CONSULTANCY PROJECT REPORT

IOT BASED REMOTE MONITORING AND MOBILE APP DEVELOPMENT FOR BATTERY MANAGEMENT SYSTEM

Submitted by

P PRAVEENKUMAR (2020505020)

Guided by

DR. N. PAPPA (Professor , Dept. of IE)

DR. S. SUTHA (Professor, Dept. of IE)

DR. S. MEYAPPAN (Asst. Professor, Dept. of IE)

Sponsored by

AMPTON ENERGY

CHENNAI

MADRAS INSTITUTE OF TECHNOLOG
CAMPUS ANNA UNIVERSITY:
CHENNAI-600 044

BONAFIDE CERTIFICATE

Certified that the consultancy project report titled "**IOT BASED REMOTE MONITORING AND APP DEVELOPMENT FOR BATTERY MANAGEMENT SYSTEM"** is the bonafide work of

PRAVEENKUMAR P **2020505020**

who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here does not form part of any project on the basis of which degree or award was conferred on an earlier occasion on this or anyother candidate.

SIGNATURE

Dr. N. PAPPA

PROJECT HEAD SUPERVISOR

Professor,

Dept. of Instrumentation Engg.

MIT Campus, Anna University

Chromepet, Chennai – 600044

SIGNATURE

Dr. S. MEYYAPAN

SUPERVISOR

Assistant Professor,

Dept. of Instrumentation Engg

MIT Campus, Anna University

Chromepet, Chennai - 600044

ACKNOWLEDGEMENT

We would like to express our special gratitude to“**Ampton Energy, Chennai”** for giving us a wonderful opportunity to implement our project in real.

A special mention to our Dean “**Dr.J.Prakash”** for supporting us throughout the project.

Our thanks and appreciation also go to our Head of the Department “**Dr.S. Srinivasan”** for encouraging us and giving some brain-storming ideas to reach our end prototype.

Our sincere thanks and gratitude to **Ms.Kanchan Saxena** of **Ampton Energy** for all the help and support provided to proceed with this project.

We highly are indebted to our mentors “**Dr.N.Pappa, Dr.S.Sutha, Dr.S.Meyyappan ”** for their guidance and supervision throughout the project. We specially thank them for giving us such great ideas , information regarding the project and the development ideas of our project which constantly encouraged us to gain knowledge in the field of Application Development.

P PRAVEENKUMAR

(2020505020)

ABSTRACT

The objective of this report pertains to the development of a mobile application tailored for Android users, aimed at facilitating the comprehensive monitoring of various parameters associated with batteries procured from Ampton Energies.

The company envisions the creation of a Battery Management System (BMS) capable of real-time monitoring and acquisition of crucial battery metrics, including State of Charge (SOC), Capacity, Temperature (both maximum and minimum), Current, Voltage, among others. Upon data acquisition from the batteries, these parameters will be promptly transmitted to cloud storage for seamless accessibility.

Subsequently, the amassed data will serve as a foundational resource for diverse applications. Within the scope of this project, particular emphasis will be placed on the development and implementation of the Ampton BMS application. This application is envisioned to offer users comprehensive insights into the current status of their Ampton batteries, ensuring convenient and sophisticated monitoring capabilities without necessitating physical proximity to the battery unit.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
1	**INTRODUCTION**	1
	1.1 OVERVIEW	1
	1.2 OBJECTIVE	2
2	**SELECTION AND DESIGN**	4
	2.1 SELECTION	4
	2.1.1 DATABSE SELECTION	4
	2.1.2 SOFTWARE SELECTION FOR DESIGN	5
	2.1.3 SOFTWARE SELECTION FOR DEVELOPMENT	6
	2.1.4 SELECTION OF FRAMEWORK	7
	2.2 DESIGN	9
3	**DEVELOPMENT AND RESULTS**	11
	3.1 SOURCE CODE EXPLANATION	11
	3.2 COMPONENTS USED IN THE APPLICATION	12
	3.2.1 SCREENS FOLDER	12
	3.2.2 STORE FOLDER	13
	3.2.3 COMPONENTS FOLDER	13
	3.3 RESULTS	14
4	**USER MANUAL**	18

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

A Battery Management System (BMS) is like a smart guardian for batteries. It keeps an eye on various important things like how much charge the battery has left, its temperature, how much power it can hold, and more. Imagine it as a caretaker that constantly checks if the battery is doing okay and if it needs any help.

Now, why do we need a BMS? Well, batteries are like the heart of many devices, from smartphones to electric cars. Just like our heart needs monitoring to keep us healthy, batteries need monitoring to keep devices running smoothly. Without a BMS, batteries could get damaged or even cause accidents like overheating or catching fire. So, having a BMS ensures that batteries stay safe, last longer, and perform their best, which is super important for all the gadgets and machines we rely on every day.

