Configuration Manual

MSc Research Project
Cloud Computing

Nithish Narayana Dasa Aswartha
Student ID: x21141762

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Nithish Narayana Dasa Aswartha
Student ID:	x21141762
Programme:	Cloud Computing
Year:	2022
Module:	MSc Research Project
Lecturer:	Rashid Mijumbi
Submission Due Date:	15-12-2022
Project Title:	Configuration Manual

Word Count: 1002 **Page Count:** 14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Nithish Narayana Dasa Aswartha

Date: 15-12-2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including	□

multiple copies)	
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Nithish Narayana Dasa Aswartha
X21141762

The motivation of the document:

This paper discusses the services that are needed to answer the research question. I am using python 3, flask, and IoT(internet of things) to interface face recognition and fingerprint and to authorize users to log into the page.

Project Title: Two-layer security authentication system for a cloud application in order to decrease cyber attacks.

1 Configuration required for development

1.1 Code Repository

The source code of the zip file is attached.

1.2 Required Run Time Programming Languages

The following packages need to be imported in order to carry out the research:

- Pip3 install opencv-python==4.5.1.48
- Pip3 install imutils==0.5.4
- Pip3 install flask==2.2.2
- Pip3 install requests==2.28.1
- Pip3 install pillow==9.3.0

After installing the above dependencies configure your mobile device with the below steps.

1. Change the mobile hotspot name to iot
2. And enter the password as 12345678 (Attached: Fig 2).
3. Connect the Fingerprint sensor which is attached to the microcontroller to the laptop using a Micro USB cable.

4. Check your mobile device in the wifi hotspot screen, It should be successfully connected to the IoT device(Fingerprint sensor attached with a microcontroller). (Attached: Fig 1).
5. Once it is successfully connected, the Fingerprint sensor should turn the white light ON.

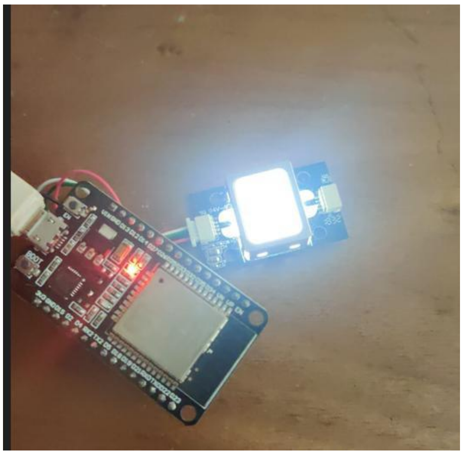
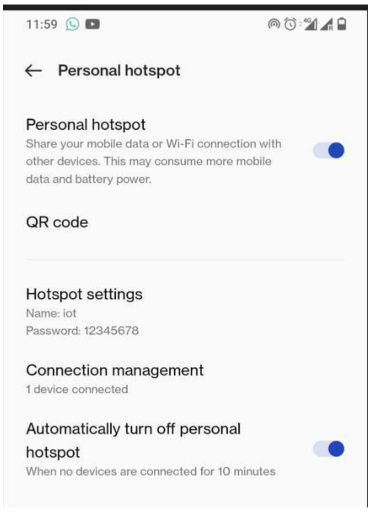


Fig 1: Fingerprint sensor
With microcontroller.



11:59

Personal hotspot

Personal hotspot
Share your mobile data or Wi-Fi connection with other devices. This may consume more mobile data and battery power.

QR code

Hotspot settings
Name: iot
Password: 12345678

Connection management
1 device connected

Automatically turn off personal hotspot
When no devices are connected for 10 minutes

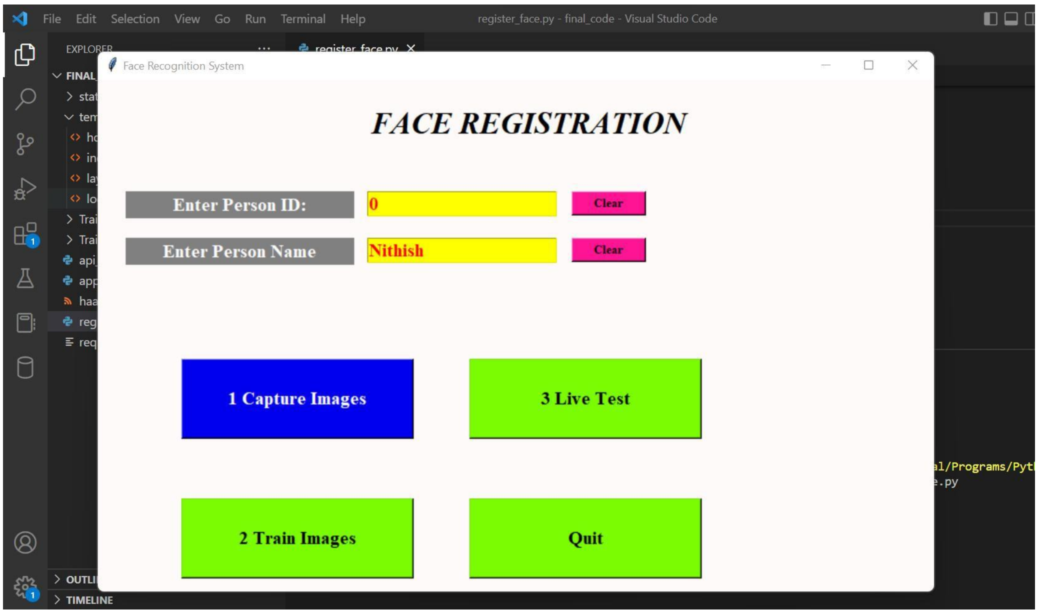
Fig 2: Hotspot connection
to IoT device.

