HEMVATI NANDAN BAHUGUNA GARHWAL UNIVERSITY

(A Central University)

Srinagar Garhwal, Uttarakhand

School of Engineering and Technology

Session (2020 - 2021)

A PROJECT REPORT ON

"Tecko Database"

Submitted in Partial fulfillment for the award of the degree of

Bachelor of Technology

in Computer Science and Engineering

HNBGU, Srinagar Garhwal (Uttarakhand)

Guided By:-

Mr. Vijay Bijlwan

Dept. of Computer Science and Engg.

Submitted By:-

Kartikeya

B.Tech (C.S.E.) - VI Sem

Roll No. 18134503008

DECLARATION

I, **Kartikeya** bearing the roll no **18134503008** , student of Computer Science and Engineering Department at Hemvati Nandan Bahuguna Garhwal University (A Central University), Srinagar (Garhwal), Uttarakhand, submit this project report entitled "**Tecko Database**" to Computer Science and Engineering Department, Hemvati Nandan Bahuguna Garhwal University, for the award of the **Bachelors of Technology degree in Computer Science & Engineering** and declaring that the work done is genuine and produced under the guidance of **Mr. Vijay Bijlwan.**, Department of Computer Science and Engineering, Hemvati Nandan Bahuguna Garhwal University.

I further declare that the reported work in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this institute or any other institute or university.

DATE: 24-09-2021

PLACE: Srinagar

Student name

Kartikeya
Roll no –

18134503008

CERTIFICATE

This is to certify that, this project report titled "**Tecko Database**" submitted by **Kartikeya** bearing roll no **18134503008** is bonafide record of the work carried out by him/her in partial fulfilment for the requirement of the award of **Bachelor of Technology** in **Computer Science and Engineering** degree from Hemvati Nandan Bahuguna Garhwal University (A Central University) at Srinagar (Garhwal), Uttarakhand.

This Project report has not been submitted to any other University or Institution for the award of any degree.

Mr. Vijay Bijlwan

Department of Computer Science & Engineering

Hemvati Nandan Bahuguna Garhwal University (A Central University)

Srinagar (Garhwal), Uttarakhand

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all people for sprinkling their help and kindness in the completion of this Project. I would like to start this moment by invoking my purest gratitude to **Mr. Vijay Bijlwan**, Department of Computer Science and Engineering, Hemwati Nandan Bahuguna Garhwal University (A Central University), Srinagar (Garhwal), Uttarakhand, my project instructor.

The completion of this project could not have been possible without his expertise and invaluable guidance in every phase at Hemvati Nandan Bahuguna Garhwal University (A Central University), Srinagar (Garhwal), Uttarakhand for helping me.

I would like to thank **Prof. M.M.S Rauthan, Prof. Y.P Raiwani,** all the lab assistants and other staffs of Computer Science and Engineering Department, Hemvati Nandan Bahuguna Garhwal University (A Central University), Srinagar (Garhwal), Uttarakhand, for their kind support. Last but not least, I would like to thank my parents and my friends for their unwavering belief despite ups and downs in my journey.

ABSTRACT

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

SQLite is a relational database management system contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program. SQLite generally follows PostgreSQL syntax

CONTENTS

Database:

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management sydtem(DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized

Evolution of Database:

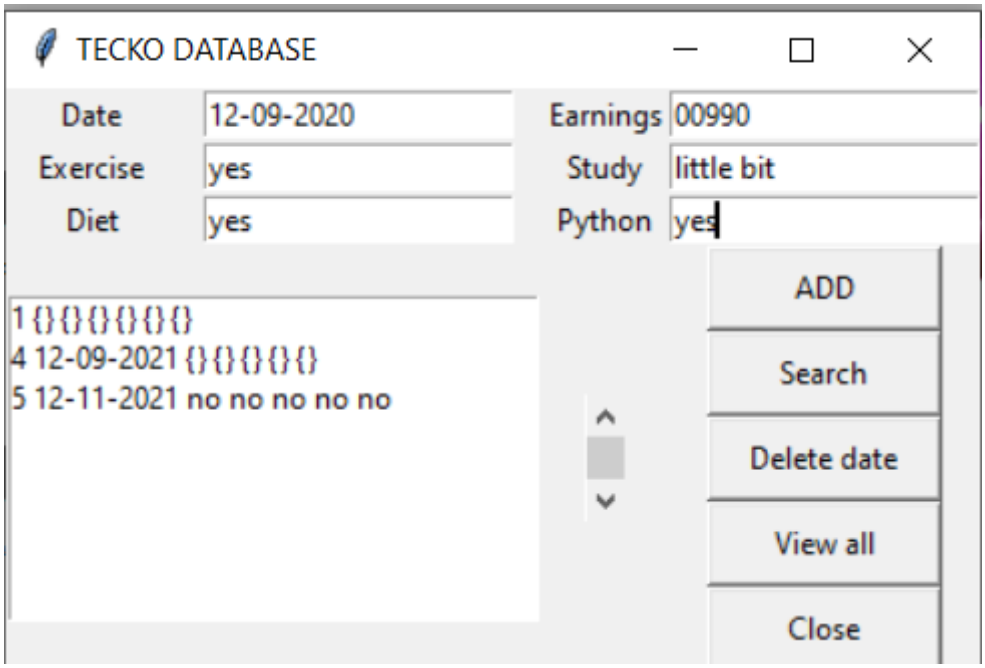
Databases have evolved dramatically since their inception in the early 1960s. Navigational databases such as the hierarchical database (which relied on a tree-like model and allowed only a one-to-many relationship), and the network database (a more flexible model that allowed multiple relationships), were the original systems used to store and manipulate data. Although simple, these early systems were inflexible. In the 1980s,relational databases became popular, followed by object-oriented databases in the 1990s. More recently, NoSQL databases came about as a response to the growth of the internet and the need for faster speed and processing of unstructured data. Today, cloud databases and self-driving databases are breaking new ground when it comes to how data is collected, stored, managed, and utilized.

