ARDENT
COMPUTECH PVT. LTD.
High-End Technology Training and Project

Ardent Computech Pvt. Ltd.
Drives you to the Industry
Module 132, SDF Building, Sector V, Salt Lake, Pin - 700091
www.ardentcollaborations.com

CERTIFIED ISO 9001:2015 CERTIFIED

UDYAM REGISTRATION
MSME

N·E·A·T
National Educational Alliance for Technology

SMART ATTENDANCE SYSTEM

A Project Report for Industrial Training and Internship

Submitted by

Rima Das
Sahali Das
Sanjita Ray
Piyali Kundu
Ratnadeep Das
Sagar Bera

In the partial fulfillment of the award of the degree of

B.Tech

in the

Computer Science & Engineering

of

Dr. Sudhir Chandra Sur Institute of Technology and Sports Complex

SurTech
JIS GROUP

at

Ardent Computech Pvt. Ltd.

Ardent Computech Pvt. Ltd.
Drives you to the Industry
Module 132, SDF Building, Sector V, Salt Lake, Pin - 700091
www.ardentcollaborations.com

ARDENT®
GROUP OF COMPANIES
INTERNSHIP | DEVELOPMENT | PROJECT
An ISO 9001 : 2015 Certified Company

CERTIFICATE FROM SUPERVISOR

This is to certify that **“Rima Das” (University Registration Number: 222550110118), “Sanjita Ray” (University Registration Number: 222550110138), “Sahali Das” (University Registration Number: 222550110128), “Sagar Bera” (University Registration Number: 222550110126), “Ratnadeep Das” (University Registration Number: 222550110114), “Piyali Kundu” (University Registration Number: 222550110103)** have completed the project titled "**Smart Attendance System**" under my supervision during the period from “12/08/2024” to “23/08/2024” which is in partial fulfillment of requirements for the award of the **B.Tech/MCA/BCA** degree and submitted to the Department of **Computer Science & Engineering** of **Dr. Sudhir Chandra Sur Institute of Technology and Sports Complex**.

Signature of the Supervisor

Date: dd/mm/yy

Name of the Project Supervisor:

ARDENT
COMPUTECH PVT. LTD.
High-End Technology Training and Project

Ardent Computech Pvt. Ltd.
Drives you to the Industry
Module 132, SDF Building, Sector V, Salt Lake, Pin - 700091
www.ardentcollaborations.com

CERTIFIED ISO 9001:2015 CERTIFIED

UDYAM REGISTRATION MSME

N·E·A·T
National Educational Alliance for Technology

BONAFIDE CERTIFICATE

Certified that this project work was carried out under my supervision

“SMART ATTENDANCE SYSTEM” is the bonafide work of

• **Name of the student: RIMA DAS**	**Signature: Rima Das**
• **Name of the student: SAHALI DAS**	**Signature: Sahali Das**
• **Name of the student: SANJITA RAY**	**Signature: Sanjita Ray**
• **Name of the student: PIYALI KUNDU**	**Signature: Piyali Kundu**
• **Name of the student: RATNADEEP DAS**	**Signature: Ratnadeep Das**
• **Name of the student: SAGAR BERA**	**Signature: Sagar Bera**

SIGNATURE

UDYOG AADHAR MSME ISO 9001 2015 CERTIFIED Microsoft Technology Associate Microsoft Office Specialist AUTODESK Certified User CERTIFIED ASSOCIATE Hewlett Packard Enterprise

Name:

PROJECT MENTOR

SIGNATURE

Name:

EXAMINERS

Ardent Original Seal

<u>ACKNOWLEDGEMENT</u>

The achievement that is associated with the successful completion of any task would be incomplete without mentioning the names of those people whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful.

We take this opportunity to express our gratitude towards our project mentor, ***SK Suman*** for giving such valuable suggestions, guidance and encouragement during the development of this project work.

Last but not the least we are grateful to all the faculty members of **Ardent Computech Pvt. Ltd.** for their support.

ABSTRACT

The "Smart Attendance System" is an innovative solution designed to automate and streamline the attendance tracking process in educational institutions and workplaces. This system utilizes facial recognition technology, which ensures accuracy and efficiency in recording attendance. By eliminating manual attendance procedures, it minimizes the chances of errors, proxy attendance, and time consumption, thereby enhancing overall productivity.

The project leverages advanced algorithms for facial recognition, integrating them with a user-friendly interface that can be accessed via mobile devices or desktop platforms. The system not only records attendance in real-time but also provides data analytics and reports, allowing administrators to monitor attendance patterns and identify irregularities. The use of cloud-based storage ensures that attendance data is securely stored and easily accessible for future reference.

In addition to improving accuracy and saving time, the Smart Attendance System promotes a contactless and seamless experience, which is particularly beneficial in the post-pandemic era. Its adaptability across different environments makes it a versatile tool for various institutions, contributing to better resource management and decision-making processes.

CONTENTS

INTRODUCTION

The "Smart Attendance System" is a cutting-edge project aimed at revolutionizing the way attendance is managed in educational institutions and workplaces. Traditional attendance methods, such as manual roll calls or swipe cards, are often time-consuming, prone to errors, and vulnerable to manipulation. In contrast, the Smart Attendance System leverages modern technology to automate and streamline the attendance process, ensuring accuracy, efficiency, and reliability.

This system primarily uses facial recognition technology, which scans and identifies individuals in real-time, recording their attendance automatically. By eliminating the need for physical interaction, such as signing sheets or swiping cards, the system not only saves time but also enhances security by reducing the risk of proxy attendance. The integration of facial recognition with cloud-based storage allows for easy access to attendance records and data analytics, enabling administrators to monitor attendance patterns, generate reports, and make informed decisions.

The Smart Attendance System is designed to be user-friendly and adaptable to various environments, from classrooms to corporate offices. Its contactless nature is particularly relevant in the current global context, where minimizing physical contact is crucial for health and safety. Furthermore, the system can be customized to meet the specific needs of different organizations, making it a versatile tool for improving operational efficiency and resource management. Overall, the Smart Attendance System represents a significant advancement in attendance management, offering a modern solution to a traditional challenge.

COMPONENTS USED

- Wi-Fi enabled device
- Microcontroller (Raspberry Pi Pico W)
- Bread Board
- LED
- Buzzer
- RFID
- 16x2 LCD with I2C driver
- Male-Male & Male-Female Wires
- Header
- USB Type-B Cable

SOFTWARE USED

- Thonny Python Code Editor
- Visual Studio Code
- Django
- Firebase

DESCRIPTION

Raspberry Pi Pico W:

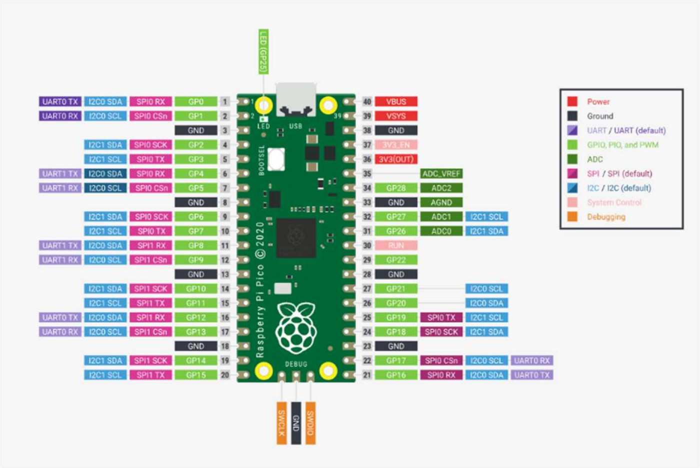
Raspberry Pi Pico W adds on-board single-band 2.4GHz wireless interfaces (802.11n) using the Infineon CYW43439 while retaining the Pico form factor. The on-board 2.4GHz wireless interface has the following features:

- Wireless (802.11n), single-band (2.4 GHz)
- WPA3
- Soft access point supporting up to four clients
- Bluetooth 5.2
  - Support for Bluetooth LE Central and Peripheral roles
  - Support for Bluetooth Classic

The antenna is an onboard antenna licensed from ABRACON (formerly ProAnt). The wireless interface is connected via SPI to the RP2040 microcontroller.

Due to pin limitations, some of the wireless interface pins are shared. The CLK is shared with VSYS monitor, so only when there isn't an SPI transaction in progress can VSYS be read via the ADC. The Infineon CYW43439 DIN/DOUT and IRQ all share one pin on the RP2040. Only when an SPI transaction isn't in progress is it suitable to check for IRQs. The interface typically runs at 33MHz.

For best wireless performance, the antenna should be in free space. For instance, putting metal under or close by the antenna can reduce its performance both in terms of gain and bandwidth. Adding grounded metal to the sides of the antenna can improve the antenna's bandwidth.



DESCRIPTION

Breadboard:

A breadboard is a reusable tool for prototyping electronic circuits, featuring a grid of holes that allow for the insertion of components without soldering. Its design includes interconnected rows and columns that simplify the process of connecting various elements like resistors, LEDs, and microcontrollers. Breadboards are essential for experimenting with circuit designs, making it easy to test and modify configurations before creating a permanent version.

LED:

An LED (Light Emitting Diode) is a semiconductor device that emits light when an electric current flows through it. Unlike traditional bulbs, LEDs are highly energy-efficient, have a longer lifespan, and are available in various colors. They are commonly used in displays, indicators, and lighting solutions. LEDs are polarity-sensitive, requiring correct orientation in a circuit, and are popular for their low power consumption and durability.

Buzzer:

A buzzer is an electronic component that produces sound when an electric current passes through it. Available in active and passive types, buzzers are commonly used in alarms, notifications, and timers. They convert electrical energy into sound waves, typically producing a tone, beep, or buzz. Buzzers are widely used in electronics for their simplicity, reliability, and ability to draw attention to important signals or events.

DESCRIPTION

RFID:

Radio Frequency Identification (RFID) refers to a wireless system comprised of two components: tags and readers. The reader is a device that has one or more antennas that emit radio waves and receive signals back from the RFID tag. Tags, which use radio waves to communicate their identity and other information to nearby readers, can be passive or active. Passive RFID tags are powered by the reader and do not have a battery. Active RFID tags are powered by batteries.

RFID tags can store a range of information from one serial number to several pages of data. Readers can be mobile so that they can be carried by hand, or they can be mounted on a post or overhead. Reader systems can also be built into the architecture of a cabinet, room, or building.

