

ARTIFICIAL INTELLIGENCE AND TECHNIQUES LAB

CSE2022L

School of Engineering and Sciences
Department of Computer Science and Engineering

Name	Aryan Saini
Roll No	210160223018
Programme	B.Tech CSE -AI
Section/Group	C/ T4

Name	Chetanya Rajpal
Roll No	210160223007
Programme	B.Tech CSE -AI
Section/Group	C/ T4

Name	Tushar Garg
Roll No	210160223005
Programme	B.Tech CSE -AI
Section/Group	C/ T4



G D Goenka University
Gurgaon, Haryana

Table of Contents

Sr. No	Content	Page No.
1.	Scope of Project Description, Stakeholders, Existing Techniques, Scope	3
2.	Key Features	3
3.	Use Case Diagram	4
4.	API Documentation Python Installation, Packages and Installation	5
5.	Code Documentation	17
6.	Citations	23

Scope of Project

A. Detailed Description of Project

- The Attendance System takes input as Lecture name, Session type, Section, and Category from a GUI form and uses the input to open an Excel (CSV) file and a folder of the related input with identity photographs of the student in the lecture of a given input with their Enrolment No. as the name of photograph. The software also has a database of students.
- A camera window opens for face recognition with a timestamp that will be closed after 20 minutes. The face recognized by the system will be compared with the photos and the Enrolment No. of the student extracted from the photo name is stored. When the Excel file is opened it contains Enrolment No. and the Name of the students in that lecture, a new column is added with the current date.
- The stored Enrolment No. is compared with Enrolment No. written in the Excel (CSV) file, when a match is found the student is marked present with a 'P'. The window can be closed by pressing the 'c' button if one wants to close it before 20 minutes. After the window is closed, the students that are unmarked are marked absent with 'A', then the file is saved and closed for later use.

B. Scope of Project

- The Attendance System can be used by faculties and teachers for the ease of marking the attendance of students. It reduces the faculty's time and effort in marking attendance.
- The files can be updated manually later if needed, introducing new features such as calculating a student attendance percentage or percentage of students present in a class.
- A new subject can be added when needed by the management. New student records can be added for new admission or change of course.
- If needed, an entry time feature can also be added to mark every student's entry timestamp.

C. Details of Stakeholders

- The stakeholders can be the various faculties and administrative members in charge of student attendance.

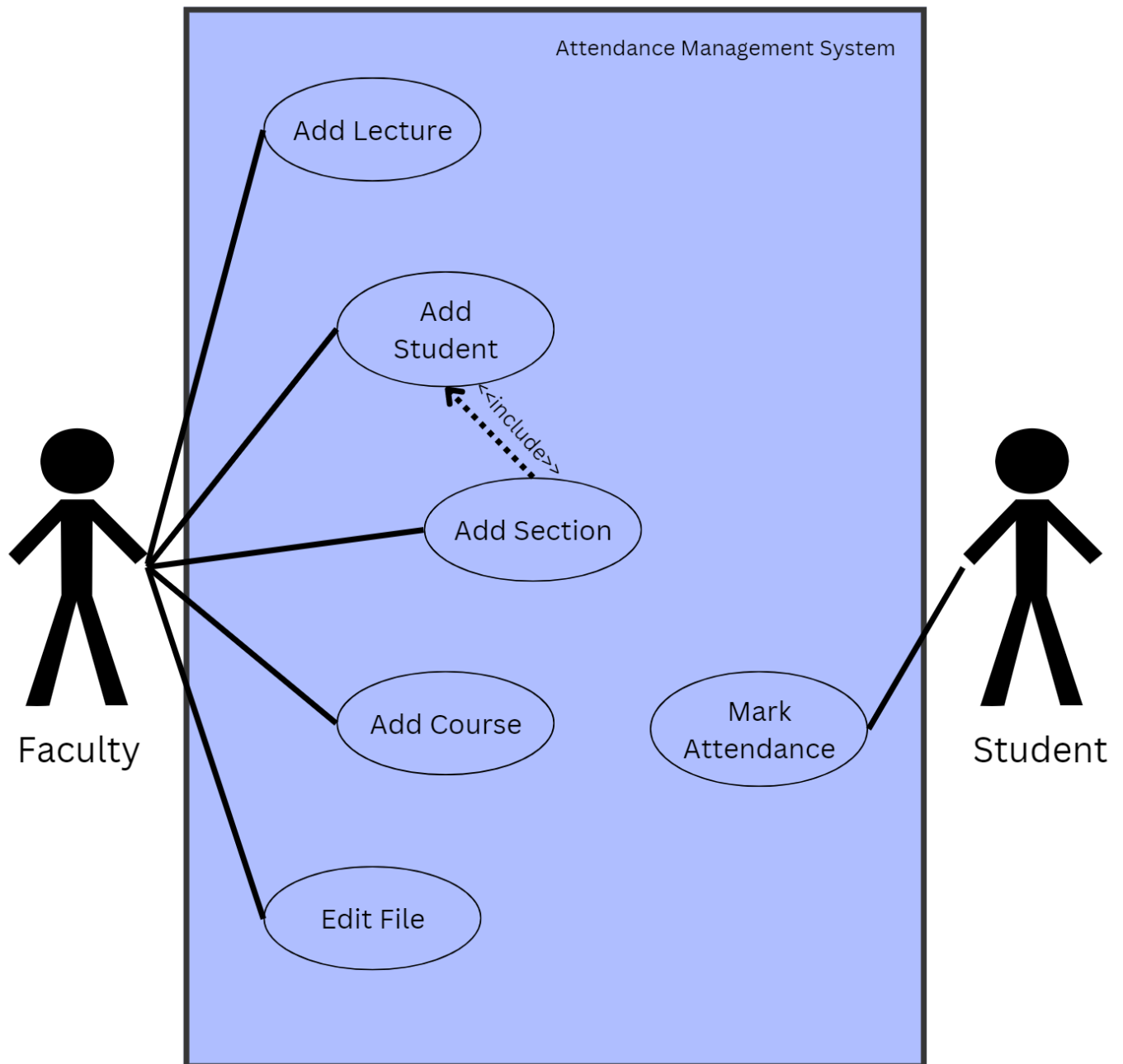
D. Existing techniques in the area

- Attendance management with a Biometric system and ID card scan system is already implemented in various colleges, schools, and offices.
- Attendance management system using face recognition is less found in practical application.

Key Features

- It uses and maintains a database of student details. With the help of this, the program can be further developed giving the user choice of adding new courses, viewing existing records, etc.
- The program takes the input of one student only once to avoid repetitive marking of students. The first entry will be considered final.
- The program also allows you to customize the time of closing the camera for attendance as per your preference. The default time set for closing is 20 minutes after the camera starts.
- GUI Interface for taking input from the faculty, it makes it easy for the user to give inputs and gives ease of use.

