**Online Movies Booking System**



Ashish Pokhrel

00172912

Computing Project

Level 5 in Computing

Softwarica College of IT and E-Commerce

Kathmandu, Nepal

27/04/2019

**PHP (Laravel Framework) with MYSQL**

**Acknowledgement**

I would like to special thanks for module leader **Mr. Kiran Rana** for supporting and guidance in completing my project.

I would also like to thanks to all my teachers, friends who have helped me with valuable suggestion and also all respondents whose response are important in completion of my project. I have fulfil the module leader expectation from my side.

Ashish Pokhrel

Batch 21

27th April, 2019

**Abstract**

The Purpose of the project was to book movie ticket through net connection, analysis data of active customers, and to use technology to book movie ticket instead of manual or paper-based booking.

The project main aimed at to develop online movie booking system for cinema hall named as **“City Movies”**. Through this Movie Booking system, customer can see lists of movies which are showing currently with showtime. Customer can book movies ticket without visiting to cinema hall through this system via internet. The main aim of this organization is to provide all information related to shows, movies, price and available facility in cinema hall etc. through use of web portals form anywhere anytime via internet connection.

The System has three users i.e. Register User who can book movie, read showtime, movies details, etc. Other one is Admin who add, edit, and delete Movie, Show, screen, hall, etc. to the database.

I have chosen Waterfall methodology for development methodology and for implementation I have used Laravel 5.7 version framework of php with XAMPP server with MySQL database with MVC design Pattern for architectural design.

Table of Contents

[CHAPTER 1 - INTRODUCTION 8](#_Toc6857813)

[1.1 Overview of project 8](#_Toc6857814)

[1.2 Justification of Project 8](#_Toc6857815)

[1.2.1 Background of Project 8](#_Toc6857816)

[1.3 Problem Statement 9](#_Toc6857817)

[1.4 Features 9](#_Toc6857818)

[1.5 Aims and Objectives 10](#_Toc6857819)

[1.5.1 Aims 10](#_Toc6857820)

[1.5.2 Objectives 10](#_Toc6857821)

[1.6 Scope 10](#_Toc6857822)

[CHAPTER 2 – Analysis 11](#_Toc6857823)

[2.1Object Oriented Analysis 11](#_Toc6857824)

[2.2 Development Methodology 11](#_Toc6857825)

[2.3 Merits and Pitfalls of Project 13](#_Toc6857826)

[2.2.1 Merits 13](#_Toc6857827)

[2.2.2 Pitfalls or Disadvantages 13](#_Toc6857828)

[2.4 Requirement Analysis 14](#_Toc6857829)

[2.5 Prioritization 14](#_Toc6857830)

[2.6 Feasibility study 16](#_Toc6857831)

[2.7 Natural Language Analysis 17](#_Toc6857832)

[2.6.1 Steps for Natural Language Analysis: 17](#_Toc6857833)

[2.8 Use-Case Diagram 17](#_Toc6857834)

[Justification 18](#_Toc6857835)

[2.9 Initial Class Diagram 19](#_Toc6857836)

[2.10 Architecture 20](#_Toc6857837)

[2.10.1 3-Tiers Structure 20](#_Toc6857838)

[Chapter 3: Design 22](#_Toc6857839)

[3.1 Structural Design 22](#_Toc6857840)

[3.1.1 Class Diagram 23](#_Toc6857841)

[Justification 23](#_Toc6857842)

[3.2 Behavioral Diagram 24](#_Toc6857843)

[3.2.1 Activity Diagram 24](#_Toc6857844)

[3.3.2 Sequence Diagram 28](#_Toc6857845)

[3.3 Database Design 32](#_Toc6857846)

[3.1 Entity Relationship Diagram 32](#_Toc6857847)

[3.2 Meta-Data 33](#_Toc6857848)

[3.4 Data flow Diagram (DFD) 35](#_Toc6857849)

[3.5 Prototype (UI Design) 36](#_Toc6857850)

[Chapter 4: Implementation 39](#_Toc6857851)

[4.1 Programming Language 40](#_Toc6857852)

[4.2 Development Environment 40](#_Toc6857853)

[Chapter 5: Testing 41](#_Toc6857854)

[5.1 Black Box testing 41](#_Toc6857855)

[5.2 White Box Testing 41](#_Toc6857856)

[Chapter 6: Other Project Issues 57](#_Toc6857857)

[6.1 Risk Management 57](#_Toc6857858)

[6.2 Configuration Management 61](#_Toc6857859)

[6.2.1Version Control 61](#_Toc6857860)

[Future Work 63](#_Toc6857861)

[User Manual 63](#_Toc6857862)

[Limitation 68](#_Toc6857863)

[Chapter 7: Conclusion of the project 69](#_Toc6857864)

[Chapter 8: Reference and Bibliography 70](#_Toc6857865)

[Chapter 9: Appendix 72](#_Toc6857866)

[Figure 1:Waterfall model 13](#_Toc6865493)

[Figure 2:Use-case Diagram for Online Movies Booking 18](#_Toc6865494)

[Figure 3: Initial Class Diagram 19](#_Toc6865495)

[Figure 4: 3-Tiers Structure 20](#_Toc6865496)

[Figure 5: MVC Design Pattern 21](#_Toc6865497)

[Figure 6: Final Class Diagram 23](#_Toc6865498)

[Figure 7: Activity Diagram for movies Booking 25](#_Toc6865499)

[Figure 8: Activity diagram for search movies 26](#_Toc6865500)

[Figure 9: activity diagram for admin 27](#_Toc6865501)

[Figure 10: Sequence Diagram for User activity 29](#_Toc6865502)

[Figure 11:Sequence Diagram for Cancel Booking Movies 31](#_Toc6865503)

[Figure 12: ER-Diagram for Online movies Booking System 32](#_Toc6865504)

[Figure 13: metadata of user 33](#_Toc6865505)

[Figure 14: metadata for shows 34](#_Toc6865506)

[Figure 15: metadata for screen 34](#_Toc6865507)

[Figure 16: metadata for movies 34](#_Toc6865508)

[Figure 17 : Metadata for Booking 34](#_Toc6865509)

[Figure 18 : Data- flow Diagram 35](#_Toc6865510)

[Figure 19: Protype of Home page 36](#_Toc6865511)

[Figure 20 : Protype of Seat Chart 37](#_Toc6865512)

[Figure 21 : Protype of Login 38](#_Toc6865513)

[Figure 22: Prototype of Register 39](#_Toc6865514)

[Figure 23: (i.) Black Box Testing for login 42](#_Toc6865515)

[Figure 24: (ii.) redirect to home after login 43](#_Toc6865516)

[Figure 25: White Box Testing for Login 43](#_Toc6865517)

[Figure 26: i) Black Box testing for User Register 44](#_Toc6865518)

[Figure 27: ii.) Redirecting to Login after User Register 45](#_Toc6865519)

[Figure 28: White Box Testing for Registeration 45](#_Toc6865520)

[Figure 29: i) Black Box for Add Movie 46](#_Toc6865521)

[Figure 30: ii.) view Movie in home page 46](#_Toc6865522)

[Figure 31: Unit Testing for add Movie 47](#_Toc6865523)

[Figure 32: i) Blackbox testing for Choose Seat 48](#_Toc6865524)

[Figure 33: ii) View on my ticket page of book ticket 48](#_Toc6865525)

[Figure 34 : White box testing for Booking ticket 48](#_Toc6865526)

[Figure 35: i.) testing for add show 49](#_Toc6865527)

[Figure 36: View in showtime after add show 50](#_Toc6865528)

[Figure 37 : White Box Testing for addShow 50](#_Toc6865529)

[Figure 38: Black Box Testing for Cancel Booking 50](#_Toc6865530)

[Figure 39: Black Box Testing for Admin Dashboard 52](#_Toc6865531)

[Figure 40: redirecting to admin dashboard after admin Login 52](#_Toc6865532)

[Figure 41 : White Box Testing for Amin 53](#_Toc6865533)

[Figure 42 i.) Black Boxing testing for delete movie in database 53](#_Toc6865534)

[Figure 43 ii.) Error during deleting movies 54](#_Toc6865535)

[Figure 44: White Box Testing for delete Movies 54](#_Toc6865536)

[Figure 45: Black Box Testing for edit movies 56](#_Toc6865537)

[Figure 46: Black Box Testing for EditMovie 56](#_Toc6865538)

[Figure 47 : Black Testing of Add screen 57](#_Toc6865539)

[Figure 48: White Box Testing for add screen 57](#_Toc6865540)

[Figure 49: GitHub for Movie Booking 61](#_Toc6865541)

[Figure 50: Tree structure 62](#_Toc6865542)

[Figure 51: Tables of movies booking system 72](#_Toc6865543)

[Figure 52: Project Folder 73](#_Toc6865544)

[Figure 53: Database for Movies booking system 74](#_Toc6865545)

[Figure 54: User Model 75](#_Toc6865546)

[Figure 55: Movie Model 76](#_Toc6865547)

[Figure 56: Show Model 76](#_Toc6865548)

[Figure 57: Booking Model 77](#_Toc6865549)

[Figure 58: Screen Model 77](#_Toc6865550)

[Figure 59: Hall Model 78](#_Toc6865551)

[Figure 60: User Controller with profile Update 79](#_Toc6865552)

[Figure 61: (i) Movie Controller inserting movies in database 80](#_Toc6865553)

[Figure 62: (ii) Movie Controller update movie in database 81](#_Toc6865554)

[Figure 63: iii) inserting movie into Database 82](#_Toc6865555)

[Figure 64 : Deleting Movie from database 83](#_Toc6865556)

[Figure 65:)(i) Show Controller for viewing showtime in view code 84](#_Toc6865557)

[Figure 66: (ii). Show Controller inserting showtime in database 85](#_Toc6865558)

[Figure 67: Seat Controller Booking Movie Code 85](#_Toc6865559)

[Figure 68: (i) Ticket Controller view and generate ticket 86](#_Toc6865560)

[Figure 69: ii) Print and cancel ticket code 87](#_Toc6865561)

[Figure 70: Hall Controller inserting hall 87](#_Toc6865562)

[Figure 71: View screen in view form 88](#_Toc6865563)

[Figure 72: inserting screen in database 89](#_Toc6865564)

[Figure 73 :Routes 90](#_Toc6865565)

[Figure 74: UI design code for Seat chart 93](#_Toc6865566)

[Figure 75: Viewing Movie in home page 94](#_Toc6865567)

[Figure 76: Admin Panel 95](#_Toc6865568)

[Figure 77: Adding Movie 95](#_Toc6865569)

[Figure 78: adding showtime 98](#_Toc6865570)

# CHAPTER 1 - INTRODUCTION

## Overview of project

Online Movie Booking system is faster in performance, reliable, secured and have user friendly UI design. System has home page where Customer can visit to home page and view details of Movie which are showing now. System has also Navigation panel where allow to register the system, change profile details and view book movie list, cancel booking and generate ticket etc. Admin can do his/her functionality by login through navbar. Customer will also Login through navigation bar and do some tasks.

## 1.2 Justification of Project

### 1.2.1 Background of Project

Online Movie booking system is web portal where you can book movies for specific date or choose timing for movies show, watch movies trailer and read reviews. You can also view Upcoming movies.

You can also know the rate for Movies and timing just browsing for anywhere and anytime on online. You just need to register in the app and login to browse to all movies which are currently running in theaters and many more facilities. Customers can book 24 hours a day from anywhere for the around globe.

Online movies booking system is very user-friendly application where you can interact with system easily. You do not need to go for theater for booking for movies. You can track everything about movies, timing of movies just clicking it of web app on online. I have used PHP for programming and My SQL for manage database of movies.

## 1.3 Problem Statement

Customer or staff of cinema hall encounter many problems by booking movie ticket on paper-based system. There is chances of duplication of ticket or seats, customer cannot get seat which they want to sit. Nowadays, customer want to do any task work quickly and shortly, so they don’t want to stand on queues for booking movies ticket in font of theaters booking counter. So that admin or manager of movies hall get difficult to manage these problems. The main problem of current running system is time consuming and complex system for both customer and staff or manager of cinema hall.

The main motive of this project is to reduce complexity of the running system in cinema hall. The proposed project will overcome all the problem related to booking process of movie ticket. This new proposed system shows all show time scheduled for movies so that any customer can see details of movies with respective show time from using web browser from anywhere anytime.

## 1.4 Features

* User can Login and Register to the system.
* User can Book movies.
* Admin can do crud operation.
* User can watch details of movies with trailer and showtime.
* User can Cancel Booking.
* User can change their profile details.
* Lists of Upcoming Movie with details.

## 1.5 Aims and Objectives

### 1.5.1 Aims

* Main aims of the proposed system are to book the movie ticket for cinemas.
* Replace to Manual Booking i.e. paper based booking.
* Promote Movie through internet.
* User can know showtime with details of movies through web application by surfing internet.

### 1.5.2 Objectives

* Booking Movies ticket from online.
* User can know details of movies with scheduled showtime by surfing internet with visiting cinema hall.
* User friendly.
* Helpful for management for decision making.
* Provide 24 hors service to customer through online.
* For make secured and reliable booking system for cinema.

## 1.6 Scope

Online movie Booking system is a web portal which helps for organization for storing and retrieving information about movies, upcoming movies, seats reservation, ticket sales etc. UI is user friendly so that user can easily use system.

# CHAPTER 2 – Analysis

Analysis is the process of identification and documentation of requirement of the proposed system. In the analysis phrase, first step is to feasibility study after that we model system in use case diagram and class diagram which is also called system modelling. (SDC, 2011)

Analysis is done to examine the requirement which is given by the user so that it is easy to find out features and importance of the system. Analysis is important parts before starting developing any software. There are many methods for analysis project like a problem identification, gathering a requirement from a user, and also develop prototype of the system. There are many techniques for analysis for project like by interview with user, questionnaire with user, observation in the workplace, etc.

# 2.1Object Oriented Analysis

An object-oriented analysis is the structure method for analysis and design by using the object-oriented concept like class and method and develop a graphical system model for SDLC process. (OmarElGabry's, 2019)

## 2.2 Development Methodology

Waterfall model is the first introduced in the software development. It is simple and easy to understand. Each phrase needs to finished before next phrase start. In this methodology, whole process is divided into separate tasks. Outputs of one phrase is input for other phrase int this methoolody.it is also called sequential process which is look like a downward waterfall, (SHARMA, 2018)

Different Phrase of Waterfalls Methodology are:

1. Requirement Analysis:

First phrase of waterfall model where analyst understand the user requirement so that they clear on what need to be design, its function and its purpose. Inputs and output are studied.

1. System Design:

Gather requirement in phrase first studied here and according to that system is design. It also helps for overall architecture of the system.

1. Implementation:

From inputs of system design, system is developed in small program called as units. After all units developed system is integrated. Each developed small unit is test in next phrase.

1. Testing

Small program developed in implementation phrase is tested in this phrase. In this phrase each unit test it functionality which is called as unit testing. Testing is so important because after deployment user should not face any error when running a system. Both functional and non-functional testing is done here.

1. Deployment:

After Testing both functional and non-functional testing in testing phrase, now system is release in user environments.

1. Maintenance:

This phrase is done after installation system to the customer environment. In this phrase modification can be make from request by customers or when some defect is available in the system. User get can full support and regular maintenance for developed system.

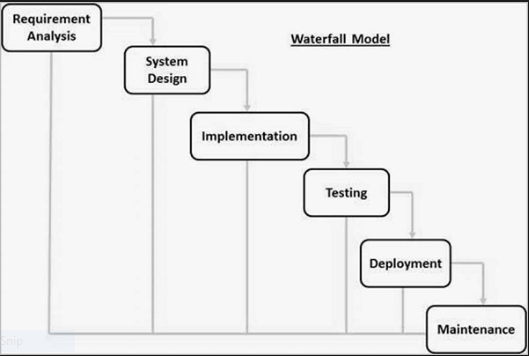


Figure 1:Waterfall model

## 2.3 Merits and Pitfalls of Project

### 2.2.1 Merits

* Always available for Reservation Seats.
* Reduce no-shows.
* Less in Phone Time.
* Get valuable information of business

### 2.2.2 Pitfalls or Disadvantages

* Online Payment system not available.
* Require internet Access.
* Virus and Trojans can steal information.

## 2.4 Requirement Analysis

Requirement analysis is the process of identification of user requirement for system developed. It is one important parts of system development. For success of any product, requirement analysis is important for analyze any project requirement. (ReQst, 2019)

## 2.5 Prioritization

Prioritization is the process of determine candidate requirement for the system which should be included for certain release. It can ensure easily that which requirement should be first and which should be last. (requirement.com, 2004).

I have used MOSCOW prioritization for requirement analysis for this project.

MOSCOW stand for

M=**Must Have**

S=**Should Have**

C=**Could Have**

W=**Would have**

There are two types of requirement

1. **Functional Requirement**

Functional Requirement is describing of how system can do.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Functional requirement | Requirement | Justification | Moscow | Dependency |
| F(R1) | User Registration to system | User must registration to system. | Must have |  |
| F(R2) | User Login to system | User must Login to System for booking movies. | Must have | F(R1) |
| F(R3) | Add Movies | Admin can add movies | Must have |  |
| F(R4) | View Movies | User can view details of movies | Must have | F(R3) |
| F(R5) | View show time | User can view details of show time of movie | Must have | F(R3) |
| F(R6) | View show date | User can view details show date of movie | Must have | F(R5) |
| F(R7) | Update User profile | User can update their profile | must have | F(R1) |
| F(R8) | Update Movies | Admin can update movies | Must have | F(R3) |
| F(R9) | Ticket generate after booking Movies | Ticket must generate after user book particular movie | Must have | F(R3), F(R10), |
| F(R10) | Seat selection | User can select seat | Must have |  |
| F(R11) | Show name | System must view show name | could have | F(R6) |
| F(R12) | Show type | System must view show type and show time | Could have | F(R6) |
| F(R13) | Delete Movie | Admin can delete movie | Must have | F(R3) |
| F(R14) | Admin Login | Admin can login to system | Must have | F(R1) |
| F(R15) | Update Admin Profile | Admin can update his/her profile | Should have | F(R1) |
| F(R16) | Payment System | Online payment system for paying fee for movies | Could have |  |
| F(R17) | Rating and Reviewing Movie | User can rate and Review Movie | Could have |  |

1. **Non-Functional Requirement**

Non-function Requirement is describing of how system behave.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Non-Functional requirement | Requirement | Justification | Moscow | Dependency |
| NF(R1) | Security of system | Make sure that System must secure and protected from unauthorized access. | should have | NF(R6) |
| NF(R2) | Portability | System should be run in every platform. | Should have |  |
| NF(R3) | Usability | System should navigation easily and easy to use. | Should have |  |
| NF(R4) | Scalability | System should store more and more information easily | Could have |  |
| NF(R5) | Maintainability | Maintenance should be done in regular basic | Must have |  |
| NF(R6) | Privacy | System should keep user information private | should have | NF(R1) |
| NF(R7) | Reliable | Precise and accurate information should given to system | Should have |  |

## 2.6 Feasibility study

Feasibility study is used to describe of idea which is ensuing a project which is legally and technically feasible and economically just able. It can also identify project is potentially success. For this project feasibility study is undertaken which are as below:

Advantages of feasibility study:

* It can provide valuable information for the project.
* By evaluating multiple factor, it has enhanced success rates.

