Table of Contents

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, introduce 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 a feature of entry time can be also added to mark the entry timestamp of every student.

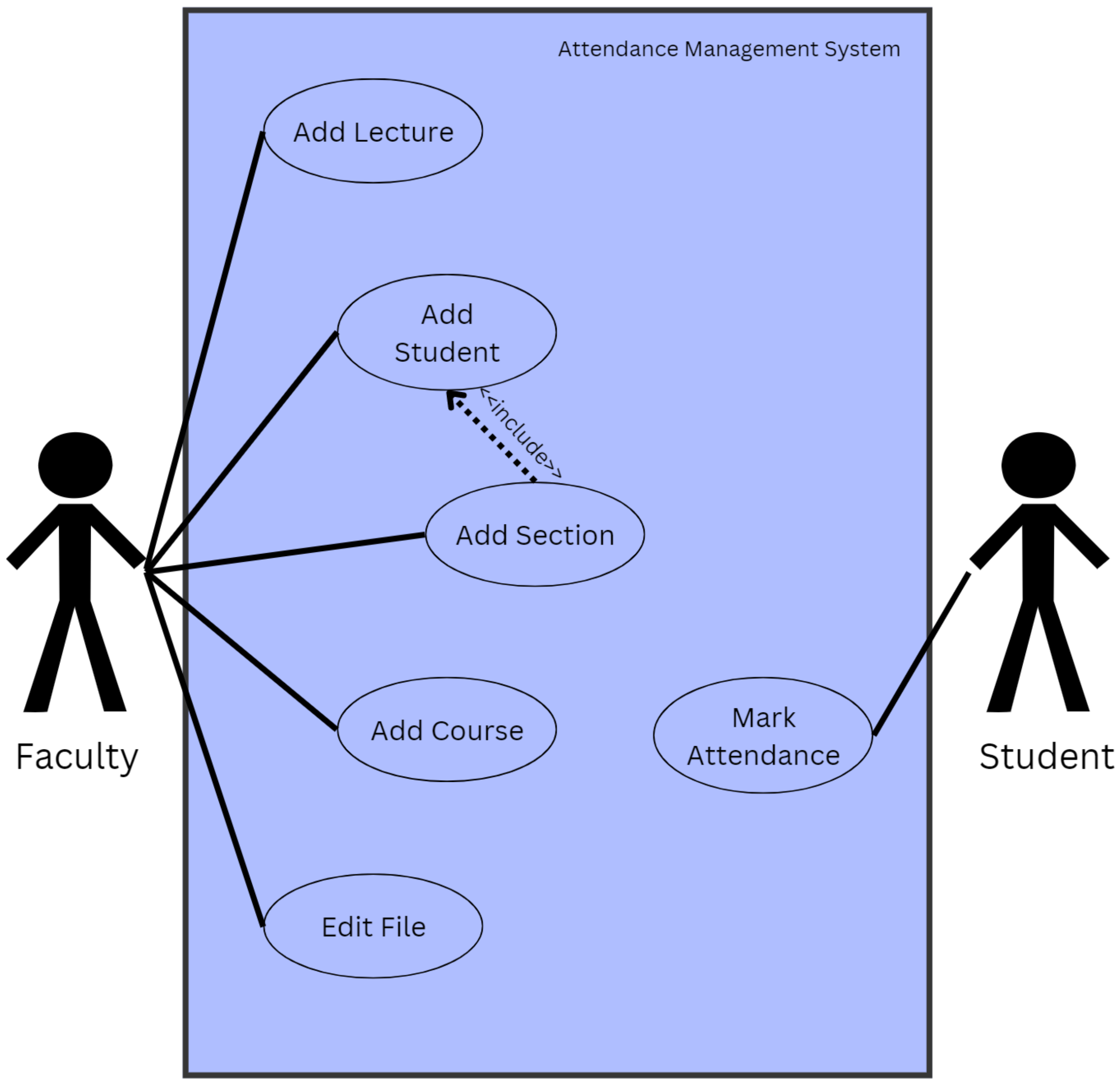
C. Details of Stakeholders

- 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.

Use Case Diagram



API Documentation

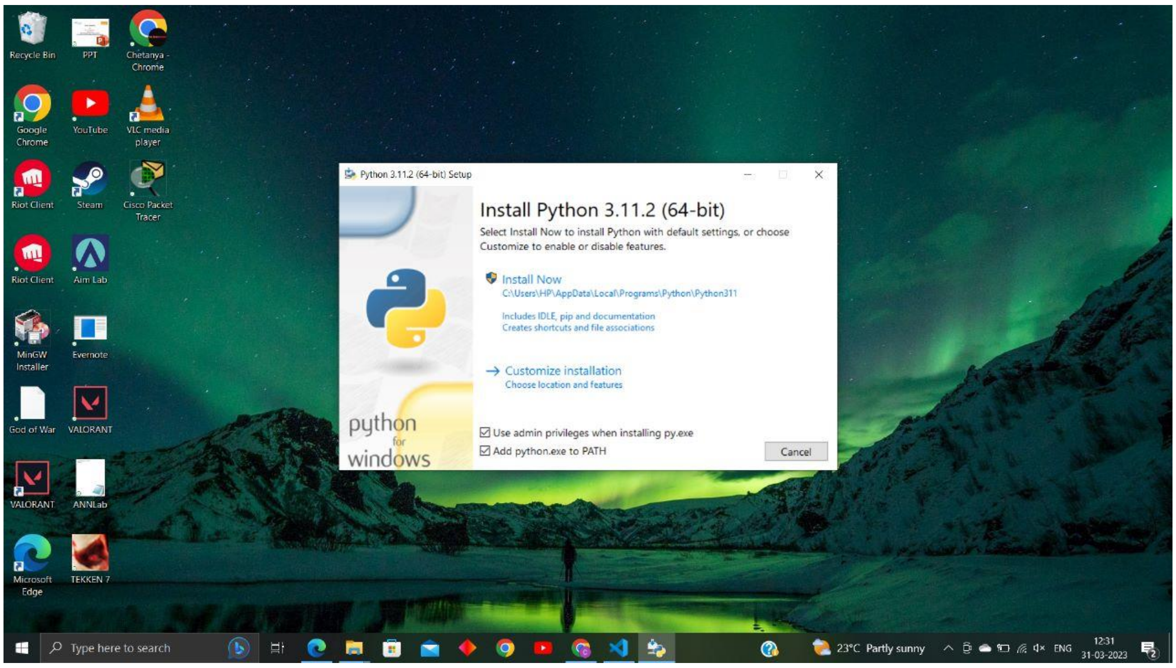
Installations

- 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.



Python 3.11.2 (64-bit) Setup

Install Python 3.11.2 (64-bit)

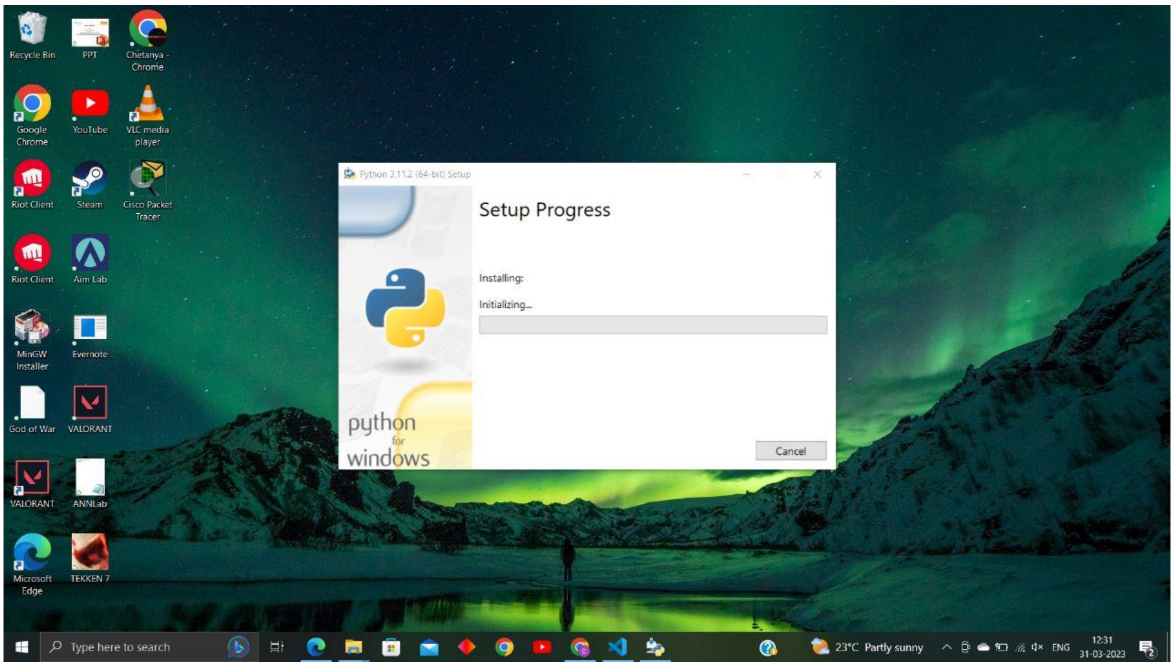
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\HP\AppData\Local\Programs\Python\Python311

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Use admin privileges when installing py.exe
☑ Add python.exe to PATH

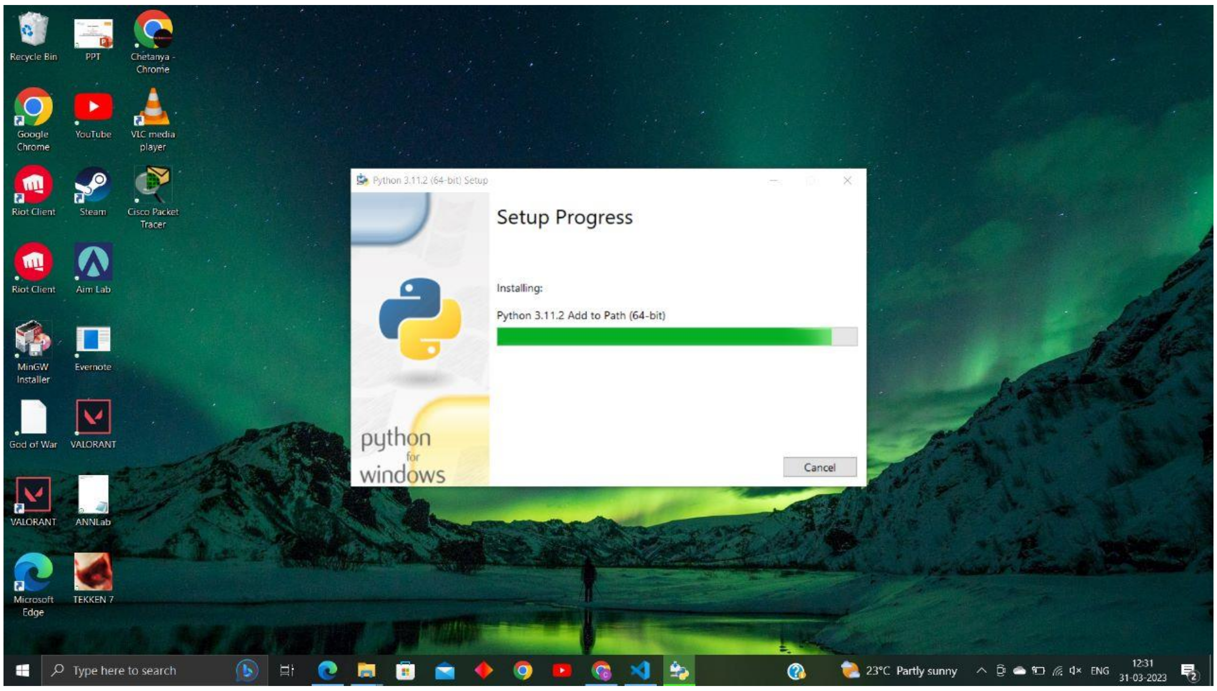


Python 3.11.2 (64-bit) Setup

Setup Progress

Installing:
Initializing...

