

### DEPARTMENT OF BASIC SCIENCE AND HUMANITITES

### INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA

**“STUDENT EXAMINATION PORTAL”**

## Submitted by:-

**Name of the Student:** Sefali Srivastava **Enrolment Number:** 12022002004065 **Section:** J

**Class Roll Number:** 56

**Stream:** Information Technology (IT) **Subject:** Programming for Problem Solving **Subject Code:** IVC-101

**Department:** Basic Science and Humanities (BSH)

Under the supervision of:-

### Prof. Swarnendu Ghosh

**Academic Year: 2022-26**

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



## CERTIFICATE OF RECOMMENDATION

We hereby recommend that the project prepared under our supervision by **Sefali Srivastava**, entitled **“Student Examination Portal”** be accepted in partial fulfilment of the requirements for the degree of partial fulfilment of the first semester.

Head of the Department Project Supervisor Basic Sciences and Humanities

IEM, Kolkata

# Introduction:-

This project is assigned to us for developing a report card of student’s examinations with the help of basic python programming language.

* 1. **Objective:**

The basic aim of the project is to create student examination portal where we need to put up basic student details and thereby with the help of a python programming, we have to create a report card by maintaining separate CSV files for the student, course, batch, department and examination.

## Organization of the Project:

The project is organised into 5 different module, namely:-

STUDENT: We have to create a student with the help of his/her basic details *vis.* student ID, name, roll number and batch name and then generate a report card showing percentage, grades in each subject and whether he have passed or failed.

COURSE: After this, we have to create a new course with details *vis.* his/her course ID, course name and marks obtained followed by his/her performance in that course and course stats with the help of a histogram.

STREAM: Now we are to create a new batch providing details *vis.* Stream

ID, stream name, department name, list of courses and list of students followed by viewing all students, all courses taught, complete performance of all the students and course stats with the help of a pie chart containing all the percentages.

DEPARTMENT: Now we have to create a new department with details *vis.* department ID, department name and list of batches followed by a clear picture of all the batches in the department, average performance of all the batches in the department and department stats with the help of a line plot.

EXAMINATION: Lastly, we are done with entering marks of all students in the examination, performance of all students in the examination and finally displaying examination stats with the help of a scatter plot.

# Database Descriptions:-

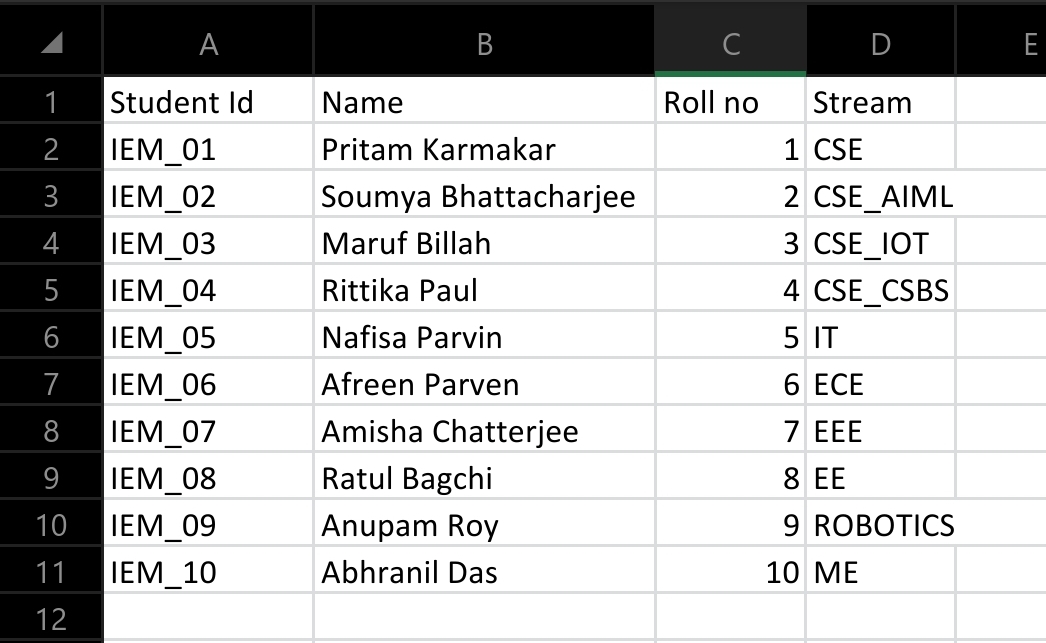
The database used in the project is CSV files.

