Name: **Ahmed Nawaz**
Roll #: **23P-0550**
Section: **4A**

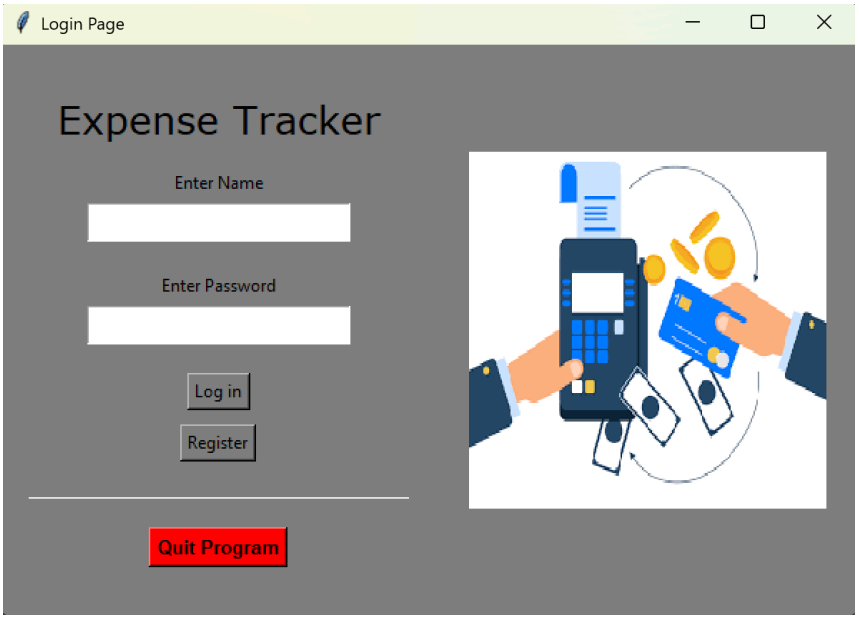
Database Systems Assignment #02

The submitted assignment contains:

1. **main.py file;** containing the GUI logic, and database handling logic
2. **database.py:** Contains the lines of code to create the tables and columns for the database, can be run multiple times, due to the 'CREATE IF NOT EXISTS' line.
3. An image containing the ERD for the tables
4. **money.jpeg:** An image for the front page of the log-in page (tried giving the front page a decent look)
5. **Demonstration Video;** A video, demonstrating how the system works and interacts with the database.
6. A pdf file, you're looking at it. 🙂.
7. **Cash_logger2.db;** containing the database and tables

System Overview

Upon running the program, you are provided with two options, either registering for a new account or logging into a pre-existing account.

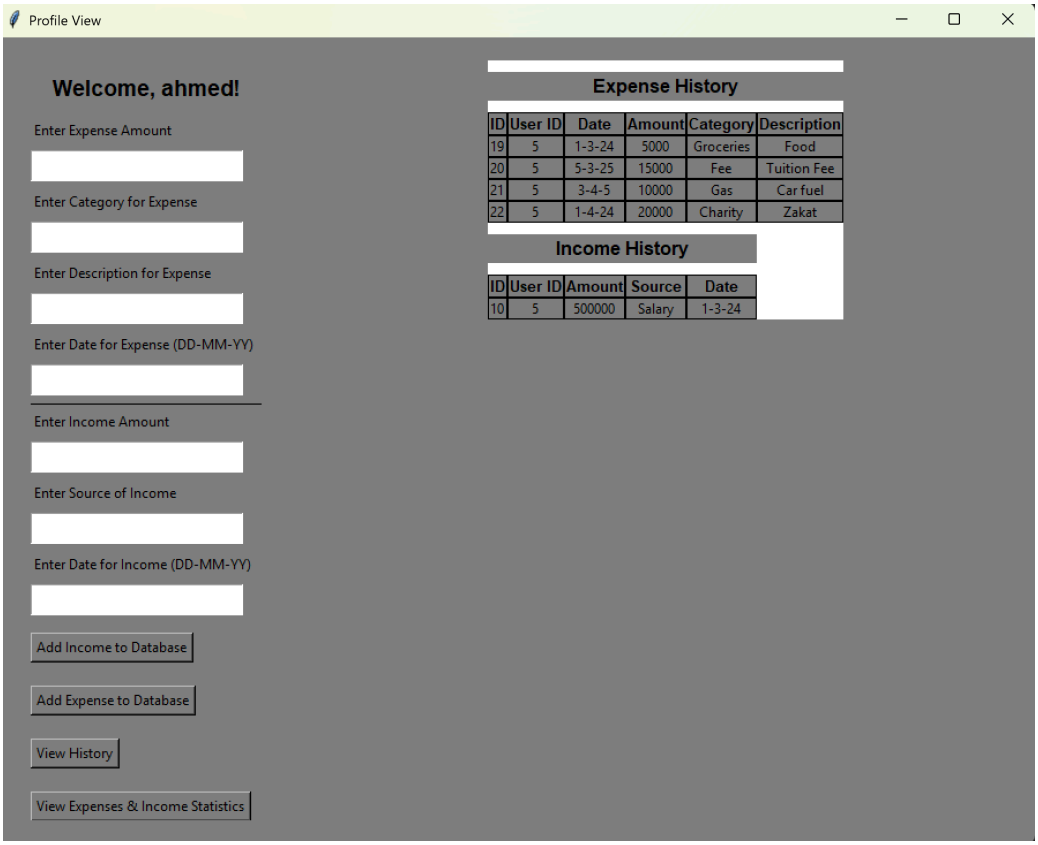


Since we have already created an account in the past, we will simply log into the account; with the following credentials

