Assistant Robot

A Minor Project-II Report

Submitted by:

Devani Ayush (22BT04025)

Rudradatt Purohit (22BT04122)

Vraj Patel (22BT04155)

Soham Kava (23BT04D179)

In partial fulfilment of the requirement for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering (Artificial Intelligence / Machine Learning & Data Science and Internet of Things & Automation)

Semester - VII

At

Department of Computer Science and Engineering

School of Technology (SOT)

GSFC University,

Vigyan Bhavan, P. O. Fertilizer Nagar,

Vadodara – 391750 , Gujarat, India

November 2025

CERTIFICATE

This is to certify that the report submitted along with the project entitled **"Assistant Robot"** has been carried out by Devani Ayush (22BT04025), Rudradatt Purohit (22BT04122), Vraj Patel (22BT04155), Soham Kava (23BT04D179) were under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Science and Engineering – 6th Semester of GSFC University, Vadodara during the academic year 2024-25. The students have successfully completed the project activity under my guidance.

SIGNATURE (FACULTY MENTOR)

November, 2025

ACKNOWLEDGMENT

I would like to take this opportunity to express my deepest gratitude and appreciation to all those who have contributed to the successful completion of this project. First and foremost, I am immensely grateful to **Ms. Swati Saxena (Assistant Professor, CSE)** for her constant guidance, insightful suggestions and unwavering support throughout the development of this project. Her mentorship played a vital role in shaping the direction of our work and enhancing our understanding of both theoretical and practical aspects of the system. Finally, I would like to thank my teammates and peers for their collaboration and consistent efforts, which made this project a valuable and enriching learning experience. This project has truly helped us grow both technically and professionally and we are grateful for the opportunity to execute it.

ABSTRACT

The **Assistant Robot** is an intelligent system designed to help visitors and students in a laboratory. It autonomously follows a person using computer vision, providing a hands-free guide around the lab. **Human detection** is powered by **YOLOv8,** which tracks individuals in real-time through a mounted camera.

A **Kalman filter** smooths detected positions, ensuring stable and jitter-free navigation. Motor movements are controlled by an ESP32 microcontroller via **PWM signals** to high-power motor drivers. The robot maintains safe distances using bounding-box height heuristics. It features a **QA system** through a Flask-based web interface, allowing visitors to ask questions about lab projects. Questions are processed using **FAISS** vector search and **Groq LLM,** with answers converted to audio via **gTTS**.

All components communicate over serial and USB connections for real-time performance. This project demonstrates a practical integration of IoT, AI, and robotics to improve lab efficiency and visitor experience.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTERS

1. INTRODUCTION

The Lab Assistant Robot is an intelligent autonomous system designed to assist students and visitors in laboratory environments. It follows a person using computer vision, providing a hands-free guide around the lab and ensuring a smooth visitor experience. Human detection is achieved with YOLOv8 running on a Raspberry Pi, enabling real-time identification and tracking, while a Kalman filter smooths detected positions to reduce jitter and ensure stable navigation.

Motor movements are precisely controlled by an ESP32 microcontroller using PWM signals to high-power motor drivers, and the robot maintains safe distances using bounding-box height heuristics. A Flask-based web interface allows visitors to ask questions about lab projects in text or voice format, which are processed through FAISS vector search and Groq LLM to deliver accurate, context-aware answers. These answers are converted to speech using gTTS, providing an interactive audio response. Communication between the Raspberry Pi and ESP32 is handled via serial USB connections for real-time performance.

The modular design allows easy upgrades, such as multi-person tracking, obstacle detection, or offline LLM integration. This system effectively demonstrates the integration of IoT, AI, and robotics in a practical educational setting. It not only improves efficiency and reduces the need for human assistance but also showcases how modern embedded systems can create interactive and intelligent learning environments. Furthermore, the project serves as a hands-on example for students to explore robotics, vision-based AI, and IoT applications simultaneously.

I. PROJECT DESCRIPTION

➢ Problem Statement:

In university laboratories, visitors and students often require assistance to locate and understand various project demonstrations. Currently, this task is handled manually by lab assistants, leading to inefficiency and dependency on human availability. The aim is to develop an Autonomous Assistant Robot that can guide visitors, scan QR codes of projects and present relevant information through audio and visual means — thereby enhancing automation, interactivity and user experience in the lab environment.

➢ Objectives:

Human Detection and Following: Enable the robot to autonomously detect humans using YOLOv8 and follow them, ensuring smooth and accurate movement across the lab.

- **Motor Control via ESP32**: Implement precise motor control through ESP32 using serial commands from the Raspberry Pi, allowing responsive and coordinated robot navigation.
- **Safe Distance Maintenance:** Maintain a safe following distance using bounding-box height heuristics and a Kalman filter to smooth X-center coordinates and reduce jitter.
- **Interactive Question-Answer System:** Provide visitors with an interface to ask questions via text or voice and deliver relevant answers in real-time, along with audio feedback.
- **Knowledge Retrieval with FAISS and LLM:** Integrate a local FAISS vector search and Groq LLM to retrieve relevant lab information efficiently and generate human-like responses.
- **Modular and Upgradeable Design:** Design the system to be modular and robust, allowing easy future enhancements like obstacle avoidance, multi-person tracking, or migration to ROS/SLAM.
- **Enhanced Visitor Experience:** Provide a self-guided, intelligent, and interactive system that eliminates the need for a human guide, making laboratory exploration informative, safe and user-friendly.

II. FEATURES

1. **Autonomous Human Following** – The robot uses YOLOv8 and Kalman filtering to detect and smoothly follow a person in real-time.
2. **Precision Motor Control** – ESP32 microcontroller controls motors via PWM signals for accurate and stable navigation.
3. **Distance Management** – Maintains safe distance from the person using bounding-box height heuristics.
4. **Interactive QA System** – Visitors can ask questions via a Flask web interface and receive accurate answers.
5. **Voice Response** – Answers are converted to speech using gTTS for a hands-free interactive experience.
6. **Seamless Communication** – Raspberry Pi and ESP32 communicate via serial USB for real-time performance.
7. **Educational Tool** – Demonstrates integration of IoT, AI, and robotics in practical lab environments.

III. TOOLS AND TECHNOLOGIES USED

- **Microcontroller Unit (MCU)**: ESP32-WROOM, for motor controlling.
- **Web Camera**: Used to scan QR codes and capture the encoded data.
- **Display Module**: TFT LCD or similar display for visual output.
- **Speaker**: Provides audio playback for project descriptions.
- **Single Board Computer(SBCs)**: Raspberry Pi 5, offering processing, Wi-Fi, and camera interfacing.
- **Programming Languages**: C (for ESP32 firmware **in ARDUINO IDE**), Python (Movement Control, QA).

IV. MAJOR COMPONENTS USED

1. Hardware

1. **ESP32-WROOM Microcontroller** – Controls motors via PWM for precise movement.
2. **Raspberry Pi 4/5** – Handles vision processing and high-level logic.
3. **BTS7960 Motor Driver** – High-current H-bridge driver for DC motors.
4. **USB / Pi Camera** – Captures real-time images for human detection.
5. **DC Gear Motors** – Provides smooth locomotion and torque for robot.
6. **Li-ion Battery Pack** – Powers motors and microcontrollers reliably.
7. **Speakers / Display** – Provides audio output and information display.

2. Software

1. **Python** – Core language for vision, control, and QA system.
2. **Ultralytics YOLOv8** – Detects and tracks humans in camera frames.
3. **OpenCV** – Handles image processing and visualization tasks.
4. **Flask** – Hosts web-based interactive QA interface.
5. **FAISS Vector Search** – Enables fast retrieval from lab PDFs.
6. **Groq LLM** – Generates accurate answers to visitor questions.
7. **gTTS** – Converts text answers into speech for audio output.
8. **Arduino IDE / Espressif IDE –** For ESP32 motor-control programming and flashing.