Disadvantages of feasibility study:

* Costly
* Wrong information

## 2.7 Natural Language Analysis

Natural Language Analysis is the process of identifying verbs, adjectives and noun in the description of project Scenario. NLA helps to get candidate class with their relationship and attributes. Nouns is related to potential class; adjectives are a potential attributes and verbs is potential functionality of the system.

### 2.6.1 Steps for Natural Language Analysis:

* Find out all nouns, verbs and adjectives from scenario.
* Remove all having synchronism.
* Remove all ambugies words.
* Remove technical details like database, network etc.
* Remove project out of scope.

Following are the Candidate Class which are analysis from NLA:

* Customer
* Movie
* Screen
* Show
* Ticket

## 2.8 Use-Case Diagram

A Use-case is a software and system engineering term that describe how user can interact with system to perform a particular operation. A use case acts as a software modelling technique that implies function to be implemented. (Techopedia, 2018)

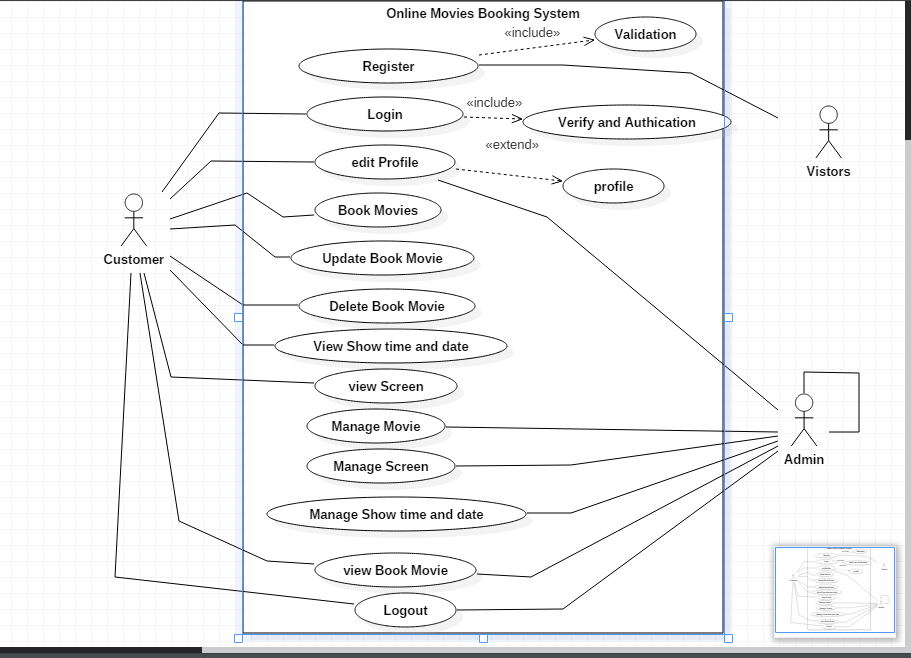


Figure 2:Use-case Diagram for Online Movies Booking

### Justification

In above diagram, there are three types of actors i.e. first one is visitor’s who is unregister customer for the system, second one is registered customer and last one is admin who manages whole system.

First actor i.e. visitor who can register to the system via registration script and he/she can become the registered customer. Now, he/she can do all things that registered customer can do to the system.

Second actor i.e. registered customer who is mainly focus by system. He/she can book movies, edit their profile, view different show time and date of movies running in the theaters and also update and delete book movie.

Lat actor i.e. admin who is system administration. He/she can manage movies, different shows times and dates, manages screen view book movies manages etc.

# 2.9 Initial Class Diagram

A class diagram is a description of relationship between different classes. It is a static diagram which represent static view of system. It describes the attributes and operations of class. It also shows a collection of classes, interface, association. (Guru, 2019)

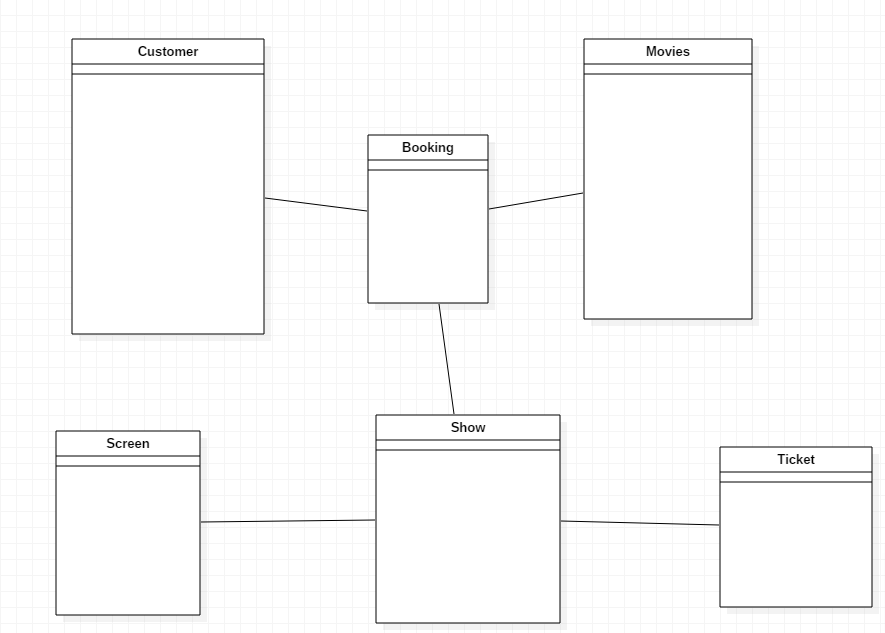


Figure 3: Initial Class Diagram

## 2.10 Architecture

### 2.10.1 3-Tiers Structure

System architecture is conceptual model that defines the structure, behavior and view of system. It describes the representation of whole system. For this project, I have use 3-tier structure. (Mitre, 2016)

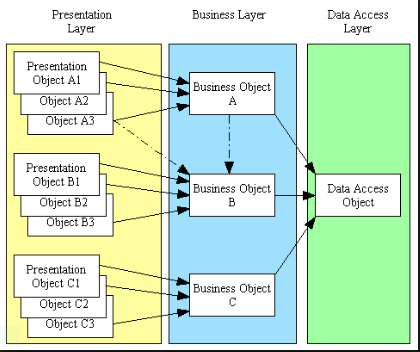


Figure 4: 3-Tiers Structure

#### 2.10.2 Design Pattern

Design pattern is a reusable solution for commonly occurring problem is software development. It is not a finished product. It is only template for how to solve any problem during development of software. (Sourcemaking, 2019)

I have used MVC design pattern for my project movies booking. MVC stands for Model view Controller. It is a structural design pattern. It broken down application into three parts i.e. model, view and controller. It is frequently used design pattern in software development industry to develop project scalable and extensible.

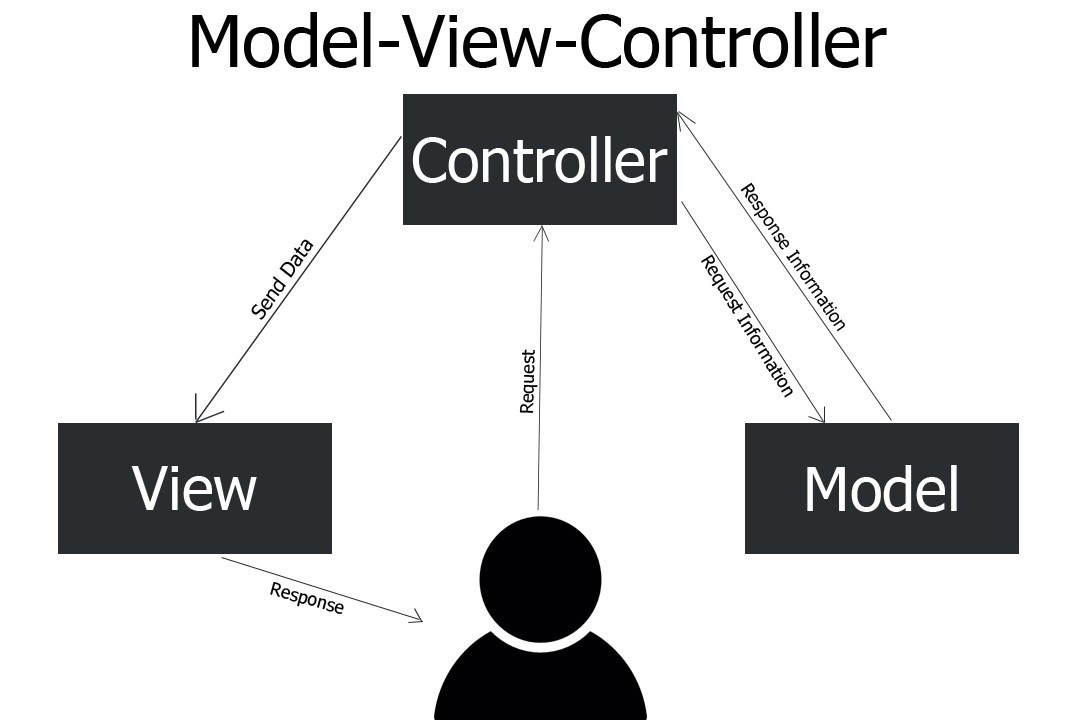
****

Figure 5: MVC Design Pattern

**Model:**

Model represents all business logic of the system

**View:**

View represents all UI parts of the system.

**Controller:**

Controller acts as mediator for view and model for communication.

# Chapter 3: Design

A design phrase is detailed document providing information about a going to developed product of process. Its main aims to ensure that development product should meet users’ requirements. There are three types of design phrase i.e. structural design, behavioral design and database design.

## 3.1 Structural Design

Structural diagram illustrates static structure of the system. It shows how the different component or module is connected and interact with each other. It cannot change unit the system development completed. It also shows the relationship of class in the system. Class diagram, ER-diagram, object diagram, Package diagram, model diagram, collaboration diagram are examples of structural diagram. For this project, I have used Class Diagram, ER-diagram which are illustrated briefly with diagram below.

## 3.1.1 Class Diagram

Class Diagram is UML, a type of static view of the system that describes the whole structure of the system by show classes, attributes, operation or method and relationship between classes. (Visual-Paradigm, 2019)

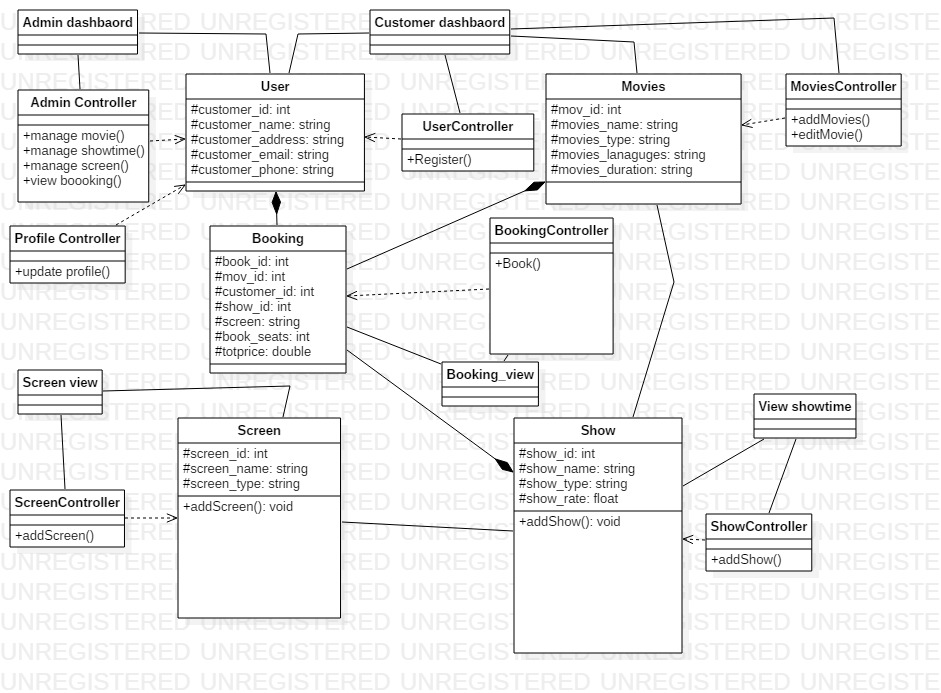


Figure 6: Final Class Diagram

### Justification

The above class diagram illustrated that each class has each controller and controller are direct associated with Database. Ticket class is dependent with show class and ticket controller. Booking is dependent on Movies and Customer class.

## 3.2 Behavioral Diagram

Behavioral diagram depicts the dynamic behavior of object with a system which can describes the changes of system over run time. Changing of requirement might change to the structure of the system. System diagram is design according to the requirement of the user. Activity diagram, sequence diagram, use-case diagram, state machine diagram, communication diagram, timing diagram, etc. are examples of behavioral diagram. (UML, 2019). For this project, I have used three behavioral diagrams i.e. Use-Case Diagram, Activity diagram, and Sequence Diagram which are illustrate as below:

### 3.2.1 Activity Diagram

Activity diagram is UML diagram which describes the dynamic aspect of the system. It is advanced version of flow chart which show logic of flow of one activity to another activity.

The notation used for activity diagram are as follow:

* Start point
* Activity or Action state
* Action flow
* Decision
* Swimlane
* Ford
* Final point

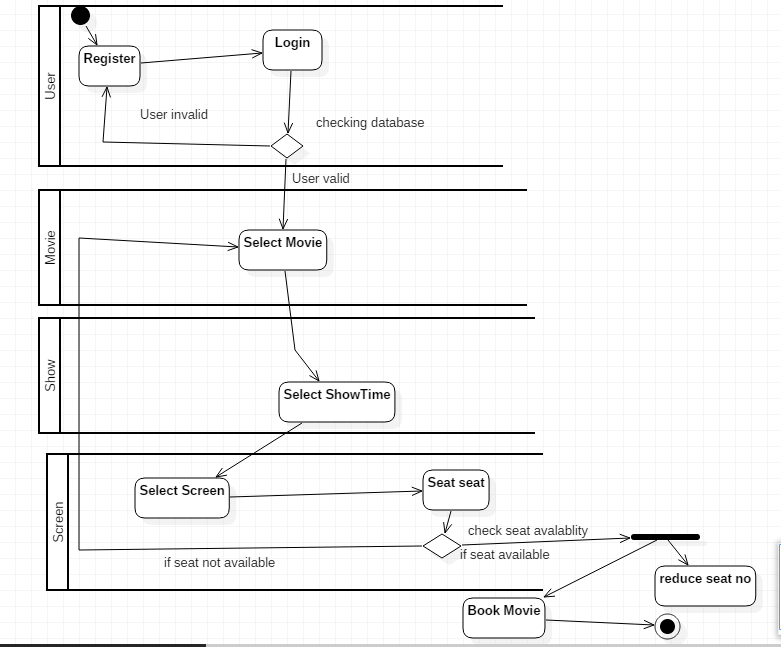


Figure 7: Activity Diagram for movies Booking

**Justification**

The above activity diagram is for booking movies for user. At first user can register and after that login in user class. If user is valid system will redirect to movies class display with movies name where user can select movies or if user is invalid system will redirect to register form. After selecting movies, now user will select show time in show class after that again select screen and seat in screen class. If seat is unavailable then system will redirect to movies list again or if seat is available two operation is performed simultaneously one is book movies and another one is reducing no of seats from particular screen.

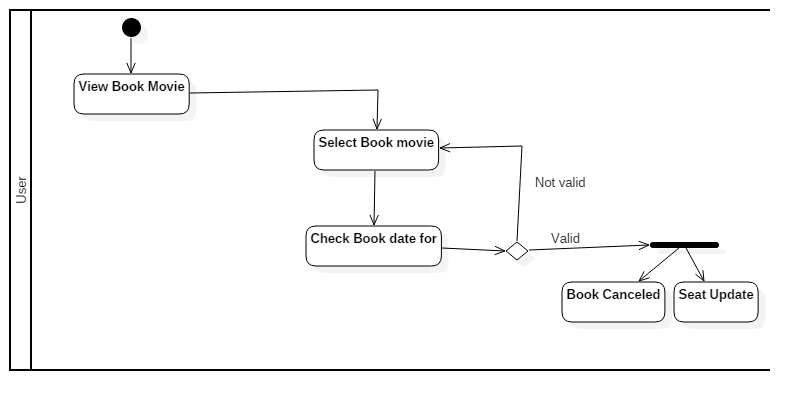


Figure 8: Activity diagram for search movies

**Justification**

Above activity diagram is for cancel book movie by the users. At first user can view movies in his/her booking lists. After that customer select book movie and if movie date is already gone then, customer will choose upcoming date where he/she has book movies. If date is not gone than user will cancel book from the database.

