ONLINE THEATRE TICKET BOOKING SYSTEM

Project Report Submitted By

AKSHAI BIJU

Reg. No: AJC17MCA-I003

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**ONLINE THEATRE TICKET BOOKING SYSTEM"** is the bona fide work of **AKSHAI BIJU (Reg.No: AJC17MCA-I003)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

Ms. Merin Chacko
Internal Guide

Rev.Fr.Dr. Rubin Thottupurathu Jose
Coordinator

Rev.Fr.Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **"ONLINE THEATRE TICKET BOOKING SYSTEM"** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date: **AKSHAI BIJU**

KANJIRAPPALLY **Reg. No: AJC17MCA-I003**

ACKNOWLEDGEMENT

First and foremost, I thank God Almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob for** providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Rev.Fr.Dr. Rubin Thottupurathu Jose** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms. Merin Chacko** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

AKSHAI BIJU

ABSTRACT

Online Theatre Ticket Booking System is to provide the users facility to book tickets for a show and to collect information regarding the films and theatres. Client has to register at the positioning to book tickets to the show. When choosing the show, the user is given a seating layout in order that he will choose seats of his selection. henceforth he's redirected to the payment entryway for creating a group action. User will update his profile, take a print out of the price tag and conjointly read his booking history. Users can book for snacks and drinks online. Each Theatres can add the seats according to their preference. Theatre Owner can manage the food booking along with ticket booking. Users can book for tickets and also snacks. Users can also see the order history of the snacks. Users can select the seats according to their preference The most aim of the project is any user they can access the system through web site at any time while not planning to the corporate. As a replacement user conjointly, he will collect all info like as movies out there, list of theatres.

CONTENT

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Customers may view the contents of any movie show at any time and may book any movie ticket as needed. The program automatically calculates the subtotal and grand total. When a visitor decides to finally book the ticket, the order information including the buyer's name, address and billing instruction is stored in the database securely and payment has been made. The combo booking is also provided at the time of booking the ticket and there's a wonderful facility of delivering the combos at your seat when you are watching the movie. You need to register a new user whenever you have first visited or site then for future it will be stored in our database permanently and you can book you movie ticket at any time you want with this username and password.

1.2 PROJECT SPECIFICATION

The proposed system enables the admin to manage theatres and movies. Theatre owner can manage all the activities of his theatre. It is a web based online project. The main aim of the project is any user they can access the system through website at any time without going to the company. As a new user also, he can collect all information like as a movie Available, list of theatres

The system includes mainly 3 users. They are:

1. Admin Module

The Administrator is the one who maintaining the users and also theatre. He can create the theatre owner. Admin is the one who adding the movies upcoming movies. He can also view the available theatres and if a theatre is no more working, he can disable that theatre. He can add movies by the requests from the theatre owner

2. User Module

The user can first Registration in enter the personal details, and User login and if you want to update personal Details and you can update it. The user collects all information like Movies details, Théâtre Details. The Users will search for the movie and then go for theatre then booking the tickets online. If he wants to cancel the movie that he booked he can cancel it. He can also order for snacks.

3. Theatre Owner Module

Theatre owner is the one who manages the shows on the particular theatre. Theatre owner can view the total bookings. If the movie he wanted to run on his theatre is not available then he can request for the movie he wanted to the admin. He can see all available shows that currently running. If the theatre is rebuilt or add new screens, he can add new screens according to that. All the activities around the particular theatre are managed by theatre owner. He can add snacks and drinks and manage snack booking. He can customize seats of his own theatre and add seats count on their theatre.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

Today, there are many web-based systems are available which provides the online booking of a movie ticket, but the refund is not an easy process while in the present system once someone has chosen the seats, they can't change it. The user who has booked the ticket needs to go to the booking counter because there is no electronic ticket system, so the customers have to wait in the queue.

2.3 DRAWBACKS OF EXISTING SYSTEM

- It is scripting language
- There is server controls in asp
- It Doesn't Support language interoperability
- Lack of security of data
- Man power
- Time consuming

2.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive. The online cinema ticket booking system will provide facility to book tickets for a movie online. A user can book tickets anytime and anywhere as the system will be online based. The Online Theatre Ticket Booking System will provide detailed information so that a customer can know about the movie and based on the information customer will book the ticket. The user can search for the movies which will be going to release in the future, so they will have an option to book the ticket in advance. The Online Theatre Ticket Booking System will make the user experience much better than the present system. The Online Theatre Ticket Booking System will provide a much better experience for booking of movie tickets. If seats are available, the customer can change the position of the seats while they will get an option to cancel of tickets.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- User friendliness and interactive.
- Minimum time required.
- Security of data.
- Minimum time needed for the various processing.
- Greater efficiency.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it give an indication of the system is economically possible for development.

The economic analysis is done to determine the benefits and savings that are expected from candidate system and compare them with costs. Thus, coming to a conclusion that weather the system is economically feasible or not. This system is cost effective as well as time effective, thereby making it economically feasible. This study presents tangible and intangible benefits from the project by comparing the developments and operational costs. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

ONLINE THEATRE TICKET BOOKING SYSTEM, GUI is simple so that users can easily use it. ONLINE THEATRE TICKET BOOKING SYSTEM is simple enough so that no training is needed.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i3

RAM - 4 GB

Hard disk - 50 GB

3.2.2 Software Specification

Front End - HTML, CSS

Backend - MYSQL, PHP

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page, it now stands for PHP:HypertextPreprocessor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting webpage. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, youneed a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

3.3.3 CSS

While HTML is used to define the structure and semantics of your content, CSS is used to style it and lay it out. For example, you can use CSS to alter the font, colour, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. In the proposed system CSS 3 is used. Animation using CSS is also used to give the website a unique look & feel

\

3.3.4 HTML

HTML (Hypertext Markup Language) is the most basic building block of the Web. HTML is a fairly simple language made up of elements, which can be applied to pieces of text to give them different meaning in a document, structure a document into logical sections, and embed content such as images and videos into a page. In the proposed system, HTML 5 is used to carry out the design part of the webpage.HTML validations are also used.

3.3.5 JAVASCRIPT

JavaScript was initially created to “make web pages alive”. The programs in this language are called scripts. They can be written right in a web page’s HTML and run automatically as the page loads. In the proposed system loading screen, different popup animations are implemented using Java Script.

3.3.6 JQUERY

jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation. The jQuery library contains the following features:

HTML/DOM manipulation, CSS manipulation, HTML event methods, Effects and animations,
AJAX

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. After

some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- Statechart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains fourcomponents.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their

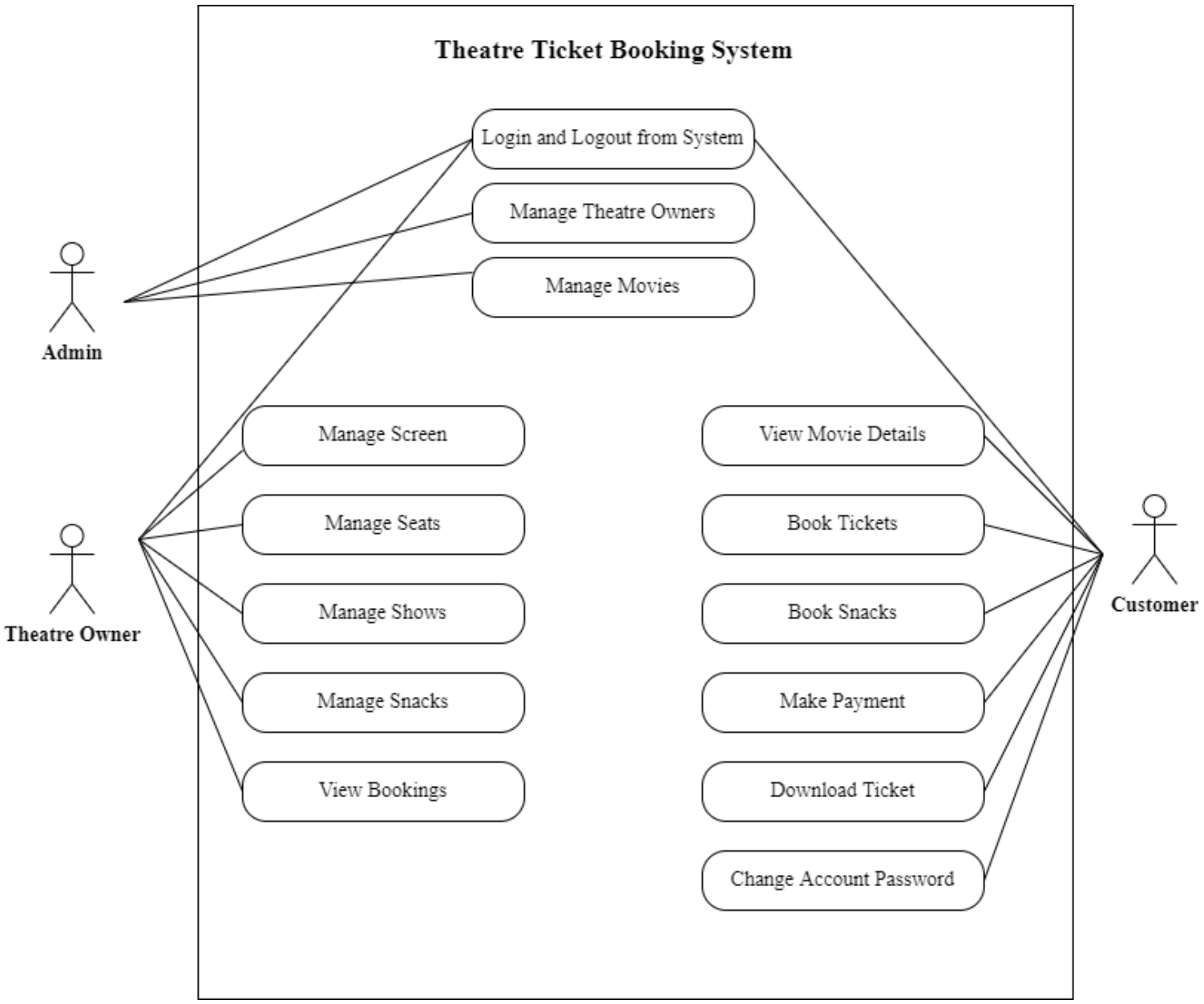
roles.

- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig 1: Use case diagram for Online Theatre Ticket Booking System



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

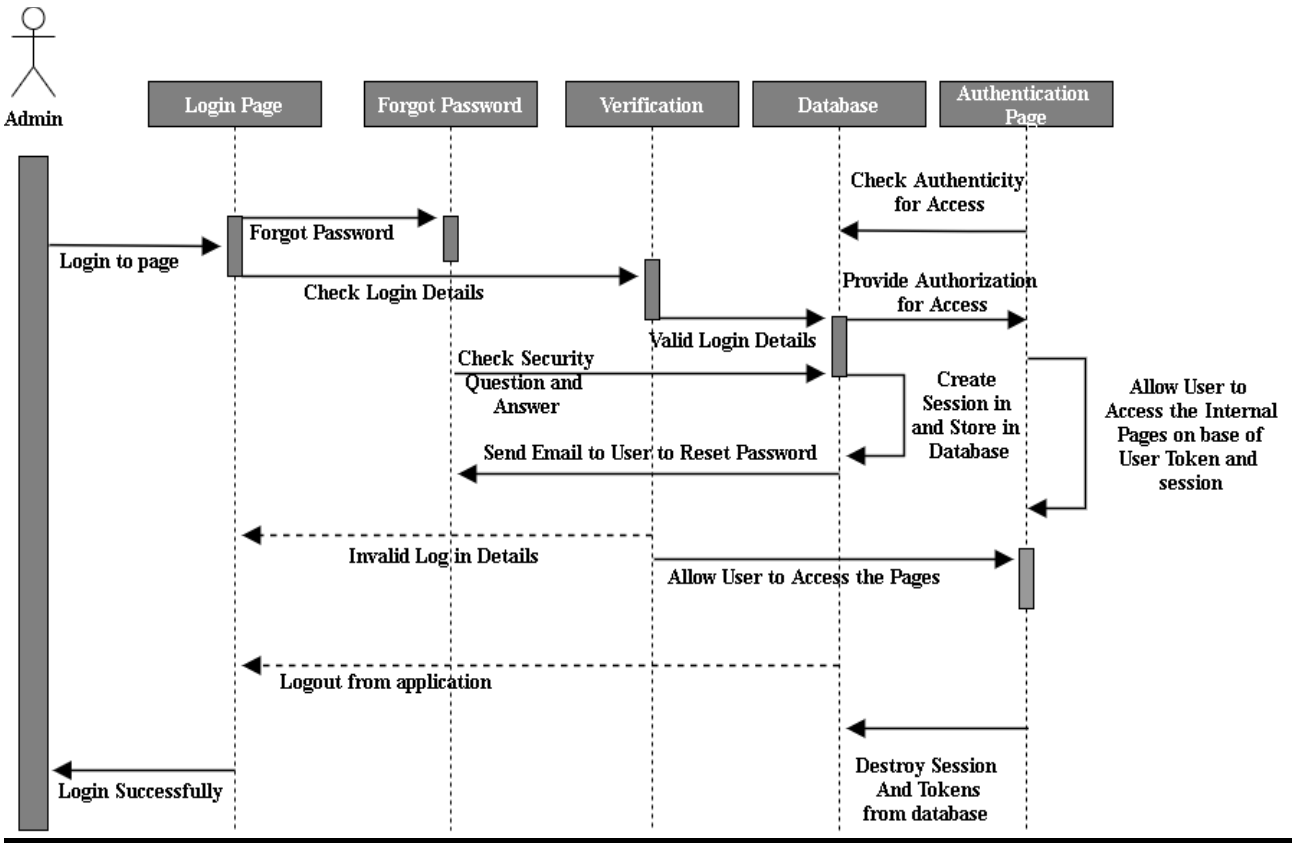
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

iv. Guards – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Fig 2: Sequence diagram for Online Theatre Ticket Booking System



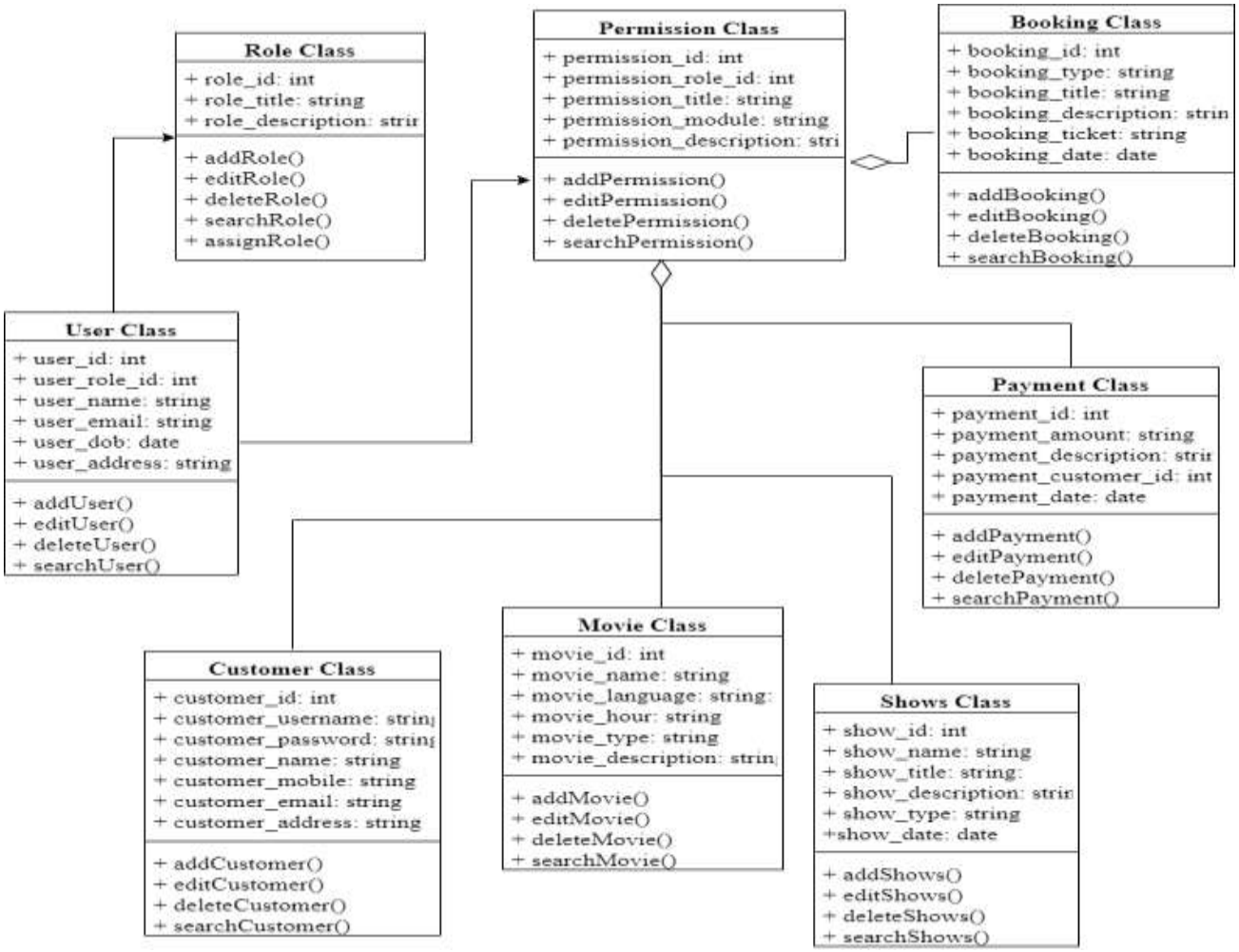
4.2.3 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Fig 3: Class diagram for Online Theatre Ticket Booking System



4.2.4 ACTIVITY DIAGRAM

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system.

Fig 4: Admin Activity diagram for Online Theatre Ticket Booking System

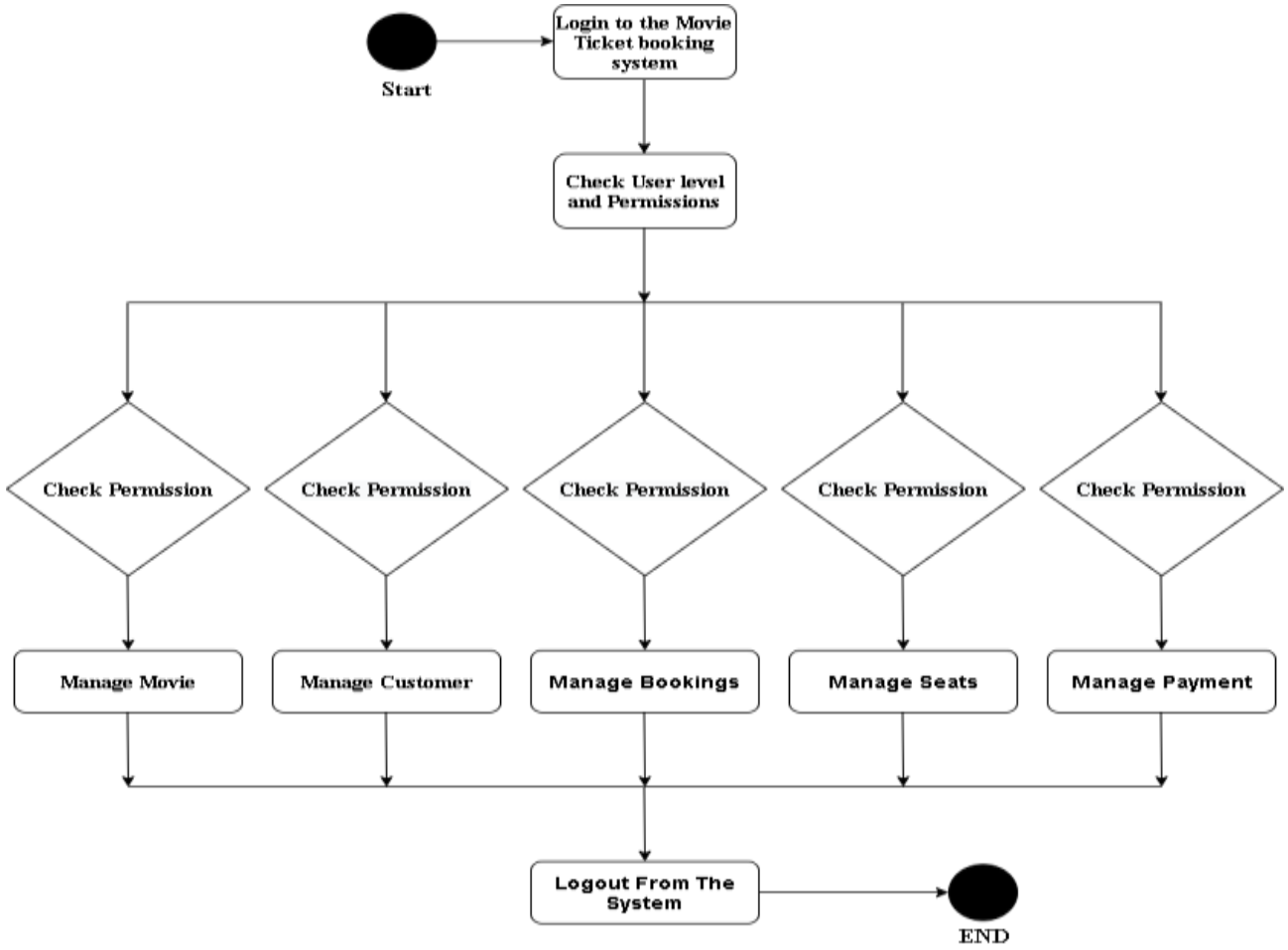
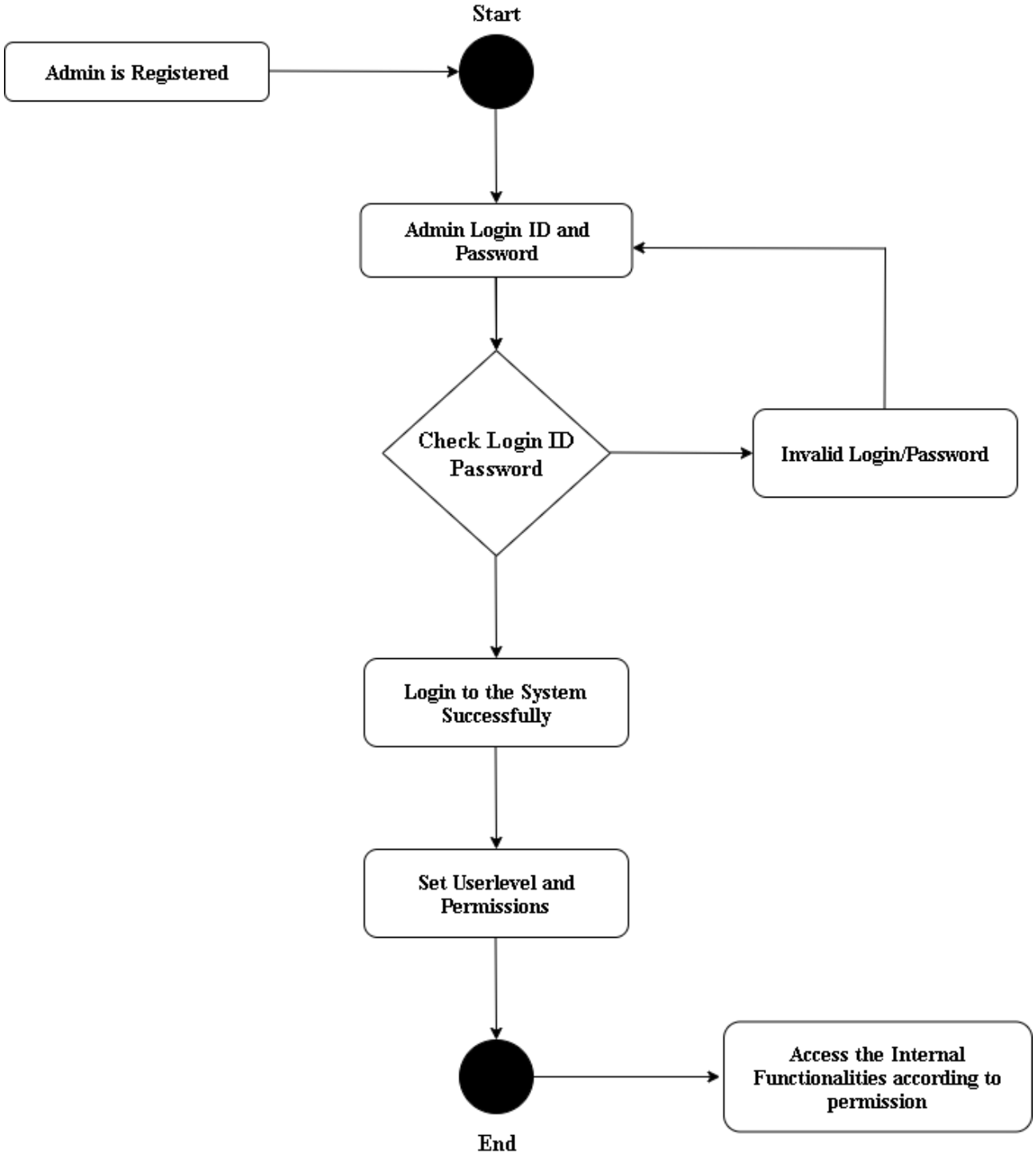


