ABSTRACT

GreetRobot is a breakthrough robotics project engineered to redefine human-robot interaction. 5 feet tall, it combines an imposing physical presence with a central screen used both as a control interface and for perfect communication. This innovative creation deploys a state-of-the-art face detection system to detect and greet people with uncanny accuracy. This feature is complemented by an LED board embedded within its face, able to express a range of emotions and therefore imbue its interactions with a sense of personality.

Equipped with wheels and Bluetooth, it can be easily moved and controlled from another device, making it able to be fun, a kind assistant, or even an educational tool. GreetRobot offers a multifaceted platform for exploration and engagement. With its fusion of advanced technology and intuitive design, GreetRobot signifies a quantum leap in the evolution of robotics, promising to captivate and inspire users across diverse domains, from entertainment to customer service and beyond.

List of Figures

TABLE OF CONTENTS

Contents **Page no**

1. INTRODUCTION

1.1 ABOUT THE PROJECT

GreetRobot is a visionary approach towards human-robot interaction, aiming to go beyond boundaries and redefine technological engagement. Physically, it stands tall at 5 feet, commanding attention, with the central screen as a gateway to seamless communication and control. At its core, GreetRobot is powered by an advanced face detection system that enables it to recognize individuals with a lot of accuracy and extend personalized greetings. What makes GreetRobot very special, though, is an expressive LED board nestled inside the face, which animates its interactions with a rich tapestry of emotions that resonate at an altogether deeper level with users.

At the core of the GreetRobot design philosophy is versatility and accessibility. Mounted on wheels and powered by Bluetooth connectivity, it transcends static robotics with fluid mobility and remote control capabilities. Whether it is floating through crowds with grace or engaging in playful interactions with children, GreetRobot adapts effortlessly to the plethora of environments and situations it might find itself in. But beneath the technological prowess lies a deeper purpose—the enrichment of lives through innovation and connection. GreetRobot is not a machine; it is a catalyst for forging meaningful relationships and wonder in all who come into contact with it.

As GreetRobot goes into unexplored territory, it carries with it a deep mission: to bridge the gap between humanity and technology, fostering along the way empathy, understanding, and curiosity. It invites us to reconsider the possibilities of human-robot coexistence, nudging us toward embracing a future where innovation and compassion intermingle. In a world where the boundaries between the digital and physical realms blur, GreetRobot stands as a beacon of hope and possibility, beckoning us into a journey of discovery and light.

1.2 OBJECTIVE

The purpose of the GreetRobot project is multifaceted and holistically approaches the advancement of human-robot interaction and technological innovation. At its core, the project seeks to design a robotic platform that integrates well into human environments through advanced capabilities and intuitive design.

First and foremost, the objective is to design a robot that can recognize and greet a person with precision and warmth. Using modern face detection technologies, GreetRobot seeks to create real connections with users by instilling familiarity and rapport in all contexts.

In addition, the project seeks to provide GreetRobot with personality, realized by an expressive LED board embedded on the robot's face. This feature enables the robot to create a wide range of emotions to increase the quality and depth of its interactions with users.

The GreetRobot is also designed to be versatile and agile, capable of navigating diverse environments and performing a variety of tasks with ease. With wheels and Bluetooth connectivity, the project seeks to empower users to remote-control and maneuver the robot for a variety of applications in diverse settings.

Most fundamentally, the goal of the GreetRobot project is to further robotics and human-robot interaction, opening possibilities for applications of the project in domains as varied as customer service, education, entertainment, and more. Blending advanced technology with user-centered design principles, the project aspires to create a game-changing platform that enhances the way we interact with and view robots in our lives.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system of the GreetRobot project gives roots to its evolution and refinement, providing valuable insights regarding current capabilities and areas for improvement. In its very core, the existing system consists of several key components that, together, allow the robot to interact with users and navigate through its environment.

First and foremost, the existing system accommodates a basic framework for physical construction, consisting of a chassis, a central screen, an LED board, and mobility features, such as wheels. This framework provides the structural integrity and mobility the robot needs to function effectively in a variety of settings.

Moreover, the existing system contains rudimentary software components that are in charge of such basic functionalities as face detection and limited communication capabilities. While such software enables the robot to undertake simple activities like face recognition and the display of some preprogrammed expressions, it lacks the intricacy and flexibility that would allow it to go beyond such basic activities.

Finally, the existing system might include basic hardware components such as cameras, sensors, and microcontrollers that provide the underlying functionality for features such as face detection and mobility. However, these components might be limited in their capabilities and should be upgraded or modified in order to fully achieve the potential of the GreetRobot project.

Generally, while the existing system of the GreetRobot project presents a promising starting point, it also identifies opportunities for betterment and extension. By building upon existing capability and addressing limitations, the project endeavors to create a more robust and sophisticated robotic platform that will be capable of delivering enhanced user experiences and interactions.

2.2 PROPOSED SYSTEM

The proposed system for the GreetRobot project introduces several significant enhancements to improve its functionality and user interaction. The basic architecture is maintained, though improved in crucial areas such as face detection, communication, expression display, mobility, and hardware architecture.

In regard to this, one of the prime enhancements in the face detection system of the GreetRobot is the integration of state-of-the-art algorithms and machine learning techniques for greater accuracy and reliability. This means a better improvement in the identification of individuals to give more personalized interactions and greetings from the GreetRobot.

Moreover, the proposed system is improving the communication interface by adding features of natural language processing and real-time chatbot integration. This will make it easier to use for users to have deeper conversations with GreetRobot, returning intelligent responses based on inquires and needs.

The LED board embedded in the face of GreetRobot is being improved for a wider range of emotions and expressions. Context and user interaction generate these expressions dynamically, whereby the robot is in a position to convey empathy and allow for engagement of users on a deeper emotional level.

Furthermore, the proposed system shows an interest in mobility features: upgraded wheels for easier navigation and the ability to detect and evade obstacles. Enhanced Bluetooth connectivity is integrated for more robust remote control functionalities, providing users with more precision and ease in the maneuvering of the robot.

