A REPORT

ON

IOT (INTERNET OF THINGS) BASED SERVER ROOM TEMPERATURE & HUMIDITY MONITORING.

BY

Names of the Students	ID.No.s
Akshay Anand	(2018B5A80274H)
Chaitanya Kulkarni	(2018A7PS0161P)
Chaitanya Vyas	(2018B4A80831P)
Kartik Vinod Jhawar	(2018B1A80654P)
Paras Taneja	(2018B2A80999P)
Saketh Kalaga	(2018B5A80929P)

AT

BHEL-HPVP(Visakhapatnam)

Visakhapatnam.

A Practice School-I Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI.
(May-June, 2020)

A REPORT

ON

IOT (INTERNET OF THINGS) BASED SERVER ROOM TEMPERATURE & HUMIDITY MONITORING.

BY

Names of the students	ID.No.	Disciplines of the Students
Akshay Anand	(2018B5A80274H)	Msc. Phy and ENI
Chaitanya Kulkarni	(2018A7PS0161P)	Computer Science
Chaitanya Vyas	(2018B4A80831P)	Msc. Maths and ENI
Kartik Vinod Jhawar	(2018B1A80654P)	MSc. Bio and ENI
Paras Taneja	(2018B2A80999P)	Msc. Chem and ENI
Saketh Kalaga	(2018B5A80929P)	Msc. Phy and ENI

Prepared in partial fulfillment of the Practice School-I Course Nos.
BITS C221/BITS C231/BITS C241

AT

BHEL-HPVP(Visakhapatnam)

Visakhapatnam.

A Practice School-I Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
(May-June, 2020)

ACKNOWLEDGEMENT

First of all we would like to thank the director of Birla Institute of Technology and Sciences, Pilani **Prof. Sudhirkumar Barai**, for giving us this opportunity to represent this esteemed institute in India's one of the largest engineering and manufacturing companies- Bharat Heavy Electricals Ltd.(BHEL), Visakhapatnam as a platform for our Practice School. We are grateful to **Sir. Amitesh Singh Rajput** for his constant guidance throughout our endeavour. Without his help, motivation, and constant support this project would have not been possible. We would also like to express our gratitude towards **Sir. Rama Krishna Sreerama** and all our BHEL mentors. Without their instructions and guidance we would have not been able to work on this project as most of us were new to this stream of Internet Of Things and so were quite reluctant to work on this project.

We also thank all our **project related guides** for their everlasting support and cooperation without which this project would have not been at this stage which was a major challenge when it comes to doing this project remotely. May it be clearing any concepts or may it be working for this report, they were always ready to help get things done together. We would also like to thank our **friends and family members** for their everlasting moral support as they were the major lifelines for us to work even being under such unpleasant Covid-19 situations.

We would like to thank the government of India and all the **BITS-PILANI faculties** who were constantly working to remove all our uncertainty related to our ongoing academic session caused due to the unprecedented challenges associated with the Covid-19 pandemic. This helped us to work on this project without any hassle.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI
(RAJASTHAN)
Practice School Division

Station: BHEL-HPVP (Visakhapatnam) **Centre**: Visakhapatnam

Duration: 6 weeks (18/May/2020 to 27/June/2020) **Date of Start**: 18/May/2020

Date of Submission: 25/June/2020

Title of the Project: IOT (Internet of Things) based Server Room Temperature & Humidity Monitoring.

Names of the students	ID.No.	Disciplines of the Students
Akshay Anand	(2018B5A80274H)	Msc. Phy and ENI
Chaitanya Kulkarni	(2018A7PS0161P)	Computer Science
Chaitanya Vyas	(2018B4A80831P)	Msc. Maths and ENI
Kartik Vinod Jhawar	(2018B1A80654P)	MSc. Bio and ENI
Paras Taneja	(2018B2A80999P)	Msc. Chem and ENI
Saketh Kalaga	(2018B5A80929P)	Msc. Phy and ENI

Name: Sir. Rama Krishna Sreerama

Designation of the expert: (Dy. Engineer/HPVP_DTG/HPVP)

Name of the PS Faculty: Sir. Amitesh Singh Rajput

Key Words: Internet of Things, DHT11 temperature and humidity sensor, ATmega328 microcontroller based Arduino Uno board, ESP8266 WiFi module, Tinkercad simulation software, Server-client communication, Web page designing.

Project Areas: The project forms the basis for many projects. It can be used for automatising the irrigation timings in an agricultural field based on the humidity present in the soil. The project can also be extended to automatise the cooling system of a room based on the temperature conditions of the room.

ABSTRACT

This report aims at giving an idea of how the Internet of Things has evolved to serve the need of this generation to automatise all the possible works. The report has been designed in such a way that it can give an overview of the work associated with the stream of Internet of Things by the medium of a project. This project basically aims at monitoring the temperature and humidity of a remotely located room over an android app or a web page.

The report explains in detail about the essential components used to accomplish this project along with their functioning in detail. It also explains the use of some popular hardwares like the Arduino Uno board, the ESP8266 Node-MCU, and the DHT11 temperature and humidity sensor. The report also talks in detail about the front end part of Web Development along with a brief discussion on android app development. There is a brief discussion in the conclusion part of the report about the results we got after performing every milestone of our project along with some of the challenges we faced and how we went on advancing in the project. The readers can also refer to the sources used to gather the information in the reference section and also refer to the glossary for meanings of some important difficult words. The report has been kept interesting and easily understandable for the readers by using proper demonstrating images and editing styles.

Signature of the Students- **Signature of the PS Faculty**

Table Of Contents

INTRODUCTION

❑ OVERVIEW OF THE PROJECT

Internet of Things or IoT is basically the interconnection of various devices over the Internet or any network such that they can share information among themselves without requiring any human to human or human to device interactions. The IoT has made the dream of humans of this generation to reduce the complexity of any work onto his fingertips or even without the use of fingertips, come true.

The project we are currently working on gives a glimpse of exactly how this dream of humans has come true or what exactly the Internet of Things can do. The project is to monitor the real time temperature and humidity of a room which is remotely located, over an Android App or on a web page. The purpose is served by using a temperature and a humidity sensor which are electronic devices capable of sensing temperature and humidity in the surrounding air and responding by providing output signals generated due to varying output voltages based on the changes in the surrounding parameters. By using some mathematics the voltage values can be linearized and mapped to exact temperature and humidity values. The sensors are connected to an external hardware known as an IoT board having the capability to connect to a network and a programming device using wireless or wired connectivity. We are using in this project the most popular microcontroller based board i.e. Arduino Uno connected with a ESP8266 WiFi module. We call this complete interconnected hardware having a board, WiFi module and sensors as one of many IoT devices.

The rest of the project deals with the use of a network such as an Internet or an intranet. The IoT device is made as a client using socket programming or simply by using libraries included in the Arduino IDE software. The client then requests the IoT server or the cloud or the database such as the Google Firebase, to store this real time sensor readings. The web browser client then simply requests this data

from the IoT cloud and displays it. In the same way, the android app client also requests this data from the IoT cloud and displays the real time readings in mobile upon connecting to the Internet.

