# Vidya Jyothi Institute of Technology
# (An Autonomous Institution)
# Aziz Nagar, Hyderabad -500075

A Project Report

On

## DESIGN AND SOFTWARE DEVELOPMENT OF AI ROBOT

Submitted for partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRICAL AND ELECTRONICS ENGINEERING**

**BY**


**Mr. A GANESH (21915A0201)**

**Ms. E AJITHA (21915A0207)**

**Mr. G RAHUL (21915A0208)**

**Ms. G GANANAND GOUD (21915A0209)**


Under the guidance of


**Dr. A.SRUJANA**

Professor

Department of Electrical and Electronics Engineering

VJIT, Hyderabad


**Department of Electrical and Electronics Engineering**

# VIDYA JYOTHI INSTITUTE OF TECHNOLOGY
## (Autonomous)

## Himayathnagar (vi), C.B. Post, R.R Dist. 500075

## CERTIFICATE

This is to certify that the project work entitled "**DESIGN AND SOFTWARE DEVELOPMENT OF AI ROBOT"** is a bonafide work carried out by **Mr. A GANESH (21915A0201) Ms. E AJITHA (21915A0207) Mr. G RAHUL (21915A0208) Mr. G GANANAND GOUD (21915A0209)** in partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY IN ELECTRICAL AND ELECTRONICS ENGINEERING to be awarded by** the **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY**, Hyderabad.

The content in this report has not been submitted to any other university or institute for the award of any degree or diploma.

| | |
|---|---|
| Internal Supervisor | Head of the Department |
| Department of EEE | Department of EEE |
| Hyderabad | Hyderabad |

**External Examiner**

# DECLARATION

This is to certify that the work reported in the present project entitled "**DESIGN AND SOFTWARE DEVELOPMENT OF AI ROBOT**" is a record of work done by us in the Department of Electrical and Electronics Engineering, Vidya Jyothi Institute of Technology(Autonomous), Jawaharlal Nehru Technological University, Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source.

## ACKNOWLEDGEMENT

We would like to express my sincere gratitude and indebtedness to my project supervisor **Dr. A.SRUJANA** for his valuable suggestions and interest throughout the course of this project

We are also thankful to Head of the department Electrical and Electronics Engineering for providing excellent infrastructure and a nice atmosphere for completing this prject successfully

We convey my heartfelt thanks to the lab staff for allowing me to use the required equipment whenever needed. Finally, we would like to take this opportunity to thank my family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work

# ABSTRACT

The primary objective of developing humanoid robots is to enable them to interact with the world in a manner similar to humans. This involves not only replicating human physical appearance but also incorporating advanced sensors, sophisticated control systems, and artificial intelligence algorithms to perform tasks, move gracefully, and even communicate with humans.

The abstract features of a humanoid robot typically include a human-like appearance, with a head, torso, arms, and legs, often equipped with sensors such as cameras, microphones, and touch sensors to perceive the environment and interact with it. Actuators such as motors and joints enable these robots to move and perform tasks with a wide range of dexterity and mobility. Humanoid robots are often controlled by advanced AI algorithms and computer systems, allowing them to make decisions, recognize objects and people, and adapt to changing situations.

Applications of humanoid robots are diverse and can include tasks in industries such as healthcare, manufacturing, entertainment, research, and education.

Overall, humanoid robots represent a cutting-edge field of robotics that seeks to bridge the gap between machines and humans, offering the potential to revolutionize industries and enhance our daily lives through their versatility and adaptability.

# LIST OF FIGURES

# TABLE OF CONTENT

# CHAPTER-1

## 1.1 Introduction

A humanoid robot is a type of robot designed to resemble the human body in terms of its physical structure and appearance. These robots are often equipped with features such as a head, torso, arms, and legs, which mimic the human form. The primary goal of humanoid robotics is to create machines that can perform tasks in a manner similar to how a human would, both in terms of physical movement and interaction with the environment.

Humanoid robots are complex anthropomorphic artificial machines. The growing interest in humanoid robots accompanied by the latest and ever-increasing technological advancements in the field of robotics, locomotion, and AI, achieved by engineers, has speeded up their development over the past decade. Moreover, because of their human-like shape, these robots can use the same equipment and environment as humans, hence, making them more compatible to be used as a building platform for the physical implementation of the Digital Twin.

Today's humanoid robots come in different shapes and sizes and are extensively being used for research and space exploration, personal assistance and care-giving, education and entertainment, search and rescue operations, manufacturing and maintenance, public relations, and most importantly healthcare sector.

A humanoid robot is a robot resembling the human body in shape. The design may be for functional purposes, such as interacting with human tools and environments, for experimental purposes, such as the study of bipedal locomotion, or for other purposes. In general, humanoid robots have a torso, a head, two arms, and two legs, though some humanoid robots may replicate only part of the body, for example, from the waist up. Some humanoid robots also have heads designed to replicate human facial features such as eyes and mouths. Androids are humanoid robots built to aesthetically resemble humans

Characteristics of Humanoids – Self-maintenance, Autonomous learning, Avoiding harmful situations to people, property, and itself , Safe interacting with human beings and the environment , Legged locomotion

**Role Of Humanoid Robots In Society**

Humanoid robots are playing an increasing role in society, and their impact will only continue to grow. These machines can interact with people and learn quickly, making them valuable to many industries. Here are some ways humanoid robots are being used in society today

Humanoid robots in healthcare can help patients by performing tasks such as picking up objects or assisting with basic medical procedures. This technology has the potential to improve patient care and reduce costs significantly.

Humanoid robots in manufacturing: They can help workers by performing tasks such as moving heavy objects or handling dangerous chemicals. This technology can save companies money and improve workplace safety.

Humanoid robots in the military can help soldiers carry out dangerous tasks such as navigating rugged terrain or attacking enemy targets. This technology has the potential to save lives and improve combat efficiency.

# CHAPTER-2
# COMPONENTS DETAILS

## 2.1 Raspberry Pi4

The Raspberry Pi 4 (R Pi 4) is a versatile and powerful single-board computer that has found extensive use in robotics applications. Its compact size, low cost, and community support make it an attractive choice for hobbyists, students, and professionals working on various robotic projects.

Control and Processing: The R Pi 4 serves as the central processing unit for many robots. It can run a variety of operating systems, including Raspberry Pi OS (formerly Arabian), Ubuntu, and others, which enables it to perform tasks like sensor data processing, motor control, and decision-making

Camera Vision: The R Pi 4's camera module and USB camera support are often used for computer vision tasks in robotics. Robots can capture images and videos, perform object recognition, image processing, and even use cameras for navigation and mapping

FIG 2.1(RASPBERRY PI4)

Robot Operating System (ROS): ROS, a popular robotics middleware framework, is compatible with the R Pi 4. It provides tools and libraries for developing complex robot control systems, making it easier to build and program robots.

Io T Integration: Raspberry Pi 4's Wi-Fi and Ethernet capabilities make it suitable for IoT (Internet of Things) integration in robotics. Robots can communicate with cloud services, receive remote commands, and send data for analysis.

## 2.2 Arduino Mega

The Arduino Mega is a micro-controller board that is often used in robotics due to its versatility, ample I/O pins, and compatibility with a wide range of sensors and actuators. It's part of the Arduino family and is known for its enhanced capabilities compared to the standard Arduino boards. Here's how the Arduino Mega is used in robotics:

Control Hub: The Arduino Mega can serve as the central control hub for a robot. It can process sensor data, control motors, servos, and other actuators, and make real-time decisions based on the robot's programming

Motor Control: The board has multiple PWM (Pulse Width Modulation) pins, making it suitable for controlling DC motors, stepper motors, and servo motors used in robot motion and manipulation


FIG 2.2 (ARDUINO MEGA)

Machine Learning: While the Arduino Mega has limited computational power compared to more advanced platforms, it can still be used for simple machine learning tasks like pattern recognition or gesture detection.

