

ONLINE MOVIE TICKET BOOKING

SUBMITTED BY

PROJECT ASSOCIATE NAME

RAJDEV KUMAR (Team Leader)

VASANTKUMAR K S

RAMESH S N

HEMANTH KANDATI

BOBIN S

SHARAT HIREMATH

ONLINE MOVIE TICKET BOOKING

CONTENTS

- Abstract -----
- Introduction -----
- System Analysis -----
- Requirement Analysis -----
- System Design -----
- ER Diagram -----
- Screenshots -----
- System Implementation -----
- Future Enhancement -----
- Conclusion -----
- Reference -----

CHAPTER-1

ABSTRACT

Movie ticket booking in web system is basically made for providing the customers an anytime and anywhere service for booking cinema tickets and providing information regarding movies and their online schedule. Due to increase in technology and digitalization online movie system has become much more popular. Instead of wasting time in queues of movie counter, it provides easy and efficient way to book the tickets online. It is quite fascination to sit at home browse the movies which have been released, read the review and then book the tickets according to our feasibility. Online cinema ticket booking system is actually 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.

The main aim of the system is to provide complete information of the movie and schedule to the customer, according to which he/she can book the tickets.

The primary purpose of making the online cinema ticket booking system is to create an automatic online based System which will provide an easy and an alternate way to book a ticket for a movie. The customer will need to Enter the information related to film after which the system will provide options to book the ticket. In this Online Cinema Ticket Booking System, the staff of the cinema hall won't have to do anything for the ticket Booking as the process will be done by the customer through the web which will reduce the amount of staff Required in the booking counter.

Booking System will provide detailed information so that a customer can know about the movie and based on The information customer will book the ticket.

The customer can be presented with a booking page, which allows them to add more of the same items or remove items from the combo items. The booking page also shows the seat type to be

booked, no of seats to be booked, service charges to be applied, ticketing cost and net payable amount to the customer.

CHAPTER-2

INTRODUCTION

2.1 INTRODUCTION

With the development of technologies, increasing digitalization, and increasing social networking, information sharing on internet has become quite. As a result of that all the online system has become very popular in the last 6-10 years.

According to the largest online purchase happened in Asia Pacific. Discussions and use of online system has become much more frequent than the offline one. Movie booking system is one such trending system now a day. It provides the user to online browse and book movie tickets. With the increase in technology there has been lots of advancement in online booking system. Users are now able to see the seating plan and they can also book refreshments online. There are also systems through which users can confirm their ticket and pay while entering to the hall.

Digitalization is one of the most amazing outcomes of technology. It has given easy accessibility of all the resources available. Wasting time in the queues of movie theatre has been ruled out due to online system. Also, reviews can be easily read online so that user can get a clear-cut picture about the movie. Online payment has also increased E-Banking which is helpful for the sound development of countries economy.

Movie ticket booking in web is the process by which customer can purchase their movie tickets directly using internet and pay through online banking. Online movie ticket booking system is measurable, cost effective and has very good user interface.

This web based Angular, Spring boot, MySQL project provides the all working of cinema ticket booking system. Movie ticket booking in web is useful for both promoters and customers equally.

Customers can make online movie ticket booking at any instance of 24 hours a day. And as this is a web based application therefore they can buy ticket from anywhere around the world. Customers can buy tickets without wasting their time.

It includes is a user interface specification whereby the user can demonstrate interface standards to be used in designing the system. Furthermore, considerations regarding non-functional requirements and system evolution are addressed.

2.2 PURPOSE

Some important goals of movie ticket booking are as follows:

- Online cinema ticket booking system project is aimed to provide facility to book cinema tickets anytime and from anywhere.
- One objective of this project is to minimize the no. of staffs at the ticket-box window.
- Promote new movies over the internet and gain maximum profit.
- Provide a 24x7 service to the customer.

CHAPTER-3

SYSTEM ANALYSIS

3.1 ANALYSIS MODEL

The model that is basically being followed is the WATER FALL MODEL, which states that the phases are organized in a linear order. First of all, the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model. The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are: -

- Requirement Analysis
- Project Planning
- System design
- Detail design
- Coding
- Unit testing
- System integration & testing

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done.

WATER FALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

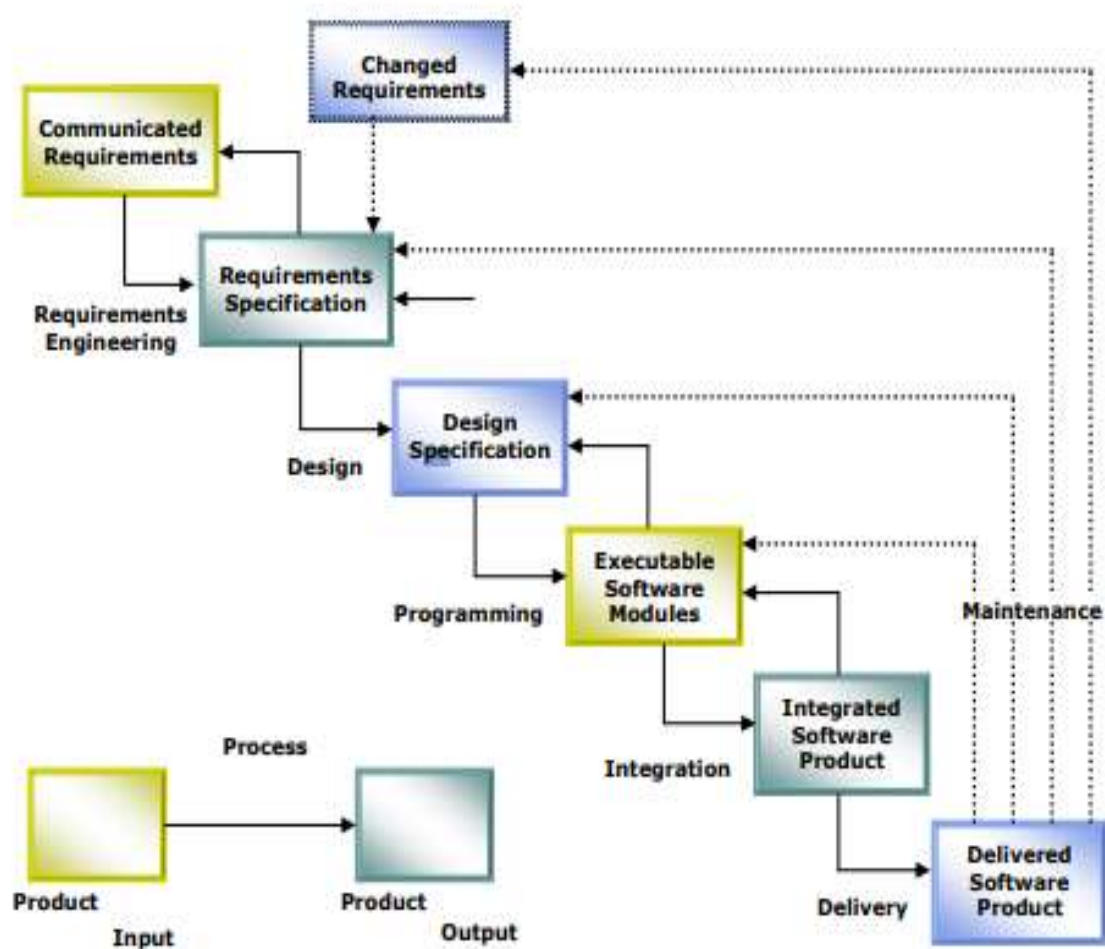





Fig: Water Fall Model

NUMBER OF MODULES

The system after careful analysis has been identified to be presented with the following modules:

The modules involved are:

- **Movie (Admin)**
 -  **Add**
 -  **Update**
 -  **Delete**
 -  **Customer Records**
- **Movie (Customer)**
 -  **Update Profile**
 -  **View Movies**
 -  **Book Ticket**
 -  **Contact Us**
- **Registration**
- **Login**

CHAPTER -3

Requirement Analysis

3.1 Requirement Analysis

Software Requirements

- Apache tomcat server 0.9
- Visual Studio
- MySQL
- Spring Tool Suite

Hardware requirements

- Operating System - Windows XP, Windows 7, Windows 10 or higher end.
- Processor - Intel Dual Core or higher version RAM - Minimum 2 GB
- Hard Disk - Minimum 40 GB

CHAPTER -4

System Design

4.1 Introduction for System Design

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement has been specified and analysed, system design is the first of the three technical activities - design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

CHAPTER-5

Web App Diagram

5.1 ER Diagram

Registration Flow chat

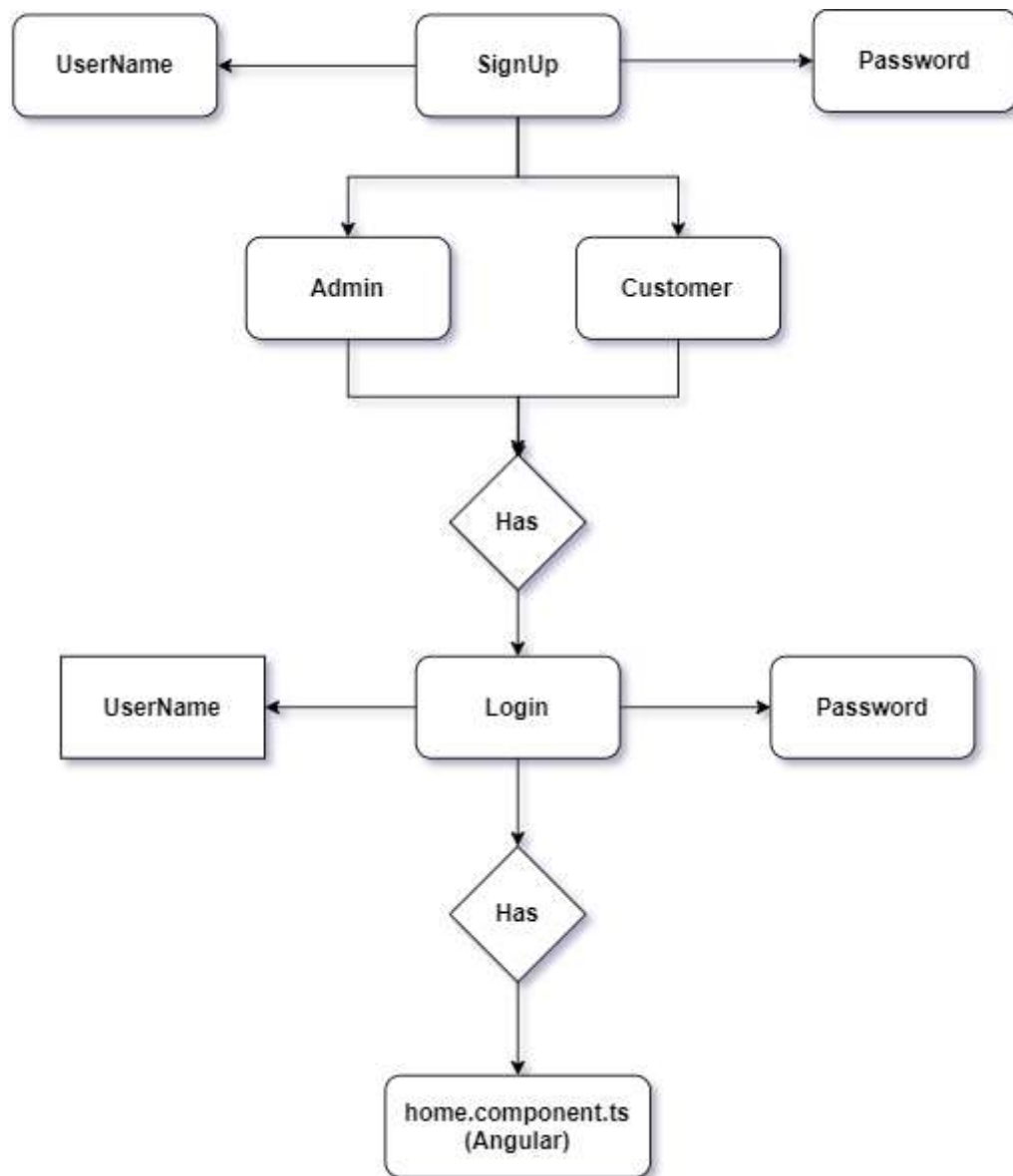


Fig (1)

5.2 Diagram structure

Customer Diagram

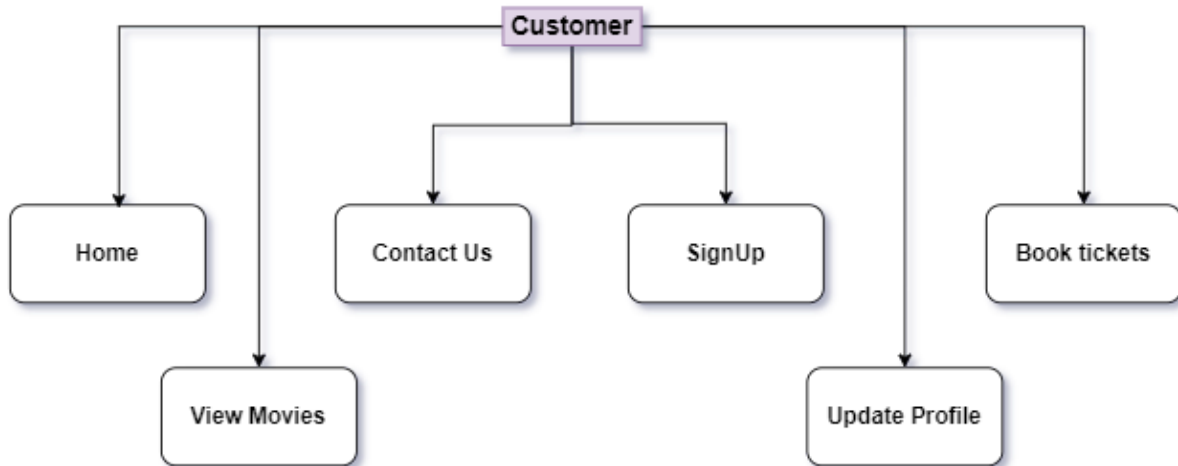


Fig (2)

