Attendance System

Under the supervision of:

Dr. Lamyaa El-Refaai Dr. Tarek Elewa

1 – Shady bahaa El-kiss

2 – Mohy El-din Ehab

3 – Marwan Mohamed Shaaban

4- Mostafa Shokry Elkamel

5- Mostafa Abdallah Mostafa

1- Abstract:

The rapid advancements in Internet of Things (IoT) technology have paved the way for innovative solutions in various domains, including attendance management systems. This research abstract presents an IoT-based attendance system that seamlessly integrates an embedded system, a mobile application, and a backend database to efficiently monitor and manage student attendance. The proposed system leverages an embedded system architecture utilizing the ESP32 microcontroller and RFID (Radio Frequency Identification) technology. The ESP32 acts as a central hub, receiving attendance data from RFID scanners placed at entry points. This data is then processed, validated, and transmitted to the backend system for storage and analysis. To provide users with real-time access to attendance information, a mobile application is developed, allowing students and administrators to view attendance records, generate reports, and receive notifications. The mobile application interfaces with the embedded system through a secure network connection, ensuring seamless data flow and user privacy. The backend system operates on a robust and scalable database, housing all student attendance data. This enables administrators to efficiently manage and retrieve records, generate comprehensive reports, and perform data analysis for various purposes, such as identifying trends and patterns. To ensure data integrity and security, the system incorporates FTP (File Transfer Protocol) for secure data transmission between the embedded system and the backend database. This guarantees the confidentiality and integrity of attendance data during transit. The IoT attendance system offers several benefits, including real-time attendance monitoring, reduced administrative workload, and improved accuracy in record-keeping. By leveraging IoT technology, this system provides a scalable and adaptable solution for educational institutions seeking to streamline attendance management processes. Future work includes expanding the system's capabilities by integrating additional features such as facial recognition or biometric authentication for enhanced security and accuracy. Moreover, ongoing research aims to optimize the system's performance and explore potential integration with other educational systems to further enhance operational efficiency.

Keywords: Internet of Things, attendance system, embedded system, ESP32, RFID, mobile application, backend database, data management, data analysis, FTP.

2- Introduction:

Problem : Traditional attendance systems:

In colleges and universities, attendance tracking is an important task that helps to monitor student's academic progress and ensure that they are meeting the required attendance criteria. Traditionally, this has been done through manual attendance systems, where professors and teachers take roll calls and mark attendance on a paper register or excel sheet. However, these manual systems have several disadvantages that can lead to errors, inefficiencies, and ultimately impact student's academic performance.

One of the main disadvantages of manual attendance systems is the potential for errors in recording attendance. Teachers and professors may make mistakes while taking roll calls or recording attendance, leading to inaccuracies that can impact student's attendance records. Moreover, students may intentionally or unintentionally mark their attendance incorrectly, which can result in incorrect attendance records.

Manual attendance systems also require a significant amount of time and effort from instructors, who have to take roll calls for each class, calculate attendance percentages, and update attendance records manually. This can be a time-consuming and tedious process, taking away valuable teaching time that could be better utilized in delivering lectures and conducting activities that improve student learning.

Another disadvantage of manual attendance systems is the lack of real-time tracking and reporting. Since attendance records are updated manually, there is often a delay in updating the attendance records, making it difficult to track a

student's attendance in real-time. This can make it challenging to identify students who are frequently absent and take corrective action.

Finally, manual attendance systems can be prone to manipulation and fraud. Students can easily forge attendance records or have someone else mark their attendance, which can result in inaccurate attendance records.

