PERFORMANCE MANAGEMENT SYSTEM

A PROJECT REPORT SUBMITTED

FOR

LABORATORY OF

COMPUTER SCIENCE (083)

IN

CLASS XII

BY

HARSHEET SHARMA (ROLL NO-21) KALLOL DEY (ROLL NO-22)

UNDER THE SUPERVISION OF

MR. GOVIND PRASAD ARYA

PGT- COMPUTER SCIENCE KENDRIYA VIDYALAYA COOCHBEHAR WEST BENGAL



TO THE KOLKATA REGION, KENDRIYA VIDYALAYA SANGATHAN

JANUARY, 2021

CERTIFICATE

This is to certified that Harsheet Sharma (Roll No-21) and Kallol Dey (Roll No-18) have carried out the project work presented in this report entitled "PERFORMANCE MANAGEMENT SYSTEM" for laboratory of computer science (083) in class XII, from Kendriya Vidyalaya Coochbehar, West Bengal under my supervision. The report embodies results of original work, and studies/implementation is carried out by the student themselves.

Date:

Mr. Govind Prasad Arya PGT- Computer Science Kendriya Vidyalaya Coochbehar West Bengal

REPORT APPROVAL SHEET

This is certified that the Project entitled "AYUSH SYSTEM" is approved for laboratory of computer science (083) in class XII, from Kendriya Vidyalaya Coochbehar, West Bengal.

Name and Signature of project group members-						
1. Harsheet Sharma (Roll No-21) 2. Kallol Dey (Roll No-22)						
(Mr. Govind Prasad Arya) Name & Signature of Project Supervisor						
Name & Signature of Internal Examiner						
Name & Signature of External Examiner						

ABSTRACT

Today is the generation of internet and computers. The most preferred mode of conveyance of information is through the mode of internet and computer application. Among that one of the important aspect is managing the data carrying information of Student's performance. Nowadays most schools have shifted to computer based marks entry for students and in this project our attempt is to simplify the process. With the help of simple and user friendly UI and fast connection the difficult task of marks entry becomes pretty simplified. For that purpose we have solely used the tkinter module python for building GUI, functions for GUI calling and MySQL for the purpose of DBMS.

ACKNOWLEDGEMENTS

It is a great pleasure to express my profound sense of gratitude and reverence to my project supervisor Mr. Govind Prasad Arya, PGT Computer Science, Kendriya Vidyalaya Cooch Behar. He was always a source of encouragement and inspiration, and constantly guided us for the accomplishment of this task with meticulous care. We owe to him the most, to have had the opportunity to accomplish the work under his guidance.

On a moral personal note, our deepest appreciation and gratitude to our beloved parents, who have been an inspiration and have provided us with unrelenting encouragement and support.

Harsheet Sharma Kallol Dey

TABLE OF CONTENTS

		Page No.
	COVER PAGE	1
	CERTIFICATE	2
	REPORT APPROVAL SHEET	3
	ABSTRACT	4
	ACKNOWLEDGEMENTS	5
CHAPTER 1:	7-8	
1.1	GUI	7
1.2	FUNCTIONS	7
1.3	DBMS	7-8
CHAPTER 2	9-19	
2.1	FRONT END	9-18
2.2	BACK END	19
2.3	LINKERS	19
CHAPTER 3:	CONCLUSION AND FUTURE ENHANCEMENT	20
REFERENCE	ES	21

CHAPTER 1

INTRODUCTION

1.1 GUI

GUI is an interface that allows users to interact with different electronic devices using icons and other visual indicators. The graphical user interfaces were created because command line interfaces were quite complicated and it was difficult to learn all the commands in it. In today's times, graphical user interfaces are used in many devices such as mobiles, MP3 players, gaming devices, smartphones etc.

Our GUI mainly contains the given 3 elements:

- a) WINDOW
- b) LABELS
- c) ENTRY BOXES
- d) DROPDOWN BOXES
- e) BUTTONS

1.2 FUNCTIONS

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing. You have already seen various functions like **print()** and **main()**. These are called built-in functions provided by the language itself, but we can write our own functions as well.

