

PES UNIVERSITY
Department of Computer Science & Engineering



DBMS - UE20CS301
Mini Project
Movie Booking System

Submitted to:

Dr. Geetha D
Associate Professor

Submitted By:

Name: Tanay Singh
SRN: PES2UG20CS364
V Semester
Section F

PES UNIVERSITY
Department of Computer Science & Engineering



DBMS - UE20CS301
Mini Project
Courier Management System

Submitted to:

Dr. Geetha D
Associate Professor

Submitted By:

Name: Siddhant Verma
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 :

a. MySQL

2) Middleware :

a. Express JS

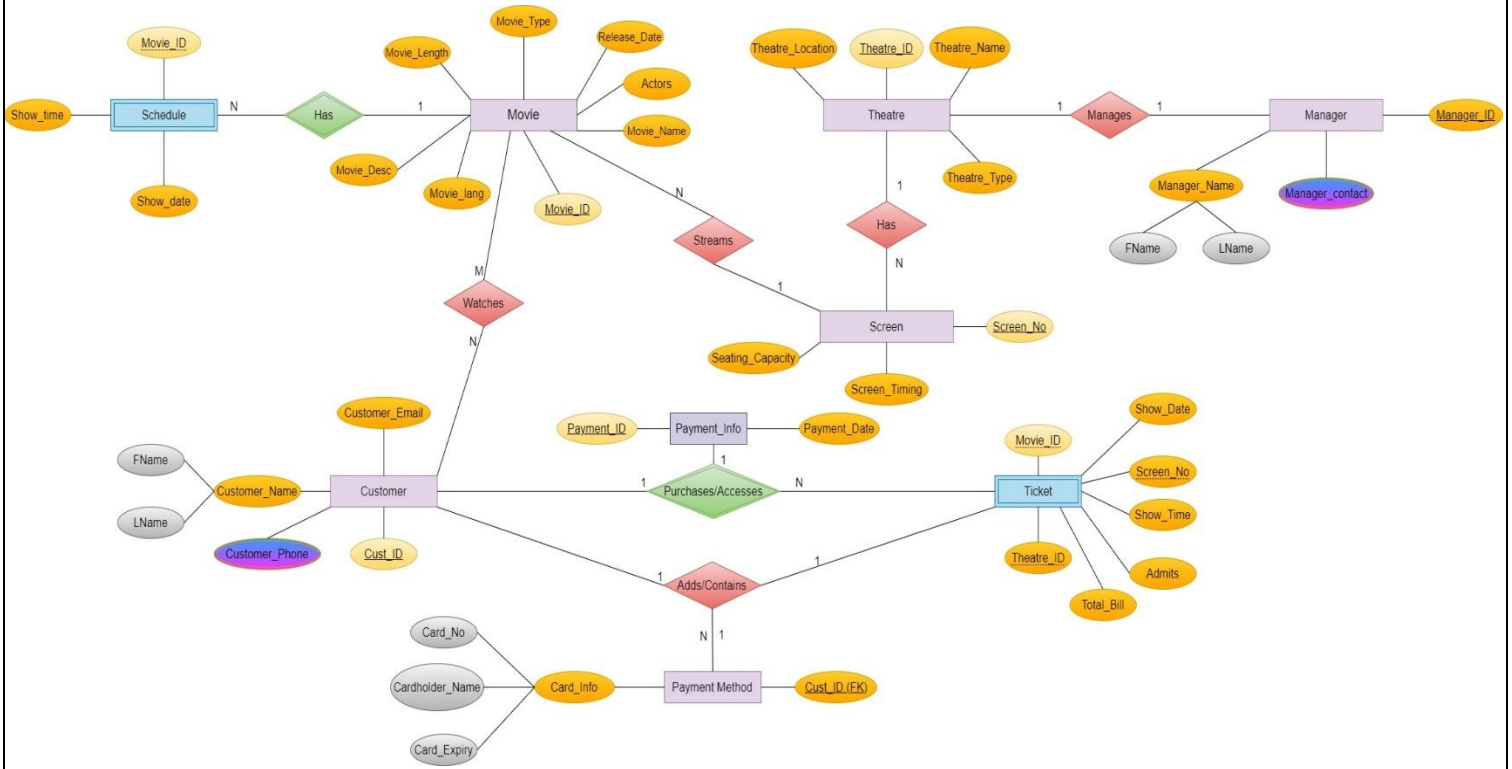
3) Frontend :

