Project Report
On
“IoT Based Doctor Eye”

Submitted in partial fulfilment for the award of

Post Graduate Diploma in Internet of Things

(PG-DIOT) from C-DAC ACTS (Pune)

Guide by:

Mr. Bhupendra Pratap Singh

Presented by:

Ms. Nancy Rastogi	**Prn: 220340126017**
Ms. Pragati Gaikwad	**Prn: 220340126009**
Ms. Sakshi Janghel	**Prn: 220340126023**
Ms. Pratiksha Panchal	**Prn: 220340126019**

Centre of Development of Advanced Computing (C-DAC) ACTS, Pune

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Ms. Nancy Rastogi	**Prn: 220340126017**
Ms. Pragati Gaikwad	**Prn: 220340126009**
Ms. Sakshi Janghel	**Prn: 220340126023**
Ms. Pratiksha Panchal	**Prn: 220340126019**

have successfully completed their project
on

“IoT based Doctor Eye”

Under the guidance of

Project Guide

Mr. Bhupendra Pratap Singh

(Project Engineer, CDAC-ACTS)

ABSTRACT

The care of new born babies are the most important and sensitive part of bio-medical domain. Some new born babies have a higher risk of mortality due to their gestational age or their birth weight. Most of the premature babies born on 32-37 weeks of gestation and are deceased due to their unmet need for warmth.

Due to this patients require continuous monitoring of certain vital body parameters such as heart rate, pulse rate and electrocardiogram (ECG) showing current health status clearly, also Many premature babies have lost their lives due to lack of proper monitoring of the incubator that leads to accidents.

In this paper we are presenting 3 - tier architecture of our prototype healthcare monitoring system using wireless sensor network (WSN) which is developed to continuously monitor certain body parameters of patient. Different biosensors available to measure heart rate, body oxygen level and temperature are attached and recorded signals are sent to server using Node MCU ESP8266 and RasberryPi wireless communication. Data is made available on remote servers for doctors using DoctorEye application and they can monitor the patient from anywhere or anytime. In case of emergency doctor can be notified using smart phone alerts. The system is useful for patients and can be used for infant or baby care and elderly care in home and hospitals.

Thus, the objective of this project is to overcome the drawbacks of an unmonitored incubator and develops an affordable and safe device for real-time monitoring of the neonatal incubator. a low cost yet effective apparatus for monitoring the important parameters like pulse rate, temperature, humidity, Oxygen of the premature baby inside an incubator.

The sensed data are passed to the doctors or nurses wirelessly by the Raspberry PI via Internet of Things (IoT) so as to take necessary actions at times to maintain an appropriate environment for the safety of the lives of premature babies.

ACKNOWLEDGEMENT

This project “**IOT Based Doctor Eye**.” was a great learning experience for me and I am submitting this work to Advanced Computing Training School (CDAC ACTS).

I am very glad to mention the name of **Mr. Bhupendra Pratap Singh** for his valuable guidance to work on this project. His guidance and support helped me to overcome various obstacles and intricacies during the course of project work.

I am highly grateful to **Ms. Risha P.R.** (Joint Director (ACTS training Centre), CDAC, for her guidance and support whenever necessary while doing this course Post Graduate Diploma in Internet of Things (PG-DIOT) through C-DAC ACTS, Pune.

My heartfelt thanks goes to **Mrs. Seema Sanjeevan** (Course Coordinator, PG-DIOT) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC ACTS,

From:

Nancy Rastogi	(220340126017)
Pragati Gaikwad	(220340126009)
Sakshi Janghel	(220340126023)
Pratiksha Panchal	(220340126019)

Content

Acknowledgement

Contents

Chapter-1

Introduction

Chapter-2

Components

Chapter-3

Background

Chapter-4

Results

Chapter-5

Chapter 1

Introduction:

1.1 Doctor Eye Industrial Use Case:

In the industrial era 4.0, the main challenge of the product is the use of digital infrastructure where the internet of things (IoT) has been developed. IoT refers to physical devices, communication networks, medical equipment, and other items that can be connected using internet communication protocols. With the existing internet network, data will be collected and processed in a centralized server.

Internet of Things cases are chosen since the topic grows quicker than enactment. In the health sector, baby DOCTOR EYE producers are facing challenges from imported products that will have innovation with the application of IoT technology. For that, we need an internet-based baby incubator monitoring system that will encourage the creation of competitiveness. The purpose of this research DOCTOR EYE is to build a prototype of an internet-based baby incubator monitoring system based on things equipped with various sensors that will send data to the server in real-time and mobile apps for monitoring facilities for parents, including the Baby health Monitoring. This study focused on how to develop smart incubators.

So, this project or use-case aims to accomplish several key things:

- Easily collect data from incompatible sensors via MQTT Communication protocol.
- The neonatal incubator should be maintained at appropriate temperature and humidity levels. These two parameters can be sensed by using DHT11 sensor. The DHT11 Temperature and Humidity sensor features a temperature and humidity sensor complex with a calibrated digital signal output.
- A user-friendly Web interface that would include dashboards, allowing constant monitoring of Baby Temperature, Oxygen and provide provision for specified actions of control.

1.2 Purpose:

The proposed system is about monitoring the activities of baby remotely. It mainly consists of sensor, hardware unit, internet enabled single board computer called Raspberry Pi.

All the peripherals interfaced with RPi module are either connected wired or wirelessly. The RPi module shall be used in headless configuration, i.e., without keyboard, mouse or monitor connected to it. A separate keypad consisting of three buttons and one LED shall be interfaced with GPIO pins of RPi module.

Humidity and temperature information plays vital roles in baby health monitoring systems, for maintain incubator environment. For maintenance of any incubator there are some parameters which are Temperature, Oxygen, Humidity.

Real-time monitoring of the Temperature, Pulse and oxygen it also alerts if the particular value gets changed compared to idle working scenario. It also includes over-the-air updates if any unpredictable problem occurs. All the data is securely published over MQTT and used for further analysis of data and health of baby .

1.3 Architecture:

In below fig. 1.1 It shows the architecture of the project. The main device is Raspberry PI which is connected over Wi-Fi to Mosquitto broker which is held at AWS EC2, all the data received from the sensors is published on a topic then the user side system subscribed to topic and stored that data on MySQL which will further used for prediction as well it is live streamed to User Interface Dashboard to identify current situation of Incubator.

Doctors can monitor all these parameters of Patients in Angular Application Dashboard where doctor can monitor the health parameter of patients.

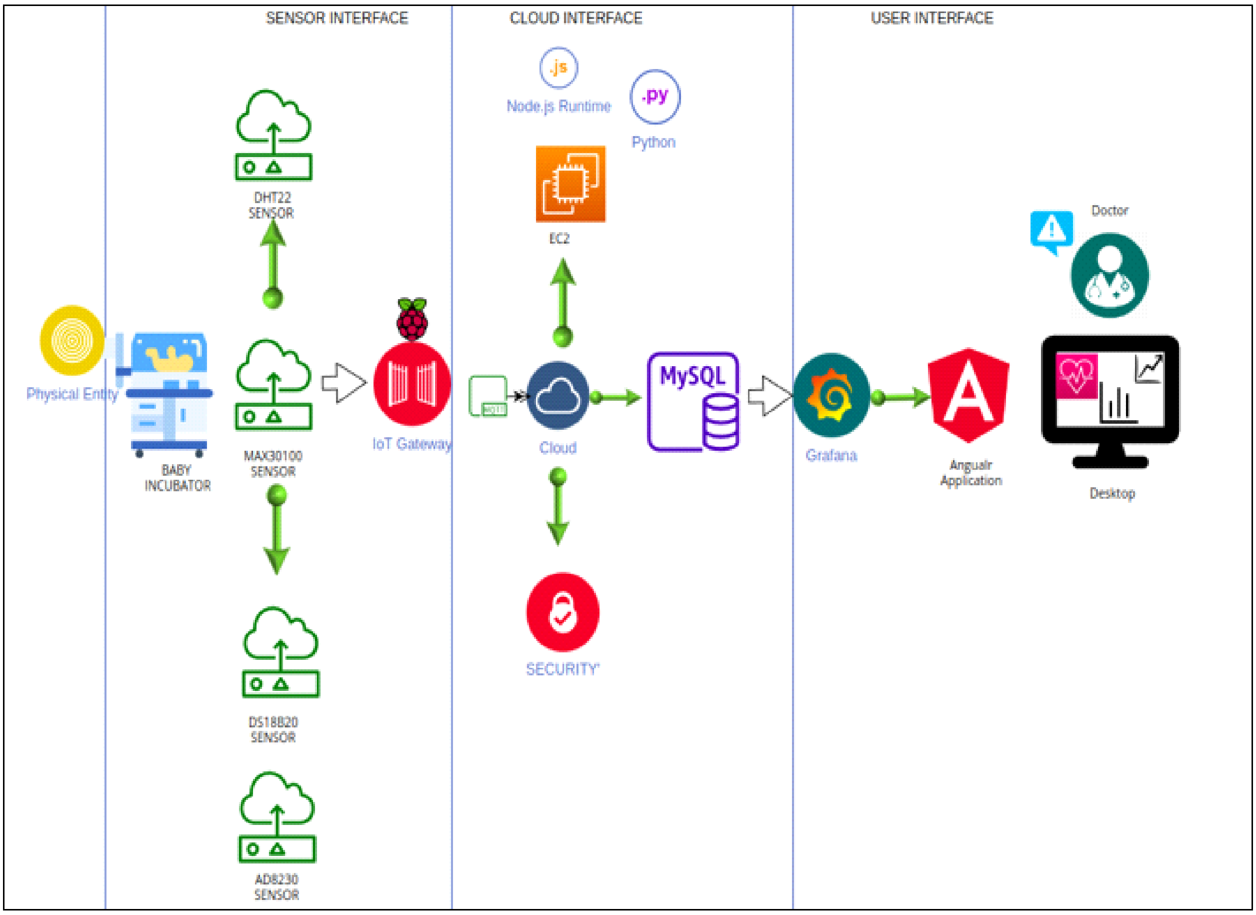


Fig.1.1 Architecture

Chapter 2

Implementation:

2.1 Schematic Diagram:

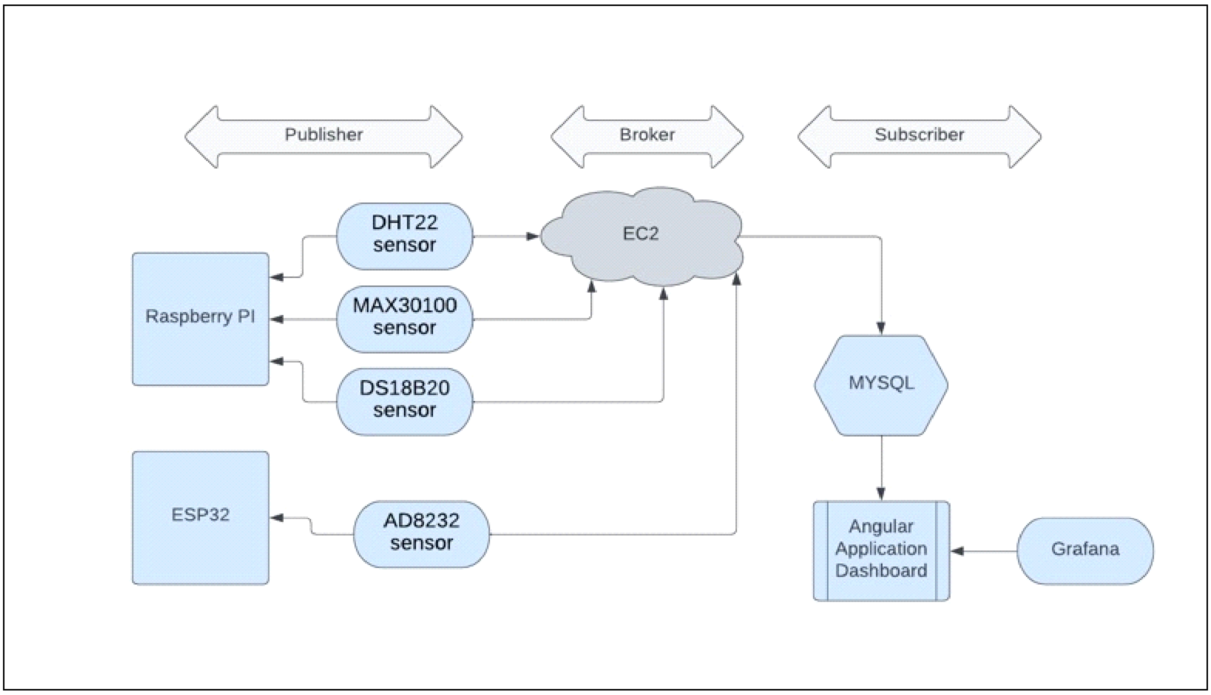


Fig. 2.1 Circuit Connection

The above fig. 3.1 shows the connection diagram of the proposed system. Raspberry PI collect the MAX30100 , DHT22 , and DS18B20 sensors and ESP32 collect the data of AD8232 ECG sensor data and send the sensor data to database .

Main components of the Circuit are :

- Raspberry PI
- ESP32 Board
- AD8232 Ecg sensor
- Max30100 Pulse Oximeter and Heart-Rate Sensor
- DHT22 Temperature sensor
- DS18B20 sensor

2.2 Sensors

2.2.1 AD8232 ECG sensor:

The AD8232 ECG sensor is a commercial board used to calculate the electrical movement of the human heart. This action can be chart like an Electrocardiogram and the output of this is an analog reading. Electrocardiograms can be very noisy, so to reduce the noise the AD8232 chip can be used. Electrocardiography is used to help diagnose various heart conditions.

The working principle of the ECG sensor is like an operational amplifier to help in getting a clear signal from the intervals simply.

The main purpose of this chip is to amplify, extract as well as filter biopotential signals which are small in the noisy conditions like those formed through the replacement of remote electrode as well as motion.

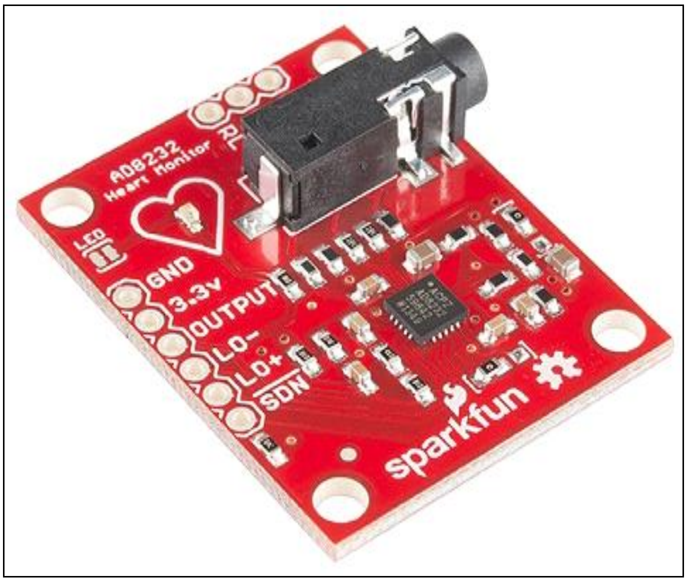


Fig.2.2.1 AD8232-ECG-sensor

The AD8232 sensor is used for signal conditioning in ECG as well as other measurement applications of biopotential. The main purpose of this chip is to amplify, extract as well as filter biopotential signals which are small in the noisy conditions like those formed through the replacement of remote electrode as well as motion.

AD8232 Pin Configuration:

The heart rate monitoring sensor like AD8232 includes the pins like SDN pin, LO+ pin, LO- pin, OUTPUT pin, 3.3V pin, and GND pin. So that we can connect this IC to development boards like Arduino by soldering pins.

Additionally, this board includes pins like the right arm (RA), left arm (LA) & right leg (RL) pins to connect custom sensors. An LED indicator in this board is used to indicate the heartbeat rhythm of humans.

The AD8232 sensor comprises a function like quick restore, used to decrease the length of long resolving tails of the HPFs. This sensor is accessible in a 4 mm ×

4 mm size, and the package of this sensor is 20-lead LFCSP. It operates from −40°C -to- +85°C but the performance is specified from 0°C -to- 70°C.

Specifications:

Operation of single supply ranges	2V to 3.5V
CMRR	80db
Electrode configurations	2 or 3
half cell potential	±300 mV
current supply	170 µA
size	4 mm × 4 mm
high-pass filter	Adjustable 2-pole
low-pass filter	Adjustable 3-pole

2.2.2 DHT22 Sensor:

The **DHT22** is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from **-40°C to 80°C** and humidity from **0% to 100%** with an accuracy of ±1°C and ±1.

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its **sensing elements** is connected with 8-bit single-chip computer.

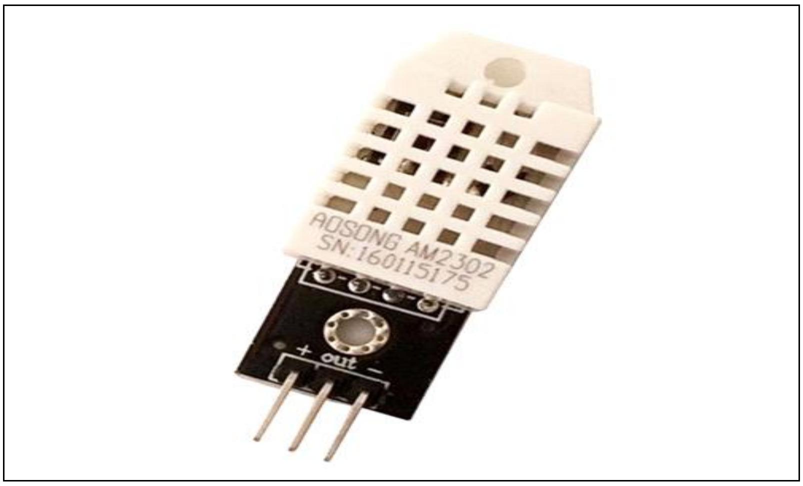


Fig.2.2.2 DHT22 sensor

Every sensor of this model is temperature compensated and calibrated in accurate **calibration chamber** and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long **transmission distance** (20m) enables **DHT22** to be suited in all kinds of harsh **application occasions**.

Specifications:

Operating Voltage	3.5V to 5.5V
Operating current	0.3mA (measuring) 60uA (standby)
Output	Serial data
Temperature Range	-40°C to 80°C
Humidity Range	0% to 100%
Resolution	Temperature and Humidity both are 16-bit
Accuracy	±0.5°C and ±1%

2.2.3 MAX30100 Sensor:

MAX30100 sensor is a device that is used to monitor the heart rate and it is also used as a pulse oximeter. The Pulse oximeter consists of Light-emitting diodes and an IR sensor. And signal processing unit to improve the quality of the output signal.

It works on the input voltage of 1.8V to 3.3V and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times.

The MAX30100 pulse oximeter can measure both the heart pulse rate and the oxygen level in the blood. The red light that the red LED emits is used to measure the pulse rate.

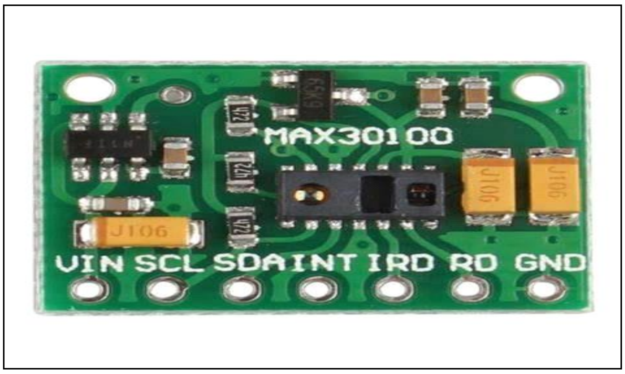


Fig.2.2.3 MAX30100 sensor

Specification:

Min Supply Voltage	1.7v
Number of Pins	14
Min Operating Temperature	-40°C
Number of Pins	14
Moisture Sensitivity Level (MSL)	3 (168Hours)
Max Supply Voltage	2V
Sensor Type	Oximeter/Heart Rate

2.2.4 DS18B20 Sensor:

The DS18B20 temperature sensor is a one-wire digital temperature sensor. This means that it just requires one data line (and GND) to communicate.

It can be powered by an external power supply or it can derive power from the data line (called "parasite mode"), which eliminates the need for an external power supply.

