**PES UNIVERSITY**

**Department of Computer Science & Engineering**



DBMS - UE20CS301

Mini Project

**Movie Booking System**

## Submitted to: Submitted By:

Dr. Geetha D Name: Tanay Singh

Associate Professor SRN: PES2UG20CS364

V Semester

Section F

**PES UNIVERSITY**

**Department of Computer Science & Engineering**



DBMS - UE20CS301

Mini Project

**Courier Management System**

## Submitted to: Submitted By:

Dr. Geetha D Name: Siddhant Verma

Associate Professor SRN: PES2UG20CS338

V Semester

Section F

# Table of Contents

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Title** | **Page No** |
| 1 | Short Description and Scope of the Project | 4 |
| 2 | ER Diagram | 5 |
| 3 | Relational Schema Diagram | 6 |
| 4 | DDL Statements | 7 |
| 5 | Populating Database | 8-10 |
| 6 | Join Queries | 11-13 |
| 7 | Aggregate Functions | 14-16 |
| 8 | Set Operations | 17-20 |
| 9 | Functions and Stored Procedures | 21-23 |
| 10 | Triggers | 24 |
| 11 | Frontend | 25-48 |
| 12 | Conclusion | 49 |

1. **Short Description and Scope of the**

# Project

# With the increase in the number of films in recent times, a large number of people are seen flocking towards movie theaters and cinemas for some daily entertainment.

# The excitement of these customers can however get possible crushed in the situation if there is a long line involved in the manual process of booking and buying tickets at the respective center.

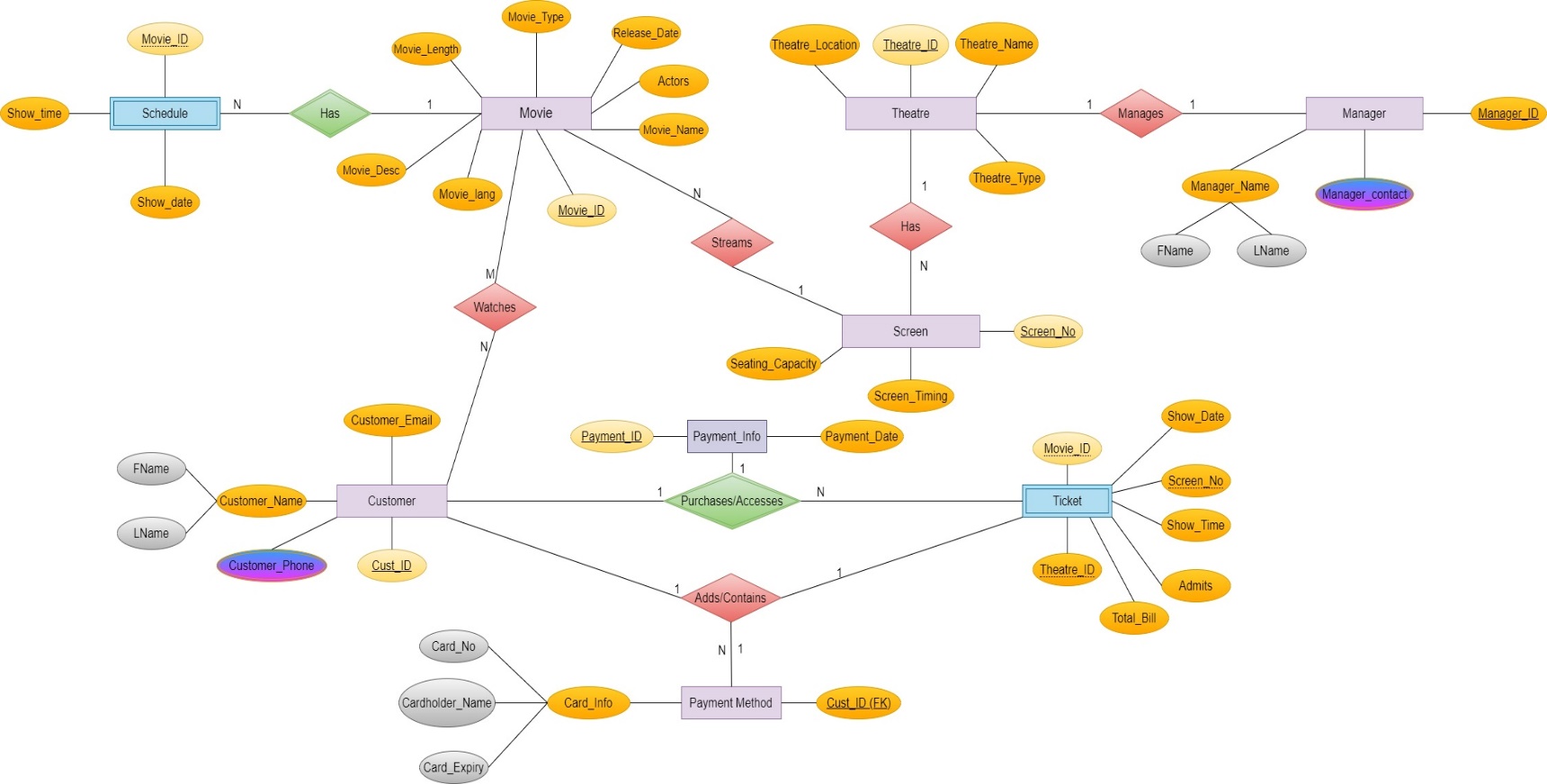
# To tackle these inconveniences, Online Movie Booking Systems were developed in order to help out the users to book tickets to their favorite movies using their Personal Systems or Smartphones at the convenience of their comfort.

# An Online Movie Booking System typically contains of a backend database that stores information such as Cinema information, Movie Information, User Information, Admin Related Features etc.

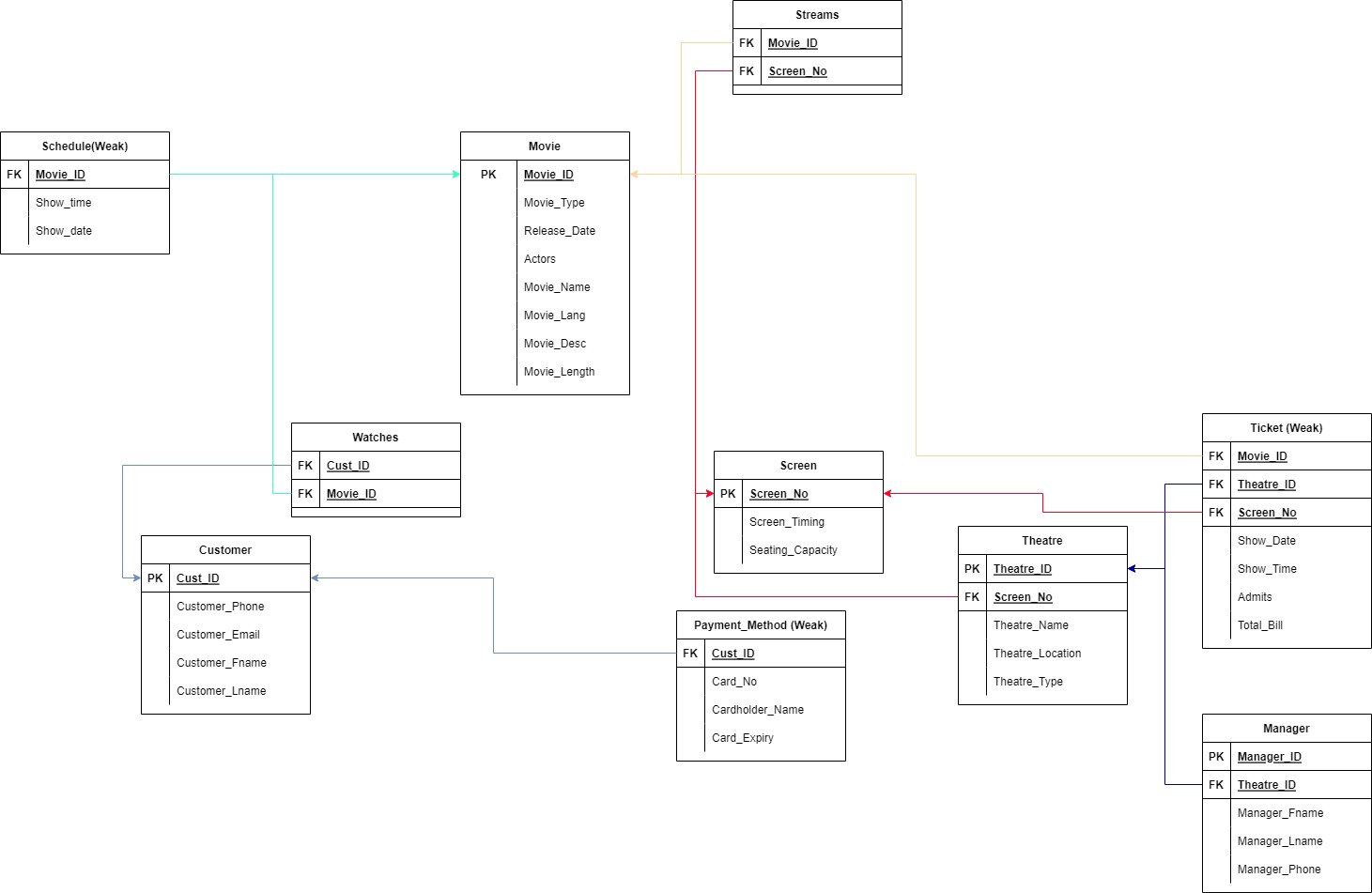
# The frontend is usually either hosted on a website for PC users and an Android/iOS app for Smartphone users.

