

# **Expense Tracker**

Mini-project-1B report submitted in partial  
fulfilment of the requirements of the degree of

## **Information Technology**

by

**Yash Nandkumar Patil    EU1194050 (33)**

**Abhijeet Digvijay Singh    EU1194049 (48)**

**Sridhar Srinivasan    EU1194047 (49)**

**Siddhi Rajesh Shirke    EU1194048 (47)**

Under the guidance of

**Ms. Shraddha S. More**  
**Assistant Professor**



**Department of Information Technology**

**St. John College of Engineering and Management, Palghar**

**University of Mumbai**

**(A.Y. 2020-2021)**

# **CERTIFICATE**

This is to certify that the S.E. mini-project entitled **“EXPENSE TRACKER”** is a bonafide work of

**“Yash N. Patil” (EU1194050) (33) , “Abhijeet D. Singh” (EU1194049) (48), “Sridhar Srinivasan” (EU1194047) (49) and “Siddhi R. Shirke” (EU1194048) (49)** submitted to University of Mumbai in partial fulfilment of the requirement for the award of the degree of **“Information Technology Engineering”** during the academic year 2020-2021.

**Ms. Shraddha More**

Guide

**Mrs. Anita Chaudhari**  
Head of Department

**Dr. G.V. Mulgund**  
Principal

# Mini-Project-1B Report Approval

This mini-project synopsis entitled **Expense Tracker** by **Yash N.patil, Abhijeet D. Singh, Sridhar Srinivasan, Siddhi R. Shirke** is approved for the degree of **Information Technology Engineering** from **University of Mumbai**.

## Examiners

1.-----

2.-----

Date:

Place:

# Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-----

Signature

**Yash N. Patil (EU1194050)**

-----

Signature

**Abhijeet D. Singh (EU1194049)**

-----

Signature

**Sridhar Srinivasan (EU1194047)**

-----

Signature

**Siddhi R. Shirke (EU1194048)**

Date:

## Abstract

- This project has been developed to cater to the dire need for an application that facilitates easy analysis of our expenses. Once aware of our expenses, we can not only keep a check but also cut down on the unrequired expenses
- This project is developed using Python at front end and MySQL at backend used for database connection
- This application will allow users to track their expenses in terms of dashboard and pie-chart
- Unique features like notes and FAQs have been introduced coupled with features like dashboard, add income/expense, delete income/expense, statistics and users to enrich the user's experience
- The system is designed such that it is an interactive and content management system dealing with creation, insertion and deletion of data
- Thus, above features of this project will help in providing a responsible lifestyle by ensuring a balance between income and expenses. It will also help in deciding future investments without worrying about savings

**Keywords** — *expenses, budget, efficiency, responsible*

## **Acknowledgments**

We would like to express our gratitude and appreciation to all those who gave us the encouragement to complete this project. A special thanks to our guide Ms. Shraddha More, whose help, suggestions and encouragement in supervising our project was valuable. Thank you for allowing us to carry out this challenging task. We welcomed her suggestions and recommendations. Many thanks go to Dr. G. V. Mulgund, Principal, SJCEM and Mrs. Anita Chaudhari, Head of Department, Information Technology Engineering. We are extending our gratitude to our family members, friends and well-wishers who supported us morally. We will forever remain grateful to all of you.

## List of Figures

<b>1</b>	<b>Home</b>
<b>2</b>	<b>Login</b>
<b>3</b>	<b>Signup</b>
<b>4</b>	<b>Menu</b>
<b>5</b>	<b>Dashboard</b>
<b>6</b>	<b>Add Income/Expense</b>
<b>7</b>	<b>Statistics</b>
<b>8</b>	<b>Delete Income/Expense</b>
<b>9</b>	<b>Notes</b>
<b>10</b>	<b>Users</b>
<b>11</b>	<b>FAQs</b>

	<b>Abstract</b>
	<b>Acknowledgments</b>
	<b>List of Figures</b>
<b>Chapter 1</b>	<b>Introduction</b>
	1.1 Introduction
	1.2 Motivation
	1.3 Problem Statement & Objective
	1.4 Scope
<b>Chapter 2</b>	<b>Review of Literature</b>
<b>Chapter 3</b>	<b>Proposed System</b>
	3.1 Block Diagram
	3.2 Algorithm
	3.3 Requirement of Hardware & Software
<b>Chapter 4</b>	<b>Results and Discussion</b>
	4.1 Flowchart
	4.2 Pseudo code
	4.3 Screenshots of the output with description
<b>Chapter 5</b>	<b>Conclusion</b>
	<b>References</b>



# **Chapter 1**

## **Introduction**

### **1.1 Introduction:**

- There is a dire need for an application that facilitates easy analysis of our expenses.
- Once aware of our expenses, we can not only keep a check but also cut down the unrequired expenses, and this will help in providing a responsible lifestyle.
- An application which is simple and user-friendly at the same time robust and efficient is the need of hour
- The project aims to develop an Expense Tracker that helps its users to keep track of their expenses and determine whether they are spending as per their set budget.

## **1.2 Motivation:**

- We thought to develop an application that tracks Expenses amid dearth for better Expense Trackers and an unreliable pen and paper approach
- This application will allow users to track their expenses in terms of dashboard and pie-chart
- Thus, on successful completion and implementation of this project, its features will help overcome challenges posed by conventional Expense Trackers and pen-paper approach

### **1.3 Problem Statement:**

- Once aware of our expenses, we can not only keep a check but also cut down the unrequired expenses
- Tracking expenses using pen and paper approach is tedious. Calculation mistakes, missing out on entries and incorrect data entry are some of the pitfalls
- Also pen and paper are difficult to carry everywhere. Expense trackers have potential to provide statistics such as pie-chart which are difficult to create in pen and paper approach
- Hence there is a dire need for an expense tracker that helps to analyze our income and expense
- Existing expense trackers fail to provide all features under single application

### **1.3 Objectives:**

- To overcome challenges posed by traditional pen and paper approach
- To provide a better alternative to weaker existing applications
- To provide a safer environment protected by a password
- To provide a simple user interface
- To create a foundation for further enhancement of Expense Trackers

## 1.4 Scope:

The scope of this project is to provide easy hassle-free tracking of expenses. This application will help overcome challenges posed by traditional pen and paper approach as well as weak existing applications. Users can create their accounts which will be protected by password set by them. Menu will have 8 options viz.

1. **Dashboard:** shows information of 5 latest income/expense inputs
2. **Add income/expense:** to add an income/expense to account
3. **Delete income/expense:** to delete an income/expense from account
4. **Statistics:** provides statistics in the form of pie-chart
5. **Notes:** to create short notes on transactions performed
6. **Users:** to view various users
7. **FAQs:** consists some of the Frequently Asked Questions
8. **Logout:** to logout of your account

## Chapter 2

### Review of Literature

[1]

**Title:** Online Income and Expense Tracker.

**Author Names:** S. Chandini, T. Poojitha, D. Ranjith, V. J. Mohammed Akram, M. S. Vani, V. Rajyalakshmi

**Journal Name with Year:** International Research Journal of Engineering and Technology (IRJET) | Volume: 06 | Issue: 03 | March-2019.

**Description:**

Income and Expense Tracker will maintain data of daily, weekly, monthly, yearly expenses, Manages your expenses and earnings in a simple and intuitive way. User can select category of expense, enter other information like user can capture photo, add location, select amount of expense etc. And this will save to the local database. User can view and sort expense as per weekly, monthly, yearly. By using this, we can reduce the manual calculations for their expenses and keep the track of the expenditure. In this, user can provide his income to calculate his total expenses per day and these results will stored for unique user. People when usually go for trips or movies with friends they can use this tracker to maintain their expense. It will be easy for them to share the bill in this tracker. This will display graph as per selected view. And user can enter his monthly income or limit of monthly Expense in this tracker.

**Advantages:**

1. This application provides statistics in the form of pie-charts
2. This application is easier to use for a naïve user

**Disadvantages:**

1. This application does not allow updation of already existing data
2. This application does not have an attractive user interface

[2]

**Author Names:** Md. Abdul Karim ,Taslima Yesmin Orin,TasnimAhmed,Ahmed Al Marouf

**Journal Name With Year:** DAFFODIL INTERNATIONAL UNIVERSITY, DHAKA, BANGLADESH |September 2019|

**Description:**

Daily Expense is a simple application for Android devices. A Daily Expense Tracker is a one kind of digital diary that helps to keep an eye on all of our money related transitions and also provides all financial activities report daily, weekly, monthly and yearly. All information is saved in offline mode so users can easily access any time and any places. User interface of the Daily Expense Tracker is very simple and attractive so it is easy to understand and the best way to record our financial data. In our daily life money is the most important portion and without we cannot last one day in the earth but if we keep on track all financial data then we can overcome this problem. Most of the people cannot track their expense and income one way they face in a money crisis and depression. This situation motivates us to make an android app to track all financial activities. Using the Daily Expense Tracker user can be tracking expenses day to day and making life tension free.

