J.D. TYTLER SCHOOL

ACADEMIC YEAR : 2021-22

PROJECT REPORT ON

SCHOOL MANAGEMENT SYSTEM

ROLL NO :

NAME : DIVYAM SETHI

CLASS : XII - C

SUBJECT : COMPUTER SCIENCE

PROJECT GUIDE: Mr. Parveen Bhatia

Department of Computer Science

J.D. TYTLER SCHOOL

NEW RAJINDER NAGAR

NEW DELHI-110060

J.D. TYTLER SCHOOL

CERTIFICATE OF COMPLETION

This is to certify that __DIVYAM SETHI__ Roll No: _____ has successfully completed the project Work entitled **SCHOOL MANAGEMENT SYSTEM** in the subject Computer Science laid down in the regulations of CBSE for the purpose of Practical Examination in Class XII to be held in J.D. Tytler School on_____.

Under Guidance Of:

Mr. Parveen Bhatia

(Dept. of Computer Science)

Examiner:

Name: _____

Signature:

TABLE OF CONTENTS

ACKNOWLEDGEMENT

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I express deep sense of gratitude to almighty God for giving me strength for the successful completion of the project.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this project.

I gratefully acknowledge the contribution of the individuals who contributed in bringing this project up to this level, who continues to look after me despite my flaws,

I express my deep sense of gratitude to the luminary The Principal Of J.D. TYTLER SCHOOL who has been continuously motivating and extending their helping hand to us.

I express my sincere thanks to the academician The Vice Principal for constant encouragement and the guidance provided during this project

My sincere thanks to **Mr. Parveen Bhatia**, A guide, Mentor all the above a friend, who critically reviewed my project and helped in solving each and every problem, occurred during implementation of the project

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

PROJECT ON SCHOOL MANAGEMENT SYSTEM

INTRODUCTION

Data management is not only a big task when it comes to storing data automatically and systematically , but it is also a difficult task , so in order to handle this problem , we should have a system for software management , which wont only help to maintain the data but will also help you to store it .

This program will help you keep record of your students , his class batch , fees data , maintain result records and generating report cards and distributing to indivisual students through their mail , reducing human efforts and doing in easiest way possible .

ENJOY!!!!!!

OBJECTIVES OF THE PROJECT

The objective of this project is to allow the management to maintain a clean and structured detail of their quiz programme.

Write programs utilizing modern software tools.

1. Apply simple principles effectively when developing small to medium sized projects.
2. Write effective procedural code to store small to medium sized information.
3. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
4. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

PROPOSED SYSTEM

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today's merciless competition where not to wise saying **"to err is human"** no longer valid, it's outdated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a much sophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomization various organisations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paper work has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done.

This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it's an age of computers of and automating such an organization gives the better look.

