# Week 4 Assignment: “Employee Manager CLI + Server”

Create a mini Node.js application that manages employee records from both a **CLI (Command Line Interface)** and a **web API**.
Use **ES Modules**, **fs.promises**, **async/await**, **streams**, and **error handling** patterns learned so far.

---

## Functional Requirements

### 1️⃣ Data Storage

- Store employee data in `data/employees.json`.
- Each employee object:

```
{
  "id": 1,
  "name": "John Doe",
  "role": "Developer",
  "salary": 65000
}
```

- Start with an empty array if file doesn’t exist.

### 2️⃣ CLI Operations

Create `employee-cli.js` to support:

```
node employee-cli.js add "John Doe" "Developer" 65000
node employee-cli.js list
node employee-cli.js get 2
node employee-cli.js delete 3
node employee-cli.js update 2 "QA Engineer" 70000
```

Use:

- `fs.promises` for file I/O
- `path` for file resolution
- Proper **try/catch** blocks and error messages.

---

### 3️⃣ HTTP API

Create `server.js` that serves the same data with routes:

| Method | Endpoint | Description |
|---|---|---|
| GET | `/api/employees` | List all employees |
| GET | `/api/employees/:id` | Get single employee |

| Method | Endpoint | Description |
|---|---|---|
| POST | `/api/employees` | Add new employee |
| PUT | `/api/employees/:id` | Update existing |
| DELETE | `/api/employees/:id` | Delete employee |
| GET | `/api/export` | Streams JSON file as a downloadable attachment |

Use:

- `http` module
- `fs.promises` for CRUD
- `createReadStream` for `/api/export`
- Return proper HTTP status codes and JSON content types.
- Handle invalid JSON input safely.

---

## 4️⃣ Utilities Module

Create `utils/logger.js`:

```
import fs from "fs";
import path from "path";

const LOG_FILE = path.join(process.cwd(), "server.log");

export async function log(message) {
  const timestamp = new Date().toISOString();
  await fs.promises.appendFile(LOG_FILE, `[${timestamp}] ${message}\n`);
}
```

Use it in both CLI and server to log all operations.

---

## 5️⃣ Error Handling

- Handle file not found, invalid JSON, and invalid user inputs.
- Add a **global** process-level handler for unhandled rejections and exceptions.
- Return meaningful 400/404/500 responses for HTTP routes.

---

## 6️⃣ Stats

Add `/api/stats` route:

- Uses `os` module to return `{ platform, totalmem, freemem, uptime }`.