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


Python 3.11.2 (64-bit) Setup

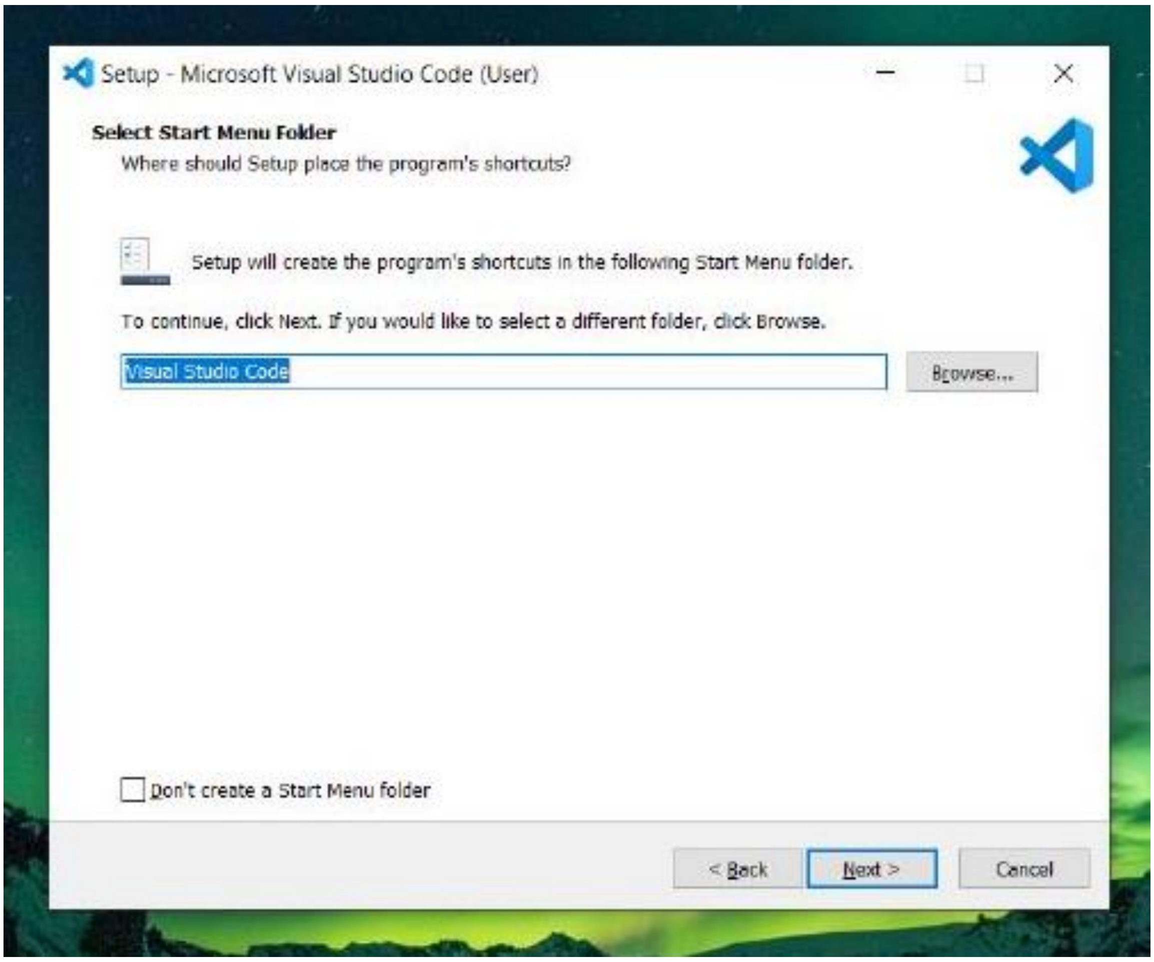
Setup Progress

Installing:

Python 3.11.2 Add to Path (64-bit)

- Visual Studio Code Installation

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


Setup - Microsoft Visual Studio Code (User)

Select Start Menu Folder

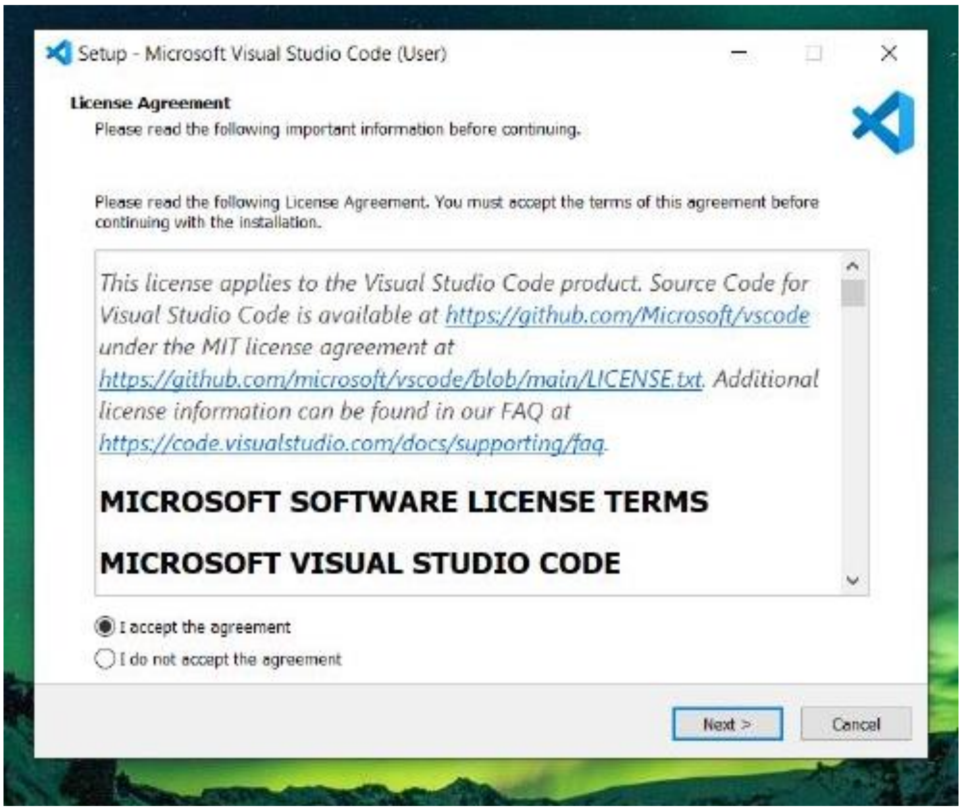
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Visual Studio Code

2. Accept the License agreement and click on Next.


Setup - Microsoft Visual Studio Code (User)

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

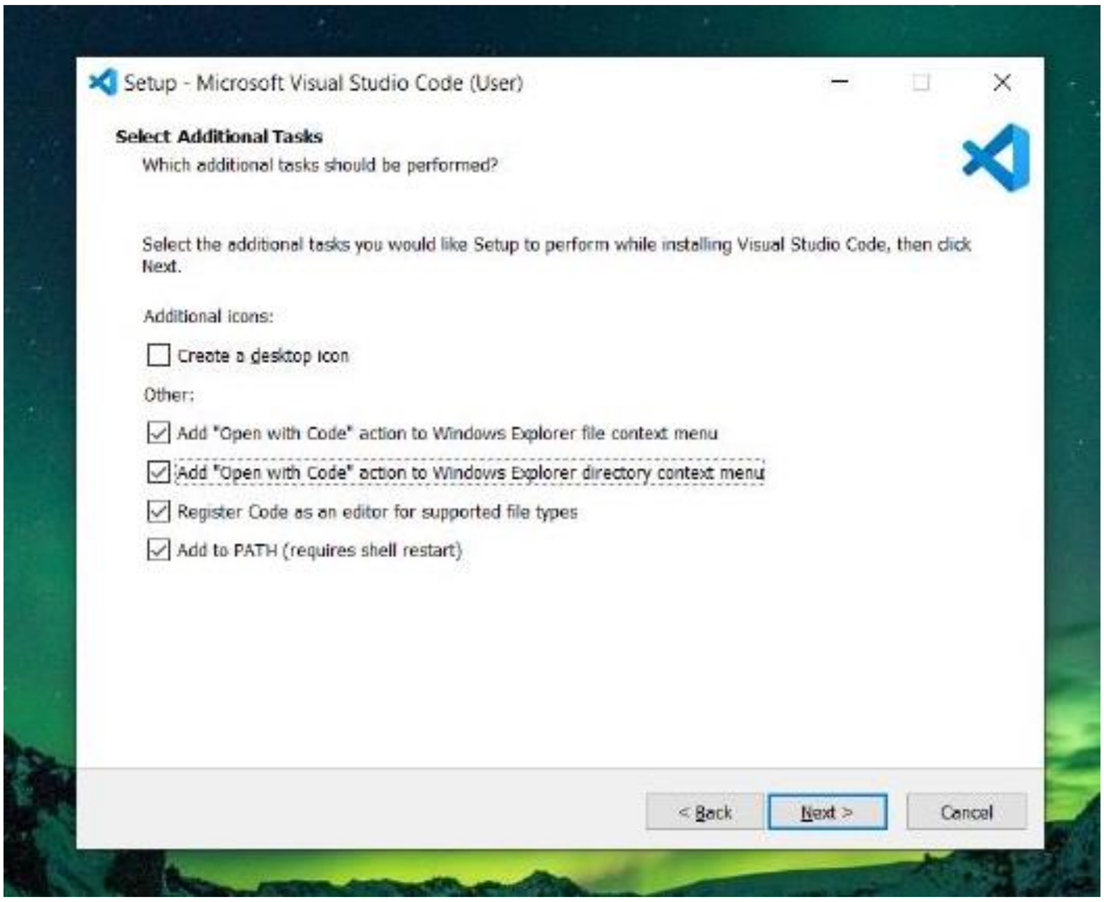
This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO CODE

[✓ I accept the agreement]
I do not accept the agreement

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


Setup - Microsoft Visual Studio Code (User)

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

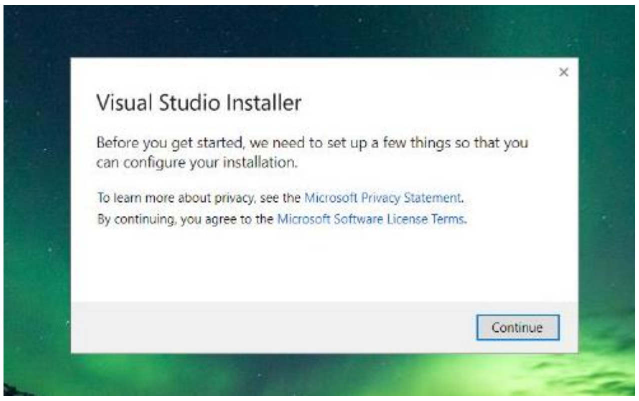
Additional icons:

☐ Create a desktop icon

Other:

☑ Add "Open with Code" action to Windows Explorer file context menu
☑ Add "Open with Code" action to Windows Explorer directory context menu
☑ Register Code as an editor for supported file types
☑ Add to PATH (requires shell restart)

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

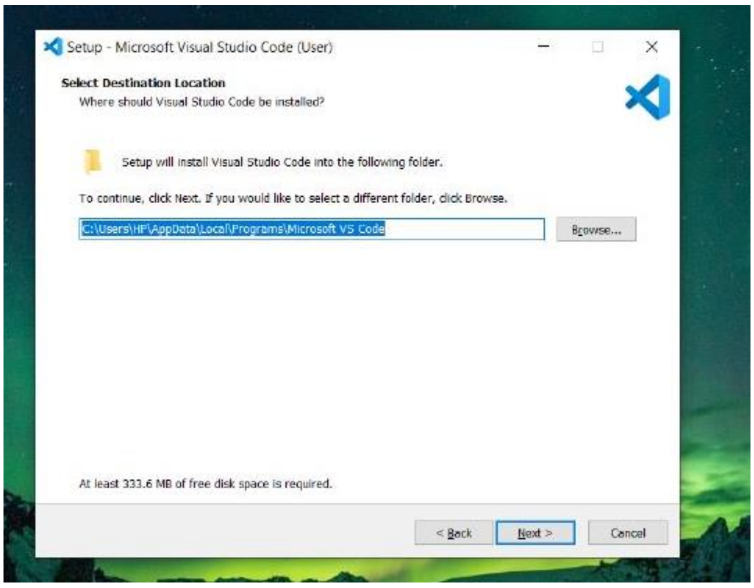


Visual Studio Installer

Before you get started, we need to set up a few things so that you can configure your installation.

To learn more about privacy, see the Microsoft Privacy Statement.

By continuing, you agree to the Microsoft Software License Terms.



Setup - Microsoft Visual Studio Code (User)

Select Destination Location

Where should Visual Studio Code be installed?

