DIGITALIZED CERTIFICATE GENERATION OF EMISSION TEST

# A PROJECT REPORT

***Submitted by***

**SONIKA.M [REGISTER NO: 211417104263]** **SUJEETHA.B [REGISTER NO: 211417104274] VAISHNAVI.G[REGISTER NO: 211417104286]**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

# COMPUTER SCIENCE AND ENGINEERING PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

**ANNA UNIVERSITY : CHENNAI 600 025**

# APRIL 2021

**BONAFIDE CERTIFICATE**

Certified that this project report**“DIGITALIZED CERTIFICATE GENERATION OF EMISSION TEST”**is the bonafide work of “**SONIKA.M(211417104263),SUJEETHA.B(211417104274),VAISHNAVI.**

**G(211417104286)”**who carried out the project work under my supervision.

# SIGNATURE SIGNATURE

**Dr.S.MURUGAVALLI,M.E.,Ph.D., P.DEEPA,M.E.,**

# HEAD OF THE DEPARTMENT ASSOCIATE PROFESSOR SUPERVISOR

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE

NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on...........................

# INTERNAL EXAMINER EXTERNALEXAMINER

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI, Dr.C.SAKTHIKUMAR,M.E.,** and

**Tmt.SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank our **Project Guide Mrs.P.Deepa,M.E.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

# SONIKA.M, SUJEETHA.B, VAISHNAVI.G.

**ABSTRACT**

Nowadays, the most common problem faced is that there is a huge increase in pollution. This is because of an increased diesel engine population has created pressures on controlling diesel PM and NOx emissions. In India there are more than 250 millions transports we are using which emits the toxic gas for best example we come to know in Delhi . Due to emission of gases how people we suffered a lot even to get oxygen . there is an oxygen bank where we take a pure O2 this is the situation existing. Emission standards implemented in 2005-2010 time frame additionally require the use of exhaust after treatment methods on new diesel engines. This is where pollutants such as NOx, CO and PM are created which leads to incomplete oxidation of fuel combustion. To avoid such toxic pollutants, our idea is to develop a system called **“Digitalized Certificate Generation Of Emission Test”**, which will be useful to monitor the emission of the vehicle. Firstly, air pollution sensors are used. Then, microcontroller such as AVR will be used to process the readings from those sensors and Wi-Fi gateways are used for pushing those data to the mango dB database using application program interface. Secondly, a web application will be developed to monitor the abnormal emission of the vehicle and a mobile application will be developed using react native to get abnormal emitting vehicle information. Finally, a certificate will be automatically generated and send through mail. Thus, this project will be useful to identify the pollution emitting vehicle easily and can control the air pollution.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO**. |
|  | **ABSTRACT** | iii |
|  | **LIST OF FIGURES** | iv |
|  | **LIST OF ABBREVIATIONS** | viii |
| **1.** | **INTRODUCTION** | **9** |
|  | 1.1 Overview | 10 |
|  | 1.2 Problem Definition | 10 |
| **2.** | **LITERATURE SURVEY** | **11** |
| **3.** | **SYSTEM ANALYSIS** | **22** |
|  | 3.1 Existing System | 22 |
|  | 3.2 Proposed system | 23 |
|  | 3.3 Requirement Analysis and Specification | 23 |
|  | 3.3.1 Input Requirements | 23 |
|  | 3.3.2 Output Requirements | 23 |
|  | 3.3.3 Functional Requirements | 24 |
|  | 3.4 Technology Stack | 25 |
| **4.** | **SYSTEM DESIGN** | **26** |
|  | 4.1. ER diagram | 26 |
|  | 4.2 Data dictionary | 27 |

**CHAPTER NO. TITLE PAGE NO**.

4.3 UML Diagrams 28

1. SYSTEM ARCHITECTURE 31
   1. [Architecture Overview 31](#_TOC_250007)
   2. [Module Design Specification 33](#_TOC_250006)
   3. Program Design Language 42
2. SYSTEM IMPLEMENTATION 46
   1. Client-side coding 46
   2. [Server-side coding 55](#_TOC_250005)
3. SYSTEM TESTING 57
   1. Test Cases & Reports / Performance Analysis 57
4. CONCLUSION
   1. [Conclusion and Future Enhancements 61](#_TOC_250004)

[APPENDICES](#_TOC_250003)

* 1. [Sample Screens 62](#_TOC_250002)
  2. [Publications 66](#_TOC_250001)

[REFERENCES 71](#_TOC_250000)

# LIST OF FIGURES

* 1. ER Diagram 21
  2. Database Design 22
  3. Sequence Diagram 23
  4. Collaboration Diagram 24
  5. Use-Case Diagram 24
  6. System Architecture 26
  7. ReactJS Architecture 27
  8. Processing Module 29
  9. ESP8266 Module 30
  10. MongoDB Architecture 31
  11. Mobile App Development 33
  12. Email Integration 36
  13. Invalid Login of Web App 44
  14. Invalid Login of Mobile App 44
  15. Kit Connected with Sensor,AVR,GPS Module 44
  16. Login Screen of Web app 45
  17. When No Data is Found 45
  18. Data Entry 45
  19. Acceptance Details Of Admin 45
  20. Login Screen of Mobile App 46
  21. User Tracking 46

A1.1 Kit Connected with Equipment 48

A1.2 Login Page Of Web App 48

A1.3 Invalid Login 48

A1.4 Data Unavailable 48

A1.5 Data Entry Into Web App 48

A1.6 Accept or Reject 48

A1.7 Login Page Of Mobile App 49

A1.8 Invalid Login 49

A1.9 User Vehicle Tracking using GPS Data Received 49

A2.1 Publication Certificates 52

A2.2 Image Of Published Journal Paper 50

# LIST OF ABBREVIATIONS

1. ST-MFGCN – A Spatiotemporal Graph Convolution Multifusion Network
2. AVR – Advanced Virtual RISC
3. EEPROM – Electrically Erasable Programmable Read Only Memory
4. SRAM – Static Random Access Memory
5. MCU – Microcontroller Unit
6. SOC – System On a Chip
7. SPI – Serial Peripheral Interface
8. PWM – Pulse-Width Modulation
9. GPS – Global Positioning System
10. SMTP – Simple Mail Transfer Protocol
11. MSA – Mail Submission Agent
12. MTA – Mail Transfer Agent
13. MDA – Mail Delivery Agent
14. MUA – Mail User Agent
15. LAN – Local Area Network

# CHAPTER – 1 INTRODUCTION

Vehicular pollution has grown at an alarming rate due to growing urbanisation in India. The air pollution from vehicles in urban areas, particularly in big cities, has become a serious problem. The pollution from vehicles has begun to tell through symptoms like cough, headache, nausea, irritation of eyes, various bronchial and visibility problems.The main pollutants emitted from the automobiles are hydrocarbons, lead/benzene, carbon monoxide, sulphur dioxide, nitrogen dioxide and particulate matter. The main cause of vehicular pollution is the rapidly growing number of vehicles.The other factors of vehicular pollution in the urban areas are 2-stroke engines, poor fuel quality, old vehicles, inadequate maintenance, congested traffic, poor road condition and old automotive technologies and traffic management system.In India, the number of vehicles increased from 0.3 million in 1951 to 58.3 million in 2001-02. About half the vehicles are concentrated in 39 metropolitan cities (cities with population of over one million). The two wheelers are the major contributors of vehicular air pollution followed by four-wheeler (e.g., car, jeep, taxi etc.), trucks and buses in decreasing order of magnitude.Delhi is a typical example of air pollution by vehicles. Table 9.6 shows that Delhi’s vehicular population increased from an insignificant of 2.17 lakh in 1971 to 44 lakh in 2004. Over

2.5 lakh vehicles were registered in Delhi in 2004—more than twice the number registered in 2003. Of the total vehicular population in Delhi in 2004, 13.37 lakh were four wheelers, 27.8 lakh two wheelers, more than 80 thousand autorickshaws, 18.4 thousand taxies, 26.9 thousand buses and 1.55 lakh goods vehicles. This figure is likely to rise to 60 lakh in 2011.

# Overview

Digitalized Certificate Generation Of Emission Test requires an embedded systems as the entire system is controlled by the microprocessor. An embedded system is a microprocessor-based computer hardware system with software that is designed to perform a dedicated function, either as an independent system or as a part of a large system. At the core is an integrated circuit designed to carry out computation for real-time operations.ReactJs and React Native features are used ,as we are going to develop web application and mobile application.For those app development,these two features will be useful. ReactJS basically is an open-source JavaScript library which is used for building user interfaces specifically for single page applications. It’s used for handling view layer for web and mobile apps. React also allows us to create reusable UI components. React Native is an open-source framework that allows to build a mobile app with only JavaScript. React Native is an open-source platform.There’s a great advantage to using a community-driven technology.

# Problem Definition

The problem is that, when the emission test center connect the device to the silencer of vehicle, emission test values which are given as input to the system are used to generate the certificate. But there will be an option to change the values and generate the fake certificate and take the printout there itself. In addition to this, the vehicle owner can give bribe to the person who generate the certificate by adjusting the values.So,there is a lack of automated system to monitor emission level in those vehicles and to avoid fake emission control certificates which are being issued through bribing. This leads to air pollution in the environment.There is no technological solution to this problem.

# CHAPTER – 2 LITERATURE SURVEY

Various techniques and approaches have been proposed and developed to control the emission of the gases,which leads to pollution.The proposed approaches have focused only on the some of the basic measures.The following shows survey did for emission control.The most popular of the existing techniques is declared as follows.

[1]. Zhenyi Xu , Yu Kang , Senior Member, IEEE, Yang Cao , Member, IEEE, and Zhijun Li , Senior Member, IEEE,”Spatiotemporal Graph Convolution Multifusion Network for Urban Vehicle Emission Prediction”[2020].Urban vehicle emission prediction can help the regulation of vehicle pollution and traffic control. However, it is hard to predict the spatiotemporal variation of vehicle emission because of the spatial interactions and temporal correlations between different road segments as well as the high nonlinearity and complexity of vehicle emission variation. The existing methods solve the problem by splitting the region into standard segments or grids based on conventional deep learning methods.To address these issues, a spatiotemporal graph convolution multifusion network (ST-MFGCN) is proposed to leverage the graph structural properties as the inherent connectivity of road network for urban vehicle emission prediction, which can capture the vehicle emission spatiotemporal variation patterns and learn the effects of complex environmental factors.

[2]. Yao Ma and Junmin Wang, Senior Member, IEEE,”Predictive Control for NOx Emission Reductions in Diesel Engine Vehicle Platoon Application ”.This paper presents a predictive control design for diesel vehicle platoons, such as medium- and heavy-duty truck platoons, to achieve lower NOx emissions. By

incorporating vehicle longitudinal dynamics control and powertrain estimation with aftertreatment system control, an enhanced emission performance can be attained for the entire platoon considering the different operating conditions for individual vehicles. The proposed method utilizes exhaust gas predictive information within a preview horizon to generate optimal control actions such that the tradeoff between NOx reduction efficiency and NH3 slip in the aftertreatment systems can be well managed.

[3]. Qilong Han 1 , Peng Liu 1 , Haitao Zhang1,\* and Zhipeng Cai2 , Member, IEEE,”A Wireless Sensor Network for Monitoring Environmental Quality in the Manufacturing Industry”.Urban industrial plant areas are highly concentrated, and air pollution is increasingly serious. The quantity of the outdoor air quality monitoring sites is insufficient.In this paper, a new type of outdoor air quality monitoring system is studied and preliminarily practiced, and proven has certain feasibility and applicability. The main contributions of this paper are: firstly, we improve the network layout by employing the Zigbee network, which is combined with factory characteristics, and collected data on Carbonic Oxide, Nitrogen Dioxide, Sulfur Dioxide, Ozone, Particulate Matter, temperature, and humidity. And then, to establish the dilution coefficient and diffusion coefficient of pollution diffusion.

[4]. Xinghan Xu and Minoru Yoneda,”Multitask Air-Quality Prediction Based on LSTM-Autoencoder Model”.With the development of the data-driven modeling techniques, using the neural network to simulate the transport process of atmospheric pollutants and constructing PM2.5 timeseries prediction model have become a hot topic. In response to this problem, this article proposes a long short-term memory (LSTM) autoencoder multitask learning model to predict PM2.5 time series in multiple locations city wide. The model could implicitly and automatically excavate the intrinsic relevance among the pollutants in different stations. And the meteorological information from the

monitoring stations is fully utilized, which is beneficial for the performance of the proposed model.