IoT (Internet of Things) Integration: The Arduino Mega can be used to connect robots to IoT platforms, enabling data collection, remote monitoring, and control of robots over the internet.

Human-Robot Interaction: With the addition of user interface components like buttons, touch screens, or Capacitive touch sensors, the Arduino Mega can facilitate human-robot interaction in applications such as tele presence robots or interactive exhibits.

## 2.3 Pi Camera

The Pi camera module, often used with Raspberry Pi boards, is a versatile camera module that can be beneficial in various robotics applications. It allows you to capture high-quality images and videos, making it a valuable component for robots that require visual perception, navigation, and object recognition.

Vision-Based Navigation: Robots can use the Pi camera to capture images or video frames and process them for navigation purposes. This can involve obstacle avoidance, path planning, and SLAM (Simultaneous Localization and Mapping) tasks, enabling the robot to move autonomously in its environment.

FIG 2.3 (PI CAMERA)

Object Detection and Recognition: The Pi camera can be used in conjunction with computer vision libraries like Open CV to detect and recognize objects in the robot's surroundings. This is particularly useful for tasks such as picking and placing objects or identifying specific items.

Gesture Control: Robots can be programmed to recognize hand gestures or body movements captured by the camera, allowing users to control the robot's actions through gestures.

## 2.4 Microphone

A microphone is a small input device that is used for multiple reasons like to record, communicate, or for voice recognition into a system. Connecting a microphone with a Raspberry Pi can allow users to create a wide range of projects like speech recognition projects, AI projects, Voice assistance applications

FIG 2.4 (MICROPHONE)

In robotic applications, dynamic microphones are often preferred when ruggedness, low power consumption, and resistance to environmental factors are crucial. They are commonly used for sound localization in robotics and can withstand challenging conditions

Condenser microphones, on the other hand, are chosen when high audio quality, sensitivity, and small form factors are required. They are suitable for applications like human-robot interaction, voice commands, and applications where capturing subtle acoustic cues is important.

Ultimately, the choice between dynamic and condenser microphones in robotics will depend on the specific needs of the robot and the intended tasks it is designed to perform.

# CHAPTER-3

## 3.1 Software

### 3.1.1 Arduino ide

The Arduino IDE (Integrated Development Environment) is the software used to write, compile, and upload code to Arduino micro controller boards.

Open Source: The Arduino IDE is an open-source software, which means its source code is freely available for the public to view, modify, and distribute. This openness encourages collaboration and the sharing of knowledge within the Arduino community.

Cross-Platform Compatibility: The Arduino IDE is compatible with various operating systems, including Windows, mac OS, and Linux. This cross-platform support allows users to write code on their preferred operating system.

Library Management: Arduino IDE includes a Library Manager, which simplifies the process of adding external libraries to your projects. Libraries contain pre-written code that you can use in your own sketches to interface with sensors, displays, and other components.

Board Manager: The IDE allows users to select the type of Arduino board they are using and install board-specific libraries and tools. This is essential because different Arduino boards may have different specifications and require different settings.

Serial Monitor: Arduino projects often involve serial communication between the micro controller and a computer. The IDE includes a Serial Monitor tool that allows users to send and receive data between the Arduino board and the computer, aiding in debugging and monitoring.

Version Control: Users can save and manage different versions of their projects directly within the Arduino IDE, facilitating version control and project organization.

### 3.1.2 Thonny

Thonny is an integrated development environment (IDE) for the Python programming language. It is designed to be beginner-friendly while also providing advanced features for experienced developers. Thonny comes with built-in support for managing Python packages, virtual environments, and a simple interface for writing and running Python code.

Integrated Package Manager: Thonny includes a built-in package manager, making it easy to install, update, and manage Python packages directly from the IDE.

Virtual Environment Support: Thonny facilitates the creation and management of virtual environments, allowing developers to isolate project dependencies and avoid conflicts between different projects.

Simple Interface: Thonny is designed to have a clean and straightforward interface, making it suitable for beginners. It provides a user-friendly experience for writing and running Python code.

Integrated REPL (Read-Eval-Print Loop): Thonny includes an integrated REPL, allowing users to execute Python code interactively and see the results immediately.

Platform Support: Thonny is available for Windows, mac OS, and Linux, making it accessible across different operating systems.

## 3.2 Libraries

### 3.2.1 Speech Recognition

Speech Recognition (formerly known as Speech Recognition):

Speech Recognition is a Python library that provides simple and convenient access to various speech recognition engines, including Google Web Speech API, Sphinx, and more.

It's easy to use and supports multiple recognition engines, making it a versatile choice.

Speech Recognition is a popular Python library for performing speech recognition and converting spoken language into text. It provides a straightforward way to integrate speech recognition capabilities into your Python applications. Here's a brief overview of how to use the Speech Recognition library:

Installation:

You can install the Speech Recognition library using pip:

bash

pip install Speech Recognition

**Basic Usage:**

Here's a basic example of how to use Speech Recognition to recognize speech from an audio file:

**Python**

```
import speech_recognition as sr
# Initialize the recognizer
recognizer = sr.Recognizer()
# Load an audio file
audio_file = "audio.wav"
# Use the recognizer to recognize speech from the audio file
with sr.AudioFile(audio_file) as source:
    audio_data = recognizer.record(source)
    try:
        # Recognize the speech using the Google Web Speech API
        text = recognizer.recognize_google(audio_data)
        print("Recognized text:", text)
    except sr.UnknownValueError:
        print("Could not understand the audio")
    except sr.RequestError as e:
        print("Error connecting to the Google Web Speech API: {0}".format(e))
```

**Example:**

We import the speech_recognition library as sr.

We create a recognizer object, recognizer, which will be used to recognize speech.

We load an audio file, audio.wav, which contains the spoken language.

We use the recognizer's record method to capture the audio data from the audio file.

We then try to recognize the speech using the Google Web Speech API (recognize_google). You can choose from various recognition engines, including Sphinx, by specifying the appropriate method.

The speech_recognition library provides a range of recognition engines and options, such as using different APIs, working with microphone input, adjusting recognition parameters, and more. It's a versatile library that can be used in a wide range of

applications where speech recognition is required, from transcription services to voice-controlled applications.

### 3.2.2 Pyttsx3

pyttsx3 is a Python library that provides a simple and cross-platform way to convert text to speech (TTS). It is often used to add speech synthesis capabilities to Python applications, making it possible to have the computer read text aloud. pyttsx3 supports multiple TTS engines, making it versatile for various operating systems. Here's a basic overview of using pyttsx3

Installation:

You can install pyttsx3 using pip:

bash

pip install pyttsx3

**Basic Usage:**

Here's an example of how to use pyttsx3 to convert text to speech:

**Python**

```
import pyttsx3
# Initialize the TTS engine
engine = pyttsx3.init()
# Convert text to speech
text = "Hello, I am pyttsx3, a text-to-speech library for Python."
engine.say(text)
# Wait for the speech to finish
engine.runAndWait()
```

**Example:**

We import the pyttsx3 library.

We initialize the TTS engine with pyttsx3.init().

We specify the text that we want to convert to speech using the engine.say(text) method.

We use engine.runAndWait() to wait for the speech to finish playing.

Voice and Configuration:

pyttsx3 allows you to configure various properties of the TTS engine, such as voice selection, speech rate, and volume.

