CHAPTER 1: INTRODUCTION

CHAPTER 1

INTRODUCTION

The global dairy industry faces unprecedented challenges in maintaining consistent milk quality, ensuring food safety, and optimizing production processes. Traditional milk quality assessment methods rely heavily on manual interventions, which are inherently limited by human capabilities, subjective interpretations, and time-consuming procedures. These conventional approaches often fail to provide real-time, comprehensive insights into milk's complex compositional dynamics.

Technological advancements in IoT and machine learning have emerged as transformative solutions capable of addressing these fundamental limitations. By integrating sophisticated sensor networks, intelligent data processing algorithms, and cloud-based infrastructure, modern technological ecosystems can revolutionize milk quality monitoring, offering unprecedented levels of precision, efficiency, and reliability.

The milk quality analysis system represents a paradigm shift in agricultural monitoring technologies. Unlike traditional approaches that focus on isolated quality parameters, this integrated solution provides a multidimensional assessment framework that simultaneously evaluates chemical composition, environmental factors, and potential contamination risks through advanced sensing and computational techniques.

Milk, being a highly complex and perishable biological product, requires continuous and sophisticated monitoring to maintain its nutritional integrity and ensure consumer safety. The proposed system addresses this critical need by creating a comprehensive technological infrastructure that can detect subtle variations in milk quality with remarkable accuracy and speed.

The integration of ESP32 microcontrollers serves as the technological backbone of the system, enabling seamless sensor data collection, processing, and transmission. These powerful microcontrollers provide the computational flexibility and connectivity required to create a robust, adaptive milk quality monitoring ecosystem that can operate across diverse environmental conditions.

1.1. PHASE DESCRIPTION

Phase	Task	Description
Phase 1	Analysis	Analyzing the core of the IEEE paper and providing a Literature review based on analysis.
	Literature Survey	Collect raw data and elaborate on literature surveys.
	System Analysis	Analyses the requirements of the project and lists the specific requirements needed.
	System Design (High-Level Design)	Overall design architecture.
Phase 2	System Design (Detailed -Level Design)	Required software architecture, source code, and performance models
	Implementation	Implement the code based on the object specification.
	Testing	Test the project according to the test specification.
	Documentation	Prepare the document for this project with the conclusion and future enhancement.

Table 1 Project Workflow

1.2. ORGANIZATION OF PROJECT REPORT

The project report is organized as follows:

Chapter 1:	**Preamble** - Gives a brief introduction of the project.
Chapter 2:	**Literature Survey** - Gives a brief overview of the survey papers and the research sourcesthat have been studied to establish a thorough understanding of the project under consideration.
Chapter 3:	**System Analysis -** Studying the existing system, arguments in favor and against theexisting solutions, defining problem statement, Objectives, proposing a new system with the advantages
Chapter 4:	**Hardware and Software Requirements** - Discusses in detail the different kinds ofrequirements needed to complete the project.
Chapter 5:	**System Design** - Gives the design description of the project, conceptual and detaileddesign well supported with design diagrams.
Chapter 6:	**Implementation** – Discusses the implementation details of the project and reasons for the use of the programming language and developmentenvironment.
Chapter 7:	**Testing** – Briefs on the testing methods used for testing the different modulesin theproject.
Chapter 8:	**Results and Performance Analysis** – Gives the snapshots and graphs of theproposedprotocols.
Chapter 9:	**Snapshots** – Snapshots of the working project.

Table 2 Organization of the report.

Conclusion- Gives the concluding remarks of the project, throwing light on its future aspects.

References: Lists the research papers, blogs, books, etc. referred to during the project work

CHAPTER 2:
LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

OVERVIEW

The literature review offers an in-depth exploration of recent advancements and research in the field of milk quality analysis, focusing on the integration of various technologies and methods to ensure milk safety and nutritional quality. It examines the global developments in milk quality monitoring systems, highlighting innovations such as **automated milk testing devices**, **infrared spectroscopy**, and **biosensors**. These technologies have greatly improved the efficiency and accuracy of milk analysis, revolutionizing the dairy industry by enabling real-time monitoring of milk quality, reducing the risk of contamination, and improving product consistency.

The review also addresses the challenges faced by the dairy industry in regions like **India**, where traditional milk quality testing methods are often manual and labor-intensive. It identifies significant drawbacks of the current techniques, including the lack of automation, high human error rates, and delays in results, which can affect the overall safety and quality of milk. Given the rapidly growing dairy demand in India, the review emphasizes the urgent need for **technological advancements** in milk quality testing to ensure better consumer safety and industry standards.

A major area of focus is the application of **infrared spectroscopy** and **biosensors** for the non-destructive analysis of milk's chemical composition, including fat, protein, and lactose content with milk quality analyzers to enhance the precision of these systems in detecting contaminants like antibiotics, pesticides, and heavy metals. These technologies not only improve the accuracy of measurements but also reduce testing time, making them ideal for large-scale dairy operations.

The adaptability of **visual computing** and **robotic systems** in automating milk quality analysis offers significant potential for future advancements in dairy processing.

[1] Title: Smart Milk Quality Analysis & Grading

Authors: S.B. Dighe, Utkarsha Kailas, Bonde Snehadip Girip, and Verma Snehika

Published year:2024

This paper introduced a comprehensive framework titled "Smart Milk Quality Analysis & Grading." Their approach integrates advanced sensor technologies to provide innovative solutions for milk quality evaluation. By leveraging cutting-edge technology, the system aims to enhance the grading and quality assessment process, ensuring reliability for consumers and producers alike. However, the primary limitation lies in refining the accuracy of the system, which is crucial for its adoption on a larger scale. The research highlights the potential of sensor-based solutions while calling for further optimization to enhance performance and adaptability.

[2] Title: Milk Quality Prediction using Machine Learning

Authors: Drashti Bhavsar, Yash Jahanpura, Vimal Kehani, and Debabrata Swain

Published year:2023

This paper explored "Milk Quality Prediction using Machine Learning." This research focuses on data preprocessing and employs sophisticated machine learning models like Random Forest and Support Vector Machines (SVMs) to predict milk quality. Their methodology demonstrated substantial promise in improving prediction accuracy, especially in detecting subtle variations in milk quality parameters. However, a notable drawback emerged: the system requires relatively longer processing times when dealing with smaller datasets. This limitation suggests that computational efficiency remains a critical area for improvement. The study emphasizes the potential of machine learning in milk quality analysis but underscores the need for scalable and efficient algorithms.

[3] Title: IoT-Based Detection of Adulteration in Milk

Authors: **K. Lalesh Givot, S. Siddon, P. Mina Choudhary, P. Putalakota, and S. Reddy**

Published year: 2023

This paper focused their efforts on developing an IoT-based solution for milk adulteration detection. Their study, titled "IoT-Based Detection of Adulteration in Milk," leverages Arduino Uno microcontrollers and portable sensors to create a real-time detection system. The approach is both innovative and practical, showcasing the versatility of IoT in addressing food safety concerns. However, the system encounters challenges in achieving complex accuracy under real-time conditions, a limitation that could affect its reliability in diverse environments. The research highlights the potential of IoT-based solutions while stressing the importance of enhanced calibration techniques and algorithmic improvements to achieve higher precision.

[4] Title:Smart Milk Quality Analysis and Grading Using IoT

Authors: **Manuhik Shubham, Telang Vibhuti, Kaduskar Pooja, and Mr. P.S. Bibye**

Published year:2023

This paper focused on "Smart Milk Quality Analysis and Grading Using IoT." This study utilized ESP32 microcontrollers, pH sensors, and ultrasonic sensors to evaluate various milk quality parameters, providing a multifaceted approach to milk quality analysis. This system demonstrated its capability to integrate multiple sensors for holistic quality assessment. However, the inability to accurately predict temperature—a critical parameter in milk grading—was identified as a major limitation. This gap highlights the need for incorporating specialized temperature sensors or further refining the existing hardware to improve accuracy. Despite these challenges, the study provides a robust foundation for future advancements in IoT-based milk quality systems.

[5] Title:An Assessment of Milk Adulteration IoT-Based Model to Identify Milk in Lab

Authors: R. Medha Kharwan, S. Swathi, G. Ashwini, and G. Sriprada

Published year : 2022

This paper conducted research titled **"An Assessment of Milk Adulteration IoT-Based Model to Identify Milk in Labs."** This study highlighted the innovative use of IoT (Internet of Things) in identifying adulteration in milk. The system integrated advanced sensors for real-time data collection and analysis. Automated alerts ensured that stakeholders were promptly informed of any discrepancies in milk quality. The research emphasized the potential of sensor integration in enhancing precision, reliability, and automation in dairy testing. However, the high cost of implementing this technology posed a limitation, particularly for small-scale dairies and rural settings. The study suggests exploring cost-efficient solutions, such as low-cost sensor technology or modular IoT systems that can be scaled based on budgetary constraints. Furthermore, the research underscored the importance of collaboration between tech providers and dairy industries to make such innovations accessible on a larger scale.

[6] Title: Analysis and Design of Fresh Milk Smart Grading System Based on IoT

Authors: S. Habibani, F. Uddin, and A. Akberman

Published year: 2022

This paper introduced the **"Analysis and Design of Fresh Milk Smart Grading System Based on IoT."** Their approach used Artificial Neural Networks (ANNs) and K-means clustering algorithms to classify milk based on quality and freshness indicators. By selecting and analyzing key features, such as temperature, pH levels, and contamination markers, the system demonstrated accurate grading capabilities. The IoT integration enabled real-time data transmission, making the system adaptable for large-scale applications in milk processing units. Despite its merits, the study highlighted **data security and privacy** as significant concerns. Since IoT-based systems often transmit sensitive data over networks, ensuring secure communication channels and protecting the data from breaches is critical. Future implementations could focus on blockchain integration or end-to-end encryption techniques to address these issues. Additionally, training personnel on secure usage of such systems would help mitigate vulnerabilities.

[7] Title: Efficient Milk Quality Detection System with IoT

Authors: H. Madan Gouda, Nayana P, Pardeep J, P. Kumar, Y. Sudhanva, and S. B. N,

Published year: 2022

This paper explored the **"Efficient Milk Quality Detection System with IoT."** This system utilized microcontrollers to handle sensor data and integrated automated payment processing for seamless supply chain management. The combination of milk quality detection and transactional functionalities was a unique contribution, making the system ideal for dairy farms, cooperatives, and retail outlets. By reducing human intervention, the system enhanced operational efficiency and minimized errors. However, challenges such as **sensor breakdowns, high costs, and frequent maintenance requirements** were significant drawbacks. These issues highlight the need for robust sensor designs, modular components, and predictive maintenance strategies to ensure smooth operation. Future research could also explore using AI-driven diagnostic tools to monitor sensor health and predict failures before they occur.

[8] Title: Automated Milk Quality Analytical with Billing System

Authors: Manjashree N. and Raveesh S.J.

Published year: 2022

This paper presented the **"Automated Milk Quality Analytical with Billing System,"** which integrated milk analysis with billing functionalities. Using an LCD display, an MCU-integrated circuit, and a mini-computer, the system allowed users to view real-time milk quality information and process payments simultaneously. This dual-purpose system was particularly beneficial for small businesses and retail operations, offering an all-in-one solution. However, its limitation lay in the requirement for **individual testing of components** to maintain accuracy. This added complexity and time to the deployment process, making it less practical for large-scale operations. Simplifying the design by incorporating self-calibrating components or automated testing mechanisms could significantly improve its scalability and ease of use. Additionally, the integration of advanced analytics tools, such as predictive quality monitoring, would further enhance the system's utility

[9] Title: Arduino-Based Milk Quality Monitoring System

Authors:Sunitha Gokavi and **Ashok Dangi**

Published year: 2022

In recent years, advancements in technology have led to the development of several innovative systems for milk quality monitoring and detection, integrating IoT and sensor- based methodologies, leveraging Arduino Uno and sensors to monitor milk quality parameters. However, the system's lack of automated feedback limits its efficiency. This added complexity and time to the deployment process, making it less practical for large- scale operations. Simplifying the design by incorporating self-calibrating components or automated testing mechanisms could significantly improve its scalability and ease of use. Additionally, the integration of advanced analytics tools, such as predictive quality monitoring, would further enhance the system’s utility

[10] Title: IoT-Based Milk Monitoring System

Authors: Hema V., **Monish Kumar N.**, **N. Nandhakumar**, and **B. Narmadha**

Published year: 2022

This paper utilizing BUZZR for audio signals and GSM modules for data transmission (TDMA, CDMA). Although effective, the system faced control issues when operating at different levels. The combination of milk quality detection and transactional functionalities was a unique contribution, making the system ideal for dairy farms, cooperatives, and retail outlets. By reducing human intervention, the system enhanced operational efficiency and minimized errors.Since IoT-based systems often transmit sensitive data over networks, ensuring secure communication channels and protecting the data from breaches is critical. Future implementations could focus on blockchain integration or end-to-end encryption techniques to address these issues. Additionally, training personnel on secure usage of such systems would help mitigate vulnerabilities.

