Smart Doorbell System: Installation and Execution

---

1. Installation Steps

1.1 Environment Setup

1. **Install Python 3.x**
   Download and install Python 3.x from the official website if it's not already installed.
2. **Install Required Libraries**
   Run the following command in the terminal to install the necessary Python libraries:
3. pip install flask RPi.GPIO requests opencv-python-headless
4. install Cmake From official website and OpenCv Libraries and also install dlib libraries by the command

   - *pip install opencv-python*
   - *Pip install flask*
   - *Pip install dlib*
   - *Pip install Face_recognition*

1.2 Hardware Setup

1. **Camera Module**
   - Connect the Camera Module to the USB port on the PC.
   - Enable the camera in the Module.
2. **Doorbell Button**
   - Connect a push button to GPIO pin 18 for input.
   - Use a pull-up or pull-down resistor as required.
3. **Buzzer and LED**
   - Connect a buzzer to GPIO pin 23.
   - Attach an LED to GPIO pin 24 with an appropriate resistor.
4. **Speaker and Microphone (Optional)**
   - Connect a USB or Bluetooth speaker and microphone for two-way audio communication.

1.3 Project File Structure

smart-doorbell-system

├── game.py

├── env setup

│ └── Feed Trained image

├── captured_images of Unknown

│ └── Unknown.jpg # (captured visitor images)

---

2. Execution Steps

2.1 Start the Application

1. Open the terminal and navigate to the project directory:
2. cd /smart-doorbell-system as python game.py
3. Run the Flask application:
4. python game.py

2.2 Access the Web Interface

- Open a web browser and go to the following URL:
- http://localhost:5000

---

3. System Usage

---

1. **Detect Visitors**
    - When the doorbell button is pressed, the system captures a photo of the visitor using the camera module.
    - The captured image is saved in the captured_images folder and displayed on the web interface.
2. **Alerts**
    - The buzzer and LED briefly activate to notify that a visitor is at the door.

