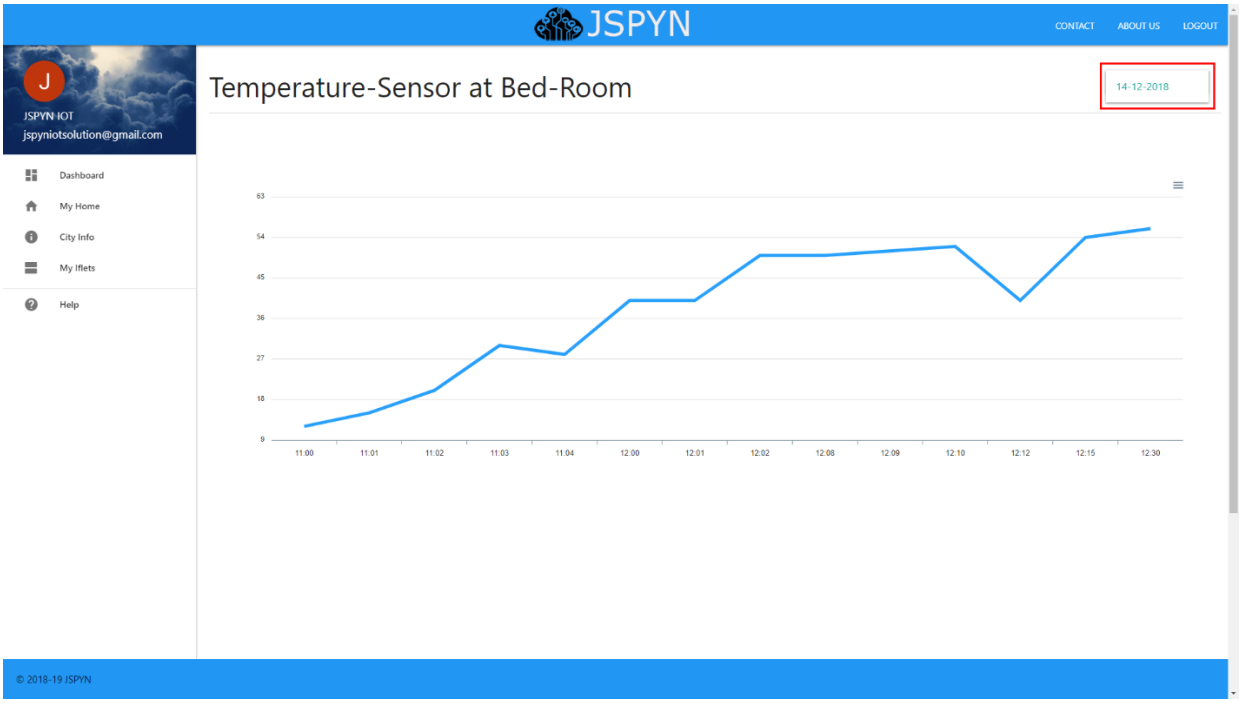
3. Each location has two functionality, clicking on the location name will display the data in a graphical format and the yellow plus button will take to iflet setup.

## Data display




1. Each sensor on clicked at the location will provide a graphical representation of the data collected by the sensor.

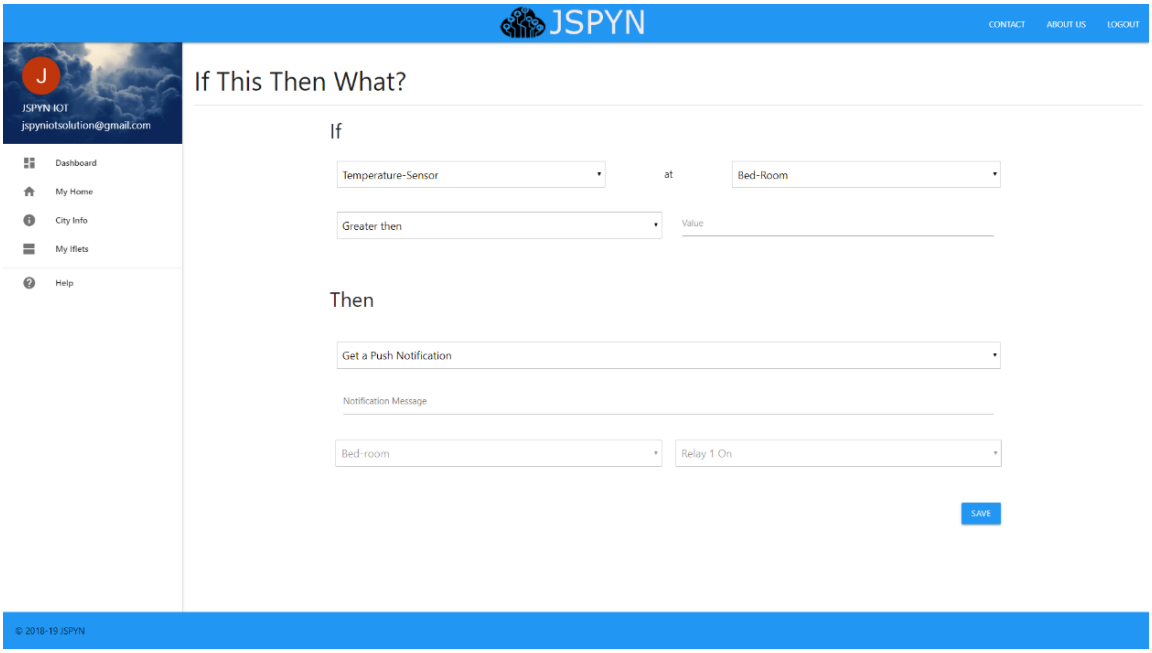
2. Top right button will allow users to view graphs from different dates.



## Iflet Configuration

Iflets are conditions, on satisfied triggers an event. They are composed of triggers and actions.

