VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Machhe, Belagavi, Karnataka 590018

Project Report
on

SOLAR PANEL REPORTING AND MONITORING

Submitted in partial fulfillment of the requirement
for the award of the degree of

Bachelor of Engineering
in
Information Science & Engineering
by

Abhishek Sahu (1BG15IS001)
Rohit Kamal (1BG15IS040)
Sarthak Sureka (1BG15IS044)
Md. Aasim Ahmad (1BG15IS065)

Under the Guidance of
Mrs. Prema Jain
Assistant Professor
Dept. of ISE, BNMIT

Vidyaya Amrutham Ashnuthe

B.N.M. Institute of Technology

(Approved by AICTE, Affiliated to VTU, ISO 9001:2008 Certified
and Accredited as grade A Institution by NAAC)
All UG branches – CSE, ECE, EEE, ISE & Mech.E accredited by NBA for academic years 2018-19 to 2020-21 & valid upto 30.06.2021
Post box no. 7087, 27th cross, 12th Main,
Banashankari II Stage, Bengaluru- 560070, INDIA
Ph: 91-80- 26711780/81/82 Email: principal@bnmit.in, www. bnmit.org

Department of Information Science and Engineering

2018 – 2019

B.N.M. Institute of Technology

(Approved by AICTE, Affiliated to VTU, ISO 9001:2008 Certified
and Accredited as grade A Institution by NAAC)
All UG branches – CSE, ECE, EEE, ISE & Mech.E accredited by NBA for academic years 2018-19 to 2020-21 & valid upto 30.06.2021

Post box no. 7087, 27th cross, 12th Main,
Banashankari II Stage, Bengaluru- 560070, INDIA
Ph: 91-80- 26711780/81/82 Email: principal@bnmit.in, www. bnmit.org

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Vidyaya Amrutham Ashnuthe

CERTIFICATE

Certified that the project work entitled **Solar Panel Reporting and Monitoring** carried out by Mr. **Abhishek Sahu** (**1BG15IS001**), Mr. **Rohit Kamal** (**1BG15IS040**), Mr. **Sarthak Sureka** (**1BG15IS044**) and Mr. **Md. Aasim Ahmad** (**1BG15IS065**) the bonafide students of **B.N.M Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Information Science & Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2018-2019. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Mrs. Prema Jain	**Dr. Girisha G.S.**	**Dr. Krishnamurthy G N**
Assistant Professor	**Professor & Head**	**Principal**
Dept. of ISE, BNMIT	**Dept. of ISE, BNMIT**	**BNMIT**

Name of the Examiners **Signature with date**

1.

2.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

We would like to thank **Shri. Narayan Rao R. Maanay,** Secretary, BNMIT, Bengaluru for providing excellent academic environment in the college.

We would like to sincerely thank **Prof. T. J. Rama Murthy**, Director, BNMIT, Bengaluru for having extended his support and encouraging us during the course of the work.

We would like to express our gratitude to **Prof. Eishwar N Maanay**, Dean, BNMIT, Bengaluru for his relentless support, guidance and assistance.

We would like to thank **Dr. Krishnamurthy G.N**, Principal, BNMIT, Bengaluru for his constant encouragement.

We would like to thank **Dr. Girisha G.S.**, Professor and Head of the Department of Information Science and Engineering who has shared her opinions and thoughts which helped us in the completion of our project successfully.

We consider it a privilege to express gratitude to our guide **Mrs. Prema Jain,** Assistant Professor, Department of Information Science and Engineering, BNMIT who has given us all the support and guidance in completing the project successfully.

We would also like to thank project coordinator **Mr. Manjunath G S**, Assistant Professor, Department of Information Science and Engineering, BNMIT for guiding in a systematic manner.

Abhishek Sahu (1BG15IS001)
Rohit Kamal (1BG15IS040)
Sarthak Sureka (1BG15IS044)
Md. Aasim Ahmad (1BG15IS065)

ABSTRACT

The Internet of Things (IoT) is a system of physical things embedded with sensors, software, electronics and connectivity to allow it to perform better by exchanging information with other connected devices, the operator or the manufacturer[1-3]. It reduces the effort of human by introducing machine to machine interaction. This work has been designed to implement cloud-based solar panel reporting and monitoring through IoT.

Solar energy is a major renewable energy source with the potential to meet many of the challenges facing the world. Use of solar energy as a power source is gradually increasingbecause of its benefits to people and the environment.Technological developments, government policy, and subsidy by the government have reduced the high costs of solar systems. The running costs are less, and the initial investment is regained leading to subsequent savings in energy costs. This happens because the input for solar energy is free and clean sunlight while fossil fuels are mined and transported over long distance. This massive scale of solar photovoltaic deployment requires sophisticated systems for automation of the plant by remote monitoring. Use of the IoT technology for supervising solar photovoltaic panels can greatly enhance its monitoring and maintenance.Temperature and humidity data collection, along withgraphical analysis of the plant helps in real time monitoring and hence facilitates preventive maintenance.

This monitoring is done through Arduino Board which is connected to firebase database. Firebasehas a free and easy accessibility over Internet. Arduino sends the frequent readings of temperature, humidity and power generation by the photovoltaic panel to the firebase database. Android Studio is used to develop an Android application in which a dashboard is created. This application facilitates the Smart Monitoring of the solar panel which displays the daily generation of power through the photovoltaic panel[4].A weekly report is generated in a graphical representation of temperature, humidity and power generation for better and efficient monitoring of the solar panel. This report can be made dynamic as per the requirement of the user. This application will help in the maintenance of the solar panel using mobile phones and is accessible 24x7 depending on the user's convenience. The statistics can be used to study the solar panel in detail.

Table of Contents

Chapter No.	Title	Page No.
1	**Introduction**	**1**
1.1	Motivation	2
1.2	Problem Statement	3
1.3	Objectives	3
1.4	Summary	3
2	**Literature Survey**	**4**
2.1	Base Paper	4
2.2	Existing Systems	5
2.3	Proposed System	7
2.4	Methodology	8
2.5	Summary	8
3	**System Requirement and Specification**	**9**
3.1	Hardware Requirements	9
3.2	Software Requirements	10
3.3	Functional Requirements	10
3.4	Non-functional Requirements	16
3.5	Summary	16
4	**Cost Estimation of the Project**	**17**
4.1	Description of COCOMO model	17

4.2	Cost Estimation	18
4.3	Summary	19
5	**System Design and Development**	**20**
5.1	Architectural Design	20
5.2	Data Flow Diagram	21
5.3	Modules	22
5.4	Flow Chart	24
5.5	Output	24
5.6	Summary	25
6	**Implementation**	**27**
6.1	Languages and Tools	27
6.2	List of Modules	29
6.3	Module Implementation	29
6.4	Summary	38
7	**Testing and Validation**	**39**
7.1	Introduction	39
7.2	Test Cases	41
7.3	Performance Evaluation	43
7.4	Summary	43
8	**Results and Discussions**	**44**
8.1	Snapshots and Description	44
8.2	Summary	47

9	**Conclusion and Future Enhancement**	**48**
9.1	Conclusion	48
9.2	Future Enhancement	48
	References	

List of Figures

Chapter No.	Figure No.	Description	Page No.
3	Fig. 3.1	Arduino UNO	11
	Fig. 3.2	Output Voltage vs Sensed Current	12
	Fig. 3.3	ACS712 Current Sensor	12
	Fig. 3.4	Voltage Sensor Module	12
	Fig. 3.5	Pinout Diagram of Voltage Sensor	13
	Fig. 3.6	Pinout Diagram of Humidity Sensor	13
	Fig. 3.7	Temperature Sensor	14
	Fig. 3.8	LDR Sensor Module	15
	Fig. 3.9	Node MCU	15
5	Fig. 5.1	System Diagram of the System	20
	Fig. 5.2	Level 0 Data Flow Diagram of Solar Panel Reporting and Monitoring System	21
	Fig. 5.3	Level 1 Data Flow Diagram of Solar Panel Reporting and Monitoring System	21
	Fig. 5.4	Level 2 Data Flow Diagram of Solar Panel Reporting and Monitoring System	22
	Fig. 5.5	Working Model of Solar Panel Reporting and Monitoring System	23

	Fig. 5.6	Connection of Sensors with Arduino UNO	23
	Fig. 5.7	Flowchart of working of the System	24
	Fig. 5.8	Output of the System	25
6	Fig. 6.1	Sensor Integration	29
	Fig. 6.2	Updating Database in Firebase	35
	Fig. 6.3	Retrieving Data	36
	Fig. 6.4	Login Screen	37
	Fig. 6.5	Signup Screens	37
	Fig. 6.6	Navigation Drawer	38
	Fig. 6.7	Panel Details	38
8	Fig. 8.1	Dashboard	44
	Fig. 8.2	Panel Health Comparison	45
	Fig. 8.3	Trends	46
	Fig. 8.4	Humidity Trend	47
	Fig. 8.5	Temperature Trend	47

List of Tables

Chapter No.	Table No.	Description	Page No.
4	Table 4.1	Coefficient for different categories of Software Projects	18
	Table 4.2	Cost Estimation of the Project	19
7	Table 7.1	Equivalence Class Testing	41
	Table 7.2	Failure Analysis	42

CHAPTER 1

INTRODUCTION

Electricity is a very basic necessity in today's world, and it can be generated using conventional methods as well as non-conventional methods. Some of the energy carriers like fossil fuels and nuclear fuels are also used, but they are not renewable resources (i.e., they are not 'refilled' by nature). They are said to be conventional sources of energy. In its broadest sense, sustainable power source can be achieved by using the solar power as source. Solar energy has the wide availability throughout the world. The generation of electricity by converting solar energy into an electric power is called as solar thermal energy, which is non-conventional source of energy. Even though various sustainable sources are available such as wind, rain, tides and geothermal, natural based biofuels and inexhaustible biomass, solar power have huge benefits. Nowadays in India, frequent power cut is very common. To tackle the power cut issues, use of renewable energy and its monitoring is important.

Advancement of wired and wireless network technologies has led to the widespread use of internet-connected mobile devices such as smart phones and tablets. Thus, resulting in a new concept called Internet of Things (IoT) which simply means the network of physical objects embedded with sensors, software, electronics and connectivity to allow it to perform better by exchanging information with other connected devices, the operator or the manufacturer [5-6]. It reduces the effort of human by introducing machine to machine interaction. In general, IoT is an information sharing environment where objects in every-day life are connected to wired and wireless networks. Recently, it is used not only for the field of consumer electronics and appliances but also in other various fields such as a smart city, healthcare, smart home, smart car, energy system, and industrial security [7-9]. This provides the connection of each object in the world by means of wireless sensor network. This is achieved by wireless sensor networks, sensor networks, 2G/3G/4G, GSM (Global System for Mobile Communications), GPRS (General Packet Radio Service), RFID (Radio Frequency Identification), Wi-Fi (Wireless Fidelity), GPS (Global Positioning System), microcontroller, microprocessor, etc.

Empowering advancements for the IoT are classified as:

1. Advancement that empower "things" to accept contextual information.
2. Advancement that empower "things" to process the relevant data,

3. Innovation to enhance security and protection.
4. Accepting the information and processing the relevant data can provide an understanding which is needed to build the "intelligence" into "things". This is the highlighted feature that differentiates IoT from standard internet.

1.1 Motivation

At present, the Solar Photovoltaic (PV) energy is one of the pivotal renewable energy sources. The solar energy is becoming a potential solution towards sustainable energy supply in future. Nowadays, rooftop solar PV systems are installed in houses, educational institutions and corporate organisations. Maharashtra, Gujarat, Andhra Pradesh, Tamil Nadu, Karnataka, Rajasthan and Haryana are the leading states which has the greatest number of rooftop solar system installations. Of 1,095-megawatt (MW) rooftop solar capacity in India, Maharashtra leads the way with maximum number of installations and capacity at 145.09MW [10], according to data from the ministry of new and renewable energy (MNRE). MNRE officials said India has set a target of reaching 40-gigawatt (GW) rooftop solar capacity by 2022. As more and more rooftop solar PV systems are getting integrated into the existing grid, there is a growing need for monitoring of real time generation data obtained from solar photovoltaic plants to optimize the overall performance of the solar power plant and to maintain the grid stability. As local monitoring is not possible for the installer, therefore, monitoring remotely is essential for every solar power plant. At this juncture harnessing the power of IoT for monitoring solar power plants by using digital technologies and more advanced computational facilities is promising.

Power generation from solar PV panel is variable in nature due to changes in solar irradiance, temperature and other factors. Thus, remote monitoring of panel is essential. By monitoring the energy forecast, households and communities using solar power can time their energy production and consumption during good weather. For developing remote monitoring system for solar PV power plant, IoT technology is used which envisions a near future where everyday objects will be armed with microcontrollers and transceivers for digital communication. The remote monitoring eliminates the hazards associated [11] with the traditional wiring systems and make data measurement and monitoring process much easier and cost effective and IoT based systems take a giant leap towards monitoring by intelligent decision making from web.

The cloud-based architecture of the remote monitoring systems and its flexibility of deployment make it most suitable for industrial purposes. The Android application dashboard management will enhance the user experience. It also merges comprehensive usage of instrumentation, electronic technology and computer software. At present, prevalent monitoring PV system approach poses some problems like low automaticity and poor real-time experience. These problems can be averted with an efficient remote environment information monitoring and reporting system.