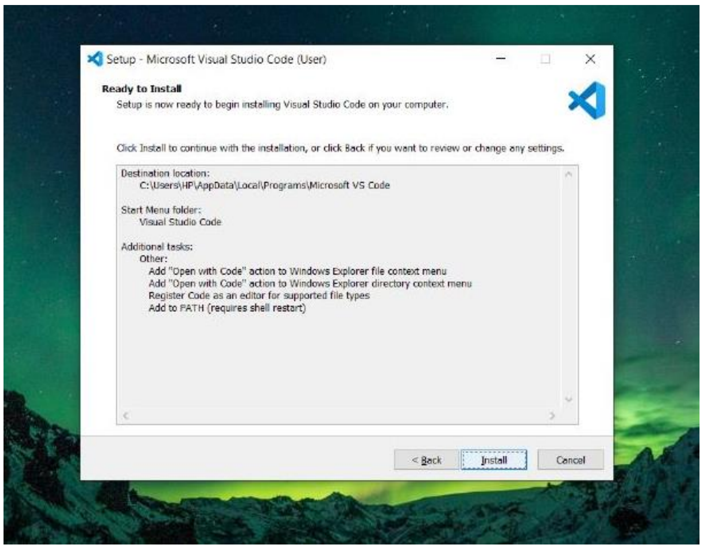
Setup will install Visual Studio Code into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Users\HP\AppData\Local\Programs\Microsoft VS Code

At least 333.6 MB of free disk space is required.

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



Setup - Microsoft Visual Studio Code (User)

Ready to Install

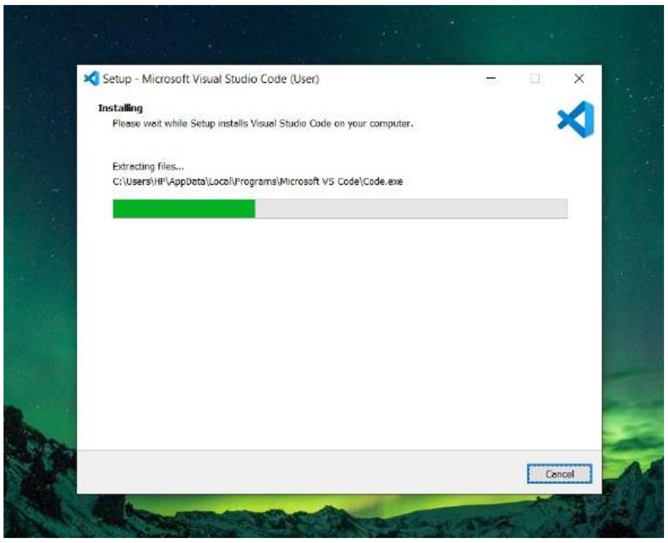
Setup is now ready to begin installing Visual Studio Code on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
      C:\Users\HP\AppData\Local\Programs\Microsoft VS Code

Start Menu folder:
      Visual Studio Code

Additional tasks:
      Other:
         Add "Open with Code" action to Windows Explorer file context menu
         Add "Open with Code" action to Windows Explorer directory context menu
         Register Code as an editor for supported file types
         Add to PATH (requires shell restart)
```



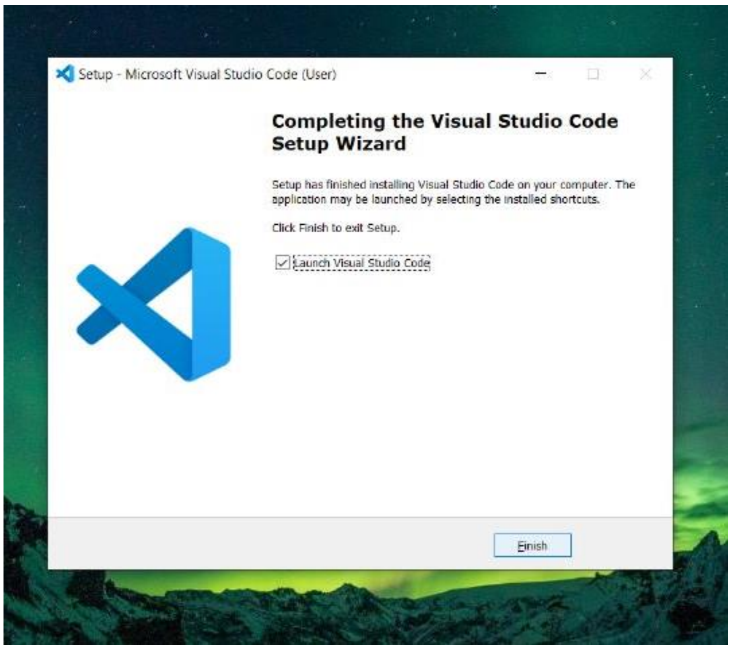
Setup - Microsoft Visual Studio Code (User)

Installing

Please wait while Setup installs Visual Studio Code on your computer.

```
Extracting files...
C:\Users\HP\AppData\Local\Programs\Microsoft VS Code\Code.exe
```

6. Click on finish to complete the process.



Setup - Microsoft Visual Studio Code (User)

Completing the Visual Studio Code Setup Wizard

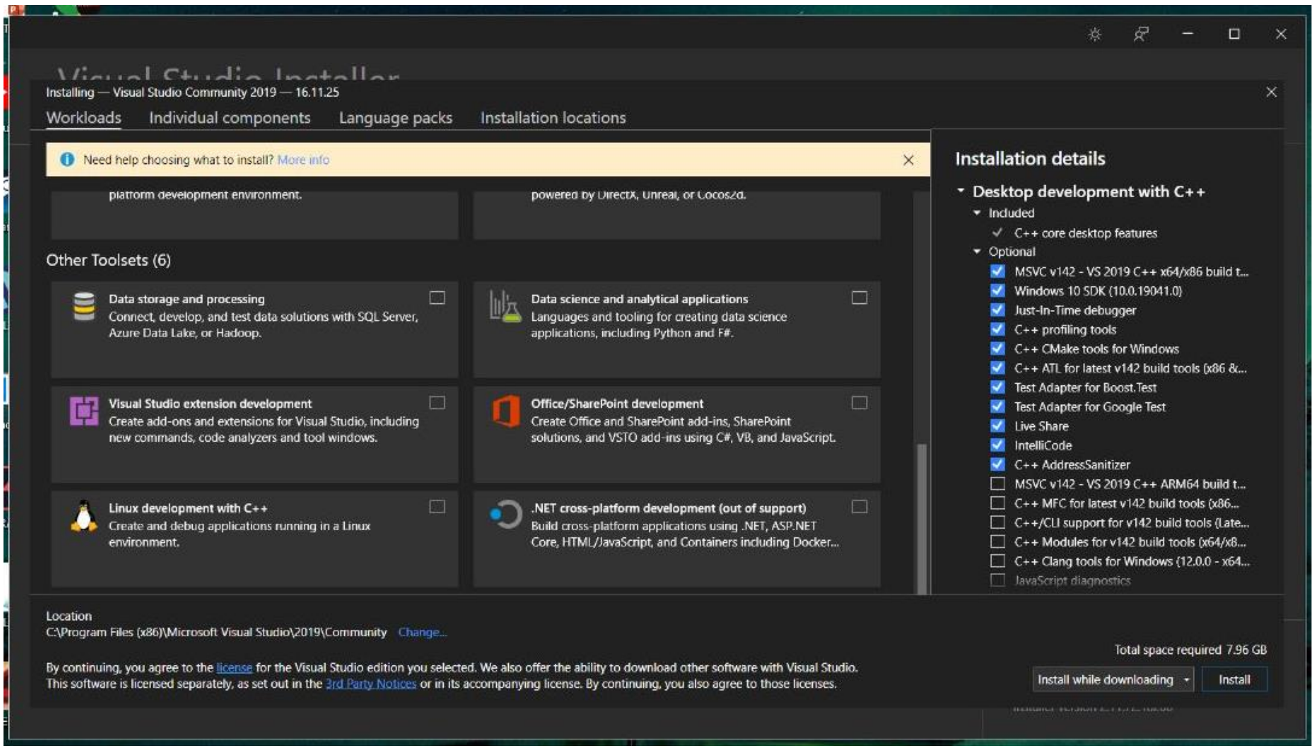
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

[✓] Launch Visual Studio Code

- 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.



Installing — Visual Studio Community 2019 — 16.11.25

Workloads | Individual components | Language packs | Installation locations

Need help choosing what to install? More info

platform development environment.

powered by DirectX, Unreal, or Cocos2d.

Other Toolsets (6)

☐ Data storage and processing
Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.

☐ Data science and analytical applications
Languages and tooling for creating data science applications, including Python and F#.

☐ Visual Studio extension development
Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.

☐ Office/SharePoint development
Create Office and SharePoint add-ins, SharePoint solutions, and VSTO add-ins using C#, VB, and JavaScript.

☐ Linux development with C++
Create and debug applications running in a Linux environment.

☐ .NET cross-platform development (out of support)
Build cross-platform applications using .NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker...

Installation details

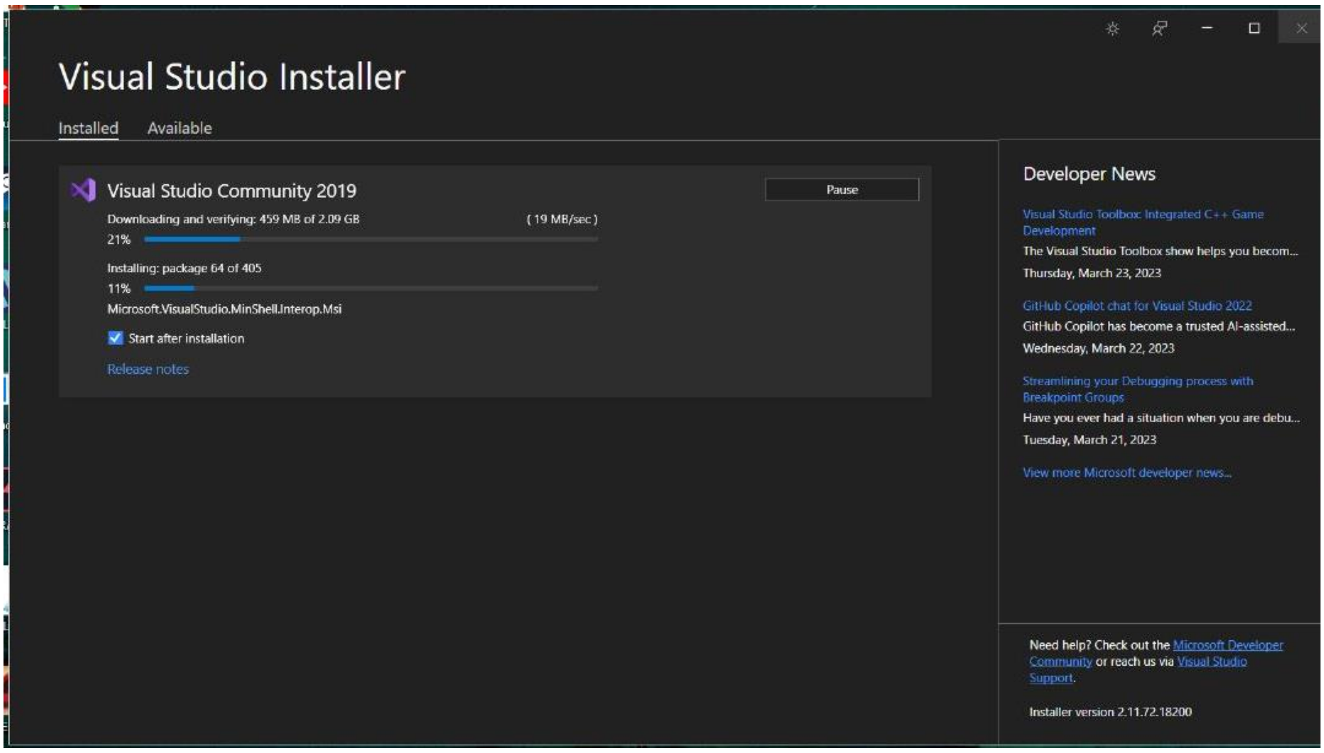
- Desktop development with C++
  - Included
    - ✓ C++ core desktop features
  - Optional
    - ☑ MSVC v142 - VS 2019 C++ x64/x86 build t...
    - ☑ Windows 10 SDK (10.0.19041.0)
    - ☑ Just-In-Time debugger
    - ☑ C++ profiling tools
    - ☑ C++ CMake tools for Windows
    - ☑ C++ ATL for latest v142 build tools (x86 &...
    - ☑ Test Adapter for Boost.Test
    - ☑ Test Adapter for Google Test
    - ☑ Live Share
    - ☑ IntelliCode
    - ☑ C++ AddressSanitizer
    - ☐ MSVC v142 - VS 2019 C++ ARM64 build t...
    - ☐ C++ MFC for latest v142 build tools (x86...
    - ☐ C++/CLI support for v142 build tools (Late...
    - ☐ C++ Modules for v142 build tools (x64/x8...
    - ☐ C++ Clang tools for Windows (12.0.0 - x64...
    - ☐ JavaScript diagnostics

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 7.96 GB

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



Visual Studio Installer

Installed | Available

Visual Studio Community 2019

Downloading and verifying: 459 MB of 2.09 GB (19 MB/sec)
21%

Installing: package 64 of 405
11%
Microsoft.VisualStudio.MinShell.Interop.Msi

☑ Start after installation

Release notes

Developer News

Visual Studio Toolbox: Integrated C++ Game Development
The Visual Studio Toolbox show helps you becom...
Thursday, March 23, 2023

GitHub Copilot chat for Visual Studio 2022
GitHub Copilot has become a trusted AI-assisted...
Wednesday, March 22, 2023

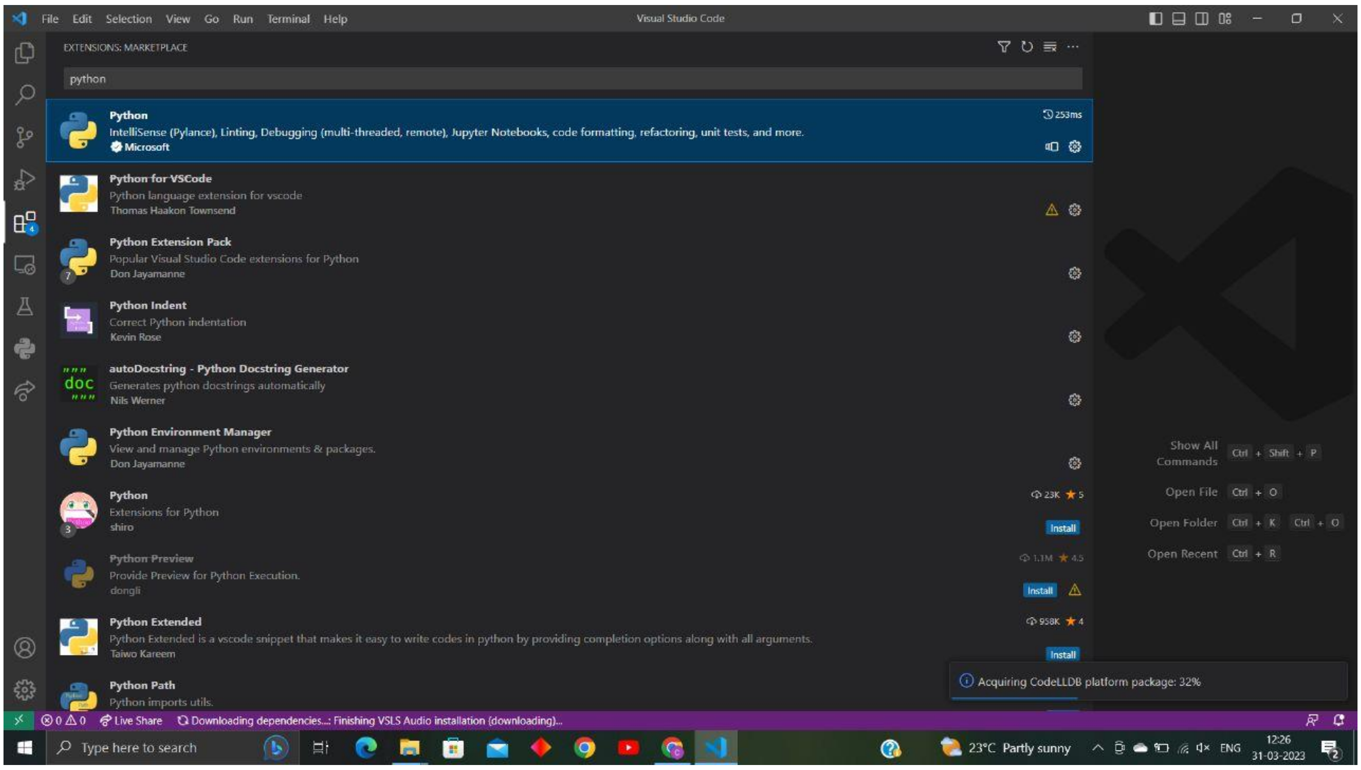
Streamlining your Debugging process with Breakpoint Groups
Have you ever had a situation when you are debu...
Tuesday, March 21, 2023

View more Microsoft developer news...

Need help? Check out the Microsoft Developer Community or reach us via Visual Studio Support.

Installer version 2.11.72.18200

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.



python

Python
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code formatting, refactoring, unit tests, and more.
Microsoft
253ms

Python for VSCode
Python language extension for vscode
Thomas Haakon Townsend

Python Extension Pack
Popular Visual Studio Code extensions for Python
Don Jayamanne

Python Indent
Correct Python indentation
Kevin Rose

autoDocstring - Python Docstring Generator
Generates python docstrings automatically
Nils Werner

Python Environment Manager
View and manage Python environments & packages.
Don Jayamanne

Python
Extensions for Python
shiro
23K ★ 5
Install

Python Preview
Provide Preview for Python Execution.
dongli
1.1M ★ 4.5
Install

Python Extended
Python Extended is a vscode snippet that makes it easy to write codes in python by providing completion options along with all arguments.
Taiwo Kareem
958K ★ 4
Install

Python Path
Python imports utils.

Show All Commands Ctrl + Shift + P
Open File Ctrl + O
Open Folder Ctrl + K Ctrl + O
Open Recent Ctrl + R

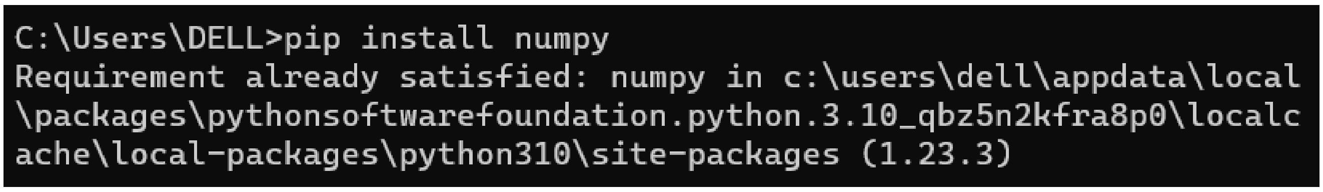
Acquiring CodeLLDB platform package: 32%

0 ⚠ 0 Live Share Downloading dependencies...: Finishing VSLS Audio installation (downloading)...

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

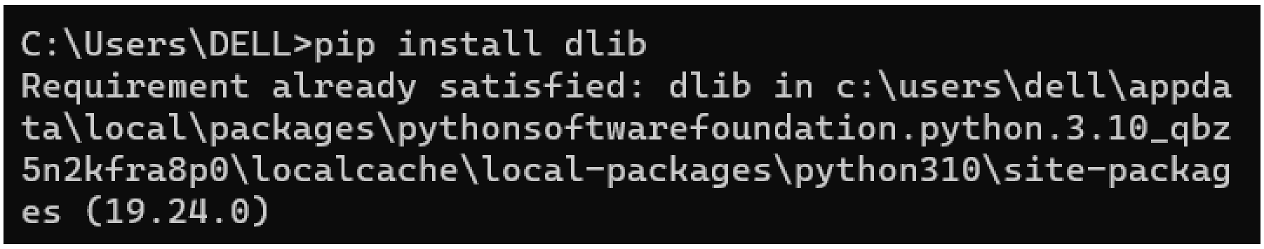


