Wireless Personal Communications (2021) 118:2921–2940
https://doi.org/10.1007/s11277-021-08162-3

IoT Based Design of Air Quality Monitoring System Web Server for Android Platform

Koel Datta Purkayastha[1] · Ritesh Kishore Mishra[1] · Arunava Shil[1] · Sambhu Nath Pradhan[1]

Accepted: 28 January 2021 / Published online: 9 February 2021

Abstract
The web technology is rapidly increased in various fields. This paper aims to create a server on cloud platform to store data and process the information collected by the air quality monitoring system (AQMS). The monitoring system is developed for transmission and reception of the information received from various data-sources with the use of sensors integrated with microcontroller. The wireless sensing real-time data are transmitted into desired form across the network through internet connection. AQMS is able to monitor concentration of CO_2, CO, NO_2, temperature and relative humidity and stores the concentration values in the database. This research deals with web service with generous API service, which acts as a messenger for different sensors to monitor the sensed data at cloud network. This process comprises a feature of porting sensed data using Channel ID and read API key assigned by the provider that is able to track data at particular intervals. Front end application is developed and hosted on the cloud platform. Additionally, an App is developed on Android platform to visualize real-time data, which are uploaded in the designed web server, displayed data on the smart phone. MIT App Inventor allows programming to develop mobile applications for Android operating system. An online survey is used to examine the trends of air quality and to gather feedback regarding the undergoing process in the environment. The results of air pollution information indicated that the app supported user's achievement.

Keywords Air pollution · Server · IoT cloud · Wireless · Android

✉ Sambhu Nath Pradhan
sambhupradhan.ece@nita.ac.in

Koel Datta Purkayastha
send2koel@gmail.com

Ritesh Kishore Mishra
riteshkm16@gmail.com

Arunava Shil
arunavashil@gmail.com

[1] Department of Electronics and Communication Engineering, NIT Agartala, Tripura 799046, India

1 Introduction

With the rapid development of industrialization and urbanization process in the world, environmental pollution has become a common problem in the global network. Environmental pollution occurs from water pollution, air pollution and soil pollution. In recent years, digital technologies have introduced scaled-down systems with integrated wireless communication nodes for different applications. Wireless sensor network applications for air quality monitoring have been developed in [1–4]. On-chip control systems have encouraged the evolution of wireless-based sensor technology [5]. The architectural design of the sensor nodes in [6] enables people to design industrial and environmental monitoring systems [7, 8]. The administration needs to focus on this section in order to master the air situation and take effective action to maintain the proper air pollution level. The wireless network infrastructures adopt new application tools in the advancement of electronics covering data collection, transfer and processing of information. The embedded system is based on the three technologies mentioned in [9–11]. In the current environment, there is a need to focus on air pollution monitoring activities. Thus, internet applications can be expected to be used on embedded systems with a variety of devices connected to the internet. The web-based pollution monitoring system connects the devices and apparatus to the network. Some important characteristics of the device such as power consumption, accuracy, communication, data collection and data exchange make the possibility of using Internet of Things (IoT) in the design of web-based pollution monitoring system. In this work, it is proposed to incorporate the Internet of Things (IoT) network in the structural design of AQMS. The increasing demand of IoT scenario becomes an essential part of day-to-day lives [12, 13]. The proposal of executing IoT vision that clarifies the effects of IoT in industry based wireless sensor network is reported in [14]. In IoT, advanced server prototypes provide air quality information and air quality indicators for a specific period of time. To implement the functionality of web server over the internet, some actions must be taken. First, web is designed on HTML languages and HTTP protocol, which provides information of a specific position to bring the connection with user interface. Authorized users can access the server by browsing the web to collect information about the required server. The polluted data that is measured by the AQMS in a geographical platform is transmitted through Wi-Fi connectivity to a back end server and data is sent by using user's android application, reported in [15]. In the hardware design of AQMS microcontroller is the main functional part of air pollution detection. Various solutions are acquired from the literature. In the proposed work, Yang et al. [16] developed a technology that provided design ideas for the extensible structure of the system. When various devices are connected to server, the whole system reflects the extensibility as defined in the data transmission protocol in [17]. In this paper, AQMS hardware design, server design and implementation of android App are addressed. From prior works, it has been observed that monitoring the level of pollution from different perspectives is a difficult task. So large infrastructure and proper management are required. If the system can segment the level of pollution as per the area, it can better monitor, provide better solutions.

The contributions of the work presented in this paper are as follows:

(i) A controllable sensor station is developed for monitoring pollutant gases along with climatic conditions in air. This is named as air quality monitoring system (AQMS). The sensors interfaced with the controller include CO_2 sensor, CO & NO_2 sensor, temperature sensor and humidity sensor. All the sensors are controlled and their positions are adjusted in a remote place actively. The system is designed such that each component in the system can be connected to or separated from the network to meet various requirements.

(ii) Designs of server and front end user interface for data analysis are implemented. The user will get updated with pollutant values on hour, day and the duration basis. The interaction with server and sensor network is obtained through JavaScript and JSON based web application. The front end design is followed by model-view-controller (MVC) structure. The back end server is created using spring boot framework in Java. The collected data will be stored in MongoDB database. A graphical user interface (GUI) is developed for analysing and viewing data.

(iii) A smart application for sensor network system is designed for visualizing real-time data on android phone. By using mobile App all the functionality available on the web browser will also be available on the smart phone that make user access control to check the pollution map in graphical way from anywhere in the city.

2 Hardware Architecture

The developed air pollution monitoring system consists of three blocks namely, wireless sensor network, internet enabled air pollution monitoring server which performs the possibility of analyzing and visualizing graphical values with respect to time and an Android App design that provides users for viewing the overall scenario of air quality in mobile phone.

The wireless sensor model is designed as a stand-alone device (AQMS) by integrating hardware modules which measures temperature, humidity in air, CO_2, CO, NO_2. The device transmits the collected pollutant information by Wi-Fi over the internet. When the sensor station is powered on, user can receive measured sensor data globally in numerical and graphical platform through the website remotely. The gathered data is displayed on the serial monitor on the local network. The proposed model for user App and web server-based air quality monitoring system is shown in Fig. 1.