Specification:

Number of Pins	3
Temperature Range	-55°C to +125°C
Power supply range	3 V to 5 V
Accuracy	+/-0.5 ºC (between the range -10ºC to 85ºC)
Sensor Type	Body Temperature Sensor

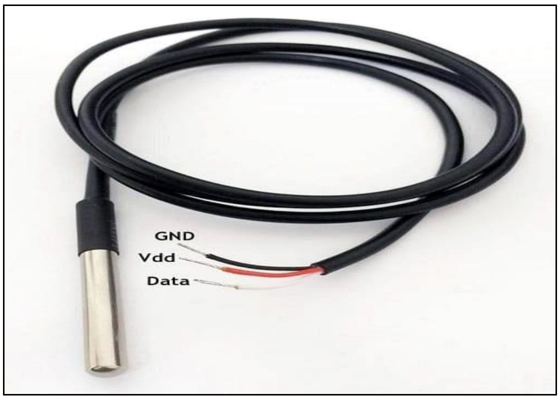


Fig.2.2.4 DS18B20 sensor

2.3 Hardware Components:

2.3.1 RASPBERRY PI:

Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

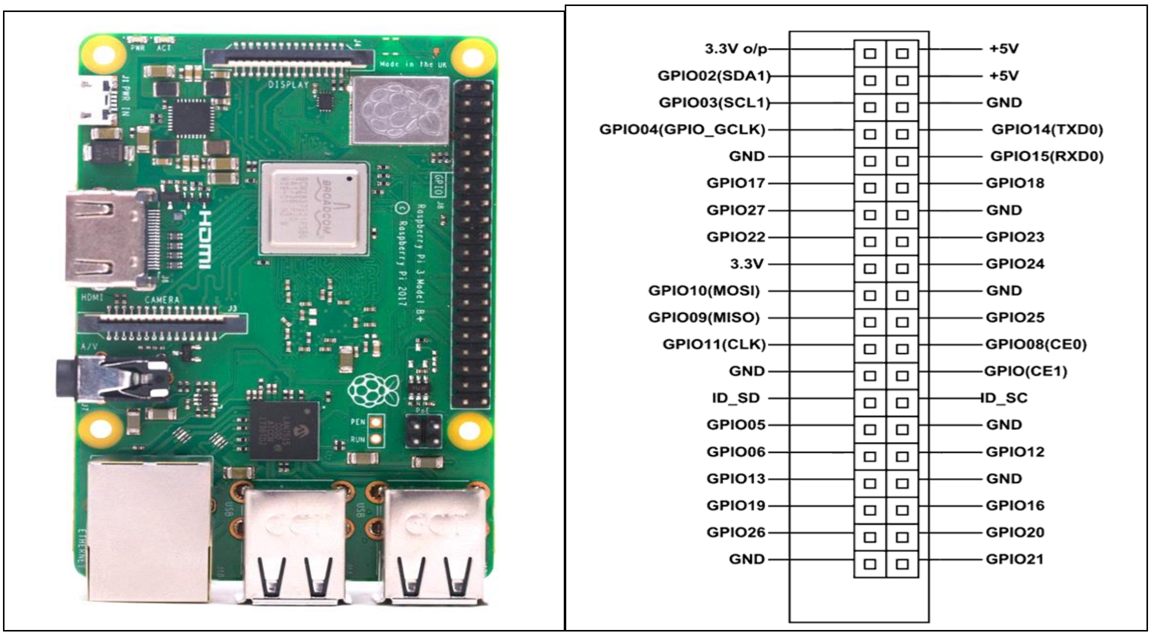


Fig.2.3.1 Raspberry PI

Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras.

As mentioned earlier PI is simply a **COMPUTER ON A SINGLE BOARD** so it cannot be used like ARDUINO development boards. For the PI to start working we need to first install OPERATING SYSTEM. This feature is similar to our PC. The PI has dedicated OS for it; any other OS will not work.

We will discuss the programming of PI in step by step below.

- Take the 16GB micro SD card and dedicate it specifically for PI OS.
- Choose and Download OS software. [https://www.raspberrypi.org/downloads/] ('NOOBS' recommended for beginners)
- Format the SD card and install OS on to the SD memory card using convenient methods.
- Take the SD card after OS installation and insert it in PI board.
- Connect monitor, keyboard and mouse
- Power the board with micro USB connector
- Once the power is tuned ON the PI will run on the OS installed in the memory card and will start from boot.
- Once all drivers are checked the PI will ask for authorization, this is set by default and can be changed.
- After authorization you will reach desktop where all application program development starts.

On the PI you can download application programs required for your use and can directly install as you do for your PC. After that you can work on developing required program and get the PI run the developed programs.

Specification:

SoC	Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
GPU	Broadcom Videocore-IV
RAM	1GB LPDDR2 SDRAM
Networking	Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
Bluetooth	Bluetooth 4.2, Bluetooth Low Energy (BLE)

Storage	Micro-SD
Dimension	82mm x 56mm x 19.5mm, 50g
GPIO	40-pin GPIO header, populated
Ports	HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

2.3.2 ESP32 Board :

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI / SDIO or I2C / UART interfaces.

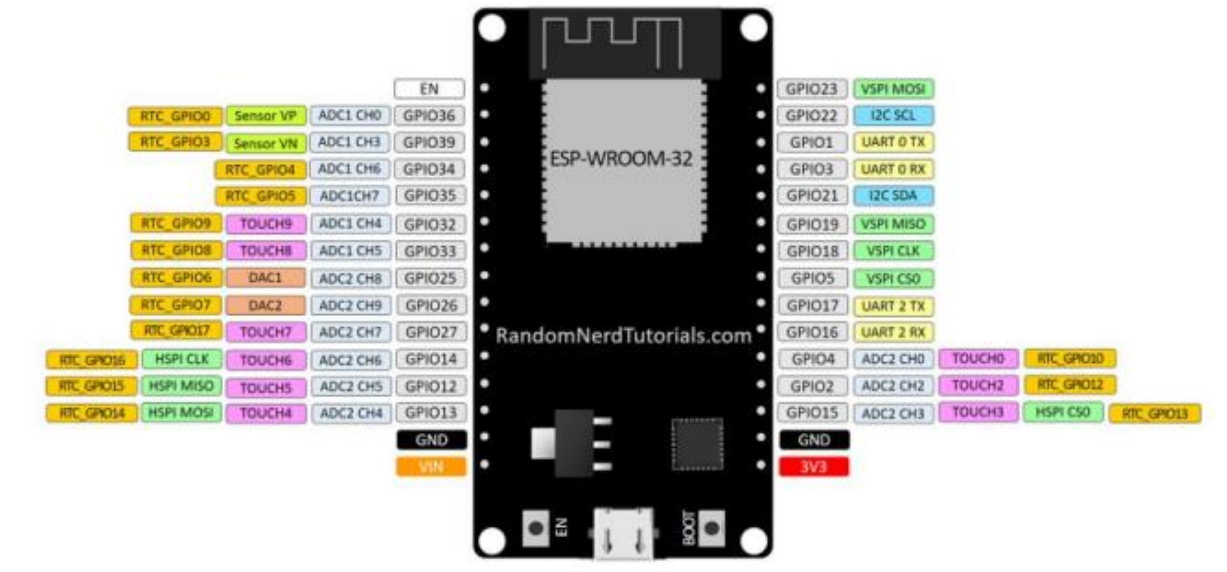


Fig.2.3.2 ESP32

Features of the ESP32 include the following :

- Processor:

  CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS Ultra low power (ULP) co-processor.

- Memory: 320 KiB RAM, 448 KiB ROM

- Wireless connectivity:

    Wi-Fi: 802.11 b/g/n

    Bluetooth: v4.2 BR/EDR and BLE (shares the radio with Wi-Fi)

- Peripheral interfaces:

    34 × programmable GPIOs

    12-bit SAR ADC up to 18 channels

    2 × 8-bit DACs

    10 × touch sensors (capacitive sensing GPIOs)

    4 × SPI

    2 × I²S interfaces

    2 × I²C interfaces

    3 × UART

    SD/SDIO/CE-ATA/MMC/eMMC host controller

    SDIO/SPI slave controller

    Ethernet MAC interface with dedicated DMA and planned IEEE 1588 Precision Time Protocol support[4]

    CAN bus 2.0

    Infrared remote controller (TX/RX, up to 8 channels)

    Motor PWM

    LED PWM (up to 16 channels)

    Hall effect sensor

    Ultralow power analog pre-amplifier

- Security:

    IEEE 802.11 standard security features all supported, including WPA, WPA2, WPA3 (depending on version)[5] and WLAN Authentication and Privacy Infrastructure (WAPI)

    Secure boot

    Flash encryption

    1024-bit OTP, up to 768-bit for customers

    Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)

- Power management:

    Internal low-dropout regulator

    Individual power domain for RTC

    5 μA deep sleep current

Wake up from GPIO interrupt, timer, ADC measurements, capacitive touch sensor interrupt

2.3.3 Jumper Wire:

A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them –simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

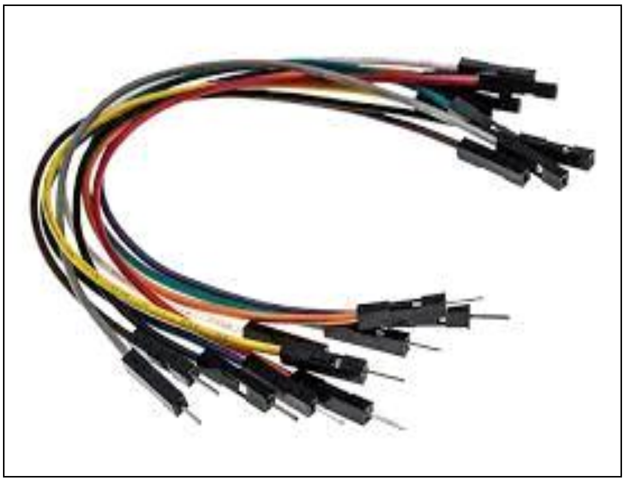


Fig 2.3.3 Jumper Wire

2.4 Software Components

2.4.1 User Interface (Angular):

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.

The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks of the Angular framework are Angular components that are organized into NgModules. NgModules collect related code into functional sets; an Angular application is defined by a set of NgModules. An application always has at least a root module that enables bootstrapping, and typically has many more feature modules.

Connecting Angular Application To database through Node.js Express :

- Node.js Express exports REST APIs & interacts with MySQL Database using Sequelize ORM.
- Angular 12 Client sends HTTP Requests and retrieves HTTP Responses using HTTP Client, consume data on the components. Angular Router is used for navigating to pages.

