

STUDENT EXAMINATION PORTAL

Submitted by

Name of the Students: AQUASA AZIZ

Enrolment Number: 12022002018003

Section: F

Class Roll Number: 26

Stream: Computer Science and Business System

Subject: Programming for Problem Solving with Python

Subject Code: IVC101

Department: Basic Science and Humanities

Under the supervision of
Prof. Dr. Swarnendu Ghosh

Academic Year: 2022-26

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE FIRST SEMESTER



DEPARTMENT OF BASIC SCIENCE AND HUMANITIES



Edit with WPS Office

INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA



CERTIFICATE OF RECOMMENDATION

We hereby recommend that the project prepared under our supervision by AQUASA AZIZ, 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
Basic Sciences and Humanities
IEM, Kolkata

Project Supervisor



Edit with WPS Office

1 Introduction

If we see at the present scenario we can clearly understand that it is a digital very educational institutions or big companies need a system to keep a record of the data of their students and employees respectively. The best way to maintain these record is by creating separate Databases and storing the necessary data. In this project we have mainly used the PYTHON Programming Language to make a database which can be further used to store necessary data.

PYTHON is a easy to understandable and user friendly language so anyone can make a programme to make such data bases according to their needs

1.1 Objective

The main objective of this project is to develope a programme for creating a database by which we can take data from the user and store it in the desired cells, Because of these project we got to learn "How to create a Database", "Relationship between several databases" , and "How to create a database using PYTHON Programming Language"

1.2 Organization of the Project

This project consists of three sections

i) **Taking data from the user:** When we run the programme a few terminal prompts

instruct us to give the correct input.

ii)**Storing the data into different databases:** After taking the inputs from the user the code analyses data and store it in its respective databases.



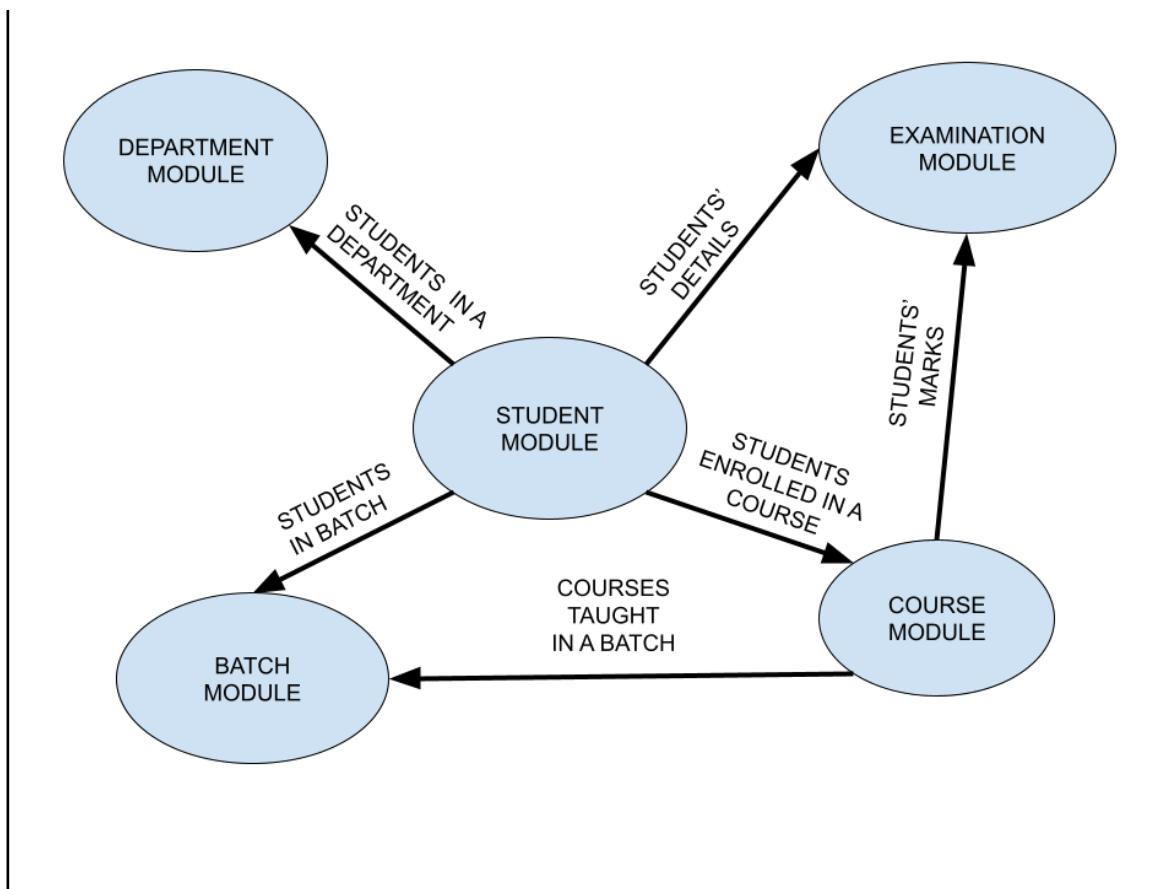
Edit with WPS Office

2 Database Descriptions

There are four databases:

- 1) STUDENT: Stores details of a student
- 2) COURSE: Stores details of all courses
- 3) BATCH: Stores details of all courses
- 4) DEPARTMENT: Stores details of all courses

3 Data Flow and E-R Diagrams





Edit with WPS Office

4. Programs :

I. main.py :

```
while(True):
    print("Press 1 to do student operations")
    print("Press 2 to do course operations")
    print("Press 3 to do batch operations")
    print("Press 4 to do department operations")
    print("Press 5 to do examination operations")
    print("Press 0 to stop")
    x = int(input("Enter your choice: "))
    if(x == 0):
        break
    elif(x == 1):
        from student import *
        while(True):
            print("Press 1 to create a student")
            print("Press 2 to update a student's details")
            print("Press 3 to remove a student")
            print("Press 4 to generate report card of a student")
            print("Press 0 to return to main menu")
            y = int(input("Enter your choice: "))
            if(y == 0):
                break
            elif(y == 1):
                student_id = input("Enter student ID: ")
                student_name = input("Enter student name: ")
                createStudent(student_id, student_name)
            elif(y == 2):
                ostudent_id = input("Enter old student ID: ")
                updateStudent(ostudent_id)
            elif(y == 3):
                student_id = input("Enter student ID: ")
                removeStudent(student_id)
            elif(y == 4):
                student_id = input("Enter student ID: ")
                reportCard(student_id)
            else:
                print("Invalid input. Try again.")
    elif(x == 2):
        from course import *
        while(True):
            print("Press 1 to create a course")
            print("Press 2 to view performance of students on course")
            print("Press 3 to show course statistics as histogram")
            print("Press 0 to return to main menu")
            y = int(input("Enter your choice: "))
```



Edit with WPS Office

```

if(y == 0):
    break
elif(y == 1):
    course_id = input("Enter course ID: ")
    course_name = input("Enter course name: ")
    createCourse(course_id, course_name)
elif(y == 2):
    course_id = input("Enter course ID: ")
    checkPerformance(course_id)
elif(y == 3):
    student_id = input("Enter course ID: ")
    courseStatistics(course_id)
else:
    print("Invalid input. Try again.")
elif(x == 3):
    from batch import *
    while(True):
        print("Press 1 to create a batch")
        print("Press 2 to view all students in a batch")
        print("Press 3 to show all courses in a batch")
        print("Press 4 to view performance of all students in a batch")
        print("Press 5 to view pie chart of percentage all students in a batch")
        print("Press 0 to return to main menu")
        y = int(input("Enter your choice: "))
        if(y == 0):
            break
        elif(y == 1):
            batch_name = input("Enter batch name: ")
            createBatch(batch_name)
        elif(y == 2):
            batch_id = input("Enter batch ID: ")
            viewStudents(batch_id)
        elif(y == 3):
            batch_id = input("Enter batch ID: ")
            viewCourses(batch_id)
        elif(y == 4):
            batch_id = input("Enter batch ID: ")
            viewPerformance(batch_id)
        elif(y == 5):
            batch_id = input("Enter batch ID: ")
            pieChart(batch_id)
        else:
            print("Invalid input. Try again.")
    elif(x == 4):
        from department import *
        while(True):
            print("Press 1 to create a department")
            print("Press 2 to view all batches in a department")
            print("Press 3 to view average performance of all batches in a department")

```



```

print("Press 4 to view line plot of department statistics")
print("Press 0 to return to main menu")
y = int(input("Enter your choice: "))
if(y == 0):
    break
elif(y == 1):
    department_id = input("Enter department ID: ")
    department_name = input("Enter department name: ")
    createDepartment(department_id, department_name)
elif(y == 2):
    department_id = input("Enter department ID: ")
    viewBatches(department_id)
elif(y == 3):
    department_id = input("Enter department ID: ")
    viewPerformanceD(department_id)
elif(y == 4):
    department_id = input("Enter department ID: ")
    linePlot(department_id)
else:
    print("Invalid input. Try again.")
elif(x == 5):
    from examination import *
while(True):
    print("Press 1 to enter marks of all students for an exam")
    print("Press 2 to view performance of all students in an exam")
    print("Press 3 to show examination statistics as a scatter plot")
    print("Press 0 to return to main menu")
    y = int(input("Enter your choice: "))
    if(y == 0):
        break
    elif(y == 1):
        course_id = input("Enter course ID: ")
        enterMarks(course_id)
    elif(y == 2):
        course_id = input("Enter course ID: ")
        viewPerformanceE(course_id)
    elif(y == 3):
        scatterPlot()
    else:
        print("Invalid input. Try again.")
else:
    print("Invalid input. Try again.")

