**B.Sc. (Ordinary) in Computer Systems and Networking**

**Advanced Scripting Project CA**

**Development of a Systems Administration tool, with a Graphical User Interface using the Kivy API**

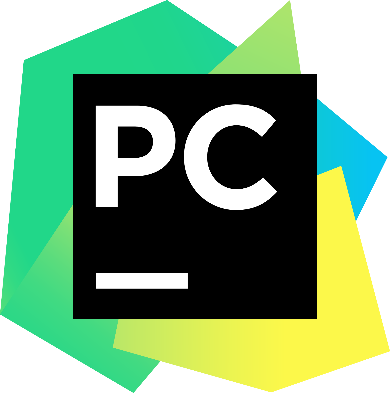


Figure 1: Technologies Logos

**Introduction**

For my Advanced Scripting project 2019 I must *“explore a python library/api that is useful for systems administration. You are to identify use cases for the library and to then implement the use cases. You are to showcase the programming skills learned in class and your ability to identify, explore/learn and use a library that is commonly used by systems administrators”.*

To this, I have decided that I will develop a python application, to perform system reporting/administration tasks, that also makes use of a Graphical User Interface (GUI).

**API and Technologies Used**

* For development of this project, I will by using the Python language (version 3.7.2), and the corresponding IDE will be PyCharm, open source community edition. This project will focus development on the Windows platform.
* The primary API I will be using for the project is **“Kivy”**. A python library that I will use primarily for the designing of the GUIs for the project. **Kivy** is an *“Open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps.”* (Kivy, n.d.)
* I will also be making use of supplementary APIs/libraries for the system administration functions.
  + “**psutil**”; a python API for *“for retrieving information on running processes and system utilization (CPU, memory, disks, network, sensors) in Python”*. (Python Software Foundation, n.d.),
  + “**datetime**”; a python API for that “*supplies classes for manipulating dates and times in both simple and complex ways.”*

**Milestone Goals**

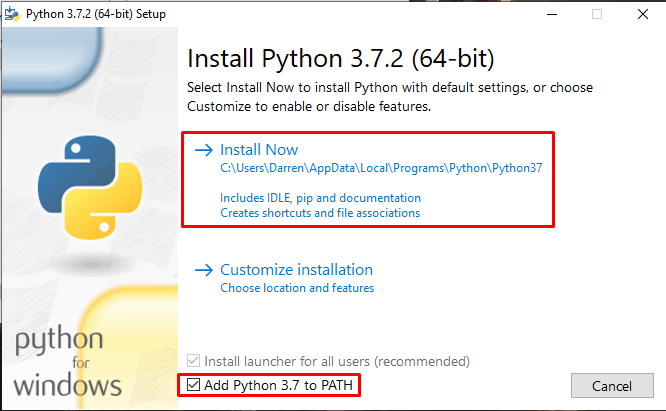
* **Milestone 2: *Demonstrate an understanding of the API/Technology***
  + Produce a prototype of the GUI; buttons that when clicked execute scripts and functions etc.
  + Create the python scripts for the various System Administration tasks.
* **Milestone 3: *Early stage development of your core idea/work***
  + Produce a prototype of the combined GUI and script functions in tandem.
* **Milestone 4: *Final Deadline, completion of project and documentation***
  + Produce and demonstrate the finished project application, GUI and scripts/functions.
  + Submit both the finished project and finished project documentation.

**Windows Installation Instructions**

**Requirements**

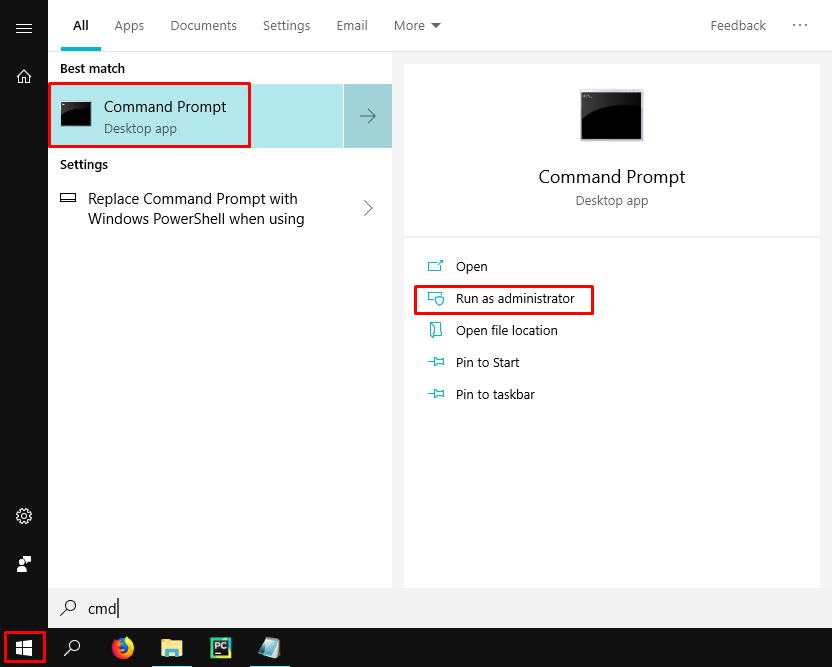
* Latest Python (project version is 3.7.2, windows installer is provided).
* PyCharm compiler (optional).

