

ONLINE MOVIE TICKET BOOKING

*Project report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

GADDE VIJITHA (22UECM0084)
GURTHURTHI ESHWAR PRASAD (22UEID0045)
KASIREDDY DEEPTHI REDDY (22UEIN0010)

10211CS212 - WEB AND MOBILE APPLICATION DEVELOPMENT

SUMMER 2024-2025

*Under the guidance of
Dr.Manikandan N K,M.E.,Ph.D.
Assistant Professor(Senior Grade)*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE AND TECHNOLOGY**

**(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA
October,2024**

CERTIFICATE

It is certified that the work contained in the project report titled "ONLINE MOVIE TICKET BOOKING" by "GADDE VIJITHA (22UECM0084), GUTHURTHI ESHWAR PRASAD (22UEID0045), KASIREDDY DEEPTHI REDDY (22UEIN0010)" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor

Dr. Manikandan N K, Assistant Professor(Senior Grade)

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr.Sagunthala R&D

Institute of Science & Technology

October, 2024

Signature of Head of the Department

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

October, 2024

Signature of Dean

Prof. Dr. S P. Chokkalingam

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

October, 2024

DECLARATION

We declare that this written submission represents my ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

GADDE VIJITHA

Date: / /

(Signature)

GURTHURTHI ESHWAR PRASAD

Date: / /

(Signature)

KASIREDDY DEEPTHI REDDY

Date: / /

APPROVAL SHEET

This project report entitled (ONLINE MOVIE TICKET BOOKING by GADDE VIJITHA (22UECM0084), GURTHURTHI ESHWAR PRASAD (22UEID0045), KASIREDDY DEEPTHI REDDY (22UEIN0010) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners**Handling faculty**

Dr.Manikandan N K M.E.,Ph.D.

Date: / /

Place:

ABSTRACT

The project objective is to book cinema tickets online. The Ticket Reservation System is an Internet based application that can be accessed throughout the Net and can be accessed by anyone who has a net connection. This application will reserve the tickets. This online ticket reservation system provides a website for a cinema hall where any user of internet can access it. User is required to login to the system and needs a credit card for booking the tickets. Tickets can be collected at the counter and 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 website provides complete information regarding currently running movies on all the screens with details of show timings, available seats. Ticket reservations are done using credit card and can be cancelled if needed. Our online tickets reservation system is one of the best opportunities for those who cannot afford enough time to get their tickets reserved standing in long queues. People can book tickets online at any time of day or night. Our reservation system also provides option to cancel the tickets which are reserved previously

Keywords:

Booking System, User Authentication, Tickets, Website, Movie, Reservation.

LIST OF FIGURES

3.1	Architecture Diagram	6
3.2	Data Flow Diagram	7
3.3	Home Page	8
3.4	Login Page	9
3.5	Form Validation for Signup Page	10
3.6	: User Information using Jquery and DOM	11
3.7	Three Tier application using Node js MySQL	12
3.8	Reactive form for User Registration using Angular	13
3.9	Routing and Navigation in Angular	14
3.10	Creation of Microservices	15
3.11	Web application to Mobile App	16
5.1	Test Image	20

LIST OF TABLES

2.1	Hardware Specifications	3
4.1	Test Result	17

LIST OF ACRONYMS AND ABBREVIATIONS

AR	Augumented Reality
CSS	Cascading Style Sheet
DFD	Data Flow Diagram
HTML	Hyper Text Markup Language
IOT	Internet of things
ML	Machine Learning
UI	User Interfaces
UX	User Experience
VR	Virtual Reality

TABLE OF CONTENTS

	Page.No
ABSTRACT	iv
LIST OF FIGURES	v
LIST OF TABLES	vi
LIST OF ACRONYMS AND ABBREVIATIONS	vii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Aim of the project	1
1.3 Project Domain	2
1.4 Scope of the Project	2
1.5 Methodology	2
2 REQUIREMENT SPECIFICATION	3
2.1 User characteristics	3
2.2 Dependencies	3
2.3 Hardware specification	3
2.4 Software specification	4
3 WEBSITE DESIGN	6
3.1 Sitemap	6
3.2 Design Phase	7
3.2.1 Data Flow Diagram	7
3.3 Front End and Back End Design	8
3.3.1 Home Page	8
3.3.2 Signup and Login page	9
3.3.3 Form Validation	10
3.3.4 Parse the webpage using Jquery and DOM	11
3.3.5 Design of Three Tier application using Node js and MySQL	12
3.3.6 Design of Reactive form for User Registration using Angular	13

3.3.7	Develop web application to implement routing and navigation in Angular . . .	14
3.3.8	Creation of Microservices	15
3.3.9	Converting web application to mobile application	16
4	TESTING	17
4.1	Testing	17
4.1.1	Test Result	17
4.1.2	Test Bugs	18
5	WEBSITE LAUNCH AND HOSTINGS	20
6	RESULTS AND DISCUSSIONS	21
6.1	Website performance	21
6.2	Security	21
6.3	Responsiveness and mobile-friendliness	22
7	CONCLUSION AND FUTURE ENHANCEMENTS	23
7.1	Conclusion	23
7.2	Future Enhancements	23
8	SOURCE CODE	25
9	SCREENSHOTS	37
10	REFERENCES	39

Chapter 1

INTRODUCTION

1.1 Introduction

Creating an online movie ticket booking platform involves building both a website and a mobile app where users can easily find, book the tickets. The website movie ticket booking is a faster, cleaner and a tad more personal website, specially designed to make your booking experience better. Log in, navigate and find out for yourselves and if time permits leave your valuable feedback. Customers may view the contents of any movie show at any time and may book any movie ticket as needed. The program automatically calculates the subtotal and grand total. When a visitor decides to finally book the ticket, the order information including the buyer's name, address and billing instruction is stored in the database securely and payment has been made. The combo booking is also provided at the time of booking the ticket and there's a wonderful facility of delivering the combos at your seat when you are watching the movie. You need to register a new user whenever you have first visited or site then for future it will be stored in our database permanently and you can book your movie ticket at any time you want with this username and password.

1.2 Aim of the project

The aim of the online movie ticket booking project is to create a user-friendly and efficient platform that simplifies the process of finding, booking the movie tickets. The main objective of the online movie ticket booking system is to manage the details of movie, Ticket, customer, show timing. It manages all the information about movie. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the movie, ticket booking, customer. It tracks the details about the customer, show timing.

1.3 Project Domain

The project domain for online movie ticket booking involves the development of a digital platform that facilitates endless options. The movie ticket booking system has a back end system with a simple and user-friendly interface that allows administrators to easily change booking settings, manage reservations, add movies and events and upload graphic seat maps etc. It is to built separate incoming tickets into clear, customizable categories, which can be organized by priority, team, source or user.

1.4 Scope of the Project

The scope of the online movie ticket booking project is to establish a comprehensive system to provide a system that allows an organization to track and manage customer support requests. It may help collecting perfect management in details in a very short time. The collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It helps in current all works related to movie ticket booking system. It will be also reduced the cost of collecting the management and collection procedure will go on smoothly. It eliminates the need to wait in long lines at the box office or travel to the pickup point and reduces the risk of misplacing or damaging traditional paper tickets.

1.5 Methodology

Back end

In the back-end development methodology for an online movie ticket booking platform, the primary focus is on creating a robust server-side infrastructure that handles the logic and data management of the application. This involves designing databases to store property information, user data, and transaction records. The back end ensures secure user authentication, manages communication between the front end and the server, and facilitates essential features like booking processing and payment integration and user information.

Front end

HTML : Hypertext Markup Language (HTML) is the standard markup language for creating web pages and applications. Each page contains a series of connections to other pages called hyperlinks. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

CSS: Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language.

Chapter 2

REQUIREMENT SPECIFICATION

2.1 User characteristics

For an online movie ticket booking system, User uses the system to fetch information about available movies, their price, duration, date and timings and majorly to book tickets. Admin manages the system and keeps it up-to-date. Admin also looks over user requirements. User can also give feedback of the movie watched, in the form of ratings. Central server of the system must be able to handle all the incoming requests simultaneously. Back up of the databases in case of hardware failure, disaster, natural calamities..