[11] Title: IoT-Based Milk Quality System Using AVR Microcontrollers

Authors: Prof. Dipti S. Bhale, **Shraddha J. Dobaiya**, and **Sanchitha S.**

Published year: 2022

This paper integrating pH and temperature sensors alongside color sensors for fat detection. Despite its advanced detection capabilities, the complexity and cost of combining these systems with payment mechanisms pose challenges for widespread adoption.This added complexity and time to the deployment process, making it less practical for large-scale operations. Simplifying the design by incorporating self-calibrating components or automated testing mechanisms could significantly improve its scalability and ease of use. Additionally, the integration of advanced analytics tools, such as predictive quality monitoring, would further enhance the system's utility

[12] Title: IoT-Based Milk Monitoring System for Adulteration Detection

Authors: Dr. G. Rajaguru, **Dr. T. Bharath Kumar**, and **Dr. S. Atilan Samuel**

Published year: 2022

This paper presented an **IoT-Based Milk Monitoring System for Adulteration Detection** in 2018, employing Arduino microcontrollers and viscosity measurement techniques. However, the system was criticized for its time-consuming setup and implementation.Their methodology demonstrated substantial promise in improving prediction accuracy, especially in detecting subtle variations in milk quality parameters. However, a notable drawback emerged: the system requires relatively longer processing times when dealing with smaller datasets. This limitation suggests that computational efficiency remains a critical area for improvement. The study emphasizes the potential of machine learning in milk quality analysis but underscores the need for scalable and efficient algorithms.

[13] Title: IoT-Based Temperature Monitoring System

Authors: Setyawan P. Sakti, **Tri Swantoro Putro**, **Devi Anggraini**, **Agis Naba**, and **Vany R. Wijayanto**

Published year: 2021

This Research paper designed an **IoT-Based Temperature Monitoring System** for milk tanks in remote areas. This system relied on sensors for data collection and mobile networks for communication, but its inability to predict temperature accuracy remains a drawback.

This added complexity and time to the deployment process, making it less practical for large-scale operations. Simplifying the design by incorporating self-calibrating components or automated testing mechanisms could significantly improve its scalability and ease of use. Additionally, the integration of advanced analytics tools, such as predictive quality monitoring, would further enhance the system's utility

[14] Title: Milk Adulteration Detection in Labs

Authors: R. Medha Kharwan, **S. Swathi**, **G. Ashwini**, and **G. Sripada**

Published year: 2019

This paper developed a model for **Milk Adulteration Detection in Labs**, focusing on sensor integration, automated alerts, and data analysis. Although effective, the high implementation cost limits its scalability.Their methodology demonstrated substantial promise in improving prediction accuracy, especially in detecting subtle variations in milk quality parameters. However, a notable drawback emerged: the system requires relatively longer processing times when dealing with smaller datasets. This limitation suggests that computational efficiency remains a critical area for improvement. The study emphasizes the potential of machine learning in milk quality analysis but underscores the need for scalable and efficient algorithms.

[15] Title: IoT-Based Milk Monitoring System

Authors: Dr. P. Ganesh Kumar, **A. Alagammai**, **B. S. Madhumita**, and **B. Ishwarya**

Published year: 2019

This paper introduced an **IoT-Based Milk Monitoring System** using Arduino UNO and pH sensors, which, while functional, offers limited features.Their methodology demonstrated substantial promise in improving prediction accuracy, especially in detecting subtle variations in milk quality parameters. However, a notable drawback emerged: the system requires relatively longer processing times when dealing with smaller datasets. This limitation suggests that computational efficiency remains a critical area for improvement.

The study emphasizes the potential of machine learning in milk quality analysis but underscores the need for scalable and efficient algorithms

[16] Title: IoT-Based Milk Analysis System Using Arduino and Mobile Integration

Authors: Kalairasi Ganesan, **Ashok John**, and **K. Dinakumar**

Published year:2019

This paper also introduced an utilizing Think Speak software to monitor milk parameters. However, this system also suffers from limited features and less accurate data analysis.This added complexity and time to the deployment process, making it less practical for large- scale operations. Simplifying the design by incorporating self-calibrating components or automated testing mechanisms could significantly improve its scalability and ease of use. Additionally, the integration of advanced analytics tools, such as predictive quality monitoring, would further enhance the system's utility.Their methodology demonstrated substantial promise in improving prediction accuracy, especially in detecting subtle variations in milk quality parameters.

[17] Title: Efficient Milk Quality Detection System with IoT

Authors: **H. Madan Gouda**, **Nayana P.**, **Pardeep J.**, **P. Kumar**, **Y. Sudhanva**

Published year: 2018

This paper designed an integrating microcontrollers for sensor data handling and automated payment systems to streamline the supply chain. While this system reduced human intervention and improved operational efficiency, it faced challenges such as frequent sensor breakdowns, high costs, and maintenance issues. Future research in this area could explore AI-driven diagnostic tools, robust sensor designs, and predictive maintenance strategies to improve system reliability and efficiency.

CHAPTER 3:
SYSTEM ANALYSIS

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Milk quality is traditionally tested using manual methods such as chemical testing and sensory evaluation. These methods are time-consuming and prone to inconsistencies due to human error. Automated systems exist but are expensive and require specialized equipment, limiting their accessibility. Milk level monitoring in containers is rarely automated, leading to inefficiencies in stock management. In the current system, Milk quality is traditionally tested using manual methods such as chemical testing and sensory evaluation.Portable milk quality testers provide on-the-go solutions for dairy farmers, enabling quick measurements of pH, fat content, and specific gravity before selling their products. Furthermore, the advent of smartphone applications has led to innovative solutions that connect with external sensors for immediate testing, allowing users to monitor quality parameters easily. Despite these advancements, many existing systems require manual intervention and may not deliver real-time data, posing challenges for small-scale dairy producers.

3.2 PROPOSED SYSTEM

The proposed system integrates IoT to automate milk quality analysis and monitoring Sensors (pH, gas, and color) measure milk quality parameters, while an ultrasonic sensor tracks milk levels storage containers.An ESP32 microcontroller gathers data from sensors and updates it to the cloud for real-time monitoring. The proposed solution is cost-effective, scalable, and suitable for both small-scale and large-scale dairy operations. The proposed system uses pH, temperature, and color sensors and ultrasonic sensors integrated with Node MCU to automatically classify milk quality.The proposed system for milk quality analysis is an advanced, automated solution that integrates several cutting-edge technologies, including robotics, machine learning, visual computing, and biosensors, to ensure the safety and quality of milk. The proposed system uses pH, temperature, and color sensors and ultrasonic sensors integrated with Node MCU to automatically classify milk quality.

3.3 PROBLEM STATEMENT

In small scale milk quality analysis is not developed. In large scale, already the setup is existed but it is costly and the components not available for every milkmen and Dairymen. To overcome these problems we can use our model which is the solution for both problem, we can implement it for small scale with the minimal cost for analyze milk quality.
The challenge is to develop an automated system that uses IoT sensors and machine learning to continuously monitor and analyze milk quality in real-time, ensuring early detection of spoilage, contamination, and adulteration. And also maintain a constant Temperature to preserve milk .

3.4 OBJECTIVES

To identify the key parameters for assessing milk quality, we will develop an automation system using the ESP32 that performs quality checks on milk. This system will analyze these parameters to ensure accurate evaluation of milk quality.

3.5 METHODOLOGY

Collect sensor data using an ESP32 microcontroller connected to pH, gas, and ultrasonic sensors. Evaluate the models for accuracy, precision, and recall, selecting the best-performing algorithm. Conduct rigorous testing to ensure system reliability and accuracy. Deploy the system in a real-world environment for validation and feedback. Data collection and processing are critical, where sensor data is gathered during milk testing and subsequently analyzed using algorithms to evaluate milk quality based on the selected parameters. Ensuring accuracy through sensor calibration and employing mathematical models to correlate readings with milk quality standards is essential. The project's reliability is validated by comparing its results with established laboratory methods, followed by user testing to gather feedback from potential users like dairy farmers and quality control labs, which informs any necessary adjustments to the design.

3.6 SUMMARY

The proposed system offers a comprehensive solution for milk quality analysis and monitoring.It leverages IoT to automate testing, ensuring accuracy, efficiency, and reliability.By integrating cloud storage , it enhances transparency and responsiveness. The system is scalable, cost-effective, and suitable for diverse dairy operations. It reduces the risks associated with spoiled milk, protecting consumers and businesses. Historical data analysis enables continuous improvement in quality assurance practices. The project aligns with the global push for digital transformation in agriculture and food safety.

3.7 MOTIVATION

Milk quality degradation poses significant health risks, including foodborne illnesses and economic losses.Traditional quality testing methods are labor-intensive, time-consuming, and prone to human error. The rising demand for fresh, high-quality milk and the need for transparency in food safety standards inspired this project. Farmers and distributors often lack affordable and efficient tools to ensure milk quality in real-time. IoT technology provides an opportunity to develop low-cost, efficient solutions for continuous monitoring of milk quality. The ability to remotely monitor milk quality and levels can reduce losses due to spoilage or overproduction. IoT with the global push toward digital transformation in agriculture and food safety. By addressing these challenges, the project aims to improve milk quality assurance while reducing costs and inefficiencies in the dairy supply chain.

The motivation for the milk quality analyzer project stems from the critical importance of ensuring high-quality milk for both consumer health and the economic well-being of the dairy industry. With rising consumer awareness about food safety, there is an urgent need for efficient and reliable quality testing, as traditional methods often fall short, being time-consuming and prone to errors. This project aims to leverage technological advancements, such as spectroscopy and sensor technology, to develop an innovative analyzer that not only provides rapid and accurate results but also empowers farmers to monitor and improve their product quality.

CHAPTER 4: HARDWARE AND SOFTWARE REQUIREMENTS

CHAPTER 4

HARDWARE AND SOFTWARE REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

Functional requirements are a set of statements that define the functions that the system should provide,how the system should react to particular input and how the system should behave in particular situations. In some cases, the functional requirements may also specify what the system should not do. A function is described as a set of inputs, the behaviour, and outputs.

The functional requirements of the proposed system are as follows:

- System shall be able to test different samples of milk
- System shall be able to assess the various milk quality parameters through respective sensors.
- System shall be able to detect the changes in the parameters over a range of time and over a range of different samples
- System shall be able to update the test details on the web server
- System shall be able to display the test report to user.
- System shall be able to be monitored through software and hardware
- System shall be able to detect and alert user for changes in the hardware and software
- System shall be able to plot and display graph between change in different parameters of milk over a range of time

4.2 NON – FUNCTIONAL REQUIREMENTS

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours such as availability, scalability, security, backup and disaster recovery. Non-Functional Requirements are the constraints on the services or function offered by the system.

The non-functional requirements of the proposed system are as follows:

Lower response time

It will take less time to respond when the sample milk is given to test

Accurate milk quality detection

It will give accurate Parameter values i.e.,pH value, turbidity value, humidity value and temperature values

Power efficient

Breadboards have many small sockets on them, and some groups of sockets are electrically connected to each other.It will take less power

Simple and flexible user interface design

Data fetched from cloud server should be in the form of report so that it will be easy for the user to understand.

4.3 SOFTWARE REQUIREMENTS

1. ESP 32 IDE : version **2.3.2**
2. Python: version **3.11.8**
3. Web Application
4. Html
5. CSS
6. JavaScript
7. Milk Quality Dataset
8. Php and vs code
9. xampp

4.4 HARDWARE REQUIREMENTS

1. ESP32
2. Pcb board, power supply
3. Sensors-ph, temparature, color, ultrasonic
4. Connection wires

4.5 LANGUAGES AND TOOLS USED

4.5.1 Python:

Python is a flexible and popular programming language noted for its ease of use, readability, and a huge ecosystem of libraries and frameworks. It's ideal for image recognition and Arduino control because of its simplicity, strong libraries, and interoperability with a variety of hardware platforms.

It is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development.

It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options. Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability.

Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers.

Additionally, Python supports the use of modules and a package, which means that programs can be designed in a modular style and code can be reused across a variety of projects.

Python is ideal for both beginners and experienced programmers. Its key features include an easy-to-learn syntax, interpreted nature for straightforward debugging, and versatility that supports various programming paradigms such as procedural, object-oriented, and functional programming.

Python boasts an extensive standard library and a rich ecosystem of third-party packages available via the Python Package Index (PyPI), making it suitable for a wide array of applications, including web development (using frameworks like Django and Flask), data science (with libraries like Pandas and NumPy), machine learning (utilizing TensorFlow and scikit-learn), automation, and even game development (via libraries like Pygame).

The language benefits from a large and active community contributing to a wealth of resources, forums, and conferences like PyCon. Python's growing significance in artificial intelligence and data science, along with its continuous evolution through regular updates, discipline.

4.5.2 Arduino IDE:

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages python. Here, IDE stands for Integrated Development Environment.

The Arduino integrated development environment (IDE) (figure 4.4.1) is a cross- platform application for Windows, macOS, Linux that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards. The Arduino IDE supports the languages C and C++ using special rules of code structuring.