Uses: RFID systems use radio waves at several different frequencies to transfer data. In health care and hospital settings, RFID technologies include the following applications:

- Inventory control
- Equipment tracking
- Out-of-bed detection and fall detection
- Personnel tracking
- Ensuring that patients receive the correct medications and medical devices
- Preventing the distribution of counterfeit drugs and medical devices
- Monitoring patients
- Providing data for electronic medical records systems
- The FDA is not aware of any adverse events associated with RFID. However, there is concern about the potential hazard of electromagnetic interference (EMI) to electronic medical devices from radio frequency transmitters like RFID. EMI is a degradation of the performance of equipment or systems (such as medical devices) caused by an electromagnetic disturbance.

DESCRIPTION

16x2 LCD with I2C driver:

16×2 LCD is an Alphanumeric display that can show up to 32 characters on a single screen. You can display more characters by scrolling the texts one by one. The I2C Module is used to reduce the no. of pins needed for the display. It enables the display to work with only four pins.
The LCD Display can be used with all Microcontroller boards like 8051, AVR, Arduino, PIC, and ARM Microcontrollers.
Most projects require an LCD display to communicate with the user in a better way. Some projects required to display warnings, errors, Sensor values, State of the input and output device, Selecting different modes of operations, Time and date display, Alert message and many more. This will give the project a better view and its operation in a more visual way.
A 16×2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5×7-pixel matrix. This LCD has two registers, namely, Command and Data.
The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc.
The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

Male-Male & Male-Female Wires:

Male-Male and Male-Female wires are essential components for building and testing electronic circuits. Male-Male wires have pins at both ends, allowing direct connections between different components on a breadboard or to other male connectors. Male-Female wires, on the other hand, have a pin on one end and a socket on the other, making them ideal for connecting components with female headers, such as microcontrollers. These jumper wires are crucial for flexible and quick circuit assembly in prototyping.

DESCRIPTION

Header Wires:

Header wires are essential connectors in electronic circuits, typically used to establish connections between different components on a circuit board or between a board and external devices. These wires have pin connectors, which can be either male or female, designed to fit into corresponding headers on circuit boards. Male header wires connect to female headers, while female header wires connect to male pins. These versatile wires are used in breadboarding, prototyping, and permanent circuit installations, providing reliable and easy-to-manage connections. Header wires are particularly useful in modular designs, where components need to be easily connected, disconnected, or replaced, ensuring flexibility and efficiency in electronic projects.

USB Type-B Cable:

A USB Type-B cable is a standard cable used for connecting various devices to a computer or power source, commonly found in older printers, scanners, and some external hard drives. The cable has a square-shaped Type-B connector on one end, which plugs into the device, and typically a USB Type-A connector on the other end, which connects to the computer or power supply. USB Type-B cables are known for their durability and reliability, providing a stable data transfer and power connection. They have been a critical component in computer peripherals, ensuring that devices can communicate with computers effectively and securely.

DESCRIPTION

Thonny Python Code Editor:

Thonny is a user-friendly Python code editor designed to simplify the coding experience, particularly for beginners and educators. Developed at the University of Tartu in Estonia, Thonny offers an intuitive interface that helps new programmers understand and learn Python programming without being overwhelmed by complex features. Its straightforward design focuses on ease of use, making it ideal for those who are just starting their journey in coding.

One of Thonny's standout features is its simple, clutter-free interface, which allows users to focus on writing and understanding code. It comes with a built-in Python interpreter, so there's no need for additional setup, enabling users to start coding right away. Thonny also provides a unique step-through debugger that lets users execute code line by line, making it easier to identify errors and understand how the code works. This feature is particularly beneficial for beginners, as it visually breaks down the execution process, highlighting variable changes and program flow.

In addition to its beginner-friendly features, Thonny supports more advanced functionalities as users grow more confident in their coding skills. It includes tools for managing packages and libraries, allowing users to easily install and update Python packages. The editor also supports various Python versions, making it flexible for different project requirements.

Thonny's simplicity doesn't come at the expense of functionality. It is lightweight yet powerful enough to handle a wide range of Python development tasks. Its built-in support for MicroPython and CircuitPython further expands its utility, making it suitable for programming microcontrollers like the BBC micro:bit and Raspberry Pi Pico.

Overall, Thonny serves as an excellent tool for learning and teaching Python, combining simplicity with powerful features that cater to both beginners and more experienced developers.

DESCRIPTION

Visual Studio Code:

Visual Studio Code (VS Code) is a highly popular, open-source code editor developed by Microsoft, known for its versatility, powerful features, and extensive customization options. Designed to support a wide range of programming languages and development frameworks, VS Code is widely used by developers for tasks ranging from simple scripting to full-scale software development. Its appeal lies in its lightweight design, yet it is packed with features typically found in more complex integrated development environments (IDEs).

One of VS Code's most significant strengths is its rich extension ecosystem. Developers can enhance the editor's functionality by installing extensions for various programming languages, frameworks, tools, and themes. These extensions allow VS Code to be tailored to individual workflows, making it adaptable for different development environments, including web development, Python, JavaScript, Java, and many others. The integrated terminal, debugger, and Git control features further streamline the development process, enabling users to write, test, and deploy code all within the same environment. VS Code's intuitive interface is designed for efficiency, featuring a customizable layout that allows developers to organize their workspace according to their preferences. The editor offers intelligent code completion, syntax highlighting, and real-time error detection, all of which significantly boost productivity and reduce coding errors. Additionally, its powerful search and navigation capabilities make it easy to manage large codebases.

Another standout feature is VS Code's seamless integration with Git and other version control systems, enabling developers to manage their code repositories directly from the editor. This integration simplifies collaboration on projects, making it easier to track changes, resolve conflicts, and maintain code quality.

Overall, Visual Studio Code is a robust and flexible tool that caters to developers of all skill levels, offering a seamless and efficient coding experience across various programming languages and frameworks.

DESCRIPTION

Django:

Django is a high-level, open-source web framework written in Python, designed to facilitate the rapid development of secure and scalable web applications. Created by experienced developers, Django emphasizes reusability, modularity, and the "Don't Repeat Yourself" (DRY) principle, which helps developers build complex applications with minimal code duplication. Its robust feature set includes an ORM (Object-Relational Mapping) system, authentication mechanisms, an admin interface, and tools for managing forms, sessions, and cookies.

One of Django’s core strengths is its “batteries-included” philosophy, meaning it comes with many built-in features that simplify common web development tasks, such as URL routing, templating, and database migrations. This reduces the need for third-party tools and allows developers to focus on writing application-specific code rather than reinventing the wheel.

Django also prioritizes security, offering protections against common web vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF) out of the box. Its user authentication system is robust, making it easier to manage users, groups, and permissions securely.

Django's scalability makes it suitable for both small projects and large-scale applications, with a community that continually contributes to its rich ecosystem of plugins and extensions. Additionally, Django's documentation is extensive and beginner-friendly, making it accessible for developers of all experience levels.

Overall, Django is a powerful framework that enables developers to build web applications quickly and efficiently, with a strong emphasis on security, scalability, and maintainability.

DESCRIPTION

Firebase:

Firebase is a comprehensive platform developed by Google for building and managing web and mobile applications. It provides a suite of cloud-based tools and services that streamline the development process, allowing developers to focus on creating high-quality applications without worrying about backend infrastructure. Firebase offers a range of features, including real-time databases, authentication, cloud storage, hosting, and analytics, making it a one-stop solution for modern app development.

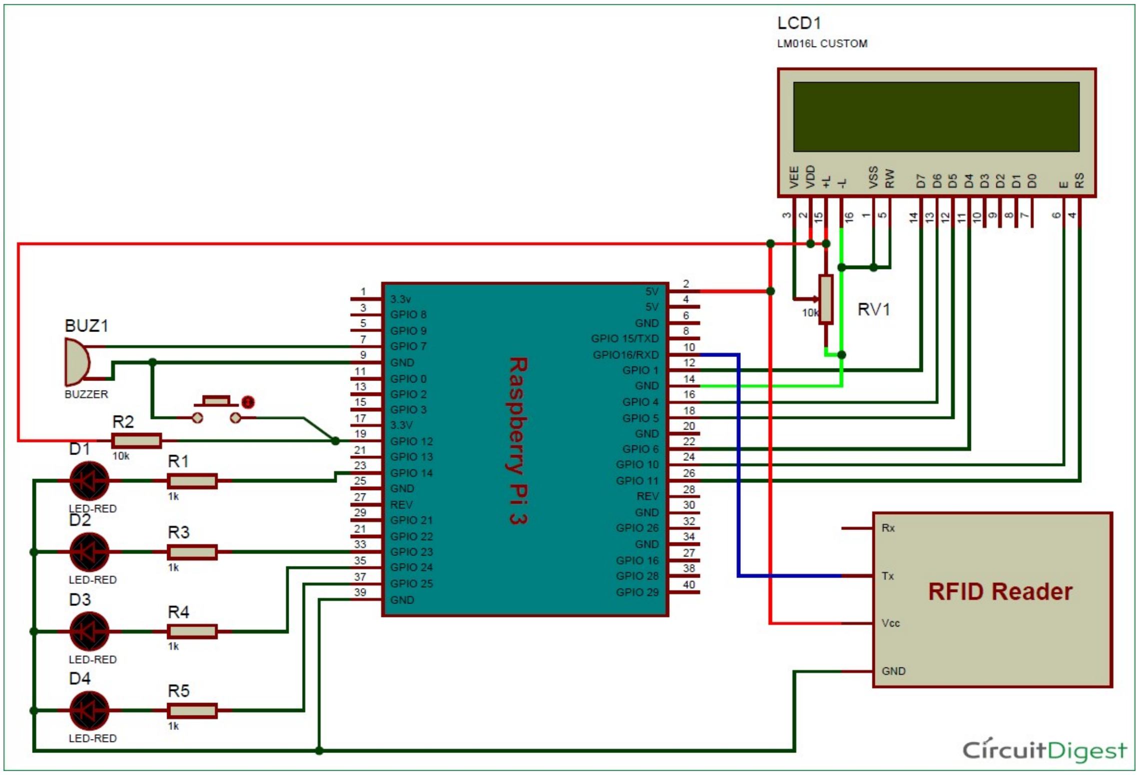
One of Firebase's standout features is its Realtime Database, which allows developers to store and sync data across all clients in real-time. This is particularly useful for applications that require live updates, such as chat apps or collaborative tools. Firebase also supports Firestore, a more flexible, scalable NoSQL database for more complex data needs. Both databases are fully managed and offer offline support, ensuring seamless user experiences even when connectivity is unstable.

