CS631 Data Management Systems Design

WALLET E- PAYMENT SYSTEM

Team 5

Megha Manoj (mm2773@njit.edu)

Tejaswini Rao Akula (ta58@njit.edu)

Vishnupriya Santhosh (vs263@njit.edu)

Department of Data Science,

Ying Wu College of Computing,

New Jersey Institute of Technology

Newark, New Jersey, 07102, United States

Under the guidance

of

Theodoratos Dimitrios, Associate Professor, Computer Science (dth@njit.edu)

TABLE OF CONTENTS

1. Introduction

Traditional brick-and-mortar banking offers customers a personal dimension. Customers visit the bank, converse with the financial advisors or representatives who can help them manage their accounts, and get answers to their queries. While this can be more reassuring to many customers, online banking has its own benefits that traditional banks cannot provide. These drawbacks are addressed and implemented via the employed E-Payment Systems.

The Wallet E-Payment System is a standard electronic or online payment system that meets users' expectations for convenience in payment transactions from anywhere and at any time. Online payments are likely to be vulnerable to fraudulent activities and technical disturbances. It can be challenging to provide optimal infrastructure and services for processing these payments. The primary objective of deploying this system is to provide users and service providers with a safe and convenient platform to make payments online, ensuring financial security. This way, time and cost-effectiveness are also guaranteed, which in turn activates more users to utilize it. The collaboration with existing payment systems like service providers and bank applications (which serve as a source of funds) allows for a seamless experience for users.

2. Goals of the Project

2.1 Phase I

Create an Enhanced Entity-Relationship Model representing the Entities, the attributes and its various types. Highlight the structural constraints, participation and whatever notations required post skimming the description of the Project we are working on. Moreover, enlist the considerations we adopted to construct the ER diagram.

2.2 Phase II

Create the logical design of the WALLET Payment Network by mapping the EER diagram into the relational schema. Analyze the ER diagram and indicate the entities, their relationships and attributes.Identify the Primary Keys and Foreign Keys on the Relational Database schema. Determine the constraints for the relational model.

2.3 Phase III

Work on deploying the Application Design of the Wallet Payment Network that collaborates the end products of Phase I and Phase II. Creation and Population of the Database Relations is to be ensured, provided that no error or violations of constraints or properties is encountered. Integration of SQL Querying with the Application Framework, displaying the development of instances for the functionalities and menus of the Wallet payment network.

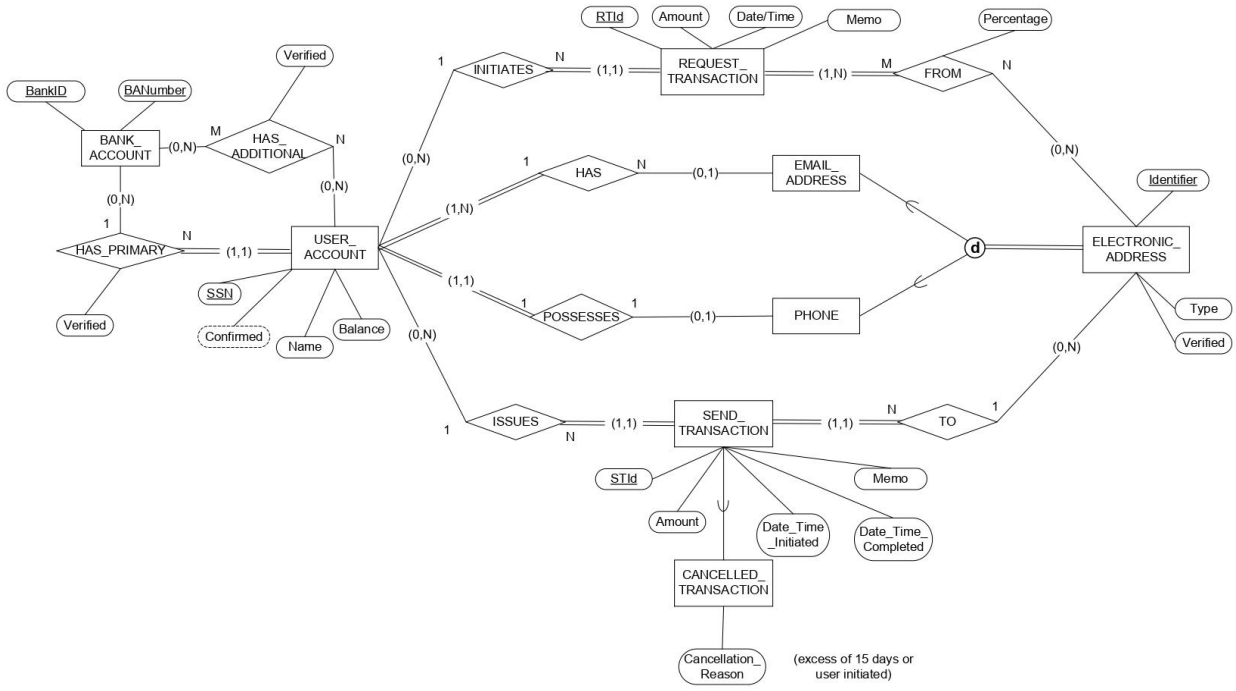
3. Implementation

Create a Wallet database to store User information and to track their transaction in a Wallet application system, ensuring user convenience without any mobility. Wallet Database is cosidered to be a payment network comprising of User information, user accounts, bank account information and transactions and statements. Each customer using the Wallet application is uniquely identified by their SSN. Each user with the Bank account linked to the Wallet application is recognized by the BankID and Bank Account Number. The application supports two types of actions, i.e., Send Transaction and Request Money Transaction each of which are identified uniquely by their Send Transaction ID and Request Transaction ID respectively. To expand on the basis of how and through which data the transactions are being processed, is by using their Identifer fields which can be either their phone number or email address. Extra information is stored in a additional information to ensure the alternative way to address actions if the primary information fails.

3.1 Enhanced Entity Relationship Design

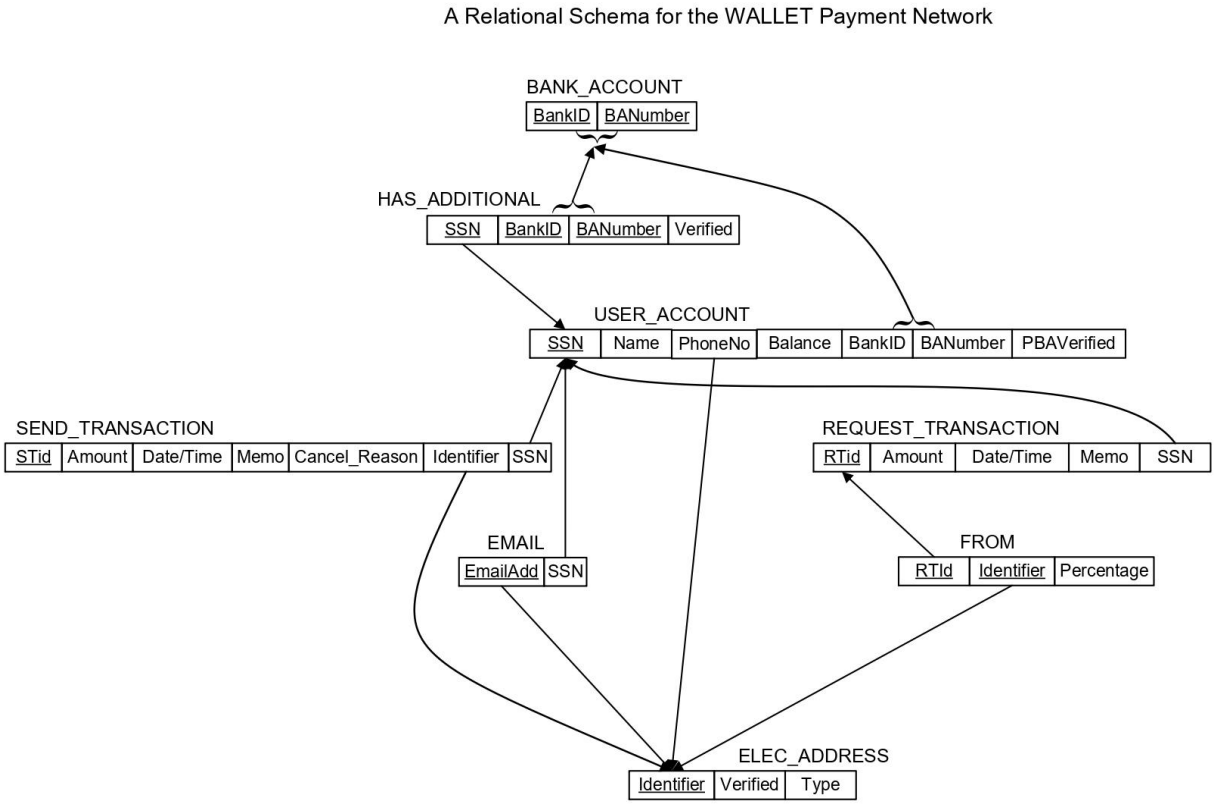
Started of with designing a **EER diagram** with thorough understanding of Wallet Project description and then building the **Relational Schema (Logical Design)** via Relational mapping techniques. Moreover we used above Designs to build a real-time usage application to carry out various functionalities and highlight the features.

