

# EnergeX AI

This Project is composed of:

- **Lumen API (PHP)** – User authentication & posts CRUD with JWT.
- **Node.js Service (TypeScript)** – Redis caching layer for performance.
- **MySQL Database** – Stores users & posts.
- **Redis** – Caches post queries.
- **React Frontend** – Simple UI for login, registration, and posts.
- **Docker Compose** – Orchestrates all services.
- **GitHub Actions** – Automated testing pipeline.

Loom Recording :

<https://www.loom.com/share/ea3b5fc685a64cb1816854eb9097c649?sid=da38a412-5dca-4974-917d-ef1cc37cc281>

<https://www.loom.com/share/13d15e8b73fd4d09abced8c56dfc40d9?sid=133b8ba3-36ff-4f45-b11f-ed67ad11babe>

## Setup Instructions

### 1. Clone the repository

<https://github.com/Yaswanth-Yash-01/Fullstack-Project-EnergeX.git>  
cd EnergeX-AI-Hiring-Test

### 2. Environment variables

Backend services has a .env Copy and configure:

#### Lumen API:

```
cp backend/lumen-backend/.env.example backend/lumen-backend/.env  
php artisan key:generate  
php artisan jwt:secret
```

## **Node.js Cache:**

```
cp backend/node-cache/.env
```

## **Frontend**

I have not included any env file (Because it is just simple frontend)

**Note :** Replace the values above with your own database credentials.

## **3. Run locally (without Docker)**

### **Lumen**

```
cd backend/lumen-backend  
composer install  
php artisan migrate  
php -S 0.0.0.0:8000 -t public
```

### **Node.js Cache**

```
cd backend/node-cache  
npm install  
npm run build  
npm run start
```

### **Frontend**

```
cd frontend/reactfrontend  
npm install  
npm run dev
```

## **4. Run with Docker**

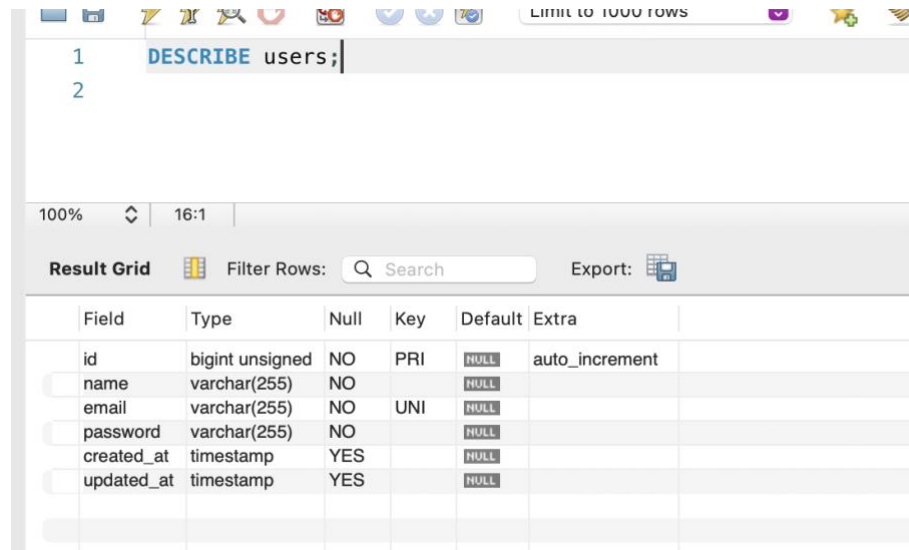
```
docker compose up --build
```

Services:

- **Lumen API** → <http://localhost:8000/api>
- **Node.js Cache API** → <http://localhost:3000/cache/>
- **Frontend** → <http://localhost:5173>