a. HTML

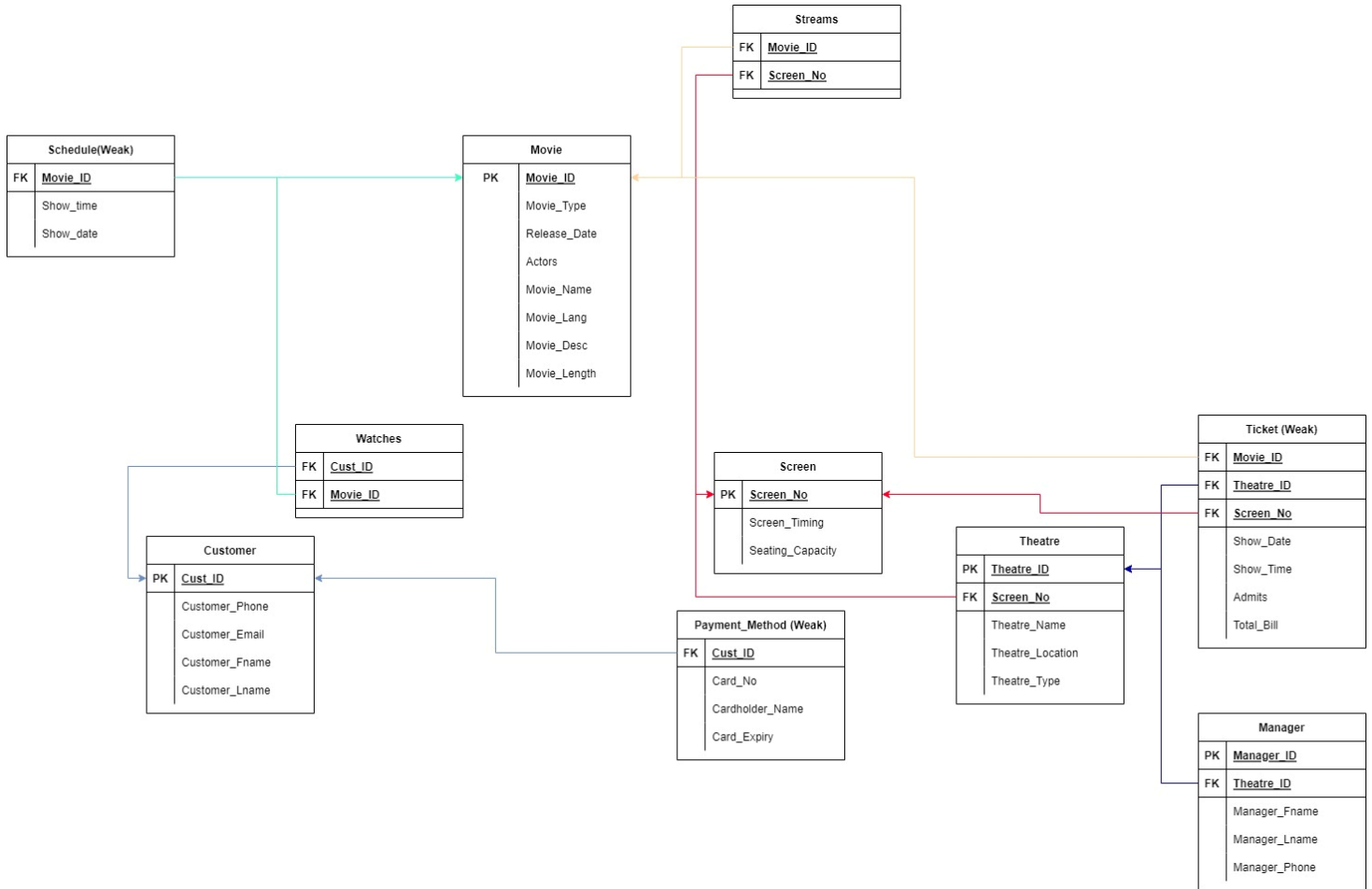
b. Bootstrap (For CSS)

c. EJS view engine

2. ER Diagram



3. Relational Schema



4. 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));
```

5. 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);

2) 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)

3) 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.

6. 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:

The screenshot shows a database management tool interface. At the top, there are tabs for 'customer', 'watches', 'Customer_Ticket_Theatre_Movi...', 'Movie_Booking_Join', 'Manager_Theatre_Join' (selected), 'movie', and 'customer'. Below the tabs is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. The main area displays a SQL query in a text editor:

```
1  /*
2   Name:Tanay Singh
3   SRN:PES2UG20CS364
4   Section:F
5  */
6  • SELECT Manager_ID,Manager_Fname,Manager_Lname,Theatre_Name,Theatre_Location,Theatre_Type
7     FROM manager,theatre
8     where
9     manager.Theatre_ID=theatre.Theatre_ID;
```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows:' input field, an 'Export:' button, and a 'Wrap Cell Content:' checkbox. The grid displays the following data:

	Manager_ID	Manager_Fname	Manager_Lname	Theatre_Name	Theatre_Location	Theatre_Type
▶	1	Paul	Jacob	PVR (Phoneix Marketcity)	Mahadevpura,Bengaluru	Single-Screen
	2	Vijay	Prasad	Meenakshi Cinema Hall	Marathalli Multiplex,Bengaluru	Single Screen
	3	Akhilesh	Kumar	Inox (Gopalan Signature Mall)	Old Madaras Road,Bengaluru	Single Screen
	4	Shabaz	Ahmed	PVR-IMAX (VR Mall)	Mahadevpura,Bengaluru	Single-Screen
	5	Jatinder	Singh	PVR (Soulspace Spirit Mall)	Bellandur,Bengaluru	Single Screen

2) Find out which customer is watching which movie

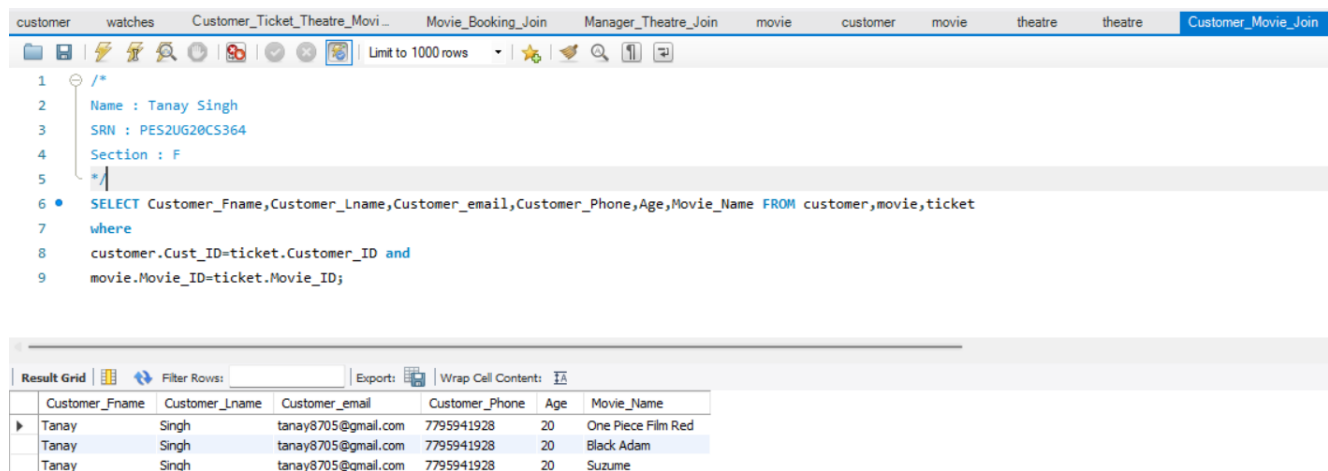
SELECT

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;



The screenshot shows a database query editor with a SQL query and its results. The query is as follows:

```
1  /*
2  Name : Tanay Singh
3  SRN : PES2UG20CS364
4  Section : F
5  */
6  SELECT Customer_Fname, Customer_Lname, Customer_email, Customer_Phone, Age, Movie_Name FROM customer, movie, ticket
7  where
8  customer.Cust_ID=ticket.Customer_ID and
9  movie.Movie_ID=ticket.Movie_ID;
```

The results are displayed in a table with the following columns: Customer_Fname, Customer_Lname, Customer_email, Customer_Phone, Age, and Movie_Name. The results are as follows:

Customer_Fname	Customer_Lname	Customer_email	Customer_Phone	Age	Movie_Name
Tanay	Singh	tanay8705@gmail.com	7795941928	20	One Piece Film Red
Tanay	Singh	tanay8705@gmail.com	7795941928	20	Black Adam
Tanay	Singh	tanay8705@gmail.com	7795941928	20	Suzume

3) 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;

customer watches Customer_Ticket_Theatre_Movi... Movie_Booking_Join Manager_Theatre_Join movie customer movie theatre theatre Customer_Movie_Join

Limit to 1000 rows

```

1  /*
2  Name : Tanay Singh
3  SRN : PES2UG20CS364
4  Section : F
5  */
6  • SELECT Customer_Fname, Customer_Lname, Movie_Name, Movie_Lang, Movie_Length, Movie_Type, Theatre_Name, Theatre_Location, Show_Date, Show_Time, Theatre_Type
7  FROM ticket, theatre, movie, customer
8  where
9  ticket.theatre_id=theatre.Theatre_ID and
10 ticket.movie_id=movie.movie_id and
11 ticket.customer_id=customer.cust_id;

```

Result Grid

	Customer_Fname	Customer_Lname	Movie_Name	Movie_Lang	Movie_Length	Movie_Type	Theatre_Name	Theatre_Location	Show_Date	Show_Time	Theatre_Type
▶	Tanay	Singh	One Piece Film Red	JP	115	Anime	PVR-IMAX (VR Mall)	Mahadevpura, Bengaluru	2022-11-21	17:00:00	Single-Screen
	Tanay	Singh	Black Adam	EN	125	Action	PVR-IMAX (VR Mall)	Mahadevpura, Bengaluru	2022-11-23	17:00:00	Single-Screen
	Tanay	Singh	Suzume	JP	122	Drama	Inox (Gopalan Signature Mall)	Old Madaras Road, Bengaluru	2022-11-30	12:00:00	Single Screen

4) 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;

customer watches Customer_Ticket_Theatre_Movi... Movie_Booking_Join Manager_Theatre_Join movie customer movie theatre theatre Customer_Movie_Join theatre

Limit to 1000 rows