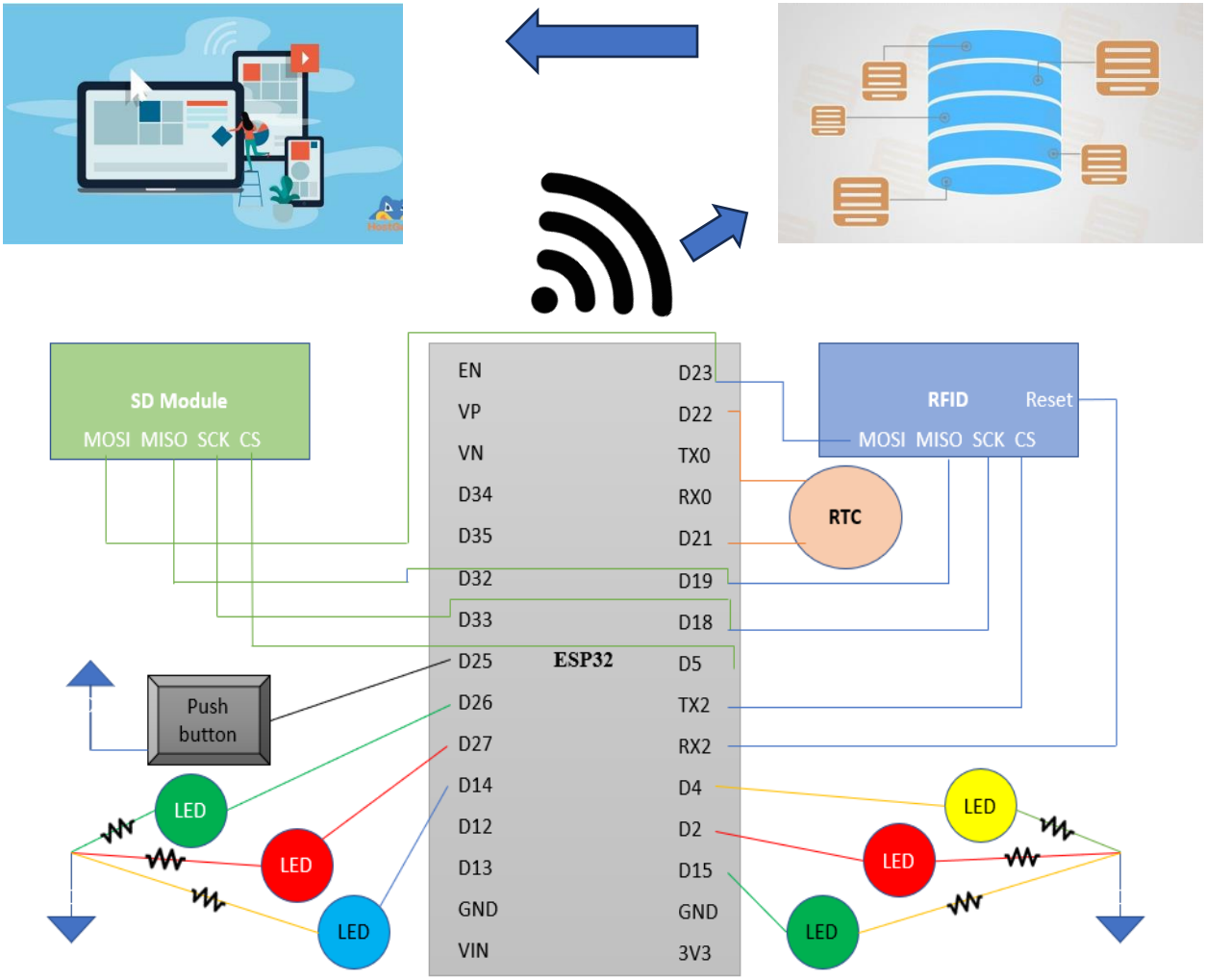
Solution: Automatic attendance system:

As a result of the previous problems, there is an obvious need for a more controlled and efficient system in terms of time, effort and accuracy. here it comes our automatic attendance system. First of all, to avoid all problems with the traditional systems, we focused on identifying the criteria that can make the system a solution for the preceded problems:

- **Automating Attendance Recording:** RFID technology streamlines the process, eliminating manual recording and reducing instructor workload.
- **Enhancing Accuracy:** RFID tags provide unique identification, minimizing errors and fraud compared to manual methods.
- **Enabling Real-Time Data Access:** The mobile app offers instant attendance updates, allowing for immediate intervention and support.
- **Secure Data Storage and Retrieval:** The backend database securely stores and manages attendance records, facilitating easy access and analysis.
- **Improved Data Integrity:** Secure data transmission protocols protect against unauthorized access and manipulation, ensuring data confidentiality.

By addressing these critical needs, our project aims to improve the efficiency, accuracy, and transparency of attendance management in educational institutions, ultimately benefiting students, instructors, and administrators alike.

