**ONLINE MOVIE TICKET BOOKING**

The project report is submitted by S.RAJESH KANNA a student of NIIT

**ACKNOWLEDGMENT**

I also express my sincere thanks to **Mr.** **DHIRAJ KUMAR** for permitting me to take up this project.

I am much privileged to have **Mr.VASUDEVA KUMARAN TECH MENTOR in NIIT** as guide for this project. I heartily thank his for overall help and support at all stages.

**INDEX**

 **Abstract.**

 **Project introduction.**

 **Problem Analysis.**

 **Administrator Section.**

 **Customer Section.**

 **Non-Functional Requirment.**

 **Used Tools And Platfrom.**

 **Project Module.**

 **MongoDb.**

 **Yeoman Generator.**

 **Express js.**

 **Nodejs.**

 **TMDb API.**

**Testing.**

 **Validations.**

 **System Security Measures.**

 **Screen shot.**

 **Bibliography.**

**Abstract:**

My Show is aimed to provide information of the movie in more usable way, according to which they can book the tickets in registered theatre by user. It proceeds through a sequence of well designed forms provided with validations to ensure consistency, reliability and most importantly correctness of information. Here, User is responsible for the registration of the User Account, The user can Change password, can view number of seats available and can book tickets. Whereas, Administrator is responsible for maintaining right information about the movies, seat availability, employee details and many more things.

The salient feature of MyShow is that website recommended best suitable theatre to the user, which is helpful for tourist or any outside people who don’t know about the current city. Another feature of the project is that is user can book ticket only by less clicks of options.

**INTRODUCTION**

**Goals and Objectives**

The main purpose of our online ticket booking system is to provide an alternate and convenient way for a customer to buy cinema tickets. It is an automatic system. After the data has been fed into the database, the staff does not need to do anything with the order once it is received through the system. In fact, there is similar system on the internet, but there is no refund method found in the existing system. The goals of our system are:

1 To provide a anytime anyplace service for the customer

2 To minimize the number of staff at the ticket box

3 To promote the film on the internet

4 To increase the profit

5 To obtain statistic information from the booking record.

**About this project**

Our online E-Ticket System (ETS) is a web-based system. The customers can buy ticket online and cancel the seat at a suitable time (2 days before the show to 1hour before the show). To enhance the refund function, all the customers have to registration become a member before buying ticket.

Staff can use the system to insert and delete data (e.g. film description , time table) which will update the webpage(webpage are dynamic page, changing according to the data in database). Also, staff can check the statistic information from the system.

General Requirements (functional)

1. The web page (e.g. The time table page, the main page) will be generated automatically according to the data in database.

2. A way in which the customer can create its own account(member registration).

3. A way in which the users (both customer and staff) can login to the system to perform different operation.

4. A way in which the customer can modify its own data.

5. A way in which the customer can place a order by just clicking the seat (which is shown on the screen) and insert some card data.(some simple operation)

6. A way in which the customer can cancel the order and get the refund.

7. A way in which the customer can check the ticket record according to the transaction number.

8. A way in which the staff can use the system to add data(e.g. film description) to the database.

9. The system can verify the data before transaction.

10. The system can generate the time table automatically(by just input the length of the film) or the time table is set by the staff.(2 operating mode for the staff to insert data).

11. The system can generate some statistic information according booking and ticket selling record.

12. Users can check film data by clicking on a certain film on main page(e.g. The cinema which will show this films).

13. Users can check a cinema data by clicking on a certain cinema on main page(e.g. which film is now showing)

 If system, which is going to be developed, is complex in nature the goals of the entire system could not be easily comprehended.

Hence the need for a more rigorous system analysis phase arose.

User Developer Managers

Generate

Request

Problem

Statement

**User**Interviews

Build

Models

Experience

Domain Knowledge

Object Model

Functional Model

**Problem Analysis**

 The basic aim of problem analysis is to obtain clear understanding of the needs of the clients and the users, what exactly is desired from the software, and what the constraints on the solution are. Analysis leads to the actual specification.

**Administrator section:**

 This section can be accessed by providing administrator password.

In this section the administrator can save the information related to movie,seats,booking,payment etc.

 In this section the administrator can edit the information related to movie,seats,booking,payment etc.

**Customer section:**

 Customer can book the movie tickets by selecting the seats of his/her choice.

 Customer can pay for tickets online by credit card.

**Non Functional Requirements :**

 It consists of following parameters :-

**Reliability**: The system will consistently perform its intended function.

For eg. The important information must be validated.

**Efficiency**: Unnecessary data will not be transmitted on the network and database server will be properly connected.

**Reusability**:The system can be reused in any organization or site of the same group, by defining the organization master definition under software license agreement.

**Integrity** : Only System Administrator has rights to access the

database, not every user can access all the information. Each user will be having rights to access the modules.

**Used Tools AndPlateform**

**Software Specification:**

**Front-end Tool: - HTML5,CSS,ANGULARJS**

 User friendly

 Low Cost Solution

 GUI feature

 Better designing aspects

**Back-end Tool: - MONGODB NO SQL**

 Security

 Portability

 Quality

**Platform**

Windows platform like: Windows 8

**Hardware Specification:**

 Intel Core i5

 Processor Speed – 2.60 GHz or above

 RAM – 4GB

 HDD – 1TB

 Monitor-15.6”

 Mouse- Normal

Keyboard- Normal

**Project Modules**

**Login Module**

This module is for both type of users(customers and admin).In this module according to the type of user(customer or admin) the further links and operations will be provided.

**Customer Module**

 As soon as a visitor registers himself as a customer,the customer can now book the movie tickets and pay for them online.

**Booking Module**

 In this module movie ticket is booked for a customer. This module contains all the information related to booking. As soon as the customer request is complete, all the booking details are displayed to him.

**Payment Module**

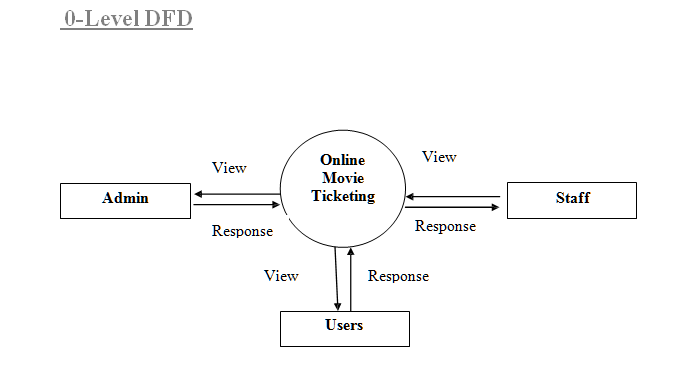
 This is the most important module because it deals with the payment of the tickets booked in the booking module. The customer can pay for the tickets before the show by cash

payment and if he wants to pay online, he can pay for the tickets by credit card.

