Integrated Digital Monitoring System For Efficient Analysis Of Industrial Processes

By

Ahmed Hassan

00000278122

Supervisor

Dr. Mian Ilyas Ahmad

School of Interdisciplinary Engineering and Sciences (SINES)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

July 2022

Integrated Digital Monitoring System For Efficient Analysis Of Industrial Processes

By

Ahmed Hassan

00000278122

Supervisor

Dr. Mian Ilyas Ahmad

A thesis submitted in conformity with the requirements for the degree of *Master of Science* in

Systems Engineering

School of Interdisciplinary Engineering and Sciences (SINES)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

July 2022

Declaration

I, *Ahmed Hassan* declare that this thesis titled "Integrated Digital Monitoring System For Efficient Analysis Of Industrial Processes" and the work presented in it are my own and has been generated by me as a result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated
3. Where I have consulted the published work of others, this is always clearly attributed
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
5. I have acknowledged all main sources of help
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself

Ahmed Hassan

00000278122

Copyright Notice

This thesis is dedicated to my beloved friends and family members.

Acknowledgments

Do they not see the birds above them with wings outspread and [sometimes] folded in? None holds them [aloft] except the Most Merciful (Allah). Indeed, He is the Al-Seer of everything" (Al-Qur'an: Surah Al-Mulk – Verse 19).

Claiming that the verse holds immense importance in my heart and has been one true source of motivation throughout this journey, would be an understatement. All praises to Allah the Almighty, who enlightened my mind and put me through numerous shades of experiences that instilled in me the strength and persistence I needed to successfully complete my research. In all honesty, I'd like to express the greatest of respects for Holy Prophet Muhammad (PBUH), the genuine beacon of knowledge.

I pay my earnest gratitude to my research supervisor Dr. Mian Ilyas Ahamd for his sincere guidance and encouragement. Apart from supervising, he inculcated in me the appetite for hard work and self-dependence. In addition, my GEC members: Dr. Shehzad Rasool, Dr. Absar ul Jabbar and Engr. Sikandar Hayat Mirza, must be acknowledged for their valuable advices and support throughout.

I also pay my heartiest gratitude to Mr. Abdullah for his guidance and motivation. Special thanks to my closest friends and SINES fellows, Mahboob ul Rasool, Umar Shahid for their support and help to pass every hurdle. I am really grateful to my lab fellows, Mahboob Ul Rasool, Umar Shahid, Faizan Bashir and Faiza Rao, for endless guidance and teachings, not only in the field of research but also for long lasting daily life.

Abstract

The problem of developing a digital monitoring system for efficient analysis of industrial processes has been considered. According to a recent study, Pakistan based industries have very high energy losses as compare to global energy losses. Such industry specific losses and other management related issues can be addressed with digital monitoring platforms. They have applications in agriculture, healthcare, aerospace, defense, and manufacturing. There are different types of digital monitoring systems and involve multiple software and hardware to achieve the required functionalities. In general, we can classify a digital monitoring system in three parts: front-end applications, back-end applications and processor/communication part. The front-end application include web and mobile-based applications that provide the required data in a user friendly format and allow human interaction with the digital platform. Back-end applications include sensors/models and actuators that are connected to the distributed components of industrial process that is being managed. Processors and Communication include the WiFi-based hardware for receiving, processing, and sending the remote data from the field instrumentation devices to database. The main objective of this work is to explore different software and hardware based components according to user requirements and suggest an integrated framework. We developed a digital monitoring system that integrate sensors and MATLAB based models to ESP32 processor and display the processed data on web and android based devices. The frontend applications have been designed to support real time sensor data and computational data along with storage of the data in a database. The complete monitoring system with the designed web and android applications is implemented on a case study of analyzing energy utilization of steel manufacturing process.

Contents

1 Introduction **1**

1.1 Background 1

1.1.1 Back-End Applications 2

1.1.2 Processors and Communication 3

1.1.3 Front-End Applications 3

1.2 Problem Statement 4

1.3 Motivation 4

1.4 Objectives 4

1.5 Thesis Organization 5

2 Literature Review **6**

2.1 Typical industrial monitoring systems 6

2.1.1 Sensing and Acquiring Data 6

2.1.2 Processors and Communication 9

2.1.3 Front-end Apps 12

2.2 Gap Analysis 14

3 Research Methodology **15**

3.1 Flow Chart 15

3.1.1 Development of digital framework 17
3.1.2 Hardware and software technologies 17
3.1.3 Back-end Technologies 18
3.1.4 Front-end Technologies 19
3.1.5 Detailed Block Diagram for Industrial Dashboard Mobile App . 19
3.1.6 Detailed Block Diagram for Industrial Dashboard Web App . . 20
3.1.7 ESP32 Development Board 24
3.2 Alternate Options 24
3.2.1 Back-end Options 25
3.2.2 Front-end Options 25
3.2.3 Processors and Communication 26
3.2.4 Mobile App Options 27
3.2.5 Database Options 27

4 Results and Discussion **29**
4.1 Validation experiment 29
4.1.1 Outcomes 29
4.2 Principal Experiment 30
4.2.1 Processor and communication 32
4.2.2 Back-end Module 32
4.2.3 Front-end Module 33
4.3 Systems Engineering Perspective 39

5 Conclusion **40**
5.1 Conclusions 40
5.2 Future directions 41

References **42**

List of Figures

1.1 Overview of data acquisition, storage, communication and visualization 2

2.1 Traffic Data monitoring illustration diagram used in [1] 7

2.2 PV Station Diagram used in [2] 8

2.3 PV Station Diagram using ESP32 in [2] 11

2.4 Data visualization diagram for SCADA System [3] 12

2.5 Smart farm monitoring system diagram [4] 13

3.1 Flowchart of proposed methodology 16

3.2 Hardware components integration 18

3.3 Data storage at server side 19

3.4 Overview of web application GUI building tools used in proposed methodology 20

3.5 Overview of mobile application GUI building tools used in proposed methodology 20

3.6 Overview of mobile application GUI building tools used in proposed methodology 21

3.7 Overview of mobile application GUI building tools used in proposed methodology 22

4.1 Minimal prototype circuit with Arduino and breadboard 30

4.2 MATLAB based computational graphs on Thingspeak 31
4.3 Sensors data in the forms of numbers visualized on initial web app . . . 31
4.4 Web based plant overview of whole industrial monitoring process . . . 34
4.5 Plant view that displays whole process information 35
4.6 Composition view of web application 36
4.7 Furnace view of web application 36
4.8 Furnace view of mobile application 37
4.9 Ladle view of web application 37
4.10 Ladle view of mobile application 38
4.11 Tundish view of web application 38
4.12 Tundish view of mobile application 39

Chapter 1

Introduction

This chapter is about background of this research, motivation of this research, objective of this research and organization of thesis.

1.1 Background

The use of technology in industries such as defense, manufacturing, aerospace, healthcare and agriculture has significantly increased the problem of monitoring the industrial process to optimize its operation. The monitoring/analysis of single or multiple components of industrial process during production or idle phase not only optimize its operation and cost but also helps in ensuring safety. For this purpose, different hardware and software technologies can be utilized, that may include MATLAB/Python based models, standalone applications, real-time sensors, microprocessors, storage devices and front-end display applications. Manufacturers are integrating new technologies and concepts such as Internet of Things (IoT), cloud computing and analytics, and artificial intelligence/machine learning into their production framework and throughout their operations. This is what we call the fourth industrial revolution (industry 4.0) starting from steam based power in the first revolution to electric power in the second and computer based automation and control in the third industry revolution. The development of a digital monitoring system for smooth and efficient operation of industrial processes itself involves large number of integrated hardware and software tools that make their utilization a challenge. The digital monitoring system typically include back-end applications and hardware; front end display applications; and communication platforms. Figure 1.1 shows a general overview of data acquisition, storage and visualizing tools.

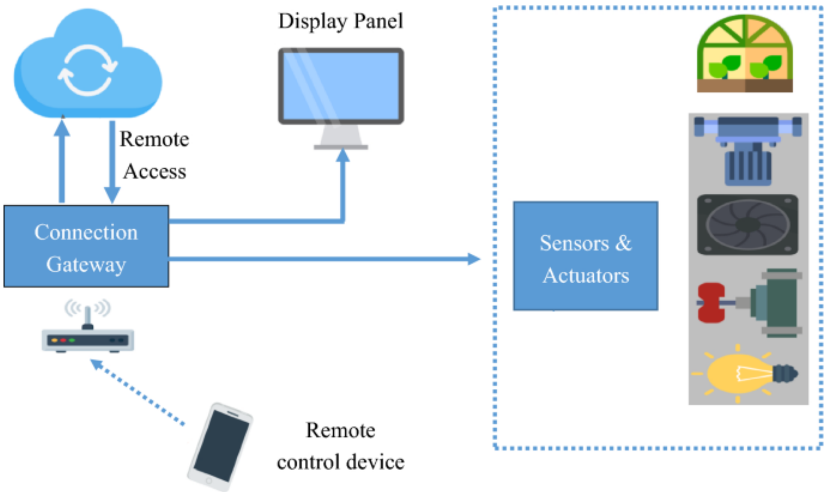

Figure 1.1: Overview of data acquisition, storage, communication and visualization

1.1.1 Back-End Applications

Back-end applications include all the industrial process specific tools and hardware that help in observing the behavior of the industry. For example, the tools/hardware used for data acquisition are back-end applications. Data acquisition is the process of gathering measurements from different sensory devices in an industrial unit, such as thermometers, pressure gauges, and pyrometers. Based on the data collected, operators and decision-makers can also take the necessary controlling measures. The sensors and actuators that make up the system's back end are shown in the right block of Figure 1.1. In addition, there might be model-based analytics that are again part of back-end applications.

1.1.2 Processors and Communication

Processors and communication refer to the hardware and software required for integrating the front-end and back-end applications. Processors play a significant part in industrial monitoring systems; they gather information from back-end technologies, process it, and integrate with hardware tools to transfer the data to front-end applications. In order to establish backup/storage of performance-related parameters for industrial components, the processed data is also transferred to a central database using a variety of physical or virtual communication mediums, including WiFi, Bluetooth, and hardcore wires or cables. This database can then be accessed at any time for additional processing. In the top left corner of Figure 1.1, the connection gateway that makes up the system's communication channel is depicted.

1.1.3 Front-End Applications

The front-end applications refer to the tools where the user engage with the digital monitoring system to observe the industrial process. This is also called client side of an application. Front-end apps are used in industrial monitoring process systems to visualize the entire process in one perspective. Using tools and software, for instance, to display the whole industrial process data and communicate with humans. For a rapid understanding of the system/process performance, different visualization techniques are utilized to show data in an organized, portable, and simplified format. Graphs, charts, and simulations are a few prominent visualization formats. To obtain a more comprehensive understanding of the industrial process, the data kept in the storage systems is retrieved for manipulation and display. The information on display serves as a resource for adopting encouraging actions that will, ideally, have an impact on the industrial process systems. The display panel and remote control that depict the applications front-end are shown in the center of Figure 1.1.

1.2 Problem Statement

The problem is to design and develop a digital monitoring system that integrates different hardware and software technologies to visually track, analyze and display data of industrial process system using web and android-based applications. There are different options that can be adopted at the front-end application level, back-end application and processor/communication level. Some of these tools are compatible with each other but require additional software for integration. A detailed overview of such digital monitoring systems for industrial process and their associated software and hardware tools from system engineering point of view is the focus of this thesis.

1.3 Motivation

The motivation behind this thesis is to get detailed information about different digital frameworks that provide instant access to all the key components of an Industrial unit. Such monitoring systems are continuously growing with various user-friendly graphical user interfaces (GUIs). The GUIs are typically designed in web-based software for computer applications and android-based software for mobile applications. These Web and mobile-based applications can be used in a wide variety of application areas. Some of these applications include real-time energy monitoring systems, smart home systems, weather prediction systems, and health care system and defense applications.

1.4 Objectives

The complete thesis work has been classified into the following four objectives.

- Identify and explore different hardware and software technologies that can be integrated with each other in order to analyze an industrial process.
- Develop a digital framework with different hardware and software technologies and visualize the associated data for monitoring of an industrial process.

- Design web and mobile-based applications that can support real time as well as computational analysis of industrial process along with the storage of sensor/model based data in the database.
- Implementation of the applications and the digital framework on an actual industrial process as a case study.

1.5 Thesis Organization

The remaining portion of the thesis is structured as discussed. In chapter 2, detailed literature review has been presented on the problem statement discussed in Section 1.2. The methodology used for development of a digital framework on monitoring of industrial process has been discussed in Chapter 3. In Chapter 4, the implementation of a digital platform for a specific case study has been carried out. Finally, in Chapter 5, the conclusions are made from the research work and some future directions are highlighted.

Chapter 2

Literature Review

This chapter gives background information on several digital frameworks that have been utilized to monitor industrial process systems.

2.1 Typical industrial monitoring systems

The digital monitoring system generally consists of communication platforms, front-end display applications, and back-end hardware and software. It takes a lot of integrated hardware and software tools to construct a digital monitoring system for smooth and efficient operation of industrial processes, which makes it difficult to use.

SCADA is an abbreviation for supervisory control and data acquisition. A SCADA system is a group of both software- and hardware-based elements that enables both local and remote monitoring and management of industrial facilities and processes [3].

