PROJECT SUCCESSES:

- Overall, the final project outcome in terms of tools used and code structure were done very well. For instance, the folder structure was a proper project, the use of a remote Docker for testing, documentation such as project conventions were done very formally, the proper use of test driven development, good object oriented design (e.g. MVC pattern), and many other things.
- The use of the Django and other third party python libraries such as Requests, Sphinx, and Pytest.
- Work load distribution was clear and transparent due to using a lot of project management tools (although see the section of *"Using Gitlab's Project Management Tools"* section).

PROJECT FAILURES:

- Unable to influence my group members to produce more work and of higher quality.
- The use of multiple social media applications to communicate caused troubles. Specifically, migrating to both WhatsApp and Slack to communicate midway broke down communication.
- The delay in formalizing git commit message conventions lead to many git clean ups.

QUALITY OF SOFTWARE DELIVERABLE:

Despite the quantity of content delivered, the quality of the deliverables was great. The GUI was sleek, simple and resizable. The overall technical complexity was high since we used Django, Sphinx, and other tools (it took really long to learn them from scratch). The coding style and structure were systematic (it would be strange if they weren't), packaging and scripting was done properly using a "setup.py" file.

EFFECTIVENESS OF THE PROCESS FOLLOWED:

In terms of completing backlog tasks, they were done rather sloppily. The reason being is that rather than focusing on a single or two tasks per week, the focus was redirected to completing a small portion of all of the tasks which implied showing functioning work for demos was rather difficult.

USING GITLAB'S PROJECT MANAGEMENT TOOLS:

This was rather tragic. The purpose of such tools is to promote group interaction yet it seems that no one was using the issue tracker, commits, builds, milestones, nor the wiki. It ended up with me spending a lot of time setting up such tools and managing them with little benefit other than for logging and mark scheme purposes. Although in terms of learning experience, it was super valuable.

LESSONS LEARNED:

- Adding two more management roles assigned to two different people. Specifically, having a **director role** who oversees the direction the project is headed and a **project coordinator role** who is responsible for dealing with group members and their assigned short term deadlines.
- From the start, having the project folders structured properly and running everything through the root directory to avoid impending import problems.
- Being stricter about group members not meeting deadlines and perhaps even raising complaints to my superiors in hopes of seeing something change (or at least try to).

There are *many* and **many** more things left to mention in all the above sections since it *just happened* to be that I ended up doing most of the group project work but I have omitted them to keep this document short.