# importing all the modules required

import mysql.connector as sqltor

import sys

from PyQt5.uic import loadUi

from PyQt5 import QtWidgets

from PyQt5.QtWidgets import QDialog, QApplication

from PyQt5.QtCore import pyqtSlot

mydb = sqltor.connect(host="localhost", user="root", passwd="password")# establishing a connection with sql database

sqlcursor = mydb.cursor()# assigning cursor to sqlcursor

class login\_pg(QDialog):

def \_\_init\_\_(self):

super(login\_pg, self).\_\_init\_\_()

loadUi("pro\_login\_pg.ui", self)# the first page , login page

self.submit\_button.clicked.connect(self.logingate)

def logingate(self):

dbname = ('login',)

sqlcursor.execute("show databases;")

data = sqlcursor.fetchall()

if dbname in data:

pass

else:# creating the database if it does not exist

sqlcursor.execute("create database login")

sqlcursor.execute("use login")

sqlcursor.execute("create table logins( username char(130), password char(130), desig char(130))")

sqlcursor.execute("insert into logins values('masterlogin','12345master67890','master')")

mydb.commit()

username = self.username\_line.text()

password = self.password\_line.text()

sqlcursor.execute("use login")

sqlcursor.execute("select password from logins where username = '"+username+"'")

data = sqlcursor.fetchall()

if password == data[0][0]:

sqlcursor.execute("select desig from logins where username = '"+username+"'")

item = sqlcursor.fetchall()

if item[0][0] == "master":

widget.setCurrentIndex(1)

if item[0][0] == "staff":

with open("username.txt", 'w') as fh:

fh.write(username)

widget.setCurrentIndex(4)

else:

self.error\_label.setText("wrong username or password , please try again....!")

class master\_main\_pg(QDialog):

def \_\_init\_\_(self):

super(master\_main\_pg,self).\_\_init\_\_()

loadUi("pro\_master\_login\_main\_pg.ui", self)

self.add\_participant\_butt.clicked.connect(self.to\_addparticipant)

self.remove\_participant\_butt.clicked.connect(self.to\_removeparticipant)

self.edit\_participant\_butt.clicked.connect(self.to\_editparticipant)

self.set\_year\_butt.clicked.connect(self.to\_setyear)

def to\_editparticipant(self):

widget.setCurrentIndex(17)

def to\_addparticipant(self):

widget.setCurrentIndex(2)

def to\_removeparticipant(self):

widget.setCurrentIndex(3)

def to\_setyear(self):

widget.setCurrentIndex(21)

class add\_participant(QDialog):

def \_\_init\_\_(self):

super(add\_participant, self).\_\_init\_\_()

loadUi("pro\_add\_participantant\_pg.ui", self)

self.add\_execute.clicked.connect(self.addpart)

self.prev\_pg\_execute.clicked.connect(self.prevpage)

def addpart(self):

sqlcursor.execute("use login")

username = self.add\_part\_username.text()

desig = self.add\_part\_desig.text()

sqlcursor.execute("use login")

sqlcursor.execute("select username from logins")

data=sqlcursor.fetchall()

for item in data:

if username == item[0]:

self.message\_label.setStyleSheet("color: rgb(255, 0, 0);font: 12pt 'Arial';")

self.message\_label.setText("username already exists")

break

else:

cmd = "insert into logins values('" + username + "', 'password' , '" + desig + "')"

sqlcursor.execute(cmd)

mydb.commit()

self.message\_label.setStyleSheet("color: rgb(0,170, 0);font: 12pt 'Arial';")

self.message\_label.setText("participant has been successfully added")

def prevpage(self):

widget.setCurrentIndex(1)

class remove\_participant(QDialog):

def \_\_init\_\_(self):

super(remove\_participant,self).\_\_init\_\_()

loadUi("pro\_remove\_participant\_pg.ui", self)

self.removepart\_remove\_execute.clicked.connect(self.remove\_part)

self.removepart\_backpg\_execute.clicked.connect(self.prevpage)

def remove\_part(self):

username = self.removeparrt\_username.text()

sqlcursor.execute("use login")

sqlcursor.execute("select username from logins")

data = sqlcursor.fetchall()

for item in data:

if username == item[0]:

cmd = "delete from logins where username = '" + username + "'"

sqlcursor.execute(cmd)

mydb.commit()

self.message\_label.setStyleSheet("color: rgb(0,170,0);font: 12pt 'Arial';")

self.message\_label.setText("the participant has been removed")

break

else:

self.message\_label.setStyleSheet("color: rgb(255, 0, 0);font: 12pt 'Arial';")

self.message\_label.setText("the given username doesn't exit")

def prevpage(self):

widget.setCurrentIndex(1)

class staff\_login\_main\_pg(QDialog):

def \_\_init\_\_(self):

super(staff\_login\_main\_pg,self).\_\_init\_\_()

loadUi("pro\_teacherlogin\_main\_pg.ui", self)

with open("username",'r') as fh:

username = fh.read().strip().title()

self.teacherlogin\_username.setText(username)

with open("notice.txt", 'r') as fh:

data = fh.read()

self.teacherlogin\_notice.setText(data)

self.addsett.clicked.connect(self.to\_additinol\_sett)

self.taecherlogin\_addmarks.clicked.connect(self.to\_addmarks)

self.taecherlogin\_editmarks.clicked.connect(self.to\_editmarks)

self.taecherlogin\_classprogress.clicked.connect(self.to\_classprogress)

self.taecherlogin\_studentprogress.clicked.connect(self.to\_stuprogress)

self.taecherlogin\_archives.clicked.connect(self.to\_archives)

def to\_additinol\_sett(self):

widget.setCurrentIndex(13)

def to\_addmarks(self):

widget.setCurrentIndex(5)

def to\_editmarks(self):

pass

def to\_classprogress(self):

pass

def to\_stuprogress(self):

pass

def to\_archives(self):

pass

class addmarks\_selectclass(QDialog):

def \_\_init\_\_(self):