An ER Diagram for the WALLET Payment Network



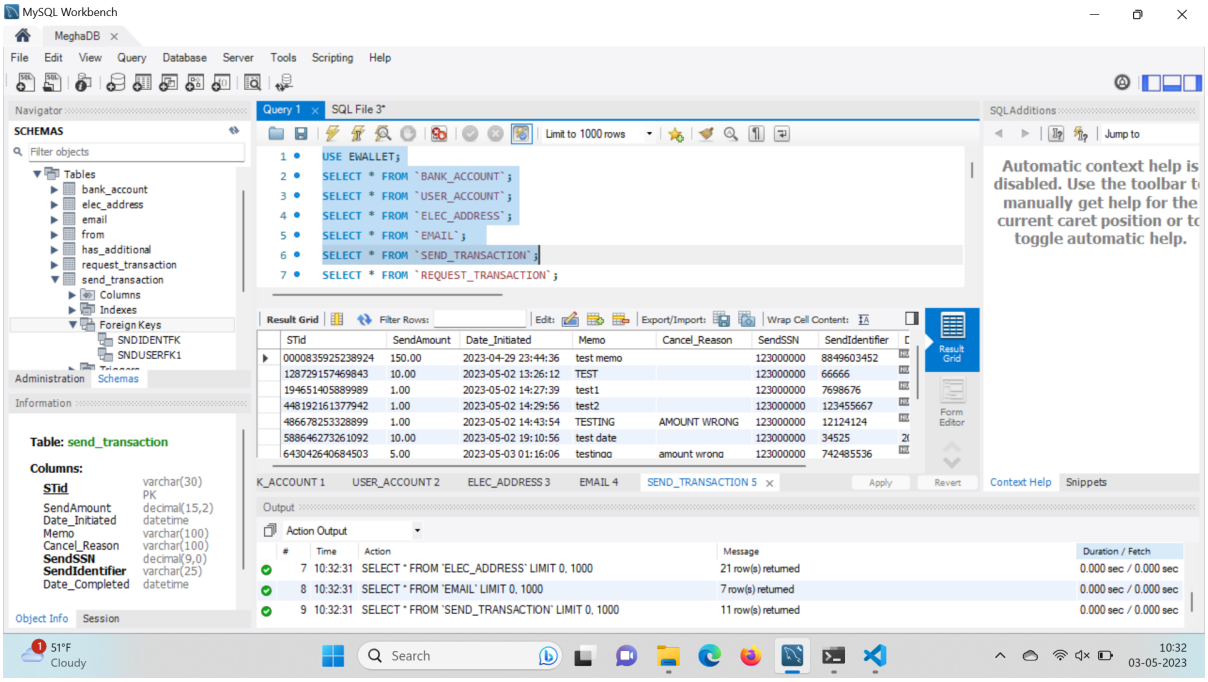
3.2 Relational Schema (Logical Database Design)

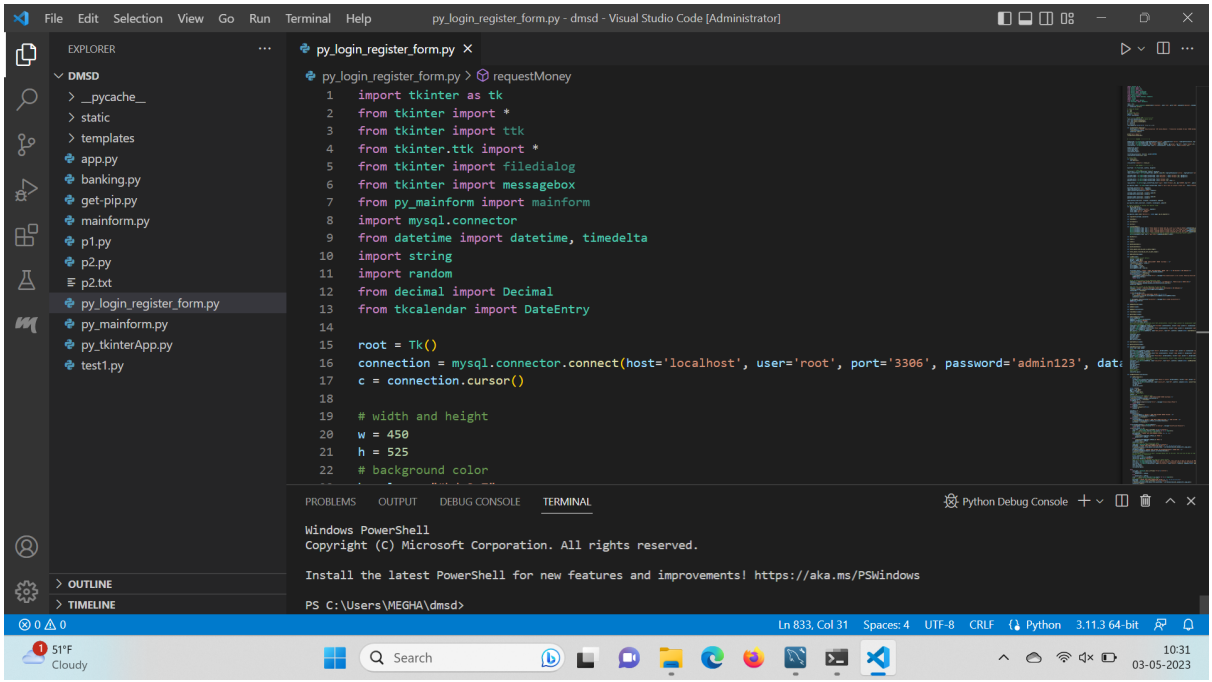
A Relational Schema depicting the referential integrity constraints, displaying the primary and foreign keys. Separate Relations are created for Subclasses EMAIL_ADDRESS and PHONE for the Superclass ELECTRONIC_ADDRESS. Relations for the relationships FROM and HAS_ADDITIONAL are created.

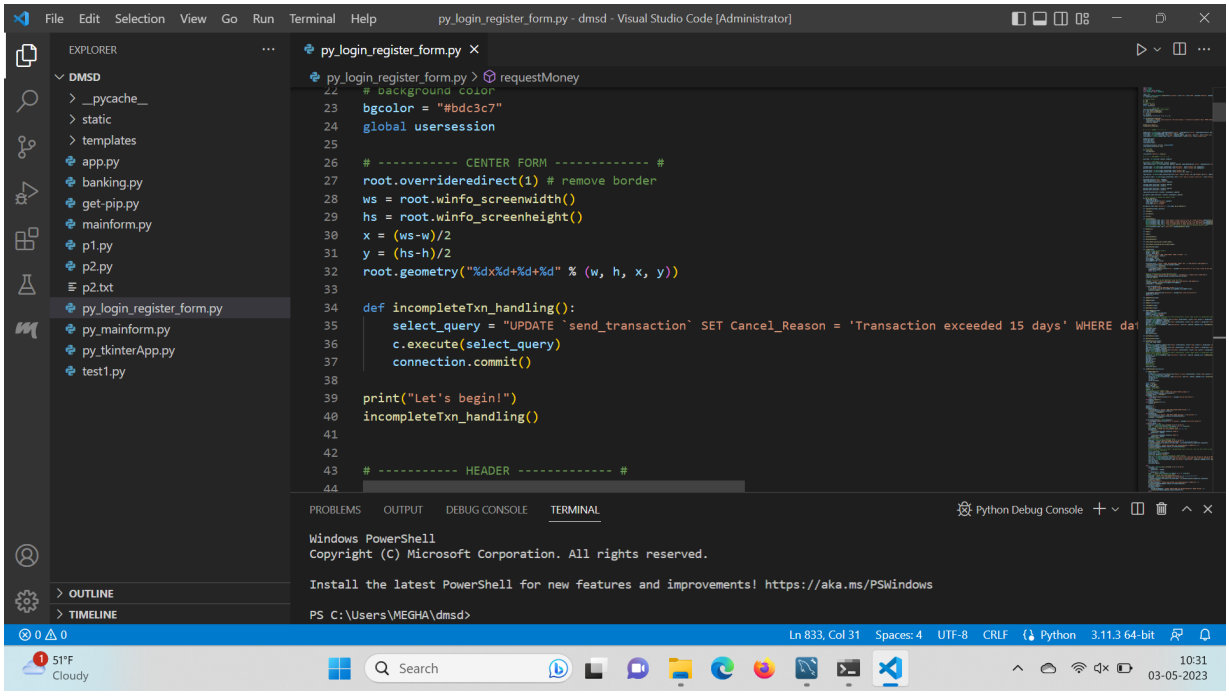


3.3 Application Design: Functionalities

We used **MYSQL** to run the SQL commands that **create, insert and populate the relations** required to store data in the “Wallet” database. Connection to the database is practiced in **VisualStudio Code** using **Python** programming while depicting the application **front view** using the **Tkinter framework** displaying each functionality on a separate frame or window.