= Source or destination of data

= Data flow

=process that transforms data



**MONGODB**

* MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time
* The document model maps to the objects in your application code, making data easy to work with
* Ad hoc queries, indexing, and real time aggregation provide powerful ways to access and analyze your data
* MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use
* MongoDB is free and open-source, published under the GNU Affero General Public License

**Yeoman Generator**

Yeoman helps you to kickstart new projects, prescribing best practices and tools to help you stay productive.

To do so, we provide a generator ecosystem. A generator is basically a plugin that can be run with the `yo` command to scaffold complete projects or useful parts.

Through our official Generators, we promote the "Yeoman workflow". This workflow is a robust and opinionated client-side stack, comprising tools and frameworks that can help developers quickly build beautiful web applications. We take care of providing everything needed to get started without any of the normal headaches associated with a manual setup.

With a modular architecture that can scale out of the box, we leverage the success and lessons learned from several open-source communities to ensure the stack developers use is as intelligent as possible.

As firm believers in good documentation and well thought out build processes, Yeoman includes support for linting, testing, minification and much more, so developers can focus on solutions rather than worrying about the little things.

**Tools**

The Yeoman workflow comprises three types of tools for improving your productivity and satisfaction when building a web app: the scaffolding tool (yo), the build tool (Gulp, Grunt etc) and the package manager (like npm and Bower).

**Express js**

Express.js, or simply Express, is a [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) for [Node.js](https://en.wikipedia.org/wiki/Node.js), released as [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software)under the [MIT License](https://en.wikipedia.org/wiki/MIT_License). It is designed for building [web applications](https://en.wikipedia.org/wiki/Web_application) and [APIs](https://en.wikipedia.org/wiki/API).[[2]](https://en.wikipedia.org/wiki/Express.js#cite_note-ExpressJS-2) It is in fact the standard server framework for Node.js

**Nodejs**

As an asynchronous event driven JavaScript runtime, Node is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection the callback is fired, but if there is no work to be done, Node will sleep.

**TMDb API**

The API service is for those of you interested in using our movie, TV show or actor images and/or data in your application. Our API is a system we provide for you and your team to programmatically fetch and use our data and/or images.

**Finding Data**

There are 3 ways to search for and find movies, TV shows and people on TMDb. They're outlined below.

/search - Text based search is the most common way. You provide a query string and we provide the closest match. Searching by text takes into account all original, translated, alternative names and titles.

/discover - Sometimes it useful to search for movies and TV shows based on filters or definable values like ratings, certifications or release dates. The discover method make this easy. For some example queries, and to get an idea about the things you can do with discover, [take a look here](https://www.themoviedb.org/documentation/api/discover).

/find - The last but still very useful way to find data is with existing external IDs. For example, if you know the IMDB ID of a movie, TV show or person, you can plug that value into this method and we'll return anything that matches. This can be very useful when you have an existing tool and are adding our service to the mix

**Testing**

**MOCHA**

Mocha is a feature-rich JavaScript test framework running on Node.js and in the browser, making asynchronous testing simple and fun. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases

**CHAI**

Chai is a BDD / TDD assertion library for node and the browser that can be delightfully paired with any javascript testing framework.

**KARAMA**

A tool called Karma is a JavaScript test runner created by the AngularJS team. Jasmine is thetesting framework that we talked about in the getting started with unit testing for AngularJS post, andKarma provides helpful tools that make it easier to us to call our Jasmine tests whilst we are writing code.

**PROTRACTOR**

Using the Protractor Automation Tool to Test AngularJS Applications. ... Protractor, formally known as E2E testing framework, is an open source functional automation framework designed specifically for AngularJS web applications. It was introduced during AngularJS 1.2 as a replacement of the existing E2E testing framework

**VALIDATIONS**

 No record can be saved till all the necessary entries are done.

 Only administrator can perform sophisticated tasks like new theater and/or delete an existing theater according to cities etc.

 For security purposes the E-mail of user is required in case he/she forgets his/her password and wants to retrieve that.

**System Security Measures**

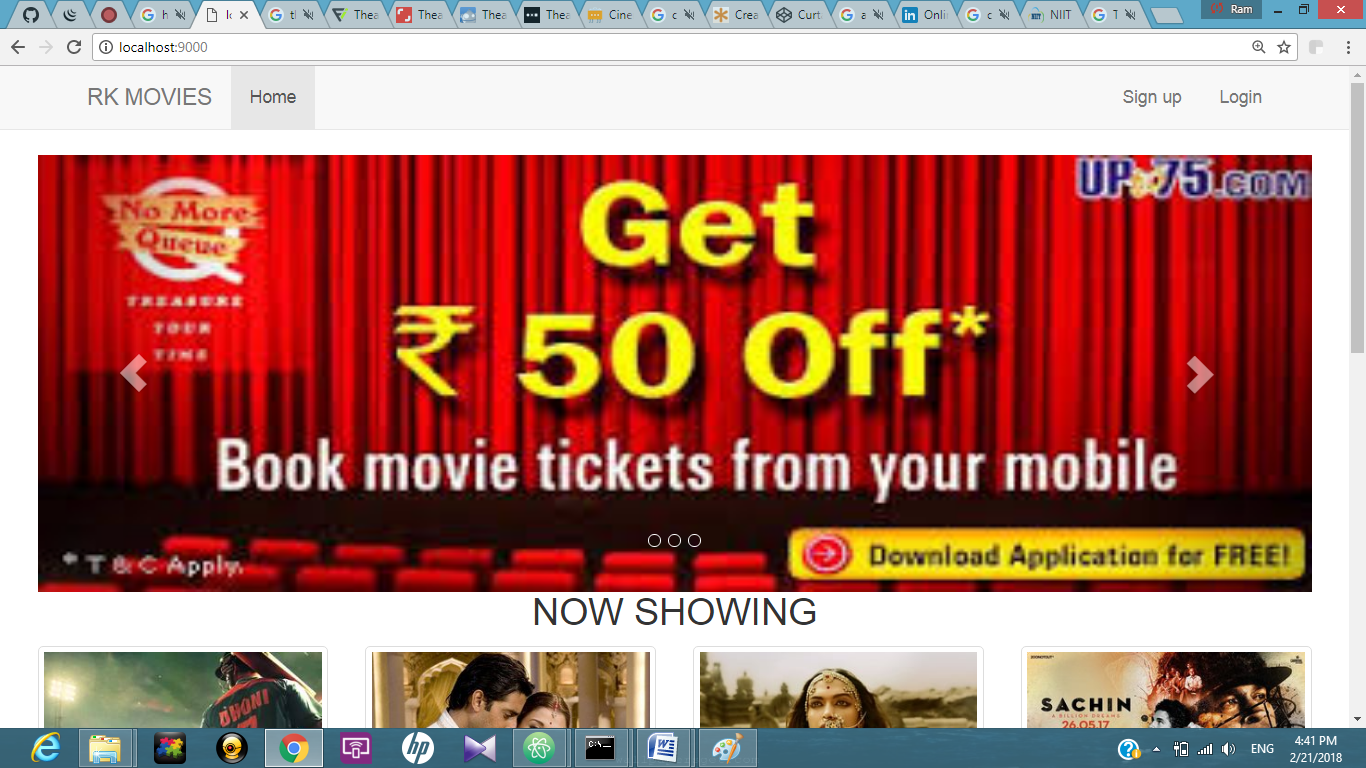
 Security prompting the user for a userid and password in our application is a potential security threat. So credential information is transferred from the browser to server are encrypted.