2.1.1 Sensing and Acquiring Data

The process of acquiring statistics and information related to industrial processes from a variety of sensory devices utilized for a specific process in an industrial unit is known as data acquisition. Operators and decision-makers may also take the appropriate supplemental steps based on the data gathered. Various types of data collecting methods are used in many investigations. Such as, SCADA is an industrial monitoring system for supervisory control and data collection that uses field instrumentation to gather data, the system includes data recording for real-time data collecting, and processing. Typically, Programmable Logic Controllers (PLCs) or Remote Terminal Units (RTUs) serve as the foundation of the architectural architecture of a SCADA system (PLCs). These RTUs and PLCs are microcontrollers or microprocessors that interact and communicate with Field Instrumentation Devices (FIDs), which include sensors, actuators, valves, pumps, and transmitters for acquiring data [3, 5]. In another study a programmable sensor module is used to measures and gathers data from temperature, humidity, soil moisture, and a water level sensor of solar irrigation pumping systems [6]. In another research a smart home monitoring system is implemented where collection of data for managing and monitoring home systems, a variety of sensors are utilized, including those for measuring the room's temperature, gas leaks, water levels, and person detection [7]. A lot of research is conducted on city traffic and public transport monitoring systems. In an investigation for transparent and responsible decision-making in smart cities, two distinct datasets Waze and Rio are employed in data-driven dashboards for traffic monitoring and two other datasets moovit and Rio are used in public transport monitoring [1]. Figure 2.1 represents the model used for traffic monitoring system.

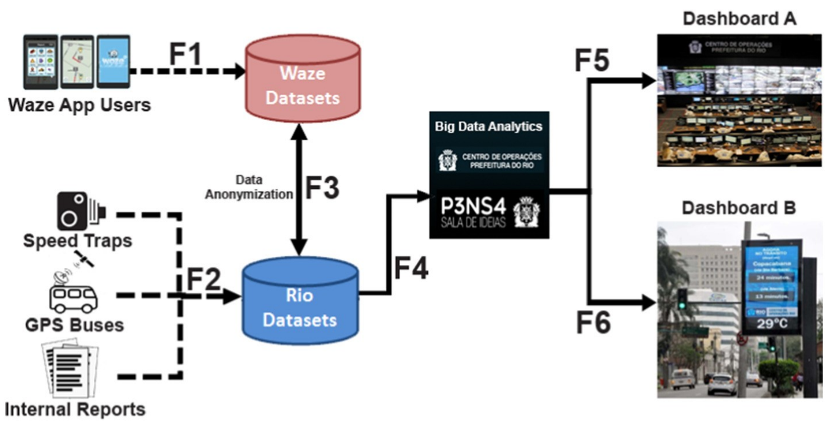

Figure 2.1: Traffic Data monitoring illustration diagram used in [1]

Numerous motors of various sizes and powers are available in the plastic recycling company. In order to forecast motor failure, information on power consumption, temperatures, vibrations, and other pertinent factors is being gathered. A low-power

System-on-a-Chip (SoC) micro-controller that gathers data is used in this study [8].

For energy monitoring, solar installations use a set of equipment. Real time data are collected by using ESP32 micro controller and employed for further processing from a variety of sensors, such as DHT11, INA219, and BH1750. Figure 2.2 represents the PV station model used in [2].

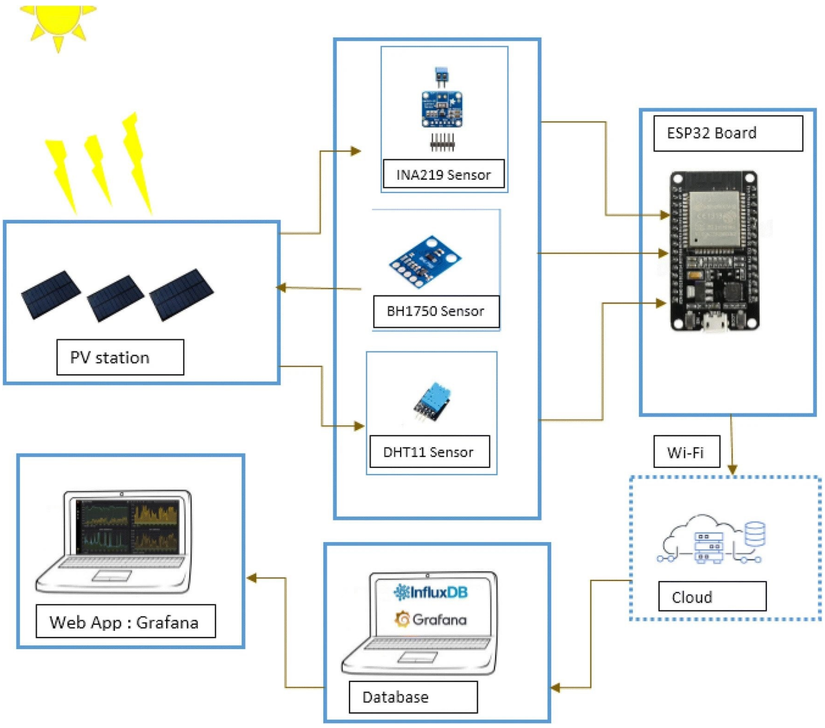

Figure 2.2: PV Station Diagram used in [2]

In a scenario involving the monitoring of workplace safety is utilizing the installed wireless sensor network where data collecting is carried out with a focus on five key parameters: temperature, fire, smoke, gas leakage, and radiation [9, 10]. In another interactive system built on the IoT (Internet of Things) combines non-intrusive sensors and data gathering equipment, reliable communication networks, cloud-based databases, and web servers to accomplish real-time monitoring of energy use in industries [11].

Sensor nodes are designed to be flexible solutions that do not need extensive wiring whereas establishing an energy-efficient operating mode is also crucial. In [12] a technique is used which switches nodes on demand when needed. A lot of research is conducted on environment variable monitoring systems in which statistical dashboards and micro controller chips are used to accomplish the goal of environment monitoring systems, for example, a node based sensor model is utilized for data collection of different environment variables such as humidity, temperature, leaf temperature and rain, from a greenhouse model [13]. Similarly, the environment's weather parameters, including humidity, temperature, and climatic quality monitoring system is investigated in [4]. Temperature and humidity are also utilized in agricultural environment factors monitoring [14].

Sensor models are not only used for industrial or environmental monitoring systems but also used in medical equipment for patient's health monitoring such as blood pressure monitoring systems [15], ECG [16], heart rate monitoring systems [17], EEG [18] and many others. In a research the usage of battery-powered chest belt of the wearable ECG sensor is reported for the measurement and streaming transmission of electrocardiogram signals. The circuit used in it, isolates, purifies, amplifies, and digitizes the ECG signals and visualize statistics on dashboard [19].

The sensor units are embedded technologies that continuously track the physical states like, energy, current and power or chemical make-up of the fuel or mobile oil of motor bikes motor cars or whatever the object being sensed. The ICMSMET's (Intelligent Control and Management System for Motorcycle Endurance Test) perception layer is used to achieve real-time signal collecting from test processes, including motorbike acceleration, torque, engine temperature, and ambient temperature and humidity [20]. In another system consists of energy monitoring nodes that employ a non-invasive CT (current transformer) sensor, a low-cost energy meter, the SD3004 energy measurement chip, and a microprocessor to detect the voltage, current, active power, and cumulative power consumption [2, 7, 8, 10, 21].

2.1.2 Processors and Communication

Processors are important components of industrial monitoring systems, data acquired from the field devices is fed to it which is then processed, and integrated with hardware tools to transfer it to front-end applications or to create backup storage of performance-related parameters for equipment. The processed data is then transferred to a central database. Various types of processors and communication channels are utilized in various industrial monitoring systems and other research studies. In a research study single board computers such as Remote Terminal Units (RTUs) and Master Terminal Units (MTUs) are used for data collection and data processing respectively. RTUs collect data from field devices such sensors and actuators where MTUs are used to process data gathered by RTUs. Another single board computer ESP32 works as a processing and communication channel between RTUs and MTUs [3]. ESP32 is commonly used microcontroller for data processing purpose [2, 3, 7, 8]. Figure 2.3 illustrates the PV station's workflow.

In [2], ESP32 DEVKIT V1 is used as a processor and a communication route between the visualization tool and the data that was gathered. In this study transmission of data takes place in two stages: The first phase includes the exchange of information between the sensors, and the controller through inter-integrated circuit protocol (I2C), the controller then communicates with the cloud service application through the WiFi protocol in the second phase of the process. On the other hand, analytical models are also employed for industrial data processing and data analysis to accomplish this goal big data analytics P3NS4 simulator is used in [1]. In another simulation experiment, a deterministic small network architecture with 5 nodes is used that acquire data from servers and after processing send it to host computer through a communication route named RF Zigbee [9, 10]. People have also developed Apache web server to provide an interactive dashboards that display statistics as well as publish data to data centers using the MQTT protocol, which is sent via wireless communication network [11]. In another study MQTT protocol is used in which measured data of several sensors subsequently be sent in a JSON (JavaScript Object Notation) format to the server through MQTT. To function as a local server, the Raspberry Pi 3 model B was used [21]. Microprocessors and embedded computers are the key components of sensor networks. These are tiny, high speed as well as low-powered for example, nRF2401 chip is microprocessor utilized for wireless communication and MSP430F149 is a microcontroller used for processing [14]. Raspberry Pi and Arduino Uno are also

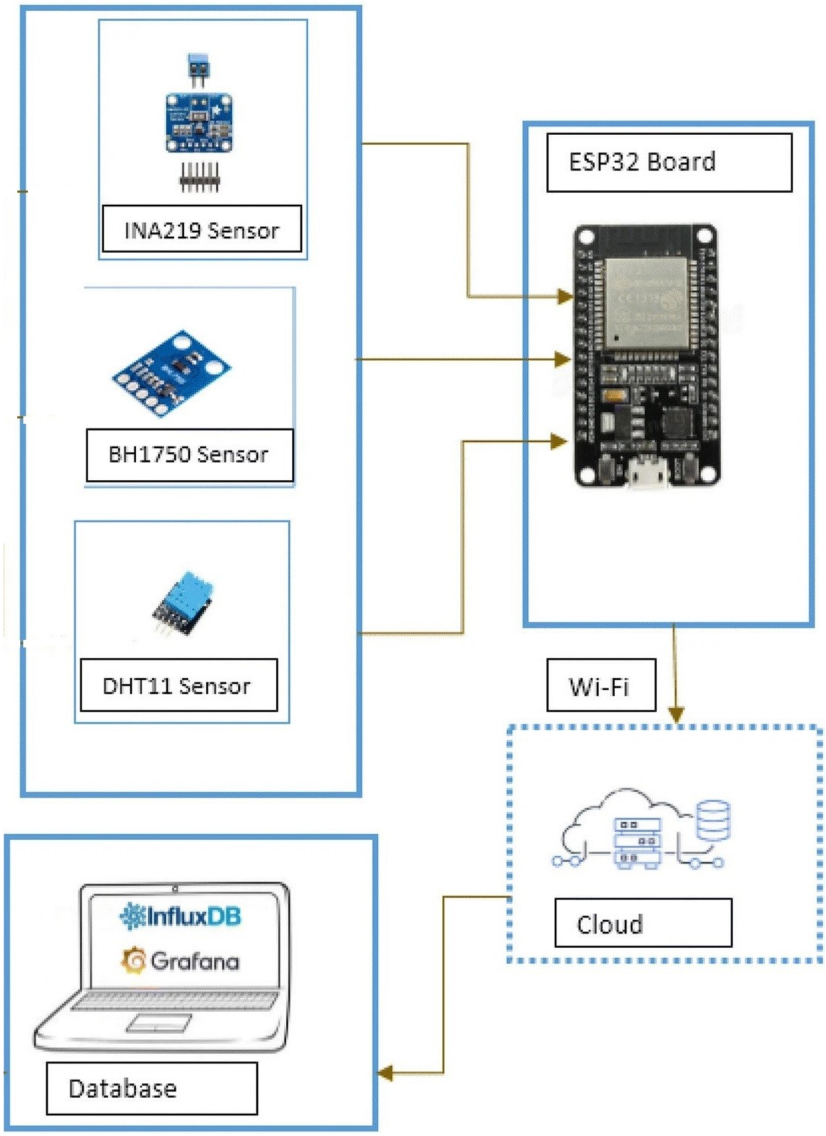

Figure 2.3: PV Station Diagram using ESP32 in [2]

the intelligent embedded computers that are utilized for different kinds of monitoring. The poultry farms monitoring systems are using Arduino and Raspberry Pi to address chicken farm related problems where both the Raspberry Pi and the Arduino Uno serve as processors and communication devices [4]. In [13] A microprocessor (MSP430 MCU) is also employed in a study that receives data from multiple sensors processes it to get useful information to monitor and manage system.

2.1.3 Front-end Apps

Front-end apps are the primary visualization tools that display statistical information regarding industrial data coming from field instrumental devices. These apps can be some web based data visualization dashboards, some mobile or desktop applications. In every monitoring setup different kind of front-end applications are demanded for statistical visualization. For example, in a city road traffic monitoring system a web based dashboard is used to visualize road traffic statistic [1]. In another study Grafana is used as front-end application. Grafana is a free and open-source platform for data monitoring and visualization. It comes with a web server, which enables users to connect to the platform from any location [2, 21]. In SCADA monitoring system a local server named thinger.io is used as a primary visualization tool [3], see Figure 2.4.