Admin Diagram

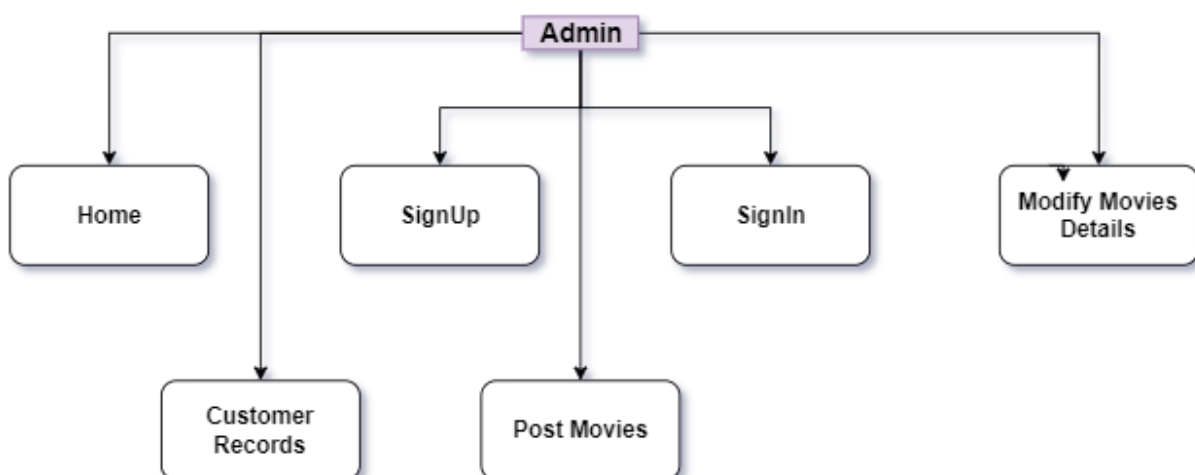


Fig (3)

Post Movie Diagram

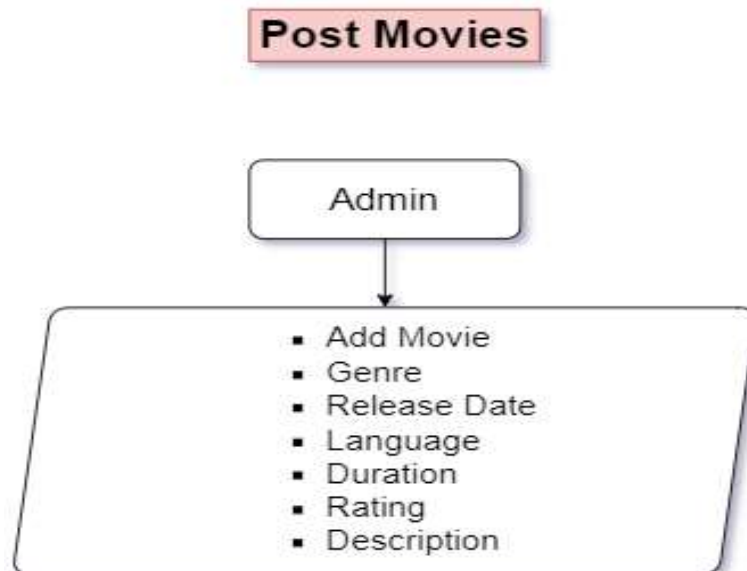


Fig (4)

View Movie

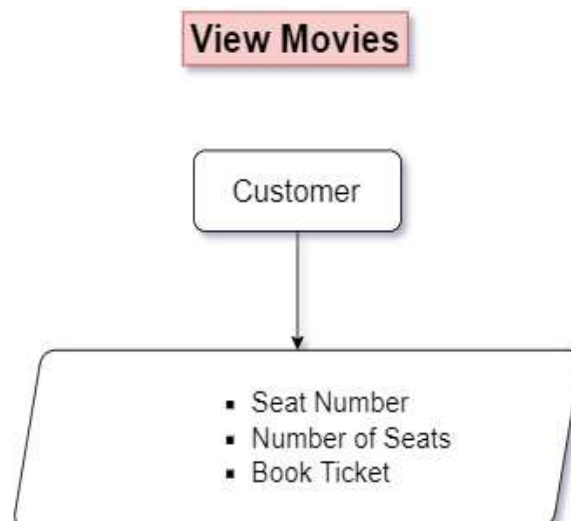


Fig (5)