A BMS mobile application adds another layer of convenience and accessibility to the monitoring and management of batteries. Here’s how:

Remote Monitoring: With a BMS mobile app, users can keep tabs on their batteries from anywhere, anytime. Whether they're at home, work, or on the go, they can quickly check the battery's status without needing to be physically present near the battery.

Real-time Updates: The app provides real-time updates on important parameters such as battery charge level, temperature, and health. Users can receive alerts and notifications if any issues arise, allowing for prompt action to be taken to prevent potential problems.

Historical Data Analysis: The app often stores historical data about the battery's performance over time. This data can be valuable for identifying patterns, detecting trends, and making informed decisions about battery usage and maintenance.

Integration with Cloud Services: Many BMS mobile apps integrate with cloud storage services, allowing data to be securely stored and accessed from multiple devices. This ensures data reliability and availability, even if the user switches devices or loses their phone.

Enhanced Safety and Efficiency: By providing users with detailed insights into their battery's condition, the app helps ensure safe operation and optimal performance. It empowers users to take proactive measures to prolong battery life, prevent damage, and maximize efficiency.

1.2 OBJECTIVE:

The project entails the selection of various parameters to be monitored within the Battery Management System (BMS) for display in a mobile application. The selected parameters include:

- SOC
- Maximum Temperature
- Minimum Temperature
- Voltage
- Capacity
- Current

Additionally, the project involves the planning of three distinct navigation pages within the mobile application::

- BMS MONITOR – This serves as the primary interface for users.

- CELL STATUS – This page showcases the aforementioned parameters for each cell of the specified battery.
- PROFILE – A dedicated section containing user information.

Furthermore, the project necessitates the identification of appropriate software for the design and development of the mobile application. Following the software selection, the mobile application will be designed, with modifications incorporated based on feedback from the company. Subsequently, the development phase will commence, focusing on the realization of the application's functionalities as outlined, followed by rigorous testing to ensure optimal performance.

In essence, the overarching objective of the project is to create a mobile application capable of measuring and displaying the specified parameters while prioritizing user experience through intuitive interface design and robust functionality.

CHAPTER 2

SELECTION AND DESIGN

2.1 SELECTION

The Selection involves in three different areas first in Design, second in the development and the third in storage.

2.1.1 DATABASE SELECTION

The Database Selected for this application is **Google's Firebase**. Firebase is a versatile platform developed by Google that offers a range of services for both IoT (Internet of Things) and mobile application development. When combined, Firebase can significantly enhance the capabilities and efficiency of IoT and mobile applications.

In the realm of IoT, Firebase provides several features that are instrumental in managing and processing data from connected devices:

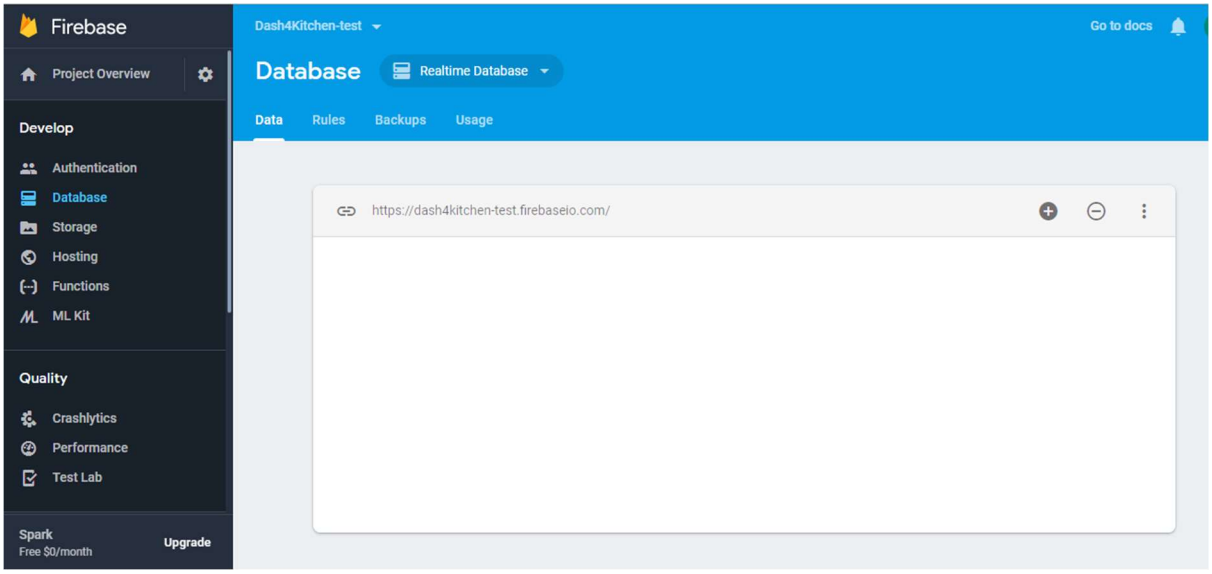


Figure: 2.1

Real-time Database: Firebase offers a NoSQL cloud database that allows for the storage and synchronization of data in real-time. This feature is particularly

beneficial for IoT applications where data from sensors and devices need to be constantly updated and accessed by multiple users or devices simultaneously.