[5]. Ke Gu, Junfei Qiao, Member, IEEE, and Weisi Lin, Fellow, IEEE,”Recurrent Air Quality Predictor Based on Meteorology- and Pollution- Related Factors”.Air quality is currently arousing drastically increasing attention from the governments and populace all over the world. In this paper, we propose a heuristic recurrent air quality predictor (RAQP) to infer air quality. The RAQP exploits some key meteorology- and pollution related variables to infer air pollutant concentrations, e.g. the fine particulate matter (PM2.5). It is natural that the meteorological factors and air pollutant concentrations at the current time have strong influences on air quality the next adjacent moment, that is to say, there exist high correlations between them.

[6]. Zhongang Qi, Tianchun Wang, Guojie Song, Weisong Hu, Xi Li∗ , Zhongfei (Mark) Zhang,”Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality”.The interpolation, prediction, and feature analysis of fine-gained air quality are three important topics in the area of urban air computing. The solutions to these topics can provide extremely useful information to support air pollution control, and consequently generate great societal and technical impacts. Most of the existing work solves the three problems separately by different models.

[7]. Maryam Ghodrati, Member, IEEE, Ali Farmani, Member, IEEE, and Ali Mir, Member, IEEE,Nanoscale Sensor-based Tunneling Carbon Nanotube Transistor for Toxic Gases Detection: A First-Principle Study[2019].A wireless safety device for gas leakage detection is proposed. The device is intended for use in household safety where appliances and heaters that use natural gas and liquid petroleum gas (LPG) may be a source of risk. The system also can be used for other applications in the industry or plants that depend on LPG and

natural gas in their operations. The system design consists of two main modules: the detection and transmission module, and the receiving module.

[8]. Hui Min Kim 1, Chang Geun Heo 1, Sung Ho Cho2, and Gwan Soo Park,”Determination Scheme for Accurate Defect Depth in Underground Pipeline Inspection by Using Magnetic Flux Leakage Sensors”.The magnetic flux leakage-type nondestructive testing has been applied for the highly efficient inspection of defects in ferromagnetic materials such as underground gas pipelines. In the system, the magnetic field is applied to magnetize a steel pipe so as to induce the leakage signal in the vicinity of defects on the pipe. From the measured sensor signals, decomposing or estimating sizes and shapes of defects are necessary.

[9]. Ravi Kishore Kodali,Greeshma, R.N.V.,Kusuma Priya Nimmanapalli,IOT Based Industrial Plant Safety Gas Leakage Detection System.Most of the fire- breakouts in industries are due to gas leaks. These cause dreadful damage to the equipment, human life leading to injuries, deaths, and environment. Currently available leakage detectors warn the people around using on-site alarms. So, this project proposes a leakage detector which sends the warning to the concerned people through SMS . This detector senses the presence of harmful gases particularly, LPG, Methane and Benzene. LPG and Methane gases catch fire easily resulting in blasts.

[10]. Sourabh Jamadagni, Priyanka Sankpal,Shwetali Patil,”Gas Leakage and Fire Detection using Raspberry Pi”[2019].This paper presents the growth in the industrial monitoring system’s design using Internet of Things (IoT). The sensor used for the development of this system is MQ-2 which detects the leak age of gas at any atmospheric condition and fire sensor as a simple and compact device for protection against fire.

[11]. Anindya Nag, Asif Iqbal Zia, Member, IEEE, Xie Li, Subhas Chandra Mukhopadhyay, Fellow, IEEE, and Jürgen Kosel,”Novel Sensing Approach for LPG Leakage Detection: Part I—Operating Mechanism and Preliminary Results”.Gas sensing technology has been among the topical research work for quite some time. This paper showcases the research done on the detection mechanism of leakage of domestic cooking gas at ambient conditions. Micro- electro mechanical systems-based interdigital sensors were fabricated on oxidized single-crystal silicon surfaces by the maskless photolithography technique. The electrochemical impedance analysis of these sensors was done to detect liquefied petroleum gas (LPG) with and without coated particles of tin oxide (SnO2) in form of a thin layer.

[12]. Maurizio Rossi, Student Member, IEEE, and Davide Brunelli, Member, IEEE,”Autonomous Gas Detection and Mapping With Unmanned Aerial Vehicles”.Unmanned aerial vehicles (UAVs) are nowadays largely employed in civil applications. One of the most promising applications is the environmental monitoring (or risk assessment). We propose a battery-powered eNose board that can be embedded with any type of drone.

[13]. Nadeem Pasha1, Varun Koushik K. M2, Shahbaaz Ahmed P3, (International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, 2017).The analysis and implementation of the home automation technology using Global System for Mobile Communication (GSM) modem to control home appliances such as light, conditional system, and security system via Short Message Service (SMS) text messages is presented in this Project. Security in the form of LPG gas leakage alert, entrance of people without permission alert is added for enhancement of security in house.The concept of serial communication and AT-commands has been applied towards development of the smart GSM-based home automation system.

[14]. Menachem Domb (IoT and Smart Home Automation, 2018).Smart home systems achieved great popularity in the last decades as they increase the comfort and quality of life. Most smart home systems are controlled by smartphones and microcontrollers. A smartphone application is used to control and monitor home functions using wireless communication techniques. We explore the concept of smart home with the integration of IoT services and cloud computing to it, by embedding intelligence into sensors and actuators, networking of smart things using the corresponding technology, facilitating interactions with smart things using cloud computing for easy access in different locations, increasing computation power, storage space and improving data exchange efficiency.

[15]. Dipanjan Bhattacharjee, Purva Bhatnagar, Sushabhan Choudhury (International Journal of Computer Applications (0975 – 8887) Volume 31– No.9, October 2011).The advancement of smart sensor technology has allowed us to design and development of a flexible reliable smart gas detection system to detect gases such as combustible air contaminants in the environment. The system composed of three modules the base station, wireless sensor array

# CHAPTER – 3 SYSTEM ANALYSIS

* 1. **EXISTING SYSTEM**

Urban vehicle emission prediction can help the regulation of vehicle pollution and traffic control. However, it is hard to predict the spatiotemporal variation of vehicle emission because of the spatial interactions and temporal correlations between different road segments as well as the high nonlinearity and complexity of vehicle emission variation. The existing methods solve the problem by splitting the region into standard segments or grids based on conventional deep learning methods, without considering that urban vehicle emission varies by graph-structured traffic road network and depends on many complex external environment factors. To address these issues, a spatiotemporal graph convolution multifusion network (ST-MFGCN) is proposed to leverage the graph structural properties.The proposed model consists of three parts: 1) a spatiotemporal graph convolution module to capture spatiotemporal dependencies by merging closeness, period, and trend sequences with temporal convolution as well as graph convolution is introduced to model the spatial dependencies; 2) an external factor component to divide multisource external factors into global and individual external features; and 3) a general fusion component to merge the spatiotemporal patterns and the external features as well as fit the mutation of emission measurement data by multifusion strategy. Finally, the proposed model is evaluated on the practical monitoring data of vehicle emission data in Hefei, and the results demonstrate that our proposed model can predict regional vehicle emissions effectively.

# PROPOSED SYSTEM

Emission control system, in automobiles, means employed to limit the discharge of noxious gases from the internal-combustion engine and other components. An increased diesel engine population has created pressures on controlling diesel PM and NOx emissions. The initial progress in diesel emission control was achieved through engine technologies, including changes in the combustion chamber design, improved fuel systems, charge air cooling, and special attention to lube oil consumption. Emission standards implemented in the 2005- 2010 timeframe additionally require the use of exhaust after treatment methods on new diesel engines. These methods include diesel particulate filters, urea- SCR catalysts, and NOx adsorbers. The combustion system includes the combustion chamber, its shape and characteristics such as charge composition, charge motion, and fuel distribution. This is where pollutants such as NOx, CO and PM are created as well as where incomplete oxidation of fuel occurs. To avoid such of toxic pollutants, we will develop a system to monitor the emission of the vehicle. We will be using microcontroller such as AVR will be used to process the readings from the air pollution sensor, GPS, wifi gateways for pushing data to the mongodb database using application program interface. A web application will be developed to monitor the abnormal emission of the vehicle. A mobile application will be developed using react native to get the abnormal emitting vehicle information. Thus this project will be useful to identify the pollution emitting vehicle easily and can control the air pollution.

* 1. **REQUIREMENTS ANALYSIS AND SPECIFICATIONS**
     1. **Input Requirements:**

**Power Supply -** A regulated power supply of 5v or 12v must ensure the operation of smoke sensor Arduino kit.

**Air Pollution Sensor** - MQ2 smoke sensor are required which is suitable for detecting the gases emitted from the vehicle. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible. The sensitivity of the sensor can also be adjusted.

* + 1. **Output Requirements:**

**ESP8266 –** ESP8266 chip is required which is a gateway module used to collect the data from the microcontroller and to push those values to Web Application ,this chip is needed.

**Web Application –** Web Application is used to check those data’s received from gateway module and it will check whether the values are normal or abnormal,then these information will be stored in the Mongo-DB Database.In short,the emission value of the vehicles will be known.

**Mobile Application** – Mobile Application is needed in order to track the vehicle using GPS.Once the value is found to be abnormal in the web app,the details of the particular vehicle like the vehicle number,where the vehicle is roaming will be shared to the nearby police via this app.

* + 1. **Functional Requirements:**

**AVR Microcontroller –** AVR Microcontroller is a processing module which is responsible for the entire function of the system.It is useful in collecting the data from the air pollution sensor. It supports the data up to eight (8) bits. ATmega- 328 has 32KB internal built-in memory. This micro-controller has a lot of other characteristics.

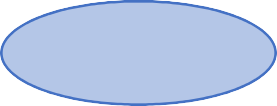
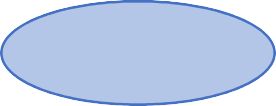
* 1. **TECHNOLOGY STACK Hardware Requirements:**
* Pc
* Power Supply
* Air Pollution Sensor
* GPS
* Micro Controller
* Wi-Fi Module
* RAM : 8 GB
* Processor : i5
* Hard disk : 1 TB