Flow of Project Diagram

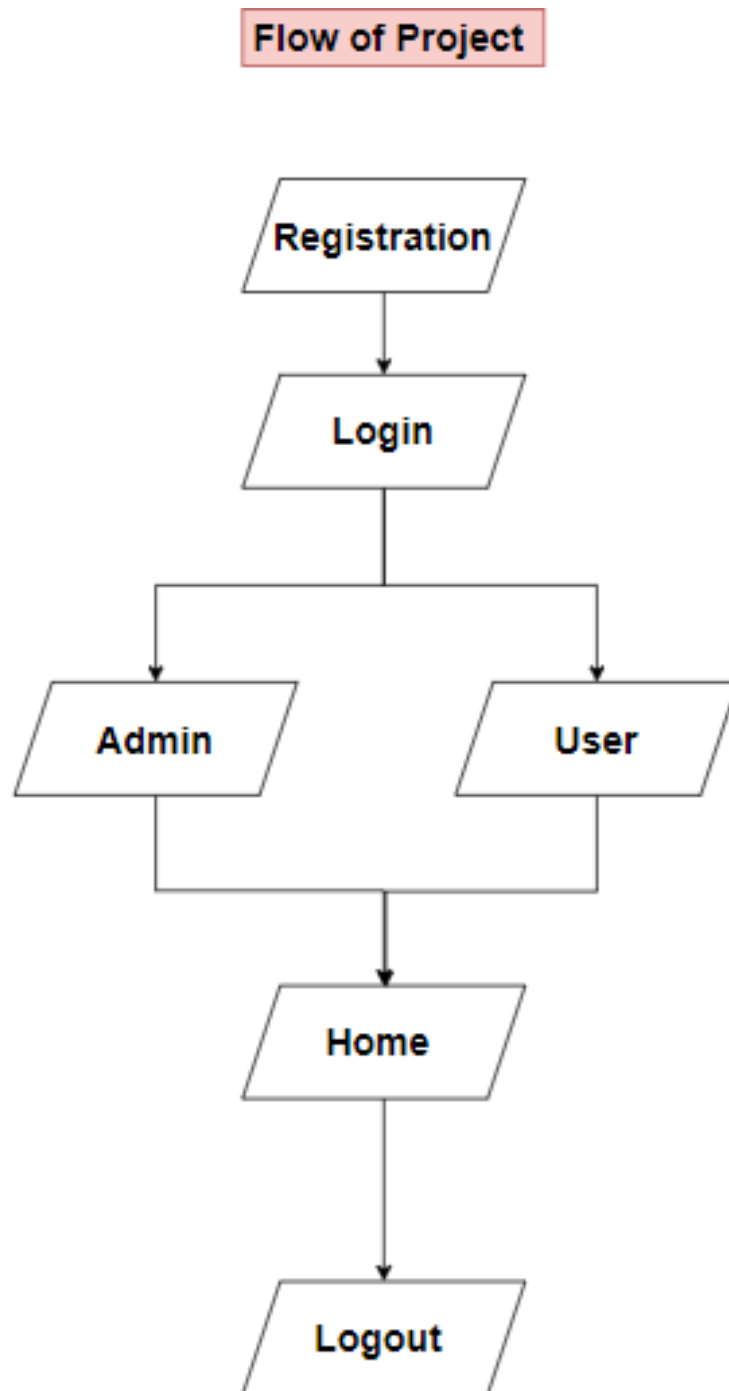
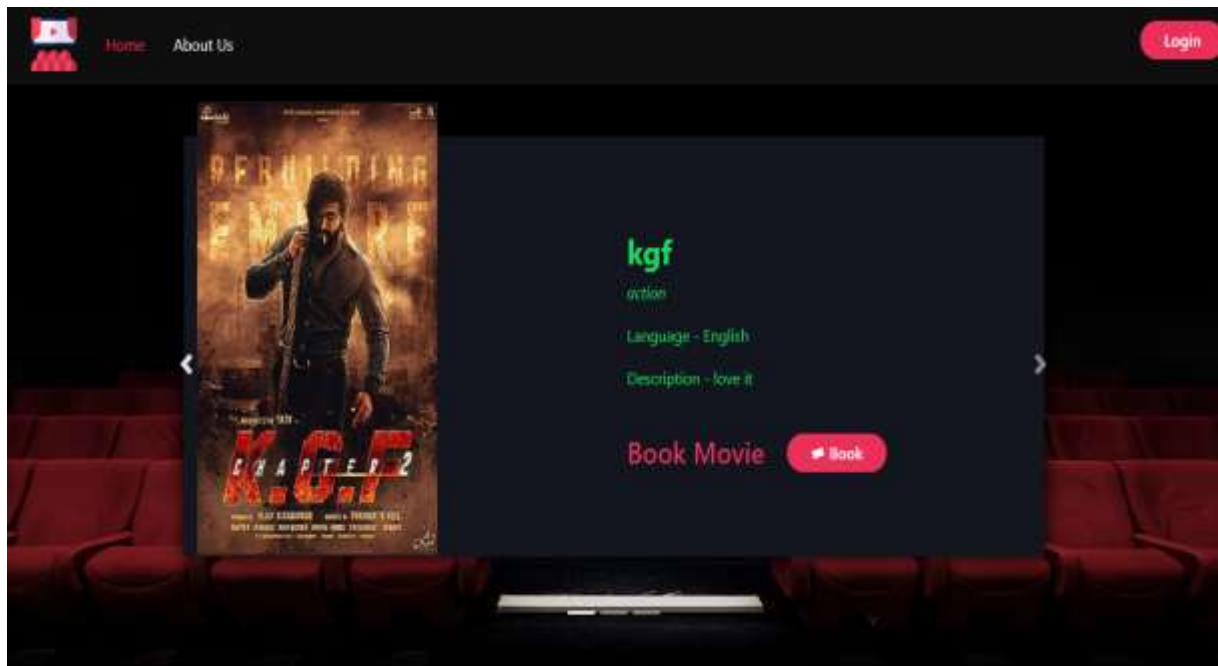


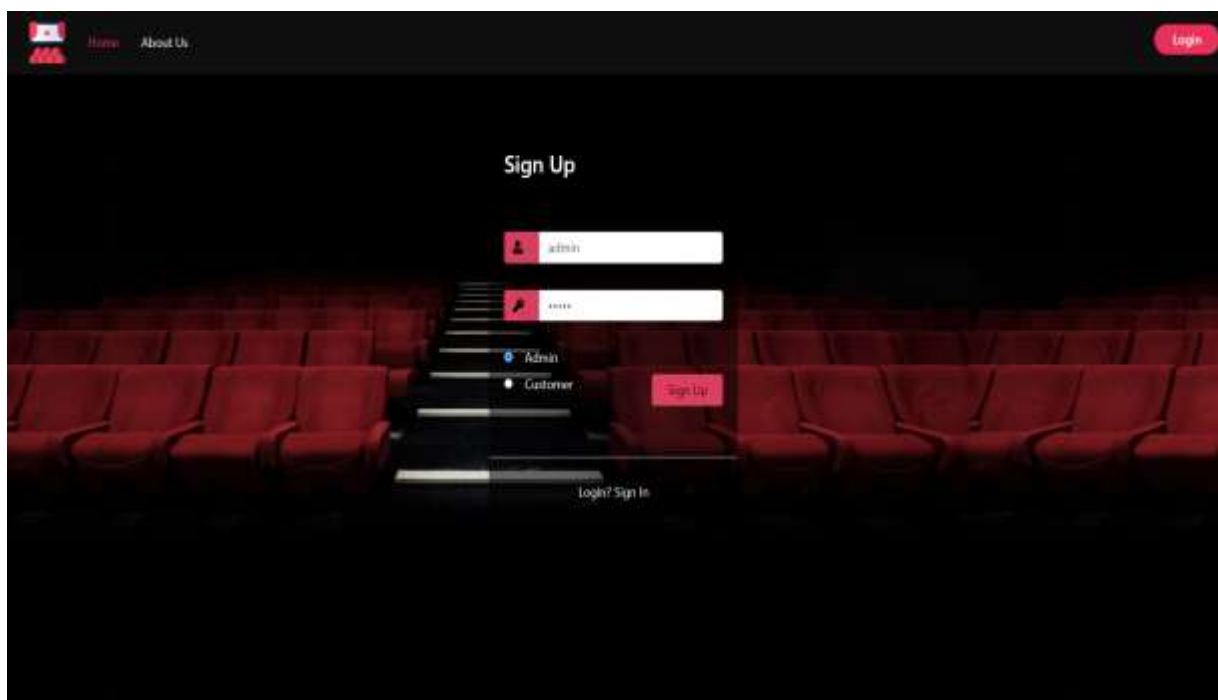
Fig (6)