## Database Schema

### Users:



1 DESCRIBE users;

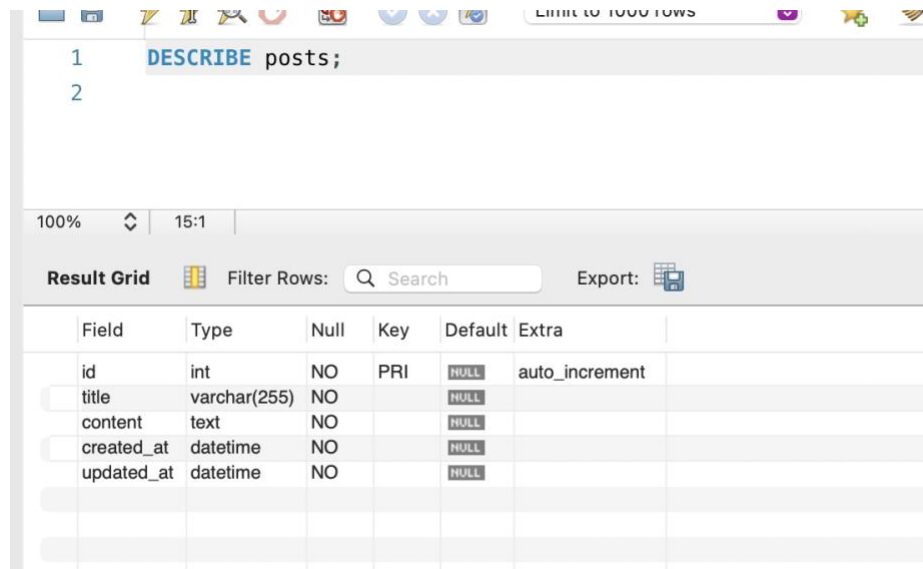
2

100% 16:1

Result Grid Filter Rows: Search Export:

	Field	Type	Null	Key	Default	Extra
	id	bigint unsigned	NO	PRI	NULL	auto_increment
	name	varchar(255)	NO		NULL	
	email	varchar(255)	NO	UNI	NULL	
	password	varchar(255)	NO		NULL	
	created_at	timestamp	YES		NULL	
	updated_at	timestamp	YES		NULL	

### Posts



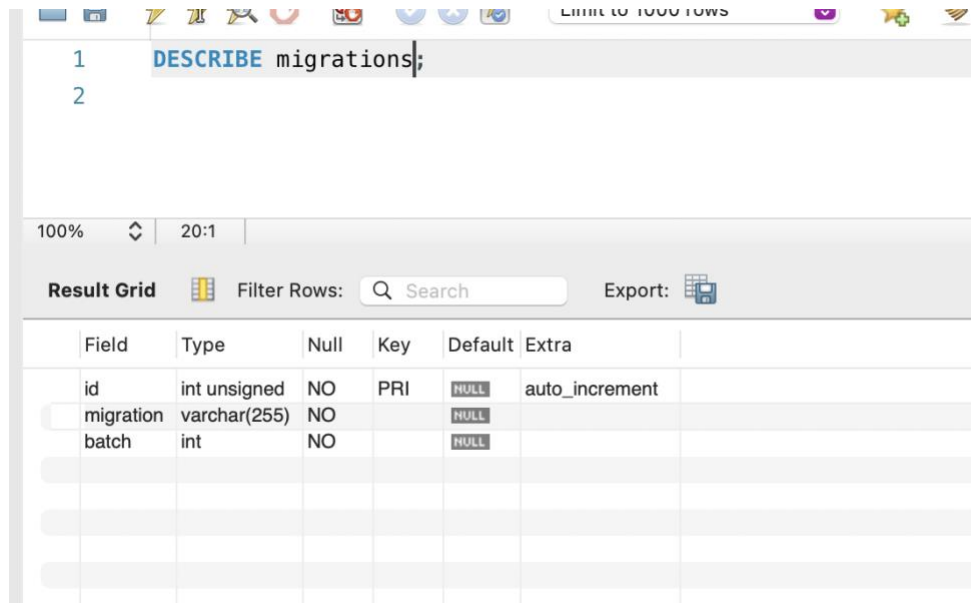
1 DESCRIBE posts;

2

100% 15:1

Result Grid Filter Rows: Search Export:

	Field	Type	Null	Key	Default	Extra
	id	int	NO	PRI	NULL	auto_increment
	title	varchar(255)	NO		NULL	
	content	text	NO		NULL	
	created_at	datetime	NO		NULL	
	updated_at	datetime	NO		NULL	



## Authentication

POST /api/register → Register user

POST /api/login → Returns JWT token

Include JWT in Authorization: Bearer <token>

## API Endpoints

### Lumen API

Method	Endpoint	Description
POST	/api/register	Register user
POST	/api/login	Authenticate + JWT token
GET	/api/posts	Fetch all posts (cached)
POST	/api/posts	Create new post
GET	/api/posts/{id}	Fetch single post (cached)
PUT	/api/posts/{id}	Update the post
DELETE	/api/posts/{id}	Deleting the post

## Node.js Cache API

Method	Endpoint	Description
GET	/cache/posts	Get cached posts
GET	/cache/posts/{id}	Get cached single post

## Testing

### Lumen

```
cd backend/lumen-backend  
./vendor/bin/phpunit --testdox
```

### Node.js

```
cd backend/node-cache  
npm test
```

### Frontend

```
cd frontend/reactfrontend  
npm test  
CI runs these automatically on each push (see .github/workflows/ci.yml).
```