SOURCE CODE

```
import os

from PIL import Image
from fpdf import FPDF

from tkinter import *
import tkinter as tk

from PIL import Image

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders

#==================================  TIME  ==================================#
import datetime

x=datetime.datetime.now()
year = str(x.year)
month = str(x.month)
date = str(x.day)
#================================  DATABASE  ================================#
import mysql.connector as conn

database = conn.connect( host = 'localhost' , user = 'root' , passwd = 'mysql123' )
dbconn = database.cursor(buffered=True)

school_details = ['School name ' , 'School Address' , 'school code' , 'school Principle' ,
'school email' ]


try :
    school_logo = Image.open('C:/Users/divya/OneDrive/Desktop/PROJECT/LOGO.jpeg')
except :
    pass

try:
    query = 'use school;'
    dbconn.execute(query)
except:
    query = 'create database school;'
    dbconn.execute(query)
    query = 'use  school;'
    dbconn.execute(query)
```

```
    query = f'''create table student ( ADMISSION_NO BIGINT NOT NULL PRIMARY KEY   , NAME
VARCHAR(20) NOT NULL
            ,  class_ varchar(10) NOT NULL , FATHER VARCHAR(20) ,  MOTHER VARCHAR(20)   ,
ADM_DATE VARCHAR(15) NOT NULL
            ,  FEES BIGINT NOT NULL , BALANCE BIGINT    , PHONE VARCHAR(13) NOT NULL ,
EMAIL VARCHAR(30)
            ,  ADDRESS VARCHAR(100) NOT NULL , DOB VARCHAR(15) NOT NULL  ) ; '''
    dbconn.execute(query)
    database.commit()

annonce = '''
enter 'S' OR 's' to get information about school :
enter 'A' OR 'a' to access student details :
enter 'O' or 'o' to get other help :       \n'''
command = input(annonce)
if command == 'S' or command == 's' :
    for detail in school_details:
        print(detail)
    try :
        school_logo.show()
    except :
        pass

elif command == 'A' or command == 'a' :
    announcement = '''
    press 'N' or 'n' to show non-payment students :
    press 'S' OR 's' to get student details by admn_no :
    press 'P' OR 'p' to get student details by phone :
    press 'A' or 'a' to add new student :
    press 'M' or 'm' to modify student details :     \n '''
    command_2 = input(announcement)

    if command_2 == 'N' or command_2 == 'n' :
        query = "SELECT * FROM student where BALANCE > 0;"
        dbconn.execute(query)
        for y in dbconn :
            print(y)

    elif command_2 == 'S' or command_2 == 's' :
        adm_no = input('enter admission number ')
        query = f"SELECT * FROM student where ADMISSION_NO = {adm_no} ;"
        dbconn.execute(query)
        for y in dbconn :
            print(y)

    elif command_2 == 'P' or command_2 == 'p' :
        phn_no = input('enter phone number ')
        query = f"SELECT * FROM student where PHONE = {phn_no} ;"
        dbconn.execute(query)
        for y in dbconn :
            print(y)

    elif command_2 == 'A' or command_2 == 'a' :
        query = f"SELECT count(*) from student ;"
```

```
        dbconn.execute(query)
        for y in dbconn :
             total_no_students = str(int(y[0]) + 1)
        print(total_no_students)

        query = f'desc student;'
        dbconn.execute(query)
        data = []
        data_record = []

        for y in dbconn :
            data.append(y[0])
            if y[0] == 'ADMISSION_NO' :
                data_record.append(str(total_no_students+year))
            elif y[0] == 'ADM_DATE':
                data_record.append(str(year+'-'+month+'-'+date))
            elif y[0] == 'BALANCE' :
                data_record.append(0)
            else :
                x = input(f'{y[0] } ;- ')
                data_record.append(x)

        value = ''
        for xy in data_record :
            if value == '' :
                value = xy
            else :
                 value = str(value) +'," '+ str(xy) + '"'

        query = f'insert into student values ({value});'
        dbconn.execute(query)
        database.commit()
        print('record added succesfully')

    elif  command_2 == 'M' or command_2 == 'm' :
        query = f"USE SCHOOL ;"
        dbconn.execute(query)

        adm_no = input("enter admission number of student to modify changes :")
        query = f"SELECT * FROM student where ADMISSION_NO = {adm_no} ;"
        dbconn.execute(query)
        for y in dbconn :
            data_record = y

        data = []
        query = f'desc student;'
        dbconn.execute(query)
        for y in dbconn :
            data.append(y[0])

        edit_dict = dict()
        for edit in range(len(data)) :
            if data[edit] != 'ADMISSION_NO' :
                entry = input(f"{data[edit]} = {data_record[edit]}   :")
```

```
                edit_dict[data[edit]] = str(entry)

        value = ''
        for xy in edit_dict :
            if (value == '' or value == ' ') == True and  edit_dict[xy] != '' :
                value =  xy + ' = " ' + edit_dict[xy] + ' " '
            elif edit_dict[xy] != '' :
                 value = str(value) + ', ' + xy + '  = " ' + edit_dict[xy] + ' " '

        query = f"USE SCHOOL ;"
        dbconn.execute(query)
        query = f'''UPDATE student
        SET {value}
        WHERE ADMISSION_NO = {adm_no};'''
        dbconn.execute(query)
        database.commit()
        print('changes done successfully')
else :
    batch_year = int(input("ENTER BATCH YEAR :-"))
    try:
        query = f'use classes_{batch_year};'
        dbconn.execute(query)
    except:
        confirm = input('''THIS CLASS BATCH DOES NOT EXIT
        press "Y" OR "y" to create a new batch else press any key ''')
        if confirm == 'Y' or confirm == 'y' :
            query = f'create database classes_{batch_year};'
            dbconn.execute(query)
            query = f'create database results_{batch_year};'
            dbconn.execute(query)

    annonce = '''
    press 'C' OR 'c' to access class details :
    press 'R' OR 'r' to access report card details :
    press any key to exit :       \n'''
    command = input(annonce)

    #====================================#  CLASSES
#====================================#
    if command == 'C' or command == 'c' :
        try:
            query = f'use classes_{batch_year};'
            dbconn.execute(query)
        except:
            confirm = input('''THIS CLASS BATCH DOES NOT EXIT
            press "Y" OR "y" to create a new batch else press any key ''')
            if confirm == 'Y' or confirm == 'y' :
                query = f'create database classes_{batch_year};'
                dbconn.execute(query)

        def get_class_student_detail(class_sec , admn_no):
            query = f"USE SCHOOL ;"
            dbconn.execute(query)
```

```
            query = f"SELECT * FROM student where ADMISSION_NO = {admn_no} ;"
            dbconn.execute(query)
            data_record = ''
            for y in dbconn :
                data_record = y

            data = []
            query = f'desc student;'
            dbconn.execute(query)
            for y in dbconn :
                data.append(y[0])

            edit_dict = dict()
            for edit in range(len(data_record)) :
                if data[edit] != 'ADMISSION_NO' :
                    edit_dict[data[edit]] = data_record[edit]

            query = f'use classes_{batch_year};'
            dbconn.execute(query)

            query = f'''select * from  class_{str(class_sec)}_{str(batch_year)} where
ADMISSION_NO = {admn_no} ; '''
            dbconn.execute(query)
            data_record = ''
            for y in dbconn :
                data_record = y
            if data_record == '' :
                print(f'{admn_no} not present in class {class_sec}')
            data = []
            query = f'desc class_{str(class_sec)}_{str(batch_year)};'
            dbconn.execute(query)
            for y in dbconn :
                data.append(y[0])

            for edit in range(len(data_record)) :
                edit_dict[(data[edit])] = data_record[edit]

            print(edit_dict)

        announcement = f'''
        press 'S' or 's' to show classes in batch {batch_year} :
        press 'A' OR 'a' to add a new class :
        press 'D' OR 'd' to get student details of a class :
        press 'C' OR 'c' to get students of a class :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student details :
        press 'R' or 'r' to remove a student :  \n '''

        command = input(announcement)
        if command ==  'S' or command == 's' :
            query = f'use classes_{batch_year};'
            dbconn.execute(query)

            query = f'show tables;'
```

```
            dbconn.execute(query)
            for y in dbconn :
              print(y)

        elif command == 'A' or command == 'a' :
            query = f'use classes_{batch_year};'
            dbconn.execute(query)

            class_sec = input('enter class and section to create a new record :')
            class_subject = int(input(f'enter subjects allocated in class {class_sec} :'))
            X = ''
            for i in range(class_subject):
                X = X +f', SUBJECT_{i+1} VARCHAR(20) '
            query = f'''create table class_{str(class_sec)}_{str(batch_year)} (
ADMISSION_NO BIGINT NOT NULL PRIMARY KEY
             , CLASS_TEACHER VARCHAR(20) NOT NULL , BEHAVIOUR VARCHAR(20) {X}  ) ; '''
            dbconn.execute(query)
            database.commit()

        elif command == 'D' or command == 'd' :
            class_sec = input('enter class and section to get student details :')
            admn_no = input('enter admission number to get student details')
            get_class_student_detail(class_sec , admn_no)

        elif command == 'C' or command == 'c' :
            class_sec = input('enter class and section to get student details :')

            query = f'use classes_{batch_year};'
            dbconn.execute(query)

            query = f'SELECT * FROM class_{str(class_sec)}_{str(batch_year)};'
            dbconn.execute(query)
            DATA = dbconn.fetchall()

            for y in DATA :
                get_class_student_detail(class_sec , y[0])

        elif command == 'W' or command == 'w' :
            query = f'use classes_{batch_year};'
            dbconn.execute(query)

            class_sec = input('enter class and section to add student details :')

            query = f'desc class_{str(class_sec)}_{str(batch_year)};'
            dbconn.execute(query)
            data = []
            data_record = []

            for y in dbconn :
                data.append(y[0])
                x = input(f'{y[0] } ;- ')
                data_record.append(x)

            value = ''
```

```
            for xy in data_record :
                if value == '' :
                    value = xy
                else :
                    value = str(value) +"," '+ str(xy) + '"'

            query = f'insert into class_{str(class_sec)}_{str(batch_year)} values
({value});'
            dbconn.execute(query)
            database.commit()
            print('record added succesfully')

        elif command == 'M' or command == 'm' :
            query = f'use classes_{batch_year};'
            dbconn.execute(query)

            class_sec = input('enter class and section to get student details :')
            adm_no = input("enter admission number of student to modify changes :")

            query = f"SELECT * FROM class_{str(class_sec)}_{str(batch_year)} where
ADMISSION_NO = {adm_no} ;"
            dbconn.execute(query)
            for y in dbconn :
                data_record = y

            data = []
            query = f'desc class_{str(class_sec)}_{str(batch_year)};'
            dbconn.execute(query)
            for y in dbconn :
                data.append(y[0])

            edit_dict = dict()
            for edit in range(len(data)) :
                if data[edit] != 'ADMISSION_NO' :
                    entry = input(f"{data[edit]} = {data_record[edit]}   :")
                    edit_dict[data[edit]] = str(entry)

            value = ''
            for xy in edit_dict :
                if (value == '' or value == ' ') == True and  edit_dict[xy] != '' :
                    value =  xy + ' = " ' + edit_dict[xy] + ' " '
                elif edit_dict[xy] != '' :
                    value = str(value) + ', ' + xy + '  = " ' + edit_dict[xy] + ' " '

            query = f'''UPDATE class_{str(class_sec)}_{str(batch_year)}
            SET {value}
            WHERE ADMISSION_NO = {adm_no};'''
            dbconn.execute(query)
            database.commit()
            print('changes done successfully')

        elif command == 'R' or command == 'r' :
            class_sec = input('enter class and section to get student details :')
            adm_no = input("enter admission number of student to remove from class :")
```

```
            query = f'DELETE FROM class_{str(class_sec)}_{str(batch_year)} WHERE
ADMISSION_NO = {adm_no} ;'
            dbconn.execute(query)
            database.commit()
            print('student removed from class successfully')

    #============================#  RESULTS #============================#
    if command == 'R'or command == 'r' :
        try:
            query = f'use classes_{batch_year};'
            dbconn.execute(query)
        except:
            confirm = input('''THIS CLASS BATCH DOES NOT HAVE RESULT
            press "Y" OR "y" to create a new else press any key ''')
            if confirm == 'Y' or confirm == 'y' :
                query = f'create database results_{batch_year};'
                dbconn.execute(query)

        def get_class_student_detail(class_sec , result_type , admn_no):
            query = f"USE SCHOOL ;"
            dbconn.execute(query)

            query = f"SELECT * FROM student where ADMISSION_NO = {admn_no} ;"
            dbconn.execute(query)
            data_record = ''
            for y in dbconn :
                data_record = y

            data = []
            query = f'desc student;'
            dbconn.execute(query)
            for y in dbconn :
                data.append(y[0])

            edit_dict = dict()
            for edit in range(len(data_record)) :
                if data[edit] != 'ADMISSION_NO' :
                    edit_dict[data[edit]] = data_record[edit]

            query = f'use results_{batch_year};'
            dbconn.execute(query)

            query = f'''select *
from  result_{str(class_sec)}_{str(result_type)}_{str(batch_year)} where ADMISSION_NO =
{admn_no} ; '''
            dbconn.execute(query)
            data_record = ''
            for y in dbconn :
                data_record = y
            if data_record == '' :
                print(f'{admn_no} not present in result of {class_sec} in {result_type}')

            data = []
```

```
            query = f'desc result_{str(class_sec)}_{str(result_type)}_{str(batch_year)};'
            dbconn.execute(query)
            for y in dbconn :
                data.append(y[0])

            for edit in range(len(data_record)) :
                edit_dict[(data[edit])] = data_record[edit]

            print(edit_dict)

        announcement = f'''
        press 'S' or 's' to show results in batch {batch_year} :
        press 'A' OR 'a' to add a new result class :
        press 'D' OR 'd' to get student details in result :
        press 'C' OR 'c' to get result of students :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student result :
        press 'R' or 'r' to remove a student :
        press 'G' or 'g' to generate pdf :
        press 'E' or 'e' to email result :  \n '''

        command = input(announcement)
        if command ==  'S' or command == 's' :
            query = f'use results_{batch_year};'
            dbconn.execute(query)

            query = f'show tables;'
            dbconn.execute(query)
            for y in dbconn :
              print(y)

