Final Year Project

EMPLOYEE ATTENDANCE MONITOR

Samuel Feerick

G00319993

Bachelor of Software & Electronic Engineering

Galway-Mayo Institute of Technology

2019/2020

Supervisor: Mairtin O'Conghaile

EMPLOYEE ATTENDENCE MONITOR

SAMUEL FEERICK
FINAL YEAR PROJECT
BENG (HON) SOFTWARE & ELECTRONIC ENGINEERING

SUPERVISOR: MAIRTIN O'CONGHAILE

Summary

This project is designed to track the day to day attendance of an employee in a workplace. This project will give two forms of login in which are signing in via an RFID tag or your fingerprint.

It will then catch your data, display it on an LCD display and then store the data on a web page which the employee and the company's boss can view to monitor the attendance and time the employee arrived at work.

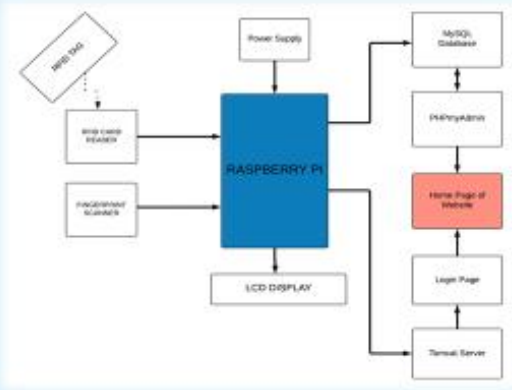
This product takes the stress and worry of manually submitting employee's attendance away from the job.

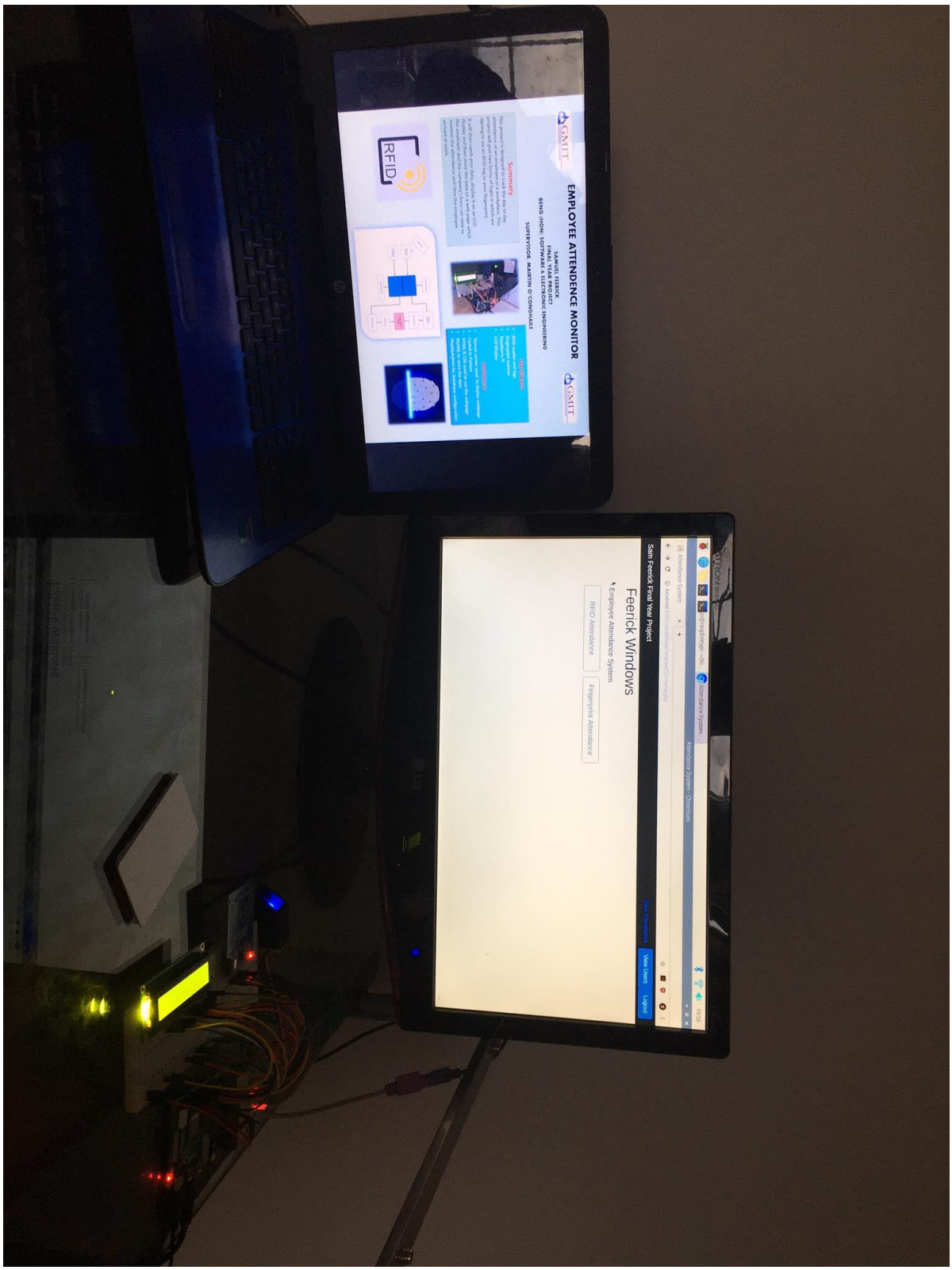
Hardware

- RFID reader and tags
- Fingerprint scanner
- Raspberry Pi
- LCD Display

Software

- Tomcat server used to deploy webpage
- Coded in Python
- HTML & CSS used to run the webpage
- MySQL to store the data
- PhpMyAdmin for Database configuration




EMPLOYEE ATTENDENCE MONITOR
SAMUEL FEERICK
FINAL YEAR PROJECT
BENG (HON) SOFTWARE & ELECTRONIC ENGINEERING
SUPERVISOR: MARTIN O CONNOHAIR
GMIT
RFID
Summary
Hardware
Software
Sam Feerick Final Year Project
Feerick Windows
Employee Attendance System
RFID Attendance
Fingerprint Attendance
View Attendance
View Users
Logout

Declaration

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Software & Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

Acknowledgements

During the process of completing my project I received a lot of tips and valuable information. I would like to make the following acknowledgements to thank them for their generosity during my project.

Especially, I would like to acknowledge and thank my project Supervisor Mairtin O'Conghaile whose guidance has helped me a lot along the way in the making of my project. Also, a special thanks to all the Lecturers and Des O'Reilly.

Bill of Materials

Component	Quantity	Price (Total)	Purchase From
Raspberry Pi 4	1	€50	Radionics
AC/DC Power Supply	1	€8	Radionics
16GB Memory card	1	€12	Radionics
RFID-RC522	1	€10	Ebay
RFID Cards	6	€10	Ebay
LCD Display	1	€5	Ebay
USB to TTL Converter	1	€10	Ebay
Fingerprint Module	1	€25	Ebay

Table of Contents

1 Summary

From working for my father's construction business from an early age, I have foreseen that he has employed various amounts of employees down through the years. But when it comes to paying employees my father finds it difficult to accurately track the attendance and time of arrival of employees and in some cases he has payed employees for days which they have not turned up for work or have arrived late or went home early.

For my FYP I wanted to come up with a system that would eliminate this issue for my father. So, I came up with the idea of designing an employee attendance monitoring system that will work perfectly in my father's construction business and various other businesses.

My system is manufactured using Radio-Frequency Identification (RFID) technology and Fingerprint sensor technology which presents data to the raspberry pi and finally outputs it to a webpage for the employer to view.

This project was done with use of Python and PHP scripts, Apache Tomcat, phpMyAdmin, MySQL and HTML.

2 Introduction

Problem and Motivation for doing this project:

My father has owned and operated his own construction business for nearly 40 years now. The is business is the manufacturing and installation of windows and doors. In that time my father has employed quite a lot of employees. As it is a small enough business he has always looked after the payroll side of the business himself. The number of employees varies between 5-15 given the demand and how busy the company is.

As my father is now approaching the latter stage of his career, memory isn't always the best and between different jobs going on in different sites and him not always being at the workshop he struggles to acknowledge the punctuality and also the honesty of his employees is not always 100%.

Overview of the Project:

Given this problem I have taken this opportunity to design a system to eradicate this problem.

So, in my project which is an Employee Attendance Monitor, it will have a number of employees working for a business. They will all be given a RFID card. On their arrival to work every day, they will swipe their card on the RFID reader. The LCD display will then Light up with their name if they are a registered employee. This system has a second way of logging attendance if the employee were to lose their RFID card. The second way of logging attendance is through the fingerprint scanner. The system will ask you to enter your name then the user will place his finger to the sensor and then the data will be saved to the database. The company's boss or payroll team will then be able to view the employee's attendance data on a website, which will show the employee's name, ID number and the date and time of arrival to work.

