DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 111, India

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

MINI PROJECT (22IDT28)

Report on

Sarathi - The Smart Wheelchair

Submitted in partial fulfillment for the award of the degree of

Bachelor of Engineering
in
Computer Science and Design

Submitted by

Aditi Mishra	**1DS24CG004**
Ambar Pavan Kumar	**1DS24CG007**
Sneha G Bhat	**1DS24CG050**
Tejaswini Shankar	**1DS24CG056**

Under the Guidance of
Dr Suma V
Vice Principal, Head
Department of Computer Science and Design
DSCE, Bengaluru

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI-590018, KARNATAKA, INDIA

2024-25

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 111, India

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

CERTIFICATE

Certified that the mini project report entitled **"Sarathi - The Smart Wheelchair"** carried out by **Aditi Mishra, Ambar Pavan Kumar, Sneha G Bhat, Tejaswini Shankar** bearing USNs **1DS24CG004, 1DS24CG007, 1DS24CG050, 1DS24CG056** bonafide students of **DAYANANDA SAGAR COLLEGE OF ENGINEERING**, an autonomous institution affiliated to VTU, Belagavi in partial fulfillment for the award of Degree of **Bachelor of Engineering in Computer Science and Design** during the year **2024-2025**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements with respect to the work prescribed for the said Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Dr Suma V **Vice Principal, Head** **Dept. of CSD, DSCE Bengaluru**	**Dr. Suma V** **Vice Principal, Head** **Dept. of CSD, DSCE, Bengaluru**	**Dr. B G Prasad** **Principal** **DSCE, Bengaluru**

Name of the Examiners	Signature with date
1.	....
2.	....

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 111, India

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

DECLARATION

We, **Aditi Mishra (1DS24CG004), Ambar Pavan Kumar (1DS24CG007), Sneha Ganesh Bhat (1DS24CG050) and Tejaswini Shankar (1DS24CG056),** respectively, hereby declare that the mini project work entitled "Sarathi - Wheelchair" has been independently done by us under the guidance of **Dr Suma V, Vice Principal, Prof & Head CSD department** and submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering in Computer Science and Design** at **Dayananda Sagar College of Engineering**, an autonomous institution affiliated to VTU, Belagavi during the academic year 2024-2025.

We further declare that we have not submitted this report either in part or in full to any other university for the award of any degree.

Aditi Mishra	**1DS24CG004**
Ambar Pavan Kumar	**1DS24CG007**
Sneha Ganesh Bhat	**1DS24CG050**
Tejaswini Shankar	**1DS24CG056**

PLACE:

DATE:

ACKNOWLEDGEMENT

The satisfaction and euphoria accompanying the successful completion of any task would be incomplete without the mention of people who made it possible and under constant guidance and encouragement the task was completed. We sincerely thank the **Management of Dayananda Sagar College of Engineering, Bengaluru.**

We express our sincere regards and thanks to **Dr. B G Prasad, Principal, Dayananda Sagar College of Engineering, Bengaluru.** His constant encouragement, guidance and valuable support have been an immense help in realizing this mini project work.

We express our sincere regards and thanks to **Dr. Suma V, Vice Principal, Head, Department of Computer Science and Design, Dayananda Sagar College of Engineering, Bengaluru.** Her incessant encouragement, guidance and valuable technical support have been an immense help in realizing this mini project. Her guidance gave us the environment to enhance our knowledge, and skills and to reach the pinnacle with sheer determination, dedication, and hard work.

We would like to express profound gratitude to our guide, **Dr Suma V, Vice Principal,Head, Department of Computer Science and Design, Dayananda Sagar College of Engineering, Bengaluru** who has encouraged us throughout the mini project. Her moral support enabled us to complete our work successfully.

We express our sincere thanks to Mini Project Coordinator **Prof. Nayana Shinde, Assistant Professor** of the **Department of Computer Science and Design** for her continuous support and guidance. We thank all teaching and non-teaching staff of the Department of Computer Science and Design for their kind and constant support throughout the academic Journey.

Aditi Mishra 1DS24CG004
Ambar Pavan Kumar 1DS24CG007
Sneha Ganesh Bhat 1DS24CG050
Tejaswini Shankar 1DS24CG056

ABSTRACT

This paper outlines the design and development of "Sarathi," a two-wheeled smart wheelchair prototype designed to enhance indoor mobility for individuals with special needs. The primary objective is to deliver an economical, accessible navigation system that supports both manual (forward and backward) movement and autonomous room-to-room navigation, all controlled through a user-friendly Flutter based mobile app. The app features a simplified one-tap interface that enables users to navigate the wheelchair to predefined rooms, allowing greater independence in daily tasks. It also includes dual-role support for both patients and guardians, chat functionality, and an emergency SOS alert system.

To ensure affordability, the system utilizes easily sourced components commonly found in embedded hardware setups, keeping the design feasible and replicable. The Flutter-based mobile application supports essential features such as user authentication (login and signup), role-based interfaces for patients and guardians, and local data storage using SharedPreferences. Patients can send SOS alerts, communicate via chat, and navigate using the interface, while guardians receive alerts and can monitor or assist through their dedicated screen. In the development process, several navigation algorithms—such as LiDAR, GPS, BLE, and A*—were evaluated, with a pre-programmed path approach ultimately chosen for autonomous mobility. The hardware consists of two DC geared motors, an L298N motor driver, and an ESP32 microcontroller for its wireless capabilities and versatility. Predefined paths are programmed to correspond with specific household locations, enabling the wheelchair to navigate independently.

Initial trials confirm that the prototype performs effectively in both manual and autonomous modes, responding swiftly to commands and transitioning smoothly between destinations. This validates the system's potential as a low-cost assistive solution for controlled indoor environments. "Sarathi" is particularly suitable for home care, rehabilitation centers, and assisted-living facilities. Future developments may include voice commands, real-time mapping, obstacle detection using low-cost sensors, and scaling the design into a full-size, production-ready wheelchair.

Keywords: *Wheelchair, Autonomous Navigation, ESP32, Healthcare, Indoor mobility, Application, Disabilities, Internet of Things(IoT), Flutter.*

Table of Contents

LIST OF TABLES

LIST OF FIGURE

LIST OF ABBREVIATIONS

Abbreviation	Full Form
ESP32	Espressif Systems Protocol 32-bit
DC	Direct Current
L298N	Dual H-Bridge Motor Driver IC
SOS	Save Our Souls (used as an emergency signal)
UI	User Interface
UX	User Experience
IoT	Internet of Things
AI	Artificial Intelligence
IDE	Integrated Development Environment
SDK	Software Development Kit
VS Code	Visual Studio Code
BLE	Bluetooth Low Energy
GPS	Global Positioning System
LiDAR	Light Detection and Ranging

A*	A-Star Algorithm (a graph traversal and pathfinding algorithm)
MQTT	Message Queuing Telemetry Transport (a lightweight messaging protocol for IoT)
ADC	Analog to Digital Converter
HTTP	Hypertext Transfer Protocol
USB	Universal Serial Bus
PWM	Pulse Width Modulation
RTOS	Real-Time Operating System
GPIO	General Purpose Input/Output

Chapter - 1

Introduction

1.1 Overview

Wheelchairs are vital mobility aids that have evolved from manual to powered versions, enhancing independence for individuals with physical disabilities. However, even powered models often demand constant manual input, posing challenges for users with limited motor control. This project introduces a smart two-wheeled wheelchair prototype that supports both manual and autonomous navigation through pre-coded paths controlled via a mobile app. By applying core robotic principles like motor control and programmable logic integrated with a mobile application it delivers an affordable, adaptable indoor navigation tool for people with special needs.

