UNIVERISTY OF ILORIN

ILORIN, KWARA STATE, NIGERIA

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

TECHNICAL REPORT ON THE STUDENTS' INDUSTRIAL WORK EXPERIENCE SCHEME (SIWES)

UNDERTAKEN AT

THE MECHATRONICS DEPARTMENT OF THE MODEL SKILLS TRAINING CENTER, INDUSTRIAL TRAINING FUND AREA OFFICE, ABUJA

BY

SHAKIR ABDULRAQIB OMOTOSHO

20/30GR080

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF A BACHELOR OF ENGINEERING (B.Eng.) DEGREE IN COMPUTER ENGINEERING

9^{th} OCTOBER 2023 – 15^{TH} DECEMBER 2023

OVERVIEW OF REPORT

The Students' Industrial Work Experience Scheme (SIWES) is an initiative established by the Industrial Training Funds (ITF) and supported by the Federal Government. Its primary objective is to provide tertiary institution students with the necessary skills and exposure to prepare them for the industry in their respective fields of study. Additionally, it offers students the opportunity to familiarize themselves with work methods and gain practical experience in handling equipment and machinery that may not be available within their educational institutions. Another key aim of the program is to facilitate networking among students and professionals from various backgrounds, thereby enhancing their communication skills and ability to interact with people upon graduation. This technical report serves as a comprehensive overview of my experience during the Students' Industrial Work Experience Scheme (SIWES) at the Model Skills Training Center (MSTC) of the Industrial Training Fund (ITF) area office in Abuja (ITF-MSTC). Throughout this program, I had the privilege of working with a range of devices and equipment, including Microcontrollers, Motor drivers, Precision Motors, as well as programming languages and frameworks such as Python, C++, Streamlit, and Artificial Intelligence. This training provided me with the opportunity to learn from and collaborate with certified professionals in the fields of mechatronics and computer engineering, both within the organization and its surrounding environment.

ACKNOWLEDGEMENT

Firstly, I would like to tender my appreciation to God for being there and helping all through my Industrial training experience.

On the second note, my special gratitude goes to the management of ITF-MSTC, Abuja for giving me the opportunity to gain knowledge and immeasurable experiences amidst certified engineers and technocrats and also accommodating me during the course of this attachment, may bless you all more abundantly.

I would also like to tender my gratitude to the engineers I worked with on site and in office starting from Engr. Famsouk, Engr Danlami, Engr Aminu, Engr Mohammed, Madam Ngozi and Mr. Ben. Thank you all for the knowledge you shared with me.

Special gratitude goes to my colleagues at ITF-MSTC that we worked together as a team and learn together through the commitment and devotion, my collaboration with you guys means a lot to me and you will all forever remembered, thank you so much for everything, thank you all.

My last appreciation goes to my amazing parents and my family for their countless support, financially, emotionally, and spiritually. May God keep blessing you more and more, there is no amount of gratitude or reward that can commensurate your deeds, I am forever indebted to you and I sure will make you all proud.

TABLE OF CONTENT

CHAPTER 5: SUMMARY, CHALLENGES, CONCLUSION, AND RECOMMENDATIONS

5.1 Summary of Attachment Activities

5.2 Challenges encountered during the Programme

5.3 Suggestions for Improvement

5.4 Conclusion

CHAPTER ONE: INTRODUCTION

1.1 BACKGROUND OF SIWES

SIWES, an acronym for Students' Industrial Work Experience Scheme, was established in 1973 by Industrial Training Funds (ITF) for the sole purpose of ensuring tertiary institution graduates have the proper skills and experience for employment by the Nigerian Industries.

SIWES was founded to be a skill training program to help expose and prepare students of tertiary institutions such as Universities, Polytechnics, and Colleges of Education, for the industrial work situation to be met after graduation. This scheme serves as a smooth transition from the classroom to the world of work and further helps in the application of knowledge. It provides students with the opportunity of acquiring the skills, exposure, and experience required in effectively managing and handling equipment that is not readily available to them in their respective institutions.

SIWES was birthed out of the growing concern noticed by industrialists that graduates of higher institutions lacked sufficient practical knowledge for employment. This problem is indeed a general one, but it is majorly evident with students who study science and technology having little or no knowledge of the practical aspect of their field hence not suitable for employment.

During the early stage, SIWES was single-handedly financed by Industrial Training Funds (ITF) not until it was too much to bear, then responsibilities were shared between the Industrial Training Funds (ITF) and the Federal Government.

The SIWES scheme duration is generally twenty-six weeks (6 months) although it may also be dependent on the field of study and the Institution. The scheme is mostly done after the first year in Polytechnics that is ND1 and done after the second year or third year in universities depending on the Institution and Course study.

The success and the effective management of this scheme have been due to the cooperation and well-played roles of the Industrial Training Funds (ITF), the Federal Government, and the Supervising agencies among others.

1.2 OBJECTIVES OF SIWES

The objectives of the SIWES program are all about making students industry-ready. Such a program is a successful attempt to help students understand the underlying principles of their future work.

The objectives of SIWES includes;

• Provide an avenue for students in institutions of higher learning to acquire industrial skills and experience relevant to their course of study;

• To prepare specialists who will be ready for any working situations immediately after Graduation

• To teach students the techniques and methods of working with machinery and equipment that may not be available within the walls of an educational institution;

• Make the transition from school to the world of work easier, and enhance students contacts for later job placement;

• To give students the ability to try and apply the given knowledge.

CHAPTER TWO: DESCRIPTION OF THE ESTABLISHMENT OF ATTACHMENT

2.1 LOCATION AND BRIEF HISTORY OF ESTABLISHMENT

The Model Skills Training Centre (MSTC), Abuja, was established by the Industrial Training Fund as a Post-Secondary Technical Institution in 2011, to provide industry relevant career and technical education programme for secondary school leavers and adult learners. It was a two-pronged effort, by the then Federal government, for the actualization of the millennium development goals of Mr. President's policy and Transformation Agenda. To date, the Model Skills Training Centre (MSTC) is progressing in developing the nation’s economy through its highly Skilled Human Resource.