Fig 5: Login Activity diagram for Online Theatre Ticket Booking System



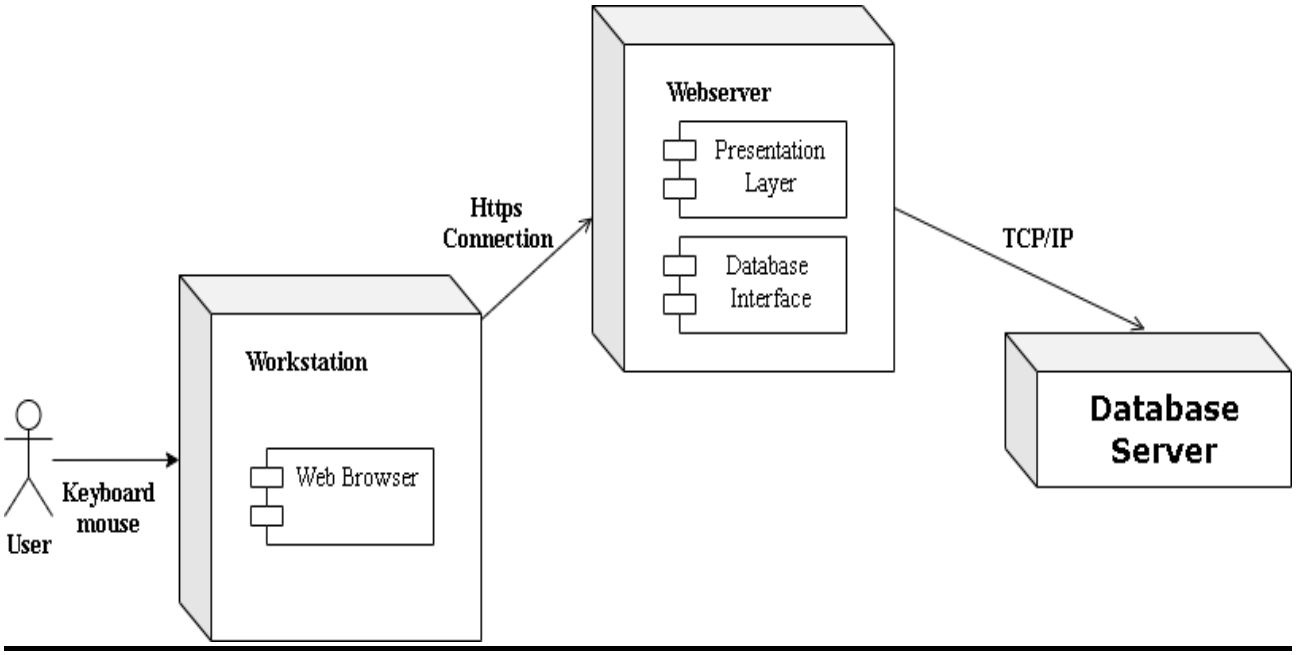
4.2.5 DEPLOYMENT DIAGRAM

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

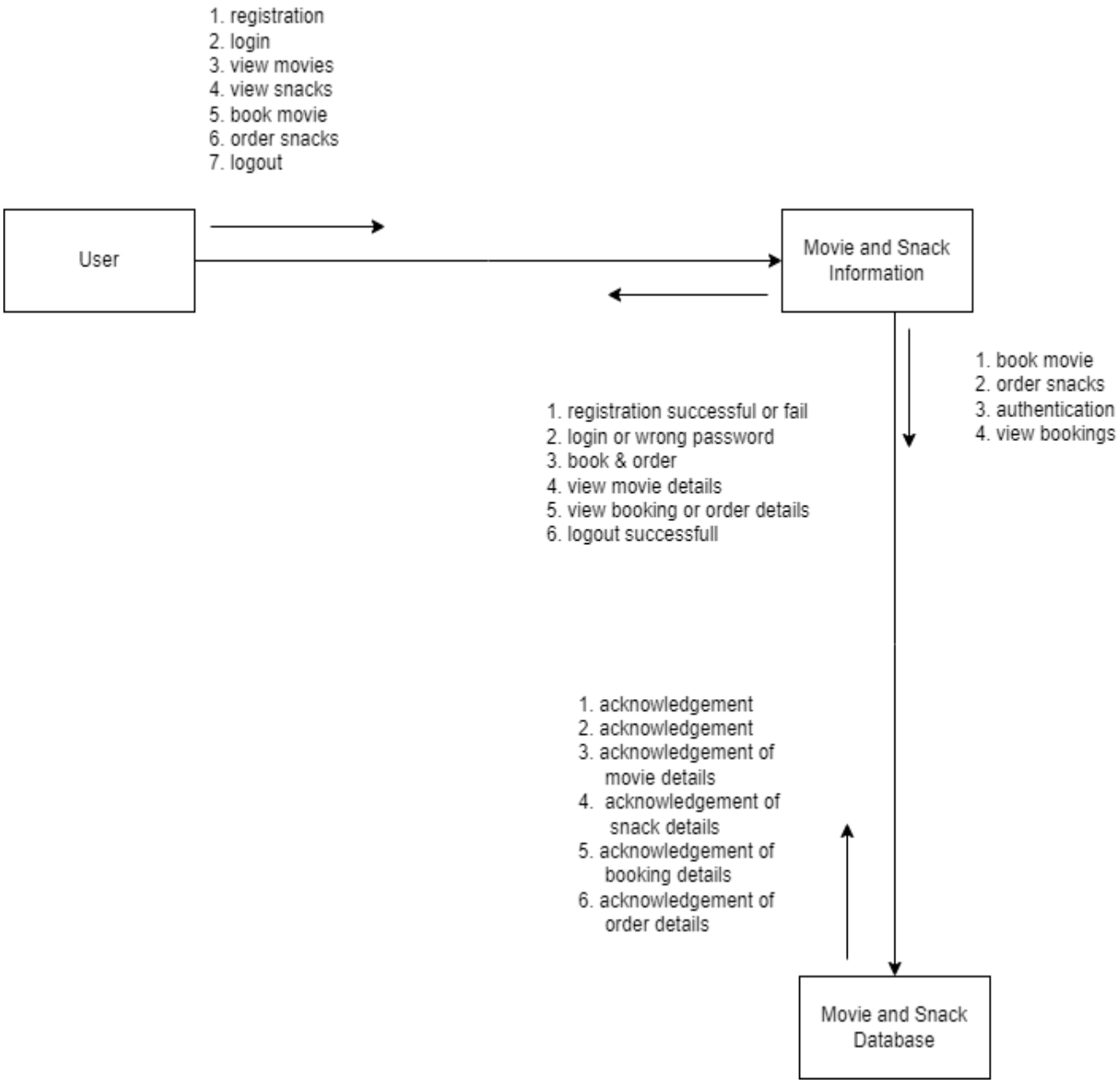
Fig 1: Deployment diagram for Online Theatre Ticket Booking System



4.2.6 COLLABORATION DIAGRAM

The Collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.
Following are the components of a component diagram that are enlisted below:

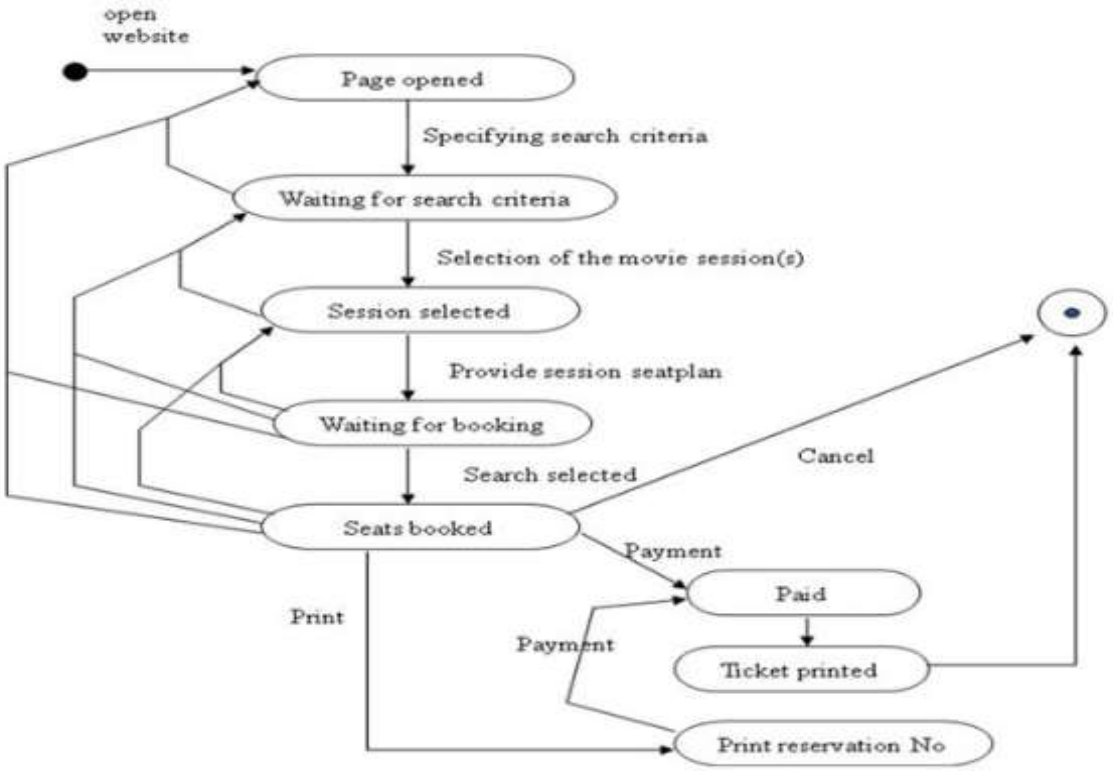
- Objects: The representation of an object is done by an object symbol with its name and class underlined, separated by a colon. In the collaboration diagram, objects are utilized in the following ways:
    - The object is represented by specifying their name and class.
    - It is not mandatory for every class to appear.
    - A class may constitute more than one object.
    - In the collaboration diagram, firstly, the object is created, and then its class is specified.
    - To differentiate one object from another object, it is necessary to name them.
- Actors: In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.
- Links: The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.
- Messages: It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.


1. registration
2. login
3. view movies
4. view snacks
5. book movie
6. order snacks
7. logout
User
Movie and Snack Information
1. book movie
2. order snacks
3. authentication
4. view bookings
1. registration successful or fail
2. login or wrong password
3. book & order
4. view movie details
5. view booking or order details
6. logout successfull
1. acknowledgement
2. acknowledgement
3. acknowledgement of movie details
4. acknowledgement of snack details
5. acknowledgement of booking details
6. acknowledgement of order details
Movie and Snack Database

4.2.7 STATE CHART DIAGRAM

A State Chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system. Following are the main purposes of using Statechart diagrams –

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

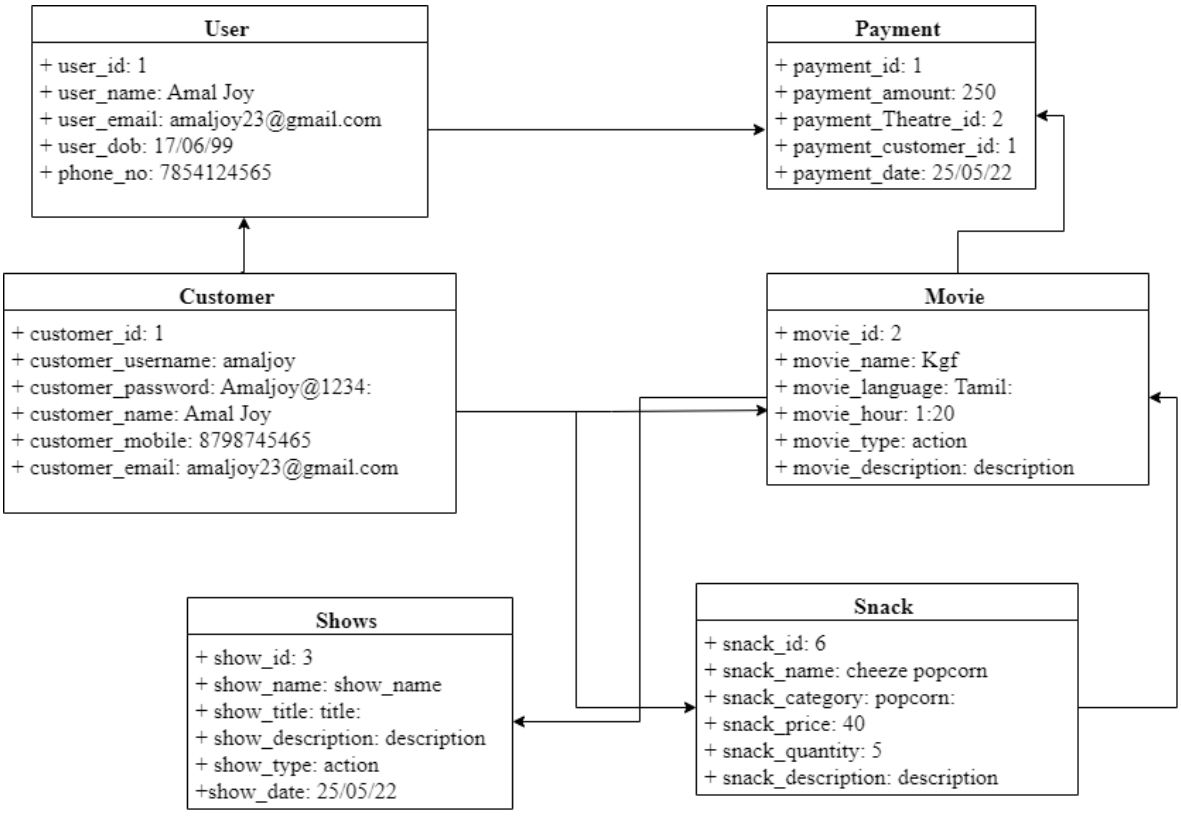


4.2.8 OBJECT DIAGRAM

Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

The purpose of a diagram should be understood clearly to implement it practically. The purposes of object diagrams are similar to class diagrams. The difference is that a class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concrete in nature. It means the object diagram is closer to the actual system behavior. The purpose is to capture the static view of a system at a particular moment. The purpose of the object diagram can be summarized as –

- Forward and reverse engineering.
- Object relationships of a system Static view of an interaction.
- Understand object behavior and their relationship from practical perspective

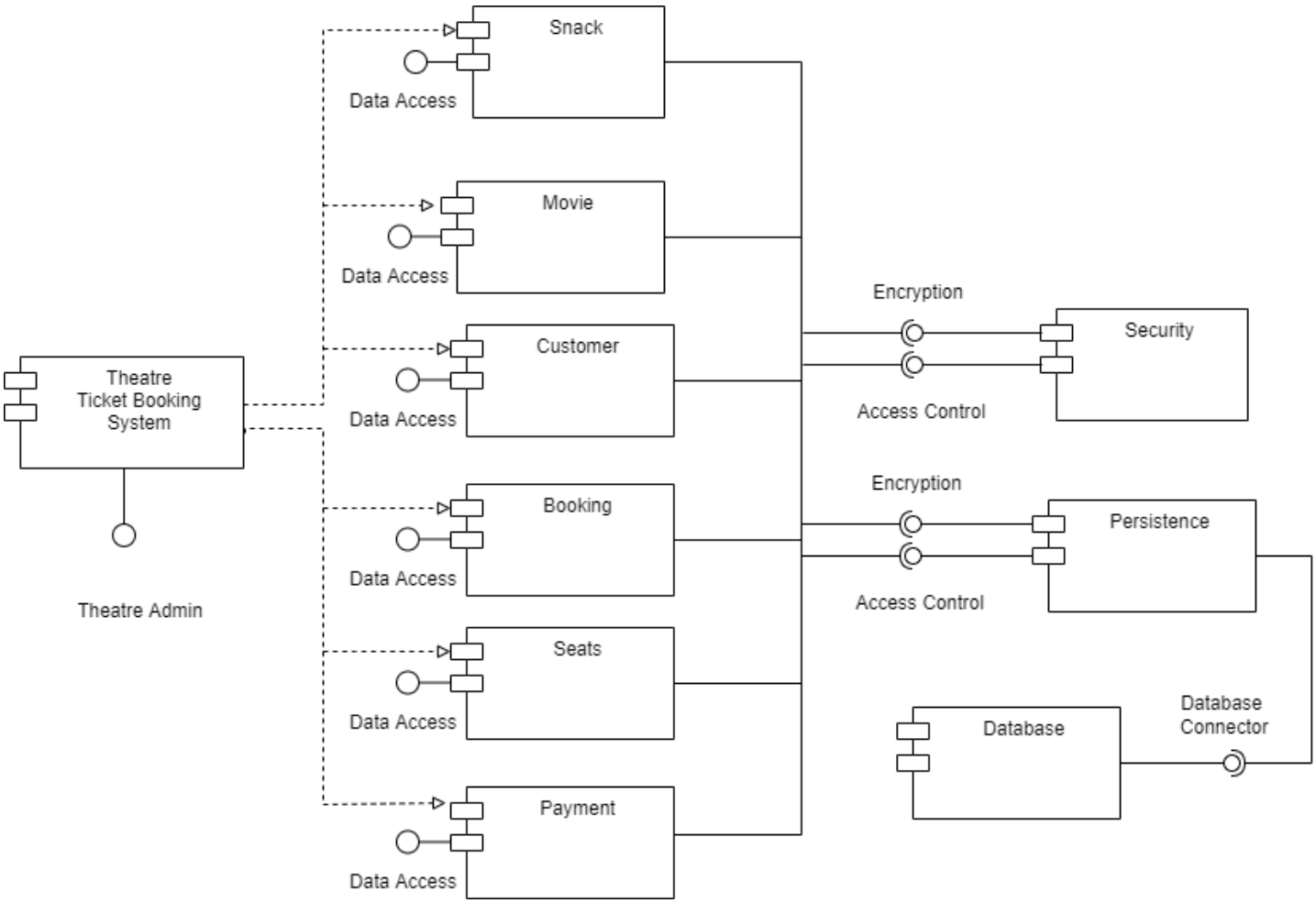


4.2.9 COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole. The purpose of the component diagram can be summarized as –

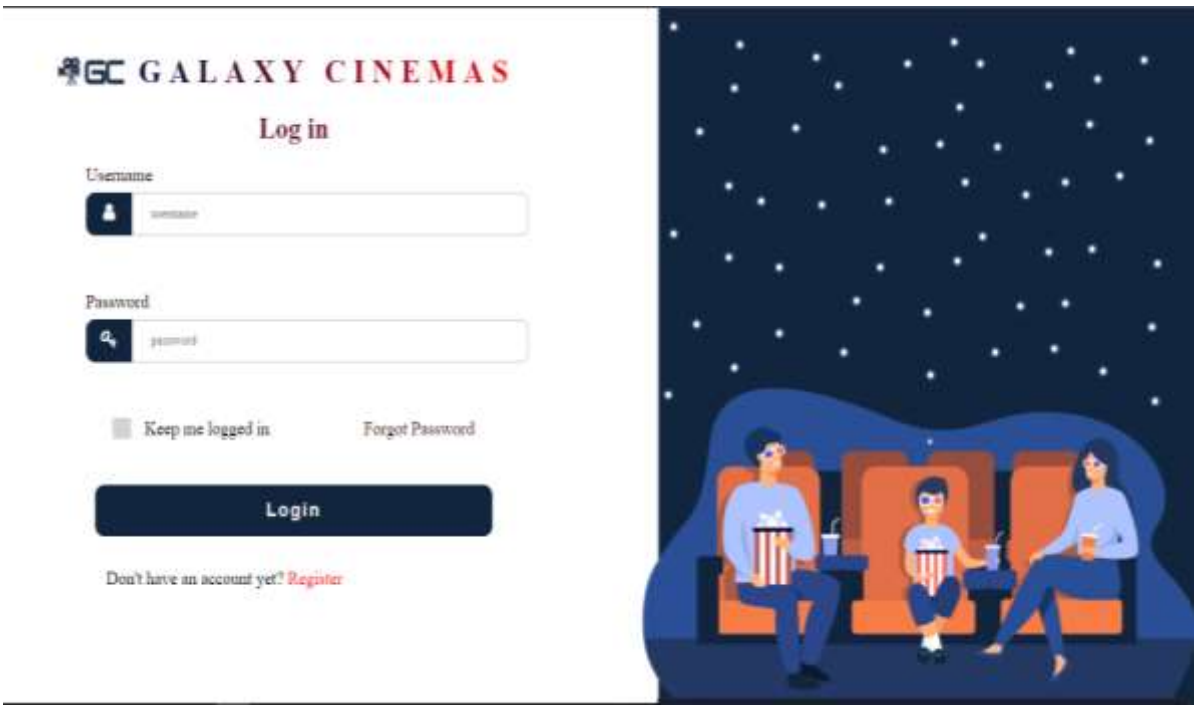
- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components.