Programming Tools Used :

1. Backend :
   1. MySQL
2. Middleware :
   1. Express JS
3. Frontend :
   1. HTML
   2. Boostrap (For CSS)
   3. EJS view engine
4. **ER Diagram**



# Relational Schema



1. **DDL statements - Building the database**

USE movie\_db;

CREATE table Movie(Movie\_ID int NOT NULL, Movie\_Type varchar(100), Release\_date Date NOT NULL, Cast varchar(100), Movie\_name varchar(50), Movie\_Lang varchar(10), Movie\_Desc varchar(200), Movie\_Length int, PRIMARY KEY(Movie\_ID));

CREATE table Customer(Cust\_ID int NOT NULL, Customer\_Phone bigint, Customer\_email varchar(50), Customer\_Fname varchar(50), Customer\_Lname varchar(50),primary key(Cust\_ID));

CREATE table Theatre(Theatre\_ID int NOT NULL, Screen\_No int NOT NULL, Theatre\_Name varchar(50) NOT NULL, Theatre\_Location varchar(50) NOT NULL, Theatre\_Type varchar(10) NOT NULL, primary key(Theatre\_ID), foreign key(Screen\_No) references Screen(Screen\_No));

CREATE table Ticket(Movie\_ID int NOT NULL, Theatre\_ID int NOT NULL, Screen\_No int NOT NULL, Show\_Date date NOT NULL, Show\_Time time NOT NULL, Admits int NOT NULL, Total\_Bill int NOT NULL, foreign key(Movie\_ID) references Movie(Movie\_ID), foreign key(Theatre\_ID) references Theatre(Theatre\_ID), foreign key(Screen\_No) references Screen(Screen\_No));

CREATE table Manager(Manager\_ID int NOT NULL, Theatre\_ID int NOT NULL, Manager\_Fname varchar(50) NOT NULL, Manager\_Lname varchar(50) NOT NULL, Manager\_Phone bigint, Primary Key(Manager\_ID), Foreign key(Theatre\_ID) references Theatre(Theatre\_ID));

# Populating the Database

1. Populating the Movie Table:

USE movie\_db;

INSERT INTO movie(Movie\_ID,Movie\_Type,Release\_Date,Movie\_Name,Movie\_Lang,Movie\_Desc,Movie\_Length)

Values

(1,'Action','2022-11-09','Black Panther: Wakanda Forever','EN',"Queen Ramonda, Shuri, M'Baku, Okoye and the Dora Milaje fight to protect their nation from intervening world powers in the wake of King T’Challa’s death. As the Wakandans strive to embrace their next chapter, the heroes must band together with the help of War Dog Nakia and Everett Ross and forge a new path for the kingdom of Wakanda.",161),

(2,'Horror','2022-10-06','Terrifier 2','EN',"After being resurrected by a sinister entity, Art the Clown returns to Miles County where he must hunt down and destroy a teenage girl and her younger brother on Halloween night. As the body count rises, the siblings fight to stay alive while uncovering the true nature of Art's evil intent.",148),

(3,'Action','2022-10-19','Black Adam','EN',"Nearly 5,000 years after he was bestowed with the almighty powers of the Egyptian gods—and imprisoned just as quickly—Black Adam is freed from his earthly tomb, ready to unleash his unique form of justice on the modern world.",125),

(4,'Horror','2022-07-27','Orphan: First Kill','EN',"After escaping from an Estonian psychiatric facility, Leena Klammer travels to America by impersonating Esther, the missing daughter of a wealthy family. But when her mask starts to slip, she is put against a mother who will protect her family from the murderous “child” at any cost.",99),

(5,'Anime','2022-08-06','One Piece Film Red','JP',"Uta — the most beloved singer in the world. Her voice, which she sings with while concealing her true identity, has been described as 'otherworldly.' She will appear in public for the first time at a live concert. As the venue fills with all kinds of Uta fans — excited pirates, the Navy watching closely, and the Straw Hats led by Luffy who simply came to enjoy her sonorous performance — the voice that the whole world has been waiting for is about to resound.",115),

(6,'Romance','2022-11-11','Falling for Christmas','EN','A newly engaged and spoiled hotel heiress finds herself in the care of a handsome, blue-collar lodge owner and his precocious daughter after getting total amnesia in a skiing accident.',93),

(7,'Romance','2022-10-06','Ticket to Paradise','EN','A divorced couple teams up and travels to Bali to stop their daughter from making the same mistake they think they made 25 years ago.',104),

(8,'Drama','2022-10-07','All Quiet on the Western Front','EN','Paul Baumer and his friends Albert and Muller, egged on by romantic dreams of heroism, voluntarily enlist in the German army. Full of excitement and patriotic fervour, the boys enthusiastically march into a war they believe in. But once on the Western Front, they discover the soul-destroying horror of World War I.',143),

(9,'Drama','2022-11-11','Suzume','JP','A modern action adventure road story where a 17-year-old girl named Suzume helps a mysterious young man close doors from the outer side that are releasing disasters all over in Japan.',122),

(10,'Anime','2022-09-24','Jujutsu Kaisen 0','JP',"Yuta Okkotsu is a nervous high school student who is suffering from a serious problem—his childhood friend Rika has turned into a curse and won't leave him alone. Since Rika is no ordinary curse, his plight is noticed by Satoru Gojo, a teacher at Jujutsu High, a school where fledgling exorcists learn how to combat curses. Gojo convinces Yuta to enroll, but can he learn enough in time to confront the curse that haunts him?",105);

1. Populating the Managers Table:

USE movie\_db;

INSERT INTO Manager(Manager\_ID,Theatre\_ID,Manager\_Fname,Manager\_Lname,Manager\_Phone)

values

(1,1,'Paul','Jacob',7794951978),

(2,5,'Vijay','Prasad',9345022878),

(3,3,'Akhilesh','Kumar',9867745803),

(4,2,'Shabaz','Ahmed',9845736123),

(5,4,'Jatinder', 'Singh',9025590255)

1. Populating the Theatre Table:

USE movie\_db;

INSERT INTO Theatre(Theatre\_ID,Theatre\_Name,Theatre\_Location,Theatre\_Type)

values

(1, 'PVR (Phoneix Marketcity)', 'Mahadevpura,Bengaluru', 'Single-Screen')

(2, 'PVR-IMAX (VR Mall)', 'Mahadevpura,Bengaluru', 'Single-Screen')

(3, 'Inox (Gopalan Signature Mall)', 'Old Madaras Road,Bengaluru', 'Single Screen')

(4, 'PVR (Soulspace Spirit Mall)', 'Bellandur,Bengaluru', 'Single Screen')

(5, 'Meenakshi Cinema Hall', 'Marathalli Multiplex,Bengaluru', 'Single Screen')

The Customer and the Ticket Tables are populated via the Frontend.

1. **Join Queries**

Showcase at least 4 join queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

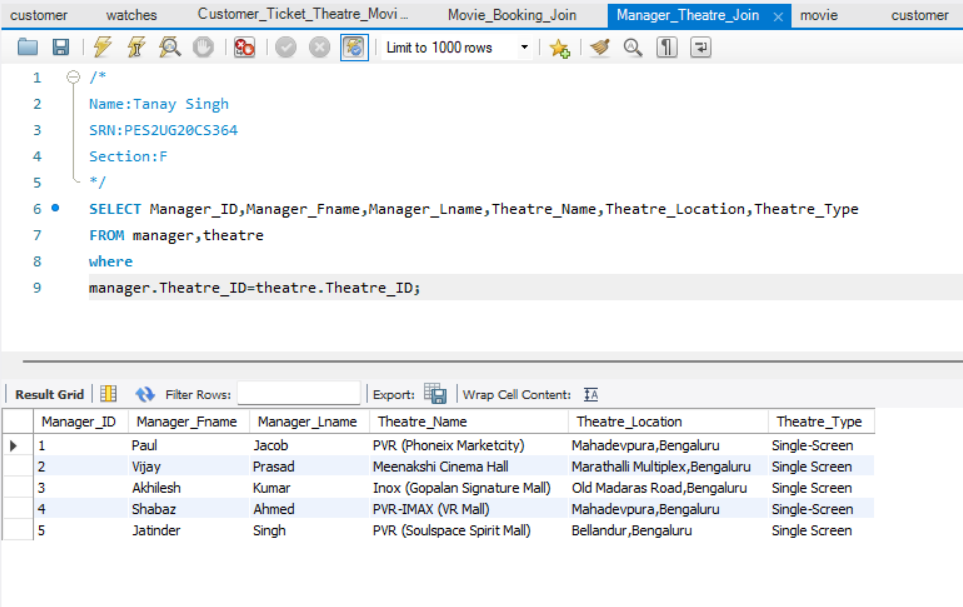
1. Find out the Managers corresponding to their respective Theatres

SELECT Manager\_ID,Manager\_Fname,Manager\_Lname,Theatre\_Name,Theatre\_Location,Theatre\_Type

FROM manager,theatre

where

manager.Theatre\_ID=theatre.Theatre\_ID;

Screenshot:

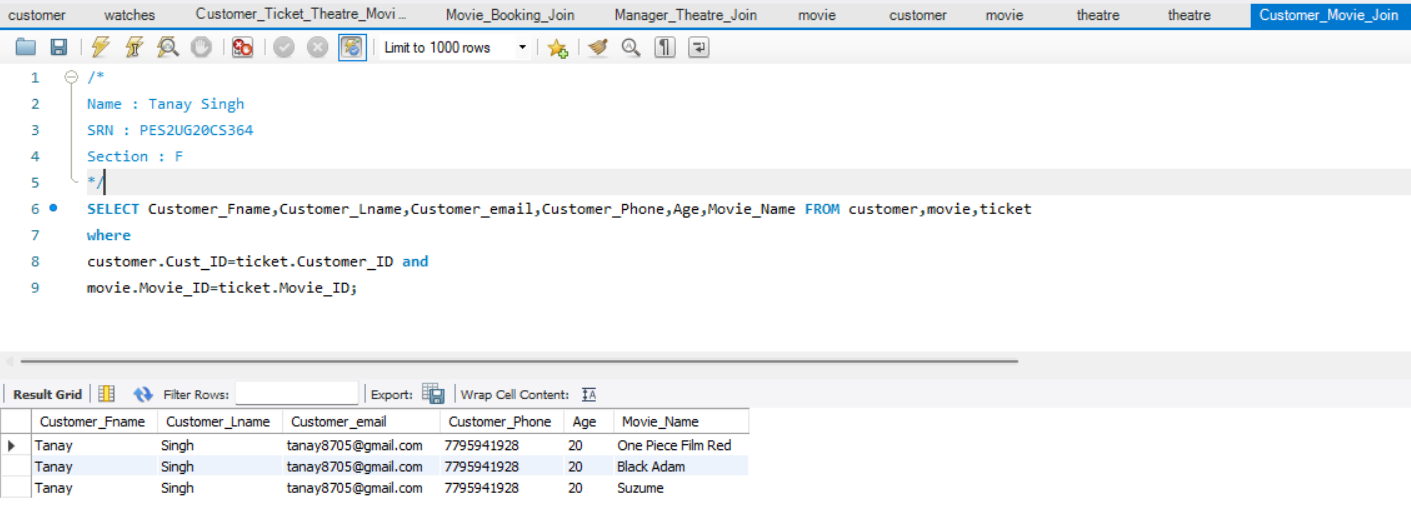
1. Find out which customer is watching which movie

SELECT Customer\_Fname,Customer\_Lname,Customer\_email,Customer\_Phone,Age,Movie\_Name FROM customer,movie,ticket

where

customer.Cust\_ID=ticket.Customer\_ID and

movie.Movie\_ID=ticket.Movie\_ID;



