

# MINI PROJECT USER REQUIREMENT SPECIFICATION

## MOVIE TICKET BOOKING SYSTEM

UE21CS351A : DATABASE MANAGEMENT SYSTEM

TEAM MEMBERS:

PRAJWAL M (PES1UG21CS421)

NIDA SAMREEN (PES1UG21CS379)

GROUP ID : G37

CLASS : SEM 5 'G' SEC

### 1.INTRODUCTION

#### PURPOSE OF THE PROJECT:

It is a web based application that allows users to book the movie tickets. It provides a user friendly platform for customers to select the movies, view show timings, choose seats and easy online transaction.

#### SCOPE OF THE PROJECT:

The web based application should be user friendly to be run on any system without any inconveniences and should be accessible by a simple internet connection to book the tickets.

### 2.PROJECT DESCRIPTION

#### PROJECT OVERVIEW:

The Movie Ticket Booking System is a complete database management system (DBMS) project made to make it easier for customers to purchase movie tickets and for administrators to handle movie-related data. This system seeks to offer administrators the capabilities to effectively manage movie listings, theatres, showtimes, and reservations while giving cinema aficionados a user-friendly interface to browse, choose, and book movie tickets online.

#### MAJOR PROJECT FUNCTIONALITIES:

The main features of a movie ticket booking system DBMS project may be separated into two groups: userfacing features and administrative features. The key features for each category are listed here:

##### User-Facing Functionalities:

###### 1. User Registration:

- Allow users to create accounts with personal information.

###### 2. Movie Listings:

- Display a list of available movies with details such as title, rating, budget and cast.
- Allow users to search and filter movies based on various criteria (e.g., release date).

###### 3. Theatre Listings:

- Present a list of theatres with information about location and seating capacity.

###### 4. Showtime Details:

- Display showtimes for each movie at different theatres, including dates, times, and ticket prices.

#### 5. Seat Selection:

- Enable users to select and reserve specific seats for a chosen showtime.
- Display a seating chart with seat availability and pricing details.

#### 6. Booking and Payment:

- Confirm and finalize the booking by selecting payment options (e.g., credit/debit card, online payment).
- Calculate and display the total price based on the selected seats and showtime.

#### 7. Profile Management:

- Allow users to update their profile information (e.g., email, phone number)

### Administrative Functionalities:

#### 1. Admin Login:

- Implement a separate login system for administrators with appropriate access controls.

#### 2. Movie Management:

- Add, edit, and delete movie listings with details like title, cast, rating, and budget.
- Upload movie posters or images.
- Set release dates and update movie information.

#### 3. Theatre Management:

- Add, edit, and delete theatres with information about location and seating capacity.

#### 4. Showtime Management:

- Create, edit, and delete showtimes for movies at different theatres.
- Specify date and time, and set ticket prices.
- Manage showtime availability and seat limits.

#### 5. Booking Management:

- View and manage user bookings, including details of booked seats, showtimes, and user information.
- Cancel or modify bookings when necessary.

#### 6. User Management:

- Admins can manage user accounts if needed
- Monitor user activity for security purposes.

#### 7. System Configuration:

- Configure system settings, such as currency and payment gateway integration.