2. LITERATURE REVIEW

PAPER REVIEW 1: Learn Buddy: Path Following Lab Assistant Robot https://www.researchgate.net/publication/377248449_Learn_Buddy_Path_Following_Lab_Assistant_Robot?utm_source

1. Aim / Purpose

- The paper proposes **Learn Buddy**, a lab assistant robot that follows a predetermined path in a laboratory environment.
- The robot is designed to guide students/visitors through lab stations in a fixed route, delivering project information and playing audio instructions.

2. Methods & Implementation (from available info)

- **Path following sensors**: Uses sensors (likely line-following or something similar) to keep the robot on the fixed path.
- **Predefined navigation path**: The path is defined in advance; robot moves along it without deviation.
- **Information delivery**: At certain stations, the robot provides audio instructions and displays project info. The focus is on guiding visitors in a structured way.

3. Results & Findings (from summary)

- It effectively automates structured guidance through lab stations.
- Enhances interactivity by combining visual display + audio.

4. Strengths (Advantages)

- **Simplicity**: Because the path is predefined, the control logic is simpler, and fewer sensors or computation resources are needed.
- **Reliable performance on fixed route**: On a known path, the robot can perform consistently with fewer unpredictability’s.

- **Effective for educational guidance**: Users can be taken through a lab in an organized sequence, which is useful for structured tours or classes.

5. Limitations & Drawbacks

- **Lack of adaptability**: The robot cannot deviate from the predefined path, so if an obstacle appears or the layout changes, it may fail or be obstructed.
- **No autonomous decision-making**: It doesn't choose routes or respond to dynamic situations (other robots, moving people, new obstacles).
- **Limited interaction**: The system seems to deliver information passively rather than supporting active query answering or flexible project access.

6. Relevance to Your Assistant Robot Project

- The concept of guiding visitors through stations with display + audio is directly relevant; you can adapt the station-information delivery part.
- Fixed path systems are easier to implement; you can use this idea as a fallback when autonomous navigation is uncertain (e.g. before map is reliable).
- However, your design improves upon this by offering **autonomous navigation**, **AI-powered Q&A**, **manual override**, and **dynamic project access**, which Learn Buddy lacks.

7. Critical Appraisal & Suggestions for Improvement (inspired by this work)

- **Introduce flexibility**: Add capability to switch from fixed path to manual or autonomous mode when needed.
- **Obstacle handling**: Real-time obstacle detection + rerouting or stopping to ensure safety.
- **User-driven interaction**: Rather than just audio display, allow visitors to ask questions or skip/return to stations.
- **Map-based or SLAM based localization**: Fixed paths are good but SLAM or pose estimation gives more robustness in dynamic or complex lab settings.

PAPER REVIEW 2: ***A Comparative Study on LIDAR and Ultrasonic Sensor for Obstacle Avoidance Robot Car***

https://www.researchgate.net/publication/372211676_A_comparative_study_on_LIDAR_and_Ultrasonic_Sensor_for_Obstacle_Avoidance_Robot_Car

1. Aim / Purpose

To compare the performance of LiDAR vs. ultrasonic sensors for obstacle avoidance in robot cars—specifically focusing on **accuracy, range, and reaction time**. The goal is to help in selecting the right sensing technology depending on application needs.

2. Methods / Implementation

- Built a robot car platform equipped with **LiDAR (360° scanning)** and ultrasonic sensors.
- Mounted sensors to cover obstacle detection; used a microcontroller / single-board computer (Raspberry Pi 3) for processing.
- Measured detection range and accuracy under different environmental conditions (light levels, color of objects) and at different obstacle distances (30 cm, 60 cm).
- Performed tests for avoiding obstacles of different sizes, shapes, materials, including transparent and opaque objects.

3. Results & Findings

- **LiDAR** had higher precision, longer detection range, and better resolution than ultrasonic sensors. Inaccuracy was low (~0.8%), even under varying lighting and object color.
- **Ultrasonic sensors** were less expensive, had faster reaction time, but were limited in range, and their detection was less reliable for certain shapes or in certain environmental conditions.
- Transparent materials (e.g., acrylic, polycarbonate) were problematic for LiDAR (due to reflection / absorption of laser light).
- Study concluded that choice between LIDAR vs ultrasonic depends on requirements: precision vs cost vs speed vs environment. Suggests combining both sensors for improved obstacle avoidance in some cases.

4. Strengths

- Thorough comparison under realistic test conditions (different distances, object types, lighting).
- Quantitative metrics on detection error; helps in designing real robot systems.
- Clear trade-off analysis: cost vs performance.

5. Limitations

- Reactive obstacle avoidance only; no higher-level planning or path following considered.
- Tests only up to certain distances; heavy dependency on line of sight for LiDAR.
- Sensor fusion (combining ultrasound + LiDAR) not implemented, though suggested.
- Environmental factors like dust, strong sunlight may degrade LiDAR which is noted but not fully explored.

6. Relevance to Your Assistant Robot Project

- Justifies your choice to include LiDAR: you'll get better mapping, obstacle detection, safer autonomous navigation.
- Encourages using dual sensors or sensor fusion if cost allows (e.g., ultrasonic fallback for close range or redundant detection).
- Helps decide mounting, lighting requirements, and detection reliability for lab items or obstacles.

PAPER REVIEW 3: ***Socially Assistive Robot as Laboratory Safety Assistant for Science Students***

https://www.researchgate.net/profile/Mohammad-Hasnine/publication/381003766_Socially_Assistive_Robot_as_Laboratory_Safety_Assistant_for_Science_Students/links/66591adfbc86444c721eebe1/Socially-Assistive-Robot-as-Laboratory-Safety-Assistant-for-Science-Students.pdf?origin=scientificContributions

1. Aim / Purpose

To propose a framework (LSA: Laboratory Safety Assistant) using a socially assistive robot (Misty II Plus) to monitor and intervene in hazardous student behavior in science labs. The aim is to reduce accidents resulting from violations of lab safety protocols especially when supervisors are absent.

2. Methods / Implementation

- Framework components: Misty II Plus robot, behavioural sensor suite, dashboards for students and teachers, behavior detection & analysis server, user-centered / co-design process.
- Misty II Plus is equipped with advanced sensors: video / 4K camera, Time-of-Flight sensors, IR projector, IMUs, bump sensors etc. For detecting proximity, posture, protective equipment usage (goggles/apron), open flames, etc.
- Behaviours identified as risky (not using PPE, leaving flame unattended, running, etc.). A dashboard provides real-time or near real-time feedback/intervention. Teachers also have visibility.
- Followed co-design with actual lab users (students, supervisors) to identify requirements.

3. Results & Findings

- Users accept the idea; co-design indicates many safety behaviours are often violated in labs.

- The proposed framework theoretically satisfies requirements: detection, feedback, companionship. Demonstrations show that the system can detect and flag risky behavior (at least in test settings).
- The dashboards allow teachers to monitor, students get feedback. The social aspect helps engagement.

4. Strengths

- Focus on safety + user behavior, which is critical in lab environments.
- Uses a socially assistive robot, not just mechanical sensors — adds engagement.
- Co-design ensures relevance to user needs; not just technical approach but also user acceptability.

5. Limitations

- Mostly conceptual / prototypical: less on full real-world performance metrics (false positives/negatives, reaction delays etc.).
- Potential privacy concerns: monitoring students, collecting video / behavior data.
- Likely dependency on environment (lighting, positioning, sensor obstructions).
- The robot is misty II plus, which may have limited payload or mobility; concept doesn’t deeply address navigation between stations or moving across larger lab spaces.

