For this interview task I was asked to build a prototype with a fully functional inventory and shop system. The approach I took was making a modular system that involved three pillar modules:

- The player module:
    - This module was big one and had submodules that handled:
        - Input: Receiving the input of the player and distribute it through the modules that needs it.
        - Animation: For handling animation of the player character and its outfits aswell
        - Movement: Separate class from input, deals with all the physics and movement of the character.
- The NPC/Shopkeeper module
    - This module handled the interaction between the player and the NPC to open the store.
- The Inventory System:
    - This inventory system handle both player and shop inventories.
        - The base of the system consists of an inventory filled with some items that can have an action (SELL, BUY, Equip, any other).
        - The system splits into
            - Controller, to handle any logic to the inventory.
            - Visual Controller, to handle any UI updates.
            - Database, for using item Data by Id
        - The item data was handled with a scriptable object called Item Database. This database had all the information about an item that could be stored agnostic of runtime gameplay.
        - The data of the item that was based on the inventory, such as quantity of item, was saved inside the inventory class through a list.
    - TODO: We can clean up a bit both player inventory and shop, with an abstract class and virtual methods, because a lot of the visual controller code is shared between the 2. Also, a lot on the UI and animation part, I implemented lean tween to start doing some more animation but just did the opening and closing of inventories.