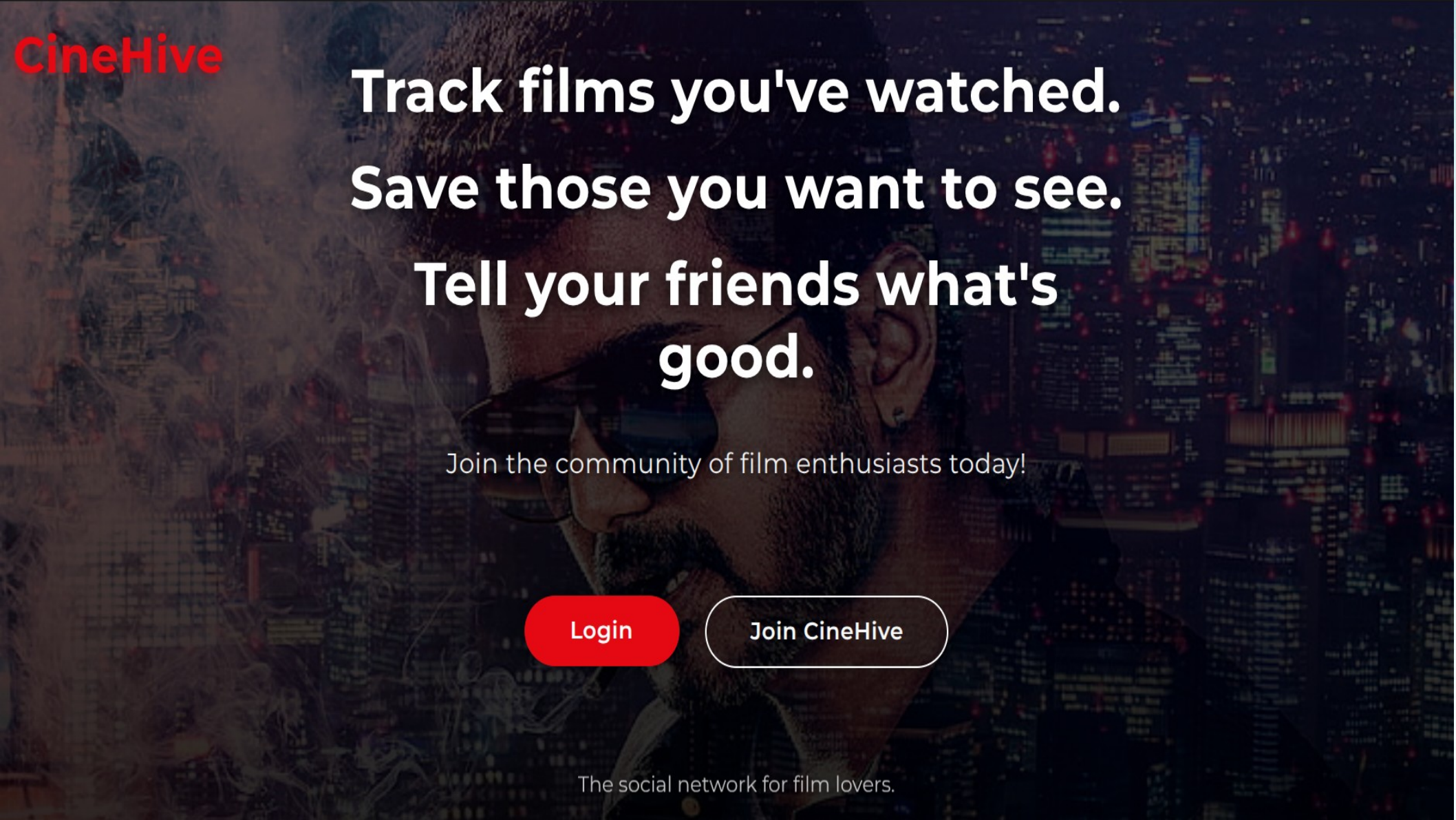
MOVIE-CATALOG MANAGEMENT

TEAM : TECHNO-VISION-KNIGHTS

PRASANNA KUMAR M	**CS23B1011**
ADHISHWAR	**CS23B1013**
DHERAJ	**CS23B1054**
SRIHARIRAM ASUVATHRAMAN	**CS23B1063**


CineHive
Track films you've watched.
Save those you want to see.
Tell your friends what's good.
Join the community of film enthusiasts today!
Login
Join CineHive
The social network for film lovers.

CineHive

Search...

Profile

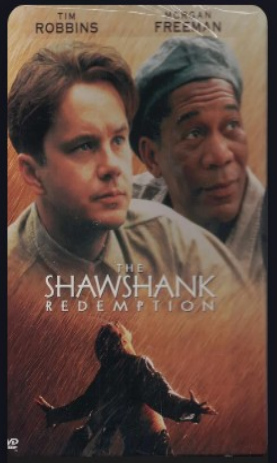
Logout

Home

Top Rated

By Language (English)

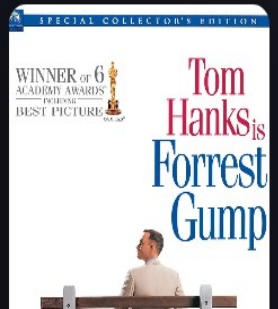
Vikram

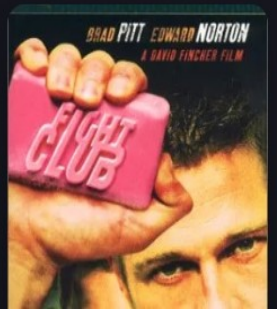


The Shawshank Redem...

The Dark Knight

C/o Kancharapalem





🛠️ TECH-STACK

REACT.JS

MYSQL

EXPRESS.JS

IMAGE HOSTING

CLOUDINARY

Why Do We Need This Website?