2.2 Dependencies

Creating an online movie ticket booking platform involves several dependencies to make everything work smoothly. For the web and app development, you'll need foundational technologies like HTML for content, CSS for styling, and JavaScript for interactive features. Additionally, for real-time updates and smooth interactions, you might use technologies like WebSocket. Security is crucial, so integrating with secure payment gateways and following best practices for data protection is essential. Overall, these dependencies work together to create a reliable, user-friendly online movie ticket booking experience.

2.3 Hardware specification

Hardware Type	Specifications
Processor Speed	1.2 GHz or above
Hard Disk Drive	40 GB
Computer RAM	512 MB

Table 2.1: Hardware Specifications

1. Processor Speed:

Specification: 1.2 GHz or above

Explanation: This refers to the clock speed of the CPU (Central Processing Unit). A speed of 1.2

GHz indicates that the processor can perform 1.2 billion cycles per second. A higher clock speed typically means better performance, allowing the computer to handle more tasks simultaneously and execute programs more efficiently. This is especially important for running modern applications and multitasking.

2. Hard Disk Drive (HDD):

Specification: 40 GB

Explanation: The storage capacity of the hard disk drive is 40 gigabytes. This space is used for storing the operating system, applications, files, and other data. While 40 GB may be sufficient for basic tasks and older software, it is relatively limited by today's standards, where larger storage options are common. Users may need to manage their files carefully or consider external storage solutions if they require more space.

3. Computer RAM (Random Access Memory):

Specification: 512 MB

Explanation: RAM is the computer's short-term memory, where data is temporarily stored for quick access by the CPU. A specification of 512 MB is quite low by current standards, as modern computers typically come with several gigabytes of RAM (e.g., 4 GB, 8 GB, or more). Limited RAM can hinder performance, especially when running multiple applications or more demanding software, as it may lead to slower response times and increased reliance on disk swapping.

2.4 Software specification

For the web and application development of an online movie ticket booking system, the following software specifications are typically needed:

- Front-End Technologies:**

HTML, CSS, JavaScript, Basic Creating Web Pages.

- Back-End Technologies:**

Node.js, Django, or Ruby on Rails: Server-side technologies to handle logic and data management.

- Database:**

MySQL, MongoDB: Databases to store information like property details, user data, and transactions.

- Development Environment:**

Text Editor or IDE (Integrated Development Environment): Examples include Visual Studio Code, Atom, or Sublime Text..

- **Project Management:**

Trello, Jira, or Asana: Tools for managing tasks and workflows.

- **Testing:**

Jest, Mocha: Testing frameworks to ensure code reliability.

- **Deployment:**

Heroku, AWS, or Netlify: Platforms to deploy and host your web and mobile applications.

These specifications help create a well-functioning, secure, and user-friendly online room rental platform across both web and mobile applications.

- **Platform:**

Windows (2000 Professional, XP, Vista, Windows 10).

Chapter 3

WEBSITE DESIGN

3.1 Sitemap

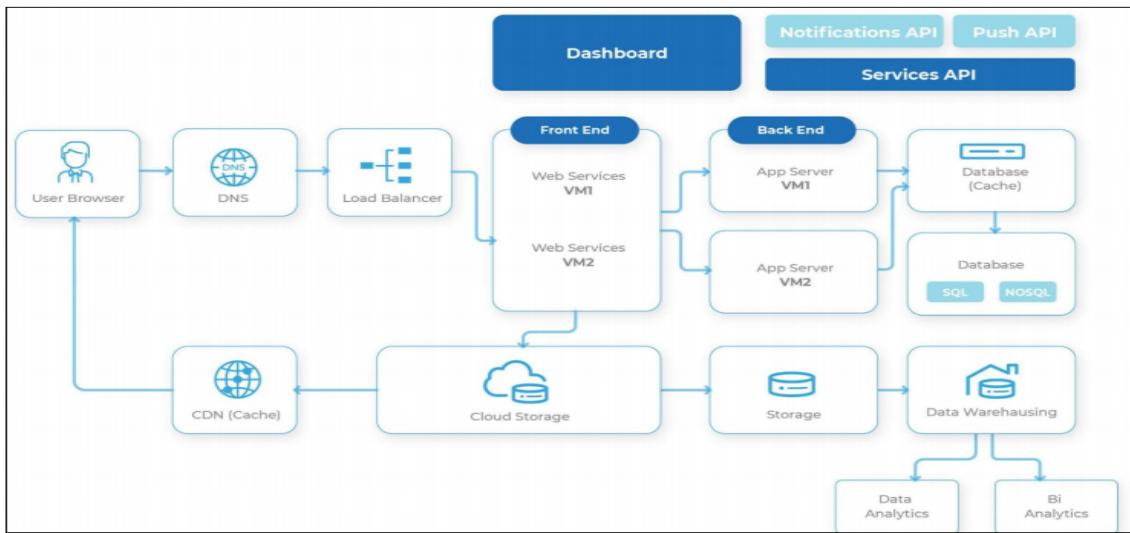


Figure 3.1: Architecture Diagram

Figure 3.1: The sitemap for an online movie ticket booking outlines the structure of the website and application in a clear and organized way. It typically includes main sections such as the homepage, where users can start their search, property listings where hosts can showcase their movie lists, a search page with filters for users to find suitable movies, a booking and payment section, and user profiles for managing personal information and bookings. This sitemap serves as a roadmap for developers, ensuring a smooth and logical flow of navigation for users as they explore and interact with the online movie ticket booking platform.

3.2 Design Phase

3.2.1 Data Flow Diagram

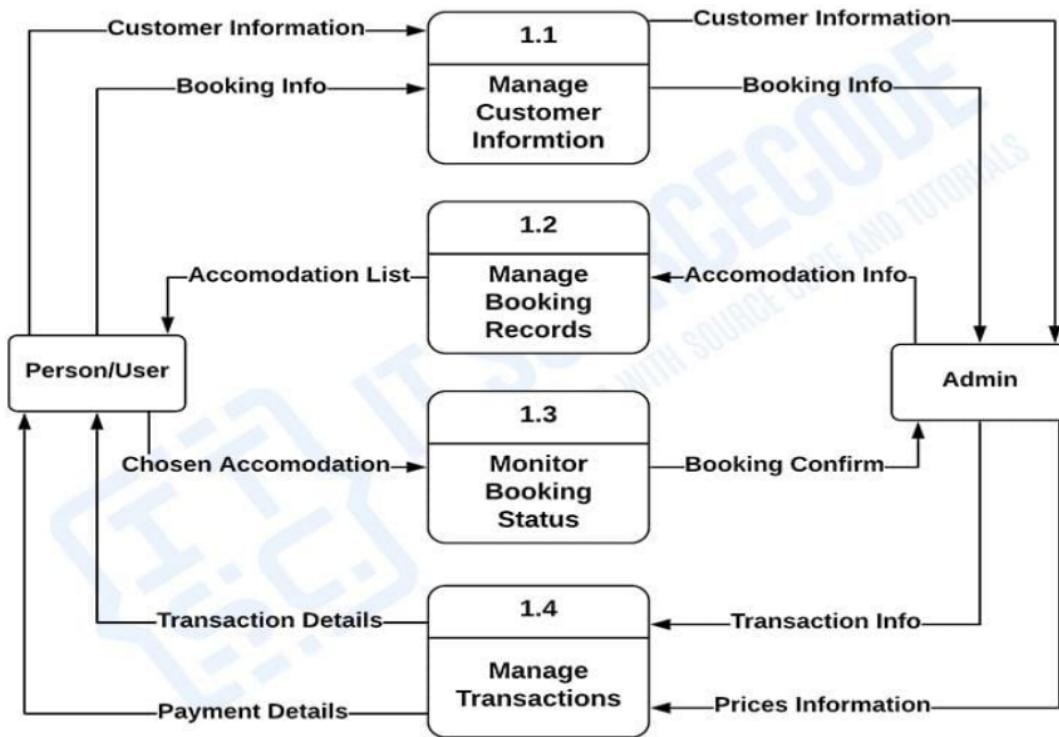


Figure 3.2: Data Flow Diagram

Figure 3.2: The DFD (Data Flow Diagram) for Online Movie Ticket Booking System describes the overall “flow” of data on the project. It is used to document the transformation of data (input-output) for project development. The online movie ticket booking system DFD consists of DFD levels 0, 1, and 2. It also uses entities, processes, and data to define the whole system

