**Employee Management System – Containerized Web App**

**Student:** Prem Chander Jebastian

**Assignment:** Advanced Containers – INFO8985

**📝 Overview**

This is a containerized full-stack Employee Management System built using the **MERN stack** (MongoDB, Express, React, Node.js). It allows CRUD operations on employee data and is packaged using **Docker** and **Docker Compose** for easy orchestration. It also includes **NGINX** as a reverse proxy and is exposed securely to the internet using **Cloudflare Tunnel**.

**🔧 Technology Stack**

* **Frontend:** React
* **Backend:** Node.js with Express
* **Database:** MongoDB
* **Reverse Proxy:** NGINX
* **Containerization:** Docker & Docker Compose
* **Public Access:** Cloudflare Tunnel

**📦 Application Components**

| **Component** | **Description** |
| --- | --- |
| **Frontend** | React SPA served via NGINX |
| **Backend** | RESTful API (Express.js) |
| **Database** | MongoDB container with persistent volume |
| **Reverse Proxy** | NGINX serves frontend and proxies API calls |
| **CI/CD** | GitHub Actions (linting, testing) |
| **Deployment** | Local Docker, exposed using Cloudflare Tunnel |

**⚙️ How It Works**

* **Backend API** handles user data with /user (POST) and /user/:id (GET).
* **Frontend** consumes the API and runs on localhost:80.
* **MongoDB** stores persistent user data using a Docker volume.
* **Docker Compose** connects all services via an internal Docker network.
* **NGINX** serves the frontend and acts as a reverse proxy to the backend.
* **Cloudflare Tunnel** makes the local app publicly accessible (bonus).

**✅ Features Completed (per assignment requirements)**

* ✅ Web API for managing users
* ✅ Database with persistent volume
* ✅ Secure container config (non-root, ENV vars)
* ✅ Reverse proxy via NGINX
* ✅ API tested via Postman
* ✅ Docker Compose orchestration
* ✅ Optional Cloudflare exposure
* ✅ Clean README documentation

**🧪 API Endpoints (for Postman)**

* POST /user — Create user  
  Example:

json

CopyEdit

{

"firstName": "Leon",

"lastName": "Jebastian"

}

* GET /user/:id — Get user by ID

**🚀 Run Locally**

docker-compose up --build

Access at:  
🌐 Frontend: http://localhost  
📡 Backend: http://localhost:5050

To expose online:

cloudflared tunnel --url http://localhost:80