

A Field Project Report on
MOVIE TICKET BOOKING

Submitted

In partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE and ENGINEERING

By

Y.Sridhar	231FA04448
M.Vinay	231FA04556
B.Tejaswi	231FA04563
MD.Umar khan	231FA04B26

Under the Guidance of

Mr.G.Murali

Assistant Professor, CSE



VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF COMPUTING AND INFORMATICS

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University)

Vadlamudi, Guntur -522213, INDIA.

April, 2025



VIGNAN'S

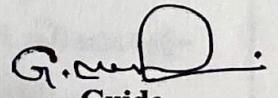
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

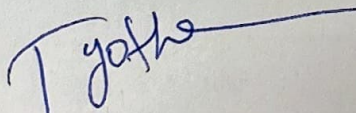
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

CERTIFICATE

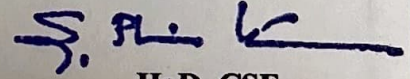
This is to certify that the field project entitled "MOVIE TICKET BOOKING" is being submitted by (Y.Sridhar & 231FA04448), (M.Vinay & 231FA04556), (B.Tejaswi & 231FA04563), and (MD.Umar Khan & 231FA04B26) in partial fulfilment of the requirements for the degree of **Bachelor of Technology (B.Tech.) in Computer Science and Engineering** at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India.

This is a bonafide work carried out by the aforementioned students under my guidance and supervision.


Guide



Project Review Committee


HoD, CSE

HoD
Dept. of Computer Science & Engineering
VFSTR Deemed to be University
VADLAMUDI - 522-213
Guntur Dist., A.P., India.



VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DECLARATION

Date: 25/4/25

We hereby declare that the work presented in the field project titled "MOVIE TICKET BOOKING" is the result of our own efforts and investigations.

This project is being submitted under the supervision of **Mr.G.Murali** , Assistant Professor,CSE in partial fulfillment of the requirements for the Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

Y.Sridhar (231FA04448)

Signature *Y. Sridhar*

M.Vinay (231FA04556)

Signature *M. Vinay*

B.Tejaswi (231FA04563)

Signature *B. Tejaswi*

MD.Umar khan (231FA04B26)

Signature *MD. Umar Khan*

TABLE OF CONTENTS

Chapter No.		Contents	Page No
1		Introduction	1-4
	1.1	Problem Definition	1
	1.2	Existing System	2
	1.3	Proposed System	2-3
	1.4	Literature Review	3-4
2		System Requirements	4-7
	2.1	Hardware & Software Requirements	5
	2.2	Software Requirements Specification(SRS)	6-7
3		System Design	7-9
	3.1	Modules of System	8
	3.2	UML Diagrams	9
4		Implementation	10-14
	4.1	Sample Code	11-13
	4.2	Test Cases	13-14
5		Results	14-19
	5.1	Output Screens	15-19
6		Conclusion	20
		References	20-21

1.Introduction:

The "fliX Booking Management System" is a dynamic and responsive web application designed to streamline the process of movie ticket booking. Created with a modern aesthetic and user experience in mind, the project combines HTML, CSS, and JavaScript to offer a seamless and interactive platform where users can explore movies, view trailers, and book their seats with ease.

Inspired by current entertainment platforms and tailored to user preferences, the system showcases a sleek dark theme with vibrant highlights, hover effects, and smooth animations that enhance usability. The front-end features a fixed header, automated slideshow, scrollable movie gallery, and an intuitive booking section. It also integrates touch-friendly navigation and responsive layouts, ensuring accessibility across devices.

Key functionalities include:

- A movie selection interface with detailed descriptions and trailer previews.
- A user-driven seat selection system with local storage support for persistence.
- Dynamic content loading based on user interaction.
- A newsletter subscription form to enhance user engagement.

This project reflects a strong focus on both functionality and visual design, creating an engaging environment for users to browse and book tickets in just a few clicks.

1.1 Problem Definition:

In the entertainment industry, especially cinema, the process of ticket booking has significantly transitioned from traditional box-office counters to online platforms. However, many current online systems still suffer from user interface limitations, lack of flexibility, delayed updates, and an overall poor user experience—especially for users on mobile or lower-end devices.

Key issues identified:

- **Cumbersome UI/UX:** Existing systems often have a cluttered design, overwhelming users with ads, popups, and irrelevant content.
- **Lack of Real-Time Interactivity:** Users cannot see real-time trailer previews or get interactive feedback while browsing or selecting seats.
- **Limited Personalization:** Few systems remember user preferences, leading to repetitive actions for frequent users.

- **Performance Issues on Mobile Devices:** Many platforms are not fully optimized for mobile screens, leading to frustrating navigation and booking errors.
- **Disjointed User Journey:** Users often have to leave the site to watch trailers or read movie details, breaking the engagement cycle.