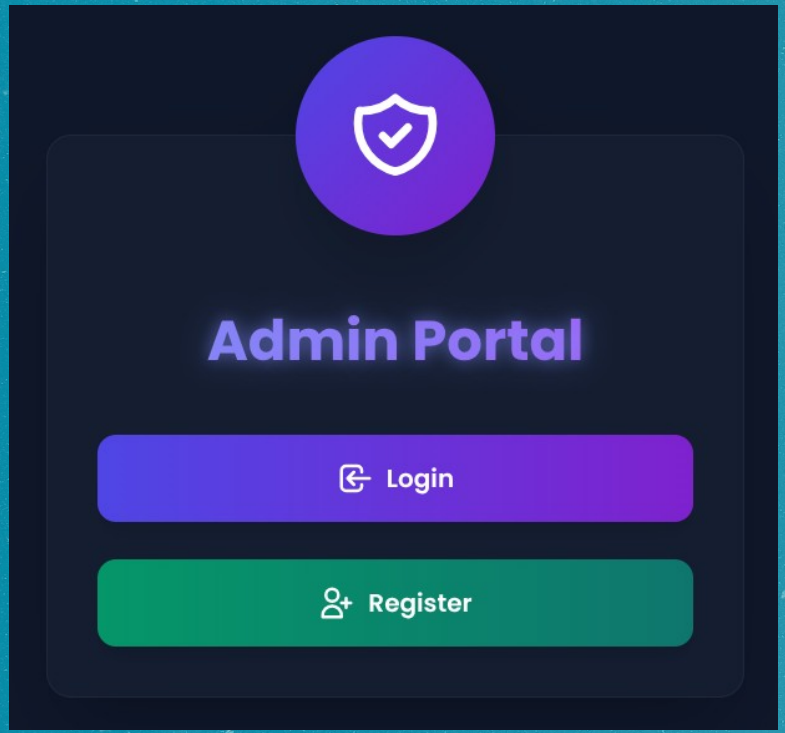
🎬 **Movie lovers**

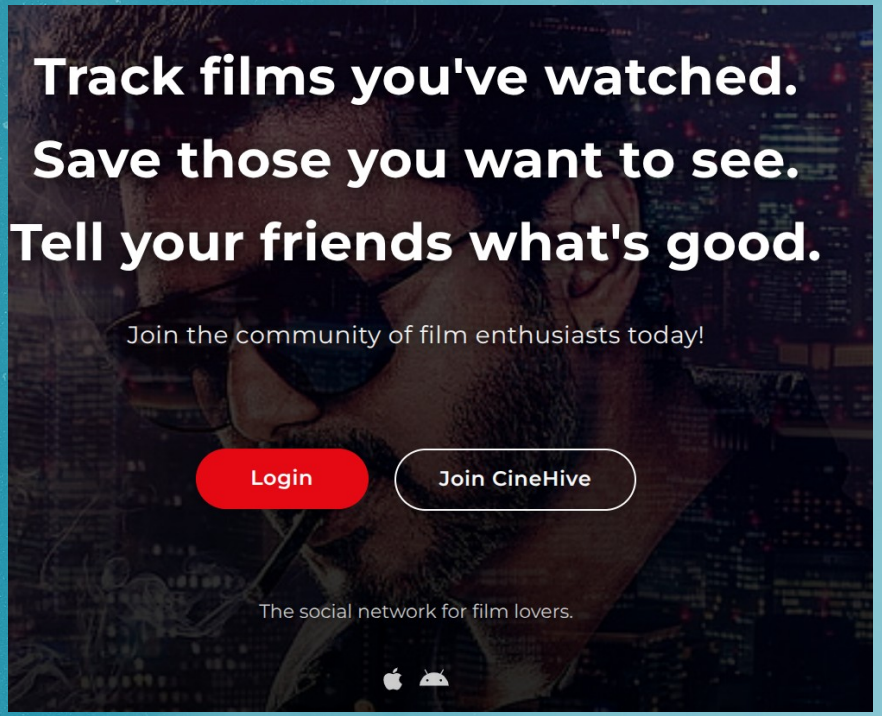
💬 **Easy access to ratings, reviews**

📌 **Helps users track their favorite or planned movies**

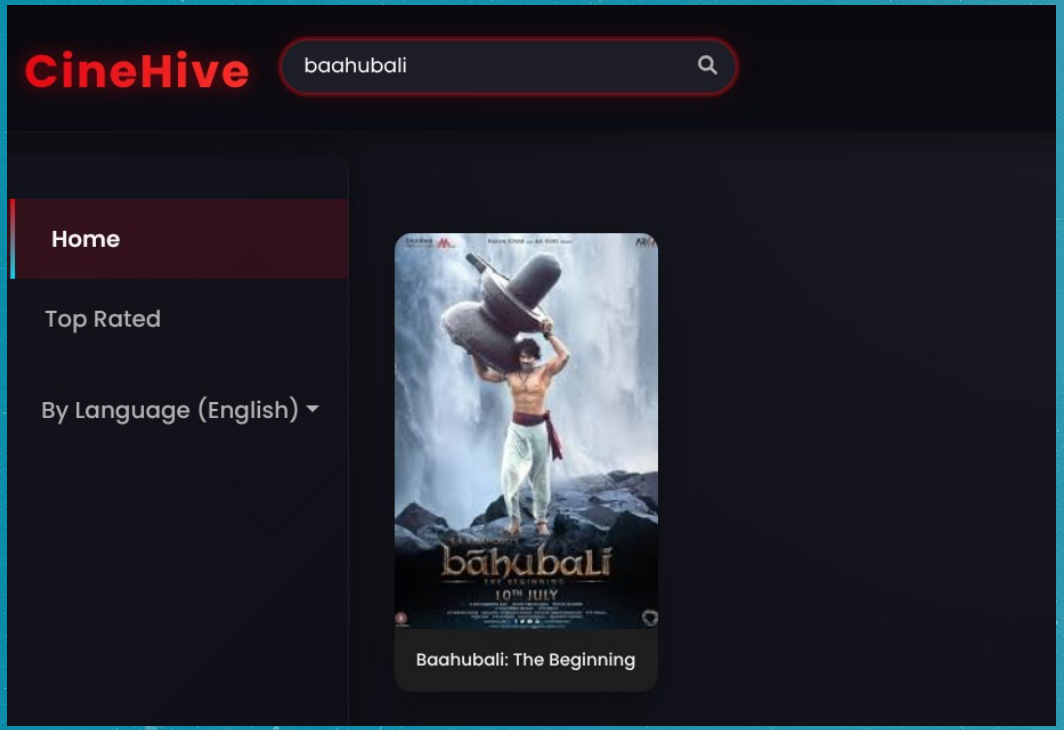
🧠 **Share opinions, discover what others think**

