1. **Introduction - 2 parts**

* State a real world problem that you intend to solve. What products do you know of that exist, but could be implemented better in some way? Where possible, use statistics to validate your argument e.g. “x” percentage of people suffer with dyslexia or “x” percentage of people fail the theoretical part of their driving test the first time they take it.
* Explain what it is you intend to build and why. What need or gaps exist for your product? This should be a single paragraph in length and should not go into any of the specific features of the solution unless it is important to do so as part of your justification.

1. **Decomposition/ Computational Amenability**

* Describe each of the features of your solution in detail and explain why this feature contributes to the solution being developed using computational means i.e. why this feature is better suited to a computational approach e.g. searching for example can be done far faster by means of a processor than a human searching through a catalogue etc.

1. **Stakeholders**

* Identify 2-4 stakeholders that would have some sort of interest in your final solution. Who are they and what are their characteristics?
* For each of your stakeholders, you need to also identify their requirements (3-5 requirements per stakeholder) e.g. how will they use the system? Will they require any specialist hardware? What data will the user have to provide to the system? What data may they expect from the system? Will the user require an output/ result from the system in a specified amount of time? Etc.

1. **Research into existing implementations**

* Identify at least two existing implementations (or something similar) to that of your own solution.
* It would be a good idea to find one very successful implementation (that that has many features that you would like to adopt in your own solution); and one poor implementation (what that does not meet the objectives that it is supposed to well).
* For each implementation, list all the good point and bad points to that implementation, along with some description of what you may adopt/ improve upon.
* Be sure to include screen shots of the implementations that you find to provide a visual representation to aid your discussion.

1. **Requirements for the proposed solution – 2 parts**

* Explain any specific requirements that you will have in order to be able to complete this project to a good standard e.g. access to computer facilities, one to one time with class teacher to discuss ideas and research etc.
* Explain any requirements of the end user e.g. any specialist hardware/ or even basic hardware and software requirements. Will they require access to any other files to run the implementation; or a Wi-Fi connection etc.?

1. **Limitations of the proposed solution**

* What are some of the things that your solution will be unable to do? Explain how you may tackle this limitations; or why it is not feasible for you to overcome these limitations.

1. **Success Criteria**

* This is a checklist of all the things that your project must be able to do in order to be deemed a success and is what you will refer to during testing and evaluation.
* Be sure to justify the importance of each of your success criteria (ensuring their validity within the mark scheme).
* Be sure to make all of your success criteria measurable/ quantifiable where possible.