1.3 **Registering face**

Once the above pre-requisite is installed properly. Open the attached source code in Visual Studio for executing the python file.

Click on register_face.py

The face registration screen will be displayed as shown below.



FACE REGISTRATION

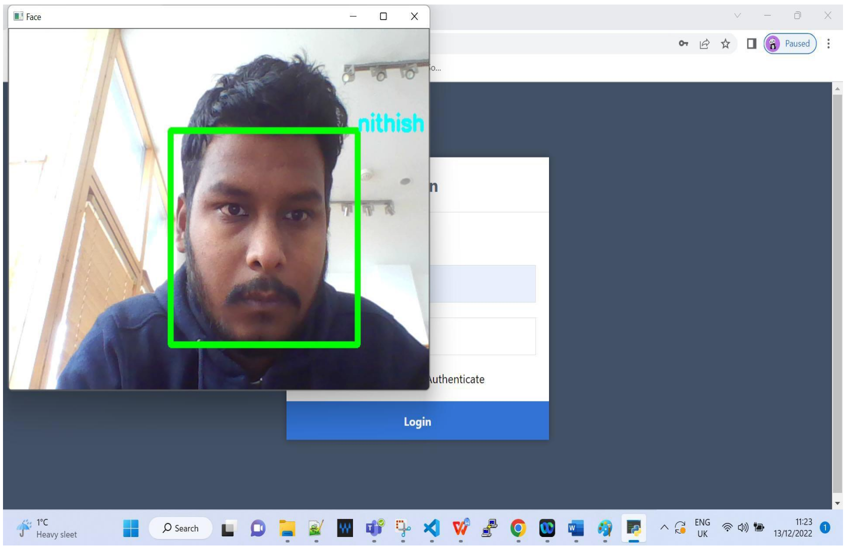
Enter Person ID: 0

Enter Person Name Nithish

Fig 1: Entering user id and name

Once after Capture Images, click on Train Images and Live Test one after another. So, the registration of face recognition will be successfully completed.

Fig 2 Verifying face recognition

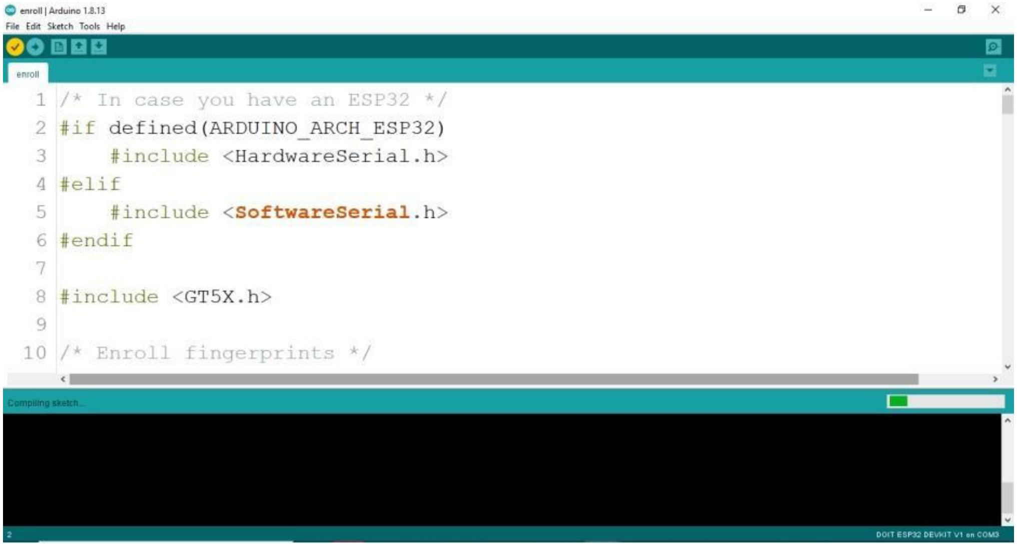


Once face registration is done the system will now be able to recognize the face successfully at the time of the login. As shown in the above screenshot Fig 2.

The Second step is to register the Fingerprint for that we are using Arduino

2. Fingerprint sensor interface using Arduino

The Arduino IDE, as the software is more often known, features an editor for composing programs, a chat window, a text terminal, a toolbar with frequently used buttons, and a selection of menus. As a result, it is possible to upload programs to the Esp32 equipment and have communications with them.

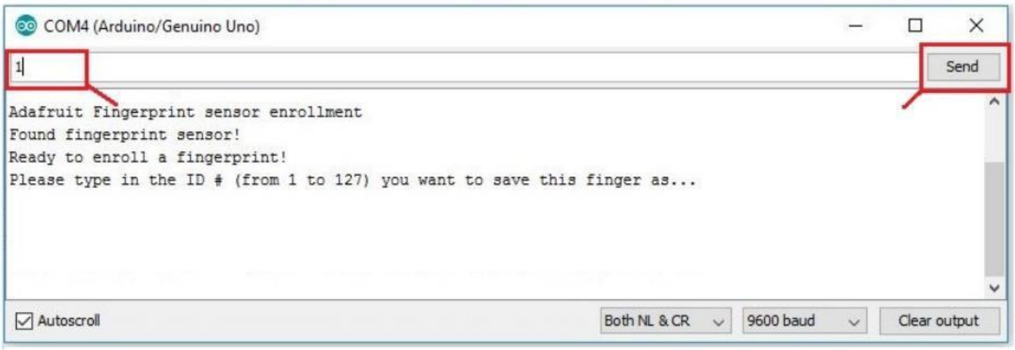


```
/* In case you have an ESP32 */
#if defined(ARDUINO_ARCH_ESP32)
    #include <HardwareSerial.h>
#elif
    #include <SoftwareSerial.h>
#endif

#include <GT5X.h>

/* Enroll fingerprints */
```

