TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Project Report

On

Theater Management

Submitted By:

Saugat Pokhrel (THA081BEI036)

Shikshit Bhatrai (THA081BEI039)

Sparsh Bashyal(THA081BEI044)

Yogesh Aryal(THA081BEI048)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

March 2025

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Project Report
On
Theater Management

Submitted By:

Saugat Pokhrel (THA081BEI036)

Shikshit Bhattarai (THA081BEI039)

Sparsh Bashyal (THA081BEI044)

Yogesh Aryal (THA081BEI048)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor's Degree in Electronics and Communication Engineering.

Under the Supervision of

Prajwal Pakka

March 2025

DECLARATION

We hereby declare that the report of the project entitled **"Theater Mangement"** which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Electronics and Communication Engineering**, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Saugat Pokhrel (Class Roll No: [081/BEI/036]) _____

Shikshit Bhattarai (Class Roll No: [081/BEI/039]) _____

Sparsh Bashyal (Class Roll No: [081/BEI/044]) _____

Yogesh Aryal (Class Roll No: [081/BEI/048]) _____

Date: March, 2025

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a minor project work entitled "**Theater Management**" submitted by **Student Name, Student Name, Student Name** and **Student Name** in partial fulfillment for the award of Bachelor's Degree in Electronics and Communication Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor's degree of Electronics and Communication Engineering.

_____

Project Supervisor
Prajwal Pakka
Department of Electronics and Computer Engineering, Thapathali Campus

_____

C Teacher
Anup Shrestha
Department of Electronics and Computer Engineering, Thapathali Campus

_____

Umesh Kanta Ghimire
Head of the Department,
Department of Electronics and Computer Engineering, Thapathali Campus

March, 2024

COPYRIGHT

ACKNOWLEDGEMENT

We would like to express my sincere gratitude to everyone who contributed to the successful completion of this Movie Ticketing System. Special thanks to our mentors and teammates for their invaluable guidance and support. We also appreciate the resources and tools that helped streamline the development process. Lastly, We are grateful for the opportunity to work on this project, which has enhanced my skills and knowledge.

Saugat Pokhrel (Class Roll No: [081/BEI/036])
Shikshit Bhattarai (Class Roll No: [081/BEI/039])
Sparsh Bashyal (Class Roll No: [081/BEI/044])
Yogesh Aryal (Class Roll No: [081/BEI/048])

ABSTRACT

This project aims to develop a Theater Management System using the C programming language. The system will provide a console-based interface for managing theater operations, including ticket booking, seat management, movie scheduling, and customer management. The system will be designed to handle multiple screens, showtimes, and ticket pricing, ensuring a smooth and efficient experience for both customers and theater staff. The project will focus on creating a robust and efficient system using C, leveraging its low-level capabilities for performance and reliability.

Keywords: Theater Management, Console-based System, Seat Management, Ticket Booking, Movie Screening

