Big Data and IoT Lab (CSE413)

Lab Report

Submitted to-

Dr. Fizar Ahmed
Associate Professor

Department of Computer Science and Engineering
Faculty of science and Information Technology (FSIT)
Daffodil International University

Submitted by-

Farhan Tanvir	**Mahmudul Hasan Khan**	**Immon Abeer**
211-15-14668	**201-15-13791**	**211-15-14662**

58th batch, Section-B, CSE Department, DIU.

Smart Attendance System

Abstract:

The Smart Attendance System represents a groundbreaking application of Internet of Things (IoT) technology aimed at revolutionizing traditional attendance management systems. By integrating various sensors, microcontrollers, and web-based interfaces, the system offers a seamless and efficient solution for tracking and managing attendance in educational institutions and other relevant settings. This report provides a comprehensive overview of the project, detailing its equipment, environment setup, circuit diagram, data collection procedure, project implementations, and concluding remarks.

Attendance management is a critical aspect of organizational operations, particularly in educational institutions. Traditional methods of manual attendance tracking are not only time-consuming but also prone to errors and manipulation. The Smart Attendance System addresses these challenges by harnessing IoT technology to automate the attendance process, thereby enhancing accuracy, efficiency, and reliability.

Equipment(s):

The Smart Attendance System utilizes the following IoT sensors and components:

1. **NodeMCU ESP8266:** This versatile microcontroller serves as the central processing unit of the system, facilitating communication between sensors and the web-based interface.
2. **MFRC522 RFID Reader:** Utilized for detecting RFID tags embedded in student ID cards, allowing for seamless identification of individuals.
3. **Servo Motor:** Employed to control physical access points such as gates or barriers, enabling automated entry upon successful attendance verification.
4. **HC-SR04 Ultrasonic Sensor:** Measures distance to ensure accurate detection of individuals within the proximity of the attendance interface.
5. **Piezo Buzzer:** Provides audible feedback to indicate the status of RFID card readings, sounding different tones for successful or declined readings.

Environment Setup:

The setup process involves configuring platforms, writing codes, and establishing connectivity with IoT devices. The following tools were employed:

- **Arduino IDE:** A versatile development environment used for programming the hardware components, including the NodeMCU ESP8266, Servo Motor, MFRC522 RFID Reader and HC-SR04 Ultrasonic Sensor.

- **XAMPP:** This software stack provides a local server environment for hosting the web-based attendance management system. PHP scripting language and MySQL database management system are utilized for dynamic web functionality and data storage.

Setting up the environments involves configuring the necessary software tools and platforms to ensure smooth operation of the Smart Attendance System. Here's a detailed guide on how to set up each environment:

- **Arduino IDE Setup:**

1. Download and install the latest version of Arduino IDE from the official Arduino website.
2. Open Arduino IDE and navigate to File > Preferences.
3. In the "Additional Board Manager URLs" field, add the following URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json
4. Click OK to save the preferences.
5. Navigate to Tools > Board > Boards Manager.
6. Search for "ESP8266" and install the ESP8266 board package.
7. Select NodeMCU 1.0 (ESP-12E Module) from Tools > Board menu.
8. Ensure the correct COM port is selected from Tools > Port menu.

- **XAMPP Setup:**

1. Download the latest version of XAMPP from the Apache Friends website.
2. Run the installer and follow the on-screen instructions to install XAMPP on your system.
3. Launch the XAMPP Control Panel and start the Apache and MySQL services.
4. Open a web browser and navigate to http://localhost/phpmyadmin.
5. Create a new database for the Smart Attendance System by clicking on the "New" button and entering the database name "rfidattendance".
6. Import the provided SQL script to create the necessary tables and structure for the database.

- **Web Program Setup:**

1. Create a new directory within the XAMPP web server directory (e.g., C:\xampp\htdocs\rfidattendance).
2. Copy the PHP files for the web-based attendance management system into the newly created directory.

3. Access the web-based interface by navigating to http://localhost/rfidattendance in a web browser.

- **Arduino Program Setup:**

1. Launch the Arduino IDE and open the "RFID_Attendance.ino" file. This action loads the code into the IDE environment, allowing for further customization and uploading.
2. Within the code, locate the designated sections for configuring your WiFi credentials. Here, you'll specify your WiFi SSID (Service Set Identifier) and password. This step enables the NodeMCU ESP8266 to connect to your wireless network seamlessly.

```
const char* ssid = "YourWiFiSSID";

const char* password = "YourWiFiPassword";
```

   Replace "YourWiFiSSID" with your network's name (SSID), and "YourWiFiPassword" with your network password.

3. Access the web-based interface associated with your RFID system. Here, you'll generate a unique device token. Upon generation, copy the token provided by the interface.
4. Navigate back to the Arduino IDE. Locate the section in the code specifically designated for the device token. This is where you'll paste the token you generated in the previous step.

```
const char* deviceToken = "YourDeviceToken";
```

5. With all configurations set, connect your NodeMCU ESP8266 to your computer using a USB cable. Then, select the appropriate board and port from the Arduino IDE's "Tools" menu.
6. Finally, initiate the upload process by clicking the "Upload" button within the Arduino IDE. This action compiles the code and transfers it to the NodeMCU ESP8266, making it operational within your RFID attendance system.

By following these steps, you'll have successfully set up the Arduino IDE for programming the hardware components and configured XAMPP to host the web-based attendance management system. This robust environment setup lays the foundation for seamless integration and operation of the Smart Attendance System.