```
C:\Users\DELL>pip install numpy
Requirement already satisfied: numpy in c:\users\dell\appdata\local
\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localc
ache\local-packages\python310\site-packages (1.23.3)
```

---

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

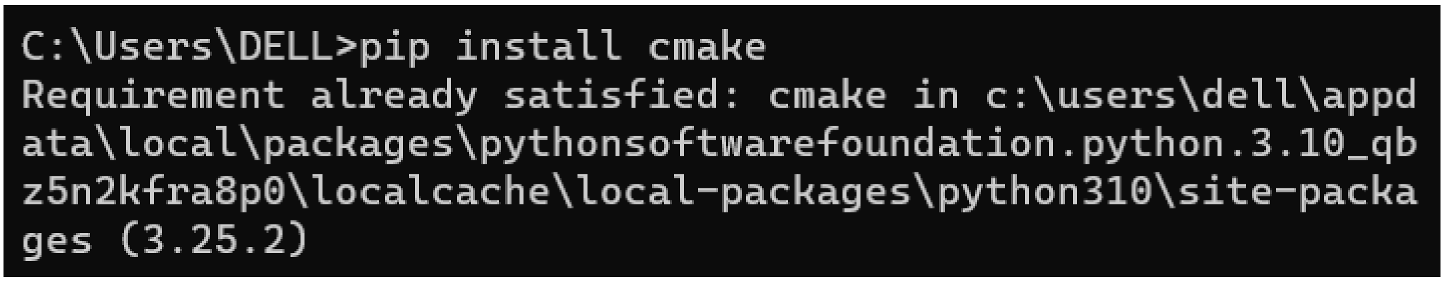


```
C:\Users\DELL>pip install dlib
Requirement already satisfied: dlib in c:\users\dell\appda
ta\local\packages\pythonsoftwarefoundation.python.3.10_qbz
5n2kfra8p0\localcache\local-packages\python310\site-packag
es (19.24.0)
```

---

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.

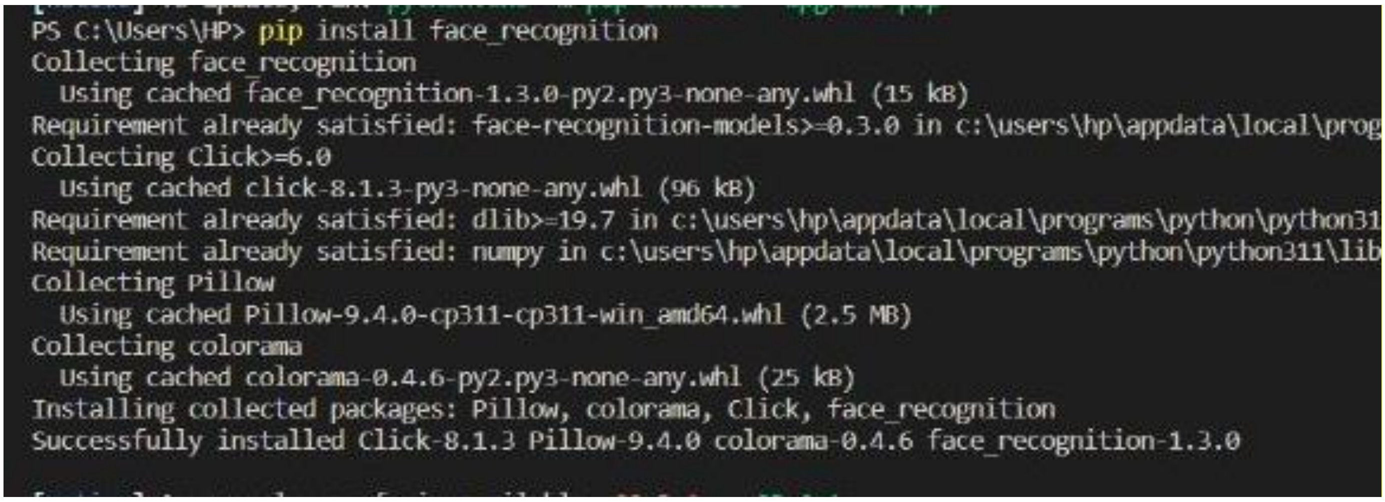
Command: pip install cmake



```
C:\Users\DELL>pip install cmake
Requirement already satisfied: cmake in c:\users\dell\appd
ata\local\packages\pythonsoftwarefoundation.python.3.10_qb
z5n2kfra8p0\localcache\local-packages\python310\site-packa
ges (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