Advantages of the proposed scheme

Several advantages over the existing system are promised by the proposed system for GreetRobot, enhancing the functionality, versatility, and user experience of the robot.

For one, the improved face detection system ensures greater accuracy and reliability in recognizing the person, thus leading to more personalized interactions and greetings. This helps to increase user engagement and can make a much stronger connection between GreetRobot and its users.

Equally important, the improved communication interface, equipped with natural language processing capabilities and real-time chatbot integration, enables more meaningful and interactive conversations. Users can receive intelligent responses tailored to their inquiries and needs, hence increasing the overall quality of interaction with the robot.

The dynamic expression display with a wide range of emotions and expressions adds to the depth and realism of the interactions of GreetRobot. In so doing, the robot is able to establish stronger bonds and leave a lasting impression on users by being empathetic and responding to users at an emotional level.

The improved mobility features, including upgraded wheels and obstacle avoidance, make the navigational prowess of GreetRobot better and smoother, thus enabling the robot to move around in various environments with much more precision and agility. This would ensure a smoother interaction and better integration into different settings.

Furthermore, the improved Bluetooth connectivity provides better functionalities with remote control, hence giving greater flexibility and ease to users in controlling the robot's movements. Whether navigation across crowded spaces or performance of a particular task, users can rely on the better mobility features of GreetRobot.

2.3 FEASIBILITY STUDY

The feasibility study of the GreetRobot project does a comprehensive analysis of its technical, economic, and operational viability. On the technical side, advanced face detection algorithms and dynamic expression displays proposed in the system are assessed in detail to ensure flawless integration and functioning. Evaluations also extend to the availability of hardware and feasibility toward implementing upgrades—such as improved mobility features—for the optimization of the performance of GreetRobot. Economic feasibility analysis looks at cost factors, including the expense for hardware procurement, software development, and maintenance costs that will continue to accrue. Potential revenue streams and long-term financial feasibility are examined to give stakeholders a sense of the project's financial outlook.

The focus of operational feasibility investigations will be on the practicality of deploying and managing GreetRobot in real-world scenarios. User acceptance, ease of use, and integration into the existing infrastructure form key points of this assessment. Moreover, possible risks and challenges during deployment are identified, and plans for mitigation are developed to guarantee a problem-free implementation. By addressing technical, economic, and operational aspects, the feasibility study provides a necessary roadmap that guides stakeholders toward making the right decisions and maximizing the chances of the project's success.

2.3.1 Details

It is through multi-dimensional inquiry that the complex details of the GreetRobot project will be examined and determined as viable and potentially successful. The technical feasibility consists of the in-depth analysis of the proposed system components, such as face detection algorithms, communication interfaces, and mobility features. This assessment evaluates the compatibility of these components, integration into the existing framework, and feasibility to implement upgrades to scale up the GreetRobot capabilities. Technical feasibility also deals with the availability of the hardware and software resources needed for the project and the deliverability of the project with the required technological setup.

Economic feasibility deals with the financial aspects of the project, which will cover cost and revenue. It includes detailed cost estimates, such as hardware procurement costs, software development costs, research and development costs, and maintenance costs over the project's life cycle. Revenue projections will be worked out, factoring in market demand, pricing, and scalability. The analysis will present the project's stakeholders with a total understanding of the financial implications of the project, which will enable informed decision-making regarding investment and resources. Through the careful consideration of technical and economic factors, the feasibility study aims at laying a solid foundation for the successful implementation and sustainability of the GreetRobot project.

2.3.2 Impact on Environment

The GreetRobot project gives top priority to the sustainability of the environment through considered design and operation practices. The robot system makes use of recyclable and durable materials, energy-efficient parts such as low-power processors and LED displays, to limit energy consumption and eventual damage to the environment. The GreetRobot's modular design enables easy upgrades and maintenance that reduce the electronic waste to a great deal, enabling the replacement of individual components instead of the whole unit.

The operation of GreetRobot is designed on energy efficiency. There are power management schemes that have been optimally designed to minimize energy consumption during active and idle states. This ensures that the effective operation of the robot is assured, while keeping the environmental impact low. Besides, since digital information is provided on customer service and education, the robot will reduce dependence on printed materials, thus contributing to the conservation of resources.

In general, the project GreetRobot presents commitment to environmental stewardship. It seeks to deliver innovative and effective solutions while ensuring responsible growth in robotic technology. GreetRobot seeks to minimize any impacts on the environment and harm with sustainable materials, energy efficiency in design, and resource-saving operations.

2.3.3 Safety

Several measures are implemented to mitigate the potential risks and hazards associated with the operation of the GreetRobot project, ensuring both the safety of users and that of the bystander.

First, the physical design of GreetRobot gives priority to safety with rounded edges, smooth surfaces, and materials that are durable enough to reduce the risk of injury upon contact. Secondly, safety sensors are integrated into the robot's chassis to detect obstacles in order to avoid collisions that could arise from its movement in crowded environments.

Moreover, GreetRobot has advanced motion detection and proximity sensors. These allow the robot to detect nearby individuals and adjust accordingly. This prevents accidental contact and provides safe interaction with users, particularly in dynamic or unpredictable situations.

In addition, the safety of GreetRobot is designed into the software and control systems. Emergency stop mechanisms and fail-safe protocols are built into the robot to bring it to a complete halt in case of malfunction or unexpected behavior that may pose risks to users and bystanders.

In addition, there are rigorous test and validation procedures throughout its development to identify and address any safety concerns proactively. These are done through simulated scenarios, stress testing, and real-world trials to ensure that GreetRobot meets stringent safety standards and regulations.

In all, safety is paramount in the GreetRobot project, and all measures have been taken to mitigate risks and ensure the safety of all interacting with the robot. The thoughtful design, coupled with advanced technology and rigorous testing, ensures that GreetRobot will offer a safe and secure experience for users in various environments.

2.3.4 Ethics

The following will be the strong ethical considerations that will guide the project, GreetRobot, in its development and deployment to ensure that integrity, responsibility, and respect for users and society are at the highest possible level.

