Description of the application and resources used

About Program

For my final project, I decided to make a banking program. The application allows the user to interact with the program through a GUI (Graphical User Interface), and the program displays the user's bank account information. The Python Tkinter library is utilized to make the GUI. User data is read from and written to a CSV file.

Resources Used

Resources used to create this application include Murach's Python Programming Chapter 18 PowerPoint (How to build a GUI program), the corresponding chapter in the Murach's textbook, GeeksforGeeks Python Tkinter web tutorial, and finally python files included in the "How to Build a GUI Program" resource provided for this course were used as a guide.

Main Menu (Dashboard/Function)

When a user runs the program, a Tkinter widget appears. The widget displays two options for the user. The user can either click "Sign Up" or "Login". If the user does not already have an account, the user will click "Sign Up" because there are no user files in the current directory for the program to draw information from.

Sign Up (Function)

Once the user clicks Sign Up, a widget will pop-up and prompt the user to enter a username, email address, and password. The user must make entries into the name, email, and password fields. The above fields cannot be left blank. If any of the fields are blank, the program will output an error message that all fields must have entries. Additionally, since this is a secure application, there are restrictions on acceptable names, emails, and passwords. If the user enters a name that already exists in the database, an error message will display that the username already exists. The program will not proceed until a unique (not in the database) name is entered. This safeguard prevents the program from overwriting an existing account with the same username. A valid password requires a minimum of eight characters, must include a capitol letter, and contain a number. The user has the option of hiding or showing their password. The password is hidden by default. If the password does not meet the above requirements, the user will not be able to continue until an acceptable password is entered. The email must be in the proper email format. For example, example@gmail.com, school@university.edu, or yahoo@yahoomail.com.

CSV File (File)

Once the user has successfully "Signed Up", a CSV file based on the username is created in the current directory. It is therefore crucial to check if the username already exists before a user creates a new account. The new user CSV file will have entries for the username, password, and email. Additionally, it will contain the user's balance which is zero dollars by default. The username is stored in row zero, the email in row one, the password in row two, and the current balance in row three. When the deposit and withdraw functions are run, the amounts are added or subtracted from the data in the third row of this file.

Login (Function)

Once the account is created, the user can log into the new account with the username and password entered. If the password is incorrect, the user will not be able to access the account. As with the “Sign Up” function, the user will have the option to hide or show their password.

Dashboard (Dashboard/Function)

Once a user successfully logs into her account, the login function disappears, and a dashboard appears in its place. The dashboard welcomes the user by name. In the dashboard, the user has four button options. The options allow the user to either view account information, deposit money into the account, withdraw money from the account, or view the current account balance.

Account Info (Function)

When the user clicks the “Account Info” button, she can view her name, email, and password.

See Balance (Function)

When the user clicks the “See Balance” button, she can view her account balance.

Withdraw (Function)

This function allows the user to deduct money from her account. Once she clicks confirm the changes are saved.

Deposit (Function)

This function allows the user to add money to her account. Once she clicks confirm, the changes are saved.

Process

1. I wanted to design a challenging project that would utilize the skills that I learned in this and the previous Python course. After much consideration, I finally decided on a banking application because it would incorporate the use of CSV files, data validation, data types, lists, loops, and other various concepts learned in both courses.

2. I began designing my project by creating a wireframe. The wireframe served as a visual representation of what the final project would look like.

3. I constructed a hierarchy diagram to roughly map-out how the code would execute and the order in which the functions would be called.

4. Next, to keep track of weekly tasks, I put together a schedule for myself so I could space out the activities and not lose track of time.

5. I researched and learned Tkinter. My sources were the Chapter 18 (murach's python programming) PowerPoint, the corresponding chapter in the textbook, and the GeekforGeeks python tutorial. This site had a handy index with Tkinter terms and instructions on how to create GUI windows, buttons, labels, etc. Additionally, I also used the Python modules included in the "How to Build a GUI" resource included in the course to play around and get a feel for how Tkinter works.

6. After getting a good understanding of Tkinter, I created an outline of what my code should do. This was useful in planning and served to test for logic errors in my program.