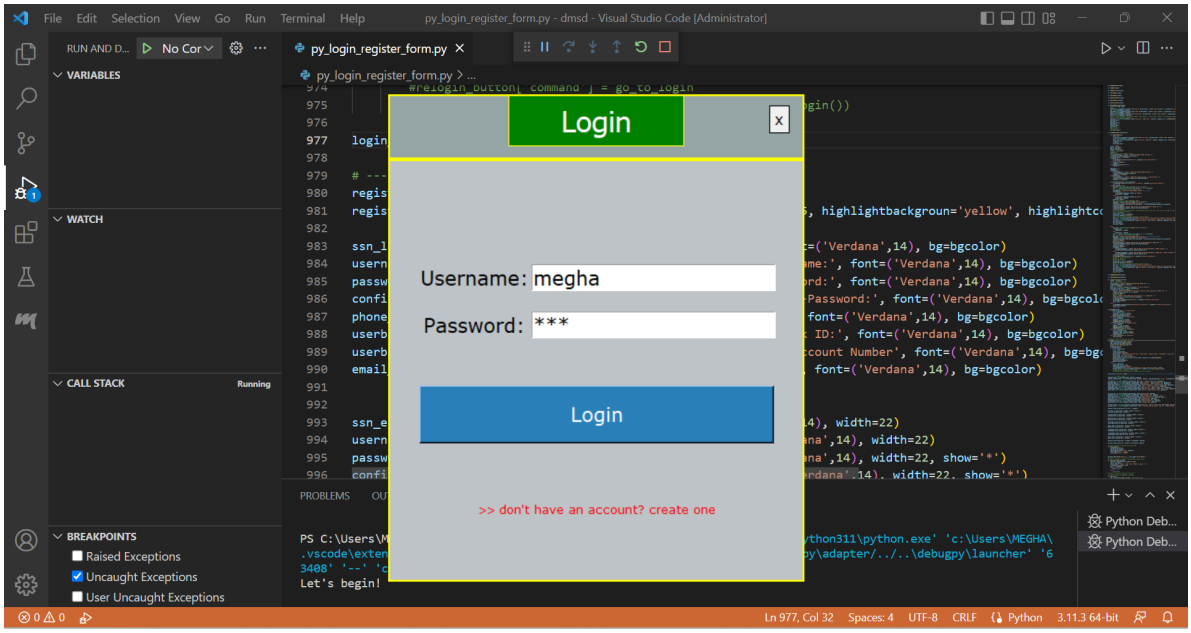


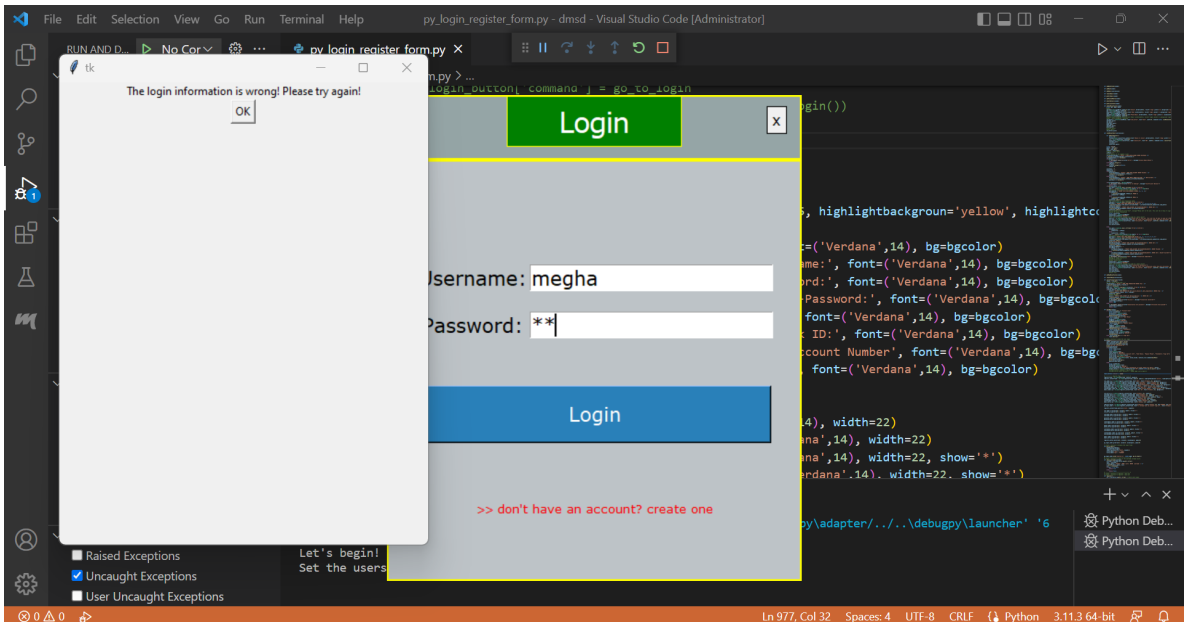

File Edit Selection View Go Run Terminal Help
py_login_register_form.py - dmsd - Visual Studio Code [Administrator]
EXPLORER
DMSD
__pycache__
static
templates
app.py
banking.py
get-pip.py
mainform.py
p1.py
p2.py
p2.txt
py_login_register_form.py
py_mainform.py
py_tkinterApp.py
test1.py
OUTLINE
TIMELINE
py_login_register_form.py
py_login_register_form.py > requestMoney
1 import tkinter as tk
2 from tkinter import *
3 from tkinter import ttk
4 from tkinter.ttk import *
5 from tkinter import filedialog
6 from tkinter import messagebox
7 from py_mainform import mainform
8 import mysql.connector
9 from datetime import datetime, timedelta
10 import string
11 import random
12 from decimal import Decimal
13 from tkcalendar import DateEntry
14
15 root = Tk()
16 connection = mysql.connector.connect(host='localhost', user='root', port='3306', password='admin123', dat
17 c = connection.cursor()
18
19 # width and height
20 w = 450
21 h = 525
22 # background color
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\MEGHA\dmsd>
Ln 833, Col 31 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
51°F Cloudy
Search
10:31 03-05-2023


File Edit Selection View Go Run Terminal Help
py_login_register_form.py - dmsd - Visual Studio Code [Administrator]
EXPLORER
DMSD
__pycache__
static
templates
app.py
banking.py
get-pip.py
mainform.py
p1.py
p2.py
p2.txt
py_login_register_form.py
py_mainform.py
py_tkinterApp.py
test1.py
OUTLINE
TIMELINE
py_login_register_form.py
py_login_register_form.py > requestMoney
# background color
bgcolor = "#bdc3c7"
global usersession
---------- CENTER FORM ------------
root.overrideredirect(1) # remove border
ws = root.winfo_screenwidth()
hs = root.winfo_screenheight()
x = (ws-w)/2
y = (hs-h)/2
root.geometry("%dx%d+%d+%d" % (w, h, x, y))
def incompleteTxn_handling():
select_query = "UPDATE `send_transaction` SET Cancel_Reason = 'Transaction exceeded 15 days' WHERE da
c.execute(select_query)
connection.commit()
print("Let's begin!")
incompleteTxn_handling()
# ---------- HEADER ------------ #
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\MEGHA\dmsd>
Ln 833, Col 31 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
51°F Cloudy
Search
10:31 03-05-2023

3.3.1

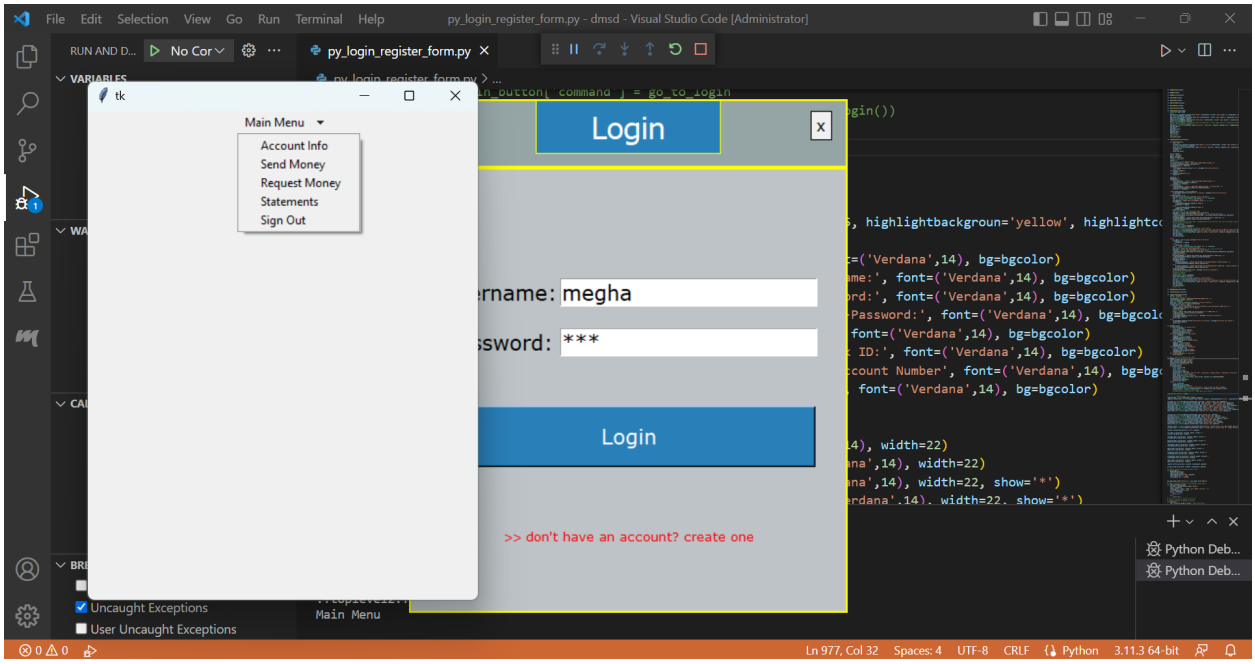
A Beginner **Login - Registration Window** is ensured to activate new customers to register using the Wallet Application and current customers to perform necessary actions through Login.