**Python**

```
import pyttsx3
engine = pyttsx3.init()
# Set the speech rate (words per minute)
engine.setProperty("rate", 150)
# Set the volume (0.0 to 1.0)
engine.setProperty("volume", 0.9)
# List available voices and select one
voices = engine.getProperty("voices")
engine.setProperty("voice", voices[1].id)  # Choose a specific voice from the list
# Convert text to speech
text = "This is a customized text-to-speech example."
engine.say(text)
# Wait for the speech to finish
engine.runAndWait()
```

Choosing a Voice:

pyttsx3 provides a list of available voices on your system. You can iterate through the list and select a voice that you prefer. Voice availability and voice names may vary depending on your operating system.

**Python**

```
import pyttsx3
engine = pyttsx3.init()
# List available voices
voices = engine.getProperty("voices")
# Print available voices and their IDs
for voice in voices:
    print("Voice ID:", voice.id)
    print("Voice Name:", voice.name)
    print("Language:", voice.languages)
    print("Gender:", voice.gender)
    print("Age:", voice.age)
    print("")
# Choose a specific voice by its ID
```

```
engine.setProperty("voice", voices[0].id)
# Convert text to speech
text = "This is an example of selecting a specific voice."
engine.say(text)
# Wait for the speech to finish
engine.runAndWait()
```

pyttsx3 is a useful library for adding speech synthesis capabilities to Python applications, whether for accessibility purposes or for creating voice-based applications.

### 3.2.3 PyAudio

PyAudio is a Python library that provides a way to work with audio input and output streams. It's commonly used for audio-related tasks, such as audio recording, audio playback, audio processing, and real-time audio streaming. PyAudio can be used to interact with microphones, speakers, and other audio devices, making it a valuable tool for a wide range of audio applications.

Here are some key features and use cases of PyAudio:

Audio Input and Recording: PyAudio allows you to record audio from microphones and other audio sources. You can specify recording parameters such as sample rate, audio format, and channel configuration.

Audio Output and Playback: You can use PyAudio to play audio on speakers and headphones. It supports various audio formats and configurations for playback.

Real-Time Audio Processing: PyAudio can be used in conjunction with other audio processing libraries to perform real-time audio analysis, synthesis, filtering, and manipulation.

Streaming and Callbacks: PyAudio supports callback functions, making it suitable for real-time audio streaming and processing applications.

Cross-Platform: It is cross-platform and works on Windows, macOS, and Linux, making it suitable for multi-platform audio applications.

Integration with NumPy: PyAudio can easily be integrated with NumPy for efficient data manipulation and analysis of audio data.

Simple and Convenient API: The PyAudio API is straightforward to use, and it provides a high-level interface for common audio tasks.

A basic example of how to use PyAudio to record audio from a microphone:

**Python**

```
import pyaudio
import wave
# Initialize PyAudio
audio = pyaudio.PyAudio()
# Set recording parameters
format = pyaudio.paInt16
channels = 1
sample_rate = 44100
chunk = 1024
record_seconds = 5
output_filename = "recorded_audio.wav"
# Open a microphone stream
stream = audio.open(format=format, channels=channels,
                    rate=sample_rate, input=True,
                    frames_per_buffer=chunk)
print("Recording...")
frames = []
# Record audio data
for _ in range(0, int(sample_rate / chunk * record_seconds)):
    data = stream.read(chunk)
    frames.append(data)
# Close the microphone stream
stream.stop_stream()
stream.close()
# Terminate PyAudio
audio.terminate()
# Save recorded audio to a WAV file
with wave.open(output_filename, 'wb') as wf:
    wf.setnchannels(channels)
    wf.setsampwidth(audio.get_sample_size(format))
    wf.setframerate(sample_rate)
    wf.writeframes(b".join(frames))
```

print("Recording saved to", output_filename)

**Example:**

We import pyaudio and use it to initialize PyAudio.

We set various recording parameters like format, channels, sample rate, chunk size, and recording duration.

We open a microphone stream using the specified parameters.

We record audio frames in a loop and store them in a list.

We save the recorded audio to a WAV file.

PyAudio is a versatile library that can be used for a wide range of audio-related tasks, including voice recognition, audio synthesis, music composition, and real-time audio processing

### 3.2.4 Pywhatkit

pywhatkit is a Python library that provides a simple interface for performing various tasks, such as sending WhatsApp messages, performing Google searches, converting text to handwriting, and more. It allows you to automate these tasks using Python scripts.Here are some common tasks that you can perform using pywhatkit:

Sending WhatsApp Messages: You can use pywhatkit to send WhatsApp messages to a specified contact with a given message.

**Python**

```
import pywhatkit as kit
kit.sendwhatmsg("+1234567890", "Hello, this is a test message.", 15, 0)
```

In this example, the function sendwhatmsg is used to send a WhatsApp message to the phone number specified in the first argument. The message is sent after a delay of 15 seconds (as specified in the second argument).

Google Search: You can perform Google searches directly from your Python script.

**Python**

```
import pywhatkit as kit
kit.search("Python programming")
```

This will open a web browser and perform a Google search for "Python programming."

Convert Text to Handwriting: pywhatkit can convert text to a handwritten style and save it as an image.

**Python**

```
import pywhatkit as kit
kit.text_to_handwriting("Hello, this is a handwritten message.")
```

This will generate a handwritten image containing the specified text.

Play YouTube Videos: You can use pywhatkit to play a YouTube video in your default web browser.

**Python**

```
import pywhatkit as kit
kit.playonyt("Python tutorial")
```

This will search YouTube for "Python tutorial" and play the most relevant video.

Other Features: pywhatkit also provides functions for performing tasks like converting text to speech, finding information about a topic on Wikipedia, and more.

Before using pywhatkit, make sure to install it using pip:

**pip install pywhatkit**

Keep in mind that pywhatkit interacts with external services and may rely on web scraping, so it's important to use it responsibly and in accordance with the terms and conditions of the services it accesses. Additionally, the library may receive updates and changes over time, so it's a good idea to refer to the official documentation for the most up-to-date information on its usage and capabilities.

### 3.2.5 Pyjokes

pyjokes is a Python library that provides a collection of jokes and one-liners. It's a fun and lightweight library that you can use to add humor to your Python scripts or applications. pyjokes comes with a variety of jokes in different categories and allows you to generate random jokes or specific types of jokes. Here's how you can use it:

**Installation:**

You can install pyjokes using pip:

Bash

```
pip install pyjokes
```

**Basic Usage:**

simple example of how to use pyjokes to get random jokes:

**Python**

```
import pyjokes
```

```
# Get a random joke
joke = pyjokes.get_joke()
# Print the joke
print(joke)
```

When you run this code, it will display a random joke from the pyjokes collection.

**Categories:**

pyjokes provides jokes in different categories, such as "neutral," "chuck," "all," "twister," and more. You can specify a category when fetching a joke:

**Python**

```
import pyjokes
# Get a random Chuck Norris joke
chuck_joke = pyjokes.get_joke(category='chuck')
# Print the Chuck Norris joke
print(chuck_joke)
```

**Customization:**

You can customize the jokes by specifying options like language and the number of jokes to retrieve:

**Python**

```
import pyjokes
# Get a French joke
french_joke = pyjokes.get_joke(language='fr')
# Get multiple jokes at once
jokes = pyjokes.get_jokes(category='neutral', language='en', num_jokes=3)
# Print the jokes
for joke in jokes:
    print(joke)
```

pyjokes is a lighthearted library that can be used to inject humor into your Python projects or just for some fun during development. It's often used in applications where a touch of humor can enhance the user experience, such as chatbots, terminal applications, or scripts meant for entertainment.