It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

The Arduino IDE is a popular microcontroller board based on the ATmega328P chip, widely used for electronics projects and prototyping.

It features 14 digital input/output pins, 6 analog inputs, a USB connection for programming, a power jack, and an ICSP header for programming the chip directly.

The IDE is designed to be easy to use for both beginners and experienced developers, thanks to its supportive programming environment, the Arduino IDE, which allows users to write, compile, and upload sketches (programs) to the board seamlessly.

Its versatility and ease of use have made it a staple in the maker and DIY communities, encouraging creativity and innovation in electronics.

The Arduino IDE versatility has led to its use in diverse applications, including robotics, automation, environmental monitoring, and interactive art installations. Its large user base and robust community support provide a rich environment for sharing knowledge.

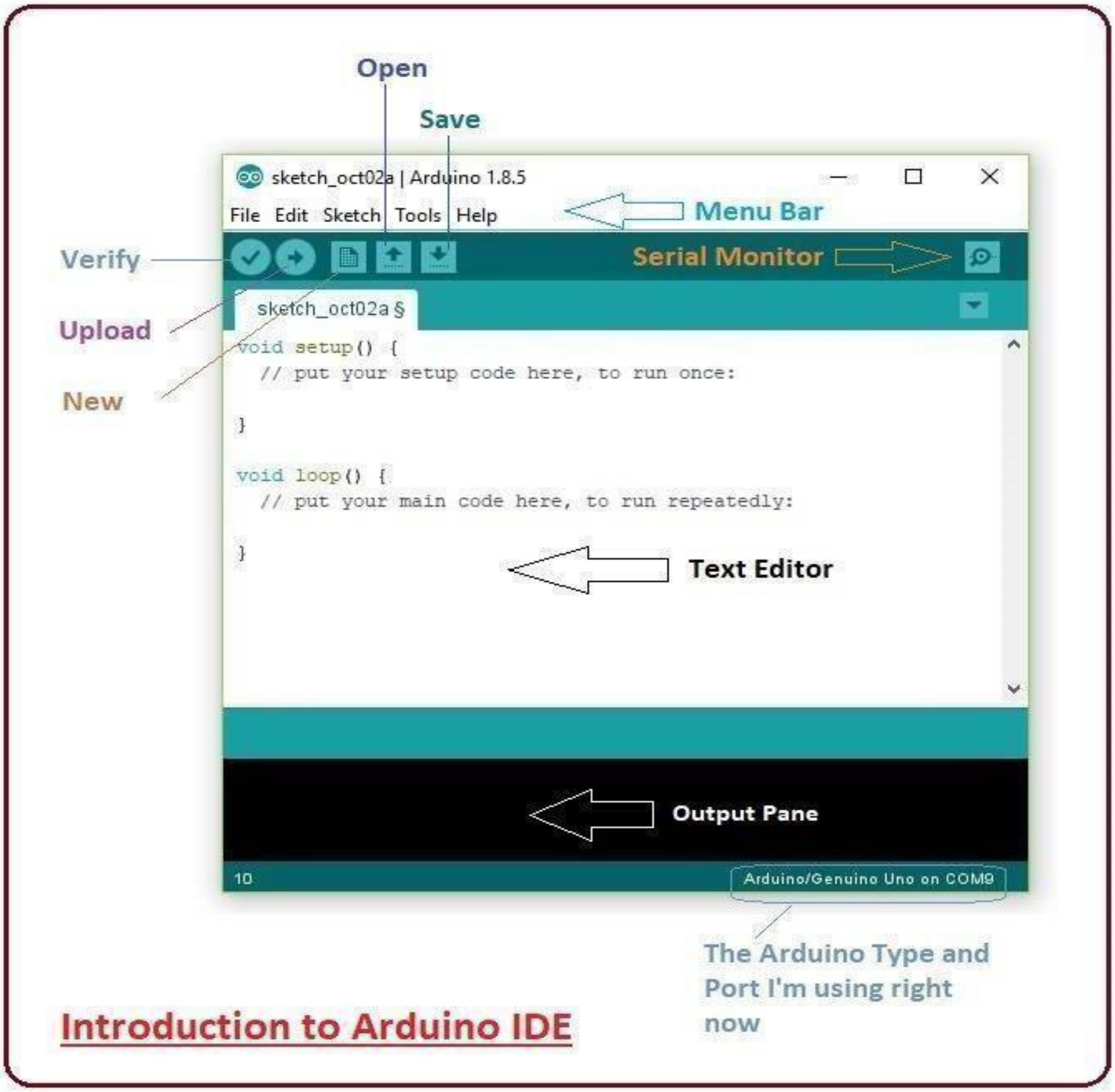


Figure 4.5.1 Arduino IDE

The IDE environment is mainly distributed into three sections:

- Menu Bar
- Text Editor
- Output Pane

4.5.3 Numpy

Numpy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

At the core of the NumPy package, is the *ND array* object. This encapsulates *n*-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences

NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an *ndarray* will create a new array and delete the original.

The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.

NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.

A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-insequence types is insufficient - one also needs to know how to use NumPy arrays.

Uses NumPy

Many of its methods are mirrored by functions in the outer-most NumPy namespace, allowing the programmer to code in whichever paradigm they prefer. This flexibility has allowed the NumPy array dialect and NumPy *ND array* class to become the *de- facto* language of multi-dimensional data interchange used in Python.

NumPy fully supports an object-oriented approach, starting, once again, with *ND array*.

For example, *ND array* is a class, possessing numerous methods and attributes.

Vectorization describes the absence of any explicit looping, indexing, etc., in the code - these things are taking place, of course, just "behind the scenes" in optimized, pre-compiled C code. Vectorized code has many advantages, among which are:

vectorized code is more concise and easier to read fewer lines of code generally means fewer bugs the code more closely resembles standard mathematical notation (making it easier, typically, to correctly code mathematical constructs) vectorization results in more "Pythonic" code. Without vectorization, our code would be littered with inefficient and difficult to read for loops.

4.5.4 Xampp

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It is designed to simplify the process of setting up a local web server for development and testing purposes. XAMPP includes Apache, MySQL, PHP, and Perl, allowing developers to easily run and test dynamic websites or web applications on their local machines without the need for an internet connection or external server.

Its user-friendly interface makes it easy to install and configure, and it supports a variety of operating systems such as Windows, Linux, and macOS.

XAMPP is widely used by web developers for prototyping, testing, and debugging, offering a convenient environment for coding without affecting live production servers**.**

4.5.5 VS CODE TOOLS:

Visual Studio Code (VS Code) is a powerful, lightweight, and highly customizable code editor developed by Microsoft. It supports a wide range of programming languages such as JavaScript, Python, C++, Java, and many more. One of its key features is an extensive library of extensions that enhance its functionality, allowing developers to integrate additional tools and languages easily.These extensions enable features like code linting, debugging, version control integration (Git), and auto-completion, among others.

VS Code also offers a built-in terminal, intelligent code suggestions, and a highly interactive debugging environment. Its robust source control integration allows seamless interaction with Git repositories directly within the editor. With an active community and continuous updates,VS Code has become one of the most popular code editors for both professional developers and hobbyists. Its flexibility, speed, and vast ecosystem of tools make it an ideal choice for software development across various platforms.

CHAPTER 5:
SYSTEM DESIGN

CHAPTER 5

SYSTEM DESIGN

The system design for the proposed Automated Milk Quality Analyzer is built around an integrated framework that combines various technologies to ensure accurate, efficient, and real-time milk quality testing. At its core, the system consists of a **robotic arm** that automates the collection and handling of milk samples, ensuring consistency and reducing human error during the sample preparation process. Once the sample is collected, **infrared spectroscopy** is employed to analyze the chemical composition of the milk, measuring key parameters like fat, protein, and lactose content. This non-destructive technique ensures that milk samples remain intact and can be tested without affecting their quality. Additionally, **biosensors** are integrated into the system to detect harmful substances such as antibiotics, pesticides, and heavy metals, ensuring that milk meets safety standards. The system also includes a **computer vision module** that uses cameras and optical sensors to evaluate the visual quality of the milk, detecting any abnormalities such as color variations, clarity, or microbial contamination

5.1 SYSTEM ARCHITECTURE AND DESIGN

System Design

A system architecture diagram would be used to show the relationship between different components. Usually they are created for systems which include hardware and software and these are represented in the diagram to show the interaction between them. The software aspect includes firmware programmed into the ESP32 to manage data acquisition and processing, along with algorithms for data calibration and conversion to ensure precise measurements.

Data storage can be local on an SD card or in the microcontroller's memory, with optional integration to cloud services for remote access and advanced analytics. The ESP32's built-in Wi-Fi or Bluetooth capabilities enable seamless communication with mobile devices, enhancing user interaction. Regular sensor calibration is essential to maintain accuracy, supported by validation against standard laboratory methods to ensure reliability. Implementing alerts for out-of-range parameters provides immediate feedback to users on compromised milk quality.

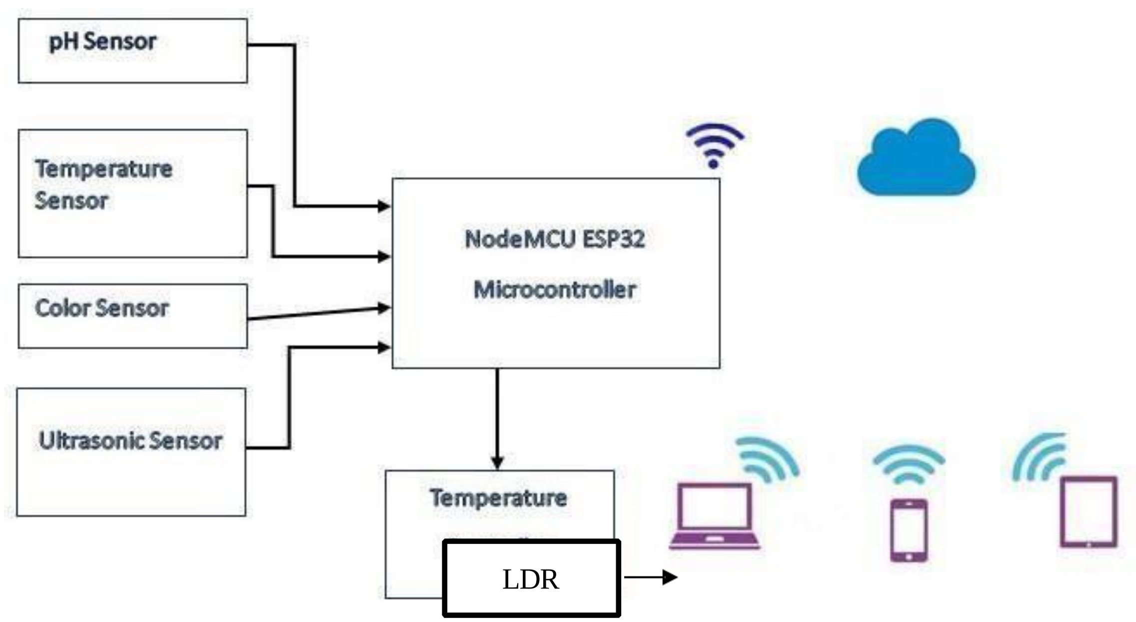


Figure 5.1.1: System Design

This diagram illustrates the block diagram of a milk quality analyzer system. It consists of several key components: a power supply that powers the system, multiple sensors including pH, gas, and UV sensors that measure specific quality parameters of milk, and an Analog- to-Digital Converter (ADC) that processes the sensor data. The microcontroller receives the converted data, processes it, and interfaces with a display to show the results in real time. Additionally, the system is connected to Wi-Fi, allowing for remote monitoring or control through a network. This design emphasizes the integration of multiple sensors and connectivity to provide comprehensive milk quality analysis.

5.2 HIGH LEVEL ARCHITECTURE

Detailed Design : System Architecture

- Project's basic principle of working is the sensing of data from the sensor .
- Convert the analog (voltage) data into digital form.
- Process the digital data and display .
- Compare the threshold value and send to the IOT page.

DATA FLOW DIAGRAMS: A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

Data Flow Diagram Level Zero:

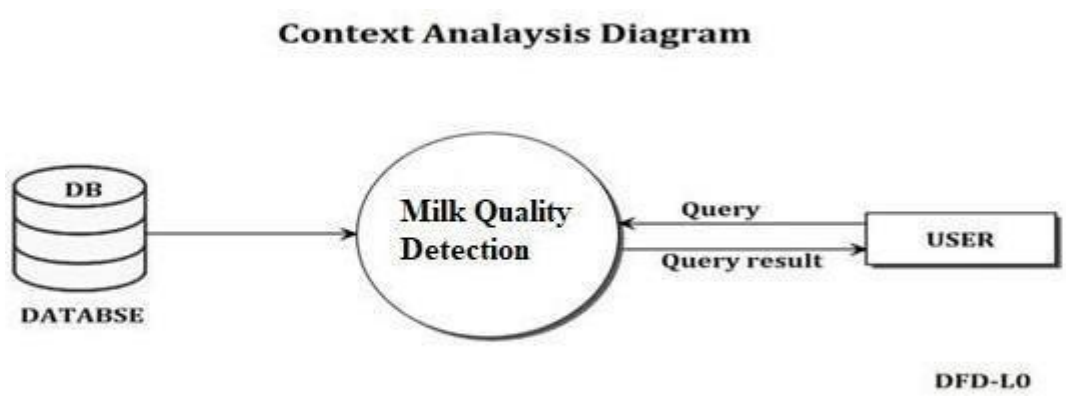


Figure 5.2.1: Context Analaysis Design

The diagram represents a context analysis for a milk quality detection system. It showcases the interaction between a user and the system via queries submitted to the "Milk Quality Detection" process. The system accesses a database (DB) to retrieve relevant data for processing the queries. The user's input leads to specific queries about milk quality, with the system returning the corresponding results based on the database information. This diagram effectively highlights the flow of information between the user, the quality detection process, and the database.Users can submit specific questions, prompting the system to process these queries and return concise results, facilitating informed decision-making about milk quality. Overall, the diagram clarifies the flow of information within the system, emphasizing the relationship between the user, the analytical process, and the supporting database, all aimed at delivering accurate assessments of milk quality.

