



**Smart
Internz**

Modern Application Development (Java SpringBoot)

Project Report

Movie Ticket Booking System

By

Team 499

Akshaj Nevgi	20BCE2645	Vellore Campus
Sharann PS	20BML0031	Vellore Campus
M Pratham	20BEC0052	Vellore Campus
Ayush Gupta	20BAC10029	Bhopal Campus

CHAPTER 1

INTRODUCTION

1.1. Overview

The project objective is to book cinema tickets online. The Movie Ticket Booking 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 movie 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 the theatre is one of the best mediums 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.

1.2. Purpose

The main purpose of maintaining a database for Movie Ticket Booking System is to reduce the manual errors involved in the bookings and cancellation of tickets and make it convenient for the customers and the providers to maintain the data about their customers and also about the seats available at them. Due to automation many loopholes that exist in the manual maintenance of the records can be removed. The speed of obtaining and processing the data will be fast. For future expansion the proposed system can be web enabled so that clients can make various enquiries about movie ticket booking. Due to this, sometimes a lot of problems occur and they are facing many disputes with the customers. To solve the above problem, we design a database which includes customer details, available movie seats, etc.

CHAPTER 2

LITERATURE SURVEY

2.1. Existing Work

This paper presents a cinema ticketing system using SpringBoot and Vue to address issues with manual ticket selling. It covers requirements analysis, system design, function implementation, and key technologies such as Vue, SpringBoot, MySql, and Redis [1]. The online movie ticket booking system designed simplifies the process of ticket booking, providing convenience and eliminating worries about ticket availability in this paper. It allows users to easily book tickets from anywhere, making their recreation time smoother and hassle-free [2]. This research aims to analyse the feasibility and requirements for a Multiplex Management system (MMS) that automates ticket reservation, show screenings, personnel management, statistical reporting, and provides personalized movie suggestions. The system has a web interface for users, admins, and employees, and utilizes machine learning to improve movie recommendations and maximize ticket sales [3].

This paper present easy and efficient way to book the tickets online. It is quite fascinating to sit at home browses the movies which have been released, read the review, and then book the tickets according to our feasibility. Movie booking system is in action for the last 15 years, but it gained its popularity after 2004 and its reach increased several folds in the past 6-10 years because of increasing net accessibility. In the proposed research paper, we have sorted out some of the major issues related to online movie management system. Due to increased traffic server problem and transaction failure are the major problems with online movie booking system. To sort out these Node JS and Mongo DB connectivity has been used. Also, there are some more features which have been addend and are new to movie booking system like mailing the review and trailer of the movie to the user while they book the ticket and showing the view of screen from different orientation of screen [4].

2.2. Proposed Solution

This paper has addressed some of the key concerns regarding online movie management systems in the study paper that is being submitted. The two main issues with the online movie booking system are transaction failure and server issues caused by increased

demand. Mongo DB connection has been utilized to organize these. Additionally, there are a few new features that have been added to the movie ticketing system, such as the ability to email users the movie review and trailer when they book their tickets and display the screen from various screen orientations [5]. This paper describes a theatre booking system that enables users to book or purchase tickets as well as search for movies within a specified radius of their present location. Users have the option of purchasing tickets using e-money received as a refund for cancelled tickets or credit cards kept in a secure wallet integrated into the program. All client-server queries are transmitted across the Internet as a network carrier. MYSQL has been utilized for Data Management [6]. This paper provides a web-based application called "Movie Ticket Purchase System" was developed. People can buy movie tickets from any movie theatre using this app. The process of registering or logging in is required prior to purchasing a ticket. The program is developed and implemented with proper attention paid to each stage of the software development life cycle. There are three panels on this website: one for the administrator, one for the theatre assistant, and one for the user or customer [7].

The project objective is to book cinema tickets in 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 [8].

CHAPTER 3

THEORITICAL ANALYSIS

3.1. Block Diagram

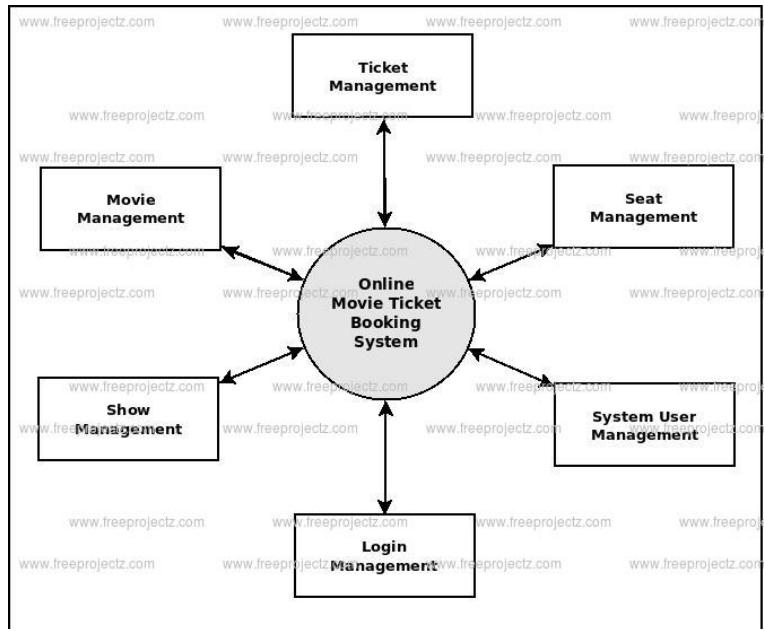


Fig 3.1 Online Ticket Booking System

User Interface: This component represents the user-facing interface through which users interact with the system. It could be a website, mobile app, or any other interface that allows users to search for movies, select showtimes, and book tickets.

Application: The application layer handles the core logic of the ticket booking system. It receives user requests from the user interface, processes them, and coordinates the interaction between different components of the system.

