A Project report on

“ Online Attendance compilation system”

Submitted by:

S. Advaith Reddy

Roll no :206 Class: XII

Under the Guidance of:

Mr.Anoop V S

PGT (Computer Science)

Department of Computer science

SAINIK SCHOOL KALIKIRI

Department of Computer science
SAINIK SCHOOL KALIKIRI

CERTIFICATE

This is to certify that **Cdt. S.Advaith Reddy,** Roll No. 206 of Class XII has prepared the report on the Project Entitled **“Online Attendance compilation system”**. The report is the result of his efforts & endeavors. The Report is found worthy of acceptance as final project report For the subject Computer Science of Class XII. He has Prepared the report under my guidance.

(Mr. Anoop V S)
PGT (Computer Science)
Department of Computer Science
SAINIK SCHOOL KALIKIRI

DECLARATION

I hereby declare that the project work entitled “**Online Attendance compilation system**”, submitted to Department of **Computer Science**, SAINIK SCHOOL KALIKIRI is **prepared** by **me**. All the **coding** is the result of me and my team’s **personal efforts.**

Cdt. S.Advaith Reddy
RollNo.206
Class XII

ACKNOWLEDGEMENT

I would like to express a deep sense of thanks & gratitude to my project guide Mr. Anoop V Sir for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice &constant motivation have been responsible for the successful completion of this project.

I also thanks to my parents for their motivation & support. I must thanks to my classmates for their timely help & support for compilation of this project.

Last but not the least; I would like to thank all those who had helped directly or indirectly towards the completion of this project.

CONTENTS

WORKING DESCRIPTION

Working Description of the project can be broken into these following smaller parts:

1. Compilation of the attendance .xlsx files
2. Management of the Attendance Compiled and stored.
3. Data Management of Cadet Details

First of all let's talk about the databases that are needed to be present in mysql to use this project.

- First is the cadets_information database which stores tables with names in the format of class<standard> for ex: class12 to store the information of cadet details of that class with attributes cadets name, rollno, section, subject (indicating Higher subject)and stores one more table with the name subject containing List of Higher Subjects of the cadets.
- Next is database in the format of class<standard> to store Attendance of the class on a particular date. Which we get after compiling

1. Compilation of the Attendance:

So if we talk about the first task of the problem, i.e, Compilation of the attendance of .xlsx files. We get the attendance of the cadets present in the class in a .xlsx file which contains the details of the cadet who was present in the class and score of his attendance.

Our job is to compile the attendance files that we got from each files into an consolidated attendance for the day. At the same time merging the attendance status of the cadets in higher subject period as they will be present only in one period. This can be done using csv file handling, openpyxl module extracting data merging and adding it to a final .xlsx file and finally coloring it and adding it to mysql in its class database.

2. Management of the Attendance compiled and stored:

The Attendance that is compile and stored in mysql as a result from the first step can be manipulated using Python-Mysql interface using mysql.connector. its all just about SQL queries and manipulating data which makes up second major task in first menu

3. Data management of cadet details.

In the database cadets_information, it contains tables with name class<standard> for ex: class12 that stores details about all cadets as described above. And this forms up the second menu of the project managing this database's tables that stores cadet details.

And hence in this way the need of the project is served and the Problem is solved.

CODE OF THE PROGRAM

(I) Program in main.py

```
import os
import datetime
import mysql.connector as myconnector
from prettytable import PrettyTable
import mysql



def refresh_screen():
    os.system('cls')
    print('-' * 20, end=' ')
    print('Online Class Attendance Compiler and Management System',
end=' ')
    print('-' * 20)



def main_menu():
    print('1: Online Class Attendance Management')
    print('2: Cadets Details Management')
    print('0: Exit')

    return choice
```

```
def get_choice(choices):
    while True:
        try:
            choice = int(input('Enter your Choice from above menu: '))
        except ValueError:
            print('Enter a Valid Integer')
        else:
            if choice not in choices:
                print('Enter a valid choice From the Menu')
                continue
            else:
                break

    return choice



def get_date():
    while True:
        date = input('Enter Date(dd-mm-yyyy) of attendance: ')
        try:
            datetime.datetime.strptime(date, '%d-%m-%Y')
        except ValueError:
            print('Enter a valid date in the dd-mm-yyyy format!')
            continue
        else:
            break

    return date
```

```
def get_fields(tablename):
    query = f'desc {tablename}'
    mycursor.execute(query)
    data = mycursor.fetchall()

    fields = []
    for i in data:
        fields.append(i[0])

    return fields


def finishing():
    input('\npress Enter to Continue')
    print('\n1: Back to Previous Menu')
    print('2: Back to Main Menu')
    print('0: Exit')
    choice = get_choice([1, 2, 0])

    return choice


def print_table(tablename, data):
    fields = get_fields(tablename)
    x = PrettyTable()
    x.field_names = fields
    for i in data:
        x.add_row(i)

    return x


def get_att_status(date, roll):
    query = f"use class{standard}"
    mycursor.execute(query)

    date_det = date.split('-')

    day = int(date_det[0])
    month = int(date_det[1])
    year = int(date_det[2])

    monthnum = {'january': 1,
                'february': 2,
                'march': 3,
                'april': 4,
                'may': 5,
                'june': 6,
                'july': 7,
                'august': 8,
                'september': 9,
                'october': 10,
                'november': 11,
                'december': 12}
    for i in monthnum.keys():
        if monthnum[i] == int(month):
            month = i.title()
            break

    tablename = f'{month}{day}_{year}'

    query = f"select * from {tablename} where Rollno = '{roll}'"
    try:
        mycursor.execute(query)
    except mysql.connector.errors.ProgrammingError:
        print('No Table Exist on the Given Date')
        return 'Error', 'Error'
    else:
        data = mycursor.fetchall()
        if data == []:
            print('No record found with the given Rollno')
            return 'Error', 'Error'
        else:
            print('Following are the Attendance Details of given Rollno
on the given date: ')
            x = print_table(tablename, data)
            print(x)

        return data, tablename



def take_att_status(period):
    print(f'Enter Attendance of the Cadet in Period {period}')
    print('1: Present')
    print('0: Absent')
    choice = get_choice([1, 0])

    if choice ==1:
        return 'Present'
    else:
        return  'Absent'

# mysql cursor initialisation
mycon = myconnector.connect(host='localhost', password='student', user='root')
mycursor = mycon.cursor()

query = 'use cadets_information'
try:
    mycursor.execute(query)
except mysql.connector.errors.ProgrammingError:
    import data



main_condition = True
while main_condition:
    try:
        refresh_screen()
        print('1: Online Class Attendance Management')
        print('2: Cadets Details Management')
        print('0: Exit')
        choice = get_choice([1, 2, 0])

        if choice == 0:
            main_condition = False

        elif choice == 1:
            print('you have chosen Online Class Attendance Management')
            sub_cond1 = True
            while sub_cond1:
                refresh_screen()
                print('1: Compile New Attendance.')
                print('2: View Previous Attendance.')
                print('3: Edit Previous Attendance of a Cadet.')
                print('4: Return to Previous Menu')
                print('0: Exit')
                choice1 = get_choice([1, 2, 3, 4, 0])

                if choice1 == 0:
                    main_condition = False
                    break


                elif choice1 == 1:
                    refresh_screen()
                    print('\nInstructions to compile New Attendance: ')
                    print('-> Save all your .xlxs files in one folder')
                    print("-> Copy this Folder's path")
                    print('-> Ensure that no other files or folders are there in that folder')
                    print("-> It is suggested to keep the filenames in
the format of classXII_Computers_P4\n   i.e in "
                          "the format of
class{standard}_{subject}_P{periodno.}")
                    print('-> Before Executing main.py Ensure that
compiler.py are in the same folder')
                    print('-> If it is first time run data.py to create
database to use the project')
                    input('\nPress Enter to Continue... ')

                    import compiler

                elif choice1 == 2:
                    while True:
                        try:
                            standard = int(input('Standard: '))
                        except ValueError:
                            print('Enter a Valid Integer!')
                        else:
                            break

                    date = get_date()
                    date_det = date.split('-')

                    day = int(date_det[0])
                    month = int(date_det[1])
                    year = int(date_det[2])

                    monthnum = {'january': 1,
                                'february': 2,
                                'march': 3,
                                'april': 4,
                                'may': 5,
                                'june': 6,
                                'july': 7,
                                'august': 8,
                                'september': 9,
                                'october': 10,
                                'november': 11,
                                'december': 12}

                    for i in monthnum.keys():
                        if monthnum[i] == int(month):
                            month = i.title()
                            break

                    tablename = f'{month}{day}_{year}'

                    query = f'use class{standard}'
                    mycursor.execute(query)

                    query = f'select * from {tablename}'
                    mycursor.execute(query)

                    data = mycursor.fetchall()

                    x = print_table(tablename, data)
```

```
                    print(f'\nAttendance of class {standard} on {date}
is as follows: ')
                    print(x)

                elif choice1 == 3:
                    print('You have chosen to Edit previous Attendance
Details of a Cadet\n')

                    while True:
                        try:
                            standard = int(input('Standard: '))
                        except ValueError:
                            print('Enter a Valid Integer!')
                        else:
                            break

                    while True:
                        try:
                            roll = int(input('Enter Rollno of the
cadet: '))
                        except ValueError:
                            print('Enter a valid Integer: ')
                        else:
                            break

                    date = get_date()
                    att_data, tablename = get_att_status(date, roll)
                    if att_data != 'Error':
                        fields = get_fields(tablename)
                        periods = fields[4: -2]
                        att_status = []
                        presents = 0

                        query = f"update {tablename} set "

                        for period in periods:
                            attstatus = take_att_status(period)
                            att_status.append(attstatus)
                            if attstatus == 'Present':
                                presents += 1

                            query += f"{period} = '{attstatus}', "

                        percentage = int((presents/len(periods))*100)
                        query += f"percentage = {percentage},
periodsattended = {presents} where rollno = '{roll}'"
                        mycursor.execute(query)

                        #calculate avg percentage
                        query = f"select sum(percentage) from
{tablename} where rollno != '->'"
                        mycursor.execute(query)
                        total_perc = int(mycursor.fetchall()[0][0])
                        mycon.commit()

                        query = f"select * from {tablename}"
                        mycursor.execute(query)
                        data = mycursor.fetchall()
                        cadets_strength = len(data)-1
                        avg_perc = total_perc/cadets_strength

                        query = f"update {tablename} set percentage =
{avg_perc} where rollno = '->'"
                        mycursor.execute(query)

                        print('Successfully Edited!')

                        query = f"select * from {tablename} where
rollno = '{roll}'"
                        mycursor.execute(query)
                        data = mycursor.fetchall()
                        x = print_table(tablename, data)
                        print(x)

                        mycon.commit()

                elif choice1 == 4:
                    break

                choice2 = finishing()
                if choice2 == 0:
                    main_condition = False
                    break
                elif choice2 == 1:
                    continue
                elif choice2 == 2:
                    break
```

```
elif choice == 2:
    print('You have chosen Cadet Details Management')
    while True:
        try:
            standard = int(input('Standard: '))
        except ValueError:
            print('Enter a Valid Integer!')
        else:
            break
    sub_cond2 = True
    query = 'use cadets_information'
    mycursor.execute(query)
    while sub_cond2:
        refresh_screen()
        print(f'\n1: See Details of class {standard}')
        print('2: See Details of a cadet')
        print('3: See Details of multiple cadets')
        print('4: Edit Details of a cadet')
        print('5: Add a New Cadet')
        print('6: Delete a Cadet')
        print('7: Return to Previous Menu')
        print('0: Exit')
        choice3 = get_choice([1, 2, 3, 4, 5, 6, 7, 0])

        if choice3 == 0:
            main_condition = False
            break
```

```
                elif choice3 == 7:
                    print('Return to previous Menu...')
                    input('Press Enter ')
                    break

                elif choice3 == 1:
                    query = 'use cadets_information'
                    mycursor.execute(query)

                    query = f'select Rollno, name, section, subject from class{standard} natural join subject order by rollno'
                    mycursor.execute(query)
                    data = mycursor.fetchall()

                    fields = get_fields(f'class{standard}')
                    fields[-1] = 'subject'
                    x = PrettyTable()
                    x.field_names = fields
                    for i in data:
                        x.add_row(i)

                    print(f'\nDetails of Class {standard} cadets is as follows: ')
                    print(x)

                elif choice3 == 2:
                    roll = int(input('Enter Roll of the cadet: '))
                    query = f"select Rollno, name, section, subject from class{standard} natural join subject where rollno = '{roll}' order by rollno"
                    mycursor.execute(query)
                    data = mycursor.fetchone()

                    fields = get_fields(f'class{standard}')
                    fields[-1] = 'Subject'

                    x = PrettyTable()
                    x.field_names = fields
                    if data == None:
                        print(f'No Cadet Found with Roll no {roll}')
                    else:
                        x.add_row(data)
                        print(x)



                elif choice3 == 3:
                    print('\nYou have chosen to see details of multiple cadets.')
                    count = int(input('How many no.of cadets details you would like to see?: '))
                    rolls = []
                    for i in range(count):
                        while True:
                            try:
                                roll = int(input(f'\nEnter RollNo of Cadet{i + 1}: '))
                            except ValueError:
                                print('Enter a Valid Integer')
                            else:
                                rolls.append(roll)
                                break

                    query = f'select rollno, name, section, subject
from class{standard} natural join subject where rollno in
{tuple(rolls)} order by rollno'
                    mycursor.execute(query)
                    data = mycursor.fetchall()

                    fields = get_fields(f'class{standard}')
                    fields[-1] = 'subject'
                    x = PrettyTable()
                    x.field_names = fields
                    foundrolls = []

                    if data == None:
                        print('No Cadet if found with any of the given
Rolls')
                    else:
                        for i in data:
                            x.add_row(i)
                            foundrolls.append(i[0])

                        if len(foundrolls) != len(rolls):
                            j = PrettyTable()
                            j.field_names = ['Rollno']
                            for roll in rolls:
                                if roll not in foundrolls:
                                    j.add_row([roll])
                            print("Following Rolls did not match with
any Cadet's info")
```

```
                            print(j)

                        print('Following Details are Found')
                        print(x)


                elif choice3 == 4:
                    print('\nYou have chosen to Edit Details of a
Cadet')
                    while True:
                        try:
                            roll = int(input('\nEnter Rollno of Cadet
whose details you want to edit: '))
                        except ValueError:
                            print('Enter a Valid Integer!')
                        else:
                            break

                    query = f"select rollno, name, section, subject
from class{standard} natural join subject where rollno = '{roll}'"
                    mycursor.execute(query)
                    data = mycursor.fetchall()

                    if data == None:
                        print(f'No cadet Found with Rollno {roll}')
                    else:
                        fields = get_fields(f'class{standard}')
                        fields[-1] = 'subject'
                        x = PrettyTable()
                        x.field_names = fields
                        for i in data:
                            x.add_row(i)


                        print('Details are as follows: ')
                        print(x)
                        confirm = input(f'Do you want to proceed to
Edit Details of cadet {data[0][1]} Attendance (y/n): ').lower()
                        if confirm != 'y':
                            print('You have chosen to cancel the
process..')
                        else:
                            values = ['Telugu', 'Hindi', 'Computer
Science', 'Biology']
                            keys = [1, 2, 3, 4]
                            rollno = int(input('Enter Rollno: '))
                            name = input('Enter Name:
').title().strip()
                            section = input('Enter Section:
').upper().strip()
                            print('Select Subject: ')
                            for i in range(len(values)):
                                print(f'{keys[i]}: {values[i]}')
                            subid = get_choice([1, 2, 3, 4])


                            query = f"update class{standard} set rollno
= '{rollno}', name = '{name}', section = '{section}', sub_id = {subid}
where rollno = '{roll}'"
                            try:
                                mycursor.execute(query)
                            except
mysql.connector.errors.IntegrityError:
```

```
                                print(f'Already a cadets information
exists with rollno {rollno}')
                            else:
                                mycon.commit()

                                query = f"select rollno, name, section,
subject from class{standard} natural join subject where rollno =
'{rollno}'"
                                mycursor.execute(query)
                                data = mycursor.fetchall()

                                fields = get_fields(f'class{standard}')
                                x = PrettyTable()
                                fields[-1] = 'subject'
                                x.field_names = fields
                                for i in data:
                                    x.add_row(i)
                                print('updated Successfully')
                                print(x)



                elif choice3 == 5:
                    print('You have chosen to Add a New Cadet.')

                    values = ['Telugu', 'Hindi', 'Computer Science',
'Biology']
                    keys = [1, 2, 3, 4]
                    rollno = int(input('Enter Rollno: '))
                    name = input('Enter Name: ').title().strip()
                    section = input('Enter Section: ').upper().strip()
                    print('Select Subject: ')
                    for i in range(len(values)):
                        print(f'{keys[i]}: {values[i]}')
                    subid = get_choice([1, 2, 3, 4])
                    subject = values[subid-1]

                    query = f"insert into class{standard} values
('{rollno}', '{name}', '{section}', {subid})"
                    try:
                        mycursor.execute(query)
                    except mysql.connector.errors.IntegrityError:
                        print(f'Cadet Already Exists with Rollno
{rollno}')
                        input('Press Enter ')
                        continue
                    except mysql.connector.errors.ProgrammingError:
                        print('Enter a Valid Input!')
                        input('Press Enter ')
                        continue

                    else:
                        fields = get_fields(f'class{standard}')
                        fields.append('subject')
                        x = PrettyTable()
                        x.field_names = fields
                        for i in [[rollno, name, section, subid,
subject]]:
                            x.add_row(i)
                        print(x)
                        mycon.commit()
                        print('Successfully Added New Cadet Details')



                elif choice3 == 6:
                    print('You have chosen to Delete a Cadets details')
                    rollno = int(input('Enter Rollno: '))

                    query = f"select rollno, name, section, sub_id,
subject from class{standard} natural join subject where rollno =
'{rollno}'"
                    mycursor.execute(query)
                    data = mycursor.fetchall()

                    if data == None:
                        print(f'No Cadet if Found with Rollno
{rollno}')
                    else:
                        fields = get_fields(f'class{standard}')
                        fields.append('subject')
                        x = PrettyTable()
                        x.field_names = fields
                        for i in data:
                            x.add_row(i)
                        print(f'Details of Rollno {rollno}: ')
                        print(x)

                        confirm = input('would you like to
proceed?(y/n): ').lower()
                        if confirm != 'y':
                            print('You have chosen to cancel the
deleting process...')

                        else:
                            query = f"delete from class{standard} where
rollno = '{rollno}'"
                            mycursor.execute(query)
                            print('Successfully Deleted')
                            mycon.commit()

                choice4 = finishing()
                if choice4 == 0:
                    main_condition = False
                    break
                elif choice4 == 1:
                    continue
                elif choice4 == 2:
                    break
    except ValueError:
        print('Enter a Valid Input!')
        input('Press Enter: ')
        continue

    except mysql.connector.errors.ProgrammingError:
        print('No Attendance is recorded on that Date!')
        input('Press Enter: ')
    except:
        print('Some Error Occurred!')
        input('Press Enter: ')
        continue
    else:
        continue

print('\n')
print('-' * 20, end=' ')
print('Online Class Attendance Compiler and Management System', end='
')
print('-' * 20)

mycon.commit()
mycon.close()
```

