**COLLEGE OF APPLIED BUSINESS AND TECHNOLOGY**

**Chabahil, Gangahity, Kathmandu**

****

**System Analysis and Design (CSC 315)**

**Submitted by:**

Prajita Dhakal

B.Sc.CSIT 5th Semester

**Submitted to:**

Mr. Marshal Basnet

**Abstract**

The Online Movie Ticket Booking System is a dynamic, web-based application designed to streamline the process of purchasing cinema tickets. This system provides users with an efficient way to book tickets from the comfort of their homes, saving them the inconvenience of standing in long queues. Users must register or log in to access the system. The backend leverages PHP and JavaScript, while the front end is crafted with HTML and CSS for a seamless user experience. The application supports two distinct panels: an Admin Panel and a Customer/User Panel. Administrators can manage movies, cinemas, screens, and schedules, while users can browse movie options, view trailers, and make bookings with ease. Designed with a user-friendly interface, this system optimizes time and enhances the overall experience for both end-users and cinema operators.

**Table of Content**

1. Introduction
   1. Online Ticket Booking System
   2. Key Features
2. Objectives
3. Scopes
4. Data Flow Diagram
5. Use Case Diagram
6. Sequence Diagram
7. Functional/Non-Functional Requirements
8. Class Diagram
9. Software Management
10. System Testing & Data Analysis
11. Conclusion

# Introduction

### Online Ticket Booking System

This case study focuses on the online booking system, which enables customers to reserve movie tickets, check screening schedules, watch trailers, and read reviews. The system’s primary objective is to provide a hassle-free, internet-based solution for booking movie tickets, enhancing customer convenience and cinema management efficiency.

The growing demand for online services has been driven by people prioritizing their professional commitments over traditional recreational activities. Leisure time is often limited, and spending hours waiting in queues is undesirable. This system addresses that challenge by offering a web- based movie ticketing platform that is both intuitive and efficient.

### Key Features

* + 1. **User Registration and Login**
       - Secure registration and authentication process for customers.

### Movie Selection

* + - * Browse movies by genre, theater, and show times.

### Additional Features

* + - * Access to upcoming movie trailers, reviews, and payment gateway integration.

### Admin Dashboard

* + - * Full system control, including adding or removing movies, managing theaters and screens, and monitoring activity logs.

### Interactive and User-Friendly GUI

* + - * Designed for ease of navigation, ensuring an appealing and efficient user experience.

# Objectives:

1. To deliver a software solution for Online Ticket Booking, streamlining the booking process and enhancing efficiency.
2. To simplify and expedite the processes of ticket reservation, inquiry, and related services.

# Goals:

This project involves a detailed study to assess the feasibility of implementing an automated Online Ticket Booking System. It entails defining goals, specifying technological requirements, distinguishing between functional and non-functional needs, and conducting a comprehensive feasibility analysis. The primary objective is to create a solid framework for developing an efficient ticket booking system, with a focus on clearly defining the project scope to ensure the final product meets the specified requirements and delivers an optimal user experience.

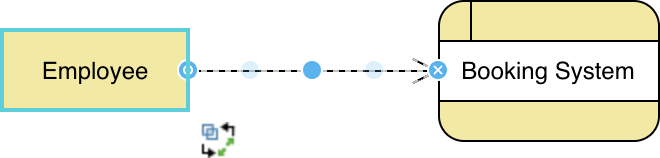
# Data Flow Diagram:

A Context Diagram is a diagram that defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it.

The context diagram for the online ticket booking system has following external entities that provide an overview of the system’s scope and boundaries:

1. Customer
2. Admin

The DFD level 0 diagram for online ticket booking system includes the following components:



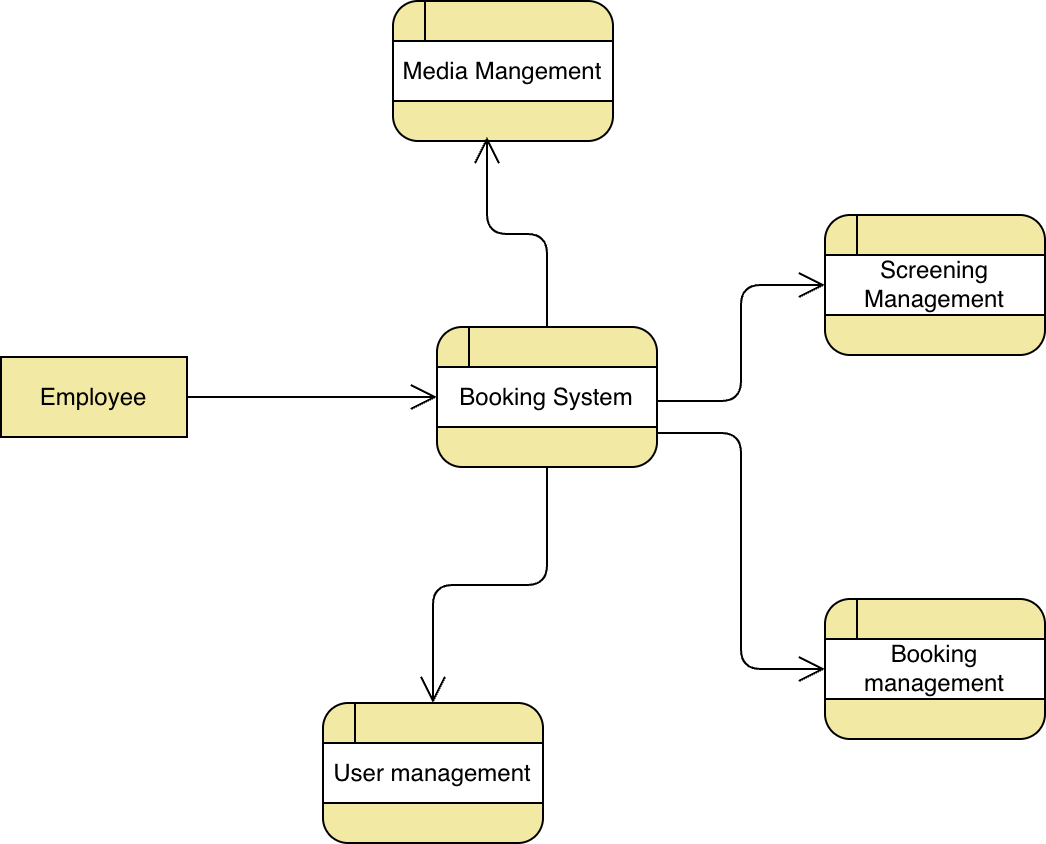
## DFD level 1:

DFD (data flow diagram) can be drawn to represent the system of different levels of abstraction.

The data flow diagram (DFD) for the ticket booking system up to level-1 shows the main functions and processes of the system, including data inputs, data processing, and data outputs.