Fig 3: Uploading user enroll program to esp32 microcontroller



1

```
Adafruit Fingerprint sensor enrollment
Found fingerprint sensor!
Ready to enroll a fingerprint!
Please type in the ID # (from 1 to 127) you want to save this finger as...
```

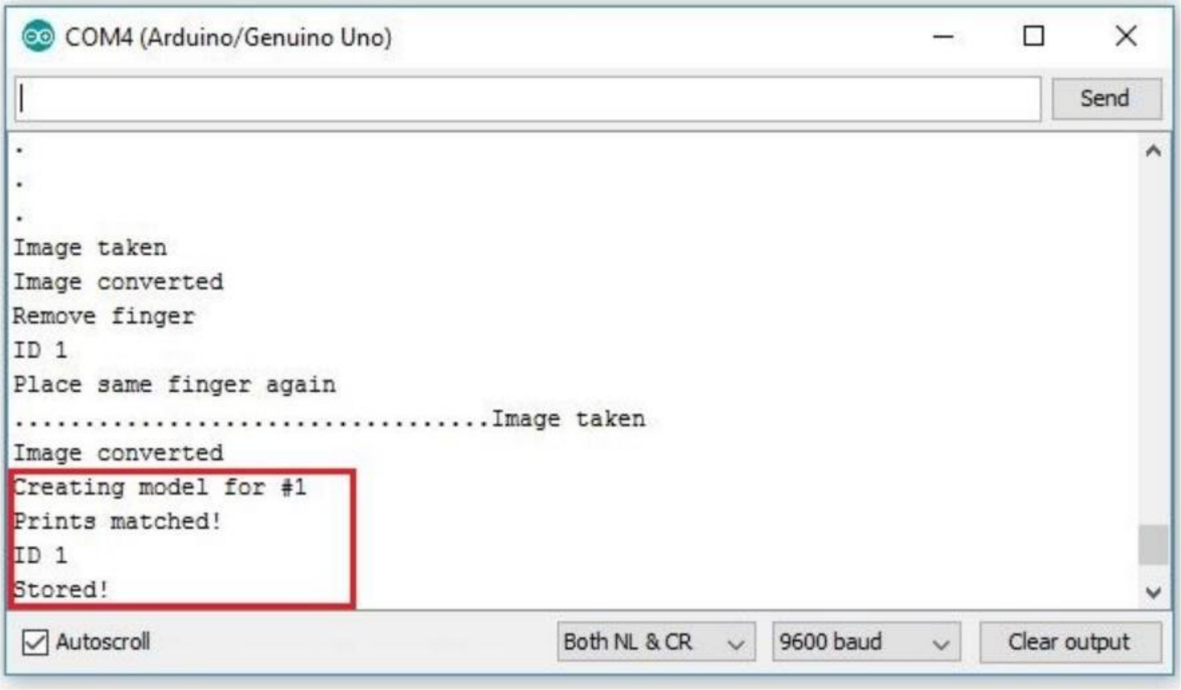
☑ Autoscroll Both NL & CR 9600 baud

2.1 New Fingerprint Enrollment:

Once you have connected the fingerprint reader module to the Arduino, you can add a new fingerprint. The Fingerprint Reader library must have been previously installed.
Step 1: Select File > Examples > Enroll in the Arduino IDE.
Simply open the serial communication at 9600 baud and upload the code.

3. Enter your fingerprint ID. If this is your first time capturing a fingerprint, Enter 1 in the top-left corner before clicking the send button.

4. Put your finger on the scanner, then adhere to the serial monitor's instructions.



```
.
.
.
Image taken
Image converted
Remove finger
ID 1
Place same finger again
..................................Image taken
Image converted
Creating model for #1
Prints matched!
ID 1
Stored!
```