super(addmarks\_selectclass, self).\_\_init\_\_()

loadUi("pro\_addmarks\_main\_pg.ui", self)

self.back.clicked.connect(self.to\_prevpg)

self.teacherlogin\_class\_1.clicked.connect(self.setclass1)

self.teacherlogin\_class\_2.clicked.connect(self.setclass2)

self.teacherlogin\_class\_3.clicked.connect(self.setclass3)

self.teacherlogin\_class\_4.clicked.connect(self.setclass4)

self.teacherlogin\_class\_5.clicked.connect(self.setclass5)

self.teacherlogin\_class\_6.clicked.connect(self.setclass6)

self.teacherlogin\_class\_7.clicked.connect(self.setclass7)

self.teacherlogin\_class\_8.clicked.connect(self.setclass8)

self.teacherlogin\_class\_9.clicked.connect(self.setclass9)

self.teacherlogin\_class\_10.clicked.connect(self.setclass10)

self.teacherlogin\_class\_11.clicked.connect(self.setclass11)

self.teacherlogin\_class\_12.clicked.connect(self.setclass12)

def setclass1(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class1\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(6)

def setclass2(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class2\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(6)

def setclass3(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class3\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(8)

def setclass4(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class4\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(8)

def setclass5(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class5\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(8)

def setclass6(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class6\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(8)

def setclass7(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class7\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(8)

def setclass8(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class8\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(8)

def setclass9(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class9\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(8)

def setclass10(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class10\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(8)

def setclass11(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class11\n", "subject\n", "test\n","stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(9)

def setclass12(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class12\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(9)

def to\_prevpg(self):

widget.setCurrentIndex(4)

class addmarks\_selectsubject\_12(QDialog):

def \_\_init\_\_(self):

super(addmarks\_selectsubject\_12,self).\_\_init\_\_()

loadUi("pro\_addmarks\_subject\_class12\_pg.ui", self)

self.back\_execute.clicked.connect(self.to\_prevpg)

self.teacherlogin\_subject\_english.clicked.connect(self.seteng)

self.teacherlogin\_subject\_evs.clicked.connect(self.setevs)

self.teacherlogin\_subject\_maths.clicked.connect(self.setmat)

self.teacherlogin\_subject\_lang2.clicked.connect(self.setlan)

self.teacherlogin\_subject\_comp.clicked.connect(self.setcom)

def seteng(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "english\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(7)

def setevs(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "evs\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(7)

def setmat(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "maths\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(7)

def setlan(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "lang2\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(7)

def setcom(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "computer\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(7)

def to\_prevpg(self):

widget.setCurrentIndex(5)

class addmarks\_selectsubject\_345678910(QDialog):

def \_\_init\_\_(self):

super(addmarks\_selectsubject\_345678910, self).\_\_init\_\_()

loadUi("pro\_addmarks\_subject\_class345678910\_pg.ui", self)

self.back\_execute.clicked.connect(self.to\_prevpg)

self.teacherlogin\_subject\_english.clicked.connect(self.seteng)

self.teacherlogin\_subject\_science.clicked.connect(self.setsci)

self.teacherlogin\_subject\_maths.clicked.connect(self.setmat)

self.teacherlogin\_subject\_lang2.clicked.connect(self.setlan)

self.teacherlogin\_subject\_comp.clicked.connect(self.setcom)

self.teacherlogin\_subject\_social.clicked.connect(self.setsoc)

def seteng(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "english\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class10\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setsci(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "science\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class10\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setmat(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "maths\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class10\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setlan(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "lang2\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class10\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setcom(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "computer\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class10\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setsoc(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "social\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class10\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def to\_prevpg(self):

widget.setCurrentIndex(5)

class addmarks\_select\_stream(QDialog):

def \_\_init\_\_(self):

super(addmarks\_select\_stream, self).\_\_init\_\_()

loadUi("pro\_addmarks\_stream\_pg.ui", self)

self.back\_execute.clicked.connect(self.to\_prevpg)

self.addmarks\_stream\_biomaths.clicked.connect(self.setbiomaths)

self.addmarks\_stream\_computer.clicked.connect(self.setcomputersci)

self.addmarks\_stream\_commerce.clicked.connect(self.setcommerce)

def setbiomaths(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[3] = "biomaths\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(10)

def setcomputersci(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[3] = "computermath\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(11)

def setcommerce(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[3] = "commerce\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(12)

def to\_prevpg(self):

widget.setCurrentIndex(5)

class select\_test\_typ1\_pg(QDialog):

def \_\_init\_\_(self):

super(select\_test\_typ1\_pg,self).\_\_init\_\_()

loadUi("pro\_addmarks\_exam\_type1\_pg.ui", self)

self.back\_execute.clicked.connect(self.to\_prevpg)

self.teacherlogin\_exam\_pt1.clicked.connect(self.setpt1)

self.teacherlogin\_exam\_pt2.clicked.connect(self.setpt2)

self.teacherlogin\_exam\_pt3.clicked.connect(self.setpt3)

self.teacherlogin\_exam\_pt4.clicked.connect(self.setpt4)

self.teacherlogin\_subjeteacherlogin\_exam\_halfyear.clicked.connect(self.sethalfyear)

self.teacherlogin\_exam\_annual.clicked.connect(self.setannual)

def setpt1(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[2] = "periodictest1\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(24)

def setpt2(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[2] = "periodictest2\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(24)

def setpt3(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[2] = "periodictest3\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(24)

def setpt4(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[2] = "periodictest4\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(24)

def sethalfyear(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[2] = "halfyearly\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(24)

def setannual(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[2] = "annual\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(24)

def to\_prevpg(self):

with open("classdetails.txt", "r") as fh:

cls=fh.readlines()

if cls[0] in ("class11\n","class12\n"):

if cls[3] == "biomaths\n":

widget.setCurrentIndex(10)

if cls[3] == "computermath\n":

widget.setCurrentIndex(11)

if cls[3] == "commerce\n":

widget.setCurrentIndex(12)

elif cls[0][-2] in ("3","4","5","6","7","8","9","0",) and cls[0][-3] in ("s","1",):

widget.setCurrentIndex(8)

elif cls[0][-2] in ("1","2") and cls[0][-3] in ("s"):

widget.setCurrentIndex(6)

class select\_subject\_biomaths\_pg(QDialog):

def \_\_init\_\_(self):

super(select\_subject\_biomaths\_pg,self).\_\_init\_\_()

loadUi("pro\_addmarks\_subject\_class1211\_biomaths\_pg.ui", self)

self.back\_execute.clicked.connect(self.to\_prevpg)

self.teacherlogin\_subject\_english.clicked.connect(self.seteng)

self.teacherlogin\_subject\_biology.clicked.connect(self.setbio)

self.teacherlogin\_subject\_maths.clicked.connect(self.setmat)

self.teacherlogin\_subject\_chemistry.clicked.connect(self.setche)

self.teacherlogin\_subject\_physcis.clicked.connect(self.setphy)

def seteng(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "english\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setbio(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "biology\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setmat(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "maths\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setche(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "chemistry\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setphy(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "physics\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def to\_prevpg(self):

widget.setCurrentIndex(9)

class select\_subject\_computersci\_pg(QDialog):

def \_\_init\_\_(self):

super(select\_subject\_computersci\_pg,self).\_\_init\_\_()

loadUi("pro\_addmarks\_subject\_class1211\_compmaths\_pg.ui", self)

self.back\_execute.clicked.connect(self.to\_prevpg)

self.teacherlogin\_subject\_english.clicked.connect(self.seteng)

self.teacherlogin\_subject\_computer.clicked.connect(self.setcomp)

self.teacherlogin\_subject\_maths.clicked.connect(self.setmat)

self.teacherlogin\_subject\_chemistry.clicked.connect(self.setche)

self.teacherlogin\_subject\_physcis.clicked.connect(self.setphy)

def seteng(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "english\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setcomp(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "computer\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setmat(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "maths\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setche(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "chemistry\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setphy(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "physics\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def to\_prevpg(self):

widget.setCurrentIndex(9)

class select\_subject\_commerce\_pg(QDialog):

def \_\_init\_\_(self):

super(select\_subject\_commerce\_pg, self).\_\_init\_\_()

loadUi("pro\_addmarks\_subject\_class1211\_commerce\_pg.ui", self)

self.back\_execute.clicked.connect(self.to\_prevpg)

self.teacherlogin\_subject\_english.clicked.connect(self.seteng)

self.teacherlogin\_subject\_computer.clicked.connect(self.setcomp)

self.teacherlogin\_subject\_buisness.clicked.connect(self.setbuss)

self.teacherlogin\_subject\_accounts.clicked.connect(self.setacc)

self.teacherlogin\_subject\_economics.clicked.connect(self.seteco)

def seteng(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "english\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setcomp(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "computer\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setbuss(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "business\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def setacc(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "accounts\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def seteco(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[1] = "economics\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

if clsdetails[0] == "class12\n":

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

def to\_prevpg(self):

widget.setCurrentIndex(9)

class additional\_settings\_pg(QDialog):

def \_\_init\_\_(self):

super(additional\_settings\_pg,self).\_\_init\_\_()

loadUi("additional\_settings\_pg.ui", self)

self.back.clicked.connect(self.to\_prevpg)

self.change\_password.clicked.connect(self.to\_chg\_pass)

self.set\_notice.clicked.connect(self.to\_set\_notice)

#self.creator\_info.clicked.connect(self.to\_creator\_info)

def to\_chg\_pass(self):

widget.setCurrentIndex(14)

def to\_prevpg(self):

widget.setCurrentIndex(4)

def to\_set\_notice(self):

widget.setCurrentIndex(15)

class change\_password\_pg(QDialog):

def \_\_init\_\_(self):

super(change\_password\_pg,self).\_\_init\_\_()

loadUi("change\_password\_pg.ui", self)

self.change\_password.clicked.connect(self.changepass)

self.back.clicked.connect(self.to\_prevpg)

def changepass(self):

newpassword=self.password\_line.text()

with open("username.txt",'r') as fh :

username = fh.read().strip()

cmd = "update logins set password = '"+ newpassword +"' where username = '"+username+"'"

sqlcursor.execute("use login")

sqlcursor.execute(cmd)

mydb.commit()

self.message\_label.setText("Password has been successfully changed")

def to\_prevpg(self):

widget.setCurrentIndex(13)

class set\_notice(QDialog):

def \_\_init\_\_(self):

super(set\_notice,self).\_\_init\_\_()

loadUi("set\_notice\_pg.ui", self)

self.change.clicked.connect(self.change\_notice)

self.back.clicked.connect(self.to\_prevpg)

def change\_notice(self):

with open("username.txt",'r') as fh:

username=fh.read().strip()

new\_notice = self.new\_notice.toPlainText()+"\n -"+username

with open("notice.txt",'w') as fh:

fh.write(new\_notice)

self.message\_label\_2.setText("notice changed successfully")

def to\_prevpg(self):

widget.setCurrentIndex(13)

class select\_test\_type2(QDialog):

def \_\_init\_\_(self):

super(select\_test\_type2,self).\_\_init\_\_()

loadUi("pro\_addmarks\_exam\_type2\_pg.ui", self)

self.back\_execute.clicked.connect(self.to\_prevpg)

self.teacherlogin\_exam\_monthly.clicked.connect(self.to\_month)

self.teacherlogin\_exam\_unit.clicked.connect(self.to\_uni)

self.teacherlogin\_exam\_preboard.clicked.connect(self.to\_pre)

def to\_month(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[2] = "monthly\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(25)

def to\_uni(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[2] = "unit\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(26)

def to\_pre(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[2] = "pre\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(27)

def to\_prevpg(self):

with open("classdetails.txt", "r") as fh:

cls = fh.readlines()

if cls[0] in ("class11\n", "class12\n"):