---

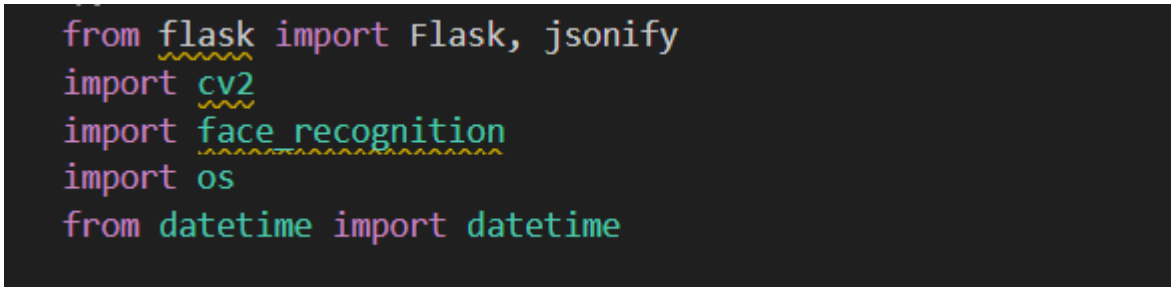
4. Error Handling

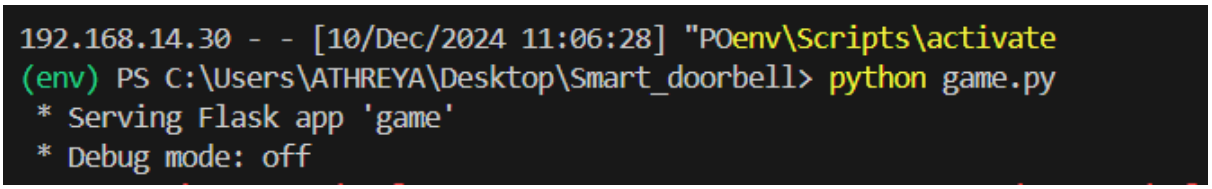
---

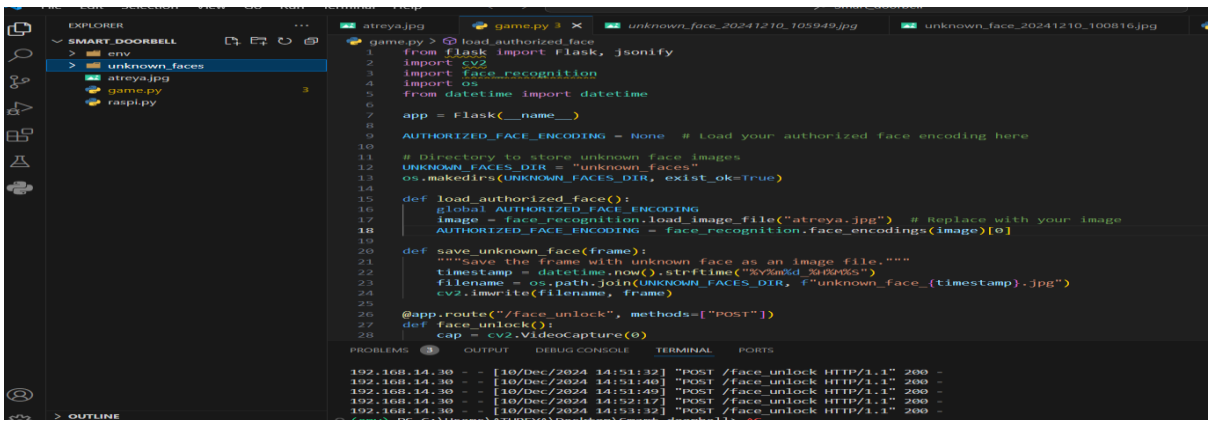
1. **Button Press Not Detected**
    - Check that the button is correctly connected to GPIO pin 18 and ensure proper resistor configuration.
2. **Camera Issues**

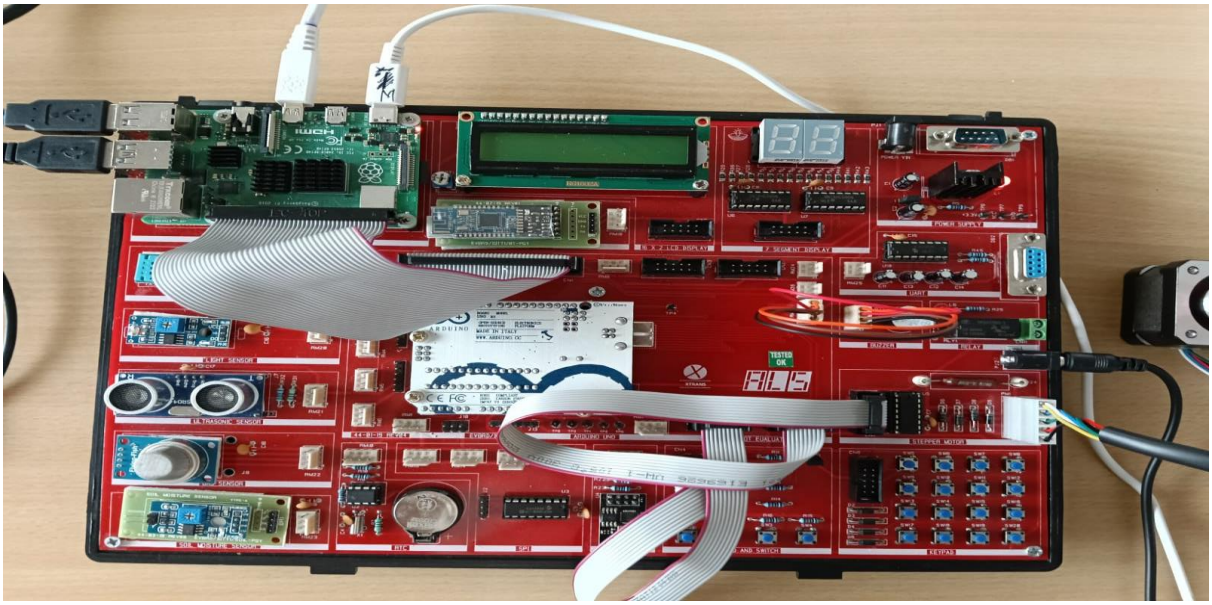
- Verify the camera is securely attached and the interface is enabled in Raspberry Pi settings.