1.2 Existing System Overview (Existing Table):

Here's a comparative table that outlines how traditional and popular booking systems function versus their limitations:

Feature	Existing Systems	Limitations
Movie Listings	Offered with posters and showtimes.	Often cluttered UI, not visually interactive
Seat Selection	Interactive seat maps available	Not visually appealing or customizable, lacks storage of preferences
Trailer Viewing	Requires redirect to YouTube or other platforms	Inconvenient, breaks user flow
Booking Confirmation	Available via email/SMS	No in-platform feedback or animation-based confirmations
Mobile Optimization	Some responsive designs	Inconsistent behavior across devices
UI Design Aesthetics	Varies, sometimes outdated	Lacks modern feel, insufficient use of transitions and user interaction cues
Performance on Low-end Devices	Often heavy and script-laden	Long load times, increased bounce rate
User Profile Integration	Login-based personalization	Not all features available to guest users; loss of preferences if not logged in

1.3 Proposed System:

The proposed system—**fliX Booking Management System**—aims to provide an engaging, responsive, and modern platform for moviegoers to:

- **Browse movies** with dynamic previews, genre tags, and rating info.
- **Watch trailers directly** without leaving the platform.

- **Select and book seats visually**, with seat status updates and user-friendly selection UI.
- **Get booking suggestions**, showtimes, and real-time data stored locally.
- **Interact with a mobile-first, responsive UI**, ensuring access across all devices.
- **Receive instant confirmation** and visual cues post-booking to improve user satisfaction.

Key Highlights:

- **Responsive Design:** Built using Flexbox and Grid layout, the UI adjusts seamlessly to various screen sizes and orientations.
- **Dark Mode Aesthetic:** The platform uses a dark theme with pastel accent colors, improving readability and modern appeal.
- **Local Storage Integration:** Seat selection and movie preferences are stored locally, improving repeat user experience without requiring login.
- **Interactive Components:** Hover animations, dynamic descriptions, auto-sliders, and embedded JavaScript functions create an engaging and fluid browsing experience.
- **Accessibility:** Keyboard navigation, screen reader-friendly labels, and simplified forms enhance inclusivity.

1.4 Literature Review:

A strong foundation for this system stems from studies and analyses in web usability, UI/UX best practices, and interactive design principles. Below are some insights from existing literature and trends:

User-Centered Design Principles:

Research by Jakob Nielsen and others has shown that websites focused on **usability and interaction** see higher user retention. By minimizing cognitive load and improving visual hierarchy, users find tasks like booking or browsing movies more intuitive.

Interactive Web Applications:

Modern web apps emphasize interactivity without needing page reloads. Studies highlight the importance of **asynchronous behavior**, use of **local storage**, and **client-side rendering** to improve responsiveness—principles applied throughout the fliX system.

Responsive Design Techniques:

According to W3C and modern frontend development practices, users expect websites to perform consistently across devices. Our use of **media queries**, **flexible containers**, and **dynamic resizing** ensures this consistency.

Engagement Through Visual Media:

Articles from platforms like Smashing Magazine and UX Planet advocate for **embedded media previews** (like trailers) to improve engagement. This system integrates such previews directly into movie cards for immediate access.

Gamification and Visual Feedback:

Visual cues like hover effects, button transitions, and confirmation modals increase perceived interactivity. Research supports that **micro-interactions** and **feedback animations** improve user trust and clarity in booking systems.

2. System Requirements:

The fliX Booking Management System is a user-friendly web application that facilitates online movie ticket booking. It includes an engaging interface featuring a slideshow of promotional banners, a responsive movie catalog, and an intuitive seat booking system. Users can browse movie details, watch trailers, and select seats based on preferred showtimes. The system uses local storage to save selections temporarily and updates ticket counts and prices dynamically. Built using standard web technologies (HTML, CSS, JavaScript), it is compatible with all modern browsers and offers a clean, dark-themed UI. With its modular design and interactive elements, the system is both scalable and maintainable, making it suitable for small-scale movie booking needs without requiring backend integration.

Functional Requirements:

- Display homepage with navigation links (Home, Movies, About, Contact Us).
- Showcase a slideshow of featured movies using radio-button-based navigation.
- List available movies with posters, descriptions, genres, and trailer links.
- Allow users to:
 - Select a movie and showtime via dropdown menus.
 - Proceed to book seats from a visual seating layout.
- Dynamically calculate and display the total ticket cost based on selected seats.
- Store movie selection and seat data locally using localStorage.
- Enable users to watch trailers via external YouTube links.
- Provide a "Subscribe to newsletter" form for user engagement.
- Display static contact information and quick navigation links in the footer.

Non-Functional Requirements:

- Smooth, responsive UI using Flexbox and CSS transitions.
- Dark theme with accent colors for modern visual appeal.
- Mobile and desktop browser compatibility using HTML5, CSS3, and JavaScript.
- Local storage ensures client-side data persistence without backend support.
- Modular and maintainable code for ease of future updates.
- Lightweight and fast loading with minimal external dependencies (e.g., Google Fonts).

