JSS MAHAVIDYAPEETHA
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
JSS Technical Institutions Campus, Mysuru – 570006

"CINE VERSE"

Report submitted in fulfillment of curriculum prescribed for the Database Management Systems (20CS510) course for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

by

INDRATEJ YT *(01JST20CS069)*	*PRADYUMNA.G. NAIK* *(01JST20CS114)*
ROHAN ACHAR *(01JST20CS131)*	*PUNEETH H.B* *(01JST20CS125)*

Under the Guidance of

Dr. Manimala S
Associate Professor,
Dept. Of CS & E,
SJCE, JSSSTU Mysore

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JSS
SCIENCE AND
TECHNOLOGY
UNIVERSITY
MYSURU

CERTIFICATE

This is to certify that the work entitled **“CINE VERSE**- **Movie Booking website”** is a Bonafide work carried out **by INDRATEJ Y T, PRADYUMNA G NAIK, PUNEETH H B and ROHAN ACHAR V** in fulfillment of the award of the degree of **Bachelor of Engineering in Computer Science and Engineering of JSS Science and Technology, Mysuru during the year 2022**. It is certified that all corrections / suggestions indicated during CIE have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the Database Management Systems (20CS510) course.

Course in Charge and Guide

Dr. Manimala S
Associate Professor,
Dept. of CS&E
SJCE, JSSSTU, Mysore

Place: Mysore

Date: 25-01-2023

CONTENTS

ABSTRACT

The online movie ticket system provides a website for a cinema hall where any internet user can access it. User is required to login to the system and needs a Credit / Debit card for booking the tickets. Watching movies with family and friends in theatres is one of the best medium of entertainment after having a hectic schedule. But all this excitement vanishes after standing in hours in long queues to get tickets booked.

The customer will need to enter the information related to film after which the system will provide complete information regarding currently running movies with details of show timings and available seats. The user can order his favorite snacks in the website itself . If seats are available, the customer can change the position of the seats while they will get an option to cancel of tickets .

Our online movie ticket system **CINE VERSE** is one of the best opportunities for those who cannot afford enough time to get their tickets reserved standing in long queue.

In this system, the staff of the cinema hall won't have to do anything for the ticket booking as the process will be done by the customer through the website which will reduce the amount of staff required in the booking counter.

INTRODUCTION

A ***Movie booking system*** is a database management system that is designed to handle the ticket reservation and seat allocation process for a movie theater. It allows customers to browse and select available showtimes, choose their seats, and purchase tickets online. This document gives detailed functional and non-functional requirements for the Online Movie Ticket Booking System. This application is basically made for providing customer an anytime and anywhere service for booking the seat in the cinema hall and to gather information about the movies online.

Movie Ticketing System Database Design is basically aimed to provide complete information of the movie and schedule to the customer, according to which he/she can easily get a ticket instantly and can book a ticket on his/her favorite movies. This application will reserve the tickets. It allows customers to browse and select available showtimes, choose their seats, and purchase tickets online.

OBJECTIVES OF THE PROJECT: -

In an Online Movie Booking System application, the main aim of this application is online booking tickets for Various movies in Registered theatres by the users and to provide a convenient way to the users to book the tickets for cinema - hall online, through which they can book tickets anytime and anywhere.

- Customer Satisfaction which is the main priority of any business is to provide customers with the very best services and to meet all their needs. An online booking platform helps a movie theatre achieve this.
- Facility to store the information of new customer & different types of movie show timing & ticket rates of different types on show class etc.

- Interest to develop a good user friendly website with many online transactions using a database.
- Facility to generate different reports & which are helpful for the management in decision making.
- Facility to change user's password account.
- Analytics in which Data collected digitally is fairly easy to analyze. Customers can also give their feedback concerning the services which the business can take into consideration

PURPOSE AND SCOPE OF THE PROJECT: -

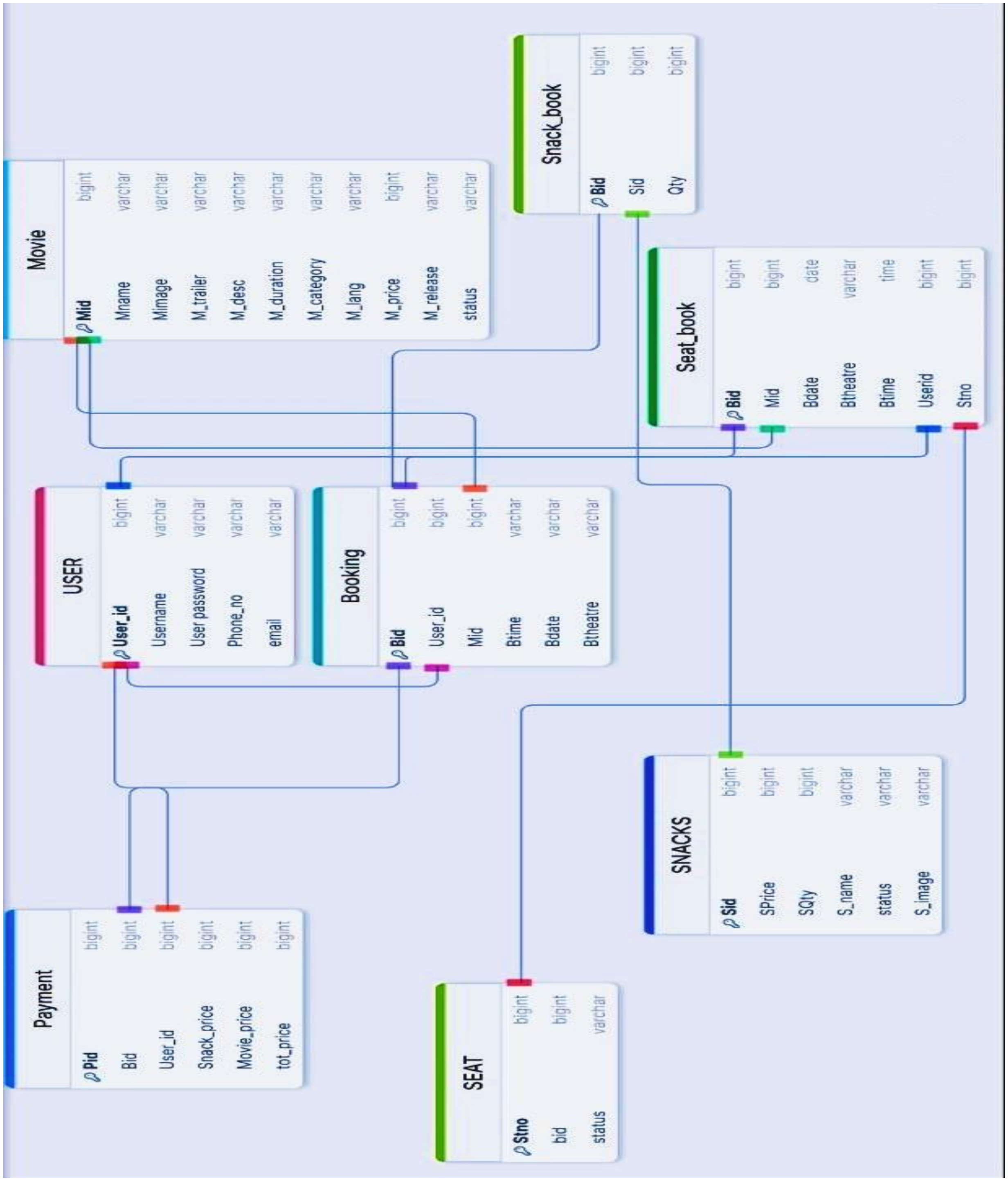
- The main purpose of our online ticket booking system is to provide another way for the user to buy cinema ticket. It is an automatic system.
- After inserting the data to database, staff need not to due with the order receive through the system. In fact, there is similar system on the internet, but there is no refund method found in the existing system.
- This system is basically aimed to provide the user the complete information of the movie, according to which the user can book the tickets and along with can order snacks and the refund facility provides more flexibility to the system.
- The goals of our system are:

    1. To provide an anytime anyplace service for the user.
    2. To minimize the number of staff at the ticket box.
    3. To promote the film on the internet.
    4. To increase the profit to obtain statistic information from the booking record.

FEATURES OF THE PROJECT : -

- **Admin Login-** allows authorized individuals to access and manage the database. This may include adding, editing, and deleting movies and other content, as well as managing user accounts and permissions.
- **User Register –** User have to create their account in our app. So that they get the notification about the latest movies released.
- **User Login -** allows registered users to access and interact with the database. This may include searching for movies, viewing details about movies, leaving ratings and reviews, and creating lists of favorite movies
- **Book Ticket -** The process of booking a ticket would typically involve the following steps:
  - ***Search for the movie:*** Users can search for the movie they want to watch by title, release date, or theater.
  - ***Select showtime***: Once the movie is found, the user can select the date and time of the show they want to attend.
  - ***Select seats:*** Users can select the seats they want to book from a seating chart, which shows the availability of seats in the theater.
  - ***Select snacks*****:** users would be able to select and purchase snacks to be delivered to their seats or pick them up at the theater.
- **Show Ticket status-** Users can check the status of a specific ticket and it shows the number of seats booked and snacks ordered and its individual price and the total price.
- **Payment –** User should pay the money after booking ticket and If user ordered snacks then its payment is also done together with ticket payment.
- **Cancel Ticket -** In some situation, if user want to cancel the ticket then user can do this if and only if certain terms and conditions are fulfilled.

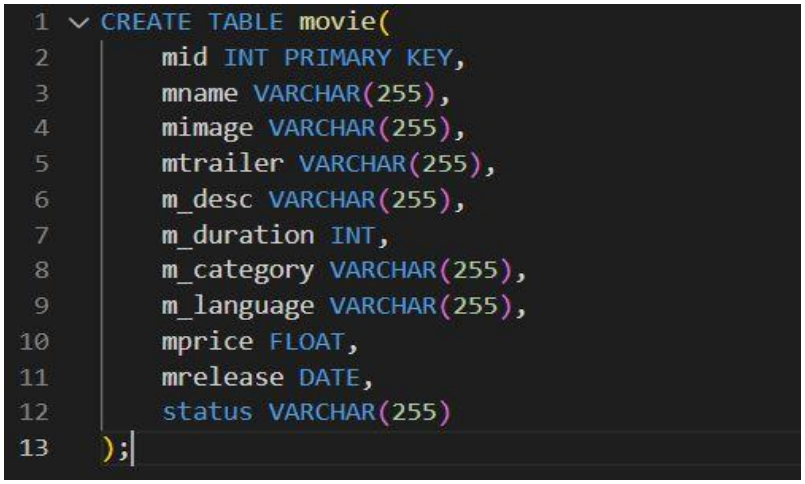
SCHEMA DIAGRAM



BACKEND IMPLEMENTATION

DATABASE TABLES CREATION

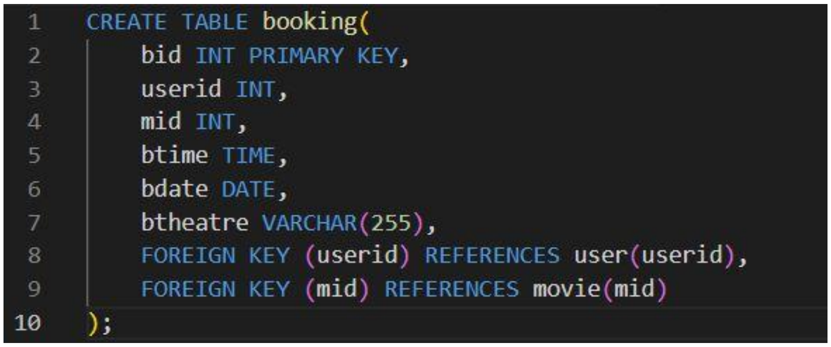
1. movie table



```sql
CREATE TABLE movie(
    mid INT PRIMARY KEY,
    mname VARCHAR(255),
    mimage VARCHAR(255),
    mtrailer VARCHAR(255),
    m_desc VARCHAR(255),
    m_duration INT,
    m_category VARCHAR(255),
    m_language VARCHAR(255),
    mprice FLOAT,
    mrelease DATE,
    status VARCHAR(255)
);
```

- Movie table is used to store the description of movies like name, image, trailer, duration, category, language, release date and present status.
- mid is the primary Key.