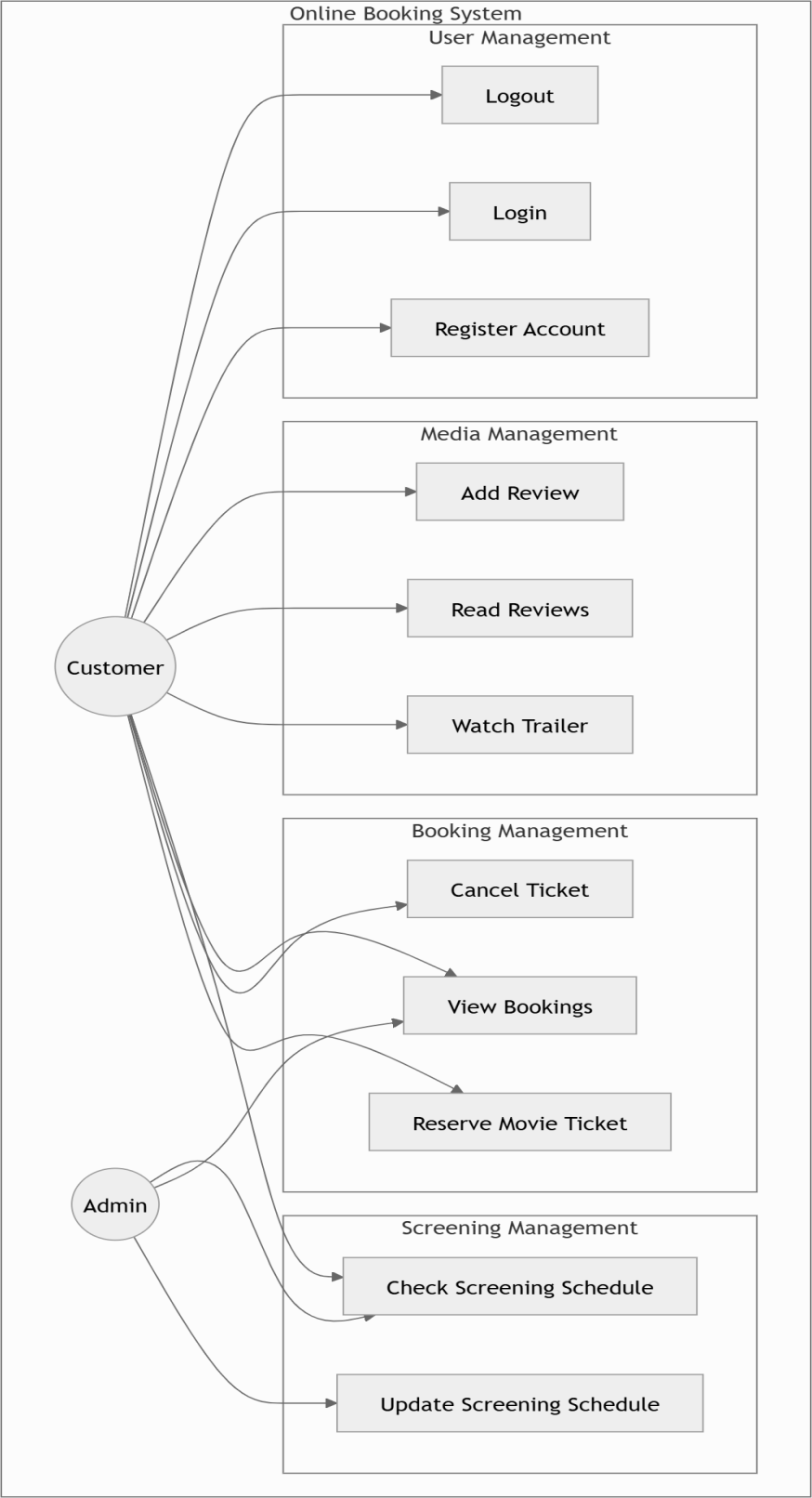
The DFD level 1 has the following data:

1. User Management System
2. Media Management System
3. Booking Management System
4. Screening Management System



# Use Case diagram:

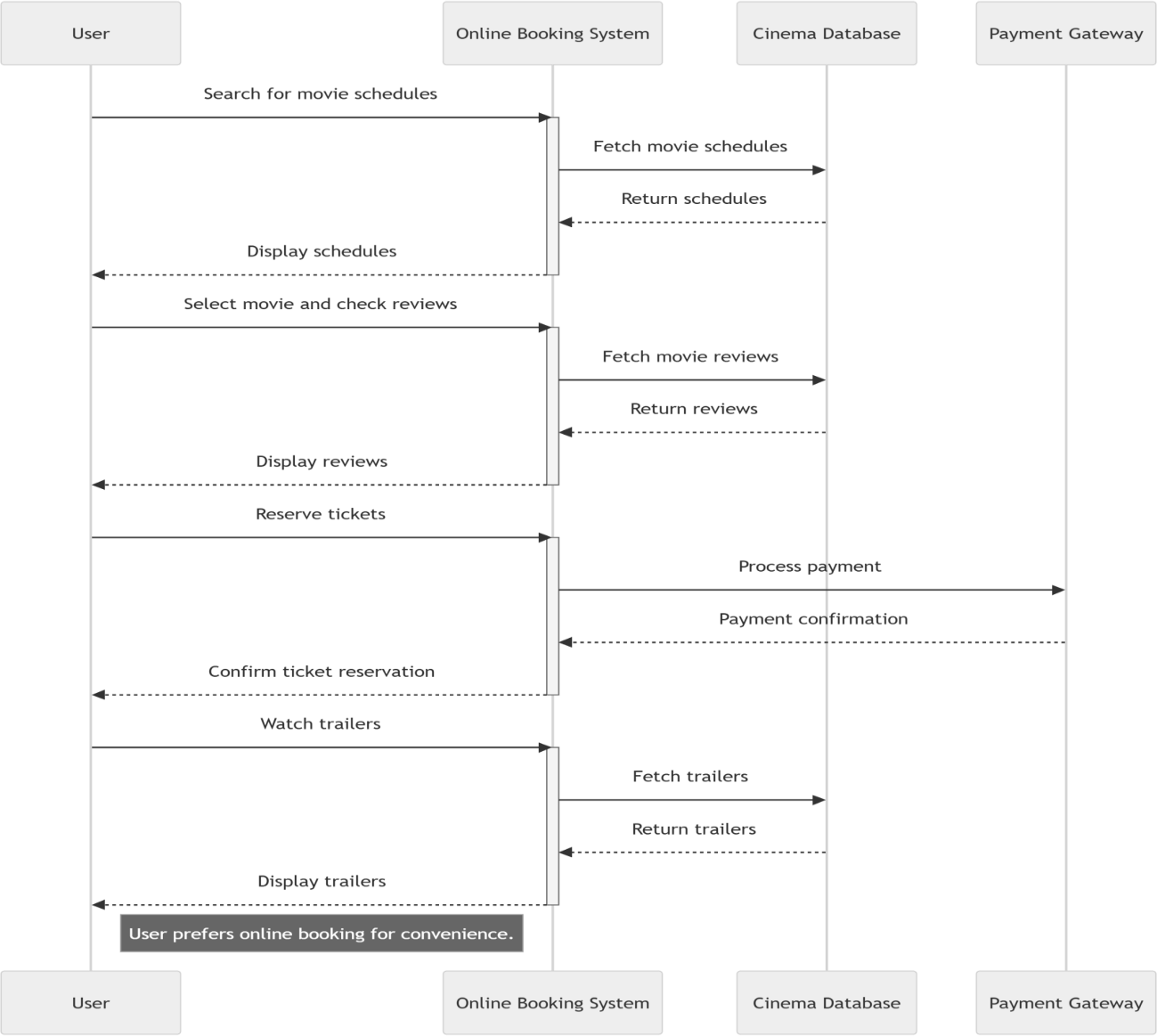
This **Use Case Diagram** represents the functionalities of the **Online Booking System** and its interaction with two primary actors: **Customer** and **Admin**. The diagram categorizes functionalities into modules, including **User Management**, **Media Management**, **Booking Management**, and **Screening Management**



* The Customer can perform actions like logging in, registering, reserving tickets, canceling bookings, adding reviews, and viewing movie details.
* The **Admin** manages screening schedules and has access to functionalities like updating schedules and overseeing bookings.

This diagram provides a high-level overview of how the system supports users and administrators in achieving their respective tasks.

# Sequence diagram:

****

This **Use Case Diagram** illustrates the interactions between the **Online Booking System** and its key actors: **Customer** and **Admin**. It showcases the system's main functionalities, such as user management, media management, booking management, and screening management, highlighting how each actor performs specific tasks like reserving tickets, adding reviews, and updating schedules. This diagram provides a clear overview of the system's features and the roles of its users.

# Functional/Non Functional Requirements:

### User Management:

* + Users must be able to register with their personal details, such as name, email, and password, to create an account.
  + The system should validate input fields to prevent errors and ensure data integrity.
  + Users should be able to view and edit their personal information, such as contact details and preferences.

### Movie Selection:

* + Users should be able to filter movies based on categories such as genre (e.g., action, drama), specific theaters, or available showtimes.
  + Enable users to view movie trailers directly within the system for better decision-making.
  + Provide an intuitive interface, such as dropdown filters or search options, to simplify the browsing process.