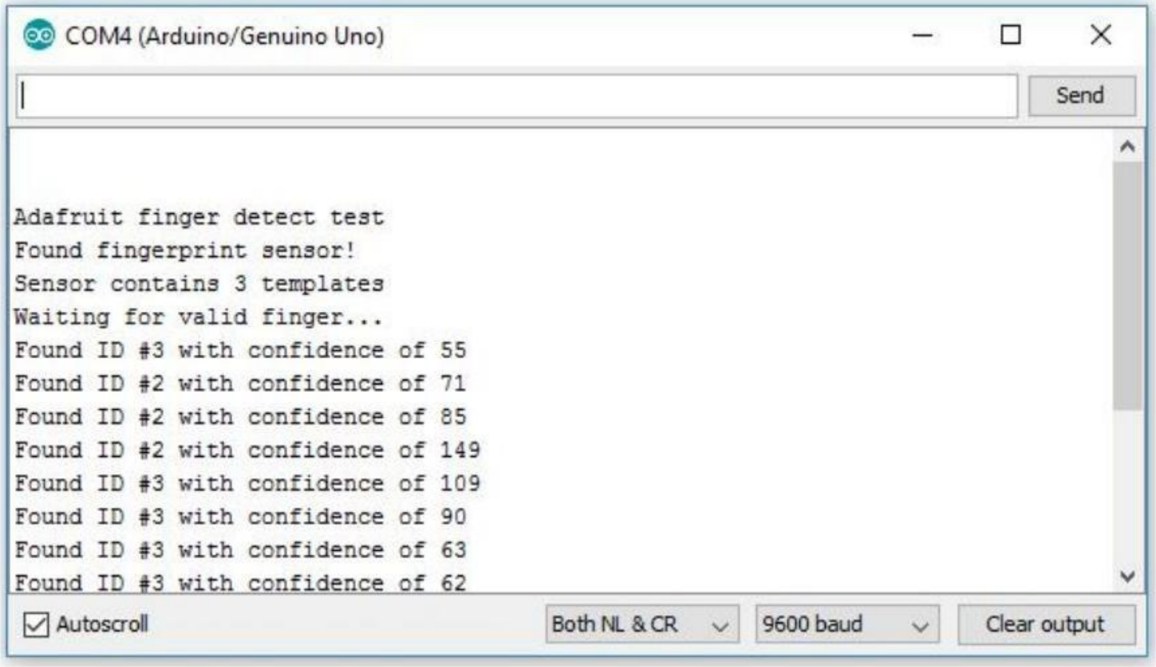
Fig 4: Entering user id and enrolling user

The scanner will require you to lay the same finger on it twice. The "Prints matched!" notification displayed below indicates that your fingerprint was successfully registered. If that doesn't work, try it again to store the fingerprints.

2.2 Finding a Match:

You should now have more than one fingerprint on different IDs. Follow these steps to locate a match with the fingerprint reader.

1. Navigate to File > Examples > Main.
In the Arduino IDE and then upload the code to your Arduino board.
2. Open the Serial Monitor and set it to 9600 bauds per second. You should be seeing the message below.
3. Put the finger you want to identify on the scan.
4. On the serial monitor, the ID that corresponds to the fingerprint is displayed. It also indicates confidence; the greater the confidence, the greater the similarity of the fingerprint and the stored fingerprint.



```
Adafruit finger detect test
Found fingerprint sensor!
Sensor contains 3 templates
Waiting for valid finger...
Found ID #3 with confidence of 55
Found ID #2 with confidence of 71
Found ID #2 with confidence of 85
Found ID #2 with confidence of 149
Found ID #3 with confidence of 109
Found ID #3 with confidence of 90
Found ID #3 with confidence of 63
Found ID #3 with confidence of 62
```

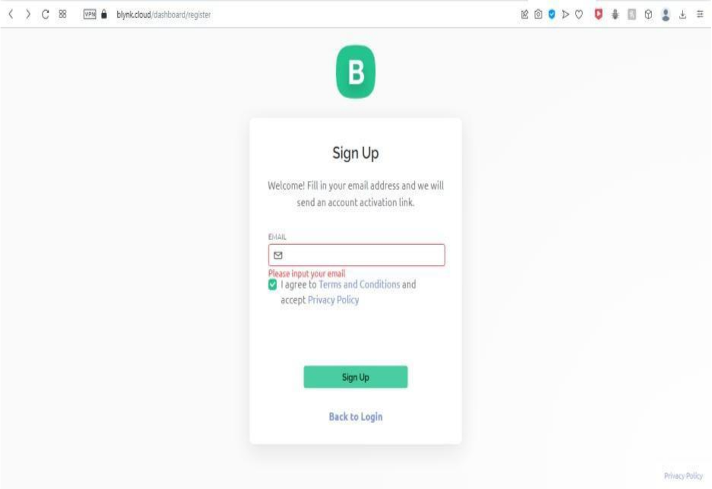
☑ Autoscroll

Both NL & CR

9600 baud

Fig 5: Validating user fingerprint match

2.3 Integrating fingerprint sensor with Blynk



blynk.cloud/dashboard/register

Sign Up

Welcome! Fill in your email address and we will send an account activation link.

EMAIL

Please input your email

☑ I agree to Terms and Conditions and accept Privacy Policy

Blynk is an IoT business that provides a platform for creating mobile (IOS and Android) applications that may link electronic devices to the Internet and allow for remote monitoring and control of these items.
And I'm using blynk in this project to check the fingerprint status or the logs.

