

A REPORT OF SIX MONTHS INDUSTRIAL TRAINING

At

SIMPLUS LABS

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR  
THE AWARD OF THE DEGREE OF

**BACHELOR OF TECHNOLOGY**

(Electronics and Communication Engineering)



Jan-June 2021

**SUBMITTED BY:**

MUNISH GOYAL

11702024

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PUNJABI UNIVERSITY PATIALA

## CERTIFICATE



## **DECLARATION**

I hereby certify that the work, which is being presented in the project report, is by **Munish Goyal** for fulfilment of the requirement for the award of the degree of **Bachelor of Technology (Electronics and Communication Engineering)** in University College of engineering, Punjabi university, Patiala. It is an authentic of my own work.

Munish Goyal

11702024

## **ABSTRACT**

The Internet of things (IoT) is the inter-networking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items—embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data. IoT allows objects to be controlled and gather information remotely across the already established network.

Salesforce is a cloud-based customer relationship management (CRM) software that accelerates business relationships and can transform the working lives of the team. Marc Benioff developed it in the late 1990s and now it has been announced as the world's most innovative company for six consecutive years by Forbes Magazine. Unlike traditional CRM software, Salesforce is an internet service. It is available with just a sign-up and logs in through a browser, and it is immediately available. It is based on cloud computing, where the customers, without the need of installing any traditional software, can access the cloud, i.e., through the internet, for their business needs.

The goal of this project is to create an IoT Temperature Monitoring System that can monitor specific criteria, as well as control specific devices. Data to be monitored are: temperature and humidity. All data is stored in an organized manner and user can generate many different analytics reports which can be further used in business requirements.

The user will be notified with an email if temperature is above critical limit. Also all data analytics reports can be individually subscribed by the user. The device must also work while power is out in the house; which means a battery back-up and Internet hot spot must be included as well.

## **ACKNOWLEDGEMENT**

It gives me a great sense of pleasure to present the report of the B.Tech Final Year Project. I owe special debt of gratitude to the department of Electronics and Communication Engineering, Punjabi University Patiala, especially to our Head of Department **Dr. Manjit Singh Bhamrah** who gave me this opportunity to work on this project. Their sincerity, thoroughness and preservice have been a constant source of inspiration for me. It is only their cognizant efforts that my endeavours have seen light of the day.

I would like to express my gratitude towards my parents & member of my industry for their kind co-operation and encouragement which help me in completion of this project. I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciations also go to my friends in developing the project and people who have willingly helped me out with their abilities.

# **TABLE OF CONTENTS**

1. Introduction
  - 1.1. Project Details
  - 1.2. Module Description
  - 1.3. Company Profile
  - 1.4. Objective
2. Cloud Computing
  - 2.1. Introduction to Cloud Computing
  - 2.2. Service Models of Cloud Computing
    - 2.2.1. Software as a Service (SaaS)
    - 2.2.2. Platform as a Service (PaaS)
    - 2.2.3. Infrastructure as a Service (IaaS)
3. Salesforce
  - 3.1. Introduction to Salesforce
  - 3.2. The Architecture of Salesforce
  - 3.3. Why Salesforce
  - 3.4. Salesforce Multiple View Controller (MVC)
  - 3.5. Different editions of Salesforce
  - 3.6. System Overview
  - 3.7. Technologies of Salesforce
  - 3.8. Benefits of Using Salesforce
4. Project Work
  - 4.1. Monitor Temperature and Analysis using Salesforce
    - 4.1.1. Overview
    - 4.1.2. What is a Temperature Monitoring System?
    - 4.1.3. Benefits of IoT temperature monitoring
  - 4.2. Development Area
    - 4.2.1. Custom Apps
    - 4.2.2. Objects
    - 4.2.3. Fields
    - 4.2.4. Relationships
    - 4.2.5. Tabs
    - 4.2.6. Apex
    - 4.2.7. Reports and Dashboards
    - 4.2.8. Process Builder
    - 4.2.9. Salesforce Sites
  - 4.3. System Hardware
    - 4.3.1. ESP8266
    - 4.3.2. Temperature Sensor
  - 4.4. Application Program Interface (API)
    - 4.4.1. Working of API
    - 4.4.2. RESTful APIs
    - 4.4.3. REST Architecture

- 4.4.4. HTTP Request
    - 4.4.5. HTTP Response
    - 4.4.6. Webhooks
- 5. The Functionality of Monitor Temperature and Analysis
  - 5.1. Sending Data to Salesforce Cloud (IoT Platform)
    - 5.1.1. Sending Data using NodeMCU
  - 5.2. Monitoring and Analysis of Data (Salesforce Cloud Platform)
    - 5.2.1. Salesforce Debug Logs
    - 5.2.2. Data Processing
    - 5.2.3. Salesforce Email Alerts
  - 5.3. User Functionality
  - 5.4. Salesforce Mobile App View
- 6. Conclusion and Future Scope
- 7. Appendix
- 8. References

## List of Figures

Figure 1: Cloud Computing

Figure 2: Service Models of Cloud Computing

Figure 3: Service Models

Figure 4: Multi Tenacy

Figure 5: Salesforce MVC

Figure 6: Different Versions of Salesforce

Figure 7: Benefits of Salesforce

Figure 8: CRM

Figure 9: Monitor Temperature and Analysis

Figure 10: Salesforce Home Page

Figure 11: Lookup relationship

Figure 11: Monitor Temperature and Analysis App Home Page

Figure 12: NodeMCU-ESP8266

Figure 13: DHT11 Temperature Sensor

Figure 14: Working of API

Figure 15: Restful API

Figure 16: REST API Architecture

Figure 17: HTTP Request and Response

Figure 18: Webhook

Figure 19: Project Functionality

Figure 20: Circuit Diagram

Figure 21: Finding Wi-Fi Connection

Figure 22: Successfully Connected to Wi-Fi

Figure 23: NodeMCU Response after hitting API

Figure 24: Salesforce Functionality

Figure 25: Salesforce Debug Log Section

Figure 26: A Debug Log

Figure 27: Wrapper class used for processing data

Figure 28: Record Inserted in Salesforce

Figure 29: Salesforce Email Alert Section

Figure 30: Email Alert Template

Figure 31: Functionality for User

Figure 32: Login Page of Salesforce

Figure 33: Home Page of Temperature Monitor and Analysis

Figure 34: Temperature Data Tab View

Figure 35: Reports Tab View

Figure 36: Email View of Dashboard

Figure 37: Salesforce Mobile View

Figure 38: Future up gradation



# Chapter 1: Introduction

## 1.1 Project Details

- Title: Monitor Temperature and Analysis using Salesforce
- Platform: Internet of Things (IoT), Salesforce (Software as a Service)

In this IoT project we are going to monitor humidity and temperature over the internet using Salesforce where we will send the current Humidity & Temperature data to the server and then the analytics to the owner through Email using the Salesforce server.

System hardware consists of:

1. DTH11 Temperature Sensor
2. ESP8622 Wi-Fi Module Device

### Key Features:

1. User friendly
2. Efficient performance
3. Great response
4. Automatic Reporting
5. Data stored in cloud

## 1.2 Module Description

- Planned approach towards working: The working in the organization will be well planned and organized.
- Accuracy: - The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information incoming from the center is accurate.
- Reliability: - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.
- No Redundancy: - In the proposed system almost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.
- Immediate retrieval of information: - The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires.
- Immediate storage of information: - In manual system there are many problems to store the largest amount of information.
- Easy to Operate: - The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

### 1.3 Company Profile

This project is being developed under the guidance of **SimplusLabs Company** Mohali. SimplusLabs is passionate about the value that we deliver to their clients. We pride ourselves on understanding the needs of our clients from the inside out. Together as partners, we will succeed with SimplusLabs, and we will bring success to our clients.

This was a ground-floor opportunity with a start-up built from experienced, professional developers and executives. Get in early and ride the wave to the top of the next generation of Salesforce.com and web development experts.

Presently, SimplusLabs is already leading the race with its competitors. Being nurtured by a team of experienced and sensitive people. They try to bond emotionally with their clients and love to go an extra mile to satisfy their needs, which is the reason that they hold the edge in the league.

SimplusLabs is a leading IT company in North India and is now going global with its centers in Canada and US. And SimplusLabs has launched its case study on application form project for us based data science boot camp training provider.

From concept building to implementation of any project, their team manages the projects efficiently up to its completion. Their tactful strategy and dedication towards quality work has given us the recognition we enjoy and that is why our clients only come back to us whenever they require any kind of web related solutions. They do not only emphasize on formulating an attractive solution to their clients, but we believe in providing a workable solution. They Aim To be a dynamic, vibrant and value-based global IT service providers centered around customer, employee and societal goals. To be a class leading and innovative IT Services Company.

### 1.4 Objective

The need for sophisticated and robust temperature monitoring systems is increasing, especially for businesses and organizations within the healthcare, food products, and electronics sectors.

Such organizations utilize temperature monitoring technologies to monitor the temperatures of their products and processes; this is especially important to safeguard their products and meet regulatory standards within the region.

Temperature monitoring can be also part of preventative reliability. This is important when a system is not performing high temperature processes, yet can be at the risk of overheating.

## Chapter 2: Cloud Computing

### 2.1 Introduction to Cloud Computing

Cloud computing is usually defined as storing and managing the data over the cloud, rather than a local server. Cloud computing is easy to understand. All applications are developed and run in the web browser. Using the internet connection, users and developers will have access to whole applications thus eliminating the complexity and overhead of the maintain environment [4].

Unlike traditional business applications which are complicated, expensive and need experts to install, run, update and secure, cloud computing can be accessed anywhere with an online connection. In traditional systems, the entire infrastructure must work together. For such type of seamless interaction, and for the smooth run of the system, a constant maintenance is always required. With cloud computing, there is no necessity to invest money in acquiring and supporting hardware and software infrastructure, thus decreasing the potential cost for users and developers.

The main impact of cloud computing is on the responsiveness of IT systems. With the cloud computing environment, we can add users and developers instantly, and the applications can be deployed rapidly into the cloud which reduces the user request response time. As the complexity of the internal systems is removed, the organization can speed up the entire IT process.

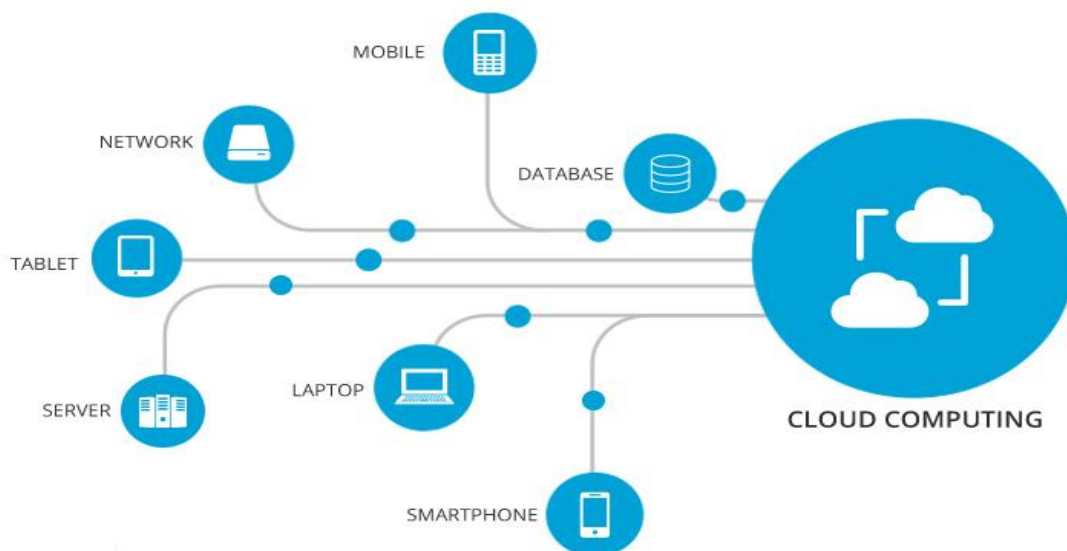


Figure 1: Cloud Computing

## **2.2 Service Models of Cloud Computing**

### **2.2.1 Software as a Service (SaaS)**

The end user can access the application which is developed by the provider on a cloud framework. The developed applications are available from different customer devices through interfaces like a web program or a program interface. Cloud infrastructure, servers, networks, storage and operating systems cannot be managed or controlled by the customer.

Cloud application services represent the increasing cloud market. Software as a service utilizes the internet to deploy the applications overseen by the vendor and whose interface will be able to access on the customer side. Many of the applications developed using SaaS will run in a web browser by using some plugins. There will not be need of any download or establishments.

The major applications which are developed using SaaS are healthcare related applications, client relationship administrations, incorporate email, and collaboration. Some of the costly ventures which are not able to considered as software vendors started using SaaS to get the upper hand and gain income.

### **2.2.2 Platform as a Service (PaaS)**

The Client can deploy onto the cloud infrastructure developed by the customers with the help of libraries, tools, services and the programming languages which are supported by the client. The underlying infrastructure of cloud and storage, servers, network or operating systems cannot be managed or controlled by the Customer.

Cloud platform services can be utilized for applications and their advancements when cloud segments are given to programming. Developers can be able to pick the structure using PaaS where the applications can be expanded to create or modify. The testing and deployment of applications become easy and fast if the PaaS is used.

Enterprise PaaS gives a self-service portal to programming engineers for overseeing computing infrastructure from information technology operations. Scalability, Software as a service enablement and multi-occupancy can be acquired by the applications using PaaS. The coding fundamental measure will be decreased for the enterprises using PaaS and the application will be converted to a hybrid model.

### **2.2.3 Infrastructure as a Service (IaaS)**

The customer can arrange systems, storage, processing and other essential computing resources in which the client can run and send arbitrary software like applications and operating systems. The hidden cloud infrastructure is not controlled or overseen by the customer but rather he can control over operating systems, storage, servers and network [14].

Cloud infrastructure services, referred to as IaaS, and are models beneficial for overseeing and observing remote data centre frameworks such as organizing, processing, stockpiling and networking services. With the help of IaaS, clients will get utility billing and power benefits.