3.3 Front End and Back End Design

3.3.1 Home Page

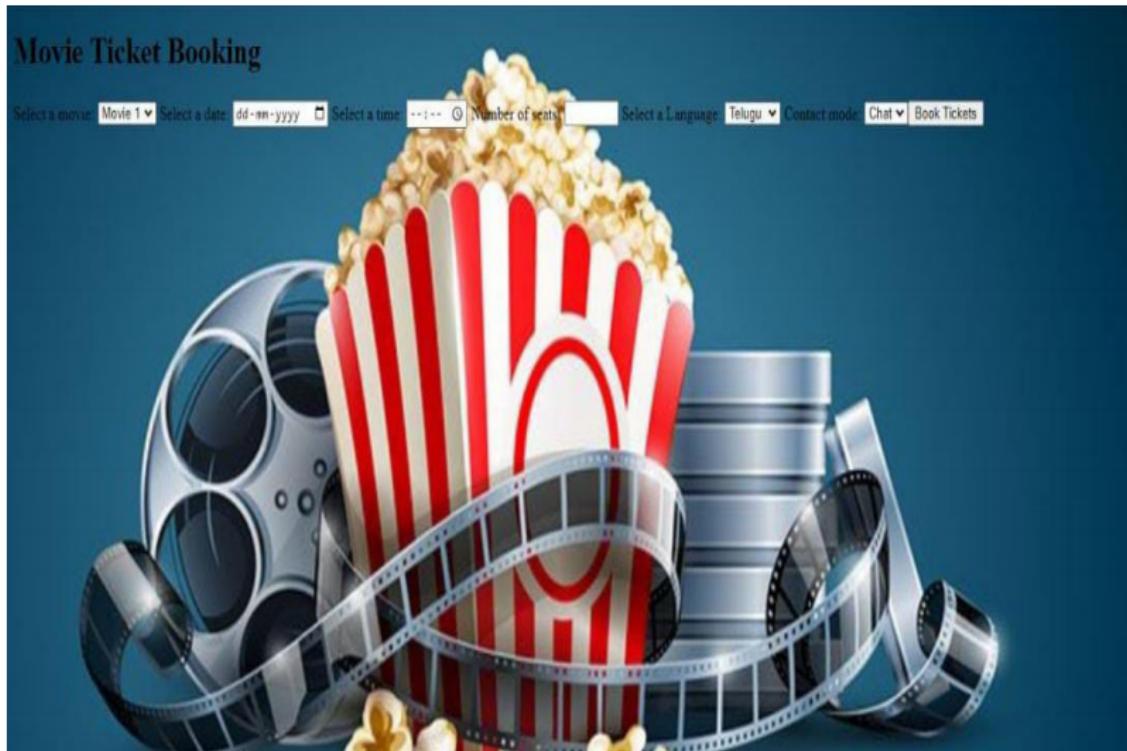


Figure 3.3: Home Page

Figure 3.3: Home Page is our complete outlook of the homepage for our website. our website has a source location and destination location at which date and search button,background image it looks like depot and there will be a login button for who are register and offers and hotel,etc..

3.3.2 Signup and Login page

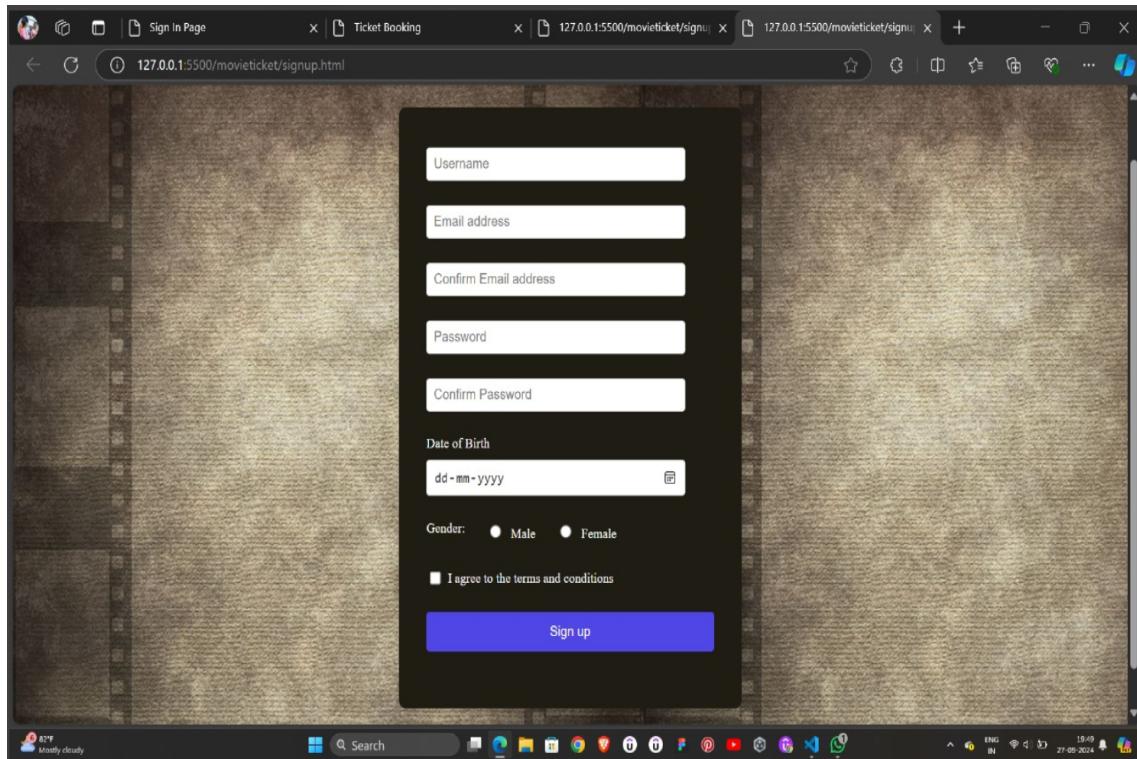


Figure 3.4: Login Page

Figure 3.4: The bus ticket booking website's signup page is the gateway for new users to join the community of travelers. It boasts a welcoming and user-friendly design, making it easy for individuals to create their accounts and start their travel journey. The signup form is comprehensive, requesting essential details such as name, email address, contact number, and a secure password. To enhance security and user trust, the page may include a password strength indicator. Once users fill out the required information and click the "Sign Up" button, they receive a verification email to confirm their account. The signup process ensures that travelers have access to personalized booking options, special promotions, and a seamless ticket reservation experience.

3.3.3 Form Validation



Figure 3.5: Form Validation for Signup Page

Figure 3.5: Form validation in a signup process is a crucial step to ensure that user-provided information is accurate and complete before creating an account. It typically involves checking that all required fields are filled out, verifying the format and validity of data such as email addresses, and confirming that passwords meet security criteria, like length and complexity. Effective form validation helps prevent errors, enhance security, and improve the overall user experience by guiding users to provide correct information and ensuring the integrity of data stored in the system

3.3.4 Parse the webpage using Jquery and DOM

The screenshot shows a web-based account signup form titled "Account Signup". At the top left, there are two buttons: "Hide" and "Show". Below the title, there are four input fields stacked vertically: "Email Id" (placeholder: "Personal email"), "Username" (placeholder: "Your username"), "Password" (placeholder: "Your password."), and "Reenter Password". At the bottom left of the form area is a button labeled "Click to Submit". The entire form is contained within a light gray box.

Figure 3.6: : User Information using Jquery and DOM

Figure 3.6: Initialization and Setup: Start by including the jQuery library in your HTML document using a script tag. Create a separate JavaScript file for your code. •Select Elements: Utilize jQuery selectors to pinpoint specific HTML elements on the webpage. These selectors can be element names, classes, IDs, or attribute values • Access and Manipulate Elements: Once you've selected elements, you can access their properties, attributes, and content using jQuery methods.