```

2  Name : Tanay Singh
3  SRN : PES2UG20CS364
4  Section : F
5  */
6  • USE movie_db;
7  • SELECT Movie_Name, Movie_Type, Movie_Length, Show_Date, Show_Time, Admits, Seat_Type
8  FROM movie, ticket
9  where
10 movie.movie_id=ticket.movie_id and
11 ticket.customer_id=1;

```

Result Grid

	Movie_Name	Movie_Type	Movie_Length	Show_Date	Show_Time	Admits	Seat_Type
▶	One Piece Film Red	Anime	115	2022-11-21	17:00:00	4	Gold
	Black Adam	Action	125	2022-11-23	17:00:00	2	Premium
	Suzume	Drama	122	2022-11-30	12:00:00	1	Gold

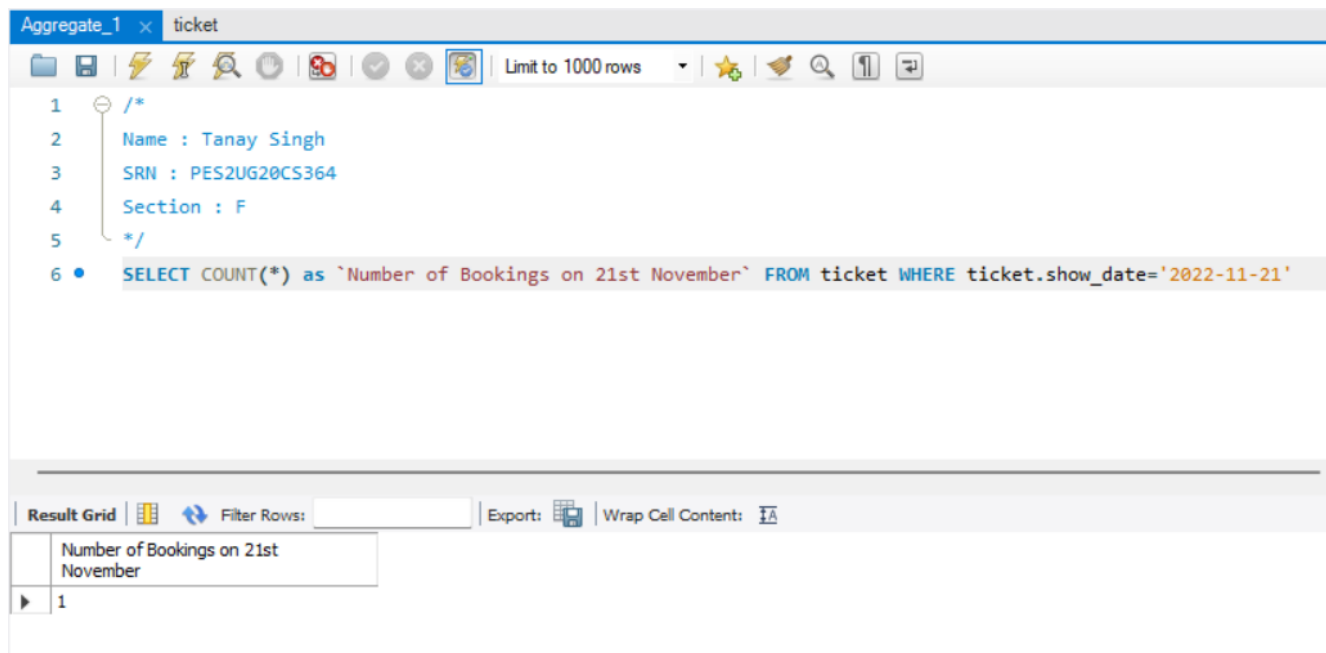
7. Aggregate Functions

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'



The screenshot shows a database query editor with a toolbar at the top. The query is written in a text area and is highlighted in blue. The results are displayed in a table below the query.

```
1 /*
2  Name : Tanay Singh
3  SRN : PES2UG20CS364
4  Section : F
5  */
6 • SELECT COUNT(*) as `Number of Bookings on 21st November` FROM ticket WHERE ticket.show_date='2022-11-21'
```

Number of Bookings on 21st November
1

2) Show the total number of customers registered to the movie booking system:

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

Aggregate_1 ticket Aggregate_2 x

Limit to 1000 rows

```

1  /*
2  Name : Tanay Singh
3  SRN : PES2UG20CS364
4  Section : F
5  */
6  • SELECT COUNT(*) as `Number of Customers Registered` FROM customer;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	Number of Customers Registered
▶	3

3) 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;

Aggregate_1 ticket Aggregate_2 ticket customer movie Aggregate_3 x

Limit to 1000 rows

```

1  /*
2  Name : Tanay Singh
3  SRN : PES2UG20CS364
4  Section : F
5  */
6  • SELECT SUM(Admits) as `Total People watching One Piece Red` FROM ticket where Movie

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	Total People watching One Piece Red
▶	5

4) 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;
```

