Project Proposal

1.) Research

Your first job is to conduct research on:

- How hardware & software is used in the industry
- Newly emerging technologies
- How 'digital' (aka technology!) could be used to meet the needs of different users
- Which Industry-specific guidelines and regulations will you need to follow?

Use this Sources Table

Use the Research Template to help you

2.) Business Context

EXAMPLE

? What is a 'Business Context'?

An outline of factors impacting the business including:

- How decisions are made
- What the business is ultimately trying to achieve

? What should it include?

- Give an overview of the business's requirements:
  - What new system/software is needed?
  - Why is it needed? (e.g. increased security? Changes to product/services? Etc.) – aka 'Project Scope'

ⓘ *This information can be found in the project brief* ⓘ

- Carry out research relating to the specific context and market environment, including:
  - Common problems and risks ⓘ *Scroll down to bottom to see list of potential 'risks'* ↓
  - Current uses of hardware and software within the identified context
  - Newly emerging technologies ⓘ *Open research template to see list of 'newly emerging tech.'*
  - Existing or potential solutions and how these meet different user needs
  - Industry/situational-specific guidelines and regulations ⓘ *List of these also on research template*

- Identify shortfalls in own skills and knowledge and plan learning opportunities to make up for these shortfalls.

Explore and understand the initial client request/project brief.

ⓘ *The purpose of all this is to assess how well you understand project brief and business environment in which it's set.* ⓘ

3.) Functional & Non-functional requirements

EXAMPLE

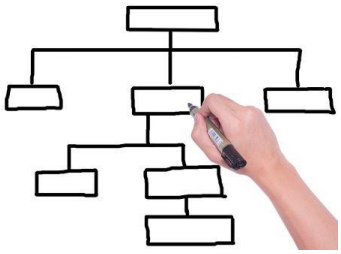
Functional	Define what a system must do (it's features)
Non-functional	Describe how the system should do it (its performance)

Differences between 'Functional' & 'Non-Functional' requirements	
Functional	**Non-Functional**
Based on what the system should do	Based on the system's quality/performance
Detail the product's features	Detail the product's properties
Essential	Not necessarily essential
Based on the user's requirements	Based on user's expectations/experience
Cover functions the software must perform	Cover aspects of good user experience
Tested by checking whether it works or not	Tested for usability, performance, stress, security
Tested first	Tested after functional testing
Important to system operation	Not always important, but may be desirable
Usually involve: Functions/processes, Transactions, Authorization levels, Reporting Requirements	Might include: Availability, Reliability, Scalability, Performance, Security, Usability
↓ ⓘ *The spec says your requirements should cover* ⓘ ↓	
The inputs required, data needed, data processing that may take place, Logic of the system, Deployment & usage platforms for the software.	*Security considerations, required accessibility features, scalability requirements, KPIs (in relation to responsiveness, load handling & reliability). + user acceptance criteria.*

4.) Decomposition

? What is '**Decomposition**'?

Decomposition is about decomposing (breaking down) the problem/system you've been faced with in the brief.
The purpose is to help make more sense of the problem and to make it easier to implement its requirements later on.

? How do I document it? EXAMPLE	
	Showing how you've decomposed the problem into smaller parts by creating a hierarchy diagram. ← You may be able to break the problem down by looking at: • Different functions (Functional Requirements) • Different independent features of the system • The needs of different users It's a good idea to assign Story Points to each part at this point

↓ ⓘ *These are also worth starting to think about at this point* ⓘ ↓

Pattern Recognition:	Looking for patterns/similarities within the problem(s) you've decomposed EXAMPLE

Can you spot any potential to make your proposed solution work more efficiently?

Using Loops?	**Using Functions/Sub-routines?**	**Using OOP (Object-Oriented)?**	**Using CSS?**

Abstraction:	Simplifying the problem(s) by removing unnecessary detail 1.05 EXAMPLE

Travellers need only see what they need to know ↑

In Software Development, abstraction can be shown by using different Low Level & High Level diagrams.

High Level:	*A 'zoomed out' view of the system/sub-system*
Low Level:	A *'zoomed in' more detailed view*

These could take many forms…
(Flowcharts, ERDs or other types of diagram).

5.) Key Performance indicators (KPIs) and user acceptance criteria EXAMPLE

? What are Key Performance Indicators (KPIs)?
- Measurable statements associated with your objectives
- They set out what you intend to achieve & by when?

? What are they for?
- To help you monitor your progress
- Can be used to measure performance or efficiency