(II) Program in compiler.py

```
import os
import csv
import shutil
import mysql.connector as myconnector
import openpyxl
from openpyxl.styles import PatternFill
from openpyxl.styles.borders import Border, Side
from datetime import date as DATE


def get_finalfiles():
    if 'finalfiles' in os.listdir(folderpath):
        shutil.rmtree(rf'{folderpath}\finalfiles')

    os.mkdir(rf'{folderpath}\finalfiles')


def create_files(file, list):
    with open(file, 'w') as f:
        f.writelines(list)
```

```
def delete_txtfiles():
    os.remove(rf'{folderpath}\rolls.txt')
    os.remove(rf'{folderpath}\names.txt')
    os.remove(rf'{folderpath}\section.txt')
    if standard > 10:
        os.remove(rf'{folderpath}\biology.txt')
        os.remove(rf'{folderpath}\computer.txt')
    else:
        os.remove(rf'{folderpath}\telugu.txt')
        os.remove(rf'{folderpath}\hindi.txt')


def getfiles_fromsql():
    query = f"select * from class{standard}"
    mycursor.execute(query)
    data = mycursor.fetchall()
    rolls = [f'{i[0]}\n' for i in data]
    names = [f'{i[1]}\n' for i in data]
    sections = [f'{i[2]}\n' for i in data]
    biorolls = [f'{i[0]}\n' for i in data if i[3] in [2, 4]]
    comprolls = [f'{i[0]}\n' for i in data if i[3] in [1, 3]]

    create_files(rf'{folderpath}\rolls.txt', rolls)
    create_files(rf'{folderpath}\names.txt', names)
    create_files(rf'{folderpath}\section.txt', sections)
    if standard > 10:
        create_files(rf'{folderpath}\biology.txt', biorolls)
        create_files(rf'{folderpath}\computer.txt', comprolls)
    else:
        create_files(rf'{folderpath}\hindi.txt', biorolls)
        create_files(rf'{folderpath}\telugu.txt', comprolls)



def getreader(filename):
    with open(rf'{filename}') as f:
        readerob = csv.reader(f)
        reader = []
        for i in readerob:
            try:
                int(i[0].split()[-1])
            except ValueError:
                try:
                    int(i[0].split()[0])
                except ValueError:
                    pass
                else:
                    if i[0].split()[0].lower() != 'members':
                        reader.append(i)
            except IndexError:
                pass
            else:
                if i[0].split()[0].lower() != 'members':
                    reader.append(i)

        return reader
```

```
def getrollsfromfile(rollfile):
    roll_list = []
    with open(rf'{folderpath}\{rollfile}') as rolls:
        lines = rolls.readlines()
        for line in lines:
            roll = int(line.strip())
            roll_list.append(roll)
    return roll_list


def getmergerolls(standard):
    if standard > 10:
        comprolls = getrollsfromfile(rf'computer.txt')
        biorolls = getrollsfromfile(rf'biology.txt')
    elif standard <= 10:
        comprolls = getrollsfromfile(rf'telugu.txt')
        biorolls = getrollsfromfile(rf'hindi.txt')
    return biorolls, comprolls


def getreqdata(data, rollslist):
    for i in data:
        try:
            rollno = int(i[0].split()[-1])
        except ValueError:
            try:
                rollno = int(i[0].split()[0])
            except ValueError:
                pass

        if rollno in rollslist:
            finaldata.append(i)



def mergefiles(filenames, standard):
    biorolls, comprolls = getmergerolls(standard)
    period = int(input('Enter Period Number: '))
    if standard > 10:
        for filename in filenames:
            if 'bio' in filename.lower():
                bioname = filename
            elif 'comp' in filename.lower():
                compname = filename

        finalfilename = f'P{period}BioComputers.csv'

    elif standard <= 10:
        for filename in filenames:
            if 'hin' in filename.lower():
                bioname = filename
            elif 'tel' in filename.lower():
                compname = filename

        finalfilename = f'P{period}TeluguHindi.csv'
    biodata = getreader(bioname)
    compdata = getreader(compname)

    getreqdata(biodata, biorolls)
    getreqdata(compdata, comprolls)

    with open(rf'{folderpath}\file.csv', 'w', newline='') as finalfile:
        writerobj = csv.writer(finalfile)
        writerobj.writerows(finaldata)

    os.rename(rf'{folderpath}\file.csv',
rf'{folderpath}\{finalfilename}')

    return bioname, compname



def getdate():
    today = input("Today's Attendance? (y/n): ").lower()
    if today == 'y':
        date = DATE.today()
        date = str(date)
        date = date.split('-')
        date.reverse()
        datestr = ''
        for i in date:
            datestr += f'{i}.'
        datestr = datestr[-2::-1][::-1]
    else:
        date = input('Enter Date of Attendance: (dd-mm-yyyy): ')
        datestr = date.replace('-', '.')

    return datestr



def get_noof_files(folderpath):
    '''Get all the xl files in the folder'''

    # Your current directory
    mydir = folderpath
    mydir.replace("\\", '/')
    mydir += '/'

    # Get all excel files include subdir
    filelist = []
    ffileslist = []
    for path, subdirs, files in os.walk(mydir):
        for file in files:
            if (file.endswith('.xlsx') or file.endswith('.xls') or
file.endswith('.XLS')) or file.endswith('.csv'):
                filelist.append(os.path.join(path, file))
    for file in filelist:
        files_sub = file.split('/')
        ffileslist.append(files_sub[-1])
    return filelist



def readxldata(ws):
    filedata4csv = []
    # starting cells that we get from the extension xl file
    rows = int(ws['A3'].value.split(':')[-1].strip())
    stringscr = ws['B7'].value
    score, getcolumns = getscore(stringscr)  # no use of score in this
function
    columns = int(getcolumns.split('/')[-1].strip())
    for row in range(7, rows + 7):
        rowval = []
        for column in range(1, columns + 3):
            if column <= 2:
                cellval = ws.cell(row=row, column=column).value
            elif column > 2:
                cellval = '?'
            rowval.append(cellval)
        filedata4csv.append(rowval)

    return filedata4csv



def convert_xl_to_csv(filenames):
    for file in filenames:
        filename = f'{file[-6::-1][::-1]}.csv'
        wb = openpyxl.load_workbook(rf'{file}')
        ws = wb[wb.sheetnames[0]]
        filedata4scv = readxldata(ws)  # get data to write into csv
        with open(rf'{filename}', 'w', newline='') as csvfile:
            writer = csv.writer(csvfile)
            writer.writerows(filedata4scv)
        os.remove(file)



def get_students_dictionary():
    with open(rf'{folderpath}\section.txt') as f1:
        sectionlist = f1.readlines()
    with open(rf'{folderpath}\rolls.txt') as f:
        i = 0
        for line in f:
            roll = int(line.strip())
            student_status.append({'roll': roll, 'section':
sectionlist[i].strip().upper()})
            i += 1

    with open(rf'{folderpath}\names.txt') as f2:
        names = f2.readlines()
    for i in range(len(student_status)):
        student_status[i]['name'] = names[i].strip().title()



def getscore(strscore):
    monthnum = {'january': 1,
                'february': 2,
                'march': 3,
                'april': 4,
                'may': 5,
                'june': 6,
                'july': 7,
                'august': 8,
                'september': 9,
                'october': 10,
                'november': 11,
                'december': 12}
    nums = strscore.split('/')
    if len(nums) != 2:
        # 02-Feb
        nums = strscore.split('-')
        for i in monthnum.keys():
            if nums[1] in i:
                nums.append(monthnum[i])
                del nums[1]

    score = (int(nums[0]) / int(nums[1])) * 100
    scorestr = f'{int(nums[0])}/{int(nums[1])}'

    return score, scorestr



def markattendance(boolean, subname, studentinfo):
    for stinfo in student_status:
        try:
            if int(stinfo['roll']) == int(studentinfo[0].split()[-1]):
                if boolean:
                    stinfo[subname] = 'present'
                else:
                    stinfo[subname] = 'absent'
        except IndexError:
            pass
        except ValueError:
            try:
                if int(stinfo['roll']) ==
int(studentinfo[0].split()[0]):
                    if boolean:
                        stinfo[subname] = 'present'
                    else:
                        stinfo[subname] = 'absent'
            except ValueError:
                pass



def markpureabsentees(subname):
    for stinfo in student_status:
        try:
            status = stinfo[subname]
        except KeyError:
            stinfo[subname] = 'absent'



def get_attendancefile(filename):
    subname = filename[-5::-1][::-1]
    reader = getreader(filename)
    for student_att in reader:
        score, scorestr = getscore(student_att[1])  # scorestr no use
in this function
        if score >= 0:
            markattendance(True, subname, student_att)
        else:
            markattendance(False, subname, student_att)

    markpureabsentees(subname)



def get_attendance(filenames):
    for csvfile in filenames:
        get_attendancefile(csvfile)



def calcstats(filenames):
    total_periods = len(filenames)
    for i in student_status:
        periods_attended = 0
        for subname in filenames:
            if i[subname[-5::-1][::-1]] == 'present':
                periods_attended += 1
        percentage = (periods_attended / total_periods) * 100
        i['total attended'] = periods_attended
        i['percentage'] = percentage



def writedata():
    filenames = get_noof_files(folderpath)
    with open(rf'{folderpath}\finalfiles\final.csv', 'w', newline='')
as finalfile:
        writerobj = csv.writer(finalfile)
