

# **IOT MANAGER - LOGIN AND PERSONA MANAGEMENT**

**A PROJECT REPORT**

*Submitted by*

**ARUN UDAY (TKM21MCA-2011)**

**to**

**The APJ Abdul Kalam Technological University**

*In partial fulfillment of the requirements for the award of the degree of*

**MASTER OF COMPUTER APPLICATION**



**Changan Kunju Musaliar College of Engineering  
Kerala**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**MAY 2023**

## DECLARATION

I undersigned hereby declare that the project report on **IOT MANAGER - LOGIN AND PERSONA MANAGEMENT**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Application of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Prof. Vaheetha Salam**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not previously served as the basis for the award of any degree, diploma, or similar title by any other University.

Kollam

20-05-2023



**ARUN UDAY**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**TKM COLLEGE OF ENGINEERING**

**KOLLAM**

**2021 - 23**



**CERTIFICATE**

This is to certify that the report entitled **IOT MANAGER - LOGIN AND PERSONA MANAGEMENT** submitted by **ARUN UDAY** (TKM21MCA-2011) to the APJ Abdul Kalam Technological University in partial fulfillment of the Masters degree in Computer Application is a bonafide record of the project work carried out by him under our guidance and supervision. This report, in any form, has not been submitted to any other University or Institute for any reason.

Internal Supervisor

Head of the Department

External Examiner

---

Date:17-05-2023

TO WHOM IT MAY CONCERN

This is to certify that Mr. Arun Uday from TKM College of Engineering, Kollam has completed the internship on “**IoT Manager**” project at **Knowledge Lens Pvt Ltd** from 16<sup>th</sup> January 2023 to 16<sup>th</sup> May 2023. During the period of his training with us he was found punctual, hardworking, and inquisitive.

We wish him all the success in future endeavors.

Yours Truly,

For **Knowledge Lens Pvt Ltd**.



Jayashree S  
Sr. HR Manager  
hr@knowledgelens.com

## ACKNOWLEDGEMENT

First and foremost, I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr. Fousia M Shamsudeen**, Head of the Department, Department of Computer Applications, for providing me with the best facilities.

I would like to thank my project guide **Prof. Vaheetha Salam**, Department of Computer Applications, who motivated me throughout the project, and my external coordinators **Mr. Athreya K S** and **Mr. Irfanuddin Shafi Ahmed**, Knowledge Lens, guided me throughout my work.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my advisor and coordinator **Prof. Natheera Beevi M** Department of Computer Applications, for her expert guidance, cooperation, and immense encouragement.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspiration throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

**ARUN UDAY**

## **ABSTRACT**

**IOT MANAGER - LOGIN AND PERSONA MANAGEMENT** is a portal that is designed for monitoring and managing IoT devices. The system comprises login and user management modules. Allowing users with different personas or roles to access the portal according to their privileges.

Persona is customizable and can be created based on the responsibilities you want to assign to the user. The default roles that will be available in the system are super admin, support admin, support user, site users, vendors, and customers. Each role has custom privileges to create, view, modify, and manage.

The login and user management system developed for the platform is a crucial component, providing users with access based on their privileges. Super admins can create, view, and update users and create custom user roles. The Asset Manager module provides a comprehensive model of each piece of equipment and its associated parameters, processes, materials, and resources, allowing industries to more effectively manage their assets, optimize performance, and reduce downtime.

The alarm and rule setup module allows users to create rules to monitor each device, and rules act as thresholds to trigger alarms when certain criteria are met.

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Company Profile . . . . .	2
1.1.1 Products . . . . .	2
1.1.2 Services . . . . .	3
1.2 Existing System . . . . .	4
1.3 Proposed System . . . . .	4
1.4 Objectives . . . . .	5
<b>2 Literature Survey</b>	<b>6</b>
2.1 Purpose of the Literature Review . . . . .	6
2.2 Related Works . . . . .	7
2.2.1 Internet of Things (IoT) . . . . .	7
2.2.2 Angular . . . . .	9
2.2.3 Cross Platform App Development . . . . .	11
2.2.4 FastAPI . . . . .	11
2.2.5 NoSQL Database . . . . .	12
2.2.6 Redis . . . . .	14
2.2.7 Python . . . . .	15
2.2.8 Difference between MongoDB and PostgreSQL . . . . .	16
2.2.9 PostgreSQL . . . . .	17
<b>3 Methodology</b>	<b>19</b>
3.1 Key Features of IoT Manager Login and Persona Management . . . . .	21
3.2 Module Description . . . . .	21
3.2.1 Login Module . . . . .	21

3.2.2	User Management Module . . . . .	22
3.2.3	User Role . . . . .	23
3.2.4	Access Groups . . . . .	24
3.3	System Specifications . . . . .	24
3.3.1	Hardware Requirements . . . . .	24
3.3.2	Software Requirements . . . . .	24
3.3.3	Software Description . . . . .	25
3.4	System Design . . . . .	26
3.4.1	System Architecture . . . . .	27
3.4.2	Module Design . . . . .	27
3.4.3	System Access Model Design . . . . .	28
<b>4</b>	<b>Result and Discussion</b>	<b>30</b>
4.1	Testing Methods . . . . .	30
4.1.1	Functionality Testing . . . . .	30
4.1.2	Unit Testing . . . . .	31
4.1.3	User Acceptance Testing . . . . .	32
4.2	Output Screens and Results . . . . .	33
<b>5</b>	<b>Conclusion</b>	<b>36</b>
5.1	Future Enhancement . . . . .	36
	<b>References</b>	<b>38</b>
	<b>Appendix</b>	<b>40</b>

# List of Figures

3.1	IoT Manager System . . . . .	19
3.2	Login System Model . . . . .	20
3.3	Login Page . . . . .	22
3.4	Access Control Model . . . . .	28
4.1	Landing Page . . . . .	33
4.2	View Users Page . . . . .	34
4.3	View Roles Page . . . . .	34
4.4	Access Group Viewing Page . . . . .	35
A.1	Login Page . . . . .	40
A.2	Landing Page . . . . .	40
A.3	Device Analytics Page . . . . .	41
A.4	View Users Page . . . . .	41
A.5	Create Users page . . . . .	41
A.6	View User Roles Page . . . . .	42
A.7	Create User Roles Page . . . . .	42
A.8	Create Access Groups Page . . . . .	42
A.9	View Access Groups Page . . . . .	43

# Chapter 1

## Introduction

**IOT MANAGER - LOGIN AND PERSONA MANAGEMENT** is a platform designed for monitoring and managing Internet of Things (IoT) devices. As the world becomes increasingly connected, the number of IoT devices we rely on for daily tasks continues to grow. However, managing and monitoring these devices can be challenging, especially when dealing with large-scale deployments. The Portal addresses this challenge by providing users with a platform where they can monitor their devices. Each device sends heart-beat messages, which are captured and provide real-time data from the devices. The data is displayed in the user interface for users to view and manage. Heartbeat messages consist of device data such as device name, software name, MAC address, and more. The last sync data is analyzed to check whether the device is online or offline. The last sync time is also converted according to the time zone of the user. All heart-beat data received from each device is stored in a MongoDB database. A heartbeat engine is created for data storage and management. Based on the MAC address of the device, which is unique across each device, the device data is stored in the database. Devices that do not send heart-beat data after a certain period of time are considered offline.

Users are provided access to the platform based on their privileges. Custom roles are created, and each role has custom privileges. Super admins provide access for users. Only users with access can log in to the system. Users can log in using Google or a custom login. Forgot password and reset password functionality is provided for normal login users. Users are automatically logged out of the system after 48 hours of inactivity. Super admins have privileges to create, view, and update users in the system, as well as the ability to create custom user roles. The login and user management system developed for the IoT Manager platform is an essential component of the overall system. By allowing users to log in and access the platform only with the appropriate privileges, this system helps to ensure the security and integrity of the data being collected and analyzed.

The Asset Manager module in this project provides a comprehensive model of each piece of equipment and its associated parameters, processes, materials, and resources. This model

allows industries to more effectively manage their assets, optimize performance, and reduce downtime. By providing real-time insights into asset performance and maintenance needs, the Asset Manager module empowers industries to make data-driven decisions and improve overall operations. This model can be reused for other industries as well. Additionally, the alarm and rule setup module in the project allows users to create rules to monitor each device. Rules act as thresholds, so when a certain threshold is reached, alarms are triggered. Rule functions allow users to reuse code created for similar purposes. Rule templates generated can be used for multiple similar scenarios. In addition to the Asset Manager and Alarm and Rule Setup modules, the IoT Manager platform includes a range of other features and tools designed to make it easy for users to manage and monitor their IoT devices. These include real-time alerts and notifications, and the ability to export data and generate reports. One of the key benefits of the IoT Manager platform is its ability to provide real-time data and insights into the performance of IoT devices. With each device sending regular heart-beat messages, users can quickly identify and address any issues that arise, reducing the risk of downtime and improving overall efficiency. As the number of IoT devices in use continues to grow, the need for effective management and monitoring tools becomes increasingly important. The IoT Manager platform is designed to meet this need, providing a powerful and flexible solution for businesses of all sizes and industries. Whether you are managing a small deployment or a large-scale network of IoT devices, IoT Manager can help you stay on top of your data and make informed decisions.

## 1.1 Company Profile

The set of Lenses offered by Knowledge Lens makes it easier and more automated to find undiscovered Big Data insights. Our goal is to transform dark data into actionable business insights. Our company, Big Data Technology Geeks, has deep industry knowledge and a variety of Big Data Projects, from Data Science to Big Data Engineering.

### 1.1.1 Products

- **UnifyTwin**

Across the industrial life cycle, UnifyTwin's Intelligent Industrial App Suite, powered by AI, serves the needs of its customers. UnifyTwin is acknowledged in the Gartner quadrant and has been validated by hundreds of industrial clients in numerous industries.

- **Glens**

Software for the real-time capture, monitoring, and analysis of environmental data, Analytics for Centralized Monitoring, Web HMI, Connectivity Solution, Device Data Acquisition, Industrial App Platform, and Analyses of Environmental Data.

- **MLens**

You can manage disaster recovery for your big data and platforms with MLens, a one-step solution. Aspects of MLens:

1. Migration of Big Data Backup
2. Disaster Recovery Automation
3. Archival, compression, and data encryption
4. Rapid Data Ingestion in Batch
5. Controlling and Planning
6. Secured Access Control

### 1.1.2 Services

- **Services for Big Data Engineering**

We offer full-service big data protection architecture, design, development, testing, and deployment.