### 3.SYSTEM FEEATURES AND FUNCTION REQUIREMENTS

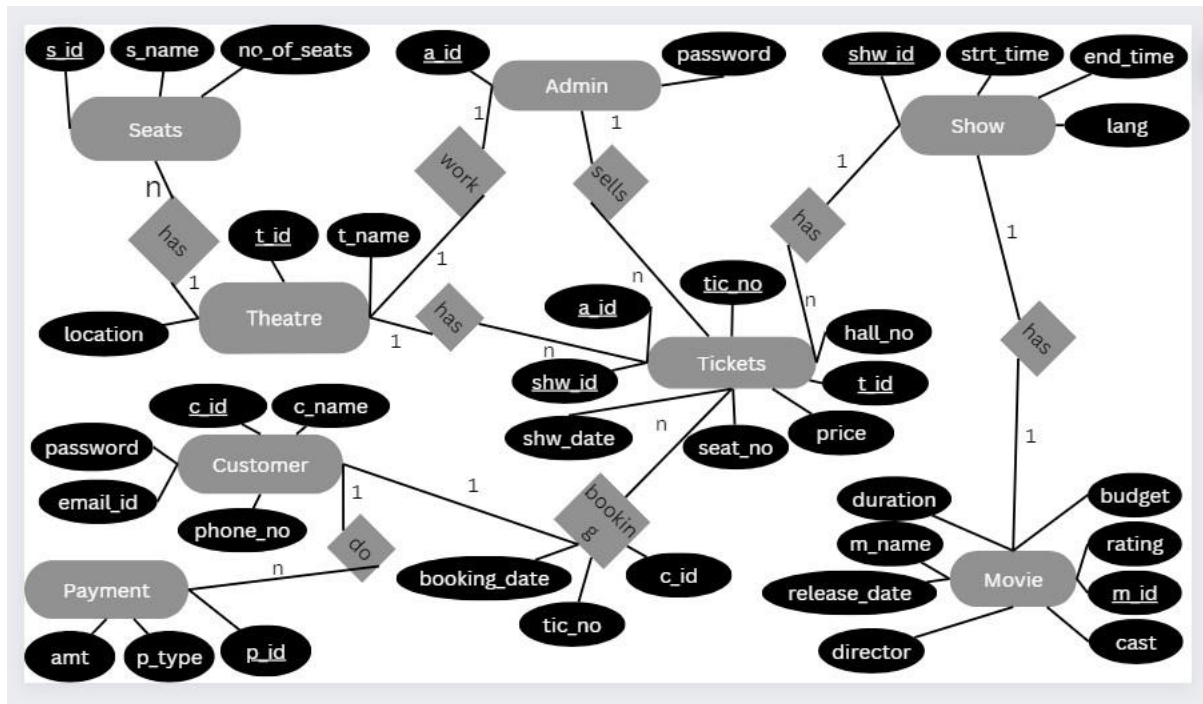
#### User-Facing Features:

1. User Registration and Authentication
2. Movie Listings
3. Theatre Listings
4. Showtime Details
5. Seat Selection
6. Booking and Payment
7. Profile Management

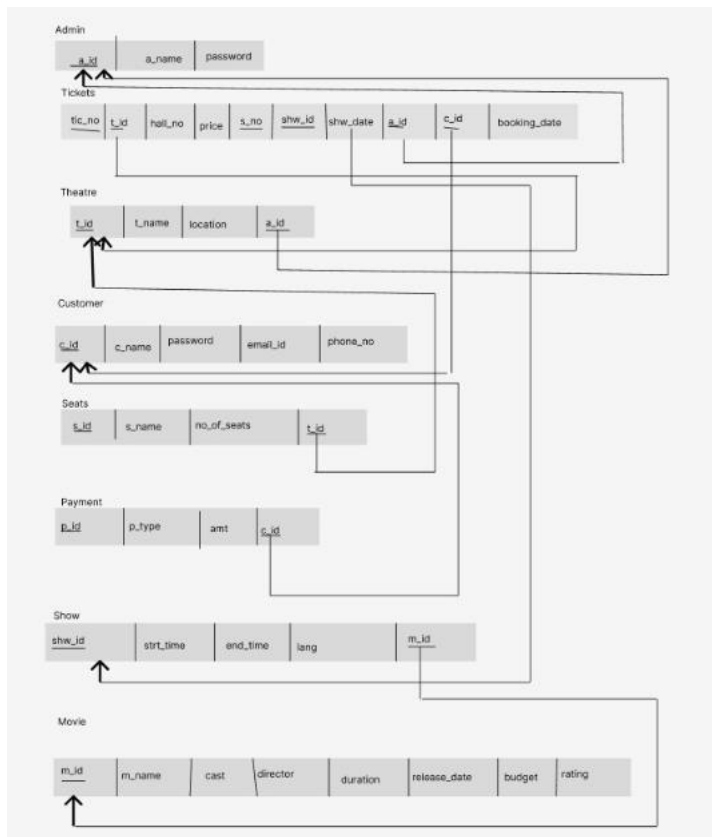
Administrative Features:

1. Admin Login
2. Movie Management
3. Theatre Management
4. Showtime Management
5. Booking Management 6. User
- Management
7. System Configuration

ER DIAGRAM:



RELATIONAL SCHEMA:



Trigger Definition:

BEGIN

DECLARE booked\_seats VARCHAR(255);

DECLARE intersect\_count INT;

