**FRAMEWORK:**

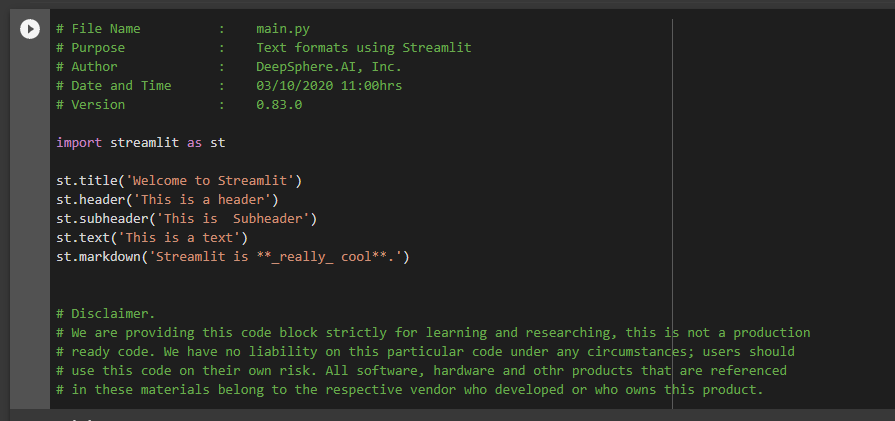
Practically a Framework is a platform to develop an application which fulfills the routine tasks that every application requires and to make it easy for the developers. Framework just gives you basic structure around which you will write your code to have the greater functionality of the system.

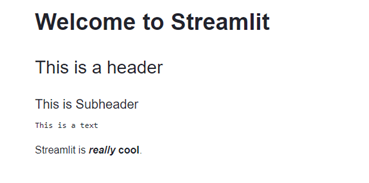
**STREAMLIT:**

Streamlit is a Framework for Machine Learning, Data Science and it is open source, most of the Developers use python for ML and Data science and one of the libraries in python used for this technology is Streamlit. Beautiful UIs can easily be designed through numerous components from the library.

**EXERCISE 1:** Text formats

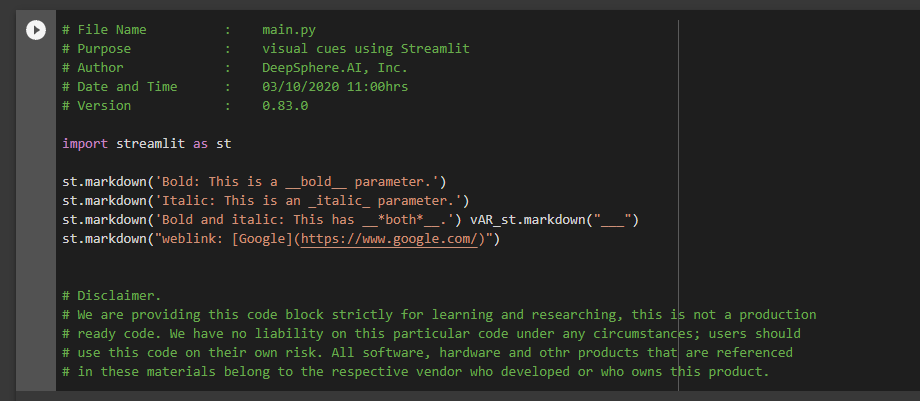
Streamlit apps usually start with a call to st.title to set the app’s title. After that, there are 2 heading levels you can use: st.header and st.subheader. Pure text is entered with st.text, and Markdown with st.markdown.