ADMIN AND REGULAR_USER

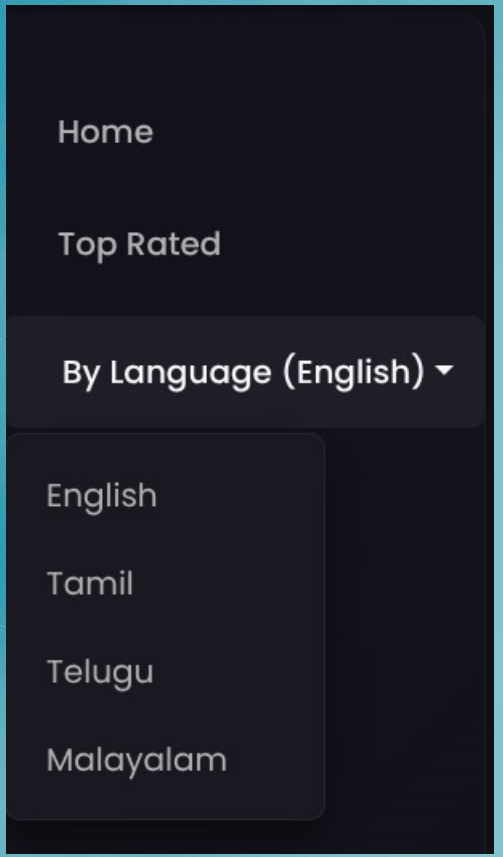




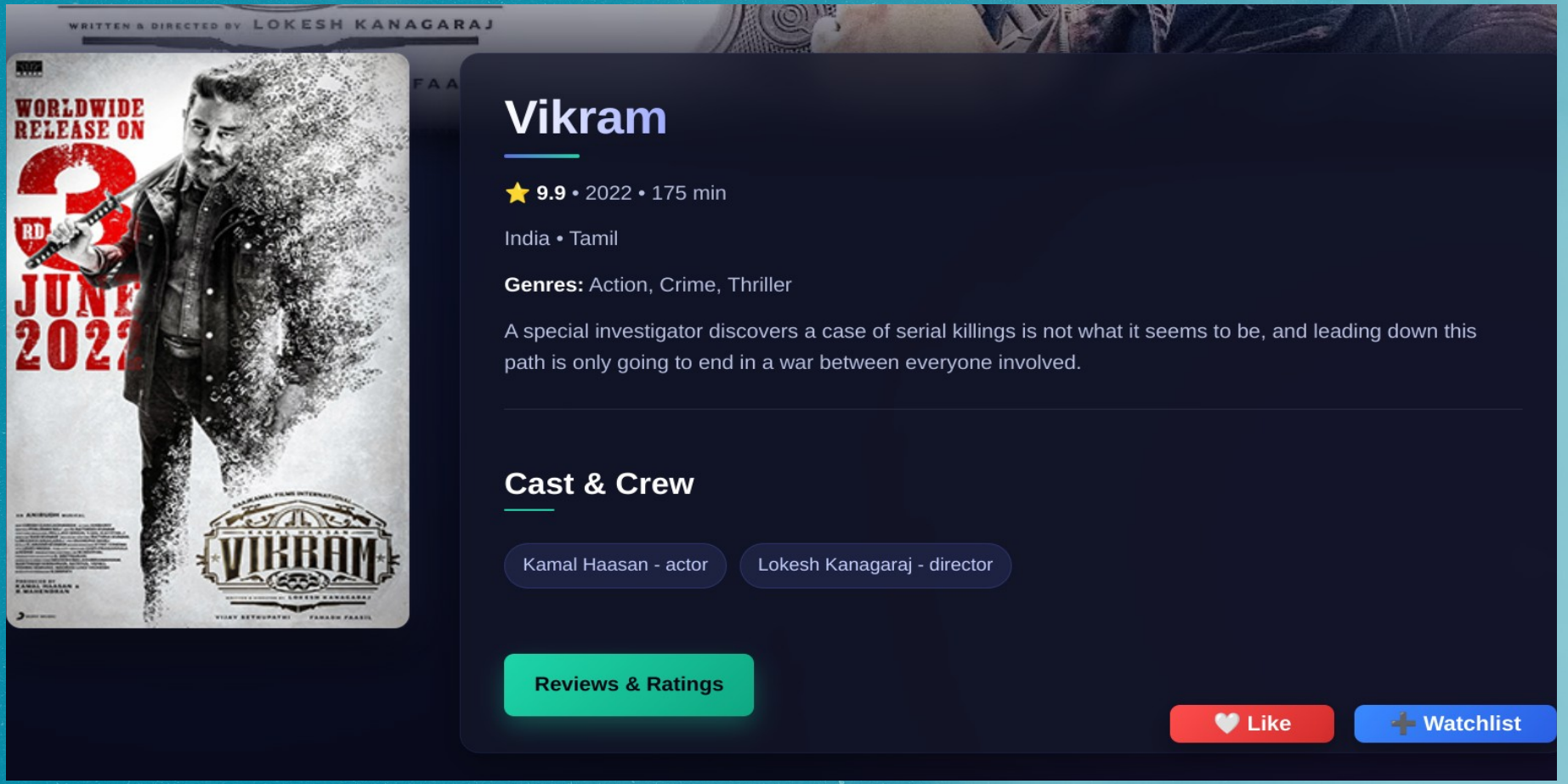
SEARCH MOVIES



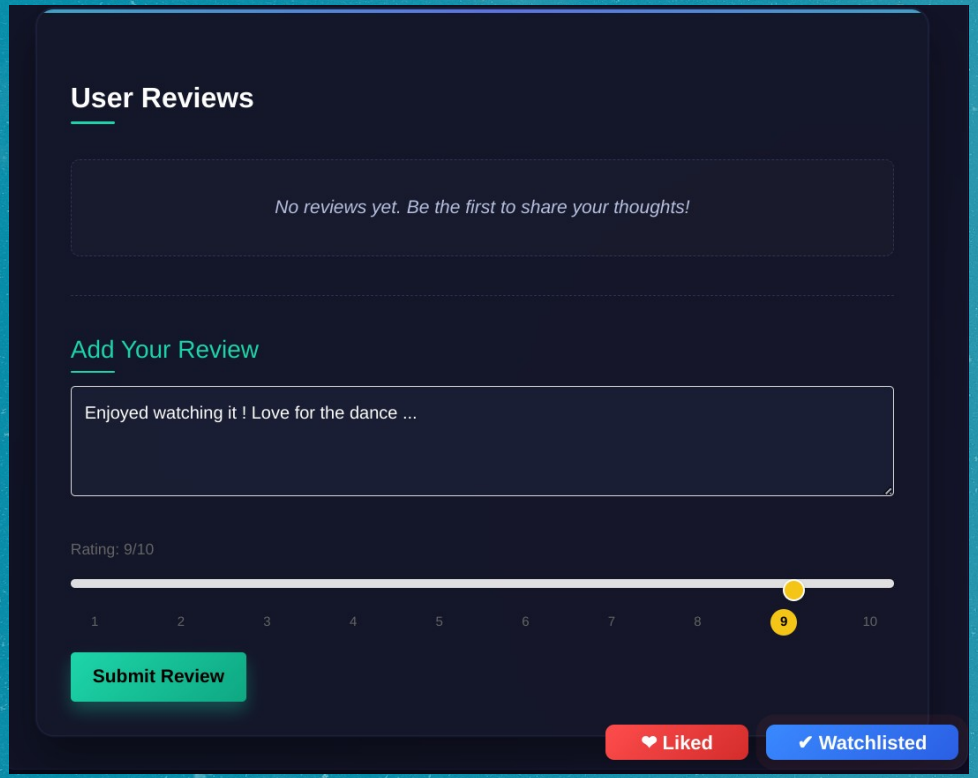
FILTERS



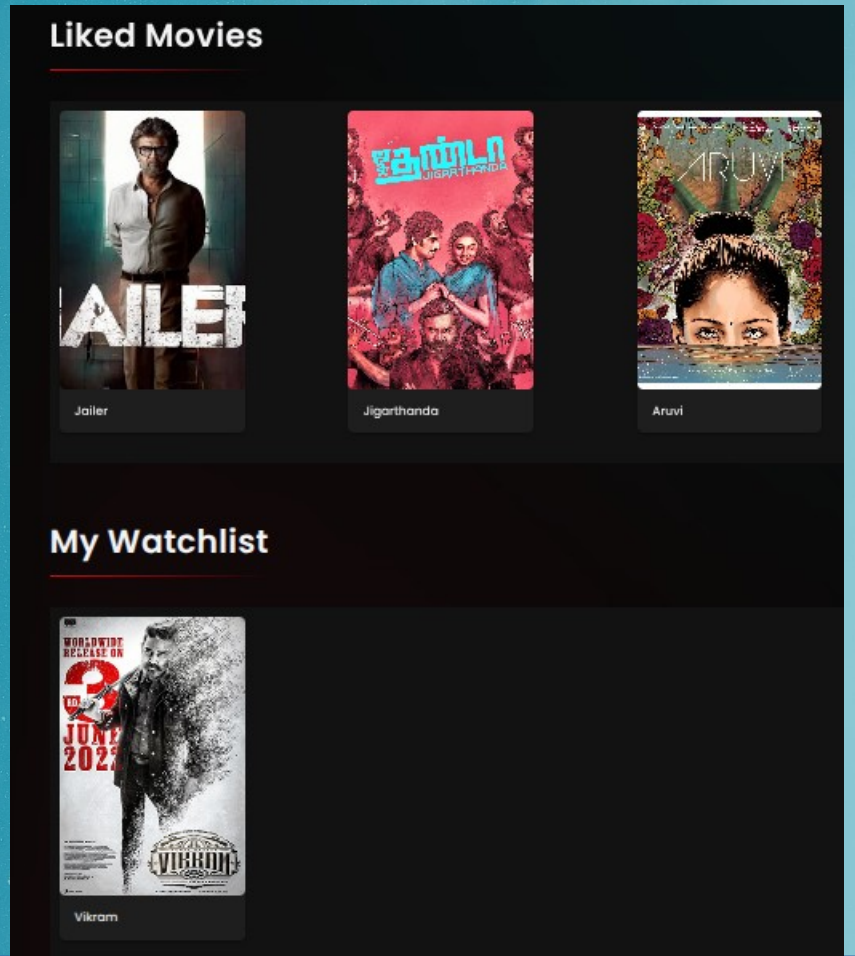
DETAILS OF THE MOVIE



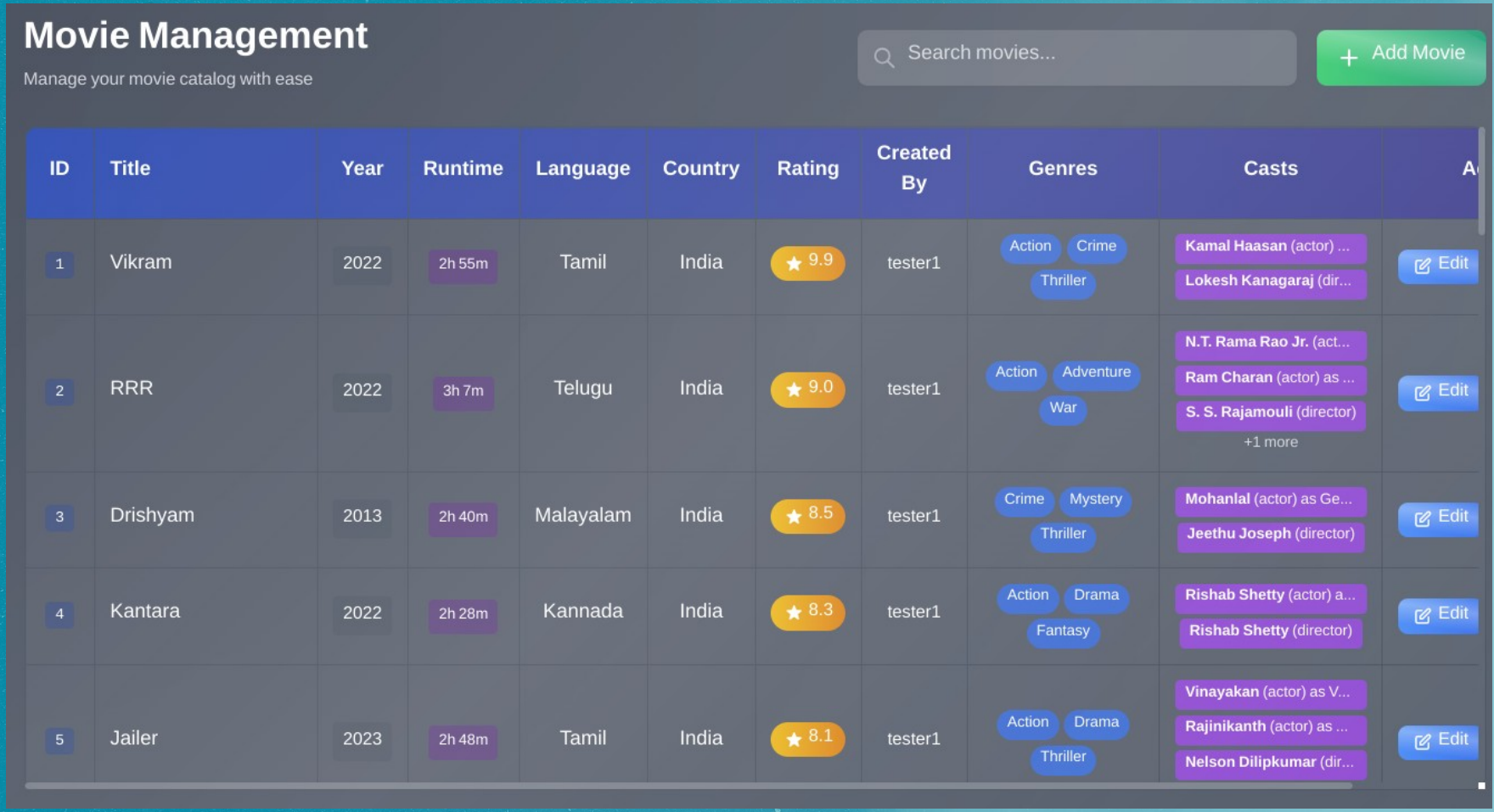
REVIEWS / RATINGS



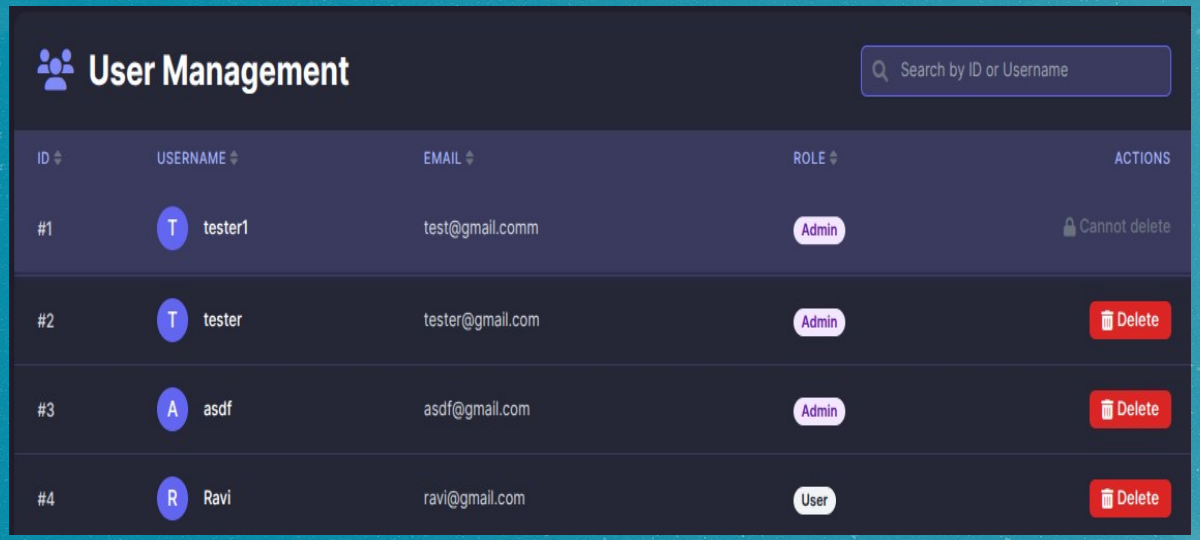
LIKES / WATCH-LISTS