Snack
Data Access
Movie
Data Access
Customer
Data Access
Booking
Data Access
Seats
Data Access
Payment
Data Access
Theatre
Ticket Booking
System
Theatre Admin
Encryption
Security
Access Control
Encryption
Persistence
Access Control
Database
Database
Connector

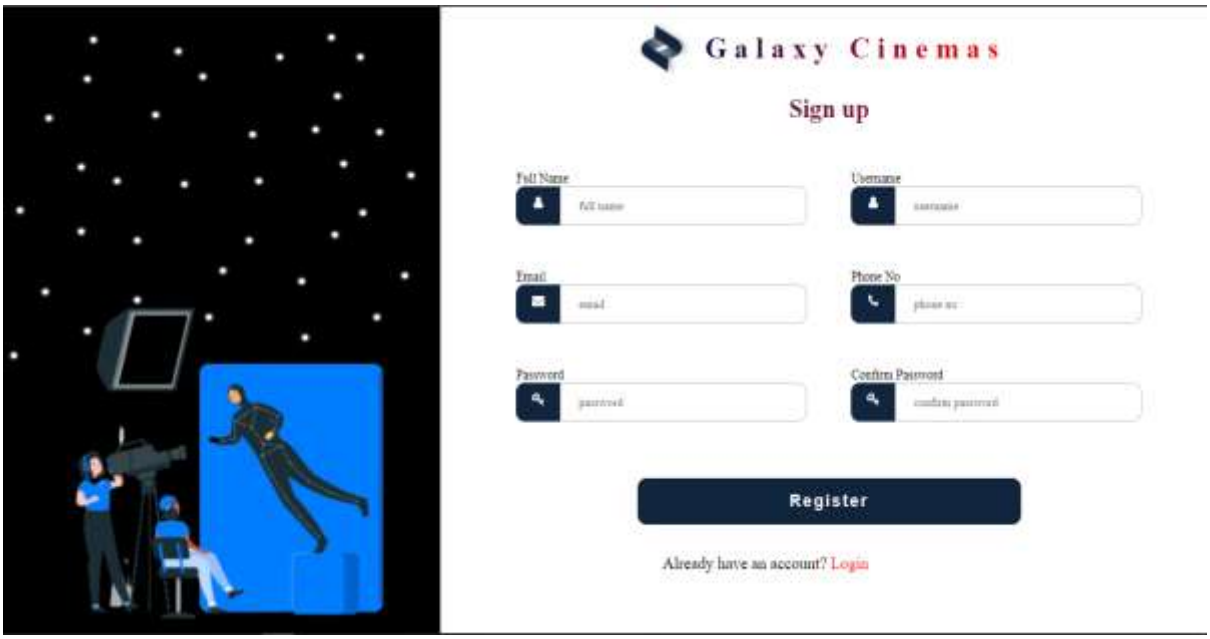
4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1 OUTPUT DESIGN