3.3.5 Design of Three Tier application using Node js and MySQL

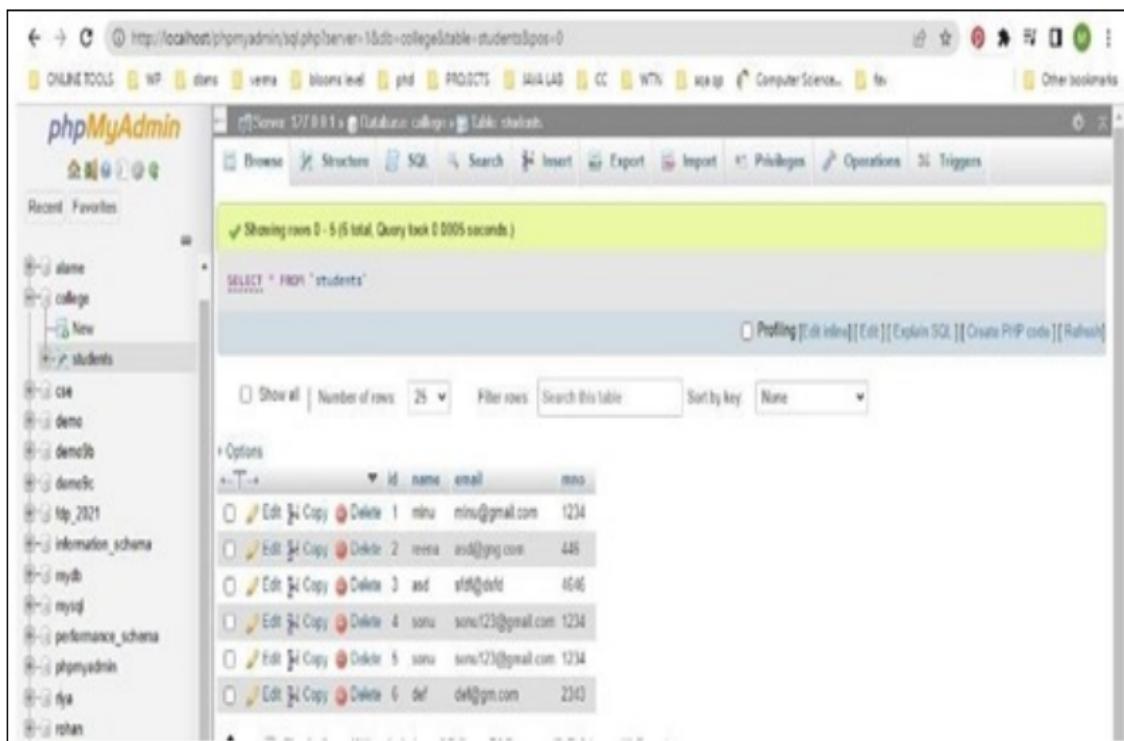


Figure 3.7: Three Tier application using Node js MySQL

Figure 3.7: depicts creating a reactive form for user registration using Angular. A Three-Tier Application is a software architecture pattern where an application is divided into three interconnected 13 components or tiers: the Presentation Tier, the Application Logic Tier, and the Data Storage Tier. Each tier has a specific role and responsibility within the application. This form allows users to input their information, by displaying the submitted data in the browser console upon submission, developers can efficiently debug and troubleshoot the registration process.

3.3.6 Design of Reactive form for User Registration using Angular

The screenshot shows a user registration form titled "Registration Form". It includes fields for First Name (Chaithanya), Last Name (Reddy), Email (vtu19978@veltech.edu.in), and Country (Australia). There is also a dropdown menu next to the country field. Under "Gender", there is a radio button for Male (selected) and one for Female. In the "Hobbies" section, there are three checkboxes: Sports (checked), Movies (checked), and Music (unchecked).

Registration Form	
First Name	Chaithanya
Last Name	Reddy
Email	vtu19978@veltech.edu.in
Country	Australia
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Hobbies	<input checked="" type="checkbox"/> Sports <input checked="" type="checkbox"/> Movies <input type="checkbox"/> Music

Figure 3.8: **Reactive form for User Registration using Angular**

Figure 3.8: Designing a reactive form for user registration using Angular involves creating a dynamic and interactive form that allows users to input their information and register for a service or application. Reactive forms in Angular are built using the Reactive Forms module, which provides a more programmatic and flexible way to manage form data and validation. Here's a theoretical overview of how to design a reactive form for user registration in Angular:

- Set Up Your Angular Project:
 - Make sure you have Node.js and Angular CLI installed.
 - Create a new Angular project using the Angular CLI: `ng new my-registration-app`.
- Import the `ReactiveFormsModule`: In your Angular application, you need to import the `ReactiveFormsModule` from `@angular/forms` in your `app.module.ts` file to enable the use of reactive forms.

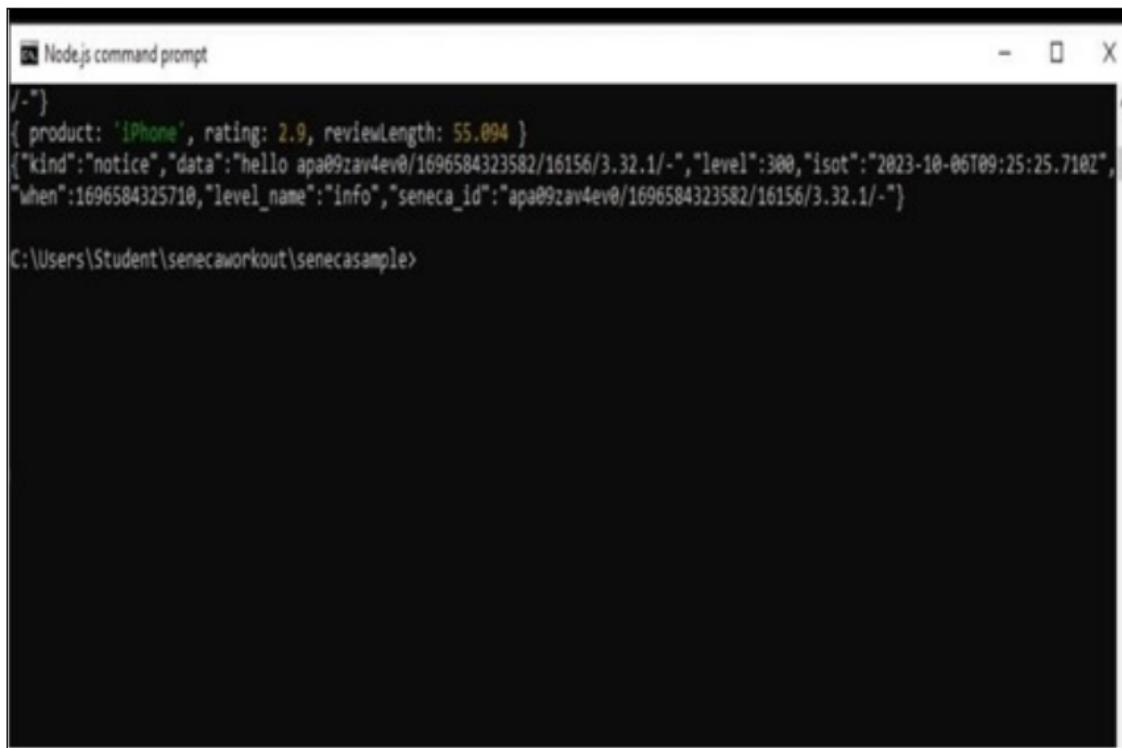
3.3.7 Develop web application to implement routing and navigation in Angular



Figure 3.9: Routing and Navigation in Angular

Figure 3.9: Angular's routing system also supports route parameters, enabling dynamic content generation. Additionally, you can implement route guards to protect routes and perform authentication or authorization checks.