1.2 Problem Statement

Development of inexpensive, easily implementable monitoring and reporting system for solar panels using IoT and cloud-based services.

1.3 Objectives

The objective of this project is to monitor the efficiency of solar panel through an IoT based network and providing acute knowledge about its dependency on environmental factors like temperature, light and humidity. An android application is used by user to view and analyze the collected data.

1.4 Summary

Use of IoT for monitoring of a solar power plant is an important step as day by day renewable energy sources are getting integrated into utility grid. Thus, automation and intellectualization of solar power plant monitoring will enhance future decision-making process for large scale solar power plant and grid integration of such plants. On implementation, the IoT based remote monitoring and reporting system for solar power plant transmits the data to the server for supervision. This system will improve energy efficiency of the system by making use of low power consuming advanced wireless modules thereby reducing the carbon footprint. Android application console-based interface will significantly reduce time of manual supervision and aid in the process of scheduling task of plant management. A provision of advance remote system manages various operations of the Solar PV plants like remote shutdown and remote management which is to be incorporated with the system as later stages.

CHAPTER 2

LITERATURE SURVEY

Various journals and conference papers were studied to understand the development of the application to monitor Solar Panel using IoT. Papers reporting significant results are discussed in the following sections.

2.1 Base Paper

An IoT Based Smart Solar Photovoltaic Remote Monitoring and Control unit, by Soham Adhya, Dipak Saha, Abhijit Das, Joydip Jana, Hiranmay Saha [12]

Using the Internet of Things Technology for supervising solar photovoltaic power generation can greatly enhance the performance, monitoring and maintenance of the plant. Massive scale of solar photovoltaic deployment requires sophisticated systems for automation of the plant monitoring remotely using web-based interfaces as majority of them are installed in inaccessible locations and thus unable to be monitored from a dedicated location. The discussion in this paper is based on implementation of new cost-effective methodology based on IoT to remotely monitor a solar photovoltaic plant for performance evaluation. This facilitates preventive maintenance, fault detection, historical analysis of the plant in addition to real time monitoring. This conceptual system is to monitor the state of a photovoltaic system through an IoT based network in order to control it remotely. The information from the sensors is transmitted via the mobile radio network. A GPRS module is employed to send data to the remote server. The sensing layer comprises of different sensors and microcontroller-based data processing of data acquired from the sensors which communicates with wireless module to initiate and transmit data to server. The data logging from the plant for real time processing is done in the network layer which includes database for storage. In the application layer sophisticated web-based services are designed based on the data collected, processed and stored. Graphical user interface helps to monitor the performance of the plant and the console advices the administrator with decision based on historical data that significantly reduces the decision-making time.

2.2 Existing Systems

Smart Power Monitoring and Control System through Internet of things using Cloud Data Storage, by Putta Sindhuja and M. S. Balamurugan [13]

Lack of resources established in the present world is initiating everyone towards energy efficient technologies. Among all these resources, power is one which needs to be monitored and controlled as per the need since electricity consumption is increasing day-by day. Internet of things reduces the effort of human by introducing machine to machine interaction. This work has been designed to implement smart power monitoring and control system through IoT using cloud data storage. Power consumed by various appliances is monitored through an ARM based controller interfaced to Hall Effect current sensors and stored in a cloud data base known as Xively. Power control of home appliances is achieved through actuators such as relays which can be controlled by client with the help of a web server. The web server is designed using Hyper Text Transfer Protocol for communication between client and server by establishing Remote Procedure Calls between client and server. The designed system enables client to monitor and control the appliances at home from anywhere availing the IoT features of the designed system thereby reducing the wastage of energy.

IoT Based Smart Energy Meter, by Birendrakumar Sahani, Tejashree Ravi, Akibjaved Tamboli, Ranjeet Pisal [14]

To overcome the issue of extra bill amount error or notification from electric board even though the bills are paid, this idea is given to eliminate the third party between the consumer and service provider, overcoming the human error. In this paper, the idea of smart energy meter using IoT and Arduino has been introduced. Arduino is used because of its energy efficiency and less power consumption. In this paper, energy meters which is already installed at our houses are not replaced, but a small modification on the already installed meters can change the existing meters into smart meters. The use of GSM module provides a feature of notification through SMS. One can easily access the meter working through web page that was designed. Current reading with cost can be seen on web page and automatic meter switching on and off is possible.

Solar Energy Monitoring System using IoT, by Suprita Patil, M. Vijayalakshmi & Rakesh Tapaskar [15]

The Internet of Things has a vision in which the internet extends into the real world embracing everyday objects. The IoT allows objects to be sensed and/or controlled remotely over existing network infrastructure, creating opportunities for pure integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. This technology has many applications like Solar cities, Smart villages, Micro grids and Solar Street lights and so on. As Renewable energy grew at a rate faster than any other time in history during this period. This system refers to the online display of the power usage of solar energy as a renewable energy. Monitoring is done through raspberry pi using flask framework. Smart Monitoring displays daily usage of renewable energy. This helps the user to analysis of energy usage. Analysis impacts on the renewable energy usage and electricity issues.

IoT based Smart Energy Monitoring, by Abhiraj Prashant Hiwale, Deepak Sudam Gaikwad, Akshay Ashok Dongare, Prathmesh Chandrakant Mhatre [16]

This paper describes the digitization of load energy usage readings over the internet. The system design eliminates the involvement of human in electricity maintenance. The user can monitor energy consumption in watts from a webpage by providing a channel id for the load. The Webpage utilizes the Thingspeak analytics to analyze the energy usage to give more detailed description and visualization of the energy usage statistics. Wi-Fi unit performs IOT operation by sending energy data of the load to the webpage which can be accessed through the channel id of the device. In this system, consumer can do power management by knowing energy usage time to time. This system utilizes an Arduino microcontroller. The unit which is generated can be displayed on the webpage through the Wi-Fi module. Smart energy meter using Wi-Fi system is designed to provide automated load energy reading over an immediate basis, to use the electricity in an optimized manner and to reduce the power wastage. The data from the system is displayed on a webpage which can be accessed by the consumer. Energy Monitoring using IOT is developed to control home appliances remotely over the cloud from anywhere in the world.

Disadvantages of Existing Systems:

Use of IoT for monitoring of a solar power plant is an important step as day by day renewable energy sources are getting integrated into utility grid. Thus, automation and intellectualization of solar power plant monitoring will enhance future decision-making process for large scale solar power plant and grid integration of such plants.

IoT based remote monitoring system for solar power plant, achieved the remote transmission of data to a server for supervision. But there are various disadvantages of these existing systems. Few are listed below:

- Collected data was not stored properly as it could have been stored in an online cloud like Firebase, from which data can be retrieved as and when required.
- Data analysis was not performed accordingly i.e., no graphical analysis was done based on the values of current, voltage, temperature and light intensity readings.
- Representation of all analysed data was not in the form of graph which could have enhanced the user experience at a great extent.
- Reporting of the analysis was poor due to placing of sensors at improper positions which lead to error in the readings.
- Monitoring of the performance of the solar panel was at a holistic level.

2.3 Proposed System

This project proposes a real time monitoring system for solar panel using the Arduino UNO, which is connected to voltage sensor, current sensor, temperature sensor, humidity sensor and light intensity sensor. This Arduino UNO is also connected to a Wi-Fi module which helps to display the measurements of current, voltage, humidity, light intensity, power generation of solar panel and ambient temperatures through Android application. Monitoring the performance of solar panels using cloud-based services connected with an Android application can be done in real time.

The monitoring system can be developed for the larger PV systems. Large scale grid integration of the solar power plants monitoring requires huge data analytics for decision making.

2.4 Methodology

The purpose of this project is to design and implement an open-source monitoring system for remote solar energy power systems that can deliver useful diagnostic information to system overseers. It is known that changes in operating temperature cause changes of current and voltage of PV cells and PV modules. Therefore, remote monitoring system is designed to monitor the values of current and voltage generated by the solar panel and the ambient temperature via an Android application installed in the smartphone using sensors connected to the Arduino. Humidity and light intensity readings are used to analyze how they affect the overall power generation. Analysis of monitoring results of solar panel output in the form of current, voltage, humidity, temperature and power values performed daily and weekly, helps to determine the overall performance of solar panel.

2.5 Summary

This chapter contains the literature survey for the project and disadvantages of existing systems, proposed system and methodology used in the project. A literature survey is a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. It is important because it describes how the proposed research is related to prior research. It shows the originality and relevance to the topic of the project.

CHAPTER 3

SYSTEM REQUIREMENT AND SPECIFICATION

The following section provides an overview of requirements necessary for the project. The requirements are divided as user requirement, hardware requirement, software requirement, Functional requirement, Non-functional requirement and operational requirement.

3.1 Hardware Requirements

- **Arduino Uno Board**: The digital and analog input/output pins equipped in this board can be interfaced to various expansion boards and other circuits. The Serial communication interface and USB port help to load the programs from computer.
- **Node MCU:** It has in built Wi-Fi module which is generally used to establish the wireless communication between the devices. But this module is not capable of 5-3V logic shifting and will require an external logic converter.
- **Solar Panel**: It is a form of photoelectric cell, defined as a device whose electrical characteristic, such as current, voltage, or resistance, vary when exposed to light. It converts the solar energy into electrical energy. Individual solar cell devices can be combined to be known as solar panels.
- **Connecting wires**: Wires to connect the modules and devices in the Arduino board and Node MCU.
- **Temperature Sensor**: Temperature Sensor to take the frequent reading of temperature values around the solar panel.
- **Current Sensor**: This is used to take the frequent reading of current values around the solar panel.
- **Humidity Sensor**: This is used to take the frequent reading of humidity values around the solar panel.
- **Voltage Sensor**: This is used to take the frequent reading of voltage values around the solar panel.
- **Light Intensity Sensor**: This is used to take the frequent reading of light intensity values around the solar panel.

3.2 Software Requirements

- **Arduino IDE**: Arduino IDE software to write and upload the programming logic onto the Arduino Uno board.
- **Programming language**: C language
- **Android Studio**: Android Studio software to develop the mobile application.

3.3 Functional Requirements

- Solar Panel
- Arduino UNO
- Current Sensor - ACS712
- Voltage Sensor Module (25V)
- Humidity Sensor - DHT11
- Temperature Sensor - LM35
- Light Intensity Sensor - LDR
- ESP8266 Wi-Fi Module
- Interfacing all the sensors with the Arduino board
- Developing the code for the sensors
- Final connection with solar panel to take the readings
- Development of prototype for Android app

Arduino Uno

- It is open source hardware and software platform
- Arduino Uno is a microcontroller board based on the ATmega328P.
- It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.
- It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.
- Operating voltage is 5V
- Flash Memory is 16 KB(ATmega328) of which 2KBused by bootloader

- SRAM is 2 kB, EEPROM is 1 kB, Clock speed is 16MHz

The diagram of Arduino UNO is depicted in Figure 3.1.

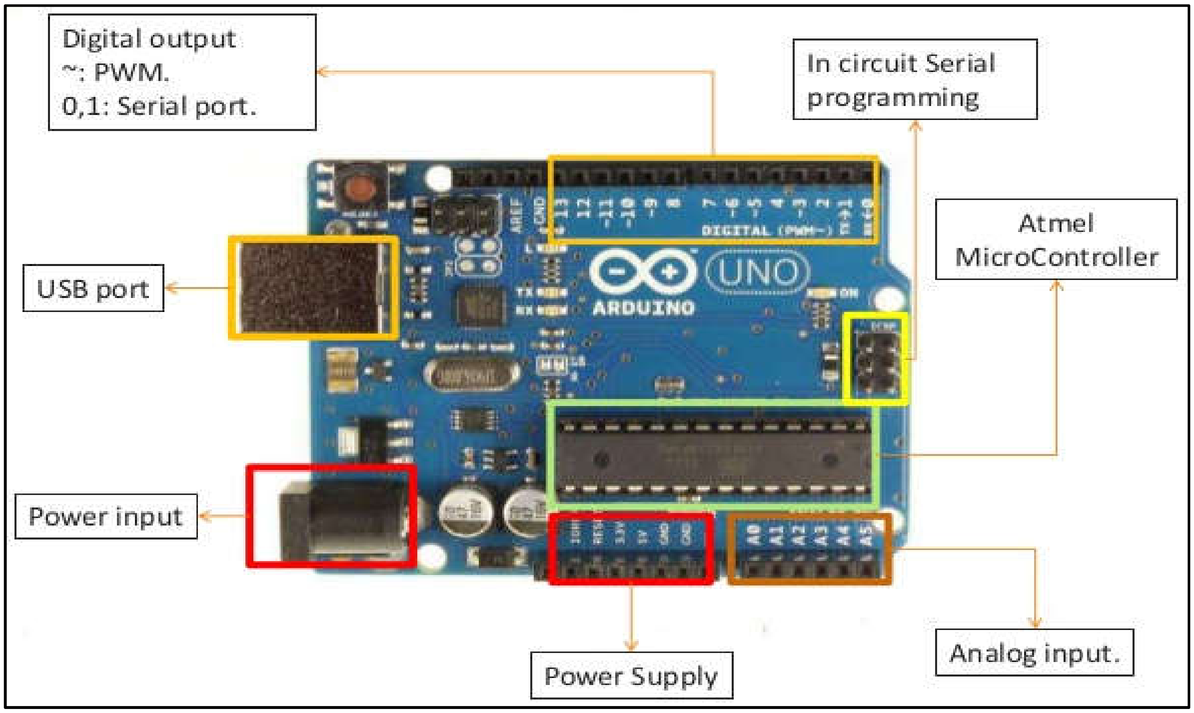


Figure 3.1: Arduino UNO

Why Arduino?