Use Case Diagram



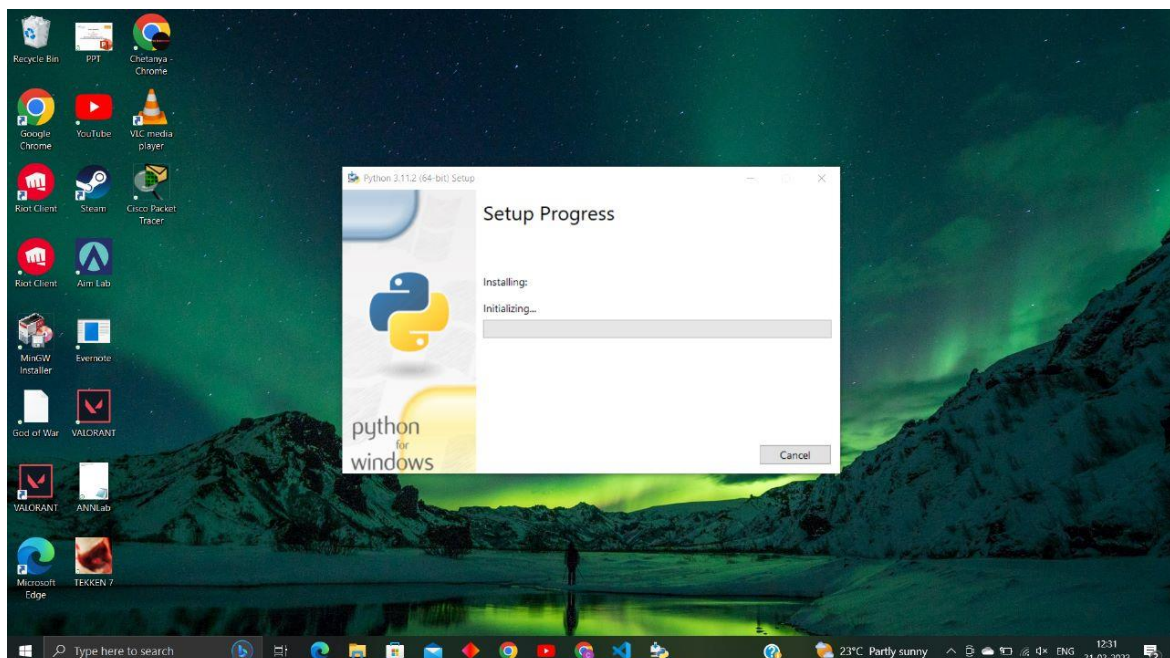
API Documentation

Installations

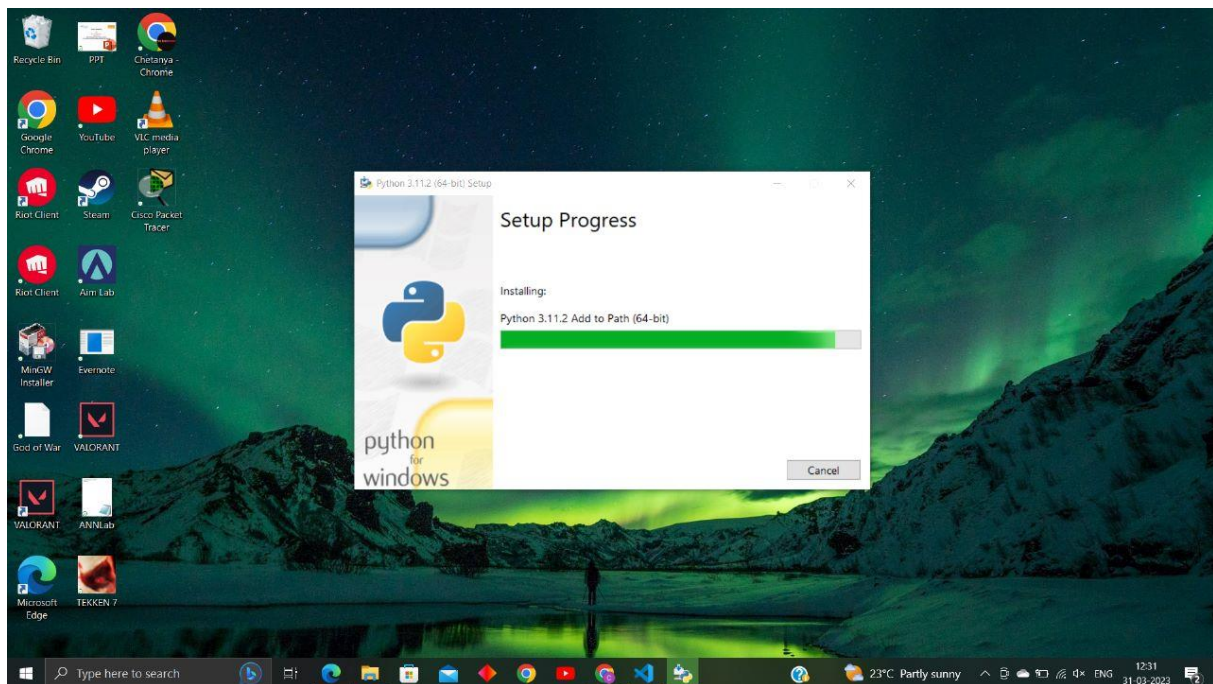
- Python Installation

Use this link to download Python: <https://www.python.org/downloads/>

1. Open the downloaded .exe file and run the set-up file.
2. Check the checkboxes and click on Install Now or click on customize installation for customization.

