Lab week 8 (ISYS2120 sem2 2022)

Welcome to week 8's lab. This week we are going to focus on practical skills for applications that access a database.

Important administration note: Due to the National Day of Mourning, all classes on Thursday September 22 are cancelled. Students from those classes should still aim to complete the steps stated below. Try on your own device, doing the work by yourself following the instructions, if possible (but connect through the VPN, see below). If you need help or advice, you can join another Zoom-based lab during the week (details and links are found on the "Zoom" tab of the unit's Canvas site). You can ask questions about any aspect on Edstem.

*Important practical note: Any work on these tasks will need to access university machines for some services (computation and/or postgresql access), and for tha, you will need to be **inside** the University network. If you are not performing this from a site on campus, you will need to use the VPN to connect to the University network, before trying any of these activities. If you work on a machine in a university computer lab, a few aspects will be different than when you work on a personal laptop.*

Step 0: Download all the materials in this weeks (wk08) tutorial on Canvas

1. unidbschema.sql
2. isys2120_2022s2_wk08_python_db_connection.ipynb
3. isys2120_2022s2_wk08_codebase_ucpu0.zip

A. Interactive queries through pgadmin4

pgadmin is a client that can access postgresql databases; it allows a variety of admin activities, such as seeing which databases and tables exits, but it also provides a window where you can type and run queries (similar to what Grok has, but without automatic grading!)

On a personal machine, if pgadmin4 is not already installed, you can download and install it, from https://www.pgadmin.org/download/
In a uni lab, it should be installed already; so search for "pgadmin" and start it running.

Once pgadmin4 starts, click on "Add New Server" link, or else "Servers" -> "Create" -> "Server". You should get a dialogue, where you can go to the tab for "Connection" and enter the details.
Prefilled should be port 5432 and maintainance database should be postgres
Fill in Hostname to be soitpw11d59.shared.sydney.edu.au
Give the username (which starts with y22s2i2120_) and password you have on the postrgesql system (they were sent to you in email – these are different from your unikey and its password!)

Once connected, in thee left panel, look at browser and scroll down through all the many databases, to find the one named the same as your account name. Click on it, and then launch the query tool (the icon probably looks like a  diskpack with a

triangle). You should then get a query window above an output window. Load the file called unidb.sql which you got from Canvas, into the query editor window, and run it with the 
Now clear the contents of the query editor, and type SELECT * FROM STUDENT;
Now run this query, and see the 6 rows appear in the output window.
You can experiment with other queries, depending on the time you have, but this should feel very similar to working with Grok.

B. Python code with pyscopg2

We will be using the university Jupyter notebook servers for this section. You can connect to these servers via a web browser: https://ucpu0.ug.cs.usyd.edu.au
You will be prompted with a Jupyter login screen. Use your normal unikey and password for this service. Once Jupyter is running, navigate to a folder that you will use for this lab, and there upload the file isys2120_2022s2_wk08_python_db_connection.ipynb which you got from Canvas.

After uploading, it should be listed under "Files" in the Jupyter display; click on the file and it should show "Running" on the right. There should be a bowser tab with the notebook (headed "ISYS2120", and then "Tutorial Week 8"); you should run each cell in turn, by selecting that cell and then using the "Run" icon. Some cells will ask you to fill in some SQL code (where it says "# [TODO]"). Be sure to run cells in order down the page; if you find you need to go back and change something in a cell, re-run that cell and everything after it.

C. Running a web app

Before we finish with the Jupyter notebook, use it to get the web-app's codebase into the file system on ucpu0. Use the main page of the Jupyter server to "upload" the compressed isys2120_2022s2_wk08_codebase_ucpu0.zip