        data = ['S.No', 'RollNo', 'CdtName', 'Class', 'Sec']
        for i in filenames:
            i = i.replace(folderpath, '')
            i = i[1:]
            names = i.split('_', 1)
            i = names[-1]
            data.append(i[-5::-1][::-1].title())
        data.append('PeriodsAttended')
        data.append('Attendance %')
        writerobj.writerow(data)

        for i in range(len(student_status)):
            data2 = [i + 1, student_status[i]['roll'],
student_status[i]['name'], standard,
                     student_status[i]['section']]
            for name in filenames:
                data2.append(student_status[i][name[-5::-1][::-
1]].title())
            data2.append(student_status[i]['total attended'])
            data2.append(student_status[i]['percentage'])
            writerobj.writerow(data2)

        data3 = [' ', ' ', 'Total Cadets Present (Period Wise)', ' ', '
']
        for subject in filenames:
            total_attended = 0
            subname = subject[-5::-1][::-1]
            for studentinfo in student_status:
                if studentinfo[subname] == 'present':
                    total_attended += 1
            data3.append(total_attended)
        data3.append('Class Average Attendance')
        totalperc = 0
        for studentdetail in student_status:
            totalperc += studentdetail['percentage']
        averageperc = int(totalperc / len(student_status))
        data3.append(averageperc)
        writerobj.writerow(data3)



def renamefinal():
    date = getdate()
    filename = f'{standard}({date}).csv'
    os.rename(rf'{folderpath}\finalfiles\final.csv',
rf'{folderpath}\{filename}')
    return filename, date



def convert_csv_to_xl(filename):
    wb = openpyxl.Workbook()
    ws = wb.active
    with open(rf'{folderpath}\{filename}', 'r') as f:
        for row in csv.reader(f):
            ws.append(row)
    xlname = f'{filename[-5::-1][::-1]}.xlsx'
    wb.save(rf'{folderpath}\{xlname}')
    os.remove(rf'{folderpath}\{filename}')
    return xlname
```

```
def openxl(xlname):
    wb = openpyxl.load_workbook(rf'{folderpath}\{xlname}')
    ws = wb[wb.sheetnames[0]]
    return wb, ws


def getcellnumber(row, column, ws):
    value = str(ws.cell(row=row, column=column))
    cell = value.split('.')
    cell1 = cell[-1]
    cell2 = cell1[-2::-1][::-1]
    return cell2


def get_absent_cellnumber(ws):
    strength = len(student_status)
    noofperiods = len(get_noof_files(folderpath)) - 1
    absent_cellnumbers = []
    for i in range(2, strength + 2):
        for j in range(6, 6 + noofperiods):
            if ws.cell(row=i, column=j).value.lower() == 'absent':
                cellnumber = getcellnumber(i, j, ws)
                absent_cellnumbers.append(cellnumber)

    return absent_cellnumbers


def boxesbetweenrange(rangeequal, uplim, lowlim, ws):
    column = len(student_status[0]) + 2
    strength = len(student_status) + 1
    boxeslist = []
    for row in range(2, strength + 2):
        if not rangeequal:
            if float(ws.cell(row=row, column=column).value) >= lowlim
and float(
                    ws.cell(row=row, column=column).value) < uplim:
                cellnumber = getcellnumber(row, column, ws)
                boxeslist.append(cellnumber)
        else:
            if float(ws.cell(row=row, column=column).value) == uplim:
                cellnumber = getcellnumber(row, column, ws)
                boxeslist.append(cellnumber)
    return boxeslist



def colour_boxes(boxes_list, colour, wb, ws):
    for i in boxes_list:
        fillpattern = PatternFill(patternType='solid', fgColor=colour)
        ws[str(i)].fill = fillpattern



def colour_headers(wb, ws):
    permanent_ones = ['A1', 'B1', 'C1', 'D1', 'E1']
    noofperiods = len(get_noof_files(folderpath)) - 1
    column = 6
    period_headboxes = []
    for i in range(6, noofperiods + 8):
        cellnumber = getcellnumber(1, i, ws)
        period_headboxes.append(cellnumber)

    colour_boxes(permanent_ones, 'ffbf00', wb, ws)
    colour_boxes(period_headboxes, 'ffff00', wb, ws)


def colourabsent(absent_cellnumbers):
    colour_boxes(absent_cellnumbers, 'ffc6cd', wb, ws)


def colouring(att0perce_boxes, att_below20perce_boxes, 
att_above20perce_boxes, att_above40perce_boxes,
              att_above60perce_boxes, att_above75perce_boxes, 
att100perce_boxes, wb, ws, xlname):
    colour_boxes(att100perce_boxes, '62bd7b', wb, ws)
    colour_boxes(att_above75perce_boxes, 'ffeb83', wb, ws)
    colour_boxes(att_above60perce_boxes, 'fdca7c', wb, ws)
    colour_boxes(att_above40perce_boxes, 'fdaa77', wb, ws)
    colour_boxes(att_above20perce_boxes, 'f98970', wb, ws)
    colour_boxes(att_below20perce_boxes, 'ff2e00', wb, ws)
    colour_boxes(att0perce_boxes, 'f7696b', wb, ws)


def border(ws):
    # ws.cell(row=3, column=2).border = thin_border
    thin_border = Border(left=Side(style='thin'),
                         right=Side(style='thin'),
                         top=Side(style='thin'),
                         bottom=Side(style='thin'))

    columns = len(student_status[0]) + 2
    strength = len(student_status) + 2

    for row in range(1, strength + 1):
        for column in range(1, columns + 1):
            ws.cell(row=row, column=column).border = thin_border



def delcsvs():
    filenames = get_noof_files(folderpath)
    for filename in filenames:
        name, file_extension = os.path.splitext(filename)
        if file_extension == '.csv':
            os.remove(filename)



def add_att_to_sql(csvname):
    csv_file = rf'{folderpath}\\{csvname}'
    query = f'create database if not exists class{standard}'
    mycursor.execute(query)

    monthnum = {'january': 1,
                'february': 2,
                'march': 3,
                'april': 4,
                'may': 5,
                'june': 6,
                'july': 7,
                'august': 8,
                'september': 9,
                'october': 10,
                'november': 11,
                'december': 12}
    lis = str(date).split('.')
    mon = int(lis[1])
    for i in monthnum.keys():
        if monthnum[i] == int(mon):
            month = i.title()
            break

    day = int(lis[0])
    year = int(lis[2])
    tablename = f'{month}{day}_{year}'

    query = f'use class{standard}'
    mycursor.execute(query)

    query = f'drop table if exists {tablename}'
    mycursor.execute(query)

    query = f'create table if not exists {tablename} (Rollno
varchar(10))'
    mycursor.execute(query)

    with open(csv_file) as f:
```

```
    reader = csv.reader(f)
    data = list(reader)

for i in data:
    del i[0]

addcolumns = []
for i in data[0][1:len(data[0]) - 1]:
    addcolumns.append(i)

for i in addcolumns:
    query = f'alter table {tablename} add column {i} varchar(35)'
    mycursor.execute(query)

query = f'alter table {tablename} add column percentage int(10)'
mycursor.execute(query)

for i in data[1:]:
    row = i
    query = f'insert into {tablename} values('
    if row[0] != ' ':
        query += f"'{row[0]}',"
    else:
        query += f"'->',"
    for details in row[1:len(row) - 1]:
        query += f"'{details}',"
    query += f'{row[-1]})'
    mycursor.execute(query)
    return tablename



# __main__


# folderpath, standard, finaldata, student_status


try:
    folderpath = input('\nEnter Folder Path: ')
    get_finalfiles()
    standard = int(input('Standard(Enter the Number): '))
    database = 'cadets_information'

    finaldata = []

    student_status = []

    # mysql cursor initialisation
    mycon = myconnector.connect(host='localhost', password='student',
user='root', database=database)
    mycursor = mycon.cursor()
    getfiles_fromsql()

    filenames = get_noof_files(folderpath)  # xl filenames in the
directory
    convert_xl_to_csv(filenames)  # converts xl to csv
    filenames = get_noof_files(folderpath)
    if standard > 10:
        compbios = input('comp/bio separate files are there?(y or n): 
').lower()
    elif standard <= 10:
        compbios = input('Telugu/Hindi separate file are there(y or 
n)?: ').lower()

    if compbios == 'y':
        bioname, compname = mergefiles(filenames, standard)
        os.remove(bioname)
        os.remove(compname)

    get_students_dictionary()
    filenames = get_noof_files(folderpath)
    get_attendance(filenames)
    calcstats(filenames)
    writedata()
    filename, date = renamefinal()
    tablename = add_att_to_sql(filename)
    xlname = convert_csv_to_xl(filename)
    wb, ws = openxl(rf'{xlname}')
    absent_cellnumbers = get_absent_cellnumber(ws)
    att100perce_boxes = boxesbetweenrange(True, 100, 100, ws)
    att_above75perce_boxes = boxesbetweenrange(False, 100, 75, ws)
    att_above60perce_boxes = boxesbetweenrange(False, 75, 60, ws)
    att_above40perce_boxes = boxesbetweenrange(False, 60, 40, ws)
    att_above20perce_boxes = boxesbetweenrange(False, 40, 20, ws)
    att_below20perce_boxes = boxesbetweenrange(False, 20, 0, ws)
    att0perce_boxes = boxesbetweenrange(True, 0, 0, ws)
    colour_headers(wb, ws)
    colourabsent(absent_cellnumbers)
    colouring(att0perce_boxes, att_below20perce_boxes,
att_above20perce_boxes, att_above40perce_boxes,
              att_above60perce_boxes, att_above75perce_boxes,
att100perce_boxes, wb, ws, xlname)
    border(ws)
    wb.save(rf'{folderpath}\\finalfiles\\{xlname}')
    os.remove(rf'{folderpath}\{xlname}')
    delcsvs()
    delete_txtfiles()

    print(f'Successfully Compiled, Check Finalfiles folder for the file
with name {xlname}')
    print(f'Successfully Added table to sql, check table {tablename} in
database class{standard}\n')

except ValueError:
    print('Enter a Valid Input!')

except:
    print('Some Error Occurred')

mycon.commit()
mycon.close()
```