1. Join the customer ,ticket ,theatre and the movie tables to find out how many customers are visiting the theatre for different movies

SELECT Customer\_Fname,Customer\_Lname,Movie\_Name,Movie\_Lang,Movie\_Length,Movie\_Type,Theatre\_Name,Theatre\_Location,Show\_Date,Show\_Time,Theatre\_Type

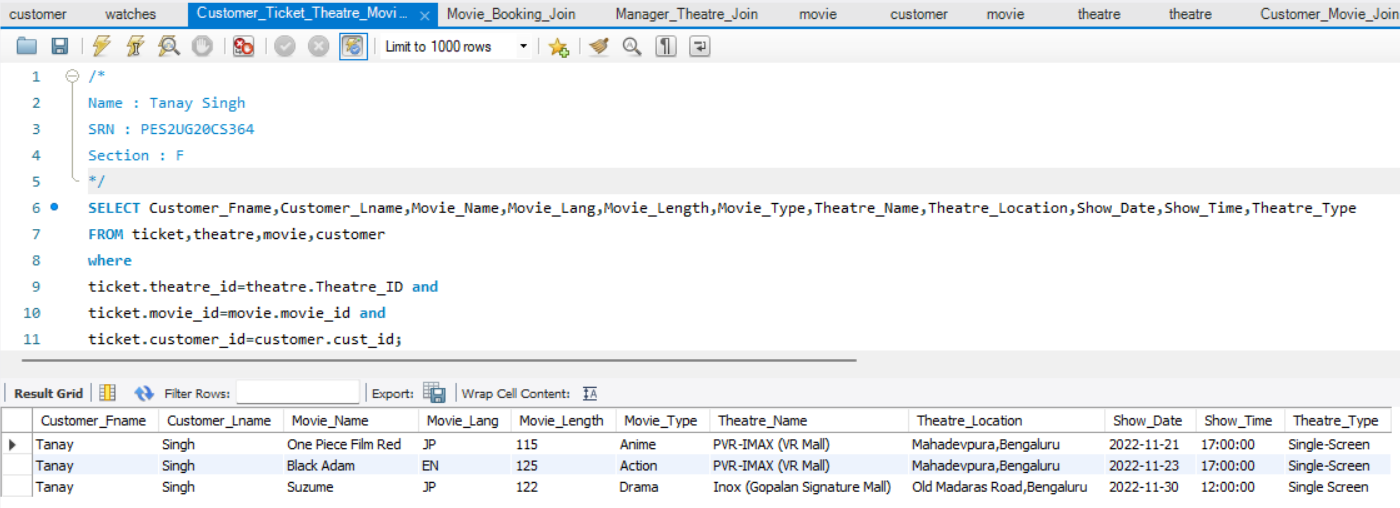
FROM ticket,theatre,movie,customer

where

ticket.theatre\_id=theatre.Theatre\_ID and

ticket.movie\_id=movie.movie\_id and

ticket.customer\_id=customer.cust\_id;



1. Find out how many people are attending a particular movie

USE movie\_db;

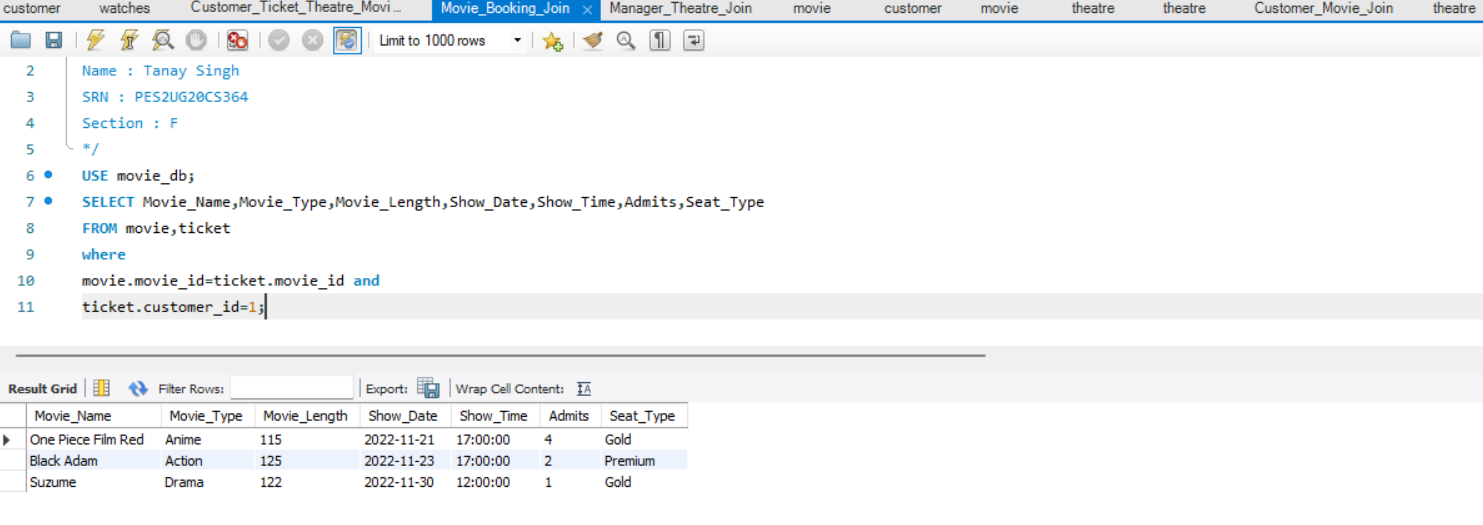
SELECT Movie\_Name,Movie\_Type,Movie\_Length,Show\_Date,Show\_Time,Admits,Seat\_Type

FROM movie,ticket

where

movie.movie\_id=ticket.movie\_id and

ticket.customer\_id=1;



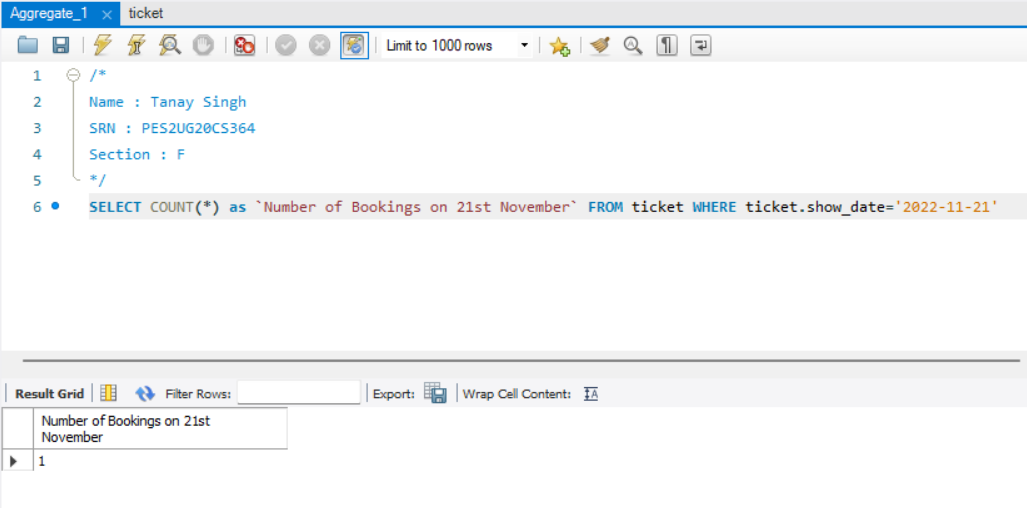
# Aggregate Functions

Showcase at least 4 Aggregate function queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

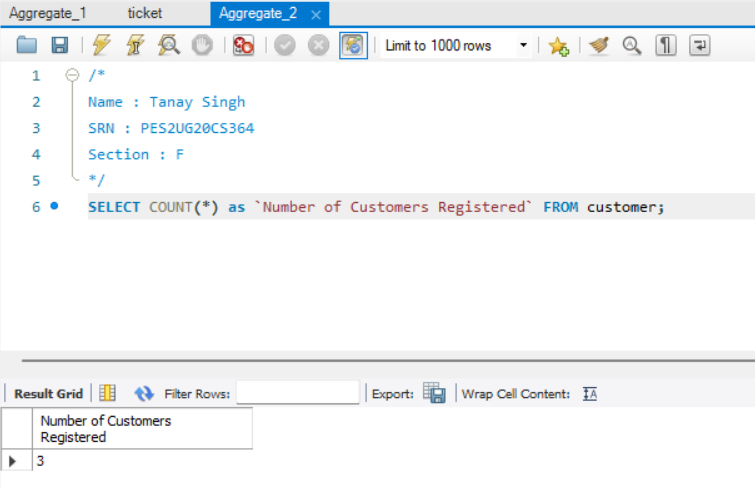
1. Show the total number of bookings on the Date: ‘21st November 2022’:

SELECT COUNT(\*) as `Number of Bookings on 21st November` FROM ticket WHERE ticket.show\_date='2022-11-21'

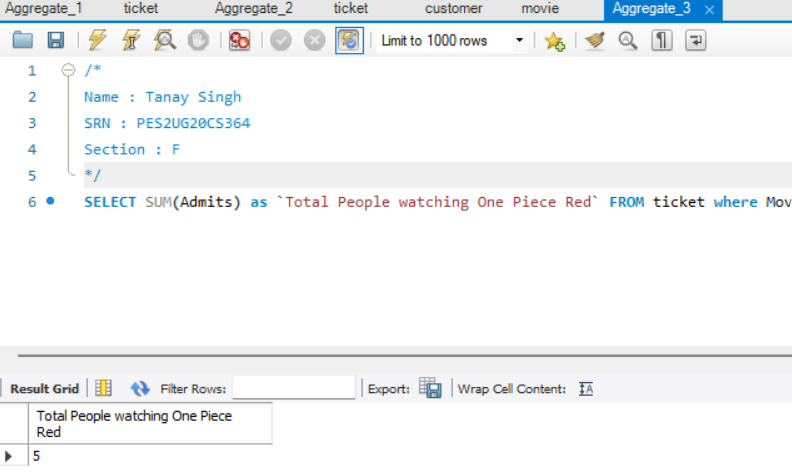


1. Show the total number of customers registered to the movie booking system:

SELECT COUNT(\*) as `Number of Customers Registered` FROM customer;