Features of Database:

- The database helps in organizing data in an organized way.
- The use of data eliminates data redundancy and provide data consistency.
- Data security gets improved with the use of a database system.
- Data can be easily managed by using a database system.
- The database system provides multiuser data access.

Introduction of Tecko Database:

Tecko database is an application of database in which you can store your regular routine like you have did in your whole day study, exercise, what you eat, the places where you visited etc. on the bases of your activity it analyse your daily routine you can check what you have did last day, week, month.



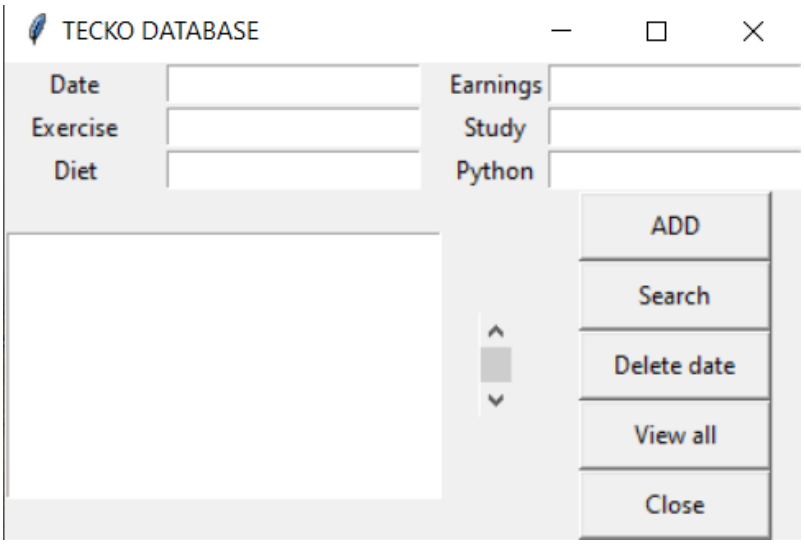
Tecko Database

Overview Of Project:

In this project we have three codes four strings a front that is during front and then back and then the database. you just need to run only one code.

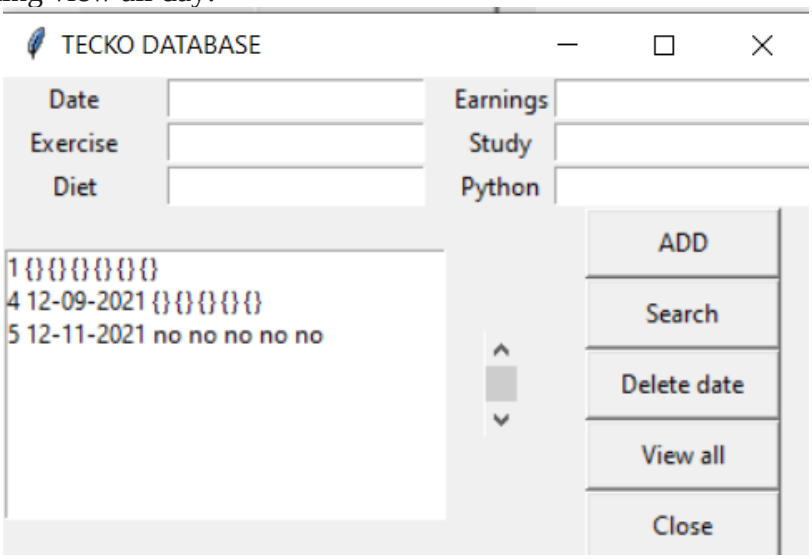
It will call others automatically.

this is how our actual applications will look like a simple extra application.

