

User Identity Management System

Sistem manajemen identitas pengguna dengan fitur autentikasi JWT, role-based access control, dan dashboard admin.

🌐 **Live Demo:**

<https://noi.queenifyofficial.site/>

📄 Fitur

- **Autentikasi JWT** - Login dengan token berbasis JSON Web Token
- **Role-Based Access Control** - Dua role: Admin dan Employee
- **Admin Dashboard** - Kelola semua pengguna (CRUD)
- **Employee Dashboard** - Lihat profil sendiri
- **Status Management** - Aktifkan/nonaktifkan pengguna

🔧 Tech Stack

Backend

- **Node.js** + **Express 5**
- **Supabase** (PostgreSQL Database)
- **JWT** untuk autentikasi

Frontend

- **React** + **Vite**
- **Axios** untuk HTTP requests

📁 Struktur Project

```

```
user-identity-tst/
├── server.js # Entry point backend
├── package.json
├── Dockerfile
├── .env # Environment variables
└── src/
 ├── data/
 │ └── supabaseClient.js
 └── middleware/
```

```
| | └── auth.js # JWT middleware
| └── routes/
| └── users.js # API routes
└── frontend/
 ├── package.json
 ├── vite.config.js
 └── src/
 ├── App.jsx
 ├── api/
 | └── api.js
 ├── components/
 | └── UserModal.jsx
 └── pages/
 ├── Login.jsx
 ├── AdminDashboard.jsx
 └── EmployeeDashboard.jsx
```

## ## 🚀 Quick Start

### ### Prerequisites

- Node.js >= 18
- npm
- Supabase account

### ### 1. Clone repository

```
```bash
git clone <repo-url>
cd user-identity-tst
````
```

### ### 2. Setup environment

Buat file ` `.env` di root folder:

```
```env
SUPABASE_URL=your_supabase_url
SUPABASE_ANON_KEY=your_supabase_anon_key
JWT_SECRET=your_jwt_secret
PORT=3040
````
```

```
3. Install dependencies
```bash
# Backend
npm install

# Frontend
cd frontend
npm install
```

4. Run development
```bash
# Terminal 1 - Backend
npm run dev

# Terminal 2 - Frontend
cd frontend
npm run dev
```

5. Build for production
```bash
# Build frontend
npm run build

# Run production
NODE_ENV=production node server.js
```

🚀 Docker Deployment

```bash
# Build image
docker build -t user-identity .

# Run container
docker run -d \
--name user-identity \

```

```

-p 3040:3040 \
-e SUPABASE_URL=your_url \
-e SUPABASE_ANON_KEY=your_key \
-e JWT_SECRET=your_secret \
-e NODE_ENV=production \
user-identity
```

🔍 API Endpoints

Authentication

Method	Endpoint	Description
POST	`/api/auth/login`	Login user

Users (Admin only)

Method	Endpoint	Description
GET	`/api/users`	Get all users
GET	`/api/users/:id`	Get user by ID
POST	`/api/users`	Create new user
PATCH	`/api/users/:id`	Update user
PATCH	`/api/users/:id/status`	Update user status

Health Check

Method	Endpoint	Description
GET	`/health`	Check API status

🔒 API Usage Examples

Login

```bash
curl -X POST https://noi.queenifyofficial.site/api/auth/login \
-H "Content-Type: application/json" \
-d '{"email":"admin@mail.com","password":"admin123"}'
```

Get Users (with token)

```

```
```bash
curl https://noi.queenifyofficial.site/api/users \
-H "Authorization: Bearer YOUR_TOKEN"
```

Create User
```bash
curl -X POST https://noi.queenifyofficial.site/api/users \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_TOKEN" \
-d '{"name": "New
User", "email": "new@mail.com", "password": "pass123", "role": "employee"}'
```

🧑 Demo Credentials

Role	Email	Password
Admin	admin@mail.com	admin123
Employee	naila@mail.com	user123

📊 Database Schema

```sql
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    role VARCHAR(50) DEFAULT 'employee',
    status VARCHAR(50) DEFAULT 'active',
    created_at TIMESTAMP DEFAULT NOW()
);
```

📄 License

MIT License
```

**## 🧑 Author**

Developed for User Identity Management Testing