Chandigarh Engineering College Jhanjeri
Mohali - 140307
Department of Computer Science & Engineering

Project Report
On
Live Logging Facial Recognition Attendance System

Project – II

Department of Computer Science & Engineering
CHANDIGARH ENGINEERING COLLEGE JHANJERI, MOHALI

In partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science & Engineering

SUBMITTED BY:

Prince Kumar	1816590
Abhinandan Kumar	1816823
Prerna Kumar	1816589

Under the Guidance of

Mrs. Sakshi Sharma
Asst. Professor

DEC 2021

Affiliated to I. K. Gujral Punjab Technical University, Jalandhar

(Batch: 2018 – 2022)

Chandigarh Engineering College Jhanjeri
Mohali - 140307
Department of Computer Science & Engineering

DECLARATION

I, Prince Kumar and my team hereby declare that the report of the project entitled "Live Logging Facial Recognition Attendance System" has not presented as a part of any other academic work to get my degree or certificate except Chandigarh Engineering College Jhanjeri, Mohali, affiliated to I.K. Gujral Punjab Technical University, Jalandhar, for the fulfilment of the requirements for the degree of B Tech in Computer Science & Engineering.

Prince Kumar
1816590
Semester: 7th

Mrs. Sakshi Sharma
Asst Professor, CSE

Dr. Anil Kumar Lamba
HOD CSE

ACKNOWLEDGEMENT

It is great pleasure to present this report on the project title **"Live Logging Facial Recognition Attendance System"** undertaken by me as part of my B. Tech degree in Computer Science & Engineering final year. I am thankful to my university for offering me such a wonderful challenging opportunity and I express my deepest thanks to all coordinators, for providing all the possible help and assistance and their constant encouragements.

I am extremely thankful to **Dr. Anil Kumar Lamba,** HoD, Computer Science & Engineering, Chandigarh Engineering College Jhanjeri, Mohali (Punjab) for valuable suggestions and motivation.

I am also thankful to all my teachers who have taught me till date in my degree, management of institute and **Dr. Rajneesh Talwar,** Director Engineering for providing me the opportunity to get the knowledge.

Prince Kumar

TABLE OF CONTENTS

Chapter- 1
Introduction

1.1 Introduction

This project is based on the technology of Computer Vision in AI, it marking attendance from detecting faces of person. It detects the face of person in real time using webcam. The Computer Vision techniques we use in this project is image analysis, face detection, face-recognition.

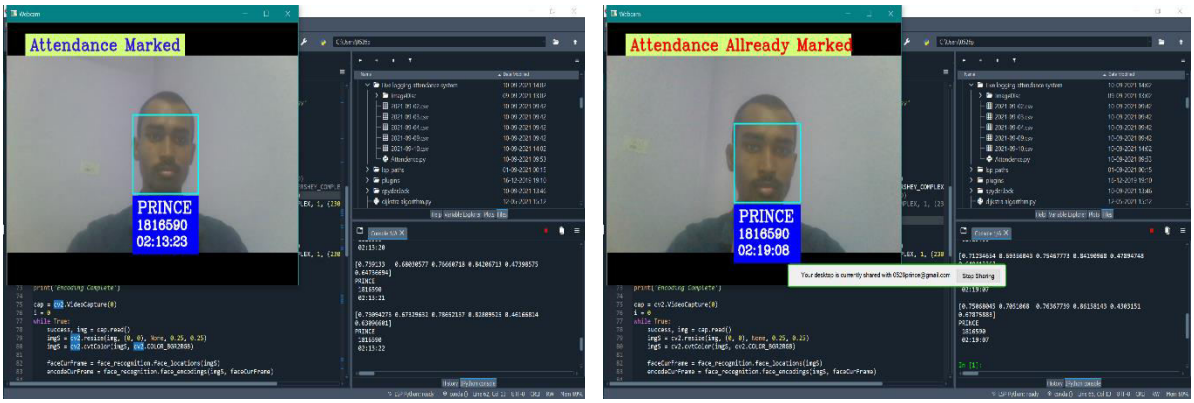
This Model is checks that captured image attendance is marked or not and take the decision based on the data which it will be get, processed it and give the output. If the attendance of that capture image is not marked then model mark it. If the attendance of that captured image is already marked then the model shows attendance already marked of that captured person image. So, we can proceed it to the next one and same process is repeated for the every person.

1.2 What is a Computer Vision?

Computer Vision is a field of Artificial Intelligence that enables computers and systems to grab information from digital images, videos and in real time videos and takes actions or make decision and recommendations based on that information. Computer Vision is simply means we make a model which enables to see, observe, understand and make decisions.

Computer vision works as like human visions. Computer visions trains machines to perform these functions but it has to do in very less times with cameras, data and algorithms. Because a model trained to observe products or process a minute, detect defects or issues, it can quickly surpass human capabilities.

Person, they want to mark their attendance



Marking Attendance

1.3 Objectives of Face Recognition Attendance System

From a long term we use the old method for taking attendance using paper or pen. One by one we call the roll no or name of student and then we mark it present or absent. We also uses many papers or pages for store the data of attendance. And also teachers are facing many problems while taking attendance. Sometimes due to miscommunication teacher mark the wrong attendance.

For store the data of attendance we use lot of papers. So, we have to cut a large amount of tree in whole world for accruing the demand of papers. According to the survey a single tree produces 10,000 to 20,000 papers.

Papermaking has an impact on environment because it destroys trees continuously in every single day. According to the data from the Global Forest Resource Assessment roughly 80,000 to 160,000 trees are cut down each day around the world for papermaking by the paper industry. So we are made a digital method to mark attendance of peoples without using pen and paper. Therefore, we are work productively and comfortably.

The goal of this project is to detect and locate human faces in webcam and encode it compare the images which are present in the student or specific directory. We need to make a directory of images where all the images of those student or persons are present which one I want to mark attendance. A set of images were provided for this purpose. The objective was to design and implement a face detector that will detect human faces in webcam similar to the images present in the directory. Where the model compares both images and mark attendance for more accurate input images. If the attendance is not marked, then attendance of particular person is marked. On the other side if the attendance is already marked then it do not overwrite it. All the data of attendance system is mark automatically and maintain the CSV sheet automatically by the model. We do not need to care about that what will be going on the backend. Person only work to show his/her faces in the webcam.

Chapter- 2

Review of Literature

2.1 Anaconda 3: You must have anaconda 3 in your system. You can install anaconda 3 using any browser. After successful installation you have a lot of choice for your programming interface such as Jupyter, Pycharm, Visual Studio, Sypder etc. For professional programming and for convenient Spyder is better than anyone and also for huge programming Spyder is more comfortable for any of those. It has more helpful tools which helps you while coding.

2.2 Spyder: As we know there are many open-source software are availabe for making a perfect model some of the open-source software is Pycharm, Atom, Spyder, Sublime Text, Jupyter. In this project we were using Spyder from Anaconda for building this model. You can also use Jupyter or any other software as according to your comfort. But we personally recommended uses Spyder for writing code and implementation. There have lots of advantages why you choose this software. Few of them are given below.

2.2.1 Advantages of Spyder:

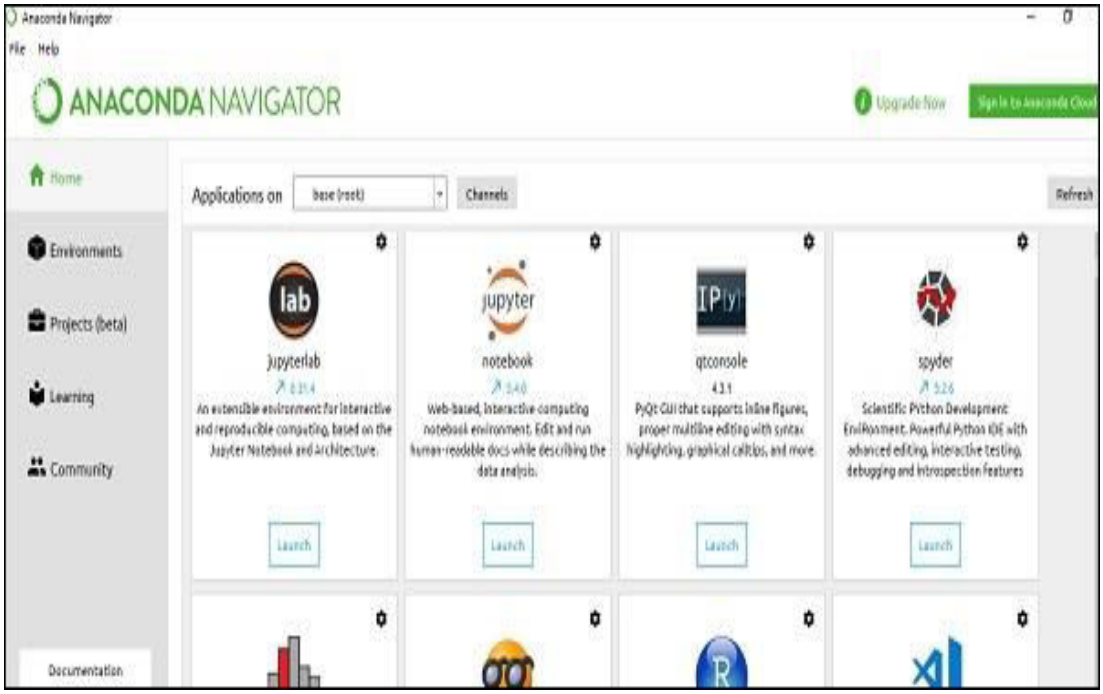
- It is a cross – platforms and open – source IDE for python. It is available for Windows, Linux, and MAC OS
- It analyse code to provide automatic code completion, horizontal/vertical splitting, and a go-to-definition.
- It is specifically designed for data scientists; hence, it integrates well with data science libraries like NumPy, Pandas, Tensorflow, OpenCV, Scipy, and so on.
- It allows you to run and create the multiple IPython console you have no restriction to work on the particular or only on the current console. You are free to work and run code or check debugging in any console that's the more interesting features why we choose Spyder.
- It includes a powerful integrated debugger which is very beneficial for analysing code in real time what is the error in particular section of program.
- It contains an integrated documentation browser which helps you study or read any time anywhere like online help, are great for beginners.

2.2.2 Steps for installing libraries:

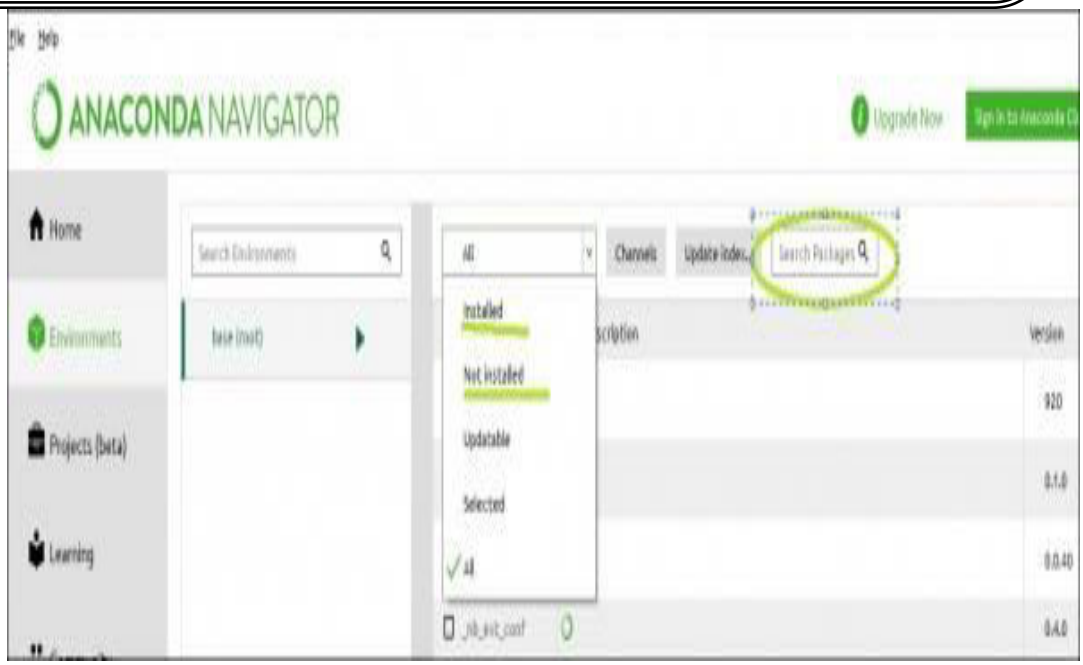
In Spyder most of the libraries is preinstalled but some of the libraries according to our requirement needed to be installed. We can install all those libraries in two ways either Anaconda Prompt/terminal or directly from Anaconda environment. We mention both of the method you can go through any of them:

Method 1:

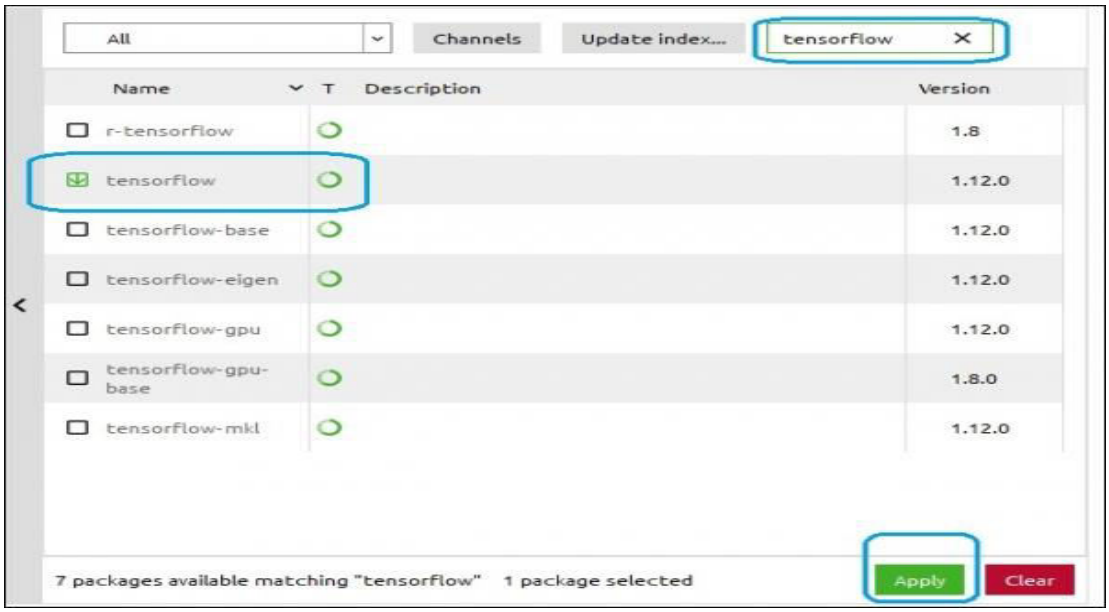
- Go to the anaconda navigator



- Click on the “Environment” tab and from there we can check which libraries are installed or not. After that click on the search bar then type libraries name which one you want to install.



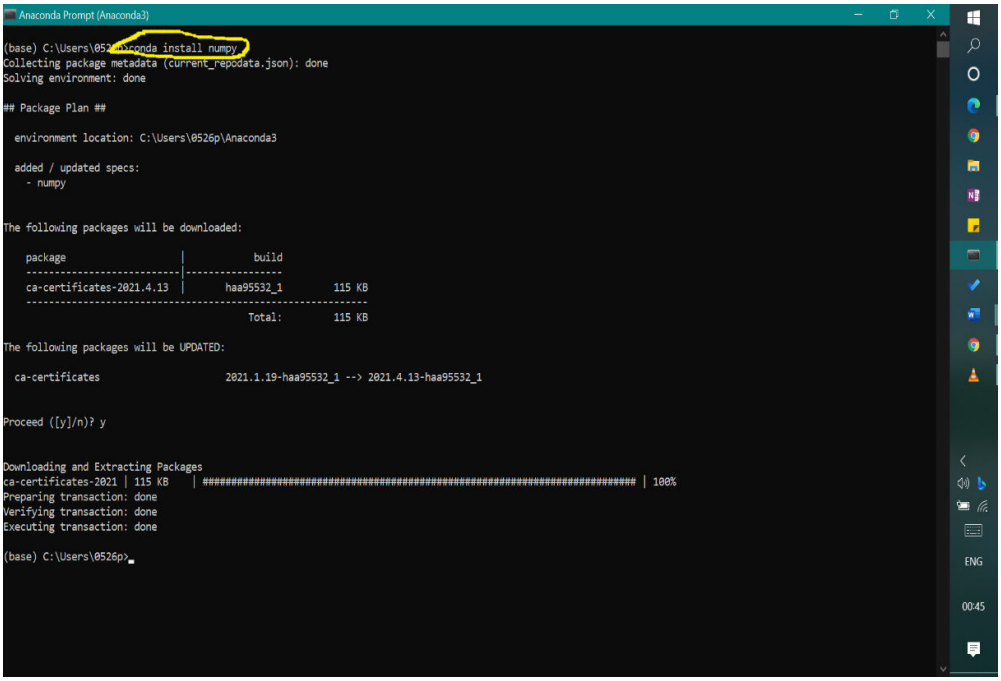
- It shows is the library installed or not. If not then select the library option you want to install and click the apply button your library is install in your Anaconda environment.



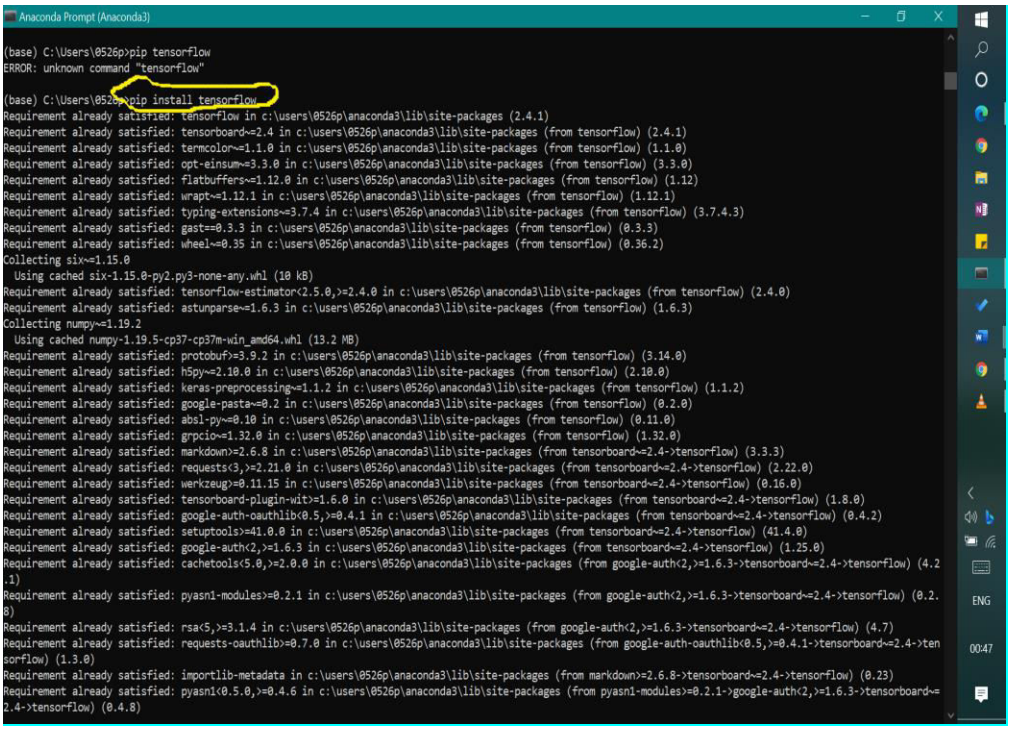
Method 2:

- Open Anaconda prompt or terminal (it also supports pip command)

condainstall numpy



pip install tensorflow (it also woks)



Chapter- 3

Problem Definition and Objectives

3.1 Facial Landmarks

Facial landmarks allow us to automatically infer the location of facial structures, including:

Eyes, Eyebrows, Nose, Mouth, Jawline

- To use facial landmarks to build a dataset of faces wearing face masks, we need to first start with an image of a person *not* wearing a face mask:

- From there, we apply face detection to compute the bounding box location of the face in the image. The next step is to apply face detection. Here we've used a deep learning method to perform face detection with OpenCV. Once we know *where* in the image the face is, we can extract the face Region of Interest (RoI).

- The next step is to extract the face ROI with OpenCV and NumPy slicing. And from there, we apply facial landmarks, allowing us to localize the eyes, nose, mouth, etc. Then, we detect facial landmarks using dlib so that we know where to place a mask on the face. Calculate the encoding of image and face distance of every image. Apply the same process on the real time image in the video frame. And compare the encoding of both the image (image in database and capture image from real time video frame). If both are matched then mark attendance else not continue to capture the face in the real time video frame.

3.2 Camera Analysis

3.2.1 Video Stream

1. Optimum resolution for the module's operation: HD or Full HD.
2. Frame rate: 10 fps or more.
3. No mirrored (horizontally flipped) stream.

3.2.2 Illumination and image quality

1. Illumination of faces in the frame must be uniform.
2. If the camera is installed opposite a bright source of light (sun behind the entrance door, etc.), it is required to adjust the exposure or brightness in such a way that the face in the frame is light.
3. The image quality must be medium or better. Significant compression artefacts are not acceptable.
4. No blurring of moving people's faces is allowed.

3.2.3 Scene and camera position

1. The faces must be fully seen in the frame.
2. There must be no mirror surfaces giving reflections in the frame (glass, mirrors, etc.).
3. Strong lateral illumination (e.g., sunlight from the window) resulting in the overexposure of one part of the face is not acceptable.

4. The camera may be placed above the face level, directly facing the people to be recognized. In such a case, the camera elevation angle must not exceed 35°. The camera must directly face the people to be recognized.

3.3 Skills

Knowledge of python is first and most compulsory key for making any Machine Learning and AI projects. Must know how and where to uses loop, conditional statements. You have to be good command in functions and OOPs. You can learn python on my github account, link is given in chapter 6 (reference).

3.4 Libraries

For building this model we uses some of the useful libraries. We need to install libraries using the above method we discuss above in chapter 2 ("steps to install libraries" topic). Libraries we need to build this system are as given below:

1. **tkinter:** Tkinter is python's default GUI library. It is based on the Tk toolkit, originally designed for the Tool Command Language (TCL). It has varies according to the other scripting languages, including Perl (Perl/ Tk), Ruby (Ruby/ Tk), and Python (Tkinter).
   Tk have the most robust set of GUI building blocks, but it is fairly simple to use, and with it, you can build GUIs that run on most platforms. Python, along with Tkinter, provides a fast and exciting way to build useful applications that would have taken much longer if you have to program directly in C/ C++ with the native windowing system's libraries.
   You will use basic building blocks known as widgets to piece together the desired. Once you get Tkinter up on your system, it will take less than 15 minutes to get your first GUI application running.
2. **cv2:** For image and video processing, we use the cv2 library for read the file directory. OpenCV (Open Source Computer Vision Library)-Python is a library of python binding designed to solve computer vision problems.
   cv2.imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format then this method returns an empty matrix.
   Note: The image should be in the working directory or a full path of image should be given.

3. **numpy:** NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, derived objects such as masked arrays and matrices and performs fast operations on array like mathematical, logical, shape, manipulation, sorting, selecting, I/O, basic linear algebra, basic statistical operations and many more.

   .

4. **face_recognition:** Face recognition is a method of identifying or verifying the identity of persons using their face. Face recognition is use for detecting faces in the images or in video frames. It contains lots of function such as “encoding” we use it for encode faces from imge or video frames. “face distance” calculate face distance to check the best match “Compare” compare two or more faces using the face encoding if it both face encode is match then it’s face distance is less and vice versa.

   Facial recognition uses computer-generated filters to transform face images into numerical expressions that can be compared to determine their similarity. These filters are usually generated by using deep learning, which uses artificial neural networks to process data

5. **os:** The OS library in python provides functions for interacting with the operating systems. It very efficient for handling file directories or handling file directories in GUI. This libraries provides a portable way of using operating system-dependent functionality. The *os* and *os.path* modules include many functions in this library is used for interact with the file system.
6. **datetime:** It uses to dealing with the date or time. Date and time are not a data type of their own, but a module named datatime can be imported to work with the date as well as time. We are using the library for that purpose for is datetime.

7. **pytz:** It is uses for grab the Indian standard time Asia/ Kolkata. Without depends on CPU or any other devices time source. It grabs automatically when it will be execute. It supports almost all time zones.

   Syntax: astimezone(t)

   Parameter: t is time to be converted

   Return: converted timezone

8. **csv:** A Comma Separated Value (CSV) file is a text file that has a specific format which allows data to be saved in a table structured format. It uses to create CSV data sheet for mark attendance or update the attendance. We can check it on externally. Or we can also convert into excel files according to their needs and desire. It is commonly used file extension when it comes to spreadsheets.

9. **PIL:** Python Imaging Library is a free and open-source library. It is used for opening, manipulating, and saving many different image file formats. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF.

10. **pathlib:** A pathlib library in python provides various classes representing file system paths with semantics appropriate for different operating systems. File paths can be represented by proper path objects instead of plain strings as before. These objects make code dealing with file paths. Easier to read, especially because / is used to join paths together.

Chapter-4

Design and Implementation

4.1 Database

4.1.1 Database Categories: We create Comma Separated Values (CSV) and store the data whatever output given by model. Data categories are tables of data, which are organized by rows and columns. Columns are also known as data fields. A row of data has entries for one or more columns in the category. When you add a data field onto a report you are seeing the information in one column of data for every row in the category.

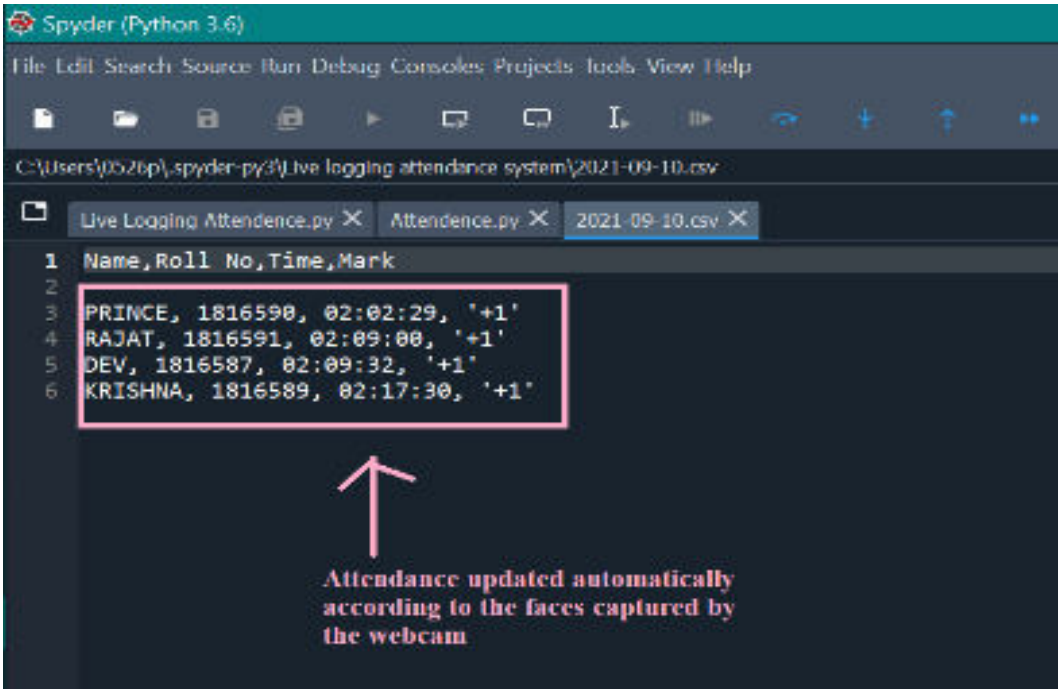


Figure 4.1.1: Database Categories

4.1.2 Data Model: Then we convert the CSV file to the Excel file. A database model is a type of data model that determines the logical structure of database and fundamentally determines in which manner data can be stored, organized and manipulated.