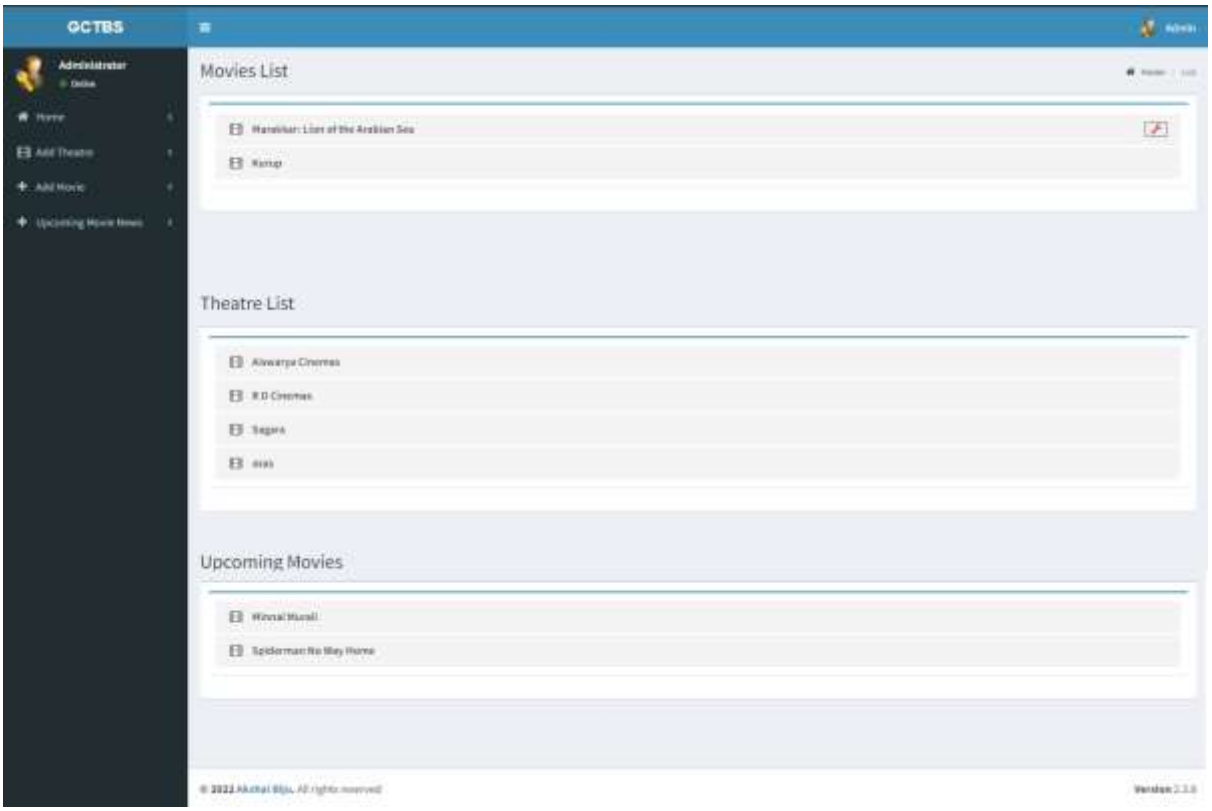
User Login

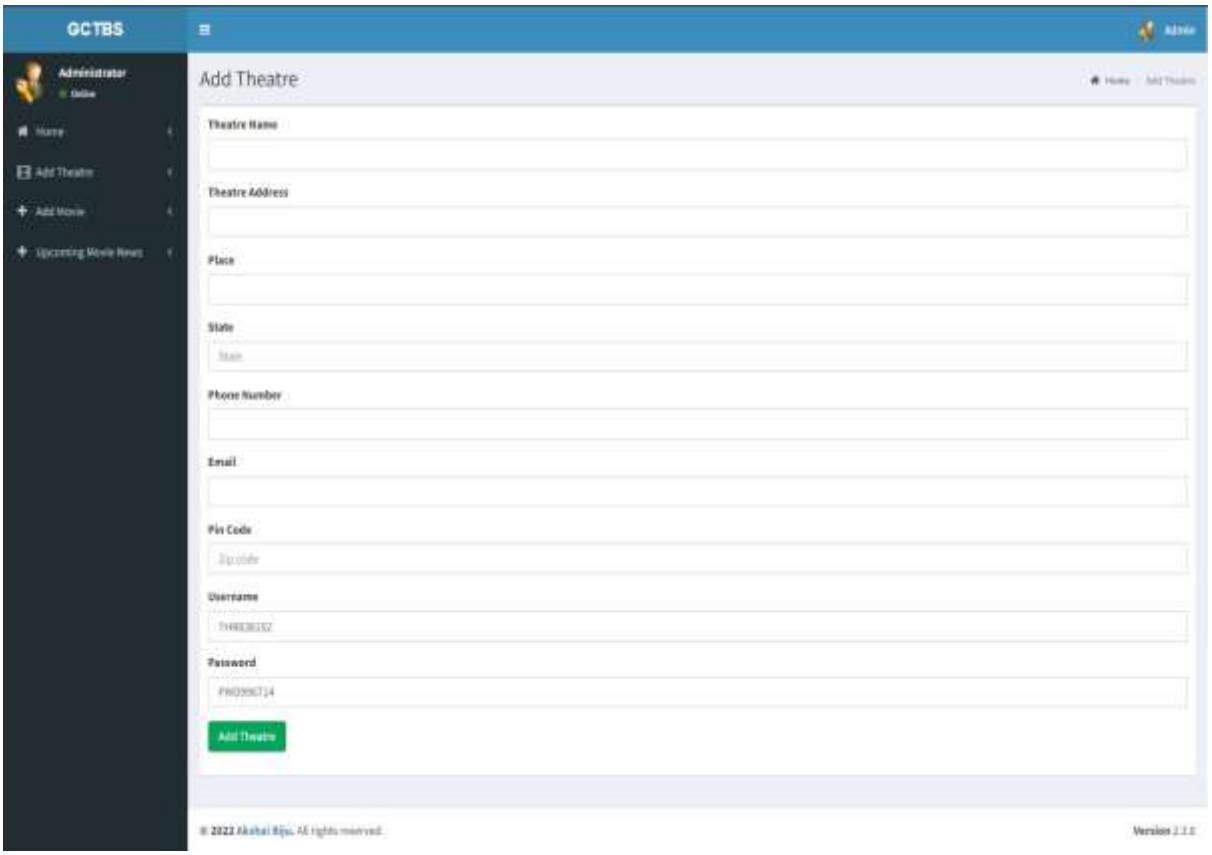


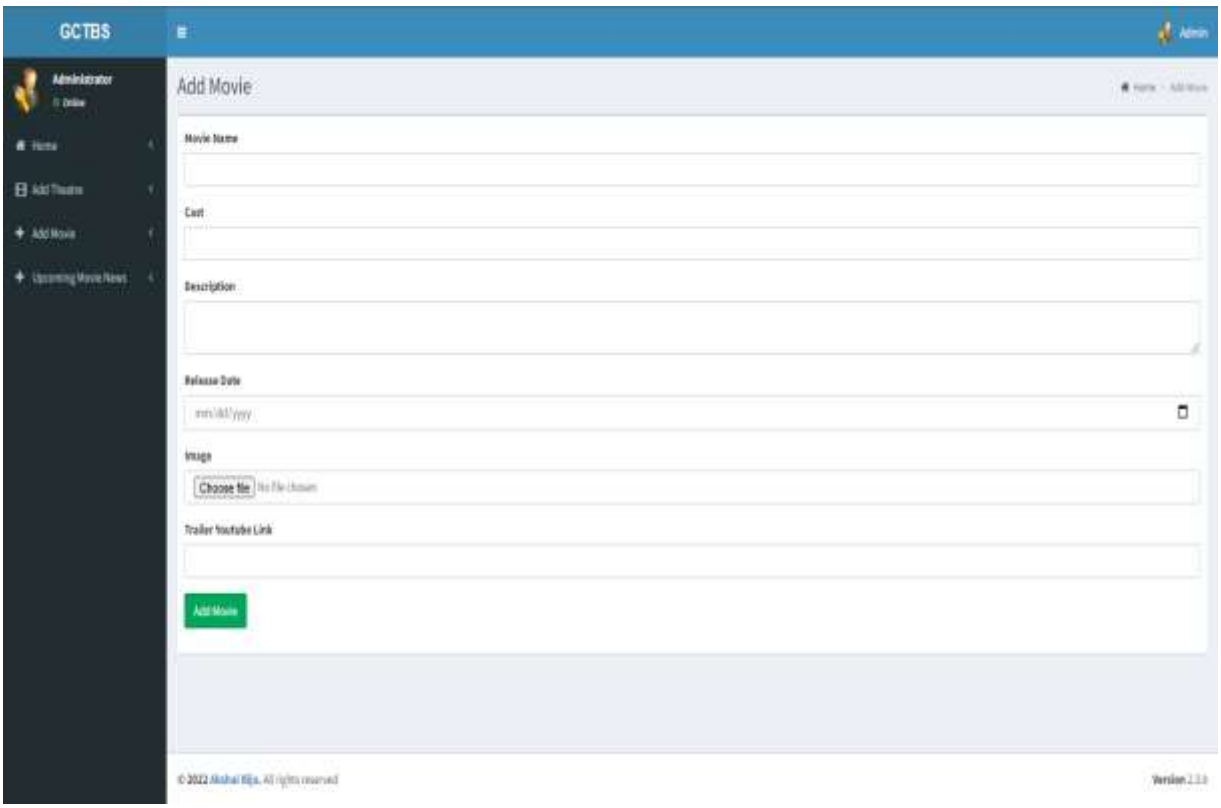
User Registration



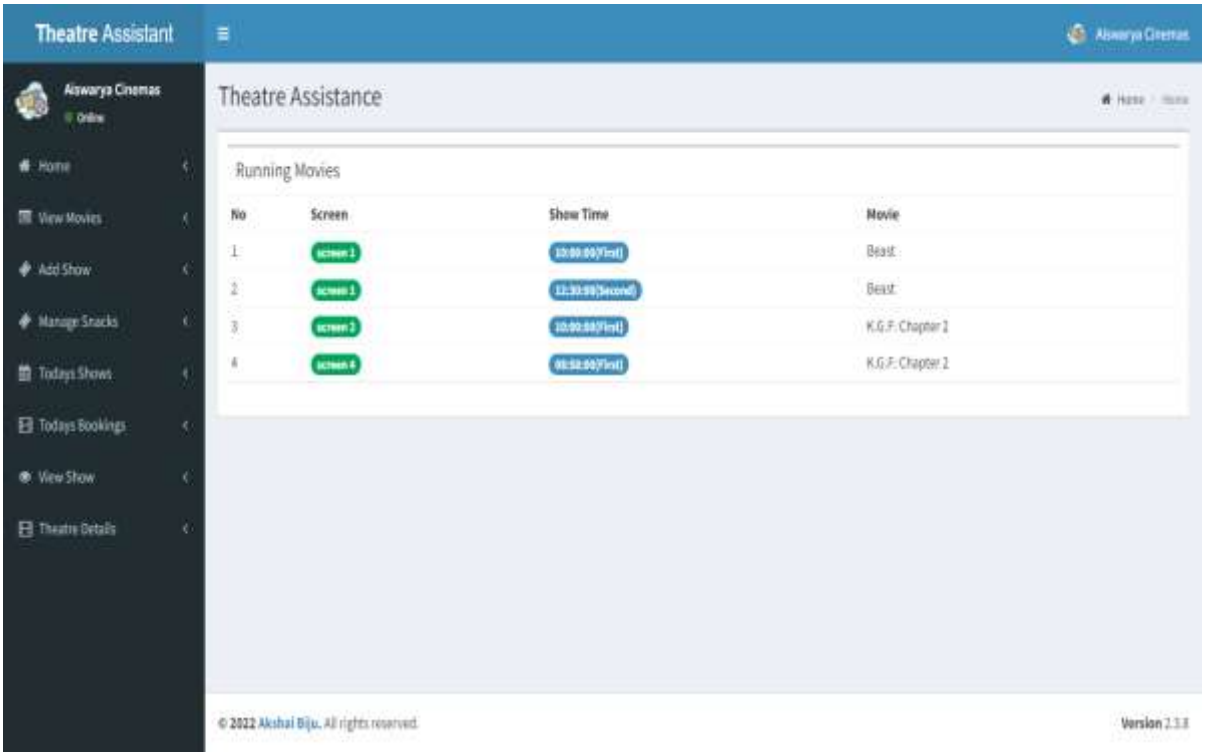
4.3.1 ADMIN PAGES

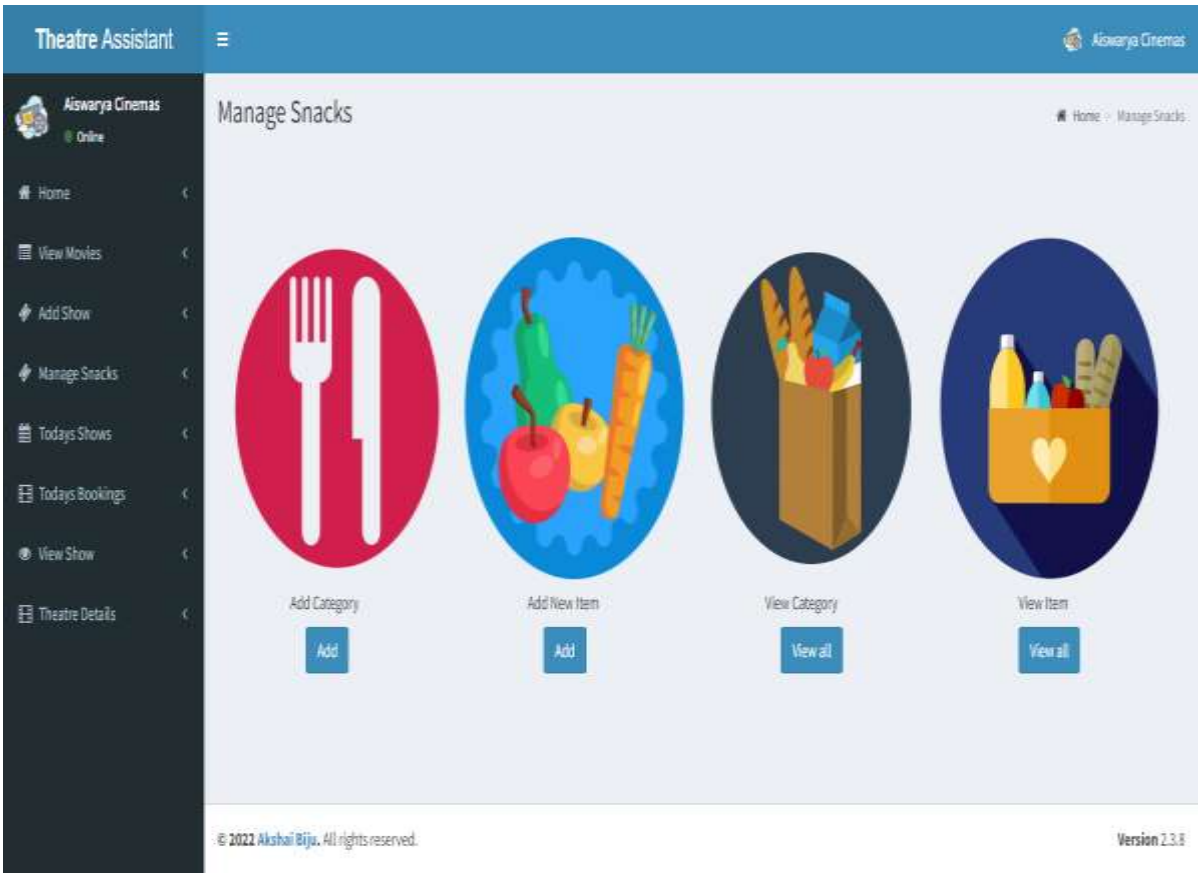


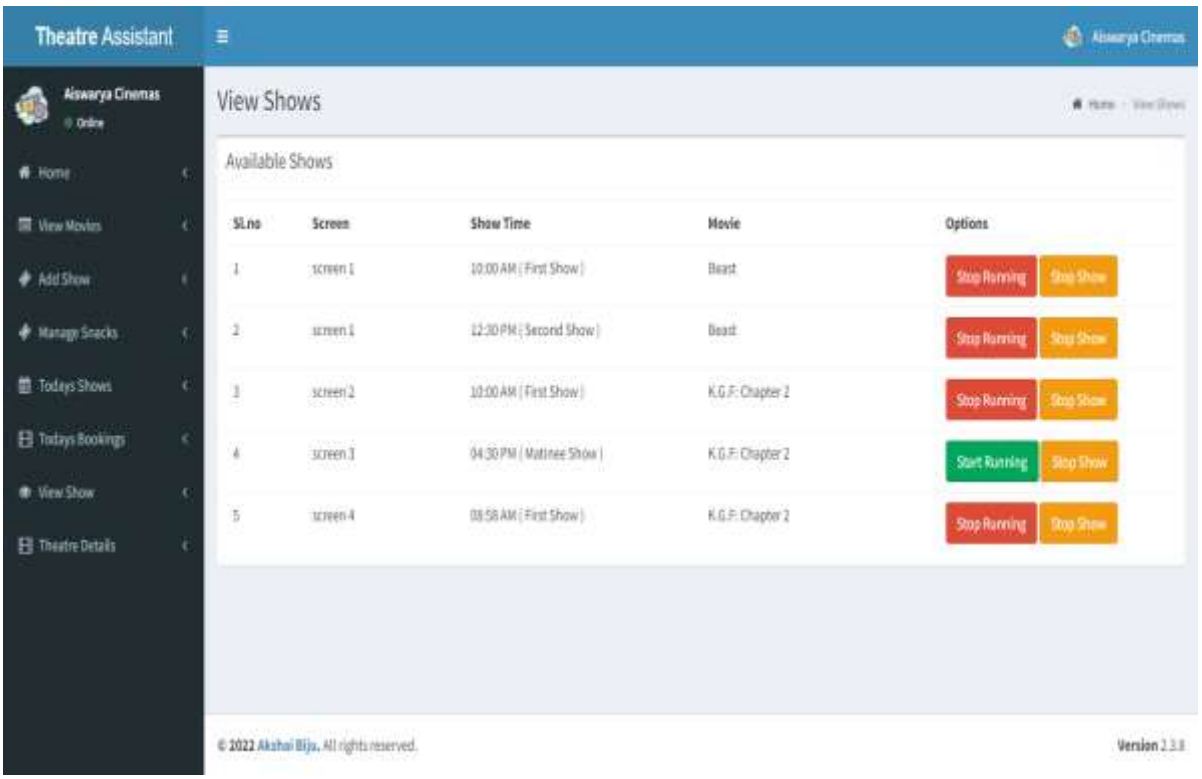




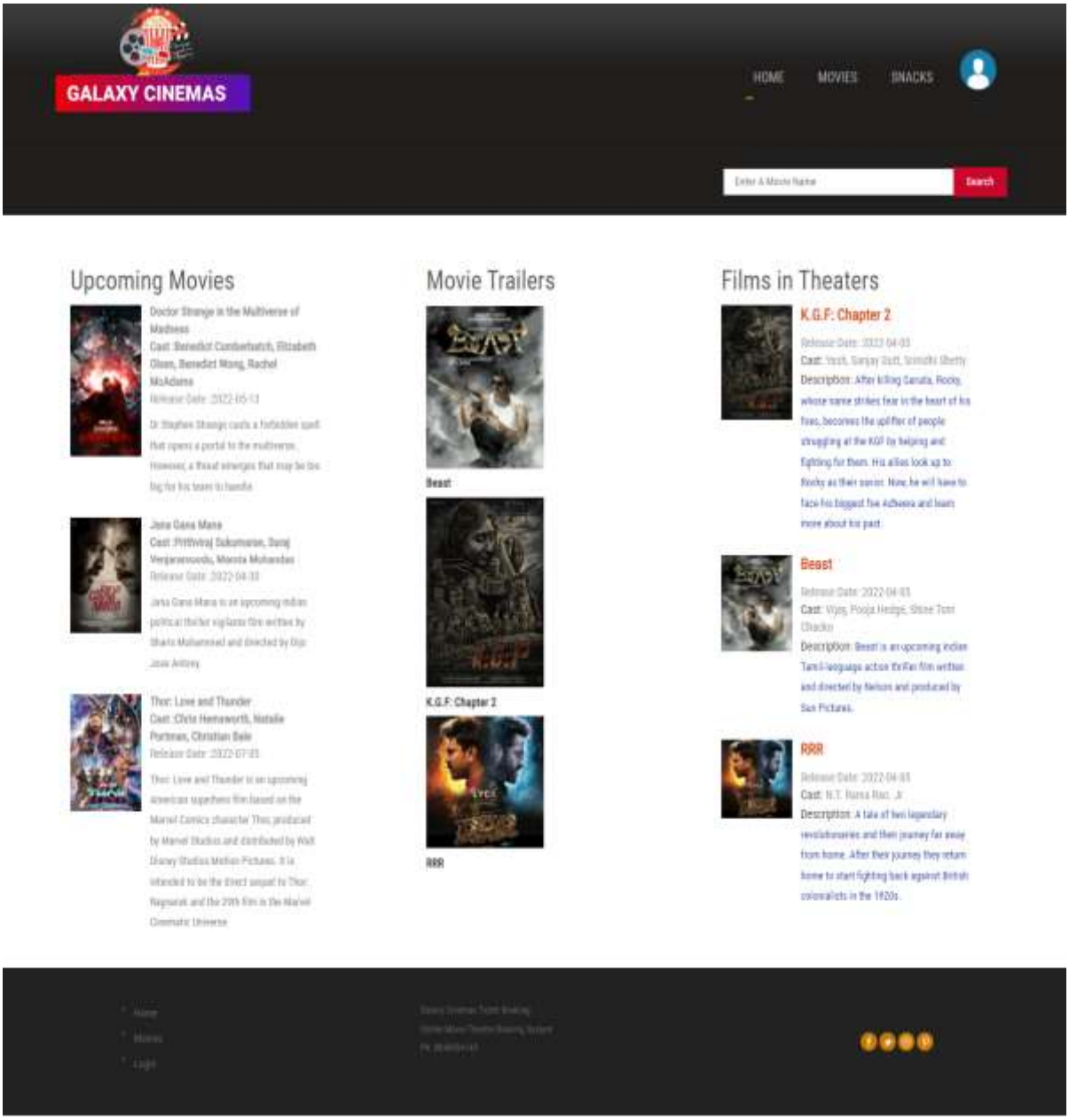
4.3.2 THEATRE ADMIN PAGES

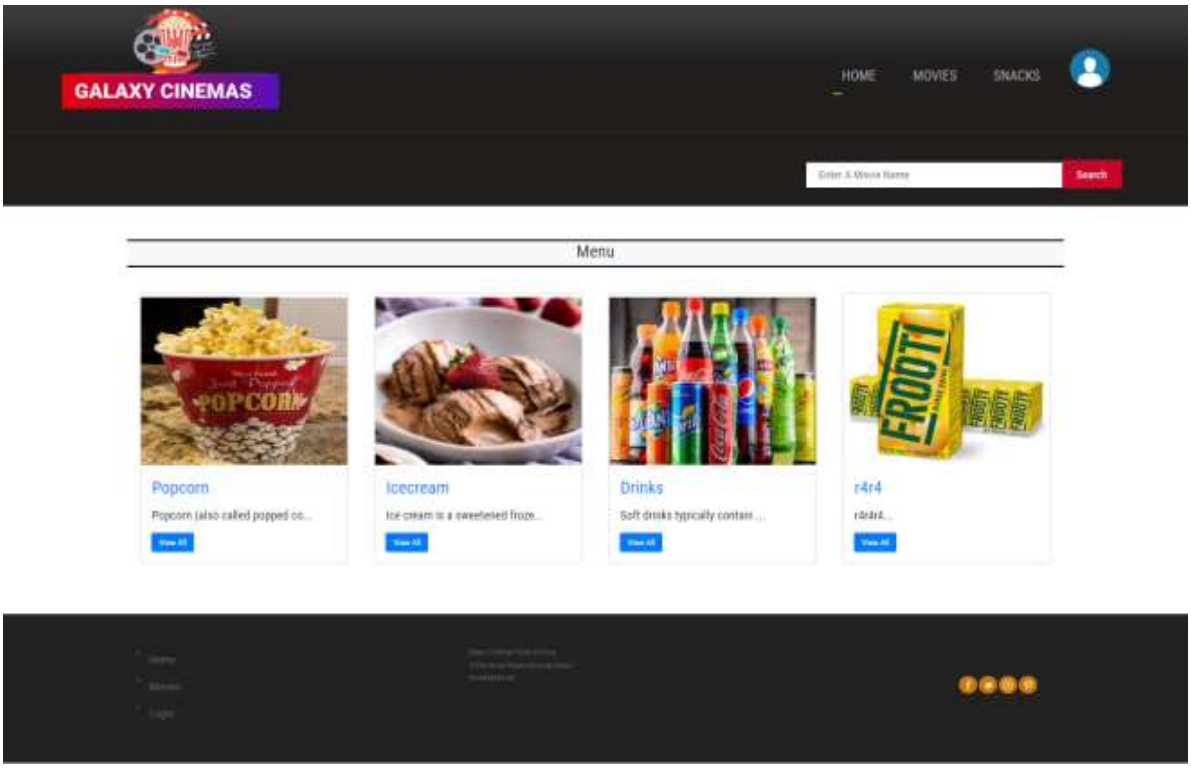


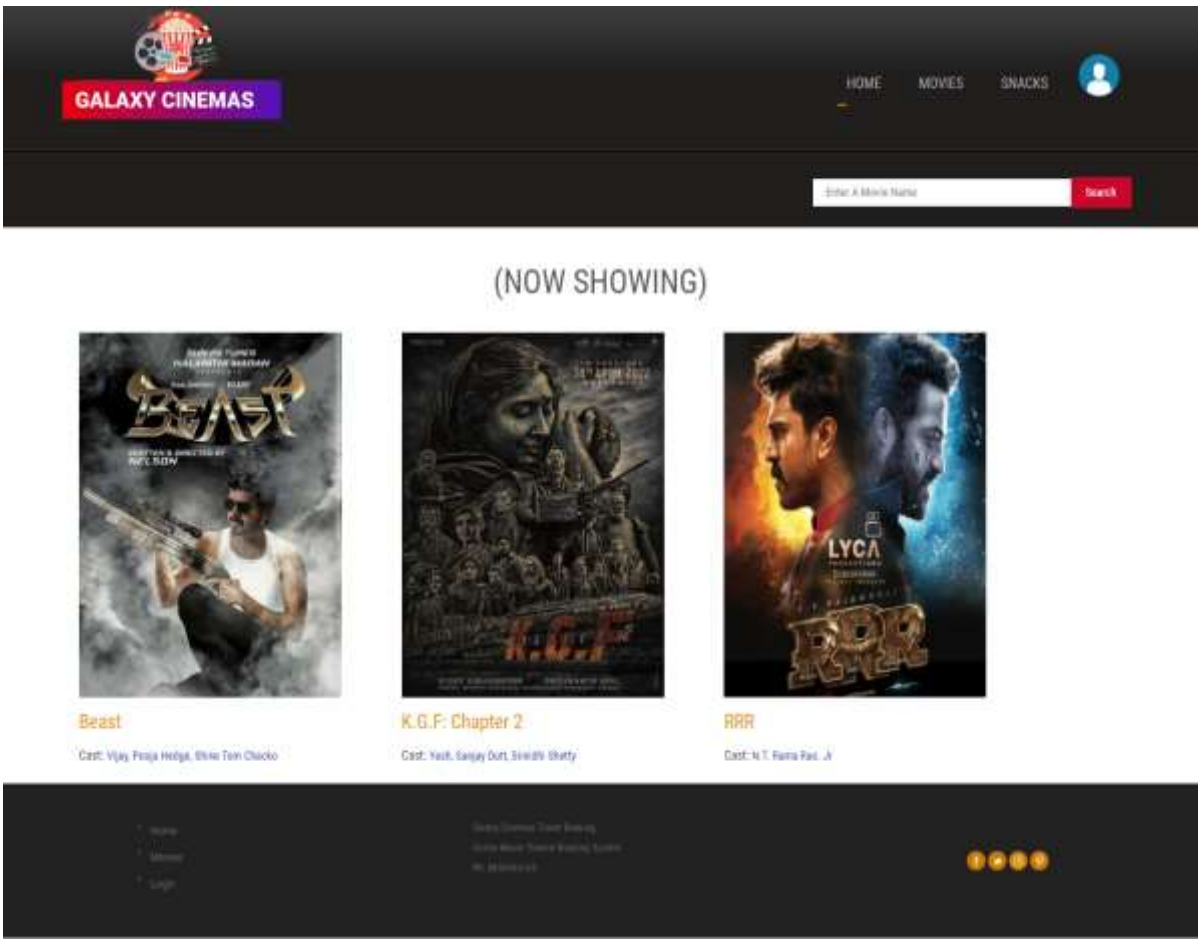

Theatre Assistant
Aiswarya Cinemas
Online
Home
View Movies
Add Show
Manage Snacks
Todays Shows
Todays Bookings
View Show
Theatre Details
Manage Snacks
Add Category
Add
Add New Item
Add
View Category
View all
View Item
View all
© 2022 Akshai Biju. All rights reserved.
Version 2.3.8

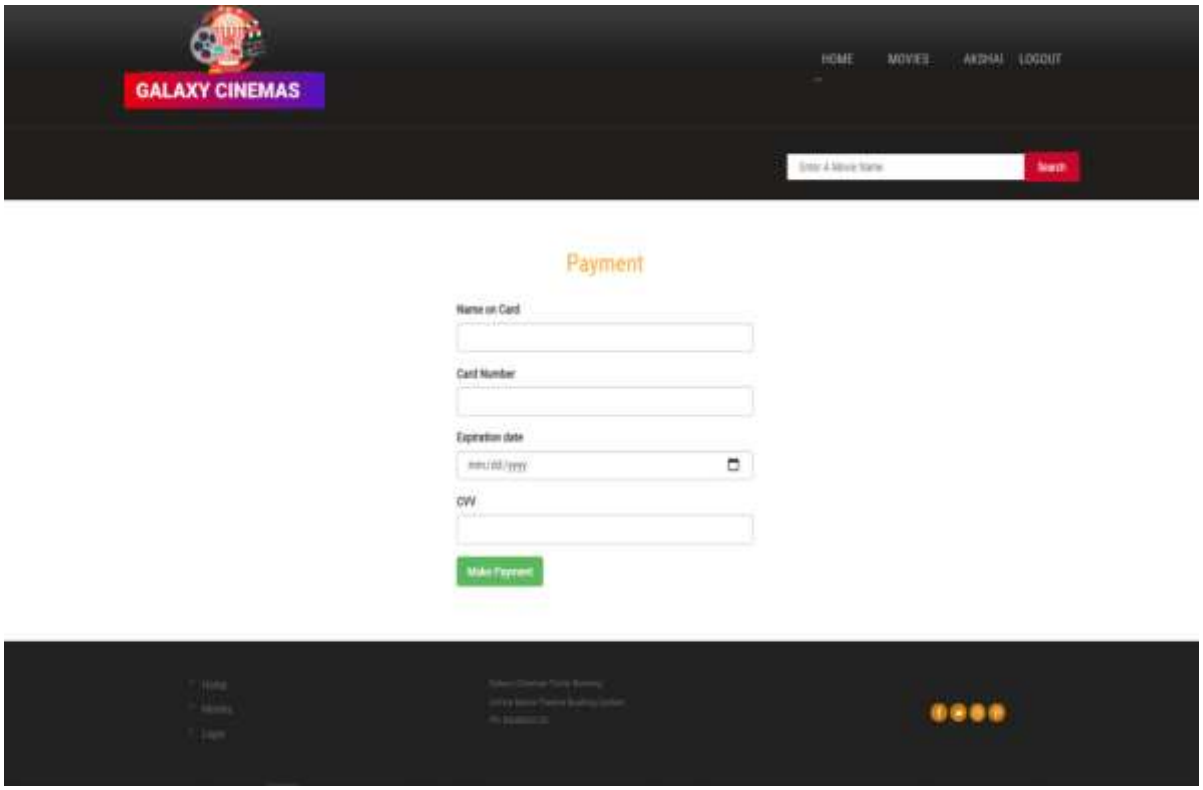

Theatre Assistant
Aiswarya Cinemas
Online
Home
View Movies
Add Show
Manage Snacks
Todays Shows
Todays Bookings
View Show
Theatre Details
View Shows
Available Shows
Sl.no
Screen
Show Time
Movie
Options
1
screen 1
10:00 AM (First Show)
Beast
Stop Running
Stop Show
2
screen 1
12:30 PM (Second Show)
Beast
Stop Running
Stop Show
3
screen 2
10:00 AM (First Show)
K.G.F: Chapter 2
Stop Running
Stop Show
4
screen 3
04:30 PM (Matinee Show)
K.G.F: Chapter 2
Start Running
Stop Show
5
screen 4
08:58 AM (First Show)
K.G.F: Chapter 2
Stop Running
Stop Show
© 2022 Akshai Biju. All rights reserved.
Version 2.3.8

4.3.2 USER PAGES




GALAXY CINEMAS
HOME
MOVIES
SNACKS
Search
Menu
POPCORN
FROOTI
Popcorn
Popcorn (also called popped co...
Icecream
Ice cream is a sweetened froze...
Drinks
Soft drinks typically contain ...
r4r4
r4r4r4...


GALAXY CINEMAS
HOME
MOVIES
SNACKS
Search
(NOW SHOWING)
BEAST
LYCA
RRR
Beast
K.G.F: Chapter 2
RRR


GALAXY CINEMAS
HOME
MOVIES
LOGOUT
Search
Payment
Name on Card
Card Number
Expiration date
CVV
Make Payment

4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.6.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attribute of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on another non-key attribute.

TABLE DESIGN

Table No : 01

Table Name : registration

Primary Key : uid

Foreign Key :

Table Description : To store user registration information

Fieldname	Data Type	Size	Description
uid	int	30	Primary key of the table
fname	varchar	30	First name of the user
uname	varchar	30	Email id of the user
email	varchar	30	Email id of the user
phno	bigint	100	Phone number of the user
pass	varchar	200	Password to login
status	varchar	20	Status of approval
isAdmin	int	5	Admin or not

Table No : 02

Table Name : tbl_booking

Primary Key : book_id

Foreign Key : screen_id

Table Description : To store booking information

Fieldname	Data Type	Size	Description
book_id	int	11	Primary key of the table
ticket_id	varchar	30	Ticket id of the user
t_id	int	11	Theatre id
user_id	int	11	User id
show_id	int	11	Show id
screen_id	int	11	Foreign key (references

			tbl_screens)
no_seats	int	3	Number of seats
seatRow	int	11	Seat position row
seatCol	Int	11	Seat position column
amount	int	5	Amount
ticket_date	date		Ticket date
date	date		date
status	int	1	Status of approval

Table No : 03

Table Name : tbl_login

Primary Key : id

Foreign Key : uid

Table Description : To store all login information

Fieldname	Data Type	Size	Description
id	int	11	Primary key of the table
uid	int	11	Foreign key (references registration)
username	varchar	50	User name
password	varchar	50	Password
user_type	int	1	User type

Table No : 04

Table Name : tbl_screens

Primary Key : screen_id

Foreign Key :

Table Description : To store Screen Informations

Fieldname	Data Type	Size	Description
screen_id	int	11	Primary key of the table
t_id	int	11	theatre id
screen_name	varchar	110	Screen name

Table No : 05

Table Name : tbl_show_time

Primary Key : st_id

Foreign Key : screen_id

Table Description : To store Informations about Show timing details

Fieldname	Data Type	Size	Description
st_id	int	11	Primary key of the table
screen_id	int	11	Foreign key (references tbl_screens)
name	varchar	40	Show name
start_time	time		Show start time

Table No : 06

Table Name : tbl_movie

Primary Key : movie_id

Foreign Key :

Table Description : To store movie details

Fieldname	Data Type	Size	Description
movie_id	int	11	Primary key of the table
t_id	varchar	11	Theatre id
movie_name	varchar	256	Name of the movie
cast	varchar	500	Cast of the movie
desc	varchar	500	Movie description
release_date	date		Movie releasing date
image	varchar	200	Image of the movie
video_url	varchar	200	Youtube Link of the Movie
status	int	1	Status of approval

Table No : 07

Table Name : tbl_news

Primary Key : news_id

Foreign Key :

Table Description : To store Upcoming Movie details

Fieldname	Data Type	Size	Description
news_id	int	11	Primary key of the table
name	varchar	100	Name of the movie
cast	varchar	100	Cast in the movie
news_date	date		Releasing date
description	varchar	800	Movie description
attachment	varchar	200	Movie image

Table No : 08

Table Name : tbl_shows

Primary Key : sid

Foreign Key : st_id, movie_id

Table Description : To store Informations of each Shows

Fieldname	Data Type	Size	Description
sid	int	11	Primary key of the table
st_id	int	11	Foreign key (references tbl_show_time)
theatre_id	int	11	Theatre id
movie_id	int	11	Foreign key (references tbl_movie)
start_date	date		Movie release date in theatre
status	int	11	Show available status
r_status	int	11	Status of approval

Table No : 09

Table Name : tbl_theatre

Primary Key : id

Foreign Key :

Table Description : To store Theatre Details

Fieldname	Data Type	Size	Description
Id	int	11	Primary key of the table
name	varchar	100	Name of the theatre
address	varchar	100	Address of the theatre
place	varchar	100	Theatre location
state	varchar	50	state
phno	bigint	100	Theatre owner phone no
mail	varchar	30	Email address
pin	int	11	pincode

Table No : 10

Table Name : tbl_categories

Primary Key : category_id

Foreign Key : t_id

Table Description : To store snack category

Fieldname	Data Type	Size	Description
category_id	int	12	Primary key of the table
category_name	varchar	255	Name of the Category
t_id	int	11	Theatre id
category_desc	text		Category Description
image	varchar	200	Image
category_createDate	datetime		Create Date

Table No : 11

Table Name : tbl_screentype

Primary Key : sty_id

Foreign Key : screen_id

Table Description : To store Seat Type

Fieldname	Data Type	Size	Description
screen_id	int	11	Foriegn key (references tbl_screens)
sty_id	int	11	Primary key of the table
type_name	varchar	100	Seat Type Name
position	int	11	Seat Type rank
scRow	varchar	100	Seat Row
scCol	varchar	100	Seat Column
seats	int	11	seats
charge	int	11	Amount

Table No : 12

Table Name : tbl_snackbook

Primary Key : order_id

Foreign Key :

Table Description : To store snack booking information

Fieldname	Data Type	Size	Description
order_id	int	11	Primary key of the table
snackId	int	11	Snack id
itemQuantity	int	100	No of Quantites
user_id	int	11	User id
amount	int	200	Price
orderDate	datetime		Date
status	int	20	status

Table No : 13

Table Name : tbl_snacks

Primary Key : snackId

Foreign Key :

Table Description : To store snacks details

Fieldname	Data Type	Size	Description
snackId	int	12	Primary key of the table
snackName	varchar	255	Snack Name
snackPrice	int	12	Price
snackDesc	text		Description
image	varchar	200	image
snackCategoryId	int	12	Category Id
snackPubDate	datetime	20	Publish Date

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

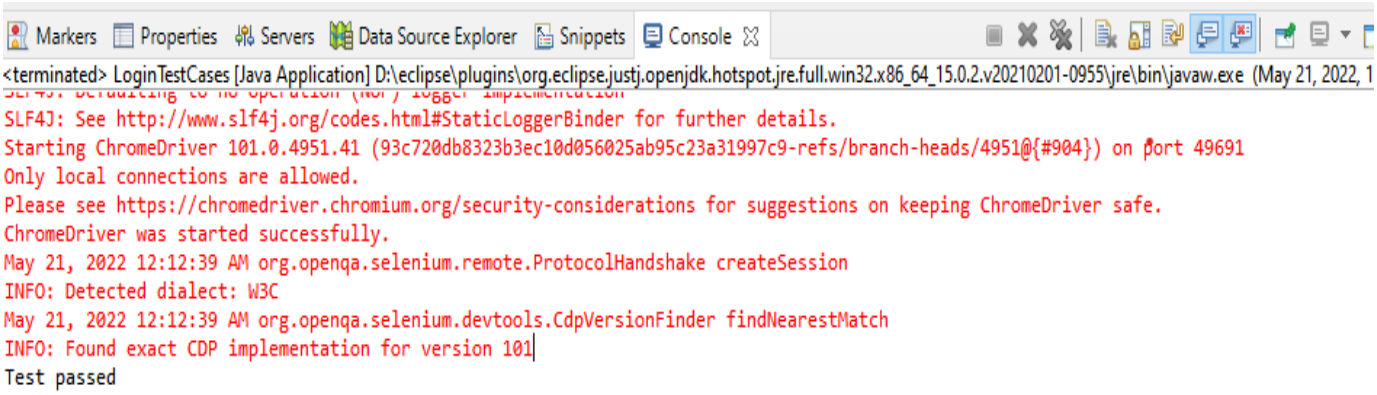
Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

5.2.1.1 Sample Test Cases

<table>
<tr><td colspan="6">Test Case 1</td></tr>
<tr><td colspan="6">Project Name: Online Theatre Ticket Booking System</td></tr>
<tr><td colspan="6">Login Test Case</td></tr>
<tr><td colspan="3">Test Case ID:login</td><td colspan="3">Test Designed By: AKSHAI BIJU</td></tr>
<tr><td colspan="3">Test Priority(Low/Medium/High):High</td><td colspan="3" rowspan="2">Test Designed Date: 18-05-2022</td></tr>
<tr><td colspan="3">Module Name: Login Screen</td></tr>
<tr><td colspan="3">Test Title: Verify login with valid username and password</td><td colspan="3">Test Execution Date: 19-05-2022</td></tr>
<tr><td colspan="3">Description: Test the Login Page</td><td colspan="3"></td></tr>
<tr><td colspan="6">Pre-Condition: User has valid user name and password</td></tr>
<tr><td>Test case steps</td><td>Test Step</td><td>Test Data</td><td>Expected Result</td><td>Actual Result</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1</td><td>Navigation to Login Page</td><td></td><td>Login Page should be</td><td>Login page displayed</td><td>Pass</td></tr>
<tr><td rowspan="3">2</td><td>Provide Valid User name</td><td>User Name: Mrudul</td><td rowspan="3">User should be able to Login</td><td rowspan="3">User Logged in and navigated to Admin Dashboard with records</td><td rowspan="3">Pass</td></tr>
<tr><td>Provide Valid Password</td><td>Password: Mrudul@123</td></tr>
<tr><td>Click on Login button</td><td></td></tr>
<tr><td rowspan="3">3</td><td>Provide Invalid User Name or password</td><td>User Name: mrudul12
Password: 1234</td><td rowspan="3">User should not be able to Login</td><td rowspan="3">Message for enter valid user or password displayed</td><td rowspan="3">Pass</td></tr>
<tr><td>Provide Null User Name or Password</td><td>Username: null Password: null</td></tr>
<tr><td>Click on Login button</td><td></td></tr>
<tr><td colspan="6">Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database</td></tr>
</table>

Code

```
package testCases;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.By;

import browserImplementation.DriverSetup;

public class LoginTestCases {
static WebDriver driver;
public static void main(String[] args) {
driver = DriverSetup.getWebDriver("http://localhost/GalaxyCinemas/Login.php");
driver.findElement(By.name("un")).sendKeys("Mrudul");
driver.findElement(By.name("psw")).sendKeys("Mrudul@123");
driver.findElement(By.name("loginbt")).click();

String actualUrl="http://localhost/GalaxyCinemas/index.php";

String expectedUrl= driver.getCurrentUrl();

if(actualUrl.equalsIgnoreCase(expectedUrl)) {

System.out.println("Test passed");

} else {

System.out.println("Test failed");

}

}

}
```



Test Case 2

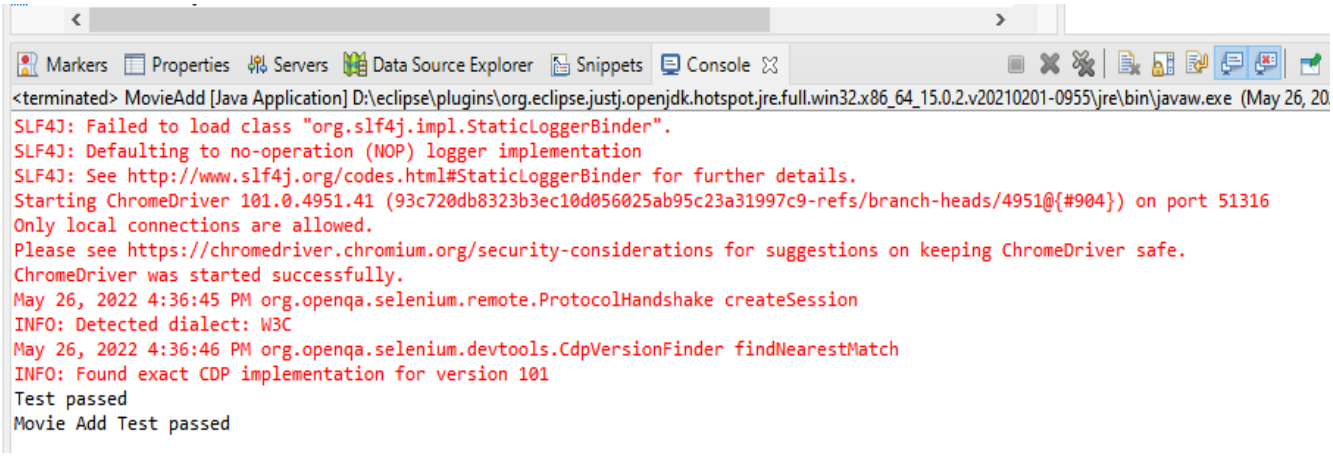
Project Name: Online Theatre Ticket Booking System					
Movie add Test Case					
Test Case ID: add movie			**Test Designed By:** AKSHAI BIJU		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 18-05-2022		
Module Name: add movie					
Test Title : Add Movie			**Test Execution Date:** 19-05-2022		
Description: Login to system and add movie and its information, if some error occurs, test will fail					
Pre-Condition: Admin has valid user name and password					
Test Case Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Login Page		Login Page should be	Login page displayed	Pass
2	Provide Valid Username	User Name: Admin	Admin should be able to Login	Admin Logged in and navigated to Admin Dashboard with records	Pass
	Provide Valid Password	Password: admin123			
	Click on Login button				
3	Provide Movie information's	Input movie details	Admin will be redirected to dashboard	Admin will be redirected to dashboard	Pass
	Click on Add Movie button				
4	Provide invalid information's	Input invalid movie details	Admin will be redirected to dashboard	Admin will be stay on that page showing error message	Pass
	Click on Add Movie button				
Post-Condition: Admin is validated with database and successfully login into account. The Account session details are logged in database					

Code

```
package testCases;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.By;

import browserImplementation.DriverSetup;

public class MovieAdd {
```

```
 static WebDriver driver;
 public static void main(String[] args) {
 driver = DriverSetup.getWebDriver("http://localhost/GalaxyCinemas/admin/");
driver.findElement(By.name("Email")).sendKeys("Admin");
driver.findElement(By.name("Password")).sendKeys("admin123");
driver.findElement(By.name("loginbt")).click();
String actualUrl="http:// localhost/GalaxyCinemas/admin/";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed");
} else {
System.out.println("Test failed");
}


  driver.get("http://localhost/GalaxyCinemas/admin/pages/add_movie.php");
  driver.findElement(By.name("name")).sendKeys("12th Man");
  driver.findElement(By.name("cast")).sendKeys("Mohanlal,Anu Sithara,Anusree");
  driver.findElement(By.name("desc")).sendKeys("12th Man is a 2022 Indian Malayalam-
language Mystery thriller film");
  driver.findElement(By.name("rdate")).sendKeys("05/28/2022");
  driver.findElement(By.name("image")).sendKeys("C:\\Users\\aksha\\Downloads\\img1.jpg");
driver.findElement(By.name("video")).sendKeys("https://www.youtube.com/watch?v=V81jMFr
awAk");
  driver.findElement(By.name("loginbt")).click();
if(driver.findElement(By.id("hideMe")).isDisplayed()) {
System.out.println("Test passed");
 } else {
```

```
System.out.println("Test failed");

} }

}
```



5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions thatwill fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system. Their confidence
- in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the

software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

This project is developed successfully and the performance is found to be satisfactory. This project is designed to meet the requirements of assigning jobs. It has been developed in PHP and the database has been built in My SQL server keeping in mind the specifications of the system.