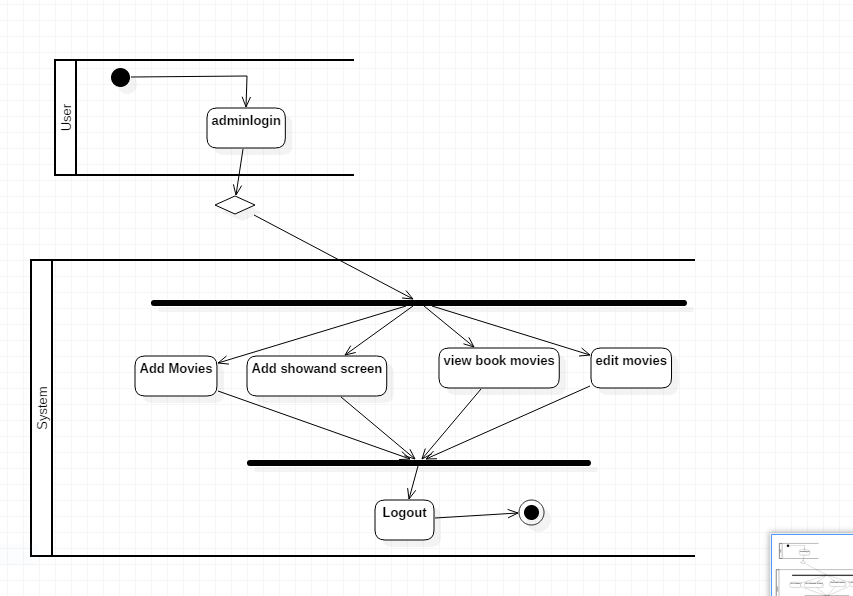


Figure 9: activity diagram for admin

**Justification**

Above activity diagram illustrated the activity done by the admin. Admin can add edit delete movies, showtime and screen. In the diagram at first admin login to system after that admin will add movies add showtime and screen view booked movie by user and edit movies.

### 3.3.2 Sequence Diagram

Sequence diagram can illustrate how object can interacts with sending message system in sequential order. It describes the runtime system. It also describes how object can pass within system. It is used by developer team to understand requirement a new and existing system. Sequence diagram for my project are as follows:

The notation used for sequence diagram are as follow:

* Life Line
* Activations bar
* Message arrows
* Comment

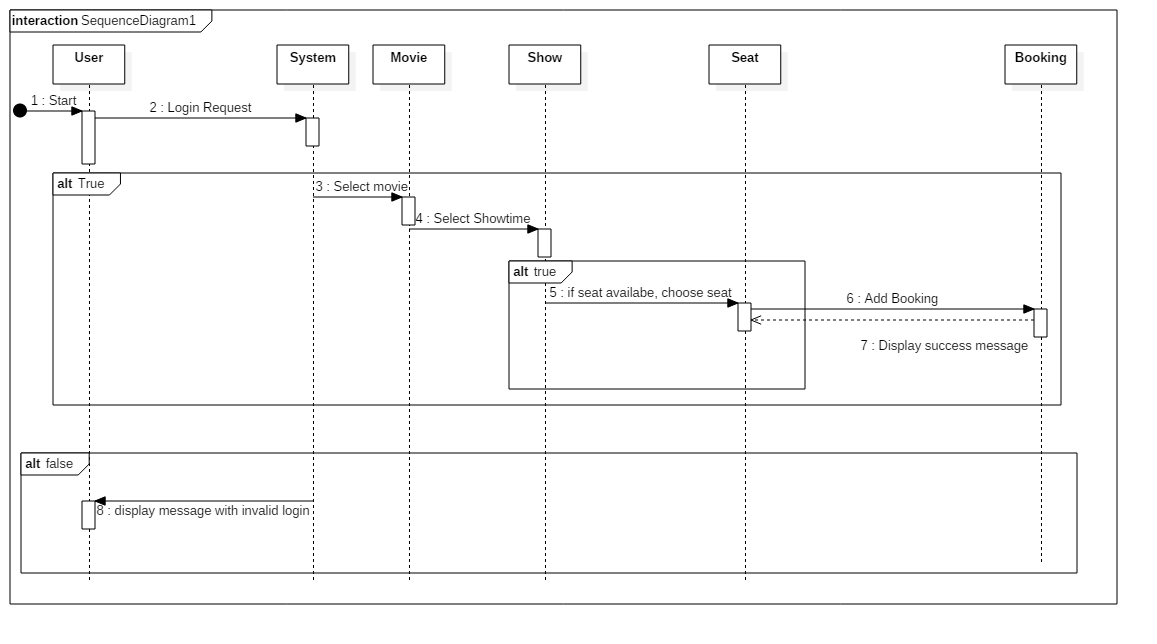


Figure 10: Sequence Diagram for User activity

**Justification**

As above diagram, it is sequence diagram for User activity in the system. It shows in diagram that User sends a Login request message with username and password to the system and system will send message to the database, and check if username and password valid then in combined fragment if else user send object to movie class after that to showtime object and again it check to database whether seat is available or not if seat available user sends object to seat and movie is book if not available user get redirect to show time again. while database send response to the system of and again system send success message to User interface.

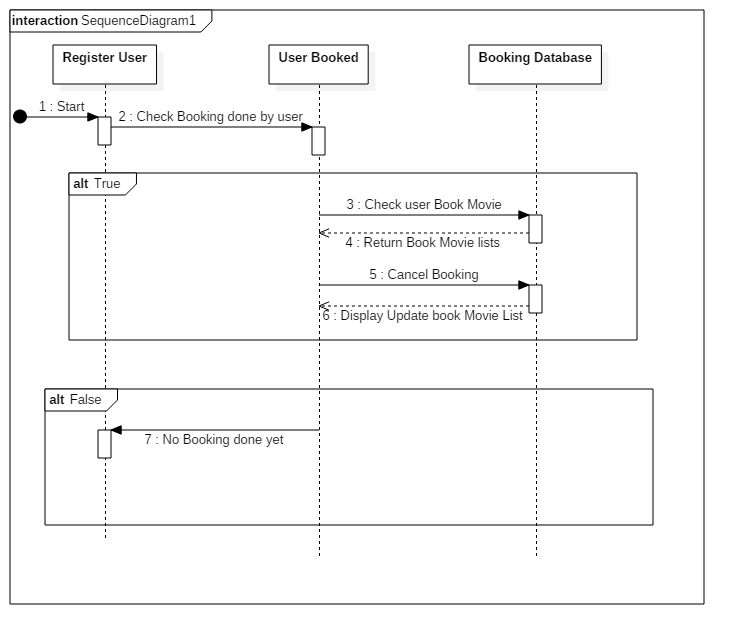


Figure 11:Sequence Diagram for Cancel Booking Movies

Above figure, it is sequence diagram for user cancel book movie from the database. At first, user check whether they have made booking or not. If booking is made, then select book movies and cancel it and send to booking database and booking database send response with new updated booking lists in UI interface. If booking is not made already, then it response message with movie not book yet.

## 3.3 Database Design

Database design is the process of designing, development and implementation of data for proposed system. The main aims of database design are to produce logical and physical model for any system. (Guru, 2019)

### 3.1 Entity Relationship Diagram

An Entity Relationship Diagram also called as ERD is a type of flowchart that illustrates how entities relate with each other within system.

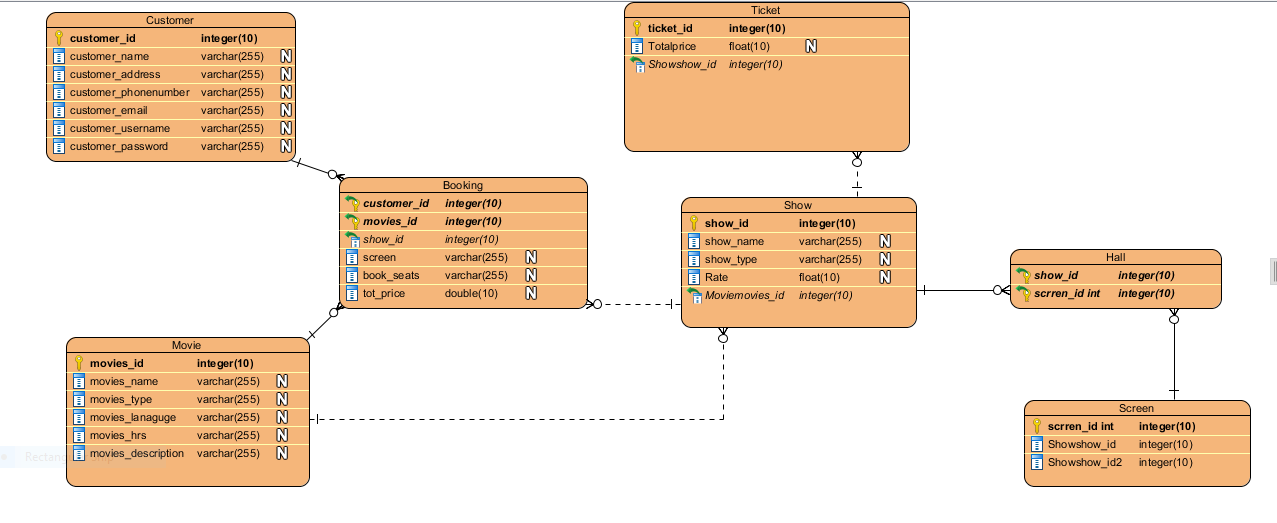


Figure 12: ER-Diagram for Online movies Booking System

In above diagram, it is ER-diagram for online movie booking system. It shows the relationship between entities of database. Above ER-diagram is developed for real working database for movies booking. In above ER-Diagram, customer and movies has many to many relationships has linking table called as Booking table between customer and movies because one Customer can book multiple movies whereas one movie can book by multiple customers. Shows and booking has one to many relationships where showed is foreign key in booking table because one show has multiple booking but one booking has for only one show only. Movies and Show table have one to many relationships because one movie has multiple show but one show has only one movie at time. Show and Screen table has many to many relationships which both have linking table called as Hall table because one show has multiple screens also one screen has multiple show. And, show and ticket has one to many relationships.

### 3.2 Meta-Data

Meta-data is data of data, which describe the context of data stores in columns helps for organize, find and understand data in Relational database management system.

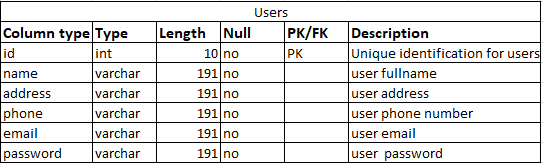


Figure 13: metadata of user

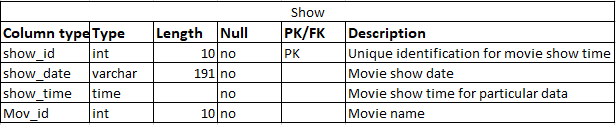


Figure 14: metadata for shows

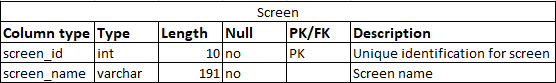


Figure 15: metadata for screen

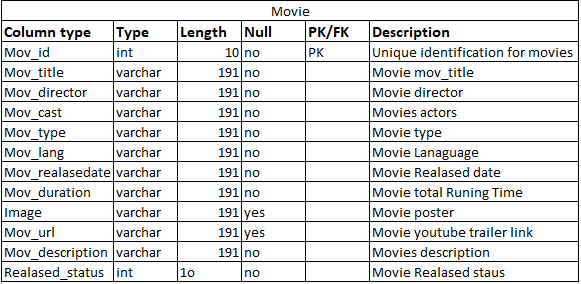


Figure 16: metadata for movies



Figure : Metadata for Booking

## 3.4 Data flow Diagram (DFD)

A DFD is data flow diagram describe graphically representing of the flow of the data within a system. It describe the processes involved in the system and transfer data to storage . (Paradigm, 2019)

The symbols used in dfd are:

* Process
* Data flow
* Data store
* External Enitity



Figure : Data- flow Diagram

## 3.5 Prototype (UI Design)

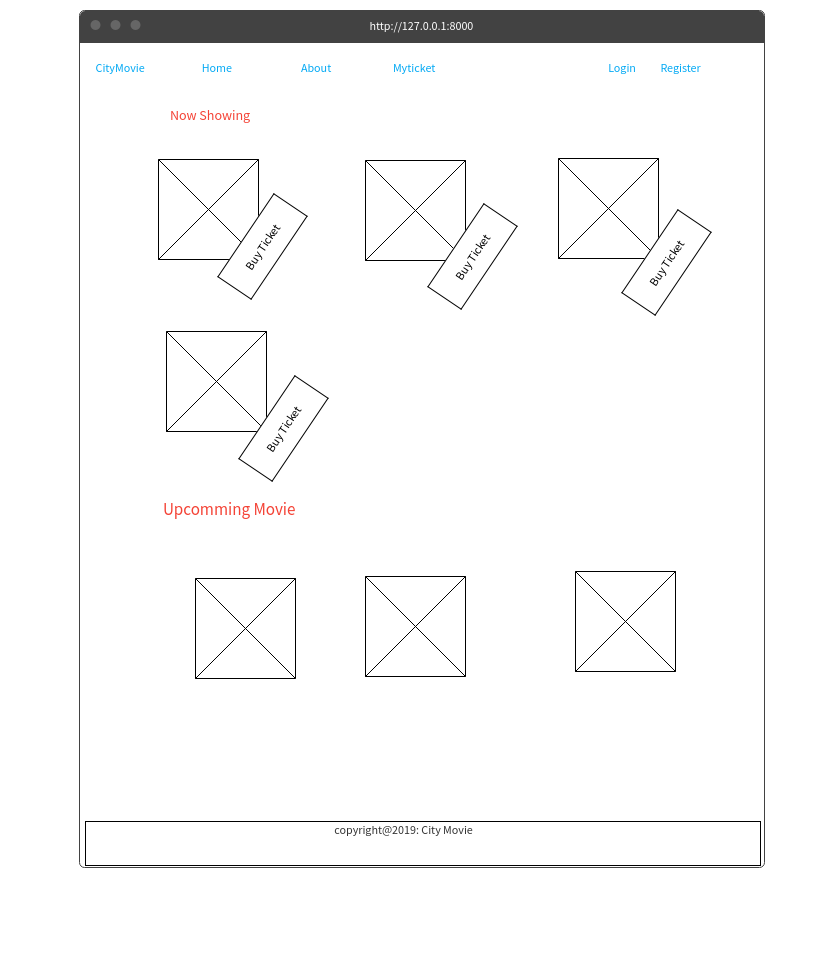


Figure : Protype of Home page

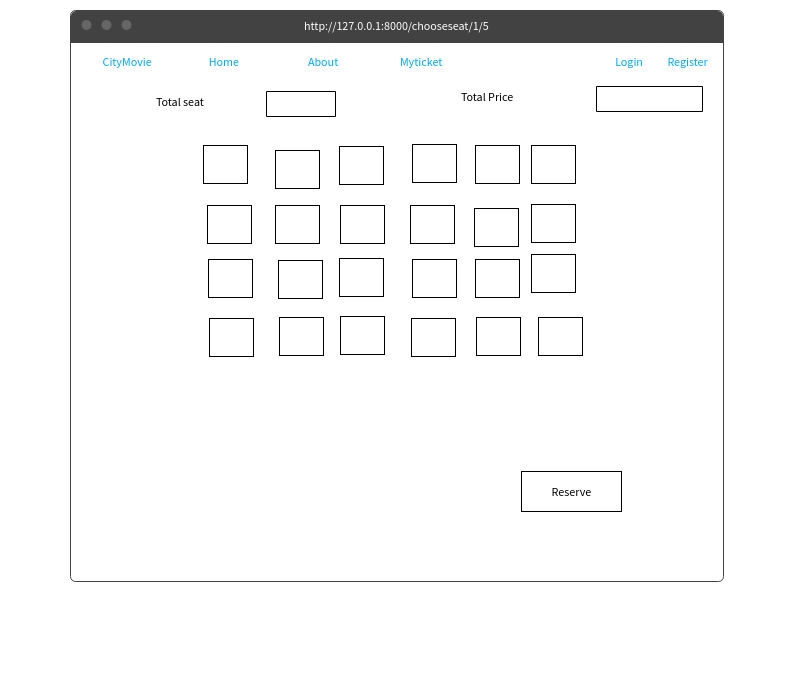


Figure : Protype of Seat Chart

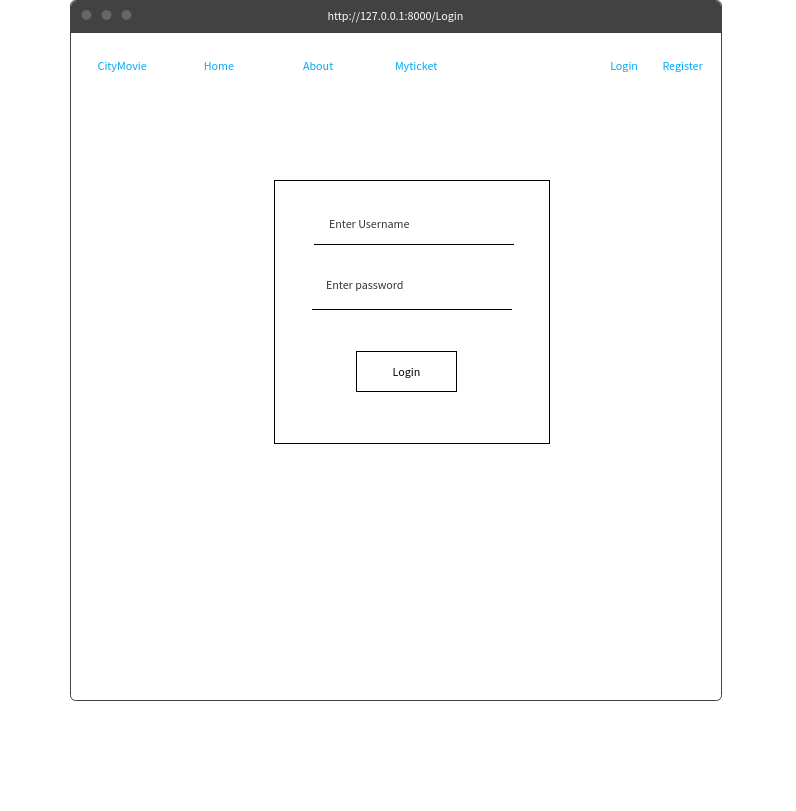


Figure : Protype of Login

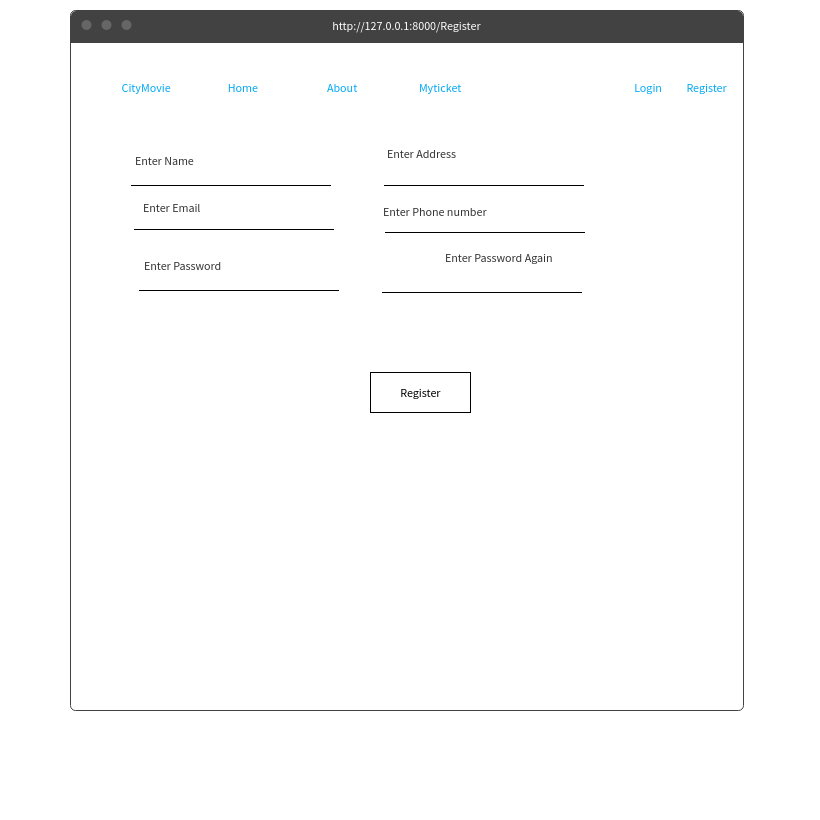


Figure : Prototype of Register

# Chapter 4: Implementation

Implementation is the process of converting logical implementation into the real working system. It also ensures that how a new system is built and how systems functions work. It is 3rd phrase of the software development lifecycle. **All Code and UI in appendix.**

## 4.1 Programming Language

Programming Language is the language which can give instructions for the computer to perform a specific task with specific output. It is used for creating a program or a system by implementation algorithm. The high-level language such as Java, C#, Python, C++, etc. are more easily to use but require compiler or interpreter to convert into machine code because computer understand only machine language like 0 and 1.