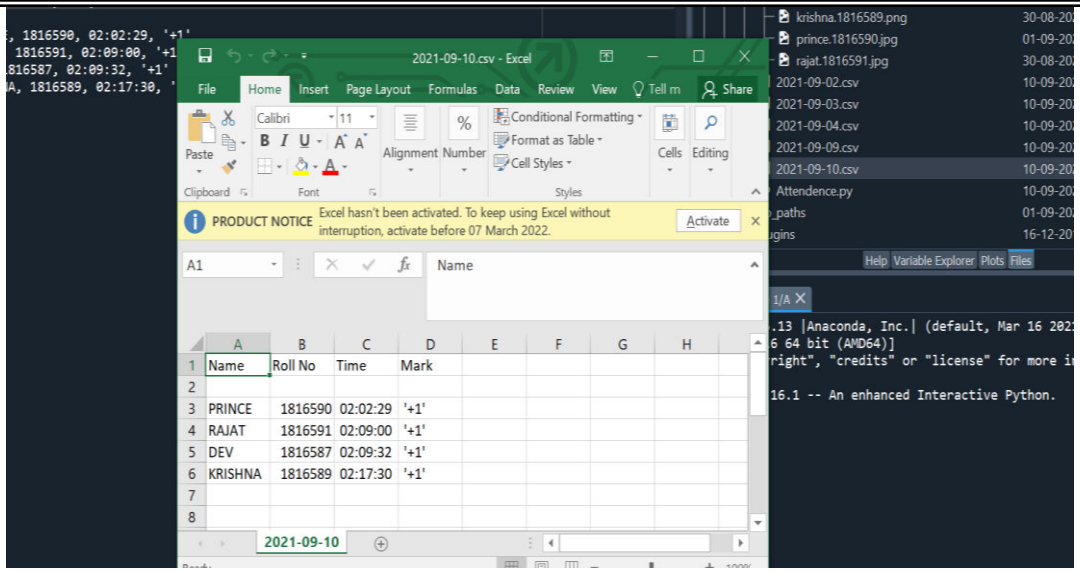


Figure 4.1.2: Data Model

4.2 Flow Chart

A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows.

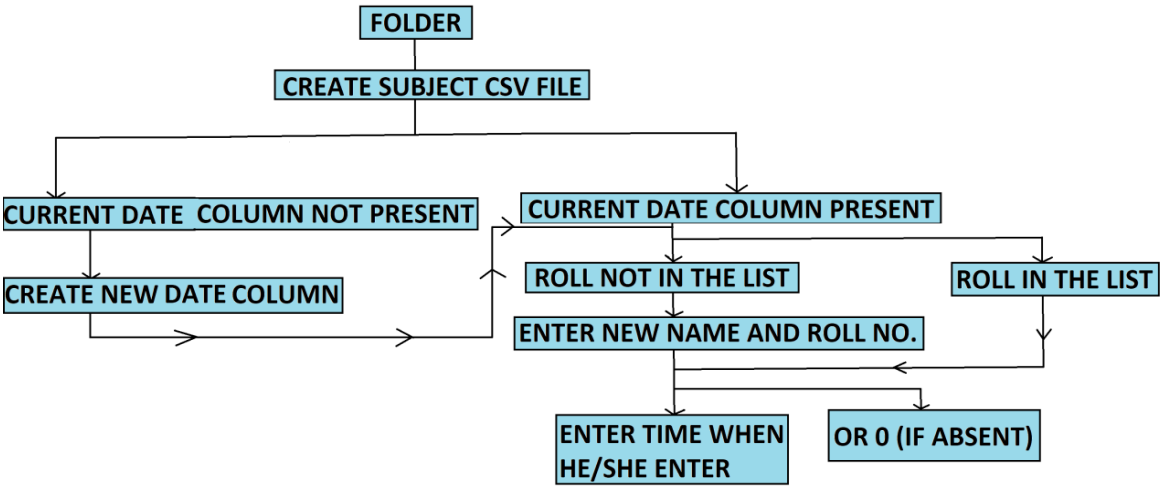


Figure 4.2: Flowchart of LLFRAS

4.3 ER Diagram

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

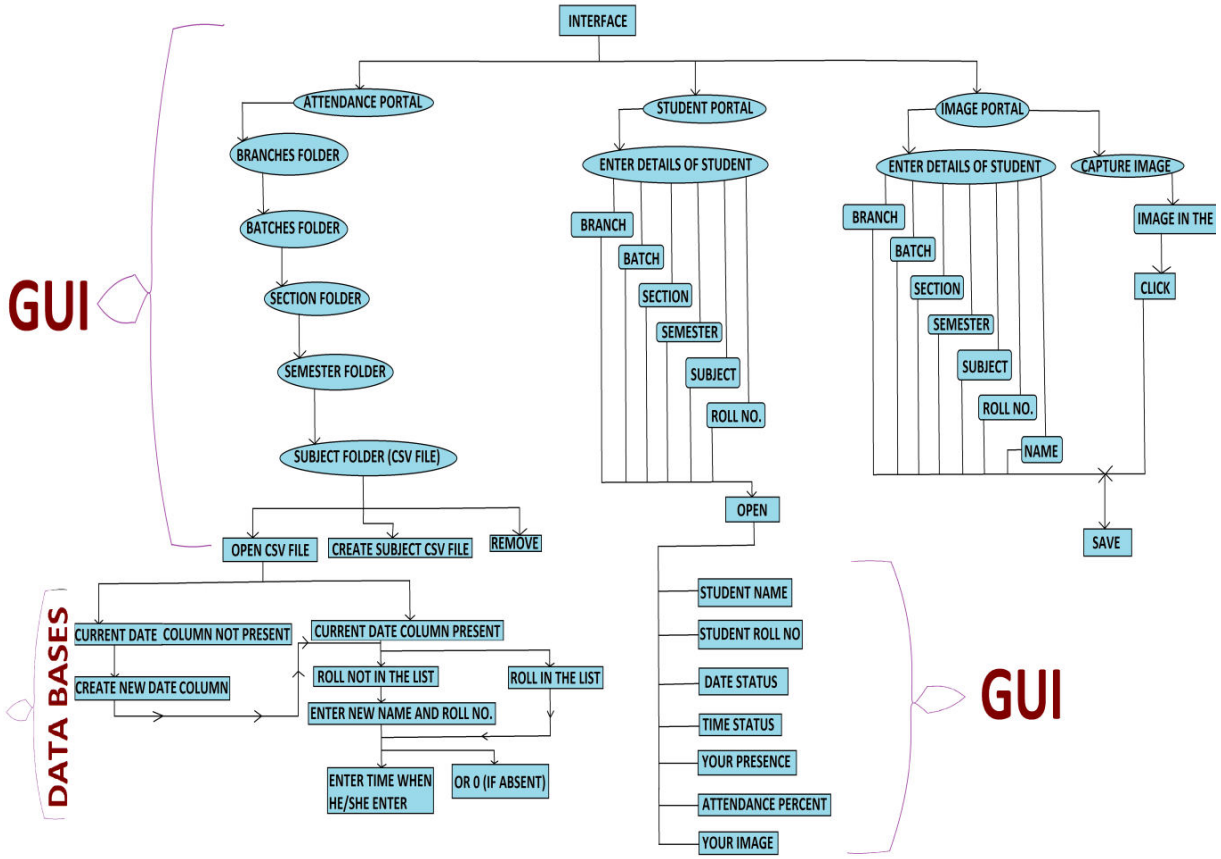


Figure 4.3: ER Diagram of LLFRAS

4.4 Methods to Build Live Logging Facial Recognition Attendance Systems

4.4.1 Loading Image Data

1. Method to read image
2. Method to display image
3. Resizing image
4. Video frames

4.4.2 Encoding of Load Image

1. Encode all the images present in the directory
2. Store all the encode list inside the list

3. While use these encode at the time of compare faces
4. Calculate the face distance of all images present in the directory
5. Lower face distance is the best match and height face distance is the worst match

4.4.3 Features Detection

1. Capture web cam images
2. Using face recognition to extract the ROI
3. ROI(Region Of Interest)

4.4.4 Video Streaming & Real Times

1. Method to capture video
1. Looping through the Frames
2. Video Streaming
3. Encode the faces in the current frame
4. Compare the captured face in real time

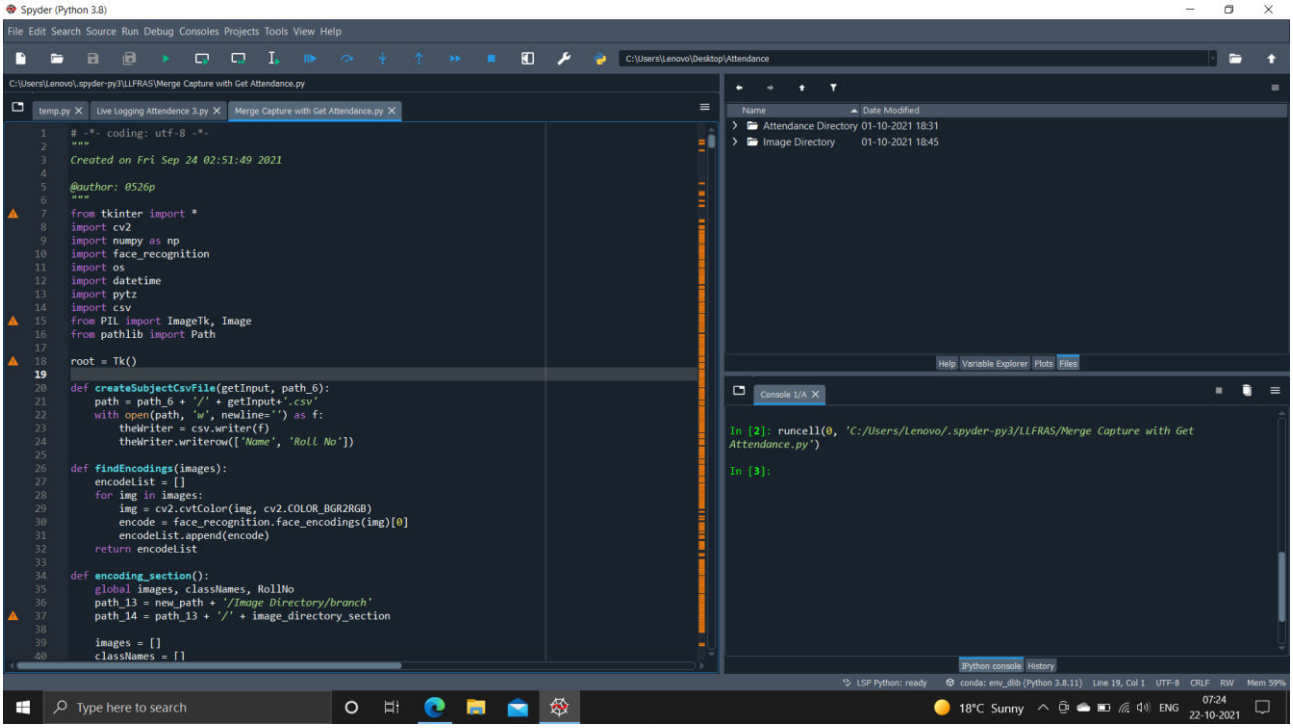
4.4.5 Create and Update CSV sheet

1. Create csv sheet of particular date
2. Update name, roll, time of matched persons
3. Check data sheet if name is already present then it is not overwrite or update again

4.5 Coding of Live Logging Facial Recognition Attendance System

From line 7 to 16 import the libraries which is used in this project. In line 7 import all the functions which is perform by the tkinter library which is used for GUI making interface. It provides the interfaces for user to access whatever information and task he want to perform it.

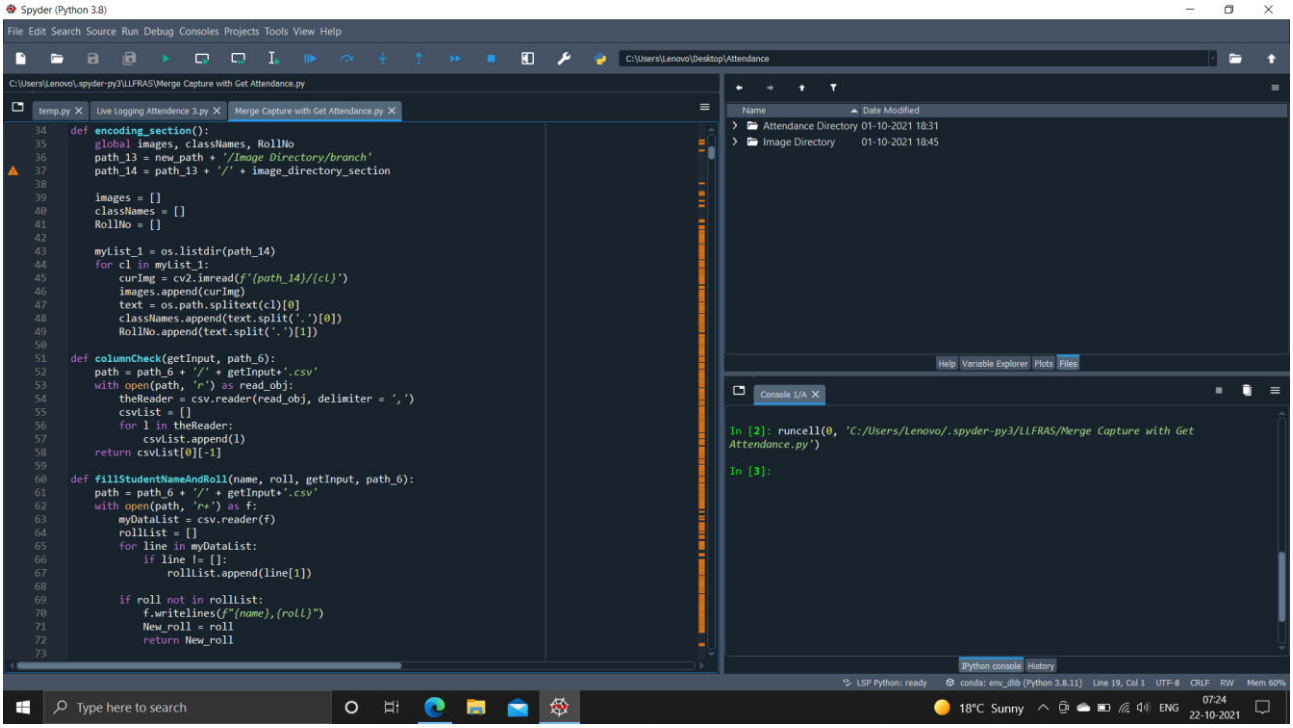
In line 8 import cv2 library for image and video processing. Line 9 is numpy library which is used for store the pixels array of the images captured in video frame. Line 10 is face-recognition library which is used for comparing database images with captured frame images. Line 11 is os library which is used for file handling. Line 12 is datetime library is used for get the actual time and date. Line 13 is pytz library which is used for getting time of a particular timezone. Line 14 is csv libraries which is used for store values seprated by commas. Line 15 is PIL (pillow) library for handling images in Tkinter. Line 16 is pathlib library which is used for getting the path of your os.



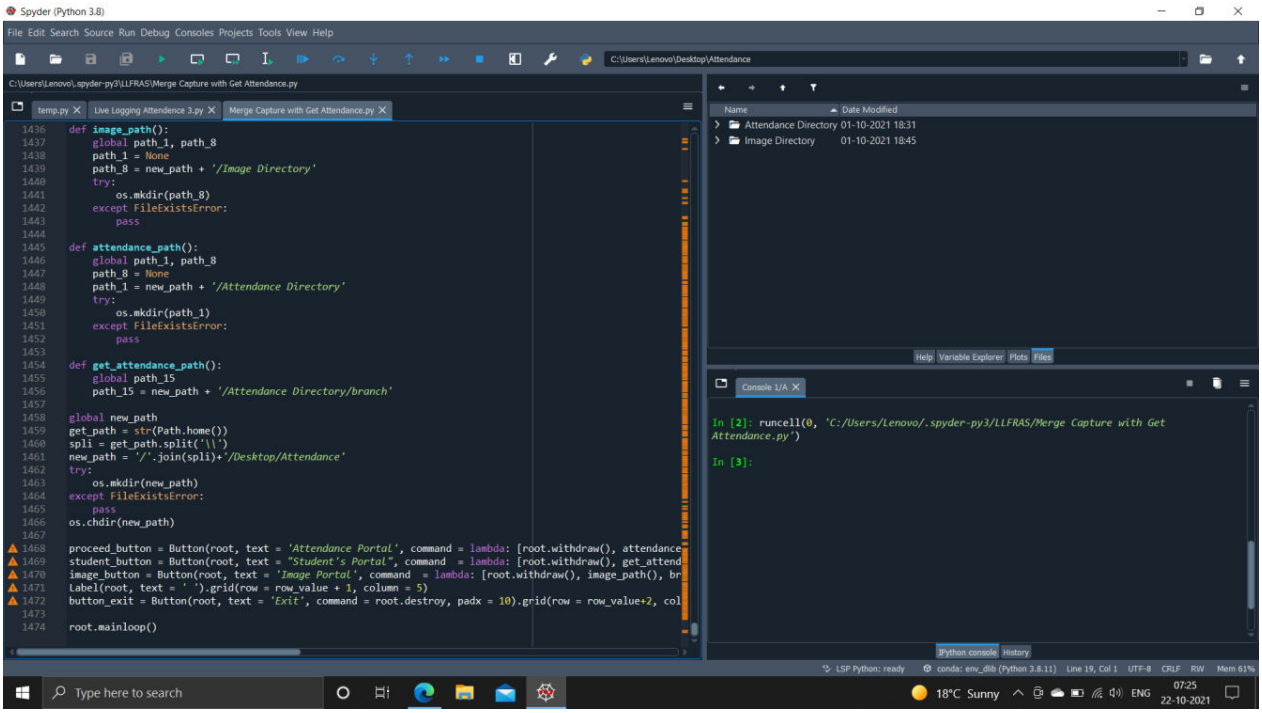
In line 18 we make a root window where we put widgets and icons. In Line 20 to 24 create a function createSubjectCsvFile which makes subjects csv file sheets. In line 26 to 32 we encode the database images using face_encoding and return encodes list of all images.

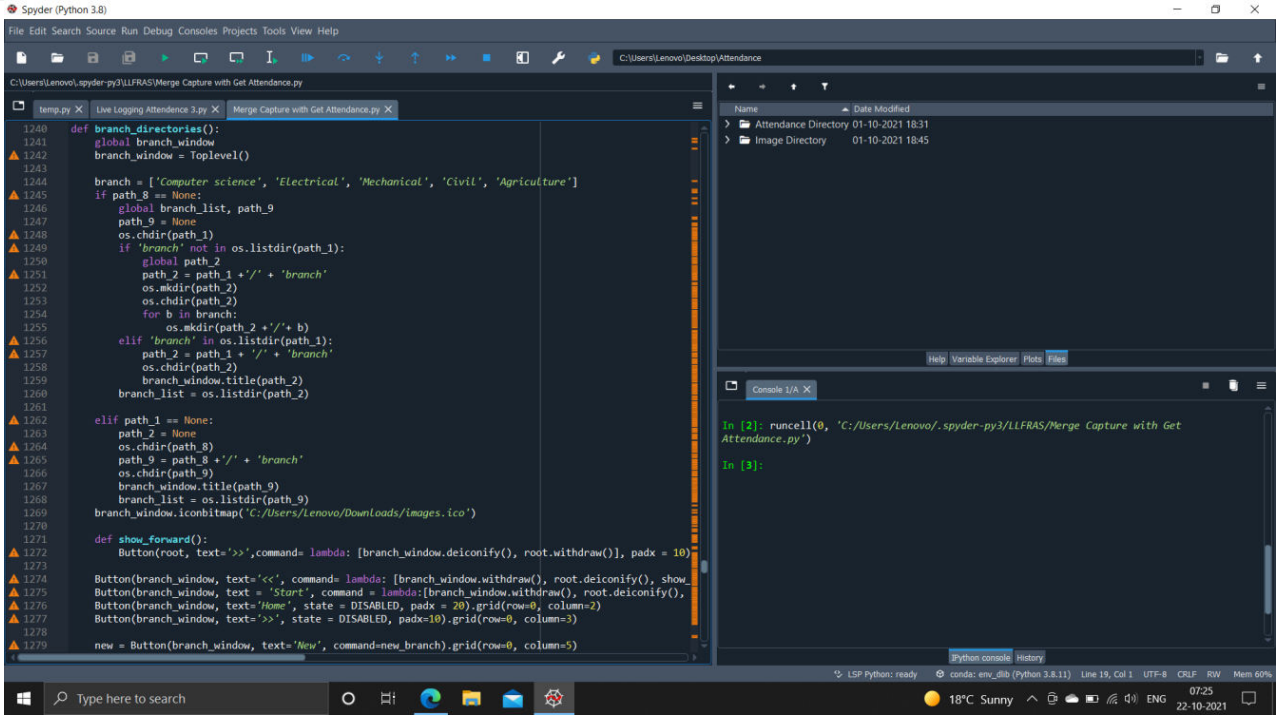
In line 34 make a function where we make a list of images which we get from assigned directory. All image name are preceded with person name and roll no or person id. So, we split it the also we make the list of person name and roll no/person id.

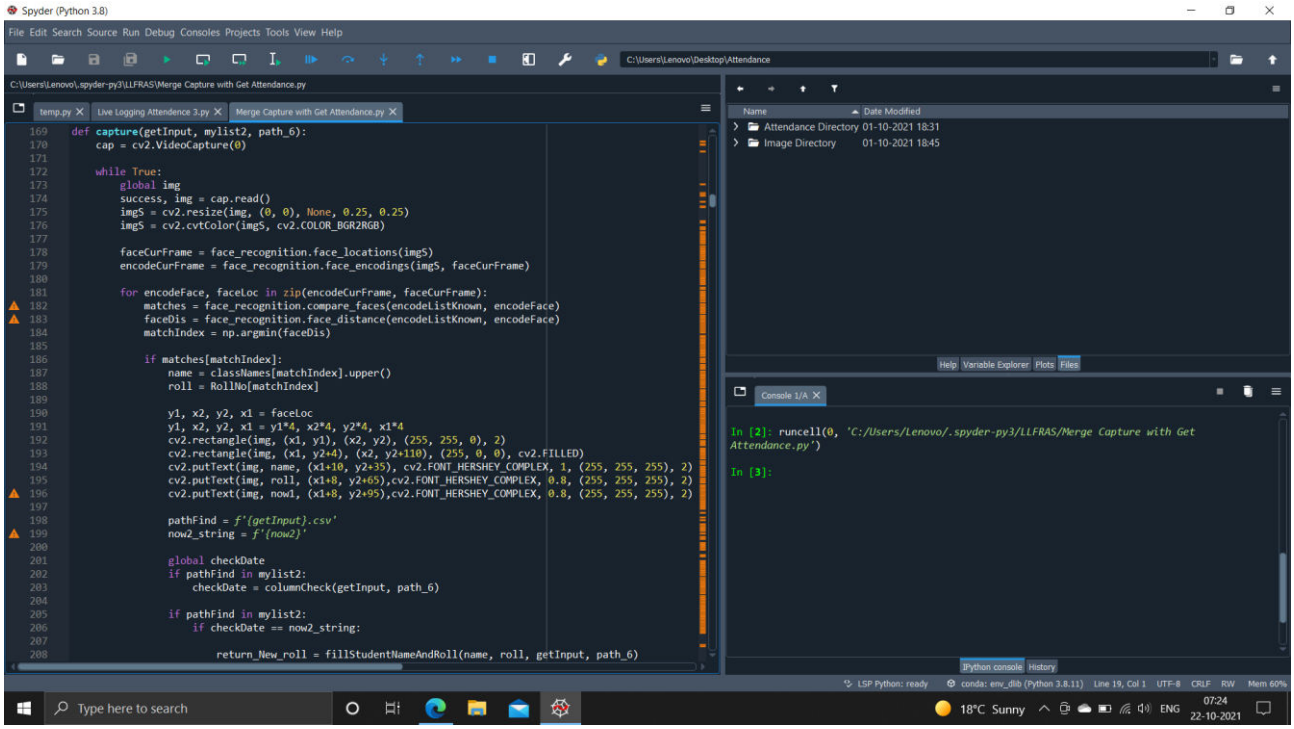
On every particular date a new column is made on that particular date. Before we made new date column of that particular date we need to check first, is there any column of that date is present or not. In line 51, function returns the last date column name which is made recently.

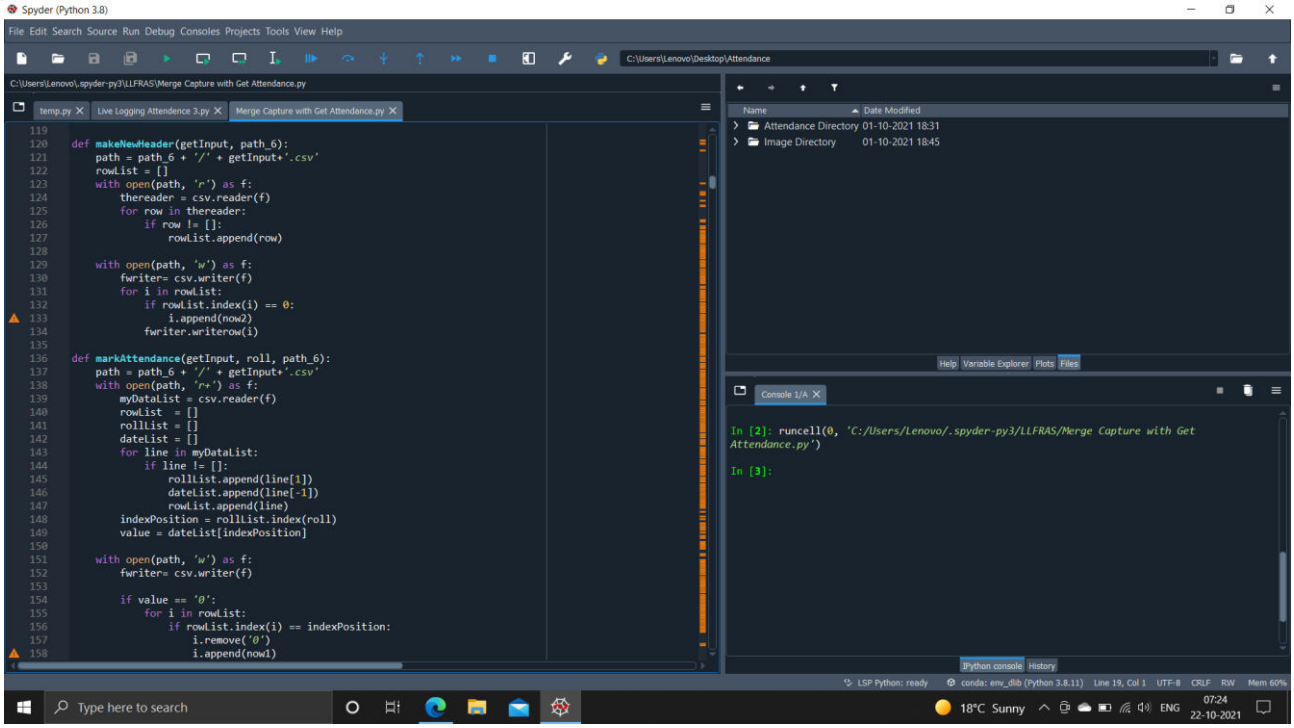


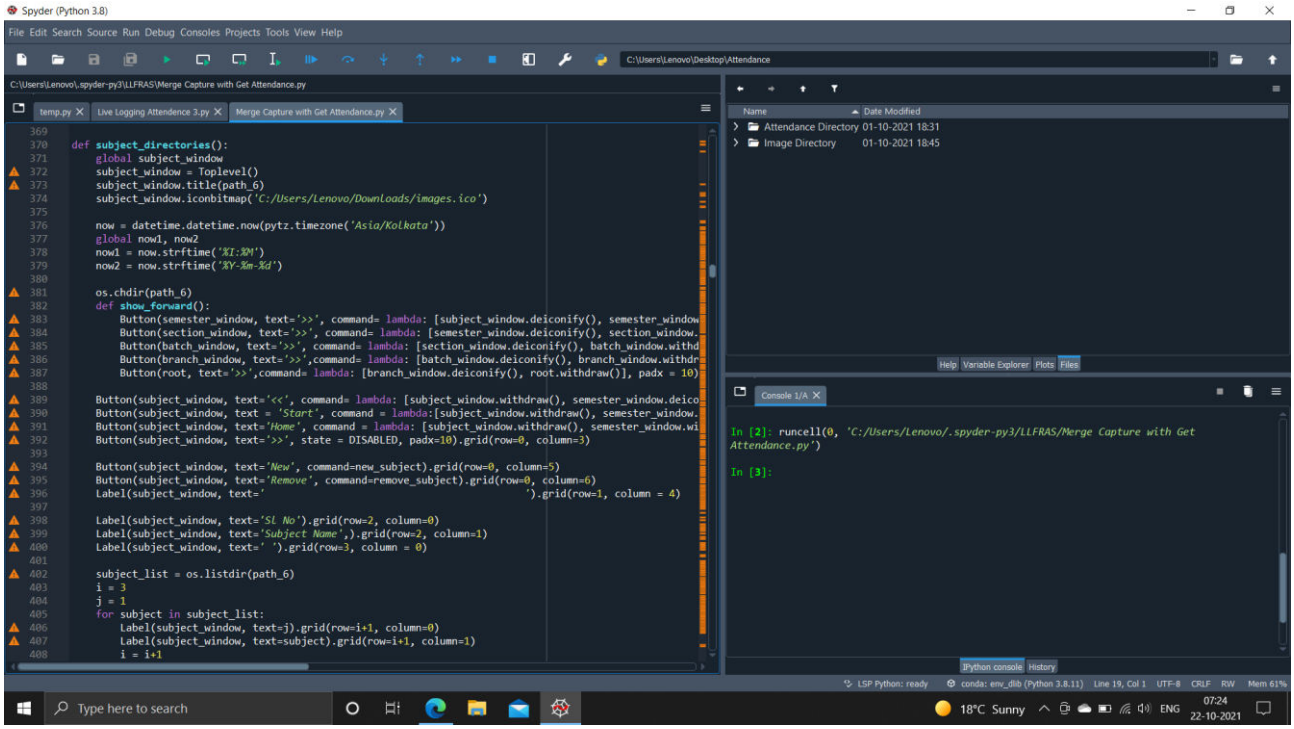
In line 60, fillStudentNameAndRoll function it open the csv sheet in read and write mode using ‘r+’ parameters. We make empty rollList list which stores the roll no of all person name contain in the csv sheet. Then it create a new row of that person name and roll no/person id if it is not present in the csv sheet. If it is already present then pass the function.