Figure 2 represents hardware support design having all sensors, microcontroller board and ADS1115 that is working as an analog extender. This analog extender is attached to the microcontroller where data is communicated with ESP8266 Wi-Fi chip over the internet.

2.1 NodeMCU: Data Processing Board

NodeMCU is an open source based IoT platform that is enabled with ESP8266 Wi-Fi microcontroller chip and ESP-12 hardware based module. It is a single board microcontroller which operates at 80-160 MHz clock frequency. The board includes serial communication, 128 KB of RAM and offers the advantage of 4 MB flash memory for both program and storage as compared to Arduino Uno. The programming environment of ESP8266 is developed in Arduino IDE. ESP8266 supports IEEE 802.11 network standard [18] and it is having HT40 Wi-Fi transceiver capability. The ESP8266EX Wi-Fi chip is integrated inside NodeMCU hardware for sending measured data in the cloud. The baudrate is selected as 115,200 for realizing wireless transceivers data collection. NodeMCU has on-board Wi-Fi functionality and can be connected to any network as a client. These resources give the opportunity to work on sensor network based challenging research field in embedded system.

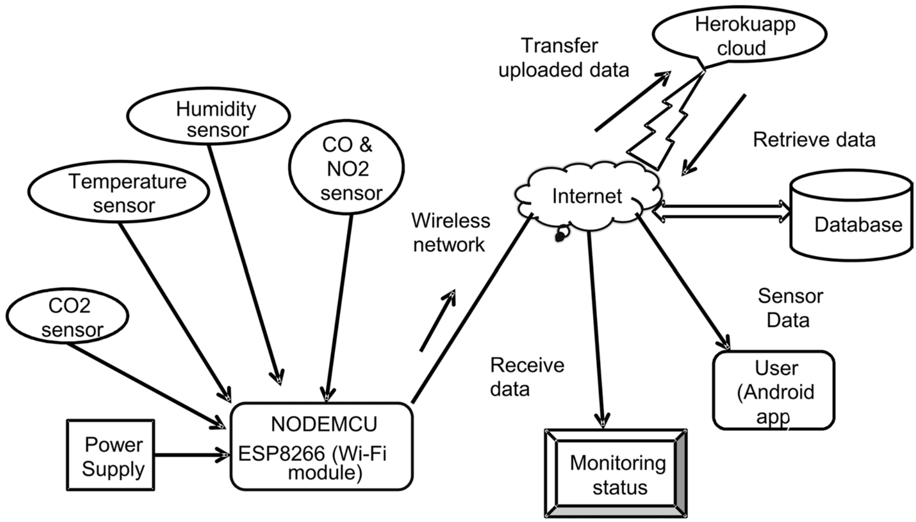


Fig. 1 System hardware blocks of proposed model

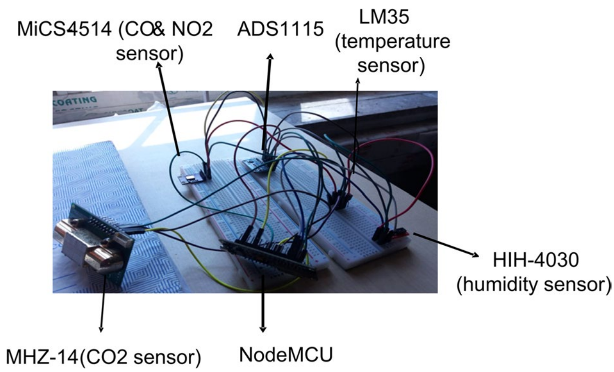


Fig. 2 Hardware setup component connection

2.2 Sensors Array

The sensor array consists of CO_2, CO, NO_2, temperature and humidity. In wireless sensor network, sensor station consists of pollution sensors, processing module, power supply and wireless communication. Sensor station is implemented for collecting data recognized by processor, which controls the operation, stores the collected data. In this paper, wireless communication is based on 802.11b/g/n Wi-Fi standard. Gas sensors are specified depending on sensitivity levels, type of sensitive gas and physical dimensions. Five gas sensors are used on the basis of performance parameters such as accuracy and low power. Table 1 shows the characteristics of sensors for pollution monitoring and operating range as given by central pollution control board (CPCB).

Table 1 Characteristics of sensor materials

Sensor	Parameter	Operating range (ppm)	Voltage (V)
MH-Z14	CO_2	350–5000	4.5–5.5
MiCS 4514	CO	0.88–29.7	4.9–5.1
	& NO_2	0.022–0.213	4.9–5.1
LM35	Temperature	−2 to 40ºC	5.0
HIH-4030	Humidity	50–80%	5.0

2.3 ADS1115 Module

ADS1115 is an analog pin extender module. This is 16-bit module which consumes low power and I2C compatible. The module is incorporated with low drift voltage regulator and an oscillator. As NodeMCU have only one analog pin, therefore ADS1115 is used for integrating multiple analog sensors in this work. The operating voltage ranges from 3.0 to 5.0 V and operating temperature ranges from -40ºC to + 125ºC. It is capable handling four single-ended or two differential inputs. This module has an average current consumption of 150uA. ADS1115 module performs data conversion at the rate up to 860 samples per second (SPS) [19].