2.1 Hardware & Software Requirements:

Hardware Requirements

Minimum:

- **Processor:** Intel Core i3 or equivalent
- **RAM:** 4 GB
- **Hard Disk:** 250 MB free space
- **Display:** 1280x720 resolution
- **Input Devices:** Keyboard and mouse
- **Internet:** Required for loading YouTube trailers and external fonts

Recommended:

- **Processor:** Intel Core i5 or higher
- **RAM:** 8 GB
- **Hard Disk:** 1 GB free space
- **Display:** Full HD (1920x1080) or higher

Software Requirements

Front-End:

- **Languages:** HTML5, CSS3, JavaScript
- **Libraries/Tools:**
 - Google Fonts API
 - Web Storage API (localStorage)

- **Browser Support:**
 - Google Chrome (latest)
 - Mozilla Firefox
 - Microsoft Edge
 - Safari (latest)
- **Code Editor:** VS Code, Sublime Text, or any web IDE

Server/Hosting (Optional for deployment):

- **Web Server:** Apache/Nginx or static hosting via GitHub Pages, Netlify, or Firebase
- **OS Compatibility:** Windows, macOS, or Linux (since it runs in-browser)

2.2 Software Requirements Specifications:

The *fliX Booking Management System* is a web-based application developed to simplify the process of browsing, selecting, and booking movie tickets online. It allows users to explore a variety of movies, view trailers, select their preferred cinema and show timings, and reserve seats through a graphical interface. The primary objective is to provide an intuitive and responsive user experience without the need for backend integration. The application is built entirely using front-end technologies such as HTML5, CSS3, and JavaScript, with interactivity powered by event-driven JavaScript and browser storage (localStorage) for saving temporary data like selected seats and movie choices.

The system is designed for use on modern web browsers and supports features like responsive layout, smooth scrolling, dynamic updates, and interactive UI components. It includes modules for movie listing, trailer viewing, ticket booking, seat selection, and a footer section with contact and subscription options. All data processing is done on the client-side, making it ideal for prototype demos or front-end portfolios. The user interface employs a dark theme with modern design elements and is mobile-friendly. While it doesn't store data on a server, it provides a realistic simulation of an online ticket booking process.

Functional Requirements:

- Dynamic movie list and trailer links
- Show selection using dropdown menus
- Seat booking interface
- Seat and movie selection storage using localStorage
- Newsletter subscription form

Non-Functional Requirements:

- Responsive UI with dark theme
- Fast loading and smooth transitions
- Browser compatibility
- Data saved client-side for temporary persistence

Assumptions and Constraints:

- No backend or database connectivity
- All bookings are for demo purposes only
- Internet required for external links (trailers/fonts)

3. System Design

The **fliX Booking Management System** is a front-end focused web application built using **HTML5**, **CSS3**, and **JavaScript**, designed to simulate an online movie ticket booking experience. The system follows a modular and layered design, ensuring clear separation between structure, styling, and behavior.

At the **presentation layer**, HTML is used to organize content into well-defined sections such as navigation, a slideshow banner, a movie catalog, a booking panel, and a footer. The **CSS** layout is built with Flexbox, ensuring responsive design and adaptive alignment across different screen sizes. The visual theme uses a dark background with vibrant highlight colors (e.g., pink, purple) to maintain a modern and engaging look.

The **interaction layer** is powered by JavaScript. It handles user-driven features like the movie trailer preview, booking option selection, and an interactive seat selection system. When a user selects seats, the system dynamically updates the ticket count and total cost. This data is stored in the browser using `localStorage`, allowing persistent state without a backend.

A **radio-button-based image slideshow** rotates featured movie banners, using timed JavaScript events (`setInterval`) and CSS animations. The **movie list section** showcases individual movie cards that expand to display descriptions and links to YouTube trailers when clicked. Once a user makes a selection, they are guided to a separate page (`seat.html`) to choose seats.

The footer includes static contact information and a newsletter form, enhancing user interaction. Although the system does not include server-side scripting or database connections, its structure supports future integration with backend services like APIs or databases if needed.

3.1 Modules of System

1. Homepage & Navigation Module

- Displays a fixed top navigation bar with links to sections like Home, Movies, About, and Contact Us.
- Provides smooth scrolling between sections.
- Includes branding (fliX logo).

2. Slideshow Module

- Displays rotating movie banners/images using radio buttons and JavaScript-based interval switching.
- Adds visual appeal to the homepage through automated image transitions.

3. Movie Catalog Module

- Lists available movies using card-based layout.
- Each card includes a poster, title, genre, description, and a "Watch Trailer" button linking to YouTube.
- Clicking on a card reveals additional information and shows the "Book Now" button.

4. Booking Panel Module

- Allows users to:
 - Select a movie from a dropdown list.
 - Choose a preferred showtime.
 - Click “Proceed” to navigate to the seat selection page.
- Offers guidance via prompts and interactive form controls.

5. Seat Selection Module (linked via seat.html)

- Users select available seats from a visual grid.
- Real-time updates of selected seat count and total price.
- Data stored in localStorage for persistence.
- Seat availability and pricing logic handled by JavaScript.