❑ Applications Of The Project

This project helps form the basis of many real time projects which one can make to simplify life. By just setting up the same hardware somewhere in your garden using an external power source and programmed IoT board can help you plan your diving out of home by giving you real time predictions of the weather.

The same project could be assembled in the main MCB or the Miniature Circuit Breaker of your industry or home such that it can alarm you whenever the temperature of the MCB goes high signalling to overvoltage and the risk of fire. You may extend this project to even activate a fire extinguisher using actuators and relays say when the temperature of the room or an industry area goes above a certain set value implying the catch of a fire. In the similar way you can also use this IoT device to automate the irrigation timings in an agricultural field and even the system can be made more smarter by connecting it to a weather forecasting model such that irrigation can be reduced when there is a prediction of rain in the near future.

❑ Something About Bhel-Hpvp

Bharat Heavy Electrical Limited (BHEL) Heavy Plates & Vessels Plant, formerly Bharat Heavy Plates and Vessels Ltd (BHPV). On 30.08.2013 Bharat Heavy Plates & Vessel (BHPV) was merged as the 17th unit of BHEL and named as BHEL-HPVP.

It is the largest fabrication industry for fabrication of process equipment and Boiler plants, in India required for fertilizers, petroleum, chemical, petro-chemical, Steel and allied industries. BHEL HPVP is equipped to manufacture a wide variety of fabricated product lines for the process industries like oil gas, space, defence etc., either to its own design or to the specification of customer's consultant. BHEL is

experienced in the design of columns, multi-layered vessels, Heat Exchangers, Liquid Oxygen and Nitrogen Unit, Evaporation Plants, Digesters, Mounted Vessels, Sulphur Recovery Unit, Gas Dehydration, Crude Stabilisation Unit, Storage Vessels etc. Some of the major functional departments in BHEL-HPVP are Administration, HR Finance Marketing, Project Management, Logistics and Material Management, etc. which all work collaboratively such that BHEL can serve its best services to its consumers.

❑ Hardwares Used

1) IoT Board

For this project, the team after consideration of various microcontroller boards, more appropriately IoT boards had decided to use the Arduino Uno microcontroller board instead of Raspberry Pi and other related boards which consumes a lot of power as compared to Arduino Uno board. It has an ATmega328 microcontroller which is key to store all the programs needed for its function.

2) Sensors

Another important factor is the availability of compatible sensors with our chosen microcontroller board. The primary task in this project is to measure temperature and humidity and we need the relevant sensors which can perform this.

We had to choosen DHT11 temperature and humidity sensors from all the other types of relevant sensors since it is digital, can measure both temperature and humidity and is easily available too.

3) ESP8266 Node-MCU

Since our project involves the communication of our board with the internet to relay the appropriate information, we need to give our board the capability to communicate with the internet. We can either use an ethernet shield with the Arduino Uno board, or we can add a Wi-Fi module ESP8266 to our board which helps connect the board to the internet using the Wi-Fi connection at the organization. In the project, we used a NodeMCU microcontroller board instead of any other Wi-Fi modules to connect to the internet and relay the temperature and humidity information.

The way we communicate and program the Arduino Uno microcontroller is using the Arduino IDE. This is the environment where we write the programs, compile our sketches, rectify errors in the sketch, debug our sketch and finally upload this sketch to our board. This environment can be used to program different families of boards which are not exclusively a part of the Arduino family as well. We can use this to program boards like NodeMCU as well.

❑ Design Philosophy Of The Project

The way our board works is the flow of information in the following way:

1) The temperature and humidity information of the surroundings is collected by the DHT11 sensor.

2) This information is sent as digital information to a digital pin on the Arduino Uno board since the DHT11 sensor is a digital sensor.

3) We use the onboard processing on the Arduino Uno to interpret this information and store this temperature and humidity information in variables of appropriate data types.

4) Using the Wi-Fi module (the NodeMCU microcontroller in this case) which connects to the internet, we push this information to the servers which host our data.

5) This stored information on the servers helps us serve the website and the Android App which is used to monitor the temperature and humidity.

❑ Subtleties Of The Project

1) DHT11 Sensor-

If the sensor does not come as a prebuilt module with a step-up resistor, then a circuit with an appropriate step-up resistor is built and then the data pin is connected to the digital pin on the Arduino Uno. Failing to do this will cause the sensor to overheat, and malfunction, finally resulting in the sensor being rendered useless by its melting and damages to the Arduino Uno board.

2) Interpreting The Input At The Digital Pin-

The digital output of the sensor is sent using a communication protocol and instead of writing extremely long sketches by analyzing each bit sent by the sensor, we can instead make use of the DHT11 library available and meant to be used while using the DHT11 sensor. Including this library makes it easier for us to use the sensor and collect information from it.

3) Connecting To The Internet-

The primary method to connect to the internet is to use the Wi-Fi module ESP8266. Proper usage of the ESP8266 involves a voltage source at 3.3V which has the capability of providing a maximum current of more than 250mA. The transience of the current draw curve is very small, and the peaks are very sudden. The voltage controller circuit on the Arduino Uno is not a specifically strong one, and it handles such sudden peaks in current draw poorly and this might cause the Wi-Fi module to crash causing loss of connectivity which is to be avoided (which reference to the 3.3V pin). Even though there is a 3.3V source pin on the Arduino Uno, it can only handle a peak current draw of around 100mA, which is nowhere near the necessary current required to run the ESP-8266 module. So, we rejected using a WiFi module.

As an alternative we used NodeMCU, which has a built-in ESP8266 module. One could implement the DHT11 sensor directly with the NodeMCU, but due to lack of processing power in the NodeMCU microcontroller, and small no. of pins, what can be done is send serial information between the Arduino Uno and NodeMCU where the serial information is the temperature and humidity data, and the NodeMCU connects to the internet to send this information to the server or the database.

4) Serial Communication Between Arduino Uno And Node-Mcu-

We can connect the TX pin of the Arduino Uno board to the RX pin of the NodeMCU board and same for the RX pin as well as to begin serial communication between the Arduino Uno and the NodeMCU. It is said that the TX pin from the Arduino Uno can be directly connected to the NodeMCU as the maximum tolerable voltage for NodeMCU microcontroller boards is 5V and the output from the TX pin from Arduino Uno never reaches 5V though this has some controversy but we used this method and got the desired results.