Good thing about functions is that they are famous with several names. Different programming languages name them differently, for example, functions, methods, subroutines, procedures, etc

There are few terminologies regarding functions. Those are:

A) ARGUMENTS:

These are the parameters passed to a function for proper utility of the function

B) VALUES:

Functions after calling returns some value and that value can be further used in the main code under which the function is defined

1.3 DBMS

Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc. For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

- CREATE: to create a database and its objects like (table, index, views, store procedure, function, and triggers)
- ALTER: alters the structure of the existing database
- DROP: delete objects from the database

- TRUNCATE: remove all records from a table, including all spaces allocated for the records are removed
- COMMENT: add comments to the data dictionary
- RENAME: rename an object

DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

- SELECT: retrieve data from a database
- INSERT: insert data into a table
- UPDATE: updates existing data within a table
- DELETE: Delete all records from a database table
- MERGE: UPSERT operation (insert or update)
- CALL: call a PL/SQL or Java subprogram
- EXPLAIN PLAN: interpretation of the data access path
- LOCK TABLE: concurrency Control

CHAPTER 2

PROJECT IMPLEMENTATION

2.1 FRONT END

The front end contains the three forms directly interacting with users. These pages are the main user interfaces. These pages are as following

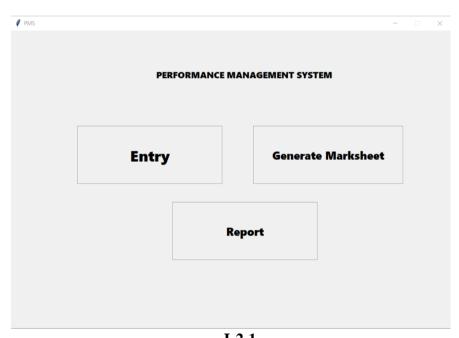
2.1 Forms containing category buttons

In the first page there are 3 buttons containing texts as the categories and user need to click one of them in order to select the category which he or she wants to access.

The whole button and UI system has been built using the tkinter module of the Python language. The soul motive behind the designing of the UI is to keep it user friendly and attractive for the user. The attractive nature of the first page is achieved through two properties:

- a) Uniformity in the different components of the page.
- b) The vibrant colours and attractive outlooks of the background.

This UI is made under special guidance of the team leader, the project advisor and some well wishers having grip in page designing. This whole UI is defined under the main/toplevel function of the python and when the program is run first this page comes in front of the user.



I-2.1

The GUI of the above mentioned page is shown in image I-2.1

```
def init (self, parent):# frontpage
    top=tk.Tk()
    top.minsize(148,1)
    top.maxsize(1924,1055)
    top.resizable(0,0)
    top.title("PMS")
   top.geometry("926x625")
         -----Label--
    label=tk.Label(top,text='PERFORMANCE MANAGEMENT SYSTEM',font=("Segoe UI Black", 14 , "bold"))
    label.place(relx=0.238, rely=0.08, height=86, width=542)
#----Button-----
    entry=tk.Button(top,text='Entry',relief="groove",font=("Segoe UI Black", 24 , "bold"))
    entry.configure(foreground='black', highlightbackground="gray85")
    entry.place(relx=0.151, rely=0.32, height=123, width=306)
    entry.configure(pady="0")
    entry.configure(command=lambda:Entry(top))
    msheet=tk.Button(top,text='Generate Marksheet',relief="groove",font=("Segoe UI Black", 18 , "bold"))
    msheet.configure(foreground='black',highlightbackground="gray85")
    msheet.place(relx=0.551, rely=0.32, height=123, width=316)
    msheet.configure(pady="0")
    msheet.configure(command=lambda:ME(top))
    rport=tk.Button(top,text='Report',relief="groove",font=("Segoe UI Black", 18 , "bold"))
rport.configure(foreground='black',highlightbackground="gray85")
    rport.place(relx=0.367, rely=0.576, height=123, width=306)
    rport.configure(pady="0")
    rport.configure(command=lambda:report(top))
global Entry
```