**Advantages:**

- 1) This application displays statistics in the form of pie-charts
- 2) Allows representation of data in form of pie chart

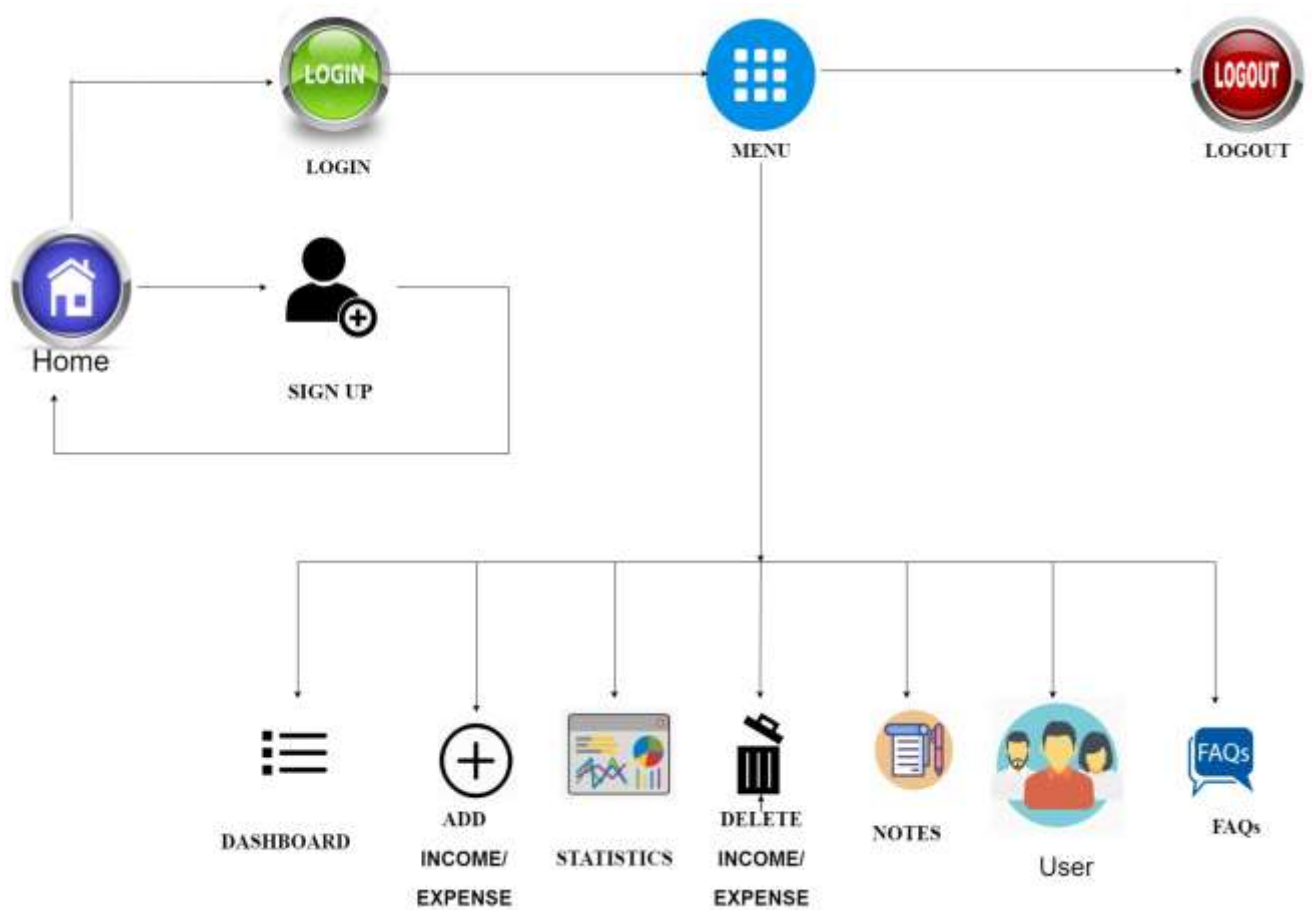
**Disadvantages:**

- 1) This application does not have an option of notes
- 2) This application is difficult for a naive user to access

# Chapter 3

## Proposed System

### 3.1 Block Diagram:



### **3.2 Algorithm:**

Step 1: Start

Step 2: on Home page, choose between New or Existing User and Exit

Step 3: if New User, go to Step 4 else go to Step 6

Step 4: if signup successful go to Step 2

Step 5: else try to signup again

Step 6: if login successful, go to Step 8 (Menu)

Step 7: if login unsuccessful, try again

Step 8: Choose from Dashboard, Add Income/Expense, Statistics, Delete Income/Expense, Notes, Users, FAQs, Logout

Step 9: if Dashboard, it displays last 5 expenses

Step 10: if Add Income/Expense, it lets you add income/expense

Step 11: if Statistics, it displays expenses in form of pie-chart

Step 12: if Delete Income/Expense, it lets you delete an already existing income/expense

Step 13: if Notes, it lets you add quick notes

Step 14: if Users, it lets you see all the users

Step 15: if FAQs, it shows answers to Frequently Asked Questions

Step 16: if Logout, it logs you out of the application and go to Step 17

Step 17: Stop



### **3.3 Hardware and Software Requirements**

#### **Hardware**

Intel(R) pentium (R) CPU G3220

4GB Ram

1 TB Harddisk

Window 7 Ultimate

#### **Software**

VS Code

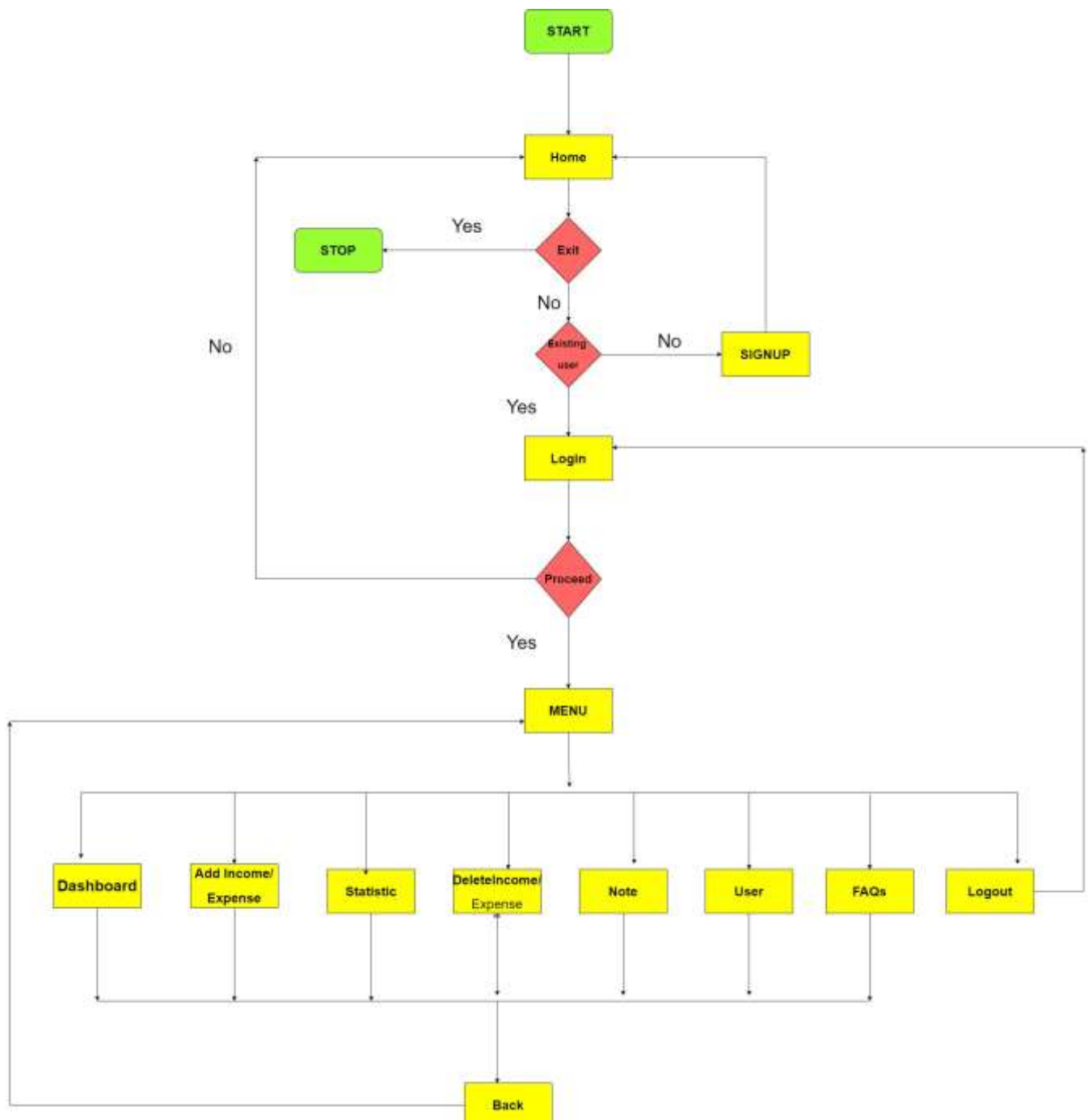
MySql Workbench 8.0 CE

Mysql.connector Python

# Chapter 3

## Results and Discussion

### 4.1 Flowchart



## 4.2 Pseudo code

```
import tkinter
import os
from tkinter import *
import tkinter.messagebox as tmsg
from tkinter.messagebox import *
from tkinter.filedialog import *
from typing import AsyncGenerator
from datetime import *
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from tkinter import ttk
t=Tk()
t.title('EXPENSE TRACKER')
t.geometry("600x400")
t.configure(bg="beige")
t=Text(t, width=40, height= 1, font=("Arial serif", "30", "bold"),
fg="navy", bg="#45aaf2", wrap=WORD)
t.insert(END,"EXPENSE TRACKER")
t.pack(side=TOP)

#-----

def login():
    f1=Frame(bg="#45aaf2",padx=125,pady=60)
    f1.place(width=600, height=400)

    l1=Label(f1,text='Username:',width=14, height=1, font=("Arial
serif", "14", "bold"))
    l2=Label(f1,text='Password:',width=14, height=1, font=("Arial
serif", "14", "bold"))
    e1=Entry(f1,width=14,font=("Arial serif", "15", "bold"))
    e2=Entry(f1,width=14,show='*',font=("Arial serif", "15", "bold"))
```

```

def verify():
    e1.get()
    e2.get()
    import mysql.connector as c
    con=c.connect(host='localhost', user='root',
passwd='abhimysql@13B', database='expense')
    cursor=con.cursor()
    cursor.execute("select pwd from signup")
    record=cursor.fetchall()
    for i in record:
        if i[0]==e2.get():
            tmsg.showinfo("Login", "Login Successful ...")
        else:
            tmsg.showinfo("Login Unsuccessful", "Please, enter valid
credentials!")

    b1=Button(f1,text='LOGIN',width=15,height=2,fg="white",
bg="blue", activebackground="lightyellow",font=("Arial serif", "12",
"bold"),command=verify)
    b2=Button(f1,text='BACK',width=15,height=2,fg="white",
bg="red", activebackground="lightyellow",font=("Arial serif", "12",
"bold"),command=f1.destroy)
    b3=Button(f1,text='PROCEED',width=15,height=2,fg="black",
bg="green2", activebackground="lightyellow",font=("Arial serif",
"12", "bold"), command=menu)
    l1.grid(row=0,column=0,padx=10)
    e1.grid(row=0,column=2)
    l2.grid(row=1,column=0,pady=10,padx=10)
    e2.grid(row=1,column=2)
    b1.grid(row=2,column=0,pady=20)
    b2.grid(row=2,column=2,pady=20)
    b3.grid(row=3,column=0)

#-----

def signup():
    f2=Frame(bg="#45aaf2",padx=115,pady=40)