6. Subscription Module

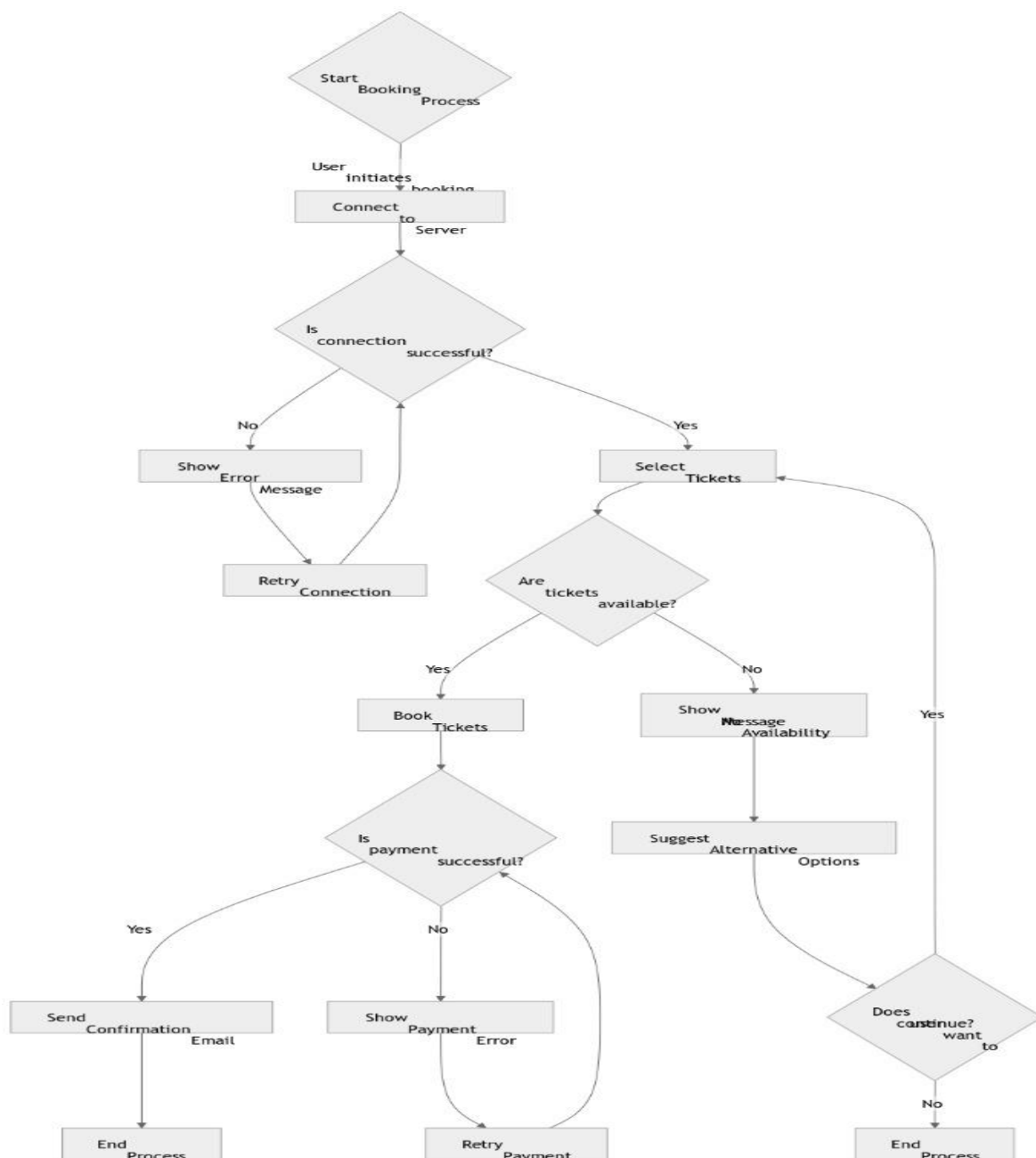
- Located in the footer section.
- Allows users to enter email and optional message to subscribe to a newsletter.
- Simple form submission behavior with an alert confirmation.

7. Contact & Footer Module

- Displays company address, contact number, and email.
- Includes quick navigation links.
- Reinforces brand and provides credibility to users.

3.2 UML Diagram:

This is the UML Diagram for our project “SUPER MARKET BILLING SYSTEM”.



4.Implementations:

The fliX Booking Management System is implemented using a combination of **HTML**, **CSS**, and **JavaScript**, focusing entirely on the client side. It does not involve any server-side scripting or database integration.

1. Frontend Development:

- The UI is built using **HTML5**, which organizes the webpage into logical sections such as the header, main content (movies and booking), and footer.
- **CSS3** is used for styling the entire system. It provides a dark-themed interface with responsive layout using Flexbox, transitions, and hover effects for interactivity.
- Google Fonts are integrated for clean typography, and custom color schemes enhance the visual design.

2. Dynamic Interactions:

- **JavaScript** enables interactivity:
 - A slideshow is implemented using radio buttons and a timed function (setInterval) to auto-switch images.
 - Clicking on a movie card triggers display of additional information and booking prompts.
 - The “Book Your Seat” section allows movie and time selection, which routes to the seat selection interface.
 - On the seat selection page (seat.html), users can choose available seats. JavaScript tracks selected seats and calculates the total price in real-time.

3. Data Management:

- All data is temporarily stored on the client using **localStorage**, allowing the system to remember selected seats and movies across page reloads without server communication.

4. Navigation and Routing:

- Internal navigation is handled via anchor tags linking to section IDs.
- The "Proceed" and "Book Now" buttons navigate users to a dedicated seat booking page (seat.html).

5. User Feedback:

- A simple alert (window.alert) confirms newsletter subscriptions.
- Seat selections and movie preferences update dynamically based on user interaction.