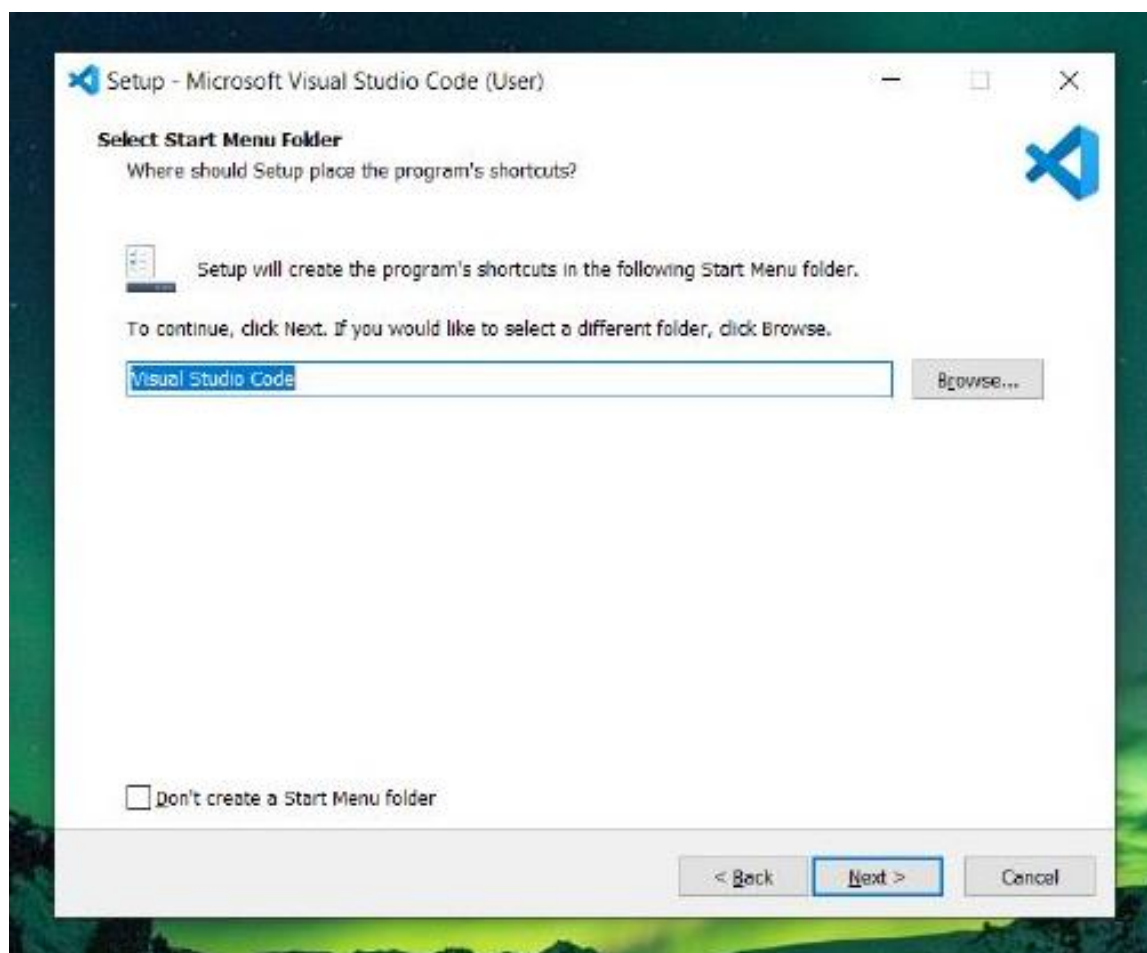


3. Let the installation progress bar finish and then click on finish button.

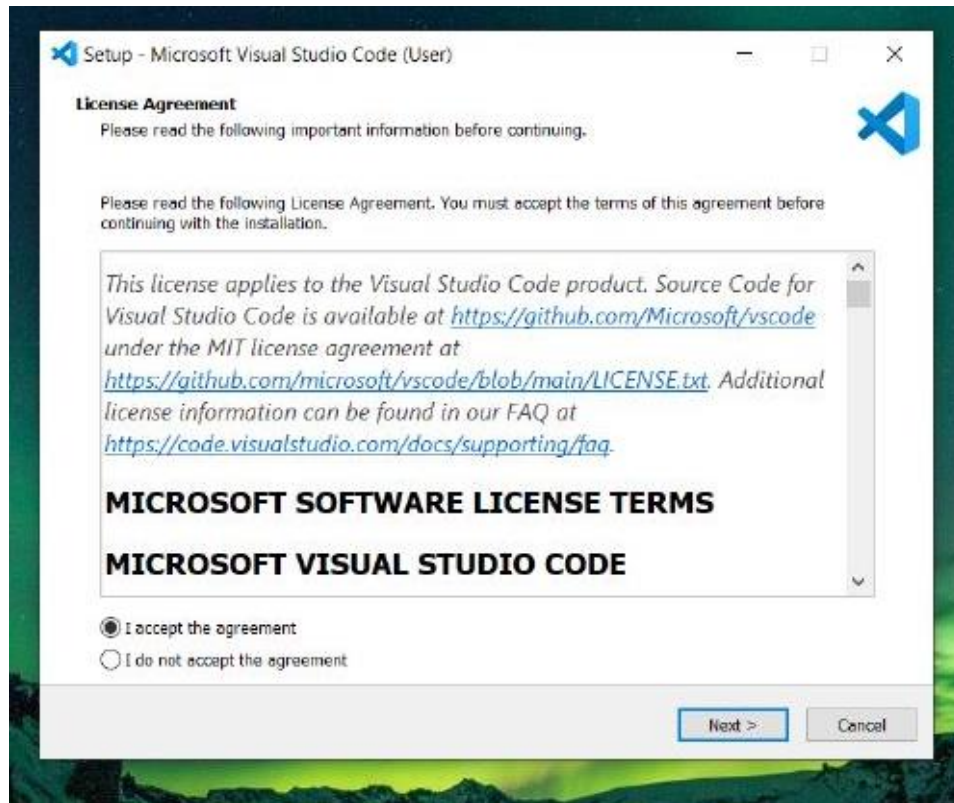


- Visual Studio Code Installation

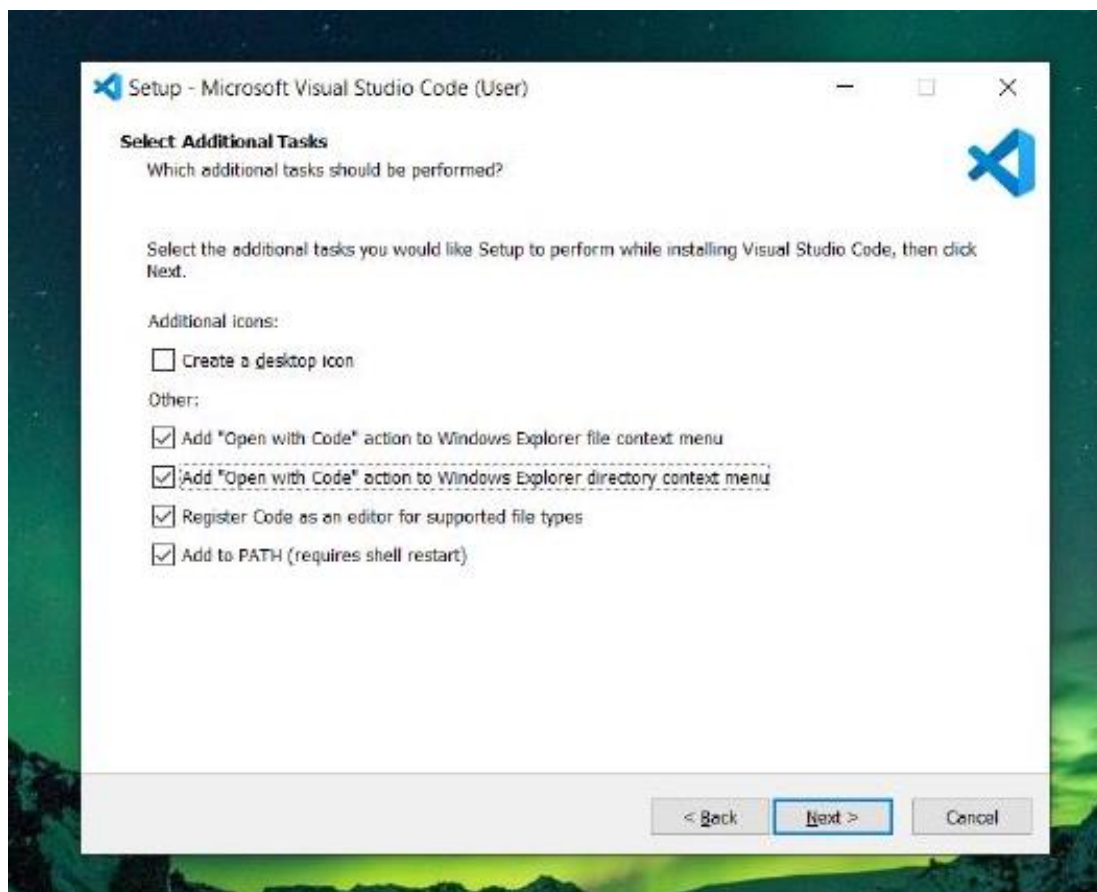
1. Choose the folder you want to select and name it. Click on Next.



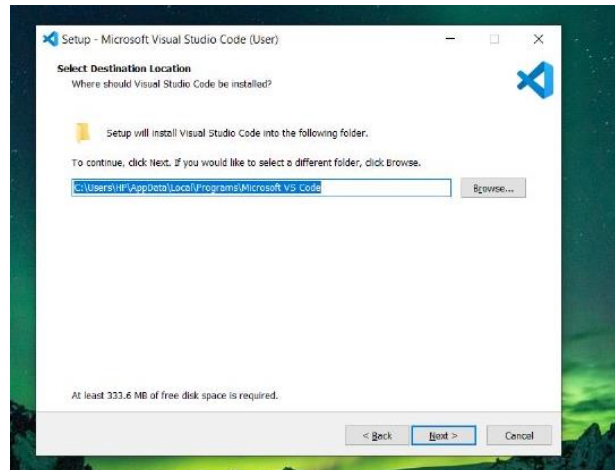
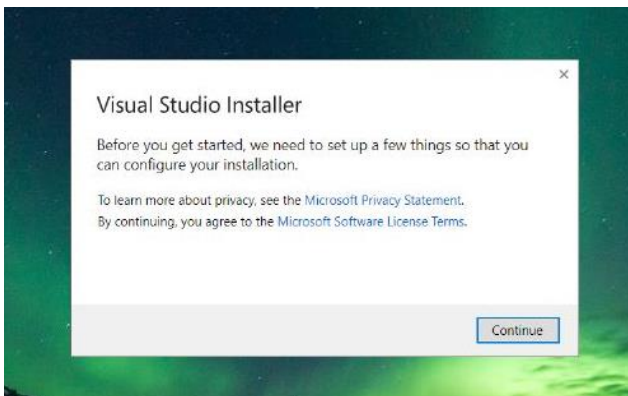
2. Accept the License agreement and click on Next.



