Documentation

Ipython notebook environment

The application is built atop an Ipython notebook environment, and can be launched in a browser using command prompt.

Dependencies

The web application is written completely in python, and its functioning is contingent on the following modules being installed on your system.

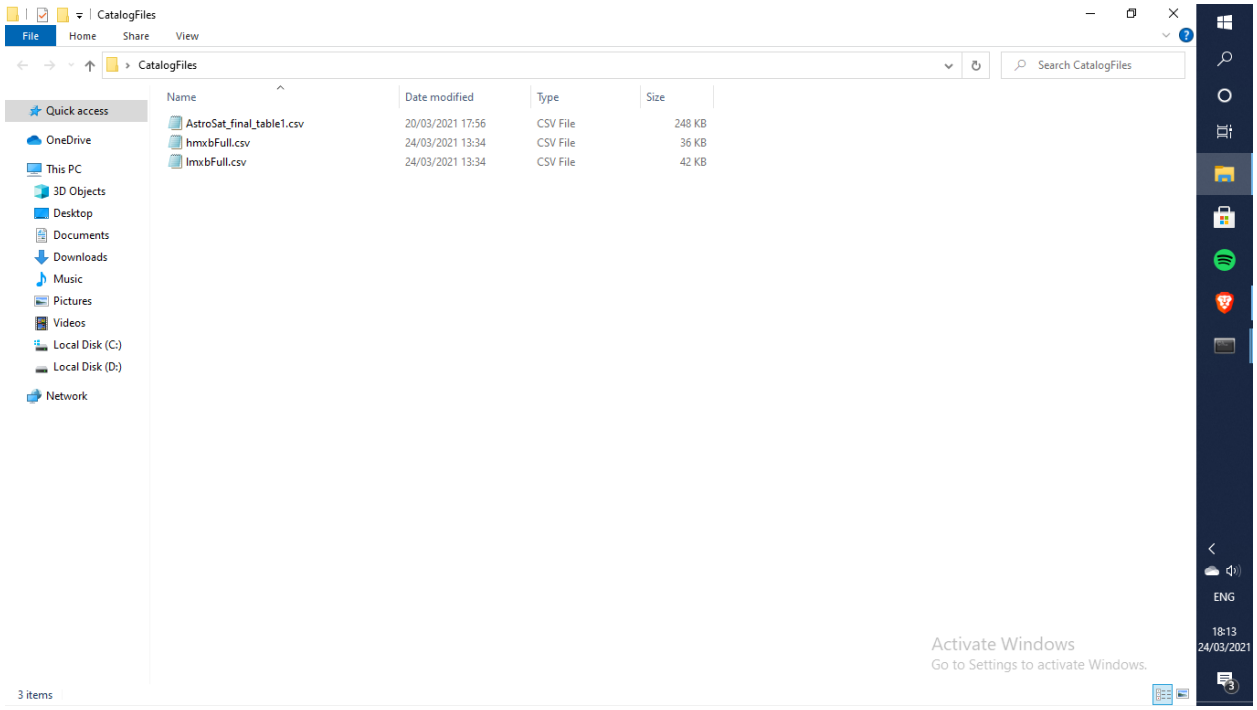
a. Ipython
b. Ipywidgets
c. Pandas
d. Astropy
e. Ipyaladin
f. Fpdf

Tutorial:

We shall demonstrate how to use the application for loading the catalogs A and B, visualising the objects in catalog A over a skymap, searching for a particular object, and finding whether it exists in catalog B.

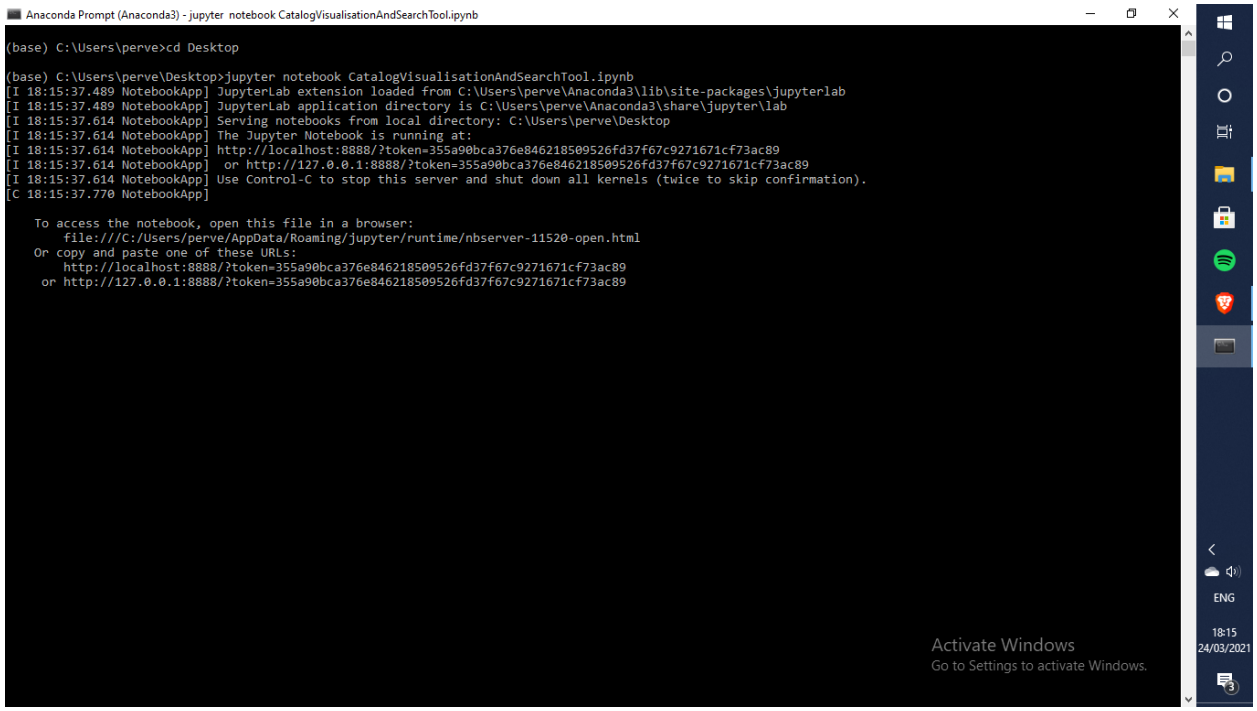
Note: The application requires some preprocessing to be performed on the files, before they are uploaded to the application. Also, simple changes in the code are required before we can add other catalogues than the one provided for the problem statement.

1. Preprocessing the files is a prior step. It simply involved linking related files by means of related entries, the process of which has been uploaded in Data preprocessing files, namely :
    - Data Preprocessing.ipynb
    - Data Preprocessing of Astrosat Observations.ipynb
    - Data Preprocessing of Astrosat Publications.ipynb
2. As a result of the preprocessing, we end up with the final files, which simply have an added column indicating the references to the objects mentioned in an entry of the catalogues. We have two catalogues A (hmxb and lmxb respectively) and catalog B (astrosat catalogue).