### 3.2.6 Wikipedia

To interact with Wikipedia programmatically in Python, you can use the "wikipedia-api" library, which allows you to access and retrieve data from Wikipedia. This library provides a simple and convenient way to access Wikipedia articles, page content, and other related information. Here's how to use the "wikipedia-api" library:

Installation:

You can install the "wikipedia-api" library using pip:

bash

pip install wikipedia-api

**Basic Usage:**

Here's an example of how to use the "wikipedia-api" library to retrieve information from Wikipedia:

**Python**

```
 import wikipediaapi
# Create a Wikipedia API object
wiki_wiki = wikipediaapi.Wikipedia('en')
# Get the summary of a Wikipedia page
page = wiki_wiki.page("Python (programming language)")
print("Title:", page.title)
print("Summary:")
print(page.summary)
```

**Example:**

We import the "wikipediaapi" library and create a Wikipedia API object for the English Wikipedia using wikipediaapi.Wikipedia('en').

We retrieve a Wikipedia page using the page() method and specify the title of the page we want to access.

We print the title of the page and its summary.

Search for Pages:

You can also search for Wikipedia pages related to a specific topic using the search() method:

**Python**

```
 import wikipediaapi
# Create a Wikipedia API object
```

```
wiki_wiki = wikipediaapi.Wikipedia('en')
# Search for pages related to a topic
search_results = wiki_wiki.page("Python programming")
# Print the titles of the search results
print("Search results:")
for result in search_results:
    print(result)
```

Accessing Page Content:

You can access the full content of a Wikipedia page using the text attribute:

**Python**

```
 import wikipediaapi
# Create a Wikipedia API object
wiki_wiki = wikipediaapi.Wikipedia('en')
# Get the content of a Wikipedia page
page = wiki_wiki.page("Python (programming language)")
# Print the full page content
print("Full page content:")
print(page.text)
```

The "wikipedia-api" library supports multiple languages, and you can specify the desired language when creating the Wikipedia API object. Additionally, it provides various methods and attributes for accessing page links, categories, and more, llowing you to build applications that retrieve and analyze Wikipedia data programmatically.

# CHAPTER-4

## 4.1 Block Diagram

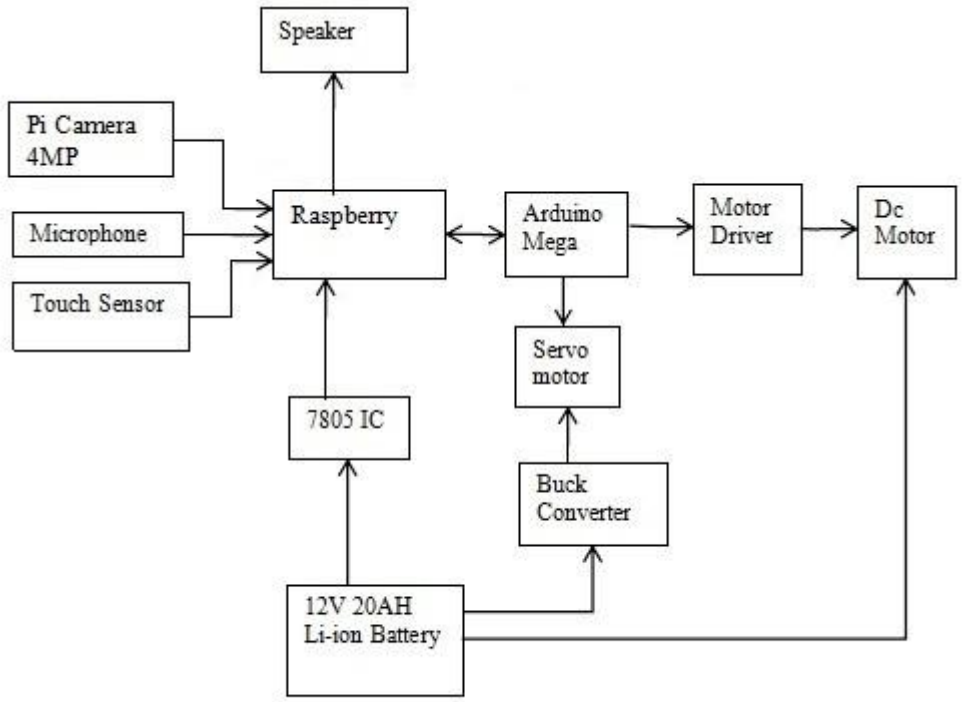



FIG 4.1 (BLOCK DAIGRAM)

## 4.2 Working:

Creating a robot with inputs from a Raspberry Pi camera, microphone, and touch sensors, and outputs to servo motors, DC motors for movement, and audio from a speaker involves integrating various hardware components and programming the Raspberry Pi to process the inputs and control the outputs.

### 4.2.1 Hardware Components:

**Raspberry Pi**: This serves as the brain of the robot, handling all input and output operations.

**Pi Camera:** The Pi Camera captures video and images for vision-based tasks.

**Microphone:** The microphone records audio input for sound-based interactions.

**Touch Sensors:** These sensors detect physical touch or contact with objects.

**Servo Motors**: Servo motors are used for precise and controlled movements, such as for camera panning or robot arm articulation.

**DC Motors:** DC motors drive the robot's movement, typically by controlling the speed and direction of the wheels.

**Speaker:** The speaker is used to produce audio output.

### 4.2.2 Software and Programming:

**Operating System:** Installed a suitable operating system (e.g., Raspberry Pi OS) on the Raspberry Pi.

**Programming**: Written Python programming language code to control the various components and handle inputs and outputs.

**Camera Input:**

Used libraries like Open CV to process video feeds from the Pi Camera.Robot can perform tasks like object detection, image recognition, or tracking based on the camera input.

**Microphone Input:**

Used a library like PyAudio to capture and process audio input.

Implement speech recognition or sound detection algorithms if needed.

**Touch Sensors:**

Connected the touch sensors to GPIO pins on the Raspberry Pi.

Written code to monitor the GPIO pins for touch events.

**Servo Motor Control:**

Connect the servo motors to GPIO pins or a PWM (Pulse Width Modulation) controller.

Use libraries like RPi. GPIO or the Adafruit PWM Servo Driver to control the servo motors. Adjust the servo angles based on camera or sensor input.

**DC Motor Control:**

Connected the DC motors to motor driver boards for movement of robot

**Speaker Output:**

Connect the speaker to the Raspberry Pi's audio output.

Used libraries like pygame or simple audio to play audio files or generate speech output.

**User Interface :**

Create a user interface for remote control or monitoring of the robot through a web interface or mobile app.

**Power Supply:** The robot has an adequate power supply to run all components.(Li-ion Battery)

## 4.3 AI integration:

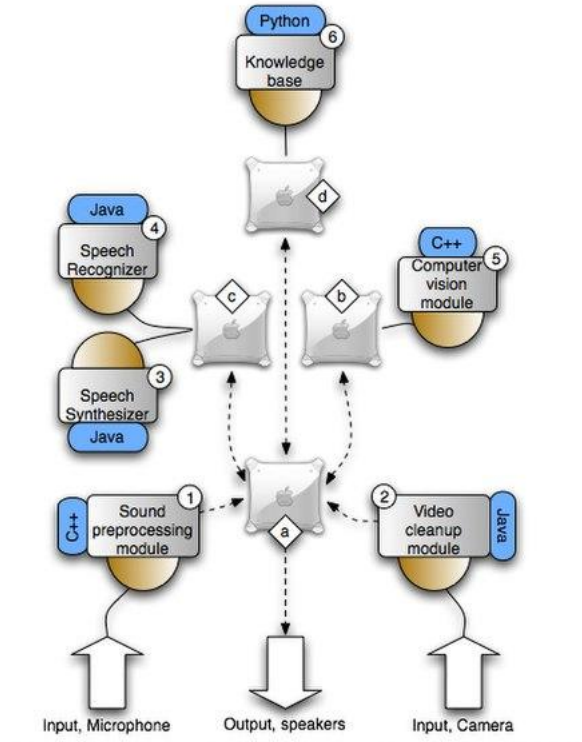
The core idea of artificial intelligence systems integration is making individual software components, such as speech synthesizers, interoperable with other components, such as common sense knowledgebases, in order to create larger, broader and more capable A.I. systems. The main methods that have been proposed for integration are message routing, or communication protocols that the software components use to communicate with each other, often through a middleware blackboard system. Most artificial intelligence systems involve some sort of integrated technologies, for example, the integration of speech synthesis technologies with that of speech recognition. However, in recent years, there has been an increasing discussion on the importance of systems integration as a field in its own right. Proponents of this approach are researchers such as Marvin Minsky, Aaron Sloman, Deb Roy, Kristinn R. Thórisson and Michael A. Arbib. A reason for the recent attention A.I. integration is attracting is that there have already been created a number of (relatively) simple A.I. systems for specific problem domains (such as computer vision, speech synthesis, etc.), and that integrating what's already available is a more logical approach to broader A.I. than building monolithic systems from scratch.

