Chapter 1:

1.1 Introduction

Robotics is an interdisciplinary field that integrates engineering, computer science, and artificial intelligence to design, construct, and operate robots. Robots are programmable machines capable of carrying out complex tasks autonomously or semi-autonomously. They are widely used in various industries, including manufacturing, healthcare, agriculture, and space exploration, to perform tasks that are dangerous, repetitive, or require high precision.

The development of robotic systems has revolutionized modern industries by increasing efficiency, reducing human error, and enabling the automation of tasks that were previously impossible or impractical. One of the most significant advancements in robotics is the development of **robotic arms**, which are mechanical devices designed to mimic the movements of a human arm. These arms are equipped with joints and end-effectors (such as grippers) that allow them to perform a wide range of tasks, from assembling products on a factory line to performing delicate surgical procedures.

Robotic Arms: Applications and Importance

Robotic arms are versatile tools used in various applications, including:

- **Industrial Automation**: In manufacturing, robotic arms are used for tasks such as welding, painting, assembly, and material handling. They improve productivity, ensure consistency, and reduce the risk of workplace injuries.
- **Healthcare**: Robotic arms are used in surgeries to perform precise and minimally invasive procedures. They are also used in rehabilitation to assist patients with mobility issues.
- **Space Exploration**: Robotic arms are deployed on space missions to perform tasks such as collecting samples, repairing equipment, and assisting astronauts.
- **Agriculture**: Robotic arms are used for tasks such as harvesting crops, pruning plants, and sorting produce.
- **Education and Research**: Robotic arms are used in academic institutions and research labs to teach robotics, conduct experiments, and develop new technologies.

The ability to control robotic arms with high precision and flexibility makes them indispensable in modern technology. However, the design and control of robotic arms present significant challenges, including the need for accurate motion control, efficient power management, and seamless integration with other systems.

Problem Statement

Traditional robotic arm control systems often rely on wired connections or pre-programmed sequences, which limit their flexibility and usability. Wired connections restrict the mobility of the robotic arm, while pre-programmed sequences may not be suitable for dynamic environments where real-time adjustments are required. Additionally, many existing systems lack user-friendly interfaces, making them difficult to operate for non-experts.

To address these limitations, there is a growing need for **wireless robotic arm control systems** that offer real-time control, user-friendly interfaces, and advanced features such as **recording and playback** of movements. Such systems would enable users to control robotic arms remotely, perform complex tasks with ease, and adapt to changing requirements.

Objectives of the Project

The primary objective of this project is to design and develop a **robotic arm control system** that addresses the limitations of traditional systems. The specific objectives include:

1. **Wireless Control**: To enable wireless control of the robotic arm using a web-based interface.
2. **Precision Control**: To provide precise control of the arm's movements (Base, Shoulder, Elbow, and Gripper) using servo motors.
3. **User-Friendly Interface**: To develop an intuitive and visually appealing web interface for controlling the robotic arm.
4. **Recording and Playback**: To implement features for recording and playing back sequences of movements.

5. **Integration of Hardware and Software**: To integrate hardware components (ESP32 microcontroller, servo motors) with software (Arduino code, HTML/CSS/JavaScript) to create a seamless control system.
6. **Real-Time Communication**: To establish real-time communication between the web interface and the robotic arm using WebSocket protocol.

Scope of the Project

This project focuses on the design and implementation of a **4-degree-of-freedom (4-DOF) robotic arm** controlled by an **ESP32 microcontroller**. The system allows users to control the arm's movements wirelessly through a web interface. The key features of the system include:

- **Real-Time Control**: Users can adjust the position of each joint (Base, Shoulder, Elbow, Gripper) in real-time using sliders on the web interface.
- **Recording and Playback**: Users can record a sequence of movements and play it back to automate repetitive tasks.
- **Wireless Connectivity**: The ESP32 microcontroller acts as a WiFi access point, enabling wireless communication between the web interface and the robotic arm.
- **User-Friendly Interface**: The web interface is designed to be intuitive and visually appealing, making it easy for users to operate the robotic arm.

The project demonstrates the integration of hardware and software to create a functional and efficient robotic arm control system. It also highlights the potential of wireless communication and web-based interfaces in robotics.

Significance of the Project

This project has several significant implications:

1. **Advancement in Robotics**: By incorporating wireless control and advanced features like recording and playback, this project contributes to the development of more flexible and user-friendly robotic systems.
2. **Educational Value**: The project serves as a valuable learning tool for students and researchers interested in robotics, IoT, and microcontroller programming.
3. **Practical Applications**: The system can be adapted for various real-world applications, such as industrial automation, healthcare, and education.
4. **Cost-Effective Solution**: The use of affordable components (ESP32, servo motors) makes the system accessible to a wide range of users, including hobbyists and small businesses.