Consequent to the above, the Management of the Industrial Training Fund (ITF), in 2007, undertook a study of three Asian countries with a similar mandate to develop skills. Here they discovered a holistic approach to technical skills development at the Institute of Technical Education (ITEE) in Singapore. To pursue this holistic approach on technical skills development, the ITF adopted a plan of action which culminated in a feasibility study to Abuja by officials of Institute of Technical Education (ITEE), Singapore.

2.2 OBJECTIVES OF ESTABLISHMENT

The main objectives of the training center are to:

• Train and develop high level of skilled manpower in selected occupational areas considered to be of priority to the Commerce and Industry.

• Empower the youths through skills acquisition programme, based on Hands-on, Minds-on and Hearts-on approach aimed at Job creation and entrepreneurship development.

Liaise with the Industries to upgrade the skills of either work force in specific areas as may be identified through performance assessment.

• Certify and issue trainees with Certificates.

2.3 ORGANIZATIONAL STRUCTURE (ORGANOGRAM)

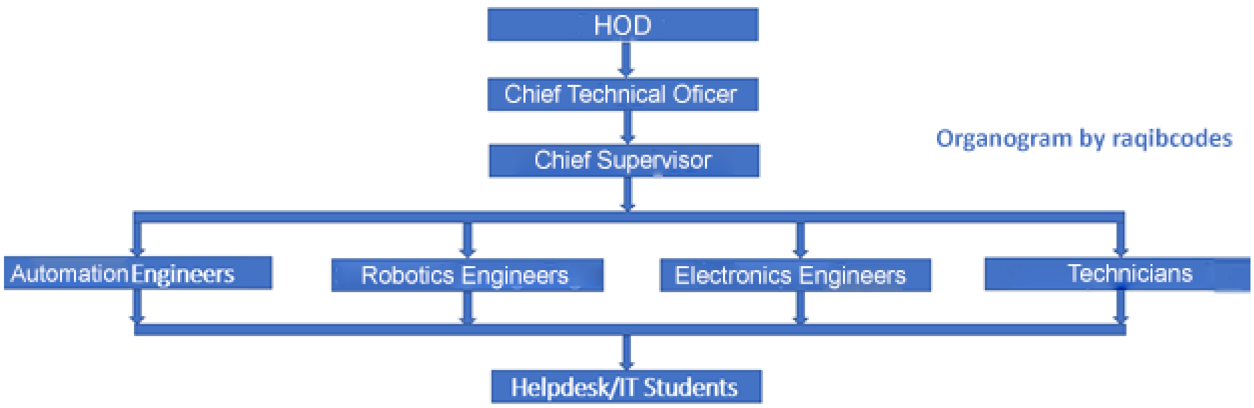

Fig 1. Organogram

2.4 THE DEPARTMENT AND IT'S FUNCTION

THE MECHATRONICS (AUTOMATION AND AUTOTRONICS TECHNOLOGY) DEPARTMENT

Mechatronics is a field of engineering that integrates electrical, electronics, computer, and mechanical engineering technology to design and control modern automated machinery, equipment, and systems. Within the ITF-MSTC Abuja, the Mechatronics department is fully equipped with state-of-the-art tools and equipment for all aspects of mechatronics technology, including robotics and artificial intelligence.

The department is overseen by the following individuals:

1. **Head of Department (HOD)**: The HOD is the most senior member of the department. They are responsible for managing and assigning roles to engineers, technicians, and other members.

2. **Chief Technical Officer (CTO):** The CTO holds the highest technical position in the department. They manage and assign roles to engineers, ensuring the smooth operation of the department.

3. **Chief Supervisor**: The Chief Supervisor is in charge of supervising the day-to-day activities and operations of the department. They also guide visitors through the department's facilities. Additionally, they have jurisdiction over the IT students within the department.

CHAPTER 3: RC CAR PROJECT EXPERIENCE

3.1 INTRODUCTION

The report presented here highlights a crucial stage in my educational development as a computer engineering student, showcasing the valuable industrial experience I gained. This practical involvement allowed me to explore the intricate connection between hardware and software, with a particular focus on assembling and programming an RC Car. The impact of this experience goes beyond the boundaries of a traditional classroom, offering a well-rounded comprehension of the real-world applications of Python and C++ programming languages. In this initial section, my objective is to provide a comprehensive overview, shedding light on the broader context surrounding this experience and emphasizing the significance of the RC Car and programming languages in my Industrial Training experience.

3.2 RC Car

The RC Car, which is an abbreviation for "radio-controlled car," refers to a small-scale vehicle that is operated using a remote control. These diminutive marvels are available in a wide range of shapes and sizes, ranging from miniature cars that can be held in the palm of one's hand to robust monster trucks capable of conquering any type of terrain. The RC Car served as a means of facilitating learning, effectively bridging the divide between theoretical concepts and their practical applications. Its dynamic characteristics enabled the investigation and experimentation of diverse elements, encompassing motors, microcontrollers, and communication modules.

3.2.1 RC Car Assembly and Components

It's crucial to understand the significance of each component used in the RC Car. The careful selection and integration of these components were pivotal in achieving a functional and responsive RC Car system.

1. Jumper Wires
   - Role: Jumper wires served as essential connectors, facilitating the seamless interconnection of various components in the RC Car assembly.
   - Application: They were used to establish electrical connections between the Arduino UNO, motor drivers, and other peripheral components, ensuring a tidy and organized layout.

2. Dual H-Bridge Motor Drivers

- Role: The dual H-bridge motor drivers were crucial for controlling the motors' speed and direction, allowing precise movement of the RC Car.
- Application: Connected to the Arduino UNO, these drivers interpreted signals from the microcontroller to regulate the power supply to the grippy wheels, enabling controlled motion.

3. Arduino UNO
   - Role: The Arduino UNO served as the brain of the RC car, processing commands and controlling various components based on the programmed logic.
   - Application: Programmed in C++, the Arduino UNO received input from the Bluetooth module, processed signals and sent commands to the dual H-bridge motor drivers steering the RC Car.