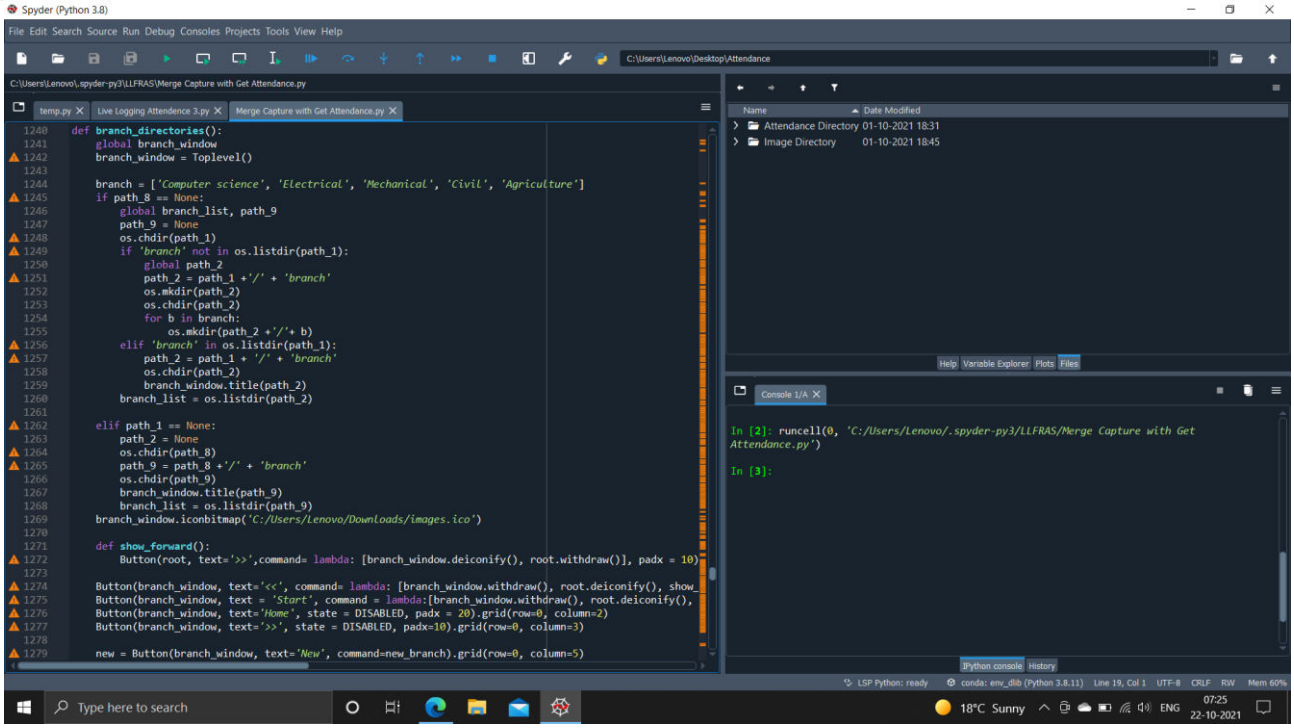


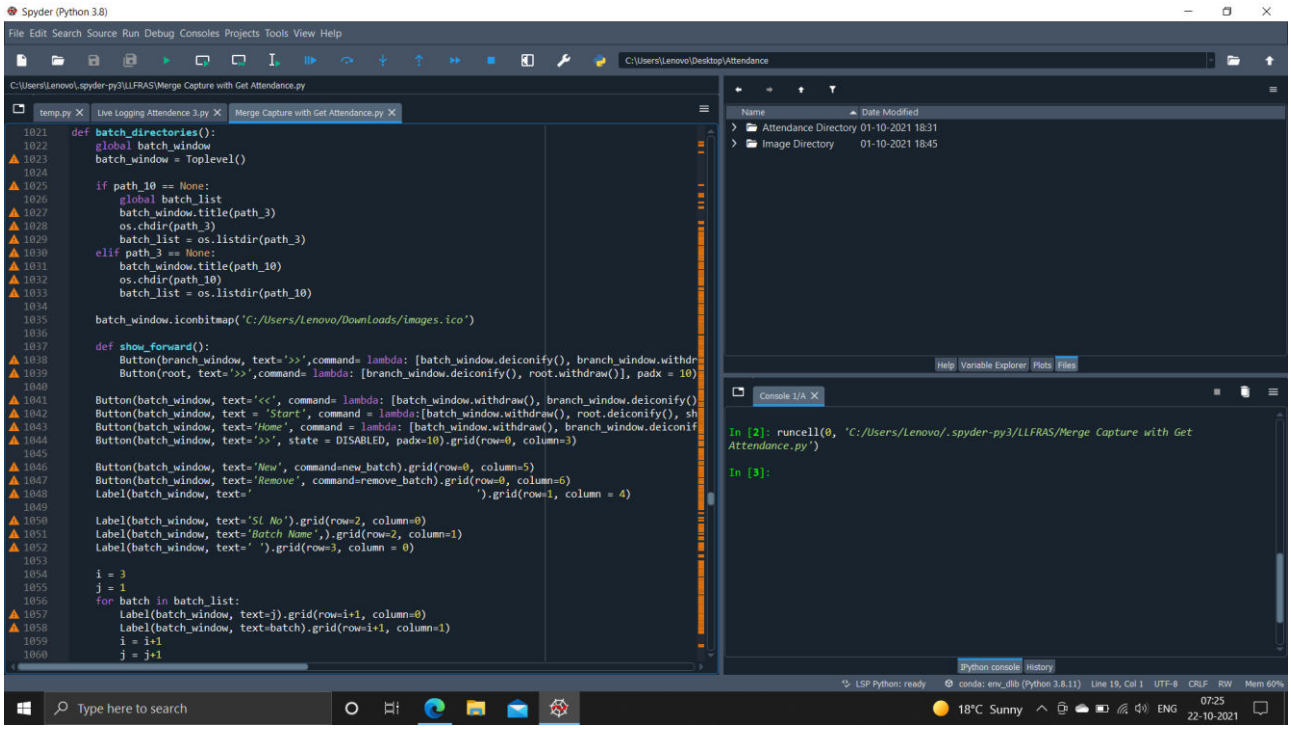

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def branch_directories():
global branch_window
branch_window = Toplevel()
branch = ['Computer science', 'Electrical', 'Mechanical', 'Civil', 'Agriculture']
if path_8 == None:
global branch_list, path_9
path_9 = None
os.chdir(path_1)
if 'branch' not in os.listdir(path_1):
global path_2
path_2 = path_1 +'/' + 'branch'
os.mkdir(path_2)
os.chdir(path_2)
for b in branch:
os.mkdir(path_2 +'/'+ b)
elif 'branch' in os.listdir(path_1):
path_2 = path_1 + '/' + 'branch'
os.chdir(path_2)
branch_window.title(path_2)
branch_list = os.listdir(path_2)
elif path_1 == None:
path_2 = None
os.chdir(path_8)
path_9 = path_8 +'/' + 'branch'
os.chdir(path_9)
branch_window.title(path_9)
branch_list = os.listdir(path_9)
branch_window.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
def show_forward():
Button(root, text='>>',command= lambda: [branch_window.deiconify(), root.withdraw()], padx = 10)
Button(branch_window, text='<<', command= lambda: [branch_window.withdraw(), root.deiconify(), show_
Button(branch_window, text = 'Start', command = lambda:[branch_window.withdraw(), root.deiconify(),
Button(branch_window, text='Home', state = DISABLED, padx = 20).grid(row=0, column=2)
Button(branch_window, text='>>', state = DISABLED, padx=10).grid(row=0, column=3)
new = Button(branch_window, text='New', command=new_branch).grid(row=0, column=5)
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 60%
Type here to search
18°C Sunny
ENG
07:25
22-10-2021

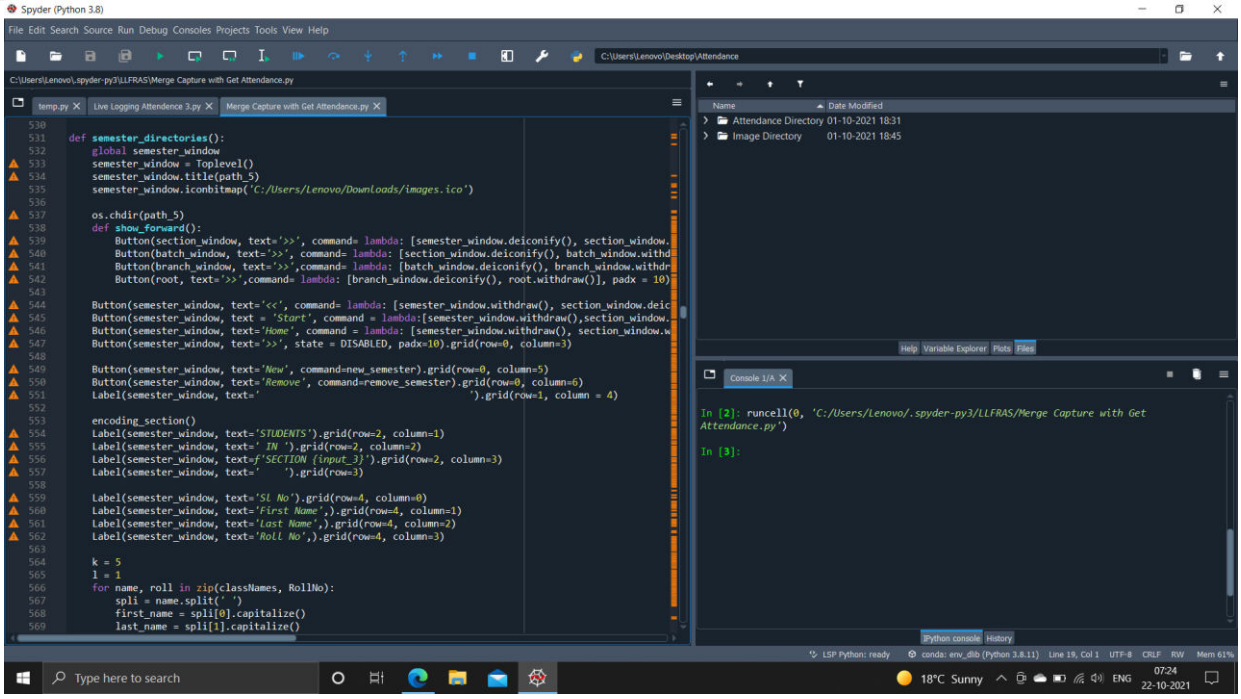

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def capture(getInput, mylist2, path_6):
cap = cv2.VideoCapture(0)
while True:
global img
success, img = cap.read()
imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)
faceCurFrame = face_recognition.face_locations(imgS)
encodeCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)
for encodeFace, faceLoc in zip(encodeCurFrame, faceCurFrame):
matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
matchIndex = np.argmin(faceDis)
if matches[matchIndex]:
name = classNames[matchIndex].upper()
roll = RollNo[matchIndex]
y1, x2, y2, x1 = faceLoc
y1, x2, y2, x1 = y1*4, x2*4, y2*4, x1*4
cv2.rectangle(img, (x1, y1), (x2, y2), (255, 255, 0), 2)
cv2.rectangle(img, (x1, y2+4), (x2, y2+110), (255, 0, 0), cv2.FILLED)
cv2.putText(img, name, (x1+10, y2+35), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
cv2.putText(img, roll, (x1+8, y2+65),cv2.FONT_HERSHEY_COMPLEX, 0.8, (255, 255, 255), 2)
cv2.putText(img, now1, (x1+8, y2+95),cv2.FONT_HERSHEY_COMPLEX, 0.8, (255, 255, 255), 2)
pathFind = f'{getInput}.csv'
now2_string = f'{now2}'
global checkDate
if pathFind in mylist2:
checkDate = columnCheck(getInput, path_6)
if pathFind in mylist2:
if checkDate == now2_string:
return_New_roll = fillStudentNameAndRoll(name, roll, getInput, path_6)
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 60%
Type here to search
18°C Sunny
ENG
07:24
22-10-2021

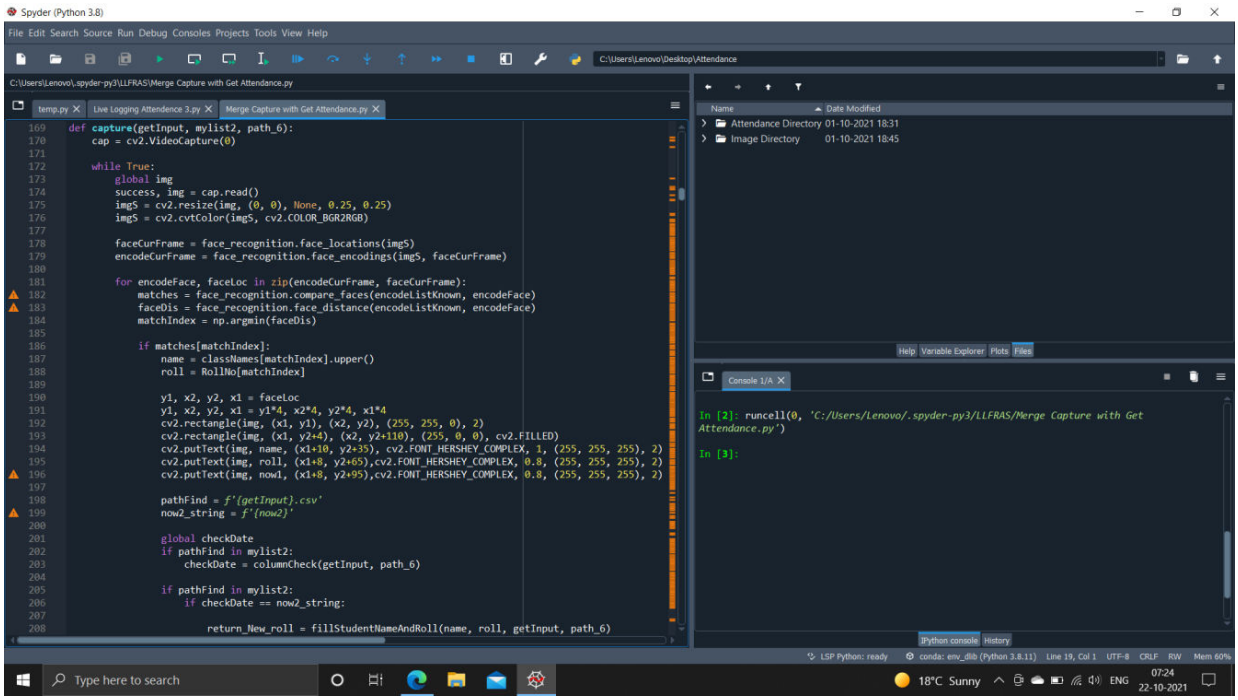

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def makeNewHeader(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
rowList = []
with open(path, 'r') as f:
thereader = csv.reader(f)
for row in thereader:
if row != []:
rowList.append(row)
with open(path, 'w') as f:
fwriter= csv.writer(f)
for i in rowList:
if rowList.index(i) == 0:
i.append(now2)
fwriter.writerow(i)
def markAttendance(getInput, roll, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'r+') as f:
myDataList = csv.reader(f)
rowList = []
rollList = []
dateList = []
for line in myDataList:
if line != []:
rollList.append(line[1])
dateList.append(line[-1])
rowList.append(line)
indexPosition = rollList.index(roll)
value = dateList[indexPosition]
with open(path, 'w') as f:
fwriter= csv.writer(f)
if value == '0':
for i in rowList:
if rowList.index(i) == indexPosition:
i.remove('0')
i.append(now1)
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 60%
Type here to search
18°C Sunny
ENG
07:24
22-10-2021

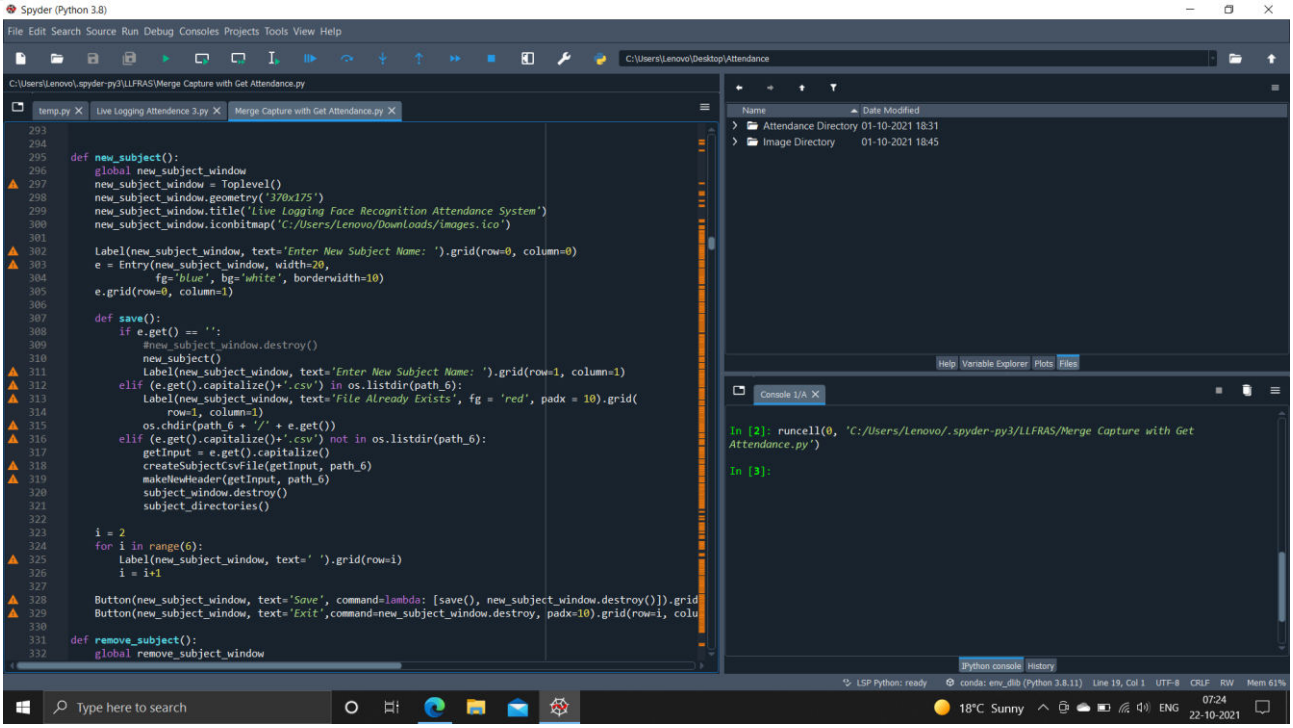

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def subject_directories():
global subject_window
subject_window = Toplevel()
subject_window.title(path_6)
subject_window.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
now = datetime.datetime.now(pytz.timezone('Asia/Kolkata'))
global now1, now2
now1 = now.strftime('%I:%M')
now2 = now.strftime('%Y-%m-%d')
os.chdir(path_6)
def show_forward():
Button(semester_window, text='>>', command= lambda: [subject_window.deiconify(), semester_window
Button(section_window, text='>>', command= lambda: [semester_window.deiconify(), section_window.
Button(batch_window, text='>>', command= lambda: [section_window.deiconify(), batch_window.withd
Button(branch_window, text='>>',command= lambda: [batch_window.deiconify(), branch_window.withdr
Button(root, text='>>',command= lambda: [branch_window.deiconify(), root.withdraw()], padx = 10)
Button(subject_window, text='<<', command= lambda: [subject_window.withdraw(), semester_window.deico
Button(subject_window, text = 'Start', command = lambda:[subject_window.withdraw(), semester_window.
Button(subject_window, text='Home', command = lambda: [subject_window.withdraw(), semester_window.wi
Button(subject_window, text='>>', state = DISABLED, padx=10).grid(row=0, column=3)
Button(subject_window, text='New', command=new_subject).grid(row=0, column=5)
Button(subject_window, text='Remove', command=remove_subject).grid(row=0, column=6)
Label(subject_window, text=' ').grid(row=1, column = 4)
Label(subject_window, text='Sl No').grid(row=2, column=0)
Label(subject_window, text='Subject Name',).grid(row=2, column=1)
Label(subject_window, text=' ').grid(row=3, column = 0)
subject_list = os.listdir(path_6)
i = 3
j = 1
for subject in subject_list:
Label(subject_window, text=j).grid(row=i+1, column=0)
Label(subject_window, text=subject).grid(row=i+1, column=1)
i = i+1
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 61%
Type here to search
18°C Sunny
ENG
07:24
22-10-2021

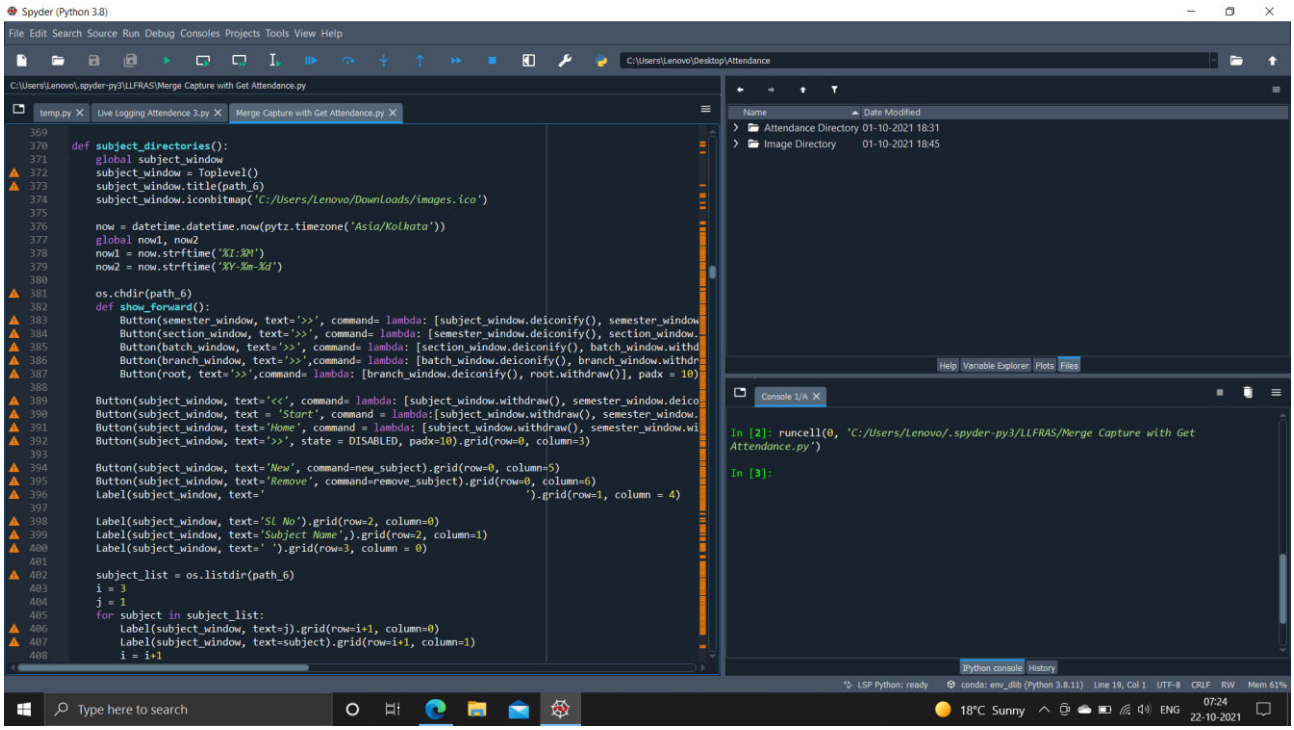

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def branch_directories():
global branch_window
branch_window = Toplevel()
branch = ['Computer science', 'Electrical', 'Mechanical', 'Civil', 'Agriculture']
if path_8 == None:
global branch_list, path_9
path_9 = None
os.chdir(path_1)
if 'branch' not in os.listdir(path_1):
global path_2
path_2 = path_1 +'/' + 'branch'
os.mkdir(path_2)
os.chdir(path_2)
for b in branch:
os.mkdir(path_2 +'/'+ b)
elif 'branch' in os.listdir(path_1):
path_2 = path_1 + '/' + 'branch'
os.chdir(path_2)
branch_window.title(path_2)
branch_list = os.listdir(path_2)
elif path_1 == None:
path_2 = None
os.chdir(path_8)
path_9 = path_8 +'/' + 'branch'
os.chdir(path_9)
branch_window.title(path_9)
branch_list = os.listdir(path_9)
branch_window.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
def show_forward():
Button(root, text='>>',command= lambda: [branch_window.deiconify(), root.withdraw()], padx = 10)
Button(branch_window, text='<<', command= lambda: [branch_window.withdraw(), root.deiconify(), show_
Button(branch_window, text = 'Start', command = lambda:[branch_window.withdraw(), root.deiconify(),
Button(branch_window, text='Home', state = DISABLED, padx = 20).grid(row=0, column=2)
Button(branch_window, text='>>', state = DISABLED, padx=10).grid(row=0, column=3)
new = Button(branch_window, text='New', command=new_branch).grid(row=0, column=5)
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 60%
Type here to search
18°C Sunny
ENG
07:25
22-10-2021

