JavaScript + Express.js Interview Challenge

Part 1 – JavaScript Array Manipulation

You are given the following sample data:

```
js
CopyEdit
const users = [
  { id: 1, name: "Alice", country: "USA", age: 25 },
  { id: 2, name: "Bob", country: "India", age: 30 },
  { id: 3, name: "Charlie", country: "USA", age: 35 },
  { id: 4, name: "David", country: "UK", age: 30 },
  { id: 5, name: "Eve", country: "India", age: 22 },
  { id: 6, name: "Alice", country: "USA", age: 25 },
];
```

Tasks

1. Reverse the array of `users` without using `.reverse()`.
2. Remove duplicate users (based on `id` or exact match).
3. Get a flat array of all `countries` without duplicates.
4. Group users by `country` into an object like:

   ```
   js
   CopyEdit
   {
     USA: [...],
     India: [...],
     UK: [...]
   }
   ```

5. Find all users who are older than 25 and sort them by `age` (ascending).
6. Return an array of just the `name` values.

Expected Output Example (for #3):

```
js
CopyEdit
["USA", "India", "UK"]
```

Part 2 – Express.js API

Create an Express.js app that uses the **same `users` array** in memory.

Routes

1. **GET `/`** → Return `"Server is running"`.

2. **GET `/users`** → Return all users.
3. **GET `/users/:id`** → Return a single user by `id` (404 if not found).
4. **POST `/users`** → Add a new user (send JSON in body).
5. **PUT `/users/:id`** → Update a user’s details.
6. **DELETE `/users/:id`** → Remove a user.

Requirements

- Use `express.json()` for JSON parsing.
- Use middleware to **log request method and URL**.
- Send **status codes** (`200`, `201`, `404`, `400`).
- Keep data **in memory** (no database needed).

Bonus

- Add `/search?country=India` to return users from that country.
- Add global error-handling middleware.

Submission

- A single folder with:
  - `arrayTasks.js` (Part 1 solutions)
  - `server.js` (Part 2 Express app)
- README with instructions on how to run both.

Time Limit: ~2 hours
Evaluation Criteria:

- Code readability
- Correctness & handling edge cases
- RESTful API design basics
- Use of ES6+ features