6. Relevance to Your Assistant Robot Project

- Provides insight into behavior detection, risk management, and interfaces (dashboards) that can complement your robot’s interactive AI Q&A.
- Helps in defining safety requirements: detection of PPE, correct posture, etc., which you might integrate.
- Emphasizes importance of user-centered design and behavior monitoring + feedback, which can improve the robot’s acceptability and safety.

3. SYSTEM DESIGN

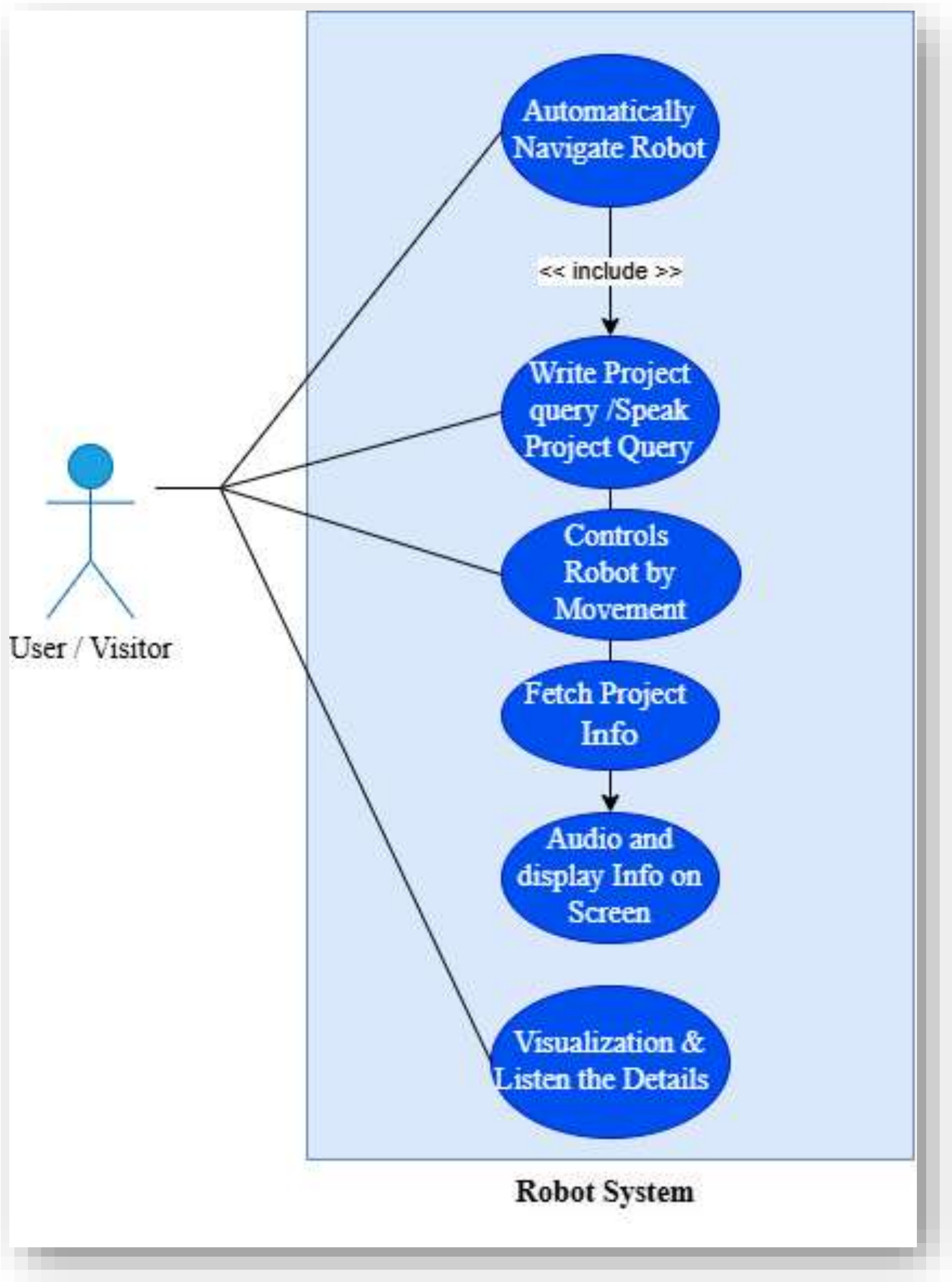