```

II. student.py :

```

import json
import csv
import pandas

```



Edit with WPS Office

```

from batch import createBatch

def createStudent(student_id, name):
    class_roll_number = int(student_id[5:7])
    batch_id = student_id[:5]
    data = [student_id, name, class_roll_number, batch_id]
    csv_reader = []
    with open("student.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))
    with open("student.csv", "a", newline = "\n") as f:
        for i in range(0, len(csv_reader)):
            if(csv_reader[i][0] == student_id):
                print("Student ID already exists")
                return
        csv_writer = csv.writer(f)
        csv_writer.writerow(data)
    with open("batch.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

check = 0
for i in range(0, len(csv_reader)):
    if(csv_reader[i][0] == batch_id):
        check = 1
    if(csv_reader[i][4] == ""):
        csv_reader[i][4] = csv_reader[i][4] + student_id
    else:
        csv_reader[i][4] = csv_reader[i][4] + ":" + student_id
df = pandas.read_csv("batch.csv")
df.loc[i-1, "list_of_students"] = csv_reader[i][4]
df.to_csv("batch.csv", index = False)
if(check == 0):
    print("Batch does not exist.... Creating new batch")
    batch_name = batch_id[:3] + " 20" + batch_id[3:] + "-" + str(int(batch_id[3:]) + 4)
    createBatch(batch_name)
    with open("batch.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

courses = []
for i in range(0, len(csv_reader)):
    if(csv_reader[i][0] == batch_id):
        courses = list(csv_reader[i][3].split(":"))
with open("course.csv", "r", newline = "\n") as f:
    csv_reader = list(csv.reader(f, delimiter=","))

for i in range(0, len(csv_reader)):
    for j in range(0, len(courses)):
        if(csv_reader[i][0] == courses[j]):
            if(csv_reader[i][2] == ""):
                temp = {}
                temp[student_id] = 0
                csv_reader[i][2] = json.dumps(temp)
            else:

```



```

        temp = json.loads(csv_reader[i][2])
        temp[student_id] = 0
        csv_reader[i][2] = json.dumps(temp)
        df = pandas.read_csv("course.csv")
        df.loc[i-1, "marks_obtained"] = csv_reader[i][2]
        df.to_csv("course.csv", index = False)

def updateStudent(ostudent_id):
    csv_reader = []
    with open("student.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    check = 0
    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == ostudent_id):
            check = 1
            break
    if(check == 0):
        print("Student ID does not exist")
        return
    while(True):
        print("Press 1 to update name")
        print("Press 2 to update student ID")
        print("Press 0 to exit")
        x = int(input("Enter your choice: "))
        if(x == 0):
            break
        elif(x == 1):
            name = input("Enter updated name: ")
            df = pandas.read_csv("student.csv")
            df.loc[i-1, "Name"] = name
            df.to_csv("student.csv", index = False)
        elif(x == 2):
            nstudent_id = input("Enter updated student ID: ")
            df = pandas.read_csv("student.csv")
            df.loc[i-1, "Student_ID"] = nstudent_id
            df.to_csv("student.csv", index = False)
            removeStudent(ostudent_id)
            createStudent(nstudent_id, csv_reader[i][1])
            ostudent_id = nstudent_id
            with open("student.csv", "r", newline = "\n") as f:
                csv_reader = list(csv.reader(f, delimiter=","))

    else:
        print("Invalid input. Try again.")

def removeStudent(student_id):
    csv_reader = []
    with open("student.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    check = 0

```



```

for i in range(0, len(csv_reader)):
    if(csv_reader[i][0] == student_id):
        check = 1
        break
if(check == 0):
    print("Student ID does not exist")
    return
df = pandas.read_csv("student.csv")
df.set_index("Student_ID")
df = df.drop(df.index[i-1])
df.to_csv("student.csv", index = False)
with open("course.csv", "r", newline = "\n") as f:
    csv_reader = list(csv.reader(f, delimiter=","))

for i in range(0, len(csv_reader)):
    if(i == 0):
        continue
    temp = csv_reader[i][2]
    temp = json.loads(temp)
    if student_id in temp:
        del temp[student_id]
    csv_reader[i][2] = json.dumps(temp)
df = pandas.read_csv("course.csv")
for i in range(1, len(csv_reader)):
    df.loc[i-1, "marks_obtained"] = csv_reader[i][2]
df.to_csv("course.csv", index = False)
with open("batch.csv", "r", newline = "\n") as f:
    csv_reader = list(csv.reader(f, delimiter=","))

for i in range(0, len(csv_reader)):
    if(i == 0):
        continue
    temp = list(csv_reader[i][4].split(":"))
    if student_id in temp:
        temp.remove(student_id)
    a = ":"
    csv_reader[i][4] = a.join(temp)
df = pandas.read_csv("batch.csv")
for i in range(1, len(csv_reader)):
    df.loc[i-1, "list_of_students"] = csv_reader[i][4]
df.to_csv("batch.csv", index = False)

def reportCard(student_id):
    name = ""
    csv_reader= []
    with open("student.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    check = 0
    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == student_id):
            check = 1

