

# Advance Web Development Project



Submitted By: Naresh Kumar Jaiswal  
Submitted To: Thomas Broussard Sir

# Application Architecture

## 1. Backend: -



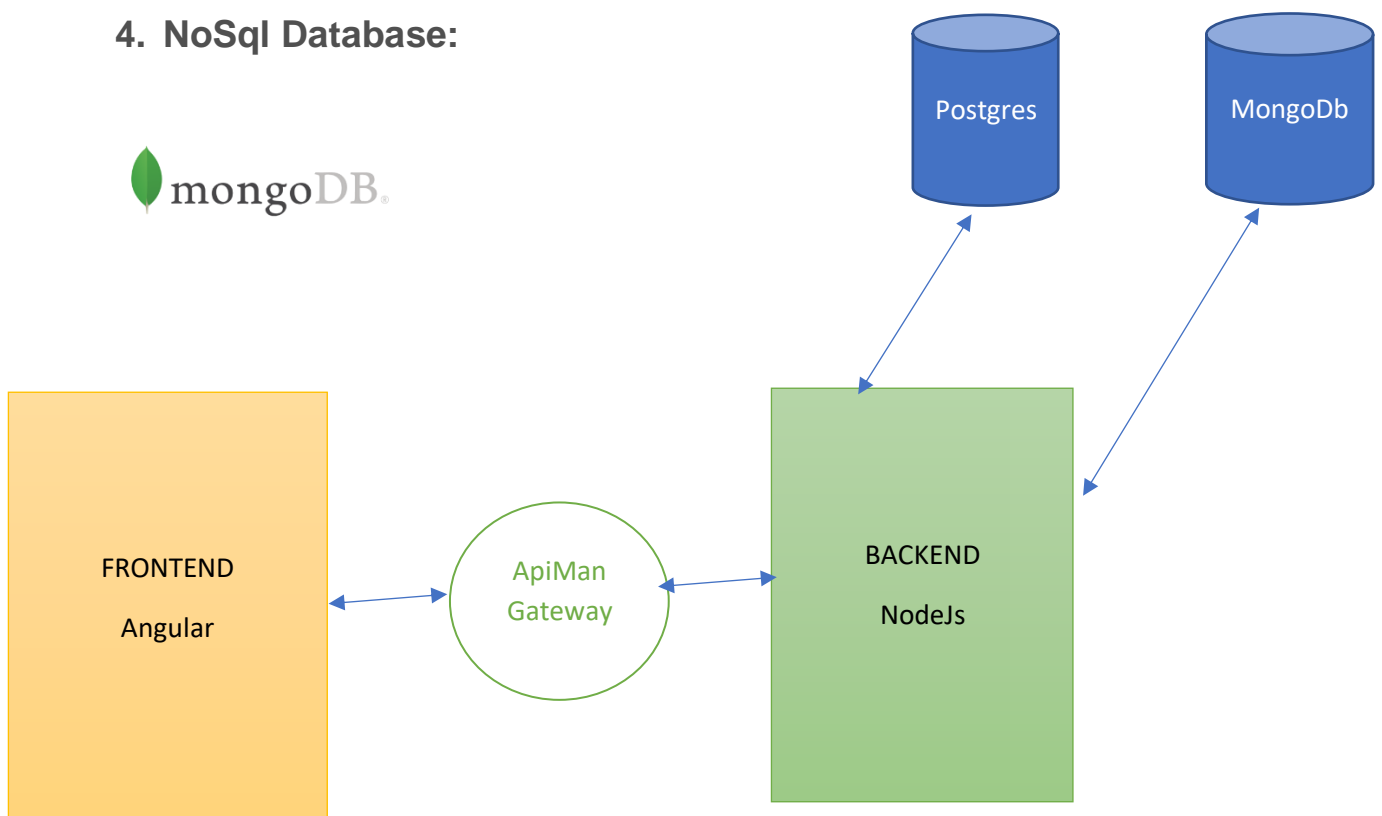
## 2. Frontend: -



## 3. Relational Database

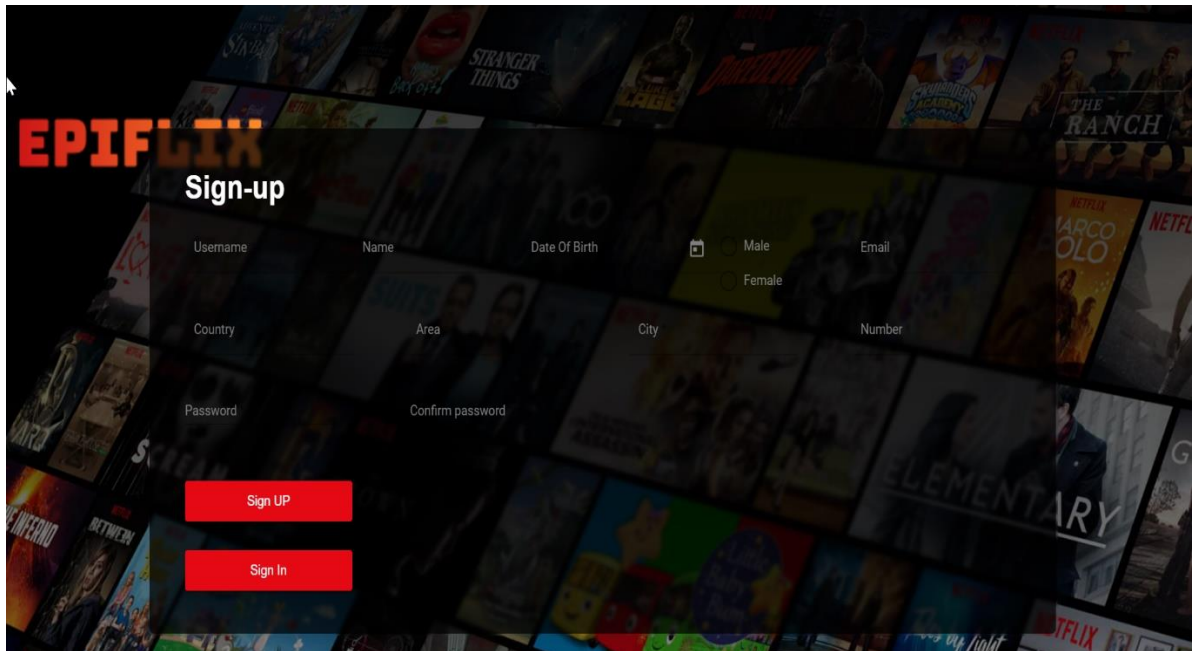


## 4. NoSql Database:



## Frontend Application Layout

### 1. Register User



The registration form is titled "EPIFLIX Sign-up" and is set against a background of various movie and TV show posters. The form includes the following fields: Username, Name, Date Of Birth, Email, Country, Area, City, Number, Password, and Confirm password. There are two radio buttons for gender selection: Male and Female. At the bottom, there are two red buttons: "Sign UP" and "Sign In".