A CSV (Comma Separated Value) file is a type of plain text file that uses specific structure to arrange tabular data. Because its a plain text file, it can contain only actual text data—in other words, printable ASCII or Unicode characters. The structure of a CSV file is given away by its name. Normally, CSV files use a comma to separate each specific data value.

* 1. **Database Samples:**

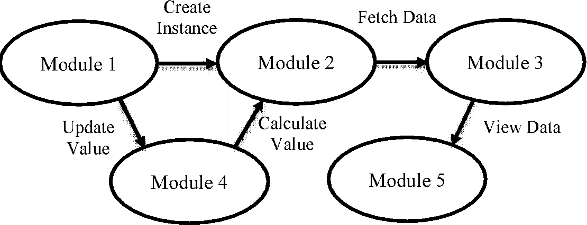
CSV sample of 1st module:--

*Students.csv*



# Data Flow and E-R Diagrams:-

Demonstrates the dependency of all the python modules written using a data flow diagram.



# Programs:-

Python program of 1st module:--

*Students Examination Portal.py*

import csv

student\_fields = ['Student ID', 'Name', ' Roll Number', 'Stream'] student\_database = 'students.csv'

def display\_menu():

print(" ") print("Student Examination Portal.") print(" ") print("1.Add New Student") print("2.View Students") print("3.Update Student") print("4.Delete Student") print("5.Calculate Grade") print("6.Quit")

def add\_student():

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 enter to continue.")

def view\_students():

global student\_fields global student\_database

print("--- Student Records: ---") print(" ")

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

for x in student\_fields: print(x, end='\t|')

print("\n

")

for row in reader: for item in row:

print(item, end="\t|") print("\n")

input("Press enter to continue.")

def update\_student(): global student\_fields global student\_database

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

roll = input("Enter Student ID 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[0]: 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)

print("Student ID", roll, "updated successfully!") else:

print("Student ID not found in our database!")

input("Press enter to continue.")

def delete\_student(): global student\_fields global student\_database

print("--- Delete Student: ---") print(" ")

roll = input("Enter Student ID to delete: ") 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[0]: 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("Student ID", roll, "deleted successfully!") else:

print("Student ID not found in our database!") input("Press enter to continue.")

def search\_student(): global student\_fields global student\_database

print("--- Search Student: ---") print(" ")

roll = input("Enter Student ID to search: ")

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

for row in reader: if len(row) > 0:

if roll == row[0]:

print("Student found with the following details...") print("Student ID: ", row[0])

print("Name: ", row[1])

print(" Roll Number: ", row[2]) print("Stream: ", row[3]) grade()

break

else:

print("Student ID not found in our database!") input("Press enter to continue.")

def grade():

print("Enter marks out of 100:")

m1 = int(input("Enter marks in 1st subject: ")) m2 = int(input("Enter marks in 2nd subject: ")) m3 = int(input("Enter marks in 3rd subject: ")) m4 = int(input("Enter marks in 4th subject: ")) m5 = int(input("Enter marks in 5th subject: ")) tmarks = m1+m2+m3+m4+m5

per = (tmarks)//5 if per >= 90:

print("Total marks = ", tmarks, "\nPercentage = ", per, "\nGrade= A\nStatus: Passed!")

elif per >= 80 and per < 90:

print("Total marks = ", tmarks, "\nPercentage = ", per, "\nGrade= B\nStatus: Passed!")

elif per >= 70 and per < 80:

print("Total marks = ", tmarks, "\nPercentage = ", per, "\nGrade= C\nStatus: Passed!")

elif per >= 60 and per < 70:

print("Total marks = ", tmarks, "\nPercentage = ", per, "\nGrade= D\nStatus: Passed!")

elif per >= 50 and per < 60:

print("Total marks = ", tmarks, "\nPercentage = ", per, "\nGrade= E\nStatus: Passed!")

else:

print("Total marks = ", tmarks, "\nPercentage = ", per, "\nGrade= F\nStatus: Failed!")