```



```

        name = csv_reader[i][1]
        break
    if(check == 0):
        print("Student ID does not exist")
        return
    f = open((student_id + ".txt"), "w")
    a = "Student ID: " + student_id + "\n"
    b = "Name: " + name + "\n"
    f.writelines([a, b])
    with open("course.csv", "r", newline = "\n") as fx:
        csv_reader = list(csv.reader(fx, delimiter = ","))
    marks = []
    subjects = []
    for i in range(1, len(csv_reader)):
        marks.append(json.loads(csv_reader[i][2]))
        subjects.append(csv_reader[i][1])
    total_marks = 0
    divs = 0
    for i in range(0, len(subjects)):
        temp = marks[i]
        if(isinstance(temp.get(student_id), int)):
            subject_marks = "Marks in " + subjects[i] + ": " + str(temp.get(student_id)) + "% \n"
            divs += 1
            total_marks += temp.get(student_id)
            f.write(subject_marks)
    grade = "Grade obtained: " + gradeCheck(total_marks/divs) + " \n"
    f.write(grade)
    f.close()

def gradeCheck(a):
    if(a >= 90):
        return "A"
    elif(a >= 80):
        return "B"
    elif(a >= 70):
        return "C"
    elif(a >= 60):
        return "D"
    elif(a >= 50):
        return "E"
    else:
        return "F"

```

III. course.py :

```

import json
import csv
import pandas

```



Edit with WPS Office

```

import matplotlib.pyplot
from collections import Counter
from student import gradeCheck
from batch import createBatch

def createCourse(course_id, course_name):
    csv_reader = []
    with open("course.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == course_id):
            print("Course ID already exists")
            return

    print("Enter batches in which course is included: ")
    students = []
    while(True):
        batch_id = input("Enter batch ID (to stop enter STOP): ")
        if(batch_id.upper() == "STOP"):
            break
        else:
            check = 0
            for i in range(0, len(csv_reader)):
                with open("batch.csv", "r", newline = "\n") as f:
                    csv_reader = list(csv.reader(f, delimiter=","))

                if(csv_reader[i][3] != ""):
                    temp = csv_reader[i][3].split(":")
                    for x in temp:
                        if(x == course_id):
                            print("Course already added")
                            continue
                if(csv_reader[i][0] == batch_id):
                    check = 1
                if(csv_reader[i][3] == ""):
                    csv_reader[i][3] = csv_reader[i][3] + course_id
                else:
                    csv_reader[i][3] = csv_reader[i][3] + ":" + course_id
            df = pandas.read_csv("batch.csv")
            df.loc[i-1, "list_of_courses"] = csv_reader[i][3]
            df.to_csv("batch.csv", index = False)

            if(check == 0):
                print("Batch does not exist.... Creating new batch")
                batch_name = batch_id[:3] + " 20" + batch_id[3:] + "-" + str(int(batch_id[3:])) + 4
                createBatch(batch_name)
                with open("batch.csv", "r", newline = "\n") as f:
                    csv_reader = list(csv.reader(f, delimiter=","))

                csv_reader[len(csv_reader) - 1][3] = csv_reader[len(csv_reader) - 1][3] + course_id
                df = pandas.read_csv("batch.csv")
                df.loc[len(csv_reader) - 2, "list_of_courses"] = csv_reader[len(csv_reader) - 1][3]
                df.to_csv("batch.csv", index = False)

```



```

        with open("batch.csv", "r", newline = "\n") as f:
            csv_reader = list(csv.reader(f, delimiter=","))
            for i in range(0, len(csv_reader)):
                if(csv_reader[i][0] == batch_id):
                    students += csv_reader[i][4].split(":")
temp = {}
for a in students:
    temp[a] = 0
data = [course_id, course_name, json.dumps(temp)]
with open("course.csv", "a", newline = "\n") as f:
    csv_writer = csv.writer(f)
    csv_writer.writerow(data)

def checkPerformance(course_id):
    csv_reader = []
    data = []
    with open("course.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    check = 0
    student_marks = {}
    for i in range(1, len(csv_reader)):
        if(csv_reader[i][0] == course_id):
            check = 1
            student_marks = json.loads(csv_reader[i][2])
            break
    if(check == 0):
        print("Course ID does not exist")
        return data
    student_ids = list(student_marks.keys())
    with open("student.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    for i in range(0, len(student_ids)):
        for j in range(0, len(csv_reader)):
            if(student_ids[i] == csv_reader[j][0]):
                print("Student ID: " + student_ids[i])
                print("Student Name: " + csv_reader[j][1])
                print("Student Roll Number: " + csv_reader[j][2])
                print("Marks obtained: " + str(student_marks.get(student_ids[i])))
                print()
                data.append([student_ids[i], csv_reader[j][1], csv_reader[j][2],
student_marks.get(student_ids[i])])
    return data

def courseStatistics(course_id):
    csv_reader = []
    with open("course.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    check = 0
    for i in range(0, len(csv_reader)):