3 Software Architecture

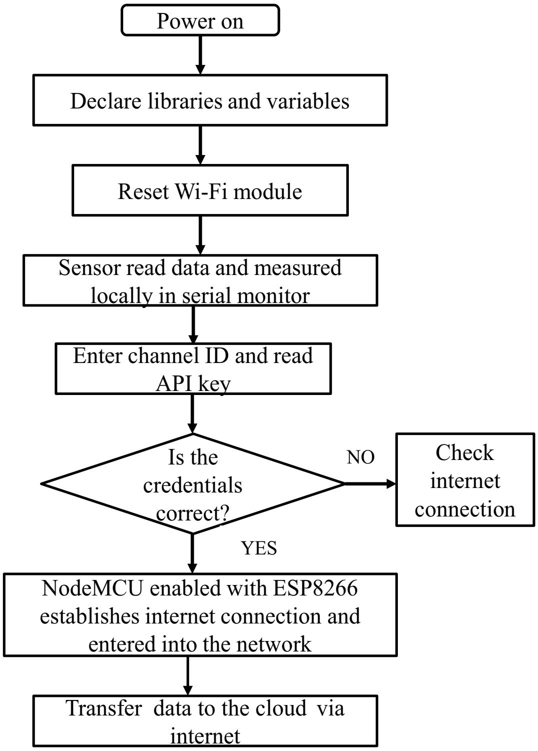
3.1 General Overview

The block diagram of software flow design is shown in Fig. 3. The architectural design of AQMS is based on wireless sensor network applications. In the first case, all sensors are initialized, data is read and messages are sent over Wi-Fi network. When the system is first powered up sensors need to be addressed. The action is taken up with USB port connected to the device. The system is performed through serial communication. The configuration allows the displaying of correctly measured parameters. The information for making connection to wireless LAN requires channel ID and API key, known as server information. The next step is to recognize the authenticity of this information, otherwise we have to check the Wi-Fi connection status and declare an object of class HTTPClient. Once connection is established, the sensor sensed the data and send the data to the server. Next we have to retrieve data with desired information and collect data from the cloud. AQMS is designed to monitor polluted data and analyze air quality in real-time and to update data over the internet by creating log data on a remote server. The availability of real-time data could make people understand about pollution patterns and it impacts on environment.

3.2 Wi-Fi Transmission

The Wi-Fi standard is chosen for data transmission in AQMS. Few sensors are used in the indoor environment monitoring system and messages can be transmitted directly through the router across the entire region. Many administrations maintain Wi-Fi coverage because of the accessibility of Wi-Fi technology. Hence, no requirement for complex

Fig. 3 Software flow chart



routing protocols like Zigbee is considered. Wi-Fi technology has the advantage of using in existing infrastructure where all facilities almost can be found at places in presence of internet connectivity. Data transmission from one device to another is handled by Machine-to-Machine communication technology and it is the key factor of wireless transmission between sensors and smart devices. The protocol for data transmission is chosen to be TCP/IP. It is utilized in establishing communication between server and client. TCP/IP is suitable for one way communication so that the receiving computer can understand the data sent to it. In addition, it is helpful for any microcontroller to access Wi-Fi network. As per the requirement of getting real-time data, extensibility is an important part of the system.

In this work, the communication is developed with Wi-Fi module and the results are displayed on computer or smart phone. The transfer process can be accomplished through multiple connections depending on the user and the server can be configured with a large capacity across different platforms.

3.3 Workflow of the Proposed Design

The workflow of the monitoring system is shown in the block level in Fig. 4. Different blocks represent database, data-source, sensors, server, front end, admin and the user.

The working of several blocks is explained by the following points:

- The admin will create a data channel for a specific place by providing the required data.

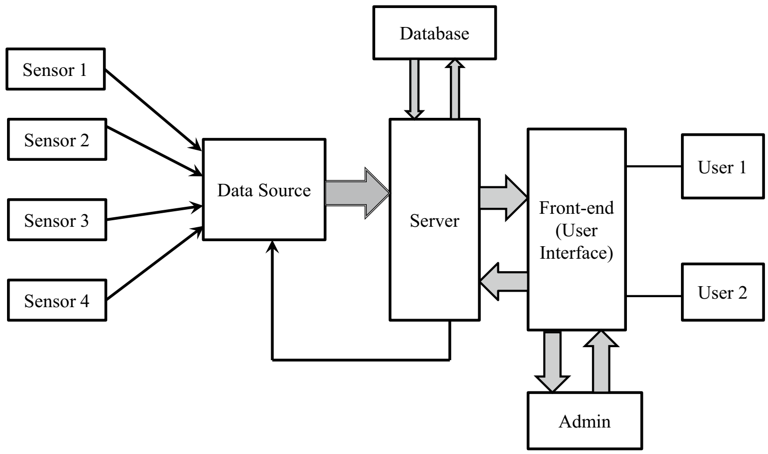


Fig. 4 Workflow of the proposed system

- The admin will configure the sensor for the particular place. This step will be one time installation only. This includes setting the API URL for that sensor to which data will be sent to the server.
- The sensor will sense the data and send to the server. The method used will be HTTP GET method.
- The server will accept the data and process the raw data from the data-sources. The role of data-source is to collect data from the sensor. The data will later be processed and displayed on the display system.
- The server will store the processed data in the database and send the response to the request from client side. Also the server authenticates the admin and users. Moreover it maintains the details of the data-sources.
- The user will request for the data and the server will send the response as requested by the user. The request may be in tabular form or in graphical form.
- The user interface will provide the platform to the user to perform the action required.

3.4 Web Front End and Back End Design

To get the pollutants data acquired by air quality monitoring system, a website is developed. The data collected by different sensors is visualized and analyzed on real-time. The collected data is stored in the database for further applications. The data-source can send the data to the server by calling add feed API. Exception handling is implemented in most of the API to avoid any runtime issues.

3.4.1 Front End Design

The front end is not considered as absolute having necessary role in the design of the overall monitoring system, it describes how the experimental data can be observed. The front end design validates the users to monitor all pollutant data remotely, which are evaluated in real-time. Evaluation principle follows the method of design and development of human–computer interaction are discussed in [20]. The user interface of the web is designed well to make it user-friendly. The front end is developed using MVC approach. MVC is a popular way to organize the applications into various logical components.

