SMART HOME WITH SMART SECURITY

A Project Report

Submitted by:

PRAKASH J (2018504042)

BARANIDHARAN S (2018504517)

ROHITH VARON S S (2018504590)

VELMURUGAN M (2018504626)

In partial fulfilment for the award

of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

DEPARTMENT OF ELECTRONICS ENGINEERING

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600044

JUNE 2022

ACKNOWLEDGEMENT

We consider it as a privilege and our primary duty to express our gratitude and respect to all those who guided and inspired us in the successful completion of the project.

We owe solemn gratitude to **Dr J. PRAKASH**, Dean, Madras Institute of Technology, for having given consent to carry out the project work at MIT Campus, Anna University.

We wish to express our sincere appreciation and gratitude to **Dr. M. GANESH MADHAN**, Professor and Head of the Department of Electronics Engineering, who has encouraged and motivated us in our endeavors.

We are extremely grateful to our project guide **Dr. A.VIJI**, Teaching Fellow, Department of Electronics Engineering, for his timely and thoughtful guidance and encouragement for the completion of the project.

We sincerely thank all our panel members **Dr. S. P. JOY VASANTHA RANI**, **Dr. A. VIJI**, and **Dr. N. JAGADEESH KUMAR** for their valuable suggestions. We also thank all the teaching and non-teaching staff members of the Department of Electronics Engineering for their support in all aspects.

PRAKASH J (2018504042)

BARANIDHARAN S (2018504517)

ROHITH VARON S S (2018504590)

VELMURUGAN M (2018504626)

BONAFIDE CERTIFICATE

Certified that this Project Report titled "**SMART HOME WITH SMART SECURITY**" is the bonafide work of **PRAKASH J** (2018504042), **BARANIDHARAN S** (2018504517), **ROHITH VARON S S** (2018504590) and **VELMURUGAN M** (2018504626) who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. M. GANESH MADHAN
Professor and Head,
Department of Electronics Engineering,
Madras Institute of Technology,
Anna University,
Chennai - 600 044.

SIGNATURE

Dr. A. VIJI
Teaching Fellow, Guide
Department of Electronics Engineering,
Madras Institute of Technology,
Anna University,
Chennai - 600 044.

ABSTRACT

This project presents the overall design of Home Automation System (HAS) with low cost and wireless system. An IOT based home automation system is developed to control various components via internet or be automatically programmed to operate from ambient conditions. The Internet of Things has developed tremendously in recent years, and it now has a wide range of applications in the smart home, Internet of Vehicles, and Industrial Internet of Things. There may be vulnerabilities present in the perception layer, transport layer, and application layer of most developing smart devices which have very simple structures. Currently, the majority of security analysis frameworks for IoT devices necessitate the use of extra high-performance devices. Simultaneously, developing "mining" malware and other attack methods that directly pillage the device's computational resources have garnered less attention. This work focused to develop and implement a smart home security analysis system in response to the aforementioned issues. The system can detect and fight against contactless attacks that smart homes. The solution can better reconcile the disparity between smart home network security requirements and device performance constraints.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SYMBOL	DESCRIPTION
MCU	Microcontroller Unit
VNC	Virtual Network Computing
WMS	Warehouse Management System
GPIO	General Purpose Input and Output
BMP	Barometric Pressure Sensor
LED	Light Emitting Diode
LDR	Light Dependent Resistor
ADC	Analog to Digital Convertor
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
GND	Ground
V	Voltage
UART	Universal Asynchronous ReceiverTransmitter

CHAPTER 1

INTRODUCTION

The Internet of Things (IoT), as well as its developments and characteristics, are discussed in this chapter. Our project's Internet of Things-based home automation system and security features are also presented.

1.1 INTRODUCTION TO IOT

With the introduction of high-speed internet, more and more people all around the world are becoming linked. The Internet of Things takes this a step further by connecting not only people but also electronic gadgets that can communicate with one another. With the decreasing cost of WIFI-enabled gadgets, this trend will only gain traction. The primary idea behind the Internet of Things is to link numerous electronic devices over a network, then obtain data from these devices (sensors), which can be dispersed in any way, and upload it to any cloud service where the data can be analyzed and processed. These data can be used in the cloud service to warn users in many ways, for as using a buzzer or sending them an email or SMS.As previously said, IoT allows for not just human-human connection, but also human-device and device-device interaction. In the case of energy, for example, IoT is being used to develop Smart Grids that can detect and respond to changes in local and global energy use, which will be an important element of any country's energy policy. Beyond the aforementioned energy example, IoT can have a significant impact in a variety of domains, such as Smart Homes, which use IoT to increase the degree of automation. Wearable technology such as smart watches and fitness bands;

connected healthcare is one of the most promising areas of IoT. The market is on the verge of exploding, with companies like Intel, Rockwell Automation, Siemens, Cisco, and General Electric leading the charge. Analysts predict that by 2020, there will be 26 billion connected devices, more than four for every human on the planet, and the industry will generate $19 trillion in cost savings and profits, with companies like Samsung and Google leading the charge. The Internet of Things (IoT) is the future technology that will bring the entire planet together in one location. All objects, things, and sensors can be connected to share data collected in multiple locations and to process/analyze that data in order to coordinate applications such as traffic signaling, mobile health monitoring in medical applications, and industrial safety assurance methods, among others. According to analysts, 50 billion dollars will be needed. By 2025, all objects in the IoT will be connected. IoT provides a diverse set of device connectivity options, as well as a variety of protocols and application features, to enable complete machine-to-machine communication.

1.2 HOME AUTOMATION SYSTEM

The project's major goal is to address the inadequacies of home security systems by delivering current situation information while the owner is away from the residence. This article focuses on a system that incorporates a camera module and provides home security, as well as components of Home Automation that rely on the Internet of Things to work. The Android software essentially transforms your smartphone into a remote control for all of your home appliances. If movement is detected at the house's entrance, a notification is issued that includes a real-time photo of the entrance. The owner of the residence will get this notification via the internet, allowing the app to send a notification. In the event of an intrusion, the owner can activate the appliances, such as opening the door if the individual is a visitor. The user can use this system to automate the turning on of lights, fans, and air conditioners, among other things. The user can utilize the Internet to access the entire IoT system from anywhere. However, the microcontroller must always be connected to the Internet. The Raspberry Pi is a tiny computer that serves as the system's server. The Raspberry Pi system works in the same way as a computer but with a much smaller setup. It includes GPIO pins and USB connectors, as well as a camera module port. Simple programs can be used to turn these pins on and off.

1.3 APPLICATIONS:

Will smart kitchen gadgets help you become a better cook? Maybe. Smart refrigerators, such as LG's Smart ThinQ, let you scan grocery store receipts and keep track of your purchases, as well as notify you when an item is ready to expire. More impressively, it offers meals based on the contents of your refrigerator and alerts you when goods need to be replaced. Smart ovens connect to your smartphone and automatically warm to the proper temperature using a recipe from your recipe collection. You may cut off the electrical supply to unused equipment and save money by automating your kitchen appliances and making them available from your smart device as well as lowering your energy consumption and costs Given the number of appliances owned by the average family, this may save a lot of money over time.

1.4 PROJECT MOTIVATION

Home computerization results in a more intelligent home and is used to provide a better and more advantageous way of life. The beauty of a home computerization framework is that it is extremely adaptable and flexible, and its capabilities are only limited by our imagination. With IoT turmoil on the horizon, it's past time to move toward a prototype that can be chosen in an infinite number of ways.

1.5 REPORT ORGANISATION:

Introduction to IoT, Home automation and security system, its scope and applications are discussed in this chapter 1, Literature surveys surveyed are explained in Chapter 2, Hardware specification are explained in Chapter 3, Face recognition system discussed in Chapter 4, software requirements will be dealt in Chapter 5. Results will be discussed in Chapters 6. Conclusions and Future scope will be covered.

CHAPTER 2
LITERATURE REVIEW

Various existing approaches for IoT based home automation with security features have been surveyed and discussed in this chapter

2.1 A SMART HOME AUTOMATION TECHNIQUE WITH RASPBERRY PI USING IOT.

AUTHOR: Vamsikrishna Patchava , Hari Babu Kandala , and P Ravi Babu

PUBLICATION: 2015 International Conference on Smart Sensors and Systems (IC-SSS)

DESCRIPTION: In this paper, we are presenting a proposed system for Smart Home Automation technique with Raspberry Pi using IoT and it is done by integrating cameras and motion sensors into a web application. Using this, we can control home appliances connected through a monitor-based internet. Raspberry Pi operates and controls motion sensors and video cameras for sensing and surveillance. For instance, it captures intruder's identity and detects its presence using simple Computer Vision Technique (CVT). Whenever motion is detected, the cameras will start recording and Raspberry Pi device alerts the owner through an SMS and alarm call.

2.2 SECURITY AND PRIVACY IN THE SMART HOME: A SURVEY OF ISSUES AND MITIGATION STRATEGIES

AUTHOR: Meral Korkmaz Kuyucu, Serif Bahtiyar, GǗkhanɗnce

PUBLICATON: 2019 4th International Conference on Computer Science and Engineering (UBMK)

DESCRIPTION: Internet of Things (IoT) devices are becoming more and more prominent in our everyday lives and have made their way into our homes. This paper surveys and synthesizes the literature on security and privacy issues that arise in a smart home. The paper continues to survey studies conducted to mitigate these flaws.

2.3 SMART HOME AUTOMATION USING MACHINE LEARNING ALGORITHMS

AUTHOR: John Jaihar, Neehal Lingayat, Patel Sapan Vijay Bhai , Gautam Venkatesh, K. P. Upla

PUBLICATION: 2020 International Conference for Emerging Technology (INCET)

DESCRIPTION: A home automation system controls lighting, temperature, multimedia systems, and appliances. Since these devices and sensors are connected to common infrastructure, they form the Internet of Things. A home automation system links multiple controllable devices to a centralized server. These devices have a user interface for controlling and monitoring, which can be accessed by using a tablet or a mobile application, which can be accessed remotely as well. Ideally, anything that can be connected to a network can be automated and controlled remotely. Smart homes must be artificially intelligent systems that need to adapt themselves based on user actions and surroundings. These systems need to carefully analyze the user needs and the conditions of the surroundings in order to predict future actions and also minimizes user interaction. Traditional home automation systems that provide only remote access and control are not that effective in terms of being `smart', so in this paper we put forward the use of concepts of different machine learning algorithms along with computer vision to shape together a smart learning automated system that controls lighting, sound and other devices based on the user's emotion.

2.4 SECURITY AND PRIVACY ISSUES FOR AN IOT BASED SMART HOME

AUTHOR: Dimitris Geneiatakis, Ioannis Kounelis, Ricardo Neisse, Igor Nai-Fovino Gary Steri, and Gianmarco Baldini

PUBLICATION: 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)

DESCRIPTION: Internet of Things (IoT) can support numerous applications and services in various domains, such as smart cities and smart homes. IoT smart objects interact with other components e.g., proxies, mobile devices, and data collectors, for management, data sharing and other activities in the context of the provided service. Though such components contribute to address various societal challenges and provide new advanced services for users, their limited processing capabilities make them vulnerable to well-known security and privacy threats. Until now various research works have studied security and privacy in IoT, validating this claim.

However, to the best of our knowledge literature lacks research focusing on security and privacy flaws introduced in IoT through interactions among different devices supporting a smart home architecture. In particular, we set up the scene for a security and privacy threat analysis for a typical smart home architecture using off the shelf components. To do so, we employ a smart home IoT architecture that enables users to interact with it through various devices that support smart house management, and we analyze different scenarios to identify possible security and privacy issues for users.

