**Student Examination Portal**

**Submitted by**

**Name of the Student:** Arghyadip Chowdhury

**Enrolment Number:** 12022002002102

**Section:** A

**Class Roll Number:** 23  
**Stream:** CSE

**Subject:** Programming for Problem Solving using Python

**Subject Code:** IVC101

**Department:** Basic Science and Humanities

Under the supervision of

<Name of the Teachers>

**Academic Year: 2022-26**

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE FIRST SEMESTER



**DEPARTMENT OF BASIC SCIENCE AND HUMANITITES**

**INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA**



**CERTIFICATE OF RECOMMENDATION**

We hereby recommend that the project prepared under our supervision by **Arghyadip Chowdhury,** entitled **Student Examination Portal** be accepted in partial fulfillment of the requirements for the degree of partial fulfillment of the first semester.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head of the Department Project Supervisor

Basic Sciences and Humanities

IEM, Kolkata

# Introduction

In this project we create various modules using Python to create databases (as CSV files) in which we can store information about students , courses , batches, departments and marks obtained by students in a particular examination. We can also display pie charts , histograms etc. on the basis of the data stored in the csv files

## Objective

To create various Python modules for a **Student Examination Portal**

## Organization of the Project

We create 5 modules by the name of STUDENT , COURSE , BATCH , DEPARTMENT and EXAMINATION using Python IDLE(3.11.0). Each of these modules

have various functions defined which they can perform.

The Functions each module can perform are as follows -

***STUDENT.py*** - Create a new Student , Update Student Details , Delete a Student from the database , Generate Report Card of a Student with Grade.

***COURSE.py*** - Create a new Course ,View performance of all students in a course, Histogram showing course statistics.

***BATCH.py*** - Create a new Batch , View list of all students in the batch , View list of all courses in the batch , Pie Chart of % of all students

***DEPARTMENT.py*** - Create a new Department , View all batches in department , Line plot showing Department statistics.

***EXAMINATION.py*** - Enter marks of students for an examination , View performance of all students in the exam ,Scatter plot of marks obtained by students.

# Database Descriptions

There are 5 databases used in this project , they are stored as CSV files, they are as follows:

***STUDENT.csv-***Stores data related to the students like student id , name , roll , batch etc.

***COURSE.csv-***Stores data related to the courses like Course id , Details of students enrolled etc.

***BATCH.csv-***Stores data related to the students like student id , name , roll , batch etc.

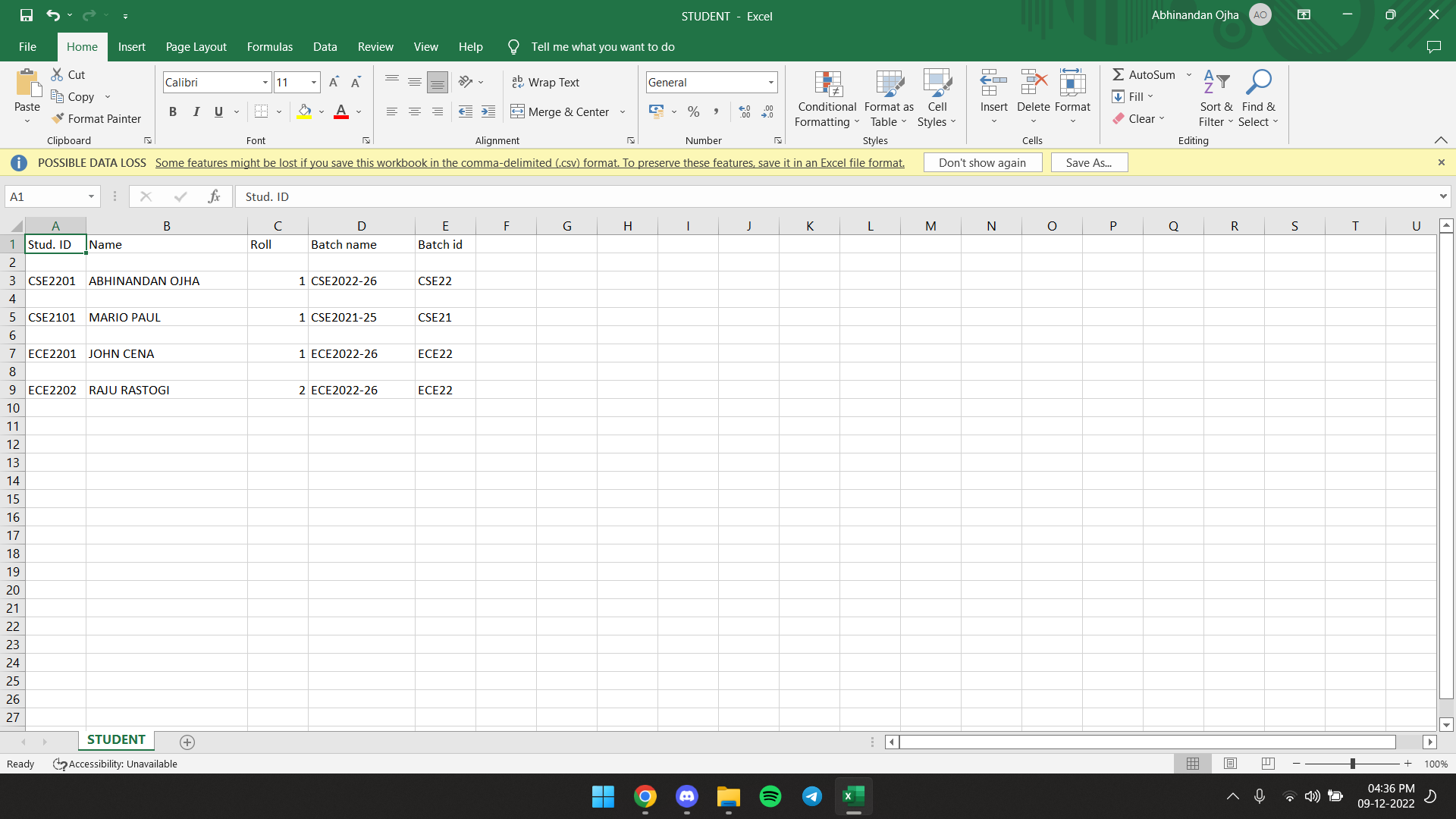
***DEPARTMENT.csv-***Stores data related to the batches like Batch id , Department etc.

