Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS
SCHOOL OF COMPUTER ENGINEERING AND TECHNOLOGY

DATABASE MANAGEMENT SYSTEM

MOVIES LIBRARY MANANGEMENT SYSTEM

MINI PROJECT REPORT

Abstract: This movie library management system mini project aims to develop a Database Management System (DBMS) to organize and manage a collection of movies in a library. The system focuses on cata-logging movies, tracking availability, managing checkouts, and providing a user-friendly interface for library staff and patrons. The goal is to streamline the management of movie resources and enhance the overall efficiency of a movie library.

Title: Movie Library Management System

Introduction: Motivation and Objectives

The motivation behind this project is to address the challenges faced by movie libraries in maintaining an organized and accessible movie collection. Traditional manual systems may lead to difficulties in tracking movie availability, managing checkouts, and providing a seamless experience for library users. The objectives of the system include:

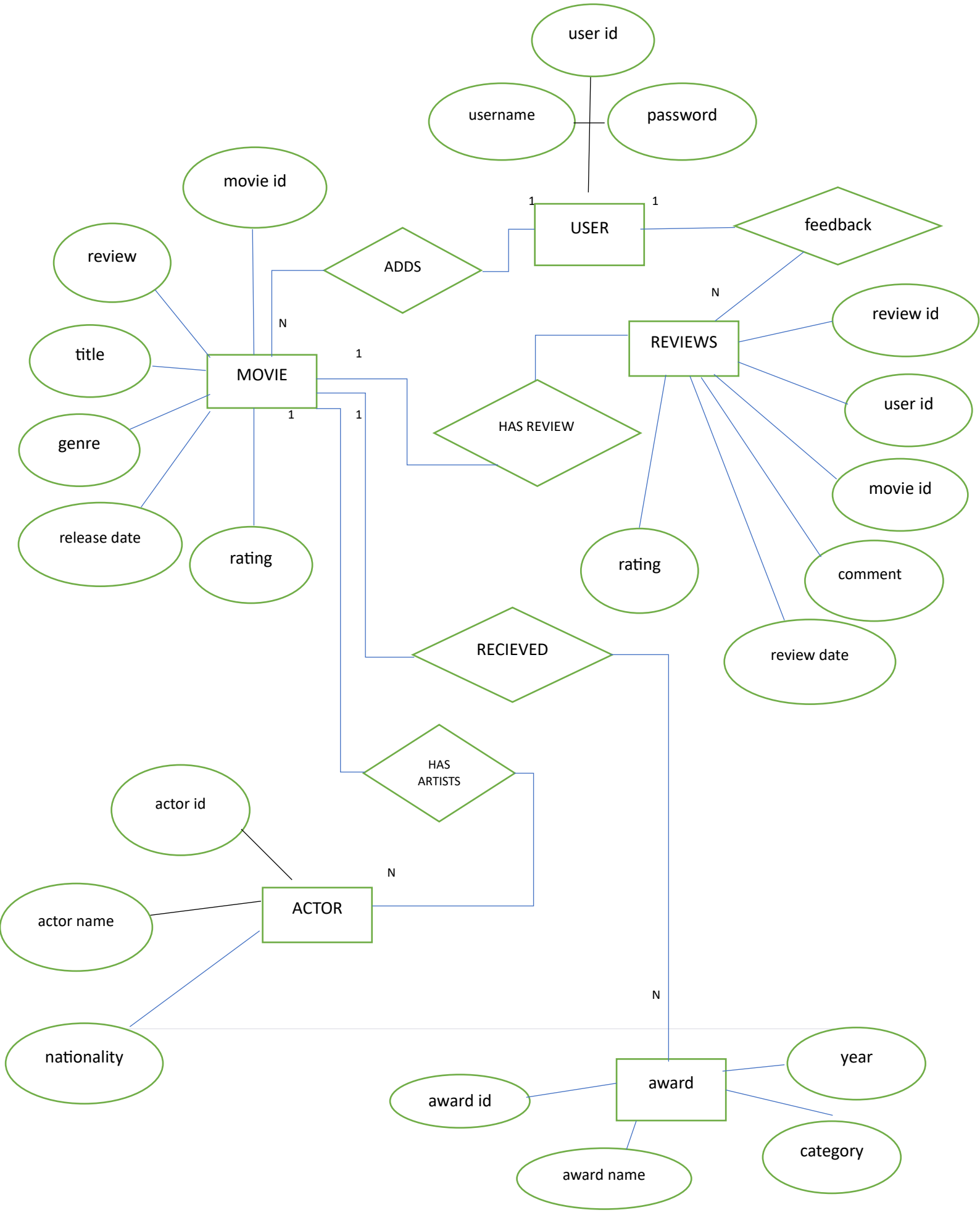
1. Efficient cata-logging and organization of movies.
2. Real-time tracking of movie availability.
3. Streamlined checkout and return processes.
4. User-friendly interface for library staff and patrons.