2.5 IOT BASED HOME AUTOMATION SYSTEM

AUTHOR: Sivapriyan R, K Manisha Rao, Harijyoti M

PUBLICATION: 2020 Fourth International Conference on Inventive Systems and Control (ICISC)

DESCRIPTION: Home automation system is gain if a significant research attention in recent years. It helps us in leading a comfortable life and quality of lifestyle is gradually uplifted. The different methodologies used in this system have been discussed. In modern days, a smartphone having an android application is used to supervise and control the appliances present in the home automation system. In this paper different types of communication methodologies such as GSM, IoT, Wi-Fi, and Bluetooth are reviewed. The pros and cons of these techniques along with their features have been presented. Based on this paper the user can choose the best suitable methodology depending upon their personal needs and specifications for implementing an efficient automation system.

2.6 HOME AUTOMATION USING INTERNET OF THINGS

AUTHOR: Shopan dey, Ayon Roy, Sandip das

PUBLICATION: 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)

DESCRIPTION: In this era of digitization and automation, the life of human beings is getting simpler as almost everything is automatic, replacing the old manual systems. Nowadays humans have made internet an integral part of their everyday life without which they are helpless. Internet of things (IoT) provides a platform that allows devices to connect, sensed and controlled remotely across a network infrastructure. In this paper we focus on home automation using smart phone and computer. The system designed is economical and can be expanded as it allows connection and controlling of a number of different devices.

CHAPTER 3
METHODOLOGY

This chapter discusses the overview of the proposed approach of IoT based home automation system. It also gives the detailed insight of each module in the proposed system and the need of the module.

3.1 SYSTEM DESIGN

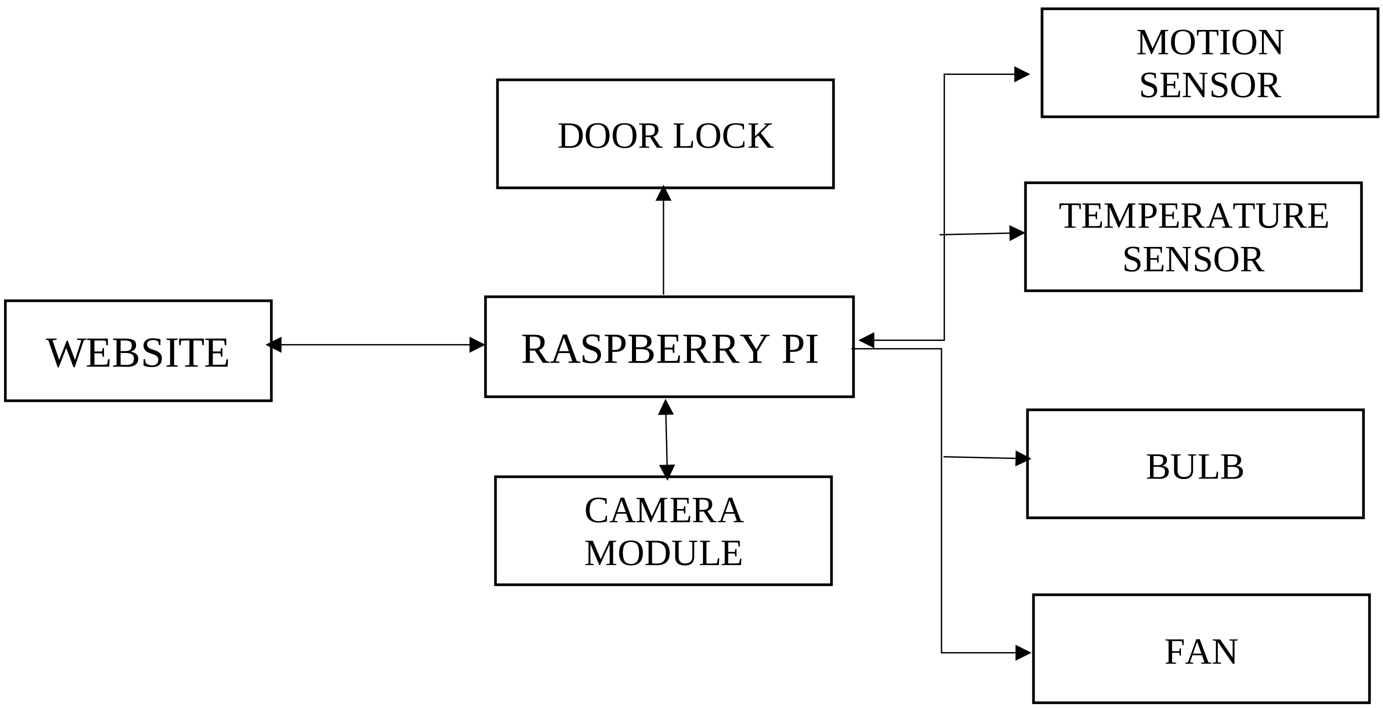


FIGURE 3.1 BLOCK DIAGRAM OF SMART HOME AUTOMATION SYSTEM

The smart home automation system consists of Raspberry pi, temperature sensor, motion sensor, bulb, fan, camera, door lock and relay which is shown in figure 3.2 and 3.3. The user first logs in to our website using their default credentials. A socket is running on the server (Raspberry Pi), which is always open and waiting for a request from the user. When a user presses the login button in the website, a client socket is generated and a connection between the Raspberry Pi and the Android device is established. The data is decrypted on the Raspberry Pi's side. The decrypted data is compared to the entries in the Raspberry Pi's memory. If the information is correct, the device receives a response, which initiates a new activity. With a simple on/off button UI, this new activity may be used to control any home appliances which is shown in figure 3.1. The Raspberry Pi handles the queries on the server side. of the data takes place. The decrypted data is compared to the entries in the Raspberry Pi's memory.

3.2 IMPLEMENTATION SETUP

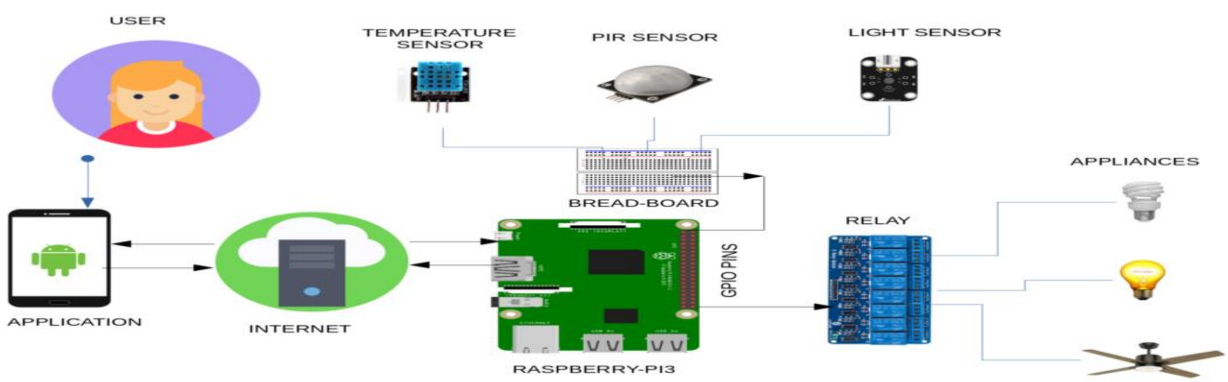


FIGURE 3.2 CONNECTION IS MADE AS PER THE DIAGRAM

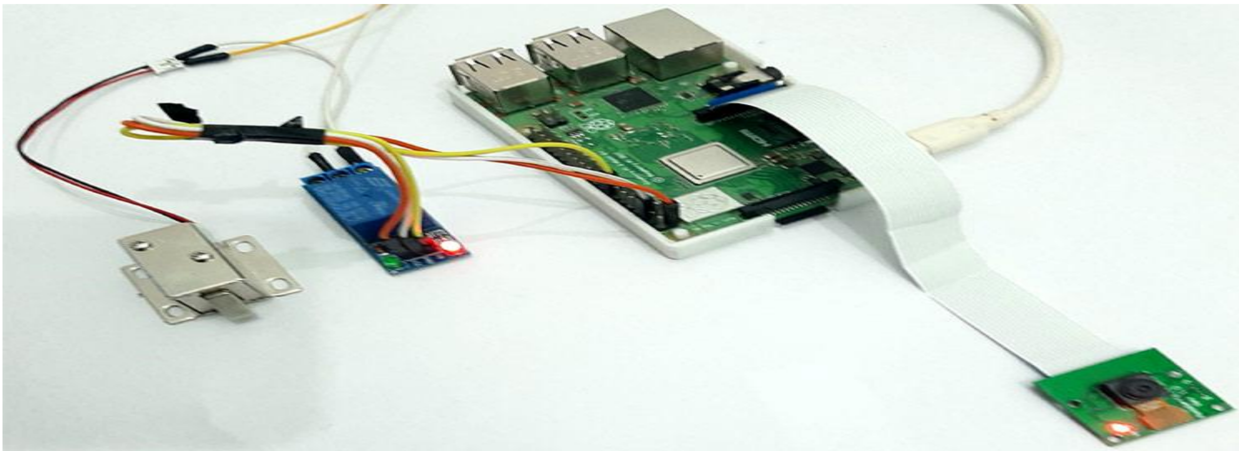


FIGURE 3.3 FACE RECOGNITION SYSTEM

3.3 RASPBERRY PI

A Raspberry Pi is a small computer that was inspired by the BBC Micro in 1981 and was built for portability. Ebben Upton designed the device in order to develop a compact and affordable device that would assist students enhance their programming skills and hardware understanding. It was suited for a variety of applications because to its compact size and low cost. As a result, numerous clients soon accepted it. The Raspberry Pi which is shown in figure 3.4 is a complete Linux computer that offers all of its features while consuming very little power.

FIGURE 3.4 RASPBERRY PI

3.4 PARTS OF RASPBERRY PI:

The functions of each part of raspberry are mentioned below shown in figure 3.5.

1. GPIO. GPIO is arguably the most important feature of the Raspberry Pi and is the equivalent of GPIO pins on the Arduino.
2. DSI Display Port - The DSI display port allows the Raspberry Pi to connect to a serial display similar to those used in tablets.
3. CSI Camera Port- The CSI camera port is a connector that allows the Raspberry Pi to connect to a Raspberry Pi camera module.
4. MicroSD Slot- This slot is used to house the microSD card that holds the Raspberry Pi operating system.
5. HDMI Port- These slots are used to connect the Pi to an HDMI screen, USB devices such as mice and keyboards, and to an ethernet connection for internet access.
6. USB Port- Power to the Raspberry Pi can be provided using either a micro USB lead to the micro USB connector (recommended) or 5V can be directly fed into the 5V GPIO pin.

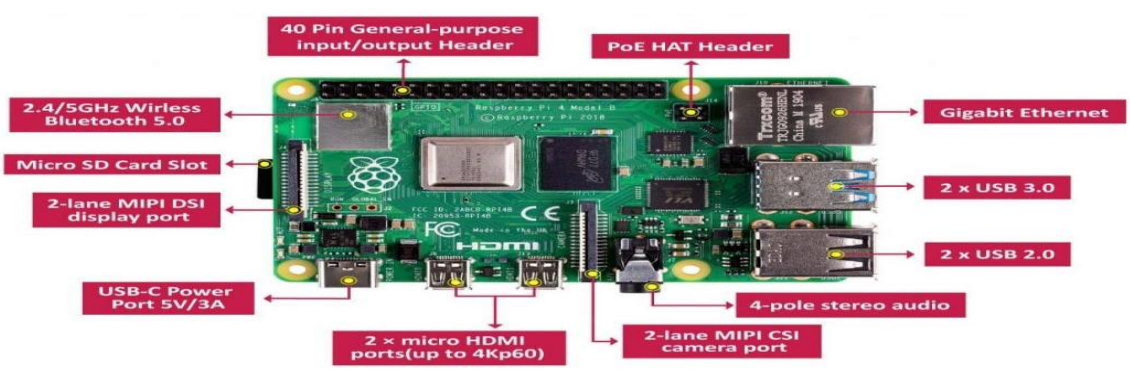


FIGURE 3.5 PARTS OF RASPBERRY PI

3.5 PIN CONFIGURATION OF RASPBERRY PI

The pin configuration of raspberry pi is given below

1. **Vin**: T5v pins and two 3v3 pins used for providing power supply, where processor works on 3.3v.
2. **Ground:** Having 8 ground pins which are un-configurable.
3. **GPIO:**There are 26 input-output pins which will be used as input or output based on programming.
4. **PWM:** In software PWM are available for all pins but in hardware PWM is available for GPIO12, GPIO13, GPIO18, and GPIO19.
5. **2 SPI bus:** These pins are used for SPI communication the pins which are used for SPI is MISO, MOSI, SCLK, CEO, and CE1.
6. **I2C:** These pins are used for communication in which DATA and CLOCK pins
7. **Tx and R x:** These pins are used for UART communication.