if cls[3] == "biomaths\n":

widget.setCurrentIndex(10)

if cls[3] == "computermath\n":

widget.setCurrentIndex(11)

if cls[3] == "commerce\n":

widget.setCurrentIndex(12)

elif cls[0][-2] in ("3", "4", "5", "6", "7", "8", "9", "0",) and cls[0][-3] in ("s", "1",):

widget.setCurrentIndex(8)

elif cls[0][-2] in ("1", "2") and cls[0][-3] in ("s"):

widget.setCurrentIndex(6)

class edit\_participant(QDialog):

def \_\_init\_\_(self):

super(edit\_participant,self).\_\_init\_\_()

loadUi("pro\_editpart\_main\_pg.ui", self)

self.back.clicked.connect(self.to\_prevpg)

self.editpart\_class\_1.clicked.connect(self.setclass1)

self.editpart\_class\_2.clicked.connect(self.setclass2)

self.editpart\_class\_3.clicked.connect(self.setclass3)

self.editpart\_class\_4.clicked.connect(self.setclass4)

self.editpart\_class\_5.clicked.connect(self.setclass5)

self.editpart\_class\_6.clicked.connect(self.setclass6)

self.editpart\_class\_7.clicked.connect(self.setclass7)

self.editpart\_class\_8.clicked.connect(self.setclass8)

self.editpart\_class\_9.clicked.connect(self.setclass9)

self.editpart\_class\_10.clicked.connect(self.setclass10)

self.editpart\_class\_11.clicked.connect(self.setclass11)

self.editpart\_class\_12.clicked.connect(self.setclass12)

def setclass1(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class1\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass2(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class2\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass3(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class3\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass4(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class4\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass5(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class5\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass6(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class6\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass7(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class7\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass8(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class8\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass9(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class9\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass10(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class10\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setclass11(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class11\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(18)

def setclass12(self):

with open("classdetails.txt", 'w') as fh:

clsdetails = (["class12\n", "subject\n", "test\n", "stream\n"])

fh.writelines(clsdetails)

widget.setCurrentIndex(18)

def to\_prevpg(self):

widget.setCurrentIndex(1)

class edit\_participant\_select\_stream(QDialog):

def \_\_init\_\_(self):

super(edit\_participant\_select\_stream,self).\_\_init\_\_()

loadUi("pro\_editpart\_select\_stream\_pg.ui", self)

self.back\_execute.clicked.connect(self.to\_prevpg)

self.addmarks\_stream\_biomaths.clicked.connect(self.setbiomaths)

self.addmarks\_stream\_computer.clicked.connect(self.setcomputersci)

self.addmarks\_stream\_commerce.clicked.connect(self.setcommerce)

def setbiomaths(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[3] = "biomaths\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setcomputersci(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[3] = "computermath\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def setcommerce(self):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

clsdetails[3] = "commerce\n"

with open("classdetails.txt", 'w') as fh:

fh.writelines(clsdetails)

widget.setCurrentIndex(19)

def to\_prevpg(self):

widget.setCurrentIndex(17)

class edit\_participant\_select\_table(QDialog):

def \_\_init\_\_(self):

super(edit\_participant\_select\_table,self).\_\_init\_\_()

loadUi("pro\_editpart\_select\_table\_pg.ui", self)

self.back.clicked.connect(self.to\_prevpg)

self.editpart\_table\_teacher.clicked.connect(self.to\_teachertable)

self.editpart\_table\_student.clicked.connect(self.to\_studenttable)

def to\_teachertable(self):

widget.setCurrentIndex(20)

def to\_studenttable(self):

widget.setCurrentIndex(22)

def to\_prevpg(self):

with open("classdetails.txt", "r") as fh:

cls = fh.readlines()

if cls[0] in ("class11\n", "class12\n"):

widget.setCurrentIndex(18)

elif cls[0][-2] in ("3", "4", "5", "6", "7", "8", "9", "0",) and cls[0][-3] in ("s", "1",):

widget.setCurrentIndex(17)

elif cls[0][-2] in ("1", "2") and cls[0][-3] in ("s"):

widget.setCurrentIndex(17)

class edit\_participant\_teacher\_table(QDialog):

def \_\_init\_\_(self):

super(edit\_participant\_teacher\_table,self).\_\_init\_\_()

loadUi("pro\_editpart\_teacher\_table\_pg.ui", self)

self.back.clicked.connect(self.to\_prevpg)

self.refresh1.clicked.connect(self.refresh)

self.submit.clicked.connect(self.add\_to\_db)

def add\_to\_db(self):

rw1 = []

rw2 = []

rwcount = self.tableWidget.rowCount()

for rw\_no in range(rwcount):

rw\_item = self.tableWidget.item(rw\_no, 0).text()

clm\_item = self.tableWidget.item(rw\_no, 1).text()

rw1.append(rw\_item)

rw2.append(clm\_item)

with open("currentyear.txt",'r') as fh:

curntyear = fh.read().strip()

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

if clsdetails[0] =="class1\n":

tablename = curntyear +"\_class1\_teacher\_table"

if clsdetails[0] =="class2\n":

tablename = curntyear +"\_class2\_teacher\_table"

if clsdetails[0] =="class3\n":

tablename = curntyear +"\_class3\_teacher\_table"

if clsdetails[0] =="class4\n":

tablename = curntyear +"\_class4\_teacher\_table"

if clsdetails[0] =="class5\n":

tablename = curntyear +"\_class5\_teacher\_table"

if clsdetails[0] =="class6\n":

tablename = curntyear +"\_class6\_teacher\_table"

if clsdetails[0] =="class7\n":

tablename = curntyear +"\_class7\_teacher\_table"

if clsdetails[0] =="class8\n":

tablename = curntyear + "\_class8\_teacher\_table"

if clsdetails[0] =="class9\n":

tablename = curntyear +"\_class9\_teacher\_table"

if clsdetails[0] =="class10\n":

tablename = curntyear + "\_class10\_teacher\_table"

