

# Monorepo with Authentication Service and Backend Application

---

This repository contains two projects:

- **login-manager**: A Node.js application that serves as a central authentication service.
- **backend\_node**: A Node.js application that relies on the **login-manager** for authentication.

## How it Works

The authentication flow is based on redirection and JSON Web Tokens (JWTs):

1. When a user tries to access a protected route in the **backend\_node** application, they are redirected to the login page of the **login-manager** application.
2. The user logs in or registers using the **login-manager**'s interface.
3. Upon successful authentication, the **login-manager** generates a JWT and redirects the user back to the **backend\_node** application.
4. The **backend\_node** application receives the JWT, stores it in a secure, HTTP-only cookie, and grants the user access to the protected routes.

## Setup and Installation

### 1. Prerequisites

- Node.js and npm
- MySQL

### 2. **login-manager** Setup

1. **Navigate to the **login-manager** directory:**

```
cd login-manager
```

2. **Install dependencies:**

```
npm install
```

3. **Set up the database:** Create a database in your MySQL server and execute the following SQL query to create the **users** table:

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  email VARCHAR(255) NOT NULL UNIQUE,
```

```
password VARCHAR(255) NOT NULL,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

4. **Configure environment variables:** Create a `.env` file in the `login-manager` directory and add the following variables:

```
DB_HOST=localhost  
DB_USER=your_db_user  
DB_PASS=your_db_password  
DB_NAME=your_db_name  
JWT_SECRET=your_super_secret_jwt_key
```

### 3. `backend_node` Setup

1. **Navigate to the `backend_node` directory:**

```
cd backend_node
```

2. **Install dependencies:**

```
npm install
```

3. **Configure environment variables:** Create a `.env` file in the `backend_node` directory and add the following variables. **Make sure to use the same `JWT_SECRET` as in the `login-manager` project.**

```
DB_HOST=localhost  
DB_USER=your_db_user  
DB_PASS=your_db_password  
DB_NAME=your_other_db_name  
JWT_SECRET=your_super_secret_jwt_key
```

## Running the Applications

1. **Start the `login-manager` application:** Open a terminal, navigate to the `login-manager` directory, and run:

```
npm run dev
```

The authentication service will be running at `http://localhost:5000`.

2. **Start the `backend_node` application:** Open another terminal, navigate to the `backend_node` directory, and run:

```
npm start
```

The backend application will be running at <http://localhost:3000>.

## How to Use

1. Open your browser and go to <http://localhost:3000>.
2. You will be automatically redirected to the login page of the authentication service (<http://localhost:5000/users/login>).
3. Log in or create a new account.
4. After a successful login, you will be redirected back to the `backend_node` application's dashboard (<http://localhost:3000/dashboard>), and you will be authenticated.