1.2 Problem Statement

Many individuals with physical disabilities rely on manual wheelchairs, which require either external assistance or significant upper-body strength. While powered wheelchairs exist, they are often expensive and complex. Furthermore, indoor navigation remains a challenge due to the absence of intuitive control systems. There is a need for a cost-effective, easy-to-use wheelchair system that can assist in navigating indoor spaces autonomously with minimal user intervention.

1.3 Objectives

- Design an autonomous mobile wheelchair using ESP32.
- Implement manual control using motor drivers.
- Integrate it with autonomous room-to-room navigation.
- Provide a user-friendly app interface for control.
- Additional features for the patient and the caretaker interface.

1.4 Motivation

The inspiration behind "Sarathi" stems from the real-world challenges faced by the elderly and physically challenged individuals in moving independently within their homes. Our aim was to create a supportive, affordable solution using basic electronics, smart control logic and implement an app. With rising interest in assistive technologies, we wanted to explore how hardware works, embedded systems and wireless communication could be combined to offer practical benefits to daily life.

1.5 Organization of the Report

- **Chapter 1 – Introduction:** Provides an overview of the project, its motivation, objectives, and scope.
- **Chapter 2 – Literature Survey:** Reviews existing smart wheelchair systems and research efforts, identifying current limitations and the need for improvement.
- **Chapter 3 – Problem Analysis and Design:** Discusses the challenges faced by traditional wheelchair users and outlines the proposed solution along with the system design approach.
- **Chapter 4 – Implementation:** Describes the development process, integration of hardware and software, and the realization of the smart wheelchair system.
- **Chapter 5 – Results:** Presents the outcomes of the implemented system, supported by data, graphs, and user observations.
- **Chapter 6 – Testing:** Details the testing methods used to evaluate system performance, reliability, and user experience.
- **Chapter 7 – Conclusion and Future Scope:** Summarizes the achievements of the project and explores possibilities for future enhancements and applications.

Chapter - 2

LITERATURE SURVEY

2.1 Introduction

The continuous advancement in assistive technologies has opened up new possibilities for enhancing the independence and quality of life for individuals with mobility challenges. Inspired by these developments, our project focuses on designing a smart wheelchair that integrates key modern technologies such as microcontrollers (ESP32), Bluetooth communication, and mobile application control.

Through an in-depth review of existing research and systems, we observed how conventional wheelchairs have evolved into intelligent platforms—offering features like remote control, obstacle avoidance, health monitoring, and smart connectivity. These insights shaped the direction of our work, allowing us to identify practical design approaches and common limitations. The literature survey thus not only reflects the progress in this field but also serves as the foundation upon which we've built our compact, affordable, and customizable smart wheelchair prototype aimed at improving user autonomy and safety.

2.2 Literature Survey

[1] ***"Revolutionizing Accessibility: Smart Wheelchair Robot and Mobile Application for Mobility, Assistance, and Home Management" by Ninura Jayasekera, Binali Kulathunge, Hirudika Premaratne, Insaf Nilam, Samantha Rajapaksha, Jenny Krishara (2024)***

This paper presents AssistEase, a wheelchair system that blends speech, gesture, and app-based controls to maximize user independence. Safety is prioritized with features like obstacle alerts and emergency notifications. The design is adaptable, offering both manual and smart controls. Trials show notable gains in user mobility and daily task efficiency. [1]

[2] ***"An Autonomous Wheelchair with Health Monitoring System Based on Internet of Things" by Lei Hou, Jawwad Latif, Pouyan Mehryar, Stephen Withers, Angelos Plastropoulos, Linlin Shen, Zulfqur Ali (2024)***

This study details a self-navigating wheelchair using IoT sensors to track health and movement in real time. AI-driven navigation ensures safe travel, while the system supports remote health monitoring and telemedicine. The interface keeps users and caregivers informed, enhancing autonomy for those with mobility challenges. [2]

[3] ***"Development of Apps Industry using Flutter: A Review" by Prabhav Shukla, Neha Tyagi, Deepanshu, Medhavi Agarwal, Shruti Jain (2024)***

This review examines Flutter's impact on app development, noting its single-codebase approach for Android, iOS, web, and desktop. Flutter's widget system and strong community speed up development and improve UI consistency. The toolkit's efficiency and scalability are highlighted as key industry benefits. [3]

[4] ***"IoT Based Wheelchair for Disabled Persons" by Manoj Prabu, Arun Kumar, Abuhasan (2025)***

This project outlines a smart wheelchair integrated with IoT components, including GPS, health sensors, and obstacle detection, while supporting both app and voice-based control. In addition to enabling mobility, the system focuses on environmental interaction through smart home integration and routine alerts. The multi-modal interface informed our decision to keep our control system modular and adaptable for users with different needs, even though our approach prioritizes affordability and reduced hardware complexity. [4]

[5] ***"The Impact of Flutter Development" by Sanket Radadiya, Sunny M. Ramchandani (2025)***

This paper explores Flutter's influence on software development, emphasizing its widget-based structure and fast performance. Its compatibility with major IDEs and support for multiple platforms have boosted productivity. Flutter's expanding role in web and AI applications is also discussed. [5]

[6] ***"Indoor Navigation System using BLE and ESP32" by Rakshith A, Navneeth V H, Dravya P S, Ujwal K Holla, K N Pushpalatha (2020)***

This study presents an indoor navigation solution using BLE for wireless communication and ESP32 for processing, with added biophysical sensors to track health metrics. The authors highlight BLE's low power consumption and suitability for short-range communication, which reinforced our choice of ESP32 for Bluetooth-based path control in indoor settings. While we excluded health monitoring, the paper validated our component selection for energy-efficient communication in structured environments. [6]

[7] *"Design and Construction of a Smart Wheelchair" by Deepak Kumara, Reetu Malhotra, S. R. Sharma (2020)*

This work discusses building a smart wheelchair with obstacle sensors and a Raspberry Pi for processing. It features user-friendly controls and emergency alerts. The design focuses on accessibility and responsive assistance for users.[7]

[8] *"Smart Wheelchair Control Using Arduino" by Maduguri Rani, Bodapatla Ramya, Kaveti Avinash, Gunturu Sainaveen, Jambuka Ruchitha (2024)*

This project explores a cost-effective wheelchair control system based on Arduino, supporting diverse input methods such as voice, gesture, Bluetooth, and eye-blink. The emphasis on multimodal control interfaces aligns with our project's goal of accessibility but contrasts with our simplified Bluetooth approach, chosen to reduce system overhead and ensure faster prototyping. [8]

[9] *"Internet of Things Hardware and Software" by Julia-Antonio, Dumitru-Marius, George Manuel (2020)*

This paper overviews IoT system architecture, highlighting the interplay between sensors, microcontrollers, communication protocols, and cloud software. It emphasizes how these elements enable real-time data collection and smart decision-making in connected environments. [9]

[10] *"Cross-Platform Innovation: The Rise and Impact of Flutter in Modern App Development" by Arjun Santhosh, Anu Tiji, Anurag T R, Naithan Thomas, Aswasthy Anup, Tintu Varghese (2024)*

This study analyzes Flutter’s impact on project structures and developer workflows. These benefits aligned with our goal to support both Android and iOS users from a single codebase. Flutter’s adoption is rising with changing trends in mobile and web development.[10]

[11] ***"Characterization and Performance Evaluation of ESP32 for Real Time Synchronized Sensor Networks" by M.J espinosa-Gavira, Agustin Aguera-Perez, J.C palomares-Salas (2024)***