-- Fetch booked seats for the given show date, theatre, and show time

SELECT selected\_seats INTO booked\_seats

FROM bookings

WHERE showdate = NEW.showdate

AND ttre = NEW.ttre

AND st = NEW.st

AND status = 'Successful'

```

LIMIT 1;

-- Check for intersection between the selected seats and booked seats
SET intersect_count = (SELECT COUNT(*)
                        FROM (
                            SELECT seat FROM (
                                SELECT
                                TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(booked_seats, ',',
n digit+1), ',', -1)) AS seat
                                FROM (
                                    SELECT @rownum:=@rownum+1 AS digit
                                    FROM information_schema.COLUMNS
                                    WHERE TABLE_NAME = 'bookings'
                                    AND COLUMN_NAME = 'selected_seats'
                                    AND @rownum < 100 -- Assuming no seat number
exceeds 100
                                ) n
                                JOIN (
                                    SELECT @rownum:=0
                                ) r
                                WHERE n digit < LENGTH(booked_seats)
                                AND
                                SUBSTRING_INDEX(SUBSTRING_INDEX(booked_seats, ',', n digit+1), ',', -
1) <> "
                                ) seats
                                WHERE seat IN (NEW.selected_seats)
                                ) AS t);

```

```

-- If there is an intersection, prevent the INSERT
IF intersect_count > 0 THEN

    SIGNAL SQLSTATE '45000'

    SET MESSAGE_TEXT = 'Some selected seats are already booked. Please
choose different seats.';

END IF;

END

```

This trigger prevents the insertion of new bookings if there are overlapping seats with already booked seats for a specific show. It retrieves booked seats based on the show date, theatre, and show time, and checks for seat intersections with the newly selected seats. If an intersection is found, it raises an exception, signaling that some selected seats are already booked, prompting users to choose different seats. The trigger ensures data integrity by avoiding conflicting seat reservations.

Procedure definition:

```

function getBookedSeats($showdate, $ttre, $st)
{
    global $conn;

    // Fetch booked seats for the given show date, theatre, and show time
    $selectedSeatsQuery = $conn->prepare("SELECT selected_seats FROM
bookings

    WHERE showdate=:showdate AND ttre=:ttre AND st=:st AND
status='Successful'");

    $selectedSeatsQuery->execute([

        ':showdate' => $showdate,

        ':ttre' => $ttre,

        ':st' => $st,

```

```

]);

$selectedSeatsResult = $selectedSeatsQuery->fetchAll(PDO::FETCH_OBJ);

// Extract booked seats from the result

$bookedSeats = [];

foreach ($selectedSeatsResult as $row) {

    $bookedSeats = array_merge($bookedSeats, explode(',', $row->selected_seats));

}

return $bookedSeats;
}

```

The `getBookedSeats` function fetches and returns an array of booked seats for a given show date, theatre, and show time. It uses a prepared SQL query to retrieve successful bookings from the bookings table. The selected seats are then extracted and merged into an array, which is returned by the function. The array contains seat numbers that have been successfully booked for the specified parameters.

Queries:

1. Aggregate function:

// Movie count

```
$moviesQuery = $conn->query("SELECT COUNT(*) AS count_movies FROM movies");
```

```
$moviesQuery->execute();
```

```
$allMovies = $moviesQuery->fetch(PDO::FETCH_OBJ);
```

// Admin count

```
$adminsQuery = $conn->query("SELECT COUNT(*) AS count_admins FROM admins");
```

```

$adminsQuery->execute();

$allAdmins = $adminsQuery->fetch(PDO::FETCH_OBJ);

// Theatre count

$theatresQuery = $conn->query("SELECT COUNT(*) AS count_theatres
FROM theatres");

$theatresQuery->execute();

$allTheatres = $theatresQuery->fetch(PDO::FETCH_OBJ);

// Booking count

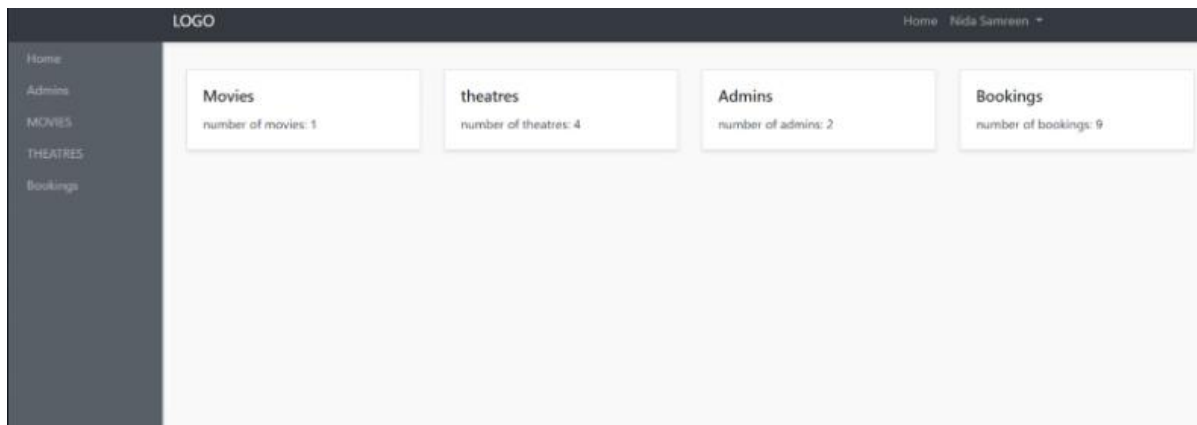
$bookingsQuery = $conn->query("SELECT COUNT(*) AS count_bookings
FROM bookings");

$bookingsQuery->execute();

$allBookings = $bookingsQuery->fetch(PDO::FETCH_OBJ);

