CS 2316
Homework 9a – GT Brokers
Due: Friday July 11th, Before 11:55pm
Out of 100 points

Files to submit: **1. HW9.py**

For Help:
- TA Helpdesk – Schedule posted on class website.
- Email TA's or use T-Square Forums

Notes:
- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- ***Do not wait until the last minute*** *to do this assignment in case you run into* problems.
- **Read the entire specifications document before starting this assignment.**

Premise

This homework is the first of a two-part assignment in which you will do some real world applications with databases and GUIs. For this first part of the assignment, you will be building a two page GUI that offers the user the chance to login, or in the case that the user does not have an account yet, register and then log in. You'll find this quite applicable to building programs with user accounts, and potentially your future CS4400 project.

Although this program does not require much new information, it does test simple SQL query and insertion abilities as well as new coding practices to create two GUI windows that can be opened and closed with the click of a button. **This new information may take some understanding, so please do not wait until the last minute to begin this assignment**.

Database Format Information

The database for this part of the assignment only has 1 table of users, conveniently called BrokerageUsers.

```
CREATE TABLE BrokerageUsers
        (USERNAME        VARCHAR(20)      PRIMARY KEY,
        FirstName        TEXT,
        Password         TEXT             NOT NULL
        Dollars          DOUBLE           NOT NULL)
```

The code above was used to create the table. As can be seen, the USERNAME and Password are non-NULL variables, but FirstName can be left blank and Null will be assigned instead of a name. Note that the USERNAME and FirstName are different. A USERNAME is required, but a FirstName is not. A USERNAME can only be stored in the database if it is 20 characters or less.

Multiple GUI Pages

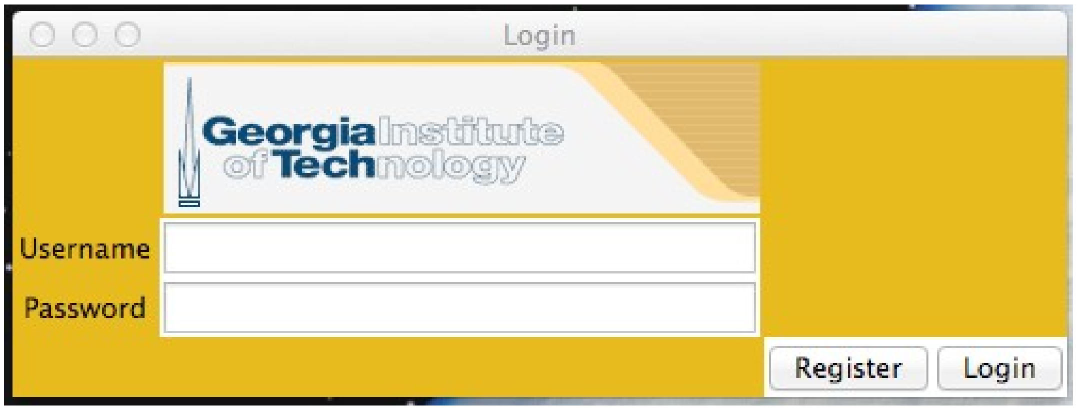
Multiple GUI window may not seem difficult, but it takes a certain method to show new windows while hiding old ones. It will involve the creation and button execution of several helper methods to transition between windows. Your __init__ method, for instance, should not actually create a GUI window, but instead call a method to do so. There will be two methods for GUI creations, two windows for writing and retrieving user information, and several helper methods for transitioning between the windows. It's up to you as to how many transition methods you will need, but one per transition should be enough. In these transition methods, you will .withdraw() a certain TopLevel window instance created for a certain window, while calling a method that deiconifies a different TopLevel instance for a different window. This means that you will be using one main Tk() instance for your main window and a TopLevel window instance for your other windows.

Method Name: __init__

This method will only be used to call your first GUI creation method for the login page window. It should create both the LOGIN GUI and the REGISTRATION GUI (but it will hide the registration GUI and only show the LOGIN GUI initially.) To keep your __init__ function from getting to large, it will call the LoginPage and RegisterPage methods to do the actual GUI creation.

Method Name: LoginPage

This method is called by the __init__ method to create the login GUI. This method is responsible for arranging and initializing your GUI. You should crate a GUI which looks like the one below:

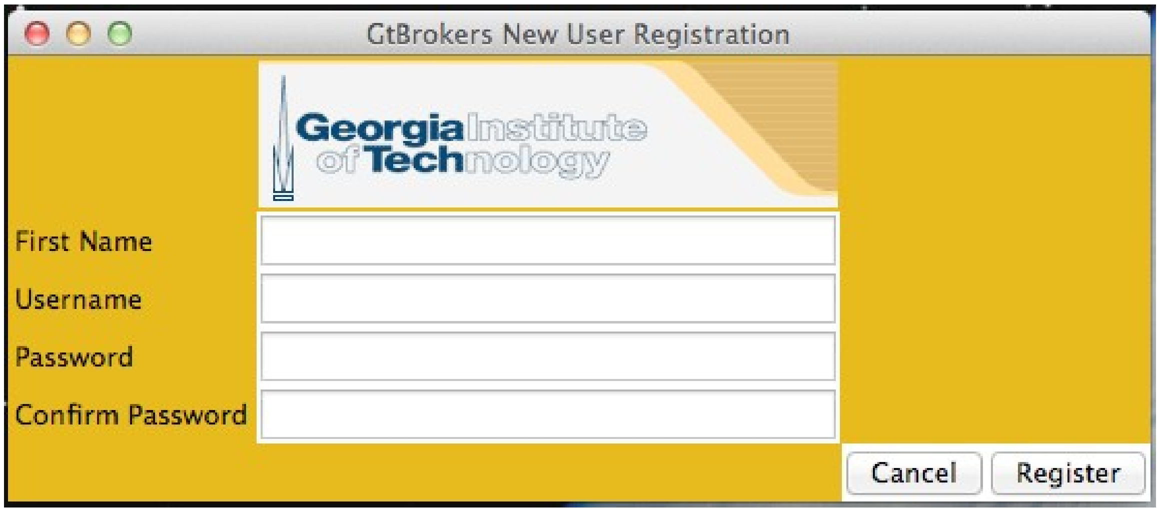


You may generate this window however you wish, but it is *highly* suggested that you use the grid layout manager. Things to note about the GUI you see:

- The top image can be any logo that is GT related. You must use an image found anywhere online that allows webscraping and retrieve the image using the urllib.request module. (in other words, you must download and display the image dynamically from the URL, you **may not** include the image as a file!)
- The labels are sticky to the east and the entry boxes are width 30 and state normal.
- The window is titled "Login"
- Clicking on the "Register" button should call a helper function that will hide the login GUI and show the Registration GUI.
- Clicking on the "Login" button should call the LoginCheck method

Method Name: RegisterPage

This method will create the registration window for filling out registration information. The registration GUI should look like this:


GtBrokers New User Registration
Georgia Institute of Technology
First Name
Username
Password
Confirm Password
Cancel
Register

You may generate this window however you wish, but it is *highly* suggested that you use the grid layout manager. Things to note about the GUI you see:

- The top image is the same as the one in the login page.
- Each entry is width 30 and state normal
- The labels are all sticky to the west
- The window is titled "GtBrokers New User Registration"
- Clicking on the "Cancel" button will call a helper method (which you could call BackToLogin for example...) to hide this GUI instance and redisplay the LoginPage window.
- Clicking on the "Register" button calls the RegisterNew method.

Method Name: Connect

Parameters: None
Return Value: Database connection

Description:
This method will use a try/except statement to attempt to connect to the database. If it cannot connect, have it display a message box asking the user to check their Internet connection. This method returns the database connection object. Use the database username and password we provided you.

Method Name: RegisterNew

Parameters: None
Return Value: None

Description:
This method will take the entries from the register page and put them into the database. You must check that both passwords entered in the two entries are the same (match) and that the username and password entries are not left blank. The password must contain at least one number and at least one uppercase letter. Use message boxes to describe any problems with the way the register page was filled out. Also, be sure to check to see if the username is already used within the database. Note that text in our database is considered the same regardless of case, so "SummetJ" and "summetJ" would be considered the same username. You will need to check for existing usernames in a case-insensitive way if you are using Python to do the check instead of using the SQL database engine with an SQL query. If so, use a message box to tell the user that the username already exists, and don't try and add it again! If everything is ok, you can insert the information into the database. (Hint: The FirstName entry can be left blank. In this case, no FirstName should be inserted into the database). Lastly, every user that registers with GT Brokers automatically gets 100,000 dollars put into their account, so be sure to include that in your SQL. Once registered, a message box should tell the user that they are now registered and a helper method should be called to hide the registration window

and display the Login window. Be sure to .commit() the database when you're done inserting data

Method Name: LoginCheck

Parameters: None
Return Value: None

Description:
This method will check to see if what was typed into the login page matches any users currently in the database. To do this, you call the Connect method, create a cursor from the database connection that is returned, and check what was passed to see if any matches are returned. You may also want to get the first name of the user to use on the 2nd part of this homework later (hint hint). If it finds no matches, it should display a message box explaining that the user entered an unrecognizable username/password combination. If it does match, you should display a message box saying you logged in successfully. You should also close the login GUI window, as you are finished with it.

A sample gtBrokers login that already exists in the database is username: gburdell3 and password: moneyMaker. You can try using this to test you login code before you have your registration code working. Please do not mess up this account information or other students will not be able to successfully use it for testing.

Grading:

You will earn points as follows for each piece of functionality that works correctly according to the specifications.

LoginPage GUI		**20**
GUI has image at top	6	
GUI has all components with proper characteristics	4	
Login button works as described	4	
Register button works as described	4	
GUI uses message boxes for errors	2	
RegisterPage GUI		**20**
GUI has image at top	6	
GUI has all components with proper characteristics	4	
Register works as described	4	
Cancel brings user back to login page	4	
GUI uses message boxes for errors	2	
LoginCheck()		**25**
Uses Connect() properly	5	
Correctly checks to find matching user	10	
Uses message boxes for errors with user match	5	
Closes GUI with match	5	
RegisterNew()		**25**
Uses Connect() properly	5	
Correctly puts user in database	10	
Uses message boxes for errors with user entries	4	
Checks for usernames already taken	4	
Commits db when done	2	
Connect()		**10**
Correctly connects to database and returns database	10	