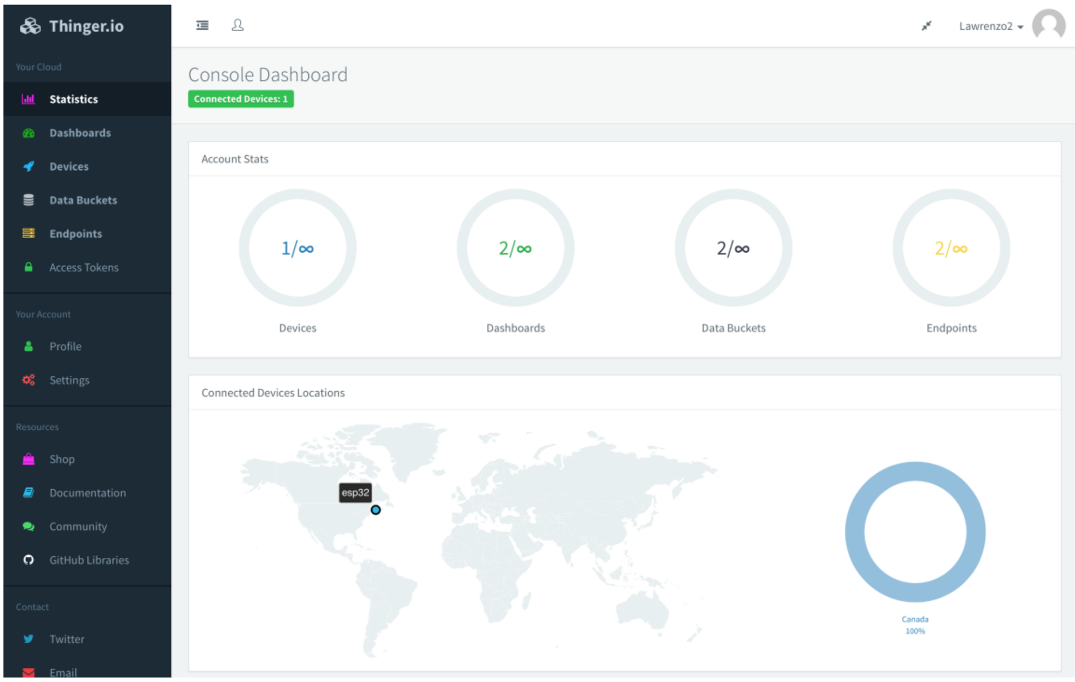

Figure 2.4: Data visualization diagram for SCADA System [3]

In another study a web based application integrated with data visualization dashboard is used which can interact with cloud software to upload sensor data, allowing users to examine the state of the field using any internet browser. It may also be linked to the web server, which is in charge of sending the client a web page when the client connects to it via the HTTP (Hypertext Transfer Protocol) protocol [6]. On the other hand some desktop based applications are also used for data visualization. For example a MATLAB based open source data cloud storage and visualization tool named Thinkspeak is used to show sensors status [7]. Lab View is also a desktop based application and a virtual instrumentation software that is used to perform the data logging, monitoring, and control operations additionally, this makes it possible for a user interface that is simple to use and for data to be accessible [9, 10]. Some other mobile based applications are also reported for monitoring home as well as industrial automation systems to control over home devices and industrial devices respectively [22]. In a research that looked at the relationship between Raspberry Pi and smartphones in which Raspberry Pi is used to analyze data and then send it to smartphones application for visualization [4], see Figure 2.5.

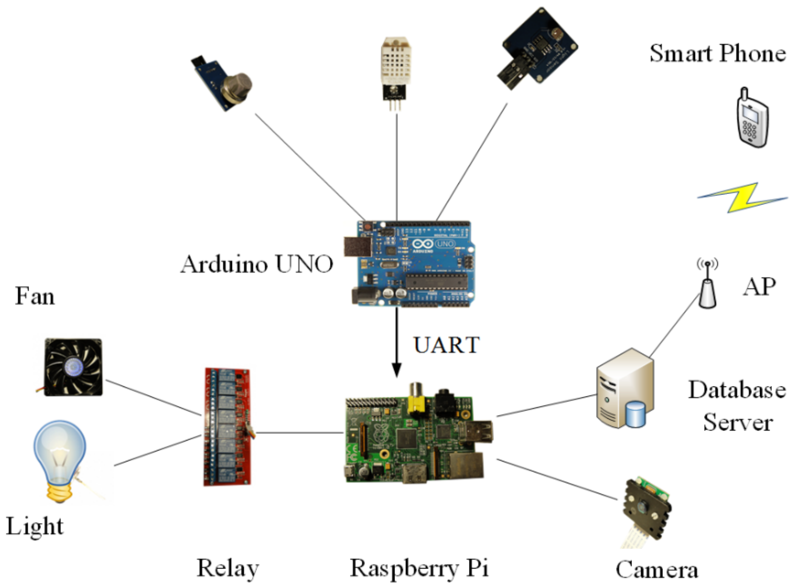

Figure 2.5: Smart farm monitoring system diagram [4]

Another web and mobile application based system is developed for iOS and Android which includes the information management system to enable for the control of all system settings as well as the monitoring of alerts and occurrences. The web application enables authorized users with access to the outside communication system to configure the system from a distant computer via the Internet. After data is visualized on Mobile and web based application for further monitoring [19] the smartphone application also offer notifications for further actions [12]. The agricultural and greenhouse setups discussed above are made efficient by using smart monitoring system. The data gathered by an environmental sensor installed in greenhouse models are recorded in the database, and the GUI (Graphical User Interface) are also created to visualize management specifics about the status of the greenhouse [13]. In another research, C programming language used for software design where Lab View is in charge of handling the collection of data presentation, analysis, and archiving for agricultural environment monitoring [14].

2.2 Gap Analysis

All the studies discussed above are limited to some extent. People researched a lot on embedded systems and made real-time working industrial monitoring systems but most of the systems are very complex in terms of hardware involvement for example a heavy amount of wiring that's an important safety concern. The studies like SCADA systems, PLC systems, and IOT systems are very efficient but in some cases may not be cost effective. We have proposed an efficient and effective Industrial Monitoring System in terms of cost, hardware involvement, user friendly front-end applications that can be operated anywhere, either from desktop station at office or from personal mobile phone. Most importantly the use of complex wires is reduced in order to minimize the safety risks.

Chapter 3

Research Methodology

This chapter explains how we achieved our four objectives. The chapter is structured in the following way: We begin with the flowchart of the complete end-to-end process, in which we identified and investigated various hardware and software technologies utilized in our methodology. Following that, we gave a quick explanation of the digital foundation for industrial process systems. Then, we presented the design of web- and mobile-based applications for visualization of the industrial monitoring system.

3.1 Flow Chart

Our flowchart shows how industrial monitoring system/process works. Across many disciplines, flowcharts are regularly used to analyze, organize, improve, and communicate frequently complicated processes in a straightforward, accessible manner. For understating more efficiently, the flowchart of the whole process is displayed in Figure 3.1.

Explanation of the flow chart

- ESP32 development boards have been integrated with all industrial sensors. The ESP32 collects the digital sensor readings, connects to a WiFi router (the SSID and password of the access point are specified in the Arduino code), and then

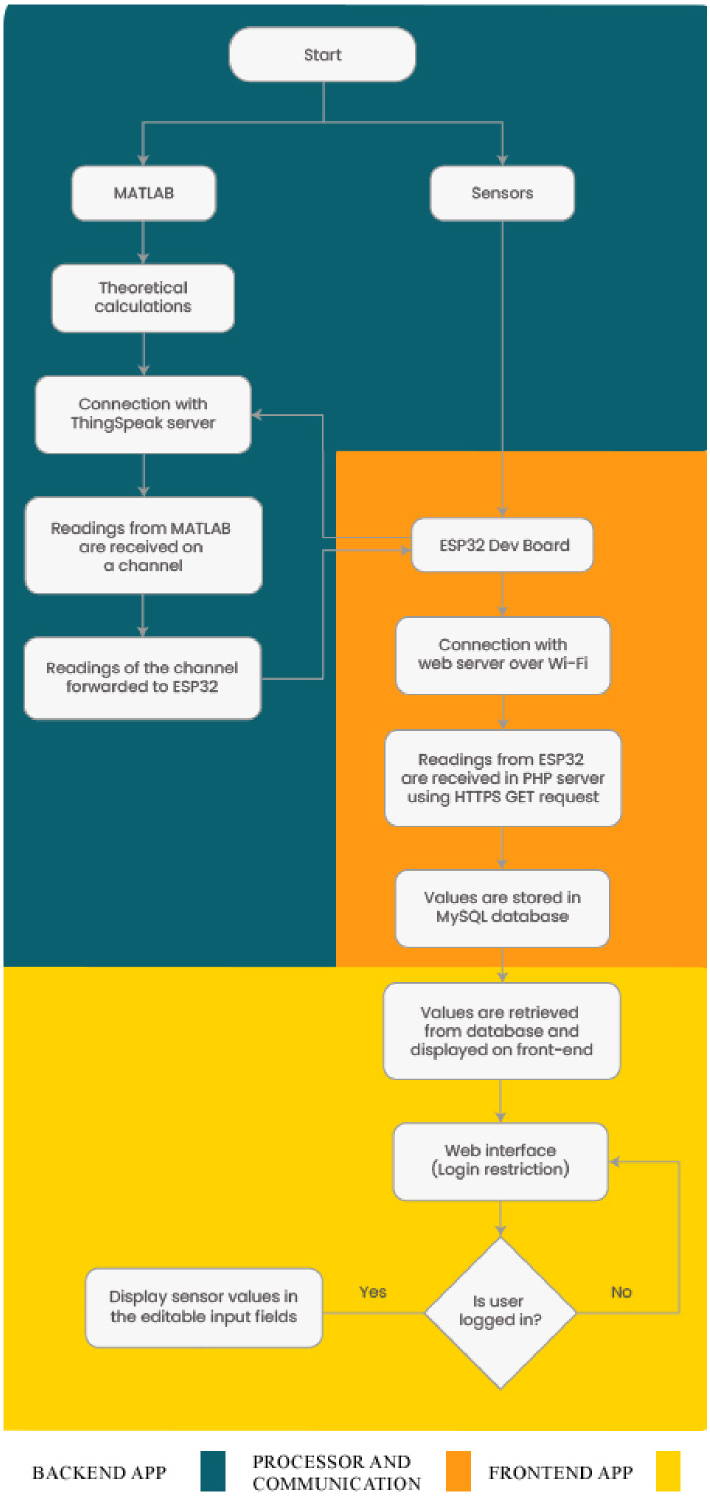

Figure 3.1: Flowchart of proposed methodology

makes a secure HTTPS connection to a web server.

- As soon as the ESP32 connects to the webserver, the readings from the sensors and the server are inserted in the webserver's URL and delivered. The HTTPS GET request protocol is used to handle the web server request.

- After receiving the HTTPS GET request, the PHP server filters the values from the URL using the Real Escape String functionality to stop malicious queries or fake values from being executed. The appropriate database tables are then populated with values.
- A login form is used to restrict access to the website's front end. No sign-up or registration form is available for adding a new user. The MySQL database table must be manually established for each new user. Users' data, including their username and password, is encrypted using MD5 and stored in a database table.
- The user sees all the readings in the pertinent blocks after logging in. Real-time database retrieval is used to get all readings. Any variation in the readings is abruptly displayed on the front end. AJAX, a web technology, was used for this.

3.1.1 Development of digital framework

Utilizing a variety of hardware and software technologies, we have developed an industrial process system as well as web and mobile apps. We have categorized and discussed the technologies below that we are using to create the required industrial monitoring system.

3.1.2 Hardware and software technologies

To display the readings, we combine a variety of industrial sensors (including the PT100, Pyrometer, Spark analyzer) with crucial information gathered from furnaces, ladles, tundishes, and element composition. We have used certain modules (MAX31865, RS485, and RS232) between the sensors and esp32 board separately from connecting them with each other so that we can easily obtain data because the aforementioned sensors aren't providing readings directly to the esp32 board. Because ESP32 contains built-in Wi-Fi and Bluetooth modules and is simple to program in Arduino software, it was chosen for the project. The sensors can be linked to its input and output ports, and the information obtained from the sensors can then be transmitted to any device that supports Wi-Fi over the internet. The figure demonstrates how various hardware parts are connected to one another in Figure 3.2.

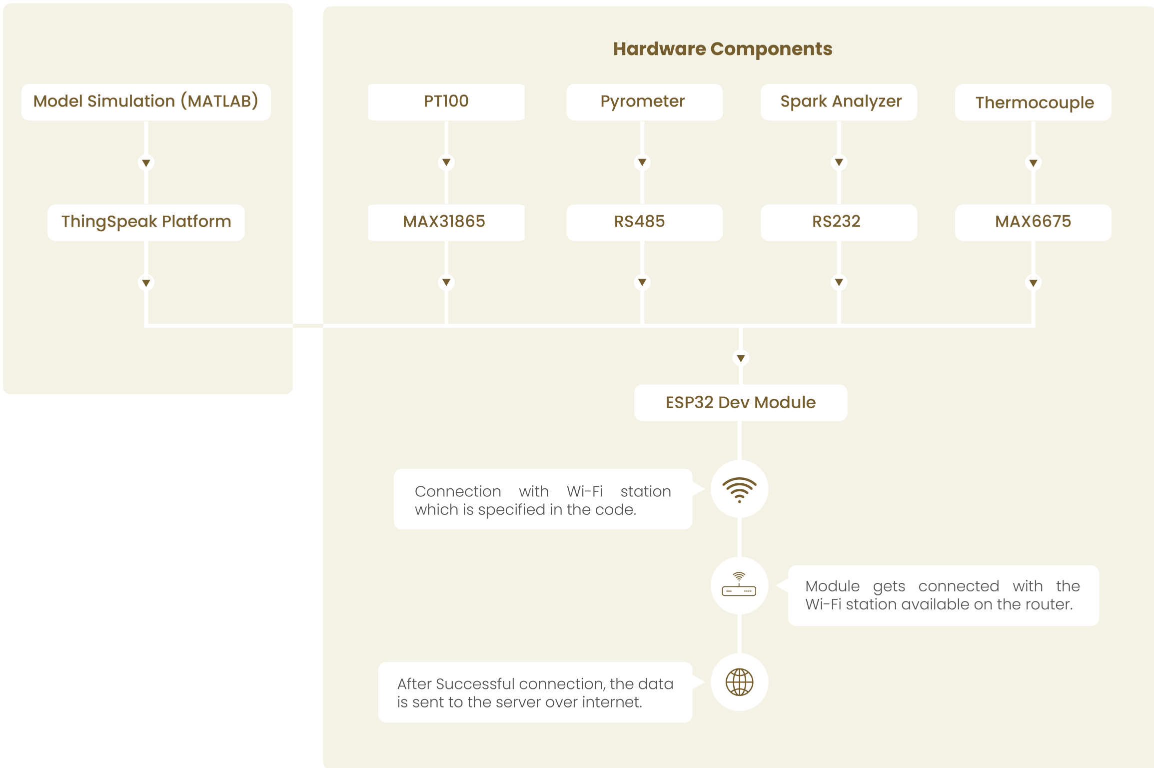

Figure 3.2: Hardware components integration

3.1.3 Back-end Technologies

Data gathered at the ESP32 board is currently forwarded to the server. As a back-end language, PHP is used. An adapted server-side programming language called PHP is quick, adaptable, and practical. PHP APIs at the server use the get request mechanism to receive data from the ESP32. Data is directed toward the database after filtering and validation. Our database is powered by MySQL. By using a get request from the client, the same APIs may be used to retrieve database data. This entire process was described in Figure 3.3.