- **Services for Big Data Security**

We are one of the specialized consulting firms offering Big Data Services.

- **Services for Big Data Analytics**

Using our pre-built analytical Lens, we deliver hidden insights from a wide range of data sources.

- **Services for the Development of Big Data Competencies**

We offer one of the top enterprise big data competency development programs without specialized big data knowledge.

## 1.2 Existing System

This software program was created using outdated technology six years ago with the intention of resolving issues regarding the administration of distant edge and IoT devices. Through the centralized platform provided by the application, users were able to easily manage vast networks of devices by monitoring, controlling, and configuring them. As the years have passed and technology has progressed, it has become increasingly clear that the older technology utilized in the creation of this application has reached its limitations. The application's ability to manage the increasing amount of data produced by IoT devices and to comply with the most recent security measures may be limited due to scalability issues. In addition, the interface may not be as modern or user-friendly, posing a challenge for users to effectively retrieve and examine data from their equipment. Modern applications offer advanced analytics and machine learning capabilities that may surpass what the application currently provides.

To put it succinctly, although the previous iteration of this online program offered a practical way to regulate isolated endpoints and Internet of Things (IoT) apparatus, its capacity to grow, safeguard, and accommodate user preferences may impede its efficacy in today's digital environment. It might be imperative to enhance the application utilizing contemporary technology and functionalities in order to be in line with the latest requirements for managing IoT devices as they evolve.

## 1.3 Proposed System

The proposed system helps people look after and keep track of their IoT devices. It is a very helpful and complete tool. This tool helps people see how well their device is working right now. They can fix any problems they find right away.

- **Login and Persona Management:** The platform offers a custom login system that provides access to users with appropriate privileges. Users can log in using Google or custom login, and forgot password and reset password functionality is provided for normal login users. The platform includes custom roles and super admins who have privileges to create, view, and update users and user roles.
- **Asset Manager:** The Asset Manager module provides a comprehensive model of each piece of equipment and its associated parameters, processes, materials, and

resources. This module allows industries to more effectively manage their assets, optimize performance, and reduce downtime. By providing real-time insights into asset performance and maintenance needs, the Asset Manager module empowers industries to make data-driven decisions and improve overall operations.

- **Alarm and Rule Setup:** The platform allows users to create rules to monitor each device, acting as thresholds. When a certain threshold is reached, alarms are triggered. Rule templates generated can be used for multiple similar scenarios, and rule functions allow users to reuse code created for similar purposes.
- **Real-time Alerts and Notifications:** The platform sends real-time alerts and notifications to users when a device is offline or when a rule is triggered.
- **Export Data and Generate Reports:** The platform provides the ability to export data and generate reports for device monitoring and management purposes.

## 1.4 Objectives

Project deliverable includes:

- Revamp the existing system with new features.
- Develop a login system with normal and Google login.
- Develop a feature for user management and role creation.
- User access group creation.
- Create forgot password and reset password functionality.
- Increase productivity and create a user-friendly platform.

# Chapter 2

## Literature Survey

A literature survey, also known as a literature review, involves analyzing scholarly sources related to a particular subject. Examining the available literature, it provides a comprehensive overview of the state of the field, allowing you to identify relevant theories, approaches, and gaps in the existing body of knowledge. When conducting a literature review from an audit perspective, the main focus is on evaluating the relevant literature. This process covers information that has been published in a specific field of study and sometimes includes information published within a specific time frame. The literature review is an indispensable tool for conducting research and is frequently used as a starting point for delving into a specific subject area. In addition to identifying important theories and concepts, a literature review can also pinpoint gaps in current knowledge and draw attention to areas where further research is necessary. By scrutinizing multiple sources, a literature review can provide a more comprehensive understanding of a given topic or issue. A well-crafted literature review can also enhance the credibility and authority of the author, as it demonstrates their familiarity with the current research and debates in the field. In certain cases, a literature review may include a meta-analysis, which involves analyzing the findings of numerous studies to uncover common patterns or trends. It is important to keep in mind that a literature review is distinct from a research paper or an argumentative essay; it is instead a focused examination of the existing research and literature on a specific topic.

### 2.1 Purpose of the Literature Review

1. The purpose of a literature review is to provide an overview and analysis of existing research and literature on a particular topic.
2. It aims to identify key theories, concepts, and findings, as well as to evaluate the strengths and weaknesses of previous studies.
3. A literature review can help to identify gaps in the current knowledge and highlight areas

where further research is needed.

4. By examining multiple sources, a literature review can provide a more comprehensive understanding of a particular topic or issue.
5. Additionally, a well-written literature review can help to establish the credibility and authority of the author, as it demonstrates their familiarity with the current research and debates in the field.
6. A literature review can be a standalone piece or part of a larger research project such as a thesis, dissertation, or research paper.

## **2.2 Related Works**

### **2.2.1 Internet of Things (IoT)**

IoT, which denotes the Internet of Things, refers to a range of interconnected devices via the internet. This comprehensive review investigates the various communication methods employed in IoT towards achieving an effective smart environment. At first, the paper outlines the definition of IoT and its components, followed by an exploration of the importance of communication methods within the IoT environment. The writers have identified the difficulties that IoT communication encounters, such as security, dependability, and expandability. Subsequently, the article presents a comprehensive examination of the diverse modes of communication utilized in the Internet of Things, sorted into three distinct classifications: wired, wireless, and hybrid methods. The writers provide a thorough examination of every communication tactic, emphasizing their strengths and weaknesses. Additionally, the writers delve into the procedures utilized in IoT communication which encompass MQTT, CoAP, and HTTP. They analyze these protocols, assessing their advantages, disadvantages, and suitability for various Internet of Things (IoT) applications through comparison and contrast. The article provides an overview of the findings of the comprehensive evaluation and highlights the importance of implementing effective communication tactics within the realm of the Internet of Things. Smart environments rely heavily on effective communication, as pointed out by the authors. To achieve optimal results, communication strategies must be tailored to the unique needs of the particular IoT application. Broadly speaking, the article offers a

comprehensive examination of the methods of communication employed in the Internet of Things, as well as their impact on facilitating a streamlined and effective ecosystem. The methodology used in the study guarantees that the conclusions are derived from a meticulous and comprehensive examination of the existing literature. This article holds great worth for both IoT communication experts and professionals, providing them with substantial understandings of the diverse communication tactics and protocols at their disposal[1].

An innovative method for deploying and developing agricultural farmland, which leverages an enhanced architecture of the Internet of Things, is introduced in the form of an Intelligent Deployment and Development Strategy. The article starts by discussing the existing condition of the agriculture sector, its difficulties, and the necessity of adopting intelligent farming methods. Afterwards, the writers deliberate on the possible uses of the Internet of Things in farming and detail its different elements such as sensors, actuators and communication protocols. The proposed IoT framework for agricultural land utilization suggested by the authors involves the integration of cloud computing, machine learning, and data analytics, aimed at improving its architecture. They offer an extensive examination of every element in the structure and elaborate on its potential in facilitating the proficient and fruitful handling of agricultural land. Moreover, the writers showcase an instance of the suggested IoT structure by contextualizing it within a Chinese strawberry plantation through a case study. They showcase how the design could potentially bring about advantages such as better utilization of resources, decreased labor expenses, and an upsurge in crop production. In conclusion, the article summarizes its findings and emphasizes the importance of the suggested IoT framework in the growth and implementation of agriculture in farmlands. The authors underscore the necessity of more comprehensive exploration and enhancement in this domain so as to completely leverage the advantageous aspects of intelligent farming. In short, the article provides a thorough examination of the suggested IoT framework for the implementation and advancement of farming in rural areas. The potential advantages of the subject are examined by the writers, who present a case study to illustrate its efficacy. The information presented in this article may prove valuable for individuals involved in agricultural research and those practicing in the field who are interested in comprehending the various possibilities for IoT employment within this sector[2].

The difficulties in handling data that arise with the Internet of Things (IoT) and the numerous methods that have been devised to tackle them. The article presents a brief overview

of IoT and the different elements that comprise the ecosystem of IoT. The article proceeds to examine the difficulties faced in managing data related to the Internet of Things (IoT), such as concerns regarding protection, the capacity to grow, and the diverse nature of the data. The writers thoroughly examine all of these obstacles and clarify their effect on the management of data in IoT. Afterwards, the article delves into diverse methods of managing data that have been created specifically to manage the enormous volume of data generated by the Internet of Things devices. Various methods such as data filtering, data aggregation, data fusion, and data mining are employed. The strategies' mechanics are detailed by the writers, along with their application for scrutinizing IoT data and obtaining useful findings. The article delves into the significance of cloud computing in managing data for IoT. Moreover, the writers deliberate on the adoption of edge computing as an alternate strategy for handling IoT data, wherein data analysis and processing happen at the network's edge - in the vicinity of the devices that are producing the data. The authors' concluding remarks emphasize the significant role of competent data handling in realizing the complete potential of IoT as they summarize their findings. According to them, implementing data management strategies such as filtering, aggregation, and mining can assist in extracting significant findings from IoT data, leading to innovation and advantages for numerous sectors. In summary, the paper offers a thorough look at the data management obstacles presented by the Internet of Things (IoT) and the diverse methods created to tackle these challenges. This underscores the significant role that data management plays in harnessing the full potential of the IoT. It should be emphasized that the realm of IoT data management is constantly progressing, and novel methodologies and technologies are constantly emerging. Hence, it is imperative to conduct additional research in order to stay updated on the latest advancements in this field[3].

### **2.2.2 Angular**

An extensively utilized platform for building web and mobile applications is Angular. The paper commences by presenting the Angular framework and delving into its remarkable features and advantages. They provide an explanation of the constituting elements of an Angular application, namely, modules, components, services, and directives. Subsequently, the article presents a comprehensive evaluation of the Angular structure, which encompasses the hierarchy of components, the detection of changes, and the injection of dependencies. The manner in which the architecture components function and combine to form a potent and

adaptable application development platform is explicated by the writers. In addition, the writers conduct a thorough examination of different Angular aspects like templates, data connection, pipes, and directives. The roles and functions of each feature are explained and illustrated with code snippets and examples. Angular testing underscores the crucial nature of verifying code and the high utility of Karma and Jasmine as testing tools. Ultimately, the paper's conclusion restates the main ideas and underscores the benefits of employing the Angular framework for the development of both web and mobile applications. In essence, the article delivers an extensive and detailed examination of the Angular framework, concentrating primarily on its structure and crucial aspects. The paper's coherence is greatly aided by the authors' inclusion of practical examples and code snippets, which effectively conveys applicable wisdom to developers and researchers concerning the capacities and operations of the Angular framework. The contents of the paper hold value for individuals involved in web and mobile application development[4].