```

Output screenshot:



## 2. Join operation:

```

$showTimesQuery = $conn->query("SELECT show_times.*, theatres.name AS
theatre_name
FROM show_times
INNER JOIN theatres ON show_times.theatre_id = theatres.id

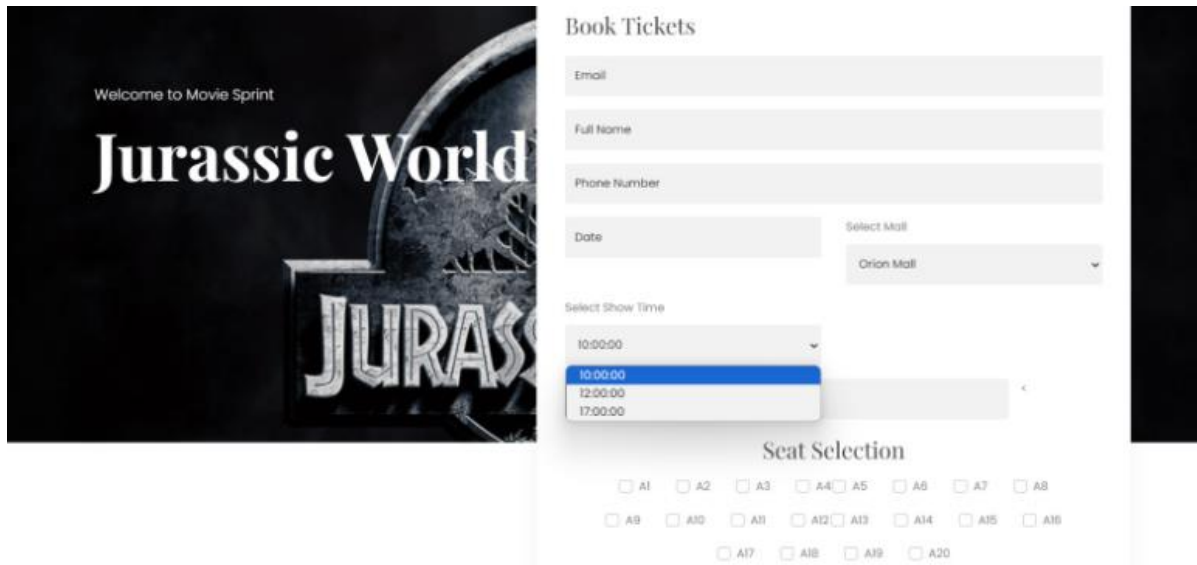
```



```
WHERE show_times.status = 1 AND
show_times.mall_id='$selectedMallId');
```

```
$showTimes = $showTimesQuery->fetchAll(PDO::FETCH_OBJ);
```

Output screenshot:



3 . Stored procedure:

```
$stmt = $conn->prepare("CALL InsertUser(:username, :email, :mypassword)");
```