Firebase's Authentication service simplifies the process of user management by providing ready-to-use authentication methods, including email and password, phone number, and social media logins like Google and Facebook. This allows developers to quickly implement secure user authentication without having to build it from scratch.

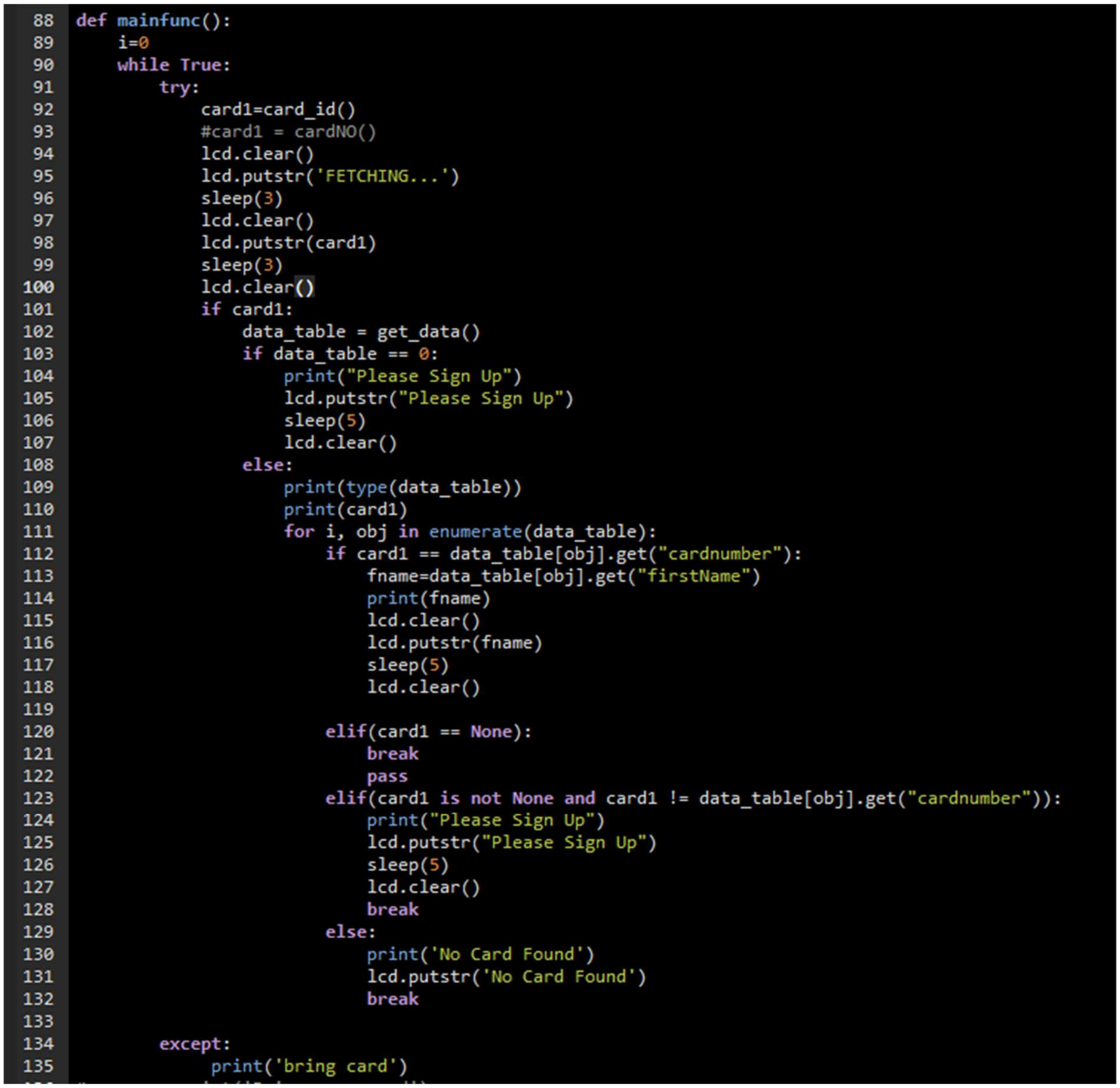
In addition, Firebase offers powerful tools like Cloud Functions, which enable developers to run backend code in response to events triggered by Firebase features or HTTPS requests, and Firebase Hosting, which delivers fast and secure static hosting for web apps.

Overall, Firebase is a versatile and powerful platform that accelerates the development process, offering a scalable and secure backend infrastructure that can grow with the needs of your application.

CIRCUIT DIAGRAM



CODE SNIPPETS

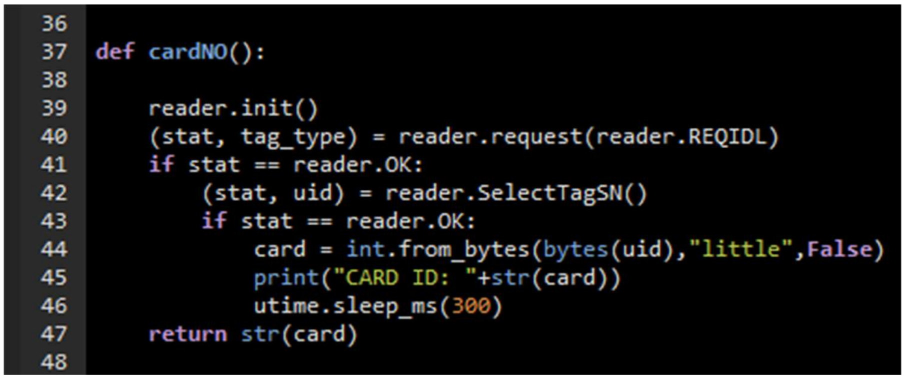


```python
def mainfunc():
    i=0
    while True:
        try:
            card1=card_id()
            #card1 = cardNO()
            lcd.clear()
            lcd.putstr('FETCHING...')
            sleep(3)
            lcd.clear()
            lcd.putstr(card1)
            sleep(3)
            lcd.clear()
            if card1:
                data_table = get_data()
                if data_table == 0:
                    print("Please Sign Up")
                    lcd.putstr("Please Sign Up")
                    sleep(5)
                    lcd.clear()
                else:
                    print(type(data_table))
                    print(card1)
                    for i, obj in enumerate(data_table):
                        if card1 == data_table[obj].get("cardnumber"):
                            fname=data_table[obj].get("firstName")
                            print(fname)
                            lcd.clear()
                            lcd.putstr(fname)
                            sleep(5)
                            lcd.clear()

                        elif(card1 == None):
                            break
                            pass
                        elif(card1 is not None and card1 != data_table[obj].get("cardnumber")):
                            print("Please Sign Up")
                            lcd.putstr("Please Sign Up")
                            sleep(5)
                            lcd.clear()
                            break
                        else:
                            print('No Card Found')
                            lcd.putstr('No Card Found')
                            break

        except:
             print('bring card')
```

The base function of this Smart Attendance System is this main function. At first, we are fetching the card id. We have a temporary firebase bucket named “Card”.

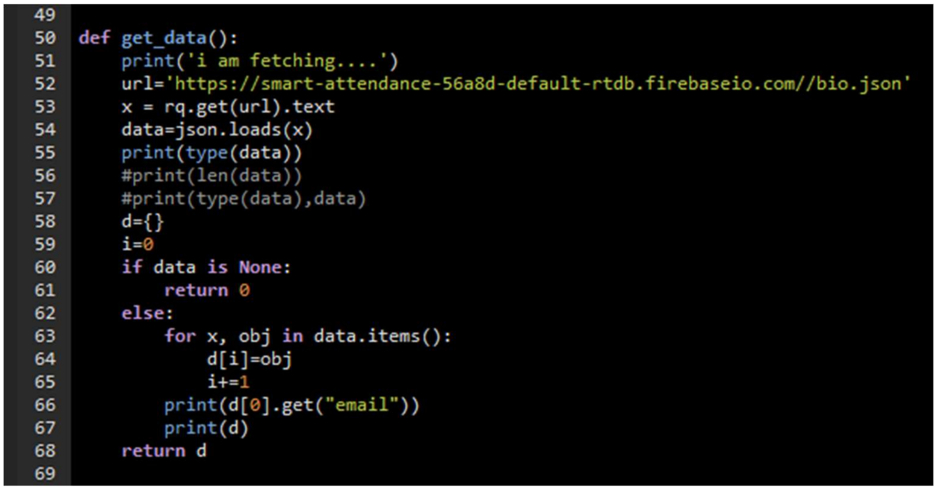
We are accessing the LCD part. We are giving a condition to compare two card numbers. 1st one is the card number that is given by the user and the other one is the already existed one in the firebase bio bucket.

If the card number given by the user matches with the existed one it greets user on the LCD. If the numbers do not match, we need to sign up to become a part of our smart attendance system. If user does not swap any card on the RFID card, then it shows “Bring Card”.



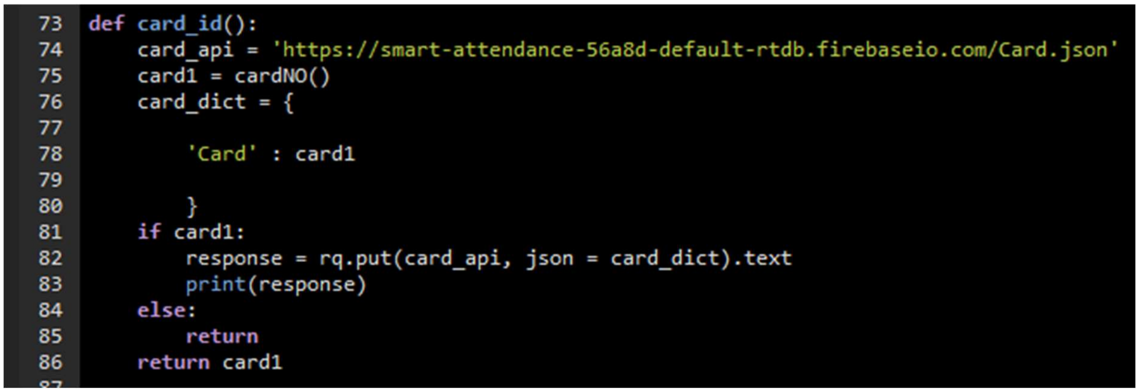
```
def cardNO():

    reader.init()
    (stat, tag_type) = reader.request(reader.REQIDL)
    if stat == reader.OK:
        (stat, uid) = reader.SelectTagSN()
        if stat == reader.OK:
            card = int.from_bytes(bytes(uid),"little",False)
            print("CARD ID: "+str(card))
            utime.sleep_ms(300)
    return str(card)
```

We are fetching the card id via RFID card. When user swaps the card over the RFID then by using this function the card id is fetched.