3 Advantages and Disadvantages of the Old and New way

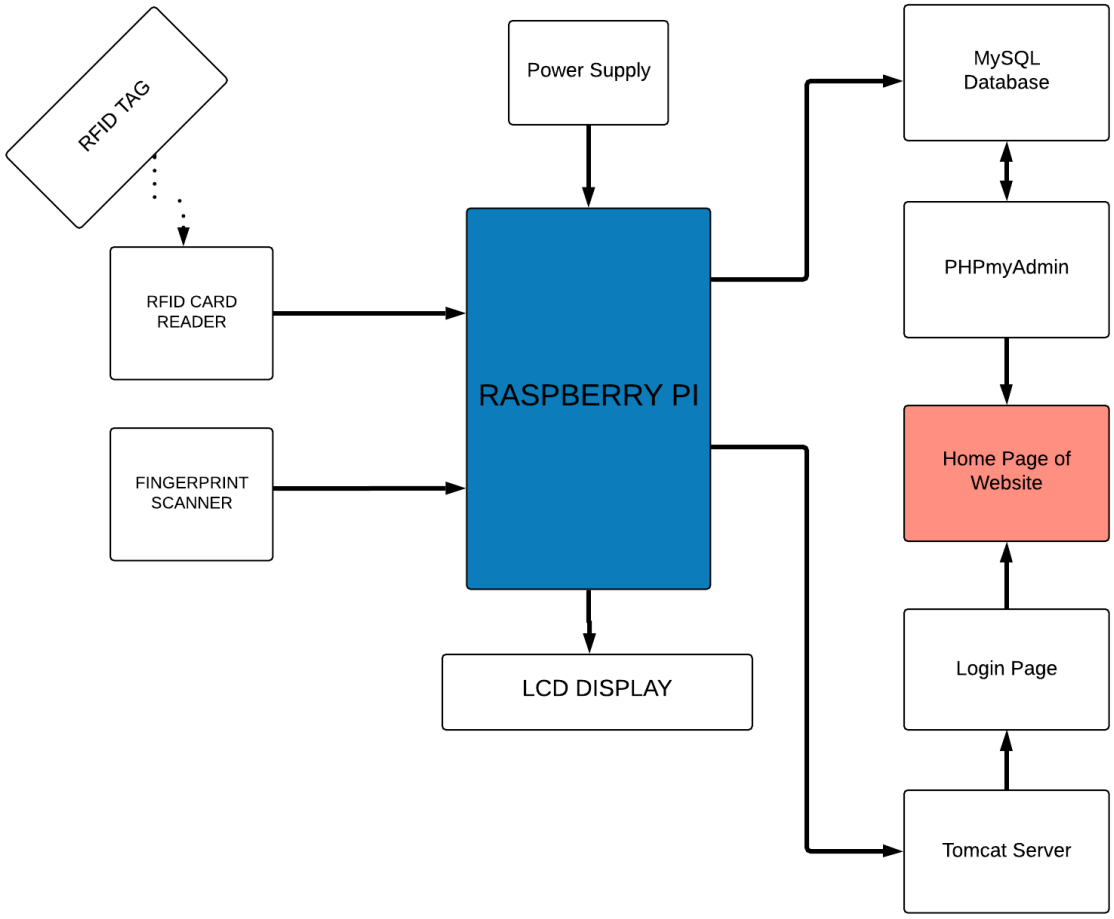
Disadvantages of the Old way:

- Time consuming filling out work hour sheets
- Lying about attendance and time of arrival
- Not accurate.
- Wasted paper

Advantages of the New way:

- Accurate
- Fast
- Employees cannot lie about Attendance
- Saves paper

4 Architecture Block Diagram



This block diagram shows the different technologies used in the project.

The Employee Attendance System is developed on the Raspberry Pi 4. The MFRC 522 RFID card reader module is used to take the data inputted from the RFID cards. The fingerprint scanner takes the data from the fingerprint as the second option of attendance. The name will then light up across the LCD display. The data is then all stored in the MySQL database. The data is sent to PhpMyAdmin then for online viewing and editing. The Apache tomcat server is then started this runs the website. First you will be asked to log in and on the completion of a successful login you will be brought to the website home page.

5 Development Platform

Raspberry Pi 4 Development Board

Raspberry Pi 4

The Raspberry Pi 4 board was used for the development of this project.

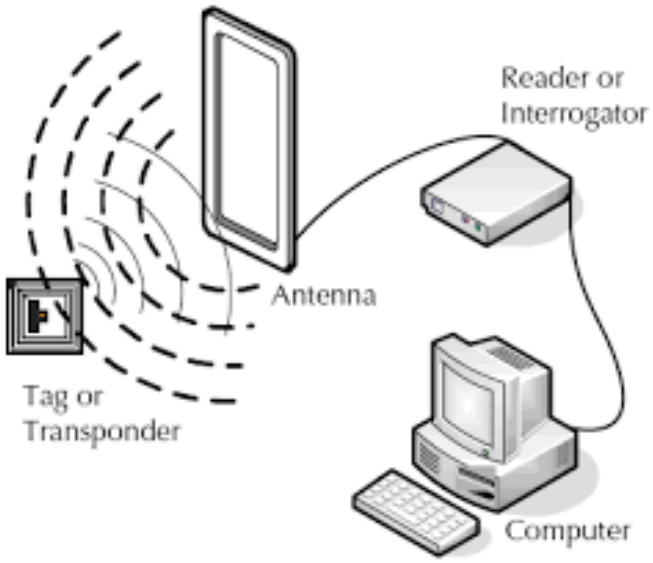
The board was released in June 2019 with a 64-bit quad core processor, it has on-board Wi-Fi, Bluetooth, 4 USB connections and 2 Micro HDMI ports. It is an ARM cortex based on popular development board designed for electronics engineers and developers. It is a single board computer working on low power. With the processing speed and memory, Raspberry Pi can be used for performing different functions at a time just like a normal PC.

The Raspberry Pi is used for controlling the whole process of this project. The Raspberry Pi receives data from the RFID module and the fingerprint module and transfers the data to the desired destinations.

6 Tools & Technologies

RFID Technology

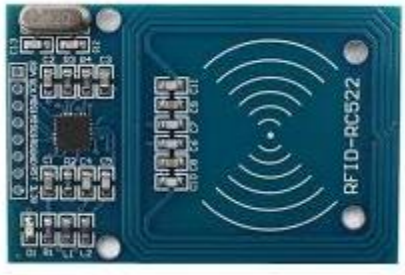
Radio Frequency Identification is a technology that uses radio waves to transfer data from an RFID card, through a reader to uniquely identifying a person. An RFID system consists of a card and a reader. The RFID reader consists of an antenna and a transceiver. The reader is usually on the continuously senses its range of operation. Whenever a RFID card enters its field of operation, the RFID reader transmits the electromagnetic waves using an antenna to communicate with the card's antenna. The cards antenna receives data from the reader, activates card and sends back electromagnetic waves with RFID card information. The transceiver in the reader receives data and passes them to the controllers. RFID systems can deliver accurate data about tagged cards and improve efficiency.



RFID Process Example

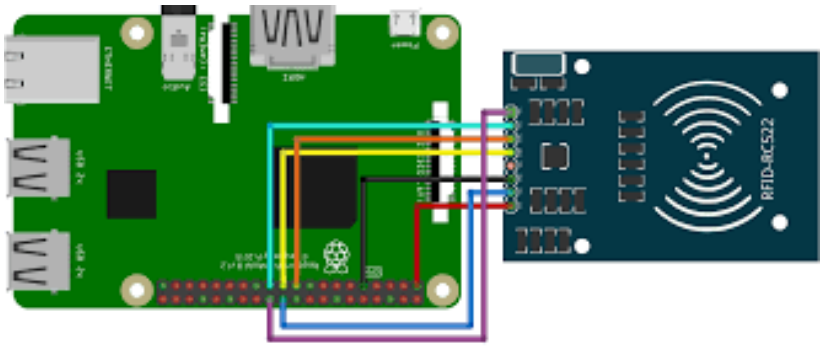
RC-522 RFID Technology

The RFID-RC522 reader module communicates with cards up to 1cm away using a 13.56MHz electromagnetic field, then sends the data to a Raspberry Pi 4 board through SPI communication. The module is connected to 3.3v on the Raspberry Pi 4 board, if connected to 5v the module is at risk of being damaged.



MFRC522 RFID reader module

In this project 2 RFID library functions are used for reading the RFID cards. The first function allows to detect whether an RFID card is present. The second function reads the RFID id. The RFID card is in an array of bytes and is assigned to a person in a hexadecimal value. Each card has a separate hexadecimal value.



Reader to Raspberry Pi Schematic Diagram I used

JM-101 Fingerprint Scanner

The JM-101 fingerprint module is an integrated fingerprint processing module that integrates the optical path and fingerprint processing part. It has the characteristics of small size, low power consumption and simple interface. It has high reliability, fast recognition speed, good wet and dry finger adaptability, and fingerprint search. high speed. JM-101 module communication interface is USB and UART communication interface.

Fingerprint sensor used the project.

The fingerprint sensor is connected to the Raspberry Pi via a USB to TTL converter.