Data Flow Diagram Level 1:

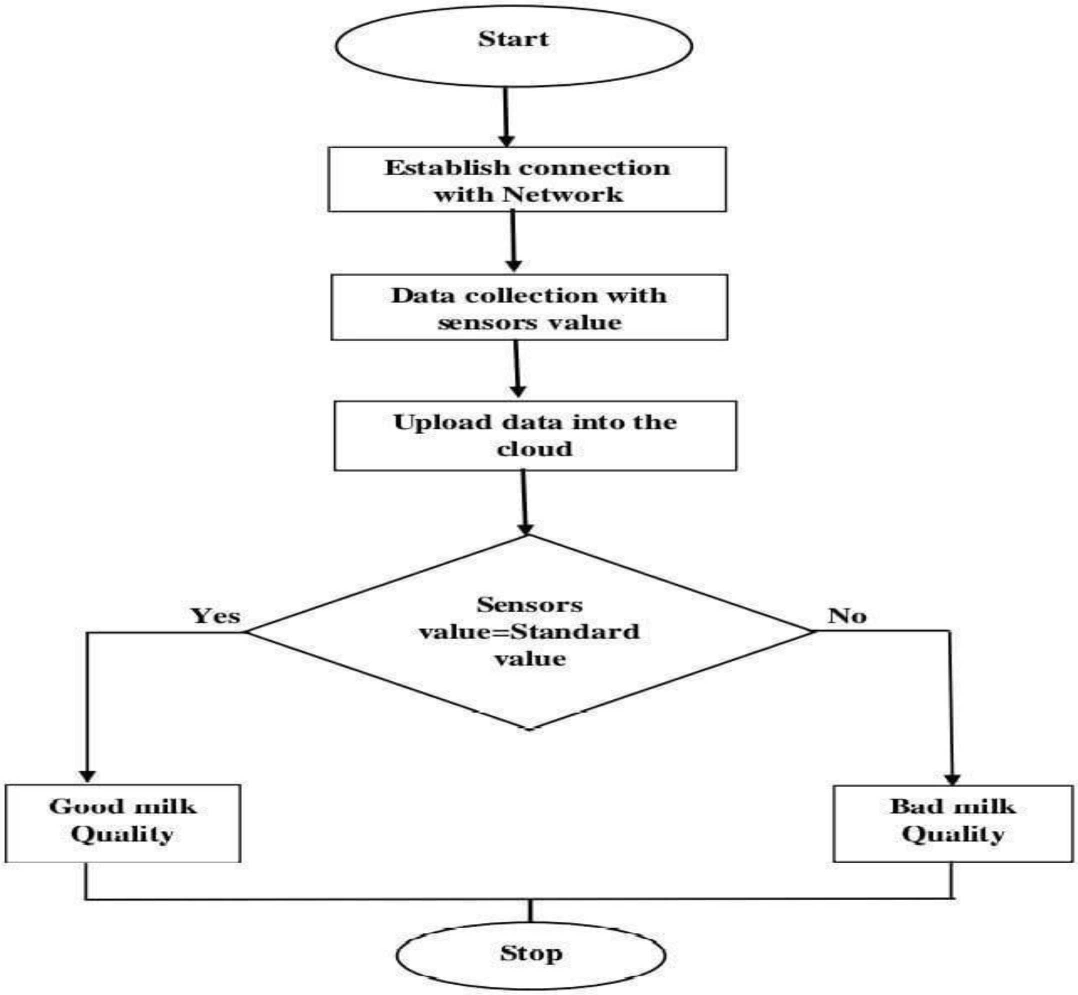


Figure 5.2.2: Data Flow Design

The flowchart outlines a comprehensive process for assessing milk quality through the integration of sensor technology and cloud computing. It begins with the **"Start"** point, signifying the initiation of the evaluation procedure. The first step is to **establish a connection with a network**, which is crucial as it allows data sharing and communication between devices. This connection often involves accessing an internet network to facilitate the transfer of information to cloud storage. After the connection is established, the next step involves **data collection with sensors**. Various sensors are deployed to monitor specific quality parameters of the milk, such as pH levels, temperature, presence of contaminants, and other chemical compositions. These sensors are designed to gather real-time data that reflects the purity and safety of the milk.

Once the data is collected, the system **uploads this information into the cloud**. Cloud storage is leveraged for its scalability and accessibility, allowing for efficient data management and analysis. The cloud system processes the collected data, enabling further insights and analytics regarding the milk's quality. Following data upload, the process reaches a **decision point**, represented by a diamond shape in the flowchart. Here, the system evaluates whether the sensor values correspond to **standard values** that have been predetermined based on industry norms or safety guidelines. This assessment is critical for ensuring food safety and quality. If the sensor readings align with the standard values, the output indicates that the milk quality is classified as **"Good,"** suggesting that the milk is suitable for consumption. This positive assessment can foster consumer trust and product integrity. Conversely, if the results do not meet the required standards, the output classifies the milk quality as **"Bad,"** indicating it is unsafe or unfit for consumption. This automatic evaluation allows for quick decision-making, preventing possible health risks. Finally, the process concludes at the **"Stop"** point, signaling the end of the evaluation. This systematic approach not only enhances efficiency but also promotes food safety by integrating modern technology into traditional dairy practices. Overall, this flowchart exemplifies how technology can be harnessed to ensure high standards in food quality management.

Class Diagram:

In software engineering, a class diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

Class diagrams are a fundamental component of Unified Modeling Language (UML) used to represent the static structure of a system by illustrating its classes, attributes, methods, and their relationships. Each class is depicted as a rectangle divided into three sections: the top section contains the class name, the middle lists its attributes with visibility indicators (public, private, protected), and the bottom details its methods. Relationships between classes are crucial, encompassing associations (basic interactions), aggregations (whole- part relationships where parts can exist independently), compositions (stronger whole-part relationships where parts are dependent on the whole), and inheritance (where one class derives from another). Each relationship is visually represented using specific notation, such as solid lines for associations and diamonds for aggregations and compositions. Class diagrams help in visualizing system architecture, guiding code generation, serving as documentation, and facilitating communication among stakeholders. They are widely utilized in object-oriented design, system modeling, and database design, making them essential tools for structuring complex software systems effectively.

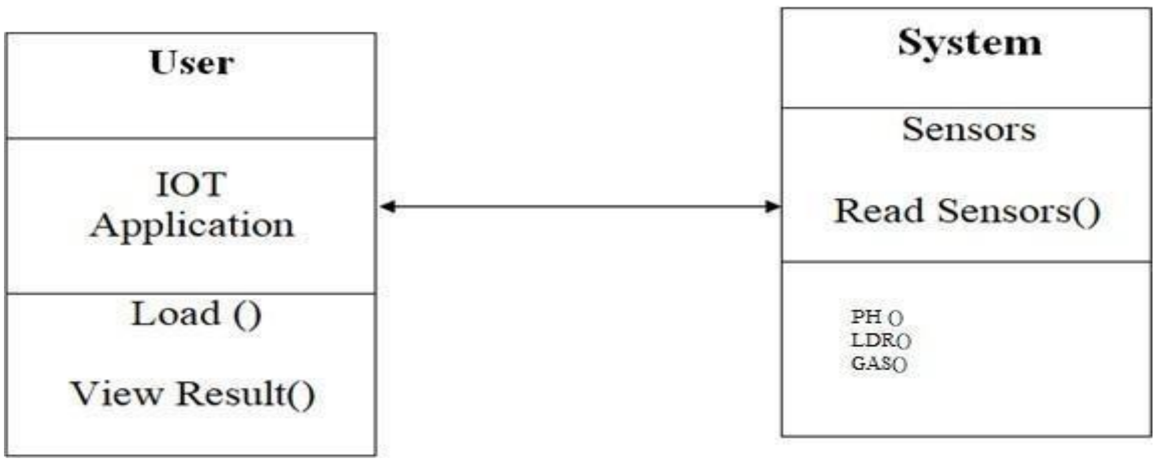


Figure 5.2.3 : Class Diagram

The diagram depicts the interaction between a user and an IoT-based milk quality analysis system. On the left, the **User** interacts with an **IoT Application**, where they can execute specific functions such as **Load()** to initialize the system and **View Result()** to check the analysis results. On the right, the **System** consists of various **Sensors** that collect data related to milk quality, including pH, LDR (Light Dependent Resistor), and gas sensors. The system processes this data through the **Read Sensors()** function, which retrieves real-time measurements from the sensors.

This interaction emphasizes the user's ability to engage with the application to load data and view insights based on the sensor readings.

Overall, this diagram captures the essence of a user-centric IoT system for milk quality analysis, showcasing the interplay between user commands and sensor-driven data processing, which ultimately aids in ensuring the safety and quality of milk products.

Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support choice, iteration and concurrency.

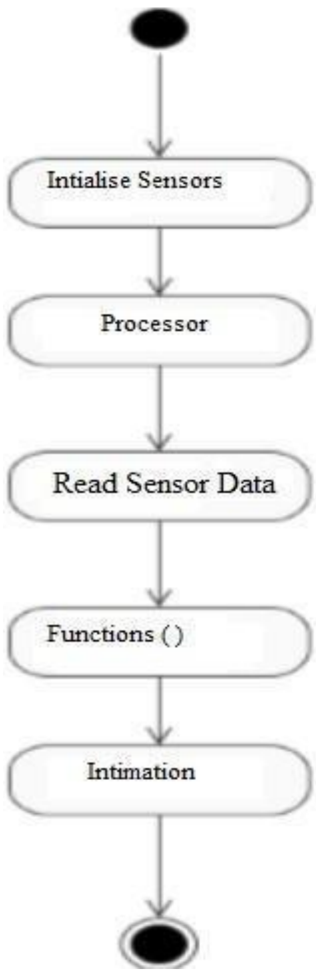


Figure 5.2.4: Activity Diagram

The diagram outlines the workflow of a milk quality monitoring system. It starts with **Initialise Sensors**, where the system sets up various sensors for data collection. This is followed by the **Processor**, which handles the data, enabling the system to **Read Sensor Data** and perform necessary calculations or analyses.

Finally, the system executes various **Functions** based on the data gathered, leading to an **Intimation**, which communicates the results or alerts related to milk quality to the user.

Sequence Diagram:

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

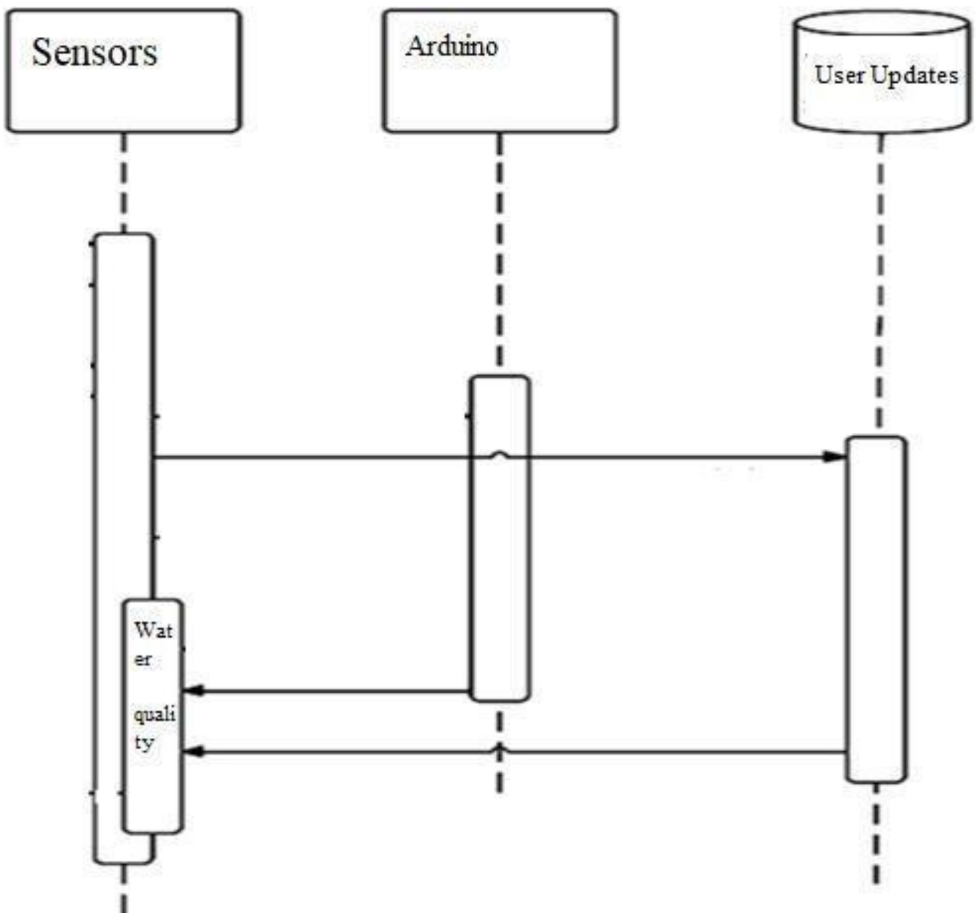


Figure 5.2.5 : Sequence Diagram00000

The diagram depicts a water quality monitoring system using sensors and an Arduino. **Sensors** collect data on water quality parameters, which is then transmitted to the **Arduino** for processing. The Arduino analyzes this data and updates a database of **User Updates** with the results. This system allows users to receive real-time information about water quality based on sensor readings.This system enables users to receive real-time information and alerts regarding water quality, facilitating prompt actions in response to any identified issues, thereby ensuring safe and reliable water resources.