- Inexpensive
- No need for external programmers, compiler licenses etc.
- Simple, clear programming environment
- All done in simple IDE, no command line voodoo
- Cross platform
- Works uniformly across Windows, Linux, MacOS
- Open source
- Both software and hardware

Current Sensor (ACS712)

- The device consists of a precise, low-offset, linear Hall sensor circuit with a copper conduction path located near the surface of the die.
- Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a proportional voltage.

The Output Voltage vs Sensed Current graph and the diagram of ACS712 Current Sensor are depicted in the Figure 3.2 and Figure 3.3 respectively.

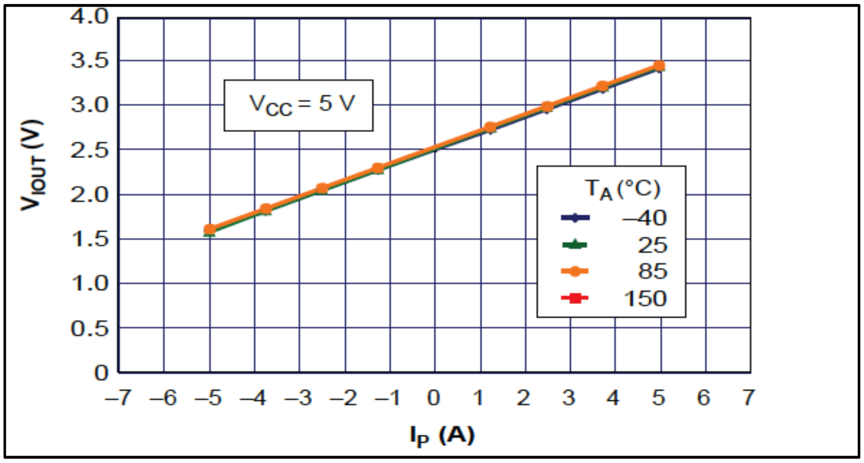


Figure 3.2: Output Voltage vs Sensed Current

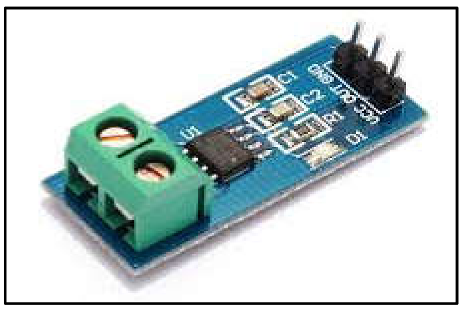


Figure 3.3: ACS712 Current Sensor

Voltage Sensor Module

- This module is based on principle of resistive voltage divider design with divider ratio 5:1
- Maximum input voltage safely withstood: 25 V
- Resistor tolerance: 1% and Resistor Value: 30K/7.5K Ohm

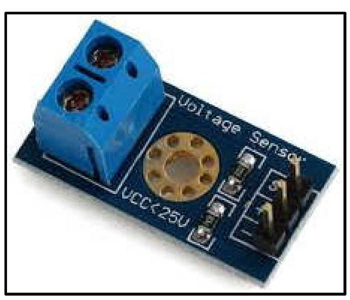


Figure 3.4: Voltage Sensor Module

The pinout diagram of voltage sensor module is shown in figure 3.5.

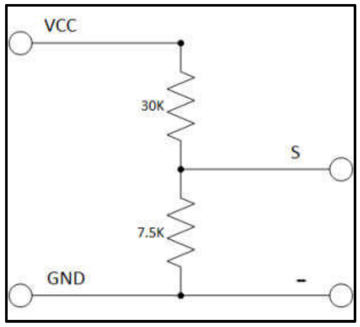


Figure 3.5: Pinout Diagram of Voltage Sensor

Humidity Sensor (DHT11)

- DHT11 is a Humidity and Temperature Sensor, which generates calibrated digital output.
- It can be interface with any microcontroller like Arduino, Raspberry Pi, etc. and get instantaneous results.
- It is a low-cost humidity and temperature sensor which provides high reliability and long-term stability.
- It consists of 3 main components: a resistive type humidity sensor, an NTC (negative temperature coefficient) thermistor and an 8-bit microcontroller, which converts the analog signals from both the sensors and sends out single digital signal.
- It can measure a humidity value in the range of 20 – 90% of Relative Humidity (RH).
- It can measure a temperature in the range of 0 – 500C.
- The sampling period of the sensor is 1 second.

The pinout diagram of humidity sensor is shown in figure 3.6.

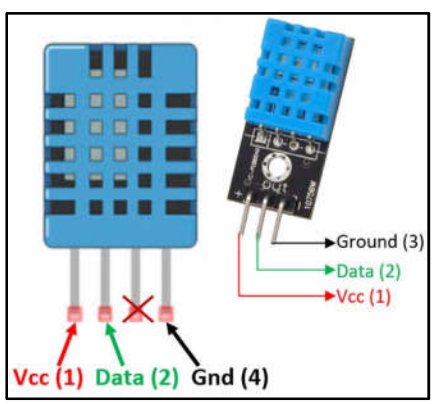


Figure 3.6: Pinout Diagram of Humidity Sensor

Temperature Sensor (LM35)

- The LM35 Temperature Sensor is precision integrated-circuit temperature device with an output voltage linearly-proportional to the Centigrade temperature.
- It does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range.
- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at25°C)
- Rated for Full −55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost
- Operates From 4 V to 30 V
- Less than 60-µA Current Drain

The pinout diagram of temperature sensor is shown in figure 3.7.

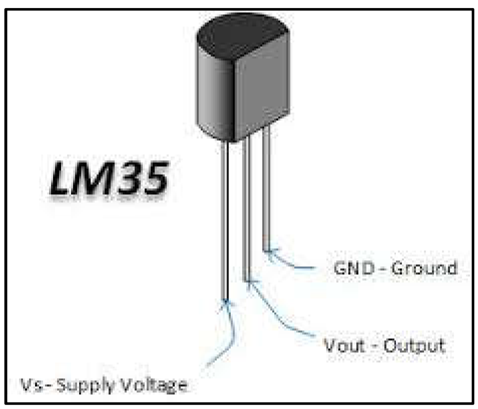


Figure 3.7: Temperature Sensor

Light Intensity Sensor Module (LDR)

- LDR sensor module is used to detect the intensity of light.
- When there is light, the resistance of LDR will become low according to the intensity of light. The greater the intensity of light, the lower the resistance of LDR.
- Input Voltage: DC 3.3V to 5V
- Output: Analog and Digital
- Sensitivity adjustable

The diagram of LDR sensor module is shown in figure 3.8.

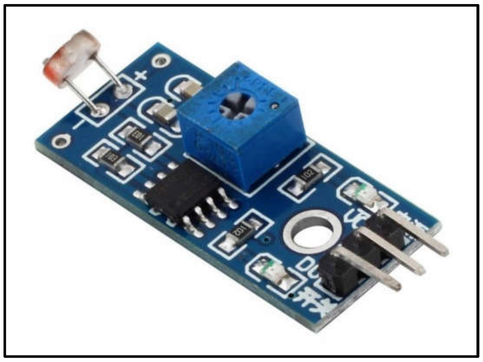


Figure 3.8: LDR Sensor Module

Node MCU

- Node MCU uses ESP8266 which is Wi-Fi enabled system on chip (SoC) module developed by Espressif system.
- It is mostly used for development of IoT (Internet of Things) embedded applications.
- It comes with capabilities of2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2), general-purpose input/output (16 GPIO),Inter-Integrated Circuit (I²C) serial communication protocol, analog-to-digital conversion (10-bit ADC).
- It employs a 32-bit RISC CPU based on the TensilicaXtensa L106 running at 80 MHz.
- It has a 64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM.
- Node MCU is low cost standalone wireless transceiver that can be used for end-point IoT developments.

The figure of Node MCU is shown in figure 3.9.

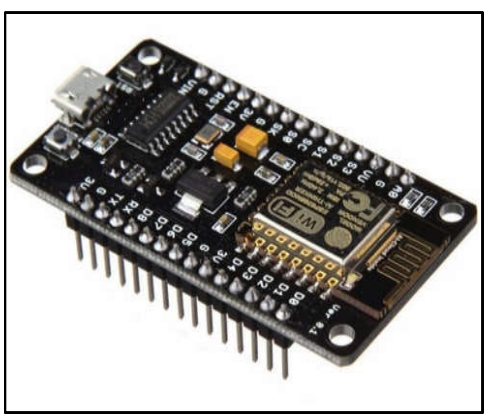


Figure 3.9: Node MCU

3.4 Non-functional Requirements

- **Performance:** Response time of the proposed system should be fast enough for the user to get better experience.
- **Scalability:** Should be scaled to all installations of solar panels at a specific place.
- **Capacity:** Capacity indicates the amount or size of input data (sensor readings) the proposed system can process without much deterioration in performance.
- **Usability:** The application shall be user friendly and doesn't require any guidance to be used. In other words, the application must be as simple as possible, so its users shall use it easily.
- **Reliability:** The application should not have any unexpected failure. In order to avoid any failure's occurrence, the specifications have been respected and followed correctly. The only problem that may occur in some cases is that the application does not provide 100% accuracy.

3.5 Summary

This chapter contains the functional and non-functional requirements, software and hardware requirements that are needed for efficient working of the proposed system. The non-functional requirements, the most important ones are the Performance, Scalability of the number of solar panels, Capacity of the readings, Usability and Reliability. Under software requirements, Arduino IDE is a free and open-source software platform for coding of Arduino UNO in C language, that aims to establish connection between sensors and Arduino UNO board. Android Studio is a software used to develop the mobile application. Under hardware requirements, interfacing all the sensors (Current Sensor, Voltage Sensor Module, Humidity Sensor, Temperature Sensor, Light Intensity Sensor Module) and the Wi-Fi Module with the Arduino UNO board is described. The development of the code for the sensors must be done properly in order to minimise errors.The final connection of all the sensors along with Arduino UNO, with the solar panel, in order to take the readings, is done carefully.

CHAPTER 4

COST ESTIMATION OF THE PROJECT

4.1 Description of COCOMO Model

COCOMO (Constructive Cost Estimation Model) was proposed by Boehm [1981]. According to Boehm, software cost estimation should be done through three stages. Basic COCOMO, intermediate COCOMO, and complete COCOMO.

The basic COCOMO model gives an approximate estimate of the project parameters. The basic COCOMO estimation model is given by the following expressions:

Effort Applied (E) = a_b(KLOC)b_b[man-months]

Development Time (D) = c_b(Effort Applied)d_b[months]

People required (P) = Effort Applied / Development Time [count]

- KLOC is the estimated size of the software product expressed in kilo lines of code.
- a1, a2, b1, b2 are constants for each category of software products.
- D is the estimated time to develop the software, expressed in months.
- Effort is the total effort required to develop the software product, expressed in person months (PMs).

Boehm postulated that any software development project can be classified into one of the following three categories based on the development complexity:

- Organic: A development project can be considered of organic type, if the project deals with developing a well understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects.
- Semi-detached projects: A developing project can be considered of semidetached type, if the development consists of a mixture of experienced and inexperienced staff. Team members may have limited experience on related systems but may be unfamiliar with some aspects of the system being developed.
- Embedded projects: A development project is considered to be of embedded type, if the software being developed is strongly coupled to complex hardware, or if the strongest regulations on the operational procedures exist.

Table 4.1: Coefficient for different categories of software projects

Software Project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

4.2 Cost Estimation

- The type of project that is being implemented is a semi-detached project.
- The values of the co-efficient are: a=3.0, b=1, c=2.5 and d=0.35. The value is KLOC=6.
- Effort applied (E) is found to be 22.31 man-months.
- Development time is found to be 7.4 months.
- Minimum number of people required is found to be 4 people.
- The effort applied, and development time as shown in the circulation above, in Table 4.1, is in compliance with the actual effort applied and development time taken.

Cost estimation of the project, also, means the approximation of cost involved by estimating the procedural costs of processes involved and use of licensed software and hardware components. The components used in the project are carefully selected based on numerous successful trials. These components are durable and long lasting, and the software used are mostly open source which supports the low-cost concept of this project.

The cost of all the hardware components and software involved is shown in Table 4.2.

Table 4.2: Cost Estimation of the Project

Components	**Cost Involved (in ₹)**
Arduino UNO	550
Node MCU	350
Current Sensor	150
Voltage Sensor	200
Light Sensor	70
Temperature and Humidity Sensor	350
Connecting Wires	250
Battery	400
Solar Panel	4200
Android Studio	Free(Open Source)
Arduino IDE	Free(Open Source)
Firebase Cloud	Free(Open Source)
Total	**6520**

4.3 Summary

This chapter emphasizes on cost estimation of the project. Cost estimation is done on the basis of COCOMO model and involvement of money for the project work. Being able to make accurate cost estimate, is key to delivering a solid project plan. Cost estimating utilizes many techniques that translate the project scope into deliverables and develop an approximation of costs of the resources needed to complete project activities.