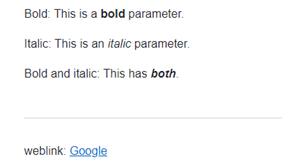
****

****

**EXERCISE 2:** Visual Cues

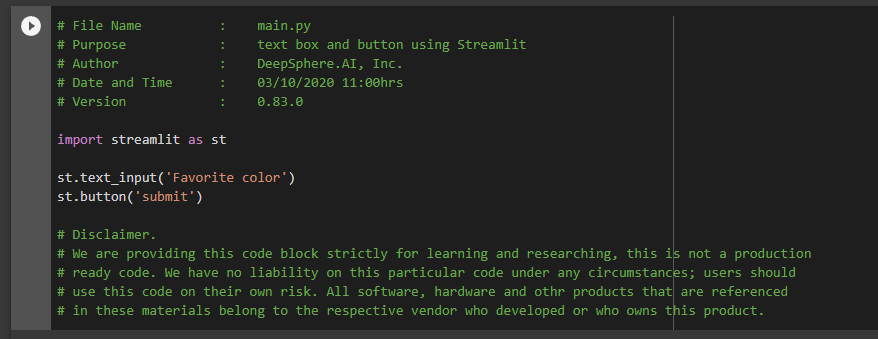
We can give some Visual cues like Bold, Italic or both combined, we can also have weblinks using st.markdown.

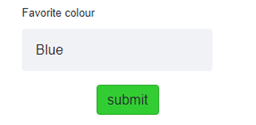
****

****

**EXERCISE 3:** Text Box

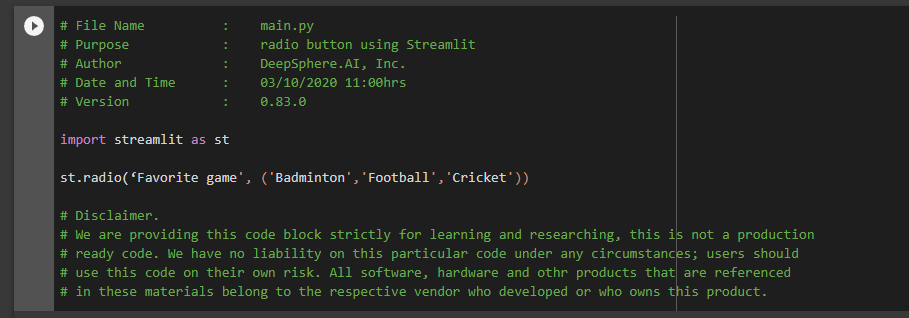
This is a function which takes the user input and we can also add a submit button as mentioned below using st.text\_input and st.button.

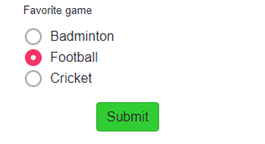
****

****

**EXERCISE 4:** Radio button

We can add a radio button which allows the user to select one option from a set using st.radio.