The paper evaluates ESP32’s performance in real-time sensor networks, examining metrics like throughput, ADC precision, and synchronization accuracy. These findings confirmed ESP32’s suitability for time-sensitive applications, aligning with our needs for responsive motor control and reliable Bluetooth communication. Their analysis helped justify our microcontroller selection over simpler alternatives like Arduino Uno. [11]

[12] ***"Real-Time Remote-Controlled Wheelchair with Wi-Fi Integration and Obstacle Detection for the Disabled and Elderly People" by Navaneeth Bhaskar, Priyanka Tupe - Waghmare, Shrinidhi B M, Ovin Vinol Pereira, Mishaal Hussain, Shamya P Shetty (2024)***

This paper presents a Wi-Fi-controlled wheelchair with a mobile app and real-time obstacle detection. Sensors prevent collisions, and remote operation supports caregivers. The system aims to improve independence and safety for disabled and elderly users. [12]

[13] ***"A Bluetooth Controlled Real-time Wheelchair with Mobile Application for Physically Challenged People" by Sd. Muntaz Begum, Kadhirivellu Ganesh, Kalicheti Jithin Reddy (2023)***

This paper outlines a wheelchair prototype using HC-05 Bluetooth modules and Arduino UNO to provide real-time directional control through a mobile app. Its design served as a benchmark for our communication architecture, although we opted for ESP32 due to its integrated Bluetooth and enhanced processing capability. [13]

[14] ***"Electrical Analysis Using ESP-32 Module in Realtime" by Erik Wahyu Pratama, Agus Kiswantono (2022)***

This article compares ESP32 and Arduino Uno for real-time voltage and current measurement. It analyzes sensor reading errors to determine accuracy. The findings guide microcontroller selection for precise sensor applications. [14]

[15] ***"Development of a Modular Real-Time Shared-control System for a Smart Wheelchair" by Vaishanth Ramaraj, Atharva Paralikar, Eung Joo Lee, Syed Muhammad Anwar, Reza Monfaredi (2022)***

This paper introduces a modular navigation add-on for wheelchairs, enabling autonomous and shared control. A stereo camera supports path planning and obstacle avoidance, especially in tight spaces. The design is cost-effective and enhances mobility for disabled children and the elderly. [20]

2.3 Summary

In developing Sarathi, we drew inspiration from several impactful studies. The mobile app-based control system in AssistEase (Jayasekera et al.) emphasized the importance of user-friendly interfaces, directly influencing our implementation of Bluetooth-based app control. The integration of biophysical sensors and IoT by Lei Hou et al. validated the scalability of ESP32 for future health-monitoring features. The works of Shukla et al. and Radadiya et al. reinforced our decision to use Flutter for building a consistent cross-platform mobile interface. Concepts like obstacle detection and smart integration from Manoj Prabu et al. informed our use of an ESP32-L298N-based hardware setup. Additionally, Rakshith A et al.'s use of BLE and ESP32 for efficient indoor communication confirmed the ESP32’s relevance for low-power, real-time operations in our system. Design insights from Deepak Kumara et al. further influenced the mechanical layout of our compact, two-wheeled frame. Altogether, these studies shaped Sarathi’s design around accessibility, modularity, and cost-efficiency—empowering users with greater independence and control.

Chapter - 3

PROBLEM ANALYSIS AND DESIGN

3.1 Introduction

In developing this project, we began by carefully analyzing the real-life challenges faced by individuals who depend on wheelchairs for mobility. Many users struggle with limitations in traditional wheelchairs, such as the lack of independent control, difficulty in communicating with caregivers, and limited support in emergency situations. These issues can significantly affect not just mobility, but also the overall quality of life and sense of autonomy. To address these concerns, the problem was broken down into key areas such as ease of movement, safety, real-time communication, and daily assistance. Our goal was to design a system that empowers the user while also keeping caregivers connected and informed. The design phase focused on planning a smart, app-controlled wheelchair system that could combine multiple useful features in a user-friendly way. This includes intuitive navigation, alerts, and support functionalities.

This section introduces how we approached the problem logically, studied the needs of users, and began shaping a solution that is both practical and supportive, laying the groundwork for the detailed technical implementation discussed in later sections.

3.2 Problem Analysis / Proposed System

For individuals with limited motor skills or cognitive challenges, this can make independent movement difficult and increase their reliance on caregivers. Giving a wheelchair the ability to navigate on its own can make a huge difference. Autonomous navigation lets users move around more freely and safely, without always needing someone else’s help. This extra independence can improve quality of life and reduce the burden on family members and caregivers

3.3 Hardware Requirements

After evaluating several options, the system design adopted pre-coded pathways for indoor movement. Fixed room-to-room paths are mapped and triggered through a mobile app, with commands sent to the ESP32 via Bluetooth. This removes the need for real-time sensors or complex algorithms, making it ideal for structured indoor layouts.

The approach balances simplicity with functionality, offering a scalable, easy-to-debug, and cost-effective solution. The table below compares other methods and highlights why pre-coded pathways are best suited for this prototype.

Feature	**LiDAR**	**RFID**	**A* Algorithm**	**Pre-Coded Pathways**
Hardware Cost	Very High	Medium	Medium	Very Low
Implementation	Requires real-time 3D mapping	Requires tags and reader setup	Requires real-time maps	Simple and effective
Accuracy	Very-high	High	High	Moderate
Reliability Indoors	High	Medium, tags may fail	Depends on sensors	High
Ease of Debugging	Difficult	Moderate	Complex	Easy
Suitable for Prototypes	No	Maybe	No	Yes

Table 3.1 Selection of System

The selection of the microcontroller plays a vital role in determining the overall performance, efficiency, and scalability of the wheelchair system. After evaluating multiple options including the Arduino Uno and Arduino Nano, the ESP32 microcontroller was chosen due to its superior processing power (240 MHz), significantly larger memory capacity (4MB), and built-in Wi-Fi and Bluetooth connectivity.

Unlike the Arduino boards, which lack native wireless support and offer lower speed, the ESP32 enables seamless integration with mobile applications and IoT-based features. Its programmable power modes also contribute to efficient energy management, making it ideal for a battery-operated mobility device. Furthermore, its compact size and moderate cost make it suitable for embedded applications where space and budget are constraints. Following is the table drawn for comparison of the same. This was made to ensure proper understanding and knowledge of the components and assessing their pros and cons [21] [22][23].

Feature	**Arduino Uno**	**Arduino Nano**	**ESP32**
Processing Power	16MHz	16MHz	240MHz
Memory	32KB	32KB	4MB
Wi-Fi Connectivity	Not supported	Not supported	Built-in Wi-Fi
Power Efficiency	Average	Good	Programmable
Cost	High	Moderate	Moderate
Size	Large	Very Compact	Compact

Table 3.2 Selection of Microcontroller

The components selected for the smart wheelchair system were chosen for their cost-effectiveness, availability, and compatibility with embedded control applications. The ESP32's integrated wireless features [21], the L298N's ability to handle dual motor control [24], and the high energy density of 18650 batteries [25] made them ideal for a compact and mobile solution.