        elif command == 'A' or command == 'a' :
            query = f'use results_{batch_year};'
            dbconn.execute(query)

            class_sec = input('enter class and section to create a new record :')
            class_subject = int(input(f'enter subjects allocated in class {class_sec} :'))
            result_type = input('enter result type :')

            X = ''
            for i in range(class_subject):
                X = X +f', SUBJECT_{i+1} VARCHAR(20) , MARKS_{i+1} INT '
            query = f'''create table
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)} ( ADMISSION_NO BIGINT NOT
NULL PRIMARY KEY
             , MAX_MARKS INT  NOT NULL , EXAM_TYPE VARCHAR(20) NOT NULL {X}  ) ; '''
            dbconn.execute(query)
            database.commit()

        elif command == 'D' or command == 'd' :
            class_sec = input('enter class and section to get student details :')
            admn_no = input('enter admission number to get student details')
            result_type = input('enter exam type :')
```

```
            get_class_student_detail(class_sec , result_type , admn_no)

        elif command == 'C' or command == 'c' :
            class_sec = input('enter class and section to get student details :')
            result_type = input('enter exam type :')

            query = f'use results_{batch_year};'
            dbconn.execute(query)
            DATA = dbconn.fetchall()

            query = f'SELECT * FROM
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)};'
            dbconn.execute(query)
            for y in DATA :
                get_class_student_detail(class_sec , result_type , y[0])

        elif command == 'W' or command == 'w' :
            query = f'use results_{batch_year};'
            dbconn.execute(query)

            class_sec = input('enter class and section to add student details :')
            result_type = input('enter exam type :')

            query = f'desc result_{str(class_sec)}_{str(result_type)}_{str(batch_year)};'
            dbconn.execute(query)
            data = []
            data_record = []

            for y in dbconn :
                data.append(y[0])
                x = input(f'{y[0] } ;- ')
                data_record.append(x)

            value = ''
            for xy in data_record :
                if value == '' :
                    value = xy
                else :
                    value = str(value) +"," '+ str(xy) + '"'

            query = f'insert into
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)} values ({value});'
            dbconn.execute(query)
            database.commit()
            print('record added succesfully')

        elif command == 'M' or command == 'm' :
            query = f'use results_{batch_year};'
            dbconn.execute(query)

            class_sec = input('enter class and section to get student details :')
            adm_no = input("enter admission number of student to modify changes :")
            result_type = input('enter exam type :')
```

```
            query = f"SELECT * FROM
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)} where ADMISSION_NO = {adm_no}
;"
            dbconn.execute(query)
            for y in dbconn :
                data_record = y

            data = []
            query = f'desc result_{str(class_sec)}_{str(result_type)}_{str(batch_year)};'
            dbconn.execute(query)
            for y in dbconn :
                data.append(y[0])

            edit_dict = dict()
            for edit in range(len(data)) :
                if data[edit] != 'ADMISSION_NO' :
                    exit = input(f"{data[edit]} = {data_record[edit]}   :")
                    if exit != '' or exit != ' ' :
                        edit_dict[data[edit]] = exit

            value = ''
            for xy in edit_dict :
                if (value == '' or value == ' ') == True and  edit_dict[xy] != '' :
                    value =  xy + '  = ' + edit_dict[xy]
                elif edit_dict[xy] != ''  :
                    value = str(value) + ', ' + xy + '  = " ' + edit_dict[xy] + ' " '

            query = f'''UPDATE
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)}
            SET {value}
            WHERE ADMISSION_NO = {adm_no} ; '''
            dbconn.execute(query)
            database.commit()
            print('changes done successfully')

        elif command == 'R' or command == 'r' :
            query = f'use results_{batch_year};'
            dbconn.execute(query)

            class_sec = input('enter class and section to get student details :')
            adm_no = input("enter admission number of student to remove from class :")
            result_type = input('enter exam type :')

            query = f'DELETE FROM
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)} WHERE ADMISSION_NO = {adm_no}
;'
            dbconn.execute(query)
            database.commit()
            print('student removed from result successfully')

        elif command == 'G' or command == 'g'  :
            class_sec = input('enter class and section to get student details :')
            result_type = input('enter exam type :')
```

```
            query = f'use results_{batch_year};'
            dbconn.execute(query)

            #=============#  GENERATE PERIODIC TEST REPORT CARD  #====================#
            def generate_report_card_periodic( edit_dict , edit_list , class_teacher ,
class_sec ) :
                admission_no  = edit_dict['ADMISSION_NO']
                name          = edit_dict['NAME']
                date_of_birth = edit_dict['DOB']
                school_details = ['J.D. TYTLER SCHOOL ' , 'NEW RAJINDER NAGAR , R BLOCK ,
NEW DELHI 110060' , '85032' , 'Ms NEENA ANDREW' , 'divyamsethi1804@gmail.com' ]

                subjects = dict()
                for xn in range(14 , len(edit_list) , 2) :
                    subjects[edit_list[xn][1]] = edit_list[xn+1][1]

                school   = school_details[0]
                exam     = edit_dict['EXAM_TYPE']
                standard = class_sec
                maximum_marks = edit_dict['MAX_MARKS']
                total_marks = 0

                pdf = FPDF('P', 'mm', 'A4')
                pdf.add_page()
                pdf.set_font('Arial','B', size = 12)

                try :
                    pdf.image('C:/Users/divya/OneDrive/Desktop/PROJECT/JDTS LOGO.jpeg' , x
= 10, y = 10, h = 50  )
                except :
                    pass

                pdf.set_font('Arial', 'B' , size = 25 )
                pdf.ln(10)
                pdf.set_text_color(34,139,34)
                pdf.cell(235,10,f'{school}' , align = 'C' )
                pdf.ln(15)

                pdf.set_text_color(255,0,0)
                pdf.cell(235,10,f'{exam}' , align = 'C' )
                pdf.ln(30)

                pdf.set_text_color(0,0,0)
                pdf.set_font('Arial', 'B' , size = 15)

                pdf.cell( 18 , 10 ,f' NAME : { name }'                    , align = 'L' )
                pdf.cell(150 , 10 ,f' ADMISSION NO : { admission_no } '   , align = 'R')
                pdf.ln(15)

                pdf.cell(18 , 10 ,f' CLASS : { standard }'       , align = 'L' )
                pdf.cell(150 , 10 ,f' DOB : { date_of_birth } '   , align = 'R')
                pdf.ln(20)

                pdf.set_font('Arial', 'B' , size = 12)
```

```
            pdf.cell(50 , 10 ,'SUBJECTS'                  , 1 , align = 'L' )
            pdf.cell(45 , 10 ,f'MARKS ( {maximum_marks} )'  , 1 , align = 'C' )
            pdf.cell(45 , 10 ,f'PERCENTAGE'               , 1 , align = 'C' )
            pdf.cell(45 , 10 ,'GRADES'                    , 1 , align = 'C' )
            pdf.ln(10)

            to_percent = 100 / int(maximum_marks)
            for key in subjects :
                try :
                    percentage = round(float(subjects[key]) * float(to_percent) , 2 )

                    if percentage > 90 :
                        grade = 'A1'
                    elif 90 >= percentage > 80 :
                        grade = 'A2'
                    elif 80 >= percentage > 70 :
                        grade = 'B1'
                    elif 70 >= percentage > 60 :
                        grade = 'B2'
                    elif 60 >= percentage > 50 :
                        grade = 'C1'
                    elif 50 >= percentage > 40 :
                        grade = 'C2'
                    elif 40 >= percentage >= 33 :
                        grade = 'D'
                    else :
                        grade = 'E'
                except :
                    grade = 'AB'
                    percentage = '-'
                pdf.cell(50 ,10 ,f'{key}'           , 1 , align = 'L' )
                pdf.cell(45 ,10 ,f'{subjects[key]}' , 1 , align = 'C' )
                pdf.cell(45 ,10 ,f'{percentage} %'  , 1 , align = 'C' )
                pdf.cell(45 ,10 ,f'{grade}'         , 1 , align = 'C' )
                pdf.ln(10)

                try :
                    total_marks += float(subjects[key])
                except :
                    pass
            marks_percentage = round(( total_marks / len(subjects) ) * to_percent , 2)

            pdf.cell(50 , 10 ,'GRAND TOTAL'           , 1 , align = 'L' )
            pdf.cell(45 , 10 ,f'{total_marks}'        , 1 , align = 'C' )
            pdf.cell(45 , 10 ,f'{marks_percentage} %' , 1 , align = 'C' )
            pdf.cell(45 , 10 ,''    , 1 )
            pdf.ln(20)

            pdf.set_font('Arial', 'B' , size = 10 )

            pdf.cell(30 , 5 ,'PERCENTAGE'  , 1 )
            pdf.cell(30 , 5 ,'MARKS'       , 1 )
            pdf.cell(30 , 5 ,'GRADE'       , 1 )
```

```
            pdf.ln(5)

            pdf.cell(30 , 5 ,'91% - 100%'  , 1 )
            pdf.cell(30 , 5 ,' 23 to 25 '  , 1 )
            pdf.cell(30 , 5 ,'  A1  '      , 1 )
            pdf.ln(5)

            pdf.cell(30 , 5 ,'81% - 90%'   , 1 )
            pdf.cell(30 , 5 ,' 21 to 22 '  , 1 )
            pdf.cell(30 , 5 ,'  A2  '      , 1 )
            pdf.ln(5)

            pdf.cell(30 , 5 ,'71% - 80%'   , 1 )
            pdf.cell(30 , 5 ,' 18 to 20 '  , 1 )
            pdf.cell(30 , 5 ,'  B1  '      , 1 )
            pdf.ln(5)

            pdf.cell(30 , 5 ,'61% - 70%'   , 1 )
            pdf.cell(30 , 5 ,' 16 to 17 '  , 1 )
            pdf.cell(30 , 5 ,'  B2  '      , 1 )
            pdf.ln(5)

            pdf.cell(30 , 5 ,'51% - 60%'   , 1 )
            pdf.cell(30 , 5 ,' 13 to 15 '  , 1 )
            pdf.cell(30 , 5 ,'  C1  '      , 1 )
            pdf.ln(5)

            pdf.cell(30 , 5 ,'41% - 50%'   , 1 )
            pdf.cell(30 , 5 ,' 11 to 12 '  , 1 )
            pdf.cell(30 , 5 ,'  C2  '      , 1 )
            pdf.ln(5)

            pdf.cell(30 , 5 ,'33% - 40%'   , 1 )
            pdf.cell(30 , 5 ,' 8 to 10 '   , 1 )
            pdf.cell(30 , 5 ,'  D  '       , 1 )
            pdf.ln(5)

            pdf.cell(30 , 5 ,'Below 33%'   , 1 )
            pdf.cell(30 , 5 ,' 0 to 7 '    , 1 )
            pdf.cell(30 , 5 ,'  E  '       , 1 )
            pdf.ln(15)

            pdf.set_font('Arial', 'B' , size = 12)

            pdf.ln(10)
            pdf.cell(20 , 10 ,f'{school_details[3]} '        , align = 'L' )
            pdf.cell(150 , 10 ,f'{ class_teacher } '      , align = 'R' )

            pdf.ln(10)
            pdf.cell(25 , 10 ,f'PRINCIPAL '       , align = 'C' )
            pdf.cell(150 , 10 ,f'CLASS TEACHER '   , align = 'R' )

            if os.path.isdir(f"D:/report card") == False :
                os.mkdir(f"D:/report card")
```

```
                if os.path.isdir(f"D:/report card/{exam} {standard}") == False :
                    os.mkdir(f"D:/report card/{exam} {standard}")
                pdf.output(f'D:/report card/{exam} {standard}/{name} {admission_no}
{standard} {exam}.pdf' , 'F')

            query = f'SELECT * FROM
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)};'
            dbconn.execute(query)
            row = dbconn.fetchall()
            for y in row :
                admn_no = y[0]
                query = f"USE SCHOOL ;"
                dbconn.execute(query)

                query = f"SELECT * FROM student where ADMISSION_NO = {admn_no} ;"
                dbconn.execute(query)
                data_record = ''
                row_1 = dbconn.fetchall()
                for y in row_1 :
                    data_record = y

                data = []
                query = f'desc student;'
                dbconn.execute(query)
                for y in dbconn :
                    data.append(y[0])

                edit_dict = dict()
                for edit in range(len(data_record)) :
                    if data[edit] != 'ADMISSION_NO' :
                        edit_dict[data[edit]] = data_record[edit]

                query = f'use results_{batch_year};'
                dbconn.execute(query)

                query = f'''select *
from  result_{str(class_sec)}_{str(result_type)}_{str(batch_year)} where ADMISSION_NO =
{admn_no} ; '''
                dbconn.execute(query)
                row_2 = dbconn.fetchall()
                data_record = ''
                for y in row_2 :
                    data_record = y
                if data_record == '' :
                    print(f'{admn_no} not present in result of {class_sec} in
{result_type}')

                data = []
                query = f'desc
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)};'
                dbconn.execute(query)
                for y in dbconn :
                    data.append(y[0])
```

```
                for edit in range(len(data_record)) :
                    edit_dict[(data[edit])] = data_record[edit]

                edit_list = [ (k ,v) for  k , v in edit_dict.items()]

                query = f'use classes_{batch_year};'
                dbconn.execute(query)

                query = f'''select * from  class_{str(class_sec)}_{str(batch_year)} where
ADMISSION_NO = {admn_no} ; '''
                dbconn.execute(query)
                row_3 = dbconn.fetchall()
                data_record = ''
                for y in row_3 :
                    data_record = y
                if data_record == '' :
                    print(f'{admn_no} not present in class {class_sec}')
                data = []
                query = f'desc class_{str(class_sec)}_{str(batch_year)};'
                dbconn.execute(query)
                for y in dbconn :
                    data.append(y[0])

                edit_dict_1 = dict()
                for edit in range(len(data_record)) :
                    edit_dict_1[(data[edit])] = data_record[edit]
                class_teacher = edit_dict_1['CLASS_TEACHER']

                generate_report_card_periodic( edit_dict , edit_list , class_teacher ,
class_sec )

            print(' report cards are generated pls check "D:\ report card"')

        elif command == 'E' or command == 'e' :
            class_sec = input('enter class and section to get student details :')
            result_type = input('enter exam type :')

            school_details = ['J.D. TYTLER SCHOOL ' , 'NEW RAJINDER NAGAR , R BLOCK , NEW
DELHI 110060' , '85032' , 'Ms NEENA ANDREW' , 'divyamsethi1804@gmail.com' ]
            fromaddr = school_details[4]

            student_dire  = "C:/Users/divya/OneDrive/Desktop/11 - c email.csv"
            purpose = f'RESULT {result_type} {class_sec}'

            def gmail_content( fromaddr , toaddr , password , admn_no , name , purpose ,
class_sec , exam ) :
                msg = MIMEMultipart()
                msg['From'] = fromaddr
                msg['To'] = toaddr
                msg['Subject'] = f'{name}   {purpose}'

                # string to store the body of the mail
                body = f'{name}   {purpose}'
                msg.attach(MIMEText(body, 'plain'))
```