I-2.2

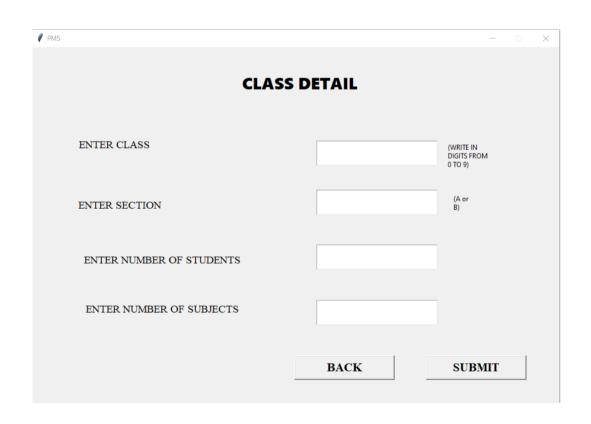
The main function of this form is to establish integrity between all the undermentioned pages. The code snippet representing the functionalities of the page is shown in the image I-2.2

2.1.2 Class Details Form

The class detail page has four attributes of tkinter(Python) namely label, button, entry box and option menu. This page gives the options to the user to enter the particulars of a class. The submit button directs the user towards the home page itself. This UI is defined under the self-defined function symptom.

The code snippet and the GUI output for the above mentioned form is shown in image I- 2.3 The main function of this page is to generate a user friendly GUI and take inputs of the class, section of that class, number of subjects and number of students of that class and insert the collected data into the MySQL table class_detail discussed later on in section 2.2.

The page class details is called through the Entry button with in between a gui for selecting between class detail and student details forms.



```
def CD(cp):#entry qui command
    #entry gui
    . . . . . . . .
    cp.withdraw()
    #This class configures and populates the toplevel window.
       #top is the toplevel containing window.
    top=tk.Tk()
    top.minsize(148,1)
    top.maxsize(1924,1055)
    top.resizable(0,0)
    top.title("PMS")
    top.geometry("926x625")
                  ---label-
    top_label=tk.Label(top,text='CLASS_DETAIL',font=("Segoe_UI_Black", 22 , "bold"))
    top label.place(relx=0.248, rely=0.062, height=46, width=482)
    Class=tk.Label(top,text='ENTER CLASS',font=("Times New Roman", 14 , "normal"))
    Class.place(relx=0.043, rely=0.231, height=56, width=212)
    section=tk.Label(top,text='ENTER SECTION',font=("Times New Roman", 14 , "normal"))
    section.place(relx=0.054, rely=0.4, height=56, width=212)
    student=tk.Label(top,text='ENTER NUMBER OF STUDENTS',font=("Times New Roman", 14 , "normal"))
    student.place(relx=0.064, rely=0.554, height=56, width=342)
    subject=tk.Label(top,text='ENTER NUMBER OF SUBJECTS',font=("Times New Roman", 14 , "normal"))
    subject.place(relx=0.064, rely=0.692, height=56, width=342)
         ----Entry--
    clas=tk.Entry(top, font="TkFixedFont", background="white")
    clas.place(relx=0.54, rely=0.262, height=44, relwidth=0.229)
    sec=tk.Entry(top,font="TkFixedFont",background="white")
    sec.place(relx=0.54, rely=0.4, height=44, relwidth=0.229)
    stu=tk.Entry(top,font="TkFixedFont",background="white")
```

The code snippet and GUI of the Class Details form is shown in images I-2.3 and I-2.4 respectively

2.1.3 Student Entry Form

Student Entry form consists of mainly four components of tkinter namely labels, buttons, entry boxes and optionmenus. All the four components together collects the inputs of the particulars of a student like Class ,Section ,Roll Number, Name and group of subjects the students have and insert all the details into the respective class and subject tables.