if clsdetails[0] =="class11\n":

if clsdetails[3] == "biomaths\n":

tablename = curntyear + "\_class11\_biomaths\_teacher\_table"

elif clsdetails[3] == "computermath\n":

tablename = curntyear + "\_class11\_teacher\_compmaths\_table"

elif clsdetails[3] == "commerce\n":

tablename = curntyear + "\_class11\_teacher\_commerce\_table"

if clsdetails[0] =="class12\n":

if clsdetails[3] == "biomaths\n":

tablename = curntyear + "\_class12\_biomaths\_teacher\_table"

elif clsdetails[3] == "computermath\n":

tablename = curntyear + "\_class12\_teacher\_compmaths\_table"

elif clsdetails[3] == "commerce\n":

tablename = curntyear + "\_class12\_teacher\_commerce\_table"

cmd = "use "+curntyear

sqlcursor.execute(cmd)

sqlcursor.execute("show tables")

qury = sqlcursor.fetchall()

for item in qury:

if item[0] == tablename:

cmd = "delete from " + tablename

sqlcursor.execute(cmd)

mydb.commit()

break

else:

cmd = "create table "+tablename+"( role varchar(50) , username varchar(50))"

sqlcursor.execute(cmd)

for i in range(len(rw1)):

cmd = "insert into " + tablename + "(role, username) values('{}','{}')".format(rw1[i], rw2[i])

sqlcursor.execute(cmd)

mydb.commit()

self.message\_label.setStyleSheet("color: rgb(0,170, 0);font: 12pt 'Arial';")

self.message\_label.setText("The table has been saved")

def refresh(self):

with open("classdetails.txt", "r") as fh:

cls = fh.readlines()

if cls[0] in ("class11\n", "class12\n"):

if cls[3] == "biomaths\n":

clsinfo = ["class teacher", "english", "biology", "physics", "chemistry", "maths"]

elif cls[3] == "computermath\n":

clsinfo = ["class teacher", "english", "computer", "physics", "chemistry", "maths"]

elif cls[3] == "commerce\n":

clsinfo = ["class teacher", "english", "computer", "accounts", "buissness", "economics"]

elif cls[0][-2] in ("3", "4", "5", "6", "7", "8", "9", "0",) and cls[0][-3] in ("s", "1",) and cls[3] == "stream\n":

clsinfo = ["class teacher", "english", "social", "science", "maths", "tamil", "hindi", "computer"]

elif cls[0][-2] in ("1", "2") and cls[0][-3] in ("s") and cls[3] == "stream\n":

clsinfo = ["class teacher", "english", "evs", "maths", "tamil", "hindi", "computer"]

self.tableWidget.setRowCount(len(clsinfo))

rowcount = 0

for item in clsinfo:

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(item))

rowcount += 1

def to\_prevpg(self):

widget.setCurrentIndex(19)

class pro\_set\_year\_pg(QDialog):

def \_\_init\_\_(self):

super(pro\_set\_year\_pg,self).\_\_init\_\_()

loadUi("pro\_set\_year\_pg.ui", self)

self.setyear\_execute.clicked.connect(self.ex\_setyear)

self.setyear\_backpg\_execute.clicked.connect(self.to\_prevpg)

def ex\_setyear(self):

year = self.setyear\_year.text()

curntyear = "year"+year

sqlcursor.execute("show databases")

qury = sqlcursor.fetchall()

for item in qury:

if item[0] == curntyear:

with open("currentyear.txt", 'w') as fh:

inf = "year" + year

fh.write(inf)

self.message\_label.setStyleSheet("color: rgb(0,170, 0);font: 12pt 'Arial';")

self.message\_label.setText("The Current year has been changed")

break

else:

cmd = "create database year" + year

sqlcursor.execute(cmd)

mydb.commit()

with open("currentyear.txt", 'w') as fh:

inf = "year" + year

fh.write(inf)

self.message\_label.setStyleSheet("color: rgb(0,170, 0);font: 12pt 'Arial';")

self.message\_label.setText("The Current year has been changed")

def to\_prevpg(self):

widget.setCurrentIndex(1)

class pro\_editpart\_student\_no\_pg(QDialog):

def \_\_init\_\_(self):

super(pro\_editpart\_student\_no\_pg,self).\_\_init\_\_()

loadUi("pro\_editpart\_no\_of\_student\_pg.ui", self)

self.enter\_exe.clicked.connect(self.getvalues)

self.back.clicked.connect(self.to\_prevpg)

def getvalues(self):

no\_students = self.no\_of\_students.text()

with open("no\_of\_students.txt",'w') as fh:

fh.write(no\_students)

widget.setCurrentIndex(23)

def to\_prevpg(self):

widget.setCurrentIndex(19)

class pro\_editpart\_student\_table\_pg(QDialog):

def \_\_init\_\_(self):

super(pro\_editpart\_student\_table\_pg,self).\_\_init\_\_()

loadUi("pro\_editpart\_student\_table\_pg.ui", self)

self.back.clicked.connect(self.to\_prevpg)

self.refresh.clicked.connect(self.to\_refresh)

self.submit.clicked.connect(self.add\_to\_db)

def add\_to\_db(self):

rw1 = []

rw2 = []

rwcount = self.tableWidget.rowCount()

for rw\_no in range(rwcount):

rw\_item = self.tableWidget.item(rw\_no, 0).text()

clm\_item = self.tableWidget.item(rw\_no, 1).text()

rw1.append(rw\_item)

rw2.append(clm\_item)

with open("currentyear.txt", 'r') as fh:

curntyear = fh.read().strip()

