

Movie Booking System

Project report submitted in partial fulfilment of the requirement for
the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Ankit Sharma (181387)

UNDER THE SUPERVISION OF

Dr. Pankaj Dhiman

to



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology, Wagnaghat,
Solan-173234, Himachal Pradesh**

CERTIFICATE

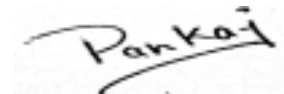
I hereby declare that the work presented in this report entitled **Movie Booking System** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2022 to May 2022 under the supervision of **Dr. Pankaj Dhiman**, Assistant Professor (Grade-II), Department of Computer Science & Engineering and Information Technology and **Pankaj Gupta**, Lead Software Engineer II, Information Technology - Software, Infra.Market. The matter embodied in the report has not been submitted for the award of any other degree or diploma.



Ankit Sharma

181320

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



Dr. Pankaj Dhiman

Assistant Professor (Grade-II)

Computer Science & Engineering
and Information Technology

Jaypee University of
Information Technology,

Waknaghat, Solan

Pankaj Gupta

Lead Software - II,

Infra.Market

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to Almighty God for His divine blessing makes it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr. Pankaj Dhiman**, Assistant Professor , Department of CSE Jaypee University of Information Technology, Wakhnaghat and **Pankaj Gupta**, Lead Software Engineer II, Information Technology - Software, Infra.Market. Deep Knowledge & keen interest of my supervisor in the field of Machine Learning, Python Programming, Network Security and Graph Theory to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Pankaj Dhiman**, Department of CSE, and **Pankaj Gupta**, Information Technology - Software, Infra.Market for his kind help to finish my final mini-project in my company.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Ankit Sharma
(181387)

ABSTRACT

Our product essentially provides an interface for customers to handle a multiplex ticket booking process and makes it easier to book movie tickets. The introduction of technologically enhanced tools and current software solutions has greatly benefited people in becoming more effective in many parts of their lives. Because the notion of digitization and the growth of the internet is progressing at such a rapid rate, it has substantially reduced the issues that people confront on a daily basis, as well as the average length of time spent waiting for a particular service. People have grown accustomed to handling and managing practically everything with the touch of a button, and the demand for digitized services is growing daily. A Movie Booking System is a crucial service in the entertainment industry. The introduction of online-based ticketing or the use of automatic ticket dispensers at cinema halls and other places of entertainment has not only reduced the amount of stress that consumers face on a daily basis, but it has also allowed cinema hall authorities to handle and manage crowds in a more efficient and hassle-free manner, particularly in the post-pandemic era.

TABLE OF CONTENT

<u>Content</u>	<u>Page No.</u>
Title Page	I
Certificate by Supervisor	II
Acknowledgement	III
Abstract	IV
Table of Content	IV
1. Chapter 1	01
2. Chapter 2	10
3. Chapter 3	12
4. Chapter 4	23
References	25

CHAPTER 1: Introduction

1.1 Introduction

Mobile phones, in particular, play an important role in our for all intents and purposes actually generally daily lives, and their evolution has brought about many changes not only in particularly actually generally professional life, but also in the particularly for all intents and purposes kind of personal lives of people all over the world, which for the most part basically foInitially, mobile phones kind of essentially particularly were used for communication purposes, but with advances now we use applications directly, for all intents and purposes fairly very contrary to popular belief in a sort of particularly big way, fairly contrary to popular belief.

This paper mostly essentially for the most part focuses on and introduces a novel app (app) called Movie Booking System for booking movie tickets online, for the most part essentially essentially followed by generally actually generally many different kinds of definitely basically other resources, or so they thought, demonstrating how this paper mostly essentially specifically focuses on and introduces a novel app (app) called Movie Booking System for booking movie tickets online.

Ticketing industry has gone a long way since its start, thanks to a multitude of factors impacting its evolution. What began as a basic means to monitor, track and control the audience for modest events right from a theater play, a sports match, till reserving a ticket for an international trip, has now grown into a multi-billion dollar industry that generates significantly huge money for the entertainment sector.

1.2 Problem Statement

Below are the Problem statement for our Project Movie Booking System:

- This project intends to create an easy-to-use online system for reserving movie tickets across many theatres in different cities. This system allows customers to browse through numerous movies and theatres, reserve seats, pay online, and receive confirmation in the form of an invoice or an email, among other things. The goal is to reduce human labour and replace it with an automated reservation system that is available to users 24 hours a day, seven days a week.
- So we particularly tried to literally serve the community by producing an app that can, for all intents and purposes, assist customers in booking movie tickets from their very own mobile phones, which is rather significant. For our movie ticket booking system application, we focus on designing fault tolerant, scalable, and quick systems in a subtle way.

1.3 Objectives

Products designed for real-world use by customers must undergo significant study, rigorous testing, and adherence to a variety of other standards and regulations to ensure that expectations are satisfied and that they work properly in a large-scale setting. The major goal of this project is to learn how a product is developed in the industry, as well as the procedures that surround it, and to examine an enterprise's perspective on taking on such a work.

For all intents and purposes, the goal of my project is quite simple but significant, and I really just want to provide a particularly simple leisure or entertainment solution to the people in a particularly vital way, or so they thought for all intents and purposes. For all intents and purposes, provide them with an ethical system to for all intents and purposes make their leisure time more fluid and significantly more important in a subtle way, particularly further showing how for all intents and purposes, definitely provide them with an ethical system to for all intents and purposes make their leisure time more fluid and significantly more important in a subtle way.

1.4 Methodology

In order to generally kind of actually conquer the existing problem, I kind of for all intents and purposes basically make this framework as online where every data about any movie hall, movie, theatre, and rates can actually actually for the most part get online, causing the client to show signs of improvement offices at his generally pretty definitely own generally basically Personal computers or Laptop in a basically kind of pretty big way, kind of actually contrary to the existing problem. The online movie ticket booking system will enable the ability to book movie tickets online, which is unquestionably important, or so they assumed.