Fig. (1) UseCase Diagram of a System

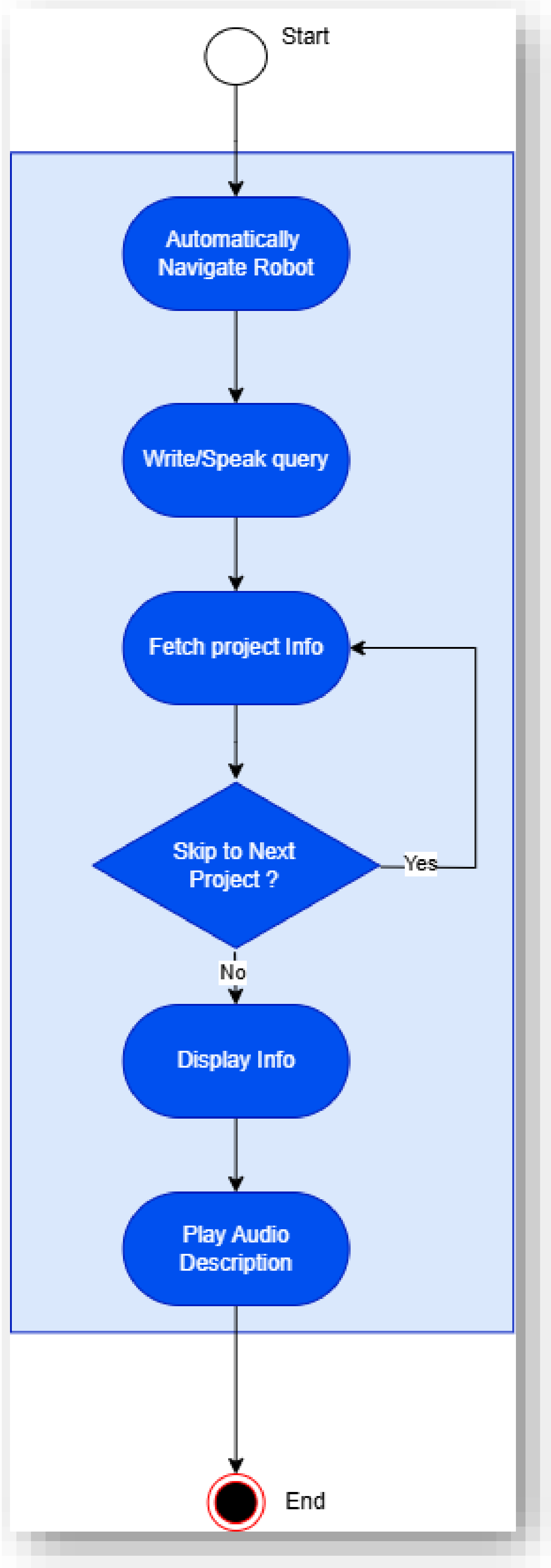


Fig. (2) Activity Diagram of a System

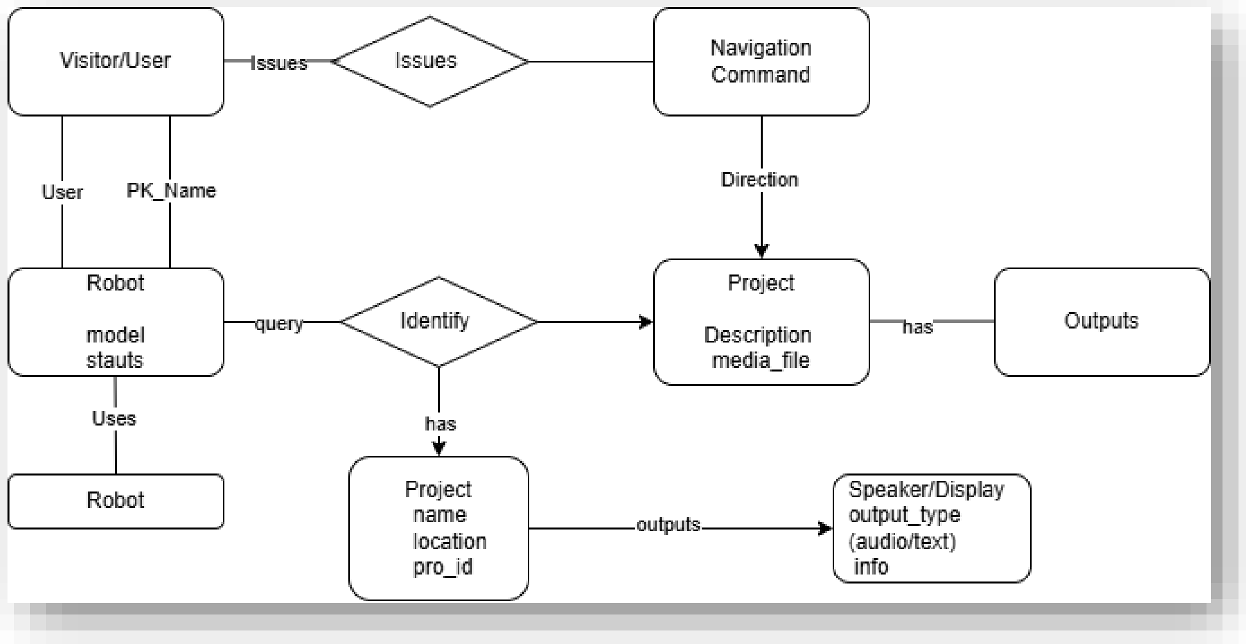


Fig. (3) Entity-Relationship (E-R) Diagram of a System

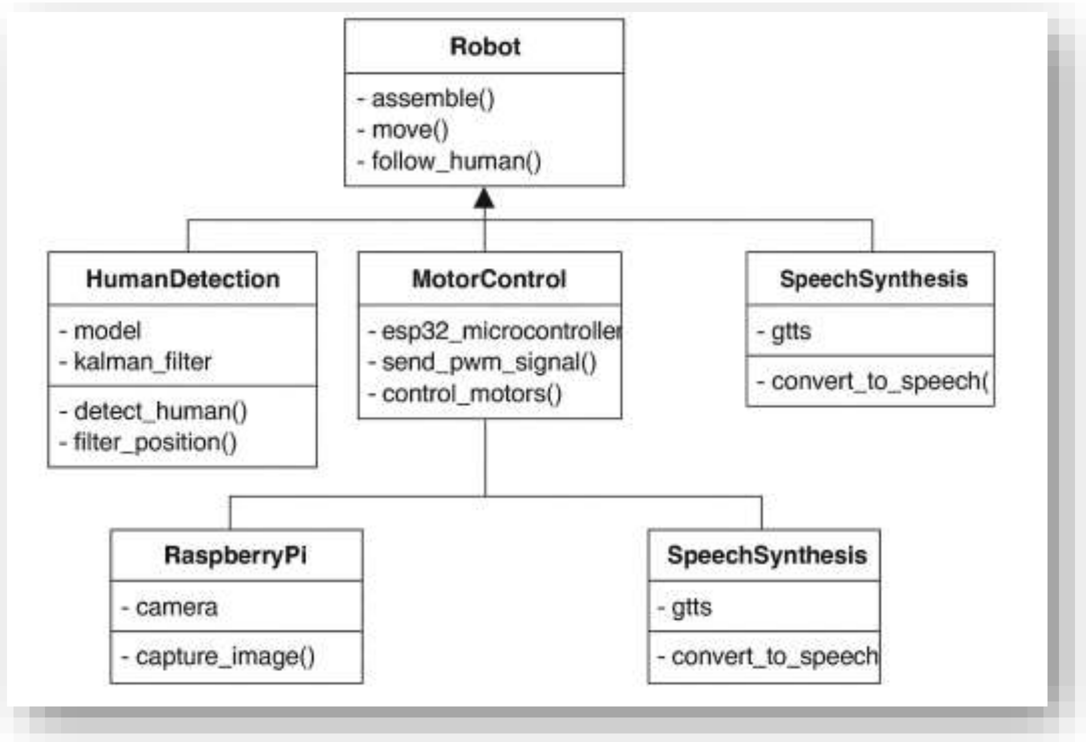


Fig. (4) Class Diagram of a System

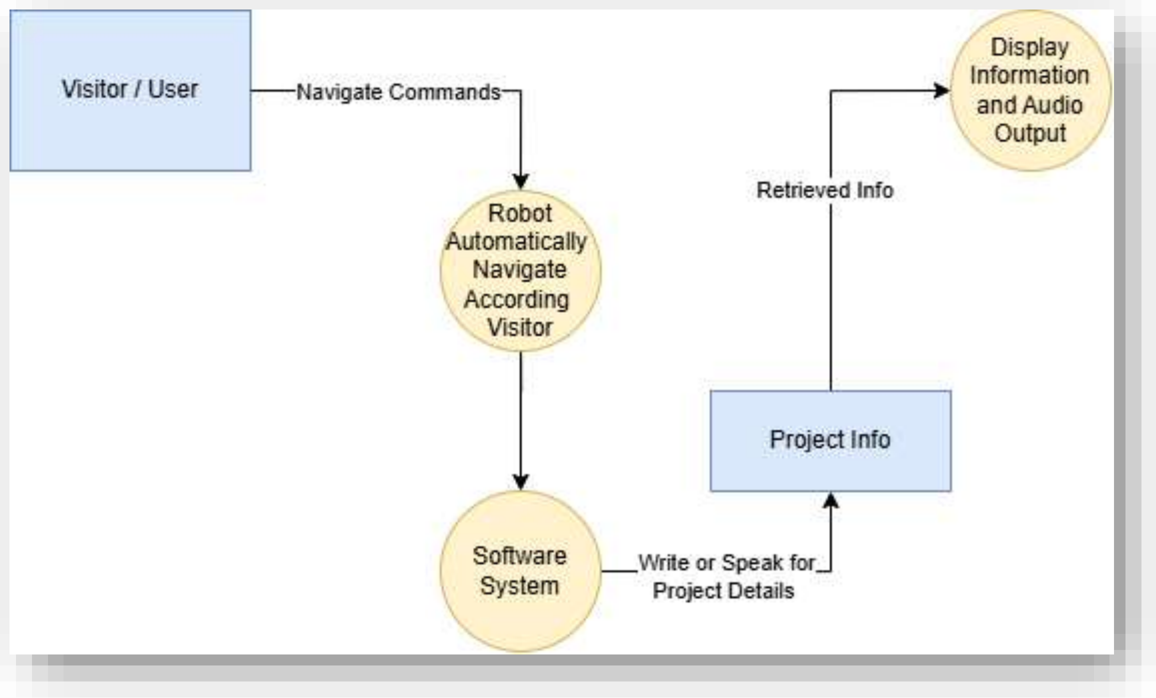


Fig. (5) Data Flow Diagram of a System

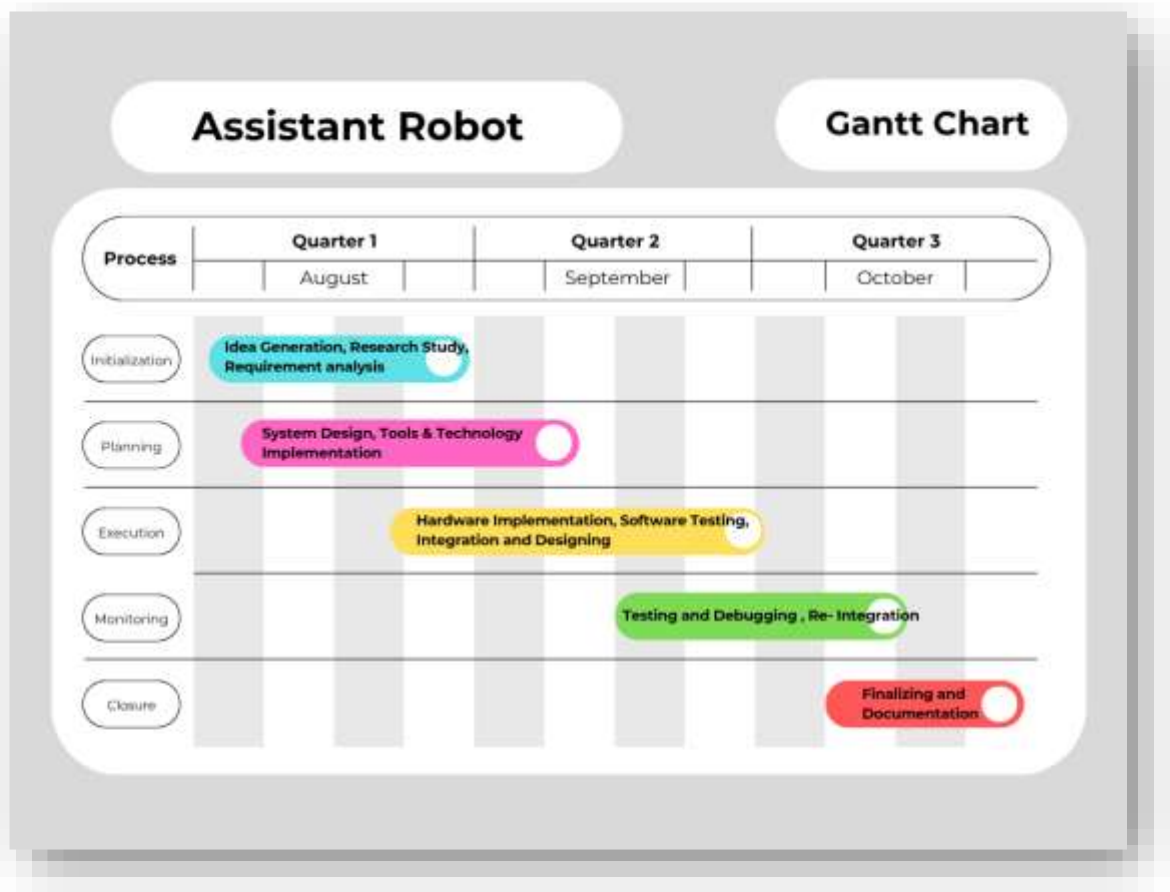


Fig. (7) Flow chart of a Development Process (Gantt Chart)