# SOFTWARE REQUIREMENTS:

* Visual Studio
* Python

# CHAPTER – 4 SYSTEM DESIGN

* 1. **ER DIAGRAM**



**LOGIN**

**ADMIT**

**TRACK**

**LOCATION**

**THRESHOLD**

**CROSSED**

YES

NO

**IGNORE**

**SEND GPS DATA**

**POLICE APP**

**SEND ALERT**

**GENERATES REPORT OF VEHICLE POLLUTION**

**ADMIN**

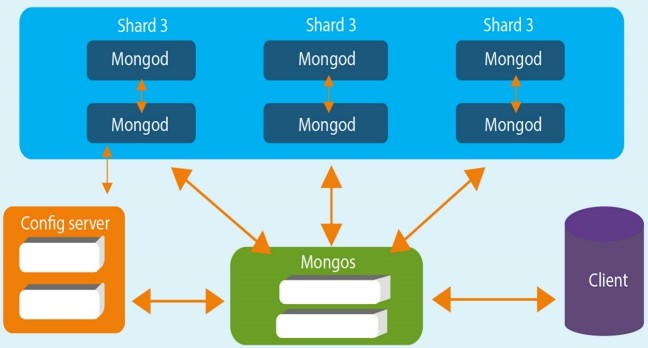
**USER VEHICLE DATA**

**LOGIN**

# Fig 4.1 ER Diagram

* 1. **Data Dictionary**

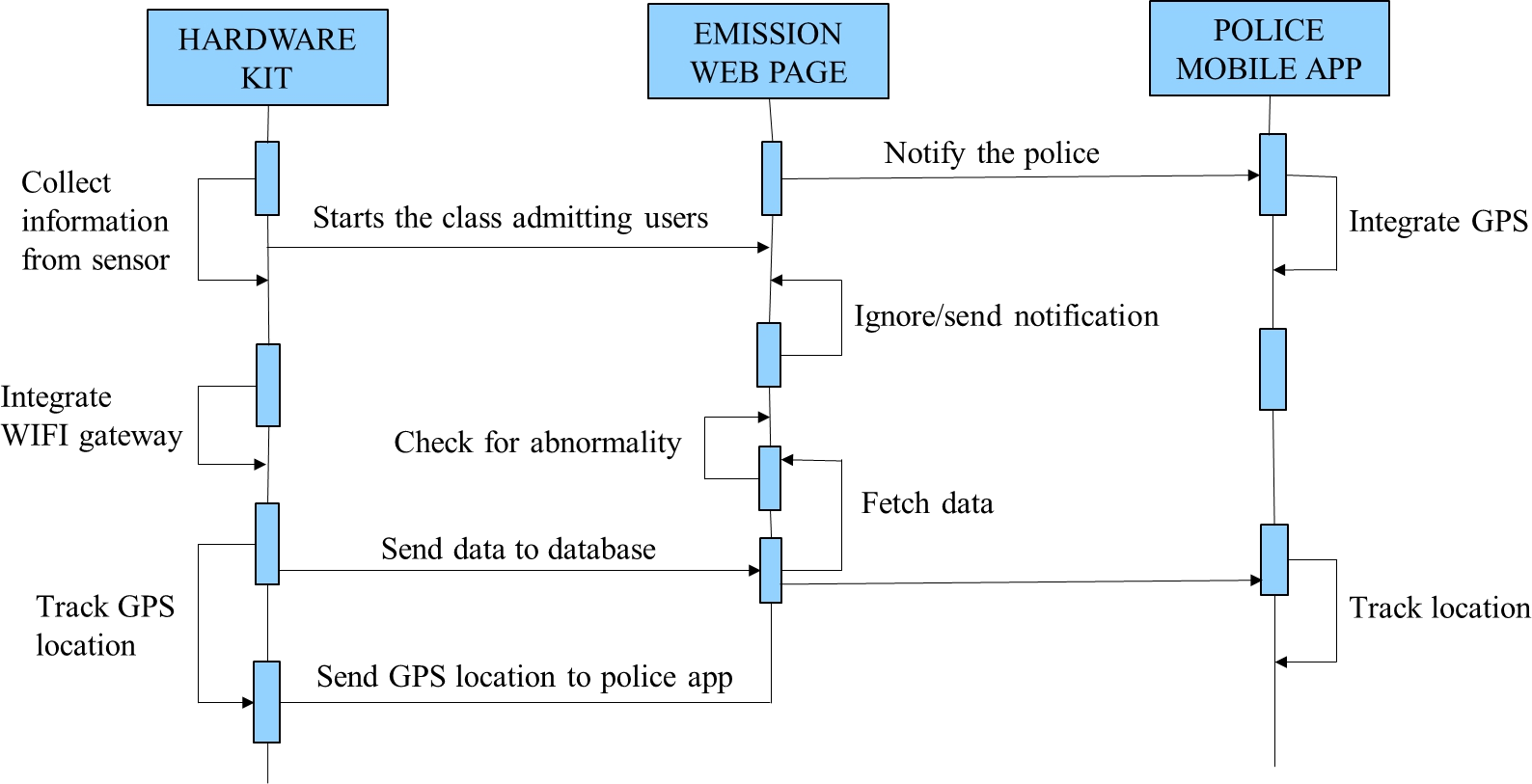
MongoDB ,an open-source cross-platform **document-oriented database program** is used for database management. Database is a **physical container for collections**. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.



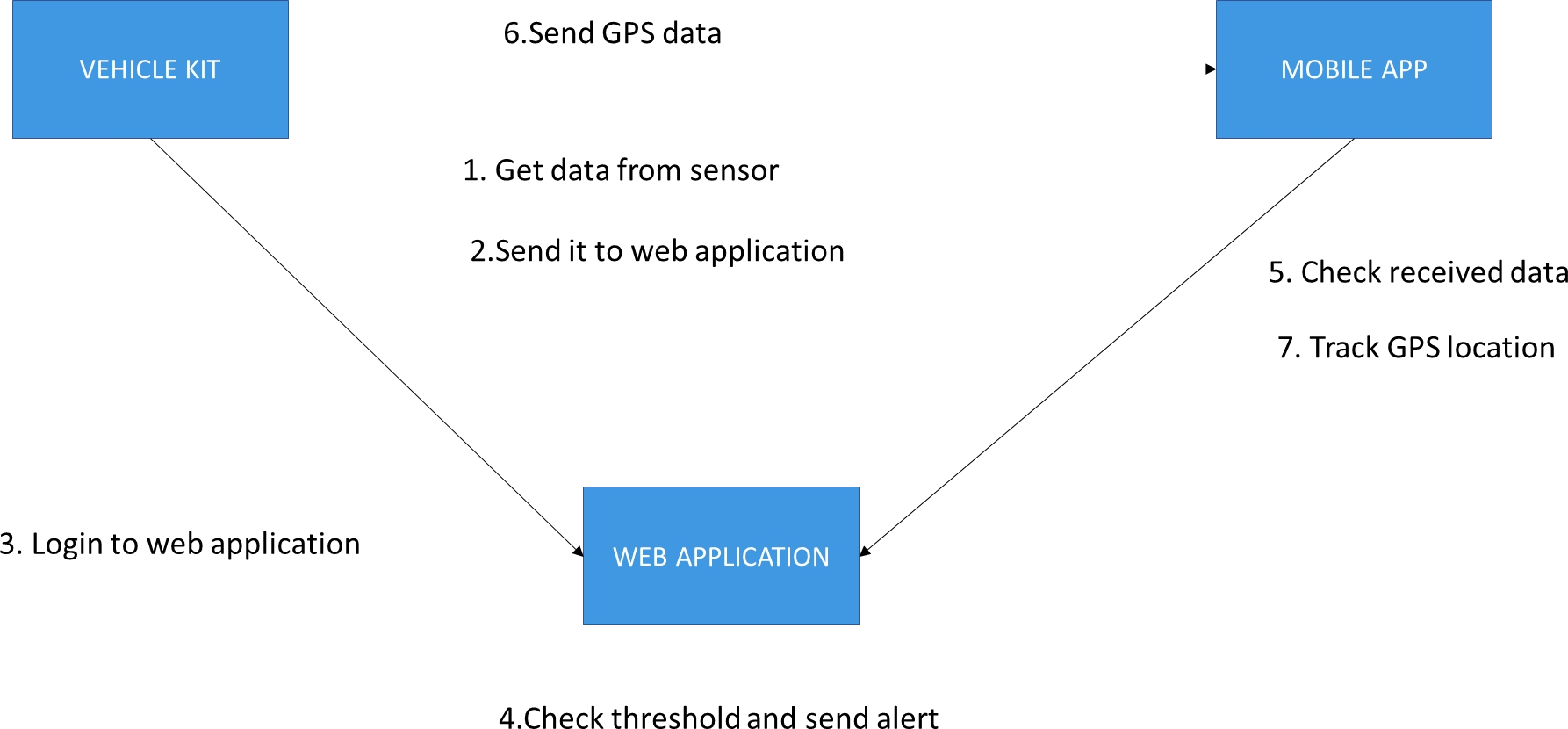
# Fig 4.2 Database Design

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

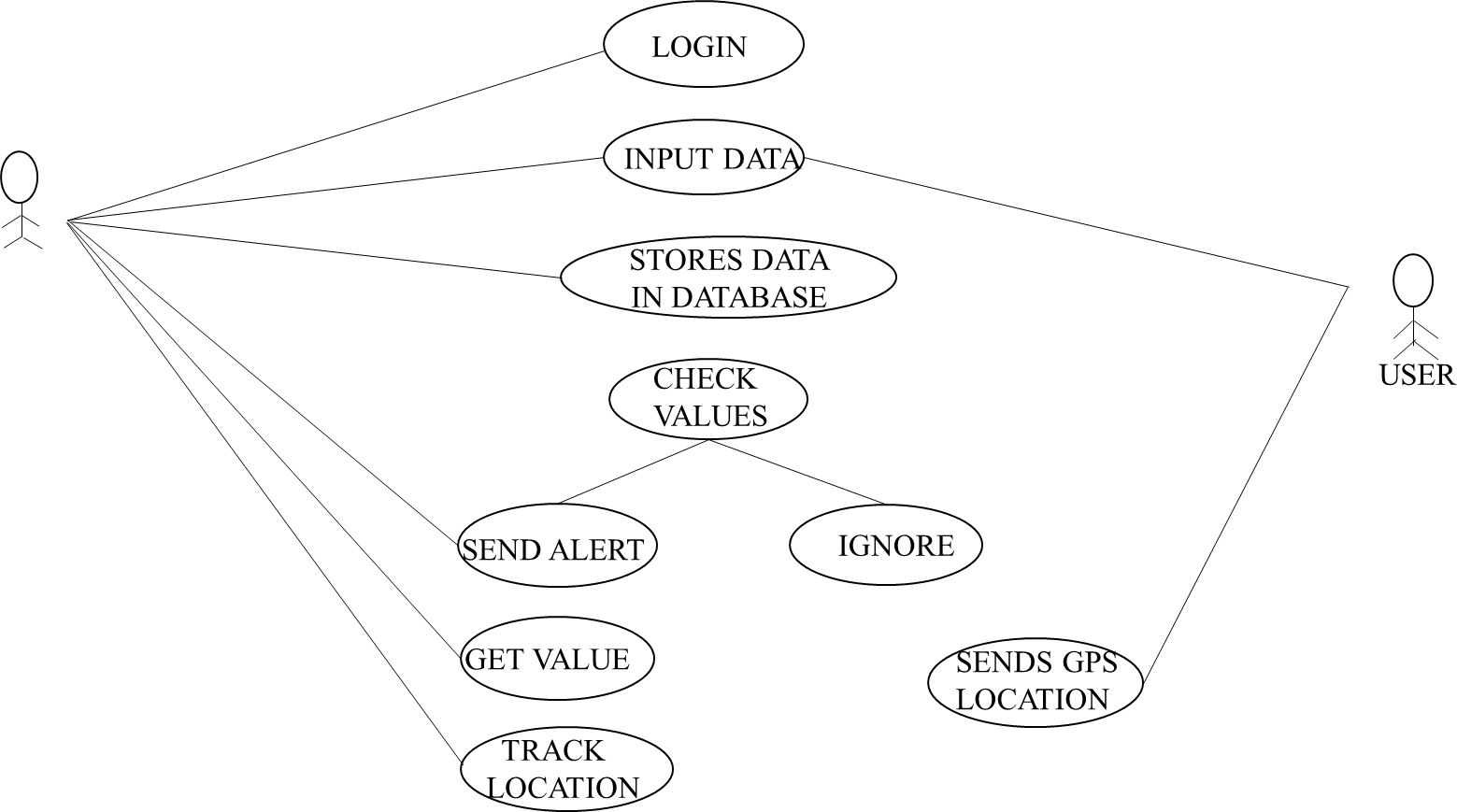
# 4.4 UML Diagrams



**Fig 4.3 Sequence Diagram**



# Fig 4.4 Collaboration Diagram

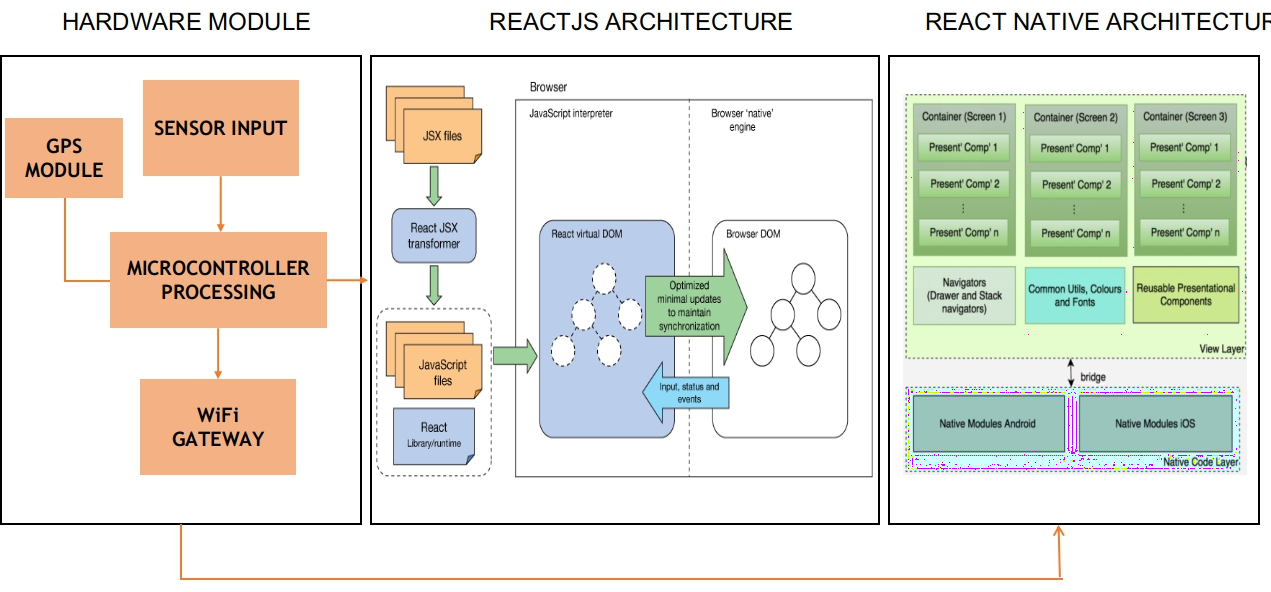


**Fig 4.5 Use-Case Diagram**

# CHAPTER – 5 SYSTEM ARCHITECTURE

# Architecture Overview

An intelligent emission control system which can automatically recognize the pollution emitting vehicle through various sensors and microcontroller.A web application will be developed to monitor the abnormal emission of the vehicle.A mobile application will be developed using react native to get the abnormal emitting vehicle information and will be helpful in tracking the pollution emitting vehicle.A certificate will be automatically generated and send through mail.Thus this project will be useful to identify the pollution emitting vehicle easily and can control the air pollution.