1.2 Motivation

Phase 1: Addressing Real-World Challenges

The development of a **robotic arm control system** is motivated by the need to address several real-world challenges in industries and everyday life. These challenges include:

1. **Automation in Industries**:
    - Industries such as manufacturing, automotive, and electronics rely heavily on automation to improve efficiency and reduce costs.
    - Traditional robotic arms are often limited by wired connections and lack of flexibility, making it difficult to adapt to dynamic environments.
    - A wireless robotic arm control system, like the one developed in this project, can provide greater flexibility and ease of use, enabling industries to automate complex tasks more effectively
2. **Precision and Safety in Healthcare**:
    - In the healthcare sector, robotic arms are used for surgeries, rehabilitation, and patient care.

- Precision and safety are critical in these applications, as even minor errors can have serious consequences.
- This project's focus on precise control and real-time adjustments ensures that the robotic arm can perform delicate tasks with high accuracy, making it suitable for healthcare applications.

3. **Accessibility and Usability**:
    - Many existing robotic arm systems are complex and require specialized knowledge to operate.
    - By developing a user-friendly web interface, this project aims to make robotic arm technology accessible to a wider audience, including students, hobbyists, and small businesses.
4. **Cost-Effective Solutions**:
    - High-end robotic arms are often expensive, making them inaccessible to many users.
    - This project uses affordable components like the ESP32 microcontroller and standard servo motors, providing a cost-effective solution without compromising functionality.

Phase 2: Technological Advancements

The motivation for this project is also driven by recent advancements in technology, which have made it possible to develop more sophisticated and accessible robotic systems. These advancements include:

1. **Internet of Things (IoT)**:
    - The integration of IoT in robotics has enabled wireless communication and remote control of devices.

- This project leverages IoT by using the ESP32 microcontroller to establish a WiFi connection, allowing users to control the robotic arm from any device with a web browser.

2. **Web-Based Interfaces**:
    - Web technologies like HTML, CSS, and JavaScript have made it easier to create intuitive and visually appealing user interfaces.
    - The web interface developed in this project provides a seamless and interactive experience for users, making it easier to control the robotic arm.

3. **Microcontroller Technology**:
    - Microcontrollers like the ESP32 have become more powerful and affordable, enabling the development of complex systems with minimal hardware.
    - The ESP32's dual-core processor and built-in WiFi capabilities make it an ideal choice for this project.

4. **Open-Source Platforms**:
    - Platforms like Arduino provide a wealth of resources, libraries, and community support, making it easier to develop and prototype robotic systems.
    - This project uses the Arduino framework to program the ESP32, ensuring compatibility with a wide range of sensors and actuators.

Phase 3: Educational and Research Value

This project is highly motivated by its potential to contribute to education and research in the fields of robotics, IoT, and embedded systems. The key educational and research motivations include:

1. **Hands-On Learning**:
    - The project provides students with an opportunity to gain hands-on experience in designing, building, and programming robotic systems.

- It covers a wide range of topics, including microcontroller programming, servo motor control, web development, and wireless communication.

2. **Interdisciplinary Approach**:
   - The project integrates concepts from multiple disciplines, including electrical engineering, computer science, and mechanical engineering.
   - This interdisciplinary approach helps students develop a holistic understanding of robotics and its applications.
3. **Research Opportunities**:
   - The project opens up new avenues for research in areas such as real-time control systems, human-robot interaction, and IoT-based automation.
   - Researchers can build upon this project to explore advanced features like machine learning, computer vision, and autonomous navigation.
4. **Skill Development**:
   - By working on this project, students and researchers can develop valuable skills such as problem-solving, critical thinking, and teamwork.
   - These skills are essential for careers in robotics, automation, and related fields.

Phase 4: Societal Impact

The development of a wireless robotic arm control system has the potential to create a positive impact on society in several ways:

1. **Empowering Small Businesses**:
   - Small businesses often lack the resources to invest in expensive automation systems.
   - This project's cost-effective solution can empower small businesses to automate tasks, improve productivity, and compete with larger companies.

2. **Enhancing Quality of Life**:
   - Robotic arms can assist individuals with disabilities or limited mobility in performing everyday tasks.
   - The user-friendly interface and precise control of this system make it suitable for assistive applications.
3. **Promoting Innovation**:
   - By making robotic technology more accessible, this project encourages innovation and creativity among students, hobbyists, and entrepreneurs.
   - It inspires the development of new applications and solutions that address real-world problems.
4. **Environmental Benefits**:
   - Automation can reduce waste and improve resource efficiency in industries.
   - This project’s focus on precision and control ensures that tasks are performed with minimal errors, reducing material waste and energy consumption.