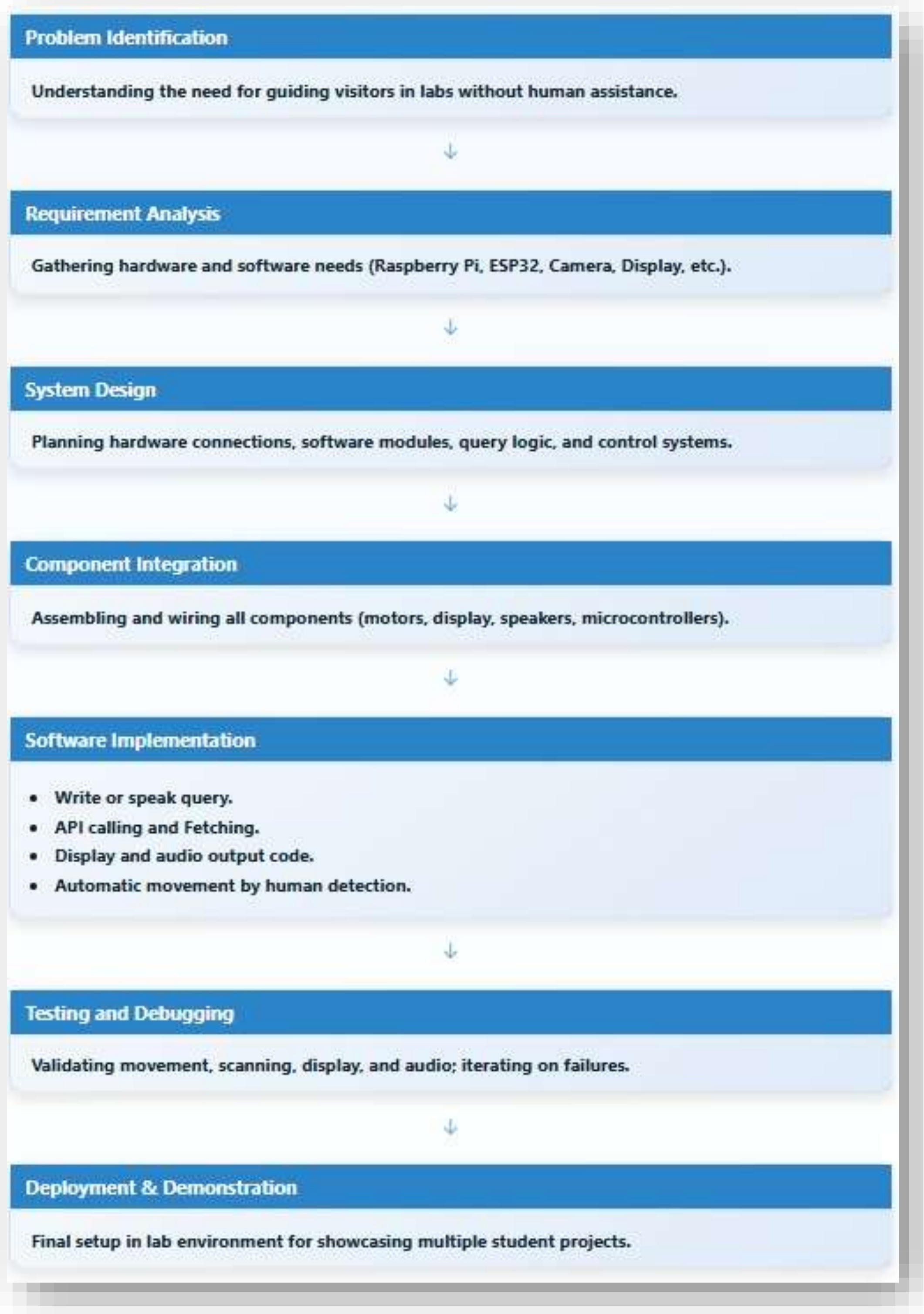


Fig. (6) Project Developmnet approach

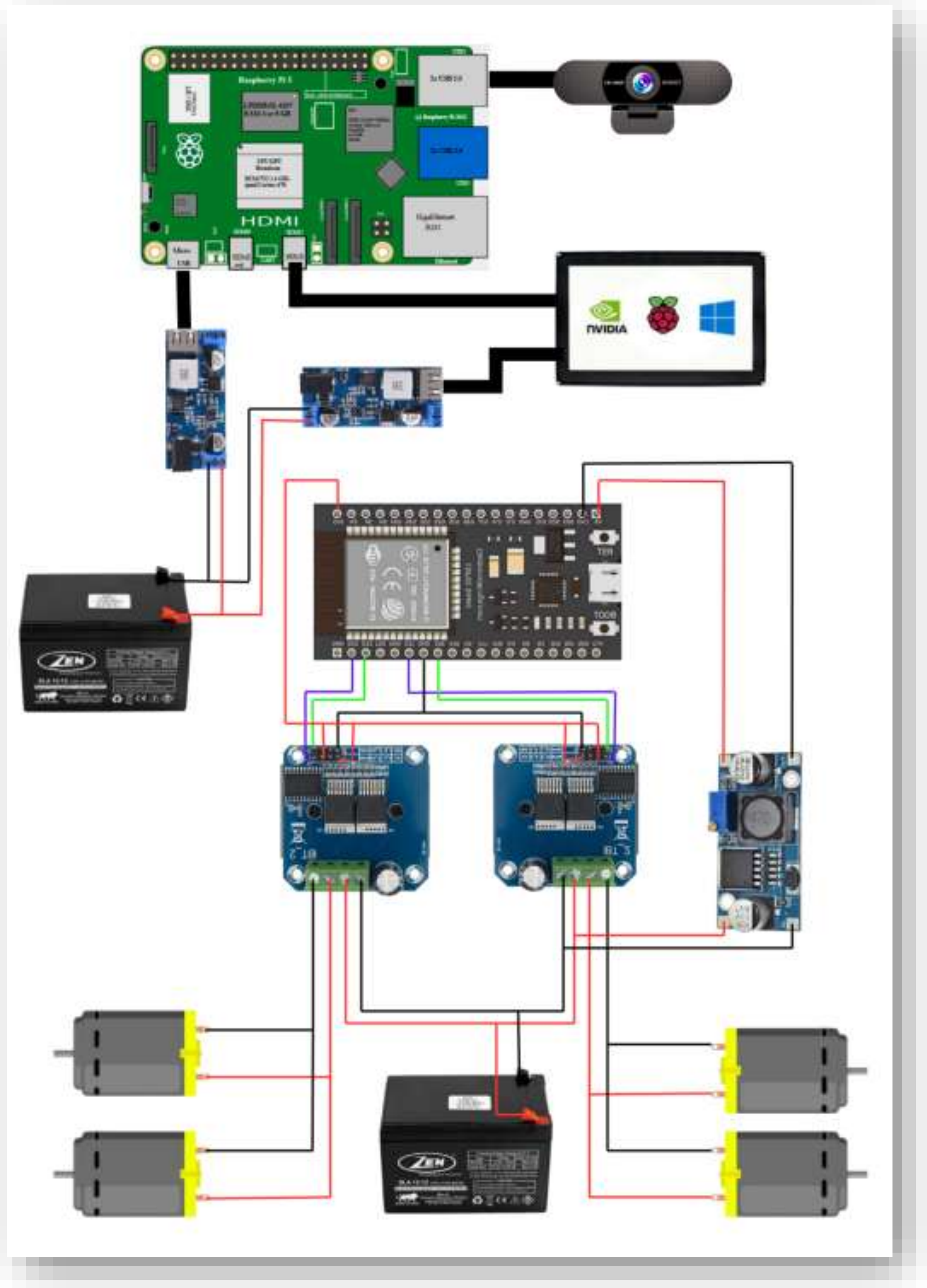
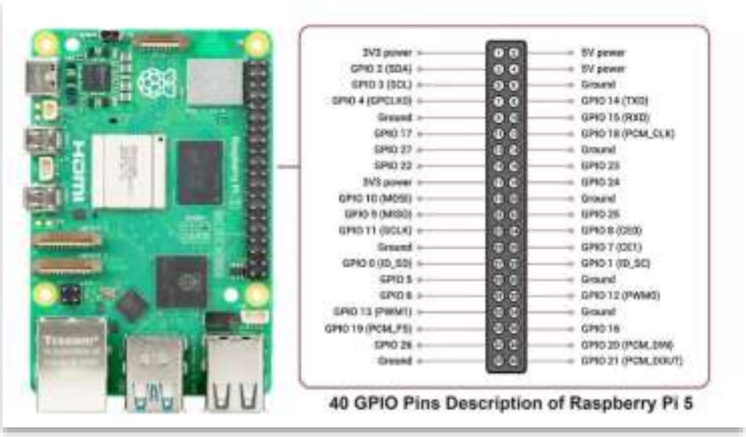


Fig. (7) Circuit Diagram

4. COMPONENTS DESCRIPTION

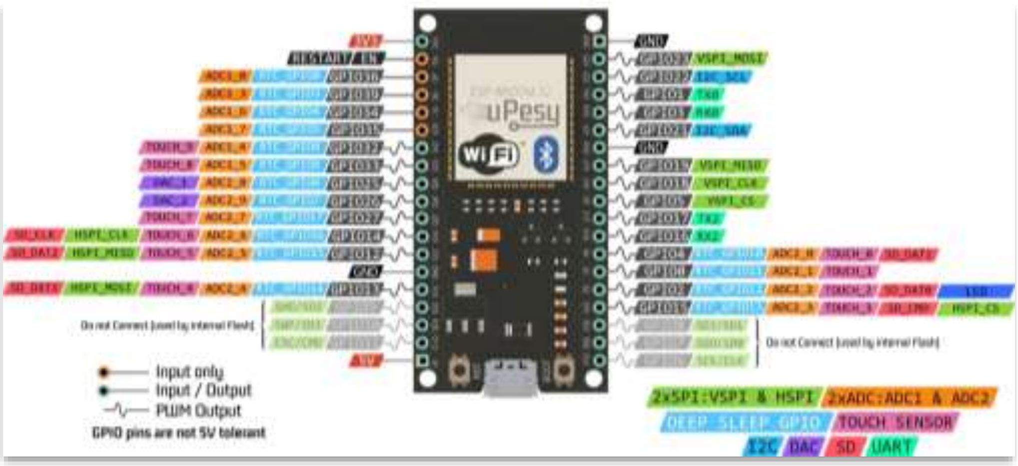
1. Raspberry Pi 5

The Raspberry Pi 5 is a powerful single-board computer used as the brain of the robot. It handles image processing for QR code scanning, manages data flow, and controls peripheral devices. Its high processing speed and GPIO support make it ideal for multi-sensor and multimedia applications.



2. ESP32-WROOM Module