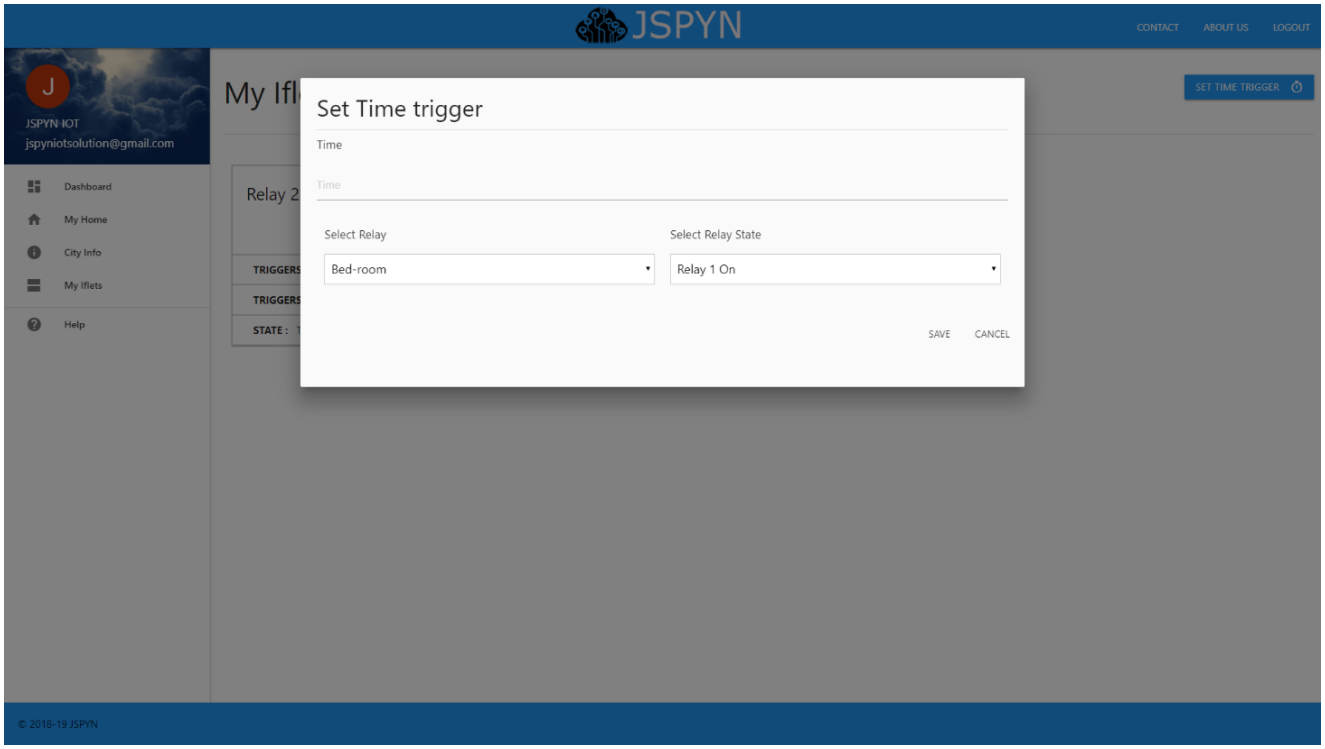
Trigger starts an iflet and the action ends it. Currently, JSPYN iflets performs only three actions sending a push notification to users android app, to trigger the specific relay and time based relay triggering.




1. The yellow plus button on each location will display a form to configure the iflet.

2. On save, all the saved iflets will be shown in the ‘My Iflets’ section of the website.


3. The top right button is to set the time trigger iflet. On click will display a form to fill to set up the trigger.

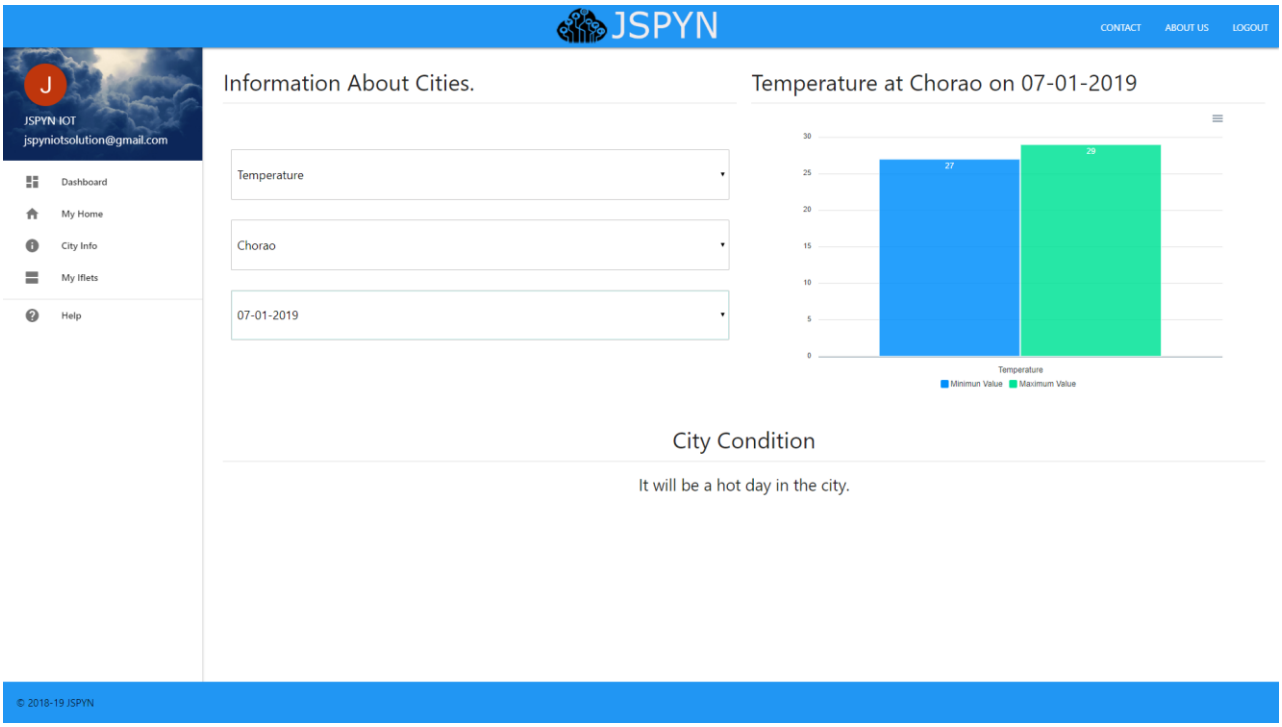


## City Info

1. City info page provides our user, information about the different conditions of a city(Air quality, sound, temperature, humidity).


2. On selecting the options from the three drop-downs graphical representation of the minimum and maximum value of the selected sensor at a particular location on specific date will be displayed.

3. If the data requested is latest, a small phrase on the city's condition will be displayed based on the data collected by the sensor at that city location.