Creating a way for people and machines to communicate on the internet for controlling factories using a program called Angular. The writers talk about why web-based HMIs are better than traditional HMIs. They are easier to get to, work on any device, and can handle bigger projects. They mention that more and more factories are using computer screens to control their machines. The writers explain about Angular, a famous free web tool to make one-page apps. They explain what it can do and why it's useful, like being able to make code that can be used over and over again, and how it allows for reactive programming. The writers explain how their online control panel works. It has two parts: an Angular program on the user's computer and a Node.js program on the server. They clarify how a computer program on your device talks to the machine controller through a RESTful software, and how the program on the server verifies who you are and stores data. The writers talk about different parts of an Angular app. The main part is where you start the app and see the menu. The dashboard part shows information from a machine in real-time. They show examples of how to use these parts of the program by giving code samples. At last, the writers show what they found out about the web-based HMI system. They asked industrial control engineers to test the system and they said it was easy to use and worked well. The writers say that their computer system for controlling machines at work is helpful and easy to use. It can work for many different situations. The paper explains how a web-based system called Angular is used for industrial automation. It discusses the benefits of using this system and provides a lot of information

about it. The people who wrote about the system say that it works well and can be used in many other places too[5].

### **2.2.3 Cross Platform App Development**

An examination of the merits and drawbacks of various strategies for cross platform mobile application development. The writers commence their discussion by defining the two methodologies for development and exploring their contrasts. Next, they explore the different elements that developers ought to take into account when deciding which of the two options to opt for. The writers offer a comprehensive analysis of past research on the subject of native versus hybrid development for cross-platform, highlighting the conclusions of each study. They emphasize the significance of performance, user satisfaction, duration and expenditure of development, divergence in platforms, and assistance of tools as critical elements to weigh when deciding between native and cross-platform hybrid development. In addition, the writers delve into a range of tools and frameworks that facilitate hybrid cross-platform development, including Apache Cordova, React Native, and Xamarin. They evaluate the functionalities and strengths of these instruments, along with their efficiency and user-friendly nature. Moreover, the authors converse about the difficulties pertaining to cross-platform hybrid development including debugging complications, platform-specific attributes, and hardware constraints. They offer suggestions on how developers can successfully tackle these obstacles. In essence, the writers deduce that both native and cross-platform hybrid development techniques offer their own merits and demerits; Nonetheless, the ultimate decision is contingent upon the particular requirements of the project along with the resources obtainable to the development crew. The article presents a beneficial summary regarding the subject matter by amalgamating insights from past research and presenting practical guidance for developers[6].

### **2.2.4 FastAPI**

FastAPI is a web API development framework designed to be used with Python. The article gives an in-depth explanation of the framework's components, structure, and instructions for constructing a basic API from the ground up. The article presents an overview of FastAPI, emphasizing its significant attributes such as efficient speed, typed notations, automated API documentation, and integral backing for contemporary Python norms such as async/await. Following that, the text proceeds to clarify the fundamental principles of both HTTP and

RESTful APIs before delving into the primary aspects of FastAPI. The article encompasses various subjects such as the delineation of endpoints, the design of data models, the management of requests and responses, the handling of mistakes, and the execution of authentication. Code snippets and examples are provided for each topic, facilitating developers in comprehending and initiating FastAPI. In brief, "Getting Started with FastAPI" is an intelligently crafted and extensive manual for constructing web APIs by utilizing FastAPI. The article serves as a strong basis to guide novice developers in utilizing the framework to create efficient web APIs with ease and speed[7].

### 2.2.5 NoSQL Database

A new technique is proposed for enabling attribute-based access control (ABAC) in NoSQL databases, which involves the implementation of this security mechanism. The inadequacies of conventional access control frameworks, like RBAC, to manage the intricate access control prerequisites of contemporary applications are highlighted by the writers. The paper puts forth a unique solution to tackle the restrictions in NoSQL database's access control mechanism by introducing an innovative approach that relies on ABAC. This technique permits access decisions to be determined by user characteristics, resource characteristics, and contextual factors. The model presented by the authors outlines a method for implementing Attribute-Based Access Control (ABAC) in NoSQL databases. It involves a language for creating access control policies and an enforcement engine to ensure compliance. Moreover, the article explores the application of the suggested ABAC framework in Apache Cassandra, a prevalent non-relational database. The writers detail the changes implemented on Cassandra in order to facilitate ABAC. These changes incorporate the integration of a policy engine and modification of the data model to accommodate attribute-based access control. The proposed ABAC model was put to the test by the authors, who evaluated its performance in Cassandra and compared it to RBAC. The study shows that while the ABAC approach exhibits comparable results to RBAC for small and medium databases, it outperforms RBAC in the case of larger databases. In essence, the article presents an innovative method for regulating access in NoSQL databases through the application of ABAC, successfully showcasing its practical implementation in Apache Cassandra. The ABAC model, as proposed, provides an adaptable and expandable access control system suitable for current applications and has the potential to be used for various other NoSQL databases[8].

This study compares how well MongoDB and MySQL work for managing IoT data in different clouds. They want to see which one is better. The writers test how well these databases work by looking at how fast they answer questions, how much work they can handle, and how easy it is to make them bigger. Usually, MongoDB works better than MySQL for programs that need to handle a lot of data quickly. This is because MongoDB can work with lots of unorganized data as it uses a document-based model. This paper looks at how well MongoDB and MySQL work for saving and getting sensor information in an IoT setup. The research showed that MongoDB was better than MySQL at finding information quickly and processing a lot of data at once. This research compares how well MongoDB and MySQL databases can keep and get social media information. The research showed that MongoDB was better than MySQL when it came to finding data quickly and being able to handle lots of data. The writers of the article test how well MongoDB and MySQL databases work for managing information from the Internet of Things (IoT) in different online places like Amazon Web Services, Microsoft Azure, and Google Cloud Platform. The writers did some tests to see how fast the databases responded to requests, how much they could handle at once, and how well they could handle larger amounts of data. The experiments found that when managing IoT data in three different cloud environments, MongoDB is faster and can handle more queries than MySQL. MongoDB works better than MySQL on AWS and Azure, but not on GCP when it comes to scaling up. Basically, when it comes to apps that need to process lots of data quickly, MongoDB is better than MySQL according to what experts have written. This study shows that these databases work differently in various cloud setups. The research made a pretend stream of information from IoT devices. They saved the information in two different databases, MongoDB and MySQL. We checked how well each database worked by asking it to do a specific set of tasks. The findings showed that MongoDB was better than MySQL in speed and efficiency for answering queries in all three cloud settings. This means that MongoDB had quicker response times and could handle more queries at a time compared to MySQL. In AWS and Azure, MongoDB is more scalable than MySQL, but not in GCP. The writers think that MongoDB's way of storing data, which is like a bunch of documents, is better for managing messy IoT data. They said that the way MongoDB is designed to group and analyze data is very effective and useful. This paper shows that MongoDB is better than MySQL for managing data in the internet of things (IoT) in the cloud. MongoDB is faster and can handle more data, which is important when processing and analyzing data in real-time for IoT applications. [9].

## 2.2.6 Redis

Redis is a type of database that is open-source and uses in-memory technology to store data. It has the ability to support multiple data structures. The article commences by discussing how conventional relational databases are insufficient for managing the rapid read and write procedures demanded by vast web-based applications. Redis was created as a substitute for customary databases, providing high-speed responsiveness and dependable scalability. The article goes on to explain the crucial attributes of Redis, including its versatility in accommodating a diverse array of applications, ranging from mere caching to intricate data analysis, thanks to its backing of data structures like lists, hashes, strings, sets, and sorted sets. The article highlights the resilient features of Redis, such as its ability to maintain accessibility through replication and failover support, even in the event of a node failure. In summary, the article emphasizes the various applications of Redis, including but not limited to, live data analysis, message waiting and user session management. Moreover, the advantages of using Redis in comparison to conventional databases like its quick response time and capacity to manage rapid read and write operations are also highlighted. This document presents a comprehensive analysis of Redis and its essential characteristics, rendering it an indispensable asset for individuals striving to comprehend this extensively employed key-value database. Redis has gained significant popularity globally as a database since its release, with a thriving community of developers actively contributing to its continuous improvement and advancement[10].

Redis is a tool that helps organizations quickly and easily access large sets of data. It is free to use, and many businesses find it useful for their needs. Redis is really helpful because it is fast and doesn't make you wait long. But sometimes, it can lose information because of things like computer problems or broken hardware. To solve a problem, a paper called "A data recovery technique for Redis using internal dictionary structure" came up with a new way to get back data by using Redis's internal dictionary structure. To get back lost information, we look at a book of words and use the information inside to find it again. The text talks about Redis, a type of storage tool that uses a special way of organizing information called a dictionary. This dictionary organizes information into key-value pairs. The dictionary is like a filing system where each word has a label (key) and a definition (value). The writers want to say that the dictionary is very important for Redis to work properly. If something goes wrong with the

dictionary, it can lead to data loss. After that, the paper explains how to recover data using the suggested method. This way includes looking through a book of words and finding missing information by using the words and their meanings saved in it. The writers say that this method works well because the dictionary has everything needed to bring back the data, like important information and details. To test how well the new method works, the writers did some tests in a pretend Redis place. The tests show that the method can get back lost information very well and without using too much extra resources, even when a lot of data is missing. The paper comes up with a new way to get back lost data in Redis by using the dictionary structure inside. This tool is useful for companies that use Redis to store data because it is easy to use and works well[11].