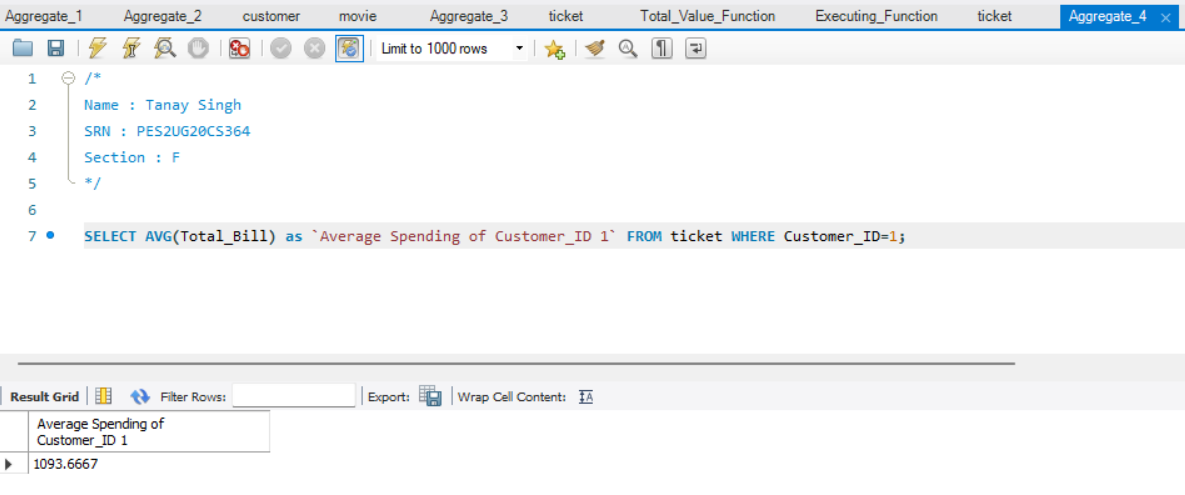


1. Show the total number of people watching the movie : “One Piece : Red”

SELECT SUM(Admits) as `Total People watching One Piece Red` FROM ticket where Movie\_ID=5;

1. Calculate the average spending of User with Customer\_ID = 1:

SELECT AVG(Total\_Bill) as `Average Spending of Customer\_ID 1` FROM ticket WHERE Customer\_ID=1;



# Set Operations

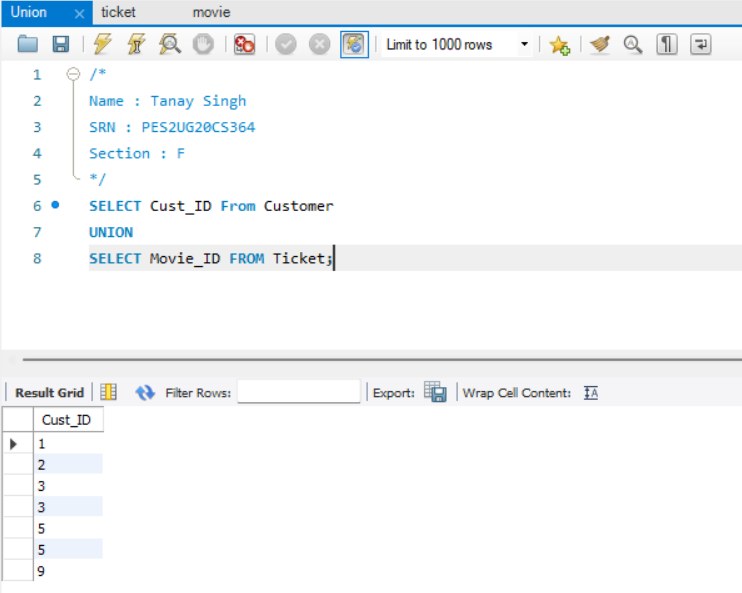
Showcase at least 4 Set Operations queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1. Union between Customer and Ticket Table

SELECT Cust\_ID From Customer

UNION

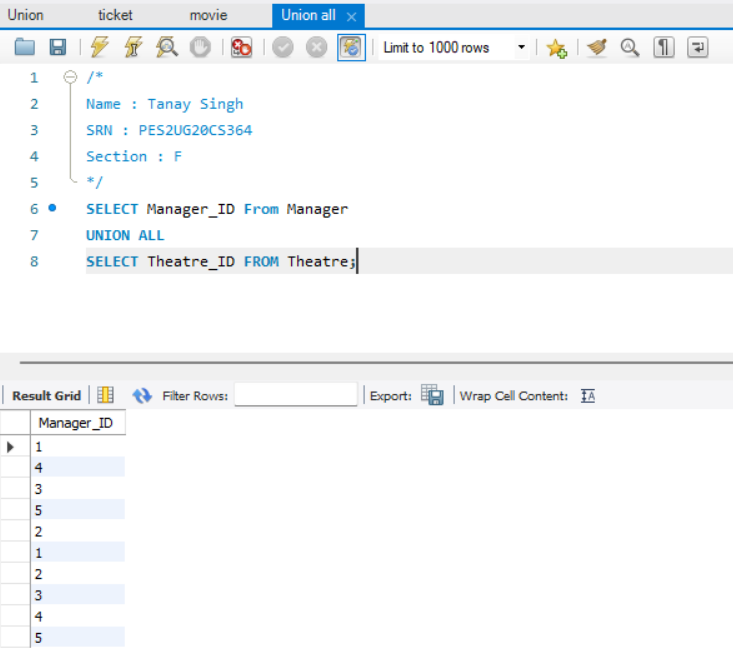
SELECT Movie\_ID FROM Ticket;

1. Union all between Manager and Theatre Table:

SELECT Manager\_ID From Manager

UNION ALL

SELECT Theatre\_ID FROM Theatre;

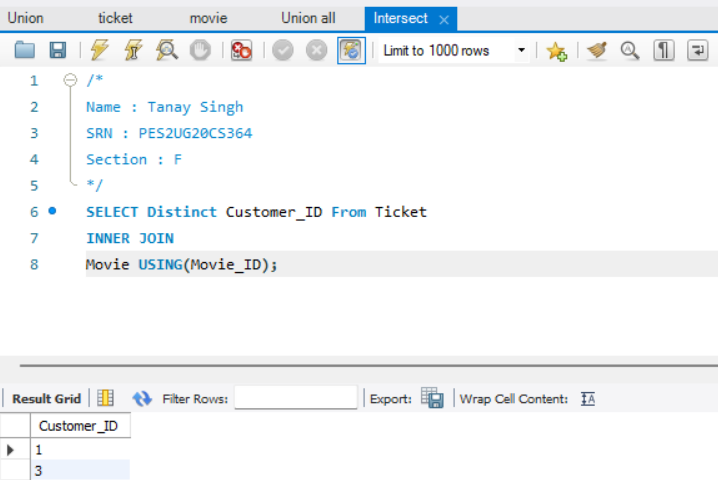


1. Intersection between Customers table and Movie table

SELECT Distinct Customer\_ID From Ticket

INNER JOIN

Movie USING(Movie\_ID);

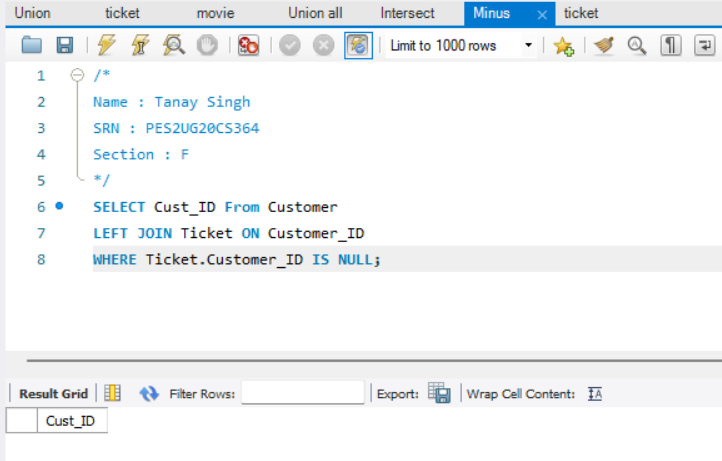


1. Minus

SELECT Cust\_ID From Customer

LEFT JOIN Ticket ON Customer\_ID

WHERE Ticket.Customer\_ID IS NULL;



# Functions and Procedures

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

1. Function to calculate Total Bill of Customer based on Admits and Seat type:

USE movie\_db;

DELIMITER $$

CREATE FUNCTION Calculate\_Total(Admits int,Seat\_Type varchar(10))

RETURNS int

deterministic

BEGIN

DECLARE total\_value INT;

IF Seat\_Type='Silver' THEN

SET total\_value=((320\*Admits)+(0.28\*320));

ELSEIF Seat\_Type='Gold' THEN

SET total\_value=((340\*Admits)+(0.28\*340));

ELSE

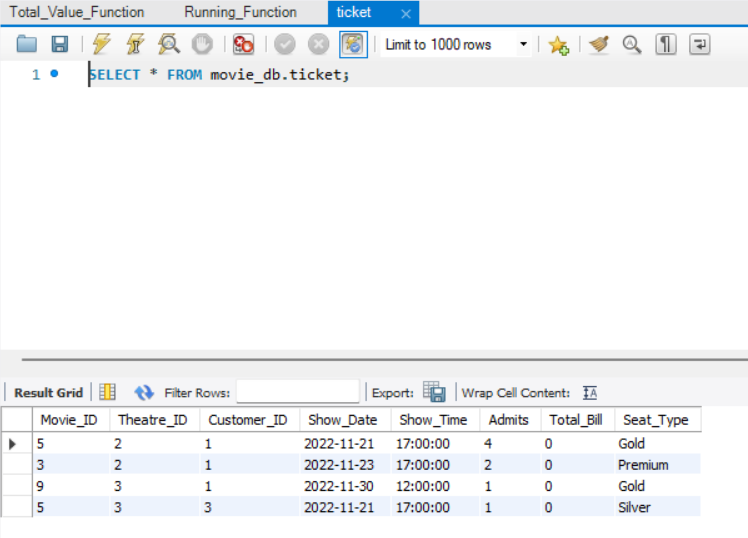
SET total\_value=((610\*Admits)+(0.28\*610));

END IF;

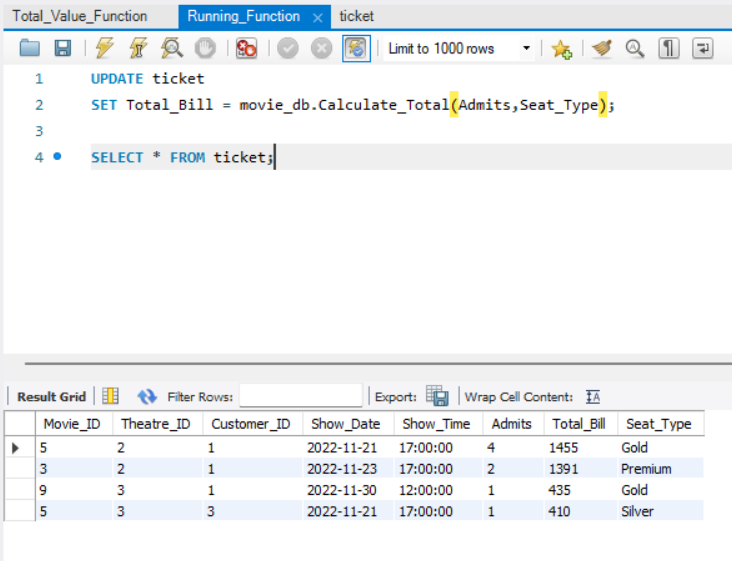
RETURN total\_value;