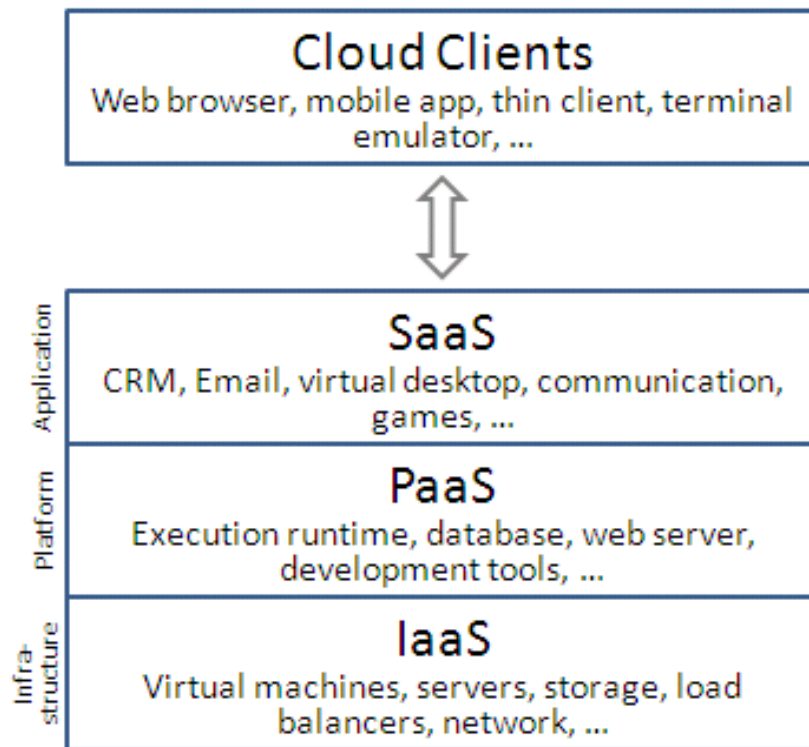


Figure 2: Service Models of Cloud Computing

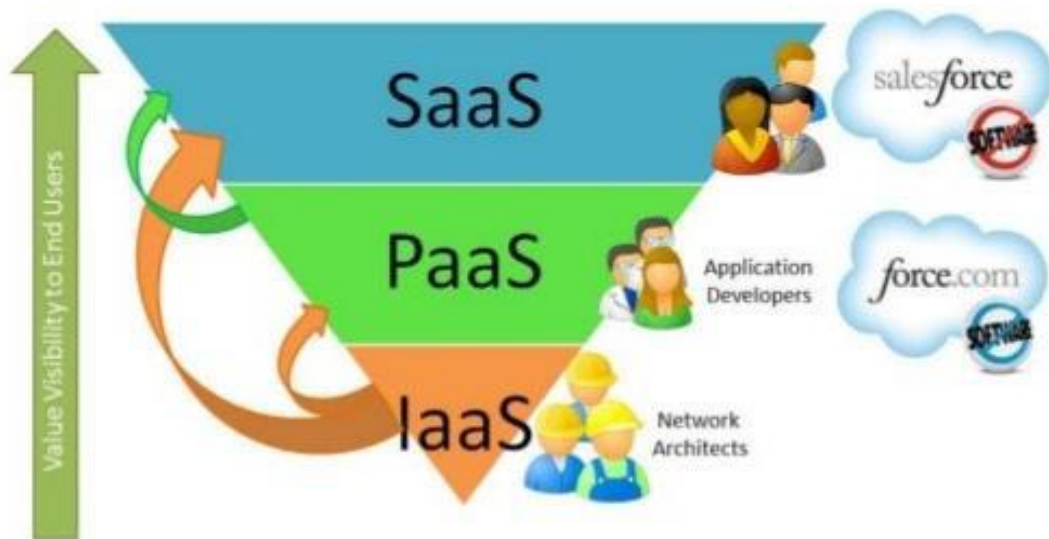


Figure 3: Service Models

## Chapter 3: Salesforce

### 3.1 Introduction to Salesforce

Salesforce is one of the world's prime cloud computing companies and number one on-demand customer relationship management (CRM). Salesforce does not need any software installation or hardware or any infrastructure like servers. All we need to access Salesforce is the internet. This empowers even the most non-techie individuals to be able to use the system and configure it as per their needs.

Established as Salesforce.com (SFDC) and its customer relationship management (CRM) service and then divided into different sectors like sales cloud, service cloud, community cloud, analytics cloud, data cloud, marketing cloud, app cloud, and so on.

Since Salesforce coordinates well with all the platforms and supports all major OS and mobile devices, it is anything but difficult to utilize Salesforce outside of the workplace, thus helps to improve productivity.

### 3.2 The Architecture of Salesforce

Salesforce has a multi-tenant Architecture. Multi-tenancy is the fundamental technology utilized as a part of the cloud to share its resources safely and cost effectively. It's much the same as bank services where various tenants cost-efficiently share a common infrastructure yet safely and with most protection from other tenants. A cloud utilizes multi-tenant infrastructure to share its assets safely among different applications and occupants (organizations, associations, and so on) that use the cloud. Some clouds utilize virtualization-based architecture to confine occupants; others utilize custom software architecture to take care of business. The multi-tenant outline of a cloud service can dramatically affect the application delivery and the profitability of IT organization.

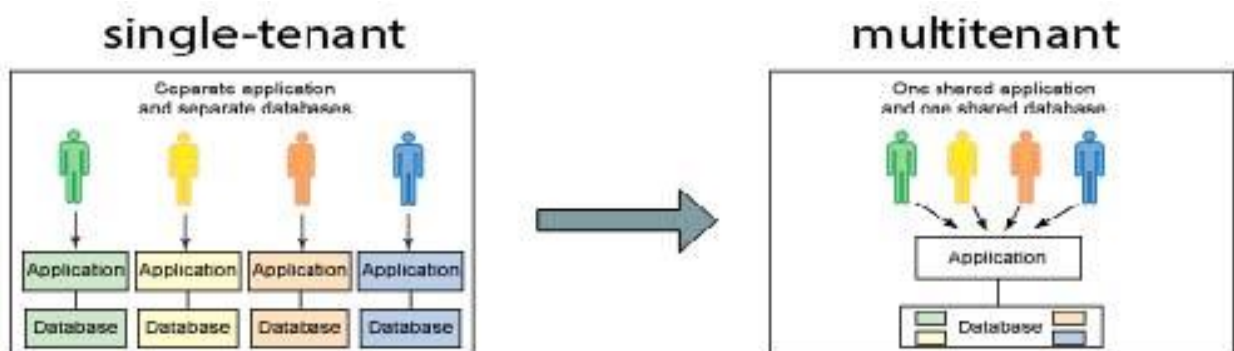


Figure 4: Multi-Tenancy

### 3.3 Why Salesforce

The first benefit of choosing Salesforce is that we do not need to install any software to build an application or run our business activities. We can focus on building the apps instead of installing, setup, repairing, etc.

Since it is the cloud-based CRM platform, any user can access it from anywhere using the internet. Any new user or business start-up can choose Salesforce for building apps or run business, as it is so affordable with lots of capabilities.

One can easily integrate the Salesforce with any third-party applications, such as Gmail, which is much easier than other CRM Software. The development cost is also very less.

### 3.4 Salesforce Multiple View Controller (MVC)

MVC is a design pattern which separates business logic from interface logic i.e. it separates the graphical interface displayed to the user with the code that manages the user actions [4].

In Salesforce, using SFDC visual force, we can write VIEW pages which are very similar to java servlets page (JSP) pages. Each visualforce (VF) page is correlated with a controller. The controller and model classes can be written using Apex language. In SFDC, controller part comprises of workflows, triggers, Apex classes and model layer comprises of fields, relationships, objects and View layer comprises of Tabs, page layouts, VF pages.

SFDC MVC mainly consists of three modules namely Model, View and Controller.

- 1) Model: Here we represent what schema and data the Salesforce used for the system representation and Subjects are a model, as every entity is mapped to some subject in Salesforce.
- 2) View: Here we represent how data and schema and visual force are used to present data to users.
- 3) Controller: Here we use controllers and interface actions to perform actions when the user interacts with visual force

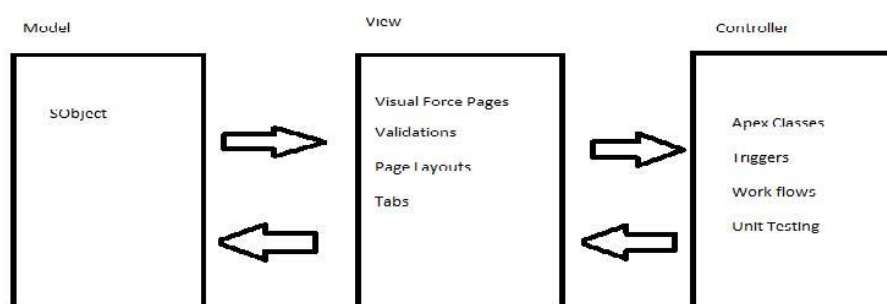


Figure 5: Salesforce MVC

### 3.5 Different editions of Salesforce

Salesforce offers different editions of its products and services depending on business needs. 1) Professional Edition: This edition offers full CRM functionality.

2) Enterprise Edition: This edition offers advanced CRM customization and administration tools along with functionalities of Professional edition.

3) Unlimited Edition: Along with functionalities offered by Enterprise Edition, Unlimited Edition even offers full mobile access, premier support, unlimited custom apps and more.

4) Developer Edition: This edition allows developers to extend the Salesforce system, integration with other applications and develop new applications and tools.

SalesforceIQ Starter	Professional	Enterprise	Unlimited
Out-of-the-box CRM for up to 5 users	Complete CRM for any size team	<b>MOST POPULAR</b> Deeply customizable CRM for your business	Unlimited CRM power and support
<b>TRY FOR FREE</b>	<b>TRY FOR FREE</b>	<b>TRY FOR FREE</b>	<b>TRY FOR FREE</b>
Automatic data capture Customizable sales tracking for one list Intelligent follow-up reminders Contact auto-complete Shared address book Sent-email notifications Seamless collaboration Smart mobile apps Chrome extension Webinars and live trainings	Account and contact management Opportunity tracking Lead management Task and event tracking Customizable reports and dashboards Mobile access and administration Chatter – company social network Outlook Side Panel and sync Role permissions Case management Campaigns** Quotes and orders** Collaborative forecasts** Mass email** Sales Data*** Sales Cloud Engage***	Get all Professional features PLUS Workflow automation Enterprise territory management Profiles and page layouts Custom app development Integration via web service API Salesforce Identity Salesforce Private AppExchange Report history tracking** Approval automation** Sales Data*** Sales Cloud Engage***	Get all Enterprise features PLUS Unlimited customizations Unlimited custom apps Multiple sandboxes Additional data storage 24/7 toll-free support Access to 100+ admin services Unlimited online training Sales Data*** Sales Cloud Engage***

Figure 6: Different Versions of Salesforce



### **3.6 System Overview**

Salesforce.com allows administrators to configure and design systems for complex implementations. Salesforce combines the power of configuration and custom development in its platform i.e. Force.com Platform. This platform user can make use of custom code, workflows rules, approval processes to implement their business logic and they can integrate the data with other applications, generate reports and do the analytics within no time. The Salesforce CRM model is used in organizations for interactions like emails, meetings, events with customers and also for prospects like sales, marketing, and support.

With Force.com, we can run business in the mobile using the Salesforce app. We can build and optimize the apps for mobile using HTML 5 and UI framework and it supports all devices with just one code base [5]. Salesforce1 downloadable app can be installed from the App store or Google play on a mobile device.

Salesforce even has an app marketplace called AppExchange where we can find pre-built business applications. AppExchange offers thousands of verified and secured apps built by others or we can develop our own app and sell it here. It's quite similar to the App store and the play store [1].

Due to its ease of access, ease of use, minimum licensing/proprietary issues, and per user cost, Salesforce becomes a power system from small to large scale industries.

### **3.7 Technologies of Salesforce**

#### **1) Apex**

Salesforce has a programming language called Apex. It is a case-insensitive, mostly typed object-oriented programming language with syntax identical Java with curly brackets and dot-notation syntax. Apex is used to run programs and procedure in Force.com such as links, buttons, and record insertion and so on with visual force custom controllers [5].

#### **2) Visualforce**

Visualforce (VF) is a framework for the Force.com platform with tag-based mark-up language identical to HTML. With the help of Visualforce, custom pages can be created for mobile apps and desktops with the help of with other front-end technologies like HTML, CSS, jquery, and JavaScript. With the Visualforce standard and custom controller features, we can build our own business logic in Apex [5].

#### **3) Lightning**

Lightning is a component-based framework for the Salesforce1 mobile app which is built on an open source Aura framework. With the lighting framework, responsive applications can be built easily. The apps build on the Lightning framework is sold or brought on AppExchange [5].




Lightning App builder for Salesforce is a tool for quick application advancement of responsive web interfaces. This interface takes into account distinctive screens to be assembled given lightning segments. This can be layouts as formats for records or particular applications [5].

### **3.8 Benefits of Using Salesforce**

#### **1) Invest in innovation, not infrastructure**

With Salesforce, we can focus on business rather than the back end as Salesforce add new features and automatic upgrades three times a year. These boundary-pushing work made Salesforce one of Forbes' Most Innovative Companies.

#### **2) Don't stress about the data security**

		
<b>Application Services</b>	<b>Network Services</b>	<b>Infrastructure Services</b>
Manage password policies, app and data access, and field-level security.	Control access with built-in HTTPS, authentication, and network security controls.	Run apps securely, with automatic scalability, backup, and recovery.

#### **3) Make Salesforce work the way we want**

Salesforce can be customized to the core to be more agile and productive and the apps can be developed with interfaces with point and click to high-end platforms [6].

#### **4) With AppExchange, find prebuilt applications in minutes**

With industry-specific third party apps, Salesforce can be spread to every division and corporation. These apps are installed by millions of people and are reviewed by thousands. So with trust on the apps, we can focus on extending the business [6].

#### **5) Work on one platform where everything works together**

With Salesforce APIs, core technologies and third party tools, we can connect and manage data from any system and from anywhere [6].

Salesforce works in Desktop, Mobile Devices, and iPad too.

Salesforce even have:

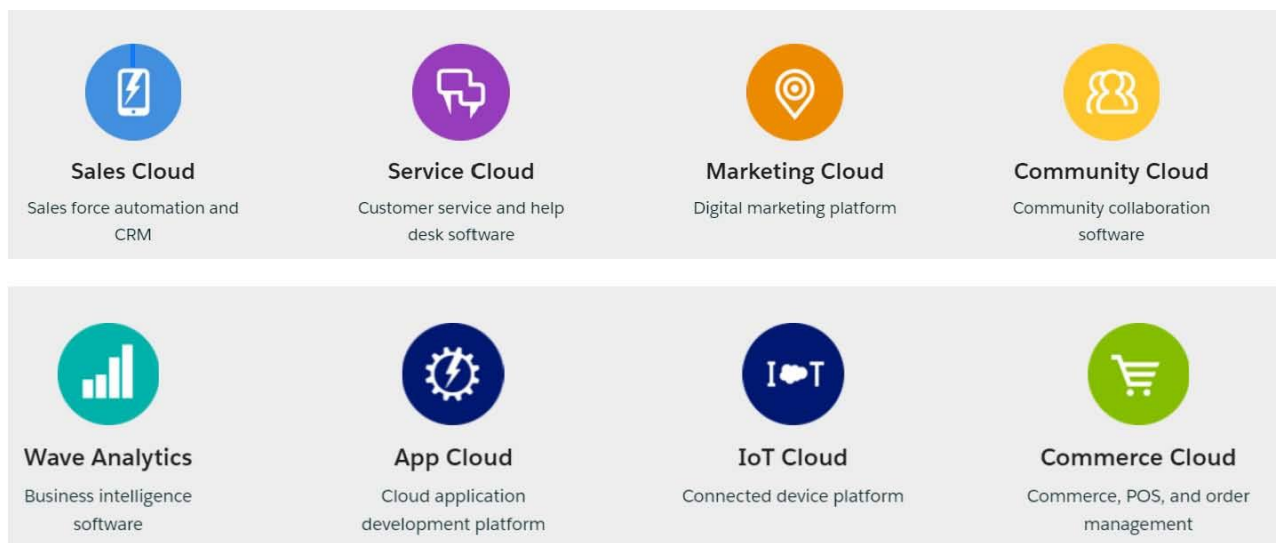


Figure 7: Benefits of Salesforce

### 3.9 Customer Relationship Management (CRM)

CRM is a software to manage all the customer and company's interactions such as scheduling tasks, emailing, texting, and many more.

