SUMMER TRAINING REPORT

On

Online Movie Ticket Booking Application

Submitted to Guru Gobind Singh Indraprastha University, Delhi (India) in partial fulfillment of the requirement for the award of the degree of

B.TECH

in

INFORMATION TECHNOLOGY

Submitted By

SHUBHAM KUKRETI

ROLL No. 02415003117

DEPTT. OF INFORMATION TECHNOLOGY

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY,
NEW DELHI-110058

SEPTEMBER 2019

ACKNOWLEDGEMENT

A project work owes its success from commencement to completion, to the people in love with developers at various stages. Let me in this page express my gratitude to all those who helped us in various stage of this project. First, I would like to express my sincere gratitude indebtedness to **Dr. Tripti Sharma** (HOD, Department of Information Technology, Maharaja Surajmal Institute of Technology New Delhi) for allowing me to undergo the summer training of 6 weeks at **ISSAC IT LAB SOLUTIONS LLP, ROHINI.**

I am grateful to our guide **Mr. Sandeep Tiwari**, for the help provided in completion of the project, which was assigned to me. Without his friendly help and guidance it was difficult to develop this project.

I am also thankful to **Dr. Prabhjot Kaur** for helping me to preparation of the final report and presentation.

Last but not least, I pay my sincere thanks and gratitude to all the mates for their support and for making our training valuable and fruitful.

CERTIFICATE(FROM THE ORGANIZATION)

ISSAC IT LAB SOLUTIONS LLP

GET SKILLED GET HIRED

An ISO 9001 : 2015 Certified Company

CERTIFICATE

OF COMPLETION

This is to certify that Mr. / Ms. SHUBHAM KUKRETI

has been successfully completed Six WEEKS training in

MERN Full Stack Development on the 14th

day of the month of July in the year 2019

For ISSAC IT LAB SOLUTIONS LLP

15.07.2019

Date Director

For ISSAC IT LAB SOLUTIONS LLP

Auth. Sign Director

CANDIDATE'S DECLARATION

I, **SHUBHAM KUKRETI**, Roll No. **02415003117**, B.Tech (Semester- 5th) of the Maharaja Surajmal Institute of Technology, New Delhi hereby declare that the Training Report entitled '**MovieOn: Online Movie Ticket Booking Application**' is an original work and data provided in the study is authentic to the best of my knowledge. This report has not been submitted to any other Institute for the reward of any other degree.

Shubham Kukreti

(Roll No.: 02415003117)

Place: New Delhi

Date: 18 September'2019

ABOUT ORGANISATION

Issac It Lab Solutions LLP Ltd in Rohini Sector 8 has a wide range of products and services to cater to the varied requirements of their customers.

Established in the year 2015, Issac It Lab Solutions Llp Ltd in Rohini Sector 8, Delhi is a top player in the category Computer Training Institutes in the Delhi. This well-known establishment acts as a one-stop destination servicing customers both local and from other parts of Delhi. Over the course of its journey, this business has established a firm foothold in it's industry.

The belief that customer satisfaction is as important as their products and services, have helped this establishment garner a vast base of customers, which continues to grow by the day. This business employs individuals that are dedicated towards their respective roles and put in a lot of effort to achieve the common vision and larger goals of the company. In the near future, this business aims to expand its line of products and services and cater to a larger client base.

It is known to provide top service in the following categories: Computer Training Institutes, Computer Training Institutes For Java, PHP Training Institutes, Computer Training Institutes For Corporate Java, Computer Training Institutes For Mobile Phone Application, Android Training Institutes, Computer Training Institutes For Advanced Java, Computer Training Institutes For Core Java.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

1.1 INTRODUCTION

My Project is based on Booking Of Movie Tickets Online. It provides user a platform to enquire about the current movies in their city and provides them the facility to book their movie ticket. This application is one of the widely used web portal in present generation for booking tickets through online which had created scope for introducing new features in existing system.

1.1.1 NEED AND OBJECTIVE

Cinema-going is one of the most popular out-of-home cultural activities, affecting a serious of social, economic and cultural phenomena in modern societies. Cinemas are considered to be an integral part of cities and they contribute to the definition of a local geography and identity. They also contribute to the preservation of the collective memory, since they constitute a significant social and cultural practice linked to a specific place, which acts as a common reference or landmark for many individuals.

Through this project we present a comprehensive solution for ticket booking in multiplexes. Theater management system, an online ticket selling software that is easy to understand, easy to use and offers the simplicity of fast point-and-click service to the customers.