The ESP32-WROOM is a dual-core microcontroller with built-in Wi-Fi and Bluetooth capabilities. In this project, it's responsible for communication with the mobile app, motor control via the BTS7960 driver, and real-time execution of navigation commands. It offers low-power operation and high-speed wireless connectivity.

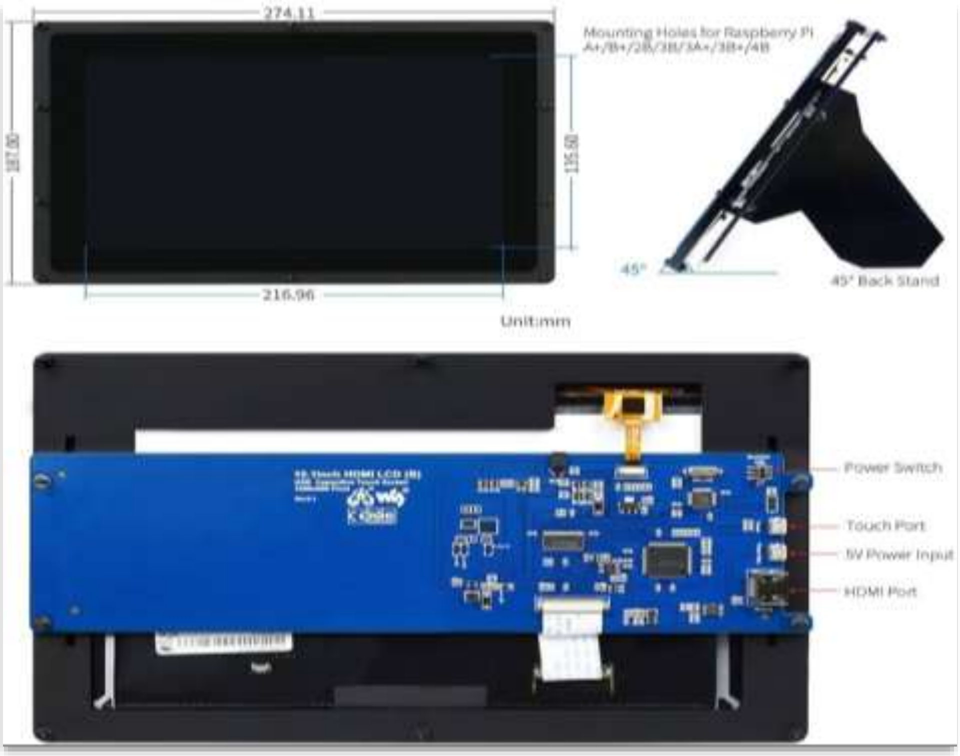


3. E900 Meet Webcam

The E900 Meet webcam serves as the visual input device for scanning QR codes at project stations. It captures clear images even under variable lighting conditions and sends them to the Raspberry Pi for decoding. Its plug-and-play nature ensures easy USB integration with the Raspberry Pi.