It contains and manages all the customer-related information such as Customer Name, Address, Phone Number, Email address, and other business-related information. The software keeps all the interactions done with customers, complaints registered by the customer, resolutions provided by the executive, and other customer activities with the particular business or product.

It also allows us to track and manage the sales, marketing, and prospects given to customers.

Before Salesforce, Customer Relationship Management (CRM) solutions were hosted on a company's own server. More cost and time it took for companies to have their own CRM solutions

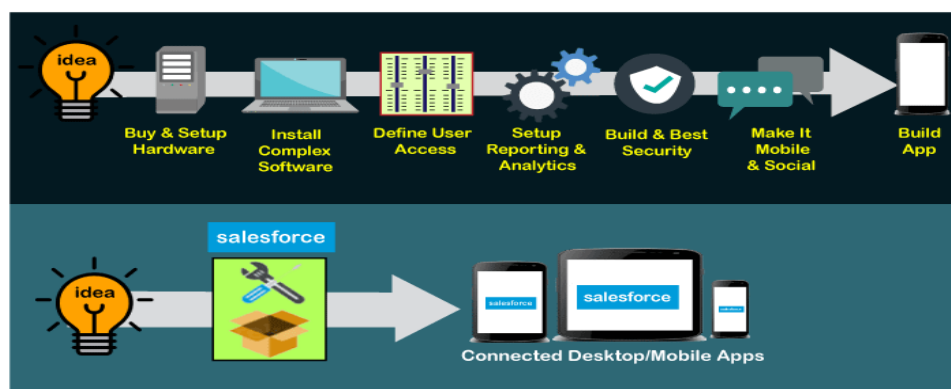


Figure 8: CRM

## Chapter 4: Project Work

### 4.1 Monitor Temperature and Analysis using Salesforce

#### 4.1.1 Overview

The Internet of Things technology has been growing rapidly and has already bypassed conventional systems in terms of features and functionalities. Its applications in the field of remote monitoring and advanced analytics are revolutionizing businesses and are offering exemplary benefits to them.

One such application of IoT is **temperature monitoring**. A temperature monitoring solution is immensely beneficial to industries where temperature plays a crucial role in defining the quality of the products. Moreover, in a cold chain, a temperature monitoring solution allow a logistic manager to monitor the temperature of the item being transported in real-time. Sensors continuously measure the temperature of a closed system and share it with a centralized platform over a network. By using this platform, an individual can remotely monitor the temperature of the system and make sure that the cargo is being shipped within a temp-controlled environment.

The implementation of this solution is really helpful for goods that are climate-sensitive. It is convenient to use and help in confirming the quality of the products during warehousing and shipping processes. So, the concept of temperature monitoring systems revolves around using technology to maintain product freshness and reduce waste by monitoring the temperature of a space.

**Monitor Temperature and Analysis** is an IoT based project. This project is combination of both software and hardware. It is the process of collecting, sending and organizing data in the Salesforce cloud.

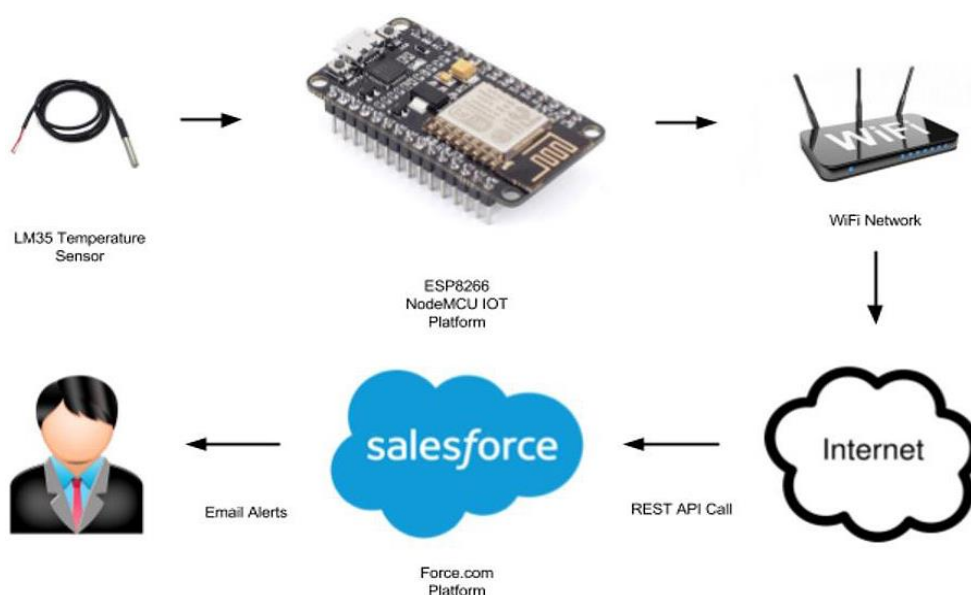


Figure 9: Monitor Temperature and Analysis

### **4.1.2 What is a Temperature Monitoring System?**

A temperature monitoring system controls and regulates the temperature of a particular environment. With a temperature monitoring system, we can easily track, control, and regulate the products' temperature in a specific environment.

A temperature monitoring system makes sure that your temperature-dependent products stay safe as they are being transported.

### **4.1.3 Benefits of IoT temperature monitoring**

The applications of temperature monitoring systems are quite versatile. They offer exemplary benefits in terms of keeping the environmental conditions stable and controlled as per specific requirements. Below are the top 5 benefits that this solution offers:

#### *A. Save time with instant notifications and alerts:*

An IoT system offers instant notifications in real-time. Hence, a temperature monitoring system allows a company to track the environmental parameters on a secure web/mobile-based platform. This eliminates redundant tasks like taking manual readings, thus saving time and elevating quick decision making.

#### *B. Productivity improvement with advanced analytics:*

The data gathered from the temperature sensors can be used to create statistical insights. These insights will include the time duration during which products deteriorate and details associated with the temperature readings. This will help the companies to improve the reliability of their warehouse and cold storage.

#### *C. Maintaining regulatory compliance:*

Companies must maintain the quality of their products in the entire cycle of the supply chain. The products should meet the safety and quality standards set by the authoritative regulatory bodies. Temperature monitoring allows companies to protect their products from the adverse effects of changing weather and thus meet regulatory compliance.

#### *D. Accessibility from remote locations:*

As IoT encompasses advanced telemetry capabilities, remote temperature monitoring from distant locations is also. The data gathered through temperature monitoring sensors can be accessed from faraway places on a web application or a mobile app.

## 4.2 Development Area

As Salesforce is a cloud-based platform, all it needs is a browser and internet connection. Once a developer or user of force.com platform logs in into salesforce.com, he sees his profile icon on the top right corner of the page, and besides that, there is a setup button which we use mostly while developing an app.

- Open a browser and enter <https://login.salesforce.com/>
- Enter Username and Password

It will take us to the Salesforce customized company domain and it will look likes as shown below.

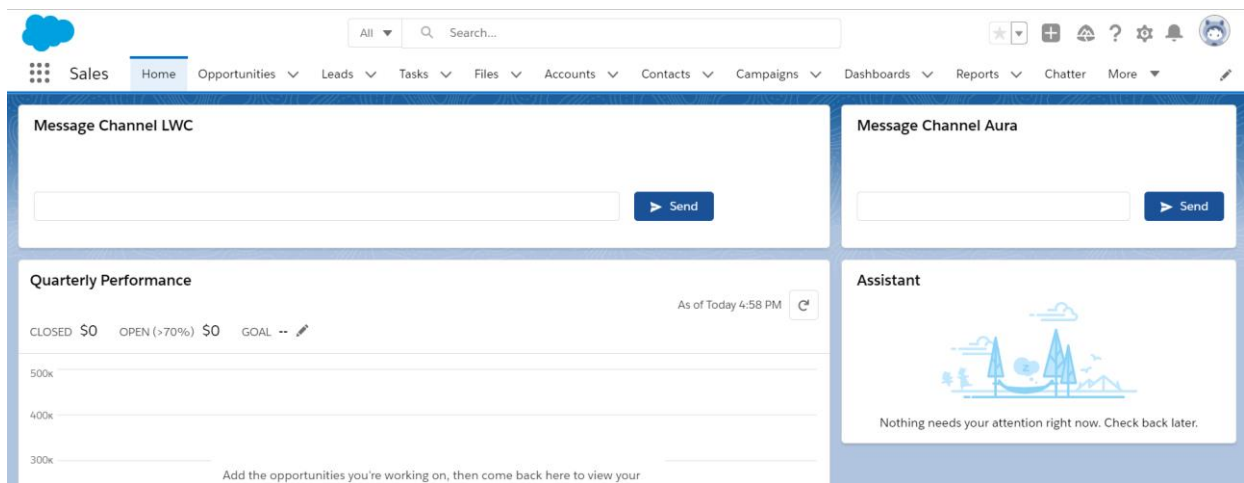


Figure 10: Salesforce Home Page

After the Setup button in the header, we can see the Help button which will give detail information about anything in this platform and on top left we can text “Sales” showing Sales app is opened. Also by clicking on nine dots we get whole list of apps available in this domain. By selecting the Monitor Temperature and Analysis app, we can see all the data/ modules related to this app along with some predefined standard data as shown below.

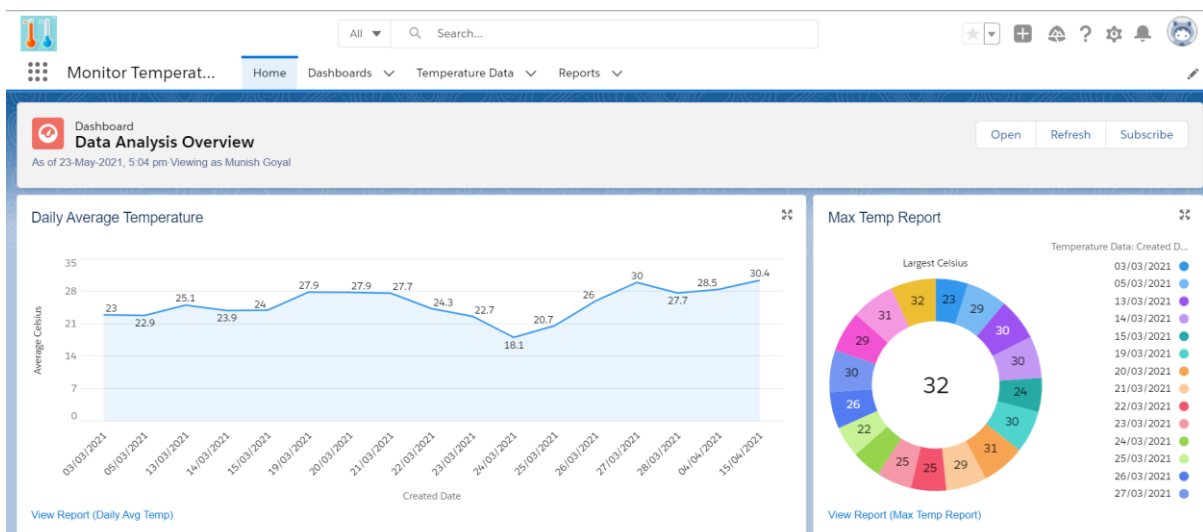


Figure 11: Monitor Temperature and Analysis App Home Page

### 4.2.1 Custom Apps

In Salesforce, creating an App is just a point and click away, to create a new app click on Setup and search for App Manager, then click on New Lightning App.

- Define the new Lightning app with App Name and Developer Name. Click **Next**.
- On the App Options screen, leave the defaults as is and click **Next**.
- On the Utility Items screen, leave the defaults as is and click **Next**.
- On the Navigation Items screen, select **Home, Chatter, Waypoints, Reports,** and **Dashboards** and move them to the Selected Items box. Then click **Next**.
- On the Assign to User Profiles screen, select **System Administrator** and move it to Selected Profiles. Then click **Save & Finish**.

Salesforce provides some predefined apps like Sales, Marketing etc. which consist of standard objects and standard fields. Now, we are creating our custom app which will consist of custom objects and custom fields to hold the data.

### 4.2.2 Objects

Objects in Salesforce are **database tables** with information. The primary object in the Salesforce data model represents accounts/companies and organizations involved in the business, such as customers, partners, and competitors. A record is similar to a row in a database table.

Objects already created by Salesforce are called standard objects. Objects we create in organization are called custom objects

Created the custom object Temperature Data in the following way,

- Click on Setup → Object Manager → Create → Custom Object.
- Enter the Label name as Temperature Data, Plural name as Temperature Datas which be used as Tab name and Object name as Temperature\_Data which will be used as API name (Unique Name). By default, the custom objects are stored with \_\_c to access through API.
- Enter the record name Temperature Data Name which is a mandatory field while creating a custom object and select its data type as Auto-Number.
- Next, check allow reports, activities and track field history check box fields, which are basically for availing report creation, assigning tasks to the user for a particular Temperature Data record and tracking the history of all field edits and updates.

In similar fashion, For the Monitor Temperature and Analysis App, one more custom object named Statistics was created for tracking email alerts.

The objects can be accessed by tabs in salesforce.com and menu items as in the salesforce1 platform which is a mobile platform.

### 4.2.3 Fields

Salesforce has provided many inbuilt fields called standard fields like “Name,” “Owner,” “Created By,” “Last Modified By” etc. Every object in Salesforce has a set of standard fields that may be applicable for capturing data for that type of an object.

Salesforce also allows users to create new fields in the system to capture additional information. These fields are called as Custom Fields. Salesforce supports a variety of data types for these fields like Text, Text Area, Rich Text Area, Number, Currency, Boolean, Email, Phone, etc.

Administrators can then configure how these fields should be set (visible/read-only/edit/mandatory).

Created the custom fields in the following way,

- Go To Temperature Data Object from Object Manager in Setup
- Fields & Relationships → New → Select Data type
- Define the field names → click Save & Next.

For the Monitor Temperature and Analysis App, following custom fields were created:

1. **Celsius:** Number field which stores temperature in Celsius
2. **Critical Temperature:** Text Field which stores the critical temperature range
3. **Fahrenheit:** Formula Number field which calculate and store temperature in Fahrenheit.
4. **Heat Index:** It is the temperature feels like to the human body when relative humidity is combined with the air temperature. Number field stores real feel temperature.
5. **Humidity:** Percent field which stores the humidity percentage.
6. **Is Critical:** Checkbox field which indicate whether temperature is critical or not.
7. **Statistics:** Lookup field of statistics object for tracking of email alerts.

