BACKEND TAKEHOME:

Question For all: How would you go about solving Issues around data privacy and security for our daily check-in feature and menstrual calendars

Simplified Coding Challenge: Basic Multiplayer Game Lobby System

Objective:
Develop a backend system that manages game lobbies for a multiplayer game. The system should allow players to create lobbies, join and leave lobbies, and start a game when all players are ready.

Core Features:

1. Lobby Management:
   - Create a game lobby with a unique ID.
   - List all available game lobbies.
   - Delete a lobby when the game starts or if it's empty.

2. Player Interaction:
   - Allow players to join or leave a lobby.
   - Start the game automatically when the lobby reaches a predetermined number of players.

3. Real-Time Updates:
   - Notify all players in a lobby of any changes (players joining/leaving, game starting) in real-time.

Technical Requirements:

- Backend Language: Use any language and framework you're comfortable with. Node.js or Python with appropriate real-time communication libraries (e.g., socket.io for Node.js) is recommended.
- Data Storage: Use in-memory data structures or a lightweight database (e.g., Redis, SQLite) to store lobby and player information.
- **Real-Time Communication:** Implement WebSockets or a similar technology to enable real-time updates to and from clients.

Key Functions:
Feel free to make more if you need

- `createLobby(playerId, lobbyDetails)`: Creates a new game lobby.
- `joinLobby(playerId, lobbyId)`: Adds a player to a lobby.
- `leaveLobby(playerId, lobbyId)`: Removes a player from a lobby.

- `startGame(lobbyId)`: Checks if the lobby is ready to start the game and deletes the lobby afterward.
- `notifyPlayers(lobbyId, message)`: Sends a real-time notification to all players in a lobby.

Deliverables:

- Source code with instructions for setup and running.
- A README documenting your design decisions and how to interact with the system (e.g., API endpoints, WebSocket events).
- Main - to test