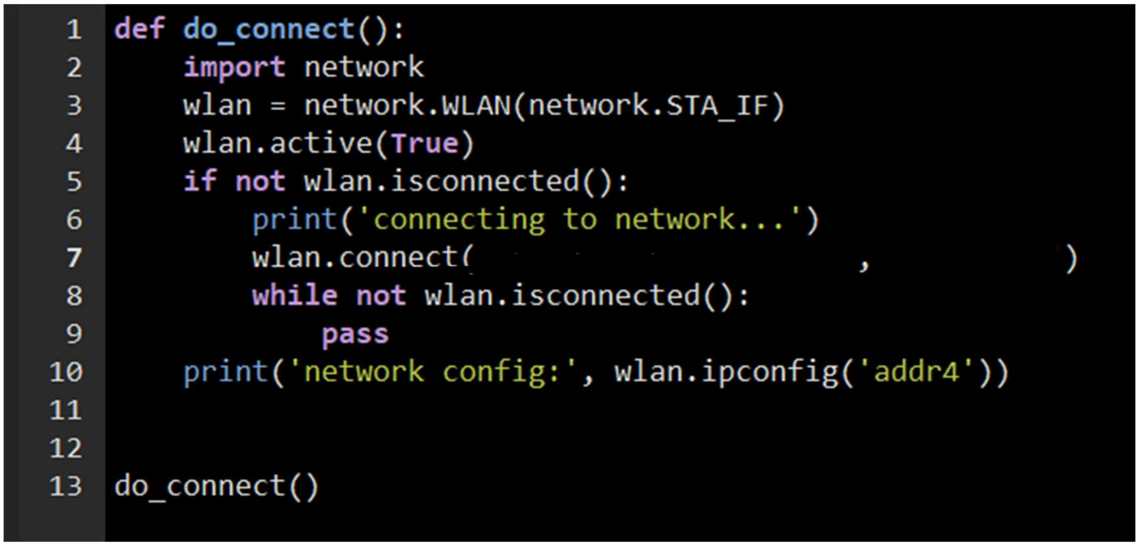
```
def get_data():
    print('i am fetching....')
    url='https://smart-attendance-56a8d-default-rtdb.firebaseio.com//bio.json'
    x = rq.get(url).text
    data=json.loads(x)
    print(type(data))
    #print(len(data))
    #print(type(data),data)
    d={}
    i=0
    if data is None:
        return 0
    else:
        for x, obj in data.items():
            d[i]=obj
            i+=1
        print(d[0].get("email"))
        print(d)
    return d
```

We are using this function to get all the details of the user from the firebase bucket.



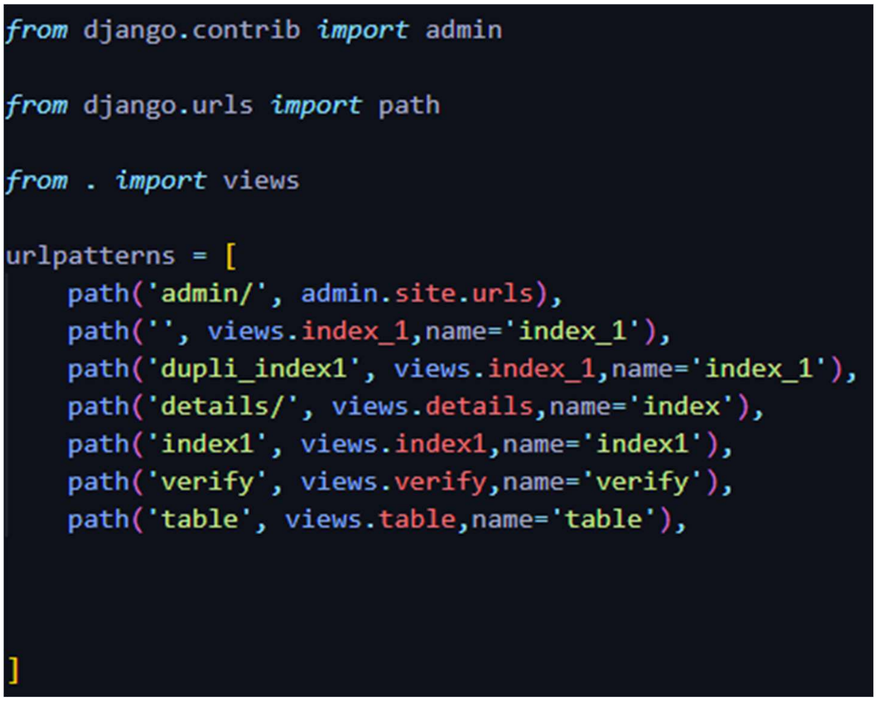
```python
def card_id():
    card_api = 'https://smart-attendance-56a8d-default-rtdb.firebaseio.com/Card.json'
    card1 = cardNO()
    card_dict = {

        'Card' : card1

        }
    if card1:
        response = rq.put(card_api, json = card_dict).text
        print(response)
    else:
        return
    return card1
```

With the help of this function, we are storing the card number given by the user temporarily in the firebase Card bucket by using put method.



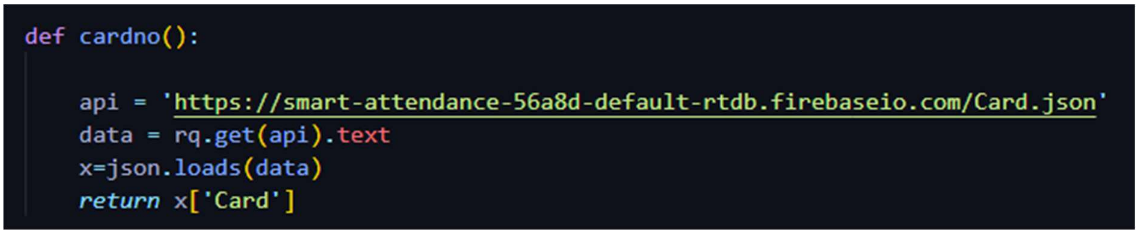
```python
def do_connect():
    import network
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    if not wlan.isconnected():
        print('connecting to network...')
        wlan.connect([redacted], [redacted])
        while not wlan.isconnected():
            pass
    print('network config:', wlan.ipconfig('addr4'))


do_connect()
```

With the help of this function, we are connecting network to the microcontroller using device hotspot.



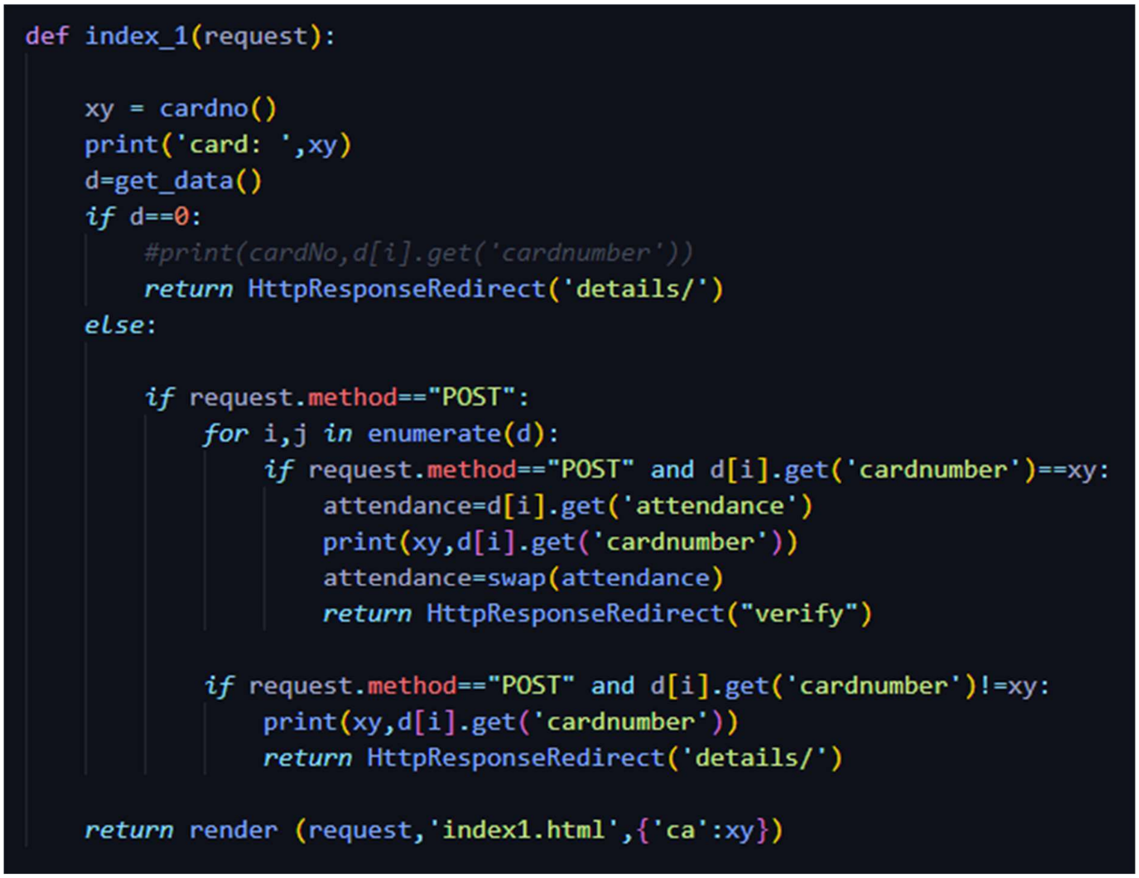
```python
from django.contrib import admin

from django.urls import path

from . import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index_1,name='index_1'),
    path('dupli_index1', views.index_1,name='index_1'),
    path('details/', views.details,name='index'),
    path('index1', views.index1,name='index1'),
    path('verify', views.verify,name='verify'),
    path('table', views.table,name='table'),



]
```

The working function of this file is to make urls for each page of our Smart Attendance System.



```python
def cardno():

    api = 'https://smart-attendance-56a8d-default-rtdb.firebaseio.com/Card.json'
    data = rq.get(api).text
    x=json.loads(data)
    return x['Card']
```