Database Management: This component manages the storage and retrieval of data related to movies, showtimes, theatres, seating availability, user information, and other relevant data. It provides the necessary data for the application layer to process user requests and make decisions.

Payment Gateway: The payment gateway handles secure online payment transactions. It integrates with different payment methods and ensures the confidentiality and integrity of financial transactions.

Movie Information: This component stores details about movies, including titles, descriptions, genres, ratings, cast, and other related information. It can be a combination of an internal database and external APIs that provide movie data.

Ticket Generation: This component generates tickets once a user successfully books a ticket. It may involve creating a unique ticket ID, associating it with the booked movie, showtime, and seat information, and providing the generated ticket to the user.

3.2. Hardware/Software Designing

Hardware Requirement:

Server Infrastructure: Sufficient server capacity to host the application and handle expected traffic. This can be physical servers or cloud-based infrastructure like Amazon Web Services (AWS) or Microsoft Azure.

Storage: Adequate storage capacity to store data such as movies, theatres, bookings, and user information. This can be in the form of hard drives, solid-state drives (SSDs), or cloud-based storage solutions.

Network Infrastructure: Reliable network connectivity to ensure smooth communication between the application server, database server, and other components.

Software Requirement:

Operating System: Choose an operating system compatible with Spring Boot, such as Linux (e.g., Ubuntu, CentOS), Windows Server, or macOS.

Java Development Kit (JDK): Install the appropriate version of JDK required for Spring Boot. Spring Boot generally supports multiple JDK versions, so choose the one recommended by the Spring Boot documentation.

Integrated Development Environment (IDE): A popular choice for Spring Boot development is IntelliJ IDEA, Eclipse, or Spring Tool Suite (STS). These IDEs provide tools and features for coding, debugging, and deploying Spring Boot applications.

Spring Boot: Download and install the necessary version of Spring Boot framework. We use either Maven or Gradle as the build tool for managing dependencies and building the project.

MySQL Database: MySQL is a popular relational database management system. Install MySQL Community Server and optionally MySQL Workbench, a graphical tool for managing MySQL databases.

Web Server: Spring Boot applications can be deployed on various web servers or servlet containers, such as Apache Tomcat, Jetty, or Undertow. Spring Boot provides embedded servers as well, eliminating the need for a separate web server in simple deployments.

Additional Libraries and Frameworks: Depending on the specific requirements of movie ticket booking system, we may need to use additional libraries and frameworks, such as Spring Data JPA for database access, Spring Security for authentication and authorization, etc.

CHAPTER 4

EXPERIMENTAL INVESTIGATIONS

Frontend development involves building the user interface (UI) and implementing the visual elements of the movie ticket booking application. It focuses on creating an intuitive and engaging user experience that allows users to interact with the application seamlessly. The following activities are part of the frontend development process:

UI Design And Layout

- Design the overall user interface (UI) for the movie booking application.
- Create wireframes and mock-ups to visualize the layout and structure of the application.
- Determine the colour scheme, typography, and overall visual style.
- Implement responsive design to ensure the application is compatible with different devices.

Movie Search And Booking UI

- Develop the movie search functionality, allowing users to search for movie based on criteria of city, language etc.,,
- Implement the movies listing page, displaying relevant movie options with details such as movie details, show timings and availability.
- Create the movie ticket booking flow, enabling users to select movies, specify movie details, and complete the booking process.

User Authentication And Profile UI

- Implement user authentication and registration functionality, including login and signup forms.
- Create user profile pages, allowing users to view and edit their personal information, manage bookings, and track their movie history.
- Design password reset and email verification flows for user account management.

Backend development involves building the server-side components and logic of the movie booking application. It focuses on handling the business logic, processing requests from

the front end, and interacting with the database. The following activities are part of the backend development process:

Server Setup And API Development

- Set up the server environment and choose a suitable backend framework such as Java Spring Boot.
- Develop the RESTful APIs for movie search, booking management, user authentication, and profile management.
- Implement server-side validation and error handling for API requests and responses.
- Integrate with external services such as payment gateways or movie data providers, if required.

Database Integration And ORM

- Set up the database server (e.g., MySQL or MongoDB) and establish the necessary database connections.
- Design the database schema based on the application requirements, including tables/entities and their relationships.
- Implement object-relational mapping (ORM) techniques (e.g., Hibernate or Spring Data) to interact with the database.
- Develop database queries and CRUD operations for movie data, user information, and bookings.

Integration is the process of combining and connecting the frontend and backend components of the flight booking application to create a unified and fully functional system. It involves establishing communication channels, exchanging data, and ensuring seamless interaction between the frontend UI and backend APIs. The following activities are part of the integration process:

Frontend-Backend Integration

- Integrate the frontend UI components with the backend APIs, ensuring proper communication and data exchange.
- Implement API calls from the front end to retrieve flight data, make bookings, and manage user-related actions.

- Handle data validation and error responses between the frontend and backend components.
- Conduct thorough testing to ensure seamless integration and compatibility between frontend and backend.

Containerization is the process of packaging an application and its dependencies into a standardized unit called a container. Containers offer a lightweight and consistent runtime environment that can be easily deployed across various platforms and environments.

Docker File Creation

- Create a Docker file that defines the necessary steps to build the application image.
- Specify the base image, install dependencies, and configure the container environment.

Building The Docker Image

- Build the Docker image using the Docker file.
- Include all the required application files and dependencies in the image.

Container Testing

- Run and test the container locally to ensure it functions as expected.
- Verify that the application runs within the container environment without issues.

Publishing The Docker Image

- Push the built Docker image to a container registry (e.g., Docker Hub) to make it accessible to the Kubernetes cluster.
- Ensure proper tagging and versioning of the image for easy identification.