The control flow of the front end web design is shown in Fig. 5. Initially, the home page will be loaded by the browser. Here the new user can register, however the existing user will have to login using the credentials. Once login is completed, the dashboard of the user will be loaded and there using various options, user can perform various task. As the front end is developed using MVC structure, all the components are made separately in the Angular framework of JavaScript for inspection of status of sensor data. Angular 4 is known as a JavaScript framework for building web applications for animation, http service and materials. HTML, the standard mark-up language is used for structuring web pages. Cascading style sheets (CSS) is a language used to describe the way of making web pages presentable. Home screen is the first webpage that is loaded in the browser when the website is accessed. From this page, user can login to the application. Figures 6 and 7 represent the home screen and login steps respectively:

Next the user will have to login with the credentials to avail the service of AQMS. The user interface (UI) will set the cookies in the browser on successful login. The dashboard has all the option to the user to perform the action. Heroku is a cloud platform to deploy, manage and scale modern Apps. Heroku CLI is a command line interface for the deployment of the application on Heroku cloud. The important feature of the Heroku functionality

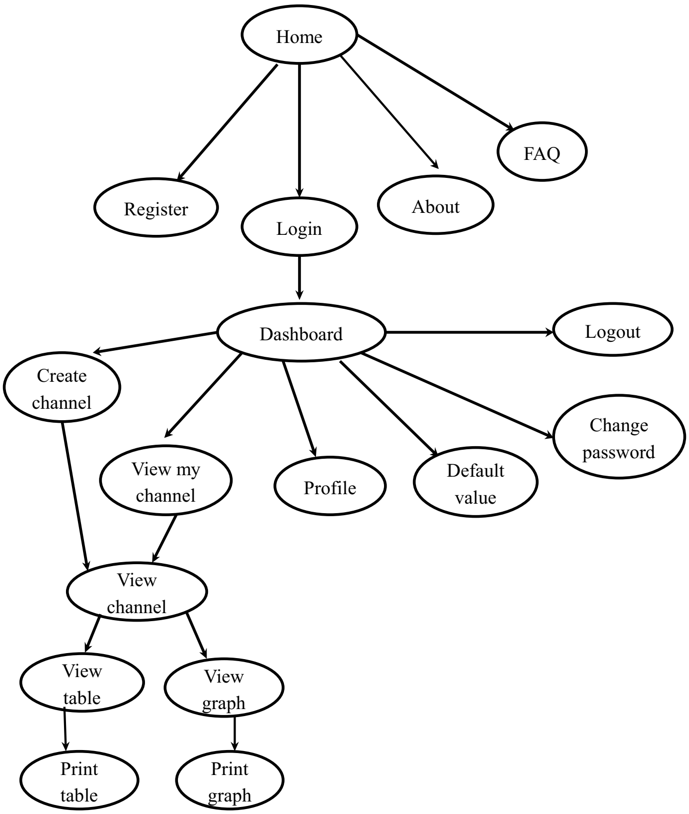


Fig. 5 Front end workflow of the web application system

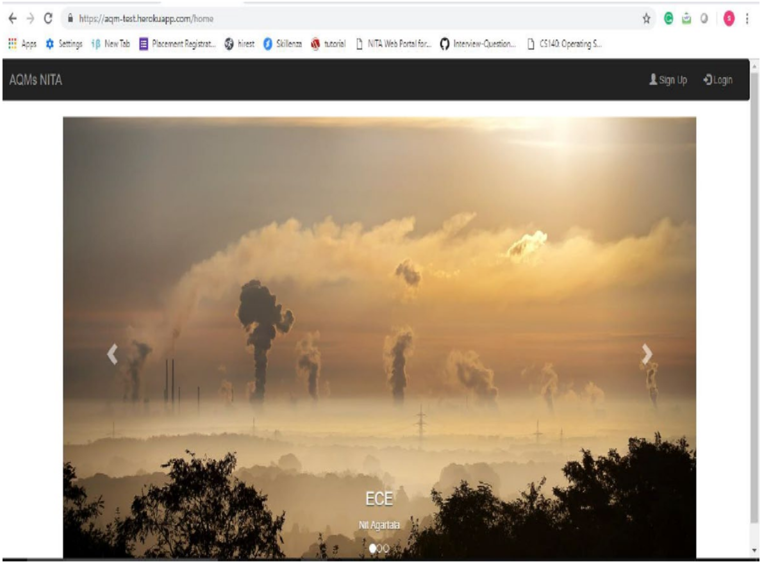


Fig. 6 Home page of the website

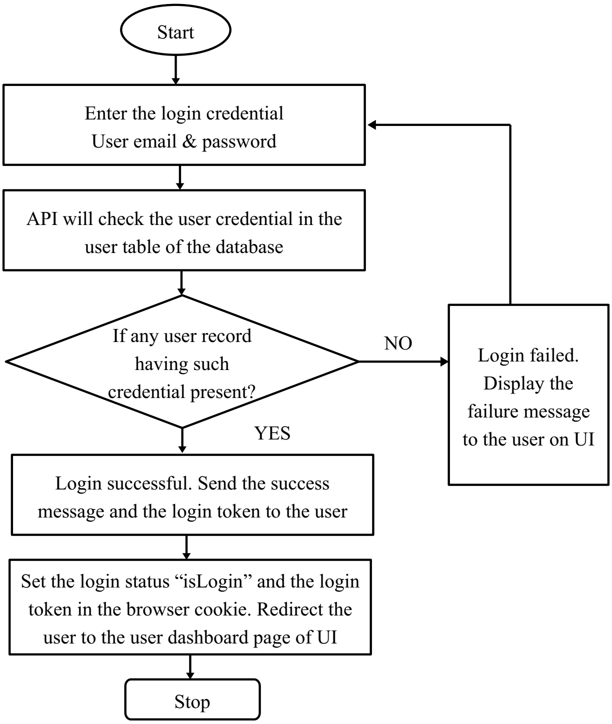


Fig. 7 Login flow chart to access the server

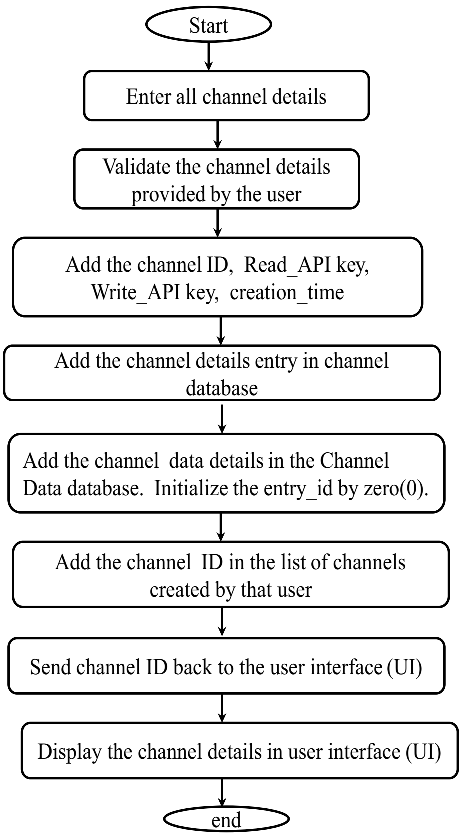
is the term channel. Channel creation allows the creation of new data-source. The channel represents the data-source of a particular location. Once channel is created then user can analyze, process, estimate and visualize the information of the sensed data. The feature Add Feed is used in the data-source i.e. at the sensor. The Add Feed URL API hits the data-source to add the data into the channel and data is stored in the database. The flowchart of the channel creation process is presented in Fig. 8.