Chapter 2:

2.1 Literature Survey

The field of robotic arm control systems has seen significant advancements over the years, driven by the need for automation, precision, and flexibility in various industries. Researchers and engineers have explored a wide range of methodologies and technologies to develop systems that can perform complex tasks with high accuracy and efficiency. This literature survey provides an overview of existing research and technologies related to robotic arm control systems, highlighting their strengths, limitations, and relevance to the current project. One of the earliest approaches to robotic arm control involved the use of wired connections and pre-programmed sequences. These systems were primarily used in industrial settings, where repetitive tasks such as welding, painting, and assembly required high precision and consistency. While effective for specific applications, these systems were limited by their lack of flexibility and inability to adapt to dynamic environments. For example, any changes in the task or environment required significant reprogramming and reconfiguration, making them unsuitable for applications that demanded real-time adjustments. Despite these limitations, traditional wired systems laid the foundation for robotic arm technology and demonstrated the potential of automation in improving productivity and reducing human error. With the advent of wireless communication technologies, researchers began exploring wireless control systems for robotic arms. These systems offered greater mobility and flexibility, enabling remote control and real-time adjustments. One notable example is the use of Bluetooth for controlling small-scale robotic arms in educational and research settings. Bluetooth-based systems are cost-effective and easy to implement, making them popular among hobbyists and students. However, they are limited by their short range and low bandwidth, which restrict their use in more complex and demanding applications. This limitation highlighted the need for more robust wireless communication protocols, such as Wi-Fi, which could provide greater range, bandwidth, and reliability.

The integration of Internet of Things (IoT) technologies into robotic arm control systems has been a significant development in recent years. IoT-based systems leverage wireless communication and cloud platforms to enable remote control and monitoring of robotic arms. For instance, Kumar et al. (2020) developed an IoT-based control system using the ESP32 microcontroller and a web interface. Their system allowed users to control a robotic arm in real-time from any device with a web browser, demonstrating the feasibility of using IoT for robotic applications. While this approach offered several advantages, such as scalability and remote access, it lacked advanced features like recording and playback, which are essential for automating repetitive tasks. This gap in functionality provided an opportunity for further innovation and inspired the development of more sophisticated systems.

Web-based control interfaces have also gained popularity due to their user-friendly design and cross-platform compatibility. Lee et al. (2019) developed a web-based control interface for a robotic arm using HTML, CSS, and JavaScript. Their interface provided real-time feedback and allowed users to control the arm's movements through sliders and buttons. The intuitive design and ease of use made the system accessible to non-experts, expanding its potential applications beyond industrial settings. However, the system was limited to basic control functions and did not incorporate advanced features like recording and playback. This limitation underscored the need for more comprehensive web-based solutions that could offer both simplicity and advanced functionality.

In addition to hardware and software advancements, researchers have explored the use of machine learning and artificial intelligence (AI) to optimize robotic arm control. Zhang et al. (2021) used machine learning algorithms to improve the accuracy and adaptability of a 6-degree-of-freedom (6-DOF) robotic arm. Their system was capable of learning from past movements and optimizing its performance for complex tasks. While this approach demonstrated significant potential, it also posed challenges such as high computational requirements and complexity, making it less accessible for small-scale applications. Nevertheless, the integration of AI and machine learning into robotic arm control systems remains a promising area of research, with the potential to enhance autonomy and adaptability.

Another important area of research is the development of low-cost robotic arm systems for education and small-scale applications. Patel et al. (2022) designed a low-cost robotic arm using Arduino and servo motors, which was specifically aimed at teaching robotics concepts to students. Their system was affordable, easy to assemble, and provided hands-on learning opportunities. However, it lacked wireless control and advanced features, limiting its functionality. This work highlighted the importance of cost-effective solutions and inspired the use of Arduino-compatible components in the current project. By combining affordability with advanced features, the proposed system aims to bridge the gap between accessibility and functionality. Despite the advancements in robotic arm control systems, several gaps and challenges remain. Many existing systems are designed for specific tasks and lack the flexibility to adapt to different applications. Additionally, some systems are too complex for non-experts to operate, limiting their usability. The lack of advanced features, such as recording and playback, is another limitation that restricts the potential of these systems. Furthermore, high-end robotic arm systems are often expensive, making them inaccessible to many users. While wireless control systems have been explored, there is a need for more robust and reliable communication protocols that can support real-time control and advanced functionality.

The current project addresses these gaps by developing a flexible and user-friendly robotic arm control system that incorporates advanced features like recording and playback. By leveraging the ESP32 microcontroller and web-based technologies, the proposed system aims to provide a cost-effective and accessible solution for a wide range of applications. The integration of wireless communication and real-time control ensures that the system can adapt to dynamic environments and perform complex tasks with high precision. Additionally, the use of a web-based interface makes the system accessible to non-experts, expanding its potential user base.

In conclusion, the literature survey highlights the significant progress made in the field of robotic arm control systems, as well as the challenges and opportunities that remain. The proposed project builds on existing research by addressing key limitations and incorporating advanced features to create a more versatile and accessible system. By combining hardware and software innovations, the project aims to contribute to the ongoing development of robotic arm technology and demonstrate its potential for real-world applications.