Kubernetes provides a platform for automating the deployment, scaling, and management of containerized applications. Deploying the application to a Kubernetes cluster enables efficient orchestration and scalability.

Kubernetes Manifest

- Create Kubernetes manifest files (YAML or JSON) to define the deployment specifications.
- Specify details such as the container image, resource requirements, and desired replicas.

Deploying The Application

- Use the Kubernetes command-line interface (kubectl) or deployment tools to apply the manifest files and deploy the application.
- Verify that the application pods are running and healthy within the cluster.

Exposing The Application

- Configure a Kubernetes service to expose the application internally within the cluster.
- Define appropriate service types (ClusterIP, NodePort, LoadBalancer) based on requirements.

Verification

- Test the deployed application to ensure it functions correctly in the Kubernetes environment.

CHAPTER 5

FLOWCHART

5.1. Flowchart

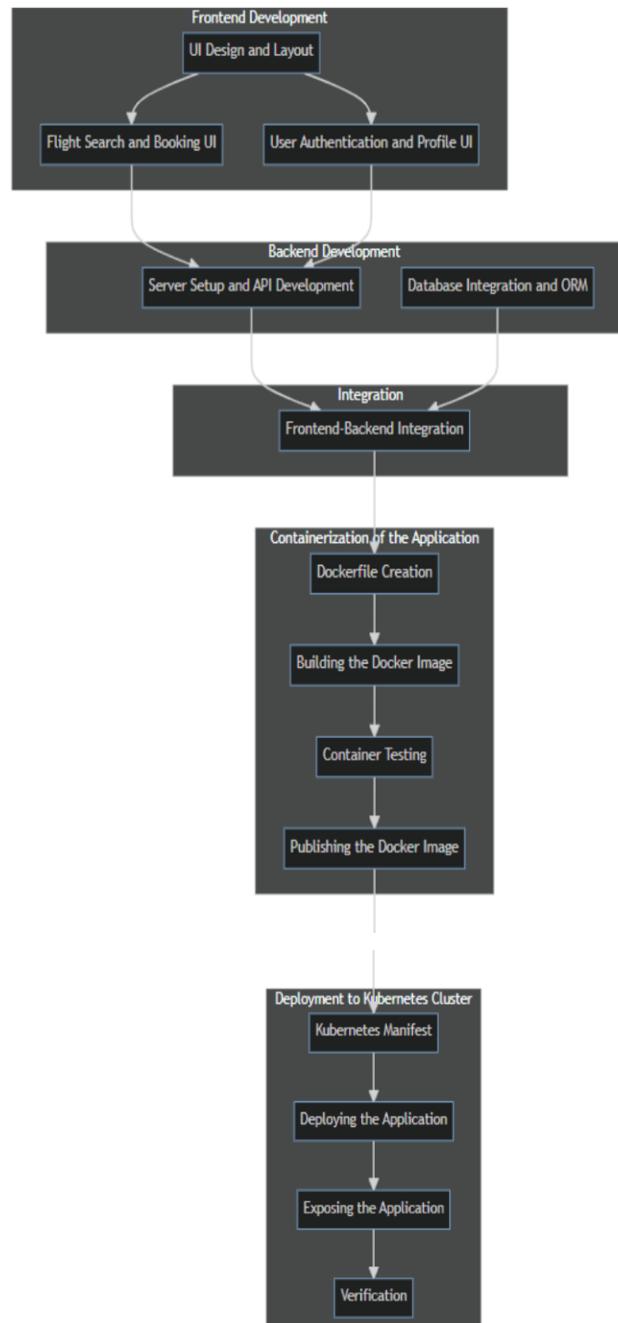


Fig 5.1 Flow Chart for the system

CHAPTER 6

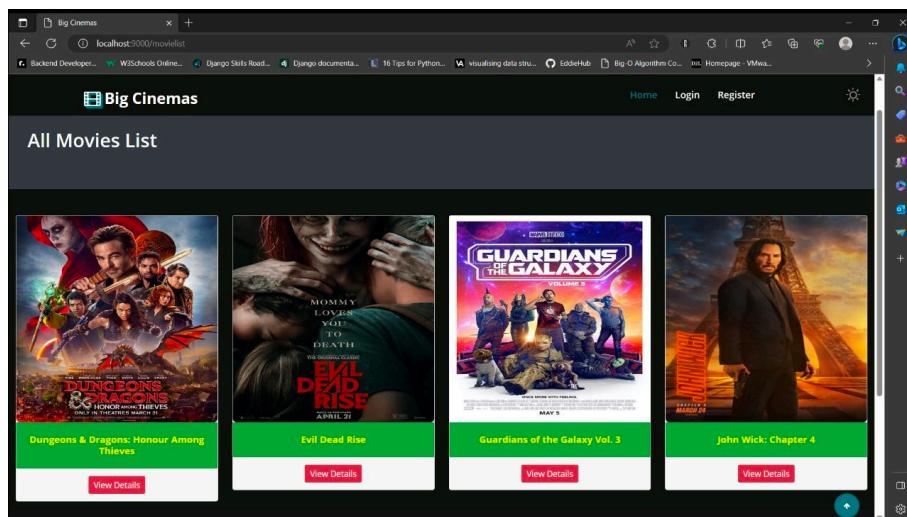
RESULT

UI Design and Layout

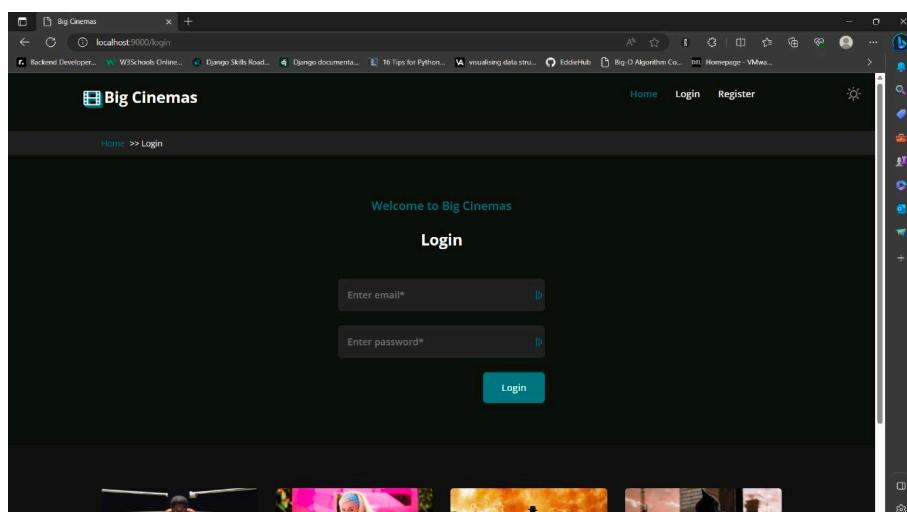
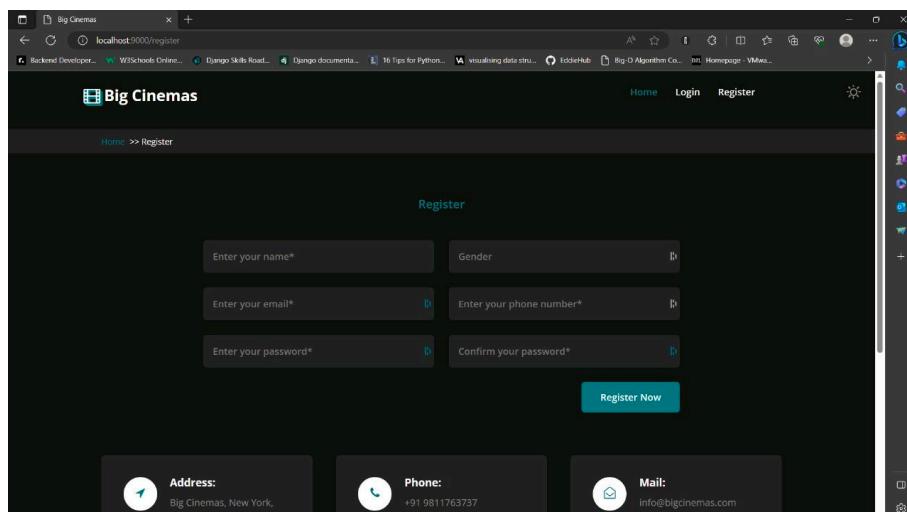
The image displays three screenshots of the Big Cinemas web application, showing different sections of the user interface.

- Screenshot 1: Home Page**
