**Table of contents**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Content** | **Page No.** |
| 1 | Name/title of the project | **1** |
| 2 | Statement of the Problem | **1** |
| 3 | Objectives and scope of the project | **1** |
| 4 | Tools/Framework used | **1** |
| 5 | Methodology / Design | **2** |
| 6 | Algorithm / Pseudocode | **3** |
| 7 | Screenshots | **7** |
| 8 | Applications and Future Enhancement of the Project | **10** |
| 9 | Conclusion | **10** |
| 10 | References | **10** |

**TITLE:**

**A DATABASE FOR COFFEE PREPARATION APP.**

**STATEMENT OF THE PROBLEM:**

**TO BUILD A PYTHON PROJECT WITH A SQLite.**

OBJECTIVES AND SCOPE OF THE PROJECT:

:

1. We build a python project that uses SQ lite data base to store data.

2. we also look at how SQL simplifies inserting, retrieving and searching data.

* Here we build a Coffee App where we are going keep track of a coffee beans and keep track of the way that we’ve made the coffee with a percolater with our expresso machine or filter
* Then we will give the ratings to show whether you liked a specific bean prepared with a specific method and by seeing the ratings one can order the coffee bean

TOOLS \FRAMEWORK USED:

1. Python
2. SQLite Database
3. Connection

Methodology / Design :

In this project we build a python project that uses a SQLite databases to store

data .SQL simplifies inserting,retrieving,and searching for data from the

table.here using the python and SQLite we build a database for coffee

preparation app.sequel is language that is used to interact with the databases.in

this project the database we used is called sequel lite

SQLite is an open-source relational database i.e. used to perform database

operations on android devices such as storing, manipulating or retrieving

persistent data from the database.

It is embedded in android bydefault. So, there is no need to perform any

database setup or administration task.Here, we are going to see the example of

sqlite to store and fetch the data. Data is displayed in the logcat. For displaying

data on the spinner or listview, move to the next page.

SQLiteOpenHelper class provides the functionality to use the SQLite database.

Algorithm / Pseudocode:

Database.py:

import sqlite3

CREATE\_BEANS\_TABLE = "CREATE TABLE IF NOT EXISTS beans (id INTEGER PRIMARY KEY,name TEXT,method TEXT, rating INTEGER);"

INSERT\_BEAN = "INSERT INTO beans (name, method, rating) VALUES(?,?,?);"

GET\_ALL\_BEANS = "SELECT \* FROM BEANS;"

GET\_BEANS\_BY\_NAME = "SELECT \* FROM beans WHERE name = ?;"

GET\_BEST\_PREPARATION\_FOR\_BEAN = """"

SELECT \* from beans

WHERE name = ?

ORDER BY rating DESC

Limit 1;"""

def connect():

return sqlite3.connect("data.db")

def create\_tables(connection):

with connection:

connection.execute(CREATE\_BEANS\_TABLE)

def add\_bean(connection,name,method,rating):

with connection:

connection.execute(INSERT\_BEAN,(name,method,rating))

def get\_all\_beans(connection):

with connection:

return connection.execute(GET\_ALL\_BEANS).fetchall()

def get\_beans\_by\_name(connection,name):

with connection:

return connection.execute(GET\_BEANS\_BY\_NAME,(name,)).fetchall()

def get\_best\_preparation\_for\_bean(connection,name):

with connection:

return connection.execute(GET\_BEST\_PREPARATION\_FOR\_BEAN,(name,)).fetchone()

app.py:

import database

MENU\_PROMPT = """--Coffee Bean App--

please choose one of these options:

1)Add a new bean.

2)See all beans.

3)Find a bean by name.

4)see which preparation method is best for a bean.

5)Exit.