Chapter 3:

Proposed Work

3.1 Overview

The **development of the Robotic Arm Control System** is structured into two primary phases: **Proposed Work** and **Methodology**. The **Proposed Work** defines the key objectives, features, and expected outcomes of the system, ensuring it meets the requirements for precision, flexibility, and automation. This phase includes selecting suitable components such as servo motors, microcontrollers (ESP32), and communication interfaces for seamless operation. It also considers integrating **IoT-based remote control** and **real-time feedback mechanisms** to enhance accuracy and responsiveness.

The **Methodology** describes the systematic approach taken to design, develop, and test the robotic arm. It begins with **requirement analysis**, identifying the necessary hardware and software. The next step involves **mechanical design** using CAD tools, followed by assembling components and programming the ESP32 microcontroller to execute movement logic. **Control algorithms** are developed to handle real-time commands, and an **IoT-based interface** is implemented for remote operation. After integration, **testing and calibration** ensure optimal performance, refining the system through iterations to enhance stability, accuracy, and efficiency.

3.2 Proposed Work

1. System Features

- Wireless Control: The system will allow users to control the robotic arm wirelessly via a web interface.
- Real-Time Adjustments: Users can adjust the position of each joint (Base, Shoulder, Elbow, Gripper) in real-time using sliders.
- Recording and Playback: The system will include a feature to record sequences of movements and play them back for automation.
- User-Friendly Interface: The web interface will be intuitive and visually appealing, making it easy for non-experts to operate.

2. Objectives

- To design and implement a 4-DOF robotic arm controlled by an ESP32 microcontroller.
- To develop a web-based control interface using HTML, CSS, and JavaScript.
- To enable real-time communication between the web interface and the robotic arm using WebSocket protocol.
- To incorporate advanced features like recording and playback for automating repetitive tasks.

3. Hardware Components

- ESP32 Microcontroller: For processing and wireless communication.

- Servo Motors: Four servo motors for controlling the arm’s joints.
- Power Supply: A 5V power supply for stable operation.
- Breadboard and Wires: For prototyping and connecting components.

4. Software Components

- Arduino IDE: For programming the ESP32.
- Web Interface: Developed using HTML, CSS, and JavaScript.
- WebSocket Protocol: For real-time communication between the web interface and ESP32.

5. Expected Outcomes

- A fully functional robotic arm control system with wireless capabilities.
- A user-friendly web interface for controlling the arm.
- Documentation and testing results to validate the system’s performance.

3.3 Methodology

1. Hardware Selection and Setup

- Select the ESP32 microcontroller and servo motors.
- Assemble the hardware components on a breadboard.
- Connect the servo motors to the ESP32’s PWM pins.

2. Software Development

- Write Arduino code to control the servo motors and handle WebSocket communication.
- Develop the web interface using HTML, CSS, and JavaScript.
- Implement the recording and playback feature in the Arduino code.

3. System Integration

- Upload the Arduino code to the ESP32.
- Host the web interface on the ESP32's built-in web server.
- Test the communication between the web interface and the robotic arm.

4. Testing and Validation

- Conduct functional testing for each component.
- Perform integration testing to ensure all components work together.
- Evaluate the system's performance through user testing.

5. Optimization and Improvements

- Optimize the system for responsiveness and accuracy.
- Implement additional features based on user feedback.
- Document the final system and prepare the project report.

Detailed Methodology

The methodology for developing the robotic arm control system was divided into several stages, each focusing on a specific aspect of the project. The first stage involved hardware selection and setup. The ESP32 microcontroller was chosen for its dual-core processor, built-in Wi-Fi capabilities, and compatibility with the Arduino framework. Four servo motors were selected to control the Base, Shoulder, Elbow, and Gripper joints of the robotic arm. A 5V power supply was used to ensure stable operation of the servo motors and ESP32. The hardware components were assembled on a breadboard, with the servo motors connected to the ESP32's PWM pins for precise control.

Once the hardware setup was complete, the focus shifted to software development. The ESP32 was programmed using the Arduino IDE, which provided a user-friendly environment for writing and uploading code. The Arduino code included functions for controlling the servo motors, handling WebSocket communication, and managing the recording and playback feature. The WebSocket protocol was chosen for real-time communication between the web interface and the ESP32, as it allowed for low-latency, bidirectional data transfer. This ensured that commands from the web interface were executed by the robotic arm without noticeable delay. The web interface was developed using HTML, CSS, and JavaScript, with a focus on creating an intuitive and visually appealing control panel. The interface included sliders for adjusting the position of each joint and buttons for recording and playing back sequences of movements. The use of web technologies made the interface accessible from any device with a web browser, enhancing the system's usability.

With the hardware and software components ready, the next step was system integration. The ESP32, servo motors, and power supply were assembled into a single system, and the Arduino code was uploaded to the microcontroller. The web interface was hosted on the ESP32's built-in web server, allowing users to access it by connecting to the ESP32's Wi-Fi network. The system was tested to ensure seamless communication between the web interface and the robotic arm. During this phase, several challenges were encountered, such as servo jitter and latency in WebSocket communication. These issues were addressed by optimizing the code and adjusting the power supply to ensure stable operation.