1.2 METHODOLOGY

To implement these objectives we can use different programming languages like NODE JS, RUBY ON RAILS, PHP, .NET and many others.

But I used NODE JS(MERN STACK) for making this project. For frontend I used ReactJs and for backend I used ExpressJs and MongoDB for database.

1.3 SOFTWARE USED

We use NODE JS , Mongo DB and VS Code as a software to implement our Project.

1.3.1 NODE JS

- Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36.
- Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.
- Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on iOS X, Microsoft Windows, and Linux.
- Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

```
Node.js = Runtime Environment + JavaScript Library
```

1.3.2. MONGO DB

MongoDB is an open source database management system (DBMS) that uses a document-oriented database model which supports various forms of data. It is one of numerous nonrelational database technologies which arise in the mid-2000s under the NoSQL banner for use in big data applications and other processing jobs involving data that doesn't fit well in a rigid relational model. Instead of using tables and rows as in relational databases, the MongoDB architecture is made up of collections and documents.

How it works

- A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JavaScript Object Notation objects but use a variant called Binary JSON (BSON) that accommodates more data types. The fields in documents are akin to the columns in a relational database, and the values they contain can be a variety of data types, including other documents, arrays and arrays of documents, according to the MongoDB user manual.

- Documents, which also must incorporate a primary key as a unique identifier, are the basic unit of data in MongoDB. Collections contain sets of documents and function as the equivalent of relational database tables. Collections can contain any type of data, but the restriction is the data in a collection cannot be spread across different databases.

- The mongo shell is an interactive JavaScript interface to MongoDB which allows users to query and update data, and conduct administrative operations. The shell is a standard component of the open source distributions of MongoDB. Once MongoDB is installed, users connect the mongo shell to their running MongoDB instances.

- A graphical user interface (GUI) called MongoDB Compass gives users a way to work with document structure, conduct queries, index data and more. The MongoDB Connector for BI allows users to connect the NoSQL database to their business intelligence tools to visualize data and create reports using SQL queries.

1.3.3 VS CODE

Why did we use Visual Studio Code?

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

Edit, build, and debug with ease

- At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.
- For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.
- And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.
- VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.

CHAPTER 2

2.1 PROJECT DESIGN

My Project includes 7 Components which are as Homepage, Movie Detail, SignIn, Menu(After SighIn), Booking, Booking History and Contact Us.

Fig 2.1(a)

Fig 2.1(b)

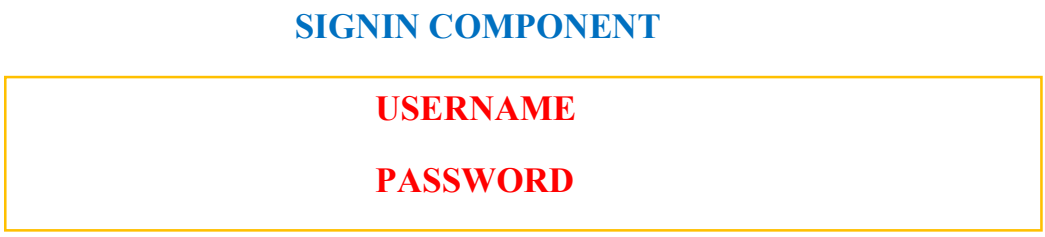

Fig 2.1(c)

MENU(After SignIn) COMPONENT

Fig 2.1(d)

BOOKING COMPONENT

Fig 2.1(e)

BOOKING HISTORY COMPONENT

Fig 2.1(f)

FOOTER COMPONENT

Fig 2.1(g)

CHAPTER 3

3.1 IMPLEMENTATION

So, first of all we create Schemas that we required to implement our project. We define Two Schemaa for my project.

- First Schema is for User who create an account on our web application .

  This Schema include two things :-

1. FirstName : First name of the User.
2. LastName : Last name of the User.
3. Email : Email address of the User
4. UserName : Which include username(alphamuneric) of User
5. Password : Password of the User.
6. BookedTickets : Number of tickets Booked by the User.

Also a unique Mongo Id is created for every User who create an account on the web portal.

- Second Schema is for Booking History. So this Schema include five things :-

1. UserName : UserName of User
2. Movie : Movie Name
3. ShowTime : Show Time of the Movie
4. SeatNo : Seat Number
5. Amount : Ticket Price
6. BookingTime: Time of Ticket Booking

Modules which we used to implement our project are as follows:

3.1.1 EXPRESS

To use express in our Project , First of all we use below command to install express and use it.

```
$ npm install express
```

where npm stands for node package manager.