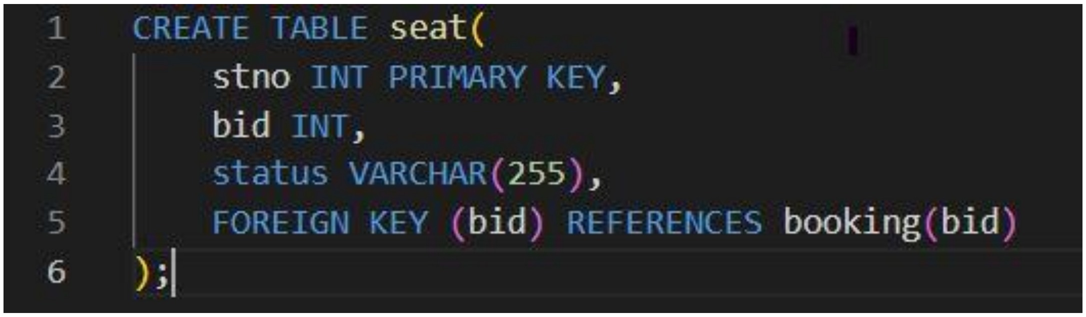
2. Booking Table



```sql
CREATE TABLE booking(
    bid INT PRIMARY KEY,
    userid INT,
    mid INT,
    btime TIME,
    bdate DATE,
    btheatre VARCHAR(255),
    FOREIGN KEY (userid) REFERENCES user(userid),
    FOREIGN KEY (mid) REFERENCES movie(mid)
);
```

- Booking table is used to store the booking details of each customer. bid is the primary key.
- It stores the movie booked, time and date of booking and theatre booked by each customer.
- Foreign key userid and mid references user and movie table.

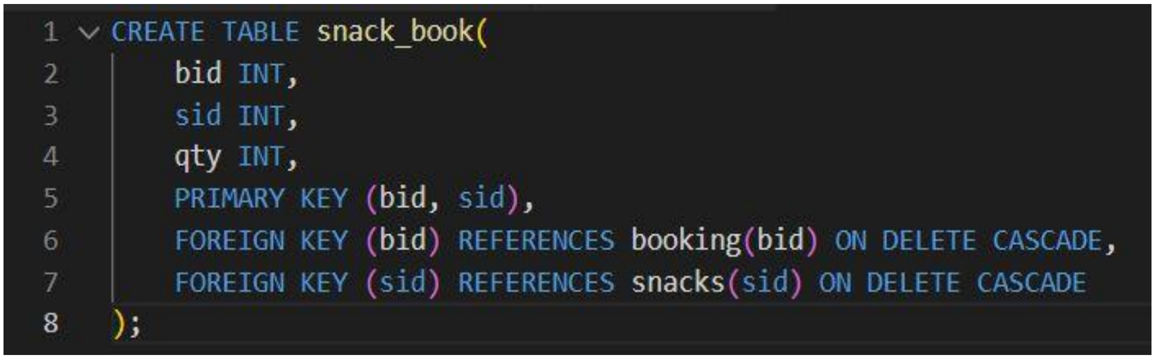
3. seat table



```sql
CREATE TABLE seat(
    stno INT PRIMARY KEY,
    bid INT,
    status VARCHAR(255),
    FOREIGN KEY (bid) REFERENCES booking(bid)
);
```

- Seat table is used to store the details of the seats booked and the status of the booking in order to accommodate different theatres and booking time in the database.
- Status is set to available by default and is updated to Not available as and when the user books a seat.
- Seat table is updated every single time a user goes on to select a new movie for different time.

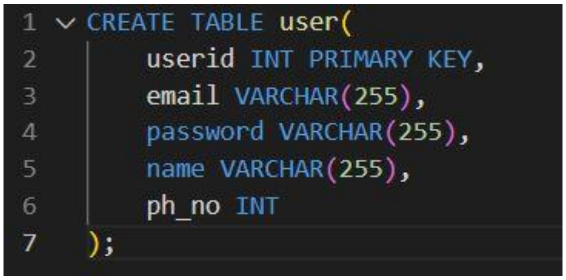
4. snack_book table



```sql
CREATE TABLE snack_book(
    bid INT,
    sid INT,
    qty INT,
    PRIMARY KEY (bid, sid),
    FOREIGN KEY (bid) REFERENCES booking(bid) ON DELETE CASCADE,
    FOREIGN KEY (sid) REFERENCES snacks(sid) ON DELETE CASCADE
);
```

- Snack_book table is used to store the details of snack booked by user.
- Combination of sid and bid is the primary key.
- Foreign key bid and sid is referencing to booking and snacks table.

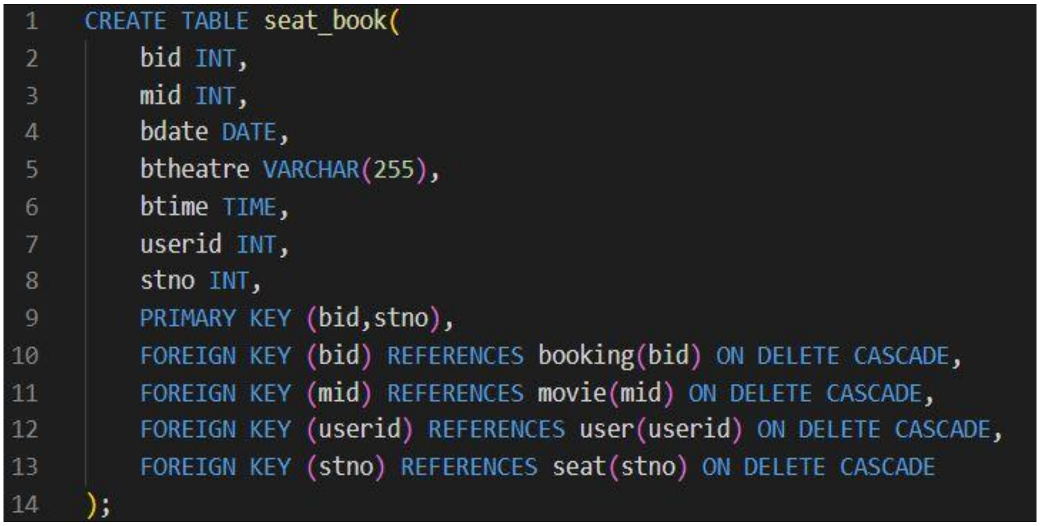
5. user table



```
CREATE TABLE user(
    userid INT PRIMARY KEY,
    email VARCHAR(255),
    password VARCHAR(255),
    name VARCHAR(255),
    ph_no INT
);
```

- User table is used to store the infoormation of all the users who login to the website.
- User table is also used to autheneticate the user login details.
- User_id is the primary key.

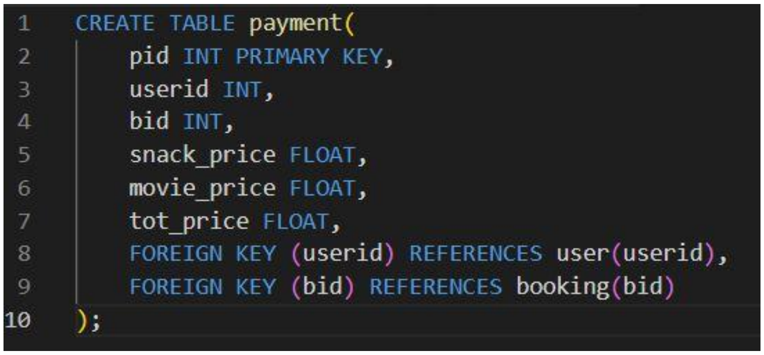
6. seat_book table



```
CREATE TABLE seat_book(
    bid INT,
    mid INT,
    bdate DATE,
    btheatre VARCHAR(255),
    btime TIME,
    userid INT,
    stno INT,
    PRIMARY KEY (bid,stno),
    FOREIGN KEY (bid) REFERENCES booking(bid) ON DELETE CASCADE,
    FOREIGN KEY (mid) REFERENCES movie(mid) ON DELETE CASCADE,
    FOREIGN KEY (userid) REFERENCES user(userid) ON DELETE CASCADE,
    FOREIGN KEY (stno) REFERENCES seat(stno) ON DELETE CASCADE
);
```

- Seat book table is used to store the seat booking details of a user.
- It is populated when a user selects a movie and show timing, date ,theatre.
- Primary key is a combination of bid and seat_no
- Foreign keys bid, mid, user_id, stno are used to refer to booking, movie, user and seat table respectively.

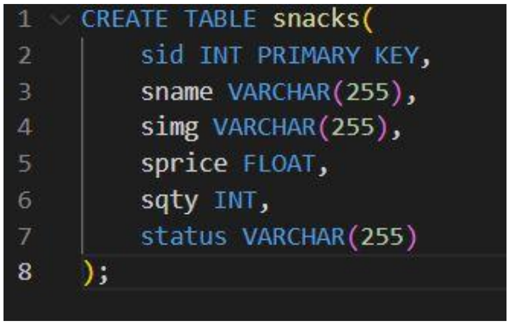
7. *Payment Table*



```
CREATE TABLE payment(
    pid INT PRIMARY KEY,
    userid INT,
    bid INT,
    snack_price FLOAT,
    movie_price FLOAT,
    tot_price FLOAT,
    FOREIGN KEY (userid) REFERENCES user(userid),
    FOREIGN KEY (bid) REFERENCES booking(bid)
);
```

- Payment table is used to store the payment details of the user .
- It stores the snack price ,movie ticket price and the total price for each user after completing of payment.

