NEA Guide structure

The project will be written up in 5 sections, each with its own subsections.

* Analysis (9 marks)
* Documented Design (12 marks)
* Technical Solution (42 marks, broken down into 2 sections of 15 and 27)
* Testing (8 marks)
* Evaluation (4 marks)

**Analysis: (9 marks across 3 levels, Level 1-3)**

Required documentation:

* A clear statement that describes the problem area and the specific problem being solved or investigated within the project.
* An outline of research and investigation into and around the topic.
* 5 card Charlie (removed from most casinos due to advantaging player)
* Programming languages for blackjack
* Insurance
* Number of decks to house advantage
* Counting cards, identify covered cards, shuffle tracking
* Different types of blackjack games
* Online blackjack (apps, websites)
* Hardware Requirements
* Research into ai for blackjack
* A statement indicating who the problem is being solved / investigated for.
* Background information in sufficient detail for a third party to understand the problem being solved / investigated.
* A list of measurable, “appropriate”, specific objectives. This should cover all functionality of the solution or areas of investigation. (Appropriate being that each objective is a single purpose objective with no ambiguity
* Any modelling of the problem that will inform the Design stage (e.g. graphs, E-R diagrams, state diagrams, data flow diagrams, flow charts etc.)

The KEY thing to identify, are the objectives of the project. These will be used to assess the success of the project. The more you identify and the more you meet, the higher the grades will be.

Additional elements that would enhance the report:

* Current System Specifications (if appropriate)
* Identification of an end user / supervisor
* Prospective users
* Modelling of the problem (Data Flow Diagrams, data dictionaries)
* Research around the investigation area

Supplementary material (non-essential)

* Proposed solution details
* Acceptable limitations (if agreed with an end user or supervisor)
* Data volumes
* Log/Diary of research

**Documented Design (12 marks across 4 levels)**

**Required documentation**:

* Structure/Hierarchy chart
* System Flowchart
* Data Flow Diagram / Class diagrams with additional details
* Non-standard diagrams that combine elements of data flow and program control are acceptable.

You should also document the following:

**Algorithms**: Processing of data should be the core focus. Show and explain key algorithm(s) that your project uses. Whether you made them or incorporated them. Use them to show how sophisticated your project is. Pseudo code, structured English or other appropriate method is acceptable.

**Data structures**: If you have data structures storing memory, these should be explained. Arrays or records might require a short explanation. Queues, Linked Lists, Hash tables could be explained in more detail.

**File Structure and organisation**: If you’re storing data in files, the structure of these files and folders should be documented and explained

**Database design**: If your project uses a database, the tables should be described and an E-R diagram used to show the table relationships

**Queries**: If the project uses a database, samples of the queries used, together with explanations should be provided.

**Human Computer Interaction (HCI):** In almost all cases, it is expected there will be on-screen interaction between the project and the user. A small number of screenshots / hard copy, with explanations and reasons for design layout and use, should be presented. HCI can be presented as explained screenshots.

**Hardware selection/design:** This is only necessary if the project requires a very specific hardware setup. Such as servers, Microbits, Raspberry Pi etc. If so, explain the suitability of the hardware and how it is used.

**Technical Solutions (Completeness of solution and Techniques used)**:

For this section, include fully annotated program listings.

Most of the marks will come from teacher/moderator assessment of the code.

Your project **WILL** be observed by a teacher and assessed against the objectives you specified at the start of the project.

Your coding will be assessed by its complexity according to the Overview Booklet starting on page 10.

**Testing (8 marks):**

There is no “minimum limit” on tests. Write up an introduction and overview before going into the testing details. Your tests however should cover 2 areas.

**Development testing**:

Showing where and how development failed or changed as a result of on-going testing while creating the software for the project. Did you change plans as a result? Why? What were the solutions around the problems found during development?

**End product testing**:

This should be a testing table that covers all the key areas of the code. For each test carried out, it is expected that you will describe and comment on the outcome of the test. Include the following:

* Purpose of the test
* Test data used
* Expected outcome
* Actual outcome

Included also screenshots of before and after the test. You should be ideally testing ALL of the objectives you defined at the start of this report.

Remember key testing areas. Robustness, Normal data, Extreme data, Erroneous data.

**Evaluation: (4marks)**

This will allow you to reflect upon the success of the project in meeting the objectives identified in the Analysis stage.

**Required documentation**

* An explanation of how well EACH objective was met and how this was achieved.
* A discussion of how the solution could be improved
* (possible) feedback from a 3rd party involved in the analysis stage

A sensible approach would be to copy the objectives out of the Analysis section and commenting upon each one, individually.

Moderators appreciate a fully honest self-appraisal, identify your strengths in this project as well as where time was insufficient to complete certain tasks due to changes in the project.