Web Applications

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Performance

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love.

3.1.2 AXIOS

To use Axios in our Project , First of all we use below command to install ejs and use it.

```
$ npm i axios --save
```

Features

- Make XMLHttpRequests from the browser
- Make http requests from node.js
- Supports the Promise API
- Intercept request and response
- Transform request and response data

3.1.3 MONGODB

The official **MongoDB** driver for Node.js. Provides a high-level API on top of **mongodb-core** that is meant for end users.

To use mongodb in our Project , First of all we use below command to install mongoose and use it.

```
$ npm i mongodb --save
```

Importing

```
// Using Node.js 'require()'
const mongoClient= require('mongodb').MongoClient ;
```

Connecting to MongoDB

First, we need to define a connection. If your app uses only one database, you should use mongoClient.connect.

```
const mongoClient = require('mongoose').Mongolient;

var mongoUrl='mongodb://localhost/my_database'

mongoClient.connect(mongoUrl,(req,res)=>{});
```

Note: If the local connection fails then try using 127.0.0.1 instead of localhost. Sometimes issues may arise when the local hostname has been changed.

3.1.4 CRYPTO-JS

Crypto-js library is used for encrypting data.

To use CRYPTO-JS in our Project , First of all we use below command to install crypto-js and use it.

```
$ npm i crypto-js –save
var crypto=require('crypto-js');
```

3.1.5 BODY-PARSER

To use body-parser in our Project , First of all we use below command to install body-parser and use it.

```
$ npm install body-parser
```

API

```
var bodyParser = require('body-parser')
// parse applications/x-www-form-urlencoded


app.use(bodyParser.urlencoded({ extended: false }))
```

These all are the modules which we used in our project.

Let start our webpages implementation one by one :-

3.1.6 Header Component

- In the Header, There is a SignIn Button for user . When the user click on that button , the SignIn/SignUp Component will show up.

- Also there is a Search Bar from which you can search a movie and check if it is available or note.

- When the user login with their email id and password , there are four options available in the MenuBar(homepage) which includes Home, Booking History, Sign out and Delete your Acount.