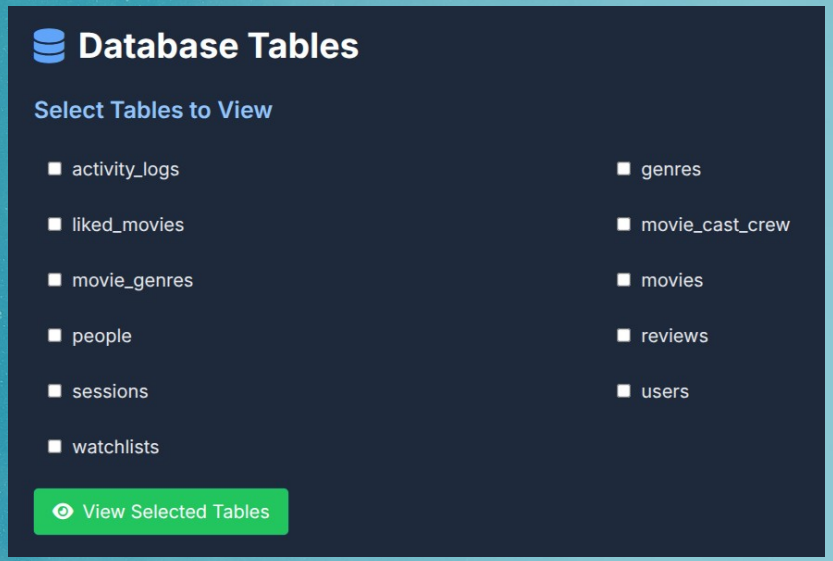
ADD / EDIT /DELETE MOVIES



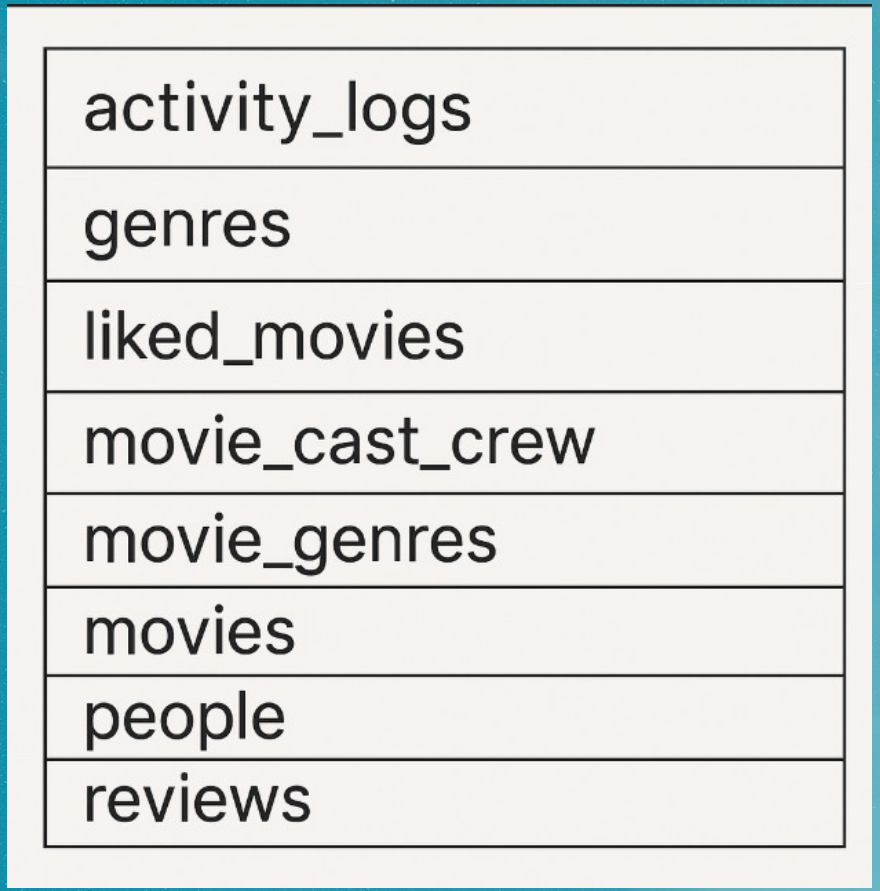
MANAGE USERS

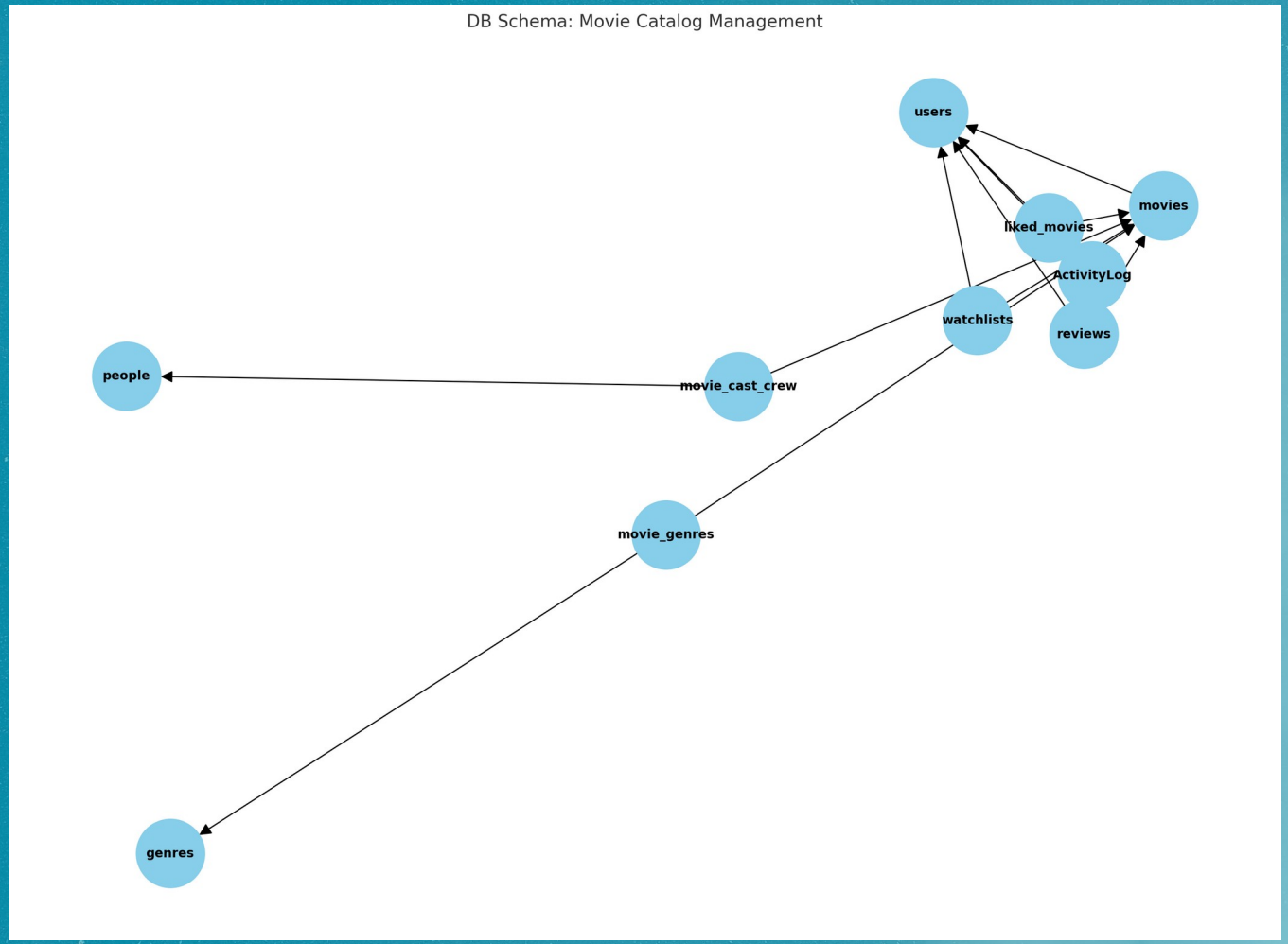


MANAGE DB



DB-SCHEMA


activity_logs
genres
liked_movies
movie_cast_crew
movie_genres
movies
people
reviews


DB Schema: Movie Catalog Management
users
movies
liked_movies
ActivityLog
watchlists
reviews
people