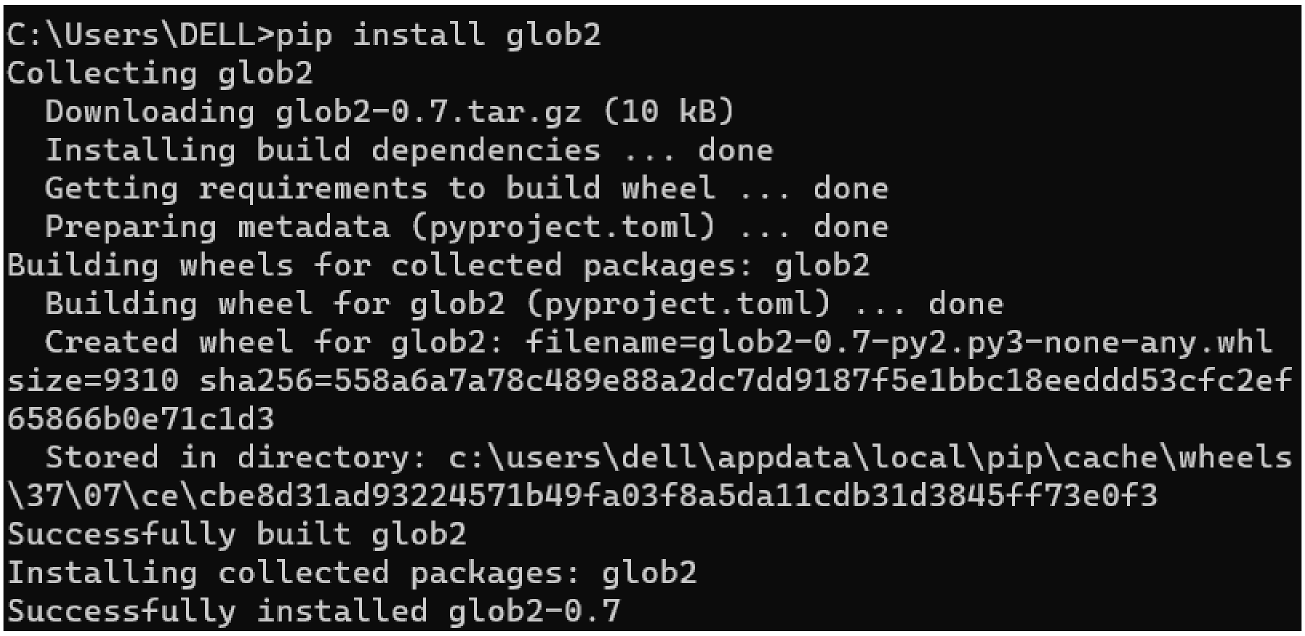


```
PS C:\Users\HP> pip install face_recognition
Collecting face_recognition
  Using cached face_recognition-1.3.0-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: face-recognition-models>=0.3.0 in c:\users\hp\appdata\local\prog
Collecting Click>=6.0
  Using cached click-8.1.3-py3-none-any.whl (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.

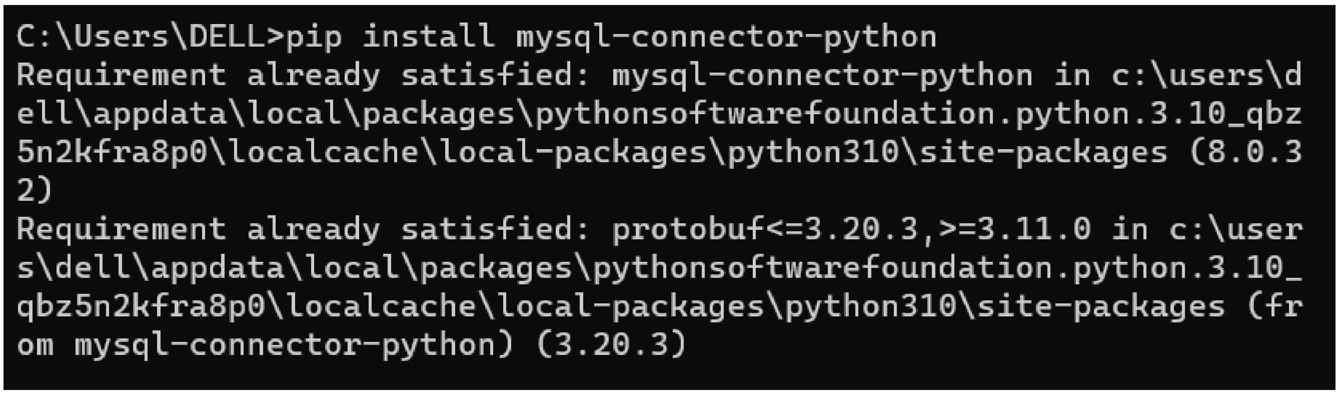
Command: pip install glob2



```
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.

Command: pip install mysql-connector-python

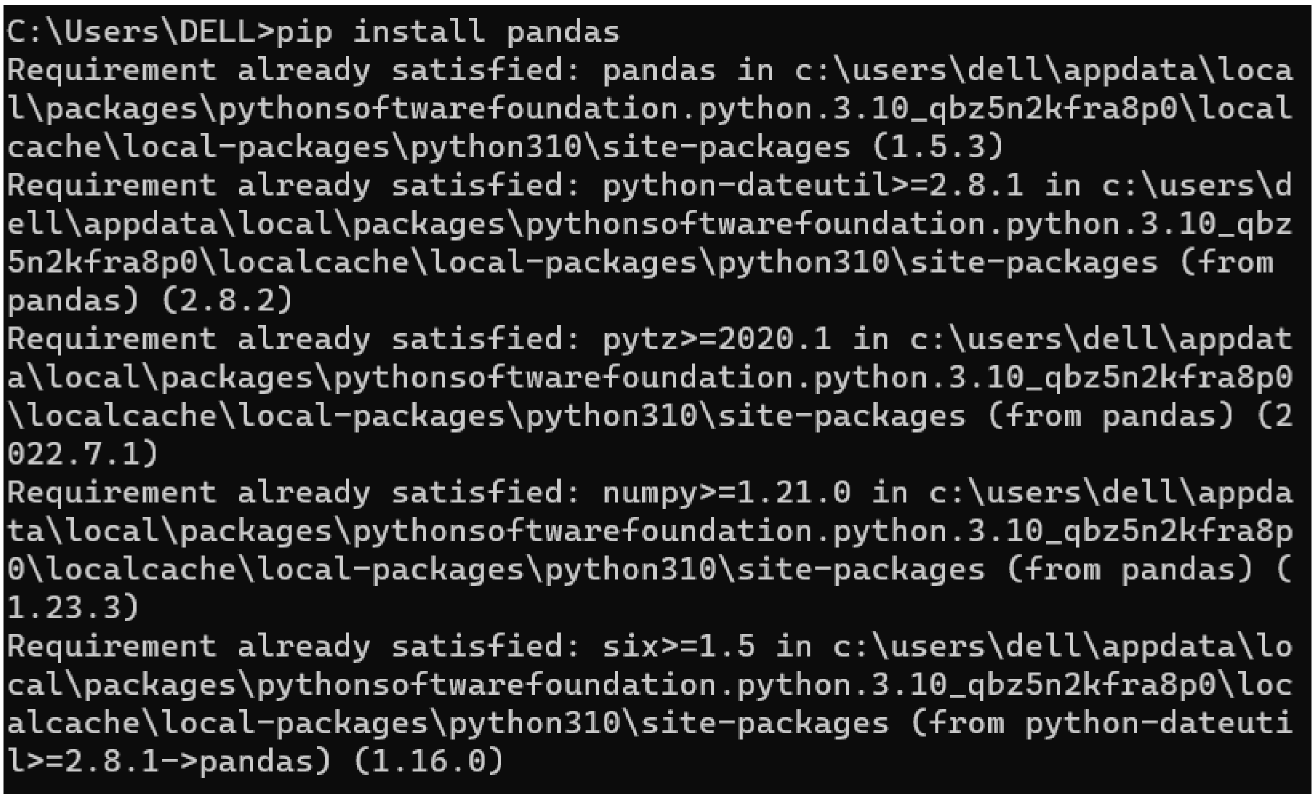


```
C:\Users\DELL>pip install mysql-connector-python
Requirement already satisfied: mysql-connector-python in c:\users\d
ell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz
5n2kfra8p0\localcache\local-packages\python310\site-packages (8.0.3
2)
Requirement already satisfied: protobuf<=3.20.3,>=3.11.0 in c:\user
s\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_
qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (fr
om 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.

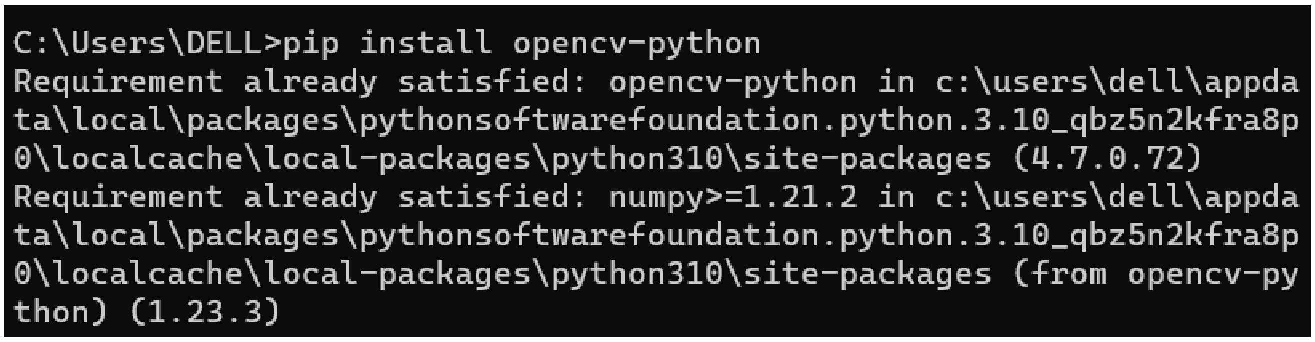
Command: pip install pandas



```
C:\Users\DELL>pip install pandas
Requirement already satisfied: pandas in c:\users\dell\appdata\loca
l\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\local
cache\local-packages\python310\site-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\d
ell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz
5n2kfra8p0\localcache\local-packages\python310\site-packages (from
pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\appdat
a\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0
\localcache\local-packages\python310\site-packages (from pandas) (2
022.7.1)
Requirement already satisfied: numpy>=1.21.0 in c:\users\dell\appda
ta\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p
0\localcache\local-packages\python310\site-packages (from pandas) (
1.23.3)
Requirement already satisfied: six>=1.5 in c:\users\dell\appdata\lo
cal\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\loc
alcache\local-packages\python310\site-packages (from python-dateuti
l>=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.

Command: pip install opencv-python

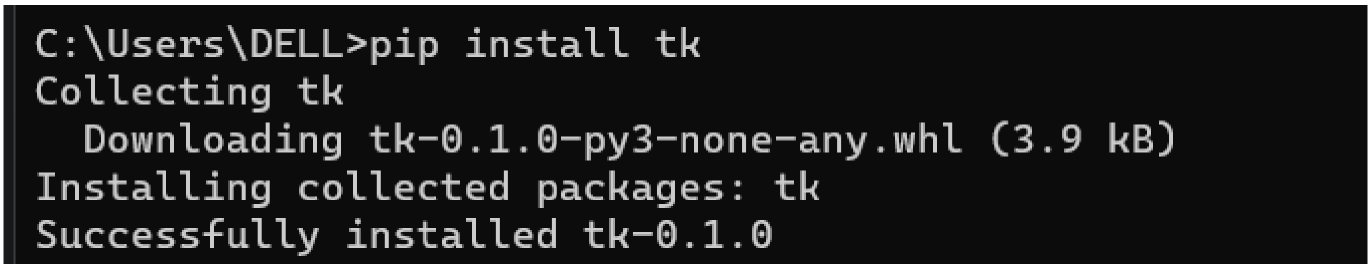


```
C:\Users\DELL>pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\dell\appda
ta\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p
0\localcache\local-packages\python310\site-packages (4.7.0.72)
Requirement already satisfied: numpy>=1.21.2 in c:\users\dell\appda
ta\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p
0\localcache\local-packages\python310\site-packages (from opencv-py
thon) (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

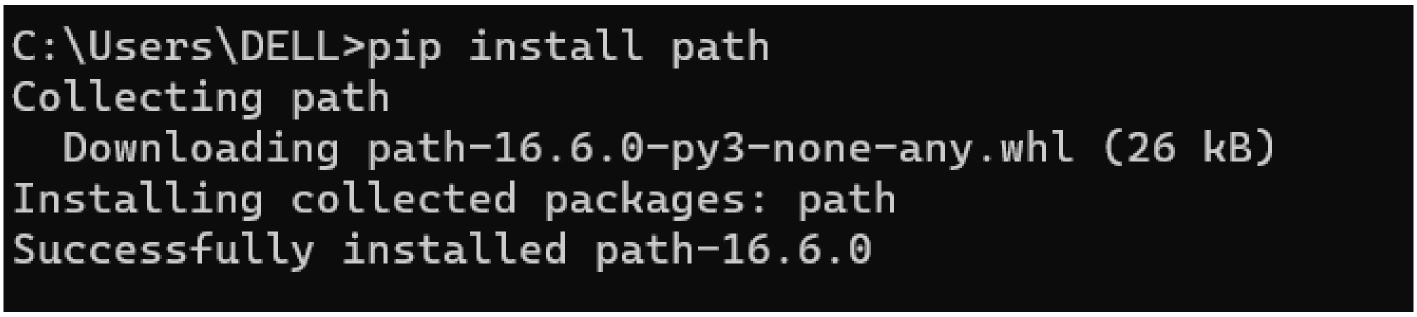