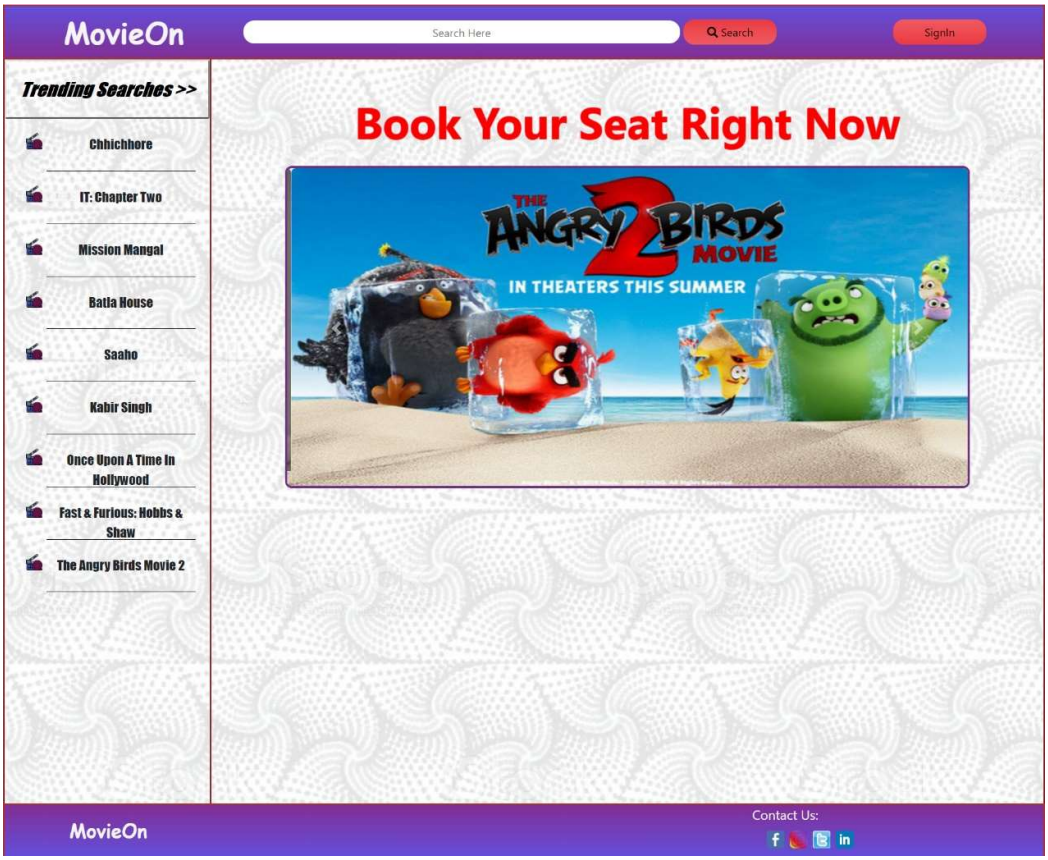

Fig 3.1.6(a) Header Before Login

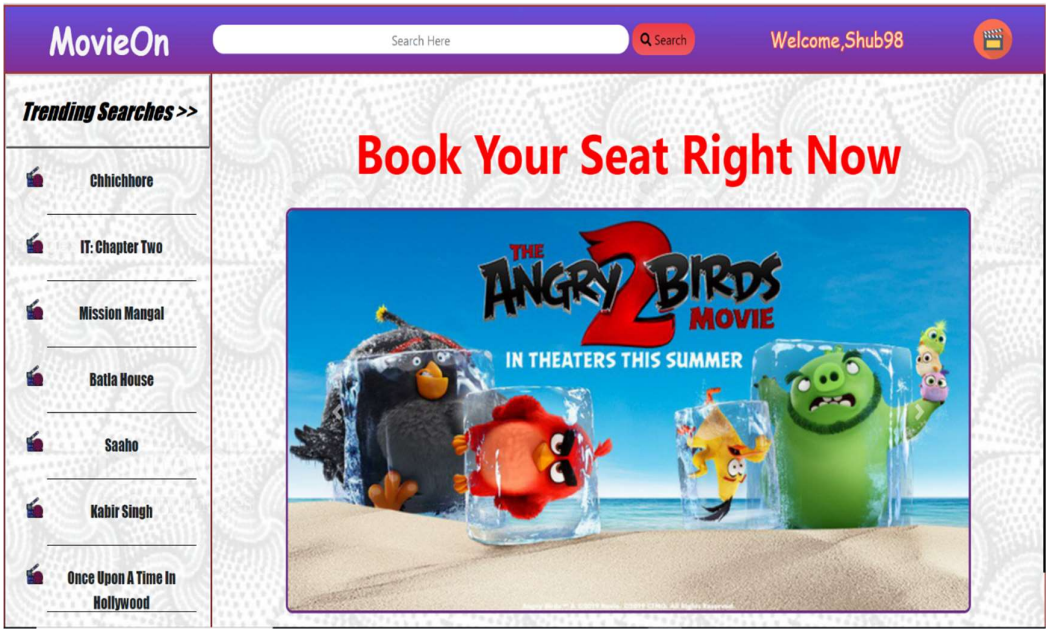

Fig 3.1.6(b) Header After Login

3.1.7 SignIn/SignUp Component

- When user click on Signin button, a signin component opens up. To book any ticket, First of all User has to create account . First User has to click on SignUp button to create an account , by providing his/her details. Then After signUp it will redirect to Login Page , where he has to fill her username and password through which he SignUp on the web Application.

- I use **Java Web Tokens**(JWT) for all this Login/SignUp criteria. After every login a token is generated for maintaining sessions with today's date which is valid only for Today. Tomorrow the token gets deleted and the user gets signed out automatically.

