MAYOOR PRIVATE SCHOOL, ABU DHABI, U.A.E

A Project Report
On

Student Tracker System

For
Grade XII CBSE 2022-23 Examination
[Computer Science]

SUBMITTED BY

NAME : Venkata Meghana Reddy Vusirika

ROLL NO : 27611420

Under the Guidance of
Ms. Shahnuma Khan

INTRODUCTION

Student Tracker System is a programmed system used for project management and note-taking. The project is designed to help students of a school coordinate deadlines, objectives, and assignments for efficiency and productivity. The primary purpose of this software is to help users schedule tasks, manage files, save documents, save words in a personalized dictionary, keep agendas, and organize their work. The project aims to design a tracking system for students which enables them to organize their work using different tools.

The project displays a menu asking them to select the tool to use. According to their selection, the system allows them to make a to-do list or organize their monthly or weekly goals or save difficult words in a personalized dictionary or make a reading list or save their syllabus semester-wise. The system also allows the users to view their previous entries and edit them.

Each of the above tasks contains a sub-menu that allows you to create a table, insert values into it and display and search the values from it. To perform each of the tasks, the system asks the user to enter the required details and allows the user to organize tasks and update the Student Tracker System database.

OBJECTIVE & SCOPE OF PROJECT

Student Tracker System is an application of a Database Management System which is used for scheduling and organizing information.

This system provides options for accessing different tools for organizing the work and provides the users with the facility to insert, display and search particular records.

The system displays an all-in-one workspace for note-taking, knowledge and data management, and project and task management. It is a file management tool offering a unified workspace, allowing users to comment on ongoing projects, make a timetable and track their goals.

THEORETICAL BACKGROUND

OVERVIEW OF PYTHON:

What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.
Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Some of the benefits of programming in Python include:

- Presence of Third Party Modules
- Extensive Support Libraries
- Open Source and Community Development
- Learning Ease and Support Available
- User-friendly Data Structures

OVERVIEW OF MYSQL:

What is SQL?

SQL is a Structured Query Language, which is a computer language for storing, manipulating, and retrieving data stored in a relational database. SQL is the standard language for Relational Database Systems. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres, and SQL Server use SQL as their standard database language.

What Can SQL do?

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows embedding within other languages using SQL modules, libraries & pre-compilers.

- Allows users to create and drop databases and tables.
- Allows users to create views, stored procedures, and functions in the database.
- Allows users to set permissions on tables, procedures, and views.

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS:

Minimum:
Processor: Intel 4th generation
Hard disk: 40GB
Ram: 2GB

Recommended:
Processor: Intel 6th generation
Hard disk: 100GB
Ram: 4GB or 8GB

SOFTWARE REQUIREMENTS:

Minimum:
OS: Windows 98,2000, XP, Mac OS X 10.9, 10.10
Tools: python 3.1.4 and MySQL 5.5.60

Recommended:
OS: Windows 7,8,9,10,11, Mac OS X 10.11, 10.12
Tools: python 3.5.0 and MySQL 5.6.4

TEAM ROLE

PROJECT TITLE:

Student Tracker System

TOTAL NO OF TEAM MEMBERS: 3

TEAM MEMBERS:

Amvi Dwivedi:

- Programming - To-do List and Weekly Goals
- Documentation
- Testing

Meghana Reddy:

- Programming - Monthly Goals and Dictionary
- Documentation
- Testing

Hannah Joseph:

- Programming - Semester Wise Syllabus and Book Tracker
- Documentation
- Testing