The main movie poster for "John Wick: Chapter 4" is prominently displayed. It features Keanu Reeves as John Wick in a dynamic pose against a cityscape background. Below the poster, a trailer button labeled "Watch Trailer" is visible. To the right, another movie poster for "Dungeons & Dragons: Honour Among Thieves" is shown, also with a "Watch Trailer" button. A "Top Picks" section follows, featuring movie posters for "Dungeons & Dragons: Honour Among Thieves", "Evil Dead Rise", "Guardians of the Galaxy Vol. 3", and "John Wick: Chapter 4".
- Screenshot 2: Top Picks**
This view shows the "Top Picks" section in more detail. It lists four movies: "Dungeons & Dragons: Honour Among Thieves", "Evil Dead Rise", "Guardians of the Galaxy Vol. 3", and "John Wick: Chapter 4". Each movie has a thumbnail image and a "Watch Trailer" button.
- Screenshot 3: Trending Now**
This view shows the "Trending Now" section. It lists five movies: "Guardians of the Galaxy Vol. 3", "John Wick: Chapter 4", "Dungeons & Dragons: Honour Among Thieves", "Evil Dead Rise", and "Guardians of the Galaxy Vol. 3" again. Each movie has a thumbnail image and a "Watch Trailer" button. Below this section, there are several small, horizontal movie thumbnails.

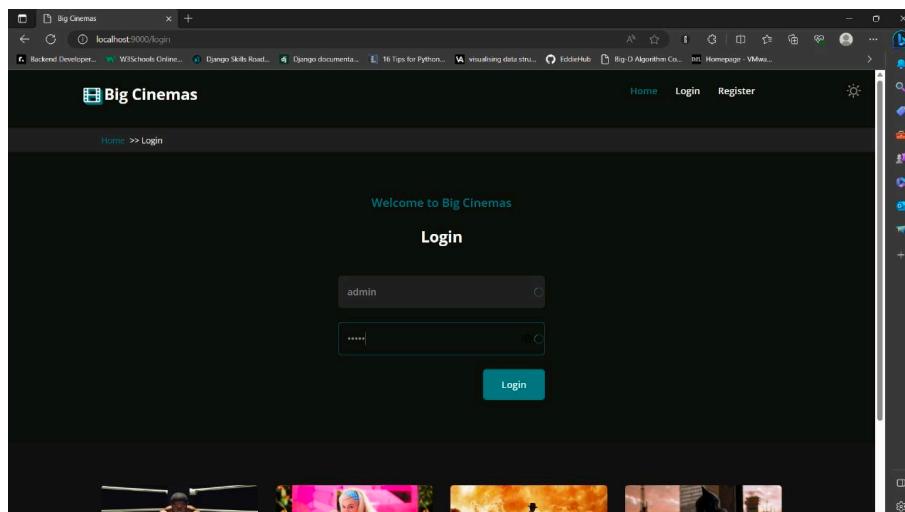
Movie Search and Booking UI



User Authentication and Profile UI



Admin Role



The screenshot shows a dark-themed web browser window titled "Movie Ticket Booking System". The address bar says "localhost:9000/movies". The left sidebar shows navigation links: "Admin Dashboard", "Movies", "Theaters", "Shows", "Customers", "Bookings", "Users", "Change Password", and "Logout". The main content area is titled "All Movies" and contains a table with the following data:

Movie ID	Movie Name	Release Date	Director	Actor	Actress	Details	Edit
4	Dungeons & Dragons: Honour Among Thieves	2023-03-30	John Francis Daley, Jonathan Goldstein	Chris Pine	Michelle Rodriguez	Details	Edit
3	Evil Dead Rise	2023-04-19	Lee Cronin	Morgan Davies	Lily Sullivan	Details	Edit
2	Guardians of the Galaxy Vol. 3	2023-05-05	James Gunn	Chris Pratt	Zoe Saldana	Details	Edit
1	John Wick: Chapter 4	2023-03-24	Chad Stahelski	Keanu Reeves	Rina Sawayama	Details	Edit

The screenshot shows a dark-themed web browser window titled "Movie Ticket Booking System". The address bar says "localhost:9000/dashboard". The left sidebar shows navigation links: "Admin Dashboard", "Movies", "Theaters", "Shows", "Customers", "Bookings", "Users", "Change Password", and "Logout". The main content area is titled "Owner Dashboard" and displays four cards with the following information:

Theaters	Movies	Customers	Bookings
2	4	1	1

Movie Ticket Booking System

Welcome, Admin.

[Admin Dashboard](#)

- [Movies](#)
- [Theaters](#)
- [Shows](#)
- [Customers](#)
- [Bookings](#)
- [Users](#)
- [Change Password](#)
- [Logout](#)

Shows

Show ID	Show Name	Show Time	End Time
1	Morning	09:30	12:30
2	Matinee	13:00	04:00
3	Evening	06:00	09:00
4	Night	10:00	12:30

Add Show

Movie Ticket Booking System

Welcome, Admin.

[Admin Dashboard](#)

- [Movies](#)
- [Theaters](#)
- [Shows](#)
- [Customers](#)
- [Bookings](#)
- [Users](#)
- [Change Password](#)
- [Logout](#)

Bookings

Booking ID	Customer	Movie	Theatre	Show	Seats	Seat Numbers	Amount	Booking Date
1	Sharann	John Wick: Chapter 4	PVR Cinemas	Morning	3	A3,A4,A5	750	2023-06-28

Movie Ticket Booking System

Welcome, Admin.

[Admin Dashboard](#)

- [Movies](#)
- [Theaters](#)
- [Shows](#)
- [Customers](#)
- [Bookings](#)
- [Users](#)
- [Change Password](#)
- [Logout](#)

Theatres

Theatre ID	Theatre Name	Theatre Location	Capacity	Ticket Price
1	PVR Cinemas	123 Street	100	250
2	Tagore Theatre	ABC Street	60	150

Add New Theatre

Theatre Name:

Theatre Address:

Theatre Seat Capacity:

Ticket Price:

[Save Theatre](#)

Customers				
Cust ID	Customer Name	Phone Number	Email ID	Gender
1	Sharann	123456789	internetbeing03@gmail.com	Male

All Users			
User ID	User Name	User Type	Password
admin	Administrator	Admin	admin
internetbeing03@gmail.com	Sharann	Customer	123

User Role

Welcome to Big Cinemas

Login

xyz@gmail.com

Login

Big Cinemas

localhost:9000

Home My Bookings Welcome, Sharann. Logout

Dungeons & Dragons: Honour Among Thieves

A charming thief and a band of unlikely adventurers undertake an epic heist to retrieve a lost relic, but things go dangerously awry when they run afoul of the wrong people. *Dungeons & Dragons: Honour Among Thieves* brings the rich world and playful spirit of the legendary roleplaying game to the big screen in a hilarious and action-packed adventure.

MARCH 24

IMAX 4DX MAX LIONSGATE

Watch Trailer

Evil Dead Rise

Moving the action out of the "Evil Dead" tells a twisted tale of sisters, played by Sutton Foster and Rose McIver, who reunite after years apart. Their reunion is cut short by the return of demons, thrusting them into survival as they face the most horrific family imaginable.

Watch Trailer

Top Picks

Show all

Big Cinemas

localhost:9000/payment/2

Home My Bookings Welcome, Sharann. Logout

Confirm Ticket

Name on Card

CVV

Card Number

Expiration Date

january 2021

VISA MasterCard AMEX

Confirm Order

Big Cinemas

localhost:9000/showticket

Home My Bookings Welcome, Sharann. Logout

Print

Booking Ticket

Booking Id	2
Customer Id	Sharann
Theatre Name	Tagore Theatre
Show Information	Night (10:00-12:30)
Booking Date	2023-06-28
No of Seats	3
Seats	C6,C7,C8
Amount	₹ 450

Big Cinemas

localhost:9000/mdetails/2

Home My Bookings Welcome, Sharann. Logout

Displaying Details

Movie Name: Guardians of the Galaxy Vol. 3

Director: James Gunn

Actor: Chris Pratt

Actress: Zoe Saldana

Description: Still reeling from the loss of Gamora, Peter Quill must rally his team to defend the universe and protect one of their own. If the mission is not completely successful, it could possibly lead to the end of the Guardians as we know them.