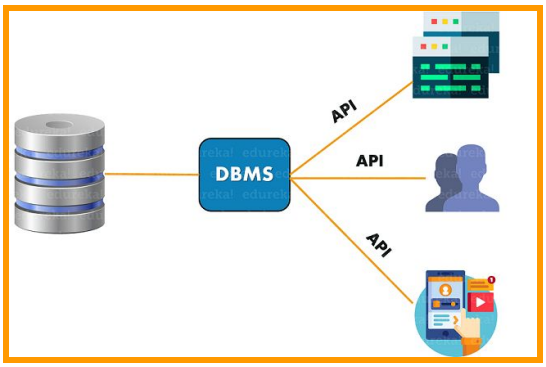
5) Using Google Firebase Database-

Since there was no access to the BHEL servers during the remote PS; the team decided to use the Google Firebase Database and website hosting service which stores the temperature and humidity information in the cloud, and this data in the database will be displayed on the website and the Android App. The SSID and the password of the local area the microcontroller board connects to are hardcoded in the sketch of the NodeMCU, along with other relevant information like the address to our database along with security information required to be able to successfully log information into the database.

DATABASE

INTRODUCTION

A **database** is a memory storage to store a collection of datas which is alterable. Whereas, a **DBMS** or Database Management System is the software that interacts with the applications like web sites, and android apps which are operated by the end users. DBMS basically provides access to the database and allows users to insert, store, delete or retrieve his data from the database. Well, due to the close relationship between the database and the DBMS, they both are referred casually as "database".



An **API** or Application Programming Interface is a tool set developed by the developers of that DBMS so as to link the database with the backend of your application. They basically guide the software components in the way you want them to interact with your application.

❑ Classification of the Databases

Databases and **DBMS** can be classified based on the various parameters such as based on the query language used to access the database (like, SQL, NoSQL), and also based on the database models they support such as relational or XML. Hence the division of databases are as follows-

➢ **SQL or relational databases:** SQL databases manipulate the data in the database using Structured Query Language (SQL). SQL has its predefined schemas which allows you to structure your data in the form of tree or relational data such as in the form of tables. But, it has a limitation that a change in small data would change the whole structure of your data and hence would be difficult for the system to manipulate the data.

➢ **NoSQL or non- relational databases:** A NoSQL database has its own dynamic schema to store your data in an unstructured manner. The data in turn can be saved in ways like document-oriented, column-oriented, graph based and object-oriented. This flexibility allows you to create unique structures of data, update the data easily and even makes retrieval of data easier for even the real time retrieval.

❑ What Is The Need Of A Realtime Database?

The database is kind of a server connecting to various clients using the Internet as the network. The aim of this project being to display the real time and stored readings sensed by the sensor located at a remote place can only be executed by using a global network which is an internet. The IoT device can be connected with

this server and can be programmed to **update this database with the sensed readings** such that these **readings can be retrieved** by any client connected to this server so as to allow monitoring of those readings remotely.

❑ WHY FIREBASE REALTIME DATABASE ?

➢ **BaaS:** BaaS or Backend-as-a-Service is a cloud service model in which developers frame all the libraries to be used and create a console which can allow the user to control the front end part with ease where all the behind-the-scenes aspects of web or mobile applications are controlled by the BaaS.

➢ **Easy to use**: It is an open source platform that gives you the snippets of already written codes in JavaScript, C programming and various other languages.

➢ **Documentation:** Firebase has its own guide book in the form of a documentation where it guides through every step from linking the firebase database to your application, to allowing the user to write and retrieve data and even allowing the retrieval of the real time datas.

➢ **Various services:** Firebase not only has realtime database service but also has services like cloud storage, authentication, hosting and Machine Learning platform, etc. So, using firebase for its database now shall help you to work with its other services with ease if you need them in the near future.

➢ **Uses NoSQL:** The firebase database uses NoSQL type of database which makes it a complete JSON (JavaScript Object Notation) object where you can store and retrieve anything just by creating a new node or by appending an existing child element without even making the system's task difficult by not using structured data like tables.

WEB DEVELOPMENT.

INTRODUCTION

A web application is a computer program that utilizes web browsers and web technology to perform tasks over the Internet. Web applications use a combination of server-side scripts (Java Servlets) to handle the storage and retrieval of the information, and client-side scripts (JavaScript and HTML) to present information to users. This allows users to interact with the company using online forms, content management systems, shopping carts and many more. In addition, the applications allow employees to create documents, share information, collaborate on projects, and work on common documents regardless of location or device.

HOW A WEB APP WORKS ?

1. The user accesses a web application via a web browser or mobile application, triggering a request to the web server over the Internet. Note that there may be security measures (i.e. firewalls or cloud access security brokers) and load balancers blocking the clients to request the servers.

2. The web server performs the requested task – such as querying the database or processing the data – then generates the results of the requested data.

3. The web server sends the results of the requested data back to the web client and this information appears on the user's display.

❑ Basics Of Back End Part Of Web Development.

Backend is the server side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend. For our project, we are using Google Firebase database as the back-end controller for our front end and so, we haven't discussed anything about the Java servlets but we have discussed how the data is processed by the database in the further sections.

❑ Basics Of Front End Part Of Web Development.

Part of a website that the user interacts with directly is termed as front end. It is also referred to as the 'client side' of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and navigation menu.Responsiveness and performance are two main objectives of the front end.

Some of the languages used to design our website are as follows-

❑ HYPER TEXT MARKUP LANGUAGE

HTML stands for **HyperText Markup Language**.

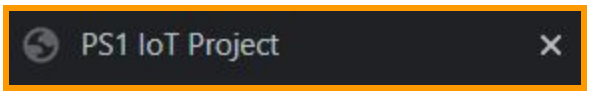
- **HyperText** is the method by which you move around on the web — by clicking on special text called **hyperlinks** which bring you to a particular page. The fact that it is *hyper* just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.

- ***Markup*** is what **HTML tags** do to the text inside them. They mark it as a certain type of text (*italicised* text, for example).

- HTML is a ***Language***, as it has code-words and syntax like any other language.

❑ APPLICATIONS OF HTML IN OUR PROJECT

A) HEADLINE

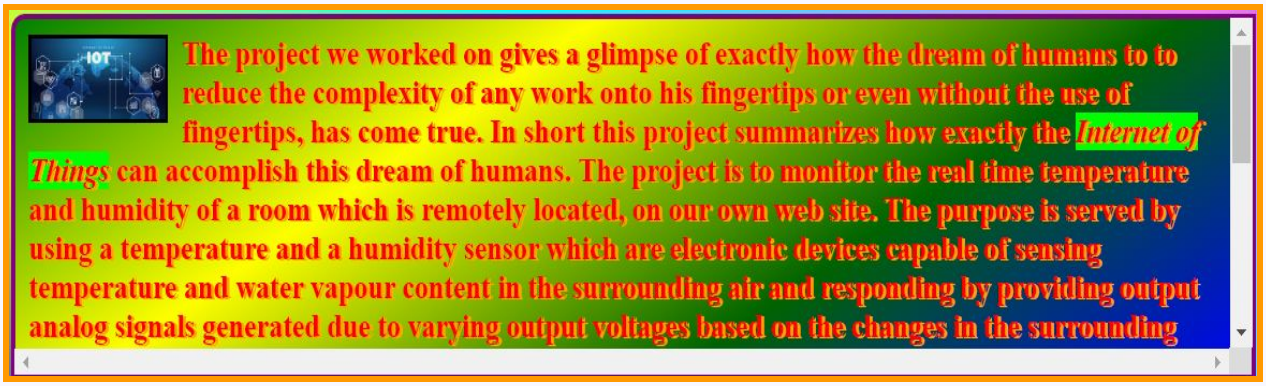


```
<!doctype html>
<html>
<head>
    <title>PS1 IoT Project</title>
    <body>
<div class="main" >
    <img   src="bhel.png" >
    <h1 class="heading">Web page for temp and humid display.</h1>
    <h1 class="heading1">  IoT Android App.</h1>
</div>
<p class="red underline" style="font-size:18px;"> <i> IoT Project</i></p>

    <p class="underline "> <b><i> I emphasis you all to surf this site to get a dynamic experience of Internet Of Things.</i></b></p>
<br>
```

