

100 feet Ring Road, BSK 3rd Stage, Bengaluru 560085

UE17CS252

Database Management Systems

Project Report

International

Movie Database

PES1201700652  
Varun R. Gupta

PES1201700276  
Hrishikesh V.

PES1201700706  
Ravendra Singh

Table of Contents

1. Introduction
2. Problem Statement
3. Requirements Specification
4. Design
   1. ER Diagram
   2. SQL
      1. DDL Statements
      2. INSERT statements for bulk loading of data
5. Implementation
   1. Web User Interfaces
      1. CRUD Operations
   2. SQL Complex Queries with results
6. Conclusion with proposed enhancements
7. References
8. Appendix
   1. Abbreviations & Acronyms
   2. Databases, Tools & Technologies Used

# Introduction

Movie databases or websites that handle information on Movies have become just as popular as movies

themselves. People rely on these websites for ratings and critic reviews.

It is important to create a platform that runs seamlessly, enabling end users and administrators to easily

access content on the websites.

We have implemented certain aspects of the movie database, along with basic user interface for making

changes.

Not only does our website display and allow users to make changes to the database, but also does it

print the showtimes of any movie in the database.

Described in this document in detail are the various features that we have implemented.

# Problem Statement

The aim of the project is to design and implement the working of a database that is typically used in

movie review/rating websites.

There needs to be implemented all the functionalities of a typical movie database such as the rating

system or the insert – delete capabilities and so on.

# Requirements Specifications

## System Requirements

1. 100KB of Disk space
2. Hardware to run a server

## Application Requirements

1. Apache or equivalent server software
2. PostgreSQL or equivalent database software
3. Web Browser

## Client Requirements

1. Easy access to information on Movies/Actors/Studios etc.
2. Means to rate any movie
3. A way to update the contents of the database

# Design

## ER Diagram

## C:\Users\varun\OneDrive\Pictures\DBMS Project - ERD.png

## Relational Schema Diagram

# SQL

## DDL Statements

-- Created by Team C9 for PES University

CREATE TABLE `MOVIE` (

`ID` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`NAME` VARCHAR(200) NOT NULL UNIQUE,

`RATING` DECIMAL(3, 1),

`RELEASE` YEAR NOT NULL,

`LANGUAGE` VARCHAR(50) NOT NULL,

`DIRECTOR` VARCHAR(100) NOT NULL,

`LOCATION` VARCHAR(100) NOT NULL,

`RUNTIME` INT NOT NULL,

`MPAA\_RATING` INT(1),

`BUDGET` DECIMAL(15,2) NOT NULL

);

CREATE TABLE `CELEBRITY` (

  `ID` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

  `NAME` VARCHAR(100) NOT NULL UNIQUE,

  `GENDER` BIT DEFAULT NULL,

  `JOB\_TITLES` VARCHAR(100) NOT NULL DEFAULT '',

  `BIO` TEXT(5000) NOT NULL,

  -- `DOB` DATE,

`BIRTH\_LOC` VARCHAR(200) NOT NULL DEFAULT ''

);

CREATE TABLE `ACTED\_IN` (

  `CELEB\_ID` INT NOT NULL,

  `MOVIE\_ID` INT NOT NULL,

  `ROLE` VARCHAR(100) NOT NULL DEFAULT '',

FOREIGN KEY (`MOVIE\_ID`) REFERENCES `MOVIE`(`ID`),

  FOREIGN KEY (`CELEB\_ID`) REFERENCES `CELEBRITY`(`ID`)

);

CREATE TABLE `SCREEN` (

  `ID` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

  `NAME` VARCHAR(100) NOT NULL DEFAULT '',

  `LOCATION` VARCHAR(100) NOT NULL DEFAULT '',

  `SERVICES` VARCHAR(100)

);

CREATE TABLE `SHOW` (

  `ID` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

  `MOVIE\_ID` INT NOT NULL,

  `SCREEN\_ID` INT NOT NULL,

  `LANGUAGE` VARCHAR(100) NOT NULL DEFAULT '',

FOREIGN KEY (`MOVIE\_ID`) REFERENCES `MOVIE`(`ID`),

  FOREIGN KEY (`SCREEN\_ID`) REFERENCES `SCREEN`(`ID`)

);

CREATE TABLE `SHOWTIME` (

  `SHOW\_ID` INT NOT NULL,

  `TIME` DATETIME NOT NULL,

FOREIGN KEY (`SHOW\_ID`) REFERENCES `SHOW`(`ID`)

);

CREATE TABLE `STUDIO` (

  `ID` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

  `NAME` VARCHAR(100) NOT NULL DEFAULT '',

  `HQ` VARCHAR(200) NOT NULL DEFAULT '',

  `CEO` VARCHAR(100) NOT NULL DEFAULT ''

  -- `SPONSORS` TEXT(500) NOT NULL DEFAULT ''

);

CREATE TABLE `PRODUCTION\_CREW` (

  `CREW\_NAME` VARCHAR(100) NOT NULL DEFAULT '',

  `JOB\_DESCRIPTION` VARCHAR(1000) NOT NULL DEFAULT '' COMMENT 'Contains job done, equipment used and set worked on.'