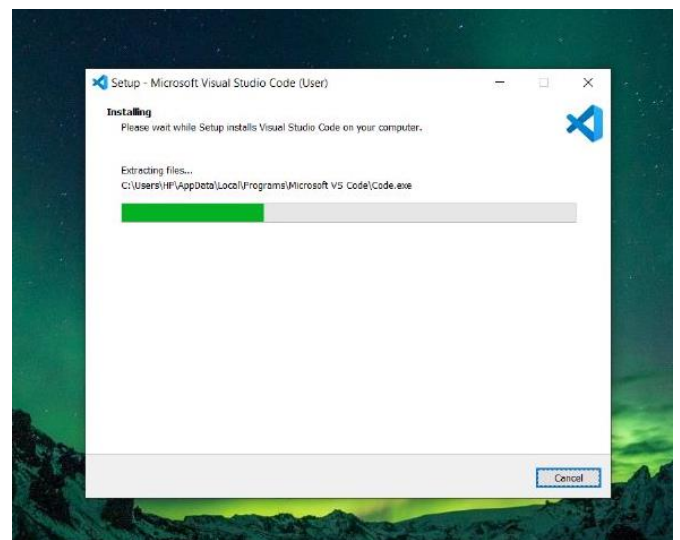
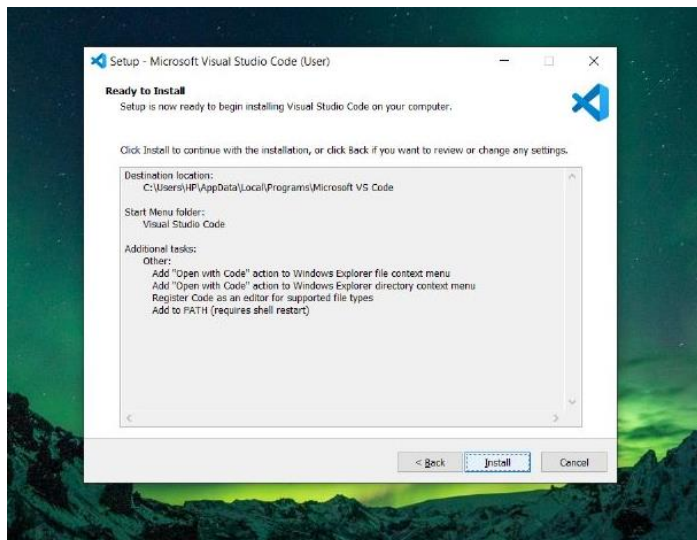
3. Choose the additional Options according to your preference and click on Next.



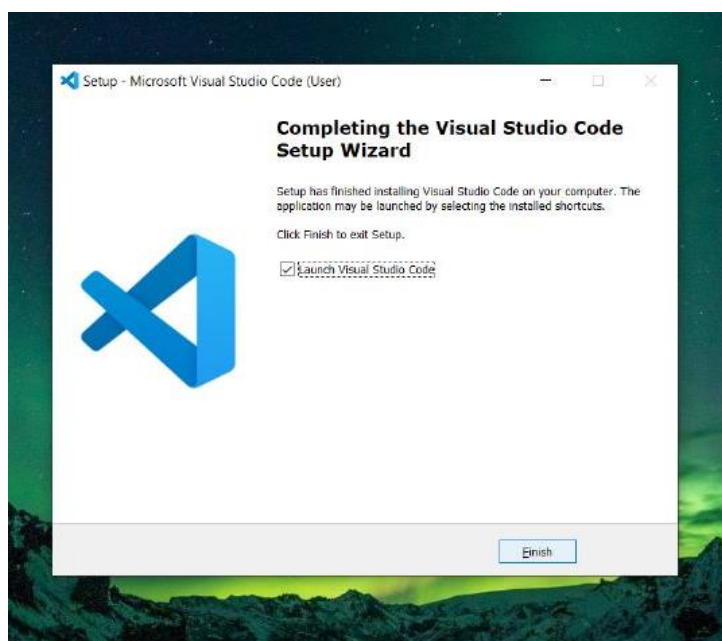
4. Click on continue, select the location of the folder for installation and click on Next.