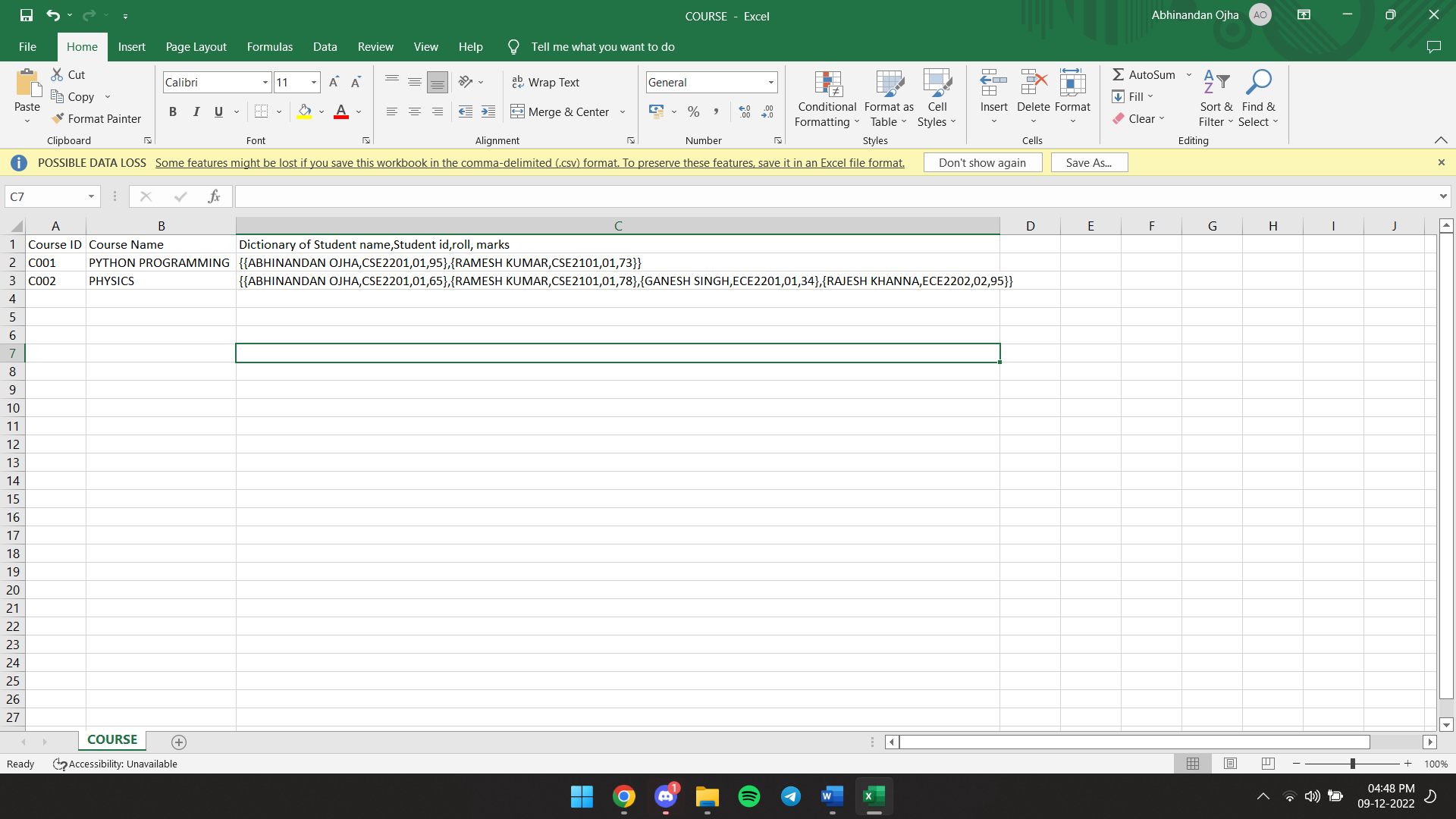
***EXAMINATION.csv-***Stores data related to the examinations like course, students and marks obtained.