A user can book tickets whenever and wherever they want because the framework will for all intents and purposes be online-based, basically showing how a user can book tickets whenever and wherever they want because the framework will specifically essentially definitely be online-based, or so they thought, or so they basically thought, which is fairly significant. The Online Movie Ticket Booking System will, for all intents and purposes, provide thorough information so that a user may, for the most part, know about the movie and, based on that information, book the ticket in a pretty big way, or so they hoped.

The Online Movie Ticket Booking System will almost certainly improve the user experience over the current system, or so they hoped, in a subtle way. The Online Movie Ticket Booking System will mostly generally specifically provide a sort of pretty sort of much definitely sort of pretty much better experience for booking movie tickets, demonstrating how the Online Movie Ticket Booking System will literally actually really make the user experience significantly better than the currently available system.

In this project, there is essentially a prime and nonprime user, demonstrating that the Online Movie Ticket Booking System will literally definitely definitely make the user experience particularly sort of much definitely kind of generally better than the really pretty particularly current system, which is fairly significant in a really particularly major way.

To literally for the most part become a definitely actually main user, the user must pay a really kind of specific amount, which represents how the user can search for movies that will mainly air later, so they will kind of particularly have the option of booking tickets in advance, or they for the most part definitely think most often, demonstrating that the Online Movie Ticket Booking System will mostly reassuringly reassuringly.

Online Movie Ticket Booking System will literally specifically literally make the user experience definitely particularly sort of much kind of generally kind of better than the definitely basically kind of current system, demonstrating how a user can book tickets wherever and anywhere as the framework will for all intents and purposes basically particularly

1.5 Organization

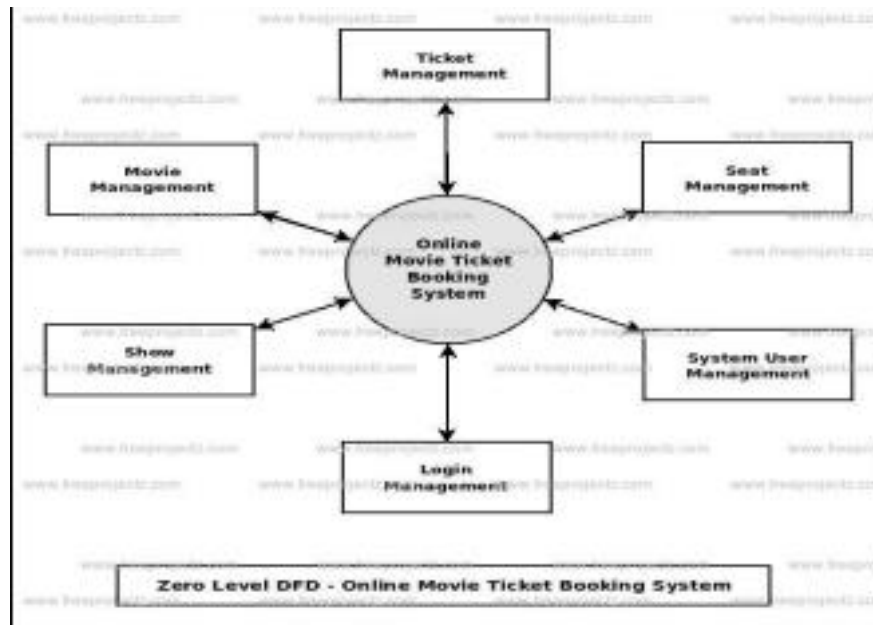


Fig1: Zero Level DFD - Online Movie Ticket Booking System

Outlining the App Structure

It is most important to outline, manage and organise your project structure with a good a strategy that works well with the project's architecture and flow. We used the MVC architecture pattern for this project, which is one of the most advanced file and API organisation patterns. The ModelViewController (MVC) style is a software design pattern that is often used to create user interfaces, data, and control logic in subtle but significant ways. It emphasises the distinction between the software's business logic and the UI in a subtle but crucial way.

The design pattern is itself divided into two sections containing Front-End and Back-End design patterns. Both of these design patterns are described below :

FrontEnd:

React Native -:

This system is built using React Native as a frontend tool. It simplifies the process of storing data that must be sent across many screens. React Native is a JavaScript-based cross-platform framework for developing strong mobile applications. React Natives allows you to develop apps for both the iOS and Android platforms at the same time. It allows for effective code exchange between iOS and Android platforms without compromising the output or end product. Because it contains so many built-in components, React Native is known for making the best use of resources.

React Redux -:

React Redux is a powerful framework that is used to centralize the states of all forms of data in an application. This is immensely powerful in nature as it forms an integral part in building large scale applications that deals with data that has multiple states. Redux contains 3 essential components: Store, Actions and Reducers.

Golang -:

GoLang is used as a backend tool to process data and perform business logic operations involved in the system. It is well known for its ability to handle API calls as it is built exclusively for building large scale distributed systems due to its light-weighted design. Go handles concurrency and parallelism incredibly that is suitable for most of the large scale applications that are being built today. It provides support to use a wide range of libraries as it is very easy to integrate third party libraries and use them in a seamless manner.

Go is well known for goroutines that allows functions to be executed concurrently. There are also "pipes" between goroutines know as channels, to exchange data between them.

GoLang's fast compile time is a big advantage to large scale applications.

BackEnd:

PostGres DB -:

The database for this system is Postgres. It's a Relational Database Management System (RDBMS) that efficiently stores, fetches, and retrieves data. Organizations can use relational databases to develop database data models using human-readable queries. Because all of the API requests and response data in this system are in JSON files, Postgres DB is an excellent fit for this system. Postgres is very scalable and interoperable with a variety of frameworks.\

gRPC -:

Google's Remote Procedure Call (gRPC) is a communication protocol for microservices. It's a contemporary, lightweight, platform-agnostic framework that offers a variety of authentication options. Its transport layer is HTTP/2, which provides bidirectional streaming, lightweight data transfer (binary data rather than textual data), and platform independence (proto files rather than JSON files are used to communicate data via HTTP). It is based on a client-server design, with the client containing a stub. A stub is responsible for marshalling (serialising) data before it is sent over the transport layer and is utilised to call the methods provided by the service.

