

# intervention

- ChatGPT is allowed  
↳ use it!!!

## RISKS

- discuss mitigation of risks
- Time scale, money (budget), overscope/underscope

## DECOMPOSITION

- break down the problem into a diagram
- diagram → structure of the website you are going to build the features on each page

## USER ACCEPTANCE CRITERIA

- list of things that the users see/feel

### 1. user interface

#### 1.1 Navigation system

##### 1.1.1 horizontal Nav Bar

##### 1.1.2 clean contrasting colours

## FUNCTIONAL / NON-FUNCTIONAL

↓  
things the system  
should do  
(focus on business  
needs)

↓  
how the system  
should do things

## KEY PERFORMANCE INDICATORS (KPIs)

- indicate the success of the project from a business perspective  
↳ should be categorised where possible

## DESCRIPTION OF PROPOSED SOLUTION

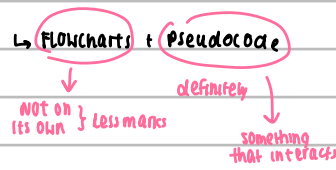
- give an overview of what your solution will be

## TIPS

- Justify everything
- thorough and accurate list of websites **TIME + DATE**
- HEADINGS

# design documentation

- No set order
- Have commentary of your decisions + reasoning



## databases

- databases are an organised way to store data

### draw SQL

Entity - table - grouping bits of data

Field - column - single item of data

record - row - collection of all fields together to make information

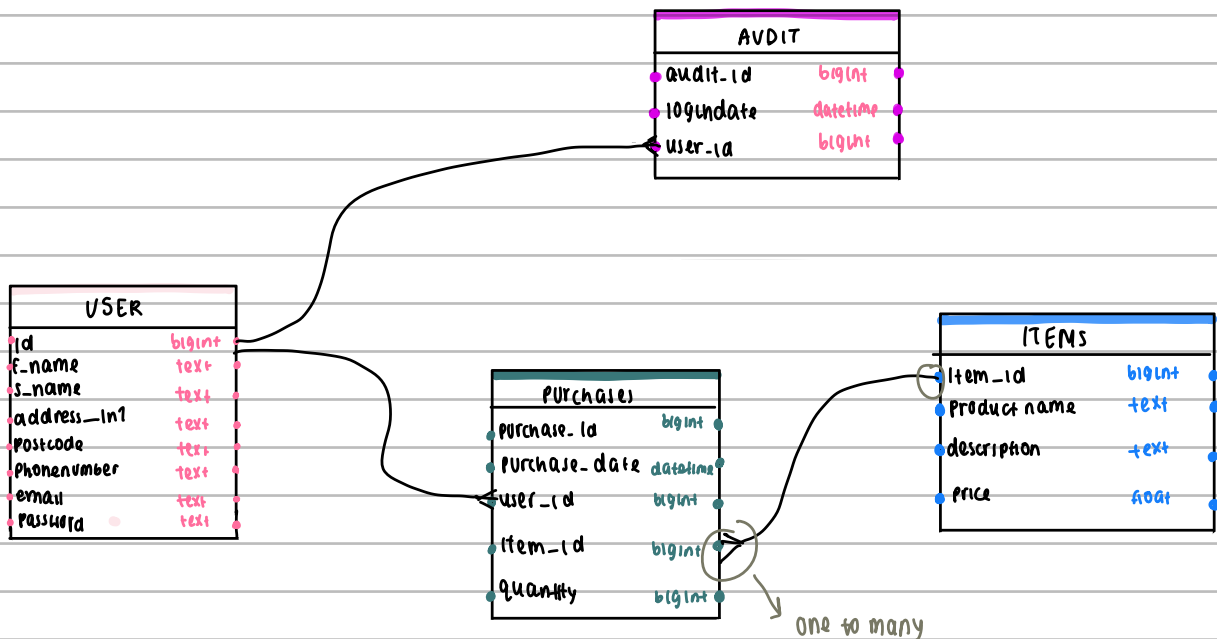
table - single table of data

redundant data - repeating data

Relational databases - more than one table linked together

primary key - uniquely identifies a record of information

foreign key - a primary key from another table



development 30 hours

minimum 3 versions of software

development log

Version discussion: What will be done in each version/why

Version 1: basic admin, VR, functional/non-functional requirements - WHY

Version 2: any remaining important functionality, VR, FR, NFR - WHY

Version 3: any other non-essential functionality,

Development diary: show the examiner what you have, tell them how it went

Reference VR, FR, NFR

End of V1: What did I plan to build, what did I actually build, why is there a difference,

any thing has change

priorities for next version - have a discussion around what you will do and why

End of Version 3: what went well, what was bad, what's changed and why: moving forward, what would you do/add/change - WHY

Readable

Maintainable

Robust

All of your code,

Export from your database

development diary

Evidence of testing