CHAPTER 5

SYSTEM DESIGN AND DEVELOPMENT

5.1 Architectural Design

The system diagram of the proposed system and its components is shown below in Figure 5.1:

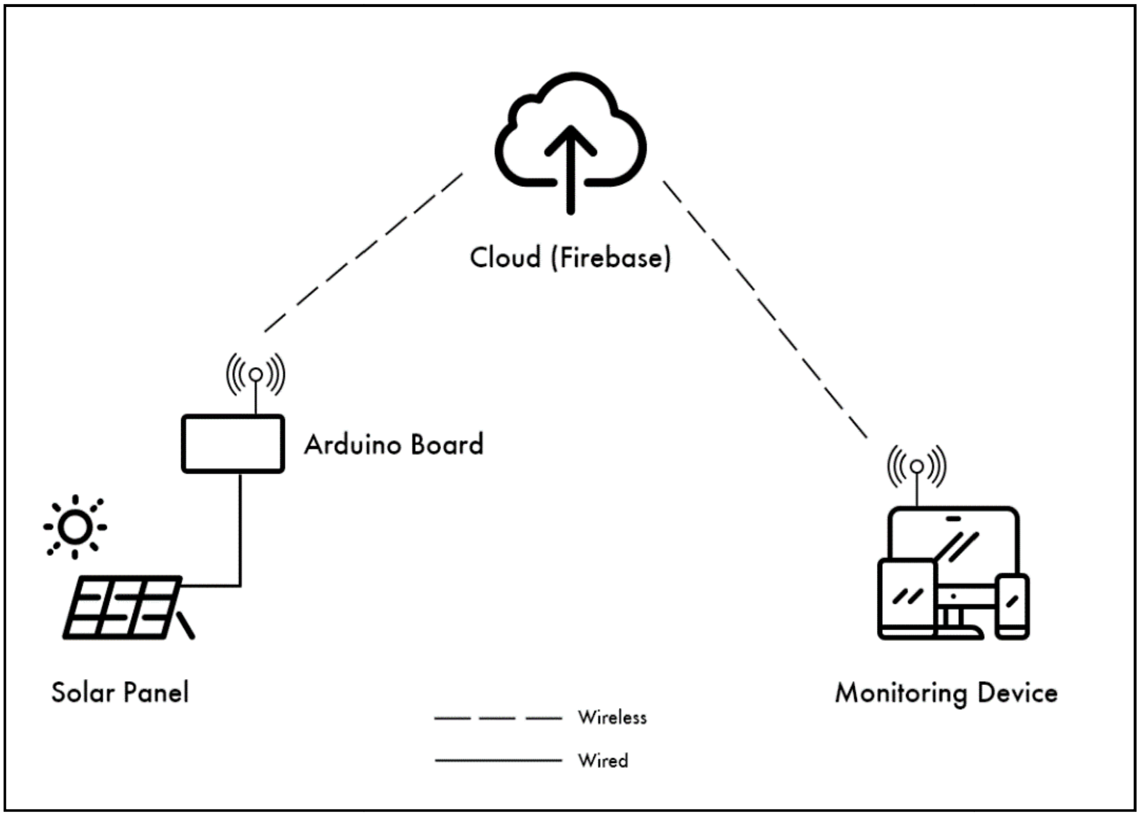


Figure 5.1: System diagram of Solar Panel Reporting and Monitoring System

The Solar Panel Reporting and Monitoring system is divided into two different parts, as shown in the Figure 5.1. The first component is a hardware-centric component that interfaces with a solar charge controller to determine diagnostic information about a power system. This component will process obtained data and report it to the firebase database. The second component of the system is the recipient software on the other end. The database is connected to an android application which will show the historical data about the solar system and provide a dashboard kind of interface through which long term statistics can be determined and effective monitoring can be conducted.The graphical representation of the collected data from the solar panel enhances the user experience at a greater extent making the solar panel management more efficient.

5.2 Data Flow Diagram:

A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Following figures show the data flow diagrams of the proposed system where each activity is identified.

Level 0 and Level 1 Data Flow Diagrams of Solar Panel Reporting and Monitoring System are shown in the Figure 5.2 and Figure 5.3.

5.2.1 Data Flow Diagram- Level 0

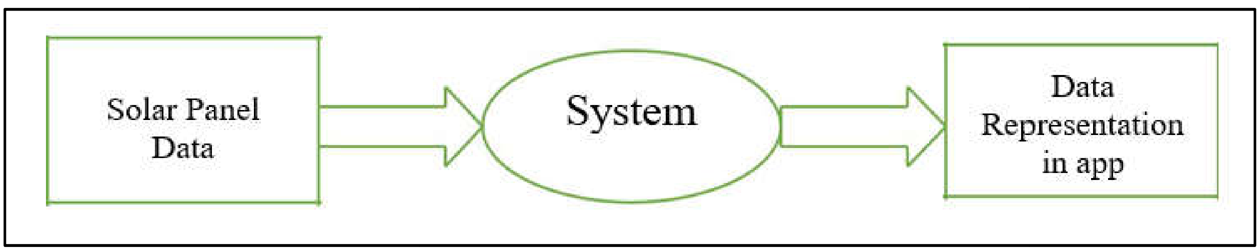


Figure 5.2: Level 0 Data Flow Diagram of Solar Panel Reporting and Monitoring System

5.2.2 Data Flow Diagram- Level 1

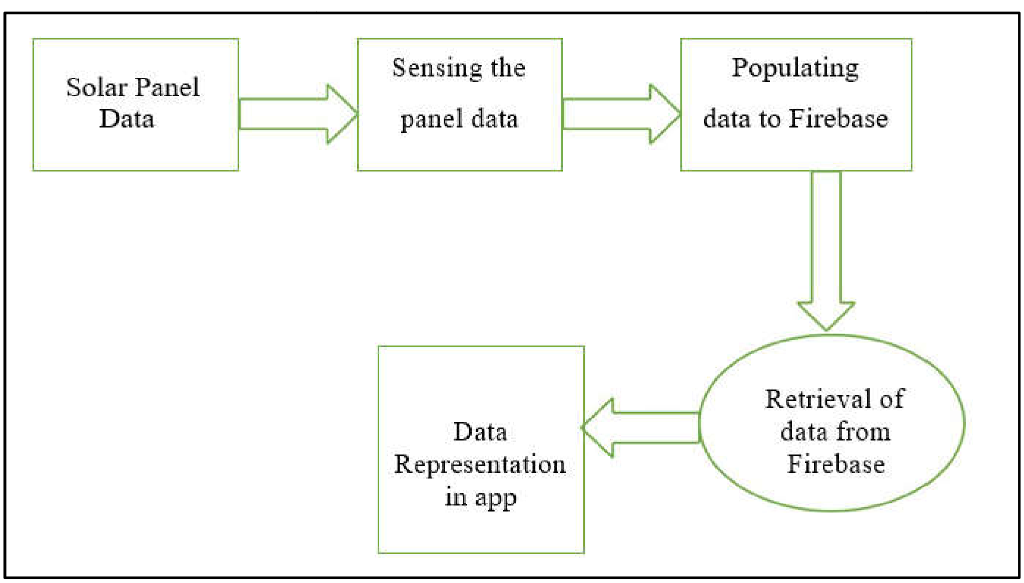


Figure 5.3: Level 1 Data Flow Diagram of Solar Panel Reporting and Monitoring System

5.2.3 Data Flow Diagram- Level 2

Level 2 Data Flow Diagram of Solar Panel Reporting and Monitoring System is shown in the Figure 5.4.

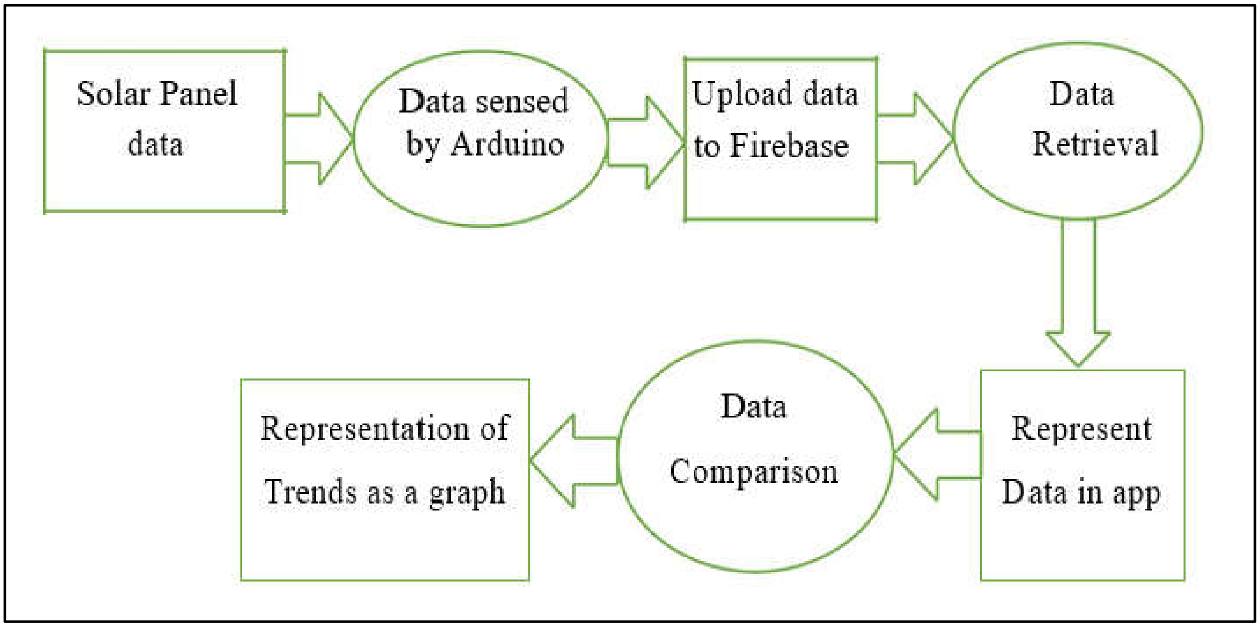


Figure 5.4: Level 2 Data Flow Diagram of Solar Panel Reporting and Monitoring System

5.3 Modules

The modules used in the Solar Panel Reporting and Reporting System, are described below:

Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller which is either to be connected to a computer with a USB cable or to be powered with an AC-to-DC adapter or battery to get started.

Node MCU

Node MCU uses ESP 8266 Wi-Fi module, which is a low-cost component with which wirelessly networkable microcontroller modules are made. It is a system-on-a-chip module with capabilities for 2.4GHz range. It employs a 32-bit RISC CPU running at 80 MHz. It is based on the

TCP/IPand performs the IoT operation. It has 64 kb boot ROM, 64 kB instruction RAM, 96 kB data RAM.

The working model diagram of the proposed system is shown below in Figure 5.5.

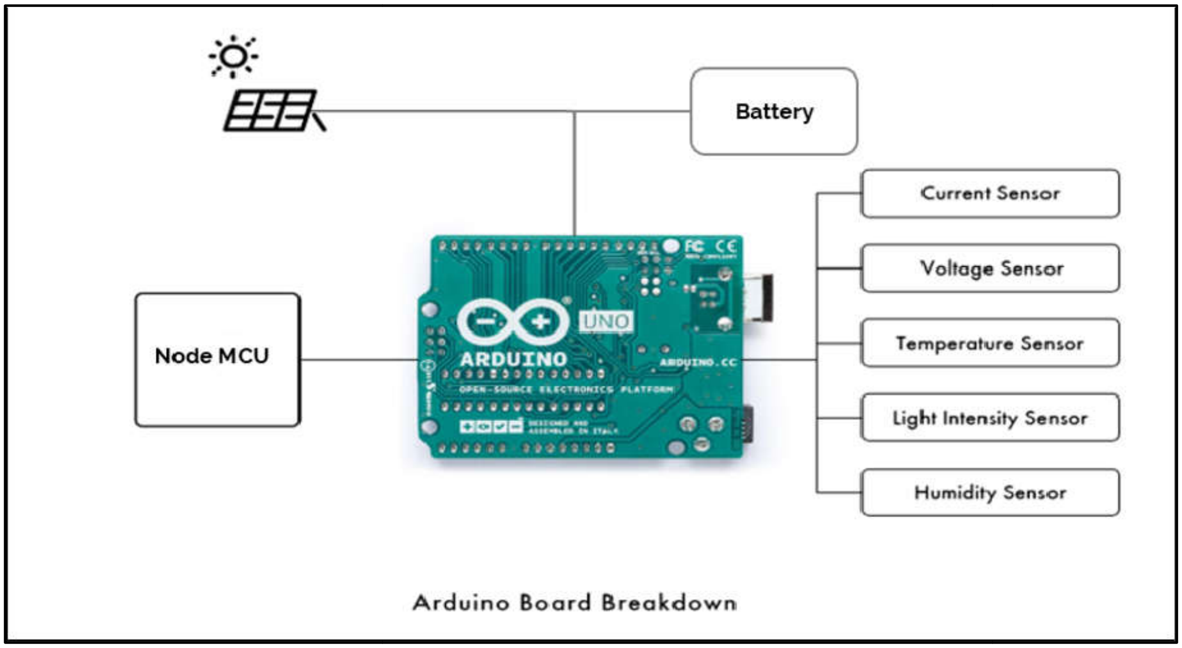


Figure 5.5: Working Model of Solar Panel Reporting and Monitoring System

Connection Figure

The connection figure of the connection of sensors to Arduino UNO and Node MCU is shown in Figure 5.6.