3.3.8 Creation of Microservices



The screenshot shows a Windows command prompt window titled "Node.js command prompt". The command entered was "node seneca sample". The output displayed is a JSON object representing a service instance:

```
{"product": "iPhone", "rating": 2.9, "reviewLength": 55.094}
{"kind": "notice", "data": "hello apn09zav4ev0/1696584323582/16156/3.32.1/-", "level": 300, "isot": "2023-10-06T09:25:25.710Z", "when": 1696584325710, "level_name": "info", "seneca_id": "apn09zav4ev0/1696584323582/16156/3.32.1/-"}
```

The command prompt path is "C:\Users\Student\senecaworkout\senecasample>".

Figure 3.10: **Creation of Microservices**

Figure 3.10: Depicts development of a micro service to the website bus ticket reservation system through nodejs using sceneca toolkit.

3.3.9 Converting web application to mobile application

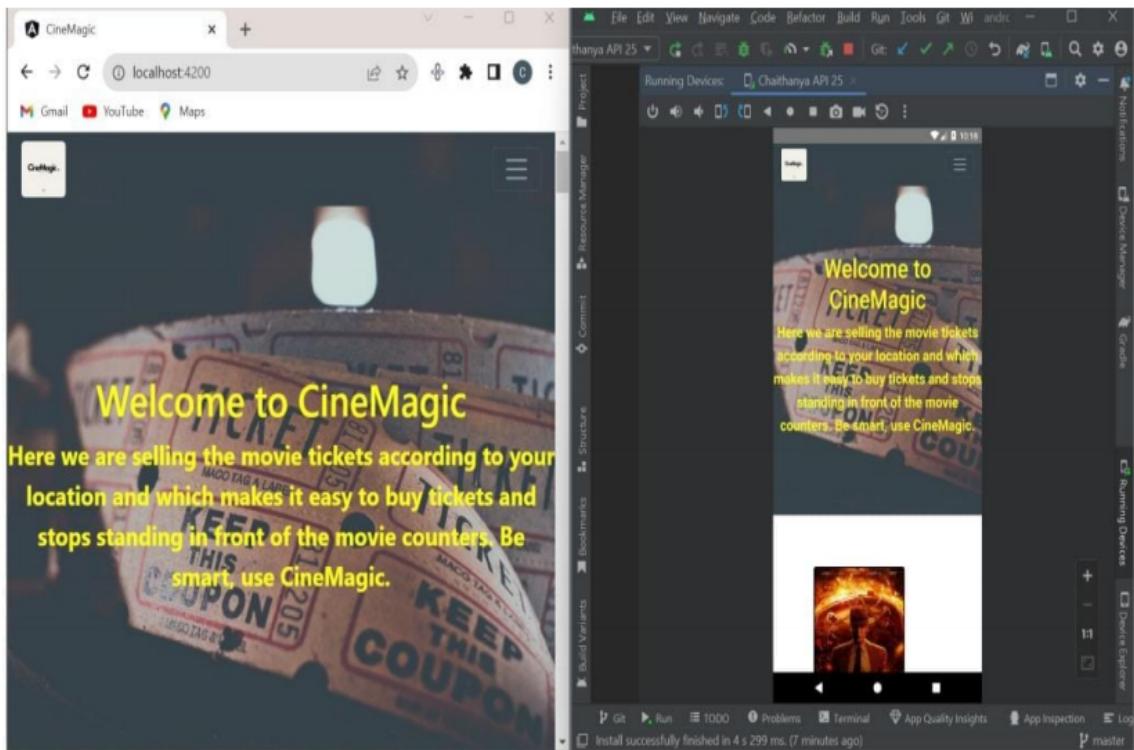


Figure 3.11: Web application to Mobile App

Chapter 4

TESTING

4.1 Testing

Testing for an online movie ticket booking platform involves several key aspects to ensure a smooth and reliable user experience. Functional testing checks if all features work as intended, from user registration and property listing to search, booking, and payment processes. Security testing is crucial to identify and fix vulnerabilities in payment transactions and user data protection. Compatibility testing ensures the platform works seamlessly across different browsers and devices. This comprehensive testing approach helps ensure the online movie ticket booking system is functional, secure, user-friendly, and performs well under various conditions.

4.1.1 Test Result

Test Id	Test Case	Purpose	Result	Expected output? (Y/N)
1	User Login	Ensure users can log in successfully.	Successful login, redirect to the user dashboard.	Yes
2	Movie Availability	Verify that the platform accurately displays the availability of movies.	Pass if the booking is successful; fail if an error occurs.	Yes
3	Payment Processing	Ensure the payment system works correctly.	Pass if payment is successful and confirmation is received; fail if an error occurs.	Yes
4	Messaging System	Confirm that users can send and receive messages.	Pass if the message is delivered; fail if there are delays.	Yes
5	Review Submission	Test the ability of users to submit reviews.	Pass if the review is visible; fail if there are submission errors.	Yes

Table 4.1: Test Result

EXPLANATION OF TEST RESULT

1 .User Login:

Purpose: Verify successful user authentication and ensure proper redirection.

Outcome: Successful logins are confirmed with redirection to the dashboard, indicating functionality is as expected.

2.Movie Availability:

Purpose: Ensure users can see available movies and book them successfully.

Outcome: Successful bookings confirm that the platform accurately reflects movie availability.

3.Payment Processing:

Purpose: Validate the payment system's functionality and reliability.

Outcome: Successful payments and receipt of confirmation indicate that the payment process is functioning as intended.

4.Messaging System:

Purpose: Ensure users can effectively communicate through the platform.

Outcome: Messages delivered without delay suggest that the messaging system operates smoothly.

5.Review Submission:

Purpose: Test the functionality of submitting reviews by users.

Outcome: Visibility of submitted reviews indicates that the review process works without errors.

4.1.2 Test Bugs

In the development and testing of an online movie ticket booking platform, several common types of bugs or issues may arise. Here are some test scenarios that could reveal potential bugs:

- User Registration:**

Test for issues in the user registration process, such as validation errors for incorrect input or unsuccessful account creation.

- Booking and Payment:**

Verify that users can smoothly complete the booking process, including selecting dates, entering payment information, and receiving confirmation. Test for payment gateway integration issues.

- Messaging System:**

Test the messaging system to ensure users can send and receive messages without any errors. Check for notification bugs.

- Reviews and Ratings:**

Test the review and rating system to confirm that users can leave feedback accurately, and that the system properly displays and updates reviews.

- **User Authentication:**

Ensure that user authentication works reliably, preventing unauthorized access and safeguarding user data.

- **Security:**

Perform security testing to uncover vulnerabilities related to user data, payment transactions, and overall system security.

- **User Feedback:**

Collect user feedback through testing phases to address any issues not captured in scripted tests and to enhance overall user experience. By systematically testing these aspects, developers can identify and address bugs to create a reliable and user-friendly online movie ticket booking platform.

Chapter 5

WEBSITE LAUNCH AND HOSTINGS



Figure 5.1: Test Image

WEBSITE URL: <https://cinemagic-falcon-14.netlify.app/>

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Website performance

The website performance for an online movie ticket booking platform is crucial for providing users with a smooth and responsive experience. The performance of a website for online movie ticket booking can vary based on several factors, including server speed, website design, and user traffic. Fast loading times, responsiveness, user interface (UI), security, scalability, optimization, user feedback, payment processing, content delivery network, regular maintenance. A well performing online movie ticket booking platform and application should deliver quick response times, seamless navigation, and timely updates, contributing to user satisfaction and trust in the platforms reliability. Regular monitoring and optimization are essential to maintain optimal performance as the platform evolves and attracts more users.

6.2 Security

Security in the development of an online movie ticket booking platform is like having a strong lock and key to protect your personal information and transactions. It ensures that only the right people those who should have access can get in. This involves using special codes to scramble sensitive details like your password and payment information, so even if someone tries to peek, they can't understand it. It's also like having a vigilant guard constantly checking for potential trouble and fixing any weak points in the system. This way, you can trust that your data is safe, and your online room rental experience is secure and worry-free. Here are some key security considerations:

- User Authentication:**

Implement secure user authentication mechanisms to ensure that only authorized individuals have access to the platform.

- Data Encryption:** Use encryption protocols (SSL/TLS) to safeguard data during transmission, preventing unauthorized access by encrypting sensitive information.