8. *Snacks Table*



```
CREATE TABLE snacks(
    sid INT PRIMARY KEY,
    sname VARCHAR(255),
    simg VARCHAR(255),
    sprice FLOAT,
    sqty INT,
    status VARCHAR(255)
);
```

- Snacks table is used to store the details of all the snacks available .
- It is a static database table only to be updated by the admin.
- Sid is the primary key

SQL queries Used in PHP

1. **Showing Movie:** To Show movies which are currently available and deleted by Admin.

```
$query="select * from movie where status='show'";
```

2. **Check Email And Password:** To authenticate admin.

```
$select = " select * from user where email = '$email' and password =
'$pass' ";
```

3. **New User:** New user insertion through register page.

```
 $insert = "insert into  user(email,password,Name,Ph_no)
VALUES('$email','$pass','$name','$phno')";
```

4. **Booking:** Booing details insertion .

```
 $q2="insert into booking(userid,mid,btime,bdate,btheatre)
      values('$userid','$mid','$time','$date','$theatre')";
```

5. **Block Already Reserved Seat:**

```
  $sql1 = "UPDATE seat set status='not_available' where   Stno='$st'";

  $sql2 = "update seat set bid='{$_SESSION['Bid']}' where   Stno='$st'";

 $sql3 = "insert into  seat_book(Bid ,Mid, Bdate,  Btheatre,  Btime, Userid,
Stno)
```

```
values('{$_SESSION['Bid']}','{$_SESSION['Mid']}','{$_SESSION['date']}','{$_SE
SSION['theatre']}','{$_SESSION['time']}','{$_SESSION['Userid']}','$st')";
```

6. **Book Snack:** New insertion of snacks to respective booking id.

```
 $q2="insert into snack_book values('$bid','$id','$i')";
```

7. **Payment Inserting & Finding All Prices:**

```
$sql1="insert into payment(Userid,Bid)

$sql2="update payment SET snack_price=(SELECT SUM(S.Sprice *
SB.qty) FROM Snacks S JOIN Snack_book SB ON S.Sid = SB.sid WHERE
SB.Bid=payment.Bid);";

 $sql3="update payment SET Movie_price=(SELECT SUM(M.Mprice *
{$_SESSION['nos']}) FROM booking B JOIN movie M ON B.Mid = M.Mid
WHERE B.Bid=payment.Bid);";

$sql4="update payment SET tot_price=Movie_price+snack_price where
pid={$_SESSION['Pid']};";
```

8. **Cancel Ticket: Queries used to perform deletion of data when user cancels ticket and to update the seat status .**

```
$sql1="delete from payment where Bid='$bid'";
$sql2="delete from snack_book where Bid='$bid'";
```

```
$sql3="select * from seat_book where Bid='$bid'";

$sql6="update seat set status='available',bid=NULL where Stno='$st'";

$sql4="delete from seat_book where Bid='$bid'";

$sql5="delete from booking where Bid='$bid'";
```

9. Admin:

- New Movie & Snack:

```
 $insert = "insert into movie
VALUES($Mid+1,'$name','$image','$link','$desc','$duration','$category'
,'$lang','$price','$date','show')";
$insert = "insert into snacks
VALUES($Sid+1,'$name','$image','$price','$qty','use')";
```

- Delete Movie & Snack:

```
$query="update movie set status='block' where Mname='$delmov'";
$query="update snacks set status='block' where Sname='$delsnk'";
```

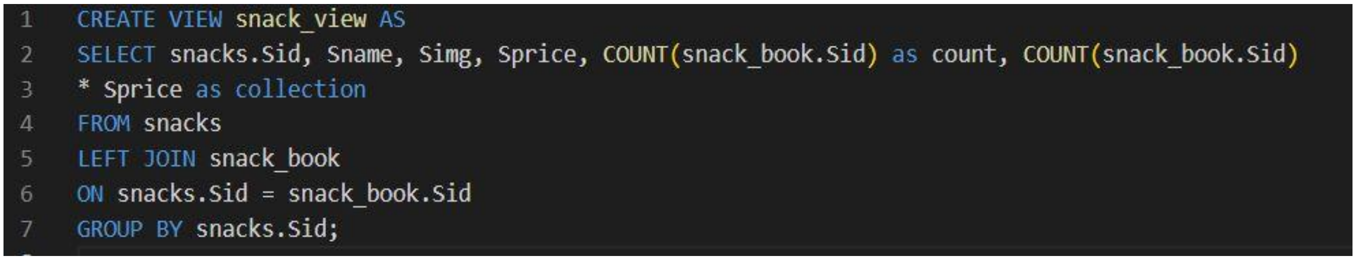
10. Trigger(delimeter):

```
CREATE TRIGGER `not_payed` AFTER INSERT ON `payment`
 FOR EACH ROW BEGIN
    DELETE FROM booking WHERE bid NOT IN (SELECT bid FROM Payment);
    DELETE FROM seat_book WHERE bid NOT IN (SELECT bid FROM Payment);
    DELETE FROM snack_book WHERE bid NOT IN (SELECT bid FROM Payment);
END
```

VIEWS

In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database. You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.

It does not store data itself, but instead retrieves data from one or more tables when it is queried. Views can be used to simplify complex queries, restrict access to certain columns of a table, or present data in a specific format.

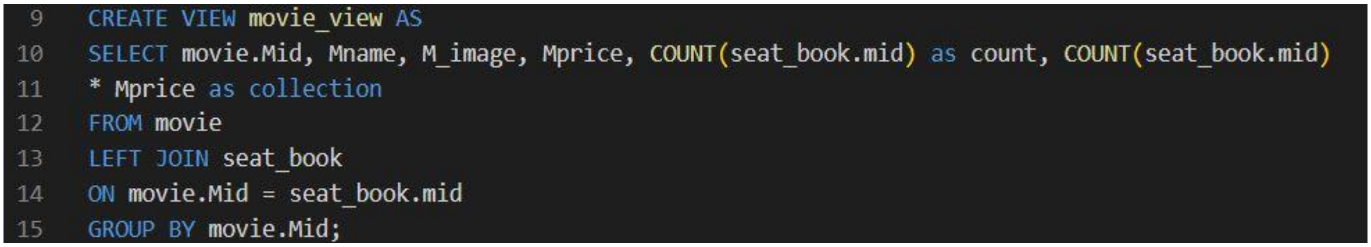


```sql
CREATE VIEW snack_view AS
SELECT snacks.Sid, Sname, Simg, Sprice, COUNT(snack_book.Sid) as count, COUNT(snack_book.Sid)
* Sprice as collection
FROM snacks
LEFT JOIN snack_book
ON snacks.Sid = snack_book.Sid
GROUP BY snacks.Sid;
```

This view creates a virtual table extracting attributes from snacks and snack book table

Showing the following details:

- Name of snack
- Image of snack
- Price of snack
- Total count of snack
- Total cost of snack



```sql
CREATE VIEW movie_view AS
SELECT movie.Mid, Mname, M_image, Mprice, COUNT(seat_book.mid) as count, COUNT(seat_book.mid)
* Mprice as collection
FROM movie
LEFT JOIN seat_book
ON movie.Mid = seat_book.mid
GROUP BY movie.Mid;
```

This view creates a virtual table extracting attributes from movie and seat book table

Showing the following details:

- Name of movie
- Image of movie
- Price of movie
- Total number of movies booked
- Collection of movie

TRIGGERS:

A trigger is a named database object that is associated with a table, and that activates when a particular event occurs for the table.

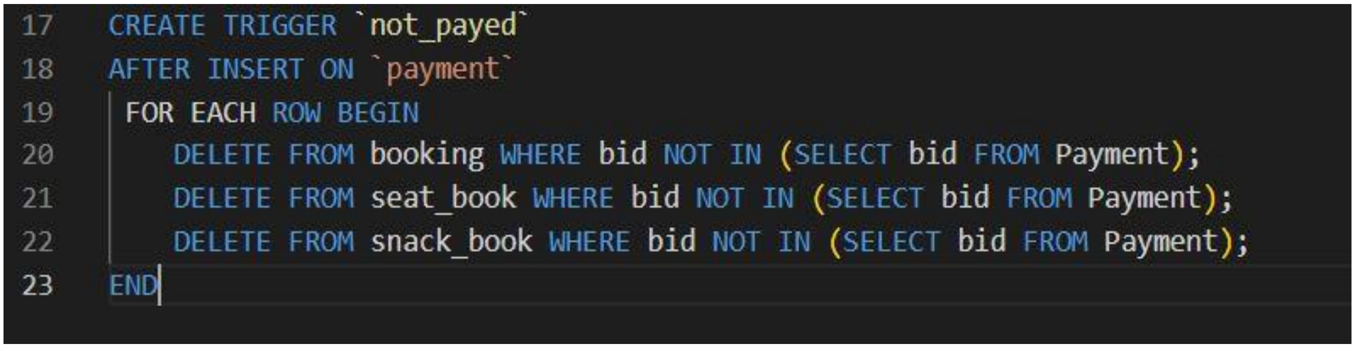
A trigger is defined to activate when a statement inserts, updates, or deletes rows in the associated table. These row operations are trigger events. For example, rows can be inserted by INSERT statement, and an insert trigger activates for each inserted row. A trigger can be set to activate either before or after the trigger event. For example, you can have a trigger activate before each row that is inserted into a table or after each row that is updated.

MySQL triggers activate only for changes made to tables by SQL statements. Triggers do not activate for changes to tables made by APIs that do not transmit SQL statements to the MySQL Server.

Trigger syntax:

```
DELIMITER //
CREATE
TRIGGER trigger_name
trigger_time trigger_event
ON tbl_name FOR EACH ROW
[trigger_order] trigger_body DELIMITER ;
trigger_time: { BEFORE | AFTER }
trigger_event: { INSERT | UPDATE | DELETE }
trigger_order: { FOLLOWS | PRECEDES }
```

1) Trigger `not_payed’

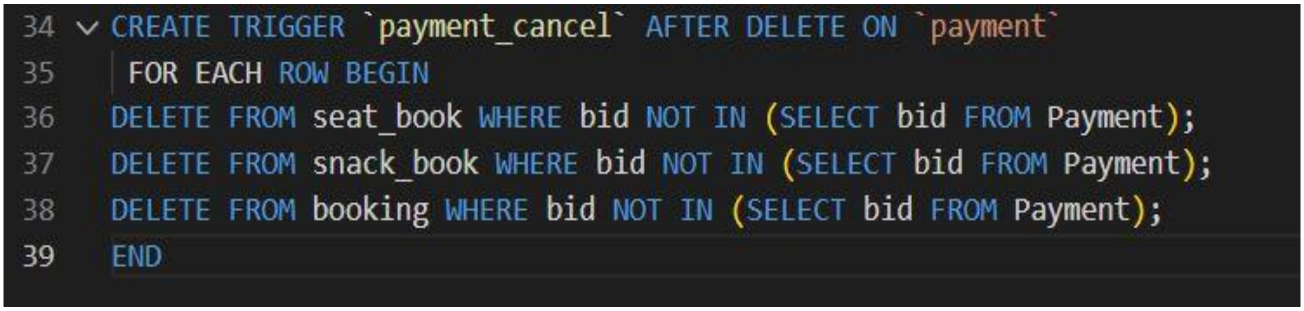


```sql
CREATE TRIGGER `not_payed`
AFTER INSERT ON `payment`
 FOR EACH ROW BEGIN
    DELETE FROM booking WHERE bid NOT IN (SELECT bid FROM Payment);
    DELETE FROM seat_book WHERE bid NOT IN (SELECT bid FROM Payment);
    DELETE FROM snack_book WHERE bid NOT IN (SELECT bid FROM Payment);
END
```

- ✓ This trigger `not payed` is activated when there is an insertion into payment tables.
- ✓ This trigger is used to delete the premature entry of data into booking, seat-book and snack-book tables.
- ✓ For example:

  When a user books a movie, Data gets registered in booking table even before payment. Similarly, user upon selecting seats and snacks, the data gets registered into respective tables.
- ✓ With the help of this trigger we are deleting these enteries if the user has not completed his payment and exits prematurely.