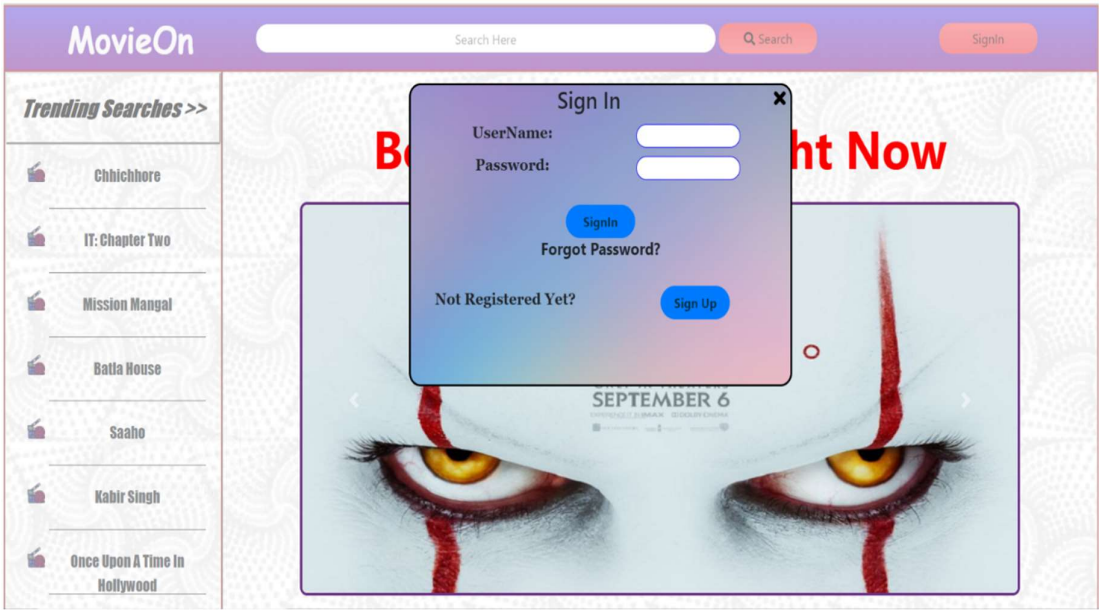

Fig 3.1.7(a) SignIn Component

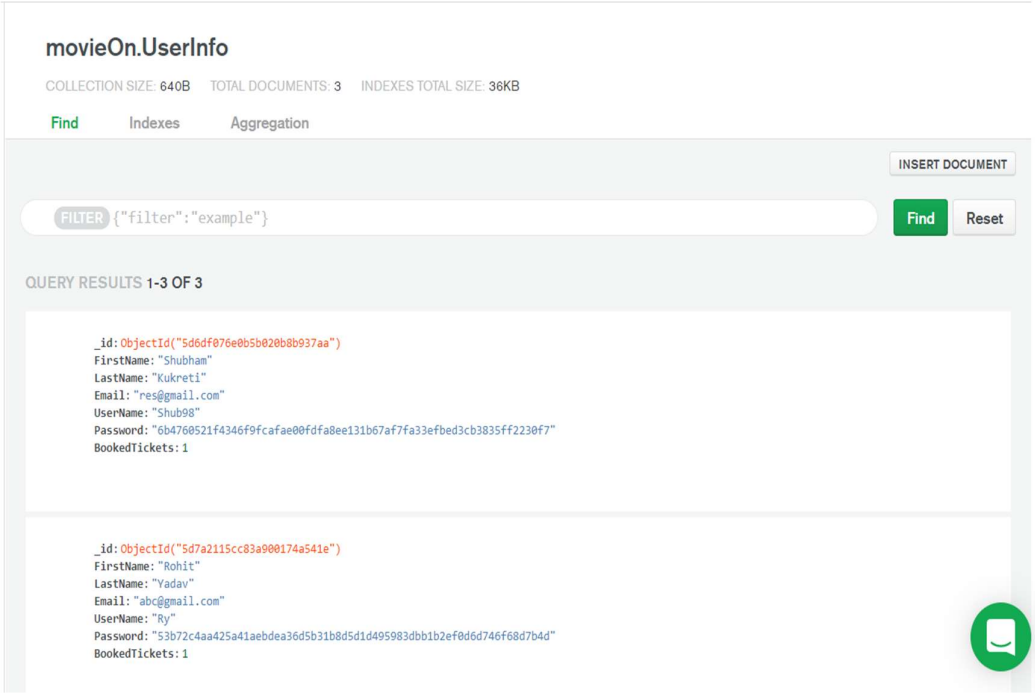

Fig 3.1.7(b) User Details Stored in Mongo DB

3.1.8 Menu Component

- After Signing In there is a button in header called menu button.
- On clicking the button there shows up a menu, which consists 4 options.
  - First, Home – On clicking it. The Website reloads and get back to homepage.
  - Second, Booking History- By clicking this user can see his/her latest booking on the website.
  - Third, Sign Out- By clicking this user can Sign Out from the Website.
  - Fourth,Delete Your Account- If User wants to delete his/her account, by clicking this button he/she can do it after confirmation.