**Integration focus:**

The focus on systems' integration, especially with regard to modular approaches, derive from the fact that most intelligences of significant scales are composed of a multitude of processes and/or utilize multi-modal input and output. For example, a humanoid-type of intelligence would preferably have to be able to talk using speech synthesis, hear using speech recognition, understand using a logical (or some other undefined) mechanism, and so forth. In order to produce artificially intelligent software of broader intelligence, integration of these modalities is necessary.

**Challenges and solutions:**

An example of how multiple modules written in miscellaneous programming languages can be utilized on multiple computers in A.I. systems integration Collaboration is an integral part of software development as evidenced by the size of software companies and the size of their software departments. Among the tools to ease software collaboration are various procedures and standards that developers can follow to ensure quality, reliability and that their software is compatible with software created by others (such as W3C standards for webpage development). However, collaboration in fields of A.I. has been lacking, for the most part not seen outside the respected schools, departments or research institutes (and sometimes not within them either). This presents practitioners of A.I. systems integration with a substantial problem and often causes A.I. researchers to have to 're-invent the wheel' each time they want a specific functionality to work with their software. Even more damaging is the "not invented here" syndrome, which manifests itself in a strong reluctance of A.I. researchers to build on the work of others.




**An example of A.I. systems integration using message protocols and blackboard systems such as OpenAIR and Psyclone**

The outcome of this in A.I. is a large set of "solution islands": A.I. research has produced numerous isolated software components and mechanisms that deal with various parts of intelligence separately. To take some examples:

Speech synthesis
FreeTTS from CMU
Speech recognition
Sphinx from CMU
Logical reasoning
OpenCyc from Cycorp
Open Mind Common Sense Net from MIT

With the increased popularity of the free software movement, a lot of the software being created, including A.I. systems, is available for public exploit. The next natural step is to merge these individual software components into coherent, intelligent systems of a broader nature. As a multitude of components (that often serve the same purpose) have already been created by the community, the most accessible way of integration is giving each of these components an easy way to communicate with each other. By doing so, each component by itself becomes a module, which can then be tried in various settings and configurations of larger architectures. Some challenging and limitations of using A.I. software is the uncontrolled fatal errors. For example, serious and fatal errors have been discovered in very precise fields such as human oncology, as in an article published in the journal Oral Oncology Reports entitled “When AI goes wrong: Fatal errors in oncological 20 research reviewing

assistance".[1] The article pointed out a grave error in artificial intelligence based on GBT in the field of biophysics.

Many online communities for A.I. developers exist where tutorials, examples, and forums aim at helping both beginners and experts build intelligent systems. However, few communities have succeeded in making a certain standard, or a code of conduct popular to allow the large collection of miscellaneous systems to be integrated with any ease.

**Methodologies:**

Constructionist design methodology

The constructionist design methodology (CDM, or 'Constructionist A.I.') is a formal methodology proposed in 2004, for use in the development of cognitive robotics, communicative humanoids and broad AI systems. The creation of such systems requires the integration of a large number of functionalities that must be carefully coordinated to achieve coherent system behavior. CDM is based on iterative design steps that lead to the creation of a network of named interacting modules, communicating via explicitly typed streams and discrete messages. The OpenAIR message protocol (see below) was inspired by the CDM and has frequently been used to aid in the development of intelligent systems using CDM.

Integrating AI on a Raspberry Pi can be done using various frameworks like TensorFlow Lite, OpenCV, or even custom Python scripts. Here's a basic example using TensorFlow Lite for image classification:

1. Install TensorFlow Lite:

   pip install tflite-runtime

2. Get a TensorFlow Lite Model:

You can either train your own model or download a pre-trained one. For example, you can get a pre-trained image classification model from TensorFlow's model zoo.

**3.** Code:

```
import tensorflow as tf
import numpy as np
from PIL import Image

# Load the TFLite model and allocate tensors.
interpreter = tf.lite.Interpreter(model_path="model.tflite")
interpreter.allocate_tensors()

# Get input and output tensors.
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

# Load an image.
img = Image.open('image.jpg').resize((input_details[0]['shape'][1],
input_details[0]['shape'][2]))
input_data = np.expand_dims(img, axis=0)

# Normalize input data (if needed).
input_mean = 127.5
input_std = 127.5
```

```
input_data = (np.float32(input_data) - input_mean) / input_std

# Set the input tensor.
interpreter.set_tensor(input_details[0]['index'], input_data)

# Run inference.
interpreter.invoke()

# Get the output tensor.
output_data = interpreter.get_tensor(output_details[0]['index'])

# Process the output (e.g., print class labels).
print(output_data)
```

## 4.4 LlamaIndex:

LlamaIndex is a data framework for LLM-based applications which benefit from context augmentation. Such LLM systems have been termed as RAG systems, standing for "Retrieval-Augemented Generation". LlamaIndex provides the essential abstractions to more easily ingest, structure, and access private or domain-specific data in order to inject these safely and reliably into LLMs for more accurate text generation. It's available in Python (these docs) and Typescript.
LLMs offer a natural language interface between humans and data. Widely available models come pre-trained on huge amounts of publicly available data like Wikipedia, mailing lists, textbooks, source code and more.

However, while LLMs are trained on a great deal of data, they are not trained on your data, which may be private or specific to the problem you're trying to solve. It's behind APIs, in SQL databases, or trapped in PDFs and slide decks.

You may choose to fine-tune a LLM with your data, but:

Training a LLM is expensive.
Due to the cost to train, it's hard to update a LLM with latest information.
Observability is lacking. When you ask a LLM a question, it's not obvious how the LLM arrived at its answer.
Instead of fine-tuning, one can a context augmentation pattern called Retrieval-Augmented Generation (RAG) to obtain more accurate text generation relevant to your specific data. RAG involves the following high level steps:

Retrieve information from your data sources first,
Add it to your question as context, and
Ask the LLM to answer based on the enriched prompt.
In doing so, RAG overcomes all three weaknesses of the fine-tuning approach:

There's no training involved, so it's cheap.
Data is fetched only when you ask for them, so it's always up to date.
LlamaIndex can show you the retrieved documents, so it's more trustworthy.
Why Context Augmentation?#
LLMs offer a natural language interface between humans and data. Widely available models come pre-trained on huge amounts of publicly available data like Wikipedia, mailing lists, textbooks, source code and more.

However, while LLMs are trained on a great deal of data, they are not trained on your data, which may be private or specific to the problem you're trying to solve. It's behind APIs, in SQL databases, or trapped in PDFs and slide decks.

You may choose to fine-tune a LLM with your data, but:

Training a LLM is expensive.
Due to the cost to train, it's hard to update a LLM with latest information.
Observability is lacking. When you ask a LLM a question, it's not obvious how the LLM arrived at its answer.