2) **Trigger `payment_cancel`**



```sql
CREATE TRIGGER `payment_cancel` AFTER DELETE ON `payment`
 FOR EACH ROW BEGIN
DELETE FROM seat_book WHERE bid NOT IN (SELECT bid FROM Payment);
DELETE FROM snack_book WHERE bid NOT IN (SELECT bid FROM Payment);
DELETE FROM booking WHERE bid NOT IN (SELECT bid FROM Payment);
END
```

- ✓ The trigger ‘”payment-cancel” is activated after a tuple is deleted from payment table (i.e. when the user cancels the payment).
- ✓ It deletes all the corresponding enteries of the booking id deleted in the seat_book,
- ✓ Snack_book, and booking table.

AGGREGATE FUNCTIONS:

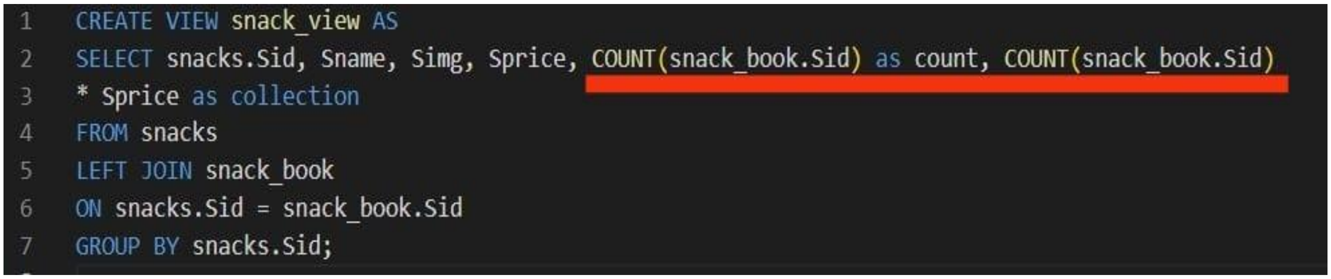
SQL aggregation function is used to perform calculations on multiple rows of a single column of a table. It returns a single value.

Aggregate functions are a type of SQL function that operate on a set of values and return a single value. They are commonly used in combination with the GROUP BY clause in a SELECT statement to perform calculations on subsets of data. Some examples of aggregate functions include:

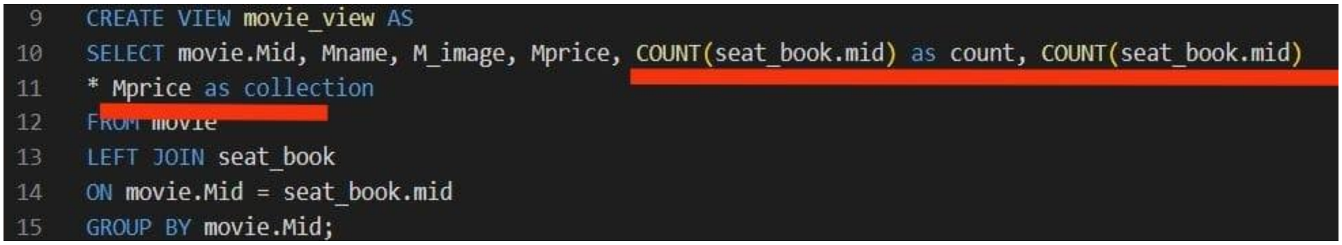
COUNT(expression): returns the number of rows that match a specified condition
SUM(expression): returns the sum of all values in a specified column or expression
AVG(expression): returns the average of all values in a specified column or expression
MIN(expression): returns the minimum value in a specified column or expression
MAX(expression): returns the maximum value in a specified column or expression

COUNT FUNCTION:

The COUNT function is used to Count the number of rows in a database table



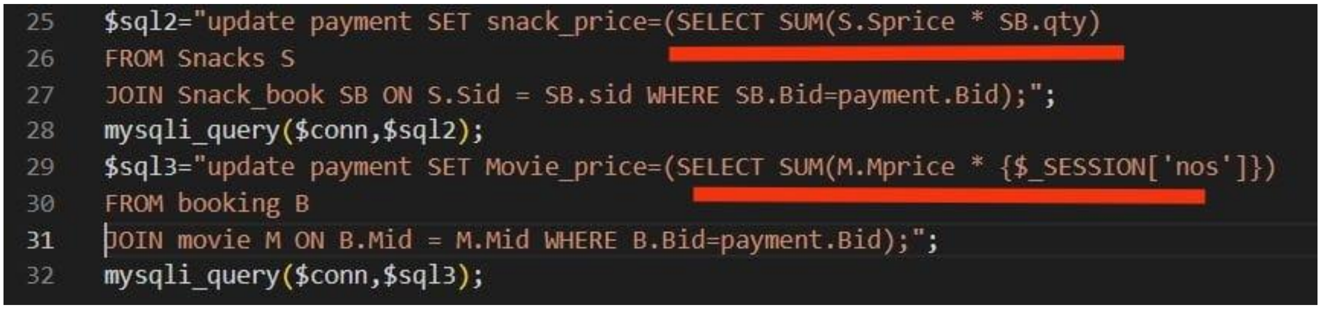
```
CREATE VIEW snack_view AS
SELECT snacks.Sid, Sname, Simg, Sprice, COUNT(snack_book.Sid) as count, COUNT(snack_book.Sid)
* Sprice as collection
FROM snacks
LEFT JOIN snack_book
ON snacks.Sid = snack_book.Sid
GROUP BY snacks.Sid;
```



```
CREATE VIEW movie_view AS
SELECT movie.Mid, Mname, M_image, Mprice, COUNT(seat_book.mid) as count, COUNT(seat_book.mid)
* Mprice as collection
FROM movie
LEFT JOIN seat_book
ON movie.Mid = seat_book.mid
GROUP BY movie.Mid;
```

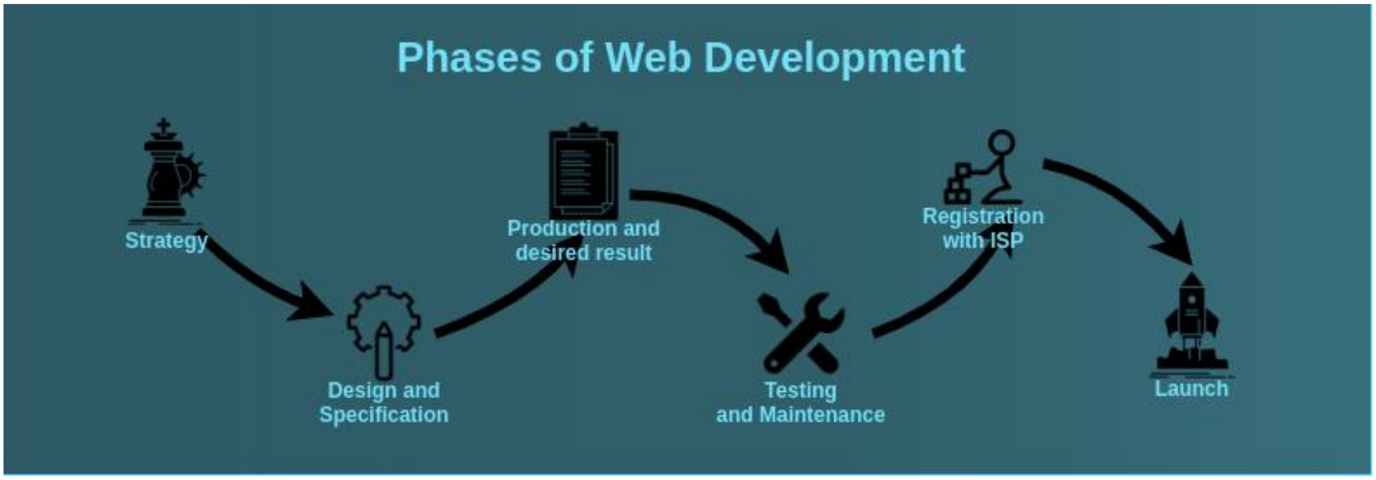
SUM FUNCTION:

Sum function is used to calculate the sum of all selected columns. It works on numeric fields only.



```
$sql2="update payment SET snack_price=(SELECT SUM(S.Sprice * SB.qty)
FROM Snacks S
JOIN Snack_book SB ON S.Sid = SB.sid WHERE SB.Bid=payment.Bid);";
mysqli_query($conn,$sql2);
$sql3="update payment SET Movie_price=(SELECT SUM(M.Mprice * {$_SESSION['nos']})
FROM booking B
JOIN movie M ON B.Mid = M.Mid WHERE B.Bid=payment.Bid);";
mysqli_query($conn,$sql3);
```

WORKING AND IMPLEMENTATION



Creation Of Database:

Here, MySQL database is deployed. MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL). It is used for storing and managing data in a relational database management system (RDMS). It is a standard language for Relational Database Systems. It enables a user to create, read, update and delete relational databases and tables. SQL allows users to query the database in several ways, using English-like statements.

Phases Of Web Development: Web Development refers to a term that includes all the processes involved in developing a web project or website. It contains the various phases such as planning, designing, testing, and launching of the web project. The web development process requires a team of experts responsible for implementing the different tasks needed to create a website.

Strategy: The first step in the web development process for a developer is to make a strategy for developing a web page or web site. In the strategy phase, web developer has to done the following:

- Deciding goals and objectives
- Developing team
- Make the appropriate analysis associated with problem and review the
- analysis
- Formulate a list of tasks
- Proposal of project to web team for developing.

Design and Specification: After the strategy-making, the next step in the web development process is to develop a planned work. Web developer has to determine the schedule and the specifications. The tasks in this phase are as follows:

- Developing approach
- Planning of contents needed for use
- Making of rough design for project
- Making of final design from rough design, if there are no considerable
- modification in rough design.
- Frame a prototype of project for developing
- Test the prototype
- If prototype is accomplished, then go to next phase phase-3 otherwise
- repeat the phase 2 until prototype is accomplished.

Production of desired result: In this phase of the web development process, the actual functional site is built. After the proper testing of the prototype, the developer has to work on developing the actual live web project. The actual live web project is built according to the requirements of the client. Web developer has to consider all the situations from the design phase to create all the features in the web project.

This phase involves both front end development and back-end development of the website. Front end development comprises of the writing codes with the basic technologies like HTML, CSS, etc. according to the web standards. It generally starts by developing the home page first and then other pages. Backend development is also completed in this phase by installing and configuring the content management systems, databases, and frameworks. After completing all the steps that were finalized in the strategy and design phase by which the original website becomes functional, it is tested in the next phase.

Testing and Maintenance: Testing is an important phase in the web development process. Testing is performed by the developers and testers to ensure the client's requirements after completion of the web project. In this phase, quality assurance and browser compatibility issues of the website are checked. Testers test all the developed features and assure the validity of the written code. Various types of testing such as integration testing, regression testing, functional testing, smoke testing, load testing, and performance testing are performed in this phase by both testing and development teams. Testing can be performed manually or automatically on the basis of the type of testing and web projects.

.

Registration with ISP: After completion of the Testing and Maintenance and removing all the bugs from the project, the next step or phase is to register the web project with the regional ISP to make the web project legal. The web project is delivered to the client after uploading the website to a server. File transfer protocol (FTP) is used to host the website to a hosting server. The client has to select and decide the ISP which provides domain name registration and web hosting services. After setup of these accounts and registering with the ISP so that the web project gets an accurate domain space at the ISP server.

