Mini Project: “Users CRUD (FS DB)”

CRUD => **C**reate, **R**ead, **U**pdate, **D**elete

Rules

- **One server file**: `app.js` (or `server.js`)
- **One DB file**: `users.json`
- `Add files if You want or need`
- Use **Express** library
- Use **`fs.promises`** for reading/writing the JSON file
- No validations required (don’t check email format, types, etc.)

Data

`users.json` is an array:

```
[
  { "id": 1, "name": "Dana" },
  { "id": 2, "name": "Avi" }
]
```

Endpoints (CRUD)

1) Get all users

GET `/users`
Returns the full array.

---

2) Get one user (path param)

GET `/users/:id`
Return the user with that `id` (or `{}` / 404—your choice).

Read about res.status() function

---

3) Create user

POST `/users`
Body: `{ "name": "..." }`
Server adds:

- `id` (auto increment: max existing id + 1)

Returns the created user.

---

4) Update user

PUT `/users/:id`
Body: `{ "name": "new name" }`
Replace the user's name.

Returns the updated user.

---

5) Delete user

DELETE `/users/:id`
Remove the user and return a simple message like:
`{ "deleted": true }`

FS.promises Requirements

In other file, You must implement:

- `readUsersFromFile()` → reads `users.json` and returns array (if file missing, start with `[]`)
- `saveUsersToFile(users)` → writes back to `users.json`

What students submit (min)

- `app.js` (single file server)
- `users.json` (starter file with 2 users)
- `README.md` with 5 example requests (curl or Postman screenshots)