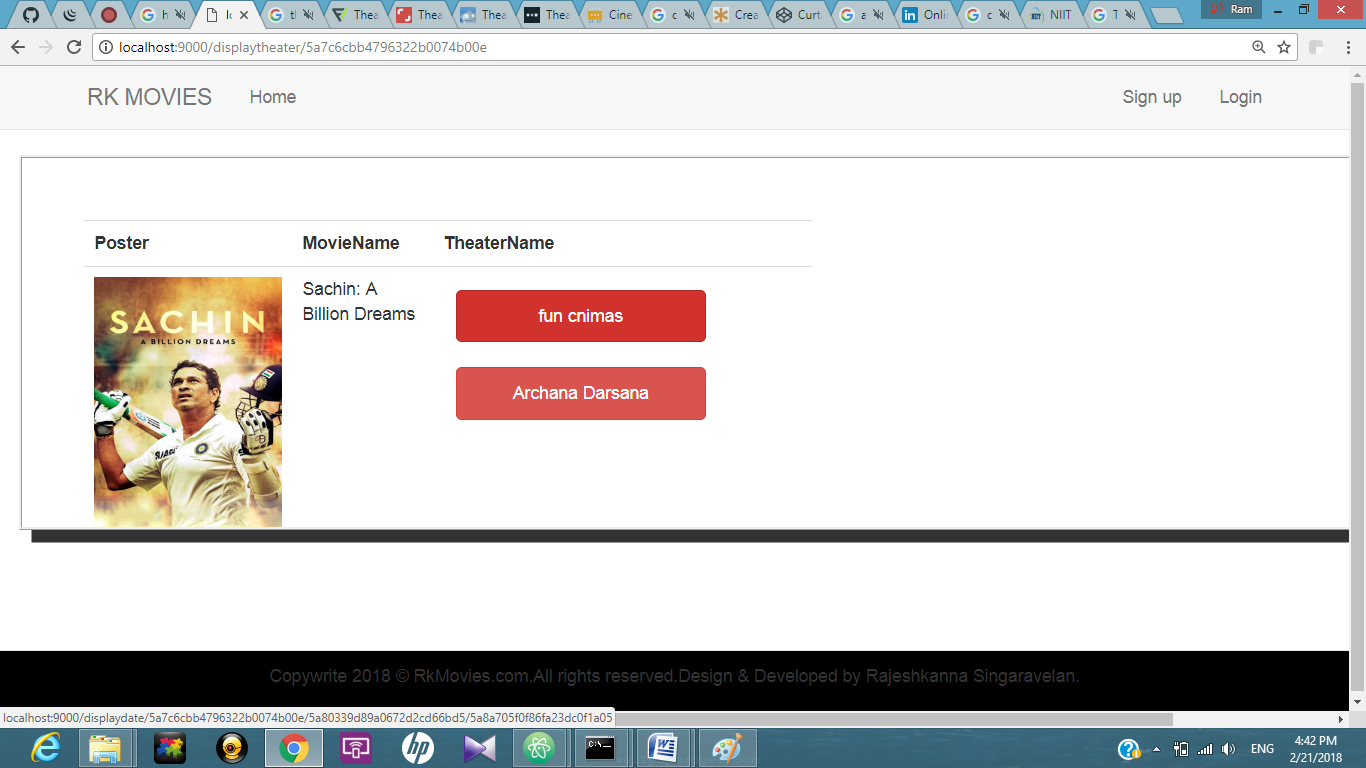
 Cookies are an easy and useful way to keep user-specific information available. However, because cookies are sent to the browser's computer, they are vulnerable to spoofing or other malicious use. So we follow these guidelines:

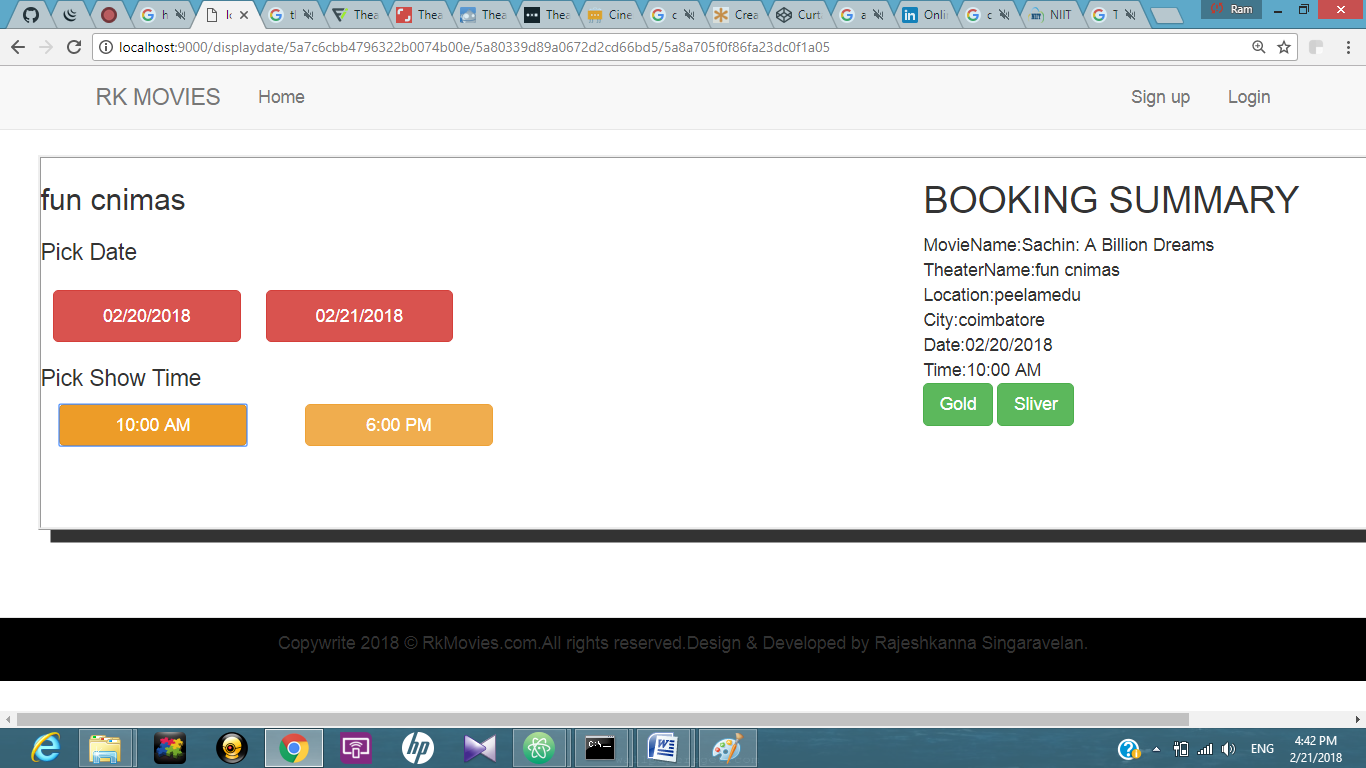
 Do not store any critical information in cookies. For example, do not store a user's password in a cookie, even temporarily.