**EPIFLIX Sign-up**

Username    Name    Date Of Birth    ☐ Male    Email  
☐ Female

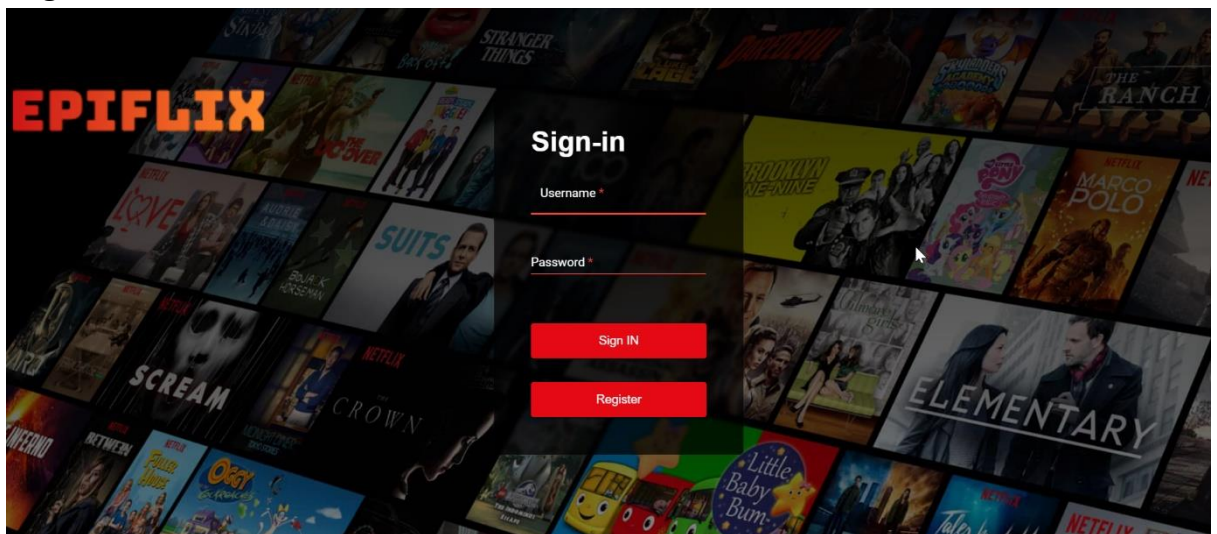
Country    Area    City    Number

Password    Confirm password

**Sign UP**

**Sign In**

### 2. Login User



The login form is titled "EPIFLIX Sign-in" and is set against the same background of movie and TV show posters. The form includes the following fields: Username and Password, both marked with an asterisk to indicate they are required. There are two red buttons at the bottom: "Sign IN" and "Register".