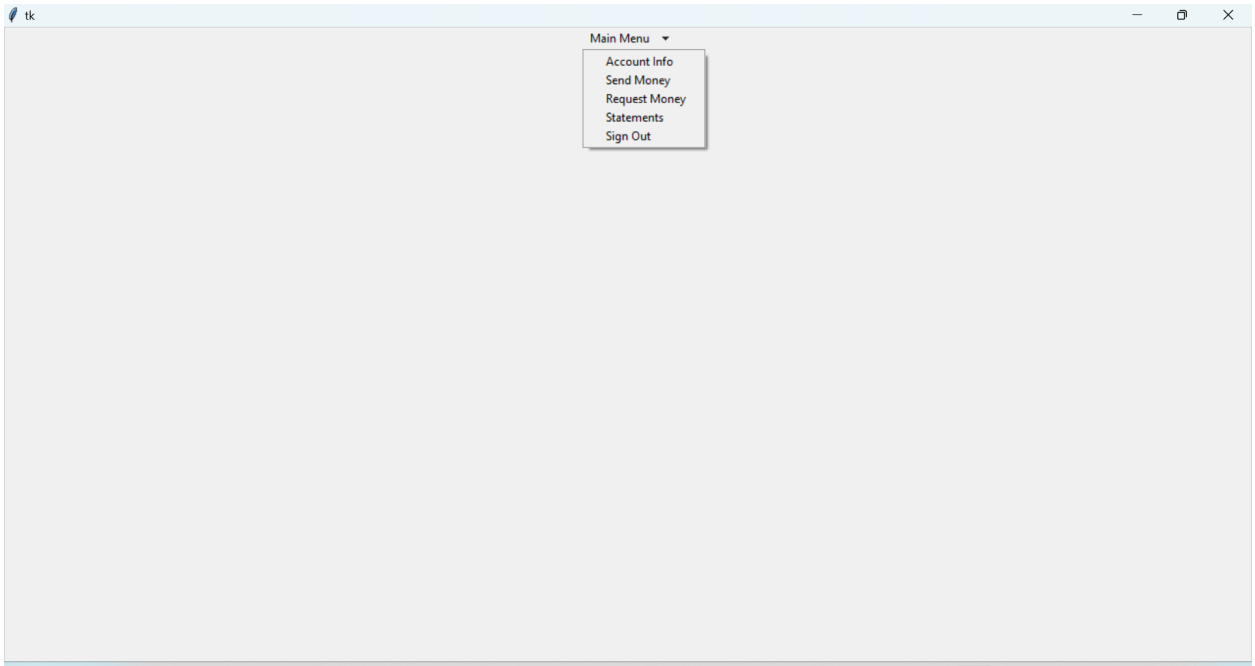




3.3.2

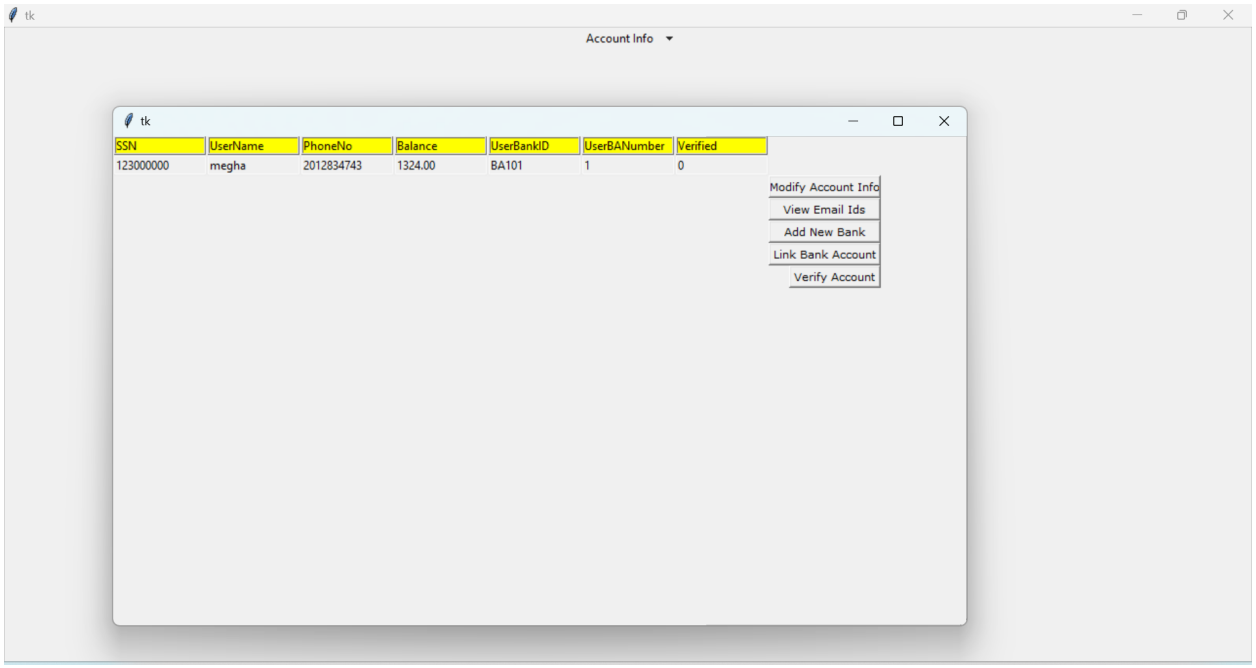
The Wallet User is enabled to the plethora of basic and related options as part of the **Main Menu**, where he or she can look into their profile (MyWallet Account), Actions to perform (SEND to and REQUEST from), (MyWallet Statements), Search Transactions. If the user has reached task completion, he or she can logout via Sign out option.