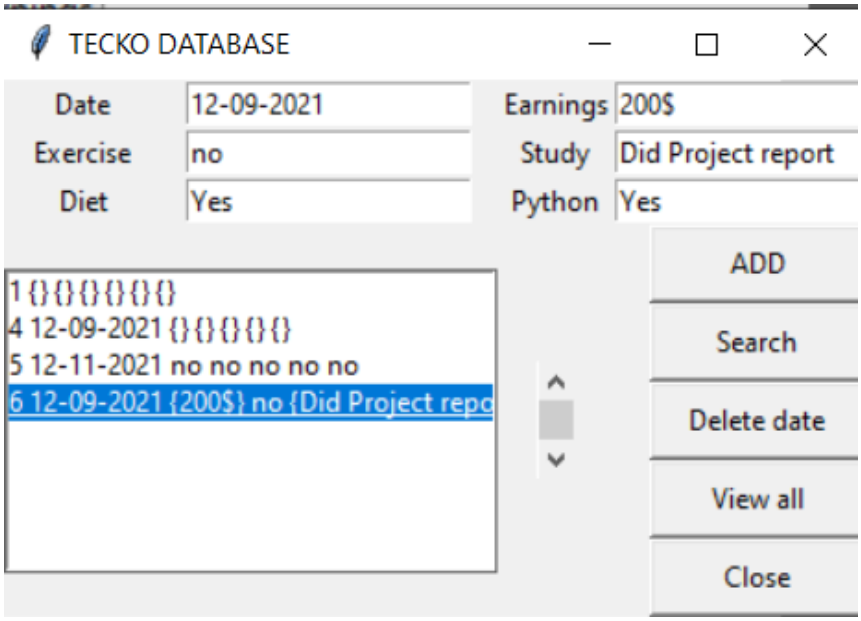


Overview

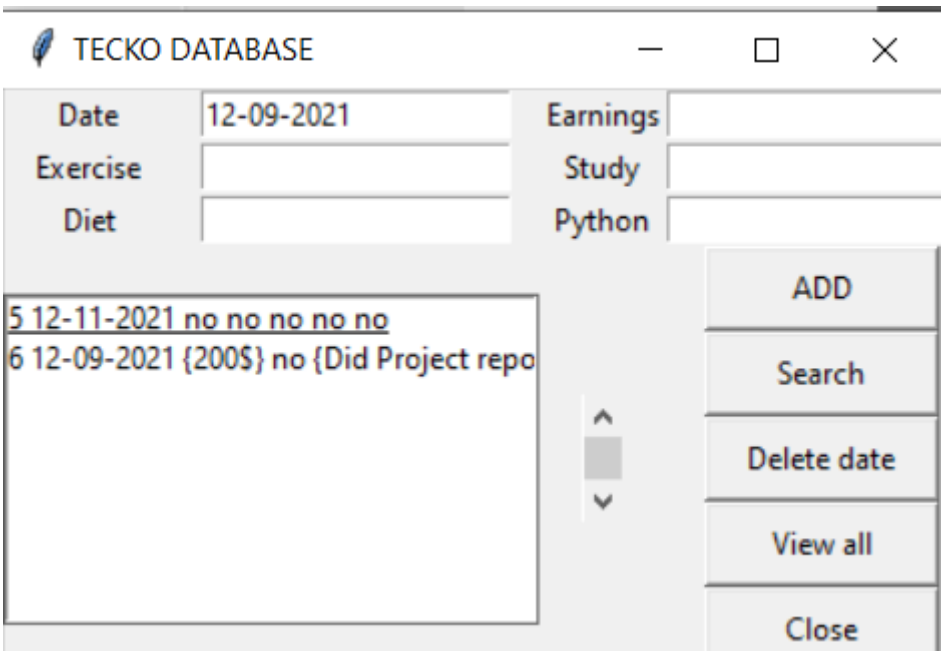
- In this one you can add any date like say today's date 2019
- your earnings at how you have how much you have on today or yesterday or whenever are writing
  i.e.- say like hundred dollars
- Exercise So like no exercise
- Study I studied sorties in my career that would involve study and exercise night. It will appear in the list box and it will be added in the database. You can view that one just by clicking view all day.



We had this entry on the 8 here I have few more entries that I ever did before.
By using this one you can just analyse everything like how much you have on this week on what days, you have exercise. You have studied you have taken data properly did Python or not.
It will help a lot if you use this one regularly.