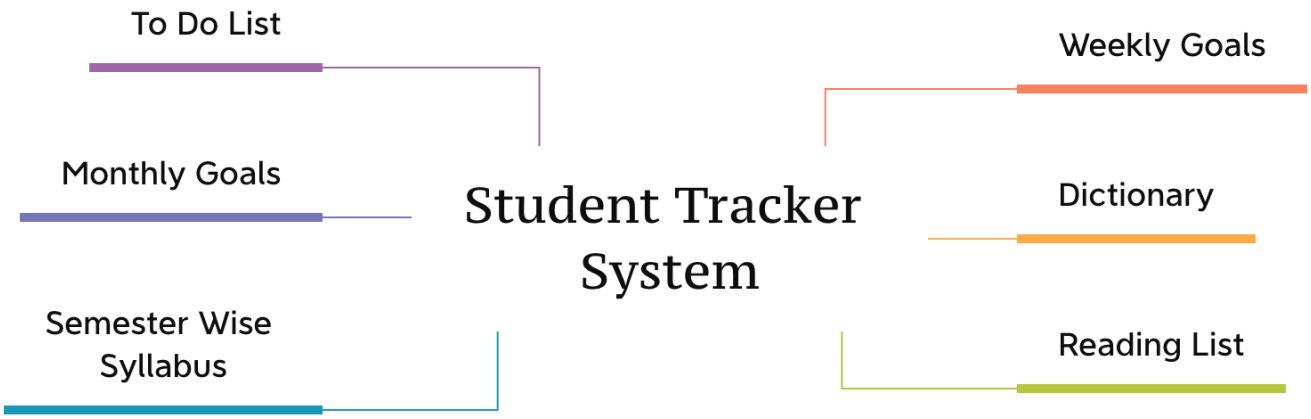
SYSTEM DESIGN AND ANALYSIS

MENU DESIGN:

S.no	Menu	Sub - Menu	Purpose
1.	To-Do List	To Create Table	Checks if the table exists, if not then creates it.
		To Insert Values	Inserts user input into the table
		To Display Values	Displays values according to the user.
2.	Weekly Goals	To Create Table	Checks if the table exists, if not then creates it.
		To Insert Values	Inserts user input into the table
		To Display Values	Displays values according to the user.
3.	Monthly Goals	To Create Table	Checks if the table exists, if not then creates it.
		To Insert Values	Inserts user input into the table
		To Display Values	Displays values according to the user.
4.	Dictionary	To Create Table	Checks if the table exists, if not then creates it.
		To Insert Values	Inserts user input into the table
		To Display Values	Displays values according to the user.
5.	Semester Wise Syllabus	Semester 1	Checks if the table exists, if not then creates it.
			Inserts user input into the table
			Displays values according to the user.

		Semester 2	Checks if the table exists, if not then creates it.
			Inserts user input into the table
			Displays values according to the user.
		Semester 3	Checks if the table exists, if not then creates it.
			Inserts user input into the table
			Displays values according to the user.
5.	Reading List	To Create Table	Checks if the table exists, if not then creates it.
		To Insert Values	Inserts user input into the table
		To Display Values	Displays values according to the user.

MIND MAP:


To Do List
Weekly Goals
Monthly Goals
Student Tracker System
Dictionary
Semester Wise Syllabus
Reading List