First and foremost, a great deal of concern is shown for the privacy and protection of data in the design of GreetRobot. The project will be built on robust encryption methods and secure data storage to ensure the safety of user data. GreetRobot face detection and recognition facilities are designed to work without storing any personal data unless strictly necessary and with explicit authorization from the user, so privacy is never compromised.

Second, transparency is an ethical issue concerning this project. In the use of the GreetRobot, users are provided with clear information on how it works, what it can do, what are its limitations, and what data it processes. This transparency builds trust and allows users to make informed decisions in their interactions with the robot.

Third, the project underlines inclusivity and accessibility: GreetRobot should be user-friendly and accessible for people with all kinds of needs and abilities. This is achieved through intuitive controls, customizable interfaces, and adaptive response so that the widest possible range of users will find the robot easy to operate.

Finally, ethical issues regarding the development and deployment of GreetRobot are continuously addressed proactively with the involvement of ethicists, user representatives, and regulatory bodies. This collaboration ensures that the project stays sensitive to societal norms and expectations, thus paving the way toward responsible ethical robotic technology development.

2.3.5 Cost

The budget of the GreetRobot project is very well-planned to ensure that this project is innovative, qualitative, and at the same time, affordable. The overall expenses can be broken down into some key areas: research and development, hardware and materials, software development, testing and validation, and operational costs.

Firstly, the research and development phase is a great percentage of the budget. This phase would deal with the intensive design work, prototyping, and iterative improvement in order to make sure that GreetRobot meets its performance and safety standards. R&D costs include the cost of engineers, designers, and other specialists in general who contribute to the innovation in the project.

Costs of hardware and materials are another big chunk. This includes the purchasing of high-quality materials for the body, screen, LED board, sensors, and mobility components. Apart from the fact that investing in durable and efficient materials increases the performance of the robot, it also means longevity and reliability for the robot.

Software development is also a very big cost. It includes creating and integrating advanced algorithms for face detection, natural language processing, and motion control. This entails costs of programming, testing, and refining the software to ensure seamless operation and user-friendly interaction.

Operational costs, which include manufacturing, marketing, distribution, and continuous maintenance, are crucial in bringing the GreetRobot into the market and supporting it after the launch. These costs ensure that the robot is produced at scale, reaches its intended audience, and receives necessary updates and support throughout its lifecycle.

2.3.6 Type

The GreetRobot is an interdisciplinary project, incorporating robotics engineering, artificial intelligence, and human-computer interaction in the creation of an interactive and multi-functional robotic platform. It deals with the design of a sturdy and efficient robot equipped with mobility systems, a central control screen, and an expressive LED board. Using the latest technologies of artificial intelligence, GreetRobot can recognize individuals, understand and react to queries by its users, and dynamically adapt its behavior. With a core focus on user-oriented design, the project makes the robot accessible and intuitive for users from various walks of life. This holistic approach aims to develop a complex robotic solution for diverse applications in customer service, education, and entertainment.

2.3.7 Standards

- **Agile Development Methodology**

The GreetRobot project is best suited to the agile development methodology. Being a state-of-the-art robotic platform, its flexibility in this methodology allows the development team to quickly adapt to technological advances and user needs. Agile will assure that GreetRobot meets their specific needs and tastes because of continuous cooperation with all stakeholders and end-users. The iterative approach to progress, together with cost-effectiveness in this methodology, perfectly matches the ambitions of the project. This is possible through the application of innovation in developing solutions that enhance user engagement and interaction quality. Its continuous development focuses on the updating of GreetRobot so it stays abreast of current robotic technologies and gives its users a responsive and dynamic experience.

- **Capability Maturity Model Integration (CMMI)**

It provides Capability Maturity Model Integration (CMMI) with a robust framework to measure and enhance the maturity level of the GreetRobot development process. Applying CMMI standards within the project ensures coherence, efficiency, and high quality throughout its lifecycle. Therein lies the continuous improvement aspect of CMMI: it complements Agile methodologies for the achievement of a culture of continuous learning and refinement. As the GreetRobot project progresses, the team systematically identifies areas for improvement: better face detection, fine-tuning motion control algorithms, or optimization of the user interface for better interaction. This iterative approach is executed with the support of CMMI's standardized processes and documentation, ensuring that each and every development phase is repeatable and coherent. Integrating CMMI into the development process would help GreetRobot reach higher levels of maturity, resulting in ever-increasing efficiency, reliability, and the successful delivery of a sophisticated robotic platform meeting the highest standards of quality and performance.

2.4 SCOPE OF THE PROJECT

The scope of the GreetRobot project covers the full development and implementation of an advanced interactive robotic platform to enhance human-robot interaction across domains. Such a project intends to integrate cutting-edge technologies, such as face detection, natural language processing, and dynamic expression display, so that the robot can be versatile and can interact with users in meaningful and personalized ways.

Other elements within the scope of the project include the design and construction of the physical structure of the robot, ensuring durability, mobility, and user-friendly aesthetics. Critical components will also include the development of sophisticated software algorithms for face recognition, chatbot interaction in real time, and responsive emotional expressions with the view of delivering a seamless and engaging user experience.

Testing and validation will also be done to ensure safety, reliability, and compliance with industry standards. This will include the obstacle avoidance system, emergency stop mechanism, and other safety features that will save users and those around from accidents. It will also encompass a modular hardware architecture to allow future upgrades and scalability, ensuring that GreetRobot will adapt to new emerging technological advancements and the ever-changing user needs.

The project will also put forward the deployment and integration of GreetRobot in different settings, such as customer service, educational institutes, and entertainment venues. It will include the development of user guides, training programs, and support systems to enable smooth implementation and user adoption.

Overall, the GreetRobot project strives to develop a robust, smart, and interactive robot platform that can remarkably improve user experiences across multiple applications.

