

*A project report  
on*

**“Home Automation With AI”**

*A project report submitted in partial fulfillment of the requirement for the Degree  
of B. Tech, in Electrical Engineering*

Bachelor of Technology  
In  
Electrical Engineering Department

Submitted by

Gautam Kumar	(20103141015)
Md. Arman	(20103141003)
Ritesh Kumar	(20103141013)

Under the Supervisor of

**Mr. LUCKY KUMAR**

**Assistant Professor, Electronics & Communication Engineering Department  
Government Engineering College Nawada.**



**GOVERNMENT ENGINEERING COLLEGE NAWADA**

**Bihar Engineering University Patna**

**March 2024**

# GOVERNMENT ENGINEERING COLLEGE NAWADA

Bihar Engineering University Patna

## CERTIFICATE

This is to certify that the project entitled "**HOME AUTOMATION WITH AI**" has been successfully completed. **Md. Arman (20103141003)**, **Gautam Kumar (20103141015)**, **Ritesh Kumar (20103141013)** bonafide Students of **Government Engineering College Nawada** in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in **Electrical Engineering** of the **Bihar Engineering University, Patna** during the academic Session 2020-2024. It is certified that all the corrections indicated for internal Assessment have been incorporated in the project report. The project report has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

**Signature of the Supervisor**

**Signature of the HOD**

**Signature of the Principa**

.....  
Mr. Lucky Kumar

.....  
Rajesh Baitha

.....  
Dr. Vinay Kumar Chaudhary

Assistant professor

Assistant Prof. & HOD

Principal

ECE Department

EE Department

GEC NAWADA

GEC NAWADA

GEC NAWADA

### External Viva

Name of Examiners

Signature & Date

1.

2.

## DECLARATION

**We Declare that -**

- a) The work contained in this project is original and has been done by me under the guidance of my supervisor.
- b) The work has not been submitted to any other Institute For any degree or diploma.
- c) We have followed the guidelines provided by the Institute in preparing the report.
- d) We have confirmed the norms and guidelines given in the ethical code of Conduct of the Institute.
- e) Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and given their details in the references.

---

Signature of the Students

**Md. Arman**

**Reg. No- 20103141003**

---

Signature of the Students

**Gautam Kumar**

**Reg. No- 20103141015**

---

Signature of the Students

**Ritesh Kumar**

**Reg. No- 20103141013**

## **ACKNOWLEDGEMENT**

First of all, We are indebted to the GOD ALMIGHTY for giving us an opportunity to excel in our efforts to complete this project on time. We are extremely grateful to **Dr. Vinay Kumar Chaudhary**, Principal, Government Engineering College, Nawada and **Dr. Rajesh Baitha**, Head of Department of Electrical Engineering, for providing all the required resources for the successful completion of our project. Our heartfelt gratitude to our project supervisor Mr. Lucky Kumar, Assistant Professor, Electronic & Communication Engineering Department, for his valuable suggestions and guidance in the preparation of the project report. We express our thanks to all cultures Mr. Shivam Kumar, Mr. Shikha Sharma, Dr. Avinash Kumar, Mr. Shashi Kumar, Mrs, Jyoti Kumari, Mr. Nikhil Kumar and all staff members and friends for all the help and co ordination extended in bringing out this project successfully in time. We will be failing in duty if we do not acknowledge with gratitude thanks to the authors of the references and other literature referred to in this project. Last but not the least; We are very much thankful to our parents who guided us in every step which we took.

Thanking you

**Md. Arman (20103141003)**

**Gautam Kumar (20103141015)**

**Ritesh Kumar (20103141013)**

## **ABSTRACT**

This project brings the convenience of home automation into your hands, utilizing the versatile Raspberry Pi 3 Model B. Imagine controlling lights, fans, and ACs across different rooms, from your smartphone, anywhere in the world. That's what this project delivers, using a user-friendly web interface. But it doesn't stop there! Talk to your appliances with voice commands for hands-free control, or interact with a Telegram chatbot for a natural language experience.

The project goes beyond basic control by integrating AI through the Gemini NLM(Natural Language Model). This opens doors for exciting future features like personalized schedules that adjust based on your preferences or even intelligent automation that considers environmental factors.

## TABLE OF CONTENTS

FRONT PAGE	i
CERTIFICATE	ii
DECLARATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
CHAPTER - 01	1-2
CHAPTER - 02	3-4
CHAPTER - 03	5-8
CHAPTER - 04	9
CHAPTER - 05	10
CHAPTER - 06	11
CHAPTER - 07	14

## INTRODUCTION

Welcome to the future of home automation! This project, "Home Automation with AI," leverages the power of Raspberry Pi 3 Model B to create a smart and intuitive system that allows you to control your home appliances effortlessly. With a combination of web-based UI, speech-based control, and integration with Telegram chatbot powered by Gemini AI, this project revolutionizes the way you interact with your home environment.