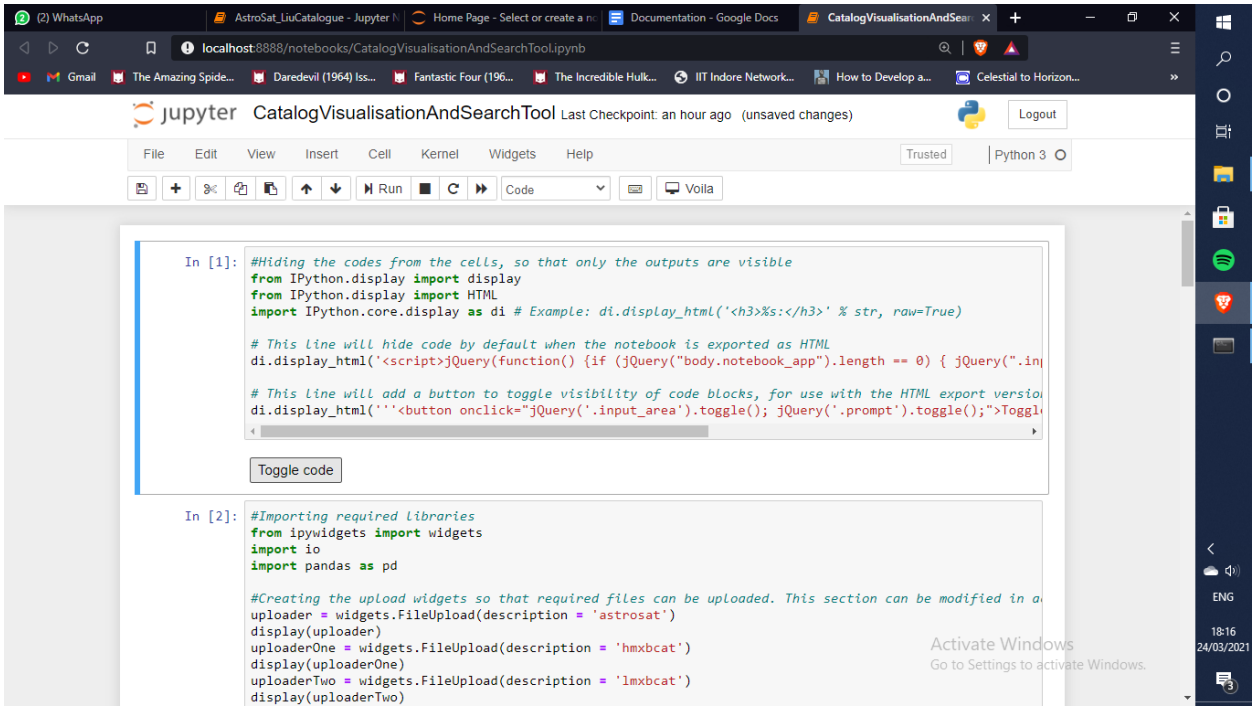


You need to have these files saved somewhere on your system so that you can upload them to the application.

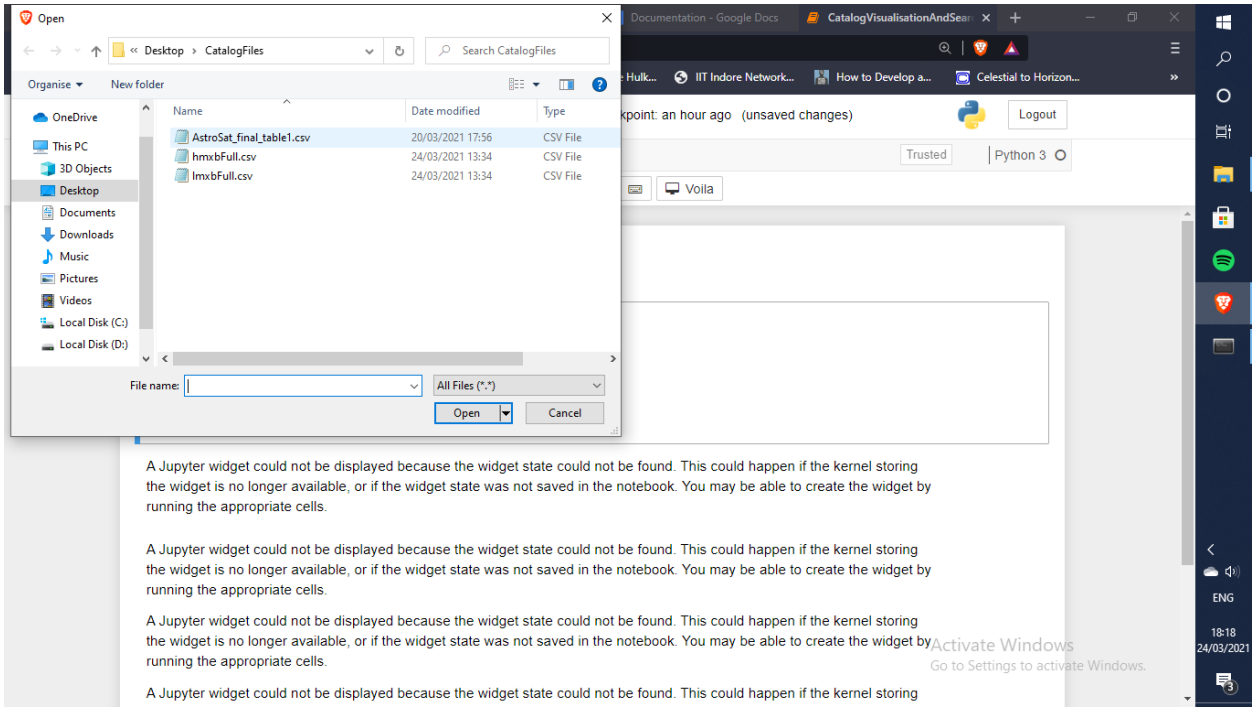
3. We now launch our application using the command prompt. Navigate to the directory where you have saved the .ipynb file and launch it.