This is the part of code used in the webpage formation showing the title of our webpage is “PS1 IoT Project” and showing the BHEL logo on the extreme right of the webpage. It has two headings showing “webpage for temperature and humid display”. There is a paragraph showing the “IoT project” in red underline.”<img>” tag is used to put the image on the website.

B) Scrolling Box

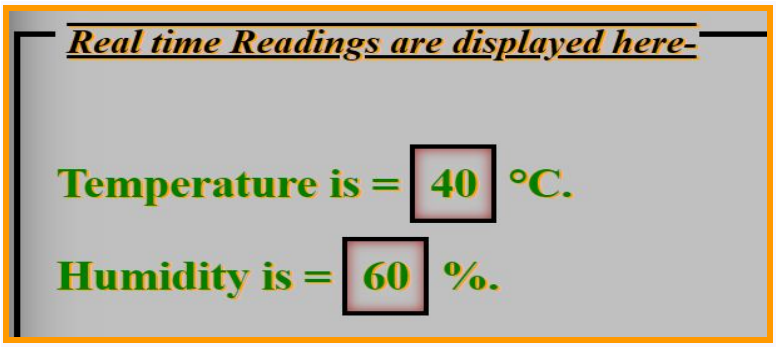



```
<p style="font-size: 1.3em; color:red; width:60%;height:200px; overflow: scroll;border:solid 3px purple; padding:10px; border-radius: 10px 0 0 0
background-image: linear-gradient(150deg, green,yellow,darkgreen,blue); ">

        <img class="box" src="images.jpg" style="display:block; float:left; margin-right: 10px; border: solid 2px black; width:100px;">
        The project we worked on gives a glimpse of exactly how the dream of humans to
         to reduce the complexity of any work onto his fingertips or even without the use of fingertips, has come true.
         In short this project summarizes how exactly the <span class="highlight">Internet of Things</span> can accomplish this dream of humans.
          The project is to monitor the real time temperature and humidity of a room which is remotely located, on our own web site.
           The purpose is served by using a temperature and a humidity sensor which are electronic devices capable of sensing temperature and water vapour content in
            the surrounding air and responding by providing output analog signals generated due to varying output voltages based on the changes
             in the surrounding parameters.
            By using some mathematics the voltage values can be linearized and mapped to exact temperature and humidity values.
             The sensors are connected to an external hardware known as
             an IoT board having the capability to connect to a network and a programming device using wireless or wired connectivity.
             We are using in this project the most popular microcontroller based board i.e. Node-MCU or ESP8266.
              We call this complete interconnected hardware having a board,
              sensors and network connecting this device to the Internet as one of many IoT devices.
The rest of the project deals with the use of a network such as an Internet or an intranet.
 The IoT device is made as a client using socket programming or simply by using libraries included in the Arduino IDE software.
  The client then requests the IoT server or the cloud or the database such as the Google Firebase, to store this real time sensor readings.
   The web browser client then simply requests this data from the IoT cloud and displays it.
</p>
```

This part of code is creating a scrolling box which has an IOT image at the left side and has text written inside the paragraph explaining about the project and IOT in brief. “<p>” is a paragraph tag and we have provided a style attribute inside that in which styles the font size , color etc. of the paragraph written under the <p> tag. We also added borders through this. The given “linear-gradient” attribute provides the background image in an inclined manner at an angle of 150 deg and has a blend of green ,yellow and blue.

c) Results panel




```
<div  class="cssbox" style="background-color: silver; padding: 15px; font-size: 30px; overflow: scroll; border:solid 4px black; max-height: 525px;" >
<fieldset style="border:solid 4px black;">
<legend>    <h1  ><b><i class="underline">Real time Readings are displayed here-</i></b></h1></legend>

<div class="newbox">
<per class="reduce" style="text-shadow: 2px 1px orange; color:green; font-style: bolder;">
Temperature is = <span id="tempDB" class="humibox"></span> °C.</per><br><br>


<per class="reduce" style="text-shadow: 2px 1px orange; color:green; font-style: bolder;">Humidity is = <span id="humidDB" class="humibox"></span> %.</per><br>
<br>
</div>
</fieldset>
```

The code shows the layout of the boxes which are showing temperature and humidity.Which takes the reading from the server. A class is constructed which constitutes the css work of the boxes. “Id” is provided to the temperature and humidity part so that the value can be given to them through the server by passing their Id. The advantage of constructing class is we can access a particular section without disturbing the other classes.

❑ CASCADING STYLE SHEETS

Cascading Style Sheets, fondly referred to as CSS, is simple a designing language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

FEATURES OF CSS

- **CSS saves time** – You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

- **Pages load faster** – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

- **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

- **Superior styles to HTML** – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

- **Multiple Device Compatibility** – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

❑ Applications Of Css In Our Project

```
body{

    font-weight:bold;
    background-color: grey;
    background-clip: border-box;
    background-image: linear-gradient(30deg, green,purple,darkgreen,blue,yellow,cyan,violet,indigo,red);
}
```

This part of CSS code is showing all that color blend in the background of the webpage in which the colors are in a linear gradient aligned at an angle of 30 degrees.

a) Buttons

```
}
#nav{
    border:solid 4px black;
    box-shadow: inset 0 0 50px grey;
    padding:10px;
}
#nav ul{
    list-style-type: none;
}
#nav li{
    display: inline-block;
}
```

This is for the styling and alignment of the buttons in the webpage. Border attribute is used and 4px width is shown. The display of the buttons are in Inline blocks. To align them all horizontally side-by-side.