The user will be able to book the ticket using this website. The relationship between company admin, Theatre Owner, and customer satisfies a good communication to complete ticketing process.

We have designed the project to provide the user with easy retrieval of data, details of theatre and necessary feedback as much as possible. In this project, the user is provided with a website that can be used to book movie tickets online. To implement this as a web application we used php as the technology. Php has advantages such as enhanced performance, scalability, built- in security and simplicity. To build any web application using PHP we need a programming language such as PHP and so on. MySQL was used as back-end database since it is one of the most popular open-source databases, and it provides fast data access, easy installation and simplicity. For front end we used HTML and CSS.

7.2 FUTURE SCOPE

- Snacks will be served to the user's seats
- Parking System
- Notification and alert about New Movies

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Complete reference for java by Simon Robinson.
- Headfirst Jsp & Servlet.
- Learn Programming in java by Lakhanpal Publishers.
- Database programming with JDBC and java by O'Reilly.
- Java and Software Design Concepts by Apress.
- Google for problem solving

WEBSITES:

- https://www.w3schools.com/js/
- https://www.w3schools.com/jquery/default.asp
- https://www.jquery.com/
- https://www.tutorialspoint.com/mysql/
- http://www.udemy/webdevelopercourse.com
- http://www.HTML.net
- http://www.udemy/webdevelopercourse.com

CHAPTER 9

APPENDIX

9.1 Sample Code

Index Page

index.php

```
<html>
<body>
<?php
include('header.php');
?>

<div class="content">
<div class="wrap">
<div class="content-top">
<div class="listview_1_of_3 images_1_of_3">
<h2 style="color:#555;">Upcoming Movies</h2>
<?php
$qry3=mysqli_query($con,"SELECT * FROM tbl_news LIMIT 5");

while($n=mysqli_fetch_array($qry3))
{
?>
<div class="content-left">
<div class="listimg listimg_1_of_2">
<img src="admin/<?php echo $n['attachment'];?>">
</div>
<div class="text list_1_of_2">
<div class="extra-wrap">
<span style="text-color:#000" class="data"><strong><?php echo
$n['name'];?></strong><br>
<span style="text-color:#000" class="data"><strong>Cast :<?php echo
$n['cast'];?></strong><br>
<div class="data">Release Date :<?php echo $n['news_date'];?>
</div>
<span class="text-top"><?php echo $n['description'];?>
</span>
</div>
</div>
<div class="clear"></div>
</div>
<?php
}
?>
</div>
<div class="listview_1_of_3 images_1_of_3">
<h2 style="color:#555;">Movie Trailers</h2>
<div class="middle-list">
<?php
```

```
$qry4=mysqli_query($con,"SELECT * FROM tbl_movie ORDER BY rand() LIMIT 6");

while($nm=mysqli_fetch_array($qry4))
{
?>

<div class="listimg1">
<a target="_blank" href="<?php echo $nm['video_url'];?>"><img src="<?php echo
$nm['image'];?>" alt=""/></a>
<a target="_blank" href="<?php echo $nm['video_url'];?>" class="link" style="text-
decoration:none; font-size:14px;"><?php echo $nm['movie_name'];?></a>
</div>
<?php
}
?>
</div>
</div>
<?php include('movie_sidebar.php');?>
</div>
</div>
<?php include('footer.php');?>
</div>
<?php include('searchbar.php');?>
</body>
</html>
```

Booking Seats

booking.php

```
 <?php include('header.php');
 error_reporting(0);
 if(!isset($_SESSION['loginstat']))
 {
   header('location:Login.php');
 }
   $qry2=mysqli_query($con,"select * from tbl_movie where
 movie_id='".$_SESSION['movie']."'");


 $movie=mysqli_fetch_array($qry2);
   ?>
 <div class="content">
   <div class="wrap">
   <div class="content-top">
   <div class="section group">
   <div class="about span_1_of_2">
            <h3><?php echo $movie['movie_name']; ?></h3>
            <div class="about-top">
```

```
<div class="grid images_3_of_2">
        <img src="<?php echo $movie['image']; ?>" alt=""/>
  </div>
  <div class="desc span_3_of_2">
  <p class="p-link" style="font-size:15px"><b>Cast : </b><?php echo $movie['cast'];
?></p>
  <p class="p-link" style="font-size:15px"><b>Release Date : </b><?php echo date('d-M-
Y',strtotime($movie['release_date'])); ?></p>
  <p style="font-size:15px"><?php echo $movie['desc']; ?></p>
  <a href="<?php echo $movie['video_url']; ?>" target="_blank"
class="watch_but">Watch Trailer</a>
</div>
<div class="clear"></div>
</div>
  <table class="table table-hover table-bordered text-center">
          <?php
          $s=mysqli_query($con,"select * from tbl_shows where
s_id='".$_SESSION['show']."'");
          $shw=mysqli_fetch_array($s);


  $t=mysqli_query($con,"select * from tbl_theatre where id='".$shw['theatre_id']."'");

  $theatre=mysqli_fetch_array($t);
  ?>
  <tr>
  <td class="col-md-6">
  Theatre
  </td>
  <td>
  <?php echo $theatre['name'].", ".$theatre['place'];?>
  </td>
  </tr>
  <tr>
  <td>
  Screen
  </td>
  <td>
  <?php

  $ttm=mysqli_query($con,"select  * from tbl_show_time where st_id='".$shw['st_id']."'");

  $ttme=mysqli_fetch_array($ttm);

  $sn=mysqli_query($con,"select  * from tbl_screens where
screen_id='".$ttme['screen_id']."'");


  $screen=mysqli_fetch_array($sn);
```

```
echo $screen['screen_name'];

?>

</td>
</tr>
<tr>
<td>
Date
</td>
<td>
<?php

  if(isset($_GET['date']))
        {
        $date=$_GET['date'];
        }
        else
        {
        if($shw['start_date']>date('Y-m-d'))
              {
              $date=date('Y-m-d',strtotime($shw['start_date'] . "-1 days"));
              }
              else
              {
              $date=date('Y-m-d');
              }
              $_SESSION['dd']=$date;
              }
              ?>



<div class="col-md-12 text-center" style="padding-bottom:20px">
  <?php
 if($date>$_SESSION['dd']){?>
  <a href="booking.php?date=<?php echo date('Y-m-d',strtotime($date . "-1 days"));?>">
  <button class="btn btn-default"><i class="glyphicon glyphicon-chevron-
  left"></i></button></a>
  <?php } ?>
  <span style="cursor:default" class="btn btn-default"><?php echo date('d-M-
  Y',strtotime($date));?></span>

<?php if($date!=date('Y-m-d',strtotime($_SESSION['dd'] . "+4 days"))){?>
        a href="booking.php?date=<?php echo date('Y-m-d',strtotime($date . "+1
days"));?>">
<button class="btn btn-default"><i class="glyphicon glyphicon-chevron-
right"></i></button></a>
  <?php }
```

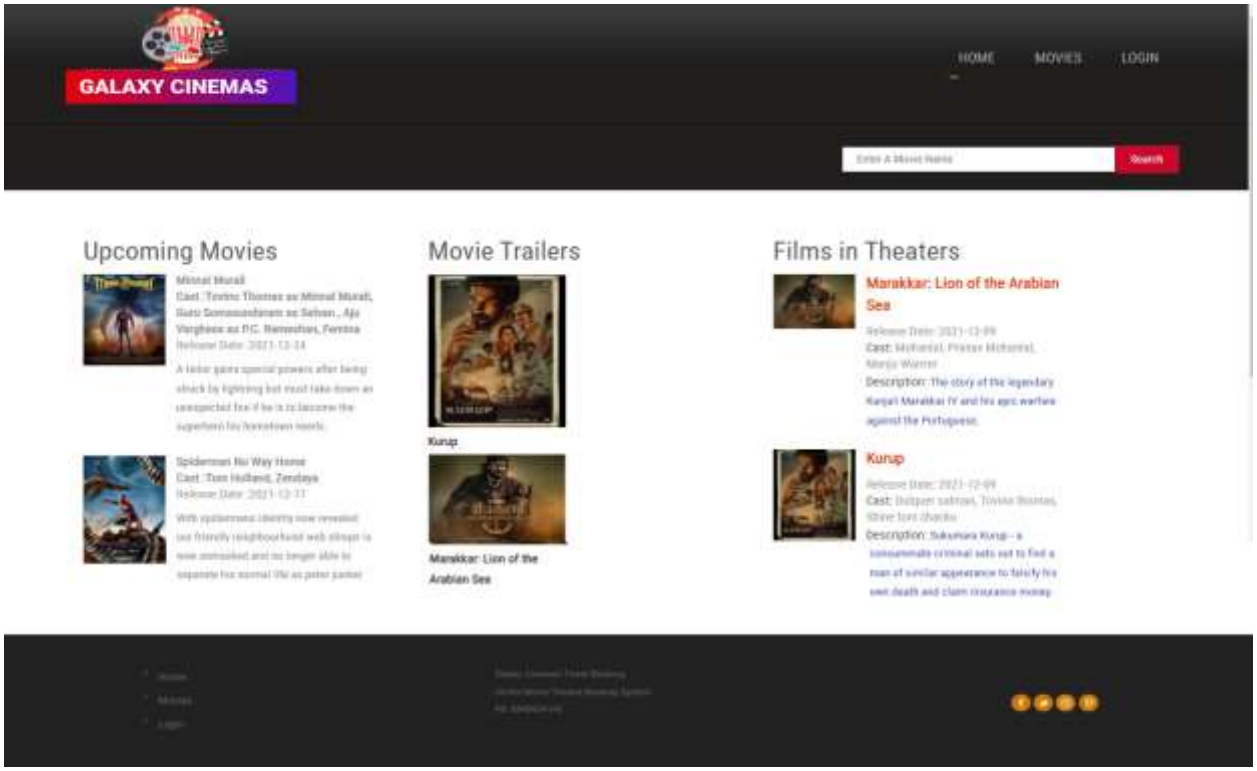
```
$av=mysqli_query($con,"select sum(no_seats) from tbl_bookings where
show_id='".$_SESSION['show']."' and ticket_date='$date'");
$avl=mysqli_fetch_array($av);
?>
</div>
  </td>
  </tr>
  <tr>
  <td>
  Available Seats
  <td>
  <td>
  <?php

  $seatcnt=mysqli_query($con,"select sum(no_seats) from tbl_bookings where
screen_id='".$_SESSION['screen']."' and show_id='".$_SESSION['show']."'" );

  $seatft=mysqli_fetch_array($seatcnt);

  echo $screen['seats']-$avl[0];
  ?>
  </td>
  </tr>
  <tr>
  <td>
  Show Time
  </td>
  <td>
  <?php echo date('h:i A',strtotime($ttme['start_time']))." ".$ttme['name'];?> Show
  </td>
  </tr>
  <tr>
  <td>
  Number of Seats
  </td>
  <td>
  <form  action="process_booking.php" method="post">
  <input type="hidden" name="screen" value="<?php echo $screen['screen_id'];?>"/>
 <input type="number" required tile="Number of Seats" max="<?php echo
$screen['seats']-$avl[0];?>" min="0" name="seats" class="form-control" value="1"
style="text-align:center" id="seats"/>
  <input type="hidden" name="amount" id="hm" value="<?php echo
$screen['charge'];?>"/>
  <input type="hidden" name="date" value="<?php echo $date;?>"/>
  </td>
</tr>
  <tr>
  <td>
  Amount
```

```
</td>
<td id="amount" style="font-weight:bold;font-size:18px">
Rs <?php echo $screen['charge'];?>
  </td>
  </tr>
  <tr>
<td colspan="2"><?php if($avl[0]==$screen['seats']){?><button type="button" class="btn
btn-danger" style="width:100%">House Full</button><?php } else { ?>
<button class="btn btn-info" style="width:100%">Book Now</button>
  <?php } ?>
          </form></td>
  </tr>
  <table>
  <tr>
  <td></td>
  </tr>
  </table>
  </div>
  <?php include('movie_sidebar.php');?>
  </div>
  <div class="clear"></div>
  </div>
  </div>


</div>


<?php include('footer.php');?>
<script type="text/javascript">
  $('#seats').change(function(){
          var charge=<?php echo $screen['charge'];?>;
          amount=charge*$(this).val();
          $('#amount').html("Rs "+amount);
          $('#hm').val(amount);
  });
</script>
```