CHAPTER-6

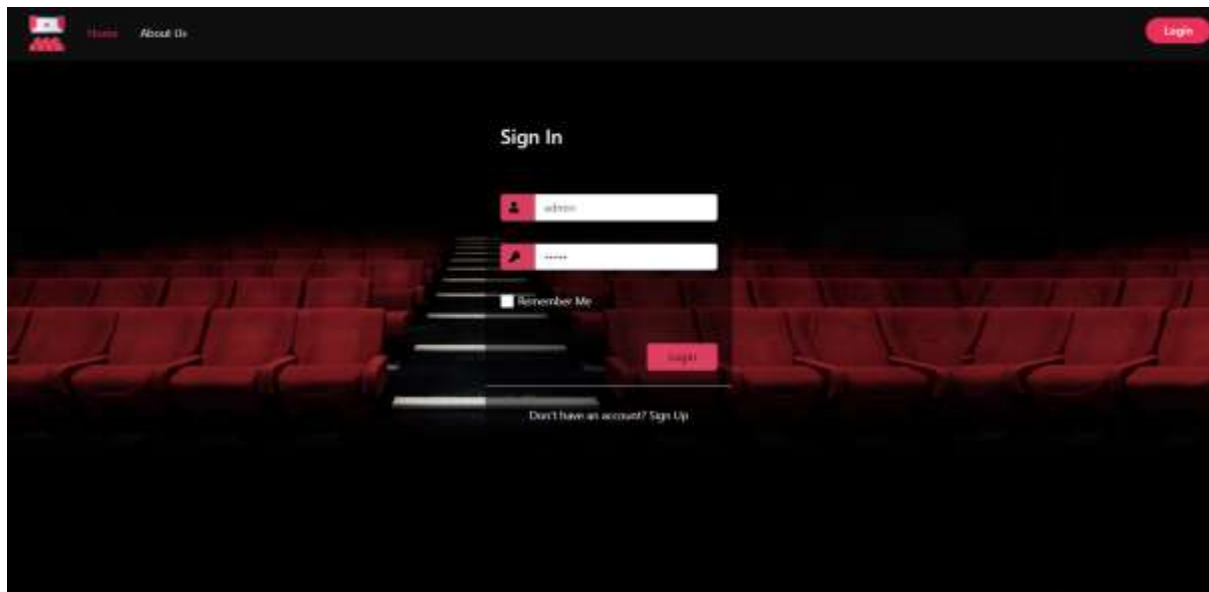
Screen Shots



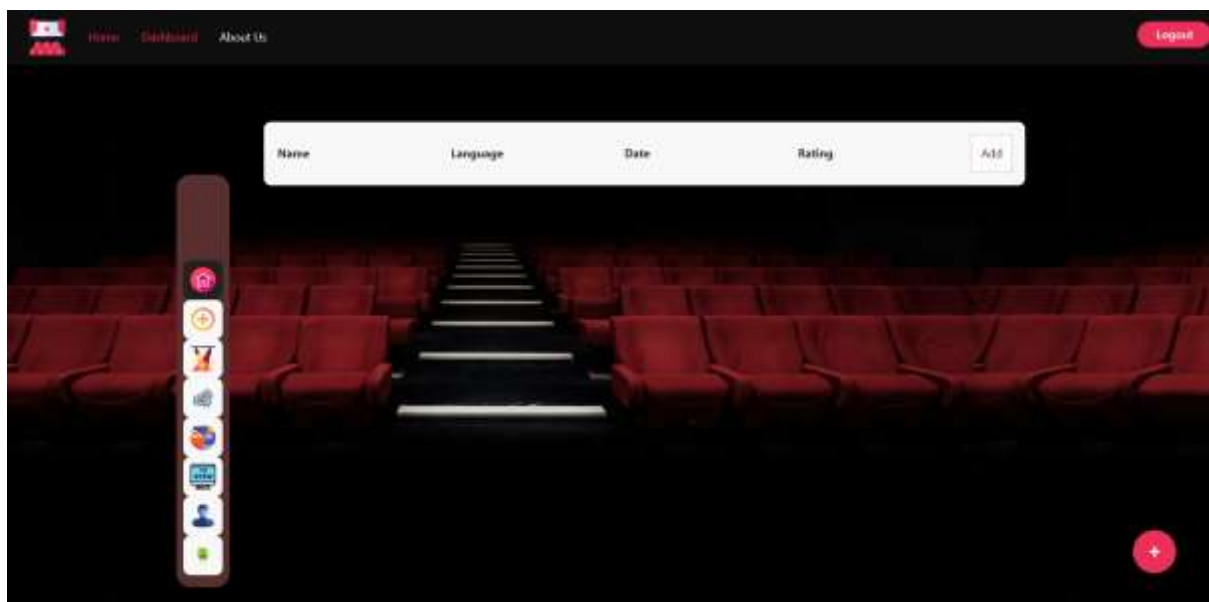
Home/Index Page



Registration Page



Login Page



Admin Page

Add Movie

Movie Name

SDP

Movie Genre

ACTION

Movie Language

ENGLISH

Movie Release

25

Movie Rating

9

Movie Description

IT IS AN BEST ACTION MOVIE

01-07-2022

Add Movie

Adding Movie Page

Theatre Name

Krishna

Theatre City

Mysore

Manager Contact

9886456878

Add Theatre

Adding Theatre Details

Screen Name

SCREEN 1

Rows

05

Columns

07

Add Screen

Adding Screen

Show Name

FanShow

01-07-2022



01-07-2022 10:00 AM

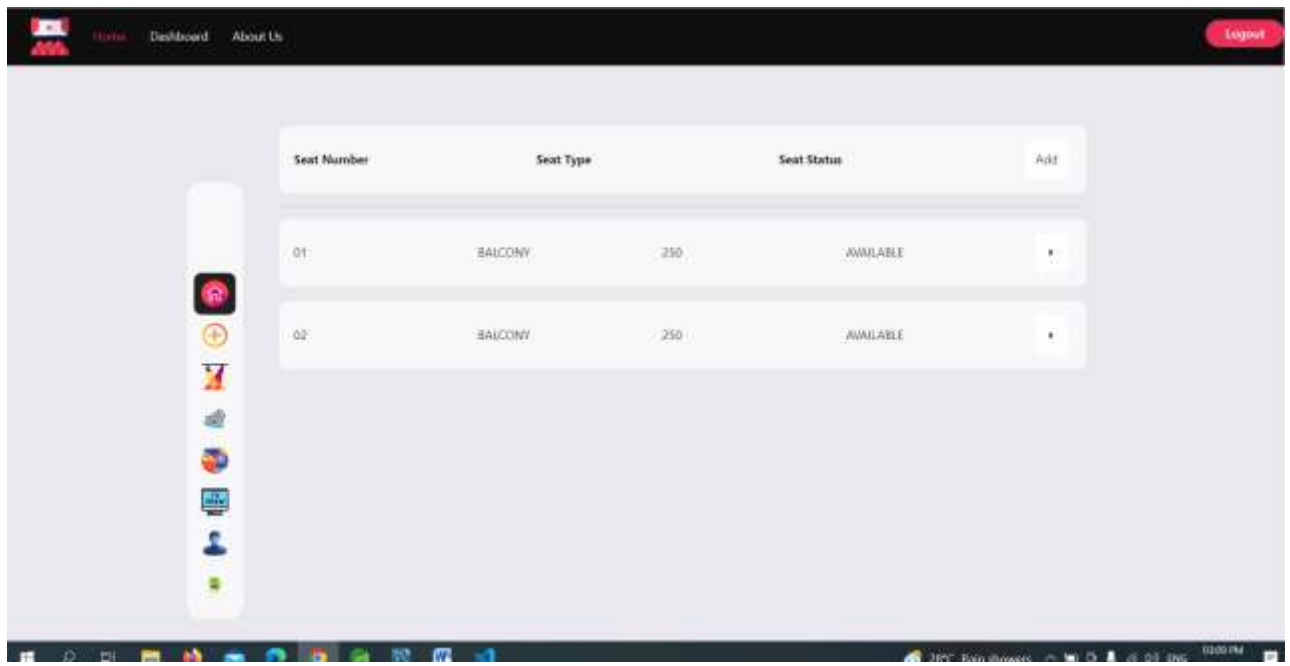


01-07-2022 12:10 PM

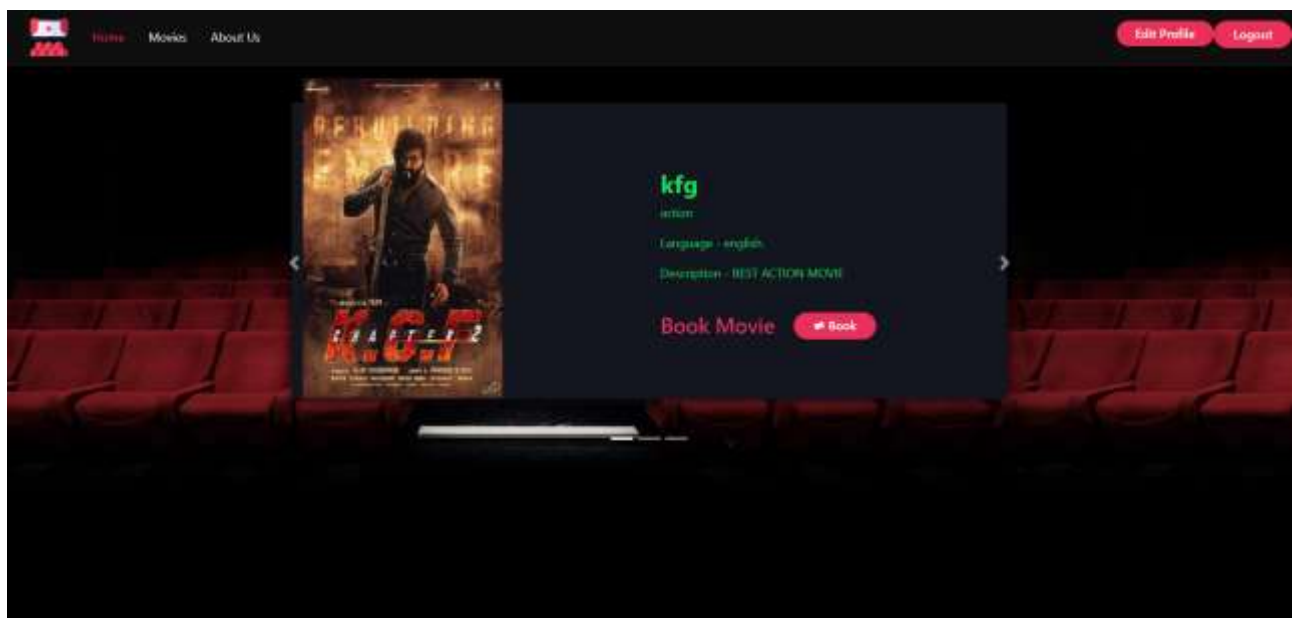


Add Show

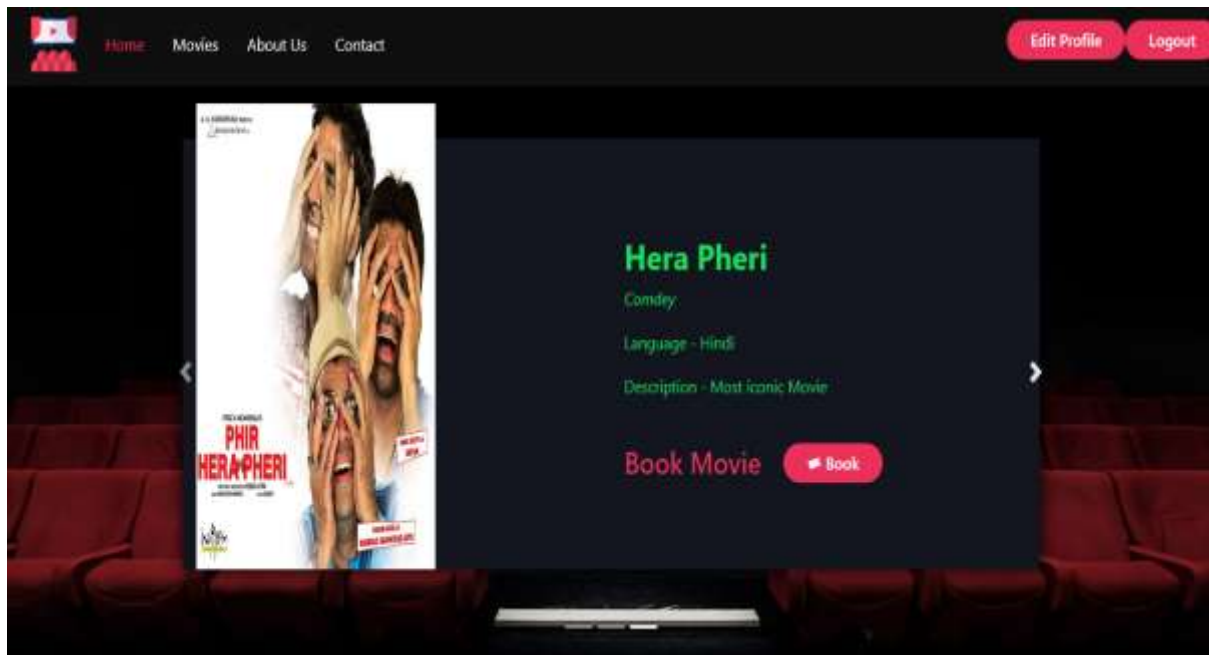
Adding Show Details



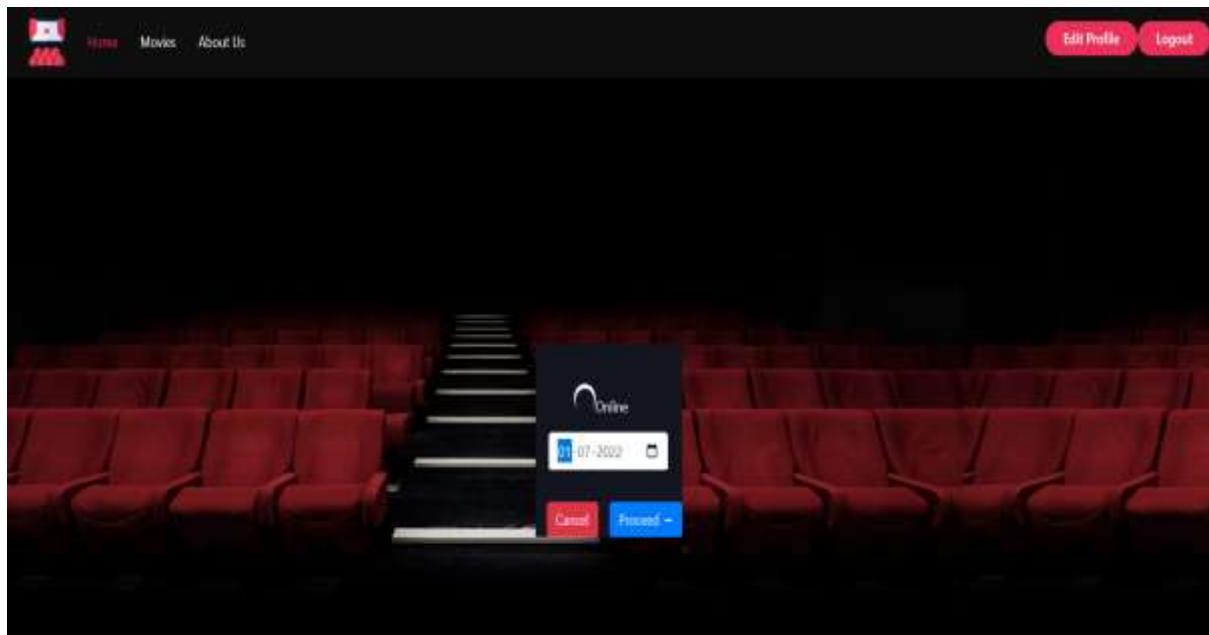
Adding Seat Page



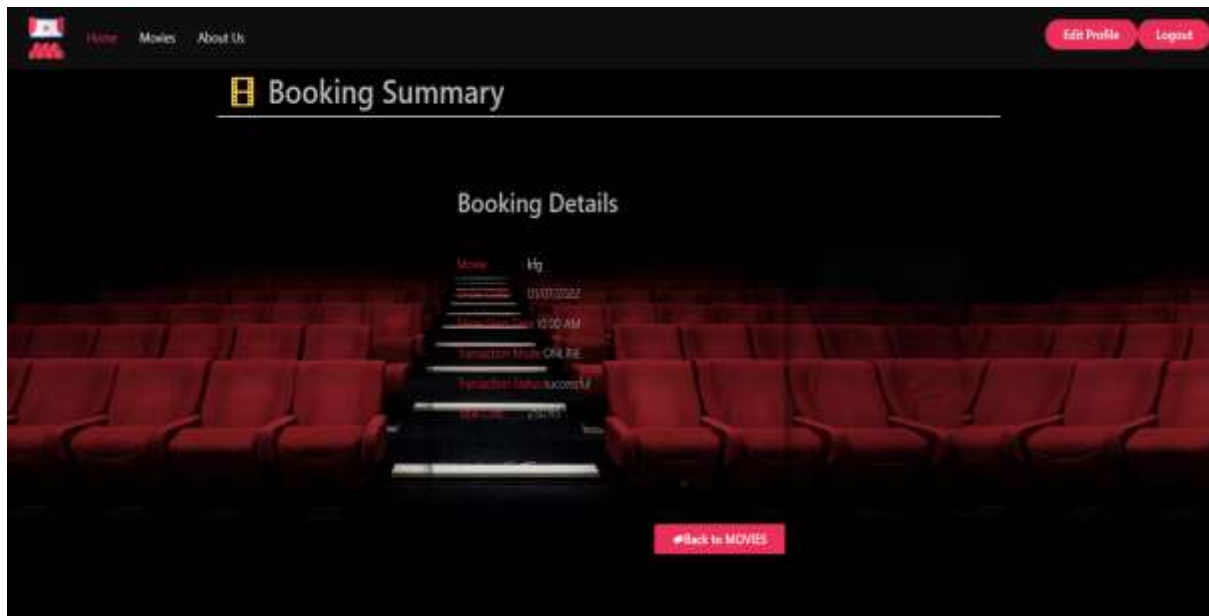
User Page



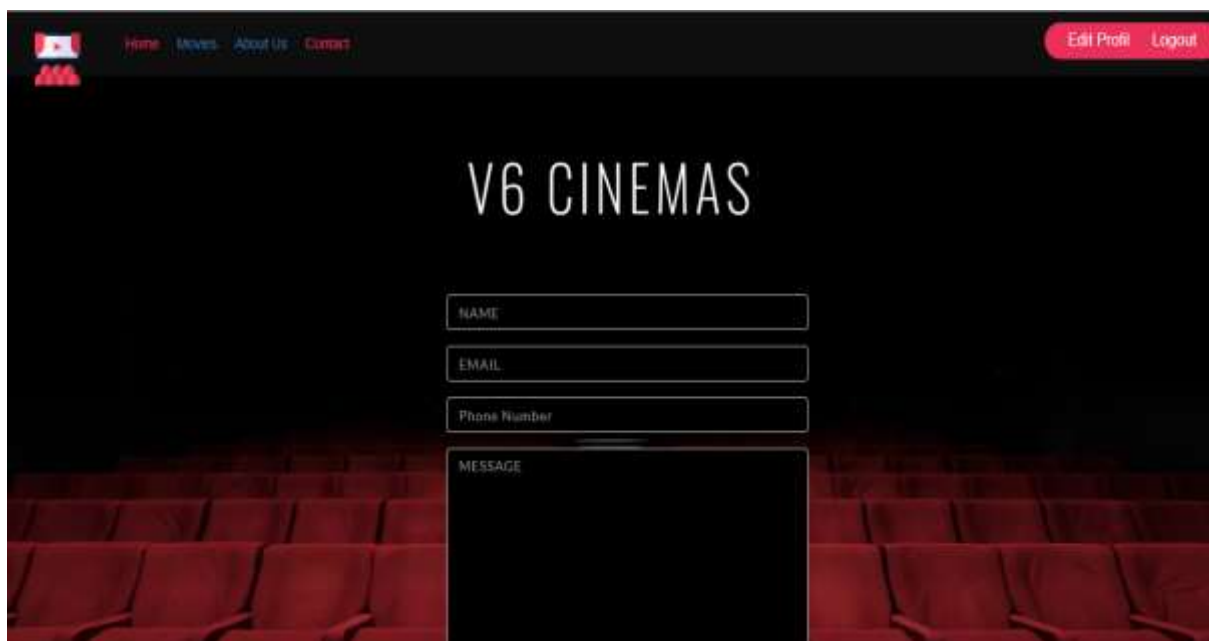
User Home Page



User Online Booking Page



User Booking Detail Page



Contact Us

CHAPTER-7