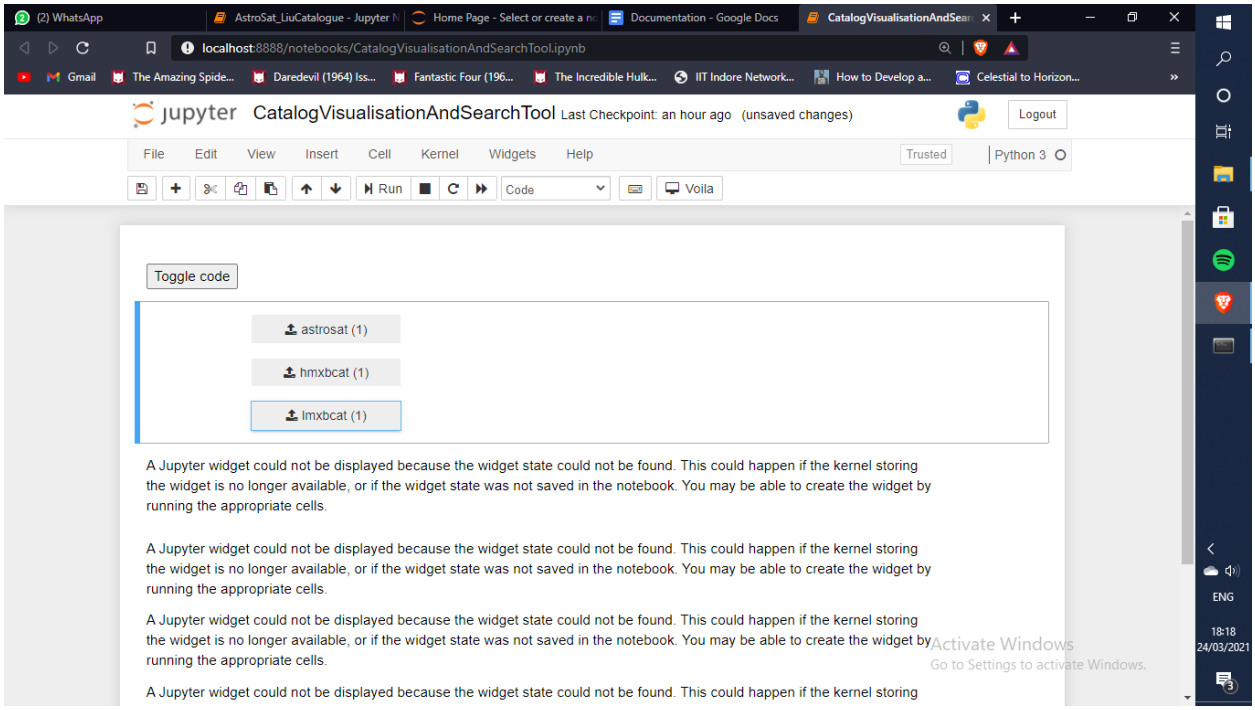


4. Hit ‘Toggle code’ to hide the code section.

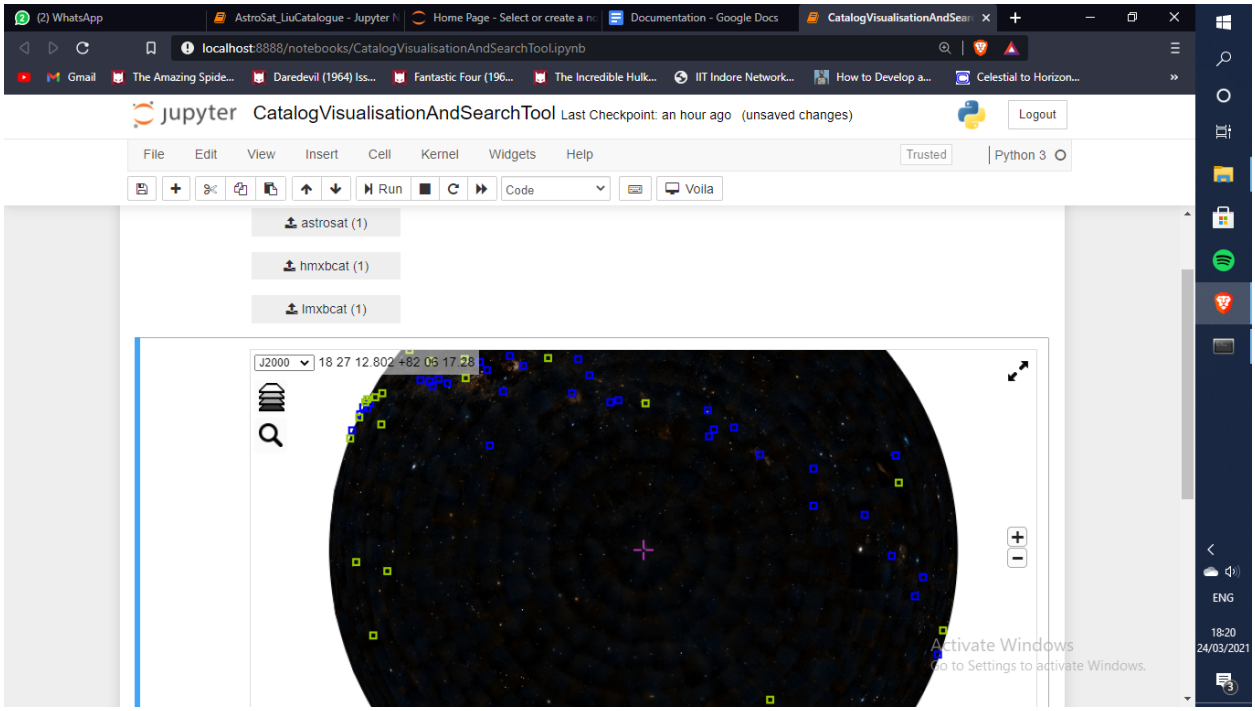


5. Press Ctrl+Enter to execute the first cell. It would ask you to upload the files under mentioned headings, add them there.