3.4.2 Back End Design

The main component block for website is the back end design which takes the responsibility of completing the central part of the system with addition of receiving messages from different gas sensors along with environmental parameter. The task is to simultaneously store current data and previously processed data in the database and return to the front end whenever required. Moreover, it plays an important role in initiating the request for hardware application in the air pollution model.

The concrete structure of the website design and framework selection for the back end along with the various technologies used in the design of server is discussed here. The programming is performed using object-oriented programming (OOP) language approach which provides a clear structure for defining abstract data types in which implementation

Fig. 8 Create channel flow chart



details are hidden. The back end API follows representational state transfer (REST) principle with the advantage of most efficient, widespread and logical standard in creating APIs for internet services. An IDE is referred as spring tool suite (STS) for the development of API using spring boot framework. The language used for back end programming is JAVA. The spring framework acts as an application framework for Java platform. Maven is a powerful project management tool based on the project object model (POM). POM is the basic unit of Maven's work. It is an XML file that contains project information. Additionally, there is information about configuration details to build the project that is used by Maven. Java Persistence API (JPA) is a Java specification used to access, persist, and manage data between Java objects and a database.

The back end server is created using spring boot framework in Java. Model classes are created in the package. The API requests are made to the controllers. A model contains the data of the application. A data can be single object or set of objects. The controller calls the method from the service class. The controller's class acts as the front controller. It is responsible for managing the flow of the Spring MVC application. The service class call the repository class method for data fetching and storage purpose. The web layer communicates only with service layer that acts as a transaction boundary. The service layer is responsible for approval and carry the business logic of our application. This layer manages the domain model objects and communicates with repository layer and other services. Repository class extends the JPA class and performs all the database related work. CrudRepository refers spring data interface. For using it, we have to produce data interface by expanding CrudRepository for a particular type. Spring allows CrudRepository implementation class at runtime in spontaneous manner. It contains various procedures namely save, findById, count etc. Spring boot automatically identify repository if package of that repository interface is matched. The entire request to the server and response from the server is done via the API calls.

4 App Inventor Architecture Design Approach

MIT App Inventor, a drag-and-drop programming tool [21] is a web technology based system in which users can design mobile applications in android devices. Android is a software that includes technical information and key applications. App Inventor is an online applications development environment. It has the facility on graphical user interface [22]. The programming offers users the ability to build innovative applications for mobile phones and also to focus on the logic. The programming type is known as block-based programming language [23]. The user interface is divided into two parts namely designer: for selecting components to be used in the App and another one is block editor: for mapping behavioral conditions of the App. Focusing on the device functionality, coding the behavior is handled with blocks which are developed in a Java web application. Corrective steps are taken to solve research field related problems for complete design of the App and writing it in coded format. For example, input, control, variable, logical, mathematical processes are identified for solving various problems at the block level and the results of the solution are specified. The steps required for the execution of the program is formulated into a group of commands as per MIT App Inventor language version. App Inventor exports APK file as a file package which can be installed on smart device providing a QR code for APK file. When the App is designed, it is tested on the desktop and then run on the Android device for live testing of the program. The results displayed in the App are compared with the real-time air pollutant data on the web page that can be accessed from any location. All the functionality available on the web browser will also be available on the mobile App.

5 Results and Discussion

When the coding is completed, it can be tested and authenticated. After the successful completion (or authentication), the user can have the ability to see sensor data output and transferred the data to the cloud through internet on global network. Data is transferred from back end design and channel data is displayed in tabular format as well as in graphical format. Then the sensor data is downloaded without any requirement of refreshing the webpage accurately for further information. Three pollutant gases CO_2, CO, NO_2 have been identified for regular monitoring the concentration level of CO_2, CO, NO_2 gases at all location. The meteorological parameters, relative humidity and temperature are also monitored by this air quality monitoring system.

5.1 Web IoT Output Results

According to CPCB, the status of air quality has been reported through a large amount of data. So it is important that information about air quality is kept in the public domain in plain linguistic language so that a common person can easily understand it. Color coding is given to make awareness about the air quality status and associated health impacts. Figure 9 shows the channel data of CO_2, CO, NO_2 with environmental parameter in graphical format.

The above results show the graphical output of the concentration level of CO_2, CO, NO_2 gases and the information about, temperature and humidity at the cloud and provide the opportunity to observe after logging in the Herokuapp website by using created username and password.

5.2 Mobile App-android Results

When the Channel ID and Read API keys are inserted, IoT-based integrated sensor results can be visualized on the portable handset. The App Inventor is included with blocks contained Java web application for designing the App. The data can be received from the server in JSON format which can be easily readable for human being. JSON is a text format and is a subset of JavaScript programming language. A user interface is designed for the purpose of viewing and study analog sensor data to some extent. The measured data based on three gases and weather condition follows central pollution control board (CPCB) table and the concentration level maintains local air quality standard. We need to follow the step-wise procedure in the different sections connected to sensing and monitoring station. The main user interface menu and corresponding App's screen are shown in Figs. 10, 11, 12, 13, 14, 15, 16 and 17 respectively. After the completion of the task the App is reviewed and also can be published in the Google Play by creating Google Play developer account. The range of gases for air quality index (AQI) supported by CPCB is presented in Table 2. Ranges given in ppm (parts per million).