```

```
f2.place(width=600, height=400)
```

```
username=StringVar
```

```
pwd=StringVar
```

```
age=StringVar
```

```
mobile_no=StringVar
```

```
l1=Label(f2,text='Username:',width=14,height=1, font=("Arial serif", "14", "bold"))
```

```
l2=Label(f2,text='Password:',width=14,height=1, font=("Arial serif", "14", "bold"),pady=10)
```

```
l3=Label(f2,text='Confirm Password:',width=14,height=1, font=("Arial serif", "14", "bold"),pady=10)
```

```
l4=Label(f2,text='Age:',width=14,height=1, font=("Arial serif", "14", "bold"),pady=10)
```

```
l5=Label(f2,text='Gender:',width=14,height=1, font=("Arial serif", "14", "bold"),pady=10)
```

```
l6=Label(f2,text='Mobile Number:',width=14,height=1, font=("Arial serif", "14", "bold"),pady=10)
```

```
e1=Entry(f2,textvariable="username",width=14,font=("Arial serif", "14", "bold"))
```

```
e2=Entry(f2,width=14,show='*', font=("Arial serif", "14", "bold"))
```

```
e3=Entry(f2,width=14,textvariable="pwd",show='*',font=("Arial serif", "14", "bold"))
```

```
e4=Entry(f2,width=14,textvariable="age",font=("Arial serif", "14", "bold"))
```

```
i=StringVar()
```

```
r1=Radiobutton(f2,text='Male', value="male", variable=i, font=("Arial serif", "14", "bold"),bg="lightyellow", activebackground="dodger blue")
```

```
r2=Radiobutton(f2,text='Female',value="female", variable=i, font=("Arial serif", "14", "bold"),bg="lightyellow", activebackground="hotpink")
```

```
e6=Entry(f2,textvariable=" mobile_no",width=14, font=("Arial serif", "14", "bold"))
```

```
e1.get()
```

```
e2.get()
```

```
e3.get()
```

```
e4.get()
```

```
e6.get()
```

```
def signpro():  
    if e2.get()==e3.get():  
        username=e1.get()  
        age=int(e4.get())  
        gender=i.get()  
        mobile_no=int(e6.get())  
        pwd=e3.get()  
        import mysql.connector as c  
        con=c.connect(host='localhost', user='root',  
passwd='abhimysql@13B', database='expense')  
        cursor=con.cursor()  
        query="Insert into signup values(%s, %s, %s, %s, %s)"  
        tup=(username,age,gender,mobile_no,pwd)  
        cursor.execute(query,tup)  
        con.commit()  
        tmsg.showinfo("SIGNUP", "Signup Successful ...")  
    else:  
        tmsg.showinfo("SIGNUP Signup Unsuccessful ...", "Password  
and Confirm Password are not same")
```

```
b1=Button(f2,text='ADD  
ACCOUNT',width=15,height=2,fg="white", bg="blue",  
activebackground="lightyellow",font=("Arial serif", "12", "bold"),  
command=signpro)
```

```
b2=Button(f2,text='BACK',width=15,height=2,fg="white",
bg="red", activebackground="lightyellow", font=("Arial serif", "12",
"bold"), command=f2.destroy)
```

```
l1.grid(row=0,column=0)
e1.grid(row=0,column=3)
l2.grid(row=1,column=0)
e2.grid(row=1,column=3)
l3.grid(row=2,column=0)
e3.grid(row=2,column=3)
l4.grid(row=3,column=0)
e4.grid(row=3,column=3)
l5.grid(row=5,column=0)
r1.grid(row=5,column=2)
r2.grid(row=5,column=3)
l6.grid(row=7,column=0)
e6.grid(row=7,column=3)
b1.grid(row=9,column=0,pady=15)
b2.grid(row=9,column=3,pady=15)
```

#-----

```
def menu():
```

```
    f3=Frame(bg="#45aaf2",padx=70,pady=60)
    f3.place(width=600, height=400)
```

```
    b1=Button(f3,text='DASHBOARD',width=23,height=2,
bg="green2", activebackground="azure",font=("Arial serif", "12",
"bold"),command=dash)
```

```
    b2=Button(f3,text='ADD
INCOME/EXPENSE',width=23,height=2, bg="orange",
activebackground="azure",font=("Arial serif", "12", "bold"),
command=addie)
```

```
    b3=Button(f3,text='STATISTICS',width=23,height=2,
bg="orange",font=("Arial serif", "12", "bold"),
activebackground="azure",command=stats)
```

```

b4=Button(f3,text='DELETE
INCOME/EXPENSE',width=23,height=2, bg="green2",
activebackground="azure",font=("Arial serif", "12", "bold"),
command=delie)
b5=Button(f3,text='NOTES',width=23,height=2,bg="green2",
activebackground="azure",font=("Arial serif", "12",
"bold"),command=notes)
b6=Button(f3,text='USERS',width=23,height=2, bg="orange",
activebackground="azure",font=("Arial serif", "12",
"bold"),command=users)
b7=Button(f3,text='FAQs',width=23,height=2,fg="white",
bg="blue", activebackground="lightyellow",font=("Arial serif", "12",
"bold"),command=faqs)
b8=Button(f3,text='LOGOUT!',width=23,height=2,fg="white",
bg="red", activebackground="lightyellow",font=("Arial serif", "12",
"bold"), command=login)
b1.grid(row=0,column=0)
b2.grid(row=0,column=1,padx=20)
b3.grid(row=2,column=0,pady=10)
b4.grid(row=2,column=1,pady=10)
b5.grid(row=4,column=0,pady=10)
b5.grid(row=4,column=0,pady=10)
b6.grid(row=4,column=1,pady=10)
b7.grid(row=6,column=0,pady=10)
b8.grid(row=6,column=1,pady=10)

```

#-----

```

def addie():
    f4=Frame(bg="#45aaf2",padx=110,pady=30)
    f4.place(width=600, height=400)
    username=StringVar
    dt=StringVar
    bname=StringVar
    amount=StringVar
    category=StringVar
    pmode=StringVar

```



```
income_expense=StringVar
l1=Label(f4,text='Username:',width=14,height=1, font=("Arial
serif", "14", "bold"),pady=5)
l2=Label(f4,text='Date (yy/mm/dd)',width=14,height=1,
font=("Arial serif", "14", "bold"),pady=5)
l3=Label(f4,text='Name',width=14,height=1, font=("Arial serif",
"14", "bold"),pady=5)
l4=Label(f4,text='Category:',width=14,height=1, font=("Arial
serif", "14", "bold"),pady=5)
l5=Label(f4,text='Amount (in ₹):',width=14,height=1, font=("Arial
serif", "14", "bold"),pady=5)
l6=Label(f4,text='Mode:',width=14,height=1, font=("Arial serif",
"14", "bold"),pady=5)
```

```
i=StringVar()
```

```
r1=Radiobutton(f4,text='Income', value="income", variable=i,
font=("Arial serif", "14", "bold"),bg="lightyellow",
activebackground="green2")
r2=Radiobutton(f4,text='Expense',value="expense", variable=i,
font=("Arial serif", "14", "bold"),bg="lightyellow",
activebackground="orange")
e1=Entry(f4,width=14,textvariable="username", font=("Arial
serif", "14", "bold"))
e2=Entry(f4,width=14,textvariable="dt", font=("Arial serif", "14",
"bold"))
e3=Entry(f4,width=14,textvariable="bname", font=("Arial serif",
"14", "bold"))
spin1= Spinbox(f4, textvariable=StringVar, values=('Food',
'Education', 'Medical', 'Travel', 'Entertainment', 'Bills', 'Grocery',
'Personal', 'Salary', 'Miscellaneous'), width=13,font=("Arial serif",
"14", "bold"))
e4=Entry(f4,width=14,textvariable="amount",font=("Arial serif",
"14", "bold"))
spin2= Spinbox(f4, textvariable=StringVar, values=('Cash', 'Card',
'UPI', 'NEFT', 'RTGS', 'IMPS', 'Salary'), width=13,font=("Arial serif",
"14", "bold"))
```

```

def addiepro():
    username=e1.get()
    dt=e2.get()
    bname=e3.get()
    category=spin1.get()
    income_expense=i.get()
    amount=int(e4.get())
    pmode= spin2.get()
    import mysql.connector as c
    con=c.connect(host='localhost', user='root',
passwd='abhimysql@13B', database='expense')
    cursor=con.cursor()
    query="Insert into inex values(%s, %s, %s, %s, %s, %s, %s)"
    tup=(username, dt, bname, category, income_expense, amount,
pmode)
    cursor.execute(query,tup)
    con.commit()
    tmsg.showinfo("ADD Income/Expense", "Added successfully
...")