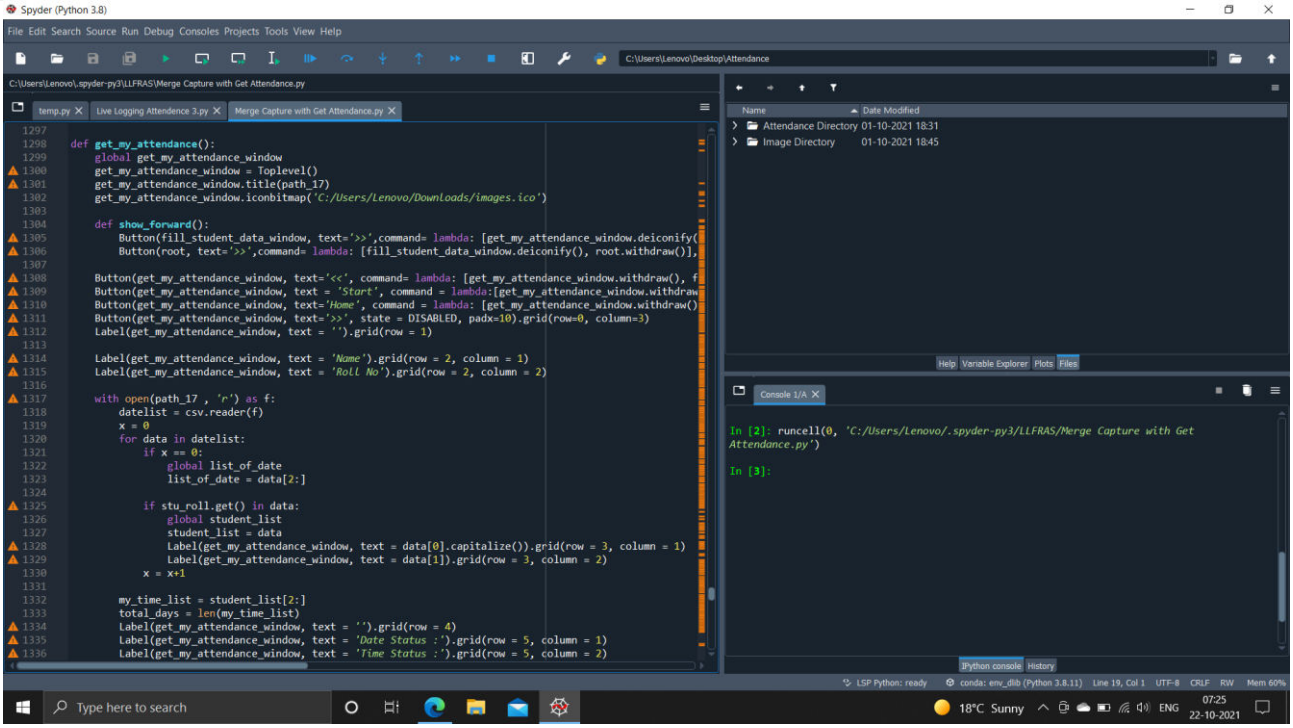

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def batch_directories():
global batch_window
batch_window = Toplevel()
if path_10 == None:
global batch_list
batch_window.title(path_3)
os.chdir(path_3)
batch_list = os.listdir(path_3)
elif path_3 == None:
batch_window.title(path_10)
os.chdir(path_10)
batch_list = os.listdir(path_10)
batch_window.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
def show_forward():
Button(branch_window, text='>>',command= lambda: [batch_window.deiconify(), branch_window.withdr
Button(root, text='>>',command= lambda: [branch_window.deiconify(), root.withdraw()], padx = 10)
Button(batch_window, text='<<', command= lambda: [batch_window.withdraw(), branch_window.deiconify()
Button(batch_window, text = 'Start', command = lambda:[batch_window.withdraw(), root.deiconify(), sh
Button(batch_window, text='Home', command = lambda: [batch_window.withdraw(), branch_window.deiconif
Button(batch_window, text='>>', state = DISABLED, padx=10).grid(row=0, column=3)
Button(batch_window, text='New', command=new_batch).grid(row=0, column=5)
Button(batch_window, text='Remove', command=remove_batch).grid(row=0, column=6)
Label(batch_window, text=' ').grid(row=1, column = 4)
Label(batch_window, text='Sl No').grid(row=2, column=0)
Label(batch_window, text='Batch Name',).grid(row=2, column=1)
Label(batch_window, text=' ').grid(row=3, column = 0)
i = 3
j = 1
for batch in batch_list:
Label(batch_window, text=j).grid(row=i+1, column=0)
Label(batch_window, text=batch).grid(row=i+1, column=1)
i = i+1
j = j+1
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 60%
Type here to search
18°C Sunny
ENG
07:25
22-10-2021

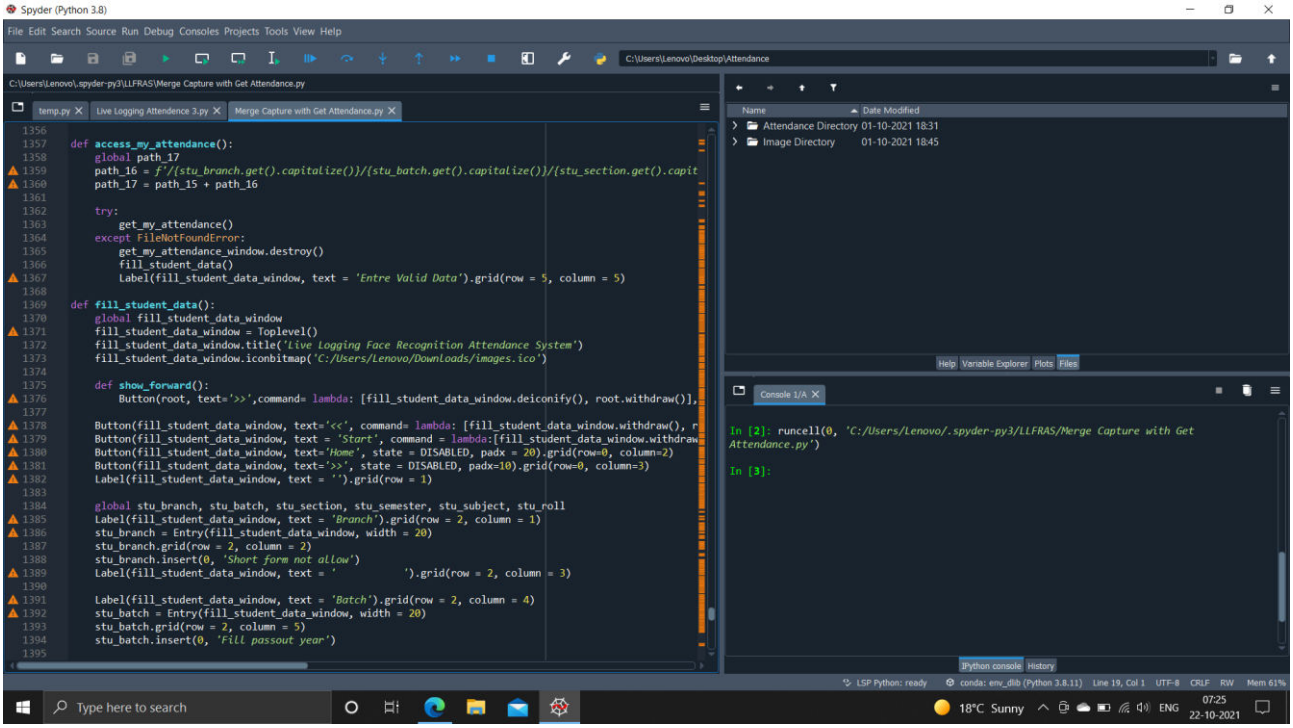

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def semester_directories():
global semester_window
semester_window = Toplevel()
semester_window.title(path_5)
semester_window.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
os.chdir(path_5)
def show_forward():
Button(semester_window, text='>>', command= lambda: [semester_window.deiconify(), section_window.
Button(batch_window, text='>>', command= lambda: [section_window.deiconify(), batch_window.withd
Button(branch_window, text='>>',command= lambda: [batch_window.deiconify(), branch_window.withdr
Button(root, text='>>',command= lambda: [branch_window.deiconify(), root.withdraw()], padx = 10)
Button(semester_window, text='<<', command= lambda: [semester_window.withdraw(), section_window.deic
Button(semester_window, text = 'Start', command = lambda:[semester_window.withdraw(),section_window.
Button(semester_window, text='Home', command = lambda: [semester_window.withdraw(), section_window.w
Button(semester_window, text='>>', state = DISABLED, padx=10).grid(row=0, column=3)
Button(semester_window, text='New', command=new_semester).grid(row=0, column=5)
Button(semester_window, text='Remove', command=remove_semester).grid(row=0, column=6)
Label(semester_window, text='').grid(row=1, column = 4)
encoding_section()
Label(semester_window, text='STUDENTS').grid(row=2, column=1)
Label(semester_window, text=' IN ').grid(row=2, column=2)
Label(semester_window, text=f'SECTION {input_3}').grid(row=2, column=3)
Label(semester_window, text=' ').grid(row=3)
Label(semester_window, text='Sl No').grid(row=4, column=0)
Label(semester_window, text='First Name',).grid(row=4, column=1)
Label(semester_window, text='Last Name',).grid(row=4, column=2)
Label(semester_window, text='Roll No',).grid(row=4, column=3)
k = 5
l = 1
for name, roll in zip(classNames, RollNo):
spli = name.split(' ')
first_name = spli[0].capitalize()
last_name = spli[1].capitalize()
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help Variable Explorer Plots Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console History
LSP Python: ready
conda: env_dlib (Python 3.8.11) Line 19, Col 1 UTF-8 CRLF RW Mem 61%
Type here to search
18°C Sunny
ENG
07:24
22-10-2021

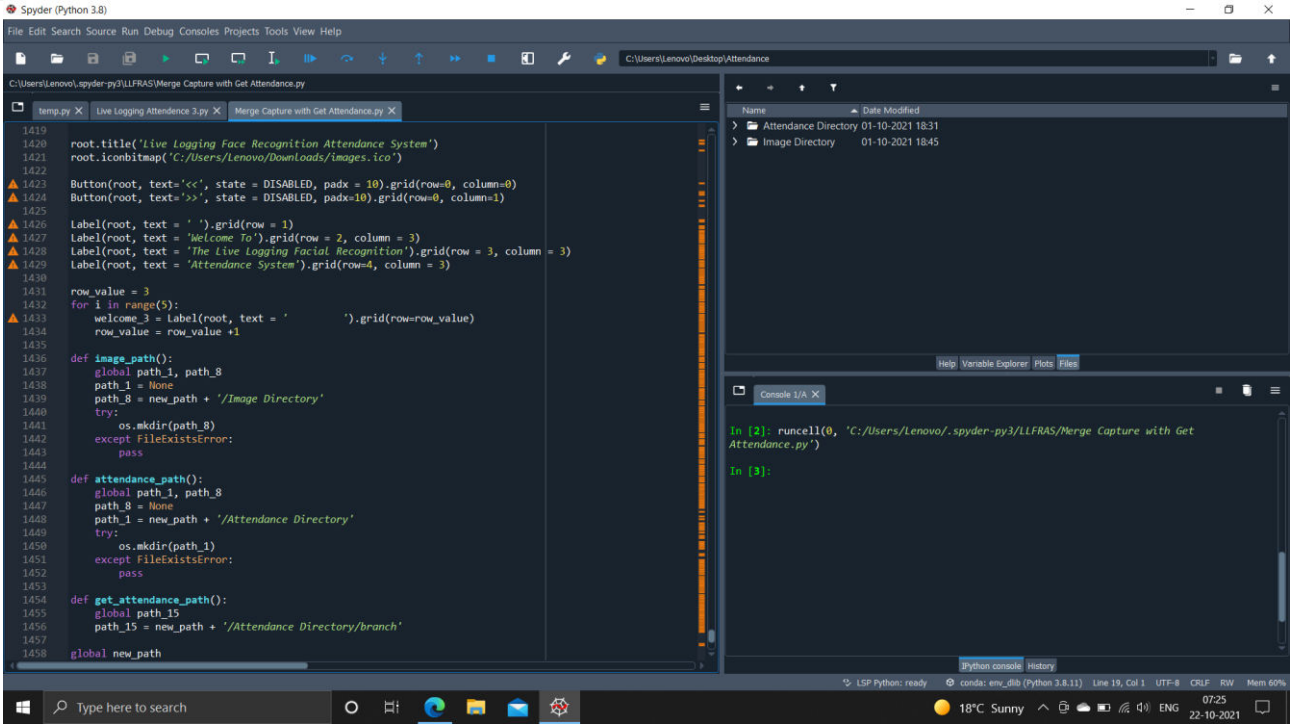

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def capture(getInput, mylist2, path_6):
cap = cv2.VideoCapture(0)
while True:
global img
success, img = cap.read()
imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)
faceCurFrame = face_recognition.face_locations(imgS)
encodeCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)
for encodeFace, faceLoc in zip(encodeCurFrame, faceCurFrame):
matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
matchIndex = np.argmin(faceDis)
if matches[matchIndex]:
name = classNames[matchIndex].upper()
roll = RollNo[matchIndex]
y1, x2, y2, x1 = faceLoc
y1, x2, y2, x1 = y1*4, x2*4, y2*4, x1*4
cv2.rectangle(img, (x1, y1), (x2, y2), (255, 255, 0), 2)
cv2.rectangle(img, (x1, y2+4), (x2, y2+110), (255, 0, 0), cv2.FILLED)
cv2.putText(img, name, (x1+10, y2+35), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
cv2.putText(img, roll, (x1+8, y2+65),cv2.FONT_HERSHEY_COMPLEX, 0.8, (255, 255, 255), 2)
cv2.putText(img, now1, (x1+8, y2+95),cv2.FONT_HERSHEY_COMPLEX, 0.8, (255, 255, 255), 2)
pathFind = f'{getInput}.csv'
now2_string = f'{now2}'
global checkDate
if pathFind in mylist2:
checkDate = columnCheck(getInput, path_6)
if pathFind in mylist2:
if checkDate == now2_string:
return_New_roll = fillStudentNameAndRoll(name, roll, getInput, path_6)
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help Variable Explorer Plots Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console History
LSP Python: ready
conda: env_dlib (Python 3.8.11) Line 19, Col 1 UTF-8 CRLF RW Mem 60%
Type here to search
18°C Sunny
ENG
07:24
22-10-2021

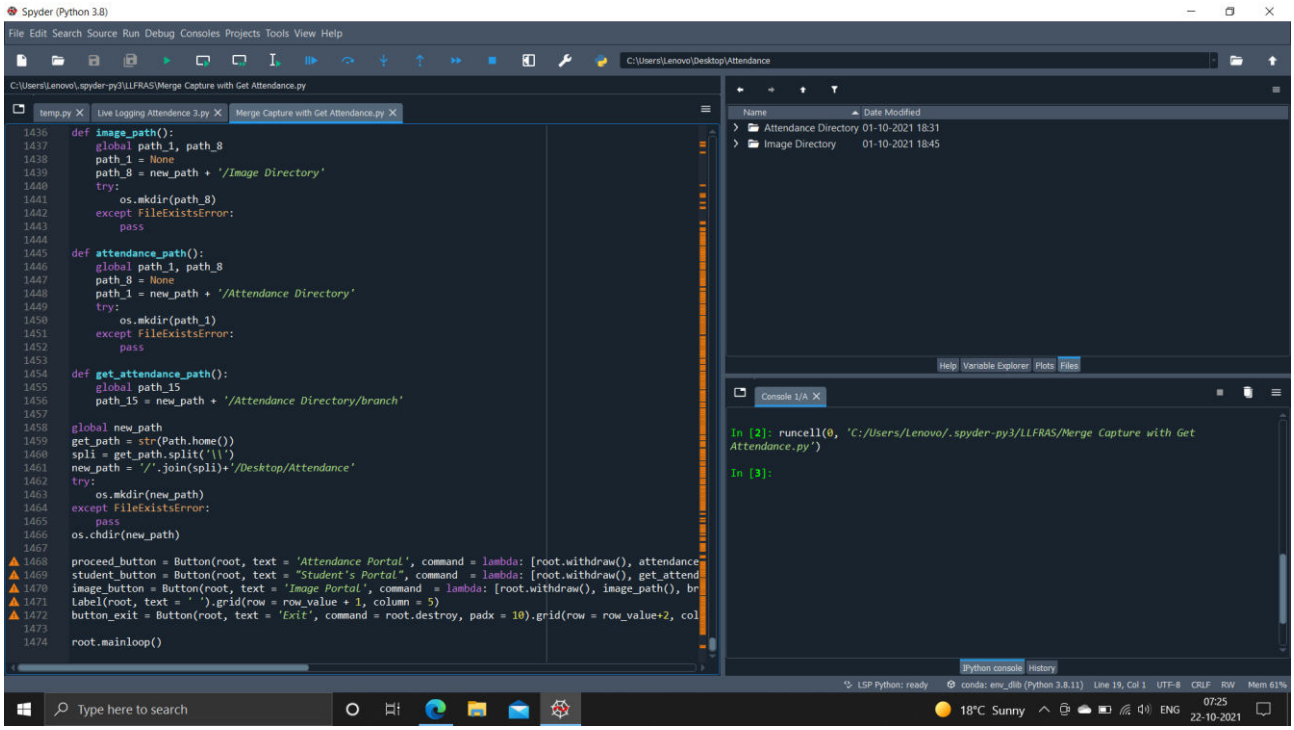

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def new_subject():
global new_subject_window
new_subject_window = Toplevel()
new_subject_window.geometry('370x175')
new_subject_window.title('Live Logging Face Recognition Attendance System')
new_subject_window.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
Label(new_subject_window, text='Enter New Subject Name: ').grid(row=0, column=0)
e = Entry(new_subject_window, width=20,
fg='blue', bg='white', borderwidth=10)
e.grid(row=0, column=1)
def save():
if e.get() == '':
#new_subject_window.destroy()
new_subject()
Label(new_subject_window, text='Enter New Subject Name: ').grid(row=1, column=1)
elif (e.get().capitalize()+'.csv') in os.listdir(path_6):
Label(new_subject_window, text='File Already Exists', fg = 'red', padx = 10).grid(
row=1, column=1)
os.chdir(path_6 + '/' + e.get())
elif (e.get().capitalize()+'.csv') not in os.listdir(path_6):
getInput = e.get().capitalize()
createSubjectCsvFile(getInput, path_6)
makeNewHeader(getInput, path_6)
subject_window.destroy()
subject_directories()
i = 2
for i in range(6):
Label(new_subject_window, text=' ').grid(row=i)
i = i+1
Button(new_subject_window, text='Save', command=lambda: [save(), new_subject_window.destroy()]).grid
Button(new_subject_window, text='Exit',command=new_subject_window.destroy, padx=10).grid(row=i, colu
def remove_subject():
global remove_subject_window
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 61%
Type here to search
18°C Sunny
ENG
07:24
22-10-2021


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def subject_directories():
global subject_window
subject_window = Toplevel()
subject_window.title(path_6)
subject_window.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
now = datetime.datetime.now(pytz.timezone('Asia/Kolkata'))
global now1, now2
now1 = now.strftime('%I:%M')
now2 = now.strftime('%Y-%m-%d')
os.chdir(path_6)
def show_forward():
Button(semester_window, text='>>', command= lambda: [subject_window.deiconify(), semester_window
Button(section_window, text='>>', command= lambda: [semester_window.deiconify(), section_window.
Button(batch_window, text='>>', command= lambda: [section_window.deiconify(), batch_window.withd
Button(branch_window, text='>>',command= lambda: [batch_window.deiconify(), branch_window.withdr
Button(root, text='>>',command= lambda: [branch_window.deiconify(), root.withdraw()], padx = 10)
Button(subject_window, text='<<', command= lambda: [subject_window.withdraw(), semester_window.deico
Button(subject_window, text = 'Start', command = lambda:[subject_window.withdraw(), semester_window.
Button(subject_window, text='Home', command = lambda: [subject_window.withdraw(), semester_window.wi
Button(subject_window, text='>>', state = DISABLED, padx=10).grid(row=0, column=3)
Button(subject_window, text='New', command=new_subject).grid(row=0, column=5)
Button(subject_window, text='Remove', command=remove_subject).grid(row=0, column=6)
Label(subject_window, text=' ').grid(row=1, column = 4)
Label(subject_window, text='Sl No').grid(row=2, column=0)
Label(subject_window, text='Subject Name',).grid(row=2, column=1)
Label(subject_window, text=' ').grid(row=3, column = 0)
subject_list = os.listdir(path_6)
i = 3
j = 1
for subject in subject_list:
Label(subject_window, text=j).grid(row=i+1, column=0)
Label(subject_window, text=subject).grid(row=i+1, column=1)
i = i+1
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 61%
Type here to search
18°C Sunny
ENG
07:24
22-10-2021


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def get_my_attendance():
global get_my_attendance_window
get_my_attendance_window = Toplevel()
get_my_attendance_window.title(path_17)
get_my_attendance_window.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
def show_forward():
Button(fill_student_data_window, text='>>',command= lambda: [get_my_attendance_window.deiconify(
Button(root, text='>>',command= lambda: [fill_student_data_window.deiconify(), root.withdraw()],
Button(get_my_attendance_window, text='<<', command= lambda: [get_my_attendance_window.withdraw(), f
Button(get_my_attendance_window, text = 'Start', command = lambda:[get_my_attendance_window.withdraw
Button(get_my_attendance_window, text='Home', command = lambda: [get_my_attendance_window.withdraw()
Button(get_my_attendance_window, text='>>', state = DISABLED, padx=10).grid(row=0, column=3)
Label(get_my_attendance_window, text = '').grid(row = 1)
Label(get_my_attendance_window, text = 'Name').grid(row = 2, column = 1)
Label(get_my_attendance_window, text = 'Roll No').grid(row = 2, column = 2)
with open(path_17 , 'r') as f:
datelist = csv.reader(f)
x = 0
for data in datelist:
if x == 0:
global list_of_date
list_of_date = data[2:]
if stu_roll.get() in data:
global student_list
student_list = data
Label(get_my_attendance_window, text = data[0].capitalize()).grid(row = 3, column = 1)
Label(get_my_attendance_window, text = data[1]).grid(row = 3, column = 2)
x = x+1
my_time_list = student_list[2:]
total_days = len(my_time_list)
Label(get_my_attendance_window, text = '').grid(row = 4)
Label(get_my_attendance_window, text = 'Date Status :').grid(row = 5, column = 1)
Label(get_my_attendance_window, text = 'Time Status :').grid(row = 5, column = 2)
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 60%
Type here to search
18°C Sunny
ENG
07:25
22-10-2021

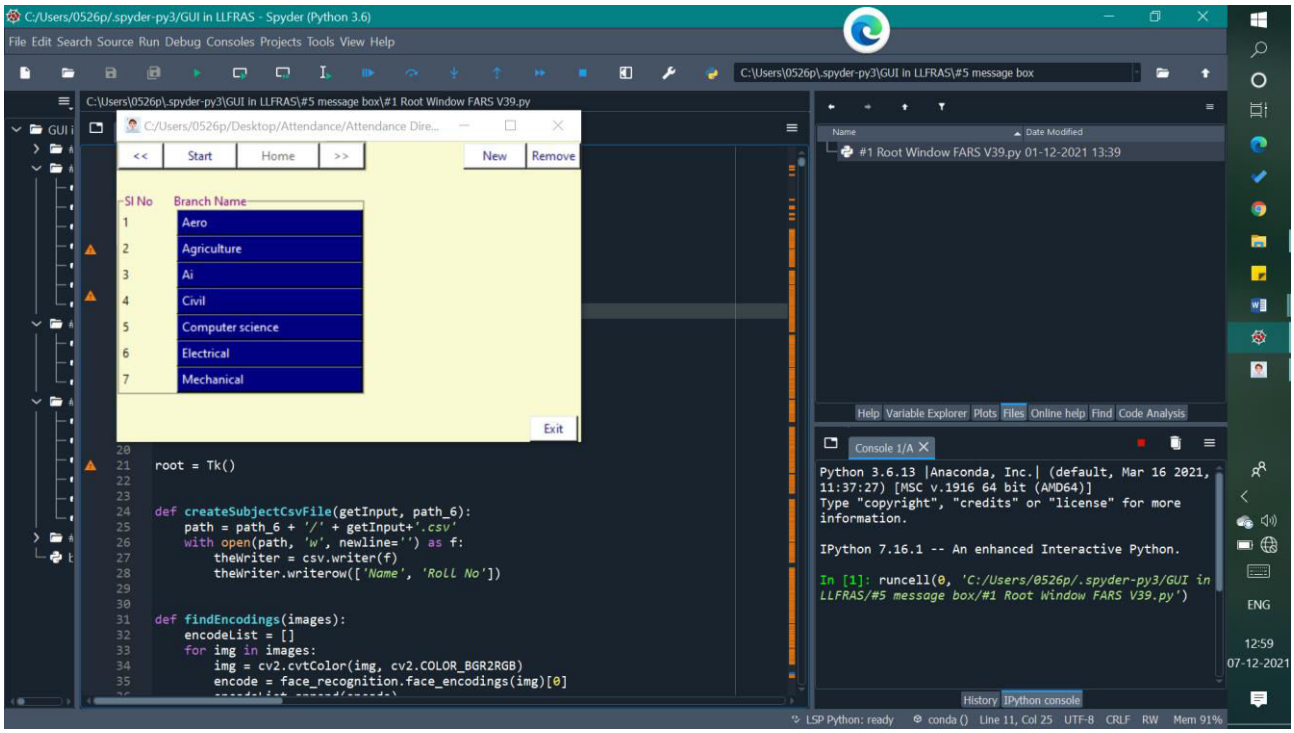

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def access_my_attendance():
global path_17
path_16 = f'/{stu_branch.get().capitalize()}/{stu_batch.get().capitalize()}/{stu_section.get().capit
path_17 = path_15 + path_16
try:
get_my_attendance()
except FileNotFoundError:
get_my_attendance_window.destroy()
fill_student_data()
Label(fill_student_data_window, text = 'Entre Valid Data').grid(row = 5, column = 5)
def fill_student_data():
global fill_student_data_window
fill_student_data_window = Toplevel()
fill_student_data_window.title('Live Logging Face Recognition Attendance System')
fill_student_data_window.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
def show_forward():
Button(root, text='>>',command= lambda: [fill_student_data_window.deiconify(), root.withdraw()],
Button(fill_student_data_window, text='<<', command= lambda: [fill_student_data_window.withdraw(),
Button(fill_student_data_window, text = 'Start', command = lambda:[fill_student_data_window.withdraw
Button(fill_student_data_window, text='Home', state = DISABLED, padx = 20).grid(row=0, column=2)
Button(fill_student_data_window, text='>>', state = DISABLED, padx=10).grid(row=0, column=3)
Label(fill_student_data_window, text = '').grid(row = 1)
global stu_branch, stu_batch, stu_section, stu_semester, stu_subject, stu_roll
Label(fill_student_data_window, text = 'Branch').grid(row = 2, column = 1)
stu_branch = Entry(fill_student_data_window, width = 20)
stu_branch.grid(row = 2, column = 2)
stu_branch.insert(0, 'Short form not allow')
Label(fill_student_data_window, text = ' ').grid(row = 2, column = 3)
Label(fill_student_data_window, text = 'Batch').grid(row = 2, column = 4)
stu_batch = Entry(fill_student_data_window, width = 20)
stu_batch.grid(row = 2, column = 5)
stu_batch.insert(0, 'Fill passout year')
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 61%
Type here to search
18°C Sunny
ENG
07:25
22-10-2021

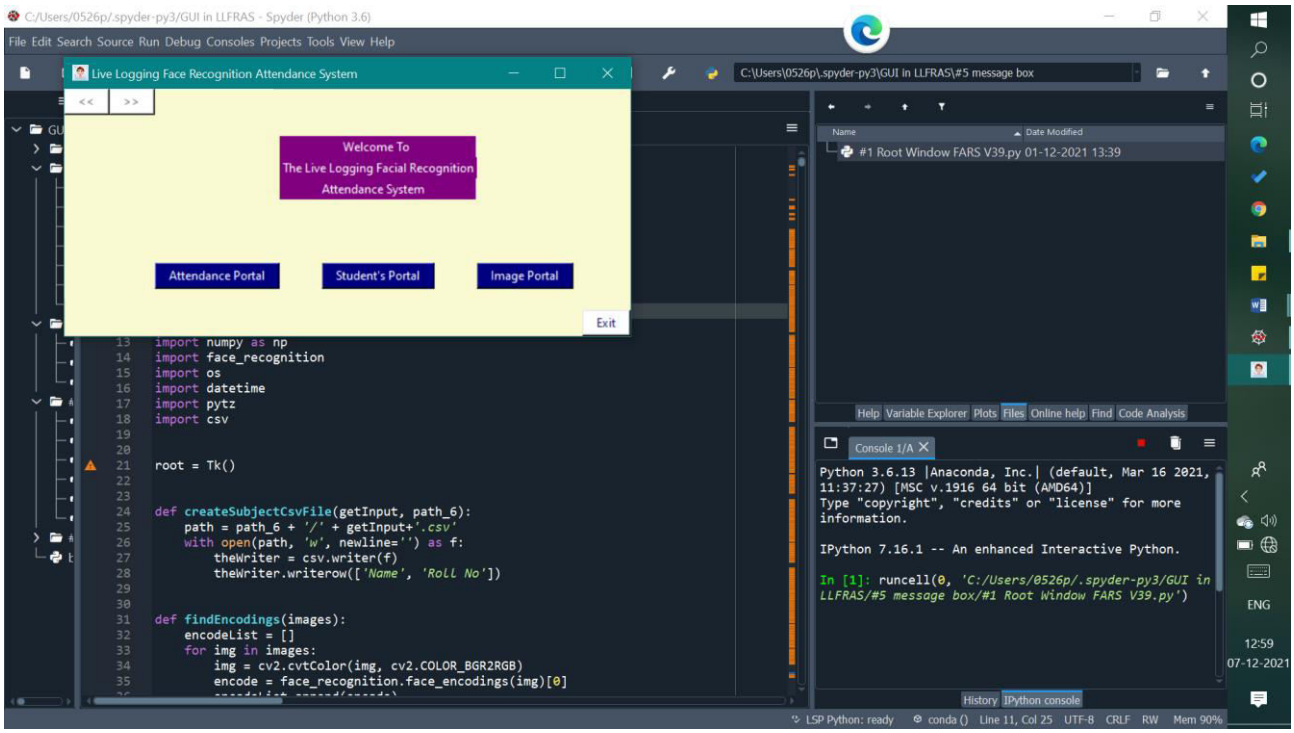

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
root.title('Live Logging Face Recognition Attendance System')
root.iconbitmap('C:/Users/Lenovo/Downloads/images.ico')
Button(root, text='<<', state = DISABLED, padx = 10).grid(row=0, column=0)
Button(root, text='>>', state = DISABLED, padx=10).grid(row=0, column=1)
Label(root, text = ' ').grid(row = 1)
Label(root, text = 'Welcome To').grid(row = 2, column = 3)
Label(root, text = 'The Live Logging Facial Recognition').grid(row = 3, column = 3)
Label(root, text = 'Attendance System').grid(row=4, column = 3)
row_value = 3
for i in range(5):
welcome_3 = Label(root, text = ' ').grid(row=row_value)
row_value = row_value +1
def image_path():
global path_1, path_8
path_1 = None
path_8 = new_path + '/Image Directory'
try:
os.mkdir(path_8)
except FileExistsError:
pass
def attendance_path():
global path_1, path_8
path_8 = None
path_1 = new_path + '/Attendance Directory'
try:
os.mkdir(path_1)
except FileExistsError:
pass
def get_attendance_path():
global path_15
path_15 = new_path + '/Attendance Directory/branch'
global new_path
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 60%
Type here to search
18°C Sunny
ENG
07:25
22-10-2021