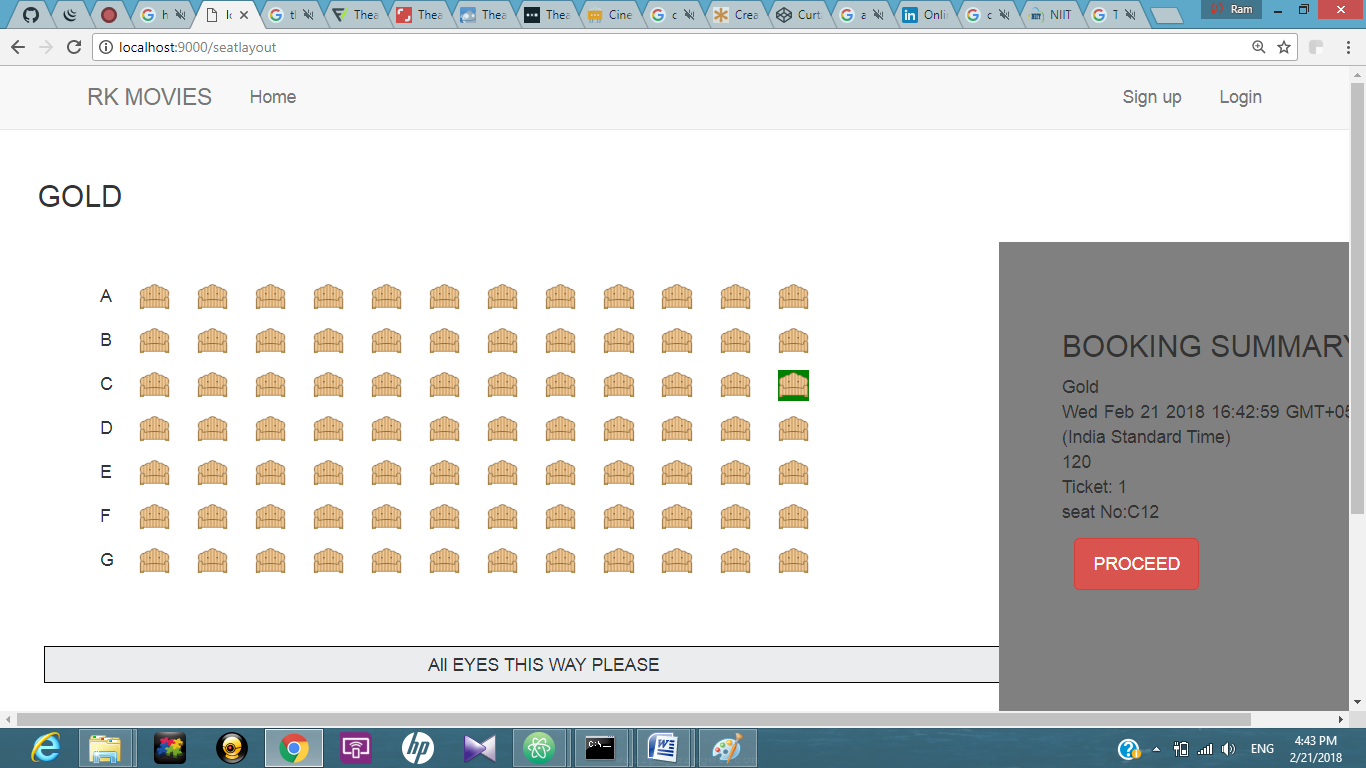
 Avoid permanent cookies if possible. Consider encrypting information in cookies. Set expiration dates on cookies to the shortest practical time we can.