5. Check the selected options if they are correct and click on Install.

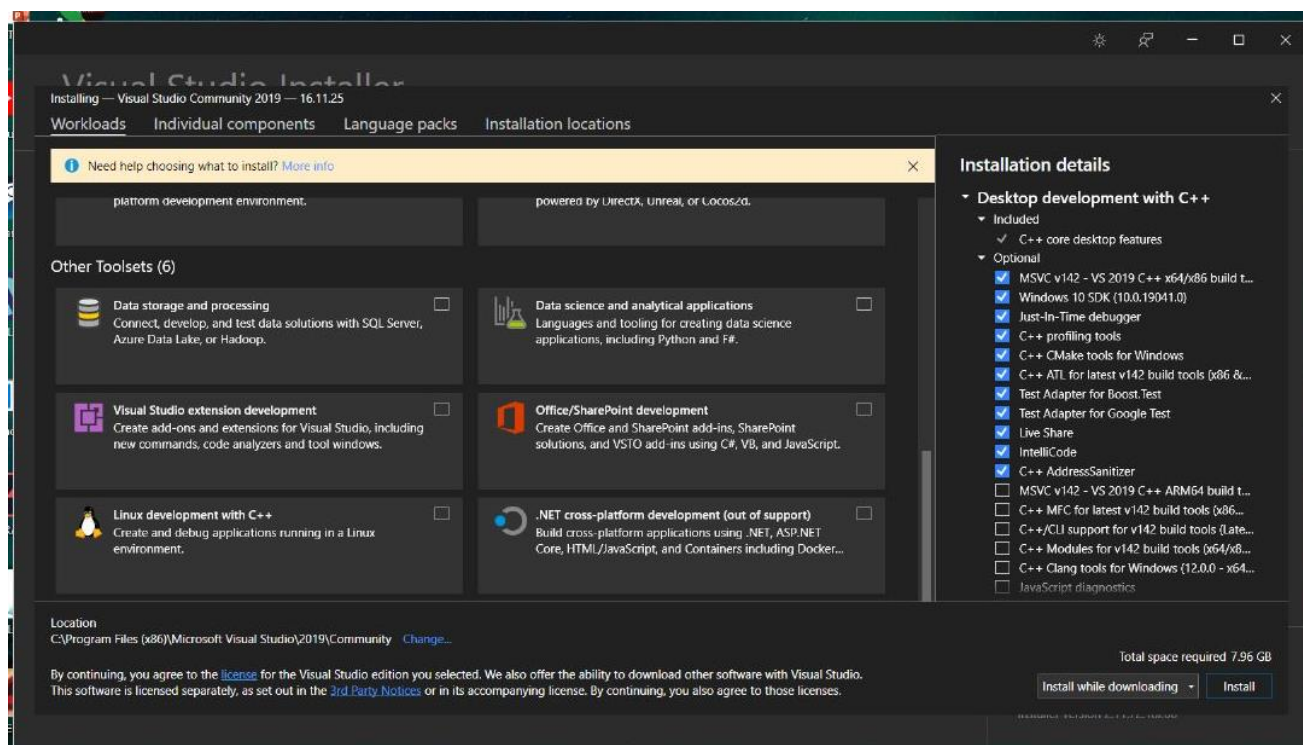


6. Click on finish to complete the process.

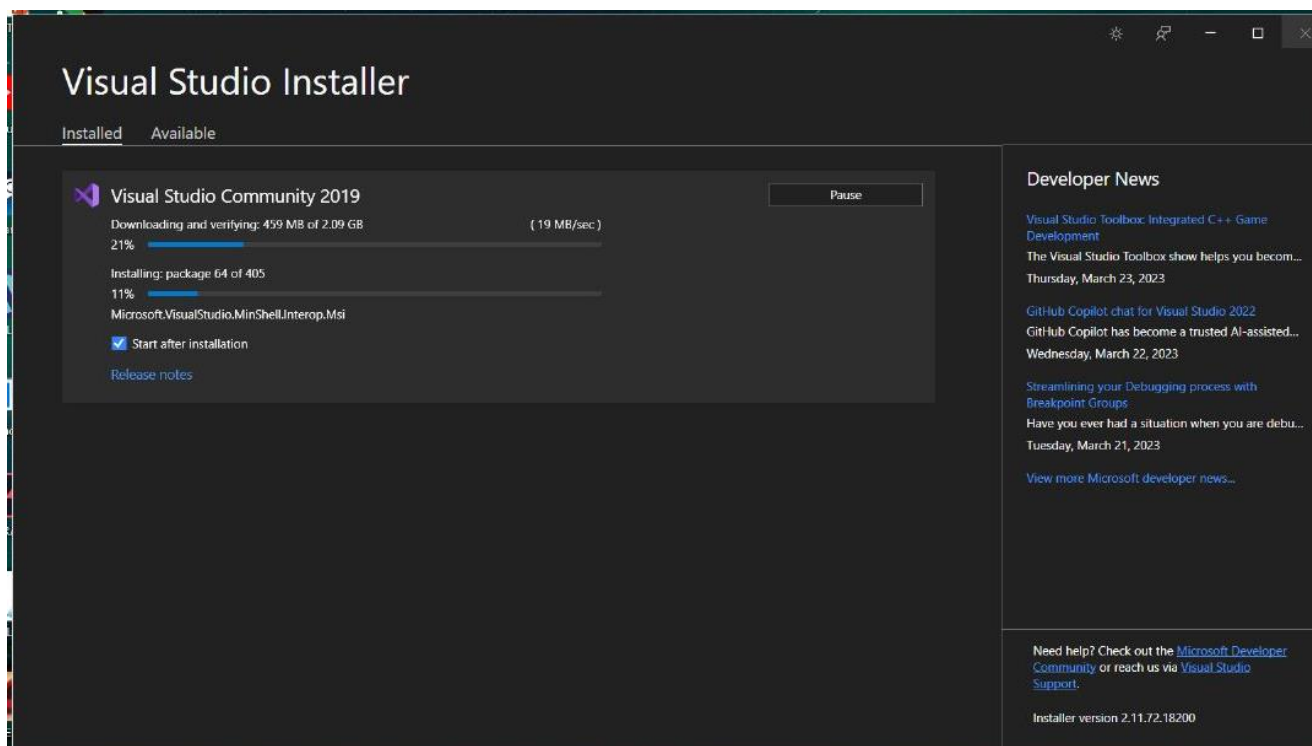


- Connecting Python and VS Code

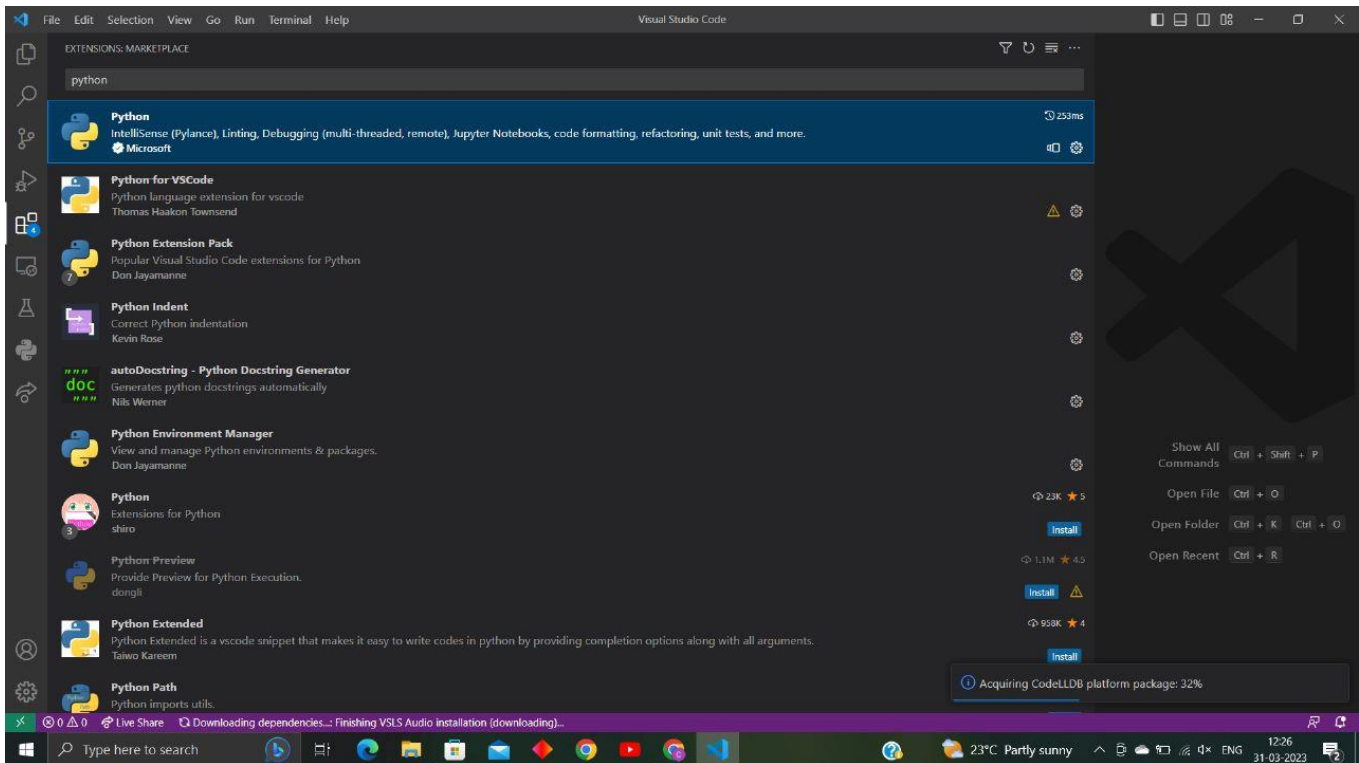
1. Choose the languages and tools required or shown in the image below and click on install or install while downloading.



2. It should start downloading and installing as shown in the image below.



3. After installation, go to the extension shortcut and search python. Click on the selected extension and install it. Now your VS Code supports Python.



Packages Required and their installations

1. NumPy: It can be used to perform a wide variety of mathematical operations on arrays. It forms the foundation of the Machine Learning stack.

Command: `pip install numpy`

```
C:\Users\DELL>pip install numpy
Requirement already satisfied: numpy in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (1.23.3)
```

2. Dlib: It is a landmark's facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person's face.

Command: `pip install dlib`

```
C:\Users\DELL>pip install dlib
Requirement already satisfied: dlib in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (19.24.0)
```

3. CMake: CMake provides many benefits for single-platform, multi-machine development environments including: The ability to automatically search for programs, libraries, and header files that may be required by the software being built.

Command: `pip install cmake`

```
C:\Users\DELL>pip install cmake
Requirement already satisfied: cmake in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (3.25.2)
```

4. Face_Recognition: It lets you recognize faces from a photograph or a folder full of photographs. There's one line in the output for each face. The data is comma-separated with the filename and the name of the person found.