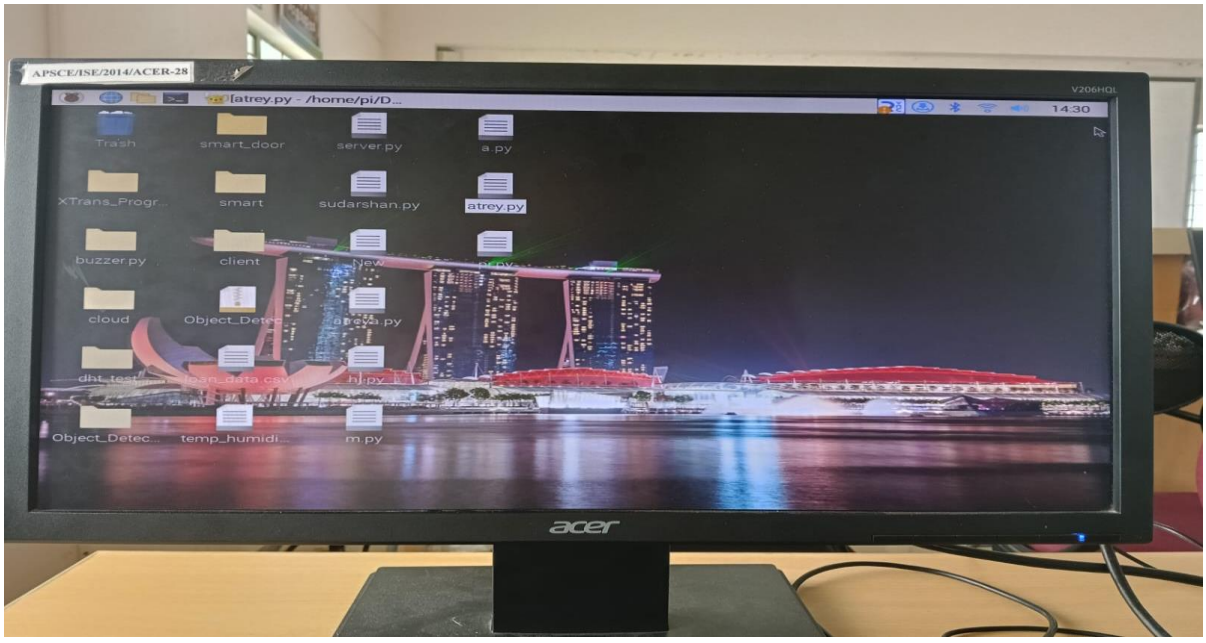
3. **Web Interface Not Loading**
   - Confirm the Flask app is running and check for any errors in the terminal logs.
4. **Audio Communication Problems**
   - Ensure the speaker and microphone are correctly connected and configured for Raspberry Pi.