Authentication: Securely managing user authentication is crucial in IoT applications to ensure that only authorized users can access and control connected devices. Firebase Authentication provides easy-to-use authentication methods, including email/password, social media logins, and custom authentication.

Analytics and Monitoring: Firebase Analytics provides insights into user engagement, retention, and user behaviour within IoT applications. These analytics can help developers optimize their applications, improve user experience, and identify opportunities for enhancing IoT device functionality.

2.1.2 SOFTWARE SELECTION FOR DESIGN:

Figma has emerged as a powerhouse tool in the realm of designing mobile applications, offering a seamless and collaborative environment for designers and developers alike. Its role in mobile app design spans across various stages of the design process, from ideation to prototyping and collaboration.

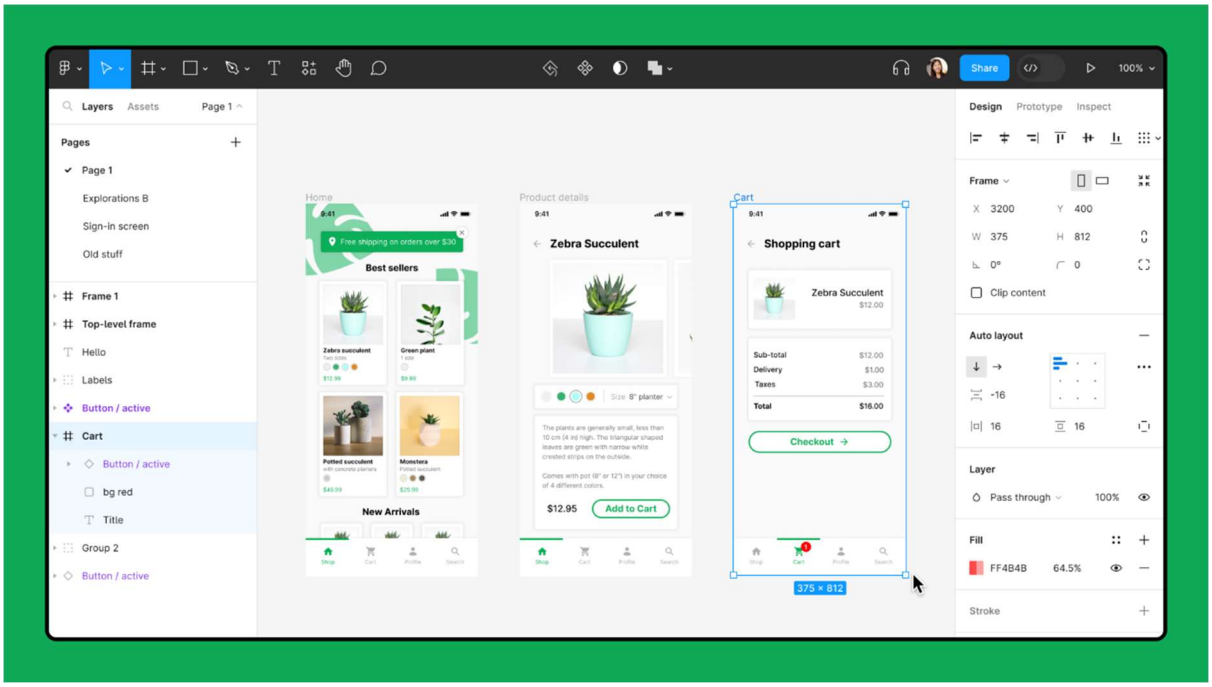


Figure: 2.2

User Interface Design: Figma provides an intuitive interface that allows designers to create visually appealing UI designs for mobile applications. With its extensive library of design elements, including icons, buttons, and components, designers can quickly mock up the user interface of their mobile app with ease. The ability to customize and reuse components streamlines the design process and ensures consistency across different screens and states of the app.

Responsive Design: Designing for mobile requires attention to responsive design principles to ensure that the app looks and functions well across different screen sizes and orientations. Figma simplifies this process by allowing designers to create responsive layouts and preview how their designs adapt to various screen sizes in real-time. This ensures that the mobile app maintains its usability and aesthetic appeal regardless of the device it is viewed on.