### Project Overview:

#### 1. Intuitive Web UI with Flask:

- **Modern Interface:** Control your home using a visually appealing and user-friendly web interface accessible from any device, anywhere in the world.
- **Precise Control:** Choose from dedicated buttons for each appliance (bedroom lights, fans, ACs, kitchen bulbs, washroom lights, geysers, living room lights, fans, ACs, TV, motor pump) to toggle their states with a single click.
- **Security:** The web interface is secured with `flask-BasicAuth`, a Python library that requires users to enter a username and password before accessing any controls. This prevents unauthorized access to your home automation system.
- **HTTPS Encryption:** Implement HTTPS to encrypt all communication between your device and the web interface, ensuring confidentiality and integrity of data.

#### 2. Voice Control:

- **Conversational Comfort:** Converse naturally with your home using voice commands to manage appliances without lifting a finger.
- **Contextual Understanding:** Gemini, the advanced language model, deciphers the intent behind your commands, even if phrased differently (e.g., "I'm going to take a bath" translates to "Turn on the geyser").
- **Personalized Experience:** Tailor the voice commands to your preferences and routines, creating a truly intuitive interaction.

### 3. Intelligent Telegram Chatbot with Gemini:

- **Enhanced Control:** Manage your home through a familiar Telegram interface, sending direct commands (e.g., "Turn on child bedroom bulb") or leveraging Gemini's understanding of your intent (e.g., "It's getting hot, turn on the AC").
- **Informative Assistant:** Gemini not only executes your commands but also engages in conversation, answering your questions and providing insights.
- **Secure Messaging:** Telegram's encryption ensures secure communication for your home automation commands.
- **Community Potential:** Explore the possibilities of integrating your chatbot with community-driven AI models, further expanding its capabilities.



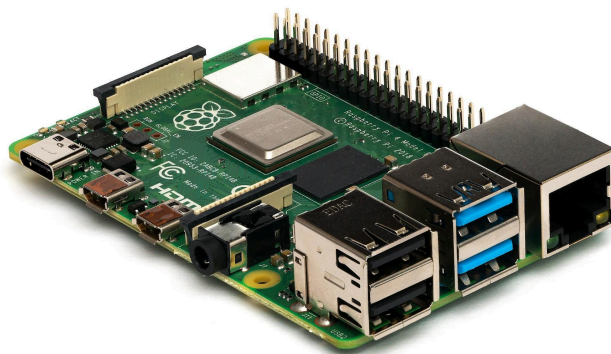
## CHAPTER - 02

### HARDWARE & SOFTWARE REQUIREMENT

#### 3.1 COMPONENTS REQUIRED

##### 3.1.1 Raspberry Pi 3 Model B

Raspberry pi is a credit card sized single board computer which was firstly developed in UK by raspberry pi foundation. Raspberry pi has total 40 pins in which 27 pins is of GPIO (General purpose input and output) and remaining 13 pins are used for VCC and GND. It is the minicomputer which it has inbuilt operating system, but it requires inbuilt SD card for booting and long-term storage. Due to this drawback one can use desktop computer



**FIG 2 - Raspberry Pi 3 Module B**

#### Specification

<b>Processor:</b>	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
<b>Memory:</b>	1GB, 2GB or 4GB LPDDR4 (depending on model)
<b>Connectivity:</b>	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2x USB 3.0 ports 2x USB 2.0 ports.
<b>GPIO:</b>	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
<b>Video &amp; sound:</b>	2x micro-HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port

4-pole stereo audio and composite video port

**Multimedia:** H.265 (4Kp60 decode);  
H.264 (1080p60 decode,1080p30 encode);  
OpenGL ES, 3.0 graphics

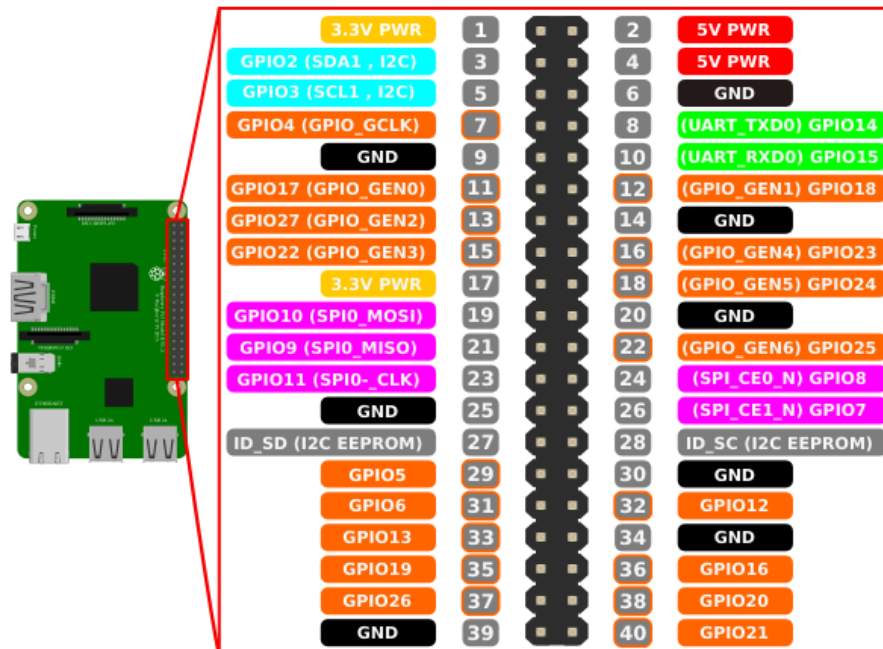
**SD card support:** Micro SD card slot for loading operating system and data storage