Component	**Features**	**Quantity**
L298N	Drives and controls the direction/speed of the two DC motors independently.	1
ESP32	Acts as the central controller; processes commands and communicates via Bluetooth.	1
DC Motors	Provides sufficient torque for moving the wheelchair; optimized for 6V-12V operation.	2
18650 Batteries	Rechargeable power source for the motors and controller; compact with the high energy density.	2

Table 3.3 Components Selected

Fig 3.1 ESP32 **Fig 3.2 L298N motor driver** **Fig 3.3 18650 batteries**

Fig 3.4 Geared DC Motor **Fig 3.5 Wheels**

3.4 Software Requirements

- Arduino IDE – Platform to write and upload the code for ESP32.
- Flutter SDK – Framework for building the app for both Android and iOS, also provides many libraries to work with.
- Android Studio / VS Code – IDEs used to develop, test, and debug the Flutter application.
- Dart Language – Programming language used to code the application in Flutter.
- Flutter was selected since it gave us:
  - Rapid Development with hot reload and a rich widget library.
  - Cross-platform Deployment, potentially enabling both Android and iOS builds in the future.

- ○ Easy integration with hardware communication protocols like Bluetooth.
- ○ Modern UI/UX support using Material Design.

Figure 3.6 Arduino IDE **Figure 3.7 Flutter** **Figure 3.8 Android Studio**

3.5 System Architecture Diagram

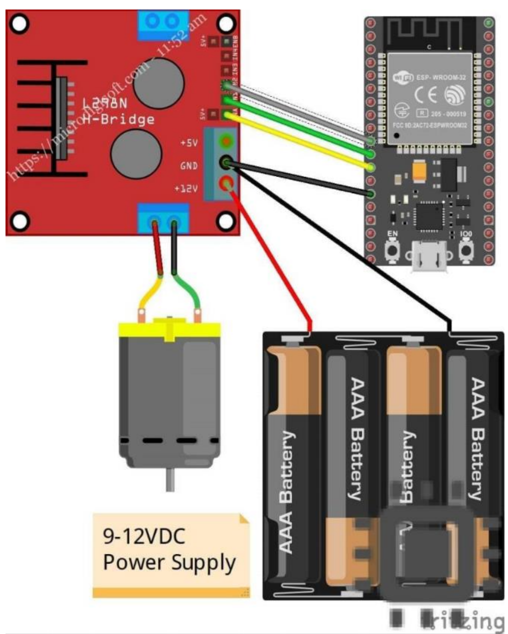


Figure 3.9 Circuit Diagram

The above diagram represents the foundational architecture used to control the motion of a smart wheelchair prototype. It integrates three essential hardware components: the ESP32 microcontroller, an L298N motor driver module, and a DC geared motor, all powered by a 9–12V

DC power source. This setup allows for efficient and responsive control of the wheelchair's movement based on Bluetooth commands sent from a mobile application.

The ESP32 is connected to the **control pins** of the L298N motor driver (IN1, IN2, and ENA) and communicates movement instructions such as forward, backward, left, right, and stop.The **L298N** is a dual H-bridge motor driver that serves as the intermediary between the ESP32 and the DC motor. It is responsible for **amplifying the control signals** received from the microcontroller and supplying the necessary current to drive the motor. The input pins (IN1 and IN2) define the **rotation direction** of the motor, while the **ENA (Enable A)** pin controls whether the motor is active. The driver also receives power directly from the external battery pack, which ensures that the motor receives sufficient voltage and current without overloading the ESP32. The batteries are the power source of the system and they provide each component with enough power to work with. The DC motors are the movement of the system and they guide the wheelchair to places.

3.6 Sequence Diagram

3.6.1 Flowchart for Manual Navigation

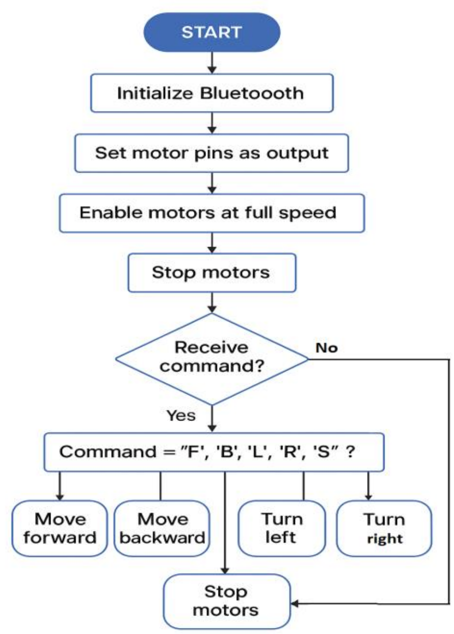


Figure 3.10 Flowchart of manual navigation

3.6.2 Algorithm

1. Start

1.1. Include required libraries (BluetoothSerial.h).

1.2. Define motor driver pins:

- IN1, IN2 (Left motor control)
- IN3, IN4 (Right motor control)
- ENA, ENB (Enable pins for speed control)

1.3. Create a BluetoothSerial object.

1.4. Define a String variable currentRoom and initialize it (e.g., "LIVING ROOM").

2. Setup Function

2.1. Start serial communication at baud rate 115200.

2.2. Begin Bluetooth with device name "ESP32-Car".

2.3. Set all motor pins (IN1–IN4, ENA, ENB) as output.

2.4. Enable motors by setting ENA and ENB to HIGH.

2.5. Call stopMotors() to ensure motors are off at startup.

3. Loop Function

3.1. **If Bluetooth data is available:**

3.1.1. Read the complete message string until newline (\n).

3.1.2. Trim spaces and convert messages to uppercase.

3.1.3. Print the received message for debugging.

3.1.4. **If message starts with "GOTO:"**

→ Extract target room from message

→ Call goToRoom(targetRoom)

3.1.5. **Else**

→ Extract the first character

→ Use switch to call:

- F → moveForward()
- B → moveBackward()
- L → turnLeft()
- R → turnRight()
- U → turnAround()
- S → stopMotors()
- Unknown → stopMotors()

4. goToRoom(targetRoom)

4.1. If targetRoom equals currentRoom, return (no action needed).

4.2. Use a series of if-else conditions to define route from currentRoom to targetRoom.

4.3. For each route:

- Call movement functions in order (e.g., moveForward(), turnLeft(), etc.)
- Use delay() to control movement durations.

4.4. After movement:

- Call stopMotors()
- Update currentRoom = targetRoom

5. Motor Control Functions

- moveForward() – Left and Right motors move forward
- moveBackward() – Both motors in reverse
- turnLeft() – Left motor backward, Right motor forward, then forward
- turnRight() – Left motor forward, Right motor backward, then forward
- turnAround() – Rotate 180°, then stop
- stopMotors() – All motor pins set to LOW

6. End Loop and Repeat

Go back to step 3 and wait for the next Bluetooth command.

3.7 Summary

This chapter presented a structured analysis of the problem faced by individuals with limited mobility and outlined the proposed solution in the form of a smart wheelchair system. It began by identifying the key challenges, such as limited autonomous control, lack of emergency response mechanisms, and absence of caregiver communication in traditional wheelchair setups. Based on this analysis, a system was designed to address these gaps through a mobile app–controlled wheelchair integrated with smart features like directional navigation, SOS alerts, and routine tracking. The design phase focused on developing a user-friendly, practical, and efficient solution that enhances both patient independence and caregiver support.

Chapter - 4

IMPLEMENTATION

4.1 Overview of System Implementation

The implementation of the Sarathi system integrates both hardware and software components to provide enhanced mobility, real-time monitoring, and remote assistance for individuals with limited mobility. The project aims to enable patients to control their wheelchair using a mobile application and allow guardians to receive emergency alerts, and communicate directly through the app.

