Database Systems Design Phase 2 Report

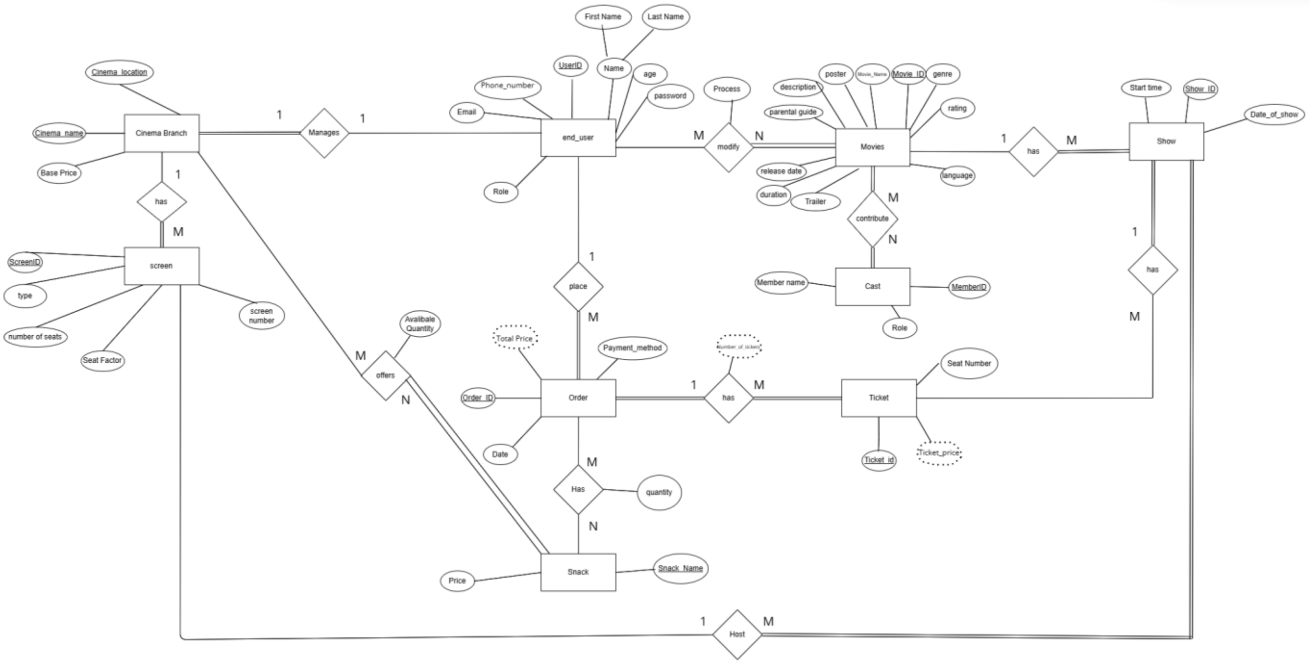
CINEMA BOOKING SYSTEM

Team Number:2
Adham Walid Said 23P0024
Eslam Mohamed Fawzy 23p0052
Carol Kamal Magdy 23P0328
Hassan Ismail 23p0152
Loay Mahmoud 23p0419
Toka Elsayed 23P0044
TA: Eng Esraa Karam

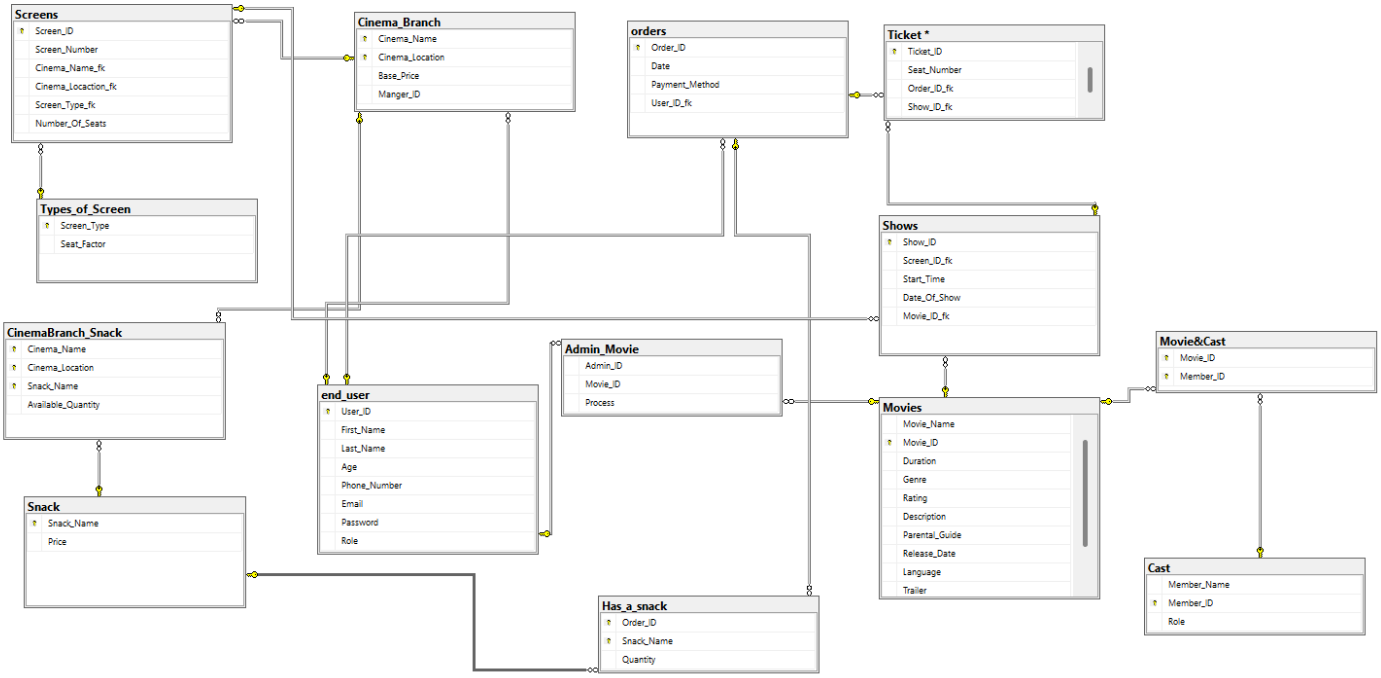
Introduction:

This report documents a **Cinema Booking System** database schema and GUI implementation, designed to manage **movie screenings, ticket sales, and admin operations**. The system tracks **customer bookings (orders, tickets, snacks)**, **movie details (casts, showtimes)**, and **admin activities (movie updates, user management)** across multiple cinema branches. The relational database ensures efficient data handling while the GUI provides intuitive access for users and administrators

1-Entity-Relationship Diagram(CORRECTED)

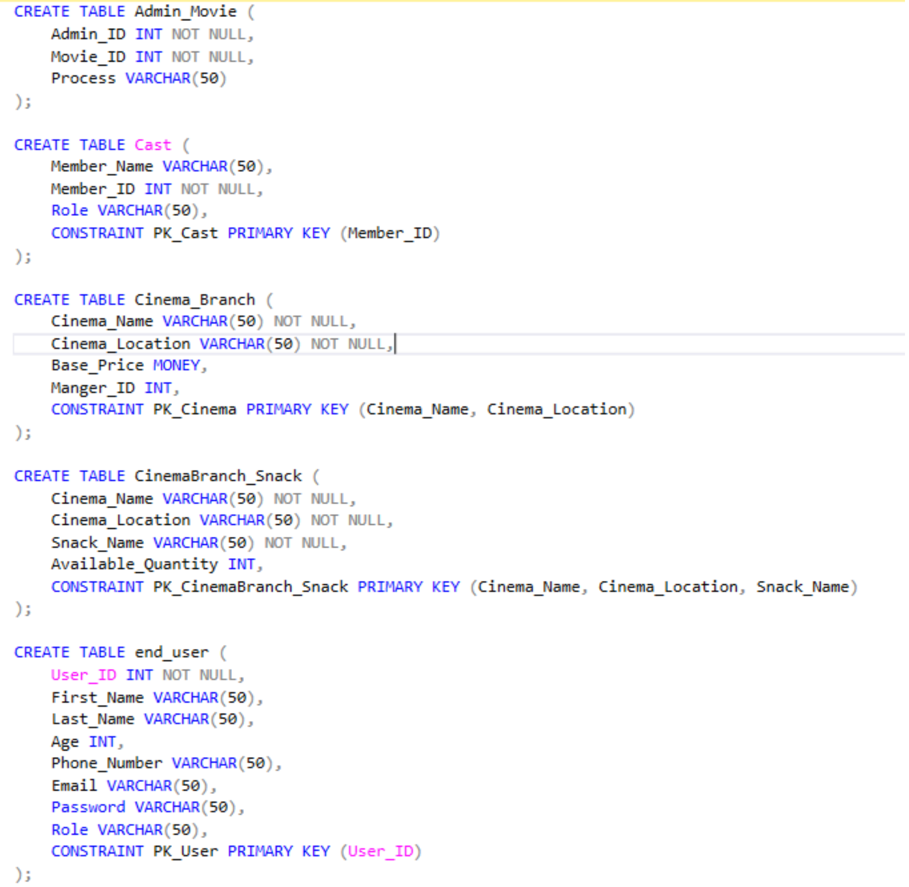


2-Schema(CORRECTED)

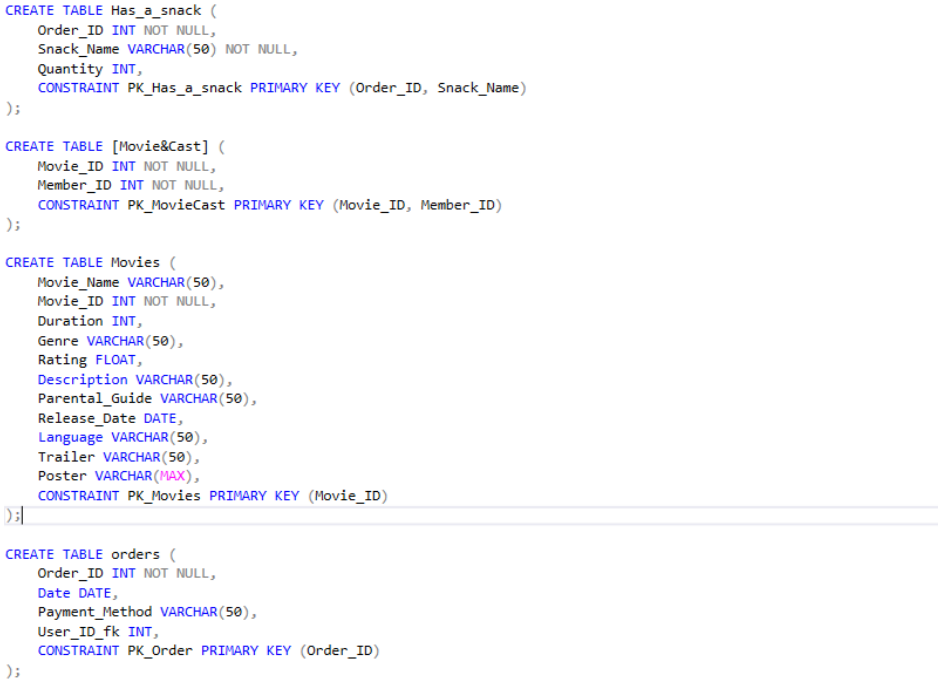


3-Script

1-Table Creation:

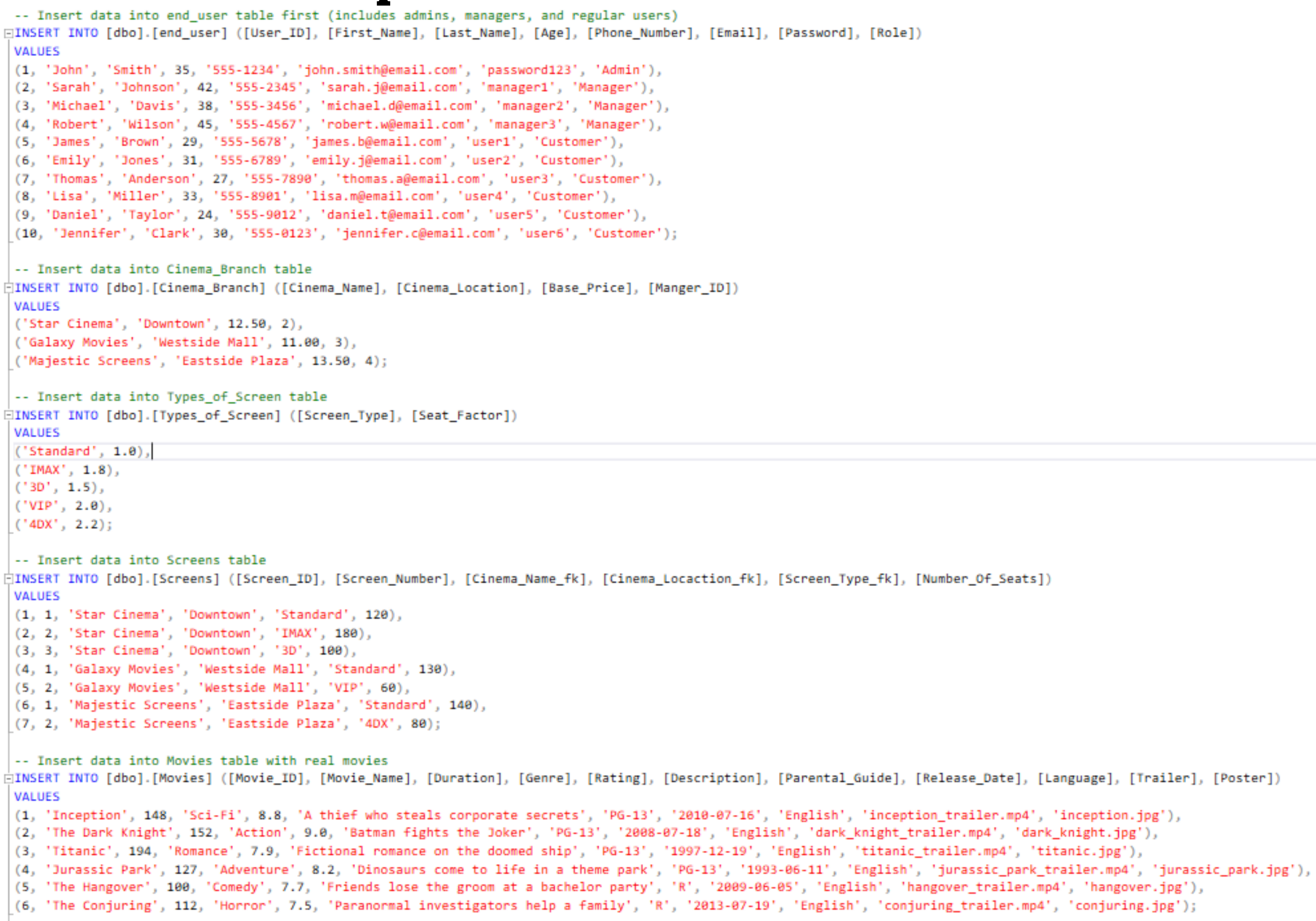


```
CREATE TABLE Admin_Movie (
    Admin_ID INT NOT NULL,
    Movie_ID INT NOT NULL,
    Process VARCHAR(50)
);

CREATE TABLE Cast (
    Member_Name VARCHAR(50),
    Member_ID INT NOT NULL,
    Role VARCHAR(50),
    CONSTRAINT PK_Cast PRIMARY KEY (Member_ID)
);

CREATE TABLE Cinema_Branch (
    Cinema_Name VARCHAR(50) NOT NULL,
    Cinema_Location VARCHAR(50) NOT NULL,
    Base_Price MONEY,
    Manger_ID INT,
    CONSTRAINT PK_Cinema PRIMARY KEY (Cinema_Name, Cinema_Location)
);

CREATE TABLE CinemaBranch_Snack (
    Cinema_Name VARCHAR(50) NOT NULL,
    Cinema_Location VARCHAR(50) NOT NULL,
    Snack_Name VARCHAR(50) NOT NULL,
    Available_Quantity INT,
    CONSTRAINT PK_CinemaBranch_Snack PRIMARY KEY (Cinema_Name, Cinema_Location, Snack_Name)
);

CREATE TABLE end_user (
    User_ID INT NOT NULL,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Age INT,
    Phone_Number VARCHAR(50),
    Email VARCHAR(50),
    Password VARCHAR(50),
    Role VARCHAR(50),
    CONSTRAINT PK_User PRIMARY KEY (User_ID)
);
```

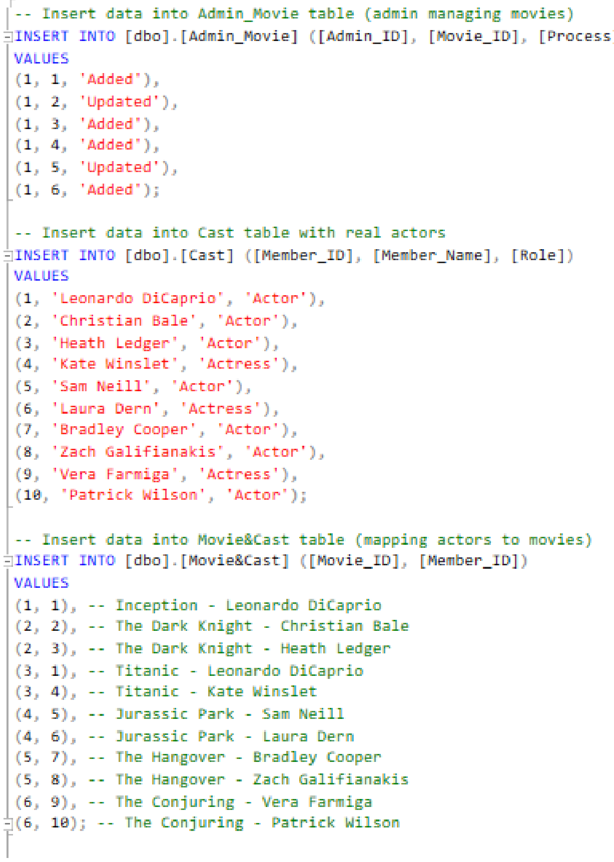


```
CREATE TABLE Has_a_snack (
    Order_ID INT NOT NULL,
    Snack_Name VARCHAR(50) NOT NULL,
    Quantity INT,
    CONSTRAINT PK_Has_a_snack PRIMARY KEY (Order_ID, Snack_Name)
);

CREATE TABLE [Movie&Cast] (
    Movie_ID INT NOT NULL,
    Member_ID INT NOT NULL,
    CONSTRAINT PK_MovieCast PRIMARY KEY (Movie_ID, Member_ID)
);

CREATE TABLE Movies (
    Movie_Name VARCHAR(50),
    Movie_ID INT NOT NULL,
    Duration INT,
    Genre VARCHAR(50),
    Rating FLOAT,
    Description VARCHAR(50),
    Parental_Guide VARCHAR(50),
    Release_Date DATE,
    Language VARCHAR(50),
    Trailer VARCHAR(50),
    Poster VARCHAR(MAX),
    CONSTRAINT PK_Movies PRIMARY KEY (Movie_ID)
);

CREATE TABLE orders (
    Order_ID INT NOT NULL,
    Date DATE,
    Payment_Method VARCHAR(50),
    User_ID_fk INT,
    CONSTRAINT PK_Order PRIMARY KEY (Order_ID)
);
```

2-Insertions Sample:



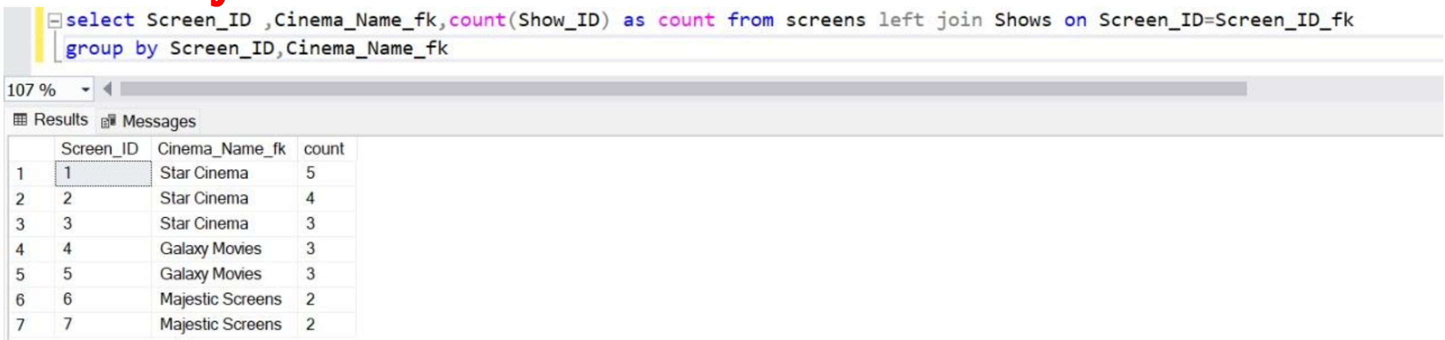
```sql
-- Insert data into end_user table first (includes admins, managers, and regular users)
INSERT INTO [dbo].[end_user] ([User_ID], [First_Name], [Last_Name], [Age], [Phone_Number], [Email], [Password], [Role])
VALUES
(1, 'John', 'Smith', 35, '555-1234', 'john.smith@email.com', 'password123', 'Admin'),
(2, 'Sarah', 'Johnson', 42, '555-2345', 'sarah.j@email.com', 'manager1', 'Manager'),
(3, 'Michael', 'Davis', 38, '555-3456', 'michael.d@email.com', 'manager2', 'Manager'),
(4, 'Robert', 'Wilson', 45, '555-4567', 'robert.w@email.com', 'manager3', 'Manager'),
(5, 'James', 'Brown', 29, '555-5678', 'james.b@email.com', 'user1', 'Customer'),
(6, 'Emily', 'Jones', 31, '555-6789', 'emily.j@email.com', 'user2', 'Customer'),
(7, 'Thomas', 'Anderson', 27, '555-7890', 'thomas.a@email.com', 'user3', 'Customer'),
(8, 'Lisa', 'Miller', 33, '555-8901', 'lisa.m@email.com', 'user4', 'Customer'),
(9, 'Daniel', 'Taylor', 24, '555-9012', 'daniel.t@email.com', 'user5', 'Customer'),
(10, 'Jennifer', 'Clark', 30, '555-0123', 'jennifer.c@email.com', 'user6', 'Customer');

-- Insert data into Cinema_Branch table
INSERT INTO [dbo].[Cinema_Branch] ([Cinema_Name], [Cinema_Location], [Base_Price], [Manger_ID])
VALUES
('Star Cinema', 'Downtown', 12.50, 2),
('Galaxy Movies', 'Westside Mall', 11.00, 3),
('Majestic Screens', 'Eastside Plaza', 13.50, 4);

-- Insert data into Types_of_Screen table
INSERT INTO [dbo].[Types_of_Screen] ([Screen_Type], [Seat_Factor])
VALUES
('Standard', 1.0),
('IMAX', 1.8),
('3D', 1.5),
('VIP', 2.0),
('4DX', 2.2);

-- Insert data into Screens table
INSERT INTO [dbo].[Screens] ([Screen_ID], [Screen_Number], [Cinema_Name_fk], [Cinema_Locaction_fk], [Screen_Type_fk], [Number_Of_Seats])
VALUES
(1, 1, 'Star Cinema', 'Downtown', 'Standard', 120),
(2, 2, 'Star Cinema', 'Downtown', 'IMAX', 180),
(3, 3, 'Star Cinema', 'Downtown', '3D', 100),
(4, 1, 'Galaxy Movies', 'Westside Mall', 'Standard', 130),
(5, 2, 'Galaxy Movies', 'Westside Mall', 'VIP', 60),
(6, 1, 'Majestic Screens', 'Eastside Plaza', 'Standard', 140),
(7, 2, 'Majestic Screens', 'Eastside Plaza', '4DX', 80);

-- Insert data into Movies table with real movies
INSERT INTO [dbo].[Movies] ([Movie_ID], [Movie_Name], [Duration], [Genre], [Rating], [Description], [Parental_Guide], [Release_Date], [Language], [Trailer], [Poster])
VALUES
(1, 'Inception', 148, 'Sci-Fi', 8.8, 'A thief who steals corporate secrets', 'PG-13', '2010-07-16', 'English', 'inception_trailer.mp4', 'inception.jpg'),
(2, 'The Dark Knight', 152, 'Action', 9.0, 'Batman fights the Joker', 'PG-13', '2008-07-18', 'English', 'dark_knight_trailer.mp4', 'dark_knight.jpg'),
(3, 'Titanic', 194, 'Romance', 7.9, 'Fictional romance on the doomed ship', 'PG-13', '1997-12-19', 'English', 'titanic_trailer.mp4', 'titanic.jpg'),
(4, 'Jurassic Park', 127, 'Adventure', 8.2, 'Dinosaurs come to life in a theme park', 'PG-13', '1993-06-11', 'English', 'jurassic_park_trailer.mp4', 'jurassic_park.jpg'),
(5, 'The Hangover', 100, 'Comedy', 7.7, 'Friends lose the groom at a bachelor party', 'R', '2009-06-05', 'English', 'hangover_trailer.mp4', 'hangover.jpg'),
(6, 'The Conjuring', 112, 'Horror', 7.5, 'Paranormal investigators help a family', 'R', '2013-07-19', 'English', 'conjuring_trailer.mp4', 'conjuring.jpg');
```



```sql
-- Insert data into Admin_Movie table (admin managing movies)
INSERT INTO [dbo].[Admin_Movie] ([Admin_ID], [Movie_ID], [Process])
VALUES
(1, 1, 'Added'),
(1, 2, 'Updated'),
(1, 3, 'Added'),
(1, 4, 'Added'),
(1, 5, 'Updated'),
(1, 6, 'Added');

-- Insert data into Cast table with real actors
INSERT INTO [dbo].[Cast] ([Member_ID], [Member_Name], [Role])
VALUES
(1, 'Leonardo DiCaprio', 'Actor'),
(2, 'Christian Bale', 'Actor'),
(3, 'Heath Ledger', 'Actor'),
(4, 'Kate Winslet', 'Actress'),
(5, 'Sam Neill', 'Actor'),
(6, 'Laura Dern', 'Actress'),
(7, 'Bradley Cooper', 'Actor'),
(8, 'Zach Galifianakis', 'Actor'),
(9, 'Vera Farmiga', 'Actress'),
(10, 'Patrick Wilson', 'Actor');

-- Insert data into Movie&Cast table (mapping actors to movies)
INSERT INTO [dbo].[Movie&Cast] ([Movie_ID], [Member_ID])
VALUES
(1, 1), -- Inception - Leonardo DiCaprio
(2, 2), -- The Dark Knight - Christian Bale
(2, 3), -- The Dark Knight - Heath Ledger
(3, 1), -- Titanic - Leonardo DiCaprio
(3, 4), -- Titanic - Kate Winslet
(4, 5), -- Jurassic Park - Sam Neill
(4, 6), -- Jurassic Park - Laura Dern
(5, 7), -- The Hangover - Bradley Cooper
(5, 8), -- The Hangover - Zach Galifianakis
(6, 9), -- The Conjuring - Vera Farmiga
(6, 10); -- The Conjuring - Patrick Wilson
```

3-Queries:

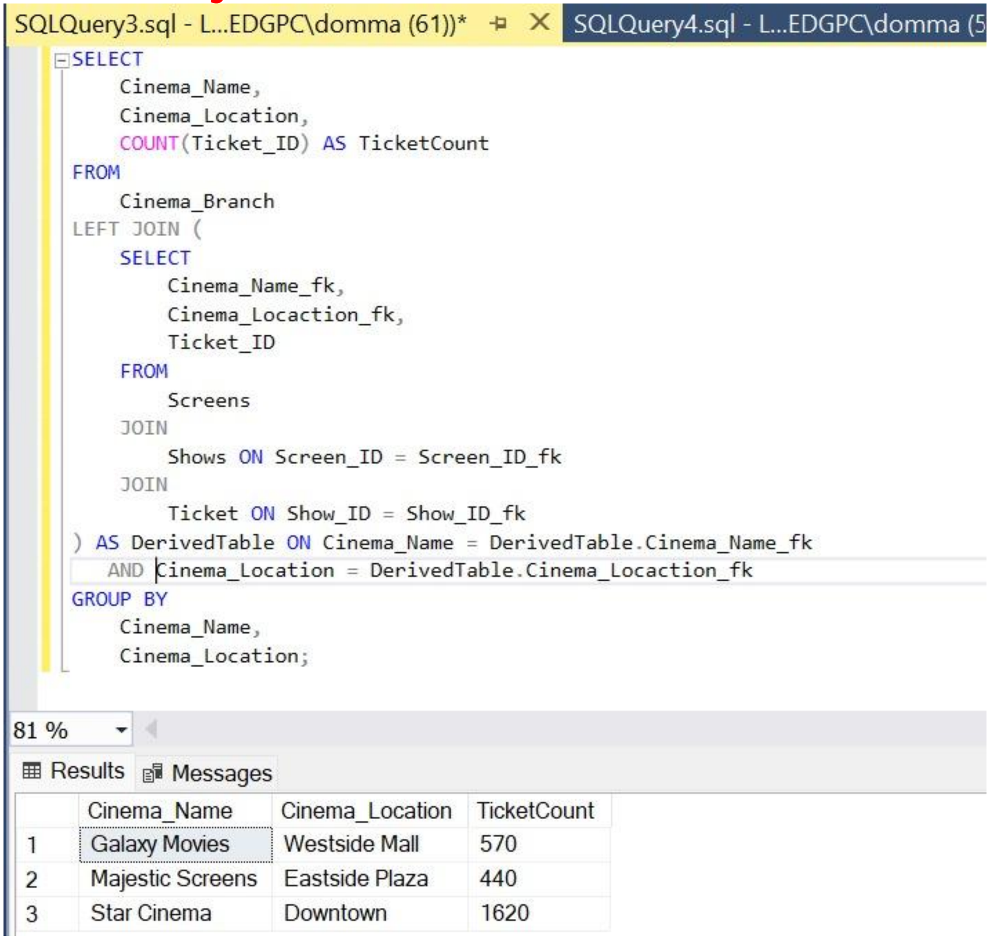
How many shows does each screen have:



```sql
select Screen_ID ,Cinema_Name_fk,count(Show_ID) as count from screens left join Shows on Screen_ID=Screen_ID_fk
group by Screen_ID,Cinema_Name_fk
```

	Screen_ID	Cinema_Name_fk	count
1	1	Star Cinema	5
2	2	Star Cinema	4
3	3	Star Cinema	3
4	4	Galaxy Movies	3
5	5	Galaxy Movies	3
6	6	Majestic Screens	2
7	7	Majestic Screens	2

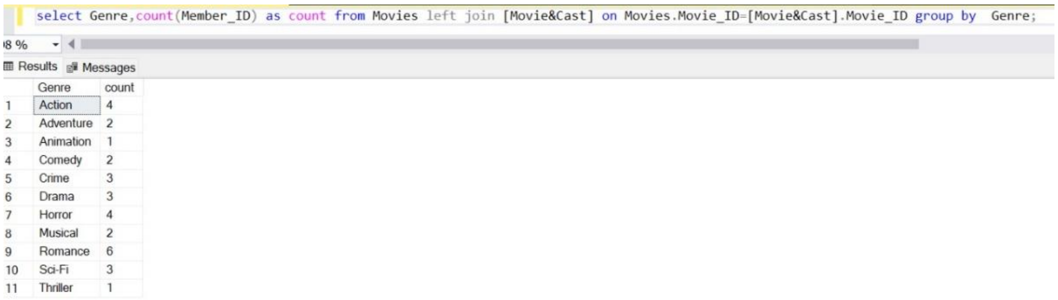
How many Tickets does Each cinema Offer


SQLQuery3.sql - L...EDGPC\domma (61))*
SQLQuery4.sql - L...EDGPC\domma (5
81 %
Results
Messages

```sql
SELECT
    Cinema_Name,
    Cinema_Location,
    COUNT(Ticket_ID) AS TicketCount
FROM
    Cinema_Branch
LEFT JOIN (
    SELECT
        Cinema_Name_fk,
        Cinema_Locaction_fk,
        Ticket_ID
    FROM
        Screens
    JOIN
        Shows ON Screen_ID = Screen_ID_fk
    JOIN
        Ticket ON Show_ID = Show_ID_fk
) AS DerivedTable ON Cinema_Name = DerivedTable.Cinema_Name_fk
   AND Cinema_Location = DerivedTable.Cinema_Locaction_fk
GROUP BY
    Cinema_Name,
    Cinema_Location;
```

	Cinema_Name	Cinema_Location	TicketCount
1	Galaxy Movies	Westside Mall	570
2	Majestic Screens	Eastside Plaza	440
3	Star Cinema	Downtown	1620