USB to TTL converter used on the project.

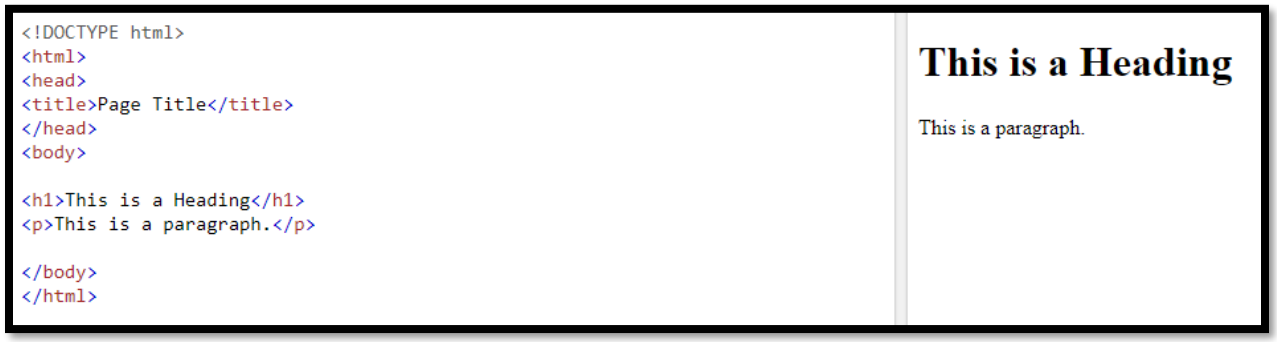
7 Software

HTML

HTML stands for Hypertext Markup Language. It was developed in 1991 as HTML 1.0 as it debuted as a hybrid of Standard Generalized Markup Language (SGML). HTML 1.0 just included 20 HTML elements.

HTML is a computer language developed to allow website creation. These websites can then be seen by anyone on the internet. HTML is the most used language on the web and it is relatively easy to use and learn.

In HTML the text in wrapped HTML tags that start with <html> and ends with </html>. These tags tell your browser about the structure of the document.



HTML Example Code, before and after

In the above figure, the HTML <html> tag tells the browser that it is a HTML document. The text between the HTML tags describes the actual web page.

The <head> tag is a container for the head element. Elements inside <head> can include scripts.

The <body> tag contains all the contents of a HTML document which will be shown such as <h1> and <p> tags shown.

The text between the <h1> tags is displayed as heading meaning the font is bigger and in bold and the text between the <p> tag is displayed as a paragraph.

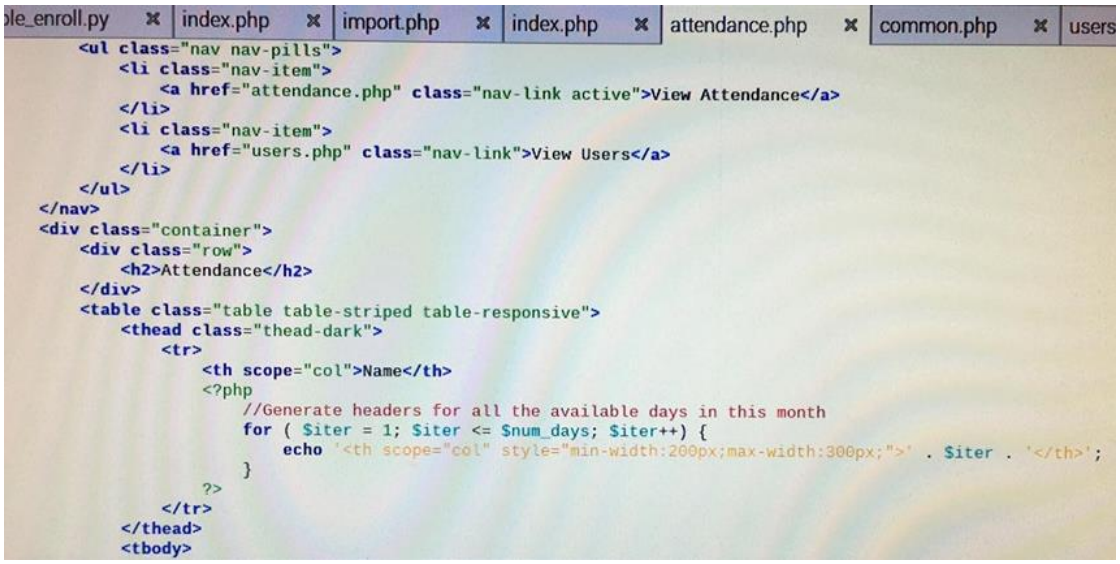
PHP

PHP is a server-side scripting language. that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages.

PHP scripts can only be interpreted on a server that has PHP installed. A PHP file contains PHP tags and ends with the extension ".php".

Advantages of PHP

The main advantage of php for using on this project is that it communicates with MySQL and PhpMyAdmin very well and very easy to use.



Example of .php code used in this project

CSS

CSS stands for Cascading Style Sheets. It describes how HTML is displayed on the screen. It saves a developer a lot of time and effort as it can control the layout of multiple web pages at once. CSS is used to define different styles for a web including the design, layout and variations in display for different devices and screen sizes. Selectors are used in CSS to select HTML elements based on their element name and id.

```
.button {
    background: linear-gradient(#eee, #ccc);
    border: 1px solid #999;
    color: #333;
    cursor: pointer;
    padding: 1em 1.5em;

    &:hover {
        background: linear-gradient(#fff, #ddd);
        color: #111;
    }
}
```

CSS Example Code

In the above figure it shows us an example of a declaration block in CSS code. A CSS declaration always ends with a semicolon declaration blocks are surrounded by curly braces.

Python

Python is a high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

Python is used in this project for controlling the hardware.

```
try:
  while True:
    lcd.clear()
    lcd.message('Place Card over\nSams RFID Reader')
    id, text = reader.read()
    cursor.execute("SELECT id FROM users WHERE rfid_uid="+str(id))
    cursor.fetchone()

    if cursor.rowcount >= 1:
      lcd.clear()
      lcd.message("Overwrite\nexisting user?")
      overwrite = input("Overwite (Y/N)? ")

      if overwrite[0] == 'Y' or overwrite[0] == 'y':
        lcd.clear()
        lcd.message("Overwriting user.")
        time.sleep(1)
        sql_insert = "UPDATE users SET name = %s WHERE rfid_uid=%s"
      else:
        continue;
```

Python Code used in my Project

In the figure above the python code is the script that is used to register an employee via the RFID reader.

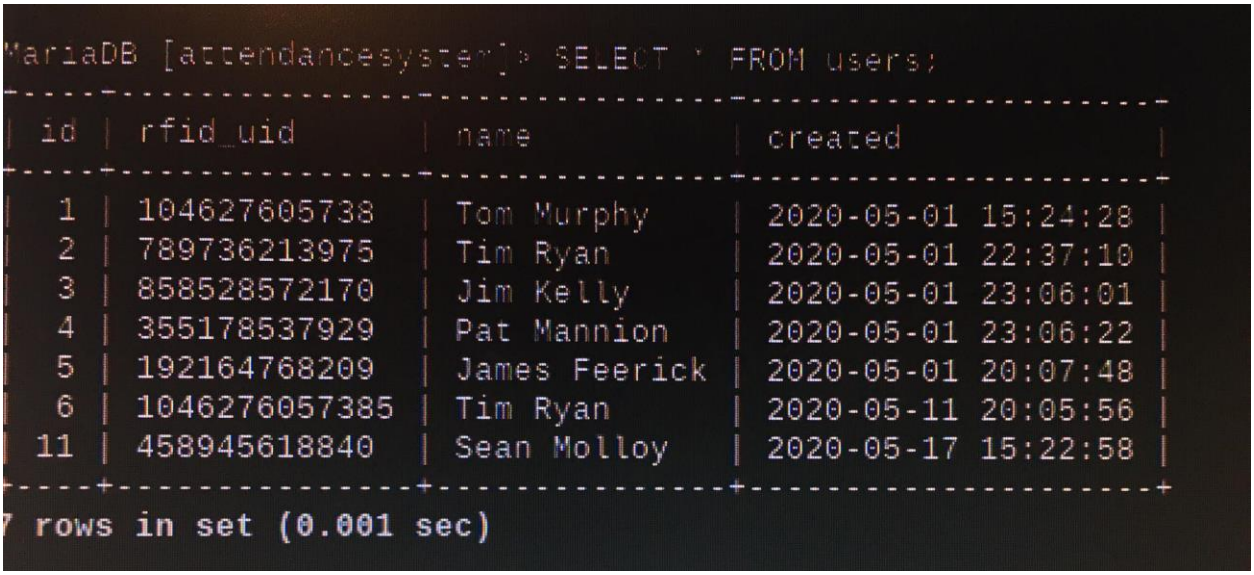
MySQL

MySQL is the database system that is used on the web. It is reliable, faster and cheaper than any other database system. Like most database systems, MySQL is a Relational Database Management System (RDBMS). This means data in the form of related tables.