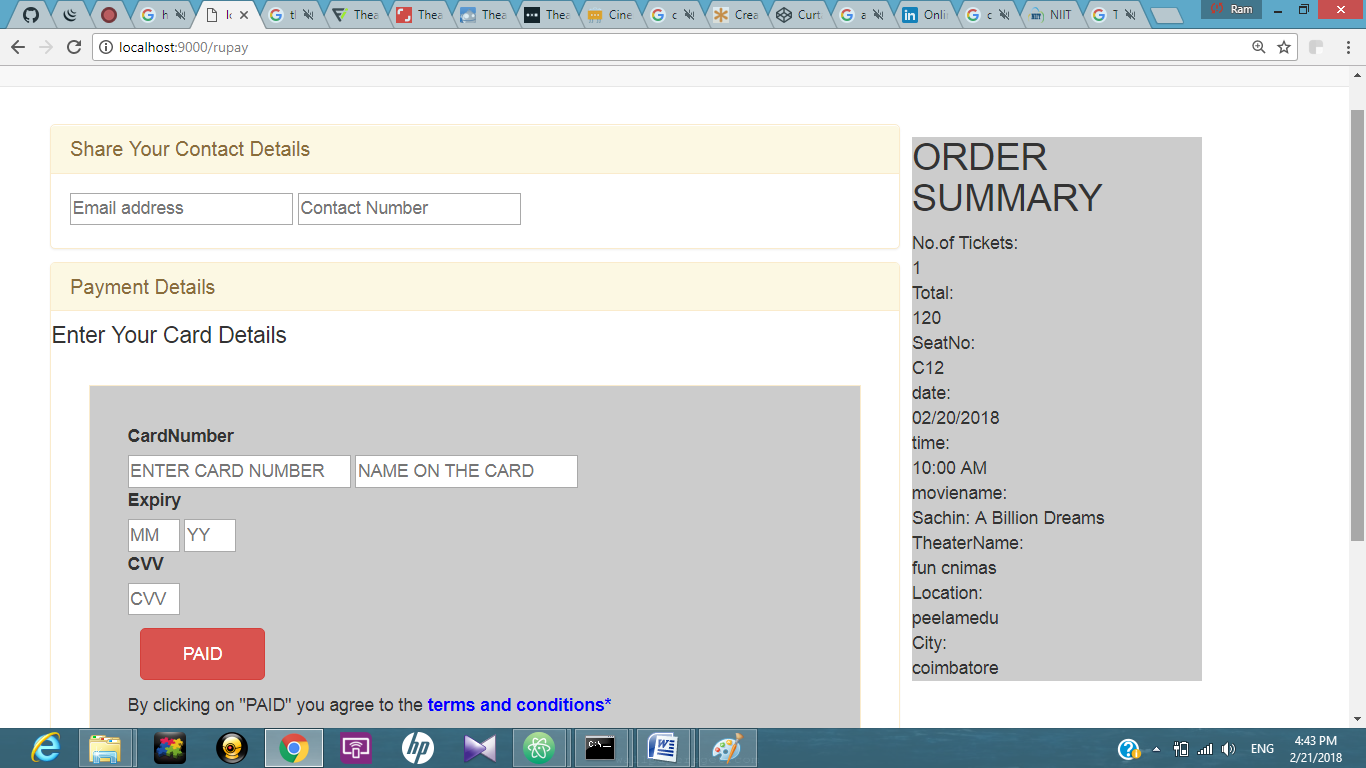
**SCREEN SHOT**

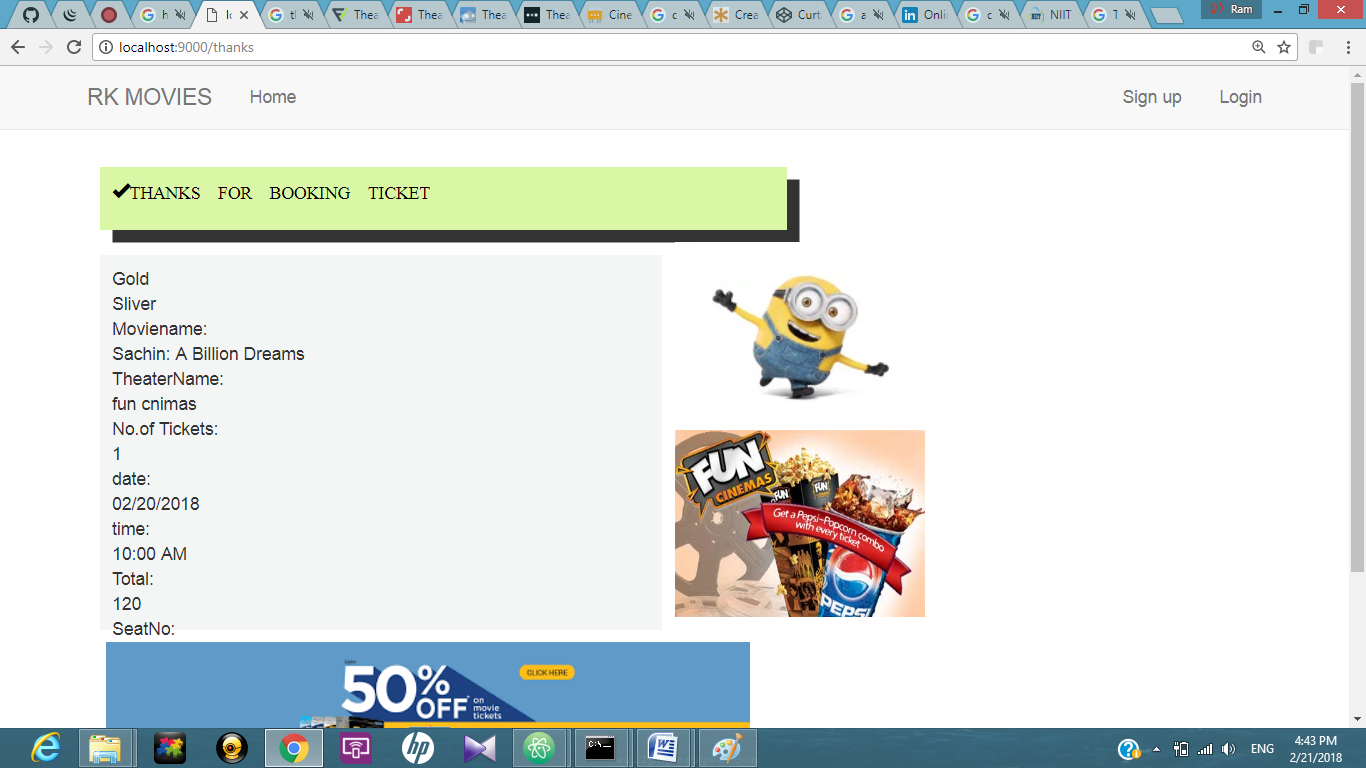