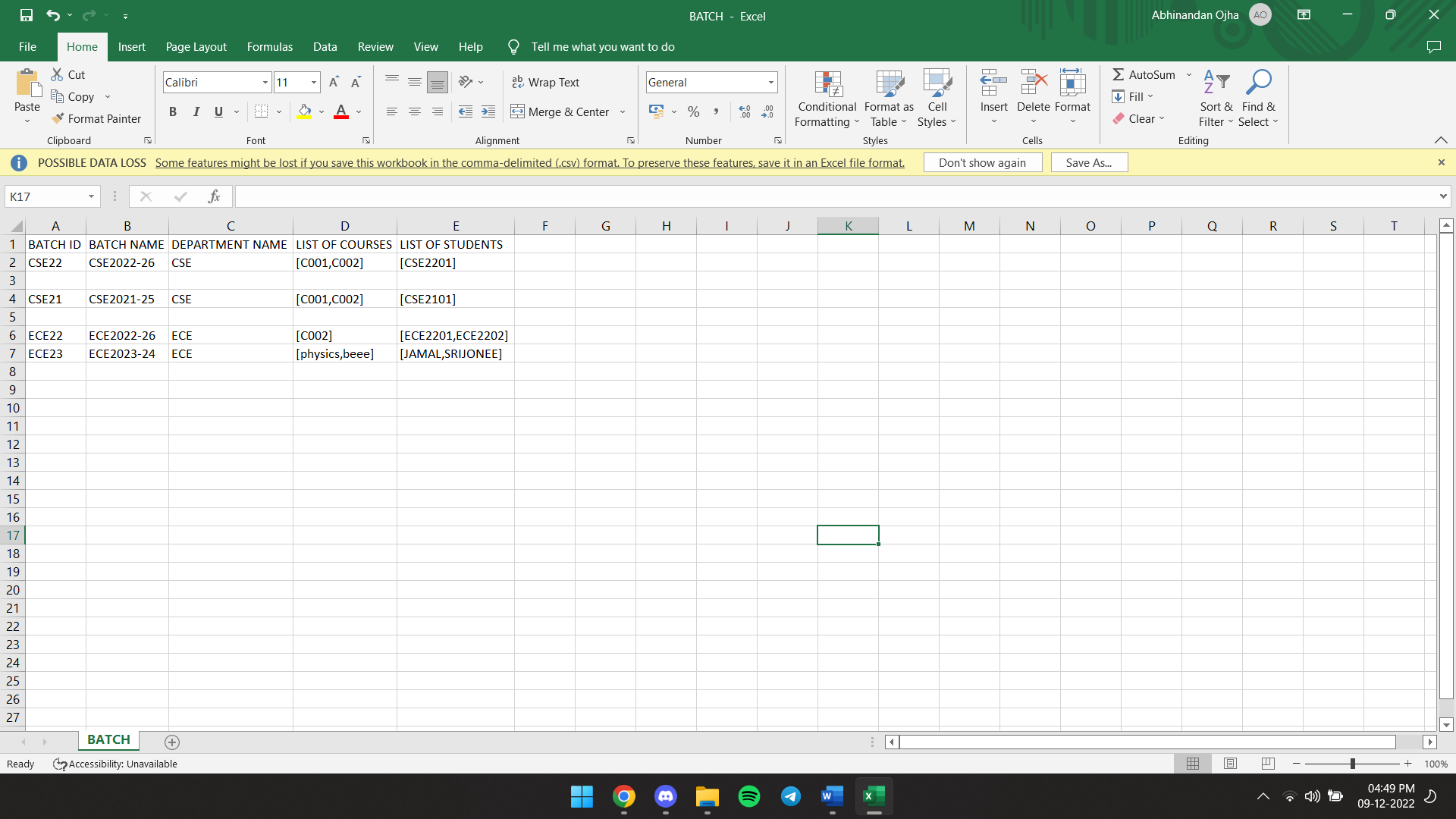
## Database Samples

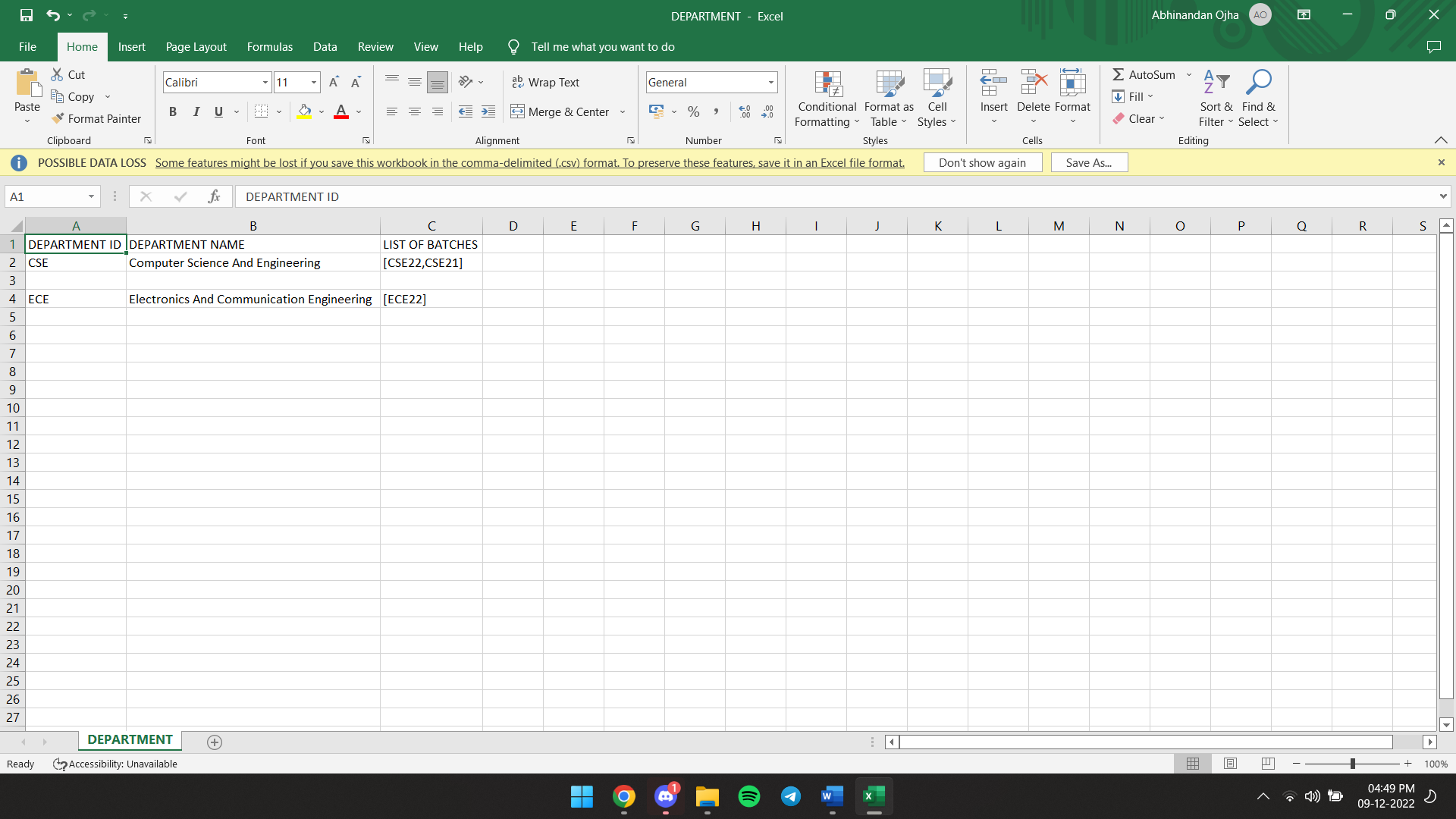
Screenshots of the sample databases created are provided below-