The normal temperature in India ranges from -2 to 40ºC and relative humidity is set from 50 to 80%.

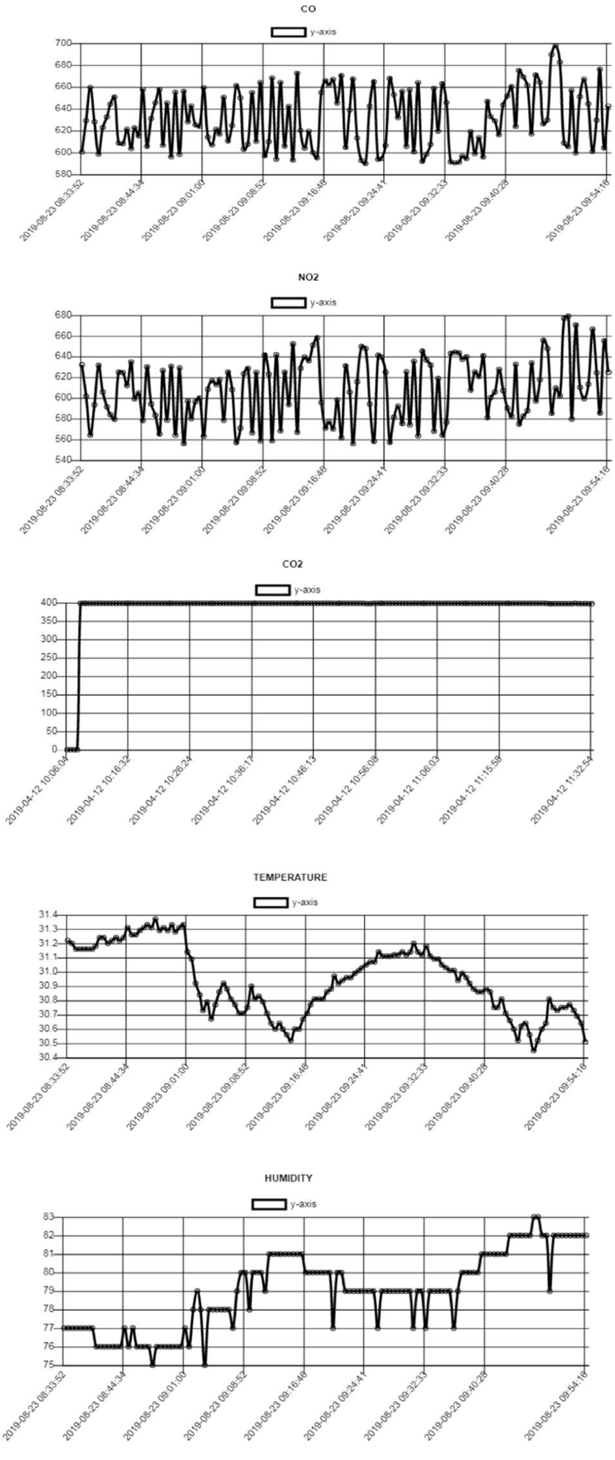


Fig. 9 Channel data of, CO, NO_2, CO_2, temperature and humidity in graphical format

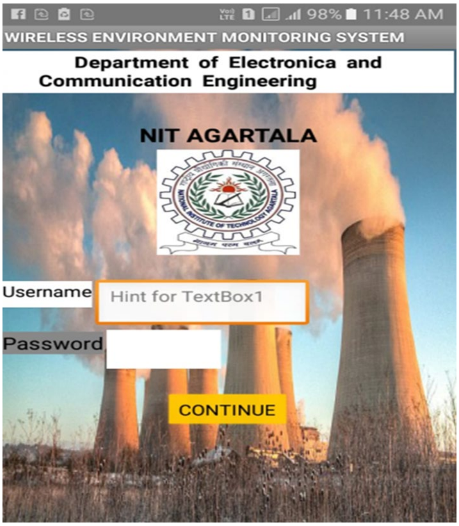


Fig. 10 Main user interface menu of mobile App

Fig. 11 Enter username and password

6 Conclusion

The developed air quality monitoring system (AQMS) is based on Indian Air Standard. The system is going to be an important tool for public information regarding the quality of air surrounding the environment. The presented design will be helpful in constructing energy efficient wireless sensor network (WSN) in sensor node level as the measured values confirmed the effectiveness in detecting ambient air quality. The designed server serves as a dedicated server for AQMS. This will increase the efficiency of the system. The datasource can send the data to the server by calling add feed API. Processing of data is done

Fig. 12 Select the wireless network

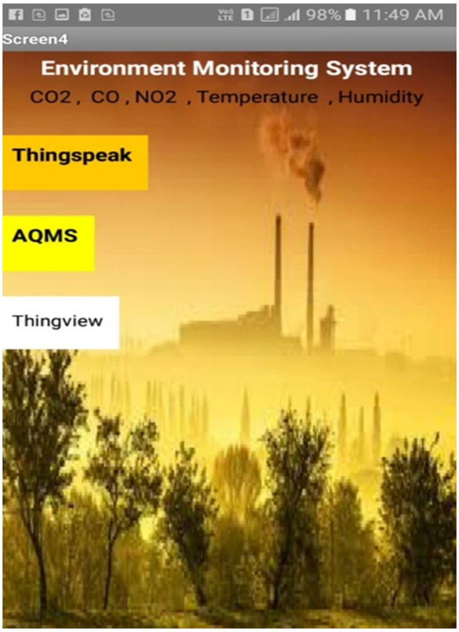


Fig. 13 Click on AQMS

on the default values set by the admin. Database is hosted on the cloud platform. The API is developed and tested. Also, the API is hosted on the cloud platform. Front end application is developed and hosted on the cloud platform URL: https://aqm-test.herokuapp.com

Further, the design successfully provides a tool that enables users to make interaction by observing air pollutant data on developed Android App. A mobile application