Problem Definition: Movie libraries often face challenges in efficiently managing their collections, resulting in difficulties for library staff and patrons in locating and checking out movies. Manual tracking systems may lead to errors, delays, and an overall lack of organization. This project aims to provide a digital solution that automates the management of a movie library, addressing issues such as inventory tracking, user management, and checkout procedures.

Tools and Technologies: The system will be developed using the following tools and technologies:

1. Database Management System: MySQL
2. Programming Language: Python (for application logic)
3. Front-end: HTML, CSS, JavaScript
4. Back-end: Python and MySQL

Database Design (ER Diagram):



Database Schema: The database schema is designed to capture essential information related to movies, actors, users, awards, reviews. The primary entities include:

1. movies: movie_id, title, genre, release_date, ratings, review

2. actors: actor_id, actor_name, nationality

3. users: user_id, username, password

4. awards: award_id, award_name, category, year, movie_id, actor_id

5. reviews: review_id, user_id, movie_id, rating, comment, review_date

DDL (Create Table and Constraints) Script:

```
create database moviemgmt;
use moviemgmt;

-- MOVIES
CREATE TABLE movies (
    movie_id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL,
    genre VARCHAR(50),
    release_date INT,
    ratings FLOAT,
    review TEXT
);

-- ACTORS
CREATE TABLE actors (
    actor_id INT PRIMARY KEY AUTO_INCREMENT,
    actor_name VARCHAR(100) NOT NULL,
    nationality VARCHAR(50)
);
select * from actors;

-- ACTORS2
select * from actors2;
CREATE TABLE actors2(
    actor_name VARCHAR(50),
    movie_name VARCHAR(50)
);

-- USERS
CREATE TABLE users (
```

```
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL,
    type VARCHAR(20) NOT NULL
);
-- AWARDS
CREATE TABLE awards (
    award_id INT PRIMARY KEY AUTO_INCREMENT,
    award_name VARCHAR(100) NOT NULL,
    category VARCHAR(50),
    year INT,
    movie_id INT,
    actor_id INT,
    FOREIGN KEY (movie_id) REFERENCES movies(movie_id),
    FOREIGN KEY (actor_id) REFERENCES actors(actor_id)
);
select * from awards;

-- REVIEWS
CREATE TABLE reviews (
    review_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    movie_id INT,
    rating FLOAT,
    comment TEXT,
    review_date DATE,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (movie_id) REFERENCES movies(movie_id)
);
```