Prototyping: Figma's prototyping capabilities enable designers to create interactive prototypes of their mobile apps without writing a single line of code. Designers can define transitions, gestures, and animations to simulate the user experience and test the flow of the app. This allows for quick iteration and validation of design ideas, leading to a more refined and user-friendly mobile app.

Collaboration: One of Figma's standout features is its robust collaboration capabilities, which make it ideal for team-based mobile app design projects. Designers can invite stakeholders, developers, and other team members to view and edit designs in real-time, facilitating seamless collaboration and communication. Comments, annotations, and version history help keep track of feedback and changes, ensuring that everyone is on the same page throughout the design process.

2.1.3 SOFTWARE SELECTION FOR DEVELOPMENT:

Visual Studio Code (VS Code) has become an integral part of the development workflow for mobile applications, offering a versatile and efficient environment for coding, debugging, and collaboration. With its extensive ecosystem of extensions and plugins, developers can customize their workflow to suit the specific requirements of mobile app development. From writing code in various programming languages like JavaScript, Swift, Kotlin, or Dart to debugging and testing mobile app functionalities, VS Code provides a seamless experience. Its integration with version control systems like Git enables smooth collaboration among team members, while features like IntelliSense and code snippets boost productivity by providing context-aware suggestions and shortcuts. Overall, VS Code plays a crucial role in streamlining the development process of mobile applications.

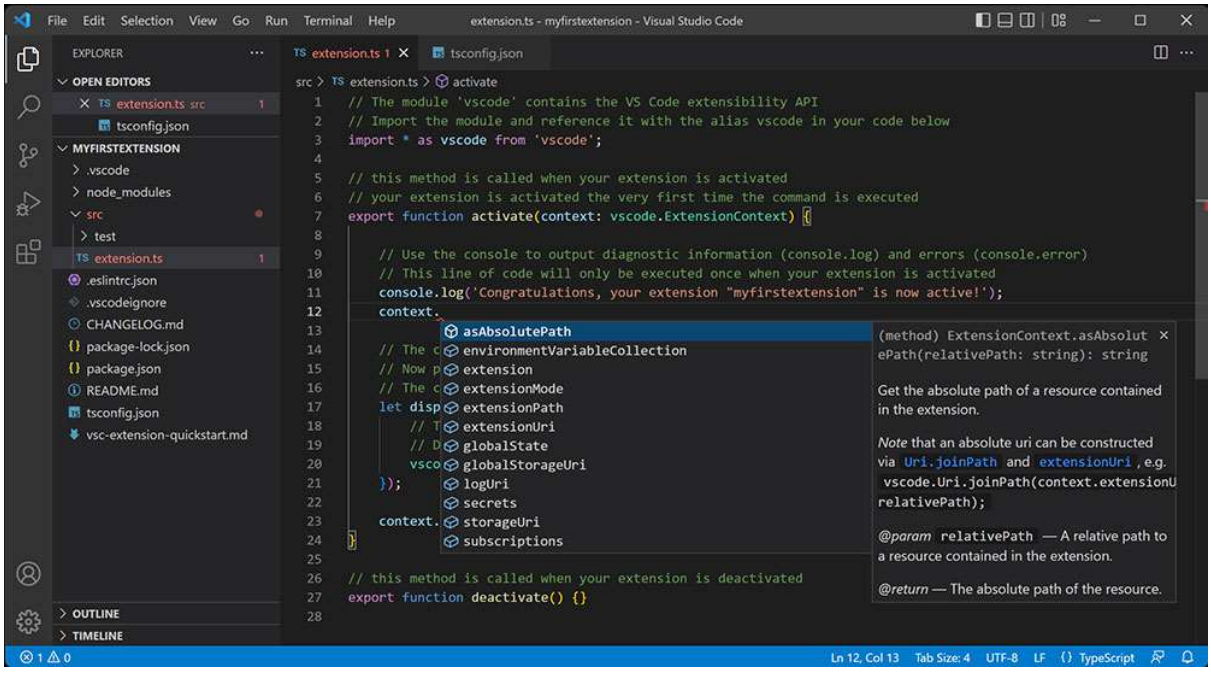


Figure: 2.3

2.1.4 SELECTION OF FRAMEWORK:

React Native has revolutionized the landscape of mobile application development by offering a powerful framework that enables developers to build high-performance, cross-platform apps using familiar web development

technologies. Its role in the development of mobile applications is multifaceted and significant:

Cross-Platform Development: One of the key advantages of React Native is its ability to write code once and deploy it across multiple platforms, including iOS and Android. This significantly reduces development time and costs as developers can leverage a single codebase to target a broader audience.