END;$$

DELIMITER ;

Before Running Function:

After Running Function :



1. Stored Procedure to calculate average spend based on customer id

USE movie\_db;

DELIMITER $$

CREATE PROCEDURE calculate\_average\_spend(Customer\_ID INT)

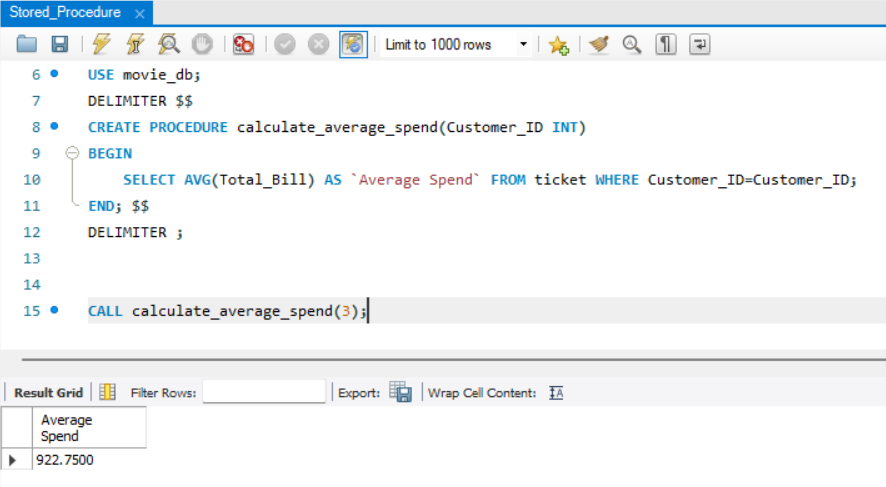
BEGIN

SELECT AVG(Total\_Bill) AS `Average Spend` FROM ticket WHERE Customer\_ID=Customer\_ID;

END; $$

DELIMITER ;

CALL calculate\_average\_spend(3);



# Triggers and Cursors

Create a Trigger and a Cursor. State the objective. Run and display the results.

1. Trigger to avoid user to book more than 10 seats for themselves.

USE movie\_db;

DELIMITER $$

CREATE trigger check\_number\_of\_admits

AFTER INSERT

ON ticket for EACH ROW

BEGIN

DECLARE error\_msg VARCHAR(255);

SET error\_msg = ('Cannot book more than 10 seats per account');

IF new.Admits>10 THEN

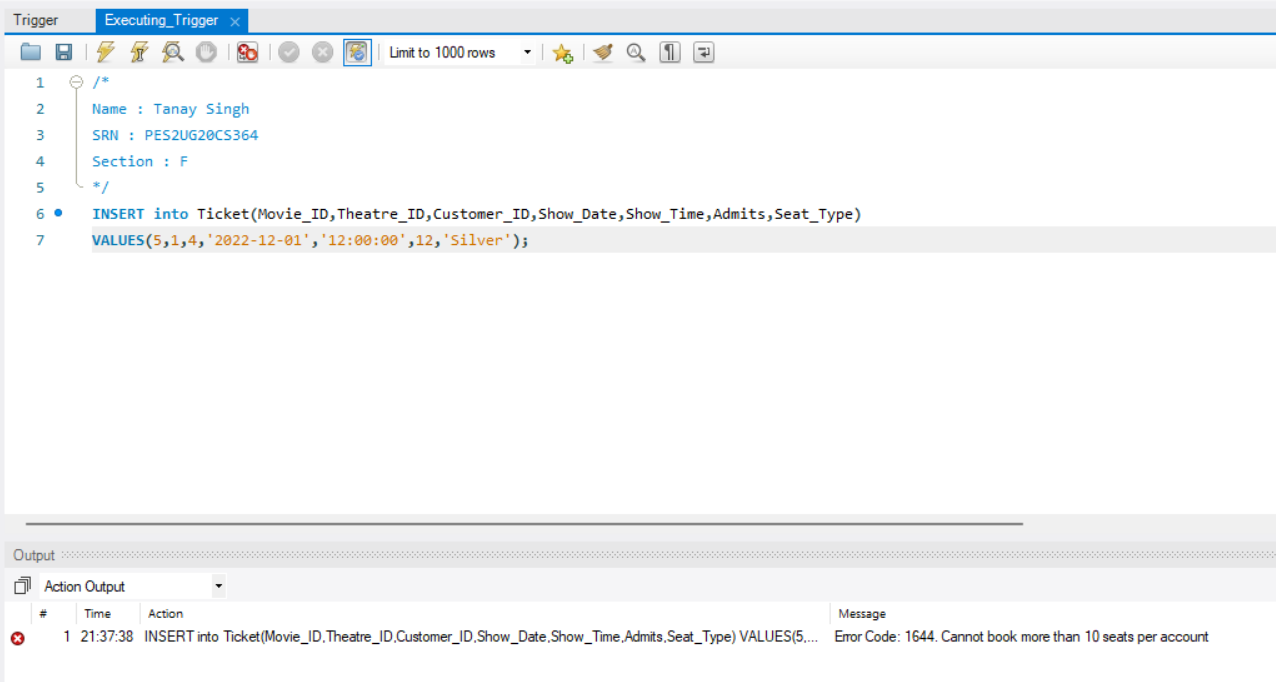
SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = error\_msg;

END IF;

END $$

DELIMITER ;

Screenshot:

# Developing a Frontend

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table
2. There should be an window to accept and run any SQL statement and display the result

Server Side Code

const express = require('express');

const app = express();

const path = require('path');

const bodyParser = require('body-parser');

const mysql = require('mysql2');

const {body} = require('express-validator');

const ejsMate = require('ejs-mate');

const ExpressError = require('./utils/ExpressError');

const methodOverride = require('method-override');

const connection = mysql.createConnection({

    host : 'localhost',

    user : 'root',

    password : 'Gola2002@#!',

    database : 'movie\_db'

});

connection.connect();

app.engine('ejs',ejsMate)

app.set('view engine','ejs');

app.set('views',path.join(\_\_dirname,'views'));

app.use(methodOverride('\_method'));

const jsonParser = bodyParser.json()

const urlencodedParser = bodyParser.urlencoded({extended:false})

app.get('/',(*req*,*res*)=>{

*res*.render('login.ejs');

})

app.get('/errorLogin',(*req*,*res*)=>{

*res*.render('errorLogin.ejs');

})

app.get('/register',(*req*,*res*)=>{

*res*.render('register.ejs')

})

app.get('/home',(*req*,*res*)=>{

    var passedName = *req*.query.user;

*res*.render('home.ejs',{username:passedName})

})

app.post('/',urlencodedParser,async(*req*,*res*)=>{

    connection.query(`SELECT \* FROM CUSTOMER where Username='${*req*.body.username}' and Password='${*req*.body.password}'`,function (*error*,*results*,*fields*) {

        if(*error*) throw *error*;

        if(*results*.length===0){

*res*.redirect("/errorLogin")

        }

        else{

            let user\_id = *results*[0].Cust\_ID

*res*.redirect(`/book\_movies?user=${*req*.body.username}&user\_id=${user\_id}`)

        }

    })

});

app.post('/register',urlencodedParser,async(*req*,*res*)=>{

        cust\_id = 0;

        connection.query('SELECT COUNT(\*) FROM Customer', function (*error*, *results*, *fields*) {

            if (*error*) throw *error*;

            cust\_id = *results*[0]['COUNT(\*)']+1;

            console.log(cust\_id)

            connection.query(`INSERT INTO Customer(Cust\_ID,Customer\_Phone,Customer\_email,Customer\_Fname,Customer\_Lname,Username,Password,Age) VALUES (${cust\_id},${*req*.body.phone\_number},'${*req*.body.email}','${*req*.body.first\_name}','${*req*.body.last\_name}','${*req*.body.username}','${*req*.body.password}',${*req*.body.age})`,

            function(*error*,*results*,*fields*){

                if(*error*) throw *error*;

*res*.redirect(`/book\_movies?user=${Boolean(*req*.body.username)?*req*.body.username:'Guest'}&user\_id=${cust\_id}`);

        })

    });

})

app.get('/book\_movies',(*req*,*res*)=>{

    connection.query('SELECT \* FROM movie',function(*error*,*results*,*fields*){

        if(*error*) throw *error*;

        let user = *req*.query.user

        if(typeof(user)==="undefined")

            user = 'Guest'

        user\_id = *req*.query.user\_id

        movie\_ids = []

        movie\_types = []

        movie\_release\_dates = []

        movie\_names = []

        movie\_lang = []

        movie\_desc = []

        movie\_length = []

        image\_sources = ['https://m.media-amazon.com/images/M/MV5BNTM4NjIxNmEtYWE5NS00NDczLTkyNWQtYThhNmQyZGQzMjM0XkEyXkFqcGdeQXVyODk4OTc3MTY@.\_V1\_FMjpg\_UX1000\_.jpg','https://m.media-amazon.com/images/M/MV5BNzEzOTZjYjYtMTI3MS00NWE1LWIyZTktYTRlMTBlYjlkMjZiXkEyXkFqcGdeQXVyMTUzMTg2ODkz.\_V1\_.jpg','https://www.theilluminerdi.com/wp-content/uploads/2022/09/black-adam-heroes.jpg','https://m.media-amazon.com/images/M/MV5BZjgwNDA3MmUtMTQ3Yy00ZDFmLTgwMTktNTBlMTQ3ZTI5MjYxXkEyXkFqcGdeQXVyMTM1MTE1NDMx.\_V1\_FMjpg\_UX1000\_.jpg','https://m.media-amazon.com/images/M/MV5BNDBhM2IxMzItOTdmZi00NmMzLWFlNTgtN2E2MDRjODU4NzEzXkEyXkFqcGdeQXVyODMyNTM0MjM@.\_V1\_.jpg','https://m.media-amazon.com/images/M/MV5BMDNhMWUxMjctZWFhNy00MjUzLWE3NjYtNTE4ZjVhYjRkZThmXkEyXkFqcGdeQXVyMTkxNjUyNQ@@.\_V1\_FMjpg\_UX1000\_.jpg','https://m.media-amazon.com/images/M/MV5BZWY2MGIxYjMtNGRiMi00ODczLWFlZjItYzE5YTdmNzdlNDEwXkEyXkFqcGdeQXVyMTMzODk3NDU0.\_V1\_.jpg','https://m.media-amazon.com/images/M/MV5BYTE1MmZiMWYtYTFmZi00YjA3LWI2ODgtMWFlNWYxZjdmNGE3XkEyXkFqcGdeQXVyNjk1Njg5NTA@.\_V1\_FMjpg\_UX1000\_.jpg','https://www.cwfilms.jp/en/news/images/12809e7f10e8ec3c6691c5c0c1be7cc50fe00ba6.jpg','https://m.media-amazon.com/images/M/MV5BYzFmMjAwMDYtNzM0Zi00NjY2LWFjMjYtMGQ1OTRiZjk5YjJkXkEyXkFqcGdeQXVyMTMwODY5NDc2.\_V1\_.jpg']