DML Queries:

```
-- Insert sample data into movies table
INSERT INTO movies (title, genre, release_date, ratings, review)
VALUES
    ('Inception', 'Sci-Fi', 2010, 8.7, 'Mind-bending thriller with stunning
visuals.'),
    ('The Shawshank Redemption', 'Drama', 1994, 9.3, 'Classic tale of hope and
redemption.'),
    ('The Dark Knight', 'Action', 2008, 9.0, 'Epic superhero film with a
brilliant performance by Heath Ledger.');

-- Insert sample data into actors table
drop table actors;
select * from actors;
INSERT INTO actors (actor_name, nationality)
VALUES
    ('Leonardo DiCaprio', 'American'),
```

```
    ('Morgan Freeman', 'American'),
    ('Christian Bale', 'British'),
    ('Tom Holland', 'UK');

-- Insert sample data into users table
INSERT INTO users (username, password)
VALUES
    ('Atharv','123'),
    ('user1','password1'),
    ('user2','password2'),
    ('user3','password3');

-- Insert sample data into awards table
INSERT INTO awards (award_name, category, year, movie_id, actor_id)
VALUES
    ('Oscar', 'Best Picture', 2011, 1, 1),
    ('Golden Globe', 'Best Actor', 1995, 2, 2),
    ('BAFTA', 'Best Supporting Actor', 2009, 3, 3);

-- Insert sample data into reviews table
INSERT INTO reviews (user_id, movie_id, rating, comment, review_date)
VALUES
    (1, 1, 9.0, 'Amazing movie! Loved the plot twists.', '2023-01-01'),
    (2, 2, 9.5, 'One of the best movies ever made.', '2023-01-02'),
    (3, 3, 8.5, 'Great performance by Christian Bale.', '2023-01-03');
```

```
select * from movies;
DELETE m1 FROM movies m1
JOIN movies m2 ON m1.title = m2.title AND m1.movie_id > m2.movie_id;
```

```
ALTER TABLE actors2
ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
```

```
alter table users add type varchar(20);
desc users;
select * from users;
SELECT username, password, type FROM users WHERE username = "user1" AND
password = "password1";
UPDATE users
SET type = 'admin'
WHERE user_id = 1;
```

DCL Queries :

```
def enable_admin_privileges(self):
        self.logged_in_as_admin = True
        self.open_movie_management_window()

    def enable_user_privileges(self):
        self.logged_in_as_admin = False
        self.open_movie_management_window()
```

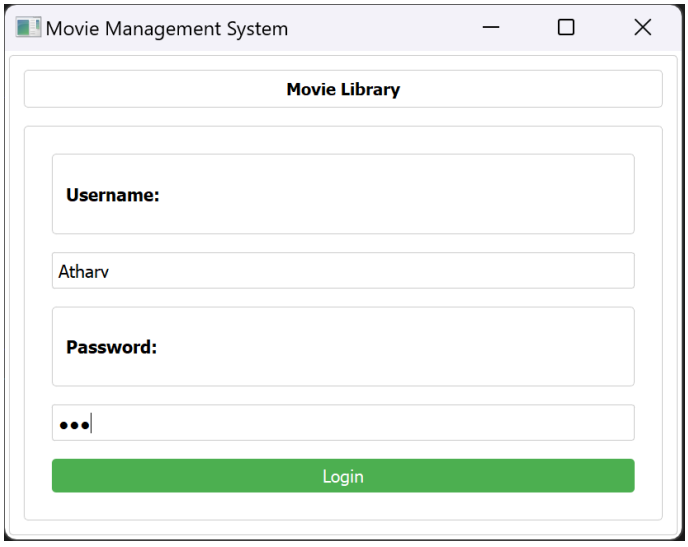
Triggers :