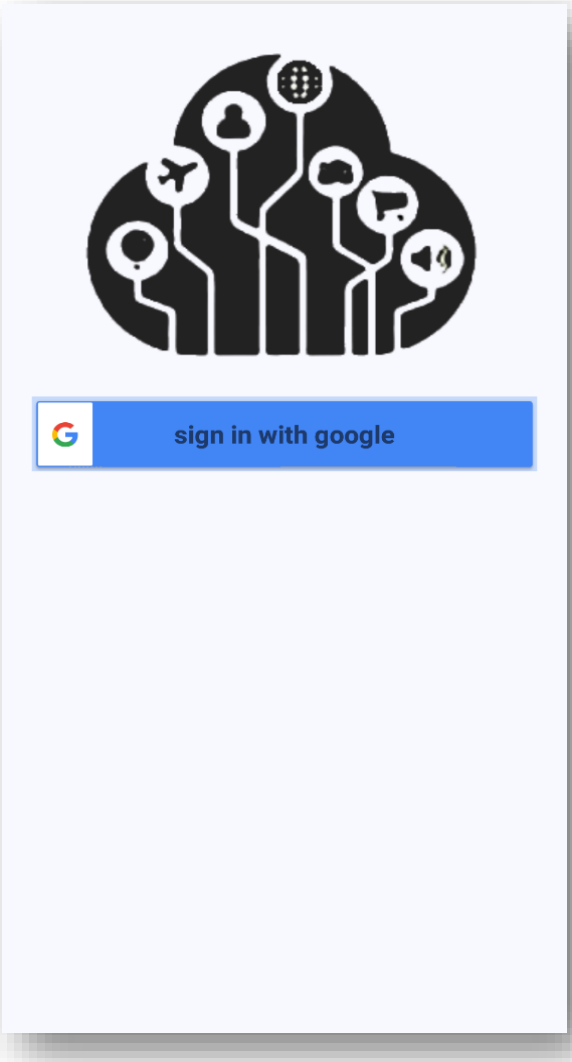


# JSPYN ANDROID APP

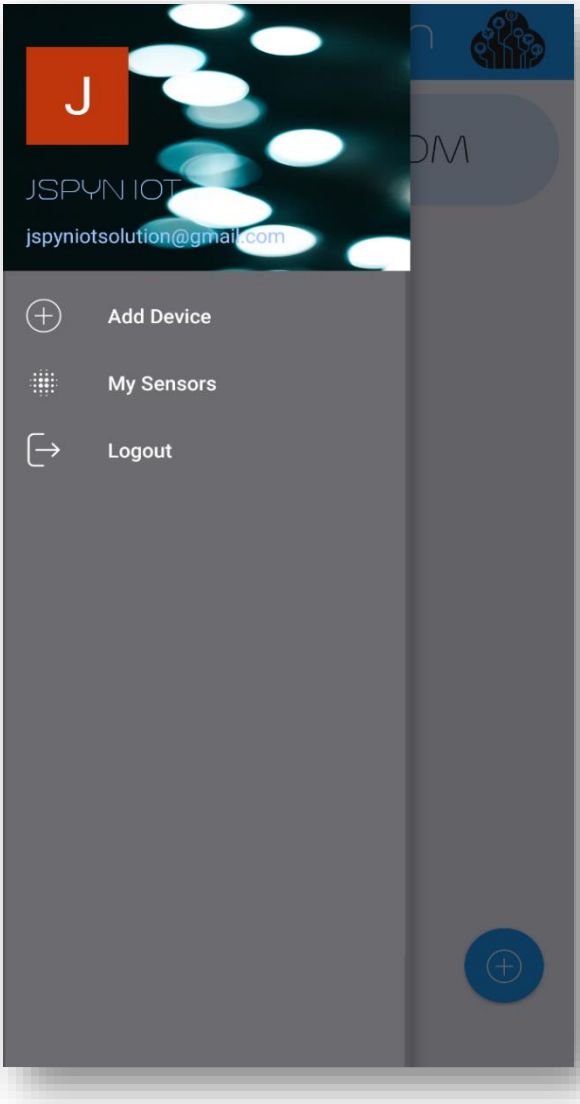
- JSPYN IOT app is used to work out with the JSPYN IOT device.
- From configuring the device to creating your custom app for your home, office or your workplace.
- The app is also configured for our **IFLET Service** so that user will not miss any notification from its devices, and sensors.

**How to go about:**

- First of all, install JSPYN IOT Android application on your android device.
- Open the app and user will be able to see the sign in/login button as shown below. (We only support Google authentication).

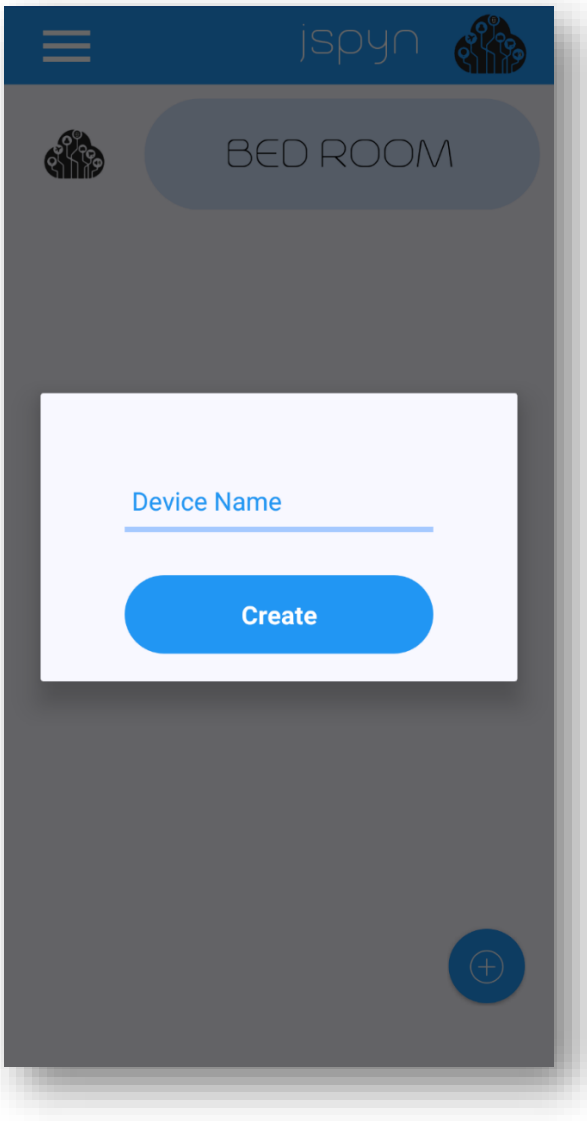



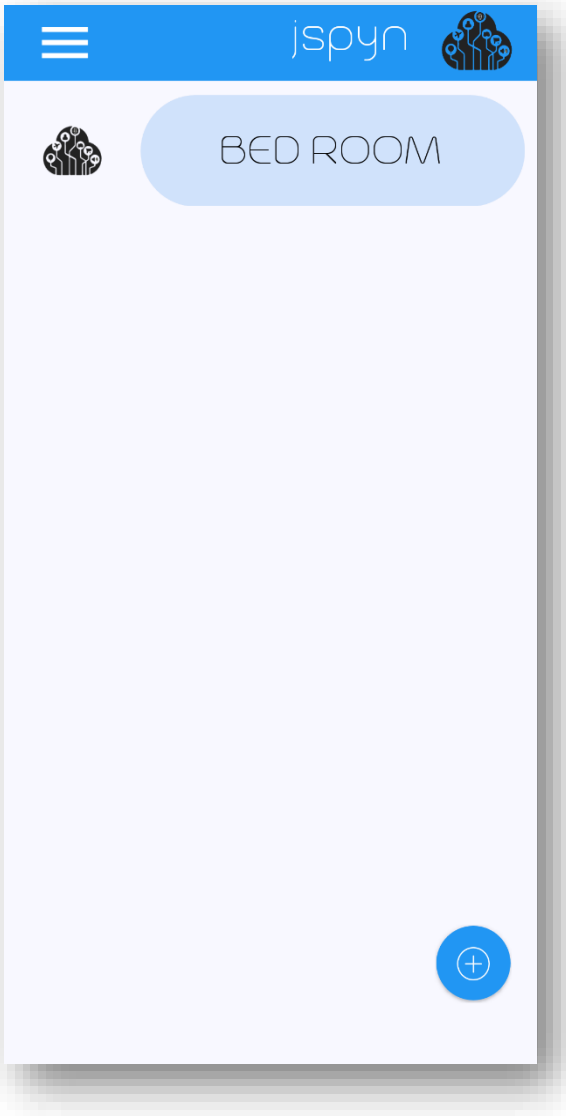
- Now the user will be able to see the floating button on their screen to add new device or user can go to the left slide drawer and click on **Add Device.**