**EPIFLIX Sign-in**

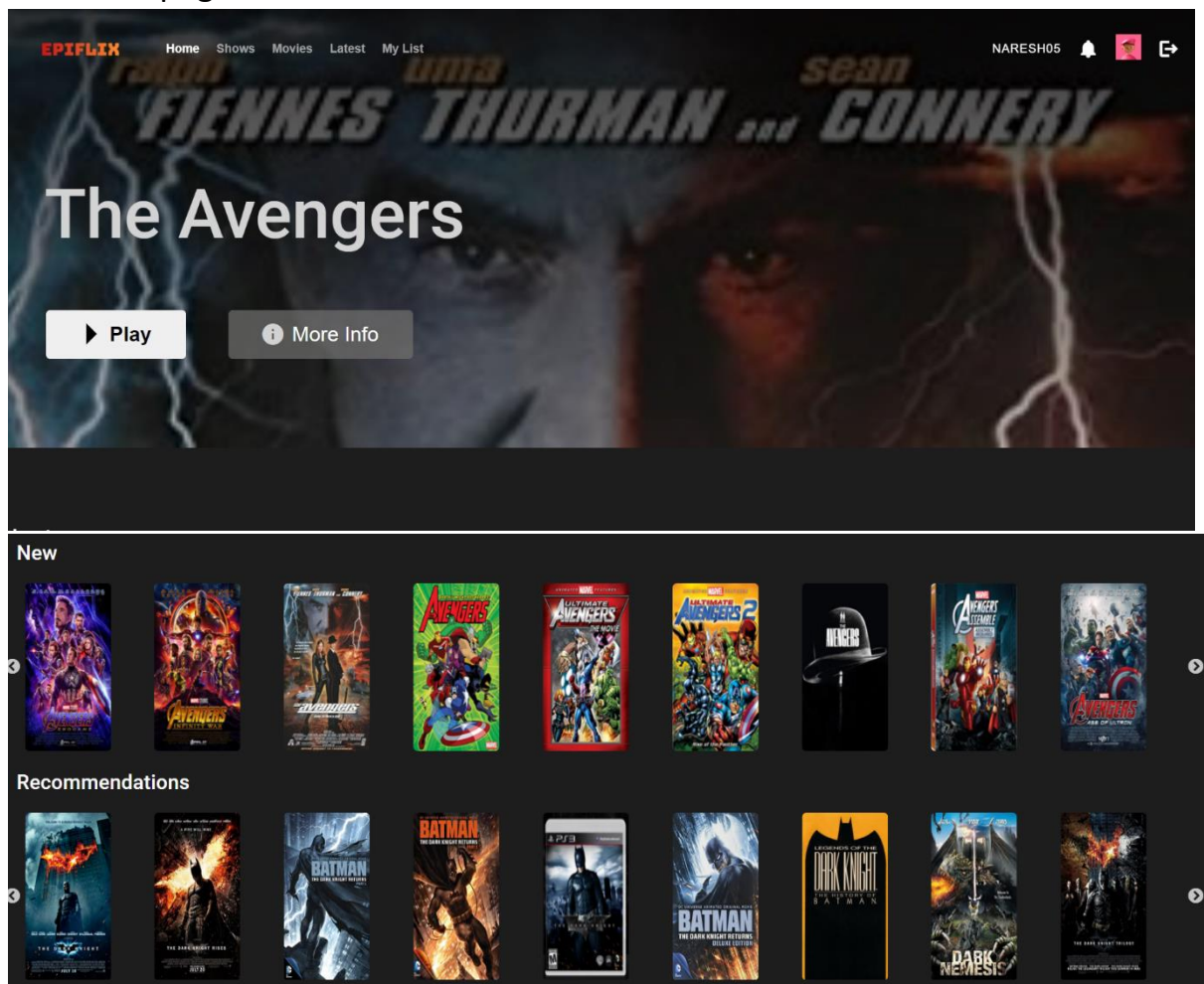
Username \*

Password \*

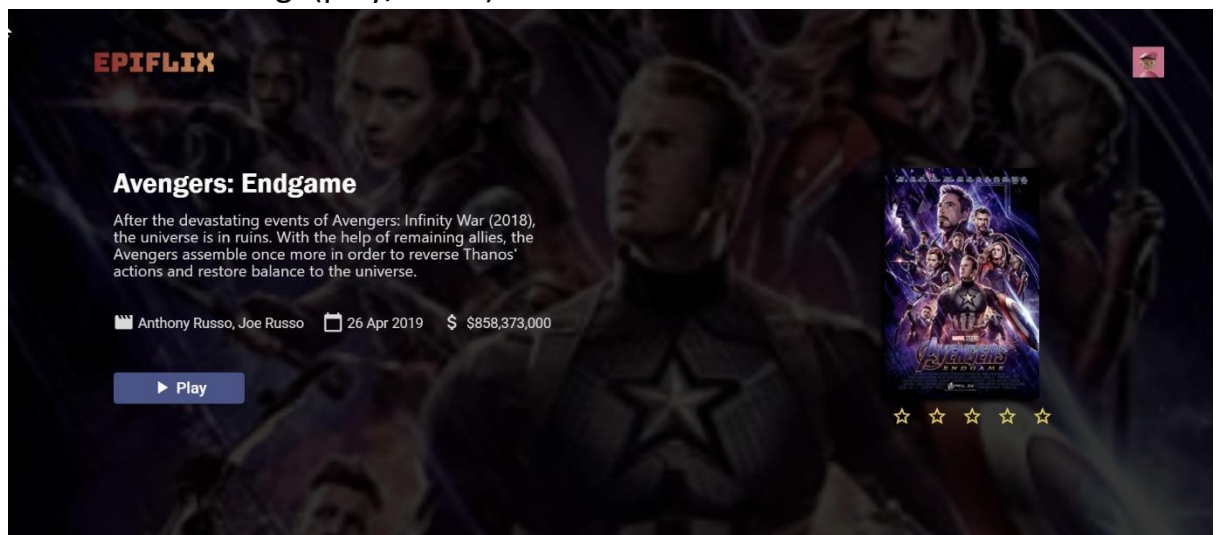
**Sign IN**

**Register**

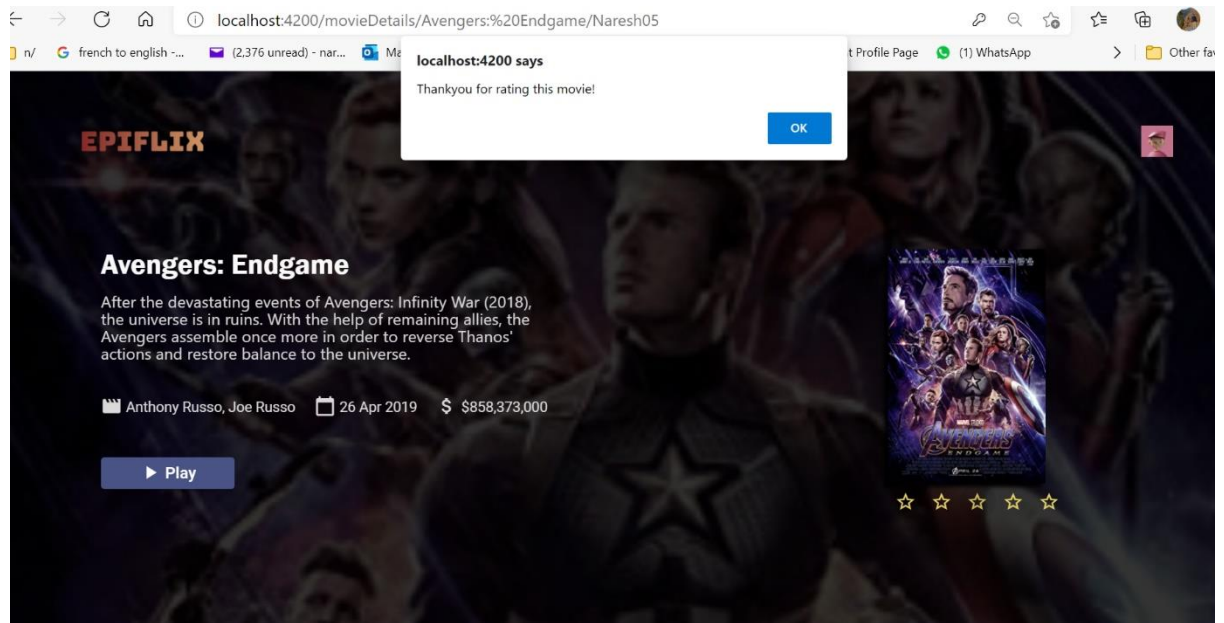
### 3. Welcome page



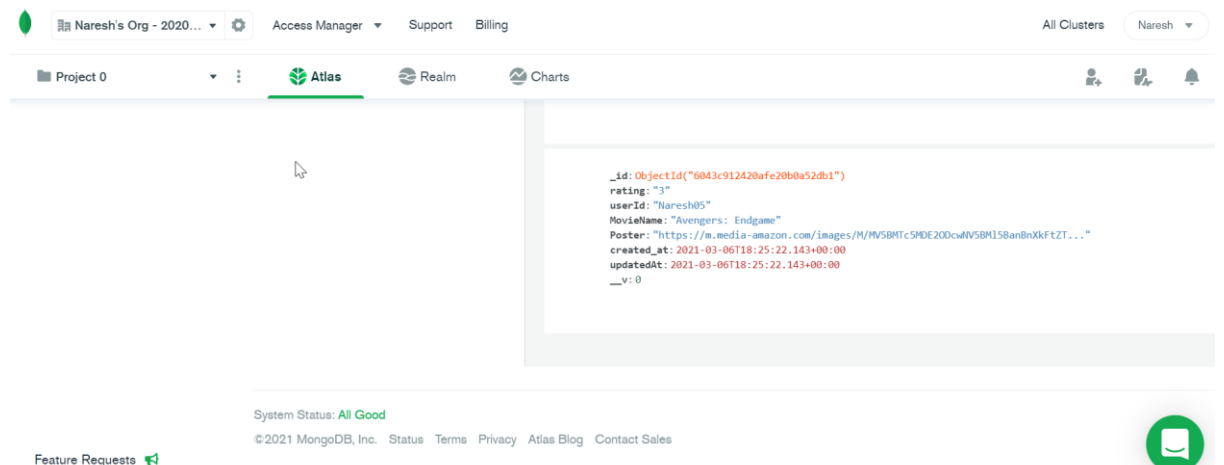
### 4. Movies Details Page(play/Mock)



## 5. Movie Rating



## 6. Rated Movies stored in Nosql Database





# Backend Nodejs Api(Postman)

## 1. Register Users

POST http://localhost:5000/users

Body (JSON):

```
{  "username": "Naresh05",  "password": "asdf@1234",  "name": "Naresh",  "birthdate": "2001-10-10",  "gender": "M",  "email": "a@gmail.com",  "country": "india",  "area": "jharkhand",  "street": "ranchi",  "city": "siridih"}
```

Status: 201 Created Time: 147 ms Size: 808 B Save Response

1 User added

## 2. Get All latest movies

GET http://localhost:5000/Movies

Status: 200 OK Time: 117 ms Size: 5.26 KB Save Response

```
[  {    "id": 13,    "title": "Avengers: Endgame",    "year": 2019,    "imdbid": "tt4154796",    "type": "movie",    "poster": "https://m.media-amazon.com/images/M/MV5BMTc5MDE2ODcwMV5BM15BanBnXkFtZTgwMzI2NzQ2MzQ0. V1_SX300.jpg",    "plot": "After the devastating events of Avengers: Infinity War (2018), the universe is in ruins. With the help of remaining allies, the Avengers assemble once more in order to reverse Thanos actions and restore balance to the universe."  },  ...]
```