Your selection:"""

def menu():

connection = database.connect()

database.create\_tables(connection)

while(user\_input :=input(MENU\_PROMPT)) != "s":

if user\_input == "1":

prompt\_add\_new\_bean(connection)

elif user\_input == "2":

prompt\_see\_all\_beans(connection)

elif user\_input == "3":

prompt\_find\_bean(connection)

elif user\_input == "4":

prompt\_find\_best\_method(connection)

else:

print("Invalid input,please try again!")

def prompt\_add\_new\_bean(connection):

name = input("enter bean name:")

method = input("enter how you've prepared it:")

rating = int(input("enter your rating score (0-100): "))

database.add\_bean(connection, name, method,rating)

def prompt\_see\_all\_beans(connection):

beans = database.get\_all\_beans(connection)

for bean in beans:

print(f"{bean[1]} ({bean[2]}) - {bean[3]}/100")

def prompt\_find\_best\_method(connection):

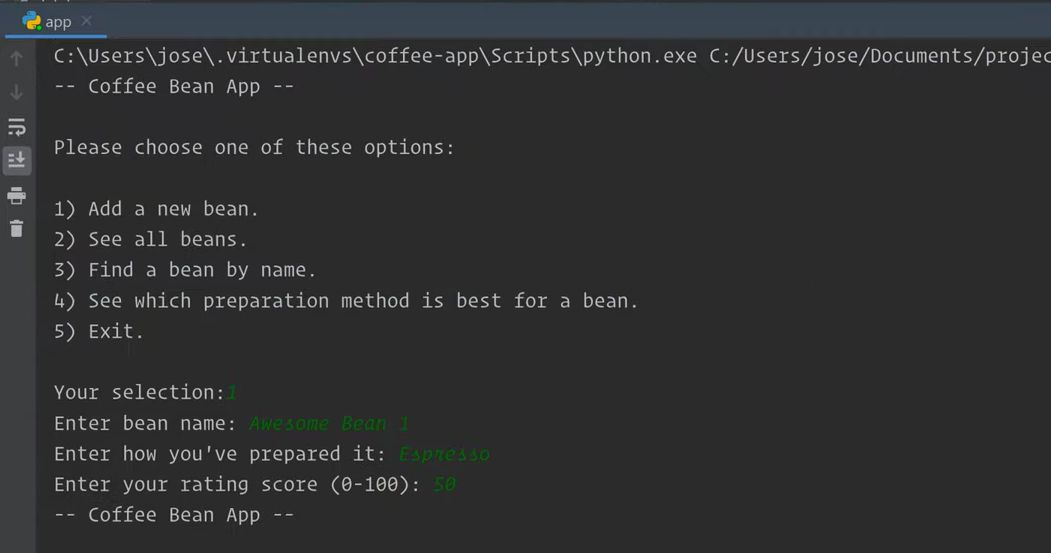
name = input("enter bean name to find: ")

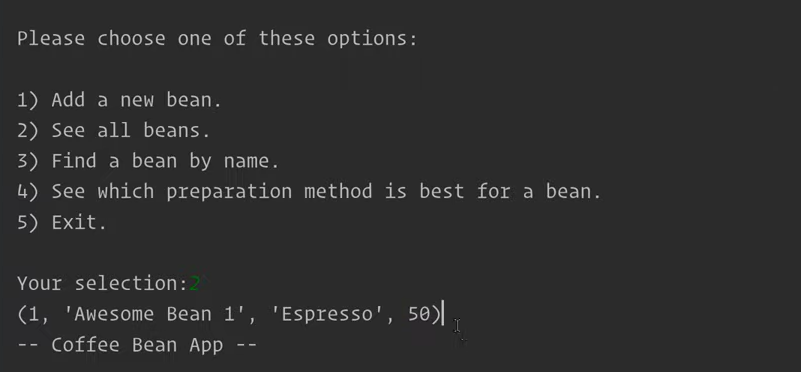
best\_method = database.get\_best\_prepartion\_for\_bean(connection, name)