Table of Contents

DECLARATION ..... **i**

CERTIFICATE OF APPROVAL ..... **ii**

COPYRIGHT ..... **iii**

ACKNOWLEDGEMENT ..... **iv**

ABSTRACT ..... **v**

Table of Contents ..... **vi**

List of Figures ..... **viii**

List of Abbreviations ..... **ix**

1. INTRODUCTION **1**

1.1 Background Introduction 1

1.2 Motivation 1

1.3 Problem Definition 1

1.4 Objectives 2

1.5 Scope and Applications 2

2. LITERATURE REVIEW **3**

Core Functions of a Theater Management System: 3

1. Content Management: 3

2. Show Scheduling 3

3. Ticketing and Reservations: 3

5. Resource Allocation 3

Benefits of Implementing a Theater Management System: 4

3. REQUIREMENT ANALYSIS **5**

3.1 Project Requirement: 5

4. SYSTEM ARCHITECTURE AND METHODOLOGY **6**

4.1 System Architecture..... 6

4.2 Parts of the Program..... 7

4.2.1 Menu..... 7

4.2.1.1 Admin Login..... 7

4.2.1.2 User Login..... 8

4 IMPLEMENTATION DETAILS..... 9

5.1 Waterfall Model..... 9

5.1.1 Requirement Analysis..... 9

5.1.2 System Design..... 9

5.1.3 Implementation..... 10

5.1.4 Testing..... 10

5.1.5 Deployment..... 10

5.1.6 Maintainance..... 11

6 RESULTS AND ANALYSIS..... 12

7 FUTURE ENHANCEMENT..... 14

8 CONCLUSION..... 15

9. REFRENCES:..... 16

List of Figures

Figure 4-1: Block Diagram of Entry Point 6
Figure 4-2: Admin Interface 7
Figure 4-3: User Interface..... **.....8**

List of Abbreviations

TMS: Theater Management System

GNU: GNU's Not Unix!

GCC: GNU C Compiler

IDE: Integrated Development Environment

1. INTRODUCTION

1.1 Background Introduction

The entertainment industry, particularly the movie theater business, has seen significant growth over the years. With the increasing number of moviegoers, managing theater operations manually has become inefficient and error-prone. The need for an automated system to handle ticket bookings, seat management, and movie scheduling is more critical than ever. A TMS aims to address these challenges by providing a centralized platform for managing all theater operations. This project will implement the system using the C programming language, leveraging its efficiency and low-level control.

1.2 Motivation

The motivation behind this project is to simplify the complex processes involved in managing a movie theater. By automating tasks such as ticket booking, seat allocation, and movie scheduling, theater staff can focus more on providing excellent customer service rather than dealing with administrative tasks. Additionally, customers will benefit from a seamless booking experience, reducing wait times and improving overall satisfaction. Using C ensures that the system is efficient, fast, and reliable, even with large amounts of data.

1.3 Problem Definition

The current manual processes for managing theater operations are time-consuming and prone to errors. Customers often face long queues, and theater staff struggle to manage multiple tasks simultaneously. The proposed Theater Management System aims to automate these processes, providing a more efficient and user-friendly solution for both customers and theater staff. The system will be implemented in C, ensuring high performance and reliability.

1.4 Objectives

The main objectives of our project are listed below To develop a console-based Theater Management System in C that automates ticket booking, seat management, and movie scheduling. To provide an intuitive user interface for both customers and theater staff.

1.5 Scope and Applications

The proposed system can be used in: Movie theaters to manage daily operations efficiently. Small to medium-sized theaters that require a lightweight and efficient system. Educational institutions for teaching C programming and system design

2. LITERATURE REVIEW

A Theater Management System (TMS) is a comprehensive software solution designed to streamline the multifaceted operations of theaters, encompassing movie scheduling, screen assignments, ticket reservations, and ticket printing. By integrating these functions, a TMS enhances efficiency and improves the overall management of theater activities. The missions of a TMS mean that it will be connected to at least all of the cinema's reading servers. This therefore makes it a natural supervision tool. It happens that some TMS integrate with the cinema's cash register system or even the advertising management system.[1]

Core Functions of a Theater Management System:

1. Content Management: TMS platforms manage Digital Cinema Packages (DCPs), including films, trailers, teasers, and advertisements. This ensures that all media content is organized and readily accessible for scheduled screenings.

2. Show Scheduling: The system facilitates the planning and scheduling of movie showtimes, allowing managers to allocate films to specific screens and times efficiently. This aids in optimizing screen utilization and meeting audience demand.

3. Ticketing and Reservations: Modern TMS solutions often integrate ticketing functionalities, enabling online and on-site ticket sales, seat reservations, and real-time availability updates. This integration enhances the customer experience by providing convenient booking options.

5. Resource Allocation: Beyond scheduling, TMS assists in managing theater resources such as seating arrangements, ensuring optimal utilization and adherence to safety protocols.