### Ticket Booking:

* + Users should be able to select a specific movie, theater, date, and showtime to reserve seats.
  + The system should show real-time seat availability and allow users to select their preferred seats.
  + Generate a booking confirmation with essential details like booking ID, date, time, and seat numbers.
* **Payment System:**
  + Allow users to pay for tickets securely through online payment methods like credit/debit cards, digital wallets, or net banking.
  + Provide transaction receipts and allow users to view payment history in their accounts.
* **Admin Dashboard:**
  + Administrators must have the ability to add new movies by providing details like title, genre, description, and trailer links.
  + Admins should be able to add or modify theater details, including location, number of screens, and capacity.
* **Safety and Security:**
  + The system should protect sensitive data, such as user credentials and payment details, using encryption techniques like AES or RSA.
  + Implement regular data backups to prevent loss of information during unexpected failures.
  + Admin and corporate dashboards should be accessible only to employees with valid credentials.

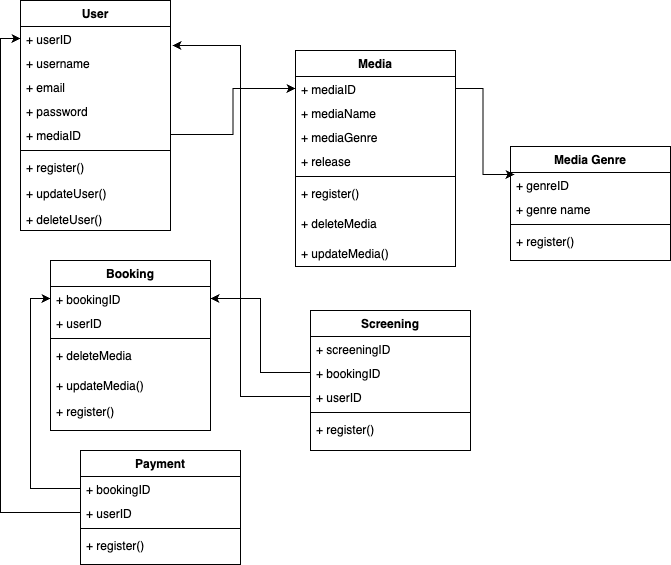
# Class Diagram

The class diagram represents the entities and their relationships in the booking system. The main class is User which has attributes such as name, phone, date of birth, email, and password, as well as a method to register, update and delete.

The Media class has attributes such as media id and methods to create a new media. The media class is used to represent all types of media or movies that the company offers for their customer.

The Booking system is where the users would book the media or show that are current in show. They process payments through the payment model and on success will have a booked show for the corresponding user.

Before booking, the user has to select the screen time and seats that are available to pick for that particular show.



# SOFTWARE MANAGEMENT

This section explains the system's front-end and back-end design in a clear and thorough manner. For front-end and back-end development process, a variety of tools has been used.

### Front-end design

**Captivating Users and Enhancing Experiences**

Front-end design is a crucial element in creating websites that impress and engage users. The design and content on the front end significantly influence how users perceive and interact with a website. Developing a front-end interface involves technologies like HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), JavaScript, and other tools.

* + - **HTML** serves as the foundation for building and structuring web pages. It provides the skeleton of the website, giving it organization and form.
    - **CSS** works alongside HTML to add styling, transforming plain structures into visually appealing and professional designs. Without CSS, a website would lack style and remain unformatted.
    - **JavaScript** adds interactive and dynamic features to websites, such as animations, transitions, and other engaging elements, making the site more functional and visually impressive.

Together, these technologies form the core components of front-end design, creating websites that are both advanced and aesthetically pleasing.

### Back-End Design

The back end of a web application comprises three essential components: the server, the application, and the database. When users submit information on the website—such as their name, theater name, number of seats, and other details—the web application processes this data and stores it in a database hosted on the server.

For efficient back-end functionality, the following core tools and components are utilized:

* + - **Sublime Text:** A versatile, cross-platform source code editor that supports various programming and markup languages. It also includes a Python API (Application Programming Interface), enabling developers to customize and extend its features.