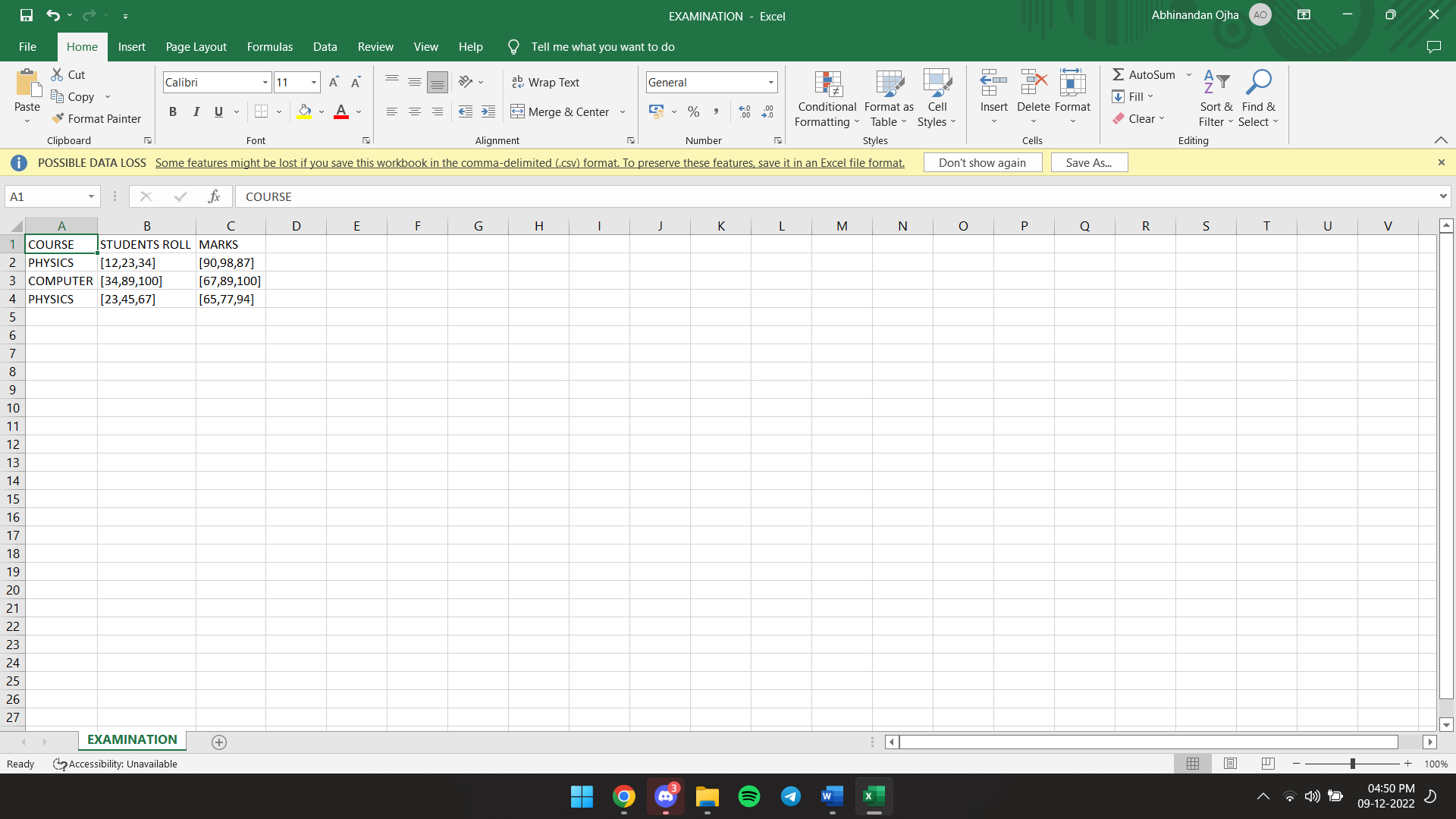
***STUDENT.csv-***

******

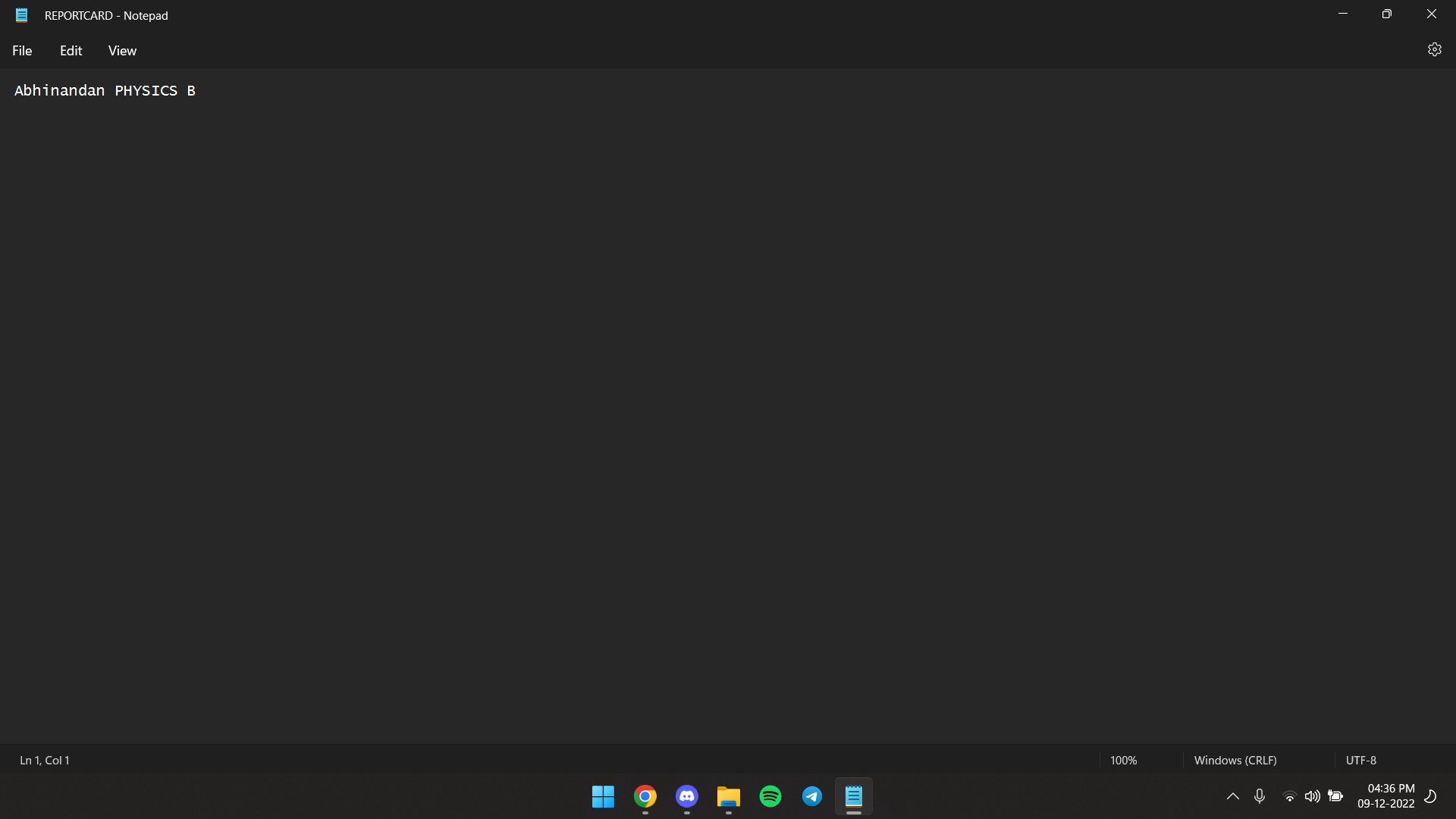
***COURSE.csv-***

***BATCH.csv-***

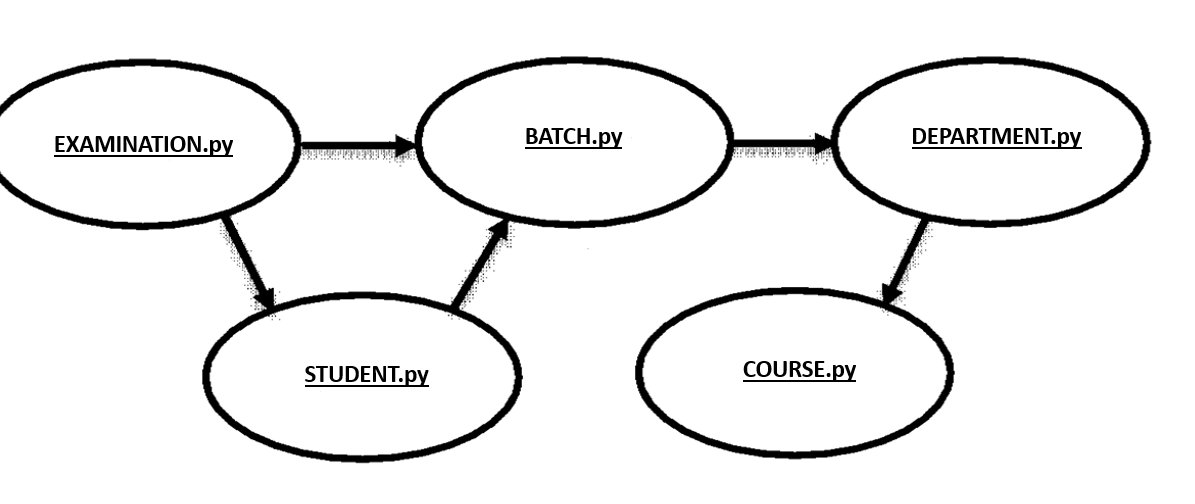
***DEPARTMENT.csv-***

***EXAMINATION.csv-***

***SAMPLE REPORT CARD CREATED USING STUDENT MODULE(.txt file - REPORTCARD.txt) -***

******

# Data Flow and E-R Diagram:

The dependance on the created python modules for the marks of students is shown in the diagram -

# Programs

The Python programs for the various modules are provided below -

1. **STUDENT.py:**

import csv

student\_fields = ['Student ID', 'Name', 'Class Roll Number', 'Batch Name', 'Batch ID']

student\_database = 'STUDENT.csv'

def add\_student():

print("-------------------------")

print("Add Student Information")

print("-------------------------")

global student\_fields

global student\_database

student\_data = []

for field in student\_fields:

value = input("Enter " + field + ": ")

student\_data.append(value)

with open(student\_database, "a", encoding="utf-8") as f:

writer = csv.writer(f)

writer.writerows([student\_data])

print("Data saved successfully")

input("Press any key to continue")

return

def update\_student():

global student\_fields

global student\_database

print("--- Update Student ---")

roll = input("Enter roll no. to update: ")

index\_student = None

updated\_data = []

with open(student\_database, "r", encoding="utf-8") as f:

reader = csv.reader(f)

counter = 0

for row in reader:

if len(row) > 0:

if roll == row[2]:

index\_student = counter

print("Student Found: at index ",index\_student)