5.3 SYSTEM COMPONENTS:

ESP32 bit Micro-Controller

ESP32 is a microcontroller board. ESP32 dev kit has 36 pins and 18 on each side of the board as shown in the picture below. It has 34 GPIO pins and each pin has multiple functionalities which can be configured using specific registers. There are many types of GPIOs available like digital input, digital output, analog input, and analog output, capacitive touch, UART communication and USB cable. A ESP32, consist of two functions:

setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed.

ESP32 is 32-bit microcontroller,provides low power & connectivity ,wifi & bluetooth for low energy consumption and Low power system on chip(SOC),clock speed of ESP32 can be controlled independently.

ESP32 has dual core processor,one is application cpu which handles code and another is protocol cpu which handles wifi and indicative peripheral for A2D controller and D2A controller .

Below mentioned are the traits of suggested imitation:

(i) **Collection of health data** : In order to determine the health status of a patient, we need to consider various health parameters and instead of using different devices for the collection of data, this single IOT device will collect real time and accurate data.

(ii).Analyzing the data : The data collected will be stored in the cloud.The variations in the data for specific duration can be plotted as graphs.This graph helps the doctor to analyze the patient's responsiveness to the medication.

(iii).Prediction of recovery rate : After Analyzing the data, we can predict the recovery rate of the Patient accurately in real-time considering the dataset.

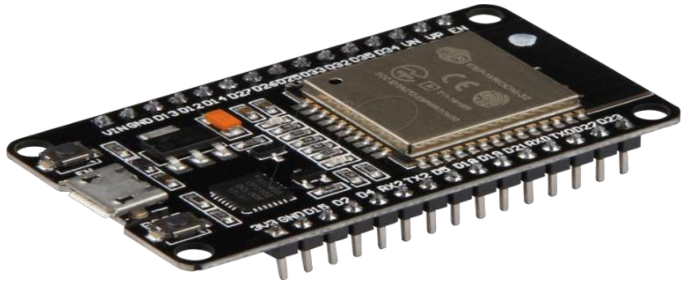


Fig 5.3.1 ESP32 Micro -Controller

Jumper Wires

Fig 5.3.2 Jumper wires

A **jump wire** (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable) is an electrical wire or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Milk PH sensor:

pH METER

Figure 5.3.3 ph sensor

Principles of operation of a pH meter

A pH meter is essentially a voltmeter with a high input impedance which measures the voltage of an electrode sensitive to the hydrogen ion concentration, relative to another electrode which exhibits a constant voltage. The key feature of the pH-sensitive electrode is a thin glass membrane whose outside surface contacts the solution to be tested. The inside surface of the glass membrane is exposed to a constant concentration of

hydrogen ions (0.1 M HCl).

Inside the glass electrode assembly, a silver wire, coated with silver chloride and immersed in the HCl solution, is called an Ag/AgCl electrode. This electrode carries current through the half-cell reaction **.** A reference electrode is needed to complete the electrical circuit. A common choice is to use another Ag/AgCl electrode as the reference. The Ag/AgCl electrode is immersed in an 0.1 M KCl solution which makes contact with the test solution through a porous fiber which allows a small flow of ions back and forth to conduct the current. The potential created at this junction between the KCl solution and the test solution is nearly zero and nearly unaffected by anything in the solution, including hydrogen ions.
Using the pH Meter: Allow the meter a few minutes to stabilize after you plug it in. When you are not using the meter, keep the electrode immersed in pH 7.0 buffer to a depth of about one inch. The meter must be calibrated by using standards of known pH before an unknown is measured. Since the unknowns are acidic, the pH 4.00 and pH 7.00 standards should be used. An accurate pH reading depends on standardization, the degree of static charge, and the temperature of the solution.

Temperature DS18B20:

Figure 5.3.4 temperature sensor

The DS18B20 is a digital temperature sensor manufactured by Maxim Integrated. It is widely used due to its accuracy, simplicity, and versatility.
Here's a description of the DS18B20 sensor:
1. Digital Temperature Sensor: The DS18B20 is a digital temperature sensor, which means it provides temperature readings in a digital format, making it easy to interface with microcontrollers and digital circuits.
2. High Accuracy: The sensor offers high accuracy with a typical measurement error of ±0.5°C over the range of -10°C to +85°C.

3. Wide Temperature Range: It can measure temperatures ranging from -55°C to +125°C, making it suitable for a wide range of applications.

4. Single Wire Interface: One of the key features of the DS18B20 is its single-wire interface, also known as the OneWire interface. This allows multiple DS18B20 sensors to be connected to a single microcontroller pin, simplifying wiring and reducing the number of required input/output pins.

5. Parasitic Power Mode: The DS18B20 can operate in parasitic power mode, where it draws power for its operation from the data line, eliminating the need for an external power supply connection. This feature is
particularly useful in applications where power consumption needs to be minimized.

6. Programmable Resolution: The sensor supports multiple user-selectable resolution modes, ranging from 9 to 12 bits. Higher resolution provides more precise temperature readings but requires longer conversion times.

7. Milk proof Versions Available: There are milk proof versions of the DS18B20 sensor, encapsulated in stainless steel tubes, which are suitable for applications where the sensor may be exposed to moisture or liquids.

8. Digital Output: The DS18B20 communicates temperature data using a digital serial interface, making it immune to noise and interference compared to analog sensors.

Overall, the DS18B20 is a versatile and reliable digital temperature sensor that finds widespread use in a wide range of temperature monitoring and control applications.

Milk Level Sensor:

A milk level sensor is a device used to measure the level of milk in a container or a body of milk. There are various types of milk level sensors, each employing different principles of operation.

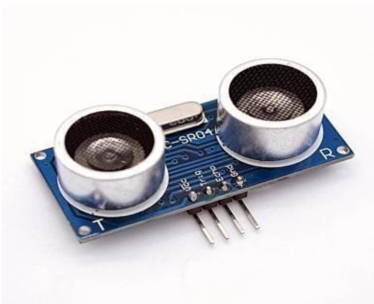


Figure 5.3.5 Ultrasonic sensor

Here's an overview of some common types:

1. Float Switches: Float switches consist of a buoyant float connected to a mechanical switch. As the milk level rises or falls, the float moves with it, triggering the switch to open or close. Float switches are simple, reliable, and cost-effective, making them suitable for many applications, including sump pumps, milk tanks, and industrial processes.

2. Ultrasonic Sensors: Ultrasonic milk level sensors use sound waves to measure the distance from the sensor to the surface of the milk. The sensor emits ultrasonic pulses, which are reflected off the milk surface and detected by the sensor. By measuring the time it takes for the pulses to travel to the milk surface and back, the sensor can determine the milk level. Ultrasonic sensors are non-contact, making them suitable for applications where contact with the milk is not desirable, such as in waste milk treatment plants and reservoirs.

Gas Sensor:

This is the MQ-3 Gas Sensor Module, which is simple to use. It can sense the presence of Alcohol in the air. The module uses our MQ-3 sensor. It simplifies the interface to the odd pin spacing of the sensor and provides an interface through standard 4 x 0.1″ header pins. MQ-3 gas sensor provides an analog output corresponding to the concentration of the gas in the air and an easy-to-use digital output. The sensitive material used for this sensor is SnO_2, whose conductivity is lower in clean air. Its conductivity increases as the concentration of alcohol gasses increases. It has a high sensitivity to alcohol and has good resistance to disturbances due to smoke, vapour and gasoline. MQ-3 gas sensor module provides both digital and analog outputs. It can be easily interfaced with Microcontrollers like Arduino Boards , Raspberry Pi etc.

This MQ-3 alcohol sensor is suitable for detecting alcohol concentration on your breath, just like your common breathalyzer. It has a high sensitivity and fast response time. The sensor provides an analog resistive output based on alcohol concentration. The drive circuit is very simple, all it needs is one resistor. A simple interface could be a 0-3.3V ADC

CHAPTER 6: IMPLEMENTATION

CHAPTER 6

IMPLEMENTATION

6.1 SYSTEM IMPLEMENTATION

Implementation is the process of converting a new or a revised system design into an operational one. The objective is to put the new or revised system that has been tested into operation while holding costs, risks, and personal irritation to the minimum. A critical aspect of the implementation process is to ensure that there will be no disrupting the functioning of the organization. The best method for gaining control while implanting any new system would be to use well planned test for testing all new programs. Before production files are used to test live data, text files must be created on the old system, copied over to the new system, and used for the initial test of each program.

Another factor to be considered in the implementation phase is the acquisition of the hardware and software. Once the software is developed for the system and testing is carried out, it is then the process of making the newly designed system fully operational and consistent in performance.

Implementation is the most crucial stage in achieving a successful system and giving the user's confidence that the new system is workable and effective. Implementation of a modified application to replace an existing one. This type of conversation is relatively easy to handle, provided there are no major changes in the system.

The development of accompanying software entails creating data acquisition systems, implementing algorithms for data processing and calibration, and designing a user-friendly interface for easy operation. Testing and calibration are crucial steps to ensure accuracy, where known samples are used for verification. After deployment, regular maintenance and user support are necessary, along with adherence to food safety regulations and standards. Finally, establishing a feedback loop is beneficial for continuous improvement based on user experiences. This comprehensive approach ensures the system effectively meets the quality assessment needs of the dairy industry.

6.2 IMPLEMENTATION METHODOLOGY OF THE PROJECT

The project is implemented in modular approach. Each module is coded as per the requirements and is tested and this process is repeated until all the modules have been thoroughly implemented.

IoT-Based System:

ESP32 Controller: Serves as the central processing unit, managing data from all connected sensors.

Sensors: **pH Sensor**: Measures the acidity or alkalinity of the milk.

Gas Sensor: Detects the presence of specific gases that indicate spoilage.

Ultrasonic Sensor: Monitors the milk level in the container to prevent overflows or shortages.

User Interaction:The user manages and monitors the system directly through the ESP32 and cloud interface.

The diagram outlines an IoT-based milk quality monitoring system. At the center is the **ESP32 Controller**, which connects various sensors (pH, gas, LDR, ultrasonic) to monitor milk quality parameters and display data . The collected data is transmitted via a **Wi-Fi Module** to **Cloud Services**, specifically **Cloud**, for storage and further analysis. Finally, the results trigger , providing real-time updates and insights on the milk quality status.

Pin Diagram

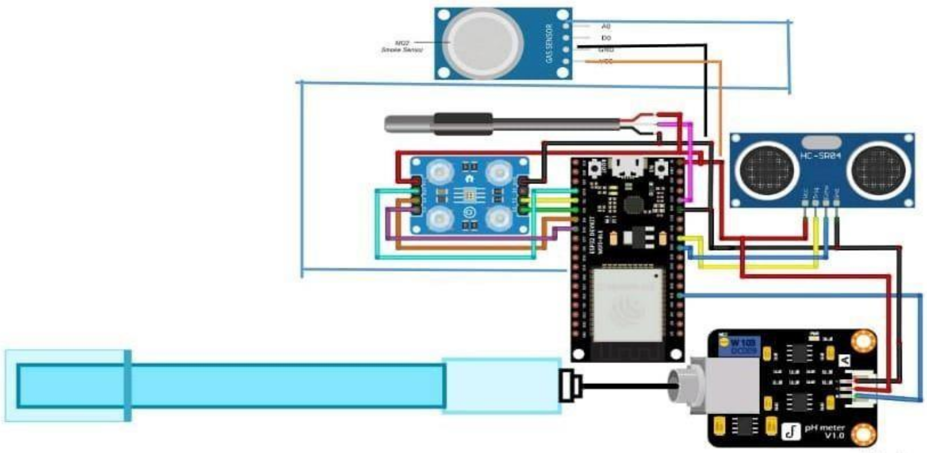


Figure 6.1: Pin diagram

The diagram illustrates a connection setup between a microcontroller, likely an ESP32, and several peripheral components including an ultrasonic sensor (HC-SR04), a pH sensor module, a motor driver, and a 4-channel relay module. The ESP32 provides power (VCC) and a common ground (GND) for all components.