*results*.forEach(function(*arrayItem*){

            movie\_ids.push(*arrayItem*.Movie\_ID)

            movie\_types.push(*arrayItem*.Movie\_Type)

            movie\_release\_dates.push(*arrayItem*.Release\_date.toDateString())

            movie\_names.push(*arrayItem*.Movie\_name)

            movie\_lang.push(*arrayItem*.Movie\_Lang)

            movie\_desc.push(*arrayItem*.Movie\_Desc)

            movie\_length.push(*arrayItem*.Movie\_Length)

        })

        console.log("User Id : ",user\_id);

*res*.render('book\_movies.ejs',{

            movie\_ids:movie\_ids,

            movie\_ids:movie\_ids,

            movie\_types:movie\_types,

            movie\_release\_dates:movie\_release\_dates,

            movie\_names:movie\_names,

            movie\_lang:movie\_lang,

            movie\_desc:movie\_desc,

            movie\_length:movie\_length,

            images : image\_sources,

            user:user,

            user\_id:user\_id

        });

    })

})

app.get('/book\_ticket',(*req*,*res*)=>{

    let movie\_id = *req*.query.movie\_id

    let user = *req*.query.user

    if(user==='Guest')

*res*.redirect('/')

    else{

        connection.query(`SELECT \* FROM Movie WHERE Movie\_ID=${movie\_id}`,function(*error*,*results*,*fields*){

            if (*error*) throw *error*;

*res*.render('booking.ejs',{

                movie\_id:movie\_id,

                movie\_name:*results*[0]['Movie\_name'],

                movie\_genre:*results*[0]['Movie\_Type'],

                movie\_lang:*results*[0]['Movie\_Lang'],

                username:user

            });

        })

    }

})

app.post('/book\_tickets',urlencodedParser,async(*req*,*res*)=>{

    let movie\_id = *req*.query.movie\_id

    let theatre\_name = *req*.body.theatre\_id;

    let theatre\_id = 0;

    if(theatre\_name==='PVR (Phoneix Marketcity)')

        theatre\_id = 1;

    else if(theatre\_name==='PVR-IMAX (VR Mall)')

        theatre\_id = 2;

    else if(theatre\_name==='Inox (Gopalan Signature Mall)')

        theatre\_id = 3;

    else if(theatre\_name==='PVR (Soulspace Spirit Mall)')

        theatre\_id = 4;

    else

        theatre\_id = 5;

    let username = *req*.query.username

    let user\_id = 0;

    connection.query(`SELECT Cust\_ID from Customer WHERE Username='${username}'`,function(*error*,*results*,*fields*){

        if(*error*) throw *error*;

        user\_id = *results*[0]['Cust\_ID']

        connection.query(`INSERT INTO Ticket(Movie\_ID,Theatre\_ID,Customer\_ID,Show\_Date,Show\_Time,Admits,Seat\_Type) VALUES (${movie\_id},${theatre\_id},${user\_id},'${*req*.body.show\_date}','${*req*.body.show\_time}',${*req*.body.admits},'${*req*.body.seat\_type}')`,function(*error*,*results*,*fields*){

            if(*error*) throw *error*;

        })

        connection.query(`UPDATE TICKET SET Total\_Bill = movie\_db.Calculate\_Total(Admits,Seat\_Type) WHERE Customer\_ID=${user\_id} and Movie\_ID=${movie\_id};`,function (*error*,*results*,*fields*){

            if(*error*) throw *error*;

        })

*res*.redirect(`/booked\_tickets?user\_id=${user\_id}`)

    });

})

app.get('/account',(*req*,*res*)=>{

    let user=*req*.query.user

    if(typeof(user)===undefined || user==='Guest')

*res*.redirect('/')

    else

*res*.render('account.ejs',{user:user})

})

app.get('/booked\_tickets',(*req*,*res*)=>{

    let user = *req*.query.user\_id;

    let username = *req*.query.user;

    let movie\_ids = [];

    let movie\_name = [];

    let movie\_type = [];

    let movie\_length = [];

    let show\_date = [];

    let show\_time = [];

    let admits = [];

    let seat\_type = [];

    let total\_bills = [];

    connection.query(`SELECT Username,ticket.Movie\_ID,ticket.Total\_Bill,Movie\_Name,Movie\_Type,Movie\_Length,Show\_Date,Show\_Time,Admits,Seat\_Type FROM movie,ticket,customer where movie.movie\_id=ticket.movie\_id and ticket.customer\_id=${user} and customer.cust\_id=${user};`,function(*error*,*results*,*fields*){

        if(*error*) throw *error*;

*results*.forEach(function(*arrayItem*){

            total\_bills.push(*arrayItem*.Total\_Bill)

            movie\_name.push(*arrayItem*.Movie\_Name)

            movie\_type.push(*arrayItem*.Movie\_Type)

            movie\_length.push(*arrayItem*.Movie\_Length)

            show\_date.push(*arrayItem*.Show\_Date.toDateString())

            show\_time.push(*arrayItem*.Show\_Time)

            admits.push(*arrayItem*.Admits)

            seat\_type.push(*arrayItem*.Seat\_Type)

            movie\_ids.push(*arrayItem*.Movie\_ID)

        })

*res*.render("booked\_tickets.ejs",{

            username:username,

            user:user,

            movie\_names:movie\_name,

            movie\_types:movie\_type,

            movie\_lengths:movie\_length,

            show\_dates:show\_date,

            show\_times:show\_time,

            admits:admits,

            seat\_types:seat\_type,

            movie\_ids:movie\_ids,

            total\_bills:total\_bills

        })

    })

})

app.get('/view\_theatres',(*req*,*res*)=>{

    let user = *req*.query.user

    let user\_id = *req*.query.user\_id

    connection.query(`SELECT Theatre\_Name,Theatre\_Location,Theatre\_Type FROM Theatre`,function(*error*,*results*,*fields*){

        if(*error*) throw *error*;

        theatre\_name = []

        theatre\_location = []

        theatre\_type = []

*results*.forEach(function(*arrayItem*){

            theatre\_name.push(*arrayItem*.Theatre\_Name)

            theatre\_location.push(*arrayItem*.Theatre\_Location)

            theatre\_type.push(*arrayItem*.Theatre\_Type)

        })

*res*.render('view\_theatre.ejs',{

            user:user,

            theatre\_locations:theatre\_location,

            theatre\_names:theatre\_name,

            theatre\_types: theatre\_type,

            user:user,

            user\_id:user\_id

        });

    })

})

app.get('/modify\_booking',(*req*,*res*)=>{

    let user = *req*.query.user

    let user\_id = *req*.query.user\_id

    let movie\_id = *req*.query.movie\_id

    connection.query(`SELECT Admits,Seat\_Type,Show\_Time FROM ticket WHERE Customer\_ID=${user\_id} AND Movie\_ID=${movie\_id}`,function(*error*,*results*,*fields*){

        if (*error*) throw *error*;

        admits = *results*[0].Admits;

        seat\_type = *results*[0].Seat\_Type;

        show\_time = *results*[0].Show\_Time;

*res*.render('modify\_booking.ejs',{

            user:user,

            user\_id:user\_id,

            movie\_id:movie\_id,

            admits:admits,

            seat\_type:seat\_type,

            show\_time:show\_time

        })

    })

})

app.put('/modify\_booking',urlencodedParser,async(*req*,*res*)=>{

    let movie\_id = *req*.query.movie\_id

    let username = *req*.query.username

    let user\_id = *req*.query.user\_id

    console.log(*req*.body)

    connection.query(`UPDATE ticket SET Seat\_Type='${*req*.body.seat\_type}',Admits=${*req*.body.admits},Show\_Time='${*req*.body.show\_time}' WHERE Customer\_ID=${user\_id} AND Movie\_ID=${movie\_id}`,function(*error*,*results*,*fields*){

        if(*error*) throw *error*;

        connection.query(`UPDATE TICKET SET Total\_Bill = movie\_db.Calculate\_Total(Admits,Seat\_Type) WHERE Customer\_ID=${user\_id} and Movie\_ID=${movie\_id};`,function (*error*,*results*,*fields*){

            if(*error*) throw *error*;

*res*.redirect(`/booked\_tickets?user\_id=${user\_id}&user=${username}`)

        })

    })

})

app.get('/delete\_booking',(*req*,*res*)=>{

    let user\_id = *req*.query.user\_id;

    let user = *req*.query.user

    let movie\_id = *req*.query.movie\_id

    connection.query(`SELECT ticket.Total\_Bill,Movie\_Name,Movie\_Type,Movie\_Length,Show\_Date,Show\_Time,Admits,Seat\_Type FROM movie,ticket,customer where ticket.movie\_id=${movie\_id} and movie.movie\_id=ticket.movie\_id and ticket.customer\_id=${user\_id} and customer.cust\_id=${user\_id};`,function(*error*,*results*,*fields*){

        if(*error*) throw *error*;

        let total\_bill = *results*[0].Total\_Bill

        let movie\_name = *results*[0].Movie\_Name

        let movie\_type = *results*[0].Movie\_Type

        let movie\_length = *results*[0].Movie\_Length

        let show\_date = *results*[0].Show\_Date

        let show\_time = *results*[0].Show\_Time

        let admits = *results*[0].Admits

        let seat\_type = *results*[0].Seat\_Type

*res*.render('delete\_booking.ejs',{

            user\_id:user\_id,

            user:user,

            movie\_id:movie\_id,

            total\_bill:total\_bill,

            movie\_name:movie\_name,

            movie\_type:movie\_type,

            movie\_length:movie\_length,

            show\_date:show\_date,

            show\_time:show\_time,

            admits:admits,

            seat\_type:seat\_type

        })

    })