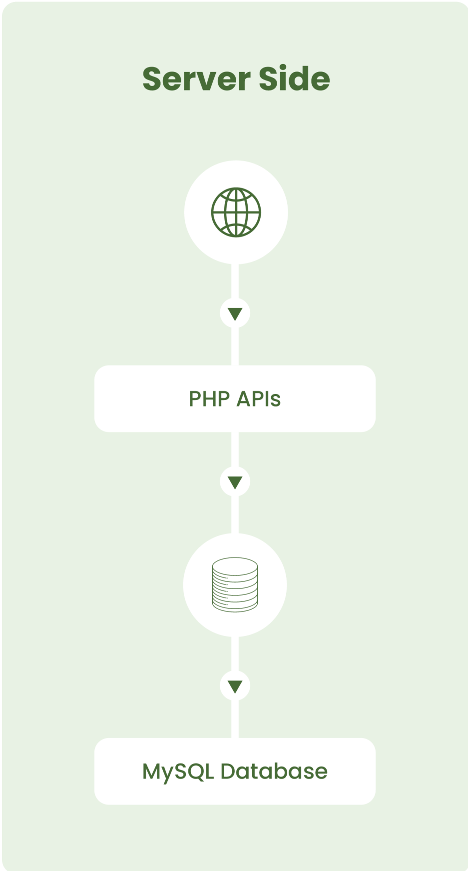

Figure 3.3: Data storage at server side

3.1.4 Front-end Technologies

There are a number of built-in APIs in the PHP language that fetch data from each table once it has been stored in the database and provide it to the client's computer. APIs are now sending the data to web and mobile applications. Figure 3.4 and figure 3.5 explains the web and mobile application as well as the programming languages we are employing.

3.1.5 Detailed Block Diagram for Industrial Dashboard Mobile App

A detailed block diagram that explains the three phases of the thesis more efficiently is shown below in figure 3.6.

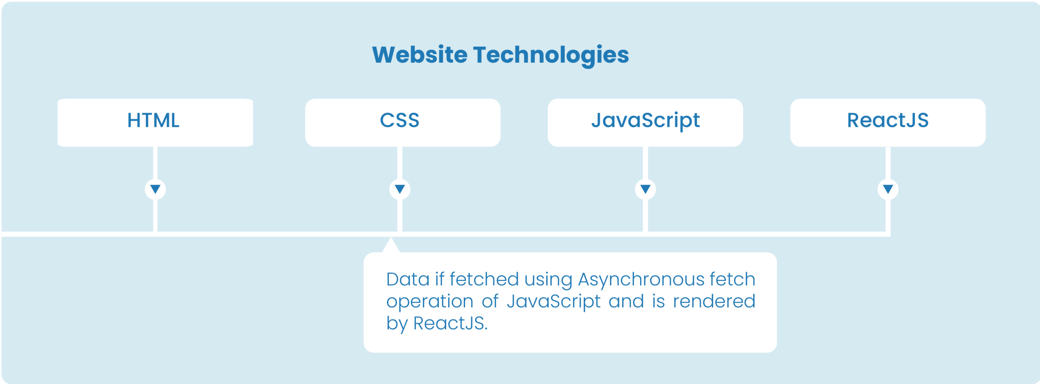

Figure 3.4: Overview of web application GUI building tools used in proposed methodology

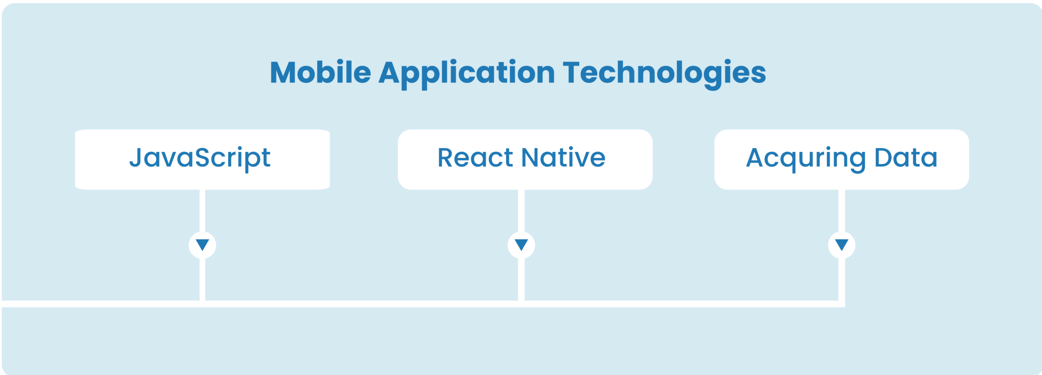

Figure 3.5: Overview of mobile application GUI building tools used in proposed methodology

3.1.6 Detailed Block Diagram for Industrial Dashboard Web App

The Web App that we have developed displays readings from several industrial sensors (such as PT100, Pyrometer, and others) as well as other critical data received from furnaces, ladles, and tundishes, such as element composition to show the status to the 'user' monitoring it. By using ESP32, data that is received from different sensors are sent to a server where all the data is protected and saved in a database and displayed on the apps for real-time monitoring. If any sensor does not respond, the user can identify it by analyzing the data and take essential measures. Figure 3.7 shows the workflow of web application development.

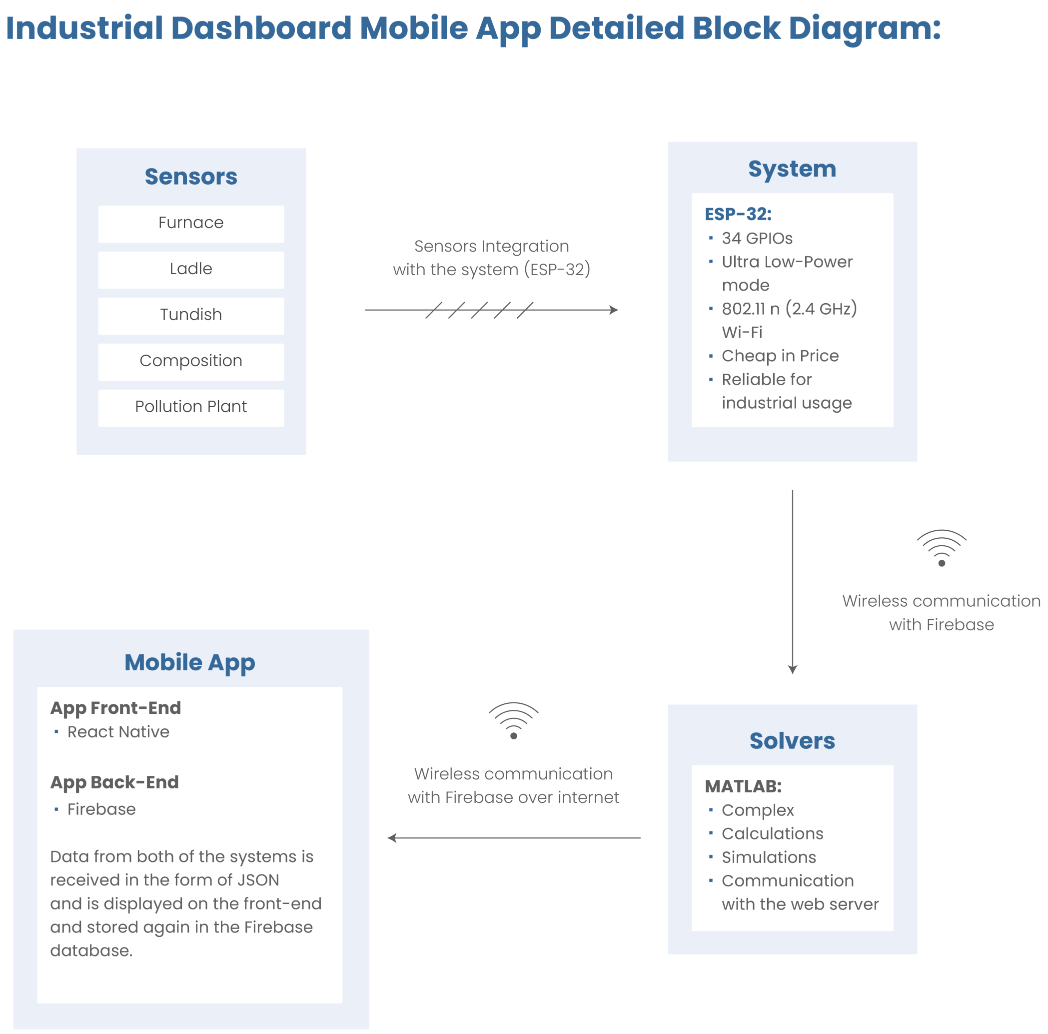

Figure 3.6: Overview of mobile application GUI building tools used in proposed methodology

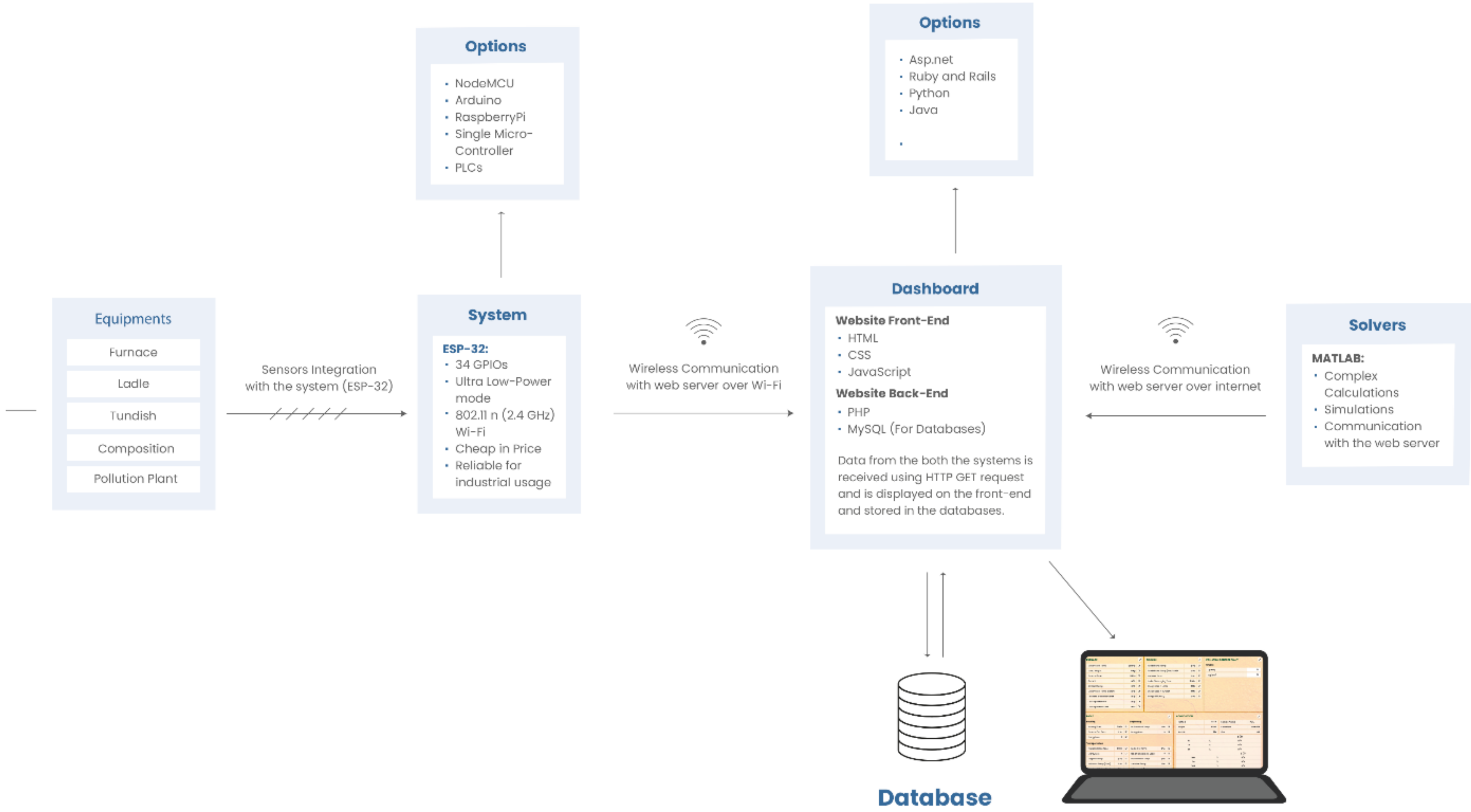

Figure 3.7: Overview of mobile application GUI building tools used in proposed methodology

- **HTML5 (Front-end)** An industry-standard markup language called HTML (Hyper Text Markup Language) is used to organize and display content on the pages of a website or a dashboard. The front-end language is HTML5. The most recent and improved version of HTML is possibly HTML5. Technically speaking, HTML is a markup language that explains the purpose, meaning, and organization of content in a document rather than a programming language.
- **CSS3** "Cascading Style Sheet" version 3 is known as CSS3. CSS, or Cascading Style Sheets, is a language that specifies the layout and formatting of web pages. It is utilized to adorn or stylize HTML documents. It determines how HTML elements will look. Additionally, CSS3 is an improved version of CSS. To help with the development of online display, CSS3 includes a number of new design features and advancements.
- **JS ES6** JavaScript, also referred to as JS, is an ECMA Script-compliant programming language. High-level programming language JavaScript has a number of paradigms and is frequently real-time compiled. This language has first-class functions, prototype-based object orientation, dynamic or effective typing, and curly-bracket syntax. ECMA Script 2015, often known as ES6, is the sixth edition of the scripting language that was published in 2015.
- **PHP (Backend)** A well-known general-purpose programming language called PHP is an effective tool for creating dynamic and interactive website and is perfect for developing online applications. PHP is a flexible, opportunistic, and rapid programming language. It serves as a backend language for building databases in the method we recommend. A PHP interpreter is often used on a web server to parse PHP code. The output of the interpreted and executed PHP code, which could be any sort of data, such as binary image data or the generated HTML, makes up all or part of an HTTP response on a web server.
- **MySQL** MySQL is a free and open-source database management system for relational databases. MySQL is a piece of software used for database management systems; it is not a programming language. The words "My" and "SQL," which

stand for Structured Query Language, are taken from the name of co-founder Michael Widenius's daughter. There are many uses for MySQL, and we utilize it for database management.