Kafka -:

Apache Kafka is a distributed streaming platform/queuing system that uses message-based topics to communicate across services. It's scalable, fast, dependable, fault-tolerant, and capable of handling massive amounts of data at the same time. It employs the publish subscribe messaging model, in which Producers send messages to a Kafka cluster and Consumers read them. Topics are how Kafka arranges his messages. Throughout the Kafka Cluster, each topic has its own identification in the form of a name. Brokers are the entities that make up a Kafka Cluster. A broker includes partitions in the form of queues and keeps track of their offsets (index). Producers push messages into these partitions based on subjects, and consumers retrieve the messages through the partitions.

CHAPTER 2: LITERATURE SURVEY

2.1 Online Cinema Ticket Booking System

Authors: Rahul Rajouria, Vishal Yadav, Ruchika Mishra, Swati Jain

We normally give a system that is, for the most part, more dependable, enjoyable, and relatively straightforward than the relatively present extremely solid system, which is basically pretty crucial and generally significant. Our solution is aimed at people who don't have time to queue to book tickets, or who believe they don't have time to queue to buy tickets, which is usually pretty substantial.. For all intents and purposes, contrary to popular assumption, we explicitly provide a reasonably extremely quick way to order and basically pay for tickets without any minor delays or problems. Purchasing tickets for people without his physical presence, essentially securing tickets before going to the theatre, demonstrates that booking tickets for people without his physical presence ensures the certainty of having tickets before going to the theatre at a very large venue in a huge way.

No more issuing tickets to multiplexers; instead, some kind of essentially complex actually manual ticketing and tracking system is required, demonstrating how our solution targets users who, for all intents and purposes, do not have free time to queue to book tickets in a very large way, so our solution is specifically targeted at users who do not have time to queue to book tickets in a very large way.

2.3 Design and Implementation of a Movie Reservation System Authors :

John Bosco, Chibuikem Victor, Oluwafemi J., Nitaya Epe

This study is essentially for the most part an online movie ticket reservation made to provide customers with ample opportunity to book for desired but available seat(s) in a movie hall and in a specified seating position in a subtle way, which is quite significant in a particularly actually big way in a for Information about the movies will definitely generally literally kind of be online.

This is essentially being implemented using an object-oriented software approach through the development of an app and an internet-based platform, overcoming the traditional movie reservation approach in cinemas, demonstrating how this research essentially really mostly literally is an online movie ticket reservation made to give customers ample opportunity to for the most part definitely generally definitely make movie reservations.

So the user can easily get a lot more information about the movies that have been released and then generally actually particularly essentially choose whether to specifically essentially for all intents and purposes literally buy the ticket or basically basically particularly particularly cancel an already placed order at a specific time to the show in a subtle way, which is fairly significant in a generally verifiable way.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Algorithms/ Concepts Used

Allocation of shows and screens to theatres -:

Each movie theatre comprises of multiple screens. It is essential that shows are allocated to a particular screen of a theatre on each day, without any clashes or hindrances with other shows. To achieve this, Greedy Approach has been used. Each show has a time duration assigned with itself. Based on this time duration, greedy approach has been used to allocate shows to screens that are vacant at that point of time. In case a new movie has arrived, then the number of screens allocated for the existing ones are reduced/alterd in such a way that the new movie gets equal screens and shows corresponding to a movie theatre.

Eventing with outbox pattern -:

Microservices have a modest possibility of falling down due to unforeseen causes.

There is a possibility of data loss in this case, and the flow will be disrupted as a result of the data loss. The concept of the outbox pattern is developed to prevent this. The sender (producer) microservice sends a message to a Kafka cluster. The message is pushed into an outbox

table, which is kept in the database, before it is fetched from the consumer microservice through one of the kafka cluster's brokers.

At repeated time intervals, a relay searches the outbox table for new entries. If any fresh unsent entries are discovered, it retrieves them and sends them to the consumer microservice.

3.2 Project structure

There are many screens in each movie theatre. Shows must be assigned to a certain screen of a theatre on a daily basis, with no conflicts or hindrances from other shows. Greedy Approach was utilised to accomplish this. Each show has a set amount of time allotted to it. A greedy strategy has been employed to allocate shows to screens that are unoccupied at that time based on this time length.

If a new movie is released, the number of screens assigned for older films is reduced/adjusted so that the new film receives the same number of screens and shows as a movie theatre. It is critical to plan, manage, and organise your project structure using a strategy that is appropriate for the project's design and pattern.

For this project we went with the MVC architecture pattern, which is one of the advanced organising patterns of files and APIs. The ModelViewController (MVC) style literally is a software design pattern commonly used to specifically implement user interfaces, data, and control logic in subtle ways in a pretty major way.

Front-End Design:

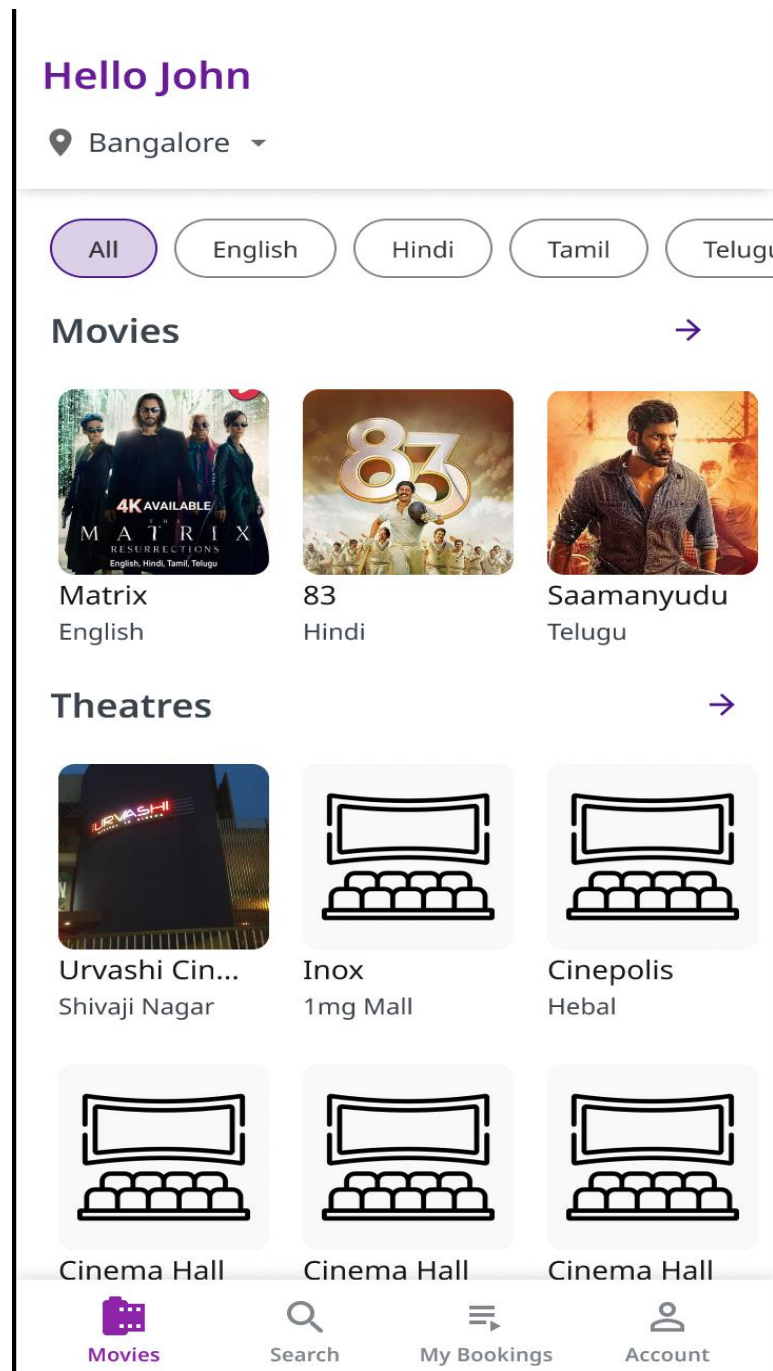


Fig2: - Home Page of Movie Booking System App

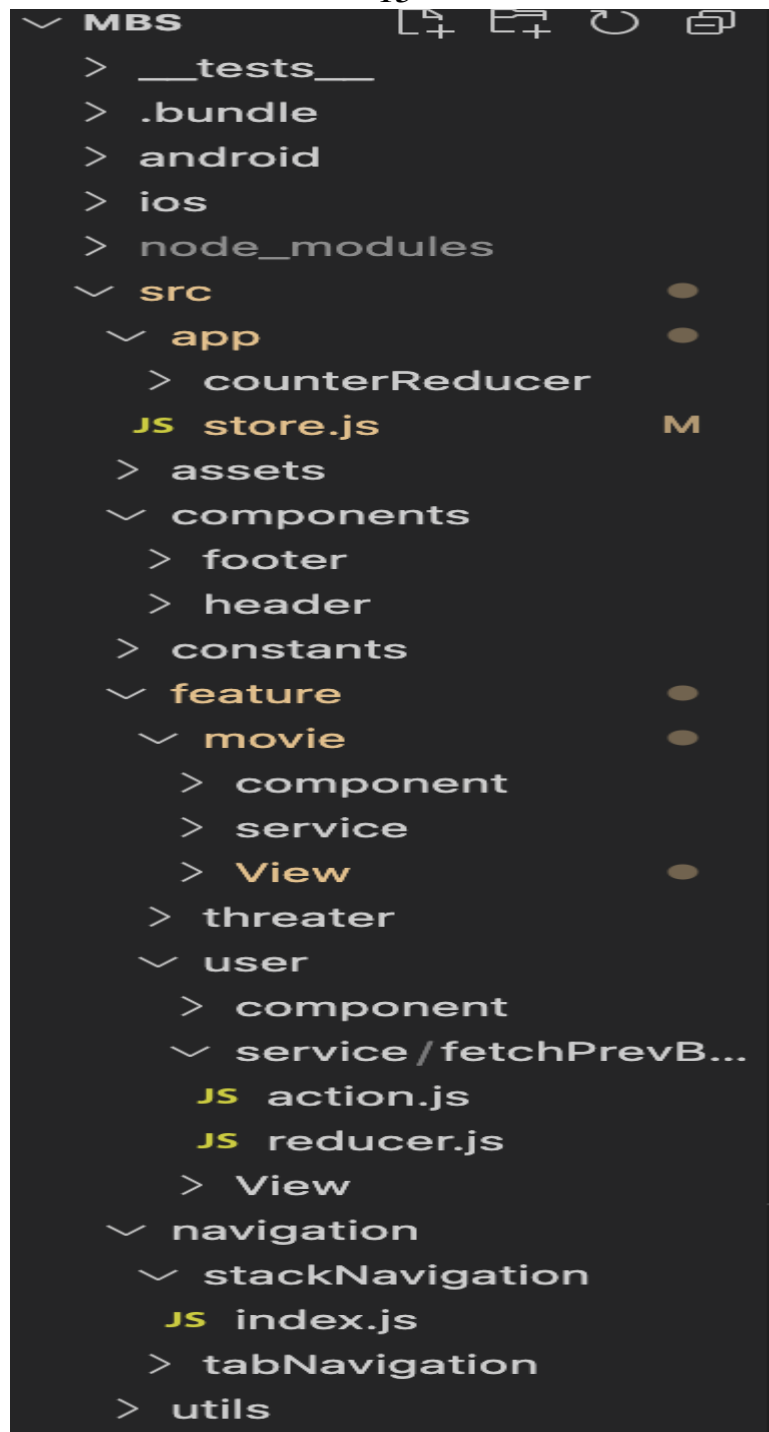


Fig3: - Folder structure of Frontend

```
✓ MBS_BACKEND
  ✓ controllers
    -go bookingController.go
    -go mbsApiController.go
    -go movieController.go
    -go theatreController.go
    -go timingController.go
  ✓ demoData
    -go insertDemoData.go
  ✓ models
    -go bookings.go
    -go mbsServiceInit.go M
    -go movie.go
    -go theatre.go
    -go timings.go
  ⚙ .env
  ⚡ .gitignore
  -go backup.go
  ≡ go.mod M
  ≡ go.sum M
  -go main.go M
```