Native-Like Performance: React Native bridges the gap between native and web app development by allowing developers to build apps with native-like performance. By using native components and APIs, React Native apps can deliver smooth animations, fluid user interactions, and fast load times, providing users with an experience that rivals that of native apps.

Reusable Components: React Native promotes a component-based architecture, where UI elements are encapsulated into reusable components. This not only improves code maintainability but also accelerates the development process as developers can reuse components across different parts of the app or even in multiple projects.

Hot Reloading: React Native's hot reloading feature allows developers to see the results of code changes in real-time without restarting the app. This rapid feedback loop speeds up the iteration process.

Large Community and Ecosystem: React Native benefits from a vibrant and active community of developers and contributors, as well as a vast ecosystem of third-party libraries, tools, and resources. This community-driven support ensures that developers have access to a wealth of knowledge, solutions, and updates.

2.2 DESIGN:

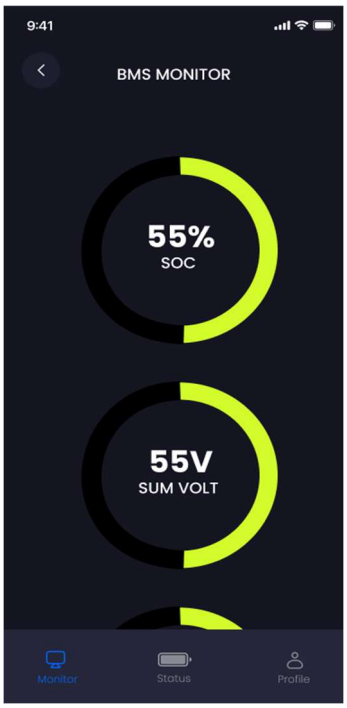


Figure: 2.4

BMS Monitor: This screen acts as the primary interface for the user. The Figure represents the BMS Monitor Screen.

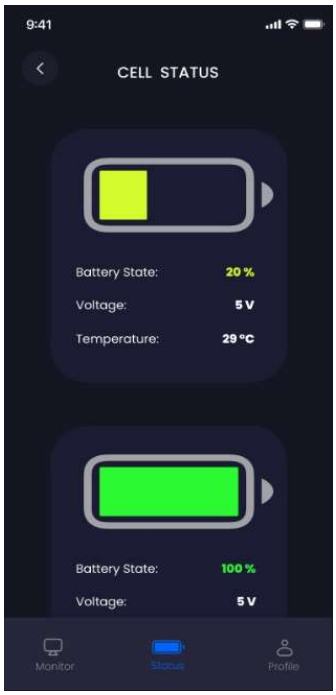


Figure: 2.5

Cell Status: The Figure showcases the aforementioned parameters in the objective for each cell of the specified battery.

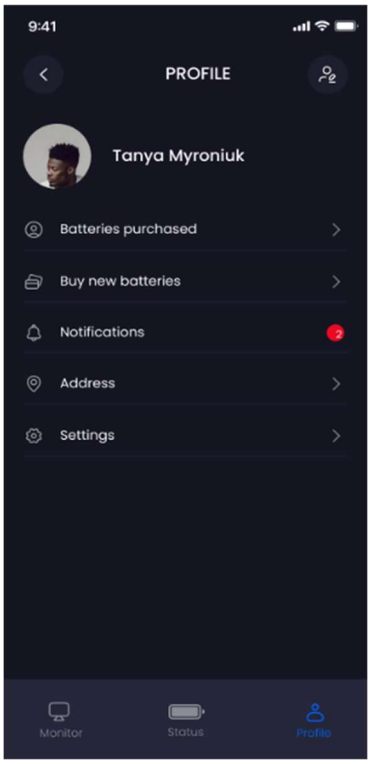


Figure: 2.6

Profile: The Figure shows a dedicated section which contains user information.

CHAPTER 3

DEVELOPMENT AND RESULTS

3.1 SOURCE CODE EXPLANATION

For the parameters listing, cell individual status and the profile page the dependencies need to be included are

@react-native-firebase/app is a crucial package within the React Native Firebase library. It serves as the entry point for integrating Firebase services into React Native applications. By including this package, you can easily access various Firebase services such as authentication, database, storage, and analytics within their React Native projects. This package handles the initialization and configuration of Firebase for the app, making it simple to set up and use Firebase services without dealing with complex setup procedures.

@react-native-firebase/database is a package that provides real-time database functionality for React Native applications using Firebase. It allows to store and sync data in real-time between clients and servers. With this package, you can create, read, update, and delete data in the Firebase Realtime Database directly from their React Native code.