Fig 6: Sign up for a new
account

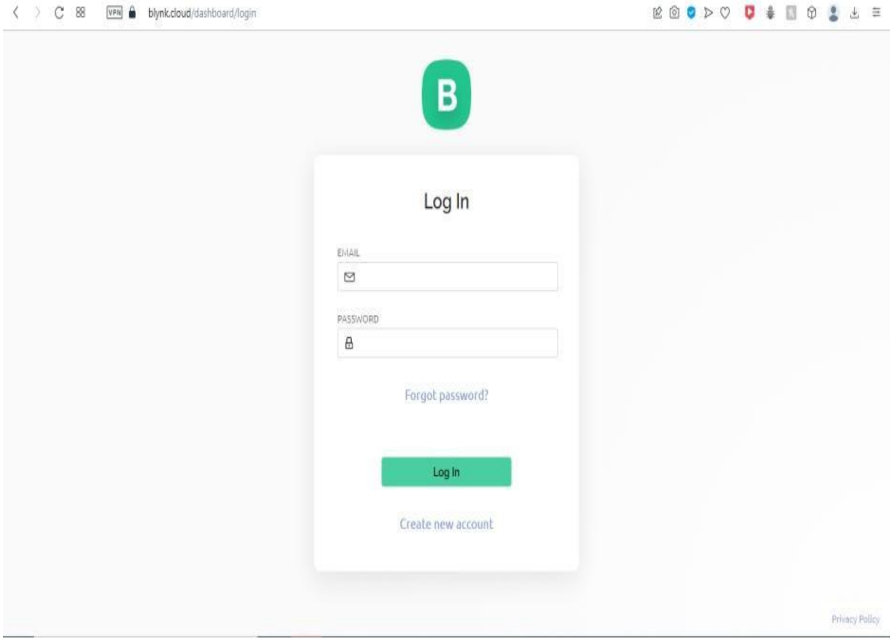
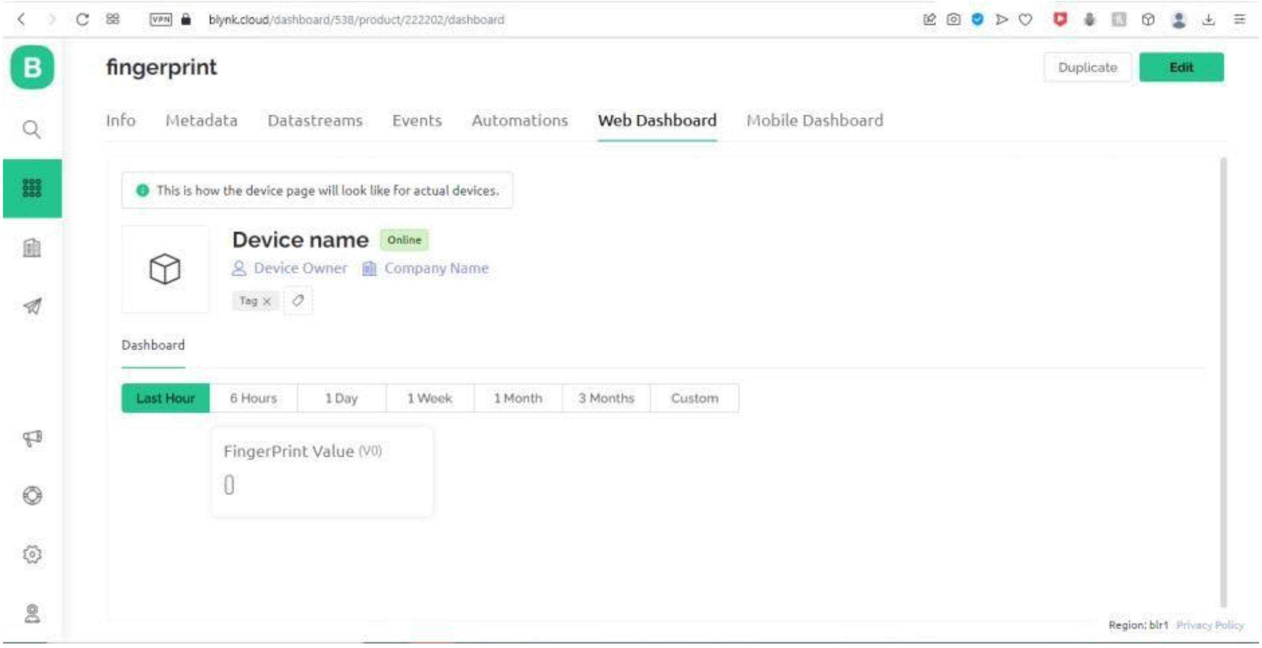


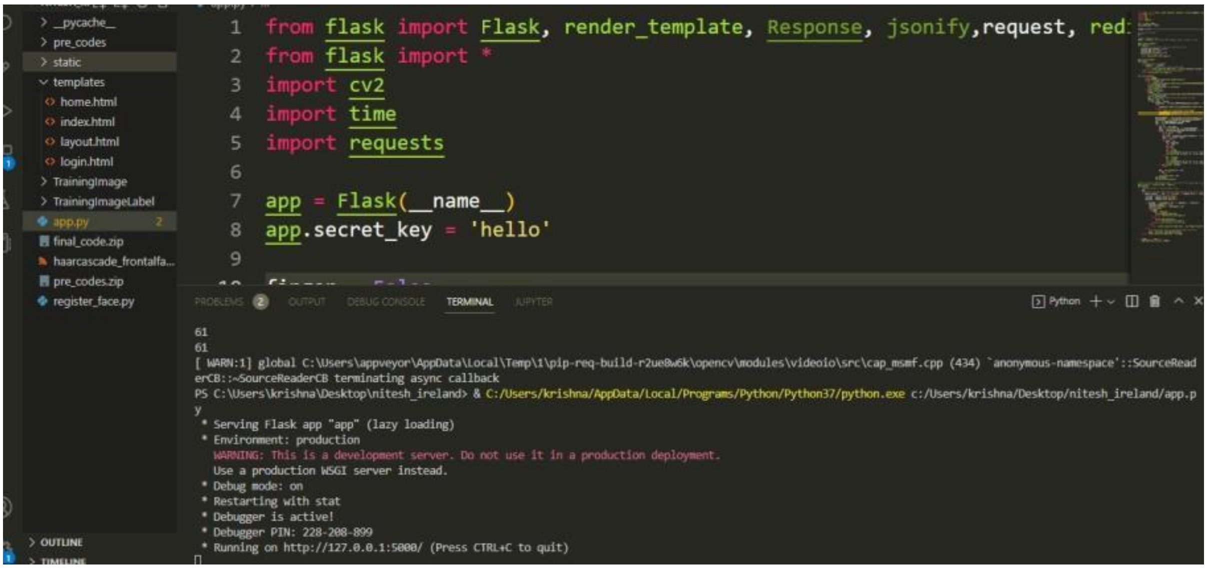
Fig 7: Login with newly registered user



This is how the device page will look like for actual devices.