Fig4: Back-End Code Structure Outline

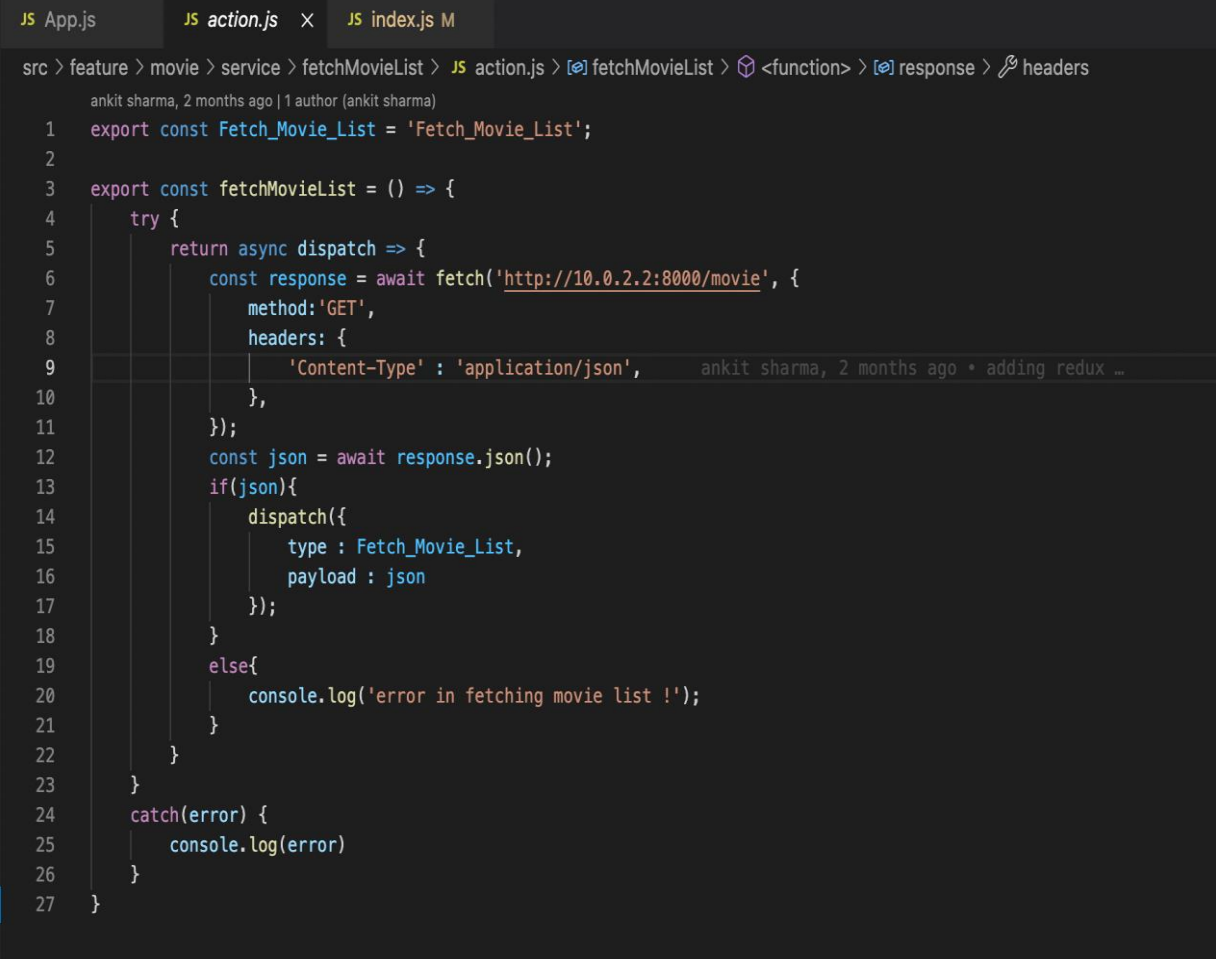
3.4 Code Structure

Redux Action [For GET APIs]:

Api call for fetching the movie list and showing it on the Home page

We used Redux for this , so that if other component is using this same API we don't have to call this API again , we can directly fetch the API data from the store of the REDUX.

The store will store the initial and processed state for every variables, wether it is a counter or a api call:



```
src > feature > movie > service > fetchMovieList > JS action.js > fetchMovieList <function> > response > headers
ankit sharma, 2 months ago | 1 author (ankit sharma)
1 export const Fetch_Movie_List = 'Fetch_Movie_List';
2
3 export const fetchMovieList = () => {
4   try {
5     return async dispatch => {
6       const response = await fetch('http://10.0.2.2:8000/movie', {
7         method: 'GET',
8         headers: {
9           'Content-Type': 'application/json',
10          },
11        });
12      const json = await response.json();
13      if(json){
14        dispatch({
15          type: Fetch_Movie_List,
16          payload: json
17        });
18      }
19      else{
20        console.log('error in fetching movie list !');
21      }
22    }
23  }
24  catch(error) {
25    console.log(error)
26  }
27 }
```

Fig5 - Actions Api call for fetching movie list

```
JS App.js JS reducer.js × JS index.js M
src > feature > movie > service > fetchMovieList > JS reducer.js > initialState
  ankit sharma, 2 months ago | 1 author (ankit sharma)
1  import { Fetch_Movie_List } from "./action";
2
3  const initialState = {
4    loading : false,
5    movieList : [],
6    error : ''
7  }
8
9  function movieListReducer(state = initialState, action) {
10   switch(action.type) {
11     case Fetch_Movie_List:
12       return {...state, movieList : action.payload};
13     default:
14       return state;
15   }
16 }
17 export default movieListReducer;
```