For My project, I have used php with Laravel Framework for converting logical into real implementation. Php stands for **Hypertext Preprocessor** which is a server-side scripting language used for development of the web application. Laravel is most used framework php for developing web systems. Laravel use MVC design Pattern. Php as follows object-oriented program. For store data in database, I have used MYSQL database. I have also php with JavaScript and ajax for user friendly.

## 4.2 Development Environment

Development Environment is the processes and programming tools which is used for development or creating any software products. It is also called as physical environment. An integrated development environment is the set of processes and tools which are integrated to provide developers for viewing Interface and development processes like code, testing and other packages. The examples of integrated development environment are Microsoft Visual Studio. There are many development environments like Microsoft Visual studio, Visual Code, Php Storm, sublime, etc. (TechTarget, 2019)

For my project, I have used Visual Code for development environment. It is a free of cost and flexible so that it is easy to use. For my project, I have installed Laravel Framework with MVC design pattern in visual code. MVC is architectural design pattern which can reuse code and also maintains the code standards.

# Chapter 5: Testing

Software Testing is the process of check whether actual result matches expected result to ensure that software is defect fee. It also helps to finds out error, missing requirement. (Guru, 2019)

## 5.1 Black Box testing

Black Box Testing is software testing that focuses on the system functionality rather than internal structure. Black box testing ensures that system output whether it is giving correct output or not. It is also called functionality testing.

## 5.2 White Box Testing

White box testing is software testing technique, which checks the code and internal structural of code. In this type of testing, tester focus on the flows of input and output through the application so that tester can easily improve the either function is working or not and also check is performance too. It is also called as unit testing, structural testing, etc.

|  |  |
| --- | --- |
| **Test No:** | 1 |
| **Purpose of test:** | get username and password and redirect to home page |
| **Test Data:** | [utttam@gmail.com](mailto:utttam@gmail.com), Uttam123 |
| **Test name** | test Login |
| **Expected Result:** | Login and redirect to home page |
| **Actual Result:** | Login Successfully and redirect to home page |
| **Test Status**: | Pass |

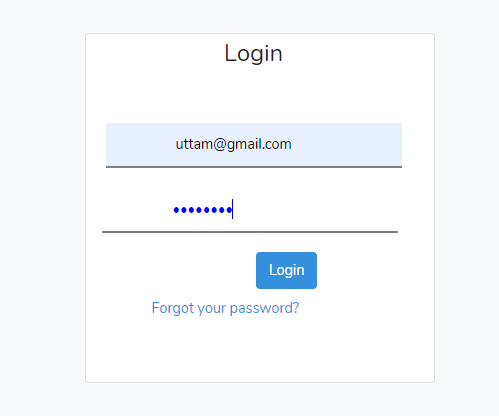


Figure 23: (i.) Black Box Testing for login

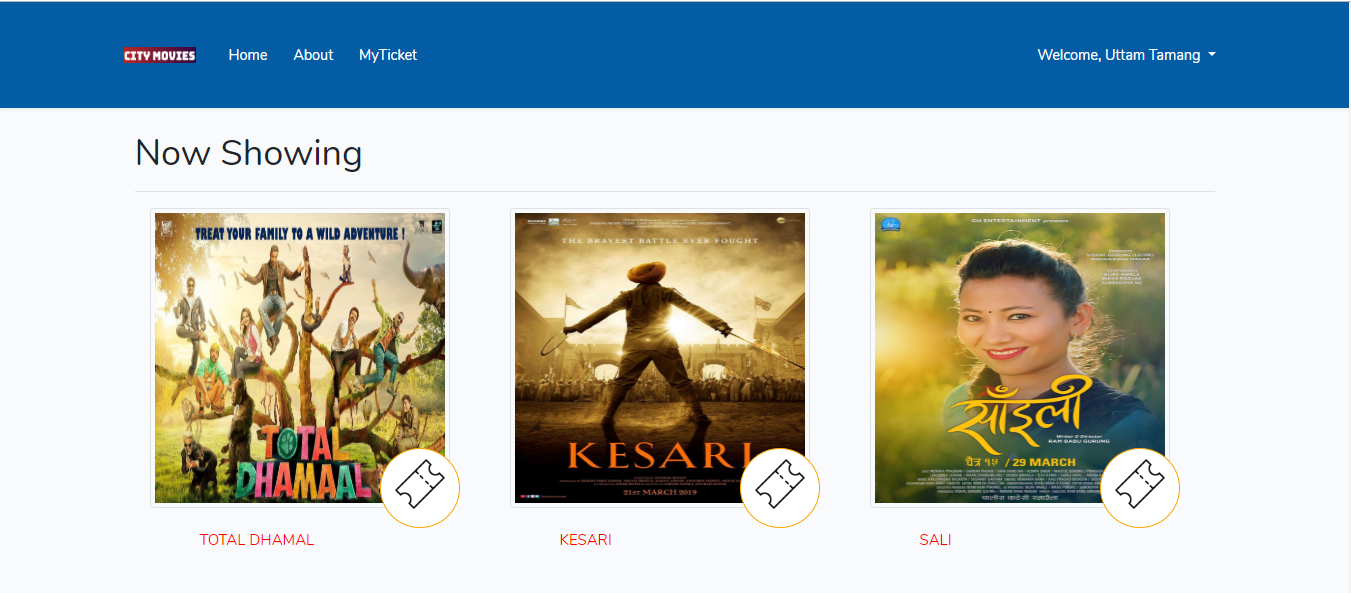


Figure 24: (ii.) redirect to home after login

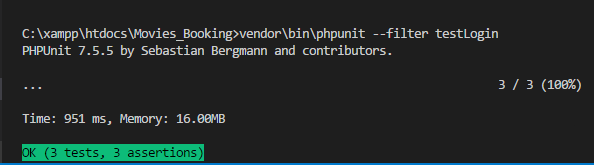


Figure 25: White Box Testing for Login

|  |  |
| --- | --- |
| **Test No:** | 2 |
| **Purpose of test:** | register to system and redirect to home page |
| **Test Data:** | Subash Thapa, Pokhara, [subashthapa901@gmail.com](mailto:subashthapa901@gmail.com),  982346789008, Subash Thapa |
|  |
| **Test name:** | test User |
| **Expected Result:** | register and redirect to Login page |
| **Actual Result:** | register Successfully and redirect to Login Page |
| **Test Status**: | Pass |

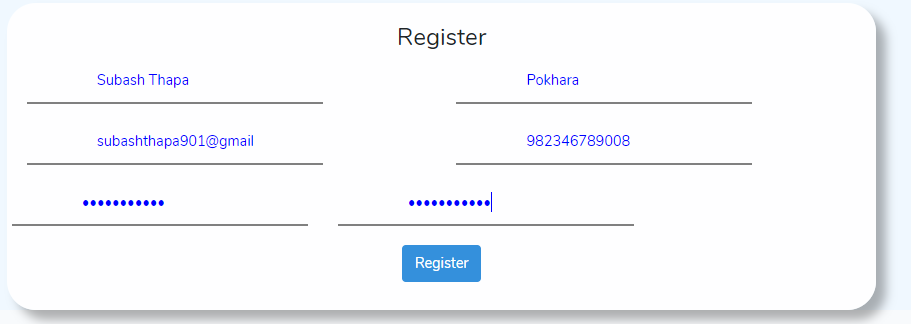


Figure 26: i) Black Box testing for User Register

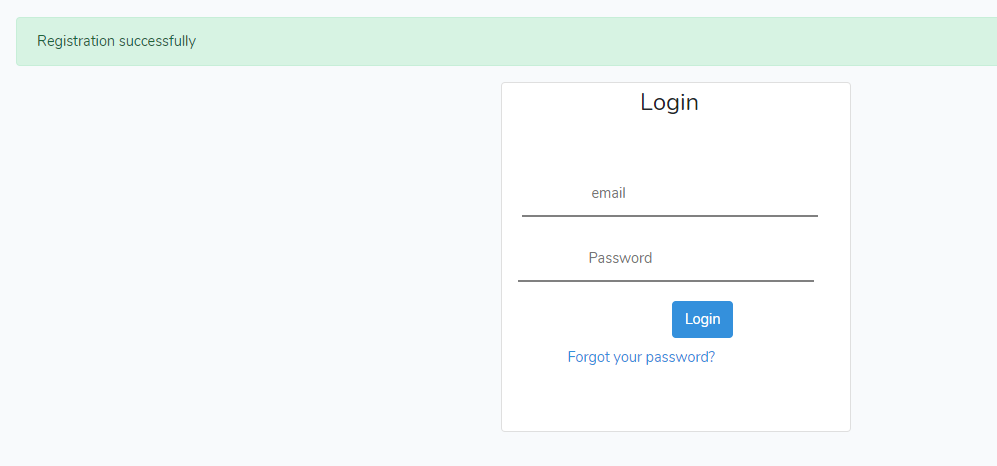


Figure 27: ii.) Redirecting to Login after User Register

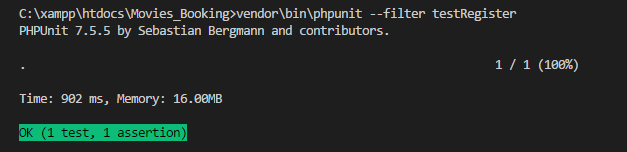


Figure 28: White Box Testing for Registeration

|  |  |
| --- | --- |
| **Test No:** | 3 |
| **Purpose of test:** | Add Movie to the system |
| **Test Data:** | JUNGLEE, chuck Russel, Vidyut jammal, Atul kulani,  Asha Bhata, Action, hindi, April 29 |
| **Test name:** | test Movie |
| **Expected Result:** | add Movie and view on home page |
| **Actual Result:** | Movie added Successfully and View to home page |
| **Test Status**: | Pass |

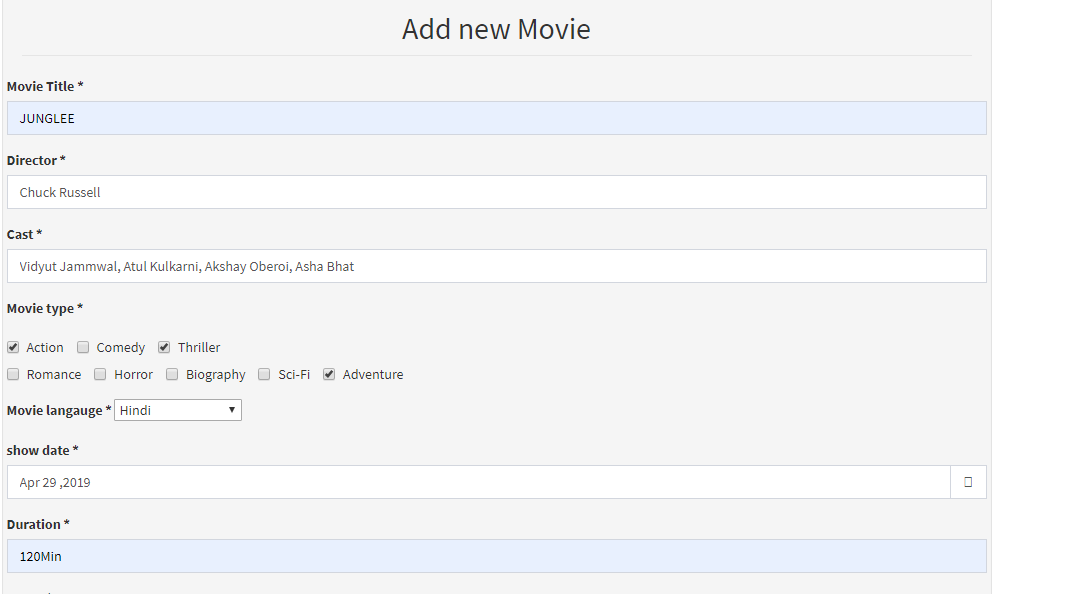


Figure 29: i) Black Box for Add Movie



Figure 30: ii.) view Movie in home page

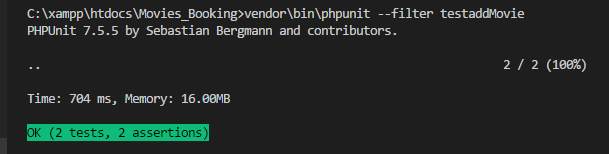


Figure 31: Unit Testing for add Movie

|  |  |
| --- | --- |
| **Test No:** | 4 |
| **Purpose of test:** | Book Movie ticket |
| **Test Data:** | Audi1, (1,2,3), Rs. 900, April 12, 2019, 8:00 AM |
| **Test name:** | test book Movie ticket |
| **Expected Result:** | book Movie Ticket and view on my ticket page |
| **Actual Result:** | Book ticket Successfully and View to my ticket page |
| **Test Status**: | Pass |

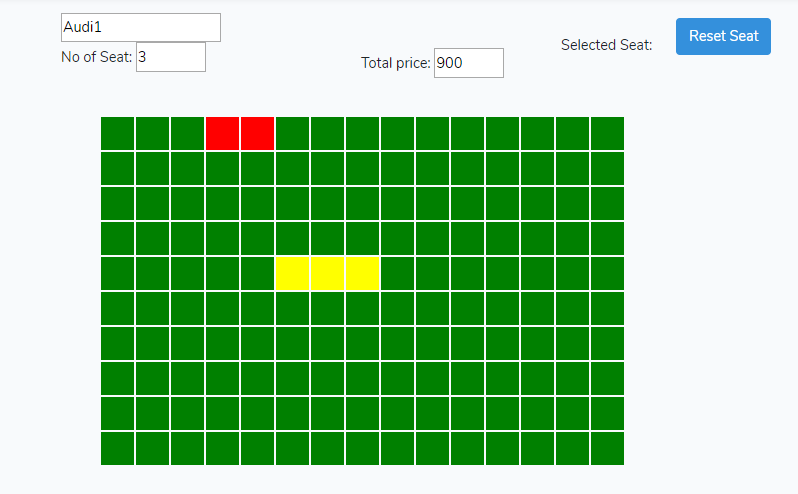


Figure 32: i) Blackbox testing for Choose Seat

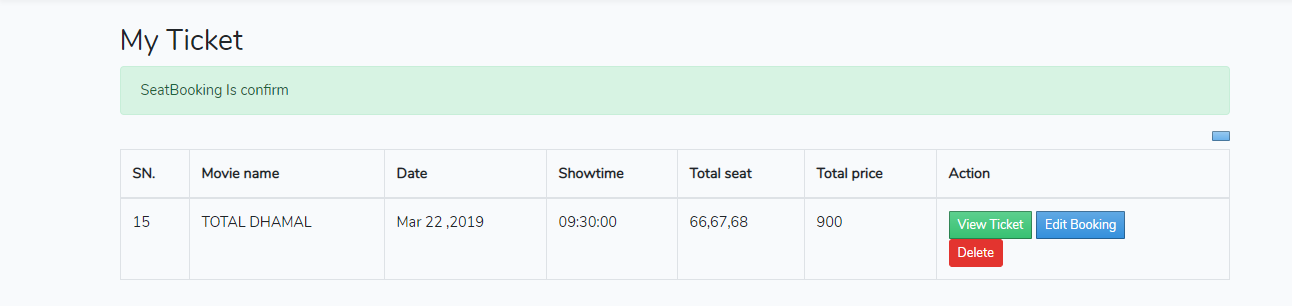


Figure 33: ii) View on my ticket page of book ticket

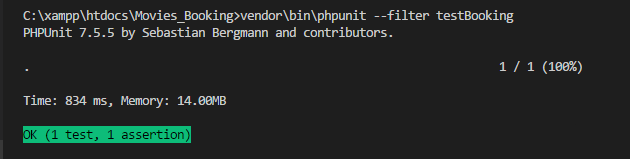


Figure 34 : White box testing for Booking ticket

|  |  |
| --- | --- |
| **Test No:** | 5 |
| **Purpose of test:** | Add showtime |
| **Test Data:** | Apr 6, 09:30 Am, Junglee, 185 |
| **Test name:** | test showtime |
| **Expected Result:** | add showtime and view on showtime page |
| **Actual Result:** | show added Successfully and View to my showtime page |
| **Test Status**: | Pass |