Fig 8: Creating new dashboard for fingerprint validation

5 Main server and web output



```python
from flask import Flask, render_template, Response, jsonify,request, red
from flask import *
import cv2
import time
import requests

app = Flask(__name__)
app.secret_key = 'hello'

```

```
61
61
[ WARN:1] global C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-r2ue8w6k\opencv\modules\videoio\src\cap_msmf.cpp (434) `anonymous-namespace'::SourceRead
erCB::~SourceReaderCB terminating async callback
PS C:\Users\krishna\Desktop\nitesh_ireland> & C:/Users/krishna/AppData/Local/Programs/Python/Python37/python.exe c:/Users/krishna/Desktop/nitesh_ireland/app.p
y
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 228-208-899
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig 9: Main program and server stats details

After running the app.py, the URL will be displayed in the console tab. As shown in the above picture (Fig 9).

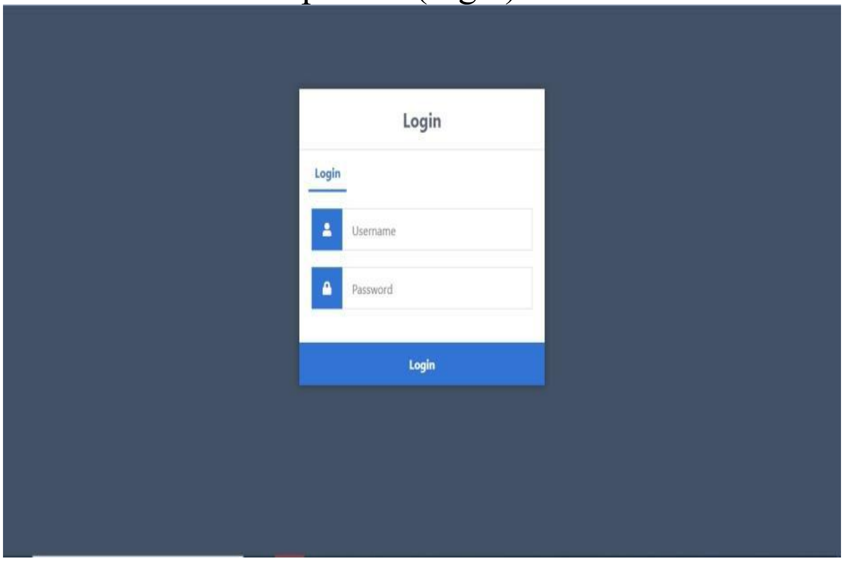
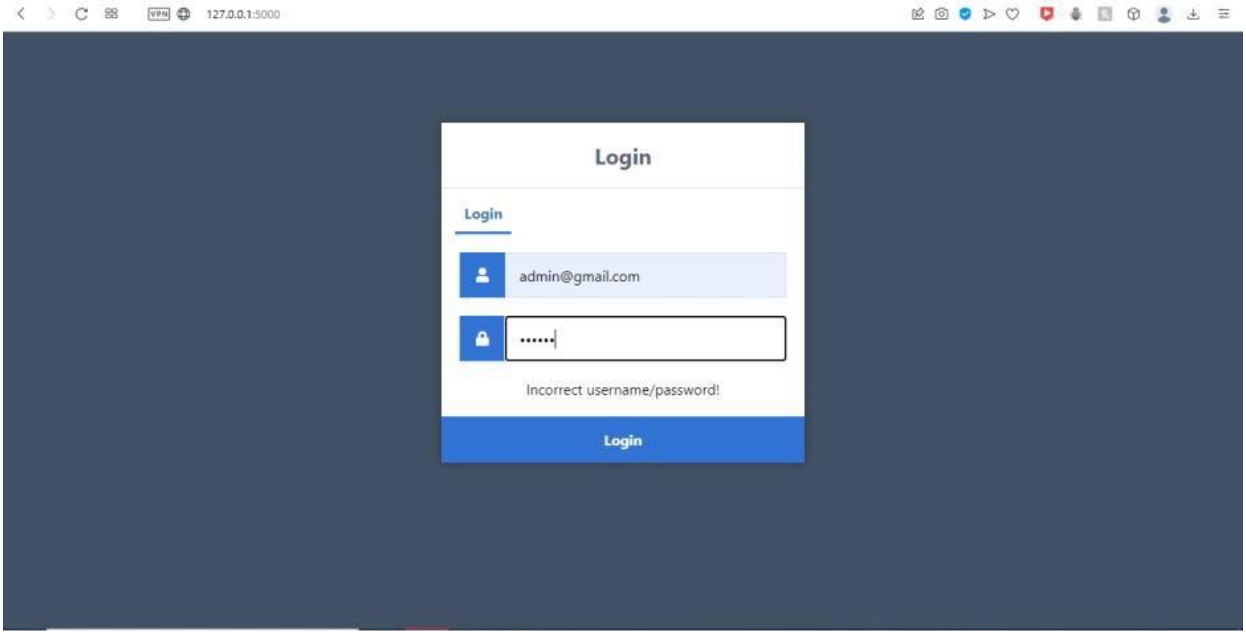


Fig 10 user login prompt

