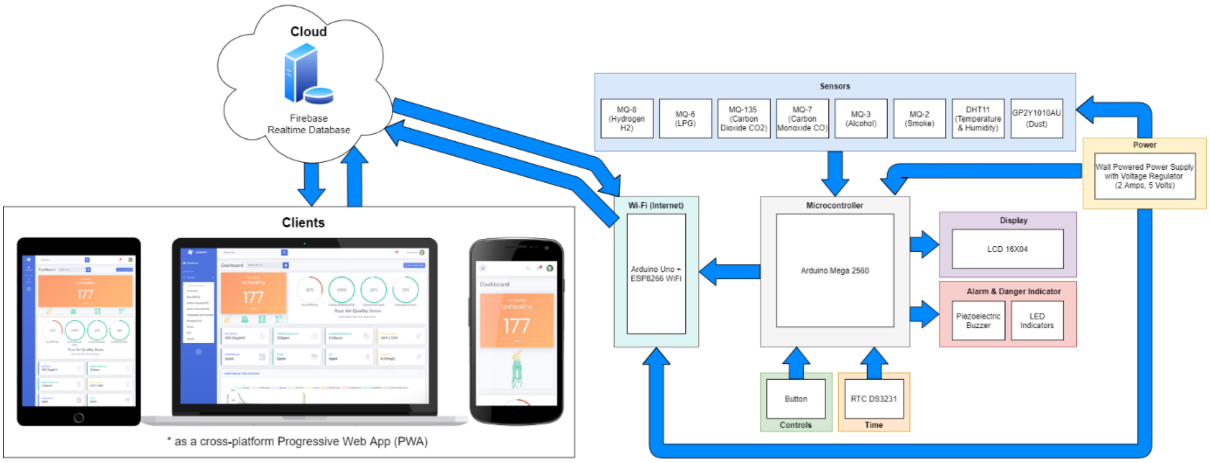
API Reference

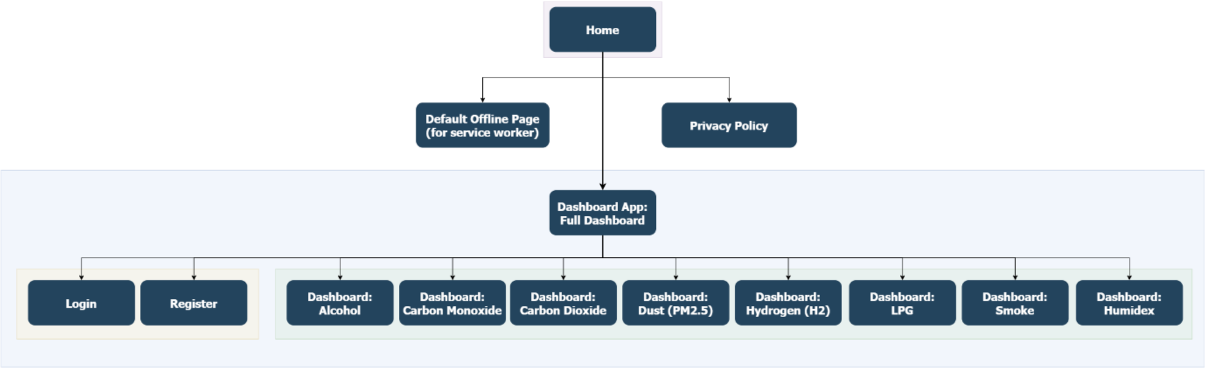
Big Picture

Essentially the system is a cross-platform software and hardware that can monitor indoor air quality. Here you can see the big picture of how the system interacts with each other:



Web App

The software created is a Progressive Web App (PWA) dashboard developed in JavaScript, HTML, and CSS. The following image contains the sitemap for the web app.



In order to create a web app that is interactive, user-friendly, and cross-platform additional libraries, tools, widgets, and other technologies are implemented throughout the application.

The technologies used to develop the web app include:

No.	Name	Description	Category
1	Firebase Realtime Database	Cloud database provider used to store sensor data from the hardware.	Database
2	Firebase Authentication	Authentication provider for user login and encrypted user credential storage.	Authentication
3	Bootstrap	Front-end, web development framework used to build the web app's user interface.	User Interface
4	JQuery	Additional JavaScript library to ease JavaScript development.	JavaScript library
5	JQueryUI	Datepicker on the dashboard.	User Interface
6	IziToast	UI notifications on the dashboard.	User Interface
7	Raphael.js	Vector graphics library	User Interface
8	Chart.js	Chart data visualization library for generating charts on the dashboard.	Charting/Data Visualization
9	Progressbar.js	Circular chart/bar on the dashboard.	Charting/Data Visualization
10	JustGage	Gauges on the dashboard.	Charting/Data Visualization
11	DataTables	Generating tables on the dashboard.	Charting/Data Visualization
12	PapaParse	CSV parser for parsing database entries.	Data Parsing
13	outliers	Library for finding outliers in the database.	Data Parsing
14	Push.js	Push notification framework for sending alerts native to the device.	JavaScript library
15	UpUp	Adds a service worker to the web app.	JavaScript library
16	Intro.js	Step-by-step onboarding tutorial on the dashboard	JavaScript library
17	js-cookie	Javascript library for handling cookies	JavaScript library
18	Moment.js	Library for parsing dates on the database entries and is very important for the web app to be cross-platform as different browsers and devices parse dates differently.	JavaScript library

Hardware

The hardware consists of Arduino microcontrollers and modules programmed using the C language connected to eight sensors sensitive to the specific pollutant monitored. The complexity of the application required that additional external libraries are implemented to achieve the research goals.

The additional external libraries used to develop the hardware include:

No.	Name	Description	Category
1	Wire	Library to communicate with I2C devices.	Modules
2	LiquidCrystal_I2C	Library to control and display values on the I2C LCD display.	Modules
3	RTClib	Library to set and get the time from the DS3231 I2C real time clock.	Modules
4	DHT Sensor Library	Library to convert the output of the DHT-11 sensor to temperature and humidity.	Sensor
5	MQSensorsLib	A collection of libraries to convert the output of the MQ sensors used to the proper units of measurements.	Sensor
6	Sharp Dust Sensor Library	Library to control the dust sensor as well as get the dust density detected.	Sensor
7	FirebaseESP8266	Library to help with communication to the Firebase Realtime Database.	Database
8	ESP8266WiFi	Library to operate the ESP8266 Wi-Fi module using an Arduino.	Wi-Fi
9	DNSServer	Library to set up a DNS Server for the WiFiManager and also.	Wi-Fi
10	ESP8266WebServer	Library to set up a local web server and is used for the WiFiManager.	Wi-Fi
11	WiFiManager	Library to set up a configuration portal to help configure the user's Wi-Fi router credentials for the hardware using a web browser connected to the hardware's access point.	Wi-Fi