### 4.2.4 Relationships

The Force.com platform supports two types of parent-child relationships between objects. They are lookup relationships and master-detail relationships. These relationships connect objects with other objects. These both work like a foreign-key relationship in a relational database. These relations are the fields in the Salesforce. We can create them as we create custom fields.

**The lookup relationship** creates a simple relation between two objects. With this relationship field, we can navigate from records in one object to the related records in another object and can create one-to-one and one-to-many relationships. Lookup relationships are appropriate when a relationship between two objects is required in some cases, but not always. In scenarios like to relate multiple parent records to the child record and to reference commonly shared data, such as reference data, a lookup relationship is used.

**The master-detail relationship** is a powerful relationship which is based on a parent-child relationship. The object on which we create a master-detail relationship is the child, and the other



one is the parent or master object which will be referenced as a field in the child object. The sharing settings of the child object can be taken from their master object. In a master-detail relationship, if we delete master object records, child objects also automatically deleted. With the master-detail relationship, we can create a roll-up summary field on parent objects.

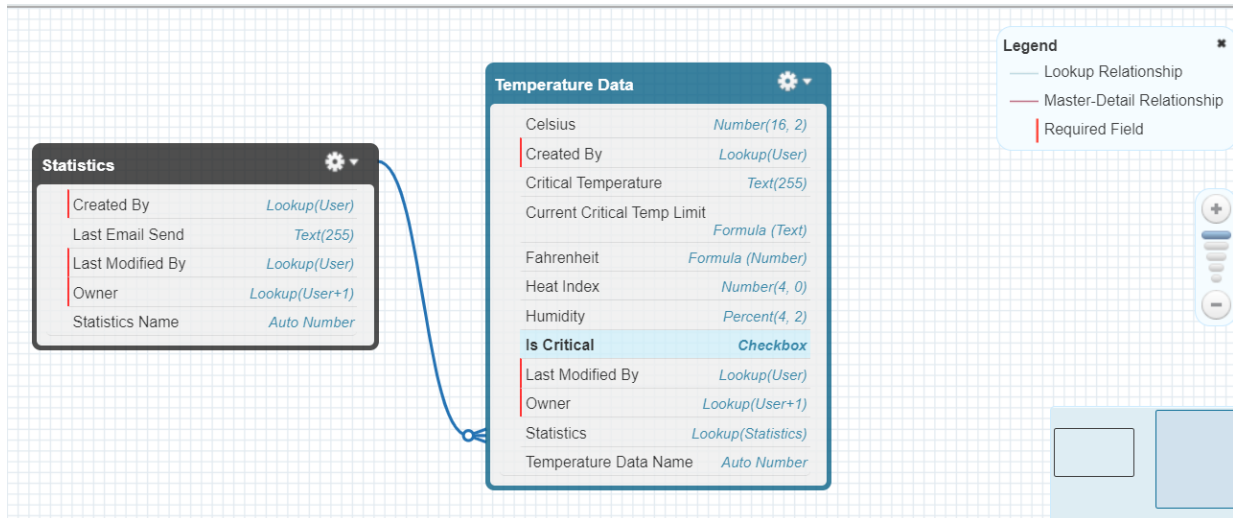


Figure 11: Lookup relationship

## 4.2.5 Tabs

Tabs are the primary way to access the main objects in the application for users in Force.com platform apps. Just like standard objects/fields Salesforce also has the standard/custom tabs.

Tabs provide an interface to access records for that particular object. Most standard objects have their tabs exposed. Users can also configure which tabs they would like to see and can rearrange their order. Users can create custom tabs for the custom objects they create. Tabs can also be used to open custom pages and links.

Created a custom Tab to the object Monitor Temperature and Analysis in the following way,

Setup → Tabs → Monitor Temperature and Analysis

## 4.2.6 Apex

Salesforce introduced Apex as the first cloud computing programming language. The syntax of Apex is quite similar to Java. It is particularly intended for building business applications to oversee Data and procedures larger ambience of the Force.com platform. The Apex lets the developers focus just on elements specific to their application by providing a productive approach to creating functionality and logic, leaving the rest of work for Force.com Platform.

Apex Classes are used in the Application are ESP8622WebhookListner and TemperatureWrapperClass.

## 4.2.7 Reports and Dashboards

As we know a company deals with lots of data, to check the data record by record is a very lengthy and time taking process. A good application should provide the overview of the data at a glance. For this purpose, Salesforce provided a tool called **Reports**. Using reports, we can generate different formats of data resides in the company at a glance view.

A dashboard is a place where we can put all reports together to give the overview of the statistics. For example, if the client wants to know the average temperature in this month then, we can create reports and find these statistics. The Reports can be created as follows,

Tab Section → Reports Tab → New Report

A Dashboard can show the reports in a visual format like Charts, Gauges, Tables, Metrics, or Visualforce pages. Force.com platform supports 20 components to be allowed in a dashboard for an organization. For example, the reports which we created before, like Pie chart and Vertical bar chart we can put up to 20 components in a dashboard. For the dashboard also we created a folder and saved all the dashboards under Dashboards folder and shared same as Reports folder. Dashboard can be created as,

Tab Section → Dashboards Tab → New Dashboard

We can also subscribe to the dashboard which will provide an email of a fixed cycle report of our dashboard.

## 4.2.8 Process Builder

Process Builder Salesforce is an automated tool that allows us to control the order of actions or evaluate the criteria for a record. It has eight actions associated with it:

- **Creating records:** This will allow you to create a new record and add different field values for it.
- **Updating records:** You can update one or more records that are somehow related to the record that started the process. This can be done either by manually entering records or by using the records from a related record.
- **Submitting the record automatically for approval:** The record that started your process can be submitted. Any other record cannot be automatically submitted.
- **Calling/triggering Apex code:** You can invoke an Apex code that you have already written within Salesforce.
- **Invoking another process:** This action will actually invoke a process to another process.

Process Builder in Salesforce supports three types of processes for automation. These types determine what will trigger the process:

- A change in a record starts whenever the record is updated or created.
- An event process will start whenever an event message is received.
- An invocable process will start whenever another process requests for it.

Each of the above processes consists of an associated criteria that determine when an action group should be executed and action groups that comprise an immediate or scheduled action

### 4.2.9 Salesforce Sites

Force.com sites enables users to build or create public website using visualforce pages and applications using force.com platform that integrates directly with salesforce.com. Force.com sites are hosted in Force.com Servers so there are no data integration issues.

Using Force.com sites you can build and serve public websites and applications directly from the Force.com platform. Force.com Sites are built using Apex and Visualforce but can also include simple HTML mark-up, JavaScript, CSS and Flash to create a visually compelling user experience.

We can create Force.com Site with our own domain name. Corporate website or intranet, Ecommerce applications, simple micro sites can be built using force.com sites. Force.com Sites is enabled for most editions but there are bandwidth and usage restrictions. We have to create the Force.com domain which is unique subdomain prefix plus force.com. Force.com domain name is used for all the site that we create in salesforce.com.

Example: - <http://esp8622wifiapi-developer-edition.ap24.force.com>

Above link is the Force.com domain name that we created in salesforce.com

To create a Force.com website go to Setup → Build → Develop → Sites.

## 4.3 System Hardware

System hardware is the collection of physical parts of the project module. It includes various modules like controllers, regulators and relay circuits.

### 4.3.1 ESP8266

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

It has 11 GPIO pins\* (General Purpose Input/output pins), and an analog input as well. This means that we can program it like any normal Arduino or other microcontroller. And on top of that, we get Wi-Fi communication, so we can use it to connect to your Wi-Fi network, connect to the Internet, host a web server with real web pages, let our smartphone connect to it, etc.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area.

It can also fetch data from internet using API's hence your project could access any information that is available in the internet, thus making it smarter.

We can program the ESP8266 using the Arduino IDE and also with Python Language.

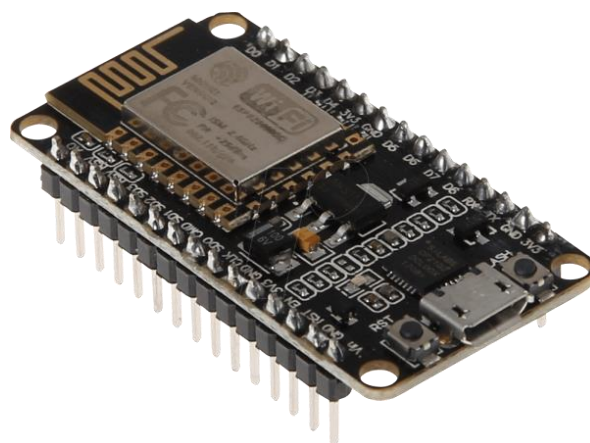


Figure 12: NodeMCU-ESP8266

### 4.3.2 Temperature Sensor

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc. to measure humidity and temperature instantaneously.

DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor. DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

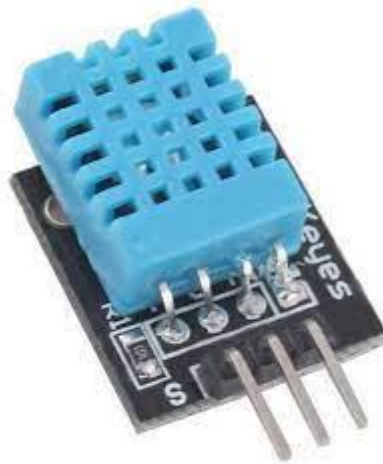


Figure 13: DHT11 Temperature Sensor

## 4.4 Application Program Interface (API)

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time we use an app like Facebook, send an instant message, or check the weather on your phone, we're using an API. Let us take an example,

Imagine you're sitting at a table in a restaurant with a menu of choices to order from. The kitchen is the part of the "system" that will prepare your order. What is missing is the critical link to communicate your order to the kitchen and deliver your food back to your table. That's where the waiter or API comes in. The waiter is the messenger – or API – that takes your request or order and tells the kitchen – the system – what to do. Then the waiter delivers the response back to you; in this case, it is the food.

Here is a real-life API example. You may be familiar with the process of searching flights online. Just like the restaurant, you have a variety of options to choose from, including different cities, departure and return dates, and more. Let us imagine that you're booking you are flight on an airline website. You choose a departure city and date, a return city and date, cabin class, as well as other variables. In order to book your flight, you interact with the airline's website to access their database and see if any seats are available on those dates and what the costs might be.

#### 4.4.1 Working of API

APIs communicate through a set of rules that define how computers, applications or machines can talk to each other. The API acts as a middleman between any two machines that want to connect with each other for a specified task. There are APIs for virtually every machine or system that expects to interact with other machines or systems.

For Example when we sign into Facebook from our phone we are telling the Facebook application that we would like to access our account. The mobile application makes a call to an API to retrieve our Facebook account and credentials. Facebook would then access this information from one of its servers and return the data to the mobile application.

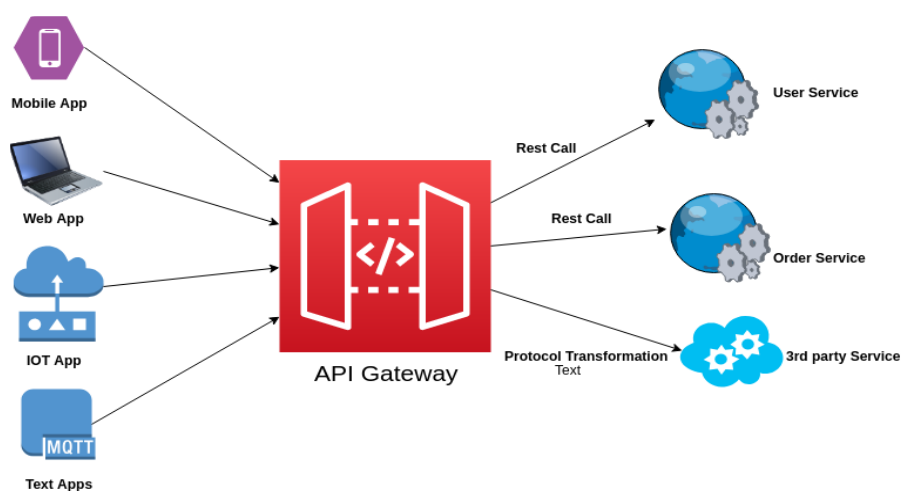


Figure 14: Working of API

### 4.4.2 RESTful APIs

RESTful Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.

An Application Programming Interface (API) is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange. RESTful Web Services make use of HTTP protocols as a medium of communication between client and server.

A client sends a message in form of a HTTP Request and the server responds in the form of an HTTP Response. This technique is termed as Messaging.

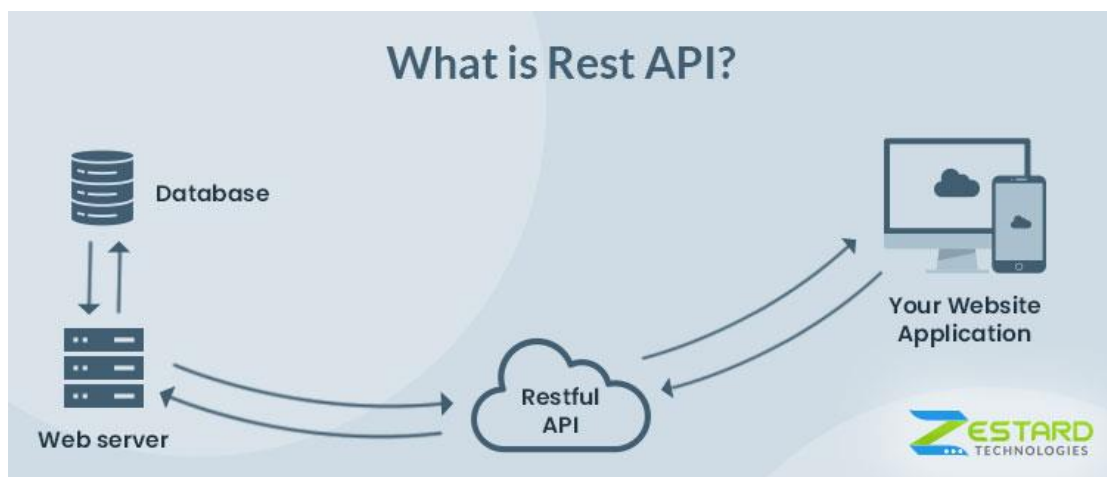


Figure 15: Restful API

### 4.4.3 REST Architecture

REST stands for REpresentational State Transfer. REST is web standards based architecture and uses HTTP Protocol. In REST architecture, a REST Server simply provides access to resources and REST client accesses and modifies the resources. Here each resource is identified by URIs/global IDs. REST uses various representation to represent a resource like text, JSON, XML. JSON is the most popular one.

Following four HTTP methods are commonly used in REST based architecture.

- GET – Provides a read only access to a resource.
- POST – Used to create a new resource.
- DELETE – Used to remove a resource.
- PUT – Used to update a existing resource or create a new resource.