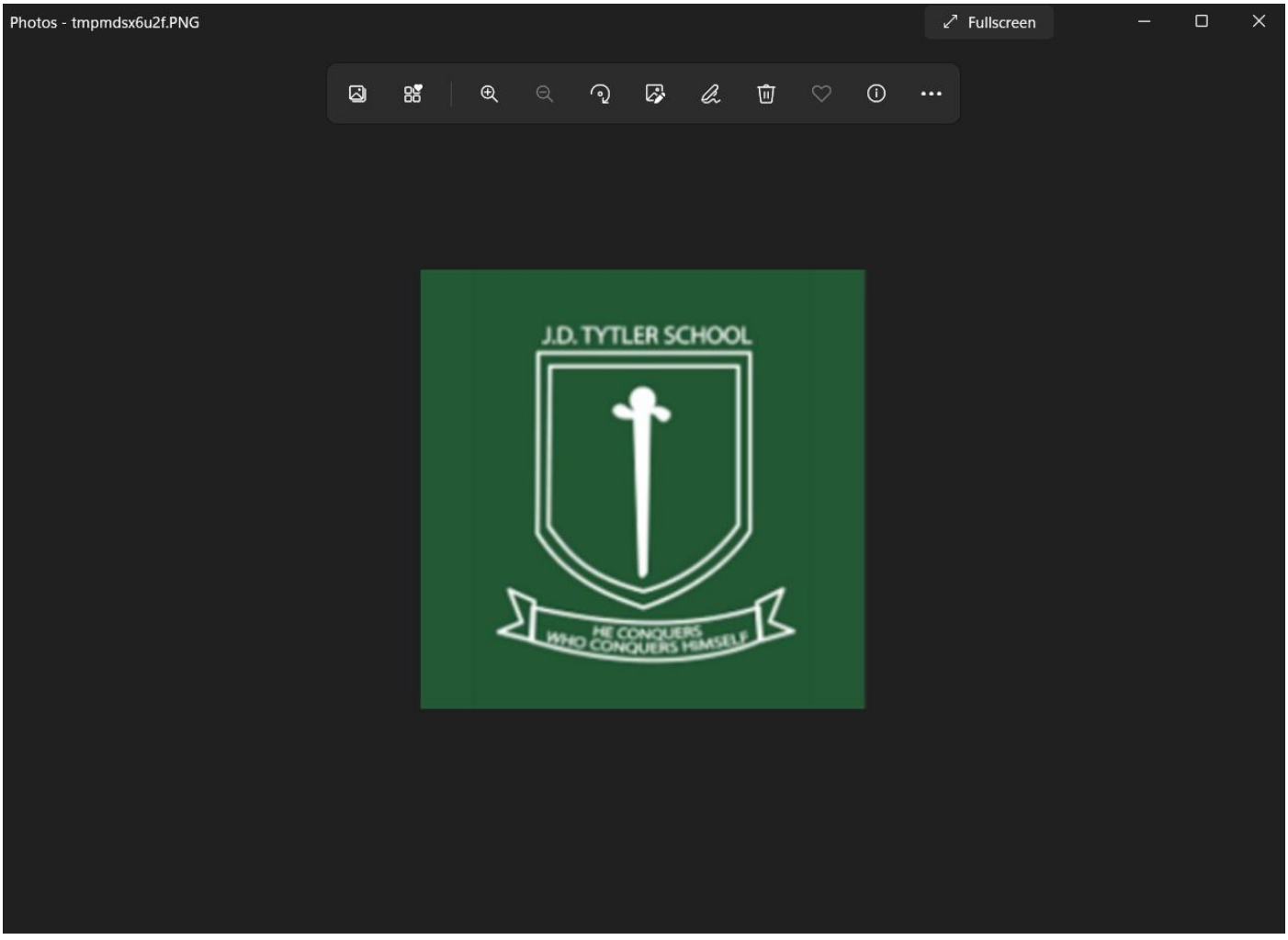
```
                filename = f"{name}.pdf"
                attachment = open( f'D:/report card/{exam} {class_sec}/{name} {admn_no}
{class_sec} {exam}.pdf' , "rb")

                p = MIMEBase('application', 'octet-stream')
                p.set_payload((attachment).read())
                encoders.encode_base64(p)
                p.add_header('Content-Disposition', "attachment; filename= %s" % filename)
                msg.attach(p)

                s = smtplib.SMTP('smtp.gmail.com', 587)
                s.starttls()
                s.login(fromaddr, password )
                text = msg.as_string()
                s.sendmail(fromaddr, toaddr, text)
                s.quit()

            def algorithum(password):
                query = f'use results_{batch_year};'
                dbconn.execute(query)

                query = f'SELECT * FROM
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)};'
                dbconn.execute(query)
                row = dbconn.fetchall()
                for y in row :
                    admn_no = y[0]
                    query = f"USE SCHOOL ;"
                    dbconn.execute(query)

                    query = f"SELECT * FROM student where ADMISSION_NO = {admn_no} ;"
                    dbconn.execute(query)
                    data_record = ''
                    for y in dbconn :
                        data_record = y

                    data = []
                    query = f'desc student;'
                    dbconn.execute(query)
                    for y in dbconn :
                        data.append(y[0])

                    edit_dict = dict()
                    for edit in range(len(data_record)) :
                        if data[edit] != 'ADMISSION_NO' :
                            edit_dict[data[edit]] = data_record[edit]

                    query = f'use results_{batch_year};'
                    dbconn.execute(query)

                    query = f'''select *
from  result_{str(class_sec)}_{str(result_type)}_{str(batch_year)} where ADMISSION_NO =
{admn_no} ; '''
```

```
                    dbconn.execute(query)
                    data_record = ''
                    for y in dbconn :
                        data_record = y
                    if data_record == '' :
                        print(f'{admn_no} not present in result of {class_sec} in
{result_type}')

                    data = []
                    query = f'desc
result_{str(class_sec)}_{str(result_type)}_{str(batch_year)};'
                    dbconn.execute(query)
                    for y in dbconn :
                        data.append(y[0])

                    for edit in range(len(data_record)) :
                        edit_dict[(data[edit])] = data_record[edit]

                    toaddr = edit_dict['EMAIL']
                    admn_no = edit_dict['ADMISSION_NO']
                    name = edit_dict['NAME']
                    exam     = edit_dict['EXAM_TYPE']

                    try :
                        gmail_content( fromaddr , toaddr , password , admn_no , name ,
purpose , class_sec , exam )
                        print(f'mailed sent succesfully to {name} admn no { admn_no}' )
                    except:
                        print(f'failed to send to {name} admn no { admn_no}' )

            root=tk.Tk()
            root.geometry("400x400")

            passw_var=tk.StringVar()

            def submit():
                passw_label = tk.Label(root, text = 'Password entered', font =
('calibre',15,'bold')).grid(row=3,column=1)
                algorithum(passw_var.get())

            passw_label = tk.Label(root, text = 'Password', font = ('calibre',15,'bold'))
            passw_entry=tk.Entry(root, textvariable = passw_var, font =
('calibre',15,'normal'), show = '*')

            sub_btn=tk.Button(root,text = 'Submit', font = ('calibre',15,'normal'),
command = submit)

            passw_label.grid(row=1,column=0)
            passw_entry.grid(row=1,column=1)
            sub_btn.grid(row=2,column=1)

            root.mainloop()
```