Testing and validation were critical to ensuring the system's reliability and performance. Functional testing was conducted to verify that each component, including the servo motors, ESP32, and web interface, worked as intended. Integration testing followed, during which the entire system was evaluated to ensure that all components worked together seamlessly. User testing was also conducted to assess the system's ease of use and performance. Feedback from users highlighted the need for additional features, such as error handling and performance enhancements, which were implemented in the final stages of development.

The final phase of the methodology focused on optimization and improvements. Based on testing feedback, the system was optimized to improve responsiveness and accuracy. For example, the servo control algorithm was refined to reduce jitter, and the WebSocket communication protocol was enhanced to minimize latency. Additional features, such as error handling and performance monitoring, were also implemented to ensure the system's reliability in real-world applications. The result was a robust and user-friendly robotic arm control system that met the project's objectives and demonstrated the potential of wireless control and web-based interfaces in robotics.

3.4 Workflow Diagram

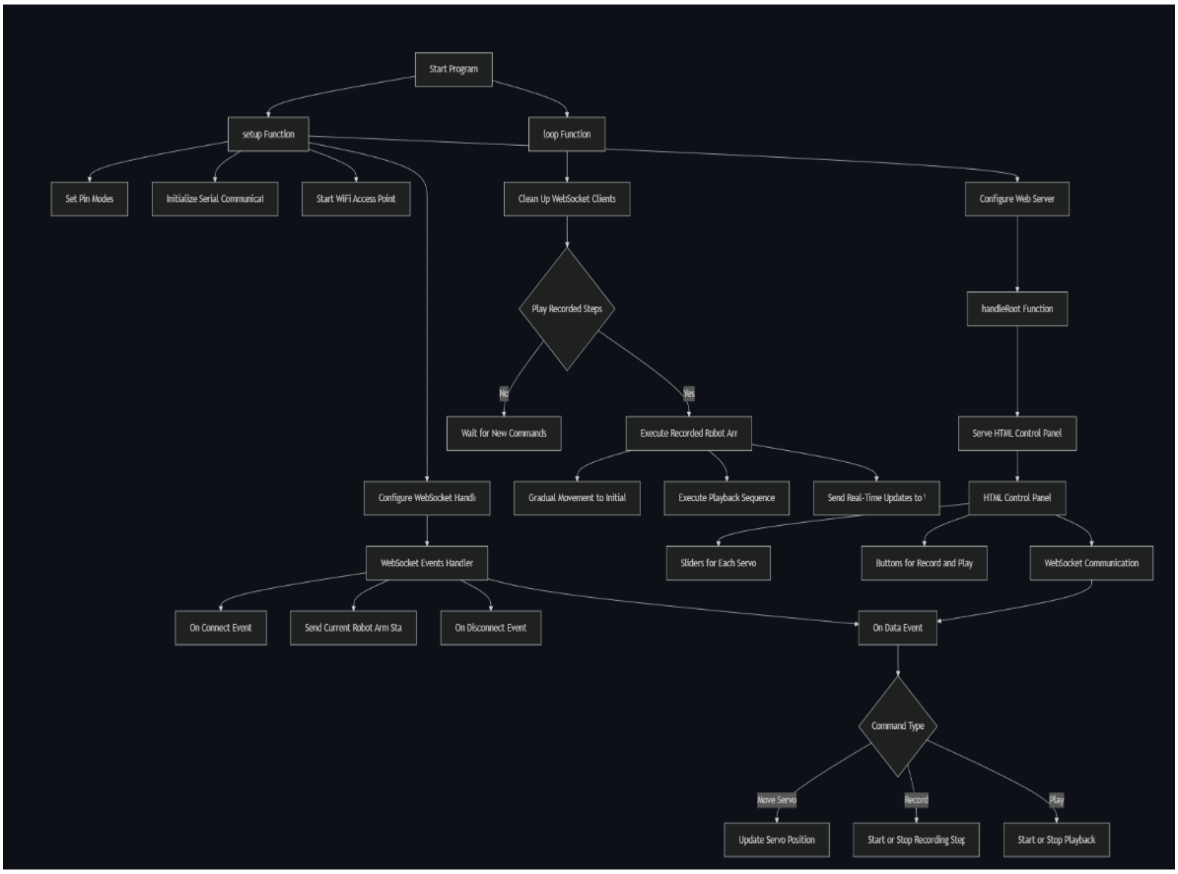

Fig.1 Workflow diagram of the Project

The workflow diagram outlines the key steps in the project's operation. It begins with setting up the Wi-Fi access point and validating clients. The system then configures the server and establishes communication with the HTML control panel. Real-time updates are sent to the control panel, enabling user interaction. The process includes monitoring validated levels, recording and playing movements, and executing payload requests. The workflow ensures seamless communication between the robotic arm and the control interface, facilitating efficient operation and user control.