```

```

b1=Button(f4,text='ADD
INOME/EXPENSE',width=20,height=2,fg="white", bg="blue",
activebackground="lightyellow",font=("Arial serif", "12",
"bold"),command=addiepro)
b2=Button(f4,text='BACK',width=15,height=2,fg="white",
bg="red", activebackground="lightyellow",font=("Arial serif", "12",
"bold"), command=menu)

```

```

r1.grid(row=0,column=0,pady=10)
r2.grid(row=0,column=2,pady=10)

```

```

l1.grid(row=1,column=0)
e1.grid(row=1,column=2)
l2.grid(row=2,column=0)
e2.grid(row=2,column=2)
l3.grid(row=3,column=0)

```

```
e3.grid(row=3,column=2)
l4.grid(row=4,column=0)
spin1.grid(row=4,column=2)
l5.grid(row=5,column=0)
e4.grid(row=5,column=2)
l6.grid(row=6,column=0)
spin2.grid(row=6,column=2)
```

```
b1.grid(row=7,column=0,pady=20)
b2.grid(row=7,column=2,pady=20,padx=40)
```

```
#-----
```

```
def delie():
    f9=Frame(bg="#45aaf2",padx=85,pady=50)
    f9.place(width=600, height=400)
    val = ['username' , 'dt', 'bname', 'category', 'income_expense',
'amount', 'pmode']
    c1 = ttk.Combobox(f9, value = val)
    E1 = Entry(f9,width=14,textvariable="username", font=("Arial
serif", "14", "bold"))
    def deliepro():
        get_c1 = c1.get()
        get_E1 = E1.get()

import mysql.connector as c
mydb = c.connect(host = "localhost", user = "root", passwd =
"abhimysql@13B",database = "expense",)
    cur = mydb.cursor()
    delt = "DELETE from inex where %s='%s'"%(get_c1,get_E1)
    cur.execute(delt)
    mydb.commit()

E1.grid(row = 0 , column = 1)
```

```

b1=Button(f9,text='REMOVE
INCOME/EXPENSE',width=25,height=2,fg="white", bg="blue",
activebackground="lightyellow",font=("Arial serif", "12",
"bold"),command=deliepro)
b2=Button(f9,text='BACK',width=15,height=2,fg="white",
bg="red", activebackground="lightyellow",font=("Arial serif", "12",
"bold"), command=menu)
c1.grid(row=1,column=0)
E1.grid(row=1,column=2)
b1.grid(row=4,column=0,pady=140)
b2.grid(row=4,column=2,pady=140,padx=40)

```

#-----

```
def faqs():
```

```

f5=Toplevel()
f5.geometry=("100x100+100+100")

```

```

t=Text(f5, width=50, height=35, font=("Arial serif", "14", "bold",
"italic"), fg='darkblue', bg='white', wrap=WORD)

```

```

t.insert(END, "1. How to 'Sign Up'?
If you are a new user click on
the new user button. It takes you to the sign up page. On the sign up
page enter your 'username' , 'password', 'confirm password'
, 'Age', 'Gender', 'Mobile Number'. After entering all your credentials,
click on the 'Add account' button. After clicking on the Add account
button a message pops up which informs about MSG.
2. How to
'Login'?
If You are an existing user , then click on the 'existing user'
button. It takes you to the Login Page . On the login page enter your
'username' and 'password'. After entering all your credentials, click on
the 'Login 'button. After clicking on the Login button a message is
generated of successful login then clicking on the ' Proceed' button .
It takes you to 'Menu' page.
3. How does the 'Dashboard' button
work?
Dashboard displays the information of the last five
transactions performed by the user. It consists of ‘username’, ‘date’,
‘name’, ‘category’, ‘income/expense’, amount and mode of payment.
4. How to Add Income/Expense ?
Click on the 'Add
Income/Expense' button. If you want to add income select 'Income Or

```

for expense select 'Expense'. Now enter 'username', 'date in the form of (YY/MM/DD)', 'name' and preset 'categories' of transaction, its 'amount' and preset 'payment modes'. After that click on 'Add Income/Expense Button'. A message informing about the addition of income/expense will be displayed on the screen.

5. How does the 'Statistics' button work?

Statistics button displays a pie-chart formed by category and amount.

6. How to Delete Income/Expense?

Click on Delete Income/Expense button. To delete income/expense Enter values of any of username, date, name category, income/expense, amount, payment modes in the entry widget.

Now click on the Remove Income/Expense button. A message informing about the removal of income/expense will be displayed on the screen.

7. What is 'Notes'?

Notes is a special feature in the menu that provides an option note down a few important things before entering them. It has three menus viz. File, Edit and About. File menu contains New, Open, Save and Exit. New option allows us to create a new note. Open option opens an already existing note. Save option saves working unsaved notes. Exit option facilitates exit from Notes. Edit menu contains Cut, Copy and Paste. Cut option allows you to cut a portion of a sentence. Copy option allows you to copy a portion of a sentence. Paste option facilitates pasting of already cut and copied portions of sentence. About option displays information about Notes.

8. What is 'Users'?

Users button displays non-confidential information pertaining to all the users who have signed up into the application.

9. What are FAQs?

FAQs stands for Frequently Asked Questions. FAQs consist of the most common possible issues that naive users might face while using this application. They facilitate smooth, easier and faster access to the application.

10. What is 'Logout'?

Logout logs you out of the application and takes you to the login page.")

```
t.pack(side=TOP)
```

```
s=Scrollbar(f5, bg="black", orient=VERTICAL,
command=t.yview)
s.pack(side=RIGHT, fill=Y)
```

#-----

```
def dash():
    f7=Frame(bg="#45aaf2",padx=20,pady=70)
    f7.place(width=600, height=400)
    l1=Label(f7,text="Username: ",width=11,height=1, font=("Arial
serif", "10", "bold"))
    l2=Label(f7,text="Date: ",width=11,height=1, font=("Arial serif",
"10", "bold"))
    l3=Label(f7,text='Name: ',width=11, height=1, font=("Arial serif",
"10", "bold"))
    l4=Label(f7,text='Category: ',width=11,height=1, font=("Arial
serif", "10", "bold"))
    l5=Label(f7,text='Income/Expense: ',width=13,height=1,
font=("Arial serif", "10", "bold"))
    l6=Label(f7,text='Amount (in ₹): ',width=11,height=1, font=("Arial
serif", "10", "bold"))
    l7=Label(f7,text='Mode: ',width=11,height=1, font=("Arial serif",
"10", "bold"))
    tmsg.showinfo("DASHBOARD", "Displaying last 5 transactions
...")

import mysql.connector as c
con=c.connect(host='localhost', user='root',
passwd='abhimysql@13B', database='expense')
cursor=con.cursor()
cursor.execute("SELECT * FROM inex limit 0,5")
i=0
for inex in cursor:
    for j in range(len(inex)):
        e = Entry(f7, width=13, fg='blue')
        e.grid(row=i+2, column=j)
        e.insert(END, inex[j])
    i=i+1
```

```
b1=Button(f7,text='BACK',width=10,height=2,fg="white",
bg="red", activebackground="lightyellow",font=("Arial serif", "10",
"bold"),command=menu)
```

```
l1.grid(row=1,column=0)
l2.grid(row=1,column=1)
l3.grid(row=1,column=2)
l4.grid(row=1,column=3)
l5.grid(row=1,column=4)
l6.grid(row=1,column=5)
l7.grid(row=1,column=6)
b1.grid(row=10,column=0,pady=15)
```

```
#-----
```

```
def users():
```

```
    f8=Frame(bg="#45aaf2",padx=20,pady=70)
    f8.place(width=600, height=400)
    l1=Label(f8,text="Username: ",width=14,height=1, font=("Arial
serif", "12", "bold"),justify=CENTER)
    l2=Label(f8,text="Age ",width=14,height=1, font=("Arial serif",
"12", "bold"),justify=CENTER)
    l3=Label(f8,text='Gender: ',width=14, height=1, font=("Arial
serif", "12", "bold"),justify=CENTER)
    l4=Label(f8,text='Mobile No: ',width=14,height=1, font=("Arial
serif", "12", "bold"),justify=CENTER)
```

```
    import mysql.connector as c
    con=c.connect(host='localhost', user='root',
passwd='abhimysql@13B', database='expense')
    cursor=con.cursor()
    cursor.execute("SELECT username, age, gender, mobile_no
FROM signup limit 0,5")
    i=0
    for signup in cursor:
        for j in range(len(signup)):
```

```

        e = Entry(f8, width=17, fg='blue', font=("Arial serif", "11",
"bold"), justify=CENTER)
        e.grid(row=i+2, column=j)
        e.insert(END, signup[j])
        i=i+1
        b1=Button(f8,text='BACK',width=10,height=2,fg="white",
bg="red",
activebackground="lightyellow",command=menu,font=("Arial serif",
"12", "bold"), justify=CENTER)

```

```

l1.grid(row=1,column=0)
l2.grid(row=1,column=1)
l3.grid(row=1,column=2)
l4.grid(row=1,column=3)
b1.grid(row=10,column=0,pady=15)

```

#-----

```

def stats():
    f9=Frame(bg="#45aaf2",padx=125,pady=60)
    f9.place(width=600, height=400)

    df = pd.read_csv('stats.csv')
    plt.style.use('bmh')

    x = df['category']
    y = df['amount']

    plt.pie(y, labels=x, radius=1.2, autopct='%0.01f%%', shadow =
True)
    plt.show()
    b1=Button(f9,text='BACK',width=10,height=2,fg="white",
bg="red",
activebackground="lightyellow",command=menu,font=("Arial serif",
"12", "bold"), justify=CENTER)
    b1.grid(row=0,column=0)