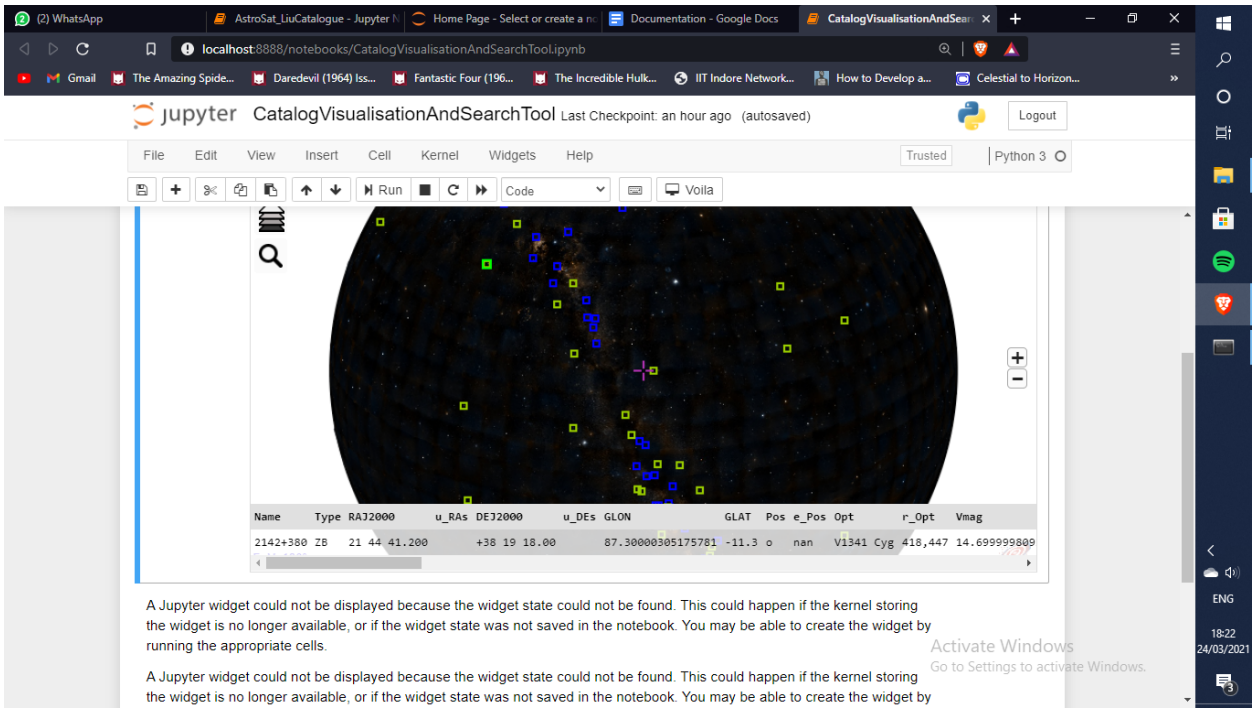




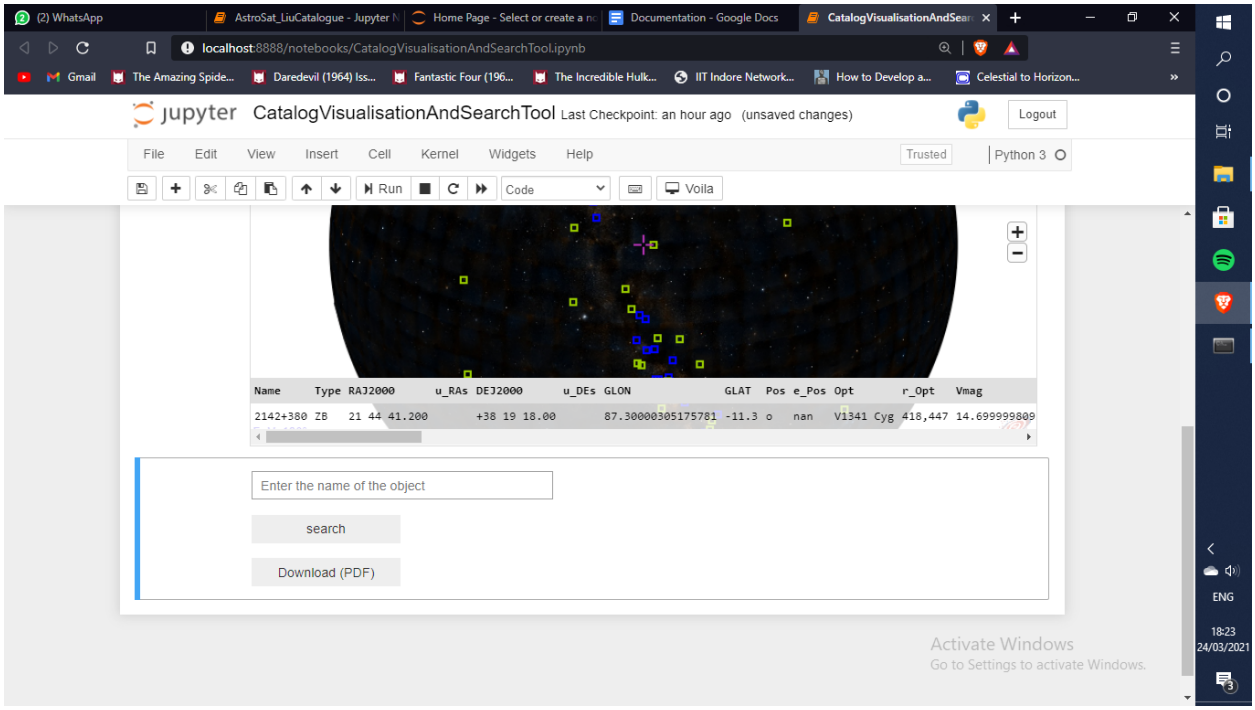
6. After adding all three files at proper places, run the next cell. This would take some time (~ 30 seconds) as the program calculates the correlations between added files and generates a list of objects to be displayed on aladin.



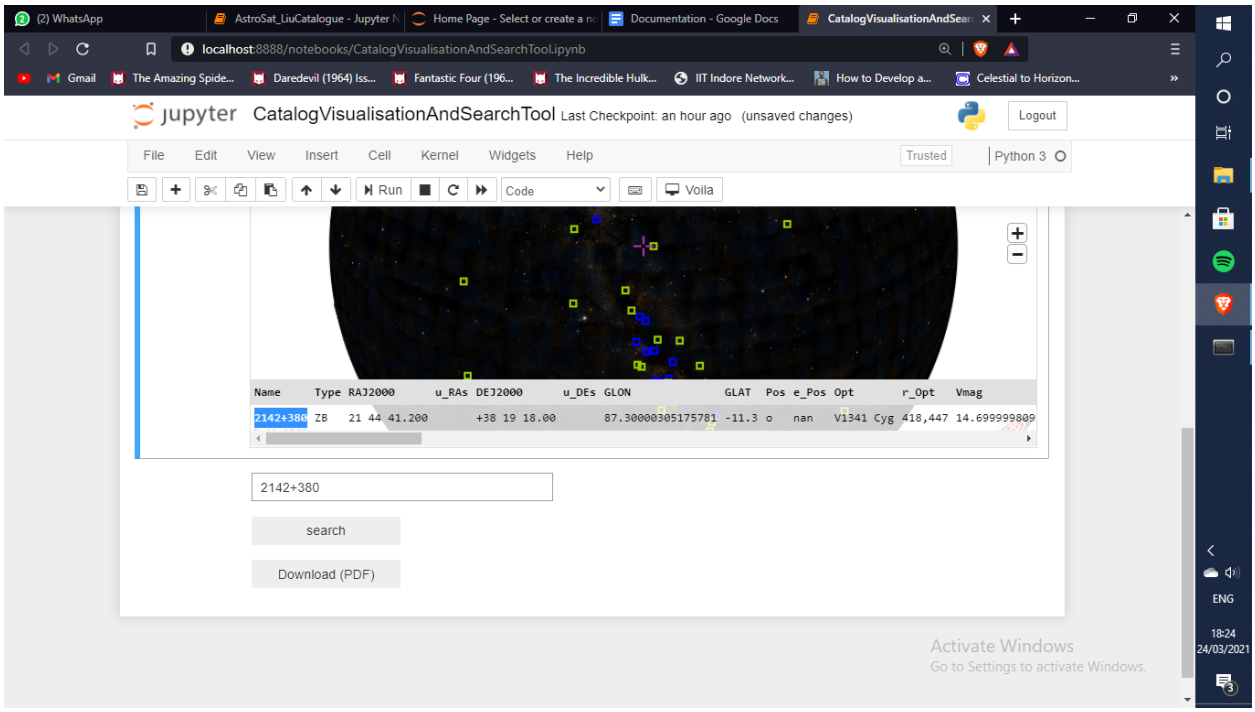
7. Zoom out and pan through to see the objects added from catalog A. The ones in blue are the objects from the hmxb catalog, while the ones in green are the objects from the lmxb catalog.
8. Select one of the objects. The pertinent related information from catalog A is displayed in a table at the bottom of the skymap.

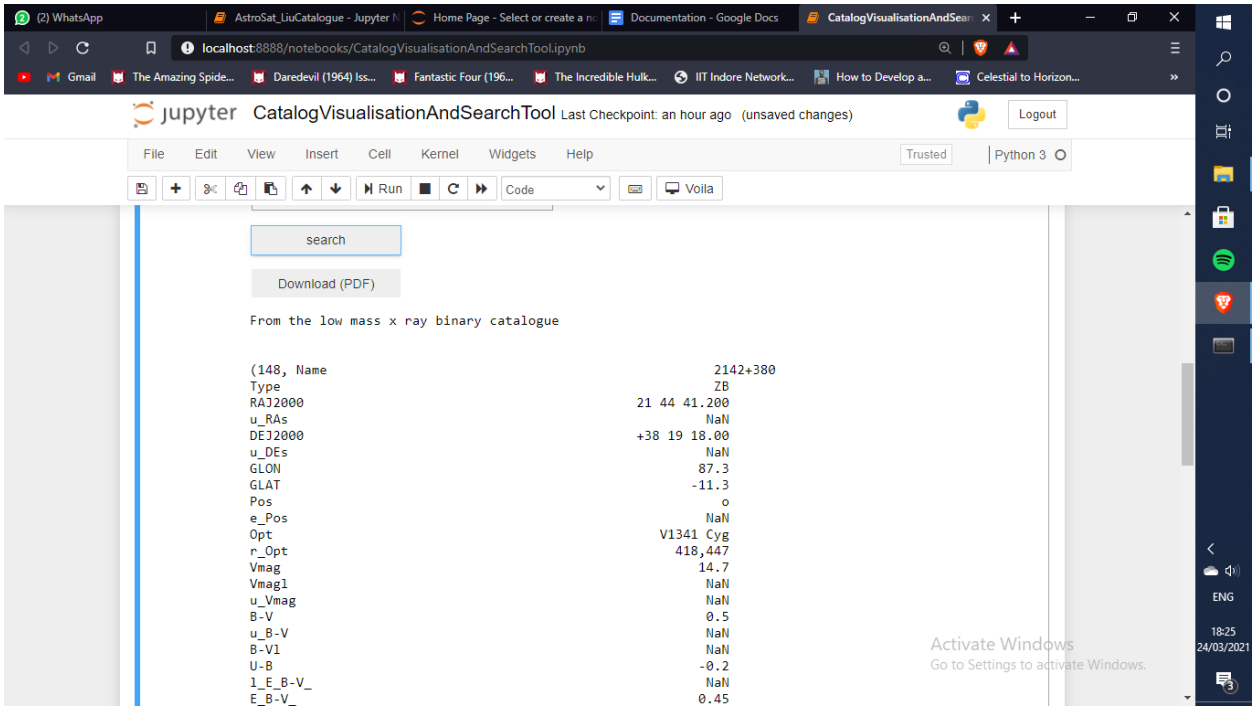


9. If you want to search for the details and obtain them in a printable format. Execute the next cell. You'll end up with a search bar.