```



```

if(csv_reader[i][0] == course_id):
    check = 1
    break
if(check == 0):
    print("Course ID does not exist")
    return
x = checkPerformance(course_id)
grades = []
for a in x:
    grades.append(gradeCheck(a[3]))
grades.sort()
letter_counts = Counter(grades)
df = pandas.DataFrame.from_dict(letter_counts, orient='index')
df.plot(kind='bar')
matplotlib.pyplot.show()

```

IV. Batch_database.py :

```

import csv
import pandas
import json
from matplotlib import pyplot
from department import createDepartment

def createBatch(batch_name):
    batch_id = batch_name[:3] + batch_name[6:8]
    department_id = batch_id[:3]
    csv_reader = []
    with open("batch.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == batch_id):
            print("Batch ID already exists")
            return
    data = [batch_id, batch_name, department_id, "", ""]
    with open("batch.csv", "a", newline = "\n") as f:
        csv_writer = csv.writer(f)
        csv_writer.writerow(data)
    print("Enter courses in batch: ")
    while(True):
        course_id = input("Enter course ID (to stop enter STOP): ")
        with open("batch.csv", "r", newline = "\n") as f:
            csv_reader = list(csv.reader(f, delimiter=","))

        if(csv_reader[len(csv_reader) - 1][3] != ""):
            check = 0
            temp = csv_reader[len(csv_reader) - 1][3].split(":")
            for x in temp:
                if(x == course_id):

```



```

        print("Course already added")
        check = 1
    if(check == 1):
        continue
    if(course_id.upper() == "STOP"):
        break
    else:
        with open("course.csv", "r", newline = "\n") as f:
            csv_reader = list(csv.reader(f, delimiter=","))

        check = 0
        for i in range(0, len(csv_reader)):
            if(csv_reader[i][0] == course_id):
                with open("batch.csv", "r", newline = "\n") as f:
                    csv_reader = list(csv.reader(f, delimiter=","))

                check = 1
                if(csv_reader[len(csv_reader) - 1][3] == ""):
                    csv_reader[len(csv_reader) - 1][3] = csv_reader[len(csv_reader) - 1][3] + course_id
                else:
                    csv_reader[len(csv_reader) - 1][3] = csv_reader[len(csv_reader) - 1][3] + ":" +
course_id
                    df = pandas.read_csv("batch.csv")
                    df.loc[len(csv_reader) - 2, "list_of_courses"] = csv_reader[len(csv_reader) - 1][3]
                    df.to_csv("batch.csv", index = False)
                if(check == 0):
                    print("Course does not exist. Please create course first.")
            with open("department.csv", "r", newline = "\n") as f:
                csv_reader = list(csv.reader(f, delimiter=","))

            check = 0
            for i in range(0, len(csv_reader)):
                if(csv_reader[i][0] == department_id):
                    check = 1
                    if(csv_reader[i][2] == ""):
                        csv_reader[i][2] = csv_reader[i][2] + batch_id
                    else:
                        csv_reader[i][2] = csv_reader[i][2] + ":" + batch_id
                    df = pandas.read_csv("department.csv")
                    df.loc[i-1, "list_of_batches"] = csv_reader[i][2]
                    df.to_csv("department.csv", index = False)
            if(check == 0):
                print("Department does not exist.... Creating new department")
                department_name = input("Enter department name: ")
                createDepartment(department_id, department_name)
                with open("department.csv", "r", newline = "\n") as f:
                    csv_reader = list(csv.reader(f, delimiter=","))

                csv_reader[len(csv_reader) - 1][2] = csv_reader[len(csv_reader) - 1][2] + batch_id
                df = pandas.read_csv("department.csv")
                df.loc[len(csv_reader) - 2, "list_of_batches"] = csv_reader[len(csv_reader) - 1][2]
                df.to_csv("department.csv", index = False)

