Diy

Group 7

Computer Vision based:
Facial recognition- Security and attendance system.

Written by

Mohd arsalan
Avnish kumar
Gourav kumar
Trisha sarwade
Vardhan reddy

Index

Acknowledgement

With lot of hardship in online mode. by removing all the barriers of virtuality, by synchronizing the group with great enthusiasm for which we are extremely grateful to our mentors
Mr. Debapratim Pandit, Mr. Arijit Mondal and Mr. Jayan Coral, for their advice and guidance without which this project would not have been complete.

We would also like to thank all the TAs for their support and help by the virtue of which this project has been successfully completed. Also, we would like to thank our parents for their motivation and patience. All our fellow batchmates as well as seniors who advised us and helped at the time of need, fellow group members for backing each other up and facing all odds together.

Ultimately, we would like to thank the supreme power for providing us strength to face all the obstacles and challenges.

Earlier Solutions

We have to maintain the system physically by many persons for monitoring this biometric system

Many records are to be maintained by involving many persons physically and it also requires huge volume of place for preservation of these records which are important for any further verification for any legal issues.

Above all, it is time consuming, less accurate , and there is a possibility for tampering of records

To overcome all the problems, it is very essential and inevitable ……

Our Solution

In this project, we will see a two step security system. It consists of face recognition and fingerprint matching. Followed by which the date and time of scanning is also recorded.
This system can be implemented in any sphere or domain of a security requirement.
It is a well built system as it is accomplished by two steps of verification. Thus toughening it more.
Hence, in this diy project we have implemented (a good project name) so that there will not be any security issues

Softwares Used

Python Libraries included:

```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime
import pyttsx3
import serial
import time
```

Displaying names of image present in the Database:

```
path = r'C:\Users\goura\Desktop\DIY_Week7_Group Project\ImagesAttendance'
images = []
classNames = []
myList = os.listdir(path)
print(myList)
for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])

print(classNames)
```

Voice alert using Pyttsx3 Library:

```
def speak(audio):
    engine.say(audio)
    engine.runAndWait()



engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty("voice", voices[0].id)
engine.setProperty("rate", 140)
engine.setProperty("volume", 1000)
```

Encoding images:

```
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
```

Attendance marking function:

```
def markAttendance(name):
    with open(r'C:\Users\goura\Desktop\DIY_Week7_Group Project\attendance01.csv','r+') as f:
        myDataList = f.readlines()
        nameList = []
        for line in myDataList:
            entry = line.split(',')
            nameList.append(entry[0])
        if name not in nameList:
            now = datetime.now()
            dtString = now.strftime('%H:%M:%S\n')
            f.writelines(f'n{name},{dtString}')
```

Capturing of images for matching using webcam:

```
while True:
    success, img = cap.read()
    #img = captureScreen()
    imgS = cv2.resize(img,(0,0),None,0.25,0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

    facesCurFrame = face_recognition.face_locations(imgS)
    encodesCurFrame = face_recognition.face_encodings(imgS,facesCurFrame)

    for encodeFace,faceLoc in zip(encodesCurFrame,facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown,encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown,encodeFace)
        #print(faceDis)
        matchIndex = np.argmin(faceDis)
```

On successful match:

```
    if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            print(name)
            time.sleep(3)

            y1,x2,y2,x1 = faceLoc
            y1, x2, y2, x1 = y1*4,x2*4,y2*4,x1*4
            cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)
            cv2.rectangle(img,(x1,y2-35),(x2,y2),(0,255,0),cv2.FILLED)
            cv2.putText(img,name,(x1+6,y2-6),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),2)
            markAttendance(name)
            cv2.imshow('Webcam',img)
            cv2.waitKey(1)
            time.sleep(5)
            speak("Face is matching with database."+name)
            speak("Please enter your finger print for authentication")
            ard = serial.Serial('com3', 9600)
            time.sleep(2)
```

Components

Arduino

Overview:

Arduino Uno is a microcontroller board based on an 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists of other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

How to use Arduino board:

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in arduino programming. Each pin operate at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using analogWrite() function.

SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.

In-built LED Pin 13: This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, it's off.

)

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz


AREF
GND
DIGITAL - PWM~
TX
RX
ARDUINO
UNO
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
Vin
A0
A1
A2
A3
A4
A5

Fingerprint sensor

The fingerprint sensor is one kind of sensor which is used in a fingerprint detection device. These devices are mainly inbuilt in the fingerprint detection module The main features of this device mainly include accuracy, better performance, robustness based on exclusive fingerprint biometric technology.