```



#-----

```
class Notepad:
```

```
    __root = Tk()
```

```
# default window width and height
```

```
    __thisWidth = 300
```

```
    __thisHeight = 300
```

```
    __thisTextArea = Text(__root)
```

```
    __thisMenuBar = Menu(__root)
```

```
    __thisFileMenu = Menu(__thisMenuBar, tearoff=0)
```

```
    __thisEditMenu = Menu(__thisMenuBar, tearoff=0)
```

```
    __thisHelpMenu = Menu(__thisMenuBar, tearoff=0)
```

```
# To add scrollbar
```

```
__thisScrollBar = Scrollbar(__thisTextArea)
```

```
__file = None
```

```
def __init__(self,**kwargs):
```

```
# Set icon
```

```
try:
```

```
self.__root.wm_iconbitmap("Notepad.ico")
```

```
except:
```

```
    pass
```

```
# Set window size (the default is 300x300)
```

```
try:
```

```
self.__thisWidth = kwargs['width']
```

```
except KeyError: pass
```

```
try:
```

```
self.__thisHeight = kwargs['height']
```

```
except KeyError: pass
```

```
# Set the window text
self.__root.title("Untitled - Note")

# Center the window
screenWidth = self.__root.winfo_screenwidth()
screenHeight = self.__root.winfo_screenheight()

# For left-aligning
left = (screenWidth / 2) - (self.__thisWidth / 2)

# For right-align
top = (screenHeight / 2) - (self.__thisHeight / 2)

# For top and bottom
self.__root.geometry('%dx%d+%d+%d' % (self.__thisWidth, left,
top))
self.__thisHeight,

# To make the textarea auto resizable
self.__root.grid_rowconfigure(0, weight=1)
self.__root.grid_columnconfigure(0, weight=1)

# Add controls (widget)
self.__thisTextArea.grid(sticky = N + E + S + W)

# To open new file
self.__thisFileMenu.add_command(label="New",

    command=self.__newFile)

# To open a already existing file
self.__thisFileMenu.add_command(label="Open",

    command=self.__openFile)

# To save current file
```

```
self.__thisFileMenu.add_command(label="Save",  
  
    command=self.__saveFile)  
  
# To create a line in the dialog  
self.__thisFileMenu.add_separator()  
  
self.__thisFileMenu.add_command(label="Exit",  
  
    command=self.__quitApplication)  
self.__thisMenuBar.add_cascade(label="File",  
  
    menu=self.__thisFileMenu)  
  
# To give a feature of cut  
self.__thisEditMenu.add_command(label="Cut",  
  
    command=self.__cut)  
  
# to give a feature of copy  
self.__thisEditMenu.add_command(label="Copy",  
  
    command=self.__copy)  
  
# To give a feature of paste  
self.__thisEditMenu.add_command(label="Paste",  
  
    command=self.__paste)  
  
# To give a feature of editing  
self.__thisMenuBar.add_cascade(label="Edit",  
  
    menu=self.__thisEditMenu)  
  
# To create a feature of description of the notepad  
self.__thisHelpMenu.add_command(label="About Notepad",  
    command=self.__showAbout)
```

```
self.__thisMenuBar.add_cascade(label="Help",  
menu=self.__thisHelpMenu)
```

```
self.__root.config(menu=self.__thisMenuBar)
```

```
self.__thisScrollBar.pack(side=RIGHT,fill=Y)
```

```
# Scrollbar will adjust automatically according to the content
```

```
self.__thisScrollBar.config(command=self.__thisTextArea.yview)
```

```
self.__thisTextArea.config(yscrollcommand=self.__thisScrollBar.set)
```

```
def __quitApplication(self):
```

```
self.__root.destroy()
```

```
# exit()
```

```
def __showAbout(self):
```

```
showinfo("Notepad","Expense Tracker")
```

```
def __openFile(self):
```

```
self.__file = askopenfilename(defaultextension=".txt",
```

```
filetypes=[("All Files","*.*"),
```

```
("Text Documents","*.txt")])
```

```
if self.__file == "":
```

```
# no file to open
```

```
self.__file = None
```

```
else:
```

```
# Try to open the file
```

```
# set the window title
```

```
self.__root.title(os.path.basename(self.__file) + " - Notepad")
```

```
self.__thisTextArea.delete(1.0,END)
```

```
file = open(self.__file,"r")
```

```
self.__thisTextArea.insert(1.0,file.read())
```

```
file.close()
```

```
def __newFile(self):  
self.__root.title("Untitled - Notepad")  
self.__file = None  
self.__thisTextArea.delete(1.0,END)
```

```
def __saveFile(self):
```

```
if self.__file == None:  
# Save as new file  
self.__file = asksaveasfilename(initialfile='Untitled.txt',  
defaulttextextension=".txt",  
filetypes=[("All Files","*.*"),  
("Text Documents","*.txt")])
```

```
if self.__file == "":  
self.__file = None  
else:
```

```
# Try to save the file  
file = open(self.__file,"w")  
file.write(self.__thisTextArea.get(1.0,END))  
file.close()
```

```
# Change the window title  
self.__root.title(os.path.basename(self.__file) + " - Notepad")
```

```
else:  
file = open(self.__file,"w")  
file.write(self.__thisTextArea.get(1.0,END))  
file.close()
```

```
def __cut(self):
self.__thisTextArea.event_generate("<<Cut>>")
```

```
def __copy(self):
self.__thisTextArea.event_generate("<<Copy>>")
```

```
def __paste(self):
self.__thisTextArea.event_generate("<<Paste>>")
```

```
def run(self):
```

```
# Run main application
self.__root.mainloop()
```

```
# Run main application
def notes():
    Notepad(width=600,height=400)
    notes.run()
```

```
#-----
```

```
def home():
    f=Frame(bg="#45aaf2",padx=37,pady=145)
    f.place(width=600, height=400)
    b1=Button(f,text='EXISTING USER',width=16,height=3,
fg="white", bg="blue",
activebackground="lightyellow",bd=0.5,font=("Arial serif", "11",
"bold"), command=login)
    b2=Button(f,text='NEW USER',width=16,height=3, fg="white",
bg="darkgreen", activebackground="lightyellow",bd=0.5,font=("Arial
serif", "11", "bold"), command =signup)
    b3=Button(f,text='EXIT',width=16,height=3, fg="white", bg="red",
activebackground="lightyellow",bd=0.5,font=("Arial serif", "11",
"bold"), command =f.destroy)

    b1.grid(row=1,column=0,padx=10)
    b2.grid(row=1,column=1,padx=20)
```

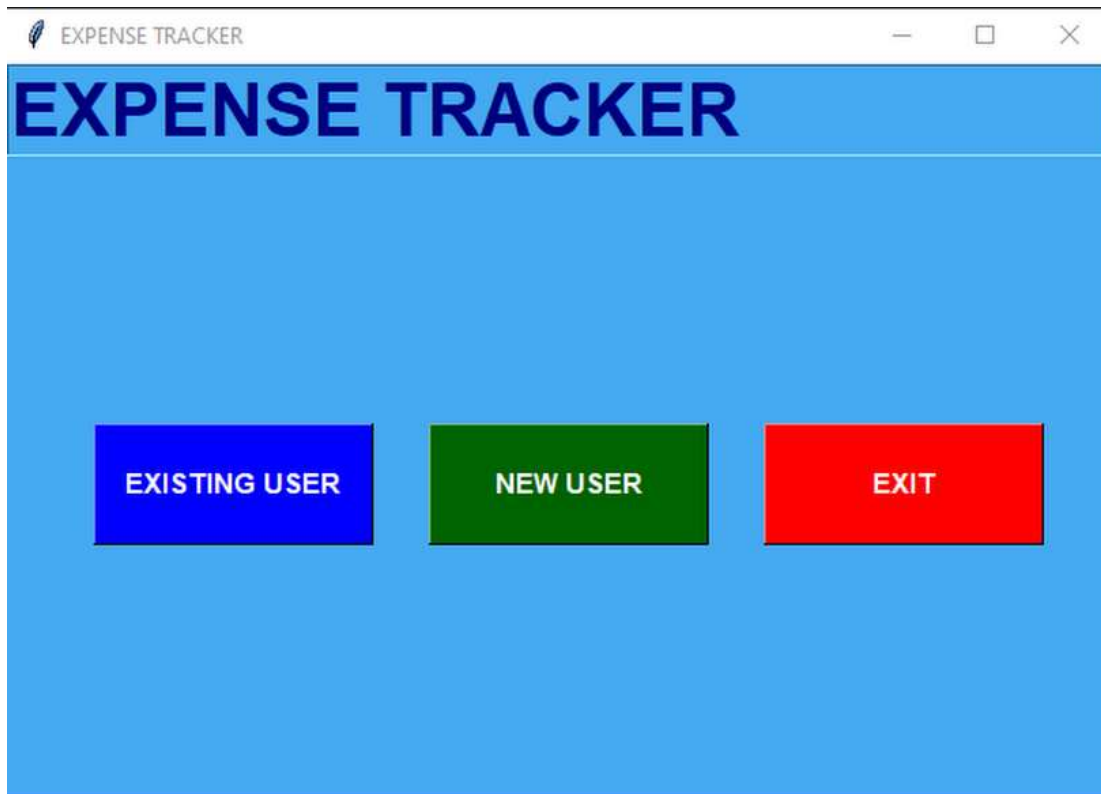
```
b3.grid(row=1,column=2,padx=10)  
f.pack()
```

```
home()  
t.mainloop()
```

### 4.3 Screenshots of the output with description

#### 1) Home:

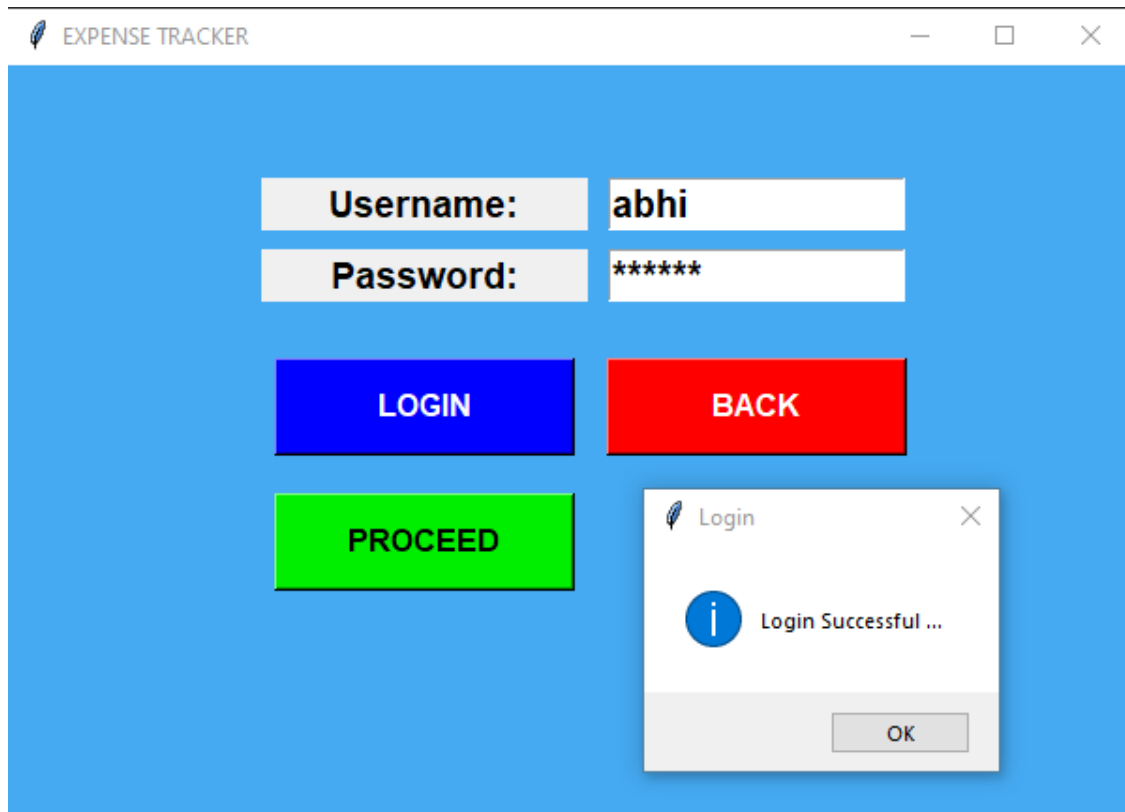
If you are a new user, click on the 'New User' button. It takes you to the sign up page. If you are an 'Existing User', then click on the 'existing user' button. It takes you to the Login Page. If you wish to leave application, click on 'Exit' button.