**Installing Python**

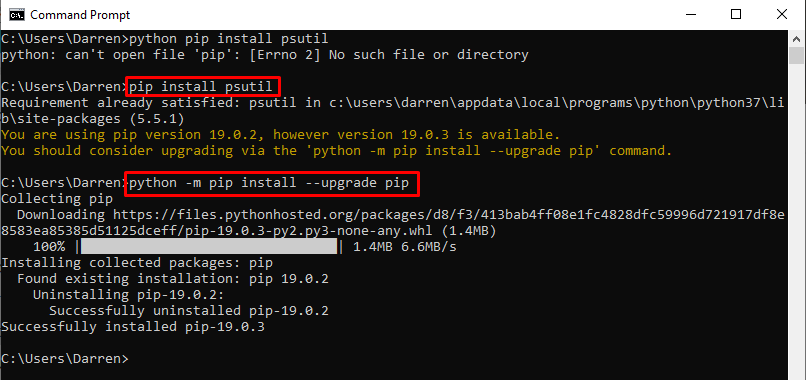
* Run the Python3.7.2 installer .exe file.
* Make sure to choose the selection **“Add Python 3.7 to PATH”** so that you can use Python/PIP commands in Windows CMD.
* 

Figure

**Checking Python Installtion, & Installing psutil**

* After you have installed Python, you’ll want to open windows CMD. If need be, choose the option to run CMD as an administrator.
* 

Figure

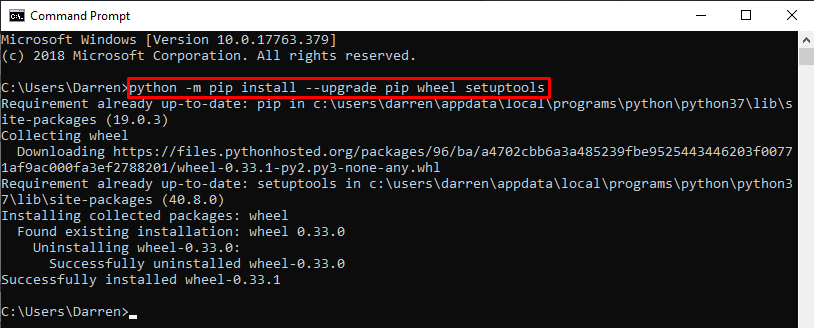
* We’ll start by making sure that Python is installed by using the below command;
  + **python --version**
* Next, we’ll install the pip utility, by using the command;
  + **pip install psutil**
* Though not necessary, if you receive the prompt to upgrade pip, you can do so using the command;
  + **python -m pip install –upgrade pip**
* 

Figure

**Installing Kivy GUI API**

* Next, run this command to install the latest pip and wheel;

**python -m pip install --upgrade pip wheel setuptools**

* 

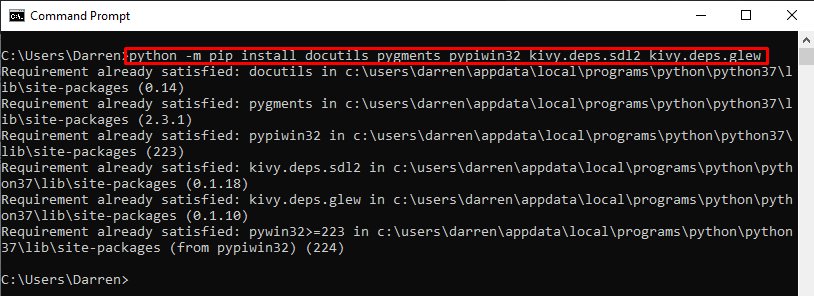
Figure

* Next, run these three commands to install the kivy dependencies;