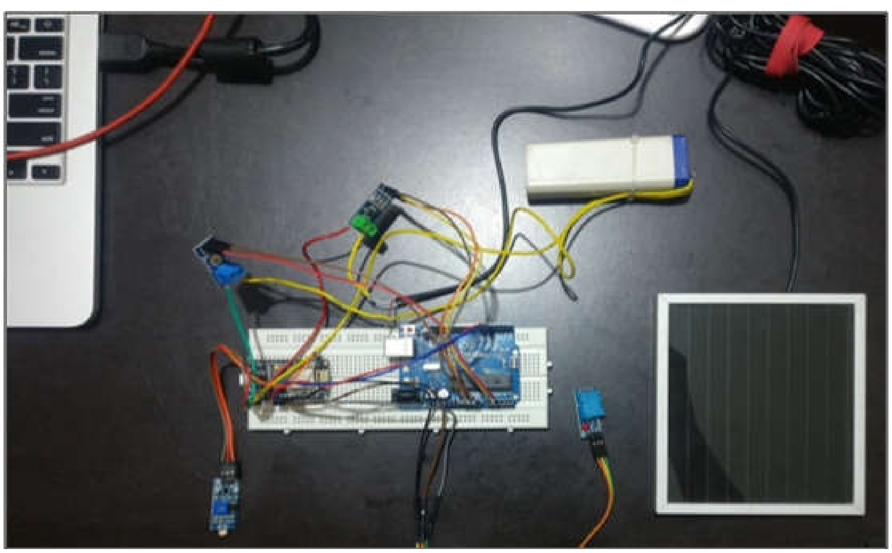


Figure 5.6: Connection of sensors with Arduino UNO

5.4 Flowchart

The flowchart of the working of the Solar Panel Reporting and Monitoring System is depicted in the Figure 5.7.

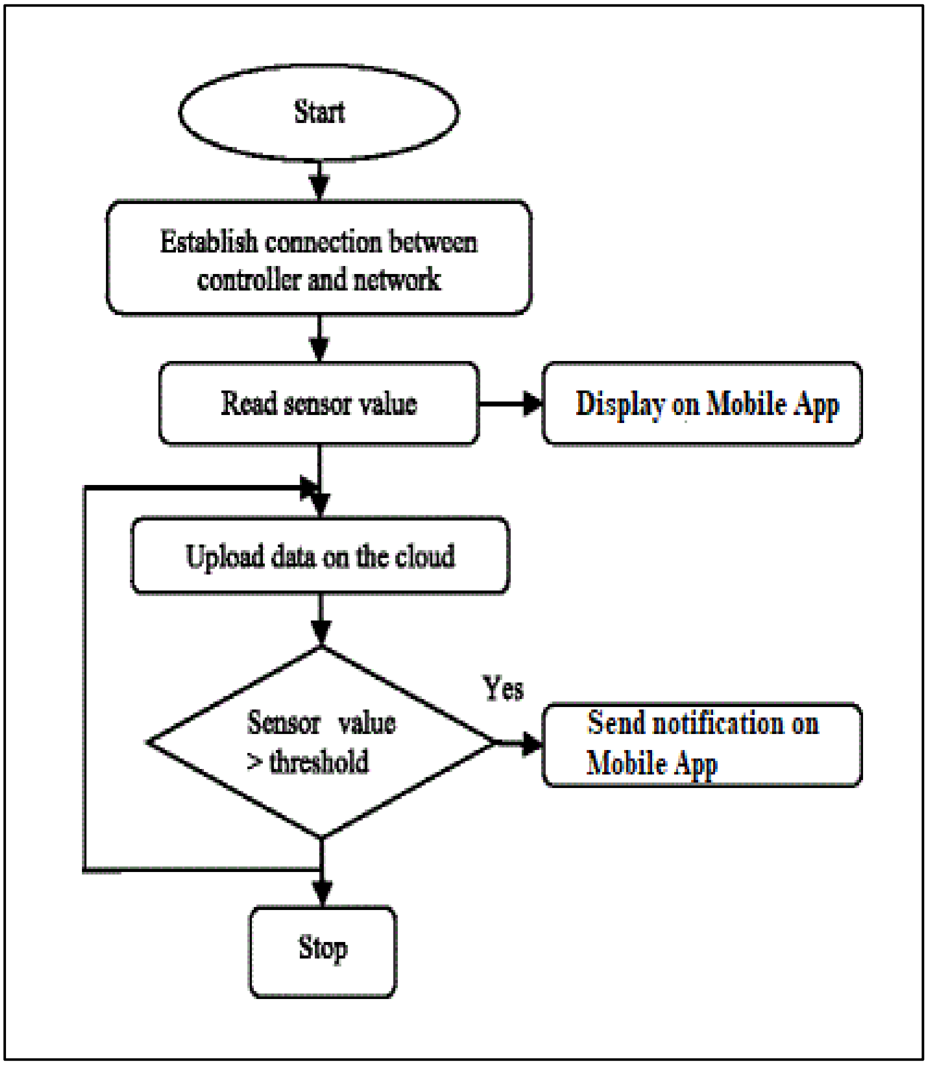


Figure 5.7: Flowchart of working of the system

5.5 Output

The current and voltage sensor readings of the Solar Panel Reporting and Monitoring System on Arduino IDE is depicted in the Figure 5.8.

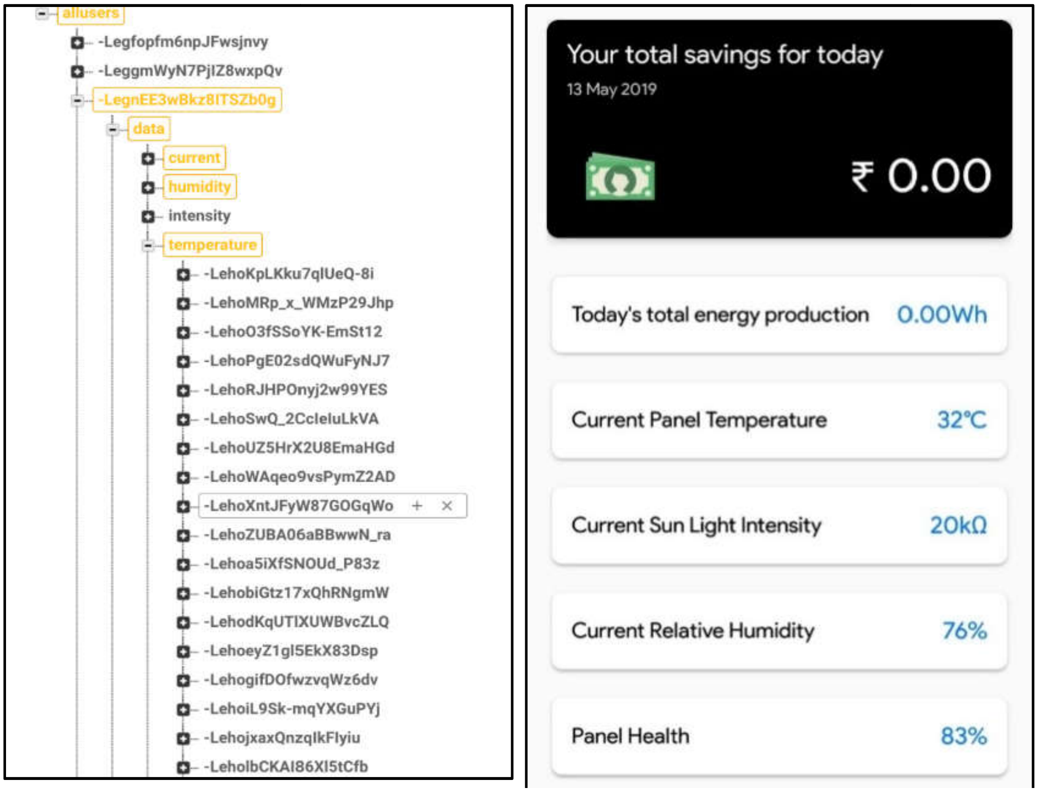


Figure 5.8: Output of the system

5.6 Summary

The purpose of this project is to design and implement an open-source monitoring system for remote solar energy power systems that can deliver useful diagnostic information to system overseers. It is known that changes in operating temperature cause changes of current and voltage of PV cells and PV modules. Therefore, remote monitoring system is designed to monitor the values of current and voltage generated by the solar panel and the ambient temperature via an Android application installed in the smartphone using sensors connected to the Arduino. Humidity and light intensity readings are used to analyze how they affect the overall power generation. Analysis of monitoring results of solar panel output in the form of current, voltage, humidity, temperature and power values performed daily and weekly, helps to determine the overall performance of solar panel.

Benefits of this system are listed below:

- The data populated about energy consumed, usage profile, billing data, etc. in the database can be used for historical analysis.

- The existing approach must be modified with the new proposed system by using sophisticated database technologies and equipped with much more embedded intelligence for faster data processing and computation.
- IoT based remote monitoring improves energy efficiency of the system by making use of low power consuming advanced wireless modules, hence, reducing the carbon footprint.
- This system populates the dedicated web server-based database with real time data of the energy consumption, that enhances the decision-making process of the concerned authority.
- Application console-based interface significantly reduces time of manual supervision and data recording and aids in the process of scheduling task of energy management.
- The graphical representation of the consumption data collected enhances the user experience at a greater extent making the energy management more efficient.
- Android application is handy, user-friendly and easy to manage.

CHAPTER 6

IMPLEMENTATION

Solar Panel Reporting and Monitoring senses data from a solar panel with the help of various sensors installed on an Arduino board that processes obtained data and report it to the firebase database. The database is connected to an android application which is developed using Android Studio IDE. It shows the historical data about the solar system and provide a dashboard kind of interface through which long term statistics can be determined and effective monitoring can be conducted. The graphical representation of the collected data is achieved using GraphView library provided by Android Studio.

6.1 Languages and Tools

Arduino IDE:

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards. The source code for the IDE is released under the GNU General Public License, version 2. It supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

C language:

C is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations. By design, C provides constructs that map efficiently to typical machine instructions, and it has therefore found lasting use in applications that were previously coded in assembly language. Such applications include operating systems, as well as various application software for computers ranging from supercomputers to embedded systems.

C was originally developed at Bell Labs by Dennis Ritchie, between 1972 and 1973. It was created to make utilities running on Unix. Later, it was applied to re-implementing the kernel of the Unix operating system. It is an imperative procedural language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal runtime support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. C language has become available on various platforms, from embedded microcontrollers to supercomputers.

Android Studio:

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.It supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go and Android Studio 3.0 or later supports Kotlin and Java 7 language features and a subset of Java 8 language features that vary by platform version. External projects backport some Java 9 features.

Java language:

Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA) which means that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to "bytecode" that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them.

As of 2018, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

6.2 List of Modules

1. Sensing Data: Sensors sense data.

2. Upload Data to Firebase: Data collected at Arduino Uno is uploaded to firebase.

3. Retrieve data from Firebase to App: Data is retrieved by android application.

4. Android Application Development: Android app is developed using Android Studio.

6.3 Modules Implementation

Sensing Data:

Various sensors like current, voltage, humidity, temperature and light are installed on Arduino UNO and Node MCU board respectively which senses data. Arduino IDE contains the source code for each sensor which allows them to give proper readings and stores them in portable form. A proper arrangement of sensors on a breadboard connected with solar panel is shown in Figure 6.1.