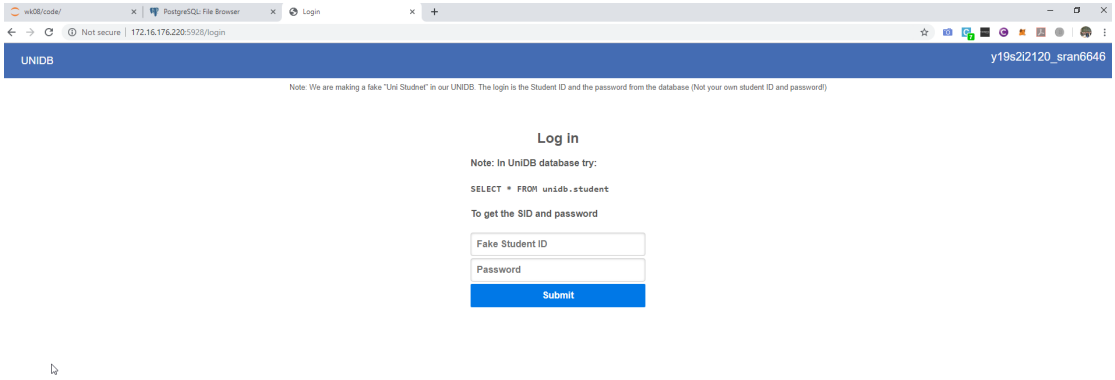
You need to login to the machine ucpu0.ug.cs.usyd.edu.au (it's the Jupyter server, but we will use it for editing and then running the web app too). Use a terminal or command window on your device or lab machine, to execute ssh abcd1234@ucpu0.ug.cs.usyd.edu.au (replace abcd1234 by your own Unikey). At the prompt for password, give your Unikey password. Alternatively, use PuTTY to connect to ucpu0.ug.cs.usyd.edu.au on port 22.

Once logged in, use "ls" (to list contents of the directory you are in) and "cd" commands (to change to a subdirectory) to get to the place where you put the file isys2120_2022s2_wk08_codebase_ucpu0.zip Now uncompress this, by running unzip isys2120_2022s2_wk08_codebase_ucpu0.zip You should now see a directory called code in that directory when you look with the Jupyter browser. Inside code, you now need to edit the 'config.ini' file. While you can use the Jupyter browser to edit any '.py' file by clicking on it, for other files, you must click on the tick box first, then select 'Edit' from the menu options. In editing the file, set username and password for the postgresql account as sent in email, and for port, use 5 followed by the last 3 digits of your student id. Then click 'File'->'Save'

Now you will move back to your still open shell/terminal. Navigate to the 'code' folder. Do 'ls' to see the same files as listed in the Jupyter browser.
To begin your flask application, you should run './threetier.command' [*Very important*: do this in the ssh or PuTTY terminal; do not try this from Jupyter interface]
If you get a permission denied error, you need to first give permission to execute by running 'chmod +x ./threetier.command'

When you run the command, the Flask webserver goes online on whichever port you specified (that is, port 5-followed-by-last-3-digits-of-your-sid).
You can open http://ucpu0.ug.cs.usyd.edu.au:YOURPORT on a web browser to browse the GUI.
To halt the server, just press Ctrl+C to terminate the process in PuTTY or the ssh trminal [It is crucial that you do this before ending the connection]

You should see



To login to the webapp, you need to pretend to be one of the application accounts, which are given by the "Students" table.

Output pane

Data Output | Explain | Messages | History

	studid integer	name character varying(20)	password character varying(10)	address character varying(50)
1	307088592	John Smith	Green	Newtown
2	305422153	Sally Waters	Purple	Coogee
3	305678453	Pauline Winters	Turkey	Bondi
4	316424328	Matthew Long	Space	Camperdown
5	309145324	Victoria Tan	Grapes	Maroubra
6	309187546	Niang Jin Phan	Robot	Kingsford

This will get you a "Welcome <name>" page, with a bar at the top with functionality to execute; but you need to do some work first.

Edit the database.py file (either with an editor that works textually, such as nano, or else from the Jupyter server interface) to fill in the code for the 'get_transcript(sid)' function. Look at how it was done for list_units() and for your previous exercises in Jupyter. Once that is done, you should be able to look at the page by clicking the appropriate option in the bar at the top of the GUI
DO NOT LEAVE THIS LAB without ending the flask processing (type Control+C in the ssh or PuTTY terminal.) You can then close the ssh or PuTTY terminal itself.

Before the end of the week
Before Sunday September 25, you need to finish the assessments that are due: SQL Tasks 14, 15; Quiz 8).