1. Describe the specific code segments you plan to modularize into functions. Explain what each function will do and whether it will have parameters, arguments, or return values.

   A. Data Loading (getlists.js, getmovies.js, getshows.js)

   - Function Name: fetchJSON(url)
   - Purpose: Loads JSON data from a given URL (used for lists, movies, and shows).
   - Parameters: url (string)
   - Returns: Promise resolving to parsed JSON
   - Use: Each file currently repeats code for fetching and parsing JSON. Modularizing this makes updates and error handling easier.

   B. Rendering Data to the Page

   - Function Name: renderItems(items, containerId)
   - Purpose: Takes an array of items (lists, movies, or shows) and displays them in a specified container.
   - Parameters: items (array), containerId (string)
   - Returns: None (updates DOM).
   - Use: The logic for creating HTML and inserting it into the page is similar in each JS file. A single function reduces repetition and makes the code easier to maintain.

   C. Recommendation Logic (recommendation.js)

   - Function Name: getRecommendations(userPrefs, items)
   - Purpose: Suggests items based on user preferences.
   - Parameters: userPrefs (object), items (array)
   - Returns: Array of recommended items.
   - Use: Encapsulates recommendation logic, making it easier to debug and improve.

   D. Function Name: showFallbackMessage()

   - Purpose: Displays a fallback message when data fails to load.
   - Parameters: None
   - Returns: None
   - Use: Improves error handling and user feedback across your site.

2. Discuss Why you chose to convert these sections into functions. Focus on how modularization improves your code's readability, reduces repetition, and simplifies debugging or future updates.

    - Readability: Functions with clear names make it obvious what each part of your JS files does.
    - Reduces Repetition: Shared logic is written once and reused in all relevant files.
    - Simplifies Debugging/Updates: If I need to fix or improve data loading or rendering, you only change one function instead of three files.
    - Flexibility: Functions with parameters and return values can handle different data types and UI needs.

3. Propose any new functions that could enhance your site's features or improve user interaction. For each new function, describe its purpose, what parameters it will take, whether it will return a value, and how it Uses into your overall project.

    A. Sort Functionality

    - Function Name: sortItems(items, key)
    - Purpose: Sorts items by a specified property (e.g., title, rating).
    - Parameters: items (array), key (string)
    - Returns: Sorted array.
    - Use: Lets users view items in their preferred order.