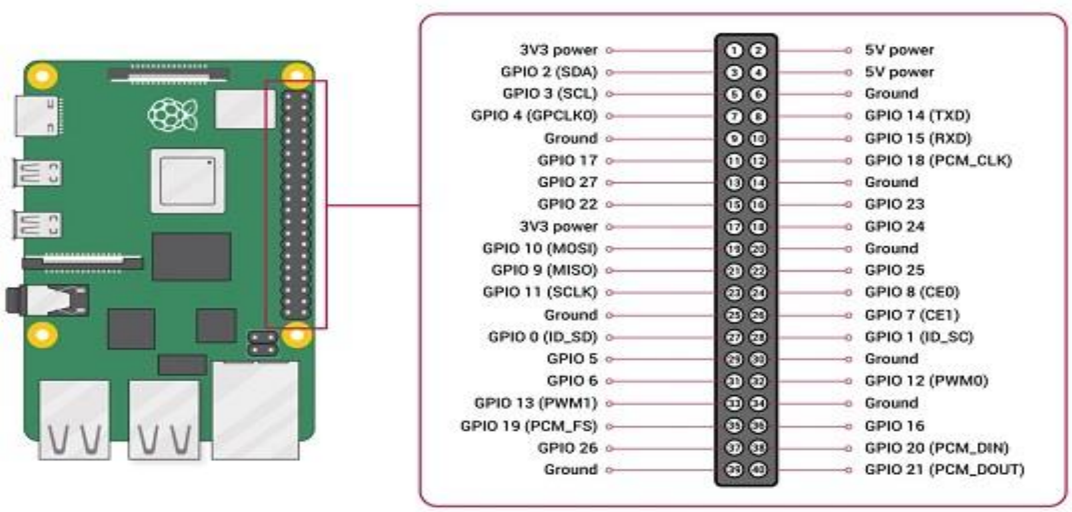


FIGURE 3.6 PIN CONFIGURATION OF RASPBERRY PI

3.6 TEMPERATURE AND HUMIDITY SENSOR

Humidity refers to the amount of water vapor in the air. The amount of humidity in the air has an impact on a variety of physical, chemical, and biological processes. Humidity in industrial settings can have an impact on product costs, as well as staff health and safety. As a result, humidity measurement is critical in semiconductor and control system industries. The amount of moisture present in a gas, which can be a mixture of water vapor, nitrogen, argon, or pure gas, is determined by humidity measurement. Humidity sensors are classified into two categories based on the measuring units they use. A relative humidity sensor and an absolute humidity sensor are the two types of humidity sensors. The DHT11 which is shown in figure 3.7 is a temperature and humidity sensor that is digital.

The DHT11 is a low-cost digital temperature and humidity sensor. This sensor may be simply connected to any microcontroller, such as an Arduino or a Raspberry Pi, to measure humidity and temperature in real time. The DHT11 humidity and temperature sensor comes in two versions: a sensor and a module. The pull-up resistor and a power-on LED distinguish this sensor from the module. A relative humidity sensor is the DHT11 shown in figure 3.7. This sensor employs a thermistor and a capacitive humidity sensor to measure the ambient air.

FIGURE 3.7 DHT11 SENSOR

A capacitive humidity detecting element and a thermistor for temperature detection make up the DHT11 sensor which is shown in figure 2.7. A moisture-holding substrate serves as a dielectric between the two electrodes of the humidity sensor capacitor. With changes in humidity levels, the capacitance value changes. The IC measures, processes, and converts the resistance values into digital form. The specifications of DHT11 sensor

1. Operating Voltage: 3.5V to 5.5V
2. Operating current: 0.3mA (measuring) 60uA (standby)
3. Output: Serial data
4. Temperature Range: 0°C to 50°C
5. Humidity Range: 20% to 90%

TABLE 3.1 PIN CONFIGURATION OF DHT11 SENSOR

S.NO	PIN NAME	PIN DESCRIPTION
1.	Vcc	Power supply 3.5 to 5.5V
2.	Data	Output pins
3.	NC	No connection
4.	Ground	Connected to the ground

3.7 PIR SENSOR

Pyroelectric Infra-red (PIR) sensors have been extensively used in indoor and outdoor applications as they are low cost, easy to use and widely available. PIR sensors respond to IR radiating objects moving in its viewing range. The current sensors give an output of logical one when they detect a hot object's motion and a logical zero when there is no moving hot object. In this method, only moving objects can be detected and the rate of false alarm is high.

A sensor that detects infrared light emitted by objects. PIR sensors are typically utilized in motion detectors that utilize PIR sensors. It's also employed in security alarm systems and automatic lighting systems. The figure below depicts a typical pin layout for the PIR sensor, which is easy to understand. Three pins make up the PIR sensor.

1. Pin1 corresponds to the drain terminal of the device, which connected to the positive supply 5V DC.
2. Pin2 corresponds to the source terminal of the device, which connects to the ground terminal via a 100K or 47K resistor. The Pin2 is the output pin of the sensor. The pin 2 of the sensor carries the detected IR signal to an amplifier from the
3. Pin3 of the sensor connected to the ground.

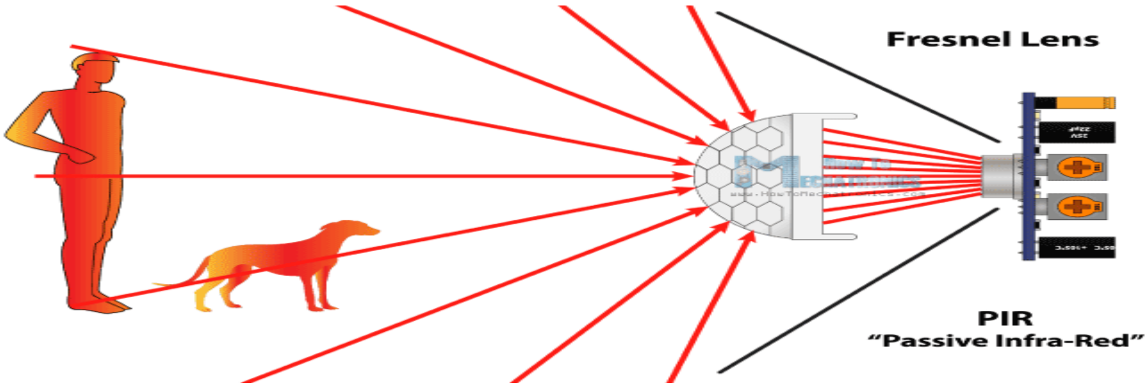


FIGURE 3.8 DETECTION OF PERSON USING PIR SENSOR

PIR sensors, in general, can detect animal/human movement within a specified range. PIR consists of a pyroelectric sensor that can detect various quantities of infrared light. The detector does not emit energy; instead, it passively receives it**.**

It is capable of detecting infrared radiation in the environment. Focusing on the optical system causes the pyroelectric device to generate a sudden electrical signal once infrared radiation from the human body particle with temperature is present. Simply put, when a human or animal walks by, the first slot of the PIR sensor is intercepted. As a result, the two bisects have a positive differential change. The sensor generates a negative differential shift between the two bisects when a human body departs the sensing area which is shown in figure 3.8.

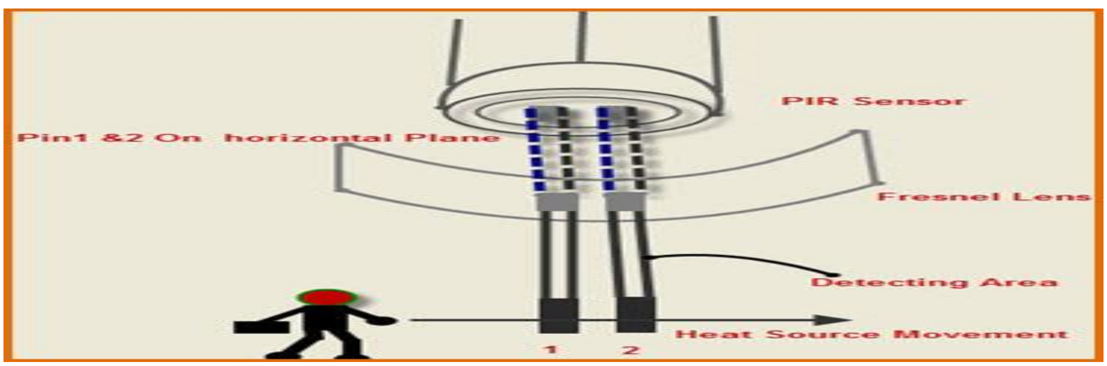


FIGURE 3.9 INTERNAL FUNCTION OF PIR SENSOR

The passive infrared sensor does not emit any energy into the environment. To sound an alarm, it receives infrared radiation from the human body. Anything that has a temperature emits infrared photons to the outside world. The human body's surface temperature ranges from 36 to 27 degrees Celsius, with the majority of its radiant energy focused in the wavelength range of 8 to 12 microns.

Infrared detectors (infrared probes) and alarm control sections are two types of passive infrared alarms. Pyroelectric detectors are the most common infrared detectors. It is used as a sensor to convert infrared radiation from humans into electricity. If human infrared radiation is shone directly on the detector, it will, of course, produce a temperature change, resulting in the emission of a signal. However, the detection distance will not increase as a result of all of this. An optical system must be installed to gather infrared light in order to extend the detection distance of the detector. Infrared radiation is usually focused using a plastic optical reflection system or a plastic Fresnel lens.

The detector's lens collects infrared radiation energy from the human body through the garment and focuses it on the pyroelectric sensor in the detecting region. When the human body moves in this surveillance mode, it enters and exits a specific field of view in a predetermined order. Because the pyroelectric sensor detects the moving human body for a short time before losing sight of it, the temperature of the pyroelectric material is continually changing due to the infrared radiation of the human body. As a result, a comparable signal, the alarm signal, is output.

3.8 5V RELAY

A 5v relay which is shown in figure 3.10 is an automatic switch that is typically used to regulate a high-current utilising a low-current signal in an automatic control circuit. The relay signal's input voltage varies from 0 to 5V.

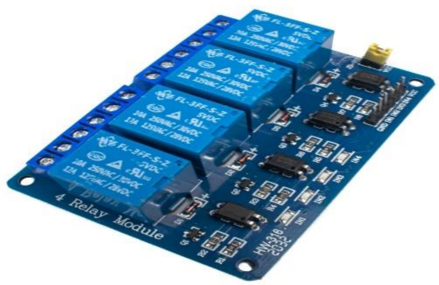


FIGURE 3.10 RELAY

The 5V relay's pin arrangement is given below. This relay has five pins, each of which has a different function as illustrated below and shown in figure 3.10.