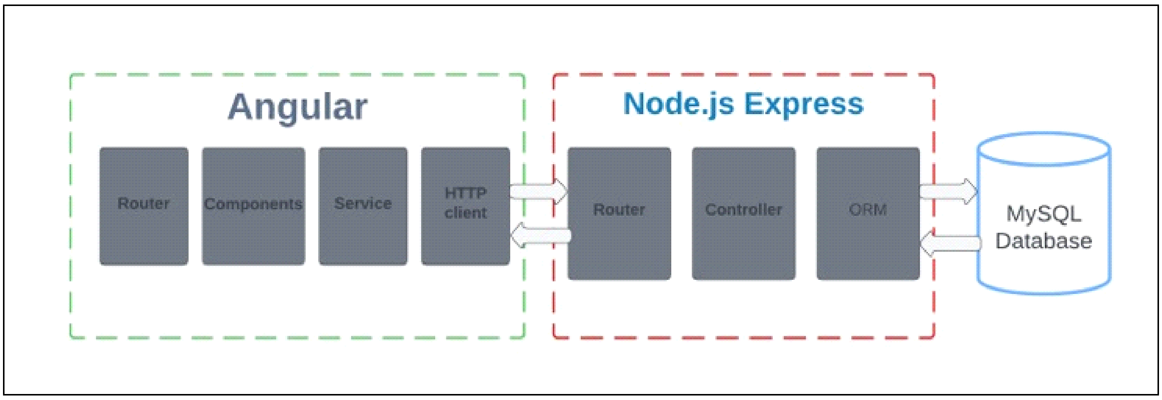
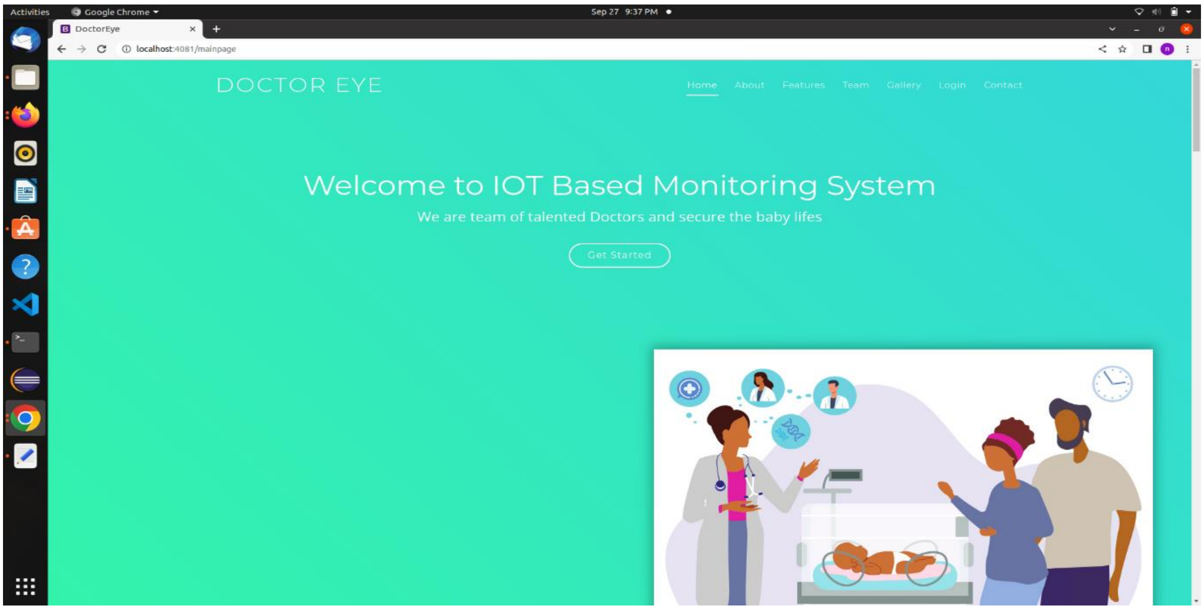


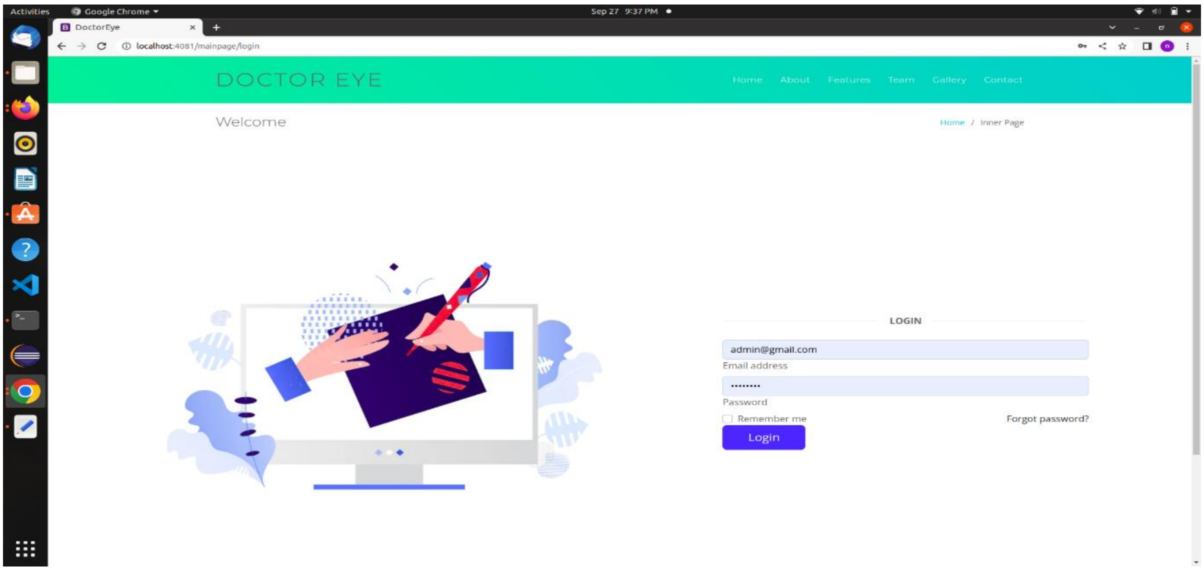
Fig.2.4.1 User-Interface

This is the user interface of angular app using which doctor can monitor the health parameter of Patients. Here this application is designed in angular so that doctor can easily monitor Patient and admin to easily add, remove, update and delete the doctor and patients details.



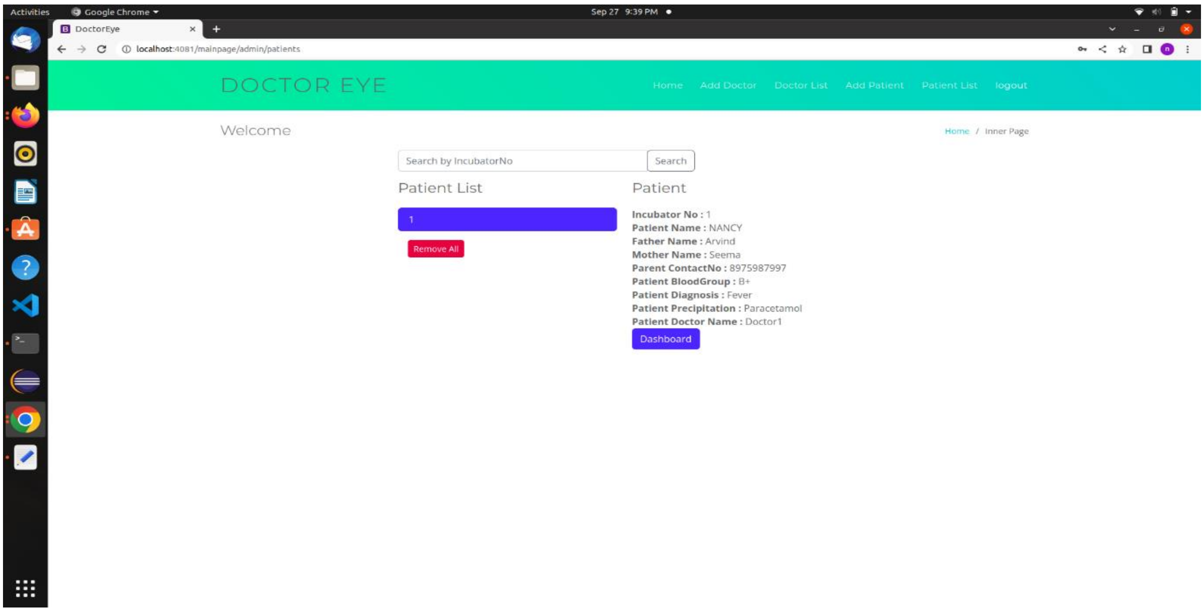
Welcome to IOT Based Monitoring System

We are team of talented Doctors and secure the baby lifes



Email address: admin@gmail.com

Password: ••••••••



Patient List

1

Patient

Incubator No : 1
Patient Name : NANCY
Father Name : Arvind
Mother Name : Seema
Parent ContactNo : 8975987997
Patient BloodGroup : B+
Patient Diagnosis : Fever
Patient Precipitation : Paracetamol
Patient Doctor Name : Doctor1

2.4.2 Node Js:

JavaScript is a light-weight object-oriented programming language that is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language. JavaScript enables dynamic interactivity on websites when it is applied to an HTML document.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

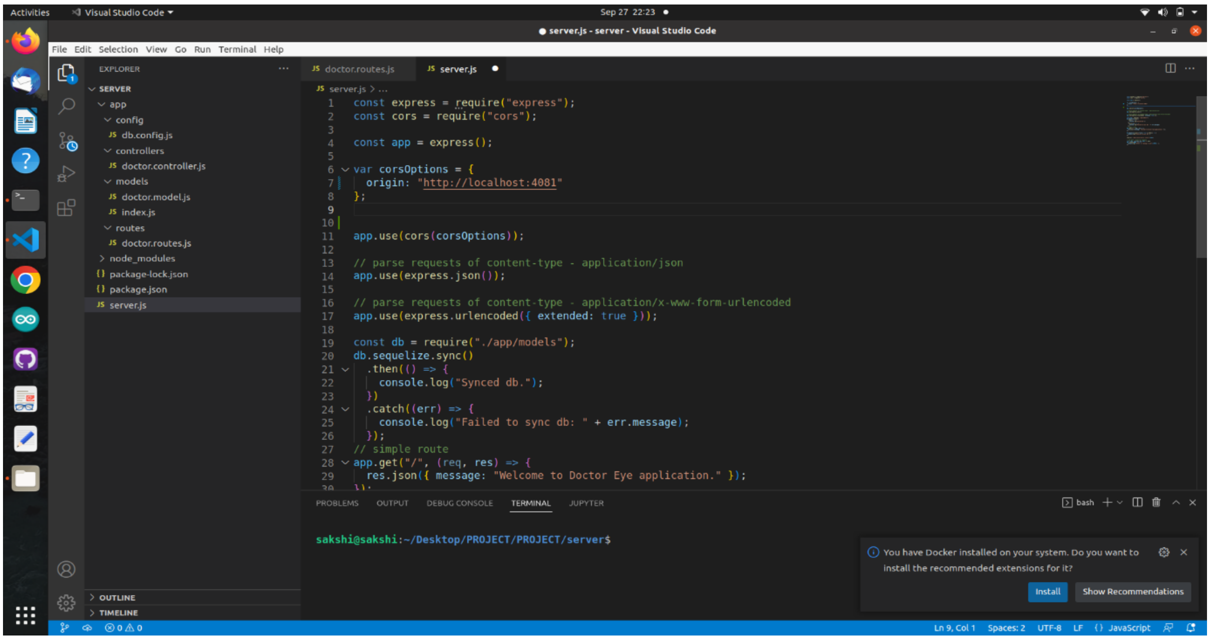
What is NPM?