(III) Program in data.py

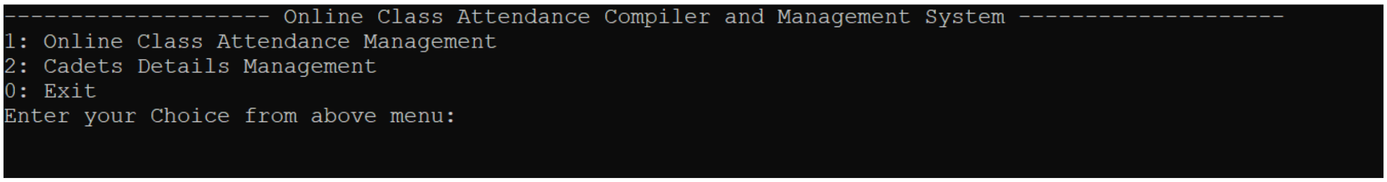
```
# This code Gets Executed only if the required databases are not
present in mysql.
import mysql.connector as myconnector


mycon = myconnector.connect(host='localhost', user='root',
password='student')
mycur = mycon.cursor()
create_database = 'create database if not exists cadets_information'
mycur.execute(create_database)
query = 'use cadets_information'
mycur.execute(query)
create_table = 'create table if not exists class12 (Rollno int(10)
primary key, Name varchar(30) not null, ' \
               'Section varchar(10) not null, sub_id int(10) not null)
'
mycur.execute(create_table)


queries = ["insert into class12 values(106, 'Swamireddy Deganath
Reddy', 'B', 3)",
           "insert into class12 values(107, 'Shourya', 'A', 4)",
           "insert into class12 values(109, 'Arrireddy Somesh Reddy',
'B', 3)",
           "insert into class12 values(110, 'Singu Surya', 'C', 3)",
           "insert into class12 values(111, 'Navneet Nischhal', 'A',
4)",
           "insert into class12 values(112, 'Rahul Kumar', 'A', 4)",
           "insert into class12 values(113, 'Somu Kumar', 'A', 4)",
           "insert into class12 values(114, 'Mittapally Sai Harsheeth',
'B', 3)",
           "insert into class12 values(115, 'Pavuluri Veda Vihar', 'A',
4)",
           "insert into class12 values(116, 'Golla Sumanth Yadav', 'A',
4)",
           "insert into class12 values(117, 'K Harsha Venkateswara
Chowdary', 'C', 3)",
           "insert into class12 values(118, 'Khatravath Charan Rathod',
'B', 3)",
           "insert into class12 values(119, 'Piyush Dev', 'A', 4)",
           "insert into class12 values(120, 'Nalli Hashish Reddy', 'C',
3)",
           "insert into class12 values(121, 'M Vedavyas Karthikeyan
Babu', 'A', 4)",
           "insert into class12 values(123, 'Kowkutla Vamshidhar
Reddy', 'A', 4)",
           "insert into class12 values(124, 'Satyam Harsha', 'B', 3)",
           "insert into class12 values(126, 'Satyam Kumar', 'C', 3)",
           "insert into class12 values(127, 'Chintakuntla Rudheer
Reddy', 'B', 3)",
           "insert into class12 values(128, 'Ankit Kumar', 'C', 3)",
           "insert into class12 values(129, 'Vikash Kumar', 'A', 4)",
           "insert into class12 values(130, 'Sharma Kunal', 'A', 4)",
           "insert into class12 values(131, 'Dasari Hari Krishna', 'B',
3)",
           "insert into class12 values(132, 'Rohit Raj', 'C', 3)",
           "insert into class12 values(133, 'Avula Venkata
Puneethkumar', 'B', 3)",
           "insert into class12 values(134, 'Nimma Harshit Reddy', 'C',
3)",
           "insert into class12 values(136, 'Diwakar Kumar', 'A', 4)",
           "insert into class12 values(137, 'Abhay Kumar', 'A', 4)",
           "insert into class12 values(138, 'Kyamaji Yashovardhan',
'B', 3)",
           "insert into class12 values(139, 'Amritesh Anand', 'A', 4)",
           "insert into class12 values(140, 'Aman Yadvendu', 'C', 3)",
           "insert into class12 values(141, 'Suguri Bharath', 'B', 3)",
           "insert into class12 values(142, 'Samarjeet Singh', 'A',
4)",
           "insert into class12 values(143, 'Saurabh Kumar', 'A', 4)",
           "insert into class12 values(144, 'Abdul Aslam', 'C', 3)",
           "insert into class12 values(145, 'Puranika K N Rishab', 'B',
3)",
           "insert into class12 values(147, 'Boggarapu Trilok', 'A',
4)",
           "insert into class12 values(148, 'Ravi Kumar', 'A', 4)",
           "insert into class12 values(149, 'Mohit Prakash', 'A', 4)",
           "insert into class12 values(150, 'Rohit Kumar', 'A', 4)",
           "insert into class12 values(151, 'Severthy Thrilokraju',
'A', 4)",
           "insert into class12 values(152, 'Kumar Swaraj', 'A', 4)",
           "insert into class12 values(153, 'Bodupally Vijay Kumar',
'A', 4)",
           "insert into class12 values(154, 'Kranthi Kumar', 'A', 4)",
           "insert into class12 values(156, 'Sadhu Sri Nandan Reddy',
'A', 4)",
           "insert into class12 values(157, 'Kuruba Harshavardhan',
'C', 3)",
           "insert into class12 values(159, 'K Lakshmi Narasimha Sai
Akash', 'B', 3)",
           "insert into class12 values(160, 'Madineni Rohith', 'C',
3)",
           "insert into class12 values(161, 'Parakala Dileep Kumar',
'B', 3)",
           "insert into class12 values(162, 'Suguru Nithin Sai Yadav',
'C', 3)",
           "insert into class12 values(163, 'Meka Sumanth Kumar', 'B',
3)",
           "insert into class12 values(164, 'Thota Rishik Sai
Santhosh', 'C', 3)",
           "insert into class12 values(165, 'Surya Lokesh Kumar', 'B',
3)",
           "insert into class12 values(166, 'Peshwakar Srikanth', 'C',
3)",
           "insert into class12 values(167, 'Pankaj Kumar', 'B', 3)",
           "insert into class12 values(168, 'Syed Mohinuddin', 'A',
4)",
           "insert into class12 values(169, 'Rasamala Chandu Kumar',
'C', 3)",
           "insert into class12 values(170, 'Vishwadeep Kumar', 'A',
4)",
           "insert into class12 values(172, 'Abhay Kumar', 'B', 3)",
           "insert into class12 values(173, 'Manish Rai', 'A', 4)",
           "insert into class12 values(175, 'Bodipedda Anand Blesson',
'C', 3)",
           "insert into class12 values(176, 'Avinash Kumar', 'A', 4)",
           "insert into class12 values(177, 'Siragam Venkata Krishna',
'B', 3)",
           "insert into class12 values(178, 'M Venkata Shisundernath
Reddy', 'C', 3)",
           "insert into class12 values(180, 'Madas Johnson', 'B', 3)",
           "insert into class12 values(181, 'Shubham Kumar', 'A', 4)",
           "insert into class12 values(182, 'Nadimikandriga Ruthvik
Kumar', 'A', 4)",
           "insert into class12 values(184, 'Malingappa Gari Akhilesh',
'A', 4)",
           "insert into class12 values(186, 'Kanipakam Aman Thoyaj
Krishna', 'A', 4)",
           "insert into class12 values(187, 'Bollu Sujith', 'C', 3)",
           "insert into class12 values(189, 'Rishabh Kumar', 'B', 3)",
           "insert into class12 values(190, 'Maloth Sri Krishna
Chandra', 'C', 3)",
           "insert into class12 values(191, 'Arepalli Ranjith Kumar',
'B', 3)",
           "insert into class12 values(192, 'M Rajesh Naik', 'C', 3)",
           "insert into class12 values(193, 'Dandu Phanidhar Mithra',
'B', 3)",
           "insert into class12 values(194, 'Virat Raj', 'C', 3)",
           "insert into class12 values(195, 'Gandham Khadar Babu', 'B',
3)",
           "insert into class12 values(196, 'Mudavath Ravi Kishore',
'A', 4)",
           "insert into class12 values(197, 'Muppireddy Yogeswar
Reddy', 'C', 3)",
           "insert into class12 values(198, 'Dharmasoth Kumar', 'B',
3)",
           "insert into class12 values(199, 'Kelawath Deva Nayak', 'C',
3)",
           "insert into class12 values(200, 'Banavath Bharath', 'A',
4)",
           "insert into class12 values(201, 'Sareddy Rohith Reddy',
'B', 3)",
           "insert into class12 values(202, 'Garadam Palli Akash', 'C',
3)",
           "insert into class12 values(203, 'Ummadi Vikash Kumar
Reddy', 'A', 4)",
           "insert into class12 values(204, 'Medapally Vignesh', 'B',
3)",
           "insert into class12 values(206, 'Surakanti Advaith Reddy',
'C', 3)",
           "insert into class12 values(207, 'Modagala Vinodkumar', 'A',
4)",
           "insert into class12 values(209, 'Thirunagiri Dhanush', 'B',
3)",
           "insert into class12 values(439, 'Gummadi Charan Sai', 'C',
3)",
           "insert into class12 values(440, 'Annareddy Dheeraj Reddy',
'A', 4)",
           "insert into class12 values(441, 'Guvvala Praveen Kumar',
'A', 4)",
           "insert into class12 values(442, 'Vasu Paul Jayakar', 'B',
3)",
           "insert into class12 values(443, 'Thummalapalli Pavan Teja',
'C', 3)",
           "insert into class12 values(515, 'Paramjyothi Ram Charan
Tej', 'B', 3)",
           "create table if not exists subject (sub_id int(10) primary
key, subject varchar(30) not null)",
           "insert into subject values(1, 'Telugu')",
           "insert into subject values(2, 'Hindi')",
           "insert into subject values(3, 'Computer Science')",
           "insert into subject values(4, 'Biology')"]


for query in queries:
    try:
        mycur.execute(query)


    except:
        pass
```

```
mycon.commit()

mycon.close()
```

____________ END OF THE CODE ________________

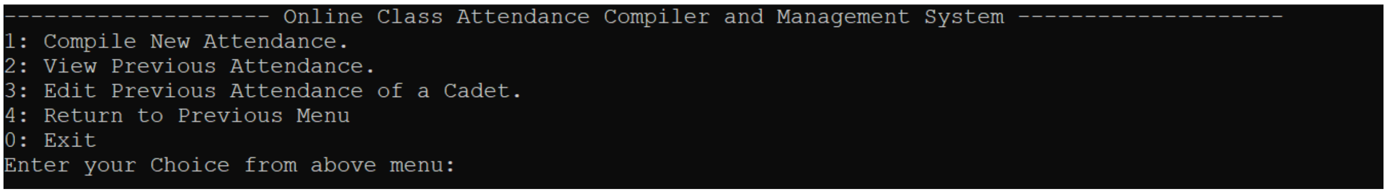
OUTPUT OF THE PROGRAM

Opening screen:-



```
-------------------- Online Class Attendance Compiler and Management System --------------------
1: Online Class Attendance Management
2: Cadets Details Management
0: Exit
Enter your Choice from above menu:
```