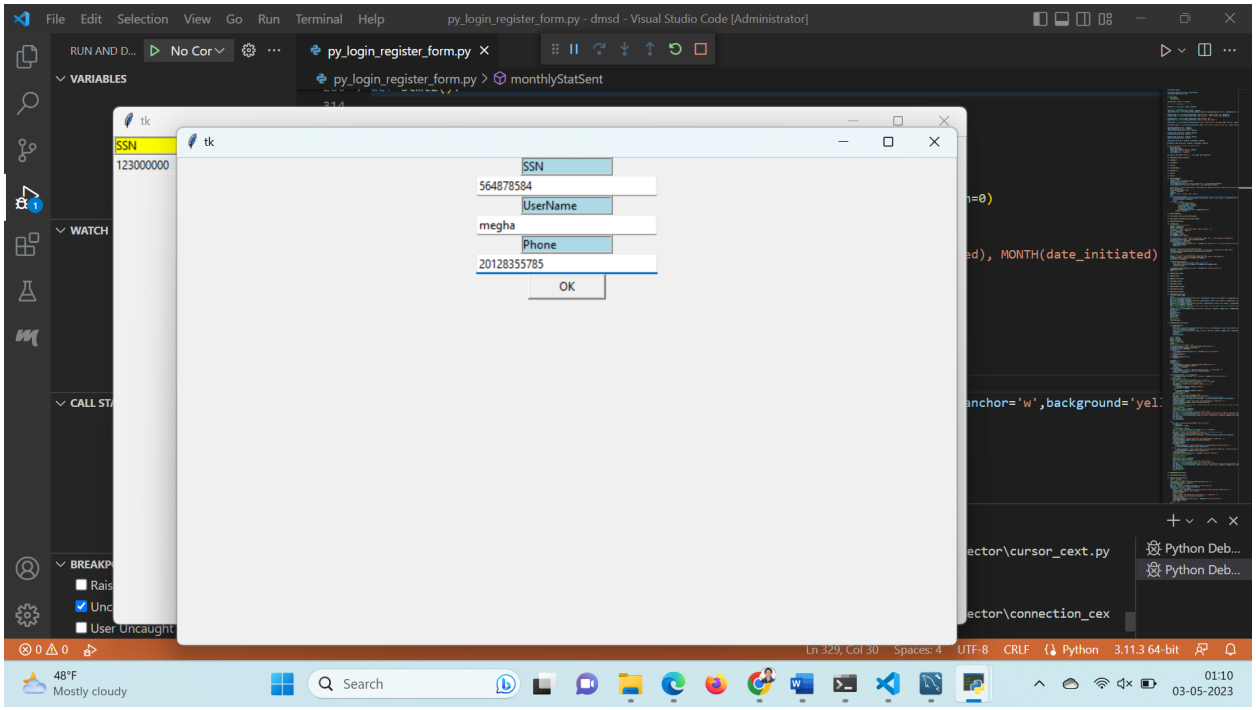


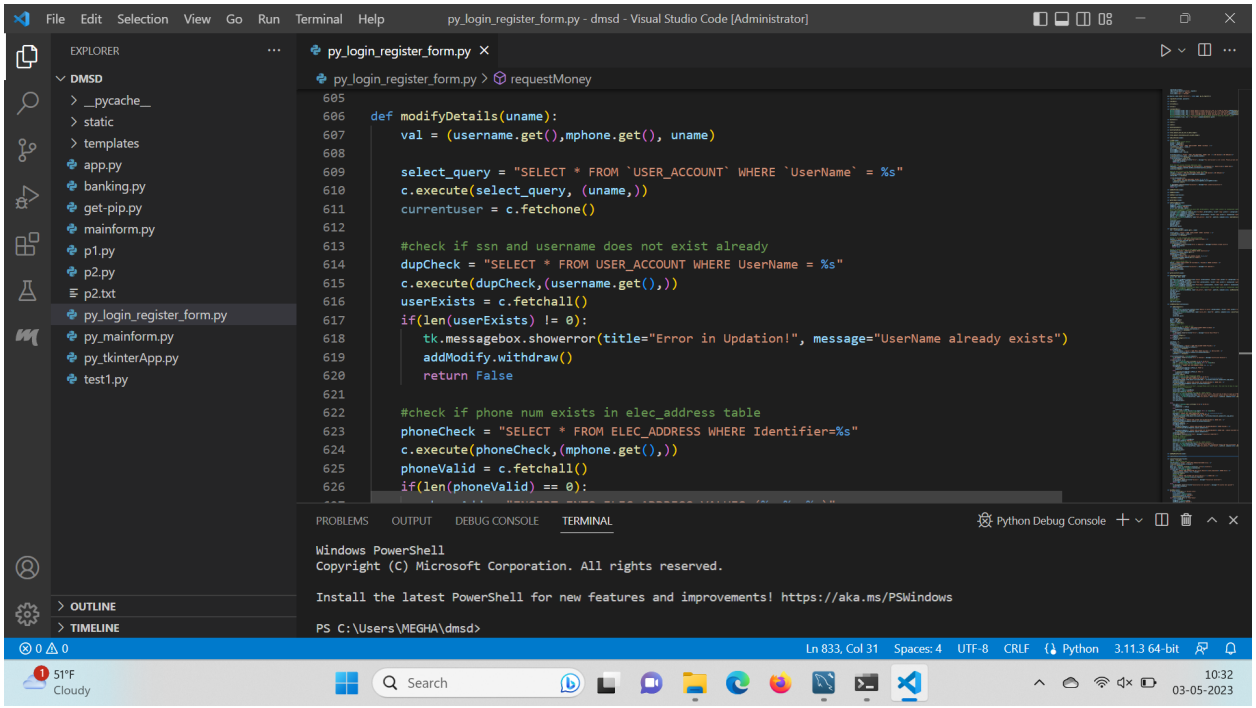


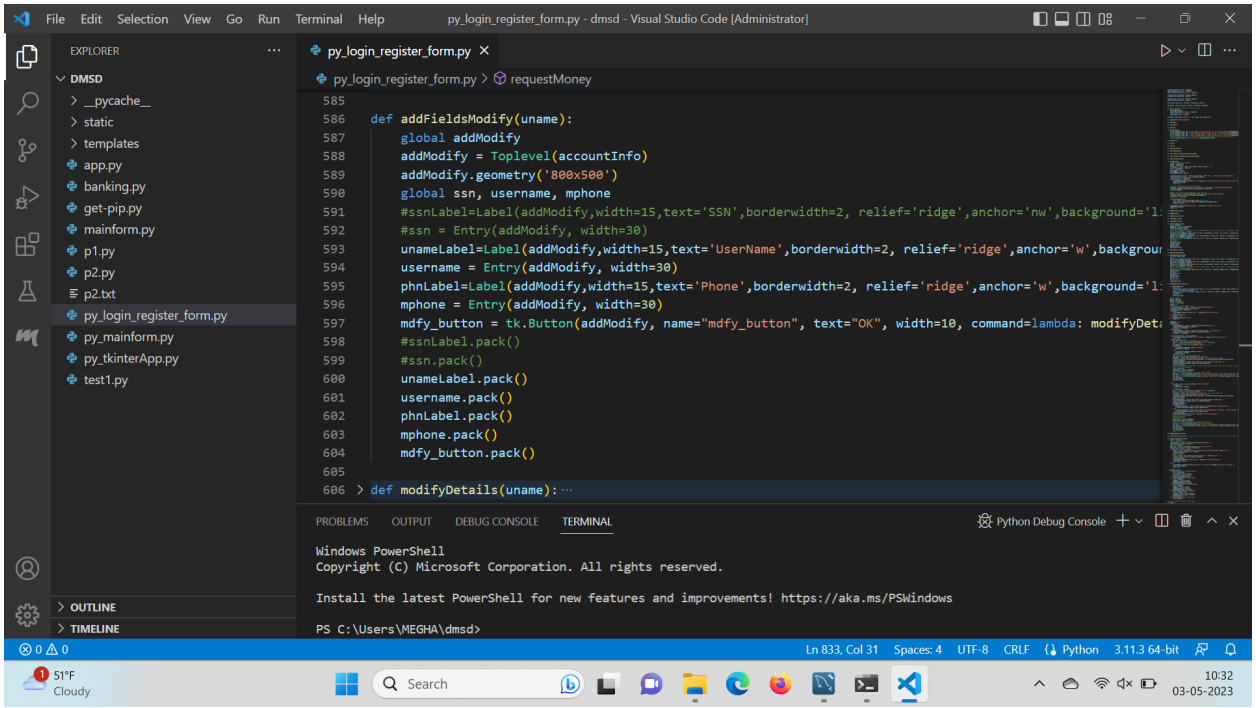
3.3.3

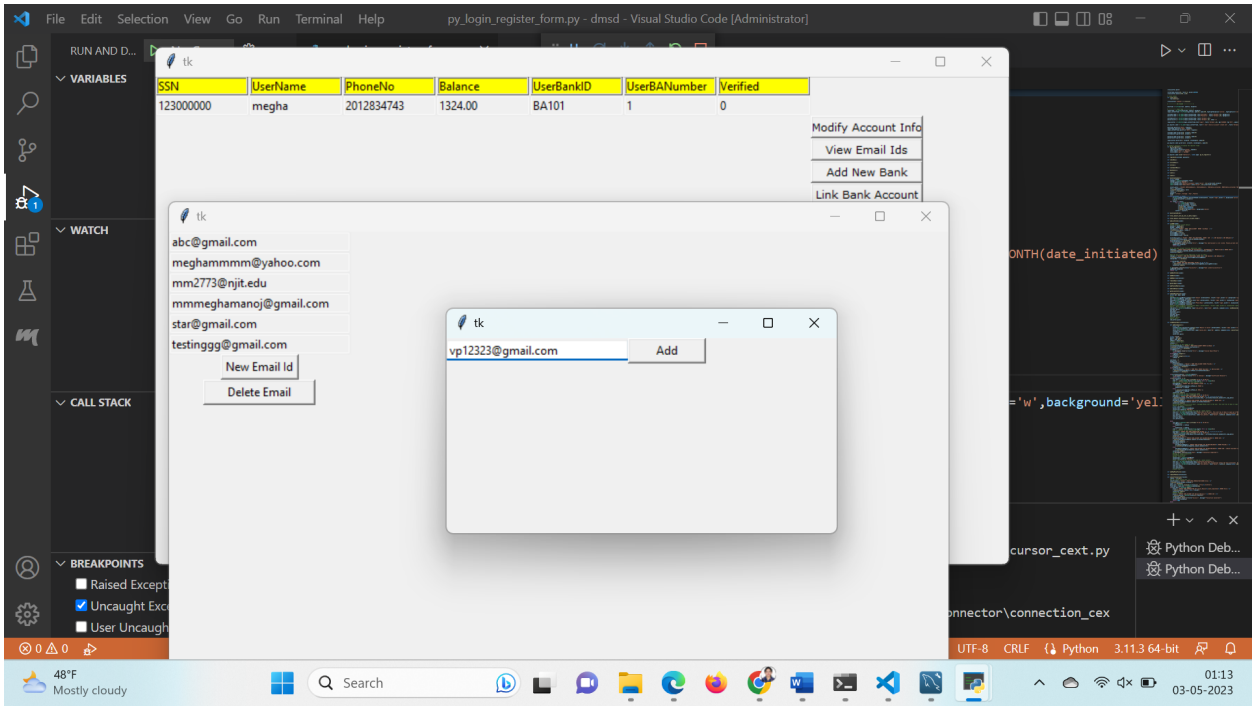
MyWallet Account is expanded to various functionalities where the user can view and modify their account information, their mode of transaction via phone number or email address. Moreover, the user can add or remove bank accounts, keeping an account as primary or default.