### **2.2.7 Python**

The programming language Python has gained popularity for its flexibility, ease-of-use, and comprehensibility. Thanks to its expansive set of pre-existing tools, robust support from developers worldwide, and ability to be applied across varied disciplines, Python has become a top choice for tasks ranging from data analysis and website creation to advanced science and AI development. In addition, the writers deliberate on a few advantages of Python in comparison to other coding languages, such as its uncomplicated acquisition, compatibility with any operating system, and adaptability. Python code samples are offered for a variety of assignments, including but not limited to data display, web extraction, and machine education. According to the authors' analysis, Python exemplifies the upcoming era of programming languages, and its widespread adoption is likely to soar due to its user-friendly nature and aptitude for adeptly managing complicated functions. It is proposed that acquiring Python skills is crucial for individuals in the technology sector, regardless of whether they are students or professionals, owing to the various prospects it offers. In summary, this article provides a succinct and understandable summary of the characteristics and uses of Python, convincingly demonstrating why it is regarded as the programming language that will lead the way in the future. The provided content features samples of code that cater to individuals with varying levels of programming expertise, from novices to professionals[12].

Internet of Things (IoT) technology is used to monitor smart environments, especially using the Python programming language. This paper studies how smart devices are used to monitor the environment. It looks at different methods and ideas by reading many things that have

already been written about it. A special computer program called PostgreSQL to help their solution work. This paper talks about how PostgreSQL can be helpful in storing and managing data for things like smart devices. It can handle a lot of data, is strong, and can work with different kinds of data. The paper looks at the problems of watching over smart environments using IoT technology, like getting data, handling it, and studying it. The writers propose an answer that uses easy-to-use computer tools like Flask, MQTT, and Pandas. This will help solve problems when handling and studying data quickly and well. This article talks about using technology to monitor our environment and keep it safe. It also mentions a specific computer program called PostgreSQL that can help manage the data gathered from these monitors. This paper talks about different ways to monitor smart environments using the Internet of Things (IoT). It explains how to use Python and PostgreSQL to do this. Researchers and people who want to do this kind of work will find the paper useful. The authors suggest using a tool called Python, along with some other software, to help monitor smart environments that use the Internet of Things. They use Flask to make websites, MQTT to send messages, and Pandas to study data and find patterns faster and easier. In simple words: The writers suggest that using PostgreSQL for managing data has many benefits, like being able to handle lots of information, being strong and not easily broken, and being able to deal with complex types of data. Finally, the paper ends by talking about what people can do in the future to learn more about this topic. The writers propose trying out new technologies like blockchain and edge computing for monitoring smart environments that use the Internet of Things. They say that it's important for people from different fields to work together to solve the difficult problems that come with using IoT technology to watch over and take care of the environment. To put it simply, The paper is a good article that tells us about how we can use IoT to monitor our environment. It gives us a useful summary of what's happening in this technology field right now. This paper looks at what other experts have already written about a topic, and also gives practical examples of how to use computer programs called Python and PostgreSQL. It can help people who are studying this subject or who want to use it in real life[13].

### **2.2.8 Difference between MongoDB and PostgreSQL**

The contrast in functionality of MongoDB and PostgreSQL. By analyzing the performance characteristics of both databases, the authors investigate their appropriateness for diverse applications across different workloads. The article serves as an introduction to the various

categories of NoSQL databases. Subsequently, it provides a summary of the distinctive aspects of MongoDB and PostgreSQL, examining the benefits and drawbacks of utilizing both. The writers also provide a concise account of the approach utilized to carry out the assessments of performance. The scholars proceed to delve into an analysis of their assessment findings. The experiments were conducted using three distinct levels of task demands: those that emphasized reading, those that focused on writing, and those that involved a combination of both. According to the study, MongoDB demonstrated superior performance in situations where there were more read operations, but PostgreSQL outshone MongoDB in instances where there were greater numbers of write operations. Yet, when faced with a combination of workloads, the disparity in performance between the two databases was negligible. The article also evaluates how distinct set-ups affect the efficiency of each database. The study concluded that adjusting specific factors such as buffer capacities, connection quantities, and indices could considerably enhance the efficiency of both databases. In their report, the writers provide a condensed version of their discoveries and offer suggestions on selecting either MongoDB or PostgreSQL depending on the specific demands of the application. According to them, MongoDB is a suitable option for applications that prioritize quick read access and the ability to expand easily. On the other hand, PostgreSQL is better suited for applications that necessitate compliance with ACID principles, advanced querying capabilities, and support for intricate data types. In general, the article offers a valuable assessment of the strengths and weaknesses of MongoDB and PostgreSQL, presenting a useful comparison between the two. It should be highlighted that the performance of the two databases can be subject to variation depending on the unique application scenario and the hardware and software setup employed[14].

### **2.2.9 PostgreSQL**

PostgreSQL Tripal, we need a strong and expandable way to handle and examine complicated genetic and physical information. This information is important for making progress in genomic research. This article talks about a way to solve a problem using a database system called PostgreSQL. PostgreSQL is a free and dependable database system that can handle a lot of information and work fast. The writers describe how they improved the way information is stored and retrieved using a tool called PostgreSQL. They used special techniques, such as dividing the data into smaller parts and arranging it in a certain way, to make it faster to find what they need. The writers explain how they connected a database called PostgreSQL to a

program called Tripal. Tripal helps make scientific databases available online for people to use freely. Integration helps users to handle and study big genetic and physical data online. This article shows how using PostgreSQL-Tripal can help manage and study big collections of information about genes and traits in different species. It gives examples of this technology working well in real situations. The examples prove the solution can handle lots of information without slowing down. This is good for researchers who need a strong and efficient way to manage their data. The paper explains a new way to handle lots of information about genes and traits using PostgreSQL and Tripal. This way is much better than the old ways of managing data. The writers show how their idea can help with real life issues, and they offer helpful tips on how to use and make it better. The article is written well and has convincing information. It's a great resource for genomics researchers who want an effective way to manage data[15].

# Chapter 3

## Methodology

**IOT MANAGER - LOGIN AND PERSONA MANAGEMENT** is a platform that allows users to access information about their product's statistics and analyses. The system has a portal that enables users to view and manage all their products in one place. Each device is programmed to send heartbeat messages at regular intervals, which contain essential information such as the device's Mac address, IP address, and other device information.

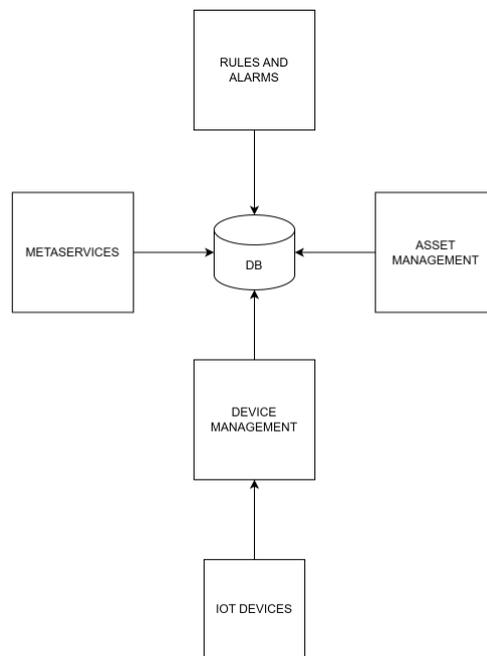


Figure 3.1: IoT Manager System

The system has a login system that provides both normal and Google login options. Users can log in using their Google account or create a new username and password for the normal sign-in. However, access to the system is only granted after the account has been authorized by the super admin. The super admin has the authority to manage users and add new users to the system. Authorized users can view their devices, get reports, and access information about their products. In case users forget their login credentials, they can use the 'forgot password' functionality in the system. They can retrieve their account using either their username or email

address.

The Mac address of each device is unique and is generated during the manufacturing process. The system's unique Mac address generation during manufacturing ensures the uniqueness of each device, making it easy to identify them in the platform. The IoT Manager offers an easy-to-use interface for users to monitor their devices. The platform provides customized views for different users based on their roles in the system. The system has predefined roles for super admin, support admins, support users, site users, vendors, and customers. You could give a specific user the ability to just view device details based on sites or locality. You can also create a new role just to manage logs, generate reports, or manage users. Create, edit, view, delete and clone abilities can also be provided based on the feature provided to the user. Additionally, the super admin can control user roles and assign customized views to each user in the system.

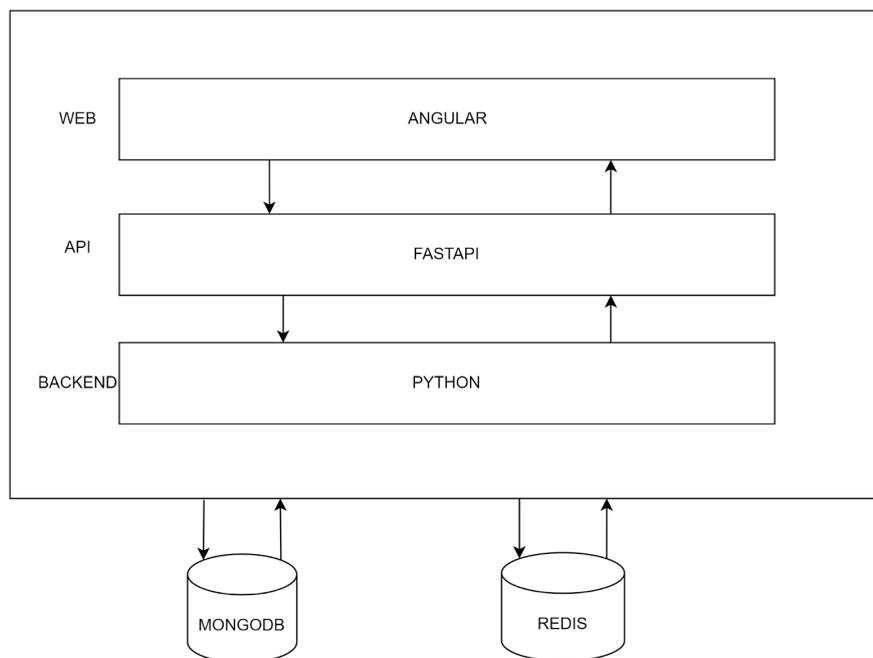


Figure 3.2: Login System Model

The IoT Manager provides users with a comprehensive platform to manage their devices, access product statistics, and receive reports. The login system with two different options, along with the personalized view for different user roles, makes it easy for users to monitor their devices and get valuable insights.