4. Li-Ion Battery
   - Role: The Li-Ion battery provided a portable and rechargeable power source for the RC car, ensuring sustained operation.
   - Application: Connected to the motor drivers and Arduino UNO, the battery powered the motors and microcontroller, enabling the RC Car's mobility.

5. Ultrasonic Sensors
   - Role: Ultrasonic sensors played a crucial role in the obstacle detection system, enabling the RC Car to sense and respond to its environment.
   - Application: Positioned strategically on the RC Car chassis, the ultrasonic sensors measured distances to detect obstacles, enhancing the navigation capabilities.

6. LED
   - Role: The LED (Light Emitting Diode) served as a visual indicator, providing feedback on the status of certain operations or functions.
   - Application: Incorporated into the project for diagnostic purposes, the LED illuminated based on specific conditions, aiding in debugging and system monitoring.

7. Microcontrollers & Grippy Wheels
   - Role: The microcontrollers and grippy wheels contributed to the overall mobility and control of the RC Car.
   - Application: Microcontrollers processed signals from various sensors, enhancing the RC Car's responsiveness, while grippy wheels ensured traction and stability during movement.

The **assembly process** involved connecting these components in a systematic manner, ensuring secure and functional integration. Jumper wires were meticulously used to establish electrical connections, linking the Arduino UNO to the motor drivers and other critical components. The dual H-bridge motor drivers were interfaced with the grippy wheels, allowing precise control over the RC Car's movements.

The Li-Ion battery served as the power source, providing energy to both the motors and the Arduino UNO. Ultrasonic sensors were strategically positioned to maximize obstacle detection capabilities, enhancing the RC Car's navigation system. The LED, while serving a diagnostic role, also added a visual dimension to the RC Car, signaling specific states or conditions.

3.2.2 Bluetooth Control System

The Bluetooth control system is a pivotal component of the RC Car project, enabling wireless communication and remote-control functionalities. The wireless communication facilitates communication between the RC car and the application used to control the car (on a mobile device) which enables real-time control via Bluetooth signals.

3.2.3 Arduino UNO Programming

- Bluetooth Library: The Arduino UNO, programmed in C++, utilized a Bluetooth library to interface with the chosen Bluetooth module. The library facilitated the establishment of a communication link between a mobile device and the RC Car.
- Communication Protocol: The C++ code on the Arduino UNO implemented a streamlined communication protocol, interpreting user inputs received via Bluetooth. This facilitated the translation of high-level commands into executable actions, ranging from basic movements to more complex maneuvers.

3.2.4 Ultrasonic Sensor Integration

The integration of ultrasonic sensors in the RC Car marked a significant advancement, enhancing the vehicle's navigation capabilities through obstacle detection. Ultrasonic sensors played a crucial role in enabling the RC Car to detect obstacles in its vicinity. By emitting ultrasonic waves and measuring the time taken for their reflection, the sensors provided real-time data on the distance between the car and surrounding objects. The data acquired from the ultrasonic sensors influenced the decision-making process of the RC Car, allowing it to adjust its movements based on the proximity of obstacles and also allowing it to navigate in dynamic environments.

Calibration procedures were also implemented to fine-tune the ultrasonic sensors, addressing the sensitivity and accuracy. This iterative process involved adjusting parameters to optimize the sensors' performance. The algorithm implemented on the Arduino UNO determined the appropriate actions based on the ultrasonic sensor data. It enabled the RC Car to execute maneuvers such as obstacle avoidance, stopping, or altering its trajectory to navigate through tight spaces.

3.2.5 Learnings and Applications

In this section, I explore the key learnings derived from the RC Car project experience, shedding light on the practical applications of the knowledge I acquired in the field of computer engineering.

3.2.5.1 Technical Proficiency

- Hardware Integration Skills: The assembly and integration of components, including ultrasonic sensors, showcased proficiency in hardware integration. Practical skills in connecting sensors, microcontrollers, and motor drivers were honed throughout the project.
- Programming Competence: The project underscored the importance of programming proficiency, especially in C++ for microcontroller logic and control. It provided a platform for refining coding skills in the context of embedded systems.

3.2.5.2 Problem-Solving and Iterative Development

- Challenges Faced: The integration of ultrasonic sensors presented challenges in terms of calibration and real-time responsiveness. The team engaged in systematic problem-solving, iteratively refining the system to overcome hurdles.
- Iterative Development: The project emphasized the iterative nature of development, where continuous testing, debugging, and refinement were essential components of achieving optimal performance.

3.2.5.3 Practical Applications

- Real-World Relevance: The RC Car project offered insights into the real-world applications of computer engineering concepts. The integration of ultrasonic sensors for obstacle detection aligns with broader applications in autonomous vehicles, robotics, and smart systems.
- Interdisciplinary Connections: Beyond technical skills, the project fostered an understanding of the interdisciplinary nature of computer engineering. It highlighted the importance of collaboration between hardware and software aspects to create a fully functional system.

CHAPTER 4: PYTHON AND C++ PROGRAMMING EXPERIENCE

4.1 INTRODUCTION

The industrial experience afforded me an opportunity to navigate through the world of Python and C++ programming. I was able to apply what I had learnt to build real-world projects with practical applications.

4.2 BASIC PYTHON PROGRAMMING

I started with a comprehensive exploration of Python's foundational elements. Delving into loops, data structures, numerical types, and conditional statements provided a robust understanding of basic programming constructs. The mastery of these fundamentals laid the groundwork for the subsequent advanced Python projects, establishing a strong coding foundation.

4.3 ADVANCED PYTHON CONCEPTS & PROJECTS

Following a robust understanding of the basics of Python programming, I was able to incorporate my learnings into projects with real world applications in the field of computer engineering and beyond.