3.5 Block Diagram

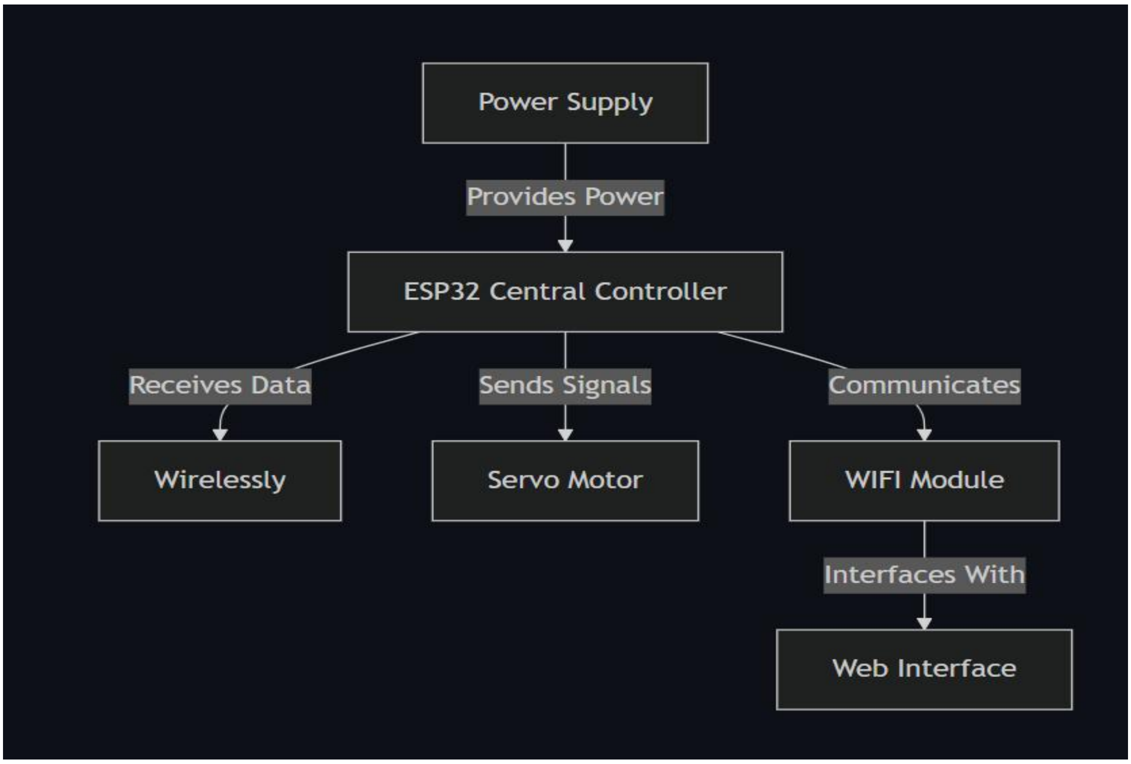

Fig.2 Block diagram of the Project

This diagram illustrates the key components and their interactions in the robotic arm control system. The **Power Supply** provides the necessary energy to run the system. The **ESP32 Central Controller** acts as the brain, receiving data, sending control signals, and managing communication. It connects wirelessly to the **Web Interface** via the **Wi-Fi Module**, enabling remote control. The **Servo Motors** interface with the ESP32 to execute precise movements based on user commands. Together, these components form a cohesive system that allows users to control the robotic arm seamlessly through a web-based interface.

1. **Power Supply**
   - Provides the necessary electrical power to all components, including the ESP32 and servo motors.
   - Ensures stable and consistent operation of the system.
2. **ESP32 Central Controller**
   - Acts as the main processing unit of the system.
   - Receives data from the web interface and sends control signals to the servo

motors.
   - Manages communication between the Wi-Fi module and other components.
3. **Wi-Fi Module**
   - Integrated into the ESP32, enabling wireless communication.
   - Facilitates real-time data exchange between the web interface and the ESP32.
4. **Web Interface**
   - A user-friendly control panel developed using HTML, CSS, and JavaScript.
   - Allows users to control the robotic arm remotely via a web browser.
   - Sends commands to the ESP32 and receives real-time updates.
5. **Servo Motors**
   - Connected to the ESP32 and responsible for moving the robotic arm's joints (Base, Shoulder, Elbow, Gripper).
   - Execute precise movements based on signals received from the ESP32.
6. **Interconnections**
   - The **Power Supply** connects to the ESP32 and servo motors to provide energy.
   - The **ESP32** communicates with the **Web Interface** wirelessly via the **Wi-Fi Module**.
   - The **ESP32** sends control signals to the **Servo Motors** to execute movements.