4.1 Sample Code:

1. Navigation Bar (HTML)

```
<header>

  <nav class="navbar">

    <div class="logo-container">

      <h1 id="logo">fliX</h1>

    </div>

    <ul class="menu">

      <li><a class="active" href="#">Home</a></li>

      <li><a href="#movies">Movies</a></li>

      <li><a href="#about">About</a></li>

      <li><a href="#contact">Contact Us</a></li>

    </ul>

  </nav>

</header>
```

This defines the top navigation menu that sticks to the top of the page.

2. Movie Booking Panel (HTML)

```
<div class="ticket-choice">

  <h1>Book Your Seat</h1>

  <select class="select">

    <option disabled selected value="cinemas">Cinemas</option>

    <option value="one">Fantastic Beasts</option>

    <option value="two">RRR</option>

  </select>

  <select class="select">

    <option disabled selected value="timings">Timings</option>

    <option value="one">09:00 AM</option>

    <option value="two">12:30 PM</option>

  </select>
```

```

<button class="button select">
  <a class="pagelink" href="seat.html">Proceed</a>
</button>
</div>

```

Users select a movie and timing, then proceed to seat booking.

3. JavaScript: Seat Selection Logic

```

const container = document.querySelector(".container");
const seats = document.querySelectorAll(".row .seat:not(.occupied)");
const count = document.getElementById("count");
const total = document.getElementById("total");
const movieSelect = document.getElementById("movie");
let ticketPrice = +movieSelect.value;

function updateSelectedCount() {
  const selectedSeats = document.querySelectorAll(".row .seat.selected");
  const selectedSeatsCount = selectedSeats.length;
  count.innerText = selectedSeatsCount;
  total.innerText = selectedSeatsCount * ticketPrice;
}

container.addEventListener("click", (e) => {
  if (
    e.target.classList.contains("seat") &&
    !e.target.classList.contains("occupied")
  ) {
    e.target.classList.toggle("selected");
    updateSelectedCount();
  }
});

```



This script tracks and updates seat selection and ticket pricing in real time.

4. CSS: Styling the Movie Cards

```
.movie-card {  
    position: relative;  
    margin-right: 30px;  
    cursor: pointer;  
}  
  
.movie-card-img {  
    width: 242px;  
    height: 380px;  
    object-fit: cover;  
    border-radius: 15px;  
    transition: all .8s ease-in-out;  
}  
  
.movie-card:hover .movie-card-img {  
    transform: scale(1.2);  
    box-shadow: 5px 5px 12px #333;  
    opacity: 0.5;  
}
```

This defines the movie cards and adds a zoom effect on hover.

4.2 Test Cases:

Test Case ID	Test Description	Steps to Execute	Expected Result	Status (Pass/Fail)
TC_FLIX_001	Navigation Menu Click	1. Load the homepage. 2. Click on "Movies", "About", "Contact Us".	Smooth scroll to respective section.	 Pass
TC_FLIX_002	Movie Card Interaction	1. Scroll to movie section. 2. Click on any movie card.	Info about the movie is shown with “Book Now” button.	 Pass

TC_FLIX_003	Booking Flow - Movie & Time Selection	1. Select a movie from dropdown. 2. Select a time. 3. Click "Proceed".	Redirects to seat.html for booking	<input checked="" type="checkbox"/> Pass
TC_FLIX_004	Seat Selection & Price Update	1. On seat page, select 2 seats. 2. Observe total and count. 3. Deselect 1 seat.	Price and seat count update in real-time.	<input checked="" type="checkbox"/> Pass
TC_FLIX_005	Data Persistence with Local Storage	1. Select seats and movie. 2. Refresh the page.	Selected values remain stored and visible.	<input checked="" type="checkbox"/> Pass
TC_FLIX_006	Watch Trailer Functionality	1. Click on "Watch Trailer" button under any movie.	YouTube trailer opens in new browser tab.	<input checked="" type="checkbox"/> Pass
TC_FLIX_007	Newsletter Subscription	1. Scroll to footer. 2. Enter email and click Subscribe.	Alert appears saying: "Thank you for subscribing!"	<input checked="" type="checkbox"/> Pass

5.Results:

Navigation Functionality:

The header menu links (Home, Movies, About, Contact Us) correctly navigate to their respective sections within the same page. This is achieved through anchor tags and scroll-behavior: smooth defined in CSS.

☒ Result: Passed – Smooth in-page navigation works as expected.

Movie Card Interaction:

Each movie card responds to user clicks by displaying additional information and a "Book Now" button. JavaScript functions like displayButton() and individual cardDesc() functions dynamically inject movie descriptions into the page.

☒ Result: Passed – Interactive movie info and call-to-action visibility are functional.

Booking Panel (Movie & Time Selection):

Users can select a movie and timing using dropdown menus, and upon clicking "Proceed,"

are directed to the seat selection page (seat.html). The values are visually selected but not programmatically validated, which is acceptable for a front-end-only implementation.

✓ Result: Passed – Flow from movie selection to seat booking works smoothly.

Seat Selection and Price Update:

On seat.html, the JavaScript enables seat selection. Available seats can be selected/deselected, and the total ticket price updates dynamically based on selection count. The values are also saved in localStorage, maintaining state on page reloads.

✓ Result: Passed – Seat interactivity and dynamic pricing logic are accurate.