The screenshot shows a database management tool interface. At the top, there are several tabs: Aggregate_1, Aggregate_2, customer, movie, Aggregate_3, ticket, Total_Value_Function, Executing_Function, ticket, and Aggregate_4 (which is active). Below the tabs is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. The main area displays a SQL query in a text editor:

```
1  /*
2  Name : Tanay Singh
3  SRN : PES2UG20CS364
4  Section : F
5  */
6
7  • SELECT AVG(Total_Bill) as `Average Spending of Customer_ID 1` FROM ticket WHERE Customer_ID=1;
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows:' input field, an 'Export:' button, and a 'Wrap Cell Content:' checkbox. The result grid shows a single row of data:

Average Spending of Customer_ID 1
1093.6667

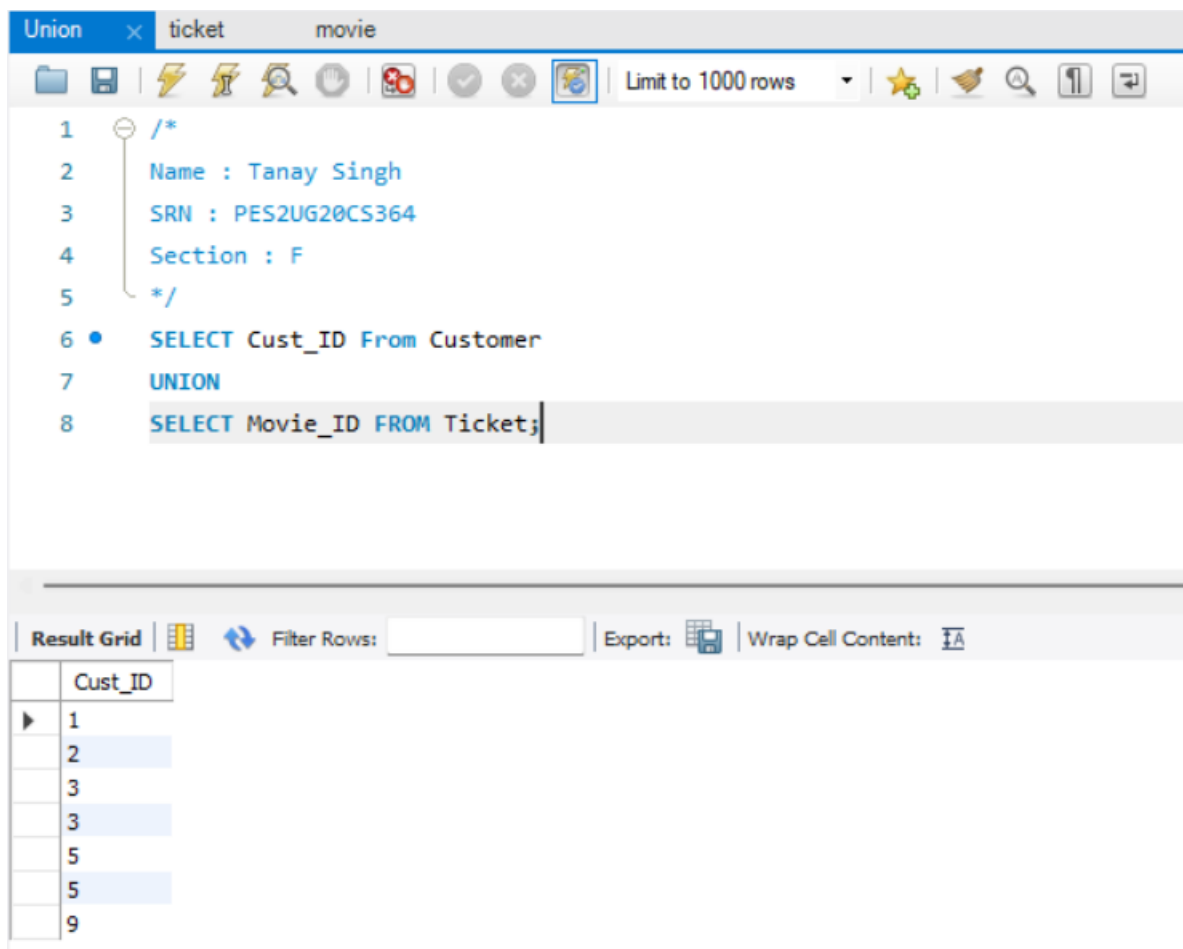
8. Set Operations

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;
```



The screenshot shows a database query editor with a query window titled 'Union' containing the following SQL query:

```
1  /*  
2  Name : Tanay Singh  
3  SRN : PES2UG20CS364  
4  Section : F  
5  */  
6  •  SELECT Cust_ID From Customer  
7  UNION  
8  SELECT Movie_ID FROM Ticket;
```

Below the query window, the 'Result Grid' is displayed, showing the results of the query. The grid has a single column labeled 'Cust_ID' and contains the following values:

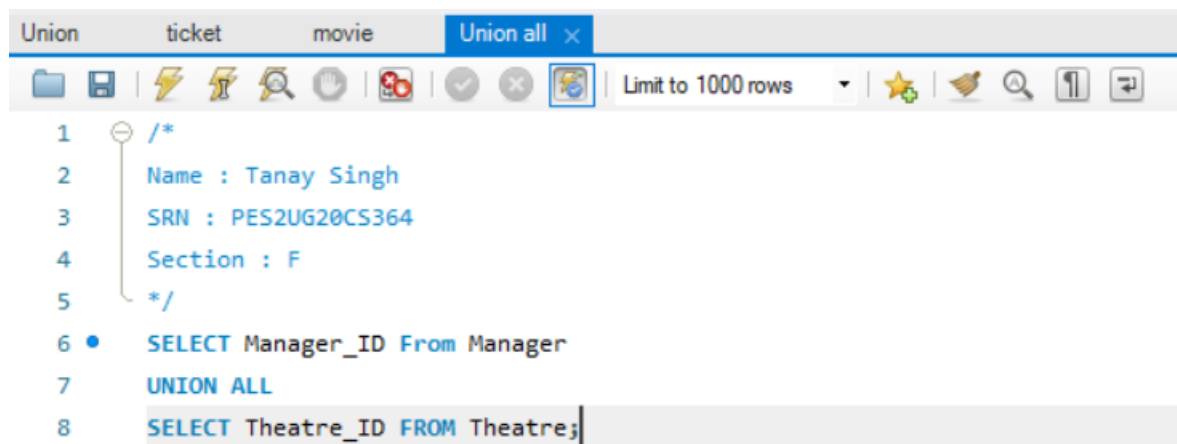
Cust_ID
1
2
3
3
5
5
9

2) Union all between Manager and Theatre Table:

SELECT Manager_ID From Manager

UNION ALL

SELECT Theatre_ID FROM Theatre;



The screenshot shows a result grid with the following data:

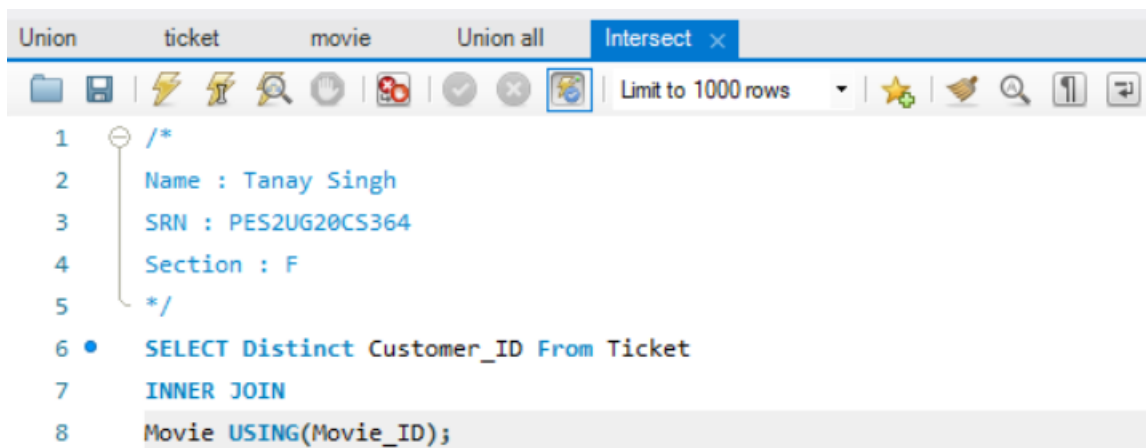
Manager_ID
1
4
3
5
2
1
2
3
4
5

3) Intersection between Customers table and Movie table

SELECT Distinct Customer_ID From Ticket

INNER JOIN

Movie USING(Movie_ID);



The screenshot shows a database result grid with a toolbar at the top. The 'Result Grid' button is highlighted. Below the toolbar, the result grid is displayed with the following data:

Customer_ID
1
3

4) Minus

SELECT Cust_ID From Customer

LEFT JOIN Ticket ON Customer_ID

WHERE Ticket.Customer_ID IS NULL;

Union ticket movie Union all Intersect **Minus** × ticket

Limit to 1000 rows

```
1  /*
2    Name : Tanay Singh
3    SRN : PES2UG20CS364
4    Section : F
5  */
6  •  SELECT Cust_ID From Customer
7     LEFT JOIN Ticket ON Customer_ID
8     WHERE Ticket.Customer_ID IS NULL;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

Cust_ID

9. 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:

SQL Editor Interface:

Running_Function ticket

Limit to 1000 rows

1 • `SELECT * FROM movie_db.ticket;`

Result Grid

	Movie_ID	Theatre_ID	Customer_ID	Show_Date	Show_Time	Admits	Total_Bill	Seat_Type
▶	5	2	1	2022-11-21	17:00:00	4	0	Gold
	3	2	1	2022-11-23	17:00:00	2	0	Premium
	9	3	1	2022-11-30	12:00:00	1	0	Gold
	5	3	3	2022-11-21	17:00:00	1	0	Silver

After Function

SQL Editor Interface:

Running_Function ticket

Limit to 1000 rows

1 `UPDATE ticket`
2 `SET Total_Bill = movie_db.Calculate_Total(Admits,Seat_Type);`
3
4 • `SELECT * FROM ticket;`

Result Grid

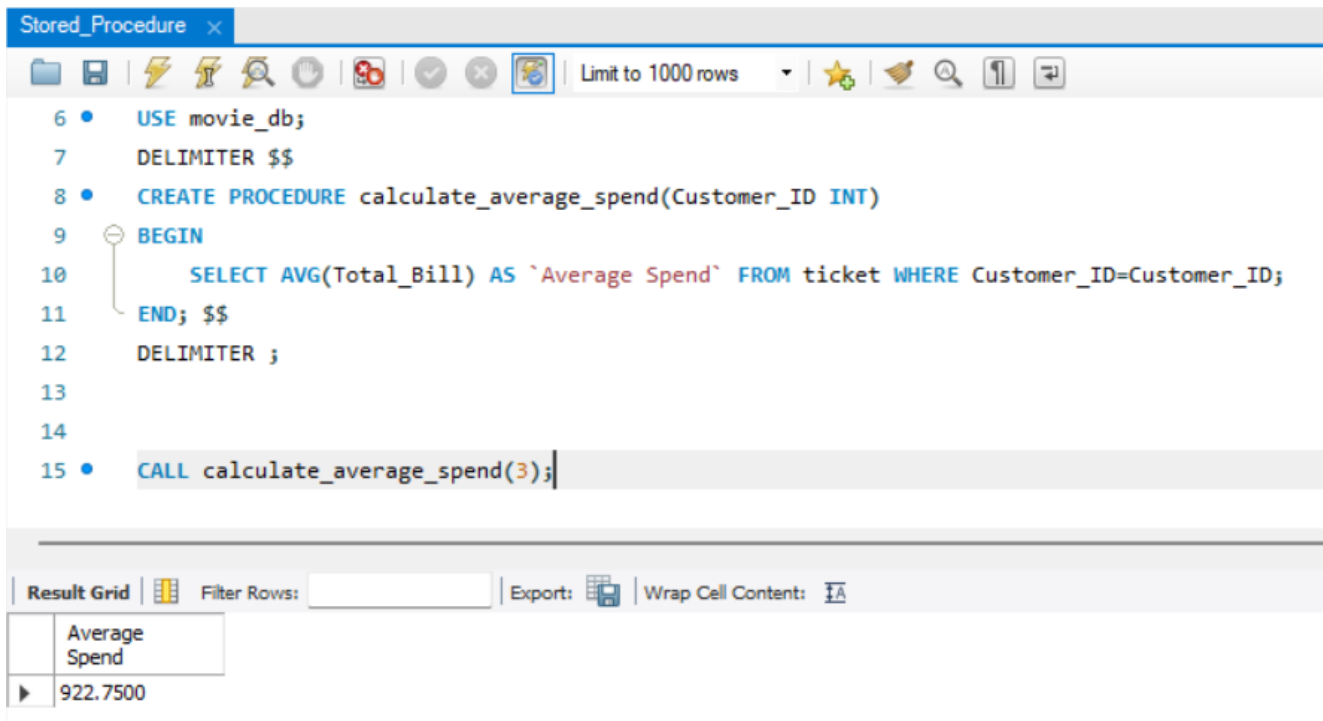
	Movie_ID	Theatre_ID	Customer_ID	Show_Date	Show_Time	Admits	Total_Bill	Seat_Type
▶	5	2	1	2022-11-21	17:00:00	4	1455	Gold
	3	2	1	2022-11-23	17:00:00	2	1391	Premium
	9	3	1	2022-11-30	12:00:00	1	435	Gold
	5	3	3	2022-11-21	17:00:00	1	410	Silver

Running :

2) 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);
```