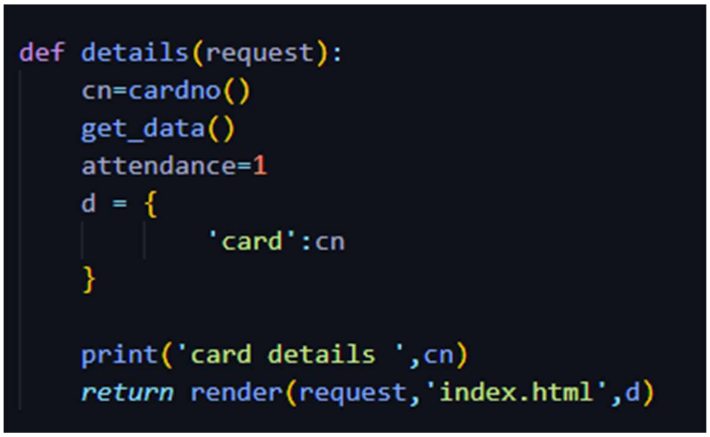
This function is fetching the Card Number from the Card Bucket of Firebase.



```
def index_1(request):

    xy = cardno()
    print('card: ',xy)
    d=get_data()
    if d==0:
        #print(cardNo,d[i].get('cardnumber'))
        return HttpResponseRedirect('details/')
    else:

        if request.method=="POST":
            for i,j in enumerate(d):
                if request.method=="POST" and d[i].get('cardnumber')==xy:
                    attendance=d[i].get('attendance')
                    print(xy,d[i].get('cardnumber'))
                    attendance=swap(attendance)
                    return HttpResponseRedirect("verify")

            if request.method=="POST" and d[i].get('cardnumber')!=xy:
                print(xy,d[i].get('cardnumber'))
                return HttpResponseRedirect('details/')

    return render (request,'index1.html',{'ca':xy})
```

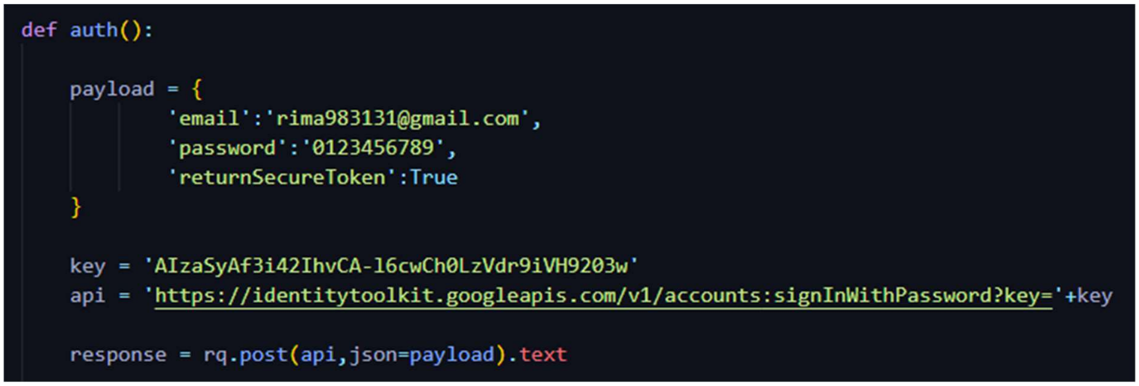
In this function we are accessing the LCD part. We are giving a condition to compare two card numbers. 1st one is the card number that is given by the user and the other one is the already existed one in the firebase bio bucket.

If the card number given by the user matches with the existed one it greets user on the LCD. If the numbers do not match, we need to sign up to become a part of our smart attendance system. If user does not swap any card on the RFID card, then it shows “Bring Card”.



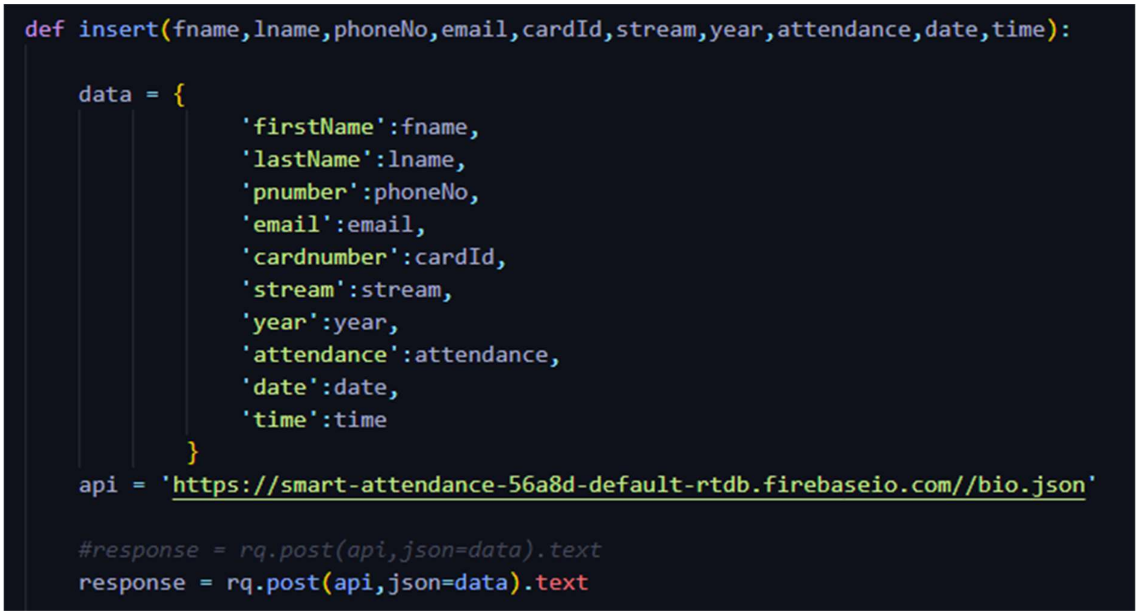
```
def details(request):
    cn=cardno()
    get_data()
    attendance=1
    d = {
            'card':cn
    }

    print('card details ',cn)
    return render(request,'index.html',d)
```

This function is rendering the registration page with card details.



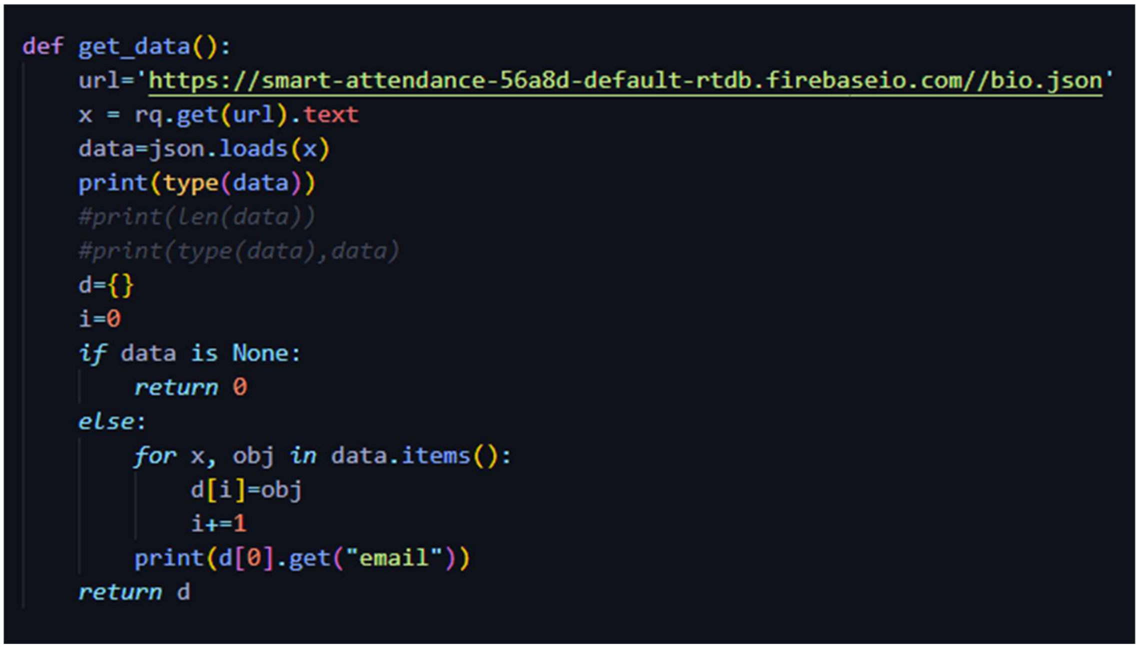
```
def auth():

    payload = {
            'email':'rima983131@gmail.com',
            'password':'0123456789',
            'returnSecureToken':True
    }

    key = 'AIzaSyAf3i42IhvCA-l6cwCh0LzVdr9iVH9203w'
    api = 'https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?key='+key

    response = rq.post(api,json=payload).text
```

This function authenticates the admin in the firebase



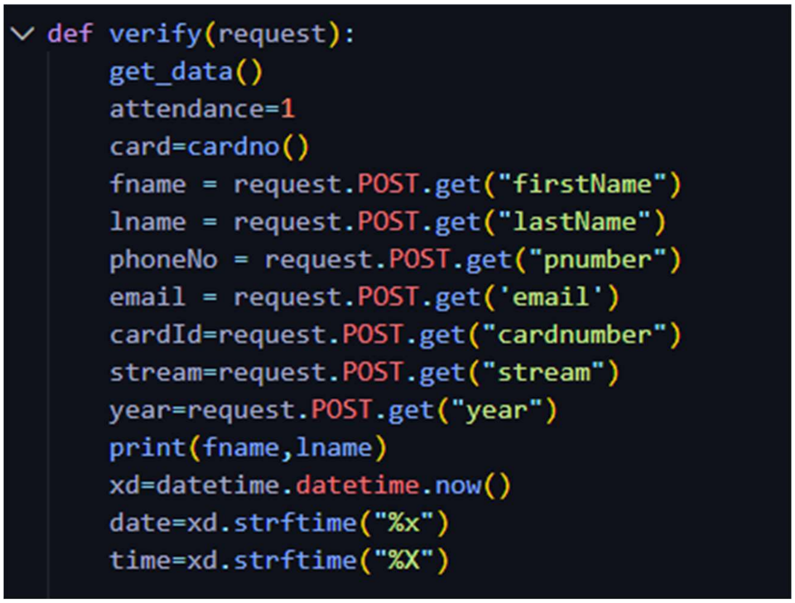
```
def insert(fname,lname,phoneNo,email,cardId,stream,year,attendance,date,time):

    data = {
                'firstName':fname,
                'lastName':lname,
                'pnumber':phoneNo,
                'email':email,
                'cardnumber':cardId,
                'stream':stream,
                'year':year,
                'attendance':attendance,
                'date':date,
                'time':time
            }
    api = 'https://smart-attendance-56a8d-default-rtdb.firebaseio.com//bio.json'

    #response = rq.post(api,json=data).text
    response = rq.post(api,json=data).text
```

This function is inserting user details in the firebase bio bucket.