3.1.7 ESP32 Development Board

The fully integrated ESP32 MCU is powered by a dual-core Tensilica Xtensa LX6 MCU and has built-in Wi-Fi and Bluetooth connectivity for use in a range of applications. ESP32 is directly programmed using the Arduino IDE (Integrated Development Environment). The ESP32 can operate reliably in industrial applications with an operational temperature. Thanks to sophisticated calibration circuitry, ESP32 can dynamically correct outside circuit issues and react to changes in the environment. To lessen the complexity of connectivity stacking on the main application CPU, the ESP32 can operate as a slave device to a host Microcontroller or as a standalone system. The ESP32 may communicate with another system to provide Wi-Fi and Bluetooth functionality via its SDIO/SPI or UART/I2C interfaces. The ESP32 includes RF baluns, filters, power amplifiers, low-noise receive amplifiers, power management modules, and antenna switches. The ESP32 provides your projects with invaluable capability and adaptability with minimal printed circuit board (PCB) needs. The ESP32 development board is what we're utilizing to connect all of the sensors to. Digital readings from the sensor are gathered by EP32. The ESP32 then establishes a connection with a Wi-Fi router (the access point's password and SSID are set in the Arduino code) and establishes a secure HTTPS connection with the web server. ESP32 and industrial sensors together (including PT100, Pyrometer).

3.2 Alternate Options

The technologies of various hardware and software have been examined here. These technologies can be connected with one another in order to examine a system for industrial monitoring, but their restricted characteristics prevent us from using them.

3.2.1 Back-end Options

One of the key components of every processing system is its data acquisition infrastructure. The backend may be developed using a variety of different technologies. Below is an explanation of some of the alternative technologies.

- **NodeJS** When it comes to computation of demanding and sophisticated tasks, NodeJS degrades performance. When an upgrade is introduced, there must be numerous code modifications.
- **ASP.NET** IIS web servers, which are needed for ASP.NET, are more expensive than typical shared hosting plans. Additionally, it uses a lot more server resources than PHP does.
- **Python** When dealing with real-time data, for instance, Python becomes slower during runtime. It also uses a lot of memory (RAM).
- **Ruby on Rails** A server side web application framework is Ruby on Rails. Ruby on Rails runs at a fairly sluggish pace. Deploying it is more difficult since you need to have a full stack.
- **MySQL** MySQL is a free and open-source database management system for relational databases. MySQL is a piece of software used for database management systems; it is not a programming language. The words "My" and "SQL," which stand for Structured Query Language, are taken from the name of co-founder Michael Widenius's daughter. There are many uses for MySQL, and we utilize it for database management.

3.2.2 Front-end Options

In visualization, front-end choices are crucial. The solutions that are presented below are some ways that we can construct GUI, however they need significant work and front-end language expertise.

- **Custom HTML, CSS, JS Coding** It is also possible to construct the front-end using simply HTML, CSS, and JS, but this can lead to code repetition and necessitate the use of complicated techniques in order to perform advanced dashboard functionalities.
- **Angular Framework** An alternative to ReactJS is the Google-developed Angular Framework. But building a dashboard with real-time data display also becomes challenging.
- **VueJS Framework** Another JS framework is VueJS, which is less well-known than ReactJS.

3.2.3 Processors and Communication

In industrial monitoring systems, processors are important because they collect data from back-end technologies, analyze it, and interact with hardware tools to deliver the information to front-end applications. For this, we also have a variety of other alternatives, but we're not utilizing them all. The technologies that are now available are described in short below.

- **Arduino** Due to the lack of a built-in Wi-Fi/Bluetooth module, which is necessary for this project, the Arduino development board is not practical in this situation.
- **Generic Micro-controller** A generic microcontroller is similar to Arduino in that it does not come pre-loaded with any modules.
- **ESP8266** The ESP8266 is a member of the same family as the ESP32, however it has less RAM and computing power.
- **PLC** PLCs require excessive labour for connecting wires as well. Finding errors is never easy, it takes a skilled team, and integrating sensors and communication devices with PLCs is much more challenging. Additionally, PLCs have a simple mechanism for acquiring data. It is also expensive.

3.2.4 Mobile App Options

A mobile application is a particular kind of application designed specifically for smartphones and tablets, sometimes referred to as mobile devices. For the development of mobile applications, there are several other solutions accessible, but they come with certain limitations. Below, we've discussed several alternatives.

- **Flutter** The code of Flutter application is same for both iOS and Android. But the syntax and characteristics of the language used by Flutter (Dart) are entirely different from those of JavaScript. It was quite straightforward to create the mobile application in React Native since it also uses JavaScript as its primary language. JS is more convenient for applications.
- **Java** Android applications can only be created using Java. Additionally, it takes a very long time to compile the application.
- **Swift** For iOS, iPadOS, Swift is a powerful and user-friendly programming language. Swift offers contemporary features that developers like, and writing Swift code is interactive and enjoyable. The syntax is short yet expressive. Swift code is secure by default and results in blazing-fast applications.

3.2.5 Database Options

A database is a well-organized collection of data that has been collected and is generally kept on a server. Below, we've covered a variety of other database alternatives.

- **Firebase** Data migration isn't very easy using Firebase. Additionally, as it's a platform created by Google, using the capability requires purchasing Firebase storage.
- **Oracle** Oracle databases are very complex. Working with Oracle database requires very strong technical experience in databases management.

- **PostgreSQL** PostgreSQL is slower than MySQL. Also, it requires additional drivers to integrate with PHP.

Chapter 4

Results and Discussion

The results of a real-time industrial monitoring system case study are discussed in this chapter. Data from sensors and all other data analysis are addressed in this chapter. The front-end of the system looks like dashboard views.

4.1 Validation experiment

Since the case study was about the automation of plant's equipment with several sensors. So, first of all we developed a minimal prototype model of embedded system to validate whether the automation problem was solvable or not. The system was consisted on a one chip computer called Arduino UNO, temperature sensor, humidity sensor, and a motion detector sensor. All the sensors were integrated on breadboard and finally the breadboard was connected with Auduino UNO using colored wires. After integration of all components, the Arduino was connected with computer via USB cable. After running this model the sensors data was streaming on output screen of computer. Figure 4.1 illustrates the first prototype we have worked on.

4.1.1 Outcomes

Objective of this experiment was to integrate sensors with one chip computer and get the meaningful information from sensors and display it on web platform. We achieved integration of hardware components of model, as well as got the data coming from sensors and also visualized it on web platform in the form of graphs as well as numbers. Figure 4.2 illustrates the visualization of MATLAB based computations in the form of graphs and Figure 4.3 illustrates the sensors' data in the form of numbers.

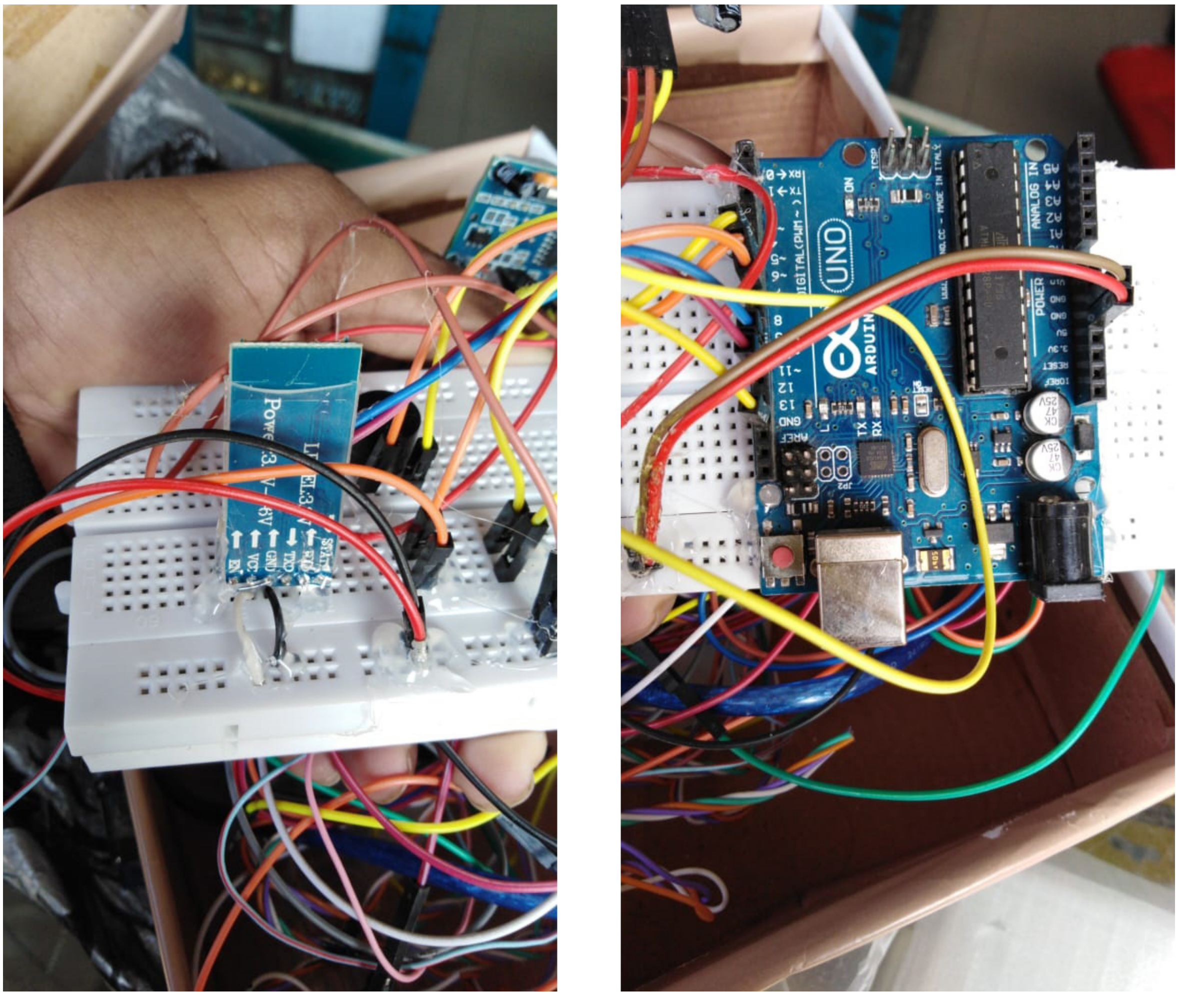

Figure 4.1: Minimal prototype circuit with Arduino and breadboard

4.2 Principal Experiment

As we have discussed that, the automation problem was originated from a real-time case study of Fazal Steel Mill. Their requirements was to automate the whole process of the mill with real time monitoring applications. So we have explored various case studies near to our problem, and also explored different hardware and software devices