```



```

def viewStudents(batch_id):
    with open("batch.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=",")) 
    check = 0
    students = []
    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == batch_id):
            check = 1
            students = csv_reader[i][4].split(":")
            break
    if(check == 0):
        print("Batch ID does not exist")
        return
    print("Students in " + batch_id + ":")
    for student in students:
        print(student)

def viewCourses(batch_id):
    with open("batch.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=",")) 
    check = 0
    courses = []
    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == batch_id):
            check = 1
            courses = csv_reader[i][3].split(":")
            break
    if(check == 0):
        print("Batch ID does not exist")
        return
    print("Courses in " + batch_id + ":")
    for course in courses:
        print(course)

def viewPerformance(batch_id):
    with open("batch.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=",")) 
    check = 0
    students = []
    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == batch_id):
            check = 1
            students = csv_reader[i][4].split(":")
            break
    if(check == 0):
        print("Batch ID does not exist")
        return
    for student in students:
        with open("student.csv", "r", newline = "\n") as f:

```



Edit with WPS Office

```

    csv_reader = list(csv.reader(f, delimiter=","))

for i in range(0, len(csv_reader)):
    if(student == csv_reader[i][0]):
        print("Student ID: " + student)
        print("Student Name: " + csv_reader[i][1])
        print("Student Roll Number: " + csv_reader[i][2])
with open("course.csv", "r", newline = "\n") as f:
    csv_reader = list(csv.reader(f, delimiter=","))

all_marks = []
for i in range(1, len(csv_reader)):
    all_marks.append(json.loads(csv_reader[i][2]))
total_marks = 0
divs = 0
for subjects in all_marks:
    if(isinstance(subjects.get(student), int)):
        total_marks += subjects.get(student)
        divs += 1
print("Percentage obtained: " + str(total_marks/divs))
print()

```

```

def pieChart(batch_id):
    with open("batch.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    check = 0
    students = []
    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == batch_id):
            check = 1
            students = csv_reader[i][4].split(":")
            break
    if(check == 0):
        print("Batch ID does not exist")
        return
    percentages = [">=90", ">=80", ">=70", ">=60", ">=50", "Failed"]
    numbers = [0, 0, 0, 0, 0, 0]
    for student in students:
        with open("course.csv", "r", newline = "\n") as f:
            csv_reader = list(csv.reader(f, delimiter=","))

        all_marks = []
        for i in range(1, len(csv_reader)):
            all_marks.append(json.loads(csv_reader[i][2]))
        total_marks = 0
        divs = 0
        for subjects in all_marks:
            if(isinstance(subjects.get(student), int)):
                total_marks += subjects.get(student)
                divs += 1
        percentage = total_marks/divs

```



```

if(percentage >= 90):
    numbers[0] += 1
elif(percentage >= 80):
    numbers[1] += 1
elif(percentage >= 70):
    numbers[2] += 1
elif(percentage >= 60):
    numbers[3] += 1
elif(percentage >= 50):
    numbers[4] += 1
else:
    numbers[5] += 1
for i in range(len(numbers) - 1, -1, -1):
    if(numbers[i] == 0):
        del numbers[i]
        del percentages[i]
pyplot.pie(numbers, labels = percentages)
pyplot.show()

```

V. departmant.py :

```

import json
import csv
from matplotlib import pyplot

def createDepartment(department_id, department_name):
    csv_reader = []
    with open("department.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter = ","))
    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == department_id):
            print("Department ID already exists")
            return
    data = [department_id, department_name, ""]
    with open("department.csv", "a", newline = "\n") as f:
        csv_writer = csv.writer(f)
        csv_writer.writerow(data)

def viewBatches(department_id):
    with open("department.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter = ","))
    check = 0
    batches = []
    for i in range(0, len(csv_reader)):
        if(csv_reader[i][0] == department_id):
            check = 1
            batches = csv_reader[i][2].split(":")
            break

```



```

if(check == 0):
    print("Department ID does not exist")
    return
print("Batches in " + department_id + ":")
for batch in batches:
    print(batch)

def viewPerformanceD(department_id):
    with open("department.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    check = 0
    batches = []
    for i in range(1, len(csv_reader)):
        if(csv_reader[i][0] == department_id):
            check = 1
            batches = csv_reader[i][2].split(":")
            break
    if(check == 0):
        print("Department ID does not exist")
        return
    if(len(batches) == 0):
        print("No batches in department")
        return
    performances = []
    for batch in batches:
        students = []
        student_performances = []
        with open("batch.csv", "r", newline = "\n") as f:
            csv_reader = list(csv.reader(f, delimiter=","))

        for i in range(0, len(csv_reader)):
            if(csv_reader[i][0] == batch):
                students = csv_reader[i][4].split(":")
                break
        for student in students:
            with open("course.csv", "r", newline = "\n") as f:
                csv_reader = list(csv.reader(f, delimiter=","))

            all_marks = []
            for i in range(1, len(csv_reader)):
                all_marks.append(json.loads(csv_reader[i][2]))
            total_marks = 0
            divs = 0
            for subjects in all_marks:
                if(isinstance(subjects.get(student), int)):
                    total_marks += subjects.get(student)
                    divs += 1
            if(divs != 0):
                student_performances.append(total_marks/divs)
            else:
                student_performances.append(0)

