CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Implementation Procedures

At the end of development, we have a very light weight device connected to a hosted web app through the internet with Wi-Fi. After the hardware was wired, soldered, coupled and programmed using Arduino IDE, it went through a rigorous testing phase for speed and durability. Next, was the software/web app. The UML Use case and Class diagrams helped to guide the next phases of development. The SQL database was built first with several tables including:

- Students
- Staff
- Machines
- Tasks
- School Fees
- Departmental Fees
- Faculty Fees
- Jobs
- Job Entries
- Faculties
- Departments
- Courses

The coding went next, with the appropriate user interface built with HTML, CSS, JavaScript, then the server-side code written with PHP. The web app has a page that accepts GET requests comprising of the Machine ID and the Unique card ID of a student. Once received, the web app confirms the task assigned to it by the lecturer then proceeds to check the database for what is required. This results a true or false value depending on how successful the task went. The code for both the hardware and

software are hosted online at: https://www.github.com/Emerald2240/rfid-smard-card-reader

4.2 Wiring / Assembly

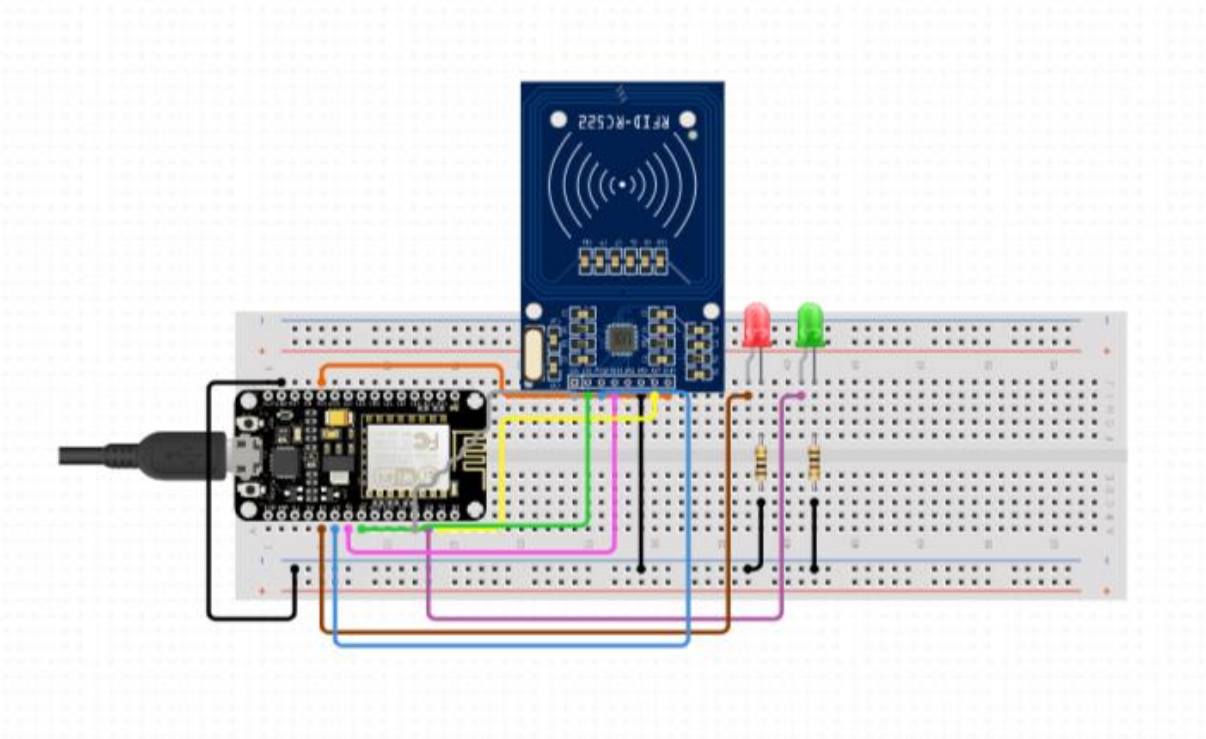

Fig 4.1: Breadboard Wiring of The System

In Fig X, the base/prototype wiring of the project is shown on a bread board. The Node MCU is powered by a type-a USB cord. The wiring configurations:

Table X: RFID Sensor and its Pin Connections to the Node MCU

MRC522 RFID Sensor Pins	Node MCU Pin
SDA	D4 (GPIO2)
SCK	D5 (GPIO14)
MOSI	D7 (GPIO13)
MISO	D6 (GPIO12)
IRQ	Not Connected

GND	GND
RST	D3 (GPIO0)
3.3V	3V3/3.3V

Table X: Other Components and their Pin Connections to the Node MCU

Other Components	Node MCU Pins
Red LED	D0 (GPIO16)
Green LED	D1 (GPIO5)

The tables above show the connections between the microcontroller, LEDs and RFID sensor. Some more LEDs and a Buzzer was added as the work progressed. When a registered RFID tag is shown to the device, it scans its unique ID and sends a request to the database. It takes less than a second for the server to respond with a true or false value. If the server returns a false, the red LED comes on for two seconds. If the reply was true, the green LED comes on for two seconds.