);

## Insert Statements

INSERT INTO `MOVIE` VALUES(1, 'The Shallows', 6.3, 2016, 'English', 'Jaume Collet-Serra', 'USA', 86, 4, 18.5);

INSERT INTO `MOVIE` VALUES(2, 'Whiplash', 8.5, 2014, 'English', 'Damien Chazelle', 'USA', 106, 3, 3.3);

INSERT INTO `MOVIE` VALUES(3, 'Top Gun', 6.8, 1986, 'English', 'Tony Scott', 'USA', 110, 3, 15);

INSERT INTO `MOVIE` VALUES(4, 'The Terminal', 7.3, 2004, 'English', 'Steven Spielberg', 'USA', 86, 4, 18.5);

INSERT INTO `MOVIE` VALUES(5, 'The Last of the Mohicans', 7.8, 1992, 'French', 'Michael Mann', 'USA', 112, 3, 40);

INSERT INTO `MOVIE` VALUES(6, 'The Fifth Estate', 6.2, 2013, 'English', 'Bill Condon', 'United Kingdom', 128, 2, 28.5);

INSERT INTO `MOVIE` VALUES(7, 'Interstellar', 8.6, 2014, 'English', 'Christopher Nolan', 'USA', 169, 5, 165.5);

INSERT INTO `MOVIE` VALUES(8, 'Crocodile Dundee', 6.6, 1986, 'English', 'Peter Faiman', 'Australia', 104, 3, 8.5);

INSERT INTO `MOVIE` VALUES(9, 'Raiders of the Lost Ark', 8.5, 1981, 'English', 'Steven Spielberg', 'USA', 115, 4, 18.5);

INSERT INTO `MOVIE` VALUES(10, 'The Good, The Bad and the Ugly', 8.9, 1966, 'English', 'Sergio Leone', 'USA', 177, 5, 1.5);

INSERT INTO `MOVIE` VALUES(11, 'Good Will Hunting', 8.3, 1997, 'English', 'Gus Van-Sant', 'USA', 126, 4, 10.5);

INSERT INTO `MOVIE` VALUES(12, 'Die Hard', 8.2, 1988, 'English', 'John Mctiernan', 'USA', 132, 4, 28.5);

INSERT INTO `CELEBRITY` VALUES(1, 'Blake Lively', 1, 'Actor', 'Known for her role in Gossip Girl', 'USA');

INSERT INTO `CELEBRITY` VALUES(2, 'J.K. Simmons', 0, 'Actor', 'Actor and a Voice Actor', 'USA');

INSERT INTO `CELEBRITY` VALUES(3, 'Miles Teller', 0, 'Actor', 'Alumnus of New York University', 'USA');

INSERT INTO `CELEBRITY` VALUES(4, 'Tom Cruise', 0, 'Actor', 'Mostly credited in Action Movies', 'USA');

INSERT INTO `CELEBRITY` VALUES(5, 'Val Kilmer', 0, 'Actor', 'Also a stage Actor', 'USA');

INSERT INTO `CELEBRITY` VALUES(6, 'Tom Hanks', 0, 'Actor', 'Really Rich guy with a worth of 400 mil', 'USA');

.

.

.

INSERT INTO `CELEBRITY` VALUES(20, 'Matt Damon', 0, 'Actor', 'One of the Most Bankable Stars', 'USA');

INSERT INTO `CELEBRITY` VALUES(21, 'Ben Affleck', 0, 'Actor', 'Actor, producer, screen writer', 'USA');

INSERT INTO `CELEBRITY` VALUES(22, 'Robin Williams', 0, 'Actor', 'Comedian', 'USA');

INSERT INTO `CELEBRITY` VALUES(23, 'Bruce Willis', 0, 'Actor', 'Actor, Producer and Singer', 'Germany');

INSERT INTO `CELEBRITY` VALUES(24, 'Alan Rickman', 0, 'Actor', 'Alumnus of Royal Academy of Dramatic Art', 'United Kingdom');

INSERT INTO `CELEBRITY` VALUES(25, 'Bonnie Bedelia', 1, 'Actor', 'Retired Actress', 'USA');

INSERT INTO `CELEBRITY` VALUES(26, 'Reginald ValJohnson', 0, 'Actor', 'Known for Cop Roles', 'USA');

INSERT INTO `ACTED\_IN` VALUES(1,1,'Lead');

INSERT INTO `ACTED\_IN` VALUES(2,2,'Antagonist');

INSERT INTO `ACTED\_IN` VALUES(3,2,'Lead');

.

.

.