3.6 User Interface Based on Web

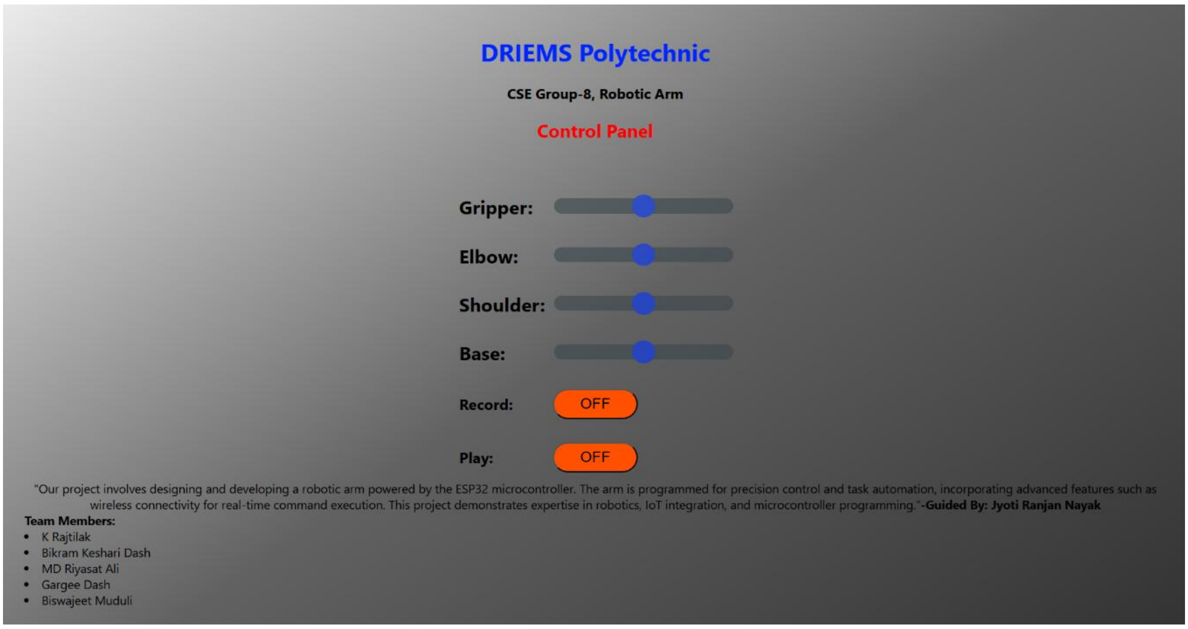

Fig.3 Web Based User Interface

1. **Header Section**
   - Displays the institution name (**DRIEMS Polytechnic**) and project title (**CSE Group-8, Robotic Arm**).
   - Establishes the context and purpose of the interface.
2. **Control Panel Section**
   - Provides sliders for controlling the robotic arm's joints:
     - **Gripper**: Controls the opening and closing of the arm's gripper.
     - **Elbow**: Adjusts the elbow joint's position.
     - **Shoulder**: Manages the shoulder joint's movement.
     - **Base**: Controls the rotation of the base.
   - Each slider allows real-time adjustments, enabling precise control of the arm's movements.
3. **Record and Play Buttons**
   - **Record**: Allows users to record a sequence of movements for later playback.
   - **Play**: Plays back the recorded sequence, automating repetitive tasks.
   - Both buttons have an **OFF** state, indicating their default inactive status.

4. **Project Description**
    - A brief description of the project is provided below the control panel.
    - Highlights key features such as:
        - Precision control and task automation.
        - Wireless connectivity for real-time command execution.
        - Integration of robotics, IoT, and microcontroller programming.
5. **Design and Usability**
    - The interface is clean, intuitive, and user-friendly.
    - Sliders and buttons are clearly labeled, making it easy for users to operate the robotic arm.
    - The use of a web-based design ensures accessibility from any device with a browser.
6. **Visual Appeal**
    - The design incorporates a professional and organized layout.
    - The use of headings, sliders, and buttons enhances the visual appeal and usability of the interface.

This web-based UI design effectively combines functionality and aesthetics, providing users with an intuitive and efficient way to control the robotic arm.