# Fig 5.1 System Architecture

**ReactJS Architecture**

React also allows us to create reusable UI components. React was first created by Jordan Walke, a software engineer working for Facebook. React first deployed on Facebook’s newsfeed in 2011 and on Instagram.com in 2012.React allows developers to create large web applications which can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in application. This corresponds to view in the MVC template. It can be used with a combination of other JavaScript libraries or frameworks, such as Angular JS in MVC. JSX stands for JavaScript XML. It is an XML/ HTML like syntax used by React. It extends the ECMAScript so that XML/ HTML like text can co-exist along with JavaScript react code. This syntax is used by the pre-processors like *Babel* to transform HTML like text found in JavaScript files into standard JavaScript objects. With JSX, we can go a step further by again embedding the HTML code inside the JavaScript. This makes HTML codes easy to understand and boosts JavaScript’s performance while making our application robust.



# Fig 5.2 ReactJS Architecture

**React Native Architecture**

Cross-platform development has become a great alternative to fully native mobile app development. Following the native mobile development approach,

you create separate apps for Android and iOS. The cross-platform development allows you to cut expenses and save time by using the same code across both platforms. The React Native framework is a rising mobile solution and is considered the future of cross-platform mobile app development.React Native is an open-source framework that allows to build a mobile app with only JavaScript. It was introduced by Jordan Walke, a Facebook software engineer, as a new technology for simpler development and a better user experience. The main distinctive of this framework is that React Native apps function just like native apps. They don’t differ from apps built on Java, Objective-C or Swift and they use the same UI building blocks as native iOS or Android apps. But with React Native, building a mobile app is much faster and less expensive.

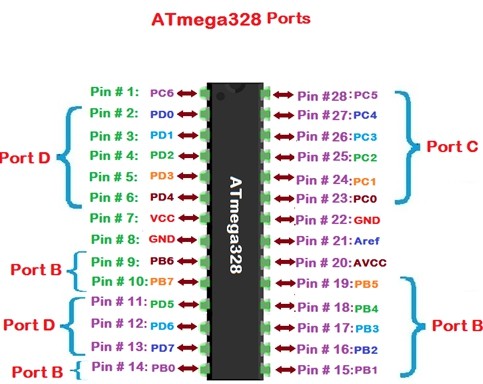
# Module Design Specification

There are about six modules used in the development of the project.They are as follows:

# Processing Module

ATmega-328 is basically an Advanced Virtual RISC (AVR) micro-controller. It supports the data up to eight (8) bits. ATmega-328 has 32KB internal built- in memory. This micro-controller has a lot of other characteristics. You should also have a look at Introduction to PIC16F877a (it's a PIC Microcontroller) and then compare functions of these two Microcontrollers. ATmega-328 is basically an Advanced Virtual RISC (AVR) micro-controller. It supports the data up to eight (8) bits. ATmega-328 has 32KB internal built-in memory.This property shows if the electric supply supplied to the micro-controller is removed, even then it can store the data and can provide results after providing it with the electric supply. Moreover, ATmega-328 has 2KB Static Random Access Memory (SRAM). Other characteristics will be explained later. ATmega 328 has several different features which make it the most popular

device in today's market. These features consist of advanced RISC architecture, good performance, low power consumption, real timer counter having separate oscillator, 6 PWM pins, programmable Serial USART, programming lock for software security, throughput up to 20 MIPS etc. ATmega-328 is mostly used in Arduino.These features consist of advanced RISC architecture, good performance, low power consumption, real timer counter having separate oscillator, 6 PWM pins, programmable Serial USART, programming lock for software security, throughput up to 20 MIPS etc. ATmega-328 is mostly used in Arduino.

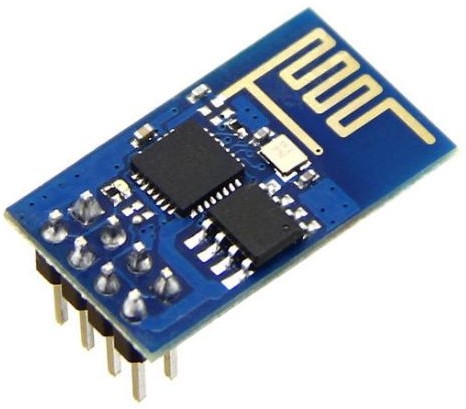


# Fig 5.3 Processing Module

* + 1. **Gateway Module**

The ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack and MCU (microcontroller unit) capability produced by Shanghai-based Chinese manufacturer, Espresso. The chip first came to the attention of western makers in August 2014 with the ESP-01 module, made by a third-party manufacturer, AI-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes- style commands. The ESP8285 is an ESP8266 with 1 MiB of built-in flash, allowing for single-chip devices capable of connecting to Wi-Fi. The successor to these microcontroller chips is the ESP32. ESP8266 (presently ESP8266EX) is a chip with which manufacturers are making wirelessly networkable micro- controller modules. More specifically, ESP8266 is a system-on-a-chip (SoC)

with capabilities for 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2), general-purpose input/output (16 GPIO), Inter-Integrated Circuit (I²C), analog- to-digital conversion (10-bit ADC), Serial Peripheral Interface (SPI), I²S interfaces with DMA (sharing pins with GPIO), UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and pulse-width modulation (PWM).



# Fig 5.4 ESP8266 Module

* + 1. **Database Integration**

MongoDB is a cross-platform, document-oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

# Database

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

# Collection

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not

enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

# Document

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.The following table shows the relationship of RDBMS terminology with MongoDB.



# Fig 5.5 MongoDB Architecture

* + 1. **Web Development**

ReactJS is JavaScript library used for building reusable UI components. According to React official documentation, following is the definition − React is a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time.

# JSX

JSX stands for JavaScript XML. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React.It is faster than normal JavaScript as it performs optimizations while translating to regular JavaScript. It makes easier for us to create templates. Instead of separating the

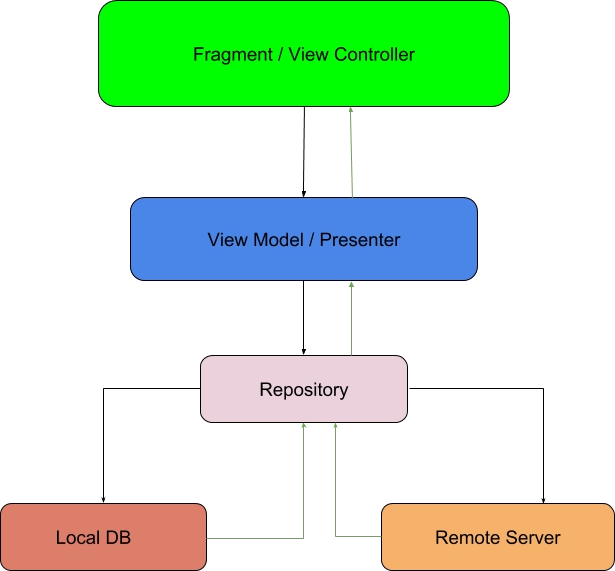
markup and logic in separated files, React uses *components* for this purpose. We will learn about components in details in further articles.

# Components

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and returns HTML via a render function.Components come in two types, Class components and Function components, in this tutorial we will concentrate on Class components.

# Mobile App Development

In this project react native is used for mobile app development.React Native is a framework that builds a hierarchy of UI components to build the JavaScript code. It has a set of components for both iOS and Android platforms to build a mobile application with a native look and feel. Mobile development has witnessed unprecedented growth. According to statistics, mobile applications will generate an estimated 188 billion U.S. dollars in revenue via app stores, advertising and in-app purchases by the year 2020. Single and business users require high- standard apps with good design. React Native seems to be a viable solution for building high-quality apps in a short time with the same performance and user- experience standards that native apps provide.



# Fig 5.6 Mobile App Development

* + 1. **Email Integration**

For integrating E-mail, the SMTP protocol is used for sending and receiving mail as configured by us. The work flow of that module can be seen as follows.

# Simple Mail Transfer Protocol (SMTP)

SMTP is the standard protocol for providing email services on a TCP/IP network. This server provides the ability to receive and send email messages.SMTP is an application-layer protocol that provides the delivery and transmission of email over the Internet. It is maintained by the Internet Engineering Task Force (IETF). SMTP is generally summed within an email client application and is composed of four key components:

* + - 1. Local user or client-end utility known as the mail user agent (MUA)
      2. Server known as mail submission agent (MSA)
      3. Mail transfer agent (MTA)
      4. Mail delivery agent (MDA)

Getting email alerts or set of data using raspberry pi python program is very useful application. All that is needed is smtplib library in the python script. There are many version of python but pi is more compatible with 3.2 and 2.7 version of it. Below are the mentioned steps of sending SMTP email using pi:

# Steps for Sending Email using Raspberry Pi

**Step 1:-** Setting up the raspberry pi module- connect the power cable and LAN cable to raspberry pi then create WIFI hotspot and connect with it.

**Step 2:-** After then open the terminal window on Pi. Then, open the putty software and paste the host name or ip address.

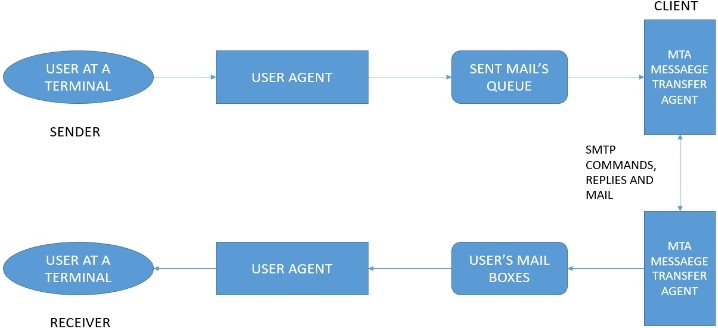
**Step 3:-** We need to update the Raspberry Pi. So, install the latest packages by using the below command.

**Step 4:-** Then use the following command – echo “hello” | mail –s “test” [xyz@gmail.com.This](mailto:xyz@gmail.com.This) command specifies the content, subject of the mail, as well as the mail id to which the mail will be delivered.

**Step 5:-** Then we need to create a new file in the python and this can be done by using the following command- nano newmailing.py

**Step 6:-** Allowing Gmail SMTP Access for Accounts with Standard Authentication.To allow access to Gmail’s SMTP server from your app

**Step 7:-** Login to the Gmail account and check the mail, if everything works correctly then a mail will be delivered to the mentioned mail id.An overall flow of the working can be seen in the figure below



# Fig 5.7 Email Integration

* 1. **PROGRAM DESIGN LANGUAGE(Pseudocode)**

# Visual Studio

In this project the Microsoft visual studio is used as an IDE. Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.Visual Studio Code supports macOS, Linux, and Windows - so you can hit the ground running, no matter the platform.At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use.For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.VS Code includes enriched built-in support for Node.js development with JavaScript and TypeScript, powered by the same underlying technologies that drive Visual Studio. VS Code also includes great tooling for web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON. Visual Studio Code includes a public extensibility model that lets developers build and use extensions, and richly customize their edit-build-debug experience.

# Arduino

In this project the Arduino software is used.The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming

and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire,There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

* Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
* Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
* Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
* Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it.