These tools and technologies ensure smooth back-end operations, enabling data to be processed and stored effectively, which is crucial for the overall functionality of the web application.

### 9.3 Database Management

For managing the website’s database, we relied on phpMyAdmin, a free, open-source application for MySQL and MariaDB database management. This tool is user-friendly and simplifies the process of interacting with the database.

On the server side, we extensively used the PHP scripting language. PHP offers robust features, supports multiple database systems, and provides the complexity required to handle advanced operations seamlessly.

# System Testing and Data Analysis:

System testing and data analysis are critical steps in ensuring the functionality, reliability, and performance of an online ticket booking system. Here’s how we can approach system testing and data analysis for the online booking system.

We employed various methods to test and evaluate our website after its development. Initially, we applied the Black-Box Testing technique. Black-box testing, also referred to as behavioral testing, is a software testing approach where the tester has no knowledge of the internal structure, design, or implementation of the application being tested.

## Unit testing

Unit testing included the following things:

* + 1. User Authentication: We made sure that our system authenticated and authorized our users correctly such that no error or role mismatch ever occurred in the system.
    2. Payment: Testing for payment for buying tickets was rigorously to make sure each transaction was completed correctly

For unit testing we used PHP Unit which is a popular PHP unit testing framework widely used in the industry.

## Load Testing:

* + 1. We made sure that the product was scalable when faced with hugely concurrent users accessing the server at the same time.
    2. We made sure that the down time for the service was kept to the minimum For Load testing we used frameworks such as k6, Locust etc.

To enhance functionality, we used modules typically maintained under open-source licenses, allowing administrators to add new features easily. The code editor supports languages like PHP, JavaScript, HTML, and CSS, among others, making it versatile for web development.

For the back-end implementation, we utilized the XAMPP platform, which stands for Cross- Platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P). XAMPP is a lightweight Apache distribution designed for developers to set up local web servers for testing purposes. Its straightforward setup makes it an efficient choice for implementing projects.

## 9.2 Data Analysis

Booking Data:

* Number of Bookings per Screening: Analyze which screenings are most popular and identify peak times for ticket sales. This can help optimize screening schedules and manage capacity.
* Booking Patterns: Track when users typically book tickets (e.g., hours, days, or months leading up to a show) to improve marketing strategies.
* Booking Demographics: Analyze user data (location, age, etc.) to personalize offers and promotions.

Payment Data:

* Payment Methods: Analyze the most popular payment methods (credit card, PayPal, etc.) and monitor transaction success rates to ensure smooth payment processing.
* Revenue Trends: Monitor revenue trends over time to identify peak seasons or successful marketing campaigns.
* Refunds: Track the number of refunds, cancellations, or chargebacks to identify possible system errors or areas for improvement in the user experience.

User Data:

* User Retention: Analyze how many users return to book tickets after their first experience. Track conversion rates and identify factors that improve retention.
* User Feedback: Collect and analyze feedback (e.g., surveys or reviews) from users to identify pain points in the booking process or interface.
* Customer Support Interactions: Analyze the number and type of support tickets raised by users to identify frequent issues or bottlenecks in the system.

System Performance Data:

* Load Times: Track response times for different pages (e.g., booking, payment) and optimize slow pages to enhance user experience.
* Error Logs: Analyze error logs for failed transactions, system crashes, or bugs that affect the system’s functionality.
* Availability: Monitor uptime and downtime to ensure the system is reliably available for users, particularly during high-traffic periods.

Screening Data:

* Venue Utilization: Analyze how full different venues are during screenings and adjust future bookings and marketing efforts based on utilization rates.
* Screening Times: Track which screening times attract more customers and adjust scheduling accordingly.

# Conclusion:

The current approach to the online booking system focuses on providing a seamless and efficient user experience for booking tickets, managing screenings, and handling payments. This insight drives the design of a streamlined online booking platform that allows users to easily browse available screenings, select seats, make payments, and receive booking confirmations through an automated system. The automation of these processes ensures faster and more accurate bookings, reducing the chances of errors and improving overall customer satisfaction. With this system, users can effortlessly manage their bookings, making the entire ticketing experience more efficient and user-friendly.