Instead of fine-tuning, one can a context augmentation pattern called Retrieval-Augmented Generation (RAG) to obtain more accurate text generation relevant to your specific data. RAG involves the following high level steps:

Retrieve information from your data sources first,
Add it to your question as context, and
Ask the LLM to answer based on the enriched prompt.
In doing so, RAG overcomes all three weaknesses of the fine-tuning approach:

There's no training involved, so it's cheap.
Data is fetched only when you ask for them, so it's always up to date.
LlamaIndex can show you the retrieved documents, so it's more trustworthy.
Why LlamaIndex for Context Augmentation?#
Firstly, LlamaIndex imposes no restriction on how you use LLMs. You can still use LLMs as auto-complete, chatbots, semi-autonomous agents, and more (see Use Cases on the left). It only makes LLMs more relevant to you.

LlamaIndex provides the following tools to help you quickly standup production-ready RAG systems:

Data connectors ingest your existing data from their native source and format. These could be APIs, PDFs, SQL, and (much) more.
Data indexes structure your data in intermediate representations that are easy and performant for LLMs to consume.
Engines provide natural language access to your data. For example:

Query engines are powerful retrieval interfaces for knowledge-augmented output.

Chat engines are conversational interfaces for multi-message, "back and forth" interactions with your data.

Data agents are LLM-powered knowledge workers augmented by tools, from simple helper functions to API integrations and more.

Application integrations tie LlamaIndex back into the rest of your ecosystem. This could be LangChain, Flask, Docker, ChatGPT, or… anything else!

## 4.4 Semantic Search :

**Design Implementation**
**Part 1 - Overall Product Specifications**
**Problem Statement** - Our goal here is to build a simple RAG application on the College details document.

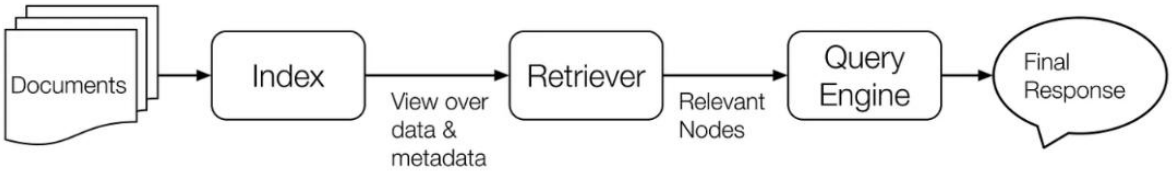
**Solution Strategy** - Build a POC which should solve the following requirements:

Users would get responses from the Insurance Document.
If they want to refer to the original page from which the bot is responding, the bot should provide a citation as well.
**Goal** - Solving the above two requirements well in the POC would ensure that the accuracy of the overall model is good and therefore further improvisations and customizations make sense.

**Data Used** – Document prepared with details of college

Tools used - LlamaIndex has been used due to its powerful query engine, fast data processing using data loaders and directory readers as well as easier and faster implementation using fewer lines of code.
**Part 2 - Solution POC**

In this section, a POC is built that was proposed in the previous step




**Step 3 - Data Loading:**
SimpleDirectoryReader was used for data loading as multiple files were stored in google drive in a folder.
**Step 4 - Building the query engine:**
Llama_Index is used to build the query engine.
**Step 5 - Checking responses and response parameters**
**Step 6 - Creating a response Pipeline**
**Step 7 - Build a Testing Pipeline**
Here we feed a series of questions to the Q/A bot and store the responses along with the feedback on whether it's accurate or not from the user

**Challenges**
It is observed that the responses were not always correct. Sometimes there were no responses. Hence prompting was done to improve the accuracy of the responses.

**Lessons Learned**
Semantic search using embeddings, and Retrieval Augmented Generation using Llama_Index is learnt.

# CHAPTER-5

## 5.1 Conclusion

In conclusion, this project involves the development of a versatile robot that incorporates inputs from a Raspberry Pi camera, microphone, and touch sensors, and delivers outputs through servo motors, DC motors for movement, and audio via a speaker. This project is a comprehensive exploration of robotics and programming on the Raspberry Pi platform, offering opportunities for innovation and experimentation in various domains. It can serve as a valuable educational experience and a foundation for more advanced robotic applications. Successful completion of this project would result in a versatile robot capable of performing tasks that involve vision, audio, and physical interaction while providing opportunities for further customization and expansion.

## 5.2 Future scope

**Autonomous Navigation and Mapping:**

Implement algorithms for autonomous navigation and mapping using the camera and sensors. Integrate technologies like SLAM (Simultaneous Localization and Mapping) for indoor mapping and localization.

**AI and Machine Learning Integration:**

Enhance the robot's capabilities by integrating AI and machine learning models for tasks like object recognition, voice command understanding, and decision-making.

Implement deep learning models for more advanced image and speech processing.

**Voice Assistant and Natural Language Processing:**

Develop the robot into a voice-controlled assistant capable of natural language understanding and responses.Expands its functionality to perform tasks based on voice commands and queries.

**Wireless Control and Remote Monitoring:**

Enable remote control and monitoring of the robot through wireless communication protocols like Wi-Fi or Bluetooth.

Create a user-friendly mobile app or web interface for remote interaction.

**Sensor Expansion:**

Add more sensors to the robot for additional capabilities, such as infrared sensors for obstacle avoidance, environmental sensors for data collection, or GPS for outdoor navigation.

**Robot Arm and Manipulation:**

Incorporate a robot arm or gripper for tasks involving manipulation, such as picking up objects or interacting with the environment.

**Human-Robot Interaction (HRI):**

Explore human-robot interaction concepts, such as gesture recognition, facial recognition, and emotional analysis, to improve the robot's ability to interact with users.

**Multi-Robot Systems:**

Create a swarm of robots that can communicate and collaborate on tasks, such as search and rescue missions or distributed environmental monitoring.

**Security and Surveillance:**

Utilize the robot for security and surveillance applications by equipping it with additional cameras, motion detectors, and alarms.

**Education and Research:**

Adapt the project for educational purposes, introducing robotics and programming concepts to students.Use the robot as a research platform for exploring topics like human-robot interaction, computer vision, and artificial intelligence.

**Customization for Specific Industries:**

Tailor the robot's capabilities and sensors to meet specific industry needs, such as healthcare (telemedicine), agriculture (crop monitoring), or logistics (inventory management).

**Open-Source Contribution:**

Contribute to open-source robotics projects and communities by sharing your code and experiences, fostering collaboration and innovation.

The future scope of this project is vast and can be customized to meet various needs and interests. As technology advances and new hardware and software solutions become available, there will be even more opportunities to enhance and expand the capabilities of your versatile robot.

**Secretary and Correspondent**

Dr. P. Rajeshwar Reddy is a Member of Legislative Council and a Government Whip of Telangana State. His passion and zeal for education is instrumental in creating Anurag Institutions. He is one of the charismatic leaders in education providing a learning environment for an intellectual, personal and societal transformation.

He obtained his Ph.D in Physics from Osmania University. He also earned the most prestigious award “Dr. Tamahanker Memorial Prize” for the Best Research Paper in the year 1996 by Magnetic Society of India.

**Joint Secretary**

Mrs. S. Neelima did her B. Tech in Electrical and Electronics Engineering from JNTU, Hyderabad and M. Tech, Power Systems from Osmania University, Hyderabad.

She served as an Engineer in AP Transco department for over two decades. Mrs. Neelima is presently Joint Secretary and takes care of the management of the institution.

**Member**

Prof.M. Govind Ram Reddy obtained his Doctorate in Chemistry from Osmania University and was associated with the University for more than three decades in various positions like Faculty, Head, Department of Chemistry, and also as Vice Principal and Principal of University College of Science.