The code given in the first picture shows the button styling when it is not clicked.Its color shown is lime with the shadow which appears at the edges are dark green. The letters are in block form and margins are given.

```
ul li a{
    background-color: lime;
    box-shadow: inset 0 0 15px darkgreen;
    display: block;
    width:175px;
    margin-right: 20px;
    margin-top:auto;
    margin-bottom: auto;
    padding: 10px;
    text-decoration: none;
    border:solid 4px black;
    text-shadow: 4px;
    height:25px;
    line-height: 25px;
    vertical-align: middle;

}
```

```
ul li a:hover{
    background-color: skyblue;
    box-shadow: inset 0 0 15px blue;
    position: relative;
    top: -0.3em;
    color:red;
    border-bottom: groove 6px grey;
    border-left: groove 6px grey;
```

The second one is the code that changes the button colour and its font color to sky blue when we click that button and gives the button a 3-D look when we hover over the buttons. The “top=-0.3em” shows the look as it will go above 0.3 units from the page so as to give a 3D look. Borders and color of the text is also shown as the color changes to red.

B) RESULTS PANEL

When we click that developer button all the names of our group members pop up this is done by the given CSS code.

The given figure shows the above statement and the photos are taken from the web page itself.

```css
/* The actual popup */
.popup .popuptext {
  visibility: hidden;
  width: 225px;
  background-color: #555;
  color: #fff;
  text-align: center;
  border-radius: 6px;
  padding: 8px 0;
  position: absolute;
  z-index: 1;
  bottom: 125%;
  left: 65%;
  margin-left: -80px;
}
```

```css
/* Popup arrow */
.popup .popuptext::after {
  content: "";
  position: absolute;
  top: 100%;
  left: 50%;
  margin-left: -5px;
  border-width: 5px;
  border-style: solid;
  border-color: #555 transparent transparent transparent;
}
```

The code given in the figure shows how the name is popped up when the “developer” button is clicked. It shows the alignment of the names in that pop up and its size. The background color and padding of the pop up is given.

The edges of the pop up message have a curved edge with “radius=6px”.the left margin is -80px and the position is absolute.

```
/* Toggle this class - hide and show the popup */
.popup .show {
  visibility: visible;
  -webkit-animation: fadeIn 1s;
  animation: fadeIn 1s;
}
```

```
/* Add animation (fade in the popup) */
@-webkit-keyframes fadeIn {
  from {opacity: 0;}
  to {opacity: 1;}
}
@keyframes fadeIn {
  from {opacity: 0;}
  to {opacity:1 ;}
}
```

These sets of code create an animation type situation of the name as the names appear after 1 second when the button is pressed as the opacity is "0" at the starting and after 1 second it goes to "1" so that the names can be seen.

❑ JAVASCRIPT

JavaScript (often shortened to **JS**) is a lightweight, interpreted, object-oriented language with first-class function, and is best known as the scripting language for Web pages, but it's used in many non browser environments as well. It is a prototype based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles.JavaScript runs on

the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event.

WEB PAGE TO DISPLAY SENSOR READINGS.

❑ LINKING OF THE WEB APP WITH THE FIREBASE DATABASE.

```
<script type="text/javascript">

  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyDh-DoUBBKYi-imXRCddHllodhiqhQI3Tg",
    authDomain: "tempandhumid123654789.firebaseapp.com",
    databaseURL: "https://tempandhumid123654789.firebaseio.com",
    projectId: "tempandhumid123654789",
    storageBucket: "tempandhumid123654789.appspot.com",
    messagingSenderId: "416102102243",
    appId: "1:416102102243:web:b6f1b8a06865419ee701d6",
    measurementId: "G-XGVRVE47VD"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
```

1. To start using Firebase database for your web application you first need to register your app in the Firebase by **creating a web project** in it. The Firebase SDK provided along with the configuration APIs must be pasted in the JavaScript file of your web page.

➔ **Web APIs:** It is simply a set of instructions, just like the personal computer API, but based in the web space. For instance, when users interact realtime on a Firebase database using a web app, they're utilizing an API to write, to store their data, to create new data, to delete existing data, and so forth. Ultimately, APIs reduce the developers job by removing the need to write the server-side code.

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.15.1/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.15.1/firebase-analytics.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.15.1/firebase-database.js"></script>
```

2. An **SDK** or Software Development Kit are the set of tools, libraries, relevant documentation, code samples, and processes. In short an SDK is a fully-fledged workshop which allows developers to create software applications like the web app. The given three *<script>* contents are the libraries provided by the firebase to read,

write, analyse, and debug your data in the database using its specific functions which makes updating or reading even the real time data easier. One among those various functions is the *firebase.initialzeApp( )* which sets the connection between this web client and the firebase server by verifying the configuration details provided by the web client

❑ Programming The WebPage Using JavaScript And Firebase Libraries To Get The Real Time Updates Of The Readings.

```
116  var database=firebase.database();
117
118  var tempChangeRef= firebase.database().ref('Temperature/');
119  var humidChangeRef= firebase.database().ref('Humidity/');
120
121  var tempChange=document.getElementById("tempDB");
122  var humidChange=document.getElementById("humidDB");
123
124
125  tempChangeRef.on('value', function(snapshot){
126      var consoleval=snapshot.val();
127      console.log(consoleval);
128
129      tempChange.innerHTML = snapshot.val();
130
131  });
132
133  humidChangeRef.on('value', function(snapshot){
134      var consoleval=snapshot.val();
135      console.log(consoleval);
136
137      humidChange.innerText = snapshot.val();
138
139  });
```

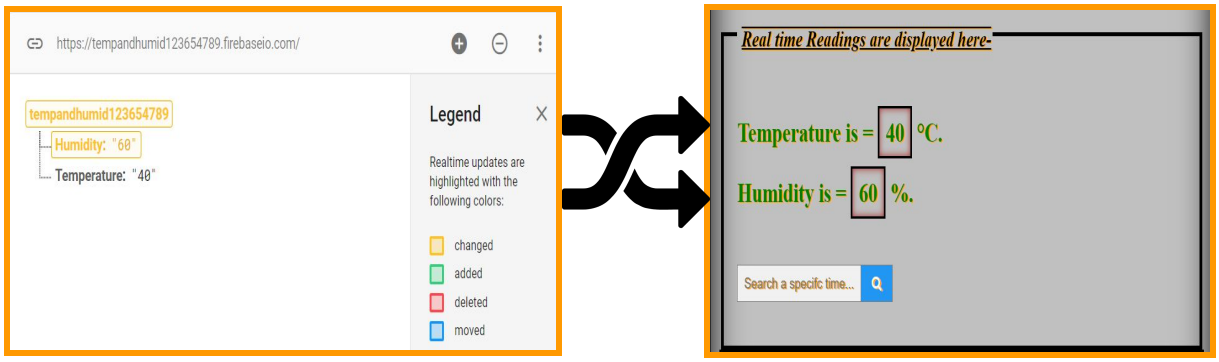
1. The algorithm used to display the real time readings (which gets updated real time in the database by the IoT device) on our web page is that after the web app gets connected to the firebase console using the above method for linking, the web client is programmed to request the database for the **snapshots of the readings** in every certain interval of time (*Line 125 and 133 of the above snippet).* Web clients are able to request a specific data by targeting the specific child element in the database using the firebase libraries (*Line 118 and 119).*

2. Now, to get the **real time updates** of the readings an *On( ) method* is used which is an asynchronous function which goes on listening to the changes without having the necessity to finish the execution of this function to move onto the next one (*Line 125 and 133).* Thus the On method after listening to the changes in the

database extracts the actual reading values using the *Val( )* function (*Line 126 and 134*).

3. JavaScript has been used to target an existing HTML paragraph element using a reference ID (*Line 121 and 122)* which later gets **appended using *innerHTML*** as soon as our web client is able to get the extracted value from the snapshot of the desired readings from the database (*Line 129 and 137).*

❑ DEMONSTRATION OF THE REAL TIME UPDATES.



Firebase Console *Real time display panel*

The above shown pictures are the screenshots of the *Firebase console* and the *Real time display panel of our website.* When the DHT11 sensor connected to the IoT device placed at a remote location **updates the real time readings** in the firebase database, the updated values get displayed immediately in our website. These updates are very spontaneous.

❑ Programming WebPage Using JavaScript And Firebase Libraries To Store The Readings.