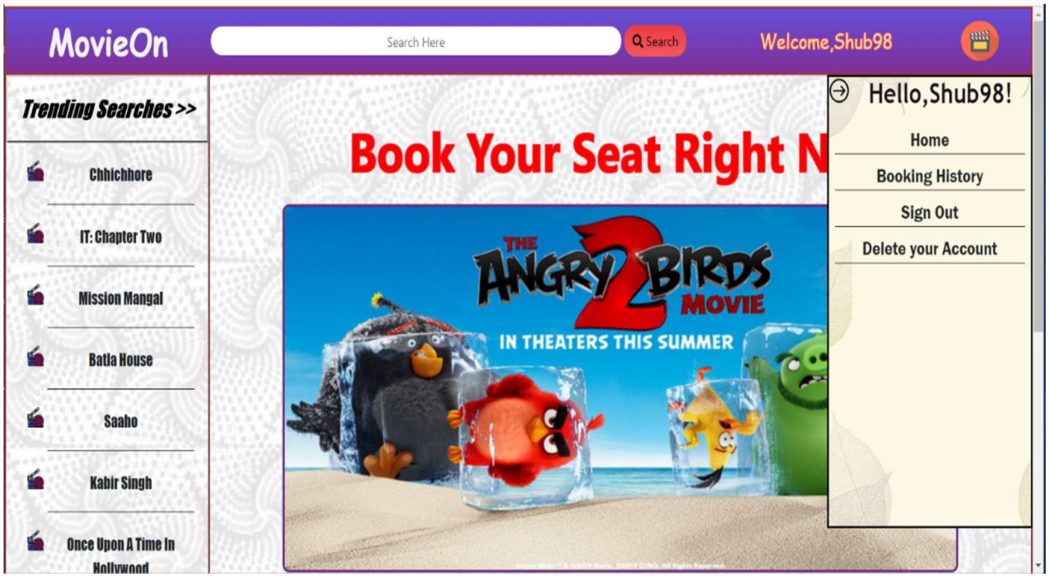

Fig 3.1.8(a)Menu Component

3.1.9 Middle Component

- It consist of two parts:-
  - Trending Searches/Latest Movies(middle-left): It Shows the List Of all the Trending Movies and Latest Movies.User can select a Movie from this list.
  - Movie Details(middle-right):It shows the details about the movie which the user has selected from the list like ratings,run time,language etc.

- In middle part(right) after selecting the movie there is a button called Book Tickets Which allow you to book the seat for your selected movie.