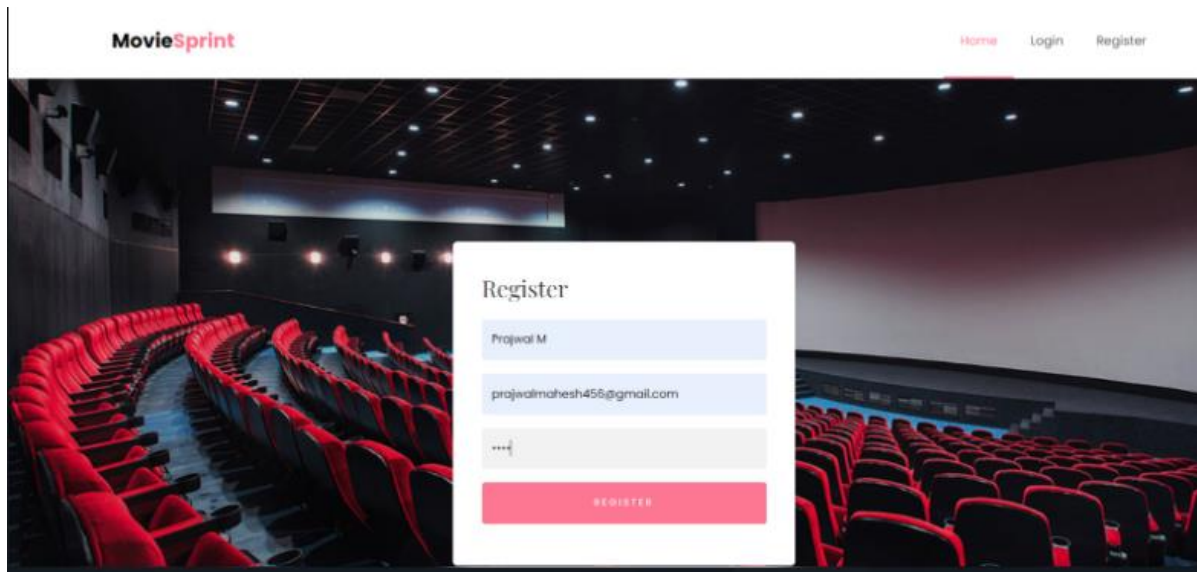
```
$stmt->bindParam(':username', $username, PDO::PARAM_STR);
```

```
$stmt->bindParam(':email', $email, PDO::PARAM_STR);
```

```
$stmt->bindParam(':mypassword', $password, PDO::PARAM_STR);
```

```
$stmt->execute();
```

Output screenshot:



Server: 127.0.0.1 Database: movie-booking Table: users

Showing rows 0 - 9 (10 total, Query took 0.0004 seconds.)

SELECT \* FROM 'users'

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

				id	username	email	mypassword	created_at
<input type="checkbox"/>	Edit	Copy	Delete	1	Nida Samreen	nidasamreen1@gmail.com	\$2y\$10\$QfFb9Sou2zkoPDbe9FQyeqS0/v66Ma5EuylI8YXnu...	2023-11-07 01:23:32
<input type="checkbox"/>	Edit	Copy	Delete	2	Saba	sabac778@gmail.com	\$2y\$10\$D0gy8cqJB8Q4mWcsixtroPqc7HlLpkW15bFoyWPjyc...	2023-11-13 11:20:40
<input type="checkbox"/>	Edit	Copy	Delete	3	Prajwal M	prajwalmahesh123@gmail.com	\$2y\$10\$d3HzgDxi9ePbMkWCiFrESejBi3YIYgv2IDEBmGRhJg...	2023-11-21 02:59:21
<input type="checkbox"/>	Edit	Copy	Delete	4	Sunaina	Sunaina234@gmail.com	\$2y\$10\$N3jVZTorIUQ8NUtSKyP9OucFEU A35Uj QnulyIG8u...	2023-11-21 19:41:30
<input type="checkbox"/>	Edit	Copy	Delete	5	neha	neha@gmail.com	\$2y\$10\$hgocJFVRVolqcTmrZyZSeuQENUL9QHJlQJ0r0okkq...	2023-11-23 14:34:29
<input type="checkbox"/>	Edit	Copy	Delete	6	hk	hk@gmail.com	\$2y\$10\$G0jDFVRB.IRW2rwebz4V.hARrImL1L1wznX80x2IDJ...	2023-11-23 14:50:07
<input type="checkbox"/>	Edit	Copy	Delete	7	sha	sha@gmail.com	\$2y\$10\$zmYFn49eqzbHJbW.IVjW9OkKOfk8 9jL 3V63Fbc31Uc...	2023-11-23 15:18:34
<input type="checkbox"/>	Edit	Copy	Delete	8	abcd	abcd@gmail.com	\$2y\$10\$nsR5iPKU21 dtawwKVKU TM0oozFO2enETTsnRJDUJZ...	2023-11-27 10:47:07
<input type="checkbox"/>	Edit	Copy	Delete	9	sha	shk@gmail.com	\$2y\$10\$IOZb/TJSu2icvuQxOPwPu.7brz1QRdoniPVk/HYIPvb...	2023-12-01 12:50:39
<input type="checkbox"/>	Edit	Copy	Delete	10	Prajwal M	prajwalmahesh456@gmail.com	\$2y\$10\$S0nPk4X3VxnlLBj1 nUaKOtd3.HmG1d1UX1nJf-SGw...	2023-12-01 20:52:37