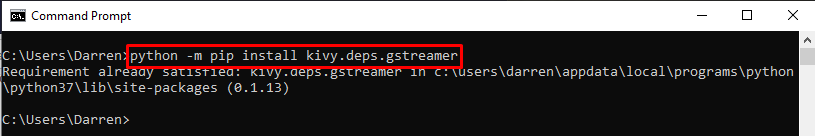
**python -m pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew**

**python -m pip install kivy.deps.gstreamer**

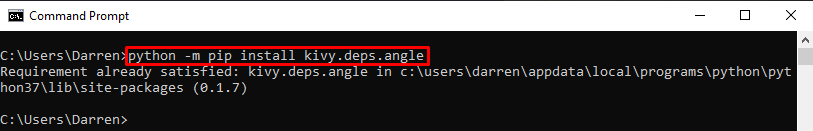
**python -m pip install kivy.deps.angle**

* ****

Figure

* ****

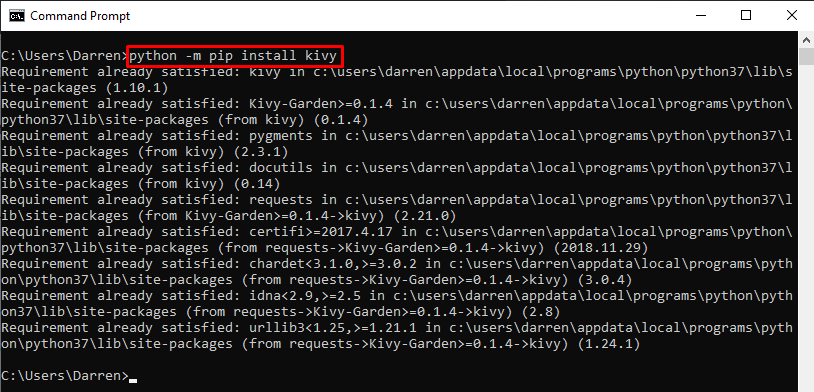
Figure

* ****

Figure

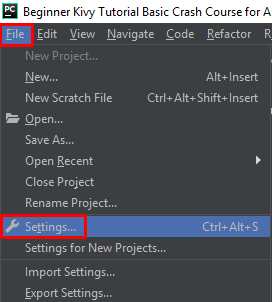
* Now, we can finally install the Kivy API itself! Use the command;