MySQL is also a client/server system. The client can run on the same computer as the server or on a different computer then the server via the internet. Almost all the major database systems are client/server systems.

MySQL supports Structured Query Language (SQL) as its database language. SQL is a standardized language and queries and updates data in a database.

A database is queried for specific information and is returned.



```
MariaDB [attendancesyste]> SELECT * FROM users;
+----+---------------+---------------+---------------------+
| id | rfid_uid      | name          | created             |
+----+---------------+---------------+---------------------+
|  1 | 104627605738  | Tom Murphy    | 2020-05-01 15:24:28 |
|  2 | 789736213975  | Tim Ryan      | 2020-05-01 22:37:10 |
|  3 | 858528572170  | Jim Kelly     | 2020-05-01 23:06:01 |
|  4 | 355178537929  | Pat Mannion   | 2020-05-01 23:06:22 |
|  5 | 192164768209  | James Feerick | 2020-05-01 20:07:48 |
|  6 | 1046276057385 | Tim Ryan      | 2020-05-11 20:05:56 |
| 11 | 458945618840  | Sean Molloy   | 2020-05-17 15:22:58 |
+----+---------------+---------------+---------------------+
7 rows in set (0.001 sec)
```

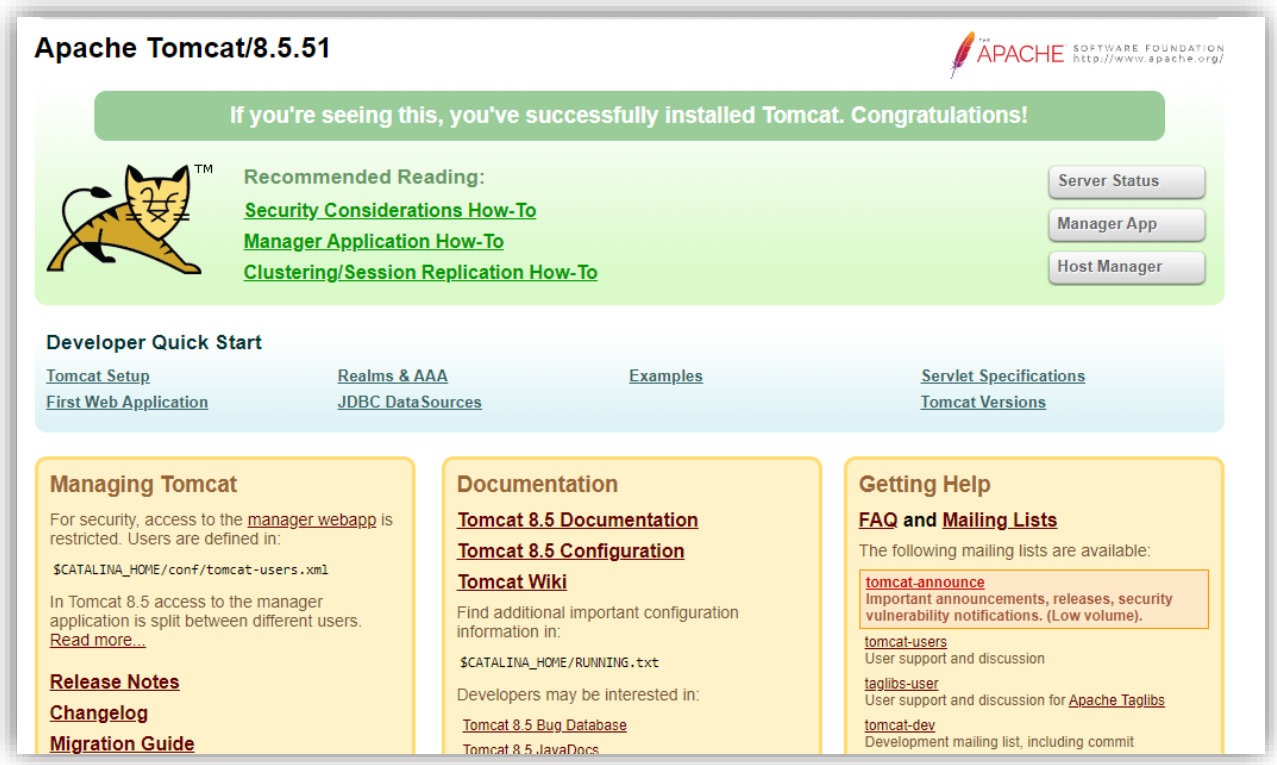
Example MySQL table that is used in the Project

Apache Tomcat 8

Apache Tomcat is an open-source Java Servlet container developed by the Apache Tomcat Foundation. The application server is designed to execute the PHP scripts and render Web pages that includes the coding.

It is used to deploy PHP scripts. A WAR (short for Web Archive) file is built in a jar file and is then dropped into the deploy directory in Tomcat.

After downloaded to the Raspberry Pi and a successful start-up. The default web applications will be displayed when visiting the link http://localhost:8080/. Upon execution the following page is displayed in the figure below.

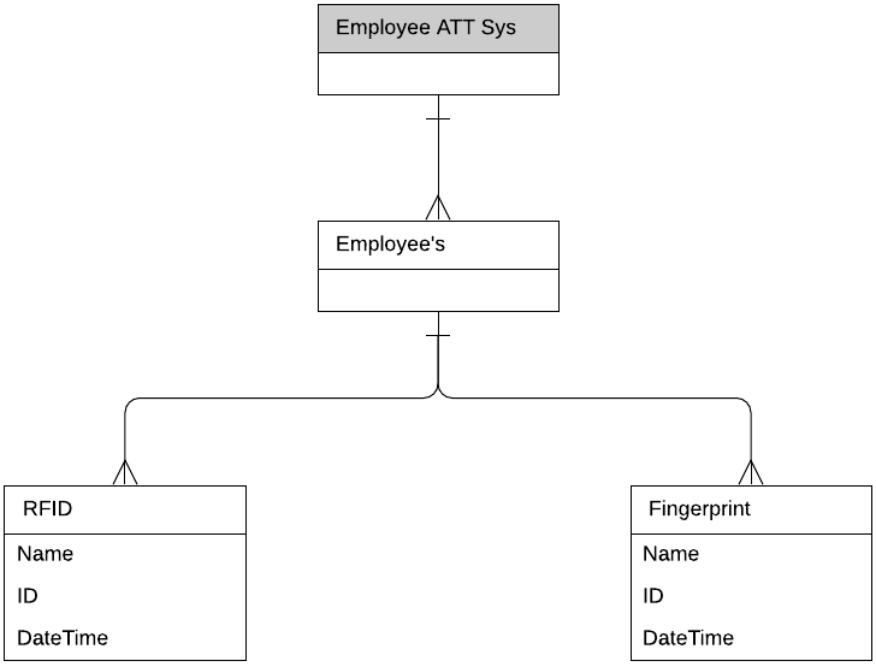


8 Implementation

For this project, a database is needed to host several tables and values. MySQL is used to create and control the database. A table is needed for each heading so that so that the values entered can be stored separately in each table.

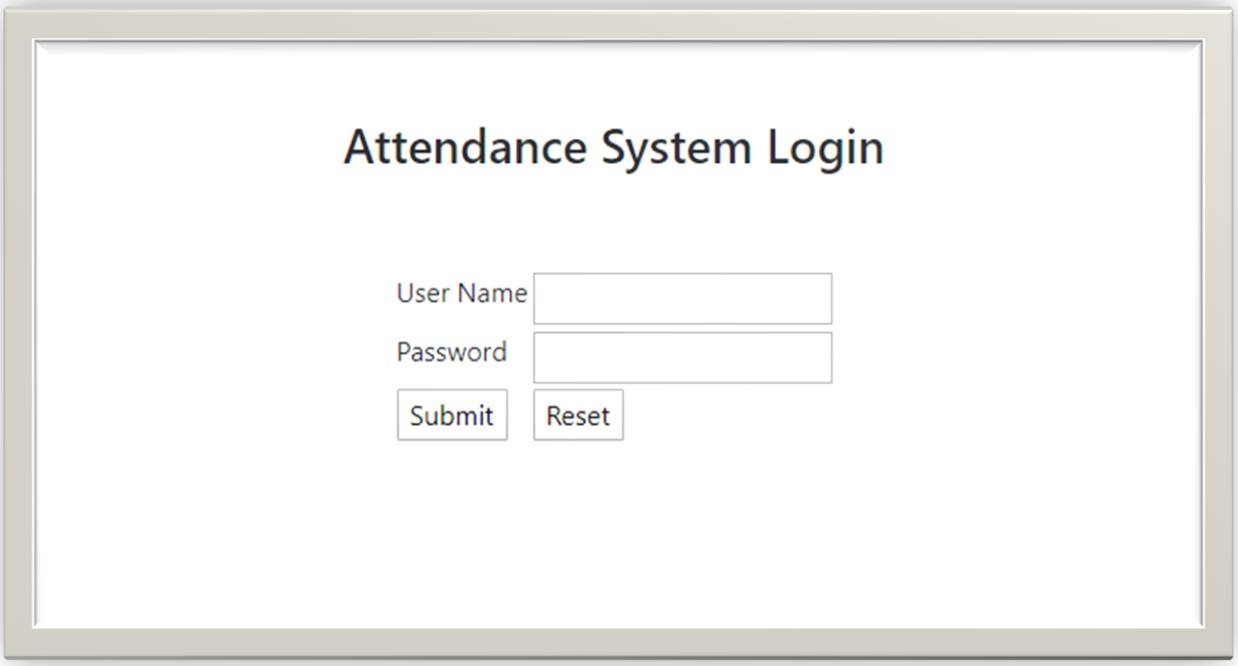
In the figure below it shows the database structure for this project.