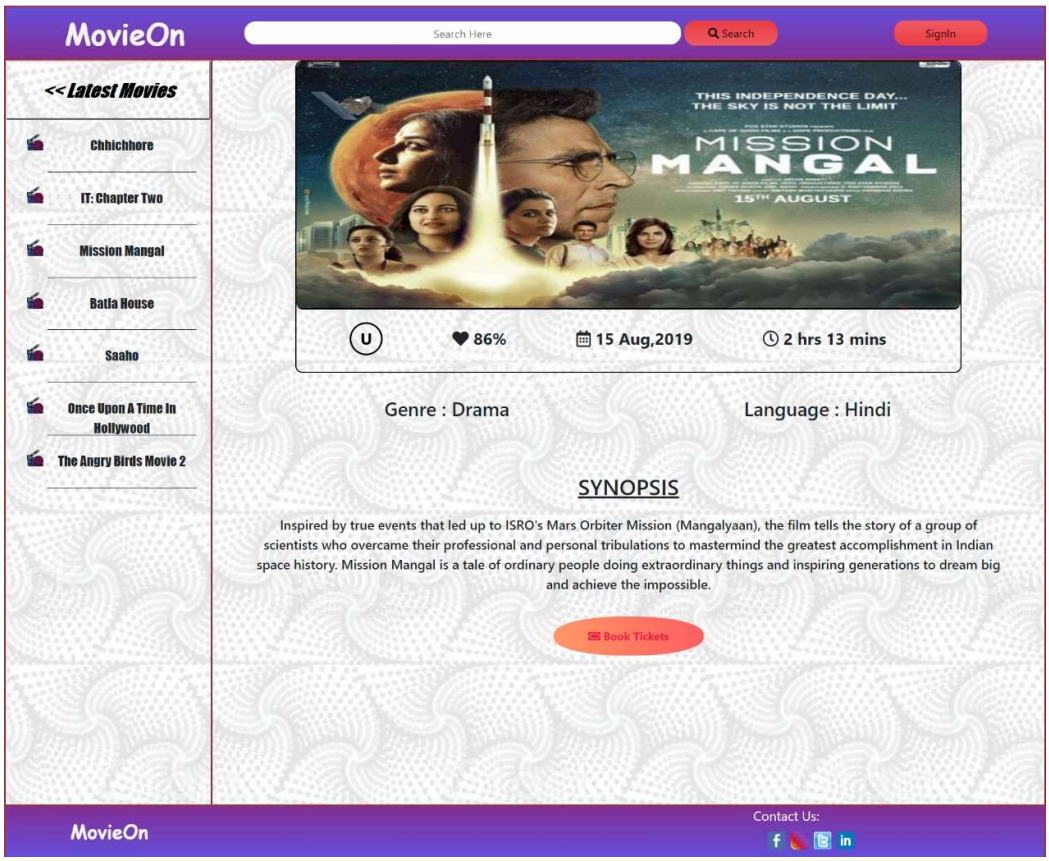

Fig 3.1.9(a) Movie Detail Component

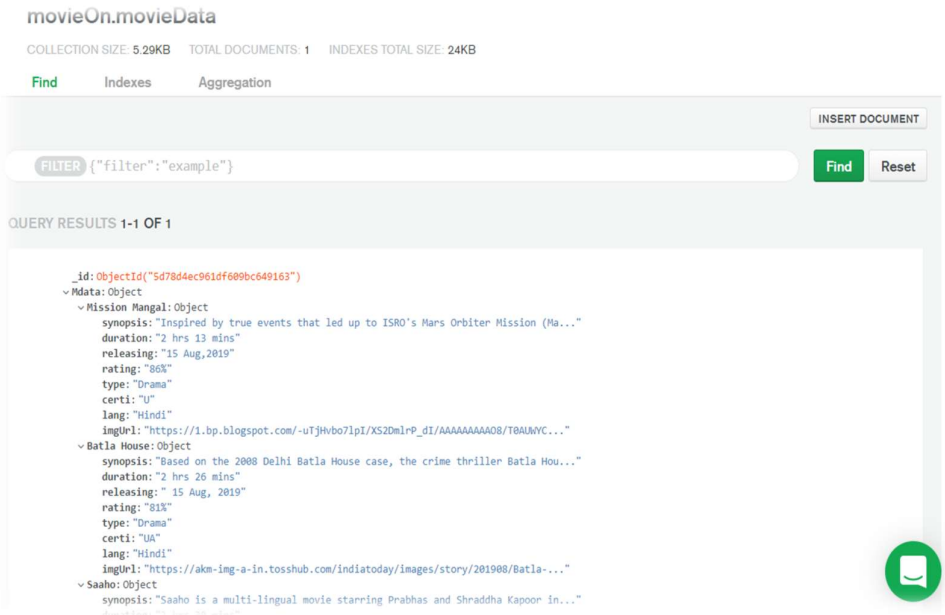

Fig 3.1.9(b) Movie details stored in Mongo DB

3.1.10 Booking Component

- When the user clicks on the Book Tickets button the Booking Component shows up.

- There are two inputs
    - First one is for show timings.User can select any timing among the options available.
    - Second one is for Seat Selection.User can enter the number of his/her choice according to the image given along side.

- Now there is a button called Check Seat Availability.Which checks the seat nunber the user has entered is available or not.

- If the seat is available, The user can book by making the payment.