Command: `pip install face_recognition`

```
PS C:\Users\HP> pip install face_recognition
Collecting face_recognition
  Using cached face_recognition-1.3.0-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: face-recognition-models>=0.3.0 in c:\users\hp\appdata\local\prog
Collecting Click>=6.0
  Using cached click-8.1.3-py3-none-any.whl (96 kB)
Requirement already satisfied: dlib>=19.7 in c:\users\hp\appdata\local\programs\python\python31
Requirement already satisfied: numpy in c:\users\hp\appdata\local\programs\python\python311\lib
Collecting Pillow
  Using cached Pillow-9.4.0-cp311-cp311-win_amd64.whl (2.5 MB)
Collecting colorama
  Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: Pillow, colorama, Click, face_recognition
Successfully installed Click-8.1.3 Pillow-9.4.0 colorama-0.4.6 face_recognition-1.3.0
```

5. Glob: The glob module, which is short for global, is a function that's used to search for files that match a specific file pattern or name. It can be used to search CSV files and for text in files.

Command: `pip install glob2`

```
C:\Users\DELL>pip install glob2
Collecting glob2
  Downloading glob2-0.7.tar.gz (10 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: glob2
  Building wheel for glob2 (pyproject.toml) ... done
  Created wheel for glob2: filename=glob2-0.7-py2.py3-none-any.whl
size=9310 sha256=558a6a7a78c489e88a2dc7dd9187f5e1bbc18eeddd53cfc2ef
65866b0e71c1d3
  Stored in directory: c:\users\dell\appdata\local\pip\cache\wheels
\37\07\ce\cbe8d31ad93224571b49fa03f8a5da11cdb31d3845ff73e0f3
Successfully built glob2
Installing collected packages: glob2
Successfully installed glob2-0.7
```

6. MySQL Connector: MySQL Connectors provide connectivity to the MySQL server for client programs. APIs provide low-level access to MySQL resources using either the classic MySQL protocol or X Protocol.^[2]

Command: `pip install mysql-connector-python`

```
C:\Users\DELL>pip install mysql-connector-python
Requirement already satisfied: mysql-connector-python in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (8.0.32)
Requirement already satisfied: protobuf<=3.20.3,>=3.11.0 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from mysql-connector-python) (3.20.3)
```

7. Pandas: It provides fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

Command: `pip install pandas`

```
C:\Users\DELL>pip install pandas
Requirement already satisfied: pandas in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from pandas) (2022.7.1)
Requirement already satisfied: numpy>=1.21.0 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from pandas) (1.23.3)
Requirement already satisfied: six>=1.5 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```


8. Open CV: It is used to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products.^[1]

Command: `pip install opencv-python`

```
C:\Users\DELL>pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (4.7.0.72)
Requirement already satisfied: numpy>=1.21.2 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from opencv-python) (1.23.3)
```

9. Tkinter: It is used to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions.

Command: `pip install tk`

```
C:\Users\DELL>pip install tk
Collecting tk
  Downloading tk-0.1.0-py3-none-any.whl (3.9 kB)
Installing collected packages: tk
Successfully installed tk-0.1.0
```

10. Path: It is used to read and write files. Carefully copy, move, and delete files. Manipulate paths and the underlying file system. Pick out components of a path.

Command: `pip install path`

```
C:\Users\DELL>pip install path
Collecting path
  Downloading path-16.6.0-py3-none-any.whl (26 kB)
Installing collected packages: path
Successfully installed path-16.6.0
```

11. Playsound: It plays a sound specified by the given file name, resource, or system event.

Command: `pip install playsound`

```
C:\Users\DELL>pip install playsound
Requirement already satisfied: playsound in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (1.3.0)
```

12. CSV: It allows for the data table to be easily retrieved into a variety of applications, they are best viewed within one that will allow one to easily manipulate data that is in columnar format.^[3]

Command: `pip install python-csv`

```
C:\Users\DELL>pip install python-csv
Collecting python-csv
  Downloading python-csv-0.0.13.tar.gz (26 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting argparse (from python-csv)
  Downloading argparse-1.4.0-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: numpy in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from python-csv) (1.23.3)
Requirement already satisfied: pandas in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from python-csv) (1.5.3)
Requirement already satisfied: matplotlib in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from python-csv) (3.6.3)
```

13. OS: It provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about operating system.

Command: `pip install os-sys`

```
C:\Users\DELL>pip install os-sys
Collecting os-sys
  Downloading os_sys-2.1.4-py3-none-any.whl (15.6 MB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15.6/15.6 MB 13.1 MB/s eta 0:00:00
Collecting pygubu (from os-sys)
  Downloading pygubu-0.30-py3-none-any.whl (118 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 118.6/118.6 kB 7.2 MB/s eta 0:00:00
Requirement already satisfied: pytz in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from os-sys) (2022.7.1)
Collecting sqlparse (from os-sys)
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.2/41.2 kB ? eta 0:00:00
Collecting progress (from os-sys)
  Downloading progress-1.6.tar.gz (7.8 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting tqdm (from os-sys)
  Downloading tqdm-4.65.0-py3-none-any.whl (77 kB)
```

14. Datetime: Python's built-in datetime library is one of the most common modules to manipulate date and time object data.

Command: `pip install datetime`

```
C:\Users\DELL>pip install datetime
Collecting datetime
  Downloading DateTime-5.1-py3-none-any.whl (52 kB)
    52.1/52.1 kB ? eta 0:00:00
Collecting zope.interface (from datetime)
  Downloading zope.interface-6.0-cp310-cp310-win_amd64.whl (204 kB)
    204.1/204.1 kB 12.9 MB/s eta 0:00:00
Requirement already satisfied: pytz in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from datetime) (2022.7.1)
Requirement already satisfied: setuptools in c:\program files\windowsapps\pythonsoftwarefoundation.python.3.10_3.10.3056.0_x64__qbz5n2kfra8p0\lib\site-packages (from zope.interface->datetime) (65.5.0)
Installing collected packages: zope.interface, datetime
Successfully installed datetime-5.1 zope.interface-6.0
```

15. Time: It provides many ways of representing time in code, such as objects, numbers, and strings. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.

Command: `pip install times`

```
C:\Users\DELL>pip install times
Collecting times
  Downloading times-0.7-py2.py3-none-any.whl (3.8 kB)
Collecting arrow (from times)
  Downloading arrow-1.2.3-py3-none-any.whl (66 kB)
    66.4/66.4 kB 3.5 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.7.0 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from arrow->times) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from python-dateutil>=2.7.0->arrow->times) (1.16.0)
Installing collected packages: arrow, times
Successfully installed arrow-1.2.3 times-0.7
```

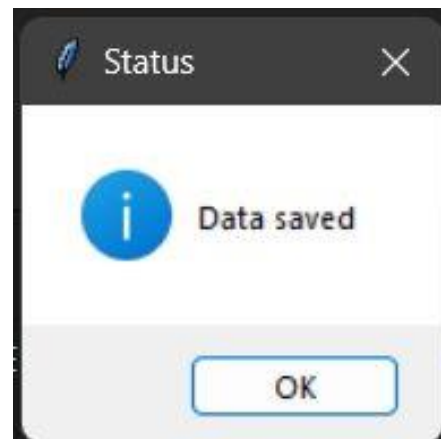
Code Documentation

Importing all the necessary libraries and modules

```
#importing related libraries
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
import face_recognition
import numpy as np
import cv2
import csv
import os
from datetime import datetime
import glob
import time
from pathlib import Path
from playsound import playsound
import mysql.connector
import pandas as pd
```

This is a function that is called on the press of the submit button of the form and the inputs are stored in a list variable. It also displays a message box showing that data saved successfully. The inputs are stored in an array for further use. The image shows the output when the function is called.