4.3.1 Sentiment Analysis on 300L Computer Engineering Students' Feedback

The first project combined sentiment analysis with a web application, offering a unique perspective on using programming for data-driven insights. I built an Artificial Intelligence (AI) model which analyses sentiment on user-inputted text to detect whether it has an underlying **Positive**, **Neutral** or **Negative** sentiment. The model was trained on feedback data collected from the immediate-past 300 level Computer Engineering students. The model underwent fine-tuning using a **BERT** model and **KerasNLP** techniques, resulting in an impressive accuracy score of 96% in extracting meaningful sentiments from textual data. To further validate and refine this emotional AI, I enlisted the help of **RoBERTa**, a powerful **transformer-based** model renowned for its sentiment analysis prowess. Its performance confirmed the accuracy of my own creation, solidifying its ability to discern the subtle nuances of human emotion hidden within textual data. Completing the project was an intuitive visualization interface, which instantly conveys insights into prevailing student sentiments.

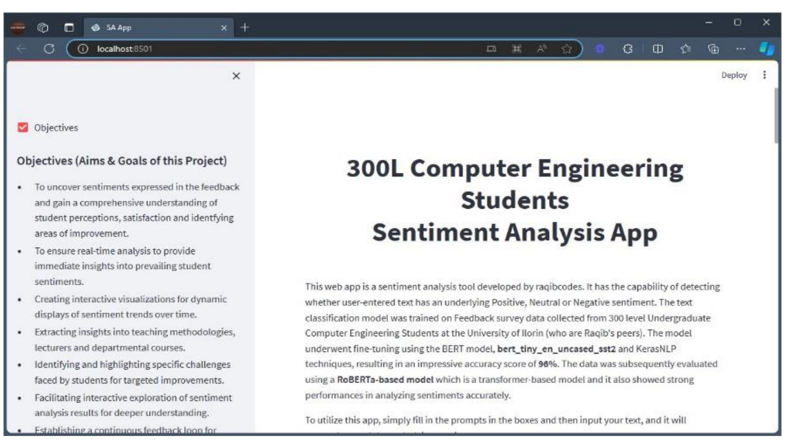

Fig 1: The UI and overview of the App. The objectives and features of the app are also seen in the snapshot.

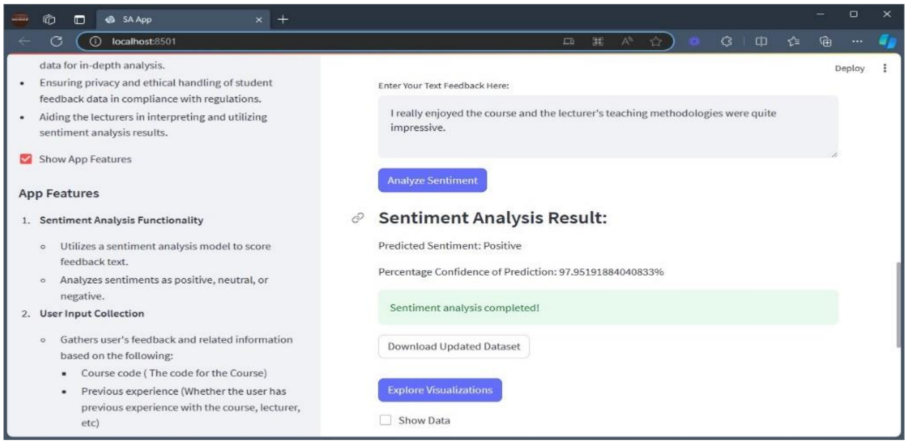

Fig 2: The app's AI model, analyzed the user's inputted text, which reads ***"I really enjoyed the course and the lecturer's teaching methodologies were quite impressive."*** *After the other options had been filled in, it confidently predicted a "Positive" sentiment for the course* ***CPE 331****, with a high confidence level of 98%.*

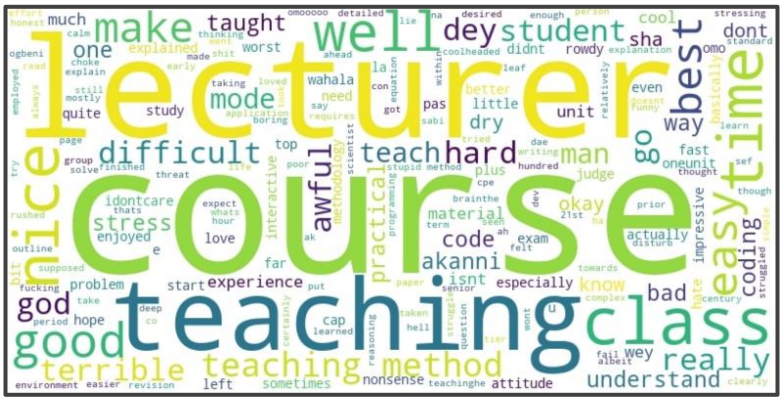

Fig 3: Word Cloud visualization of feedback text data collected from students

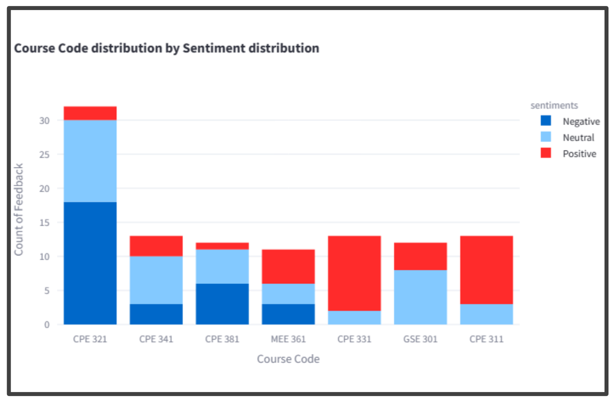

Fig 4: Sentiments across courses

.