Launch: This is the last phase of the web development process. Project is launched after getting registered with ISP. after launching, web project is publicly accessed by the users of the particular domain. The tasks performed in the launch phase are as follows.

- Migration of data.
- Launching of server.
- Merging of code

NORMAL FORMS OF TABLES

A database schema is said to be in **3rd Normal Form (3NF)** if it meets the following criteria:

- It is in First Normal Form (1NF), which means that all data is atomic (i.e. not repeating groups of data) and each table has a primary key.
- It is in Second Normal Form (2NF), which means that all non-primary key columns are dependent on the primary key.
- There are no transitive dependencies between non-primary key columns.

All the tables here have a primary key and all the non-primary key columns are dependent on the primary key. In addition, there is no transitive dependency present.

1. **For example**, in the "seat" table, the "stno" is the primary key and "bid" is the foreign key, which references the "bid" column in the "booking" table. The "status" column is dependent on the "stno" primary key, and there is no transitive dependency between "bid" and "status".

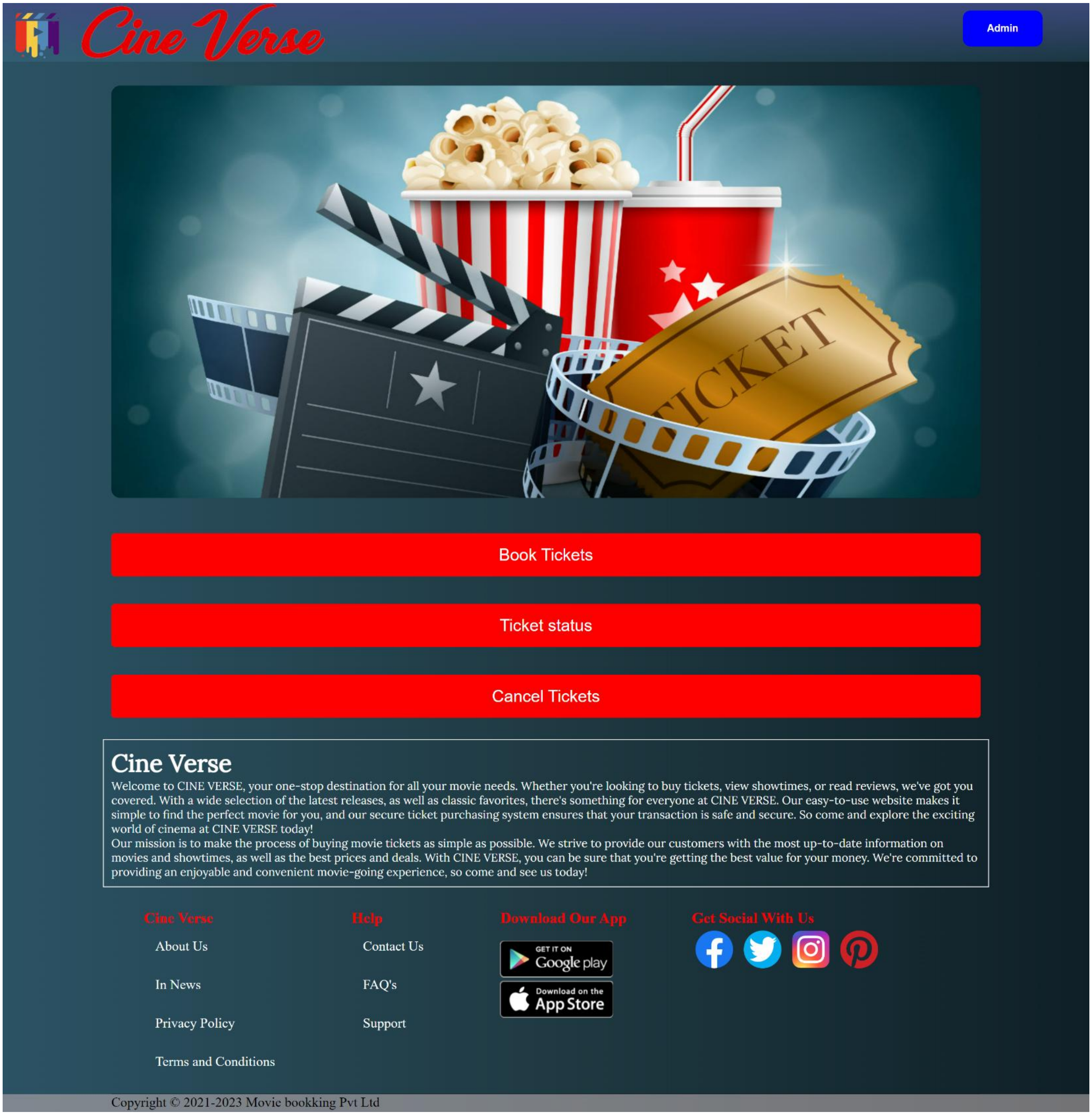
2. **For example**, in the "booking" table, the "bid" column is the primary key, and the "userid" and "mid" columns are foreign keys that reference the "userid" and "mid" columns in the "user" and "movie" tables respectively. The "btime", "bdate", and "btheatre" columns are dependent on the "bid" primary key and there is no transitive dependency between them.

3. **Similarly**, in the "snack_book" table, the "bid" and "sid" columns are foreign keys that reference the "bid" column in the "booking" table and "sid" column in "snacks" table respectively, and the "qty" column is dependent on the composite primary key "bid" and "sid" and there is no transitive dependency between them.

4. In short, the schema follows the best practices of database design, the tables are normalized and there is no data redundancy, which ensures the data integrity and consistency.

FRONT END IMPLEMENTATION

1. HOME PAGE



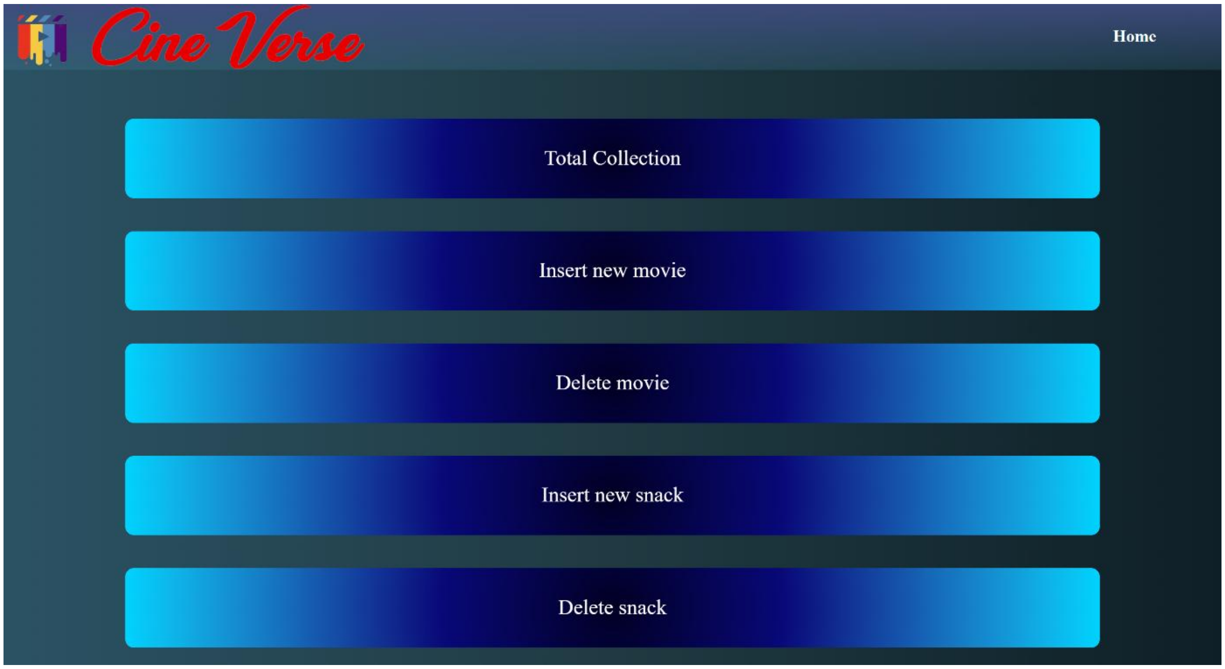
Cine Verse

Welcome to CINE VERSE, your one-stop destination for all your movie needs. Whether you're looking to buy tickets, view showtimes, or read reviews, we've got you covered. With a wide selection of the latest releases, as well as classic favorites, there's something for everyone at CINE VERSE. Our easy-to-use website makes it simple to find the perfect movie for you, and our secure ticket purchasing system ensures that your transaction is safe and secure. So come and explore the exciting world of cinema at CINE VERSE today!

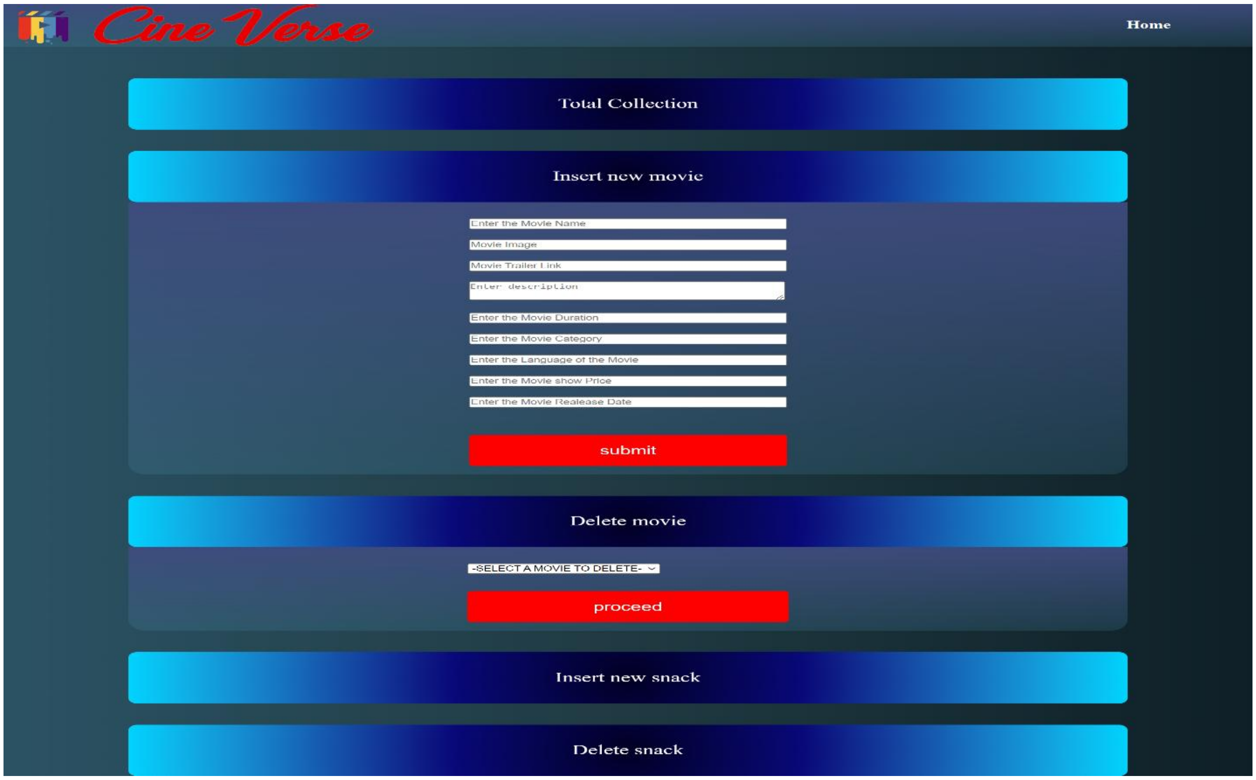
Our mission is to make the process of buying movie tickets as simple as possible. We strive to provide our customers with the most up-to-date information on movies and showtimes, as well as the best prices and deals. With CINE VERSE, you can be sure that you're getting the best value for your money. We're committed to providing an enjoyable and convenient movie-going experience, so come and see us today!