The R307 fingerprint module is used in this project. It is a fingerprint sensor with adafruit fingerprint library The user can store the fingerprint data in the module and can configure it in 1:1 or 1: N mode for identifying the person. A level converter (like MAX232) is required for interfacing with PC serial port.

R307 Fingerprint Module consists of high-speed DSP processor, high-performance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition, stable performance, simple structure, with fingerprint entry, image processing, fingerprint matching, search and template storage and other functions.

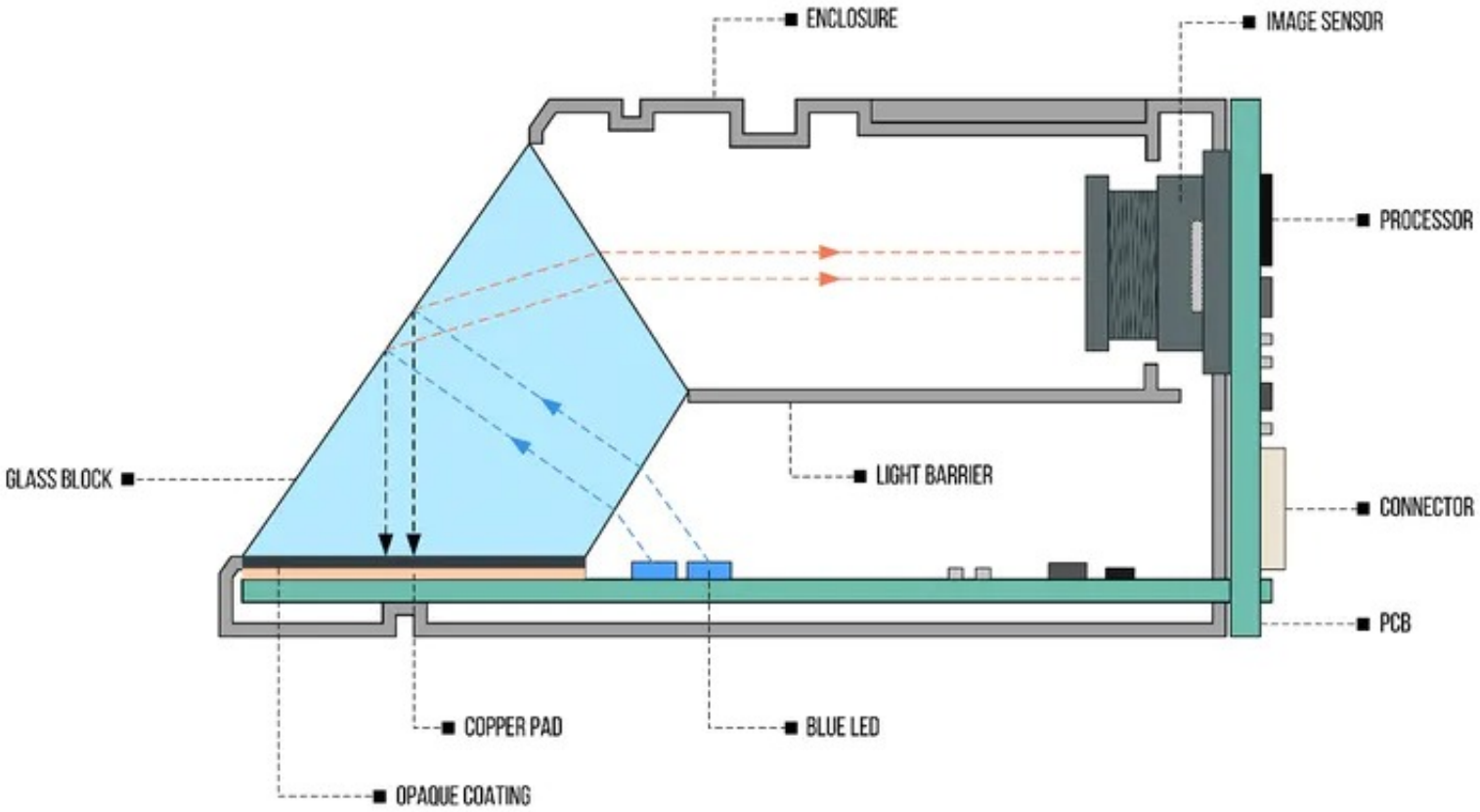


FIG. R307 FINGERPRINT SCANNER CROSS-SECTION

Fingerprint working code:(.ino)