```
data_1=[]
def submit_data(data_1):
    messagebox.showinfo('Status','Data saved')
    category=''
    session_type=''
    section=''
    subject=sub_entry.get()
    if session_entry.get()==1:
        session_type='LAB'
    elif session_entry.get()==2:
        session_type='THEORY'
    if section_entry.get()==1:
        section='A'
    elif section_entry.get()==2:
        section='C'
    elif section_entry.get()==3:
        section='BOTH'
    if category1.get()==1:
        category+='T1'
    if category2.get()==1:
        category+='T2'
    if category3.get()==1:
        category+='T3'
    if category4.get()==1:
        category+='T4'
    data=[]
    data.append(subject)
    data.append(session_type)
    data.append(section)
    data.append(category)
    for x in data:
        data_1.append(x)
```



This is the code for the frame window that takes input of the Subject, Session type, Section, and Category. A dropdown menu or combobox is used for the Subject and a list is used to display the subject options in the menu. A radio button is used for the input of session type and section. A radio button helps us to select only one true value from multiple options. A checkbox is used for input of Category. A checkbox enables us to take more than one input from multiple options. Grid() function is used to place the labels and input area in the frame window.

```

window = Tk()
window.title('Lecture Information')
frame=Frame(window)
frame.pack()
main_frame=LabelFrame(frame,text='')
main_frame.grid(row=0,column=0,sticky='news',padx=20,pady=30)
#Subject Input
sub_lb=Label(main_frame,text='SUBJECT',font=('Cambria',17,'bold'))
sub_lb.grid(row=0,column=0,padx=20,pady=20)
sub_options=['CHE1003(Bio-sciences_Nanotechnology)',
             'CSE2022(Artificial Intelligence Techinques and Methods)',
             'CSE2101(Artificial Neural Network)',
             'CSE2102(Computer Network)',
             'CSE2103(Software Engineering and Project Management)',
             'CSE2718(Computer Architecture)']
sub_entry=ttk.Combobox(main_frame,values=sub_options,width=50,height=25)
sub_entry.grid(row=0,column=1,sticky='w',padx=20,pady=20)
#Session type input
session_lb=Label(main_frame,text='SESSION',font=('Cambria',17,'bold'))
session_lb.grid(row=1,column=0,padx=20,pady=20)
session_entry=IntVar()
session_op1=Radiobutton(main_frame,text='LAB',variable=session_entry,value=1,font=('Courier',15))
session_op2=Radiobutton(main_frame,text='THEORY',variable=session_entry,value=2,font=('Courier',15))
session_op1.grid(row=1,column=1,sticky='w',padx=10)
session_op2.grid(row=1,column=1,padx=10)

#Section Input
section_lb=Label(main_frame,text='SECTION',font=('Cambria',17,'bold'))
section_lb.grid(row=2,column=0,padx=20,pady=20)
section_entry=IntVar()
section_op1=Radiobutton(main_frame,text='A',variable=section_entry,value=1,font=('Courier',15))
section_op2=Radiobutton(main_frame,text='C',variable=section_entry,value=2,font=('Courier',15))
section_op3=Radiobutton(main_frame,text='BOTH',variable=section_entry,value=3,font=('Courier',15))
section_op1.grid(row=2,column=1,sticky='w',padx=10)
section_op2.grid(row=2,column=1,padx=10)
section_op3.grid(row=2,column=1,sticky='e',padx=10)

#Category Input
category_lb=Label(main_frame,text='CATEGORY',font=('Cambria',17,'bold'))
category_lb.grid(row=3,column=0,padx=20,pady=20)
category1 = IntVar()
category2 = IntVar()
category3 = IntVar()
category4 = IntVar()
c1 = Checkbutton(main_frame, text = "T1",
                 variable = category1,
                 onvalue = 1,
                 offvalue = 0,
                 height = 2,
                 width = 1,
                 font=('Courier',15))

```



```

c2 = Checkbutton(main_frame, text = "T2",
                 variable = category2,
                 onvalue = 1,
                 offvalue = 0,
                 height = 2,
                 width = 1,
                 font=('Courier',15))

c3 = Checkbutton(main_frame, text = "T3",
                 variable = category3,
                 onvalue = 1,
                 offvalue = 0,
                 height = 2,
                 width = 1,
                 font=('Courier',15))

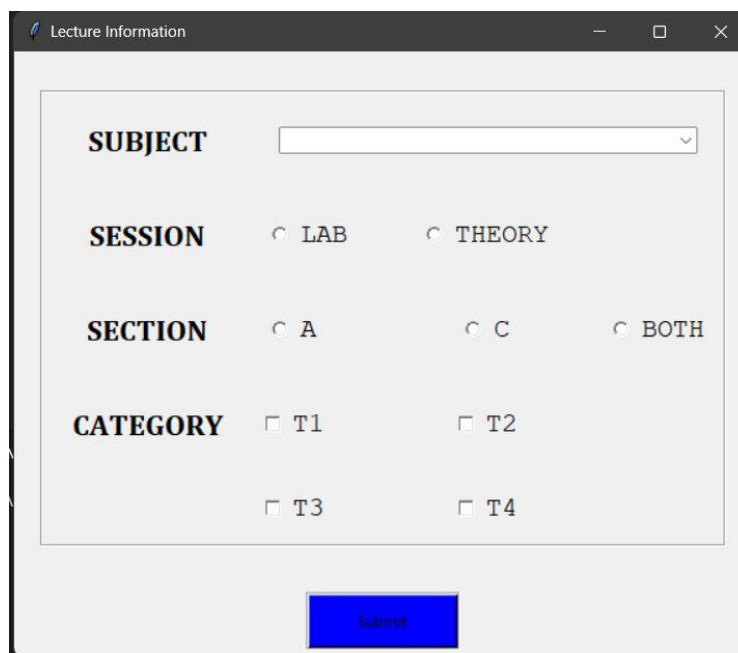
c4 = Checkbutton(main_frame, text = "T4",
                 variable = category4,
                 onvalue = 1,
                 offvalue = 0,
                 height = 2,
                 width = 1,
                 font=('Courier',15))
c1.grid(row=3,column=1,padx=10,sticky='w')
c2.grid(row=3,column=1,padx=10)
c3.grid(row=4,column=1,padx=10,sticky='w')
c4.grid(row=4,column=1,padx=10)

#submit button
data_1=[]
main_frame2=LabelFrame(frame,text='')
main_frame2.grid(row=5,column=0,padx=20,pady=5)
submit=Button(main_frame2,text='Submit',width=15,height=2,bg='blue',activebackground='green',command=Lambda:submit_data(data_1))
submit.grid(row=5,column=0)

window.mainloop()

```

Below is the output of the basic GUI interface when the window frame is called by `window.mainloop()`. The interface design can be modified as per the need of the user.



Declaring list variables for storing the known enrollments and know face encodings of the images stored in the database file and establishing MySQL connection for updating information from open-cv to database. The database will be used to update the CSV files later. Store the input value from the frame window in the variable and extract the subject code from the Subject input. Creating variables that store the file path of the images stored in file.