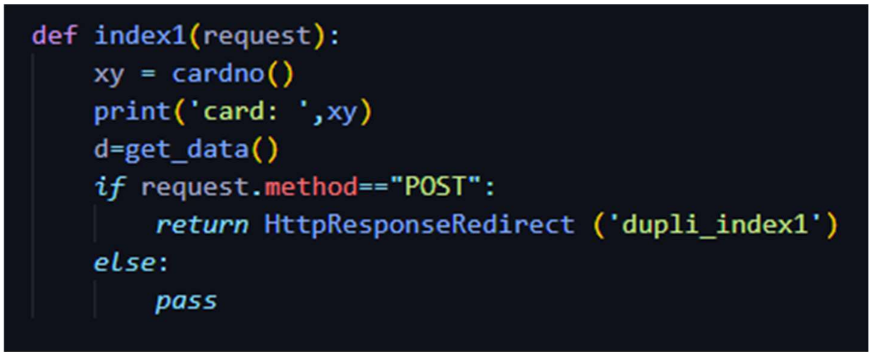
```
def get_data():
    url='https://smart-attendance-56a8d-default-rtdb.firebaseio.com//bio.json'
    x = rq.get(url).text
    data=json.loads(x)
    print(type(data))
    #print(len(data))
    #print(type(data),data)
    d={}
    i=0
    if data is None:
        return 0
    else:
        for x, obj in data.items():
            d[i]=obj
            i+=1
        print(d[0].get("email"))
    return d
```

We are using this function to get all the details of the user from the firebase bucket.



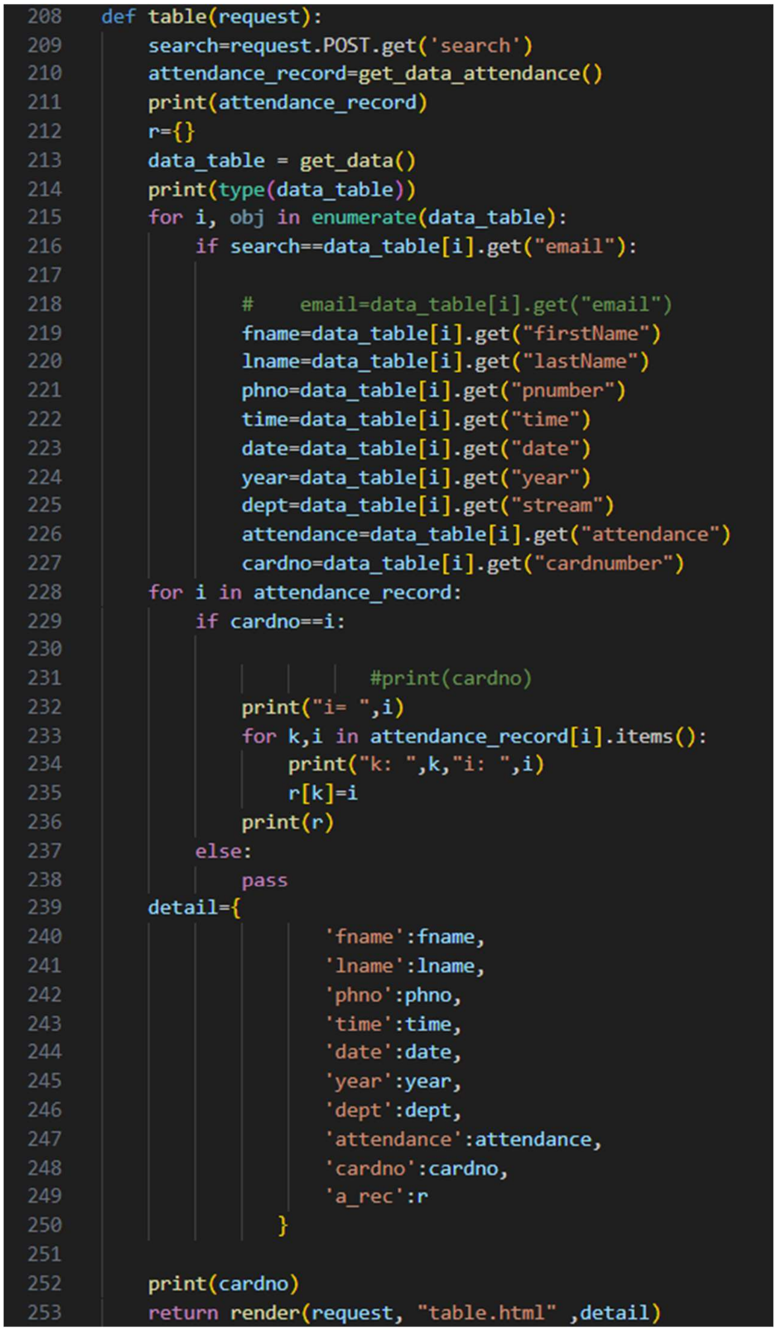
```python
def verify(request):
    get_data()
    attendance=1
    card=cardno()
    fname = request.POST.get("firstName")
    lname = request.POST.get("lastName")
    phoneNo = request.POST.get("pnumber")
    email = request.POST.get('email')
    cardId=request.POST.get("cardnumber")
    stream=request.POST.get("stream")
    year=request.POST.get("year")
    print(fname,lname)
    xd=datetime.datetime.now()
    date=xd.strftime("%x")
    time=xd.strftime("%X")
```

This function is used to verify user. Basically, when user signs up it shows that the sign up is successful and the user’s attendance is counted. If the user has not signed up yet it redirects so that the person needs to sign up first then the attendance will be counted.



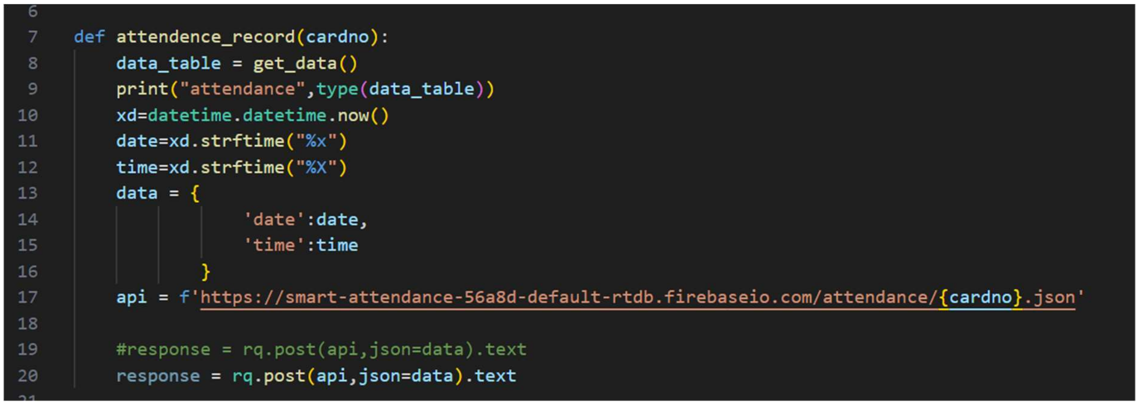
```python
def index1(request):
    xy = cardno()
    print('card: ',xy)
    d=get_data()
    if request.method=="POST":
        return HttpResponseRedirect ('dupli_index1')
    else:
        pass
```

This function is rendering the first page as well.



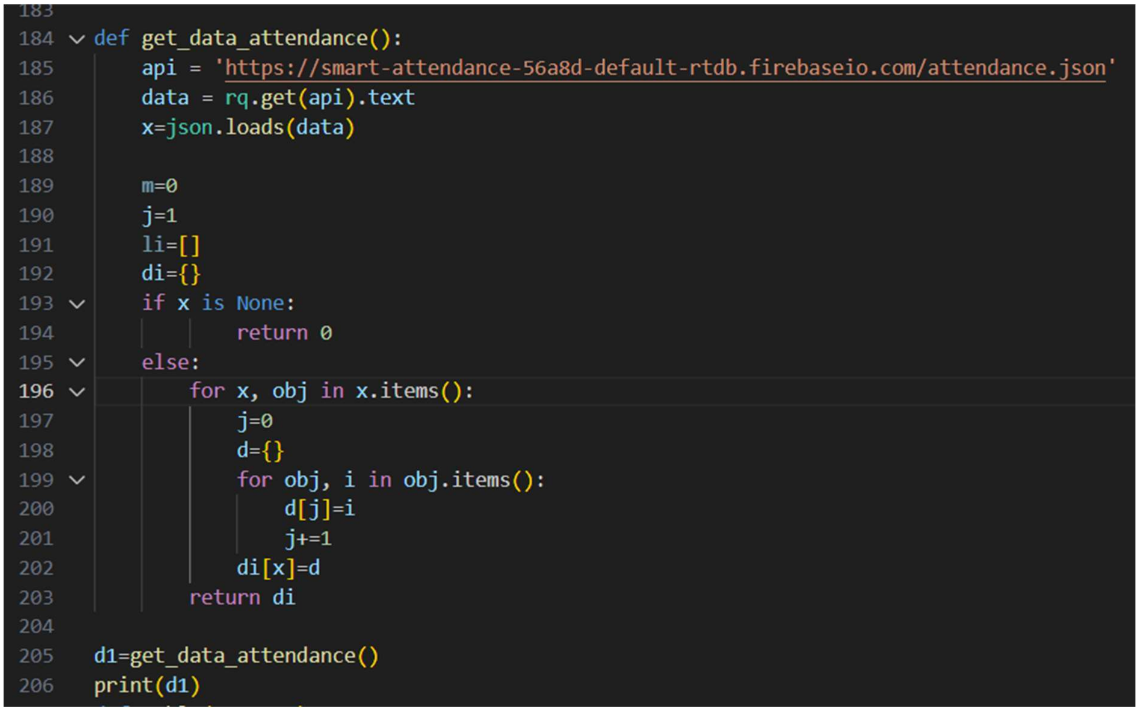
```python
def table(request):
    search=request.POST.get('search')
    attendance_record=get_data_attendance()
    print(attendance_record)
    r={}
    data_table = get_data()
    print(type(data_table))
    for i, obj in enumerate(data_table):
        if search==data_table[i].get("email"):

            #    email=data_table[i].get("email")
            fname=data_table[i].get("firstName")
            lname=data_table[i].get("lastName")
            phno=data_table[i].get("pnumber")
            time=data_table[i].get("time")
            date=data_table[i].get("date")
            year=data_table[i].get("year")
            dept=data_table[i].get("stream")
            attendance=data_table[i].get("attendance")
            cardno=data_table[i].get("cardnumber")
    for i in attendance_record:
        if cardno==i:

                       #print(cardno)
            print("i= ",i)
            for k,i in attendance_record[i].items():
                print("k: ",k,"i: ",i)
                r[k]=i
            print(r)
        else:
            pass
    detail={
                   'fname':fname,
                   'lname':lname,
                   'phno':phno,
                   'time':time,
                   'date':date,
                   'year':year,
                   'dept':dept,
                   'attendance':attendance,
                   'cardno':cardno,
                   'a_rec':r
               }

    print(cardno)
    return render(request, "table.html" ,detail)
```

If user searches for details this function renders the student data page where the student's information is stored and the counted attendance of that very user as well.



```
def attendence_record(cardno):
    data_table = get_data()
    print("attendance",type(data_table))
    xd=datetime.datetime.now()
    date=xd.strftime("%x")
    time=xd.strftime("%X")
    data = {
                'date':date,
                'time':time
            }
    api = f'https://smart-attendance-56a8d-default-rtdb.firebaseio.com/attendance/{cardno}.json'

    #response = rq.post(api,json=data).text
    response = rq.post(api,json=data).text
```

This function is storing user data about attendance into the sub bucket of “Attendance” named “Card Number”.