9.2 Screen Shots

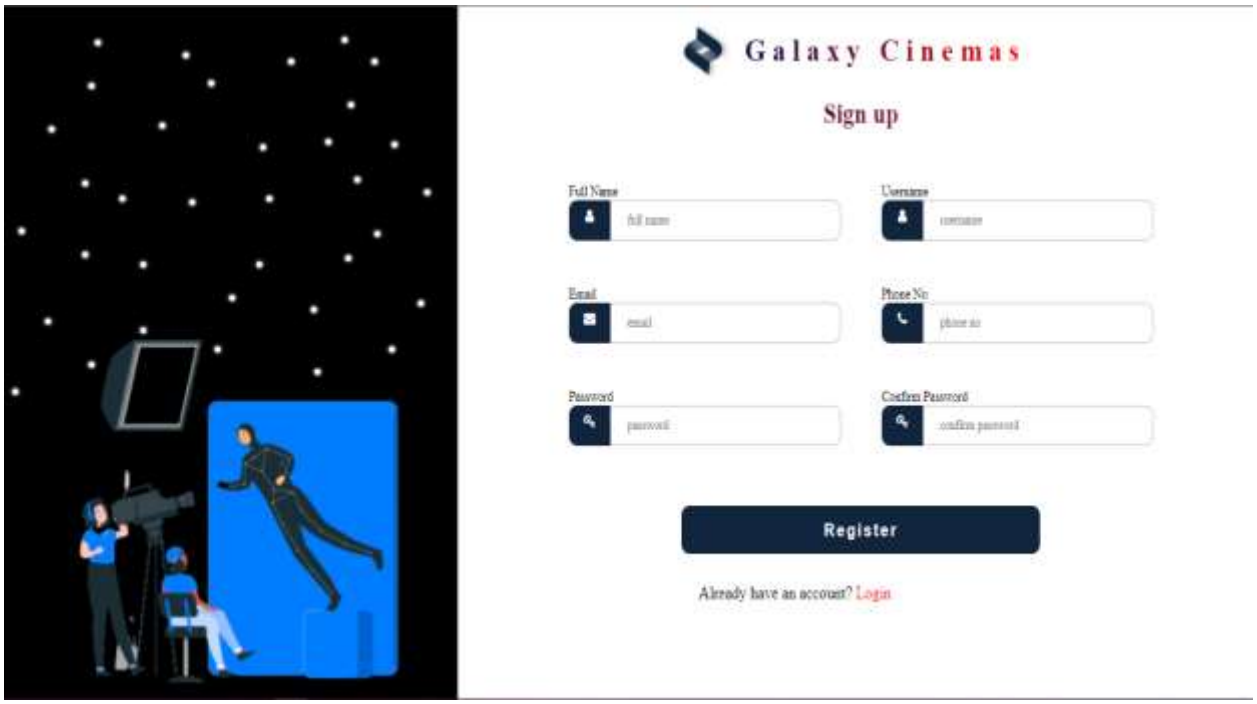
Common Page

Index Page

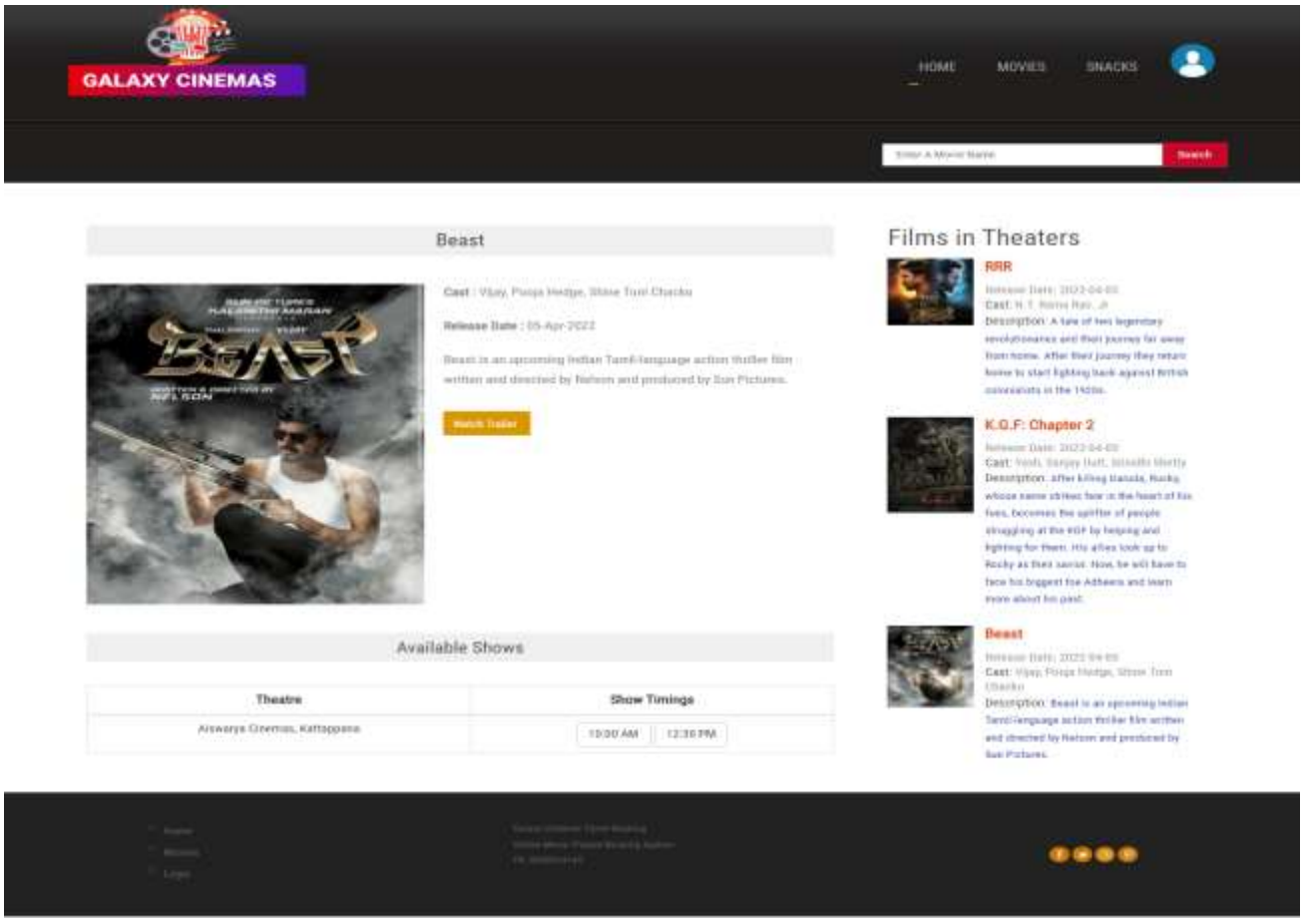


Login Page

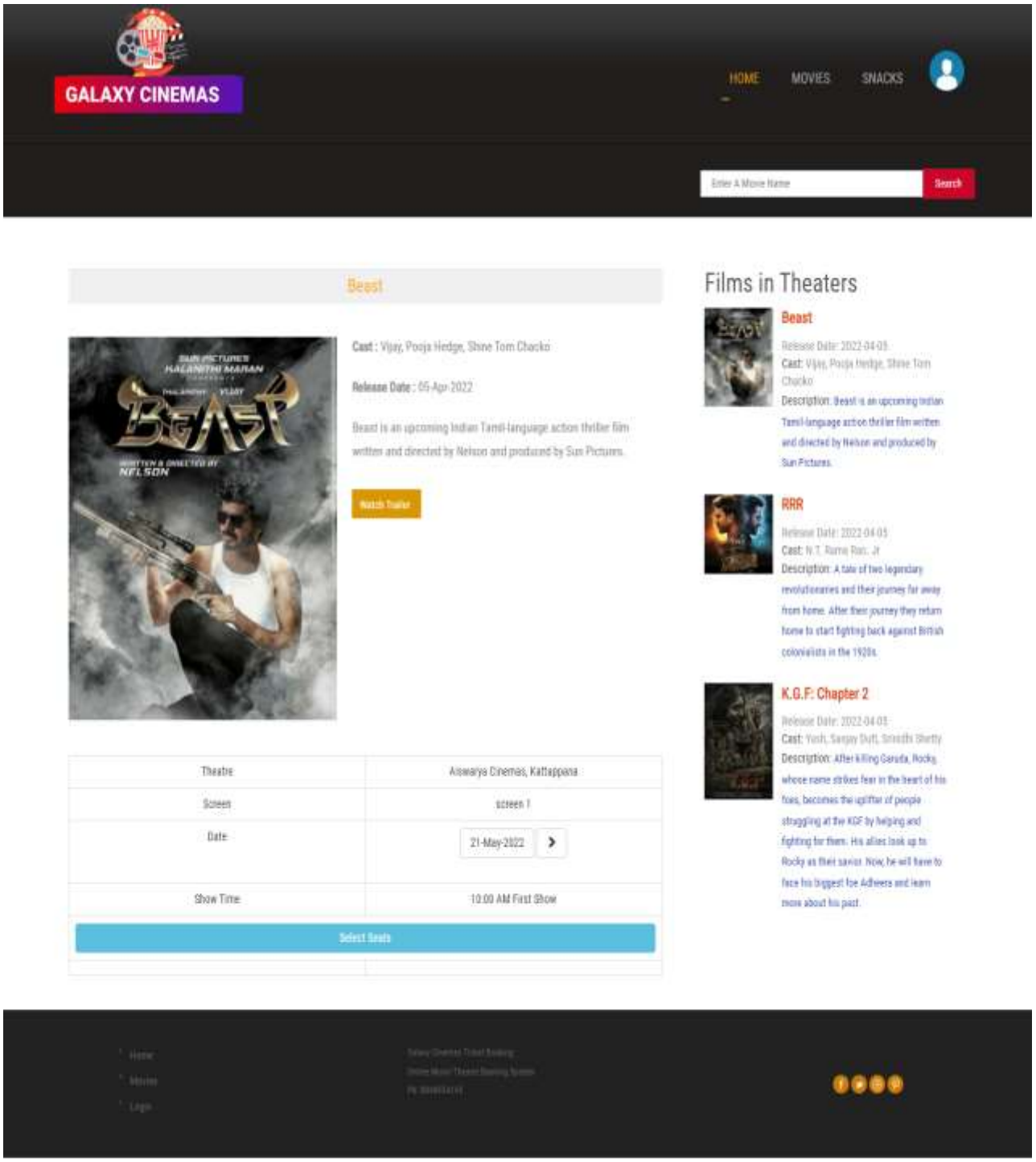
Registration Page



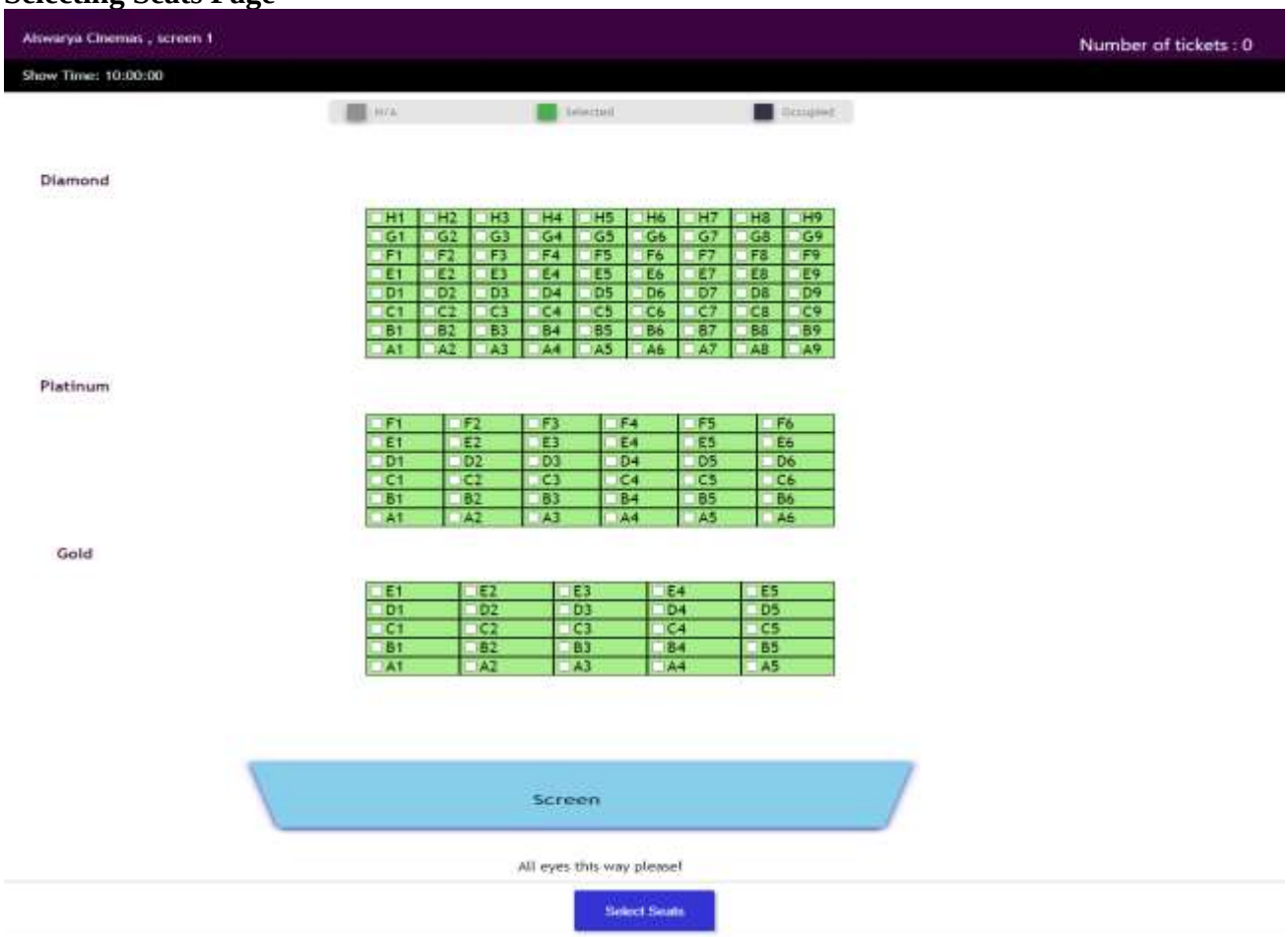
Available shows



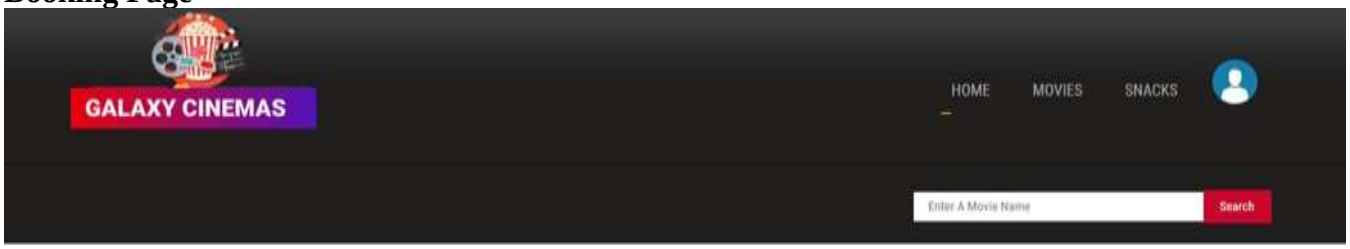
Seat Select


GALAXY CINEMAS
HOME
MOVIES
SNACKS
Enter A Movie Name
Search
Beast
Cast : Vijay, Pooja Hedge, Shine Tom Chacko
Release Date : 05-Apr-2022
Beast is an upcoming Indian Tamil-language action thriller film written and directed by Nelson and produced by Sun Pictures.
Watch Trailer
Theatre
Aiswarya Cinemas, Kattappana
Screen
screen 1
Date
21-May-2022
Show Time
10.00 AM First Show
Select Seats
Films in Theaters
Beast
Release Date: 2022-04-05
Cast: Vijay, Pooja Hedge, Shine Tom Chacko
Description: Beast is an upcoming Indian Tamil-language action thriller film written and directed by Nelson and produced by Sun Pictures.
RRR
Release Date: 2022-04-05
Cast: N.T. Rama Rao, Jr
Description: A tale of two legendary revolutionaries and their journey far away from home. After their journey they return home to start fighting back against British colonialists in the 1920s.
K.G.F: Chapter 2
Release Date: 2022-04-05
Cast: Yash, Sanjay Dutt, Srinidhi Shetty
Description: After killing Garuda, Rocky, whose name strikes fear in the heart of his foes, becomes the uplifter of people struggling at the KGF by helping and fighting for them. His allies look up to Rocky as their savior. Now, he will have to face his biggest foe Adheera and learn more about his past.

Selecting Seats Page



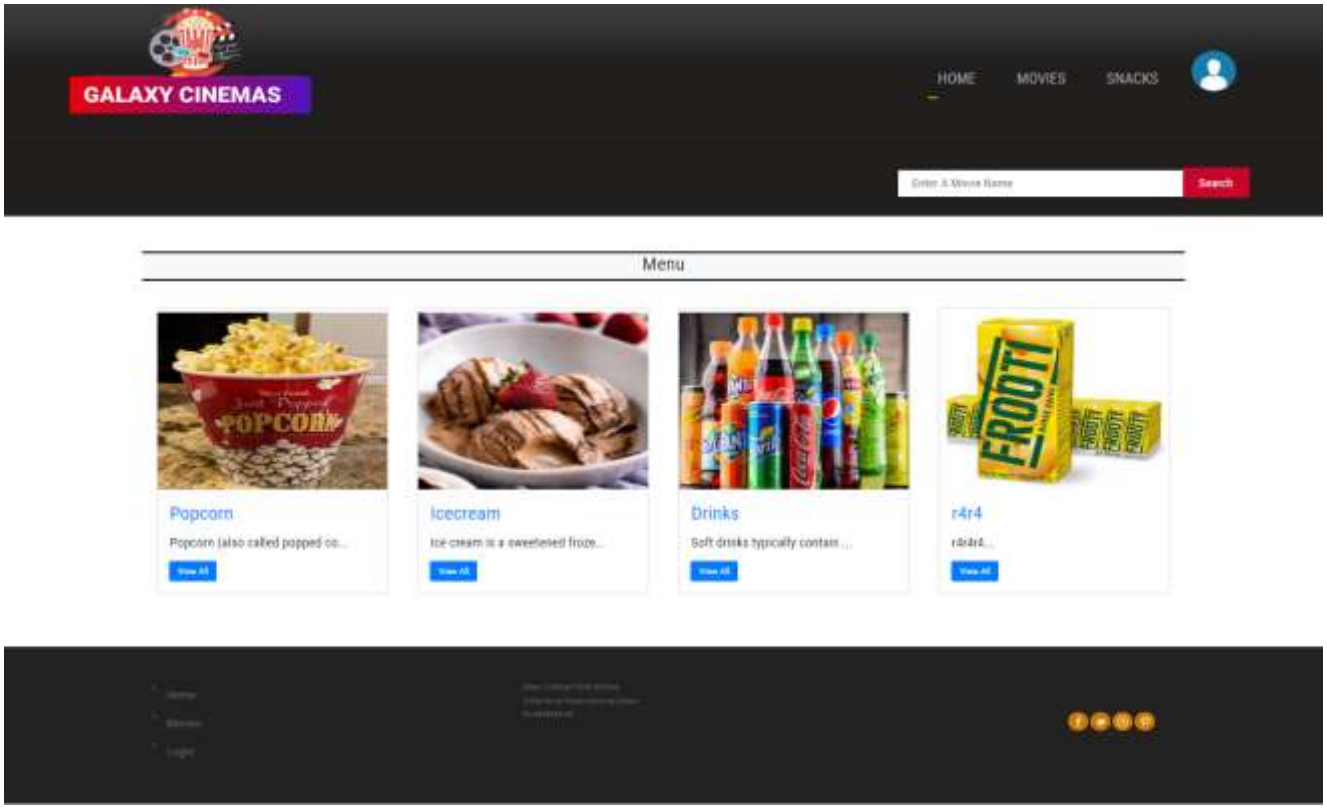
Booking Page



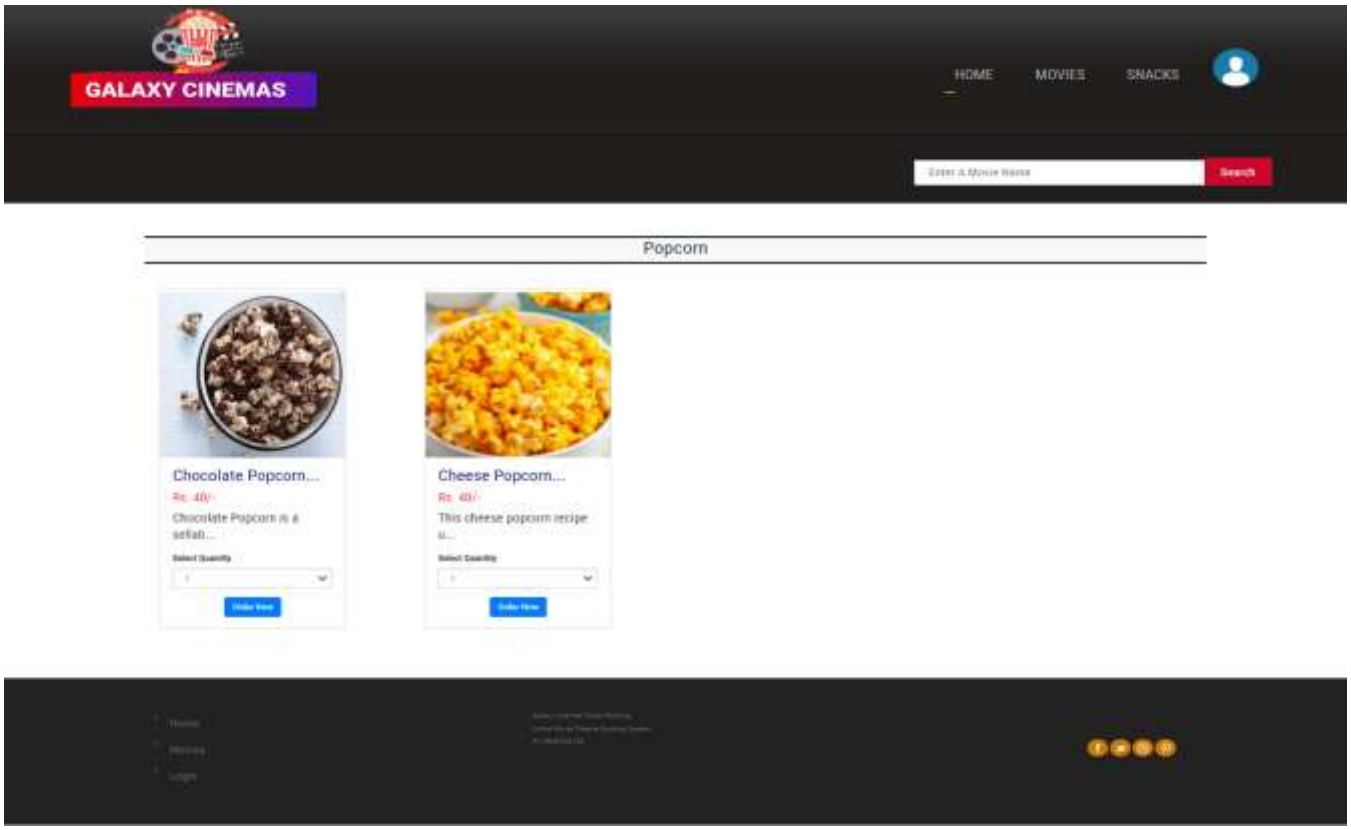
Theatre	Aiswarya Cinemas, Kattappana
Screen	screen 1
Total Seats	2
Show Time	10:00 AM First Show
Seat No	H6 G6
Amount	Rs 120

