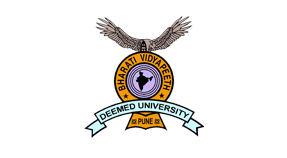
**BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY),**

**COLLEGE OF ENGINEERING,**

**PUNE-411043**

**Department of Information Technology**

**A**

**Project Based Learning**

**Artificial Intelligence and Machine Learning**

**On**

**“ Pneumonia Detection”**

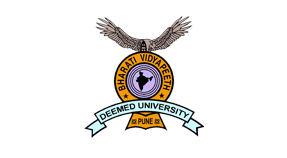
**Submitted By**

|  |  |  |
| --- | --- | --- |
| Roll No. | PRN No. | Name |
| 26. | 2214110259 | Ajay Patidar |
| 28. | 2214110261 | Hars Rai |
| 60. | 2214110409 | Ayush Kumar |

**Under the Guidance of**

**Prof. Milind Gayakwad**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

****

**Certificate**

This is to certify that the work under Project Based Learning (PBL) for the topic **“Pneumonia Detection”** is carried out by “**Ajay Patidar, Hars Rai & Ayush Kumar”** under the guidance of **Prof. Milind Gayakwad** in partial fulfillment of the requirement for the degree of “**Bachelor of Technology in Information Technology Semester-V**” of **“Bharati Vidyapeeth (Deemed to be University), Pune”** during the academic year **2023-2024**.

**Date**:- **Prof. Prakash Devale**

**GUIDE**

**ACKNOWLEDGEMENT**

We would like to express deepest appreciation towards Dr Vidula Sohoni, Principal, Bharati Vidyapeeth (Deemed to Be University) College of Engineering, Pune, India and Dr.Sandeep Vanjale Head of the Department (IT), who invaluable supported us in completing this project.

We are profoundly grateful to Prof. Prakash Devale, Project guide for her expert guidance and continuous encouragement throughout to see that this project rights, its target since its commencement to its completion.

At last we express our sincere heartfelt gratitude to all the staff members of IT Department who helped us directly or indirectly during this course of work.

**Ajay Patidar**

**Hars Rai**

**Ayush Kumar**

**B.Tech. (IT) Semester – V**

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Title | Page No. |
| 1. | Introduction | 4 |
| 2. | Circuit Diagram | 6 |
| 3. | Components & Software Used | 7 |
| 4. | New Features | 8 |
| 5. | Applications | 9 |
| 6. | Execution Steps | 10 |
| 7. | Conclusion | 13 |

# **INTRODUCTION**

In today's digital age, voice assistants have become an integral part of our lives, simplifying tasks and providing convenience through natural language interactions. We'll explore how Raspberry Pi, a low-cost, credit-card-sized computer, can serve as a powerful platform for creating custom voice assistants. We'll delve into the significance of IT infrastructure management in this process and discuss the key components and requirements involved. We'll examine the potential of Raspberry Pi in voice assistant development, highlight the role of IT infrastructure management, and provide insights into the essential components needed for building our own AI-powered assistant.

**Introduction to Voice Assistants:**

A voice assistant is a digital assistant that uses voice recognition, language processing algorithms, and voice synthesis to listen to specific voice commands and return relevant information or perform specific functions as requested by the user.

* Voice assistants have become an integral part of our daily lives, offering convenience and efficiency in performing various tasks.
* A voice assistant is a digital assistant that uses voice recognition, language processing algorithms, and voice synthesis to listen to specific voice commands and return relevant information or perform specific functions as requested by the user.
* They utilize natural language processing (NLP) and artificial intelligence (AI) algorithms to understand spoken commands and execute corresponding actions.

Today, voice assistants are integrated into many of the devices we use on a daily basis, such as cell phones, computers, and smart speakers. Because of their wide array of integrations, There are several voice assistants who offer a very specific feature set, while some choose to be open ended to help with almost any situation at hand.