2.5 SYSTEM CONFIGURATION

2.5.1 Software Requirements

Operating system: Linux

IDE: VS Code, Arduino IDE

Programming languages: Python,C++

2.5.2 Hardware Requirements

CPU: Quad-core ARM A57 @ 1.43 GHz

GPU: 128-core Maxwell

RAM: 4GB

Hard Drive Storage: 128GB

3. LITERATURE OVERVIEW

The literature overview of the GreetRobot project covers current research, developments, and technological advancements in the domains of robotics, HRI, and AI. It delves into related literature to understand the current state-of-the-art technologies, methodologies, and challenges in the development of interactive robotic platforms such as GreetRobot.

Research in the field of robotics spans from the development of the robot body and enhancements in mobility and sensing. In this context, researchers have tried to find new approaches for enhancing the mobility of robots, which include wheeled locomotion and obstacle avoidance algorithms, thereby ensuring smooth navigation in any environment. At the same time, the development of sensors, such as cameras, LiDAR, and proximity sensors, enables the efficient perception and interaction of robots with their surroundings.

The domain of human-robot interaction has mostly dealt with the creation of robots that communicate with users and engage them in an intuitive and effective manner. Research has been carried out exploring strategies for improving user experience, such as natural language processing algorithms for seamless communication, techniques of emotion recognition to empathize with users, and gesture recognition systems for control in an intuitive manner.

The literature review also probes into AI research relevant to the GreetRobot project, namely machine learning, deep learning, and reinforcement learning algorithms. These AI techniques play a crucial role in enabling robots to learn from data, adapt to changing environments, and improve their performance over time. Equally important, studies delve into the ethical dimensions of AI in robotics regarding issues of privacy, bias, and transparency, thereby raising concerns about responsible AI development.

In all, the literature overview offers valuable insights and inspiration for the GreetRobot project, informing its design, development, and implementation strategies. Through the use of existing research and state-of-the-art technologies, GreetRobot seeks to go one step further in the field of human-robot interaction, thus delivering a sophisticated and engaging robotic platform that will enhance user experiences in a variety of applications.

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

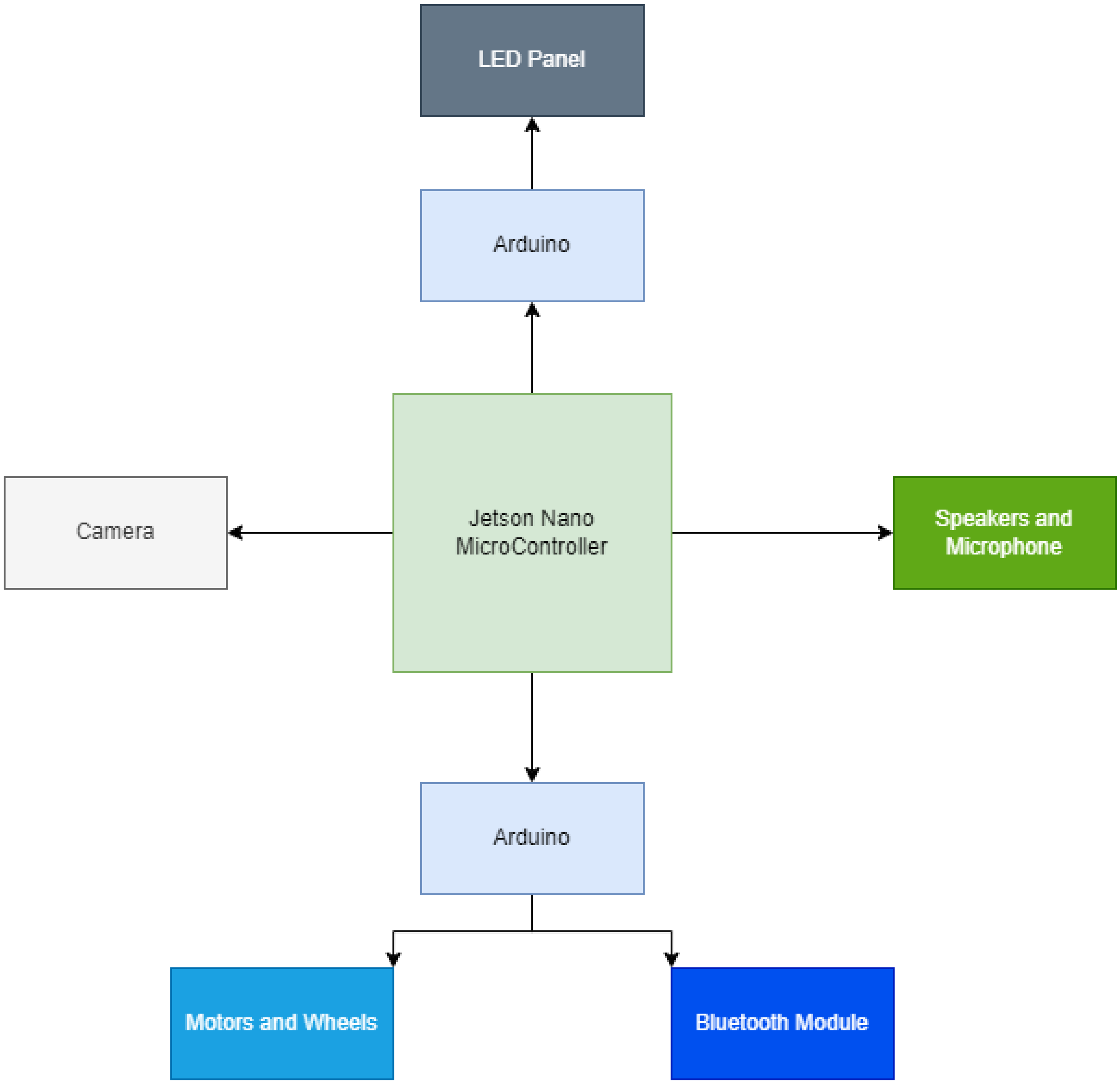

Fig : 4.1 GreetRobot Architecture

4.1.1 MODULE DESCRIPTION

Jetson Nano

Jetson Nano is the central control unit for GreetRobot, meaning the advanced GPU processing and new AI capabilities it provides are unmatched. This makes it an ideal choice for embedded systems because it is small in size and consumes very little power. Its various connectivity options, such as USB, Ethernet, and GPIO pins, help to make integration with other hardware components easy. Besides, support for the most popular AI frameworks like TensorFlow and PyTorch eases the development and deployment of AI models, allowing GreetRobot to identify and communicate with the user via real-time processing of data from sensors and cameras.