**python -m pip install kivy**

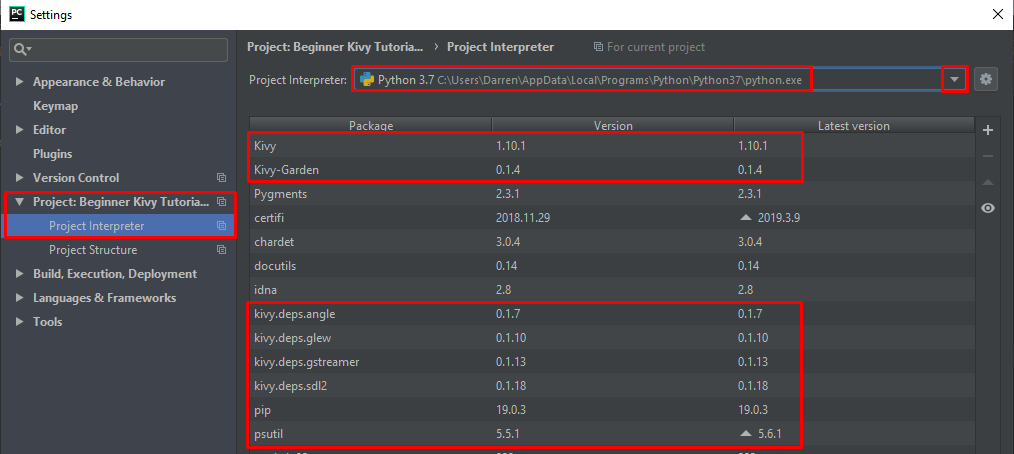
* ****

Figure

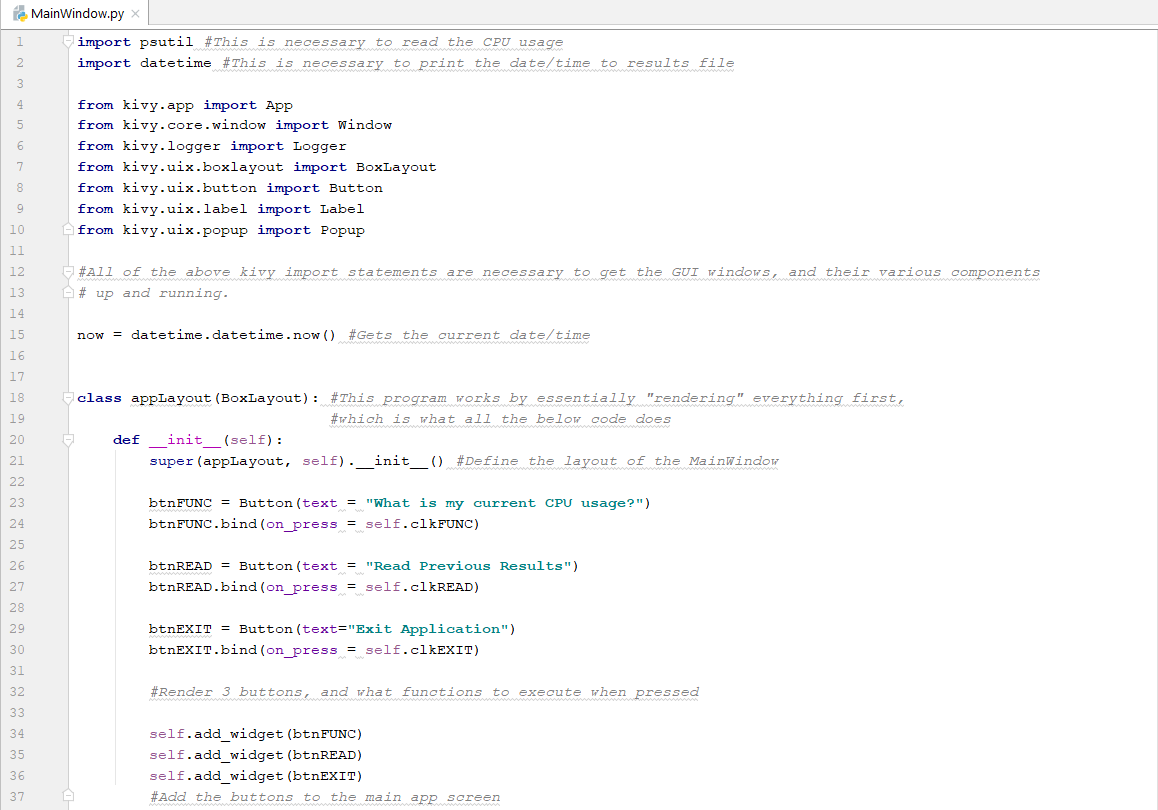
**Making sure the correct Project Interpreter is selected in PyCharm**

* With the project open in PyCharm, in the top left corner select “File”, then select “Settings”.
* 