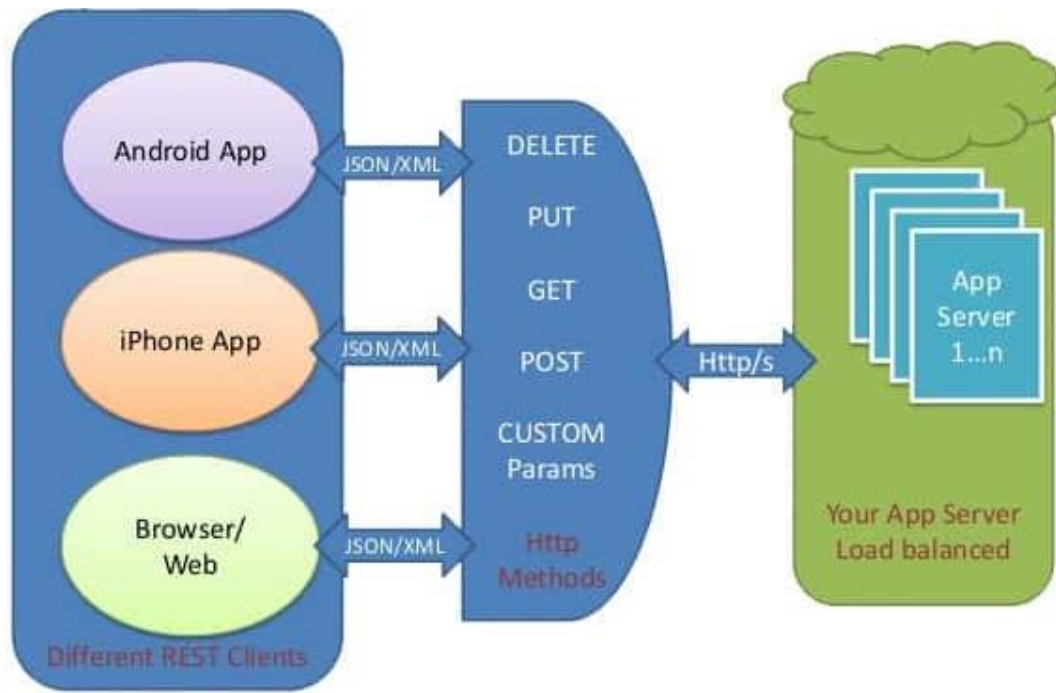


Figure 16: REST API Architecture

#### 4.4.4 HTTP Request

An HTTP request is an action to be performed on a resource identified by a given Request-URL. Request methods are case-sensitive, and should always be noted in upper case. There are various HTTP request methods, but each one is assigned a specific purpose.

HTTP requests work as the intermediary transportation method between a client/application and a server. The client submits an HTTP request to the server, and after internalizing the message, the server sends back a response. The response contains status information about the request.

An HTTP Request has five major parts –

- Verb – Indicates the HTTP methods such as GET, POST, DELETE, PUT, etc.
- URI – Uniform Resource Identifier (URI) to identify the resource on the server.
- HTTP Version – Indicates the HTTP version. For example, HTTP v1.1.
- Request Header – Contains metadata for the HTTP Request message as key-value pairs. For example, client (or browser) type, format supported by the client, format of the message body, cache settings, etc.
- Request Body – Message content or Resource representation.



### 4.4.5 HTTP Response

HTTP Response is the server's information as a result of the client's request. Additionally, it acts as an acknowledgment that the performance of the requested action is successful. In case there is an error in carrying out the client's request, the server responds with an error message. Moreover, the HTTP responses come as plain text formatted in either JSON or XML format, just like the HTTP requests. In the next section, let us see how an HTTP response looks.

An HTTP response is made by a server to a client. The aim of the response is to provide the client with the resource it requested, or inform the client that the action it requested has been carried out; or else to inform the client that an error occurred in processing its request.

An HTTP response contains:

- Status/Response Code – Indicates the Server status for the requested resource. For example, 404 means resource not found and 200 means response is ok.
- HTTP Version – Indicates the HTTP version. For example HTTP v1.1.
- Response Header – Contains metadata for the HTTP Response message as key value pairs. For example, content length, content type, response date, server type, etc.
- Response Body – Response message content or Resource representation.

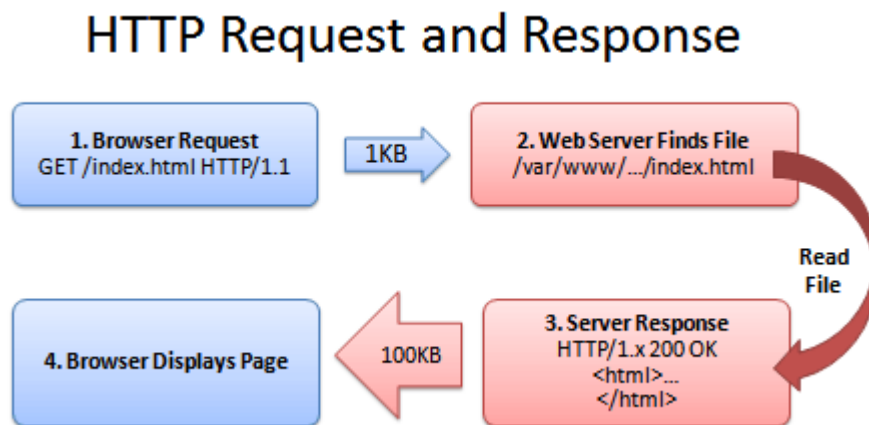


Figure 17: HTTP Request and Response

## 4.4.6 Webhooks

Webhooks are one of a few ways web applications can communicate with each other. It allows you to send real-time data from one application to another whenever a given event occurs.

It is used to notify you any time someone checks in, so you'd be able to run any processes that you had in your application once this event is triggered. The data is then sent over the web from the application where the event originally occurred, to the receiving application that handles the data. With most APIs there's a request followed by a response. No request is required for a webhook, it just sends the data when it's available.

To use a webhook, you register a URL with the company providing the service. That URL is a place within your application that will accept the data and do something with it. In some cases, you can tell the provider the situations when you'd like to receive data. Whenever there's something new, the webhook will send it to your URL.

This exchange of data happens over the web through a "webhook URL." A webhook URL is provided by the receiving application, and acts as a phone number that the other application can call when an event happens.

Only it's more complicated than a phone number, because data about the event is sent to the webhook URL in either JSON or XML format. This is known as the "payload." Here's an example of what a webhook URL looks like with the payload it's carrying:

```
https://yourapp.com/data/12345?customer=bob?value=10.00?item=paper  
To: yourapp.com/data/12345  
Customer: Bob  
Value: 10.00  
Item: Paper
```

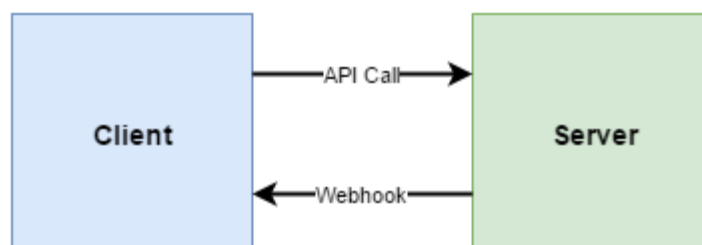


Figure 18: Webhook

## Chapter 5: The Functionality of Monitor Temperature and Analysis

The Monitor Temperature and Analysis can be divided into three parts:

1. Sending Data to the Cloud (IoT Platform)
2. Monitoring and Analysis of Data (Salesforce Cloud Platform)
3. User Functionality

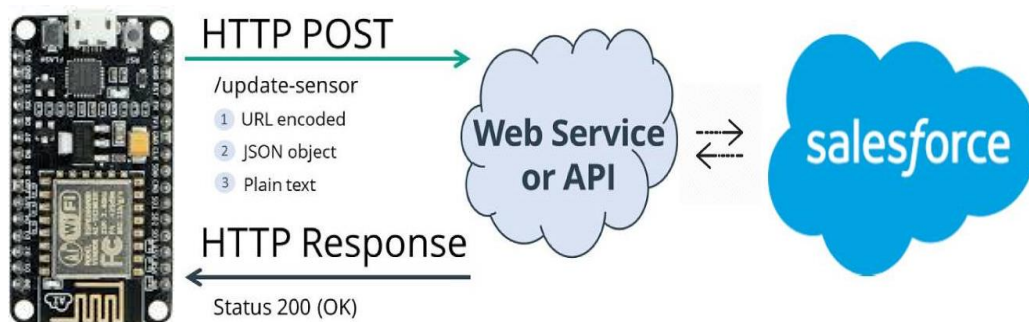


Figure 19: Project Functionality

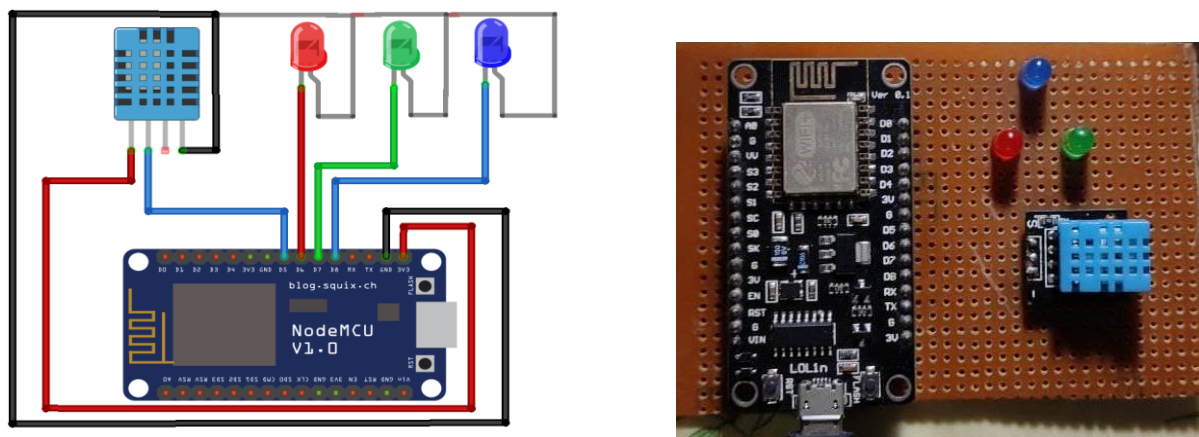


Figure 20: Circuit Diagram

### 5.1 Sending Data to Salesforce Cloud (IoT Platform)

Since the Internet of things (IoT) is the inter-networking of physical devices that enable these objects to collect and exchange data. IoT allows objects to be controlled and gather information remotely across the already established network.

So we use IoT technology to send the data to Salesforce cloud service.

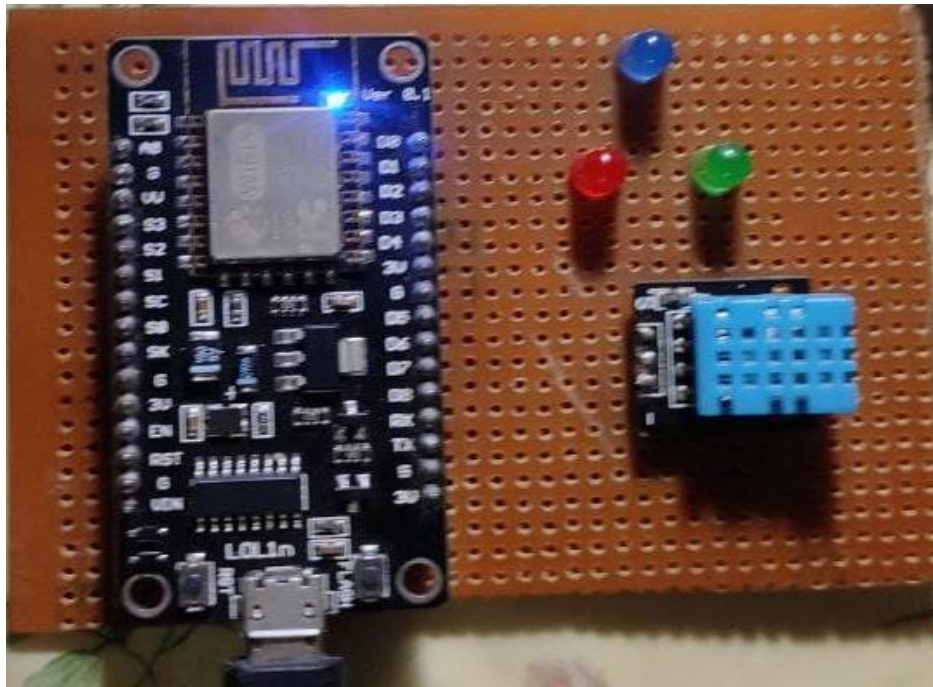


Figure 21: Finding Wi-Fi Connection

```
COM5
.....scandone
18:09:02.152 -> state: 0 -> 2 (b0)
18:09:02.195 -> state: 2 -> 3 (0)
18:09:02.195 -> state: 3 -> 5 (10)
18:09:02.195 -> add 0
18:09:02.195 -> aid 7
18:09:02.195 -> cnt
18:09:02.237 -> .
18:09:02.324 -> connected with SADMG, channel 6
18:09:02.324 -> dhcp client start...
18:09:02.477 -> .ip:192.168.1.16,mask:255.255.255.0,gw:192.168.1.1
18:09:02.641 -> Connected to WiFi network

[Autoscroll] [Show timestamp] [No line ending] [115200 baud] [Clear output]
```

Figure 22: Successfully Connected to Wi-Fi

### 5.1.1 Sending Data using NodeMCU

The sensor collects Temperature, Humidity and Heat Index data and NodeMCU posts a GET request to the Salesforce Domain Server. After posting request, the API returns a response code 200 which means data is successfully sent to the Salesforce Server. Also a response body is returned which indicates the status i.e. "Record Inserted Successfully". A green LED will get turned ON for 1 minute indicating the API is successful.

If response body consists of "Record Inserted Successfully Turn On" then Air Conditioner or any other appliance connected to NodeMCU will get automatically turned ON. A blue LED will also get turned ON indicating the Air Conditioner is ON.

Afterwards, if the temperature condition becomes normal i.e. Temperature is below critical range then the appliance connected to device will get automatically turned OFF.

If the API is unsuccessful to request the data then the red LED will get turned on for 4 mins.