# CHAPTER – 6

**SYSTEM IMPLEMENTATION**

# CLIENT-SIDE CODING Mobile App

import React, { useState, useEffect } from 'react'; import { Router } from 'react-router-dom'; import { createBrowserHistory } from 'history'; import { Chart } from 'react-chartjs-2';

import { ThemeProvider } from '@material-ui/styles'; import validate from 'validate.js';

import mqtt from 'mqtt'; import axios from 'axios';

import { chartjs } from './helpers'; import theme from './theme';

import 'react-perfect-scrollbar/dist/css/styles.css';

// import './assets/scss/index.scss';

import validators from './common/validators'; import Routes from './Routes';

const browserHistory = createBrowserHistory(); export default function App() {

useEffect(() => {

// if (useflag == 0) {

// setuseflag(1)

// console.log("MqTT called")

var client = mqtt.connect("ws://broker.hivemq.com:8000/mqtt"); client.on("message", (topic, payload) => {

var mqttvalue = payload.toString(); console.log("++++++++++++++++++")

console.log(mqttvalue) console.log("++++++++++++++++++") var splits = mqttvalue.split(","); console.log(splits[0]) console.log(splits[1]) console.log(splits[2]) console.log(splits[3]) console.log(splits[4]) console.log(splits[5])

if (splits[5] == "1") {

const admission\_create = { mobile\_number:splits[0], vehicle\_number: splits[1], air\_pollution\_level: splits[2], lat:splits[3],

lon:splits[4]

};

const options = {

url: 'http://192.168.1.112:5008/pollution\_api', method: 'POST',

headers: {

'Content-Type': 'application/json',

},

data: JSON.stringify(admission\_create

}

console.log (options) axios(options)

.then(response => {

if (response.status == 200 ) {

console.log ("777777777777777777777777777777")

console.log(response.data)

console.log ("777777777777777777777777777777")

}

}

}

});

client.on("connect", () => { client.subscribe("panimalar\_pollution");

// client.publish("1999", "Updated!"); console.log("++++++++++++++++++++Connected to MQTT

Broker.++++++++++++++++++++++++++++++++");

});

}, [])

return (

<ThemeProvider theme={theme}>

<Router history={browserHistory}>

<Routes />

</Router>

</ThemeProvider>

);

}

**Web App**

import React, { Component, Fragment } from "react";

import PushNotification from "react-native-push-notification"; import Login from "./src/Login";

import

{

SafeAreaView, StyleSheet, ScrollView, View,

Text, StatusBar, FlatList,

} from 'react-native'; import {

Header, LearnMoreLinks, Colors, DebugInstructions, ReloadInstructions,

} from 'react-native/Libraries/NewAppScreen';

export default class PushController extends Component { constructor(props) {

super(props); this.state = { pushData: [],

tokenvalue:'token is empty'

}

}

componentDidMount() { let self = this;

PushNotification.configure({

// (optional) Called when Token is generated (iOS and Android)

onRegister: function (token) {

// console.log("TOKEN:", token);

self.setState({ tokenvalue: token

});

// testing(token)

},

// (required) Called when a remote or local notification is opened or received

onNotification: function (notification) {

// console.log("NOTIFICATION:", notification);

// process the notification self.\_addDataToList(notification);

// required on iOS only (see fetchCompletionHandler docs: https://github.com/react-native-community/react-native-push-notification-ios)

// notification.finish(PushNotificationIOS.FetchResult.NoData);

},

// ANDROID ONLY: GCM or FCM Sender ID (product\_number) (optional - not required for local notifications, but is need to receive remote push notifications)

senderID: "681967282456",

// IOS ONLY (optional): default: all - Permissions to register. permissions: {

alert: true, badge: true, sound: true

},

// Should the initial notification be popped automatically

// default: true popInitialNotification: true,

/\*\*

* (optional) default: true
* - Specified if permissions (ios) and token (android and ios) will requested or not,
* -if not, you must call PushNotificationsHandler.requestPermissions() later

\*/

requestPermissions: true

});

}

\_renderItem = ({ item }) => (

<View key={item.title}>

<Text style={styles.title}>{item.custom\_title}</Text>

<Text style={styles.message}>{item.custom\_message}</Text>

</View>

);

\_addDataToList(data) {

let array = this.state.pushData; array.push(data);

this.setState({ pushData: array

});

}

render() { return (

<Login tokenvalue ={this.state.tokenvalue}/>

);

}

}

const styles = StyleSheet.create({ scrollView: {

backgroundColor: Colors.lighter,

},

listHeader: { backgroundColor: '#eee', color: "#222",

height: 44,

padding: 12

},

title: { fontSize: 18,

fontWeight: 'bold', paddingTop: 10

},

noData: { paddingVertical: 50,

},

noDataText: { fontSize: 14, textAlign: 'center',

},

message: { fontSize: 14,

paddingBottom: 15, borderBottomColor: "#ccc", borderBottomWidth: 1

},

engine: {

position: 'absolute', right: 0,

},

body: {

backgroundColor: Colors.white, paddingHorizontal: 20,

paddingVertical: 10,

},

sectionContainer: { marginTop: 32,

paddingHorizontal: 24,

},

sectionTitle: { fontSize: 24,

fontWeight: '600', color: Colors.black,

},

sectionDescription: { marginTop: 8,

fontSize: 18,

fontWeight: '400', color: Colors.dark,

},

highlight: { fontWeight: '700',

},

footer: {

color: Colors.dark, fontSize: 12,

fontWeight: '600',

padding: 4,

paddingRight: 12, textAlign: 'right',

},

});

import Login from "./src/Login"; import {

SafeAreaView, StyleSheet, ScrollView, View,

Text, StatusBar, FlatList,

} from 'react-native'; import {

Header, LearnMoreLinks, Colors, DebugInstructions, ReloadInstructions,

} from 'react-native/Libraries/NewAppScreen';

export default class PushController extends Component { constructor(props) {

super(props); this.state = { pushData: [],

tokenvalue:'token is empty'

}

}

componentDidMount() { let self = this;

PushNotification.configure({

// (optional) Called when Token is generated (iOS and Android) onRegister: function (token) {

// console.log("TOKEN:", token); self.setState({

tokenvalue: token

});

// testing(token)

},

// (required) Called when a remote or local notification is opened or received

onNotification: function (notification) {

// console.log("NOTIFICATION:", notification);

// process the notification self.\_addDataToList(notification);

// required on iOS only (see fetchCompletionHandler docs: https://github.com/react-native-community/react-native-push-notification-ios)

// notification.finish(PushNotificationIOS.FetchResult.NoData);

},

// ANDROID ONLY: GCM or FCM Sender ID (product\_number) (optional - not required for local notifications, but is need to receive remote push notifications)

senderID: "681967282456",

// IOS ONLY (optional): default: all - Permissions to register. permissions: {

alert: true, badge: true, sound: true

},

// Should the initial notification be popped automatically

// default: true popInitialNotification: true,

/\*\*

* (optional) default: true
* - Specified if permissions (ios) and token (android and ios) will requested or not,
* -if not, you must call PushNotificationsHandler.requestPermissions() later

\*/

requestPermissions: true

});

}

\_renderItem = ({ item }) => (

<View key={item.title}>

<Text style={styles.title}>{item.custom\_title}</Text>

<Text style={styles.message}>{item.custom\_message}</Text>

</View>

);

\_addDataToList(data) {

let array = this.state.pushData; array.push(data); this.setState({

pushData: array

});

}

render() { return (

<Login tokenvalue ={this.state.tokenvalue}/>

);

}

}

const styles = StyleSheet.create({ scrollView: {

backgroundColor: Colors.lighter,

},

listHeader: { backgroundColor: '#eee', color: "#222",

height: 44,

padding: 12

},

title: { fontSize: 18,

fontWeight: 'bold', paddingTop: 10

},

noData: { paddingVertical: 50,

},

noDataText: { fontSize: 14, textAlign: 'center',

},

message: { fontSize: 14,

paddingBottom: 15, borderBottomColor: "#ccc", borderBottomWidth: 1

},

engine: {

position: 'absolute', right: 0,

},

body: {

backgroundColor: Colors.white, paddingHorizontal: 20,

paddingVertical: 10,

},

sectionContainer: { marginTop: 32,

paddingHorizontal: 24,

},

sectionTitle: { fontSize: 24,

fontWeight: '600', color: Colors.black,

},

sectionDescription: { marginTop: 8,

fontSize: 18,

fontWeight: '400', color: Colors.dark,

},

highlight: { fontWeight: '700',

},

footer: {

color: Colors.dark, fontSize: 12,

fontWeight: '600',

padding: 4,

paddingRight: 12, textAlign: 'right',

},

});

# SERVER-SIDE CODING

const express = require('express') const eslint = require('eslint')

const bodyParser = require('body-parser') const cookieParser = require('cookie-parser') const pollution\_api = require('./pollution\_api')

var cors = require('cors')

const middleware = require('./middleware') const port = process.env.PORT || 5008 const app = express()

app.use(cors())

app.use(bodyParser.json({limit: '10mb', extended: true})) app.use(bodyParser.urlencoded({limit: '10mb', extended: true}))

app.use(cookieParser()) app.post('/pollution\_api',pollution\_api.createData)

app.get('/pollution\_api',pollution\_api.listData) app.post('/pollution\_api\_get\_mobile',pollution\_api.get\_with\_mobile) app.put('/pollution\_api/:id',pollution\_api.editUser)

app.use(middleware.handleValidationError) app.use(middleware.handleError) app.use(middleware.notFound)

const server = app.listen(port,"0.0.0.0", () => console.log(`Server listening on port ${port}`)

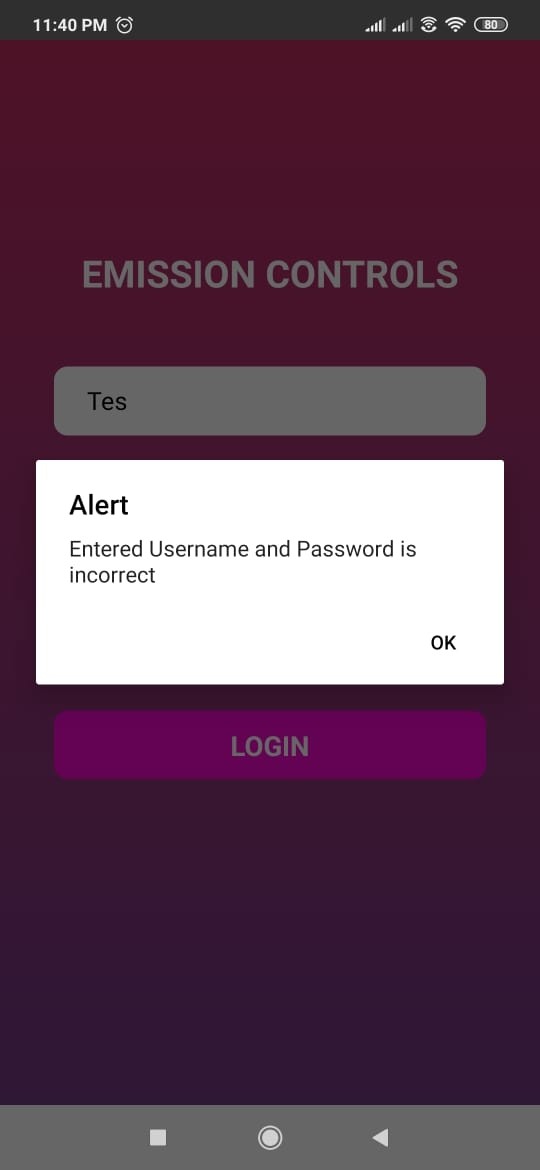
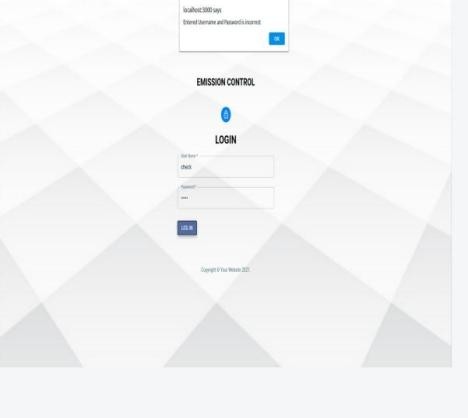
)

if (require.main !== module) { module.exports = server

# CHAPTER – 7 SYSTEM TESTING

**7.1 TEST CASES AND REPORTS**

If the user forgot his password, he/she must be able to reset password. The test for this scenario is to check if the username is in proper and credentials are validated. If user does not enter the correct username and password properly, then the error will be displayed.