Also you can just delete the values from here by clicking delete or selecting the particular value click delete and view all that will not be there.
I delete my first entry & second entry they are the same entries delete this one also and this one also now viewable



we have only two entries that is 5 and 6 after that one here we are done with the add and delete. After that one you can also do something like Here search like if you want to find it how many days.I hope on 600 just right six under search you will get all the days when you have on

deadline you want analyses of when you have not studied so just tried not studied or whatever do what you are using their search you will get the days when you have not studied.
So basically they are on the only fault and on for that kind of study.
So this will be a little application but takes a lot of time to be and will help you if you use this one this one also contains a scroll bar.There then just add this time and again and again to see this scroll bar.
So this also a close button that can destroy the window and you get out of this.
Now if you view all you have load of here and this school while also working Nagel so this is all about the application that what we have to build here and again this close button just clicking this on the application will be closed.
So if you are feeling compatibility databases and data again go for this one and if not then go with me again this project I suggest you to work with and complete this one and if you get any down there

Development Tools and System Requirement:

- **Language-**

  1. **Python(Tkinter)-** Python is used for frontend & backend. Tkinter is use for giving GUI based design
  2. **SQLite3-** For saving data in database.

- **Editor-**

  1. **VS Code**-This editor is uded for editing the all code of my project.

Python & SQlite3

Architecture of Tecko Database

A. Frontend

1.Labels

I created one file named frontend.py, So in this section we are working on frontend.
Like all the labels we have six labels here.
So we require six labels here.So let's begin first of all from big doughin both starred label and then define a window geeky and just fiend or main loop.They will now first of all let me begin with the single label that is going to be.And I hope you remember how we define the widgets just as capital a label define their value that like in windows. And the test you want on that same date.They will now just provide the grid metal here and green and drew equal to zero golem equal to zero because we have this one in the ready initial stage.So save this one and then I run my front end or be white.There we go with a small data label in the corner No.
And change these two L2 three and four and five and six maybe go.Now based on that one.

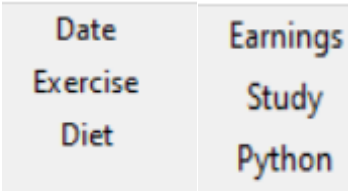
- First of all on the first we have the date hail the second one.
- Then we have exercise.So he exercise up to that one.
- We are going to his study then died.

Zero and column is zero and zero one.So change this one just do.And then we have data one in which we have the exercise and study.So exercise in studying that one.
Just exercise in column detail and study in column two.
Then in the row two which is actually third but with zero unless this one is two we have a date type story and item by item column zero and pattern on column two for one that one you will get all the six labels but they are one after the another because the label with column 0 sorry the columns you have you have defines us zero into the one is in between these when we define entry between them it would just look like this one.

frontend.py backend.py

frontend.py > get_selected_row

```
from tkinter import *
import backend

def get_selected_row(event):
    global selected_row
    index = list.curselection()[0]
    selected_row = list.get(index)  #Creating Labels
    e1.delete(0,END)
    e1.insert(END,selected_row[1])
    e2.delete(0,END)
    e2.insert(END,selected_row[2])
    e3.delete(0,END)
    e3.insert(END,selected_row[3])
    e4.delete(0,END)
    e4.insert(END,selected_row[4])
    e5.delete(0,END)
    e5.insert(END,selected_row[5])
    e6.delete(0,END)
    e6.insert(END,selected_row[6])
```



Labels

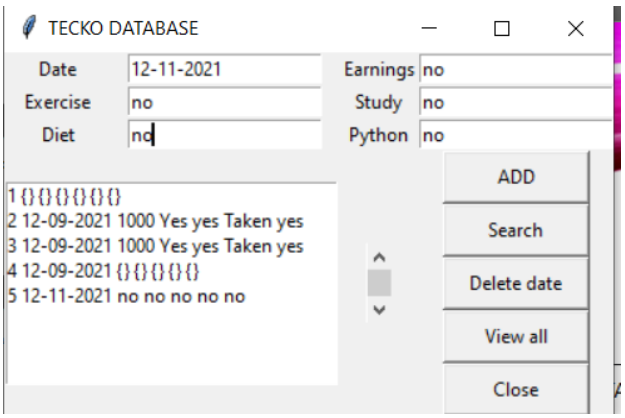
2.Entries, List Box & ScrollBar-

- A list box is a graphical control element that allows the user to select one or more items from a list contained within a static, multiple line text box.
- Scrollbar is basically use for viewing all data in database.

```
#Creating ScrollBar

sb = Scrollbar(win)
sb.grid(row=3,column=2,rowspan=9)
```

```
#Creating Entries

date_text = StringVar()
e1 = Entry(win, textvariable=date_text)
e1.grid(row=0,column=1)

earning_text = StringVar()
e2 = Entry(win, textvariable=earning_text)
e2.grid(row=0,column=3)

exercise_text = StringVar()
e3 = Entry(win, textvariable=exercise_text)
e3.grid(row=1,column=1)

study_text = StringVar()
e4 = Entry(win, textvariable=study_text)
e4.grid(row=1,column=3)

diet_text = StringVar()
e5 = Entry(win, textvariable=diet_text)
e5.grid(row=2,column=1)

python_text = StringVar()
e6 = Entry(win, textvariable=python_text)
e6.grid(row=2,column=3)
```

```
#Creating ListBox

list = Listbox(win,height=8,width=35)
list.grid(row=3,column=0,rowspan=9,columnspan=2)

#Creating ScrollBar

sb = Scrollbar(win)
sb.grid(row=3,column=2,rowspan=9)

#Binding the List Function

list.bind('<<ListboxSelect>>',get_selected_row)
```