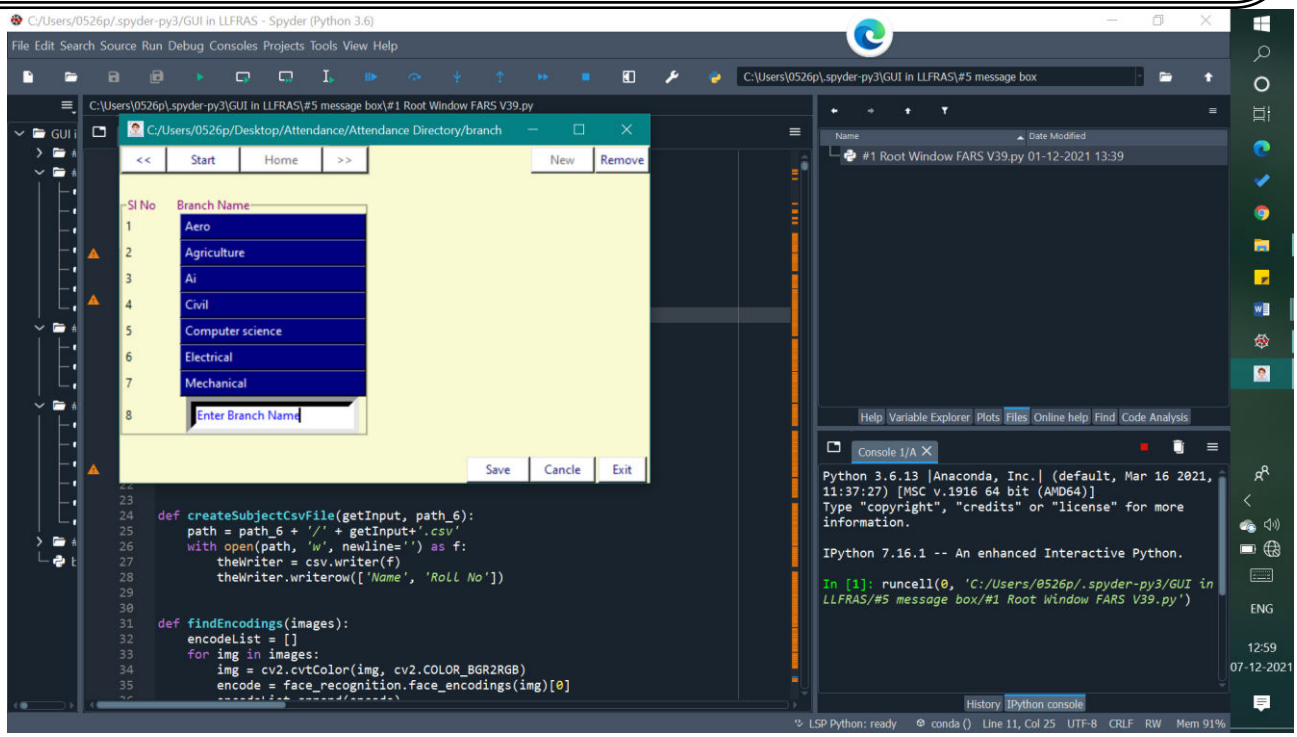

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Lenovo\Desktop\Attendance
C:\Users\Lenovo\.spyder-py3\LLFRAS\Merge Capture with Get Attendance.py
temp.py
Live Logging Attendence 3.py
Merge Capture with Get Attendance.py
def image_path():
global path_1, path_8
path_1 = None
path_8 = new_path + '/Image Directory'
try:
os.mkdir(path_8)
except FileExistsError:
pass
def attendance_path():
global path_1, path_8
path_8 = None
path_1 = new_path + '/Attendance Directory'
try:
os.mkdir(path_1)
except FileExistsError:
pass
def get_attendance_path():
global path_15
path_15 = new_path + '/Attendance Directory/branch'
global new_path
get_path = str(Path.home())
spli = get_path.split('\\')
new_path = '/'.join(spli)+'/Desktop/Attendance'
try:
os.mkdir(new_path)
except FileExistsError:
pass
os.chdir(new_path)
proceed_button = Button(root, text = 'Attendance Portal', command = lambda: [root.withdraw(), attendance
student_button = Button(root, text = "Student's Portal", command = lambda: [root.withdraw(), get_attend
image_button = Button(root, text = 'Image Portal', command = lambda: [root.withdraw(), image_path(), br
Label(root, text = ' ').grid(row = row_value + 1, column = 5)
button_exit = Button(root, text = 'Exit', command = root.destroy, padx = 10).grid(row = row_value+2, col
root.mainloop()
Name
Date Modified
Attendance Directory 01-10-2021 18:31
Image Directory 01-10-2021 18:45
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: runcell(0, 'C:/Users/Lenovo/.spyder-py3/LLFRAS/Merge Capture with Get Attendance.py')
In [3]:
IPython console
History
LSP Python: ready
conda: env_dlib (Python 3.8.11)
Line 19, Col 1
UTF-8
CRLF
RW
Mem 61%
Type here to search
18°C Sunny
ENG
07:25
22-10-2021

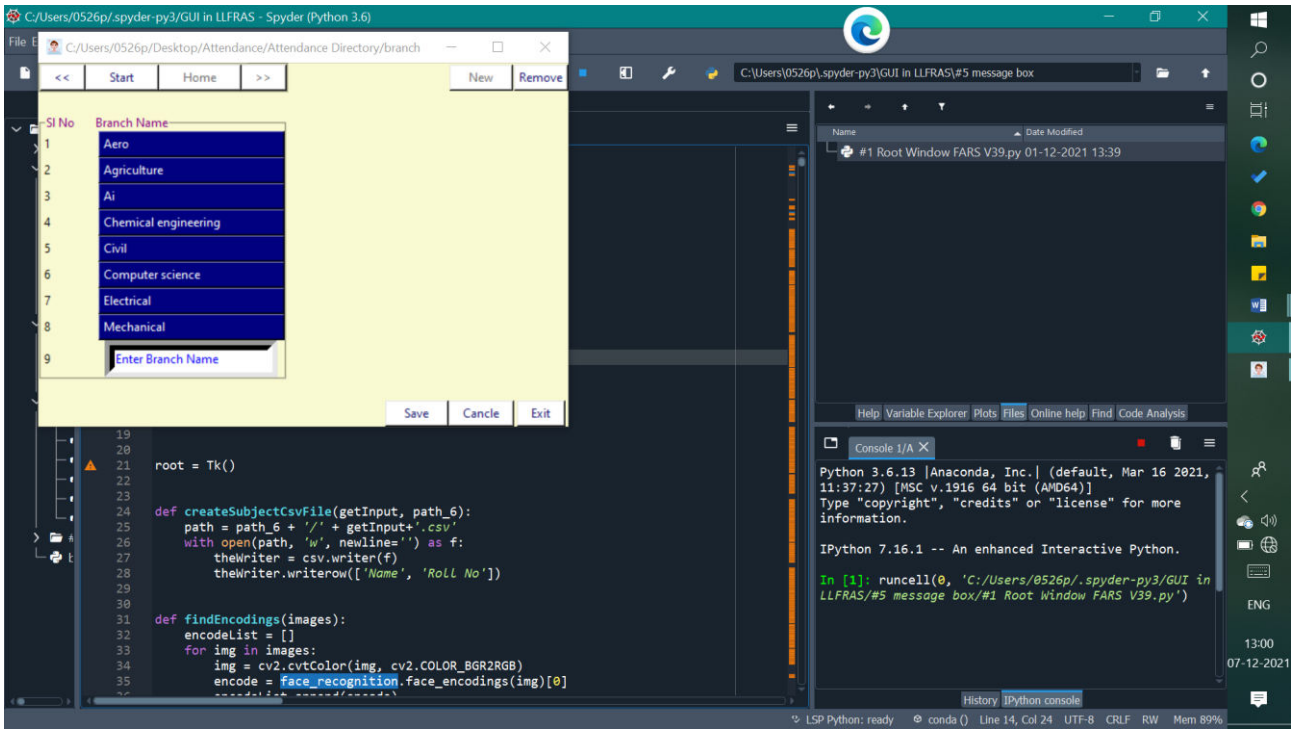
Chapter 5

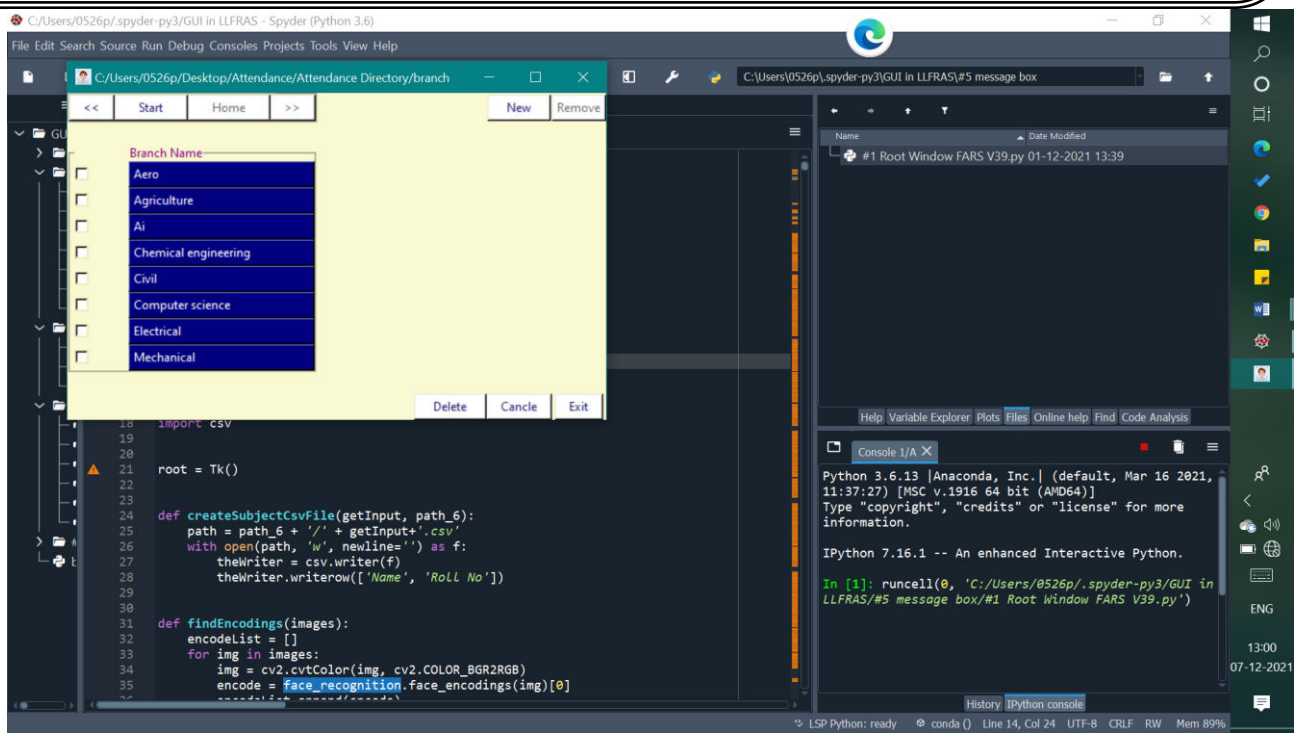
Results and Discussion

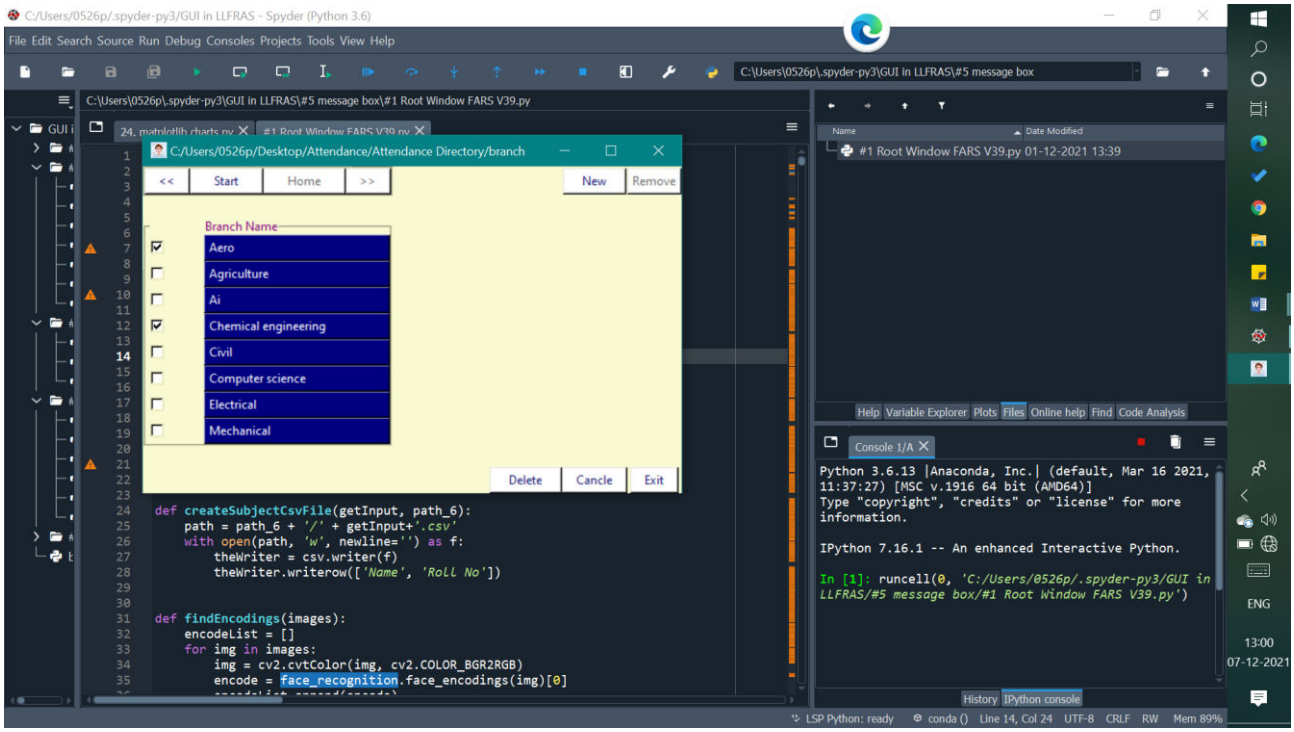


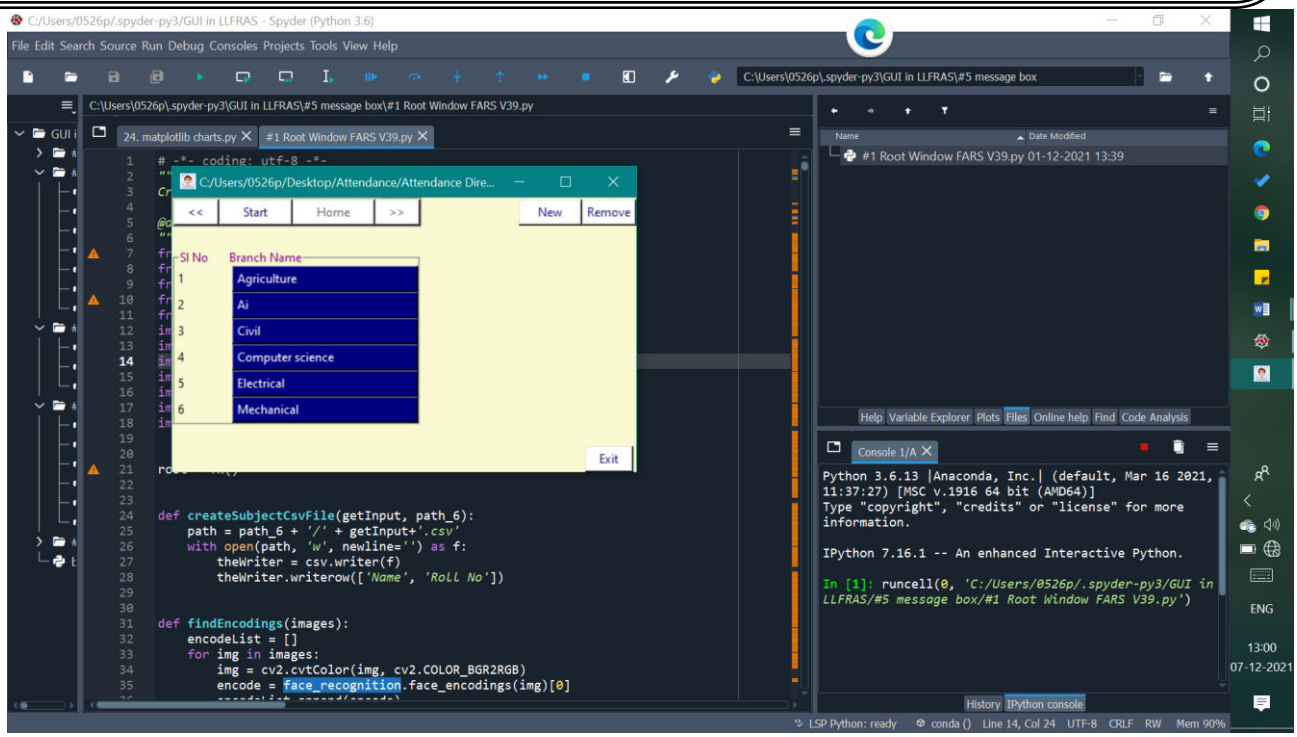


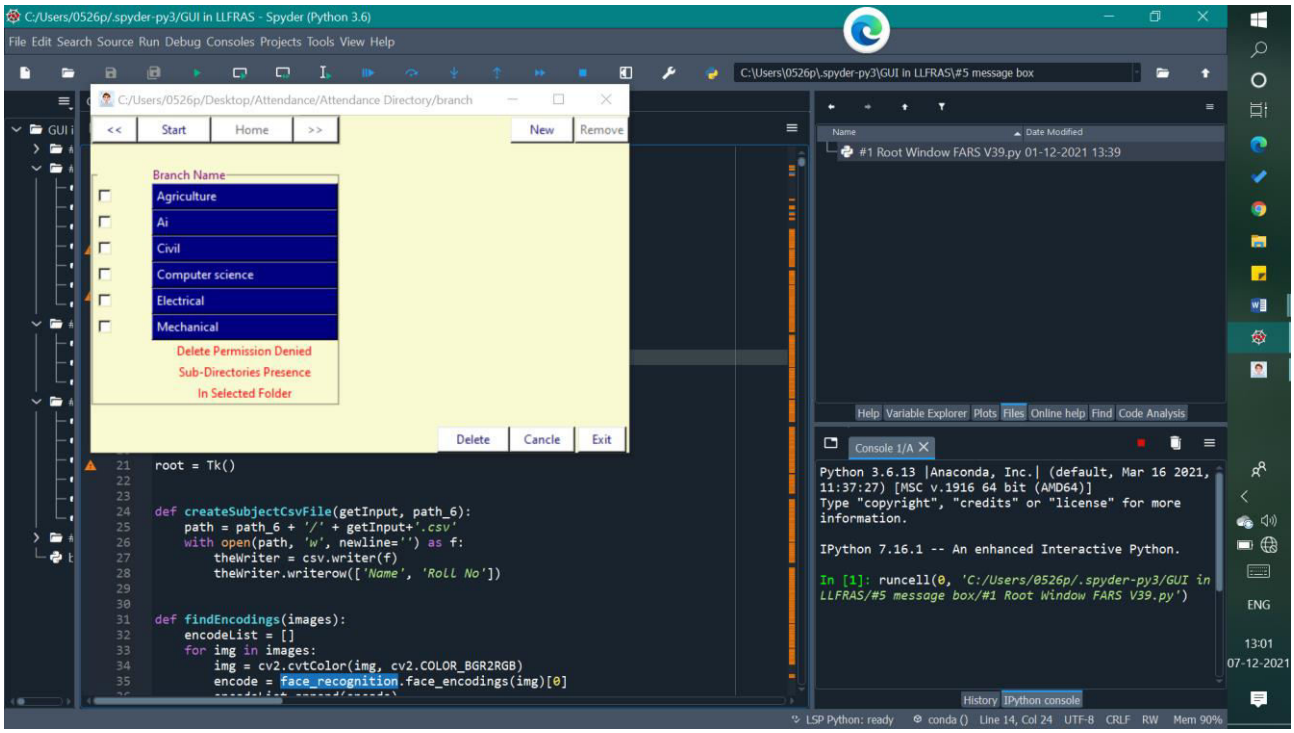

C:/Users/0526p/Desktop/Attendance/Attendance Directory/branch
<< Start Home >> New Remove
Sl No Branch Name
1 Aero
2 Agriculture
3 Ai
4 Civil
5 Computer science
6 Electrical
7 Mechanical
8 Enter Branch Name
Save Cancle Exit
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

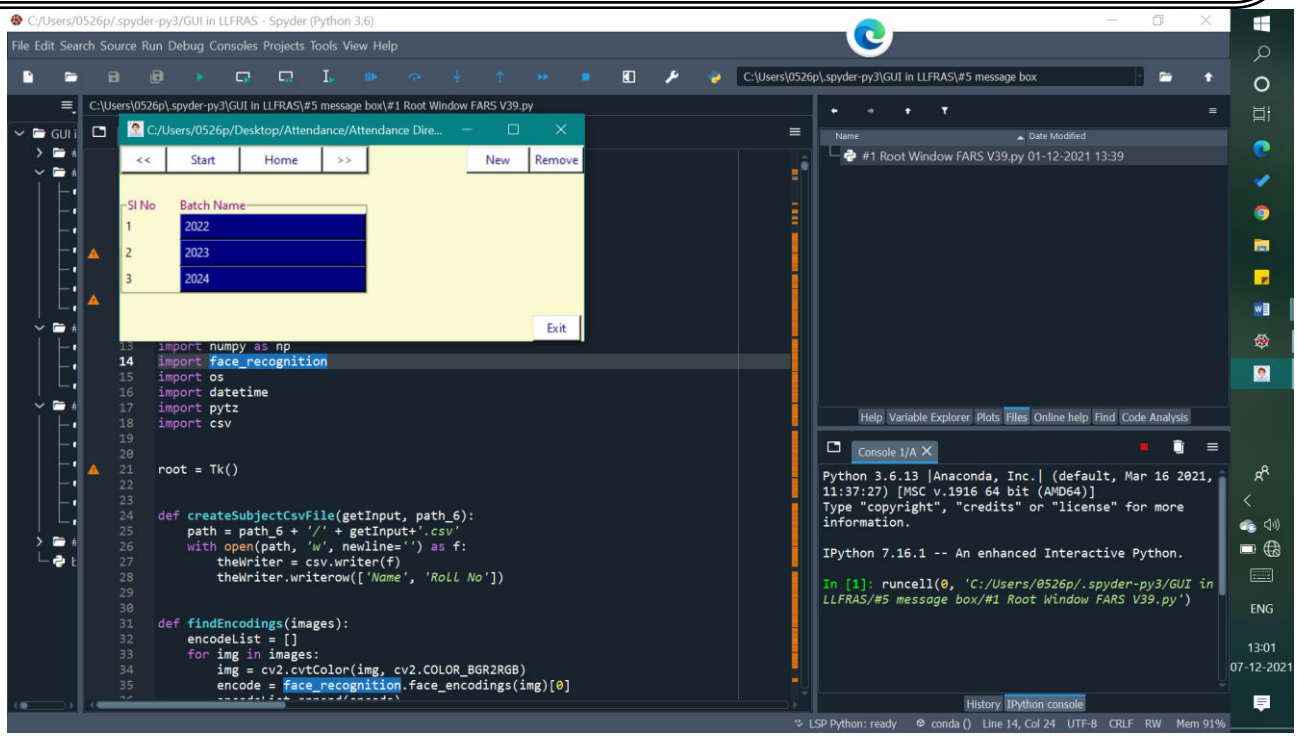

C:/Users/0526p/Desktop/Attendance/Attendance Directory/branch
<< Start Home >> New Remove
Sl No Branch Name
1 Aero
2 Agriculture
3 Ai
4 Chemical engineering
5 Civil
6 Computer science
7 Electrical
8 Mechanical
9 Enter Branch Name
Save Cancle Exit
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

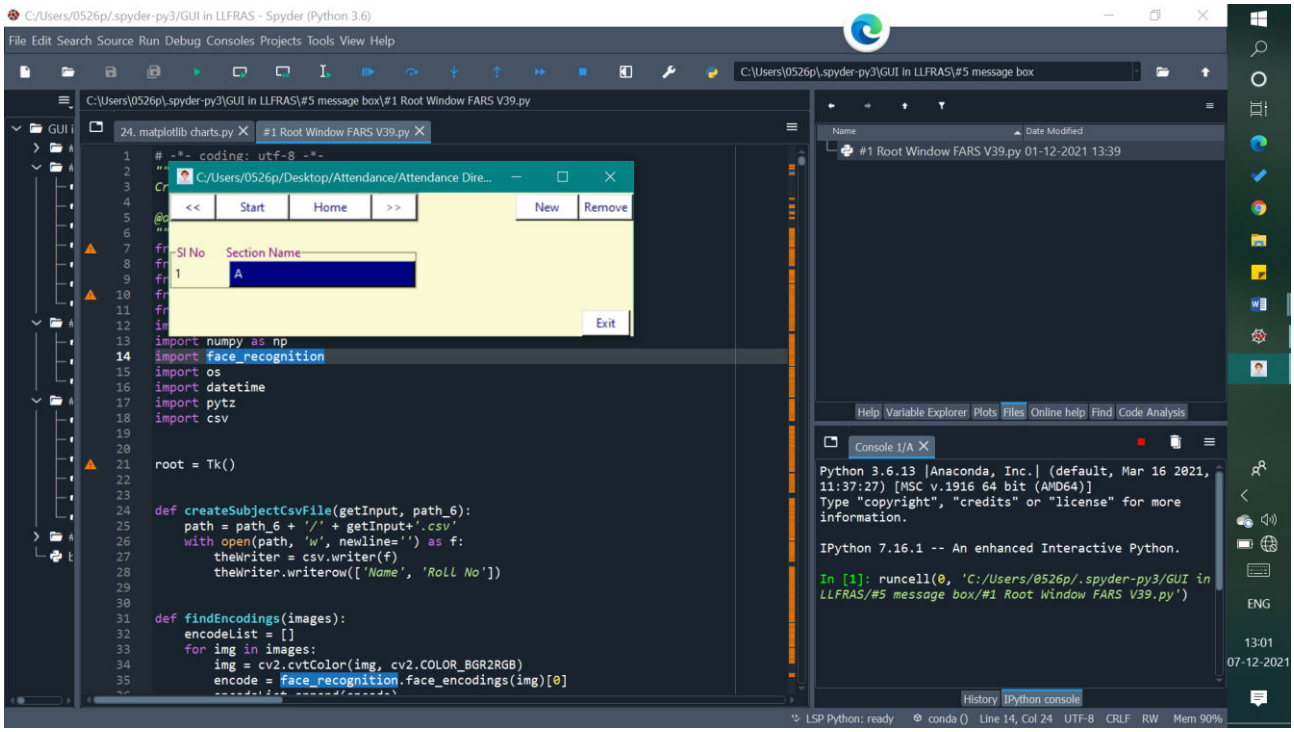

C:/Users/0526p/Desktop/Attendance/Attendance Directory/branch
<< Start Home >> New Remove
Branch Name
Aero
Agriculture
Ai
Chemical engineering
Civil
Computer science
Electrical
Mechanical
Delete Cancle Exit
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