NPM is a package manager for Node.js packages, or modules. The NPM program is installed on your computer when you install Node.js.

2.4.3 Express JS:

Express is a minimal and flexible Node.js web application framework that provide a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. It is a web application framework for Node.js, released as free and open-source software. Express.js basically helps you manage everything, from routes, to handling requests and views.

Express API: - An API is always needed to create mobile applications, single page applications. A popular architectural style of how to structure and name these APIs and the endpoints is called REST (Representational Transfer State).



```
const express = require("express");
const cors = require("cors");

const app = express();

var corsOptions = {
  origin: "http://localhost:4081"
};


app.use(cors(corsOptions));

// parse requests of content-type - application/json
app.use(express.json());

// parse requests of content-type - application/x-www-form-urlencoded
app.use(express.urlencoded({ extended: true }));

const db = require("./app/models");
db.sequelize.sync()
  .then(() => {
    console.log("Synced db.");
  })
  .catch((err) => {
    console.log("Failed to sync db: " + err.message);
  });
// simple route
app.get("/", (req, res) => {
  res.json({ message: "Welcome to Doctor Eye application." });
```

```
sakshi@sakshi:~/Desktop/PROJECT/PROJECT/server$
```

You have Docker installed on your system. Do you want to install the recommended extensions for it?

2.4.4 Arduino Software (IDE):

The Arduino Integrated Development Environment - or Arduino Software (IDE) contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

Inexpensive

Cross-platform

Simple, clear programming environment

Open source and extensible software

Open source and extensible hardware

2.4.5 AWS CLOUD:

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 offers the broadest and deepest compute platform with choice of processor, storage, networking, operating system, and purchase model.

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

2.4.5 Data Visualization:

Grafana is opensource visualization and analytics software. It allows you to query, visualize, alert on, and explore your metrics no matter where they are stored. In plain English, it provides you with tools to turn your time-series database (TSDB) data into beautiful graphs and visualizations.

Grafana dashboards are excellent tools for gaining insight into time-series data. In addition, Grafana readily integrates with InfluxDB and Telegraf to make monitoring of sensor, system and network metrics much easier and far more insightful.

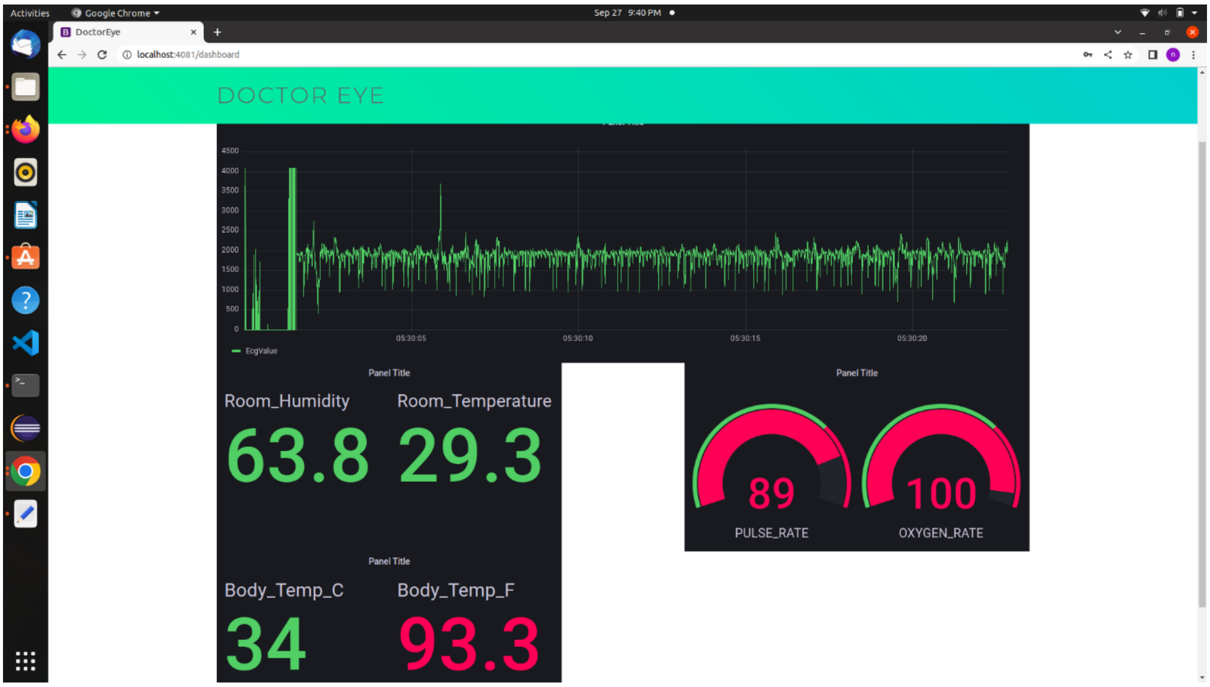
The process of setting up a Grafana dashboard and integrating it with various data sources is straightforward. Grafana template variables enable you to create dynamic dashboards that you can make changes to in real-time.

Getting started with Grafana:
Step 1: Install Grafana
Step2: Log in
To log in to Grafana for the first time:

1. Open your web browser and go to http://localhost:3000/. The default HTTP port that Grafana listens to is 3000 unless you have configured a different port.
2. On the login page, enter admin for username and password.
3. Click Log In. If login is successful, then you will see a prompt to change the password.
4. Click OK on the prompt, then change your password.

Step 3: Create a dashboard
To create dashboard: 23

1. Click the + icon on the left panel, select Create Dashboard, and then click Add new panel.
2. In the New Dashboard/Edit Panel view, go to the Query tab.
3. Configure your query by selecting -- Grafana -- from the data source selector. This generates the Random Walk dashboard.
4. Click the Save icon in the top right corner of your screen to save the dashboard.
5. Add a descriptive name, and then click Save.



Room_Humidity 63.8

Room_Temperature 29.3

PULSE_RATE 89

OXYGEN_RATE 100

Body_Temp_C 34

Body_Temp_F 93.3

Chapter-3

Background:

3.1 Data Storage:

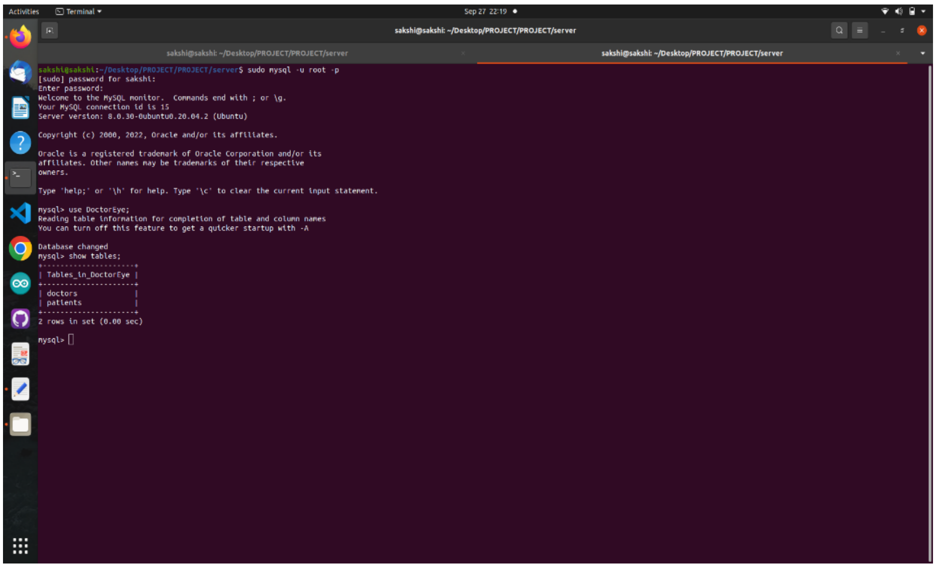
In this project we are using mysql atlas for data storage. MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications.

The most common use for mySQL however, is for the purpose of a web database. It can be used to store anything from a single record of information to an entire inventory of available products for an online store.

This server is used for data operations like querying, filtering, sorting, grouping, modifying, and joining the tables present in the database.

Some of the common features of MySQL are:

- It is easy to use and free of cost to download.
- It contains a solid data security layer to protect important data.
- It is based on client and server architecture.
- It supports multithreading which makes it more scalable.
- It supports multithreading which makes it more scalable.
- The performance of MySQL is fast, efficient, and reliable.


Activities
Terminal
Sep 27 22:19
sakshi@sakshi: ~/Desktop/PROJECT/PROJECT/server
sakshi@sakshi: ~/Desktop/PROJECT/PROJECT/server
sakshi@sakshi: ~/Desktop/PROJECT/PROJECT/server

```
sakshi@sakshi:~/Desktop/PROJECT/PROJECT/server$ sudo mysql -u root -p
[sudo] password for sakshi:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.30-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use DoctorEye;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+--------------------+
| Tables_in_DoctorEye |
+--------------------+
| doctors            |
| patients           |
+--------------------+
2 rows in set (0.00 sec)

mysql> 
```

- It is compatible with many operating systems like Windows, macOS, Linux, etc.