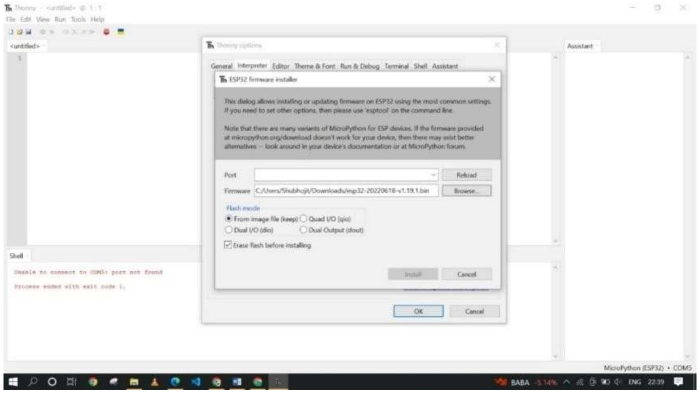


```
def get_data_attendance():
    api = 'https://smart-attendance-56a8d-default-rtdb.firebaseio.com/attendance.json'
    data = rq.get(api).text
    x=json.loads(data)

    m=0
    j=1
    li=[]
    di={}
    if x is None:
        return 0
    else:
        for x, obj in x.items():
            j=0
            d={}
            for obj, i in obj.items():
                d[j]=i
                j+=1
            di[x]=d
        return di

d1=get_data_attendance()
print(d1)
```

When we are searching for information of a particular user through his/her email id, this function is fetching the attendance record from the sub bucket named “card number”.

MICROPYTHON INSTALLATION SNAPS

Step 1


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
Thonny options
General Interpreter Editor Theme & Font Run & Debug Terminal Shell Assistant
ESP32 firmware installer
Reload
Browse...
Install
Cancel
OK
Cancel
Assistant

This dialog allows installing or updating firmware on ESP32 using the most common settings. If you need to set other options, then please use 'esptool' on the command line.

Note that there are many variants of MicroPython for ESP devices. If the firmware provided at micropython.org/download doesn't work for your device, then there may exist better alternatives -- look around in your device's documentation or at MicroPython forum.

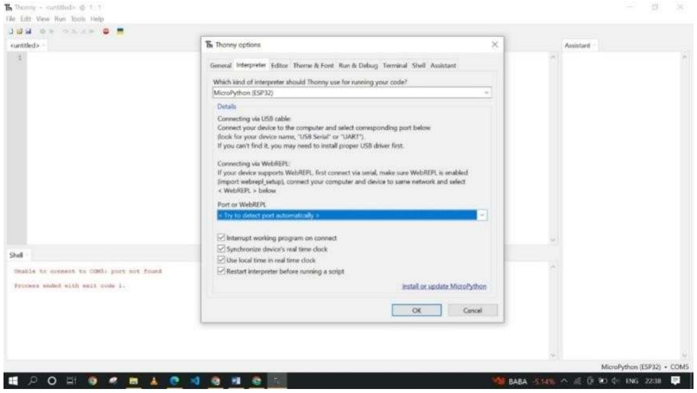
Port
Firmware C:/Users/Shubhojit/Downloads/esp32-20220618-v1.19.1.bin

Flash mode
● From image file (keep) ○ Quad I/O (qio)
○ Dual I/O (dio) ○ Dual Output (dout)
☑ Erase flash before installing

```
Unable to connect to COM5: port not found

Process ended with exit code 1.
```

MicroPython (ESP32) • COM5

Step 2


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
Thonny options
General Interpreter Editor Theme & Font Run & Debug Terminal Shell Assistant
OK
Cancel
Assistant

Which kind of interpreter should Thonny use for running your code?
MicroPython (ESP32)

Details

Connecting via USB cable:
Connect your device to the computer and select corresponding port below
(look for your device name, "USB Serial" or "UART").
If you can't find it, you may need to install proper USB driver first.

Connecting via WebREPL:
If your device supports WebREPL, first connect via serial, make sure WebREPL is enabled
(import webrepl_setup), connect your computer and device to same network and select
< WebREPL > below

Port or WebREPL
< Try to detect port automatically >

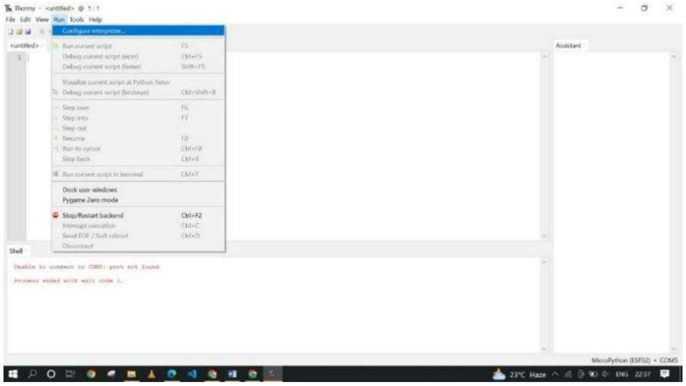
☑ Interrupt working program on connect
☑ Synchronize device's real time clock
☑ Use local time in real time clock
☑ Restart interpreter before running a script

Install or update MicroPython

```
Unable to connect to COM5: port not found

Process ended with exit code 1.
```

MicroPython (ESP32) • COM5

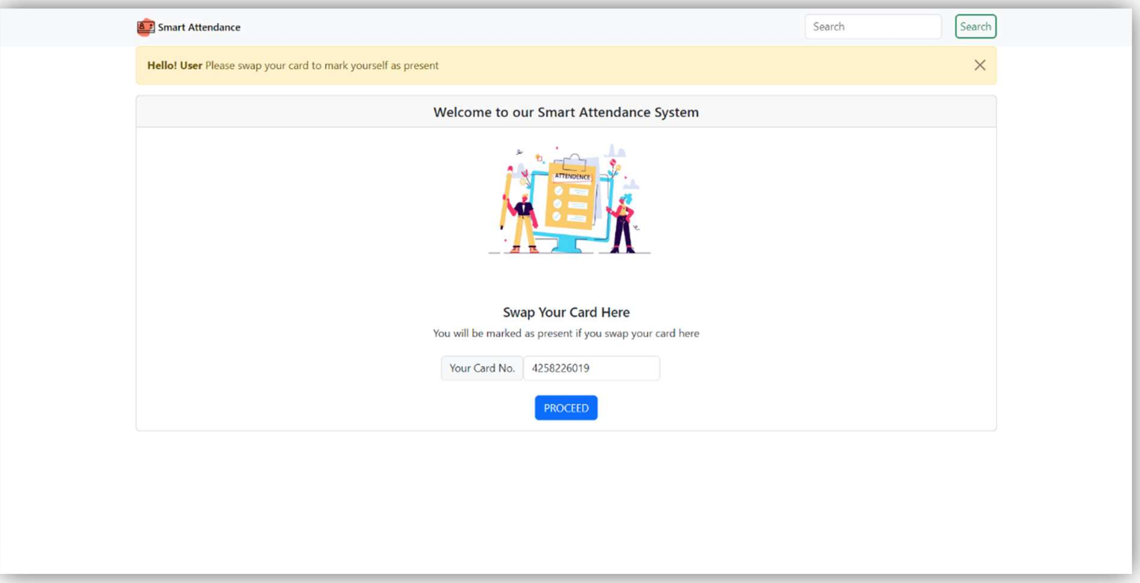
Step 3


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
Configure interpreter...
Run current script F5
Debug current script (nicer) Ctrl+F5
Debug current script (faster) Shift+F5
Visualize current script at Python Tutor
Debug current script (Birdseye) Ctrl+Shift+B
Step over F6
Step into F7
Step out
Resume F8
Run to cursor Ctrl+F8
Step back Ctrl+B
Run current script in terminal Ctrl+T
Dock user windows
Pygame Zero mode
Stop/Restart backend Ctrl+F2
Interrupt execution Ctrl+C
Send EOF / Soft reboot Ctrl+D
Disconnect
Assistant
Shell

```
Unable to connect to COM5: port not found

Process ended with exit code 1.
```

MicroPython (ESP32) • COM5

INTERFACE SNAPSHOTS



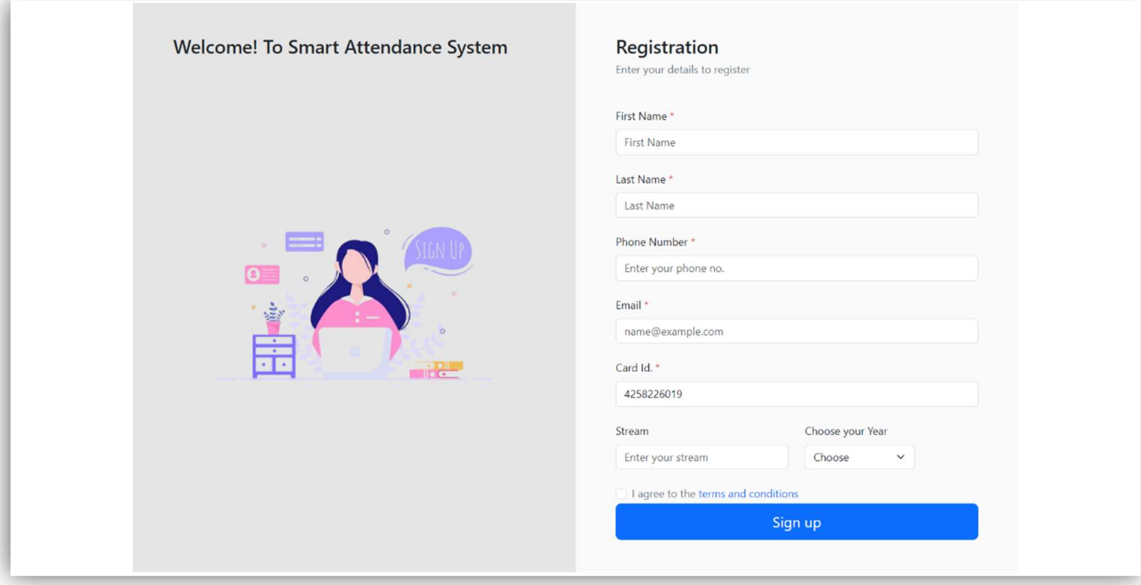
Hello! User Please swap your card to mark yourself as present

Welcome to our Smart Attendance System

Swap Your Card Here

You will be marked as present if you swap your card here

Your Card No. 4258226019



Welcome! To Smart Attendance System

Registration

Enter your details to register

First Name * First Name

Last Name * Last Name

Phone Number * Enter your phone no.