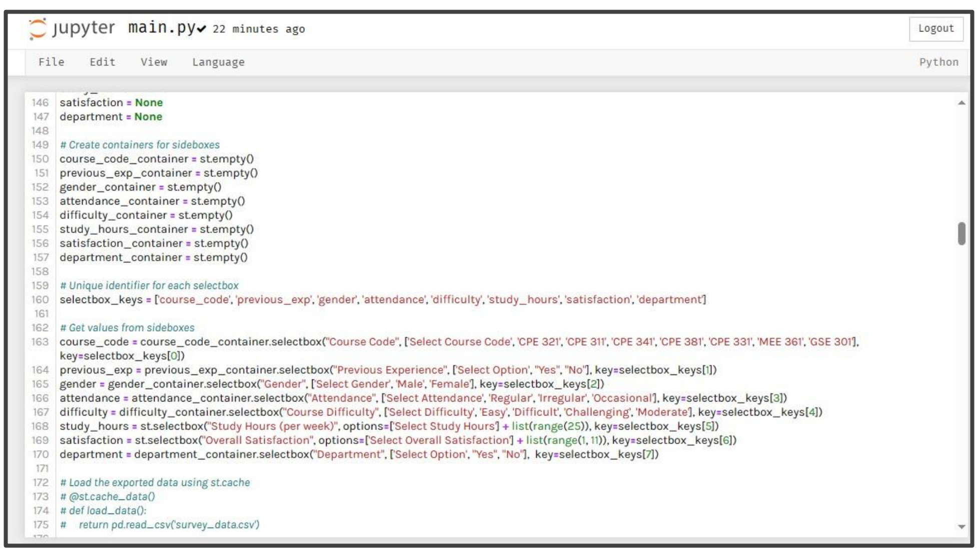

Fig 5: Code snippets of the App and AI model.

Link to Application: https://predictsentiment.streamlit.app/

4.3.2 WhatsApp Chats Analysis Tool

This WhatsApp Chats Analysis tool, developed using Streamlit and Python, leverages Natural Language Processing (NLP) techniques to provide a user-friendly experience. By exploring their chats, users gain valuable insights and a deeper understanding of their messaging history. The tool effortlessly presents compelling statistics, such as information about the most active group members and peak activity times. It takes users on an engaging journey through their messages, bringing them to life through the power of data science. Python's capabilities in creating sophisticated applications with a focus on user experience are showcased through the use of regular expressions for data parsing, the development of an intuitive Graphical User Interface (GUI), and the integration of interactive visualizations. The inspiration for this tool came from my analysis of the department group chat on WhatsApp.