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
void setup()
{
Serial.begin(9600);
while (!Serial);
delay(100);
Serial.println("fingertest");
pinMode(12, OUTPUT);
pinMode(11, OUTPUT);
// set the data rate for the sensor serial port
finger.begin(57600);



if (finger.verifyPassword()) {
Serial.println("Found fingerprint sensor!");
} else {
Serial.println("Did not find fingerprint sensor :(");
while (1) {
delay(1);
}
}
finger.getTemplateCount();
Serial.print("Sensor contains "); Serial.print(finger.templateCount);
Serial.println(" templates");
Serial.println("Waiting for valid finger...");
}
int c=0;
char d;
```

```
void loop()
{
if(Serial.available())
{
d=Serial.read();
}
if(d=='a'){
while(c<3){
getFingerprintIDez();
delay(500);
digitalWrite(12, LOW);
digitalWrite(8, LOW);
c++;
}
}
d="";
}
uint8_t getFingerprintID() {
uint8_t p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.println("No finger detected");
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
return p;
default:
Serial.println("Unknown error");
```

```
return p;
}

p = finger.image2Tz();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
default:
Serial.println("Unknown error");
return p;
}
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_NOTFOUND) {
Serial.println("Did not find a match");
return p;
} else {
```

```
Serial.println("Unknown error");
return p;
}
{digitalWrite(8, HIGH);
delay(3000);
digitalWrite(8, LOW);
Serial.print("Not Found");
Serial.print("Error");
return finger.fingerID;
}

Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);
return finger.fingerID;
}

{
delay(3000);
Serial.write("o");
digitalWrite(12, HIGH);

delay(7000);