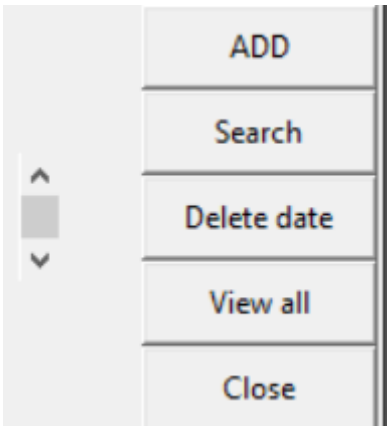


Lables,ListBox & Scrollbar

3.Buttons:

Here we have basically five buttons in the column three and we'd have differentials like three four five six seven. So let's begin with that one. Most often simply button widget where you need them in window decks to Newt there. And then just be window green. There you go.

```
#Creating Buttons

b1 = Button(win,text='ADD',width=12,pady=5,command=add_command)
b1.grid(row=3,column=3)

b2 = Button(win,text='Search',width=12,pady=5,command=search_command)
b2.grid(row=4,column=3)

b3 = Button(win,text='Delete date',width=12,pady=5,command=delete_command)
b3.grid(row=5,column=3)

b4 = Button(win,text='View all',width=12,pady=5,command=view_command)
b4.grid(row=6,column=3)

b5 = Button(win,text='Close',width=12,pady=5,command = win.destroy)
b5.grid(row=7,column=3)
```



Buttons

B. Backend

I created new code file named backend .py. In this module we are working on this file. import SQLite3 maybe go then just define a function that will connect and create my database.Going to be a new database here.

Here we don't have any file of that one previous

- First disconnect.
- Second because a third execution so Kosovo would execute.
- And after that one connection to commit.

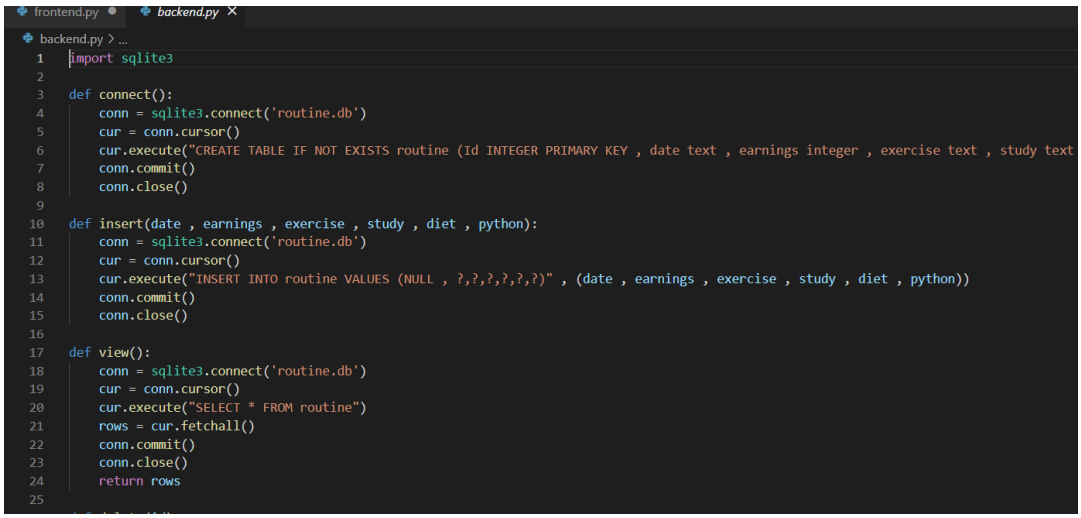
- And in last close the connection there we go.

First of all in Connect destroyed as you like three don't connect and pass the database name you want to create several beam no baby maybe go after that one create a super connection to go Zo legal.So here we have because a variable names.

CREATE TABLE.

So we'll create a table here with name Ruby after that one.In that we have the things that we require of these things.Date of Ning exercise the date and bite.And in addition in addition to these I'm also adding primary key data that is going to be an I.D. that's something like the things that I presented with numbers div so. And that will be an integer and this one is going to be a primary key.So the user know required to add this one it will auto increment with each of the values so a key will be assigned to every single entry.After that let's move to the ones that we require first of all the date which is going to be a trigger text.