There are three tables. One for saved employee's, then the two other tables are the entry tables. One for taking the RFID attendance. And the other table is for if the employee lost their card, they can sign in using the fingerprint sensor.



After the database was created the next stage of the project was the development and design of the website. A simple design was created for the overall website to make it simple and easy to use, done using CSS and HTML. In the below figure it shows the login page which was the first page created for this website. For this page I created the username and password in the .php script as I did not need multiple users for the website. On this page the user has the option to enter the Username and Password, submit it if its correct or reset it if its wrong. If the user then enters a correct Username and password, they will then have access to the home screen of the website.

Below is the first page you see when you execute the website through the Tomcat server.



Below is a snippet of the .php file for creating the login page.

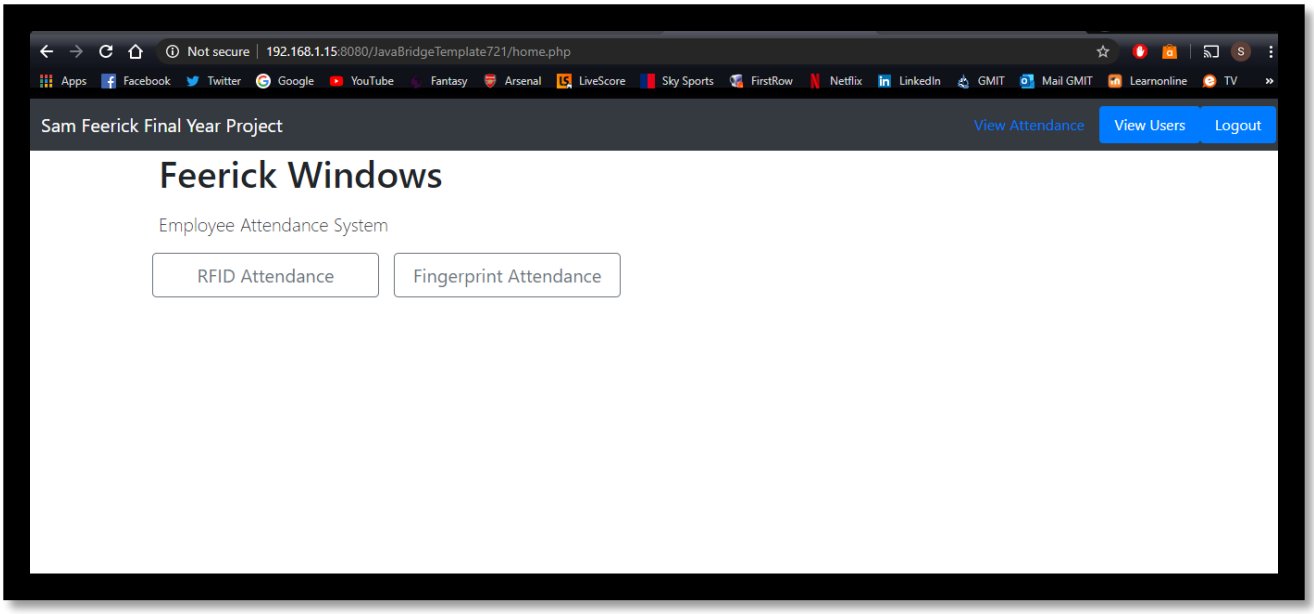
```
<h3>Attendance System Login</h3>
        <form id="login-form" method="post" action="login_authenticate.php">
                <table border="0.5">
                        <br>
                        <tr>
                                <td><label for="user_id">User Name</label></td>
                                <td><input type="text" name="user_id" id="user_id"></td>
                        </tr>
                        <br>
                        <tr>
                                <td><label for="user_pass">Password</label></td>
                                <td><input type="password" name="user_pass" id="user_pass"></td>
                        </tr>
                        <tr>
                                <td><input type="submit" value="Submit"></td>
                                <td><input type="reset" value="Reset"></td>
                        </tr>
                </table>
        </form>
```

Below is the Authentication page for this login.

```php
<?php
session_start();

if (isset($_POST['user_id']) and isset($_POST['user_pass'])){
        // Assigning POST values to variables.
        $username = $_POST['user_id'];
        $password = $_POST['user_pass'];

        if($_POST['user_id'] == "samfeerick" && $_POST['user_pass'] =="123456789"){
                echo "pass";
                echo "<script type='text/javascript'>alert('Login Credentials verified')</script>";
                $_SESSION['user'] = "samfeerick";
                header("Location: http://localhost:8080/JavaBridgeTemplate721/home.php");

        }else{
                echo "<script type='text/javascript'>alert('Invalid Login Credentials')</script>";
                header("Location: http://localhost:8080/JavaBridgeTemplate721/login.php");

        }
}
```

Rather than setting up a register page for signing up for using the website, I created a .php script that will set up the database for the user. If I want to set up more users for the website all I have to do is create a new .php file with a new username and password for the user.

In the below picture it shows the homepage of the website.




```
38                  Employee Attendance System
39              </p>
40              <div class="row mx-n2">
41                  <!-- if users needed uncommment this block
42                  <div class="col-md px-2">
43                      <a href="users.php" class="btn btn-lg btn-outline-secondary w-100 mb-3">Users</a>
44                  </div>
45                  -->
46                  <div class="col-md px-2">
47                      <a href="attendance.php" class="btn btn-lg btn-outline-secondary w-100 mb-3" >RFID Attendance</a>
48                  </div>
49
50                  <div class="col-md px-2">
51                      <a href="fingerprint.php" class="btn btn-lg btn-outline-secondary w-100 mb-3" >Fingerprint Attendance</a>
52                  </div>
53
54              </div>
```

Above is a snippet of code used to program the home page of the webpage.

On the Home page it shows the various different options you can choose. If you click on the "RFID Attendance" button, it will bring you to the attendance page that the employees signed in with their RFID cards.

```
78                  foreach($users as $user) {
79                      echo '<tr>';
80                      echo '<td scope="row">' . $user['id'] . '</td>';
81                      echo '<td scope="row">' . $user['name'] . '</td>';
82                      echo '<td scope="row">' . $user['rfid_uid'] . '</td>';
83
84                      //Iterate through all available days for this month
85                      for ( $iter = 1; $iter <= $num_days; $iter++) {
86
87                          //For each pass grab any attendance that this particular user might of had for that day
88                          $attendance = $database->select("attendance", [
89                              'clock_in'
90                          ], [
91                              'user_id' => $user['id'],
92                              'clock_in[<>]' => [
93                                  date('Y-m-d', mktime(0, 0, 0, $current_month, $iter, $current_year)),
94                                  date('Y-m-d', mktime(24, 60, 60, $current_month, $iter, $current_year))
95                              ]
96                          ]);
97
```

Above is a snippet of how I built the "RFID Attendance" page. It shows how I used the TimeStamp within the code to catch the time and date of Attendance.

The button “Fingerprint Attendance” is used for viewing the attendance the employee has submitted using his fingerprint. This is the second alternative method of submitting attendance and is used if the employee has mislaid or forgot to bring his RFID card to work that day.

```
62          </div>
63          <table class="table table-striped table-responsive">
64              <thead class="thead-dark">
65                  <tr>
66                      <th scope="col">Id</th>
67                      <th scope="col">Name</th>
68                      <th scope="col">Timestamp</th>
69
70                  </tr>
71              </thead>
72              <tbody>
73                  <?php
74                      //Loop through all our available users
75                      foreach($users as $user) {
76                          echo '<tr>';
77                          echo '<td scope="row">' . $user['id'] . '</td>';
78                          echo '<td scope="row">' . $user['Name'] . '</td>';
79                          echo '<td scope="row">' . $user['TimeStamp'] . '</td>';
80                          echo '</tr>';
81                      }
```

Above is a snippet of the code used to design the .php script for displaying the data for the fingerprint attendance.