System Implementation

Frontend

INDEX.HTML

```

<section class="home">
  <section class="latest-movies" id="home">
    <div class="container">
      <div id="carouselExampleIndicators" class="carousel slide" data-
ride="carousel">
        <ol class="carousel-indicators">
          <li data-target="#carouselExampleIndicators" data-slide-to="0"
class="active"></li>
          <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
          <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
        </ol>
        <div class="carousel-inner">
          <div class="carousel-item" *ngFor="let movie of latestMovies; let i
= index" [ngClass]="{ active: i == 0 }">
            <div class="latest-movies-slider">
              <div class="row latest-movies-slide">
                <div class="col-lg-6 col-md-5">
                  <div class="latest-movies-content">
                    
                  </div>
                </div>
              </div>
              <div class="details col-lg-6 col-md-7">
                <div class="latest-movies-content">
                  <h2>
                    <strong>{{ movie.movieName }} </strong>
                  </h2>
                  <p>
                    <i>{{ movie.movieGenre }}</i>
                  </p>
                  <p>Language - {{ movie.movieLanguage }}</p>
                  <p>
                    Description -
                    {{ movie.movieDescription }}
                  </p>
                  <div class="book-movie">
                    <h3>Book Movie</h3>
                    <a class="theme-btn" [routerLink]="['/movie/view',
movie.movieId]"><i
                      class="icofont icofont-ticket"></i> Book
                    </a>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>
</section>

```



```
        </div>
      </div>
    </div>
  </div>
</div>
<div>
  <a class="carousel-control-prev" href="#carouselExampleIndicators"
role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleIndicators"
role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
</div>
</section>

</section>
<router-outlet></router-outlet>
```