student\_data = []

for field in student\_fields:

value = input("Enter " + field + ": ")

student\_data.append(value)

updated\_data.append(student\_data)

else:

updated\_data.append(row)

counter += 1

if index\_student is not None:

with open(student\_database, "w", encoding="utf-8") as f:

writer = csv.writer(f)

writer.writerows(updated\_data)

else:

print("Roll No. not found in our database")

input("Press any key to continue")

def delete\_student():

global student\_fields

global student\_database

print("--- Remove Student ---")

roll = input("Enter roll no. to remove: ")

student\_found = False

updated\_data = []

with open(student\_database, "r", encoding="utf-8") as f:

reader = csv.reader(f)

counter = 0

for row in reader:

if len(row) > 0:

if roll != row[2]:

updated\_data.append(row)

counter += 1

else:

student\_found = True

if student\_found is True:

with open(student\_database, "w", encoding="utf-8") as f:

writer = csv.writer(f)

writer.writerows(updated\_data)

print("Roll no. ", roll, "deleted successfully")

else:

print("Roll No. not found in our database")

input("Press any key to continue")

def report():

a= input("Student Name:")

b=input ("enter subject name:")

c=int(input("enter marks"))

if c>90:

x='A'

elif c>80:

x='B'

elif c>70:

x='C'

elif c>60:

x='D'

elif c>50:

x='E'

elif c<40:

x='FAIL(F)'

z = open("REPORTCARD.txt","w")

l=[a, b ,x]

z.writelines(l)

print("STUDENT MODULE FUNCTIONS")

print("1.Create A Student")

print("2.Update Student Details")

print("3.Remove a student's details from database")