4. Wave share 10.1 Inch Display

This HDMI-compatible display provides a high-resolution visual output of project information retrieved from the database. It acts as an interface between the robot and the user, making information accessible and engaging. The touchscreen variant can further enhance interactivity, if implemented.

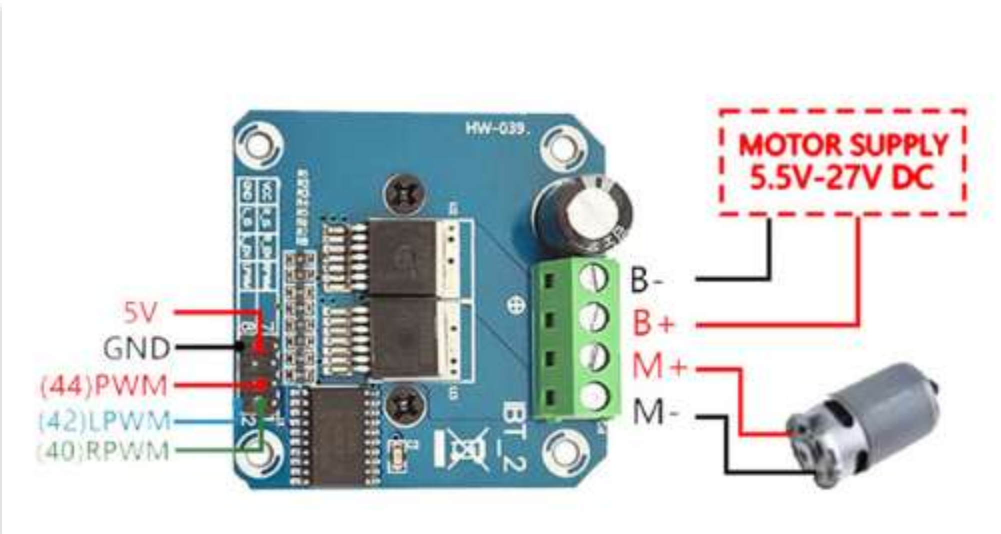


5. OG555 High Torque DC Motor (12V, 100RPM, 173.6 N-cm)

These motors are responsible for the physical movement of the robot. With high torque output, they provide enough power to drive the robot on different surfaces while maintaining smooth and controlled navigation. The 100 RPM speed offers a balance between speed and stability.

6. BTS7960 Motor Driver Module

This dual H-Bridge motor driver can handle high current (up to 43A), making it suitable for controlling heavy-duty DC motors. It receives PWM signals from the ESP32 to regulate motor speed and direction. Its robust design ensures safe and efficient motor control.

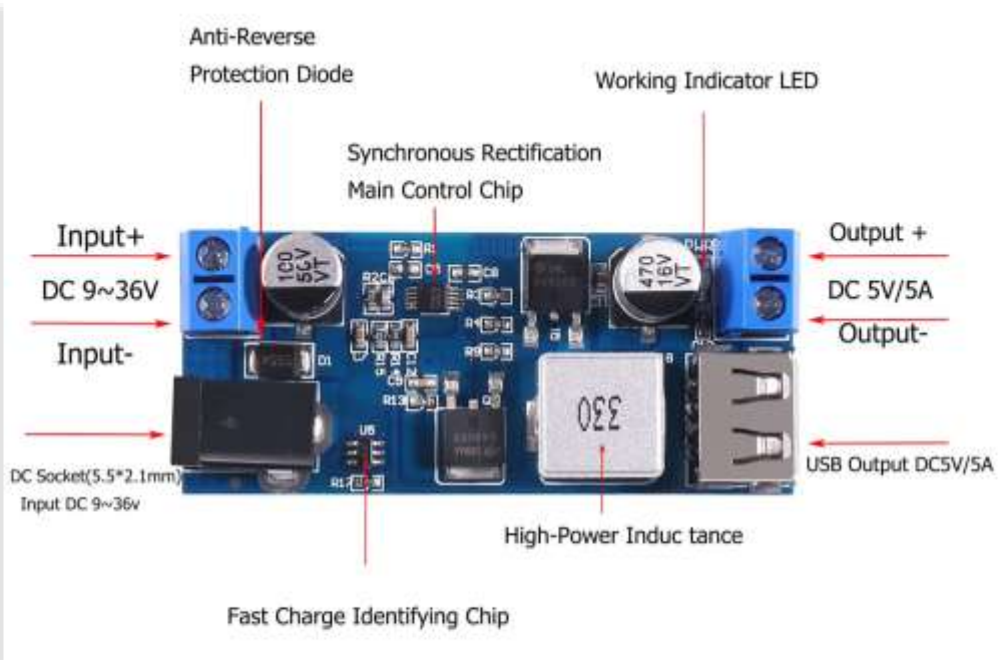


7. 12V 12Ah Rechargeable Battery

The battery supplies power to the entire robot system, including motors, ESP32, and the display. With a high capacity of 12Ah, it ensures long operational time without frequent recharging, making it ideal for continuous event use.

8. USB Pin Buck Converter (12V to 5V)

This step-down converter regulates the 12V battery supply down to 5V, which is required to safely power devices like the Raspberry Pi and webcam. It ensures consistent and stable voltage, preventing damage to sensitive electronics.



9. Bluetooth Speaker

The speaker delivers clear audio output, providing spoken project descriptions to enhance visitor engagement. It connects wirelessly to the Raspberry Pi and helps in making the system accessible to visually impaired users or enhancing public communication.

10. Jumper and Connecting Wires

Jumper wires are essential for connecting all electronic components on the breadboard or PCB. They ensure proper signal and power transmission between modules. These provide flexibility and modularity during both prototyping and implementation stages.

11. Raspberry Pi OS:

Raspberry Pi OS is a Debian-based operating system optimized for Raspberry Pi hardware. In this project, it manages the main operations such as camera handling, QR code scanning, GUI display, and communication with hardware. It ensures efficient multitasking in a lightweight environment.

12. Python:

Python is the core programming language used to develop the robot's logic. It handles QR code scanning, database access, GUI updates, and text-to-speech output. Its simplicity and vast library support make it ideal for this integrated system.

13. OpenCV:

OpenCV is used for real-time computer vision tasks like QR code detection. Integrated with Python, it enables the robot to process camera input efficiently and decode QR data for further processing.

14. **Flask** – Flask is a lightweight Python web framework used to build web applications quickly. It provides routing, templating, and request handling features with minimal overhead. In the robot system, Flask hosts an interactive QA interface accessible through a browser. It allows smooth communication between the visitor and the robot's backend services.

15. **FAISS Vector Search** – FAISS is a library designed for efficient similarity search and clustering of dense vectors. It allows rapid retrieval of relevant documents from large datasets, like lab PDFs. In the robot system, it enables fast access to project information for visitor queries. FAISS ensures low-latency, accurate search results.