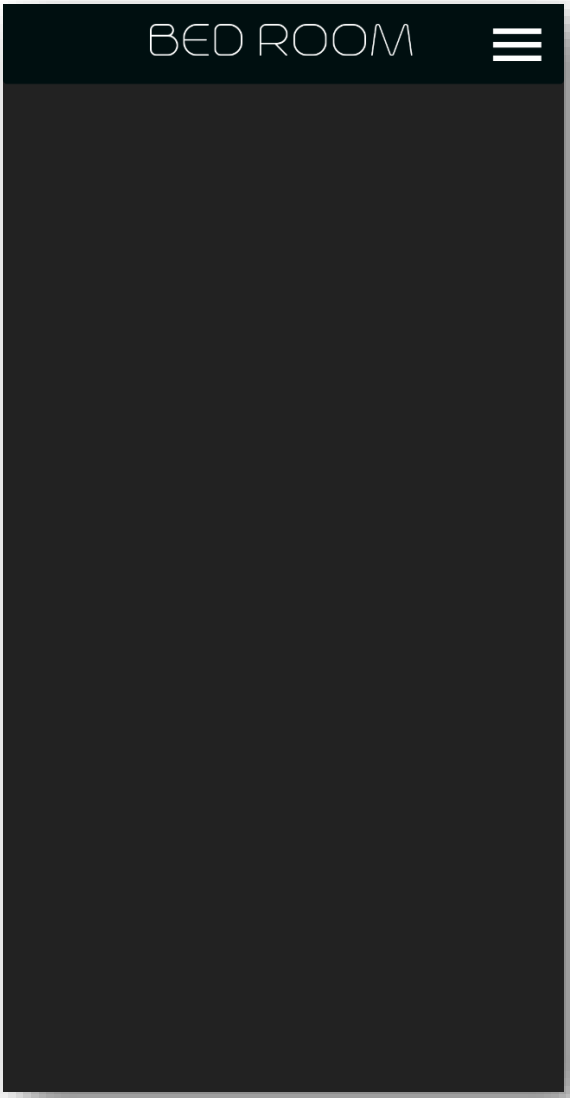
## Adding devices

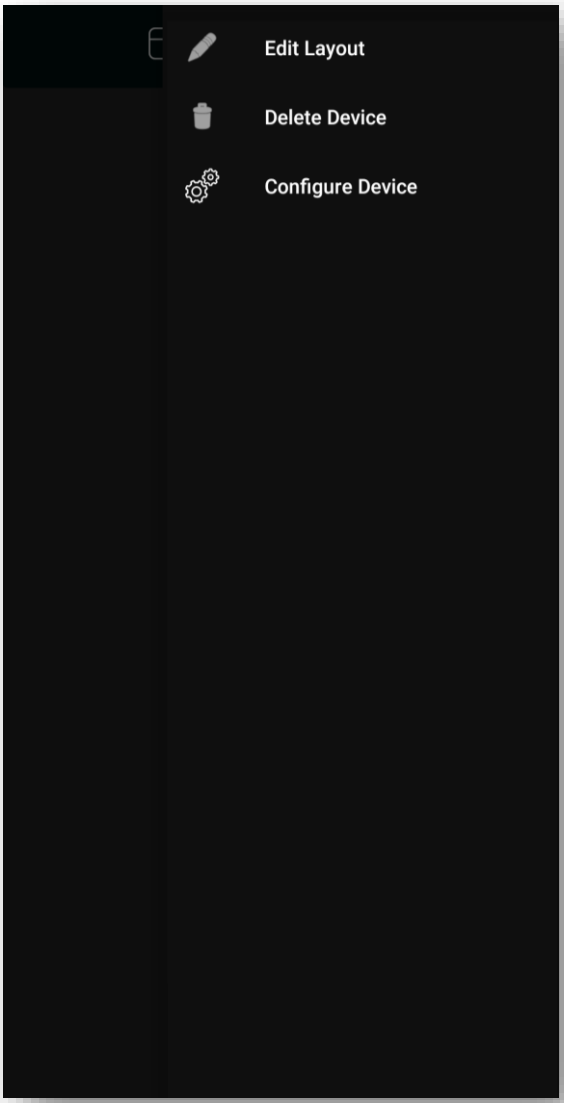
- The dialog box will appear as soon as you click.
- Give the name to the device and you will be able to see the device on the screen.
- Likewise you can add as many devices as you want and this device will be your custom apps.