movie_cast_crew
movie_genres
genres

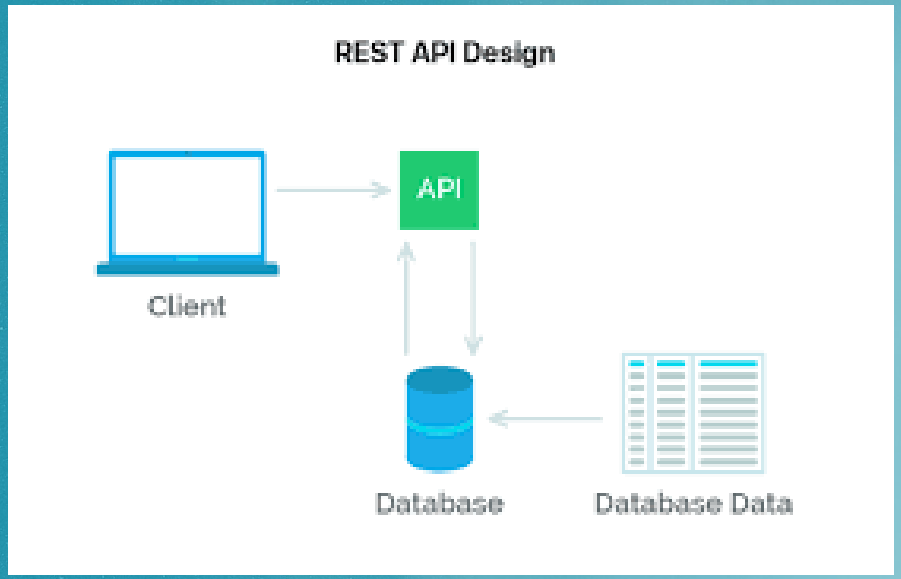
Database Tables and Fields

USERS	MOVIES	REVIEWS	GENRES	MOVIE_GENRES	PEOPLE	MOVIE_CAST_CREW	WATCHLISTS	LIKED_MOVIES	ACTIVITYLOG
id	id	id	id	movie_id	id	id	user_id	user_id	id
username	title	movie_id	name	genre_id	name	movie_id	movie_id	movie_id	user_id
email	release_year	user_id				person_id			action_type
role	runtime	content				character_name			entity_type
password	synopsis	rating				role			entity_id
	language	likes_count							created_at
	average_rating	dislikes_count							
	created_by								

ROLE OF DATABASE

CRUD

- Create
- Read
- Update
- Delete



CREATE

```
  const [result] = await db.query(
    'INSERT INTO users (username, email, password, role) VALUES (?, ?, ?, ?)',
    [username, email, hashedPassword, role]
  );
```

```
  await connection.query("INSERT INTO movie_genres (movie_id, genre_id) VALUES (?, ?)", [movieId, genreId]);
}
```

```
await connection.query(
  "INSERT INTO movie_cast_crew (movie_id, person_id, character_name, role) VALUES (?, ?, ?, ?)",
  [movieId, personId, characterName, role]
);
```