**Input power:** 5V DC via USB-Connector (minimum 3A1 )  
5V DC via GPIO header (minimum 3A1 )  
Power over Ethernet (PoE)-enabled  
(Requires separate PoE HAT)

**Environment:** Operating temperature 0-50°C

**Production lifetime:** The Raspberry Pi 3 Model B will remain in production

**FIG 3 - Raspberry pi 3 Model B pin**



### WEB - UI

#### 1. Introduction

This project showcases a home automation system powered by the flexibility of Python and the user-friendly interface of Flask. It allows you to control your electronic appliances remotely using a secure web UI, transforming your living space into a smart and responsive environment.

Leveraging the Raspberry Pi's GPIO pins, this project connects to various devices like bulbs, fans, TVs, and even a motor pump, granting you complete control over their functionality. With the power of AI integrated seamlessly (describe your AI functionalities here), you can experience even higher levels of automation and personalization.

The project features a visually appealing and intuitive web UI built using HTML and CSS. Divided into six sections, each representing a room in your home, the interface provides dedicated controls for each appliance. With a simple toggle button click, you can switch devices on or off, ensuring effortless control from anywhere, anytime.

#### 2. Project Objectives

- Develop a user-friendly and secure web interface for controlling various home appliances (bulbs, fans, ACs, geyser, TV, motor pump) using a smartphone from anywhere.

#### 3. Technical System Overview

##### Hardware:

- Raspberry Pi as the central processing unit.

- GPIO pins connected to electronic devices through relays or transistors.

#### **Software:**

- Python programming language.
- Flask web framework for the web UI.
- RPi.GPIO library for GPIO pin control.
- Flask-BasicAuth library for security

## **4. Functionality & Implementation Details**

#### **Web UI:**

- The web UI (index.html) displays six sections, each representing a room with various appliances.
- Each appliance has a toggle button that sends POST requests to the Flask server. The server processes the requests, extracts the GPIO pin and action ("on" or "off"), and calls the `gpio` function.
- The `gpio` function sets the corresponding pin high (on) or low (off) using RPi.GPIO.