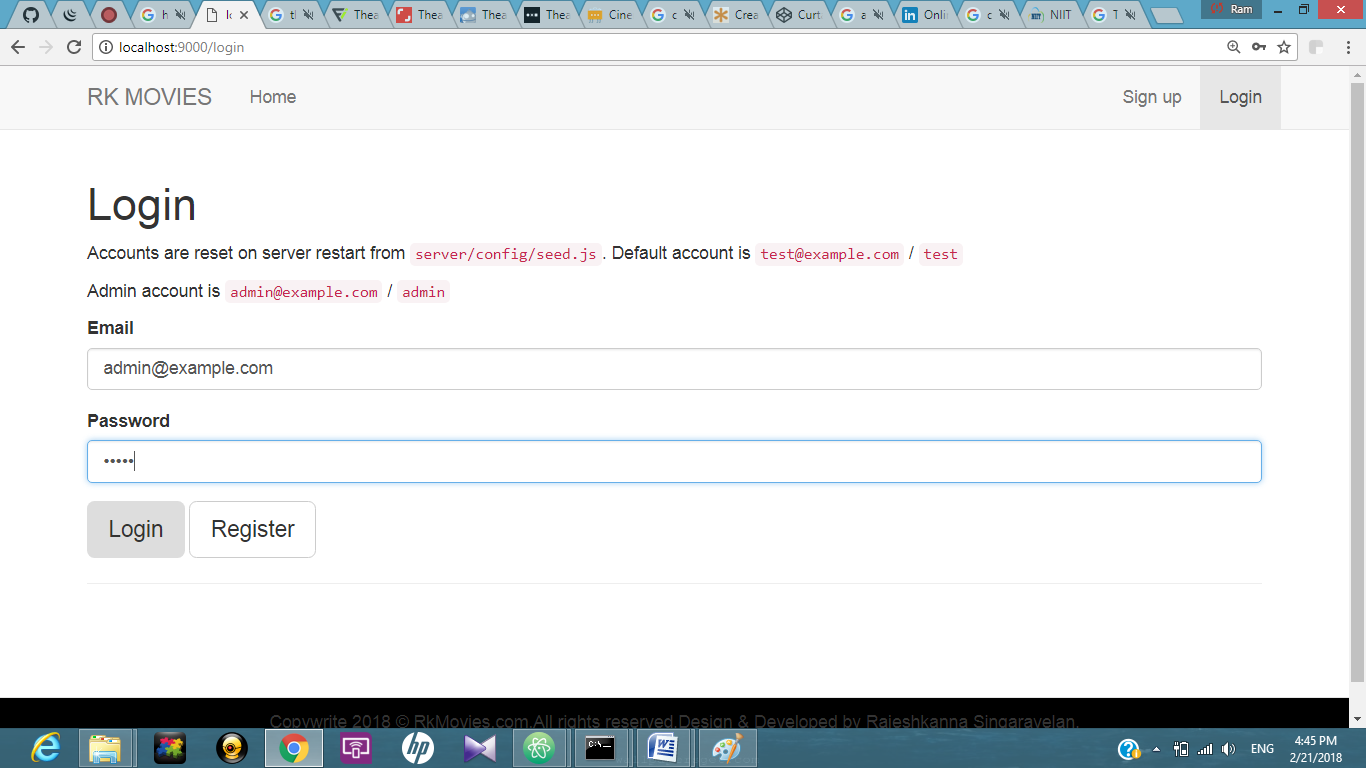


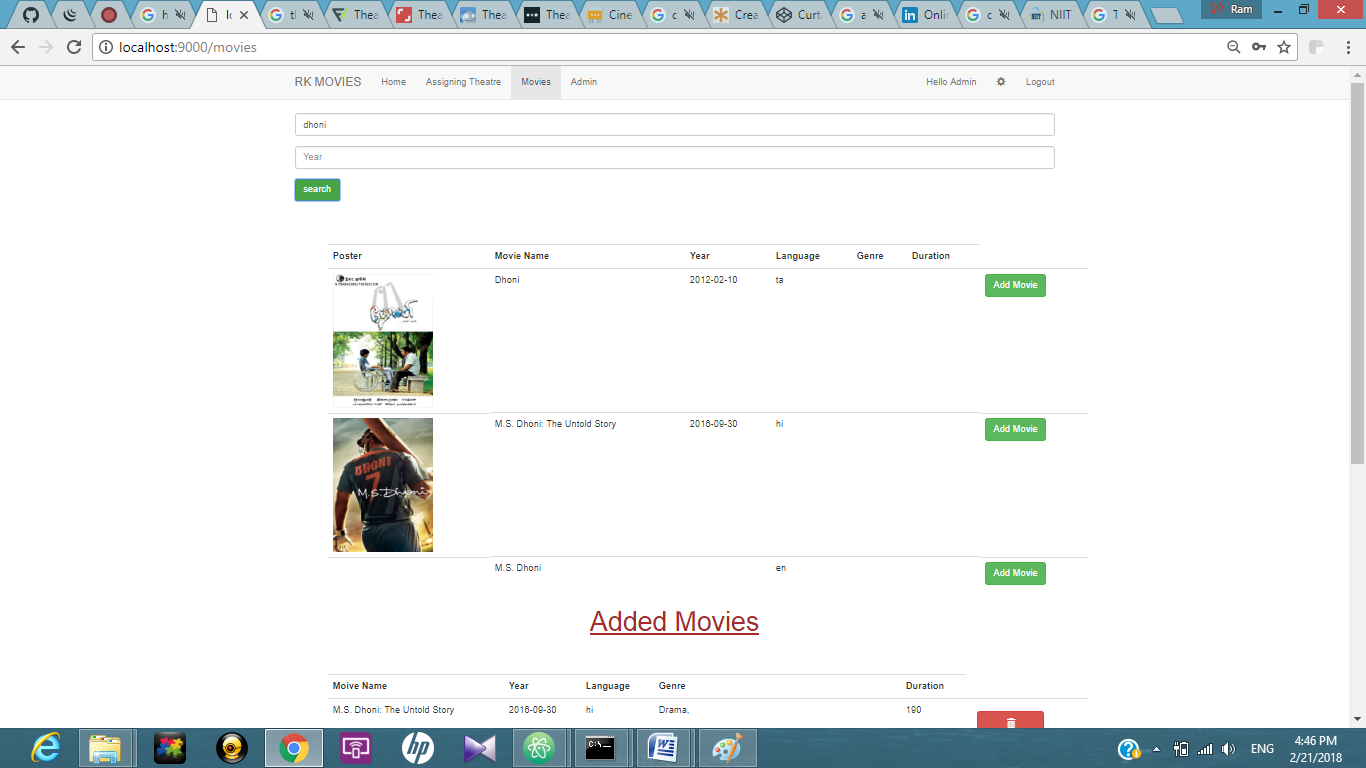


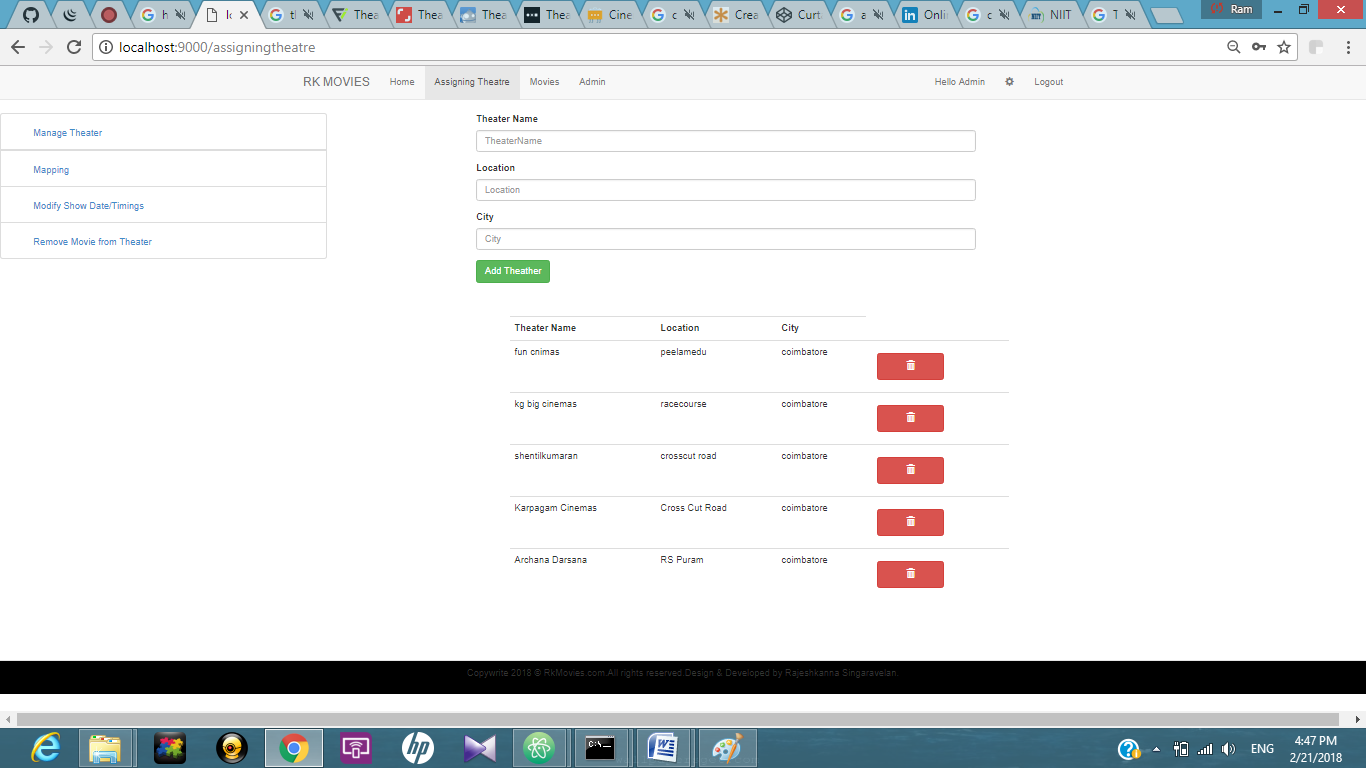