Book Now

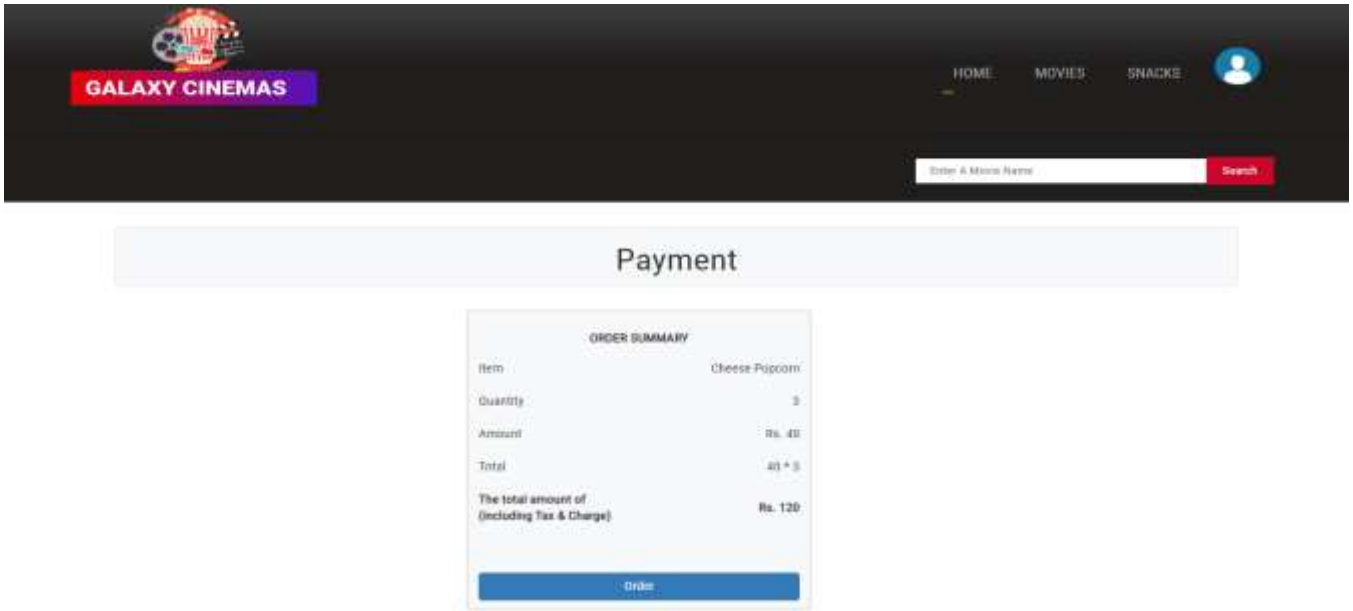
Snack Category Select



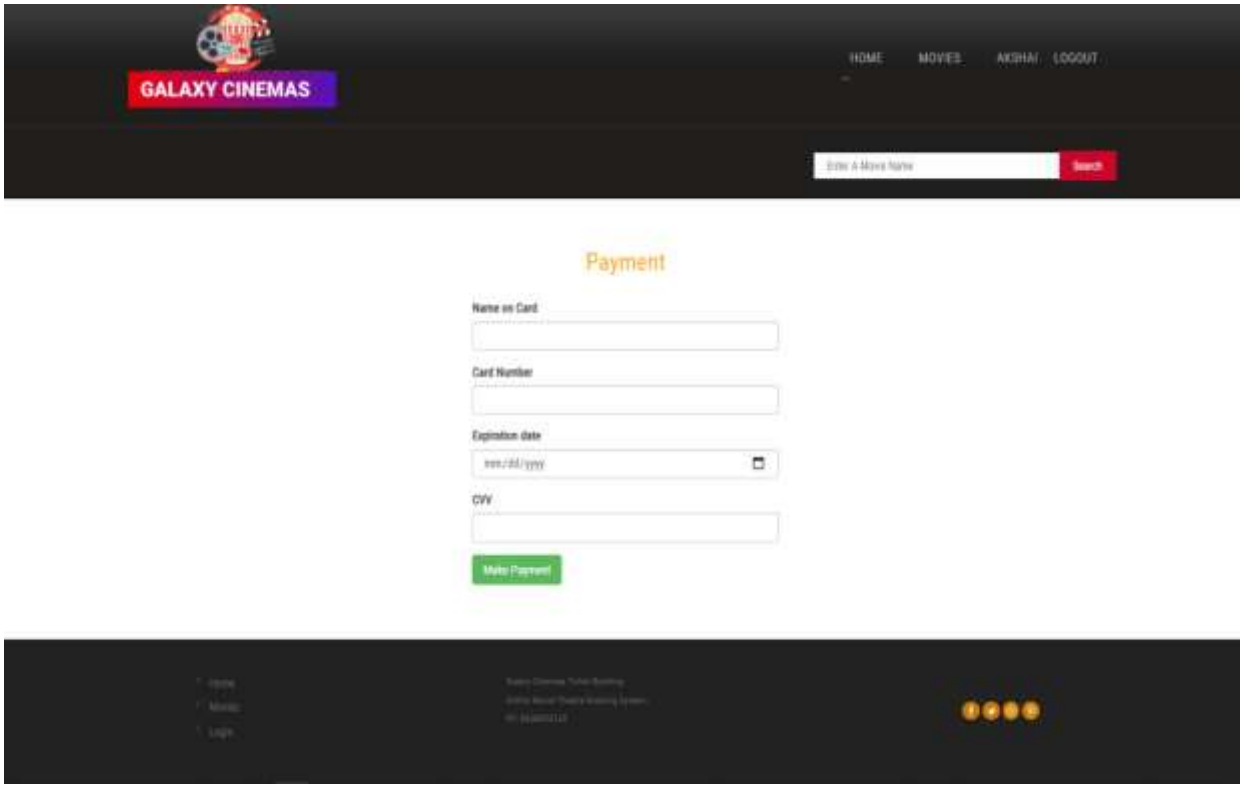
Snack Booking



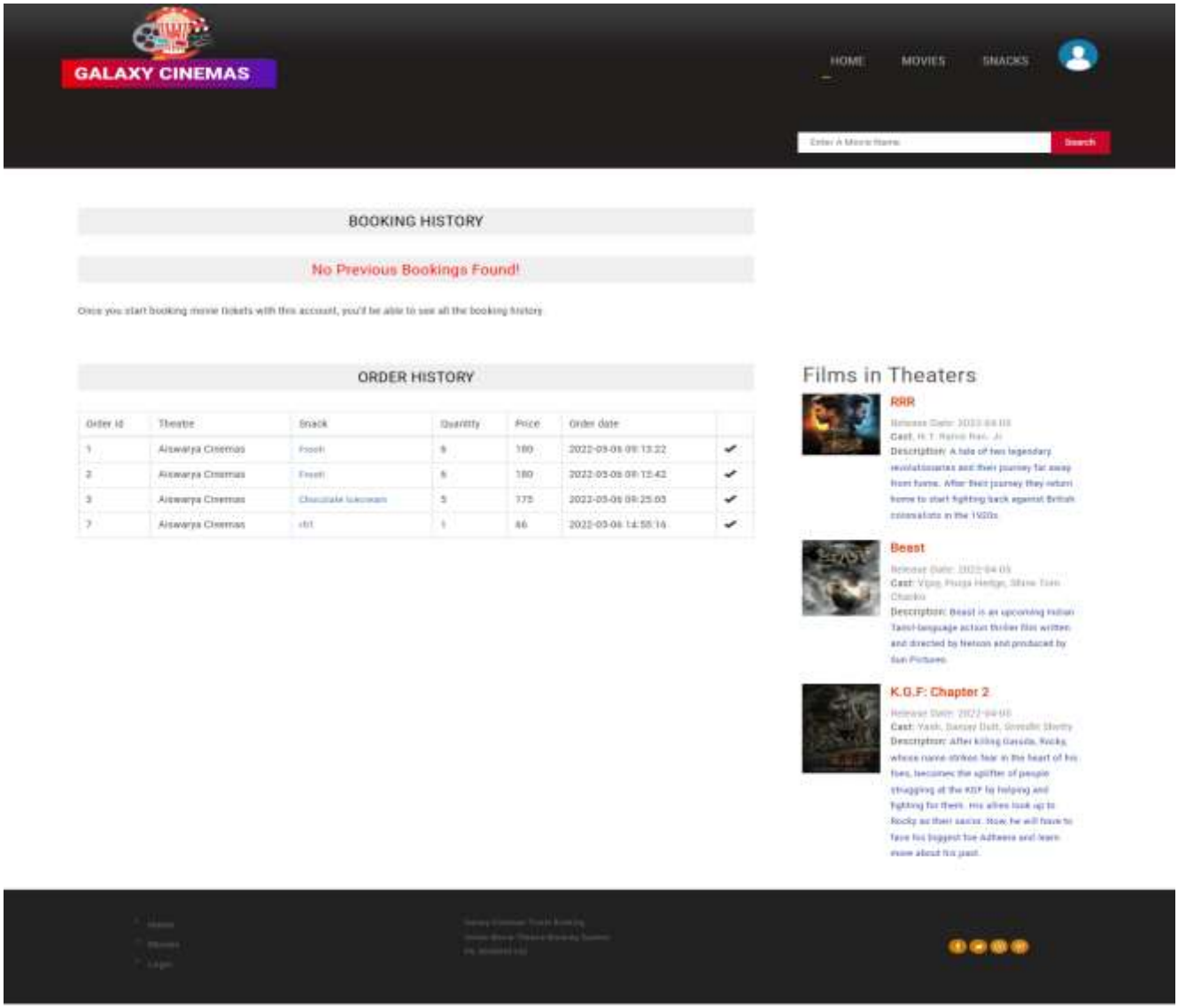
Order Snack



Payment Page

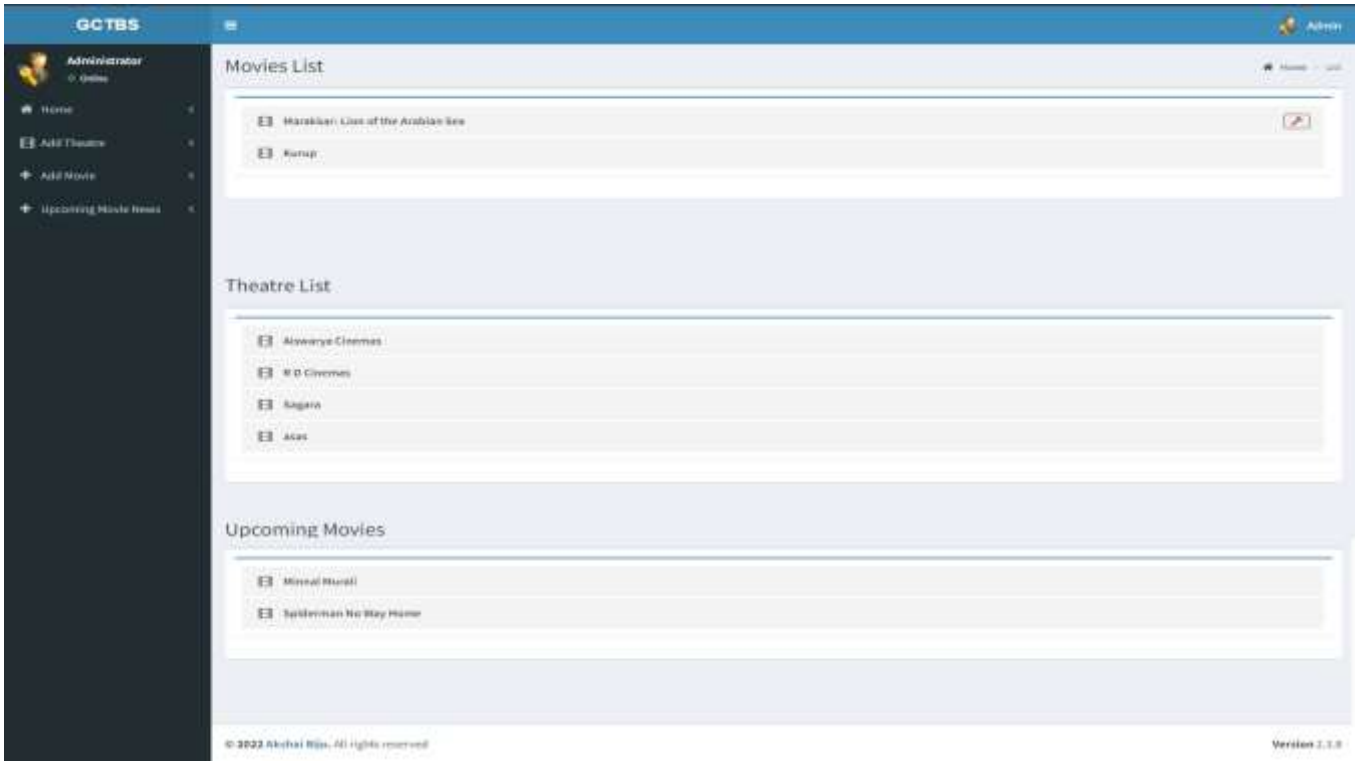


Booking History

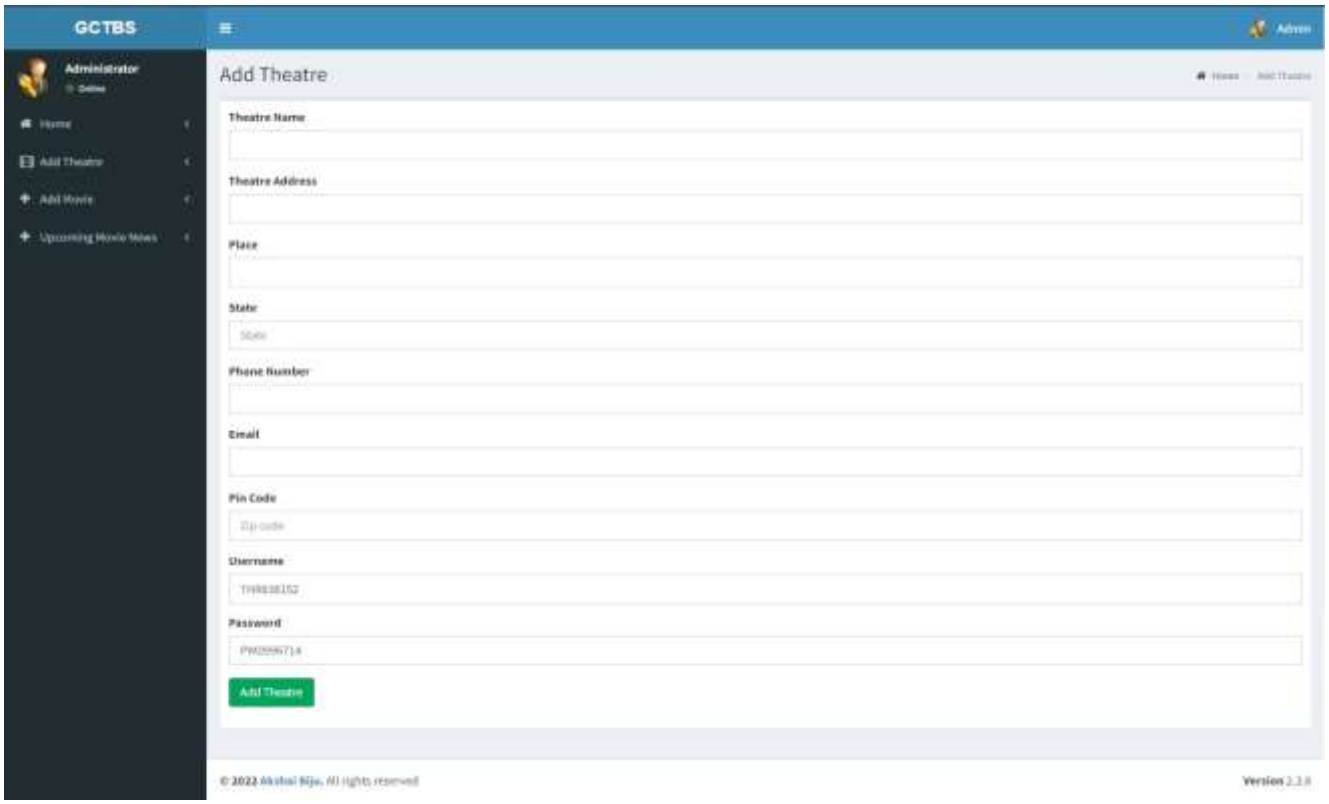


Admin Pages

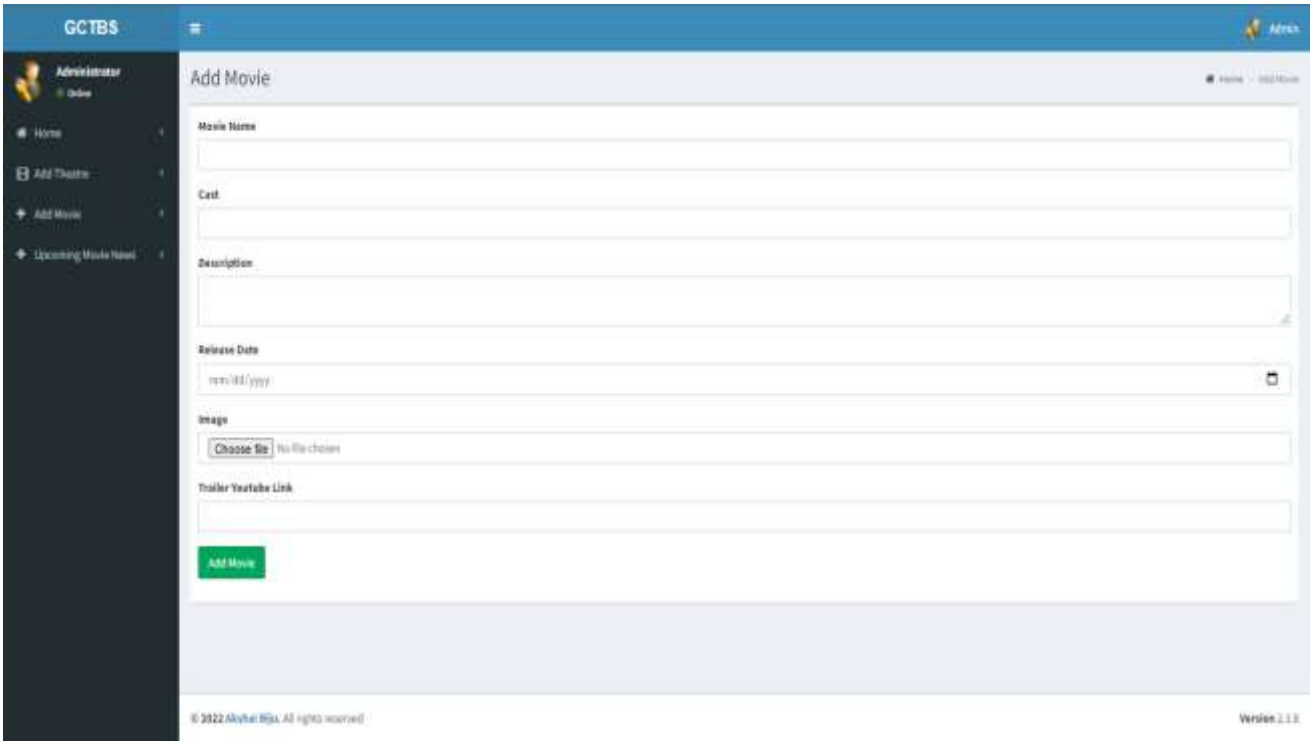
Index page



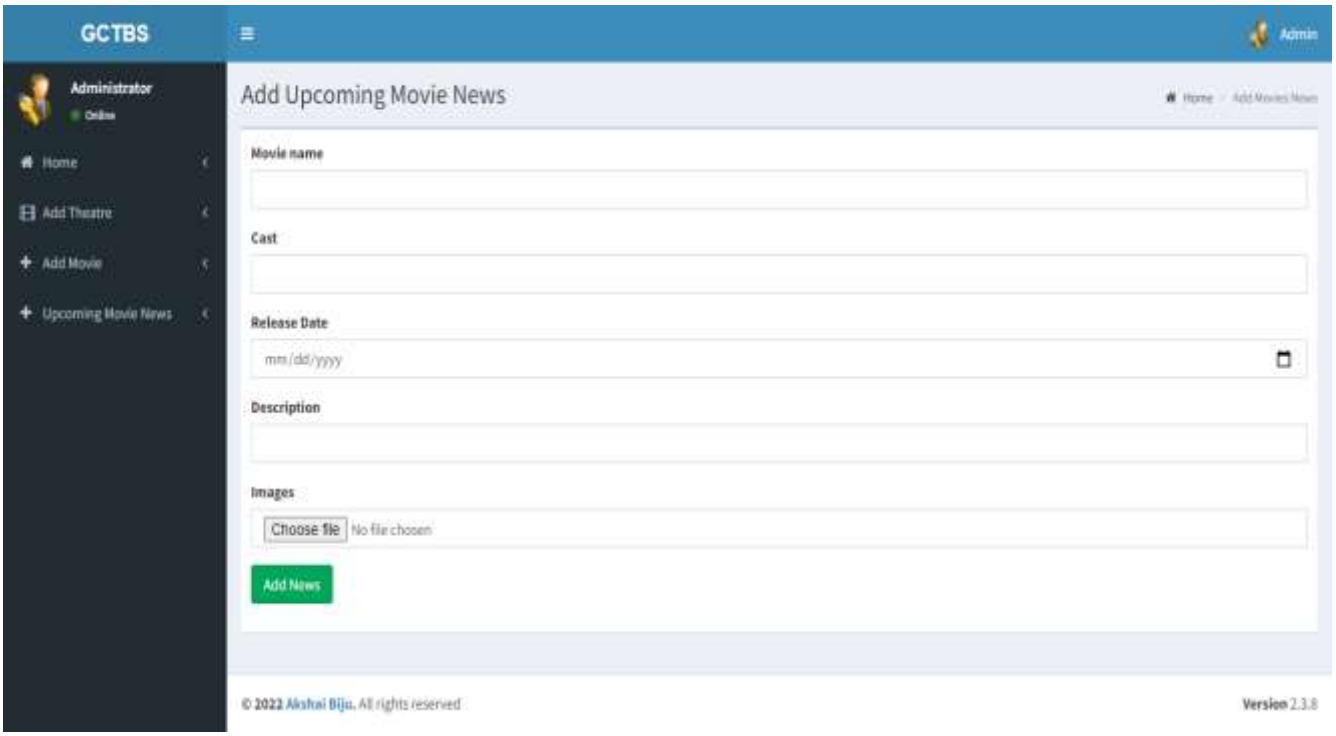
Add Theatre



Add Movie

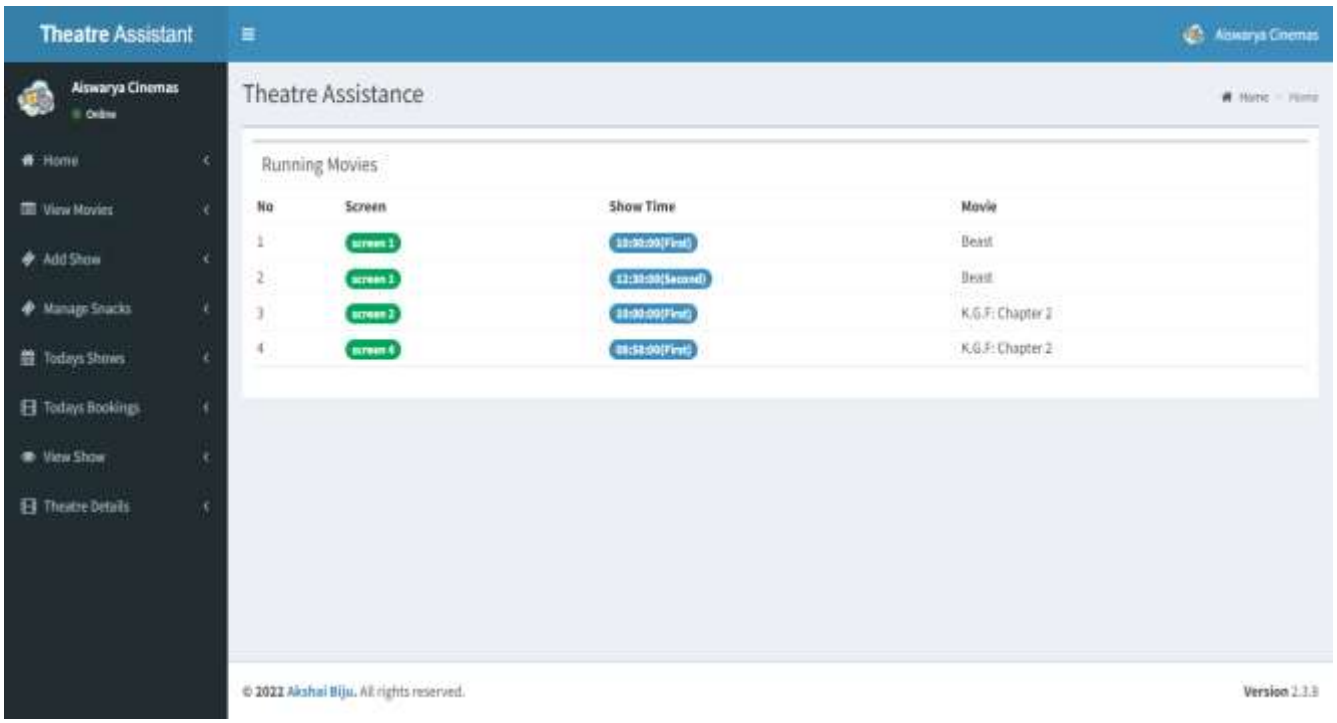


Add Upcoming Movie News

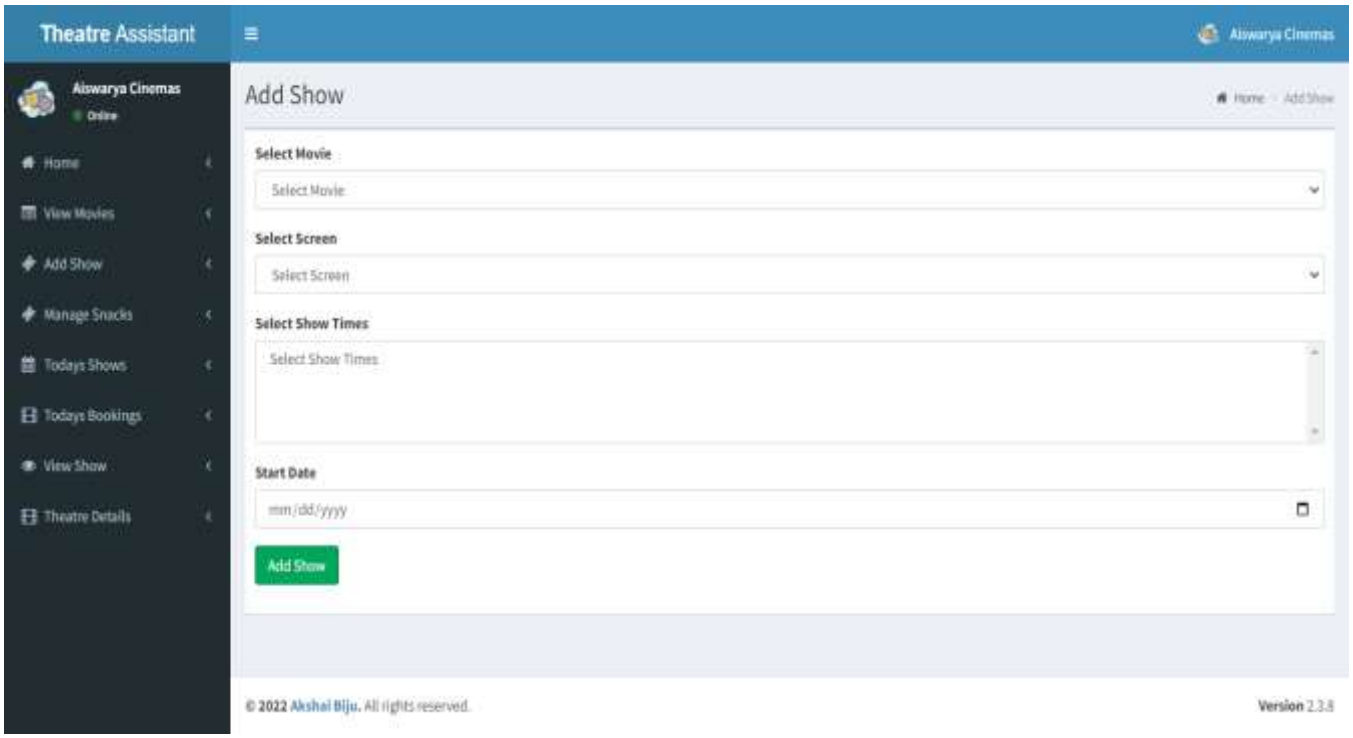