NOTE: After every 5 Minutes the API will get hit. This delay is customizable and can be changed according to the business requirement.

```
COM5
18:19:01.062 -> Humd:86.0
18:19:01.062 -> Temp:30.0
18:19:01.062 -> HeatIndex:39.5
18:19:01.112 -> {"humidity":"86.00","temp":"30.00","heatIndex":"39.46"}
18:19:03.853 -> HTTP Response code: 200
18:19:03.893 -> RESPONSE STATUS: Record inserted Successfully
18:19:03.893 -> Turned OFF Air Conditioner
```

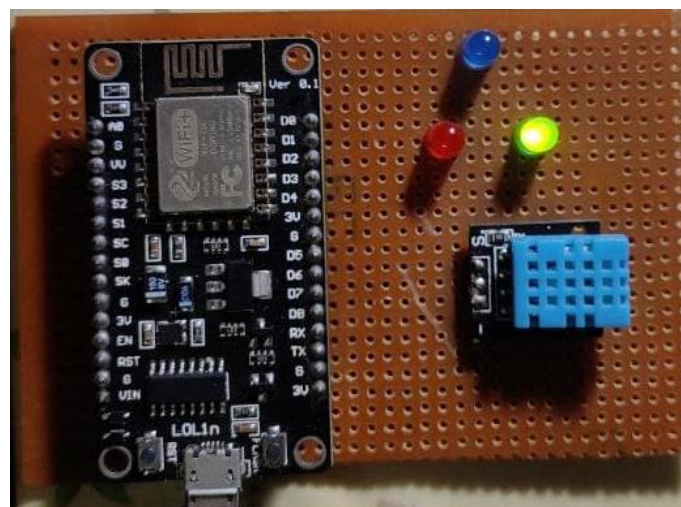


Figure 23: NodeMCU Response After hitting API

## 5.2 Monitoring and Analysis of Data (Salesforce Cloud Platform)

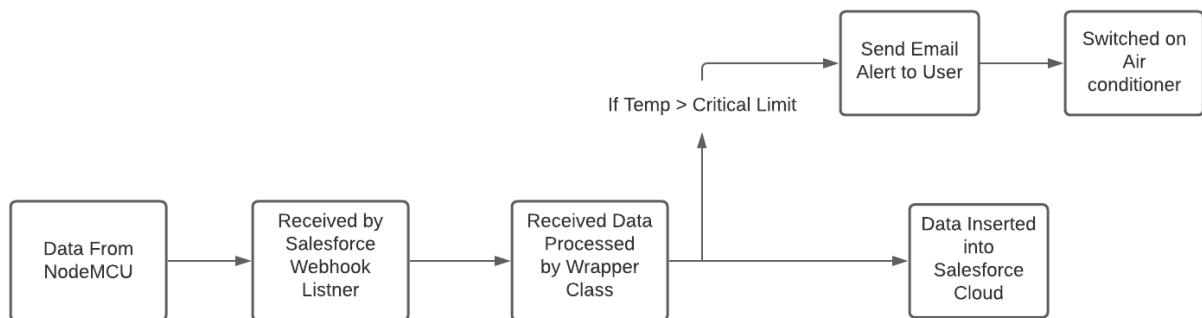



Figure 24: Salesforce Functionality

### 5.2.1 Salesforce Debug Logs

Salesforce debug logs manages the track of events (transactions) that happens in the Salesforce organization. It contains information about all the transactions that are happening on the Salesforce and keeps track of time, the status of transactions, etc.

Salesforce debug logs generates when a user uses Trace Flag. Trace flags filter the logs generated by the transaction. It contains debug level, start-end time, type of the log (ERROR, WARN, DEBUG), and the status of the job/transactions. Once you set the Trace Flag, the system will generate the debug log when a user performs the transaction. These logs can be useful for developers and integration partners.

Salesforce Debug Logs can hold the information of Database changes, HTTP requests, Resources used and errors in Apex, Automated workflow rules and Status of the transactions.

 **Debug Logs**

### Debug Logs

[Help for this Page](#)

A debug log records database operations, system processes, and errors that occur when executing a transaction or while running unit tests. The system generates a debug log for a user every time that user executes a transaction and the user has a trace flag with start and expiration dates that contain the transaction's start time. You can monitor and retain debug logs for the users specified below.

One SFDC\_DevConsole debug level is shared by all DEVELOPER\_LOG trace flags in your org.

View: All [Create New View](#)

User Trace Flags						
Action	Name	Log Type	Requested By	Start Date	Expiration Date	Debug Level Name
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">Filters</a>	Goyal, Munish	DEVELOPER_LOG	Munish Goyal	24/05/2021, 5:40 pm	24/05/2021, 5:45 pm	SFDC_DevConsole
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">Filters</a>	Site Guest User: ESP8622 Wifi	USER_DEBUG	Munish Goyal	24/05/2021, 5:26 pm	25/05/2021, 5:26 am	SFDC_DevConsole

[Previous Page](#) | [Next Page](#)

### Debug Logs

[Delete All](#)

User	Request Type	Application	Operation	Status	Duration (ms)	Log Size (bytes)	Start Time
<a href="#">View</a>   <a href="#">Download</a>   <a href="#">Delete</a>	Api	Unknown	/ESP8622WebhookListener	Success	265	23,841	05/24 17:32:23

[Delete All](#)

Figure 25: Salesforce Debug Log Section



Apex Debug Log		Help for this Page
ESP8622 Wifi Site Guest User		
		Download Delete
Apex Debug Log Detail		
User	ESP8622 Wifi Site Guest User	
Status	Success	
Request Type	Api	
Duration (ms)	265	
Log	<pre> 51.0 APEX_CODE,FINEST;APEX_PROFILING,INFO;CALLOUT,INFO;DB,INFO;NBA,INFO;SYSTEM,DEBUG;VALIDATION,INFO;VISUALFORCE,INFO;WAVE,INFO;WORKFLOW,INFO 12:02:23.0 (517607) USER_INFO [EXTERNAL] 0055g000003v7X5 esp8622_wifi@esp8622wifiapi-developer-edition.ap24.force.com (GMT+00:00) Greenwich Mean Time 12:02:23.0 (573587) EXECUTION_STARTED 12:02:23.0 (579356) CODE_UNIT_STARTED [EXTERNAL] 01p5g000006whSI ESP8622WebhookListner.doGet() 12:02:23.0 (898324) HEAP_ALLOCATE [79] Bytes:3 12:02:23.0 (989431) HEAP_ALLOCATE [84] Bytes:152 12:02:23.0 (1010656) HEAP_ALLOCATE [399] Bytes:408 12:02:23.0 (1028704) HEAP_ALLOCATE [412] Bytes:408 12:02:23.0 (1041703) HEAP_ALLOCATE [520] Bytes:48 12:02:23.0 (1070988) HEAP_ALLOCATE [139] Bytes:6 12:02:23.0 (1125445) HEAP_ALLOCATE [EXTERNAL] Bytes:252 12:02:23.0 (1131793) HEAP_ALLOCATE [EXTERNAL] Bytes:25 12:02:23.0 (1153286) SYSTEM_METHOD_ENTRY [1] RestRequest.RestRequest() </pre>	

Figure 26: A Debug Log

## 5.2.2 Data Processing

The data received by webhook listener is then processed by a wrapper class (an apex class).

A wrapper or container class is a class, a data structure which contains different objects or collection of objects as its members. A wrapper class is a custom object defined by a programmer wherein he defines the wrapper class properties.

```

File Edit Debug Test Workspace Help < >
TemperatureWrapperClass.apxc
Code Coverage: None API Version: 51
1 public class TemperatureWrapperClass {
2     public Double humidity;
3     public Double temp;
4     public Double heatIndex;
5 }

```

Figure 27: Wrapper class used for processing data

Once, the data is processed then ESP8622WebhookListener apex class gets executed and checks the conditions with respect to processed data. And the record is inserted.

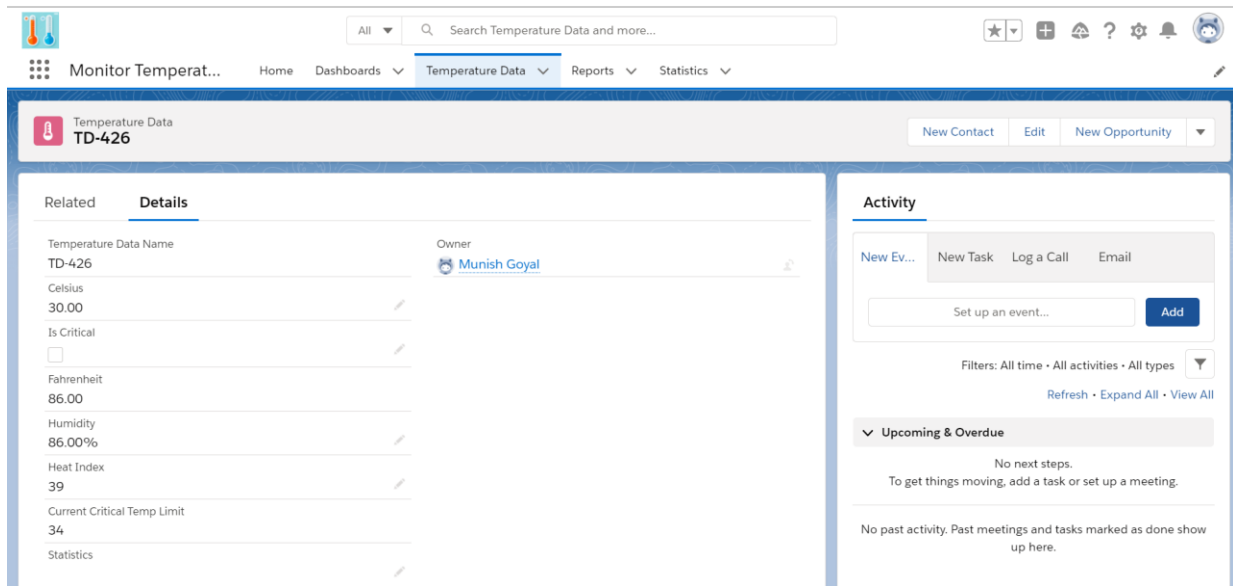


Figure 28: Record Inserted in Salesforce

During apex class execution, if the temperature is greater or equal to the critical limit then an email alert is sent to the client's email id indicating that the temperature at his/her store or warehouse is critical. It also specifies that air condition is also turned on automatically to maintain the temperature.

### 5.2.3 Salesforce Email Alerts

Email alerts are emails generated by an automated process and sent to designated recipients. These actions consist of the standard text and list of recipients for an email. You can associate email alerts with processes, flows, workflow rules, approval processes, or entitlement processes. They're also available through the Invocable Actions REST API endpoint.

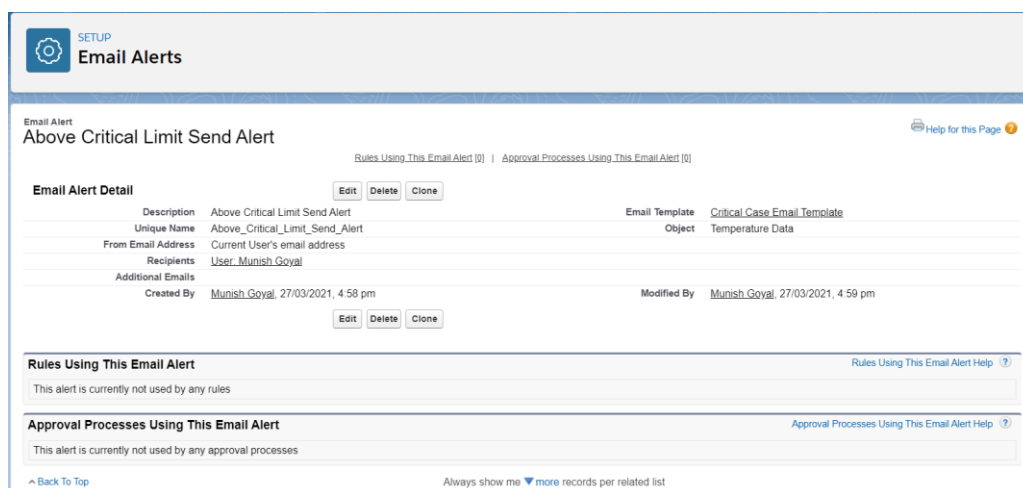


Figure 29: Salesforce Email Alert Section



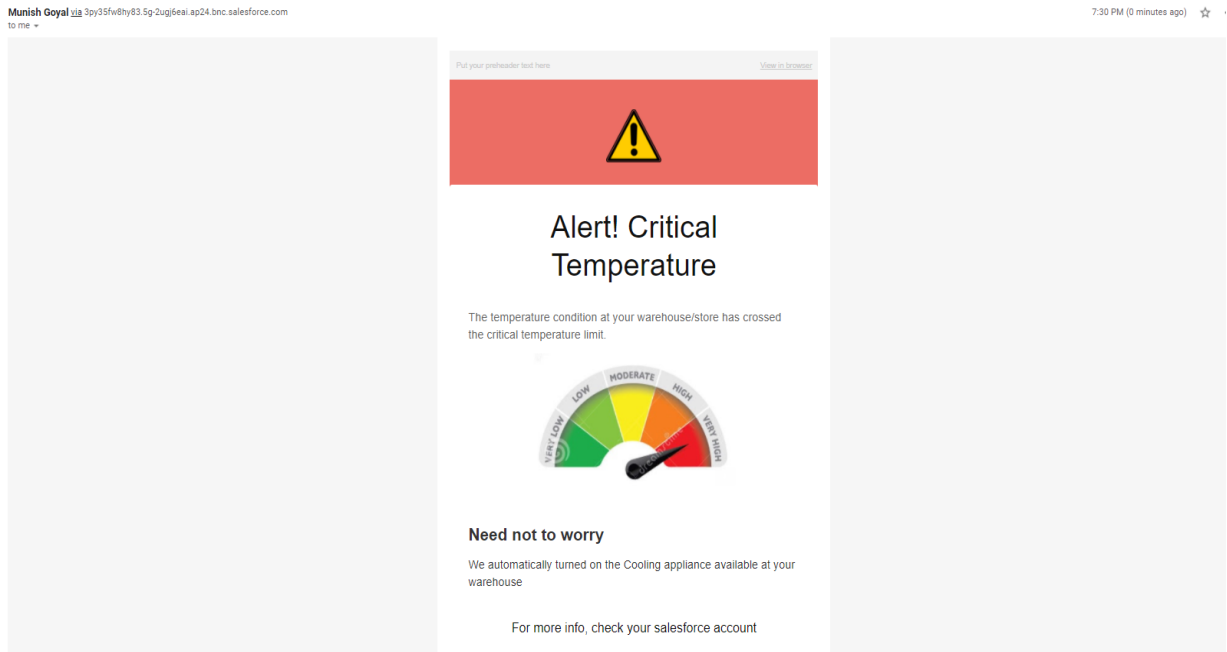


Figure 30: Email Alert Template

### 5.3 User Functionality

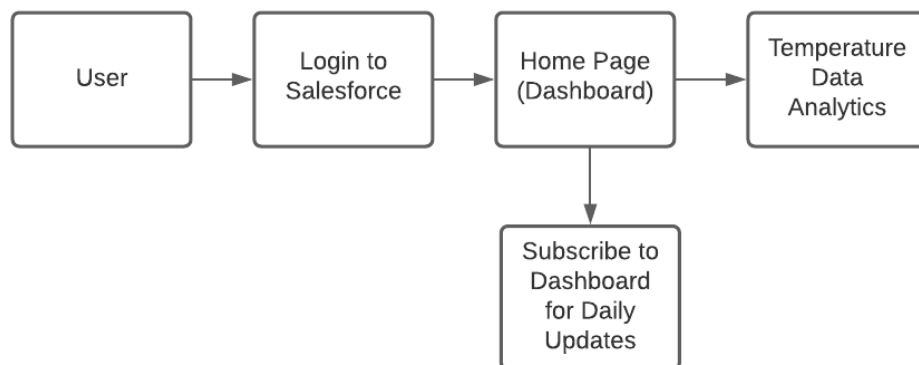


Figure 31: Functionality for User

The user can do Data Analytics according to their business requirement and also can subscribe to the dashboard for daily updates.