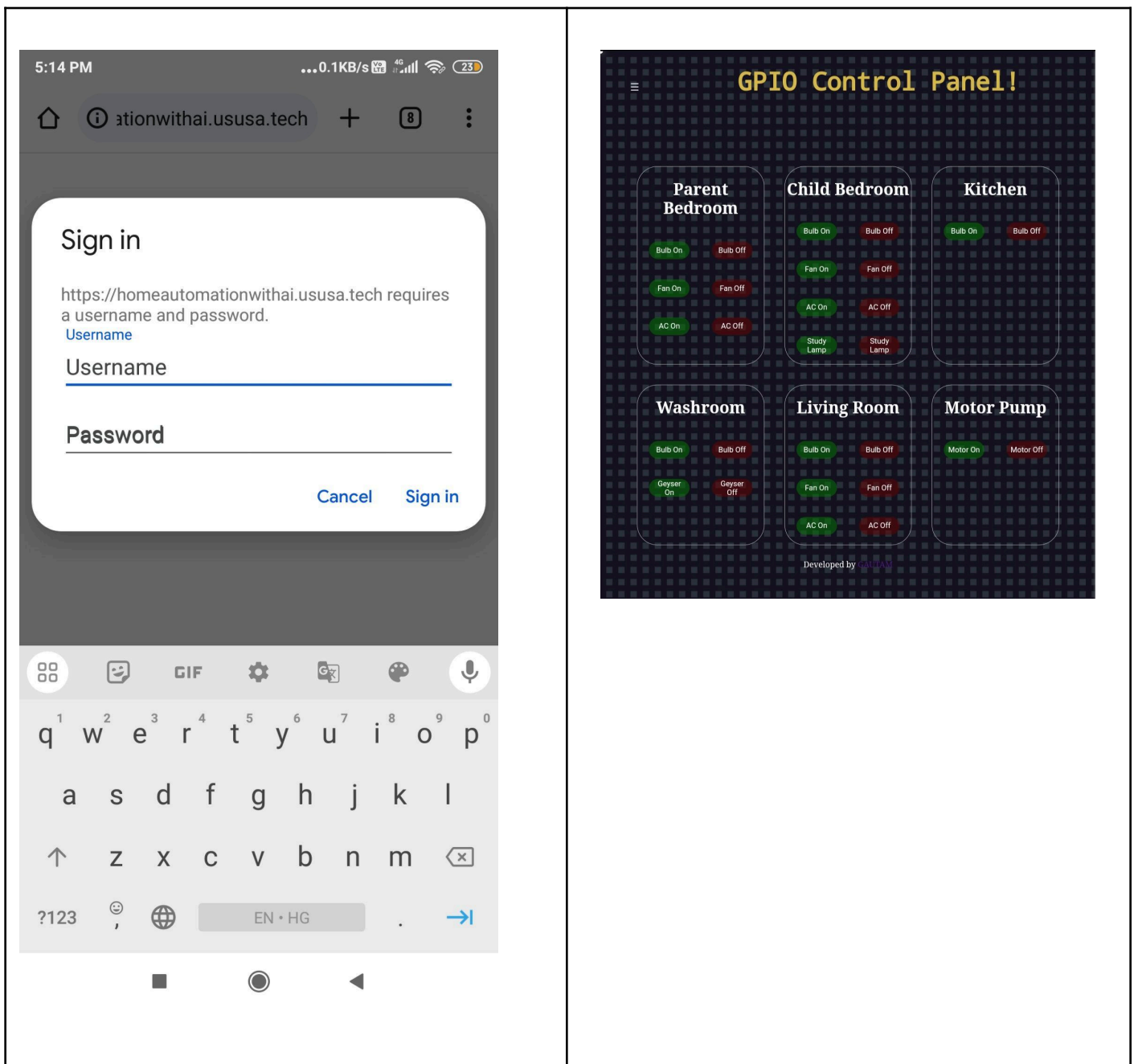
#### **Security:**

To ensure secure access to the web interface, I implemented `Flask-BasicAuth`. Users are prompted to enter their username and password, which are stored in the python program. Only authorized users can control appliances, preventing unauthorized access. The login process is streamlined and user-friendly, requiring minimal interaction. In the future, I plan to potentially explore role-based authorization for more granular control.

## **6. Conclusion**

This project demonstrates the feasibility of building a secure Web-based home automation system using Flask. The project successfully showcases remote control of appliances through a user-friendly and secure web UI.

## 7. screenshots for visualization.



**Note:-** You can access the HTML code in the '**index.html**' file and the Flask application code in the '**app.py**' file on the GitHub repository <https://github.com/GEC-ian/Home-Automation-With-AI>, or scan the QR code provided for direct access.



# CHAPTER - 04

## AI VOICE ASSISTANT

### 1. INTRODUCTION

This project introduces "Jarvis," a powerful and secure AI voice assistant designed to transform your home into an intelligent and responsive environment. Powered by the versatility of Python, the LangChain framework, and the advanced language understanding of Google Gemini Pro, Jarvis offers convenient voice-based control over your electronic appliances.

Using the Raspberry Pi, Jarvis establishes a direct link to devices like lights, fans, and other smart home equipment. The AI's natural language processing capabilities allow for intuitive voice commands, providing a seamless and personalized control experience.

For example, instead of manually flicking a switch, a simple phrase like "Jarvis, turn on the bedroom lights" can transform your space. The AI is also capable of taking command from indirect command and in normal language. Like if a Child says Jarvis I am going to school It will switch off all the equipment of the child room. Jarvis aims to enhance the comfort and automation of your home while maintaining strong privacy through on-device processing.

### 2. Project Objectives

#### Primary Objectives

- Voice-Controlled Home Automation: Enable users to control lights, appliances, and other connected smart devices using natural language voice commands.
- On-Device Intelligence: Utilize the Google Gemini Pro language model, in conjunction with LangChain, for localized language processing and command interpretation. This prioritizes user privacy and reduces reliance on cloud services.
- Intuitive User Experience: Design voice interactions that are simple, efficient, and resemble natural conversation, minimizing the need for rigid command structures.

#### Secondary Objectives (Potential for Expansion)

- Customization: Allow users to define personalized commands or aliases for devices (e.g., "Jarvis, activate movie mode" could dim lights and turn on the TV).
- Routines: Implement the ability to create routines, where a single command triggers a sequence of actions across multiple devices (e.g., "Jarvis, good night" could turn off lights, lock doors, and lower the thermostat).
- Proactive Assistance: Explore ways for Jarvis to provide helpful suggestions or reminders based on context (e.g., "It's getting late, would you like me to turn on the bedroom lights?").
- Device Compatibility: Continuously expand the types of supported smart home devices and control protocols for greater versatility.