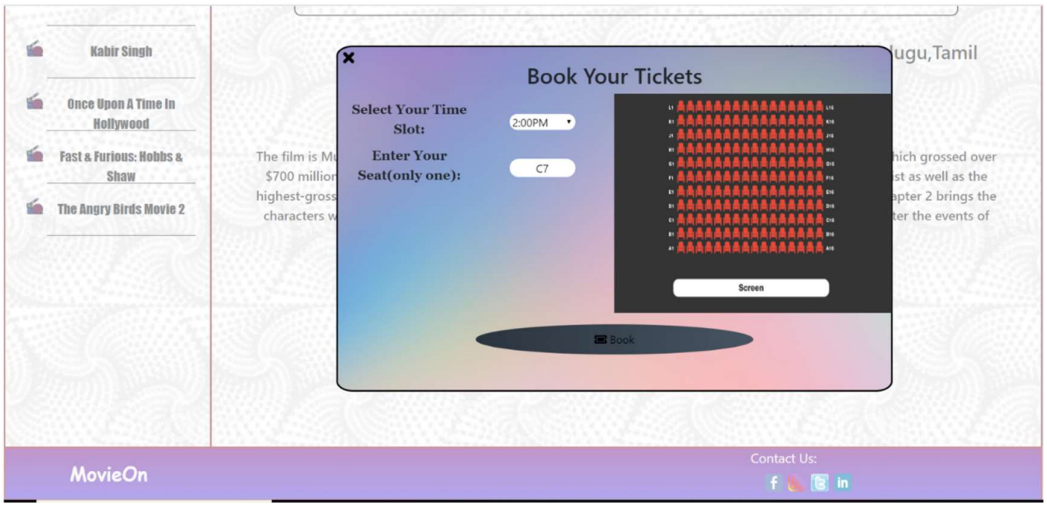

Fig 3.1.10(a) Booking Component

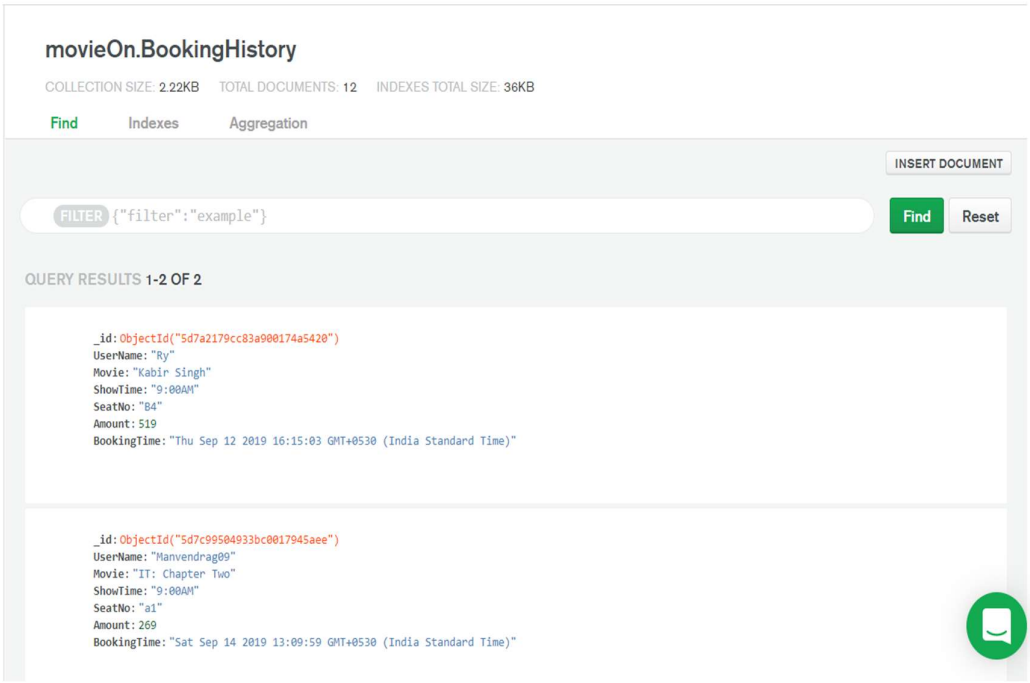

Fig 3.1.10(b) Booking Details stored in mongodb

CHAPTER 4

4.1 CONCLUSION AND FUTURE SCOPE

4.1.1 CONCLUSION

- There is a wide acceptance of online platform in today era in all over the world due to internet facilities available
- Online platform complete all the needs of people in all the situation and provide many features to all users.
- We provide a safe and secure applicaction so that users can't have any issue regarding our application.
- Always keep the data of the users encrypted using different algorithms and keep updating our application time to time.
- Outperformed most competitors .

4.1.2 FUTURE SCOPE

This system is initially developed for Project purpose. But without making any changes in this system it can be expanded for all big Companies. This system is capable of adopting any new feature. This system can be upgraded with new features by making very less changes.

The new features that can be added to this system without making vast changes are:

- Adding more features to the Application like Payment gateway etc.
- Adding More Features to Booking Interface.
- Adding of movie trailers with the details.
- Make our frontend more interactive through which user can easily interact with our application.
- We are also think about to take a bill no or other user side surety to complete that job.
- Connecting our application with real servers.

REFERENCES

- https://nodejs.org/en/
- https://www.mongodb.com/
- https://www.npmjs.com/
- https://www.npmjs.com/package/mongodb
- https://www.w3schools.com/
- https://getbootstrap.com/