```
1   var count=0;
2
3   function changetostore(){
4
5       var divmain=document.getElementById("real1");
6
7       var date=new Date();
8
9       var div=document.getElementById("real1");
10      div.appendChild(main);
11
12      var attributemain=document.createAttribute("id");
13      attributemain.value="mainstorebox";
14      main.setAttributeNode(attributemain);
15
16      var datehead= document.createElement("h1");
17      leg.appendChild(datehead);
18      datehead.innerHTML=date;
19
20      var element= document.createElement("p");
21      main.appendChild(element);
22      var text=document.createTextNode("Temperature is = ");
23      element.appendChild(text);
24      var element1=document.createElement("span");
25      element.appendChild(element1);
26      var text1=document.createTextNode("Value here");
27      element1.appendChild(text1);
28      element.appendChild(document.createTextNode("°C."));
29      ///////Same steps for storing Humidity readings//////
30
```

1. The snippet of code shown here is based on the logic that to store a snapshot of the readings synchronized with the time at that moment, **an infinite loop is created**. This loop uses JavaScript functions for **creating new elements** (*Line 20 and 24)* inside a targeted section of the HTML code of our web page (*Line 5).* By the creation of a new text node inside those new elements make them ready to display anything on the web page. The new attributes are also created inside these new elements using the *createAttributes* function to style these elements (*Line 12-14).*

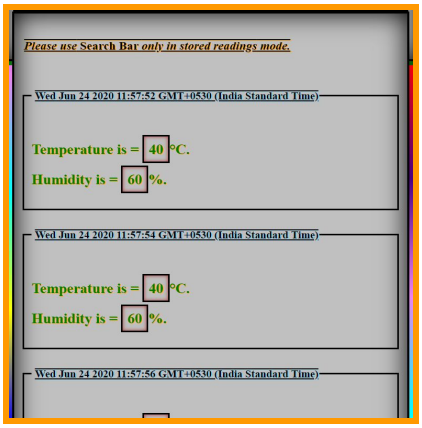
2. Now, to store the readings in the corresponding new element created for a loop, *Date( ) object* has been used to **synchronize the readings along with the time** of the capture (*Line 7 and 18).* This will help users to monitor the readings corresponding to a specific time as well. For the ease of the user, a search bar is also provided on the web site for monitoring specific time readings.

3.