SOURCE CODE

```
import mysql.connector as s
con = s.connect(host = "localhost", user = "root", password =
"root", database = "Student_Tracker_System")
cur = con.cursor()
def todotable():
    try:
                cur.execute("create table Todo(Day date, Task
varchar(100), Completion varchar(10))")
        print("Table Created")
    except:
        print("Table Exist")
        todoinsert()

def todoinsert():
    Day = input("Enter date in YYYY-MM-DD form: ")
    Task = input("Enter task in less than 100 words: ")
    Completion = input("Enter done or not done: ")
    query = "insert into ToDo values(%s,%s,%s)"
    rec = (Day, Task, Completion)
    cur.execute(query,rec)
    con.commit()
    print('Added Successfully')

def tododisplay():
      d = input("Enter date to display corresponding task in
YYYY-MM-DD form: ")
    query = "select * from todo where Day = %s"
    d1 = (d,)
    cur.execute(query,d1)
    data = cur.fetchone()
```

```
    for i in data:
        print(i)

def weeklytable():
    try:
            cur.execute("create table Weekly_Goals(Start_Day date,
End_Day date, Task varchar(100), Completion varchar(10))")
        print("Table Created")
    except:
        print("Table Exists")
        weeklyinsert()

def weeklyinsert():
    Start_Day = input("Enter start date in YYYY-MM-DD form: ")
    End_Day = input("Enter end date in YYYY-MM-DD form: ")
    Task = input("Enter task in less than 100 words: ")
    Completion = input ("Enter done or not done: ")
    query = "insert into Weekly_Goals values(%s,%s,%s,%s)"
    rec = (Start_Day, End_Day, Task, Completion)
    cur.execute(query,rec)
    con.commit()
    print('Added Successfully')

def weeklydisplay():
    sd = input("Enter start date to display corresponding task in
YYYY-MM-DD form: ")
    ed = input("Enter end date to display corresponding task in
YYYY-MM-DD form: ")
    query = "select * from Weekly_Goals where Start_Day >= %s and
End_Day <=%s"
    d1 = (sd,ed)
    cur.execute(query,d1)
    data = cur.fetchone()
```

```
    for i in data:
        print(i)

def monthlytable():
    try:
         cur.execute("create table Monthly_Goals(Month varchar(15)
Primary key, Goals varchar(100))")
        print("Table Created")
    except:
        print("Table Exists")
        monthlyinsert()

def monthlyinsert():
    m = input("Enter the month: ")
      g = input("Enter the goal of the month in less than 100
words: ")
    query = "insert into Monthly_Goals values(%s,%s)"
    rec =(m,g)
    cur.execute(query,rec)
    con.commit()
    print('Added Successfully')

def monthlydisplay():
    d = input("Enter the month to display the task: ")
    query = "select * from Monthly_Goals where month = %s"
    d1=(d,)
    cur.execute(query,d1)
    data = cur.fetchone()
    for i in data:
        print(i)

def dicttable():
    try:
```

```
            cur.execute("create table Dictionary(Word varchar(50),
Meaning varchar(200))")
        print("Table Created")
    except:
        print("Table Exists")
        dictinsert()

def dictinsert():
    w = input('Enter the new word: ')
    m = input('Enter the meaning of the new word: ')
    Rec = (w, m)
    Cmd = "insert into Dictionary values(%s,%s)"
    cur.execute(Cmd,Rec)
    con.commit()
    print('Added Successfully')

def dictdisplay():
    try:
        w = input('Enter the word to search: ')
          cur.execute("select Meaning from Dictionary where Word =
'%s'" %(w,))
        data = cur.fetchone()
        for i in data:
            print(i)
    except:
        print('Error Found')

def sem1table():
    try:
          cur.execute("create table Semester1(Subject varchar(30),
Syllabus varchar(100))")
        print("Table Created")
```

```
    except:
        print("Table Exist")
        sem1insert()

def sem2table():
    try:
         cur.execute("create table Semester2(Subject varchar(30),
Syllabus varchar(100))")
        print("Table Created")
    except:
        print("Table Exist")
        sem2insert()

def sem3table():
    try:
         cur.execute("create table Semester3(Subject varchar(30),
Syllabus varchar(100))")
        print("Table Created")
    except:
        print("Table Exist")
        sem3insert()

def sem1insert():
    Sub1 = input("Enter the subject: ")
    Syl1 = input("Enter the syllabus in less than 100 words: ")
    query = "insert into Semester1 values(%s,%s)"
    rec = (Sub1,Syl1)
    cur.execute(query,rec)
    con.commit()
    print('Added Successfully')
```

```
def sem2insert():
    Sub2 = input("Enter the subject: ")
    Syl2 = input("Enter the syllabus in less than 100 words: ")
    query = "insert into Semester2 values(%s,%s)"
    rec = (Sub2,Syl2)
    cur.execute(query,rec)
    con.commit()
    print('Added Successfully')

def sem3insert():
    Sub3 = input("Enter the subject: ")
    Syl3 = input("Enter the syllabus in less than 100 words: ")
    query = "insert into Semester3 values(%s,%s)"
    rec = (Sub3,Syl3)
    cur.execute(query,rec)
    con.commit()
    print('Added Successfully')

def sem1display():
    d = input("Enter the subject: ")
    query = "select * from semester1 where Subject = %s"
    d1 = (d,)
    cur.execute(query,d1)
    data = cur.fetchone()
    for i in data:
        print(i)

def sem2display():
    d = input("Enter the subject: ")
    query = "select * from semester2 where Subject = %s"
    d1 = (d,)
    cur.execute(query,d1)
```

```
    data = cur.fetchone()
    for i in data:
        print(i)

def sem3display():
    d = input("Enter the subject: ")
    query = "select * from semester3 where Subject = %s"
    d1 = (d,)
    cur.execute(query,d1)
    data = cur.fetchone()
    for i in data:
        print(i)

def booktable():
    try:
          cur.execute("create table BookTracker(Day date, Bookname
varchar(100), Liked_or_Disliked varchar(10))")
        print("Table Created")
    except:
        print("Table Exists")
        bookinsert()

def bookinsert():
    Day = input("Enter date in YYYY-MM-DD form: ")
    Bookname = input("Enter name of the Book: ")
    Liked_or_Disliked = input("Enter liked or disliked: ")
    query = "insert into BookTracker values(%s,%s,%s)"
    rec = (Day, Bookname, Liked_or_Disliked)
    cur.execute(query,rec)
    con.commit()
    print('Added Successfully')
```

```
def bookdisplay():
     d = input("Enter date to display corresponding task in
YYYY-MM-DD form: ")
   query = "select * from BookTracker where Day = %s"
   d1=(d,)
   cur.execute(query,d1)
   data = cur.fetchone()
   for i in data:
       print(i)

print('************************  Welcome to Student Tracker
System ************************')
print()
print('''Menu:
Type 1 - To Do List
Type 2 - Weekly Goals
Type 3 - Monthly Goals
Type 4 - Dictionary
Type 5 - Semester Wise Syllabus
Type 6 - Reading List
Type 7 - Exit''')
print()

while True:
    q = int(input('Enter the number to choose from the main menu:
'))
   while True:
       if q == 1:
           print('''Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit''')
```

```
            t = int(input("Enter the number as per your choice:
"))
            if t == 1:
                todotable()
            elif t == 2:
                todoinsert()
            elif t == 3:
                tododisplay()
            elif t == 4:
                print('Bye')
                break
            else:
                print('Error Found')
                con.commit()
        elif q == 2:
            print('''Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit''')
            w = int(input("Enter the number as per your choice:
"))
            if w == 1:
                weeklytable()
            elif w == 2:
                weeklyinsert()
            elif w == 3:
                weeklydisplay()
            elif w == 4:
                print('Bye')
                break
            else:
                print('Error Found')
                con.commit()
```

```
        elif q == 3:
            print('''Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit''')
             m = int(input("Enter the number as per your choice:
"))
            if m == 1:
                monthlytable()
            elif m == 2:
                monthlyinsert()
            elif m == 3:
                monthlydisplay()
            elif m == 4:
                print('Bye')
                break
            else:
                print('Error Found')
                con.commit()
        elif q == 4:
            print('''Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit''')
             dic = int(input("Enter the number as per your choice:
"))
            if dic == 1:
                dicttable()
            elif dic == 2:
                dictinsert()
```

```
            elif dic == 3:
                dictdisplay()
            elif dic == 4:
                print('Bye')
                break
            else:
                print('Error Found')
                con.commit()
        elif q == 5:
            print('''Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit''')
            n = int(input('Enter the number of the semester: '))
            if n == 1:
                    s1 = int(input('Enter the number as per your
choice: '))
                if s1 == 1:
                    sem1table()
                elif s1 == 2:
                    sem1insert()
                elif s1 == 3:
                    sem1display()
                else:
                    print('Error Found')
            elif n == 2:
                    s2 = int(input('Enter the number as per your
choice: '))
                if s2 == 1:
                    sem2table()
                elif s2 == 2:
                    sem2insert()
```

```
                elif s2 == 3:
                    sem2display()
                else:
                    print('Error Found')
            elif n == 3:
                     s3 = int(input('Enter the number as per your
choice: '))
                if s3 == 1:
                    sem3table()
                elif s3 == 2:
                    sem3insert()
                elif s3 == 3:
                    sem3display()
                else:
                    print('Error Found')
            elif n == 4:
                print('Bye')
                break
            else:
                print('Error Found')
                con.commit()
        elif q == 6:
             print('''Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit''')
              b = int(input("Enter the number as per your choice:
"))
             if b == 1:
                 booktable()
             elif b == 2:
                 bookinsert()
```

```
        elif b == 3:
            bookdisplay()
        elif b == 4:
            print('Bye')
            break
        else:
            print('Error Found')
            con.commit()
elif q == 7:
     print('Bye')
     break
else:
   print('Error Found')
```