# Fig 7.1 Invalid Login of Web App Fig 7.2 Invalid Login of Mobile App

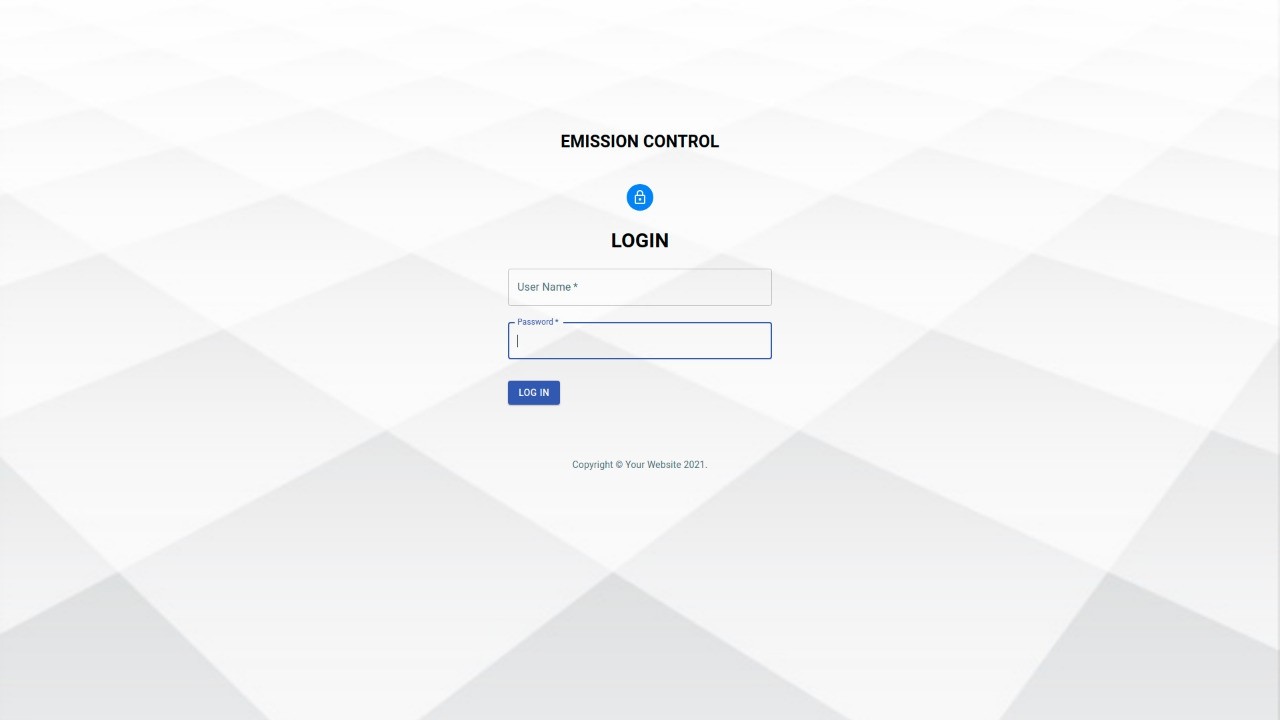
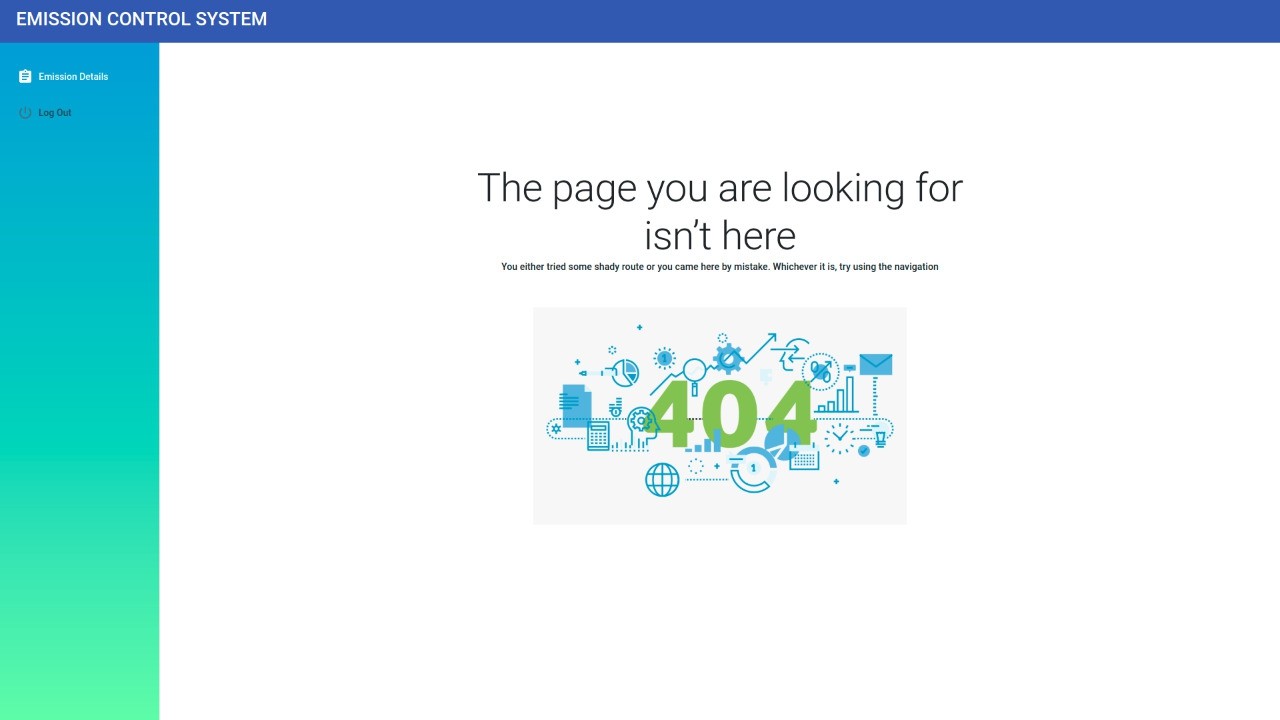
**Performance Analysis**

1. Initially, the kit is connected with air pollution sensor, microcontroller and GPS module and then power supply is given.



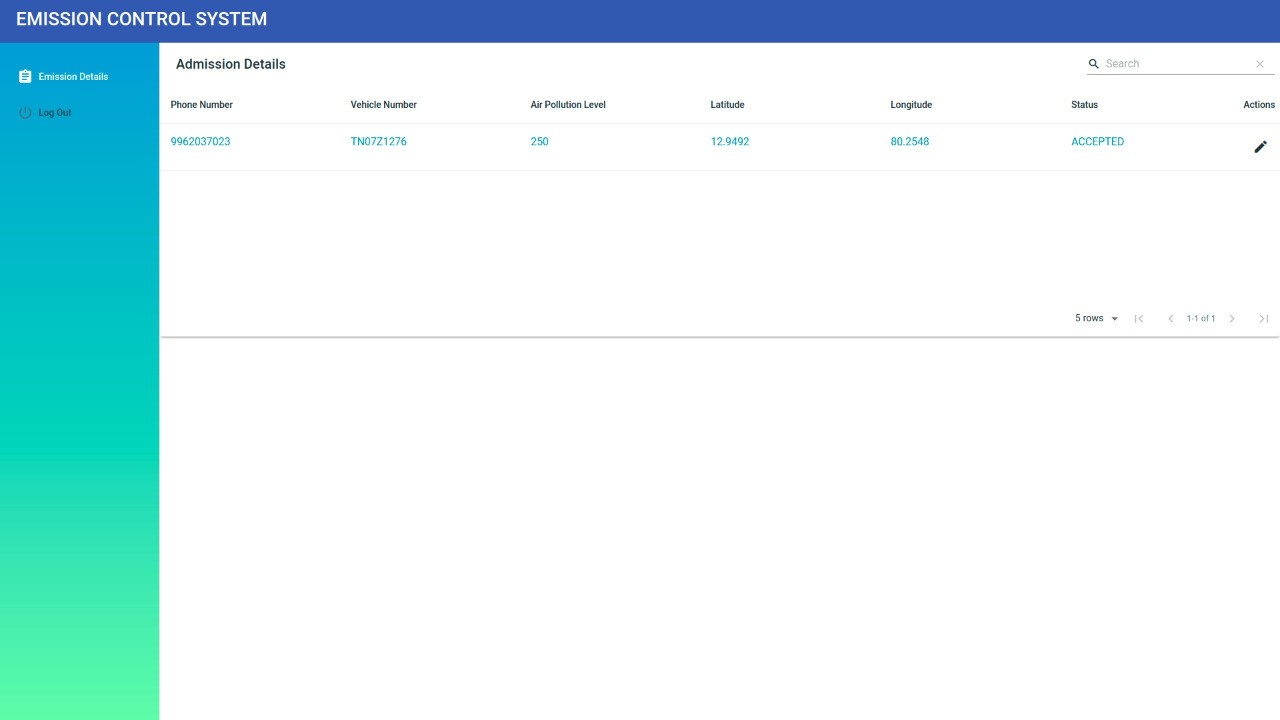
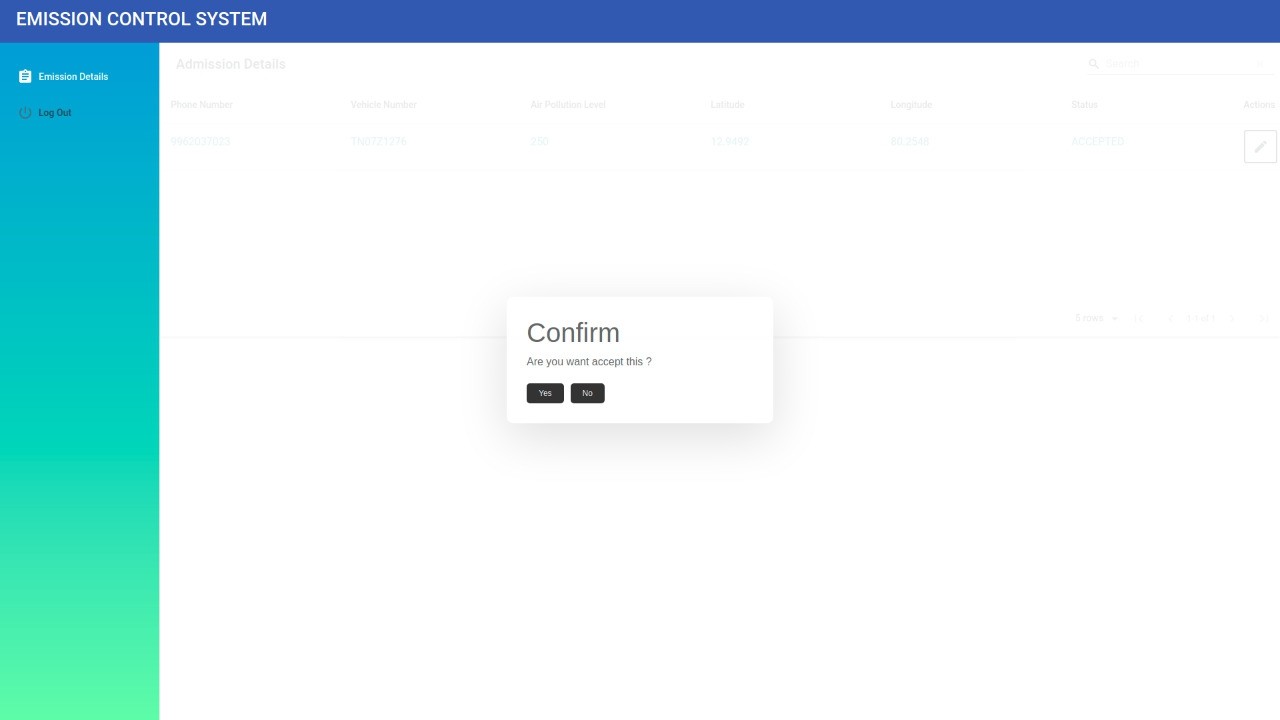
# Fig 7.3 Kit Connected with Sensor,AVR,GPS Module

1. The first image indicates the login page of the web application, the admin need to login with the account created and the image indicated when no entry found

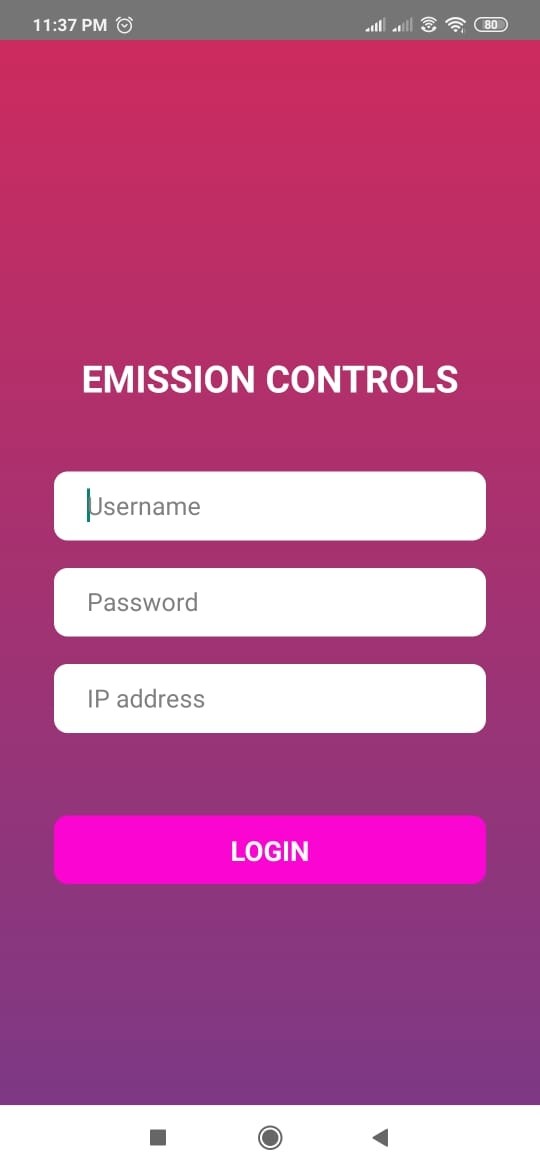
# Fig 7.4 Login Screen of Web App Fig 7.5 When no Data is Found

1. Once the gases are detected, those values will be stored in the web app to admin needs to accept or decline based on the values obtained.