})

app.delete('/delete\_booking',urlencodedParser,async(*req*,*res*)=>{

    let user\_id = *req*.query.user\_id

    let movie\_id = *req*.query.movie\_id

    let user = *req*.query.user

    connection.query(`DELETE FROM ticket WHERE Customer\_ID=${user\_id} AND Movie\_ID=${movie\_id}`,function(*error*,*results*,*fields*){

        if(*error*) throw *error*;

*res*.redirect(`/booked\_tickets?user\_id=${user\_id}&user=${user}`)

    })

})

app.all('\*', (*req*, *res*, *next*) => {

     next(new ExpressError('Page not Found', 404))

})

app.listen(8000,()=>{

    console.log("Listening to port 8000");

})

Login Page

<%- include('loginHead',{title:"Login Page"}) -%>

<body>

    <div *class*="d-flex flex-column min-vh-100 justify-content-center align-items-center border-bottom">

        <div>

            <h1 *class*="display-6 text-wrap">Please Enter your login credentials here</h1>

        </div>

        <br>

        <div>

            <form *method*="post" *action*="/">

                <p>

                    <label *for*="username">Username : </label>

                    <input *type*="text" *name*="username" *id*="username">

                </p>

                <p>

                    <label *for*="password">Password : </label>

                    <input *type*="password" *name*="password" *id*="password">

                </p>

                <div *class*="d-flex justify-content-center">

                    <p>

                        <button *type*="submit" *class*="btn btn-primary mb-3">Submit</button>

                    </p>

                </div>

            </form>

        </div>

        <div><button *onclick*="location.href = '/register';" *id*="myButton" *class*="btn btn-success">If you are a new user, please click this link</a></div>

    </div>

</body>

Registration Page

<%- include('registerHead',{title:"User Registration"}) -%>

<body>

    <div *class*="d-flex flex-column min-vh-100 justify-content-center align-items-center border-bottom">

        <div>

            <h1>Please enter your details here : </h1>

        </div>

        <br>

        <div>

            <form *method*="post" *action*="/register" *class*="mx-auto row g-3 needs-validation" *novalidate*>

                <div *class*="col-md-4">

                    <label *for*="first\_name" *class*="form-label">Enter your first name : </label>

                    <input *type*="text" *class*="form-control" *name*="first\_name" *id*="validationFName" *required*>

                    <div *class*="valid-feedback">Looks Good!</div>

                </div>

                <div *class*="col-md-4">

                    <label *for*="last\_name" *class*="form-label">Enter your last name : </label>

                    <input *type*="text" *class*="form-control" *name*="last\_name" *id*="validationLName" *required*>

                    <div *class*="valid-feedback">Looks Good!</div>

                </div>

                <div *class*="col-md-4">

                    <label *for*="phone\_number" *class*="form-label">Enter your phone number : </label>

                    <input *type*="tel" *id*="validationPhoneNumber" *class*="form-control" *name*="phone\_number" *pattern*="^(\+91[\-\s]?)?[0]?(91)?[789]\d{9}$" *aria-describedby*="inputGroupPrepend" *required*>

                    <div *class*="valid-feedback">Looks Good!</div>

                    <div *class*="invalid-feedback">Please enter a correct phone number.</div>

                </div>

                <div *class*="col-md-6">

                    <label *for*="email" *class*="form-label">Enter your email id : </label>

                    <input *type*="email" *class*="form-control" *id*="validationEmail" *name*="email" *required*>

                    <div *class*="valid-feedback">Looks Good</div>

                    <div *class*="invalid-feedback">Please enter a valid email id</div>

                </div>

                <div *class*="col-md-3">

                    <label *for*="username" *class*="form-label">Enter a suitable username : </label>

                    <input *type*="text" *id*="username" *name*="username" *onblur*="blurFunction()" *class*="form-control" *pattern*="^(?=[a-zA-Z0-9.\_]{8,20}$)(?!.\*[\_.]{2})[^\_.].\*[^\_.]$" *required*>

                    <div *class*="valid-feedback">Username unique</div>

                    <div *class*="invalid-feedback">Username already exists!</div>

                </div>

                <div *class*="col-md-3">

                    <label *for*="password">Enter a suitable password : </label>

                    <input *type*="password" *id*="password" *name*="password" *class*="form-control" *pattern*="^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{5,10}$" *required*>

                    <div *class*="valid-feedback">Looks Good</div>

                    <div *class*="invalid-feedback">Please enter the password in the following format :One special character, One Uppercase Letter, One Number, One lowercase letter , Min length : 5, Max Length : 10</div>

                </div>

                <div *class*="col-md-3">

                    <label *for*="age">Enter your age : </label>

                    <input *type*="number" *name*="age" *id*="age" *class*="form-control" *required*>

                </div>

                <div *class*="col-12">

                    <button *class*="btn btn-primary" *type*="submit">Register</button>

                </div>

            </form>

        </div>

    </div>

</body>

<script>

    var username\_box = document.getElementById('username');

    username\_box.addEventListener('blur',function(){

        let user\_name = username\_box.innerText;

        if(user\_name in user\_list){

            user\_name.classList.add('invalid-feedback')

        }

        else{

            user\_name.classList.add('valid-feedback')

        }

    })

    (() => {

        'use strict'

*// Fetch all the forms we want to apply custom Bootstrap validation styles to*

        const forms = document.querySelectorAll('.needs-validation')

*// Loop over them and prevent submission*

        Array.from(forms).forEach(*form* => {

*form*.addEventListener('submit', *event* => {

                if (!*form*.checkValidity()) {

*event*.preventDefault()

*event*.stopPropagation()

            }

*form*.classList.add('was-validated')

            }, false)

        })

    })()

</script>

## View Currently Available Movies

<!DOCTYPE *html*>

<html *lang*="en">

    <head>

        <meta *charset*="UTF-8">

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

        <meta *name*="viewport" *content*="width=device-width,initial-scale=1">

        <title>Current Movies</title>

        <nav *class*="navbar navbar-expand-lg" *style*="background-color: #e3f2fd;">

            <div *class*="container-fluid">

              <a *class*="navbar-brand" *href*="/account?user=<%=user%>">Welcome <%=user%></a>

              <button *class*="navbar-toggler" *type*="button" *data-bs-toggle*="collapse" *data-bs-target*="#navbarNavAltMarkup" *aria-controls*="navbarNavAltMarkup" *aria-expanded*="false" *aria-label*="Toggle navigation">

                <span *class*="navbar-toggler-icon"></span>

              </button>

              <div *class*="collapse navbar-collapse" *id*="navbarNavAltMarkup">

                <div *class*="navbar-nav">

                  <a *class*="nav-link" *href*="/view\_theatres?user=<%=user%>&user\_id=<%=user\_id%>">Theatres</a>

                  <a *class*="nav-link <%= user==='Guest'?'disabled':'nav-link'%>" *href*="/booked\_tickets?user\_id=<%=user\_id%>&user=<%=user%>">Booked Tickets</a>

                </div>

              </div>

            </div>

        </nav>

        <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" *rel*="stylesheet" *integrity*="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" *crossorigin*="anonymous">

        <script *src*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" *integrity*="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" *crossorigin*="anonymous"></script>

        <style>

            .card:hover {

                transform: scale(1.1);

            }

            .card{

                transition: transform 0.5s ease;

            }

        </style>

    </head>

    <body>

        <div *class*="d-flex justify-content-center align-items-center">

            <h1>Currently Available Movies</h1>

        </div>

        <br>

        <div *class*="row" *style*="margin-left: 20px;margin-right: 10px;">

            <% for (let i = 0;i<5;i++){%>

                <div *class*="col-sm-6 card d-flex flex-column mx-auto" *style*="width: 18rem;">

                    <img *src*=<%=images[i]%> *class*="card-img-top img-fluid rounded mx-auto d-block" *alt*="...">

                    <div *class*="card-body d-flex flex-column">

                        <h4 *class*="card-title"><%=movie\_names[i]%></h5>

                        <h5 *class*="card-subtitle mb-2 text-muted"><%=movie\_lang[i]%></h5>

                        <h6 *class*="card-subtitle mb-2 text-muted"><%=movie\_types[i]%></h6>

                        <p *class*="card-text"><%=movie\_desc[i]%></p>

                        <p *class*="card-text lead">Release date : <%=movie\_release\_dates[i]%></p>

                        <p *class*="card-text"><strong>Runtime : <%=movie\_length[i]%> minutes </strong></p>

                        <a *href*="/book\_ticket?movie\_id=<%=movie\_ids[i]%>&user=<%=user%>" *class*="btn btn-primary mt-auto">Book Movie</a>

                    </div>

                </div>

            <% } %>

        </div>

        <br>

        <br>

        <br>

        <div *class*="row" *style*="margin-left: 20px;margin-right: 10px;">

            <% for (let i = 5;i<10;i++){%>

                <div *class*="col-sm-6 card d-flex flex-column mx-auto" *style*="width: 18rem;">

                    <img *src*=<%=images[i]%> *class*="card-img-top img-fluid rounded mx-auto d-block" *alt*="...">

                    <div *class*="card-body d-flex flex-column">

                        <h5 *class*="card-title"><%=movie\_names[i]%></h5>

                        <h5 *class*="card-subtitle mb-2 text-muted"><%=movie\_lang[i]%></h5>

                        <h6 *class*="card-subtitle mb-2 text-muted"><%=movie\_types[i]%></h6>

                        <p *class*="card-text"><%=movie\_desc[i]%></p>

                        <p *class*="card-text lead">Release date : <%=movie\_release\_dates[i]%></p>

                        <p *class*="card-text"><strong>Runtime : <%=movie\_length[i]%> minutes </strong></p>

                        <a *href*="/book\_ticket?movie\_id=<%=movie\_ids[i]%>&user=<%=user%>" *class*="btn btn-primary mt-auto">Book Movie</a>

                    </div>

                </div>

            <% } %>

        </div>

    </body>

## View Theatres

<%- include('viewTheatreHead',{title:'View Theatres',user:user}) -%>

<body>

    <table *class*="table">

        <thead>

            <tr>

                <th *scope*="col">Theatre Name</th>

                <th *scope*="col">Theatre Location</th>

                <th *scope*="col">Theatre Type</th>

            </tr>

        </thead>

        <tbody>

            <% for(let i = 0;i<theatre\_names.length;i++){%>

                <tr>

                    <td><%=theatre\_names[i]%></td>

                    <td><%=theatre\_locations[i]%></td>

                    <td><%=theatre\_types[i]%></td>

                </tr>

            <% } %>

        </tbody>

    </table>

</body>

## View Booked Tickets

<!DOCTYPE *html*>

<html *lang*="en">

    <head>

        <meta *charset*="UTF-8">

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

        <nav *class*="navbar navbar-expand-lg" *style*="background-color: #e3f2fd;">

            <div *class*="container-fluid">

              <a *class*="navbar-brand" *href*="/book\_movies?user=<%=username%>&user\_id=<%=user%>">Movie Booking System</a>