```
39  ////////Firebase/////////
40      var database=firebase.database();
41      var tempChangeRef= firebase.database().ref('Temperature/');
42      var humidChangeRef= firebase.database().ref('Humidity/');
43      var tempChange=document.getElementsByClassName("newnode1");
44      var humidChange=document.getElementsByClassName("newnode2");
45
46  tempChangeRef.once('value', function(snapshot){
47      var consoleval=snapshot.val();
48      console.log(consoleval);
49
50      tempChange[count].innerHTML=snapshot.val();
51
52  });
53
54  count++;
55  }
```

To extract values from the requested snapshot of the desired readings by the web client, *Val( ) function* has been used (*Line 47*) and using *innerHTML* a tagged new element is appended with the corresponding values. ***Once( ) method*** has been used which is a synchronous function which gets triggered once, stores the corresponding time synchronized readings and then stops itself such that it gets triggered again only in the next loop inside a new paragraph element to store the next readings (after 2 seconds).

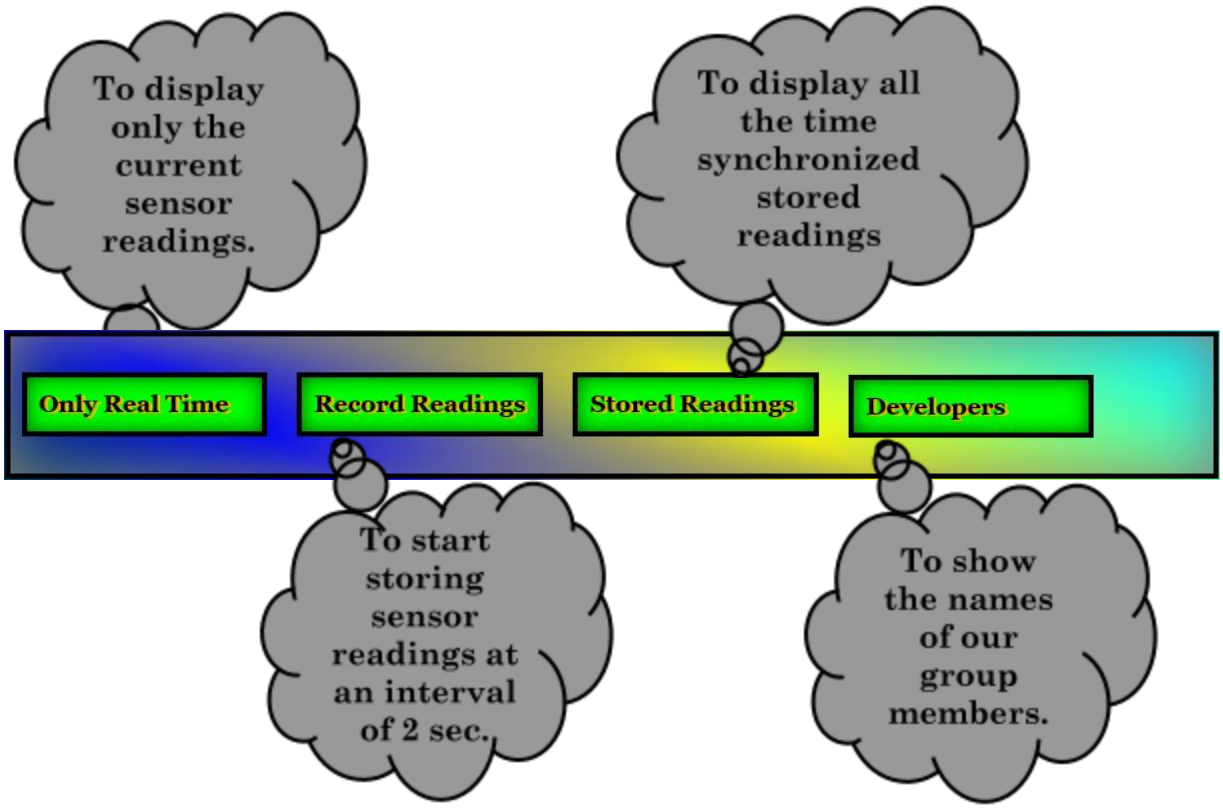
❑ Demonstration Of The Stored Readings.



The shown is the ***stored readings panel*** *of our website* which displays all the readings corresponding to the time and date of their capture. The panel can even be

run in the background and accessed anytime using the Stored readings button on our website which can let the readings store for years and years.

❑ Buttons Functionalities Of Our WebSite.



The complete website can be accessed using this link.

Android Application Development.

Introduction To Android Studio IDE.

1) Android

It is an open source mobile operating system primarily designed for touchscreen devices such as smartphones and tablets. It is the modified version of Linux based kernel and other open source softwares. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. Android 10 is the latest release of the OS which was released in September 2019.

2) Android Studio

It is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Features Of The Android Studio

a. Gradle-based build support.

b. Android-specific refactoring and quick fixes.

c. Lint tools to catch performance, usability, version compatibility and other problems.

d. ProGuard integration and app-signing capabilities.

e. Template-based wizards to create common Android designs and components.

f. A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.

g. Support for building Android Wear apps.

h. Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.

i. Android Virtual Device (Emulator) to run and debug apps in the Android studio.

3) Java Or Kotlin

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features. While IntelliJ states that Android Studio is built on supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

❑ JAVA FOR ANDROID APP DEVELOPMENT

1. Introduction- Java is an object oriented programming language developed by Oracle. The language is statically-typed and provides a good number of features for the development. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them.

2. Statically-typed – Programming at its core is really about working with data. Pieces of data are stored as variables, which are basically containers that hold data. Statically-typed languages like Java require us to declare what type of data each variable (or container) will hold. So for example, if a variable is supposed to hold a number, we need to say so, and we won't be allowed to put something else like a letter in it. Statically-typed also means that all the variables will be checked before the program even runs, and we'll be presented with an error if we forget to declare a type of data or declare the wrong one.

3. Object-oriented – An object-oriented language is one that is built around the concept of objects. In the physical world, take a look around the room and think of each thing as an object. We can then create and manipulate all sorts of objects to do different things in our app. The objects can be instantiated from the respective classes and can have specific methods and attributes.

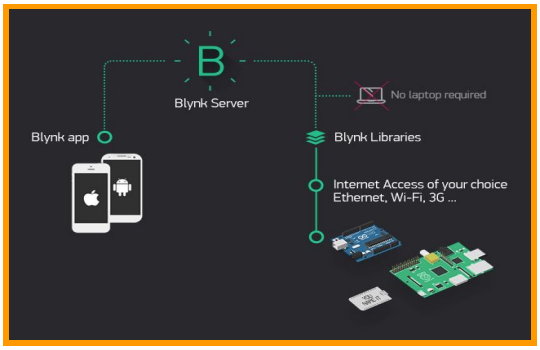
❑ KOTLIN

1. **Introduction -** Kotlin is another programming language supported by the Android Studio IDE for development purposes. s a cross-platform, statically typed, general-purpose programming language with type inference. Kotlin is designed to interoperate fully with Java, and the JVM version of its standard library depends on the Java Class Library, but type inference allows its syntax to be more concise.

2. **Procedural programming -** In addition to the classes and methods (called member functions in Kotlin) of object-oriented programming, Kotlin also supports procedural programming with the use of functions. Kotlin functions (and constructors) support default arguments, variable-length argument lists, named arguments and overloading by unique signatures. Class member functions are virtual, i.e. dispatched based on the runtime type of the object they are called on.

❑ BLYNK APP TO DISPLAY THE RESULTS

1. **Blynk** is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets. Before going to develop our own app we implemented the monitoring functionality with Blynk app. The app provides virtual functionality to simulate the hardware which includes various microcontrollers including Arduino Uno, DHT11 sensor and a wifi module. The app provides a good starting point for the IoT based project with basic functionalities.



2. The app provides a functionality of virtual pins - to provide exchange of any data between hardware and Blynk mobile app. Virtual pins are different from Digital and Analog Input/Output (I/O) pins. They are physical pins on your microcontroller board where you connect sensors and actuators. Blynk lets you control any hardware connected to Digital and Analog pins without having to write any additional code.

3. The app also provides their own libraries for the project which can be used in the Arduino IDE for the interfacing. These libraries are available open source on the Github repository of the app.

4. Although the app provides a good functionality for the monitoring it has a limitation that it can be used only on a local area network (LAN). This limitation is overcome in the app developed by our team. The developed app can be used over the internet.

Android Application to Display Results

We have made an android app to display temperature(in Celsius) and humidity(in percentage) readings stored in the database that are updated from a DHT11 sensor. Our app was made using java programming in android studio.

❑ Creating An Android Application

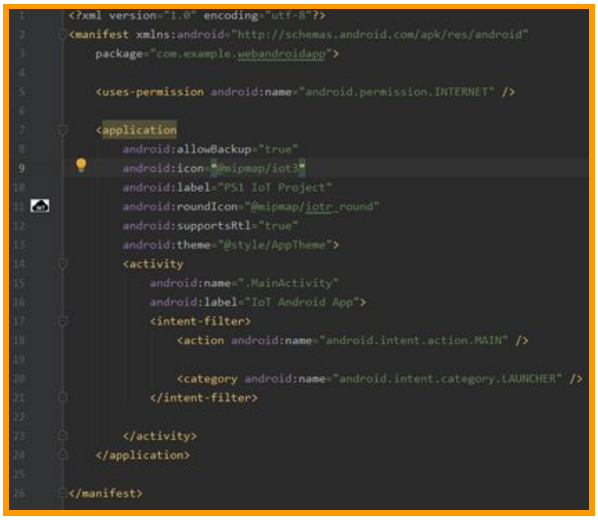
1. Our android application is based on the same idea as that of our webpage. It is because we have loaded data from a webpage using its URL. Data is not extracted directly from our firebase server. That is we are just displaying data from webpage to android app . Instead of going to all that effort of retrieving, processing and then formatting this data, it's easier to simply embed the web page itself in your android application's layout.

2. The Android platform supports a wide range of content including content pulled straight from external web pages. We have embedded web content in our Android applications, using WebViews. The WebView component acts like any other Android View, so we can embed it anywhere in your app's layout from Android Studio's palette.

3. The code snippet given below is the main code for enabling our android app to convert its display into a webview. Line 24 and 25 says that the android app is now a client which connects with our website using its URL and thus loads this URL onto its screen.