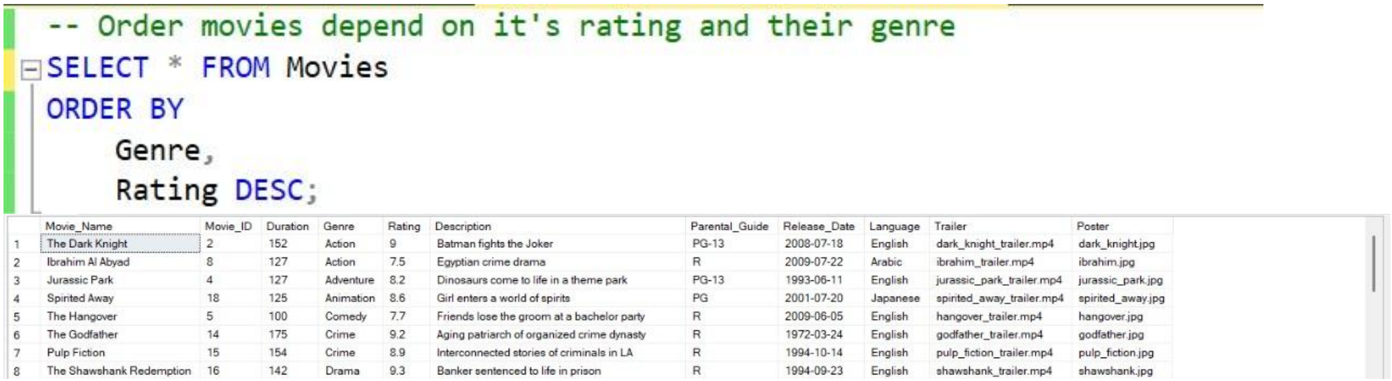
How many members of cast in each genre



```sql
select Genre,count(Member_ID) as count from Movies left join [Movie&Cast] on Movies.Movie_ID=[Movie&Cast].Movie_ID group by  Genre;
```

	Genre	count
1	Action	4
2	Adventure	2
3	Animation	1
4	Comedy	2
5	Crime	3
6	Drama	3
7	Horror	4
8	Musical	2
9	Romance	6
10	Sci-Fi	3
11	Thriller	1

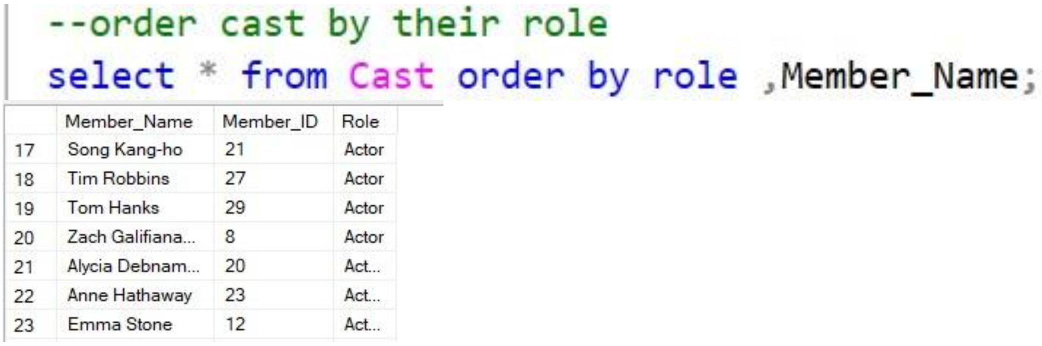
Movies listed in order of highest rating to least



```sql
-- Order movies depend on it's rating and their genre
SELECT * FROM Movies
ORDER BY
     Genre,
     Rating DESC;
```

	Movie_Name	Movie_ID	Duration	Genre	Rating	Description	Parental_Guide	Release_Date	Language	Trailer	Poster
1	The Dark Knight	2	152	Action	9	Batman fights the Joker	PG-13	2008-07-18	English	dark_knight_trailer.mp4	dark_knight.jpg
2	Ibrahim Al Abyad	8	127	Action	7.5	Egyptian crime drama	R	2009-07-22	Arabic	ibrahim_trailer.mp4	ibrahim.jpg
3	Jurassic Park	4	127	Adventure	8.2	Dinosaurs come to life in a theme park	PG-13	1993-06-11	English	jurassic_park_trailer.mp4	jurassic_park.jpg
4	Spirited Away	18	125	Animation	8.6	Girl enters a world of spirits	PG	2001-07-20	Japanese	spirited_away_trailer.mp4	spirited_away.jpg
5	The Hangover	5	100	Comedy	7.7	Friends lose the groom at a bachelor party	R	2009-06-05	English	hangover_trailer.mp4	hangover.jpg
6	The Godfather	14	175	Crime	9.2	Aging patriarch of organized crime dynasty	R	1972-03-24	English	godfather_trailer.mp4	godfather.jpg
7	Pulp Fiction	15	154	Crime	8.9	Interconnected stories of criminals in LA	R	1994-10-14	English	pulp_fiction_trailer.mp4	pulp_fiction.jpg
8	The Shawshank Redemption	16	142	Drama	9.3	Banker sentenced to life in prison	R	1994-09-23	English	shawshank_trailer.mp4	shawshank.jpg

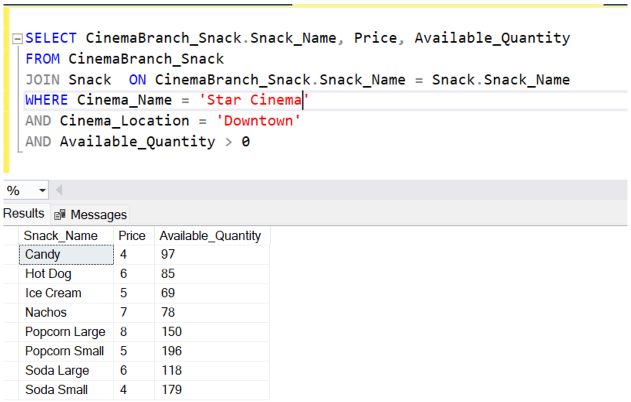
Viewing All cast members



```sql
--order cast by their role
select * from Cast order by role ,Member_Name;
```

	Member_Name	Member_ID	Role
17	Song Kang-ho	21	Actor
18	Tim Robbins	27	Actor
19	Tom Hanks	29	Actor
20	Zach Galifiana...	8	Actor
21	Alycia Debnam...	20	Act...
22	Anne Hathaway	23	Act...
23	Emma Stone	12	Act...

Find available snacks at a specific cinema:

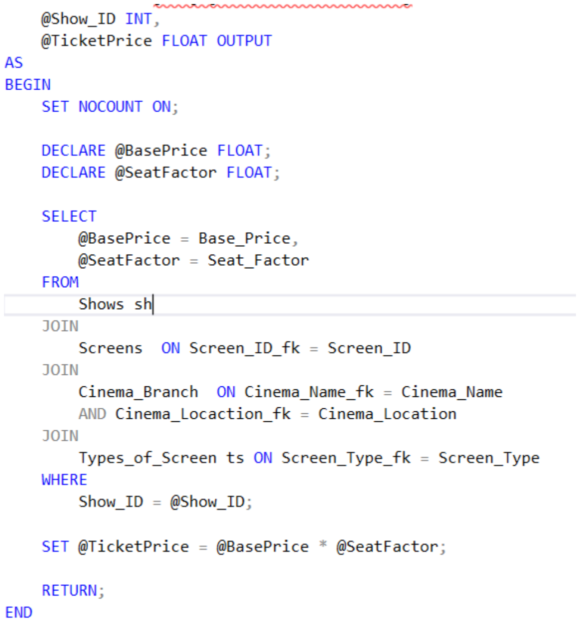


```sql
SELECT CinemaBranch_Snack.Snack_Name, Price, Available_Quantity
FROM CinemaBranch_Snack
JOIN Snack  ON CinemaBranch_Snack.Snack_Name = Snack.Snack_Name
WHERE Cinema_Name = 'Star Cinema'
AND Cinema_Location = 'Downtown'
AND Available_Quantity > 0
```

Snack_Name	Price	Available_Quantity
Candy	4	97
Hot Dog	6	85
Ice Cream	5	69
Nachos	7	78
Popcorn Large	8	150
Popcorn Small	5	196
Soda Large	6	118
Soda Small	4	179

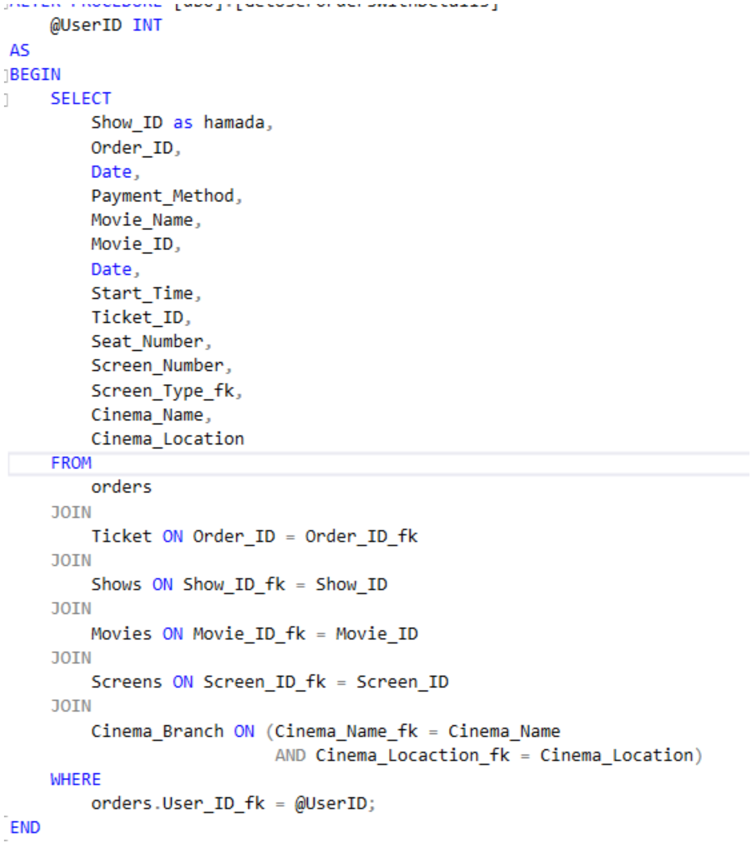
3-Procedures:

-CalculateTicketPrice:



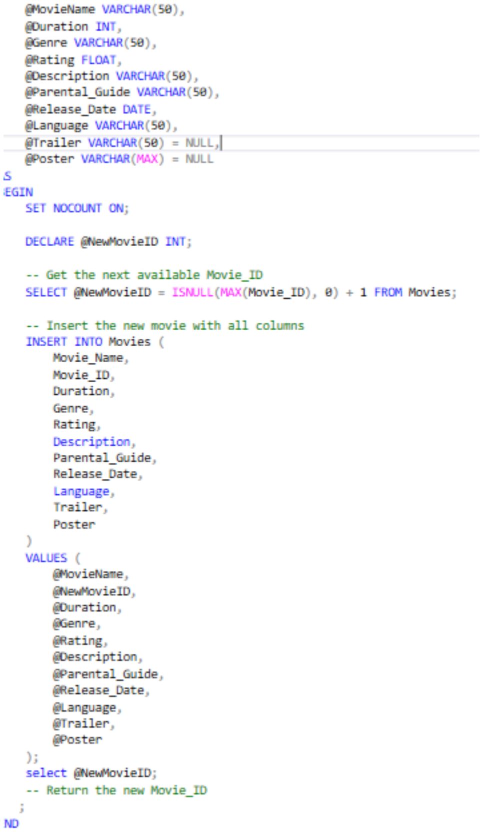
```
    @Show_ID INT,
    @TicketPrice FLOAT OUTPUT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @BasePrice FLOAT;
    DECLARE @SeatFactor FLOAT;

    SELECT
        @BasePrice = Base_Price,
        @SeatFactor = Seat_Factor
    FROM
        Shows sh
    JOIN
        Screens  ON Screen_ID_fk = Screen_ID
    JOIN
        Cinema_Branch  ON Cinema_Name_fk = Cinema_Name
        AND Cinema_Locaction_fk = Cinema_Location
    JOIN
        Types_of_Screen ts ON Screen_Type_fk = Screen_Type
    WHERE
        Show_ID = @Show_ID;

    SET @TicketPrice = @BasePrice * @SeatFactor;

    RETURN;
END
```

-GetUserOrdersWithDetails



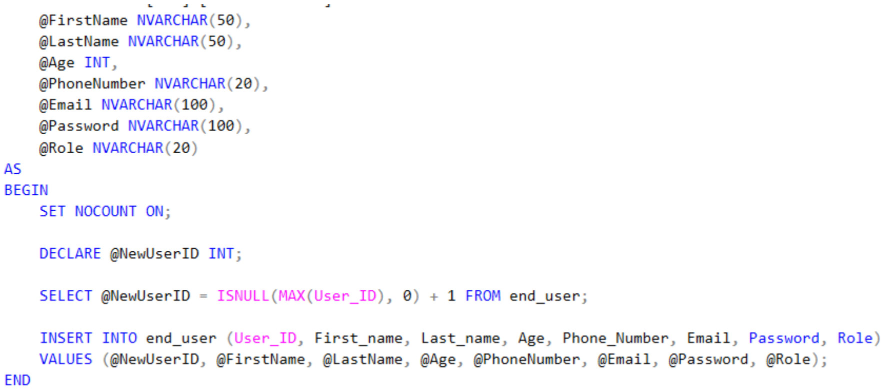
```
    @UserID INT
AS
BEGIN
    SELECT
        Show_ID as hamada,
        Order_ID,
        Date,
        Payment_Method,
        Movie_Name,
        Movie_ID,
        Date,
        Start_Time,
        Ticket_ID,
        Seat_Number,
        Screen_Number,
        Screen_Type_fk,
        Cinema_Name,
        Cinema_Location
    FROM
        orders
    JOIN
        Ticket ON Order_ID = Order_ID_fk
    JOIN
        Shows ON Show_ID_fk = Show_ID
    JOIN
        Movies ON Movie_ID_fk = Movie_ID
    JOIN
        Screens ON Screen_ID_fk = Screen_ID
    JOIN
        Cinema_Branch ON (Cinema_Name_fk = Cinema_Name
                      AND Cinema_Locaction_fk = Cinema_Location)
    WHERE
        orders.User_ID_fk = @UserID;
END
```

-InsertNewMovie



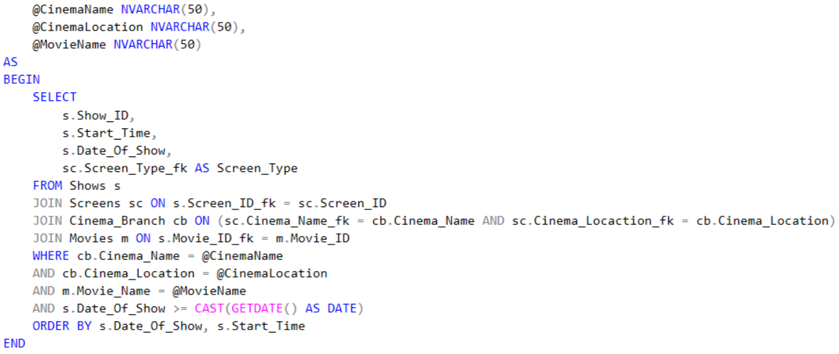
```
    @MovieName VARCHAR(50),
    @Duration INT,
    @Genre VARCHAR(50),
    @Rating FLOAT,
    @Description VARCHAR(50),
    @Parental_Guide VARCHAR(50),
    @Release_Date DATE,
    @Language VARCHAR(50),
    @Trailer VARCHAR(50) = NULL,
    @Poster VARCHAR(MAX) = NULL
S
EGIN
    SET NOCOUNT ON;

    DECLARE @NewMovieID INT;

    -- Get the next available Movie_ID
    SELECT @NewMovieID = ISNULL(MAX(Movie_ID), 0) + 1 FROM Movies;

    -- Insert the new movie with all columns
    INSERT INTO Movies (
        Movie_Name,
        Movie_ID,
        Duration,
        Genre,
        Rating,
        Description,
        Parental_Guide,
        Release_Date,
        Language,
        Trailer,
        Poster
    )
    VALUES (
        @MovieName,
        @NewMovieID,
        @Duration,
        @Genre,
        @Rating,
        @Description,
        @Parental_Guide,
        @Release_Date,
        @Language,
        @Trailer,
        @Poster
    );
    select @NewMovieID;
    -- Return the new Movie_ID
  ;
ND
```

-InsertNewUser



```
    @FirstName NVARCHAR(50),
    @LastName NVARCHAR(50),
    @Age INT,
    @PhoneNumber NVARCHAR(20),
    @Email NVARCHAR(100),
    @Password NVARCHAR(100),
    @Role NVARCHAR(20)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @NewUserID INT;

    SELECT @NewUserID = ISNULL(MAX(User_ID), 0) + 1 FROM end_user;

    INSERT INTO end_user (User_ID, First_name, Last_name, Age, Phone_Number, Email, Password, Role)
    VALUES (@NewUserID, @FirstName, @LastName, @Age, @PhoneNumber, @Email, @Password, @Role);
END
```

sp_GetMovieShowtimes



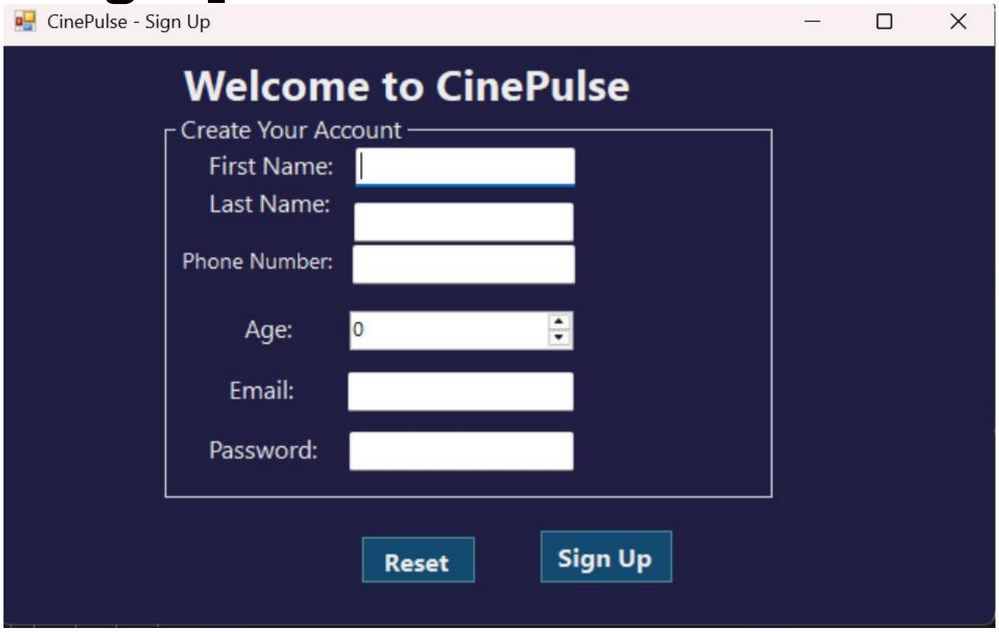
```
    @CinemaName NVARCHAR(50),
    @CinemaLocation NVARCHAR(50),
    @MovieName NVARCHAR(50)
AS
BEGIN
    SELECT
        s.Show_ID,
        s.Start_Time,
        s.Date_Of_Show,
        sc.Screen_Type_fk AS Screen_Type
    FROM Shows s
    JOIN Screens sc ON s.Screen_ID_fk = sc.Screen_ID
    JOIN Cinema_Branch cb ON (sc.Cinema_Name_fk = cb.Cinema_Name AND sc.Cinema_Locaction_fk = cb.Cinema_Location)
    JOIN Movies m ON s.Movie_ID_fk = m.Movie_ID
    WHERE cb.Cinema_Name = @CinemaName
    AND cb.Cinema_Location = @CinemaLocation
    AND m.Movie_Name = @MovieName
    AND s.Date_Of_Show >= CAST(GETDATE() AS DATE)
    ORDER BY s.Date_Of_Show, s.Start_Time
END
```

4-Constraints:

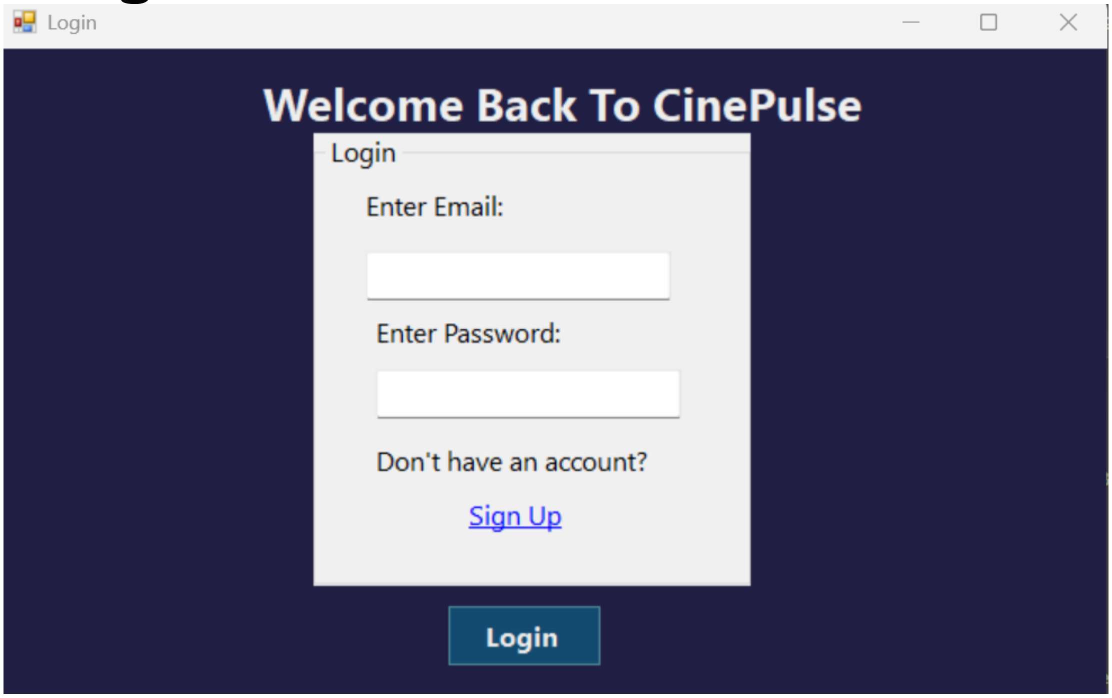
```
ALTER TABLE Users
ADD CONSTRAINT chk_age CHECK (age >= 13);
----------------------------------------------------

ALTER TABLE Movies ADD CONSTRAINT
CK_Movie_ParentalGuide
CHECK (Parental_Guide IN ('G', 'PG', 'PG-13', 'R', 'NC-17'));
------------------------------------------------------------------------------
-- Ensure seat count is reasonable
ALTER TABLE Screens ADD CONSTRAINT CK_Screen_Seats
CHECK (Number_Of_Seats > 0 AND Number_Of_Seats <= 500);
------------------------------------------------------------------------------------

-- Validate screen types
ALTER TABLE Screens ADD CONSTRAINT CK_Screen_Type
CHECK (Screen_Type_fk IN ('Standard', 'IMAX', '3D', 'VIP', '4DX'));
----------------------------------------------------------------------------------

ALTER TABLE end_user ADD CONSTRAINT CK_User_Role
CHECK (Role IN ('Admin', 'Manager', 'Customer'));


ALTER TABLE Ticket ADD CONSTRAINT CK_Ticket_SeatNumber
CHECK (Seat_Number > 0);
```

4-Gui ScreenShots

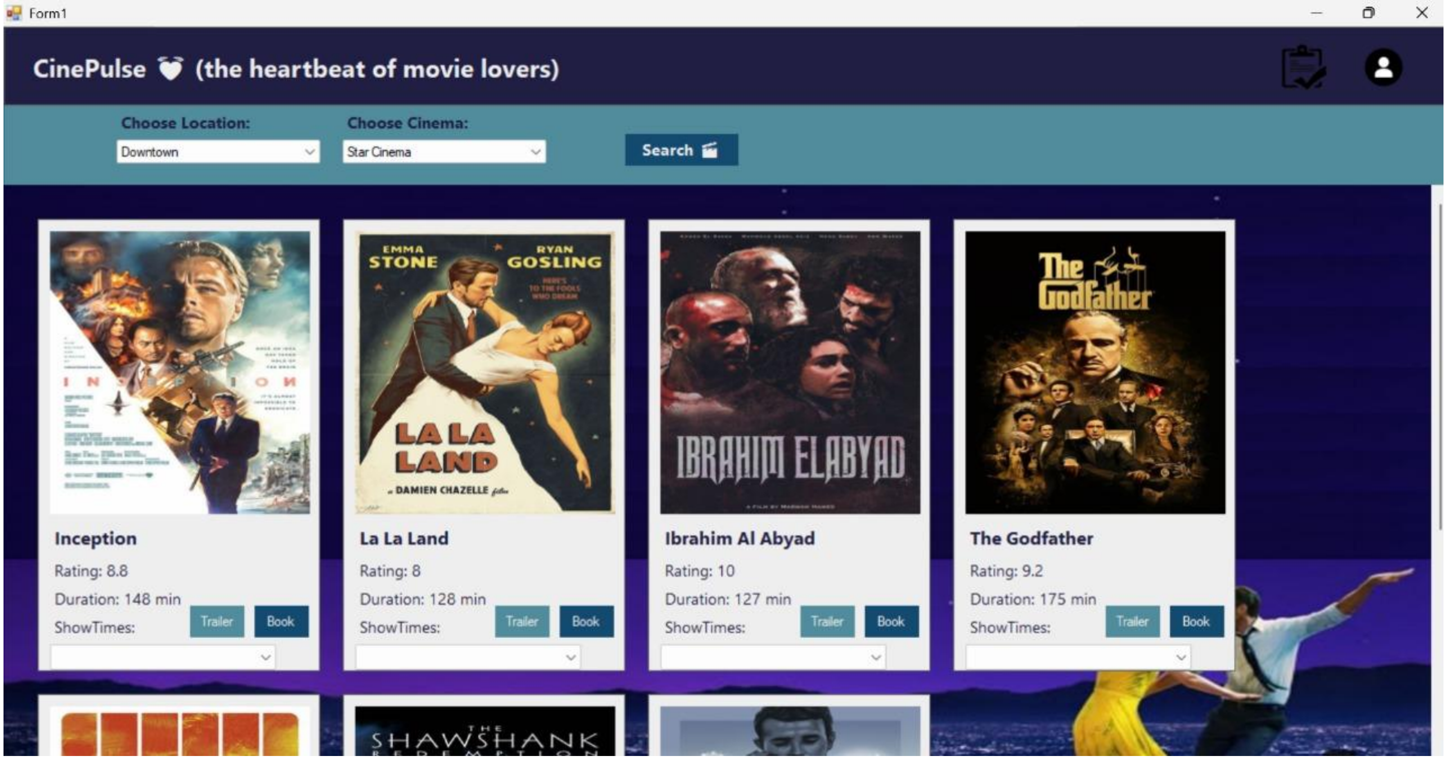
1-Signup



2-Login



3-Customer HomePage with movies shown



Form1

CinePulse (the heartbeat of movie lovers)

Choose Location: Downtown

Choose Cinema: Star Cinema

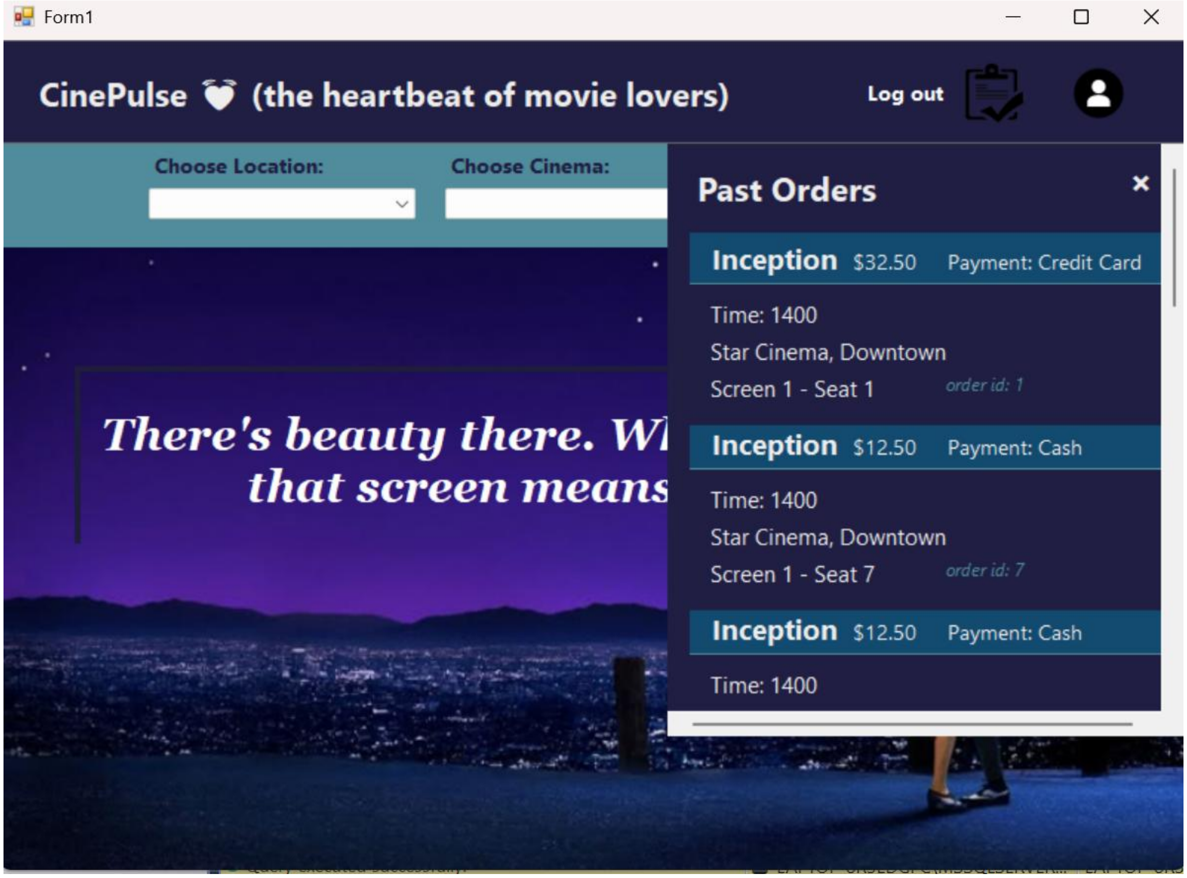
Inception
Rating: 8.8
Duration: 148 min
ShowTimes:

La La Land
Rating: 8
Duration: 128 min
ShowTimes:

Ibrahim Al Abyad
Rating: 10
Duration: 127 min
ShowTimes:

The Godfather
Rating: 9.2
Duration: 175 min
ShowTimes:

4-Past orders



Form1

CinePulse (the heartbeat of movie lovers)

Choose Location:

Choose Cinema:

There's beauty there. Wi
that screen means

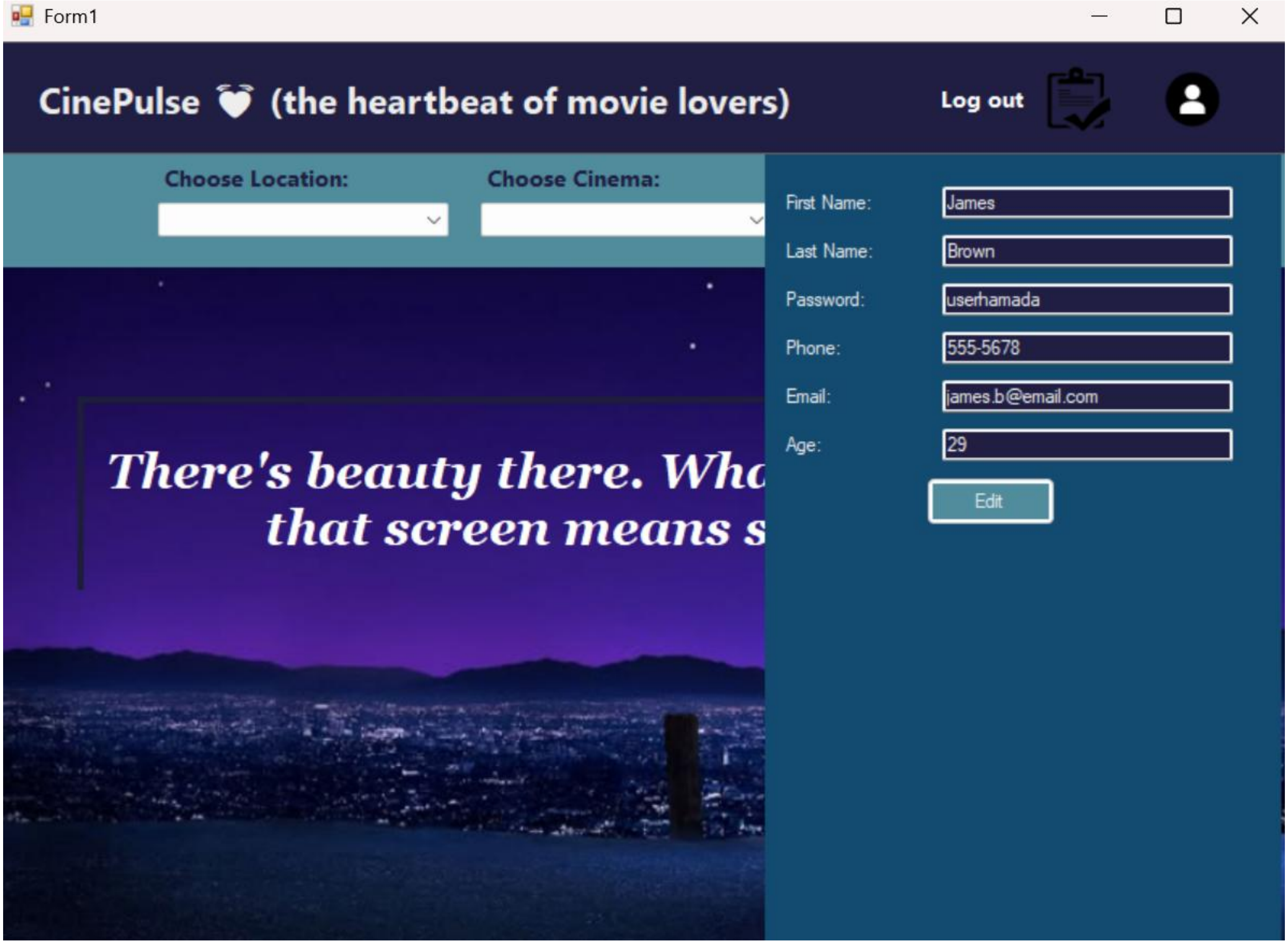
Past Orders

Inception $32.50 Payment: Credit Card
Time: 1400
Star Cinema, Downtown
Screen 1 - Seat 1 *order id: 1*

Inception $12.50 Payment: Cash
Time: 1400
Star Cinema, Downtown
Screen 1 - Seat 7 *order id: 7*

Inception $12.50 Payment: Cash
Time: 1400

5-View Profile/Edit


Form1
CinePulse (the heartbeat of movie lovers)
Log out
Choose Location:
Choose Cinema:
There's beauty there. Wha
that screen means s
Edit

First Name: James

Last Name: Brown

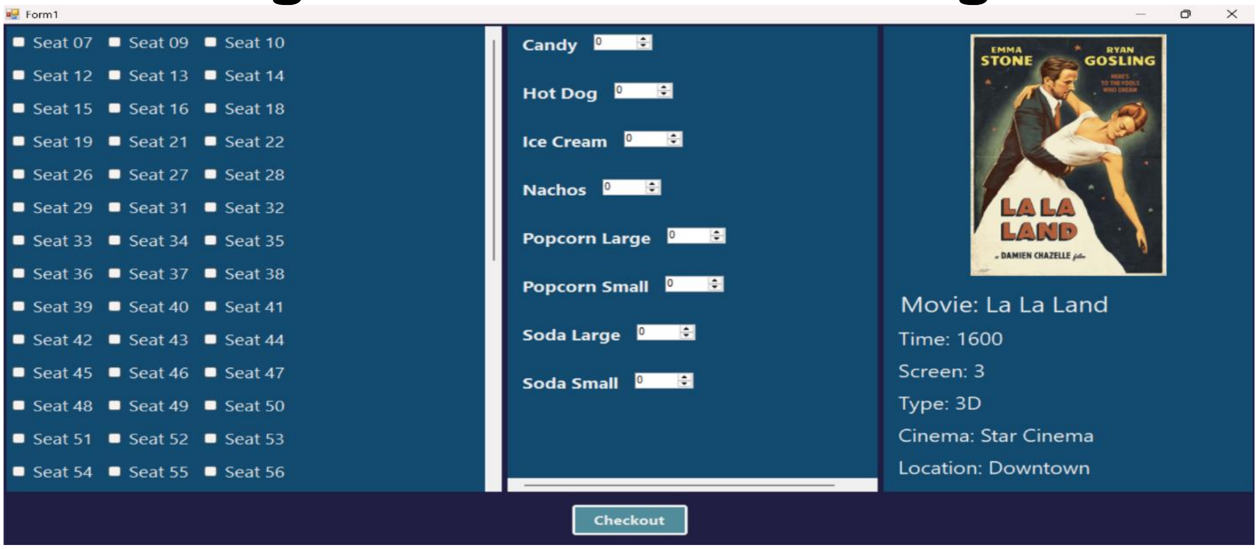
Password: userhamada

Phone: 555-5678

Email: james.b@email.com

Age: 29

6-Choosing seats/Snacks/Finalizing Order



□ Seat 07 □ Seat 09 □ Seat 10
□ Seat 12 □ Seat 13 □ Seat 14
□ Seat 15 □ Seat 16 □ Seat 18
□ Seat 19 □ Seat 21 □ Seat 22
□ Seat 26 □ Seat 27 □ Seat 28
□ Seat 29 □ Seat 31 □ Seat 32
□ Seat 33 □ Seat 34 □ Seat 35
□ Seat 36 □ Seat 37 □ Seat 38
□ Seat 39 □ Seat 40 □ Seat 41
□ Seat 42 □ Seat 43 □ Seat 44
□ Seat 45 □ Seat 46 □ Seat 47
□ Seat 48 □ Seat 49 □ Seat 50
□ Seat 51 □ Seat 52 □ Seat 53
□ Seat 54 □ Seat 55 □ Seat 56