Copyright © 2021-2023 Movie bookking Pvt Ltd

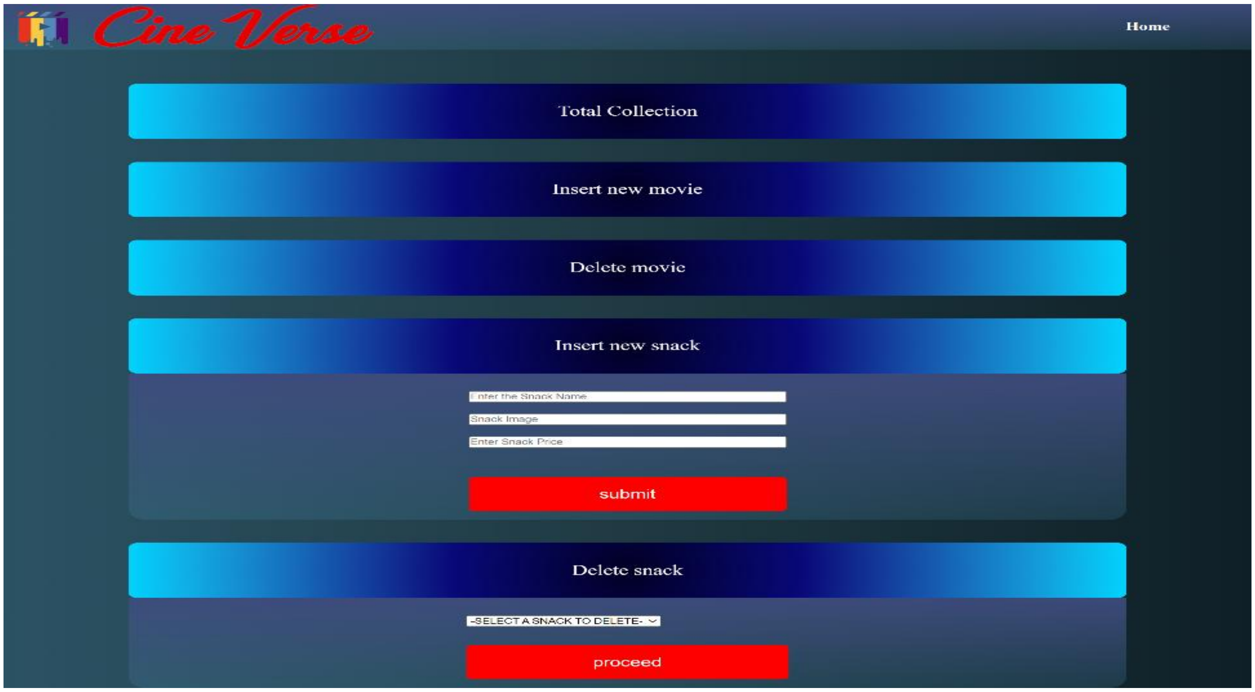
2. ADMIN PAGE



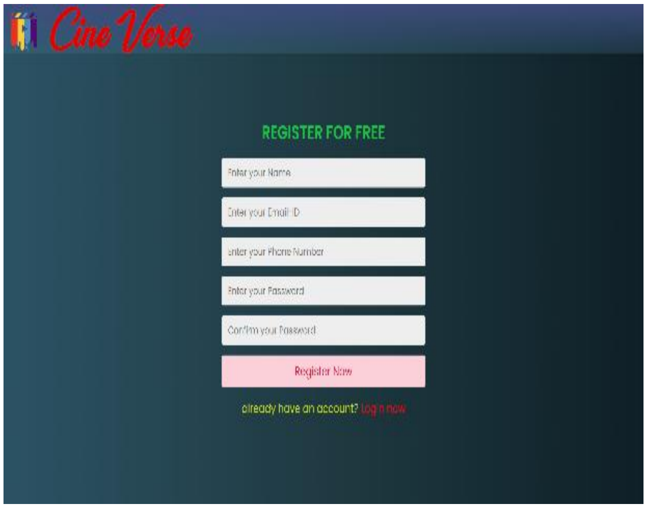
3. ADMIN MOVIE INSERTION & DELETION



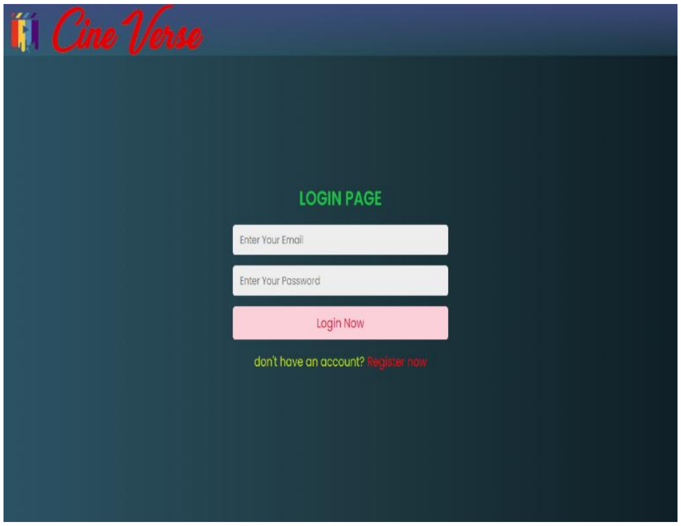
4. ADMIN SNACKS INSERTION & DELETION



5. USER REGISTRATION & LOGIN

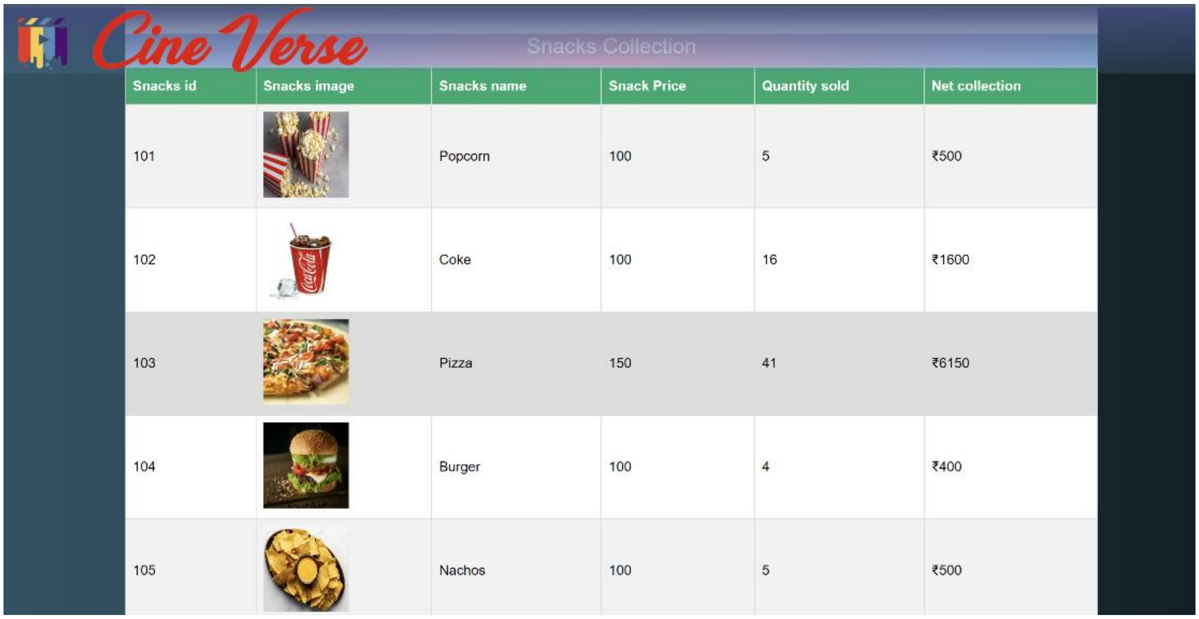


already have an account? Log in now



don't have an account? Register now

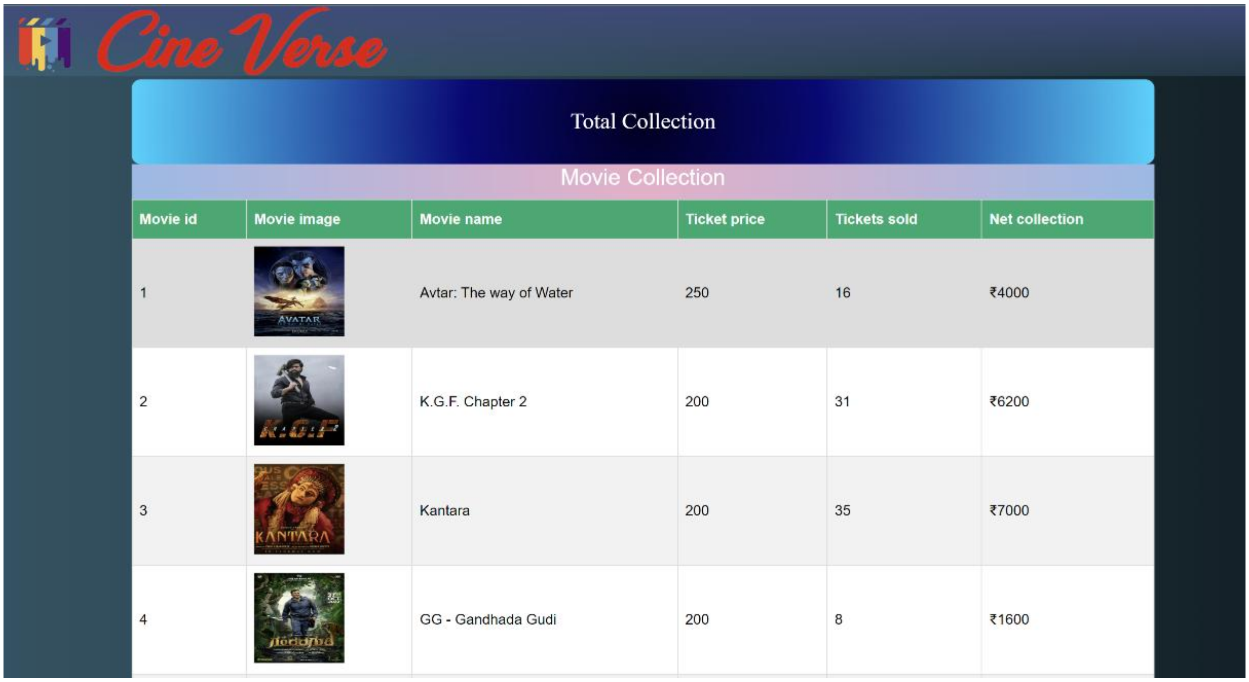
6. NET COLLECTION (SNACK)