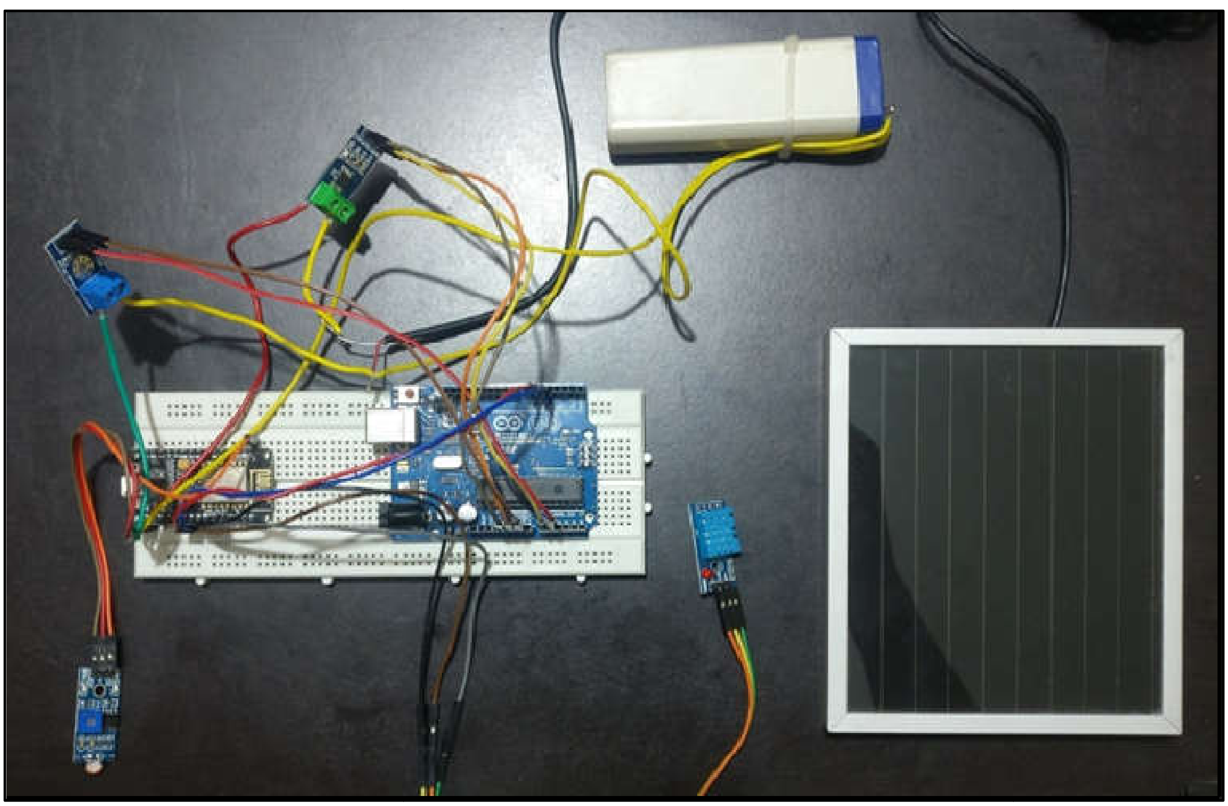


Figure 6.1: Sensor Integration

Pseudo Code: to send sensed data from sensors through Arduino to Node MCU.

```
#include <SoftwareSerial.h>
#include <ArduinoJson.h>
SoftwareSerials(5,6);
#define VIN A0      //define the Arduino pin A0 as voltage input (V in)
const float QOV =   0.5 * 5.0;           //set quiescent Output voltage of 0.5V
int offset =20;         //set the correction offset value
float voltage;          //internal variable for voltage
float current;
void setup(){
Serial.begin(9600);
s.begin(9600);
}
void loop(){
    float voltage_raw=  (5.0 / 1023.0)* analogRead(VIN);   //read the voltage from sensor
    float voltag=  voltage_raw - QOV + 0.012;         //0.000 to make voltage zero if no current
    float current = voltag / 0.1;

    int volt = analogRead(A1);          //read the input
    double voltage = map(volt,0,1023, 0, 2500)+offset;
    //map 0-1023 to 0-2500 & add correction offset
    voltage /=100;   //divide by 100 to get the decimal values
delay(500);
StaticJsonBuffer<1000>jsonBuffer;
JsonObject& root = jsonBuffer.createObject();
    root["data1"] = voltage;
    root["data2"] = current;
    if(s.available()>0){
root.printTo(s);
    }
}
```

Upload Data to Firebase:

Data from Node MCU board is uploaded to firebase database which acts as an open bulky storage unit for all sensed data. Firebase manages all data real-time in the database. So, the exchange of data to and fro from the database is easy and quick.

Pseudo Code: to transfer data from Node MCU to Firebase.

```
#include "DHT.h"   //including the library of DHT11 temperature and humidity sensor
#define DHTTYPE DHT11
#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <SoftwareSerial.h>
SoftwareSerial s(D6,D5);
#define dht_dpin D4
#include <NTPClient.h>
#include <WiFiUdp.h>
#define NTP_OFFSET +19800         //in seconds
#define NTP_INTERVAL 60*1000            //in milliseconds
#define NTP_ADDRESS "europe.pool.ntp.org"
WiFiUDPntpUDP;
NTPClienttimeClient(ntpUDP, NTP_ADDRESS, NTP_OFFSET, NTP_INTERVAL);
#define FIREBASE_HOST "my-solr.firebaseio.com"         //database apiurl
#define FIREBASE_AUTH "Hgj0eZAqaVeir7Sx8CJvKxLHABx0qsp01Tr3V4sa"
//database secret
DHT dht(dht_dpin, DHTTYPE);
const int LDR = A0;   //defining LDR PIN
```

```
int input_val = 0;        //to store input values
const char* ssid="#SKYNET";
const char* password = "password#@#@4693";
void setup() {
Serial.begin(9600);
dht.begin();
Serial.print("Wifi connecting to ");
Serial.println( ssid );
WiFi.begin(ssid,password);
Serial.println();
Serial.print("Connecting");
while( WiFi.status() != WL_CONNECTED ){
delay(500);
Serial.print(".");
    }
Serial.println("Wifi Connected Success!");
Serial.print("NodeMCU IP Address: ");
Serial.println(WiFi.localIP());
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);    //connect to Firebase
s.begin(9600);
    while (!Serial)
        continue;
timeClient.begin();
}
void loop() {
```

```
    float h = dht.readHumidity();
    float t = dht.readTemperature();
input_val = analogRead(LDR);        //reading input
StaticJsonBuffer<1000>jsonBuffer;
JsonObject& root = jsonBuffer.parseObject(s);
    if (root == JsonObject::invalid())
        return;
root.prettyPrintTo(Serial);
    float data1=root["data1"];
    float data2=root["data2"];
timeClient.update();
    unsigned long epochTime = timeClient.getEpochTime();
JsonArray& array1 = jsonBuffer.createArray();
JsonObject& object1 = array1.createNestedObject();
    object1["timestamp"] = epochTime;
    object1["value"] = data1;
Firebase.push("allusers/-LegnEE3wBkz8ITSZb0g/data/voltage/",array1);

JsonArray& array2 = jsonBuffer.createArray();
JsonObject& object2 = array2.createNestedObject();
    object2["timestamp"] = epochTime;
    object2["value"] = data2;
Firebase.push("allusers/-LegnEE3wBkz8ITSZb0g/data/current/",array2);

JsonArray& array3 = jsonBuffer.createArray();
```

```
JsonObject& object3 = array3.createNestedObject();
    object3["timestamp"] = epochTime;
    object3["value"] = h;

Firebase.push("allusers/-LegnEE3wBkz8ITSZb0g/data/humidity/",array3);
JsonArray& array4 = jsonBuffer.createArray();
JsonObject& object4 = array4.createNestedObject();
    object4["timestamp"] = epochTime;
    object4["value"] = t;

Firebase.push("allusers/-LegnEE3wBkz8ITSZb0g/data/temperature/",array4);
JsonArray& array5 = jsonBuffer.createArray();
JsonObject& object5 = array5.createNestedObject();
    object5["timestamp"] = epochTime;
    object5["value"] = input_val;
Firebase.push("allusers/-LegnEE3wBkz8ITSZb0g/data/intensity/",array5);
delay(5000);
}
```

Transmission of data from Arduino IDE to firebase database is depicted in Figure 6.2 as shown below. This data is shown in JSON format. When a new branch is added to the database i.e., new user being added, new sensor being added etc., the color type of the key in the key-value pair changes to green for few seconds indicating new branch being added. Similarly, yellow color depicts that the data related to the keys are being updated. In the underlying figure, it can be easily identified that current, humidity and temperature data are being updated whereas intensity data is already been updated.

Figure 6.2: Updating Database in Firebase

Retrieving Data from Firebase:

Data from firebase is retrieved by android application so that it can be used to develop graphs and end-user results.

Retrieval of data from firebase database to android application is depicted in Figure 6.3 as shown below.

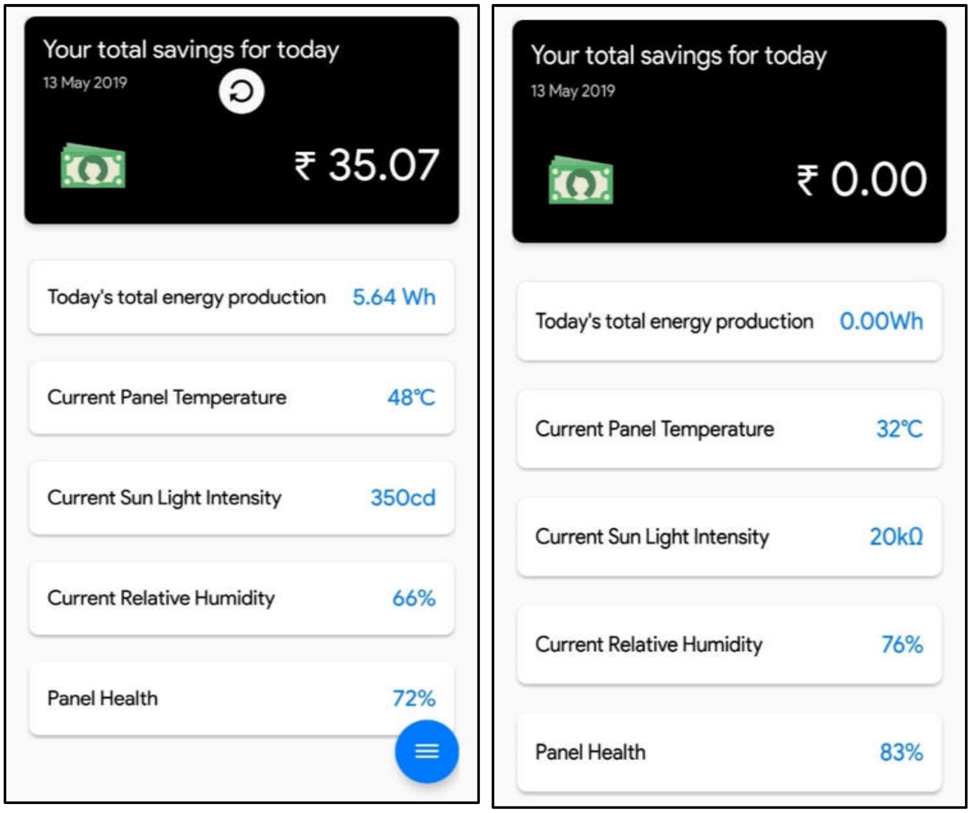


Figure 6.3: Retrieving Data

Android App Development:

An android application is developed using Android Studio IDE which displays data sensed by sensors placed on solar panel. It is developed by writing code in Java language and using appropriate layout features provided by this tool. It displays the final output to the user after analyzing it with previously stored data.

Login screen consists of link to user registration screen wherein user can register itself and can enter solar panel details. The login screen enables the user to log into the application to view the status of solar panel along with live sensor readings. This screen is made attractive to enhance user experience.

Login-cum-welcome screen is depicted in Figure 6.4.

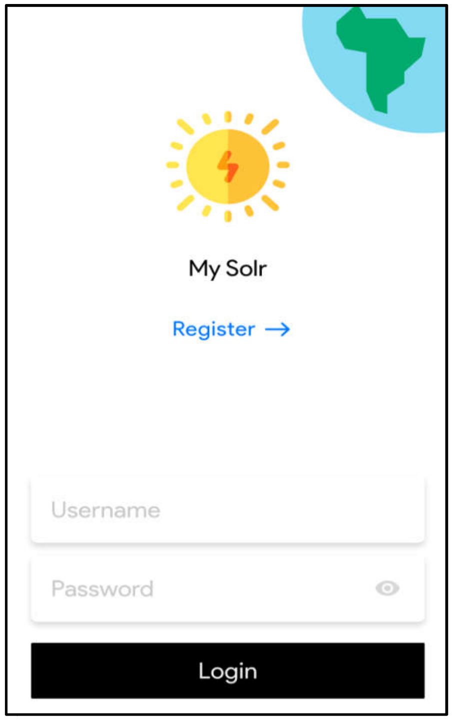


Figure 6.4: Login Screen

Signup screens are shown in Figure 6.5.These screens include user registration and panel data entry.

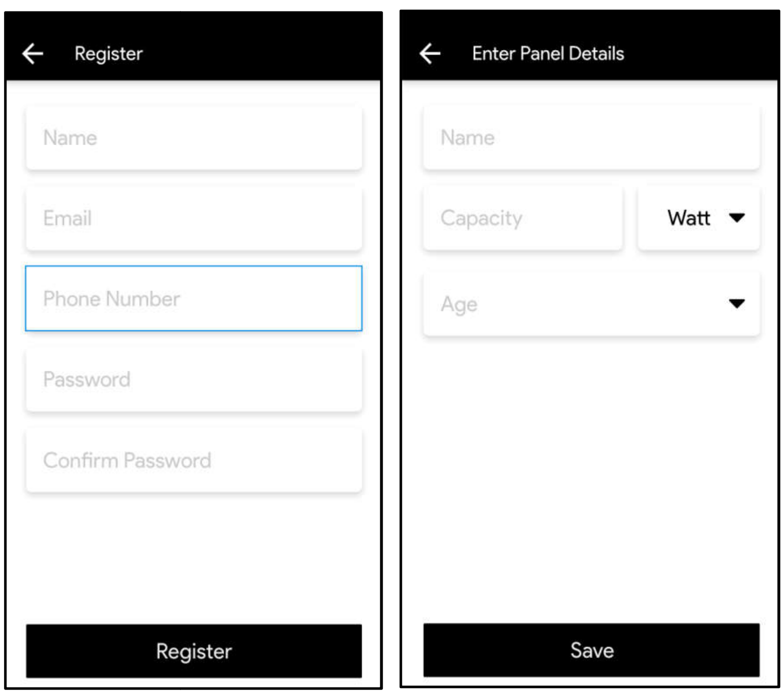
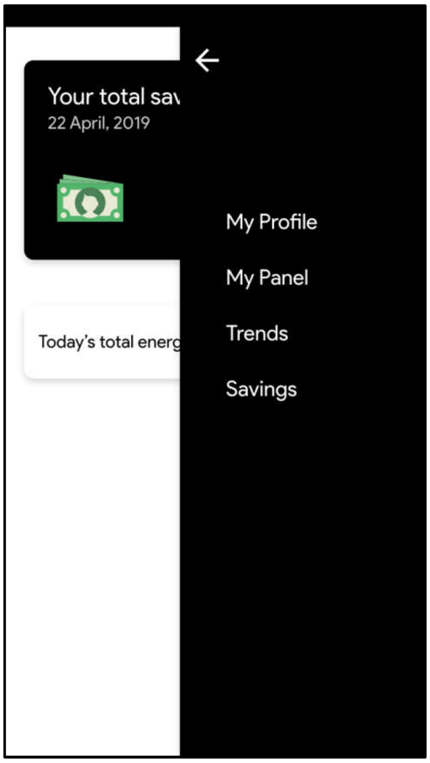


Figure 6.5: Signup Screens

Navigation drawer and solar panel details are depicted in Figure 6.6 and Figure 6.7 respectively.