while True:

display\_menu()

choice = input("Enter your choice: ") if choice == '1':

add\_student() elif choice == '2':

view\_students() elif choice == '3':

update\_student() elif choice == '4':

delete\_student() elif choice == '5':

search\_student() else:

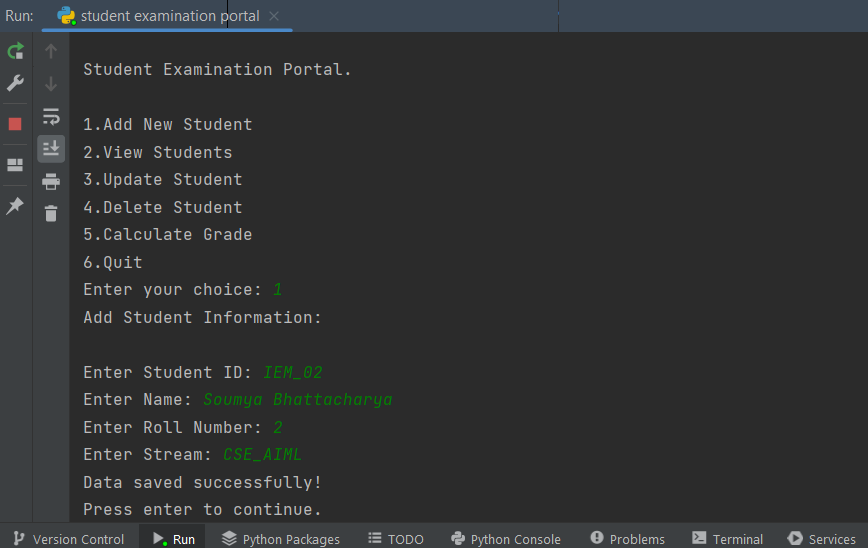
break

print(" ") print("Thank you for using our system.") print(" ")

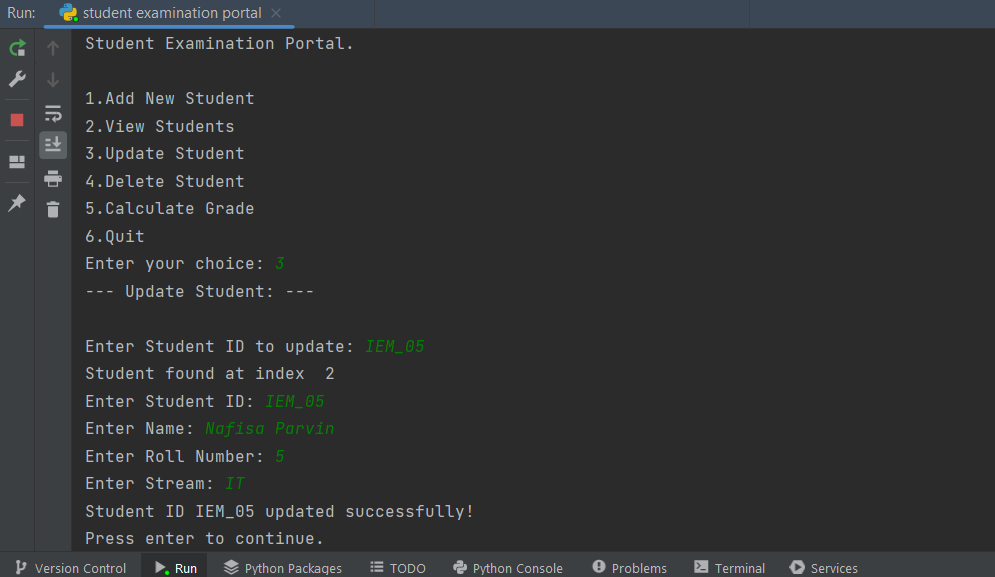
# Outputs:-

Sample outputs(screenshot) to demonstrate the functionalities in programs.