print("4.Generate Student Report card")

z=int(input("Enter your choice"))

if z==1:

add\_student()

elif z==2:

update\_student()

elif z==3:

delete\_student()

elif z==4:

report()

else:

print("Invalid choice")

1. **COURSE.py:**

import csv

course\_fields = ['Course ID', 'Course Name',' Student Name,Student ID,Roll,Score as a Dictionary(multiple student records can be stored as a sub dictionary(ex:{{},{}})']

course\_database = 'COURSE.csv'

def add\_course():

print("-------------------------")

print("Create New Course")

print("-------------------------")

global course\_fields

global course\_database

course\_data = []

for field in course\_fields:

value = input("Enter " + field + ": ")

course\_data.append(value)

with open(course\_database, "a", encoding="utf-8") as f:

writer=csv.writer(f)

writer.writerows([course\_data])

print("New Course saved successfully")

input("Press any key to continue")

return

def display():

global course\_database

with open(course\_database, "r", encoding="utf-8") as f:

reader = csv.reader(f)

for row in reader:

if len(row)>0:

print (row[2])

def histogram():

from matplotlib import pyplot as plt

import numpy as np

x=np.array([A,B,C,D,E,F])

y=np.array([1,1,1,0,0,1])

axs.hist(x,y)

plt.show()

print("COURSE MODULE FUNCTIONS")

print("1.Create New Course")

print("2.Performance of students in Course")

print("3.Course statistics as Histogram")

z=int(input("Enter your choice"))

if z==1:

add\_course()

elif z==2:

display()

elif z==3:

histogram()

else:

print("Invalid choice")