Fig : 4.1.1 Jetson Nano

Audiovisual Perception and Communication Modules

For GreetRobot's communication with users, the audio-visual perception and communication modules, comprising the camera, speakers, and microphone, become indispensable. While the camera captures the visual information, allowing for functions like face recognition, the speakers provide the audio output, carrying speech and other sounds. The microphone is used for voice recognition and input, by which GreetRobot can converse with users and answer their queries. These modules put together allow GreetRobot to perceive its environment, recognize faces, and communicate effectively with users through both visual and audio channels of communication.

Fig : 4.1.2 Camera , Microphone and Speakers

Arduino LED Panel

The Arduino LED panel enhances the communication capabilities of GreetRobot in its ability to provide visual expression and feedback. In control is an Arduino microcontroller, which provides a panel display with patterns, colors, and messages conveying emotions, notifications, and status indicators. Programmable, GreetRobot can dynamically create LED effects and further maximize user interaction and visual expression capability.

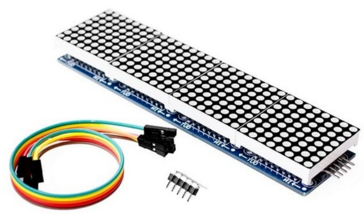

Fig : 4.1.3 LED Panel

Arduino Motors and Wheels with Bluetooth

The movement and remote control functionality of GreetRobot are provided with the Bluetooth-enabled Arduino motors and wheels. The motors drive movement and steering, and the wheels ensure smooth navigation on different surfaces. With Bluetooth communication, users have the means to control GreetRobot remotely, providing flexibility and mobility within different environments. All of these modules come together to provide the foundation for GreetRobot's physical movement and interaction capabilities, giving it the ability to move freely through spaces and react with user commands agilely and precisely.

Fig : 4.1.4 Bluetooth, Motor

5.IMPLEMENTATION

5.1 IMPLEMENTATION

GreetRobot project implementation involves a sequence of steps to give life to the conceptual design by integrating hardware components, developing software algorithms, and testing the system for functionality and performance. The implementation process can be divided into several key phases:

- **Hardware Assembly and Integration:**
  Hardware components of GreetRobot, including Jetson Nano, camera, speakers, microphone, LCD screen, Arduino LED panel, motors, wheels, and Bluetooth module, are assembled. Each component is integrated into the chassis of the robot carefully, ensuring proper connectivity and functionality.
- **Software Development:**
  After assembly of the hardware, work is focused on software development that includes coding for functions such as face detection and recognition, natural language processing, motion control, LED panel control, and Bluetooth communication. AI algorithms are developed and trained to enable GreetRobot to interact with users intelligently and respond to commands and queries.
- **Integration and Testing:**
  With hardware and software parts in place, integration testing is conducted to ensure that all parts of GreetRobot work together seamlessly. This includes the testing of communication between different modules, verification of sensor inputs, and validation of the performance of AI algorithms. Any issues or bugs identified through testing are resolved and fixed to ensure the reliability and stability of the system.
- **User Interface Design:**
  This is the user interface design phase for developing an intuitive and engaging interface for the control and interaction with GreetRobot. It involves designing the screens and menus for the LCD display, the implementation of touch controls, and ensuring the optimization of the user experience for ease of use and accessibility.

- **Deployment and Optimization:**

  When the entire system is tested and validated, GreetRobot is ready for deployment in real-world environments. In this phase, final optimization is also made for boosting the performance, reliability, and user satisfaction of the systems. Users are also supported in training and maintenance materials while using GreetRobot effectively.

- **Monitoring and Maintenance:**

  After being deployed, continued monitoring and maintenance are needed to guarantee that the GreetRobot continues working and performing at high levels. The idea here is that routine software updates, hardware maintenance, and troubleshooting of problems that arise during operation are done in a timely manner. The continuous feedback from users on system performance is useful for planning improvements to be done in the future on GreetRobot**.**