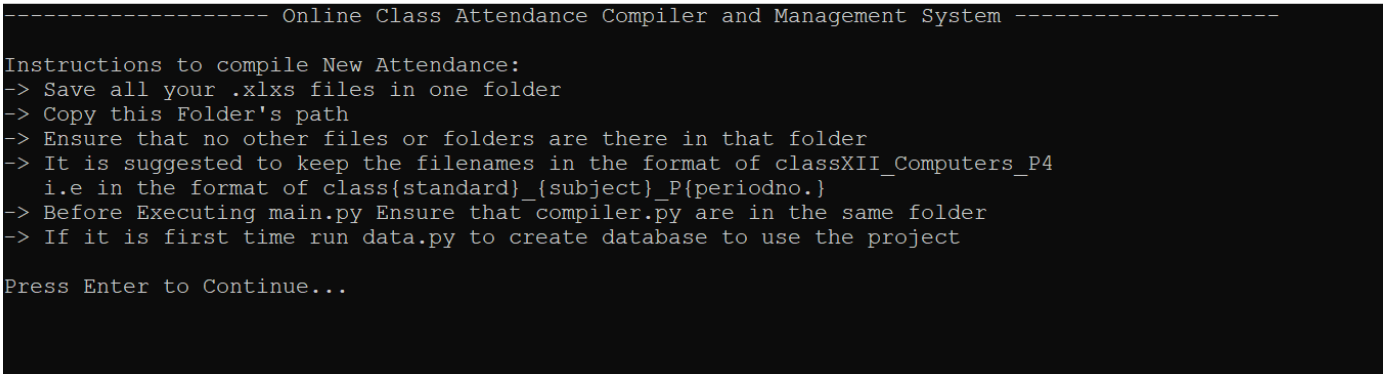
If we opt the 1st choice we see the menu:-



```
-------------------- Online Class Attendance Compiler and Management System --------------------
1: Compile New Attendance.
2: View Previous Attendance.
3: Edit Previous Attendance of a Cadet.
4: Return to Previous Menu
0: Exit
Enter your Choice from above menu:
```

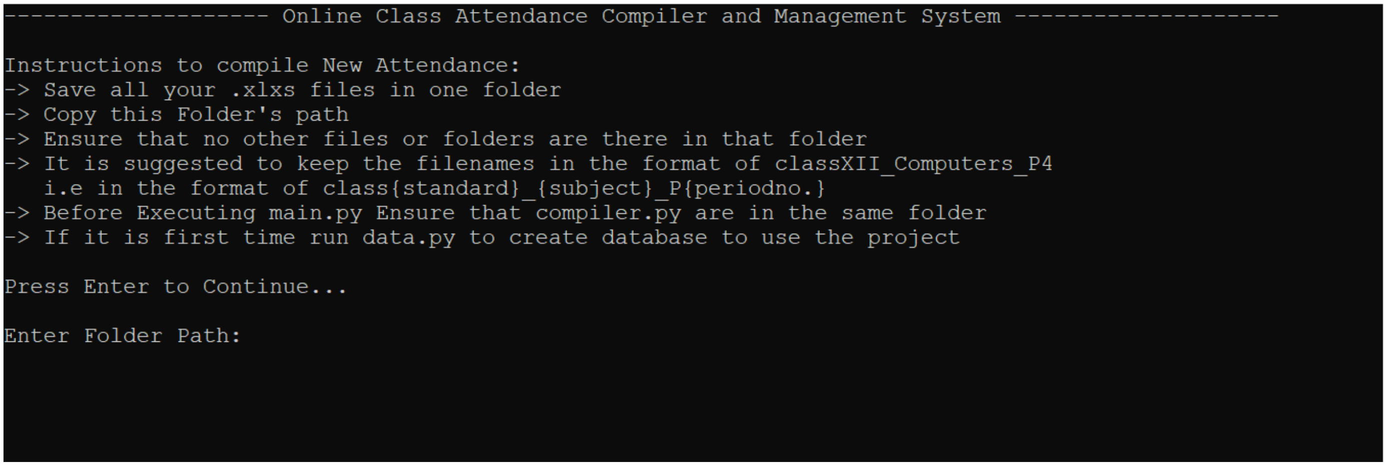
Here we see the options to manage the attendances or to create new ones …

If we opt for creating a new attendance we see Instruction to compile attendance properly without any Error:-



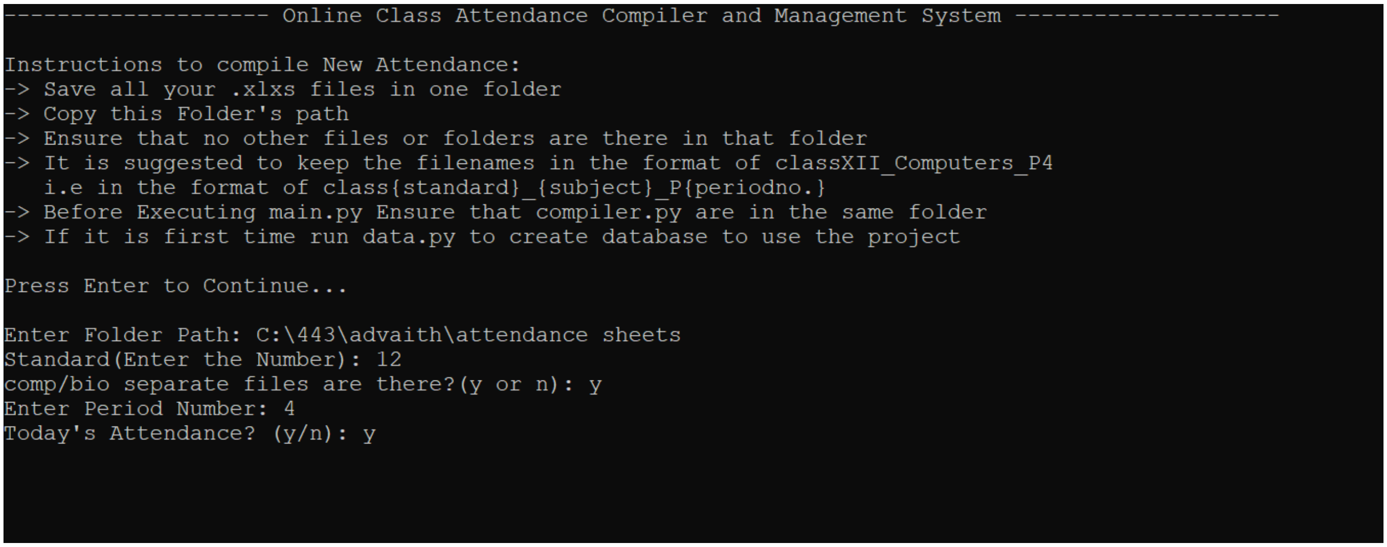
```
-------------------- Online Class Attendance Compiler and Management System --------------------

Instructions to compile New Attendance:
-> Save all your .xlxs files in one folder
-> Copy this Folder's path
-> Ensure that no other files or folders are there in that folder
-> It is suggested to keep the filenames in the format of classXII_Computers_P4
   i.e in the format of class{standard}_{subject}_P{periodno.}
-> Before Executing main.py Ensure that compiler.py are in the same folder
-> If it is first time run data.py to create database to use the project

Press Enter to Continue...
```

After clicking enter and continuing.. we are asked to enter folder path



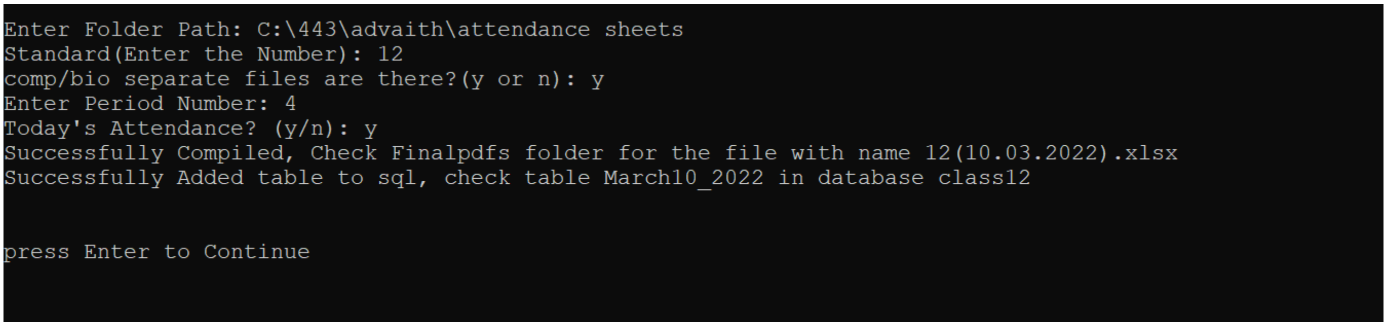
```
-------------------- Online Class Attendance Compiler and Management System --------------------

Instructions to compile New Attendance:
-> Save all your .xlxs files in one folder
-> Copy this Folder's path
-> Ensure that no other files or folders are there in that folder
-> It is suggested to keep the filenames in the format of classXII_Computers_P4
   i.e in the format of class{standard}_{subject}_P{periodno.}
-> Before Executing main.py Ensure that compiler.py are in the same folder
-> If it is first time run data.py to create database to use the project

Press Enter to Continue...

Enter Folder Path:
```

Here we have to enter the path to the folder where the excel sheets are stored and enter the class whose attendance it is , if the standard is 11 / 12 you will be asked to specify whether there are separate attendance .xlsx files in the folder. if yes, enter the period number of the computer/biology period .Then enter the date of the attendance sheets



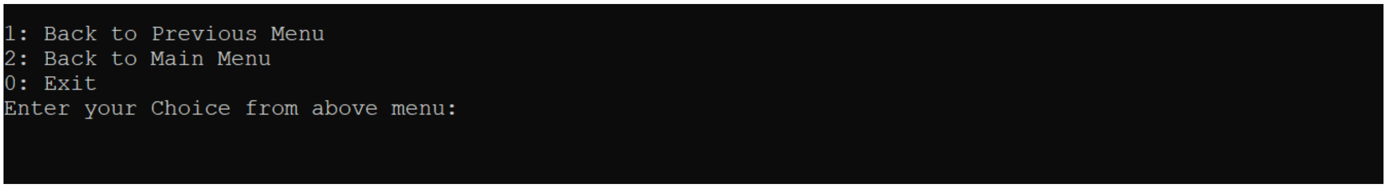
```
-------------------- Online Class Attendance Compiler and Management System --------------------

Instructions to compile New Attendance:
-> Save all your .xlxs files in one folder
-> Copy this Folder's path
-> Ensure that no other files or folders are there in that folder
-> It is suggested to keep the filenames in the format of classXII_Computers_P4
   i.e in the format of class{standard}_{subject}_P{periodno.}
-> Before Executing main.py Ensure that compiler.py are in the same folder
-> If it is first time run data.py to create database to use the project

Press Enter to Continue...

Enter Folder Path: C:\443\advaith\attendance sheets
Standard(Enter the Number): 12
comp/bio separate files are there?(y or n): y
Enter Period Number: 4
Today's Attendance? (y/n): y
```

If compiled successfully we see the message and can find the finalfiles folder where the compiled excel sheet will be present...



```
Enter Folder Path: C:\443\advaith\attendance sheets
Standard(Enter the Number): 12
comp/bio separate files are there?(y or n): y
Enter Period Number: 4
Today's Attendance? (y/n): y
Successfully Compiled, Check Finalpdfs folder for the file with name 12(10.03.2022).xlsx
Successfully Added table to sql, check table March10_2022 in database class12


press Enter to Continue
```

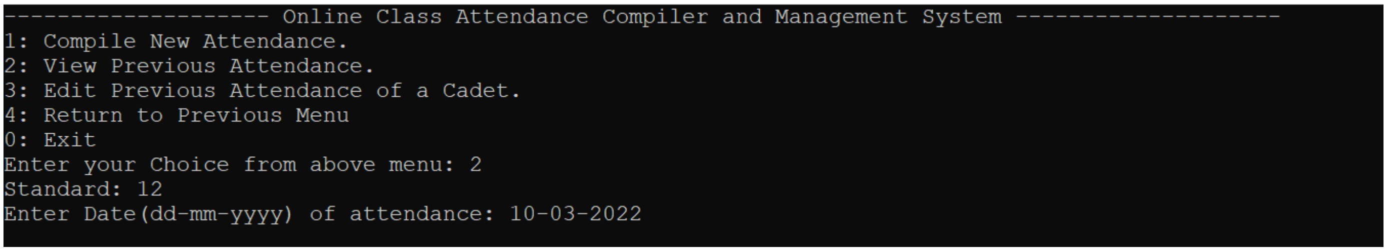
After this you see this menu



```
1: Back to Previous Menu
2: Back to Main Menu
0: Exit
Enter your Choice from above menu:
```