Benefits of Implementing a Theater Management System:

- **Operational Efficiency:**

By automating routine tasks, TMS reduces manual labor, minimizes errors, and allows staff to focus on enhancing customer service.

- **Enhanced Customer Experience:**

Features like online booking, seat selection, and timely information updates improve the patron's experience, leading to increased satisfaction and loyalty.

- **Data-Driven Decision Making:**

TMS provides valuable insights through data analytics, helping managers make informed decisions regarding scheduling, marketing strategies, and resource allocation.

Challenges and Considerations:

While TMS offers numerous advantages, theaters must consider factors such as system integration with existing infrastructure, staff training, and data security. Selecting a system that aligns with the theater's specific needs and scale is crucial for maximizing the benefits of TMS implementation. In conclusion, a Theater Management System serves as a pivotal tool in modern cinema operations, offering a centralized platform to manage various aspects of theater management efficiently. Its integration into theater operations not only streamlines processes but also enhances the overall experience for both staff and patrons.

3. REQUIREMENT ANALYSIS

3.1 Project Requirement:

1)GCC compiler

2)An IDE

3) Git

4) GitHub

5)Make(optional) [2]

4. SYSTEM ARCHITECTURE AND METHODOLOGY

4.1 System Architecture

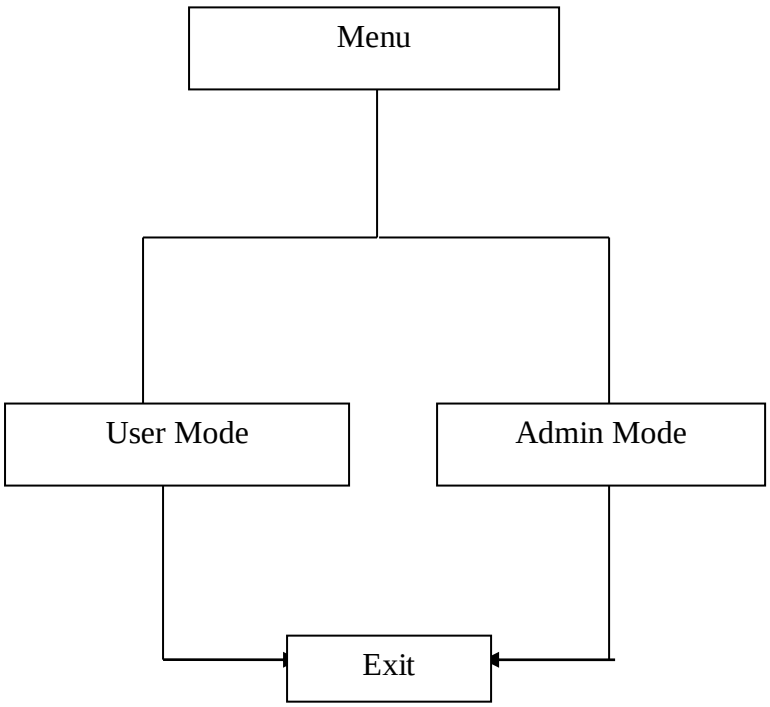


Figure 0-1: Block Diagram of entry point

4.2 Parts of the Program

4.2.1 Menu

The home page of the application has the title “Theater Management System” and the menu which shows login and create account and the users privilege will be checked in program and access will be given accordingly.

4.2.1.1 Admin Login

The admin login menu requires the admin login credentials which should be filled up by the respective admin to do the activities related to theater. The activities can be selected from the menu presented.