The screenshot shows a SQL IDE window titled "Stored_Procedure". The SQL editor contains the following code:

```
6 • USE movie_db;
7 • DELIMITER $$
8 • CREATE PROCEDURE calculate_average_spend(Customer_ID INT)
9 • BEGIN
10 •     SELECT AVG(Total_Bill) AS `Average Spend` FROM ticket WHERE Customer_ID=Customer_ID;
11 • END; $$
12 • DELIMITER ;
13
14
15 • CALL calculate_average_spend(3);
```

The IDE interface includes a toolbar with icons for file operations, execution, and search. A status bar at the top right indicates "Limit to 1000 rows". Below the editor, the "Result Grid" is displayed, showing the output of the query:

Average Spend
922.7500

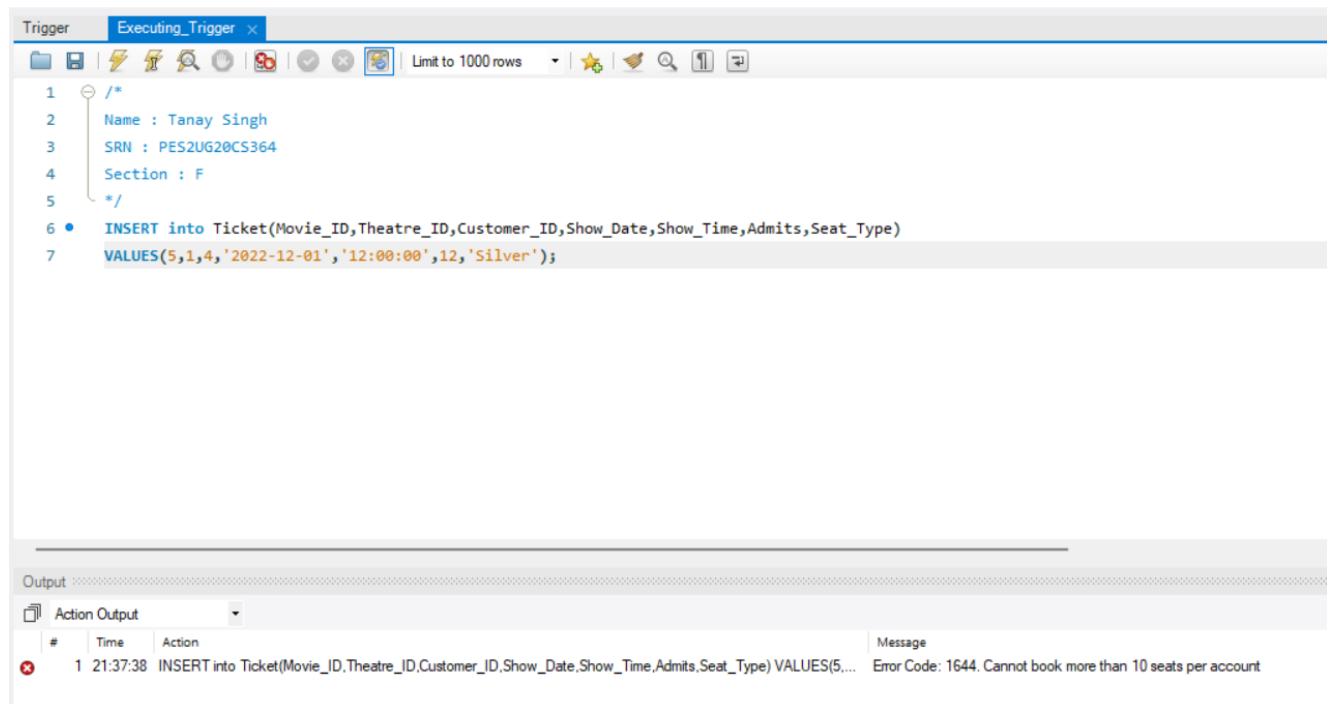
10. Triggers and Cursors

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:



11. 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/MV5BNTM4NjIxNmEtYWE5NS00NDczLTkyNWQtYThhNmQyZGQzMjM0XkEyXkFqcGdeQ
XVyODk4OTc3MTY@._V1_FMjpg_UX1000_.jpg', 'https://m.media-
amazon.com/images/M/MV5BNzEzOTZjYjYtMTI3MS00NWE1LWIyZTk0YTRlMTBlYjlkMjZiXkEyXkFqcGdeQ
XVyMTUzMTg2ODkz._V1_.jpg', 'https://www.theilluminerdi.com/wp-
content/uploads/2022/09/black-adam-heroes.jpg', 'https://m.media-
amazon.com/images/M/MV5BZjgwNDA3MmUtMTQ3Yy00ZDFmLTgwMTktNTBlMTQ3ZTI5MjYxXkEyXkFqcGdeQ
XVyMTM1MTE1NDMx._V1_FMjpg_UX1000_.jpg', 'https://m.media-
amazon.com/images/M/MV5BNDBhM2IxMzItOTdmZi00NmMzLWFlNTgtN2E2MDRjODU4NzEzXkEyXkFqcGdeQ
XVyODMyNTM0MjM@._V1_.jpg', 'https://m.media-
amazon.com/images/M/MV5BMDNhMWUxMjctZWZhNy00MjUzLWE3NjYtNTE4ZjVhYjRkZThmXkEyXkFqcGdeQ
XVyMTkxNjUyNQ@@._V1_FMjpg_UX1000_.jpg', 'https://m.media-
amazon.com/images/M/MV5BZWY2MGIxYjMtNGRiMi00ODczLWFlZjItYzE5YTdmNzd1NDEwXkEyXkFqcGdeQ
XVyMTMzODk3NDU0._V1_.jpg', 'https://m.media-
amazon.com/images/M/MV5BYTE1MmZiMwYtYTFmZi00YjA3LWI2ODgtMWF1NWYxZjdmNGE3XkEyXkFqcGdeQ
XVyNjk1Njg5NTA@._V1_FMjpg_UX1000_.jpg', 'https://www.cwfilms.jp/en/news/images/12809e7
f10e8ec3c6691c5c0c1be7cc50fe00ba6.jpg', 'https://m.media-
amazon.com/images/M/MV5BYzFmMjAwMDYtNzM0Zi00NjY2LWFlZjYtMGQ1OTRiZjk5YjJkXkEyXkFqcGdeQ
XVyMTMwODY5NDc2._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_Da
te,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.toString())
        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='${req.body.sho
w_time}' WHERE Customer_ID=${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
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]%></h4>
                        <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]%></h5>
        <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>
</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>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col">
          <div class="card">
            <div class="card-body">
              <div class="text-center">
                <h2>Movie Booking System</h2>
                <h3>View currently streaming movies</h3>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        </div>
    </nav>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>Update Booking</title>
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivrivr.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.jsdelivrivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
      <script
src="https://cdn.jsdelivrivr.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>
          </div>
        </div>
      </div>
    </body>
  </html>
```

```

        <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>
</body>

```

Login Page Screenshot

Movie Booking System

Please Enter your login credentials here

Username :

Password :


Submit

If you are a new user, please click this link

Movie Page Screenshot:


Welcome tanay8705 Theatres Booked Tickets

Currently Available Movies




Black Panther: Wakanda Forever
EN
Action

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.




Terrifier 2
EN
Horror

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




Black Adam
EN
Action

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



Orphan: First Kill
EN
Horror

After escaping from an Estonian psychiatric facility, Leena Klammer travels to America by impersonating Esther, the missing daughter of a wealthy



One Piece Film Red
JP
Anime

Uta — the most beloved singer in the world. Her voice, which she sings with while concealing her true identity, has been described as the sound of the

Theatres Page Screenshot:

Movie Booking System View currently streaming movies		
Theatre Name	Theatre Location	Theatre Type
PVR (Phoneix Marketcity)	Mahadevpura,Bengaluru	Single-Screen
PVR-IMAX (VR Mall)	Mahadevpura,Bengaluru	Single-Screen
Inox (Gopalan Signature Mall)	Old Madaras Road,Bengaluru	Single Screen
PVR (Soulspace Spirit Mall)	Bellandur,Bengaluru	Single Screen
Meenakshi Cinema Hall	Marathalli Multiplex,Bengaluru	Single Screen

Booked Tickets

Movie Booking System		
<div><div>Booking 1</div><div>One Piece Film Red Anime Runtime : 115 minutes Show Date : Mon Nov 21 2022 Show Time : 17:00:00 Total seats booked : 4 Seat Type : Gold Total Spent : ₹ 1455 <div><div>Modify Booking</div><div>Delete Booking</div></div></div></div>	<div><div>Booking 2</div><div>Suzume Drama Runtime : 122 minutes Show Date : Wed Nov 30 2022 Show Time : 12:00:00 Total seats booked : 1 Seat Type : Gold Total Spent : ₹ 435 <div><div>Modify Booking</div><div>Delete Booking</div></div></div></div>	<div><div>Booking 3</div><div>All Quiet on the Western Front Drama Runtime : 143 minutes Show Date : Mon Nov 21 2022 Show Time : 17:00:00 Total seats booked : 4 Seat Type : Premium Total Spent : ₹ 2611 <div><div>Modify Booking</div><div>Delete Booking</div></div></div></div>

Modify Booking

Movie Booking System View currently streaming movies	
<div><div>Select preferred seat type :</div><div>Gold(Back-Row Seats) - (₹: <div>▼</div>)</div></div> <div><div>Enter the number of seats :</div><div>4</div></div> <div><div>Select the show time :</div><div>5 PM <div>▼</div></div></div> <div><div>Update Booking</div></div>	

Delete Booking

Current booking

One Piece Film Red

Anime

Runtime : 115 minutes

Show Date : Mon Nov 21 2022

00:00:00 GMT+0530 (India
Standard Time)

Show Time : 17:00:00

Total seats booked : 4

Seat Type : Gold

Total Spent : ₹ 1455

Go Back

Are you sure you want to
delete this booking?

Delete

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>