```
C:\Users\DELL>pip install tk
Collecting tk
  Downloading tk-0.1.0-py3-none-any.whl (3.9 kB)
Installing collected packages: tk
Successfully installed tk-0.1.0
```

---

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

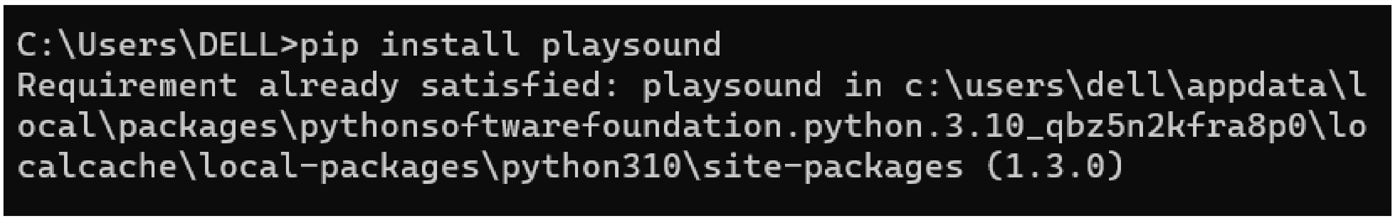


```
C:\Users\DELL>pip install path
Collecting path
  Downloading path-16.6.0-py3-none-any.whl (26 kB)
Installing collected packages: path
Successfully installed path-16.6.0
```

---

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

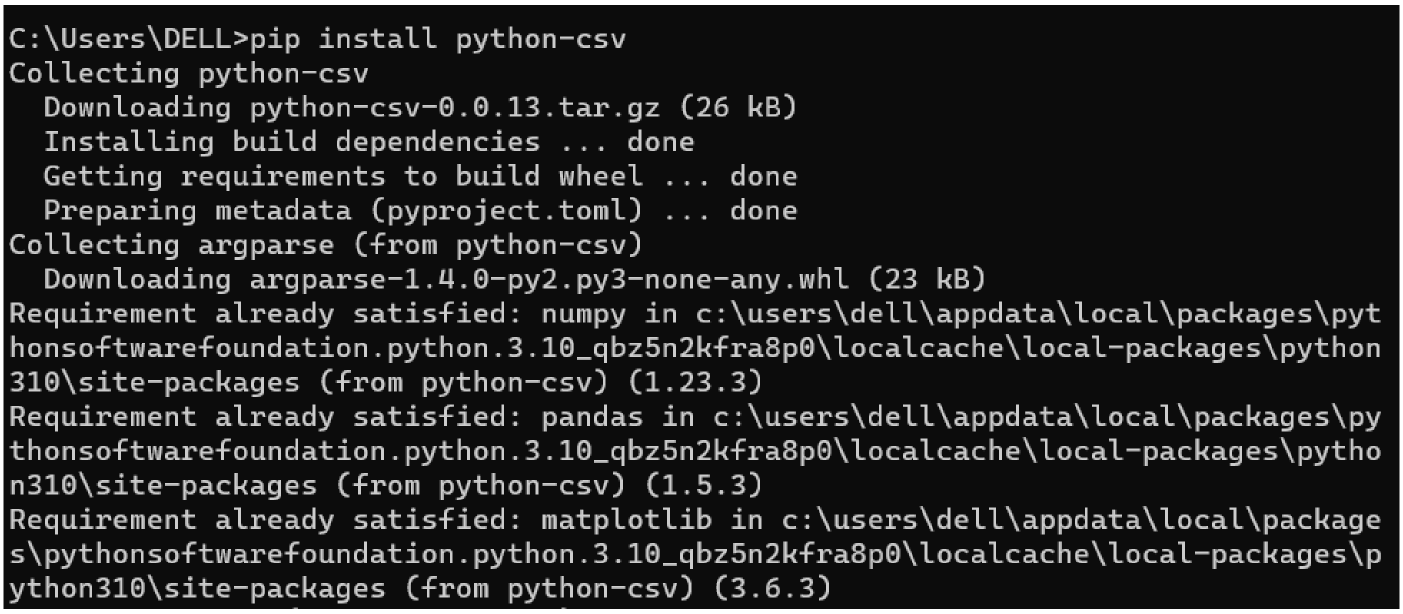
Command: pip install playsound



```
C:\Users\DELL>pip install playsound
Requirement already satisfied: playsound in c:\users\dell\appdata\l
ocal\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\lo
calcache\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.

Command: pip install python-csv

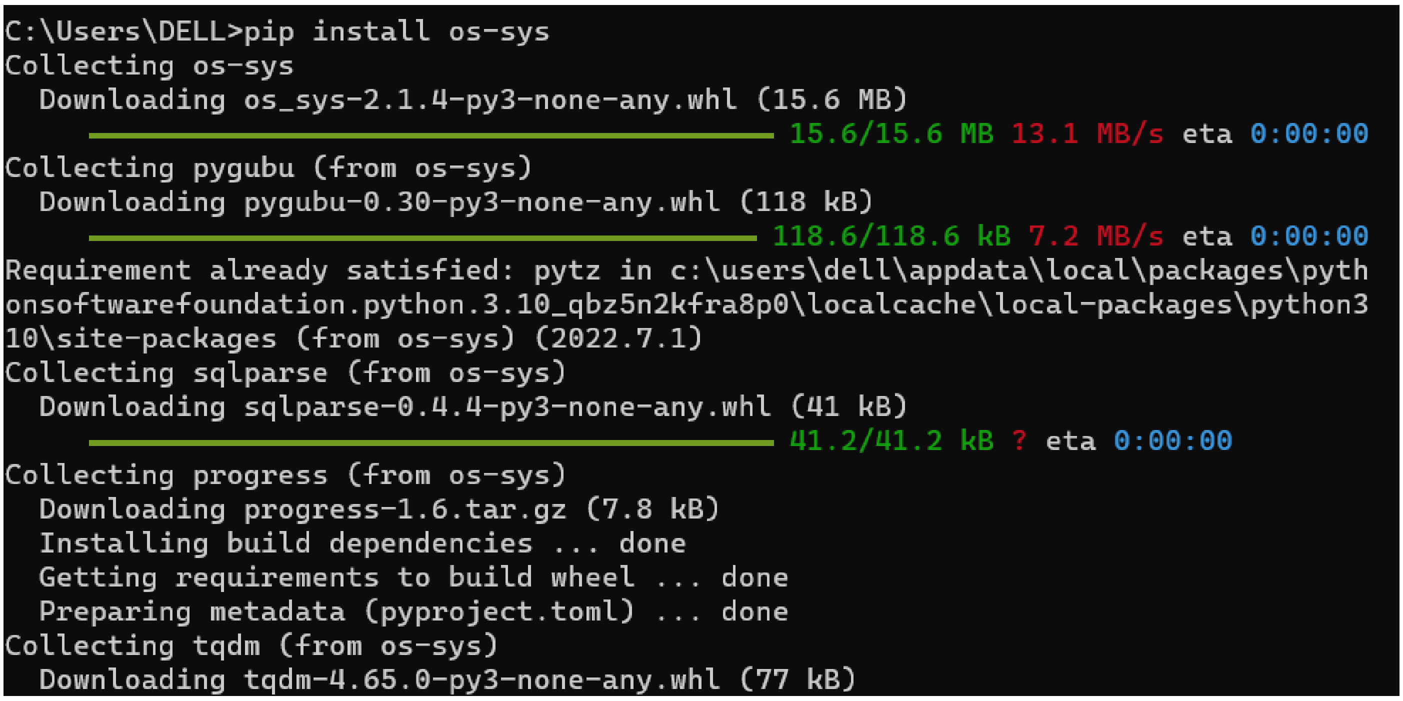


```
C:\Users\DELL>pip install python-csv
Collecting python-csv
  Downloading python-csv-0.0.13.tar.gz (26 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting argparse (from python-csv)
  Downloading argparse-1.4.0-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: numpy in c:\users\dell\appdata\local\packages\pyt
honsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python
310\site-packages (from python-csv) (1.23.3)
Requirement already satisfied: pandas in c:\users\dell\appdata\local\packages\py
thonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\pytho
n310\site-packages (from python-csv) (1.5.3)
Requirement already satisfied: matplotlib in c:\users\dell\appdata\local\package
s\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\p
ython310\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.

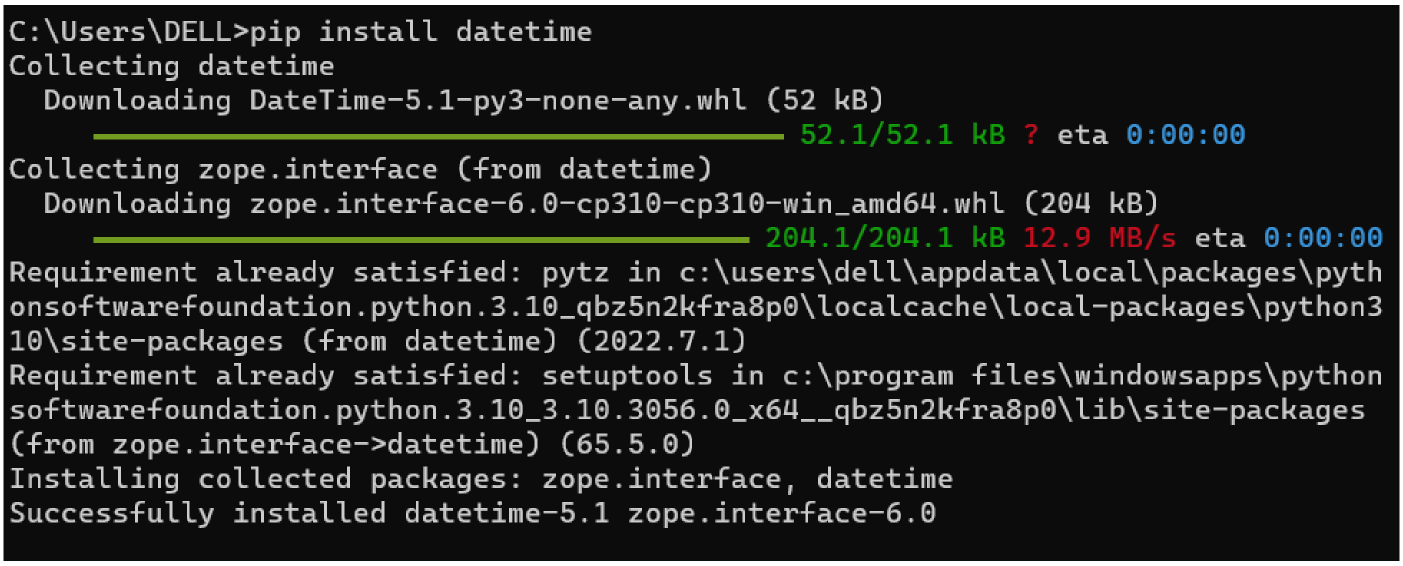
Command: pip install os-sys



```
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\pyth
onsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python3
10\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