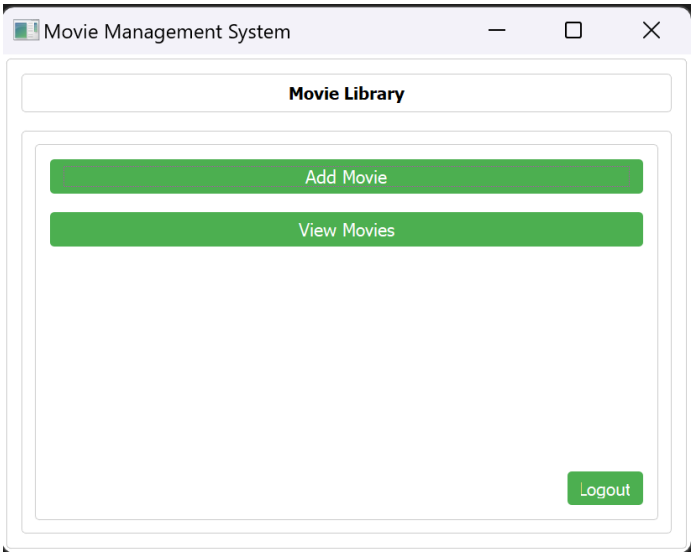
```
-- trigger to delete duplicate movie data
DELIMITER //

CREATE TRIGGER delete_duplicates
BEFORE INSERT ON movies
FOR EACH ROW
BEGIN
    DECLARE movie_count INT;
    SET movie_count = (SELECT COUNT(*) FROM movies WHERE title = NEW.title);
    IF movie_count > 1 THEN
        DELETE FROM movies WHERE title = NEW.title LIMIT 1;
    END IF;
END;
//

DELIMITER ;
```

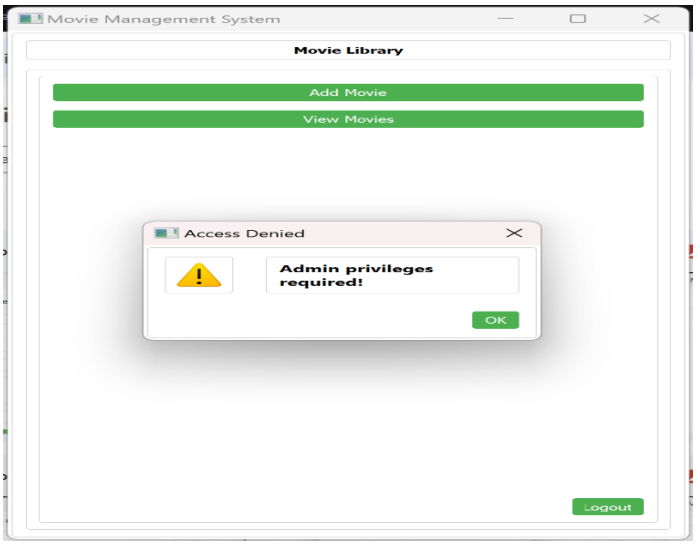
PL/SQL Procedure :

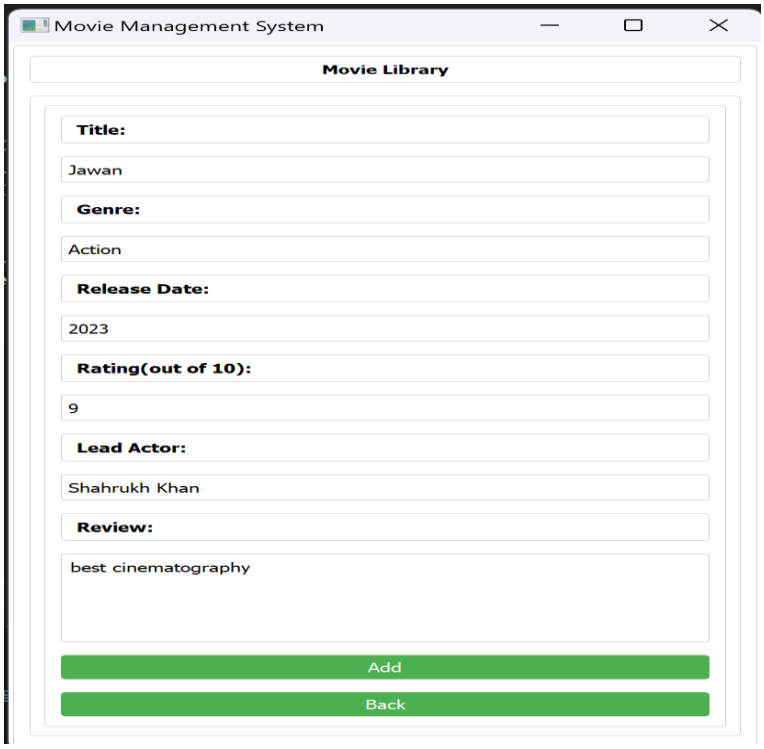
```
-- procedure to show all data in the movies table
delimiter //
create procedure show_data(in movie_id int)
begin
select *from movies;
end//
```