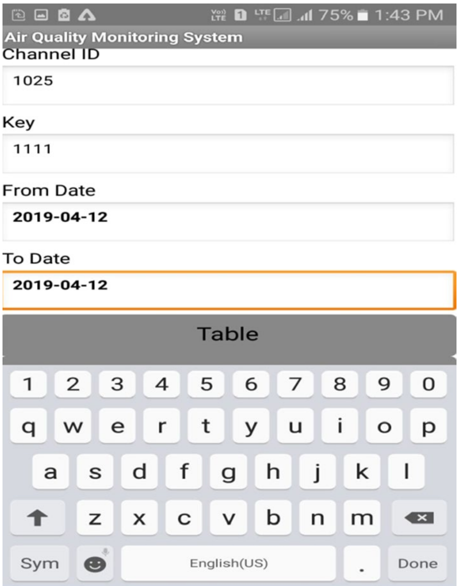
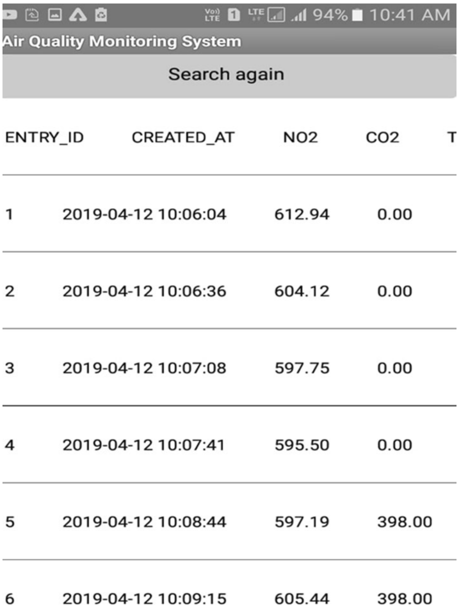


Fig. 14 Enter channel ID and read API key

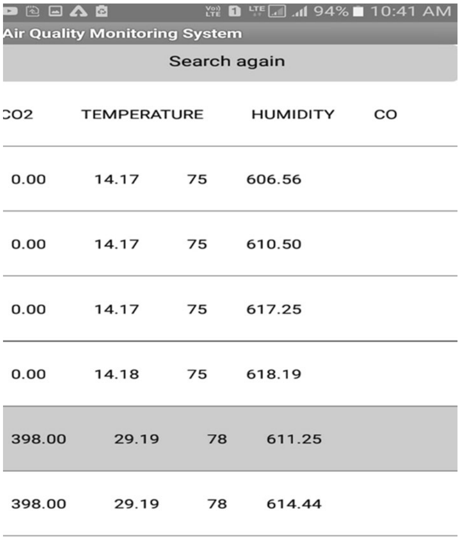


ENTRY_ID	CREATED_AT	NO2	CO2	T
1	2019-04-12 10:06:04	612.94	0.00	
2	2019-04-12 10:06:36	604.12	0.00	
3	2019-04-12 10:07:08	597.75	0.00	
4	2019-04-12 10:07:41	595.50	0.00	
5	2019-04-12 10:08:44	597.19	398.00	
6	2019-04-12 10:09:15	605.44	398.00	

Fig. 15 NO_2, CO_2 data on Android phone

is created to provide the same UI facility in the mobile App so that the person from any location can see the current status of sensing parameters on smart phone. The scope of the server and mobile App becomes a broad possibility of application in future study.

Acknowledgements This work is supported by SMDP-C2SD project sponsored by MeitY, Govt. of India.

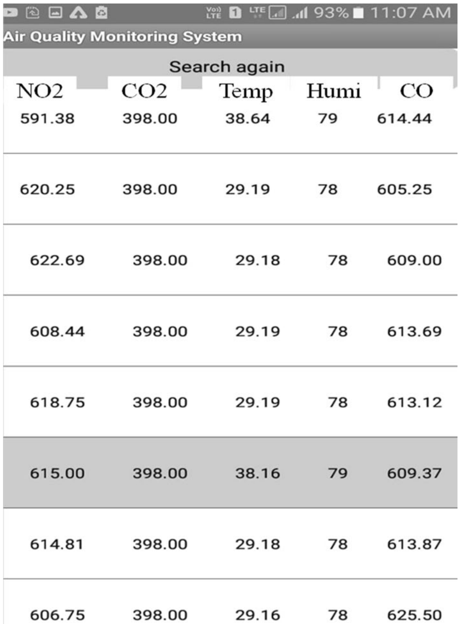


Air Quality Monitoring System

Search again

CO2	TEMPERATURE	HUMIDITY	CO
0.00	14.17	75	606.56
0.00	14.17	75	610.50
0.00	14.17	75	617.25
0.00	14.18	75	618.19
398.00	29.19	78	611.25
398.00	29.19	78	614.44

Fig. 16 Temperature, humidity & CO value on Android phone



Air Quality Monitoring System

Search again

NO2	CO2	Temp	Humi	CO
591.38	398.00	38.64	79	614.44
620.25	398.00	29.19	78	605.25
622.69	398.00	29.18	78	609.00
608.44	398.00	29.19	78	613.69
618.75	398.00	29.19	78	613.12
615.00	398.00	38.16	79	609.37
614.81	398.00	29.18	78	613.87
606.75	398.00	29.16	78	625.50

Fig. 17 Sensor data on Android phone

References

1. Chung, W., & Oh, S. (2006). Remote monitoring system with wireless sensor module for room environment. *Sensors and Actuators B: Chemical, 113*(1), 64–70. https://doi.org/10.1016/j.snb.2005.02.023.
2. Zhao, Z., Wang, J., Fu, C., Liu, Z., Liu, D., & Li, B. (2018). Design of a smart sensor network system for real-time air quality monitoring on green roof. *Journal of Sensors*. https://doi.org/10.1155/2018/1987931.

