■ Part 2 – Frontend Task: User Directory Table

This document provides a complete explanation, line-by-line code, and setup instructions for the User Directory Table React project.

■ Objective:
To create a responsive React app that fetches and displays users from https://reqres.in/api/users with search, sort, pagination, and filtering.

■ Features:
- Fetch user list from API
- Display users in table format
- Search, sort, filter, and paginate
- Responsive UI using TailwindCSS
- Deployed on Vercel

■ main.jsx

```
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App";
import "./index.css";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

■ App.jsx

```
import React, { useEffect, useState } from "react";
import UserTable from "./components/UserTable";
import Loader from "./components/Loader";

const App = () => {
  const [users, setUsers] = useState([]);
  const [filtered, setFiltered] = useState([]);
  const [loading, setLoading] = useState(false);
  const [page, setPage] = useState(1);
  const [search, setSearch] = useState("");
  const [sortKey, setSortKey] = useState("");
  const [filter, setFilter] = useState("");

  const fetchUsers = async (pageNum = 1) => {
    setLoading(true);
    const res = await fetch(`https://reqres.in/api/users?page=${pageNum}`);
    const data = await res.json();
    setUsers(data.data);
    setFiltered(data.data);
    setLoading(false);
  };

  useEffect(() => {
    fetchUsers(page);
  }, [page]);

  useEffect(() => {
    let result = users.filter(
      (u) =>
        u.first_name.toLowerCase().includes(search.toLowerCase()) ||
        u.email.toLowerCase().includes(search.toLowerCase())
    );
    if (filter) result = result.filter((u) => u.email.endsWith(filter));
```

```
    if (sortKey) result = [...result].sort((a, b) => a[sortKey].localeCompare(b[sortKey]));
    setFiltered(result);
  }, [search, sortKey, filter, users]);

  return (
    <div className="p-4 max-w-5xl mx-auto">
      <h1 className="text-3xl font-bold text-center mb-6">■ User Directory Table</h1>
      <div className="flex flex-wrap gap-3 justify-between items-center mb-4">
        <input
          type="text"
          placeholder="■ Search by name or email..."
          className="border px-3 py-2 rounded w-full sm:w-auto"
          value={search}
          onChange={(e) => setSearch(e.target.value)}
        />
        <select className="border px-3 py-2 rounded" onChange={(e) => setSortKey(e.target.value)}>
          <option value="">Sort By</option>
          <option value="first_name">First Name</option>
          <option value="email">Email</option>
        </select>
        <select className="border px-3 py-2 rounded" onChange={(e) => setFilter(e.target.value)}>
          <option value="">Filter by domain</option>
          <option value="reqres.in">@reqres.in</option>
          <option value="reqres.com">@reqres.com</option>
        </select>
      </div>
      {loading ? <Loader /> : <UserTable users={filtered} />}
      <div className="flex justify-center mt-4 gap-2">
        <button
          className="px-4 py-2 border rounded disabled:opacity-50"
          onClick={() => setPage((p) => p - 1)}
          disabled={page === 1}
        >
          ■ Prev
        </button>
        <span className="px-3 py-2 font-medium">{page}</span>
        <button
          className="px-4 py-2 border rounded"
          onClick={() => setPage((p) => p + 1)}
        >
          Next ■
        </button>
      </div>
    </div>
  );
};

export default App;
```

■ UserTable.jsx

```
import React from "react";

const UserTable = ({ users }) => {
  return (
    <div className="overflow-x-auto shadow-md rounded-lg">
      <table className="min-w-full text-sm text-left">
        <thead className="bg-gray-100">
          <tr>
            <th className="py-3 px-4">Avatar</th>
            <th className="py-3 px-4">First Name</th>
            <th className="py-3 px-4">Last Name</th>
            <th className="py-3 px-4">Email</th>
          </tr>
        </thead>
        <tbody>
          {users.map((u) => (
            <tr key={u.id} className="border-b hover:bg-gray-50">
              <td className="py-3 px-4">
                <img src={u.avatar} alt={u.first_name} className="w-10 h-10 rounded-full" />
              </td>
              <td className="py-3 px-4">{u.first_name}</td>
              <td className="py-3 px-4">{u.last_name}</td>
              <td className="py-3 px-4">{u.email}</td>
            </tr>
```

```
          ))}
        </tbody>
      </table>
    </div>
  );
};

export default UserTable;
```

■ Loader.jsx

```
import React from "react";

const Loader = () => {
  return (
    <div className="flex justify-center items-center py-20">
      <div className="w-10 h-10 border-4 border-blue-500 border-t-transparent rounded-full animate-sp
    </div>
  );
};

export default Loader;
```

■ index.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;

body {
  font-family: "Inter", sans-serif;
  background-color: #f9fafb;
}
```

■ Deployment: Push code to GitHub and deploy via Vercel.
■ Live Demo Example: https://user-directory-yourname.vercel.app