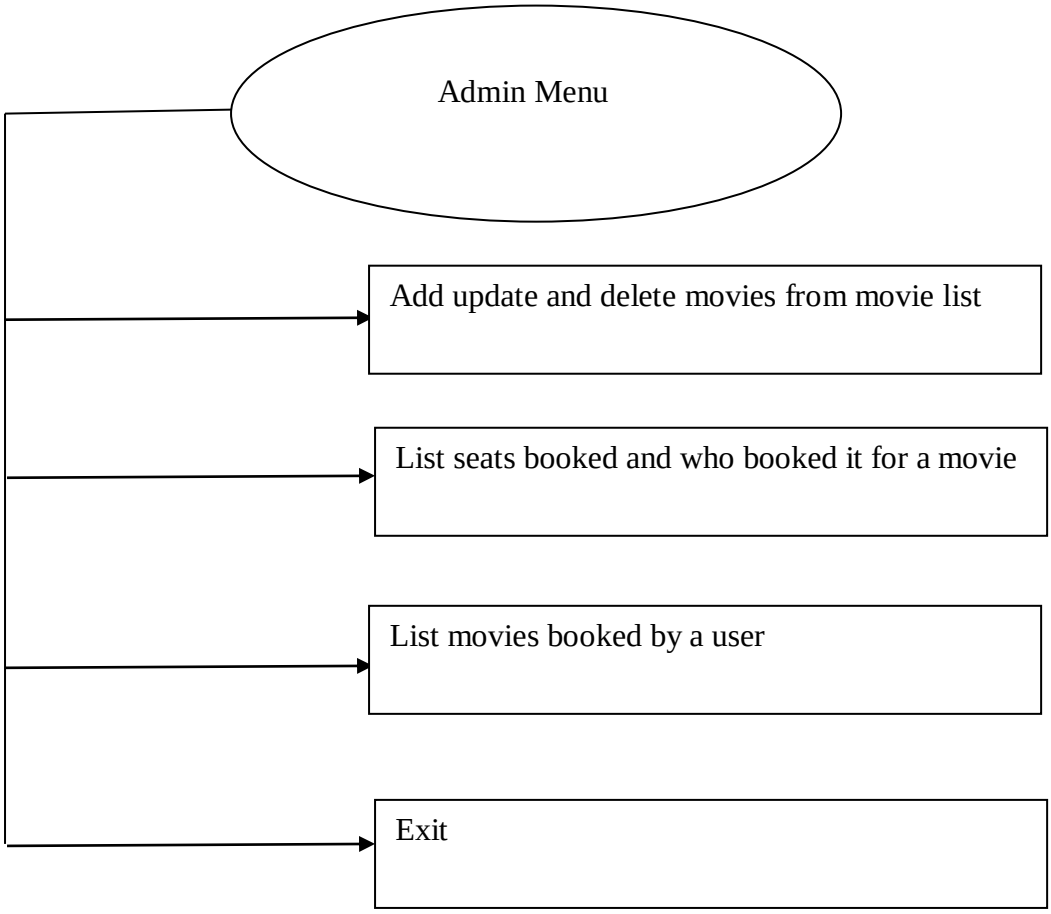


Figure 4.2 Admin interface

4.2.1.2 User Login

The user can see the movies that are being shown and will be able to book seat's accordingly.

