**DOCUMENTATION OF MOVIE BOOKING SYSTEM PROJECT**

The **Online Movie Ticket Booking System** is an advanced web-based project developed using PHP and MySQL. It is designed to replicate the functionality of real-life online ticketing systems

**Key Features:**

* **User-Friendly Interface**: Ensures seamless interaction for users, similar to professional online ticketing platforms.
* **Comprehensive Booking System**: Allows users to search, select, and book tickets for movies with ease.
* **Real-Life Scenarios**: Accurately simulates real-world ticketing processes, from booking to payment confirmation.

This project is a perfect example of applying web development concepts to solve real-world problems. It provides a hands-on experience for students, enhancing their skills in web technologies, database management, and system design. The Online Movie Ticket Booking System is not only a practical academic project but also a step towards understanding the workings of professional online systems.

**TABLE OF CONTENT**

* Problem Definition
* Customer Requirement Specification
* Project Plan
* E-R Diagrams
* Project Folder and Files Structure
* Algorithms
* GUI Standards Document
* Interface Design Document

# Problem Definition

The **Online Movie Booking System** is a web-based platform designed to streamline the process of booking movie tickets. The system addresses the common challenges faced by moviegoers, offering a convenient, user-friendly solution to manage their movie plans efficiently.

### Key Objectives:

* **Advance Booking**: Users can book tickets in advance without the need to visit the theater.
* **Movie Information**: The portal provides comprehensive details about movies, including show timings, trailers, and user reviews.
* **Ease of Access**: Users can access the platform from the comfort of their homes, eliminating the need to physically visit theaters.
* **Seamless Registration**: By registering on the web portal, users can access all available movies, theaters, and additional features.

### Benefits:

1. **Convenience**: No need to wait in queues at the theater; users can book tickets anytime, anywhere.
2. **Comprehensive Movie Tracking**: Users can easily browse movies, view ratings, and trailers, and check show times and ticket prices.
3. **Class Selection**: The system offers two seat categories for booking:
   * Standard
   * Vip

The **Online Movie Ticket Booking System**, developed in PHP, is designed to deliver a seamless, efficient experience for moviegoers. With its intuitive interface and robust features, the system is a modern solution to traditional ticket booking challenges.

# Customer Requirement Specifications

## Overview

The **Online Movie Booking System** aims to provide an efficient, user-friendly platform for moviegoers to book tickets, view showtimes, watch trailers, and access movie reviews. The system is designed for two primary user types: **Visitors/Registered Users** and **Admins**, each with specific roles and functionalities.

## Features

1. **User Registration and Login**
   * Enables users to create accounts and log in securely.
2. **Showtime Viewing**
   * Users can view show times by selecting a preferred date.
3. **Movie Information**
   * Detailed reviews and ratings available for each movie.
   * Trailers to help users decide on a movie.
4. **Ticket Booking**
   * Users can book tickets by selecting a movie, show timing, and seat category.
5. **Seat Categories**
   * Vip
   * Standard
6. **Special Concessions**
   * Discounted ticket prices for children aged 3 to 12 years.
7. **Payment Gateway**
   * Not Integrated
8. **Admin Dashboard**
   * Centralized platform for managing theaters, movies, and shows.
9. **Theater Management**
   * Admin can add, delete, or edit theaters.
10. **Movie Management**
    * Admin can add, delete, or edit movies for each theater.
11. **Show Management**
    * Admin can manage show timings and pricing for all seat categories.

The **Online Movie Booking System** ensures a smooth and enjoyable experience for users while providing admins with all necessary tools to maintain and enhance the platform.

# Project Plan

This plan outlines the development of the **Online Movie Booking System** within one month using HTML, CSS, JavaScript, jQuery, AJAX, Tailwind CSS, Chart.js, and Video.js.

## Week 1: Setup and Design

* **Requirement Analysis**: Finalize features and database schema.
* **Environment Setup**: Install tools, configure Tailwind CSS, Chart.js, and Video.js.
* **Design**: Create wireframes and static pages for Home, Login/Registration, Movie Details, and Admin Dashboard.
* **Database**: Design and create tables for users, movies, theaters, bookings, and reviews.