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

if clsdetails[0] == "class1\n":

tablename = curntyear + "\_class1\_student\_table"

if clsdetails[0] == "class2\n":

tablename = curntyear + "\_class2\_student\_table"

if clsdetails[0] == "class3\n":

tablename = curntyear + "\_class3\_student\_table"

if clsdetails[0] == "class4\n":

tablename = curntyear + "\_class4\_student\_table"

if clsdetails[0] == "class5\n":

tablename = curntyear + "\_class5\_student\_table"

if clsdetails[0] == "class6\n":

tablename = curntyear + "\_class6\_student\_table"

if clsdetails[0] == "class7\n":

tablename = curntyear + "\_class7\_student\_table"

if clsdetails[0] == "class8\n":

tablename = curntyear + "\_class8\_student\_table"

if clsdetails[0] == "class9\n":

tablename = curntyear + "\_class9\_student\_table"

if clsdetails[0] == "class10\n":

tablename = curntyear + "\_class10\_student\_table"

if clsdetails[0] == "class11\n":

if clsdetails[3] == "biomaths\n":

tablename = curntyear + "\_class11\_biomaths\_student\_table"

elif clsdetails[3] == "computermath\n":

tablename = curntyear + "\_class11\_teacher\_student\_table"

elif clsdetails[3] == "commerce\n":

tablename = curntyear + "\_class11\_teacher\_student\_table"

if clsdetails[0] == "class12\n":

if clsdetails[3] == "biomaths\n":

tablename = curntyear + "\_class12\_biomaths\_student\_table"

elif clsdetails[3] == "computermath\n":

tablename = curntyear + "\_class12\_teacher\_student\_table"

elif clsdetails[3] == "commerce\n":

tablename = curntyear + "\_class12\_teacher\_student\_table"

cmd = "use " + curntyear

sqlcursor.execute(cmd)

sqlcursor.execute("show tables")

qury = sqlcursor.fetchall()

for item in qury:

if item[0] == tablename:

cmd = "delete from " + tablename

sqlcursor.execute(cmd)

mydb.commit()

break

else:

cmd = "create table " + tablename + "( rollno varchar(50) , username varchar(50))"

sqlcursor.execute(cmd)

print("hi")

for i in range(len(rw1)):

cmd = "insert into " + tablename + "( rollno , username) values('{}','{}')".format(rw1[i], rw2[i])

sqlcursor.execute(cmd)

mydb.commit()

self.message\_label.setStyleSheet("color: rgb(0,170, 0);font: 12pt 'Arial';")

self.message\_label.setText("The table has been saved")

def to\_refresh(self):

with open("no\_of\_students.txt", "r") as fh:

no\_stu = int(fh.read().strip())

self.tableWidget.setRowCount(no\_stu)

rowcount = 0

for i in range(1,(no\_stu+1)):

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

if clsdetails[0] == "class1\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(100+i)))

rowcount += 1

if clsdetails[0] == "class2\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(200+i)))

rowcount += 1

if clsdetails[0] == "class3\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(300+i)))

rowcount += 1

if clsdetails[0] == "class4\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(400+i)))

rowcount += 1

if clsdetails[0] == "class5\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(500+i)))

rowcount += 1

if clsdetails[0] == "class6\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(600+i)))

rowcount += 1

if clsdetails[0] == "class7\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(700+i)))

rowcount += 1

if clsdetails[0] == "class8\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(800+i)))

rowcount += 1

if clsdetails[0] == "class9\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(900+i)))

rowcount += 1

if clsdetails[0] == "class10\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(1000+i)))

rowcount += 1

if clsdetails[0] == "class11\n":

if clsdetails[3] == "biomaths\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(1100+i)+"bm"))

rowcount += 1

elif clsdetails[3] == "computermath\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(1100+i)+"cm"))

rowcount += 1

elif clsdetails[3] == "commerce\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(1100+i)+"cc"))

rowcount += 1

if clsdetails[0] == "class12\n":

if clsdetails[3] == "biomaths\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(1200+i)+"bm"))

rowcount += 1

elif clsdetails[3] == "computermath\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(1200+i)+"cm"))

rowcount += 1

elif clsdetails[3] == "commerce\n":

self.tableWidget.setItem(rowcount, 0, QtWidgets.QTableWidgetItem(str(1200+i)+"cc"))

rowcount += 1

def to\_prevpg(self):

widget.setCurrentIndex(19)

class pro\_addmarks\_student\_table\_pg(QDialog):

def \_\_init\_\_(self):

super(pro\_addmarks\_student\_table\_pg,self).\_\_init\_\_()

loadUi("pro\_addmarks\_student\_table\_pg.ui", self)

self.back.clicked.connect(self.to\_prevpg)

self.refresh.clicked.connect(self.to\_refresh)

self.submit.clicked.connect(self.add\_to\_db)

def add\_to\_db(self):

rw1 = []

rw2 = []

rw3 = []

rwcount = self.tableWidget.rowCount()

for rw\_no in range(rwcount):

rw\_item = self.tableWidget.item(rw\_no, 0).text()

rw2\_item = self.tableWidget.item(rw\_no, 1).text()

rw3\_item = self.tableWidget.item(rw\_no, 3).text()

rw1.append(rw\_item)

rw2.append(rw2\_item)

rw3.append(rw3\_item)

with open("currentyear.txt", 'r') as fh:

curntyear = fh.read().strip()

dat\_typ\_1 = {"english\n": "english", "evs\n": "evs", "maths\n": "maths", "lang2\n": "lang2","computer\n": "computer"}

dat\_typ\_2 = {"english\n": "english","science\n": "science","social\n": "social","lang2\n": "language","maths\n": "maths", "computer\n": "computer"}

dat\_typ\_3 = {"english\n": "english","biology\n": "biology","physics\n": "physics","chemistry\n": "chemistry", "maths\n": "maths"}

dat\_typ\_3 = {"english\n": "english","computer\n": "computer","accounts\n": "accounts","economics\n":"economics", "business\n": "business"}

dat\_typ\_3 = {"english\n": "english","computer\n": "computer","physics\n": "physics","chemistry\n":"chemistry", "maths\n": "maths"}

test\_typ1 = {"periodictest1\n": "periodic\_test\_1", "periodictest2\n": "periodic\_test\_2", "periodictest3\n":"periodic\_test\_3", "periodictest4\n":"periodic\_test\_4", "halfyearly\n":"halfyearly", "annual\n":"annual"}