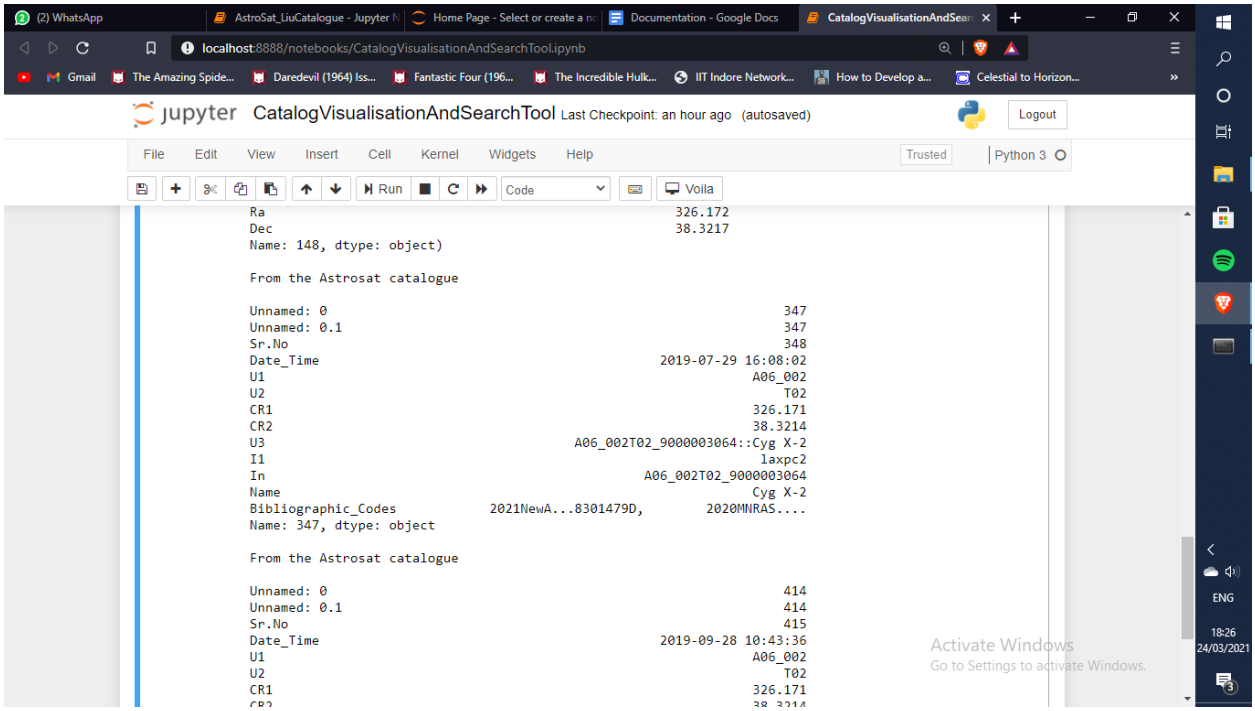


10. Copy the name of the object. In this case, '`2142+380`' and paste it into the search bar. Press the 'search' button.

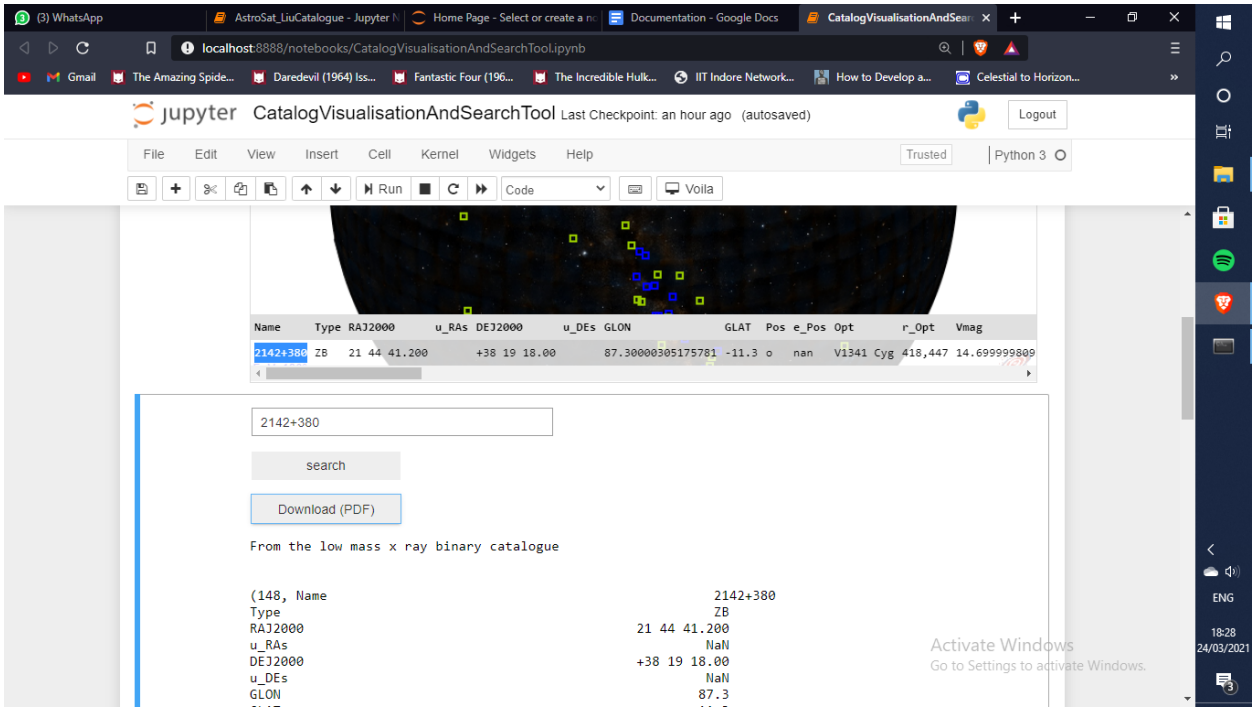


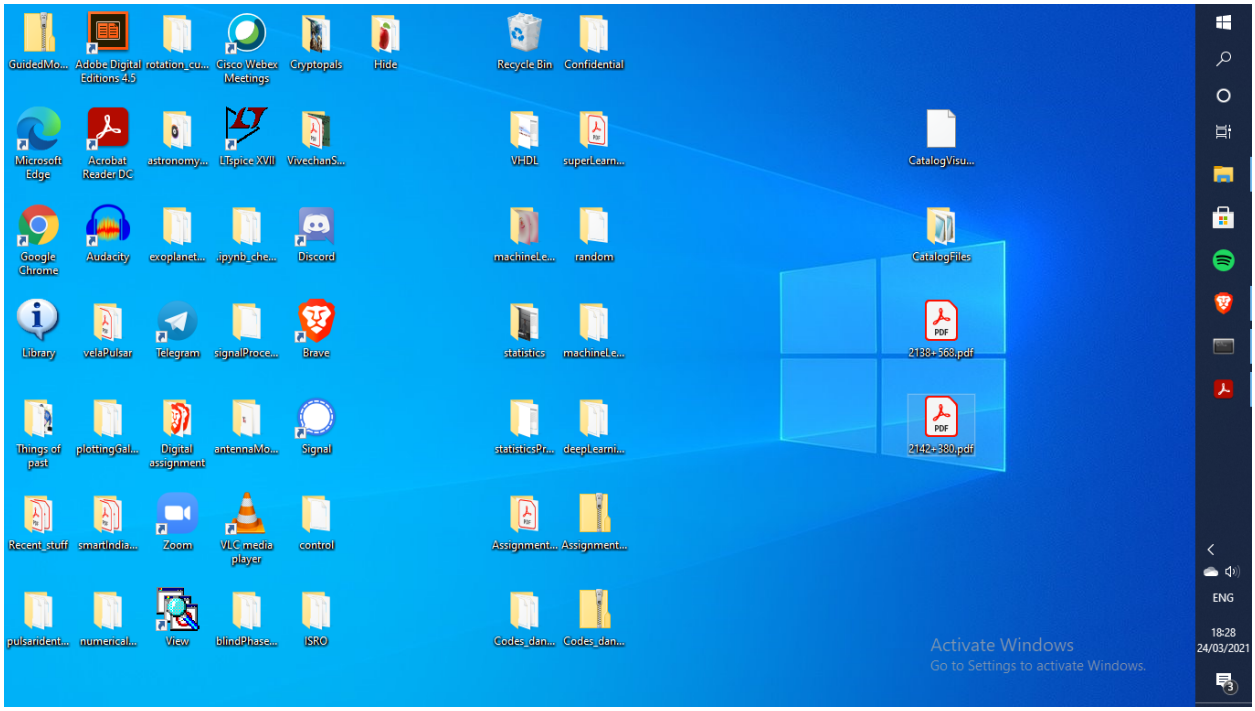


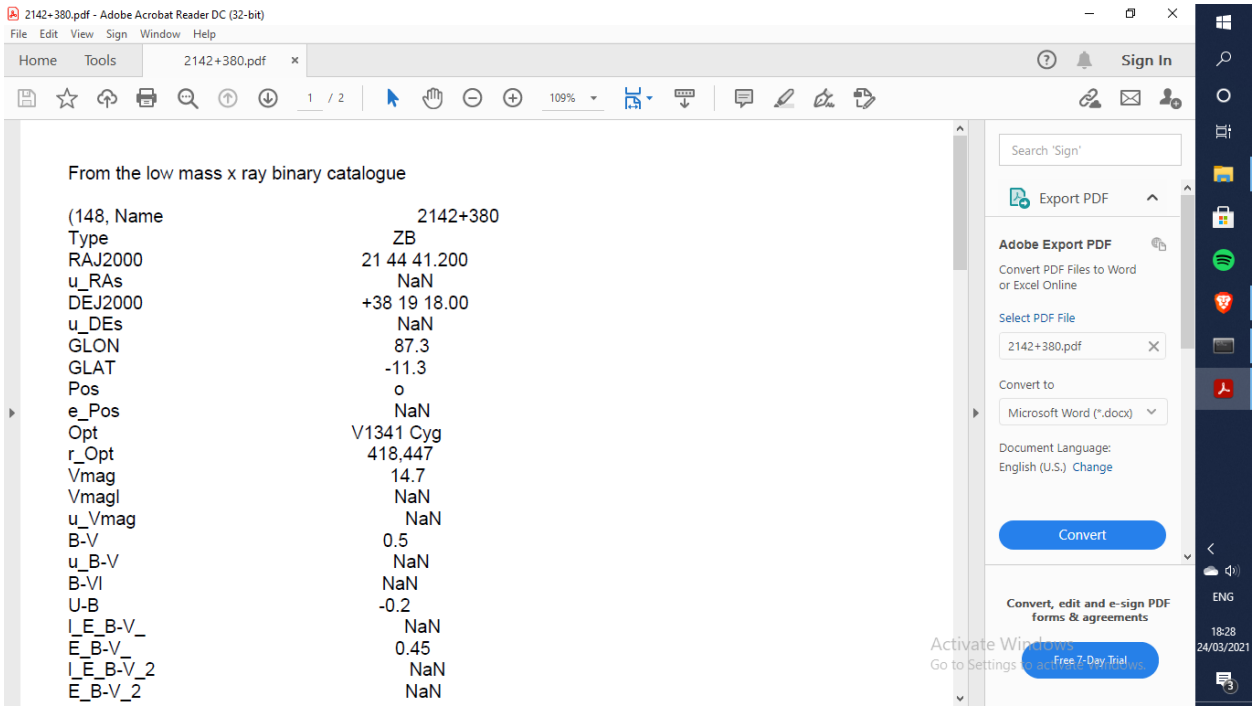
11. You'll end up with a list of the details of the object in question, such as its name, its coordinates in different systems, its photometric data, alternative names, and