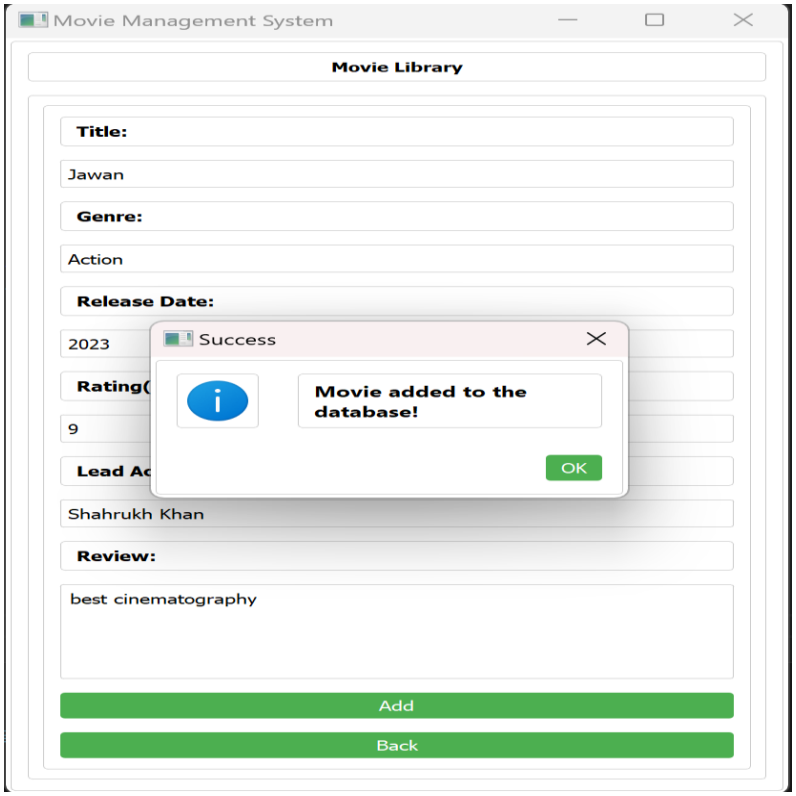
Frontend GUI Screenshots :