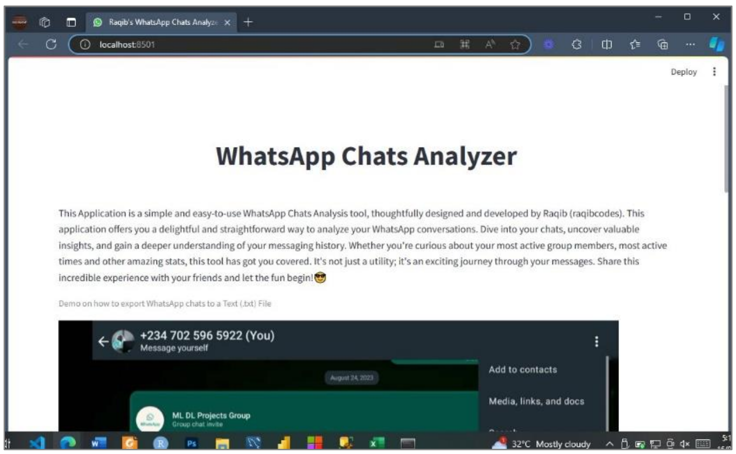

Fig 1: App Overview

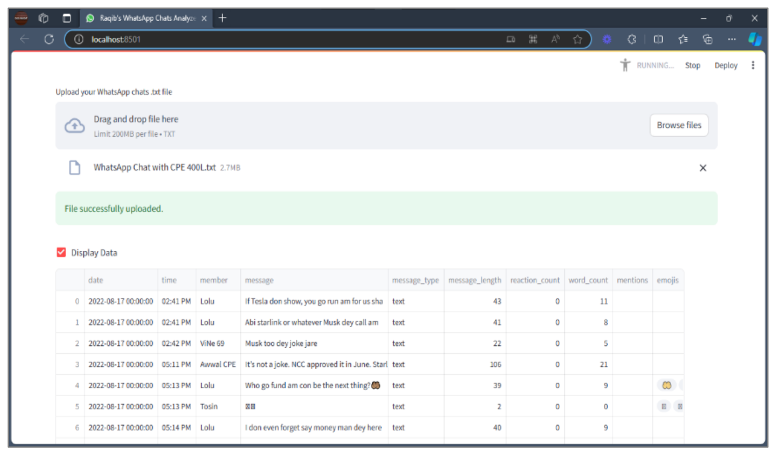

Fig 2: The chats file to analyze, “WhatsApp Chat with CPE 400L.txt” uploaded

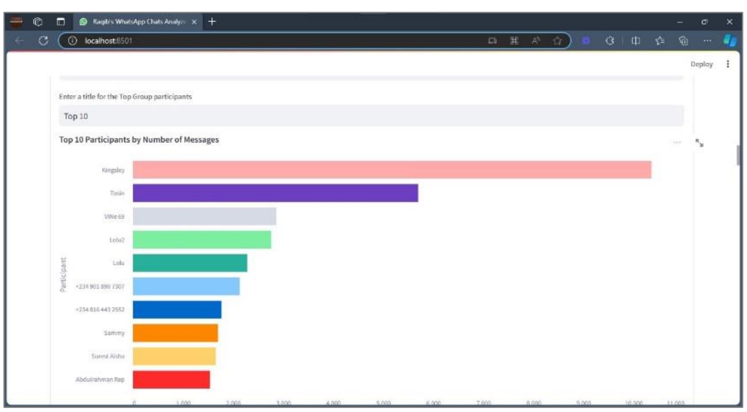

Fig 3: Most Active Group Members analyzed

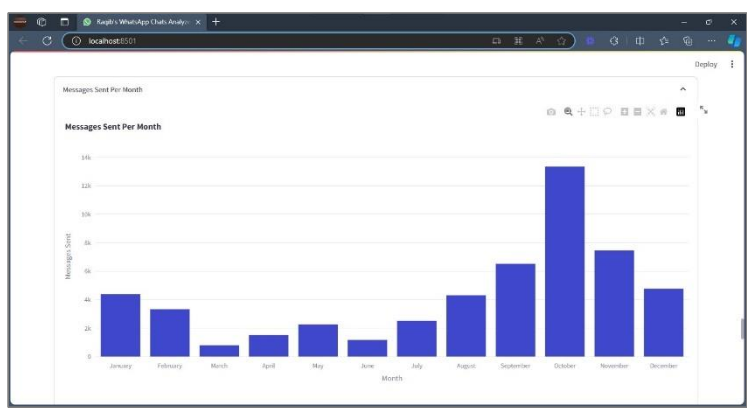

Fig 4: Number of chats messages sent per month.

```
# Create a DataFrame
df = pd.DataFrame(data, columns=['date', 'time', 'member', 'message'])

df['message_type'] = df['message'].apply(lambda x: 'media' if 'attached' in x else 'text')
df['message_length'] = df['message'].apply(len)
df['reaction_count'] = df['message'].apply(lambda x: len(re.findall(r'👍|❤️|😂|😢|😮|😡|🎉|👏', x)))
df['word_count'] = df['message'].apply(lambda x: len(re.findall(r'\b\w+\b', x)))
df['mentions'] = df['message'].apply(lambda x: ', '.join(re.findall(r'@(\w+)', x)))
# df['message'] = df.apply(lambda row: row['message'].replace(f"{row['member']}: ", ""), axis=1)
# df['message'] = df['message'].str.replace(f"{df['member']}: ", "", regex=True)
# df['member'] = df['member'].apply(lambda x: re.escape(x))
# df['message'] = df['message'].str.replace(f"{df['member']}: ", "", regex=True)

# Function to remove member names from messages
def remove_member_names(df):
    for index, row in df.iterrows():
        message = str(row['message'])  # Ensure 'message' is a string
        member = str(row['member'])    # Ensure 'member' is a string
        if message.startswith(member + ': '):
            df.at[index, 'message'] = message[len(member) + 2:]
    return df

# Apply the function to remove member names
df = remove_member_names(df)

# get emojis
def extract_emojis(message):
    return [emoji.emojize(c) for c in message if emoji.is_emoji(c)]

df['emojis'] = df['message'].apply(extract_emojis)
```

Fig 5: Code snippet of App Tool

Link to Application: https://whatsappchats.streamlit.app/

4.3.3 Python, Streamlit & Artificial Intelligence for Audio Transcription (Speech-to-Text)

The exploration of Python, Streamlit and Artificial Intelligence for the development of an audio transcription application showcased the impressive versatility of the language across different domains. Throughout this endeavor, I customized Streamlit components, integrated speech-to-text libraries, and efficiently managed file operations, all of which emphasized the crucial role of Python and AI in creating user-friendly and functional applications.

By applying the knowledge gained from exploring Python, I successfully executed a project that resulted in a dynamic application capable of accurately transcribing audio into text. This application provides users with two convenient options: the ability to upload an audio file for transcription or to directly record audio within the app, receiving real-time speech transcription. This project not only served as an enjoyable experience but also demonstrated the practical application of the skills I acquired, effectively bridging theoretical learning with tangible real-life solutions.