5.2 SAMPLE CODE

Face Recognition Code

```
import face_recognition

import cv2

from datetime import datetime, timedelta

import numpy as np

import pickle

import pyttsx3

from inputimeout import inputimeout

engine = pyttsx3.init()

rate = engine.getProperty('rate')

volume = engine.getProperty('volume')

voices = engine.getProperty('voices')

engine.setProperty('rate', 150)

engine.setProperty('voice', voices[22].id)

engine.setProperty('volume',0.5)

# Initialize lists to store known face encodings and metadata

known_face_encodings = []

known_face_metadata = []


def save_known_faces():

    # Save known faces to a file

    with open("known_faces.pkl", "wb") as face_data_file:
```

```
        face_data = [known_face_encodings, known_face_metadata]
        pickle.dump(face_data, face_data_file)
        print("Known faces backed up to disk.")


def load_known_faces():
    global known_face_encodings, known_face_metadata
    try:
        with open("known_faces.pkl", "rb") as face_data_file:
            known_face_encodings, known_face_metadata = pickle.load(face_data_file)
            print("Known faces loaded from disk.")
            print(known_face_metadata)
    except FileNotFoundError as e:
        print("No previous face data found - starting with a blank known face list.")
        pass


def register_new_face(face_encoding, face_image):
    # Add a new person to the known faces list
    known_face_encodings.append(face_encoding)

    known_face_metadata.append({
        "first_seen": datetime.now(),
        "first_seen_this_interaction": datetime.now(),
        "last_seen": datetime.now(),
```

```
        "registrations": [datetime.now()],
        "seen_count": 1,
        "seen_frames": 1,
        "face_image": face_image,
        "First": "John",
        "Last": "Doe",
    })


def lookup_known_face(face_encoding):
    # Check if the face is in the known faces list
    metadata = None
    if len(known_face_encodings) == 0:
        return metadata

    face_distances = face_recognition.face_distance(known_face_encodings,
face_encoding)
    best_match_index = np.argmin(face_distances)

    if face_distances[best_match_index] < 0.65:
        metadata = known_face_metadata[best_match_index]
        metadata["last_seen"] = datetime.now()
        metadata["seen_frames"] += 1
```

```
        if datetime.now() - metadata["first_seen_this_interaction"] >
timedelta(minutes=5):
            metadata["first_seen_this_interaction"] = datetime.now()
            metadata["seen_count"] += 1
            metadata["registrations"].append(datetime.now())

    return metadata

def main_loop():
    video_capture = cv2.VideoCapture(0)
    number_of_faces_since_save = 0

    while True:
        ret, frame = video_capture.read()
        small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
        rgb_small_frame = small_frame[:, :, ::-1]

        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame,
face_locations)

        face_labels = []
        for face_location, face_encoding in zip(face_locations, face_encodings):
```

```
metadata = lookup_known_face(face_encoding)


if metadata is not None:
    time_at_door = datetime.now() - metadata['first_seen_this_interaction']
    face_label = f"At door {int(time_at_door.total_seconds())}s"
    if time_at_door.total_seconds() <= 1:
        print("Hello")
        engine.say("Hello")
        print(metadata)
        if metadata["First"]!="John":
            engine.say(metadata["First"])
            print(metadata["First"])
        engine.runAndWait()


else:
    face_label = "New visitor!"
    top, right, bottom, left = face_location
    face_image = small_frame[top:bottom, left:right]
    face_image = cv2.resize(face_image, (150, 150))
    register_new_face(face_encoding, face_image)


face_labels.append(face_label)
```

```
    for (top, right, bottom, left), face_label in zip(face_locations, face_labels):
        top *= 4
        right *= 4
        bottom *= 4
        left *= 4

        # cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
        # cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255),
cv2.FILLED)
        # cv2.putText(frame, face_label, (left + 6, bottom - 6),
cv2.FONT_HERSHEY_DUPLEX, 0.8, (255, 255, 255), 1)

    number_of_recent_visitors = 0
    for metadata in known_face_metadata:
        if datetime.now() - metadata["last_seen"] < timedelta(minutes=5) and
metadata["seen_frames"] > 5:
            x_position = number_of_recent_visitors * 150
            face_image_resized = cv2.resize(metadata["face_image"], (150, 150))
            frame[30:180, x_position:x_position + 150] = face_image_resized
            number_of_recent_visitors += 1

            visits = metadata['seen_count']
            visit_label = f"{visits} visits"
```

```
            if visits == 1:
                visit_label = "First visit"
            # cv2.putText(frame, visit_label, (x_position + 10, 170),
cv2.FONT_HERSHEY_DUPLEX, 0.6, (255, 255, 255), 1)

        # if number_of_recent_visitors > 0:
        #     cv2.putText(frame, "Visitors at Door", (5, 18),
cv2.FONT_HERSHEY_DUPLEX, 0.8, (255, 255, 255), 1)

        # cv2.imshow('Video', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            save_known_faces()
            break

        if len(face_locations) > 0 and number_of_faces_since_save > 100:
            save_known_faces()
            number_of_faces_since_save = 0
        else:
            number_of_faces_since_save += 1

    video_capture.release()
    cv2.destroyAllWindows()
```

```
if _name_ == "_main_":
    load_known_faces()
    main_loop()
```

Chatbot. Code

```
from chatterbot import ChatBot

import pyttsx3

import speech_recognition as sr

import serial

import time


chatbot = ChatBot ('S')

engine = pyttsx3.init()

rate = engine.getProperty('rate')

volume = engine.getProperty('volume')

voices = engine.getProperty('voices')

engine.setProperty('rate', 100)

engine.setProperty('voice', 'en-usa')

engine.setProperty('volume',0.7)

arduino = serial.Serial(

port = '/dev/ttyACM0',

baudrate = 9600,

bytesize = serial.EIGHTBITS,

parity = serial.PARITY_NONE,

stopbits = serial.STOPBITS_ONE,

timeout = 5,

# xonxoff = False,
```

```
# rtscts = False,

# dsrdtr = False,

# writeTimeout = 2

)

arduino.write(bytes('L\n','utf-8'))

r = sr.Recognizer()


engine.say("Bot Initiated")

arduino.write(bytes('S\n','utf-8'))

engine.runAndWait()

arduino.write(bytes('L\n','utf-8'))


# print("hello")

# while True:

#     arduino.write("SPEAK".encode('utf-8'))

#     time.sleep(10)

   #

while True:

   with sr.Microphone() as source:

       try:

           r.adjust_for_ambient_noise(source)

           audio = r.listen(source)

           res =r.recognize_google(audio)
```

```
    print("hey")
    if "bye" in res:
        break
    arduino.write(bytes('S\n','utf-8'))
    engine.say(chatbot.get_response(res))


    engine.runAndWait()
    time.sleep(2)
    arduino.write(bytes('L\n','utf-8'))
except:
    engine.say("Please Repeat, could not recognise")
    engine.runAndWait()
    print("problem")
    pass
```

Training of Chatbot Code

```
import json
import random
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer

# Load the data from the JSON file
data = json.loads(open('chat.json', 'r').read())

# Create a ChatBot instance
chatbot = ChatBot('S')

# Create a ListTrainer instance
trainer = ListTrainer(chatbot)

# Iterate through the intents in the data
for intent in data['intents']:
    # Add patterns to the training data
    patterns = intent['patterns']
    responses = intent['responses']

    # Train the chatbot with patterns and responses
    for pattern in patterns:
        trainer.train([pattern, responses[random.randint(0, len(responses) - 1)]])
```