## 3.1 Key Features of IoT Manager Login and Persona Management

- Login page with normal and Google login.
- Forgot password functionality.
- User management:
  1. User creation.
  2. Updation (Role).
  3. Deletion.
- User Role:
  1. Create new roles.
  2. Add privileges to each role.
  3. Create, update, delete, and clone ability.
- Change Password functionality.
- Access is based on roles and privileges.

## 3.2 Module Description

### 3.2.1 Login Module

The login module contains a login page through which the users can access the system. The system is developed to accommodate both normal and Google login. Only the super admin can register a user into the system. During the creation of the user, the super admin could define the role of the user and provide necessary privileges to that specific user.

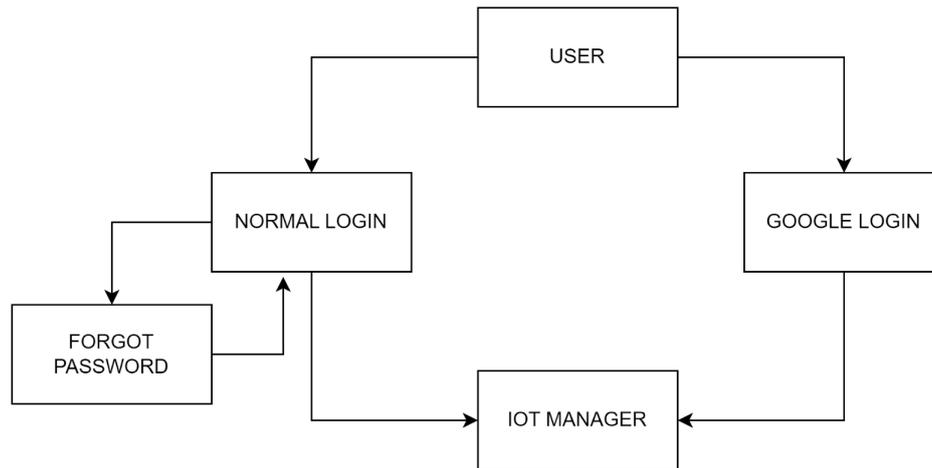


Figure 3.3: Login Page

Google login is provided using the client id generated and validated in the backend using the token generated after the login. Each user in the system will have a unique user id generated which is then used for identifying the user. Forgot password functionality allows the user to easily recover their accounts. An email will be generated with an encrypted token that will be used for the recovery purpose. After successful login, the user will be redirected to the IoT Manager home page. User login authentication is done through cookies and Redis db data. When a user logs in to the system login information will be stored as a cookie and redis data in the server. Only logged users can access the portal. After 48 hours of inactivity, the users will be automatically logged out of the portal.

### 3.2.2 User Management Module

This module is only available for the admin or we can create a new user role for the same. The user management module controls the users in the system, also allowing the admins to add new users. The user registration page lets the admin decide the roles and the login type of the user. Either the user can be made to log in using a Google account or by normal login. Google account users do not require a username for their login. Login time and account creation time are stored when necessary which can also be managed while registering a user. Admin can set the password for the new account once the account is created the admins will not be able to modify the password. The update is a limited functionality only certain fields are allowed for updation. User deletion allows the admin to remove a user from the system.

### 3.2.3 User Role

The default user roles available in the system are:

- **Super Admin:** A user with full access to all features of the application. They are responsible for managing and maintaining the entire application, including user access, system configurations, and settings. They have extensive knowledge of the application's functionalities and can perform advanced tasks such as creating custom reports and dashboards.
- **Support Admin:** A user responsible for managing user access and providing support to users who have questions or issues with the application. They can create and manage user accounts, reset passwords, and troubleshoot common issues. They are knowledgeable about the application's basic functionalities and can provide basic technical support to users.
- **Support User:** A user who provides support to site users and vendors. They have access to the application's support features and can assist users with common issues such as device configuration and troubleshooting.
- **Site User:** A user who works at a manufacturing site and uses the application to manage and monitor edge and IoT devices. They have access to the features that enable them to monitor device performance, track device status, and configure device settings. They have basic technical knowledge about the application's functionalities and can perform tasks such as setting up alerts and creating custom reports.
- **Vendor:** A user who provides devices and services to the manufacturing site. They use the application to monitor device performance, track device status, and configure device settings. They have a good understanding of the application's functionalities and can perform tasks such as integrating new devices and providing support to the site user.
- **Customer:** A user who uses the manufacturing products produced by the manufacturing site. They have access to the application's features that enable them to monitor product quality, track product delivery status, and provide feedback on products. They have basic technical knowledge about the application's functionalities and can perform tasks such as providing feedback on product quality and delivery.

- Overall, the application's user personas have varying levels of technical knowledge and responsibilities within the application. The personas enable the application to provide different levels of access and support to users with different roles and responsibilities, ensuring that the application meets the needs of all users.

### 3.2.4 Access Groups

Access groups can be created by the admin. This module is for grouping users with different roles into a group of users with similar access functionalities to the system. You can create new access groups with a description to know what the access group is for.

## 3.3 System Specifications

The software and hardware specifications recognized for the system on the basis of their requirements are specified in this section.

### 3.3.1 Hardware Requirements

- Processor: Minimum 1 GHz (Recommended 2GHz or more)
- Hard Drive: Minimum 4 MB (Recommended 6 GB or more)
- Memory (RAM): Minimum 1 GB (Recommended 4 GB or above)
- Internet connection

### 3.3.2 Software Requirements

- Angular - Front End
- Fast API Python - Back End
- Web Server - Apache or Nginx
- Mongo, PostgreSQL, Redis, Timescale - Database
- Windows, Mac, Linux, Any - OS
- Mozilla Firefox, Microsoft Edge, Any - Browser

### 3.3.3 Software Description

- **Angular:** Angular is an open-source framework for developing web-based applications that is based on TypeScript. It is currently one of the most widely used UI frameworks. Angular makes it easy to create faster web applications and is a platform that simplifies the process of creating web applications. Angular provides a powerful and straightforward template syntax that allows for the rapid generation of UI views. The key features of Angular include its component-based structure, use of TypeScript, and the ability to create services that can be used across multiple applications.
- **Python:** Python is a high-level, object-oriented programming language with an interactive, interpreted design that is widely used for general-purpose programming. It allows developers to express complex concepts in just a few lines of code and has a vast library and an extensive community of supporters. Additionally, Python runs on various operating systems, including Windows, Linux, and macOS. Guido van Rossum created Python in the late 1980s as a replacement for the ABC programming language. It was initially released as Python 0.9. In 1991. Python 2.0 introduced new features such as cycle-detecting garbage collection, list comprehensions, reference counting, and support for Unicode in 2000. Python 3.0, released in 2008, was a significant update that was not entirely backward compatible with previous versions. Python 2 was abandoned in 2020 with the release of version 2.7.18. Python is a great choice for developers due to its compatibility, readability, maintainability, interactive nature, and robust standard library.
- **MongoDB:** A flexible and scalable NoSQL database system known for its versatility, MongoDB manages document-oriented information. It is widely used by high-profile businesses due to its several features such as flexible schema design, load balancing, and horizontal scalability. Unlike relational databases, NoSQL databases store data schema-less, in multiple collections and nodes, and are scaled horizontally, supporting limited join queries. MongoDB offers support for multiple programming languages, including C, C++, C#, Go, Java, Python, Ruby, and Swift. Python is a great choice for developers and is used with MongoDB due to its compatibility, readability, maintainability, interactive nature, and robust standard library. MongoDB can also be used as a file storage system, known as GridFS, and provides different ways to perform aggregation operations.

- **PostgreSQL:** PostgreSQL is a robust object-relational database system that merges SQL language with various abilities to store and scale data efficiently. Its development has spanned over 30 years, and it was initially part of the POSTGRES project at the University of California, Berkeley. PostgreSQL is famous for its reliability, data integrity, and wide range of features, making it a dependable choice for developers. The open-source community behind PostgreSQL is devoted to providing high-performing and innovative solutions, solidifying its reputation. Moreover, PostgreSQL has been ACID-compliant since 2001 and is compatible with all major operating systems.
- **FastAPI:** FastAPI is a Python-based web framework that enables developers to create APIs quickly and efficiently. It is designed to be user-friendly and provide high performance. FastAPI uses modern Python features, such as type hints and async programming, to help developers write high-performance code. It also includes built-in support for popular databases such as MongoDB. Additionally, it offers tools for data validation, serialization, and documentation. FastAPI is lightweight and easy to learn, making it a popular choice for developers who want to build fast and scalable APIs.
- **Redis DB:** Redis is a popular open-source in-memory data structure store that serves as a versatile database, cache, and message broker. It offers support for various data structures, including strings, hashes, lists, sets, and sorted sets. Redis is renowned for its impressive speed and its ability to handle vast amounts of data. Besides, it comes with several built-in features such as replication, transactions, Lua scripting, and pub/sub messaging. Redis is widely used in web applications, message queuing systems, and real-time analytics. It also provides client libraries for different programming languages, including Python.

### 3.4 System Design

The system design phase effectively bridges the gap between the issue area and the current system, addressing the solution area - specifically, "how to implement?" During this stage, the SRS documentation is converted into an effective implementation format that also outlines the system's operation. The complex task of system development is broken down into manageable sub-activities that work together to achieve the ultimate goal of system development. A good

design enables the production of efficient code and minimizes implementation size based on the application and project requirements.

System design comprises two levels: logical design and physical design. Logical design provides an abstract illustration of the system's inputs, outputs, and data flow, meeting user needs in describing inputs (sources), outputs (destinations), databases (data storage), and processes (data flows). Logical design is crucial for the successful implementation of a corporate database. An inadequate logical design may require costly modifications to data collection, storage, and security later on. A sound preliminary design simplifies database development and evaluation, contributing to successful execution. The system analyst specifies user requirements in detail, essentially dictating the information flow into and out of the system and the necessary data sources during logical system design.

When developing a system's logical design, the system analyst specifies user needs with enough detail to determine how information enters and leaves the system and the required data sources. Diagrams of data flow and E-R diagram modeling are used. Additionally, the input/output media is specified, the database is designed, and backup procedures are established. Finally, system implementation is planned.

### 3.4.1 System Architecture

The proposed architecture for the IoT Manager platform is a web-based client-server architecture, where the server is responsible for managing the devices and data storage, while the client provides the user interface for monitoring and managing the IoT devices. The server side includes a database management system and a web server that receives data from the devices and sends it to the client via APIs. The client side includes a web application that can be accessed via any modern web browser.