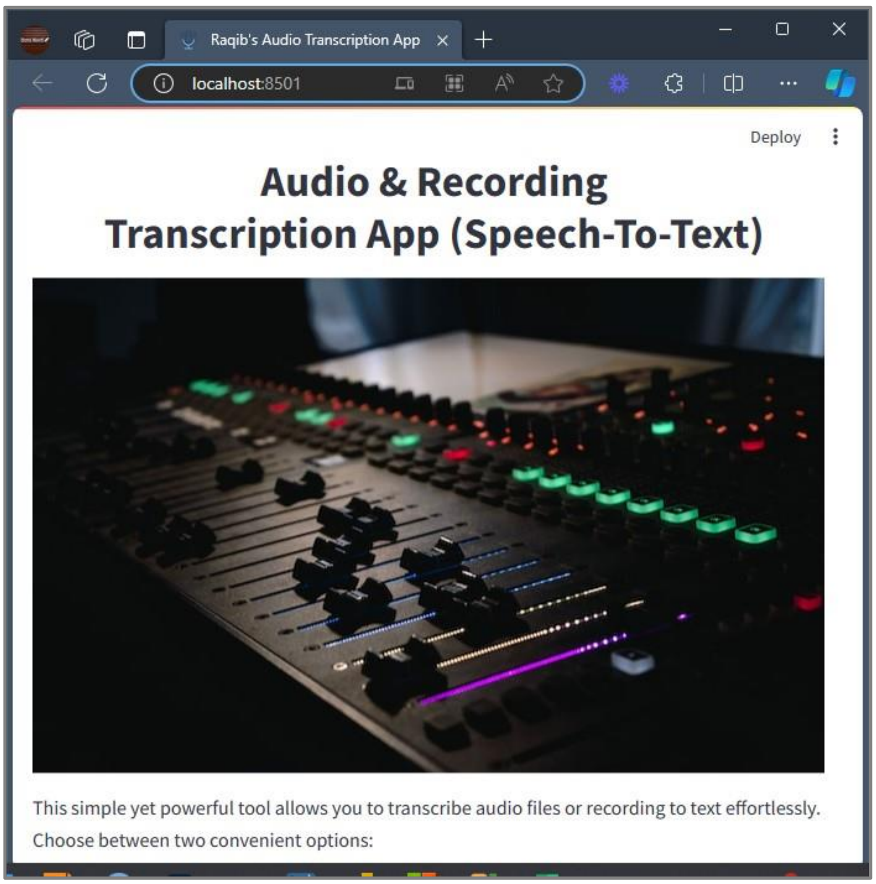

Fig 1: The App's User Interface

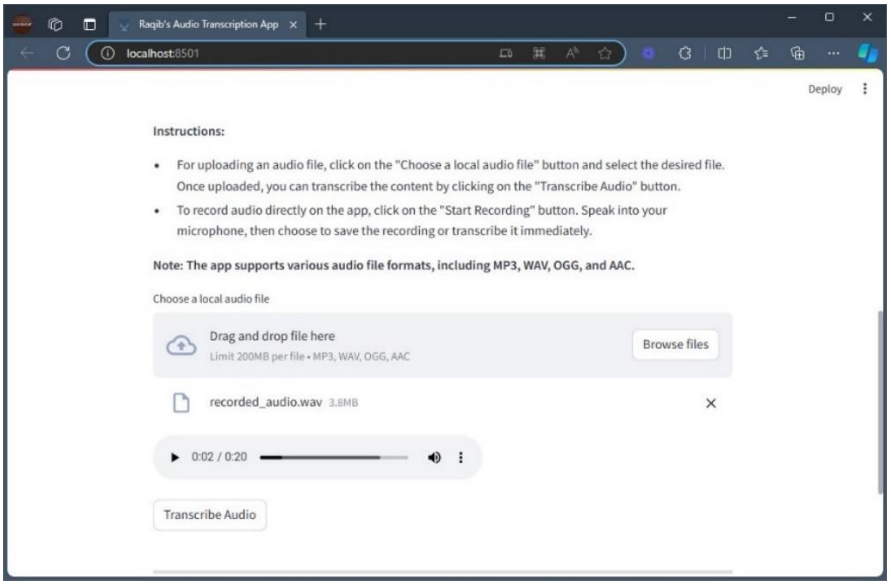

Fig 2: The audio file to be transcribe, "recorded_audio.wav" uploaded and ready to be transcribed using the AI technology.

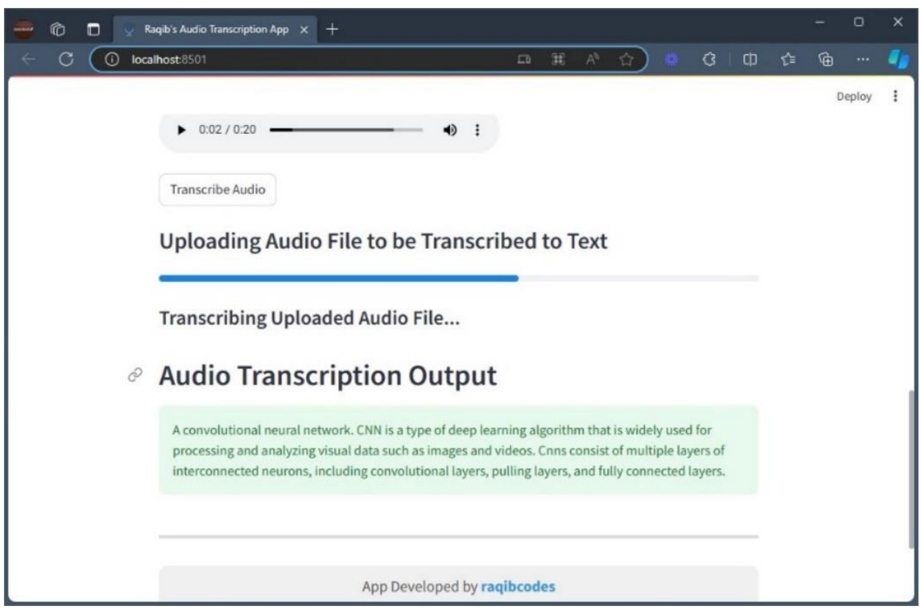

Fig 3: Transcription Results of the Audio file.

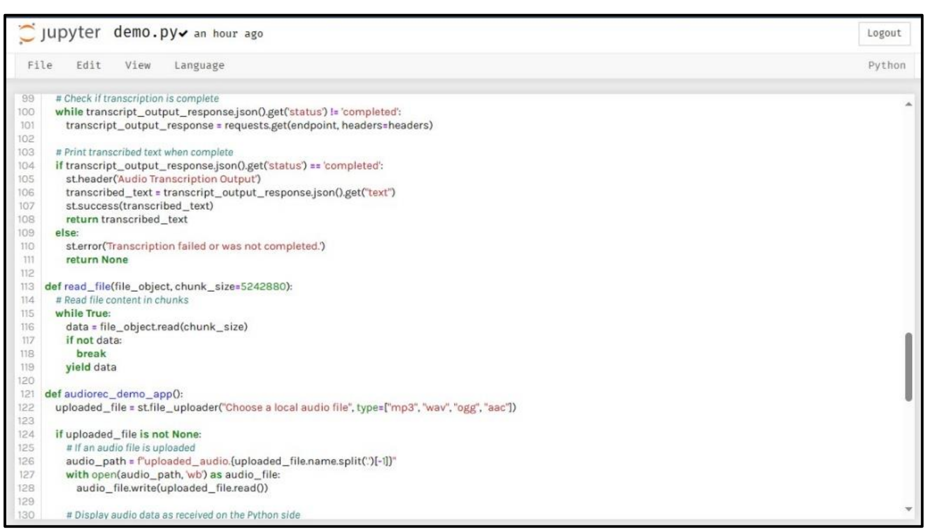

Fig 4: Code for implementing the audio transcription leveraging AI technology.

Link to Application: https://audiotranscription.streamlit.app/

4.4 C++ in Microcontroller Programming

The utilization of C++ within the RC Car project proved to be essential for microcontroller programming. The C++ code played a vital role in governing the movements of the RC Car, as it interpreted signals received from the Bluetooth module and enabled real-time decision-making. This successful integration showcased the practical implementation of C++ in embedded systems and control programming.