orbital and pulse periods. These informations are dependent on the catalog one uploads, and aren't calculated by the application.
12. In the event that the object was present in the Astrosat catalog (catalog B) as well, you'll see related details from the Astrosat catalog mentioned as well.



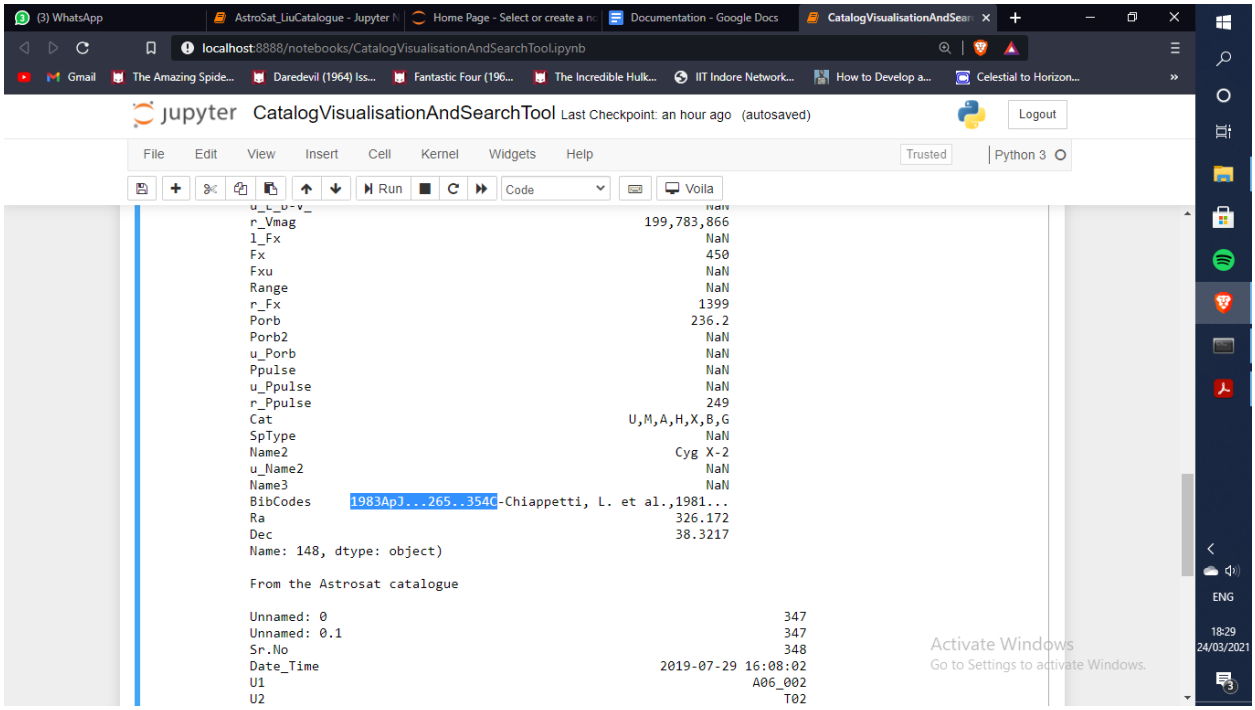
13. Click on ‘Download(PDF)’ and you end up with a file on your desktop, thus saving the above details for future reference.