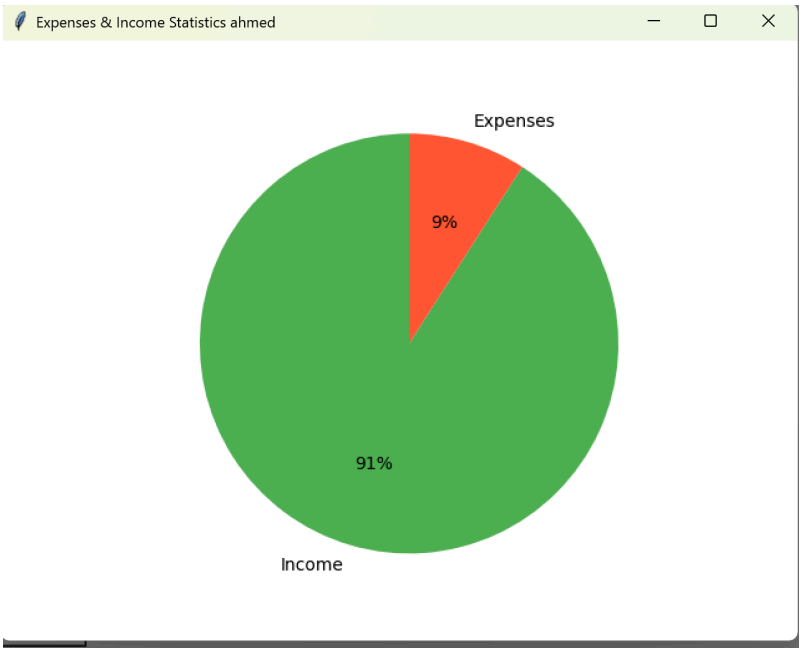
	ID	name	password
1	5	ahmed	12345

Again, because we tested our program at every step of the way, we already have pre-existing data inserted into the tables, though the video will demonstrate its proper working and functionality.
We are greeted with a ‘Welcome’ message along with the user’s name. Expense and Income Data can be seen displayed at the right side of the screen, each segregated from the other. Moreover, you can see input blocks on the left side of the screen, expense and income also segregated by a small black line. At the bottom we see 4 buttons, to enter data into the income table, to enter data into the expense table, to refresh the display page; when entering new data, it won’t display itautomatically, the

refresh button has to be clicked. And another button to show the stats for expense and income



Clicking on the '**View Expenses & Income Statistics**' button results in a window popping out displaying a pie chart displaying the Expense Breakdown by Category, and how much of it is used out of total income. Although the percentage values are off by a tiny bit, E.g Expenses here should be 10%, since the sum of total expenses is 50,000 , and sum of total income is 500,000. We tried fixing this error, but we had no success, and since it is only a tiny bit inaccurate, we let it slide.



<u>This is the end of the road, and that is all there is to the program.</u>

Now for the database. The database has the following tables;
Expenses
Income
Users

Users

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	ID	INTEGER	🔑							*NULL*
2	name	TEXT								*NULL*
3	password	TEXT								*NULL*

Income

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	ID	INTEGER								NULL
2	User_ID	INTEGER								NULL
3	Amount	INTEGER								NULL
4	Source	TEXT								NULL
5	Date	TEXT								NULL

Expenses

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	ID	INTEGER								NULL
2	User_ID	INTEGER								NULL
3	Date	TEXT								NULL
4	Amount	INTEGER								NULL
5	Category	TEXT								NULL
6	Description	TEXT								NULL

Created by the following lines of code (In the database.py file)

```python
# Database Setup
conn = sqlite3.connect("Cash_logger.db")
c = conn.cursor()
c.execute("""CREATE TABLE IF NOT EXISTS User (
          ID INTEGER PRIMARY KEY AUTOINCREMENT,
          name TEXT,
          password TEXT)""")
c.execute("""CREATE TABLE IF NOT EXISTS Expenses (
          ID INTEGER PRIMARY KEY AUTOINCREMENT,
          User_ID INTEGER,
          Date TEXT,
          Amount INTEGER,
          Category TEXT,
          Description TEXT,
          FOREIGN KEY (User_ID) REFERENCES User(ID))""")
c.execute("""CREATE TABLE IF NOT EXISTS INCOME (
                ID INTEGER PRIMARY KEY AUTOINCREMENT,
                User_ID INTEGER,
                Amount INTEGER ,
                Source TEXT ,
                Date TEXT ,
                FOREIGN KEY (User_ID) REFERENCES User(ID))""")
conn.commit()
conn.close()

root.mainloop()
```