The ultrasonic sensor has its **TRIG** pin connected to a digital GPIO pin on the ESP32 to trigger sound pulses, while the **ECHO** pin is linked to another GPIO for receiving reflected sound signals. The pH sensor outputs its readings through an analog pin on the ESP32, allowing for pH measurement. The motor driver is powered separately but controlled via GPIO pins on the ESP32, enabling motor operation.

Finally, the relay module connects to the ESP32's digital pins to switch devices on and off. Proper grounding and voltage matching are essential for reliable operation in this configuration, which facilitates various functionalities such as distance measurement and environmental sensing in embedded applications.

BLOCK DIAGRAM

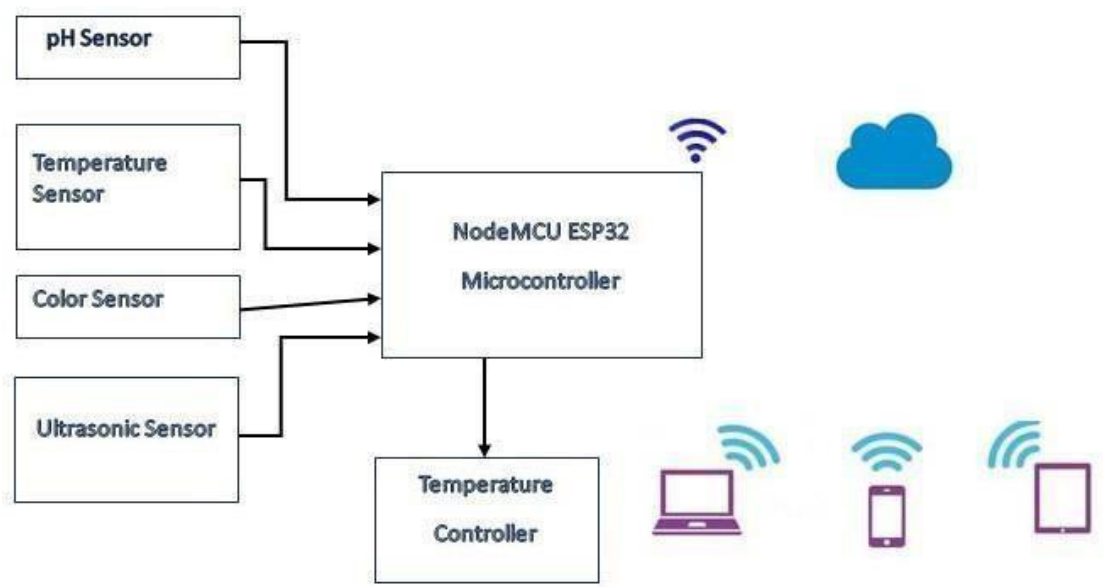


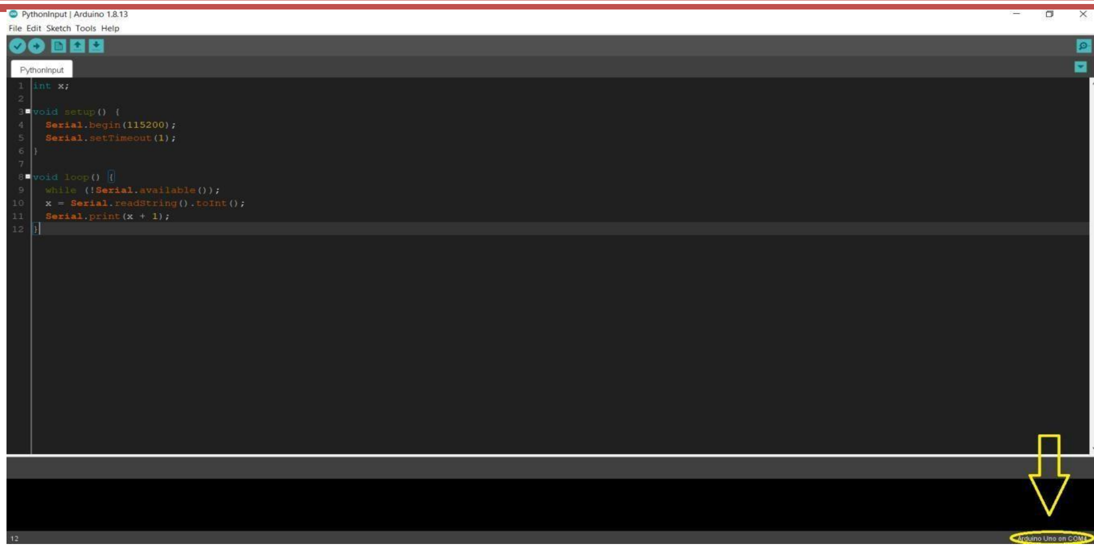
Figure 6.2 block diagram

In a Milk Quality Analyzer IoT-based Project, the system checks the quality of milk using sensors and shares the results online.

Sensors: These sensors measure important qualities of the milk:

- pH Sensor: Measures acidity.
- Temperature Sensor: Measures how hot or cold the milk is.
- Fat Sensor: Measures the fat content in the milk.
- Other Sensors: Measure things like milk quality

It consists of several key components: a power supply that powers the system, multiple sensors including pH, gas, and UV sensors that measure specific quality parameters of milk, and an Analog- to-Digital Converter (ADC) that processes the sensor data. The microcontroller receives the converted data, processes it, and interfaces with a display to show the results in real time. Additionally, the system is connected to Wi-Fi, allowing for remote monitoring or control through a network.



To initiate a connection with the Arduino from Python, we first have to figure out which COM Port the Arduino is on. We can simply see in which port our Arduino is on. Arduino code is structured around two main

functions: setup() and loop(). The setup() function is executed once at the start, where you initialize variables, set pin modes, and include any necessary libraries. The loop() function then runs continuously, allowing you to implement repetitive tasks.

Within the code, you can define variables to store values, create custom functions for reusable code segments, and utilize various libraries to simplify interactions with hardware. For instance, a simple program might read a temperature from a sensor and turn on an LED if the temperature exceeds a certain threshold. Overall,

```
int x;
void
setup() {
Serial.begi
n(115200);
Serial.setTi
meout(1);
}
void loop() {
while (!Serial.available());
x =
```

```
Serial.rea
dString()
.toInt();
Serial.pr
i nt(x +
1);
}
```

Arduino IDE (Integrated development Environment) is fully developed into functionality of full of libraries, as long as programming the Arduino UNO in Embedded C language is possible because Arduino IDE can compile both Arduino code as well as AVR standard code.When designing software for a smaller embedded system with the 8051, it is very common place to develop the entire product using assembly code. With many projects, this is a feasible approach since the amount of code that must be generated is typically less than 8 kilobytes and is relatively simple in nature. If a hardware engineer is tasked with designing both the hardware and the software, he or she will frequently be tempted to write the software in assembly language.The trouble with projects done with assembly code can is that they can be difficult to read and maintain, especially if they are not well commented. Additionally, the amount of code reusable from a typical assembly language project is usually very low. Use of a higher-level language like C can directly address these issues. A program written in C is easier to read than an assembly program.Since a C program possesses greater structure, it is easier to understand and maintain. Because of its modularity, a C program can better lend itself to reuse of code from project to project. The division of code into functions will force better structure of the software and lead to functions that can be taken from one project and used in another, thus reducing overall development time. A high order language such as C allows a developer to write code, which resembles a human's thought process more closely than does the equivalent assembly code. The developer can focus more time on designing the algorithms of the system rather than having to concentrate on their individual implementation. This will greatly reduce development time and lower debugging time since the code is more understandable

}

6.3 SOURCE CODE

6.3.1 Farmer

```php
<?php
// Enable error reporting for debugging
ini_set('display_errors', 1);
error_reporting(E_ALL);
// Database connection parameters
$servername = "localhost"; // Replace with your server details
$username = "root"; // Replace with your database username
$password = ""; // Replace with your database password
$dbname = "milk"; // Replace with your database name

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die(json_encode(['error' => 'Connection failed: ' . $conn->connect_error]));
}

// SQL query to fetch all data from the milk_quality_analyzer table
$sql = "SELECT id, temperature_value, ph_sensor_value, color_sensor_value,
gas_sensor_value, ultrasonic_sensor_value, quality_status, created_at
        FROM milk_quality_analysis";

// Execute the query
$result = $conn->query($sql);
```

```
// Check if records are found
if ($result->num_rows > 0) {
    $records = [];

    // Fetch all records
    while ($row = $result->fetch_assoc()) {
        $records[] = [
            'timestamp' => $row['created_at'],
            'temperature' => $row['temperature_value'],
            'ph_level' => $row['ph_sensor_value'],
            'color_temp' => $row['color_sensor_value'],
            'gas_level' => $row['gas_sensor_value'],
            'ultrasonic_level' => $row['ultrasonic_sensor_value'],
            'milk_quality' => $row['quality_status']
        ];
    }

    // Return the records as JSON
    echo json_encode(['records' => $records]);
} else {
    echo json_encode(['error' => 'No data found']);
}

// Close connection
$conn->close();
?>
```

6.3.2 Index

```
<?php
session_start();

if (isset($_SESSION['username'])) {
```

```
    header("Location: milk.php"); // Redirect to home page if already logged in
    exit();
}
?>
```

6.3.3 Login

```
<?php
// Database connection
$host = "localhost";
$dbname = "milk";
$username = "root";
$password = "";

session_start();

try {
    // Create a new PDO connection
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $user = $_POST['username'];
        $pass = $_POST['password'];

        // Query to check if the username and password match
        $stmt = $pdo->prepare("SELECT * FROM users WHERE username = :username
AND password = :password");
        $stmt->bindParam(':username', $user);
        $stmt->bindParam(':password', $pass);
        $stmt->execute();

        if ($stmt->rowCount() > 0) {
            $_SESSION['username'] = $user; // Store the username in session
```

```
            header("Location: home.html"); // Redirect to the main page after successful login
            exit();
        } else {
            // Redirect with error message if invalid login
            header("Location: login.html?error=Invalid username or password.");
            exit();
        }
    }
} catch (PDOException $e) {
    echo "Error: " . $e->getMessage();
    exit();
}
?>
```

6.3.4 Logout

```
<?php
session_start();
session_unset(); // Unset all session variables
session_destroy(); // Destroy the session
header("Location: login.html"); // Redirect to the login page
exit();
?>
``
```

6.3.5 Signup

```
<?php
// Database connection
$host = "localhost";
$dbname = "milk";
$username = "root";
$password = "";
try {
    // Create a new PDO connection
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
```

```
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $user = $_POST['username'];
        $pass = $_POST['password'];

        // Check if the username already exists
        $stmt = $pdo->prepare("SELECT * FROM users WHERE username = :username");
        $stmt->bindParam(':username', $user);

  $stmt->execute();
        if ($stmt->rowCount() > 0) {
            // If the username exists, redirect with an error
            header("Location: signup.html?error=Username already exists.");
            exit();
        } else {
            // Insert new user into the database
            $stmt = $pdo->prepare("INSERT INTO users (username, password) VALUES
(:username, :password)");
            $stmt->bindParam(':username', $user);
            $stmt->bindParam(':password', $pass);
            $stmt->execute();  // Redirect to login page after successful registration
            header("Location: login.html?message=Registration successful. Please log in.");
            exit();
        }
    }
} catch (PDOException $e) {
    // Handle errors
    echo "Error: " . $e->getMessage();
    exit();
}
?>
```

CHAPTER 7:
SOFTWARE TESTING

CHAPTER 7

SOFTWARE TESTING

Testing is an important phase in the development life cycle of the product, this was the phase where the error remaining from all the phases was detected. Hence testing performs a very critical role in quality assurance and ensuring the reliability of the software. The testing determines the program reliability of the software. During the testing, the program to be tested was executed with a set of test cases and the output of the program for the test cases was evaluated to determine whether the program is performing as expected. Errors were found and corrected by using the following testing steps and the correction was recorded for future reference. Thus, a series of testing was performed on the system before it was ready for implementation.

There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation not merely a matter of creating and following a routine procedure.

7.1 FUNCTIONALITY TESTING

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered. Functional testing is conducted to evaluate the compliance of a system or component with specified functional requirements. Functional testing usually describes what the system does.

- The keras integration is successfully established.
- UI elements properly rendered on the application.
- Real-Time results are visible on the application.
- After object identification, appropriate instructions are transferred to the hardware.
- Proper alert messages are displayed.

7.2 COMPATIBILITY TESTING

Compatibility testing is a part of non-functional testing conducted on application software to ensure the application's compatibility with the different computing environments.