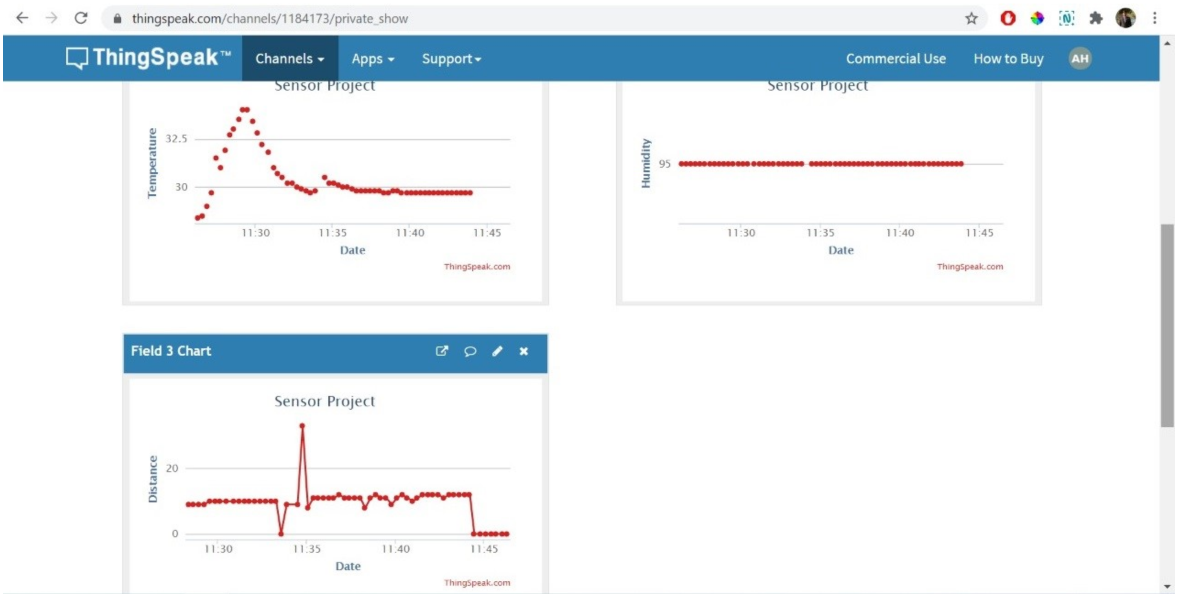

Figure 4.2: MATLAB based computational graphs on Thingspeak

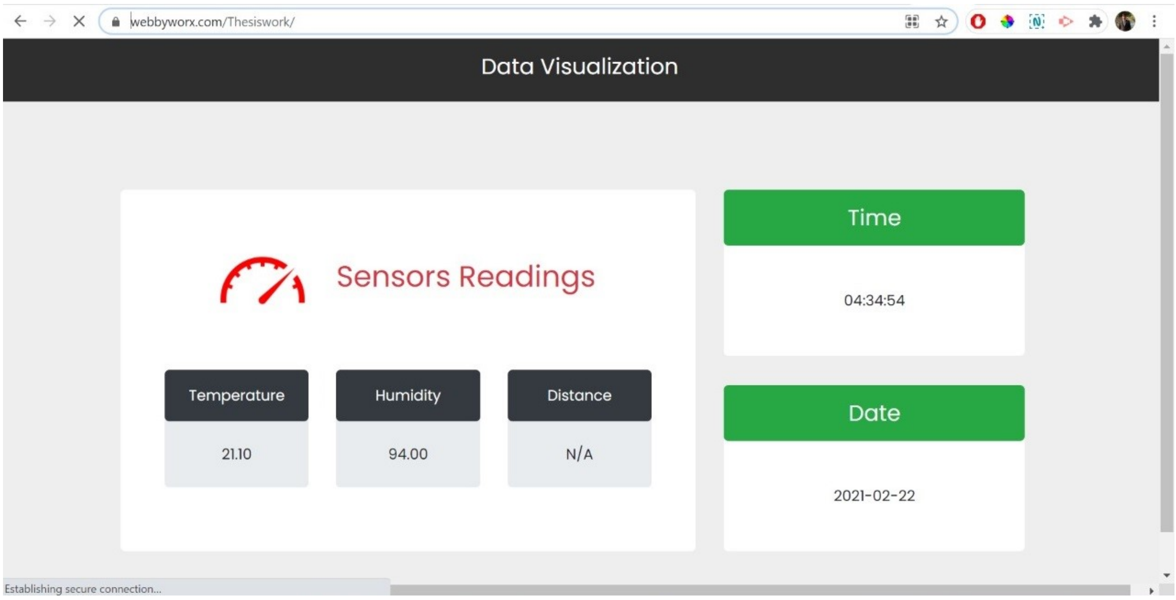

Figure 4.3: Sensors data in the forms of numbers visualized on initial web app

and components that we needed to use in our work. We have also discussed in methodology that, our system is consisted on three main building blocks: 1) Processor and communication block, 2) Back-end module, 3) Front-end module.

Firstly, the processor and communication is main building block which meant to be acted like a gateway communicator between front-end and back-end modules. Secondly, the aim of creating a back-end module was to acquire data from hardware, sensors and actuators and make this module to act like a database or data-bank. Thirdly, the aim of creating front-end application was not only to display the real-time data coming from sensors but also to display statistics like variation of estimated and actual parameters.

4.2.1 Processor and communication

Since we have mentioned in methodology, the device used for processing and communication is ESP32. ESP32 is a one chip computer that has built in WiFi modules embedded on it which we are using for data communication. First of all ESP32 receives data from sensors connected to it as well as from MATLAB simulation model. Secondly, after reception the data is being processed to get meaningful information to decipher whether the temperature, power consumption, operating time and composition of the equipment is in acceptable range or not. Thirdly, this information is being uploaded on database and the database is connected with mobile and web application. Lastly, on mobile and web application there are dashboards that display statistics of actual measured response as well as estimated response.

4.2.2 Back-end Module

There are two types of data used in this case study. First one is real time data called measured data that is coming from sensors, belongs to specific equipment like, furnace, tundish, and ladle. Second one is model based data called estimated data that is coming from MATLAB simulation algorithm. After acquisition data is being sent to ESP32 module that is used as a gateway channel between back-end and front-end module. Receives data from sensors and send it to database by using some database API, for further operations. Data is being saved in the form of tables. These tables are plant perspective table, the furnace view table, the ladle view table, and the tundish view table. The entries in each table are being updated with a pause of five seconds between each update. The reason for this is the code being used for the ESP-32 device gets update after every five minutes to avoid any dispute and to make sure the data is consistent. There are a variety of application programming interfaces (APIs) that are constructed using PHP that can be used to get data from each table and then give it to clients who have requested it. APIs are now being used to facilitate the transfer of data between two distinct categories of computing devices. The first application is a smartphone app and the second one is a web based application.

4.2.3 Front-end Module

Front-end of web application is built using ReactJS front-end language. And mobile application's front-end is built using ReactNative language. ReactJS and ReactNative languages use same back-end API to receive data from back-end. Both web and mobile applications are same in terms of visualization. These application results can be accessed anywhere either form home or in the office.

Web Application and Mobile Application

The web and mobile application are distributed in following interfaces.

1. Plant View (Home Screen)
2. Composition
3. Furnace View
4. Ladle View
5. Tundish View

Plant View

As observed, the screen is showing data from various sensors that have been given through the APIs that acquire data from database. Real-time data is fetched for each value. The Plant View updates promptly if a sensor's value changes. Additionally, there are graphs that present a graphical representation of the estimated energy losses of furnaces, ladles, and tundishes based on back-end calculations. All the data and calculated results can be displayed in a broad overview on the home screen (also known as Plant View). The plant view is depicted in Figure 4.4 and Figure 4.5.

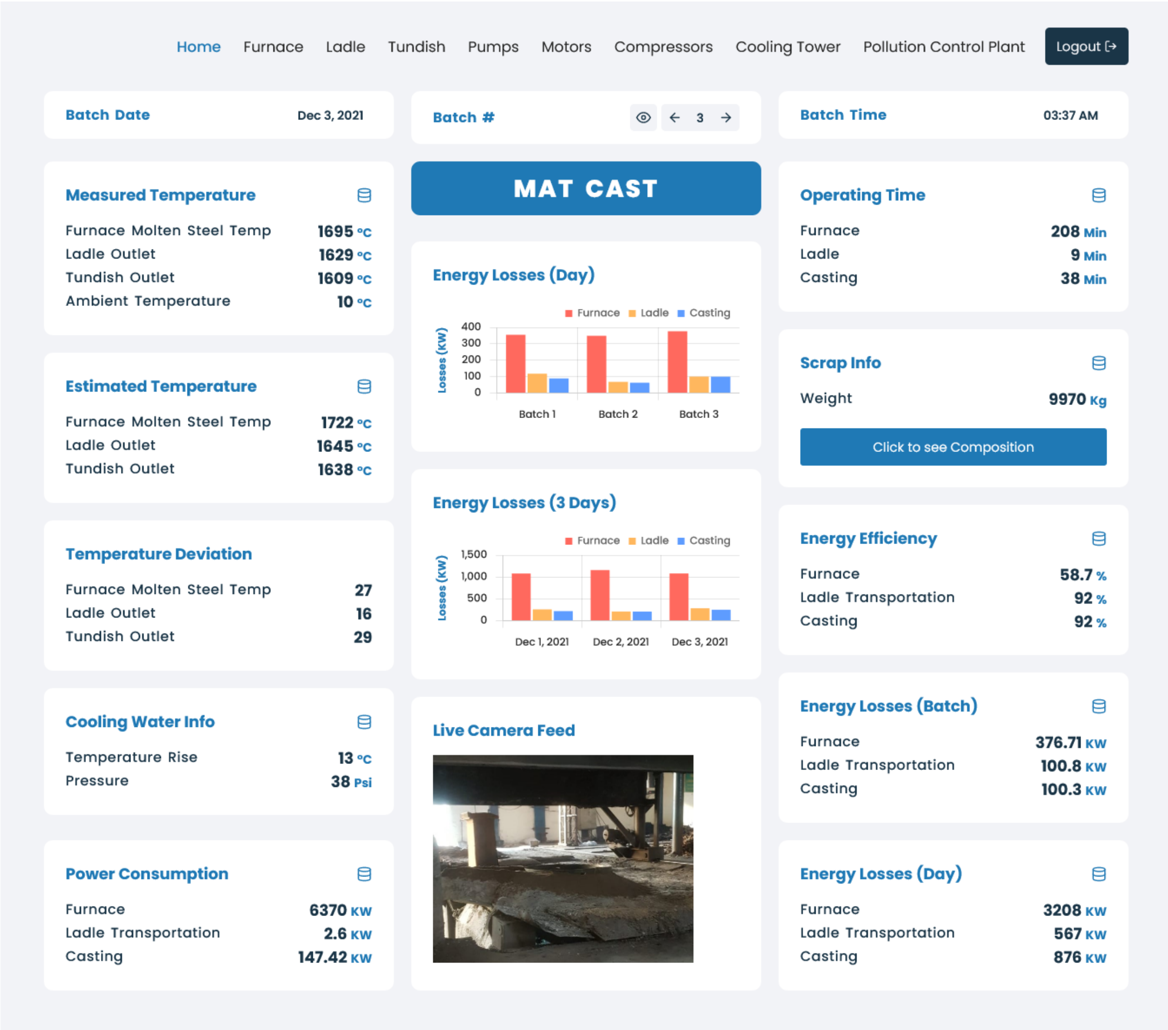

Figure 4.4: Web based plant overview of whole industrial monitoring process

Composition

In order to determine the composition of the elements in a sample, composition data is being acquired using the Spark Analyzer programme in the MATCAST lab. An API retrieves the data from that system, adds it to the database, and displays it in the Composition area. The data for composition are shown in Figure 4.6.

Additionally retrieved in real-time, this data includes a number of samples from several batches each day. Every day, various batches of composition data are collected, and different samples are taken for each batch. Each element's value from the current

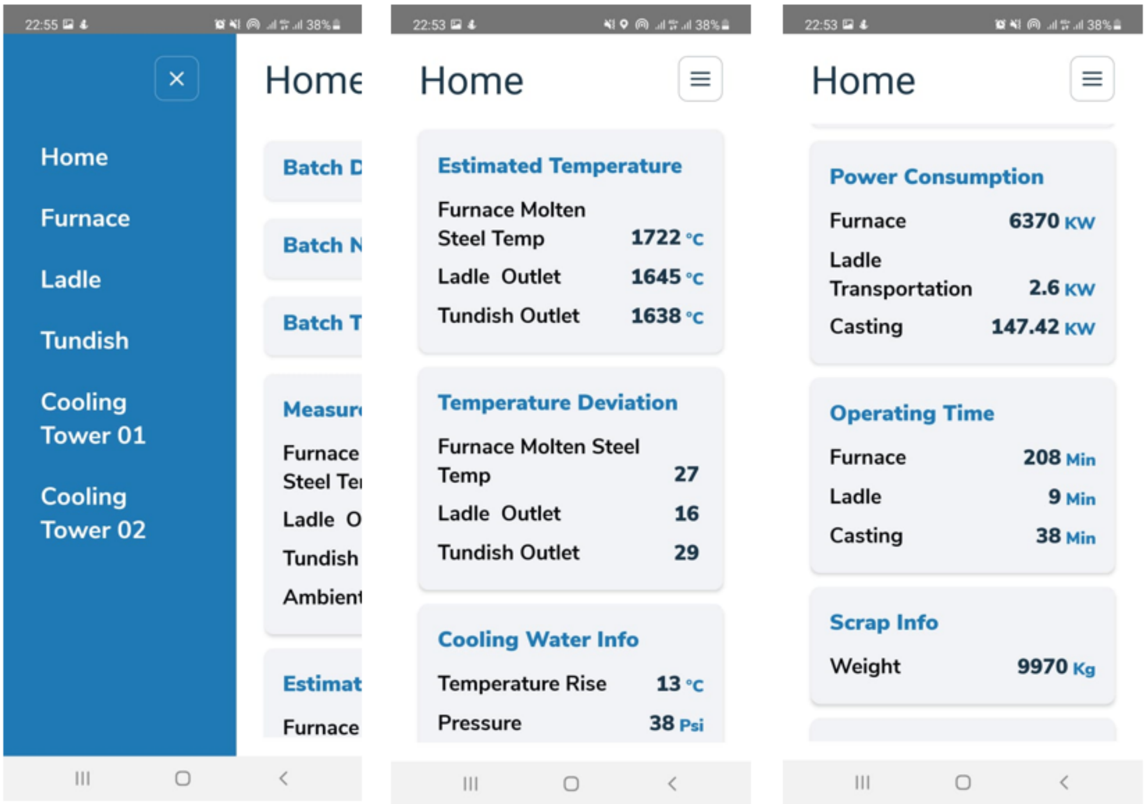

Figure 4.5: Plant view that displays whole process information

sample is represented by the data in the tables.