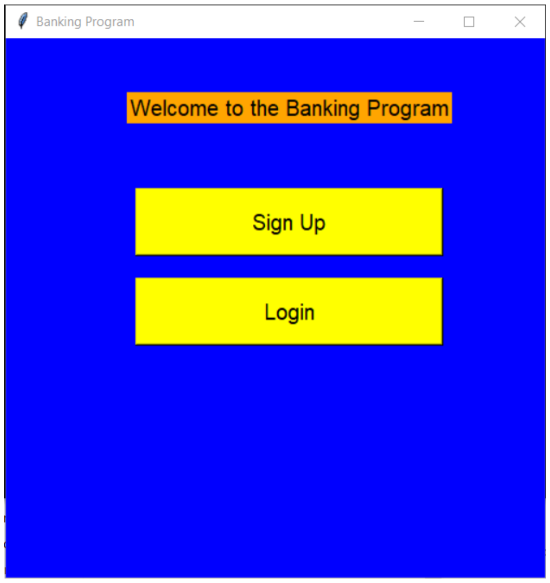
7. An algorithm was created for the logic I would need to make my application work.

8. Next, I began to write the lines of code. I had to perform debugging to fix any errors I discovered.

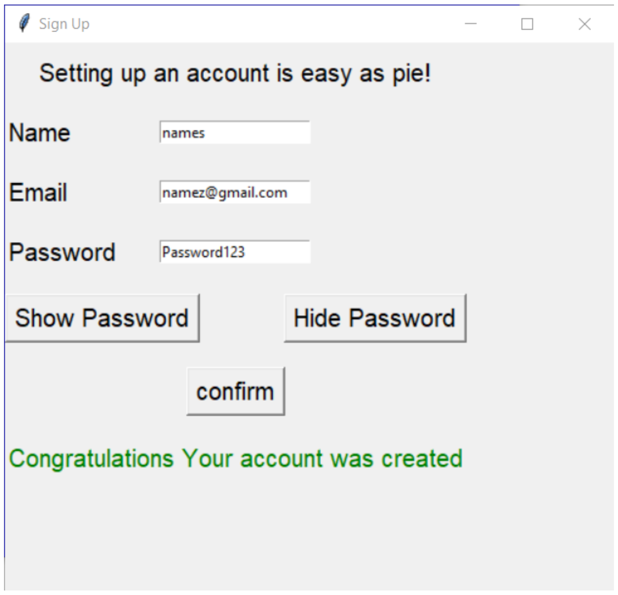
9. Finally, I wrote a comprehensive list of all the scenarios I needed to check for while debugging and testing my code.

Implementation

For the program, I implemented Tkinter to make an interactive GUI for the user. I used Tkinter to make an interactive dashboard that gives a user an option to either "Sign Up" or "Login".



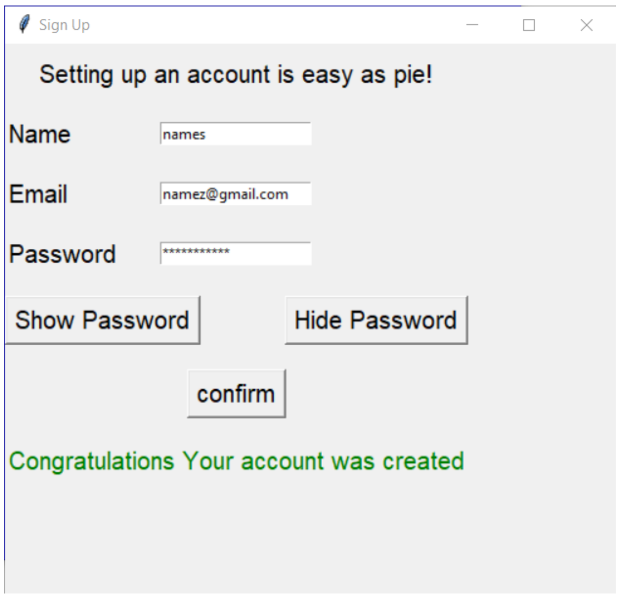
I also used Tkinter to create a Sign-Up screen. When a user decides to click "Sign Up", they must input a username, password, and email. A user would type the information into the Tkinter window. Additionally, the user would have the option to hide/show her password.



Setting up an account is easy as pie!