READ

```
const [castCrew] = await db.query(`
  SELECT mcc.id, mcc.character_name, mcc.role,
         p.id AS person_id, p.name
  FROM movie_cast_crew mcc
  JOIN people p ON mcc.person_id = p.id
  WHERE mcc.movie_id = ?
`, [id]);
```

```
const [movies] = await connection.query(
  "SELECT * FROM movies WHERE id = ?",
  [movieId]
);
```

```
// Get cast and crew for the movie
const [movieCasts] = await connection.query(
  "SELECT * FROM movie_cast_crew WHERE movie_id = ?",
  [movieId]
);
```

```
const [movies] = await db.query(
  `
  SELECT
    m.id, m.title, m.average_rating
  FROM movies m
  ORDER BY m.title ASC
  `
);
```

```
const [genres] = await db.query("SELECT * FROM genres ORDER BY name");
```

UPDATE

```
// Update average rating and rating count
await db.query(
  `UPDATE movies
  SET average_rating = (average_rating * rating_count + ?) / (rating_count + 1),
  rating_count = rating_count + 1
  WHERE id = ?`,
  [numericRating, movieId]
);
```

```
    await connection.query(
      `UPDATE movies SET
       title = ?, release_year = ?,
        average_rating = ?,
        runtime = ?, synopsis = ?,
        language = ?, country = ?
        WHERE id = ?`,
      [
        title, release_year,
        rating,
        runtime || null, synopsis || null,
         language || null,country || null,
        movieId,
      ]
    );
```

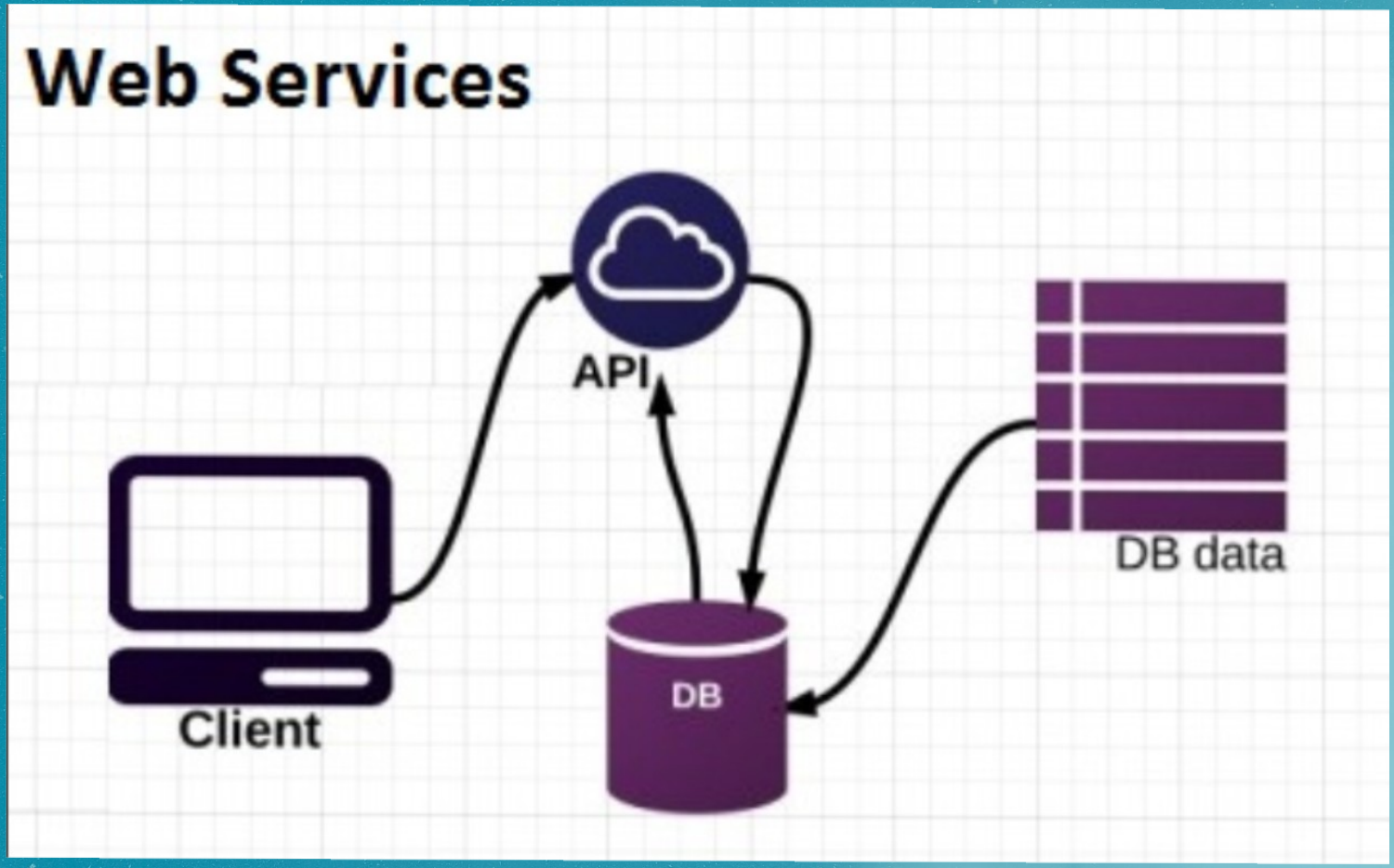
DELETE

```
// Delete existing genres and re-insert them
await connection.query("DELETE FROM movie_genres WHERE movie_id = ?", [movieId]);
```

```
// Delete existing cast and crew and re-insert them
await connection.query("DELETE FROM movie_cast_crew WHERE movie_id = ?", [movieId]);
```

```
await db.query("DELETE FROM people WHERE id = ?", [id]);
```

```
await db.query("DELETE FROM genres WHERE id = ?", [id]);
```


Web Services
API
Client
DB
DB data