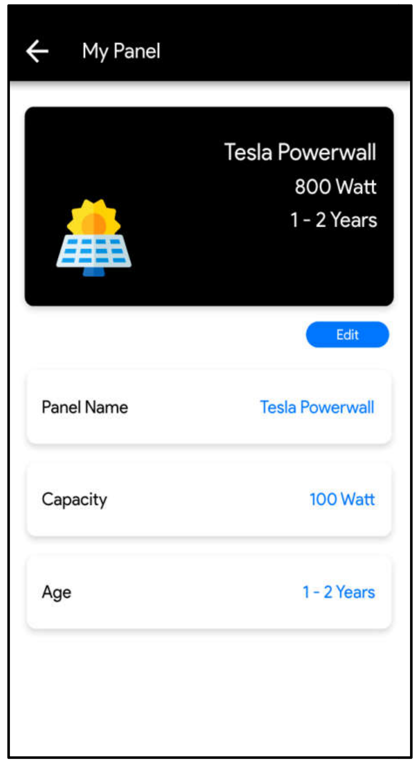


Figure 6.6: Navigation Drawer **Figure 6.7: Panel Details**

6.4 Summary

Solar Panel Reporting and Monitoring is achieved involving these steps: sensing data, interfacing it with firebase, retrieving data from firebase in android studio and building android app. Sensors are installed on microcontroller board and each sensor's source code is written using Arduino IDE. Data collected at Arduino UNO board is uploaded to firebase which supports real-time exchange of data. Data stored in firebase is retrieved by android app for final display. Android Studio is used to develop an application that displays data fetched from firebase to the user in a customized format.

CHAPTER 7

TESTING AND VALIDATION

7.1 Introduction

Testing is the most important phase in the development of any software. It is the final verification and validation activity within an organization. Software testing is the technical investigation of the product under test, to provide stakeholders with quality related information. In particular, software system testing is a method of verification involving the execution of software systematically to detect defects. For testing the objective should be to design a test that systematically uncovers different classes of errors and to do so with minimum amount of time and effort. Testing is carried out during the implementation phase to verify that the software behaves as intended by its designer and after the implementation phase is complete. Later testing phase confirms with the requirements of the system.

7.1.1 Testing Objectives

The main objective of testing is to prove that the software product meets a set of pre-established acceptance criteria under a prescribed set of environmental circumstances. Other objectives of software testing include:

- Detection and prevention of defects
- Verification and validation of user requirements
- Improve the ease of integration and revision of components
- Focus on accurate and reliable results
- To ensure that the software requirement specification is met
- Evaluate the capabilities of a system and its performance

7.1.2 Testing Process

Test cases should be based primarily on the software requirements and developed to verify correct functionality and to establish conditions that reveal potential errors. They are the key factors to the success of any system. Performance of a system is analyzed based on the cases written for each and every module of a system.

7.1.3 Testing Strategies

A testing strategy is a general approach to the testing process rather than a method of devising particular system or component tests. Different testing strategies may be adopted depending on the type of system to be tested and the development process used. The following testing strategies are used to depict the errors or results of modules of the project.

Unit Testing is a level of software testing where all the independent units of the project are tested to ensure that the information properly flows in and out of the program. It is the first level of software testing and is performed prior to integration testing. The purpose is to validate that each unit of the software performs as designed. Data is examined to ensure that the modules are working properly. The modules were also tested in order to give appropriate messages in case of errors. Unit testing increases confidence in changing and maintaining code.

Integration Testing is a phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes the modules that have been unit tested as input, groups them in larger aggregates, applies tests to those aggregates, and delivers the integrated system ready for system testing as its output. The purpose of integration testing is to detect any inconsistency between the software units that are integrated.

System Testing is conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing takes all the integrated software components that have successfully passed integration testing as its input. It seeks to detect defects both within the inter-modules and the system. It tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specification. Usually black box testing method is used for performing system testing. Independent testers perform system testing.

Equivalence Class Testing is applied on the temperature ranges on which the solar panel is operating, which are divided into four classes of 'Temperature Ranges' ranging between 25°C to 45°C in an interval of 5°C. These classes are further divided based on 'Day Type' i.e., Sunny

Day or Humid Day. This has helped in understanding the effect of temperature on power generation from solar panel.

7.2 Test Cases

Test cases are the documents that cover all possible scenarios for specific requirement. It contains Test Case ID, Condition (Input), Day Type, Expected Output, Actual Output and Result. Test Cases based on Equivalence Class Testing are depicted in Table 7.1. For each temperature range condition, two test cases are written (each for Sunny and Humid day type). In order to perform the test, the solar panel is exposed to 9 hours of sunlight between 8 AM to 5 PM. The solar panel used for the test is of small size which generates less power.

Table 7.1: Equivalence Class Testing

TEST CASE ID	CONDITION (Temperature Range)	DAY TYPE	EXPECTED OUTPUT (in kWh)	ACTUAL OUTPUT (in kWh)	RESULT
1	40°C – 45°C	Sunny	0.50	0.40	Fail
2		Humid	0.40	0.32	Fail
3	35°C – 40°C	Sunny	0.45	0.42	Pass
4		Humid	0.35	0.32	Pass
5	30°C – 35°C	Sunny	0.40	0.39	Pass
6		Humid	0.27	0.30	Pass
7	25°C – 30°C	Sunny	0.20	0.20	Pass

8		Humid	0.15	0.10	Pass

Table 7.2 depicts the failure analysis of solar panel wherein the 'Condition' column shows sensor failure conditions (input) for each of the sensors used and their impact on power generation (output). Extra two test cases are written to show the conditions wherein all the sensors fail to operate and, all the sensors are working fine.

Table 7.2: Failure Analysis

TEST CASE ID	CONDITION (Sensor Failure)	DAY TYPE	EXPECTED OUTPUT (in kWh)	ACTUAL OUTPUT (in kWh)	RESULT
1	Current Sensor Fails	Sunny	No Result	No Result	Pass
2		Humid	No Result	No Result	Pass
3	Humidity and Temperature Sensors Fail	Sunny	0.45	0.45	Pass
4		Humid	0.35	0.32	Pass
5	Light Intensity Sensor Fails	Sunny	0.45	0.46	Pass
6		Humid	0.35	0.33	Pass
7	Voltage Sensor Fails	Sunny	No Result	No Result	Pass
8		Humid	No Result	No Result	Pass

9	All Sensors Fail	Sunny	No Result	No Result	Pass
10		Humid	No Result	No Result	Pass
11	All Sensors Working	Sunny	0.45	0.45	Pass
12		Humid	0.35	0.36	Pass

7.3 Performance Evaluation

Performance evaluation is necessary to evaluate how effective the proposed system is. The solar panel is tested in various conditions like sunny day, rainy day, cloudy day etc. to evaluate it against several test cases so that the credibility of the system can be tested. The power generation is displayed correspondingly against each test case to facilitate analysis of solar panel.

7.4 Summary

A brief description about testing and validation techniques is discussed with their applicability in project. Different set of test cases are shown for which the corresponding results are displayed. Performance evaluation is done by evaluating the model for various test cases and calculating the percentage of positive results. It is necessary to find how effective the proposed system is.

CHAPTER 8

RESULTS AND DISCUSSIONS

8.1 Snapshots and Description

Snapshots are shown as results for better understanding of the project and for the end user to visualize various processes. Various snapshots of different app screens show how end user can see the output produced by solar panel.

The dashboard page, as shown in figure 8.1, shows the live data readings of the sensors regarding panel temperature, energy production, light intensity, humidity, panel health, voltage and current. This dashboard also indicates total saving done, for the day, on power generation.

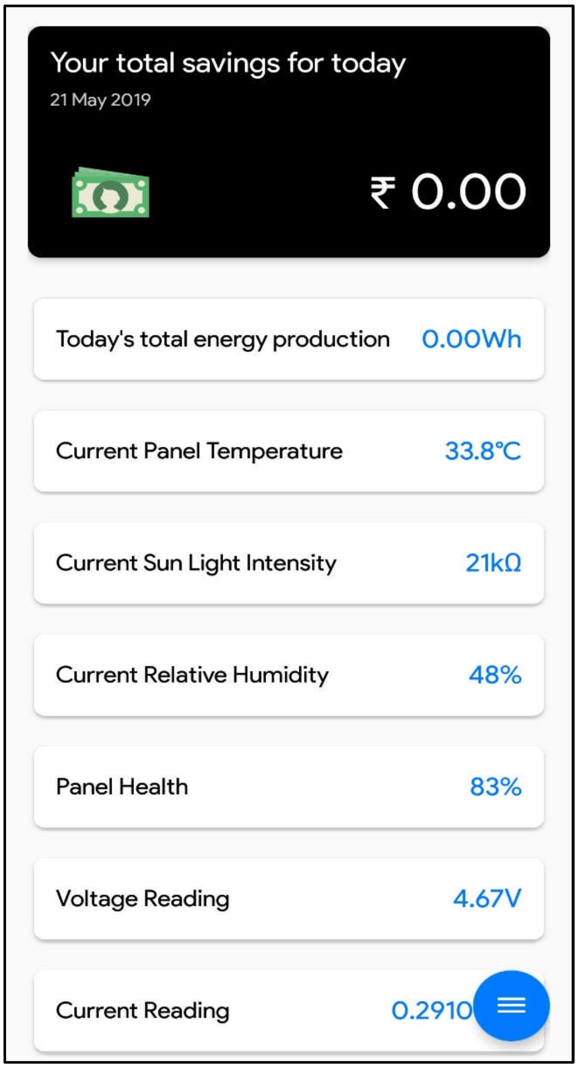


Figure 8.1: Dashboard

Panel health comparison is shown in Figure 8.2. The panel health can be categorized as:

1. Critical – less than 60%
2. Good – between 60% to 80%
3. Very Good – above 80%

Panel health analysis is very important to keep a check on solar panel condition regularly. It has already been evidenced that the health of solar panel reduces with an annual decrease of 1% to 2%. A sudden decrease in panel health percentage shows that the solar panel may have some serious issue which need to be addressed as early as possible. Low panel health adversely affects the power generation.

