UNIVERSITI TEKNOLOGI MARA

KEDAH BRANCH

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS (CDIM144)

IML208: PROGRAMMING FOR LIBRARIES

INDIVIDUAL PROJECT:

STUDENT'S MARK CALCULATION SYSTEM

Prepared by:

HANI RASYIQAH IRDINA BT HASSANAL ZAIRI

2022863864

GROUP KCDIM1443F

Prepared for:

MR. AIRUL SHAZWAN BIN NORSHAHIMI

Submission date:

4TH JANUARY 2024

INDIVIDUAL PROJECT:

STUDENT'S MARK CALCULATION SYSTEM

PREPARED BY:

HANI RASYIQAH IRDINA BT HASSANAL ZAIRI

2022863864

GROUP KCDIM1443F

CDIM144 – DIPLOMA IN LIBRARY INFORMATICS

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA (UITM)

KEDAH BRANCH

TABLE OF CONTENT

1.0 INTRODUCTION

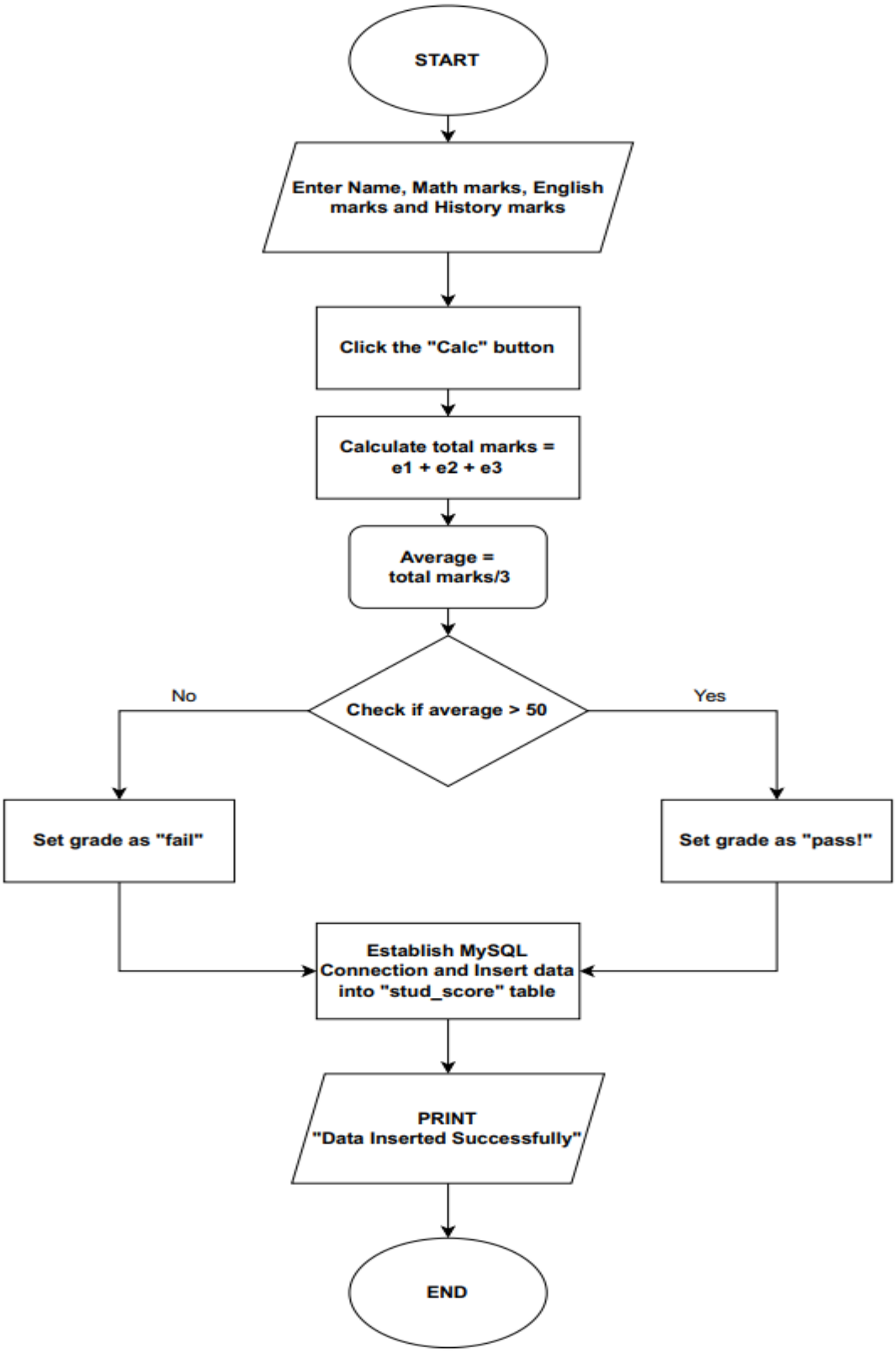
The Student Mark Calculation System is a simple Python program application that built using Tkinter for graphical user interface (GUI) and MySQL Connector to link the pyhton program with database. This system needs users to enter a student's details and grades for various subjects such as Mathematics, English, and History marks and the system will compute the overall marks, average, and grade. Then, the data will be stored in a MySQL database.

The GUI includes input fields for the student's name and marks in three subjects which are Math, English, and History. Labels effectively direct the user by displaying the purpose of each input field (e.g., Name, Math Marks). Next, based on the user's input, the total marks of the three subjects, average, and grade are dynamically displayed on the interface. The processes of calculation and storage are initiated by a "Calc" button. The average calculation process is calculated using the formula of adding all the total marks and dividing by three because there are three subject marks used. If the average marks exceed 50, the system will set a grade as "pass" while if the average is less than 50, the system will set a grade as "fail".

So, for the database interaction, after establishing a connection to the MySQL database, the system generates an interactive cursor. After that, the data is entered into the "stud_score" table, which contains the student's name, subject score, total score, average and grade. After entering the data into the stud_score table in MySQL Connection, the system will print "Data Inserted Successfully". Therefore, database interactions are managed through the implementation of appropriate error handling, and appropriate messages are printed to the console.

To conclude, the structure for organizing student data, computing marks, and storing data in a MySQL database is provided by the Student Marks Calculation System. It acts as a basis for additional work and improvements to meet specific user needs and raise the standard of usability around.

2.0 FLOWCHART



3.0 PYTHON CODE

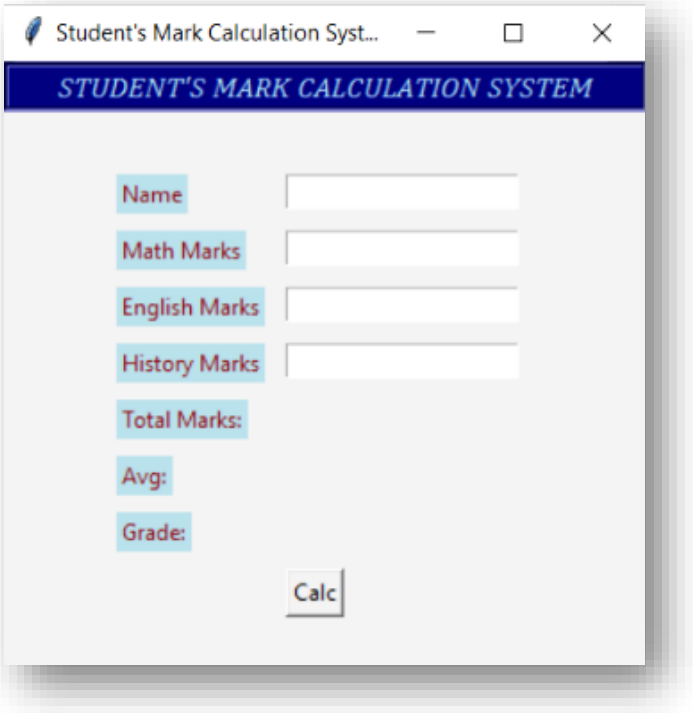
C: > Users > user > Downloads > UITM PART 3 > hanirasyiqah.py > StudentMarkCalculationSystem > __init__

```python
import tkinter as tk
import mysql.connector

class StudentMarkCalculationSystem:
    def __init__(self, master):
        self.master = master
        self.master.title("Student's Mark Calculation System")
        self.name = tk.Entry(master)
        self.e1 = tk.Entry(master)
        self.e2 = tk.Entry(master)
        self.e3 = tk.Entry(master)
        self.totText = tk.StringVar()
        self.avgText = tk.StringVar()
        self.gradeText = tk.StringVar()

        tk.Label(master, text="Name", fg="firebrick4", bg="light blue").place(x=60, y=60)
        tk.Label(master, text="Math Marks",  fg="firebrick4", bg="light blue").place(x=60, y=90)
        tk.Label(master, text="English Marks",  fg="firebrick4", bg="light blue").place(x=60, y=120)
        tk.Label(master, text="History Marks",  fg="firebrick4", bg="light blue").place(x=60, y=150)
        tk.Label(master, text="Total Marks:",  fg="firebrick4", bg="light blue").place(x=60, y=180)
        tk.Label(master, text="Avg:",  fg="firebrick4", bg="light blue").place(x=60, y=210)
        tk.Label(master, text="Grade:",  fg="firebrick4", bg="light blue").place(x=60, y=240)

        self.name.place(x=150, y=60)
        self.e1.place(x=150, y=90)
        self.e2.place(x=150, y=120)
        self.e3.place(x=150, y=150)

        result = tk.Label(master, text="", textvariable=self.totText).place(x=150, y=180)
        average = tk.Label(master, text="", textvariable=self.avgText).place(x=150, y=210)
        grade = tk.Label(master, text="", textvariable=self.gradeText).place(x=150, y=240)

        tk.Button(master, text="Calc", height=1, width=3, command=self.Ok).place(x=150, y=270)

    def Ok(self):
        result = int(self.e1.get()) + int(self.e2.get()) + int(self.e3.get())

        self.totText.set(result)

        average = result/3
        self.avgText.set(average)

        if (average > 50):
            grade = "pass!"
        else:
            grade = "fail"

        self.gradeText.set(grade)

        # Establish a connection to the MySQL server
        mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            database="student_marks"
        )

        # Create a cursor object to interact with the database
        cursor = mydb.cursor()

        # Inserting data into a table
        sql = "INSERT INTO `stud_score` (stud_name, math_score, english_score, history_score, total_marks, average, grade) VALUES (%s, %s, %s, %s, %s, %s, %s)"
        val = (self.name.get(), self.e1.get(), self.e2.get(), self.e3.get(), self.totText.get(), self.avgText.get(), self.gradeText.get())

        try:
            cursor.execute(sql, val)
            mydb.commit()
            print("Data inserted successfully!")
        except mysql.connector.Error as err:
            print(f"Error: {err}")
            mydb.rollback()
```

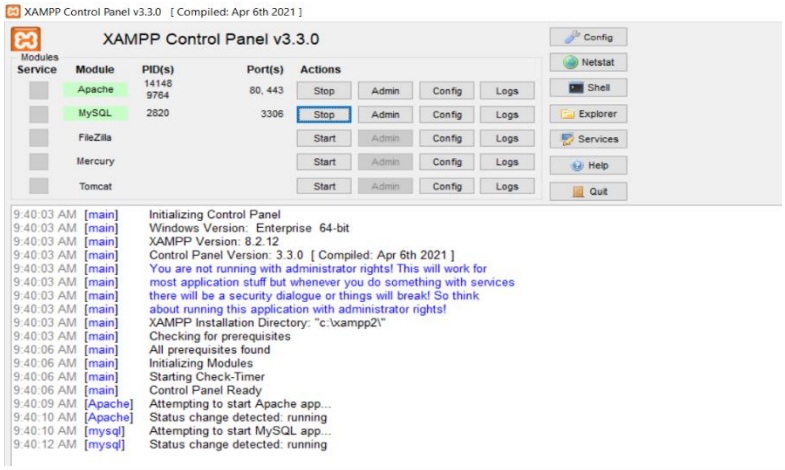
C: > Users > user > Downloads > UITM PART 3 > hanirasyiqah.py > StudentMarkCalculationSystem > Ok

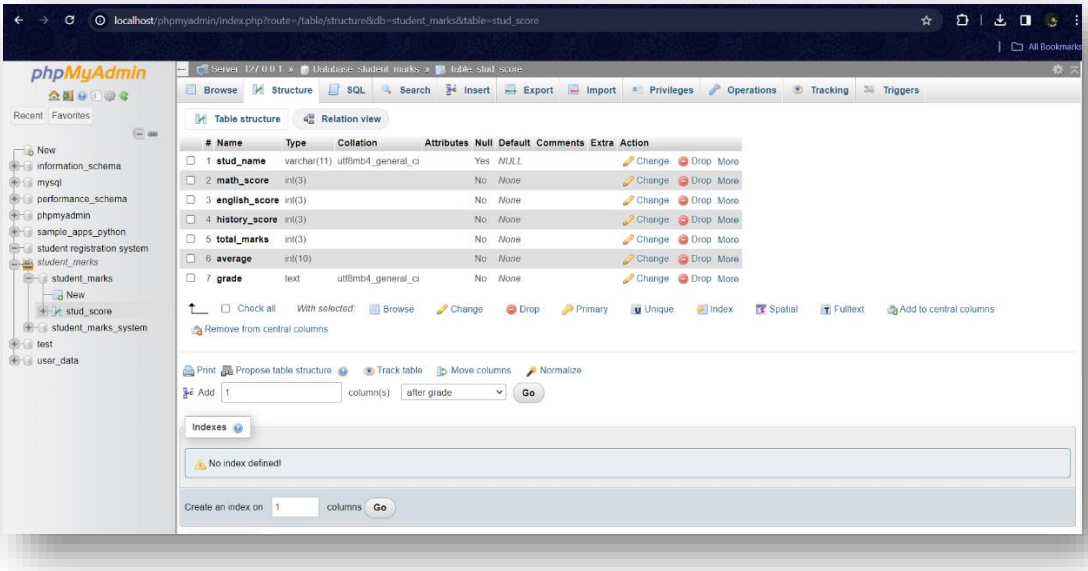
```python
        cursor.close()
        mydb.close()


if __name__ == "__main__":
    root = tk.Tk()
    app = StudentMarkCalculationSystem(root)
    root.geometry("300x300")
    label= tk.Label(root,text= "STUDENT'S MARK CALCULATION SYSTEM",font=("Cambria", 12, "italic"), bg= "navy blue",
                fg="light blue", bd=3, relief="groove")
    label.pack(ipadx=400)
    root.mainloop()
```

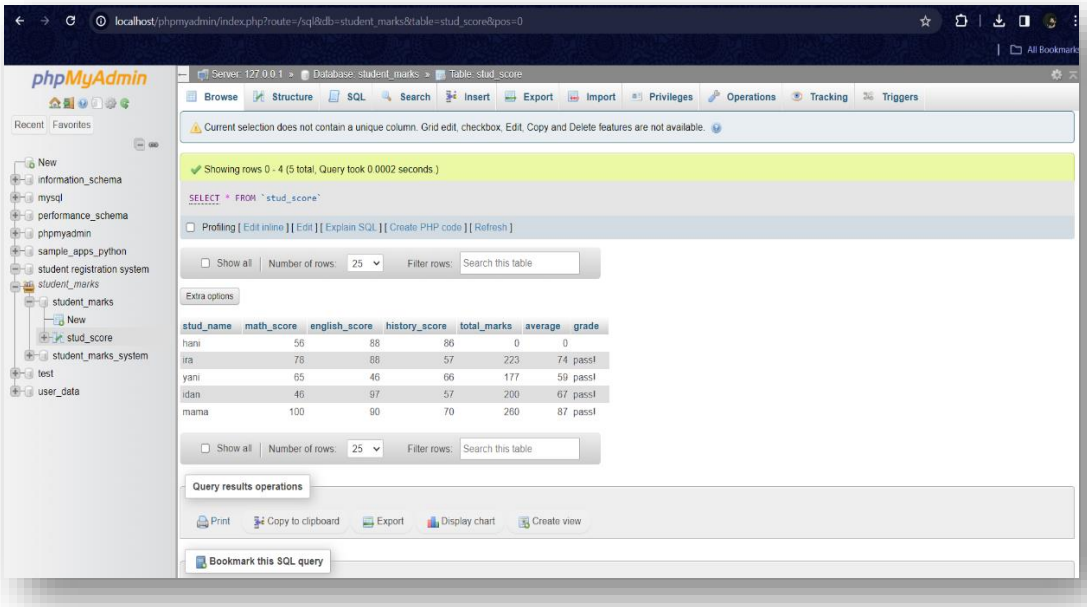
4.0 GRAPHICAL USER INTERFACES (GUI)



5.0 DATABASE


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Service Module PID(s) Port(s) Actions
Apache 14148 9764 80, 443 Stop Admin Config Logs
MySQL 2820 3306 Stop Admin Config Logs
FileZilla Start Admin Config Logs
Mercury Start Admin Config Logs
Tomcat Start Admin Config Logs
Config
Netstat
Shell
Explorer
Services
Help
Quit
9:40:03 AM [main] Initializing Control Panel
9:40:03 AM [main] Windows Version: Enterprise 64-bit
9:40:03 AM [main] XAMPP Version: 8.2.12
9:40:03 AM [main] Control Panel Version: 3.3.0 [Compiled: Apr 6th 2021]
9:40:03 AM [main] You are not running with administrator rights! This will work for
9:40:03 AM [main] most application stuff but whenever you do something with services
9:40:03 AM [main] there will be a security dialogue or things will break! So think
9:40:03 AM [main] about running this application with administrator rights!
9:40:03 AM [main] XAMPP Installation Directory: "c:\xampp2\"
9:40:03 AM [main] Checking for prerequisites
9:40:06 AM [main] All prerequisites found
9:40:06 AM [main] Initializing Modules
9:40:06 AM [main] Starting Check-Timer
9:40:06 AM [main] Control Panel Ready
9:40:09 AM [Apache] Attempting to start Apache app...
9:40:10 AM [Apache] Status change detected: running
9:40:10 AM [mysql] Attempting to start MySQL app...
9:40:12 AM [mysql] Status change detected: running


localhost/phpmyadmin/index.php?route=/table/structure&db=student_marks&table=stud_score
phpMyAdmin
Server: 127.0.0.1 » Database: student_marks » Table: stud_score
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Table structure Relation view
Name Type Collation Attributes Null Default Comments Extra Action
1 stud_name varchar(11) utf8mb4_general_ci Yes NULL Change Drop More
2 math_score int(3) No None Change Drop More
3 english_score int(3) No None Change Drop More
4 history_score int(3) No None Change Drop More
5 total_marks int(3) No None Change Drop More
6 average int(10) No None Change Drop More
7 grade text utf8mb4_general_ci No None Change Drop More
Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext Add to central columns
Remove from central columns
Print Propose table structure Track table Move columns Normalize
Add 1 column(s) after grade Go
Indexes
No index defined!
Create an index on 1 columns Go


localhost/phpmyadmin/index.php?route=/sql&db=student_marks&table=stud_score&pos=0
phpMyAdmin
Server: 127.0.0.1 » Database: student_marks » Table: stud_score
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.
Showing rows 0 - 4 (5 total, Query took 0.0002 seconds.)
SELECT * FROM `stud_score`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all Number of rows: 25 Filter rows: Search this table
Extra options
stud_name math_score english_score history_score total_marks average grade
hani 56 88 86 0 0
ira 78 88 57 223 74 pass!
yani 65 46 66 177 59 pass!
idan 46 97 57 200 67 pass!
mama 100 90 70 260 87 pass!
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query

6.0 CONCLUSION

In conclusion, the Student Marks Calculation System is a well-designed Python program with a graphical user interface (GUI) built using Tkinter and integrated with a MySQL database via the MySQL Connector. The system efficiently handles user input for student details and grades in three subjects, performs real-time calculations for total marks, averages, and grades, and further stores the data in a MySQL database table named "stud_score." To guarantee accurate data entry into the database, the system employs suitable error handling during database interactions. System reliability is enhanced by feedback mechanisms like the "Data Entered Successfully" message. Overall, the Student's Mark Calculation System provides a strong framework for managing database interactions, scoring calculations, and student data organization.

In summary, this assignment has taught me a lot of new things, like how to investigate Python and its functions in more detail. I also learned from this assignment to never give up until I get the desired outcome—that is, to keep working until Python can be executed. In addition, this assignment has taught me to be more persistent and patient. I have also worked hard to ask for assistance, pick up tips from friends, and occasionally lend a hand. In addition, I would like to express my appreciation and gratitude to my lecturer, Sir Airul for his continuous guidance and consultation so that I can complete this individual assignment successfully.