**Raspberry Pi:**

The Raspberry Pi Foundation, a UK charity that aims to educate people about computing, developed Raspberry Pi in association with Broadcom, and launched it in 2012. The Raspberry Pi has wireless internet connectivity, HDMI ports for monitors, USB ports for accessories, and ample processing power and RAM. It can work with any input and output hardware device, such as a monitor, television, mouse, or keyboard.

* Raspberry Pi is a small, affordable, single-board computer developed by the Raspberry Pi Foundation.
* It provides a versatile platform for DIY projects, education, and prototyping due to its compact size and low cost.
* Raspberry Pi supports various programming languages and interfaces, making it ideal for building custom solutions.
* The Raspberry Pi has a CPU, memory, LAN, USB, and micro HDMI ports, and a slot for a micro SD card. The original Pi had a single-core 700MHz CPU and just 256MB RAM, while the latest model has a quad-core CPU clocking in at over 1.5GHz
* Raspberry Pi has many practical uses, including media streaming and extending the range of your Wi-Fi network. Some say that the Raspberry Pi has at least the same technical capabilities as a Mac mini, though Apple's M2 still has a lot more performance to spare.

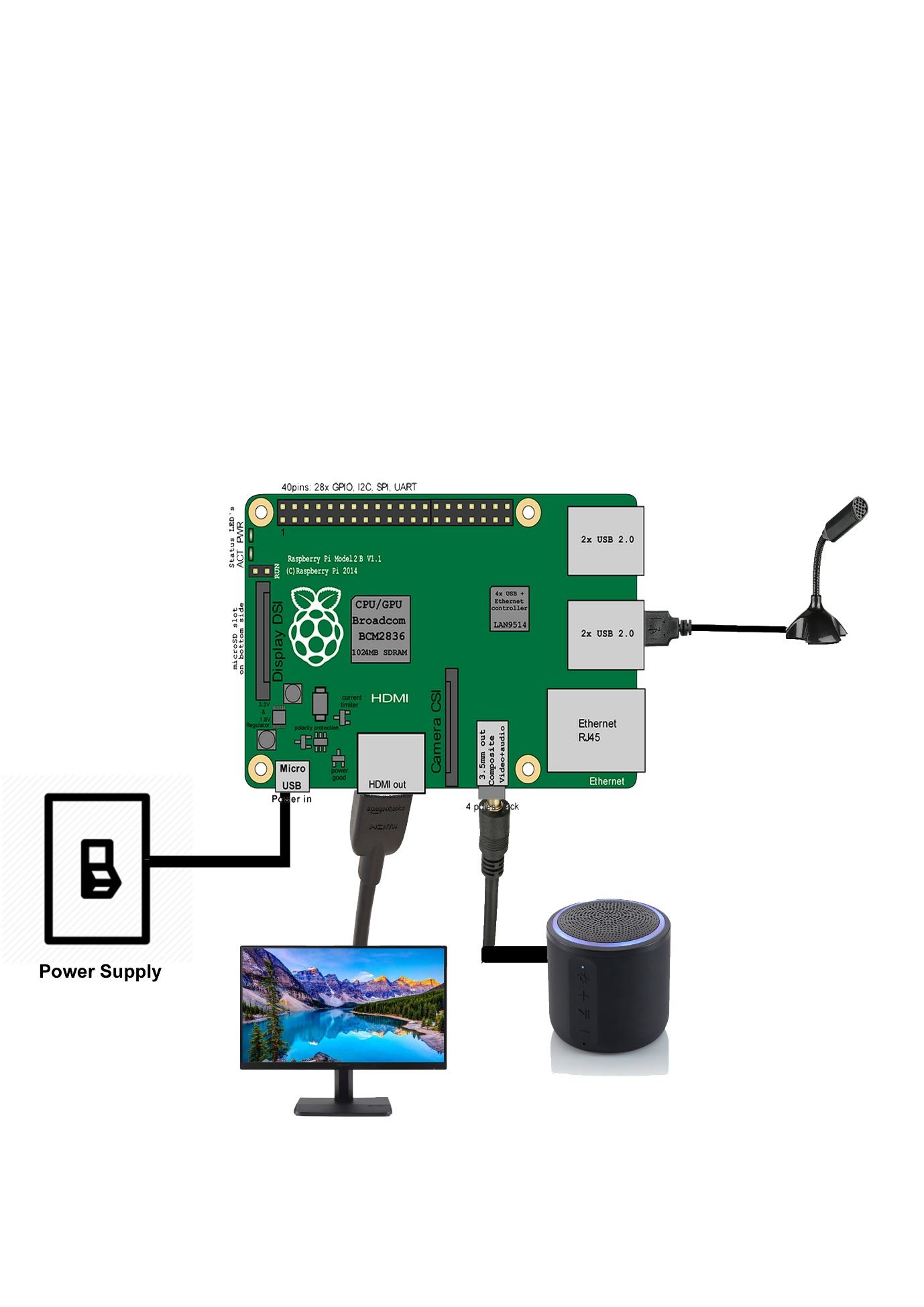
**Project Overview:**

* The aim of this project is to create a voice assistant using Raspberry Pi, allowing users to interact with the device through spoken commands.
* By integrating hardware components, software libraries, and cloud services, we can develop a functional voice assistant capable of performing a range of tasks.

**Key Features:**

* Speech Recognition: Implementing speech recognition algorithms to interpret spoken commands accurately.
* Natural Language Understanding (NLU): Utilizing NLP techniques to extract meaning from user queries and generate appropriate responses.
* Task Automation: Enabling the voice assistant to perform tasks such as setting reminders, playing music, retrieving information from the web, controlling smart home devices, and more.
* Customization: Providing users with the ability to customize the voice assistant's functionality and add new capabilities according to their preferences.