OUTPUT

1) VIEWING SCHOOL DETAILS : -

```
enter 'S' OR 's' to get information about school :
enter 'A' OR 'a' to access student details :
enter 'O' or 'o' to get other help :
S
J.D. TYTLER SCHOOL
NEW RAJINDER NAGAR , R BLOCK , NEW DELHI 110060
85032
Ms NEENA ANDREW
jdtytlerschoola@gmail.com
PS C:\Users\divya>
```



2) VIEWING / EDITING STUDENT DETAILS : -

```
enter 'S' OR 's' to get information about school :
enter 'A' OR 'a' to access student details :
enter 'O' or 'o' to get other help :
A

    press 'N' or 'n' to show non-payment students :
    press 'S' OR 's' to get student details by admn_no :
    press 'P' OR 'p' to get student details by phone :
    press 'A' or 'a' to add new student :
    press 'M' or 'm' to modify student details :
```

A) VIEWING NON PAYMENT STUDENTS DETAILS : -

```
    press 'N' or 'n' to show non-payment students :
    press 'S' OR 's' to get student details by admn_no :
    press 'P' OR 'p' to get student details by phone :
    press 'A' or 'a' to add new student :
    press 'M' or 'm' to modify student details :
 n
(3, ' YASH MAINI', ' XII_C', ' KAPIL MAINI', ' MEETU MAINI', datetime.date(2022, 3, 23), 200, 5000, 8920028757, '
 mainiyash2@gmail.com', ' T 1063 PUNJABI BASTI KAROL BAGH DELHI', datetime.date(2004, 6, 30))
```

B) VIEW STUDENT DETAILS BY ADMISSION NUMBER : -

```
    press 'N' or 'n' to show non-payment students :
    press 'S' OR 's' to get student details by admn_no :
    press 'P' OR 'p' to get student details by phone :
    press 'A' or 'a' to add new student :
    press 'M' or 'm' to modify student details :
 S
enter admission number 3
(3, ' YASH MAINI', ' XII_C', ' KAPIL MAINI', ' MEETU MAINI', datetime.date(2022, 3, 23), 200, 0, 8920028757, ' ma
iniyash2@gmail.com', ' T 1063 PUNJABI BASTI KAROL BAGH DELHI', datetime.date(2004, 6, 30))
```

C) VIEW STUDENT DETAILS BY PHONE NUMBER : -

```
    press 'N' or 'n' to show non-payment students :
    press 'S' OR 's' to get student details by admn_no :
    press 'P' OR 'p' to get student details by phone :
    press 'A' or 'a' to add new student :
    press 'M' or 'm' to modify student details :
 P
enter phone number 8920028757
(3, ' YASH MAINI', ' XII_C', ' KAPIL MAINI', ' MEETU MAINI', datetime.date(2022, 3, 23), 200, 5000, 8920028757, '
 mainiyash2@gmail.com', ' T 1063 PUNJABI BASTI KAROL BAGH DELHI', datetime.date(2004, 6, 30))
```

D) TO ADD A NEW STUDENT : -

```
    press 'N' or 'n' to show non-payment students :
    press 'S' OR 's' to get student details by admn_no :
    press 'P' OR 'p' to get student details by phone :
    press 'A' or 'a' to add new student :
    press 'M' or 'm' to modify student details :
 A
4
NAME ;- DIVYAM SETHI
class_ ;- XII_C
FATHER ;- SANJAY SETHI
MOTHER ;- RACHNA SETHI
FEES ;- 5000
PHONE ;- 9911713714
EMAIL ;- divyamsethi1804@gmail.com
ADDRESS ;- OUTRAM LINES , MUKHERJEE NAGAR DELHI
DOB ;- 2004-04-18
record added succesfully
```

