

Setup Instructions

1. Prerequisites

- Node.js (v16 or above recommended)
- PostgreSQL installed and running
- npm or yarn package manager
- Git (optional, for cloning repo)

2. Clone the repository (if applicable)

- `git clone https://github.com/your-repo/user-access-management.git`
`cd user-access-management`

3. Install dependencies

- `npm install`

4. Configure environment variables

- Create a `.env` file in the root folder with the following variables:

- `PORT=3000`

`DB_HOST=localhost`

`DB_PORT=5432`

`DB_USERNAME=db_username`

`DB_PASSWORD=db_password`

`DB_NAME=user_access_db`

`JWT_SECRET=jwt_secret_key`

`CLIENT_ORIGIN=http://localhost:5173`

5. Setup the database

- Create the database manually or via `psql`:

`CREATE DATABASE user_access_db;`

Run TypeORM migrations to create tables:

`npx typeorm migration:run`

6. Start the backend server

- `npm run dev`

Server will start on `http://localhost:3000` by default.

7. Frontend setup:-

Login page-

The screenshot shows the Login page of the Leucine application. The page has a purple header bar with the text "Leucine" on the left and "Login" and "Signup" on the right. The main content area is light gray. In the center, there is a white card titled "Login". Inside the card, there are two input fields: the first is labeled "Username" and contains the text "adminuser1"; the second is labeled "Password" and contains a series of dots. Below these fields is a purple button labeled "Login".

Signup page-

The screenshot shows the Signup page of the Leucine application. The page has a purple header bar with the text "Leucine" on the left and "Login" and "Signup" on the right. The main content area is light gray. In the center, there is a white card titled "Sign Up". Inside the card, there are four input fields: "Username", "Email", "Password", and "Employee" (which is a dropdown menu). Below these fields is a purple button labeled "Sign Up".

Request Access-

The screenshot shows the Request Access page of the Leucine application. The page has a purple header bar with the text "Leucine" on the left and "Request Access" and "Logout" on the right. The main content area is light gray. In the center, there is a white card titled "Request Access". Inside the card, there are three dropdown menus: "Software:" (with the text "--Select Software--"), "Access Type:" (with the text "--Select Access Type--"), and "Reason:" (with a text area). Below these fields is a purple button labeled "Submit Request". Below the card, there is a green button labeled "View All Requests".

Review All Previous Request-

Refresh All Requests

All Previous Requests

Software	Access Type	Reason	Status
Visual Studio Code	Read	Need access for daily tasks	Approved
Visual Studio Code	Read	Admin testing create request	Approved
Visual Studio Code	Read	qwerty	Approved
Visual Studio	Write	qwerty	Approved
Visual Studio	Write	zxcvbn	Approved
Visual Studio	Write	work	Approved
Visual Studio	Admin	qwerty	Rejected
Visual Studio Code	Write	sdfg	Pending
Visual Studio Code	Read	sdfgh	Pending
Visual Studio Code	Write	dfl	Rejected
Visual Studio Code	Read	brj	Approved
Visual Studio	Admin	qwerty	Pending

Pending Request page-

Leucine

Pending Requests

Logout

Pending Requests

Show All Requests

ID	User	Software	Access Type	Reason	Status	Actions
10	employeuser	Visual Studio Code	Write	sdfg	Pending	<div>ApproveReject</div>
11	Prashant	Visual Studio Code	Read	sdfgh	Pending	<div>ApproveReject</div>
12	Prashant1	Visual Studio	Admin	qwerty	Pending	<div>ApproveReject</div>

All Request-

All Requests

Show Pending Requests

ID	User	Software	Access Type	Reason	Status	Actions
1	employeuser	Visual Studio Code	Read	Need access for daily tasks	Approved	Read-only
2	adminuser1	Visual Studio Code	Read	Admin testing create request	Approved	Read-only
3	employeuser	Visual Studio Code	Read	qwerty	Approved	Read-only
4	employeuser	Visual Studio	Write	qwerty	Approved	Read-only
5	adminuser1	Visual Studio	Write	zxcvbn	Approved	Read-only
7	Prashant	Visual Studio	Write	work	Approved	Read-only
6	employeuser	Visual Studio	Admin	qwerty	Rejected	Read-only
10	employeuser	Visual Studio Code	Write	sdfg	Pending	<div>ApproveReject</div>
11	Prashant	Visual Studio Code	Read	sdfgh	Pending	<div>ApproveReject</div>
8	employeuser	Visual Studio Code	Write	dfl	Rejected	Read-only
9	Prashant	Visual Studio Code	Read	brj	Approved	Read-only
12	Prashant1	Visual Studio	Admin	qwerty	Pending	<div>ApproveReject</div>

Create Software page-

Leucine

Create SoftwareRequest AccessPending RequestsLogout

Create New Software

Name:

Description:

Access Levels:

☐ Read

☐ Write

☐ Admin

Create Software

API Documentation

- Base URL
<http://localhost:3000/api>

1. Auth Endpoints

Method	Endpoint	Description	Request Body	Response
POST	/auth/signup	Register new user	{ username, password }	{ message, userId }
POST	/auth/login	Login and get JWT	{ username, password }	{ token, role, username }

2. Software Endpoints (Admin only)

Method	Endpoint	Description	Request Body	Response
POST	/software	Create new software	{ name, description, accessLevels: [] }	{ message, softwareId }
GET	/software	List all software	None	[{ id, name, description, accessLevels }]

3. Access Requests (Employee & Manager)

Method	Endpoint	Description	Request Body	Response
POST	/requests	Submit new access request	{ softwareId, accessType, reason }	{ message, requestId }

Database Schema

Users table

```
CREATE TABLE "user" (  
  id SERIAL PRIMARY KEY,  
  username VARCHAR(255) UNIQUE NOT NULL,  
  password VARCHAR(255) NOT NULL,  
  role VARCHAR(20) NOT NULL CHECK (role IN ('Employee', 'Manager', 'Admin'))  
);
```

Software table

```
CREATE TABLE software (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  description TEXT NOT NULL,  
  access_levels TEXT[] NOT NULL -- Array of strings e.g. ['Read', 'Write', 'Admin']  
);
```

Requests table

```
CREATE TABLE request (  
  id SERIAL PRIMARY KEY,  
  user_id INTEGER NOT NULL REFERENCES "user"(id) ON DELETE CASCADE,  
  software_id INTEGER NOT NULL REFERENCES software(id) ON DELETE CASCADE,  
  access_type VARCHAR(20) NOT NULL CHECK (access_type IN ('Read', 'Write', 'Admin')),  
  reason TEXT NOT NULL,  
  status VARCHAR(20) NOT NULL CHECK (status IN ('Pending', 'Approved', 'Rejected'))  
  DEFAULT 'Pending'  
);
```