4.1.1 Hardware Module (ESP32 Microcontroller Based)

- **Microcontroller**: ESP32 is used for its built-in Bluetooth capabilities, low power consumption, and versatility.
- **Motor Driver**: L298N Dual H-Bridge Motor Driver is used to control the DC motors based on directional commands.
- **Motors**: Geared DC motors are used to drive the wheelchair wheels.
- **Power Supply**: 18650 lithium-ion batteries are used to power the ESP32 and motors.
- The ESP32 receives movement commands (Forward, Backward, Left, Right) over Bluetooth from the mobile application and drives the motors accordingly.

4.1.2 Software Module (Flutter Application)

- Developed using the Flutter framework for cross-platform compatibility and a responsive UI.
- The Saarthi Smart Wheelchair App is a thoughtfully designed mobile application built using Flutter, tailored specifically for individuals with limited mobility and their caregivers. It forms the core control and communication interface between a Bluetooth-enabled ESP32-powered wheelchair and its users. The app not only facilitates wheelchair movement but also enhances patient care through personalized features like medicine reminders, profile management, and real-time emergency alerts.

- Designed with inclusivity in mind, Saarthi is split into two separate usage roles: Patient and Guardian, each with its unique functionality and interface, enabling independent use by patients and attentive monitoring by caregivers.

On launching the app, users are prompted to select their role – either as a Patient or a Guardian. Based on this selection:

- The Patient is asked to enter personal details such as name, age, gender, phone number, and address.
- The Guardian enters details like name, age, phone number, and address.

All this information is securely stored in Firebase Firestore, ensuring data persistence across sessions and devices.

The patient screen is designed to provide direct, accessible control of the wheelchair via Bluetooth. Commands are transmitted to the ESP32 controller in real-time using classic Bluetooth communication through the flutter_bluetooth_serial package.

The patient can also send semantic commands like "GOTO:BEDROOM", which the ESP32 interprets and translates into a series of movements to reach the desired destination. This abstraction from manual control to room-based navigation enhances usability for individuals with limited dexterity.

A prominent SOS button and a personal in-app chat feature on the patient screen allows immediate alert to the guardian during emergencies. This triggers real-time alerts, ensuring rapid response and improving safety.

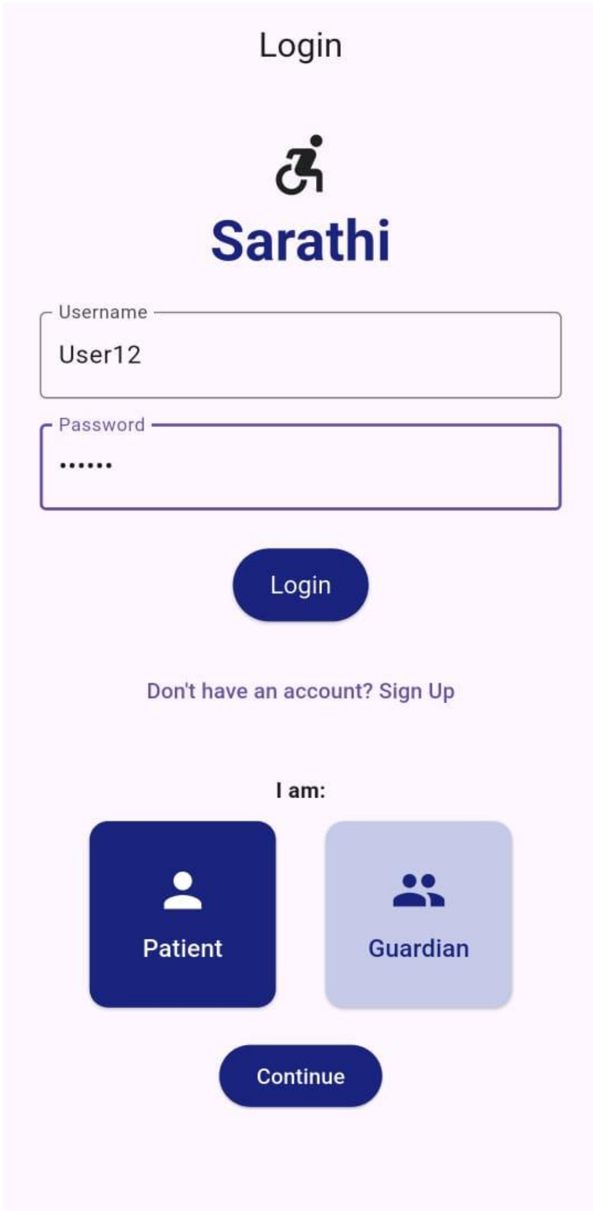


Figure 4.1 Login Page

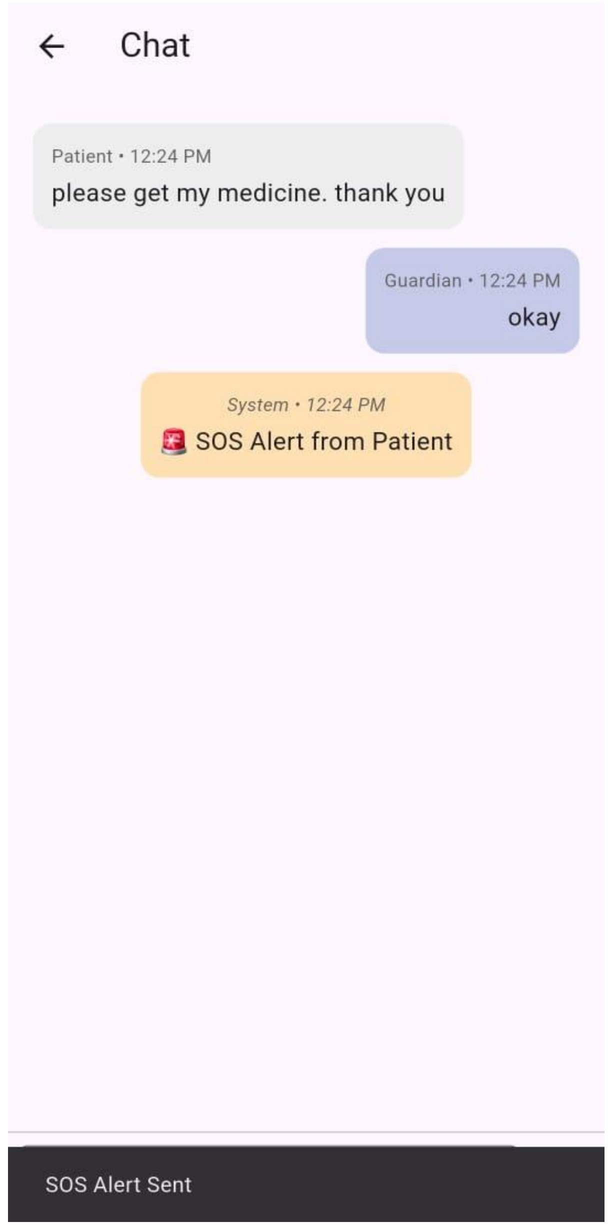
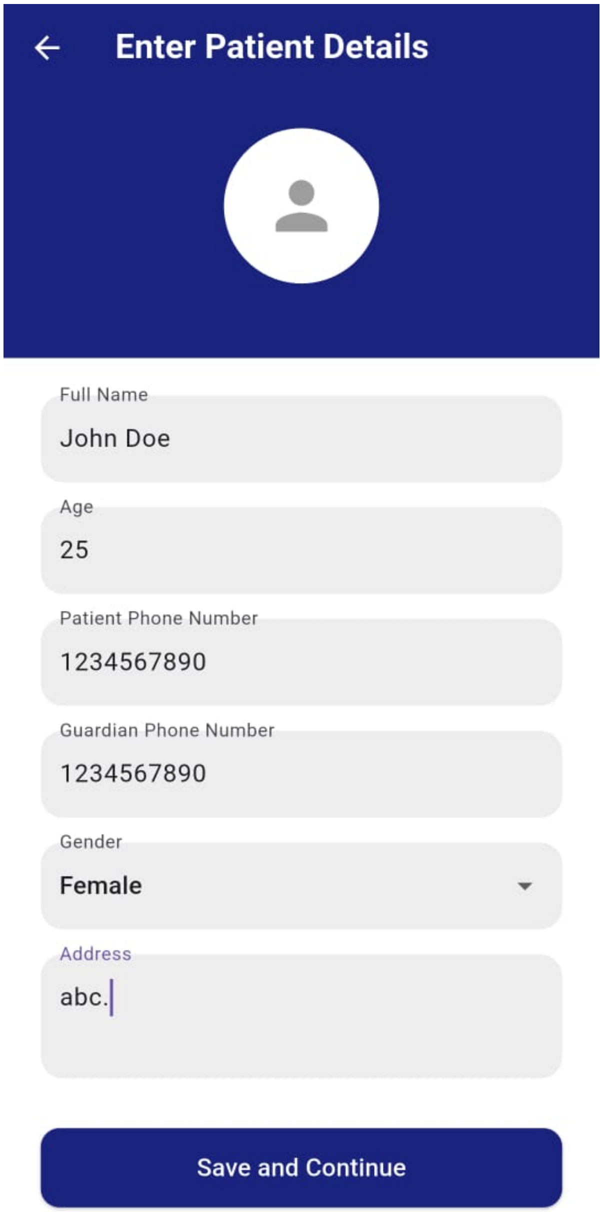