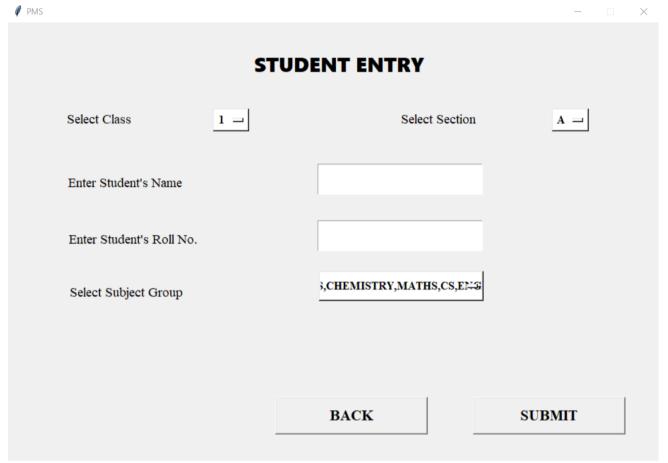
The gui is guided with some functions for proper collection and insertion of the data into the database. The code snippet containing the functions is shown in image I-2.5, I-2.6 and I-2.7 and the Interface is shown in the image I-2.8

```
Label=tk.Label(top,text="STUDENT ENTRY",font=("Segoe UI Black",22,"bold"))
Label.place(relx=0.238, rely=0.048, height=56, width=502)
clas=tk.Label(top,text="Select Class",font=("Times New Roman",14,"normal"))
clas.place(relx=0.043, rely=0.192, height=36, width=182)
section=tk.Label(top,text="Select Section",font=("Times New Roman",14,"normal"))
section.place(relx=0.562, rely=0.192, height=36, width=182)
Name=tk.Label(top,text="Enter Student's Name",font=("Times New Roman",14,"normal"))
Name.place(relx=0.018, rely=0.336, height=36, width=302)
rn=tk.Label(top,text="Enter Student's Roll No.",font=("Times New Roman",14,"normal"))
rn.place(relx=0.025, rely=0.464, height=36, width=312)
sub=tk.Label(top,text="Select Subject Group",font=("Times New Roman",14,"normal"))
sub.place(relx=0.025, rely=0.576, height=46, width=292)
#------
#------
classes = [1,2,3,4,5,6,7,8,9,10,11,12] #etc
variabl= tk.IntVar(top)
variabl.set(classes[0])
cl = tk.OptionMenu(top, variabl, *classes)
cl.place(relx=0.313, rely=0.192, relheight=0.058,relwidth=0.060)
cl.configure(background="white")
cl.configure(disabledforeground="#a3a3a3")
cl.configure(font=('Times New Roman', 12, 'bold'))
cl.configure(foreground="#000000")
secti = ['A','B'] #etc
variable = tk.StringVar(top)
variable.set(secti[0])
```

-----Lapels-----

```
sect.place(relx=0.832, rely=0.192, relheight=0.058, relwidth=0.060)
    sect.configure(background="white")
    sect.configure (disabledforeground="#a3a3a3")
    sect.configure(font=('Times New Roman', 12, 'bold'))
    sect.configure(foreground="#000000")
    n=tk.Entry(top,font=('Times New Roman',16,'bold'))
    n.place(relx=0.475, rely=0.448, height=44, relwidth=0.253)
    r=tk.Entry(top, font=('Times New Roman', 16, 'bold'))
    r.place(relx=0.475, rely=0.32, height=44, relwidth=0.253)
    s=['PHYSICS, CHEMISTRY, MATHS, CS, ENGLISH', 'PHYSICS, CHEMISTRY, MATHS, BIO, ENGLISH', 'PHYSICS, CHEMISTRY, HINDI, BIO, ENGLIS
    var= tk.StringVar(top)
    var.set(s[0])
    su=tk.OptionMenu(top , var , *s)
su.place(relx=0.475, rely=0.56, relheight=0.074, relwidth=0.256)
    su.configure(background="white")
    su.configure(disabledforeground="#a3a3a3")
    su.configure(font=('Times New Roman', 12, 'bold'))
    su.configure(foreground="#000000")
                           ----submit function--
    def sub():
         if str(r.get()).isdigit() == False :
             w=tk.Tk("error")
             w.geometry("200x120")
             w.resizable(0,0)
             tk.Label(w,text="INVALID ROLL NUMBER",font=("Times New Roman",16,"bold")).place(x=1,y=1,height=100,width=
        elif len(str(n.get()))==0:
             w=tk.Tk("error")
             w.geometry("200x120")
             w.resizable(0,0)
                                                    I-2.6
           tk.Label(w,text="PLEASE ENTER NAME",font=("Times New Roman",16,"bold")).place(x=1,y=1,height=100,width=200)
       elif str(n.get()).isdigit() == True :
           w=tk.Tk("error")
           w.geometry("400x120")
           w.resizable(0,0)
           tk.Label(w.text="INVALID NAME", font=("Times New Roman", 16, "bold")).place(x=1, v=1, height=100, width=400)
       else:
               con=mc.connect(user="root",host="localhost",password="root",database="pms")
               cr=con.cursor()
               cr.execute("INSERT INTO %d%s('RN', 'NAME', 'SUBJECTS') VALUES(%d,%s,%s)"%(int(r.get()),str(n.get()),str(var.get())))
               con.commit()
           except:
               w=tk.Tk("error")
           w.geometry("200x120")
w.resizable(0,0)
           tk.Label(w,text="CLASS DETAIL UNAVAILABLE",font=("Times New Roman",16,"bold")).place(x=1,y=1,height=100,width=200)
   submit=tk.Button(top,text="SUBMIT",font=("Times New Roman",16,"bold"))
   submit.place(relx=0.713, rely=0.848, height=53, width=216)
   submit.configure(command=lambda:sub())
   bck=tk.Button(top,text="BACK",font=("Times New Roman",16,"bold"))
   bck.place(relx=0.41, rely=0.848, height=53, width=216)
   bck.configure(command=lambda:back(sp,top))
global ReportEntry
def ReportEntry(rp):
   rp.withdraw()
```