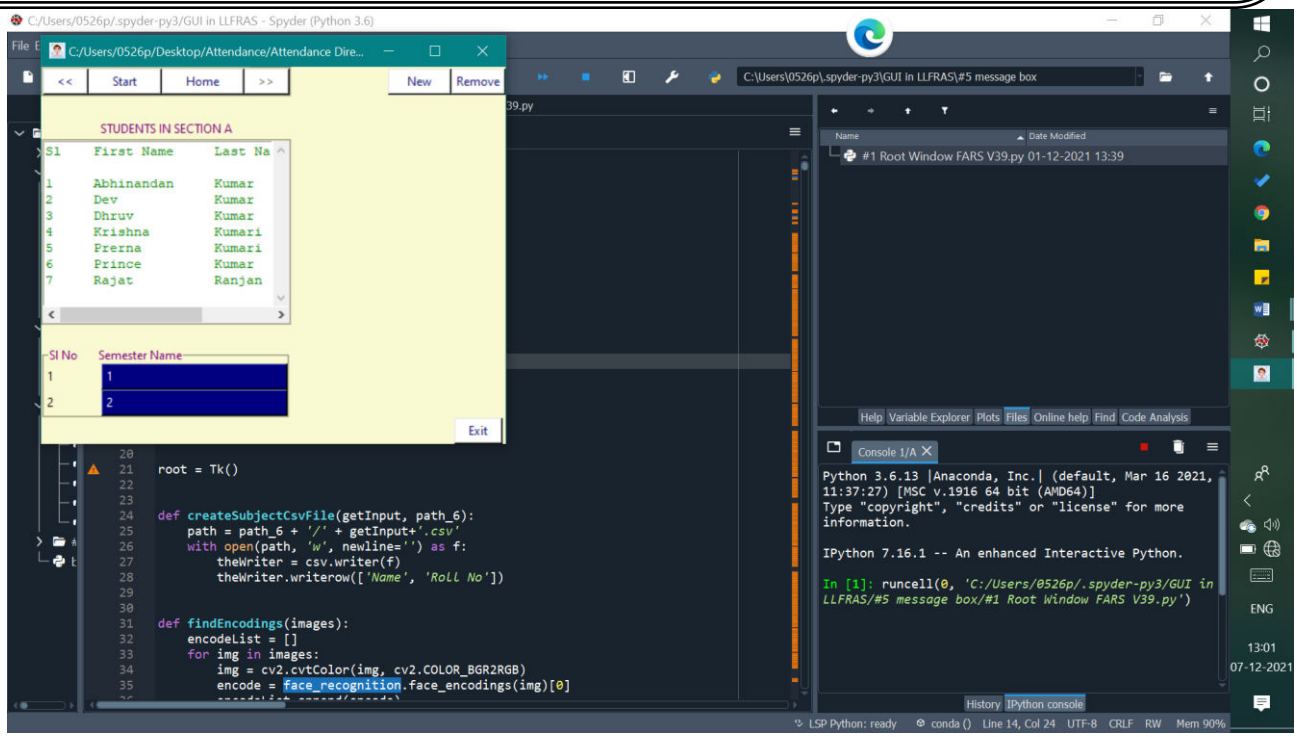

C:/Users/0526p/Desktop/Attendance/Attendance Directory/branch
<< Start Home >> New Remove
Branch Name
Aero
Agriculture
Ai
Chemical engineering
Civil
Computer science
Electrical
Mechanical
Delete Cancle Exit
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

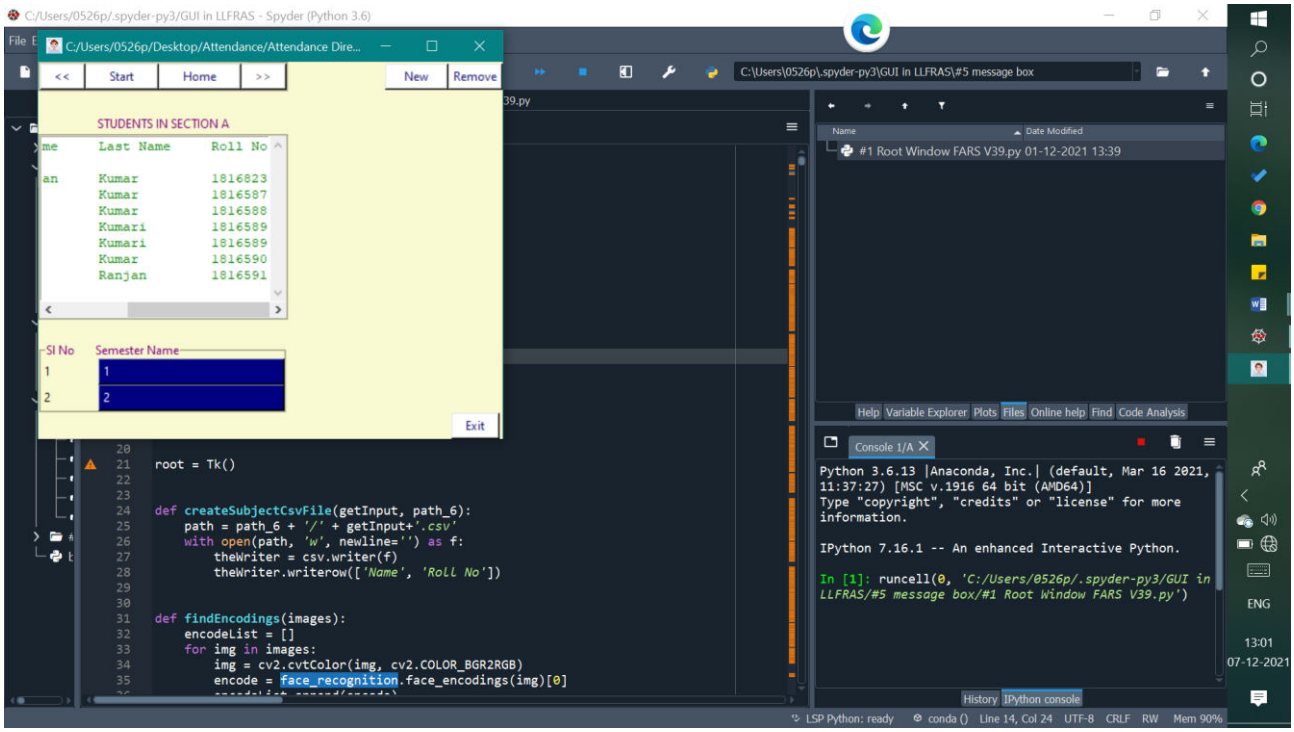

C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:/Users/0526p/Desktop/Attendance/Attendance Dire...
<< Start Home >> New Remove
Sl No Branch Name
1 Agriculture
2 Ai
3 Civil
4 Computer science
5 Electrical
6 Mechanical
Exit
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'RoLL No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

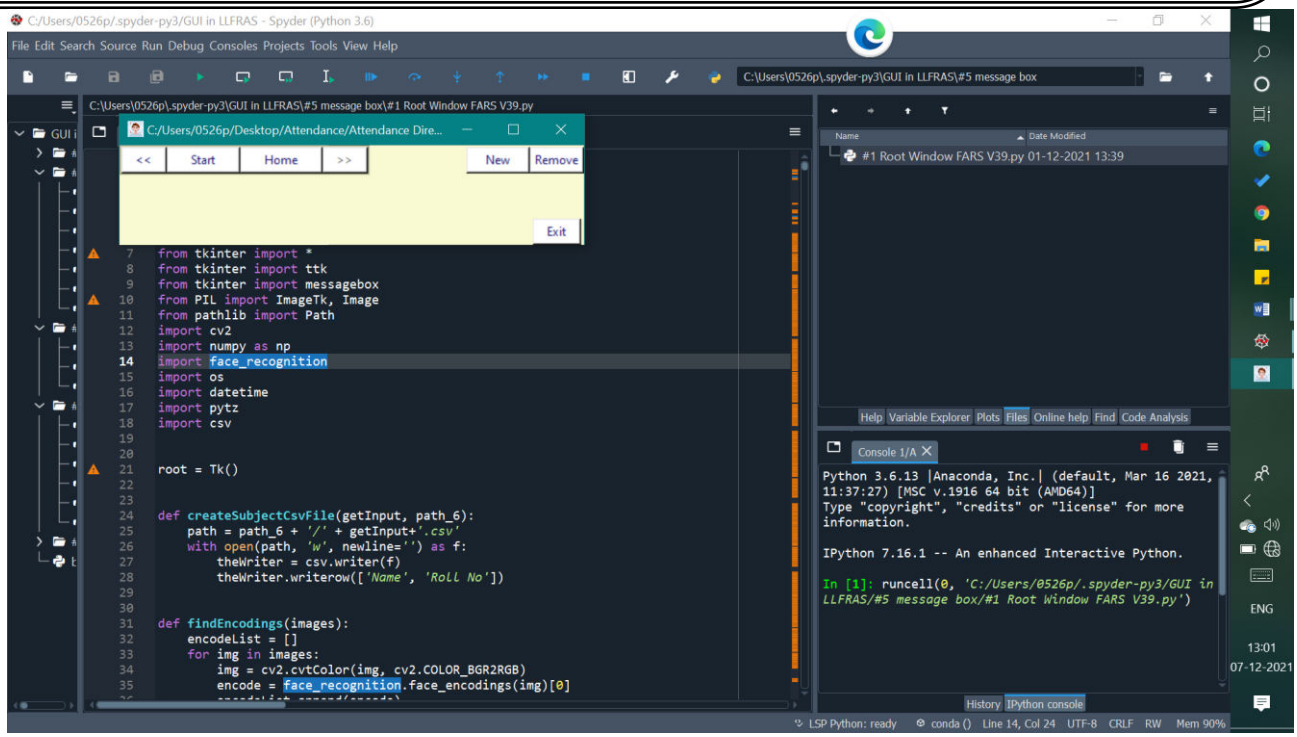

C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:/Users/0526p/Desktop/Attendance/Attendance Directory/branch
<< Start Home >> New Remove
Branch Name
Agriculture
Ai
Civil
Computer science
Electrical
Mechanical
Delete Permission Denied
Sub-Directories Presence
In Selected Folder
Delete Cancle Exit
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'RoLL No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

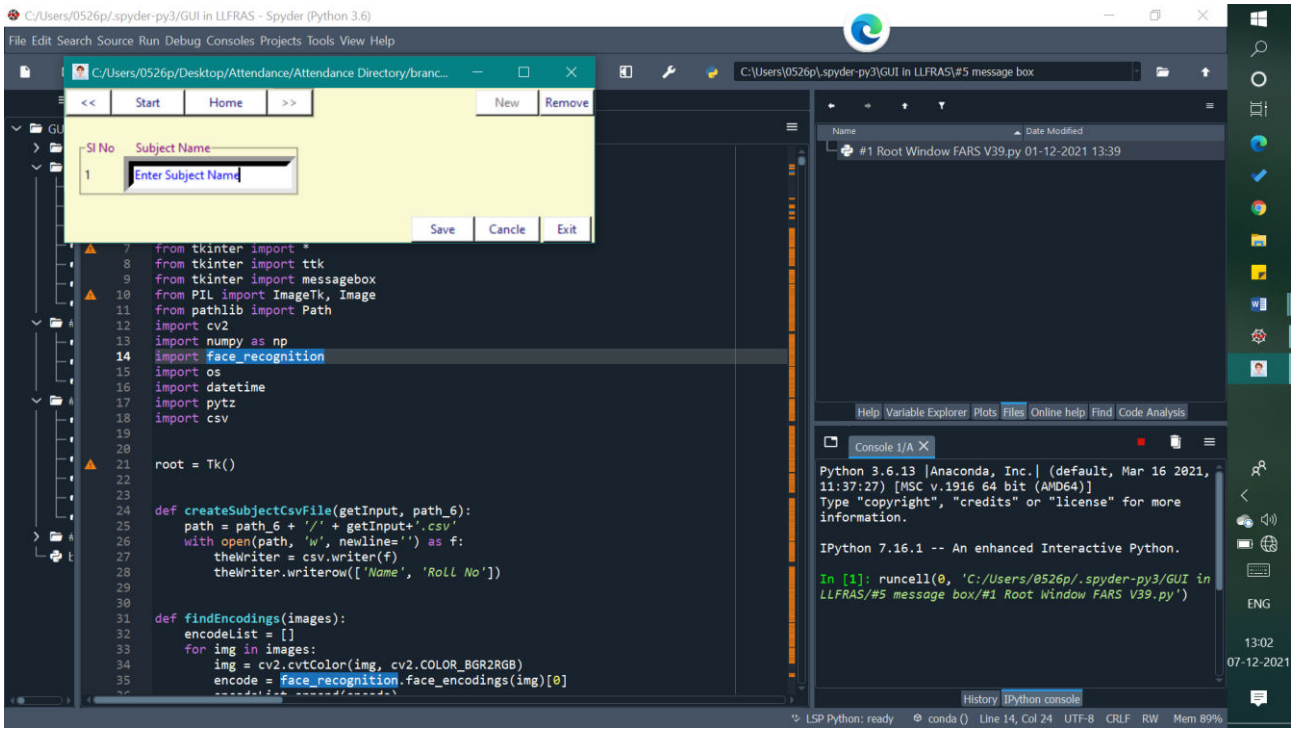

C:/Users/0526p/Desktop/Attendance/Attendance Dire...
<< Start Home >> New Remove
Sl No Batch Name
1 2022
2 2023
3 2024
Exit
import numpy as np
import face_recognition
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

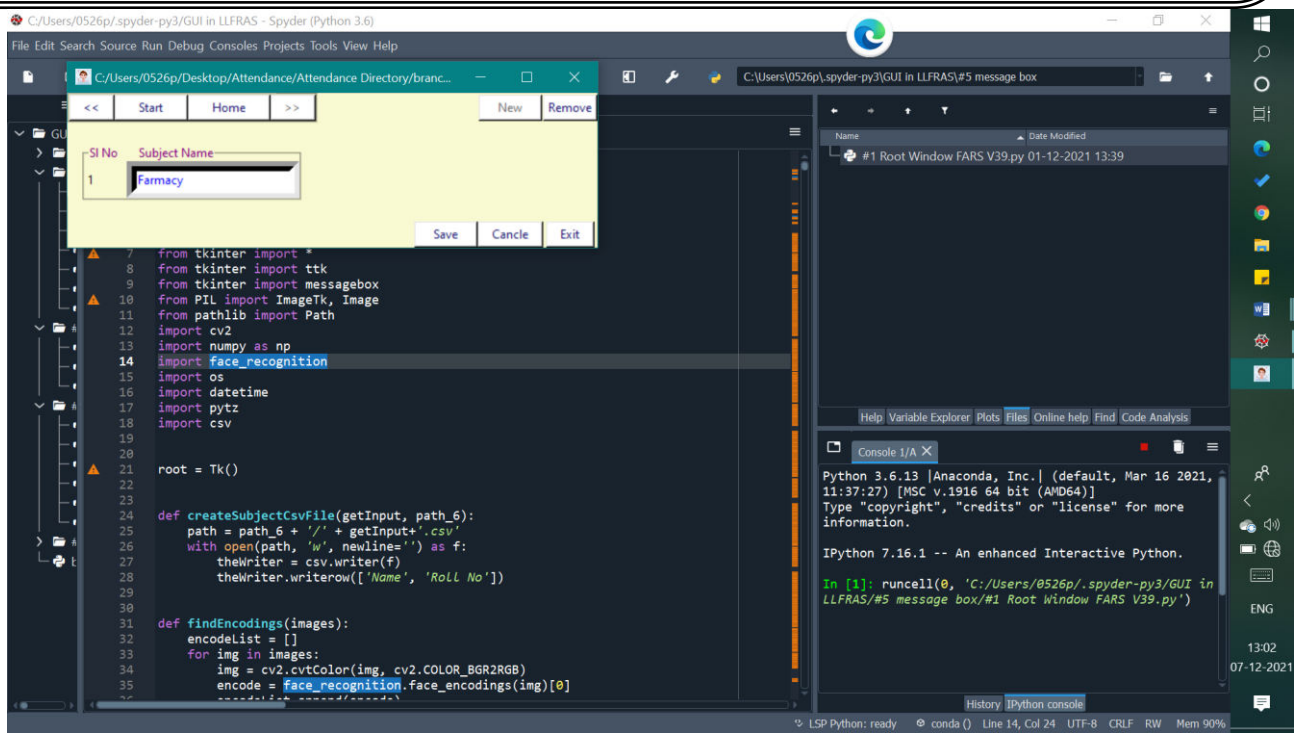

C:/Users/0526p/Desktop/Attendance/Attendance Dire...
<< Start Home >> New Remove
Sl No Section Name
1 A
Exit
import numpy as np
import face_recognition
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

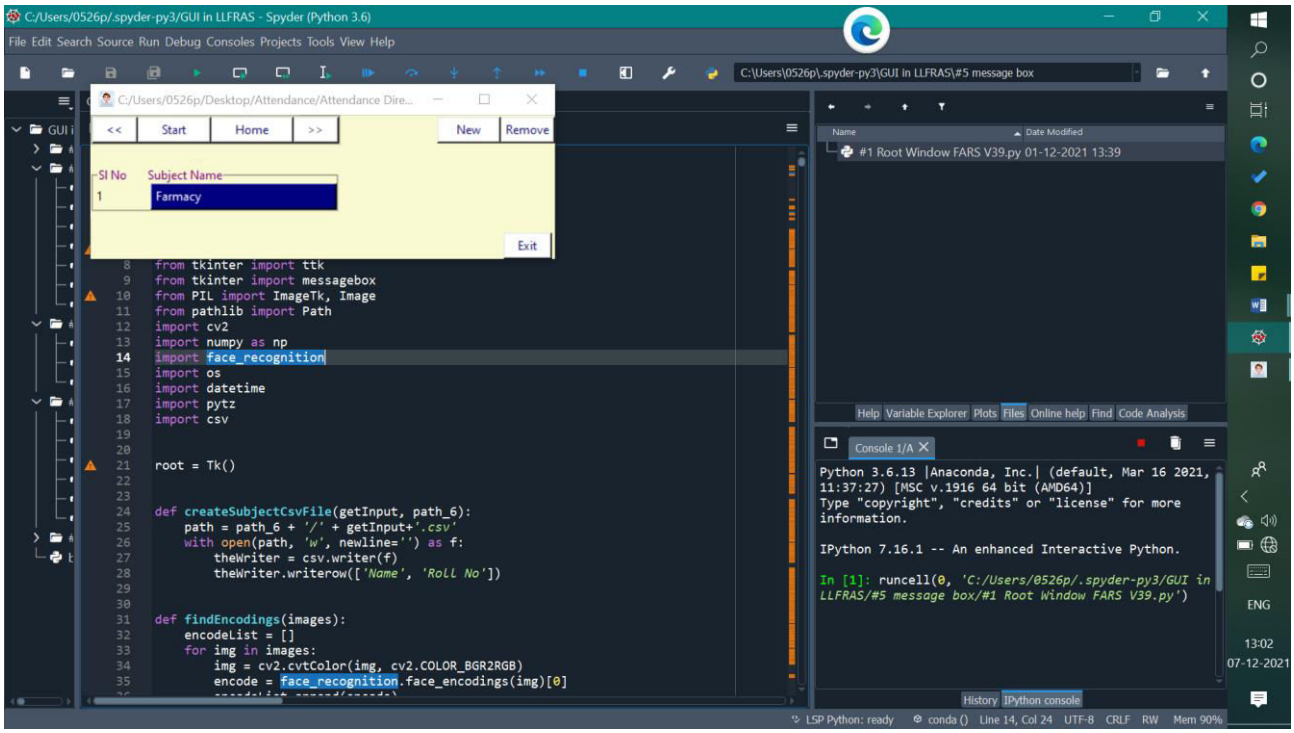

C:/Users/0526p/Desktop/Attendance/Attendance Dire...
Start
Home
New
Remove
STUDENTS IN SECTION A
Sl First Name Last Na
1 Abhinandan Kumar
2 Dev Kumar
3 Dhruv Kumar
4 Krishna Kumari
5 Prerna Kumari
6 Prince Kumar
7 Rajat Ranjan
Sl No Semester Name
Exit
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

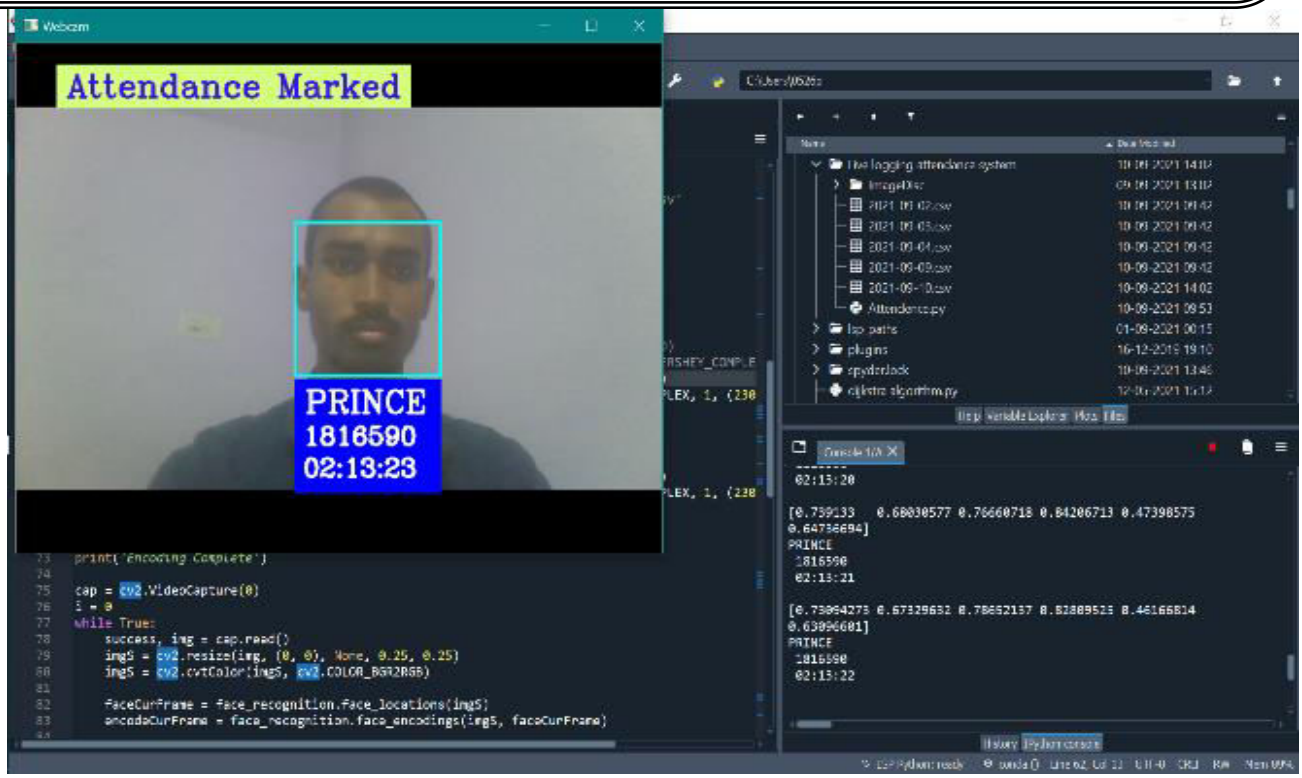

C:/Users/0526p/Desktop/Attendance/Attendance Dire...
Start
Home
New
Remove
STUDENTS IN SECTION A
Last Name Roll No
Kumar 1816823
Kumar 1816587
Kumar 1816588
Kumari 1816589
Kumari 1816589
Kumar 1816590
Ranjan 1816591
Sl No Semester Name
Exit
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

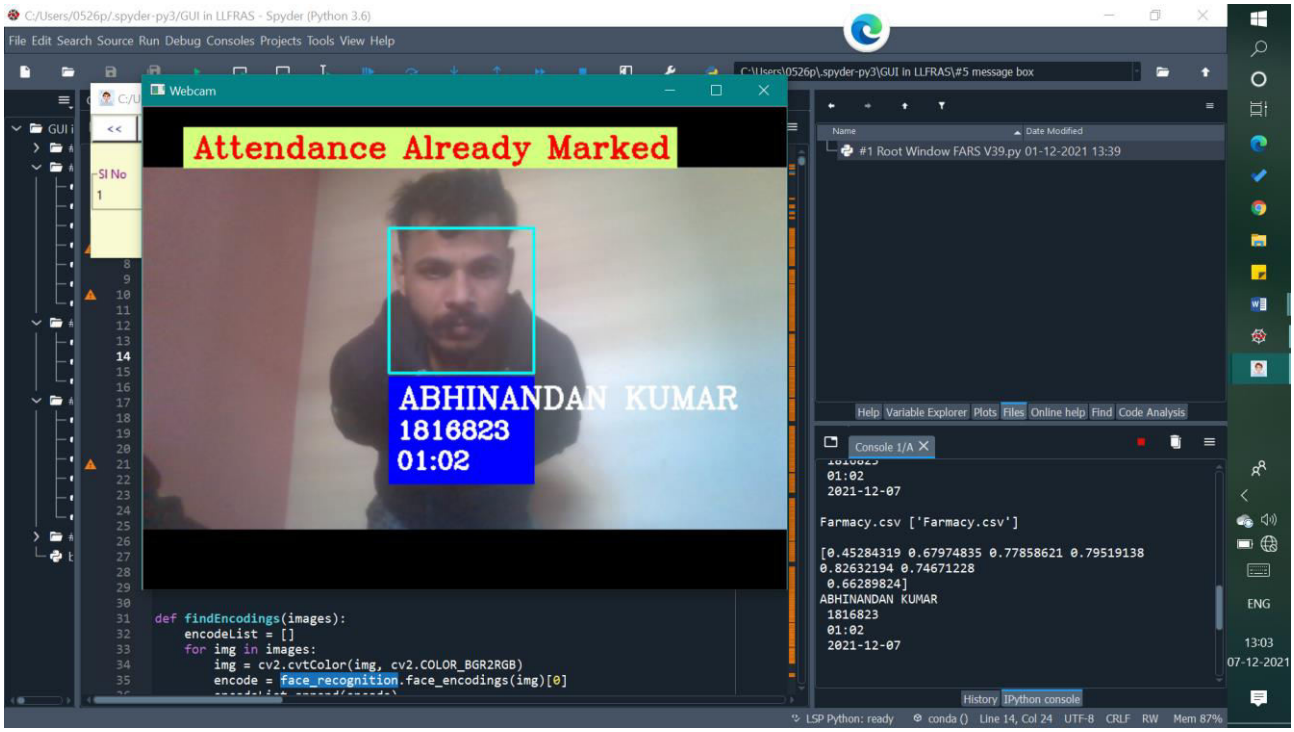

C:/Users/0526p/Desktop/Attendance/Attendance Dire...
<< Start Home >> New Remove Exit
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
from PIL import ImageTk, Image
from pathlib import Path
import cv2
import numpy as np
import face_recognition
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