# Fig 7.6 Data Entry Fig 7.7 Acceptance Details of Admin

1. Then the police needs to login into the mobile app developed and those abnormal values and data of the vehicle will be sent to that app.



# Fig 7.8 Login Screen of Mobile App Fig 7.9 User Tracking

**CHAPTER – 8 CONCLUSION**

# 8.1 CONCLUSION AND FUTURE ENHANCEMENTS

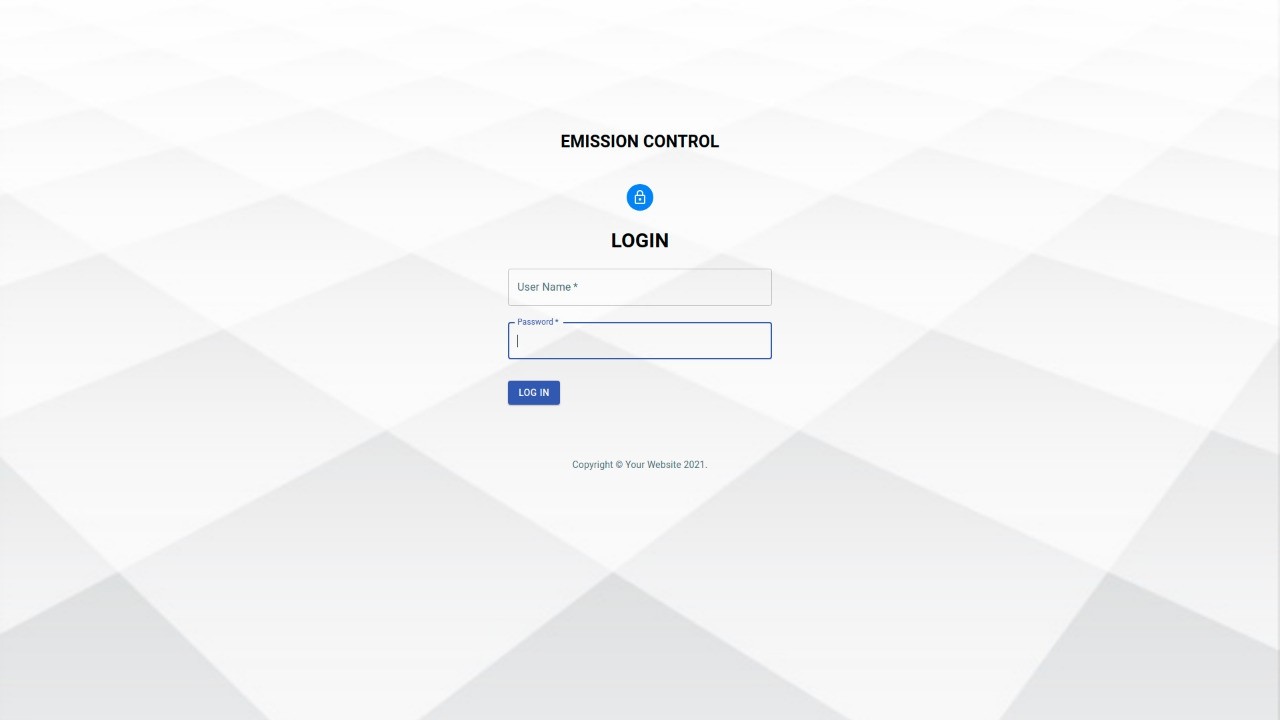
It is used to identify the level of NO2 emitted from vehicles. It makes all the certificates digital to achieve paperless work and also maintain authenticated and authorized process. Managing the centralized system in simple and efficient way. This is the most secured contribution to smart city. The IoT based emission test system is an efficient and reliable system that ensures that the data collected by the smoke sensors at the ETC is accurate and the integrity is maintained. The system addresses two important disadvantages of the existing system; firstly, the system induces automation and digitization. Secondly, the system ensures integrity and authenticity of data. This improved mechanism will help in controlling the amount of smoke emission through vehicles into the environment and gradually will help reduce the air pollution that is one of the major environmental issues.

# Future Scope

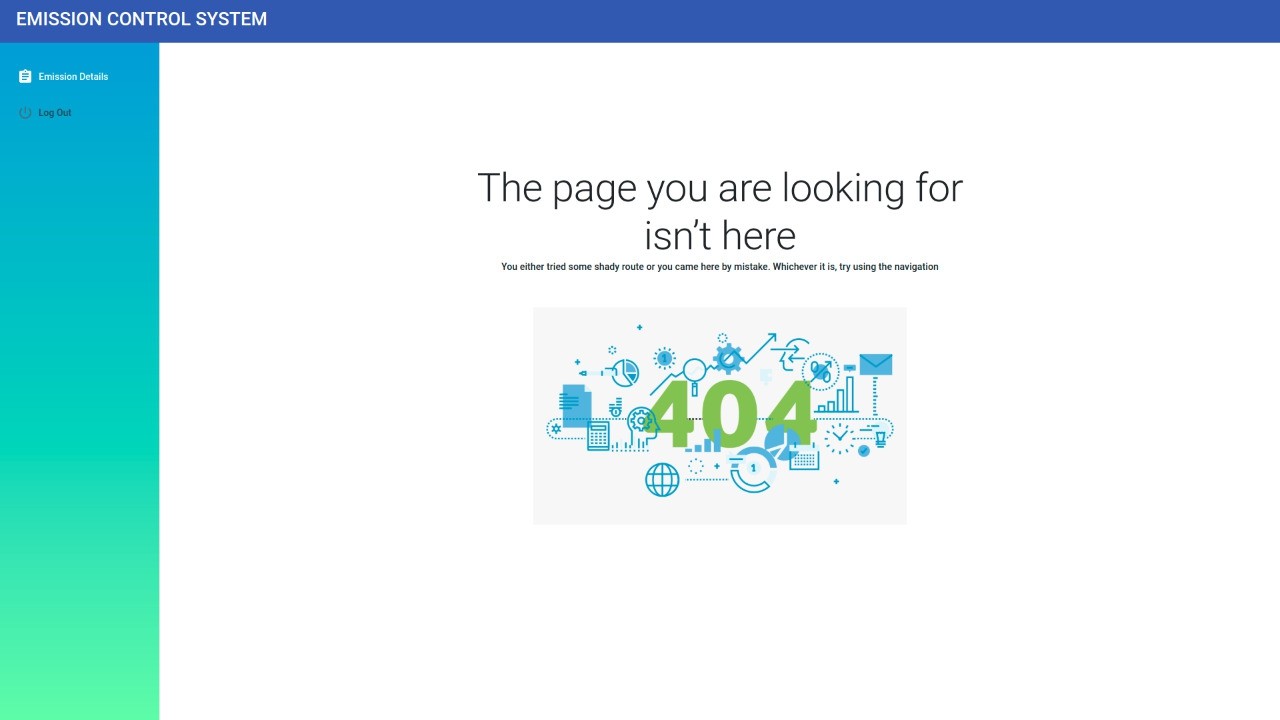
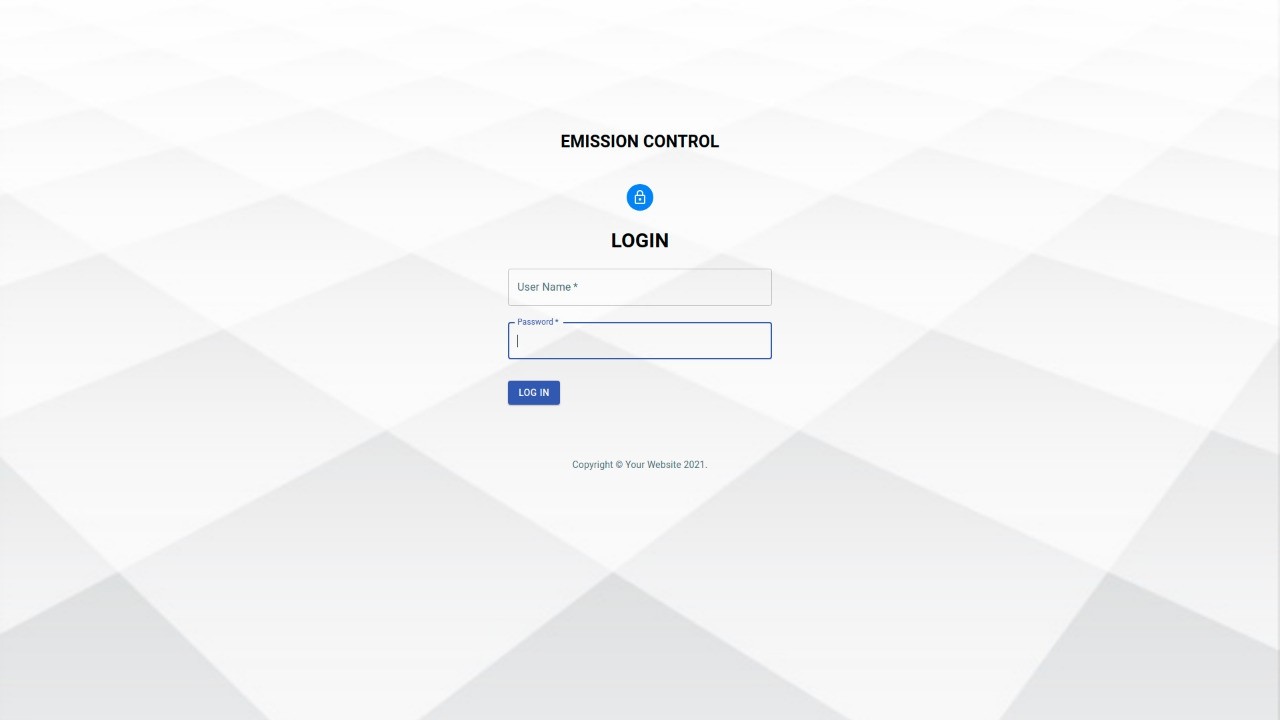
The proposed system has not implemented image processing at CA to verify the authenticity and hence requires an administrator to approve or reject the generation of the certificate. This feature can be considered for the future work.