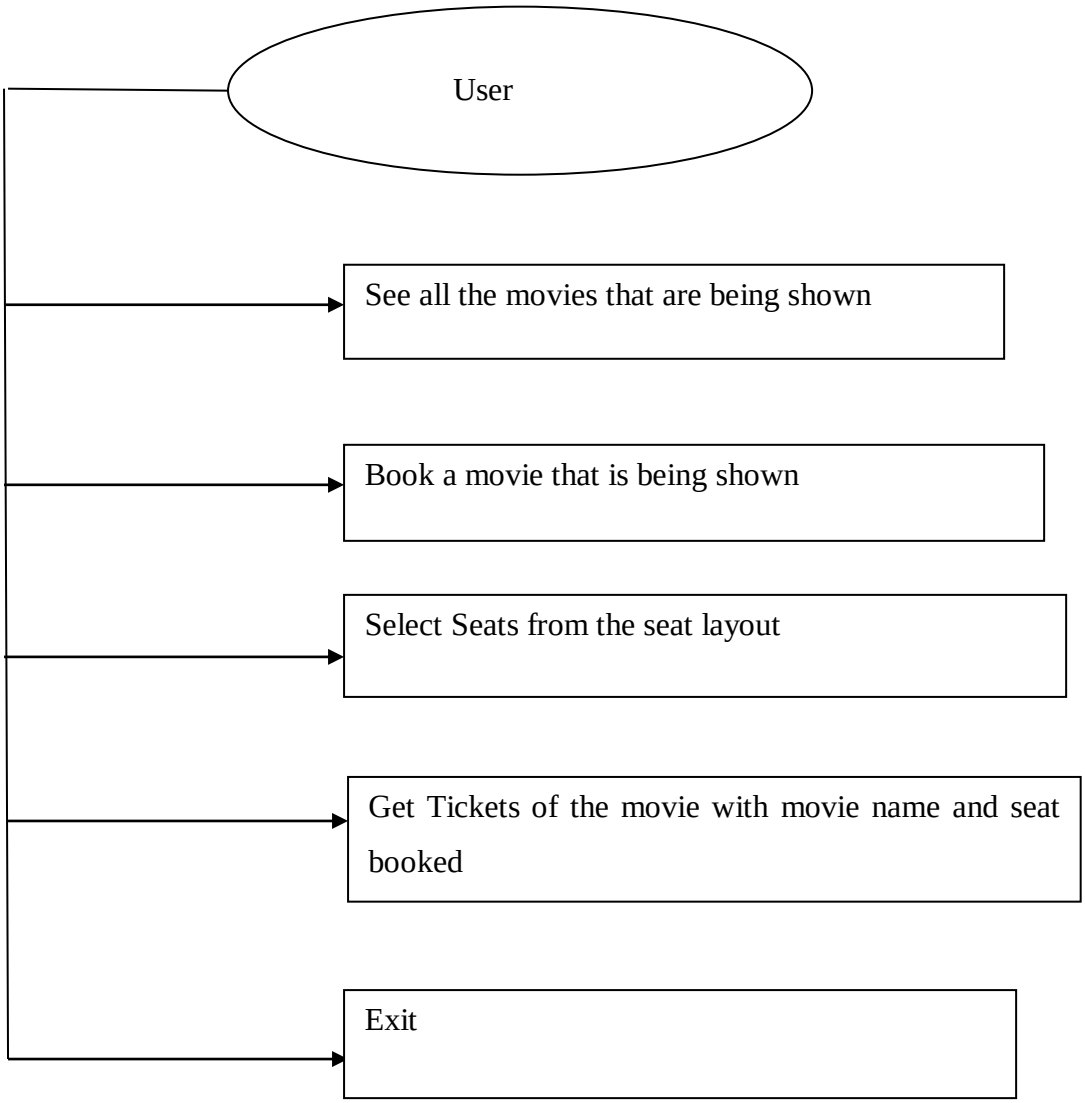


Figure: 4.3 User Interface

4 IMPLEMENTATION DETAILS

In Implementation we apply the concept of the waterfall model in the program, files.

5.1 Waterfall Model

The waterfall model is used in step by step analysis of the movie ticketing system. This approach makes the system structured, easy to understand, and maintainable**.**

5.1.1 Requirement Analysis

- It is the planning stage.
- The work such as manage movie bookings**,** including adding/removing movies**,** booking seats, and viewing reservations falls under requirement analysis.
- It defines core requirement such as storing movies (movies.txt), storing bookings (bookings.txt), admin and user authentication (defs.h defines constants like ADMIN_USERNAME) , managing seats in a theater (seats[ROWS][COLS]).

5.1.2 System Design

- The system is modular, meaning different functionalities are implemented in separate functions.
    - Movie Management: addMovie(), removeMovie(), loadMovies(), saveMovies()
    - Seat Management: initializeSeats(), showSeats(), updateSeatsFromBookings()
    - Booking System**:** writeBookingToFile(), showUserBookings()
    - Admin Controls: adminMenu(), checkAdmin()

- defs.h acts as the blueprint by defining constants and function prototypes.