Figure 4.2 Patient - Guardian Chat



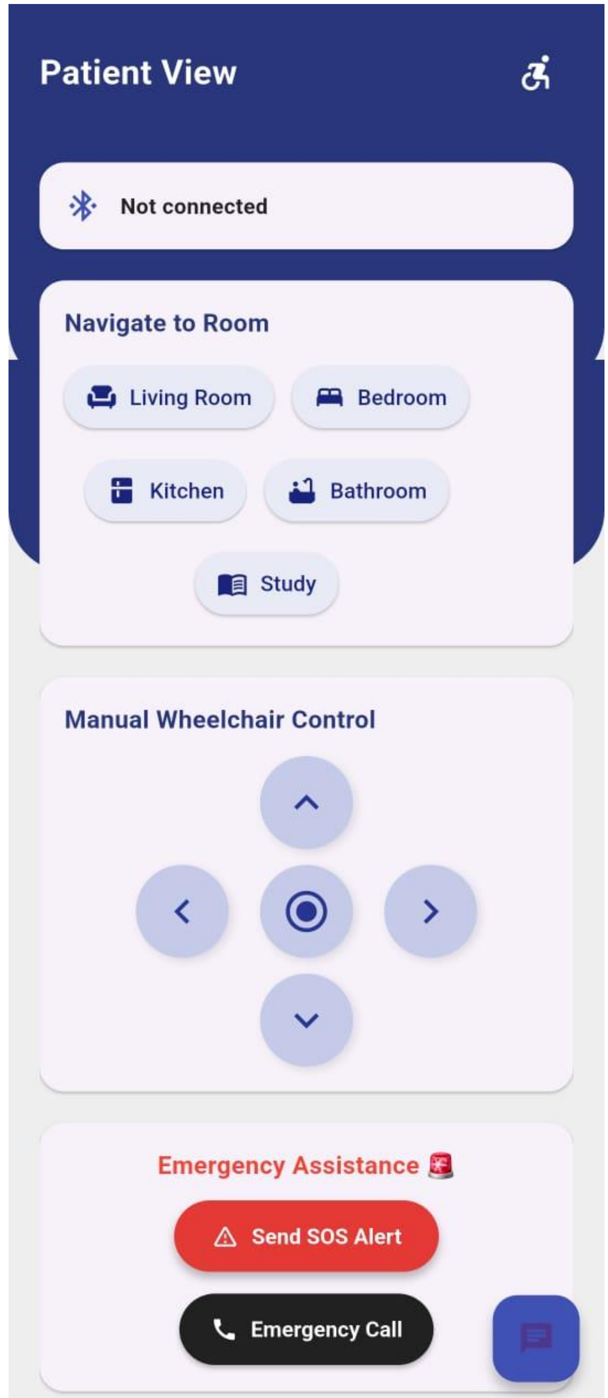


Figure 4.3 Patient Details

Figure 4.4 Patient Interface

4.2 Code Snippets

The following section contains the codes written to achieve the hardware and software implementation and their integration with each other. The software used to do it are: Arduino IDE, VS Code and the languages used are : C++ , Dart, etc.

4.2.1 Arduino IDE

The following code was written for the manual navigation of the wheelchair which was later uploaded to the ESP32. It uses the basic logic of the speed of the motors being regulated for each kind of movement done by it. The commands are a single letter like, ‘F’, ‘B’, ‘R’, ‘L’, etc. Left and right turns are executed in place instead of along a curve. The speed of the motors can be changed according to the requirement.

```
void loop() {
  if (SerialBT.available()) {
    char command = SerialBT.read();
    command = toupper(command);
    if (!isAlpha(command)) return;

    Serial.print("Received: ");
    Serial.println(command);

    switch (command) {
      case 'F': moveForward(); break;
      case 'B': moveBackward(); break;
      case 'L': turnLeft(); break;
      case 'R': turnRight(); break;
      case 'S': stopMotors(); break;
      case 'U': turnAround(); break;
      default: stopMotors(); break;
    }
  }
}

// Movement functions
void moveForward() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
```

Figure 4.5 Manual Navigation Arduino code

This Arduino sketch enables an ESP32-controlled smart wheelchair (or robot car) to navigate between rooms using Bluetooth commands. It introduces semantic navigation—users can simply send room names (like "GOTO:BEDROOM") instead of manual movement commands.Depending on the source and destination room, the goToRoom() function executes a specific sequence of moves and turns using delay() for fixed-duration movement.

```
// Room-to-room navigation logic
void goToRoom(String targetRoom) {
  if (targetRoom == currentRoom) return;

  // LIVING ROOM routes
  if (currentRoom == "LIVING ROOM" && targetRoom == "BEDROOM") {
    moveForward(); delay(2000);
    turnLeft(); delay(400);
    moveForward(); delay(2000);
  } else if (currentRoom == "LIVING ROOM" &&
                 targetRoom == "KITCHEN") {
    moveForward(); delay(3000);
  } else if (currentRoom == "LIVING ROOM" &&
                 targetRoom == "STUDY ROOM") {
    moveForward(); delay(1500);
    turnLeft(); delay(400);
    moveForward(); delay(2500);
  }
```

Figure 4.6 Room-to-Room Navigation code

4.2.2 Flutter Application

The following snippets showcase the Flutter import structure and widget initialization for the *Saarthi Smart Wheelchair App*, reflecting the project's modular and well-organized coding approach. The imports include essential Flutter packages such as material.dart, Bluetooth functionality (flutter_bluetooth_serial), permissions management (permission_handler), and UI navigation through separate screens like patient_screen.dart, guardian_screen.dart, and authentication pages. This modular file structure helps keep the codebase maintainable and scalable by separating concerns across different functional screens.

```dart
import 'package:flutter/material.dart';
import 'package:permission_handler/permission_handler.dart';

// Screens
import 'screens/home_screen.dart';
import 'screens/patient_screen.dart';
import 'screens/guardian_screen.dart';
import 'auth/login_page.dart';
import 'auth/signup_page.dart';
import 'screens/patient_details_page.dart';
import 'screens/guardian_details_page.dart';
```