# APPENDICES

# SAMPLE SCREENS



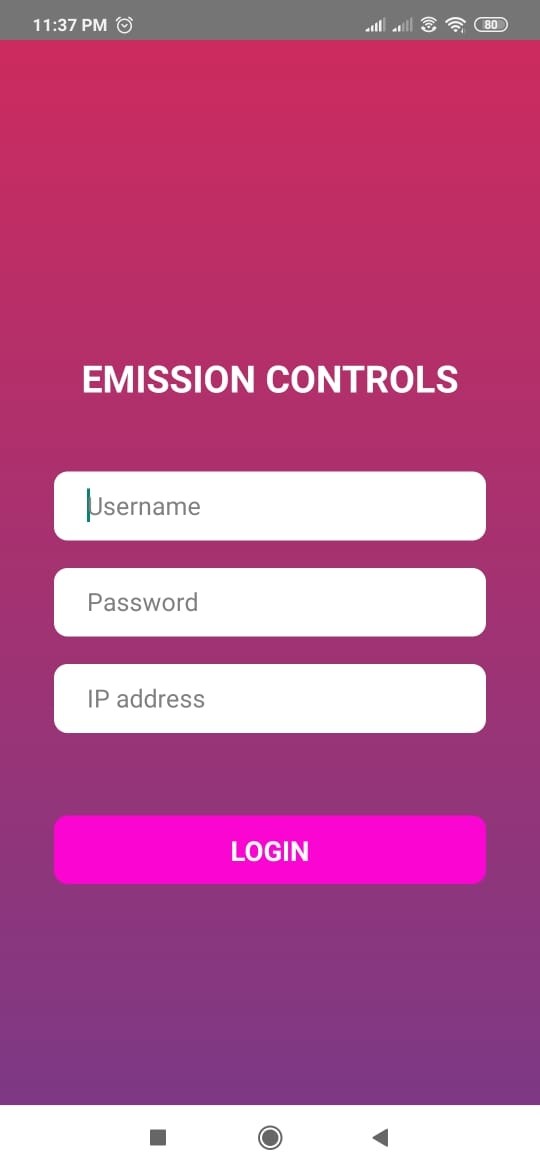
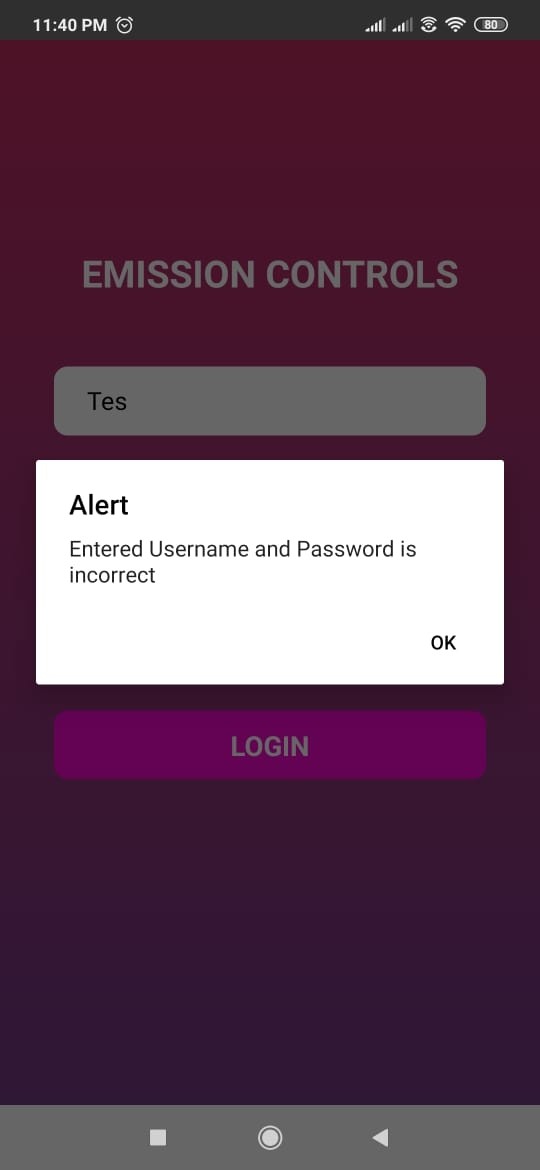
# Fig A1.1 Kit Connected with Equipments Fig A1.2 Login Page of Web App



**Fig A1.3 Invalid Login Fig A1.4 Data Unavailable**

# Fig

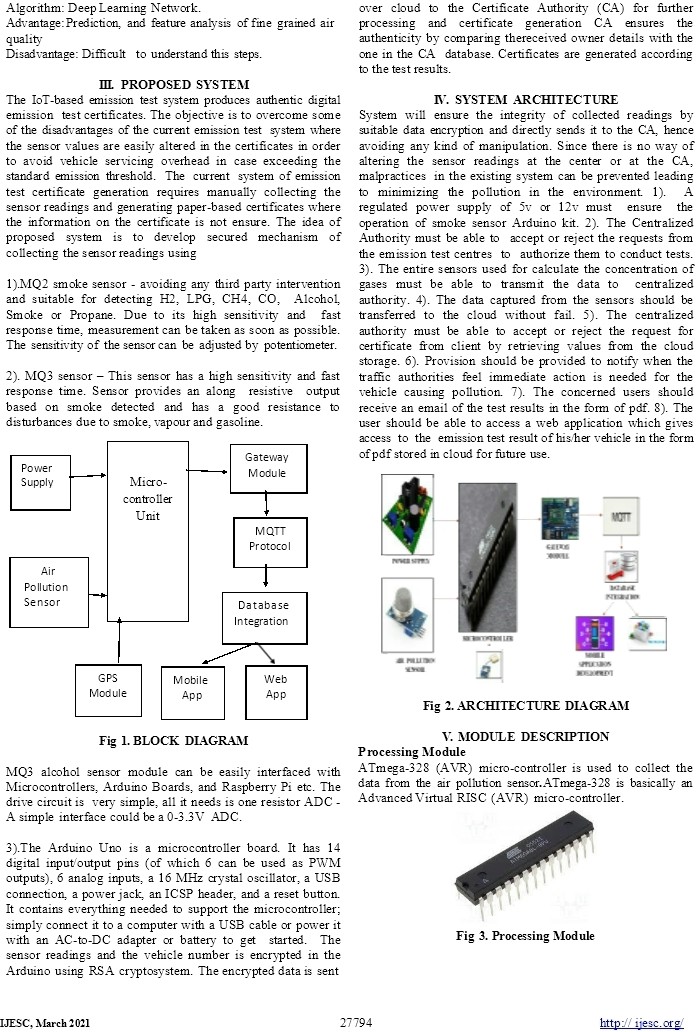
**A1.5 Data Entry into Web App Fig A1.6 Accept or reject**

# Fig A1.7 Login of mobile app Fig A1.8 Invalid login



**Fig A1.9 User vehicle tracking Using GPS Data Received**



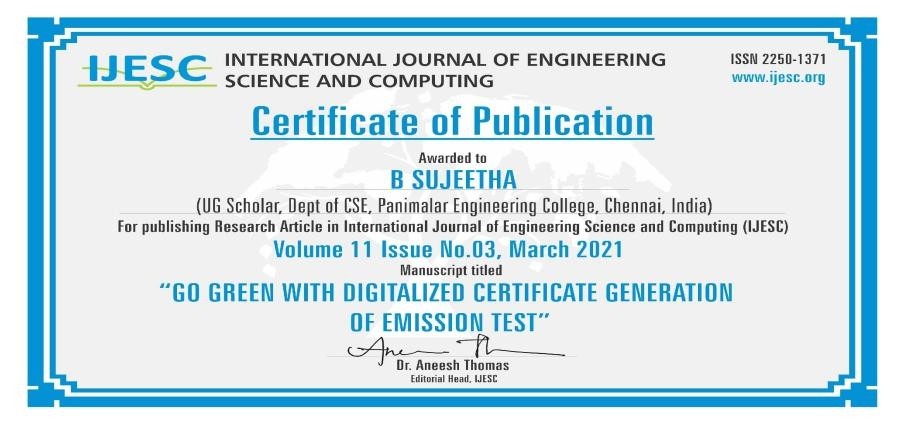
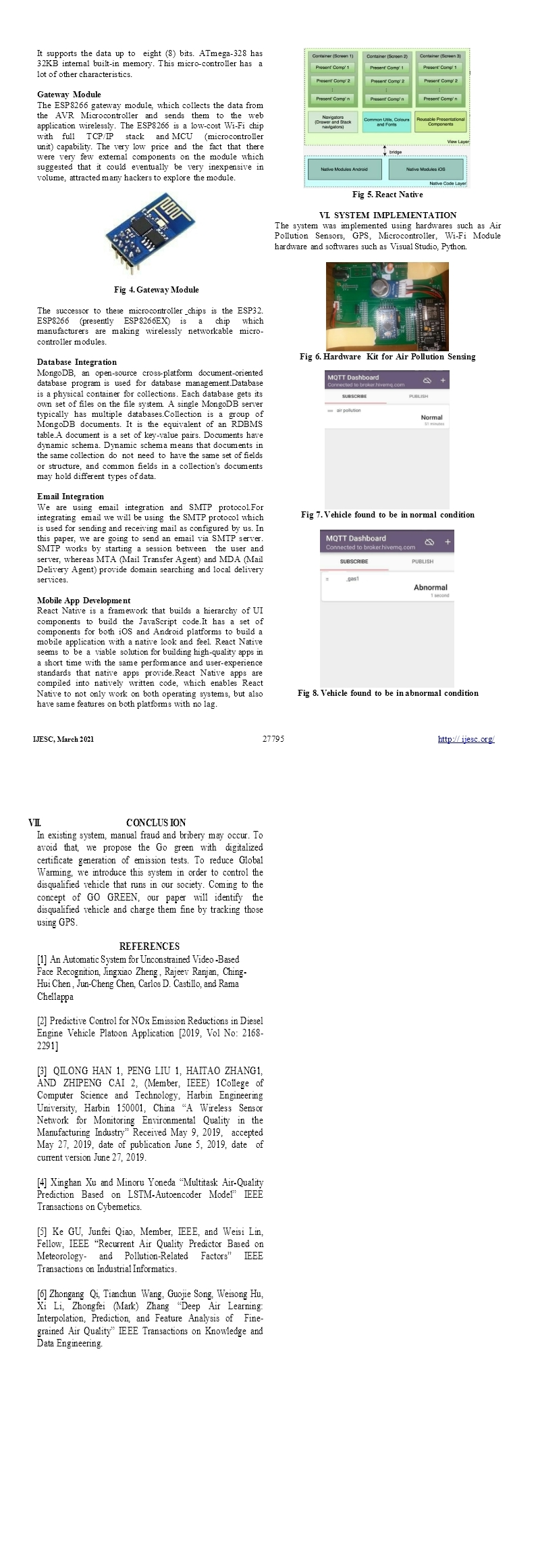
# PUBLICATIONS

**A.2 PUBLICATIONS**

# Fig A2.1 Image of Journal Paper Published

**Fig A2.1 Published Journal Paper**

**50**



# Fig A2.2 Published Certificates

# REFERENCES

1. An Automatic System for Unconstrained Video-Based Face Recognition, Jingxiao Zheng , Rajeev Ranjan, Ching-Hui Chen , Jun-Cheng Chen, Carlos D.

Castillo, and Rama Chellappa

1. Predictive Control for NOx Emission Reductions in Diesel Engine Vehicle Platoon Application[2019, Vol No: 2168-2291]
2. QILONG HAN 1, PENG LIU 1, HAITAO ZHANG1, AND ZHIPENG CAI 2, (Member, IEEE) 1College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China “A Wireless Sensor Network for Monitoring Environmental Quality in the Manufacturing Industry” Received May 9, 2019, accepted May 27, 2019, date of publication June 5, 2019, date of current version June 27, 2019.
3. Xinghan Xu and Minoru Yoneda “Multitask Air-Quality Prediction Based on LSTM-Autoencoder Model” IEEE Transactions On Cybernetics.
4. Ke Gu, Junfei Qiao, Member, IEEE, and Weisi Lin, Fellow, IEEE “Recurrent Air Quality Predictor Based on Meteorology- and Pollution-Related Factors” IEEE Transactions on Industrial Informatic.
5. Zhongang Qi, Tianchun Wang, Guojie Song, Weisong Hu, Xi Li∗, Zhongfei (Mark) Zhang “Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality” IEEE Transactions on Knowledge and Data Engineering.
6. Zhenyi Xu , Yu Kang , Senior Member, IEEE, Yang Cao , Member, IEEE, and Zhijun Li , Senior Member, IEEE,”Spatiotemporal Graph Convolution Multifusion Network for Urban Vehicle Emission Prediction”[2020]
7. Yao Ma and Junmin Wang, Senior Member, IEEE,”Predictive Control for NOx Emission Reductions in Diesel Engine Vehicle Platoon Application ”.
8. Maryam Ghodrati, Member, IEEE, Ali Farmani, Member, IEEE, and Ali Mir, Member, IEEE,Nanoscale Sensor-based Tunneling Carbon Nanotube Transistor for Toxic Gases Detection: A First-Principle Study[2019]
9. Hui Min Kim 1, Chang Geun Heo 1, Sung Ho Cho2, and Gwan Soo Park,”Determination Scheme for Accurate Defect Depth in Underground Pipeline Inspection by Using Magnetic Flux Leakage Sensors”
10. Ravi Kishore Kodali,Greeshma, R.N.V.,Kusuma Priya Nimmanapalli,IOT Based Industrial Plant Safety Gas Leakage Detection System
11. Sourabh Jamadagni, Priyanka Sankpal,Shwetali Patil,”Gas Leakage and Fire Detection using Raspberry Pi”[2019]