test\_typ2 = {"monthly\n":"monthly","unit\n":"unit","pre\n":"pre"}

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

if clsdetails[0] == "class1\n":

if clsdetails[1] == "english\n":

pass

if clsdetails[0] == "class2\n":

tablename = curntyear + "\_class2\_"

if clsdetails[0] == "class3\n":

tablename = curntyear + "\_class3\_"

if clsdetails[0] == "class4\n":

tablename = curntyear + "\_class4\_"

if clsdetails[0] == "class5\n":

tablename = curntyear + "\_class5\_"

if clsdetails[0] == "class6\n":

tablename = curntyear + "\_class6\_"

if clsdetails[0] == "class7\n":

tablename = curntyear + "\_class7\_"

if clsdetails[0] == "class8\n":

tablename = curntyear + "\_class8\_"

if clsdetails[0] == "class9\n":

tablename = curntyear + "\_class9\_"

if clsdetails[0] == "class10\n":

tablename = curntyear + "\_class10\_"

if clsdetails[0] == "class11\n":

if clsdetails[3] == "biomaths\n":

tablename = curntyear + "\_class11\_biomaths\_"

elif clsdetails[3] == "computermath\n":

tablename = curntyear + "\_class11\_compmaths\_"

elif clsdetails[3] == "commerce\n":

tablename = curntyear + "\_class11\_commerce\_"

if clsdetails[0] == "class12\n":

if clsdetails[3] == "biomaths\n":

tablename = curntyear + "\_class12\_biomaths\_"

elif clsdetails[3] == "computermath\n":

tablename = curntyear + "\_class12\_compmaths\_"

elif clsdetails[3] == "commerce\n":

tablename = curntyear + "\_class12\_commerce\_"

if clsdetails[0] in ("class12\n"):

if clsdetails[3] == "biomaths\n":

tablename = tablename +dat\_typ\_3[clsdetails[2]+"\_"+test\_typ2[clsdetails]]

elif clsdetails[3] == "computermath\n":

clsinfo = ["class teacher", "english", "computer", "physics", "chemistry", "maths"]

elif clsdetails[3] == "commerce\n":

clsinfo = ["class teacher", "english", "computer", "accounts", "buissness", "economics"]

elif clsdetails[0][-2] in ("3", "4", "5", "6", "7", "8", "9", "0",) and clsdetails[0][-3] in ("s", "1",) and clsdetails[3] == "stream\n":

clsinfo = ["class teacher", "english", "social", "science", "maths", "tamil", "hindi", "computer"]

elif clsdetails[0][-2] in ("1", "2") and clsdetails[0][-3] in ("s") and clsdetails[3] == "stream\n":

clsinfo = ["class teacher", "english", "evs", "maths", "tamil", "hindi", "computer"]

cmd = "use " + curntyear

sqlcursor.execute(cmd)

sqlcursor.execute("show tables")

qury = sqlcursor.fetchall()

for item in qury:

if item[0] == tablename:

cmd = "delete from " + tablename

sqlcursor.execute(cmd)

mydb.commit()

break

else:

cmd = "create table " + tablename + "( rollno varchar(50) , username varchar(50))"

sqlcursor.execute(cmd)

print("hi")

for i in range(len(rw1)):

cmd = "insert into " + tablename + "( rollno , username) values('{}','{}')".format(rw1[i], rw2[i])

sqlcursor.execute(cmd)

mydb.commit()

self.message\_label.setStyleSheet("color: rgb(0,170, 0);font: 12pt 'Arial';")

self.message\_label.setText("The table has been saved")

def to\_refresh(self):

with open("currentyear.txt",'r') as fh:

curntyear = fh.read().strip()

cmd = "use " + curntyear

sqlcursor.execute(cmd)

with open("classdetails.txt", 'r') as fh:

clsdetails = fh.readlines()

if clsdetails[0] == "class1\n":

tablename = curntyear + "\_class1\_student\_table"

if clsdetails[0] == "class2\n":

tablename = curntyear + "\_class2\_student\_table"

if clsdetails[0] == "class3\n":

tablename = curntyear + "\_class3\_student\_table"

if clsdetails[0] == "class4\n":

tablename = curntyear + "\_class4\_student\_table"

if clsdetails[0] == "class5\n":

tablename = curntyear + "\_class5\_student\_table"

if clsdetails[0] == "class6\n":

tablename = curntyear + "\_class6\_student\_table"

if clsdetails[0] == "class7\n":

tablename = curntyear + "\_class7\_student\_table"

if clsdetails[0] == "class8\n":

tablename = curntyear + "\_class8\_student\_table"

if clsdetails[0] == "class9\n":

tablename = curntyear + "\_class9\_student\_table"

if clsdetails[0] == "class10\n":

tablename = curntyear + "\_class10\_student\_table"

if clsdetails[0] == "class11\n":

if clsdetails[3] == "biomaths\n":

tablename = curntyear + "\_class11\_biomaths\_student\_table"

elif clsdetails[3] == "computermath\n":

tablename = curntyear + "\_class11\_compmaths\_student\_table"

elif clsdetails[3] == "commerce\n":

tablename = curntyear + "\_class11\_commerce\_student\_table"

if clsdetails[0] == "class12\n":

if clsdetails[3] == "biomaths\n":

tablename = curntyear + "\_class12\_biomaths\_student\_table"

elif clsdetails[3] == "computermath\n":

tablename = curntyear + "\_class12\_compmaths\_student\_table"

elif clsdetails[3] == "commerce\n":

tablename = curntyear + "\_class12\_commerce\_student\_table"

cmd = "select \* from "+tablename

sqlcursor.execute(cmd)

val = sqlcursor.fetchall()

self.tableWidget.setRowCount(len(val))

for rwcnt in range(len(val)):

for clmcnt in range(2):

self.tableWidget.setItem(rwcnt, clmcnt, QtWidgets.QTableWidgetItem(val[rwcnt][clmcnt]))