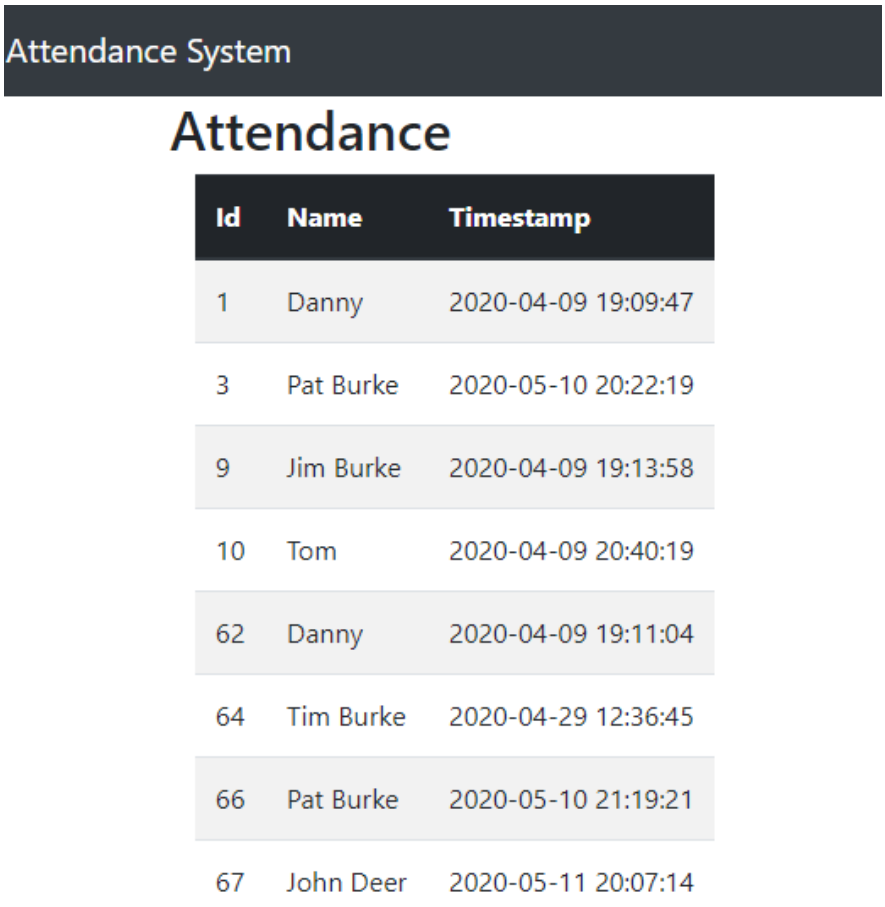
```
 9  //Grab all users from our database
10  $users = $database->select("record_table", [
11      'id',
12      'Name',
13      'TimeStamp'
14  ]);
```

The above lines of code are used to transfer the data from the MySQL database to the website.

```php
//Check if we have a year passed in through a get variable, otherwise use the current year
if (isset($_GET['year'])) {
    $current_year = int($_GET['year']);
} else {
    $current_year = date('Y');
}

//Check if we have a month passed in through a get variable, otherwise use the current year
if (isset($_GET['month'])) {
    $current_month = $_GET['month'];
} else {
    $current_month = date('n');
}

//Calculate the amount of days in the selected month
$num_days = cal_days_in_month(CAL_GREGORIAN, $current_month, $current_year);
```

The above code is used to capture the correct date and time for submitting the attendance.

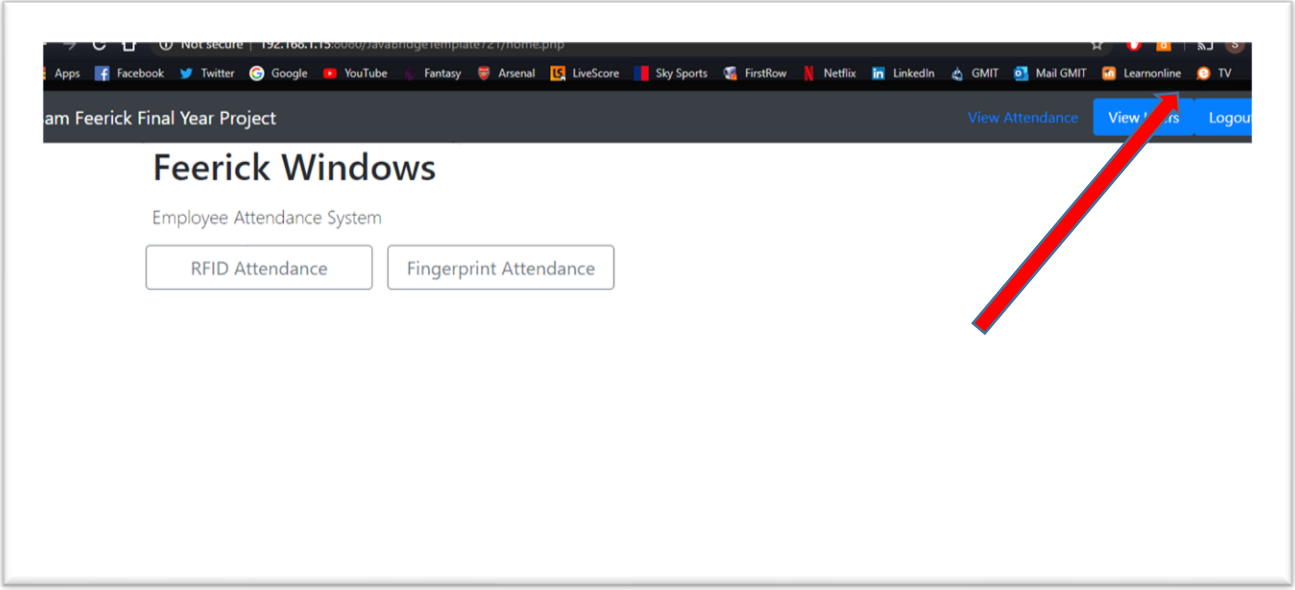
Below is an example of the page for the fingerprint attendance and shows the ID number, the Name and the Time and Date of the Employee that signed in on the specific day.



Attendance System

Attendance

Id	Name	Timestamp
1	Danny	2020-04-09 19:09:47
3	Pat Burke	2020-05-10 20:22:19
9	Jim Burke	2020-04-09 19:13:58
10	Tom	2020-04-09 20:40:19
62	Danny	2020-04-09 19:11:04
64	Tim Burke	2020-04-29 12:36:45
66	Pat Burke	2020-05-10 21:19:21
67	John Deer	2020-05-11 20:07:14



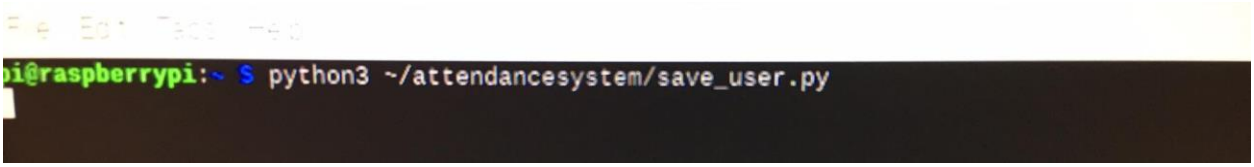
The last feature on the webpage is the Logout button which will bring the user back to the login page.

```php
<?php
// Always start this first
session_start();

// Destroying the session clears the $_SESSION variable, thus "logging" the user
// out. This also happens automatically when the browser is closed
session_destroy();
header("Location: http://localhost:8080/JavaBridgeTemplate721/login.php");

?>
```

The above code configures the logout.

Now the website must integrate with the hardware of this project.

So, for recording RFID Attendance the employee must be registered to an RFID card. The way this is done is running the saveuser.py script. So, I will run the script as shown below:

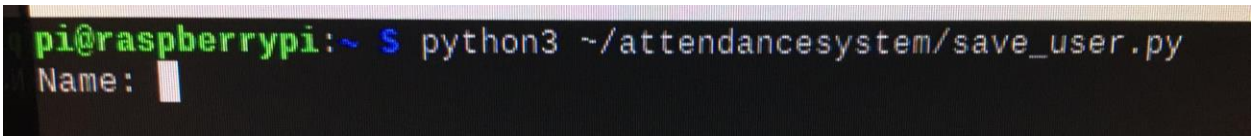


That will then ask the user to place a RFID card for the reader displaying the following message on the LCD display:

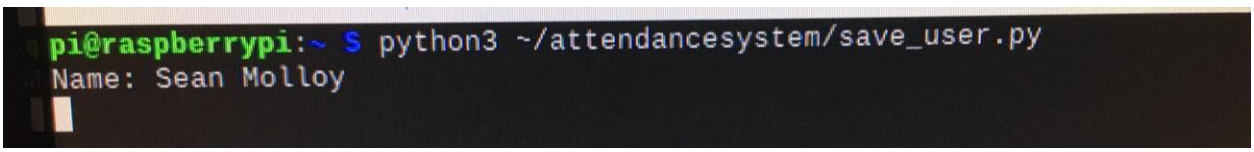

```
20  try:
21    while True:
22      lcd.clear()
23      lcd.message('Place Card over\nSams RFID Reader')
24      id, text = reader.read()
25      cursor.execute("SELECT id FROM users WHERE rfid_uid="+str(id))
26      cursor.fetchone()
27
```

Code snippet save_user.py script

When the employee flashes the card over the reader, they will then be asked to enter their name as shown:



For this example, I've enter the following name:



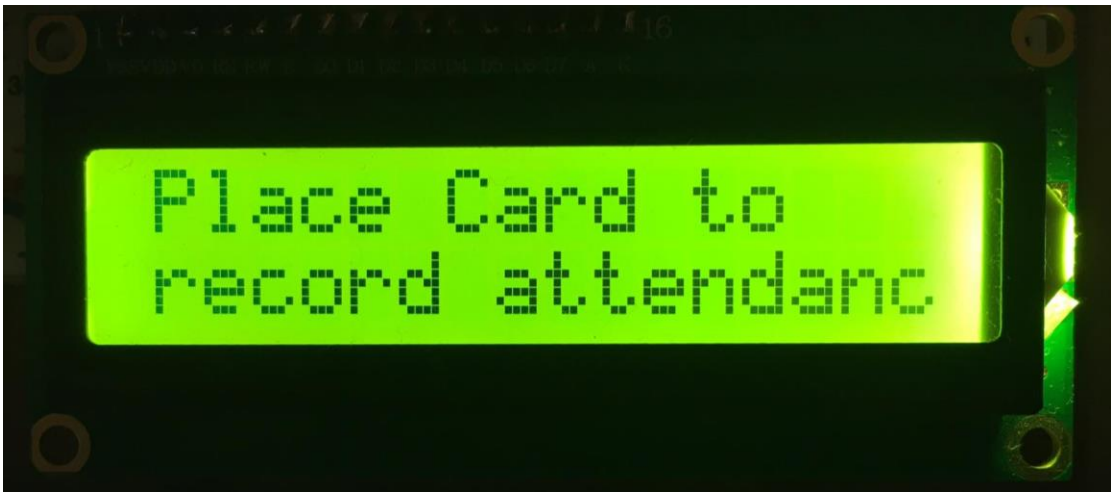
Now that RFID card is now registered to the employee “Sean Molloy”

Now the person and the card are assigned to the database that employee is now set up to capture his daily attendance at work.

This is done by running the check_attendance.py script as shown below:

```
pi@raspberrypi:~ $ python3 ~/attendancesystem/check_attendance.py
```

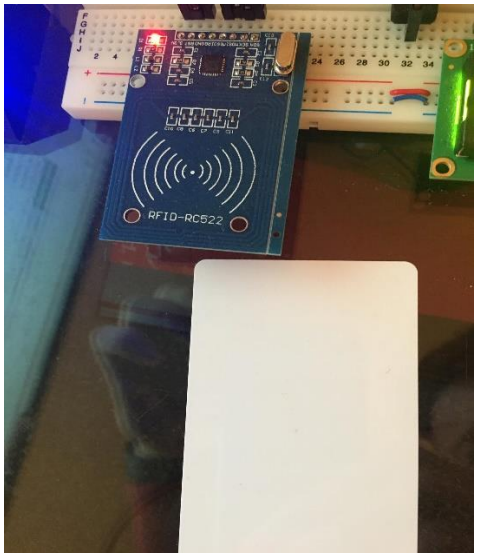
The following message will then appear on the LCD display:




```
20  try:
21    while True:
22      lcd.clear()
23      lcd.message('Place Card to\nrecord attendance')
24      id, text = reader.read()
25
26      cursor.execute("SELECT id, name FROM users WHERE rfid_uid="+str(id))
27      result = cursor.fetchone()
28
```

Code snippet check_attendance.py script

A card will then be swiped over the reader:



“Sean Molloy” is then signed in at work.

All the data is stored on the mySQL database as shown:

```
MariaDB [attendancesystem]> SELECT * FROM users;
+----+--------------+---------------+---------------------+
| id | rfid_uid     | name          | created             |
+----+--------------+---------------+---------------------+
|  1 | 104627605738 | Tom Murphy    | 2020-05-01 15:24:28 |
|  2 | 789736213975 | Tim Ryan      | 2020-05-01 22:37:10 |
|  3 | 858528572170 | Jim Kelly     | 2020-05-01 23:06:01 |
|  4 | 355178537929 | Pat Mannion   | 2020-05-01 23:06:22 |
|  5 | 192164768209 | James Feerick | 2020-05-01 20:07:48 |
|  6 | 1046276057385| Tim Ryan      | 2020-05-11 20:05:56 |
| 11 | 458945618840 | Sean Molloy   | 2020-05-17 15:22:58 |
+----+--------------+---------------+---------------------+
7 rows in set (0.001 sec)
```

To then submit Fingerprint Attendance this is done with the following command:

```
i@raspberrypi:~ $ python2 /usr/share/doc/python-fingerprint/examples/example_er
oll.py
```

This then asks the employee to enter their name:

```
i@raspberrypi:~ $ python2 /usr/share/doc/python-fingerprint/examples/example_er
oll.py
urrently used templates: 6/300
lease enter your name...
```

I then enter the employees name:

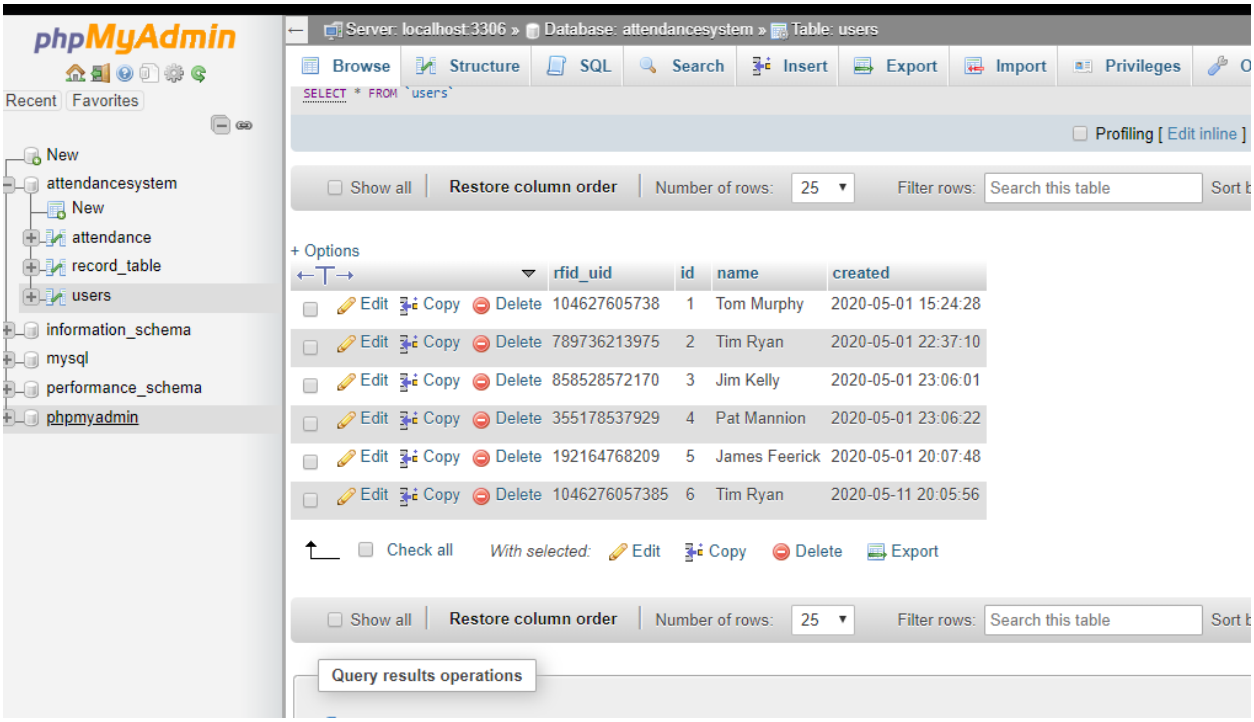
```
@raspberrypi:~ $ python2 /usr/share/doc/python-fingerprint/examp
ll.py
rrently used templates: 6/300
ease enter your name...Joe Fitz
aiting for finger...
```

The employee will the be prompted to place their finger on the fingerprint module:

The persons fingerprint along with their name will then be stored in the database:

```
pi@raspberrypi:~ $ python2 /usr/share/doc/python-fingerprint/examp
roll.py
Currently used templates: 6/300
Please enter your name...Joe Fitz
Waiting for finger...
Remove finger...
Waiting for same finger again...
Finger enrolled successfully!
DB record added!
```

All the data of the two methods of Attendance will then be store and viewed and can also be edited in the PhpMyAdmin webpage.



ID numbers, names etc can be changed here before it is pushed to the website that is run through the Apache Tom Cat server.

9 Problem Solving

I have faced many challenges during the implementation of this project.

- Trying to get fingerprint sensor working.

I had a major problem trying to get the fingerprint sensor working. I had it set up using gpio pins but it wouldn't work as I had the LCD display and the RFID reader already connected using the gpio pins and the pi just wasn't strong enough to power the fingerprint module as well. So, after doing a bit of research I found an alternate way of connecting it through a USB to TTL converter. This fixed my problem

- Latency problems with the Pi.

As I have quite a lot of hardware in the project the Raspberry Pi has shut down on me multiple times, leading to loss of code. I fixed this problem through backing up all my work on the cloud as often as I could.

