REQUIREMENTS

- **R.1 –** High investment in metadata quality, with the use of a standard (Dublin Core adaptation) that third-parties can easily follow as well which allows for an automation of merges and a comprehension of required hardware, data interpretation processes and other necessary conditions.

- **R.2 –** Have as much data as possible open to the public in repositories such as Git, Zenodo, Harvard Dataverse…
  Private data can exist but only if necessary.

- **R.3 –** Minimize restrictions as much as possible in order to fully embrace a " as free as possible " philosophy. For this, block only specific uses of the dataset, ensure due credit to contributors and authors and avoid licenses.

- **R.4 –** Create well documented "README" files to facilitate third-parties interactions with our data.
  This includes but is not restricted to:
  > Good description and definition of the contents of each file as well as the way they interact with others;
  > Units of measure;
  > Identification of the parameters of each table;
  > Hardware and software requirements and restrictions;
  > Specification of needed environmental conditions;
  > Data processing guide;
  > Description of associated datasets;
  > Support contacts.

- **R.5 –** Maintain a high and consistent abstraction level to ensure the dataset is flexible enough to be merged or used for the broadest amount of goals.

- **R.6 –** Implement a program structure with future mergers and expansions from submissions in mind, following a crowd-sourced approach.

- **R.7 –** Basic UI with main focus on the travel between a select few number of pages essential to the user (video submission, video editing, dataset uses, etc…) with the implementation of a small amount of features such as recording, uploading and downloading content.

- **R.8 –** Modular framework to minimize the impact of errors and ease work load distribution between peers.

TOOLS

- **Hardware –** RGB camera
  Specific environment conditions (tbd)

- **Software –** Mongo DB // Python // React // Flask or Node.js

- **Version control & communication –** GitHub // Discord // Microsoft Teams // Pen and Paper

- **Documentation -** Yet to be determined. Possibly Overleaf, Microsoft Word and Excel