Snacks Collection

Snacks id	Snacks image	Snacks name	Snack Price	Quantity sold	Net collection
101		Popcorn	100	5	₹500
102		Coke	100	16	₹1600
103		Pizza	150	41	₹6150
104		Burger	100	4	₹400
105		Nachos	100	5	₹500

7. NET COLLECTION (MOVIE)

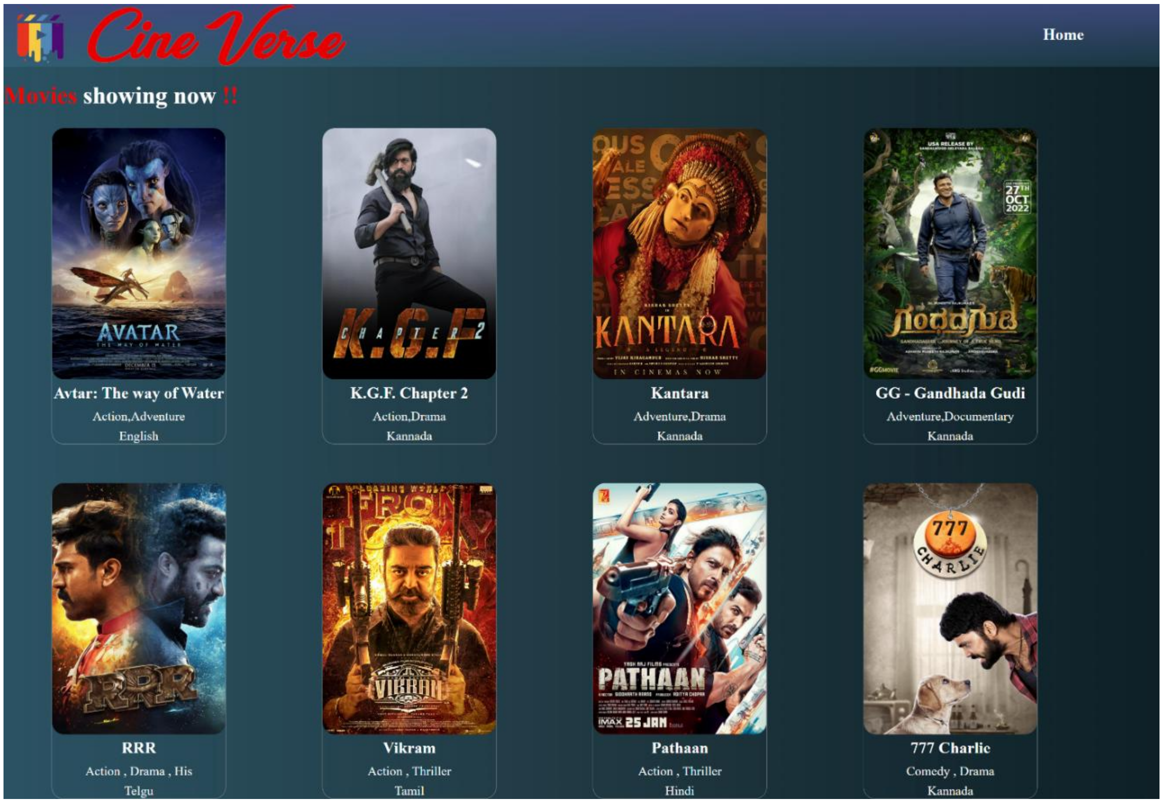


Total Collection

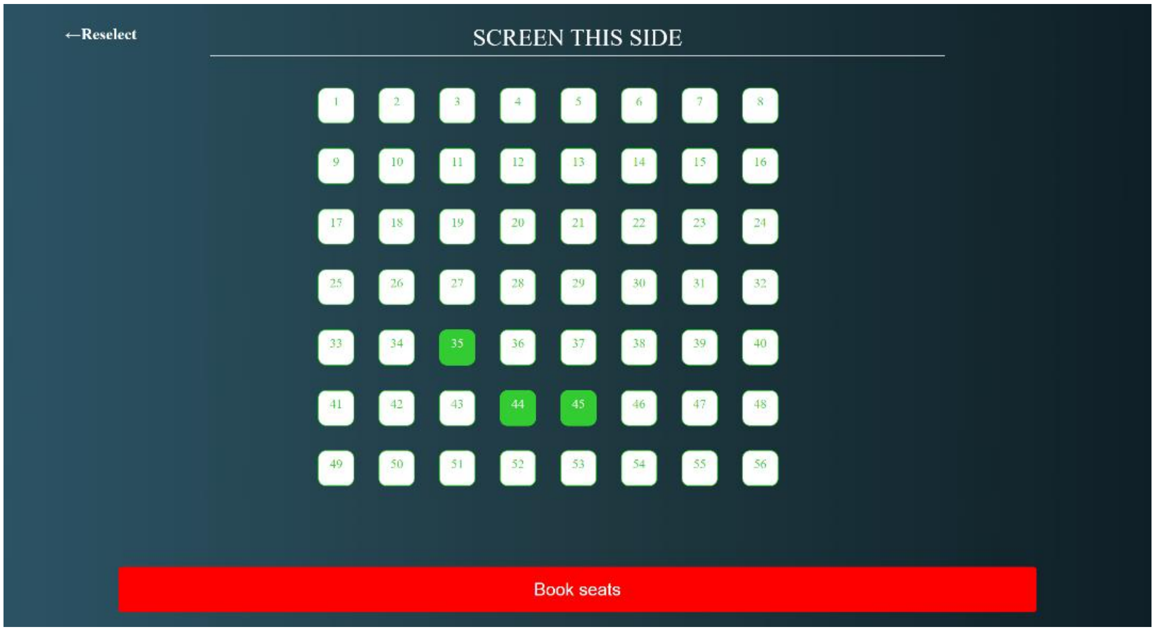
Movie Collection

Movie id	Movie image	Movie name	Ticket price	Tickets sold	Net collection
1		Avtar: The way of Water	250	16	₹4000
2		K.G.F. Chapter 2	200	31	₹6200
3		Kantara	200	35	₹7000
4		GG - Gandhada Gudi	200	8	₹1600

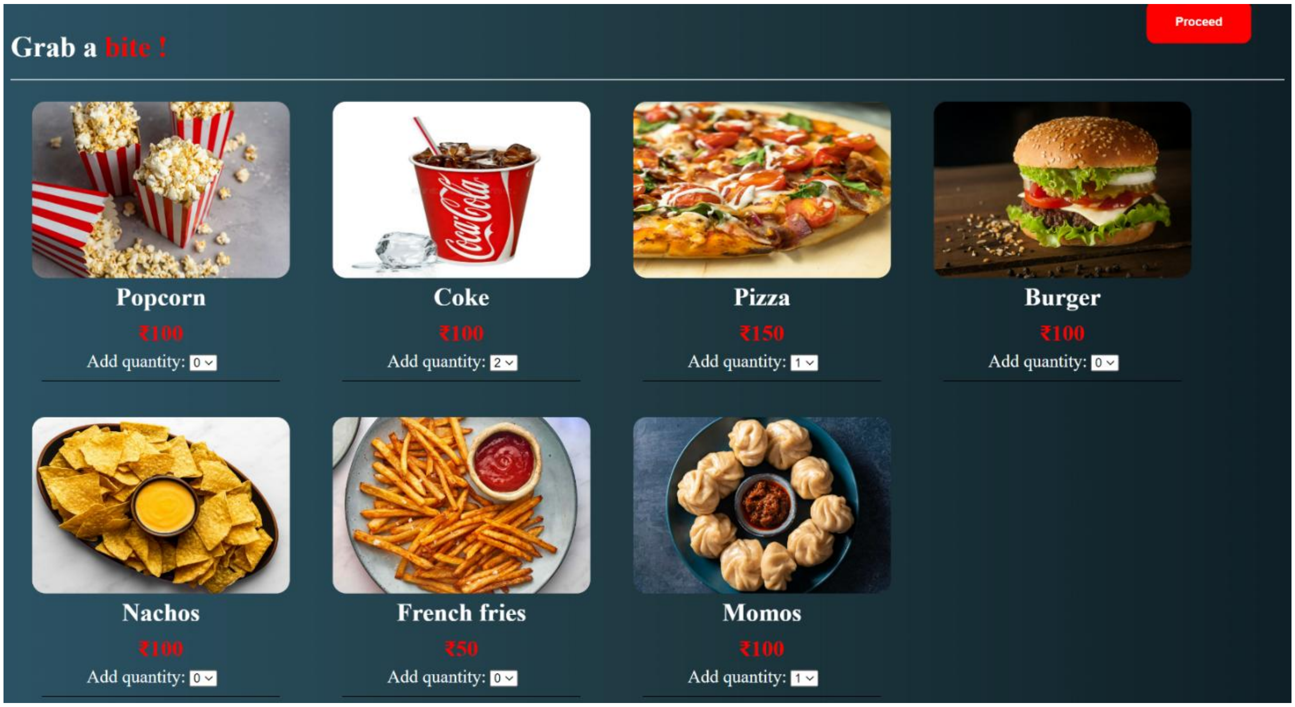
8. MOVIE BOOKING PAGE



9. SEAT SELECTION PAGE



10. SNACKS SELECTION PAGE



Grab a bite !