- Secure Payment Processing:**

Integrate reliable and secure payment gateways to protect financial transactions. Ensure compliance with industry standards for payment security.

- Authorization Controls:**

Implement robust authorization controls to restrict access to certain features and data, ensuring that users can only access information for which they have permission.

By incorporating these security measures, the online movie ticket booking platform can provide a secure environment for users, building trust and confidence in the protection of their personal and financial information. Regular monitoring, updates, and adherence to security best practices contribute to the overall security posture of the platform.

6.3 Responsiveness and mobile-friendliness

Responsiveness and mobile-friendliness mean that the platform works well and looks good on different devices, like computers, tablets, and smartphones. A responsive platform adjusts itself automatically so that all the features and information are easy to see and use, no matter what size screen someone is using. It's like a flexible piece of clothing that fits comfortably no matter your size. Being mobile-friendly is particularly important because many people use their phones to access websites and apps. Overall, making a virtual doctor platform responsive and mobile-friendly ensures that it's convenient and accessible for everyone, no matter what device they're using to access it. This mobile-friendly approach enhances the user experience, making it convenient for people to search for, book, and manage movies while on the go. It's all about providing a consistent and enjoyable experience, regardless of the device users choose to access the platform.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, the development of an online movie ticket booking platform involves creating a dynamic and user-centric ecosystem that simplifies the process of finding and booking movies streaming. By seamlessly integrating web and mobile applications, the platform caters to a diverse range of users, including owners, with a focus on responsiveness and user-friendly interfaces. The success of such a platform hinges on robust back-end technologies managing data and security, coupled with intuitive front-end designs that prioritize ease of use. Through testing and continuous improvement, the aim is to ensure a reliable, secure, and efficient online space that fosters trust between hosts and guests. Ultimately, the online movie ticket booking platform seeks to redefine the movie experience, making it accessible, secure, and enjoyable for all users involved.

7.2 Future Enhancements

Future enhancements for online movie ticket booking platforms in web and application development could include the integration of emerging technologies and the introduction of additional features to enhance user experience. Some potential advancements might include:

- Augmented Reality (AR) and Virtual Reality (VR):**

Develop AR features that provide interactive content related to the movie or allow users to unlock special promotions at the theater.

- Chat box and AI assistant:**

Implement chatbots or AI-powered assistants to answer user inquiries, assist with bookings, and provide real-time information.

- Language Translation and Multilingual Support:**

Enhance accessibility by incorporating language translation features, ensuring that users from different linguistic backgrounds can comfortably navigate and use the platform.

- Event booking:**

Diversify offerings by allowing users to book tickets for live events, concerts, and other entertainment activities through the same platform.

- **Multi platform development:**

Develop dedicated mobile apps for iOS and Android, ensuring a tailored experience for both major mobile platforms.

- **Analytics and data insights:**

Utilize data analytics to gain insights into user behavior, booking patterns, and popular movies, helping you make data-driven decisions.

- **Voice search and booking:**

Implement voice search and booking capabilities, allowing users to find movies and book tickets using voice commands.

These future enhancements aim to keep online movie ticket booking platforms innovative, competitive, and aligned with evolving user expectations and technological trends.

Chapter 8

SOURCE CODE

```
<html>
  <head>
    <link rel="stylesheet" href="signup.css" />
  </head>
  <body>
    <div class="container">
      <div class="form-wrapper">
        <div class="form-container">
          <form method="POST" action="#">
            <div class="form-group">
              <label class="label" for="username"></label>
              <input
                class="input"
                required
                type="text"
                name="username"
                id="username"
                placeholder="Username"
              />
            </div>

            <div class="form-group">
              <label
                class="label"
                for="email"
                placeholder="Email address"
              ></label>
              <input
```

```
        class="input"
        required
        type="email"
        name="email"
        id="email"
        placeholder="Email address"
    />
</div>

<div class="form-group">
    <label class="label" for="confirm-email"></label>
    <input
        class="input"
        required
        type="email"
        name="confirm-email"
        id="confirm-email"
        placeholder="Confirm Email address"
    />
</div>

<div class="form-group">
    <label class="label" for="password"></label>
    <input
        class="input"
        required
        type="password"
        name="password"
        id="password"
        placeholder="Password"
    />
</div>

<div class="form-group">
```

```

<label class="label" for="confirm-password"></label>
<input
    class="input"
    required
    type="password"
    name="confirm-password"
    id="confirm-password"
    placeholder="Confirm Password"
/>
</div>

<div class="form-group">
    <label class="label" for="dob">Date of Birth</label>
    <input class="input" required type="date"
        name="dob" id="dob" />
</div>

<div class="form-group gender-group">
    <span class="label">Gender:</span>
    <label class="radio-label">
        <input
            type="radio"
            class="form-radio"
            name="gender"
            value="Male"
        />
        Male
    </label>
    <label class="radio-label">
        <input
            type="radio"
            class="form-radio"
            name="gender"
            value="Female"
        />
        Female
    </label>
</div>

```

```

        />
        Female
    </label>
</div>

<div class="form-group checkbox-group">
    <input
        class="checkbox"
        type="checkbox"
        name="terms-and-condition"
        id="terms-and-condition"
    />
    <label class="checkbox-label" for="terms-and-condition">
        I agree to the terms and conditions</label>
    </div>

    <div class="form-group">
        <a href="signin.html">
            <button class="submit-btn" type="submit">
                Sign up</button></a>
        </div>
    </form>
</div>
</div>
</body>
</html>

.container {
    display: flex;
    flex-direction: column;
    justify-content: center;
    padding: 3rem 1rem;
}

```

```
}

body {
    background-image: url("Old\ Movie\ Background\ Pictures\
    Wallpaper\ Image\ For\ Free\ Download\ -\ Pngtree.jpeg");
}

.form-wrapper {
    margin-top: 2rem;
    margin-left: auto;
    margin-right: auto;
    width: 100%;
    max-width: 400px;
}

.form-container {
    padding: 2rem;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    border-radius: 8px;
    background: linear-gradient(0deg, rgb(30, 28, 19) 0%,
    rgb(30, 28, 19) 100%);
}

.form-group {
    margin-bottom: 1.5rem;
}

.label {
    display: block;
    font-size: 14px;
    font-weight: 500;
    color: white;
    margin-bottom: 0.5rem;
}
```

```
.input {  
    display: block;  
    width: 90%;  
    padding: 0.5rem;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
    font-size: 14px;  
    color: #333;  
    box-sizing: border-box;  
}  
  
.input:focus {  
    border-color: #4f46e5;  
    outline: none;  
    box-shadow: 6 6 4 2px rgba(79, 70, 229, 0.2);  
}  
  
.gender-group {  
    display: flex;  
    align-items: center;  
    gap: 1.5rem;  
}  
  
.radio-label {  
    font-size: 14px;  
    color: white;  
}  
  
.form-radio {  
    margin-right: 0.5rem;  
}  
  
.checkbox-group {  
    display: flex;
```

```
    align-items: center;
}

.checkbox {
    margin-right: 0.5rem;
}

.checkbox-label {
    font-size: 14px;
    color: white;
}

.submit-btn {
    display: inline-block;
    width: 100%;
    padding: 0.75rem;
    font-size: 14px;
    font-weight: 500;
    color: white;
    background-color: #4f46e5;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    transition: background-color 0.3s;
}

.submit-btn:hover {
    background-color: #4338ca;
}

.submit-btn:focus {
    outline: none;
    box-shadow: 0 0 0 2px rgba(79, 70, 229, 0.5);
}
```

```

// Ensure DOM is fully loaded
document.addEventListener('DOMContentLoaded', () => {
    // Variables to hold elements
    const searchInput = document.getElementById('search');
    const movieCards = document.querySelectorAll('.movie-card');

    // Search functionality: filters movie list based on user input
    if (searchInput) {
        searchInput.addEventListener('keyup', function() {
            const filter = searchInput.value.toLowerCase();
            movieCards.forEach((card) => {
                const title = card.querySelector('h3').textContent;
                toLowerCase();
                if (title.includes(filter)) {
                    card.style.display = "block";
                } else {
                    card.style.display = "none";
                }
            });
        });
    }

    // Highlight the selected movie on hover
    movieCards.forEach((card) => {
        card.addEventListener('mouseenter', function() {
            card.style.boxShadow = "0 4px 8px rgba(0,0,0,0.2)";
        });

        card.addEventListener('mouseleave', function() {
            card.style.boxShadow = "none";
        });
    });
});
<?php