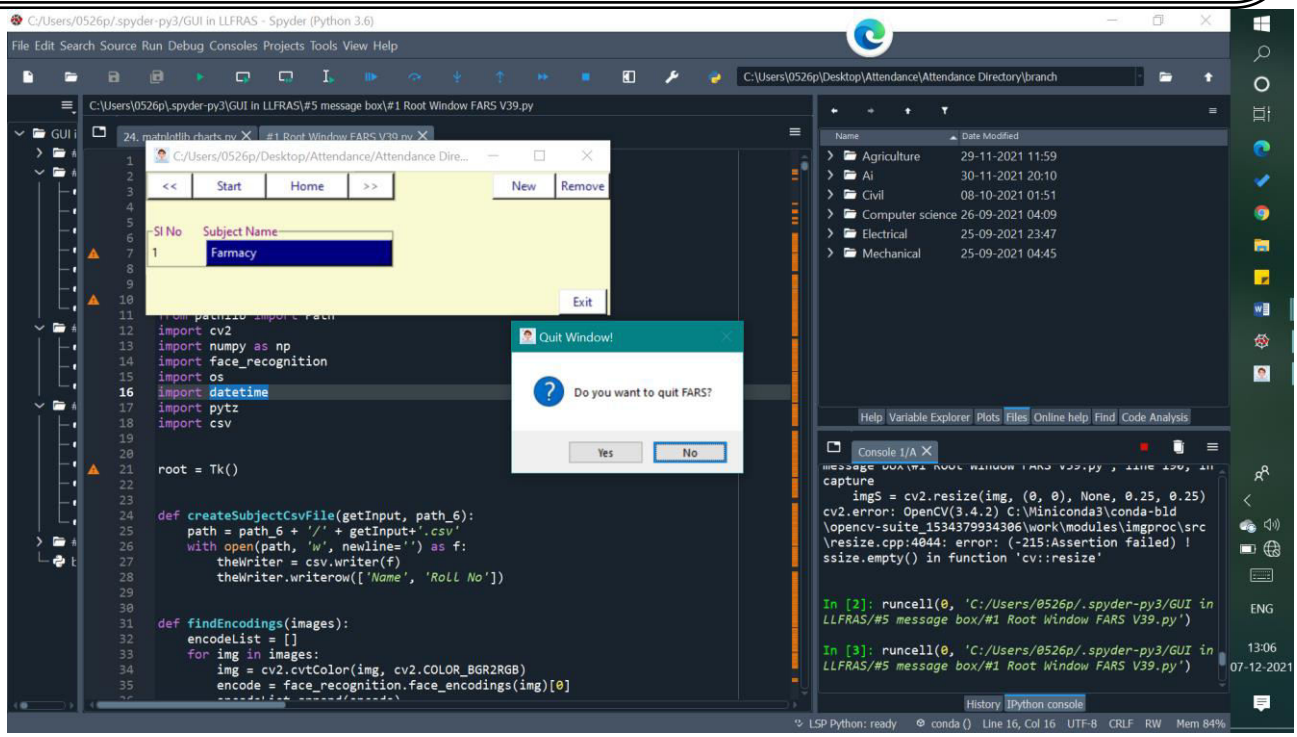

C:/Users/0526p/Desktop/Attendance/Attendance Directory/branc...
<< Start Home >> New Remove
Sl No Subject Name
1 Enter Subject Name
Save Cancle Exit
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
from PIL import ImageTk, Image
from pathlib import Path
import cv2
import numpy as np
import face_recognition
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#1 Root Window FARS V39.py 01-12-2021 13:39
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

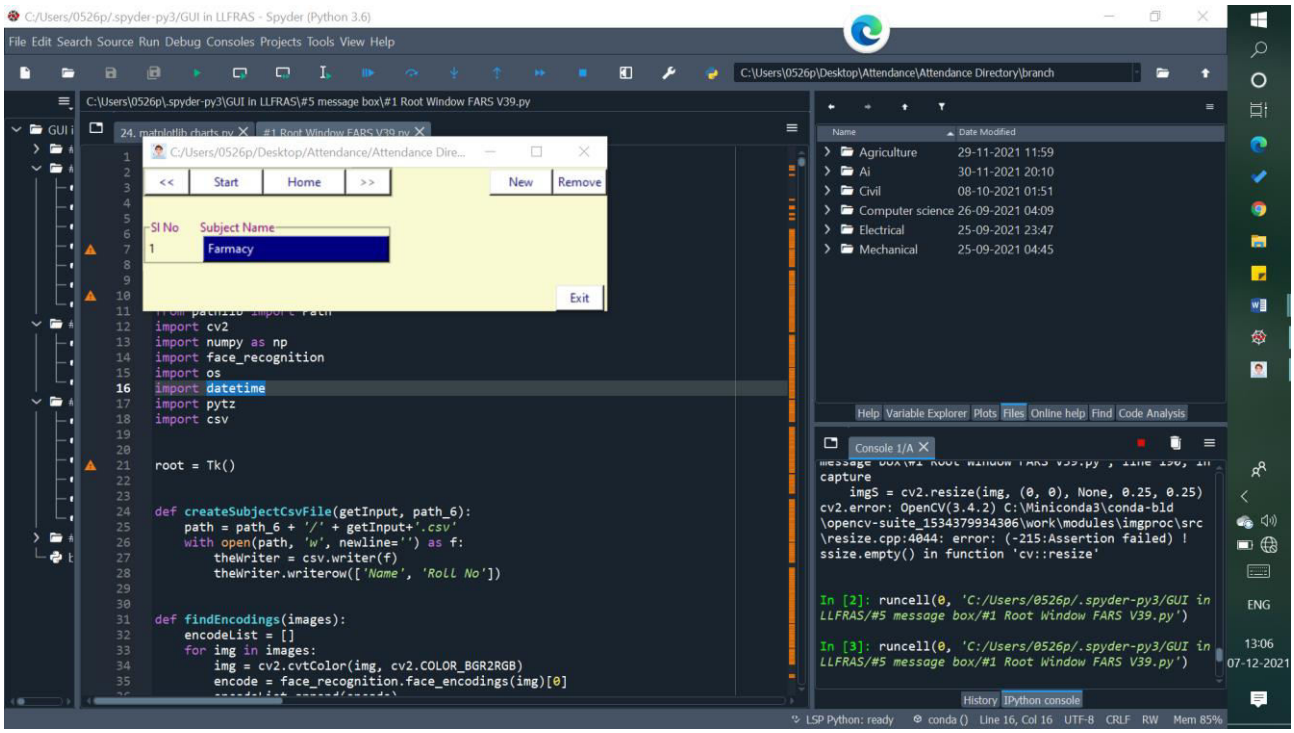

C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:/Users/0526p/Desktop/Attendance/Attendance Directory/branc...
<< Start Home >> New Remove
Sl No Subject Name
1 Farmacy
Save Cancle Exit
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
from PIL import ImageTk, Image
from pathlib import Path
import cv2
import numpy as np
import face_recognition
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#5 message box
#1 Root Window FARS V39.py 01-12-2021 13:39
Help Variable Explorer Plots Files Online help Find Code Analysis
Console 1/A
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
History IPython console
LSP Python: ready conda () Line 14, Col 24 UTF-8 CRLF RW Mem 90%
ENG
13:02
07-12-2021

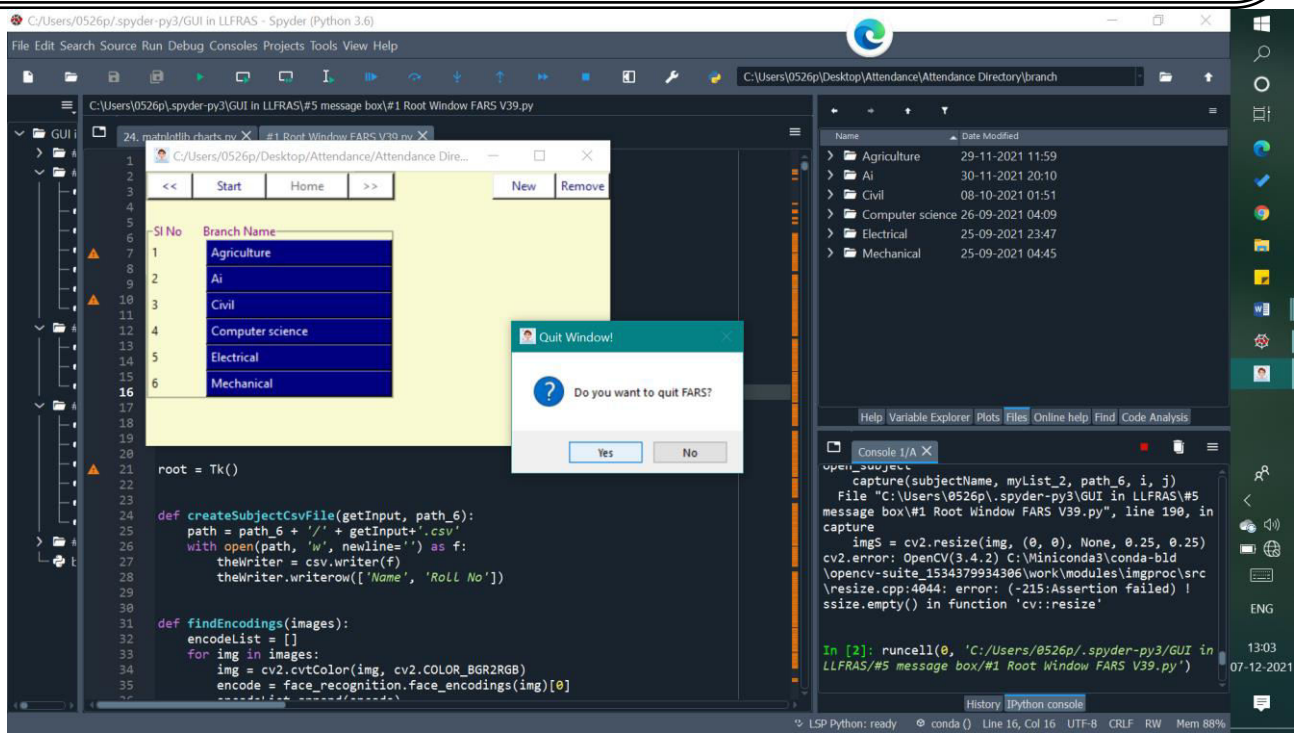

C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:/Users/0526p/Desktop/Attendance/Attendance Dire...
<< Start Home >> New Remove
Sl No Subject Name
1 Farmacy
Exit
from tkinter import ttk
from tkinter import messagebox
from PIL import ImageTk, Image
from pathlib import Path
import cv2
import numpy as np
import face_recognition
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
#5 message box
#1 Root Window FARS V39.py 01-12-2021 13:39
Help Variable Explorer Plots Files Online help Find Code Analysis
Console 1/A
Python 3.6.13 |Anaconda, Inc.| (default, Mar 16 2021, 11:37:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.16.1 -- An enhanced Interactive Python.
In [1]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
History IPython console
LSP Python: ready conda () Line 14, Col 24 UTF-8 CRLF RW Mem 90%
ENG
13:02
07-12-2021

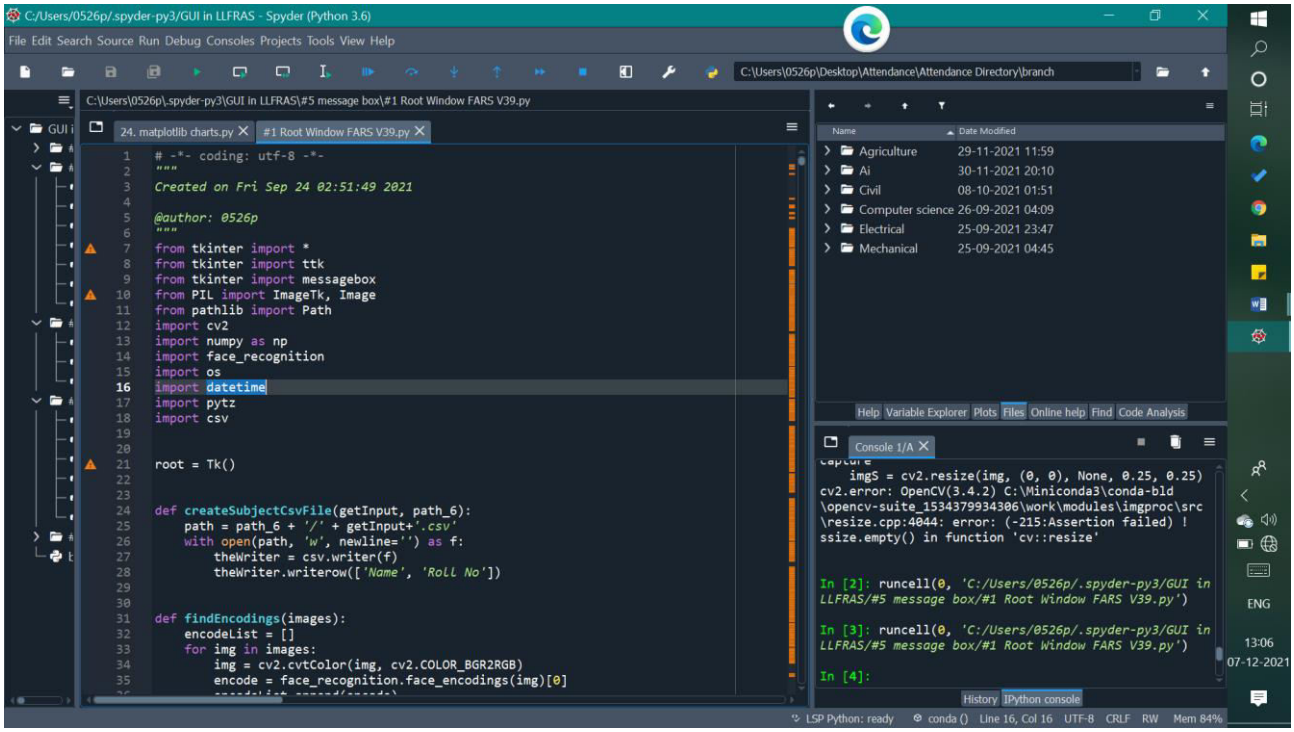

Webcam
Attendance Marked
PRINCE
1816590
02:13:23
print('Encoding Complete')
cap = cv2.VideoCapture(0)
while True:
success, img = cap.read()
imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)
faceCurFrame = face_recognition.face_locations(imgS)
encodeCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)
02:13:20
[0.739133 0.68030577 0.76660718 0.84206713 0.47398575
0.64736694]
PRINCE
1816590
02:13:21
[0.73064275 0.67329632 0.78652137 0.83889525 0.46166814
0.63096601]
PRINCE
1816590
02:13:22

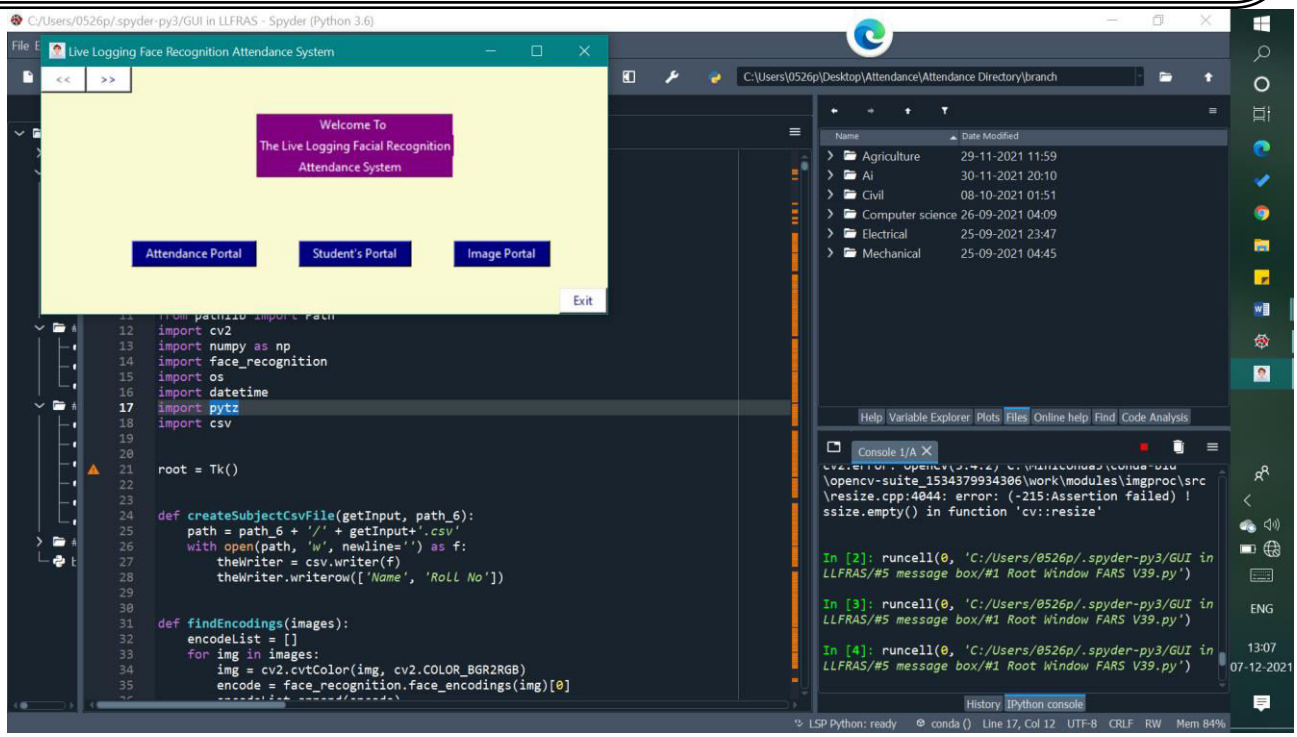

Webcam
Attendance Already Marked
ABHINANDAN KUMAR
1816823
01:02
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
01:02
2021-12-07
Farmacy.csv ['Farmacy.csv']
[0.45284319 0.67974835 0.77858621 0.79519138
0.82632194 0.74671228
0.66289824]
ABHINANDAN KUMAR
1816823
01:02
2021-12-07

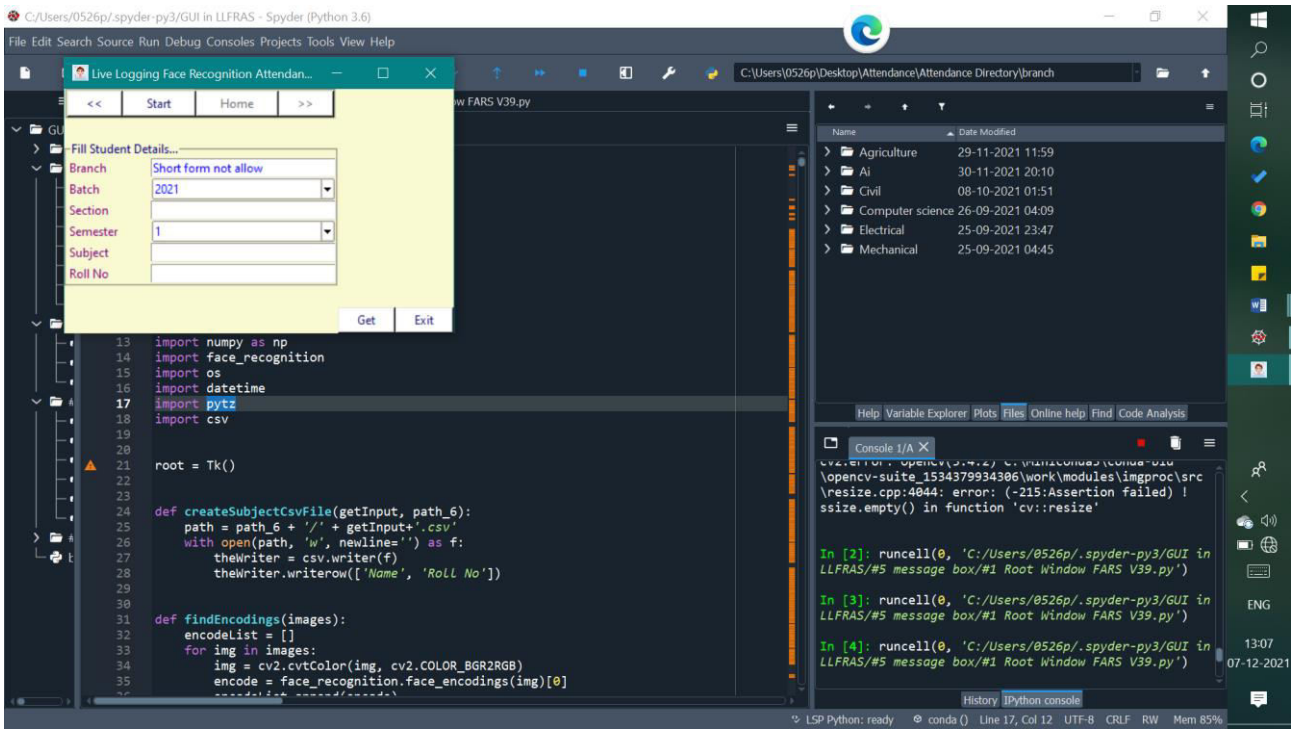

C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\0526p\Desktop\Attendance\Attendance Directory\branch
C:\Users\0526p\.spyder-py3\GUI in LLFRAS\#5 message box\#1 Root Window FARS V39.py
C:/Users/0526p/Desktop/Attendance/Attendance Dire...
<< Start Home >>
New Remove
Sl No Subject Name
1 Farmacy
Exit
Quit Window!
Do you want to quit FARS?
Yes No
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
Help Variable Explorer Plots Files Online help Find Code Analysis
Console 1/A
capture
imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
cv2.error: OpenCV(3.4.2) C:\Miniconda3\conda-bld\opencv-suite_1534379934306\work\modules\imgproc\src\resize.cpp:4044: error: (-215:Assertion failed) !ssize.empty() in function 'cv::resize'
In [2]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [3]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
History IPython console
LSP Python: ready conda () Line 16, Col 16 UTF-8 CRLF RW Mem 84%
ENG
13:06
07-12-2021

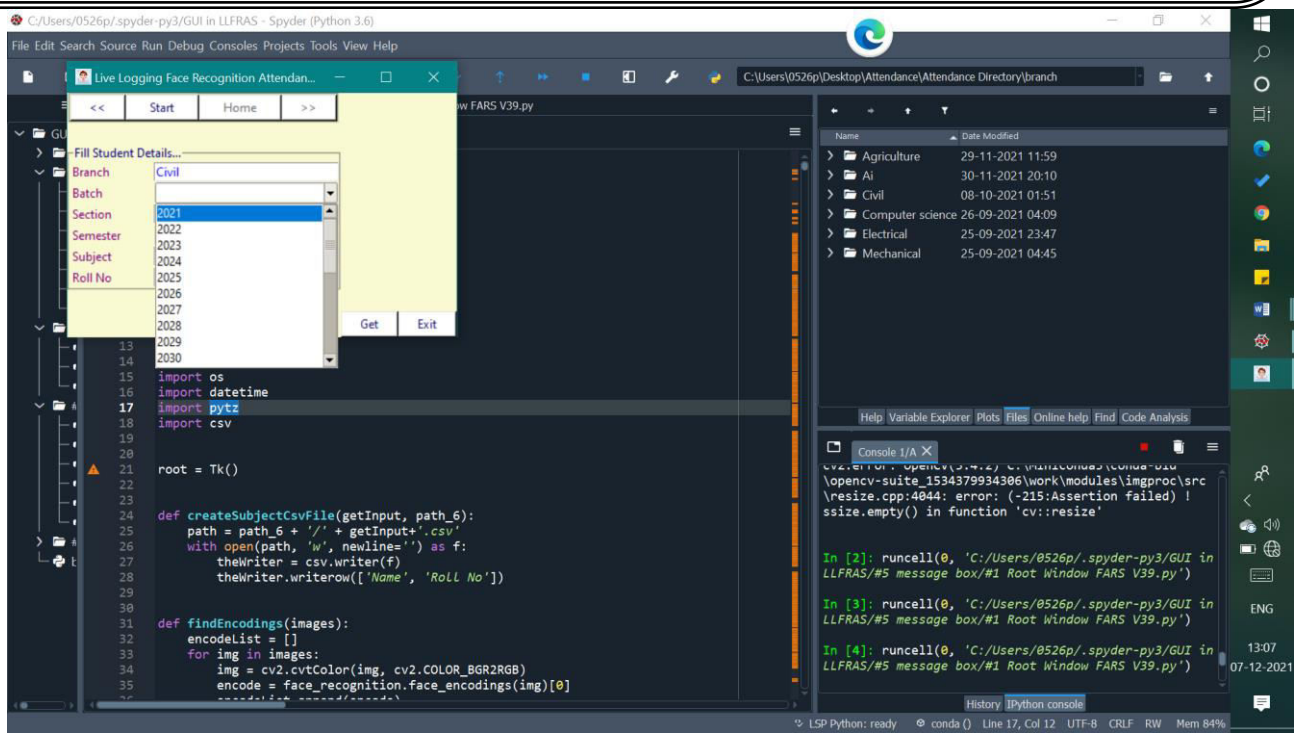

C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\0526p\Desktop\Attendance\Attendance Directory\branch
C:\Users\0526p\.spyder-py3\GUI in LLFRAS\#5 message box\#1 Root Window FARS V39.py
C:/Users/0526p/Desktop/Attendance/Attendance Dire...
<< Start Home >>
New Remove
Sl No Subject Name
1 Farmacy
Exit
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
Help Variable Explorer Plots Files Online help Find Code Analysis
Console 1/A
capture
imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
cv2.error: OpenCV(3.4.2) C:\Miniconda3\conda-bld\opencv-suite_1534379934306\work\modules\imgproc\src\resize.cpp:4044: error: (-215:Assertion failed) !ssize.empty() in function 'cv::resize'
In [2]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [3]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
History IPython console
LSP Python: ready conda () Line 16, Col 16 UTF-8 CRLF RW Mem 85%
ENG
13:06
07-12-2021

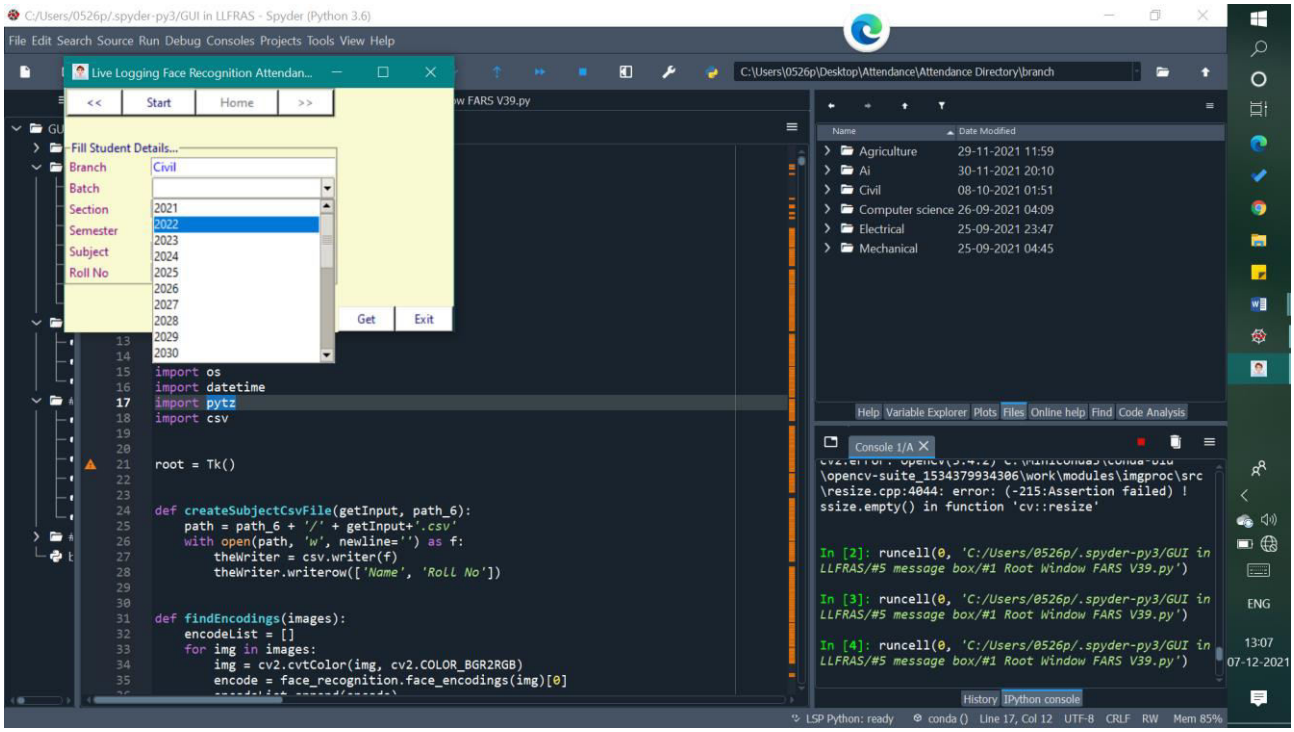

C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\0526p\Desktop\Attendance\Attendance Directory\branch
C:\Users\0526p\.spyder-py3\GUI in LLFRAS\#5 message box\#1 Root Window FARS V39.py
24. matplotlib charts.py
#1 Root Window FARS V39.py
C:/Users/0526p/Desktop/Attendance/Attendance Dire...
<<
Start
Home
>>
New
Remove
Sl No
Branch Name
1
Agriculture
2
Ai
3
Civil
4
Computer science
5
Electrical
6
Mechanical
Quit Window!
Do you want to quit FARS?
Yes
No
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
Name
Date Modified
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
Help Variable Explorer Plots Files Online help Find Code Analysis
Console 1/A
capture(subjectName, myList_2, path_6, i, j)
File "C:\Users\0526p\.spyder-py3\GUI in LLFRAS\#5 message box\#1 Root Window FARS V39.py", line 190, in capture
imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
cv2.error: OpenCV(3.4.2) C:\Miniconda3\conda-bld\opencv-suite_1534379934306\work\modules\imgproc\src\resize.cpp:4044: error: (-215:Assertion failed) !ssize.empty() in function 'cv::resize'
In [2]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
History IPython console
LSP Python: ready
conda ()
Line 16, Col 16
UTF-8
CRLF
RW
Mem 88%
ENG
13:03
07-12-2021