### 3.4.2 Module Design

The **IOT MANAGER - LOGIN AND PERSONA MANAGEMENT** is designed in such a way that the users can login in to the system either using general login using username and password or by using Google login. All the API's are designed and implemented using FastAPI framework which is faster than any other API that is available. For the backend python is used for the development which is one of the most flexible and versatile programming language which also adds to the core design of the system.

- Login Module: It consists of two ways to login using username and password and Google login.
- User Management: Admins can added new users to the system. During the creation of the user we can also assign roles of the users.
- Role Management: Custom roles with custom roles can be created.
- Access Groups: Groups based on access can be created and assigned to each roles and users.

### 3.4.3 System Access Model Design

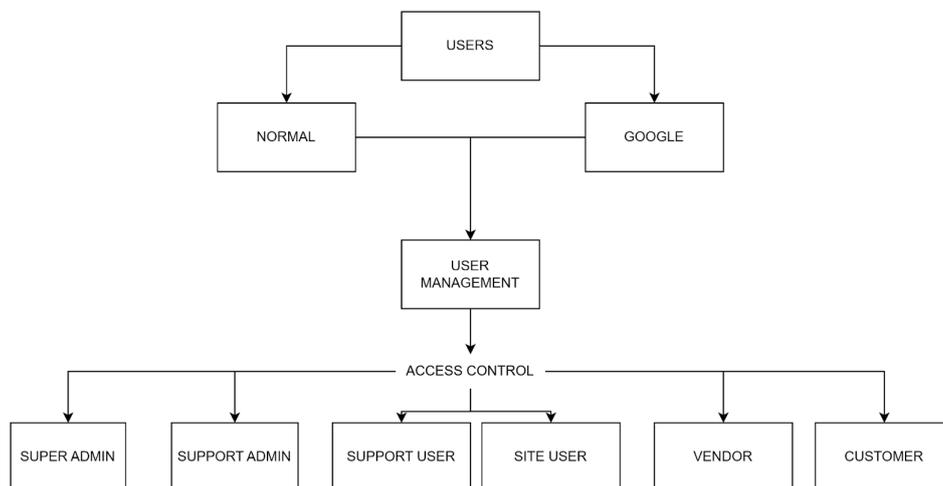


Figure 3.4: Access Control Model

The above-shown access control model describes how access is given to each user. Users log in as either normal or Google user. This is assigned by the admins of the system. Access control is divided into different roles available in the system. Also, custom roles can be created and made available to be given to users.

- Super Admin: Super Admin is a user that has complete access to the features of the application. He manages all the users and configures settings and device data. They can perform advanced operations like the creation of custom reports, dashboards, and asset management.

- **Support Admin:** Support Admin is able to manage users and provide support to user queries and issues that arise in the system. Create users provide passwords, and troubleshoot issues. Provide technical support to users.
- **Support User:** The user who supports site users and vendors. Support users and assist users with solving common issues.
- **Site User:** Site user has access to features that allows them to monitor, track and configure their devices. They are able to perform basic technical functions on the application and can perform tasks such as setting up alerts and report creation.
- **Vendor:** They monitor device performance and track and configure device settings like the site user but these user act as vendors to other customers. We can easily distinguish vendors from customers. Sometimes the company acts as the vendor but in most cases, we sell devices to other vendors who then sell these devices to their customers.
- **Customer:** They are able to monitor, track and manage their device. These devices are directly sold to the customer from the company. The devices sold by other vendors to their customers are not under this category.
- **Custom Roles:** We can create custom roles with custom privileges. These roles can then be given to any user. In scenarios such as we want a specific role that allows the user to manage a device on a specific site then we can customize the role to that purpose and assign that role to the user. Also, we can reuse this role and enable a different user to manage the same site.

# Chapter 4

## Result and Discussion

**IOT MANAGER - LOGIN AND PERSONA MANAGEMENT** provides a reliable and efficient solution for monitoring and managing IoT devices. The heartbeat messages captured in real-time provide users with up-to-date device data, including the device name, software name, MAC address, and online/offline status. The MongoDB database and heartbeat engine ensure the secure storage and management of all device data. With IoT Manager, users can easily manage and monitor large-scale IoT deployments, improving their overall efficiency and reducing the risk of device failure. Additionally, the platform's ability to analyze heart-beat data allows users to proactively identify and resolve potential issues, minimizing downtime and ensuring optimal performance. With the login system and user management system provided in the portal, the users could easily access their device information.

The flexibility to create custom roles allows management to provide different users with different responsibilities. Access groups allow the admins to have more control over the user privileges. Google user data will be collected to ensure the authenticity of the user. Data collected includes photos, phone numbers, and other information that's disclosed by the user. The user management module in the system allows the admin to have full control over the users in the system.

### 4.1 Testing Methods

Testing ensures that the system is error-free based on criteria that are anticipated by the user or by the organization. A system may have high-end or low-end performance based on the environment in which it operates.

#### 4.1.1 Functionality Testing

These tests check if the part of the system that is supposed to do something actually works well or not. The QA team makes some tests based on what needs to be checked, and then they test

them out. QA team tests the functionalities according to the requirements of the product.

- **Login Functionality:** This test verifies that users can log in using both Google and custom login methods. Also, test the forgot password and reset password functionality for custom login users. Reset functionality sends a mail to the registered email account. Using the link provided in the mail the user can reset their password. These need to be tested.
- **Role-Based Access Control:** This test verifies that users are provided access to the platform based on their assigned roles and privileges. It should also test that custom roles can be created and that super admins can create, view, and update users and custom roles.
- **Session Management:** This test verifies that users are automatically logged out of the system after 48 hours of inactivity. It should also test that users who try to access the system after being logged out are redirected to the login page.
- **Data Security:** This test verifies that only authorized users have access to the platform and that user data is secured. It also tests that the system prevents unauthorized access attempts and that user data is encrypted during storage.
- **Error Handling:** This test verifies that the system provides appropriate error messages for various scenarios, such as incorrect login credentials or attempting to access restricted features. It should also test that the system logs errors for debugging purposes.

### 4.1.2 Unit Testing

In this project, unit testing is an important aspect of validation testing, aimed at identifying defects in individual sections of the project. Unit testing is used to evaluate the functionality and performance of each system unit or component in isolation. This testing helps developers to identify and resolve issues early on, thereby minimizing the risk of encountering problems during deployment or integration. A successful validation test confirms that the system meets the requirements, while a failed test indicates that modifications are necessary. Unit testing is a critical step in the validation process, enabling developers to ensure that the system performs as expected and meets the needs of all stakeholders.

The portal is tested against multiple test cases during validation testing. Logging in to the system is one of the important part of the system. valid credentials needs to given in order to

access the system. Username is unique to all the users and it is provided by the admin. password is matched against a regex structure which must have 8 alphanumeric characters. Both the data is tested and validated using the data available in the database. After it is validated the user is given access based on the user role which also is validated from the database. For the google login an id token is generated during the sign-in which is then used for getting google user details. Then the generated data is used for validation with the database information.

In the user registration, which can only be accessed by admin is tested properly. User are added with name, username, email, password and other information which is also validated. This process also performed on the data update.

### 4.1.3 User Acceptance Testing

The success of any system depends largely on user acceptance. To ensure this, the system must undergo continuous evaluation and testing throughout the development phase. This entails maintaining open communication with potential end-users and incorporating their feedback into the system's design. In this project, a menu-driven system, input screen design, and output screen design were carefully considered and implemented to achieve user approval. User-centered design is the process of involving users in the design and development of the system, ensuring that their needs and preferences are met. This leads to higher levels of user satisfaction and system adoption.

Validate the user interface of IoT Manager to ensure it is user-friendly, visually appealing, and responsive across different devices and browsers. Verify that all UI elements are displayed correctly, and users can easily interact with them. UI is created based on the users that will be accessing the system. On each iterative development, users are involved and with their constant feedback and criticism the UI is developed and deployed.

Test Scenarios:

- Login to the IoT Manager platform with valid credentials. Navigate to the heart-beat data section of the platform. Verify that the heart-beat data for all my IoT devices is displayed correctly, including the device name, software name, MAC address, and last sync time converted to my local time zone.
- Select one of my devices and verify that I can view detailed information about the device, including its status (online/offline) and any alerts or notifications related to the device.

- Simulate a device going offline by disabling its network connection. Wait for the system to detect the device’s offline status and generate a notification.
- Verify that I receive a notification, either by email or through the platform, indicating that the device is offline. Reconnect the device and wait for the system to detect its online status.
- Verify that the system correctly updates the device’s status to online and generates a notification indicating that the device is back online.
- Log out of the system and verify that my session is ended and I am unable to access the platform without valid credentials.

## 4.2 Output Screens and Results

### 1. Landing Page

Device ID	Location	Site ID	Last Sync	VPN IP	Public IP	Mac Address	App Version	Firmware Version	Actions
idm-10000	Abu Dhabi	Site_7	14-04-2023 12:38:17	10.1	19.1	182	4.6	5.0	[Refresh] [Refresh] [Refresh] [Refresh]
idm-107429	Abu Dhabi	Site_7	14-04-2023 12:37:40	10.1	19.2	182	5.0	5.1.1	[Refresh] [Refresh] [Refresh] [Refresh]
idm-10713	Abu Dhabi	Site_19	09-09-2019 14:37:12	10.8	94.1	182	4.4.3	4.4.3	[Refresh] [Refresh] [Refresh] [Refresh]
idm-10716	Abu Dhabi	Site_21	14-04-2023 12:35:44	10.10	111	182	6.3.1	6.3	[Refresh] [Refresh] [Refresh] [Refresh]
idm-10766	Abu Dhabi	Site_291 (g4b)	27-09-2022 16:24:45	10.8	122	182	4.6.1	5.0	[Refresh] [Refresh] [Refresh] [Refresh]
idm-10767	Abu Dhabi	Site_6	18-09-2019 15:04:05	10.1	19.1	182	4.4.4	4.4.4	[Refresh] [Refresh] [Refresh] [Refresh]
idm-10711	Abu Dhabi	Site_2	29-07-2020 10:52:27	10.1	19.12	182	4.6.1	4.6.1	[Refresh] [Refresh] [Refresh] [Refresh]
idm-10765	Abu Dhabi	Site_2	05-10-2020 11:48:39	10	19.1	182	4.6.1	4.6.1	[Refresh] [Refresh] [Refresh] [Refresh]
idm-10712	Abu Dhabi	Site_2	14-04-2023 12:35:24	10.8	19	182	4.6	5.1.1	[Refresh] [Refresh] [Refresh] [Refresh]

Total Rows: 4929

Figure 4.1: Landing Page

The landing page differs for each user depending on the user’s role. This page gives adequate information to the user about their devices. The buttons available on the device details page give information about device status, device upgrade logs, device terminal page, and device audit logs.

