

# DESIGN DOCUMENTATION

---

# WHAT IS A DESIGN PACK?

- In this section you need to "design" how the whole system will look / work / store data etc.
- WITH ALL of your thoughts / decisions recorded and justified
- Your designs need to be detailed enough that you could give your design pack to someone else and they should be able to build what you have envisioned.

# ORGANISATION OF YOUR DESIGNS

- There is no set order to a design section,
- Order them how you feel works for you
- Just remember to have commentary of your decisions and reasoning (linked to the functional and non-functional criteria you have identified)

# OVERVIEW / DECOMPOSITION DIAGRAM

- Have a large flow chart style diagram which shows the overall structure of your site and link to database and information will flow.
- Explain the diagram and what it shows.
- SHOULD BE THE FIRST THING IN THE DESIGN SECTION

# ER DIAGRAM FOR YOUR DATABASE

- Build an ER Diagram showing the entities (tables) you will have and the types of connections they will have (one many etc)
- Justify / explain how you have come up with this and why it will be the best design for the solution you are working towards

# DATA FLOW DIAGRAM

- Show an overview, with your database shown as well, and how data will flow between various pages and into/out of the database.
- e.g. Showing how the userid will be "flow" between pages, which is initially collected from database when a user logs in.

# TABLE STRUCTURE (DATA DICTIONARY)

- Give a list of the fields (columns) you will have in each table, including primary key and foreign keys
- List the data type you will use for each of the fields and justify your decisions.
- Give an example of a piece of data that will be stored in each of the fields.

# DATA STRUCTURES / VARIABLES LIST

- You need to build a table of all the variables you think you will need, probably best done after the rest of the design, or as you are designing.
- Example table for you to use:

Variable name	Scope	Data type	Where used	Example data	Reasoning
Dbcursor	Global, but passed as a parameter	Database cursor type (Object)	Across the whole solution	n/a	This is a needed element to work with / on getting data into and out of the database.

# COLOUR SCHEME

- Show what colours you intend to use in your website
- Make use of a colour picker to choose complimenting colours into a palatte
- Justify how these colours link to each other and to the theme of the scenario
- Explain where you are likely to use each of them e.g. font colour, heading colours etc.

# INTERFACE DESIGNS

- Use a tool like figma etc to mock up what you intend to make the solution look like
- Make use of your intended colour scheme
- Make use of the styles of font / headings etc you intend to you
- Make sure there is a commentary to go along with why it looks like it does (maybe compare it to competitors and explain your decisions.

# ALGORITHM DESIGNS - FLOWCHARTS

- You should use flowcharts to show the main flow through a process e.g. registration with password checking, checking for unique username etc.
- This are not so much about smaller algorithms (like inserting into database or checking password strength) but more about the bigger system like registering, producing analytics.

# ALGORITHM DESIGNS – PSEUDOCODE

- Yes, you have to make use of pseudocode.
- You can/should add in a flowchart too.. But you have to have pseudocode
- You DO NOT need to design code to make the page appear
- Just design code / algorithms that do something important e.g.

Connect to the database... design it

Insert to the database....design it

Get out of the database... design it

Process data from the database... design it

Process data from input.. Design it

# TEST STRATEGY

- Develop a testing table of what you intend to test and how you will test it, including data you will use as part of the test.
- Justify these tests and the data as to why they are good / useful.

Date of test	Component to be tested	Type of test to be carried out
<b>18 April 2024</b>	Navigation bar	Black box testing -> Functional testing: Integration testing
<b>18 April 2024</b>	Homepage	White box testing, Black box testing -> Functional testing: Integration testing, Unit testing

# TIPS

- Can someone else in the group build your solution from your design?
- Have you linked each part of your design to your non-functional and functional requirements (use the numbers and justified how it links / solves that requirement)