digitalWrite(12, LOW);
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);
} }
```

Solenoid lock

This DC 12V Cabinet Door Lock Electric Lock Assembly Solenoid can be used for locking shell-machine, storage shelf, file cabinet, etc. The hidden way of unlocking can be used for an emergency. The lock works as the circuit disconnects, and it will unlock as the instant power-on. It is steady, durable, and energy-saving and has a long lifespan. In the anti-theft and shockproof design, the lock is better than other kinds of locks. After connecting the wires and when the current is available, the electric lock can control the door's opening and closing.

12V Solenoid lock has a slug with a slanted cut and a good mounting bracket. It's basically an electronic lock, designed for a basic cabinet, safe or door. When 9-12VDC is applied, the slug pulls in so it doesn't stick out and the door can be opened. It does not use any power in this state. It is very easy to install automatic door lock systems like electric door locks with the mounting board. This solenoid in particular is nice and strong.

Specifications of 12V solenoid lock:

Operating voltage : 12VDCDraws 650mA at 12V,

500 mA at 9V when activated

Designed for 1-10 seconds long activation time

Wire length: 222.25mm

Relay module

A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit.

Relays are used to provide time delay functions. They are used to time the delay open and delay close of contacts. Relays are used to control high voltage circuits with the help of low voltage signals. Similarly they are used to control high current circuits with the help of low current signals.

The advantage of relays is that it takes a relatively small amount of power to operate the relay coil, but the relay itself can be used to control motors, heaters, lamps or AC circuits which themselves can draw a lot more electrical power.

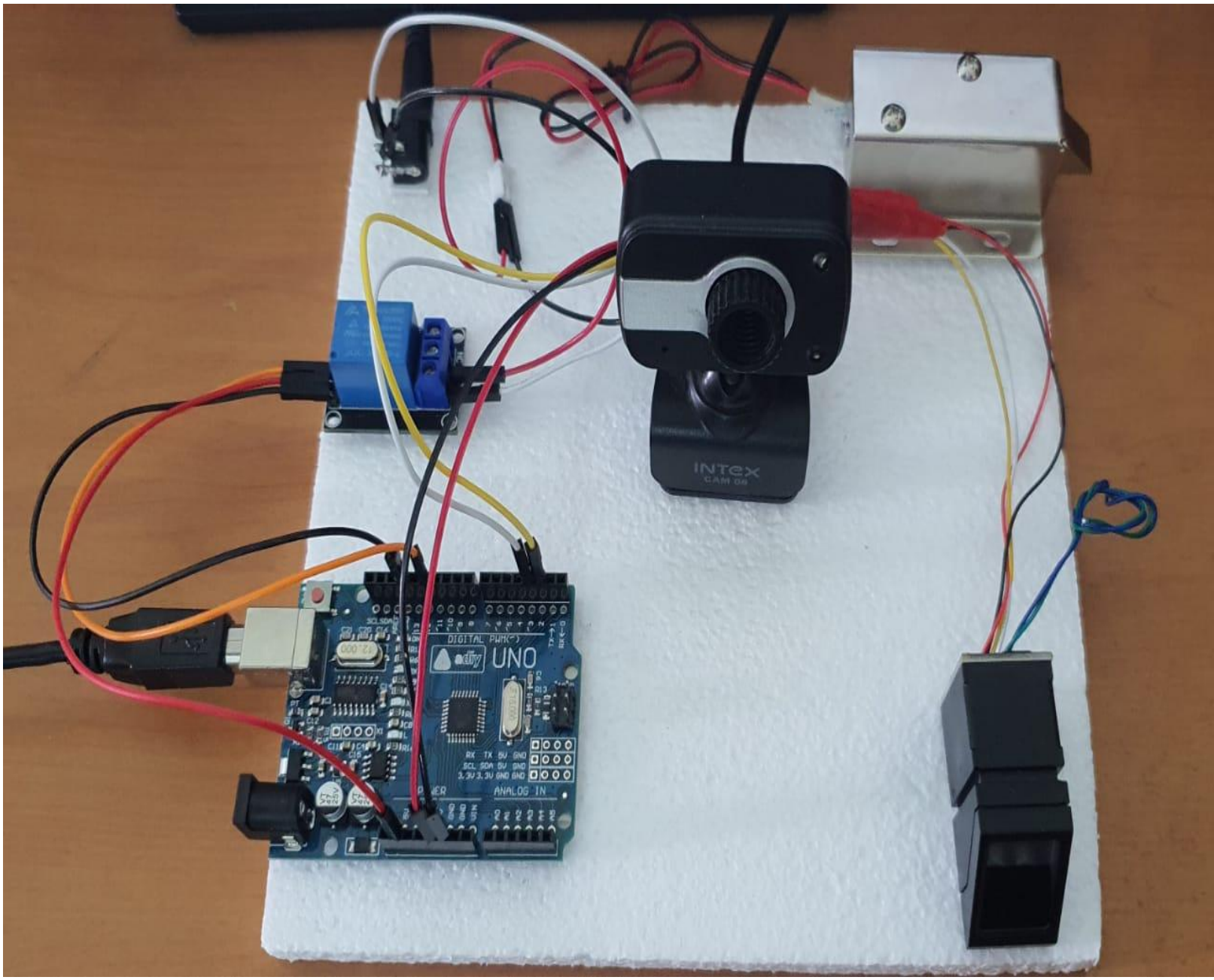
Connecting wires:

Connecting wires provide a medium to an electrical current so that they can travel from one point on a circuit to another. In the case of computers, wires are embedded into circuit boards to carry pulses of electricity.

In a basic circuit, the wire comes from one terminal of a power source, then connects to a switch that determines whether the circuit is open or closed. The connected wires of a device are used to draw power and electricity, and perform specific tasks. Before a current can travel through the wire, the circuit has to be closed; in other words, there cannot be any breaks in the path.

Electricity cannot easily travel through air, and if it does there is a risk of stray current leaking into the surroundings and causing damage or failing to power the appliance.

Circuit Diagram



Working of the project

Firstly, the setup is initialised by connecting the camera to the arduino UNO board and its stimulation is begun by the code.

The fingerprint sensor is connected to the Arduino board and its stimulation is also begun by the code.

Next, the camera starts. A python code is used by the camera to recognize the face.

If the face is recorgnized then command coes to finger print or else that trail is terminated

The fingerprint sensor scans the fingerprint and records it.

If the fingerprint is recognized,Attendance is also taken by recording the date and time of scanning. the door opens with the help of solenoid.

Application

This project has many applications. Some of the have already been mentioned above.

A few more are:

1)Schools

2)Offices

3)Safety lockers

4)Banks

5)Hospitals

Major features:

1)To accomplish digitisation of the current system.

2)This is a smart enhanced security lock.

3)Dual security is enabled in the form of fingerprint and face recognition .

4)This lock is portable , lightweight and durable .

5)This lock system is versatile as we can use it in a door ,as a bank lock ,or any such high security place.

Learnings:

1)Learned the usage, features and working of arduino, fingerprint sensors and solenoi

2)Gave us a basic idea of implementing programming to real life projects.

3)Practical understanding of working in a team and coordinating efficiently with eac other.