```

```

// Start session
session_start();

// Include database configuration file
include('db_config.php');

// Fetch all movies from the database
$sql = "SELECT * FROM movies";
$result = mysqli_query($link, $sql);

// Check if user is logged in
$isLoggedIn = isset($_SESSION['user_id']);

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Online Movie Ticket Booking</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>Welcome to Movie Booking System</h1>
        <nav>
            <ul>
                <li><a href="index.php">Home</a></li>
                <?php if ($isLoggedIn) : ?>
                    <li><a href="logout.php">Logout</a></li>
                <?php else: ?>
                    <li><a href="login.php">Login</a></li>
                    <li><a href="signup.php">Sign Up</a></li>
                <?php endif; ?>
            </ul>
        </nav>
    </header>

```

```

        </nav>

    </header>

<div class="movie-list">
    <h2>Available Movies</h2>
    <?php if (mysqli_num_rows($result) > 0): ?>
        <div class="movies">
            <?php while ($movie = mysqli_fetch_assoc($result)): ?>
                <div class="movie-card">
                    <h3><?php echo htmlspecialchars($movie['title']) ; ?></h3>
                    <p><strong>Genre:</strong>
                        <?php echo htmlspecialchars($movie['genre']) ; ?></p>
                    <p><strong>Duration:</strong>
                        <?php echo $movie['duration'] ; ?> minutes</p>
                    <p><strong>Release Date:</strong>
                        <?php echo date("F j, Y", strtotime($movie['release_date'])); ?></p>
                    <p><?php echo htmlspecialchars($movie['description']); ?></p>
                    <a href="book.php?movie_id=<?php echo $movie['id']; ?>" class="btn">Book Now</a>
                </div>
            <?php endwhile; ?>
        </div>
    <?php else: ?>
        <p>No movies available at the moment.
            Please check back later.</p>
    <?php endif; ?>

```

```

</div>

<footer>
    <p>&copy; <?php echo date("Y"); ?> Movie Booking System.
    All rights reserved.</p>
</footer>

</body>
</html>

<?php
CREATE TABLE movies (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    genre VARCHAR(100) NOT NULL,
    duration INT NOT NULL,      -- Movie duration in minutes
    release_date DATE NOT NULL,
    description TEXT,
    poster_url VARCHAR(255)    -- URL for movie poster image
);
CREATE TABLE theaters (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    location VARCHAR(255) NOT NULL
);
CREATE TABLE showtimes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    movie_id INT NOT NULL,
    theater_id INT NOT NULL,
    show_date DATE NOT NULL,
    show_time TIME NOT NULL,
    available_seats INT NOT NULL,
    FOREIGN KEY (movie_id) REFERENCES movies(id),
    FOREIGN KEY (theater_id) REFERENCES theaters(id)
);