Whenever the URL is open in a new tab, the login screen will be displayed. Credential are:

Username: admin@gmail.com
Password: password


127.0.0.1:5000

Login

Login

admin@gmail.com

••••••

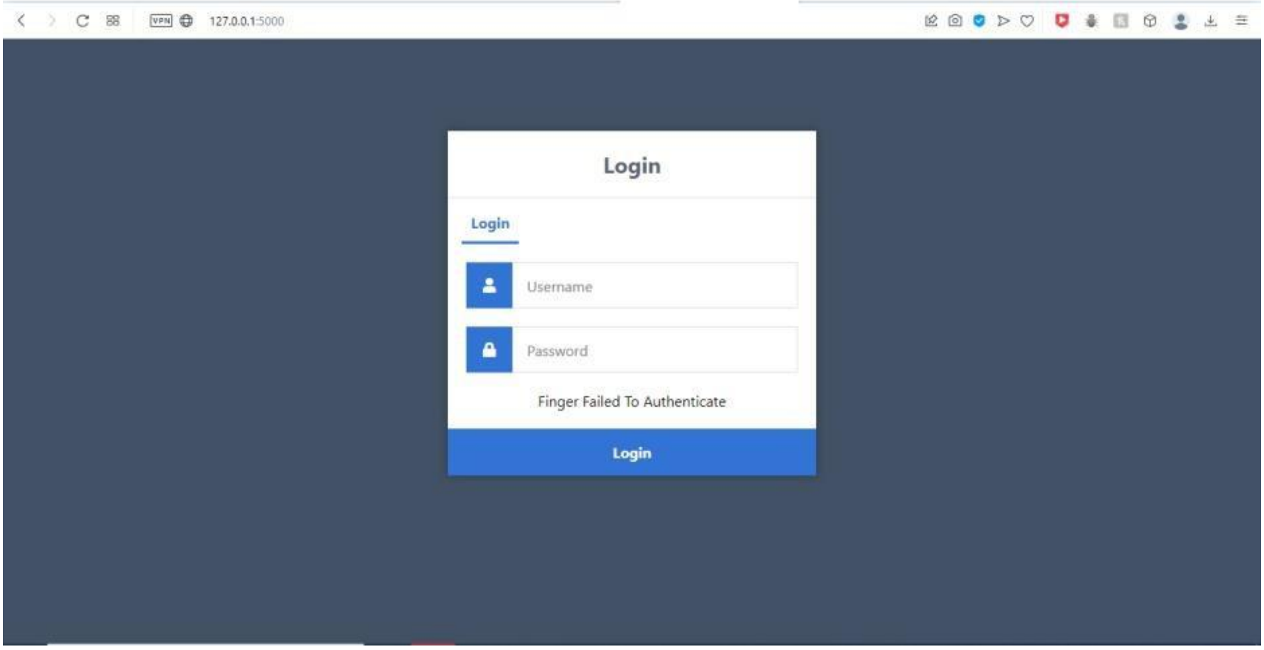
Incorrect username/password!

Login

Fig 11 invalid user or password prompt

After typing the credentials in the textbox, clicking on the login button will display the error message as shown in the below picture Fig 12.

3. Second level of authentication failed case (fingerprint)


127.0.0.1:5000

Login

Login

Username

Password

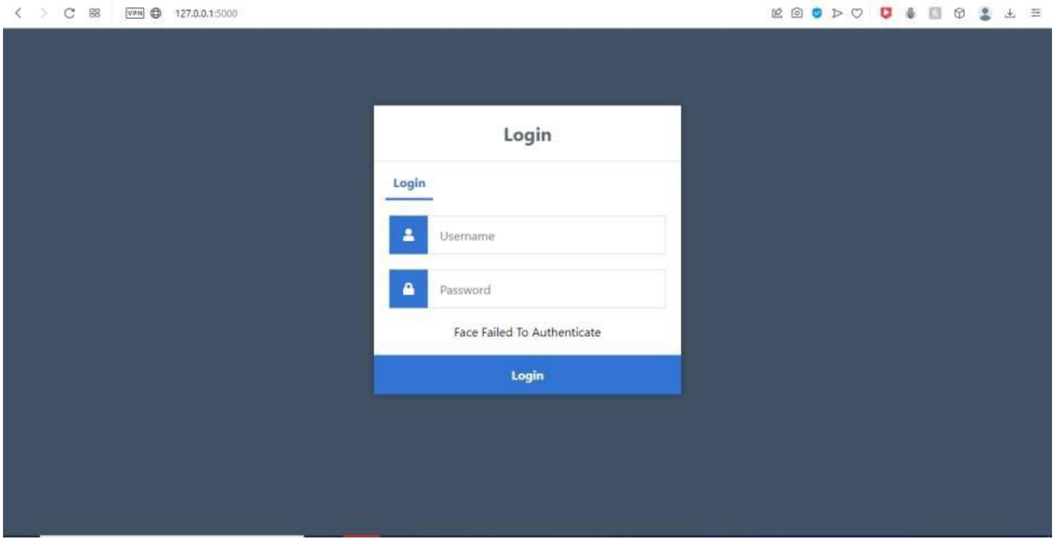
Finger Failed To Authenticate

Login

Fig 12 fingerprint authentication failed

Because Fingerprints should be validated before clicking on the login. Just place the registered finger on the fingerprint sensor then login will be successful. If any not registered finger is placed on the sensor error message will be displayed.

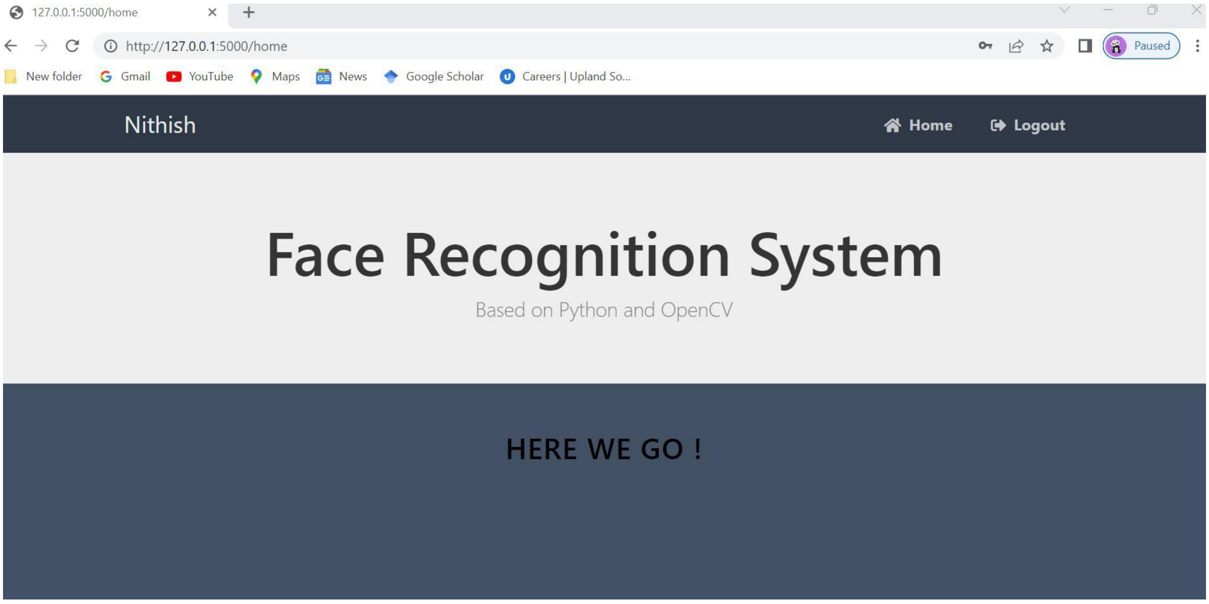
3.1 Face recognition failed case (3rd level)



Face Failed To Authenticate

Fig 13 face recognition failed for the user