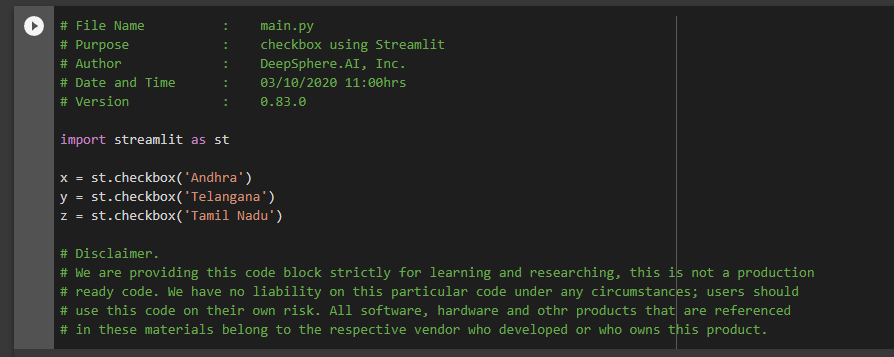
****

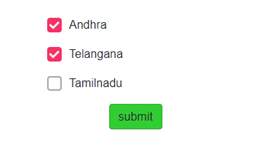
****

**EXERCISE 5:** Radio button

We can add Check boxes using st.checkbox when more than one option

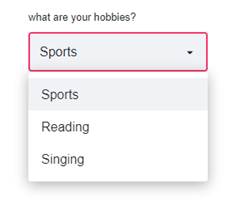
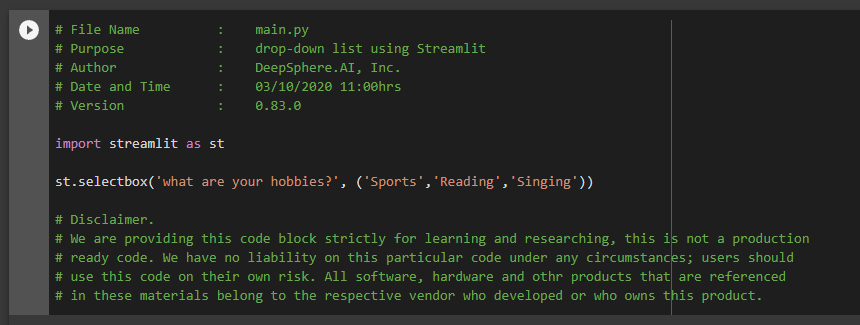
may need to be checked.

****

****

**EXERCISE 6:** Drop-down list

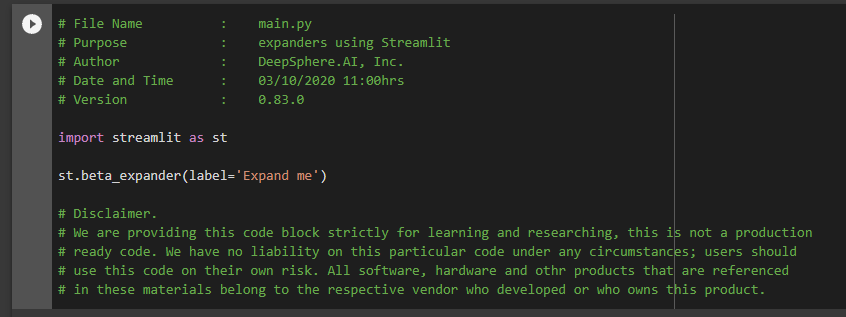
We can add a drop down menu using st.selectbox. The advantage of this is that they conserve screen space.

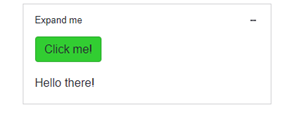
****

**EXERCISE 7:** Expanders

An expander is a function that allows you to show or hide the content when it is clicked, same as in a dropdown menu.

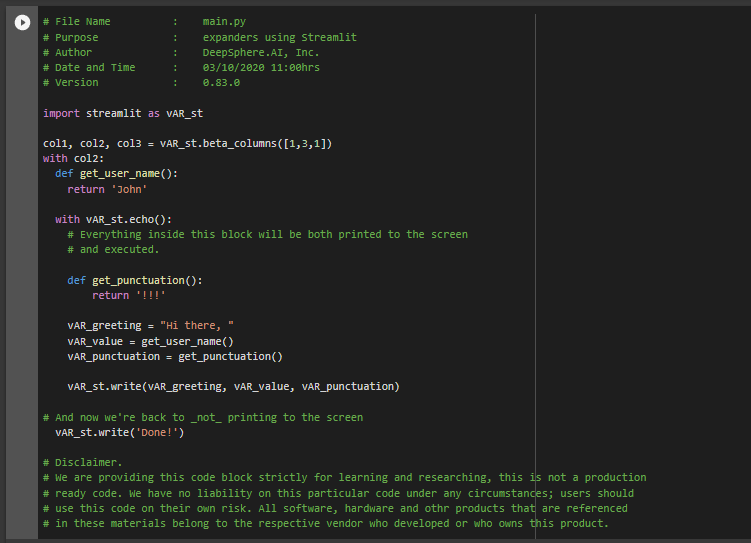
We can also add a button inside the expander. And also can give some commands to perform if the button is clicked.

****

****

**EXERCISE 8:** Displaying code

In streamlit with the help of st.echo() we can make the middle section of the code visible.Whatever code is present inside this function will be visible on the screen, and it will also be executed.