# **CIRCUIT DIAGRAM**



# **Component and Software Used:**

**Components:**

**Raspberry Pi:** The central component of the project, Raspberry Pi serves as the main computing platform. It provides the processing power and interfaces necessary to run the voice assistant software and interact with external hardware.

**Microphone:** A microphone is required to capture audio input from the user. It can be either a standalone USB microphone or a microphone module connected to the Raspberry Pi's audio input port.

**Speaker:** A speaker is used to output audio responses generated by the voice assistant. It can be connected to the Raspberry Pi's audio output port.

**Internet Connection:** An internet connection is essential for accessing online resources and services required for tasks such as speech recognition, natural language processing, and retrieving information from the web.

**Software:**

**Operating System (OS):** Raspberry Pi typically runs on a Linux-based operating system such as Raspbian or Ubuntu. These OS distributions provide the necessary tools and libraries for software development and device interaction.

**Python Programming Language:** Python is commonly used for programming Raspberry Pi projects due to its simplicity and extensive libraries. The voice assistant software is likely to be written in Python, utilizing libraries for speech recognition, natural language processing, and interaction with external hardware.

**Speech Recognition Library:** A speech recognition library is used to convert audio input from the microphone into text that can be processed by the voice assistant. Popular libraries include Google's Speech Recognition API, CMU Sphinx, and PocketSphinx.

**Natural Language Processing (NLP) Library:** NLP libraries are used to analyze and understand the meaning of user queries expressed in natural language. These libraries enable the voice assistant to extract relevant information and generate appropriate responses. Examples of NLP libraries include NLTK (Natural Language Toolkit), spaCy, and TensorFlow.

**Text-to-Speech (TTS) Library:** A TTS library is used to convert text responses generated by the voice assistant into audible speech output. This allows the assistant to communicate with users through spoken dialogue. Common TTS libraries include pyttsx3, gTTS (Google Text-to-Speech), and Amazon Polly.

**APIs and Services**: The voice assistant may interact with various online APIs and services to perform tasks such as retrieving information from the web, accessing weather forecasts, controlling smart home devices, and more. Examples include Google Assistant API, OpenWeatherMap API, and Home Assistant..