Theatre Owner Pages

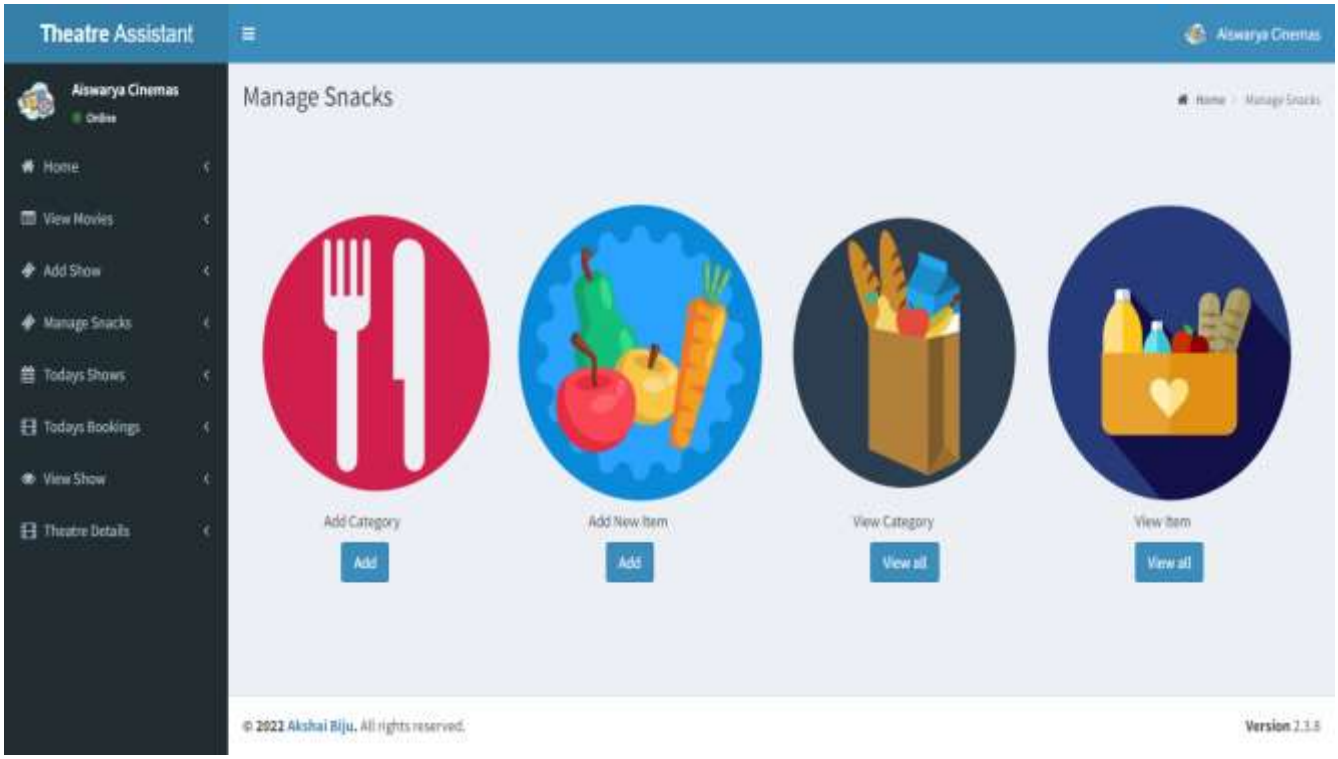
Index page



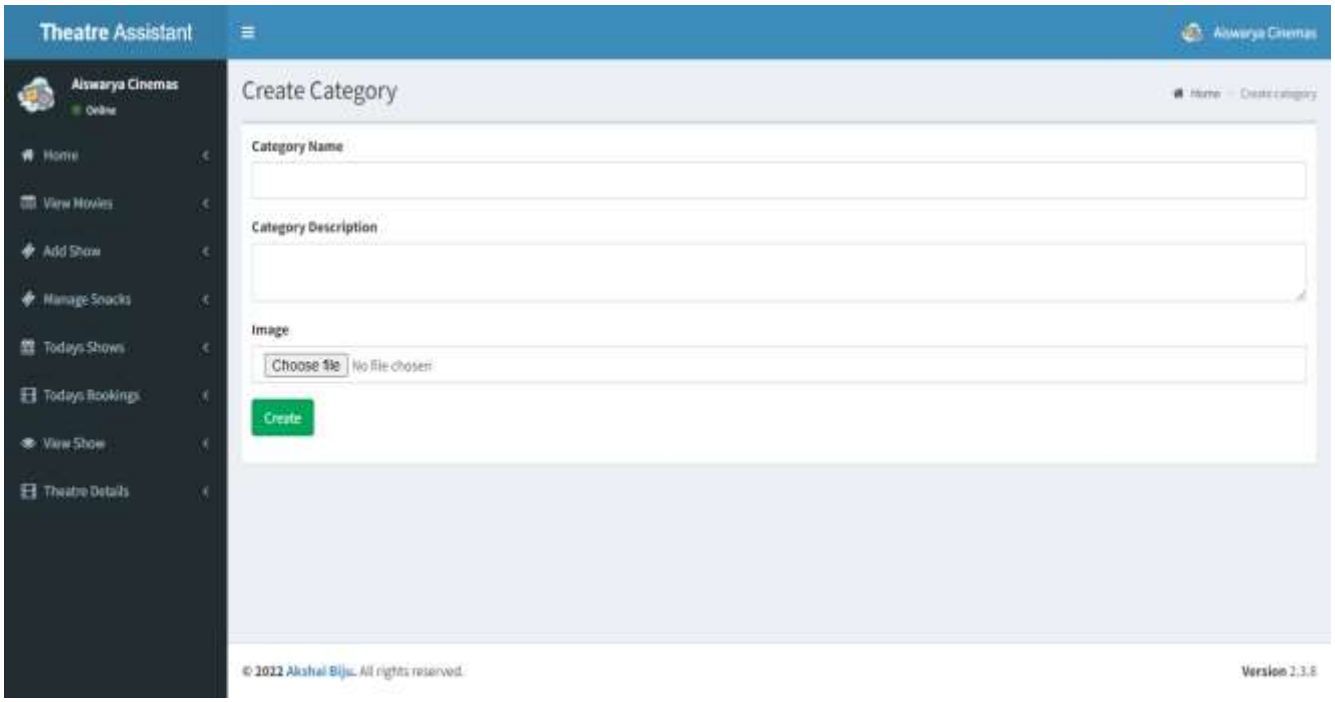
Add Show



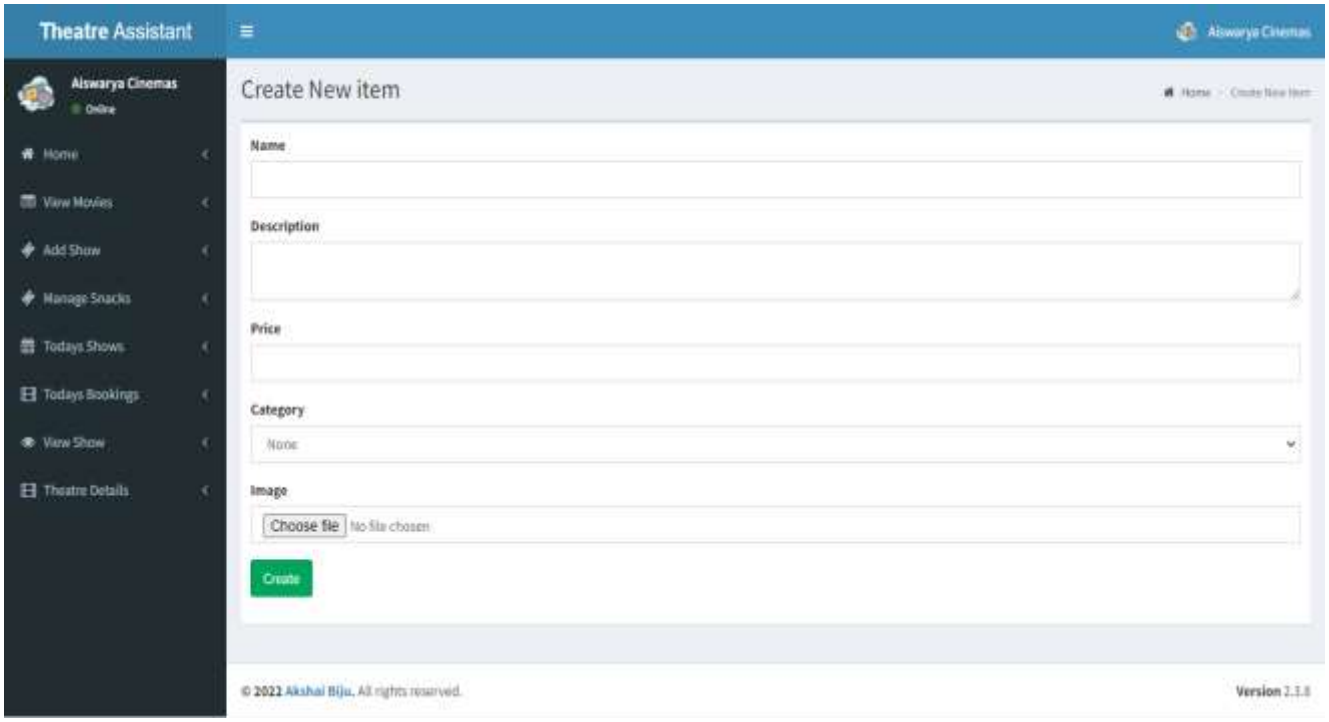
Manage Snacks



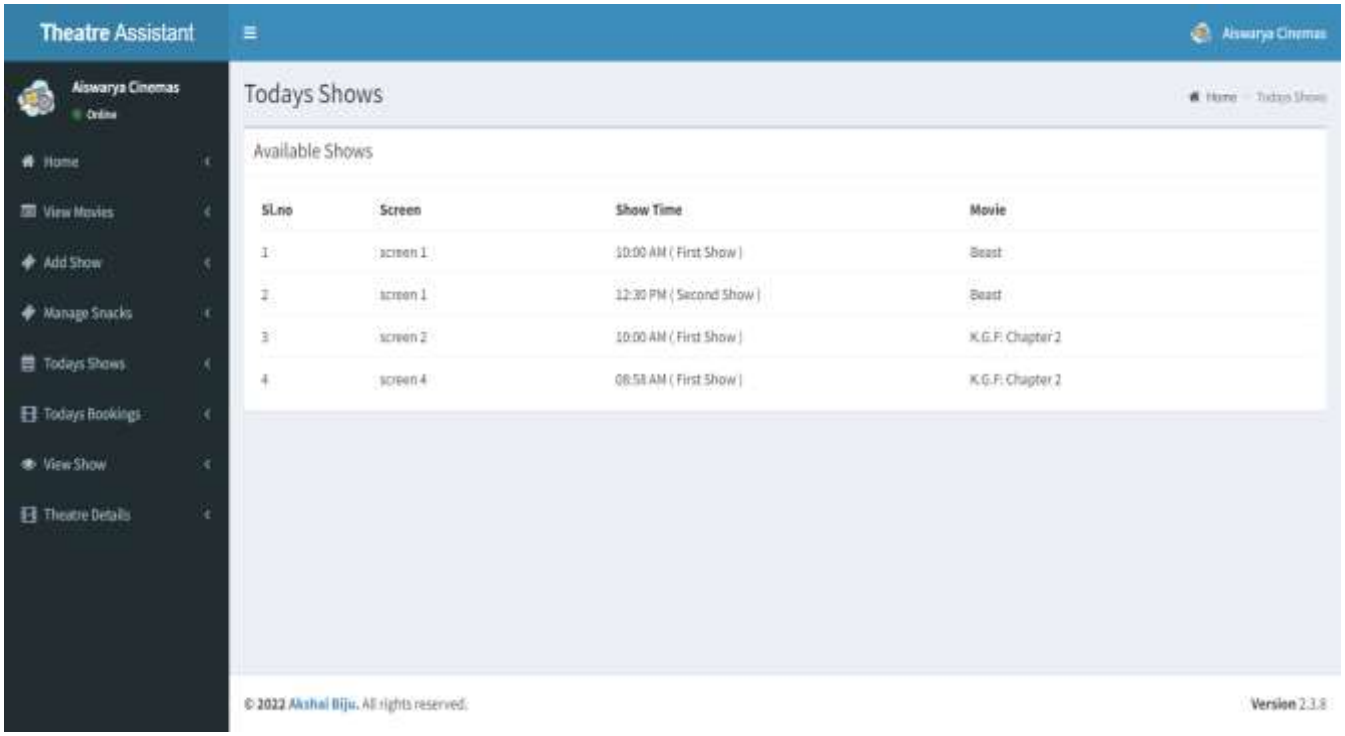
Add Category



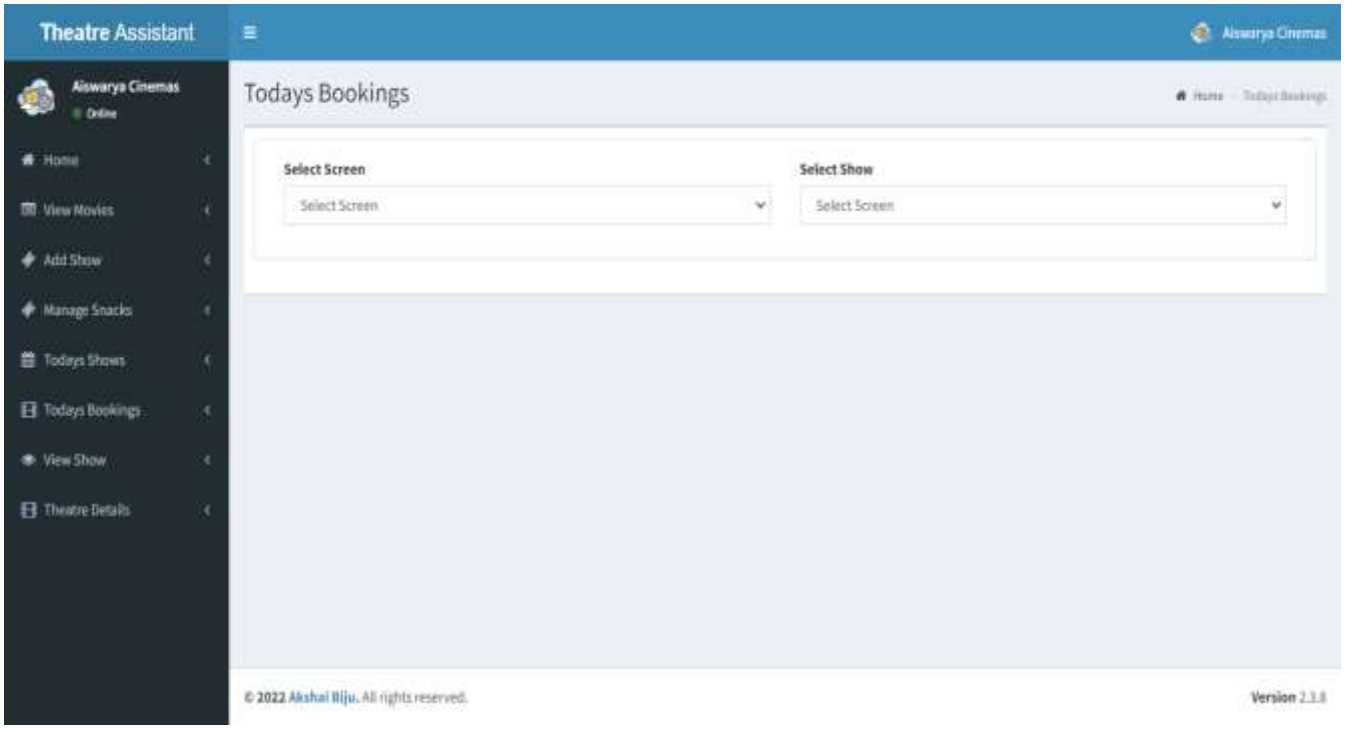
Add Item



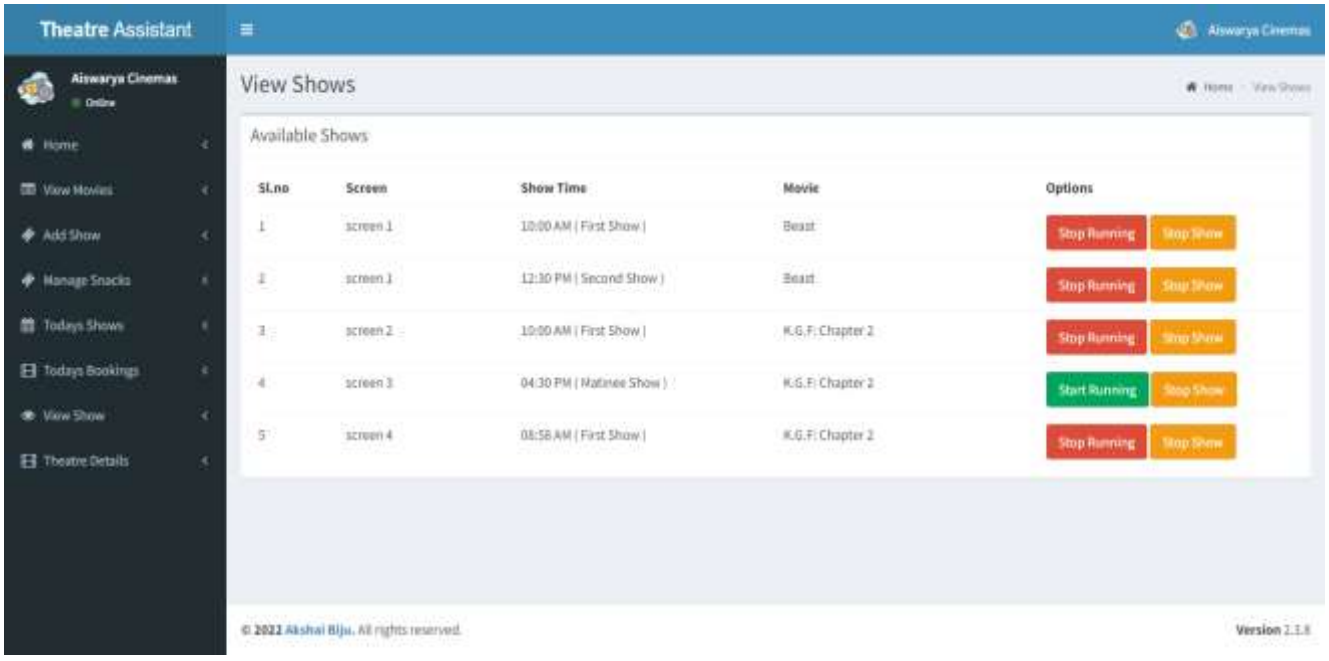
Todays Show



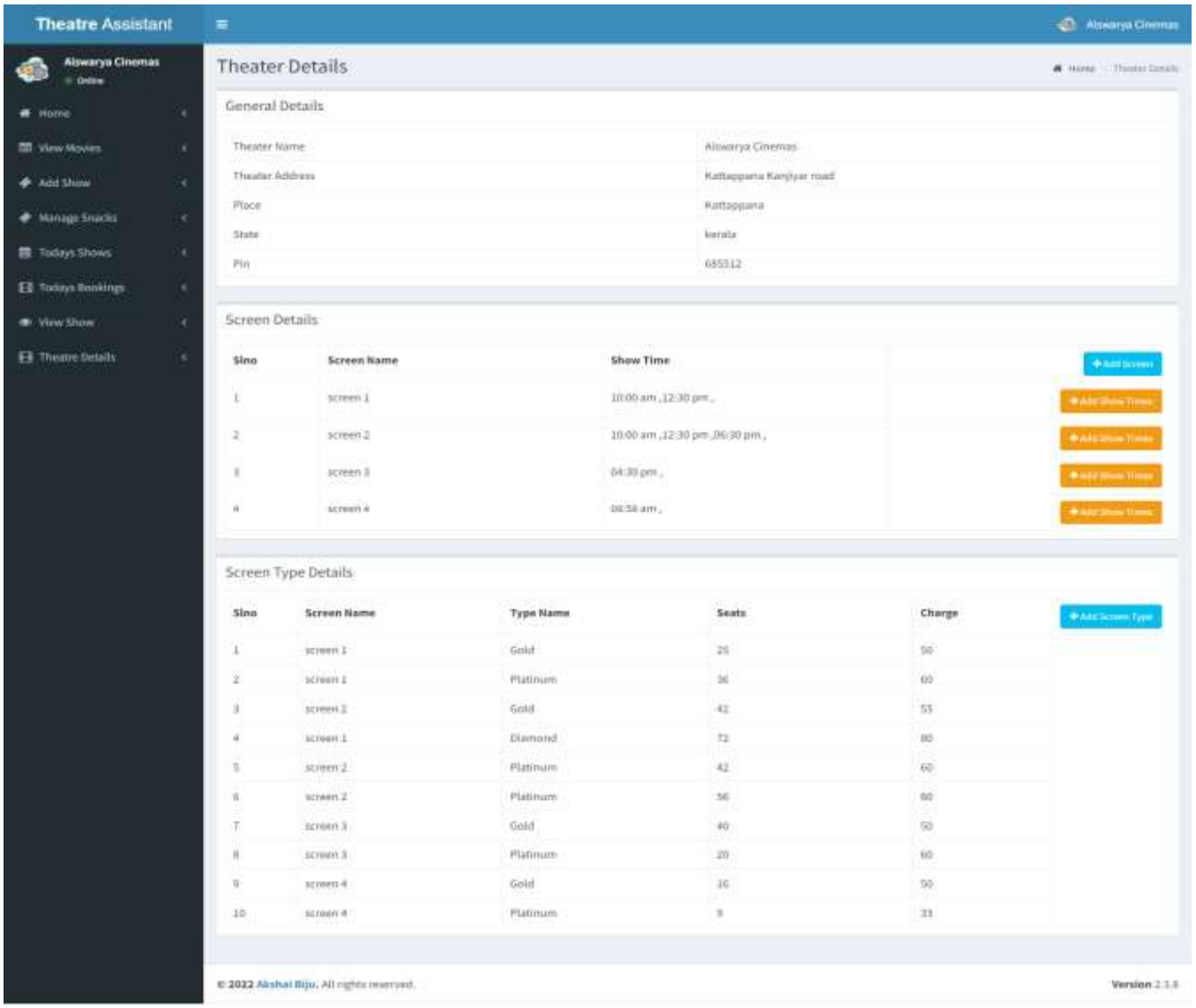
Todays Booking



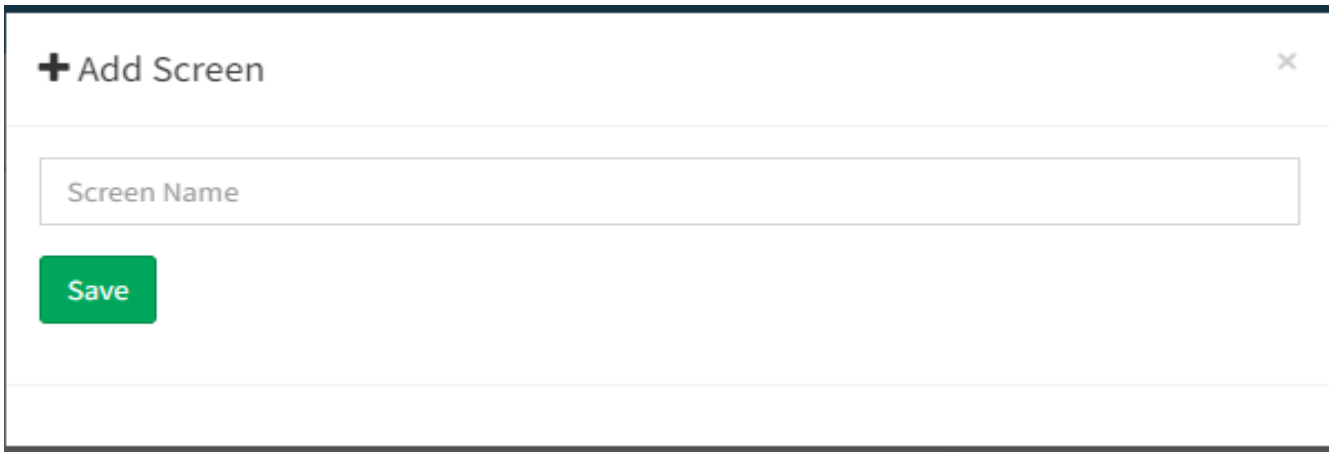
View Show



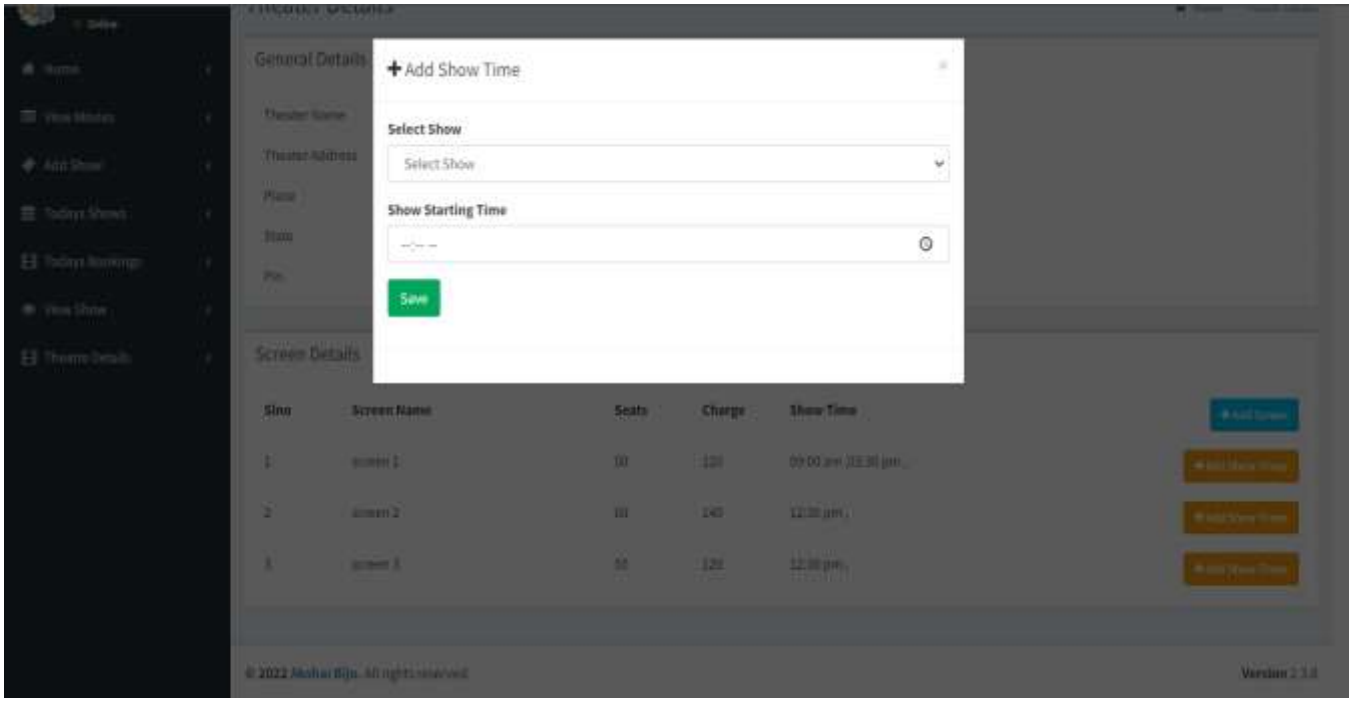
Theatre Details



Add Screen



Add Show



Add Seat Type