5.1.3 Implementation

- The system is implemented in C programming language, using:
  - File handling (for storing movies and bookings).
  - Data structures (2D arrays for seats, arrays for movies).
  - Functions for modularity (separating responsibilities).
- Code files:
  - main.c → Controls the overall execution.
  - defs.h → Stores constants and function declarations.
  - admin.c → Manages movie-related functions.
  - seats.c → Handles seat-related operations.

5.1.4 Testing

- The code has basic error handling for file operations:
  - Checks if movies.txt exists before reading.
  - If bookings.txt is missing, it does not crash.
  - Admin authentication prevents unauthorized access (checkAdmin()).
- Testing can be done by:
  - Running the program to see if movies and bookings load properly.
  - Trying to add/remove movies and checking if the file updates correctly.
  - Booking a seat and verifying if it appears as "X" in the seat layout.

5.1.5 Deployment

- The program can be compiled (gcc main.c -o movie_booking) and run on a terminal-based environment.
- The data is persistent (movies and bookings are stored in text files).
- Future improvements (e.g., adding a GUI) can be considered.

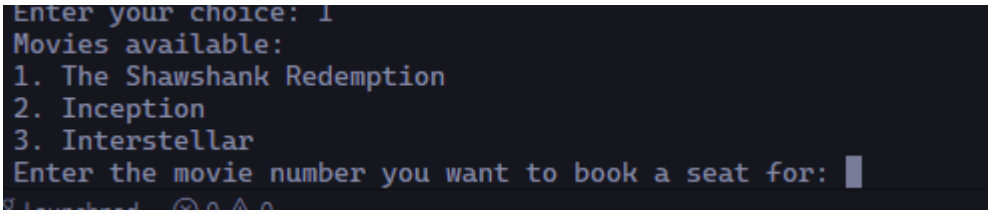
5.1.6 Maintainance

- Adding new features (e.g., user login, cancel booking).
- Fixing issues (e.g., handling duplicate bookings).
- Improving security (e.g., encrypting passwords instead of storing plain text).

6 RESULTS AND ANALYSIS

The Theater Management System successfully implemented all core functionalities, including movie scheduling, seat booking and ticket generation. The system was tested extensively, and the results demonstrate its effectiveness in managing theater operations efficiently.