What are the values do we have there.
do feel save and now run daily over the values one that is the primary key given by this one and one to 2019 to under no exercise not stood a take on dead Python so perfect that one is working perfectly fine that we try to insert one more here so that it will be shown for that so one thousand ninety hundred and then did exercise up to that one studied then light taken and ideally and after that one did Python labial run that one again. And now just print view.And if you remember in delete we also need to provide a quagmire because we need to delete all the values



C. Routine.db-

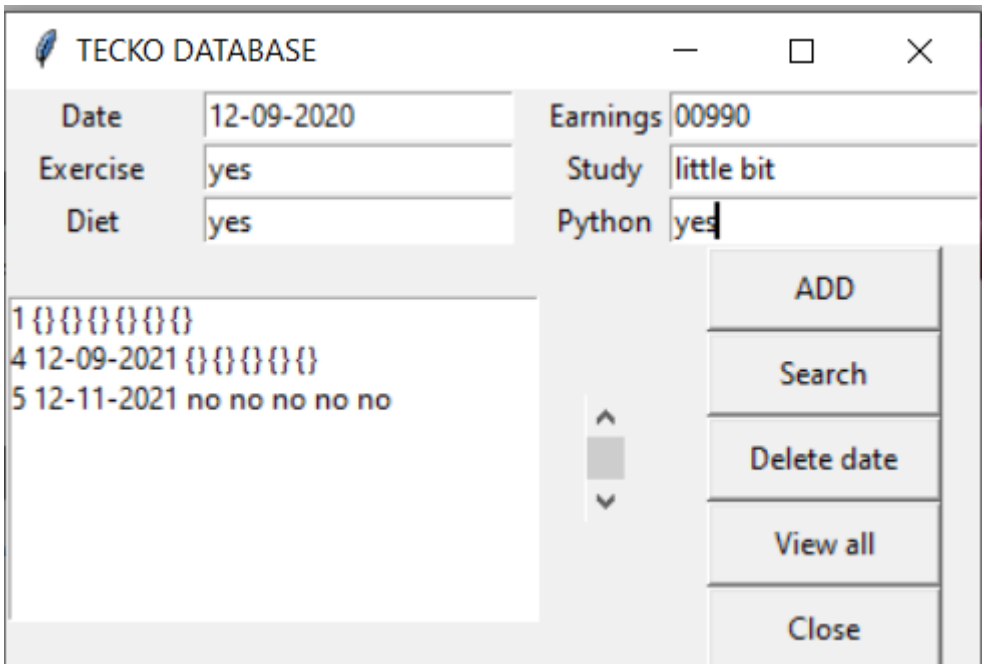
There is 3rd file named **routine.db** , where all of our data is saved.We don’t need to access this file , we can access only frontend file .

Features of Tecko Database

1.Adding:

The program we get the values by using the get matter and just insert them in the back end. that's very simple And you can also print the particular added values here so that user can have an idea that what he has entered. add button and right there. at this point every of these value is added in the back end. we have just inserted all the values in the entries by get matter. after this one to print values here again do the same.

Something like 10 10 to 19 say this time 500 did exercise did study.an ideal day here.now just click Add. here we have this entry right now here. And when I click viewable there we have all the entries now that we try one more here 20
no exercise taken no study did bite at their fuel day we had this in 2010 219 500.
No exercise no study Date Taken did bite.



Adding Data

2.Searching & Viewing the Data

Insert first let we have 1 1 2 0 1 9 then hundred after that one did exercise.
Then after that one what we have there study so did study after that one night taken and then last. View and we had three entries.

- One is with one to 2019.
- Then one by nineteen then four for ninety so now that we remove these lines and define of as such.

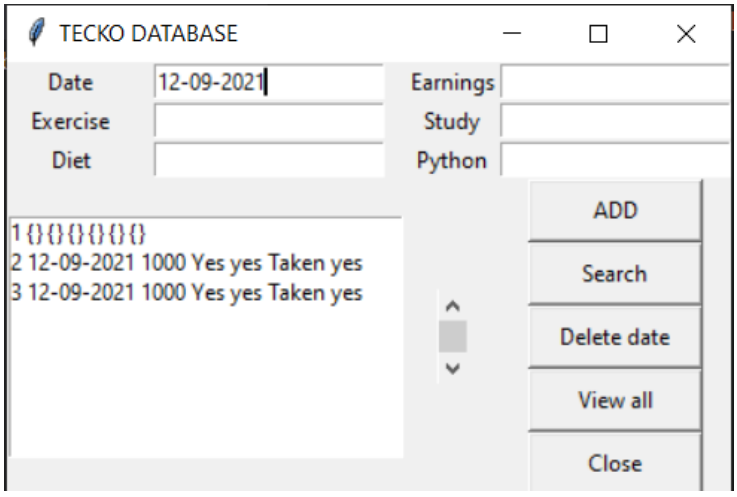
Search is something that we have not done before and is very easy. User did not enter anything then by default their value is null. So by this you posited me does to this such comma after that one.