4.3 Coding

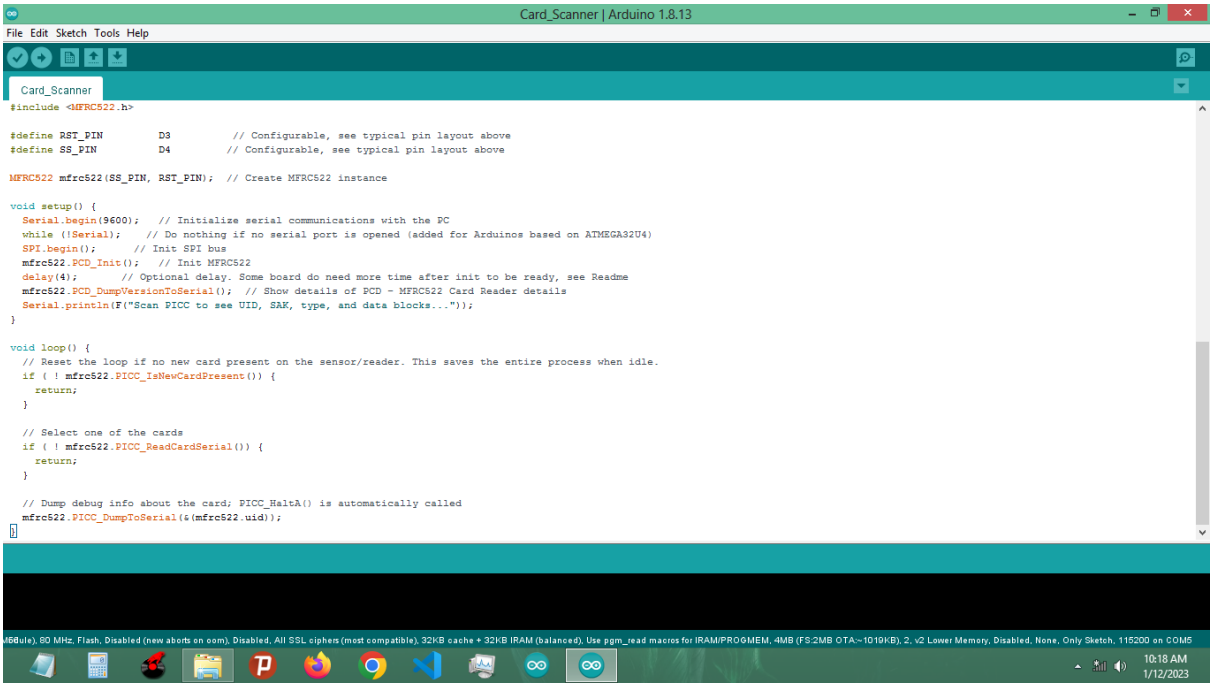

Fig 4.2: Arduino IDE

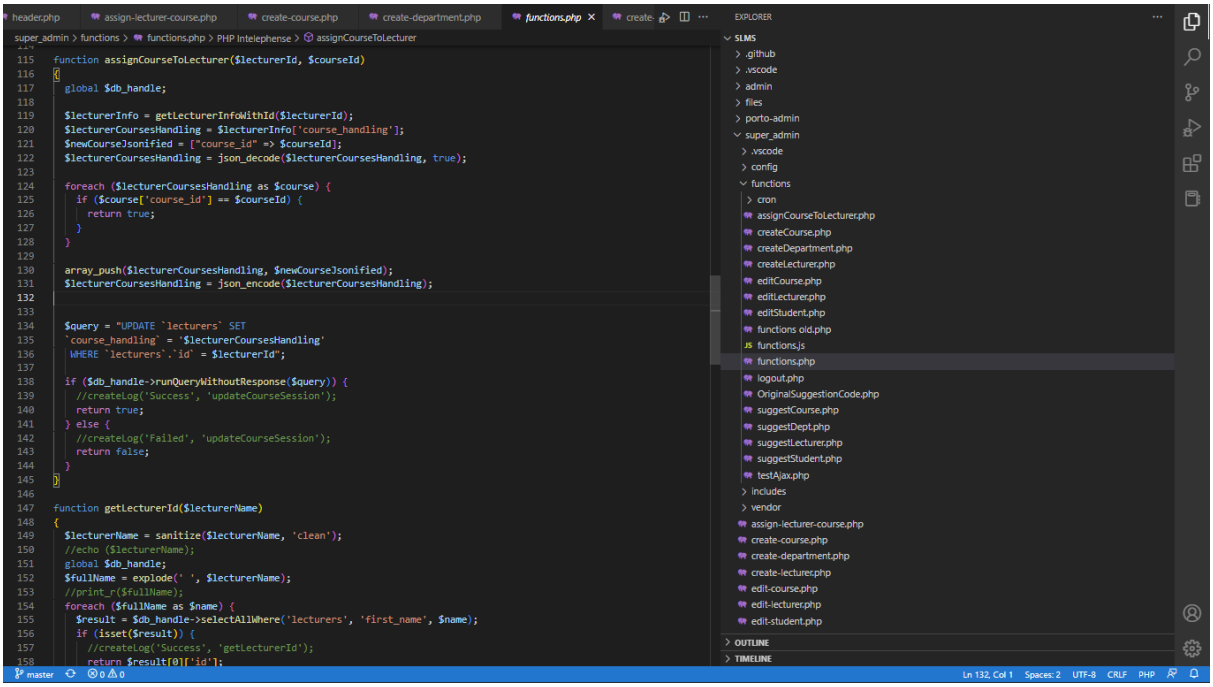

Fig 4.3: Microsoft VS Code IDE

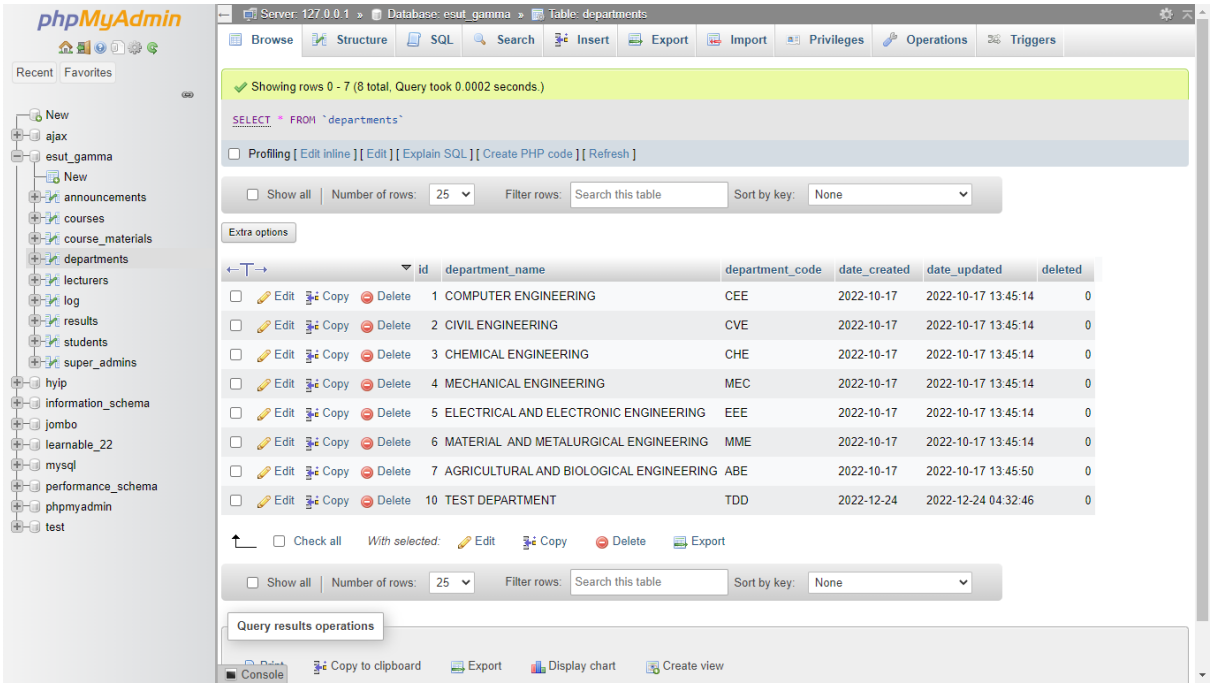

Fig 4.4: XAMPP PHPMyAdmin Screenshot

The above figures show the two IDE's used to build the hybrid. The first is the official Arduino IDE used to write and upload C code to the Node MCU. The second is Microsoft's Visual Studio Code, an open source IDE that supports multiple languages including HTML, CSS, JavaScript, MySQL and PHP. The third image is a screenshot of the localhost XAMPP PHPMyAdmin Database management system specifically installed so code can be run on my computer.