@react-navigation/bottom-tabs is a package that enables the creation of bottom tab navigation in React Native applications using the React Navigation library. It allows developers to implement a common navigation pattern seen in mobile apps, where primary navigation options are displayed at the bottom of the screen as tabs.

expo is a comprehensive platform for building, deploying, and managing cross-platform mobile applications using React Native and other web technologies. It

provides a set of tools, libraries, and services that streamline the development process, enabling developers to create high-quality mobile apps with ease.

react this package includes all the necessary packages that should be include to run the react based application.

react-native is a framework package that enables developers to build mobile applications using Javascript and React. It allows for the development of cross-platform apps that can run on both iOS and Android devices.

react-native-circular-progress-indicator is package used to display circular progress indication in React native application. The circular progress indicator is implemented in BMS Monitor Screen.

axios is a package that is used to make HTTP requests, in the context of our application it is used for the sign in and log in functionality.

3.2 COMPONETS USED IN THE APPLICATION

App.js is the main entry folder where it contains the access to the navigation screens, login and sign up screens.

3.2.1 SCREENS FOLDER

BMSMonitorScreen.js: this file is loaded when the user login to the application, this file has the four parameters data(SOC, Capacity, Sum Volt, Current) in the array called arr thus the array values are looped inside a map function which returns a component CircularProgress.

BatteryStatus.js: this file is loaded when the user navigates to the Cell Status screen, this file contains individual cell parameters stored in an array arr and this is again rendered to the screen by traversing through the array arr.

ProfileScreen.js: when the user navigates to the profile screen this file will be loaded this screen contains the user information that can be shared with the company.

LoginScreen.js: This file will be loaded when the user first opens the app after downloading, then the user should provide its login credentials to enter into the application

SignupScreen.js: This file will be loaded when the user needs to register to his mobile application.

3.2.2 STORE FOLDER

auth-content.js: the file is responsible for authentication thus it stores whether a particular user is authenticated or not, it provides login and logout functionality, thus it determines whether the BMS application should be loaded or not with respect to the credibility of user credential.

data.js: this file is responsible to query and retrieve data form the firebase cloud storage and decode the data into user readable data and send to the components in another folder to display it.

3.2.3 COMPONENTS FOLDER

These folder contains two sub folders that contains child component files for the above mentioned parent component files:

Auth folder contains **Input.js** that collects the user credentials from the text box for authentication and send it to the **AuthContent.js** where the credentials are checked whether it is valid for login or registers the user if the user is new to the application.

The ui folder consists of the child components **Button.js, FlatButton.js, IconButton.js, LoadingOverlay.js** that as their name describes performs the function of Button and Loading overlay.

3.3 RESULTS

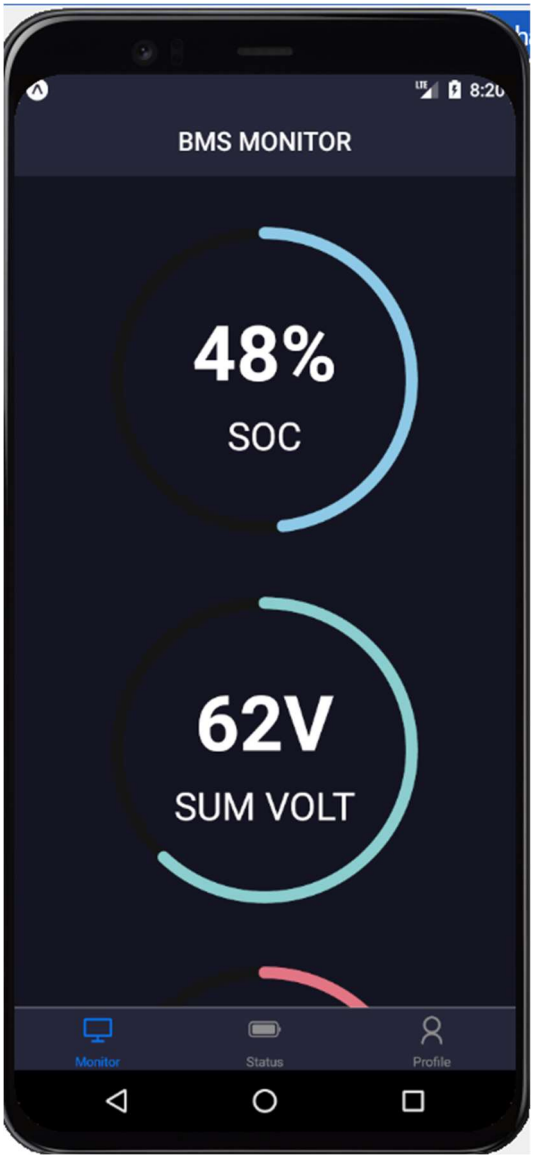


Figure: 3.1

The **Figure 3.1** shows the Battery monitor screen which displays the parameters that has been listed in the objective.