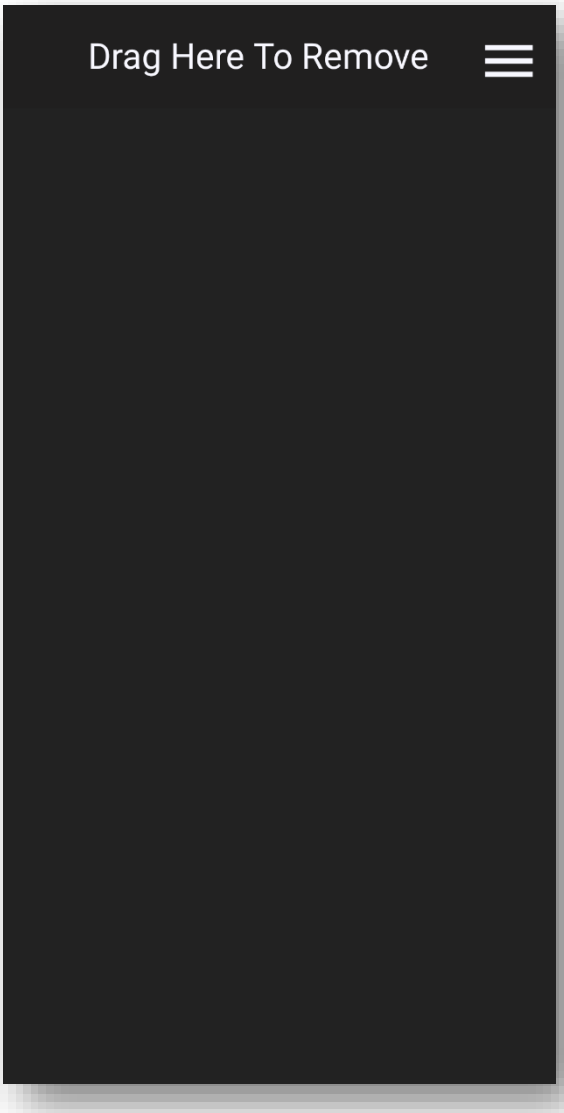



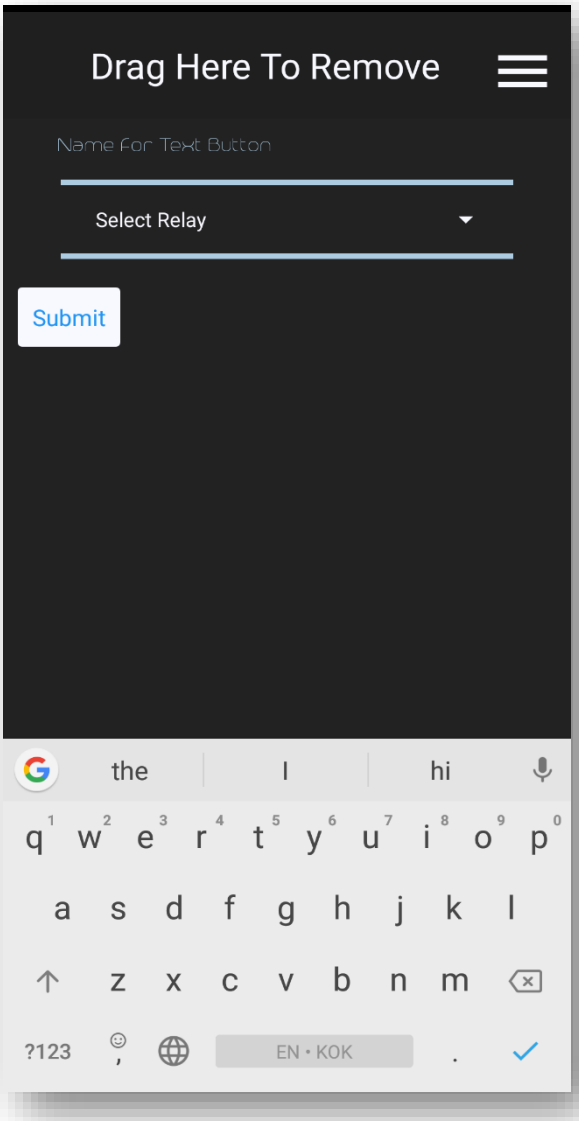
- Now click on the device and you will be taken on next screen which will be blank.
- The upright menu button will provide three options i) Edit layout ii) delete device iii) Configure device.



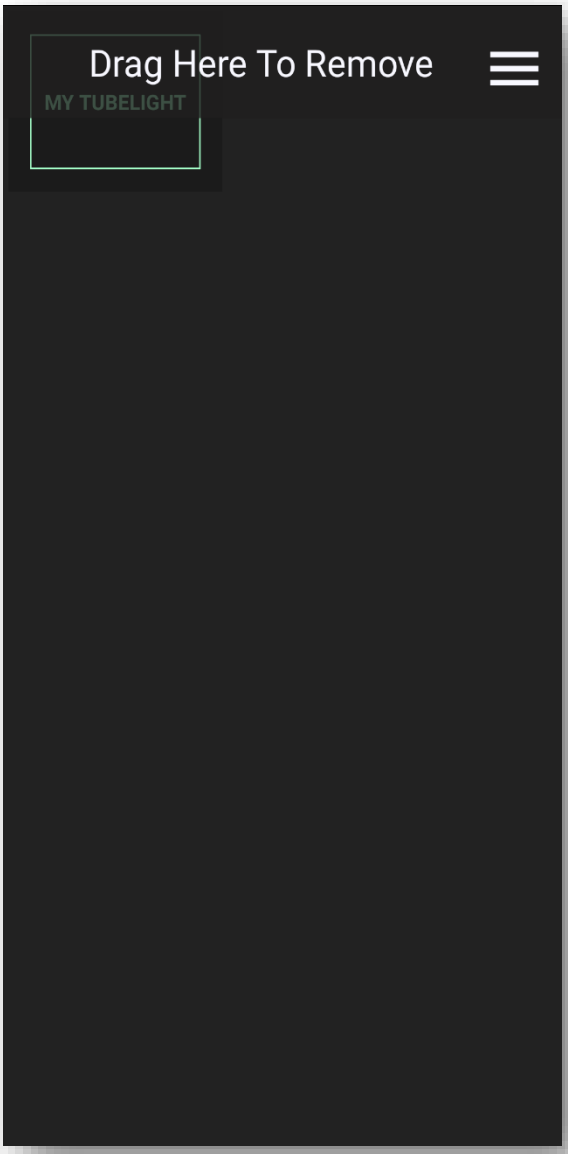



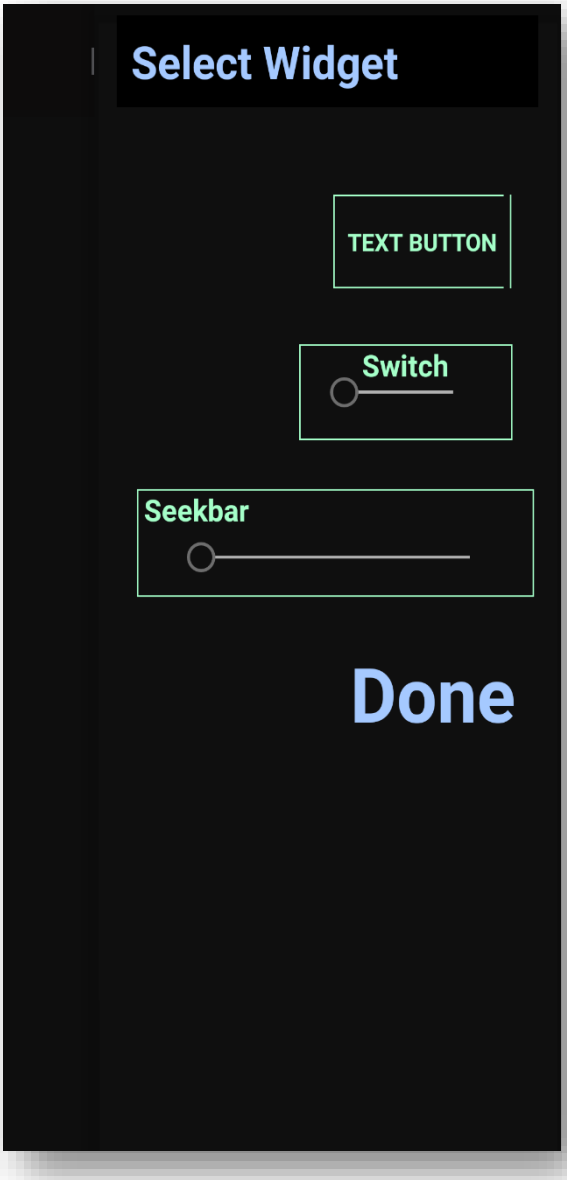
- **Edit device** will contain all the options to add widgets to point to our JSPYN IOT board with one simple form.