As above using simple database connectivity we can easily connect data base with our front end. And this way user data is stored safely and retrieve easily whenever required for the purpose of processing.

In our project all patient personal information and doctor details will get stored in the data base. And using this stored information authentication is provided while login process.

Also all the data received from heart rate sensor , Room temperature and humidity sensor , pulse oximeter sensor and body temperature sensor is stored and used to check users health condition.

3.2 Communication protocol:

The MQTT protocol was invented in 1999 by Andy Stanford-Clark (IBM) and Arlen Nipper. They needed a protocol for minimal battery loss and minimal bandwidth to connect with oil pipelines via satellite. The acronym for MQTT is MQ telemetry transport or simply MQTT protocol.

MQTT would be standardized under the wings of OASIS [The Organization for the Advancement of Structured Information Standards (OASIS)]. 2014 MQTT became an officially approved OASIS standard. OASIS ratified the new MQTT 5 specification. This new MQTT version introduced new features to MQTT that are required for IoT applications deployed on cloud platforms, and those that require more reliability and error handling to implement mission-critical messaging.

MQTT is a Client-Server publish/subscribe messaging transport protocol. It is an application layer protocol built on TCP protocol. It is lightweight, open, simple, and designed so as to be easy to implement. It is a very lightweight and binary protocol, and due to its minimal packet overhead, MQTT excels when transferring data over the wire in comparison to protocols like HTTP. Another important aspect of the protocol is that MQTT is extremely easy to implement on the client-side. Ease of use was a key concern in the development of MQTT and makes it a perfect fit for constrained devices with limited resources today.

Publish-Subscribe Pattern:

The publish/subscribe pattern (also known as pub/sub) provides an alternative to a traditional client-server architecture. In the client-server model, a client communicates directly with an endpoint. The pub/sub model decouples(separates) the client that sends a message (the publisher) from the client or clients that receive the messages (the subscribers). 5 The publishers and subscribers never contact each other directly. In fact, they are not even aware that the other exists. The connection between them is handled by a third component (the broker). The job of the broker is to filter all incoming messages and distribute them correctly to subscribers.

The most important aspect of pub/sub is the decoupling of the publisher of the message from the recipient (subscriber). Pub/Sub scales better than the traditional client-server approach. This is because operations on the broker can be highly parallelized and messages can be processed in an event-driven way.

3.2.2 Client, Broker / Server, and Connection Establishment:

Client:

- Both publishers and subscribers are MQTT clients.
- An MQTT client is any device (from a microcontroller up to a full-fledged server) that runs an MQTT library and connects to an MQTT broker over a network.
- Basically, any device that speaks MQTT over a TCP/IP stack can be called an MQTT client.

Broker:

- The broker is at the heart of any publish/subscribe protocol.
- Depending on the implementation, a broker can handle up to thousands of concurrently connected MQTT clients.
- The broker is responsible for receiving all messages, filtering the messages, determining who is subscribed to each message, and sending the message to these subscribed clients.
- The broker also holds the session data of all clients that have persistent sessions, including subscriptions and missed messages.
- Another responsibility of the broker is the authentication and authorization of clients.
- Usually, the broker is extensible, which facilitates custom authentication, authorization, and integration into backend systems.
-

MQTT Connection:

The MQTT protocol is based on TCP/IP. Both the client and the broker need to have a TCP/IP stack. The MQTT connection is always between one client and the broker. Clients never connect to each other directly. To initiate a connection, the client sends a CONNECT message to the broker. The broker responds with a CONNACK message and a status code. Once the connection is established, the broker keeps it open until the client sends a disconnect command or the connection breaks.

Publish:

- An MQTT client can publish messages as soon as it connects to a broker. MQTT utilizes the topic-based filtering of the messages on the broker.
- Each message must contain a topic that the broker can use to forward the message to interested clients.
- Typically, each message has a payload that contains the data to transmit in byte format.

Subscribe:

Publishing a message doesn’t make sense if no one ever receives it. In other words, if there are no clients to subscribe to the topics of the messages. To receive messages on topics of interest, the client sends a SUBSCRIBE message to the MQTT broker.

MQTT Over TLS:

Transport Layer Security (TLS) and Secure Sockets Layer (SSL) provide a secure communication channel between a client and a server. At the core, TLS and SSL are cryptographic protocols which use a handshake mechanism to negotiate various parameters to create a secure connection between the client and the server. After the handshake is complete, an encrypted communication between client and server is established and no attacker can eavesdrop any part of the communication. Servers provide a X509 certificate (typically issued by a trusted authority) that clients use to verify the identity of the server.

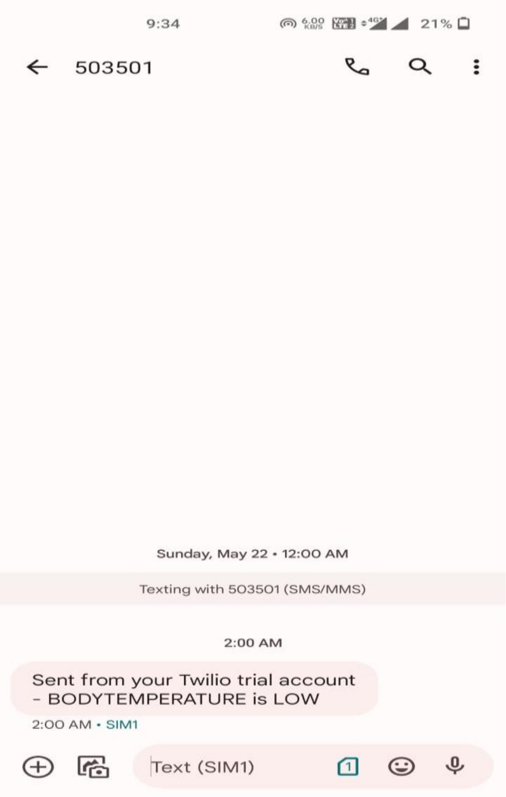
MQTT relies on the TCP transport protocol. By default, TCP connections do not use an encrypted communication. To encrypt the whole MQTT communication, many MQTT brokers allow use of TLS instead of plain TCP. If you use the username and password fields of the MQTT CONNECT packet for authentication and authorization mechanisms, you should strongly consider using TLS.

Port 8883 is standardized for a secured MQTT connection. The standardized name at IANA is “secure-mqtt”. Port 8883 is exclusively reserved for MQTT over TLS.

3.3 Twilio :

Twilio is a communications platform that gives developers access to SMS, MMS, voice, and other messaging capabilities through simple HTTP requests.

Twilio offers a real-time API for sending messages in any language, from anywhere in the world. Developers can send text messages, make phone calls, record voicemails, and even browse the internet.



Sunday, May 22 • 12:00 AM

Texting with 503501 (SMS/MMS)

