Debounce Search Input

**Level 0: Context Diagram**

Your Debounce Search App prevents excessive API calls by waiting 300ms after the user stops typing before filtering the data. This improves performance and reduces unnecessary re-renders.

**Entities & Processes**

* **User** → Enters a search query.
* **Search System (Process 1.0)** → Handles input, applies debounce logic, and filters users.
* **Local Data Store (localStorage)** → Stores and retrieves user details.

**Diagram Representation**

+-------------------------+

| User |

+-------------------------+

|

v

+---------------------------+

| Search System |

| (Process 1.0) |

+---------------------------+

|

v

+-----------------------------+

| Local Data Store |

| (localStorage) |

+------------------------------+

**Level 1: Detailed Breakdown**

**Process 1.1 - User Input (Search.js)**

* Captures input in a search bar.
* Updates searchQuery state whenever the user types.

**Process 1.2 - Debounce Function (useEffect in Search.js)**

* Introduces a **300ms delay** (setTimeout).
* Cancels the previous search (clearTimeout) if the user keeps typing.
* Ensures filtering only happens **after the user stops typing**.

**Process 1.3 - Fetch/Search Users (Search.js)**

* **Filters users from localStorage** instead of an API.
* Matches results based on searchQuery (case-insensitive).

**Process 1.4 - Display Results (UserCard.js)**

* Displays **filtered user names**.
* Clicking a name shows **full details** (name, age, location, email).
* If no users match, **"No users found"** is displayed.

**Diagram Representation**

+--------------------------------+

| User |

+--------------------------------+

|

v

+----------------------------------------+

| Process 1.1 - User Input |

| (Search.js - Input Field) |

+----------------------------------------+

|

v

+------------ --------------------------------+

| Process 1.2 - Debounce Logic |

| (useEffect - 300ms Delay) |

+---------------------------------------------+

|

V

+--------------------------------------------+

| Process 1.3 - Fetch/Search Users |

| (Filters from localStorage) |

+-----------------------------------------------+

|

v

+----------------- ----------------------------+

| Process 1.4 - Display Results |

| (Shows users or "No results") |

+------------------------------------------------+

**Data Flow**

**User starts typing** → Updates searchQuery state.  
**Debounce function (300ms delay)** → Waits for user to stop typing.  
**Search logic runs after 300ms** → Filters users based on the input.  
**Results displayed dynamically**:

* **Matching users found** → Shows names.
* **No users found** → Displays "No users found".