E) TO MODIFY STUDENTS DETAILS : -

```
    press 'N' or 'n' to show non-payment students :
    press 'S' OR 's' to get student details by admn_no :
    press 'P' OR 'p' to get student details by phone :
    press 'A' or 'a' to add new student :
    press 'M' or 'm' to modify student details :
 m
enter admission number of student to modify changes :3
NAME =  YASH MAINI   :
class_ =  XII_C   :
FATHER =  KAPIL MAINI   :
MOTHER =  MEETU MAINI   :
ADM_DATE = 2022-03-23   :
FEES = 200   :
BALANCE = 0   :5000
PHONE = 8920028757   :
EMAIL =  mainiyash2@gmail.com   :
ADDRESS =  T 1063 PUNJABI BASTI KAROL BAGH DELHI   :
DOB = 2004-06-30   :
changes done successfully
```

3) VIEWING CLASS DETAILS AND RESULT : -

```
enter 'S' OR 's' to get information about school :
enter 'A' OR 'a' to access student details :
enter 'O' or 'o' to get other help :
O
ENTER BATCH YEAR :-2022

    press 'C' OR 'c' to access class details :
    press 'R' OR 'r' to access report card details :
    press any key to exit :
```

A) TO GET CLASS DETAILS : -

```
   press 'C' OR 'c' to access class details :
   press 'R' OR 'r' to access report card details :
   press any key to exit :
C

       press 'S' or 's' to show classes in batch 2022 :
       press 'A' OR 'a' to add a new class :
       press 'D' OR 'd' to get student details of a class :
       press 'C' OR 'c' to get students of a class :
       press 'W' or 'w' to add new student :
       press 'M' or 'm' to modify student details :
       press 'R' or 'r' to remove a student :
```

- **TO GET CLASSES IN GIVEN BATCH : -**

```
       press 'S' or 's' to show classes in batch 2022 :
       press 'A' OR 'a' to add a new class :
       press 'D' OR 'd' to get student details of a class :
       press 'C' OR 'c' to get students of a class :
       press 'W' or 'w' to add new student :
       press 'M' or 'm' to modify student details :
       press 'R' or 'r' to remove a student :
 S
('class_xii_c_2022',)
```

- **TO ADD A NEW CLASS IN GIVEN BATCH : -**

```
        press 'S' or 's' to show classes in batch 2022 :
        press 'A' OR 'a' to add a new class :
        press 'D' OR 'd' to get student details of a class :
        press 'C' OR 'c' to get students of a class :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student details :
        press 'R' or 'r' to remove a student :
 A
enter class and section to create a new record :XI_C
enter subjects allocated in class XI_C :5
```

- **TO GET STUDENT DETAILS IN GIVEN CLASS BATCH : -**

```
        press 'S' or 's' to show classes in batch 2022 :
        press 'A' OR 'a' to add a new class :
        press 'D' OR 'd' to get student details of a class :
        press 'C' OR 'c' to get students of a class :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student details :
        press 'R' or 'r' to remove a student :
 D
enter class and section to get student details :XII_C
enter admission number to get student details3
{'NAME': ' YASH MAINI', 'class_': ' XII_C', 'FATHER': ' KAPIL MAINI', 'MOTHER': ' MEETU MAINI', 'ADM_DATE': dateti
me.date(2022, 3, 23), 'FEES': 200, 'BALANCE': 5000, 'PHONE': 8920028757, 'EMAIL': ' mainiyash2@gmail.com', 'ADDRES
S': ' T 1063 PUNJABI BASTI KAROL BAGH DELHI', 'DOB': datetime.date(2004, 6, 30), 'ADMISSION_NO': 3, 'CLASS_TEACHER
': ' DR AZHAR ASLAM KHAN', 'SUBJECT_1': ' ENGLISH', 'SUBJECT_2': ' PHYSICS', 'SUBJECT_3': ' CHEMISTRY', 'SUBJECT_4
': ' MATHS', 'SUBJECT_5': ' BIOLOGY'}
PS C:\Users\divya>
```

- **TO GET STUDENTS DETAIL IN GIVEN CLASS BATCH : -**

```
        press 'S' or 's' to show classes in batch 2022 :
        press 'A' OR 'a' to add a new class :
        press 'D' OR 'd' to get student details of a class :
        press 'C' OR 'c' to get students of a class :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student details :
        press 'R' or 'r' to remove a student :
 C
enter class and section to get student details :XII_C
{'NAME': ' DIVYAM SETHI ', 'class_': 'xii', 'FATHER': ' SANJAY SETHI ', 'MOTHER': ' RACHNA SETHI ', 'ADM_DATE': No
ne, 'FEES': 5000, 'BALANCE': 0, 'PHONE': 99393393, 'EMAIL': ' divyamsethi1804@gmail.com ', 'ADDRESS': 'outdnjf', '
DOB': None, 'ADMISSION_NO': 1, 'CLASS_TEACHER': ' DR AZHAR ASLAM KHAN', 'SUBJECT_1': ' ENGLISH ', 'SUBJECT_2': ' p
hysics', 'SUBJECT_3': ' chemistry', 'SUBJECT_4': ' maths', 'SUBJECT_5': ' computer '}
{'NAME': ' YASH MAINI', 'class_': ' XII_C', 'FATHER': ' KAPIL MAINI', 'MOTHER': ' MEETU MAINI', 'ADM_DATE': dateti
me.date(2022, 3, 23), 'FEES': 200, 'BALANCE': 5000, 'PHONE': 8920028757, 'EMAIL': ' mainiyash2@gmail.com', 'ADDRES
S': ' T 1063 PUNJABI BASTI KAROL BAGH DELHI', 'DOB': datetime.date(2004, 6, 30), 'ADMISSION_NO': 3, 'CLASS_TEACHER
': ' DR AZHAR ASLAM KHAN', 'SUBJECT_1': ' ENGLISH', 'SUBJECT_2': ' PHYSICS', 'SUBJECT_3': ' CHEMISTRY', 'SUBJECT_4
': ' MATHS', 'SUBJECT_5': ' BIOLOGY'}
```

- **TO ADD STUDENT IN GIVEN CLASS BATCH : -**

```
        press 'S' or 's' to show classes in batch 2022 :
        press 'A' OR 'a' to add a new class :
        press 'D' OR 'd' to get student details of a class :
        press 'C' OR 'c' to get students of a class :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student details :
        press 'R' or 'r' to remove a student :
 W
enter class and section to add student details :XI_C
ADMISSION_NO ;- 4
CLASS_TEACHER ;- MRS RACHNA SETHI
BEHAVIOUR ;-
SUBJECT_1 ;- ENGLISH
SUBJECT_2 ;- PHYSICS
SUBJECT_3 ;- CHEMISTRY
SUBJECT_4 ;- MATHS
SUBJECT_5 ;- CS
record added succesfully
```

- **TO MODIFY STUDENT DETAILS IN GIVEN CLASS BATCH : =**

```
        press 'S' or 's' to show classes in batch 2022 :
        press 'A' OR 'a' to add a new class :
        press 'D' OR 'd' to get student details of a class :
        press 'C' OR 'c' to get students of a class :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student details :
        press 'R' or 'r' to remove a student :
 M
enter class and section to get student details :XI_C
enter admission number of student to modify changes :4
CLASS_TEACHER =  MRS RACHNA SETHI    :
BEHAVIOUR =     :INTELLIGENT
SUBJECT_1 =  ENGLISH    :
SUBJECT_2 =  PHYSICS    :
SUBJECT_3 =  CHEMISTRY    :
SUBJECT_4 =  MATHS    :
SUBJECT_5 =  COMPUTER SCIENCE     :
changes done successfully
```

- **TO REMOVE STUDENT IN GIVEN CLASS BATCH : -**

```
        press 'S' or 's' to show classes in batch 2022 :
        press 'A' OR 'a' to add a new class :
        press 'D' OR 'd' to get student details of a class :
        press 'C' OR 'c' to get students of a class :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student details :
        press 'R' or 'r' to remove a student :
 R
enter class and section to get student details :XI_C
enter admission number of student to remove from class :4
student removed from class successfully
```

A) TO ACCESS RESULT : -

```
    press 'C' OR 'c' to access class details :
    press 'R' OR 'r' to access report card details :
    press any key to exit :
R

        press 'S' or 's' to show results in batch 2022 :
        press 'A' OR 'a' to add a new result class :
        press 'D' OR 'd' to get student details in result :
        press 'C' OR 'c' to get result of students :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student result :
        press 'R' or 'r' to remove a student :
        press 'G' or 'g' to generate pdf :
        press 'E' or 'e' to email result :
```