I-2.7



I-2.8

2.1.4 Marks Entry form or the Marksheet Form

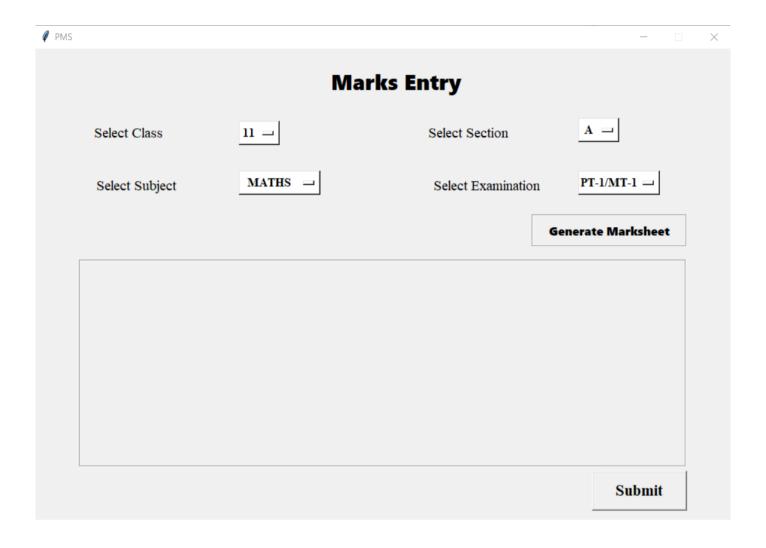
This form also contains the above mentioned four elements of tkinter and in addition a grid consisting of labels and entry boxes are also present. The main functions of the previous four elements are similar but the function of the grid here is to collect the marks and the roll numbers of the respective students whose marks are to be entered. The collected marks is then stored into the database for further use while reporting.

The code snippet and interactive GUI output is shown in images I-2.9,I-2.10 and I-2.11 respectively. The empty space shown in the gui is for generating the grid for the entry which is not shown here but will be shown while live implementation.

```
-----Entry & DropDownBox-----
s=L
var= tk.IntVar(top)
var.set(s[0])
cl=tk.OptionMenu(top , var , *s) cl.place(relx=0.292, rely=0.15, relheight=0.058 , relwidth=0.062) cl.configure(background="white")
cl.configure (disabledforeground="#a3a3a3")
cl.configure(font=('Times New Roman', 12, 'bold'))
cl.configure(foreground="#000000")
k=["A","B"]
v=tk.StringVar(top)
v.set(k[0])
\label{eq:sec_tk.optionMenu} \begin{array}{lll} \texttt{sec=tk.OptionMenu} \, (\texttt{top} & , & \texttt{v} & , & \texttt{*k}) \\ \texttt{sec.place} \, (\texttt{relx=0.778}, & \texttt{rely=0.144}, & \texttt{relheight=0.058} & , & \texttt{relwidth=0.062}) \\ \end{array}
sec.configure(background="white")
sec.configure(disabledforeground="#a3a3a3")
sec.configure(font=('Times New Roman', 12, 'bold'))
sec.configure(foreground="#000000")
sb=["MATHS", "HINDI", "ENGLISH", "BIO", "CS", "EVS", "SCIENCE", "PHYSICS", "CHEMISTRY", "BIOLOGY", "ECONOMICS", "POLITICAL
V=tk.StringVar(top)
V.set(sb[0])
                             , V , *sb)
sub=tk.OptionMenu(top
sub.place(relx=0.292, rely=0.256, relheight=0.058, relwidth=0.120)
sub.configure (background="white")
sub.configure(disabledforeground="#a3a3a3")
sub.configure(font=('Times New Roman', 12, 'bold'))
sub.configure(foreground="#000000")
exa=['PT-1/MT-1','PT-2/MT-2','PT-3/MT-3','HALF YEARLY','SESSION ENDING']
vex=tk.StringVar(top)
vex.set(exa[0])
```