Main Code

```
import os
import tkinter as tk
from tkinter import simpledialog
import subprocess
import threading

# Global variables to store script processes
speak_process = None
face_process = None

def start_scripts():
    global speak_process, face_process
    print("Starting both scripts...")
    speak_process = subprocess.Popen(["python3.6", "speak.py"])
    face_process = subprocess.Popen(["python3.6", "face.py"])

def stop_scripts():
    global speak_process, face_process
    print("Stopping both scripts...")
    if speak_process:
        speak_process.terminate()
        speak_process = None
    if face_process:
        face_process.terminate()
        face_process = None

def execute_script1():
```

```
    stop_scripts()
    print("Executing ChatBot script...")
    chatbot_thread = threading.Thread(target=run_chatbot_script)
    chatbot_thread.start()

def run_chatbot_script():
    subprocess.Popen(["python", "chatbot.py"])

def execute_script2():
    stop_scripts()
    print("Executing GreetBot script...")
    greetbot_thread = threading.Thread(target=run_greetbot_script)
    greetbot_thread.start()

def run_greetbot_script():
    subprocess.Popen(["python", "greetbot.py"])

def on_exit():
    stop_scripts()
    root.destroy()
    shutdown_system()

def open_on_screen_keyboard():
    subprocess.Popen(["/usr/bin/onboard"], shell=True)

def close_button_clicked():
    open_on_screen_keyboard()
    root.after(1000, ask_password)  # Delay opening the password dialog
```

```
def ask_password():
    password = simpledialog.askstring("Password", "Enter password:", show='*')
    if password == "BOOT":  # Replace with your actual password
        stop_scripts()
        on_exit()


def shutdown_system():
    os.system("sudo shutdown now")  # Adjust the shutdown command based on your system


root = tk.Tk()
root.title("AdminBot")
root.attributes("-fullscreen", True)
root.protocol("WM_DELETE_WINDOW", on_exit)


frame = tk.Frame(root, bg="lightblue")
frame.pack(expand=True, fill=tk.BOTH)
frame.grid_rowconfigure(0, weight=1)
frame.grid_columnconfigure(0, weight=1)
close_button = tk.Button(root, text="Close", command=close_button_clicked, font=("Arial", 16), bg="red", fg="white")
close_button.pack(side=tk.TOP, anchor=tk.NE, padx=10, pady=10)


button1 = tk.Button(frame, text="ChatBot", command=execute_script1, height=15, width=40, font=("Arial", 20), bg="grey", fg="white")
button2 = tk.Button(frame, text="GreetBot", command=execute_script2, height=15, width=40, font=("Arial", 20), bg="grey", fg="white")


button1.grid(row=0, column=0, padx=(50, 10), pady=10)
```

```
button2.grid(row=0, column=1, padx=(10, 50), pady=10)

# start_scripts()

root.mainloop()
```

6. TESTING

6.1 TESTING

GreetRobot's testing is a crucial phase in the development of the project, involving unit, integration, system, user acceptance, performance, security, regression, and iterative testing. In the unit tests, individual hardware and software components are tested for their functioning. Integration testing checks the interaction among the modules in a smooth fashion. System testing evaluates the overall behavior and performance of GreetRobot under real-world conditions, while user-acceptance testing validates its usability and effectiveness from the end-user's perspective. Performance testing checks its response time, reliability, and scalability, and security testing checks for vulnerabilities. Regression testing ensures that updates do not introduce new defects; iterative testing, based on user feedback, drives continuous refinements. This comprehensive way of testing assures the validation of GreetRobot on functionality, reliability, and user satisfaction, making it ready for deployment in various applications.

6.2 TEST CASES

S.No	TestCase	Expected Output	Actual Output	Result
1	Face Recognition.	Successful detection and recognition of faces.	Successful detection and recognition of faces	Success
2	Speech Interaction	Accurate understanding and response to user queries	Accurate understanding and response to user queries	Success
3	Motion Control	Smooth navigation	Smooth navigation	Success
4	LED Panel Display	Clear display of emotions	Clear display of emotions	Success
5	Bluetooth Communication	Successful pairing and communication with external devices	Successful pairing and communication with external devices	Success
6	Integration Testing	Seamless interaction between hardware and software components	Seamless interaction between hardware and software components	Success

7. OUTPUT SCREENS

Fig 7.1: GreetRobot

8. CONCLUSION

8.1 CONCLUSION

The end of the GreetRobot project is the consequence of hard research, design, development, and testing for an innovative and interactive robotic platform. Throughout the project journey, the team managed to integrate complex hardware modules, such as the Jetson Nano, camera, speakers, and motors, with advanced software algorithms like those of face recognition, natural language processing, and motion control. The result is an extremely versatile and smart robot able to interface with users through facial recognition, speech interaction, and expressive LED displays.

The successful realization of GreetRobot offers a wide array of opportunities for its application within diverse areas, such as customer service, education, and entertainment. Using state-of-the-art technologies and a user-centered design approach, GreetRobot provides a unique and engaging experience for users, enhances human-robot interaction, and paves the way for future innovations in robotics.

Concurrently, a further refinement and optimization process will continue in order to develop an improved performance, reliability, and user satisfaction of GreetRobot. The continuous flow of feedback from the users and other stakeholders will inform the iterative refinements of the robot's functionality and user experience so that GreetRobot is kept at the leading edge of robotics technology. As GreetRobot finds its place in real-world environments, it promises to change how we interact with robots and opens a new road toward advanced and engaging robotics.

8.2 FUTURE ENHANCEMENTS

Future enhancements to the GreetRobot project offer much room to take it to the next level in capability, versatility, and user experience. The following areas open up development opportunities and refinement pathways for new developments in human-robot interaction:

- **Better AI Integration:** Integrating more advanced AI algorithms would give GreetRobot an even more significant cognitive set of capabilities for better understanding and replying to more complicated user queries. Making him learn and adapt from the surroundings using deep learning techniques could further fine-tune his skills in recognizing faces, emotions, and speech patterns with more exactness and subtlety.
- **Multimodal Interaction:** Expanding the interaction modalities of GreetRobot beyond voice and face recognition will improve the user engagement levels. It would allow users to interact with GreetRobot using gestures through the incorporation of gesture recognition technology, thereby expanding the interaction possibilities and allowing the system to capture diverse user preferences.
- **Extended Mobility**: Improved mobility capabilities of GreetRobot will significantly open up its usage in a wide array of applications and settings. Advanced navigation systems, such as SLAM, can be applied to GreetRobot for improved navigation in complex environments, thus making the robot move around freely and flexibly in dynamic and crowded environments.
- **Advanced Expressiveness:** Further work on the expressive ability of GreetRobot will enable it to display emotions and intentions more successfully. More enhanced algorithms for generating facial expressions and a wider range of emotive displays on the LED panel can lead to richer and deeper interactions, thus creating a closer bond with the users on an emotional level.