- **TO GET RESULTS IN GIVEN BATCH : -**

```
        press 'S' or 's' to show results in batch 2022 :
        press 'A' OR 'a' to add a new result class :
        press 'D' OR 'd' to get student details in result :
        press 'C' OR 'c' to get result of students :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student result :
        press 'R' or 'r' to remove a student :
        press 'G' or 'g' to generate pdf :
        press 'E' or 'e' to email result :
 S
('result_xii_c_pt_1_2022',)
```

- **TO ADD NEW RESULT RECORD IN GIVEN BATCH : -**

```
        press 'S' or 's' to show results in batch 2022 :
        press 'A' OR 'a' to add a new result class :
        press 'D' OR 'd' to get student details in result :
        press 'C' OR 'c' to get result of students :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student result :
        press 'R' or 'r' to remove a student :
        press 'G' or 'g' to generate pdf :
        press 'E' or 'e' to email result :
 A
enter class and section to create a new record :XI_C
enter subjects allocated in class XI_C :5
enter result type :FINAL
```

- **TO GET STUDENT RESULT IN GIVEN BATCH : -**

```
        press 'S' or 's' to show results in batch 2022 :
        press 'A' OR 'a' to add a new result class :
        press 'D' OR 'd' to get student details in result :
        press 'C' OR 'c' to get result of students :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student result :
        press 'R' or 'r' to remove a student :
        press 'G' or 'g' to generate pdf :
        press 'E' or 'e' to email result :
 D
enter class and section to get student details :XII_C
enter admission number to get student details1
enter exam type :PT_1
{'NAME': ' DIVYAM SETHI ', 'class_': 'xii', 'FATHER': ' SANJAY SETHI ', 'MOTHER': ' RACHNA SETHI ', 'ADM_DATE': No
ne, 'FEES': 5000, 'BALANCE': 0, 'PHONE': 99393393, 'EMAIL': ' divyamsethi1804@gmail.com ', 'ADDRESS': 'outdnjf', '
DOB': None, 'ADMISSION_NO': 1, 'MAX_MARKS': 50, 'EXAM_TYPE': ' PERIODIC TEST', 'SUBJECT_1': ' ENG', 'MARKS_1': 50,
 'SUBJECT_2': ' PHY', 'MARKS_2': 50, 'SUBJECT_3': ' CHEM', 'MARKS_3': 50, 'SUBJECT_4': ' MATH', 'MARKS_4': 50, 'SU
BJECT_5': ' CSC', 'MARKS_5': 50}
```

- **TO GET STUDENT RESULT IN GIVEN BATCH : -**

```
        press 'S' or 's' to show results in batch 2022 :
        press 'A' OR 'a' to add a new result class :
        press 'D' OR 'd' to get student details in result :
        press 'C' OR 'c' to get result of students :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student result :
        press 'R' or 'r' to remove a student :
        press 'G' or 'g' to generate pdf :
        press 'E' or 'e' to email result :
 C
enter class and section to get student details :XII_C
enter exam type :PT_1
{'NAME': ' DIVYAM SETHI ', 'class_': 'xii', 'FATHER': ' SANJAY SETHI ', 'MOTHER': ' RACHNA SETHI ', 'ADM_DATE': No
ne, 'FEES': 5000, 'BALANCE': 0, 'PHONE': 99393393, 'EMAIL': ' divyamsethi1804@gmail.com ', 'ADDRESS': 'outdnjf', '
DOB': None, 'ADMISSION_NO': 1, 'MAX_MARKS': 50, 'EXAM_TYPE': ' PERIODIC TEST', 'SUBJECT_1': ' ENG', 'MARKS_1': 50,
 'SUBJECT_2': ' PHY', 'MARKS_2': 50, 'SUBJECT_3': ' CHEM', 'MARKS_3': 50, 'SUBJECT_4': ' MATH', 'MARKS_4': 50, 'SU
BJECT_5': ' CSC', 'MARKS_5': 50}
{'NAME': ' YASH MAINI', 'class_': ' XII_C', 'FATHER': ' KAPIL MAINI', 'MOTHER': ' MEETU MAINI', 'ADM_DATE': dateti
me.date(2022, 3, 23), 'FEES': 200, 'BALANCE': 5000, 'PHONE': 8920028757, 'EMAIL': ' mainiyash2@gmail.com', 'ADDRES
S': ' T 1063 PUNJABI BASTI KAROL BAGH DELHI', 'DOB': datetime.date(2004, 6, 30), 'ADMISSION_NO': 3, 'MAX_MARKS': 3
0, 'EXAM_TYPE': ' TERM - I', 'SUBJECT_1': ' ENGLISH', 'MARKS_1': 0, 'SUBJECT_2': ' PHYSICS', 'MARKS_2': 0, 'SUBJEC
T_3': ' CHEMISTRY', 'MARKS_3': 0, 'SUBJECT_4': ' MATHS', 'MARKS_4': 0, 'SUBJECT_5': ' BIOLOGY', 'MARKS_5': 0}
```

- **TO ADD STUDENT RESULT IN GIVEN RESULT BATCH : -**

```
        press 'S' or 's' to show results in batch 2022 :
        press 'A' OR 'a' to add a new result class :
        press 'D' OR 'd' to get student details in result :
        press 'C' OR 'c' to get result of students :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student result :
        press 'R' or 'r' to remove a student :
        press 'G' or 'g' to generate pdf :
        press 'E' or 'e' to email result :
 R
enter class and section to get student details :XI_C
enter admission number of student to remove from class :4
enter exam type :FINAL
student removed from result successfully
```

- **TO ADD STUDENT RESULT IN GIVEN RESULT BATCH : -**

```
        press 'S' or 's' to show results in batch 2022 :
        press 'A' OR 'a' to add a new result class :
        press 'D' OR 'd' to get student details in result :
        press 'C' OR 'c' to get result of students :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student result :
        press 'R' or 'r' to remove a student :
        press 'G' or 'g' to generate pdf :
        press 'E' or 'e' to email result :
 W
enter class and section to add student details :XI_C
enter exam type :FINAL
ADMISSION_NO ;- 4
MAX_MARKS ;- 50
EXAM_TYPE ;- FINAL EXAMS
SUBJECT_1 ;- ENGLISH
MARKS_1 ;- 100
SUBJECT_2 ;- PHYSICS
MARKS_2 ;- 100
SUBJECT_3 ;- CHEMISTRY
MARKS_3 ;- 100
SUBJECT_4 ;- MATHS
MARKS_4 ;- 100
SUBJECT_5 ;- COMPUTER SCIENCE
MARKS_5 ;- 100
record added succesfully
```

- **TO EDIT STUDENT RESULT IN GIVEN RESULT BATCH : -**

```
        press 'S' or 's' to show results in batch 2022 :
        press 'A' OR 'a' to add a new result class :
        press 'D' OR 'd' to get student details in result :
        press 'C' OR 'c' to get result of students :
        press 'W' or 'w' to add new student :
        press 'M' or 'm' to modify student result :
        press 'R' or 'r' to remove a student :
        press 'G' or 'g' to generate pdf :
        press 'E' or 'e' to email result :
 M
enter class and section to get student details :XI_C
enter admission number of student to modify changes :4
enter exam type :FINAL
MAX_MARKS = 50   :100
EXAM_TYPE =  FINAL EXAMS   :
SUBJECT_1 =  ENGLISH   :
MARKS_1 = 100   :
SUBJECT_2 =  PHYSICS   :
MARKS_2 = 100   :
SUBJECT_3 =  CHEMISTRY   :
MARKS_3 = 100   :
SUBJECT_4 =  MATHS   :
MARKS_4 = 100   :
SUBJECT_5 =  COMPUTER SCIENCE   :
MARKS_5 = 100   :
changes done successfully
```

- **TO GENERATE REPORT CARD IN GIVEN RESULT BATCH : -**

```
       press 'S' or 's' to show results in batch 2022 :
       press 'A' OR 'a' to add a new result class :
       press 'D' OR 'd' to get student details in result :
       press 'C' OR 'c' to get result of students :
       press 'W' or 'w' to add new student :
       press 'M' or 'm' to modify student result :
       press 'R' or 'r' to remove a student :
       press 'G' or 'g' to generate pdf :
       press 'E' or 'e' to email result :
 G
enter class and section to get student details :XII_C
enter exam type :PT_1
C:\Users\divya\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\
Python39\site-packages\fpdf\fpdf.py:710: UserWarning: Substitutting Arial by core font Helvetica
  warnings.warn("Substitutting Arial by core font Helvetica")
 report cards are generated pls check "D:\ report card"
```