```
directory=[]
images=[]
known_encoding=[]
all_enrollment_numbers=[]
n=0

#Establishing database connection
sql=mysql.connector.connect(host="localhost",user="root",password="root")
cursor=sql.cursor()
query="use attendance_db"
cursor.execute(query)

subject=data_1[0]
session_type=data_1[1]
section=data_1[2]
category=data_1[3]
subject_code,subject_name=subject.split("(")

# Providing file path of the images
filePathAT1='D:/Users/DELL/Desktop/python ai class/face rec/photo/Section A/T1'
filePathAT2='D:/Users/DELL/Desktop/python ai class/face rec/photo/Section A/T2'
filePathCT3='D:/Users/DELL/Desktop/python ai class/face rec/photo/Section C/T3'
filePathCT4='D:/Users/DELL/Desktop/python ai class/face rec/photo/Section C/T4'
```

Fetching the CSV file based on the given input and adding a column to the existing file by the name of the current date. By default, all students are marked absent. Create list variables to store the encodings of the faces located by the face recognition system. Store the time at which the camera window starts. This variable will be used to close the camera window after 20 minutes to end the attendance session.

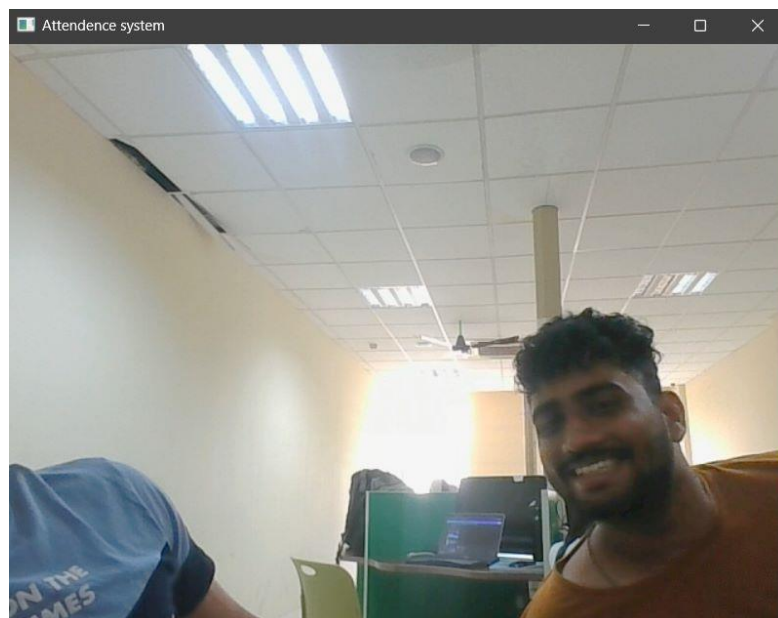
```
# Creating csv file
now = datetime.now()
current_date = now.strftime("%d-%m-%Y")
# current_date="17-03-2023"
if session_type=="THEORY":
    record_path="D:/Users/DELL/Desktop/python ai class/face rec/records/"+subject_code+"/"+session_type+"/"+section+".csv"
elif session_type=="LAB":
    record_path="D:/Users/DELL/Desktop/python ai class/face rec/records/"+subject_code+"/"+session_type+"/"+category+".csv"
df=pd.read_csv(record_path)
total_columns=len(df.axes[1])
df = pd.read_csv(record_path)
df[current_date]="A"
df.to_csv(record_path,index=False)
df = pd.read_csv(record_path)

# Creating variables for face recognition
face_locations = []
face_encodings = []
s=True
marked=[]
start_time=time.time()
```

The camera window opens for face recognition. The face located are encoded and their best matches are found in the database. The student recognized are marked present in the CSV file and their enrollments are stored in a list so that they don't get marked again and again for a particular session. When a student is recognized, an output is generated saying that the student is marked and a sound is played. The window closes automatically after 20 minutes. It can also be closed by pressing key 'c'.

```
# Initialising face recognition
cap = cv2.VideoCapture(0)
while True:
    _, frame = cap.read()
    small_frame = cv2.resize(frame, (0,0), fx=0.25, fy=0.25)
    rgb_small_frame = small_frame[:, :, ::-1]
    if s:
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
        for face_encoding in face_encodings:
            matches = face_recognition.compare_faces(known_encoding, face_encoding)
            enrol_num = ""
            face_distance = face_recognition.face_distance(known_encoding, face_encoding)
            best_match_index = np.argmin(face_distance)
            if matches[best_match_index]:
                enrol_num = all_enrollment_numbers[best_match_index]
            if enrol_num in all_enrollment_numbers:
                if enrol_num in marked:
                    continue
                else:
                    now = datetime.now()
                    font = cv2.FONT_HERSHEY_SIMPLEX
                    org = (50, 50)
                    fontScale = 1 # setting attributes for the text
                    fontColor = (0, 255, 0)
                    thickness = 3
                    lineType = 2
                    cv2.putText(frame, 'Marked',
                                org,
                                font,
                                fontScale,
                                fontColor,
                                thickness,
                                lineType) # writing text on opencv window
                    playsound("D:/Users/DELL/Desktop/python ai class/face rec/message-incoming-132126.mp3")
                    current_time = now.strftime("%H:%M:%S") # set current time
                    print(enrol_num, "is marked present")
```

The image below shows the face recognition camera window when the video capture function is called. The dimension and properties of window are set by using functions of Open-CV.



The image below shows the CSV file after the attendance is marked. It can be observed that the students with 'P' are the one who were detected by the face recognition system.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
39	'210160205007	Ritik Baghel	A											
40	'210160223001	Shruti Jaiswal	A											
41	'210160223002	Himanshu Singh	A											
42	'210160223003	Bandaru Hema	A											
43	'210160223005	Tushar Garg	A											
44	'210160223007	Chetanya Rajpal	P											
45	'210160223008	Mridul Chopra	A											
46	'210160223009	Neha	A											
47	'210160223010	Shobhit Seth	A											
48	'210160223011	Ivyanshu Bhardwaj	A											
49	'210160223012	Ugraj Rundhanke	A											
50	'210160223014	Shubham	P											
51	'210160223016	Janhavi Chauhan	A											
52	'210160223017	Vinay	A											
53	'210160223018	Aryan Saini	P											
54	'210160223019	Anshuta Jain	A											
55	'210160223020	Lakshita Raghav	A											
56	'210160223021	Kalpna Joshi	A											
57	'210160223022	Aftab Alam	A											
58	'210160223023	Lakshya	A											

```

print("Entry time:",current_time)
if session_type=="LAB":
    query="Select enrollment, name, entry_time from students where category='"+category+"'"
elif session_type=="THEORY":
    if section=="BOTH":
        query="select enrollment, name, entry_time from students"
    elif section=="A" or section=="C":
        query="Select enrollment, name, entry_time from students where section='"+section+"'"
cursor.execute(query)
result=cursor.fetchall()
a=0
for i in result:
    if i[0]==enrol_num:
        df.loc[a, current_date] = "P"
        df.to_csv(record_path, index=False)
        a+=1
marked.append(enrol_num)
# closes the program after 20 minutes
if (time.time()-start_time)>1200:
    print("Times up!, no attendance will be marked now")
    exit(0)

# Naming the openCV window
cv2.imshow("Attendance system",frame)
if cv2.waitKey(1) == ord('c'):# window closes when "c" is pressed
    break

cap.release()
cv2.destroyAllWindows()

```

This is the console output generated when a student is recognized along with their entry time.cn

```

210160223018 is marked present
Entry time: 12:38:54
210160223007 is marked present
Entry time: 12:39:18
PS C:\NEW FOLDER\CODE\artificial intelligence\Attendance_system>

```

Citations

[1] OpenCV - Overview - GeeksforGeeks. (2019, September 23). GeeksforGeeks.

<https://www.geeksforgeeks.org/opencv-overview/>

[2] MySQL :: MySQL Connector/Python Developer Guide. (n.d.). MySQL :: MySQL Connector/Python Developer Guide.

<https://dev.mysql.com/doc/connector-python/en/>

[3] Bonthu, H. (2021, August 21). How to Read and Write With CSV Files in Python? Analytics Vidhya.

<https://www.analyticsvidhya.com/blog/2021/08/python-tutorial-working-with-csv-file-for-data-science/>