3.7 3D Parts

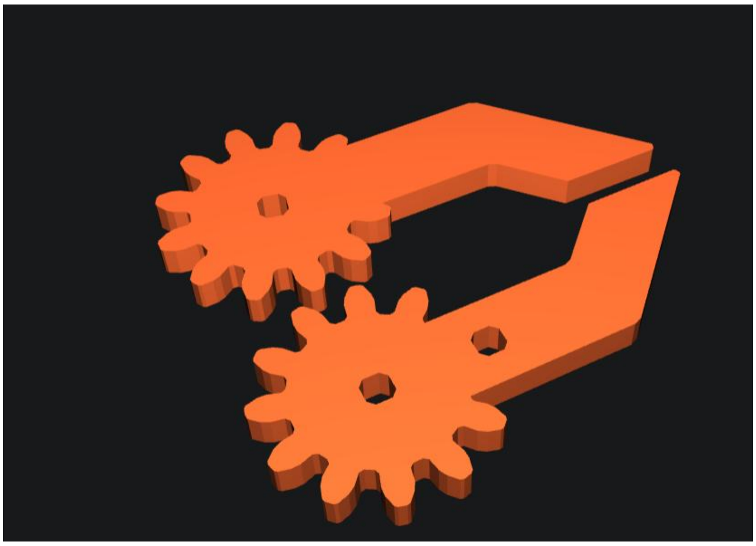

Fig.4 3D Printed Parts

The robotic arm's body is constructed using **3D printing technology**, with **PLA (Polylactic Acid) plastic** as the primary material and **PETG (Polyethylene Terephthalate Glycol)**

filament for added strength and durability. The use of PLA ensures that the body is lightweight, cost-effective, and environmentally friendly, as PLA is derived from renewable resources like cornstarch or sugarcane. PETG, on the other hand, enhances the structural integrity of the arm, providing resistance to impact, heat, and wear, which is essential for the robotic arm's functionality and longevity.

The body has been designed with precision, featuring **net dimensions of 12.7 cm in length, 12.7 cm in width, and 3.8 cm in height**. These compact dimensions make the robotic arm highly portable and suitable for a variety of applications, including educational demonstrations, small-scale automation, and hobbyist projects. The 3D printing process allows for intricate designs and customization, ensuring that the body accommodates all necessary components, such as the servo motors, ESP32 microcontroller, and wiring, without compromising on stability or performance.

The combination of PLA and PETG materials strikes a balance between lightweight construction and robust durability, making the robotic arm both easy to handle and capable of withstanding the stresses of repeated use. This innovative use of 3D printing technology not only reduces manufacturing costs but also demonstrates the potential of additive manufacturing in creating functional and efficient robotic systems.

3.8 Final Model

Fig.5 Final Model View

The final model of the robotic arm is a fully assembled, functional system designed for precision control and automation. It features a **3D-printed body** made from PLA plastic and PETG filament, ensuring a lightweight yet durable structure. The arm is powered by **four servo motors** controlling the Base, Shoulder, Elbow, and Gripper, providing smooth and accurate movements. An **ESP32 microcontroller** serves as the central controller, enabling wireless communication via Wi-Fi for real-time control through a **web-based interface**. The system includes advanced features like **recording and playback** of movements, allowing users to automate repetitive tasks. The compact design, with dimensions of **12.7L x 12.7W x 3.8H cm**, makes it portable and versatile for various applications. The integration of hardware and software ensures seamless operation, while the user-friendly interface enhances accessibility. This project showcases expertise in robotics, IoT, and microcontroller programming, making it a practical and innovative solution for automation and educational purposes.

References

[1] Kurbanhusen Mustafa, S., Yang, G., Huat Yeo, S., Lin, W. and Chen, M., 2008. Self-calibration of a biologically inspired 7 DOF cable-driven robotic arm. Mechatronics, IEEE/ASME Transactions on, 13(1), pp.66- 75.

[2] Kim, H.S., Min, J.K. and Song, J.B., 2016. Multiple-Degree-of-Freedom Counterbalance Robot Arm Based on Slider-Crank Mechanism and Bevel Gear Units. IEEE Transactions on Robotics, 32(1), pp.230-235.

[3] Kim, H.J., Tanaka, Y., Kawamura, A., Kawamura, S. and Nishioka, Y., 2015, August. Development of an inflatable robotic arm system controlled by a joystick. In Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on (pp. 664-669). IEEE.

[4] Mohammed, A.A. and Sunar, M., 2015, May. Kinematics modeling of a 4-DOF robotic arm. In Control, Automation and Robotics (ICCAR), 2015 International Conference on (pp. 87-91). IEEE.

[5] Siciliano, B., 2009. Robotics. London: Springer

[6] Zhang, Z., 2015. Wearable sensor technologies applied for post-stroke rehabilitation (Doctoral dissertation, RMIT University).

[7] Qassem, M.A., Abuhadrous, I. and Elaydi, H., 2010, March. Modeling and Simulation of 5 DOF educational robot arm. In Advanced Computer Control (ICACC), 2010 2nd International Conference on (Vol. 5, pp. 569-574). IEEE.

[8] Bhuyan, A.I. and Mallick, T.C., 2014, October. Gyro-accelerometer based control of a robotic Arm using AVR microcontroller. In Strategic Technology (IFOST), 2014 9th International Forum on (pp. 409-413). IEEE.

[9] Anon, 2016. The Ninth International Symposium. [online] Robotics Research. Available at: http://Robotics Research: The Ninth International Symposium [Accessed 4 Apr. 2016]

[10] Condit, R. and Jones, D.W., 2004. Stepping motors fundamentals. Microchip Application Note: AN907,[Online]. Available: www.microchip.com