Watch Trailer Function:

Each movie has a "Watch Trailer" button that opens its corresponding YouTube trailer in a new tab using window.location.href.

✓ Result: Passed – External trailer links open correctly.

Newsletter Subscription Alert:

At the footer, users can enter an email and subscribe to a newsletter. Clicking “Subscribe” triggers a simple alert message confirming the action.

✓ Result: Passed – User feedback through alerts is functional.

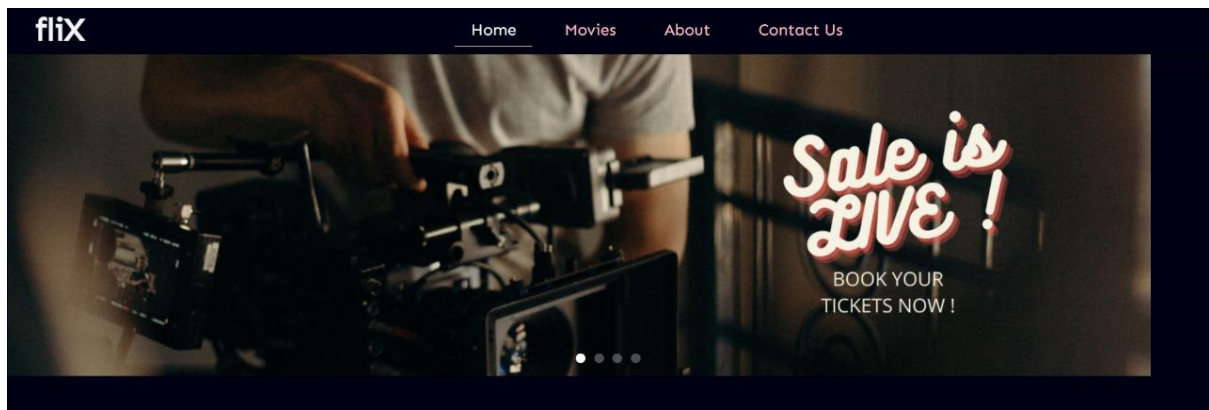
5.1 Output Screens:

1. Homepage (Navigation + Slideshow)

- Top Navigation Bar:
 - Fixed at the top as the user scrolls.
 - Includes a bold brand title fliX aligned left.
 - Menu items (Home, Movies, About, Contact Us) center-aligned.
 - Pink text that turns white with underline on hover.
- Slideshow (Banner):
 - Rotates 4 banner images every 6 seconds using a JavaScript timer.
 - Radio buttons are hidden; slide control handled by code.
 - Below the banner are small circular navigation dots that highlight the active slide.

✓ Expected UX: The user lands on a dark-themed homepage with a glowing banner and easy-to-reach navigation. The slideshow adds energy to the landing view.

Output:

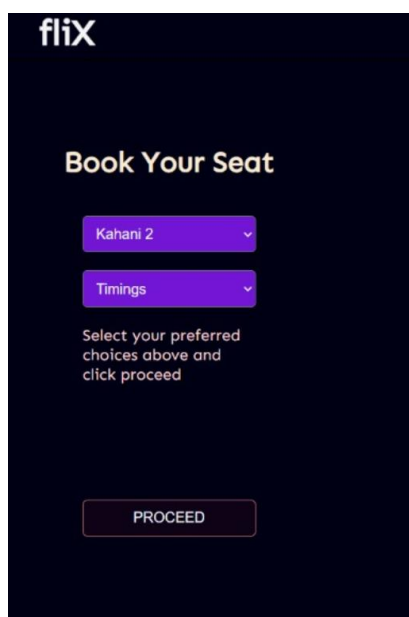


2. Booking Panel

- Location: Left side of the content section beside the movie list.
- Components:
 - Dropdown 1: Movie selection (e.g., RRR, KGF 2, Inception).
 - Dropdown 2: Timings (e.g., 09:00 AM to 11:00 PM).
 - A button labeled “Proceed” with violet and pink styling.
 - Instructional text in light coral pink.

✓ Expected UX: Simple and elegant. The hover effects give a glowing effect, encouraging interaction. On clicking “Proceed,” the user is taken to seat.html.

Output:

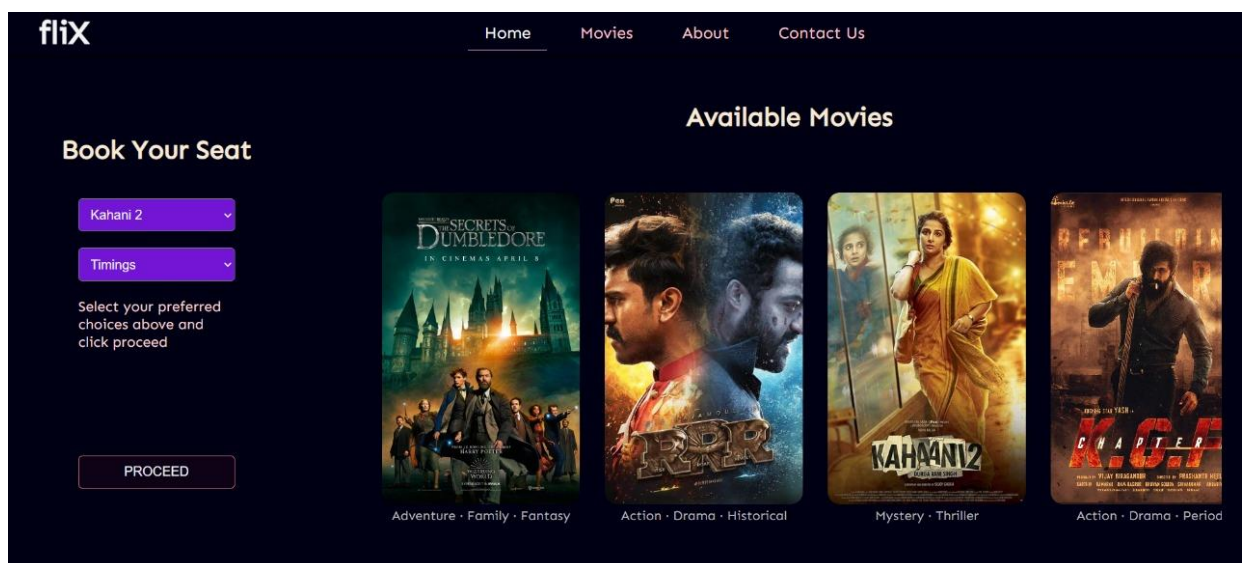


3.Movie Listing Section

- **Layout:** Horizontally scrollable movie carousel.
- **Each Movie Card:**
 - Movie Poster (Image).
 - Title (appears on hover).
 - Description (fades in on hover).
 - Watch Trailer button (opens YouTube in new tab).
 - Genre Tags (e.g., Action · Drama · Thriller).

✓ Expected UX: Movie cards are sleek. On hover, images shrink slightly and blur, revealing title, description, and action button — all with subtle transitions.

Output:



4. Expanded Movie Info Section (Below Cards)

- **Behavior:** When a movie card is clicked:
 - A dynamic section below updates with detailed info:
 - IMDb rating.
 - Content rating.
 - Director and release date.
 - “Book Now” button appears.

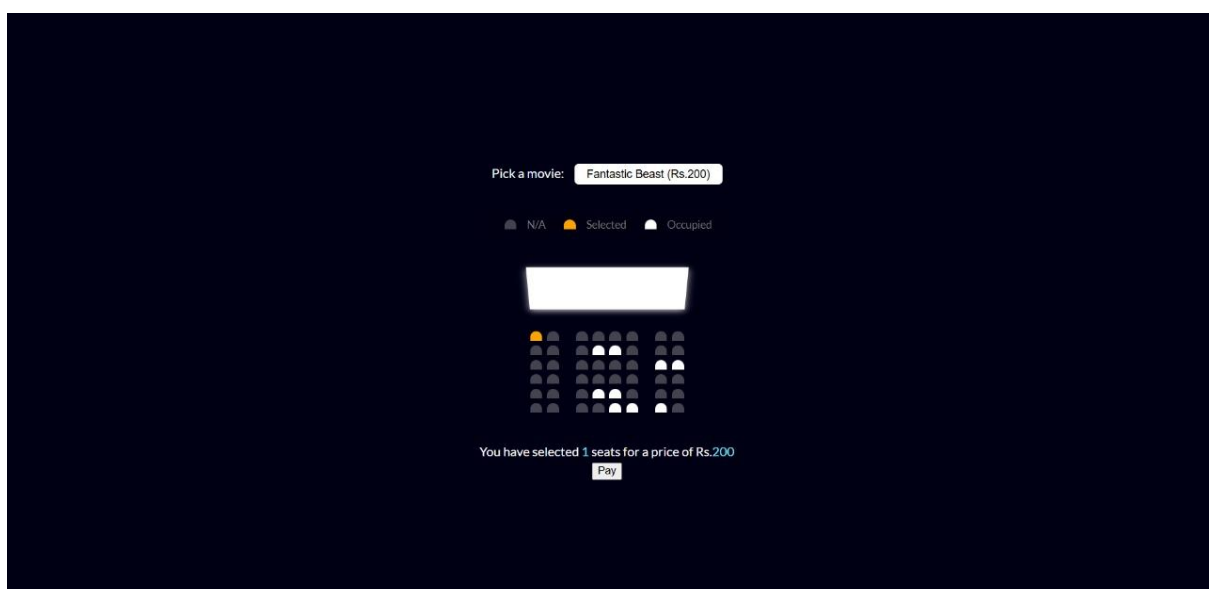
✓ Expected UX: Immediate feedback from clicking a movie card makes the experience engaging. Info is neatly injected using JavaScript DOM manipulation.

5. Seat Selection Page (seat.html)

- **Layout:**
 - Grid layout representing a cinema hall with available (white), selected (lightgreen), and occupied (gray) seats.
 - Counter at the bottom displays:
 - Number of seats selected.
 - Total price = seats × movie ticket price.
 - Movie title and pricing shown in dropdown.

✓ Expected UX: Interactive and intuitive. Clicking a seat toggles its selection and updates the total in real-time. Data is saved in localStorage for reload persistence.