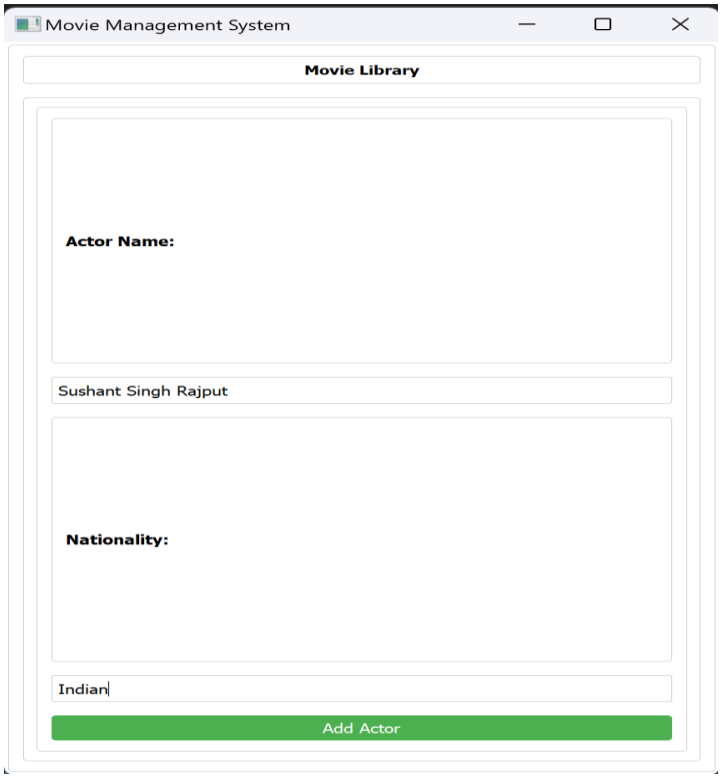


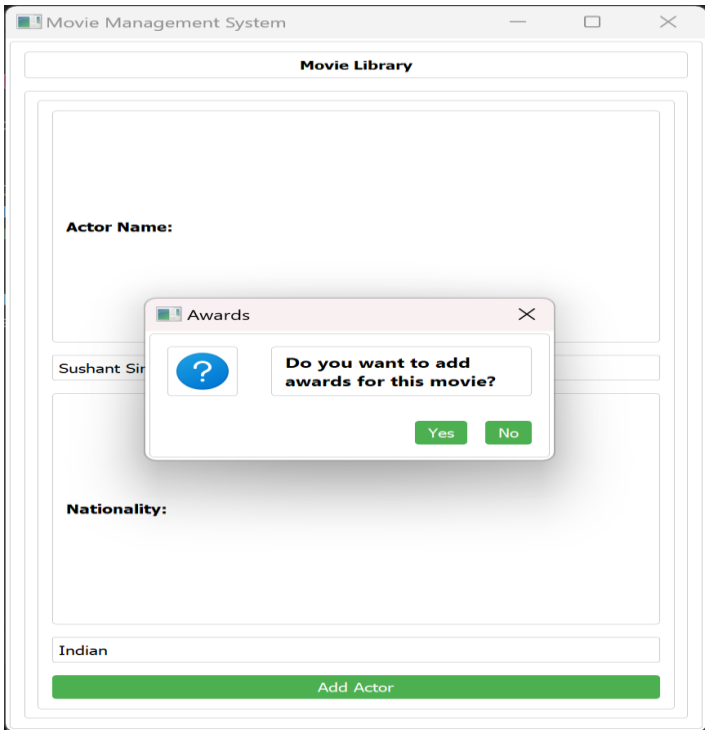


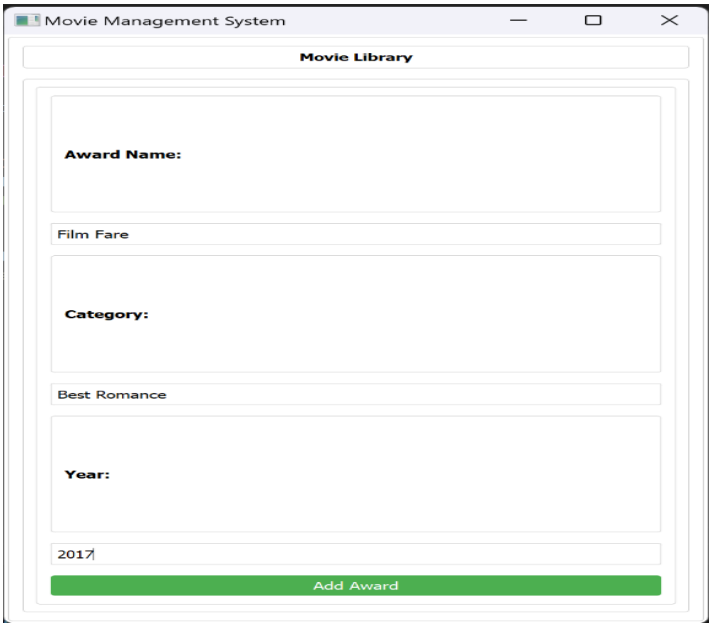


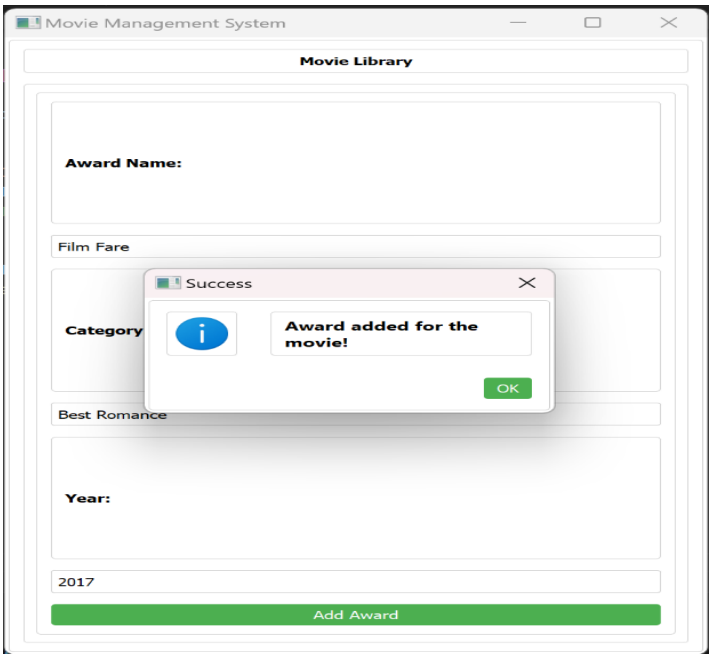

Movie Management System
Movie Library
Title:
Jawan
Genre:
Action
Release Date:
2023
Rating(out of 10):
9