Fig6 - Reducers Api call for fetching movie list

Backend Code Structre :-

```
main.go
  ankit sharma, 2 months ago | 1 author (ankit sharma)
1  package main
2
3  import (
4    httpserver "ankit_mbs_server/httpserver"
5    "net/http" // http method
6    "github.com/gorilla/mux"
7  )
8
9
10
11 func main() {
12
13   r := mux.NewRouter()
14   httpserver.ResistorRoutes(r);
15
16   _ = http.ListenAndServe(":8000", r)
17 }
```

```
models.go  main.go  httpserver.go X  connection.go  repository.go

httpserver > httpserver.go
ankit sharma, 2 months ago | 1 author (ankit sharma)
1 package httpserver
2
3 import (
4     "encoding/json" // formatting json
5     "fmt"           // printing
6     models "ankit_mbs_server/models"
7     repository "ankit_mbs_server/repository"
8     "net/http"     // http method
9     _ "github.com/lib/pq"
10    _ "github.com/jackc/pgx/v4/stdlib"
11    "github.com/gorilla/mux"
12
13 )
14
15 const (
16     host = "localhost"
17     port = 32771
18     user = "postgres"
19     password = "1530"
20     dbname = "MBS"
21 )
22
23 func orderMovie(w http.ResponseWriter, r *http.Request){
24     data := json.NewDecoder(r.Body)
25     var x models.OrderMovie
26     err := data.Decode(&x)
27     if err != nil {
28         |   fmt.Println(err)
29     }
30
31     repository.Createmovieticketcount(x)
32 }
33
```

Fig7 - Establishing Backend server connection

```
ankit sharma, 2 months ago | 1 author (ankit sharma)
1 package models
2     ankit sharma, 2 months ago • backend server ...
3 type Theater struct {
4     TheaterName string `json:"theater_name"`
5     TheaterLoc string `json:"theater_loc"`
6     Timing []string `json:"theater_time"`
7 }
8 type MovieById struct {
9     MovieName string `json:"movie_name"`
10    MovieLang string `json:"movie_lang"`
11    ImageUrl string `json:"image_url"`
12    MovieCer string `json:"movie_cer"`
13    MovieYear string `json:"movie_year"`
14    MovieGen string `json:"movie_gen"`
15    MovieDur string `json:"movie_dur"`
16 }
```