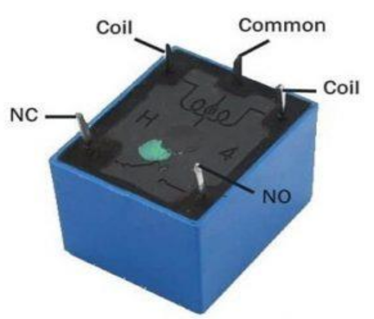


FIGURE 3.11 PIN CONFIGURATION OF RELAY

Pin1 (End 1): This pin is used to activate the relay; typically, one end of this pin is linked to 5V while the other end is connected to ground.

Pin2 (End 2): The Relay is activated using this pin.

Pin3 (Common (COM)): To activate the Load, connect this pin to the main terminal.

Pin4 (Normally Closed (NC)): The load's second terminal connects to either the NC or NO pins. This pin will be ON before the switch if it is linked to the load.

Pin5 (Normally Open (NO)): If the load's second terminal is connected to the NO pin, the load is shut off before the switch is turned on. Some the **features of the 5V relay** include the following.

1. Normal Voltage is 5V DC
2. Normal Current is 70Ma
3. AC load current Max is 10A at 250VAC or 125V AC
4. DC load current Max is 10A at 30V DC or 28V DC
5. It includes 5-pins & designed with plastic material
6. Operating time is 10msec
7. Release time is 5msec
8. Maximum switching is 300 operating per minute

The specifications of a 1- channel relay module include the following.

1. Voltage supply ranges from 3.75V – 6V
2. Quiescent current is 2Ma
3. Once the relay is active then the current is ~70mA
4. The highest contact voltage of a relay is 250VAC/30VDC

The current supply is used by the relay to open and close switch contacts. This is usually accomplished by using a coil to magnetize the switch contacts and draw them together once engaged. Once the coil is not strengthened, a spring separates them. There are primarily two advantages to employing this approach. The first is that the needed current for activating the relay is lower than the current consumed by relay contacts for switching. Another advantage is that both the contacts and the coil are galvanically separated, which means there is no electrical connection between them. Various applications of Relay modules are used in different applications which include the following.

1. Used in over voltage/under voltage protection system
2. Mains Switching
3. Speed control of motors through start-delta converters
4. Automatic electrical appliances
5. Electrical isolation in between high & low power sources
6. Lights
7. AC voltage load switching using less voltage DC
8. Delivery of Isolate
9. Home Automation Projects

3.9 SOLENOID DOORLOCK

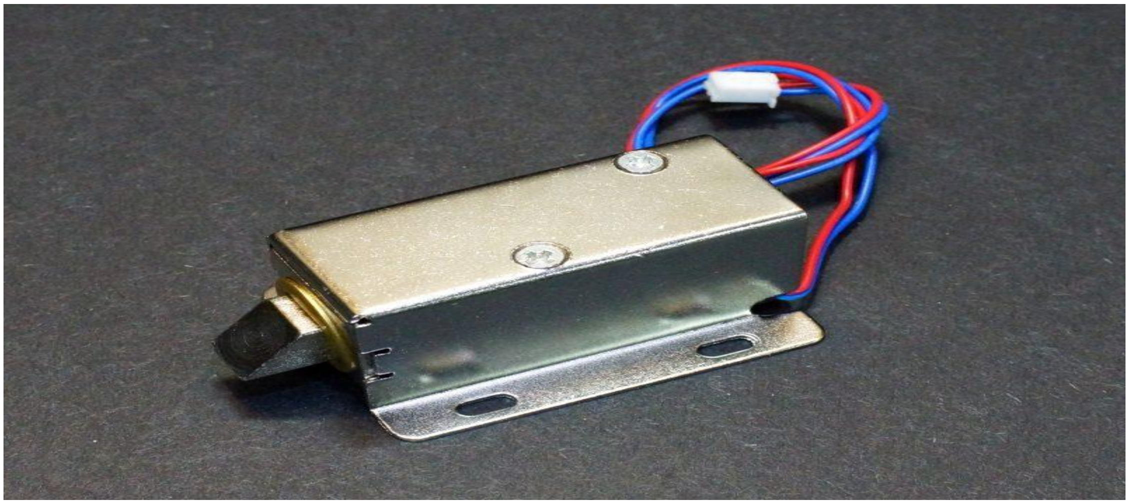


FIGURE 3.12 SOLENOID DOOR LOCK

The slug on this 12V solenoid lock which is shown in figure 3.12 has a slanted cut and a nice mounting bracket. It's essentially an electronic lock for a standard cabinet, safe, or door. The slug draws in when 9-12VDC is applied, so it doesn't stick out and the door can be opened. In this state, it does not utilize any electricity.

Automatic door lock systems, such as electronic door locks, are simple to install with the mounting board. This specific solenoid is beautiful and strong.

1. Operating voltage: 12VDC
2. Draws 650mA at 12V, 500 mA at 9V when activated
3. Designed for 1-10 seconds long activation time
4. Wire length: 222.25mm

The solenoid is based only on the "electromagnetism" hypothesis. If a metal core is placed inside the ring, the magnetic lines of flux focus on the increasing body of the coil's induction compared to the air core when current flows through the coil and a magnetic field is formed in it.A diagonally cut attachment and a matching mounting bracket are included with this lock. DC creates a magnetic field when power is delivered, which moves the slug inside and retains the door in the unlocked state. Some of the features of door lock system

1. Ultra-compact electric lock of excellent quality.
2. The body is made of iron.
3. Rustproof, long-lasting, safe, and simple to use.
4. Suction that sucks the iron tightly and locks the door.
5. The principle of electric magnetism is used in this lock.
6. With the mounting board, the electric door lock or other automatic door lock systems are simple to install.

CHAPTER 4

FACE RECOGNITION DOOR LOCK SYSTEM

4.1 INTRODUCTION

This chapter discusses the overview of the proposed approach for face recognition system. It also gives detailed insight of each module in the proposed system and the need of the module in the system.

4.2 SYSTEM DESIGN

Capture the image and convert the image into grey scale. Then detect the face from the image using face detector. The complete flow is shown in Figure 4.1.

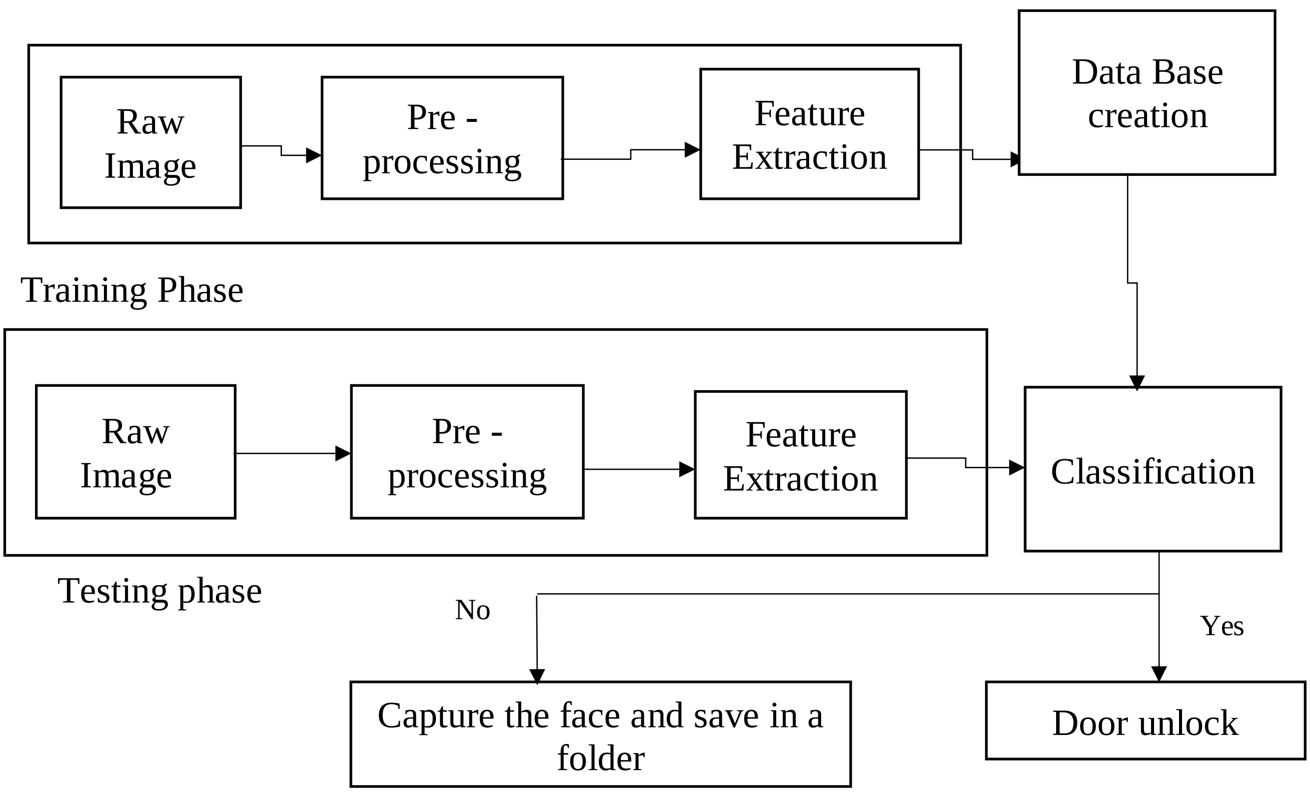


FIGURE 4.1 BLOCK DIAGRAM OF FACE RECOGNITION SYSTEM

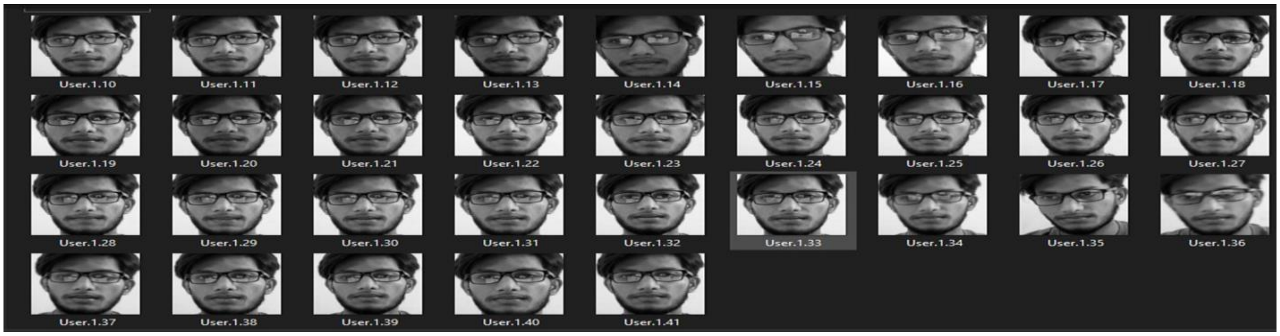


FIGURE 4.2 DATA BASE COLLECTION

Create a folder to store the data for authentication as shown in Figure 4.2. After preprocessing the data feature extraction is done using LBP operator. The LBP operator takes a local neighborhood around each pixel, thresholds the pixels of the neighborhood at the value of the central pixel and uses the resulting binary-valued image patch as a local image descriptor. It was originally defined for 3x3 neighborhoods, giving 8-bit integer LBP codes based on the eight pixels around the central one. Formally, the LBP operator takes the form where in this case runs over the 8 neighbors of the central pixel c, and are the gray-level values at c and n, and s (u) is 1 if and 0 otherwise. The LBP encoding process is illustrated in Figure.4.3.