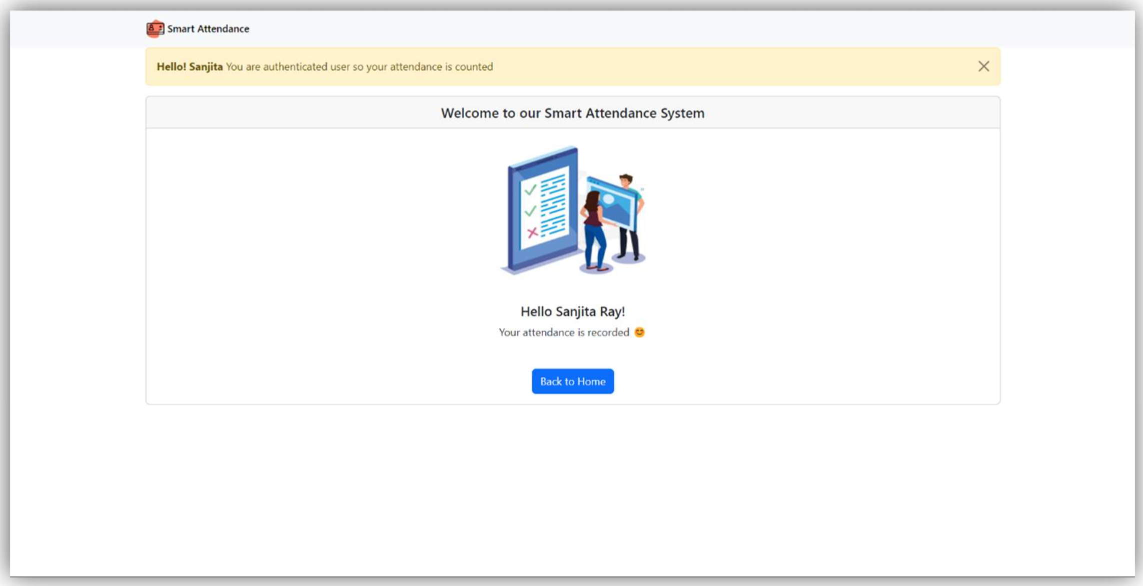
Email * name@example.com

Card Id. * 4258226019

Stream Enter your stream

Choose your Year Choose

□ I agree to the terms and conditions



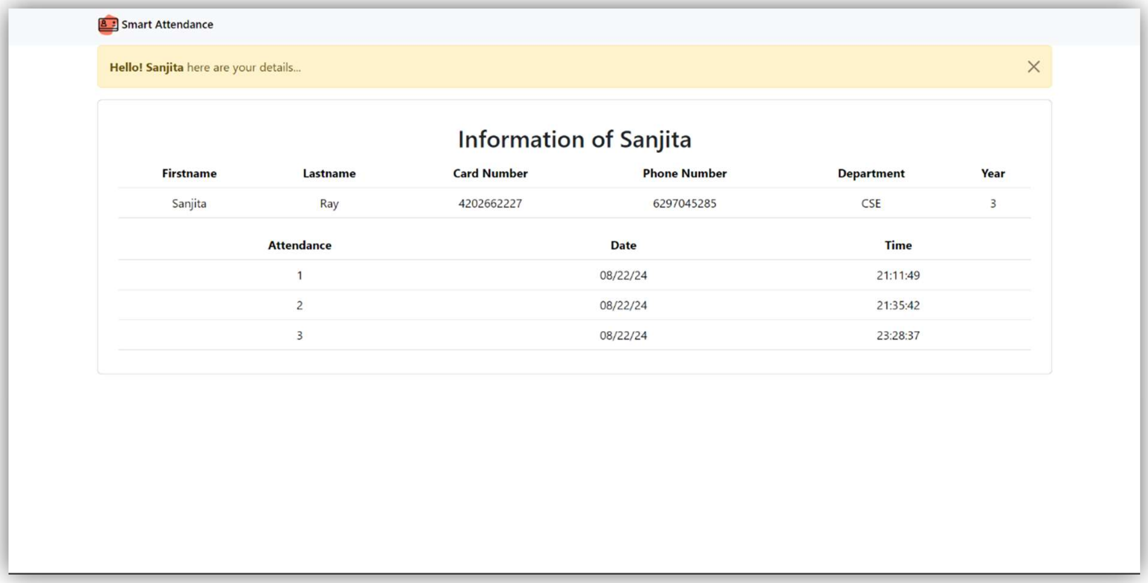
Smart Attendance

Hello! Sanjita You are authenticated user so your attendance is counted ×

Welcome to our Smart Attendance System

Hello Sanjita Ray!

Your attendance is recorded 😊



Smart Attendance

Hello! Sanjita here are your details... ×

Information of Sanjita

Firstname	Lastname	Card Number	Phone Number	Department	Year
Sanjita	Ray	4202662227	6297045285	CSE	3

Attendance	Date	Time
1	08/22/24	21:11:49
2	08/22/24	21:35:42
3	08/22/24	23:28:37

OTHER APPLICATIONS OF IOT

The Internet of Things (IoT) is revolutionizing various industries by connecting devices and systems to the internet, enabling them to communicate, collect, and exchange data. Beyond the commonly known applications like smart homes and wearable devices, IoT is being utilized in a wide array of other areas, each with unique benefits and implications for efficiency, safety, and convenience.

1. Healthcare and Medical Devices:
In the healthcare sector, IoT is making significant strides through the development of smart medical devices and remote patient monitoring systems. IoT-enabled devices, such as wearable health monitors, can track vital signs like heart rate, blood pressure, and glucose levels in real-time. This data is then transmitted to healthcare providers, allowing for timely interventions and continuous monitoring of patients, particularly those with chronic conditions. Additionally, IoT is being used to manage and maintain medical equipment, ensuring they are always operational and reducing downtime through predictive maintenance.

2. Smart Cities:
IoT is playing a crucial role in the development of smart cities, where connected devices and sensors are used to manage urban infrastructure more efficiently. Applications include smart traffic management systems that optimize traffic flow and reduce congestion, smart street lighting that adjusts brightness based on activity levels, and waste management systems that optimize collection routes based on real-time data from smart bins. These technologies not only enhance the quality of life for residents but also contribute to sustainability by reducing energy consumption and environmental impact.

3. Agriculture and Farming:
IoT is transforming agriculture through precision farming techniques that use connected devices to monitor and manage crops, soil, and livestock. Sensors placed in fields can track soil moisture, temperature, and nutrient levels,

enabling farmers to optimize irrigation and fertilization processes. IoT also facilitates automated farming equipment, such as tractors and drones, that can be controlled remotely or operate autonomously, improving efficiency and reducing labor costs. Livestock monitoring systems, which track the health and location of animals, also contribute to better herd management and increased productivity.

4. Industrial IoT (IIoT):

In industrial settings, IoT, often referred to as Industrial IoT (IIoT), is being used to optimize manufacturing processes, improve supply chain management, and enhance worker safety. Connected sensors and devices on production lines can monitor machinery in real-time, predicting failures before they occur and scheduling maintenance proactively. This reduces downtime and extends the lifespan of equipment. Additionally, IoT is used in inventory management systems to track goods throughout the supply chain, ensuring timely deliveries and reducing losses.

5. Retail and Supply Chain:

In the retail sector, IoT is enhancing the shopping experience through smart shelves that track inventory levels and automated checkout systems that use sensors to detect items in a shopper’s cart, enabling seamless transactions. IoT is also improving supply chain management by providing real-time tracking of goods from warehouses to retail outlets. This ensures better inventory management, reduces losses due to spoilage or theft, and enhances transparency throughout the supply chain.

6. Environmental Monitoring:

IoT is increasingly being used for environmental monitoring, where connected sensors are deployed to track air and water quality, monitor weather patterns, and detect pollution levels. These systems can provide real-time data that helps governments, organizations, and researchers respond more effectively to environmental challenges. For example, IoT sensors can detect changes in water quality in rivers and lakes, providing early warnings of contamination and helping to protect ecosystems.

7. Smart Grids and Energy Management:
IoT is integral to the development of smart grids, which use connected devices to manage electricity distribution more efficiently. Smart meters in homes and businesses provide real-time data on energy consumption, helping utilities balance supply and demand, reduce outages, and improve energy efficiency. IoT is also being used in renewable energy systems, where sensors track the performance of solar panels and wind turbines, optimizing energy production and maintenance.

CONCLUSION

In conclusion, the Smart Attendance System represents a significant advancement in the management of attendance within educational institutions and workplaces. By integrating facial recognition technology with real-time data processing and secure cloud storage, this system not only enhances the accuracy and efficiency of attendance tracking but also reduces the potential for human error and fraud. Its contactless operation and user-friendly interface make it particularly relevant in today's world, where hygiene and convenience are paramount. The Smart Attendance System is a valuable tool that streamlines administrative processes, supports data-driven decision-making, and ultimately contributes to a more productive and secure environment.

REFERENCES

- https://iotdesignpro.com/projects/iot-based-smart-attendance-system-project-using-nodemcu

- https://www.mdpi.com/2079-9292/11/19/3151

- https://github.com/Sanskriti1110/IoT-Based-Smart-Attendance-System