- This application is compatible with all the devices enabled with python and TensorFlow.
- It is compatible with all desktops.

7.3 PERFORMANCE TESTING

In software quality assurance, performance testing is in general a testing practice performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability, and resource usage. Performance testing, a subset of performance engineering

7.4 INTERFACE TESTING

Interface Testing is performed to evaluate whether systems or components pass data and control correctly to one another. It is to verify if all the interactions between these modules are working properly and errors are handled properly

1. The Python script interfaces with the Arduino UNO using the serial port connection.
2. Exceptions are handled effectively by the system.

7.5 UNIT TESTING

Unit testing is a software testing method by which individual units of source code sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine whether they are fit for use. It is often done by the programmer by using sample input and observing its corresponding outputs Test cases are written for:

- Check if the loop, method, or function is working fine.
- Misunderstood or incorrect, arithmetic precedence.
- Incorrect initialization

Test Case ID	Test Case Title	Expected Output	Result
1	Working of Arduino UNO board	Arduino UNO board is functional	PASS
2	Start of Python script	Request for serial port number	PASS
3	Establish connect between Python script and Arduino UNO	Connection successful	PASS
4	Connecting Webcam With Python script	Live Video stream visible	PASS
5	Empty Platform	No Action is performed	PASS
6	Hand Rejection	No Action is performed	PASS
7	Indexed object placed on platform	Action performed. Object is sorted	PASS
8	Non-indexed object placed on platform	No action Performed	PASS

Table 3 Test Cases

CHAPTER 8:
RESULTS AND PERFORMANCE ANALYSIS

CHAPTER 8

RESULTS AND PERFORMANCE ANALYSIS

The following snapshots and graphs define the results or outputs that we will get after step by step execution of each proposed protocol for different values of time and speed.

8.1. RESULT ANALYSIS

The IoT-based system was tested for its ability to monitor and analyze milk quality. Data collected from pH sensors, gas sensors, was sent to Cloud for visualization and storage.The ultrasonic sensor successfully monitored the milk levels in the container and alerted users when levels dropped below a threshold.

The results of a milk quality analyzer project typically focus on assessing various parameters to ensure the safety, freshness, and nutritional value of milk. Key parameters measured include fat content, which determines the creaminess and energy value; protein levels, which contribute to nutritional quality; and solid-not-fat (SNF), which includes proteins, lactose, and minerals.

The analyzer also evaluates the pH level to check for acidity changes that indicate spoilage and measures specific gravity to detect potential adulteration with water or other substances. Advanced systems may assess microbial contamination to ensure milk is free from harmful pathogens and determine somatic cell count (SCC), a key indicator of milk quality and cow health. Additionally, some analyzers detect adulterants like detergents, starch, or synthetic substances.

These results, typically presented as numerical values or overall quality scores, provide valuable insights for farmers, dairy producers, and consumers, ensuring milk meets safety and quality standards.

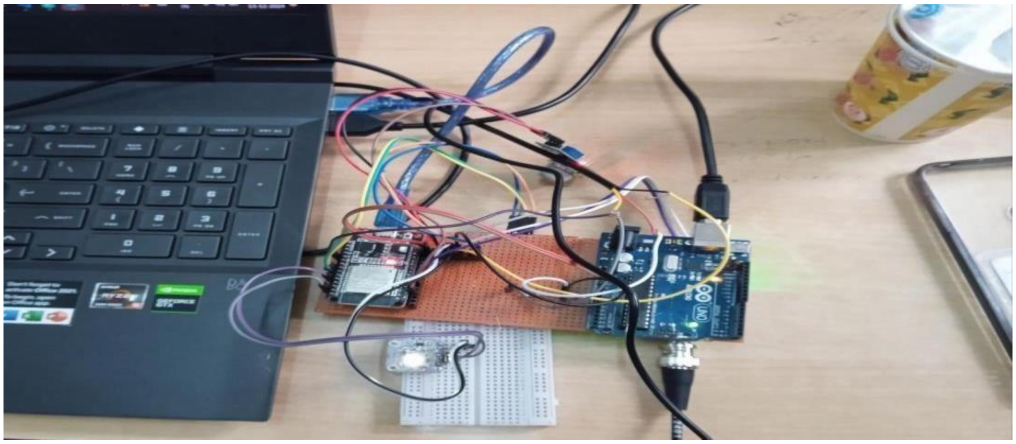


Figure 8.1 final model

The final model of an IoT-based milk quality analyzer project is designed to assess the quality of milk in real-time using a combination of sensors and IoT technology. The goal of this system is to monitor various parameters that indicate the ph,temperature, gas content, and overall quality of the milk.

Figure 8.2 welcome page

The goal of the Milk Quality Analyzer IoT-based project is to evaluate and monitor the quality of milk using real-time sensor data. The system aims to provide insights into key milk quality parameters such as fat content, protein levels, temperature, pH value, and adulteration detection.

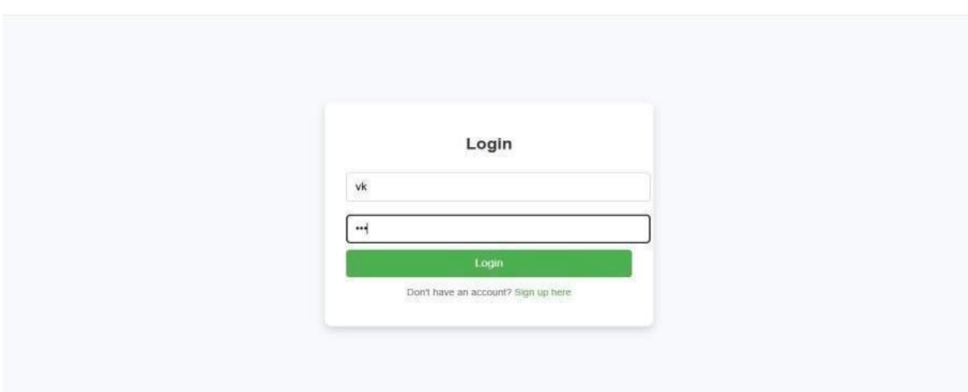


Figure 8.3 login page

Figure 8.4 welcome page

Dashboard: Users can visualize the results of the milk quality analysis on a user-friendly dashboard. This dashboard typically displays parameters like fat content, protein levels, pH, and temperature in real-time. **Historical Data**: The system allows users to track the quality trends over time, identifying potential issues in milk quality early and improving the decision-making process.

Home About Farmer Real-time

Milk Quality Data

Timestamp	Temperature	pH Level	Color Temperature	Gas Level	Ultrasonic Level	Milk Quality
2024-12-24 20:05:09	27	6.5	yellow	4095	0	Good
2024-12-24 20:05:15	30	6.8	yellow	4095	0	Good
2024-12-24 20:05:21	28	6.8	yellow	4095	0	Good
2024-12-24 20:05:26	25	6.8	yellow	4095	0	Good
2024-12-24 20:05:31	30	6.8	yellow	4095	0	Good
2024-12-24 20:05:37	34	6.6	yellow	4095	0	Good
2024-12-24 20:05:42	29	6.8	yellow	4095	0	Good
2024-12-24 20:05:47	32	6.5	yellow	4095	0	Good
2024-12-24 20:05:53	33	6.7	yellow	4095	0	Good
2024-12-24 20:05:58	29	6.7	yellow	4095	0	Good
2024-12-24 20:06:04	30	6.8	yellow	4095	0	Good
2024-12-24 20:06:09	27	6.6	yellow	4095	0	Good
2024-12-24 20:06:15	35	6.7	yellow	4095	0	Good

Figure 8.5 milk quality data page

And at the end of the analysis period, the system generates detailed reports on the milk quality, helping stakeholders assess product consistency and make necessary

IOT BASED MILK QUALITY ANALYZER

improvements. The IoT-based Milk Quality Analyzer provides a reliable, real-time assessment of milk quality by continuously monitoring key parameters. The result analysis ensures that only high-quality milk reaches consumers, improving food safety and nutrition standards. Alerts and data visualization enhance decision-making, enabling timely interventions and optimizations in the milk production and distribution chain.

This description encapsulates how data is collected, analyzed, and presented in an IoTbased Milk Quality Analyzer system. It emphasizes the real-time monitoring, automated alerts, and detailed reporting that collectively support the overall goal of maintaining milk quality and safety.

Figure 8.6 reciept page

The result analysis ensures that only high-quality milk reaches consumers, improving food safety and nutrition standards. Alerts and data visualization enhance decision-making, enabling timely interventions and optimizations in the milk production and distribution chain.

CHAPTER 9: SNAPSHOTS

CHAPTER 9

SNAPSHOTS

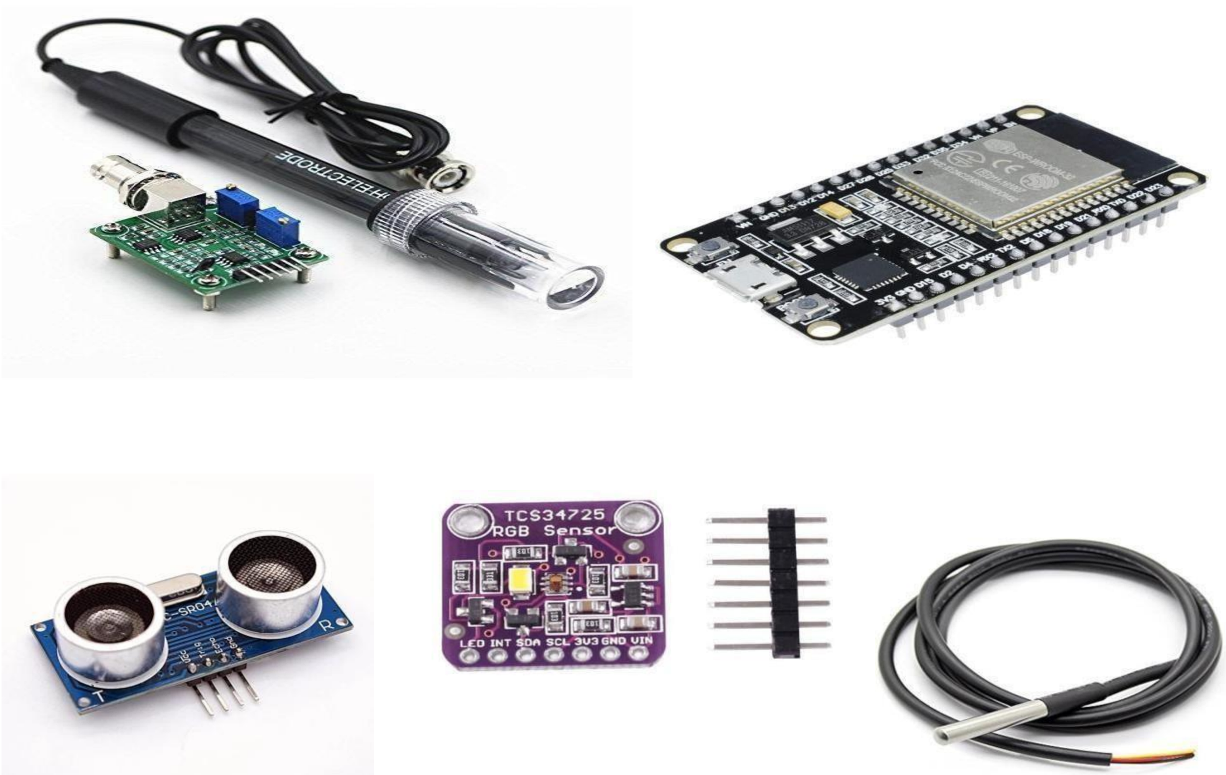


Figure 9.1 Electronic Components

pH, color, ultrasonic, and temperature sensors are essential tools for monitoring and measuring various parameters in industrial, environmental, and scientific applications. pH sensors measure the acidity or alkalinity of a solution using a glass electrode and a reference electrode, commonly applied in water quality monitoring, food safety, and chemical processing