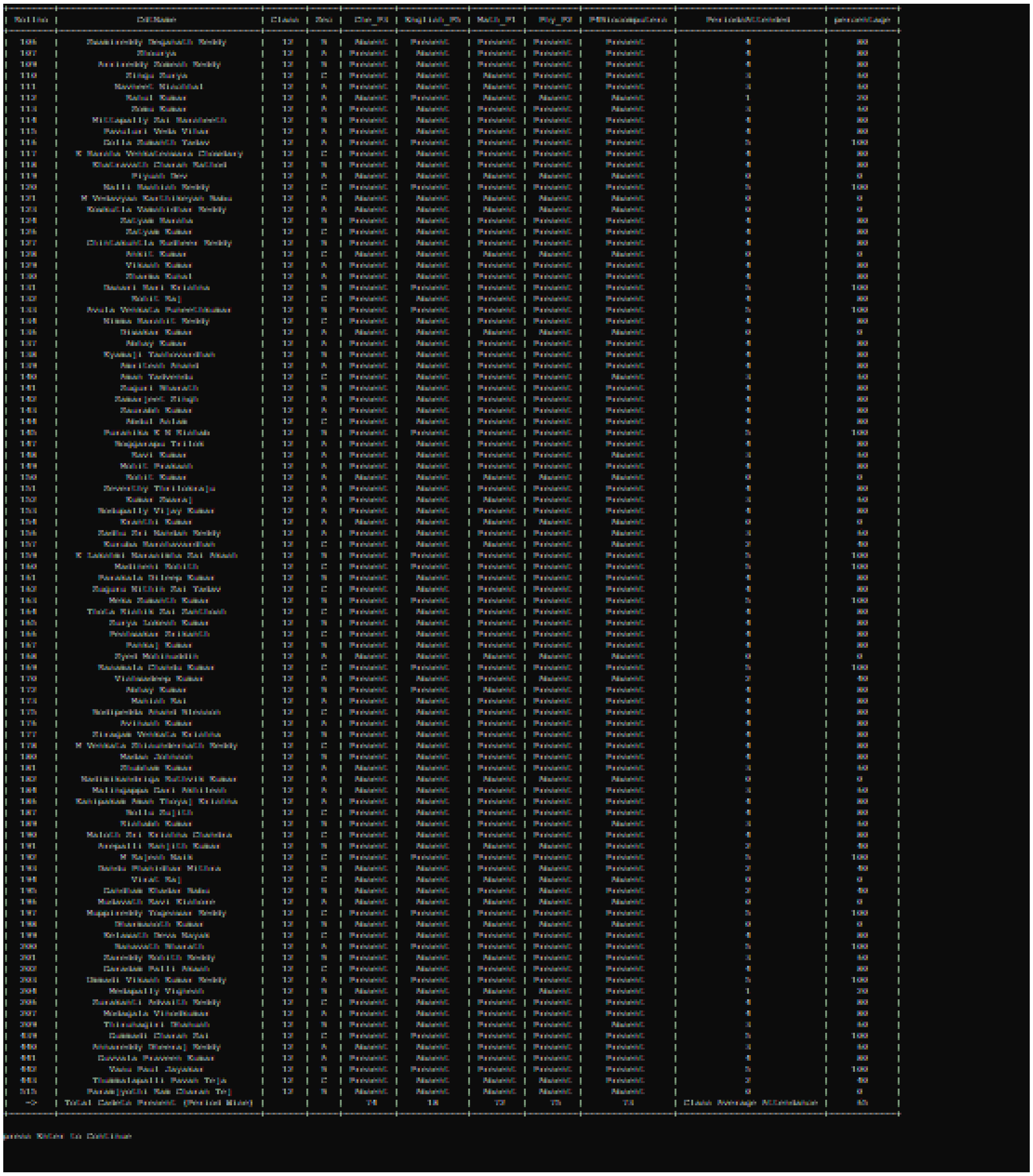
Going back to the pervious menu we again have the menu for online attendance management system now.

To see a previous attendance we enter 2 and fill the asked details .



```
-------------------- Online Class Attendance Compiler and Management System --------------------
1: Compile New Attendance.
2: View Previous Attendance.
3: Edit Previous Attendance of a Cadet.
4: Return to Previous Menu
0: Exit
Enter your Choice from above menu: 2
Standard: 12
Enter Date(dd-mm-yyyy) of attendance: 10-03-2022
```

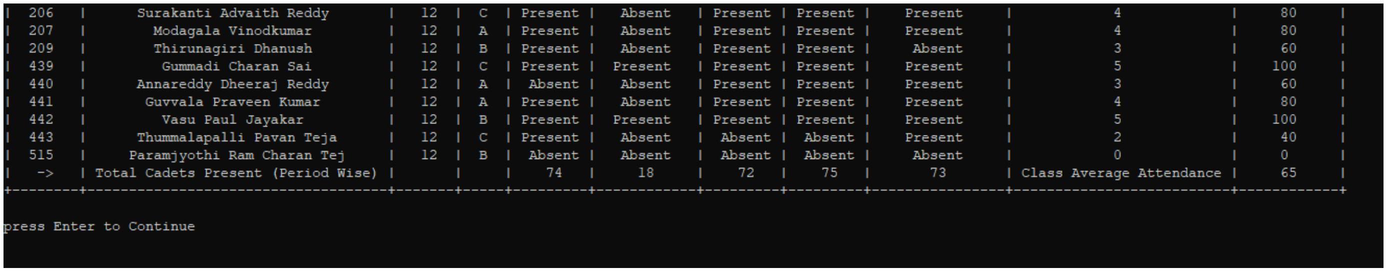
Proceeding we see the previous attendance statement of the class and date if existing in a petty table



```
|--------|------------------------------------------|-------|-----|---------|------------|---------|---------|----------------|-----------------------------|------------|
| RollNo |                 CdtName                  | Class | Sec |  Che_P3 | English_P5 | Math_P1 |  Phy_P2 | PEBiocomputers |       PeriodsAttended       | percentage |
|--------|------------------------------------------|-------|-----|---------|------------|---------|---------|----------------|-----------------------------|------------|
|  106   |        Sammireddy Deepanath Reddy        |  12   |  B  |  Absent |   Present  | Present | Present |     Present    |              4              |     80     |
|  107   |                 Shaurya                  |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  109   |          Anireddy Sumesh Reddy           |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  110   |               Singa Surya                |  12   |  C  | Present |   Absent   |  Absent | Present |     Present    |              3              |     60     |
|  111   |             Navneet Nischal              |  12   |  A  | Present |   Absent   |  Absent | Present |     Present    |              3              |     60     |
|  112   |               Rahul Kumar                |  12   |  A  |  Absent |   Present  |  Absent |  Absent |     Absent     |              1              |     20     |
|  113   |                Sonu Kumar                |  12   |  A  | Present |   Absent   | Present | Present |     Absent     |              3              |     60     |
|  114   |        Mittapally Sai Sandeesh           |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  115   |           Pavaluri Veda Vihar            |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  116   |           Golla Sumanth Yadav            |  12   |  A  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  117   |   K Harsha Venkateswara Chowdary         |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  118   |        Khatravath Charan Rathod          |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  119   |                Piyush Dev                |  12   |  A  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  120   |           Nalli Sashish Reddy            |  12   |  C  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  121   |       M Velavyan Karthikeyan Babu        |  12   |  A  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  123   |        Konkatla Vamshidhar Reddy         |  12   |  A  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  124   |              Satyam Sarika               |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  126   |               Satyam Kumar               |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  127   |       Chintakuntla Sudheer Reddy         |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  128   |               Ankit Kumar                |  12   |  C  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  129   |               Vikash Kumar               |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  130   |               Sharma Rahul               |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  131   |           Dasari Hari Krishna            |  12   |  B  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  132   |                Rohit Raj                 |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  133   |       Avula Venkata Puneethkumar         |  12   |  B  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  134   |           Nimma Harshit Reddy            |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  136   |              Diwakar Kumar               |  12   |  A  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  137   |               Abhay Kumar                |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  138   |          Kyasaji Yashwardhan             |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  139   |             Amritesh Anand               |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  140   |             Aman Yadavendu               |  12   |  C  | Present |   Absent   | Present | Present |     Absent     |              3              |     60     |
|  141   |              Rajuri Bharath              |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  142   |             Ramanjeet Singh              |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  143   |              Saurabh Kumar               |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  144   |               Mohd Aslam                 |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  145   |          Puranika K N Rishab             |  12   |  B  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  147   |           Nagaparapu Tilak               |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  148   |                Ravi Kumar                |  12   |  A  | Present |   Absent   | Present | Present |     Absent     |              3              |     60     |
|  149   |              Mohit Prakash               |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  150   |               Rohit Kumar                |  12   |  A  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  151   |          Revathy Thrilokraju             |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  152   |              Kumar Swaraj                |  12   |  A  | Present |   Absent   |  Absent | Present |     Present    |              3              |     60     |
|  153   |          Bolapally Vijay Kumar           |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  154   |              Kranthi Kumar               |  12   |  A  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  156   |         Sadhu Sai Mahesh Reddy           |  12   |  A  | Present |   Absent   | Present | Present |     Absent     |              3              |     60     |
|  157   |          Kundan Nandhavardhan            |  12   |  C  | Present |   Absent   |  Absent | Present |     Absent     |              2              |     40     |
|  159   |    K Lakshmi Narasimha Sai Akash         |  12   |  B  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  160   |            Nallireddi Rohith             |  12   |  C  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  161   |          Parakala Dileep Kumar           |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  162   |         Raguru Nithin Sai Yadav          |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  163   |           Mitta Sumanth Kumar            |  12   |  B  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  164   |        Thota Rishik Sai Santosh          |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  165   |           Surya Lokesh Kumar             |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  166   |           Prabhakar Srikanth             |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  167   |              Pankaj Kumar                |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  168   |             Syed Muhiuddin               |  12   |  A  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  169   |          Kasamala Chetan Kumar           |  12   |  C  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  170   |            Vishwadeep Kumar              |  12   |  A  | Present |   Absent   | Present |  Absent |     Absent     |              2              |     40     |
|  172   |               Abhay Kumar                |  12   |  B  | Present |   Present  |  Absent | Present |     Present    |              4              |     80     |
|  173   |               Manish Rai                 |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  175   |        Nalipelli Akhil Rithesh           |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  176   |              Avinash Kumar               |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  177   |         Sirugam Venkata Krishna          |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  178   |   M Venkata Shivasainath Reddy           |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  180   |              Mohan Abhinash              |  12   |  B  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  181   |              Shubham Kumar               |  12   |  A  |  Absent |   Absent   | Present | Present |     Present    |              3              |     60     |
|  182   |      Nellimitadiga Rithvik Kumar         |  12   |  A  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  184   |        Mallipappa Sai Akhilesh           |  12   |  A  |  Absent |   Absent   | Present | Present |     Present    |              3              |     60     |
|  186   |   Kanipakam Akash Thayaj Krishna         |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  187   |               Bolla Sujith               |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  188   |              Rishabh Kumar               |  12   |  B  | Present |   Absent   | Present | Present |     Absent     |              3              |     60     |
|  190   |      Malothu Sai Krishna Chandra         |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  191   |         Annapalli Ranjith Kumar          |  12   |  B  | Present |   Absent   |  Absent |  Absent |     Present    |              2              |     40     |
|  192   |               M Rajesh Naik              |  12   |  C  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  193   |          Dalai Phanidhar Mishra          |  12   |  B  |  Absent |   Absent   | Present |  Absent |     Present    |              2              |     40     |
|  194   |                Virat Raj                 |  12   |  C  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  195   |           Gaddam Charan Babu             |  12   |  B  |  Absent |   Absent   | Present |  Absent |     Present    |              2              |     40     |
|  196   |          Mudavath Ravi Kishore           |  12   |  A  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  197   |       Muppireddy Tejeswar Reddy          |  12   |  C  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  198   |            Dharmavath Kumar              |  12   |  B  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|  199   |           Islavath Deva Nayak            |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  200   |            Banavath Bharath              |  12   |  A  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  201   |           Sanreddy Rohith Reddy          |  12   |  B  | Present |   Absent   |  Absent | Present |     Present    |              3              |     60     |
|  202   |           Gandham Palli Akash            |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  203   |        Ummadi Vikash Kumar Reddy         |  12   |  A  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  204   |           Bolapally Vignesh              |  12   |  B  |  Absent |   Absent   |  Absent |  Absent |     Present    |              1              |     20     |
|  205   |         Sarabanti Advaith Reddy          |  12   |  C  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  207   |           Nakapala Vinodkumar            |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  209   |           Thirumagiri Dhanush            |  12   |  B  | Present |   Absent   | Present | Present |     Absent     |              3              |     60     |
|  439   |           Gummadi Charan Sai             |  12   |  C  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  440   |        Akkareddy Dheeraj Reddy           |  12   |  A  |  Absent |   Absent   | Present | Present |     Present    |              3              |     60     |
|  441   |          Guvvala Praveen Kumar           |  12   |  A  | Present |   Absent   | Present | Present |     Present    |              4              |     80     |
|  442   |            Vanu Paul Jayakar             |  12   |  B  | Present |   Present  | Present | Present |     Present    |              5              |    100     |
|  443   |        Thummalapalli Pavan Teja          |  12   |  C  | Present |   Absent   |  Absent |  Absent |     Present    |              2              |     40     |
|  513   |       Paramjyothi Ram Charan Tej         |  12   |  B  |  Absent |   Absent   |  Absent |  Absent |     Absent     |              0              |     0      |
|   ->   |    Total Cadets Present (Period Wise)    |       |     |    74   |     18     |    72   |    75   |       73       |  Class Average Attendance   |     65     |
|--------|------------------------------------------|-------|-----|---------|------------|---------|---------|----------------|-----------------------------|------------|

press Enter to Continue
```