## 3. Get movies details by name.

GET http://localhost:5000/Movies/The Avengers

Status: 200 OK Time: 90 ms Size: 1.54 KB Save Response

```
{  "title": "The Avengers",  "plot": "Earths mightiest heroes must come together and learn to fight as a team if they are going to stop the mischievous Loki and his alien army from enslaving humanity.",  "director": "Joss Whedon",  "released": "04 May 2012",  "poster": "https://m.media-amazon.com/images/M/MV5BYWE1NTdjOWQyTQ2Ny00Nzc5LWExYzYtNmRlOThmOTE2N2I4XkEyXkFqcGdeQXVyNjUwNzk3NDc0. V1_SX300.jpg"}
```

## 4. Give rating to the movies.

POST http://localhost:5000/MovieRating

Send Save

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies C

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▼ Beauti

```
1 {
2   "rating": "5",
3   "userId": "Naresh05",
4   "MovieName": "Ave",
5   "poster": "https://m.media-amazon.com/images/M/MV5BNBYxNjQyMjAtNTdiOS00NGYwLWFmNTAtNTNmYjU5ZGI2YTl1XkEyXkFqcGdeQXVyMTMxODk2OTU@._v1_SX300.jpg"
6 }
```

Body Cookies Headers (16) Test Results Status: 200 OK Time: 94 ms Size: 836 B Save Response

Pretty Raw Preview Visualize JSON ▼

```
1 {
2   "message": "Rating well created !"
3 }
```

## 5. Get rated movies details.

GET http://localhost:5000/MovieRating/Naresh05

Send Save

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
-----	-------	-------------	-----	-----------

Body Cookies Headers (16) Test Results Status: 200 OK Time: 116 ms Size: 1002 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼

```
1 [
2   {
3     "_id": "60413812ffeb625e08eff2a9",
4     "commentContent": "commentContent",
5     "userId": "Naresh05",
6     "MovieName": "Tarzaan",
7     "created_at": "2021-03-04T19:42:10.617Z",
8     "updated_at": "2021-03-04T19:42:10.617Z",
9     "__v": 0
10  }
11 ]
```

## 6. Recommendation movies

GET http://www.omdbapi.com/?s=The+Dark+Knight&apikey=af8b62df

Send Save

Params ● Authorization Headers (7) **Body** Pre-request Script Tests Settings Cookies Cod

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> s	The+Dark+Knight			
<input checked="" type="checkbox"/> apiKey	af8b62df			
Key	Value	Description		

Body Cookies (1) Headers (16) Test Results Status: 200 OK Time: 355 ms Size: 2.63 KB Save Response ▼

Pretty Raw Preview Visualize JSON ▼

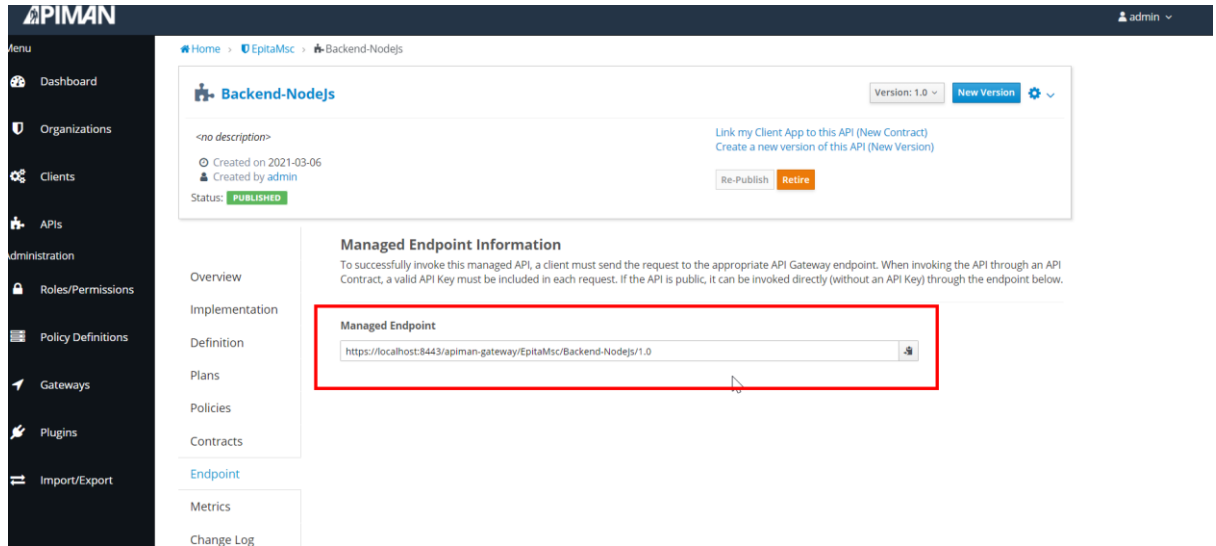
```
1 {
2   "Search": [
3     {
4       "Title": "The Dark Knight",
5       "Year": "2008",