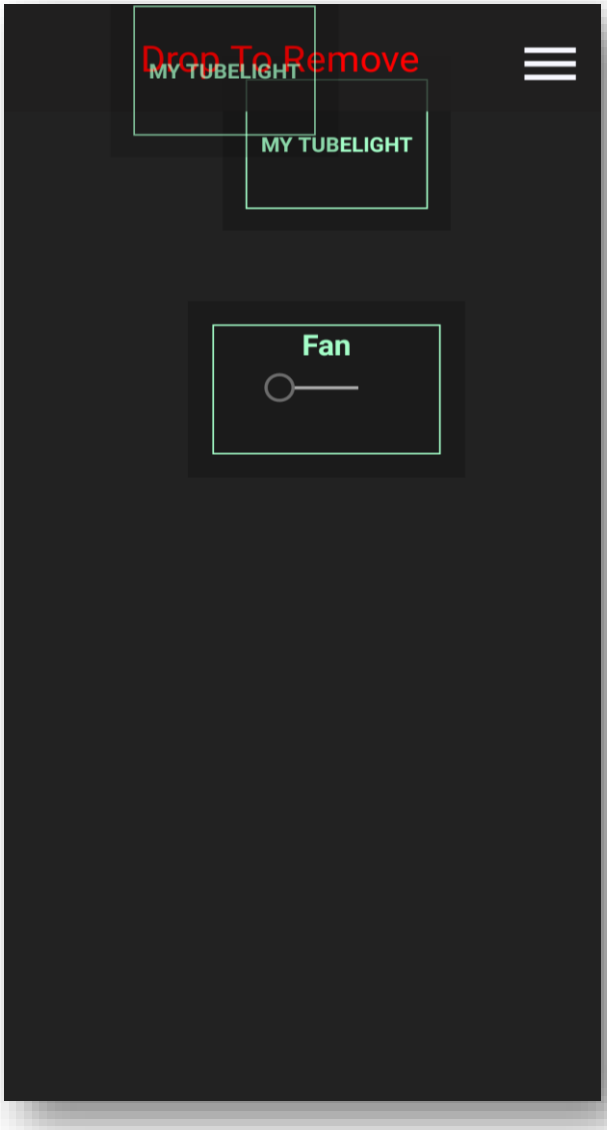



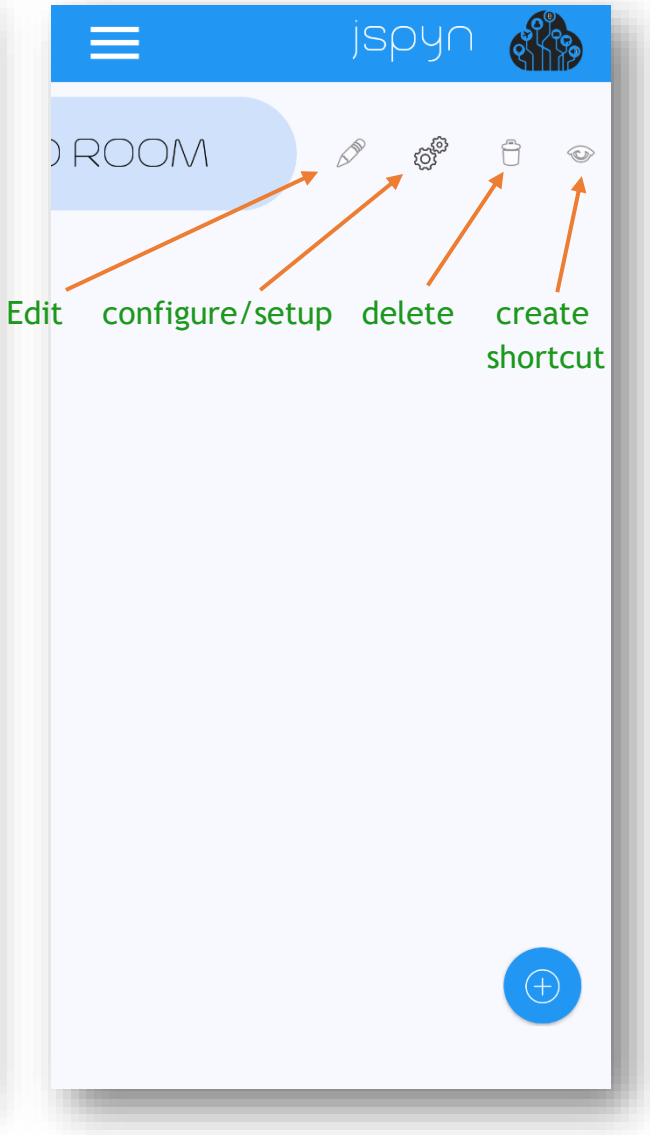
- The UI is flexible enough so that user can just design their own app with just simple drag and drop and remove any widget as and when he/she wants.



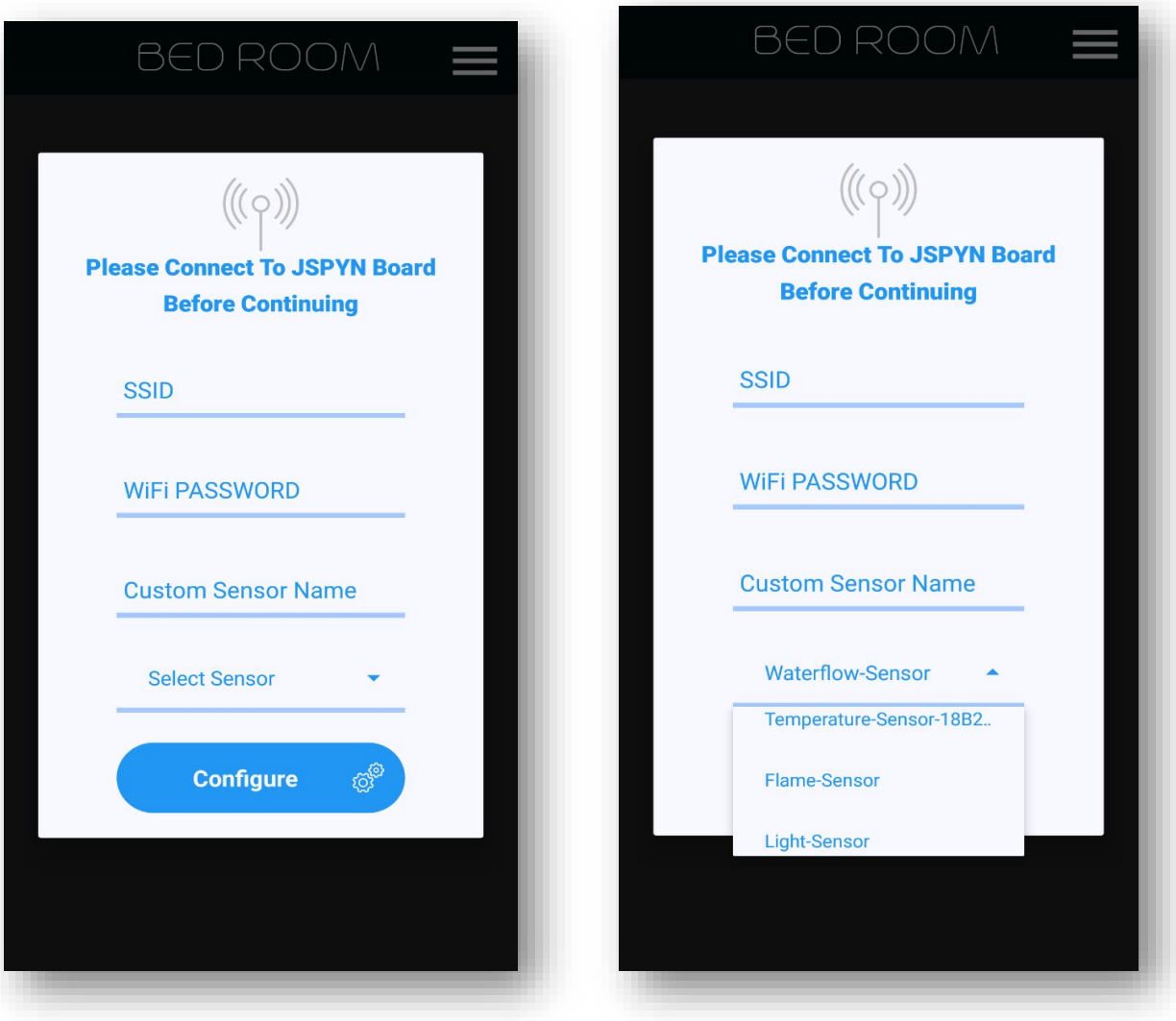
- **Delete Device** will allow you to delete your device.
- **Configure device** will allow you to Configure device explanation in the next section.
- The App has shortcuts for frequent users as shown below.
- The user can generate app shortcut using the last option above.