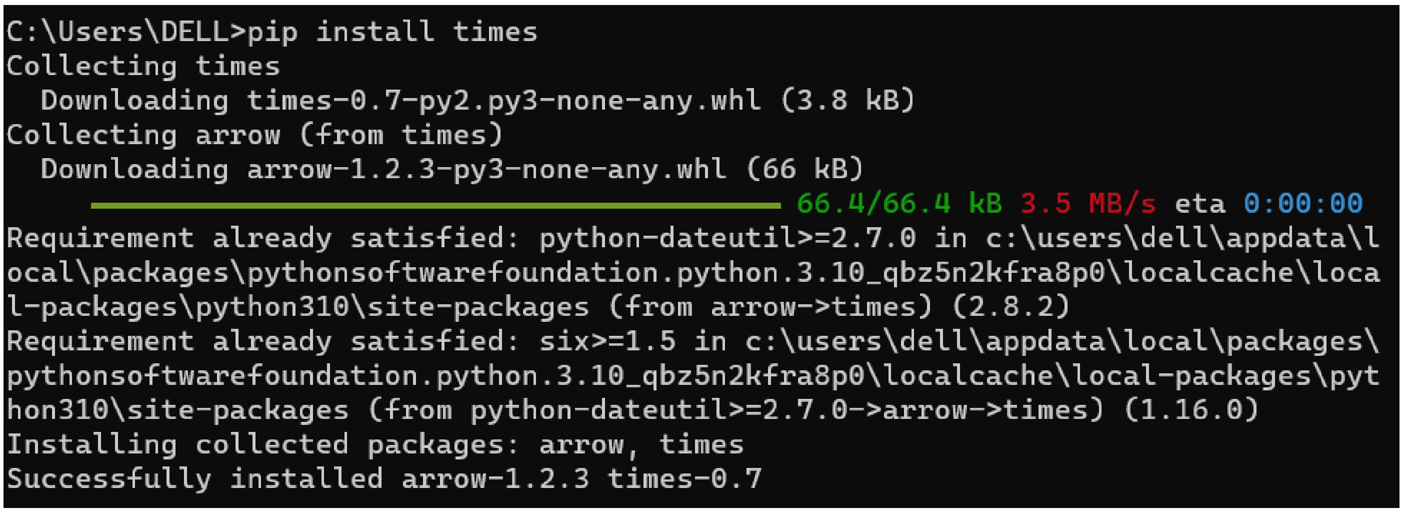


```
C:\Users\DELL>pip install datetime
Collecting datetime
  Downloading DateTime-5.1-py3-none-any.whl (52 kB)
     ---------------------------------------- 52.1/52.1 kB ? eta 0:00:00
Collecting zope.interface (from datetime)
  Downloading zope.interface-6.0-cp310-cp310-win_amd64.whl (204 kB)
     -------------------------------------- 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.

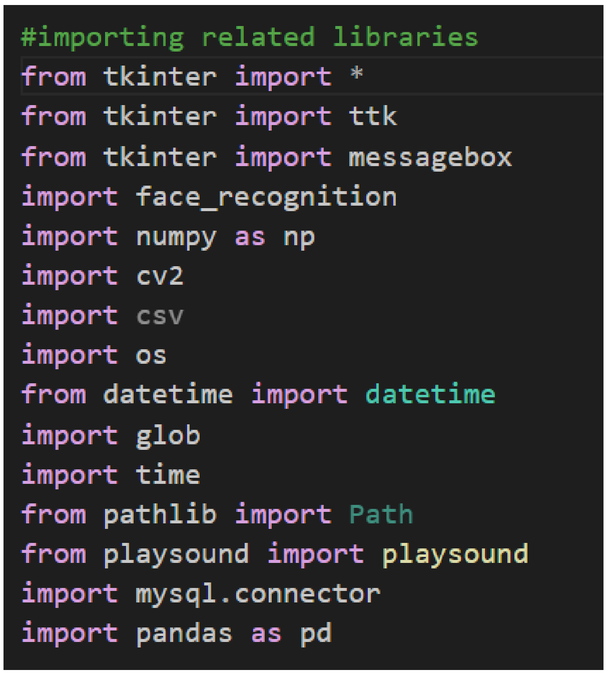
Command: pip install times



```
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
```

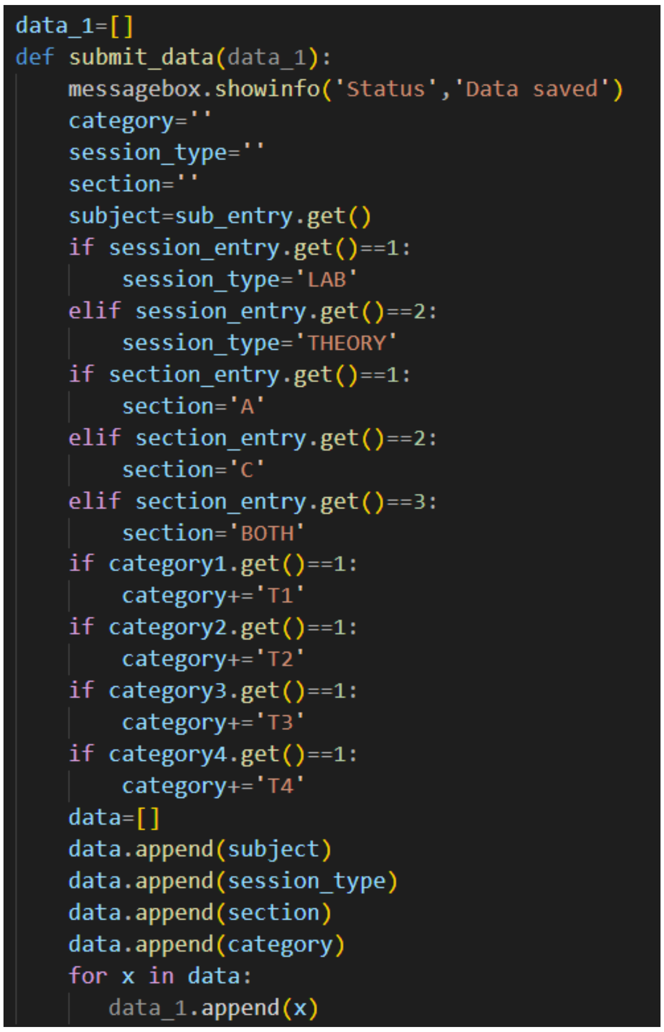
Code Documentation

Importing all the necessary libraries and modules



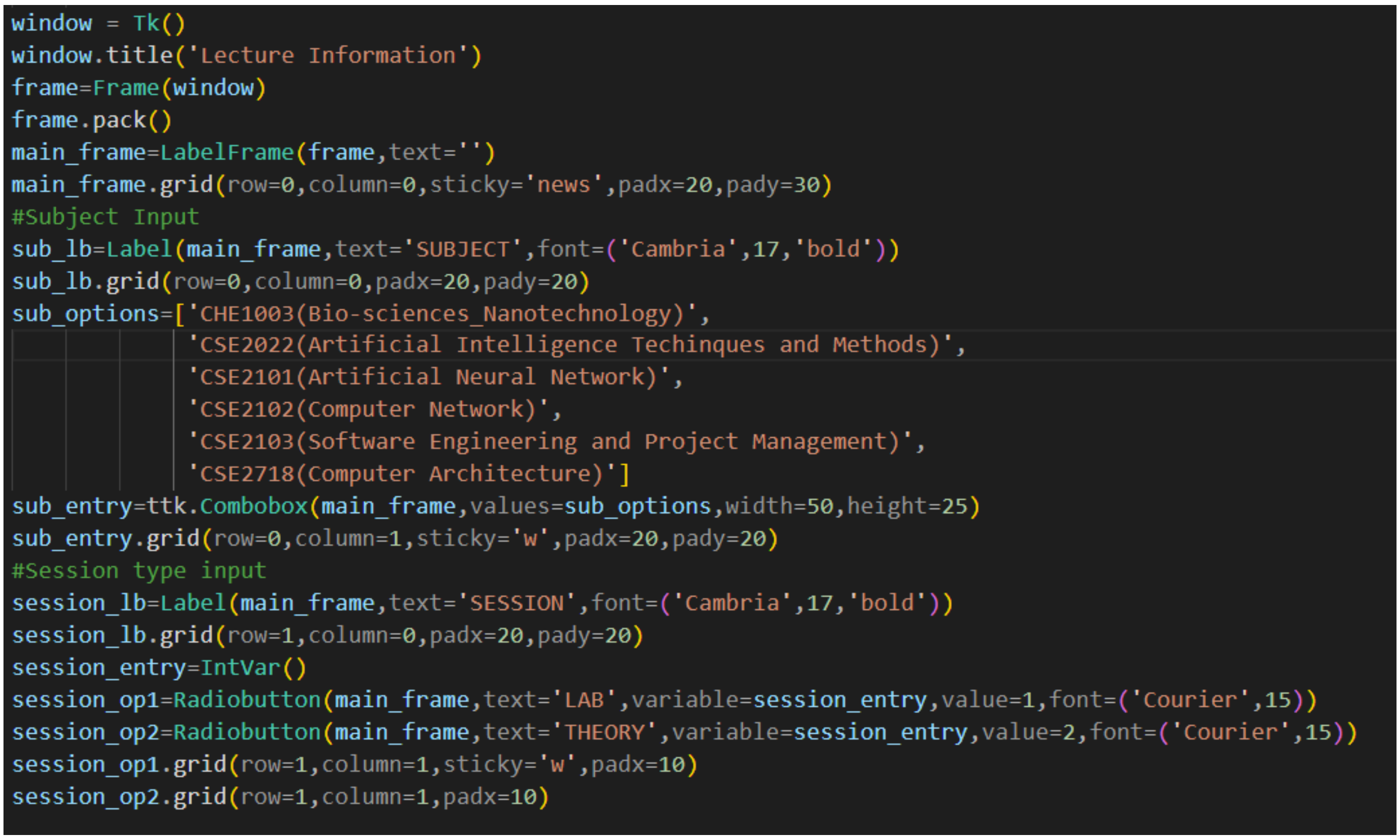
```python
#importing related libraries
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
import face_recognition
import numpy as np
import cv2
import csv
import os
from datetime import datetime
import glob
import time
from pathlib import Path
from playsound import playsound
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 a array for further use.

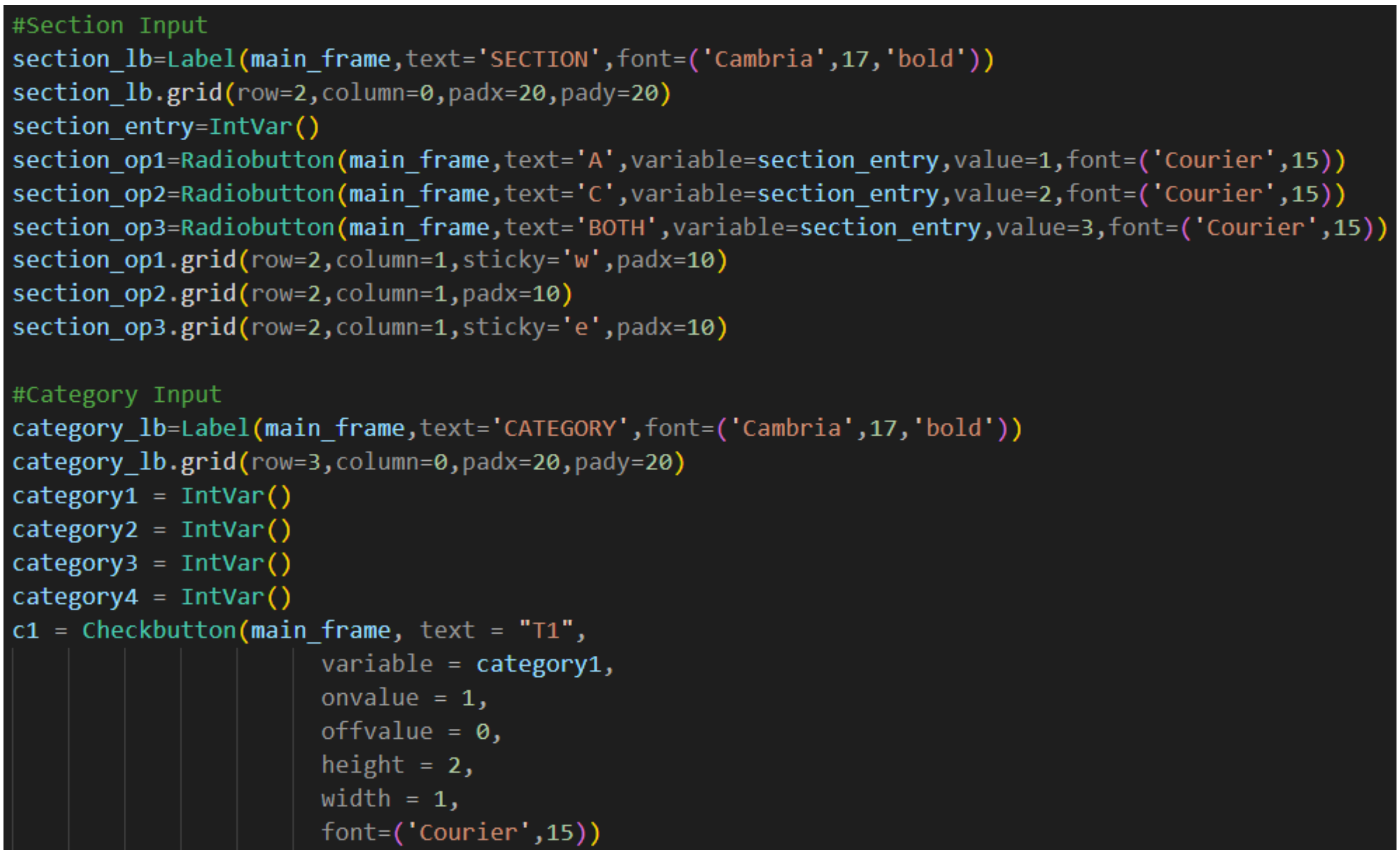


```python
data_1=[]
def submit_data(data_1):
    messagebox.showinfo('Status','Data saved')
    category=''
    session_type=''
    section=''
    subject=sub_entry.get()
    if session_entry.get()==1:
        session_type='LAB'
    elif session_entry.get()==2:
        session_type='THEORY'
    if section_entry.get()==1:
        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)
```

It 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.