#### 2) Login:

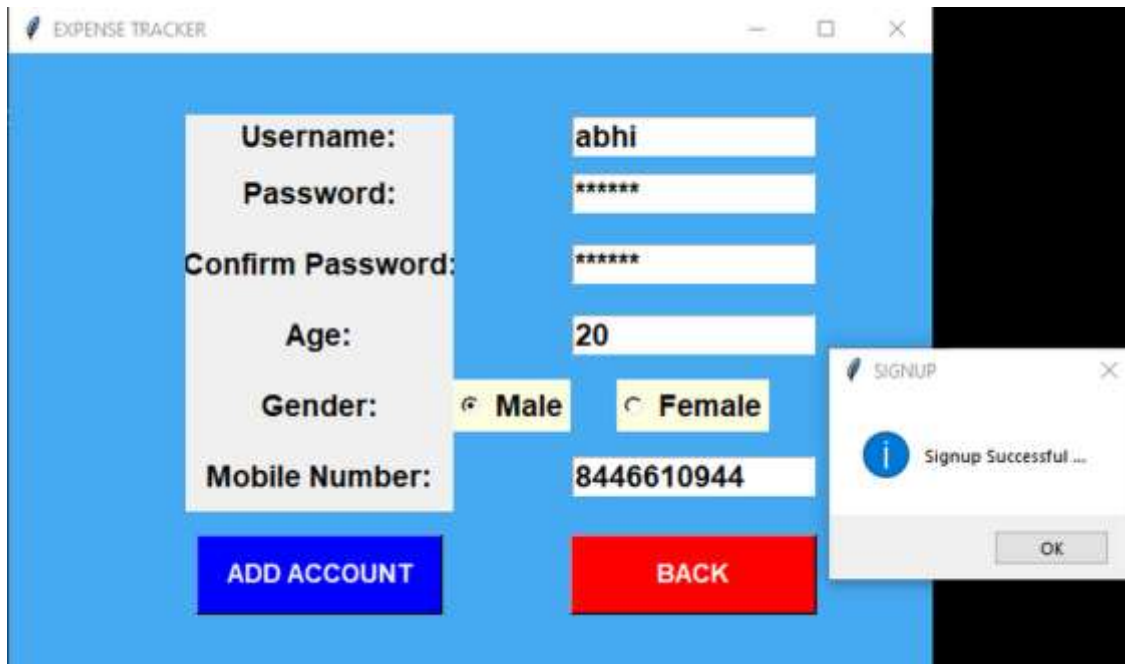
On the login page enter your 'username' and 'password'. After entering all your credentials, click on the 'Login' button. After clicking on the Login button a message is generated of successful login then clicking on the ' Proceed' button . It takes you to 'Menu' page.





### 3) Signup:

On the sign up page enter your 'username', 'password', 'confirm password', 'Age', 'Gender', and 'Mobile Number'. After entering all your credentials, click on the 'Add account' button. After clicking on the Add account button a message pops up which informs about signup.



The image shows a web application window titled "EXPENSE TRACKER". It features a signup form with the following fields and values: Username: "abhi", Password: "\*\*\*\*\*", Confirm Password: "\*\*\*\*\*", Age: "20", Gender: "Male" (selected), and Mobile Number: "8446610944". There are two buttons at the bottom: "ADD ACCOUNT" (blue) and "BACK" (red). A small "SIGNUP" dialog box is open on the right, displaying a message icon and the text "Signup Successful ...", with an "OK" button.

Username: abhi

Password: \*\*\*\*\*

Confirm Password: \*\*\*\*\*

Age: 20

Gender: ☒ Male ☐ Female

Mobile Number: 8446610944

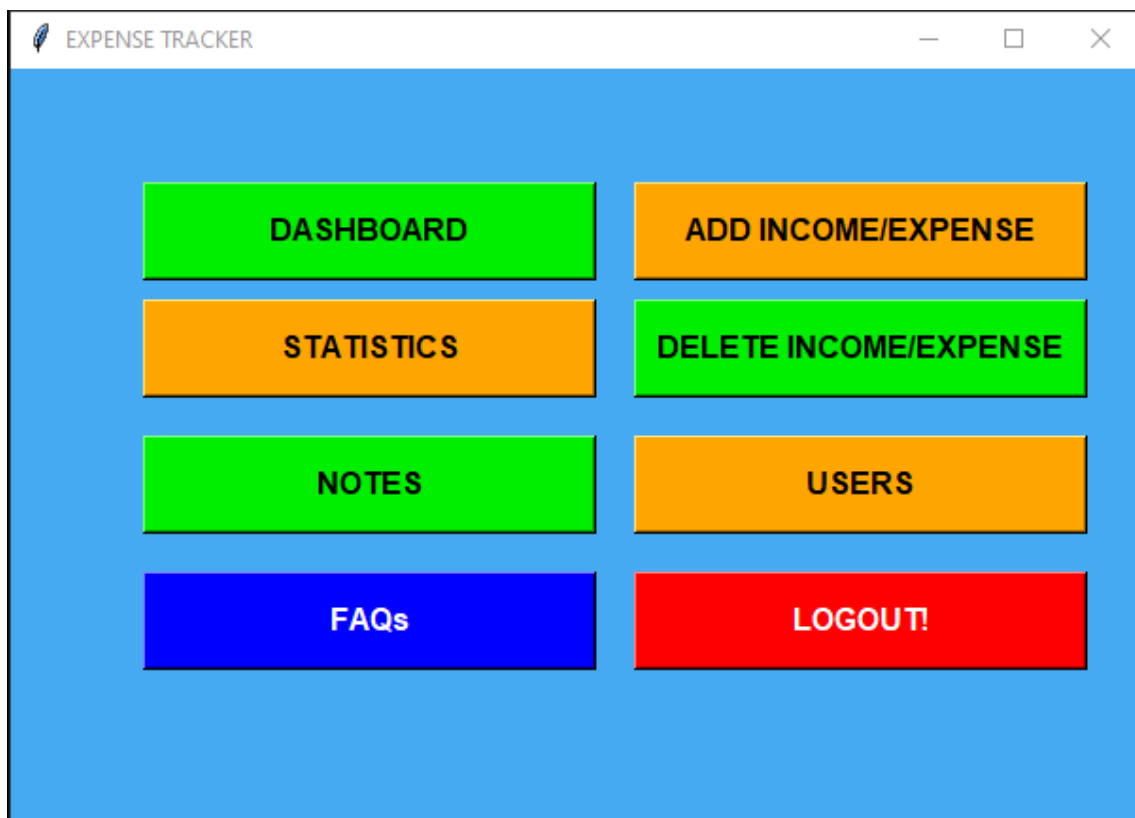
ADD ACCOUNT BACK

SIGNUP

Signup Successful ...

OK

4) **Menu:** Contains 8 different options to choose from as stated below



The image shows a web application window titled "EXPENSE TRACKER". It displays a menu with eight options arranged in two columns. The options are: DASHBOARD (green), ADD INCOME/EXPENSE (orange), STATISTICS (orange), DELETE INCOME/EXPENSE (green), NOTES (green), USERS (orange), FAQs (blue), and LOGOUT! (red).

DASHBOARD

ADD INCOME/EXPENSE

STATISTICS

DELETE INCOME/EXPENSE

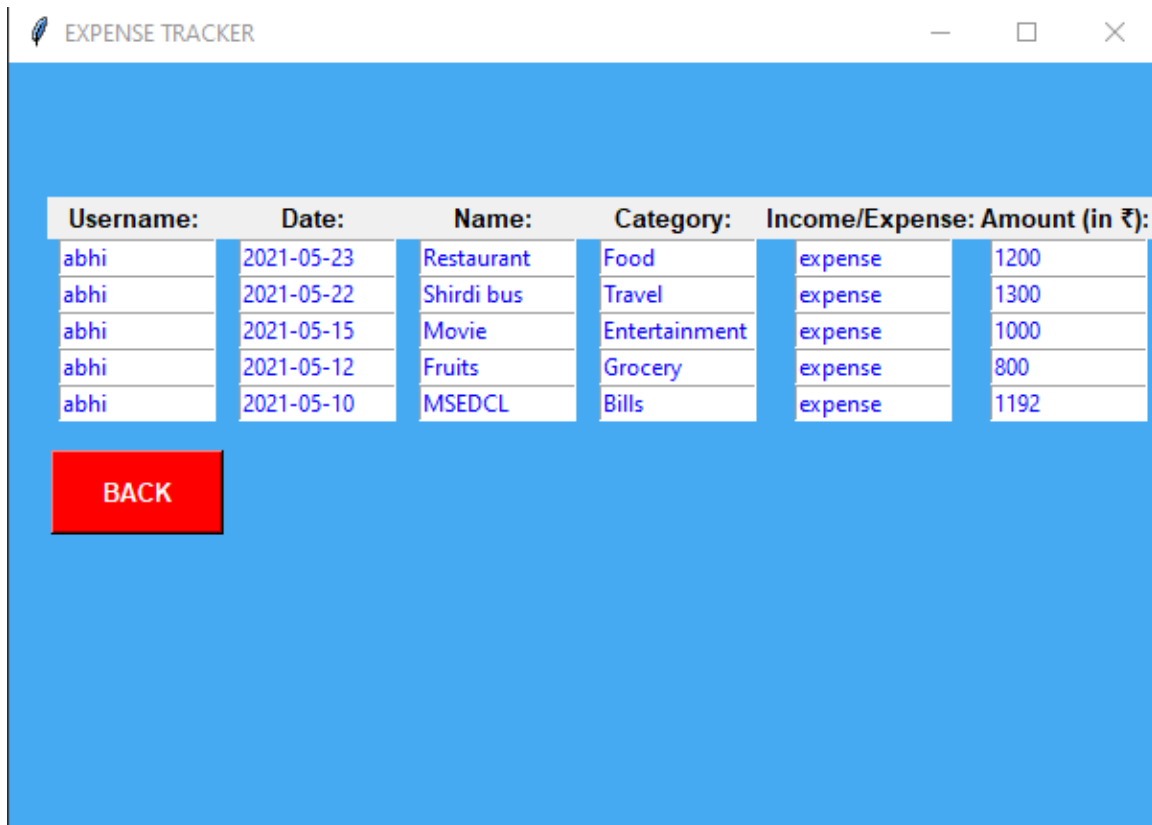
NOTES

USERS

FAQs

LOGOUT!