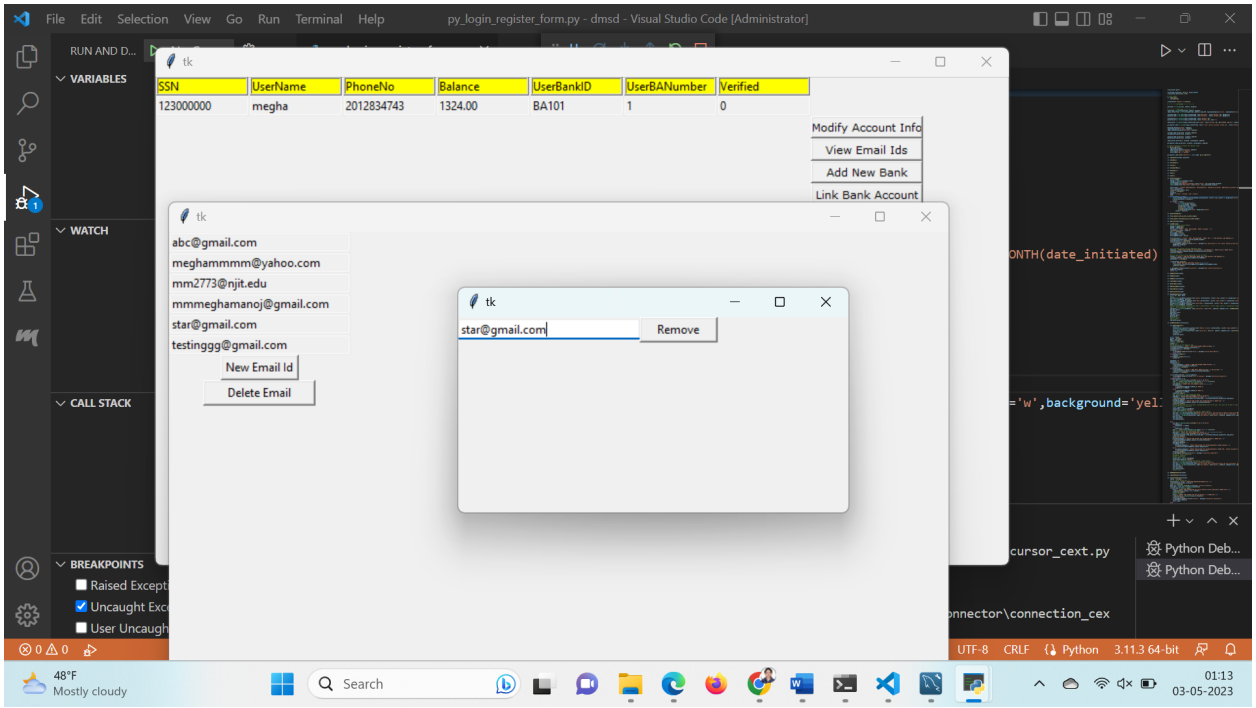


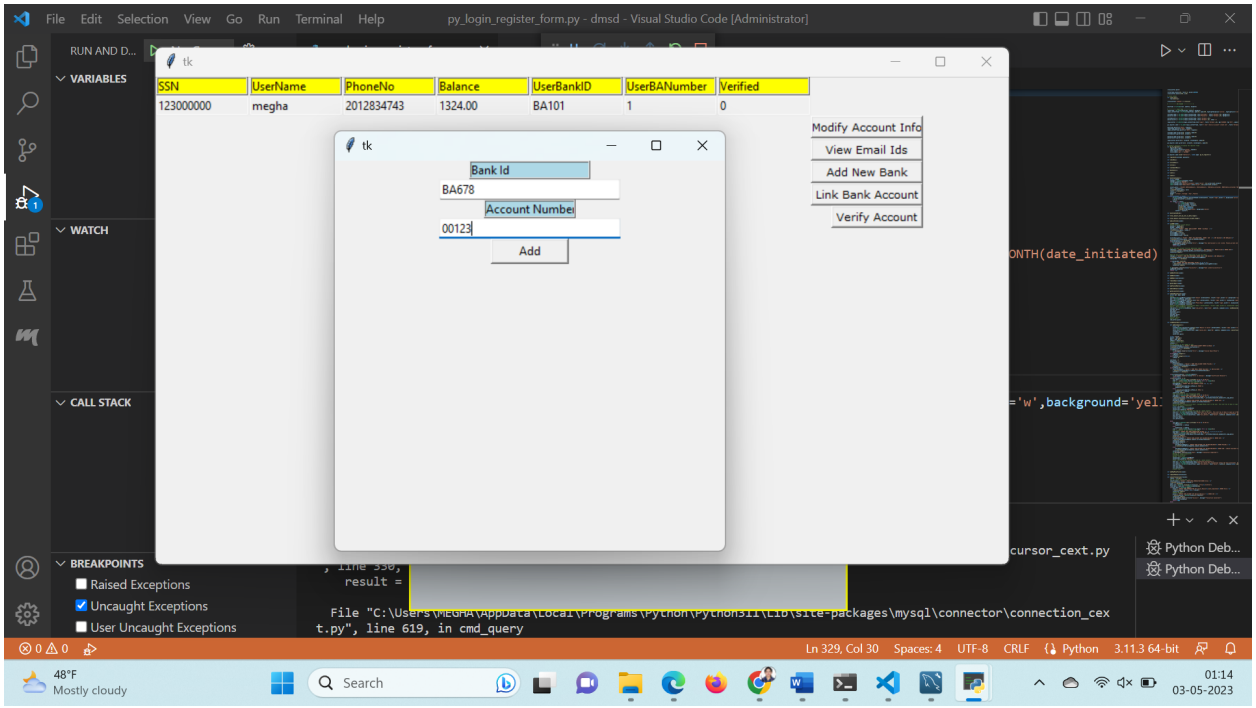


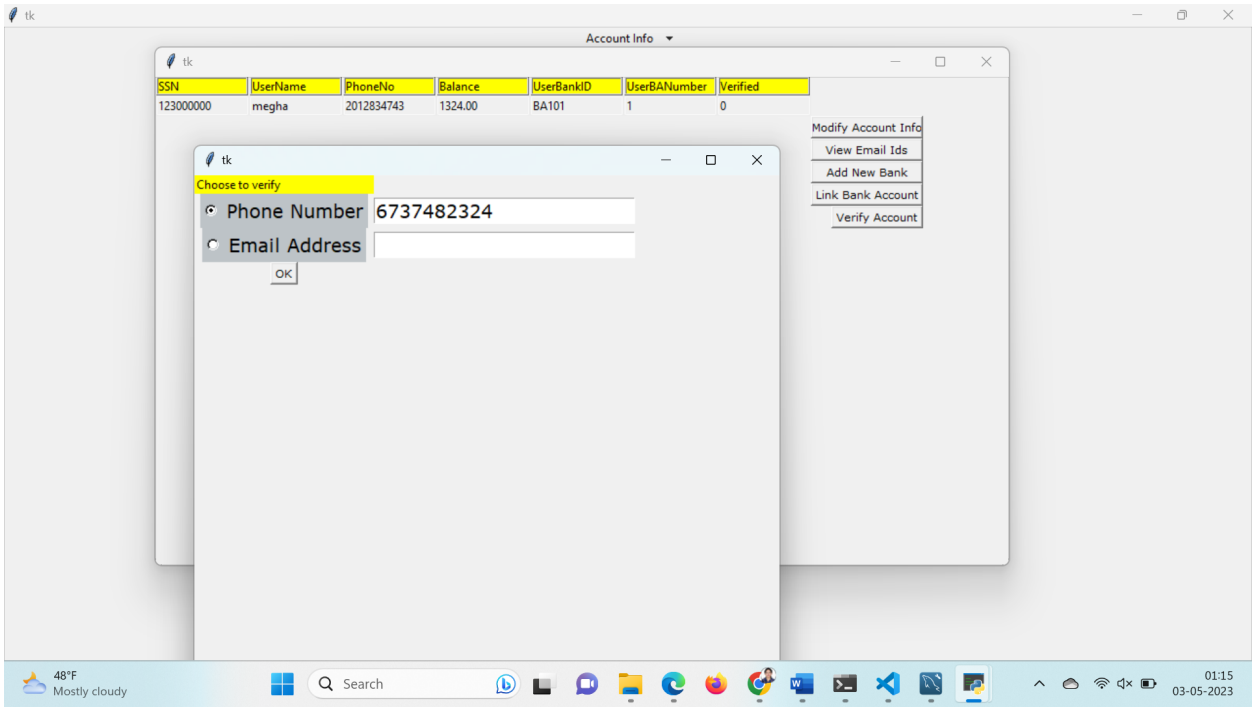

File Edit Selection View Go Run Terminal Help
py_login_register_form.py - dmsd - Visual Studio Code [Administrator]
EXPLORER
DMSD
__pycache__
static
templates
app.py
banking.py
get-pip.py
mainform.py
p1.py
p2.py
p2.txt
py_login_register_form.py
py_mainform.py
py_tkinterApp.py
test1.py
OUTLINE
TIMELINE
py_login_register_form.py
py_login_register_form.py > requestMoney
605
606 def modifyDetails(uname):
607 val = (username.get(),mphone.get(), uname)
608
609 select_query = "SELECT * FROM `USER_ACCOUNT` WHERE `UserName` = %s"
610 c.execute(select_query, (uname,))
611 currentuser = c.fetchone()
612
613 #check if ssn and username does not exist already
614 dupCheck = "SELECT * FROM USER_ACCOUNT WHERE UserName = %s"
615 c.execute(dupCheck,(username.get(),))
616 userExists = c.fetchall()
617 if(len(userExists) != 0):
618 tk.messagebox.showerror(title="Error in Updation!", message="UserName already exists")
619 addModify.withdraw()
620 return False
621
622 #check if phone num exists in elec_address table
623 phoneCheck = "SELECT * FROM ELEC_ADDRESS WHERE Identifier=%s"
624 c.execute(phoneCheck,(mphone.get(),))
625 phoneValid = c.fetchall()
626 if(len(phoneValid) == 0):
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\MEGHA\dmsd>
Ln 833, Col 31 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
51°F Cloudy
Search
10:32 03-05-2023


File Edit Selection View Go Run Terminal Help
py_login_register_form.py - dmsd - Visual Studio Code [Administrator]
EXPLORER
DMSD
__pycache__
static
templates
app.py
banking.py
get-pip.py
mainform.py
p1.py
p2.py
p2.txt
py_login_register_form.py
py_mainform.py
py_tkinterApp.py
test1.py
OUTLINE
TIMELINE
py_login_register_form.py
py_login_register_form.py > requestMoney
585
586 def addFieldsModify(uname):
587 global addModify
588 addModify = Toplevel(accountInfo)
589 addModify.geometry('800x500')
590 global ssn, username, mphone
591 #ssnLabel=Label(addModify,width=15,text='SSN',borderwidth=2, relief='ridge',anchor='nw',background='l
592 #ssn = Entry(addModify, width=30)
593 unameLabel=Label(addModify,width=15,text='UserName',borderwidth=2, relief='ridge',anchor='w',backgrou
594 username = Entry(addModify, width=30)
595 phnLabel=Label(addModify,width=15,text='Phone',borderwidth=2, relief='ridge',anchor='w',background='l
596 mphone = Entry(addModify, width=30)
597 mdfy_button = tk.Button(addModify, name="mdfy_button", text="OK", width=10, command=lambda: modifyDeta
598 #ssnLabel.pack()
599 #ssn.pack()
600 unameLabel.pack()
601 username.pack()
602 phnLabel.pack()
603 mphone.pack()
604 mdfy_button.pack()
605
606 def modifyDetails(uname):
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\MEGHA\dmsd>
Ln 833, Col 31 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit
51°F Cloudy
Search
10:32 03-05-2023


File Edit Selection View Go Run Terminal Help
py_login_register_form.py - dmsd - Visual Studio Code [Administrator]
tk
SSN UserName PhoneNo Balance UserBankID UserBANumber Verified
123000000 megha 2012834743 1324.00 BA101 1 0
Modify Account Info
View Email Ids
Add New Bank
Link Bank Account
abc@gmail.com
meghammmm@yahoo.com
mm2773@njit.edu
mmmeghamanoj@gmail.com
star@gmail.com
testinggg@gmail.com
New Email Id
Delete Email
vp12323@gmail.com
Add
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
Raised Except
Uncaught Exc
User Uncaugh
cursor_cext.py
Python Deb...
Python Deb...
UTF-8 CRLF Python 3.11.3 64-bit
48°F
Mostly cloudy
Search
01:13
03-05-2023