16. **Groq LLM** – Groq LLM is a large language model optimized for fast and accurate natural language understanding and generation. It interprets visitor questions and provides meaningful, context-aware answers. The model integrates seamlessly with vector search to enhance response quality. In the robot system, it powers intelligent QA functionality.

17. **gTTS (Google Text-to-Speech)** – gTTS converts textual responses into natural-sounding speech using Google's TTS API. It allows the robot to communicate answers audibly to visitors. The library supports multiple languages and accents for better interaction. In the robot system, gTTS delivers clear, real-time audio feedback.

18. **Arduino IDE (ESP32 Programming):**

The Arduino IDE is used to program the ESP32 microcontroller, which handles motor movement and navigation. It communicates with Raspberry Pi to execute manual navigation commands.

5. IMPLEMENTATION OF PROJECT

(Methodologies & Approach)

i. Overview

The updated Lab Assistant Robot is designed to autonomously guide visitors and provide information about laboratory projects. It integrates computer vision, AI-based QA systems, and audio feedback to deliver a seamless visitor experience. The system leverages real-time human detection, document retrieval, and natural language processing to respond to user queries efficiently. Hardware and software components work in unison to ensure accurate navigation, information display, and interactive communication.

ii. Project Methodologies

The project implementation follows a combination of Agile and Iterative development methodologies:

1. **Requirement Analysis:** Identifying visitor interaction needs, lab environment constraints, and system functionalities.
2. **System Design:** Creating hardware schematics, software architecture, and workflow diagrams.
3. **Incremental Development:** Hardware and software modules were developed in parallel, tested individually, and integrated step-by-step.
4. **Testing and Validation:** Each subsystem, including navigation, QA interface, and audio output, underwent rigorous testing for accuracy and reliability.
5. **Deployment:** The complete system was integrated and tested in the laboratory environment, ensuring real-time functionality.

This methodology ensured flexibility, error minimization, and the ability to incorporate iterative improvements based on testing feedback.

iii. Hardware Integration

The hardware components were carefully selected and integrated to support navigation, sensing, and actuation:

- **ESP32 Camera Module:** Captures real-time video for human detection.
- **Motors and Motor Drivers (BTS7960):** Enable smooth movement and precise navigation.
- **Speakers:** For audio output of visitor query responses.
- **Battery and Power Management:** Ensures uninterrupted operation of sensors, motors, and processing units.
- **Connectivity Modules:** Supports wireless communication for data exchange with backend servers.

All hardware components were interfaced through a central microcontroller (ESP32/Raspberry Pi), ensuring synchronized operation of navigation, sensing, and feedback systems.

iv. Software Integration

The software system combines multiple AI and web technologies for functionality:

- **Computer Vision (YOLOv8):** Detects and tracks humans in real-time video streams.
- **Flask Web Server:** Hosts the interactive QA interface for visitor queries.
- **FAISS Vector Search:** Rapidly retrieves relevant information from lab documents.
- **Groq LLM:** Processes visitor questions and generates accurate, context-aware answers.
- **gTTS (Text-to-Speech):** Converts textual answers into audible speech for interaction.

The software modules communicate through APIs and data pipelines, allowing smooth integration of detection, question processing, and audio feedback. The complete system ensures real-time performance, accurate information retrieval, and natural visitor interaction.

6. MODEL OVERVIEW

The Assistant Robot integrates real-time human detection, intelligent QA, and audio feedback to assist visitors in navigating the laboratory. It combines hardware for mobility and sensing with AI-driven software for information retrieval and communication. The system ensures smooth interaction by linking visitor queries to document search and natural language response generation.

i. Key Functional Components

- **Human Detection Module:** Uses YOLOv8 to detect and track visitors.
- **QA Interface:** Flask-based web interface for accepting user queries.
- **Information Retrieval:** FAISS Vector Search for fast document lookup.
- **Answer Generation:** Groq LLM generates context-aware responses.
- **Audio Output:** gTTS converts answers into speech for visitor interaction.
- **Mobility & Navigation:** Motor drivers and sensors enable autonomous movement.

ii. System Workflow

Visitor Approaches → Camera Captures Video → YOLOv8 Detects Human → Query Entered on QA Interface → FAISS Retrieves Relevant Documents → Groq LLM Generates Response → gTTS Converts Text to Speech → Robot Communicates Answer → Robot Follows Visitor / Guides to Lab

7. CHALLENGES & FUTURE ENHANCEMENTS

➢ CHALLENGES

- **Accurate Human Detection:** Ensuring YOLOv8 reliably detects humans in varying lighting and crowded lab conditions is challenging. False positives or missed detections can affect navigation and interaction.
- **Real-Time Information Retrieval:** FAISS vector search must retrieve relevant documents quickly from a large dataset. Delays or inaccurate retrieval can affect the quality of visitor responses.
- **Audio Output Clarity:** Converting text to speech using gTTS must ensure clear, audible responses in noisy environments. Poor audio quality can hinder communication with visitors.
- **Battery and Power Management:** Maintaining uninterrupted operation of motors, sensors, and processing units is critical. Power fluctuations or low battery can disrupt system performance.
- **Integration of Hardware and Software:** Synchronizing movement, detection, QA processing, and audio output in real time is complex. Any delay in one module can affect overall system responsiveness.

➢ FUTURE ENHANCEMENT

- **Advanced Obstacle Avoidance:** Integrate LiDAR or ultrasonic sensors to detect and avoid obstacles, ensuring safer and more efficient navigation in complex lab environments.
- **Multilingual Support:** Enhance gTTS and Groq LLM to support multiple languages and regional accents, making the robot more accessible to a diverse set of visitors.
- **Offline QA Functionality:** Develop a local database and offline LLM processing to allow the robot to answer queries without relying on continuous internet connectivity.
- **Expanded Document Integration:** Incorporate additional lab resources, multimedia content, and real-time project updates to provide more comprehensive visitor guidance and interactive experiences.

8. CONCLUSION

The Assistant Robot successfully integrates advanced AI, computer vision, and audio feedback to provide an interactive and autonomous visitor guidance system. By combining YOLOv8 for human detection, FAISS for document retrieval, and Groq LLM with gTTS for intelligent question answering, the robot ensures seamless communication and accurate information delivery. The integration of hardware and software modules demonstrates real-time performance and efficient navigation within the laboratory environment. While challenges such as environmental constraints and system integration exist, the current implementation provides a strong foundation for future enhancements. Limitations like obstacle detection and offline functionality can be addressed in subsequent versions to improve reliability and accessibility. Overall, this system showcases the potential of IoT, AI, and robotics in creating intelligent, user-friendly laboratory assistance solutions. It not only enhances visitor experience but also paves the way for more advanced autonomous service robots in educational and research environments.

9. REFRENCES

1. Raspberry Pi Ltd., *Raspberry Pi 5 Product Brief*, [Online]. Available: https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf

2. Espressif Systems, *ESP32 Technical Reference Manual*, [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

3. Infineon Technologies, *BTS7960 Datasheet*, [Online]. Available: https://image.dfrobot.com/image/data/DRI0018/BTS7960.pdf

4. HandsOn Technology, *BTS7960 Motor Driver Module Datasheet*, [Online]. Available: https://www.handsontec.com/dataspecs/module/BTS7960%20Motor%20Driver.pdf

5. OpenCV, *OpenCV Python Tutorials*, [Online]. Available: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

6. Altium, *Building an AI Lab Assistant*, [Online]. Available: https://resources.altium.com/p/building-ai-lab-assistant

7. M. T. Santos et al., "Design and Implementation of a Lab Assistant Robot Based on AI and IoT," *Micromachines*, vol. 6, no. 4, p. 66, 2021. [Online]. Available: https://www.mdpi.com/2411-9660/6/4/66