Learner Details

- **Name:Nikhil k**
- **Enrollment Number: su625mr004**
- **Batch / Class: Mern Stack**
- **Assignment: Filter**
- **Date of Submission: 06-07-2025**

Problem Solving Activity 1.1

1. Program Statement

Build a React component that filters and displays a list of users based on a search input. As the user types, the displayed list updates in real-time, showing only matching results.

---

2. Algorithm

1. Accept the current search term and a method to update it (setSearch) as props.
2. Filter the users array based on whether each user□s name includes the search term (case-insensitive).
3. Render:
    - A heading and input field
    - A dynamically filtered list of matching user names

---

3. Pseudocode

```
START
  RECEIVE users, search, and setSearch as props
  FILTER users where name includes the search text (ignore case)
  DISPLAY input field bound to search state
  ON input change, update search using setSearch
  DISPLAY list of filtered users
END
```

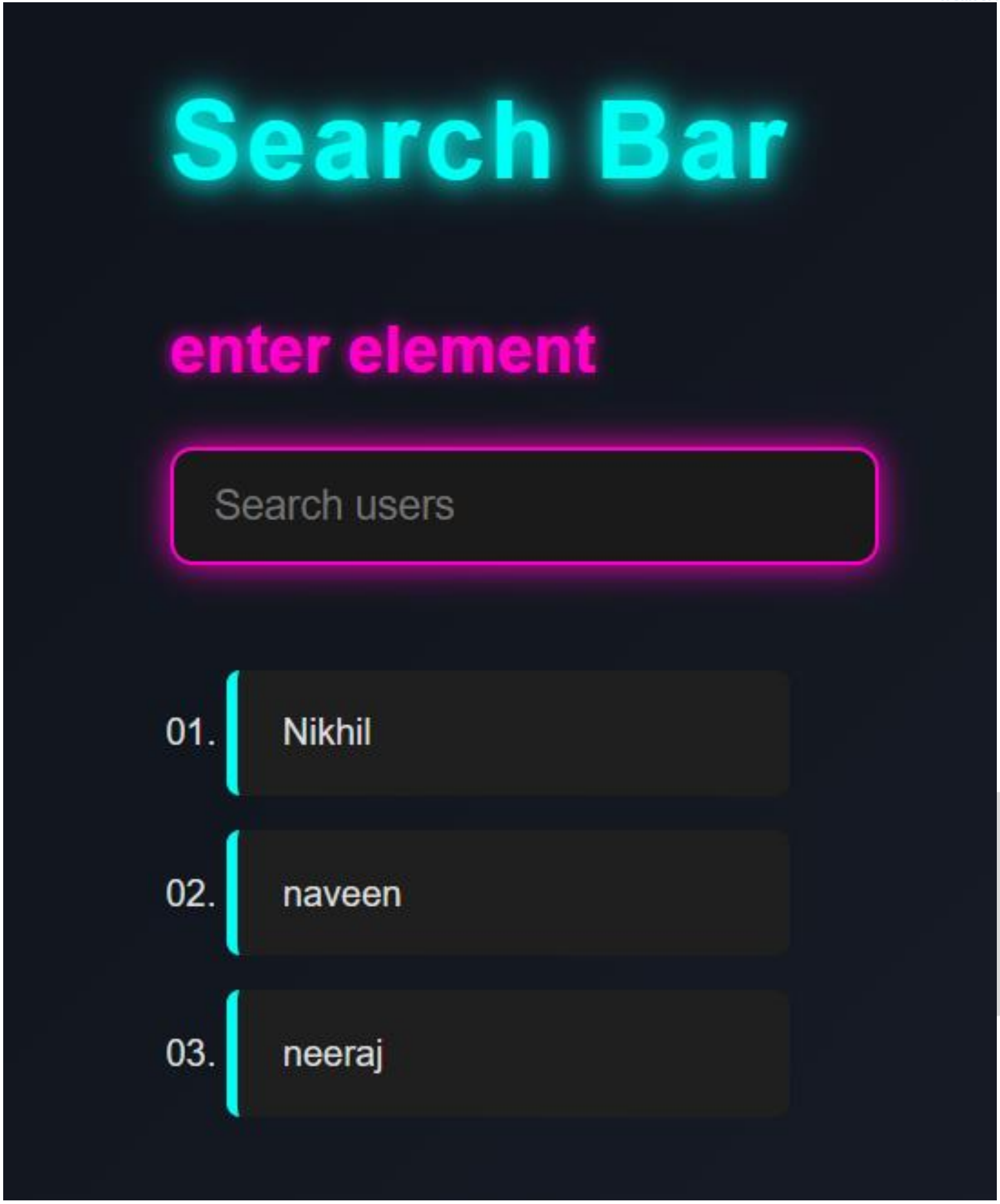
4. Program Code

```
import React from 'react';


const Userfilter = ({ search, setSearch, users }) => {
  const filteredUsers = users.filter(user =>
    user.name.toLowerCase().includes(search.toLowerCase())
  );


  return (
    <div>
      <h2>enter element</h2>
      <input type="text" placeholder="Search users"
        value={search}
        onChange={(e) => setSearch(e.target.value)}
      />
      <ol>
        {filteredUsers.map(user => (
          <li key={user.id}>{user.name}</li>
        ))}
      </ol>
    </div>
  );
};


export default Userfilter;
```

5. Test Cases

Search Term	Example Users	Expected Output
""	[{id: 1, name: "Alice"}, {name: "Bob"}]	Alice, Bob
"a"	[{id: 1, name: "Alice"}, {name: "Bob"}]	Alice
"BO"	[{id: 1, name: "Alice"}, {name: "Bob"}]	Bob (case-insensitive)
"x"	[{id: 1, name: "Alice"}, {name: "Bob"}]	(empty list)

6. Screenshots of Output



Search Bar

enter element

Search users

01. Nikhil

02. naveen

03. neeraj

---

7. Observation / Reflection

- I learned how to filter an array in real-time based on user input using .filter() and .includes().
- Passing search and setSearch as props helped reinforce how **controlled components** work in React.

- This task improved my understanding of how React handles **state lifting** and real-time UI updates.

Problem Solving Activity 1.2

Follow the same Structure as problem Solving Activity 1.1.