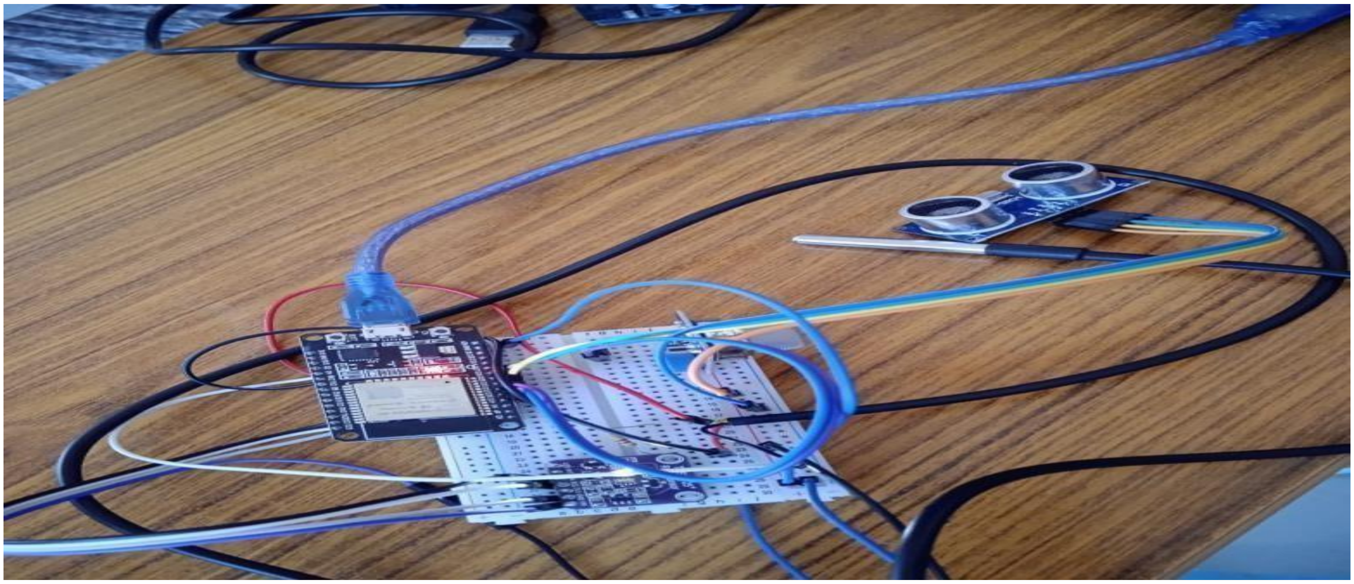


Figure 9.2 Model testing

Sensor Calibration and Integration Model testing for an IoT-based milk quality analyzer begins with ensuring the sensors used for detecting milk quality parameters (such as temperature, pH, fat content, and protein levels) are accurately calibrated

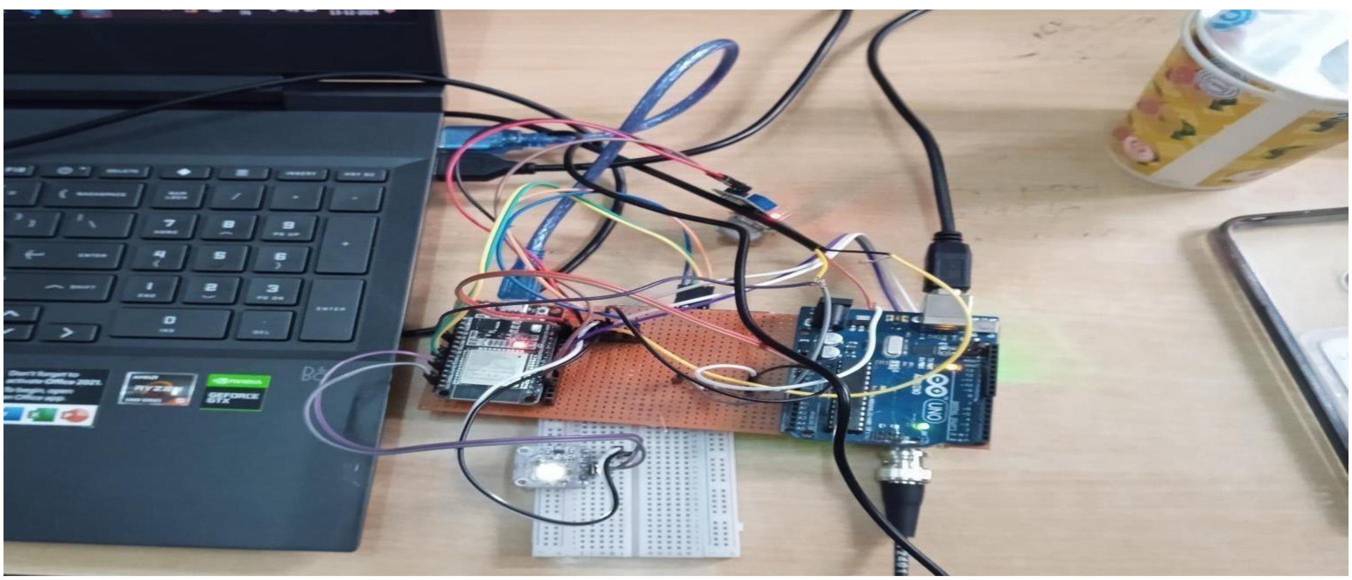


Figure 9.3 final model

The goal of the Milk Quality Analyzer IoT-based project is to evaluate and monitor the quality of milk using real-time sensor data. The system aims to provide insights into key milk quality parameters such as fat content, protein

levels, temperature, pH value, and adulteration detection
e

Figure 9.4 Welcome page

The goal of the Milk Quality Analyzer IoT-based project is to evaluate and monitor the quality of milk using real-time sensor data. The system aims to provide insights into key milk quality parameters such as fat content, protein levels, temperature, pH value, and adulteration detection.

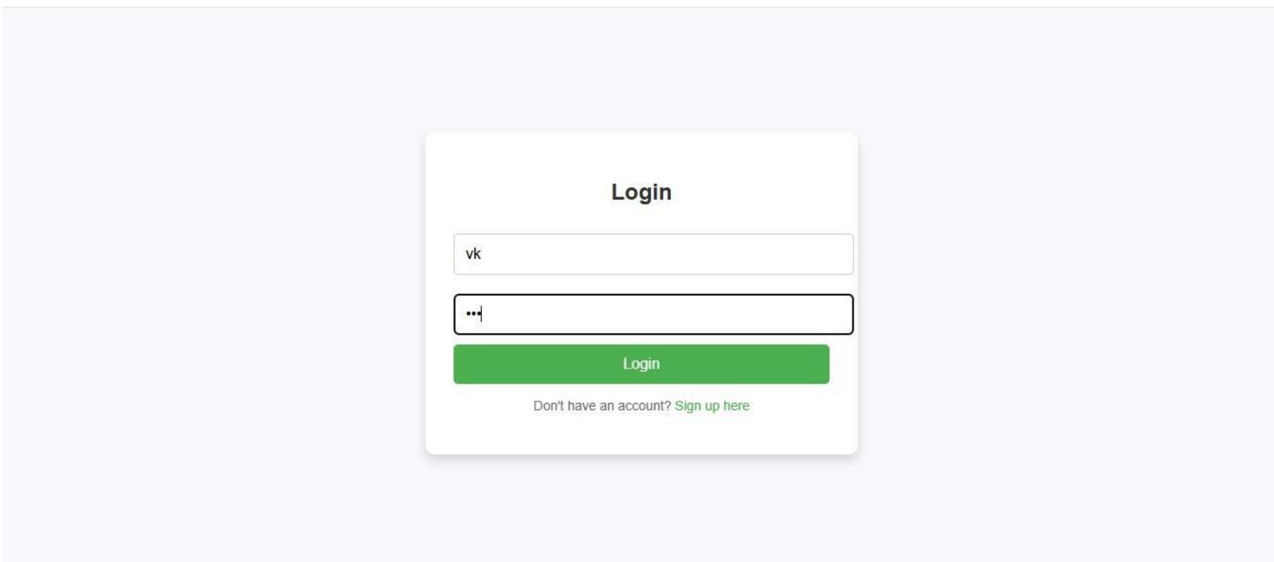


Figure 9.5 Login page

This dashboard typically displays parameters like fat content, protein levels, pH, and temperature in real-time. Historical Data: The system allows users to track the quality trends over time, identifying potential issues in milk quality early and improving the decision-making process

Figure 9.6 Home page

And at the end of the analysis period, the system generates detailed reports on the milk quality, helping stakeholders assess product consistency and make necessary improvements. The IoT-based Milk Quality Analyzer provides a reliable, real-time assessment of milk quality by continuously monitoring key parameters.

This description encapsulates how data is collected, analyzed, and presented in an IoTbased Milk Quality Analyzer system. It emphasizes the real-time monitoring, automated alerts, and detailed reporting that collectively support the overall goal of maintaining milk quality and safety.

Figure 9.7 receipt

The result analysis ensures that only high-quality milk reaches consumers, improving food safety and nutrition standards. Alerts and data visualization enhance decision- making, enabling timely interventions and optimizations in the milk production and distribution chain

Figure 9.8 Project Team

Figure 9.9 Project Team with Guide

CONCLUSION

We concluded that the proposed **Milk Quality Analyzer** system offers a groundbreaking solution to the challenges faced in the dairy industry regarding milk quality monitoring, the system provides a highly efficient, accurate, and scalable approach to milk quality analysis. It automates the traditionally manual and labor-intensive process of testing milk, ensuring consistent and rapid results with reduced human error. The system’s ability to assess key parameters like fat, protein, lactose content, microbial load, and the presence of harmful contaminants ensures that milk meets the highest safety and nutritional standards. Moreover, its real-time monitoring and instant feedback capabilities significantly enhance operational efficiency, improve product quality, and bolster consumer confidence. This system is especially valuable in regions like India, where the dairy industry faces unique challenges related to urbanization and varying waste composition. By leveraging advanced technologies, the system can help address these issues, revolutionizing milk quality testing and contributing to a more sustainable and transparent dairy industry. Ultimately, this **Milk Quality Analyzer** represents a significant step forward in improving food safety, operational efficiency, and the overall quality of milk production, positioning it as a vital tool for the future of the dairy industry.

FUTURE SCOPE

1. Enhanced Speed: Future initiatives should prioritize quicker sorting to handle more trash in less time.
2. Expansion of Recycling Options: Expanding recycling options has the potential to reduce waste even further.
3. Remote Monitoring: Remote monitoring allows operators to monitor the system from anywhere, enhancing overall management.
4. Miniaturization: Smaller and more portable systems may be employed in a variety of situations, such as households and small companies.
5. Data Analysis: Analysing data from the system can give insights into trash creation trends and advise future decisions.
6. Integration with Conveyor Belt: Using a conveyor belt could streamline garbage feeding and sorting, leading to increased efficiency.

REFERENCES

[1] [1] IEEE Paper - R. PITCHAI AND RICHARD F. HILL, Design of a Synoptic Milk Quality Monitoring System for Narragansett Bay(1972), pp. 17-24

[2] IEEE Paper - PEDRO M. RAMOS, J. M. DIAS PEREIRA, HELENA M. GEIRINHAS RAMOS AND A. LOPES RIBEIRO, A Four-Terminal Milk-Quality-Monitoring Conductivity Sensor (2008), pp. 577-583

[3] IEEE Paper - N VIJAYAKUMAR AND R RAMYA, The Real Time Monitoring of Milk Quality in IoT Environment(2015)

[4] IEEE Paper - PRASHANT SALUNKE AND JUI KATE, Advanced Smart Sensor Interface in Internet of Things for Milk Quality Monitoring(2017), pp. 208-302

[5] IEEE Paper - MONIRA MUKTA, SAMIA ISLAM, SURAJIT DAS BARMAN, AHMED WASIF REZA, M SADDAM HOSSAIN KHAN, IoT based Smart Milk Quality Monitoring System(2019), pp. 669-673

IoT Environment(2015)

[6] IEEE Paper - S.B. DIGHE, UTKARSHA KAILAS, BONDE SNEHADIP GIRIP, AND VERMA SNEHIK **,** Smart Milk Quality Analysis & Grading (2024)

[7] IEEE Paper - DRASHTI BHAVSAR, YASH JAHANPURA, VIMAL KEHANI, AND DEBABRATA SWAIN **,** Milk Quality Prediction using Machine Learning (2023)

[8] IEEE Paper - K. LALESH GIVOT, S. SIDDON, P. MINA CHOUDHARY, P. PUTALAKOTA, AND S. REDDY - IoT-Based Detection of Adulteration in Milk **(2023)**

[9] IEEE Paper - R. MEDHA KHARWAN, S. SWATHI, G. ASHWINI, AND G. SRIPRADA, assessment of Milk Adulteration IoT-Based Model to Identify Milk in Lab (2022)

[10] IEEE Paper - H. MADAN GOUDA, NAYANA P, PARDEEP J, P. KUMAR, Y. SUDHANVA, AND S.B. N, Efficient Milk Quality Detection System with IoT**(**2022)

[11] IEEE Paper - KALAIRASI GANESAN**,** ASHOK JOHN**,** AND K. DINAKUMAR - IoT-Based Milk Analysis System Using Arduino and Mobile Integration (2019)

[12] IEEE Paper - DR. P. GANESH KUMAR, A. ALAGAMMAI, B. S. MADHUMITA, AND B. ISHWARYA

[13] IoT-Based Milk Monitoring System (2019)

[14] IEEE Paper - H. MADAN GOUDA, NAYANA P, PARDEEP J, P. KUMAR, Y. SUDHANVA, AND S. B. N, EFFICIENT Milk Quality Detection System with IoT (2022)

[15] IEEE Paper - SETYAWAN P. SAKTI**,** TRI SWANTORO PUTRO**,** DEVI ANGGRAINI**,** AGIS NABA**, AND** VANY R. WIJAYANTO, IoT-Based Temperature Monitoring System **(**2021)

https://www.cookinghacks.com/documentation/tutorials/waspmote/index.html#waspmote _vs_ESP32`https://www.iotacommunications.com/blog/smart-milk-quality-monitoring-system-usingiot