**Development Document**

* *Concept:* A short description of your core idea (~50 words).
* *Specification:* A clear outline of the specification of your application (~100 words).
* *Research, Development and Technical Description:* Details your research into creating the application, as well as details on how the application was built, and how it functions. For example, What APIs were used? Why were they chosen? Were any alternatives tried? How were they utilised? (~550 words).
* *Critical Reflection:* A critical evaluation of the successes and limitations of your proof of concept. This should also describe what personal learning you need to pursue in future to refine your skills (~300 words).

I was tasked with the creation of a proof-of-concept web application that utilises at least one API and/or libraries that have been covered with in the module or other comparable web API’s that I found myself. With this task at hand, I conducted detailed research into the creation of an API to find one suitable for my proof-of-concept web application.

In order to achieve my object, not only will an API and/or library will be required, but also fundamental knowledge of HTML, JavaScript and CSS. Furthermore, the use of a source-code editor will be required to compile the necessary code, for this assessment I will be using Visual Studio Code for its utilisation of embedded Git as well as having prior experience with this IDE. It is also a much more streamlined code editor that is suitable for a task such as this, rather than using a more complex IDE required for larger scale projects. The culmination of these skills and assets, produced using VS Code, will help produce a fully functional UI that can be deployed to the web and will allow the user to interact with my selected API.

When I started investigating what API to use, I first conducted external research into the different API’s that have been discussed in my Tomorrow’s Web module. One such demonstration was the “Hand tracker” Tracker” showcased in Week 8, this application was extremely intuitive, utilising JavaScript and the user’s webcam to interact with the application. It uses Machine Learning to take the data from a user’s webcam of the positioning of their hand. The data is returned asynchronously, giving a response of the hand positioning which updates in real-time on the screen before rendering points of the hands. This technology can be employed to create a UI that allows the user to draw different graphics, which was also displayed during the lecture as an example of one of the uses using this API. Following this example, I investigated other potentials use of this API to be implemented within my own proof-of-concept. One such concept that I explored the implementation of was to use this API to track the hand movements of the user and for it to recognise certain positions, specifically an open palm, a closed fist and the extension of the index and middle finger. This real-time webcam recognition could then identify when these specific hand positions were made to create essentially a ‘move’ from the game “Rock, Papers, Scissors”, and can be used to compete in a game with another player. However, due to the intense GPU usage required by the machine learning, and the discovery of my web-cam being inoperative, I continued on with my research for another more feasible API.

This directed me towards an API that had not been discussed in class, a weather API which can be utilised to display a variety of weather information found within an array. I researched the documentation of