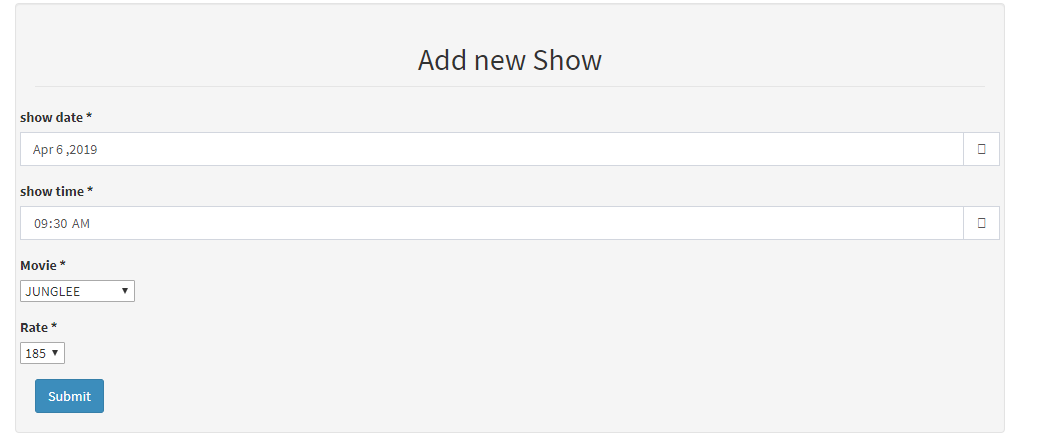


Figure 35: i.) testing for add show

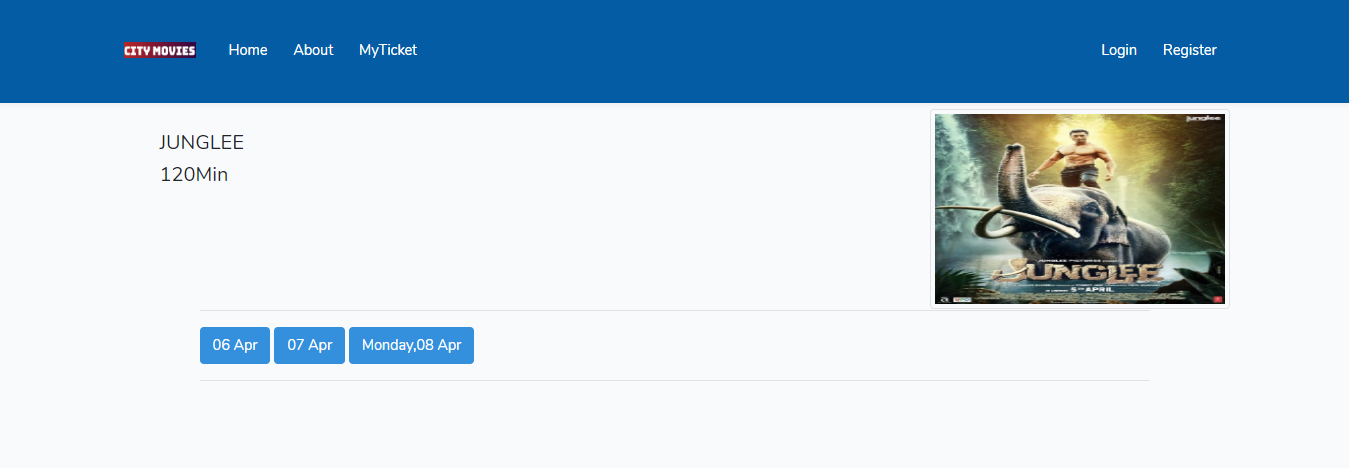


Figure 36: View in showtime after add show

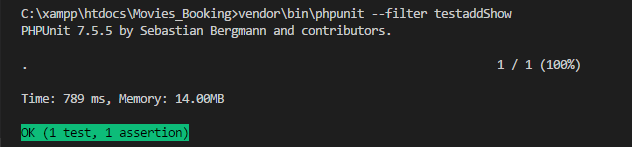


Figure 37 : White Box Testing for addShow

|  |  |
| --- | --- |
| **Test No:** | 6 |
| **Purpose of test:** | Cancel Booking |
| **Test name:** | test book |
| **Expected Result:** | cancel booking |
| **Actual Result:** | cancel booking Successfully |
| **Test Status**: | Pass |

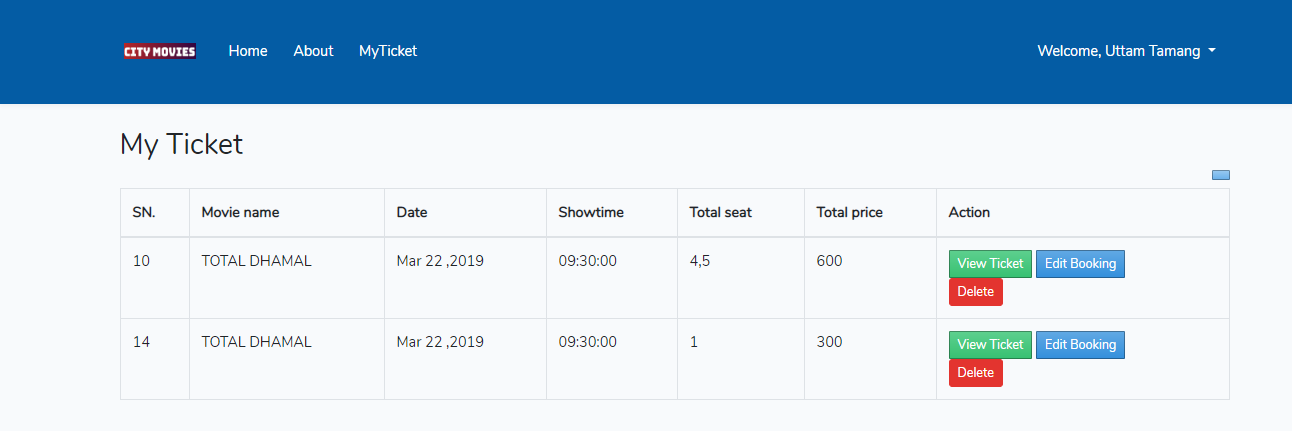


Figure 38: Black Box Testing for Cancel Booking

|  |  |
| --- | --- |
| **Test No:** | 7 |
| **Purpose of test:** | Access by Admin |
| Test Data: | [aashishpokhrel146@gmail.com](mailto:aashishpokhrel146@gmail.com), Aashish123 |
| **Test name:** | test Admin Login |
| **Expected Result:** | retrieve username and password and redirect admin |
|  | Dashboard |
| **Actual Result:** | Login Successfully and redirect to Admin dashboard |
| **Test Status**: | Pass |

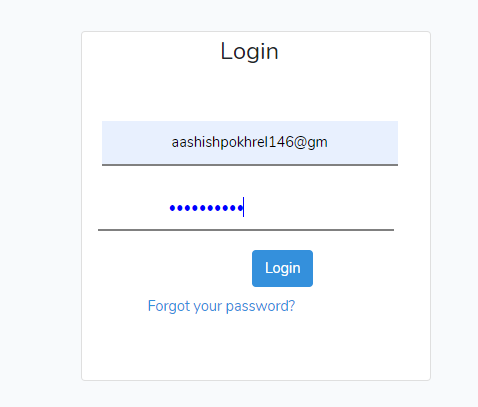


Figure 39: Black Box Testing for Admin Dashboard

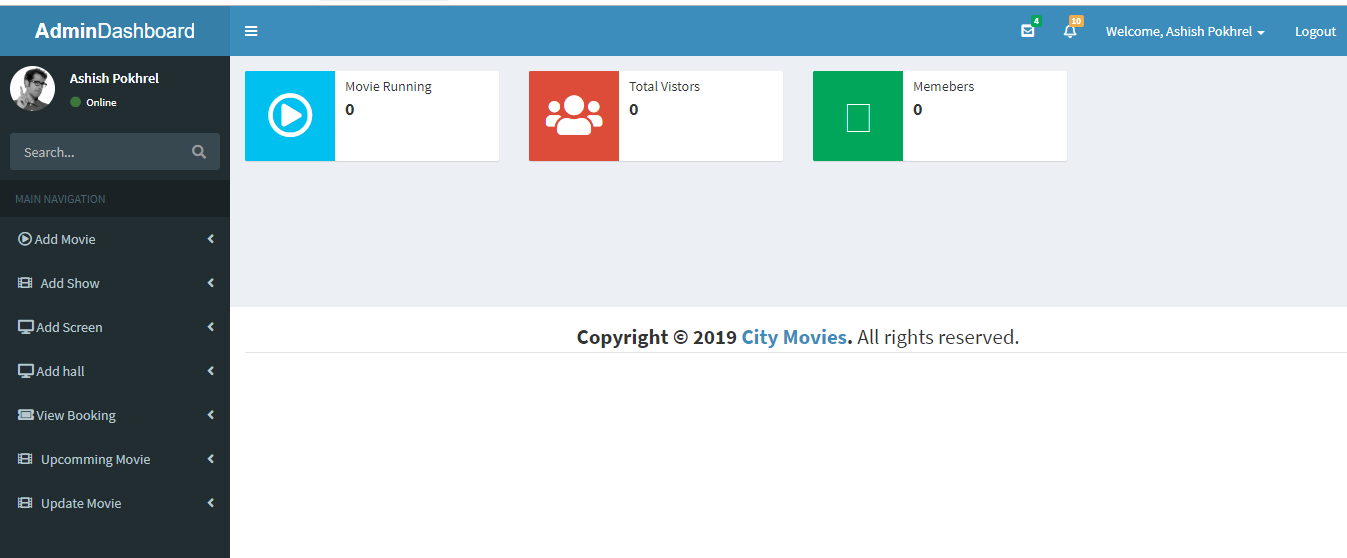


Figure 40: redirecting to admin dashboard after admin Login

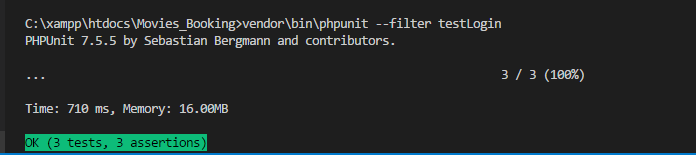


Figure 41 : White Box Testing for Amin

|  |  |
| --- | --- |
| **Test No:** | 8 |
| **Purpose of test:** | Delete Movie from database |
| **Test name:** | test Delete Movie |
| **Expected Result:** | Movie should be deleted |
| **Actual Result:** | Display with error message |
| **Test Status**: | Fail |

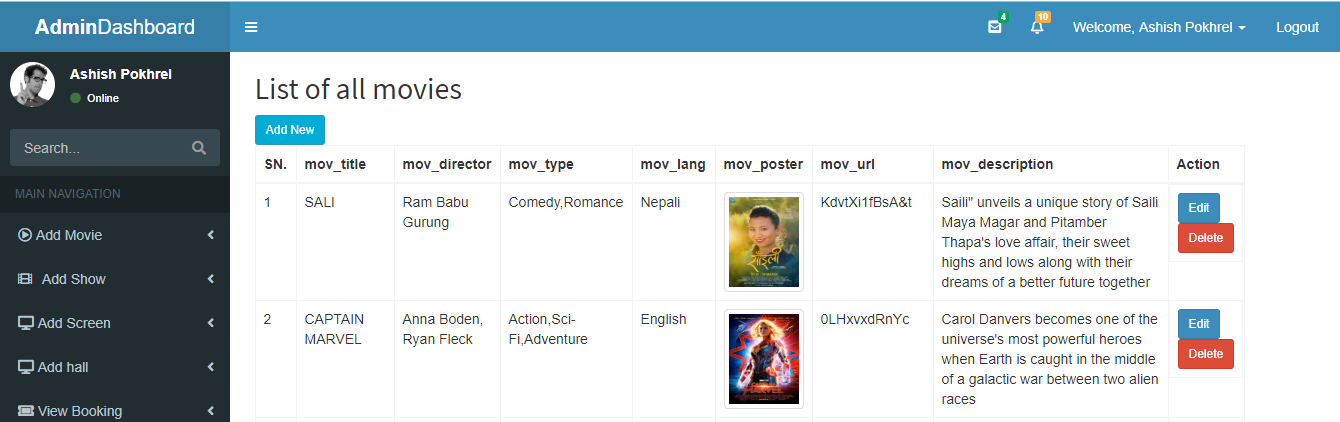


Figure 42 i.) Black Boxing testing for delete movie in database

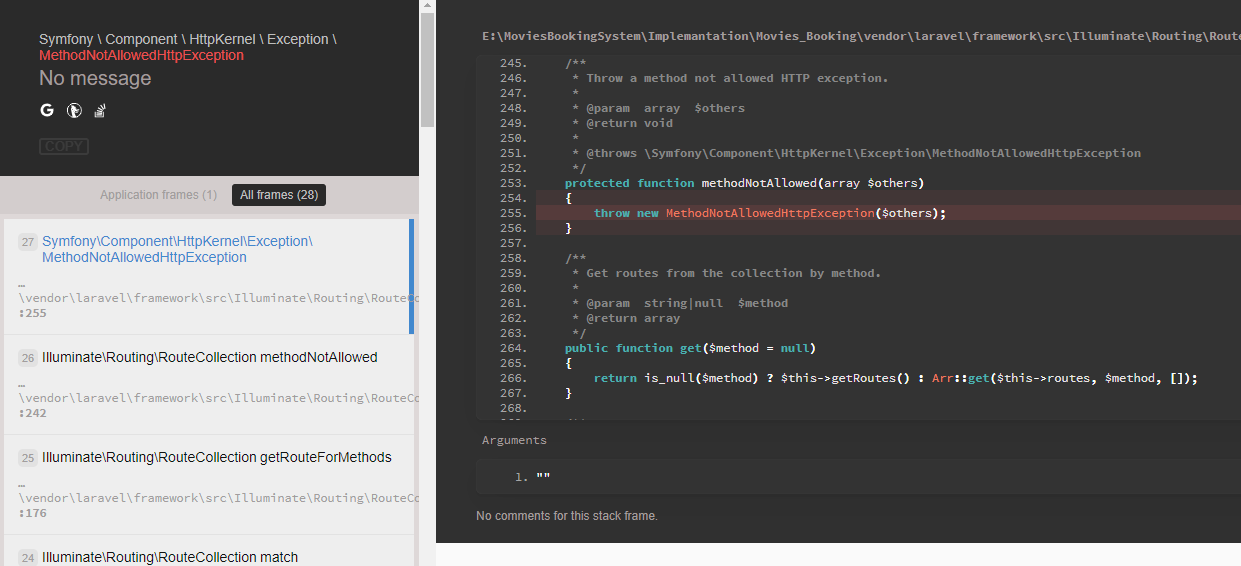


Figure 43 ii.) Error during deleting movies

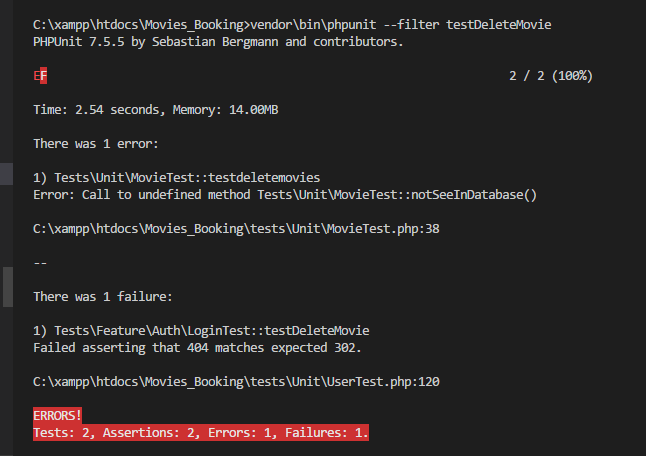


Figure 44: White Box Testing for delete Movies

|  |  |
| --- | --- |
| **Test No:** | **9** |
| **Purpose of test:** | Edit Movie from database |
| **Test Data** |  |
| **Test name:** | test edit Movie |
| **Expected Result:** | Movie should be updated |
| **Actual Result:** | Display with successfully updated message |
| **Test Status**: | Pass |

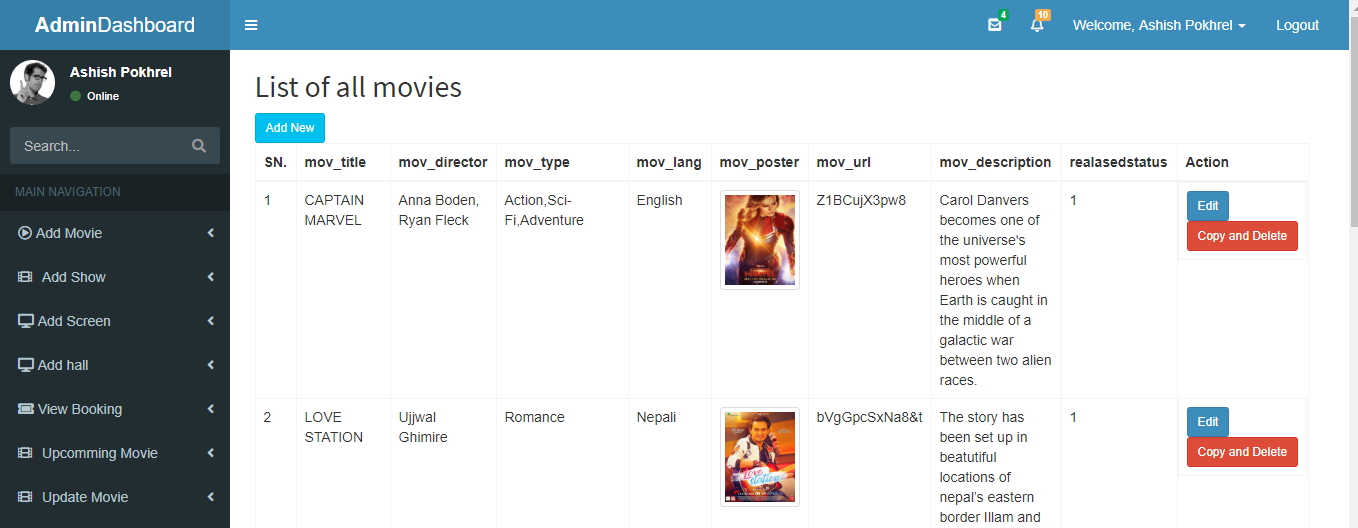


Figure 45: Black Box Testing for edit movies

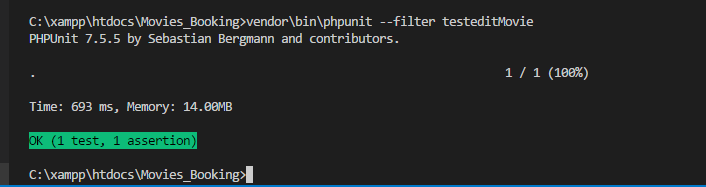


Figure 46: Black Box Testing for EditMovie

|  |  |
| --- | --- |
| **Test No:** | **10** |
| **Purpose of test:** | Add screen from database |
| **Test Data** | Audi 1 |
| **Test name:** | test add screen |
| **Expected Result:** | screen should be add in database |
| **Actual Result:** | screen is added successfully. |
| **Test Status**: | Pass |