Choose Date: 28-06-2023

Choose Theatre: Tagore Theatre

Choose Show: Night

Book Now

Big Cinemas

localhost:9000/seatbook

Home My Bookings Welcome, Sharann. Logout

Movie : Guardians of the Galaxy Vol. 3

Available Selected Occupied

	1	2	3	4	5	6	7	8	9	10
A										
B										
C										
D										
E										
F										

Confirm Now

Booking Information

Booking Date: 2023-06-28

Theatre Name: Tagore Theatre

Show Time: Night

Per Seat Ticket Price: 150

Number of Seats: 3

Seat Numbers: C6,C7,C8

Book Now

Big Cinemas

localhost:9000/history

Home My Bookings Welcome, Sharann. Logout

Booking History

Booking ID	Movie Name	Theatre Name	No. of Tickets	Seat Numbers	Amount	Booking Date
2	Guardians of the Galaxy Vol. 3	Tagore Theatre	3	C6,C7,C8	₹450	2023-06-28
1	John Wick: Chapter 4	PVR Cinemas	3	A3,A4,A5	₹750	2023-06-28

Server Setup and API Development

controllers	24-06-2023 18:33	File folder
models	24-06-2023 18:33	File folder
repository	24-06-2023 18:33	File folder
services	24-06-2023 18:33	File folder
MovieBookingApplication.java	24-06-2023 18:33	Java Source File 1 KB

```
application.properties | Movie-Booking/pom.xml
1 spring.mvc.view.prefix=/WEB-INF/views/
2 spring.mvc.view.suffix=.jsp
3
4
5 spring.datasource.url=jdbc:mysql://localhost/moviebooking?createDatabaseIfNotExist=true&useUnicode=true
6 spring.datasource.username=root
7 spring.datasource.password=Akshaj12345
8 spring.jpa.show-sql=true
9 spring.jpa.hibernate.ddl-auto=update
10 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect
11
12 spring.mail.host=smtp.gmail.com
13 spring.mail.port=587
14 spring.mail.username=akshajnevgi@gmail.com
15 spring.mail.password=password
16 spring.mail.properties.mail.smtp.auth=true
17 spring.mail.properties.mail.smtp.starttls.enable=true
```

```
application.properties | Movie-Booking/pom.xml
https://maven.apache.org/xsd/maven-4.0.0.xsd (xs:i:schemaLocation)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>2.5.7</version>
9     <relativePath/> 
10  </parent>
11  <groupId>com.movie</groupId>
12  <artifactId>Movie-Booking</artifactId>
13  <version>0.0.1-SNAPSHOT</version>
14  <name>Movie-Booking</name>
15  <description>Hospital Backend</description>
16  <properties>
17    <java.version>1.8</java.version>
18  </properties>
19  <dependencies>
20    <dependency>
21      <groupId>org.springframework.boot</groupId>
22      <artifactId>spring-boot-starter-data-jpa</artifactId>
23    </dependency>
24    <dependency>
25      <groupId>org.springframework.boot</groupId>
26      <artifactId>spring-boot-starter-web</artifactId>
27    </dependency>
28    <dependency>
29      <groupId>org.springframework.boot</groupId>
30      <artifactId>spring-boot-devtools</artifactId>
31      <scope>runtime</scope>
32      <optional>true</optional>
33    </dependency>
34    <dependency>
35      <groupId>mysql</groupId>
```

Database Integration and ORM



1 • SELECT * FROM moviebooking.bookings;

Result Grid

bid	amount	bdate	noseat	seats	custid	movieid	showid	theatreid
1	750	2023-06-28	3	A3,A4,A5	1	1	1	1
2	450	2023-06-28	3	C6,C7,C8	1	2	4	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

bookings1 x Apply Revert

1 • SELECT * FROM moviebooking.customer;

Result Grid

id	email	gender	phone	uname
1	internetbeing03@gmail.com	Male	123456789	Sharann
*	NULL	NULL	NULL	NULL

customer 2 x Apply Revert

1 • SELECT * FROM moviebooking.movie;

Result Grid

mid	actor	actress	banner	desc
1	Keanu Reeves	Rina Sawayama	banners/ea187659-94a6-4f48-a601-be6146e1...	John Wick (Keanu Reeves) uncovers a path to ...
2	Chris Pratt	Zoe Saldana	banners/5c2ac24a-638f-42ce-8f43-ac5e8c35e8...	Still reeling from the loss of Gamora, Peter Quill ...
3	Morgan Davies	Lily Sullivan	banners/cf97c82f-4bd1-495b-9ee0-18ac6c1410...	Moving the action out of the woods and into th...
4	Chris Pine	Michelle Rodriguez	banners/5d9d2643-e840-4649-955d-101a212e...	A charming thief and a band of unlikely adventu...
*	NULL	NULL	NULL	NULL

movie 3 x Apply Revert

1 • SELECT * FROM moviebooking.theatre;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

tid	address	price	seats	tname
1	123 Street	250	100	PVR Cinemas
2	ABC Street	150	60	Tagore Theatre
*	NULL	NULL	NULL	NULL

theatre 4 × Apply Revert

1 • SELECT * FROM moviebooking.theatre_show;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

id	endtime	showname	starttime
1	12:30	Morning	09:30
2	04:00	Matinee	13:00
3	09:00	Evening	06:00
4	12:30	Night	10:00
*	NULL	NULL	NULL

theatre_show 5 × Apply Revert

1 • SELECT * FROM moviebooking.user;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

userid	id	pwd	role	uname
admin	0	admin	Admin	Administrator
internetbeing03@gmail.com	1	123	Customer	Sharann
*	NULL	NULL	NULL	NULL