**3) BATCH.py:**

import csv

batch\_fields = ['BATCH ID', 'BATCH NAME', 'DEPARTMENT NAME', 'LIST OF COURSES', 'LIST OF STUDENTS']

batch\_database = 'BATCH.csv'

def add\_batch():

print("-------------------------")

print("Add Batch Information")

print("-------------------------")

global batch\_fields

global batch\_database

batch\_data = []

for field in batch\_fields:

value = input("Enter " + field + ": ")

batch\_data.append(value)

with open(batch\_database, "a", encoding="utf-8") as f:

writer = csv.writer(f)

writer.writerows([batch\_data])

print("Data saved successfully")

input("Press any key to continue")

return

def display\_students():

global batch\_database

with open(batch\_database, "r", encoding="utf-8") as f:

reader = csv.reader(f)

for row in reader:

if len(row)>0:

print (row[4])

def display\_course():

global batch\_database

with open(batch\_database, "r", encoding="utf-8") as f:

reader = csv.reader(f)

for row in reader:

if len(row)>0:

print (row[3])

def pie():

from matplotlib import pyplot as plt

import numpy as np

student = ['ABHINANDAN', 'RAMESH', 'GANESH', 'RAJESH']

percent= [89,99,32,78]

fig = plt.figure(figsize =(10, 7))

plt.pie(percent, labels = student)

plt.show()

print("BATCH MODULE FUNCTIONS")

print("1.Create New Batch")

print("2.List of students in batch")

print("3.List of courses in batch")

print("4.Batch Statistics in pie plot")

z=int(input("Enter your choice"))

if z==1:

add\_batch()

elif z==2:

display\_students()

elif z==3:

display\_course()

elif z==4:

pie()

else:

print("Invalid choice")

**4) DEPARTMENT.py:**

import csv

dep\_fields = ['DEPATMENT ID', 'DEPATMENT NAME', 'LIST OF BATCHES']

dep\_database = 'DEPARTMENT.csv'

def add\_department():

print("-------------------------")

print("Add Depatment Information")

print("-------------------------")

global dep\_fields

global dep\_database

dep\_data = []

for field in dep\_fields:

value = input("Enter " + field + ": ")

dep\_data.append(value)

with open(dep\_database, "a", encoding="utf-8") as f:

writer = csv.writer(f)

writer.writerows([dep\_data])

print("Data saved successfully")

input("Press any key to continue")

return

def display\_batches():

global dep\_database

with open(dep\_database, "r", encoding="utf-8") as f:

reader = csv.reader(f)

for row in reader:

if len(row)>0:

print (row[2])

print("DEPARTMENT MODULE FUNCTIONS")

print("1.Create New Department")

print("2.List of Batches")

z=int(input("Enter your choice"))

if z==1:

add\_department()

elif z==2:

display\_batches()

else:

print("Invalid choice")

**5) EXAMINATION.py:**

import csv

def entermarks():

a=input("Enter Course Name of Examination")

b=input("Enter all the Student's Roll number in list form" )

c=input("Enter respective marks of the students out of 100 in list form")

print("Data saved")

x=[a,b,c]

with open("EXAMINATION.csv" ,"a") as e:

writer=csv.writer(e)

writer.writerow(x)

def performance():

with open("EXAMINATION.csv" ,"r") as e:

r=csv.reader(e)

for row in r:

print(row)

def plot():

import matplotlib.pyplot as plt

import numpy as np

x1 = np.array([12,23,34])

x2 = np.array([34,89,100])

y1 = np.array([90,98,87])

y2 = np.array([67,89,100])

plt.scatter(x1, y1, color='green')

plt.scatter(x2, y2, color='red')

plt.show()

print("EXAMINATION MODULE FUNCTIONS")

print("1.Enter Marks of Students")

print("2.View Performance of Students")

print("3.Exam Statistics in scatter plot")

z=int(input("Enter your choice"))

if z==1:

entermarks()

elif z==2:

performance()

elif z==3:

plot()