# How JSPYN IOT device will get Internet access and how the user will be able to Control it?

## Device configuration:

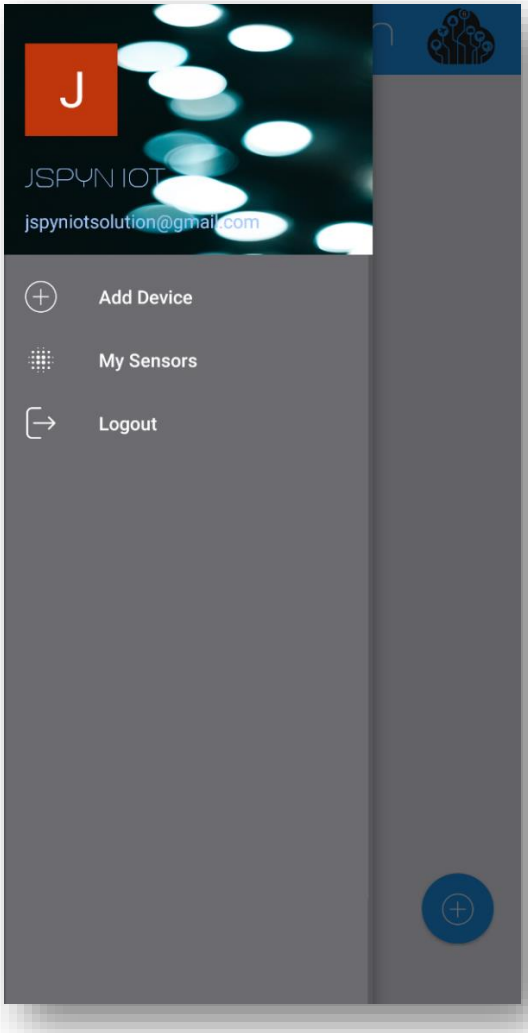


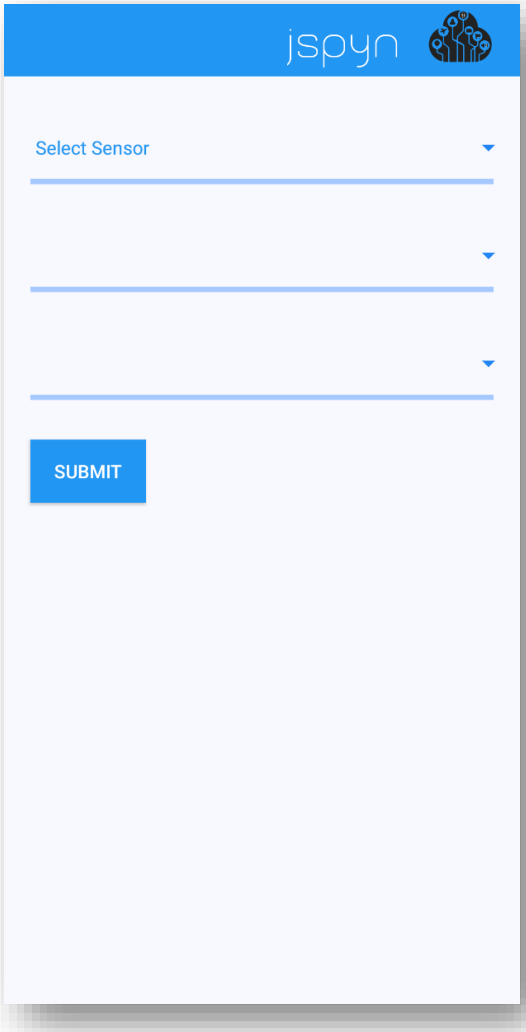

- You have already created the device and now we will call it Your Custom App.
- To connect JSPYN IOT board to the internet you have to first turn on the device.
- Now you will be able to see the green LED which will be on for 3 seconds.
- To put the device on setup mode then press and hold the Button marked **Setup** before led turn off.

- Now LED will turn off and on this means that the device is in setup mode and you can configure.
- While the device is in setup mode it will create its own Access Point with name **JspynIOT.**
- Now, go to app and open your custom App go to drawer and click on **Configure device** and fill the respective field for **SSID, PASSWORD** of your local home network or any other network which you want to connect & **CUSTOM SENSOR NAME,** & **SENSOR** (Don't select the sensor if you don't have any).
- Proceed by clicking the Configure button but before that don't forget to disconnect the internet and connect to JSPYN IOT board.
- Once configured wait for five seconds and press Reset Button and now wait for sometime till green LED stop blinking till it connects to your home network and blue LED glow constantly.
- Here you go, now your device is ready to control and upload sensor data to the cloud.