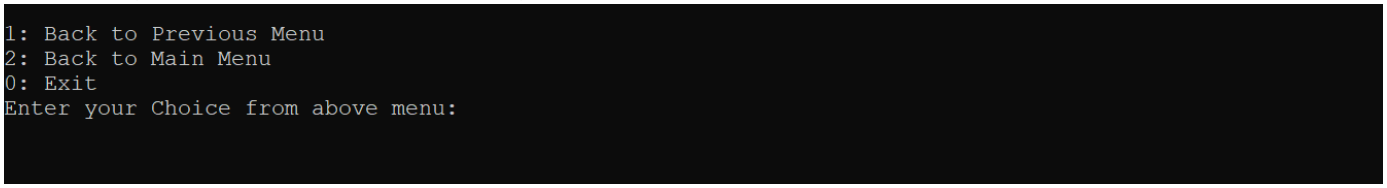
This is a very long table, because of which it is not so clear in the picture

It also gives us the period wise list of total cadets present and class average attendance



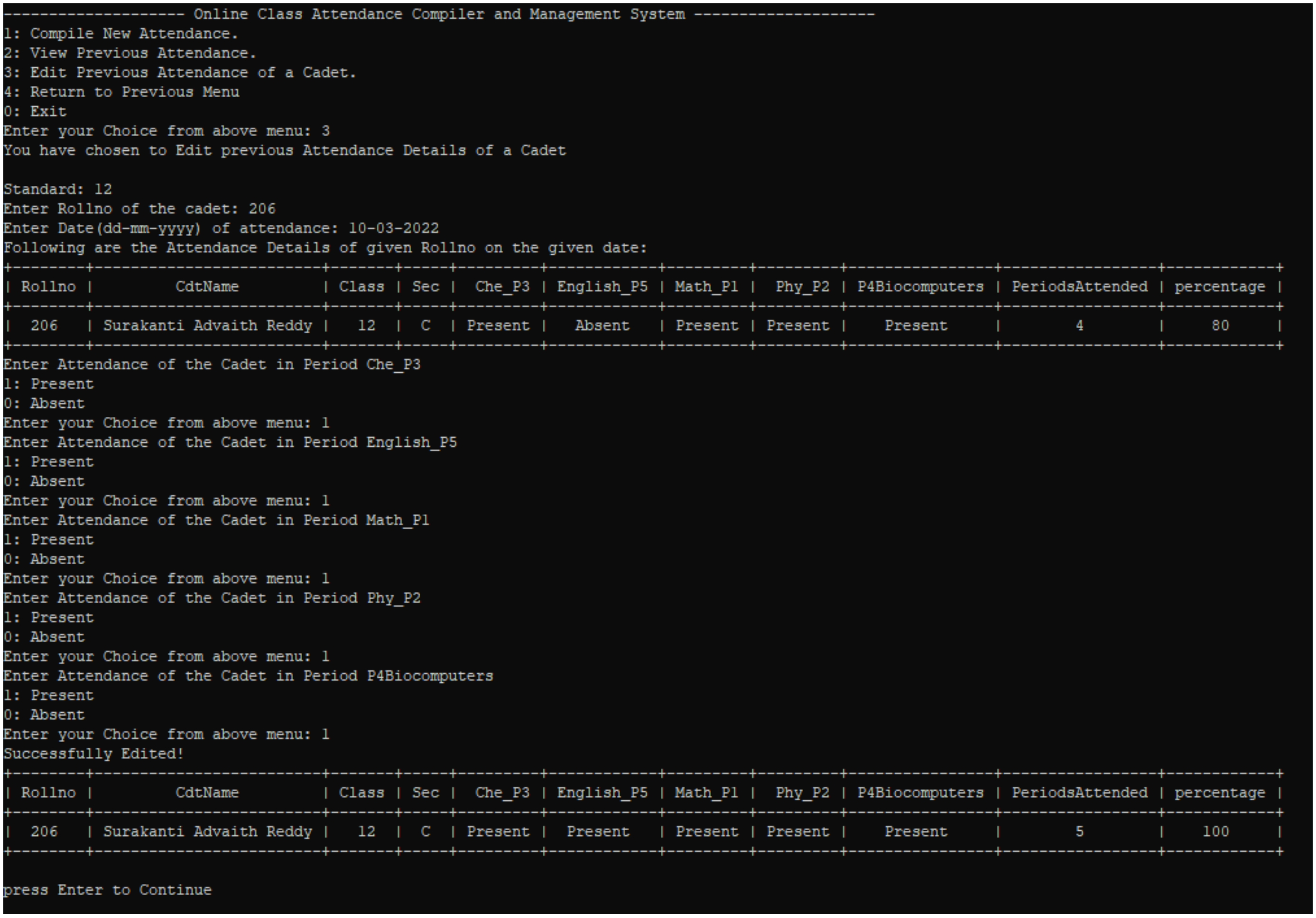
```
|  206   |      Surakanti Advaith Reddy       |   12  |  C  | Present |   Absent    | Present | Present |    Present     |           4              |    80     |
|  207   |        Modagala Vinodkumar         |   12  |  A  | Present |   Absent    | Present | Present |    Present     |           4              |    80     |
|  209   |        Thirunagiri Dhanush         |   12  |  B  | Present |   Absent    | Present | Present |     Absent     |           3              |    60     |
|  439   |         Gummadi Charan Sai         |   12  |  C  | Present |  Present    | Present | Present |    Present     |           5              |   100     |
|  440   |      Annareddy Dheeraj Reddy       |   12  |  A  |  Absent |   Absent    | Present | Present |    Present     |           3              |    60     |
|  441   |       Guvvala Praveen Kumar        |   12  |  A  | Present |   Absent    | Present | Present |    Present     |           4              |    80     |
|  442   |         Vasu Paul Jayakar          |   12  |  B  | Present |  Present    | Present | Present |    Present     |           5              |   100     |
|  443   |      Thummalapalli Pavan Teja      |   12  |  C  | Present |   Absent    |  Absent |  Absent |    Present     |           2              |    40     |
|  515   |     Paramjyothi Ram Charan Tej     |   12  |  B  |  Absent |   Absent    |  Absent |  Absent |     Absent     |           0              |    0      |
|   ->   | Total Cadets Present (Period Wise) |       |     |   74    |     18      |   72    |   75    |      73        | Class Average Attendance |    65     |
+--------+------------------------------------+-------+-----+---------+-------------+---------+---------+----------------+--------------------------+-----------+


press Enter to Continue
```

Moving forward we are again faced with the menu



```
1: Back to Previous Menu
2: Back to Main Menu
0: Exit
Enter your Choice from above menu:
```

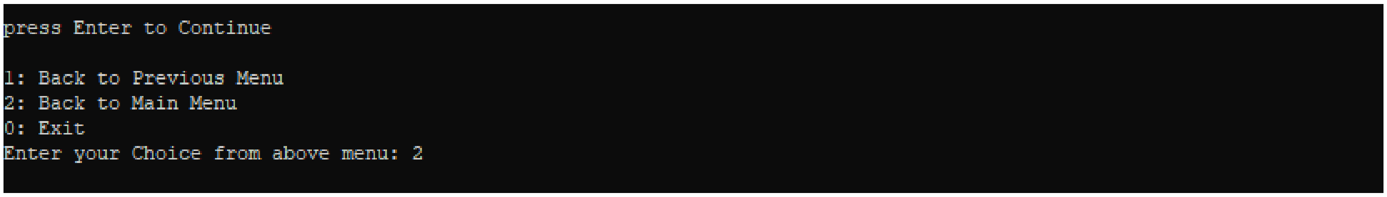
Now if we try to edit the attendance of a cadet we see



```
-------------------- Online Class Attendance Compiler and Management System --------------------
1: Compile New Attendance.
2: View Previous Attendance.
3: Edit Previous Attendance of a Cadet.
4: Return to Previous Menu
0: Exit
Enter your Choice from above menu: 3
You have chosen to Edit previous Attendance Details of a Cadet

Standard: 12
Enter Rollno of the cadet: 206
Enter Date(dd-mm-yyyy) of attendance: 10-03-2022
Following are the Attendance Details of given Rollno on the given date:
+--------+-------------------------+-------+-----+---------+------------+---------+---------+----------------+-----------------+------------+
| Rollno |         CdtName         | Class | Sec |  Che_P3 | English_P5 | Math_P1 |  Phy_P2 | P4Biocomputers | PeriodsAttended | percentage |
+--------+-------------------------+-------+-----+---------+------------+---------+---------+----------------+-----------------+------------+
|  206   | Surakanti Advaith Reddy |   12  |  C  | Present |   Absent   | Present | Present |    Present     |        4        |     80     |
+--------+-------------------------+-------+-----+---------+------------+---------+---------+----------------+-----------------+------------+
Enter Attendance of the Cadet in Period Che_P3
1: Present
0: Absent
Enter your Choice from above menu: 1
Enter Attendance of the Cadet in Period English_P5
1: Present
0: Absent
Enter your Choice from above menu: 1
Enter Attendance of the Cadet in Period Math_P1
1: Present
0: Absent
Enter your Choice from above menu: 1
Enter Attendance of the Cadet in Period Phy_P2
1: Present
0: Absent
Enter your Choice from above menu: 1
Enter Attendance of the Cadet in Period P4Biocomputers
1: Present
0: Absent
Enter your Choice from above menu: 1
Successfully Edited!
+--------+-------------------------+-------+-----+---------+------------+---------+---------+----------------+-----------------+------------+
| Rollno |         CdtName         | Class | Sec |  Che_P3 | English_P5 | Math_P1 |  Phy_P2 | P4Biocomputers | PeriodsAttended | percentage |
+--------+-------------------------+-------+-----+---------+------------+---------+---------+----------------+-----------------+------------+
|  206   | Surakanti Advaith Reddy |   12  |  C  | Present |  Present   | Present | Present |    Present     |        5        |    100     |
+--------+-------------------------+-------+-----+---------+------------+---------+---------+----------------+-----------------+------------+

press Enter to Continue
```

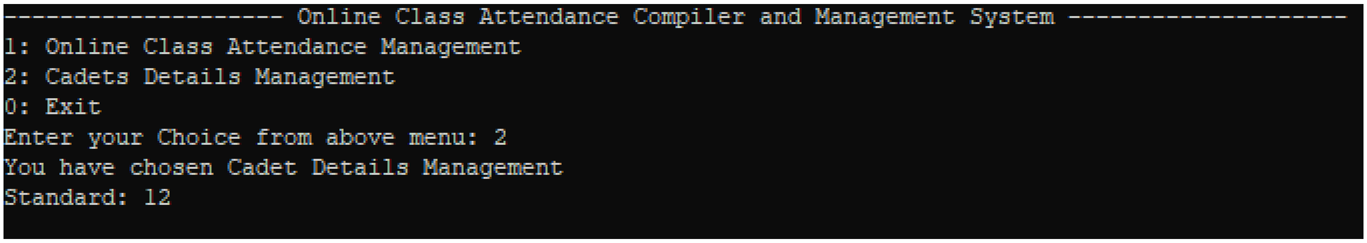
Thus we can edit the attendance of a cadet …

Moving on, now that we have seen how the online attendance compilation and management system works we can now move on to the 2nd option on the main menu i.e. the Cadets Details Management part of the program