4.

```
$selectedSeatsQuery = $conn->prepare("SELECT * FROM bookings
```

```
WHERE showdate=:showdate AND ttre=:ttre AND st=:st AND
payment_status='Successful');
```

```
$selectedSeatsQuery->execute([
```

```
':showdate' => $showdate,
```

```
':ttre' => $ttre,
```

```

        ':st' => $st,
    ];

    $selectedSeatsResult = $selectedSeatsQuery->fetchAll(PDO::FETCH_OBJ);

    // Extract booked seats from the result
    $bookedSeats = [];

    foreach ($selectedSeatsResult as $row) {
        $bookedSeats = array_merge($bookedSeats, explode(',', $row->selected_seats));
    }

    // Assuming A1 to A20 are the available seats
    $allSeats = array_map(function ($num) {
        return "A$num";
    }, range(1, 20));

    // Calculate available seats by finding the difference between all seats and
    booked seats

    $availableSeats = array_diff($allSeats, $bookedSeats);

```

#	T	id	showtime	the	st	noo	email	phone_number	full_name	seat_status	selected_seats	movie_name	theatre_name	status	payment	user_id	created_at	payment_status		
<input type="checkbox"/>	81	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	91	2023-11-26	1	1	ndasamreen1@gmail.com	8050781081	Nida Samreen	1	A3	Jurassic World	Successful	50	1	2023-11-24 15:53:12	Successful	
<input type="checkbox"/>	82	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	93	2023-11-30	1	1	2	ndasamreen1@gmail.com	8050781081	Nida Samreen	1	A4,A5	Jurassic World	Successful	100	1	2023-11-24 22:56:44	Successful
<input type="checkbox"/>	83	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	83	2023-11-26	1	1	1	ndasamreen1@gmail.com	8050781081	Nida Samreen	1	A18	Jurassic World	Successful	50	1	2023-11-27 10:55:12	Successful
<input type="checkbox"/>	84	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	94	2023-11-26	2	1	2	abcs@gmail.com	1734567890	abcs	1	A11,A12	Jurassic World	Successful	100	11	2023-11-27 10:48:55	Successful
<input type="checkbox"/>	85	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	95	2023-11-26	2	1	2	abcs@gmail.com	543456	ghfg	1	A17,A18	Jurassic World	Successful	100	8	2023-11-27 10:52:03	Successful
<input type="checkbox"/>	86	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	96	2023-11-26	2	2	2	abcs@gmail.com	543456	ghfg	1	A17,A18	Jurassic World	Successful	100	8	2023-11-27 10:52:03	Successful
<input type="checkbox"/>	87	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	97	2023-12-03	1	1	1	ndasamreen1@gmail.com	8050781081	Nida Samreen	1	A1	Jurassic World	Successful	50	1	2023-12-01 10:38:06	Successful
<input type="checkbox"/>	88	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	98	2023-12-06	1	1	2	ndasamreen1@gmail.com	8050781081	sh	1	A2,A3	Jurassic World	Successful	100	1	2023-12-01 12:21:38	Successful
<input type="checkbox"/>	89	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	99	2023-12-06	1	1	2	sha@gmail.com	543456	shk	1	A6,A6	Jurassic World	Successful	100	8	2023-12-01 12:52:34	Successful
<input type="checkbox"/>	100	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	100	2023-12-04	1	1	1	ndasamreen1@gmail.com	8050781081	Nida Samreen	1	A1	Jurassic World	Successful	50	1	2023-12-01 21:17:49	Successful

Welcome to Movie Sprint

# Jurassic World

prajwalmaresh123@gmail.com

Prjwal

9606021551

12/4/2023

Select Mall

Orion Mall

Select Show Time

10:00:00

1

Seat Selection

☒ A1

☐ A2

☐ A3

☐ A4

☐ A5

☐ A6

☐ A7

☐ A8

☐ A9

☐ A10

☐ A11

☐ A12

☐ A13

☐ A14

☐ A15

☐ A16

☐ A17

☐ A18

☐ A19

☐ A20

Screen

BOOK AND PAY NOW

localhost says

Some selected seats are already booked. Please choose different seats.

OK