Popcorn
₹100
Add quantity: 0

Coke
₹100
Add quantity: 2

Pizza
₹150
Add quantity: 1

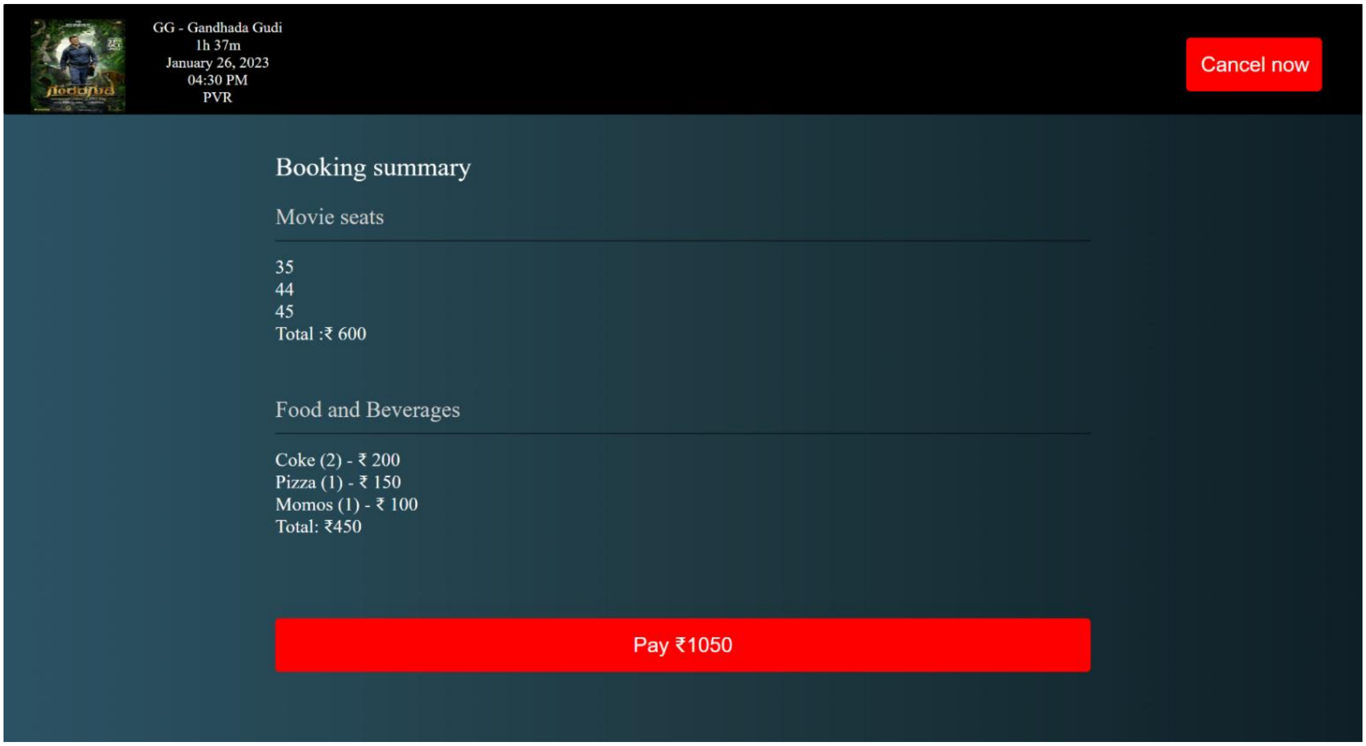
Burger
₹100
Add quantity: 0

Nachos
₹100
Add quantity: 0

French fries
₹50
Add quantity: 0

Momos
₹100
Add quantity: 1

11. BOOKING SUMMARY PAGE



GG - Gandhada Gudi
1h 37m
January 26, 2023
04:30 PM
PVR

Booking summary

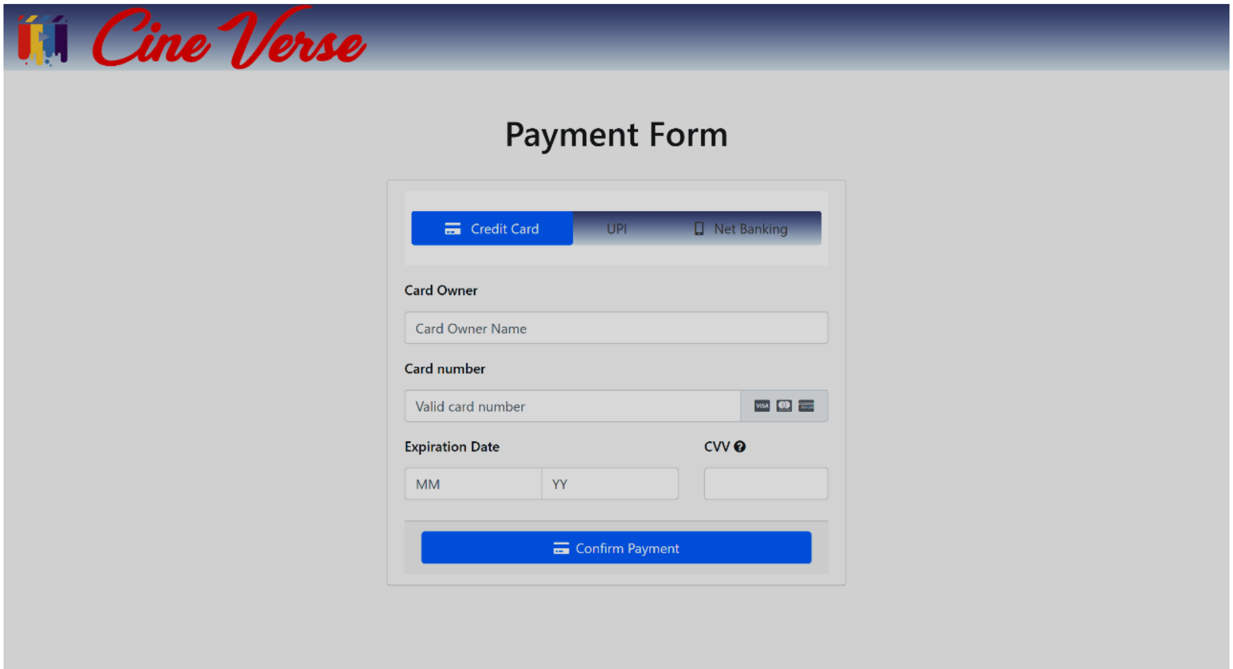
Movie seats

35
44
45
Total :₹ 600

Food and Beverages

Coke (2) - ₹ 200
Pizza (1) - ₹ 150
Momos (1) - ₹ 100
Total: ₹450

12. PAYMENT PAGE



Payment Form

Card Owner

Card Owner Name

Card number

Valid card number

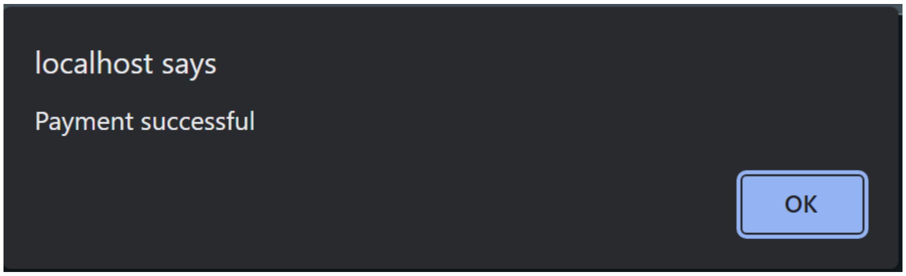
Expiration Date

MM

YY

CVV

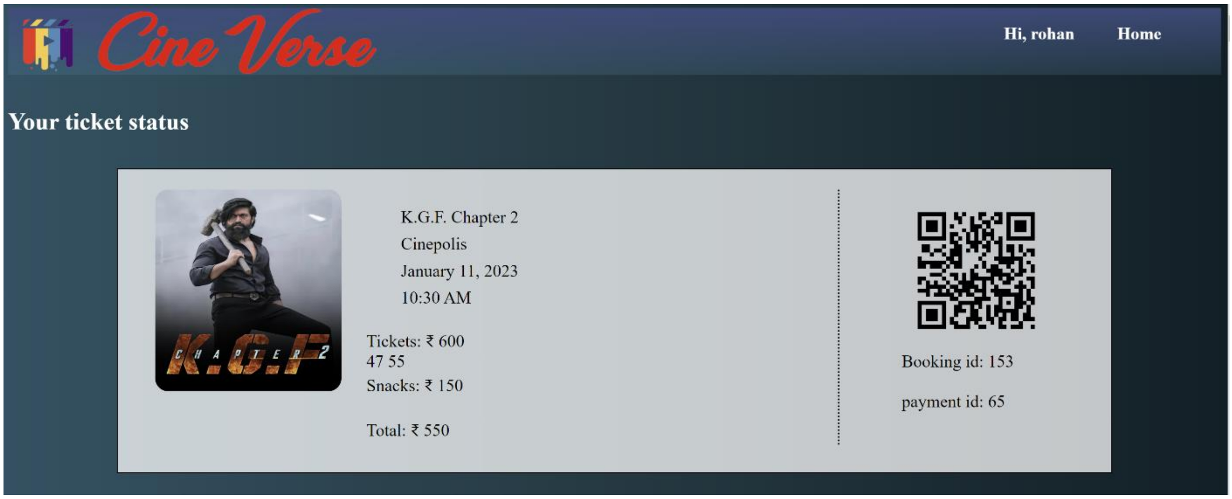
13. PAYMENT CONFIRMATION ALERT



localhost says

Payment successful

14. TICKET STATUS



Your ticket status

K.G.F. Chapter 2
Cinepolis
January 11, 2023
10:30 AM

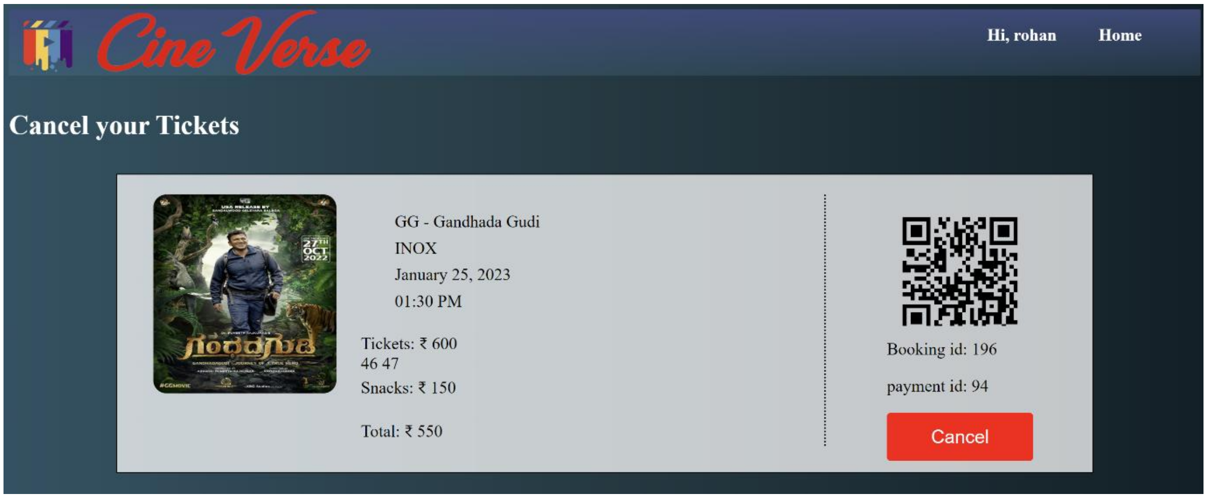
Tickets: ₹ 600
47 55
Snacks: ₹ 150

Total: ₹ 550

Booking id: 153

payment id: 65

15. CANCELATION OF TICKET



Cancel your Tickets

GG - Gandhada Gudi
INOX
January 25, 2023
01:30 PM

Tickets: ₹ 600
46 47
Snacks: ₹ 150

Total: ₹ 550

Booking id: 196

payment id: 94

CONCLUSION

CINE VERSE is a movie booking website that provides users with a convenient and efficient way to book movies and purchase snacks online.
The project consists of both frontend and backend components, with the frontend providing a user-friendly interface for users to interact with the website, and the backend managing and storing the data that is used by the frontend.

The backend of the project is implemented using a database schema that includes tables for storing information about seats, snacks, payments, users, bookings, and movies. The schema is designed to follow the best practices of database design and is in 3rd Normal Form, which ensures data integrity and consistency.

The frontend of the project includes several pages that allow users to interact with the data stored in the backend. The home page provides an overview of the website and the available movies. The admin login page allows an administrator to insert and delete snacks and movies. The movie selection page allows users to browse and select movies to watch. The seat selection page allows users to select seats for the selected movie. The snack selection page allows users to select snacks to purchase with their booking. The payment page allows users to make payments for their bookings. Users also have options to view the status of their tickets and to cancel their tickets.

In conclusion, CINE VERSE is a well-designed and implemented movie booking website that provides a convenient and efficient way for users to book movies and purchase snacks online. It is easy to use, secure and has the ability to track the status of the tickets and options to cancel the tickets. This type of booking system is beneficial for both movie goers and the theatre owners as it reduces the hassle of standing in queue to buy tickets and also increases the revenue for the theatre owners by providing online booking.

REFERENCES

- Fundamentals of Database Systems by Elmsari
- https://www.w3schools.com/html/default.asp
- https://www.w3schools.com/css/default.asp
- https://www.w3schools.com/js/default.asp
- https://www.php.net/docs.php
- https://www.youtube.com/