SOURCE OUTPUT

STRUCTURES OF THE TABLES:

Table Todo:

Field	Type	Null	Key	Default	Extra
Day	date	YES		NULL	
Task	varchar(100)	YES		NULL	
Completion	varchar(10)	YES		NULL	

Table Weekly:

Field	Type	Null	Key	Default	Extra
Start_Day	date	YES		NULL	
End_Day	date	YES		NULL	
Task	varchar(100)	YES		NULL	
Completion	varchar(10)	YES		NULL	

Table Monthly Goals:

Field	Type	Null	Key	Default	Extra
Month	varchar(15)	NO	PRI	NULL	
Goals	varchar(100)	YES		NULL	

Table Dictionary:

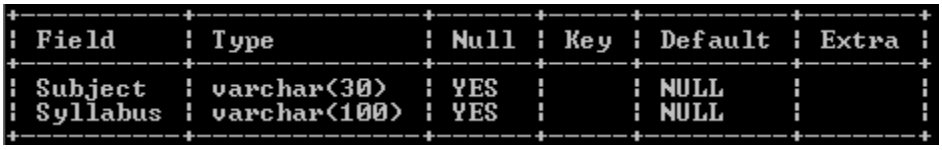
Field	Type	Null	Key	Default	Extra
Word	varchar(50)	NO	PRI	NULL	
Meaning	varchar(200)	YES		NULL	

Semester Wise Syllabus:

- Table Semester1:

Field	Type	Null	Key	Default	Extra
Subject	varchar(30)	YES		NULL	
Syllabus	varchar(100)	YES		NULL	

- Table Semester2:

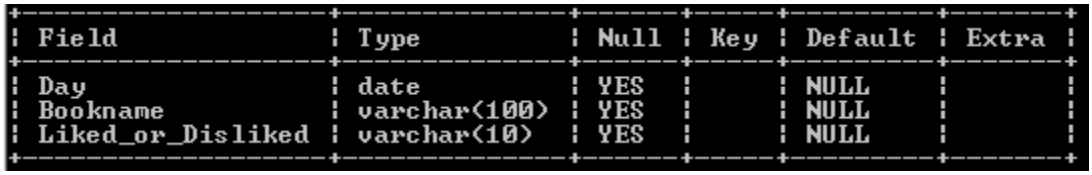


Field	Type	Null	Key	Default	Extra
Subject	varchar(30)	YES		NULL	
Syllabus	varchar(100)	YES		NULL	

- Table Semester3:

Field	Type	Null	Key	Default	Extra
Subject	varchar(30)	YES		NULL	
Syllabus	varchar(100)	YES		NULL	

Table Book Tracker:



Field	Type	Null	Key	Default	Extra
Day	date	YES		NULL	
Bookname	varchar(100)	YES		NULL	
Liked_or_Disliked	varchar(10)	YES		NULL	