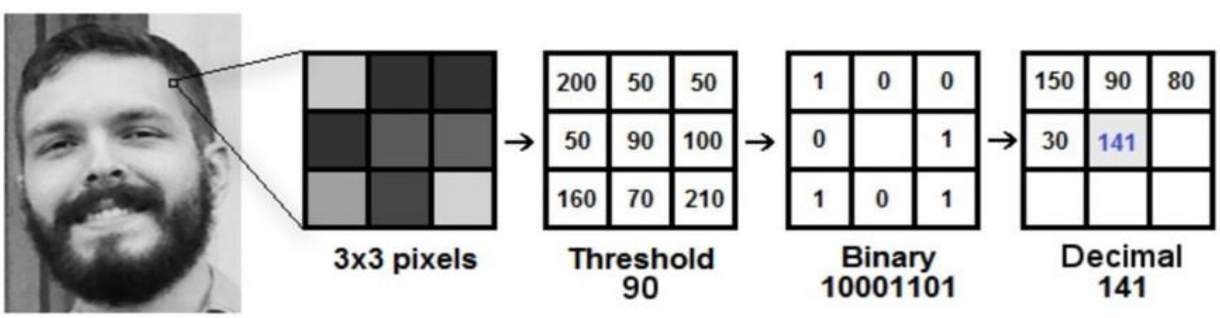


FIGURE 4.3 LBP OPERATION

The first defined LBPs for neighborhoods of different sizes, thus making it feasible to deal with textures at different scales. The second defined the so-called uniform patterns: an LBP is “uniform” if it contains at most one 0-1 and one

1-0 transition when viewed as a circular bit string. Uniformity is important because it characterizes the patches that contain primitive structural information such as edges and corners. Ojala *et al.* observed that although only 58 of the 256 8-bit patterns are uniform, nearly 90% of all observed image neighborhoods are uniform and many of the remaining ones contain essentially noise. Thus, when histogramming LBPs the number of bins can be reduced significantly by assigning all non uniform patterns to a single bin, typically without losing too much information. Distance measure is used for matching. After gathering the face samples train the recognizer for these samples so that it can predict the faces accurately. Recognize face from the live video. Once the RASPBERRY PI recognizes any saved face, it will make the relay module high to open the solenoid lock. If the sample doesn’t match it will capture the image and save it to a folder.

CHAPTER 5
SOFTWARE REQUIREMENTS

Software requirements deal with defining software resourcerequirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

The Software Requirements for this project include:

1. Virtual Network Computing (VNC) Viewer
2. Thonny

5.1 VIRTUAL NETWORK COMPUTING (VNC) VIEWER

The RFB protocol is used to convey control events and screen pixel data from one computer to another over a network. Real VNC created a simple yet effective protocol. VNC Viewer displays the desktop of the computer that VNC Server captures in real time. VNC Viewer collects your input (mouse, keyboard, or touch) and delivers it to VNC Server, which injects it and allows you to control your computer remotely. For the remote computer you want to control, you'll need a VNC Server, and for the computer or mobile device you want to control, you'll need a VNC Viewer. You have the option of pre-installing and licensing this software or downloading and running them as needed. A viewer, on the other hand, is a software application that displays the contents of a digital file. VNC Viewer is shown in Figure 5.1 is used to operate local PCs and mobile devices from a remote location.

VNC Viewer software installed on a device such as a computer, tablet, or smart phone can access and control a machine at another location. It's a graphical desktop sharing system that lets you control the desktop of a distant computer (running VNC Server) from your device, and it sends keyboard, mouse, and touch events to VNC Server so you can control the machine after you're connected.

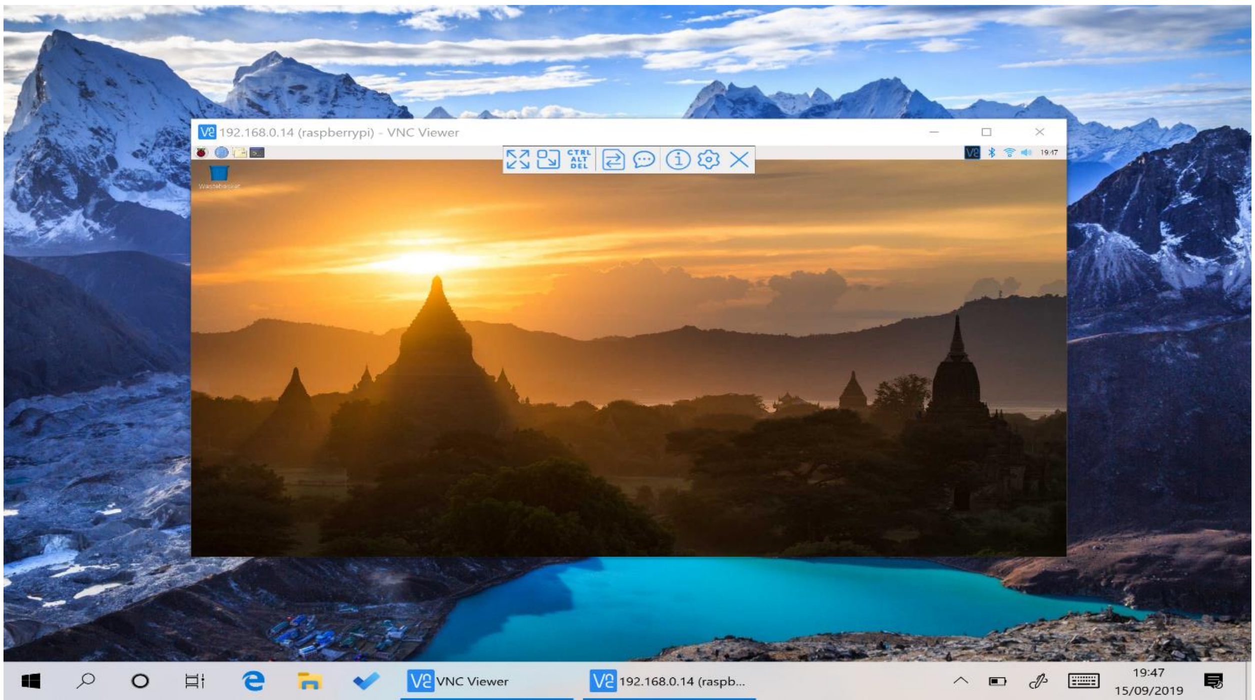


FIGURE 5.1 VNC VIEWER REMOTE ACCESS A RASPBERRY PI

5.2 THONNY

Thonny is a new integrated development environment (IDE) that comes with the Raspbian with PIXEL operating system's newest edition. It's now much easier to learn to code thanks to Thonny. You don't need to install anything because Thonny comes with Python 3.6 built in. Simply launch the programme from the Menu > Programming menu. It has a number of advanced capabilities that aren't available in the Python 3 (IDLE) application that comes with Raspbian. You'll see a new script editor and a shell when you first launch Thonny. You enter a programme in the script editor and run it in the shell, same like in Python 2/3 IDLE. The shell can then be used to interact directly with the programme, allowing you to access variables, objects, and other features. Thonny comes with a number of extra features that make it ideal for learning programming which is shown in Figure 5.2. One of the nicest features is the debug mode, which is both powerful and simple to use. Instead of running your programme, it goes line by line through the code. You can see how variables and objects are generated, as well as how values are provided into functions and compared.

Debuggers are frequently found in sophisticated IDEs, but they usually require you to manually create breakpoints (places where the programme freezes so you can examine the code). Thonny takes a much more direct approach. It also contains a number of panels that allow you to look at things like variables, objects, and the heap (the memory space where items are stored).

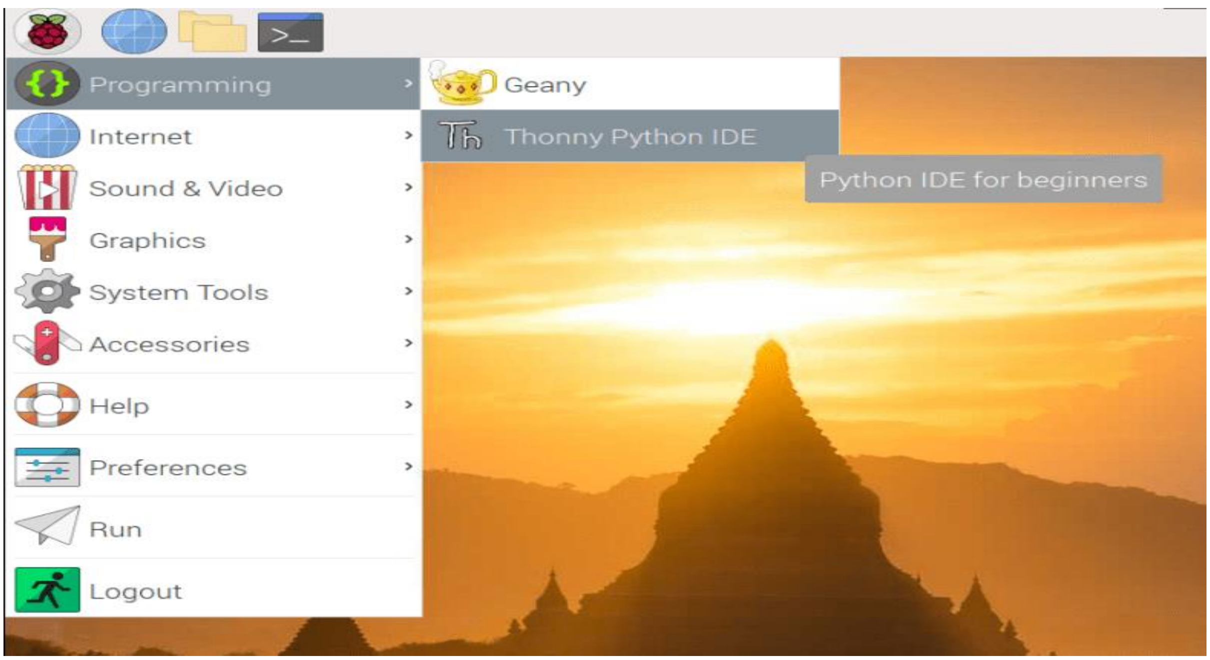


FIGURE 5.2 THONNY HOME PAGE IN RASPBERRYPI

CHAPTER 6

ALGORITHMS

6.1 CODE FOR LOGIN INTERFACE

```
<html>
<head>
<meta name = "viewport" content = "width=device-width">
<title>Home Automation</title>
<style>
	body {
		background: url('bg.jpg');
		background-repeat: no-repeat;
		background-size:  cover;
		background-position: 35%;
	}
	.login-div {
		display:flex;
		flex-direction: column;
		align-items: center;
		border: 2px solid;
		max-width: 400px;
		border-radius: 10px;
		background: lightblue;
		position: absolute;
		top: 50%;
		left: 50%;
		transform: translate (-50%, -50%);
		padding: 10px;
	}
	.login-div * {

		margin: 10px;
		font-size: 20px;
	}
	.fail {
```

```
                color: red;
                font-face: 'Monospace';
                display: none;
        }
</style>
</head>
<body>
<div class="login-div">
        <input type = "text" id="user_txt" placeholder = "Enter Username">
        <input type = "password" id="pass_txt" placeholder = "Enter Password">
<button onclick="login_clk()">LOGIN</button>
<div id="user_fail" class="fail">User Authentication Failed</div>
</div>
<script>
var users = {
                "Velmurugan": "velraj",
                "Baranidharan": "barani"
                "Prakash": "prakash"
                "Rohith": "rohith"
        }
```