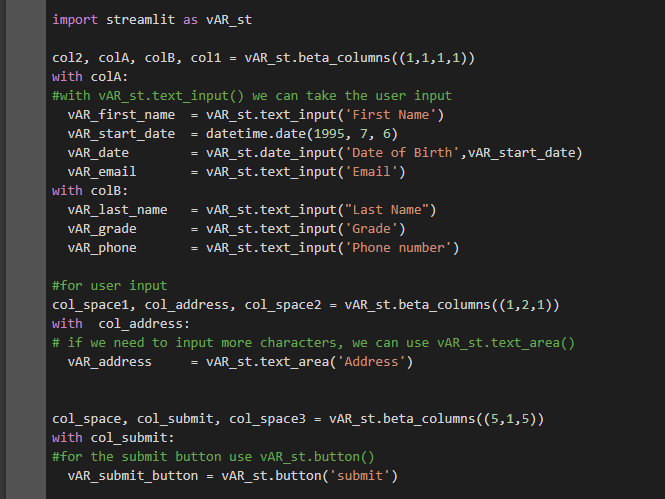
****

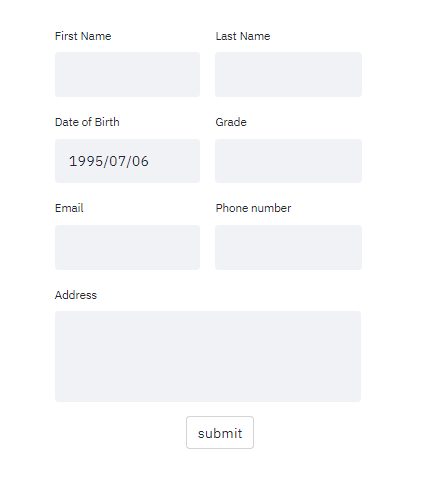
****

**EXERCISE 9:** Input Form

So, we can create a simple Input Form from st.text\_input()

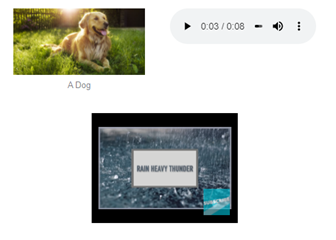
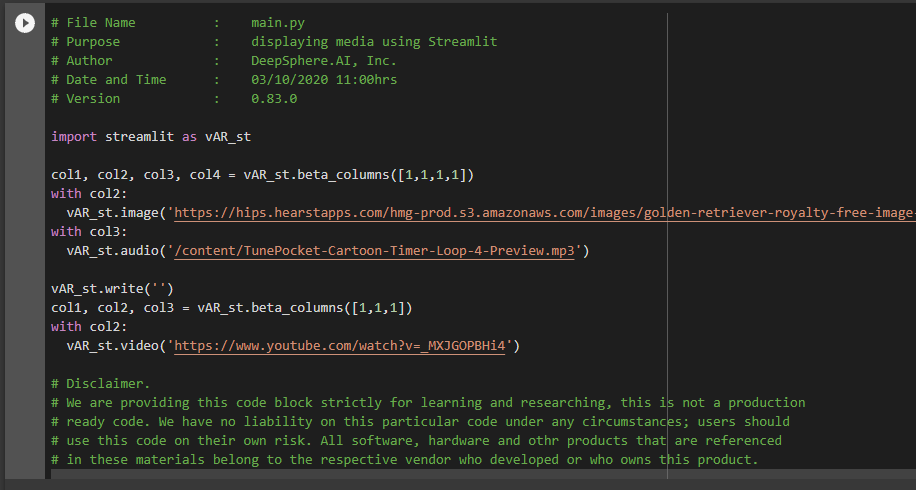
we can use st.input\_area() if we want to have the user input in the next lines. And we can use st.beta\_columns(), so that we can have the text inputs side-by-side.