Table 2 Air quality index

AQI range	CO range	CO_2 range	NO_2 range
Good	0–0.87	0–350	0–0.021
Satisfactory	0.88–1.75	350–450	0.022–0.042
Moderate	1.76–8.73	450–600	0.043–0.095
Poor	8.74–14.85	600–1000	0.096–0.149
Very poor	14.86–29.7	1000–2500	0.149–0.213
Severe	29.8 +	2500–5000	0.213 +

3. Thakur, D., Kumar, Y., Kumar, A., et al. (2019). Applicability of wireless sensor networks in precision agriculture: A review. *Wireless Personal Communications, 107,* 471–512. https://doi.org/10.1007/s11277-019-06285-2.
4. Purwanto, P., Suryono, S., & Sunarno, S. (2019). Design of air quality monitoring system based on web using wireless sensor network. In *2019 3rd international conference of chemical and materials engineering*. https://doi.org/https://doi.org/10.1088/1742-6596/1295/1/012043
5. Chong, C., & Kumar, S. P. (2003). Sensor networks: Evolution, opportunities, challenges. *Proceedings of the IEEE, 91*(8), 1247–1256. https://doi.org/10.1109/jproc.2003.814918.
6. Gao, R., & Fan, Z. (2006). Architectural design of a sensor node controller for optimized energy utilization in sensor networks. *IEEE Transactions on Instrument and Measurement, 55*(2), 415–428. https://doi.org/10.1109/TIM.2006.870321.
7. Salvadori, F., Campos, M., Sausen, P., Camargo, R., Gehrke, C., Rech, C., et al. (2009). Monitoring in industrial systems using wireless sensor network with dynamic power management. *IEEE Transactions on Instrument and Measurement, 58*(9), 3104–3111. https://doi.org/10.1109/TIM.2009.2016882.
8. Gallo, D., Landi, C., & Pasquino, N. (2009). Multisensor network for urban electromagnetic field monitoring. *IEEE Transactions on Instrument and Measurement, 58*(9), 3315–3322. https://doi.org/10.1109/TIM.2009.2022384.
9. Molderink, A., Bakker, V., Bosman, M. G. C., et al. (2010). Management and control of domestic smart grid technology. *IEEE Transactions on Smart Grid, 1*(2), 109–119. https://doi.org/10.1109/tsg.2010.2055904.
10. Pan, M.-S., Yeh, L.-W., & Chen, Y.-A. (2008). A WSN-based intelligent light control system considering user activities and profiles. *IEEE Sensors Journal, 8*(10), 1710–1721. https://doi.org/10.1109/JSEN.2008.2004294.
11. Lu, S., Huang, Xi., & Cui, Li. (2009). Design and implementation of an ASIC-based sensor device for WSN applications. *IEEE Transactions on Consumer Electronics, 55*(4), 1959–1967. https://doi.org/10.1109/TCE.2009.5373756.
12. Meisner, D., Sadler, C. M., Barroso, L. A., Weber, W. D., & Wenisch, T. F. (2011). Power management of online data-intensive services. In *2011 38th annual international symposium on computer architecture (ISCA), San Jose*, pp. 319–330. https://doi.org/https://doi.org/10.1105/2000064.2000103
13. Jo, J., Jo, B., Kim, J., Kim, S., & Han, W. (2020). Development of an IoT-based indoor air quality monitoring system. *Journal of Sensors*. https://doi.org/10.1155/2020/8749764.
14. Kruger, C. P., & Hancke, G. P. (2014). Implementing the Internet of Things vision in industrial wireless sensor networks. In 2014 *IEEE international conference industrial and information systems*, pp. 627–632. https://doi.org/https://doi.org/10.1109/INDIN.2014.6945586
15. Suryawanshi Shashikant, U., Dhang, Deepganga., Chougule Ashish, A., & Mote Shailendra, B. Implementation of embedded wireless air pollution monitoring system. *IOSR Journal of Electronics and Communication Engineering (IOSR-JMCE)* ISSN: 2278–2834-, ISBN: 2278–8735, PP: 27–30.
16. Yang, H., Qin, Y., Feng, G., & Ci, H. (2013). Online monitoring of geological CO2 storage and leakage based on wireless sensor networks. *IEEE Sensors Journal, 13,* 556–562. https://doi.org/10.1109/JSEN.2012.2223210.
17. Lu, C. (2017). Improving system extensibility via an IOT- interoperable platform for dynamic smart homes. In 2017 *International conference on applied system innovation (ICASI)*, pp. 13–17 (in Japan). https://doi.org/https://doi.org/10.1109/ICASI.2017.7988138

18. Espressif Systems (2015). ESP8266EX Datasheet. [Online]. Available: https://cdn-shop.adafruit.com
19. Texas Instruments (2018) ADS1115 Data Sheet. [Online]. Available: www.ti.com
20. Jacko, J. A. (2011). Human-computer interaction design and development approaches. Springer (in Berlin Heidelberg).
21. Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice programming environments. Scratch & App Inventor: a first comparison. In 2014 *IDEE'14 Proceedings of the 2014 workshop on interaction design in educational environments* 1. New York: ACM. https://doi.org/https://doi.org/10.1145/2643604.2643613
22. Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2014). App inventor 2, 2nd edn. O'Reilly Media 362.
23. Meehan, D., & Sabin, M. (2013). Quiz Power: A mobile app with app inventor and XAMPP service integration. In 2013 *SIGITE '13 Proceedings of the 14th annual ACM SIGITE conference on information technology education* (New York: ACM), pp. 103–108. https://doi.org/https://doi.org/10.1145/2512276.2512300

Koel Datta Purkayastha received the M.Tech . degree in Microelectronics and VLSI Design from HeritageInstitute of Technology, Kolkata, India, in 2014. She is currently pursuing Ph.D. degree from NIT Agartala, India.Her research area included with embedded systems, sensors, wireless sensor networks, Internet of Things. She isalso working as a Project Assistant, SMDP-C2SD Project in NIT Agartala.

Ritesh Kishore Mishra received B.Tech. degree in Electronics and Communication Engineering from NITAgartala, India, in 2019. He is interested in cloud computing, IoT for industrial applications.

Arunava Shil received B.Tech. degree in Electronics and Communication Engineering from NIT Agartala, India, in2019. He is interested in sensors, wireless networks, IoT for industrial applications.

Sambhu Nath Pradhan received his M.Tech degree from BESU and PhD form Indian Institute of Technology Kharagpur. Currently he is an Associate Professor, Department of Electronics and Communication Engineering in National Institute of Technology Agartala. His research interest includes Low Power and Thermal Aware VLSI Design and Synthesis.