**Smart Facial Recognition Locking System**

**Student Name:** John Joey Wright  
**Student ID:** 20105823

**1. Introduction**

This project implements a smart access system using facial recognition, built around a Raspberry Pi. When a user presses a button, the system checks the ambient light level using a Grove Light Sensor. The current light condition and system status are displayed on a 16x2 LCD screen.

f light conditions are acceptable, a picture is taken using a connected camera (either a Raspberry Pi Camera Module or USB webcam). The image is uploaded to a local web server (running in a container on the student's laptop) which performs facial recognition using DeepFace, an open-source Python facial recognition library. If the face is verified as a known user, the system plays a happy beep and unlocks the door via a relay. If the face is not recognized, a sad beep is played and access is denied.

**2. System Architecture Diagram**

A diagram of a computer system

AI-generated content may be incorrect.

**Figure 1:** System Architecture showing hardware and logic flow between Raspberry Pi, camera, sensors, LCD, and the local web server.

**3. Code Snippets and Explanations**

**Light Sensor Setup:**

light\_sensor = GroveLightSensor(channel=2)

if light\_sensor.light < LIGHT\_THRESHOLD:

lcd\_update\_bottom("Turn on light")

**LCD Update Functions:**

def lcd\_update\_top(line=""):

lcd.setCursor(0, 0)

lcd.write(line[:16].ljust(16))

def lcd\_update\_bottom(line=""):

lcd.setCursor(1, 0)

lcd.write(line[:16].ljust(16))

**Camera Capture with OpenCV:**

cap = cv2.VideoCapture(0)

ret, frame = cap.read()

cv2.imwrite("image1.jpg", frame)

cap.release()

**Sending to Server:**

files = {'image1': open('image1.jpg', 'rb')}

response = requests.post(SERVER\_URL, files=files)

**Decision Handling:**

if verified:

happy\_beep()

unlock\_door()

else:

sad\_beep()

**4. Screenshots**

A screen with a blue screen on it

AI-generated content may be incorrect.

Figure idle

A screen with a blue display

AI-generated content may be incorrect.

Figure taking picture/uploading picture

A digital display on a piece of duct tape

AI-generated content may be incorrect.

Figure User Verified

A hand holding a blue screen

AI-generated content may be incorrect.

Figure Light Level Low

**5. Challenges Faced and Solutions**

**1. Raspberry Pi Camera Failure**

* **Problem:** No data received from sensor error
* **Solution:** Switched to USB webcam and used OpenCV

**2. LCD Display Issues**

* **Problem:** Text overlapping or not clearing
* **Solution:** Created separate update functions for each line

**3. Verification Feedback Delay**

* **Problem:** LCD stuck on user name
* **Solution:** Reset LCD message after a delay using time.sleep(5)

**4.** **Azure Issues:**

* **Problem**: Azure student account has very strict limits and wont allow you to do a lot of things like make a container, use the facial recognition or create a logic app.
* **Solution**: I found no solution so I didn’t use Azure

**5. Facial Recognition Service Selection**

* **Problem:** Azure Cognitive Services were restricted by student subscription limits.
* **Solution:** Switched to using **DeepFace**, a powerful and easy-to-integrate open-source facial recognition library that runs entirely on the local machine.