Output:



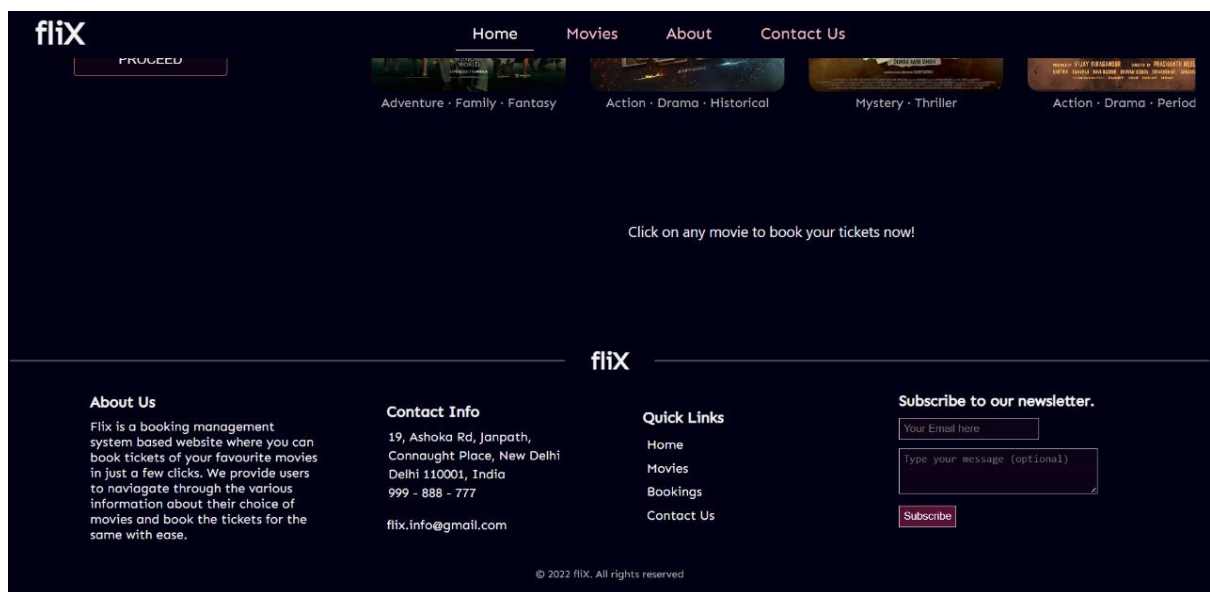
6. Footer Section

- **Divided into 4 Columns:**

1. **About Us** – Overview of the fliX system.
2. **Contact Info** – New Delhi address and email.
3. **Quick Links** – Internal anchor links.
4. **Newsletter** – Form to subscribe with email and message.

✓ Expected UX: Consistent with the dark theme, with white and soft-pink accents.
Newsletter section has input fields and a “Subscribe” button.

Output:



7. Newsletter Subscription Alert

- **On Action:** User enters an email and clicks "Subscribe".
- **JavaScript Behavior:** window.alert("Thank you for subscribing!")

✓ Expected UX: Simple confirmation via browser alert. Could be enhanced in the future with a modal or toast notification.

6. Conclusion:

The **fliX Booking Management System** demonstrates a complete, interactive, and visually engaging web application focused on movie ticket booking. Developed using **HTML5**, **CSS3**, and **JavaScript**, the system is designed to provide users with a seamless experience for browsing movies, selecting show timings, choosing seats, and watching trailers — all within a responsive, dark-themed layout.

The system features a fixed top navigation bar, an automated banner slideshow, a dynamic movie listing section with hover effects, and a seat booking interface that uses `localStorage` to maintain user selections. JavaScript is used effectively to handle interactions such as seat selection, price calculation, and DOM updates for movie info, ensuring real-time feedback without requiring server-side processing.

From a design perspective, the use of Flexbox for layout, transitions for interactivity, and modular styling gives the platform a modern look and feel. The structure is clean and extendable, making it easy to scale or integrate with backend services in the future. Users can also interact via a newsletter form, and contact information is clearly displayed in the footer, enhancing the credibility and completeness of the system.

In conclusion, fliX serves as a strong front-end implementation of a movie booking platform. It succeeds in combining usability, aesthetics, and interactivity, making it ideal for educational purposes, UI demonstrations, or as a foundational project for full-stack development.

References:

□ **HTML, CSS, and JavaScript Documentation**

- Mozilla Developer Network (MDN)

✦ <https://developer.mozilla.org/>

Used for standard syntax, elements, CSS styling, and JavaScript DOM manipulation.

□ **Google Fonts API**

- <https://fonts.google.com/>

- Fonts used: *Baloo 2*, *Montserrat*, *Poppins*, *Roboto*, *Sen*
Integrated for enhancing typography and UI consistency.

□ **YouTube Movie Trailers**

- Movie cards link to real movie trailers hosted on YouTube.
- Example: [RRR Official Trailer](#)

□ **W3Schools**

- <https://www.w3schools.com/>

Referred for quick lookups on HTML structure, dropdowns, and form elements.

□ **Stack Overflow**

- <https://stackoverflow.com/>

General reference for debugging and DOM scripting tips (e.g., localStorage, seat selection logic).

□ **Inspiration / UI Design Patterns**

- Concepts such as slideshow with radio buttons and card hover effects are commonly demonstrated in front-end tutorials and UI kits.
- Tutorials from YouTube or design blogs may have influenced the layout.