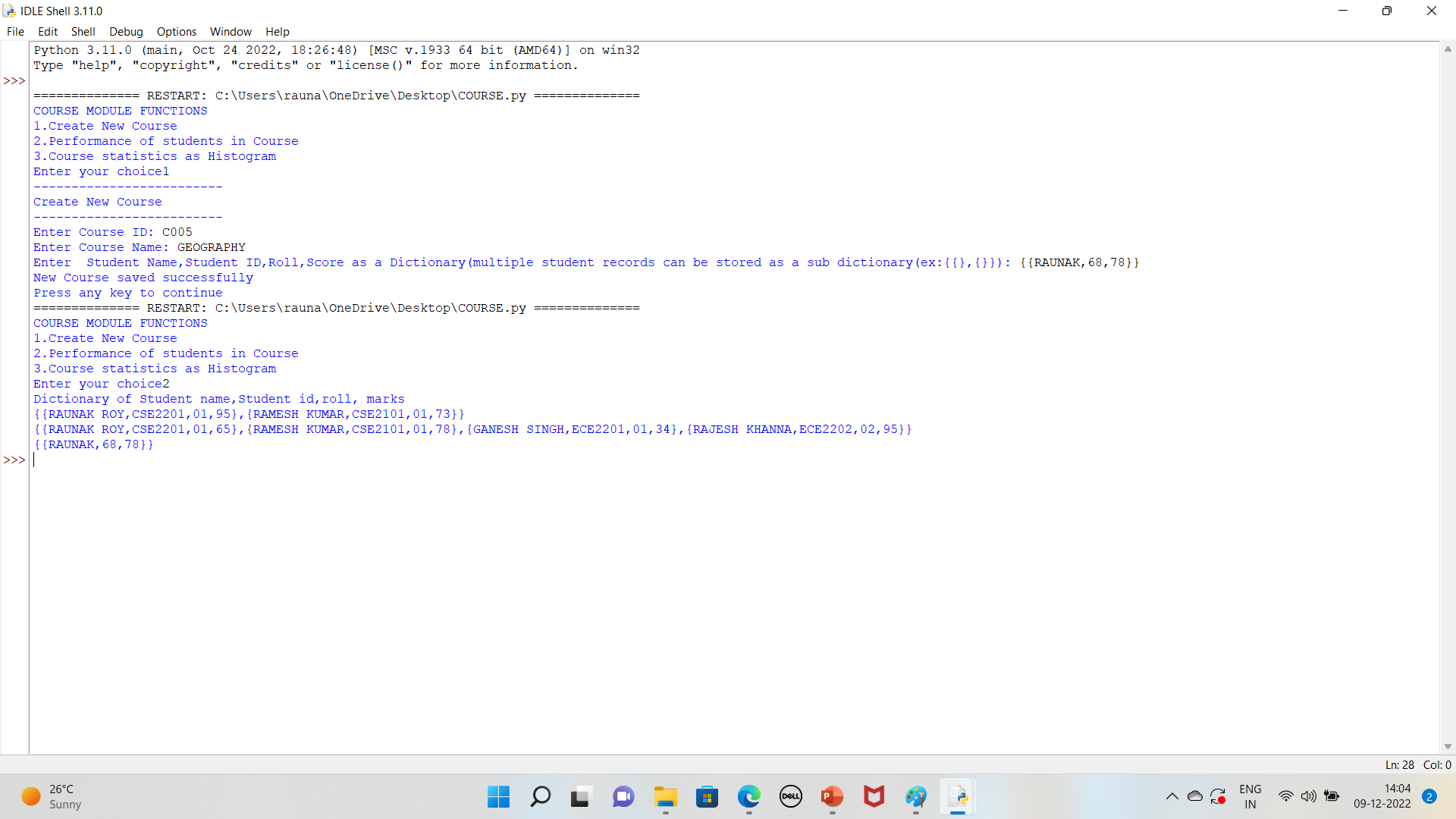
else:

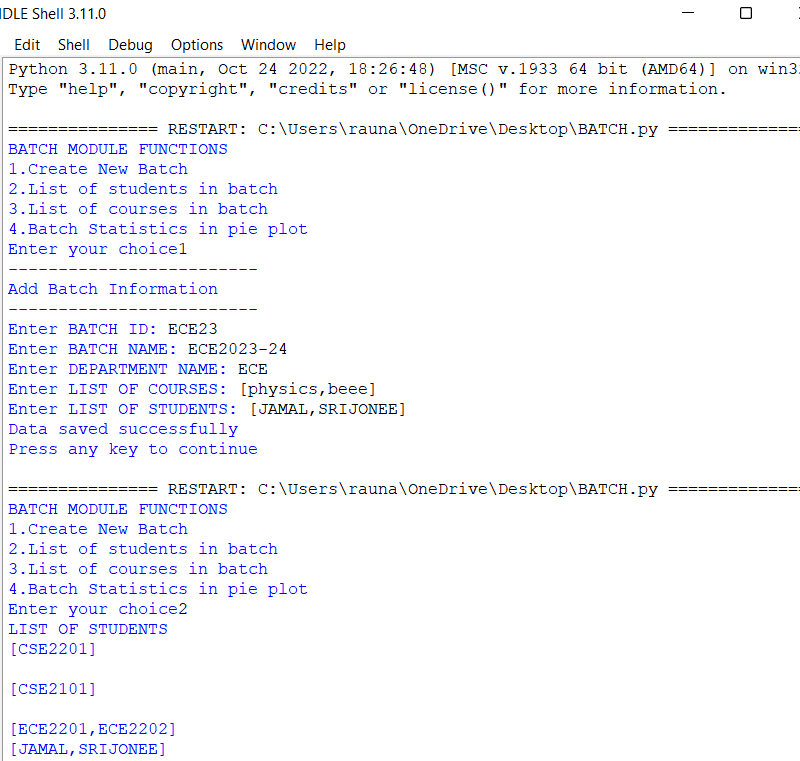
print("Invalid choice")

# Outputs

Sample outputs for the different functions of the modules are given below as screenshots(For displaying the histogram,pie chart etc numpy and matplotlib.pyplot libraries need to be downloaded by pip)-

A)***For STUDENT.py:***

B)***For COURSE.py:***

C)***For BATCH.py:***

D)***For DEPARTMENT.py:***

E)***For EXAMINATION.py:***

******