Task 1 – Multi-Channel Signal Viewer

Introduction: Monitoring the vital signals is a crucial aim in any ICU room.

Description: Develop a **web application** that illustrates multi-channel signal viewer that has the following features:
- The user can browse his PC to open any signal file. Each group will need to provide samples from three different medical signals (e.g. ECG, EMG, EEG,…etc). Each signal type should have an example for normal signal and abnormal signal.
- Your application should contain two main identical graphs. The user can open different signals in each graph. i.e. each graph has to have its own controls. The user can run each graph independently or link both graphs via a button in the UI. When the graphs are linked, the two graphs must display the same time frames, signal speed, and same viewport if zoomed or panned (i.e. if the user zoom on one graph, the other graph should apply the same exact zoom as well). If the link of the two graphs is disabled, then each graph can run its signals independently.
- In any of the two graphs, when the user opens a signal file, the signal should show up in the cine mode (i.e. a running signal through time, similar to the one you see in the ICU monitors). Do NOT open a signal in a static mode. If the signal ends, there should be a rewind option to either stop the signal or start running it again from the beginning.
- The use can manipulate the running signals through UI elements that provide the below function:
  - Change color,
  - Add a label/title for each signal,
  - Show/hide,
  - Control/customize the cine speed,
  - Zoom in/out,
  - Pause/play/rewind(on/off),
  - Scroll/Pan the signal in any direction (left, top, right, bottom). Scroll is performed through sliders, and pan is performed through the mouse movements.

  During these manipulations, you need to take care of the boundary conditions! Intuitively, no scroll/pan should be allowed before your signal starts or after it ends or above its maximum values or below its minimum values. No user expects to see an empty graph coz he scrolled the signal too much to the top for example. Note: Ofcourse, all manipulations will be applied on all the opened signals (viewed or hidden).
- Exporting & Reporting: For the sake of reporting, the user can export the current status of the graph along with some data statistics on the displayed signals to a pdf file. You need to generate the pdf contents via the code. i.e. Do NOT take a snapshot image and convert it to a pdf file!
  - Data statistics can be mean, std, duration, min and max values for each signal. These numbers should show up in a nice table in the pdf file. The table can have the signals in different rows and the values in different columns.

Code practice:
- Use proper variable and function names. If I do not understand what your variable is roughly doing without asking for your explanation, then this is NOT a proper name! Examples for non-proper names: x, y, counter, ss, ii, s_i,…etc. Each non-proper variable or function name will be penalized with -10% of the whole task grade.

General Notes:
- This is a task for an engineer who has had reasonable experience with software programs. And thus, s/he is expected to provide convenient, user-friendly software. Do NOT invent a feature or a user-interaction that you had never seen before in another software. Do NOT INVENT but rather IMMITAE what you have experienced before with software. If you feel you are very smart that your feature is completely new and no one thought about or saw it before, then we do NOT want to see it either! During delivery, you will always be asked this question "where did you see this feature before?" either for features related to signal viewers or related to dealing with the computer or software in general.
- If you are new to signal viewers, try to download a couple and experience them. There are tons of free downloadable viewers on the internet. Not seeing a viewer before is NOT an excuse for the previous note.