### 3. Technical System Overview

The technical system comprises several components working together to achieve the desired functionality:

- **LLM Gemini Pro with Langchain:** Utilized for natural language processing and understanding user commands.
- **Google Generative AI:** Integrated for generating responses to user queries and commands in a conversational manner.
- **Porcupine Wake Word Detection:** Implemented to wake the AI assistant upon hearing a specific wake word, ensuring hands-free interaction.
- **Raspberry Pi 3 Model B:** Serves as the hardware platform for hosting the AI assistant, ensuring portability and affordability.
- **Edge TTS (Text-to-Speech):** Used for converting text responses generated by the AI assistant into audible speech for user feedback.
- **On/Off Module:** Controls the switching on and off of household equipment based on the commands received from the user via the voice assistant.



## 4. Functionality & Implementation Details

The functionality of the system is achieved through the following steps:

- **Speech Recognition:** Utilizing the SpeechRecognition library, the system listens for user commands spoken to the AI assistant.
- **Natural Language Understanding:** Langchain with LLM Gemini Pro processes the user's speech to understand the intent behind the command.
- **Response Generation:** Google Generative AI generates appropriate responses or actions based on the user's command.
- **Wake Word Detection:** Porcupine detects a predefined wake word, activating the AI assistant to listen to the user's command.
- **Device Control:** Upon understanding the user's command, the system interfaces with the On/Off module to switch the designated household equipment on or off accordingly.
- **Feedback Generation:** Edge TTS converts the AI assistant's response into audible speech, providing feedback to the user regarding the execution of their command.

## 5. Conclusion:

The project successfully demonstrates the integration of AI technologies into home automation, offering users a convenient and intuitive means of controlling household equipment through voice commands. By leveraging Raspberry Pi and various software libraries and APIs, the system achieves efficient and cost-effective implementation. Further enhancements could include expanding the range of supported commands, improving natural language understanding, and integrating additional smart devices for broader home automation capabilities. Overall, the project showcases the potential of AI-driven solutions in enhancing everyday tasks and improving quality of life.

**Note:-** You can see the code used in making Jarvis AI on github by going to this link

[https://github.com/mdarman4/HomeAutomationWithAI\\_Jarvis](https://github.com/mdarman4/HomeAutomationWithAI_Jarvis)

Or Scan the QR code

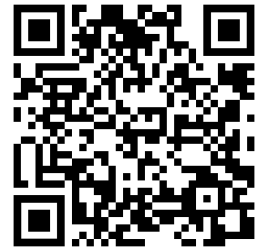


Fig. Speaker

## CHAPTER - 05

### TELEGRAM CHATBOT

#### INTRODUCTION

The project introduces an AI-driven chatbot developed using LLM Gemini Pro with Langchain, aimed at streamlining home automation tasks through conversational interactions. Running on the Raspberry Pi 3 Model B, Jarvis enables users to effortlessly control household equipment with text inputs. Leveraging Python, the project integrates various libraries and APIs to ensure smooth functionality and seamless interaction.

#### Project Objectives:

The primary objectives of the project are outlined as follows:

- Develop an AI-driven chatbot capable of comprehending natural language commands for home automation tasks.
- Utilize LLM Gemini Pro with Langchain to effectively process and interpret user commands.
- Integrate the Telegram API to facilitate communication and command reception from users via text inputs.
- Implement the system on Raspberry Pi 3 Model B to ensure portability and accessibility.
- Enable Jarvis to efficiently control household equipment based on user commands, enhancing convenience and efficiency in home management.

#### Technical System Overview:

The technical system comprises several components working together to achieve the desired functionality:

- **LLM Gemini Pro with Langchain:** Utilized for natural language processing, enabling Jarvis to accurately understand and interpret user commands.

- **Telegram API:** Integrated for seamless communication between users and the AI chatbot, facilitating command reception and response delivery through text inputs.
- **Raspberry Pi 3 Model B:** Acts as the hardware platform for hosting the AI chatbot, ensuring efficient execution and portability.
- **On/Off Module:** Controls the switching on and off of household equipment based on the commands received from users via the Telegram interface.
- **Python Libraries:** Various Python libraries such as Telegram, Langchain, and on\_off are utilized for seamless integration and functionality within the system.

### Functionality & Implementation Details:

The functionality of the system is implemented through the following steps:

- **Command Reception:** Utilizing the Telegram API, the system listens for incoming commands from users, whether through text inputs.
- **Natural Language Processing:** LLM Gemini Pro with Langchain processes and interprets user commands, extracting the intent behind the request effectively.
- **Response Generation:** Based on the interpreted command, Jarvis generates appropriate responses and executes the corresponding actions.
- **Device Control:** The on\_off module interfaces with household equipment, enabling Jarvis to switch them on or off based on user commands.
- **Communication Interface:** Telegram serves as the interface through which users interact with Jarvis, providing a convenient platform for issuing commands and receiving feedback.