Figure 4.7 The main( ) function in Flutter

The part shows the implementation of the PatientScreen, which extends StatefulWidget to manage dynamic Bluetooth interactions and UI updates. This screen plays a critical role in enabling real-time wheelchair control by responding to user input and sending commands to the ESP32 via Bluetooth. The use of packages like url_launcher and typed_data also hints at advanced functionalities like launching external apps or handling byte-level data. Together, these imports and widget structures helped in building a responsive, permission-aware, and hardware-integrated Flutter application tailored for accessibility and assistive mobility.

```dart
import 'package:flutter/material.dart';
import 'package:flutter_bluetooth_serial/flutter_bluetooth_serial.dart';
import 'chat.dart';
import 'package:permission_handler/permission_handler.dart';
import 'dart:typed_data';
import 'package:url_launcher/url_launcher.dart';

class PatientScreen extends StatefulWidget {
  const PatientScreen({super.key});

  @override
  State<PatientScreen> createState() => _PatientScreenState();
}
```

Figure 4.8 A snippet from patient_screen.dart

Chapter - 5

RESULTS

5.1 Introduction

This chapter presents the results obtained after implementing and testing the proposed Smart Wheelchair system. The aim was to evaluate how well the system meets the objectives outlined during the design phase, such as smooth mobility control, real-time communication, emergency responsiveness, and daily assistance features. The results discussed in this section include both hardware and software performance, app functionality, and user interaction. Functional testing was conducted to verify the responsiveness of the mobile controls, the reliability of communication between the app and hardware, and the usability of features like the SOS alert and chat system.

5.2 Description of the Results

After successfully developing the mobile application, the next phase involved integrating it with the physical wheelchair hardware using an ESP32 microcontroller. The ESP32 was programmed to receive Bluetooth signals from the Flutter app and translate them into motor control commands via the L298N motor driver. Through extensive testing, it was confirmed that the wheelchair could accurately interpret and respond to manual movement commands (such as forward, backward, and stop) as well as autonomous navigation instructions corresponding to predefined room paths. The system exhibited quick response times and stable Bluetooth connectivity, ensuring that user inputs from the app were executed in near real-time by the motors. Transitions between commands were smooth, and the SOS functionality triggered reliable alerts on the guardian interface. These results validated the effectiveness of the overall system, demonstrating that the Flutter-ESP32 integration could deliver a functional, low-cost, and responsive smart wheelchair suitable for indoor use. The prototype behaved reliably under various test conditions, confirming its potential as a viable assistive solution for individuals requiring enhanced indoor mobility.

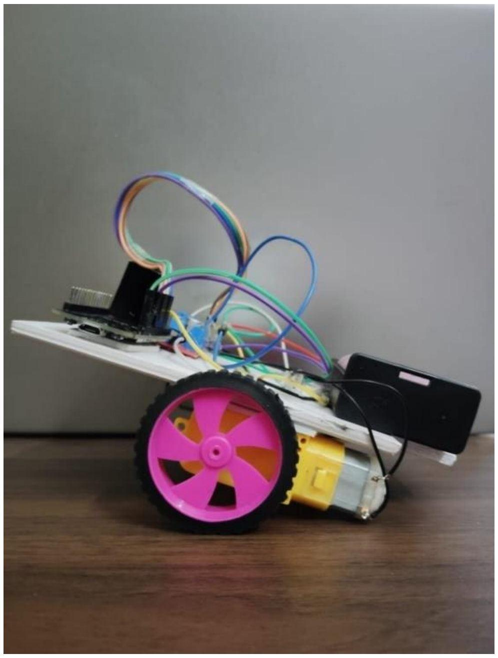


Figure 5.1 : Side view of Sarathi

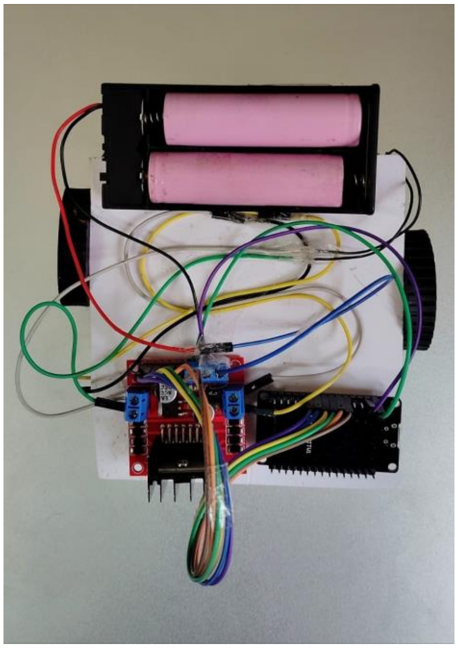


Figure 5.2 : Top view of Sarathi

5.3 Summary

This chapter presented the results obtained from testing the Smart Wheelchair system 'Sarathi'. Through performance analysis, user feedback, and functional testing, the system was shown to perform reliably across all key features—including movement control, SOS alerts, and real-time communication. Visual tools such as Wokwi(1) helped illustrate the effectiveness of each component. Overall, the system successfully met its design objectives, demonstrating a high level of usability, responsiveness, and integration between hardware and the mobile application. The positive outcomes from testing indicate that the Smart Wheelchair has the potential to improve mobility and safety for users while offering peace of mind to caregivers.

Chapter - 6

TESTING

Testing is a critical phase in the development of the Sarathi smart wheelchair system, as it validates the functional correctness, reliability, and responsiveness of both the hardware and the mobile application. The testing process was divided into three categories: motor response and Bluetooth communication, Flutter application interface, and end-to-end performance.

6.1 Objectives of Testing

- Ensure accurate command transmission from the Flutter application to the ESP32 microcontroller via Bluetooth.
- Verify the correct movement of the wheelchair in manual (F, B, L, R, S, U) and autonomous (GOTO:<ROOM>) modes.
- Test responsiveness and consistency of chat, SOS alerts, and medicine reminders
- Validate Firebase Firestore data integrity and real-time updates.

6.2 Test Cases

Test Case ID	Test Description	Input/Action	Expected Output	Result
TC-01	Manual Forward Movement	Send “F” via app	Both motors activate and move wheelchair forward	✓ Pass
TC-02	Room Navigation: LIVING ROOM → BEDROOM	Send GOTO:BEDROOM	Executes programmed path and stops in BEDROOM	✓ Pass
TC-03	Turn Left in Place	Send “L”	Left motor reverses, right motor forward, turns left	✓ Pass
TC-04	SOS Alert Functionality	Press SOS button	Guardian receives alert in Firestore and notification	✓ Pass

TC-05	Chat Messaging Sync	Patient sends message	Guardian sees message update in real-time	✓ Pass
TC-06	Add Medicine Reminder	Add reminder in app	Local notification triggered at selected time	✓ Pass
TC-07	Stop Motors	Send “S”	All motor pins LOW, wheelchair halts immediately	✓ Pass
TC-08	GOTO command error handling	Send GOTO:UNKNOWN	System does not move, prints fallback debug message	✓ Pass

Table 6.1 Test Case Results

6.3 Observations

- Latency: Command response time via Bluetooth was under 200 ms on average.
- Accuracy: Autonomous navigation reliably reached correct rooms when tested in mapped conditions.
- Stability: App did not crash during extended use; Firebase sync remained stable across sessions.
- Limitations: No obstacle detection implemented, so physical obstructions can affect path accuracy.