More than 60 Papers were published in national and international journals and he guided 16 students to achieve their doctorate degrees. He is member of the VJES

**Principal**

Professor Eadala Saibaba Reddy, ME. (Hons) Roorkee, PhD (Nottingham, UK), Post Doc. (Halifax, Canada), Post Doc (Birmingham, UK) an eminent Civil Engineer and a Professor of Civil Engineering JNTUHCEH. Dr. Reddy served as Professor in the Civil Department, functioned in many administrative capacities, as Head of the Department, Vice-Principal, Director Academic and Planning, Director Academic Audit Cell, Officer on Special Duty to Vice- Chancellor, Registrar, Rector (Pro-Vice

Chancellor) of JNT University Hyderabad. He has been a member in a number of expert selection committees constituted by Govt. of Andhra Pradesh, UGC, AICTE, Union Public Service Commission, A.P. Public Service Commission, Member of Search Committee for selection of Vice-Chancellor, Member of Board of Governors of Veer Surendra Sai Institute of Medical Science and Research. He is also on many other Academic/ Administrative/ Governing Bodies of various Institutions and Universities. He was Convener for EAMCET for two consecutive years 2007 and 2008.

Dr Reddy a recipient of several Scholarships and Fellowships, including: an UGC Scholarship, Commonwealth Scholarship (UK), Commonwealth Fellowship (U.K.), & Post-Doctoral Fellowship (Canada). A recipient of a number of National and International Awards including: American Society for Testing and Materials (ASTM) Award (USA), for outstanding Research paper for the year 1998, Shamshare Prakash Research Award by University of Roorkee (presently IIT Roorkee) Year 2000, Engineer of the Year Award from Govt. of A.P. and Institution of Engineers (India) for the Year 2009. Best Teacher Award from Govt. of A.P. for the year 2012.

Professor Reddy as a visiting faculty has visited over 20 Foreign Universities, including University of Birmingham, UK, University of New Orleans USA, Kansas State University (USA) and University of Illinois at Chicago (USA). He delivered invited Lectures and Chaired various National and International conferences all over the world. He is a member of over 12 National and International Professional bodies, guided 7 PhD Scholars and several M. Tech. projects, presently Guiding 3 PhD Scholars. He has published over 150 Research papers in National and International journals and Authored four Text Books in Civil and Environmental Engineering.

**Dean**

Dr. Padmaja obtained her B.Tech. in Chemical Engineering and M.Tech. in Biochemical Engineering from A.U. College of Engineering Andhra University, Visakhapatnam. Subsequently, she received her Ph. D. in Chemical Engineering from College of Technology, Osmania University and Post Graduate Diploma in Patent Law from Nalsar University, Hyderabad. Dr. Padmaja has guided several B.Tech / M.Tech students and published numerous papers in reputed journals/conference proceedings.

Dr. Padmaja is currently working as Dean, Accreditation & Rankings at Vidya Jyothi Institute of Technology, Hyderabad. She has nearly 20 years of Experience in Industry and Academia in the areas of teaching, research and development and consulting. She has been working with Anurag Group of Institutions for the past seven years in various capacities. Prior to joining academics, she worked at Uni-Sankyo Ltd., and Jupiter Bio-sciences Ltd., Hyderabad managing large R&D projects of highly interdisciplinary nature involving Chemical Engineering and Biotechnology. She was responsible of launching of several new products like enzymes, amino acids, nutracueticals and mucopolysaccharides beginning from basic research in the lab to taking all the way to bulk production. She has taken the initiative in setting up and managing several Labs and facilities for research and development, quality control

and diagnostics. She has profound knowledge of GLP/GMP processes and ISO standards.

**Sr. Administrative Officer**

Mr. Venkatachalam has done his M. Sc. (Physics) and M. Phil. He has over 17 years of experience in academics and worked in various senior positions in the teaching and administrative roles in professional colleges. Through his student friendly approach and lively attitude he adds a lot of value to the administrative system.

**Academic Coordinator**

Mrs. Srilatha is awarded two Gold Medals in Academics by Osmania University. She has 25 years of teaching experience and 12 years as a principal. She is an able administrator and has proficiency in academics and administrative activities and is known for commitment. She has perseverance to regularly interact and motivate the students to achieve their goals.

| Name | Qualification | Designation |
|---|---|---|
| Prof B S Murthy | Director IIIT, Hyderabad | Chairman |
| Dr. P. Rajeshwar Reddy | M. Sc, Ph. D., MLC, Govt.Secretary & Correspondent of VJES | Member |
| Mr Krishna Palla | B.Tech, MPS (USA) | Member |
| Mrs. S. Neelima | M.Tech., Joint Secretary of VJES | Member |
| Dr. L Ramakrishna Reddy | President Auropro Soft | Member |
| Dr. E. Sai Baba Reddy | Principal, VJIT | Member Secretary |
| Prof Sreeram Venkatesh | Dept of MECH,Principal, University College of Engineering , OU | Member |

| Name | Qualification | Designation |
|---|---|---|
| Dr M T Naik | Professor of ME, JNTUH CEH | Member |
| Mr R Venkata Chalam | Sr. Adminstrative Officer | Member |
| Dr. A. Padmaja | M.Tech, Ph.D, Dean Accreditations & RankingsVJIT | Member |
| Dr. B. Vijaya Kumar | M.Tech, Ph.D Professor in CSE | Member |
| Mrs G Srilatha | Academic Coordinator | Member |
| Dr. Balaji Utla | Registrar,Anurag University | Member |

**Laboratories**

The VJIT's laboratories are far more advanced and ever accessible to students .This encourages students to spend more time on practicals. They are guided by experienced teaching and non-teaching staff. Students get easy access to the labs and are free to work on any projects they like in conjunction with the faculty.
We have 11 high-end computer labs with latest configuration and required software with ample UPS support.

**Conference Halls**

VJIT has three spacious conference halls with the state-of-the-art audio visual equipment. The First Hall has a seating capacity of 500, the second & third with the capacity to accommodate 200. The spaces in the conference halls have been designed with good acoustic systems and ambiance. Conferences, Workshops, Seminars, Tech Fests, Faculty Development Programs and Student Training & Development Programs are organized regularly in these facilities and they have been well appreciated by the visiting dignitaries from premier academic institutions and corporate organizations.

## Library & Information Centre

The Vidya Jyothi Institute of Technology (VJIT) Central Library is the soul of the Institution. The Central Library supports the Institution in realizing its primary goal of imparting the quality Technical Education and to promote the research activities. The Central Library acquires, process, preserves and disseminate the information to the user Community for attaining the academic and research goals. The Objective of the Central Library is to empower knowledge. The Central Library provides information services and access to bibliographic as well as full-text digital and printed resources to support the scholarly and informational needs of the Institute community.

Housed in an independent and spacious building, The Central Library has an extensive collection of Books, Scientific and Technical Journals accompanied by a vast collection of Electronic Resources. The central Library comprises of Reference Section, Circulation Section, Journals Section, Newspaper Section, Reprography Section, Discussion Room, SC/ST Book Bank Section and Digital Library. The Library has automated all its activities by Koha Integrated Library Software to provide better, effective and immediate services to user fraternity. Apart from the Central Library, departmental libraries have set up to cater to the needs of the students, faculty and Research Scholars at the departmental level.

Well-equipped with modern facilities with WiFi enabled contains a myriad of resources including CD- ROMs, online databases, audio-video cassettes, books, e-journals, theses, reports, monographs, full-text and bibliographical databases among others.