### Conclusion:

In conclusion, the project successfully demonstrates the integration of AI technologies and home automation principles to develop Jarvis, an AI-driven chatbot capable of controlling household equipment through conversational interactions. By leveraging Raspberry Pi and Python programming, the system achieves efficient execution and seamless integration,

enhancing convenience and automation in home management tasks. Future enhancements could include expanding the range of supported commands, improving natural language understanding, and integrating additional features for enhanced functionality and user experience. Overall, Jarvis represents a significant advancement in the field of AI-driven home automation, offering users a convenient and intuitive solution for managing their homes effectively.

**Note:-** You can see the code used in making Telegram Chat Bot on github by going to this link

[https://github.com/mdarman4/HomeAutomationWithAI\\_telegramchab](https://github.com/mdarman4/HomeAutomationWithAI_telegramchab)

Or Scan the QR code

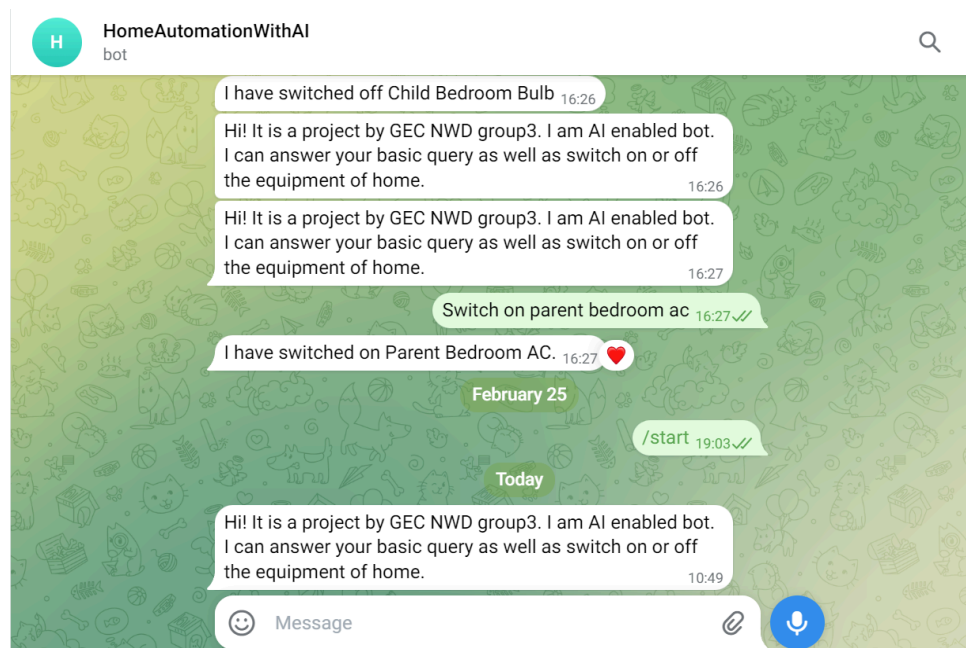


Fig. Telegram ChatBot



# Detailed Hardware Report

This document details the hardware design and implementation of my home automation project using a Raspberry Pi. The project aims to demonstrate remote control of various electronic appliances, including lights, fans, a water pump, and a TV, utilizing the Raspberry Pi's GPIO pins and readily available components.

### Required Materials:

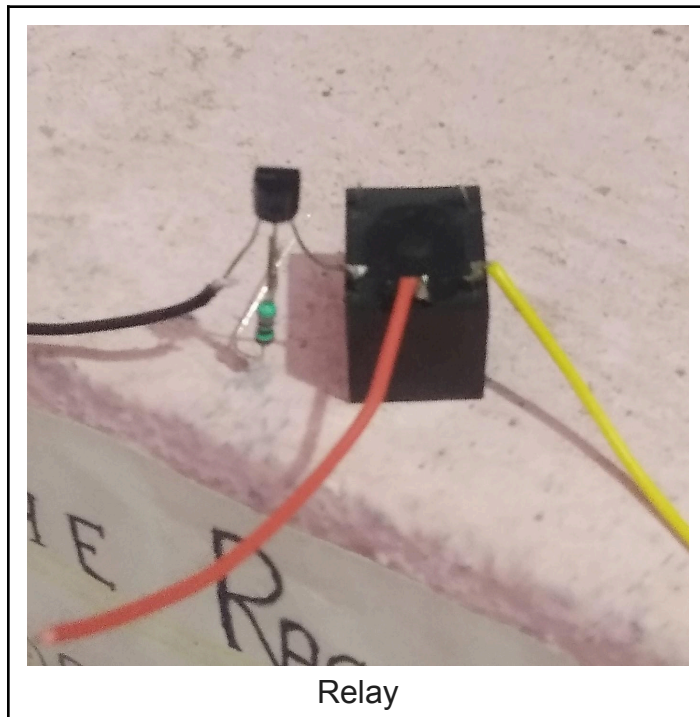
- Raspberry Pi 3 Model B
- Cardboard box
- Chart paper
- 5 different color wires
- 3 Mini fans
- 3 5V DC motors
- 1 5V DC water pump
- 4 5V relays
- 4 transistors (e.g., NPN 2N2222)
- 4 1k ohm resistors
- 11 LED bulbs (1.5-3V, different colors)
- Switch

### Hardware Assembly:

1. **Base Structure:** I constructed a base platform using cardboard and chart paper to provide a visual representation of a room and organize the components.
2. **LED Bulbs:** Each LED bulb was connected to a separate GPIO pin via its positive lead. The negative leads of all bulbs were connected to a common

ground wire and then to a ground pin on the Raspberry Pi. This ensures individual control of each bulb.

