

# **Amrudin Balić - Movie App**

## **Introduction:**

First things first.

I wish to thank you for giving me this honor and opportunity to prove myself in this task.

I know that my task was to make an option to follow a movie, but I modified the project a little bit and added a few of my own tweaks.

I am really looking forward to your answer and I cannot explain to you how excited I am about it.

So. Let me get to the point.

## **Project Tech Stack:**

**-Backend ( Server Side ) - Laravel/PHP**

**-Frontend ( Client Side ) - BootStrap5 , jQuery**

**-API Testing: Postman**

## **Project Setup:**

Just create Database in phpMyAdmin - Name: "hulkapps"

All tables are already defined. Just run migrations and seeders.

File structure is like in any normal Laravel app except that in public/script I added my script and css files.

## Project Explanation:

So, here is the general project explanation.

I had used Laravel to build Rest APIs and defined their endpoints inside **routes/api.php**.

I had generally used comments to explain my code and what it does.

So for the Backend side I had used logic to build an api's and on Frontend side I used mostly AJAX calls to "pin" those endpoints of my backend structure.

After making an AJAX call using jQuery library, I handled the response whether it be positive or negative via success or error callback.

To build UI's I used Bootstrap5 CSS framework. So, the app is half responsive.

So, the pages like **main** page are in background are using AJAX call to fetch all Movies that are currently available in the database.

On **the user details** page, I am retrieving username from **cookie** that is previously stored when the user is **authenticated** on **the login page**.

All the API testing is done in **Postman** App and don't worry I am going to export you everything so you can test it too.

All API routes are clearly defined and settled for you to test the API.

Project also contains "Secret Admin" panel where you have access to a list of all movies and users and you can simply delete them.

Next slide: **API Mapping | Web Routes**

## REST API MAP:

- **User API**
  - **Options:**
    - **Create User**
    - **Get All Users**
    - **Get Specific User**
    - **Update specific user**
    - **Delete User**
- **Movie API**
  - **Options:**
    - **Create Movie**
    - **Get All Movies**
    - **Get Specific Movie**
    - **Update specific Movie**
    - **Delete Movie**
    - **(!)Follow Movie**

## Authentication:

**Login Authentication:** <http://localhost:8000/api/auth/login>

## Web Routes / Avabile Pages :

### Login

- <http://localhost:8000/>
- **File : login.blade.php**

### Register

- <http://localhost:8000/register/register>
- **File : register.blade.php**

### Main

- <http://localhost:8000/main>
- **File : main.blade.php**

## User Details

- <http://localhost:8000/userDetails>
- **File : userDetails.blade.php**

## Deleted

- <http://localhost:8000/deleted>
- **File : deleted. Blade.php**

## Admin

- <http://localhost:8000/secret/admin>
- **File : admin.blade.php**

## Controllers:

1. **UserController**
2. **MovieController**
3. **AuthController**

## Models:

1. **User Model**
2. **Movies Model**

## Migrations And Tables

- **Users Table**
- **Movies Table**

## Seeders

- **User Seeder ( Two users up with the project )**
- **Movies Seeder – Faker function used to seed movies**