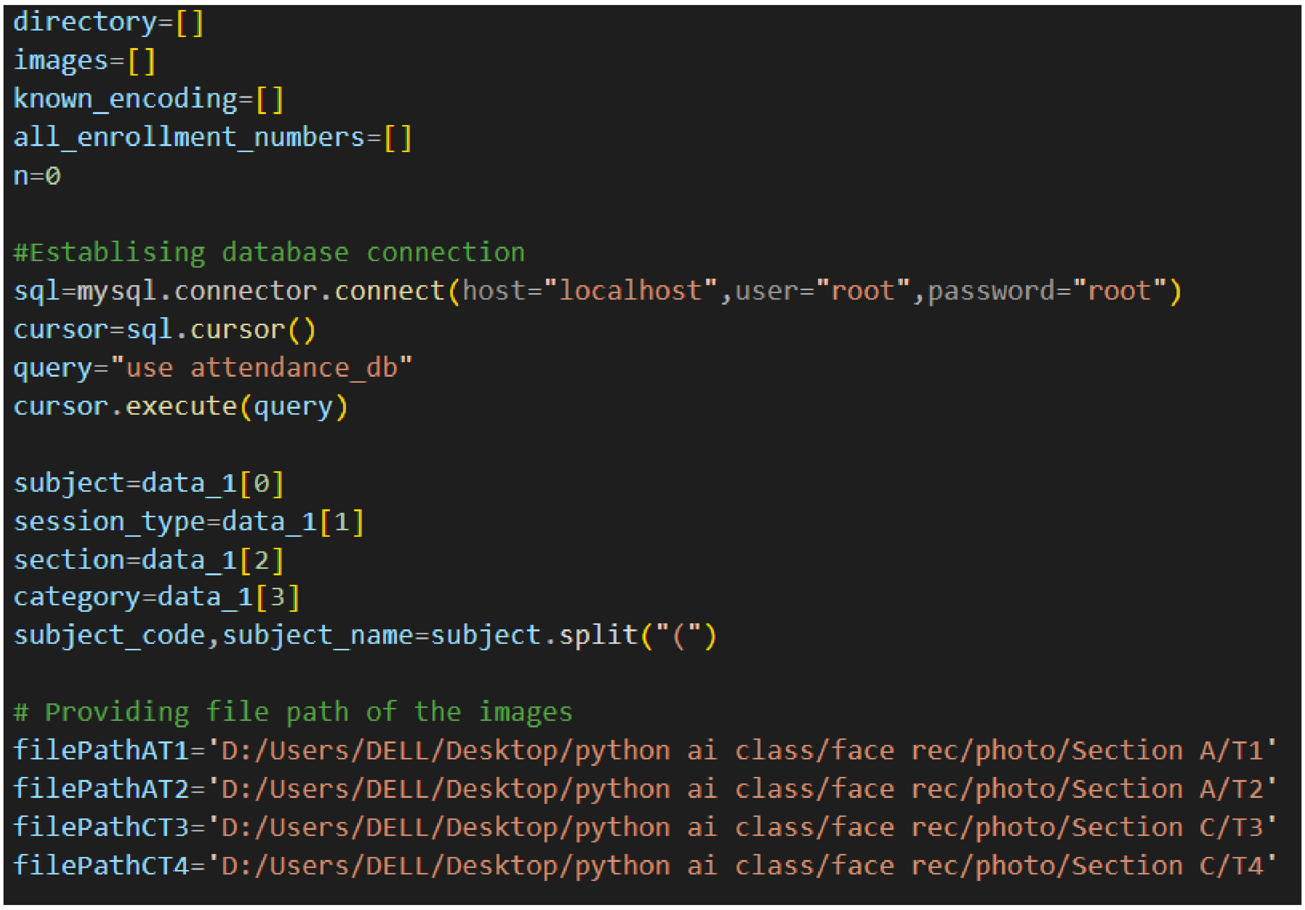
```python
window = Tk()
window.title('Lecture Information')
frame=Frame(window)
frame.pack()
main_frame=LabelFrame(frame,text='')
main_frame.grid(row=0,column=0,sticky='news',padx=20,pady=30)
#Subject Input
sub_lb=Label(main_frame,text='SUBJECT',font=('Cambria',17,'bold'))
sub_lb.grid(row=0,column=0,padx=20,pady=20)
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)
```



```python
#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))
```

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 frame window in variable and extract the subject code from the Subject input. Creating variables which store the file path of the images stored in file.



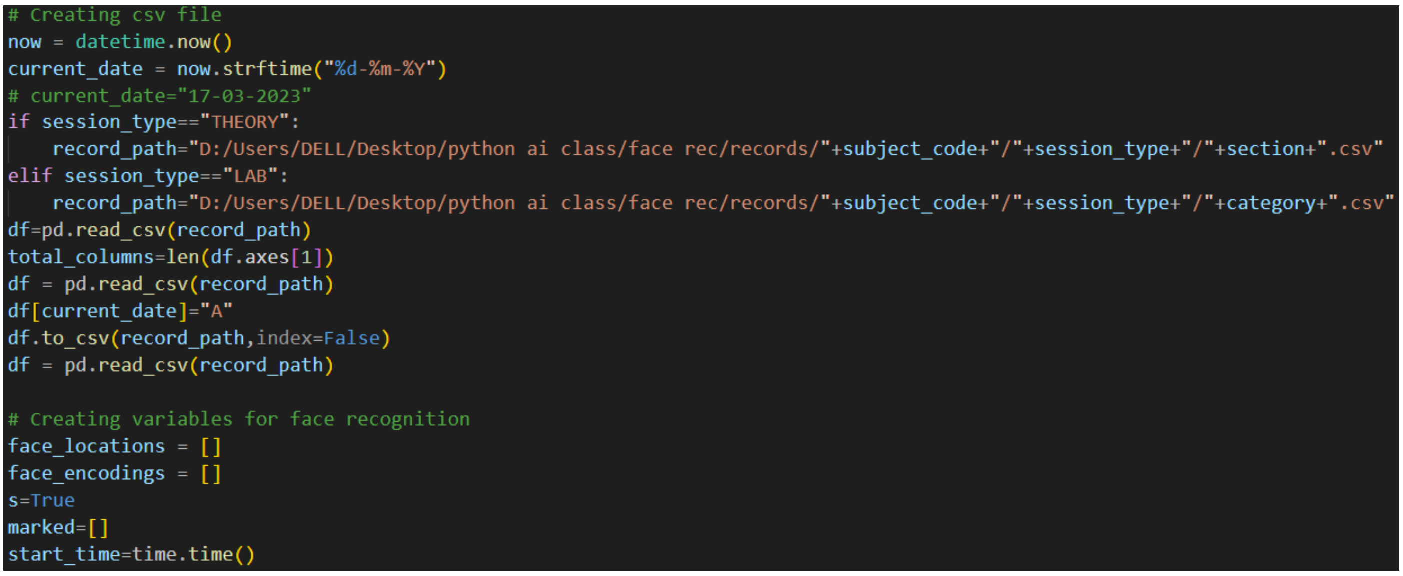
```
directory=[]
images=[]
known_encoding=[]
all_enrollment_numbers=[]
n=0

#Establising 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'
```

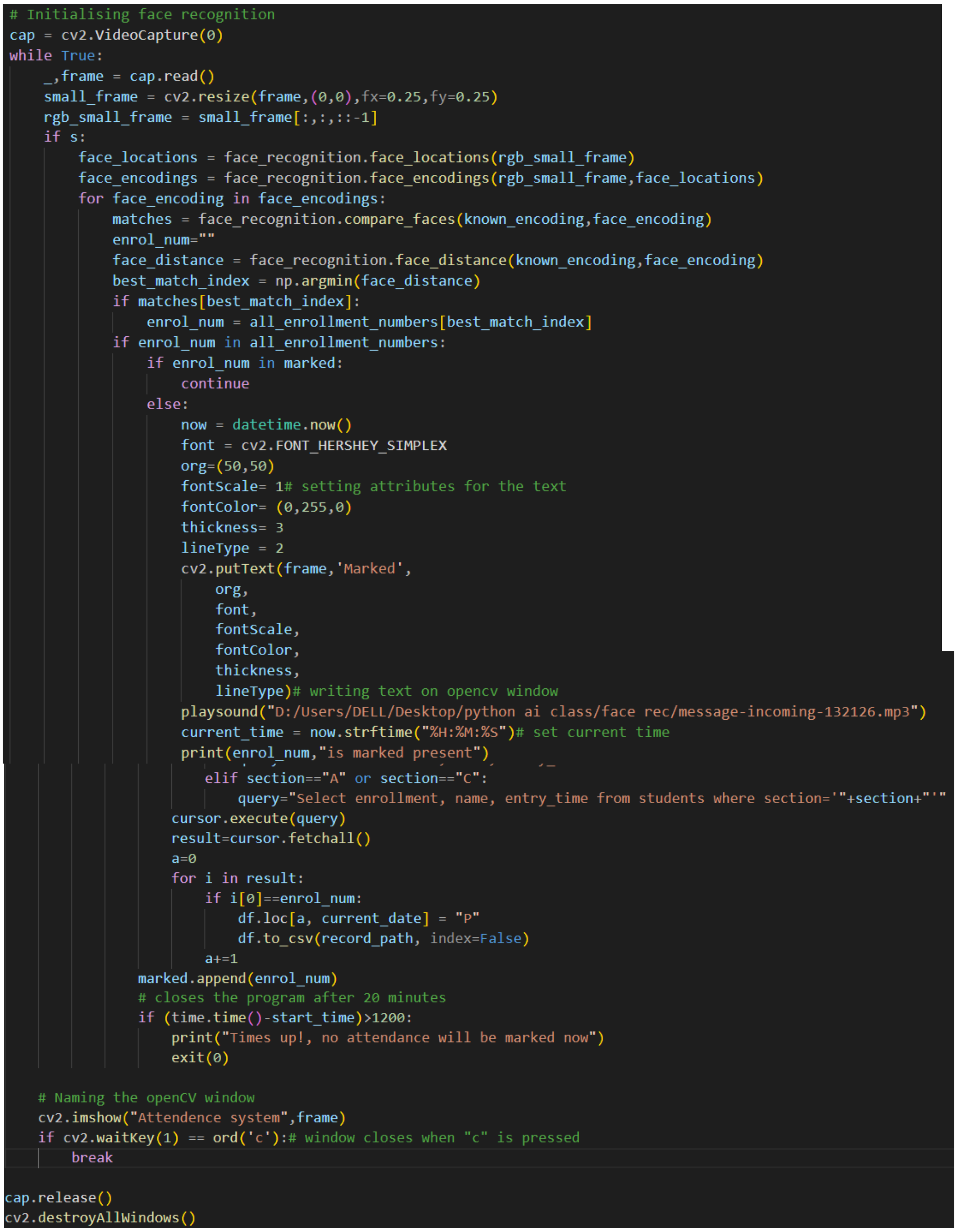
Fetching the csv file based on the given input and adding a column to the existing file by the name of current date. By default, all students are marked absent. Create list variables to store the encodings of the faces located by the face recognition system. Store the time at which the camera window start. 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 lost 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 cleses 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")
      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("Attendence system",frame)
if cv2.waitKey(1) == ord('c'):# window closes when "c" is pressed
    break

cap.release()
cv2.destroyAllWindows()
```

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/