OUTPUT OF THE CODE:

```
************************* Welcome to Student Tracker System *************************

Menu:
Type 1 - To Do List
Type 2 - Weekly Goals
Type 3 - Monthly Goals
Type 4 - Dictionary
Type 5 - Semester Wise Syllabus
Type 6 - Reading List
Type 7 - Exit

Enter the number to choose from the main menu: 1
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 1
Table Exist
Enter date in YYYY-MM-DD form: 2022-11-28
Enter task in less than 100 words: Do exercise
Enter done or not done: done
Added Successfully
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 3
Enter date to display corresponding task in YYYY-MM-DD form: 2022-11-28
2022-11-28
Do exercise
done
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 4
Bye
```

```
Enter the number to choose from the main menu: 2
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 1
Table Exists
Enter start date in YYYY-MM-DD form: 2022-11-28
Enter end date in YYYY-MM-DD form: 2022-12-04
Enter task in less than 100 words: Write chemistry notes
Enter done or not done: not done
Added Successfully
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 3
Enter start date to display corresponding task in YYYY-MM-DD form: 2022-11-28
Enter end date to display corresponding task in YYYY-MM-DD form: 2022-12-04
2022-11-28
2022-12-04
Write chemistry notes
not done
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 4
Bye
```

```
Enter the number to choose from the main menu: 3
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 1
Table Created
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 2
Enter the month: June
Enter the goal of the month in less than 100 words: Drink 8 glasses of water
Added Successfully
```

```
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 3
Enter the month to display the task: June
June
Drink 8 glasses of water
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 4
Bye
Enter the number to choose from the main menu: 4
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 1
Table Created
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 2
Enter the new word: Rural
Enter the meaning of the new word: Countryside
Added Successfully
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 3
Enter the word to search: Rural
Countryside
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 4
Bye
```

```
Enter the number to choose from the main menu: 5
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 1
Enter the number as per your choice: 1
Table Created
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 1
Enter the number as per your choice: 2
Enter the subject: Mathematics
Enter the syllabus in less than 100 words: Trigonometric Functions
Added Successfully
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 1
Enter the number as per your choice: 3
Enter the subject: Mathematics
Mathematics
Trigonometric Functions
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 2
Enter the number as per your choice: 1
Table Created
```

```
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 2
Enter the number as per your choice: 2
Enter the subject: Physics
Enter the syllabus in less than 100 words: Ray Optics
Added Successfully
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 2
Enter the number as per your choice: 3
Enter the subject: Physics
Physics
Ray Optics
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 3
Enter the number as per your choice: 1
Table Created
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 3
Enter the number as per your choice: 2
Enter the subject: Chemistry
Enter the syllabus in less than 100 words: Amines
Added Successfully
```

```
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 3
Enter the number as per your choice: 3
Enter the subject: Chemistry
Chemistry
Amines
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 1
Enter the number as per your choice: 4
Error Found
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number of the semester: 4
Bye
```

```
Enter the number to choose from the main menu: 6
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 1
Table Created
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 2
Enter date in YYYY-MM-DD form: 2022-10-08
Enter name of the Book: Harry Potter and Half Blood Prince
Enter liked or disliked: liked
Added Successfully
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 3
Enter date to display corresponding task in YYYY-MM-DD form: 2022-10-08
2022-10-08
Harry Potter and Half Blood Prince
liked
Sub Menu:
Type 1 - To Create Table
Type 2 - To Insert Values
Type 3 - To Display Values
Type 4 - To Exit
Enter the number as per your choice: 4
Bye
```

USER MANUAL

REQUIREMENTS:

System with Windows10 (recommended)
Python3 Version
MYSQL DataBase

STEP 1:

- Extract the “Student Tracker System” to a folder.

STEP 2:

- Run the following code in the MySQL database before running the code.
  Create database student_tracker_system;
- Open the “Student Tracker System” file and run it.
- Student Tracker System is the project file with a built-in SQL database and python functions.
- When it runs, it will call for the functions and databases which are deployed to create the required SQL database, tables, and python functions.
- The file is created in a way that the user may view the main menu section and enter the number to select the tool to use as per the instructions given in the menu to make a to-do list, organize their monthly or weekly goals, etc.

CAUTIONS:

- Make sure that the required hardware and software are installed.
- Try to follow the instructions and give only "VALID" inputs.
  ex: Do not give alphabetical input when asked for a date, which is a numerical value and in standard date form
- Code explanations and directions are commented throughout the code. If any confusion occurs, just ask.

BIBLIOGRAPHY

BOOKS:

- Computer Science with Python Textbook by Sumita Arora (Class XI)
- Computer Science with Python Textbook by Sumita Arora (Class XII)

LINKS:

- https://www.google.com
- https://www.python.org.in
- https://www.mysql.org