```



```

total_marks = 0
divs = 0
for x in student_performances:
    total_marks += x
    divs += 1
if(divs != 0):
    performances.append(total_marks/divs)
else:
    performances.append(0)
total_marks = 0
divs = 0
for i in range(0, len(batches)):
    total_marks += performances[i]
    divs += 1
avg_percentage = 0
if(divs != 0):
    avg_percentage = total_marks/divs
print("Average percentage obtained by all batches in " + department_id + ": " +
str(avg_percentage))

def linePlot(department_id):
    with open("department.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))
    check = 0
    batches = []
    for i in range(1, len(csv_reader)):
        if(csv_reader[i][0] == department_id):
            check = 1
            batches = csv_reader[i][2].split(":")
            break
    if(check == 0):
        print("Department ID does not exist")
        return
    if(len(batches) == 0):
        print("No batches in department")
        return
    performances = []
    for batch in batches:
        students = []
        student_performances = []
        with open("batch.csv", "r", newline = "\n") as f:
            csv_reader = list(csv.reader(f, delimiter=","))
        for i in range(0, len(csv_reader)):
            if(csv_reader[i][0] == batch):
                students = csv_reader[i][4].split(":")
                break
        for student in students:
            with open("course.csv", "r", newline = "\n") as f:
                csv_reader = list(csv.reader(f, delimiter=","))
```



```

all_marks = []
for i in range(1, len(csv_reader)):
    all_marks.append(json.loads(csv_reader[i][2]))
total_marks = 0
divs = 0
for subjects in all_marks:
    if(isinstance(subjects.get(student), int)):
        total_marks += subjects.get(student)
        divs += 1
if(divs != 0):
    student_performances.append(total_marks/divs)
else:
    student_performances.append(0)
total_marks = 0
divs = 0
for x in student_performances:
    total_marks += x
    divs += 1
if(divs != 0):
    performances.append(total_marks/divs)
else:
    performances.append(0)
pyplot.plot(batches, performances)
pyplot.show()

```

VI. examination.py :

```

import csv
import json
import pandas
from matplotlib import pyplot

def enterMarks(course_id):
    csv_reader = []
    with open("course.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter=","))

    check = 0
    course_name = ""
    student_marks = {}
    for i in range(1, len(csv_reader)):
        if(csv_reader[i][0] == course_id):
            check = 1
            course_name = csv_reader[i][1]
            student_marks = json.loads(csv_reader[i][2])
            break
    if(check == 0):
        print("Course ID does not exist")

```



Edit with WPS Office

```

        return
student_ids = list(student_marks.keys())
print("Course name: " + course_name)
for student in student_ids:
    marks = int(input("Enter marks obtained by " + student + ": "))
    student_marks[student] = marks
df = pandas.read_csv("course.csv")
df.loc[i - 1, "marks_obtained"] = json.dumps(student_marks)
df.to_csv("course.csv", index = False)

def viewPerformanceE(course_id):
    csv_reader = []
    with open("course.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter = ","))
    check = 0
    student_marks = {}
    for i in range(0, len(csv_reader)):
        if(csv_reader[i][1] == course_id):
            check = 1
            student_marks = json.loads(csv_reader[i][2])
            break
    if(check == 0):
        print("Course ID does not exist")
        return
    student_ids = list(student_marks.keys())
    for student in student_ids:
        marks = student_marks[student]
        print("Marks obtained by " + str(marks))

def scatterPlot():
    csv_reader = []
    with open("course.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter = ","))
    all_marks = []
    for i in range(1, len(csv_reader)):
        all_marks.append(json.loads(csv_reader[i][2]))
    batches = []
    students = []
    with open("batch.csv", "r", newline = "\n") as f:
        csv_reader = list(csv.reader(f, delimiter = ","))
    for i in range(0, len(csv_reader)):
        batches.append(csv_reader[i][0])
        students.append(csv_reader[i][4].split(":"))
    for course in all_marks:
        batch_performances = []
        batchesX = []
        for i in range(0, len(batches)):
            total_marks = 0
            divs = 0