Sports

Sports play an important role in shaping up the personality and fitness of a person and to give a truly global experience to all the students of VJIT. Sports environments matching global standards are provided at the college.
VJIT College is founded in 1999, since the inception college has been actively involved in sports and has been undisputed champions in Games like Cricket, Volleyball, Kabaddi, Badminton, Basketball, Wrestling, Boxing, Chess and Table Tennis.
College has a very good infrastructure of volleyball courts, basketball court, kabaddi courts, cricket ground and nets, and Table Tennis Hall which accommodates 06 TT tables. Also has a Gym with latest Gym equipment's.
Hall which accommodates a space of more than 10 Carom Boards, and 10 Chess Boards. Massive hall which is dedicated to the sport Wrestling and Boxing where champions are made.
College has a outdoor games facility with games like, Volley ball (2 courts) ,

Throwball (1 court) , Basketball (1 court) Kabaddi 2 (Courts) Kho kho field, Football field, Cricket Ground.

**Transport**

The college has its own fleet of buses operating from many points in the city of Hyderabad and the suburbs. The starting points and routes of the College Buses: College Bus Timings & Route Details 2024. For more details contact Transport In-Charge Mr. K. Pradeep Reddy, 9866999116.

## EEE Team

## Head of the Department

Dr. A. Srujana completed her Bachelor's degree from Kakatiya Institute of Technology & Science, Warangal, and M. Tech in Power Electronics from JNTU, Hyderabad. She was awarded Ph.D by JNTUH,Hyderabad for her research in the area of HVDC.She has worked at various capacities in prominent Engineering colleges as Principal, Professor and HOD. She has guided fifteen M. Tech projects & 2 Ph. D students. Her expertise in research is witnessed with her publication of 32 research papers in core technical areas in indexed journals and also in both National and International peer reviewed conferences. She has about 23 years of experience in Academics. Her areas of interests are power Electronics, FACTS, HVDC and POWER SYSTEMS.

Faculty Members

| Name | Designation |
|---|---|
| Dr Vuppalanchu Srujana | HOD |
| Mr Daravastu Srinivas | Associate Professor |
| Mrs K Swapna | Assistant Professor |
| Mr Tummeti Parameshwar | Associate Professor |
| Mr Hussain Shaik | Associate Professor |
| Dr Siva Prasad Syamala | Professor |
| Mr Kammari Sathish Kumar | Associate Professor |
| Mr M Vijay Kumar | Assistant Professor |
| Mr Paidipamula Nageswara Rao | Associate Professor |
| Mrs S Chaitanya | Assistant Professor |
| Dr Dumpala Bala Gangi Reddy | Professor |

| | |
|---|---|
| Mrs Vemuri Vijaya LakshmI | Associate Professor |
| Dr C N Ravi | Professor |
| Mr P Naga Muneendra | Assistant Professor |
| Mr Raju Lakkabattula | Assistant Professor |
| Mr Vikram Chandha | Assistant Professor |
| Mrs P Vaishnavi devi | Assistant Professor |
| Mr Sudhakar Reddy Bhumireddy | Assistant Professor |
| Mr Karnam Rajeev | Assistant Professor |
| Mr Shamanety Suresh | Assistant Professor |
| Ms Abbe Doddi Bolla Bhavana Reddy | Assistant Professor |
| Mrs K Haritha | Assistant Professor |
| Mr Bhaskarla Rajesh | Assistant Professor |
| Ms R Keerthi | Assistant Professor |
| Ms M Jhansi Lakshmi | Assistant Professor |
| Mrs Attaluri Srilatha | Assistant Professor |
| Mr Audirala Mohan Das | Assistant Professor |
| Mr Shaik Mohammod Zaffarullah | Associate Professor |
| Dr S Rajesh Kumar | Associate Professor |
| Mr NLV Prasada Rao | Associate Professor |
| Mr M Soujanya | Assistant Professor |
| Mr Bodhanam V Siva Subrahmanyam | Assistant Professor |
| Mrs SH B Ireena | Assistant Professor |
| Dr Mathi Dileep Krishna | Assistant Professor |
| Mr P Satheesh | Assistant Professor |
| Mrs Pandikuntla Murali | Assistant Professor |
| Mrs Vasanthapuram Anuradha | Assistant Professor |
| Mr Pasumarthi Hemanth Kumar | Assistant Professor |
| Mr D Srikanth | Assistant Professor |
| Dr Devagowda Siddegowda | Associate Professor |
| Ms Machupalli Anitha | Assistant Professor |

Block is located near  Main Entrance.Mechanical department is in A-Block.

Block is located beside A-Block.Workshop is located in B-Block.

Block is in the middle of S-Block and N-Block.CSE department is located in C-Block.

Block is located in front of E-Block.Civil department and MBA department are located in D-Block.

Block is located at back entrance.ECE department and 1st Year students are located in E-Block.

Block is behind Volleyball court.EEE department is located in N-Block.

Block is beside cafeteria.AI department is located in S-Block.

**Head of the Department CSE:**

Dr. D.Aruna Kumari, has completed B.Tech in CSE from Shri Vishnu Engineering college , Bhimavaram in the year 2005,  M.Tech in CSE from Koneru Lakshmaiah college of Engineering and Ph.D in CSE  in the year 2014  from K L University. She received a grant of Rs. 1, 18,61,430/- from DST for  Developing 4 villages to enhance their Agricultural Sustainability trough IoT for Social and Economic Empowerment of Scheduled Caste Communities (in Moinabad Mandal, Rangareddy District, Telangana State)" in the year March,2023.

**Head of the IT Department :**

Srinivasulu completed his Bachelor's degree (Computer Science & Engineering) from Karnataka University, Master's degree from Anna University, Chennai. He is currently pursuing research in the area of Cloud Computing. He has 21 years of Academic, Research and Industry experience. Working as Professor & Head in Department of Information Technology since December 2011, he worked as Senior Assistant Professor in Department of Computer Science & Engineering, PES Institute of Technology, Bangalore and Lecturer in Department of CSE, SRM Engineering College, Chennai. He has around 20 journal articles, national and international conference publications to his credit. He is the member of ACM, ISTE and CSI.

**Head of the AI Department:**

Dr. A. Obulesh working as Assoc. Prof. at Vidya Jyothi Institute of Technology, in the Dept. of Artificial Intelligence has an incredible experience of 14 years in academics under various capacities. He received his Ph.D. in Computer Science & Engineering from Jawaharlal Nehru Technological University, Kakinada (JNTUK), India. The M.Tech. in CSE from Rajeev Gandhi Memorial College of Engineering and Technology(RGMCET), Nandyal which is affiliated to JNT University, Hyderabad, in 2006.

**Head of the ECE Department:**

Dr.M. Rajendra Prasad obtained his B. Tech (ECE) from SK University, M.E (Digital System Design) from Osmania University and was awarded Ph.D. in Internet of Things from Osmania University. A versatile academician with extensive experience of 23 years in teaching and research, he has been serving Vidya Jyothi Institute of Technology for the last 18 years.

**Head of the CIVIL Department:**

Dr. Pallavi Badry completed Bachelor degree in Civil Engineering from Visvesvaraya National Institute of Technology, Nagpur and Ph. D in Geotechnical Earthquake Engineering from IIIT Hyderabad. She has more than 15 years of experience in teaching, research and industry. Her expertise in research benchmarked with around 50 technical papers in indexed journals and in National and International peer-reviewed conferences. She guided 50+ BTech students and 15 Mtech students and 1 PhD student. She has 3 book chapters, 1 book and 1 International patent on her credits.

**Head of the MECHANICAL Department:**

Dr. G Sreeram Reddy has done his B.Tech (Mech.) from Osmania University, Hyderabad, M. Tech. (Energy Systems) from JNTUH and was awarded a Ph.D. in Mechanical Engineering from JNTUH.His career in academics spans more than 19 years. He has worked with various corporate as Design Engineer, before taking charge as the Head of the Department of Mechanical Engineering. He also contributed his services as the Placement Officer and Student Coordinator at VJIT. He has to his credit 25 peer-reviewed technical papers published in National and International journals. He has applied 4 Patents with Indian Patent Office on numerous technical innovative concepts.