2:00 AM

Sent from your Twilio trial account - BODYTEMPERATURE is LOW

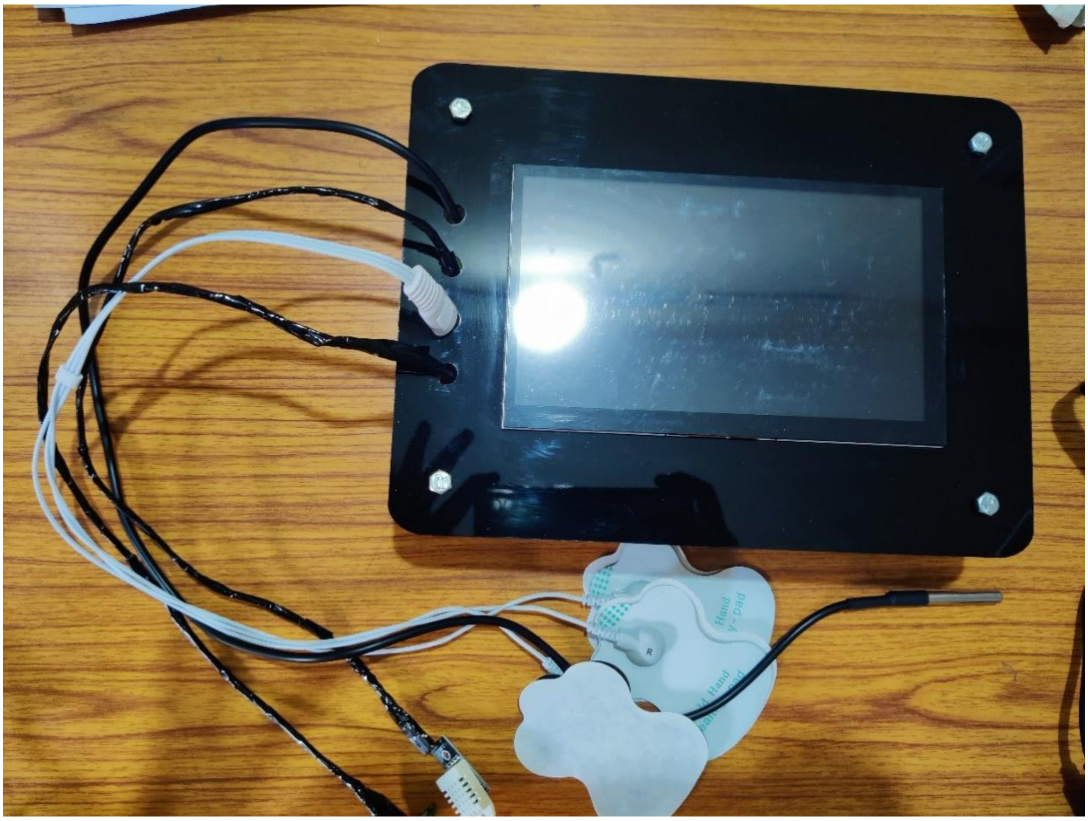
2:00 AM • SIM1

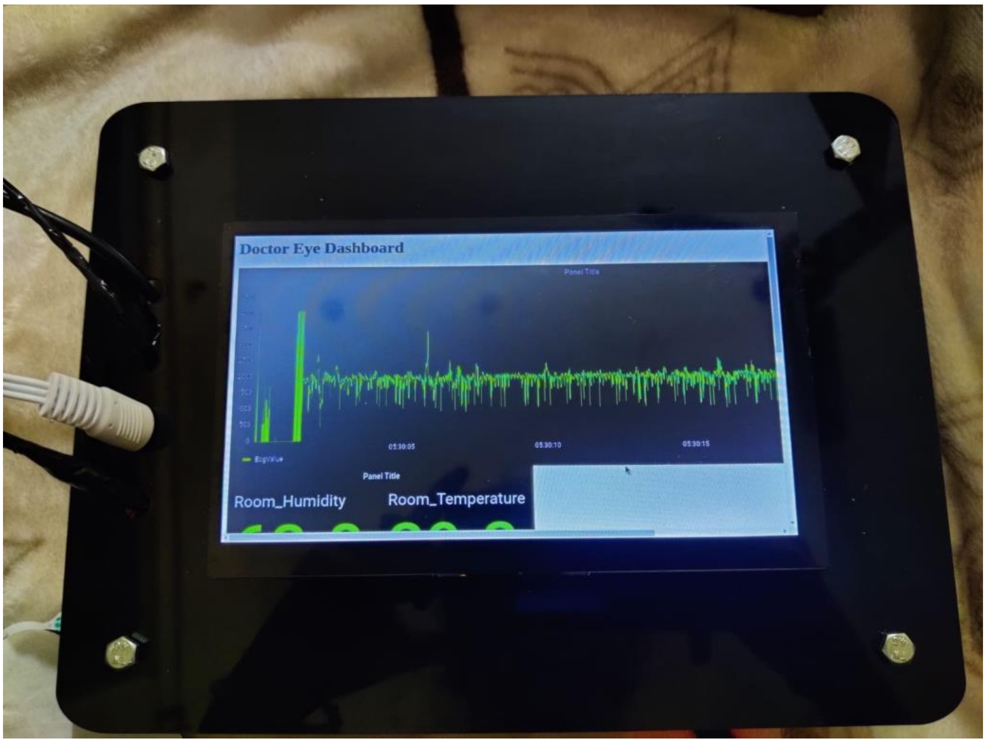
Fig.3.2 Twilio Message

3.4 Work Description:

Fig shows a detailed diagram of the project. All sensors' data is taken with the help of Raspberry PI. Collected data is published with the help of MQTT protocol. Here the proposed system is using AWS EC2 as a broker. Host machine is used as a subscriber which will collect the sensor data and store it on the MySQL database.

All sensor data is shown on Grafana. On Grafana Temperature, Humidity, Heartbeat and Oxygen
Are shown.

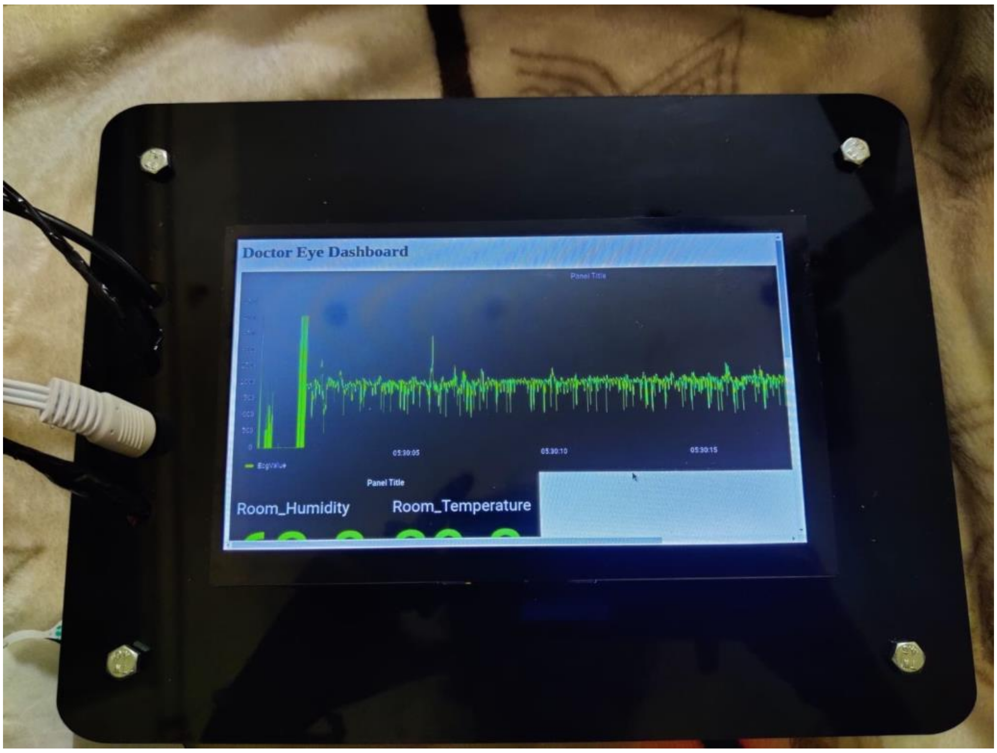



Doctor Eye Dashboard
Panel Title
0
05:30:05
05:30:10
05:30:15
EcgValue
Panel Title
Room_Humidity
Room_Temperature

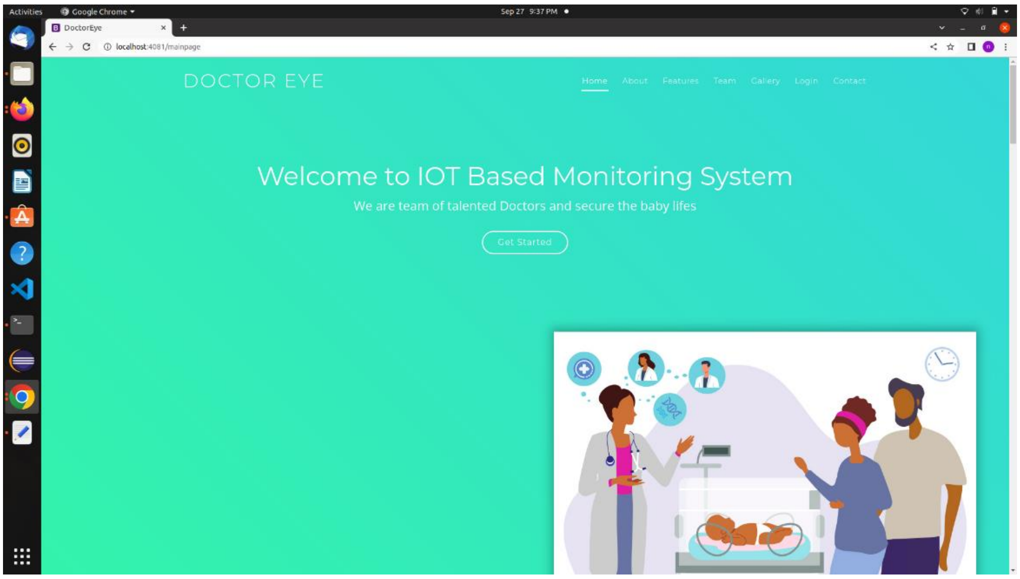
Chapter-4

Results

4.1 Output of Grafana Dashboard



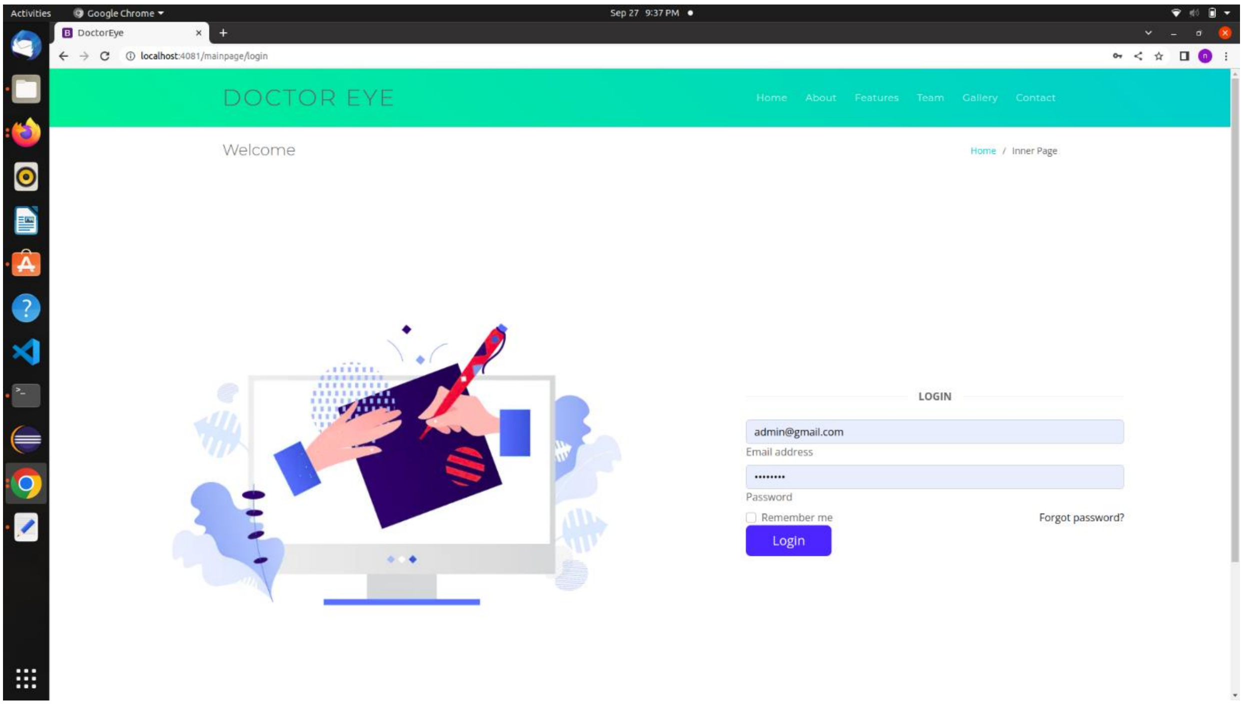
4.2 Output of User Interface (Angular Application) :