This is a search common and perfectly working fine.So all of our commands are working fine here.Now we have a perfect database.

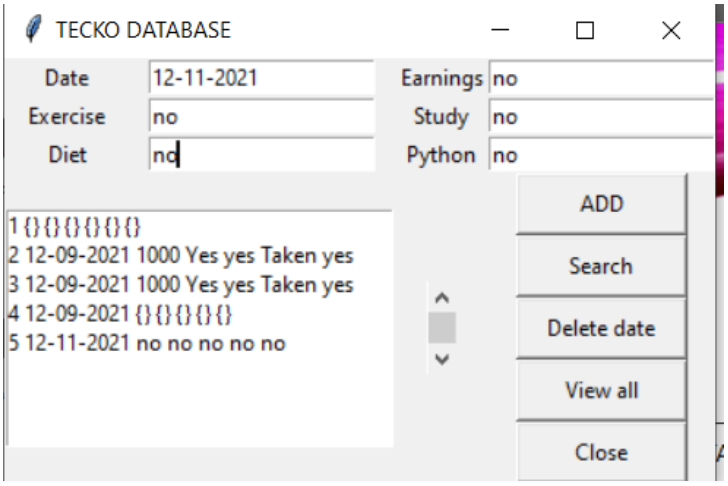
Just there are two things too many first one.Just imagine we are working with other application and at this point we had the database.What if it doesn't have the database.So we have done here something like we have first connected this one by using the function first connected at that point that is created a table in the database.But what if there is no database then we need to again call that one.

So this is the first thing we need to add here a connect.But what if there is already existing data.So in that case it will provide me an ETA like I had saved this one.
And when I run this one I got the data by table who already exist.So what.
In that case.So what we will do here is we may make this one remain here and we will add one more line here.

Create table if not existence.So if the table in the database doesn't exist that create a table.Otherwise we have that table already so if we do this fun and run the file legging then deal in that case we could no end game no way to.
So this is complete about a bacon and now we did not require any change in despair we would just add a line here in the boat back and that's enough save this one and we are done with this.



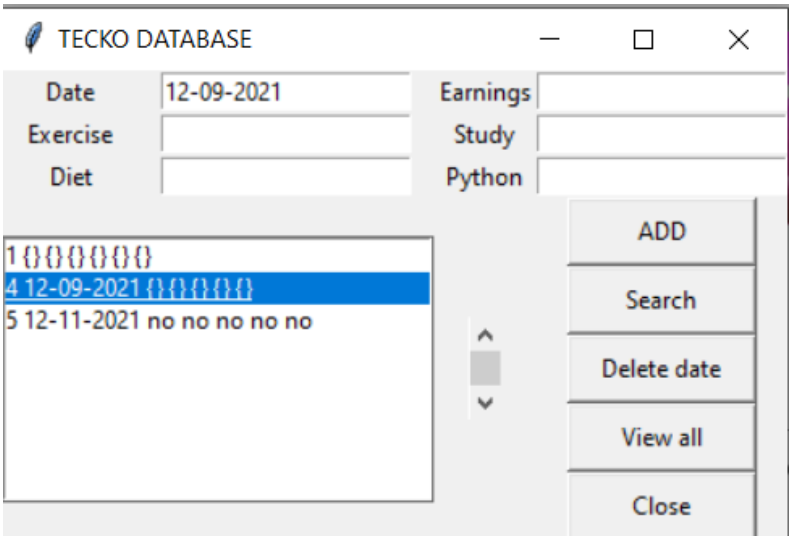
Searching Data



Viewing Data

3.Deleting

Just the delete on left and delete Come on it's very simple.
Just here we have this one.
We need to select any value and click on delete.
And that back and all that one is also very easy.
Here you are clicking the values in this list box but we require something that can store that particular value so that the program can have an idea that which of the rows it must has to delete.
I hope you get that one but I want to see him like these all these clicks are random so plus we let make that clear. And now let me move to one.
And that is something like selected row the row I have selected there.
And now what we will do we will just get the index of that particular row and parse that particular index to delete in that way that particular rule will be deleted from the database and how we will do that one.
Now the six day we do not have that. Now I'm going to delete this one with 1 & 2 delete view all day we go with that one so that it's now deleted. So here we are done with our database.