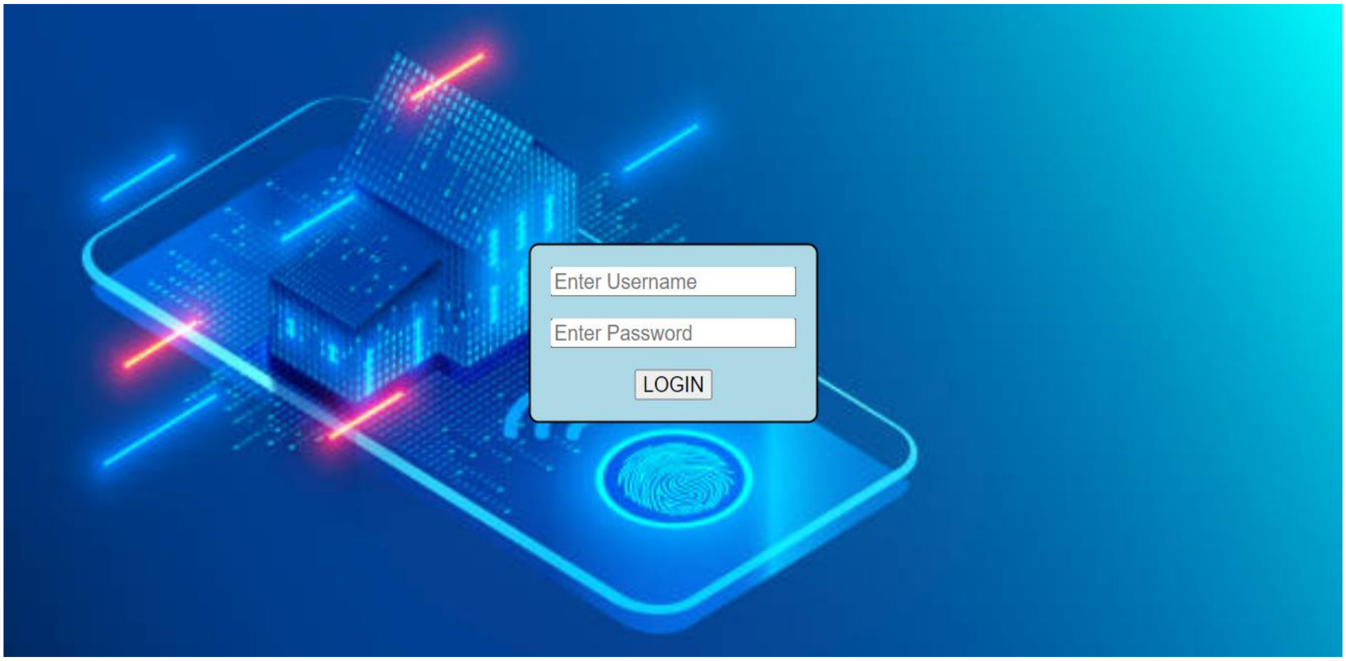


FIGURE 6.1 IMAGE USED IN LOGIN PAGE

6.2 STYLE.CSS CODE FOR WEBPAGE

Code for adding styles to Home automation webpage

```
{
   box-sizing: border-box;
}
html {
   /* background-color:red; */
   /* background-image: linear-gradient (hsl(35, 94%, 45%),
      hsl(40,74%, 52%), hsl(51, 68%, 63%)); */
   background-repeat: no-repeat;
   /* background-image: url(home.jpg);
   width: 100%;
   height: 100%;
   background-size: 500px 300px; */
}
body {
   font-weight: bold;
   font-family: -apple-system,
      BlinkMacSystemFont,
      "Segoe UI",
      Roboto,
      Helvetica,
      Arial,
      sans-serif,
      "Apple Color Emoji",
      "Segoe UI Emoji",
      "Segoe UI Symbol";
   height: 100%;
   line-height: 1.5;
   -webkit-touch-callout: none;
   -webkit-user-select: none;
   -khtml-user-select: none;
   -moz-user-select: none;
   -ms-user-select: none;
   user-select: none;
   background-image: url(h.jpg);
   /* width: 100%;
```

```
    height: 100%; */
    /* background-size: auto; */
    background-repeat: no-repeat;
    background-size: cover;
}
h5 {
    color: #fff;
    margin: 1%;
}
img {
    filter: grayscale(100%);
}
p {
    margin: 0;
}
ul {
    list-style: none;
    padding: 0;
}
.content {
    margin: 50px;
}
.status {
    color: #fff;
    margin: 0 0 5% 1%;
}
.button {
    background-color: hsl(35, 55%, 80%);
    border-radius: 10px;
    color:red;
    display: flex;
    margin: 1%;
}
.button-wrapper {
    display: flex;
    flex-wrap: wrap;
}
.scene {
    align-items: center;
    font-size: 10pt;
```

```
  height: 50px;
  padding-left: 1rem;
  width: 240px;
}
.accessory {
  flex-direction: column;
  font-size: 9pt;
  height: 115px;
  padding: 1rem .75rem;
  width: 110px;
}
.on,
.thermostat {
  background-color: hsl(104, 100%, 100%);
  color: black;
}
.on .fa-home {
  color: hsl(33, 93%, 54%);
}
.on .fa-lightbulb {
  color: hsl(48, 100%, 50%);
}
.fa-thermometer-three-quarters {
  color: hsl(199, 94%, 67%);
}
.thermostat:active {
  background-color: hsl(35, 55%, 80%);
  color: gray;
  filter: grayscale(100%);
}
.button-state {
  color: gray;
}

.fas {
  font-size: 2em;
  margin-bottom: 5px;
}
.fa-home {
  font-size: 1.5em;
```

```
   margin: .25em;
}
.bulb {
   margin-bottom: 5px;
   width: 14px;
}
.temperature {
   margin-bottom: 5px;
   width: 28px;
}
```

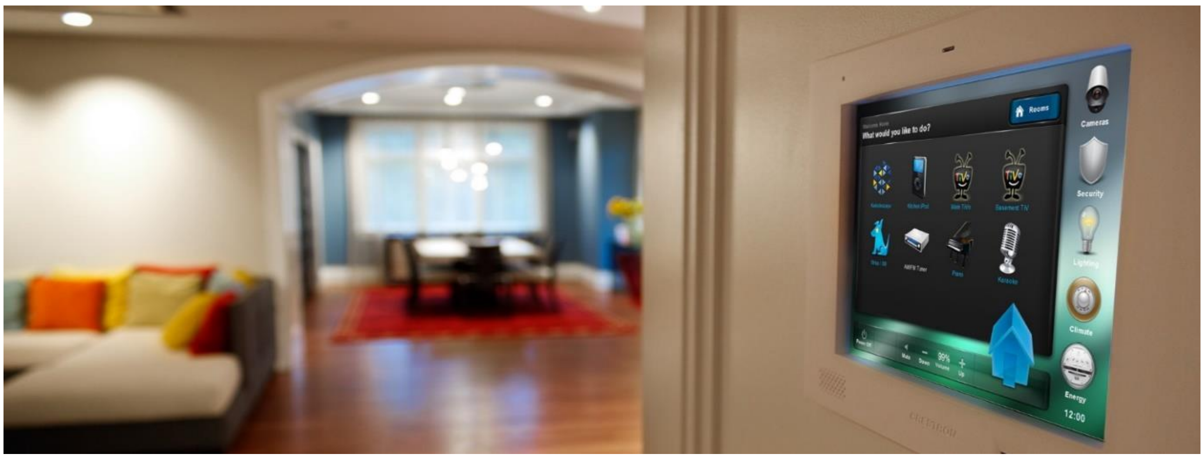


FIGURE 6.2 IMAGES USED IN CREATION OF WEB PAGE

6.3 WEBPAGE HTML CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <script src="https://kit.fontawesome.com/07ce60a6d3.js"></script>
   <link rel="stylesheet" href="styles.css">
   <title>Home</title>
</head>
<body onclick="body_clk();" >
   <div class="content">
      <!-- <div class="status">
         <h1>The Hideaway</h1>
```

```
        <ul>
          <li>2 accessories not responding.</li>
          <li>3 lights on.</li>
          <li>Motion detected.</li>
          <li>and 3 More ></li>
        </ul></div> -->

<marquee style="width:100%"> WELCOME TO THE SMART HOME WITH
SECURITY HERE WE CAN CONTROL HOME APPLIANCES FROM
ANYWHERE ALSO WE CAN MONITOR THE HOME SECURITY FROM
ANYWHERE </marquee>
      <h5>Favorite Scenes</h5>
        <ul class="button-wrapper">
          <li class="button scene" id="good-morning"><i class="fas fa-
home"></i>Good Morning</li>
          <li class="button scene" id="good-evening"><i class="fas fa-
home"></i>Good Evening</li>
          <li class="button scene" id="leave-home"><i class="fas fa-
home"></i>Leave Home</li>
          <li class="button scene" id="tv-time"><i class="fas fa-home"></i>TV
Time</li>
          <li class="button scene" id="good-afternoon"><i class="fas fa-
home"></i>Good Afternoon</li>
          <li class="button scene" id="good-night"><i class="fas fa-
home"></i>Good Night</li>
          <li class="button scene" id="arrive-home"><i class="fas fa-
home"></i>Arrive Home</li>
      </ul>
        <h5>Favorite Accessories</h5>
         <ul class="button-wrapper">
          <li class="button accessory" id="fan_switch" onclick="fan_click(event)">
            <i class="fas fa-lightbulb"></i>
            <p>Fan</p>
            <p class="button-state">Off</p>
          </li>
          <li class="button accessory" id="light_switch"
onclick="light_click(event)">
            <i class="fas fa-lightbulb"></i>
            <p>Light</p>
            <p class="button-state">Off</p>
```

```
          </li>
          <!-- <li class="button accessory thermostat" id="kitchen-my-ecobee"> <i
class="fas fa-thermometer-three-quarters"></i>
             <p>Kitchen My ecobee</p>
             <p class="button-state">Cool to 70&deg;</p>
          </li> -->
          <li class="button accessory" id="temp_switch"
onclick="temp_click(event)">
             <i class="fas fa-lightbulb"></i>
             <p>Room temperature</p>
             <p class="button-state" id="temp_show"></p>
          </li>
          <li class="button accessory" id="pir_switch" onclick="pir_clk(event)">
             <i class="fas fa-lightbulb"></i>
             <p>Check Intruder</p>
             <p class="button-state">Off</p>
          </li>
          <li class="button accessory" id="tv-room-tv-corner">
             <i class="fas fa-lightbulb"></i>
             <p>Sensor 3</p>
             <p class="button-state">Off</p>
          </li>
          <li class="button accessory" id="office-floor-lamp">
             <i class="fas fa-lightbulb"></i>
             <p>Sensor 4</p>
             <p class="button-state">Off</p>
          </li>
          <li class="button accessory" id="lock_switch" onclick="lock_click()">
             <i class="fas fa-lightbulb"></i>
             <p>Camera Module</p>
             <p class="button-state">Off</p>
            </li>

       </ul>
 <h4 >THIS WEBSITE DESIGNED BY
<li>
BARANIDHARAN
</li>
<li>
VELMURUGAN
```

```
</li>
<li>
PRAKASH
</li>
<li>
ROHITHVARON
</li>
</h4>
<style>
h4{color:orange;
  font-size:20px;}
</style>
   </div>
</body>
   <script src="index.js"></script>
</html>
```

6.4 JAVA SCRIPT CODE

```
var light_on = false, fan_on = false;
var rasp_ip = location.protocol + "//" + location.host;
var prev_timeout;

var prev_time = parseInt(localStorage.getItem("prev_time"));
if (prev_time && (Date.now()-prev_time)> 30*1000){
        location.href = "login.html";
}

function body_clk(){
        console.log('reset_timer');
        if(prev_timeout) clearTimeout(prev_timeout);
//      prev_timeout = setTimeout(()=>{
//              alert("Timeout! Please login again");
//              location.href = "login.html";
//      }, 60*1000);
}
function light_click(event){
        if(light_on){
                fetch(`${rasp_ip}/light_click/?x=0`);
```

