**FINGERPRINT AND FACE RECOGNITION DOOR LOCK SYSTEM**

**TEAM MEMBERS:**

**Subitsha T L K - 2018115114**

**Prathiksha P - 2018115074**

Our project deals with the design and implementation of a two-level secure locking system using Raspberry Pi. The module contains a secured face recognizer and authenticate finger print system for automatic door unlocking.Only when both fingerprint and face recognition inputs match with an authorised user, the door lock gets unlocked. This provides essential security to our homes, workplaces, bank lockers etc and sends alert through the LED module.

**HARDWARE COMPONENTS REQUIRED:**

1. **Raspberry Pi**
2. **Fingerprint Scanner with board**
3. **Relay Module**
4. **Green and Red LED**
5. **Jumper wires**
6. **Pi Camera**
7. **Motor**

**SOFTWARE REQUIRED:**

1. **Open CV**
2. **Python IDE**

**FACE RECOGNITION SYSTEM DESCRIPTION:**

There are few objectives to design the face detection system. The objectives are:

1. To design real time face detection system.

2. To utilize the face detection system based on Haar classifier.

3. To develop face detection system using open CV.

We have attached our program codes in the .zip file under a folder named “Code”.

**CODE FOLDER CONTENTS**

1. encode\_faces.py – This file will find the faces in our dataset and encode them into 128-d vectors.
2. pi\_face\_recognition.py – This is the main code.
3. encodings.pickle – The face encodings are stored in this pickle file.
4. haarcascade\_frontalface\_default – This is used to detect and localize faces in frames. We rely on OpenCV’s Haar cascade file

**DATASET FOLDER CONTENTS**

This folder contains the dataset that we used to train this project. The pictures are stored here.

**DEMO SHOWN TO MAM FOLDER CONTENTS**

This folder contains the demo video that we showed during demo.

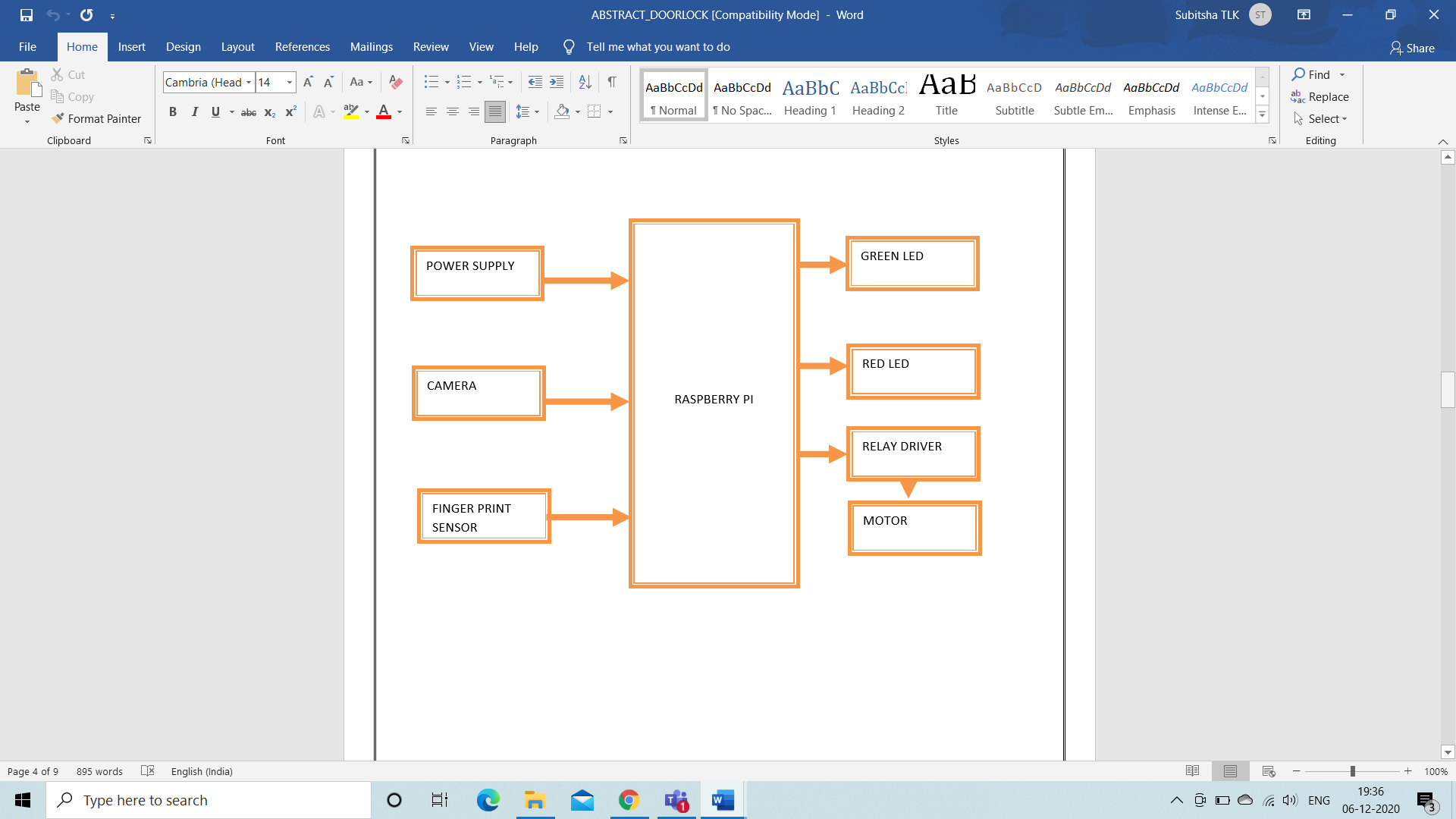
**VIDEOS OF DIFFERENT CASES OF THE PROJECT**

In this folder, we have the videos of different possible cases of the project.

**Case 1**: When an unauthorized fingerprint is given as input, “Unauthorized” is printed in the command line and the door remains locked. **Red LED lights up**. This case is demonstrated in the video named “unauthorized fingerprint\_demo” and “unauthorized fingerprint\_demo2”.

**Case 2**: When an authorised fingerprint and an unauthorized face recognition are given as inputs, “Unauthorized” is printed in the command line and the door remains locked. **Red LED lights up**. This case is demonstrated in the video named “authorized fingerprint\_unauth face\_demo.”

**Case 3**: When both the fingerprint and face recognition inputs are matched, then the user is an authorized person and the door lock unlocks. The command line prints the name of the user, say user-1 or user-2. **Green LED lights up.** This case is demonstrated in the video named “demo\_embedded”.

****

**HOW TO RUN THIS CODE?**

**In the Raspberry pi terminal line type the following,**

**python pi\_face\_recognition.py –cascade haarcascade\_frontalface\_default.xml --encodings encodings.pickle**