3- System architecture:



The idea of the system was to send an automatic excel sheet that resembles the attendance of a specific lecture through a network to the central server in the college, then the data stored in a database accessible by user interface (website/ mobile/ desktop). Excel sheets are filled according to unique IDs identified by some RFID cards so that when a student put his card on the device, his ID is registered in the excel sheet with the timestamp the process was applied at. Sending files to the server is done by a push button. Once there is someone who pushes the button, the last file created is uploaded to the server through FTP protocol and saved there inside a folder with the name of today's date and the file is named by the hour and minute in which the button pressed. And the data is automatically updated on the website, application, and database. During the whole process, an SD card holds the files as permanent storage that can be emptied at any

time. The whole system is controlled by ESP32 microcontroller and programmed by C/C++.
Based on that we designed a block diagram that represents the connections between the microcontroller (ESP32) and the different components as shown in the above figure

The block diagram consists of various components:

- **Embedded System**
  - **RFID module**: it is a radio frequency identification system that consists of two main components, a tag attached to the object to be identified, and a reader that reads the tag. A reader consists of a radio frequency module and an antenna that generates a high frequency electromagnetic field. When cards are written with some IDs, RFID is able to identify which ID belongs to every card. The module works with SPI communication protocol.

  - **SD module**: The SD card module is especially useful for projects that require data logging. ESP32 can create a file in an SD card to write and save data using the SD library. There are different models from different suppliers, but they all work in a similar way, using the SPI communication protocol.

  - **Push button**: a mechanical button when it is pressed one time, all the files in the SD card will be sent to the central server.

  - **RTC**: The real-time clock keeps track of seconds, minutes, hours, days, dates, months, and years. For months with less than 31 days, the date at the end of the month is automatically modified, including leap year corrections. The clock has an AM/PM indication and works in either a

24-hour or 12-hour mode. It is useful in creating folders with right date and recording the timestamps at which students record their attendance.

- **Database**: based on SQL and has the following data relations:
  - **Department table**: describes the ID and name of the departments.
  - **Student table**: contains students' data (name – national id – email – academic year – department).
  - **Instructor table**: contains instructors' data (name – national id – email - position)
  - **Course table**: contains courses' data (name – year – semester)
  - **Enroll table**: specifies which courses students are enrolled in
  - **Teach table**: specifies which courses instructors teach
  - **Attend table**: specifies the dates of attendance for each student in each course
  - **Course_schedule table**: defines the dates of courses
  - **Absent_days table**: specifies how many days a student is absent in each course
  - **Pending_warning table**: specifies the pending warning status for each student in each course
  - **Warnings table**: specifies the warning status for each student in each course
  - **historical_student_records** table: contains a storage for identifying students by name, national id and email.
  - **studentssn_history table**:
  - **pending_ill_reports table**: contains the reports of illness absence including data and text of report for each student in each class
- **Back End system**: based on Node.js and Express, includes different APIs for using the database:

- Getting students data
- Getting courses data
- Getting reports
- Updating days status
- Processing days
- Confirming warnings
- Creating historical student records
- Login
- Downloading excel sheets for every thing

```
// Route to get attendance data for a course and student
http://localhost:3000/attendance/1/100001
http://localhost:3000/attendance/2/100002
for the following api and function make sure to check first if the student is enrolled in the course


// Route to get student ID by email
http://localhost:3000/student/student1@example.com

// Route to get courses enrolled by a student
http://localhost:3000/student/100001/courses

// Route to get students enrolled in a course
http://localhost:3000/course/1/students

// Route to get attendance summary for a student
http://localhost:3000/student/100001/attendance-summary
http://localhost:3000/student/100002/attendance-summary

// Route to get attendance summary for a course
http://localhost:3000/course/1/attendance-summary

// Route to get overall attendance report
http://localhost:3000/attendance-report
```

- **User interface** (Website / Mobile application / Desktop): based on flutter, a cross platform that produces different interfaces for desktop, mobile, and web. Each one has the following composition:
- **LEDs**: LEDs indicate specific conditions that happen to the systems as following:

Pin	Indication
D26	successful files upload
D27	failed files upload
D14	successful Wi-Fi connection (blinking while connecting)
D2	No RFID card
D15	successful card reading
D4	No SD card or RTC not connected

System algorithm:

- Setting the button and LEDs to output voltage when it is needed.
- Starting SPI communication (for SD and RFID modules).
- Checking for RTC and SD functionality. If they don't work the corresponding LEDs will output high.
- If push button is pressed, a function that is responsible for uploading files will be called.

- Checking if there is a new card to read. If not, the program will not do anything until there is a new one.
- Getting today's date to create a folder on the server titled by it if it is not found.
- If the operation of appending to file and sending it works successfully, all LEDs of success will turn on. And data will be stored in the database which can be accessed through the user interface.

4- Related work and challenges:

Some challenges faced us during the implementation of hardware circuit and software program including:

- **The slow transfer of files when sent byte by byte from SD card**: This was solved by creating a buffer that holds a large number of bytes to be sent together. The file is then divided to huge blocks of bytes instead of individual bytes. The remainder from sending the fixed number of blocks is sent at the last to complete the file without corruption.
- **The need for a server to test the system:** we overcame that by creating a server for testing on freevar. When the SD card sends a file then we transmitted it through the website and then receive it on another device. After that, we installed filezella to ensure that files are received correctly or not.
- **Errors due to connection failure, not receiving the file and so on:** this was solved by specifying the specific errors in the code so that when any of them occurs, there is a LED that can identify where the problem came from.
- **Errors related to getting and sending data to database:** solved by adding testing APIs to the back-end system and creating a simple web page for seeing the results.

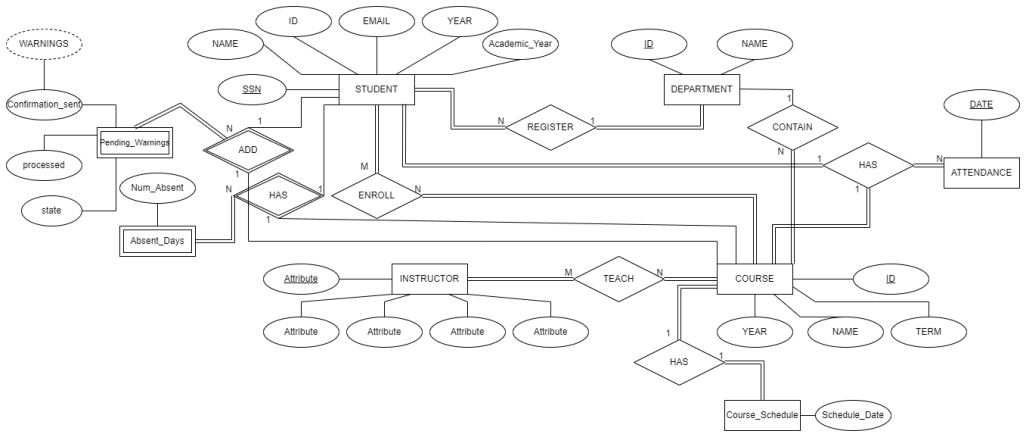
5- Implementation of software and hardware:

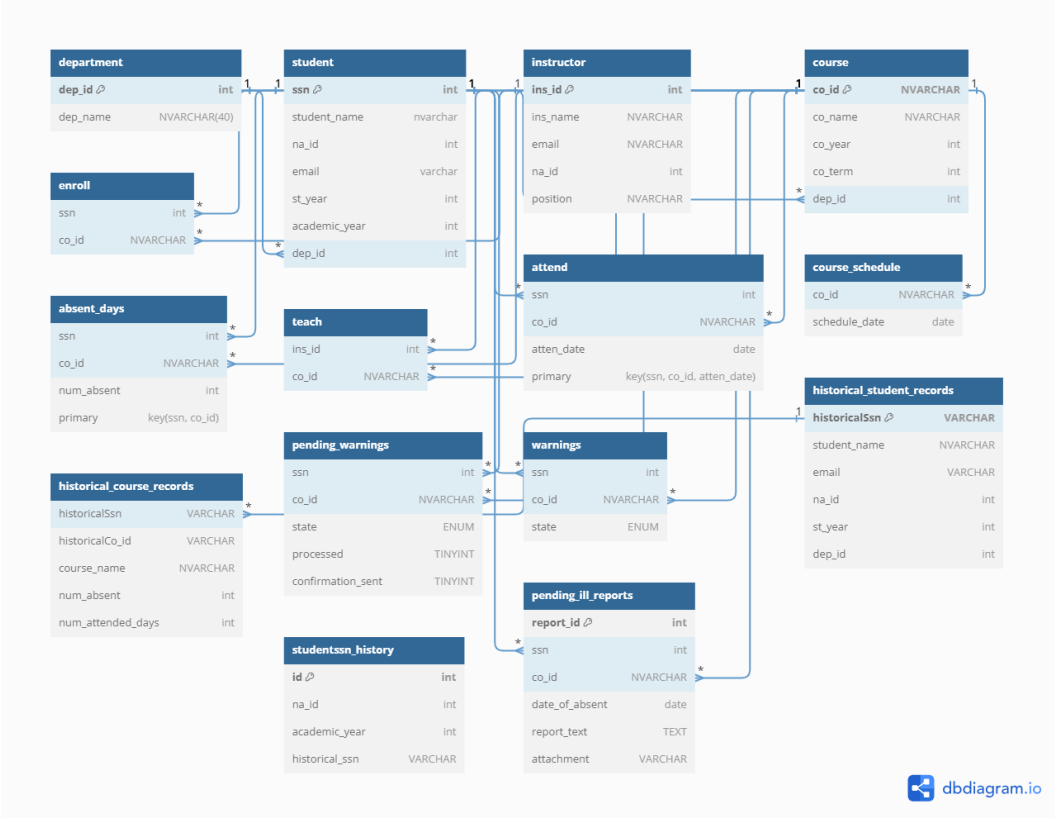
Hardware components:

Component	Picture
Micro SD Card Module for Arduino or MCU	
DS3231 High Precision (Real Time Clock) RTC Module	
ESP32 Development Board (WIFI and Bluetooth)	
RFID Reader-Writer Kit 13.56 Mhz	

Database and back-end software:

The database is composed of several tables related by the following diagram:



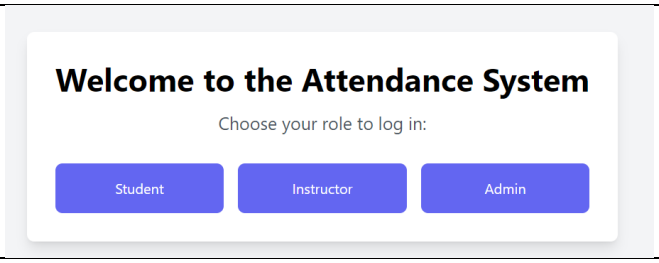
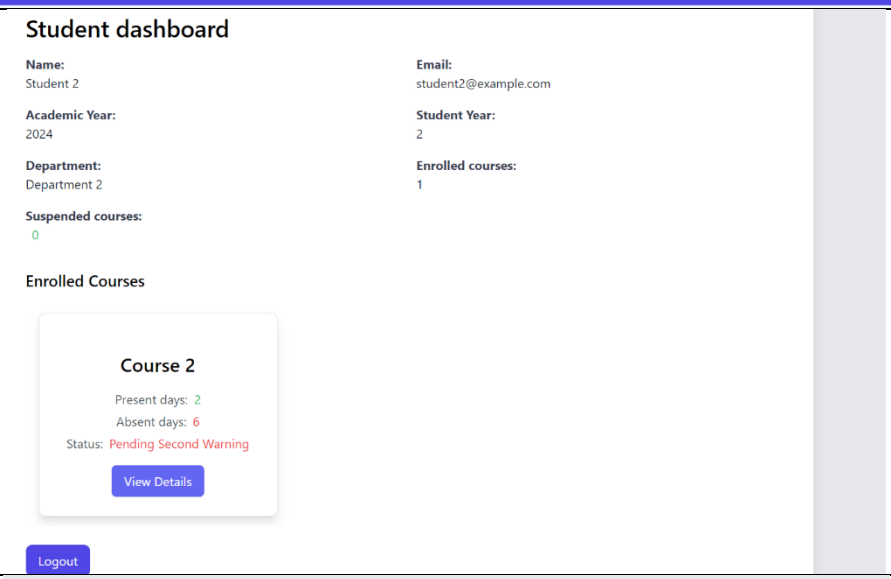
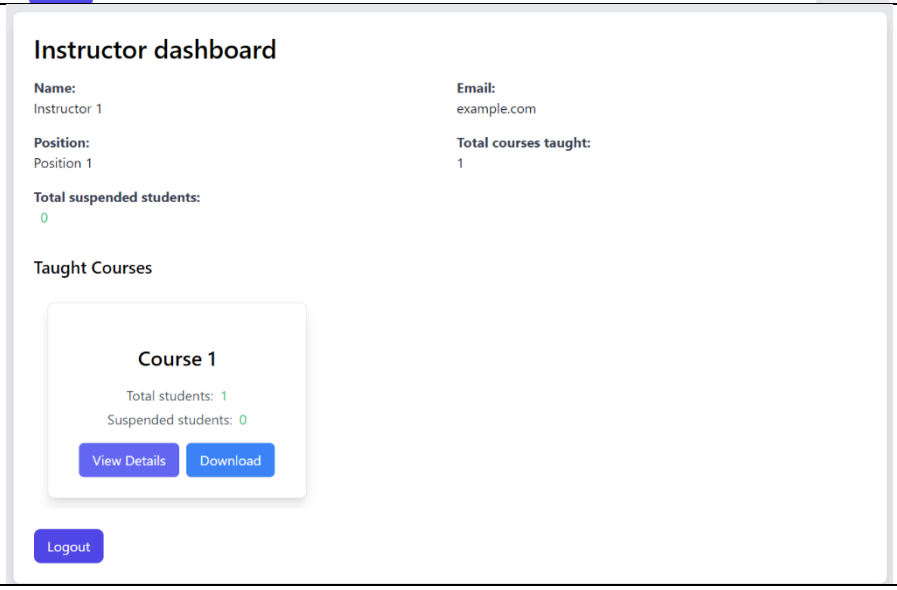


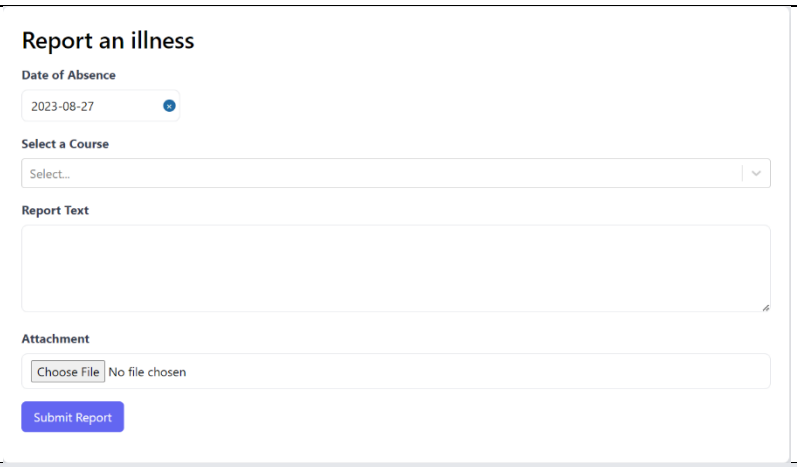
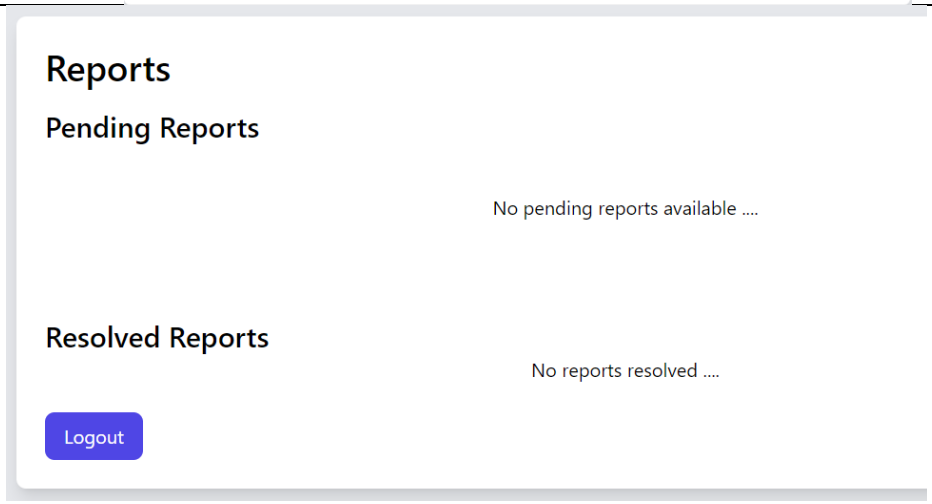
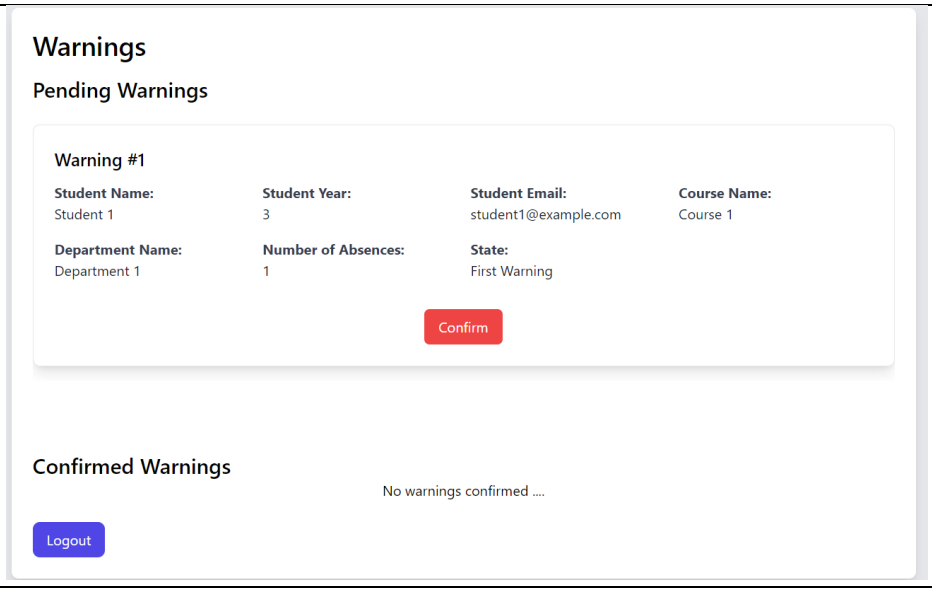
The APIs designed to integrate with these tables are:

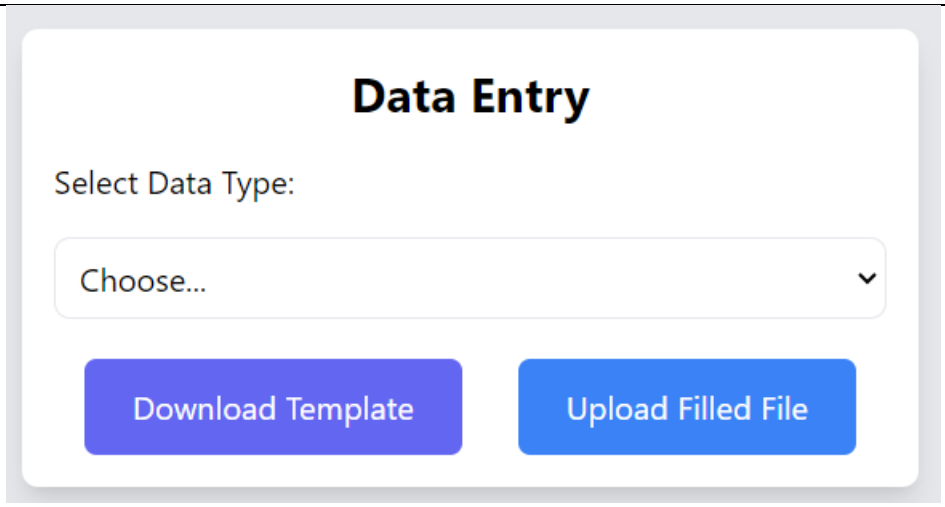
- Get Student ID by Email
- Get Student's Enrolled Courses
- Get Student's Attendance for a Course
- Get Course's Enrolled Students
- Get Student's Attendance Summary
- Get Course's Attendance Summary
- Get Overall Attendance Report
- Update Absent Days
- Update Pending Warnings
- Process All Pending Warnings
- Get Pending Warnings for Instructor to Confirm
- Confirm Warning
- Create Historical Student Records
- Get Historical Courses with Attendance for a student
- Get Historical Course Students with Attendance for a course
- Update Student SSN History
- Login a Student
- Login an Instructor
- Report Illness
- Get Illness Reports for Course
- Confirm Illness Report
- Download Attendance Data for All Students
- Get Attendance Data for All Students Enrolled in a Course
- Download Excel Template for Student
- Download Excel Template for Course
- Download Excel Template for Department
- Download Excel Template for Instructor
- Upload and Process Filled Excel File for Student
- Upload and Process Filled Excel File for Department
- Upload and Process Filled Excel File for Instructor
- Upload and Process Filled Excel File for Course

Frontend:

It is composed of the following features:

Feature	Picture
Login Page	
Task bar	
Student dashboard	
Instructor dashboard	

Illness report	
Report approval	
Warning approval	

Data entery	

6- Testing:

Phase 1:

Sending files from SD card to a server created on freevar. The file was received on filezilla program to be shown on the laptop.