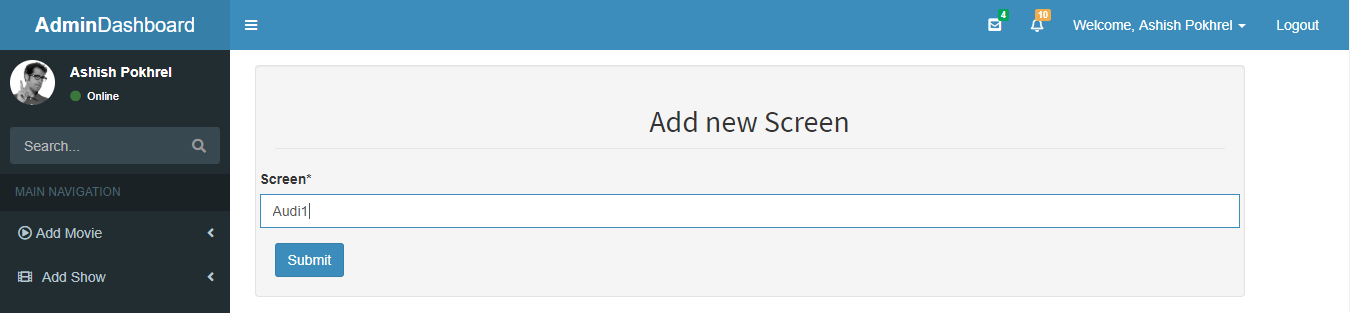


Figure 47 : Black Testing of Add screen

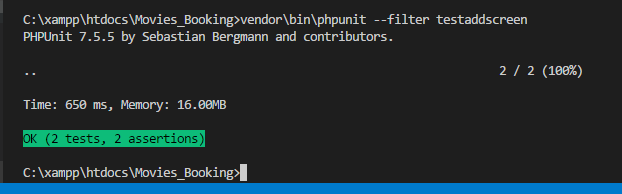


Figure 48: White Box Testing for add screen

# Chapter 6: Other Project Issues

## 6.1 Risk Management

Software Development uses different technological advancements and also required high levels of knowledge for developing any software. Due to these factors, every software development has some uncertainly which is called as Risk. (CAST, 2019)

For success of any software project, the first and foremost things is that to identify risk that will create obstacles during software development. Project Risk is also known as Potential problem for any project. Risk is also the activity which can compromise the success of any software project. It can lose cost, time of software project.

In software development, Risk Management is the process of identifying, controlling and managing risks which are evolved before or during development process. Generally, there are three process for risk management i.e. are as follows: (Proffesionalqa, 2019)

1. Risk identification

This is the first steps of risk management where Potential risks are identified which may affect software during software development.

1. Risk Analysis

This is the second step of risk management where identified risks in first steps is analysis. Each risk is analysis to know their possible outcomes and prioritization.

1. Risk Control

This is third steps of risks management where risks are managed, control and remove for get success result.

**Impact = Likelihood \* Consequence**

Risk Likelihood values are shown as follows

|  |  |
| --- | --- |
| Likelihood | Value |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Risk Consequence values are shown below

|  |  |
| --- | --- |
| Consequence | Value |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

Risk Consequence values are shown below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | Risks | Likelihood | Consequences | Impact | Solution |
| 1 | Insufficient resources | 2 | 3 | 6 | All required resources for the project should be collected. |
| 2 | Hard Disk Failure | 1 | 5 | 5 | Data must back up. |
| 3 | Requirement does not meet | 2 | 5 | 10 | Proper planning should be done. |
| 4 | Server Failure | 1 | 4 | 4 | Online Backup should be done. |
| 5 | Lack of skill | 1 | 3 | 3 | Training should provide for skills employees. |
| 6 | Scheduling problem | 2 | 4 | 8 | Divide tasks in different schedule |
| 7 | Lack of cost | 2 | 4 | 8 | Cost estimation should be done properly. |
|  |  |  |  |  |  |



## 6.2 Configuration Management

The term refers to the system which track hardware, software and related information of the system. Configuration management is involving practices of processing system changes systematically with updating system while maintain the system integrity. To achieve of goal of the system, configuration management should be implemented with details policies, procedures to manage to version. (Guru, 2019)

### 6.2.1Version Control

Version Control is a software tools that helps to manage different version of source code of developer’s teams. It can allow us to re-back whole project into previous state. It can easily find problem when someone modifies source code lastly. There are three types of version control i.e. Local version control, centralized version control and distributed version control. (Git, 2019)



Figure 49: GitHub for Movie Booking

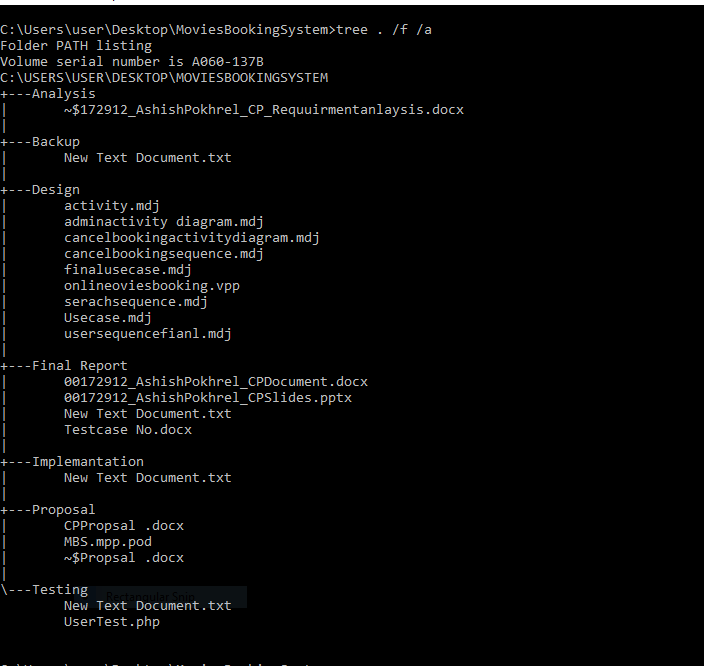


Figure 50: Tree structure

## Future Work

Given Project is academic so that I have provided with limited time, so that I cannot include all features which are told in proposal. These features will come in future update.

* Online Payment System
* Rating and Reviewing Movie
* Security Features
* Live Chat Features to

## User Manual

Clients are new for the system after the new system deployment in the client environments. To teach them how to used a new system, user manual is created for telling working mechanism of the system.

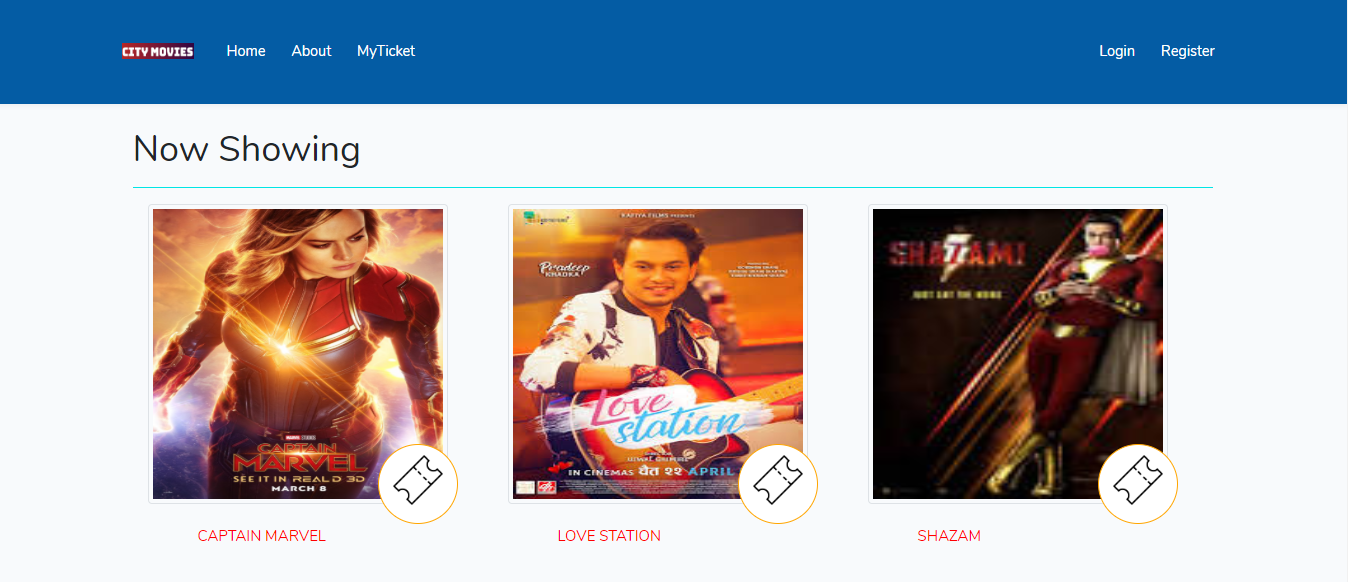
* **Un-Registered User**

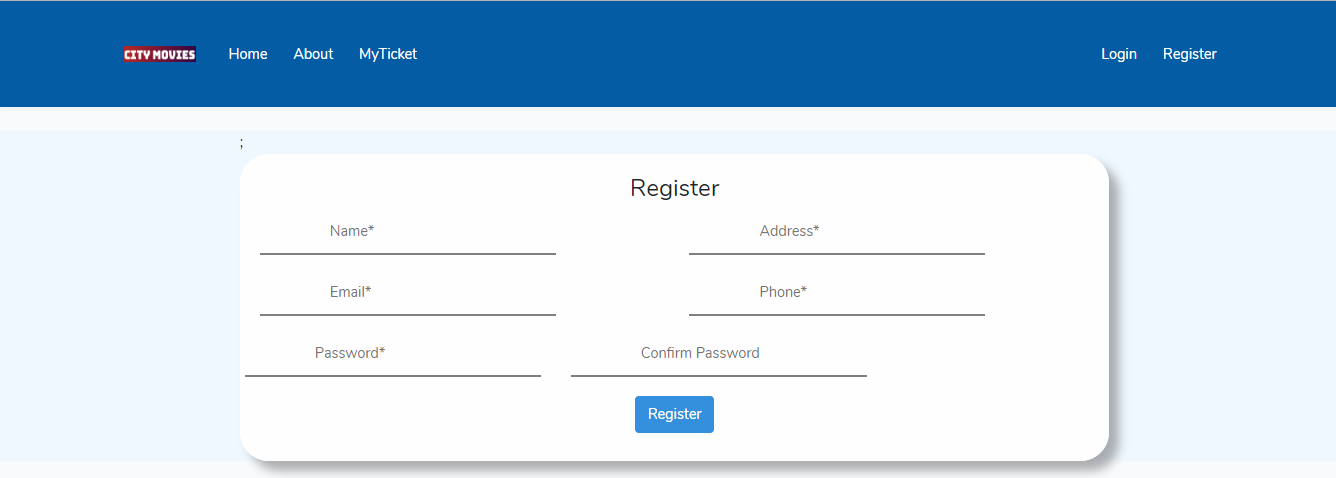
Unregistered can Registration and view only details of movies and showtime to the system. They cannot book movies without registration to the system. Unregistered user can registration to the system by:

**Steps for Registration:**

* Click on ***Register Button*** on top right side.

* Then, Registration form will open and fill all details in textbox and then, click on ***Register button***.





Click This Register Button for Register after filling form

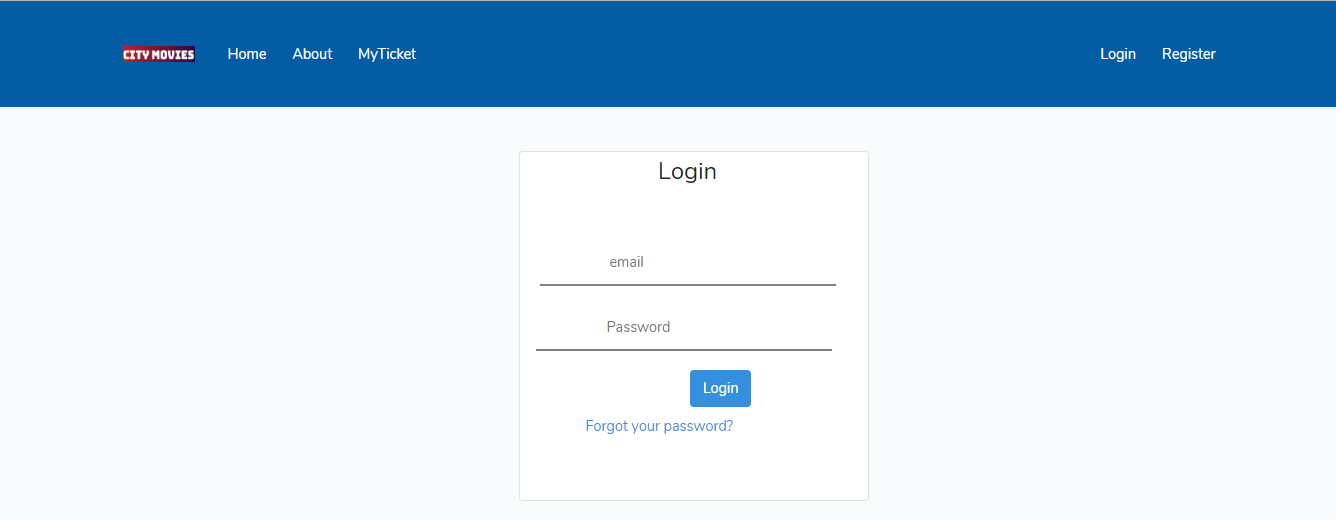
**Registered User**

Registered User can Login, view movies, book movies and update their profile and cancel booking movies.

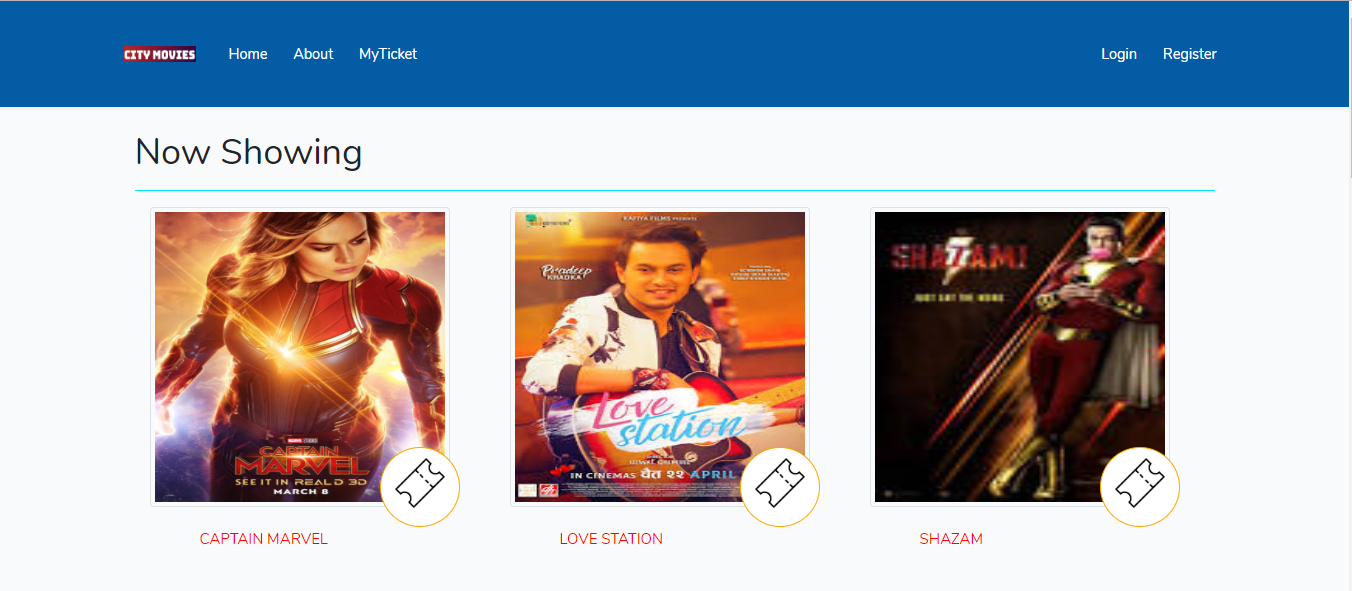
**Steps for Registered user**

* **Login**

Registered should type ***username and password*** which are enter during registration time and click on ***login button*** after that Home page appears.

