**PROJECT SYNOPSIS REPORT**

**ON**

**Movie Booking Website**

**SUBMITTED**

**TO**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**FOR**

**Backend Engineering**



**Submitted To:-** **Submitted By:-**

Mr. Rahul. Aman (2310990275).

Aryan Singla (2310990304).

Deepanshu Jindal (2310990321).

Gopesh Goyal (2310990337).

# **Index**

|  |  |  |
| --- | --- | --- |
| **Sr. no.** | **Topic** | **Page No.** |
| **1** | **Problem Statement** | **2** |
| **2** | **Title of project** | **2** |
| **3** | **Objective & Key Learning’s** | **2** |
| **4** | **Options available to Execute the project** | **2** |
| **5** | **Tech Stack** | **3** |
| **6** | **Advantages/ Disadvantages** | **3** |
| **7** | **Implementation Strategy** | **4** |
| **8** | **Conclusion** | **5** |
| **9** | **References** | **5** |

### **Problem Statement:-**

Traditional movie ticket booking methods are often time-consuming, inconvenient, and prone to errors. Users face long queues, limited seat availability, and lack of real-time updates, which can lead to frustration and missed opportunities. Similarly, administrators struggle with manually managing movie listings, schedules, and bookings, resulting in inefficiencies and inaccurate records. The Movie Booking Website addresses these issues by providing a **user-friendly online platform** where users can browse movies, select show timings, choose seats, and book tickets securely. Administrators can efficiently manage movie details, ensuring smooth and accurate operations.

**Title of Project:- Movie Booking Website**

**Objective & Key Learnings:-**

* **Develop a user-friendly online platform for booking movie tickets.**
* **Allow users to browse movies, select show timings, and choose seats.**
* **Enable administrators to manage movies, posters, and schedules efficiently.**
* **Working with MERN stack (MongoDB, Express.js, React.js, Node.js).**
* **Designing responsive and interactive interfaces.**
* **Improving debugging, problem-solving, and project management skills.**

**Options Available to Execute the Project:-**

* **Web-Based Application: Accessible via any modern web browser on desktop or mobile.**
* **Full-Stack MERN Implementation: Using MongoDB, Express.js, React.js, and Node.js.**
* **Local Development: Developing and testing the project on a local machine using VS Code and localhost server.**
* **Database Options: Using MongoDB Atlas for cloud database or local MongoDB for development purposes.**
* **Version Control: Using Git and GitHub to manage source code and collaboration.**

### **Tech Stack:**

**Frontend:**

* **React.js** – For building dynamic and responsive user interfaces.
* **HTML, CSS, Bootstrap** – For styling and layout.

**Backend:**

* **Node.js** – Server-side runtime environment.
* **Express.js** – To create RESTful APIs and handle server routing.

**Database:**

* **MongoDB** – NoSQL database to store users, movies, and booking information.
* **MongoDB Atlas / Compass** – For managing and visualizing the database.

**Tools & Others:**

* **VS Code** – Integrated development environment.
* **Postman** – To test APIs.
* **Git/GitHub** – Version control and source code management.
* **Web Browser** – For running and testing the application.

### **Advantages and Disadvantages:**

### **Advantages:**

#### Enables online booking anytime, anywhere.

#### Reduces long queues and waiting time at theaters.

#### Provides real-time seat availability to avoid double bookings.

#### Maintains booking history for users.

#### Allows administrators to manage movies, posters, and schedules efficiently.

#### Improves user experience with a responsive and interactive interface.

#### Disadvantages:

* Requires stable internet connection for access.
* Users may face technical issues or bugs in case of improper maintenance.
* Initial development requires time and technical expertise.
* Online payments (if integrated later) may pose security risks if not properly implemented.
* Limited to theaters and movies added in the system; new theaters need admin updates.

**Implementation Strategy:**

**1. Requirement Analysis**

* Identify functional and non-functional requirements for users and administrators.
* Study existing systems to find limitations and areas for improvement.
* Determine features like booking, seat selection, and admin controls.

**2. System Design**

* Plan architecture using **MERN stack**.
* Design database schema for users, movies, and bookings.
* Create wireframes and UI layout for responsive frontend.

**3. Frontend Development**

* Build responsive pages using **React.js, HTML, CSS, and Bootstrap**.
* Implement user interfaces for browsing movies, selecting seats, and booking tickets.
* Ensure cross-device compatibility (desktop, tablet, mobile).

**4. Backend Development**

* Implement server-side logic using **Node.js and Express.js**.
* Create **RESTful APIs** for frontend-backend communication.
* Handle authentication, booking logic, and admin operations.

**5. Database Integration**

* Set up **MongoDB collections** for users, movies, and bookings.
* Ensure proper data validation and storage.
* Enable real-time seat availability updates.

**6. Testing**

* Conduct functional testing to verify all features.
* Perform usability testing to ensure smooth user experience.
* Optimize performance and fix bugs.

**7. Deployment**

* Host application on platforms like **Heroku, Vercel, or a local server**.
* Ensure accessibility through web browsers.
* Monitor application performance post-deployment.

**8. Maintenance & Updates**

* Regularly fix bugs and errors.
* Update movie listings, posters, and show timings.
* Implement new features like online payment and recommendations.

### **Conclusion:**

The Movie Booking Website provides a **simple and efficient solution** for online ticket booking, eliminating long queues and ensuring real-time seat availability. Users can browse movies, select show timings, choose seats, and track their booking history conveniently. Administrators can manage movie listings, posters, and schedules effectively, ensuring smooth operations. Built using the **MERN stack**, the system integrates frontend, backend, and database technologies to deliver a responsive and interactive experience. This project not only **enhances user convenience** but also improves operational efficiency, laying the foundation for future enhancements like online payments, personalized recommendations, and mobile accessibility.

**References:-**

**https://www.clay.com/**

[**https://www.mongodb.com/docs/**](https://www.mongodb.com/docs/)

[**https://expressjs.com/**](https://expressjs.com/)

[**https://www.geeksforgeeks.org/chatbot-theory-explained/**](https://www.geeksforgeeks.org/chatbot-theory-explained/)

[**https://medium.com/build-your-own-ai-chatbot-a-beginners-guide**](https://medium.com/@suraj_bansal/build-your-own-ai-chatbot-a-beginners-guide-to-rag-and-langchain-0189a18ec401)