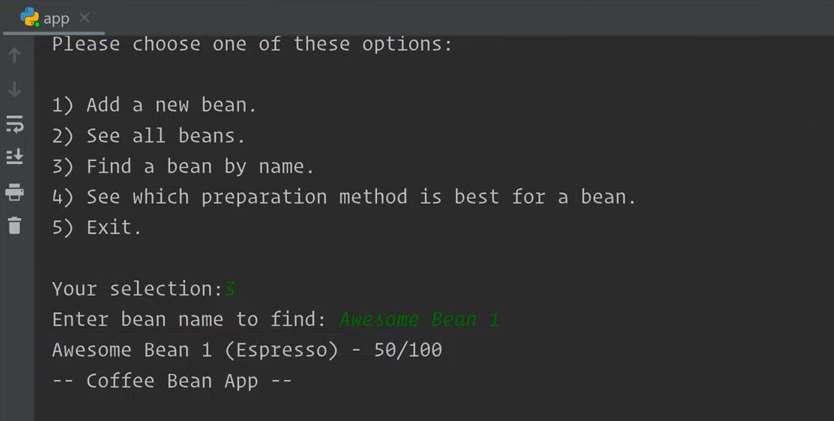
print(f"the best preparation method for {name} is: {best\_method[2]}. ")

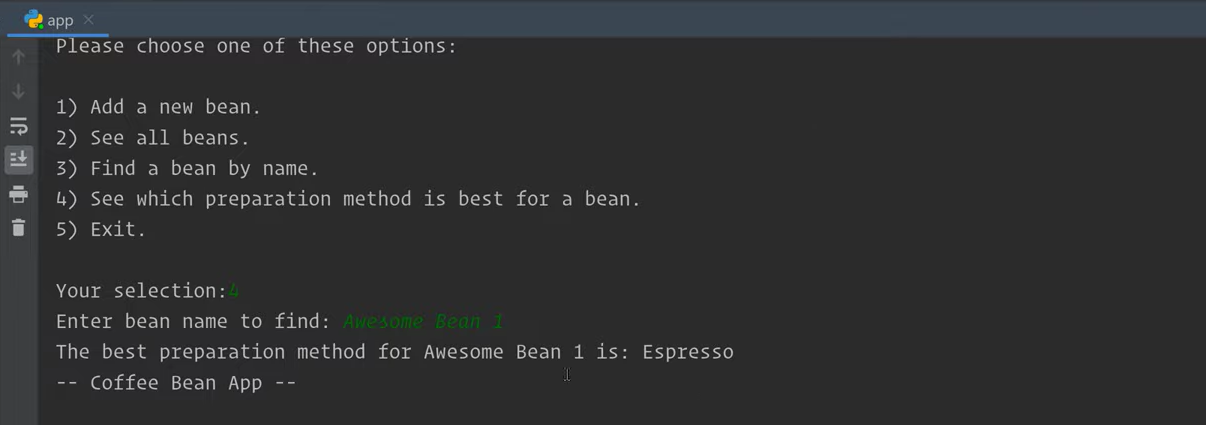
menu()

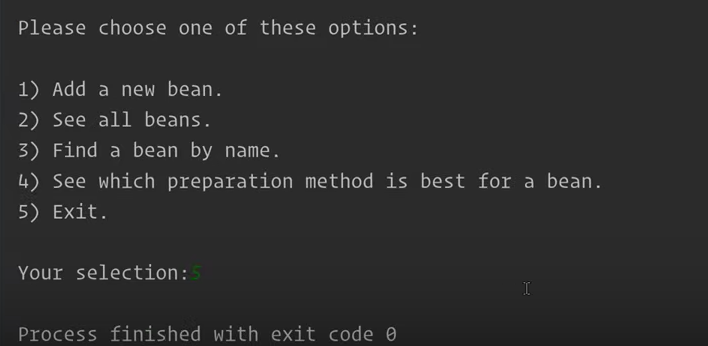
Screenshots:











Applications and Future Enhancement of the Project:

1.It is used to prepare a coffee preparation app. In which we give rating of particular coffee to show that whether we liked a specific bean.

2.It keeps track of a coffee beans we are going to keep track of the way that we have made the coffee with a percolater or machine.

3.By seeing the rating one can order the desired coffee in future.

Conclusion:

The breadth and scope of the SQL commands provide the capability to create and manipulate a wide variety of database objects using the various CREATE, ALTER, and DROP commands. Those database objects then can be loaded with data using commands such as INSERT. The data can be manipulated using a wide variety of commands, such as SELECT, DELETE, and TRUNCATE, as well as the cursor commands, DECLARE, OPEN, FETCH, and CLOSE.

REFERENCES:

https://www.geeksforgeeks.org/