```

# ApiMan Gateway

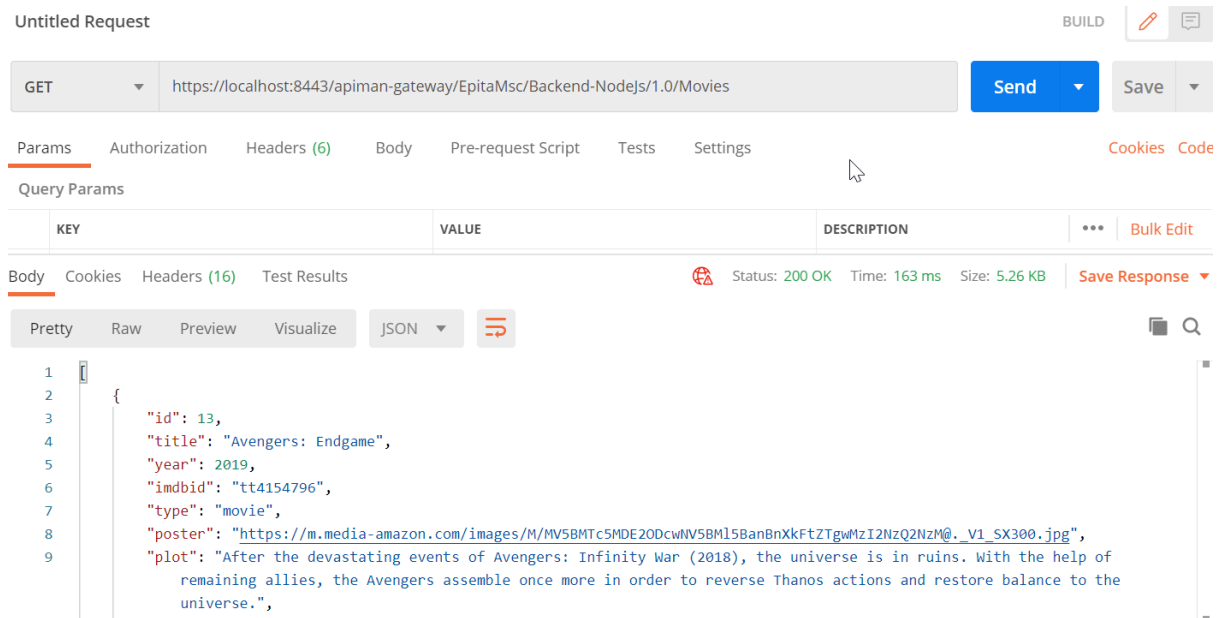
## 1. Managed EndPoint

<https://localhost:8443/apiman-gateway/EpitaMsc/Backend-NodeJs/1.0>



## 2. Sample:-

<https://localhost:8443/apiman-gateway/EpitaMsc/Backend-NodeJs/1.0/Movies>





# Database Configuration:-

## 1. MongoDB

Path- ./epita\_awt\_project\_api/.env

PORT = 5000

MONGO\_URL =

mongodb+srv://Epita\_user:info@12345@cluster0.fifto.mongodb.net  
/NodeJsConnect?retryWrites=true&w=majority

## 2. Postgres

Path 1 - .\epita\_awt\_project\_api\routes\queryLs.js

Path2 - .\epita\_awt\_project\_api\routes\movies.js

```
const pool = new Pool({  
  user: 'postgres',  
  host: 'localhost',  
  database: 'postgres',  
  password: 'info@12345',  
  port: 5432,  
})
```

```
const pool = new Pool({  
  user: 'postgres',  
  host: 'localhost',  
  database: 'postgres',  
  password: 'info@12345',  
  port: 5432,  
})
```

*THANK YOU*