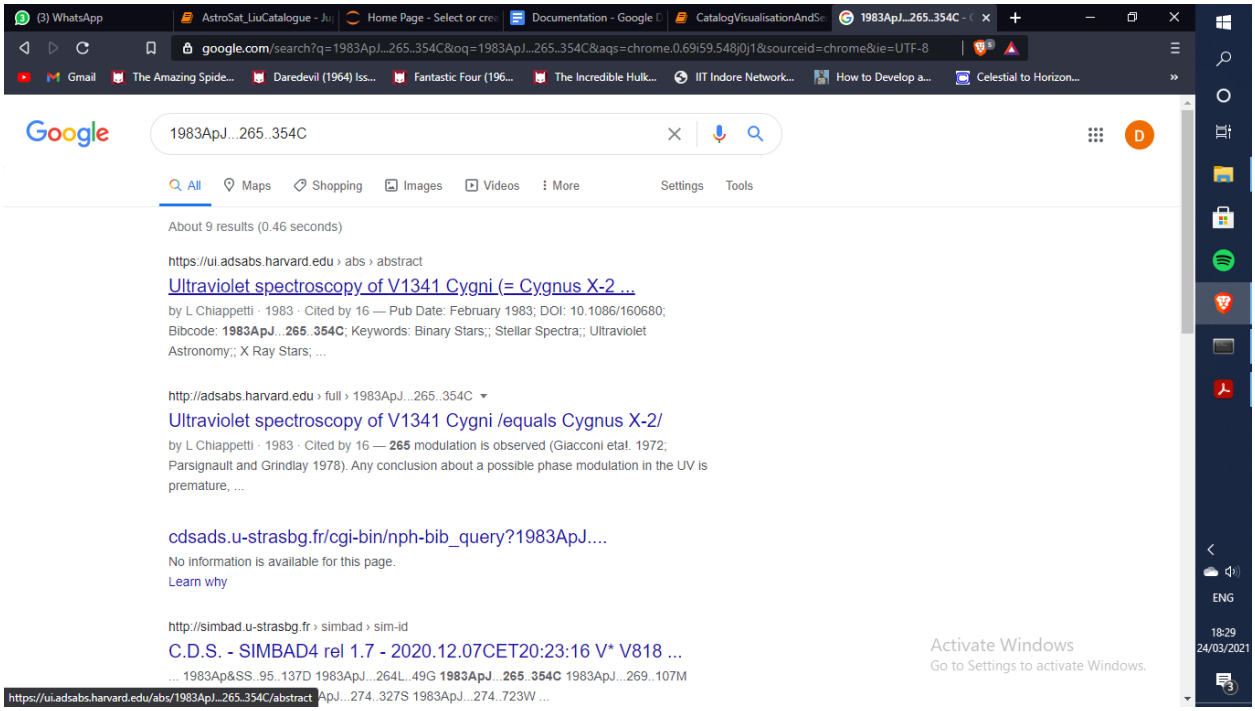




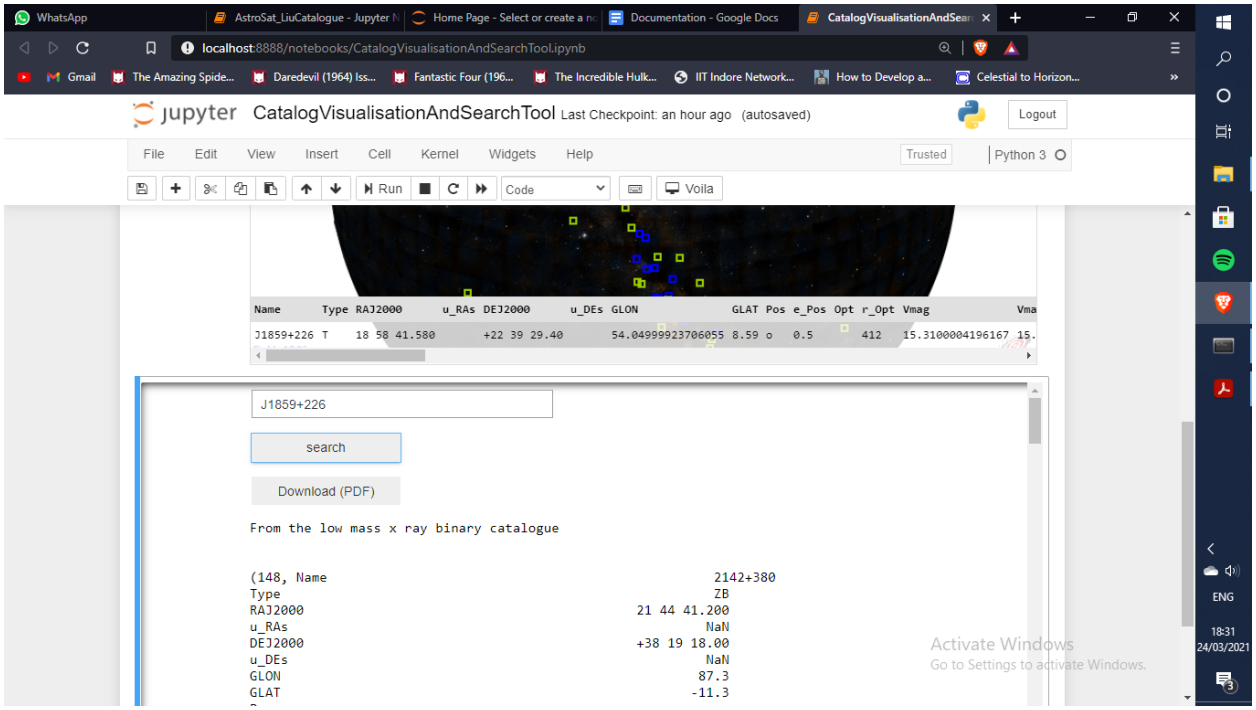


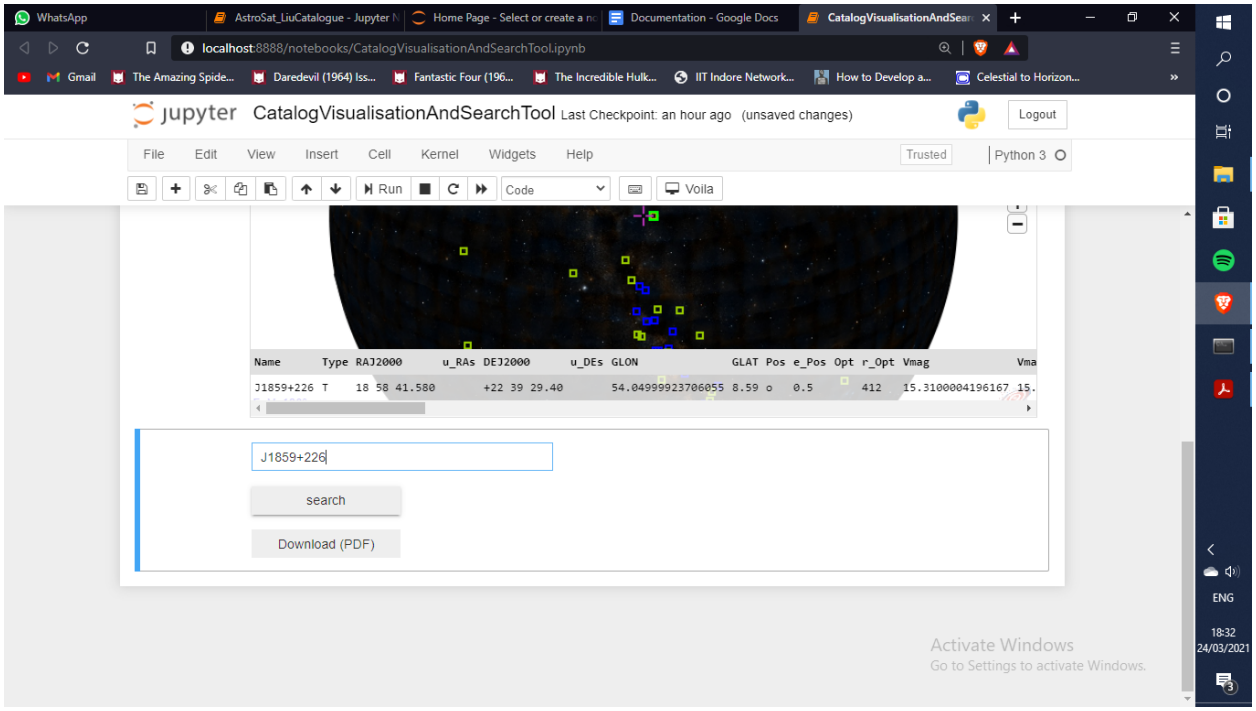
14. Let's see how this data could be used. We can search for the relevant papers using the bib-codes.

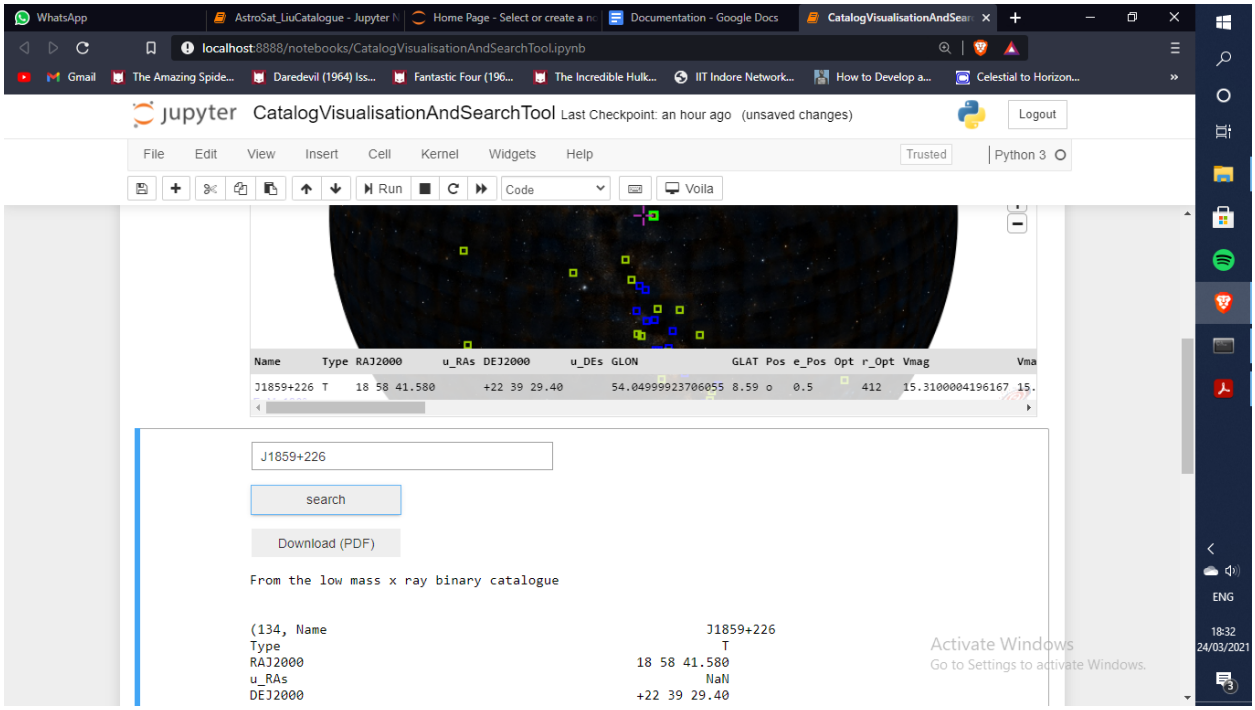




15. Now, if you want to search for the details of another object. You must rerun the last cell (which is a caveat and mentioned appropriately under it). Simply copying and pasting the names of the object, won't suffice.


localhost:8888/notebooks/CatalogVisualisationAndSearchTool.ipynb
jupyter CatalogVisualisationAndSearchTool Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Voila
Name Type RAJ2000 u_RAs DEJ2000 u_DEs GLON GLAT Pos e_Pos Opt r_Opt Vmag Vma
J1859+226 T 18 58 41.580 +22 39 29.40 54.04999923706055 8.59 o 0.5 412 15.3100004196167 15.
J1859+226
search
Download (PDF)
From the low mass x ray binary catalogue
(148, Name 2142+380
Type ZB
RAJ2000 21 44 41.200
u_RAs NaN
DEJ2000 +38 19 18.00
u_DEs NaN
GLON 87.3
GLAT -11.3
Activate Windows
Go to Settings to activate Windows.
ENG
18:31
24/03/2021


localhost:8888/notebooks/CatalogVisualisationAndSearchTool.ipynb
jupyter CatalogVisualisationAndSearchTool Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Voila
Name Type RAJ2000 u_RAs DEJ2000 u_DEs GLON GLAT Pos e_Pos Opt r_Opt Vmag Vma
J1859+226 T 18 58 41.580 +22 39 29.40 54.04999923706055 8.59 o 0.5 412 15.3100004196167 15.
J1859+226
search
Download (PDF)
Activate Windows
Go to Settings to activate Windows.
ENG
18:32
24/03/2021



Caveats

1. The code needs to have some minor adaptations before another catalogue could be added and visualised. This includes changing the number of upload buttons and the code for comparing the data between different catalogues (which requires you to have done some preprocessing if required as well). The changes are minor though, and different kinds of catalogues can be incorporated, compared and visualised.
2. The search bar needs to be run every time you want to search for the details of a new object from the catalog. Otherwise, the details of the old object shall remain unchanged, even when a new object name is added and the search button is pressed.

Future improvements

1. Higher flexibility as far as the addition of catalogues is concerned.
2. Eliminating the need to rerun the ‘search cell’ each time one requires to search for a new object from the same catalog.