****

****

**EXERCISE 10:** Display media

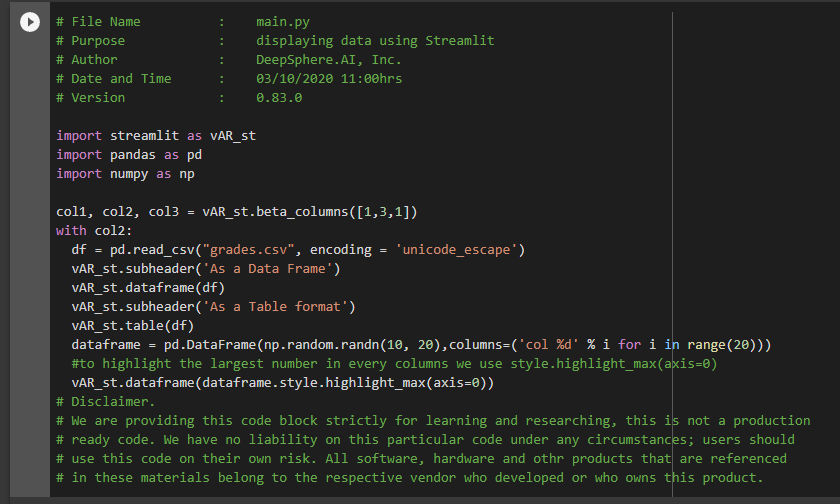
We can display images, audio and video files by simple functions like st.image(), st.audio(), st.video()respectively. We can also set a caption and width to the image.

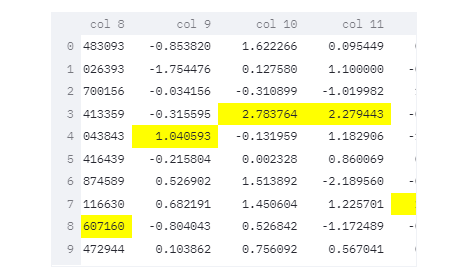
****

**EXERCISE 11:** Display data

We can display a data frame as an interactive Table format, files like .dox or .csv can be displayed.

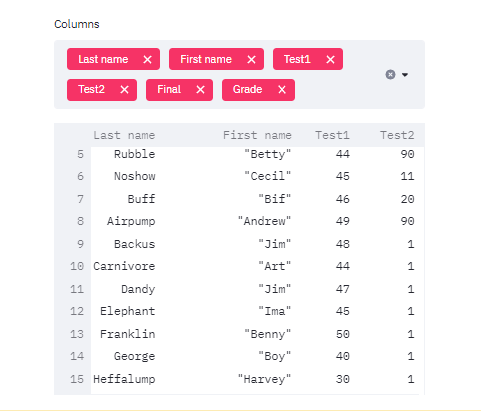
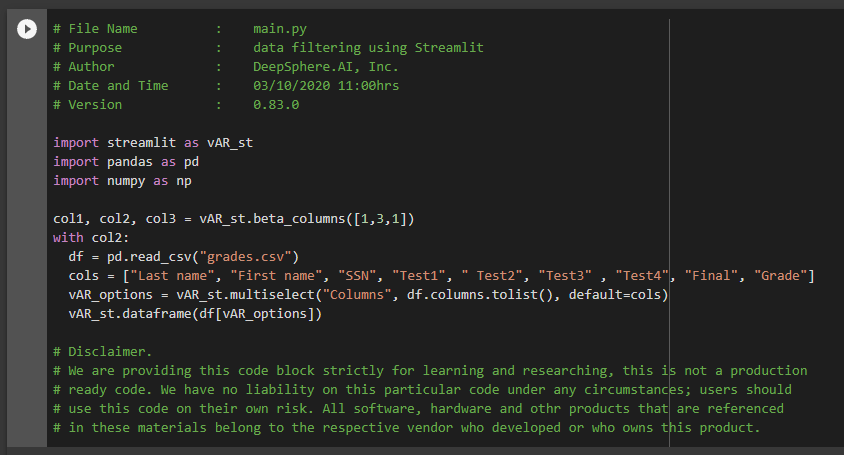
And we can also highlight the highest or the lowest number in the row or in the column df = pd.read\_csv(fifle\_name), st.dataframe(df).