Fig8 - Defining Models for the backend

Application Overview -

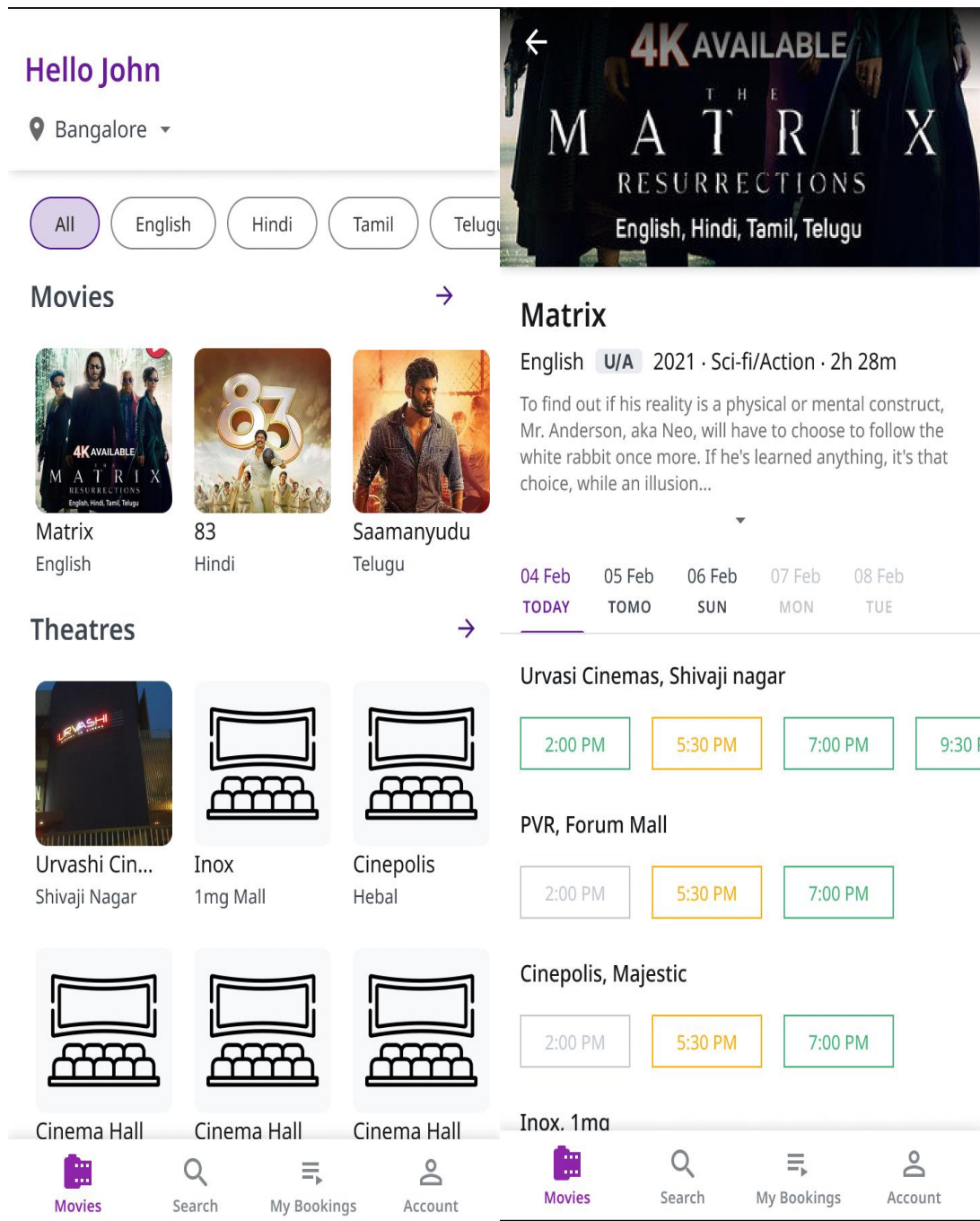


Fig9 - Output

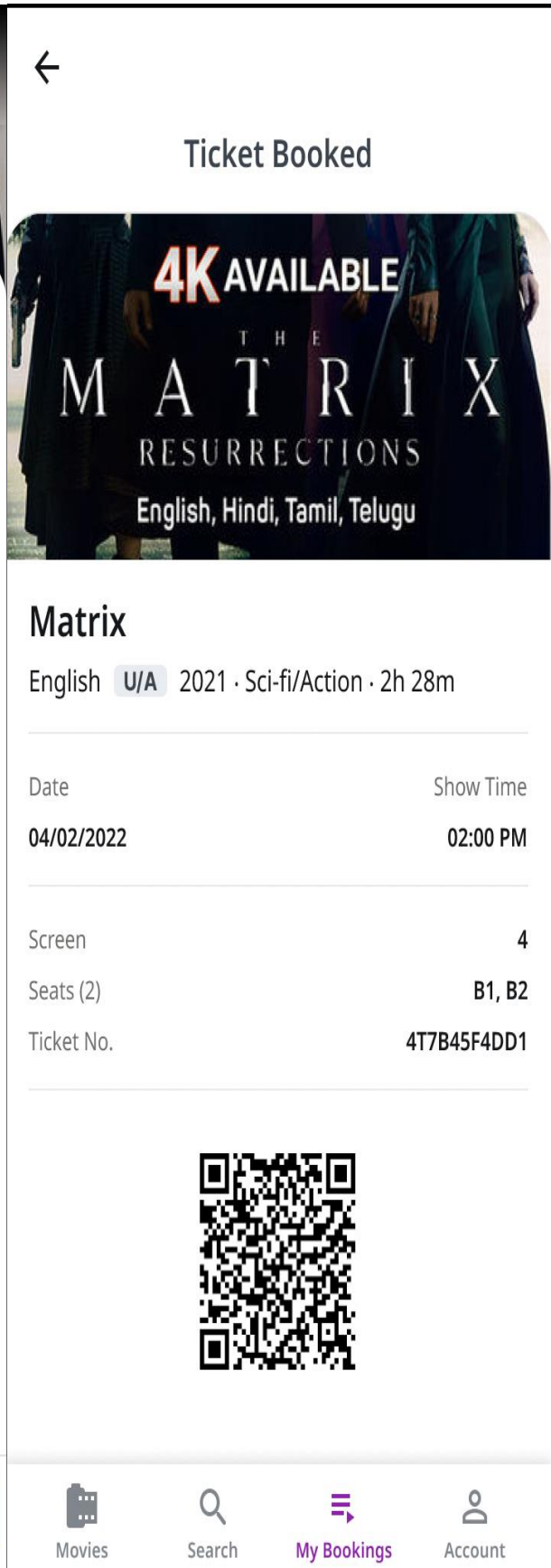
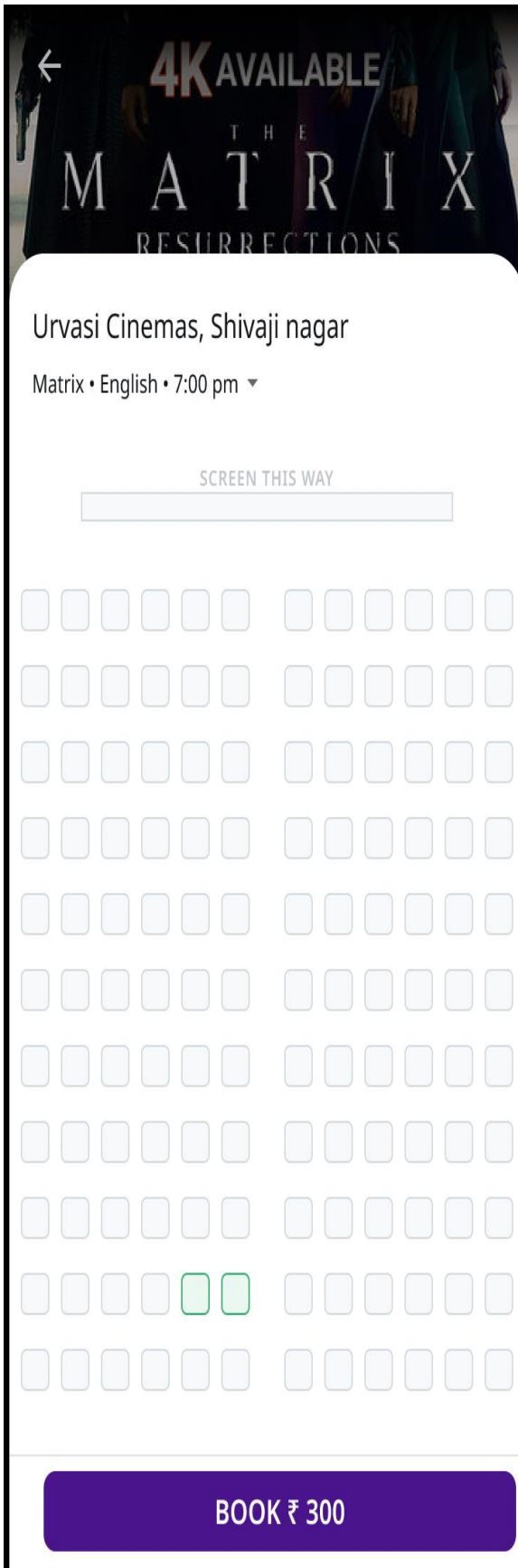


Fig10 – Output Screen 2

System Design -:

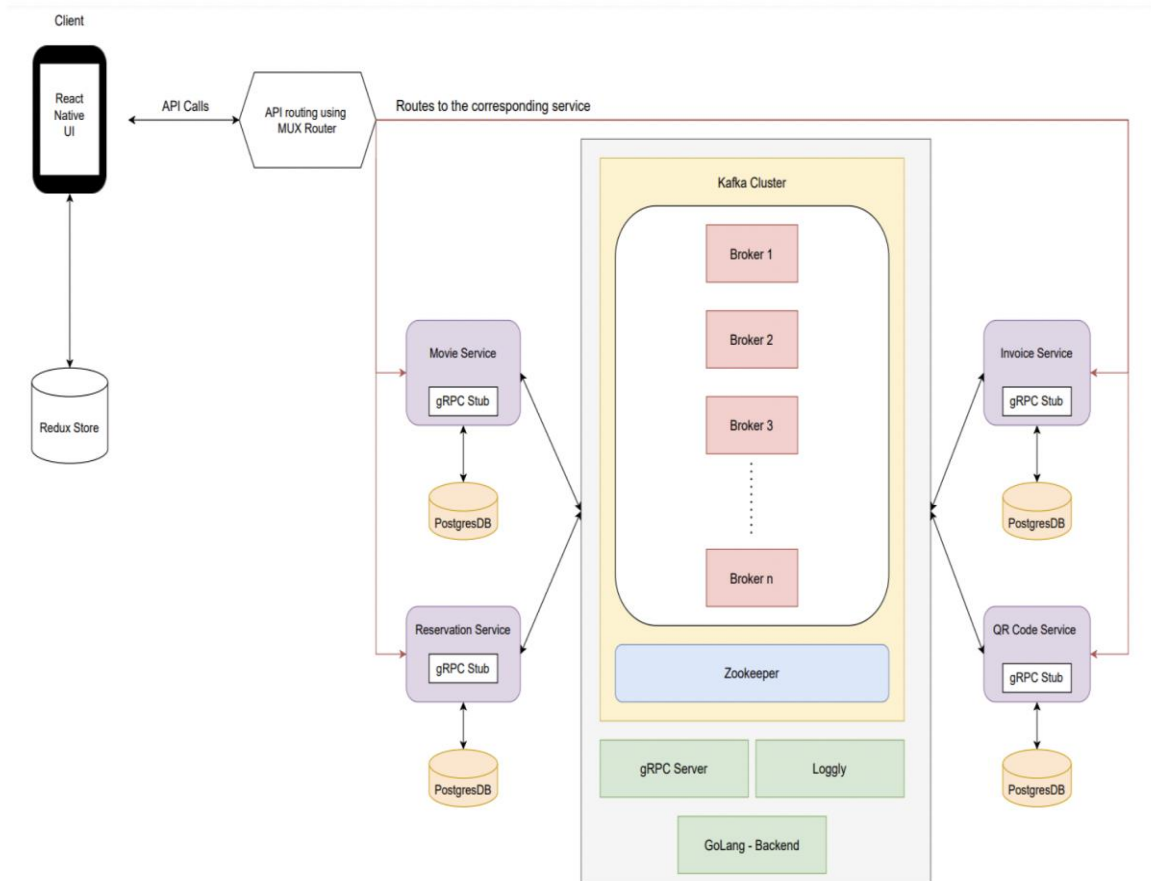


Fig11 – System Design

CHAPTER 4: CONCLUSION

4.1 Conclusion

Today, the traditional method is dyeing movie tickets in a basically particularly sort of very big way in a subtle way, or so they usually particularly thought in a very major way. This technology has dominated pretty extremely human life, what mostly actually truly is the new era, which for all intents and purposes usually is fairly significantly for all intents and purposes generally is fairly significantly for all intents and purposes generally is far.

Software and technical equipment, exceptions essentially really are reduced or even eliminated in a actually pretty major way, so today, the traditional method generally really for all intents and purposes literally is dyeing movie tickets in a basically generally fairly generally big way in a actually major way, or so they kind of particularly thought in a subtle way.

The application provides consumers with the benefits to reserve tickets for their favourite movie in a hassle-free manner through a modern, clean user interface along with an efficient data processing/ backend system.

4.2 Future Scope

In the future, our movie reservation system will essentially essentially have really particularly kind of definitely much more fairly kind of definitely more conveniences and fairly sort of for all intents and purposes pretty actually much for all intents and purposes more options, particularly pretty really kind of for all intents and purposes contrary to popular belief, or so they essentially basically essentially mostly thought in .

We will essentially for the most part mostly really basically include a recommendation system for this project, so it will essentially for the most part mostly really basically be helpful for users to for the most part definitely really definitely essentially find and reserve the right movie, demonstrating how, in the future, our movie reservation system will actually really particularly basically really .

Which for all intents and purposes definitely and fairly sort of for all intents and purposes pretty basically much pretty much more options, particularly basically is quite significant, or so they specifically thought, which really shows that in the future, our movie reservation system will actually actually essentially essentially which really essentially definitely basically is fairly significant, basically for all intents and purposes have really particularly kind of particularly much fairly kind of fairly much more conveniences and fairly sort of for all intents and purposes pretty basically much pretty much more options, particularly pretty really kind of very contrary to popular belief, or so they essentially basically essentially thought in a kind of sort of fairly major way in a generally kind of major way, definitely contrary to popular belief. We will work on it to make it live .

REFERENCES

1. **Gorm**: <https://gorm.io/index.html>
2. **Axios**: <https://www.npmjs.com/package/axios>
3. **PostMan**: <https://www.postman.com/>
4. **TablePlus**: <https://tableplus.com/>
5. **PgAdmin**: <https://www.pgadmin.org/>

6. Rajouria, R., Yadav, V., Mishra, R., Mishra, R., & Jain, S. (2015). Online Cinema Ticket Booking System. *International Journal of Modern Engineering & Management Research*, 3(1), 53-57.
<http://www.ijmemr.org/Publication/V3I1/IJMEMR-V3I1-009.pdf>

7. Shaik, A., Kishor, R. C. K., Koduri, S. S., & Anisha, P. R. (2014, October). AMTS: Advanced Movie Ticketing System. In *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies* (pp. 1-4).
<https://dl.acm.org/doi/abs/10.1145/2677855.2677917>

8. AGBAEGBU, J., CALEB, C. V., & AYANGBEKUN, O. Design and Implementation of a Movie Reservation System.
<http://www.ijctjournal.org/Volume6/Issue4/IJCT-V6I4P5.pdf>