9. BIBLIOGRAPHY

9.1 WEBSITE REFERNCES

1. https://jfrog.com/connect/post/installing-cuda-on-nvidia-jetson-nano/
2. https://dev.to/ajeetraina/install-cuda-on-jetson-nano-2b06
3. https://medium.com/@ageitgey/build-a-face-recognition-system-for-60-with-the-new-nvidia-jetson-nano-2gb-and-python-46edbddd7264
4. https://www.geeksforgeeks.org/how-to-install-tkinter-on-linux/
5. https://chatterbot.readthedocs.io/en/stable/
6. https://realpython.com/build-a-chatbot-python-chatterbot/

10. APENDICES

10.1 Methodologies Used

An Agile software development approach was followed to ensure flexibility, adaptability, and continuous improvement in the development of the GreetRobot project. Scrum methodology was used within the Agile framework for managing project tasks and iterations effectively. Following Scrum principles, time-boxing iterations, regular stand-up meetings, and backlog prioritization, development activities were planned, executed, and reviewed. Elements of the Kanban methodology have been adopted to visualize workflow and optimize task management, especially in the integration and test phases of the project.

Tools and Technologies:

A complete software toolset and set of technologies were employed while working on GreetRobot; the list includes the following: Visual Studio Code as the base Integrated Development Environment, Git for version control management, Python as a programming language for software development, TensorFlow for Machine Learning and AI features, and Arduino IDE for development on microcontrollers.

Software Architecture

The software architecture of GreetRobot is laid out in a modular and layered way, incorporating constituent elements of both client-server architecture and microservices architecture. The system is segregated into clearly defined modules: face detection and recognition, speech processing, motion control, LED panel display, and Bluetooth communication, each interacting with each other via well-defined interfaces. This architecture supports scalability, maintainability, and ease of integration with new features and functionalities.

Testing Methodologies

Many different test methodologies have been put to use within the development and testing phases of GreetRobot: unit testing, integration testing, system testing, and acceptance testing. Unit tests were conducted on the functionality of different units, while integration tests were conducted on the interaction between units. System tests evaluated the overall behavior and performance of the robot, and acceptance tests checked that user requirements were met. Automated testing frameworks, such as pytest and Selenium, were used for designing test cases, executing test cases, and reporting test results in an automated fashion.

Quality Assurance Processes

Quality assurance processes were in place throughout the development of GreetRobot to assure reliability, performance, and security. Code reviews, static code analysis, and peer testing were carried out in order to reveal and fix possible problems at the early stages of development. Furthermore, security audits and compliance checks were carried out to meet industry standards and regulations with the purpose of ensuring the integrity and confidentiality of user data.

Maintenance and Support Procedures

Maintenance and support procedures have been set up so that GreetRobot is in continuous use with reliability. Issues that have been reported were tracked through a bug tracking system, prioritized, and regularly released software updates for the reduction or elimination of bugs, performance enhancement, and addition of new features. Processes have been implemented for customer support to answer queries and offer support when needed. SLAs and escalation procedures are set to manage customer expectations and ensure that problems are solved promptly.

10.2 TESTING METHODS USED

Types Of Tests

Manual Testing

Manual testing is the process of testing software by hand to learn more about it, to find what is and isn't working. This usually includes verifying all the features specified in requirements documents, but often also includes the testers trying the software with the perspective of their end users in mind. There are lots of sophisticated tools on the market to help with manual testing like Test Pad.

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

• Valid input: identified classes of valid input must be accepted.

• Invalid input: identified classes of invalid input must be rejected.

• Functions: identified functions must be exercised.

• Output: identified classes of applications must be exercised.

Systems/procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify business process flows, data fields, re defined processes, successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Testing

System testing ensures that the entire integrated software system needs requirement. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre driven process links and integration points.

System testing in the "Vakeel" application involves evaluating its entire functionality to ensure it meets requirements and operates correctly. This includes testing user journeys like registration, query submission, and response receipt, as well as integration points with external services and data consistency. By conducting system testing, the development team ensures the reliability and efficiency of the "Vakeel" platform, identifying and addressing any issues to deliver a seamless user experience.

White Box Testing

White box testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black box testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Block box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a block box, you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

11. PLAGIARISM REPORT

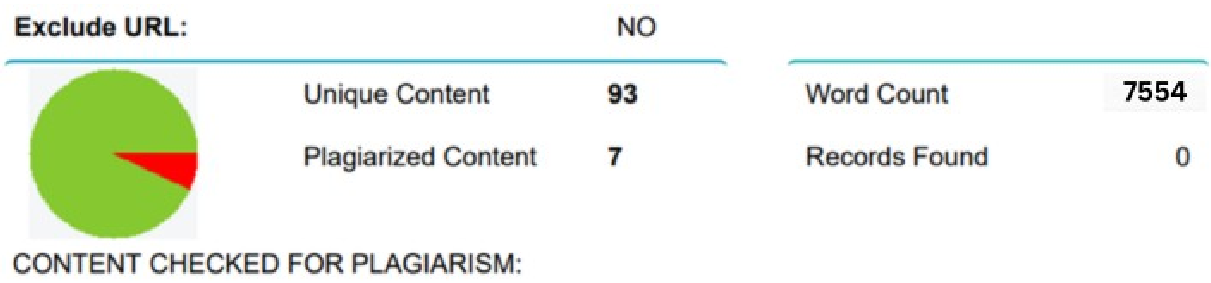

Fig 11 Generated Report