Furnace View

More thorough information on the furnace can be found in the furnace view. The molten steel temperature was one example of generic information about furnaces in Plant View, but this view has more precise calculations and data that almost entirely covers all aspects of furnace information. Furnace view of web app is is depicted in Figure 4.7 and mobile app is depicted in 4.8. As can be seen, the information for each batch comprises a lot of information on the furnace, including its running time, energy losses, and power usage. On the basis of this information, calculations are made, and the outcomes are used to create the graphs displayed in the Plant View.

Ladle View

The Ladle View contains extensive information regarding practically every component of Ladle, much like the Furnace View does. Similar to Furnace, this information is used

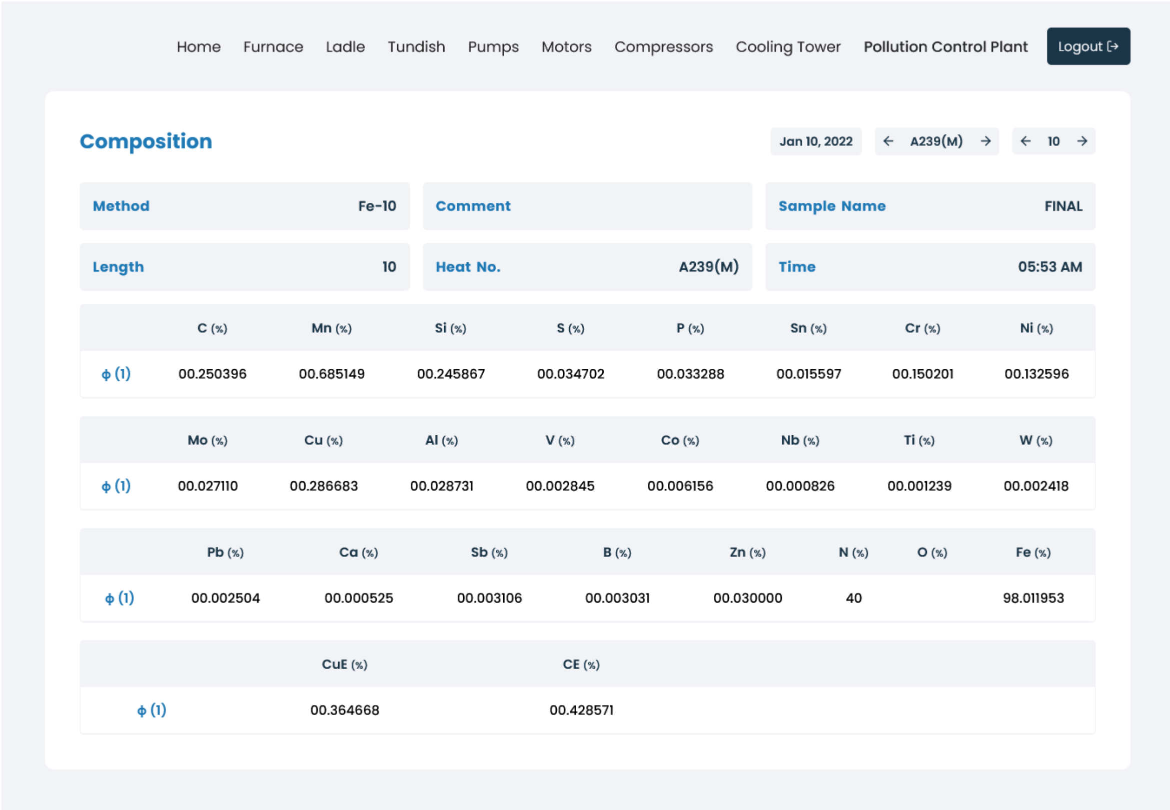

Figure 4.6: Composition view of web application

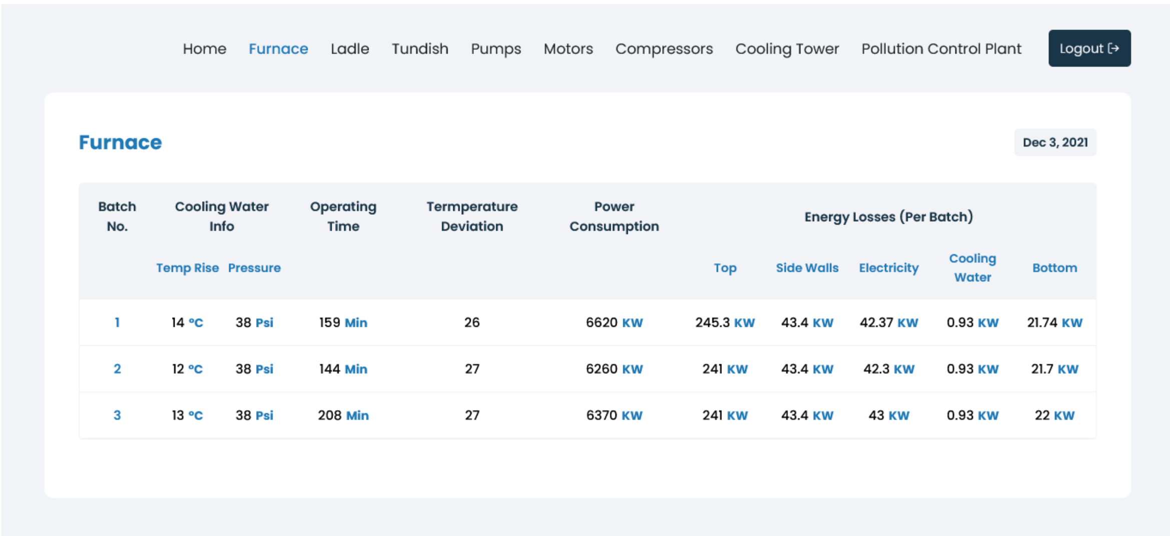

Figure 4.7: Furnace view of web application

in the calculations on the back end, and the outcomes are used in the graphs displayed in the Plant View. Figure 4.9 displays ladle view of web application and figure 4.10 shows ladle view of mobile application.

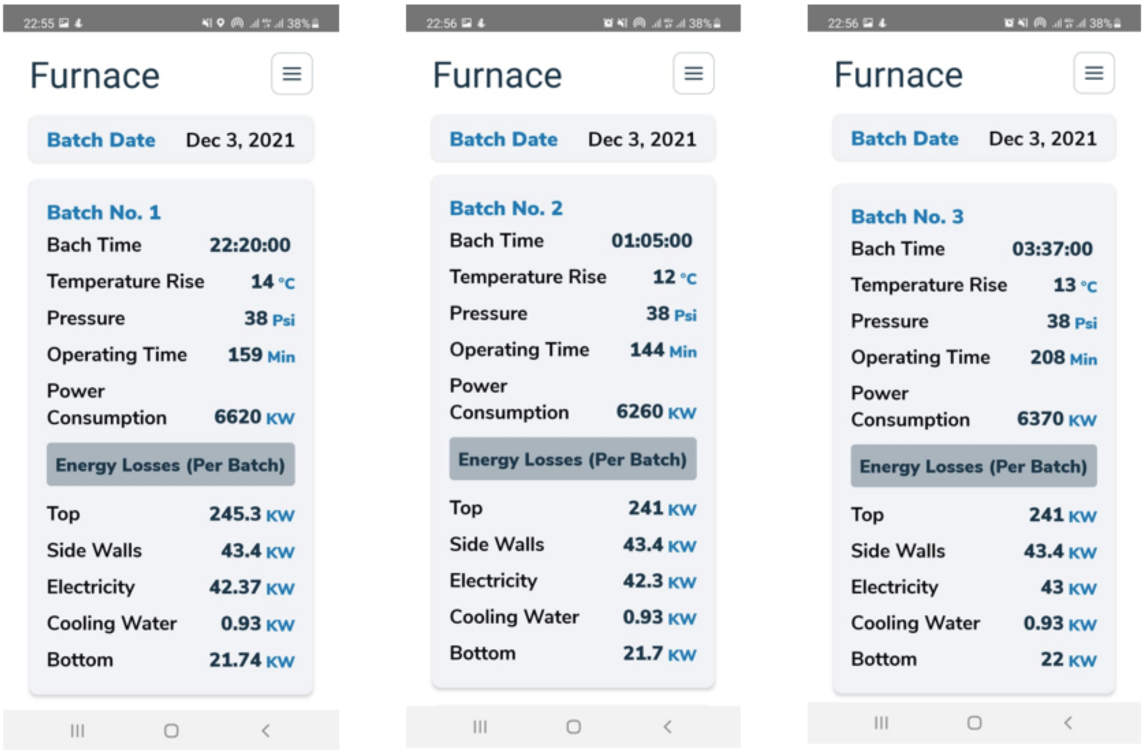

Figure 4.8: Furnace view of mobile application

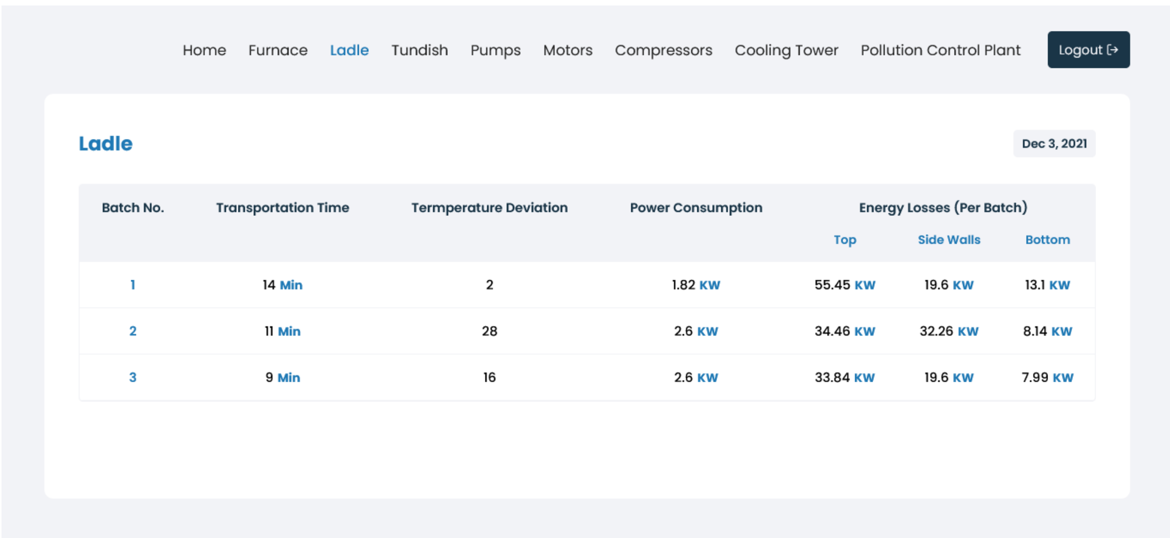

Figure 4.9: Ladle view of web application

Tundish View

The detailed information is available on Tundish View. The Tundish View of web application is seen in Figure 4.11. Information is used for calculations on the back end,

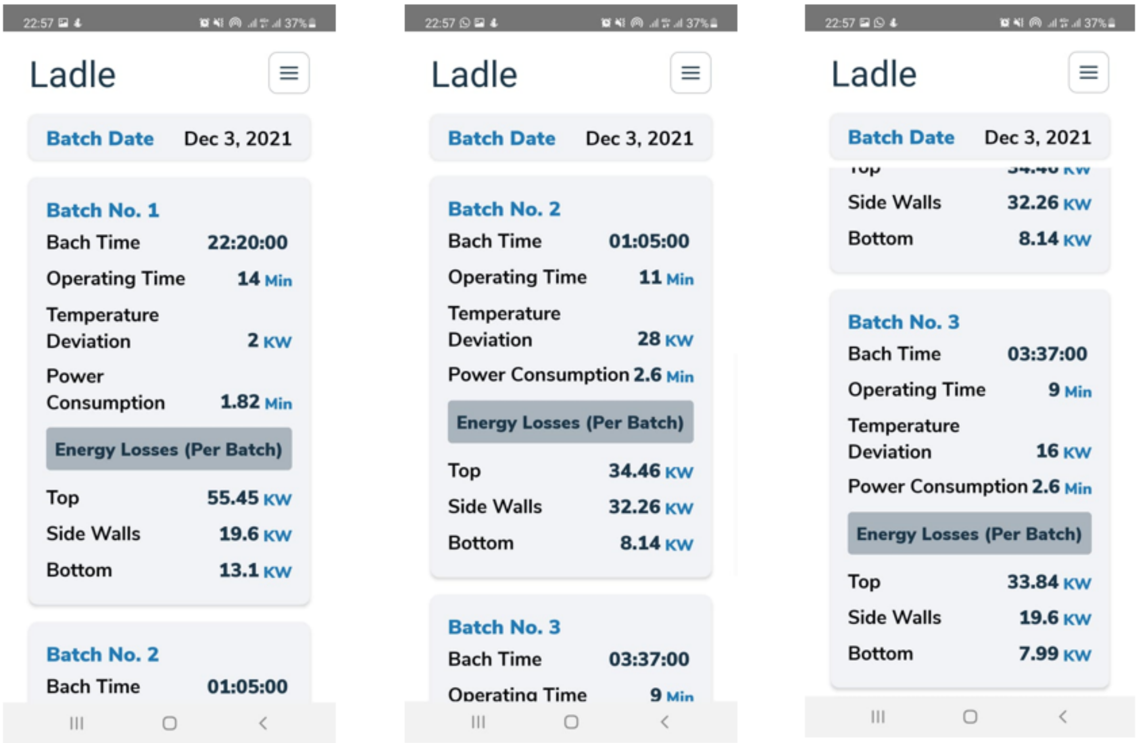

Figure 4.10: Ladle view of mobile application

and then outcomes are shown on front-end app. Figure 4.12 shows tundish view of mobile application.