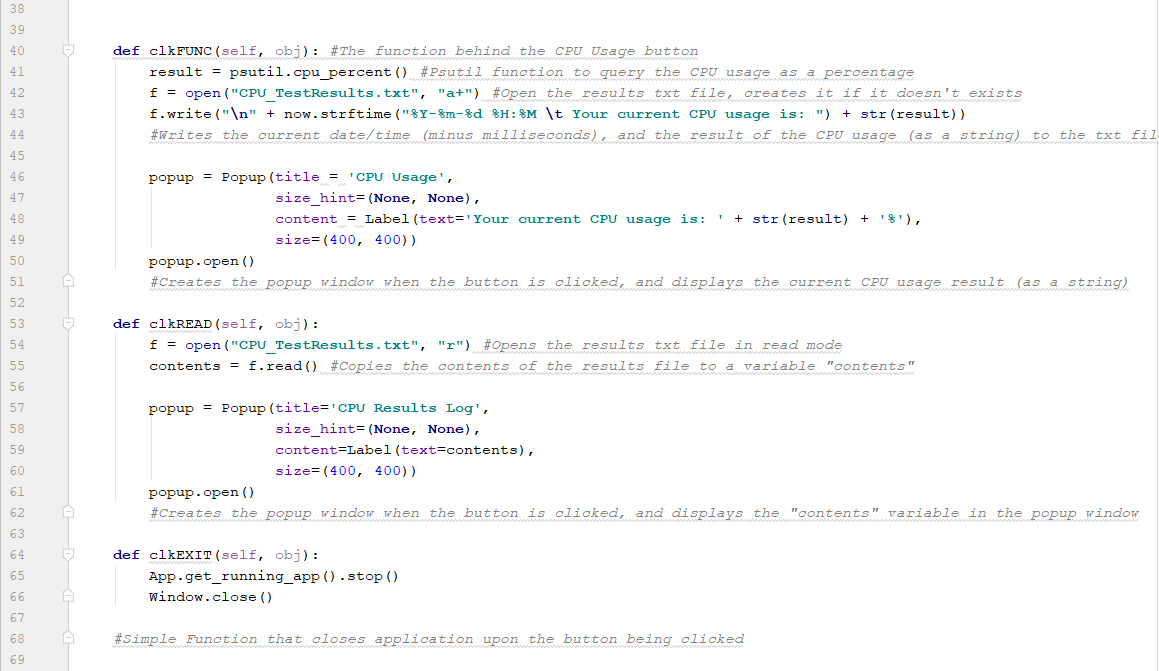
Figure

* Press the drop down arrow for “Project: xxx”, and click on “Project Interpreter”.
* In the drop down arrow, for the “Project Interpreter:”, select the Python 3.7 install. You can verify you are choosing the correct one, by checking the installed packages to see that Kivy & psutil appears.
* 

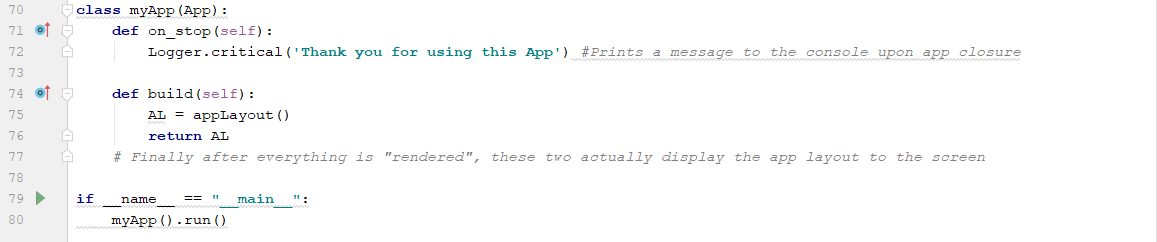
Figure

**Overview of Project Code**

Figure



Figure



Figure

**Summary of Learnings**

* Troubleshooting and debugging of code.
* Installing Python/pip on a windows platform.
* Troubleshooting of python install on windows.
* Installation of dependencies and packages for Kivy, psutil, etc.
* Converting floats to strings for plain text output
* Kivy API
  + The layout of the standard app
  + The various imports required for buttons, widgets, etc.
  + How to make the output of python functions appear in the popup windows.
  + Creating buttons, tying functions to those buttons
  + Popup windows
* Psutil
  + How to output CPU usage as a percentage, convert it to a string.
* Datetime
  + Display the current time/date, and then display the current time/date without the milliseconds (now.strftime).

**Future Work**

* Adding additional functions, such as printing network configuration settings, RAM usage, etc.
* Clean-up the overall appearance of the GUI windows/popup, custom colour scheme, etc.

**Conclusion**

I was warned as to difficulty of undertaking a GUI application for my project. It provided a numerous amount of challenges, even getting the API installed and working was difficult as instructions provided by Kivy themselves were quite unclear, and also getting everything running correctly on the windows platform was quite difficult too. Another challenge was also getting traditional python functions to work within the Kivy GUI windows and popups.

Overall though, I am quite happy with the results I achieved with my project, and was happy to have worked on something that I wouldn’t describe as a traditional use of Python.

**Guides, Instructions & Miscellaneous Info used during Project**

* Psutil Documentation: <https://psutil.readthedocs.io/en/latest/>
* Datetime example:<https://www.saltycrane.com/blog/2008/06/how-to-get-current-date-and-time-in/>
* Python read, append txt file:<https://www.guru99.com/reading-and-writing-files-in-python.html#2>
* Kivy:
  + https://kivy.org/doc/stable/guide/basic.html
  + <https://pythontic.com/app/kivy/popup>
  + <https://kivy.org/doc/stable/api-kivy.uix.popup.html>
  + <https://www.youtube.com/watch?v=1d3eRCdyNHQ>
  + <https://www.youtube.com/watch?v=RJigyKq9D4M>

# References

Kivy. (n.d.). *Kivy - Home*. Retrieved from https://kivy.org/#home

Python Software Foundation. (n.d.). *psutil 5.5.1*. Retrieved from Python Software Foundation.org: https://pypi.org/project/psutil/