DOCTOR EYE

Welcome to IOT Based Monitoring System

We are team of talented Doctors and secure the baby lifes



Welcome

LOGIN

admin@gmail.com

Email address

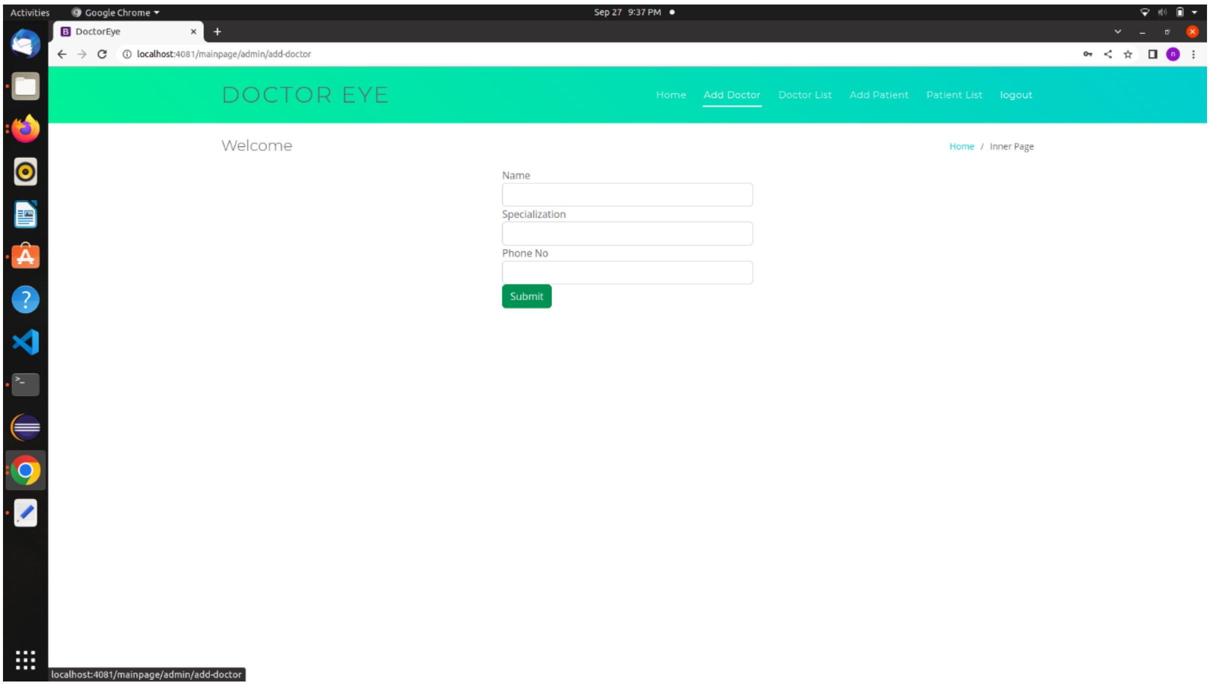
••••••••

Password

□ Remember me

Forgot password?

Login



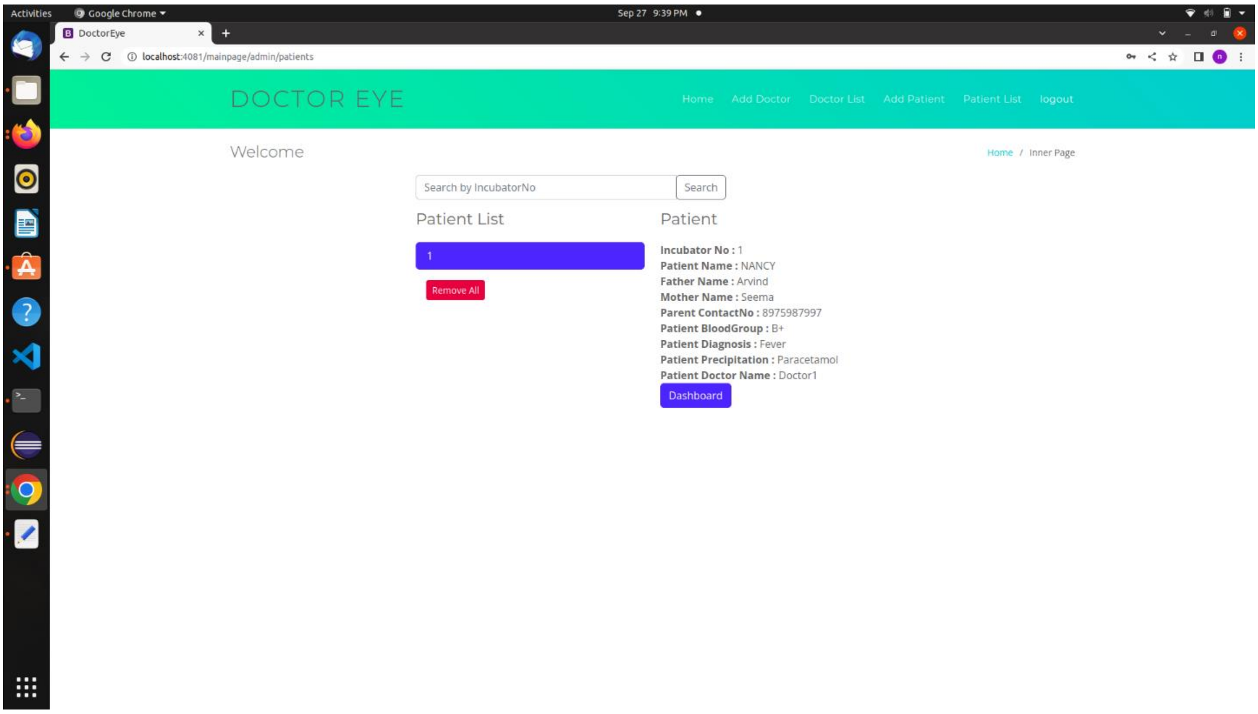
Welcome

Name

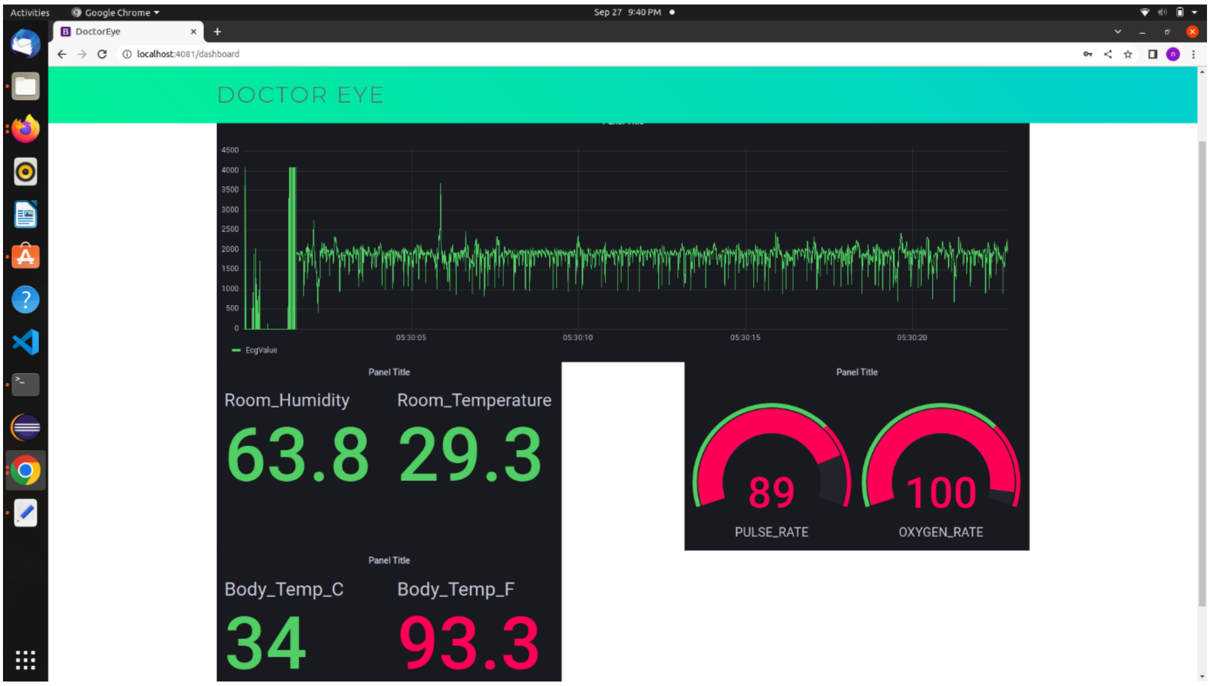
Specialization

Phone No

Submit



Incubator No : 1
Patient Name : NANCY
Father Name : Arvind
Mother Name : Seema
Parent ContactNo : 8975987997
Patient BloodGroup : B+
Patient Diagnosis : Fever
Patient Precipitation : Paracetamol
Patient Doctor Name : Doctor1



Chapter-5

Conclusion

5.1 Conclusions:

IoT makes everyday objects ‘smart’ by enabling them to transmit data and automate tasks, without requiring any manual intervention. An IoT device could be something as simple as a health monitoring system, or as complex as a smart city with sensors across all its regions.

The Internet of Things offers numerous benefits to organizations. Most modern enterprises are already leveraging IoT to automate and simplify many of their daily tasks. Connected devices are being integrated into existing and evolving business processes. The industrial Internet of Things focuses on automation - not just to lower operating costs but also to increase productivity, improve business processes, enhance customer experience, expand to new markets, and generate additional revenue streams. Industrial IoT is all about being well-informed to make timely and better decisions and keep you safe and informed about current situation.

Through this project we try to of implement IoT based remote health monitoring systems. As technology makes our life faster and easier. But this comfort comes with lots of responsibilities. The compact sensors with IoT will make a huge impact on every patient’s life, that even though they are away from home and physician, this helps them to reduce the fear of danger. The sensory data can be acquired in home or work environments. Also, the challenges in sensing, analytics and prediction of the disease are also highlighted and those can be addressed to provide a seamless integraton into the medical field

Remote monitoring System will continuously monitor patient heart beat rate , ambient temperature and humidity , pulse oximeter and body temperature , and doctor will monitoring the health of the patient from anytime anywhere. This way along with health monitoring it will give information of patient health.

5.2 Future Scope

We have Done this project keeping baby incubator system in mind. But if see the whole structure of the project, we can also use it to monitor the patient who have any disease which continuous monitoring, old aged people, as well as we can used this device for regular check up of patient as our device monitor patient body temperature, heart rate, pulse and oximeter rate and ambient temperature and humidity, which doctor can continuous monitor from anytime and anywhere.

We have analysed and done survey in panchawati area and what we have we found is many old aged people live alone as their children live away from them for work or for studies. So, from them our device is very useful because their children can monitor the health of the patient from the anywhere and the children can consult the doctor regarding their parents' health to the doctor.