Circuit Diagram:

The circuit diagram depicts the interconnectedness of the hardware components, illustrating how signals flow between sensors, microcontrollers, and actuators. It provides a visual representation of the system architecture, aiding in the understanding and implementation of the project.

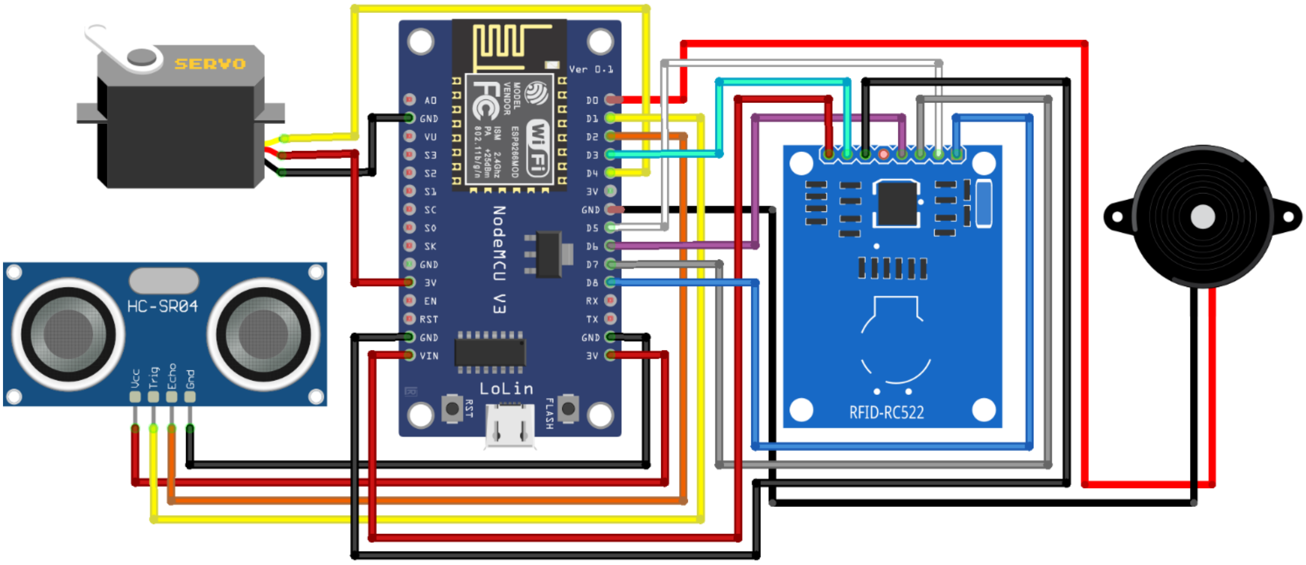

Figure: Circuit Diagram of Smart Attendance System

Data Collection Procedure:

The data collection process initiates when a student taps their ID card on the RFID scanner. The RFID sensor promptly reads the unique identifier embedded in the card's RFID tag and transmits this information to the NodeMCU ESP8266 microcontroller. The NodeMCU, functioning as the central data processing unit, processes the received data and subsequently forwards it to a web-based interface hosted on a XAMPP server.

Administrators can seamlessly access this intuitive interface to manage a variety of tasks. These include viewing real-time attendance records, adding or removing student entries, configuring additional RFID scanners for expanded coverage, and executing other critical administrative functions.

This streamlined system not only simplifies the attendance tracking process but also provides a robust platform for administrative efficiency and data-driven decision-making.

Project Implementations:

The implementation process involves several stages:

1. **Hardware Setup:** The hardware components are assembled according to the circuit diagram, ensuring proper connections and functionality.
2. **Software Development:** Code is written and uploaded to the NodeMCU ESP8266 using the Arduino IDE to enable communication with sensors and data transmission to the web-based interface.
3. **Web Interface Development:** The web-based attendance management system is developed using PHP scripting language and MySQL database management system. The interface allows administrators to interact with the system, view attendance records, and perform administrative tasks.
4. **Testing and Deployment:** The system is rigorously tested for functionality, accuracy, and reliability. Once deemed ready, it is deployed for real-world usage in educational institutions or other relevant settings.

Conclusion:

The Smart Attendance System represents a significant advancement in attendance management technology, offering a sophisticated yet user-friendly solution for automating the attendance process. By leveraging IoT technology, the system streamlines operations, reduces manual effort, and enhances overall efficiency. Its potential applications extend beyond educational institutions to various industries where attendance tracking is essential. Moving forward, further enhancements and refinements can be made to optimize performance and expand functionality, making the Smart Attendance System a valuable asset in modern organizational management.

References:

1. Steps to Setup Arduino IDE for NODEMCU ESP8266 [https://www.instructables.com/Steps-to-Setup-Arduino-IDE-for-NODEMCU-ESP8266-WiF]
2. MFRC522 RFID Reader Interfaced with NodeMCU [https://www.instructables.com/MFRC522-RFID-Reader-Interfaced-With-NodeMCU/]
3. Interfacing Servo Motor with NodeMCU [https://www.instructables.com/Interfacing-Servo-Motor-With-NodeMCU]
4. Distance Measurement Using HC-SR04 Via NodeMCU [https://www.instructables.com/Distance-Measurement-Using-HC-SR04-Via-NodeMCU]