```

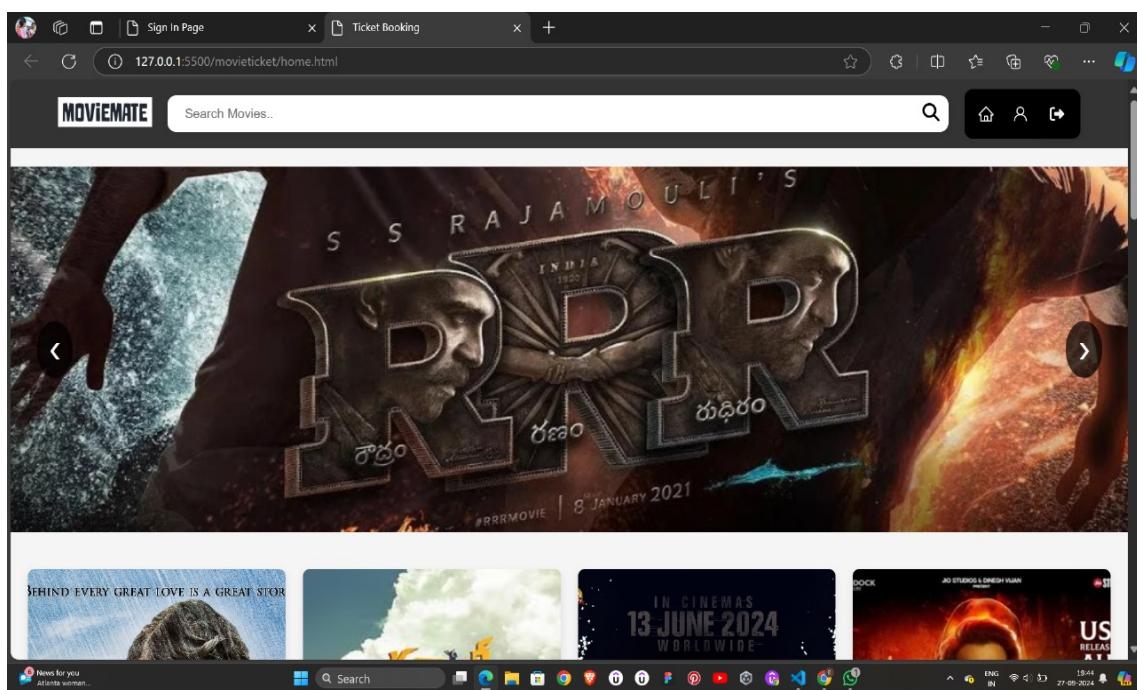
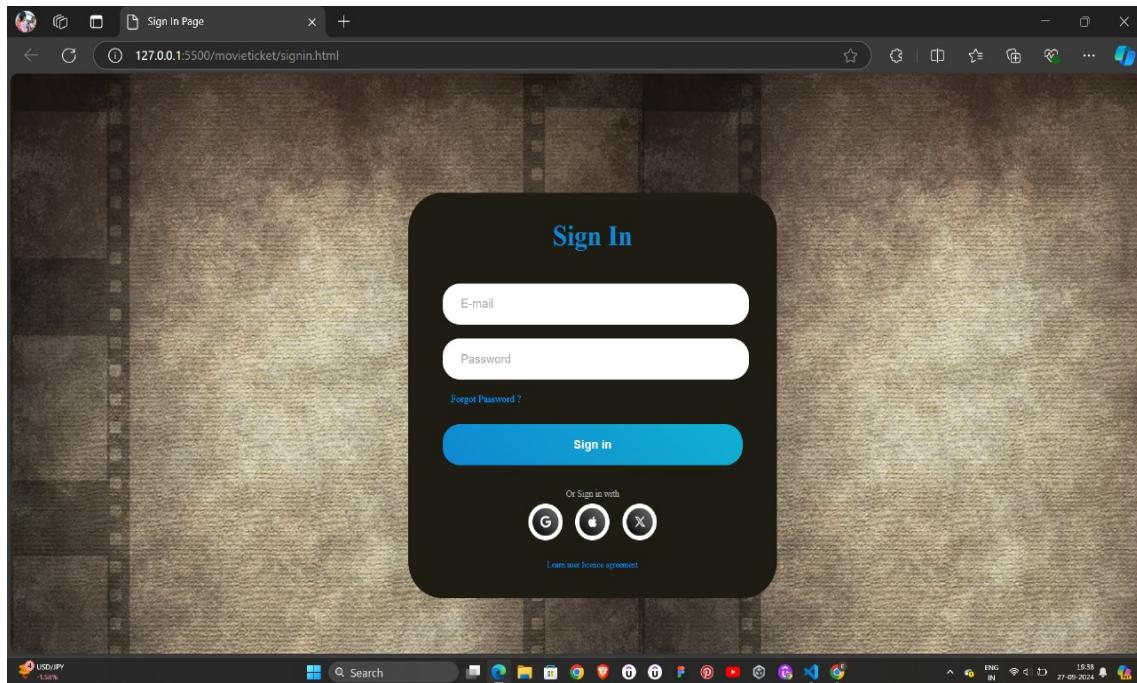
```
INSERT INTO movies (title, genre, duration,
release_date, description, poster_url)
VALUES
('Inception', 'Sci-Fi', 148, '2010-07-16',
'A thief who steals corporate secrets
using dream-sharing technology is tasked
with planting an idea into the mind of a CEO.',
'inception.jpg'),
('The Dark Knight', 'Action', 152, '2008-07-18',
'Batman faces off against the Joker in Gotham City.', 'dark_knight.jpg'),
('Avengers: Endgame', 'Action', 181, '2019-04-26',
'The Avengers assemble once more in order
to reverse Thanos\'s actions and restore balance
to the universe.',
'endgame.jpg');

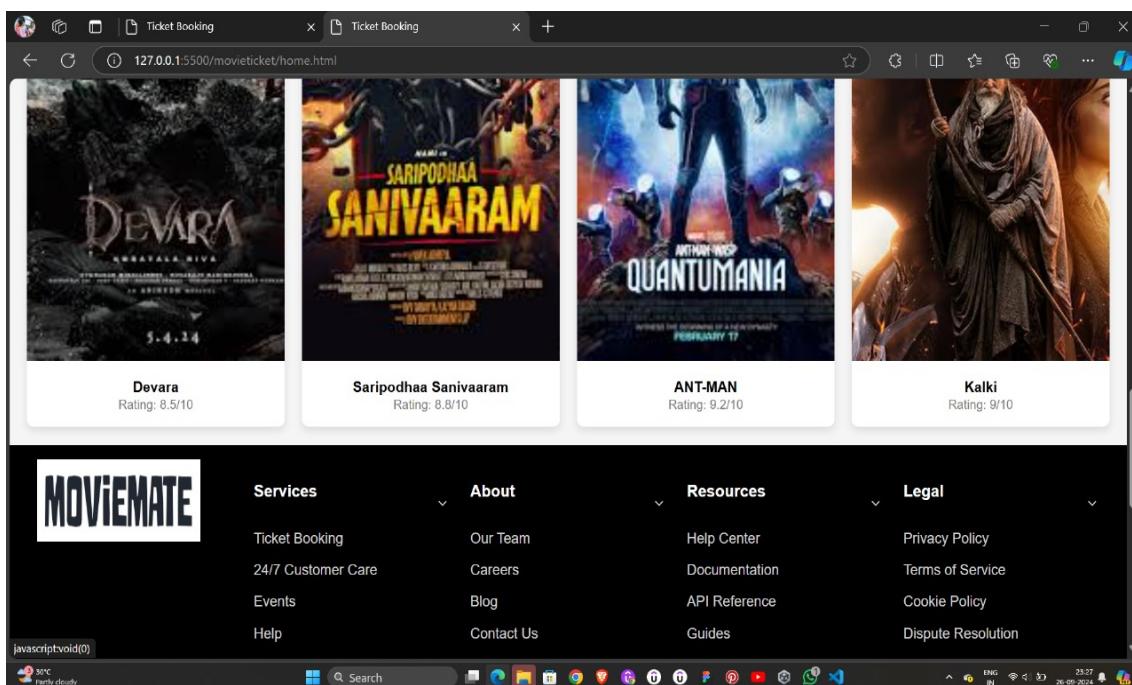
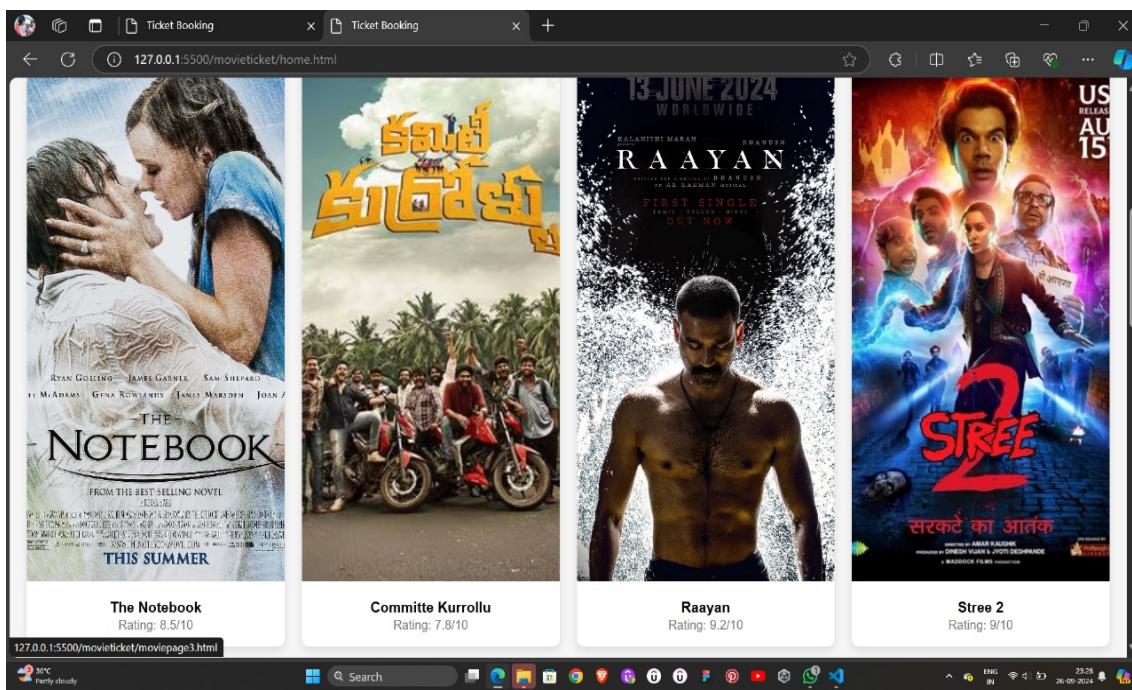
INSERT INTO theaters (name, location)
VALUES
('Cineplex Downtown', '123 Main St, City Center'),
('Grand Cinema', '456 Elm St, Northside');

INSERT INTO showtimes (movie_id, theater_id,
show_date, show_time, available_seats)
VALUES
(1, 1, '2024-09-28', '18:00:00', 100),
(1, 2, '2024-09-28', '21:00:00', 50),
(2, 1, '2024-09-28', '20:00:00', 75),
(3, 2, '2024-09-29', '19:30:00', 120);
```

Chapter 9

SCREENSHOTS





Chapter 10

REFERENCES

[1]"Fandango(1970): Fandango is a well-known platform for purchasing movie tickets online and offers a user-friendly experience.

[2]"Atom Tickets(1995): Atom Tickets is another popular app and website that allows you to browse showtimes, book tickets, and even order concessions.

[3]"BookMyShow(2010): If you're in India, BookMyShow is a widely used platform for booking movie tickets and also offers event ticketing services.

[4]"AMC Theatres(2000): If you frequent AMC theaters, their website and mobile app make it easy to book tickets and explore movie showtimes.

[5]"Regal Cinemas(1990): Regal Cinemas has its own online ticket booking platform for users interested in their theaters.

[6]"Cineworld(2002): If you're in the UK or other European countries, Cineworld's website and app are suitable for booking movie tickets.

[7] T. P. S. R. R. Adhikari(2018): "Enhancing User Experience in Online Ticket Booking Systems," International Journal of Computer Applications, vol. 179.

[8]"Showcase Cinemas(1999): Showcase Cinemas' website and app enable users to book tickets and discover movie listings.

[9]"Netflixed(2000): The Epic Battle for America's Eyeballs" by Gina Keating - This book provides insights into the rise of streaming platforms and how they have influenced movie distribution and viewing.

[10]"Blockbuster(1995): How Hollywood Learned to Stop Worrying and Love the Summer" by TomShone - While not exclusively about online ticket booking, it covers the evolution of the movie industry in the digital age.

[11] N. Patel and M. Sharma(2022): "User Satisfaction in Online Ticket Booking: A Case Study," Journal of Management and Business Research, vol. 12.