def to\_prevpg(self):

with open("classdetails.txt",'r') as fh:

val = fh.readlines()

if val[0] in ("class10\n","class12\n"):

widget.setCurrentIndex(16)

else:

widget.setCurrentIndex(7)

class name\_collector\_for\_month(QDialog):

def \_\_init\_\_(self):

super(name\_collector\_for\_month, self).\_\_init\_\_()

loadUi("name\_collector\_for\_month.ui", self)

self.enter\_exe.clicked.connect(self.up\_values)

self.back.clicked.connect(self.to\_prevpg)

def up\_values(self):

val = self.name.text()

with open("month\_name.txt",'w') as fh:

fh.write(val)

widget.setCurrentIndex(24)

def to\_prevpg(self):

widget.setCurrentIndex(16)

class name\_collector\_for\_unit(QDialog):

def \_\_init\_\_(self):

super(name\_collector\_for\_unit, self).\_\_init\_\_()

loadUi("name\_collector\_for\_unit.ui", self)

self.enter\_exe.clicked.connect(self.up\_values)

self.back.clicked.connect(self.to\_prevpg)

def up\_values(self):

val = self.name.text()

with open("month\_name.txt",'w') as fh:

fh.write(val)

widget.setCurrentIndex(24)

def to\_prevpg(self):

widget.setCurrentIndex(16)

class name\_collector\_for\_pre(QDialog):

def \_\_init\_\_(self):

super(name\_collector\_for\_pre, self).\_\_init\_\_()

loadUi("name\_collector\_for\_pre.ui", self)

self.enter\_exe.clicked.connect(self.up\_values)

self.back.clicked.connect(self.to\_prevpg)

def up\_values(self):

val = self.name.text()

with open("month\_name.txt",'w') as fh:

fh.write(val)

widget.setCurrentIndex(24)

def to\_prevpg(self):

widget.setCurrentIndex(16)

with open("notice.txt",'r') as fh:

line = fh.readlines()

if len(line) == 0:

with open("notice.txt",'w') as fh1:

data = '''Welcome to mark register

the notice will be soon added once someone adds it'''

fh1.write(data)

app = QApplication(sys.argv)

widget = QtWidgets.QStackedWidget()

mainwindow = login\_pg()

master\_main\_pg = master\_main\_pg()

add\_participant = add\_participant()

remove\_participant = remove\_participant()

staff\_login\_main\_pg = staff\_login\_main\_pg()

addmarks\_selectclass = addmarks\_selectclass()

addmarks\_selectsubject\_12 = addmarks\_selectsubject\_12()

addmarks\_selectsubject\_345678910 = addmarks\_selectsubject\_345678910()

select\_test\_typ1\_pg = select\_test\_typ1\_pg()

addmarks\_select\_stream = addmarks\_select\_stream()

select\_subject\_biomaths\_pg = select\_subject\_biomaths\_pg()

select\_subject\_computersci\_pg = select\_subject\_computersci\_pg()

select\_subject\_commerce\_pg = select\_subject\_commerce\_pg()

additional\_settings\_pg = additional\_settings\_pg()

change\_password\_pg = change\_password\_pg()

set\_notice = set\_notice()

select\_test\_type2 = select\_test\_type2()

edit\_participant = edit\_participant()

edit\_participant\_select\_stream = edit\_participant\_select\_stream()

edit\_participant\_select\_table = edit\_participant\_select\_table()

edit\_participant\_teacher\_table = edit\_participant\_teacher\_table()

pro\_set\_year\_pg = pro\_set\_year\_pg()

pro\_editpart\_student\_no\_pg = pro\_editpart\_student\_no\_pg()

pro\_editpart\_student\_table\_pg = pro\_editpart\_student\_table\_pg()

pro\_addmarks\_student\_table\_pg = pro\_addmarks\_student\_table\_pg()

name\_collector\_for\_month = name\_collector\_for\_month()

name\_collector\_for\_unit = name\_collector\_for\_unit()

name\_collector\_for\_pre = name\_collector\_for\_pre()

widget.addWidget(mainwindow)# index = 0

widget.addWidget(master\_main\_pg)# index = 1

widget.addWidget(add\_participant)# index = 2

widget.addWidget(remove\_participant)# index = 3

widget.addWidget(staff\_login\_main\_pg)# index = 4

widget.addWidget(addmarks\_selectclass)# index = 5

widget.addWidget(addmarks\_selectsubject\_12)# index = 6

widget.addWidget(select\_test\_typ1\_pg)# index = 7

widget.addWidget(addmarks\_selectsubject\_345678910)# index = 8

widget.addWidget(addmarks\_select\_stream)# index = 9

widget.addWidget(select\_subject\_biomaths\_pg)# index = 10

widget.addWidget(select\_subject\_computersci\_pg)# index = 11

widget.addWidget(select\_subject\_commerce\_pg)# index = 12

widget.addWidget(additional\_settings\_pg)# index = 13

widget.addWidget(change\_password\_pg)# index = 14

widget.addWidget(set\_notice)# index = 15

widget.addWidget(select\_test\_type2)# index = 16

widget.addWidget(edit\_participant)# index = 17

widget.addWidget(edit\_participant\_select\_stream)# index = 18

widget.addWidget(edit\_participant\_select\_table)# index = 19

widget.addWidget(edit\_participant\_teacher\_table)# index = 20

widget.addWidget(pro\_set\_year\_pg)# index = 21

widget.addWidget(pro\_editpart\_student\_no\_pg)# index = 22

widget.addWidget(pro\_editpart\_student\_table\_pg)# index = 23

widget.addWidget(pro\_addmarks\_student\_table\_pg)# index = 24

widget.addWidget(name\_collector\_for\_month)# index = 25

widget.addWidget(name\_collector\_for\_unit)# index = 26

widget.addWidget(name\_collector\_for\_pre)# index = 27

widget.show()

try:

sys.exit(app.exec\_())

except:

print("exiting")

with open("classdetails.txt", "r") as fh:

print(fh.read())