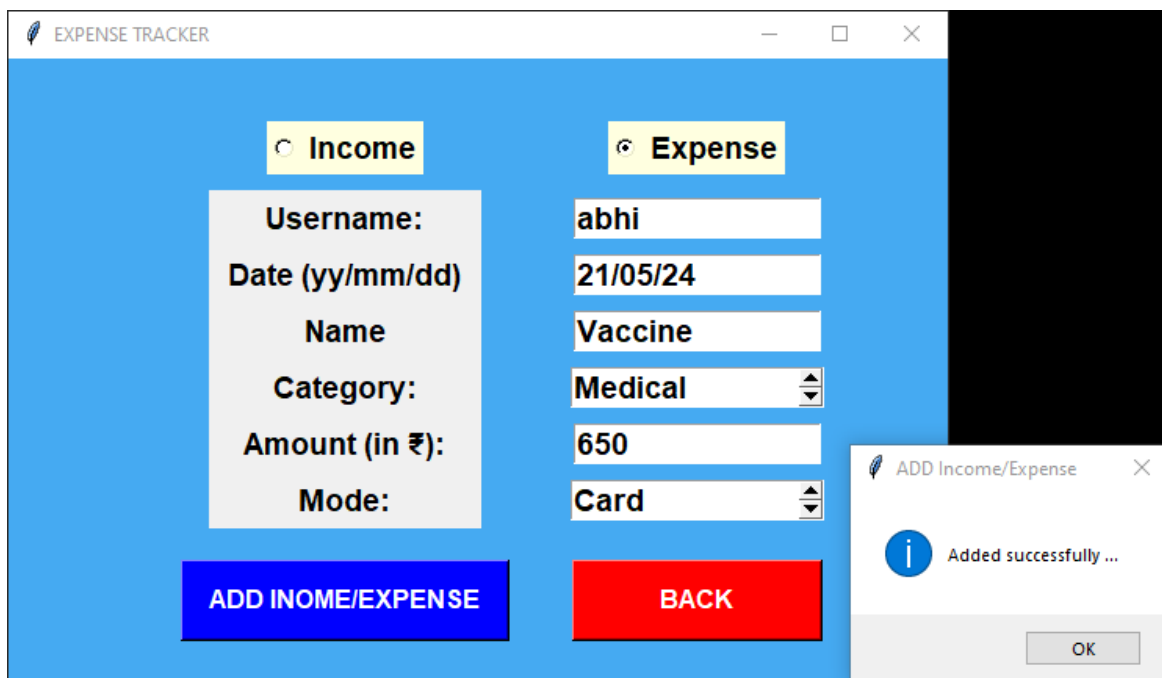
**5) Dashboard:** It shows information of 5 latest income/expense inputs



The dashboard window titled "EXPENSE TRACKER" displays a table of the 5 latest income/expense inputs. The table has columns for Username, Date, Name, Category, Income/Expense, and Amount (in ₹). Below the table is a red "BACK" button.

Username:	Date:	Name:	Category:	Income/Expense:	Amount (in ₹):
abhi	2021-05-23	Restaurant	Food	expense	1200
abhi	2021-05-22	Shirdi bus	Travel	expense	1300
abhi	2021-05-15	Movie	Entertainment	expense	1000
abhi	2021-05-12	Fruits	Grocery	expense	800
abhi	2021-05-10	MSEDCL	Bills	expense	1192

**6) Add Income/Expense:** To add an income/expense to account



The "Add Income/Expense" form window is shown. It has two tabs: "Income" and "Expense". The "Expense" tab is selected. The form contains fields for Username, Date (yy/mm/dd), Name, Category (dropdown), Amount (in ₹), and Mode (dropdown). Below the form are two buttons: "ADD INCOME/EXPENSE" and "BACK". A small notification window titled "ADD Income/Expense" is open, showing a message "Added successfully ..." and an "OK" button.

**Income** **Expense**

Username: abhi

Date (yy/mm/dd): 21/05/24

Name: Vaccine

Category: Medical

Amount (in ₹): 650

Mode: Card

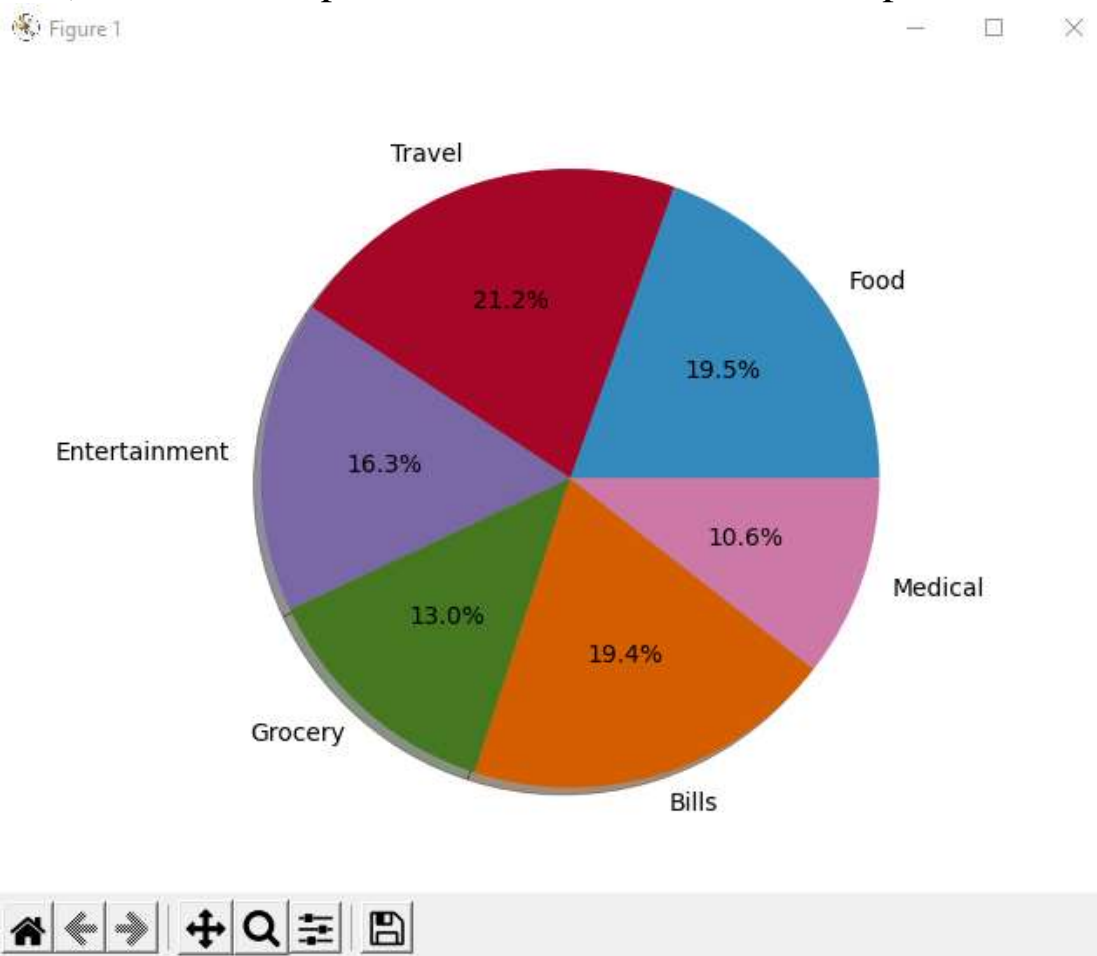
**ADD INCOME/EXPENSE** **BACK**

**ADD Income/Expense** X


i Added successfully ...