              <button *class*="navbar-toggler" *type*="button" *data-bs-toggle*="collapse" *data-bs-target*="#navbarNavAltMarkup" *aria-controls*="navbarNavAltMarkup" *aria-expanded*="false" *aria-label*="Toggle navigation">

                <span *class*="navbar-toggler-icon"></span>

              </button>

            </div>

        </nav>

        <meta *name*="viewport" *content*="width=device-width,initial-scale=1">

        <title>Booked Tickets</title>

        <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" *rel*="stylesheet" *integrity*="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" *crossorigin*="anonymous">

        <script *src*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" *integrity*="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" *crossorigin*="anonymous"></script>

    </head>

    <body>

        <div *class*="row" *style*="margin-left: 20px;margin-right: 10px;">

            <% for(let i = 0;i<movie\_names.length;i++){ %>

                <div *class*="col-sm-6 card mb-3 flex-column mx-auto" *style*="max-width:18rem;margin-top: 2em;">

                    <div *class*="card-header">Booking <%=i+1%></div>

                    <div *class*="card-body d-flex flex-column">

                        <h5 *class*="card-title"><%=movie\_names[i]%></h5>

                        <h6 *class*="card-subtitle"><%=movie\_types[i]%></h6>

                        <p *class*="card-text">

                            Runtime : <%=movie\_lengths[i]%> minutes

                        </p>

                        <br>

                        <p *class*="card-text">

                            Show Date : <%=show\_dates[i]%>

                        </p>

                        <br>

                        <p *class*="card-text">

                            Show Time : <%=show\_times[i]%>

                        </p>

                        <br>

                        <p *class*="card-text">

                            Total seats booked : <%=admits[i]%>

                        </p>

                        <br>

                        <p *class*="card-text">

                            Seat Type : <%=seat\_types[i]%>

                        </p>

                        <br>

                        <h5 *class*="card-text">Total Spent : ₹ <%=total\_bills[i]%></h1>

                        <div *class*="row">

                            <div *class*="col-6">

                                <a *class*="btn btn-success" *href*="/modify\_booking?user\_id=<%=user%>&user=<%=username%>&movie\_id=<%=movie\_ids[i]%>">Modify Booking</a>

                            </div>

                            <div *class*="col">

                                <a *class*="btn btn-danger" *href*="/delete\_booking?user\_id=<%=user%>&user=<%=username%>&movie\_id=<%=movie\_ids[i]%>">Delete Booking</a>

                            </div>

                        </div>

                    </div>

                </div>

            <%} %>

        </div>

    </body>

## Update Bookings

<!DOCTYPE *html*>

<html *lang*="en">

    <head>

        <meta *charset*="UTF-8">

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

        <nav *class*="navbar navbar-expand-lg" *style*="background-color: #e3f2fd;">

            <div *class*="container-fluid">

              <a *class*="navbar-brand" *href*="#">Movie Booking System</a>

              <button *class*="navbar-toggler" *type*="button" *data-bs-toggle*="collapse" *data-bs-target*="#navbarNavAltMarkup" *aria-controls*="navbarNavAltMarkup" *aria-expanded*="false" *aria-label*="Toggle navigation">

                <span *class*="navbar-toggler-icon"></span>

              </button>

              <div *class*="collapse navbar-collapse" *id*="navbarNavAltMarkup">

                <div *class*="navbar-nav">

                  <a *class*="nav-link" *href*="/book\_movies?user=<%=user%>&user\_id=<%=user\_id%>">View currently streaming movies</a>

                </div>

              </div>

            </div>

        </nav>

        <meta *name*="viewport" *content*="width=device-width,initial-scale=1">

        <title>Update Booking</title>

        <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" *rel*="stylesheet" *integrity*="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" *crossorigin*="anonymous">

        <script *src*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" *integrity*="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" *crossorigin*="anonymous"></script>

    </head>

    <body>

        <div *class*="d-flex justify-content-center align-items-center flex-column"  *style*="margin-top: 5em;"">

            <div *class*="card text-bg-info border-primary mb-3" *style*="padding: 2em;">

                <form *class*="validated-form" *method*="post" *action*="/modify\_booking?\_method=PUT&movie\_id=<%=movie\_id%>&username=<%=user%>&user\_id=<%=user\_id%>" *novalidate*>

                    <div *class*="row">

                        <div *class*="col-6">

                            <label *class*="form-label" *for*="seat\_type">Select preferred seat type : </label>

                        </div>

                        <div *class*="col">

                            <select *class*="form-select" *name*="seat\_type" *id*="seat\_type" *aria-label*="Select Seats">

                                <option *value*="Silver" <%=seat\_type==='Silver'?'selected':''%>>Silver (Front-Row Seats) - (₹320)</option>

                                <option *value*="Gold" <%=seat\_type==='Gold'?'selected':''%>>Gold(Back-Row Seats) - (₹340)</option>

                                <option *value*="Premium" <%=seat\_type==='Premium'?'selected':''%>>Premium (Recliner Seats) - (₹610) </option>

                            </select>

                        </div>

                    </div>

                    <br>

                    <div *class*="row">

                        <div *class*="col-6">

                            <label *class*="form-label" *for*="admits">Enter the number of seats : </label>

                        </div>

                        <div *class*="col">

                            <input *class*="form-control" *type*="number" *name*="admits" *id*="admits" *min*="1" *value*="<%=admits%>" *required*>

                        </div>

                        <div *class*="valid-feedback">

                            Looks Good!

                        </div>

                        <div *class*="invalid-feedback">

                            Please select a valid number of admits

                        </div>

                    </div>

                    <br>

                    <div *class*="row">

                        <div *class*="col-6">

                            <label *class*="form-label" *for*="show\_time">Select the show time : </label>

                        </div>

                        <div *class*="col">

                            <select *class*="form-select" *name*="show\_time" *id*="show\_time" *aria-label*="Select Time show">

                                <option *value*="06:00:00" <%=show\_time==='06:00:00'?'selected':''%>>6 AM</option>

                                <option *value*="08:00:00" <%=show\_time==='08:00:00'?'selected':''%>>8 AM</option>

                                <option *value*="12:00:00" <%=show\_time==='12:00:00'?'selected':''%>>12 PM</option>

                                <option *value*="14:00:00" <%=show\_time==='14:00:00'?'selected':''%>>2 PM</option>

                                <option *value*="17:00:00" <%=show\_time==='17:00:00'?'selected':''%>>5 PM</option>

                                <option *value*="20:00:00" <%=show\_time==='20:00:00'?'selected':''%>>8 PM</option>

                                <option *value*="23:00:00"<%=show\_time==='23:00:00'?'selected':''%>>11 PM</option>

                            </select>

                        </div>

                    </div>

                    <br>

                    <br>

                    <div *class*="d-flex justify-content-center">

                        <button *type*="submit" *class*="btn btn-primary">Update Booking</button>

                    </div>

                </form>

            </div>

        </div>

    </body>

## Delete Booking

<!DOCTYPE *html*>

<html *lang*="en">

    <head>

        <meta *charset*="UTF-8">

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

        <nav *class*="navbar navbar-expand-lg" *style*="background-color: #e3f2fd;">

            <div *class*="container-fluid">

              <a *class*="navbar-brand" *href*="/book\_movies?user=<%=user%>&user\_id=<%=user\_id%>">Movie Booking System</a>

              <button *class*="navbar-toggler" *type*="button" *data-bs-toggle*="collapse" *data-bs-target*="#navbarNavAltMarkup" *aria-controls*="navbarNavAltMarkup" *aria-expanded*="false" *aria-label*="Toggle navigation">

                <span *class*="navbar-toggler-icon"></span>

              </button>

            </div>

        </nav>

        <meta *name*="viewport" *content*="width=device-width,initial-scale=1">

        <title>Delete Booking</title>

        <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" *rel*="stylesheet" *integrity*="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" *crossorigin*="anonymous">

        <script *src*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" *integrity*="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" *crossorigin*="anonymous"></script>

    </head>

    <body>

        <div *class*="row" *style*="margin-left: 20px;margin-right: 10px;">

                <div *class*="col-sm-6 card mb-3 flex-column mx-auto" *style*="max-width:18rem;margin-top: 2em;">

                    <div *class*="card-header">Current Booking</div>

                    <div *class*="card-body d-flex flex-column">

                        <h5 *class*="card-title"><%=movie\_name%></h5>

                        <h6 *class*="card-subtitle"><%=movie\_type%></h6>

                        <p *class*="card-text">

                            Runtime : <%=movie\_length%> minutes

                        </p>

                        <br>

                        <p *class*="card-text">

                            Show Date : <%=show\_date%>

                        </p>

                        <br>

                        <p *class*="card-text">

                            Show Time : <%=show\_time%>

                        </p>

                        <br>

                        <p *class*="card-text">

                            Total seats booked : <%=admits%>

                        </p>

                        <br>

                        <p *class*="card-text">

                            Seat Type : <%=seat\_type%>

                        </p>

                        <br>

                        <h5 *class*="card-text">Total Spent : ₹ <%=total\_bill%></h5>

                        <br>

                        <div *class*="alert alert-info" *role*="alert">

                            <a *href*="/booked\_tickets?user\_id=<%=user\_id%>&user=<%=user%>" *class*="btn btn-success" *role*="button">Go Back</a>

                        </div>

                        <form *class*="validated-form" *method*="post" *action*="/delete\_booking?\_method=DELETE&user\_id=<%=user\_id%>&user=<%=user%>&movie\_id=<%=movie\_id%>" *novalidate*>

                            <div *class*="alert alert-warning" *role*="alert">

                                Are you sure you want to delete this booking?

                            </div>

                            <div *class*="d-flex justify-content-center">

                                <button *type*="submit" *class*="btn btn-danger">Delete</button>

                            </div>

                        </form>

                    </div>

                </div>

        </div>

    </body>

## Login Page Screenshot

## 

## Movie Page Screenshot:

## 

## Theatres Page Screenshot:

## 

## Booked Tickets

## 

## Modify Booking

## 

## Delete Booking

## 

## 12. Conclusion

The set goals while starting this project as mentioned in the synopsis have been fulfilled, including a frontend with basic CRUD operations.

Future iterations of this project include, OAuth user account authentication, Usage of front-end libraries such as ReactJS to make the website more dynamic in nature and inclusion of more tables in the backend to expand the range of this project.

**References (If any)**

1. **Web Development Bootcamp -** <https://www.udemy.com/course/the-web-developer-bootcamp/>
2. **Mysql2 documentation -** <https://www.npmjs.com/package/mysql2>