```
		light_switch.getElementsByClassName('button-state')[0].innerHTML
= "Off";
	} else {
		fetch(`${rasp_ip}/light_click/?x=1`);
		light_switch.getElementsByClassName('button-state')[0].innerHTML
= "On";
	}
	light_switch.classList.toggle("on");
	light_on = !light_on;
		}
function fan_click(event){
	if(fan_on){
		fetch(`${rasp_ip}/fan_click/?y=0`);
		fan_switch.getElementsByClassName('button-state')[0].innerHTML =
"Off";
	} else {
		fetch(`${rasp_ip}/fan_click/?y=1`);
		fan_switch.getElementsByClassName('button-state')[0].innerHTML =
"On";
		}
	fan_switch.classList.toggle("on");
	fan_on = !fan_on;
}
function temp_click(event){
	fetch(`${rasp_ip}/temp_click/?y=1`).then(res=>res.text()).then(data=> {
		temp_show.innerHTML = data;
		var hum = parseFloat(data.slice(data.search("Humidity=")+9));
		if(hum>55)
			alert("High Temp and Humidity");

		data.search("Humidity=")
	});
}
function lock_click(event){
	fetch(`${rasp_ip}/lock_click/?y=1`).then(res=>res.text()).then(data=> {
		console.log (data);
	});
}
var pir_inter = null;
function pir_clk(){
```

```
        if(!pir_inter){
                pir_inter = setInterval(()=>{
        fetch(`${rasp_ip}/pir_click`).then(res=>res.text()).then(data=> {
                        console.log (data==`"detected"`);
                        if(data==`"detected"`){
                                pir_switch.style.background = "coral";
                        }else{
                            pir_switch.style.background = "lightgreen";
                        }
                        });
            }, 1000)
        } else {

            pir_switch.style.background = "";
            clearInterval(pir_inter);
            pir_inter = null;
        }
}
//body_clk();
```

6.5 SERVER PYTHON CODE

```
#%%
from fastapi import FastAPI
from fastapi.staticfiles import StaticFiles
import RPi.GPIO as GPIO
import time
import cv2
import numpy as np
import os
import Adafruit_DHT
 DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 4
PIR_input = 29                     #read PIR Output
LED = 32                           #LED for signalling motion detected
def set_board():
   GPIO.setwarnings(False)
   GPIO.setmode(GPIO.BOARD)
   GPIO.setup(3, GPIO.OUT)
   GPIO.setup(16, GPIO.OUT)
```

```
    GPIO.setup(PIR_input, GPIO.IN)
    GPIO.setup(LED, GPIO.OUT)
    GPIO.output(LED, GPIO.LOW)
set_board()
app = FastAPI()
print("server_start")
app.mount("/public", StaticFiles(directory="public"), name="static")
@app.get("/light_click/")
async def light_click(x):
    if(x=="1"):
        GPIO.output(3, GPIO.HIGH)
        print('on')
    else:
        GPIO.output(3, GPIO.LOW)
        print('off')
@app.get("/fan_click/")
async def fan_click(y):
    if(y=="1"):
        GPIO.output(16, GPIO.HIGH)
        print('on')
    else:
        GPIO.output(16, GPIO.LOW)
        print('off')
@app.get("/temp_click/")
async def temp_click():
    set_board()
    humidity, temperature = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)
    if humidity is not None and temperature is not None:
        return ("Temp={0:0.1f}C Humidity={1:0.1f}%".format(temperature,
humidity))
    else:
        return ("Sensor failure. Check wiring.");
@app.get("/pir_click/")
async def pir_click():
    if(GPIO.input(PIR_input)):
        GPIO.output(LED, GPIO.HIGH)
        return("detected")
    else:
        GPIO.output(LED, GPIO.LOW)
        return("not detected")
```

```
@app.get("/lock_click/")
async def lock_click():
   os.environ['DISPLAY'] = ':0'
    relay = 24
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(relay, GPIO.OUT)
    GPIO.output(relay ,0)
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('newtrainer.yml')
    cascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath);
    font = cv2.FONT_HERSHEY_SIMPLEX
    # iniciate id counter
    id = 0
    # names related to ids: example ==> Marcelo: id=1,  etc
    names = ['barani', 'Prakash', 'Ilza', 'Z', 'W']
    # Initialize and start realtime video capture
    cam = cv2.VideoCapture(0)
    cam.set(3, 640)  # set video widht
    cam.set(4, 480)  # set video height
    # Define min window size to be recognized as a face
    minW = 0.1 * cam.get(3)
    minH = 0.1 * cam.get(4)
    while True:
        ret, img = cam.read()
        #img = cv2.flip(img, -1)  # Flip vertically
        cv2.imwrite("frame1.jpg",img);
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(
            gray,
            scaleFactor=1.2,
            minNeighbors=5,
            minSize=(int(minW), int(minH)),
        )
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
            id, confidence = recognizer.predict(gray[y:y + h, x:x + w])
            print(confidence)
            # Check if confidence is less them 100 ==> "0" is perfect match
if (confidence < 40):
```

```
            id = names[id]
 confidence = "  {0}%".format(round(100 - confidence))
            GPIO.output(relay,1)
            print("Openning lock")
            cam.release()
            cv2.destroyAllWindows()
            return ("OPEN")
         else:
            id = "unknown"
 confidence = "  {0}%".format(round(100 - confidence))
            GPIO.output(relay,0)
       cv2.putText(img,str(id), (x + 5, y - 5), font,1, (255, 255, 255), 2)
       cv2.putText(img,str(confidence),(x+5,y+h-5),font,1,(255,255,0),1)
      cv2.imshow('camera', img)
      k = cv2.waitKey(10) & 0xff  # Press 'ESC' for exiting video
      if k == 27:
         break
   # Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
   cam.release()
   cv2.destroyAllWindows()
```

6.6 DATASET CREATION CODE

```
import cv2
import os
# Call the built-in camera of the notebook, so the parameter is 0. If there are other
cameras, you can adjust the parameter to 1, 2
os.environ['DISPLAY'] = ':0'
cap = cv2.VideoCapture(0)
# Cascade classifier is a cascaded classifier for face detection in Opencv
face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
face_id = input('\n enter user id:')

print('\n Initializing face capture. Look at the camera and wait ...')
count = 0
while True:
   # Read pictures from camera
   sucess, img = cap.read()
   # Turn to grayscale picture
```

```
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # Face detection
    faces = face_detector.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+w), (255, 0, 0))
        count += 1
        # Save image
        cv2.imwrite("newDataset/facerecog." + str(face_id) + '.' + str(count) + '.jpg',
gray[y: y + h, x: x + w])
        cv2.imshow('image', img)
    # Keep the picture going.
k = cv2.waitKey(1)
    if k == 27:   # Exit camera by esc key
        break
    elif count >= 100:  # Get 30 samples and exit the camera
        break
# Turn off camera
cap.release()
cv2.destroyAllWindows ()
```

6.7 PYTHON CODE USED IN TRAINER BLOCK

```
import cv2
import numpy as np
from PIL import Image
import os
# Path for face image database
path = 'newDataset'

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
# function to get the images and label data
def getImagesAndLabels(path):
    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    ids = []
    for imagePath in imagePaths:
        PIL_img = Image.open(imagePath).convert('L') # grayscale
        img_numpy = np.array(PIL_img,'uint8')
        id = int(os.path.split(imagePath)[-1].split(".")[1])
```

```
        faces = detector.detectMultiScale(img_numpy)
        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)
    return faceSamples,ids
print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))
# Save the model into trainer/trainer.yml
recognizer.write('newtrainer.yml')
# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))
```

6.8 PYTHON CODE USED IN RECOGNIZER BLOCK

```
import cv2
import numpy as np
import os
import RPi.GPIO as GPIO
import time
os.environ['DISPLAY'] = ':0'
relay = 24
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(relay, GPIO.OUT)
GPIO.output(relay ,0)
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('newtrainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);
font = cv2.FONT_HERSHEY_SIMPLEX
# iniciate id counter
id = 0
# names related to ids: example ==> Marcelo: id=1,  etc
names = ['barani', 'Prakash', 'Ilza', 'Z', 'W']
# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
cam.set(3, 640)  # set video widht
cam.set(4, 480)  # set video height
# Define min window size to be recognized as a face
minW = 0.1 * cam.get(3)
```

```
minH = 0.1 * cam.get(4)
while True:
    ret, img = cam.read()
    #img = cv2.flip(img, -1)  # Flip vertically
    cv2.imwrite("frame1.jpg",img);
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.2,
        minNeighbors=5,
        minSize=(int(minW), int(minH)),
    )

    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
        id, confidence = recognizer.predict(gray[y:y + h, x:x + w])
        print(confidence)
        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 40):
            id = names[id]
            confidence = "  {0}%".format(round(100 - confidence))
            GPIO.output(relay,1)
            print("Openning lock")
        else:
            id = "unknown"
            confidence = "  {0}%".format(round(100 - confidence))
            GPIO.output(relay,0)
        cv2.putText(img, str(id), (x + 5, y - 5), font, 1, (255, 255, 255), 2)
        cv2.putText(img, str(confidence), (x + 5, y + h - 5), font, 1, (255, 255, 0), 1)
    cv2.imshow ('camera', img)
    k = cv2.waitKey (10) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
# Do a bit of cleanup
print ("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```

CHAPTER 7
RESULTS AND DISCUSSIONS

7.1 LOGIN PAGE

FIGURE 7.1 LOGIN PAGE

The login page dashboard is shown in figure 7.1. The user can login into the portal using the appropriate login credentials.

7.2 WEBPAGE

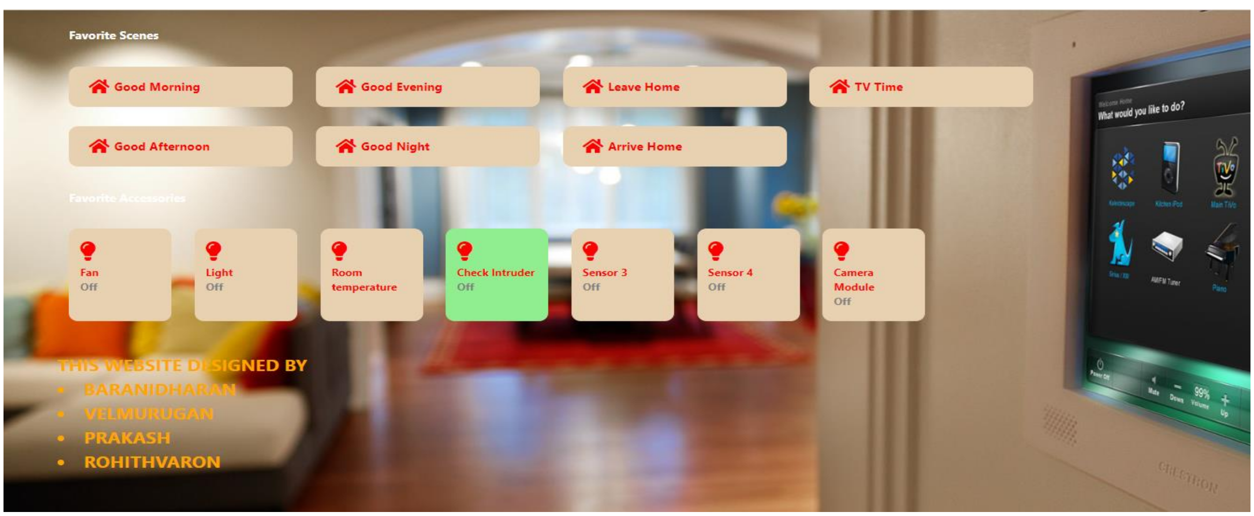


FIGURE 7.2 USER INTERFACE

Figure 7.2 depicts the user interface created specifically for the smart home control system. Using the user interface, the user is able to remotely control the operation of the fan, light, camera module, and intruder detector module. The interface may also be used to check the temperature of the room.