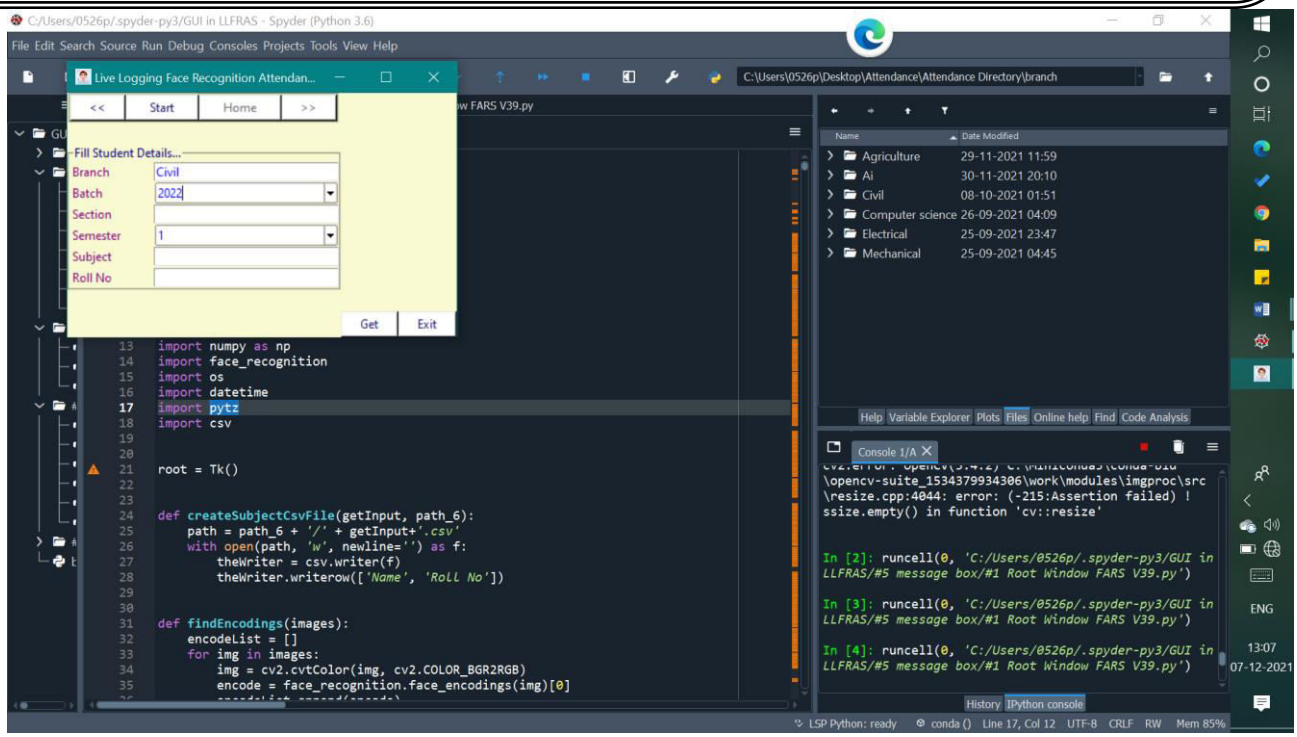

C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\0526p\Desktop\Attendance\Attendance Directory\branch
C:\Users\0526p\.spyder-py3\GUI in LLFRAS\#5 message box\#1 Root Window FARS V39.py
24. matplotlib charts.py
#1 Root Window FARS V39.py
-*- coding: utf-8 -*-
"""
Created on Fri Sep 24 02:51:49 2021
@author: 0526p
"""
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
from PIL import ImageTk, Image
from pathlib import Path
import cv2
import numpy as np
import face_recognition
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
Name
Date Modified
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
Help Variable Explorer Plots Files Online help Find Code Analysis
Console 1/A
imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
cv2.error: OpenCV(3.4.2) C:\Miniconda3\conda-bld\opencv-suite_1534379934306\work\modules\imgproc\src\resize.cpp:4044: error: (-215:Assertion failed) !ssize.empty() in function 'cv::resize'
In [2]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [3]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [4]:
History IPython console
LSP Python: ready
conda ()
Line 16, Col 16
UTF-8
CRLF
RW
Mem 84%
ENG
13:06
07-12-2021

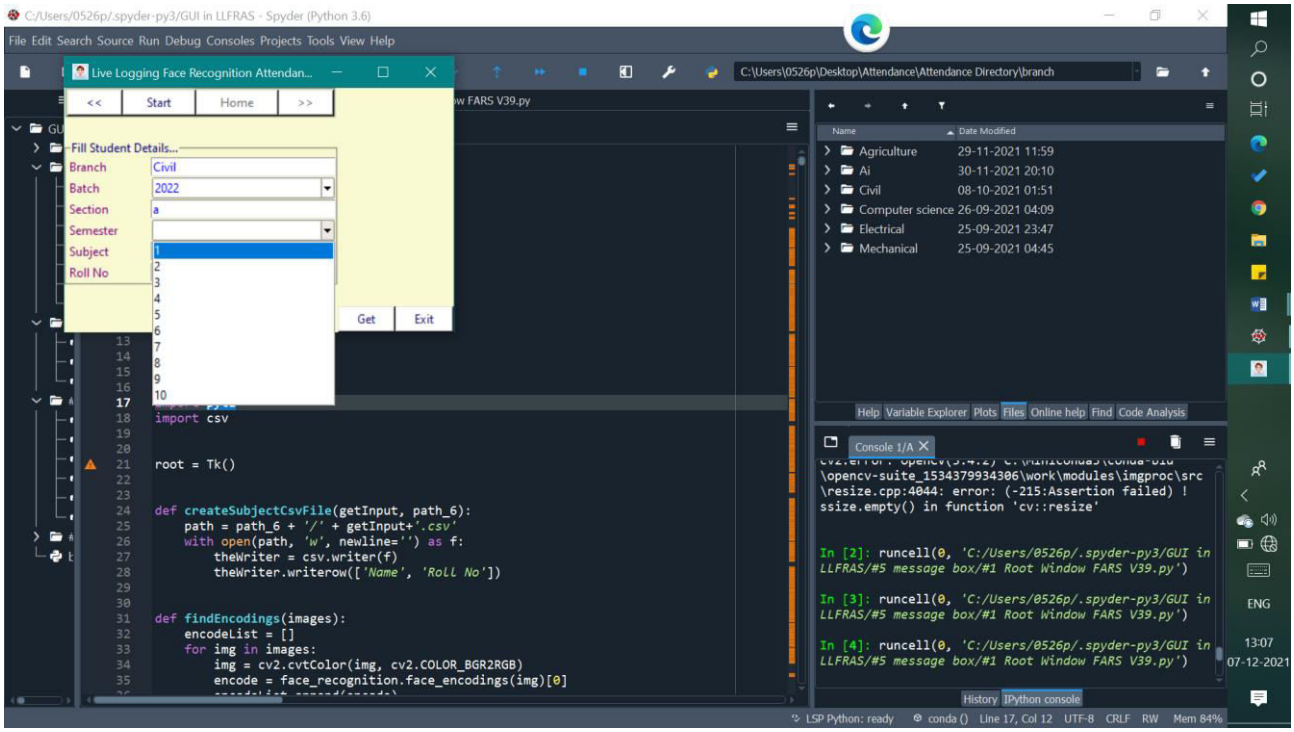

Live Logging Face Recognition Attendance System
Welcome To
The Live Logging Facial Recognition
Attendance System
Attendance Portal
Student's Portal
Image Portal
Exit
C:\Users\0526p\Desktop\Attendance\Attendance Directory\branch
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
import cv2
import numpy as np
import face_recognition
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'RoLL No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
\opencv-suite_1534379934306\work\modules\imgproc\src
\resize.cpp:4044: error: (-215:Assertion failed) !
ssize.empty() in function 'cv::resize'
In [2]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in
LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [3]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in
LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [4]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in
LLFRAS/#5 message box/#1 Root Window FARS V39.py')

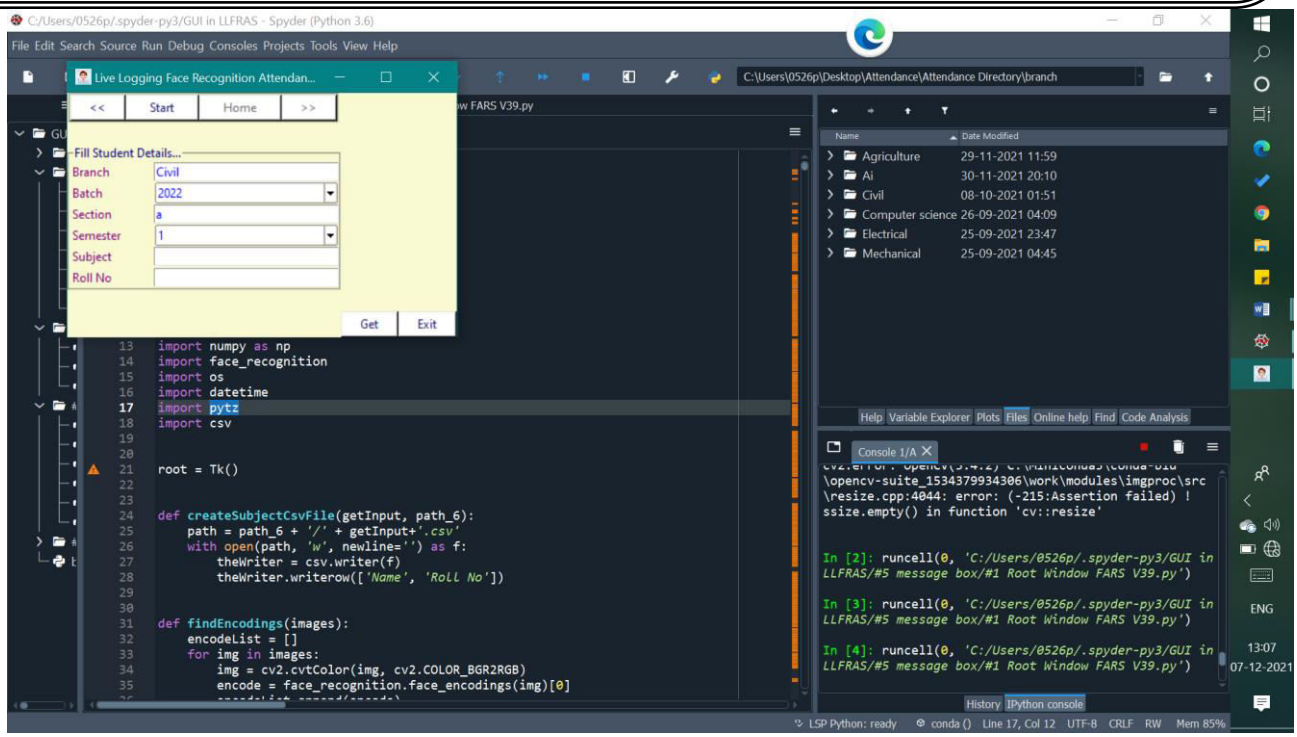

Live Logging Face Recognition Attendan...
Start
Home
Fill Student Details...
Branch
Short form not allow
Batch
2021
Section
Semester
1
Subject
Roll No
Get
Exit
C:\Users\0526p\Desktop\Attendance\Attendance Directory\branch
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
\opencv-suite_1534379934306\work\modules\imgproc\src
\resize.cpp:4044: error: (-215:Assertion failed) !
ssize.empty() in function 'cv::resize'

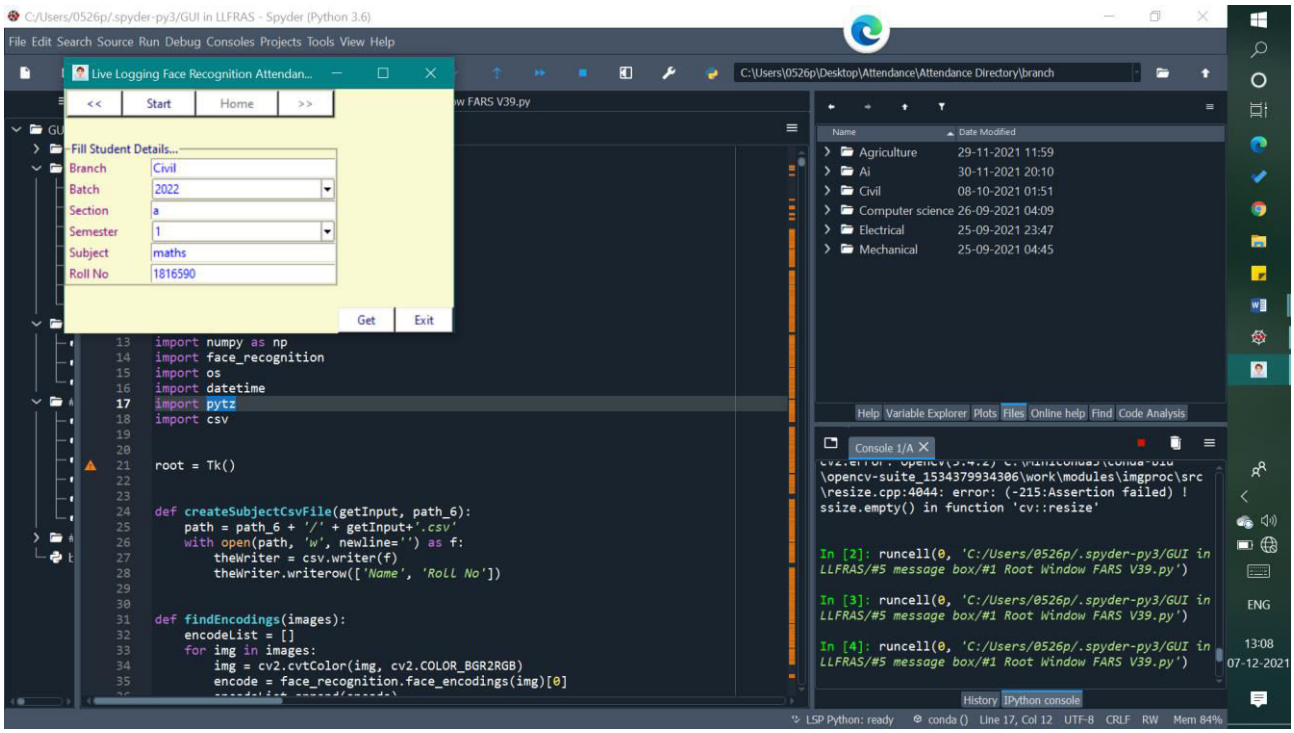

Live Logging Face Recognition Attendan...
Fill Student Details...
Branch
Civil
Batch
Section
Semester
Subject
Roll No
Get
Exit
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
def findEncodings(images):
Agriculture
Ai
Civil
Computer science
Electrical
Mechanical

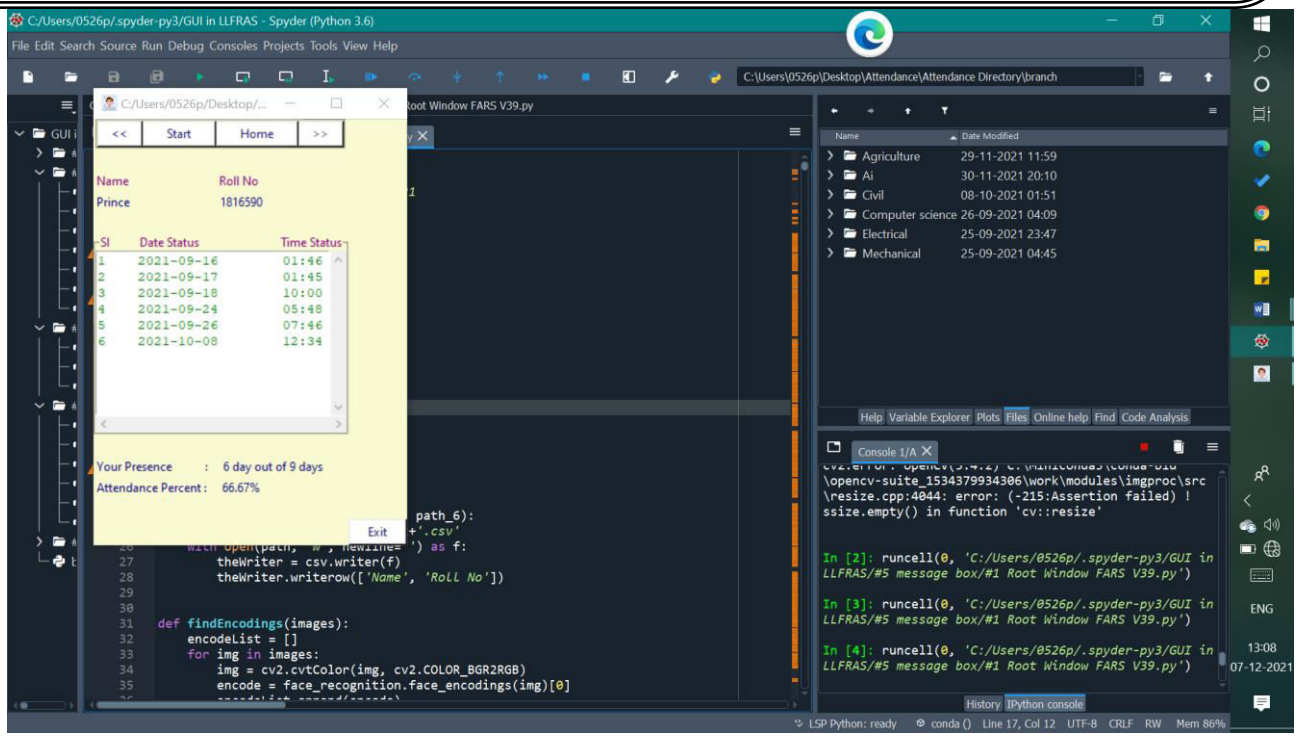

Live Logging Face Recognition Attendan...
Fill Student Details...
Branch
Civil
Batch
Section
Semester
Subject
Roll No
Get
Exit
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
def findEncodings(images):
Agriculture
Ai
Civil
Computer science
Electrical
Mechanical

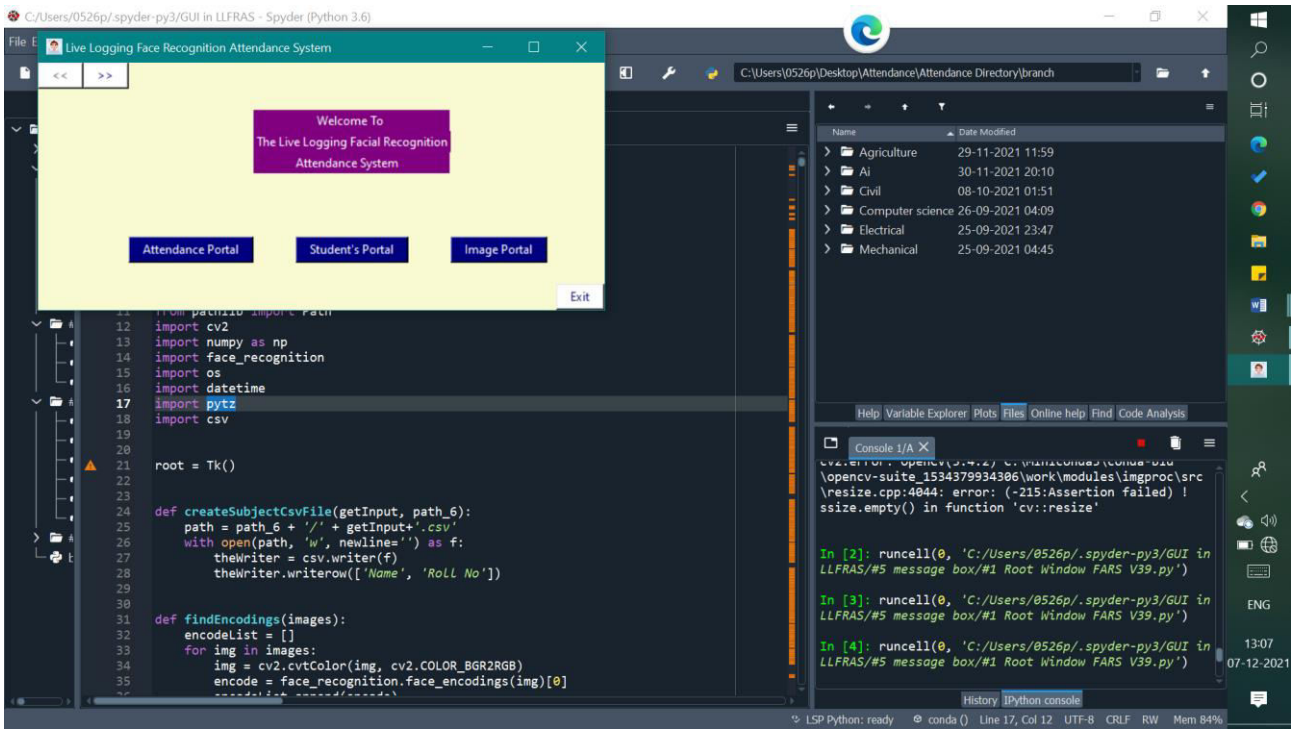

Live Logging Face Recognition Attendan...
Start
Home
Fill Student Details...
Branch
Civil
Batch
2022
Section
Semester
1
Subject
Roll No
Get
Exit
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45


Live Logging Face Recognition Attendan...
Start
Home
Fill Student Details...
Branch
Civil
Batch
2022
Section
a
Semester
Subject
Roll No
Get
Exit
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45


Live Logging Face Recognition Attendan...
Start
Home
Fill Student Details...
Branch
Civil
Batch
2022
Section
a
Semester
1
Subject
Roll No
Get
Exit
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
root = Tk()
def createSubjectCsvFile(getInput, path_6):
def findEncodings(images):


Live Logging Face Recognition Attendan...
Start
Home
Fill Student Details...
Branch
Civil
Batch
2022
Section
a
Semester
1
Subject
maths
Roll No
1816590
Get
Exit
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
root = Tk()
def createSubjectCsvFile(getInput, path_6):
def findEncodings(images):


C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Start
Home
Name
Roll No
Prince
1816590
Sl
Date Status
Time Status
1 2021-09-16 01:46
2 2021-09-17 01:45
3 2021-09-18 10:00
4 2021-09-24 05:48
5 2021-09-26 07:46
6 2021-10-08 12:34
Your Presence : 6 day out of 9 days
Attendance Percent : 66.67%
Exit
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
In [2]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [3]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [4]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')


C:/Users/0526p/.spyder-py3/GUI in LLFRAS - Spyder (Python 3.6)
Live Logging Face Recognition Attendance System
Welcome To
The Live Logging Facial Recognition
Attendance System
Attendance Portal
Student's Portal
Image Portal
Exit
import cv2
import numpy as np
import face_recognition
import os
import datetime
import pytz
import csv
root = Tk()
def createSubjectCsvFile(getInput, path_6):
path = path_6 + '/' + getInput+'.csv'
with open(path, 'w', newline='') as f:
theWriter = csv.writer(f)
theWriter.writerow(['Name', 'Roll No'])
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
Agriculture 29-11-2021 11:59
Ai 30-11-2021 20:10
Civil 08-10-2021 01:51
Computer science 26-09-2021 04:09
Electrical 25-09-2021 23:47
Mechanical 25-09-2021 04:45
In [2]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [3]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')
In [4]: runcell(0, 'C:/Users/0526p/.spyder-py3/GUI in LLFRAS/#5 message box/#1 Root Window FARS V39.py')

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

We made an AI that handles frontend and backend databases, data access controls, file handling, exception handling, Computer Vision and GUI. System controls the interface how to represent data in interfaces, thus user interact with models without any trouble and comfortably. We use Tkinter library for build interface and explain all libraries in chapter 2 which we used for building interface. Using SQL build databases of students. Computer Vision capture the video frames and compare the images form databases. If that person attendance is doesn't marks then it marks the attendance else it marks to 0 for that person. For all particular date it make a new date column where we marks attendance of that particular person. If any person is new in the class and other then it check first that person data is already present in the sheet or not. If that person data is not present in the list then it make a new row for that person and marks the attendance from that date when he presence in the class and rest of the previous date column is marked zero.

There is a three portals Attendance Portal, Student Portal, and Image Portal. In Attendance Portal we generally use these portal for marking attendance and all the process we explain in first paragraph. In Student portal we represent the data of that particular person. So, for accessing the data of particular person first we make an interface where people fill his/her basic information then after he/she will be able to get the data. The data will be like your name, roll no, date-status (all date when you will be present in the class or appear in the class), and times status (the time when you are come on that particular date). After that table it also shows your total no of date (where you are appear for the class) from the total number of date (where class will be done). In the very next row your total percentage (where you present in the class) from the total no days (where class will be happen)

In the Image Portal, we create data of new student or person so, they mark their attendance further. There is provide an interface where we enter the basic data of student or person such as for student branch, batch, section, roll no, first name, middle name, last name, phone no, email id. After that in the next row we click the button and capture the screenshot of person or students.

6.2 Future Scope

As we know few years ago biometric attendance system is launched which is very efficient, very fast, and secure method. We use this method for marking attendance using fingerprint of every single person by person. It reduces the physical effort of file handling and wasting time. It has one disadvantages which we easily mark proxy attendance of any person. System doesn't differentiate that is that person same or different. So, I build the AI which mark attendance without any proxy, and secure method by using face.

Our AI, mark attendance of person using face of particular person. For reducing the proxy attendance marking system, we use a method to track eye blinking and eye motion tracking system which differentiate that is that any image or video file, if it is then it should not mark attendance. If it is not which means person present in real time and the image capture is in real time frame. It using the face patterns and LBPH algorithm for making the face patterns and using face recognition we calculate the face distance of persons/students face. Thus we differentiate the faces between the person/students.

We use this model to everywhere in the world where we need to mark attendance of person. In/out times when he/she will enter in and when he/she will be out such as Security system, Bank security, business purposes, IT organization, School, Institute, College, Universities, and so on…

We also use this technology in online meeting system, online classes, conference, seminar, webinar, and so on.

We also uses the features of entry and exit control in this system. Time limit for marking attendance. If you are not present within this time limit then your attendance will be mark zero. If you are present within this time limit then your attendance will be mark present. It also has a features of entry control what is the time limit you are able to mark attendance. If you are enter the class or something else within this time limit then you are able to mark your attendance.

6.1 Download Anaconda –

https://www.anaconda.com/products/individual - Downloads

6.2 Learn Python –

https://github.com/PrinceKumarKeshri/Python-Basic

6.3 Tkinter documentation –

https://docs.python.org/3/library/tkinter.html

6.4 OpenCV documentation –

https://opencv.org/releases/

6.5 NumPy installation –

https://numpy.org/install/

6.6 NumPy documentation –

https://numpy.org/doc/stable/

6.7 Face Recognition –

https://github.com/ageitgey/face_recognition

6.8 OS documentation –

https://docs.python.org/3/library/os.html

6.9 DateTime documentation –

https://docs.python.org/3/library/datetime.html

6.10 Pytz installation –

https://pypi.org/project/pytz/

6.11 Pytz documentation –

https://www.askpython.com/python-modules/python-pytz-module

6.12 CSV documentation –

https://docs.python.org/3/library/csv.html

6.13 PIL documentation –

https://pillow.readthedocs.io/en/stable/

6.14 Pathlib documentation –

https://docs.python.org/3/library/pathlib.html