- Problem with the LCD display

I had a problem trying to get the LCD display trying to work. The messages that I was trying to produce on the display were not displaying. After rewiring it multiple times, it turned out to be a potentiometer problem.

10 Conclusion

In this project, the employee attendance monitor is developed using Python and PHP scripting languages, HTML and CSS is used for designing the website which have met the system goals and objectives. A simple user-friendly website is designed and easily accessible. The system offers a change in the way employers can have their employees register their attendance on a daily basis through the RFID card and if they were to lose the card they can sign in using their fingerprint, this is what makes this product unique.

The Employee Attendance Monitor that is developed is fully responsive and meets all expectations. A lot of research was done on RFID technology to try and implement a way of utilizing it towards an easier way of monitoring attendance. The RC522 RFID module works very well in this project. It provides a much easier way for the employer to gather attendance as all the employee has to do is swipe their card on the reader on the way into the production facility.

Adding the fingerprint sensor as an alternate method of attendance works very well in this project. It gives the employee the option to sign in even if they were to lose their RFID card.

Looking at future development of this project, I would say there is a lot of scope for development. The website could be much more advanced as there are lots more features that could be added such as a page to view employees attendance for the that month, a place to view if employee's are late, the list is endless what can be added to the website.

Overall, I am moderately happy with the finished product. If I had more time I would have tried developing the project more in all areas especially the website but in reflection I think the concept that I created is good and my project meets all my goals for a final year project in align with where my software coding skills are at.

Also, I think this project has been hugely beneficial to me. My coding skills have improved drastically. My project management skills have improved also.

11 References

[1] Raspberry Pi, available:

https://www.raspberrypi.org/help/

[2] Raspberry Pi, available:

https://en.wikipedia.org/wiki/Raspberry_Pi

[3] Python Programming on Raspberry Pi, available:

https://www.raspberrypi.org/documentation/usage/python/

[4] Python Programming on Raspberry Pi, available:

https://pythonprogramming.net/introduction-raspberry-pi-tutorials/

[5] LCD Display on Raspberry Pi, available:

https://www.raspberrypi.org/forums/search.php?keywords=lcd&sid=e20c712256210e5d5d28740ce65595b3

[6] RFID on Raspberry Pi, available:

https://www.raspberrypi.org/forums/search.php?keywords=rfid

[7] RFID on Raspberry Pi, available:

https://www.raspberrypi-spy.co.uk/2018/02/rc522-rfid-tag-read-raspberry-pi/

[8] Fingerprint module on Raspberry Pi, available:

https://tutorials-raspberrypi.com/how-to-use-raspberry-pi-fingerprint-sensor-authentication/

[9] PHP on Raspberry Pi, available:

https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md

[10] MySQL, available:

https://www.raspberrypi.org/forums/viewtopic.php?t=263371

[11] PhpMyAdmin, available:

https://www.phpmyadmin.net/

[12] CSS & HTML, available:

https://www.w3schools.com/html/html_css.asp

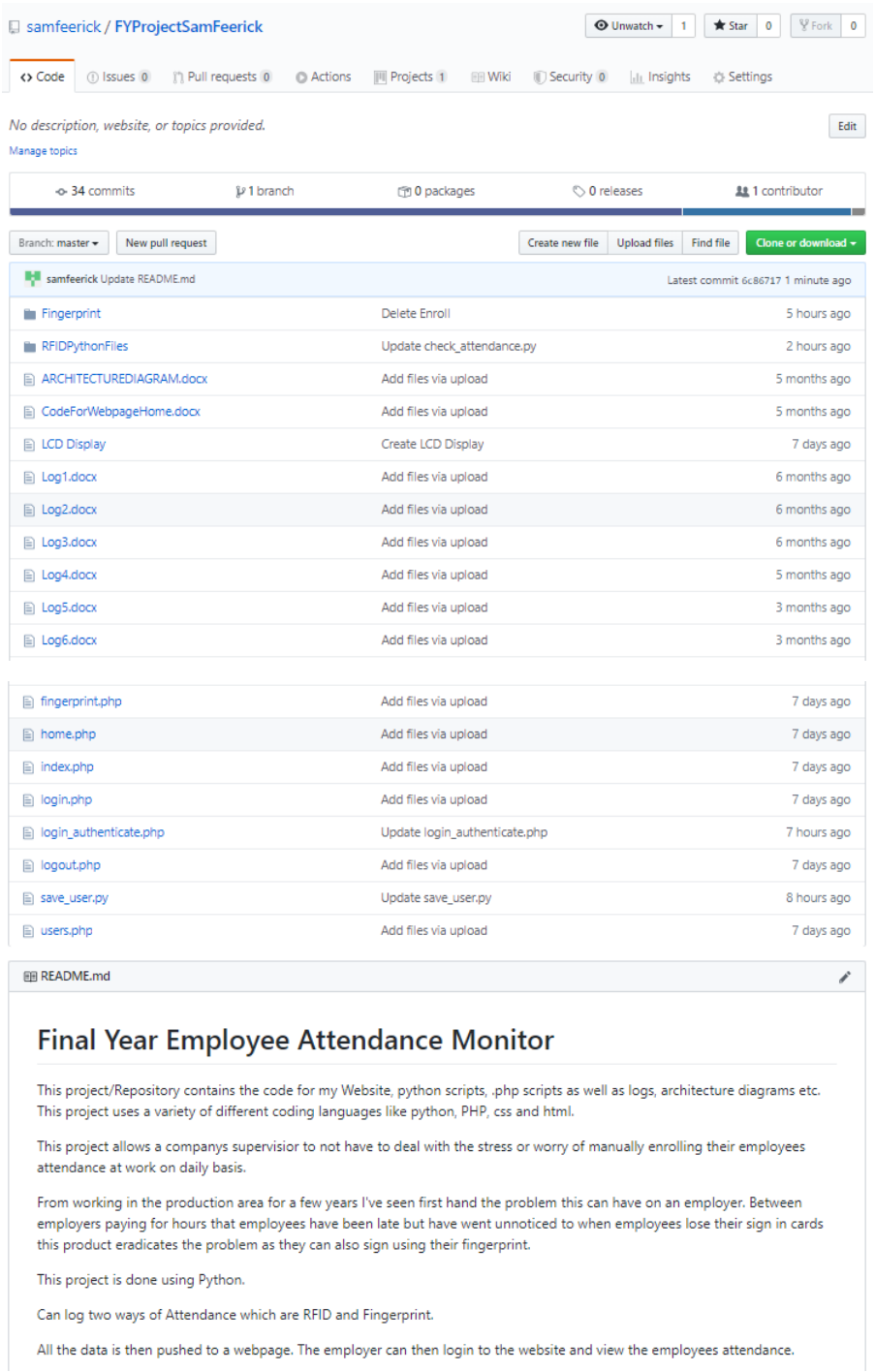
[13] Apache Tomcat Server on Raspberry Pi, available:

http://tomcat.apache.org/

12 Code

All the code for my project can be found on my GitHub account using the following link

https://github.com/samfeerick/FYProjectSamFeerick



samfeerick / FYProjectSamFeerick

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 1 | Wiki | Security 0 | Insights | Settings

No description, website, or topics provided. Edit

Manage topics

34 commits | 1 branch | 0 packages | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

samfeerick Update README.md — Latest commit 6c86717 1 minute ago

Fingerprint	Delete Enroll	5 hours ago
RFIDPythonFiles	Update check_attendance.py	2 hours ago
ARCHITECTUREDIAGRAM.docx	Add files via upload	5 months ago
CodeForWebpageHome.docx	Add files via upload	5 months ago
LCD Display	Create LCD Display	7 days ago
Log1.docx	Add files via upload	6 months ago
Log2.docx	Add files via upload	6 months ago
Log3.docx	Add files via upload	6 months ago
Log4.docx	Add files via upload	5 months ago
Log5.docx	Add files via upload	3 months ago
Log6.docx	Add files via upload	3 months ago
fingerprint.php	Add files via upload	7 days ago
home.php	Add files via upload	7 days ago
index.php	Add files via upload	7 days ago
login.php	Add files via upload	7 days ago
login_authenticate.php	Update login_authenticate.php	7 hours ago
logout.php	Add files via upload	7 days ago
save_user.py	Update save_user.py	8 hours ago
users.php	Add files via upload	7 days ago

README.md

Final Year Employee Attendance Monitor

This project/Repository contains the code for my Website, python scripts, .php scripts as well as logs, architecture diagrams etc. This project uses a variety of different coding languages like python, PHP, css and html.

This project allows a companys supervisior to not have to deal with the stress or worry of manually enrolling their employees attendance at work on daily basis.

From working in the production area for a few years I've seen first hand the problem this can have on an employer. Between employers paying for hours that employees have been late but have went unnoticed to when employees lose their sign in cards this product eradicates the problem as they can also sign using their fingerprint.

This project is done using Python.

Can log two ways of Attendance which are RFID and Fingerprint.

All the data is then pushed to a webpage. The employer can then login to the website and view the employees attendance.