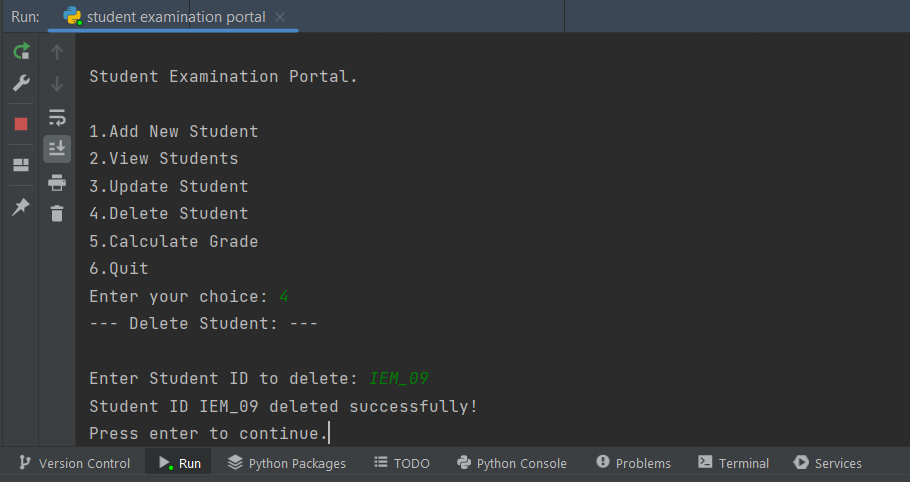
1. Creating a student using Student ID, Name, Class Roll Number and Batch Name.



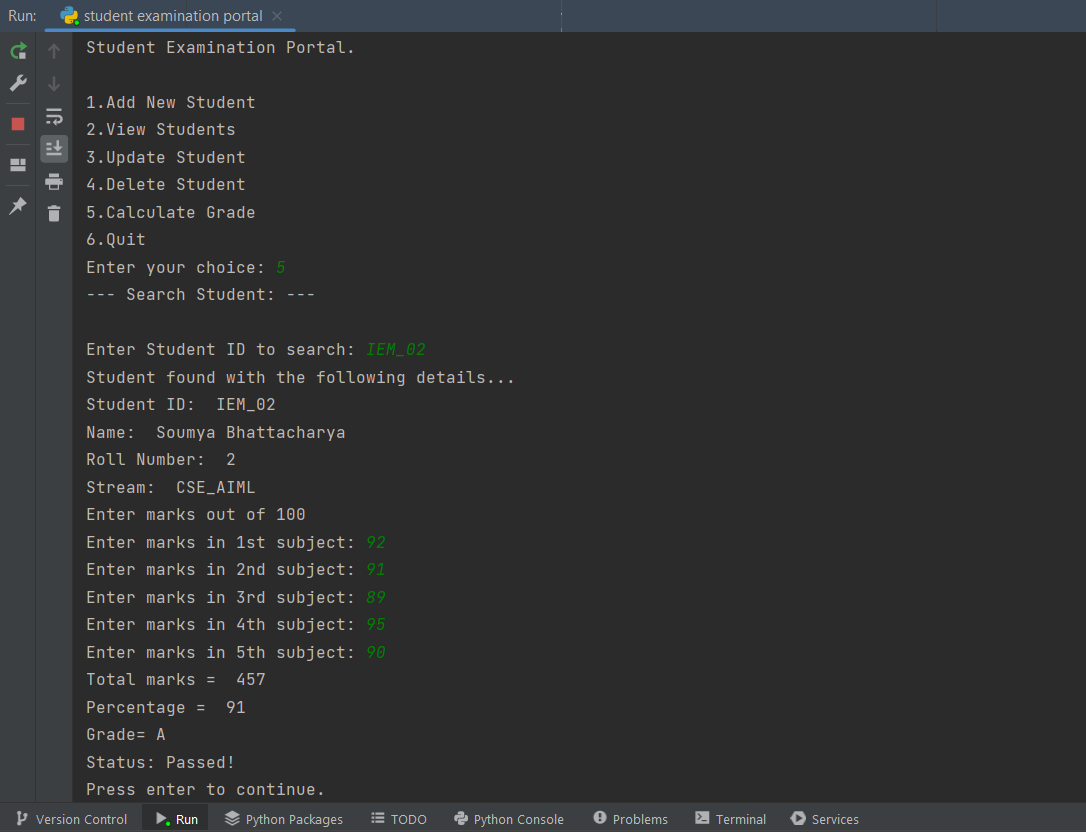
1. Updating student details.



1. Removing a student from the database.



1. Generating a report card (text file)of student showing percentage , grade in each subject and wheather he passed or failed with all the marks uploaded.



**THANK YOU!!**