INSERT INTO `ACTED\_IN` VALUES(24,12,'Antagonist');

INSERT INTO `ACTED\_IN` VALUES(25,12,'Supporting');

INSERT INTO `ACTED\_IN` VALUES(26,12,'Supporting');

INSERT INTO `SCREEN` VALUES(1,'PVR','Forum Mall', 'Food, Drinks, Wi-Fi');

INSERT INTO `SCREEN` VALUES(2,'PVR','Orion Mall', 'Food, Drinks, Wi-Fi');

INSERT INTO `SCREEN` VALUES(3,'PVR','Vega City Mall', 'Food, Drinks, Wi-Fi');

INSERT INTO `SCREEN` VALUES(4,'IMAX','Forum Mall', 'Food, Drinks, Wi-Fi');

INSERT INTO `SCREEN` VALUES(5,'IMAX','Vega City Mall', 'Food, Drinks, Wi-Fi');

INSERT INTO `SCREEN` VALUES(6,'INOX','Garuda Mall', 'Food, Drinks, Wi-Fi');

INSERT INTO `SCREEN` VALUES(7,'INOX','Bangalore Central Mall', 'Food, Drinks');

INSERT INTO `SCREEN` VALUES(8,'INOX','Mantri Square Mall', 'Food, Drinks, Wi-Fi');

INSERT INTO `SCREEN` VALUES(9,'INOX','Garuda Swagath Mall', 'Food, Drinks');

INSERT INTO `SHOW` VALUES(1,1,1,'English');

INSERT INTO `SHOW` VALUES(2,8,1,'English');

INSERT INTO `SHOW` VALUES(3,7,1,'English');

.

.

.

INSERT INTO `SHOW` VALUES(25,6,9,'English');

INSERT INTO `SHOW` VALUES(26,9,9,'English');

INSERT INTO `SHOW` VALUES(27,11,9,'English');

INSERT INTO `SHOWTIME` VALUES(1,'2019-04-01 10:15:00');

INSERT INTO `SHOWTIME` VALUES(1,'2019-04-01 14:00:00');

INSERT INTO `SHOWTIME` VALUES(1,'2019-04-01 19:30:00');

.

.

.

INSERT INTO `SHOWTIME` VALUES(1,'2019-04-02 10:15:00');

INSERT INTO `SHOWTIME` VALUES(27,'2019-04-03 14:00:00');

INSERT INTO `SHOWTIME` VALUES(27,'2019-04-03 19:30:00');

INSERT INTO `STUDIO` VALUES(1, 'Warner Brothers', 'Burbank, CA', 'Kevin Tsujihara');

INSERT INTO `STUDIO` VALUES(2, 'Fox', 'Los Angeles, CA', 'Stacey Snider');

INSERT INTO `STUDIO` VALUES(3, 'Paramount Pictures', 'Hollywood, CA', 'Jim Gianopolous');

INSERT INTO `STUDIO` VALUES(4, 'Universal', 'Universal City, CA', 'Ronald Meyer');

INSERT INTO `STUDIO` VALUES(5, 'Sony', 'Culver City, CA', 'Tony Vinciquerra');

INSERT INTO `STUDIO` VALUES(6, 'Walt Disney Studios', 'Burbank, CA', 'Alan Bergman');

INSERT INTO `PRODUCTION\_CREW` VALUES('Thorin', 'Crew Leader');

INSERT INTO `PRODUCTION\_CREW` VALUES('Fili', 'Mic');

INSERT INTO `PRODUCTION\_CREW` VALUES('Kili', 'Lights');  
.  
.  
.  
INSERT INTO `PRODUCTION\_CREW` VALUES('Carl', 'Refreshments');  
INSERT INTO `PRODUCTION\_CREW` VALUES('Josh', 'Makeup');  
INSERT INTO `PRODUCTION\_CREW` VALUES('Tim', 'Studio Assets');

# Complex Queries

SELECT \* FROM (CELEBRITY JOIN ACTED\_IN ON ID = CELEB\_ID);

1|Blake Lively|1|Actor|Known for her role in Gossip Girl|USA|1|1|Lead

2|J.K. Simmons|0|Actor|Actor and a Voice Actor|USA|2|2|Antagonist

3|Miles Teller|0|Actor|Alumnus of New York University|USA|3|2|Lead

4|Tom Cruise|0|Actor|Mostly credited in Action Movies|USA|4|3|Lead

5|Val Kilmer|0|Actor|Also a stage Actor|USA|5|3|Antagonist

.  
.  
.

SELECT M.NAME, C.NAME FROM MOVIE as M, CELEBRITY as C, ACTED\_IN as A WHERE MOVIE\_ID IN (SELECT MOVIE\_ID FROM ACTED\_IN WHERE C.ID = CELEB\_ID);

Crocodile Dundee|Blake Lively

Die Hard|Blake Lively

Good Will Hunting|Blake Lively

Interstellar|Blake Lively

Raiders of the Lost Ark|Blake Lively  
.  
.  
.

SELECT SUM(RATING), MAX(RATING), AVG(RATING) FROM MOVIE;

92.0|8.9|7.66666666666667

SELECT SUM(BUDGET), MAX(BUDGET), MIN(BUDGET), AVG(BUDGET) FROM MOVIE;

356.8|165.5|1.5|29.7333333333333

SELECT COUNT(\*) FROM CELEBRITY;

26

SELECT COUNT(\*) FROM ACTED\_IN GROUP BY MOVIE\_ID;

1

2

2

2

1

2

3

2

2

2

3

4

SELECT MOVIE.NAME, CELEBRITY.NAME, ACTED\_IN.ROLE FROM((ACTED\_IN INNER JOIN MOVIE ON MOVIE.ID = ACTED\_IN.MOVIE\_ID) INNER JOIN CELEBRITY ON CELEBRITY.ID = ACTED\_IN.CELEB\_ID);

The Shallows|Blake Lively|Lead

Whiplash|J.K. Simmons|Antagonist

Whiplash|Miles Teller|Lead

Top Gun|Tom Cruise|Lead

Top Gun|Val Kilmer|Antagonist

The Terminal|Tom Hanks|Lead

The Terminal|Catherine Zeta Jones|Supporting

.  
.  
.