OK

7) **Statistics:** It provides statistics in the form of pie-chart



8) **Remove Income/Expense:** To delete an income/expense from account

 EXPENSE TRACKER


dt

21/05/12

REMOVE INOME/EXPENSE

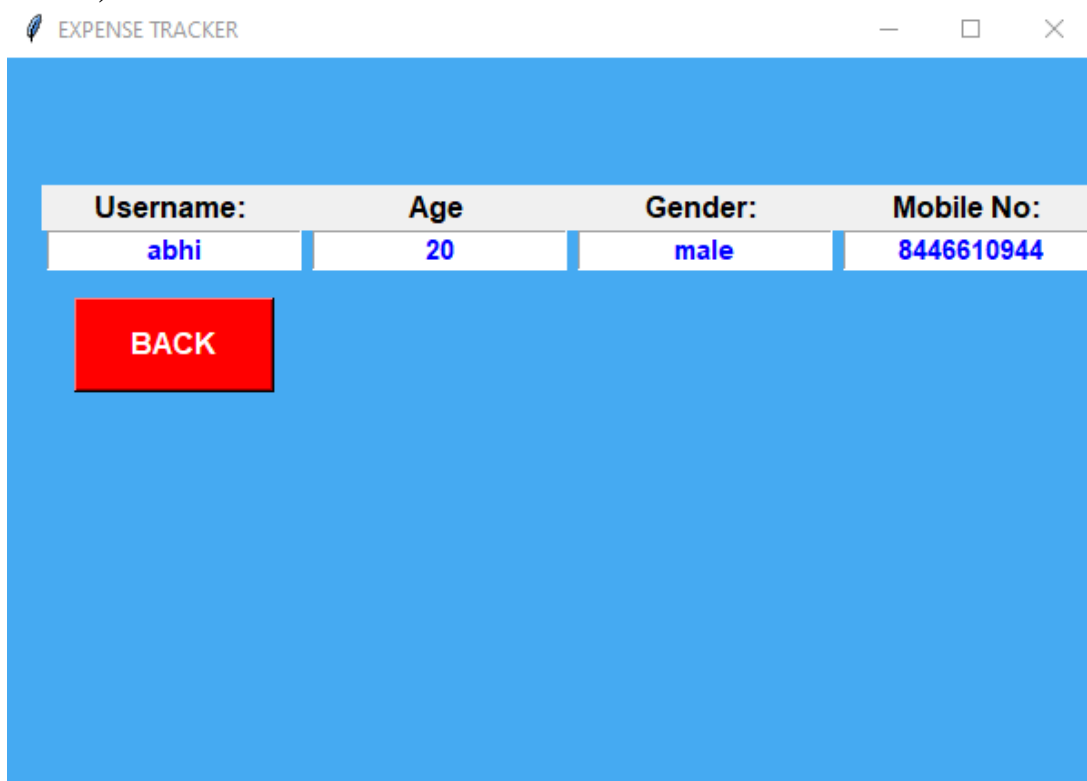
BACK

### 9) Notes: To create short notes on transactions performed

 Untitled - Note

File Edit Help

10) **Users:** To view various users

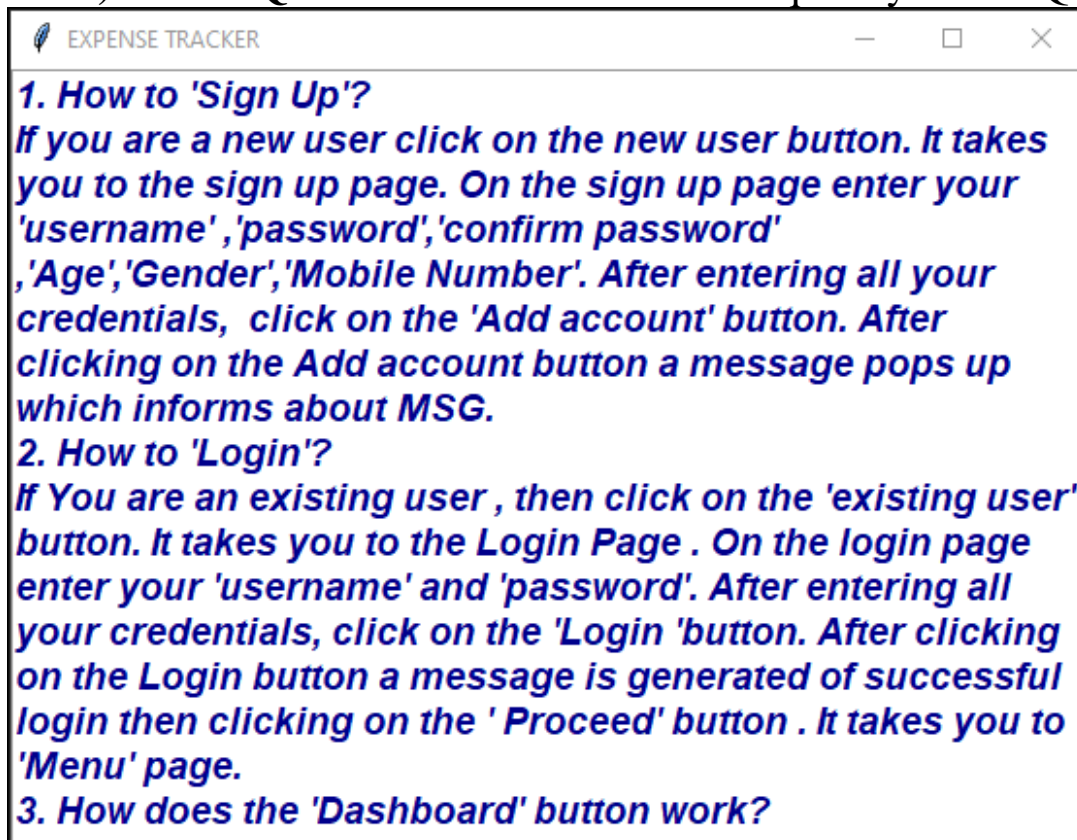


The image shows a screenshot of a web application titled "EXPENSE TRACKER". The main content area has a light blue background. At the top, there is a table with four columns: "Username:", "Age", "Gender:", and "Mobile No:". The table contains one row of data: "abhi", "20", "male", and "8446610944". Below the table, there is a red button with the text "BACK" in white capital letters.

Username:	Age	Gender:	Mobile No:
abhi	20	male	8446610944

BACK

11) **FAQs:** It consists some of the Frequently Asked Questions



The image shows a screenshot of a web application titled "EXPENSE TRACKER". The main content area has a light blue background. It displays a list of frequently asked questions (FAQs) in blue text. The questions are numbered 1, 2, and 3. The first question is "1. How to 'Sign Up'?" followed by a paragraph of instructions. The second question is "2. How to 'Login'?" followed by a paragraph of instructions. The third question is "3. How does the 'Dashboard' button work?".

1. **How to 'Sign Up'?**  
*If you are a new user click on the new user button. It takes you to the sign up page. On the sign up page enter your 'username' , 'password', 'confirm password' , 'Age', 'Gender', 'Mobile Number'. After entering all your credentials, click on the 'Add account' button. After clicking on the Add account button a message pops up which informs about MSG.*

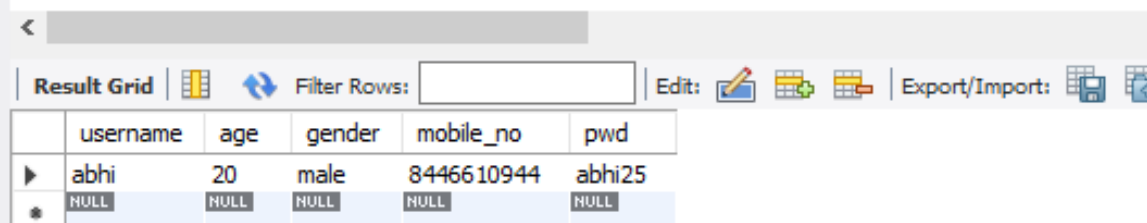
2. **How to 'Login'?**  
*If You are an existing user , then click on the 'existing user' button. It takes you to the Login Page . On the login page enter your 'username' and 'password'. After entering all your credentials, click on the 'Login 'button. After clicking on the Login button a message is generated of successful login then clicking on the ' Proceed' button . It takes you to 'Menu' page.*

3. **How does the 'Dashboard' button work?**

12) **Logout:** To logout of your account

13) **Signup Table:**

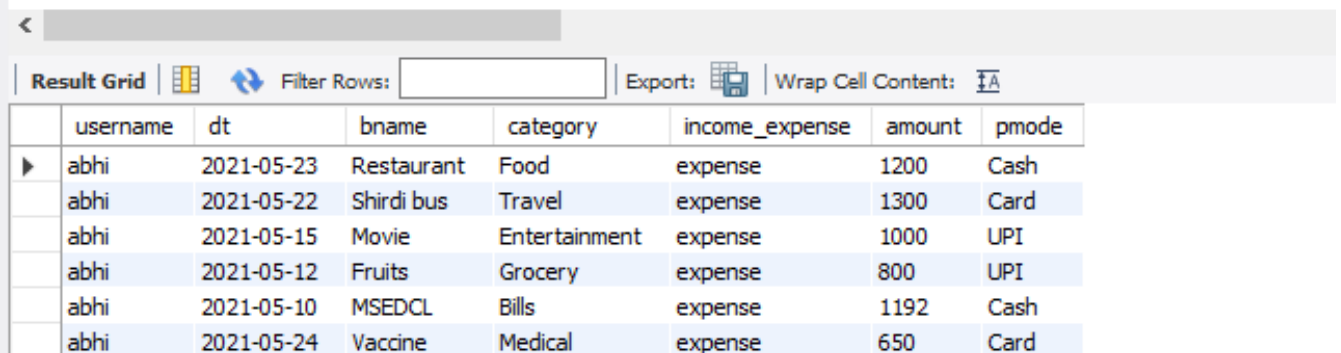
```
1 • SELECT * FROM signup;
```



	username	age	gender	mobile_no	pwd
▶	abhi	20	male	8446610944	abhi25
*	NULL	NULL	NULL	NULL	NULL

14) **Income/Expense Table:**

```
1 • SELECT * FROM inex;
```



	username	dt	bname	category	income_expense	amount	pmode
▶	abhi	2021-05-23	Restaurant	Food	expense	1200	Cash
	abhi	2021-05-22	Shirdi bus	Travel	expense	1300	Card
	abhi	2021-05-15	Movie	Entertainment	expense	1000	UPI
	abhi	2021-05-12	Fruits	Grocery	expense	800	UPI
	abhi	2021-05-10	MSEDCL	Bills	expense	1192	Cash
	abhi	2021-05-24	Vaccine	Medical	expense	650	Card

## **Chapter 5**

### **Conclusion**

Users will experience a smooth and stable application with new and efficient features that will enrich interaction between user and the application. The system is designed to be more user friendly without compromising on its features. Thus, tracking expenses has become much easier with this application.



## References:

[1] S. Chandini, T. Poojitha, D. Ranjith, V. J. Mohammed Akram, M. S. Vani, V. Rajyalakshmi, “Online Income and Expense Tracker”, *International Research Journal of Engineering and Technology (IRJET)* / Volume: 06, Issue: 03, March 2019.

[2] Md. Abdul Karim ,Taslina Yesmin Orin,Tasnim Ahmed,Ahmed Al Marouf, “Daily Expense Tracker: An Android Based Mobile Application”, *DAFFODIL INTERNATIONAL UNIVERSITY DHAKA, BANGLADESH* / September 2019