I-2.9

```
global ME
def ME (mp): #exit gui command
    mp.withdraw()
    top=tk.Tk()
    top.minsize(148,1)
    top.maxsize(1924,1055)
    top.resizable(0,0)
    top.title("PMS")
    top.geometry("926x625")
    con=mc.connect(user="root",database="pms",host="localhost",password="root")
    cr=con.cursor()
    cr.execute("SELECT CLASS, SECTION FROM class detail")
    f=cr.fetchall()
    L=[]
    for i in f:
        L.append(i[0])
    label=tk.Label(top,text='Marks Entry',font=("Segoe UI Black", 22 , "bold"))
label.place(relx=0.335, rely=0.032, height=46, width=342)
    sc=tk.Label(top,text='Select Class',font=("Times New Roman", 14 , "normal"))
    sc.place(relx=0.065, rely=0.16, height=26, width=132)
    ss=tk.Label(top,text='Select Section',font=("Times New Roman", 14 , "normal"))
    ss.place(relx=0.551, rely=0.16, height=26, width=132)
    ssub=tk.Label(top,text='Select Subject',font=("Times New Roman", 14 , "normal"))
ssub.place(relx=0.076, rely=0.272, height=26, width=132)
    se=tk.Label(top,text='Select Examination',font=("Times New Roman", 14 , "normal"))
    se.place(relx=0.551, rely=0.272, height=26, width=182)
    #-----Entry & DropDownBox-----
```



I-2.11

2.1.5 Report Form

The report form gives the report of the performance of the student by collecting the marks and the weightages from the database which was preciously present. The report page shows the end of our project as it is the last page and contains a button to restart the process. This page contains all the elements mentioned in the marks entry form.

The code snippets and the gui of the page is shown in images I-2.12, I-2.13, I-2.14, I-2.15 repectively

```
global ReportEntry
def ReportEntry(rp):
    rp.withdraw()
    top=tk.Tk()
    top.minsize(148,1)
    top.maxsize(1924,1055)
    top.resizable(0,0)
    top.title("PMS")
    top.geometry("926x625")
    Label=tk.Label(top,text="PRE-REPORT",font=("Segoe UI Black",24,"normal"))
   Label.place(relx=0.324, rely=0.016, height=47, width=324)
    Class=tk.Label(top,text="SELECT CLASS",font=("Times New Roman",16,"normal"))
   Sec=tk.Label(top.text="SELECT SECTION".font=("Times New Roman".16."normal"))
    rn=tk.Label(top,text="ENTER ROLL NUMBER",font=("Times New Roman",16,"normal"))
global report.
def report (rp):
    rp.withdraw()
    top=tk.Tk()
    top.minsize(148,1)
    top.maxsize(1924,1055)
    top.resizable(0,0)
    top.title("PMS")
    top.geometry("926x625")
           -----Frames--
    tf=tk.Frame(top,relief="groove",borderwidth="2")
    tf.place(relx=0.0, rely=0.0, relheight=0.216, relwidth=1.001)
```

bck.configure(command=lambda:back(sp,top))

