Home Assignment questions:

1. **Unity Versions**: ***Which version do you currently work with? Have you worked with different versions, and did you encounter challenges in transitioning?***
   Since I opened a fresh project on my new computer I didn't have any versions so I've used Unity 6 for the first time! I didn't get down to all the new features but some UI and menus changes are pretty nice (build menu for example)
   I've used a lot of unity versions since probably unity 5, so too many to count, transitioning can be hard sometimes especially with an old project with external dependencies (that are not updated or supported) but in this case I had no trouble since it's a new project.
2. **3D Assets and Models**: ***Do you have experience with 3D asset integration in Unity? Please share a project example, including how you handled models and lighting for performance.***
   I have worked with a lot of 3d assets whether with artist I worked with or without them (asset store and online) an example (from long ago) was integrating into unity the cannon (sort of) on this game, in mobile games and online game where resources sometimes limited there's not a lot of real time lighting since it can be consuming to the resources we would usually try to fake the effect with a shader or some sort of creative solution if non is found then pre baked lighting and use static lighting as much as possible
3. **Web Accessibility (WebGL or Alternatives)**: ***Do you have experience developing Unity products for the web, e.g., with WebGL? If yes, please describe a specific project and challenges faced.***
   I have both, in my first job as a unity developer we did Educational games that needed to work on webGL as well, first of all the browser have evolved since then but still the resources of the browser can be limited and unity wasn't very efficient with the build so the size of the builds were big and a lot of memory and security issues since the browser has privacy restrictions.
   I also found that at least at that time WebGL wasn't a big priority for unity and not supported as well, which left some critical bugs (a few with UI that I remember) that were not fixed (I had to make a Text editor in unity and I was left without options)
   An alternative that I know of is Pure JS and some smaller engines that are much more efficient and are used to create games limited in size such as playable ads (1.5 Mb) here all the assets are optimized and nothing added which is not needed
4. **Salary Expectations**: ***What are your salary expectations for this position?***
   I would ask for around 35-40K but I always say that salary is a lot but not everything when it comes to making a decision i've always found that when both sides are interested an agreement could be made
5. **Availability**: ***Do you have a specific date when you could start?***
   I'm currently in Australia and I plan to come back around the start of december, but I have a little flexibility and also I can start working from here in some capacity

Documentation

- Instructions to run locally:
  - I have used Visual code's extension named LiveServer using this tutorial it worked pretty well and quickly since I didn't need the privacy part just use http instead of https
  - Any other local server options should work such as Docker and other solutions
  - I tested it on Chrome and Safari on PC, Iphone 13 and Iphone 15, unfortunately i have no android devices for testing
- Assets and Libraries:
  - I've used the following unity packages and other built in solutions:
    - DoTween
    - TMP (included in UI now)
    - JSONUtility
    - Unity Networking
    - Tasks
    - VisualCode for unity

- Overview of code structures and decisions:
  - **Componentes:**
    - StoreManager
      - Initialize the store at the moment just the shelf but can easily be expanded for dynamic shelfs, receives server data (or mock data for testing) and initializes the shelf
    - ServerManager
      - Used by store manager to acquire server data and parse it to classes, for StoreManager to use
      - Holds the Data classes, In a more complex situation where more data manipulation would be required the data classes would be in a different file
      - Can be easily expanded to also update server (might also require changes in server)
    - ShelfManager
      - Initializes the books and position them dynamically
    - Book
      - Holds book data that can be edited and potentially updated to server
      - Holds UI and Visuals and logic for display

- **Structures and Decisions**
  - I chose a bookstore because this allowed me an interesting, interactive and visually appealing design that will be more engaging to the user than a UI heavy option
  - ShelfManager holds and initializes the books, can be easily expanded to get the books data back to StoreManager and to the server, with changes to server data more than one can be displayed (small changes to code will also be required)
  - The book holds most of the actual “Game” logic, and the user engagement is with it most of the code is for user interaction and edit, along with movement to engage the user, with almost the same code and a change of Model the book can be changed to any other product
  - Separated ServerManager from StoreManager for less dependency on the implementation and separation of responsiblities of the server side as well as readability
  - Used data classes to parse server data for readability and reusability throughout the code
  - I used Mock Data to insure I can properly test the “Game” itself without any “noise” from server side, once the implementation was done I integrated with the actual server data, in this case it was simple but in bigger projects this method help to separate responsibilities more clearly even between members of a team, an interface between those sides will help maintain clean code and will help with the integration
  - I didn’t add any confirmation messages since the user can see thor changes immediately and I thought more UI Panels will just distract the user
  - This assignment is a pretty straight forward case but I wanted to work on it as an base code that can be changed and expanded with as little change as possible with defined responsibilities and roles
  - My documentation philosophy (from CLEAN CODE - highly recommended!) is that Class names and function names would be understood without added documentation, so the documentation I added is mostly to explain myself in the exercise but my main objective is that reading the code would be clear as possible from first glance and would hopefully be self explanatory 🙂