To do analytics, User needs to log into Salesforce with his credentials.

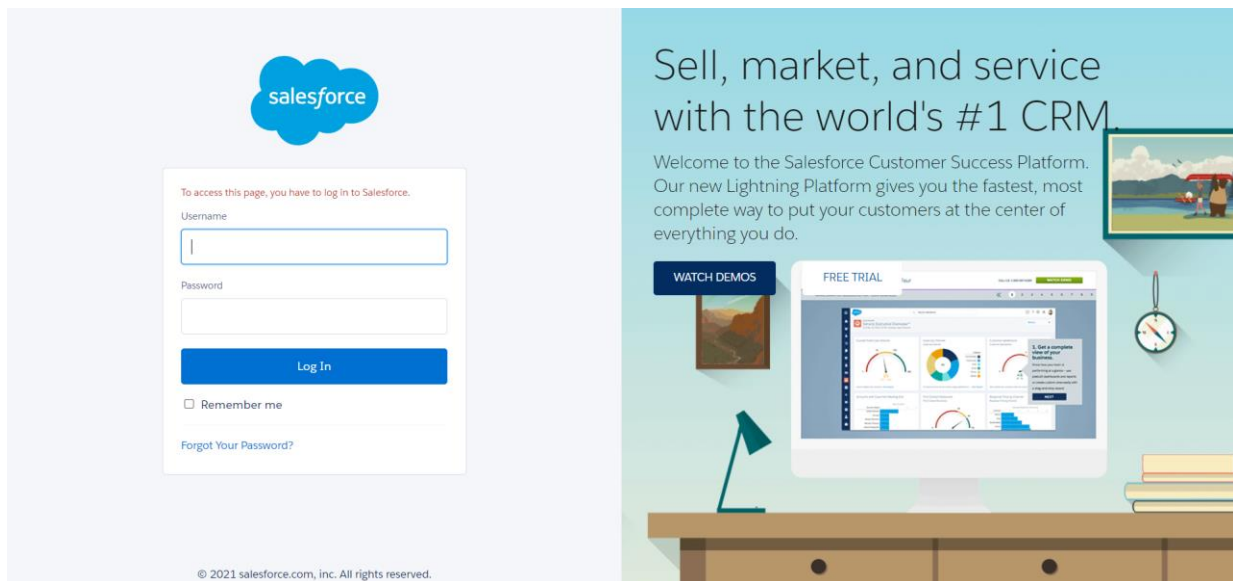


Figure 32: Login Page of Salesforce

Once the admin logs into the Salesforce, the log in page is redirected to home page. In the home page, the user finds Standard Page components like apps, tabs, menu and he will also find custom modules like calendar, task to perform, etc., In the home page, the user can navigate between all the tabs, existing applications, and other components.

Home Page can be customized depending on the requirements. The layout of the page is fixed while the components like tabs can be customized. In this app, home page is customized.

In the customized home page, manager can view Dashboards containing Statistics of data

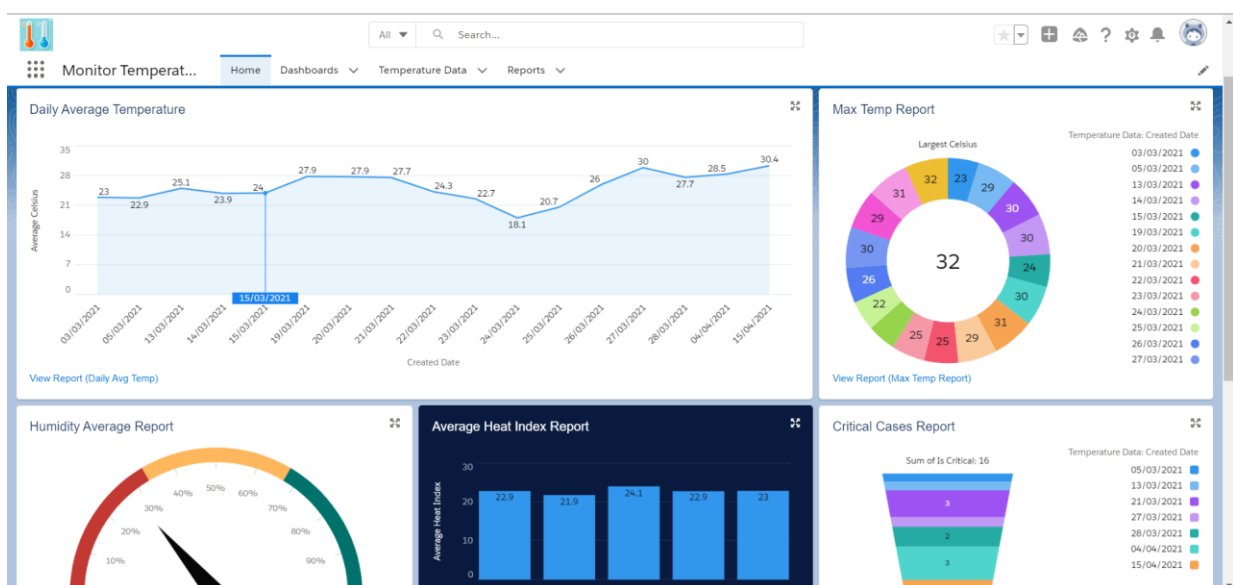
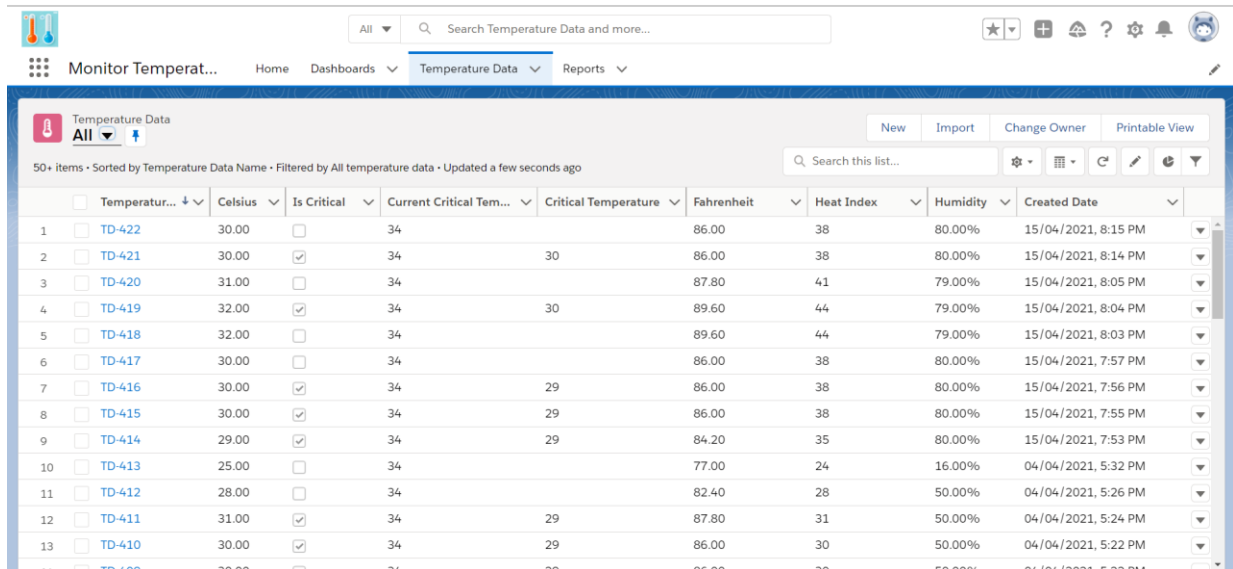


Figure 33: Home Page of Temperature Monitor and Analysis

Temperature Data tab has the database of temperature data records received by NodeMCU. The user has access to delete and view the records available. The feature to arrange the Temperature Data by filters like recently viewed, recently created and recently modified is also available.

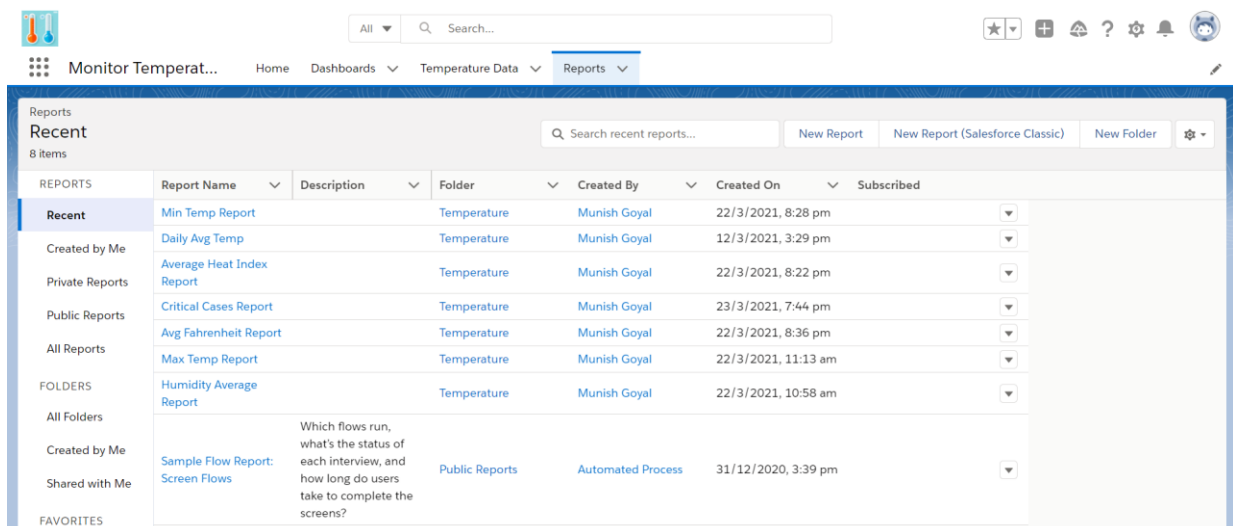


	Temperature...	Celsius	Is Critical	Current Critical Tem...	Critical Temperature	Fahrenheit	Heat Index	Humidity	Created Date
1	TD-422	30.00	<input type="checkbox"/>	34		86.00	38	80.00%	15/04/2021, 8:15 PM
2	TD-421	30.00	<input checked="" type="checkbox"/>	34	30	86.00	38	80.00%	15/04/2021, 8:14 PM
3	TD-420	31.00	<input type="checkbox"/>	34		87.80	41	79.00%	15/04/2021, 8:05 PM
4	TD-419	32.00	<input checked="" type="checkbox"/>	34	30	89.60	44	79.00%	15/04/2021, 8:04 PM
5	TD-418	32.00	<input type="checkbox"/>	34		89.60	44	79.00%	15/04/2021, 8:03 PM
6	TD-417	30.00	<input type="checkbox"/>	34		86.00	38	80.00%	15/04/2021, 7:57 PM
7	TD-416	30.00	<input checked="" type="checkbox"/>	34	29	86.00	38	80.00%	15/04/2021, 7:56 PM
8	TD-415	30.00	<input checked="" type="checkbox"/>	34	29	86.00	38	80.00%	15/04/2021, 7:55 PM
9	TD-414	29.00	<input checked="" type="checkbox"/>	34	29	84.20	35	80.00%	15/04/2021, 7:53 PM
10	TD-413	25.00	<input type="checkbox"/>	34		77.00	24	16.00%	04/04/2021, 5:32 PM
11	TD-412	28.00	<input type="checkbox"/>	34		82.40	28	50.00%	04/04/2021, 5:26 PM
12	TD-411	31.00	<input checked="" type="checkbox"/>	34	29	87.80	31	50.00%	04/04/2021, 5:24 PM
13	TD-410	30.00	<input checked="" type="checkbox"/>	34	29	86.00	30	50.00%	04/04/2021, 5:22 PM

Figure 34: Temperature Data Tab View

The Report tab consists of all the Statistics reports of the data created by user. These reports gives us easy understanding of the data analysis.

User can also subscribe to an individual report to receive emails regarding updates.



REPORTS	Report Name	Description	Folder	Created By	Created On	Subscribed
Recent	Min Temp Report		Temperature	Munish Goyal	22/3/2021, 8:28 pm	<input type="checkbox"/>
Created by Me	Daily Avg Temp		Temperature	Munish Goyal	12/3/2021, 3:29 pm	<input type="checkbox"/>
Private Reports	Average Heat Index Report		Temperature	Munish Goyal	22/3/2021, 8:22 pm	<input type="checkbox"/>
Public Reports	Critical Cases Report		Temperature	Munish Goyal	23/3/2021, 7:44 pm	<input type="checkbox"/>
All Reports	Avg Fahrenheit Report		Temperature	Munish Goyal	22/3/2021, 8:36 pm	<input type="checkbox"/>
	Max Temp Report		Temperature	Munish Goyal	22/3/2021, 11:13 am	<input type="checkbox"/>
FOLDERS	Humidity Average Report		Temperature	Munish Goyal	22/3/2021, 10:58 am	<input type="checkbox"/>
All Folders	Sample Flow Report: Screen Flows	Which flows run, what's the status of each interview, and how long do users take to complete the screens?	Public Reports	Automated Process	31/12/2020, 3:39 pm	<input type="checkbox"/>

Figure 35: Reports Tab View

The Dashboard tab consists of all the Dashboard created by user. These Dashboard shows a mesmerising view of all the reports in an organizing way.

User can create as many dashboards containing different reports. User can also subscribe to different dashboards to receive emails regarding updates.

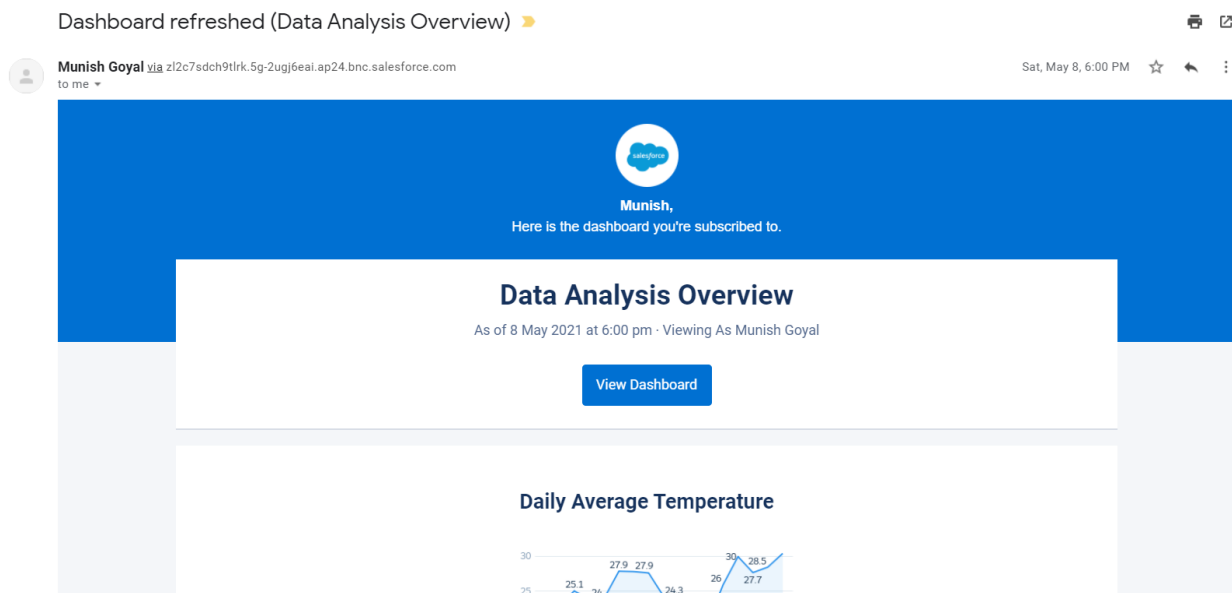


Figure 36: Email View of Dashboard

## 5.4 Salesforce Mobile App View

The Salesforce mobile app is an enterprise-class app that provides your users with instant access to your company's CRM data from a phone or tablet.

This mobile CRM contains many of the features already contained in the desktop version and has potential for more. Hence, Salesforce Mobile CRM is gaining much popularity due to the problems it solves. As an industry leading CRM platform, Salesforce offers most of the features one can ask for in its mobile CRM. Therefore, there are a number of benefits to be leveraged from the Salesforce Mobile App. Once you have implemented the mobile version of Salesforce, the only step left to reap most of its benefits would be to train your sales team in using it.

Some of the benefits mobile CRM provides are:

1. Easier Access to Information
2. Availability of More Accurate Information

3. Improved Functionality
4. Offline Mobility of Salesforce

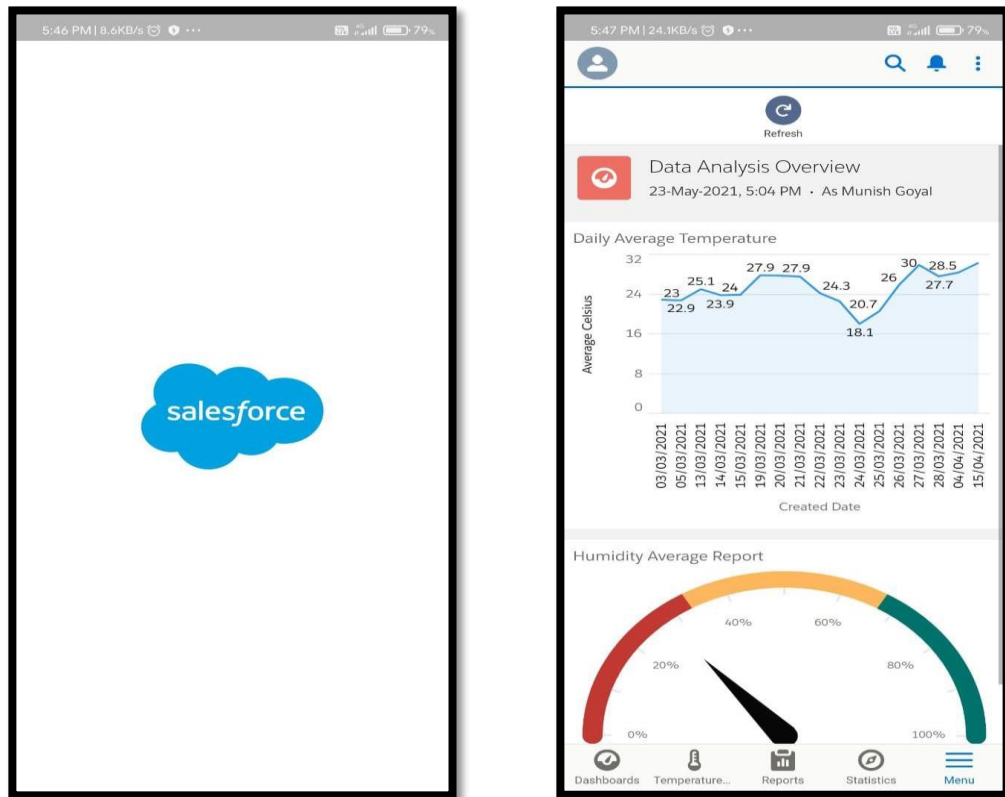


Figure 37: Salesforce Mobile View

Using Salesforce mobile app, client can easily manage his business wherever in the world. Records can be updated and all analysis can be viewed in app.

## **Chapter 6: Conclusion and Future Scope**

### **Conclusion**

The project titled as Monitor Temperature and Analysis is an IoT based project which is to use to improve business requirements by keeping the temperature conditions normal and doing all analysis. It is designed in Salesforce platform plus IoT platform. In this system, user can manage the record of temperature inserted by NodeMCU IoT device and can do all the analysis as per their need. This system is very interesting. This software is developed with scalability in mind. Additional modules can be easily added when necessary.

The software is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work successfully. Thus the system has fulfilled all the objectives identified and is able to get implemented in real world.

The project has been completed successfully with the maximum satisfaction of the organization. The constraints are met and overcome successfully. The system is designed as like it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. This software has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

The lessons learnt after developing this project are as follows:

- Before developing any project the requirement should be made well clear so that after developing it the programmer does not have to change it.
- Software and hardware constraints should be kept in mind.
- Time and cost are those constraints, which are never told but always accounted for.
- Project should be error free and made in such a way so that modifications can be done in future.
- A professional should have a vision to see beyond the user-defined requirement.
- All members of the project team have to be cooperative with each other.
- User should be given proper training about how to use project.

### **FUTURE SCOPE**

The future of this project depends on whether the author has enough spare time over the next 2 months to continue with the developing. The author feels that last few remaining features would round off the system. If the author was to try to sell this system then more system testing would have to be done, in a particular a more comprehensive real – world. Testing environment would have to be adopted along with some real word's usage. Multiple concurrent users would be command in real world usage but have been difficult to test for considering there was only tester

involved in this project. This type of system would benefit for the hardware in case of a system failure for the software in case of newly found bugs, in return of a subscription free.

- Add another modules/service.
- Add services for customer to update him/herself.

As this project is an IoT based project, so this project is capable of handling mains voltage i.e. 220v. According to the business requirement, this project can be upgraded so that it can easily handle 220v devices.

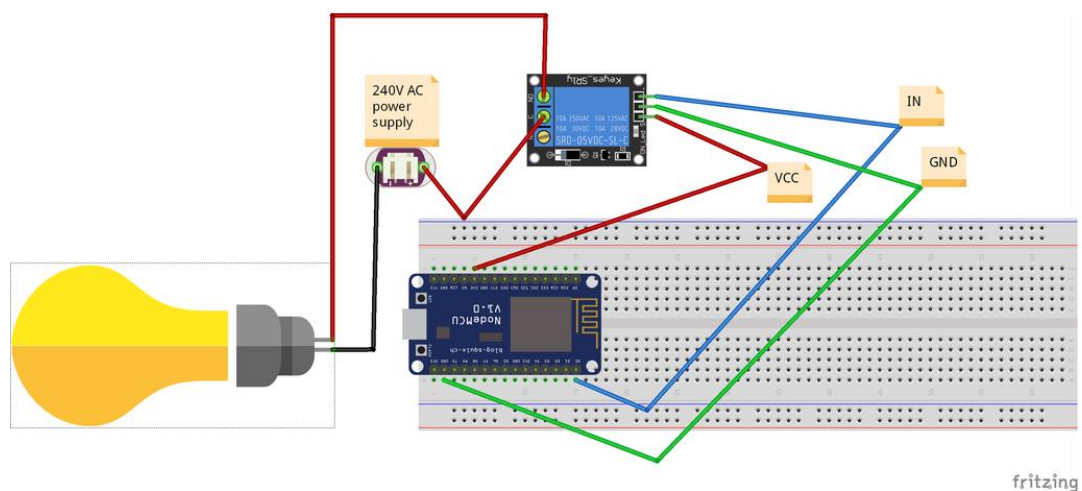


Figure 38: Future up gradation

## Appendix

Source code for NodeMCU written in Arduino IDE:

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <DHT.h>

#define DHTTYPE DHT11
#define DHTpin 14    //D5 of NodeMCU is GPIO14
#define errorPin 12   //D6
#define successPin 13 //D7
#define appliancePin 15 //D8

DHT dht(DHTpin, DHTTYPE);

const char* ssid = "SADMG";
const char* password = "S@dMg@12";

unsigned long lastTime = 0;
unsigned long timerDelay = 60*1000*5; //5 min delay
int successDelay=60*1000*2; //2min delay
int errorDelay= 60*1000*4; //4 min delay
void setup() {
    Serial.begin(115200);
    dht.begin();
    pinMode(successPin, OUTPUT);
    pinMode(errorPin, OUTPUT);
    pinMode(appliancePin, OUTPUT);
    pinMode(LED_BUILTIN, OUTPUT);
    WiFi.begin(ssid, password);
}

void loop() {
    while(WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        digitalWrite(LED_BUILTIN, HIGH);
        delay(100);
        digitalWrite(LED_BUILTIN, LOW);
        delay(100);
        if(WiFi.status() == WL_CONNECTED){
            Serial.println("Connected to WiFi network");
        }
    }

    //Serial.print("LED Blinked and Time is ");
    //Serial.println(millis() - lastTime);
    digitalWrite(LED_BUILTIN, LOW);
    delay(100);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(5000);

    if(millis() - lastTime>successDelay){
        digitalWrite(successPin, LOW);
    }
    if(millis() - lastTime>errorDelay){
        digitalWrite(errorPin, LOW);
    }

    if ((millis() - lastTime) > timerDelay) {
        if(WiFi.status()== WL_CONNECTED){

            float humidity = dht.readHumidity();
```



```

float temperature = dht.readTemperature();
float heatIndex = dht.computeHeatIndex(temperature, humidity, false);
Serial.print("Humd:");
Serial.println(humidity, 1);
Serial.print("Temp:");
Serial.println(temperature, 1);
Serial.print("HeatIndex:");
Serial.println(heatIndex, 1);

HTTPClient http;

if(String(humidity) != String("nan") && String(temperature) !=
String("nan")){
    //Serial.println("Not nan");

    String httpRequestData = "{\"humidity\":\"" + String(humidity) +
    "\",\"temp\":\"" + String(temperature) + "\",\"heatIndex\":\"" +
    String(heatIndex) + "\"}";
    Serial.println(httpRequestData);

    String serverName = "https://esp8622wifiapi-developer-
    edition.ap24.force.com/services/apexrest/ESP8622WebhookListener?temperature=
    "+httpRequestData;
    // SHA1 fingerprint of the certificate
    String fingerprint = "18 8e 9d 11 95 ed 70 15 29 02 e4 a8 cf 6f 04
    e9 da 10 74 b4";

    http.begin(serverName,fingerprint);
    int httpStatusCode = http.GET();

    Serial.print("HTTP Response code: ");
    Serial.println(httpStatusCode);
    if(httpStatusCode==200){
        String response=http.getString();
        Serial.print("RESPONSE STATUS: ");
        Serial.println(response);
        if (response.startsWith("Turn ON")) {
            Serial.println("Turned ON Air Conditioner");
            digitalWrite(appliancePin, HIGH);
        } else{
            Serial.println("Turned OFF Air Conditioner");
            digitalWrite(appliancePin, LOW);
        }
        digitalWrite(errorPin, LOW);
        digitalWrite(successPin, HIGH);
    }else{
        digitalWrite(successPin, LOW);
        digitalWrite(errorPin, HIGH);
    }
    }else{
        digitalWrite(successPin, LOW);
        digitalWrite(errorPin, HIGH);
    }
    http.end();
}
else {
    Serial.println("WiFi Disconnected");
}
lastTime = millis();
}
}

```

## Source code for Webhook Listener written in Apex language

```
@RestResource(urlMapping='/ESP8622WebhookListner')
global class ESP8622WebhookListner {
    @HttpGet
    global static void doGet() {
        RestContext.response.addHeader('Content-Type', 'application/json');
        String responseBody='';
        String tempReading = RestContext.request.params.get('temperature');
        TemperatureWrapperClass tempWrapObj =
        (TemperatureWrapperClass)system.JSON.deserialize(tempReading,TemperatureWra
pperClass.class);
        System.debug(tempWrapObj.humidity);
        System.debug(tempWrapObj.temp);
        System.debug(tempWrapObj.heatIndex);

        Double Humidity = Double.valueOf(tempWrapObj.humidity);
        System.debug('Humidity Value'+Humidity);
        Double Temperature = Double.valueOf(tempWrapObj.temp);
        System.debug('Temperature Value'+Temperature);
        Double heatIndex = Double.valueOf(tempWrapObj.heatIndex);
        System.debug('Heat Index Value'+heatIndex);

        Temperature_Data__c temperatureRecord = new Temperature_Data__c();
        temperatureRecord.Celcius__c = Temperature;
        temperatureRecord.Humidity__c = Humidity;
        temperatureRecord.Heat_Index__c = heatIndex;

        Double tempLimit = double.valueOf(Label.Max_Temp_Send_Email);//
custom label value

        Statistics__c statObj = new Statistics__c();
        if(Temperature>=tempLimit){
            temperatureRecord.Is_Critical__c = true;
            statObj.Last_Email_Send__c=Datetime.now();
            System.debug('StatObj:'+statObj);

            temperatureRecord.Critical_Temperature__c=Label.Max_Temp_Send_Email;
            responseBody+='Turn ON ';
        }
        try {
            System.debug(temperatureRecord);
            insert temperatureRecord;
            statObj.Related_Temperature_Record__c=temperatureRecord.Id;
            if(Temperature>=tempLimit){
                insert statObj;
            }
            responseBody+='Record inserted Successfully';
            RestContext.response.responseBody = Blob.valueOf(responseBody);
        } catch (Exception ex) {
            //If the insertion fails, send error response
            System.debug(ex.getCause()+''+ex.getLineNumber()+''
'+ex.getLineNumber());
            RestContext.response.responseBody = Blob.valueOf('error');
        }
    }
}
```

## References

- [1] <http://www.desynit.com/good-systems-blog/salesforce/why-salesforce/>
- [2] <http://www.dummies.com/business/customers/why-use-salesforce/>
- [3] <https://www.biz4intellia.com/blog/temperature-monitoring/>
- [4] [Extremely effective CRM Solution using Salesforce](#)
- [5] <https://en.wikipedia.org/wiki/Salesforce.com#Technologies>
- [6] <http://www.salesforce.com/salesforce-advantage/>
- [7] [https://resources.docs.salesforce.com/sfdc/pdf/salesforce\\_creating\\_on\\_demand\\_apps.pdf](https://resources.docs.salesforce.com/sfdc/pdf/salesforce_creating_on_demand_apps.pdf)
- [8] [https://en.wikipedia.org/wiki/Cloud\\_computing](https://en.wikipedia.org/wiki/Cloud_computing)
- [9] [http://csc.columbusstate.edu/eckart/classes/cpsc6125/topics/topic\\_74.shtml](http://csc.columbusstate.edu/eckart/classes/cpsc6125/topics/topic_74.shtml)
- [10] <http://intocloud.org/paas-platform-as-a-service/>