REGISTRATION.HTML

```
<section id="login">
  <div class="container">
    <h3 style="color: red">{{ errorMessage }}</h3>
    <form [formGroup]="loginForm" (ngSubmit)="doLogin()">
      <div class="d-flex justify-content-center h-100">
        <div class="card">
          <div class="card-header">
            <h3>Sign In</h3>
            <div class="d-flex justify-content-end social_icon"></div>
          </div>
          <div class="card-body">
            <div class="input-group form-group">
              <div class="input-group-prepend">
                <span class="input-group-text"><i class="fas fa-
user"></i></span>
              </div>
              <input type="text" class="form-control"
formControlName="username" placeholder="Username" required />
            </div>
```

```

        <div class="input-group form-group">
            <div class="input-group-prepend">
                <span class="input-group-text"><i class="fas fa-
key"></i></span>
            </div>
            <input type="password" class="form-control"
formControlName="password" placeholder="Password" required />
        </div>
        <div class="row align-items-center remember">
            <input type="checkbox" />Remember Me
        </div>
        <div class="form-group">
            <button type="submit" value="Login" class="btn float-right
login_btn">
                Login
            </button>
        </div>
    </div>
    <hr />
    <div class="card-footer">
        <div class="d-flex justify-content-center links">
            Don't have an account?<a [routerLink]="['/signUp']">Sign Up</a>
        </div>
    </div>
</div>
</form>
</div>
</section>
```

CONTACT US.HTML

```
<!DOCTYPE html>
<html>

<head>
    <title>Contact Us</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="contact.component.css">
    <link
href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"
rel="stylesheet">
    <link href='https://fonts.googleapis.com/css?family=Lato:400,100,300'
rel='stylesheet' type='text/css'>
```

```
<link href="https://fonts.googleapis.com/css?family=Raleway:100,400,700"
rel="stylesheet">
```

```
<link href="https://fonts.googleapis.com/css?family=Oswald:200,400,700"
rel="stylesheet">
```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.5.0/css/font-awesome.min.css">
```

```
<script src="https://use.fontawesome.com/20ab91acc4.js"></script>
<link rel='stylesheet'
href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css'>
</head>
```

```
<body>
```

```
<section id="contact">
```

```
<form #form="ngForm">
```

```
<h1 class="section-header">V6 Cinemas</h1>
```

```
<div class="contact-wrapper">
```

```
<form id="contact-form" class="form-horizontal" role="form">
```

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

```
<div class="col-sm-12">
```

```
<input type="text" class="form-control" id="name"
```

```
[(ngModel)]="employee.name"
```

```
placeholder="NAME" name="name" value=""
```

```
required>
```

```
</div>
```

```
</div>
```

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

```
<div class="col-sm-12">
```

```
<input type="email" class="form-control"
```

```
id="email" email #prmEmail="ngModel"
```

```
[(ngModel)]="employee.email"
```

```
placeholder="EMAIL" name="email" value="" required
```

```
#idfield="ngModel">
```

```
</div>
```

```
</div>
```

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

```
<div class="col-sm-12">
```

```
<input type="number" class="form-control"
```

```
name="number" [(ngModel)]="employee.phone"
```

```
placeholder="Phone Number" required=""
```

```
id="phone">
```

```
</div>
```

```
</div>
```

```
        <textarea class="form-control" rows="10"
[(ngModel)]="employee.message" placeholder="MESSAGE"
        name="message" required></textarea>

        <button class="btn btn-primary send-button" id="submit"
type="submit" value="SEND">
            <div class="send-text">
                <span class="send-text"
(click)="createEmployee()">SEND</span>
            </div>
        </button>

    </form>
</div>
</form>
</section>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></scri
pt>
    <script
src='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js'></sc
ript>
    <script src="./script.js"></script>
</body>

</html>
```

Middle ware (Spring)

home.component.TS

```
import { Component, OnInit } from '@angular/core';
import { Movie } from '../movie/movie';
import { MveeService } from '../movie/mvee.service';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css'],
})
export class HomeComponent implements OnInit {
  topRatedMovies: Movie[];
  latestMovies: Movie[];
  constructor(private mService: MveeService) {}

  ngOnInit(): void {
    this.mService.getAllMovies().subscribe((data: Movie[]) => {
      this.topRatedMovies = data
        .sort((a, b) => (a.movieRating > b.movieRating ? -1 : 1))
        .slice(0, 8);
    });
    this.mService.getAllMovies().subscribe((data: Movie[]) => {
      this.latestMovies = data
        .sort((a, b) => (a.movieDate > b.movieDate ? -1 : 1))
        .slice(0, 3);
    });
  }
}
```

login.component.TS

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { LogginService } from '../loggin.service';
@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css'],
})

export class LoginComponent implements OnInit {
  loginForm: FormGroup;
  errorMessage: string;
  constructor(
    private fb: FormBuilder,
    private lService: LogginService,
    private router: Router
  ) {}

  ngOnInit(): void {
    this.loginForm = this.fb.group({
      username: ['', Validators.required],
      password: ['', Validators.required],
    });
  }

  doLogin() {
    this.lService
      .login(this.loginForm.value.username, this.loginForm.value.password)
      .subscribe(
        (res: any) => {
          if (res.user.role === 'admin') {
            this.lService.setUserRole(true);
            console.log(this.lService.getUserRole());
          } else {
            this.lService.setUserRole(false);
          }

          if (res.user.customer != null) {
            this.lService.setCustomerId(res.user.customer.customerId);
          }
          this.lService.setLoginStatus(res.loginStatus);

          this.router.navigate(['']);
        },
      ),
  }
```

```
        (error: any) => {
            this.errorMessage = error;
        }
    );
}
}
```

MovieFinalApplication.java

```
package com.cg.mts;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;
import springfox.documentation.swagger2.annotations.EnableSwagger2;

@EnableSwagger2
@SpringBootApplication
@ComponentScan("com.cg")
public class MovieFinalApplication
{
    public static void main(String[] args)
    {
        SpringApplication.run(MovieFinalApplication.class, args);
    }
}
```

CHAPTER-8

Future Enhancement

- This System being web-based and an undertaking of Cyber Security Division, needs to be thoroughly tested to find out any security gaps.
- A console for the data centre may be made available to allow the personnel to monitor on the sites which were cleared for hosting during a particular period.
- We adding Online payment like methods like Paypal, UPI and card transactions.

CHAPTER-9

Conclusion

9.1 Conclusion

It has been a great pleasure for us to work on this exciting and challenging project. This project proved good for us as it provided practical knowledge of not only programming in MySQL & Angular web based application and some extent Web Application & MySQL Database, but also about all handling procedure related with “ONLINE MOVIE TICKET BOOKING”. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

BENEFITS:

The project is identified by the merits of the system offered to the user. The merits of this project are as follows:

- It's a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover, there is restriction for his that he cannot change the

primary data field. This keeps the validity of the data to longer extent.

- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features it will increase the efficiency, accuracy and transparency.

LIMITATIONS:

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.

Chapter-10

References

Web Sites:

<http://www.bookmyshow.com>

<http://www.google.com>

<https://w>We adding Online payment like methods like Paypal, UPI and card transactions.

<ww.wikipedia.org>