user 6 × Apply Revert

Docker Image Creation

```
1 FROM openjdk:8
2 EXPOSE 8080
3 ADD target/movie-booking-docker.jar movie-booking-docker.jar
4 ENTRYPOINT ["java","-jar","/movie-booking-docker.jar"]
```

```
PS C:\Users\hp\Onedrive\Desktop\movie-booking3> ls

Directory: C:\Users\hp\Onedrive\Desktop\movie-booking3

Mode                LastWriteTime       Length Name
----                -----        ----  --
da---l             30-06-2023 14:30          .mvn
da---l             30-06-2023 14:30          .settings
da---l             30-06-2023 14:30          src
da---l             30-06-2023 14:59          target
-a---l             24-06-2023 18:33         1268 .classpath
-a---l             24-06-2023 18:33         395 .gitignore
-a---l             24-06-2023 18:33         1132 .project
-a---l             30-06-2023 14:58         145 Dockerfile
-a---l             24-06-2023 18:33         1286 HELP.md
-a---l             24-06-2023 18:33         10070 mvnw
-a---l             24-06-2023 18:33         6608 mvnw.cmd
-a---l             30-06-2023 14:58         2245 pom.xml

PS C:\Users\hp\Onedrive\Desktop\movie-booking3> docker build -t movie-booking-docker.jar .
[+] Building 3.3s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 184B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/openjdk:8
=> [auth] library/openjdk:8 pull token for registry-1.docker.io
=> [internal] load build context
=> => transferring context: 44.99MB
=> CACHED [1/2] FROM docker.io/library/openjdk:8@sha256:8e863cc57215cfb181bd319736d0baef625fe8f150577f9eb58bd937 0.0s
=> [2/2] ADD target/movie-booking-docker.jar movie-booking-docker.jar
=> exporting to image
=> => exporting layers
=> => writing image sha256:0cb5e6c068a9b5bbcffe2e4a389632a89ec7cdc802612edcf36865a2415cbaa8 0.0s
=> => naming to docker.io/library/movie-booking-docker.jar
PS C:\Users\hp\Onedrive\Desktop\movie-booking3>
```

```
Windows PowerShell x + PS C:\Users\hp\Onedrive\Desktop\movie-booking3> docker image ls
REPOSITORY           TAG      IMAGE ID      CREATED          SIZE
movie-booking-docker.jar  latest   0cb5e6c068a9  About a minute ago  570MB
PS C:\Users\hp\Onedrive\Desktop\movie-booking3>
```

Pushing Image to dockerhub

```
Windows PowerShell x + PS C:\Users\hp\Onedrive\Desktop\movie-booking3> docker push sharann18/movie-booking-docker.jar
Using default tag: latest
The push refers to repository [docker.io/sharann18/movie-booking-docker.jar]
437e1ed1cb64: Pushed
6b5aafff44254: Pushed
53a0b163e995: Pushed
b626401ef603: Pushed
9b55156abf26: Pushed
293d5db30c9f: Pushed
03127cdb479b: Pushed
9c742cd6c7a5: Pushed
latest: digest: sha256:85b74b36ad952835516bc65e67bfd36d57e6da14965a110da66235212329376d size: 2007
PS C:\Users\hp\Onedrive\Desktop\movie-booking3>
```

CHAPTER 7

ADVANTAGES AND DISADVANTAGES

Advantages:

1. Scalability: Java Spring Boot and MySQL are highly scalable technologies, allowing the movie ticket booking system to handle a large number of concurrent users and transactions effectively.
2. Robustness: Java Spring Boot provides a robust and reliable framework for developing web applications. It offers built-in error handling, exception management, and security features, ensuring the system's stability and reliability.
3. Flexibility: Java Spring Boot allows for easy integration with various third-party libraries, APIs, and frameworks, enabling the movie ticket booking system to incorporate additional functionalities and adapt to changing requirements.
4. Security: Java Spring Boot provides numerous security features, including authentication, authorization, and data encryption, which are crucial for protecting sensitive user information and preventing unauthorized access.
5. Performance: By leveraging the power of Java and the efficient database management capabilities of MySQL, the movie ticket booking system can deliver high-performance and low-latency responses, ensuring a smooth user experience.

Disadvantages:

1. Learning Curve: Java Spring Boot can have a steep learning curve for developers who are new to the framework. It may take time and effort to understand the concepts, configurations, and best practices, especially for inexperienced developers.
2. Complexity: Developing a movie ticket booking system using Java Spring Boot and MySQL involves handling multiple components, such as controllers, services, repositories, and database queries. Managing the complexity of the system may require advanced software engineering skills.
3. Scalability Limitations: While Java Spring Boot and MySQL can handle a significant amount of traffic, extremely high-demand scenarios may require additional scaling techniques, such as load balancing or distributed architectures, to ensure optimal performance.
4. Database Management: Although MySQL is a reliable and widely used database management system, it may require proper optimization and configuration to handle large datasets efficiently. Inadequate database design or queries could impact performance.
5. Maintenance: Java Spring Boot and MySQL require ongoing maintenance and updates to address security vulnerabilities, bug fixes, and performance improvements. Regular monitoring and upgrades are necessary to keep the system up-to-date.

CHAPTER 8

APPLICATIONS

Systems built using Java Spring Boot and MySQL have a wide range of applications across various industries. Some common applications include:

1. E-commerce Platforms: Java Spring Boot and MySQL can be used to develop robust e-commerce platforms that handle product catalog management, order processing, payment integration, user authentication, and inventory management.
2. Content Management Systems (CMS): Content management systems, such as blogs, news portals, or website builders, can be built using Java Spring Boot and MySQL to handle content creation, editing, publishing, and user management.
3. Customer Relationship Management (CRM) Systems: CRM systems that manage customer data, track interactions, and automate sales and marketing processes can be developed using Java Spring Boot and MySQL.
4. Inventory Management Systems: Java Spring Boot and MySQL can be utilized to build inventory management systems that track stock levels, manage purchase orders, handle suppliers, and generate reports on inventory performance.
5. Human Resource Management Systems (HRMS): HRMS applications that handle employee records, payroll management, leave management, and performance evaluation can be developed using Java Spring Boot and MySQL.
6. Booking and Reservation Systems: Systems for booking and reservations, such as hotel booking systems, restaurant reservation systems, or event ticketing systems, can be built using Java Spring Boot and MySQL to handle availability, booking, and payment processing.
7. Financial Management Systems: Java Spring Boot and MySQL can be employed to develop financial management systems for budgeting, expense tracking, invoicing, and financial reporting.
8. Healthcare Information Systems: Healthcare applications, including electronic medical records (EMR), hospital management systems, or telemedicine platforms, can be developed using Java Spring Boot and MySQL to manage patient information, appointments, and medical data.
9. Educational Systems: Java Spring Boot and MySQL can be used to create learning management systems (LMS), student information systems (SIS), or online course platforms that handle course enrolment, grading, content delivery, and student management.
10. Social Media Platforms: Social media platforms, such as community forums or social networking sites, can be developed using Java Spring Boot and MySQL to handle user profiles, friend connections, content sharing, and real-time interactions.