The code for both is hosted on GitHub at: https://www.github.com/Emerald2240/rfid-smard-card-reader

4.4 Implementation Results

After the assembling, wiring, soldering, coding, debugging and testing; here are the results.

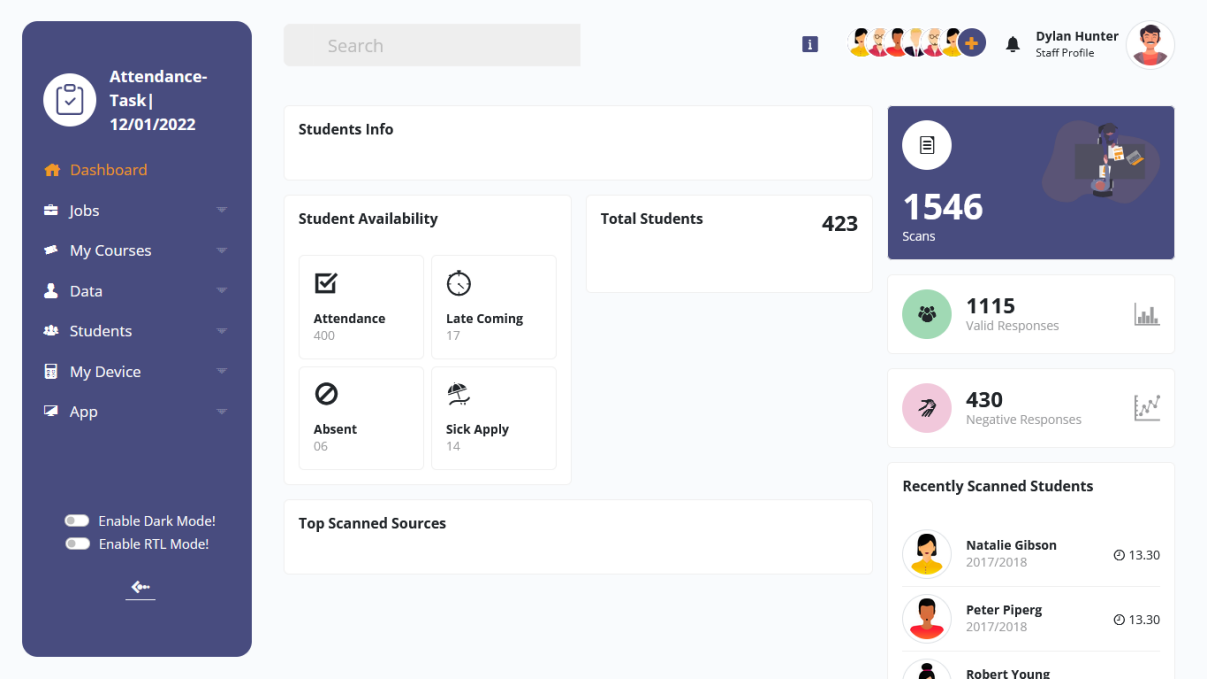

Fig 4.5: Screenshot of Web App Staff Side Dashboard

The figure above shows a simple screenshot of the web app UI. This particular screenshot contains very basic stats for the “Attendance Task” Job created by the currently logged in staff. Stats like Students Info, Attendance, Late coming, Absentee, Total Scans, Responses and so on. The staff is also able to download this data in two formats: CSV and Excel.

4.5 Summary of Results

The aim of this project was to build a smart card reading device that connects to the internet and stores its data online. As at the final day of development, I can say this application has reached its main purpose as stated in Chapter One.

After building and running the device for the first time, here are the results that were observed:

1. The machine is extremely Lean and can work for hours without any signs of slowing down or heat.
2. Because of its low data output, the machine is extremely fast. Returning card results within 200 milliseconds.
3. The machine is really portable and can easily fit into a pocket.
4. My beta testers found the Web app UI really clean and easy to use.
5. Data online can be easily sorted and retrieved through the web app.
6. Users loved the idea of downloading the data as excel or csv but preferred excel because of its popularity.
7. The device has shown to be secure. External readers cannot make use of the tag unique ID’s except in the extreme case of forceful collection/Stealing
8. The web app/web server showed no stress whatsoever after repeated use