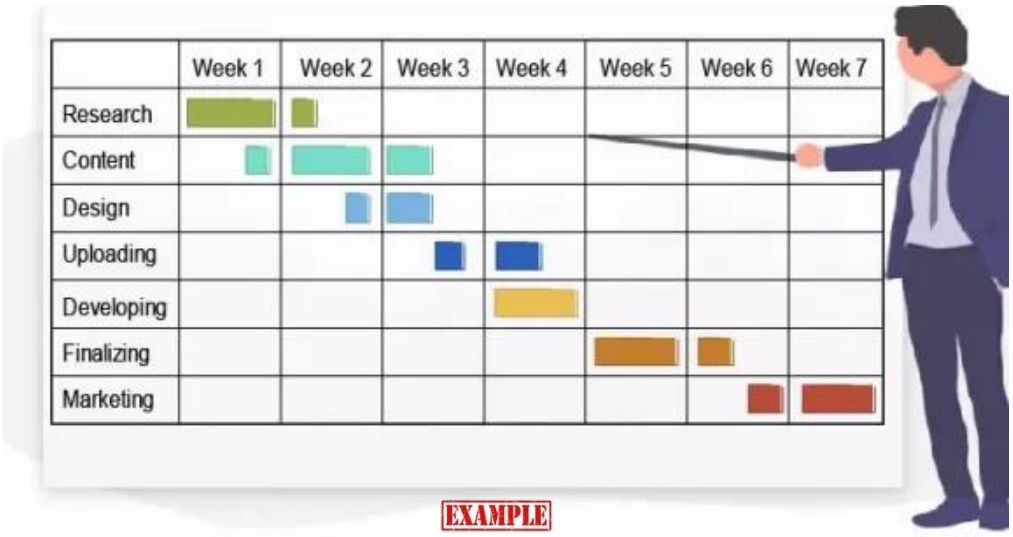
KPIs can be set to measure performance based on:

Code Quality	Efficiency (lines of code), complexity, scalability, maintainability (*# comments*)
Testing Quality	Tests performed, successful tests, bugs/defects found, bugs/defects fixed
System Performance	Requirements met, page load time, security, responsiveness, load handling, reliability
Productivity	Tasks completed; hours worked - can be based around sprints vs story points (if Agile)
Customer Satisfaction	Client/user satisfaction survey ratings/scores – perhaps from surveys carried out

6.) Description of proposed solution:

EXAMPLE

- Description of what you are proposing to create
- Timeline dates & milestones (probably a good idea to create a Gantt Chart for this):



7.) Justification:

- How solution meets needs of client and users
- How will your solution fulfil the needs of other key stakeholders?

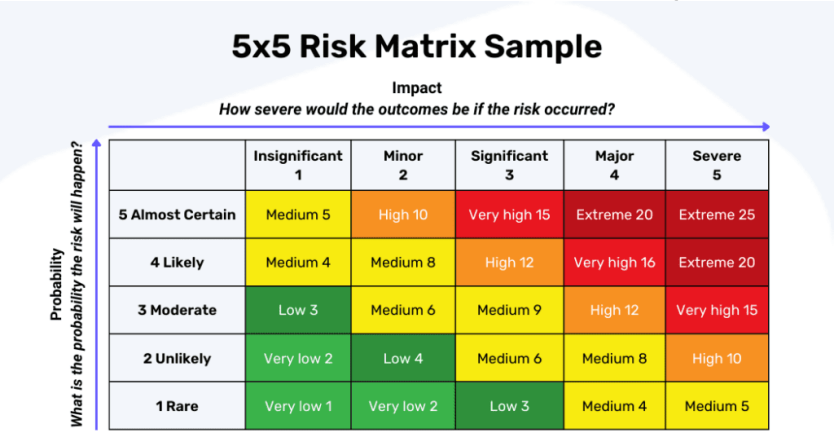
Stakeholder Group	What are their needs?	How my solution will fulfil their needs?
Customers	1. Order coffee for collection quickly and efficiently	My solution will
	2. Be able to book a place at either of the three restaurants	
	…..	
Shop Staff	Clearly see orders placed and their collection times	
…..	…..	

- How potential risks will be mitigated 🖇

? What are some of the risks you can talk about?

Data & system security	**Malicious / accidental damage**	**Compatibility with other systems**	**Slow speed of development (Getting behind schedule)**
Not meeting functional & non-functional requirements	**Not meeting key performance indicators (KPIs)**	**Legal and ethical considerations (list below)**	**End-users reluctant to engage**
Product Reach (People not being aware of the new system, or how to use it)			

You should also create a Risk Assessment Matrix



	Insignificant 1	Minor 2	Significant 3	Major 4	Severe 5
5 Almost Certain	Medium 5	High 10	Very high 15	Extreme 20	Extreme 25
4 Likely	Medium 4	Medium 8	High 12	Very high 16	Extreme 20
3 Moderate	Low 3	Medium 6	Medium 9	High 12	Very high 15
2 Unlikely	Very low 2	Low 4	Medium 6	Medium 8	High 10
1 Rare	Very low 1	Very low 2	Low 3	Medium 4	Medium 5

? What can you put in place to help mitigate these risks?

Backups	**Security**	**Personnel, skills and training**	**Business continuity planning**	**Disaster recovery planning**
Confidentiality, integrity and availability (CIA)				

? How will legal & regulatory requirements be addressed?

Intellectual property	**Consumer protection**	**Age ratings and classifications**	**Advertising laws**
Data protection and privacy	**Copyright and patent**	**Gambling legislation**	**Territorial restrictions**
System security	**Equality and diversity**	**ISO/IEC/IEEE 90003:2018**	**W3C**
Responsibilities concerning staff and employment practices			

? What ethical considerations will you need to consider?

Codes of conduct	**Professional practice**	**Software licensing**	**Inclusion and diversity**