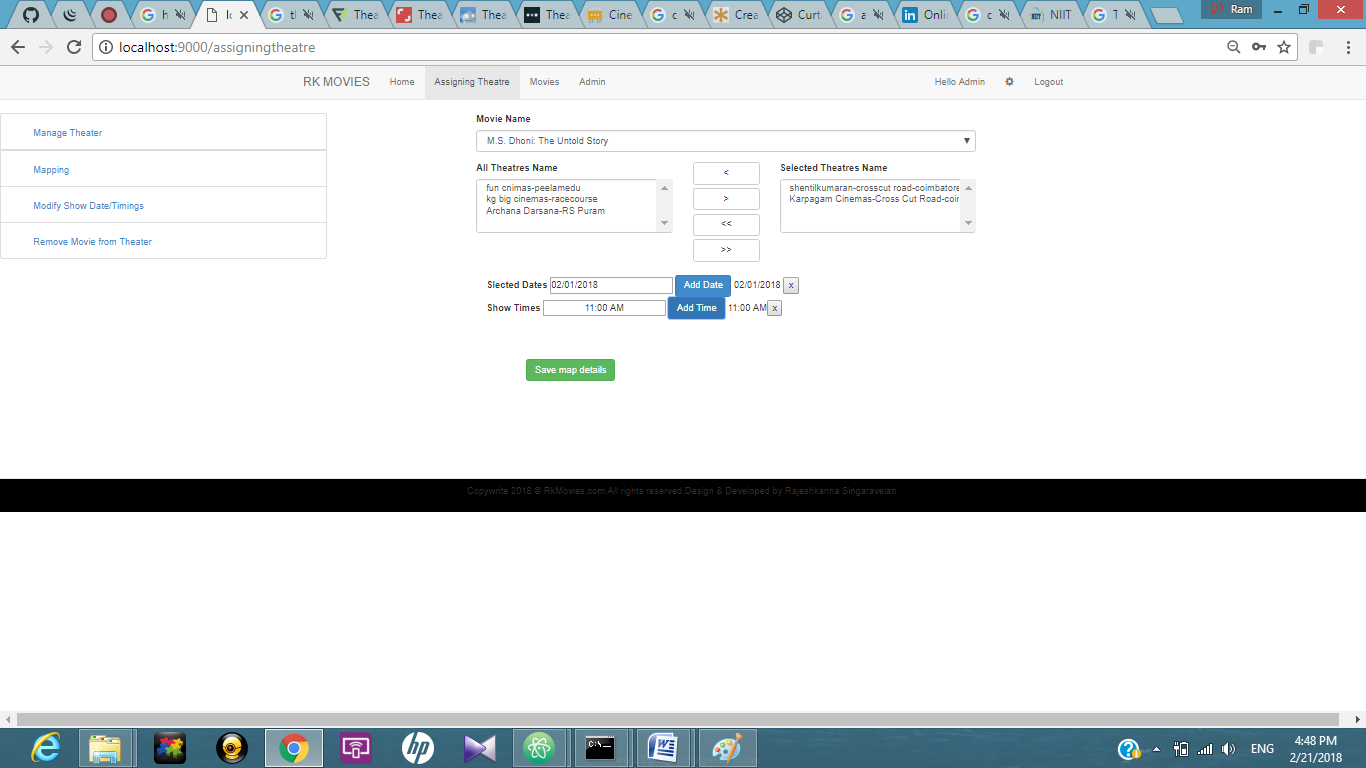


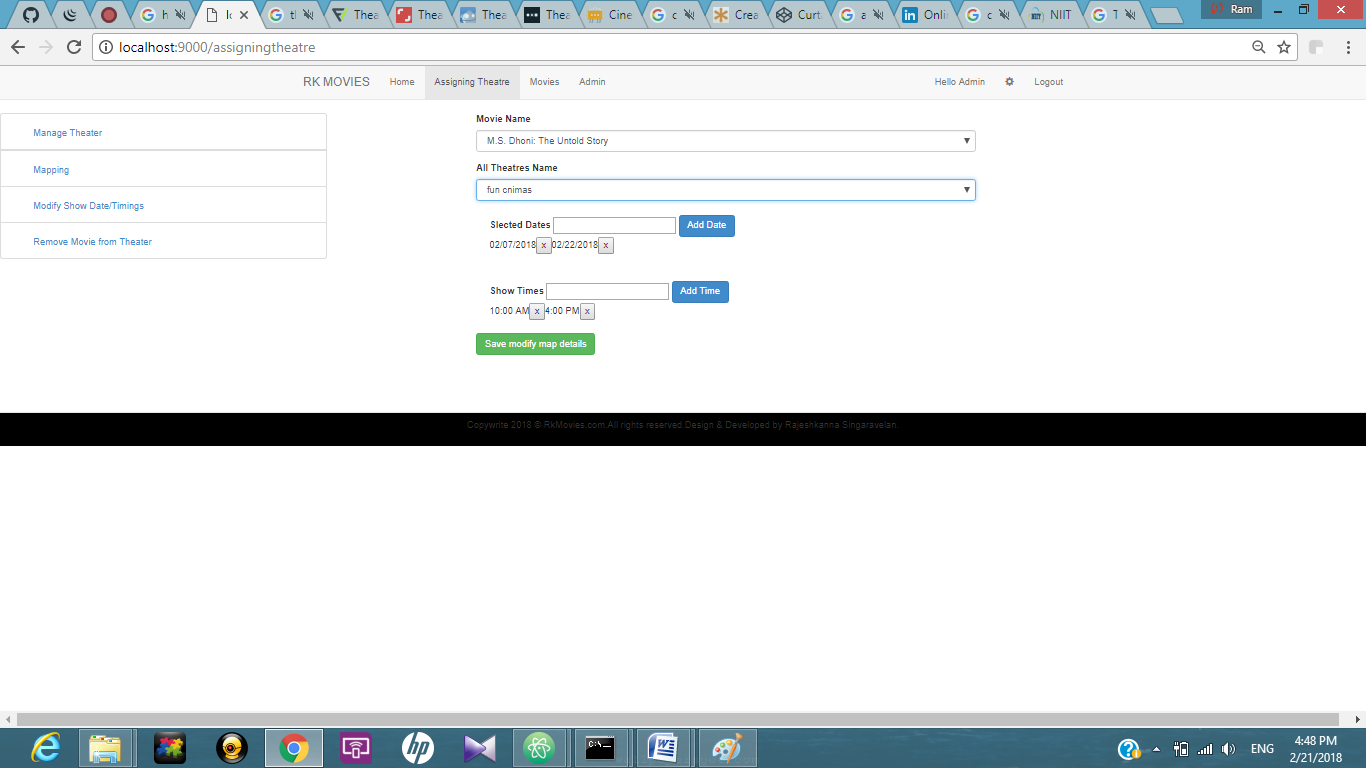


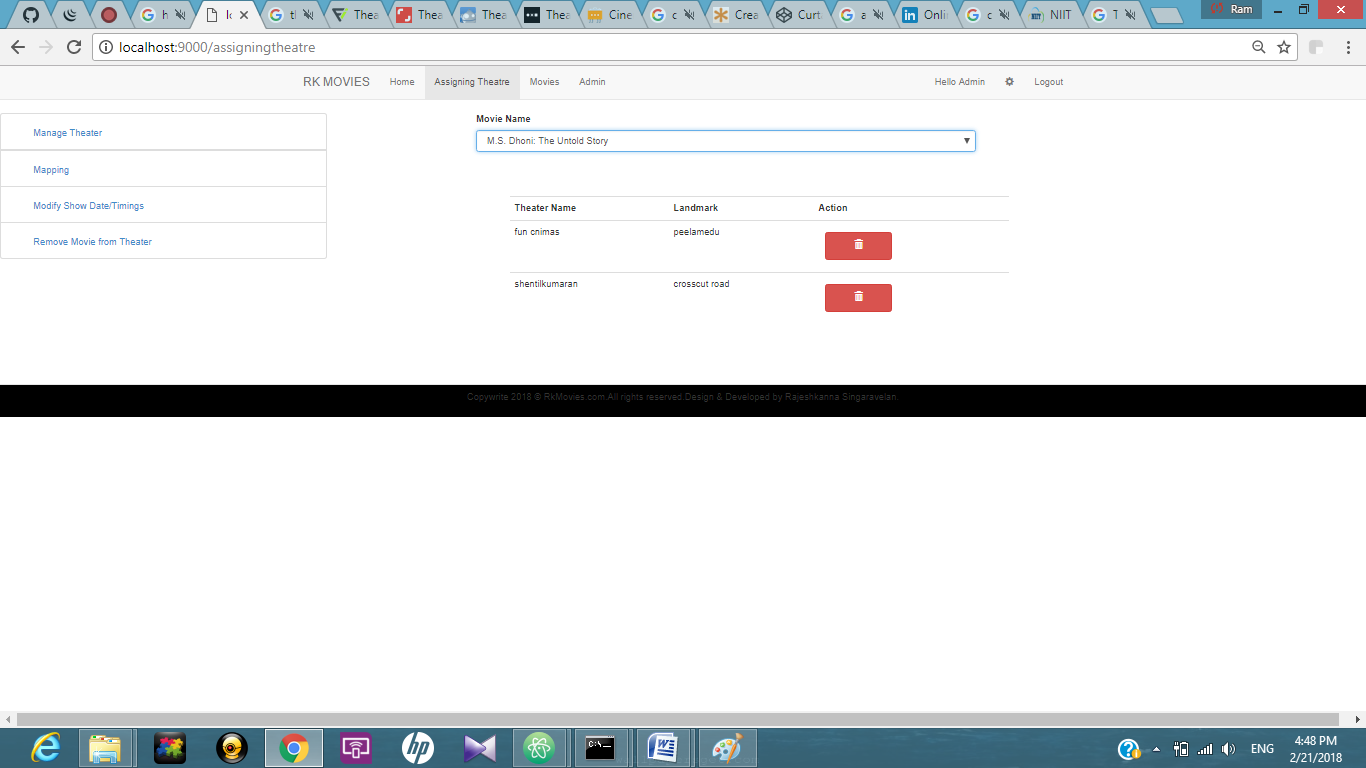












**BIBLIOGRAPHY**

 [WWW.WIKIPEDIA.COM](http://WWW.WIKIPEDIA.COM)

 WWW.Google.com

www.angularjsbook.com