7.3 GLOWING BULB

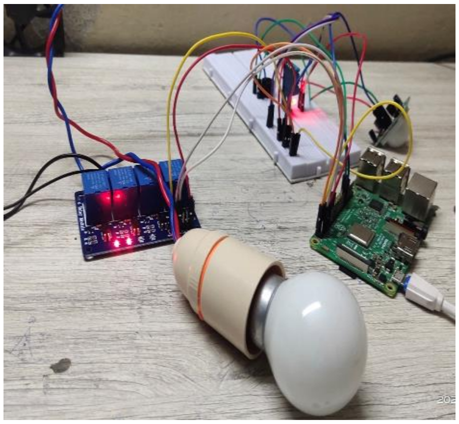


FIGURE 7.3 BULB IN OFF STATE

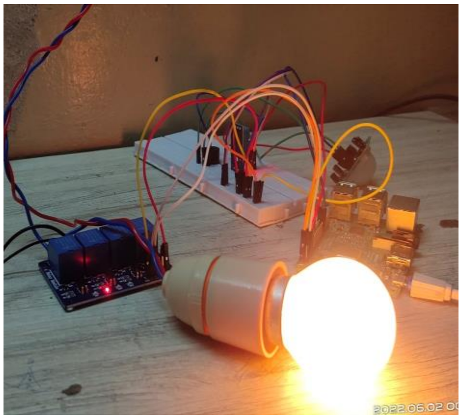
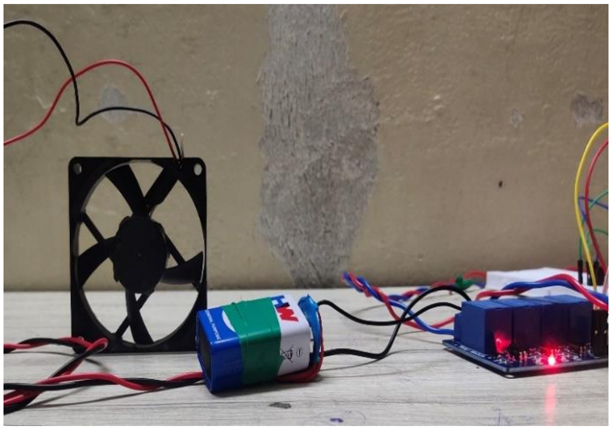


FIGURE 7.4 BULB IN ON STATE

The Raspberry Pi sends the 3.3V signal to the relay module, turning on the light module, when the user turns on the light module via the user interface. Using the same button on the user interface, the light can be turned OFF. The light bulb is shown in ON and OFF states in Figures 7.3 and 7.4 before and after the light module is turned on using UI.

7.4 FAN CONTROL



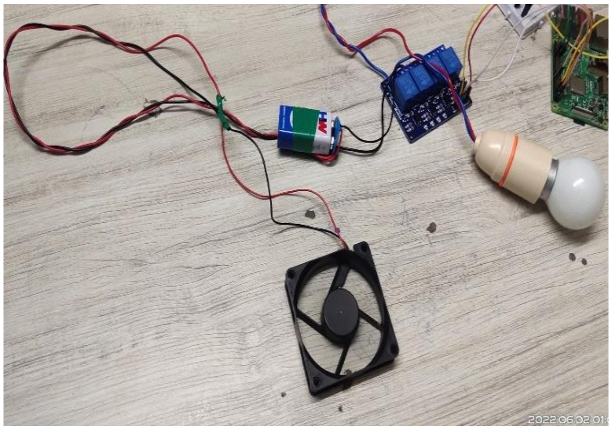


FIGURE 7.5
FAN IN OFF STATE

FIGURE 7.6
FAN IN ON STATE

The fan is shown as being off in Figure 7.5. The Raspberry Pi server receives the command from the user when they click the ON button on the website, and the matching GPIO pin is triggered with a 3.3V output and connected to the power supply relay. The fan, which is connected as illustrated in figure 7.6, is then powered by the power supply relay.

7.5 DHT11 SENSOR

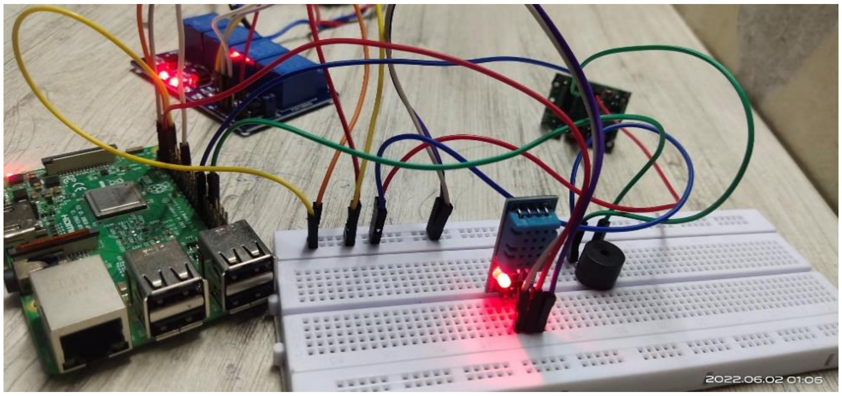


FIGURE 7.7
DHT11 SENSOR

The DHT11 sensor is attached to the Raspberry Pi in Figure 7.7. When a user clicks the website's ON button, the sensor replies by sending the temperature and humidity readings, which are then sent to the website. The user is presented with an alert message if the temperature exceeds the threshold value that has been established.

7.6 PIR SENSOR

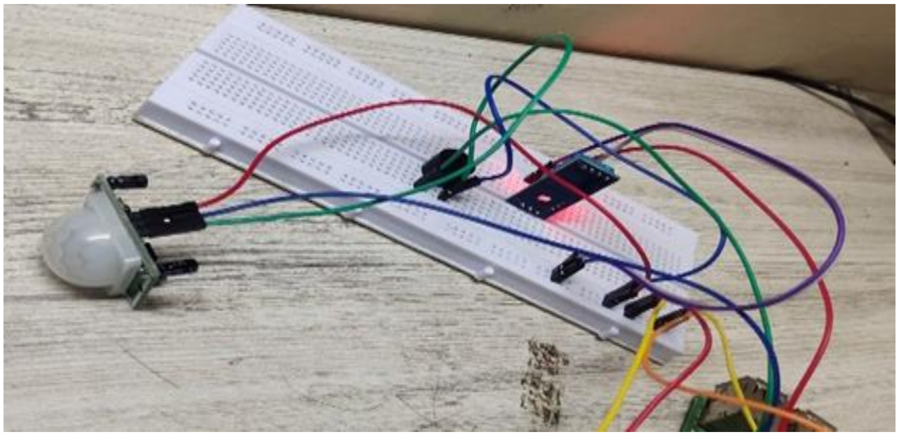


FIGURE 7.8
PIR SENSOR

The Raspberry Pi and PIR sensor are shown coupled in Figure 7.8. Information about an intruder is relayed from the PIR sensor via the Raspberry Pi server to the website. When an intruder is detected, the website displays a red alert indicator, which becomes green when no intruder is present.

7.7 DOORLOCK

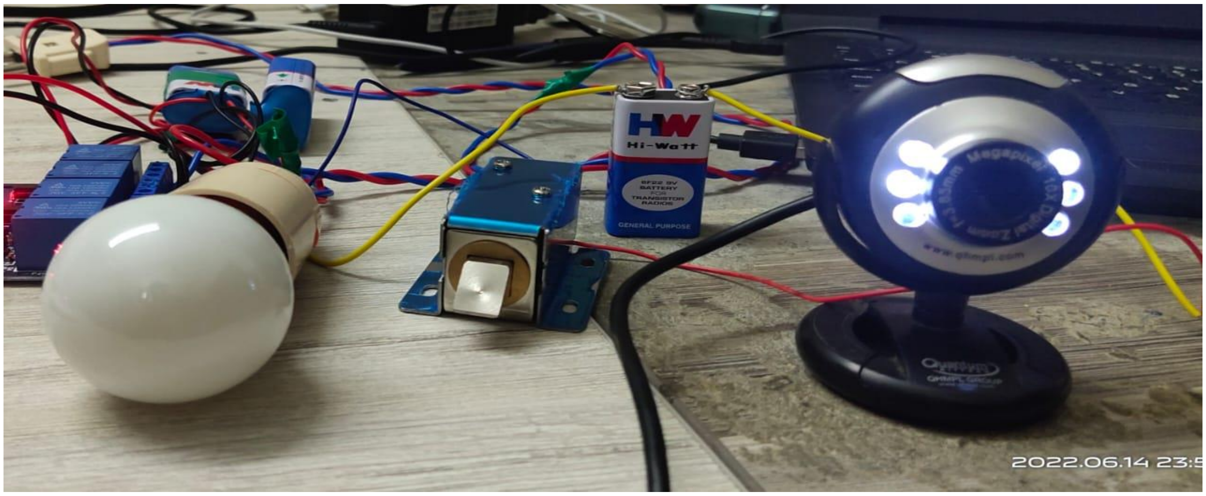


FIGURE 7.9 DOOR IN LOCK STATUS

Figure 7.9 demonstrates how the door lock and camera module work together to detect an intruder while simultaneously unlocking the door when a recognised face is spotted.

CONCLUSIONS AND FUTURE SCOPE

Our project's main goal is to use an Android smart phone to conveniently operate home appliances while also providing comprehensive home security and safety features. Different emerging technologies, such as retina/fingerprint scanning, can also be used to log in. To improve the security system's accuracy, we can integrate image processing. This system would employ a trusted face database to assess if an intruder had been detected or if the alarm was false.

Home automation technologies have the potential to make homes even smarter in the future. Sensors such as motion sensors, light sensors, and temperature sensors can be integrated into homes to allow automated device switching based on conditions. More energy can be saved by ensuring that the house is occupied before turning on devices, checking brightness, and turning off lights that aren't needed. The technology can be tightly connected with home security solutions to give homeowners more control and protection. The next stage would be to expand this technology to automate a large-scale environment like offices or factories.

The smart home appliances use the least amount of energy possible. This is due to the fact that they adjust the result using stored user data. As a result, a significant amount of energy is saved. Home automation is only going to get bigger and better in the future. The Internet of Things is already bringing a slew of creative and intriguing solutions to the table. Sensors in parking lots will tell whether or not a space is available. Voice commands may be used to manufacture a favorite beverage, or Smartphone commands could be used to close the blinds in the living room while people are away on vacation. The possibilities of home automation are endless. However, the issue of sustainability is the most useful benefit. Home automation is based on the notion of sustainability, which promotes resource efficiency.

REFERENCES

1. M. R. Alam, M. B. I. Reaz and M. A. M. Ali, "A Alam, Muhammad Raisul, Mamun Bin Ibne Reaz, and Mohd Alauddin Mohd Ali. “A review of smart homes: Past, present, and future.” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42, no. 6 (2012): 1190-1203.
2. Enrique Correa, Correa, E., A. Jonker, M. Ozo, and R. Stolk. “Emotion Recognition using deep convolutional neural networks.” Tech. Report IN4015 (2016).
3. Minaee S, Abdolrashidi Minaee, Shervin, and Amirali Abdolrashidi. ”Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network.” arXiv preprint arXiv:1902.01019 (2019).
4. Amarappa, S., and S. V. Sathyanarayana. “Data classification using Support vector Machine(SVM), a simplified approach.” Int. J. Electron. Comput. Sci. Eng 3 (2014): 435-445
5. Al-Ali, A.R. ; Dept. of Comput. Eng., American Univ., United Arab Emirates ; AL-Rousan, M., "Java-based home automation system"2004.
6. Stefan Marksteiner, Víctor Juan Exposito Jimenez, Heribert Valiant, Herwig Zeiner, " Internet of Things Business Models, Users, and Networks", 2017