****

****

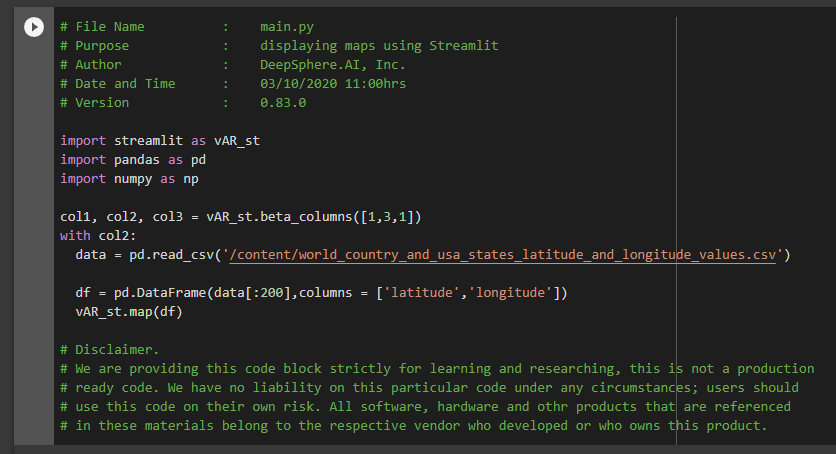
**EXERCISE 12:** Data filtering

We can filter data in the table by st.multiselect().

****

**EXERCISE 13:** Displaying maps

st.map is a function which displays the map when we upload a data frame containing the information about longitudes and latitudes.

****

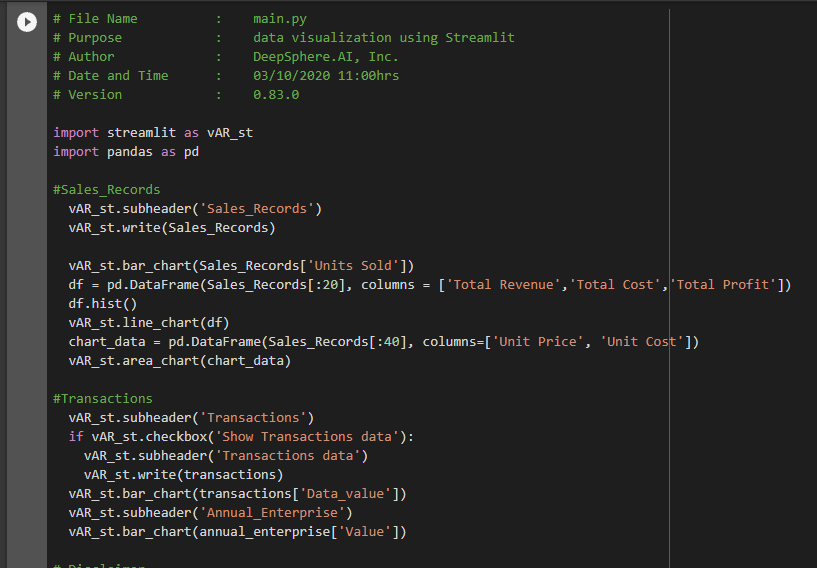
**EXERCISE 14:** Data visualization

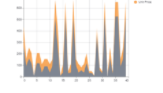
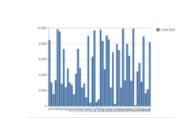
Data visualization is a technique for creating images, diagrams to communicate a message. Visualizations are more appealing and easy to understand. There are some Standard charts like

for line charts - st.line\_chart,

for bar chart - st.bar\_chart,

for area chart - st.area\_chart.