Name	names
Email	namez@gmail.com
Password	Password123

Congratulations Your account was created

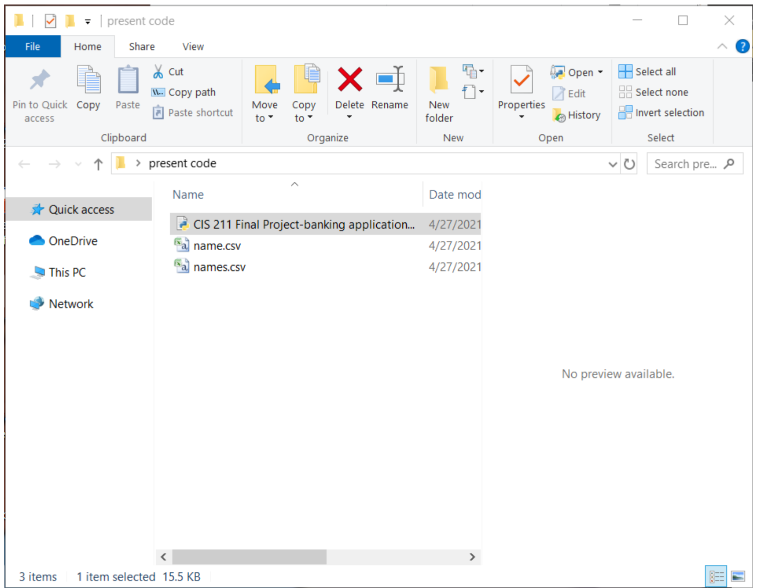


Setting up an account is easy as pie!

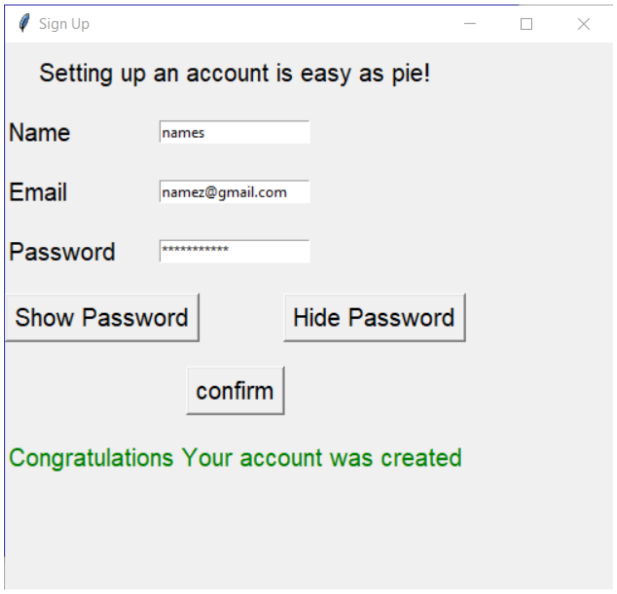
Name	names
Email	namez@gmail.com
Password	***********

Congratulations Your account was created

I implemented a CSV file to save and retrieve data. A CSV file is created from a successful "Sign Up" based on the username.



Name	Date mod
CIS 211 Final Project-banking application...	4/27/2021
name.csv	4/27/2021
names.csv	4/27/2021

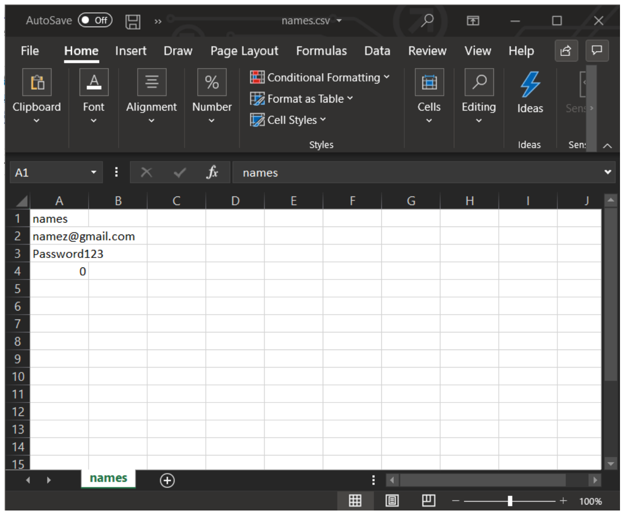

Sign Up

Setting up an account is easy as pie!

Name	names
Email	namez@gmail.com
Password	***********

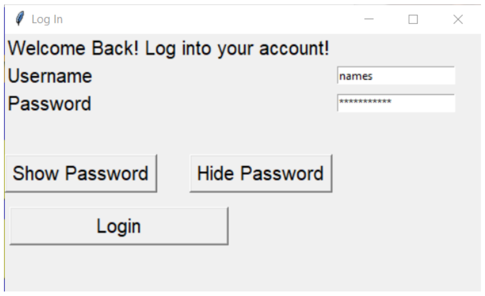
Congratulations Your account was created

This CSV file stores the username, email, password, and sets the initial account balance to $0.


names.csv

	A
1	names
2	namez@gmail.com
3	Password123
4	0

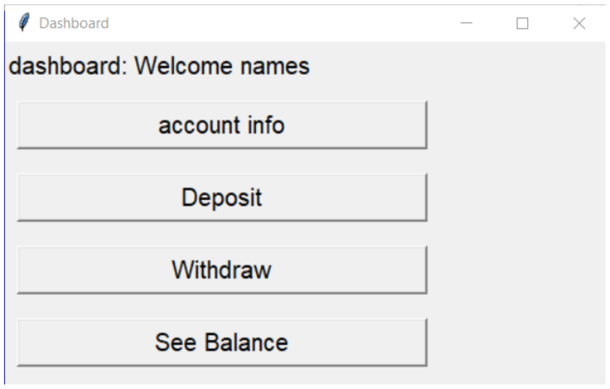
If all the requirements are met, the user is given the option to login. The program will then check to see if the CSV file exists and makes sure all the information entered is correct before opening the file to perform operations.