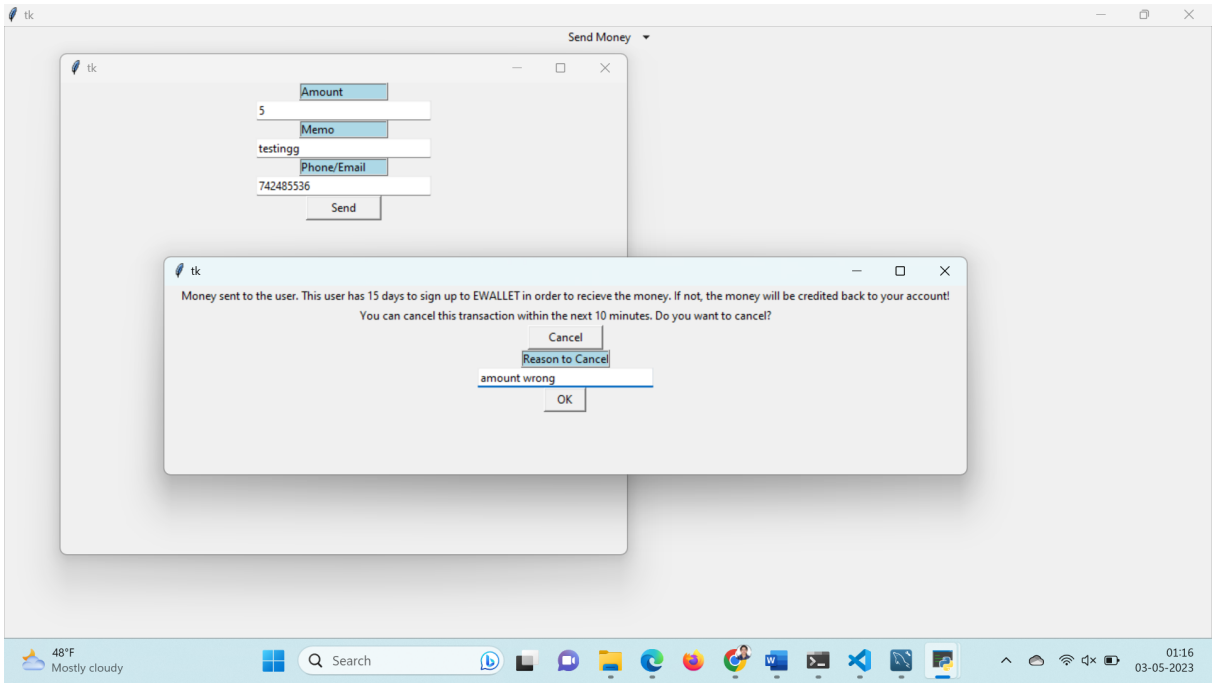

File Edit Selection View Go Run Terminal Help
py_login_register_form.py - dmsd - Visual Studio Code [Administrator]
tk
SSN UserName PhoneNo Balance UserBankID UserBANumber Verified
123000000 megha 2012834743 1324.00 BA101 1 0
Modify Account Info
View Email Ids
Add New Bank
Link Bank Account
abc@gmail.com
meghammmm@yahoo.com
mm2773@njit.edu
mmmeghamanoj@gmail.com
star@gmail.com
testinggg@gmail.com
New Email Id
Delete Email
star@gmail.com
Remove
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
Raised Except
Uncaught Exc
User Uncaugh
cursor_cext.py
Python Deb...
Python Deb...
UTF-8 CRLF Python 3.11.3 64-bit
48°F
Mostly cloudy
Search
01:13
03-05-2023

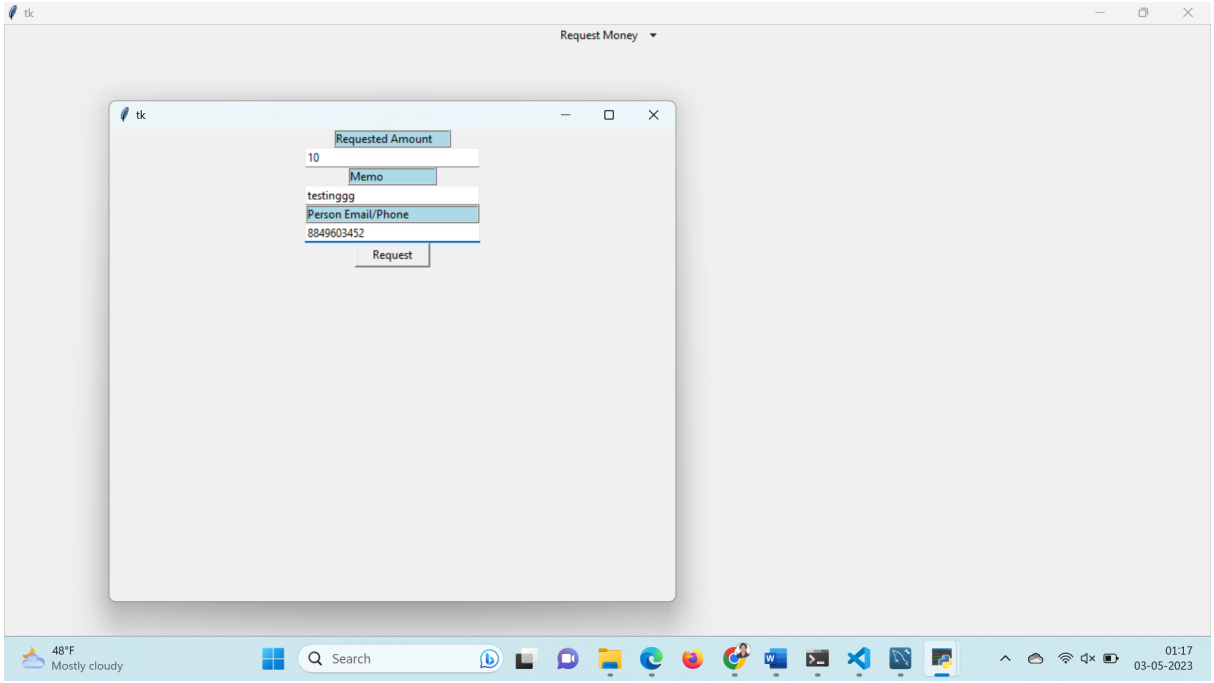

File Edit Selection View Go Run Terminal Help
py_login_register_form.py - dmsd - Visual Studio Code [Administrator]
tk
SSN
UserName
PhoneNo
Balance
UserBankID
UserBANumber
Verified
123000000
megha
2012834743
1324.00
BA101
1
0
Modify Account Info
View Email Ids
Add New Bank
Link Bank Account
Verify Account
Bank Id
BA678
Account Numbe
00123
Add
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions
result =
File "C:\Users\MEGHA\AppData\Local\Programs\Python\Python311\Lib\site-packages\mysql\connector\connection_cex
t.py", line 619, in cmd_query
cursor_cext.py
Python Deb...
Python Deb...
Ln 329, Col 30
Spaces: 4
UTF-8
CRLF
Python
3.11.3 64-bit
48°F
Mostly cloudy
Search
01:14
03-05-2023


tk
Account Info
SSN
UserName
PhoneNo
Balance
UserBankID
UserBANumber
Verified
123000000
megha
2012834743
1324.00
BA101
1
0
Modify Account Info
View Email Ids
Add New Bank
Link Bank Account
Verify Account
Choose to verify
Phone Number
6737482324
Email Address
OK
48°F
Mostly cloudy
Search
01:15
03-05-2023

3.3.4

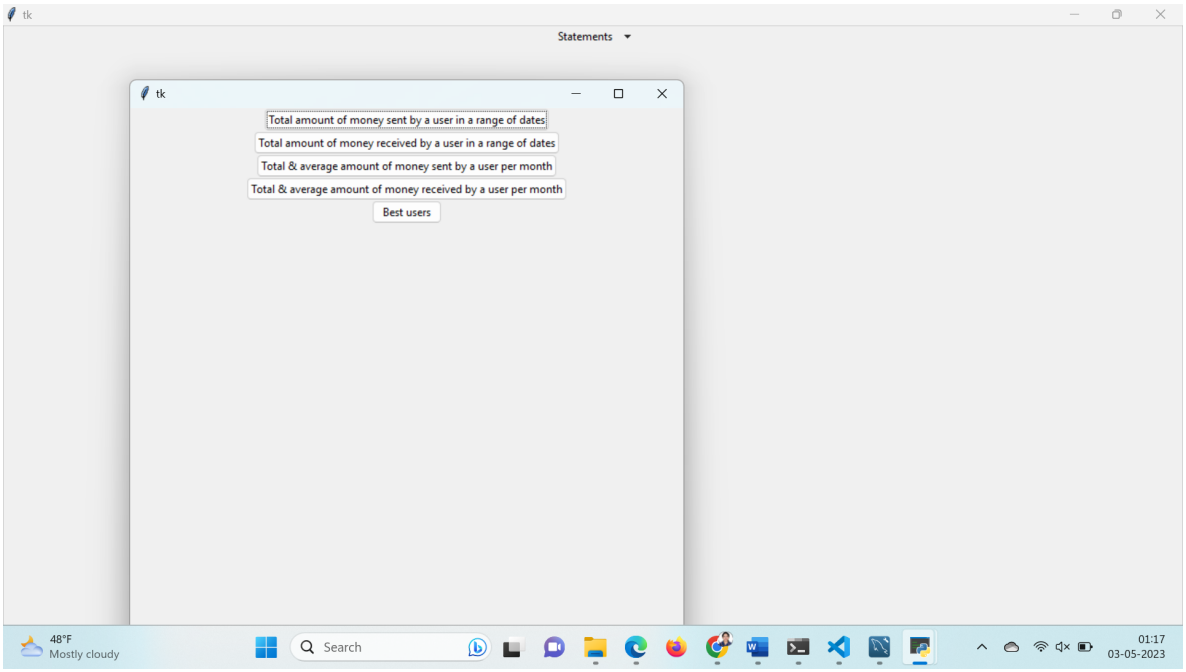
Send to and Request From actions are enabled as options of Main Menu that allow the user to send amount to a user on a phone number or email_address linked to the account where the money can be transferred. If transferred, we enabled a special functionality that ensures the status of receiving money within 15 days. If the destination account user fails to accept the transaction, the transaction will be cancelled and the amount will be credited back to the source account.