## Week 2: User Interface

* **User Pages**: Develop responsive layouts with Tailwind CSS, integrate Video.js for trailers, and add jQuery/AJAX for interactivity.
* **Admin Dashboard**: Build dashboard with Chart.js for user stats, booking trends, and revenue analysis.
* **Authentication**: Implement secure login and registration for users and admins.

## Week 3: Backend Integration

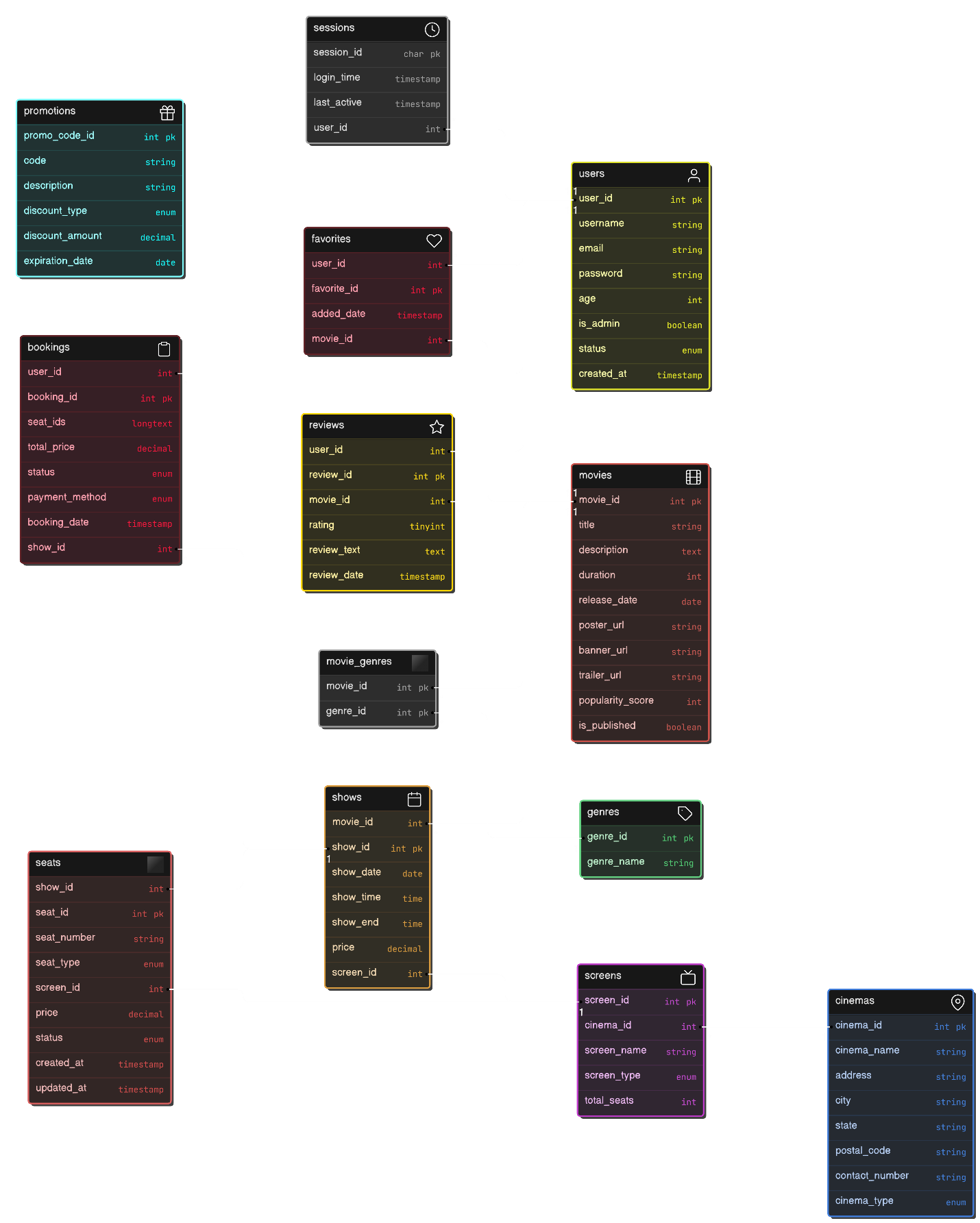
* **Development**: Create PHP scripts for user, movie, and booking management. Use AJAX for asynchronous updates.
* **Admin Functions**: Implement CRUD operations for theaters, movies, and show timings.