**New Features:**

1. **Time Table Management:**The assistant will allow users to schedule and manage their daily or weekly timetable by adding, updating, or removing entries. Users can ask the assistant about upcoming events or get reminders of scheduled tasks.
2. **Alarm Functionality:**The voice assistant will include the ability to set alarms for specific times. Users can interact using commands like “Set an alarm for 7 AM tomorrow” or “Cancel my alarm for 6 PM.”
3. **Remember Anything (Note-taking):**This feature allows users to give the assistant information to remember, which can be retrieved later. For example, “Remember my meeting at 4 PM,” and later, “What do I have at 4 PM?”
4. **Attendance Management:**Users can track attendance by marking who is present or absent for a specific event or class. This is useful for students or professionals managing group activities or meetings.

# **Applications:**

1. **Home Automation:**
   * The voice assistant can control smart home devices such as lights, thermostats, door locks, and security cameras. Users can simply issue voice commands to turn devices on or off, adjust settings, or receive status updates.
2. **Personal Assistant:**
   * Users can utilize the voice assistant to manage their schedules, set reminders, create shopping lists, and organize tasks. The assistant can also provide updates on calendar events, weather forecasts, news headlines, and traffic conditions.
3. **Entertainment:**
   * The voice assistant can play music, podcasts, audiobooks, and radio stations based on user preferences. It can also provide recommendations for movies, TV shows, and online videos.
4. **Information Retrieval:**
   * Users can ask the voice assistant to search the internet for information on various topics, such as historical facts, definitions, recipes, product reviews, and DIY tutorials. The assistant can read out search results or summarize relevant content.
5. **Education and Learning:**
   * The voice assistant can assist students with homework, research, and studying by providing explanations, answering questions, and offering educational resources. It can also quiz users on academic subjects or language learning.
6. **Accessibility Support:**
   * The voice assistant can be a valuable tool for individuals with disabilities or special needs, providing hands-free access to information, communication, and assistance. It can help with tasks such as reading text aloud, dictating messages, and navigating digital interfaces.
7. **Health and Wellness:**
   * The voice assistant can offer health tips, exercise routines, meditation guidance, and nutritional advice. It can also track fitness goals, monitor sleep patterns, and remind users to take medications or hydrate.
8. **Business and Productivity:**
   * In a professional setting, the voice assistant can assist with tasks such as scheduling meetings, sending emails, managing contacts, and accessing business data. It can also facilitate collaboration by setting up conference calls or sharing documents.
9. **Interactive Projects:**
   * Developers and hobbyists can integrate the voice assistant into interactive projects and installations, such as interactive exhibits, art installations, voice-controlled robots, and gaming experiences.
10. **Custom Applications:**
    * The modular nature of the voice assistant project allows for customization and expansion to suit specific requirements. Users can develop custom skills, add new functionalities, and integrate with third-party services to create tailored solutions for unique use cases.