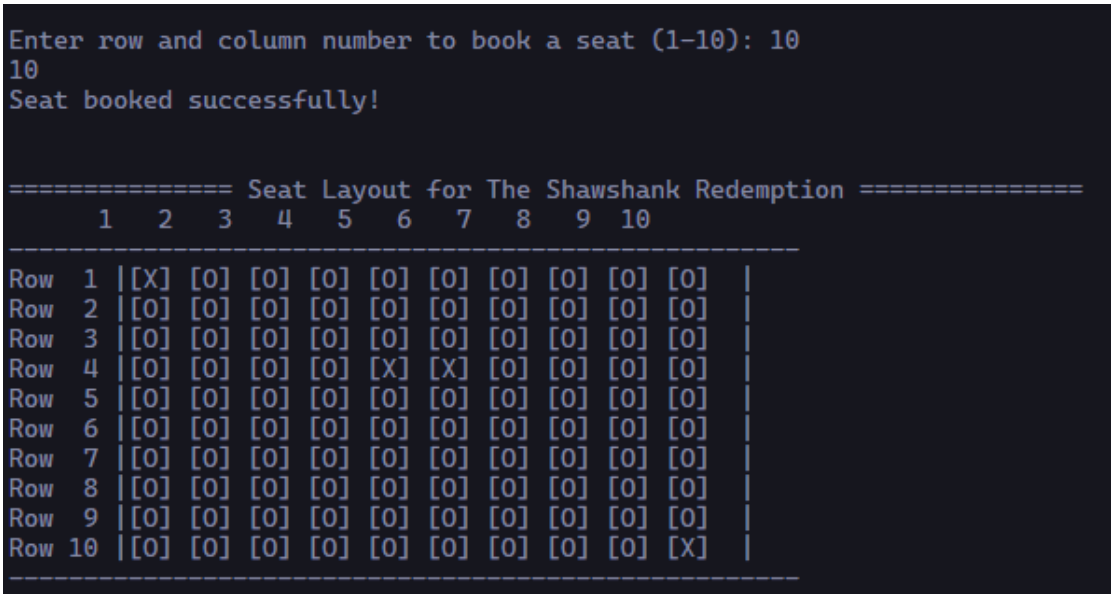
Movie Scheduling



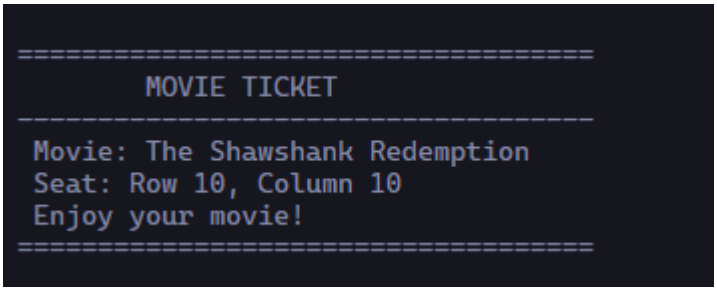
The system allows administrators to add, update, and delete movie. A sample output of the movie schedule is shown above

Seat Booking:

Users can select seats from a 10x10 seating arrangement. The system updates seat availability in real-time. Below is an example of seat availability before and after booking:

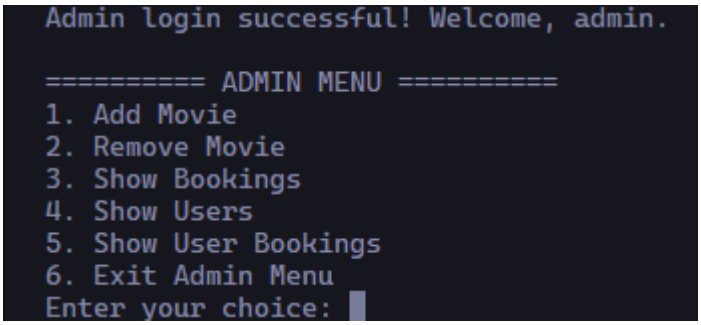


Ticket Generation:



Admin Menu

The admin menu provides all the features for admin method to how to access it



7 FUTURE ENHANCEMENT

Some of the future enhancements in the projects can be:

- **Adding User Interface:** Transition from a command-line interface to a graphical user interface using libraries like GTK or SDL to make the system more user-friendly.
- **Database Integration:** Database likes SQL , SQLite, MongoDB can be added for more flexibility in the system.
- **Multi-User Support :** Implementation of different user roles like: admin, staff, customer with varying levels of access to the system.
- **Payment System Integration:** We can use different payment processers(esewa , Khalti, PayPal) to the app for online payment

8 CONCLUSION

In conclusion, the theater management system developed in C represents a robust and efficient solution for managing the various aspects of theater operations, from scheduling and ticket booking to customer management and reporting. Throughout the project, we successfully implemented core functionalities such as seat allocation, movie scheduling, and ticket sales, demonstrating the power and flexibility of the C programming language in building structured and performance-oriented applications.

9. REFRENCES:

[1] Wikimedia, “Theater Management System,” Wikipedia.org, Apr. 21, 2011. https://fr.wikipedia.org/wiki/Theater_Management_System (accessed Feb. 07, 2025)

[2] C. to, “standard UNIX utility and programming language for build automation,” Wikipedia.org, Jan. 29, 2003. https://en.wikipedia.org/wiki/ Make_(software) (accessed Mar. ,2025).