6.4 Summary

All key functionalities of the SARATHI wheelchair were successfully tested. Both manual control and room-based navigation worked as intended. App functionalities such as SOS alerts, chat, and medicine reminders performed reliably. These results confirm the system’s suitability for use in indoor environments, particularly where cost-effective and intuitive control is essential for user independence.

Chapter - 7
CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

Our ‘Sarathi’ project was developed with a clear goal: to empower individuals with mobility challenges by offering them greater control over their movement and daily routines. Traditional wheelchairs often limit a user's ability to act independently, making them reliant on constant caregiver assistance. This project addresses that gap by introducing an app-controlled system that brings mobility, safety, and support into the user's own hands. By integrating features like directional navigation, emergency alerts, and a communication interface, the system gives patients the ability to move freely, reach out for help when needed, and stay engaged in their daily activities without depending entirely on others. This enhanced level of autonomy not only improves practical functionality but also contributes significantly to the user’s emotional well-being. Knowing that they can safely navigate, send an alert, and stay connected with their caregiver boosts the user’s confidence and sense of security. It encourages a more independent lifestyle while ensuring that help is always within reach. The successful implementation and positive testing outcomes confirm that this smart solution can truly improve the quality of life for people who rely on mobility aids.

7.2 Future Scope

Despite the essential role that wheelchairs play in the lives of individuals with mobility challenges, their design has seen relatively limited innovation over the decades. Most conventional wheelchairs, even electric ones, remain basic in function—focusing solely on movement without addressing the user’s broader needs for communication, safety, or daily support. This gap highlights the urgent need to rethink and redesign the wheelchair as more than just a mobility aid. The Smart Wheelchair developed in this project marks a step in that direction by introducing app-based control, emergency alerts, and communication features. However, this is only the beginning. There is significant room for further development to truly transform the wheelchair into an intelligent, responsive, and health-aware companion.

Building on this foundation, future enhancements could include:

1. **Multimodal Control:** Implement voice- or eye-gaze interfaces (using packages like speech_to_text or WebGazer.js bindings) for users with minimal motor function.
2. **Dynamic Obstacle Avoidance:** Integrate low-cost ultrasonic or LiDAR sensors and run simple A* or potential-field algorithms on the ESP32 for real-time path correction.
3. **Health Monitoring:** Add pulse oximetry and posture detection using analog sensors, reporting vitals via MQTT to Firebase or a private dashboard.
4. **Geofencing & GPS:** Use a BLE or GPS module to define safe zones within a home; trigger notifications if the user approaches restricted areas.
5. **Smart-Home Integration:** Expose an HTTP or MQTT API on the ESP32 to control IoT devices (lights, doors, thermostats) through the wheelchair app, enabling context-aware environmental adjustments.

These additions can turn the wheelchair from a passive device into a proactive assistant—helping users not just move, but live with greater independence, confidence, and dignity. As technology continues to advance, so must the assistive tools we rely on, and this project lays a strong foundation for that necessary evolution.

REFERENCES

[1] Revolutionizing Accessibility: Smart Wheelchair Robot and Mobile Application for Mobility, Assistance, and Home Management *by Ninura Jayasekera, Binali Kulathunge, Hirudika Premaratne, Insaf Nilam, Samantha Rajapaksha, Jenny Krishara*

[2] An Autonomous Wheelchair with Health Monitoring System Based on Internet of Things *by Lei Hou, Jawwad Latif, Pouyan Mehryar, Stephen Withers, Angelos Plastropoulos, Linlin Shen, Zulfqur Ali*

[3] Development of Apps Industry using Flutter: A Review *by Prabhav Shukla, Neha Tyagi, Deepanshu, Medhavi Agarwal, Shruti Jain*

[4] IoT Based Wheelchair for Disabled Persons *by Manoj Prabu, Arun Kumar, Abuhasan*

[5] The Impact of Flutter Development *by Sanket Radadiya, Sunny M. Ramchandani*

[6] Indoor Navigation System using BLE and ESP32 *by Rakshith A, Navneeth V H, Dravya P S, Ujwal K Holla, K N Pushpalatha*

[7] Design and Construction of a Smart Wheelchair *by Deepak Kumara, Reetu Malhotra, S. R. Sharma*

[8] Smart Wheelchair Control Using Arduino *by Maduguri Rani, Bodapatla Ramya, Kaveti Avinash, Gunturu Sainaveen, Jambuka Ruchitha*

[9] Internet of Things Hardware and Software *by Julia-Antonio, Dumitru-Marius, George Manuel*

[10] Cross-Platform Innovation: The Rise and Impact of Flutter in Modern App Development *by Arjun Santhosh, Anu Tiji, Anurag T R, Naithan Thomas, Aswasthy Anup, Tintu Varghese*

[11] Characterization and Performance Evaluation of ESP32 for Real Time Synchronized Sensor Networks *by M.J espinosa-Gavira, Agustin Aguera-Perez, J.C palomares-Salas*

[12] Real-Time Remote-Controlled Wheelchair with Wi-Fi Integration and Obstacle Detection for the Disabled and Elderly People *by Navaneeth Bhaskar, Priyanka Tupe - Waghmare, Shrinidhi B M, Ovin Vinol Pereira, Mishaal Hussain, Shamya P Shetty*

[13] A Bluetooth Controlled Real-time Wheelchair with Mobile Application for Physically Challenged People *by Sd. Muntaz Begum, Kadhirivellu Ganesh, Kalicheti Jithin Reddy*

[14] Electrical Analysis Using ESP-32 Module in Realtime *by Erik Wahyu Pratama*

[15] Development of a Modular Real-Time Shared-control System for a Smart Wheelchair *by Vaishanth Ramaraj, Atharva Paralikar, Eung Joo Lee, Syed Muhammad Anwar, Reza Monfaredi*

[16] https://www.ijresm.com/Vol.3_2020/Vol3_Iss4_April20/IJRESM_V3_I4_120.pdf

[17] https://www.e3s-conferences.org/articles/e3sconf/pdf/2024/77/e3sconf_icsget2024_01016.pdf

[18] https://mesopotamian.press/journals/index.php/BJIoT/article/view/629

[19] https://scispace.com/papers/revolutionizing-accessibility-smart-wheelchair-robot-and-3paaonm64i

[20] https://www.ivysci.com/articles/3942060__Revolutionizing_Accessibility_Smart_Wheelchair_Robot_and_Mobile_Application_for_Mobility_Assistance_

[21] https://www.jdaic-journal.org/index.php/about/article/download/37/24

[22] https://journal.esrgroups.org/jes/article/view/1537

[23] https://www.nature.com/articles/s41598-025-00716-w

[24] https://thefuturalabs.com/flutter-app-development-trends-in-2024/

[25] Espressif Systems. (2022). "ESP32 Technical Reference Manual."

[26] Arduino Documentation, "Arduino Uno Rev3," Datasheet and Specs

[27] Arduino Nano Product Page. https://store.arduino.cc/products/arduino-nano

[28] TutorialsPoint: "DC Motor with L298N Motor Driver,"

[29] Battery University. "18650 Battery Basics,"

[30] Wokwi - World's most advanced ESP32 Simulator

[31] https://www.tinkercad.com/

[32] Software | Arduino

[33] Flutter - Build apps for any screen

[34] ESP32 Bluetooth Controlled Car : 11 Steps (with Pictures) - Instructables

[35] ESP32-Robot-TB6612FNG/ESP32_2wheel_robot_tb6612.ino at main ·

[36] mkconer/ESP32-Robot-TB6612FNG

[37] https://firebase.studio/