Phase 2:

Sending files from SD card to local ftp server.

Phase 3:

Sending files from SD card to the university server. The file was received on filezilla program and also shown onto the server.

Phase 4:

Combining the ftp server sending and receiving project with RFID to read the data and RTC to store the time then tested for the last time with the university server.

Phase 5:

Testing the database contents on the mobile application and dashboard

7- Results:

The project achieved the following:

- measuring the time between students' recordings, as it is stored accurately in the excel sheets and database with prevention of student duplication.
- Creating a folder for everyday attendance. Each file in it represents one lecture/section named with the hour and minute.
- Checking for modules working and sending operation
- Creating a backup for files with the help of SD card. It can be freed anytime.
- Sending the last created file when the push button is pressed.
- Making the network works with its maximum benefit.
- Maintenance is achievable by a little effort due to the clear connections.
- APIs are all tested successfully and can retrieve any information needed.

8- Discussion:

The present study focused on the development and implementation of an IoT-based attendance system comprising an embedded system, a mobile application, and a backend database. The system aimed to address the limitations of traditional attendance systems by leveraging advanced technologies such as ESP32, RFID, FTP, and mobile applications. In this section, we will discuss the key findings, implications, limitations, and future directions of the research.

The results of the study demonstrated the effectiveness and efficiency of the IoT attendance system in accurately recording and managing student attendance. The embedded system, powered by the ESP32 microcontroller, successfully integrated RFID technology for seamless attendance data collection. The use of RFID tags and scanners enabled automated and contactless attendance tracking, reducing the possibility of errors and proxy attendance. The integration of FTP ensured secure and reliable transmission of attendance data between the embedded system and the backend database.

The mobile application provided real-time access to attendance information for students and administrators. It allowed users to view their attendance records, generate reports, and receive notifications. The mobile application's intuitive

interface and seamless integration with the embedded system facilitated convenient and timely access to attendance data, enhancing transparency and accountability.

The backend system, operating on a robust database, served as a central repository for storing and managing the attendance records of all students. It provided administrators with the ability to efficiently retrieve attendance data, generate comprehensive reports, and perform data analysis. The backend system's scalability and adaptability ensured its suitability for accommodating large student populations and future growth.

The IoT attendance system offered several advantages over traditional methods. It significantly reduced the time required for attendance management, as automated data collection eliminated the need for manual recording and compilation. The system also minimized the occurrence of errors, such as illegible handwriting or incorrect data entry. The real-time accessibility of attendance information empowered students and administrators to make prompt decisions based on accurate and up-to-date data.

Despite the promising outcomes, there are some limitations to consider. Firstly, the study focused on a specific implementation of the IoT attendance system, and the results may not be directly generalizable to other settings. Further research is needed to evaluate the system's performance and effectiveness in different educational or organizational contexts.

Secondly, while the use of RFID technology offers convenience and accuracy, it requires initial setup and distribution of RFID tags to students. Ensuring compliance and cooperation from all users may pose challenges and require appropriate measures to address resistance or non-compliance.

Lastly, the study did not explore the integration of additional features such as facial recognition or biometric authentication, which could further enhance the system's security and accuracy. Future research should investigate the feasibility and impact of incorporating such features into the IoT attendance system.

9- Conclusion:

IoT attendance system presented in this research demonstrated its potential to revolutionize attendance management in educational institutions and organizational

settings. The integration of the embedded system, mobile application, and backend database provided an efficient, accurate, and accessible solution for recording and managing student attendance. The system's use of advanced technologies and its potential for scalability make it a promising tool for streamlining attendance processes and improving operational efficiency. Further research and implementation efforts should focus on addressing the system's limitations and exploring additional features to enhance its functionality and usability.