SELECT \* FROM MOVIE LEFT JOIN ACTED\_IN ON MOVIE.ID = ACTED\_IN.MOVIE\_ID

1|The Shallows|6.3|2016|English|Jaume Collet-Serra|USA|86|4|18.5|1|1|Lead

2|Whiplash|8.5|2014|English|Damien Chazelle|USA|106|3|3.3|2|2|Antagonist

2|Whiplash|8.5|2014|English|Damien Chazelle|USA|106|3|3.3|3|2|Lead

3|Top Gun|6.8|1986|English|Tony Scott|USA|110|3|15|4|3|Lead

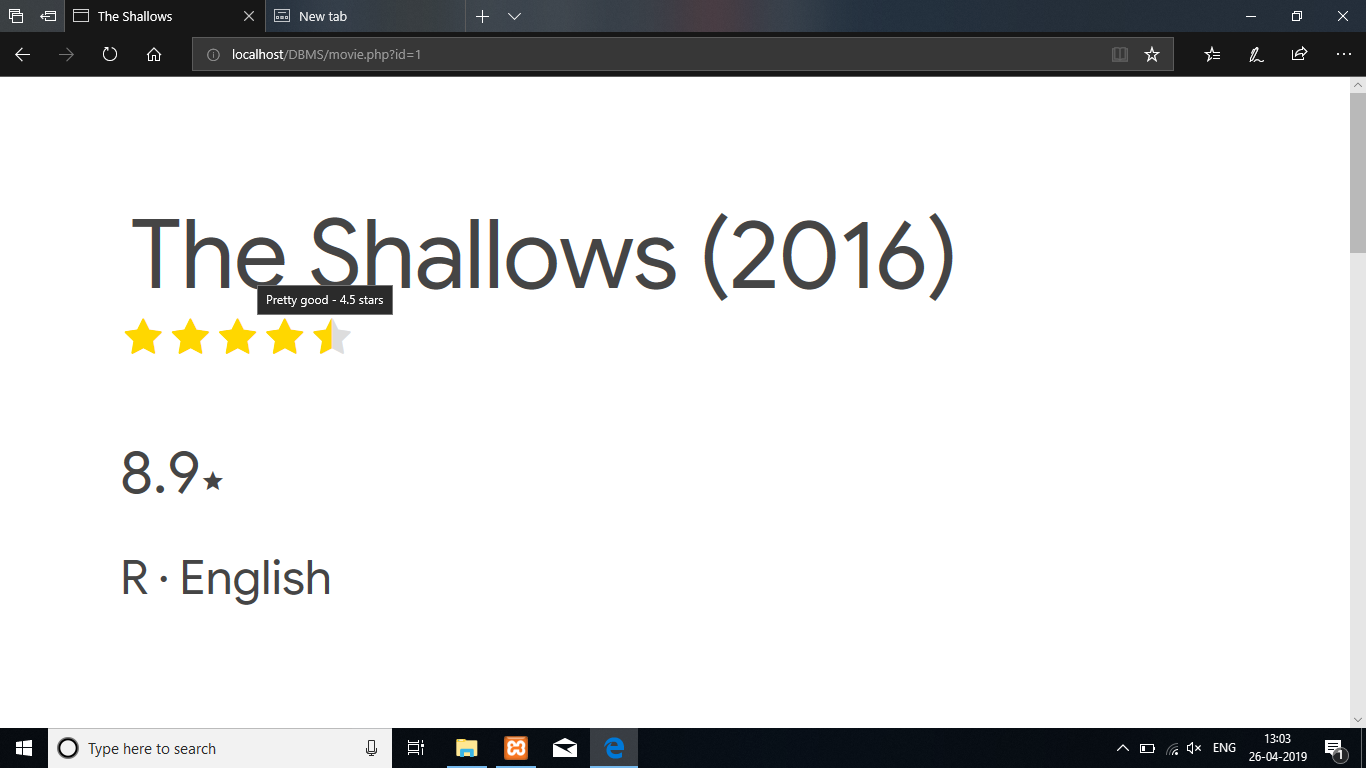
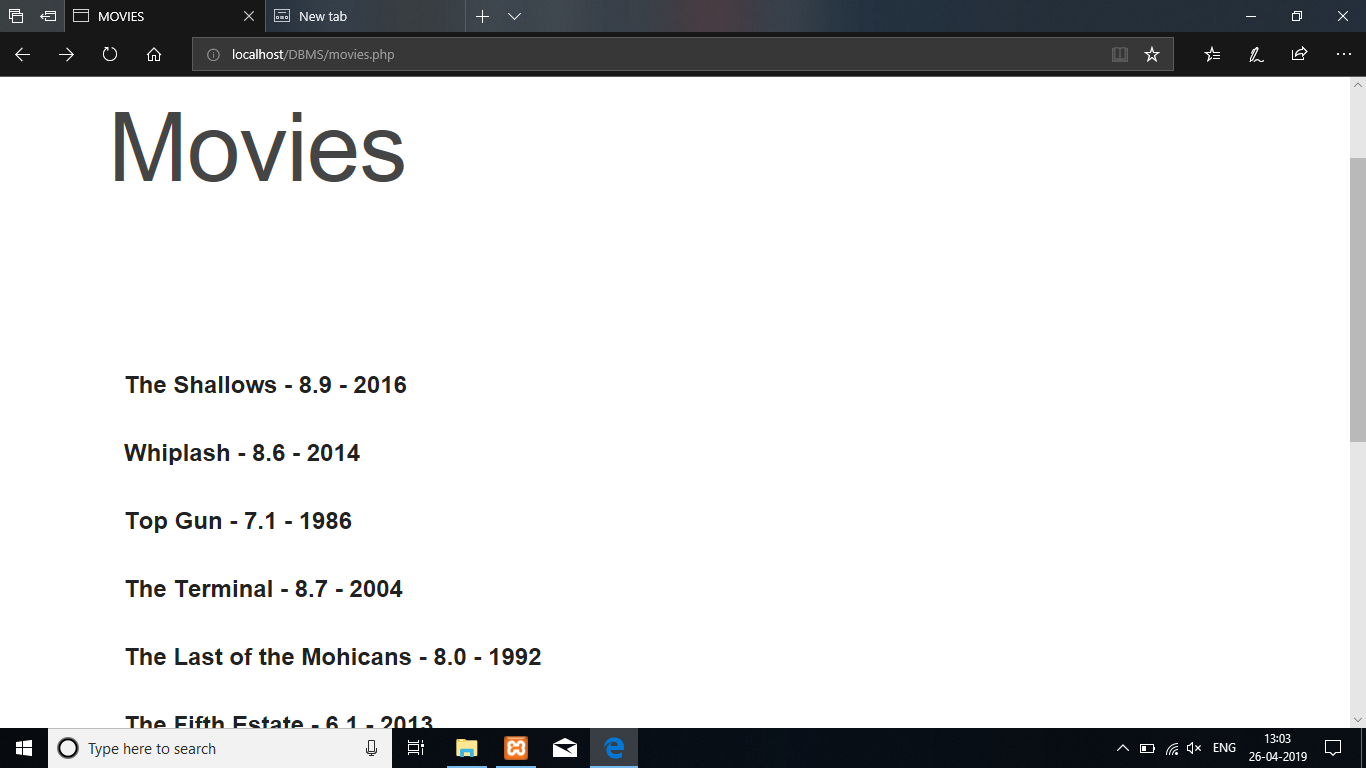
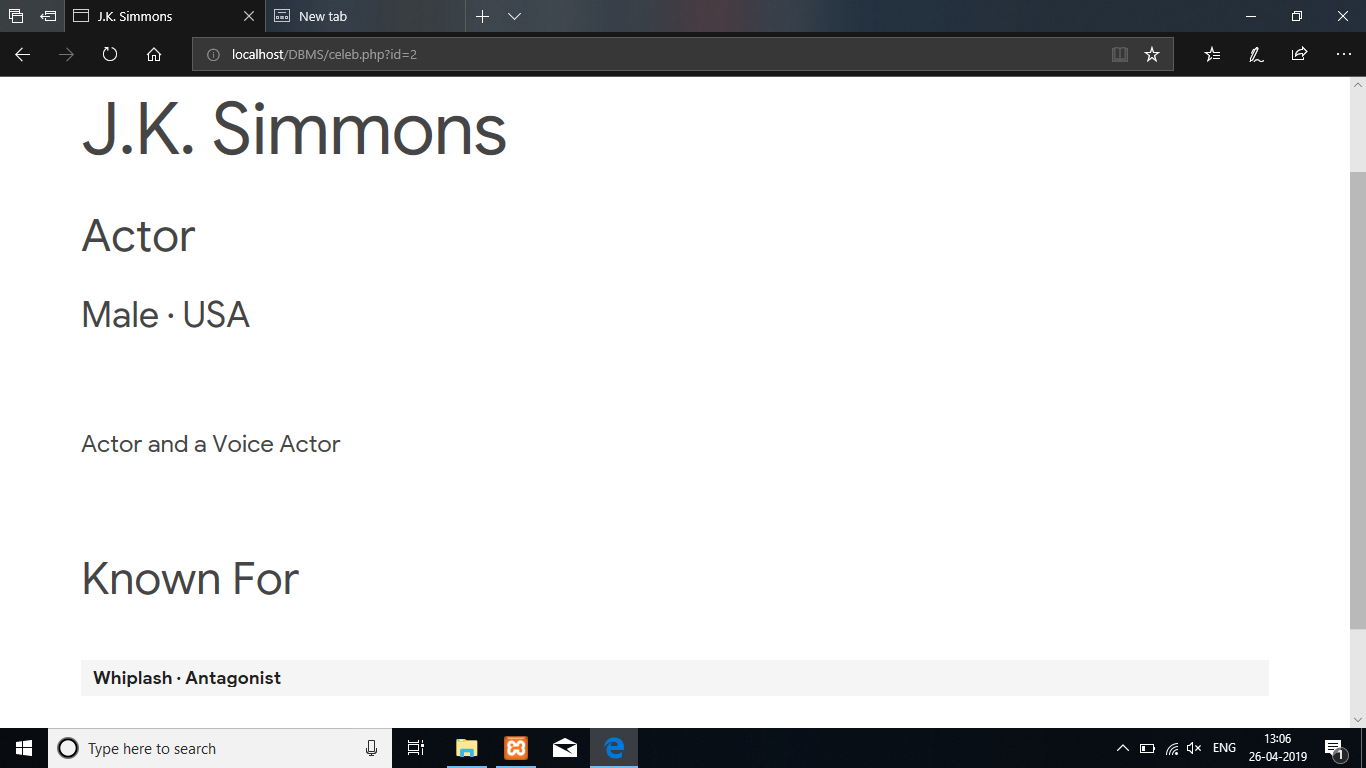
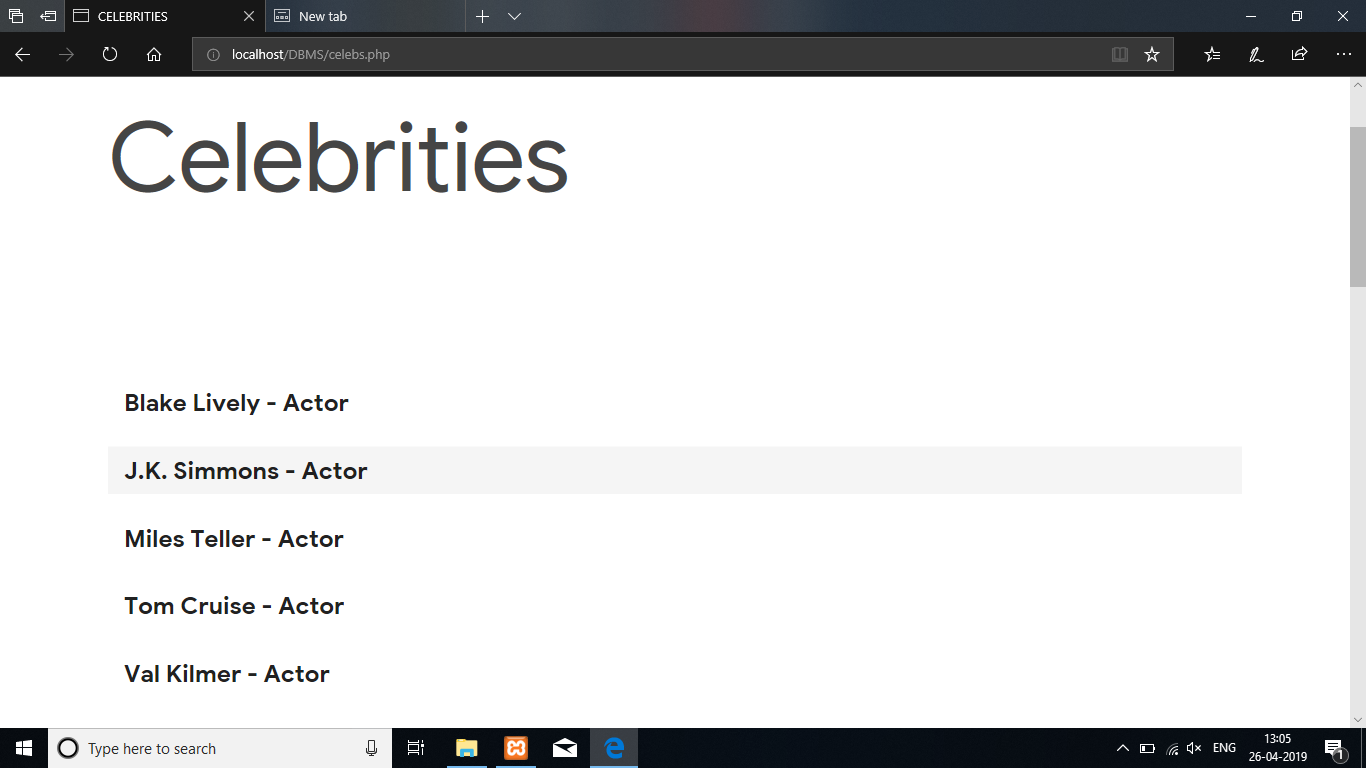
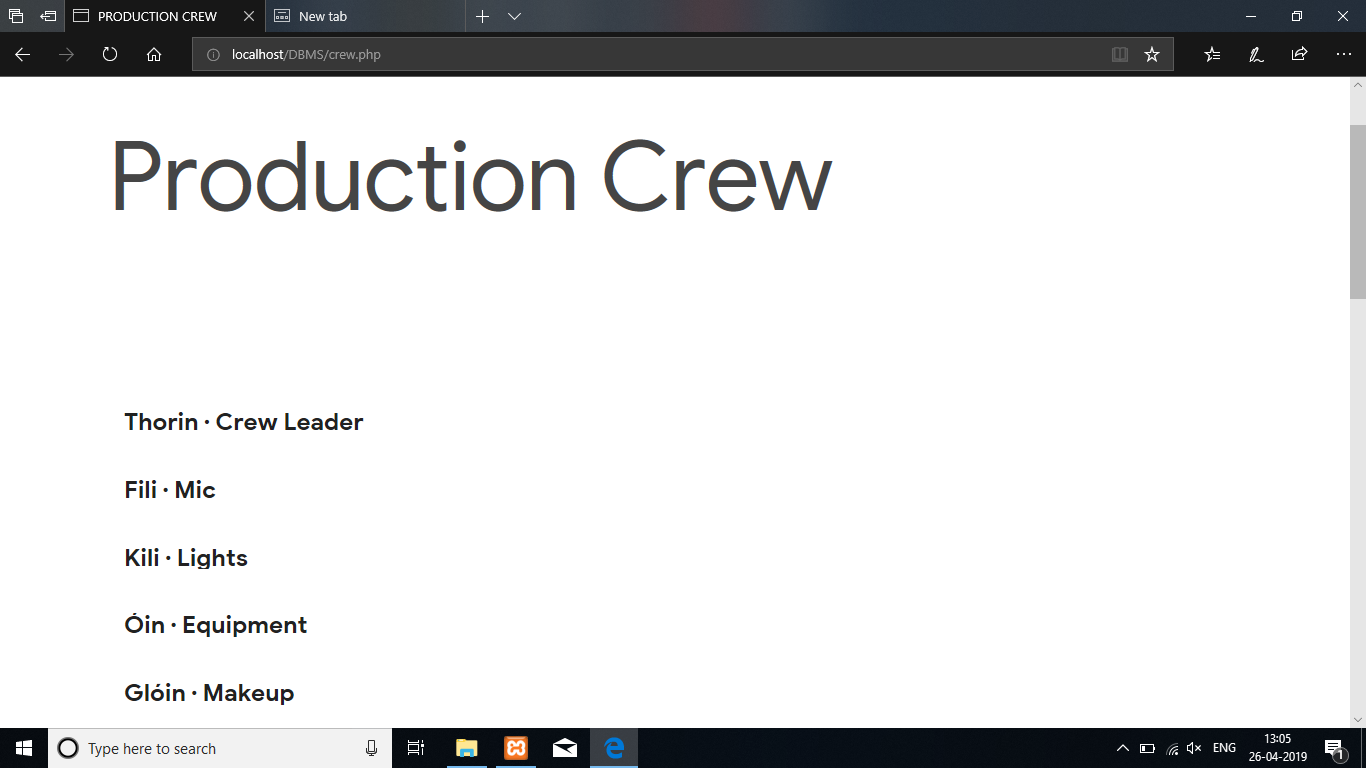
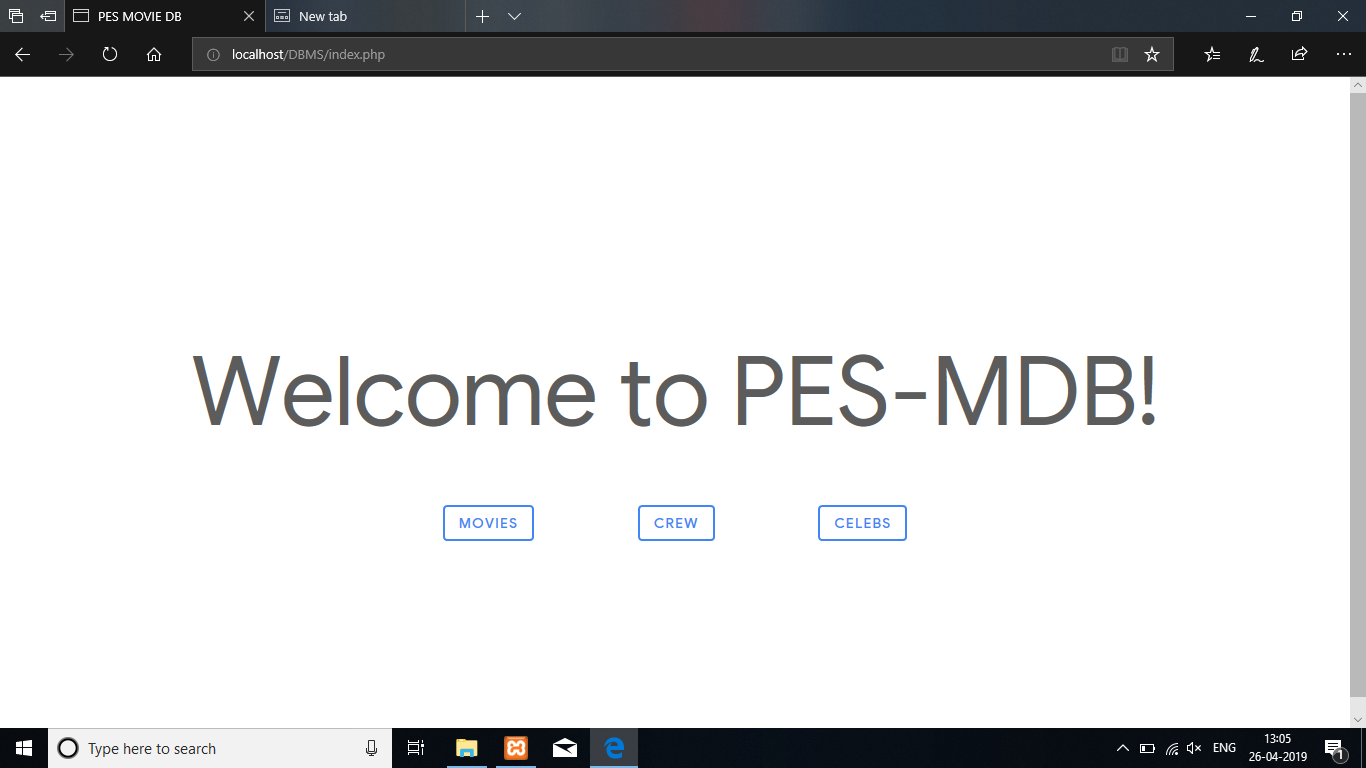
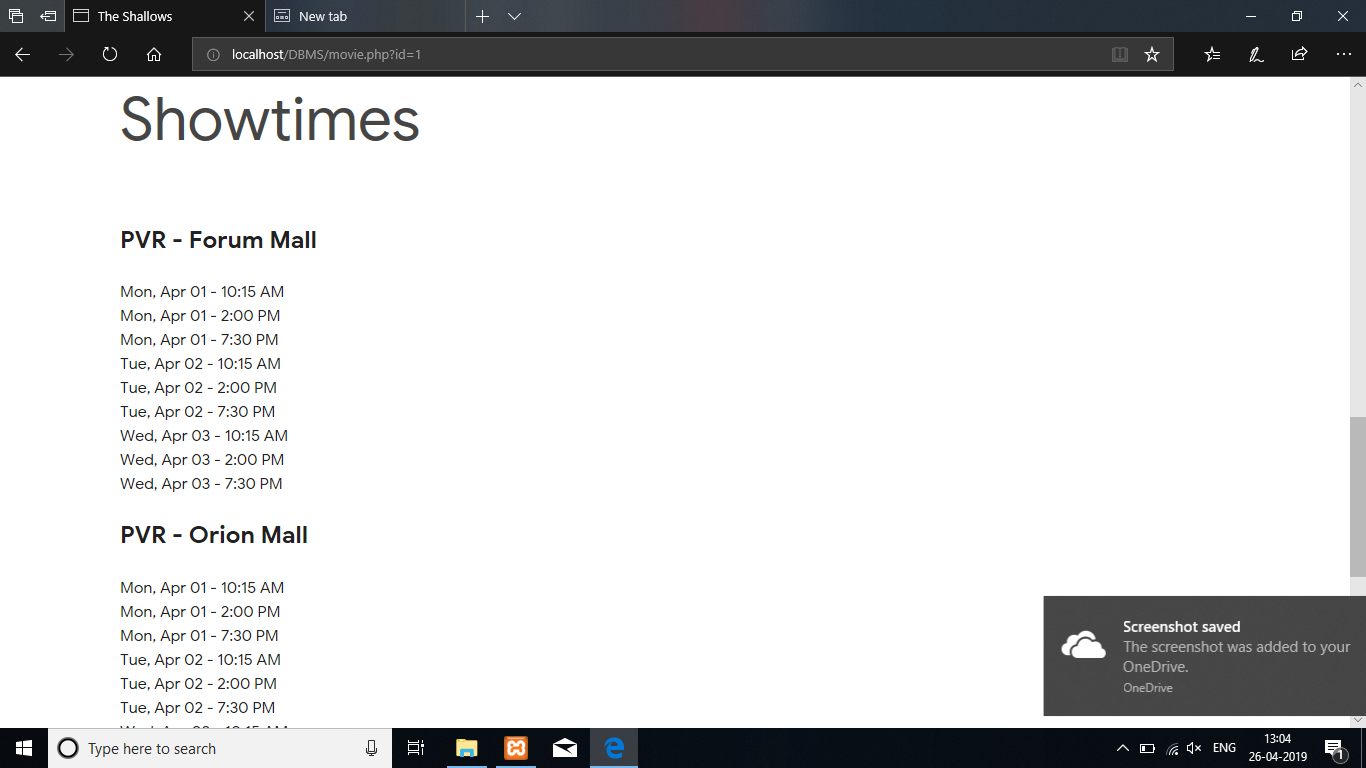
3|Top Gun|6.8|1986|English|Tony Scott|USA|110|3|15|5|3|Antagonist

4|The Terminal|7.3|2004|English|Steven Spielberg|USA|86|4|18.5|6|4|Lead

4|The Terminal|7.3|2004|English|Steven Spielberg|USA|86|4|18.5|7|4|Supporting

5|The Last of the Mohicans|7.8|1992|French|Michael Mann|USA|112|3|40|8|5|Lead

Web User Interface

# Conclusions and Proposed Enhancements

The movie database is restricted to displaying movie details and celebrity details. A proposed solution would be to implement an interactive website where the users and the clients can create online communities.

NOSQL can be used in place of relational database to enable video streaming for including trailers of movies and red-carpet interviews.

# References

* + - 1. W3 Schools
      2. Fundamentals of Database systems Ramez Elmasri and Shamkant Navathe

# Appedix

## Abbreviations

*SQL: Structured Query Language*

*CRUD: Create Read Update Delete*

*DDL: Data Definition Language*

*DML: Data Manipulation Language*

*ER Diagram: Entity Relationship Diagram*

## Tools and Technologies Used

1 XAMPP

2 PHP

3 MySQL

4 Microsoft Edge