# **Execution Steps (Steps/procedure to show the working of the model):**

1. **Setup Raspberry Pi:**
   * Begin by connecting the Raspberry Pi to a monitor, keyboard, and mouse. Insert the SD card with the operating system installed (e.g., Raspbian).
   * Connect a microphone to the Raspberry Pi's USB port and a speaker to the audio output jack.
2. **Install Operating System and Dependencies:**
   * Follow the instructions to install the operating system (OS) on the Raspberry Pi. You can use NOOBS or download Raspbian directly.
   * Once the OS is installed, open a terminal and update the system by running commands like **sudo apt-get update** and **sudo apt-get upgrade**.
   * Install necessary Python dependencies and libraries for speech recognition, natural language processing, and text-to-speech conversion. For example, you can install the SpeechRecognition library using **pip install SpeechRecognition**.
3. **Write Python Code for Voice Assistant:**
   * Create a new Python file (e.g., **voice\_assistant.py**) and start writing code for your voice assistant.
   * Break down the functionality into separate functions, such as **listen\_for\_command()** to capture audio input, **recognize\_speech()** to convert speech to text, **process\_query()** to analyze user queries, **generate\_response()** to formulate responses, and **speak\_response()** to convert text to speech.
   * Utilize libraries such as SpeechRecognition for speech recognition, NLTK for natural language processing, and pyttsx3 for text-to-speech conversion.
4. **Test Voice Assistant Functions:**
   * Test each function of the voice assistant individually to ensure they work correctly.
   * Use print statements or logging to debug and verify the output of each function.
   * For example, test speech recognition by speaking commands and checking if they are accurately transcribed.
5. **Integrate with APIs and Services:**
   * Obtain API keys or credentials for services you want to integrate with, such as weather APIs, news APIs, or smart home APIs.
   * Modify your Python code to include functions that interact with these APIs, such as sending HTTP requests and parsing JSON responses.
   * Integrate the API calls into relevant parts of your voice assistant's logic, such as retrieving weather forecasts or fetching news headlines.
6. **Run the Voice Assistant Application:**
   * Save your Python code and run it in the terminal by executing **python voice\_assistant.py**.
   * Ensure that the microphone and speaker are connected and working properly.
   * Monitor the terminal output for any errors or exceptions that may occur during runtime.
7. **Interact with the Voice Assistant:**
   * Speak commands or ask questions aloud to interact with the voice assistant.
   * Observe how the assistant responds to your inputs and executes various tasks.
   * For example, ask the assistant about the weather forecast or instruct it to turn on a smart light connected to your home network.
8. **Evaluate Performance and Fine-Tuning:**
   * Evaluate the performance of the voice assistant in terms of accuracy, responsiveness, and overall user experience.
   * Identify any areas for improvement, such as speech recognition errors or slow response times.
   * Adjust parameters, fine-tune algorithms, and optimize code to enhance the assistant's performance.
9. **Demonstrate Use Cases:**
   * Demonstrate various use cases and scenarios to showcase the capabilities of the voice assistant.
   * Show how it can control smart home devices, retrieve information from the web, set reminders, play music, and perform other tasks.
   * Provide examples of real-world interactions to illustrate the practical utility of the voice assistant.
10. **Gather Feedback and Iterate:**
    * Gather feedback from users or observers who interact with the voice assistant.
    * Listen to their suggestions, comments, and concerns about the assistant's functionality and usability.
    * Use the feedback to make iterative improvements to the voice assistant, addressing any issues or shortcomings identified during testing and demonstration.

# **Conclusion:**

In conclusion, developing a system for a voice assistant using Raspberry Pi presents a compelling opportunity to explore the intersection of hardware and software, while creating a versatile and accessible tool for users. Through careful planning, integration of key components, and attention to detail, such a project can yield numerous benefits and opportunities for learning and innovation.  
  
By adding extended features like Time Table, Alarm, Remember Anything, and Attendance, the voice assistant project using Raspberry Pi offers users a powerful tool for personal productivity and management. Raspberry Pi’s affordability and flexibility make it a valuable platform for creating custom solutions that enhance user experience through automation and AI.

By leveraging the capabilities of Raspberry Pi, including its affordability, flexibility, and community support, developers can design voice assistants tailored to specific needs and preferences. Whether it's enhancing personal productivity, automating home tasks, or improving accessibility for users with disabilities, the possibilities are vast.

Furthermore, the process of developing a voice assistant system using Raspberry Pi offers valuable insights into IT infrastructure management, software development, natural language processing, and human-computer interaction. It encourages experimentation, problem-solving, and collaboration, fostering a deeper understanding of technology and its potential to improve our daily lives.

As the field of voice assistant technology continues to evolve, the role of Raspberry Pi as a platform for innovation and experimentation remains invaluable. Whether for educational purposes, personal projects, or commercial applications, the combination of Raspberry Pi and voice assistant development opens doors to creativity and exploration in the realm of human-machine interaction.

# **Teachers Remarks :**

|  |
| --- |
|  |
| 20 |

Marks Obtained