```

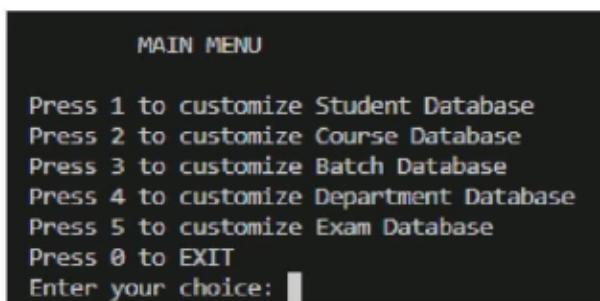


```

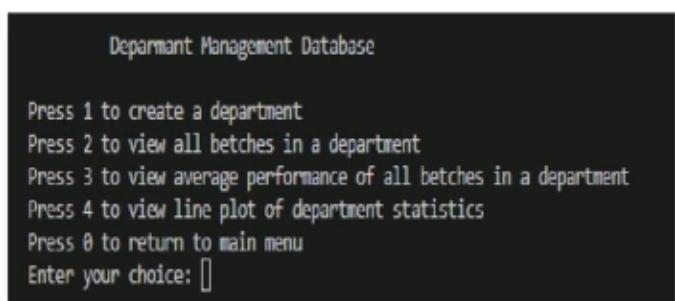
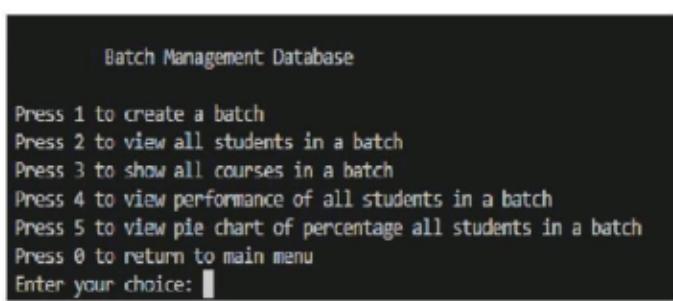
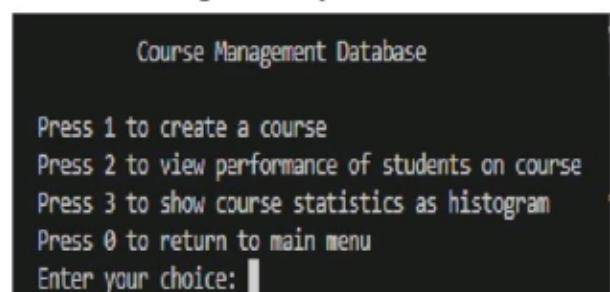
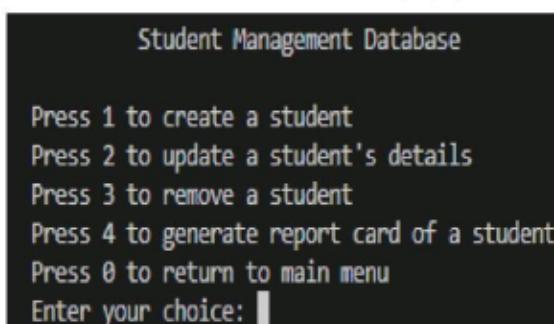
check = 0
for student in students[i]:
    if(student == students[i][0]):
        if(not isinstance(course.get(student), int)):
            check = 1
            break
    total_marks += course.get(student)
    divs += 1
if(check == 1):
    continue
else:
    batchesX.append(batches[i])
    batch_performances.append(total_marks/divs)
pyplot.scatter(batchesX, batch_performances)
pyplot.show()

```

5. Outputs :



In Main Menu if I enter 1,2,3,4 and 5 then, outputs will be respectively -



Examination Management Database

Press 1 to enter marks of all students for an exam
Press 2 to view performance of all students in an exam
Press 3 to show examination statistics as a scatter plot
Press 0 to return to main menu
Enter your choice: |

and the created tables are -

1. student.csv :

	A	B	C	D
1	Student_ID	Name	Class_Roll_Number	Batch_ID
2	CSE2201	Setu Kumar	1	CSE22
3	CSE2101	Bill Gates	1	CSE21
4	ECE2201	Billie Eilish	1	ECE22
5	ECE2202	Sundar Pichai	2	ECE22
6				

2. course.csv :

	A	B	C
1	course_id	course_name	marks_obtained
2	C001	Python Programming	{"CSE2201": 95,"CSE2101": 73}
3	C002	Physics	{"CSE2201": 65,"CSE2101": 78,"ECE2201": 34,"ECE2202": 95}
4			

3. batch.csv :

	A	B	C	D	E
1	batch_id	batch_name	department_name	list_of_courses	list_of_students
2	CSE22	CSE 2022-26	CSE	C001:C002	CSE2201
3	CSE21	CSE 2021-25	CSE	C001:C002	CSE2101
4	ECE22	ECE 2022-26	ECE	C002	ECE2201:ECE2202
5					

4. department.csv :

	A	B	C
1	department_id	department_name	list_of_batches
2	CSE	Computer Science and Engineering	CSE22:CSE21
3	ECE	Electronics and Communication Engineering	ECE22
4			
5			



Edit with WPS Office