

COVID ROOM MONITOR

By Stephen O'Malley



What is the Covid Room Monitor?

- The Covid Room Monitor is a project that I design for my FYP.
 - The Covid Room Monitor , Monitors a room to check it for the conditions that may cause the increased spread of Covid-19.
 - It does this by monitoring the CO2 levels in a room which have been linked to an increased risk of Covid propagation.
 - It monitors these reading using:
 - ESP-32 to gather the information .*
 - Webserver to display said information.*
-

ESP-32

- The ESP-32 is Embedded hardware that gathers the room monitoring data.
 - Data is collected by the ESP-32 using an MQ-135 CO2 Sensor that can measure the CO2 in PPM in a room.
 - This Data is then send via the AWS IoT Core MQTT to a the Full-Stack Webserver to store and display the information.
-

Full-Stack Web Server: Back

- NodeJS was used as the Back End of my Web Server, it handles all communication between the Front End and the database along with entering data into said database.
 - NodeJS subscribes to the MQTT feed on the AWS IoT Core and enters that data in the MongoDB database using Mongoose.
 - NodeJS also handles the http routing requests from the Front End which is basically the Front end asking to Send or Receive information to and from the Node Server.
 - As mentioned, we also have a database called MongoDB which is an open-source NoSQL database.
-

Full Stack Web Server: Front

- For the Front end choose to use React with the NextJS Framework.
- This Framework allowed me to use Server-Side Rendering which allows the webpage to load significantly faster than with react.
- On the front end I used two things to display my data:

Table for the current data.

Graphs for the overall data.

- For the Graphs I used the CanvasJS Library which is the best open-source graphing library.
-

AWS

- The entire MERN Full-Stack was hosted on an EC2 virtual Server.
 - This allowed me to run the Stack 24/7 without having to tie it to my local PC.
 - Not having the Stack on my local PC allowed me to access the webpage from anywhere as long as the EC2 server was running the stack.
-