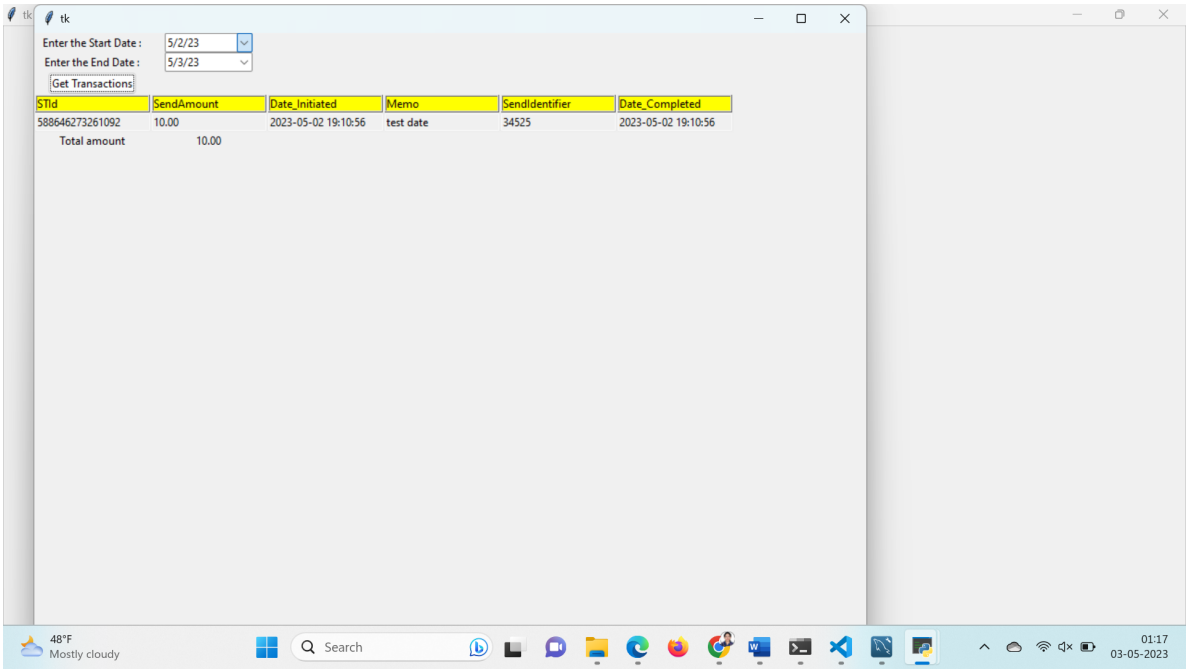


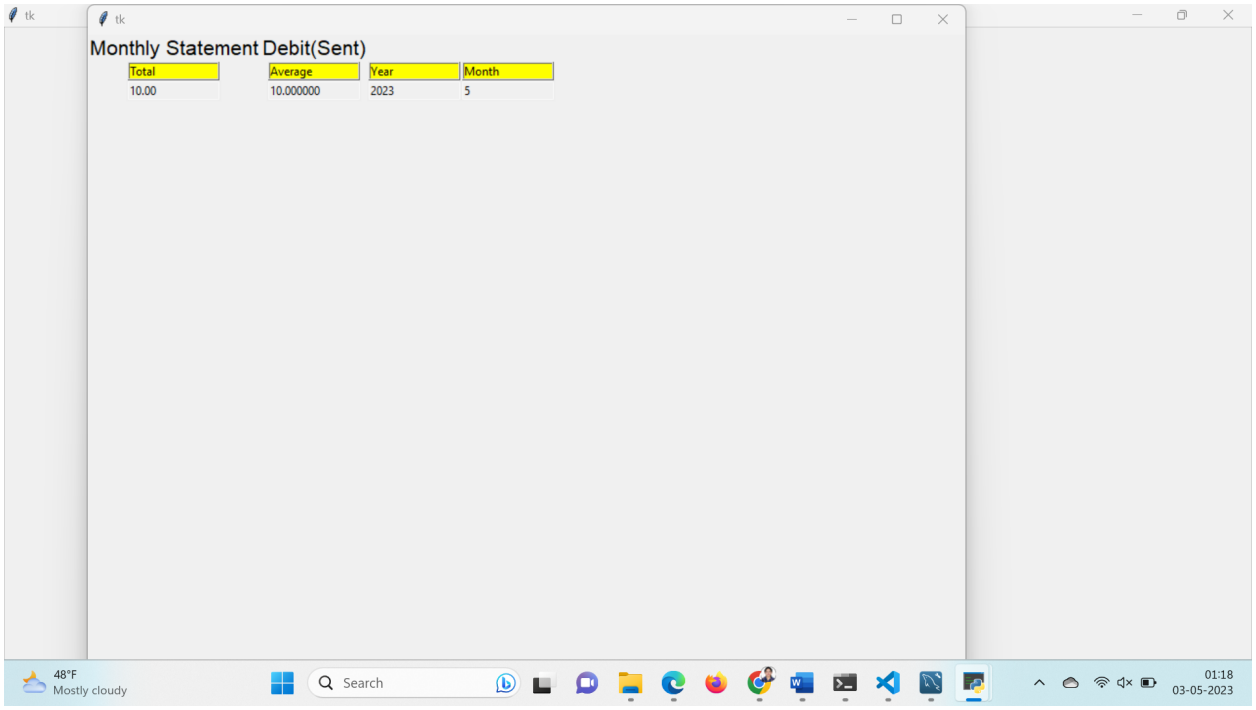


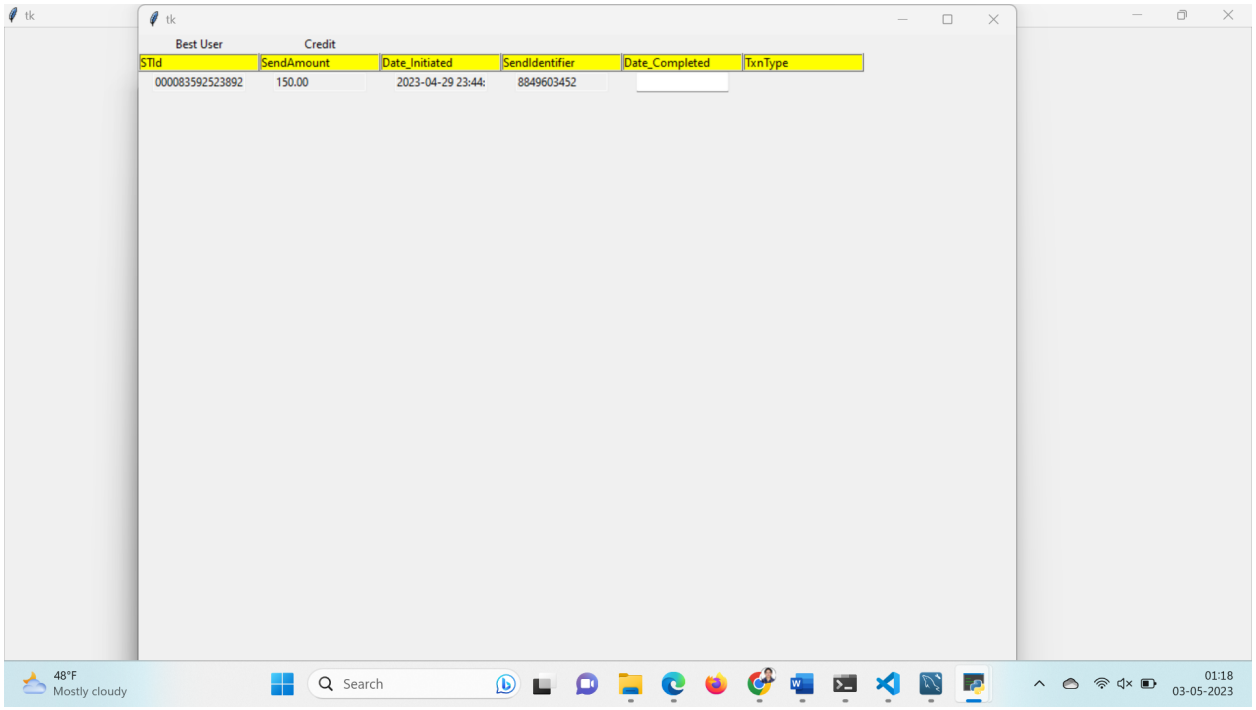
3.3.5

Additionally, MyWallet Statement Window enables the user to look into various interesting functionalities that can display the total amount of money they sent or received in a range of dates or in a specific month with a basic numeric value. Also, the maximum amount transactions that go out or come in per month. Wallet Application can also show the maximum total amount of money sent or received referred to as the BEST USERS from all our Wallet Users.






tk
Monthly Statement Debit(Sent)
Total
Average
Year
Month
10.00
10.000000
2023
5
48°F
Mostly cloudy
Search
01:18
03-05-2023


tk
Best User
Credit
STId
SendAmount
Date_Initiated
SendIdentifier
Date_Completed
TxnType
000083592523892
150.00
2023-04-29 23:44:
8849603452
48°F
Mostly cloudy
Search
01:18
03-05-2023

4. Challenges

Connecting the Database to the Application and ensuring a clean Integration with repetitive testing, and no errors has been a cumbersome task as a whole. Making some functionalities work has been quite challenging too.

Creasting an Entity Relationship Model felt challenging as it required combined synthesizing of thoughts to come up with the right one, moreover it is the building block for the complete project. Our assumptions were a little contrast with that of the actual Deliverable, but we improvised on our cognition to understand the defining entities, attributes and its types with constraints applied.

Translating the ER Model via Relational Mapping to create a Logical Design model felt challenging during the last steps of Algorithm when we had to choose the best option from the four. Choosing any of those options could affect on the overall result of the Application and in what ways we could represent and populate the data effcicienty.

Scenarios where it required us to create complex SQL queries was challenging to align with the objective as it required us to pay more attention to semantic errors, optimization while testing and illustrating the end results in different conditions.

Though we are familiar with Python Programming, developing an application System in Python was a little challenging as we had previously worked in Java (Java Database Connectivity). Figuring out on the framework, its Interface design patterns and how to execute in chronology and depict on the real-time Frame has been challenging.

5. Conclusion

We as a group, are grateful for the opportunity to work with the Professor, to learn and practice on the development of the Wallet Database Application System, though we faced challenges in each stage, it has been an exciting and beneficial experience to improve our skills to career development. This helped us create our foundation and concepts towards being a Database Designer.