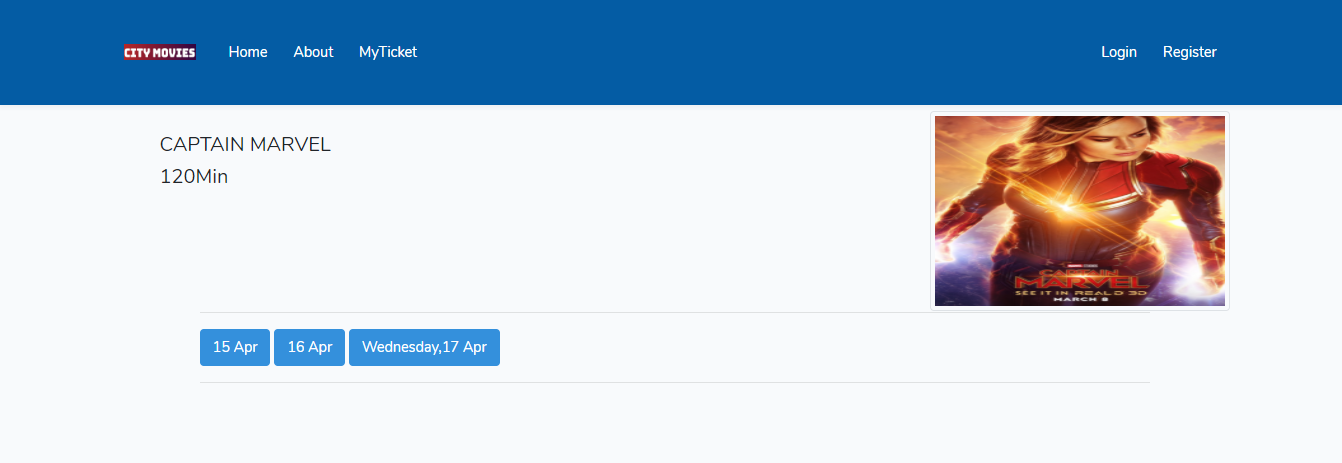


* Click on ***Button with Ticket image*** from selection of Movie.

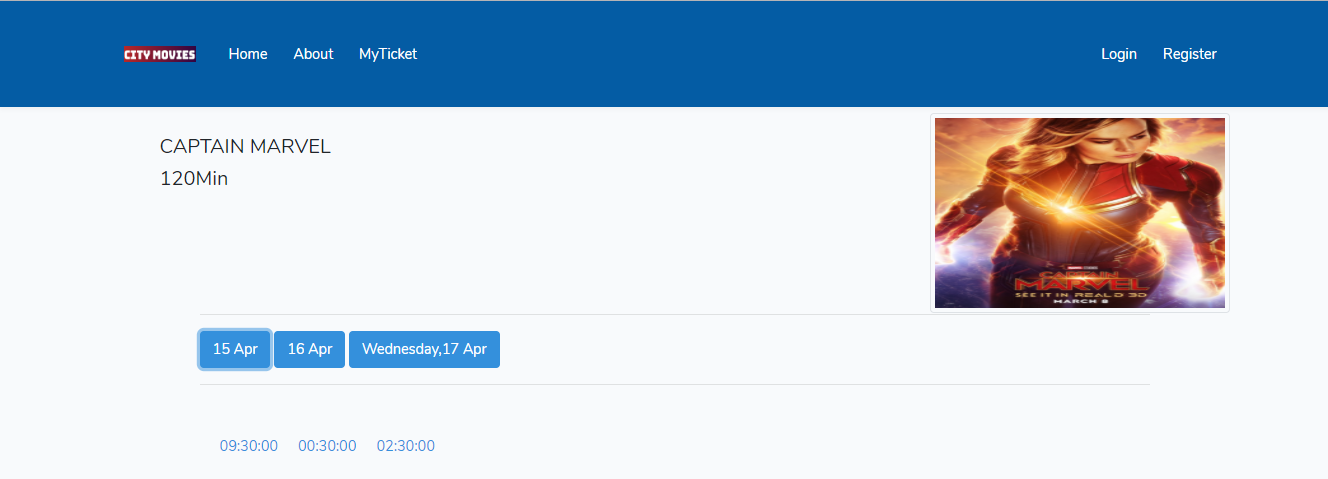


Click Here for select Movie

* Click on ***Button*** particular date for showtime and after clicking it display available showtime and ***Click on showtime which you want***.

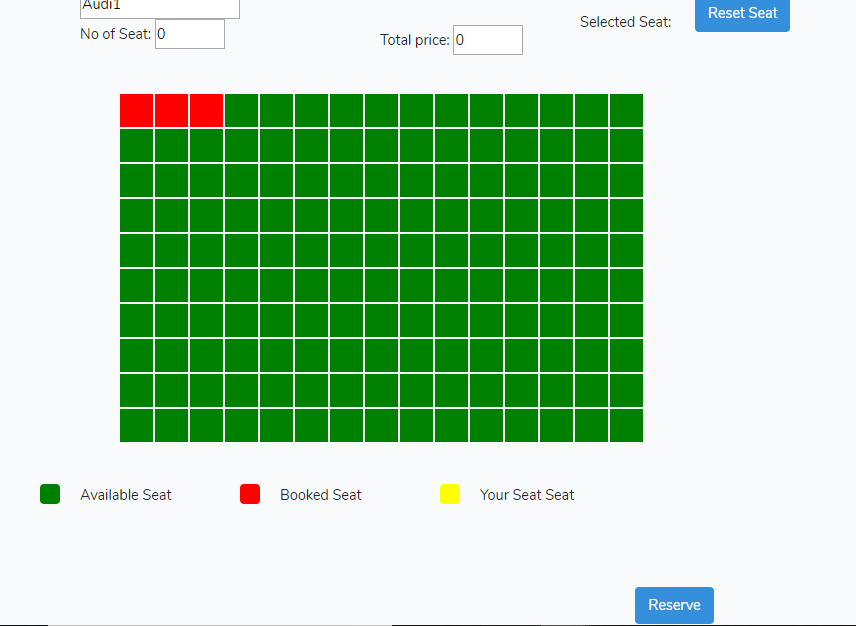


Click Here to select Showtime



Select any showtime

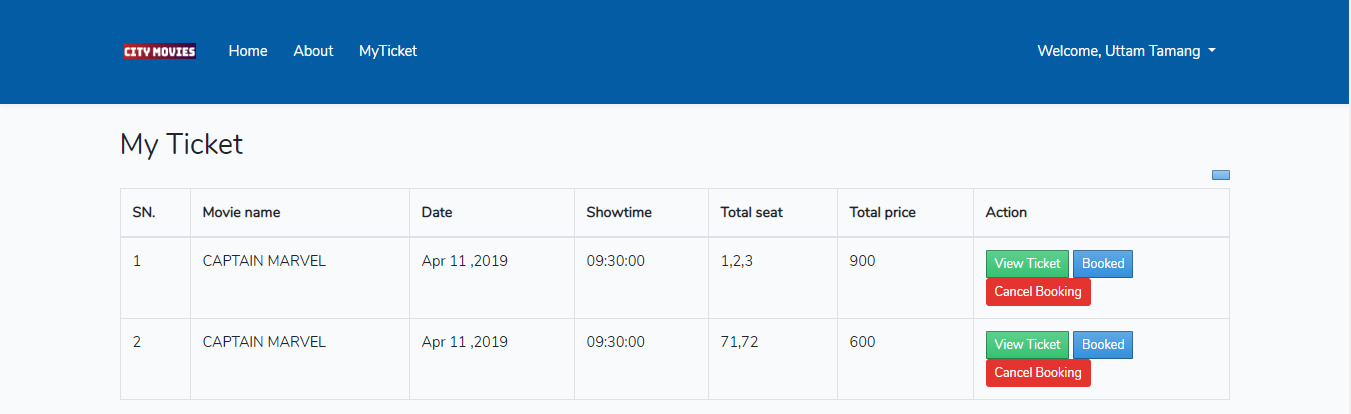
* Select seat for the movie. ***One click***, for select and ***double click*** for deselect. If you want to select ready unselect seat, then you should ***Reset seat Button*** again.
* Click ***Reserve Button*** for Booking Movie Seats.



Click Here for Reserve seat

Click reseat seat if you want to select already deselect seat

* **Cancel Booking**
* Go to ***My ticket*** on top navigation bar.
* Click ***View Ticket*** for Generating Ticket.
* Click ***Cancel Booking*** for cancel booking of reserve movie seats.



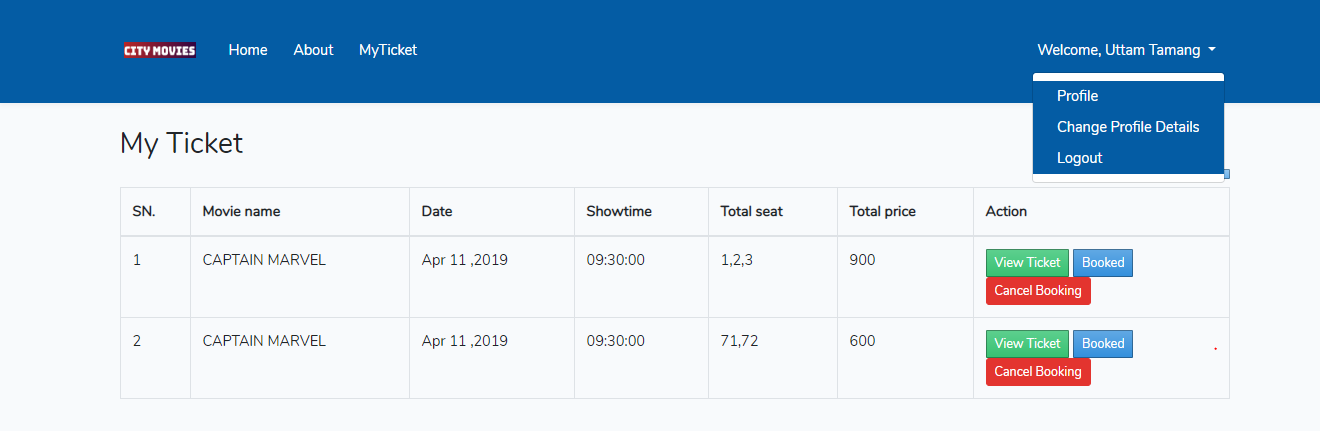
View Ticket

For cancel Booking

* **For Logout, profile Update**
* Click on right side of navbar on ***downward button***
* Click ***Profile button*** for update profile.

Update profile

* Click ***Logout button*** for Logout from system.



Logout

## Limitation

Following are the limitation of project:

* Online Payment System is not available.
* User can only cancel booking not to update booking.

# Chapter 7: Conclusion of the project

Online Movies Booking system is a customized and user friendly a web application where you can book seat for movies in theaters from anywhere anytime. It has facilities of booking movies, read reviews of movies, watch trailer, see lists of upcoming movies. Admin can add lists of movies. Suitable breakdown and scheduling are done properly. Design pattern MVC and waterfall methodology is used for the project. The only one limitation of the system is online payment system is not available.

# Chapter 8: Reference and Bibliography

CAST, 2019. [Online]   
Available at: https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects

Geeks, 2019. [Online]   
Available at: https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/

Git, 2019. [Online]   
Available at: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Guru, 2019. [Online]   
Available at: https://www.guru99.com/database-design.html

Guru, 2019. [Online]   
Available at: https://www.guru99.com/software-testing-introduction-importance.html

Guru, 2019. [Online]   
Available at: https://www.guru99.com/software-configuration-management-tutorial.html

Mitre, 2016. [Online]   
Available at: https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-architecture

Proffesionalqa, 2019. [Online]   
Available at: http://www.professionalqa.com/risk-management-activity

ReQst, 2019. [Online]   
Available at: https://reqtest.com/requirements-blog/requirements-analysis/

requirement.com, 2004. [Online]   
Available at: http://www.requirements.com/Glossary/RequirementsPrioritization/tabid/121/Default.aspx

SDC, 2011. [Online]   
Available at: http://sdc.net.au/services/application-development/analysis-and-specification.aspx

SHARMA, L., 2018. [Online]   
Available at: https://www.toolsqa.com/software-testing/waterfall-model/

Sourcemaking, 2019. [Online]   
Available at: https://sourcemaking.com/design\_patterns

Techopedia, 2018. [Online]   
Available at: https://www.techopedia.com/definition/25813/use-case

TechTarget, 2019. [Online]   
Available at: https://searchsoftwarequality.techtarget.com/definition/development-environment

UML, 2019. [Online]   
Available at: https://www.uml-diagrams.org/uml-24-diagrams.html

Visual-Paradigm, 2019. [Online]   
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

# Chapter 9: Appendix

Implementation of Laravel MVC design pattern

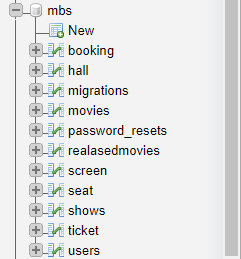


Figure 51: Tables of movies booking system

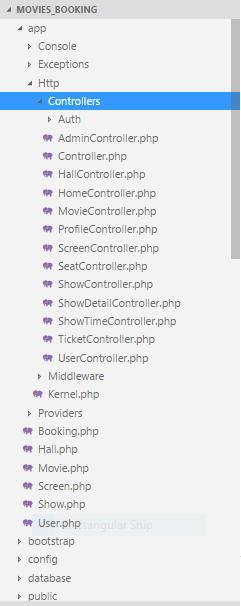


Figure : Project Folder

v

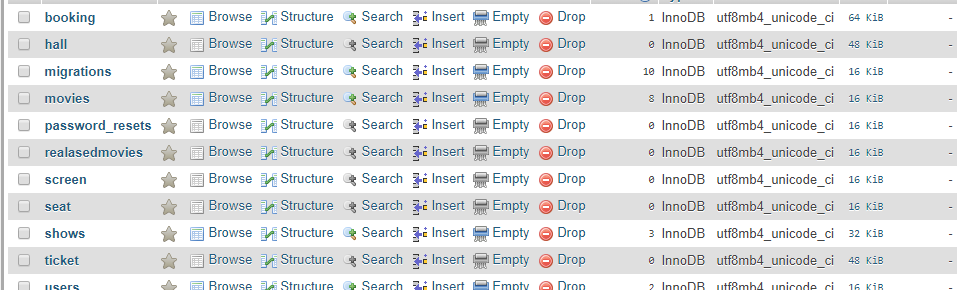


Figure 53: Database for Movies booking system

**Model**

1. **User Model**



Figure 54: User Model

1. Movie Model



Figure 55: Movie Model

1. Show Model

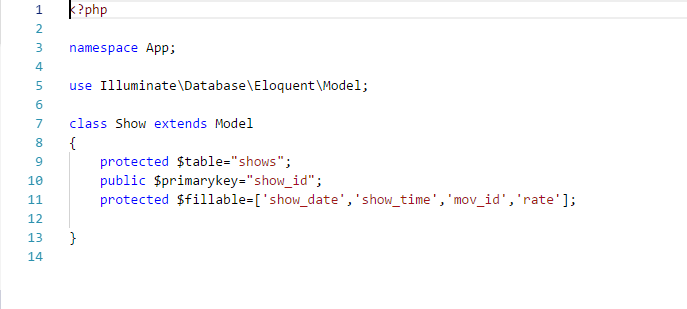


Figure 56: Show Model

Iv) Booking Model

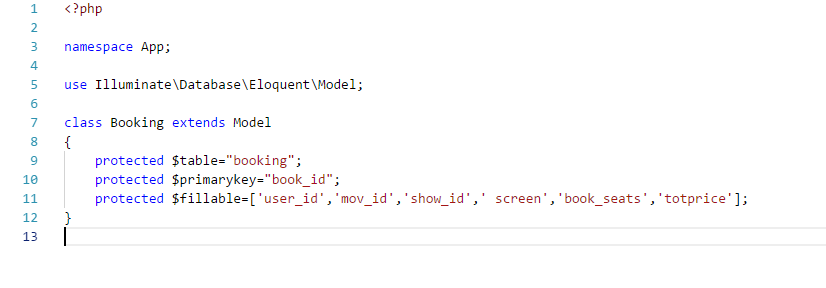


Figure 57: Booking Model

V) Screen Model

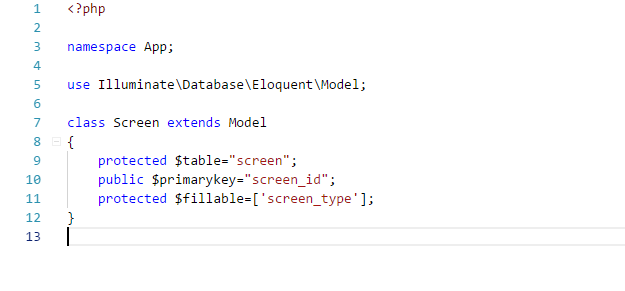


Figure 58: Screen Model

Vi) Hall Model

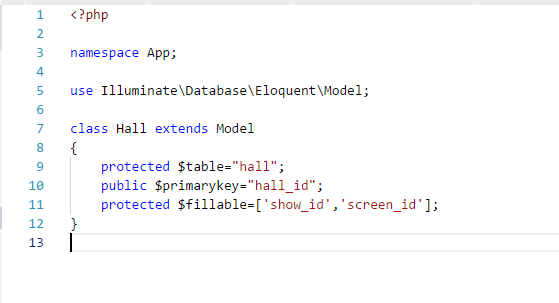


Figure 59: Hall Model

**Controller**

1. **User Controller**
2. 