Log In

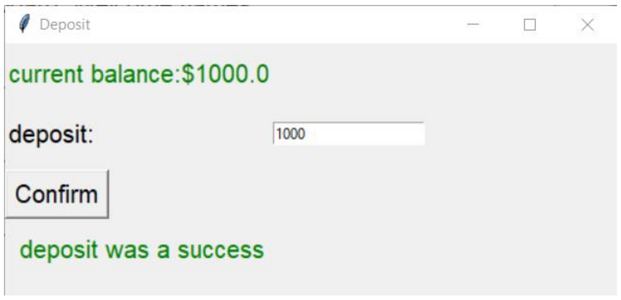
Welcome Back! Log into your account!

Username	names
Password	***********

If all the information is correct, the Tkinter dashboard window will appear. The dashboard presents the user with four options: account info, deposit, withdraw, and see balance.



Tkinter was implemented to create a GUI for the deposit and withdraw windows. Data is added to or subtracted from the CSV file. The deposit function allows the user to deposit money into her account. The withdraw function allow the user to withdraw money from her account.

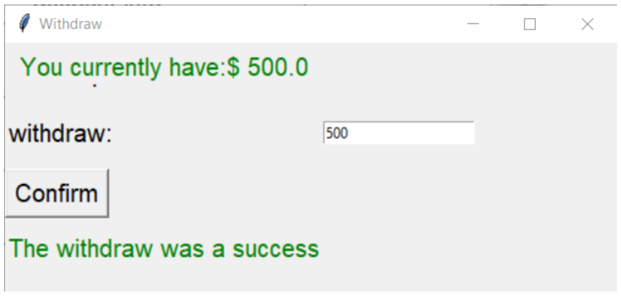

Deposit

current balance:$1000.0

deposit: 1000

Confirm

deposit was a success


Withdraw

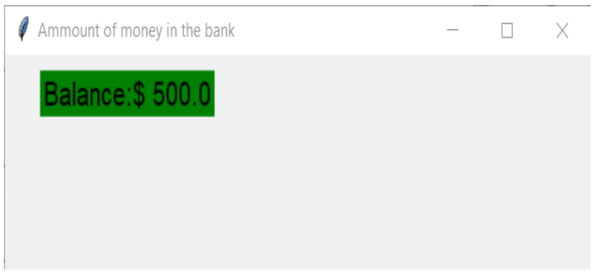
You currently have:$ 500.0

withdraw: 500

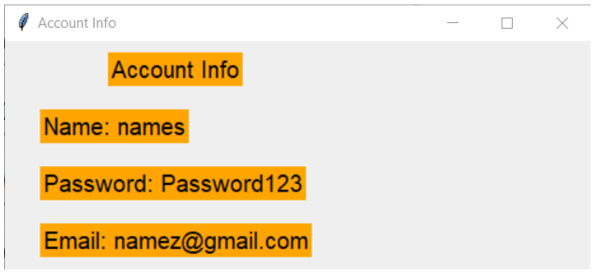
Confirm

The withdraw was a success

The last two Tkinter windows display user account information. The balance function displays the balance. The account info function shows the user's account info.


Ammount of money in the bank

Balance:$ 500.0


Account Info

Account Info

Name: names

Password: Password123

Email: namez@gmail.com

Learning Outcomes

While developing this project, I had the following learning outcomes.

1. I learned how to search the current directory for a specific file.
2. I learned how to search for specific data in a file.

3. I programmed a function that created a CSV file to capture specific user input.
4. I utilized and converted different data types in Python.
5. I Incorporated Tkinter to make an interactive Graphical User Interface.
6. I developed a GUI that had a user interface which consisted of frames, buttons, labels, and text entry fields in a grid format.
7. I used a CSV file to save and retrieve the data used by my project.
8. I designed a wireframe to visually map my Python application.
9. I utilized an algorithm to plan logic for creating my Python application.
10. I learned how to use data validation to control user input.
11. I learned how to use the re module to search regular expressions.