Candy 0

Hot Dog 0

Ice Cream 0

Nachos 0

Popcorn Large 0

Popcorn Small 0

Soda Large 0

Soda Small 0

Movie: La La Land

Time: 1600

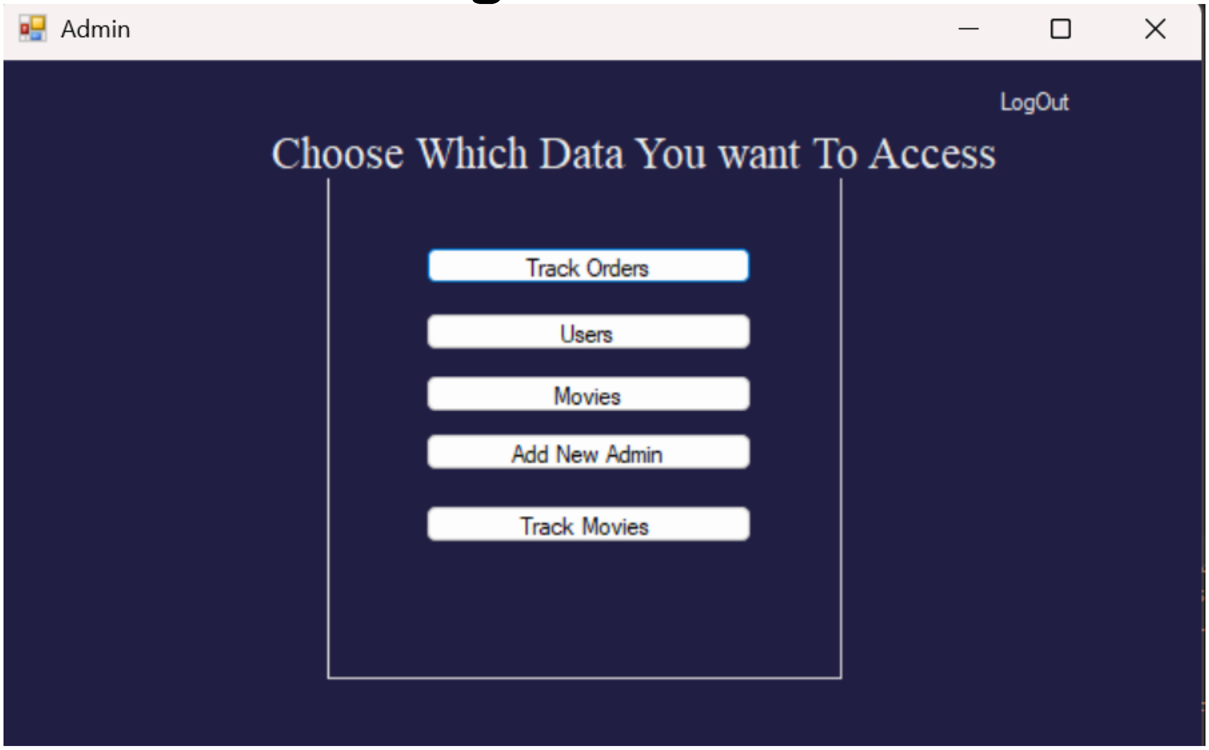
Screen: 3

Type: 3D

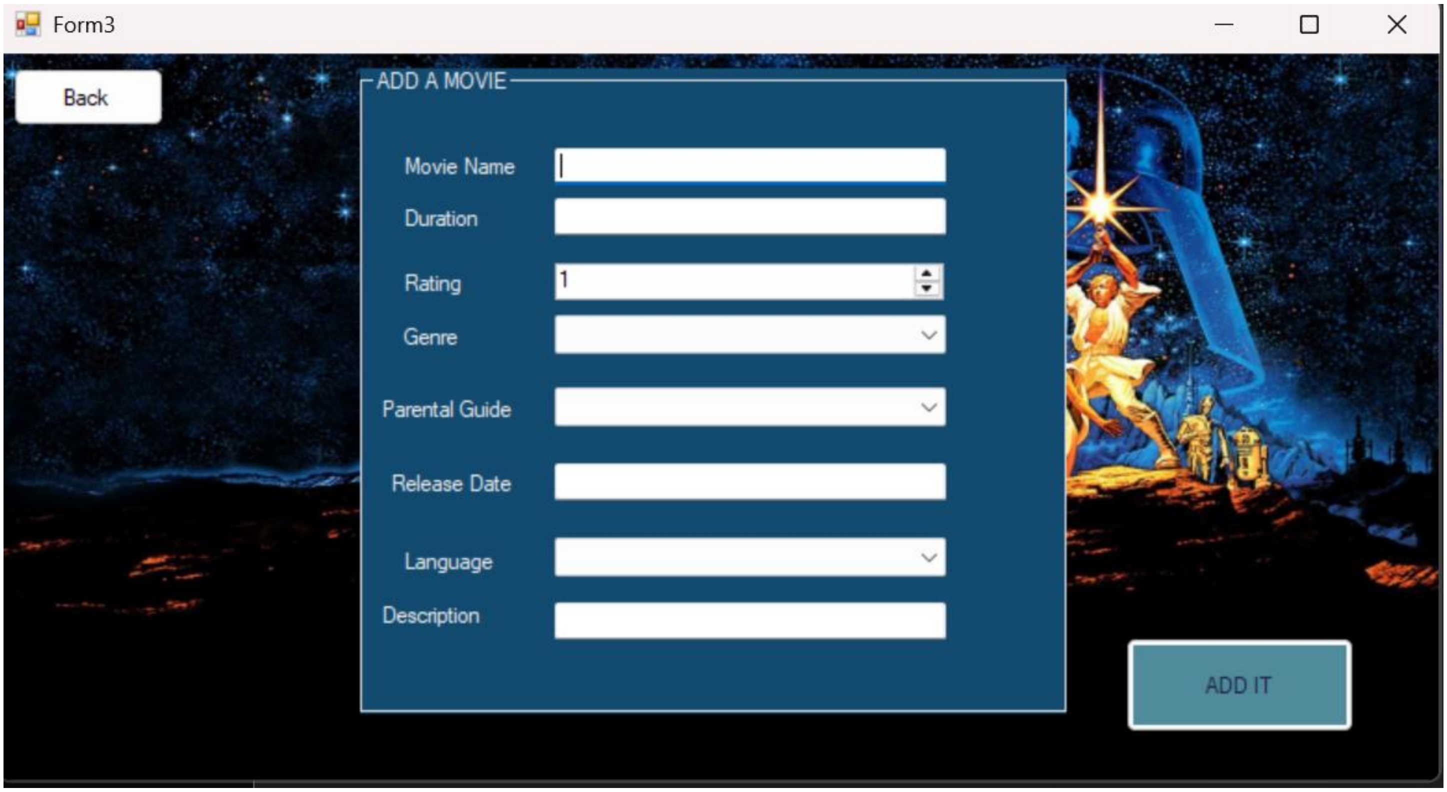
Cinema: Star Cinema

Location: Downtown

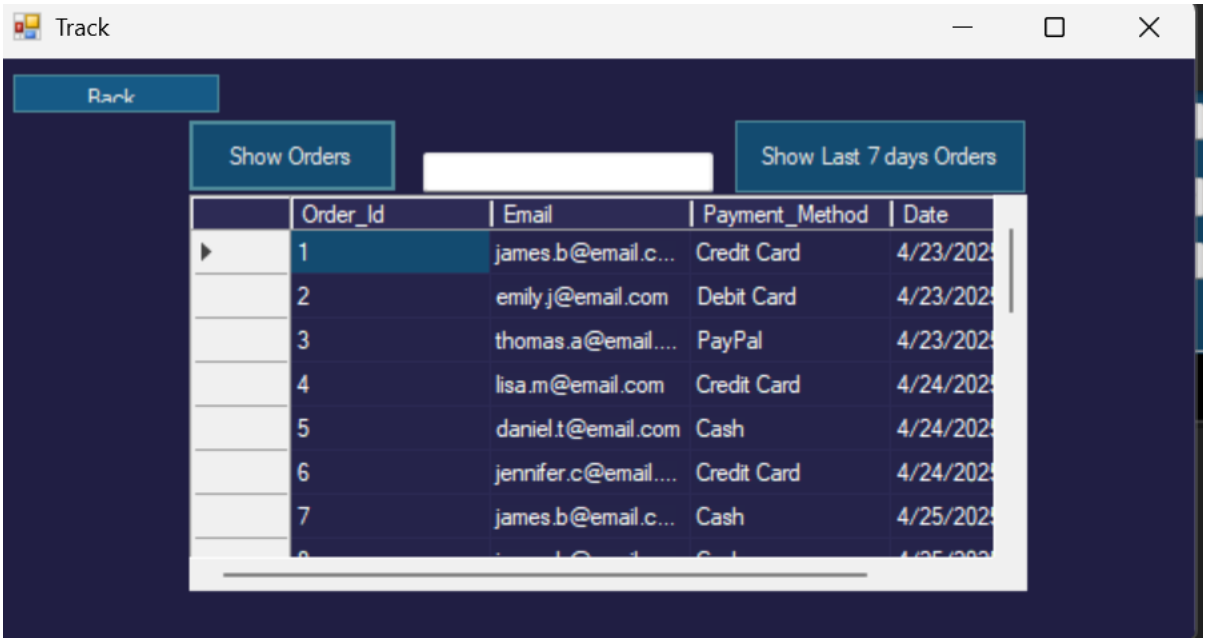
7-Admin HomePage



7-Add A Movie

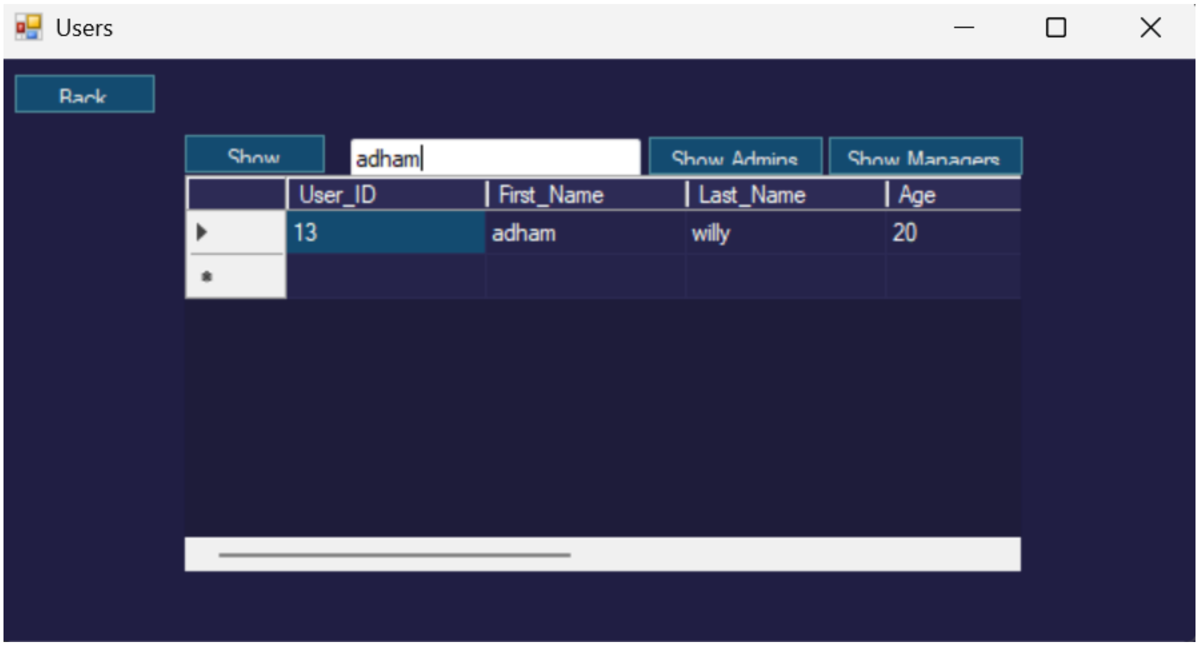


8-View Orders



Order_Id	Email	Payment_Method	Date
1	james.b@email.c...	Credit Card	4/23/202
2	emily.j@email.com	Debit Card	4/23/202
3	thomas.a@email....	PayPal	4/23/202
4	lisa.m@email.com	Credit Card	4/24/202
5	daniel.t@email.com	Cash	4/24/202
6	jennifer.c@email....	Credit Card	4/24/202
7	james.b@email.c...	Cash	4/25/202

9-View Users

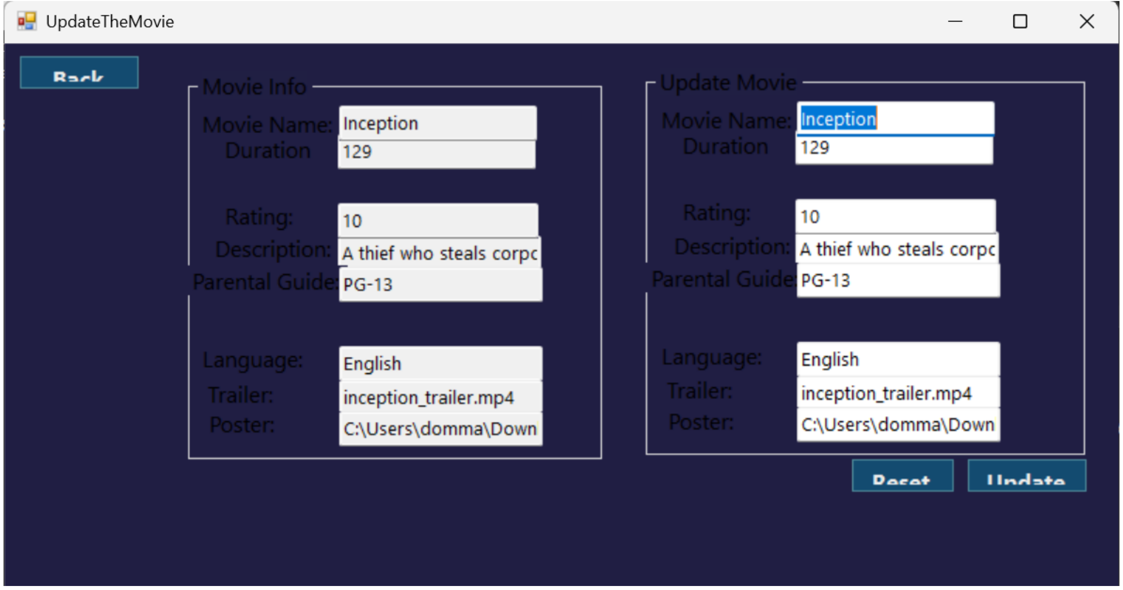


adham

User_ID	First_Name	Last_Name	Age
13	adham	willy	20

Note:all Admins Functions are mainly view,except only in movies he can add and delete too

10-Update Movie:



Movie Info

Movie Name: Inception
Duration 129
Rating: 10
Description: A thief who steals corpc
Parental Guide: PG-13
Language: English
Trailer: inception_trailer.mp4
Poster: C:\Users\domma\Down

Update Movie

Movie Name: Inception
Duration 129
Rating: 10
Description: A thief who steals corpc
Parental Guide: PG-13
Language: English
Trailer: inception_trailer.mp4
Poster: C:\Users\domma\Down

5-Scenario:

The cinema booking system provides a user-friendly graphical interface that integrates seamlessly with a backend database to manage user and booking data efficiently. The application begins with a login or sign-up process. When a new user signs up, their information is stored in the database and they are registered as a customer. Upon logging in, the system distinguishes between two types of users: customers and administrators. Customers have access to features such as viewing and editing their personal data, checking their past orders, and browsing available shows. They can select a specific location and choose a cinema to explore available movies and showtimes. Once a show is selected, the customer can proceed to select their preferred seats and snacks before finalizing the order. The completed order is then recorded in the database, the selected seats are marked as reserved, and the order is added to the customer's past orders history. On the other hand, administrators have broader access privileges within the system. After logging in, an admin can view comprehensive data from the database, including orders, users, and movie listings. They also have the authority to manage movies by adding new entries, updating existing ones, or removing outdated films. This structured workflow ensures both user accessibility and administrative control, maintaining system integrity and usability.