Below is the C++ code that controls the RC car;

```
// Define motor control pins
const int leftMotorForwardPin = 13;
const int leftMotorReversePin = 12;
const int rightMotorForwardPin = 11;
const int rightMotorReversePin = 10;
const int ledPin = 9;
char command;

void setup() {
  // Set pin modes in the setup
  pinMode(leftMotorForwardPin, OUTPUT);
  pinMode(leftMotorReversePin, OUTPUT);
  pinMode(rightMotorForwardPin, OUTPUT);
  pinMode(rightMotorReversePin, OUTPUT);
pinMode(ledPin, OUTPUT);

  // Start Serial communication
  Serial.begin(9600);
}
```

```
void loop() {
  // Check if there is data available on Serial
  if (Serial.available()) {
    command = Serial.read();
    Serial.println(command);
    executeCommand(command);
  }
}

void executeCommand(char cmd) {
  // Perform motor control based on received command
  switch (cmd) {
    case 'F':
      moveForward();
      break;
    case 'B':
      moveBackward();
      break;
    case 'L':
      turnRight();
      break;
    case 'R':
      turnLeft();
      break;
    case 'W':
      turnLedOn();
      break;
    case 'w':
```

```
      turnLedOff();
      break;
    case 'S':
      stopMotors();
      break;
    default:
      // Handle unknown commands
      break;
  }

  // Introduce a delay for smoother operation
  delay(100);
}

// Helper functions for different commands

void moveForward() {
  digitalWrite(leftMotorForwardPin, HIGH);
  digitalWrite(rightMotorForwardPin, HIGH);
}

void moveBackward() {
  digitalWrite(leftMotorReversePin, HIGH);
  digitalWrite(rightMotorReversePin, HIGH);
}

void turnRight() {
  digitalWrite(leftMotorForwardPin, HIGH);
```

```
}

void turnLeft() {
  digitalWrite(rightMotorForwardPin, HIGH);
}

void turnLedOn() {
  digitalWrite(ledPin, HIGH);
}

void turnLedOff() {
  digitalWrite(ledPin, LOW);
}

void stopMotors() {
  digitalWrite(leftMotorForwardPin, LOW);
  digitalWrite(leftMotorReversePin, LOW);
  digitalWrite(rightMotorForwardPin, LOW);
  digitalWrite(rightMotorReversePin, LOW);
}
```

Functionality of the Arduino Code

The Arduino code serves as the control logic for the RC car, defining the actions triggered by specific commands received via serial communication. In the setup phase, the code initializes the pin modes for motor control and an LED indicator, setting the groundwork for subsequent operations. Serial communication is established to facilitate communication between the Arduino microcontroller and an external device, such as a mobile application, allowing users to send commands wirelessly.

Within the main loop, the code constantly checks for incoming data on the serial port. Upon detecting a command, it is read and printed to the serial monitor for monitoring purposes. The executeCommand function is then invoked to interpret and execute the received command. The switch-case structure within executeCommand directs the flow based on the specific command received. ***For example, 'F'***

triggers both motors to move forward, 'B' initiates backward movement, 'L' turns the car right, 'R' turns it left, 'W' activates the LED, 'w' deactivates the LED, and 'S' stops all motors.

To implement these actions, the code employs helper functions for each distinct command. For instance, moveForward simultaneously activates the forward pins for both left and right motors, facilitating forward motion. Similarly, turnRight and turnLeft control the motor pins to enable corresponding turns, while turnLedOn and turnLedOff manipulate the LED pin for on and off states, respectively. The stopMotors function ensures that all motor pins are set to a low state, bringing the RC car to a complete halt.

To enhance the operational smoothness, a delay of 100 milliseconds is introduced after each command execution. This C++ code encapsulates the fundamental control logic for the RC car, providing a clear and modular structure for handling different commands and ensuring the seamless integration of hardware components.

CHAPTER 5: SUMMARY, CHALLENGES, CONCLUSION, AND RECOMMENDATIONS

5.1 SUMMARY

During my time at ITF-MSTC, Abuja, I had the opportunity to delve into various aspects of computer engineering. One of the highlights was working on an RC car, where I gained hands-on experience in both assembly and programming. Throughout the weeks, I had the chance to explore Bluetooth and ultrasonic sensor-controlled models, while also honing my skills in C++ and Python programming. These skills were then applied to real-world projects, allowing me to see the practical applications of what I had learned. This chapter of my experience also focused on deepening my understanding of Python and its advanced problem-solving capabilities, as well as integrating C++ into microcontroller programming. Overall, this hands-on experience greatly enhanced my mastery of these valuable skills.

5.2 CHALLENGES FACED DURING THE TIME OF TRAINING

Throughout the training period, several challenges were encountered. Notable ones included the problems of Bluetooth communication in the initial stages, requiring dedicated troubleshooting and collaborative problem-solving. The integration of ultrasonic sensors posed calibration and real-time responsiveness challenges, demanding iterative development and testing. Additionally, the presence of obsolete equipment posed obstacles in obtaining accurate results.

These challenges, however, served as valuable learning opportunities, fostering problem-solving skills and a deeper understanding of hardware-software interactions.

5.3 SUGGESTIONS

Considering the difficulties encountered, it is recommended to improve the training modules by placing greater emphasis on resolving Bluetooth communication problems and offering more detailed instructions on integrating ultrasonic sensors. Furthermore, it is advisable to allocate additional funds to the department in order to acquire modern equipment, while also considering the possibility of repairing and refurbishing any outdated equipment that is not completely damaged. Moreover, the inclusion of interactive problem-solving sessions or collaborative projects would enhance the learning experience and promote teamwork.

5.4 CONCLUSIONS

The SIWES training program has had a profound impact on me, transforming my perspective and providing me with practical engineering experience. Through this opportunity, I have gained invaluable technical knowledge that has played a crucial role in my professional growth. From hands-on tasks like assembling and programming RC cars to exploring programming languages such as Python and C++, the program has given me a comprehensive understanding of the interplay between hardware and software.

Moreover, this experience has allowed me to forge meaningful connections and network with seasoned experts and professionals in the field. Learning from their vast experience has been instrumental in my personal development throughout this program. The practical applications I have undertaken, the challenges I have successfully overcome, and the collaborative problem-solving I have engaged in have all contributed to the cultivation of a strong skill set in the field of computer engineering.

5.5 RECOMMENDATIONS

As a recommendation, future training programs could explore advanced topics in robotics and autonomous systems, allowing students to delve deeper into the integration of sensors, actuators, and complex algorithms. Additionally, incorporating industry-relevant projects and exposure to emerging technologies could further prepare students for the evolving landscape of computer engineering.