1. **Video Rendering Architecture Design:**

Please define and describe Your architectural vision for video (.mp4) & animated gif (.gif) rendering, based on SVGator’s animated SVG, using server-side rendering.

Animated SVGs exported with SVGator can be controlled using [SVGator JS API](https://www.svgator.com/help/getting-started/svgator-player-js-api).

* 1. Specs:
     1. Allowing more parallel processes
     2. Limiting maximum processes to run at a given time, having over-demand processes set on pending
     3. Ensuring acceptable waiting time for all user groups (free, lite, pro)
     4. Ensuring visual feedback to the user on progress of rendering
  2. Input of the process:
     1. User-initiated SVGator export of animated SVG (using JS as export option & PRO account - see [example](https://cdn.svgator.com/js/player-api/assets/Work-Animation.svg?v=2))
     2. Export settings (resolution, fps, format, quality, speed)
  3. Output of the process:
     1. Rendered video (or animated gif)

Expected result would be a document describing architectural, functional & technical components, their relations to one another & the whole end to end process (feel free to use diagrams).

* 1. Output:
     1. White paper / a document describing the end to end process
     2. w/ a possible diagram

1. **CSS Animations Converter:**

Build a basic front-end application with the following functionality:

* Allow the user to upload an SVG
* Find all (CSS) animations in the given SVG & build out a JSON structure from them
* Display both the animated SVG & the JSON content
* Offer possibility to download the given JSON (w/o the need to send to a back-end server)
* Offer possibility to upload a JSON file (w/o the need to send to a back-end server)
* Apply the animations from JSON to the uploaded SVG (assuming that all elements, ids, tags, etc are matching between the animated SVG the JSON has been exported from and the one uploaded)
* ~~Example SVGs are uploaded in the same directory~~
* Example SVGs to test with can be found in the same directory as this document