Lead Actor:
Shahrukh Khan
Review:
best cinematography
Add
Back

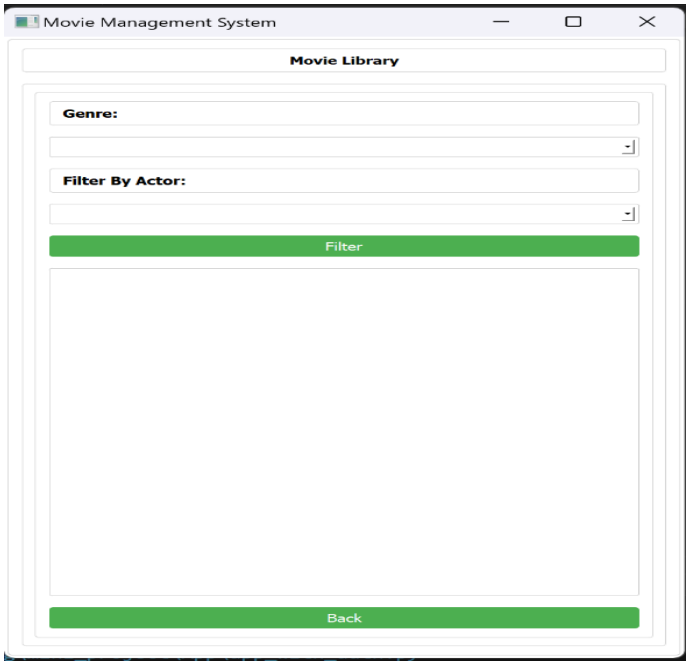

Movie Management System
Movie Library
Title:
Jawan
Genre:
Action
Release Date:
2023
Success
Movie added to the database!
OK
Shahrukh Khan
Review:
best cinematography
Add
Back