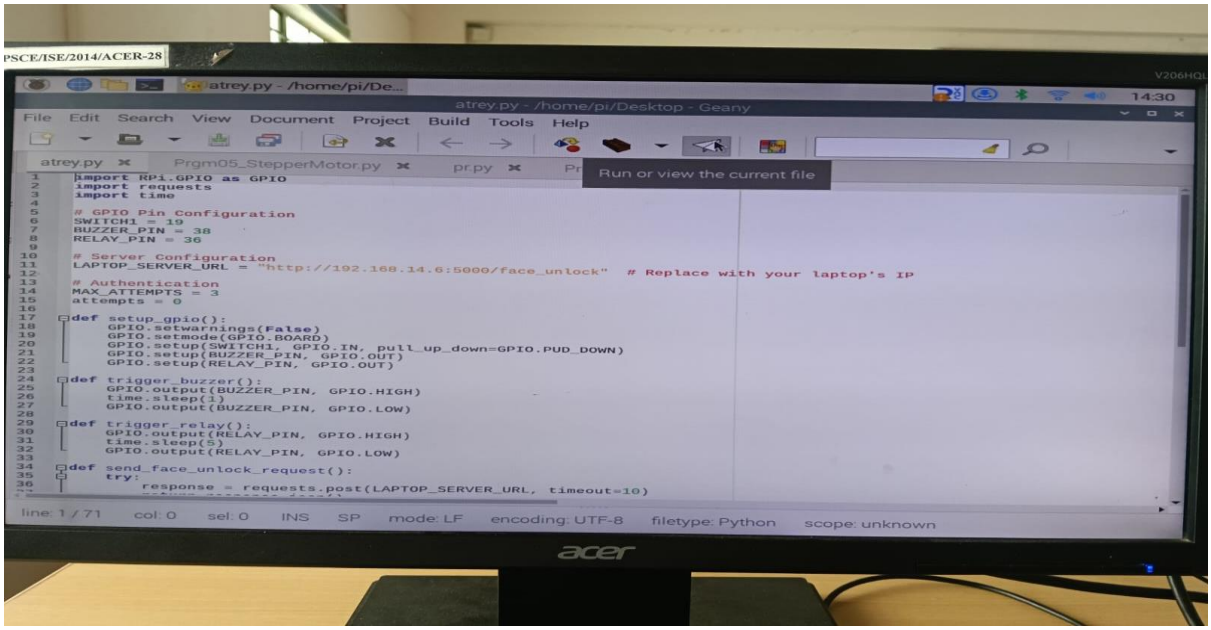


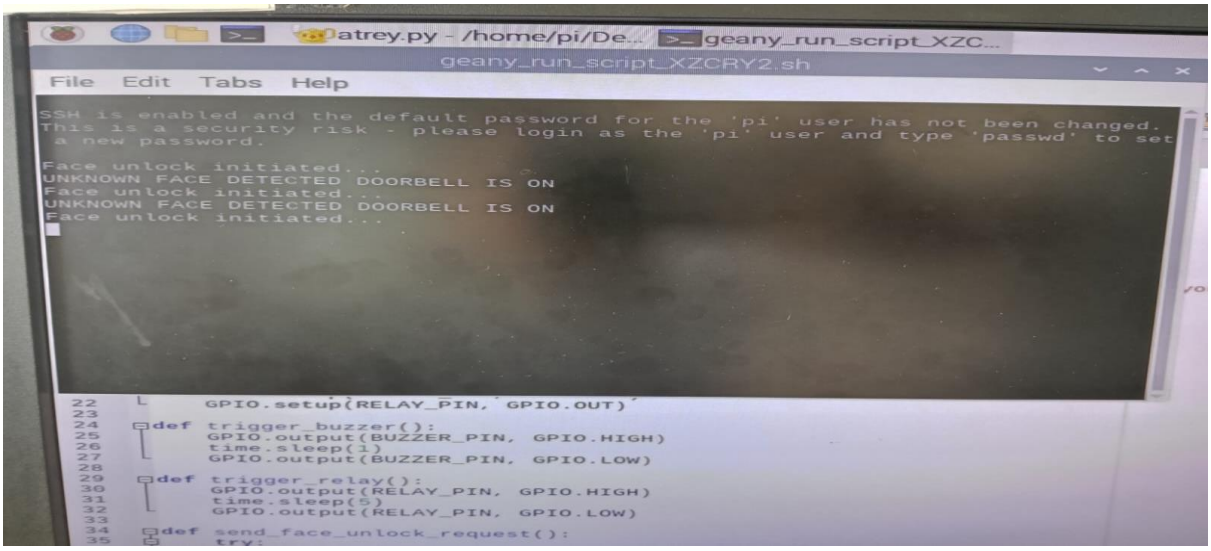







APSCE/ISE/2014/ACER-28
V206HQL
[atrey.py - /home/pi/D...
14:30
Trash
smart_door
server.py
a.py
XTrans_Progr...
smart
sudarshan.py
atrey.py
buzzer.py
client
New
cloud
Object_Detec...
dht_test
hi.py
Object_Detec...
temp_humidi...
m.py
acer


PSCE/ISE/2014/ACER-28
atrey.py - /home/pi/Desktop - Geany
File Edit Search View Document Project Build Tools Help
Run or view the current file
import RPi.GPIO as GPIO
import requests
import time
# GPIO Pin Configuration
SWITCH1 = 19
BUZZER_PIN = 38
RELAY_PIN = 36
# Server Configuration
LAPTOP_SERVER_URL = "http://192.168.14.6:5000/face_unlock" # Replace with your laptop's IP
# Authentication
MAX_ATTEMPTS = 3
attempts = 0
def setup_gpio():
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(SWITCH1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(BUZZER_PIN, GPIO.OUT)
GPIO.setup(RELAY_PIN, GPIO.OUT)
def trigger_buzzer():
GPIO.output(BUZZER_PIN, GPIO.HIGH)
time.sleep(1)
GPIO.output(BUZZER_PIN, GPIO.LOW)
def trigger_relay():
GPIO.output(RELAY_PIN, GPIO.HIGH)
time.sleep(5)
GPIO.output(RELAY_PIN, GPIO.LOW)
def send_face_unlock_request():
try:
response = requests.post(LAPTOP_SERVER_URL, timeout=10)
line: 1 / 71 col: 0 sel: 0 INS SP mode: LF encoding: UTF-8 filetype: Python scope: unknown
acer


atrey.py - /home/pi/De...
geany_run_script_XZC...
geany_run_script_XZCRY2.sh
File Edit Tabs Help
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.
Face unlock initiated...
UNKNOWN FACE DETECTED DOORBELL IS ON
Face unlock initiated...
UNKNOWN FACE DETECTED DOORBELL IS ON
Face unlock initiated...
GPIO.setup(RELAY_PIN, GPIO.OUT)
def trigger_buzzer():
GPIO.output(BUZZER_PIN, GPIO.HIGH)
time.sleep(1)
GPIO.output(BUZZER_PIN, GPIO.LOW)
def trigger_relay():
GPIO.output(RELAY_PIN, GPIO.HIGH)
time.sleep(5)
GPIO.output(RELAY_PIN, GPIO.LOW)
def send_face_unlock_request():
try: