📘 BERM: Budget and Expense Register & Monitoring System prototype

# 🔧 Technology Stack

|  |  |
| --- | --- |
| Layer | Technology |
| Backend | Node.js (Express) |
| Frontend | HTML, CSS, JavaScript |
| Database | MySQL |
| Auth System | Hybrid (LDAP for login, SQL for roles) |
| LDAP Server | OpenLDAP (Dockerized) |

# 👥 User Roles and Permissions

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Authentication | Privileges | Access Level |
| Admin | LDAP + SQL | Add Users & Budget Heads, Approve All Requests, View Reports | Full Control |
| Manager | LDAP + SQL | Approve/Reject Budget & Change Requests | Moderate Control |
| User | LDAP + SQL | Submit Budget and Change Requests | Submit-only Access |

# 🔐 Default LDAP User Credentials

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Username | Password | Role |
| Admin User | adminuser | admin123 | Admin |
| Manager User | manageruser | manager123 | Manager |
| Normal User | normaluser | user123 | User |

# 🧩 System Features

## ✅ Login / Logout (Hybrid Auth)

Login using LDAP credentials, roles fetched from SQL DB.

## ✅ Admin Functions

Can add users, add budget heads, and approve or reject all requests.

## ✅ Manager Functions

Can approve or reject budget and change requests submitted by users.

## ✅ User Functions

Can submit budget requests and change requests, and view their statuses.

## ✅ Reporting

Admins and managers can view summaries of requests with filters and status.

# ⚙️ LDAP Setup Instructions (Docker)

1. Start the LDAP container:

docker run -d --name ldap-server -p 389:389 \  
 -e LDAP\_ORGANISATION="Test Org" \  
 -e LDAP\_DOMAIN="testorg.com" \  
 -e LDAP\_ADMIN\_PASSWORD="admin123" \  
 osixia/openldap:1.5.0

2. Add users from `add-users.ldif`:

docker cp add-users.ldif ldap-server:/add-users.ldif  
docker exec -it ldap-server ldapadd -x -D "cn=admin,dc=testorg,dc=com" -w admin123 -f /add-users.ldif

# 🚀 How to Run the Project

1. Set up MySQL, create a database named `berm`, and run the `schema.sql` script.

2. Start the LDAP container and add users using the above instructions.

3. Install Node.js dependencies from the `backend` folder:

npm install

4. Run the Node.js server:

node server.js

5. Open your browser and go to <http://localhost:3000>

ER diagram for the database for prototype