I-2.12

```
nf=tk.Frame(top,relief="groove",borderwidth="2")
nf.place(relx=0.0, rely=0.208, relheight=0.232, relwidth=1.001)
mf=tk.Frame(top,relief="groove",borderwidth="2")
mf.place(relx=0.0, rely=0.432, relheight=0.456, relwidth=1.001)
             --Labels--
kv=tk.Label(tf,text="KENDRIYA VIDYALAYA COOCHBEHAR",font=("Seqoe UI Black",22,"normal"))
kv.place(relx=0.13, rely=0.074, height=46, width=700)
add=tk.Label(tf,text="ADRESS: NILKUTHI BABURHAT, COOCHBEHAR-PIN:736156")
add.place(relx=0.303, rely=0.37, height=26, width=407)
session=tk.Label(tf,text="SESSION:2020-2021",font=("Segoe UI Black",16,"normal"))
session.place(relx=0.357, rely=0.593, height=36, width=282)
clas=tk.Label(nf,text="CLASS :",font=("Times New Roman",14,"normal"))
clas.place(relx=0.032, rely=0.138, height=32, width=88)
section=tk.Label(nf,text="SECTION :",font=("Times New Roman",14,"normal"))
section.place(relx=0.368, rely=0.138, height=32, width=114)
roll numbr=tk.Label(nf,text="ROLL NUMBER :",font=("Times New Roman",14,"normal"))
roll numbr.place(relx=0.67, rely=0.138, height=32, width=174)
name=tk.Label(nf,text="NAME :",font=("Times New Roman",14,"normal"))
name.place(relx=0.032, rely=0.621, height=32, width=81)
cl=tk.Label(nf,text="12",relief="groove",font=("Times New Roman",14,"normal"))
cl.place(relx=0.173, rely=0.138, height=32, width=48)
s=tk.Label(nf,text="B",relief="groove",font=("Times New Roman",14,"normal"))
s.place(relx=0.551, rely=0.138, height=32, width=30)
rn=tk.Label(nf,text="21",relief="groove",font=("Times New Roman",14,"normal"))
rn.place(relx=0.897, rely=0.138, height=32, width=38)
```

I-2.14

₽ PMS					- 🗆 X			
KENDRIYA VIDYALAYA COOCHBEHAR ADRESS: NILKUTHI BABURHAT, COOCHBEHAR-PIN:736156 SESSION:2020-2021								
CLASS:	12	SECTION:	В	ROLL NUMBER :	21			
NAME :	HARSHEET S	HARMA						
					1			
				H	OME			

I-2.15

2.2 BACK END

The backend consists of two main components; the database and the linkers. The database is a MySQL database and the connection is established through mysql.connector module of the python.

2.2.1 DATABSASE

The database is the backbone of any application or program and in this program the database is a mysql database. The name of the database used in this application is PMS. The database contains variable number of relations containing the information of the subjects of the students, class containing the students and the details of all classes collectively. The database is accessed through python using the mysql.connector module. At first the sql query is executed and after running the query the data is fetched in a variable and then further the variable is used for further operations. More information will be clear by viewing the code later on in this segment.

2.3 LINKERS

The linkers are basically page specific and every linking function of every page is given same name, i.e. submit function. The submit function first establishes a connection between the database and the python code. When the connection is established it follows certain constraints and if the constraints are overcome then the page function quits and the previous page is recalled.

CHAPTER 3

CONCLUSION & FUTURE ENHANCEMENT

This chapter summarizes the whole project and gives a brief idea about the project. All the facts discussed in all the above sections. This software is efficient in maintaining Students details and can easily perform operations on students record. This software also reduces the work load of the teachers in school as all the details are stored in computer system and whenever the detail marks of student needed it can be searched and displayed on the screen.

5.2 FUTURE SCOPE

Further we are going to work on the code optimization phase of the compiler and we are going to make our compiler more efficient and dynamic. Also in future, this system can launch on a web portal for easy online entry of students details and marks and their parents can login and check marks and download the reports of their children

References

- [1] <u>https://stackoverflow.com/questions/32577726/python-3-tkinter-how-to-word-wrap-text-in-tkinter-text</u>
- [2]

 $\underline{https://www.tutorialspoint.com/python/tk_relief.htm\#:\sim:text=Advertisements,be\%\,20used\%\,20for\%\,2}\\ \underline{0relief\%\,20attribute}.$

[3] https://www.delftstack.com/howto/python-tkinter/how-to-pass-arguments-to-tkinter-button-command/