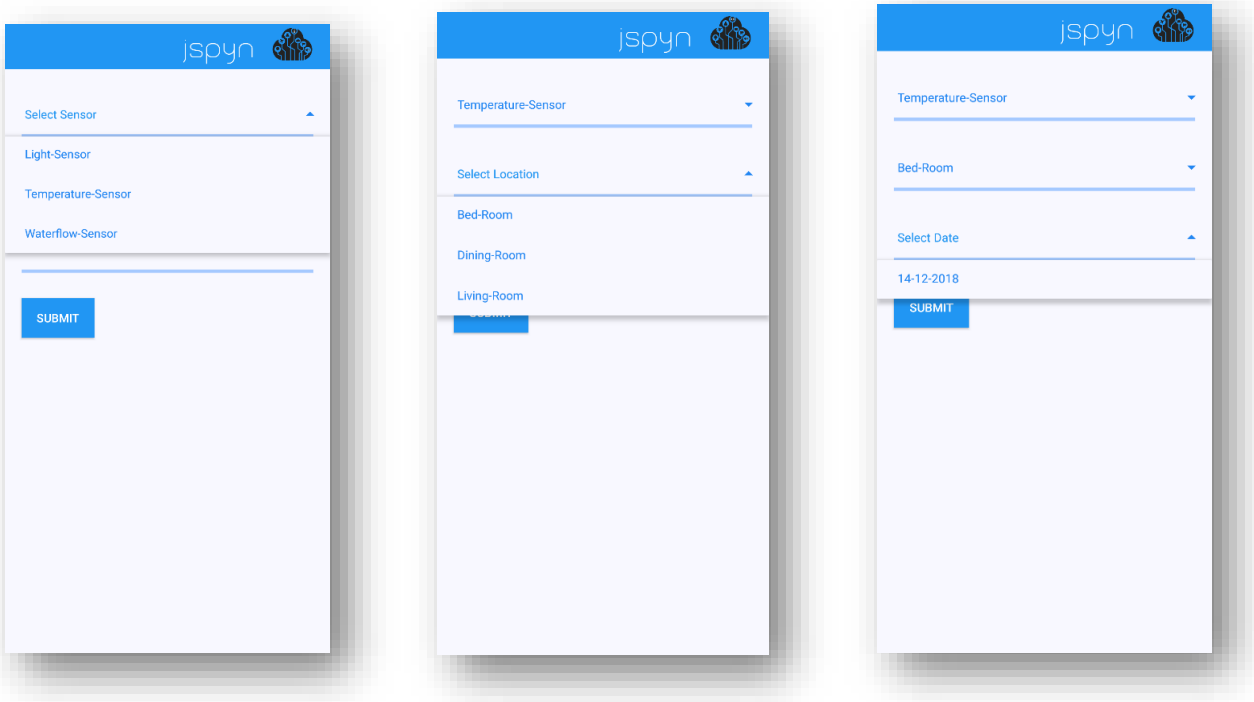
## Charts View

- Users can view collected sensors data in graphical format from the ‘My sensors’ options from the left slide drawer in the app.
- A form with drop-down menus will appear as soon as you click the option.

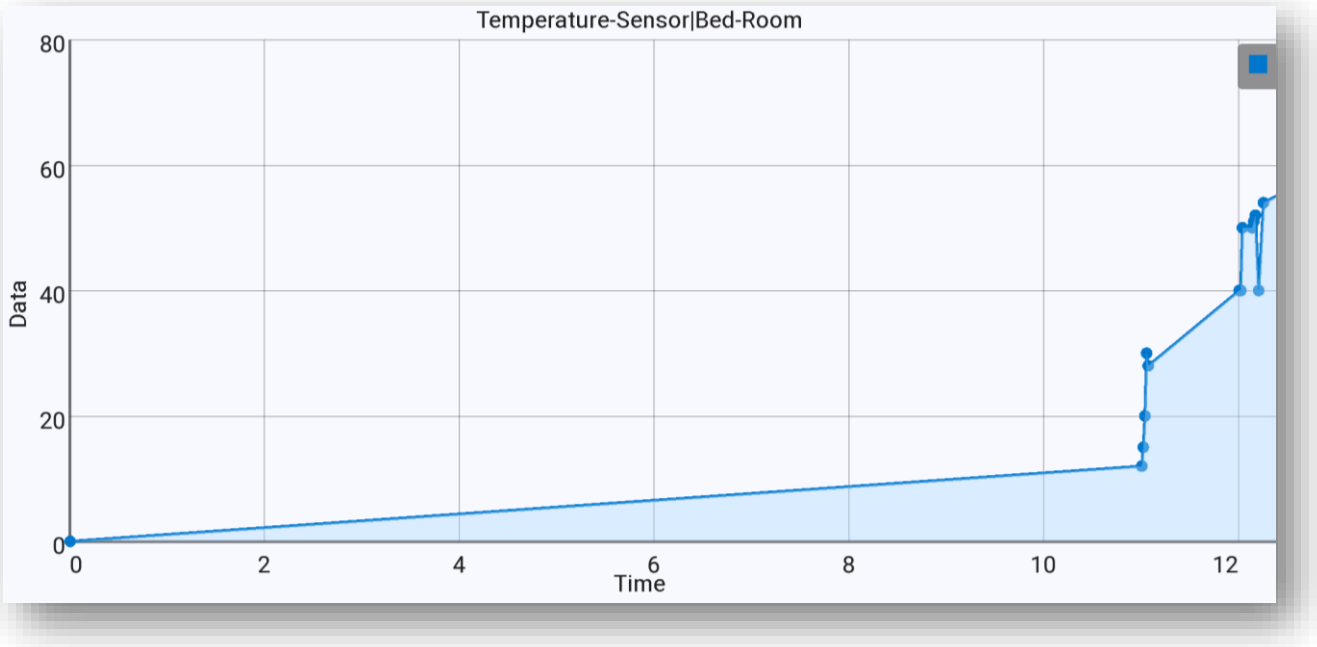


- The three dropdowns will show the sensors configured, the sensor location and the dates at which data is available respectively.




- Once submitted will display the sensor data in a graphical format for the selected options.



# 10. IMPROVEMENT AND ENHANCEMENTS

# IMPROVEMENT AND ENHANCEMENTS

Like JSPYN there are many other platforms providing similar services to IoT enthusiasts, most popular being the Blynk along with Ubidots, AWS IoT core, Pycom, etc.

Blynk being the most popular platform for IoT as it is compatible with ESP8266, Arduino, Raspberry Pi, and their own hardware called the SparkFun board which costs approximately 2000 rupees. Their mobile app provides easy connectivity to IoT boards and sensors, providing easy control to the devices and graphical data representation.

Ubidots on the other hand doesn't provide any of their hardware but gives a dashboard and a mobile app to configure, control boards and sensors. It provides a very interactive UI to control devices and view sensor data in graphical formats.

The AWS IoT core provides IoT solution with their AWS IoT Device SDK which helps easily and quickly connect hardware device or mobile application to AWS IoT Core. The AWS IoT Device SDK enables devices to connect, authenticate, and exchange messages with AWS IoT Core using the MQTT, HTTP, or WebSockets protocols.

The WiPy is a tiny MicroPython enabled WiFi and Bluetooth IoT development platform. With 1KM WiFi range, Espressif ESP32 chipset and dual processor, that lets you add sensor shields and antennas in a bundle.

From all the IoT platforms discussed above JSPYN is different in many ways. All the services above require manual connection of every sensor to every board, whereas you only need to connect your AC switch inputs to JSPYN IoT board’s respective relays and all the sensors you need are pre-connected. However, JSPYN also does have a place for developers to connect their boards through JSPYN service but it may not be as compatible like the services above. Iflets are the highlighting feature of JSPYN, which not most of IoT services providers provide. Currently, JSPYN Iflets only support push notifications, triggering relays and time trigger functionality but can support a wide variety of other actions.

As far as improvements to JSPYN is concerned, JSPYN city hub, given ample of time to collect data of a city can predict the city condition in advance with data analysis and other machine learning technologies. As JSPYN city hub can support a wide variety of sensors it can be used by farmers to find out the water logging capability or humidity of the soil by using the soil humidity sensor and even monitoring water flow during irrigation by using the water flow sensor and much more.

JSPYN IoT board, on the other hand, can be integrated with services like IFTTT and Dialog Flow to integrate voice commands. Through voice commands integration physically disabled people can also have the benefit of our service to automate their household appliances.

JSPYN is made in a cost-effective way and developed in such a way that it is not constrained with the upcoming technologies, but made in such a way that it will support them with easy integration, making it completely incompatibility free with other technologies.

# 11. REFERENCE

# Reference


- https://www.google.co.in
- https://nodejs.org/en
- https://www.npmjs.com
- https://materializecss.com
- https://expressjs.com
- https://www.tutorialspoint.com
- https://www.w3schools.com
- https://www.electronicshub.org
- http://www.circuitstoday.com
- https://www.wikipedia.org
- https://stackoverflow.com
- https://github.com