## 2. User Details Page

Users /  
Users

User ID	Full Name	Email	Role	Access Group	Created On	Last Login	
<input type="text" value="Search User ID"/>	<input type="text" value="Search Full Name"/>	<input type="text" value="Search Email"/>	<input type="text" value="Select Role"/>	<input type="text" value="Select Access Grc"/>			
<a href="#">Admin</a>	Admin	@gmail...	Admin	Admin		17 Apr 2023 22:45	
<a href="#">head</a>	Head	@gma...	Head	Head			
<a href="#">_Super...</a>	Supervisor	@gmail...	Supervisor	Supervisor			
<a href="#">_Quality</a>	Quality	@gmail...	Quality	Quality			
<a href="#">Admin_SR23</a>	Admin_SR23	@g...	Super Admin	Super Admin	04 Aug 2022 12:54	11 Aug 2022 10:45	

Showing 9 records of 9

Figure 4.2: View Users Page

This page gives information about the users in the system. Using this page the admins can edit and delete users in the portal. Admins will not be able to edit the username, email, and name of the user. In the users viewing page create new and existing users can be created. The add existing option allows the admin to add new user roles to the existing user.

## 3. View Roles Page

User Roles /  
User Roles

Role	Description	Actions
<input type="text" value=""/>	<input type="text" value=""/>	
<a href="#">Admin</a>	Admin	
<a href="#">Plant Head</a>	Plant Head	
<a href="#">Supervisor</a>	Supervisor	
<a href="#">Super Admin</a>	Super Admin	
<a href="#">Vendor</a>	Vendor	
<a href="#">Customer</a>	Customer	

Rows: 8

Figure 4.3: View Roles Page

User roles can be created using create new option available in the system. On this page, the admin can view all the user roles that are created, also the admin can edit and delete

the user roles. By providing the description for each role, it will be easy to identify the user roles.

#### 4. View Access Groups Page

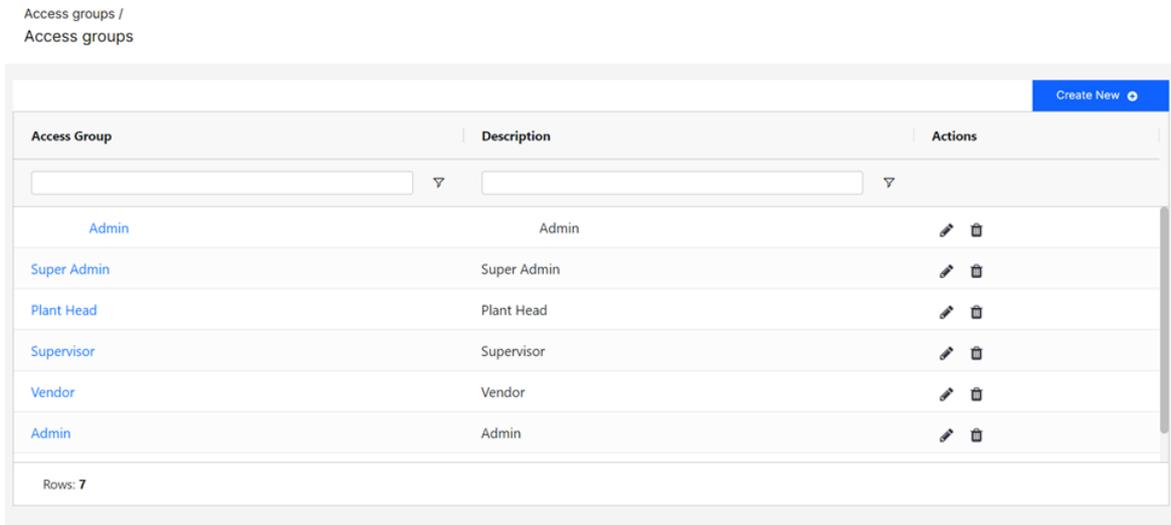


Figure 4.4: Access Group Viewing Page

Admin can easily use the access group page to create, update, and delete the access groups. Access groups provide more control over the user roles. The access groups act as an identifier for the user roles that are created.

# Chapter 5

## Conclusion

The goal of the portal is to provide an efficient and easier way to access the user's device information and insights into the data provided by each device. **IOT MANAGER - LOGIN AND PERSONA MANAGEMENT** web application is designed to handle a large number of devices and users. The architecture of the application is scalable, with the ability to handle increasing traffic and data without affecting performance. Using newer and more advanced technologies such as FastAPI, redis, Mongo and Python allows the system to be faster and more reliable. Security is a critical issue when dealing with IoT devices, and the web application is designed with robust security features to protect against potential cyber-attacks. This includes authentication, authorization, encryption, and secure communication protocols. The user interface of the web application is user-friendly, intuitive, and easy to use. It provides users with the ability to monitor, control, and configure devices quickly and easily. The portal provides real-time monitoring capabilities to detect and respond to any anomalies or issues with the devices. This includes models for real-time data analysis and alerting mechanisms.

### 5.1 Future Enhancement

In the future, the system can be incorporated with more advanced analytics capabilities to gain deeper insights from device data, including predictive analytics, anomaly detection, and machine learning algorithms. This can enable proactive device management and predictive maintenance, optimizing device performance and minimizing downtime. Developing a mobile application for the portal to provide users with on-the-go access to device information, alerts, and management features, enabling remote monitoring and management of devices. By integrating voice assistants and NLP technologies into the platform, users can interact with IoT devices using natural language commands, making device management more intuitive and user-friendly. Adding more robust device management features, such as remote device configuration, firmware rollback, and device grouping for easier management of

multiple devices with similar configurations. By Continuously improving the scalability and performance of the portal to handle increasing volumes of devices, data, and users, ensuring smooth and efficient operation even as the number of devices and users grows.

# References

- [1] Souri, Alireza, "A systematic review of IoT communication strategies for an efficient smart environment." *Transactions on Emerging Telecommunications Technologies* 33.3 (2022): e3736. DOI: 10.1002/ett.3736
- [2] Li, Xiaofen, "Intelligent Deployment and Development Strategy of Agricultural Farmland Based on Improved Architecture of Internet of Things." *Mathematical Problems in Engineering* 2022 (2022). DOI: 10.1155/2022/5213535
- [3] Diène, Bassirou, "Data management techniques for Internet of Things." *Mechanical Systems and Signal Processing* 138 (2020): 106564. DOI: 10.1016/j.ymssp.2019.106564
- [4] Geetha G, "Interpretation and Analysis of Angular Framework." *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*. IEEE, 2022. DOI: 10.1109/ICPECTS56089.2022.10047474
- [5] Jeng, Shyr-Long, and Wei-Hua Chieng, "Web-based HMI of industrial controllers for general purpose." *2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*. IEEE, 2020. DOI: 10.1109/ICKII50300.2020.9318905
- [6] Pinto, Carlos Manso, and Carlos Coutinho, "From native to cross-platform hybrid development." *2018 international conference on intelligent systems (IS)*. IEEE, 2018. DOI: 10.1109/IS.2018.8710545
- [7] Lathkar, Malhar, "Getting Started with FastAPI." *High-Performance Web Apps with FastAPI: The Asynchronous Web Framework Based on Modern Python*. Berkeley, CA: Apress, 2023. 29-64. DOI: 10.1007/978-1-4842-9178-8\_2
- [8] Gupta, Eeshan, "Enabling Attribute-based Access Control in NoSQL Databases." *IEEE Transactions on Emerging Topics in Computing* (2022). DOI: 10.1109/TETC.2022.3193577
- [9] Eyada, Mahmoud Moustafa, "Performance evaluation of IoT data management using MongoDB versus MySQL databases in different cloud environments." *IEEE access* 8 (2020): 110656-110668. DOI: 10.1109/ACCESS.2020.3002164

- 
- [10] Sacco, Andres, "Redis: Key/Value Database." *Beginning Spring Data: Data Access and Persistence for Spring Framework 6 and Boot 3*. Berkeley, CA: Apress, 2022. 189-212. DOI: 10.1007/978-1-4842-8764-4\_7
- [11] Chopade, Rupali, and Vinod Pachghare, "A data recovery technique for Redis using internal dictionary structure." *Forensic Science International: Digital Investigation* 38 (2021): 301218. DOI: 10.1016/j.fsidi.2021.301218
- [12] Sharma, Akshansh, "Python: the programming language of future." *International Journal of Innovative Research in Technology* 6.12 (2020): 115-118.
- [13] Haji, Saad Hikmat, and Amira B. Sallow, "IoT for smart environment monitoring based on Python: a review." *Asian Journal of Research in Computer Science* 9.1 (2021): 57-70.
- [14] Makris, Antonios, "MongoDB Vs PostgreSQL: A comparative study on performance aspects." *GeoInformatica* 25 (2021): 243-268. DOI: 10.1007/s10707-020-00407-w
- [15] Sanderson, Lacey-Anne, "A PostgreSQL Tripartite solution for large-scale genotypic and phenotypic data." *Database* 2021 (2021)