CHAPTER 9

CONCLUSION AND FUTURE WORK

This project develops a movie ticketing system based on the SpringBoot and JSP. The system is developed in front and back ends separately. The front-end uses JSP, the back-end uses the SpringBoot framework, and the MySql database is used for data storage. The SpringBoot framework used in this system simplifies system configuration and facilitates development. The system implements functions including login, registration, ticket booking, order query, film arrangement query, movie query, order management, film arrangement management, user query, order cancellation, data statistics, and film management, etc., which is convenient for cinema staff and the general public to use. Now a days the traditional booking of movies is very backdated. People are more into online things. People need easy solution.

They are controlled by the internets through this online booking system we can make internet useful to us. Its controlling people but by using this in our day to day life to make our life easier thus we can assure that internet will not ruin our time. Yes in this online movie ticket booking system there were some lacking and limitations. But we always think about a user friendly and simple solution system and we hope the system is just like that. Future scope include making the system more user-friendly and making the system more easier to mass people.

There are several potential areas for future work and enhancements in a movie ticket booking system developed using Java Spring Boot and MySQL. Some of these areas include:

1. **Mobile Application Development:** Developing a dedicated mobile application for the movie ticket booking system can provide a more convenient and accessible user experience. The mobile app can leverage the system's backend built with Java Spring Boot and MySQL to offer features such as ticket booking, seat selection, push notifications, and user-friendly interfaces.
2. **Integration with Third-Party APIs:** Enhancing the system's functionality by integrating with third-party APIs can offer additional features and services. For example, integrating with payment gateways, movie databases, or location-based services can provide seamless payment processing, access to comprehensive movie information, and personalized recommendations based on the user's location.
3. **Real-Time Seat Availability:** Implementing a real-time seat availability feature can allow users to view the current status of seats in a theater. This can help users make informed decisions about seat selection and improve the overall booking experience.
4. **Personalization and Recommendation Engine:** Incorporating a recommendation engine based on user preferences, viewing history, and ratings can enhance the system's ability to suggest movies, genres, or theaters that align with the user's interests. This personalization can improve user engagement and satisfaction.
5. **Social Media Integration:** Integrating the movie ticket booking system with social media platforms can enable users to share their movie choices, reviews, or ticket purchases with their friends. This can enhance the social aspect of the platform and potentially drive user acquisition through word-of-mouth marketing.

6. Analytics and Reporting: Implementing analytics and reporting capabilities within the system can provide valuable insights into user behavior, ticket sales, popular movies, and revenue generation. This data can be utilized for business decision-making, marketing strategies, and improving overall system performance.
7. Advanced Booking Options: Introducing advanced booking options such as group bookings, premium seating, or special packages can cater to different user preferences and increase revenue streams.
8. User Feedback and Ratings: Incorporating a feedback and rating system can allow users to provide reviews, ratings, and feedback on movies, theaters, and their overall experience. This feedback can be used to improve the system's offerings and provide valuable insights to other users.
9. Integration with Loyalty Programs: Integrating the movie ticket booking system with loyalty programs or reward systems can incentivize users to book tickets and engage with the platform regularly. This can help in building customer loyalty and increasing user retention.
10. Integration with Digital Ticketing Standards: Adopting digital ticketing standards, such as QR codes or mobile ticketing, can enhance the ticket validation process and improve efficiency at theaters.

CHAPTER 10

REFERENCES

- [1] Zhang, Xiaolin. "Design and Implementation of Cinema ticketing System Based on Spring Boot and Vue." Scientific Journal of Intelligent Systems Research Volume 4.5 (2022).
- [2] Tazin, Nur-E. Online Movie Ticket booking System. Diss. United International University, 2018.
- [3] Shroff, Anshul, et al. "Multiplex Regulation System with Personalised Recommendation Using ML." 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2021.
- [4] Roy, Archit, Vinit Shahdeo, and Rajesh Kaluri. "A Comparative Study in Online Movie Ticket Booking System." Research Journal of Engineering and Technology 10.1 (2019): 16-20.
- [5] Archit Roy, Vinit Shahdeo, Rajesh Kaluri. A Comparative Study in Online Movie Ticket Booking System. Research J. Engineering and Tech. 2019;10(1):16-20. doi: 10.5958/2321-581X.2019.00004.7
- [6] Oludele, Awodele, et al. "Application software for graphical interface online theatre booking system." International Journal of Computer Trends and Technology 1.1 (2011): 61-69.
- [7] Islam, Gazi Zahirul, et al. "Implementation of an efficient web-based movie ticket purchasing system in the context of Bangladesh." vol 19 (2020): 828-836.
- [8] Sarkar, Punyaslok & Noel, Mrs. (2020). A Project on Online Ticket Booking System

GitHub Link:

<https://github.com/Akshaj-N/Movie-Ticket-Booking.git>

Project Demo Link:

https://drive.google.com/file/d/1ia7gm0BrNSNiMI1BOkq-ZRLVLKDz45GC/view?usp=drive_link