## Week 4: Testing and Deployment

* **Testing**: Perform unit and integration testing, ensure responsiveness and browser compatibility.
* **Feedback**: Fix bugs and optimize performance.
* **Deployment**: Launch system on a server and provide setup documentation.

## Deliverables

* Fully functional system with responsive user/admin interfaces.
* Admin dashboard with Chart.js and integrated Video.js trailers.
* Complete database and CRUD functionality.
* Comprehensive project documentation.

s

# E-R Diagram of Database

# Project Folder Structure

### 1. ****admin**** (Backend management)

This folder contains the logic for the administration panel of the booking system. It has controllers for handling various actions (fetching, deleting, updating), views for displaying admin pages, and includes like headers.

* **controllers**: PHP files to manage various operations like adding, deleting, fetching, and updating data related to movies, cinemas, bookings, etc.
  + Example: fetch\_movies.php, delete\_movie.php, update\_movie.php
* **images**: Contains images used in the admin panel, like theatre.jpg.
* **includes**: Files like header.php which are included in pages to provide common structure (e.g., header).
* **bookings.php**: Manage user bookings.
* **cinemas.php**: Manage cinema-related operations.
* **movies.php**: Admin page to manage movie data.
* **genres.php**: Manage movie genres.
* **users.php**: Manage user data.
* **index.php**: Likely the entry point for the admin panel.
* **logout.php**: Log out from the admin panel.

### 2. ****assets**** (Static resources)

This folder contains images used in the system.

* **images**: Includes various images for movies, banners, and logos, e.g., avengers.jpg, banner.webp, logo.png, hollywood.jfif.

### 3. ****controllers**** (User-side interactions)

These PHP files handle the user-facing logic for the movie booking system, like booking seats, fetching movie data, user authentication, and payment processing.

* **add\_to\_favorites.php**: Add movies to the user's favorites.
* **book\_seats.php**: Handle seat booking logic.
* **delete\_booking.php**: Delete a user's booking.
* **fetch\_bookings.php**: Fetch a user's bookings.
* **fetch\_movies.php**: Fetch movie data.
* **login.controller.php**: User login handling.
* **logout.php**: User logout.
* **payment.php**: Payment processing logic.
* **register.controller.php**: Handle user registration.
* **remove\_favorite.php**: Remove a movie from favorites.

### 4. ****frameworks**** (External libraries)

This folder contains third-party libraries used in the project.

* **jquery-3.7.1.min.js**: The minified version of the popular jQuery library for handling JavaScript interactions.

### 5. ****includes**** (Common HTML structure)

Files for parts of the website that are reused across pages.

* **footer.php**: Contains the footer part of the website.
* **header.php**: Contains the header part of the website.

### 6. ****library**** (External resources)

This folder holds libraries that provide additional functionality for the site.

* **videojs**: Contains files to integrate Video.js, a video player library.
  + video-js.css: Styling for the video player.
  + video.min.js: Minified script for the video player.

### 7. ****views**** (Frontend views)

The views folder holds the templates or pages that the user interacts with directly. It contains various PHP files for displaying pages like the booking form, login, movie details, etc.

* **uploads**: Folder to handle user uploads (empty here).
* **bookings.php**: Displays user bookings.
* **book\_seat.php**: Displays the seat booking interface.
* **favorites.php**: Shows user's favorite movies.
* **login.php**: Login page for users.
* **movie.php**: Displays details about a single movie.
* **movies.php**: Lists all available movies.
* **payment.php**: Displays payment interface.
* **register.php**: Registration page for new users.
* **video-js.css**: CSS for styling the video player on the frontend.

### 8. ****Additional Files****

* **database\_structure.txt**: Contains the structure of the database, likely describing tables and relationships.
* **index.css**: Contains global CSS styles for the website.
* **index.php**: Likely the entry point for the public-facing website.

# Algorithms

# GUI Standard Document

# Interface Design Document