J.D. TYTLER SCHOOL

PERIODIC TEST

NAME : DIVYAM SETHI **ADMISSION NO : 1**

CLASS : XII_C **DOB : None**

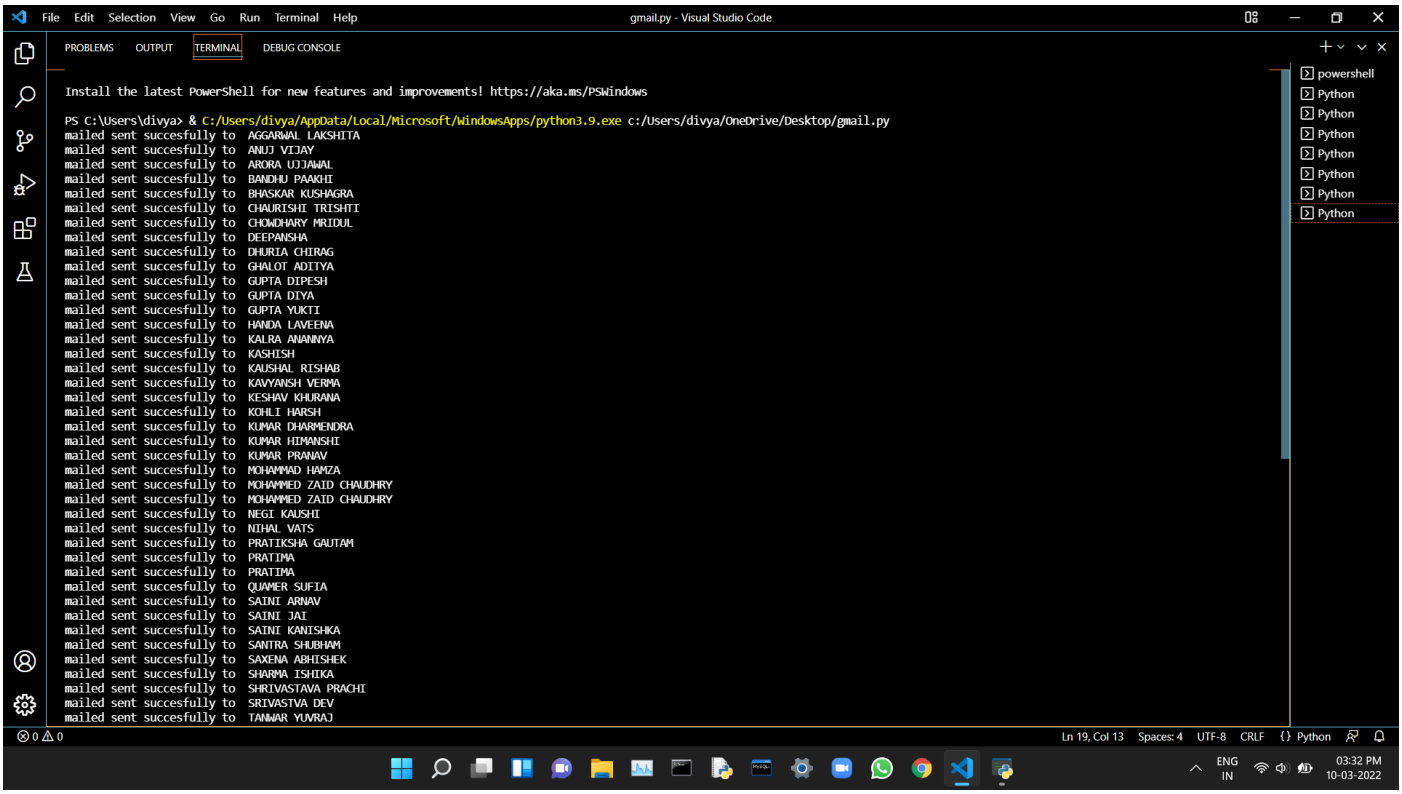
SUBJECTS	MARKS (50)	PERCENTAGE	GRADES
ENG	50	100.0 %	A1
PHY	50	100.0 %	A1
CHEM	50	100.0 %	A1
MATH	50	100.0 %	A1
CSC	50	100.0 %	A1
GRAND TOTAL	250.0	100.0 %	

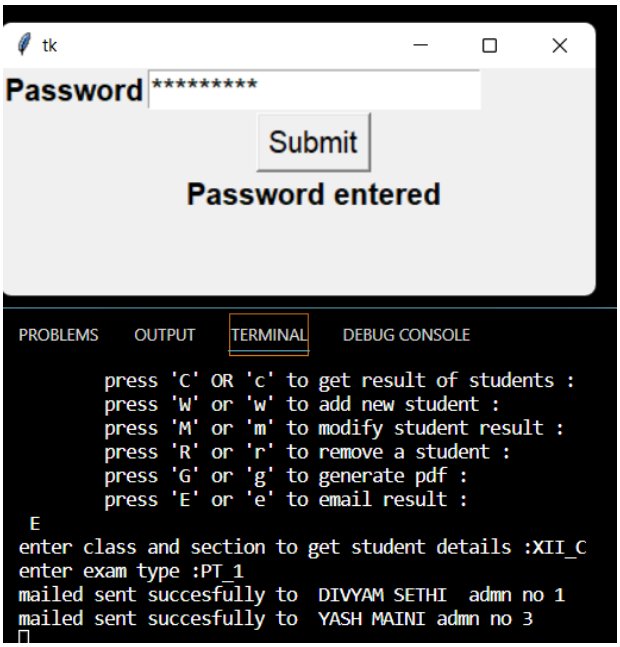
PERCENTAGE	MARKS	GRADE
91% - 100%	23 to 25	A1
81% - 90%	21 to 22	A2
71% - 80%	18 to 20	B1
61% - 70%	16 to 17	B2
51% - 60%	13 to 15	C1
41% - 50%	11 to 12	C2
33% - 40%	8 to 10	D
Below 33%	0 to 7	E

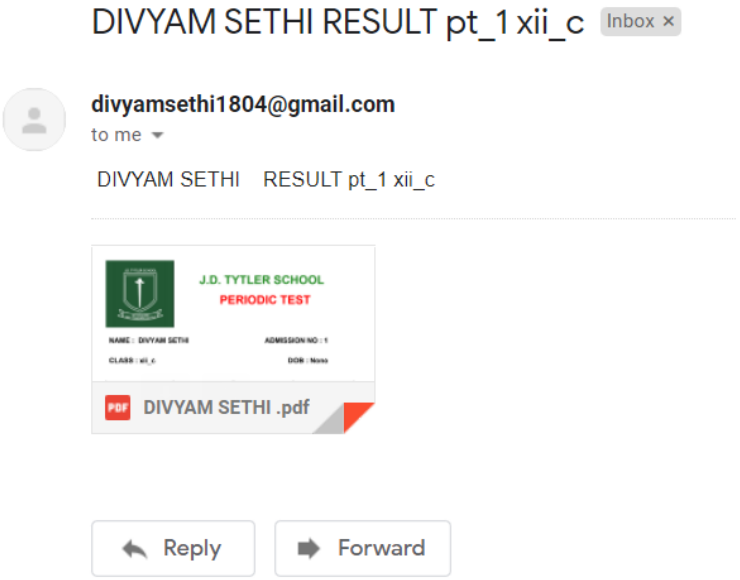
Ms NEENA ANDREW **DR AZHAR ASLAM KHAN**

PRINCIPAL **CLASS TEACHER**

- **TO EMAIL REPORT CARD IN GIVEN RESULT BATCH : -**







HARDWARE AND SOFTWARE REQUIREMENTS

I.OPERATING SYSTEM : WINDOWS 7 AND ABOVE

II. PROCESSOR : PENTIUM(ANY) OR AMD
ATHALON(3800+- 4200+ DUAL CORE)

III. MOTHERBOARD : 1.845 OR 915,995 FOR PENTIUM 0R MSI
K9MM-V VIA K8M800+8237R PLUS
CHIPSET FOR AMD ATHALON

IV. RAM : 512MB+

V. Hard disk : SATA 40 GB OR ABOVE

VI. CD/DVD r/w multi drive combo: (If back up required)

VII. FLOPPY DRIVE 1.44 MB : (If Backup required)

VIII. MONITOR 14.1 or 15 -17 inch

IX. Key board and mouse

X. Printer : (if print is required – [Hard copy])

SOFTWARE REQUIREMENTS:

I. Windows OS
II. Python with MySQLconnector module downloaded

BIBLIOGRAPHY

1. ***Website: https://www.geeksforgeeks.org***
2. ***Website: https://www.w3resource.com***
3. ***Website: https://stackoverflow.com***

***