3. **Relay Modules:** I utilized five 5-pin relays to control higher-power appliances like fans, motors, and the water pump. Each relay was connected as follows:
- **Transistor:** The transistor's emitter pin was connected to the Raspberry Pi's ground, the base pin to a GPIO pin (with a 1k ohm resistor for current limiting), and the collector pin to the relay's control pin.
  - **Relay Coil:** The relay coil was powered by a 5V power bank, with its positive lead connected to the relay's middle pin and the negative lead to the Raspberry Pi's ground.
  - **Appliance:** The appliance (e.g., fan motor) was connected to the relay's normally closed (NC) pin and 5V power bank(-)ve pin. When the Raspberry Pi sends a signal to the GPIO pin, the relay activates, connecting the appliance to the power supply.





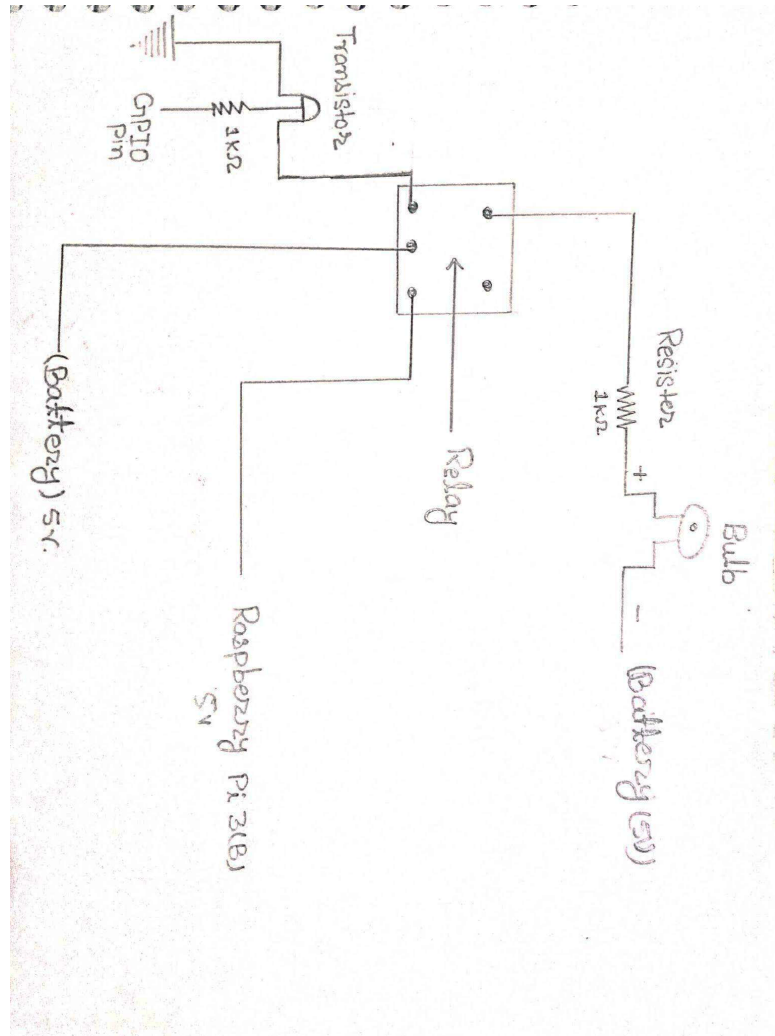


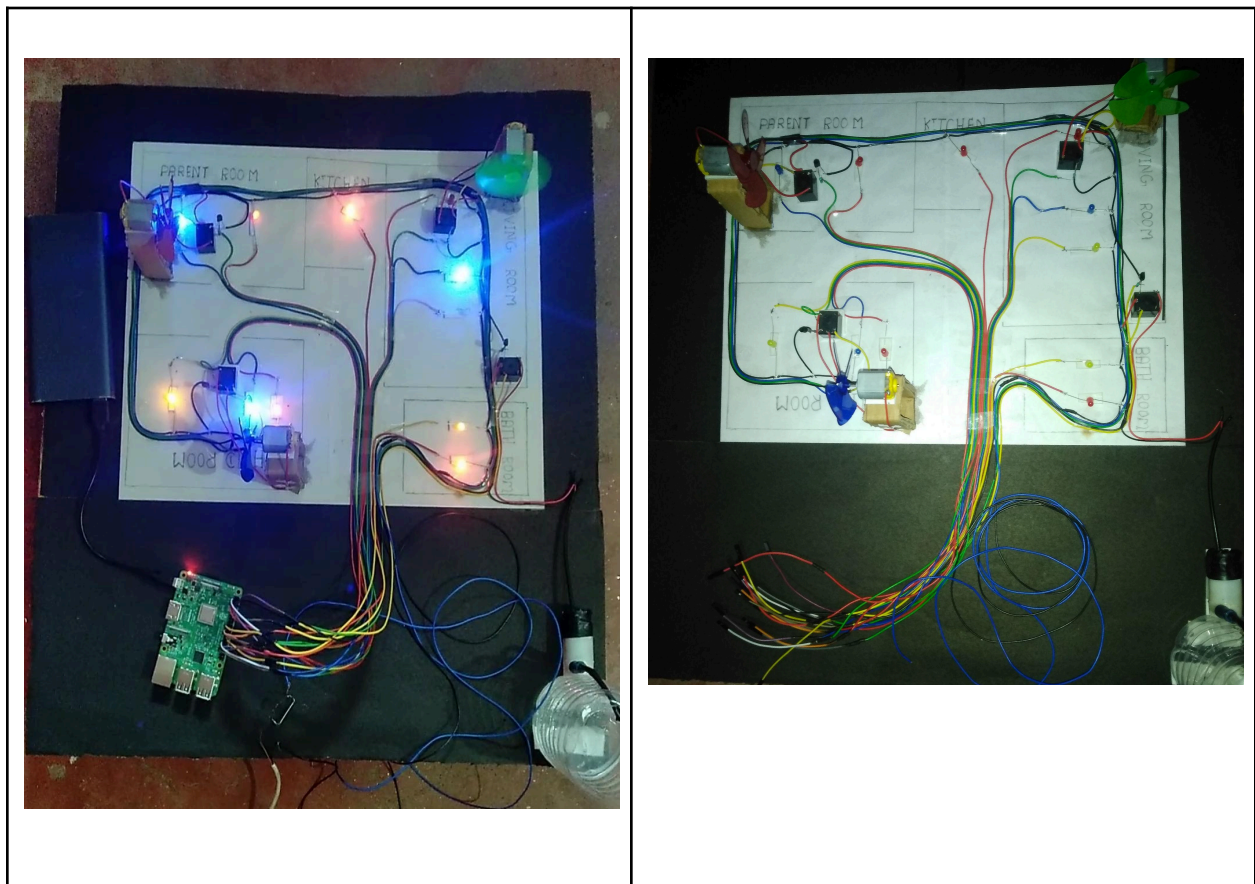
Fig. RELAY Connection Diagram

4. **Water Pump:** The water pump connection followed the same principle as the fans and motors, using a relay and transistor controlled by a GPIO pin.
5. **Power Supply:** The Raspberry Pi and relays were powered by a 5V power bank, ensuring sufficient power for all components.

### Testing and Functionality:

Once assembled, I connected the Raspberry Pi to a web interface or command-line interface and wrote Python code to control the GPIO pins. This allowed me to turn individual LED bulbs on and off, activate fans and motors, and control the water pump remotely. The project successfully demonstrated the control of various appliances using the Raspberry Pi 3 Model B, showcasing the potential of home automation projects.

### Image for visualization:



### Future Improvement / Future Plan

For future enhancements of the project, I have several plans:

- 1. Integration of Fire Alarms:** I intend to incorporate fire alarms into the system to enhance safety measures within the automated home environment. Fire alarms will be strategically placed to detect smoke or fire, triggering alerts and appropriate actions.
- 2. Integration of Security Cameras:** Implementing security cameras will provide real-time monitoring and surveillance of the premises. This addition will enable homeowners to remotely view live footage of their property, enhancing security and peace of mind.
- 3. Integration of Temperature Sensors:** Including temperature sensors will enable the system to monitor and regulate indoor temperature levels effectively. These sensors will contribute to creating a comfortable and energy-efficient environment by adjusting heating and cooling systems as needed.
- 4. Implementation of a Single Slider Button:** Instead of separate on/off buttons for each appliance, I plan to introduce a single slider button interface. This interface will display the current state of GPIO pins, allowing users to control multiple appliances with ease and efficiency. The slider button will provide a seamless and intuitive user experience, simplifying the operation of the automated home system.
- 5..Effortless Customization:** The modular design allows for easy addition or removal of appliances as needed, ensuring flexibility.

These Future enhancements aim to enhance functionality, safety, and user experience by integrating advanced features like fire alarms, security cameras, temperature sensors, and a single slider button interface. This will make the automated home system more versatile,

intelligent, and user-friendly, aligning with the goal of creating a comprehensive and innovative solution for modern living needs.