```
1  package com.example.webandroidapp;
2
3  import ...
12
13 public class MainActivity extends AppCompatActivity {
14     private WebView myWebView;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20
21         myWebView = (WebView)findViewById(R.id.web);
22         WebSettings webSettings = myWebView.getSettings();
23         webSettings.setJavaScriptEnabled(true);
24         myWebView.loadUrl("https://tempandhumid123654789.web.app/");
25         myWebView.setWebViewClient(new WebViewClient());
26
```

4. Our App uses the internet connection of that android device and establishes a connection to the web page and displays the result. Line 5 of the code snippet given below permits our android app to connect to our mobile's internet. Application part of the snippets is to style the icon, label name of our android app. In short this part specifies the android apps view over the GUI of your mobile.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.webandroidapp">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/iot3"
        android:label="PS1 IoT Project"
        android:roundIcon="@mipmap/iotr_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="IoT Android App">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

        </activity>
    </application>

</manifest>
```

❑ UI Of Our Android Application

- Styles and themes on Android separate the details of your app design from the UI structure and behavior. It is similar to stylesheets in web design. To create a style or theme, first we have to open project's res/values/styles.xml file and follow the following steps:

1. Add a <style> element with a name that uniquely identifies the style.

2. Add an <item> element for each style attribute we want to define.

The name in each item specifies an attribute that would be otherwise used as an XML attribute in our layout. The value in the <item> element is the value for that attribute.

```xml
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>
```

This part of code styles the view of our android app by defining the app theme and colour ascents.

- Hence some of the features of our android application looks:

Our android app has four functionalities which can be performed by pressing these four buttons:

1) Only real time button- By pressing this button we can display continuous real time readings with some constant time delay of 2 sec.

2) Record readings button – This button is used to store real time readings.

3) Stored readings button – This button is used to display all the stored readings.

4) Developers button – This button shows names of our team members.

❑ How Our Android Application Works Step-By-Step

1. The user accesses a web page via our android application, triggering a request to the web server over the Internet on the client's android device.

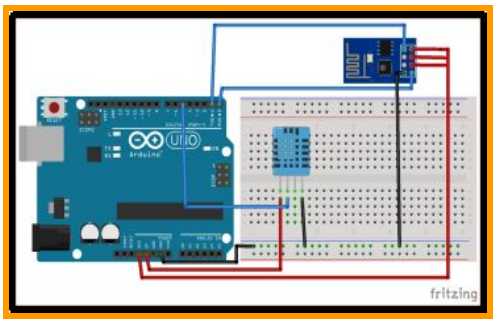
2. The web server then performs the requested task and generates the results of the requested data.

3. The web server delivers the requested information to the client and the information appears on the user's display.

Conclusion

This project aims at monitoring IoT based server room temperature and humidity for the Bharat Heavy Electricals Limited, Visakhapatnam. This will help them in not only monitoring their server's room temperature and humidity but also to prevent any casualty from occurring.

❑ Sending Data To The Database

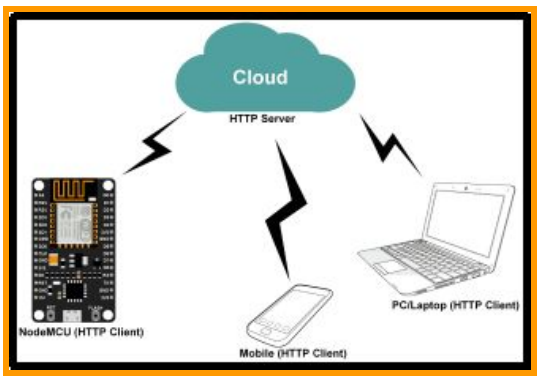


❑ We started off by setting up the apparatus which includes connection of an Arduino Uno board with a DHT11 sensor and a NodeMCU module. After the connection has been made the DHT11 sensor will send the data (room temperature and humidity readings) to the cloud via the micro-controller and the wifi-module. This is for the hardware part of the project and the data has been reached to the cloud.

❑ Data is sent in the form of Serial Data from the Arduino Uno to NodeMCU. NodeMCU sends this data to Google Firebase Database.

❑ The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client.

❑ Retrieving Data From The Database



❑ Second step is to retrieve the data from the cloud or in this case Google Firebase Database.

❑ Connected web client requests Firebase Database for the snapshot of the readings.

❑ Firebase Libraries are used to target the specific child element in the database.

❑ Developing A web Page

❑ We have developed a web-page using HTML, CSS and JavaScript.

❑ HTML and CSS are used to design and style the page and do the frontend part which includes the client-side development.

❑ JavaScript is used to control the behavior of different elements.

❑ Displaying The Sensor Readings

❑ The Main role in the **Web Page** for displaying the data is of JavaScript which appends the HTML elements upon receiving the snapshot of the readings from the Firebase Database.

❑ Developing An Android Application

❑ The Mobile Application plays the role of a client and requests data from the web-page which in turn requests data from the Firebase Database.

❑ Android Studio links the UI of the Application with the Firebase Database.

- Since this application uses internet connectivity to extract data from the web-page using its URL and so it can be functioned from anywhere unlike Blynk.

- Blynk is an already existing app for the same project but it only works for a LAN (Local Area Network).

Therefore, this project will greatly help in reducing the possibility of any casualty or damage to company caused by heating up of the servers as a buzzer will go off as soon as the temperature of the servers exceeds a specified limit and this can be achieved from anywhere as it uses internet connectivity.

RECOMMENDATIONS

Nothing can be made perfect in this world and this applies to our project as well. So there are some recommendations to make this project even more effective and better. Below is the list of some recommendations to make:

- To make the functioning of our project fully automatic, implementation of an AI bot can be done whose work is to decide what to do with the servers at all ranges of our readings of the Room Temperature and Humidity.
- We are using a separate wifi module instead we could use an IoT board which contains the wifi module itself, that will optimize the space consumption by the hardware.
- For the security reasons as it uses company servers, we can also build a login page using authorization from the Google Firebase. We need to maintain a database of the company personnel so that only those who are registered can login to the site.

REFERENCES

★ **Database**

https://en.wikipedia.org/wiki/Database

★ **Types of databases**

https://www.geeksforgeeks.org/difference-between-sql-and-nosql/

★ **Linking the web app with the firebase**

https://nordicapis.com/what-is-the-difference-between-an-api-and-an-sdk/

★ **The link to our website**

https://tempandhumid123654789.web.app/

★ **About the web development**

★ **web application**

https://www.riverbed.com/in/faq/how-does-web-application-work.html

★ **Frontend and Backend**

https://www.geeksforgeeks.org/frontend-vs-backend/

Glossary

- ★ **BHEL** - Bharat Heavy Electricals Limited
- ★ **HTML** - Hypertext Mark-up Language
- ★ **CSS** - Cascading Style Sheets
- ★ **IOT** - Internet of Things
- ★ **LAN** - Local Area Network
- ★ **UI** - User Interface
- ★ **XML** - eXtensible Markup Language
- ★ **Sketch** - An Arduino program