Figure 60: User Controller with profile Update

Ii) Movie Controller

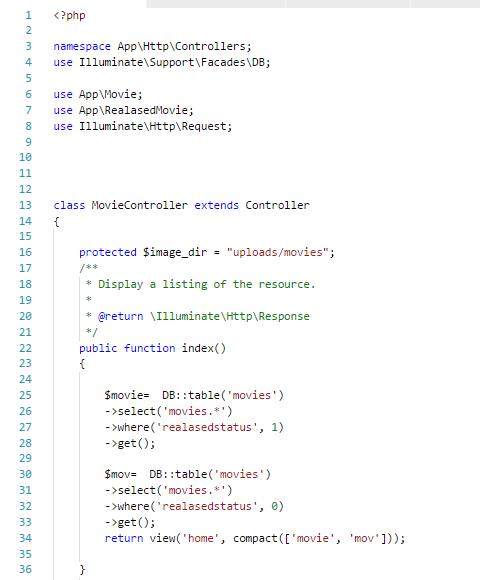


Figure 61: (i) Movie Controller inserting movies in database



Figure 62: (ii) Movie Controller update movie in database

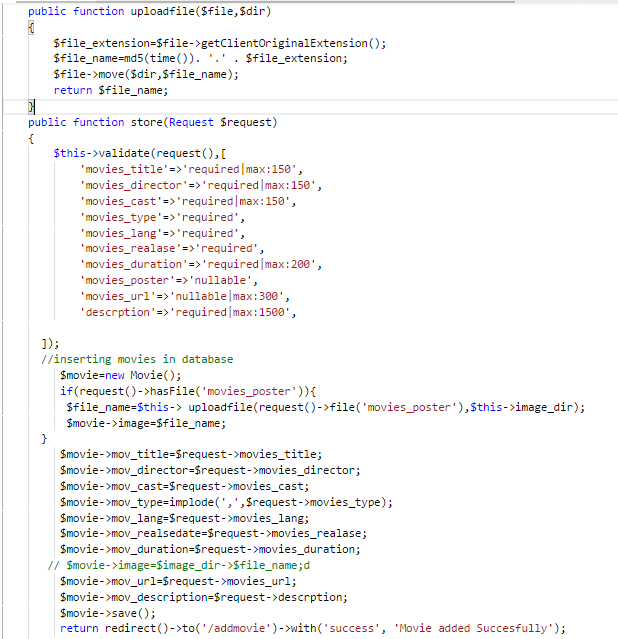


Figure : iii) inserting movie into Database

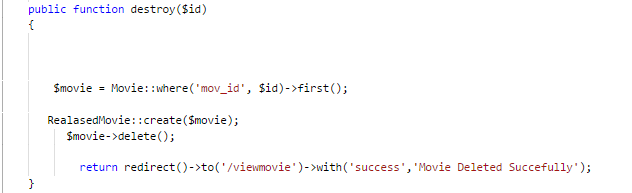


Figure : Deleting Movie from database

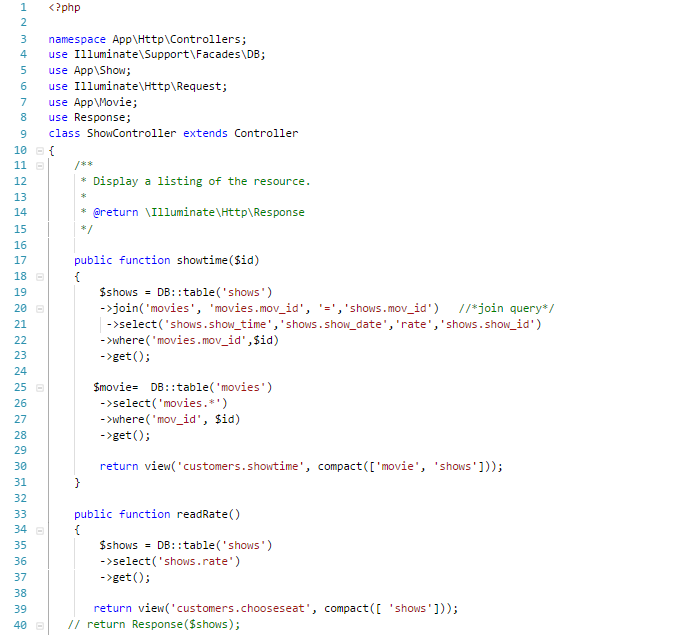
Iii) Show Controller

Figure 65:)(i) Show Controller for viewing showtime in view code

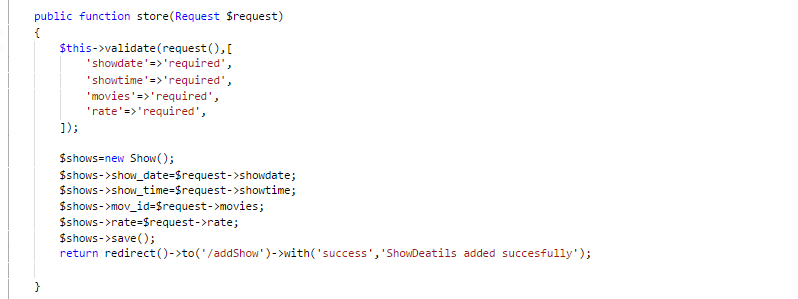


Figure 66: (ii). Show Controller inserting showtime in database

Iii) Seat Controller



Figure 67: Seat Controller Booking Movie Code

1. Ticket Controller

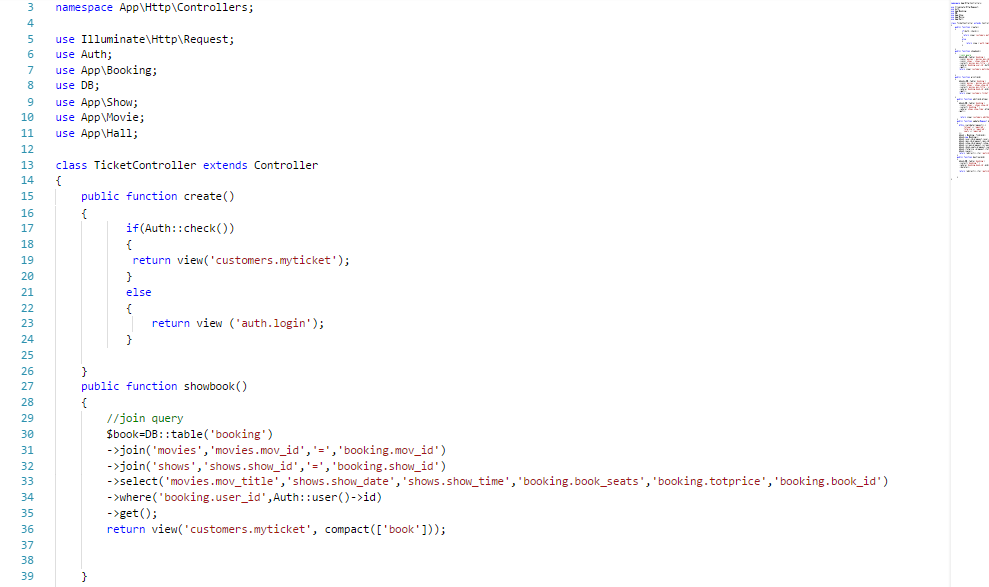


Figure 68: (i) Ticket Controller view and generate ticket

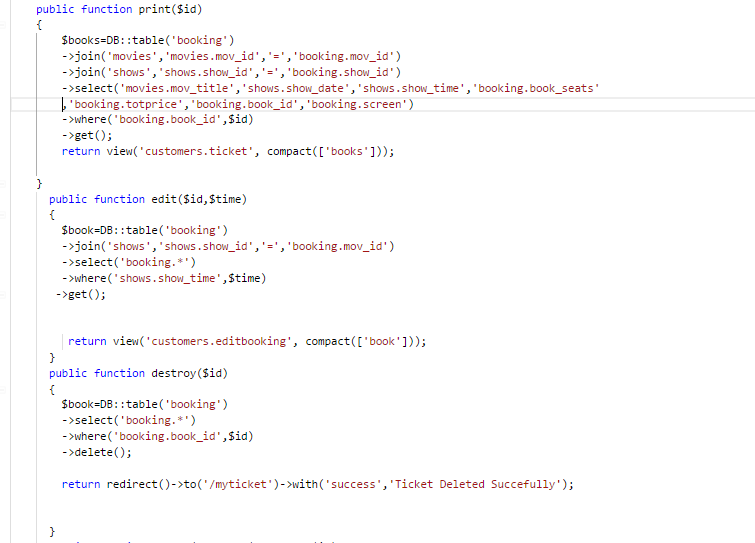


Figure : ii) Print and cancel ticket code

V) Hall Controller

Figure 70: Hall Controller inserting hall

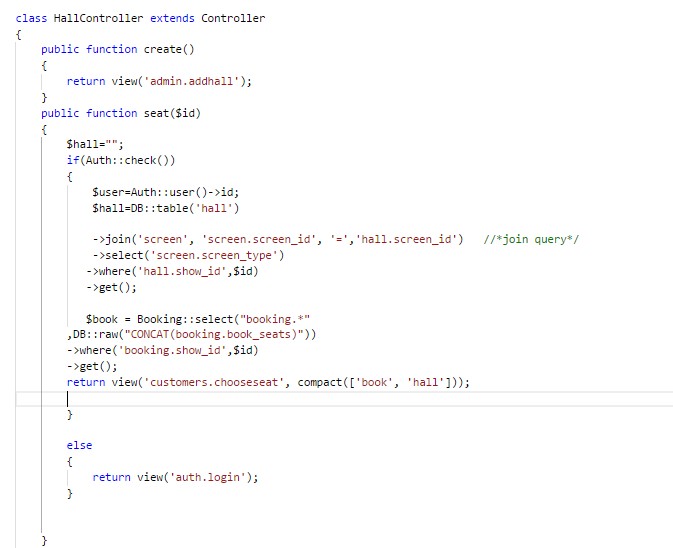


Figure 71: View screen in view form



Figure 72: inserting screen in database



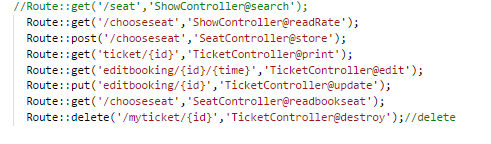


Figure 73 :Routes

1. UI Design

ii) Seat Chart Design Code

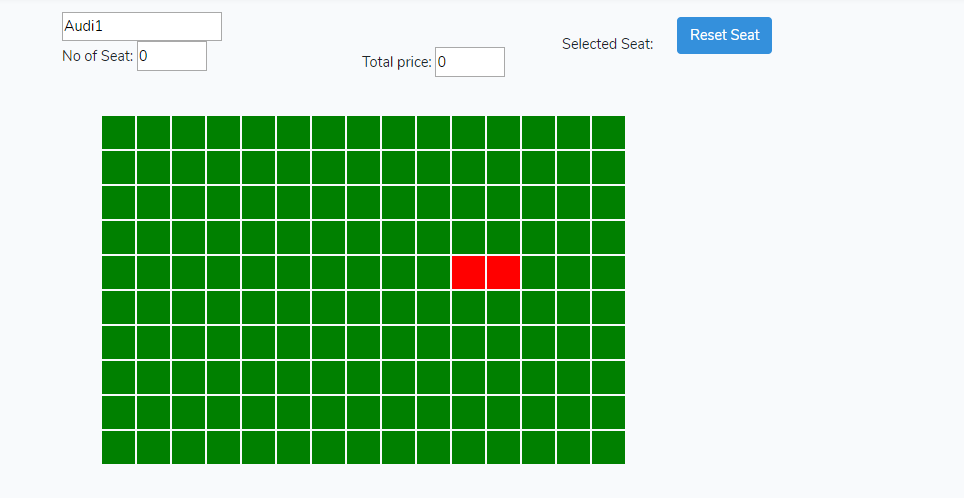








Figure : UI design code for Seat chart

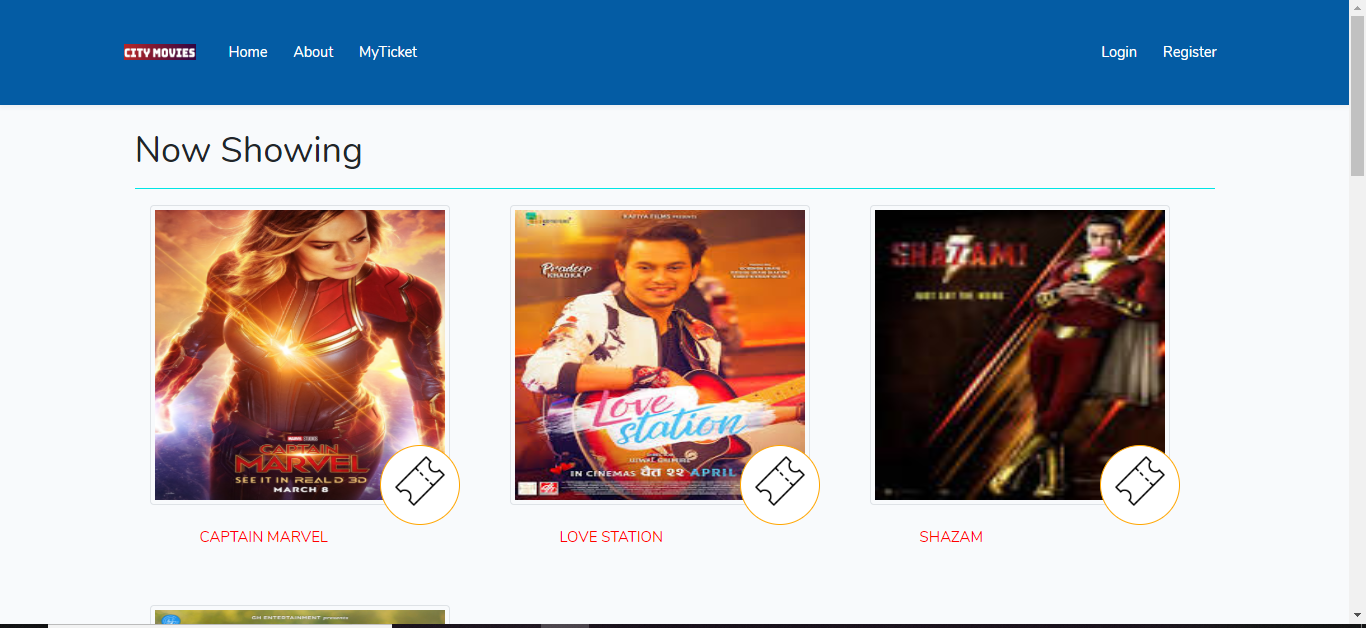




Figure : Viewing Movie in home page

1. Admin Dashboard

Figure 76: Admin Panel

Iii) Add Movie Form

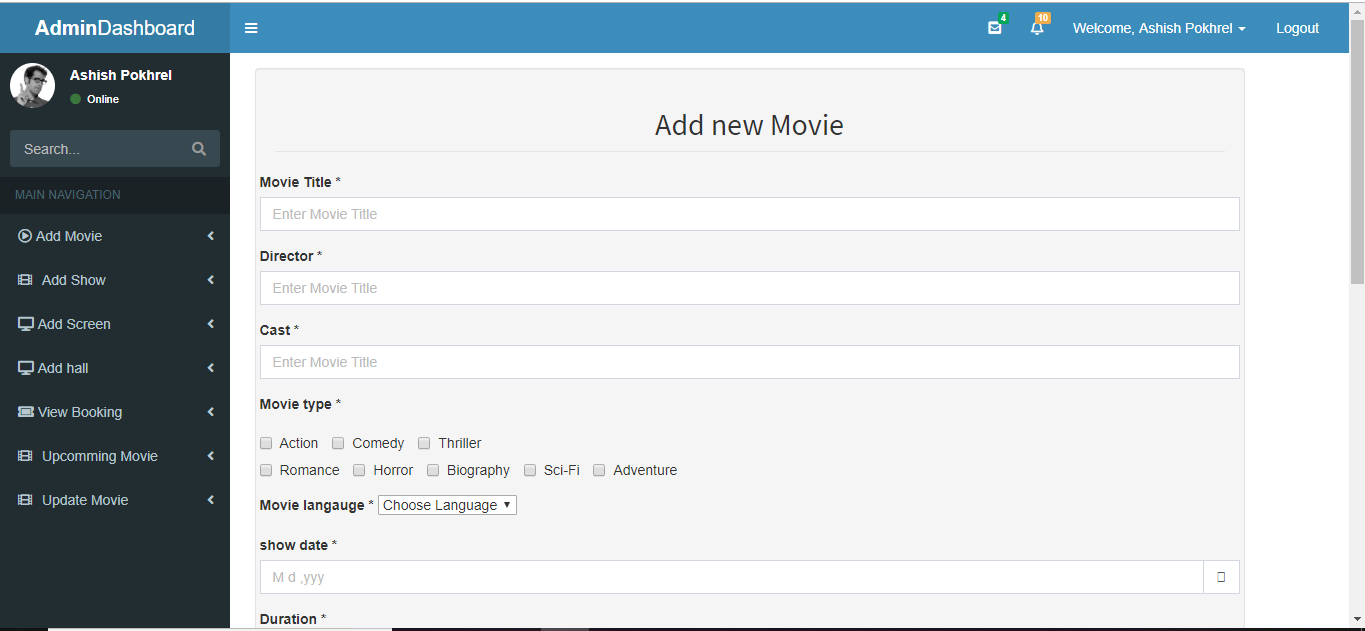


Figure : Adding Movie







Iv) Add Showtime

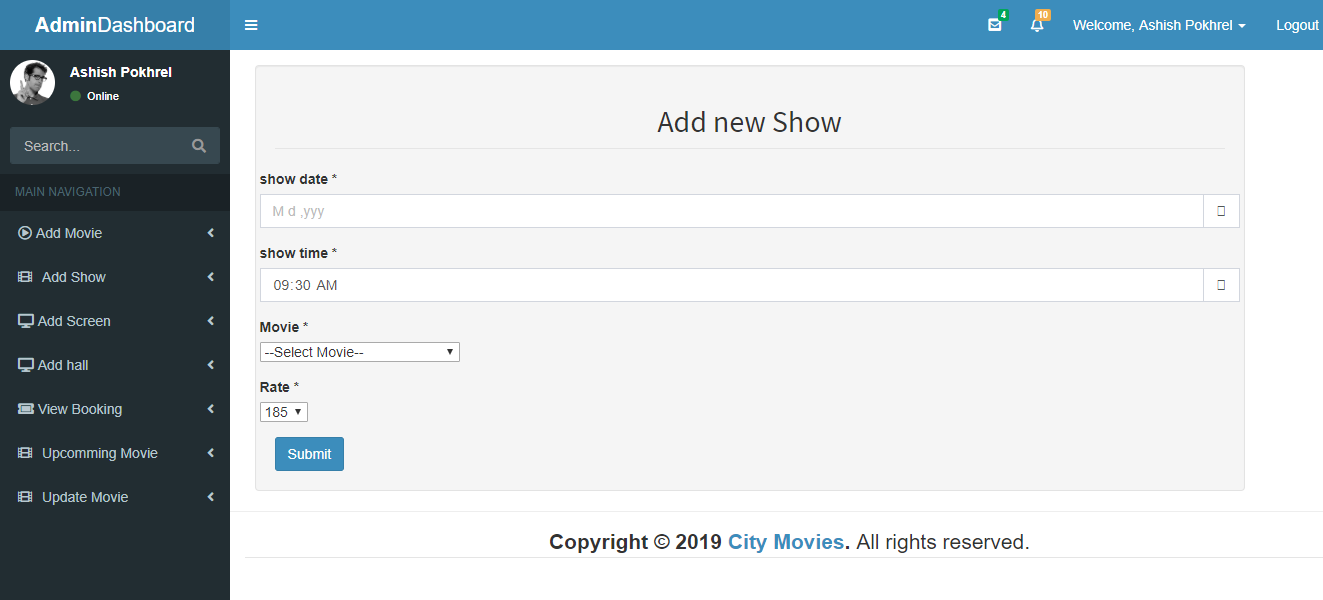




Figure : adding showtime