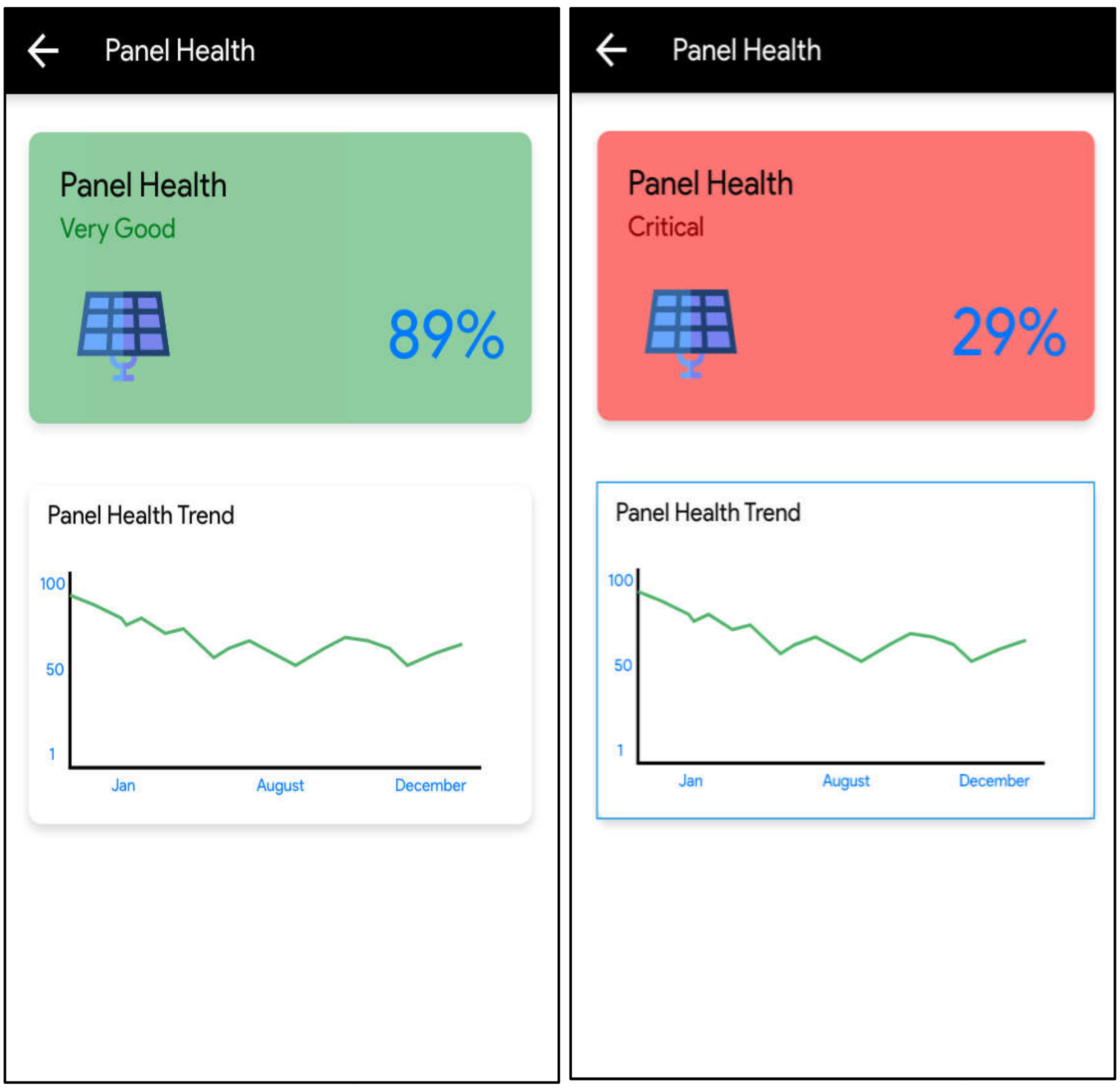


Figure 8.2: Panel Health Comparison

Trend analysis graphs are shown in the Trends section as depicted in Figure 8.3. It consists of voltage, current, temperature, light intensity as well as humidity trends.

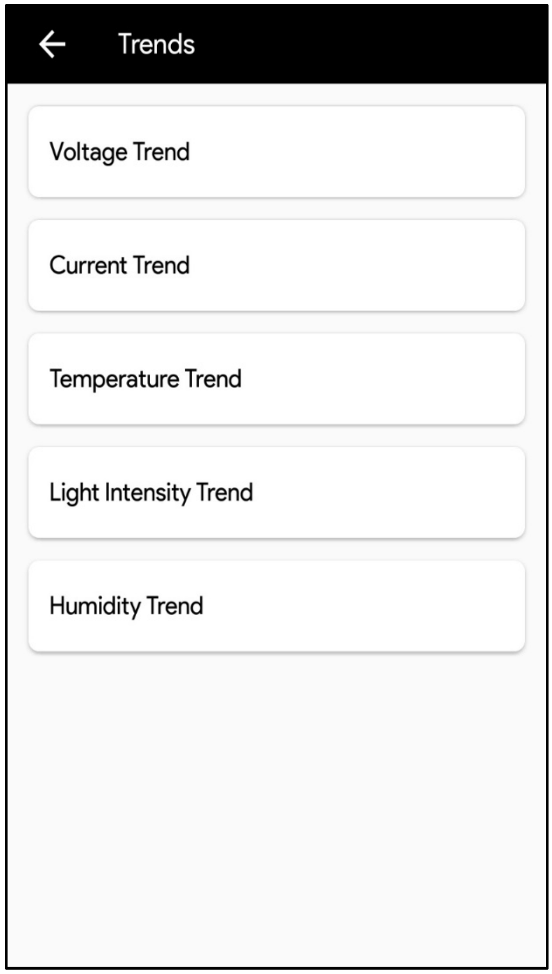


Figure 8.3: Trends

Trend analysis is very useful for comparative analysis of solar data readings in order to measure the overall performance of solar panel over a period of time and which helps the user to study more about effect of all the factors on solar energy production and behavior of solar panel over a period of time.

The graphs drawn to show the trends of voltage, current, temperature, light intensity and humidity consists of live data i.e., the current and updated data is shown. These graphs keep on updating after every five seconds. This can be easily identified on the graph as black dots, depicting the time at which the new value is shown on the graph. On all the plotted graphs, the x-axis contains timestamp at which data is being read whereas, the y-axis contains the readings of various sensors being used.

Figure 8.4 and Figure 8.5 illustrates the trend analysis of humidity and temperature respectively, in the form of graphs. It can be easily figured out in the Figure 8.4 that the humidity reading has exceeded 90% at one point, depicting very humid weather and, it has gone down below 30% too. Similarly, maximum panel temperature recorded, as shown in Figure 8.5, is around 48 °C.

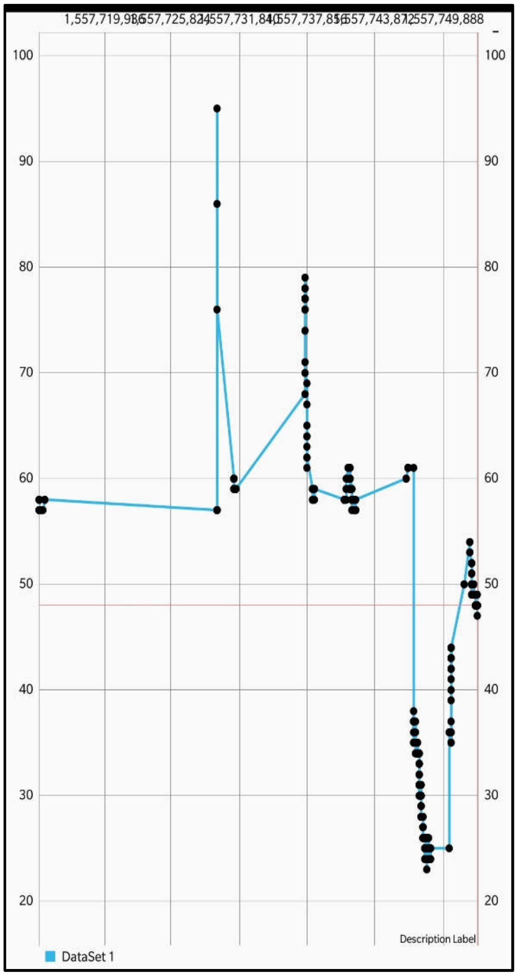


Figure 8.4: Humidity Trend

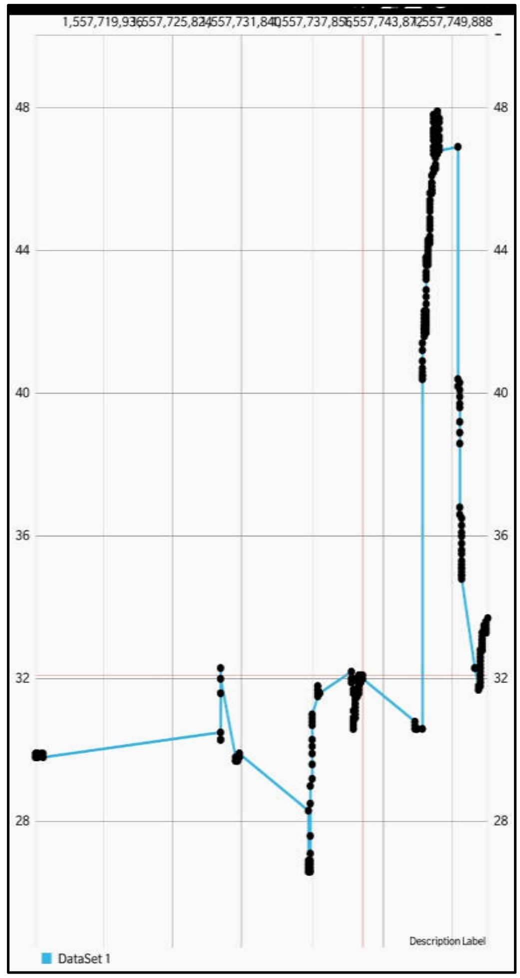


Figure 8.5: Temperature Trend

8.2 Summary

This chapter deals with the result of the project work done. It includes the snapshots of the outcomes of the android application developed for the purpose of monitoring and reporting of solar panel data collected in a well-organized manner. Results, basically, refer to output or end point that comes as a result of the completion of the activities and or processes that have been performed as part of the project. Trends are represented graphically to study about the factors affecting the power generation by the solar panel. This enhances ease in monitoring solar panels.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

9.1 Conclusion

The significance of solar energy in today's world is unprecedented. The automation and intellectualization of solar power plant monitoring will enhance future decision-making for these plants. Smart monitoring of a solar panel can be done by integrating IOT technologies with handy monitoring devices like mobile phones. Different sensors are installed on microcontroller board and are encoded using Arduino IDE to generate proper readings. Arduino IDE is then connected to the firebase for real time data transfer and storage. These data are then received by Android application developed using Android Studio. Once a user opens this application, he/she is presented with various options which he can choose to monitor data produced by solar panel. The data fetched from firebase can be compared with historical data and a graph showing comparative results is displayed on screen which user can analyze for performance evaluation of the panel.

9.2 Future Enhancement

Future enhancement of the project includes:

- Better hardware architecture to reduce the installation time to solar panel.
- Use of additional sensors such as GPS sensor to track location of solar panel.
- Use of sensors in battery to show the remaining power of battery in percentage.
- Weekly and monthly charts to show the power generation and sensor readings for better understanding of performance of solar panel.
- Alarming notifications in the application, as and when the system finds faulty operation of solar panel.
- Use of artificial intelligence and machine learning to automate the task of finding errors and faults in sensors and solar panel.

REFERENCES

[1] Brown, Eric (13 September 2016). "Who Needs the Internet of Things?". Linux.com.Retrieved 23 October 2016

[2] Brown, Eric (20 September 2016). "21 Open Source Projects for IoT". Linux.com.Retrieved 23 October 2016

[3] Hendricks, Drew. "The Trouble with the Internet of Things".London Datastore.Greater London Authority. Retrieved 10 August 2015

[4] Achim Woyte, Mauricio Richter, David Moser, Stefan Mau, Nils Reich, Ulrike Jahn"Monitoring of Photovoltaic Systems: Good Practices and Systematic Analysis" 28thEuropean PV Solar

[5] Xi Chen, Limin Sun; Hongsong Zhu; Yan Zhen; Hongbin Chen, "Application ofInternet of Things in Power-Line Monitoring",2012International Conference onCyber-Enabled Distributed Computingand Knowledge Discovery (CyberC),978-1-4673-2624-7

[6] Byeongkwan Kang, Sunghoi Park, Tacklim Lee, and Sehyun Park, "IoT-based Monitoring System usingTri-level Context Making Modelfor Smart Home Services",2015 IEEE International Conference onConsumer Electronics (ICCE)

[7] Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. IEEE InternetThings J. 2014, 1, 22–32

[8] Medagliani, P.; Leguay, J.; Duda, A.; Rousseau, F.; Duquennoy, S.; Raza, S.; Ferrari, G.; Gonizzi, P.; Cirani, S.; Veltri, L.; et al. Internet of Things Applications—From Research and Innovation to Market Deployment; The River PublisherSeries in Communication: Delft, The Netherlands, 2014; pp. 287–313

[9]Elmangoush, A.; Alhazmi, A.; Magedanz, T.; Schuch, W.; Estevez, C.; Ehijo, A.; Wu, J.; Nguyen, T.; Ventura, N.;Mwangama, J.; et al. Towards Unified Smart City Communication Platforms. In Proceedings of the Workshopon Research in Information Systems and Technologies, Chillán, Chile, 16 October 2015

[10] Badri Chatterjee. "Maharashtra leads with most rooftop solar installations in India"Hindustan Times, Mumbai,23 July 2018

[11] Suciu Constantin, Florin Moldoveanu, Radu Campeanu, IoanaBaciu,Sorin Mihai Grigorescu, BogdanCarstea, Vlad Voinea, "GPRS BasedSystem for Atmospheric Pollution Monitoring and Warning", 1-4244-0361-2006 IEEE

[12] Soham Adhya, Dipak Saha, Abhijit Das, Joydip Jana, HiranmaySaha. "An IoT Based Smart Solar Photovoltaic Remote Monitoring and Control unit." 2016 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC)

[13] Putta Sindhuja, M. S. Balamurugan. "Smart Power Monitoring and Control Systemthrough Internet of things usingCloud Data Storage." Indian Journal of Science and Technology, Vol 8(19), August 2015

[14] Birendrakumar Sahani, Tejashree Ravi, Akibjaved Tamboli, Ranjeet Pisal. "IoT Based Smart Energy Meter." International Research Journal of Engineering and Technology (IRJET), Vol 4(4), April 2017

[15] Suprita Patil, M. Vijayalashmi, RakeshTapaskar. "Solar Energy Monitoring System using IoT." Indian Journal of Scientific Research(IJSR), Vol 15(2): 149-155, 2017

[16]Abhiraj Prashant Hiwale, Deepak Sudam Gaikwad, Akshay Ashok Dongare, Prathmesh Chandrakant Mhatre. "IoT based Smart Energy Monitoring." International Research Journal of Engineering and Technology (IRJET), Vol 5(3), March 2018