# Appendix

## Screenshots

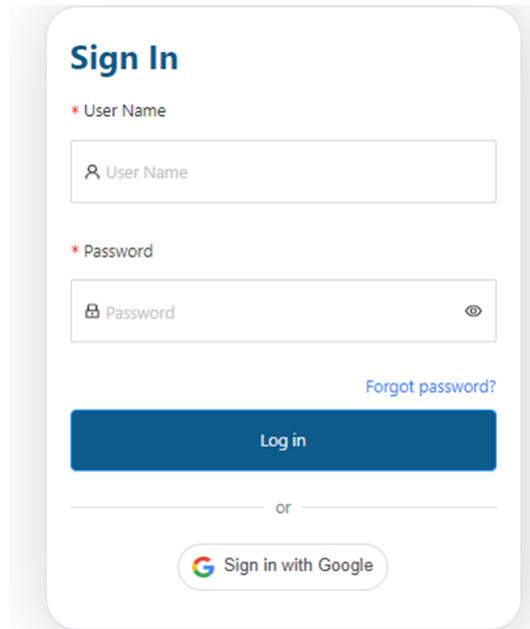


Figure A.1: Login Page

Online Devices : 0		Offline Devices : 4929		Discontinued Devices : NA					
Device ID	Location	Site ID	Last Sync	VPN IP	Public IP	Mac Address	App Version	Firmware Version	Actions
device-10001	XXXXXXXXXXXXXXXXXXXX	000_7	14-04-2023 12:08:17	10.1	15.1	08:2	4.6	5.0	
device-10002	XXXXXXXXXXXXXXXXXXXX	000_7	14-04-2023 12:07:40	10.1	15.2	08:2	5.0	5.1.1	
device-10003	XXXXXXXXXXXXXXXXXXXX	000_19	09-09-2019 14:37:12	10.8	04.1	08:2	4.4.1	4.4.1	
device-10004	XXXXXXXXXXXXXXXXXXXX	000_20	14-04-2023 12:05:44	10.10	111	08:2	5.1	5.1	
device-10005	XXXXXXXXXXXXXXXXXXXX	000_201(005)	27-09-2022 16:24:45	10.8	122	08:1	4.6.1	5.0	
device-10006	XXXXXXXXXXXXXXXXXXXX	000_1	18-09-2019 15:04:05	10.1	15.1	08:2	4.4.1	4.4.1	
device-10007	XXXXXXXXXXXXXXXXXXXX	000_2	21-07-2020 19:02:07	10.1	15.12	08:2	4.6.1	4.6.1	
device-10008	XXXXXXXXXXXXXXXXXXXX	000_2	05-10-2020 11:40:09	10	15.1	08:1	4.6.1	4.6.1	
device-10009	XXXXXXXXXXXXXXXXXXXX	000_2	14-04-2023 12:05:24	10.8	15	08:2	4.6	5.1.1	

Total Rows: 4929

Figure A.2: Landing Page

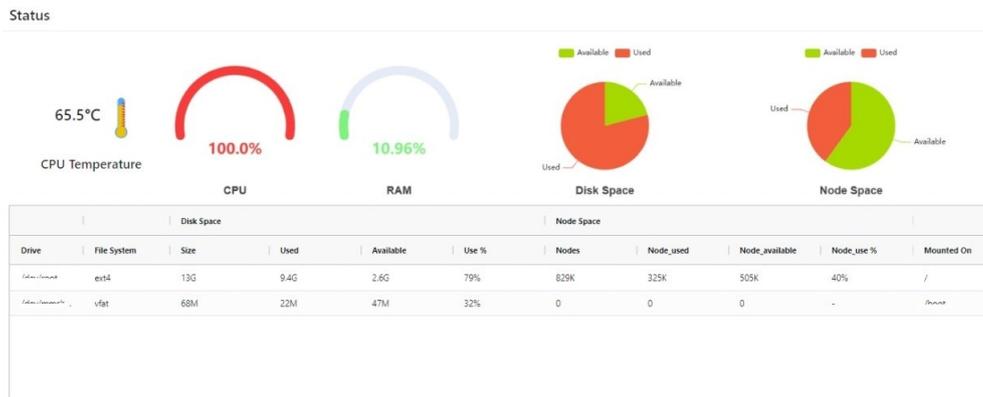


Figure A.3: Device Analytics Page

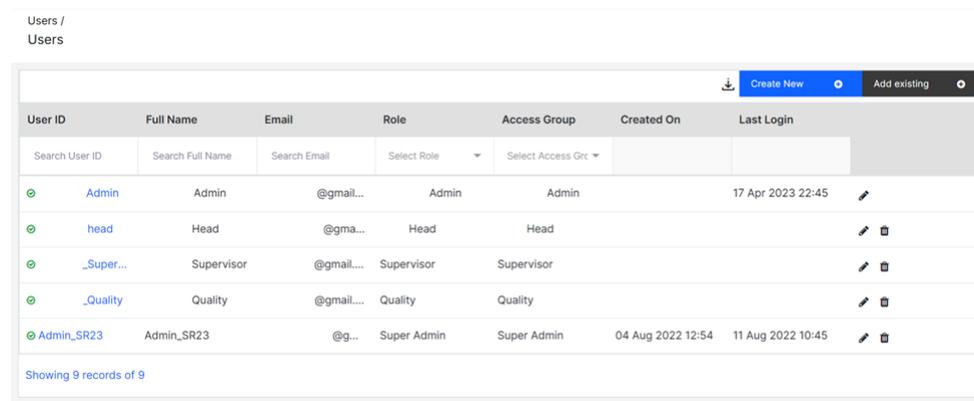


Figure A.4: View Users Page

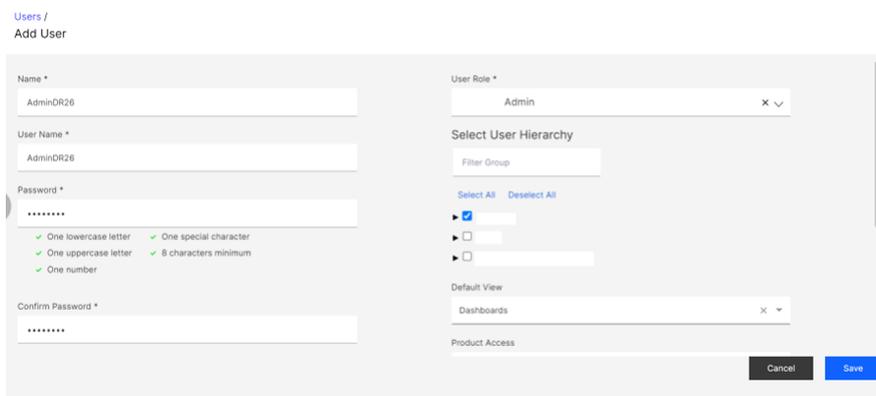


Figure A.5: Create Users page

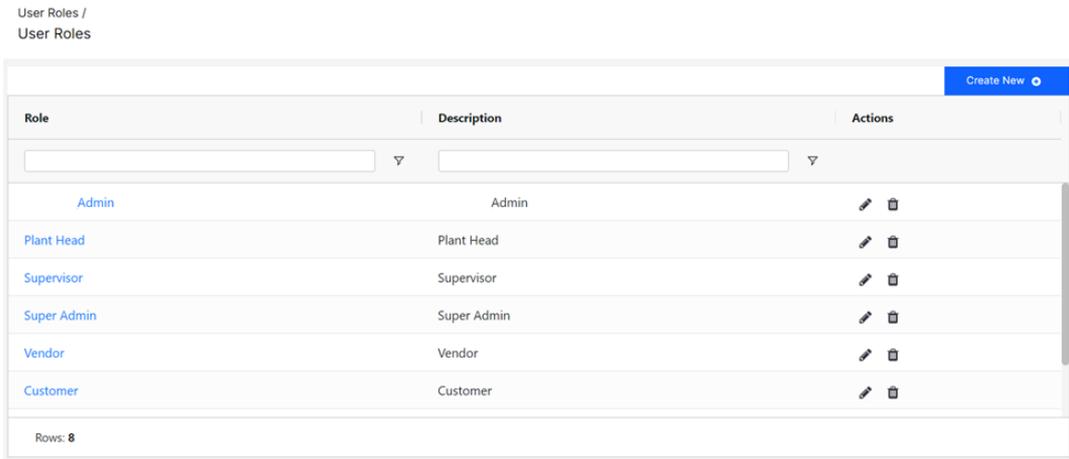


Figure A.6: View User Roles Page

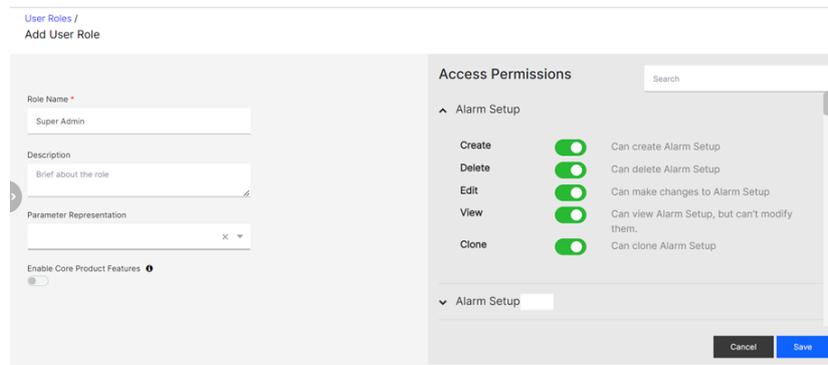


Figure A.7: Create User Roles Page

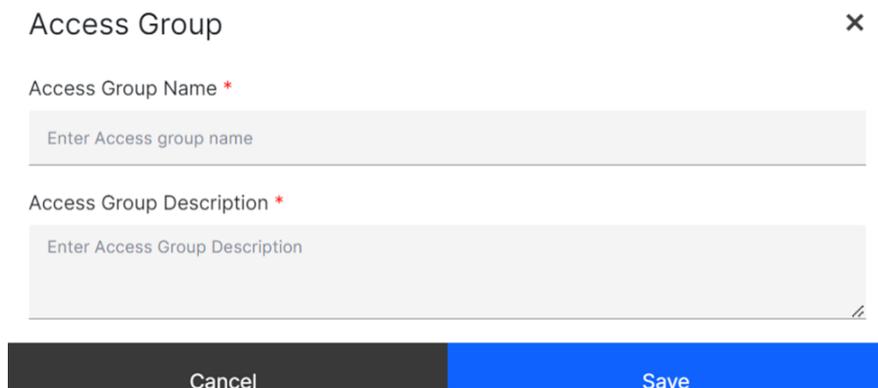


Figure A.8: Create Access Groups Page

Access groups /  
Access groups

Create New

Access Group	Description	Actions
<a href="#">Admin</a>	Admin	
<a href="#">Super Admin</a>	Super Admin	
<a href="#">Plant Head</a>	Plant Head	
<a href="#">Supervisor</a>	Supervisor	
<a href="#">Vendor</a>	Vendor	
<a href="#">Admin</a>	Admin	

Rows: 7

Figure A.9: View Access Groups Page