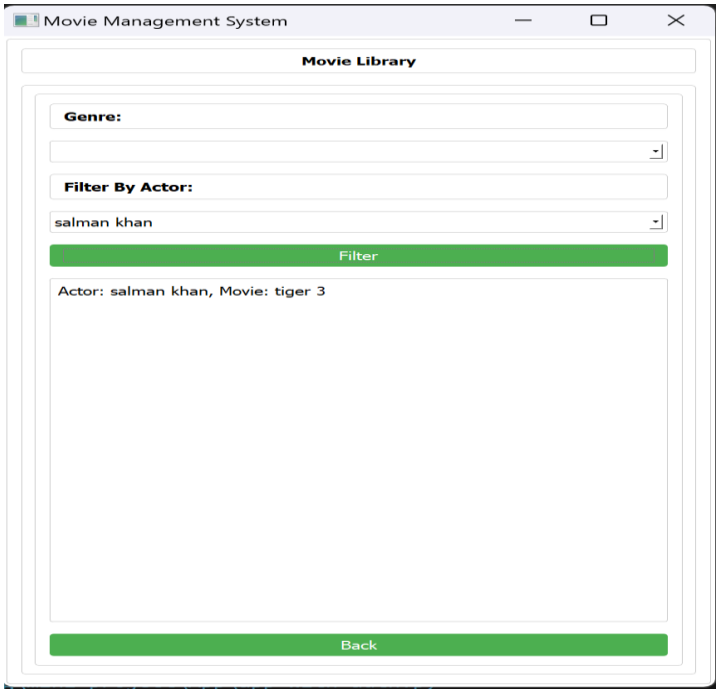

Movie Management System
Movie Library
Actor Name:
Sushant Singh Rajput
Nationality:
Indian
Add Actor

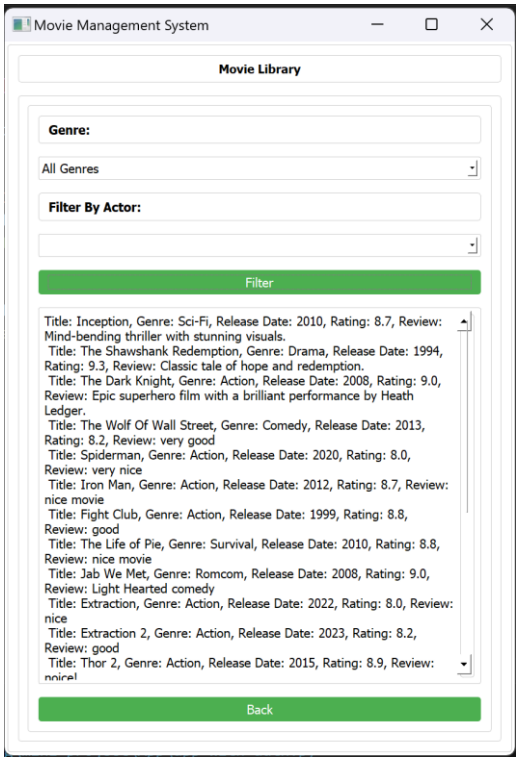

Movie Management System
Movie Library
Actor Name:
Awards
Do you want to add awards for this movie?
Yes
No
Nationality:
Indian
Add Actor


Movie Management System
Movie Library
Award Name:
Film Fare
Category:
Best Romance
Year:
2017
Add Award


Movie Management System
Movie Library
Award Name:
Film Fare
Category
Success
Award added for the movie!
OK
Best Romance
Year:
2017
Add Award


Movie Management System
Movie Library
Genre:
Filter By Actor:
Filter
Back


Movie Management System
Movie Library
Genre:
Filter By Actor:
salman khan
Filter
Actor: salman khan, Movie: tiger 3
Back



Conclusion : The Movie Library Management System provides an efficient and user-friendly platform for organizing, cata-logging, and retrieving movie information. With its intuitive interface and robust features, users can easily add, edit, and search for movies, enhancing the overall management and accessibility of the movie library. This system streamlines the process of maintaining a well-organized collection, ultimately improving the user experience and efficiency of library operations

Group Member :

1. **Atharv Chinchkar – 44**
2. **Aditya Jain – 40**
3. **Ritesh Goyal – 36**
4. **Krishna Mahajan - 48**