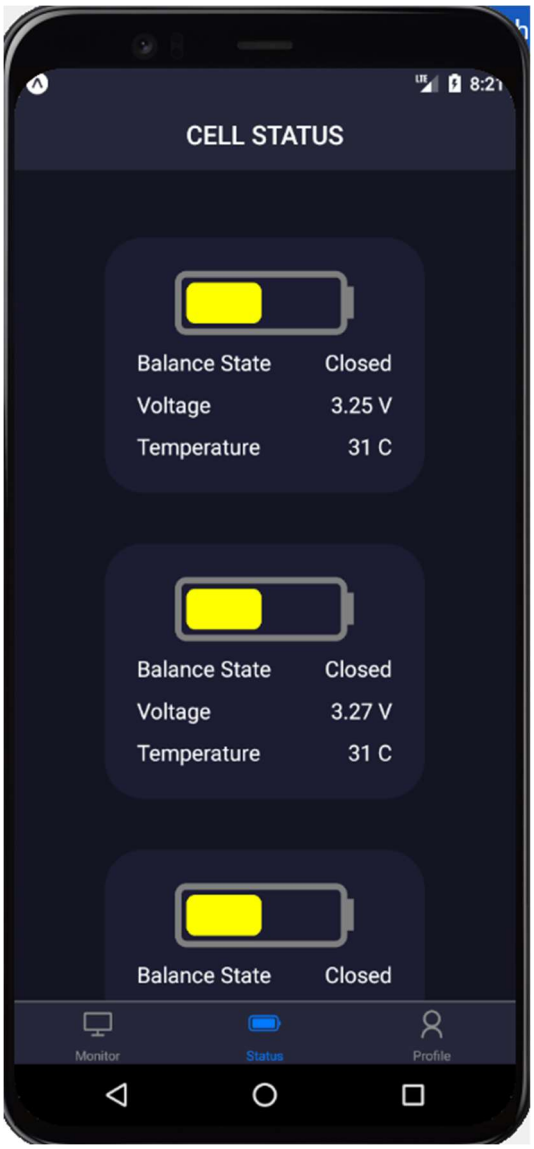


Figure: 3.2

The **Figure 3.2** shows the individual cell status of the given battery and its parameters are show as given.

Additionally the app has a login and sign up functionality which allows the user of the app to login by company given credentials and sign up functionality for new users.

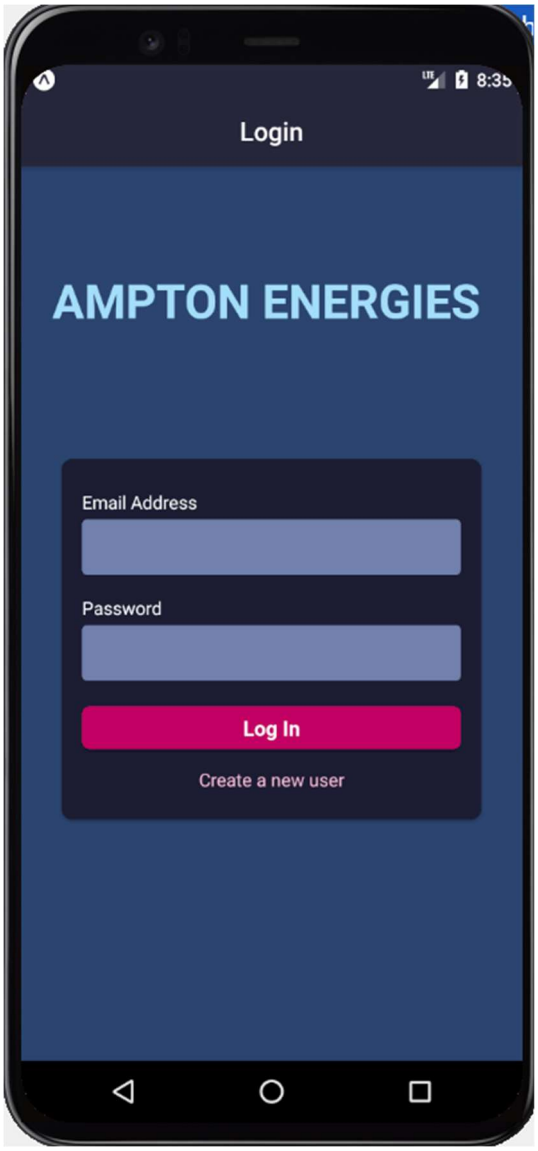


Figure: 3.3

The **Figure 3.3** shows the login functionality that allows user to login into the app using his company given credentials the login page has error check functionality so that it helps the users not to enter wrong credentials leading to crashing the app.