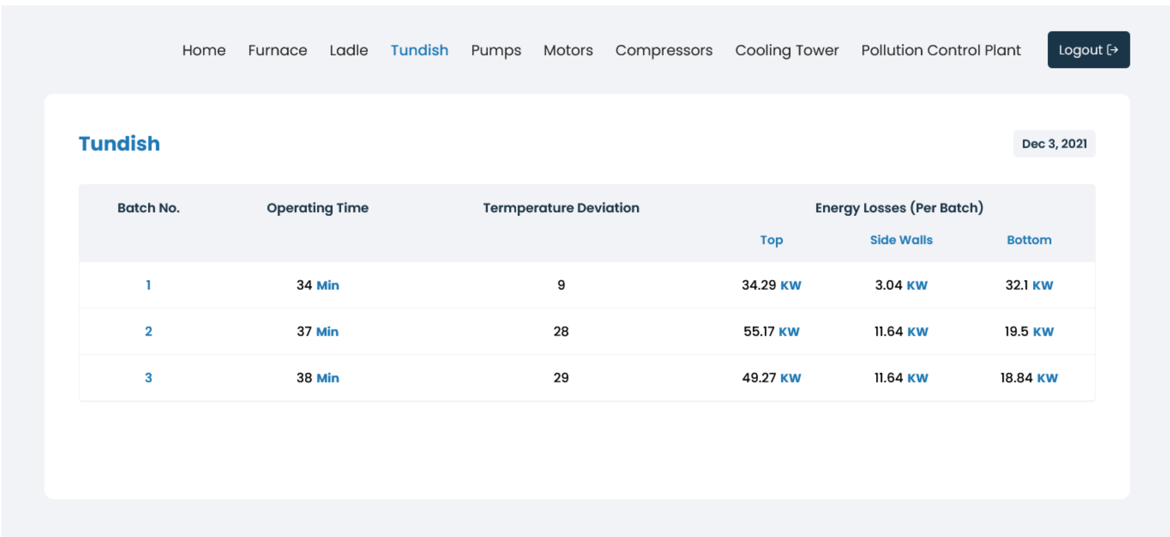

Figure 4.11: Tundish view of web application

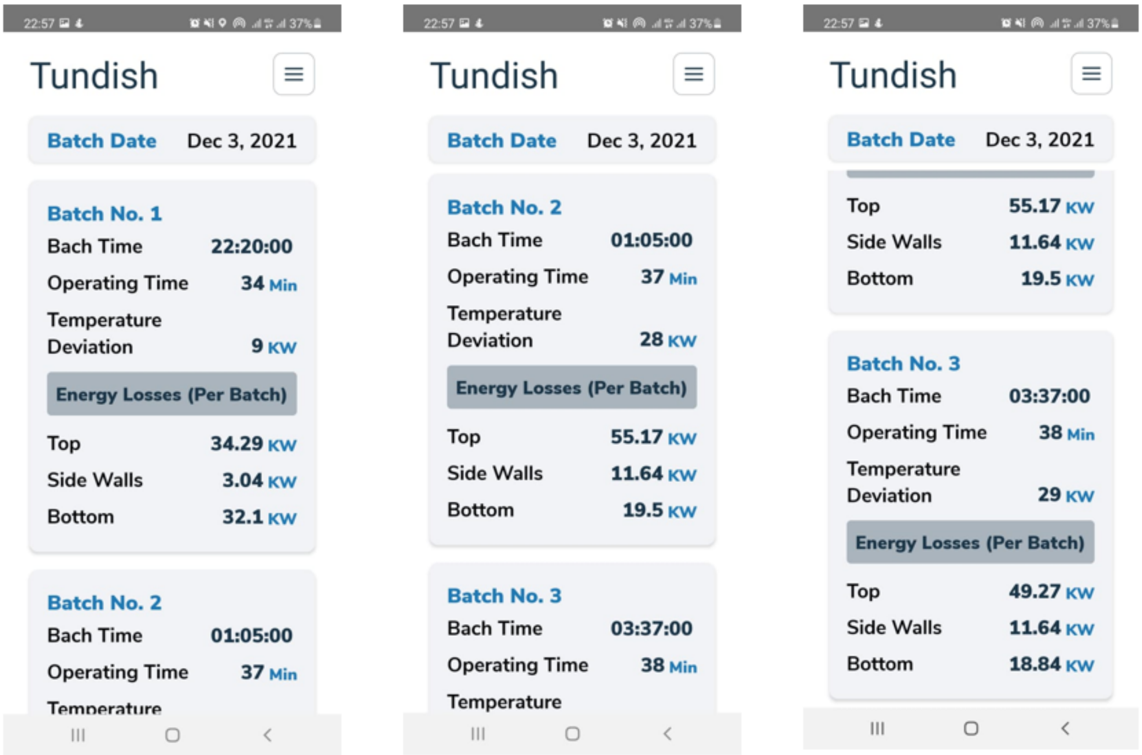

Figure 4.12: Tundish view of mobile application

4.3 Systems Engineering Perspective

The main goal of this study was to manage it according to the systems engineering principles. This was a complex problem because a lot of components and modules were included in it. All the modules used in it are from diverse domains for example, hardware such as sensors and single chip computers, software such as mobile application, web application and database application. All these components and modules are dependent on each other so, there are integrated in such a way that they can inter communicate wisely. These components are integrated with each other in such a way they can share information with each other. The complexity of the system is that all the components are dependent on each other.

Chapter 5

Conclusion

5.1 Conclusions

In this chapter, we'll conclude our work along with some recommendations for future directions.

Previously industries were working manually according to traditional set of rules. There was no efficient way or cost effective way to monitor their performance automatically. To address this issue we created an integrated industrial monitoring system that shows real-time updates of the machines running in industry. Significance of the system is that, it provides aid in decision making when something is not going well in industry. There is one plant view four major dashboard views: 1) furnace view, 2) ladle view, 3) composition view and 4) tundish view. All of the views hold their respective data such as temperature, power consumption, energy loss and operating time. Whereas plant view is main dashboard where only important information is displayed that is coming from sensors and MATLAB model, simply the plant view is the bird eye view of the whole process. It not only displays statistics but also displays video stream coming from an IP camera. When the temperature, composition or operating time consumption is not in acceptable range, the operators can visualize it on screen and can also see the video stream of the camera covering the equipment and can take necessary actions to avoid disasters. This system is not very costly, scale-able and user can view all the statistics in single screen.

If we talk about Systems Engineering perspective, this was a complicated issue since it had several parts and modules. Each module contains several components that are coming from a variety of fields, including hardware like sensors and single-chip processors and software like mobile applications, web applications, and database. Due to the interdependence of all these parts and modules, they have been integrated to allow for intelligent intercommunication.

5.2 Future directions

We have implemented this system to monitor everything happening in industry. We foresee, in future we will extend this system to automatically take actions on the bases of sensors' response. We will integrate our system with control switches so that automatically take switch on and off, shutdown and restart the equipment when something goes wrong. This is a standalone application and works for one industry but it can be extended in terms of network. In future we will work for a client server model of this application where all the applications of multiple industries will be connected to a server and the server side of the application will displays in important information regarding all client applications.

References

[1] Ricardo Matheus, Marijn Janssen, and Devender Maheshwari. Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities. *Government Information Quarterly*, 37(3), jul 2020. ISSN 0740624X. doi: 10.1016/j.giq.2018.01.006.

[2] Youssef Cheddadi, Hafsa Cheddadi, Fatima Cheddadi, Fatima Errahimi, and Najia Es-sbai. Design and implementation of an intelligent low-cost IoT solution for energy monitoring of photovoltaic stations. *SN Applied Sciences*, 2(7):1–11, 2020. ISSN 25233971. doi: 10.1007/s42452-020-2997-4. URL https://doi.org/10.1007/s42452-020-2997-4.

[3] Lawrence Oriaghe Aghenta and Mohammad Tariq Iqbal. Low-cost, open source IoT-based SCADA system design using thinger.IO and ESP32 thing. *Electronics (Switzerland)*, 8(8), aug 2019. ISSN 20799292. doi: 10.3390/electronics8080822.

[4] Siwakorn Jindarat and Pongpisitt Wuttidittachotti. Smart farm monitoring using Raspberry Pi and Arduino. *I4CT 2015 - 2015 2nd International Conference on Computer, Communications, and Control Technology, Art Proceeding*, (I4ct):284–288, 2015. doi: 10.1109/I4CT.2015.7219582.

[5] Marcel Nicola, Claudiu-Ionel Nicola, Marian Duță, and Dumitru Sacerdoțianu. SCADA Systems Architecture Based on OPC and Web Servers and Integration of Applications for Industrial Process Control. *International Journal of Control Science and Engineering*, 8(1):13–21, 2018. doi: 10.5923/j.control.20180801.02. URL http://journal.sapub.org/control.

[6] Shatadru Bipasha Biswas and M. Tariq Iqbal. Solar Water Pumping System Control Using a Low Cost ESP32 Microcontroller. In *Canadian Conference on Electrical and Computer Engineering*, volume 2018-May. Institute of Electrical and Electronics Engineers Inc., aug 2018. ISBN 9781538624104. doi: 10.1109/CCECE.2018.8447749.

[7] Ch Rajendra Prasad. Internet of Things Based Home Monitoring and Device Control Using Esp32 Smart System Design View project Patents View project. Technical Report 8, 2019. URL https://www.researchgate.net/publication/334226986.

[8] Obudai Egyetem, IEEE Hungary Section, and IEEE Industrial Electronics Society. *INES 2020 : IEEE 24th International Conference on Intelligent Engineering Systems : proceedings : July 8-10, 2020, Reykjavik, Iceland*. ISBN 9781728110592.

[9] V. Rajesh, J. M. Gnanasekar, R. S. Ponmagal, and P. Anbalagan. Integration of wireless sensor network with cloud. *ITC 2010 - 2010 International Conference on Recent Trends in Information, Telecommunication, and Computing*, pages 321–323, 2010. doi: 10.1109/ITC.2010.88.

[10] V. Rajesh, J. M. Gnanasekar, R. S. Ponmagal, and P. Anbalagan. Integration of wireless sensor network with cloud. *ITC 2010 - 2010 International Conference on Recent Trends in Information, Telecommunication, and Computing*, pages 321–323, 2010. doi: 10.1109/ITC.2010.88.

[11] Shaoiun Gan, Kang Li, Yanxia Wang, and Che Cameron. IoT Based Energy Consumption Monitoring Platform for Industrial Processes. *2018 UKACC 12th International Conference on Control, CONTROL 2018*, pages 236–240, 2018. doi: 10.1109/CONTROL.2018.8516828.

[12] Juan Aponte-Luis, Juan Antonio Gómez-Galán, Fernando Gómez-Bravo, Manuel Sánchez-Raya, Javier Alcina-Espigado, and Pedro Miguel Teixido-Rovira. An efficient wireless sensor network for industrial monitoring and control. *Sensors (Switzerland)*, 18(1), 2018. ISSN 14248220. doi: 10.3390/s18010182.

[13] Dae Heon Park and Jang Woo Park. Wireless sensor network-based greenhouse environment monitoring and automatic control system for dew condensation prevention. *Sensors*, 11(4):3640–3651, 2011. ISSN 14248220. doi: 10.3390/s110403640.

[14] Yingli Zhu, Jingjiang Song, and Fuzhou Dong. Applications of Wireless Sensor Network in the agriculture environment monitoring. *Procedia Engineering*, 16: 608–614, 2011. ISSN 18777058. doi: 10.1016/j.proeng.2011.08.1131. URL http://dx.doi.org/10.1016/j.proeng.2011.08.1131.

[15] Fábio A. Ferreira Marques, David M.D. Ribeiro, Márcio F.M. Colunas, and João P. Silva Cunha. A real time, wearable ECG and blood pressure monitoring system. *Proceedings of the 6th Iberian Conference on Information Systems and Technologies, CISTI 2011*, 2011.

[16] Sameer Sonkusale. *Sensors for Vital Signs: ECG Monitoring Systems*, pages 221–243. Springer New York, New York, NY, 2022. ISBN 978-1-4614-3447-4. doi: 10.1007/978-1-4614-3447-4_2. URL https://doi.org/10.1007/978-1-4614-3447-4_2.

[17] Jesse R Codling, Ann Arbor, and Luke F Cohen. Poster Abstract : SeatBeats Heart Rate Monitoring System using Structural Seat Vibrations. pages 511–512, 2022. doi: 10.1109/IPSN54338.2022.00056.

[18] Lindsay Brown, Jef Van De Molengraft, Refet Firat Yazicioglu, Tom Torfs, Julien Penders, and Chris Van Hoof. A low-power, wireless, 8-channel EEG monitoring headset. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10*, pages 4197–4200, 2010. doi: 10.1109/IEMBS.2010.5627393.

[19] Elisa Spanò, Stefano Di Pascoli, and Giuseppe Iannaccone. Low-Power Wearable ECG Monitoring System for Multiple-Patient Remote Monitoring. *IEEE Sensors Journal*, 16(13):5452–5462, 2016. ISSN 1530437X. doi: 10.1109/JSEN.2016.2564995.

[20] Shuxing Du, Bo Liu, Huixun Ma, Guoying Wu, and Peng Wu. IIOT-Based intelligent control and management system for motorcycle endurance test. *IEEE Access*, 6:30567–30576, 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2841185.

[21] Komkrit Chooruang and Kraison Meekul. Design of an IoT Energy Monitoring System. *International Conference on ICT and Knowledge Engineering*, 2018-November:48–51, 2019. ISSN 2157099X. doi: 10.1109/ICTKE.2018.8612412.

[22] R. A. Ramlee, M. A. Othman, M. H. Leong, M. M. Ismail, and S. S.S. Ranjit. Smart home system using android application. *2013 International Conference of Information and Communication Technology, ICoICT 2013*, pages 277–280, 2013. doi: 10.1109/ICoICT.2013.6574587.