Deleting Data

Source codes:

Fronted.py

```
from tkinter import *
import backend

def get_selected_row(event):
    global selected_row
    index = list.curselection()[0]
    selected_row = list.get(index)  #Creating Labels
    e1.delete(0,END)
    e1.insert(END,selected_row[1])
    e2.delete(0,END)
    e2.insert(END,selected_row[2])
    e3.delete(0,END)
    e3.insert(END,selected_row[3])
    e4.delete(0,END)
    e4.insert(END,selected_row[4])
    e5.delete(0,END)
    e5.insert(END,selected_row[5])
    e6.delete(0,END)
    e6.insert(END,selected_row[6])



def delete_command():
    backend.delete(selected_row[0])

def view_command():
    list.delete(0,END)
    for row in backend.view():
        list.insert(END,row)

def search_command():
    list.delete(0,END)
    for row in backend.search(date_text.get(),earning_text.get(),exercise_text.get(),study_text.get(),diet_text
.get(),python_text.get()):
        list.insert(END,row)

def add_command():
    backend.insert(date_text.get(),earning_text.get(),exercise_text.get(),study_text.get(),diet_text.get(),pytho
n_text.get())

    list.delete(0,END)
    list.insert(END,(date_text.get(),earning_text.get(),exercise_text.get(),study_text.get(),diet_text.get(),pyt
hon_text.get()))

win = Tk()
```

```
win.wm_title('TECKO DATABASE')

l1 = Label(win, text='Date')
l1.grid(row=0,column=0)
l2 = Label(win, text='Earnings')
l2.grid(row=0,column=2)
l3 = Label(win, text='Exercise')
l3.grid(row=1,column=0)
l4 = Label(win, text='Study')
l4.grid(row=1,column=2)
l5 = Label(win, text='Diet')
l5.grid(row=2,column=0)
l6 = Label(win, text='Python')
l6.grid(row=2,column=2)

#Creating Entries

date_text = StringVar()
e1 = Entry(win, textvariable=date_text)
e1.grid(row=0,column=1)

earning_text = StringVar()
e2 = Entry(win, textvariable=earning_text)
e2.grid(row=0,column=3)

exercise_text = StringVar()
e3 = Entry(win, textvariable=exercise_text)
e3.grid(row=1,column=1)

study_text = StringVar()
e4 = Entry(win, textvariable=study_text)
e4.grid(row=1,column=3)

diet_text = StringVar()
e5 = Entry(win, textvariable=diet_text)
e5.grid(row=2,column=1)

python_text = StringVar()
e6 = Entry(win, textvariable=python_text)
e6.grid(row=2,column=3)

#Creating ListBox

list = Listbox(win,height=8,width=35)
list.grid(row=3,column=0,rowspan=9,columnspan=2)

#Creating ScrollBar
```

```
sb = Scrollbar(win)
sb.grid(row=3,column=2,rowspan=9)

#Binding the List Function

list.bind('<<ListboxSelect>>',get_selected_row)

#Creating Buttons

b1 = Button(win,text='ADD',width=12,pady=5,command=add_command)
b1.grid(row=3,column=3)

b2 = Button(win,text='Search',width=12,pady=5,command=search_command)
b2.grid(row=4,column=3)

b3 = Button(win,text='Delete date',width=12,pady=5,command=delete_command)
b3.grid(row=5,column=3)

b4 = Button(win,text='View all',width=12,pady=5,command=view_command)
b4.grid(row=6,column=3)

b5 = Button(win,text='Close',width=12,pady=5,command = win.destroy)
b5.grid(row=7,column=3)

win.mainloop()
```

Backend.py

```
import sqlite3

def connect():
    conn = sqlite3.connect('routine.db')
    cur = conn.cursor()
    cur.execute("CREATE TABLE IF NOT EXISTS routine (Id INTEGER PRIMARY KEY , date text , ea
rnings integer , exercise text , study text , diet text ,python text)")
    conn.commit()
    conn.close()

def insert(date , earnings , exercise , study , diet , python):
    conn = sqlite3.connect('routine.db')
    cur = conn.cursor()
    cur.execute("INSERT INTO routine VALUES (NULL , ?,?,?,?,?,?)" , (date , earnings , exercise , study ,
diet , python))
    conn.commit()
```

```
    conn.close()

def view():
    conn = sqlite3.connect('routine.db')
    cur = conn.cursor()
    cur.execute("SELECT * FROM routine")
    rows = cur.fetchall()
    conn.commit()
    conn.close()
    return rows

def delete(id):
    conn = sqlite3.connect('routine.db')
    cur = conn.cursor()
    cur.execute("DELETE FROM routine WHERE id=? ", (id,))
    conn.commit()
    conn.close()

def search(date='' , earnings='' , exercise='' , study='' , diet='' , python=''):
    conn = sqlite3.connect('routine.db')
    cur = conn.cursor()
    cur.execute("SELECT * FROM routine WHERE date=?  OR earnings=? OR exercise=? OR study=? O
R diet=? OR python=?" , (date , earnings , exercise , study , diet , python))
    rows = cur.fetchall()
    conn.commit()
    conn.close()
    return rows

connect()
```

Future Scope & Conclusion:

- We can use this database is as Routine Calender
- We don’t require any type internet connection.
- In future, we implement more labels.

References:

1. https://medium.com/analytics-vidhya/programming-with-databases-in-python-using-sqlite-4cecbef51ab9
2. https://stackoverflow.com/