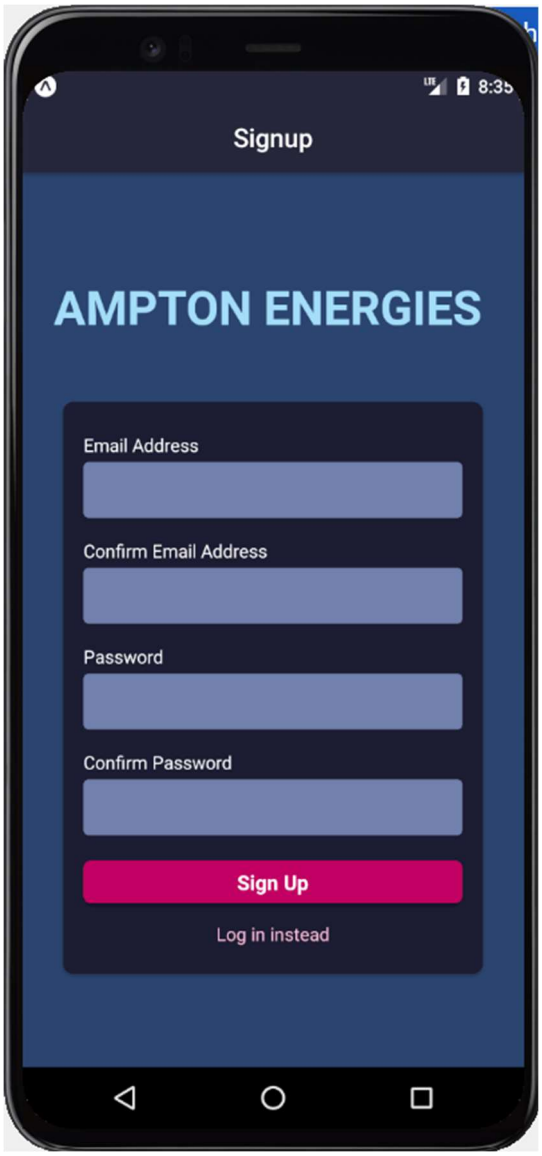
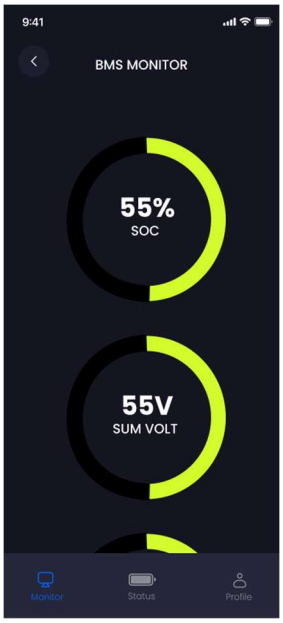


Figure: 3.4

The **Figure** shows the sign up functionality for the new users this page has also has the same error check methods so that the user can conveniently login and signup into the app.

CHAPTER 4

USER MANUAL



This is the first screen that appears when you open the BMS MOBILE APP.

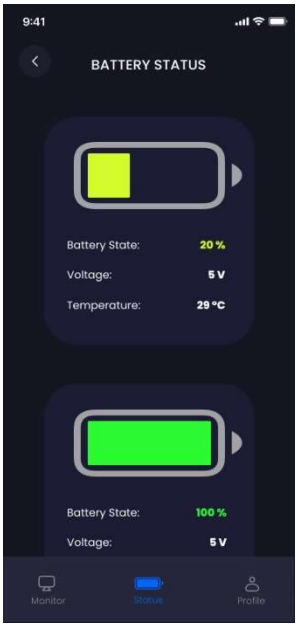
The Screen appears is the **BMS Monitor Screen**.

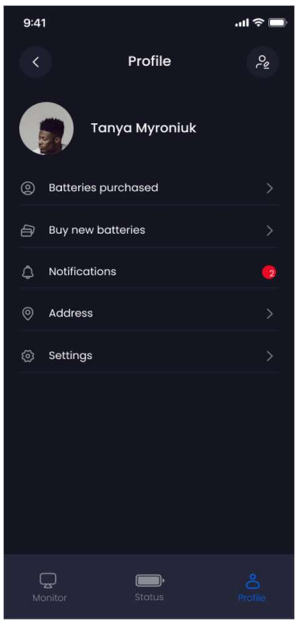
The User can scroll down this screen to monitor the parameters of the current battery which is set.

The Navigation bar routes the page to **Battery Monitor**, **Battery Status** and **User Profile**.

The User navigates to the **Battery Status** page by using the **Status** route in the navigation bar below.

This Screen gives the abstract status all batteries purchased by an individual User.





The User navigates to the **Profile** page using the **Profile** route in the navigation bar below.

This page shows the personal details of the User.

The User can edit their personal information through the edit option.