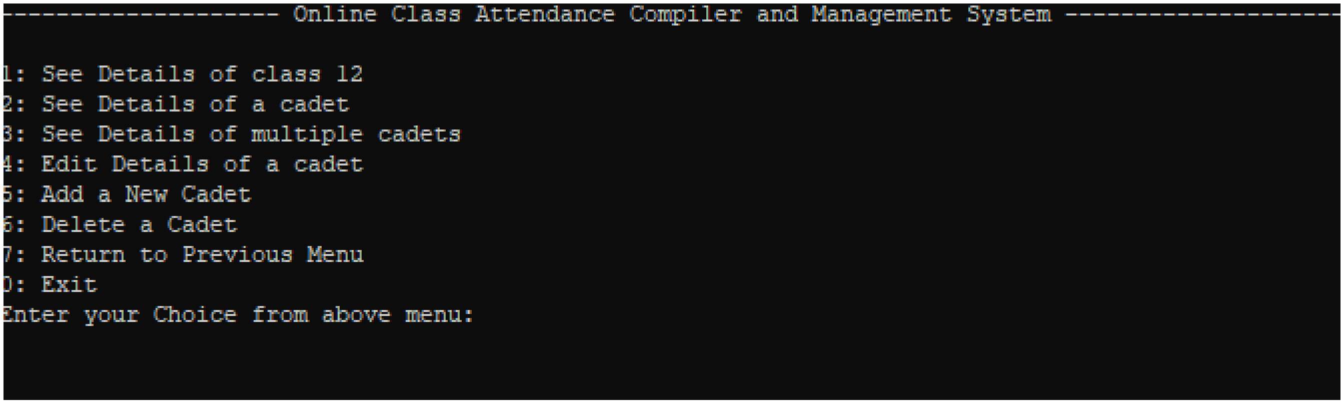
```
press Enter to Continue

1: Back to Previous Menu
2: Back to Main Menu
0: Exit
Enter your Choice from above menu: 2
```

Before entering into the 2nd menu of the program we are asked the standard of the cadet's details that we want to manage



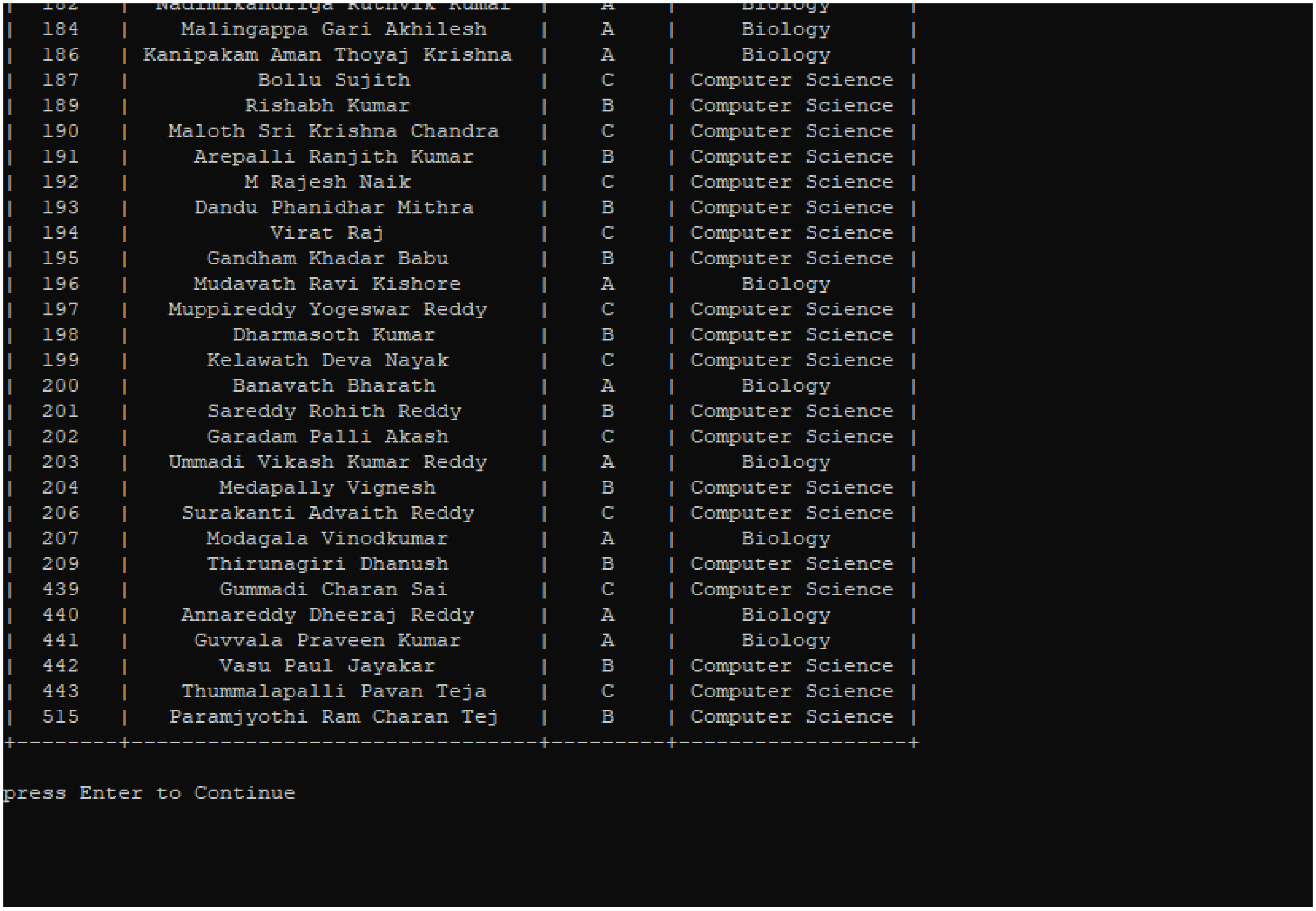
```
-------------------- Online Class Attendance Compiler and Management System --------------------
1: Online Class Attendance Management
2: Cadets Details Management
0: Exit
Enter your Choice from above menu: 2
You have chosen Cadet Details Management
Standard: 12
```

We being a tad bit biased towards class 12 have entered our class and find ourselves facing another menu to deal with



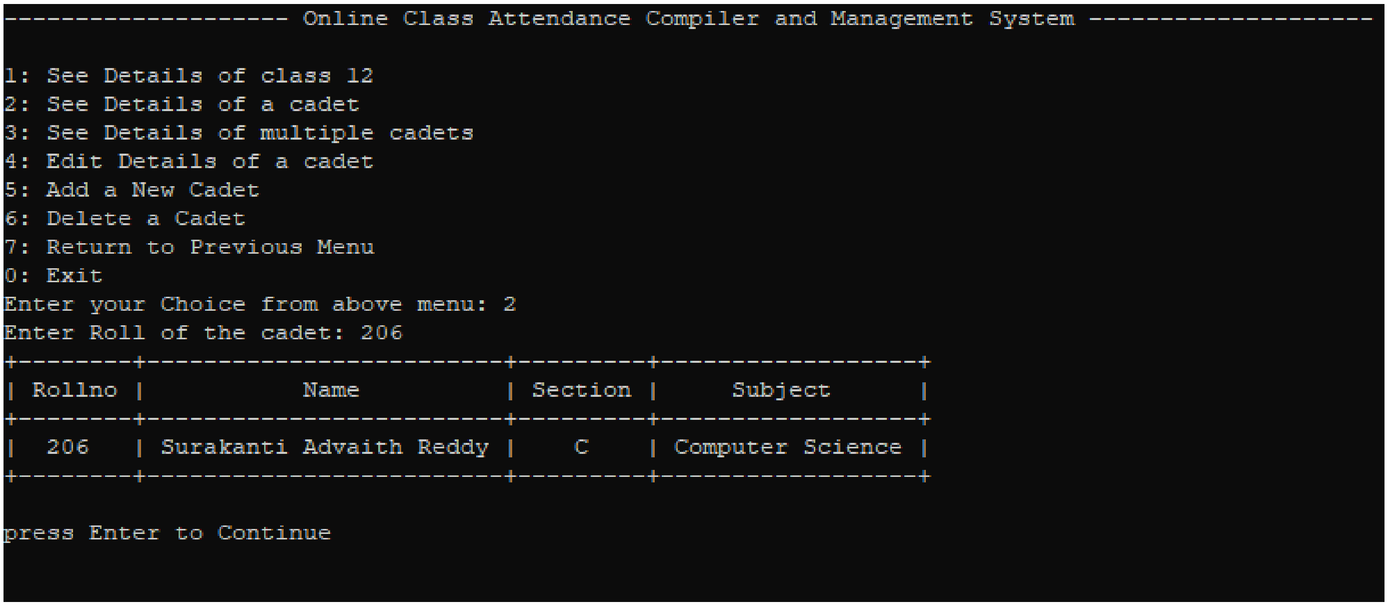
```
-------------------- Online Class Attendance Compiler and Management System --------------------

1: See Details of class 12
2: See Details of a cadet
3: See Details of multiple cadets
4: Edit Details of a cadet
5: Add a New Cadet
6: Delete a Cadet
7: Return to Previous Menu
0: Exit
Enter your Choice from above menu:
```

Choosing to see the details of the cadets we see the roll no, name , section and the optional subject that they chose in class 11…



```
|  182   |  Nadimikandriga Ruthvik Kumar   |    A    |      Biology       |
|  184   |    Malingappa Gari Akhilesh     |    A    |      Biology       |
|  186   | Kanipakam Aman Thoyaj Krishna   |    A    |      Biology       |
|  187   |          Bollu Sujith           |    C    | Computer Science |
|  189   |          Rishabh Kumar          |    B    | Computer Science |
|  190   |   Maloth Sri Krishna Chandra    |    C    | Computer Science |
|  191   |     Arepalli Ranjith Kumar      |    B    | Computer Science |
|  192   |         M Rajesh Naik           |    C    | Computer Science |
|  193   |     Dandu Phanidhar Mithra      |    B    | Computer Science |
|  194   |           Virat Raj             |    C    | Computer Science |
|  195   |      Gandham Khadar Babu        |    B    | Computer Science |
|  196   |     Mudavath Ravi Kishore       |    A    |      Biology       |
|  197   |   Muppireddy Yogeswar Reddy     |    C    | Computer Science |
|  198   |       Dharmasoth Kumar          |    B    | Computer Science |
|  199   |      Kelawath Deva Nayak        |    C    | Computer Science |
|  200   |       Banavath Bharath          |    A    |      Biology       |
|  201   |      Sareddy Rohith Reddy       |    B    | Computer Science |
|  202   |      Garadam Palli Akash        |    C    | Computer Science |
|  203   |   Ummadi Vikash Kumar Reddy     |    A    |      Biology       |
|  204   |       Medapally Vignesh         |    B    | Computer Science |
|  206   |    Surakanti Advaith Reddy      |    C    | Computer Science |
|  207   |      Modagala Vinodkumar        |    A    |      Biology       |
|  209   |      Thirunagiri Dhanush        |    B    | Computer Science |
|  439   |      Gummadi Charan Sai         |    C    | Computer Science |
|  440   |    Annareddy Dheeraj Reddy      |    A    |      Biology       |
|  441   |     Guvvala Praveen Kumar       |    A    |      Biology       |
|  442   |       Vasu Paul Jayakar         |    B    | Computer Science |
|  443   |    Thummalapalli Pavan Teja     |    C    | Computer Science |
|  515   |   Paramjyothi Ram Charan Tej    |    B    | Computer Science |
+--------+---------------------------------+---------+------------------+

press Enter to Continue
```

Choosing to view a particular cadet's details shows us the following



```
-------------------- Online Class Attendance Compiler and Management System --------------------

1: See Details of class 12
2: See Details of a cadet
3: See Details of multiple cadets
4: Edit Details of a cadet
5: Add a New Cadet
6: Delete a Cadet
7: Return to Previous Menu
0: Exit
Enter your Choice from above menu: 2
Enter Roll of the cadet: 206
+--------+-------------------------+---------+------------------+
| Rollno |           Name          | Section |     Subject      |
+--------+-------------------------+---------+------------------+
|  206   | Surakanti Advaith Reddy |    C    | Computer Science |
+--------+-------------------------+---------+------------------+

press Enter to Continue
```

And opting to see the details of a few select cadets gives us the option to select exactly which cadet's details we want to see

LIMITATIONS

1. This Whole Program is Usable only if the online class attendance is recorded using the Extension: google meet attendance collector.

2. Extension is only applicable for recording attendance of google meets making the program to compile only google meet classes’ attendance.

BIBLIOGRAPHY

1. Computer Science with Python by Sumita Arora.

2. Python Crash Course by Eric Matthes

3. www.google.com – referred about openpyxl, files manipulation, OS module and related Topics.