Later it checks for face recognition. When the webcam is closed then it will display the error message(Fig 13). After the fingerprint recognition webcam should be active. The system will verify the registered face. Once it is done, the home page will be displayed (Fig 14).



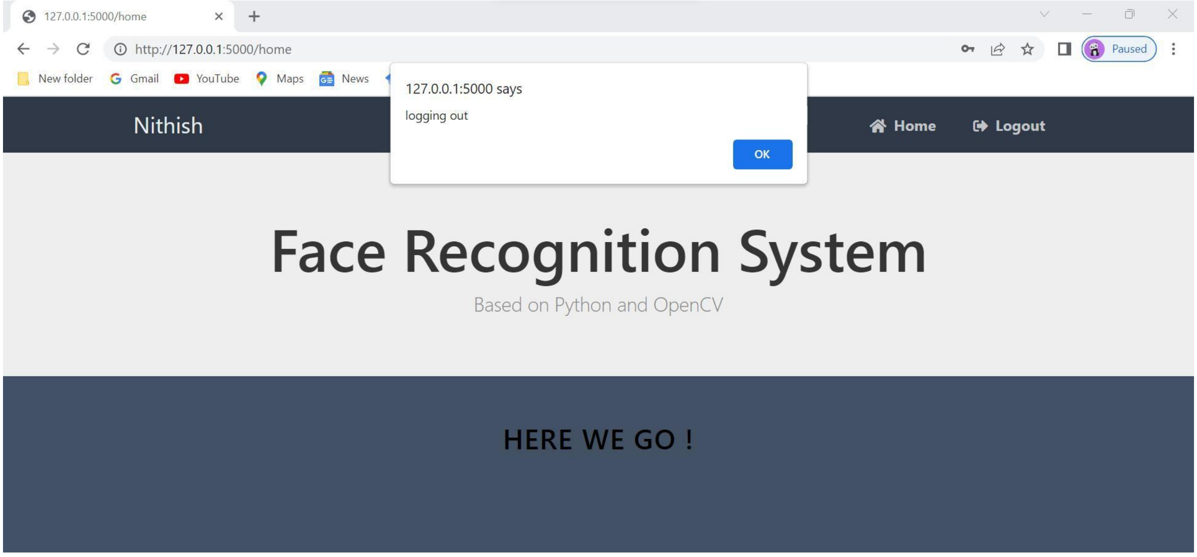
Face Recognition System

Based on Python and OpenCV

HERE WE GO !

Fig 14: After successful login

After successful login, whenever timeout occurred, the web browser will be automatically logged out. (Fig 15)



127.0.0.1:5000 says

logging out

Face Recognition System

Based on Python and OpenCV

HERE WE GO !

Fig 15: Timeout

References

1. P., K. (2020). A sensor based IOT monitoring system for electrical devices using BLYNK framework. *September 2020*, *2*(3), 182–187. https://doi.org/10.36548/jei.2020.3.005
2. Meenakshi, N., Monish, M., Dikshit, K. J., & Bharath, S. (2019). Arduino based Smart Fingerprint Authentication System. *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*. https://doi.org/10.1109/iciict1.2019.8741459