**1.Database Schema Implementation**

In this project I have utilized mysql database for storing user information.I have created table for storing user information such as email and password in order authenticate user.the table contain three fields id ,email and password.

* the id field has datatype of int ,autoIncremented and it is primary key it used to identify each row uniquely;
* email and password have datatype of varchar

**2.API Implementation**

I integrated my application with TMDB(The Movie Database) database for fetching data about the movie.

* **To search for movies by title:** I used the following endpoint;

[https://api.themoviedb.org/3/search/movie?query=badboys&api\_key=apikey](https://api.themoviedb.org/3/search/movie?query=u&api_key=apikey)

Method:get

Purpose:allow user search movie by title

* **Fetching Movie Details:**For obtaining detailed information about a specific movie, including synopsis, rating, title, and poster path, I made use of the following endpoint:
* <https://api.themoviedb.org/3/movie/343611?api_key=API_KEY>
* Method:get
* Purpose:Provides comprehensive details about a particular movie based on its unique movie\_id
* **Obtaining Movie Posters:**To get poster images for movies, I used this endpoint:
* <https://image.tmdb.org/t/p/w500/b0F4vbkVIWEk849tWKL04WcLJ6S.jpg>
* Method:get;
* Purpose:to retrieve image;
* **Fetching Cast List:** to get the cast list of a specific movie, I utilized this endpoint:
* <https://api.themoviedb.org/3/movie/550/credits?api_key=apikey>
* Method:get
* Purpose:to get cast list and crew data about movie based on movie\_id;
* **Accessing Movie Trailers**: To retrieve movie trailers, I made use of this endpoint:
* <https://api.themoviedb.org/3/movie/343611/videos?api_key=apikey>
* Method:get
* Purpose:to get the trailer of movie based on movie\_id
* **Searching Movies by Genre**:to search movie by category
* <https://api.themoviedb.org/3/discover/movie?api_key=apikey&with_genres=28>
* Method:get
* Purpose:to search movie by category

1. User Authentication

**User Registration:**

When a user registers, the password they provide is securely hashed before being stored in the database. This ensures that the actual password is never stored in its original form, enhancing security.

**User Login:**

* **Authentication:** When a user attempts to log in, the system first authenticates the user's credentials, verifying that the provided username and password match those stored in the database.
* **Token Generation**: If the user's credentials are successfully authenticated, a JSON Web Token (JWT) is generated. This token serves as a secure means of authorization, granting access to protected resources.