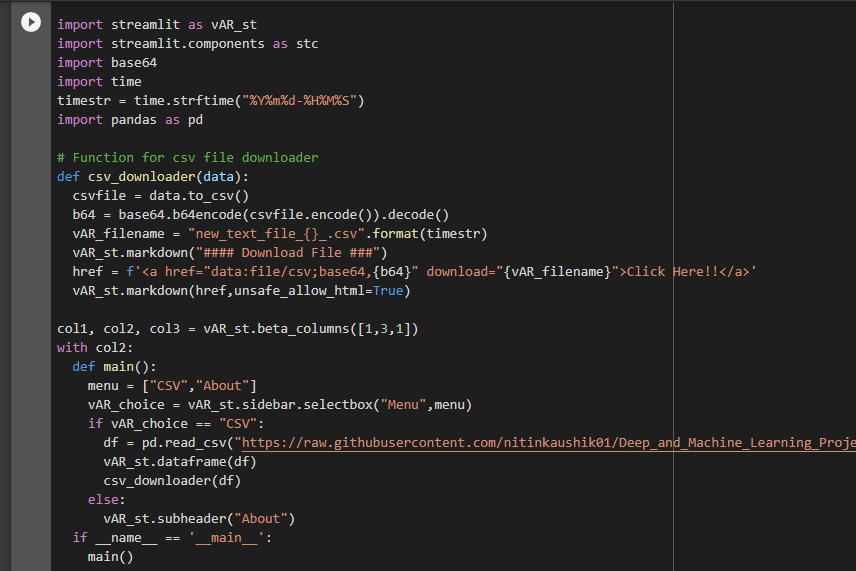
****

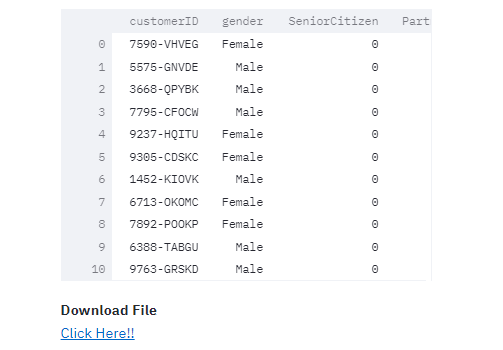
****

**EXERCISE 15:** Downloadable data

With the help of Base64, which takes any form of data stored in binary format across the channels and transforms it into a long string of plain text.

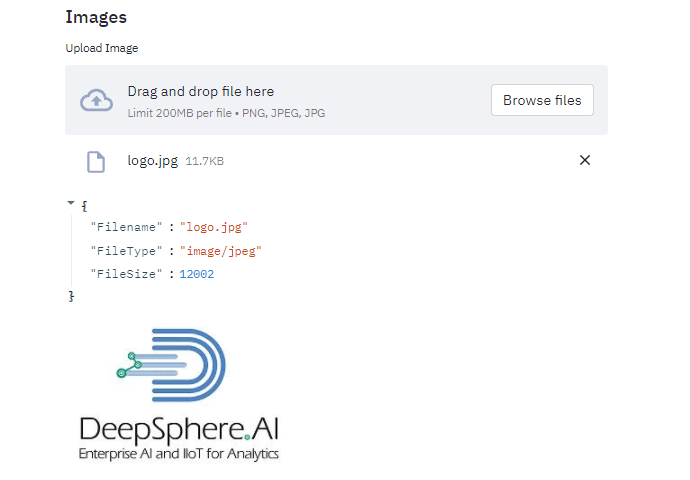
B64encode() can encode the string into the binary form. And finally the decode() function returns a value that is the same datatype as the first result in the list.

****

****

**EXERCISE 16:** Uploading files

Upload files to streamlit by the function st.file\_uploader()

****

**EXERCISE 17:** Cache

Cache memory is important because it improves the efficiency of data retrieval. It stores program instructions and data that are used repeatedly. It collects temporary data to help websites, browsers, and apps load faster. Streamlit also provides a function called st.cache() which does pretty similar work. It checks a few things like:

1. The input parameters that you called the function with

2. The value of any external variable used in the function

3. The body of the function

4. The body of any function used inside the cached function

When we run the App, if streamlit has seen these 4 components for the first time, It runs and stores the result in the local cache. And when we re-run the app, if none of the values, body of the functions, input parameters haven't changed then the cached function is called. None of the components are changed, so streamlit skips the executions and returns the output previously stored in the cache.