

# **TRIBHUVAN UNIVERSITY**

## **INSTITUTE OF ENGINEERING**

## **PULCHOWK CAMPUS**

**PROJECT REPORT ON MOVIE TICKET BOOKING SYSTEM**

SUBMITTED BY: SUBMITTED TO:

Prajwal Devkota (078BEL065) Department of Electronics and

Prashant Bisokarma (078BEL067)

Pratik Adhikari (078BEL069) Computer Engineering

# **Acknowledgement**

First and foremost we would like to express our sincere gratitude towards our subject teacher Bibha Sthapit along with our lab teacher for providing us this opportunity to perform this project to enhance our knowledge and learn to code in C++ programming language. We are very grateful towards them for their assist and guidance throughout this semester as well. Along with them, we also owe our friends and seniors as of their help and support in different steps of this project.

Also we would like to thank the Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus for providing us the opportunity to develop and get a new experience of working in a team. We are grateful to our friend for the help and feedback they provided whenever we needed.

Last but not the least, we would like to thank everyone who directly or indirectly helped us in this project.

**ABSTRACT**

This project work is a course project submitted to the Department of Electronics and Computer Engineering in the partial fulfillment of the requirements for the course on Object Oriented Programming for our academic session in Bachelors of Electrical Engineering, Second Year First Part as prescribed in the syllabus designed by Institute of Engineering, Tribhuvan University.

The main aim of this project was to develop a user-friendly program using an Object Oriented Programming language, C++. For this project, we made a “Movie Ticket Booking System”, which is more like a ticket reservation system. The goal of creating this booking system is to thoroughly know the concept of object oriented programming language. We have used WxWidgets for Graphical User Interface (GUI) which is a popular C++ library that provides a framework for creating cross-platform graphical user interfaces (GUIs) and applications. It allows developers to write code in C++ while creating native-looking GUIs for various operating systems. This ticket booking system is fully functioning but still there is a lot of room for improvement. We plan to add more features, and to improve to make the project user friendly in a secured approach.

**Table of contents Page No.**

1. Introduction………………………………………... 1

2. Objective…………………………………………... 1

3. Applications………………………………………... 2

4. Literature Review………………………………….. 3

5. Methodology………………………………………. 9

6. Implementation and Results………………………. 10

7. Problem faced and solution……………………….. 30

8. Limitation and future enhancements……………… 31

9. Conclusion………………………………………… 32

10. References……………………………………….. 33

# **Introduction**

Welcome to the world of seamless movie-going experiences with our cutting-edge C++ project – the Movie Ticket Booking System. This innovative interface revolutionizes the way movie enthusiasts secure their seats, bringing convenience and flexibility to the forefront. By harnessing the power of object-oriented programming, we've crafted an intuitive platform that empowers both movie halls and customers. With features like dynamic seat selection, personalized user profiles, and secure payment gateways, our system ensures an effortless journey from movie selection to confirmed booking. Join us as we delve into the realms of C++ programming to create a user-friendly, efficient, and engaging movie ticket booking solution.

Movie Ticket Booking System is an interface that can be used by any movie hall to provide their customers with a platform to reserve or book the tickets of the movie they need with the feature to select the date and shift of the movie. It consists user registration and login for uniqueness in selection of the seats, which later can be confirmed after payment only. We will create this system using the object oriented approach in C++ language by trying to make it as user friendly as possible.

# **Objectives**

Major objectives of our project can be bulleted as follows:

* To learn to code using concepts of Object Oriented Programming,
* To explore different features of C++ programming language,
* To learn to work in groups and develop communication and problem solving skills,
* To learn to create a user friendly interactive front end interface,
* To be able to self-create an programs required in the days to come,
* To build an attractive UI for the users to help them select options available for the customers.

1. **Applications**

A Movie Ticket Booking System is a software application designed to facilitate the reservation of movie tickets by customers. A movie ticket booking system provides an interface where users can create accounts or profiles with their personal information, choose their preferred seats in the theaters.

It includes details such as movie title, synopsis, price, duration, release date, genre etc. It also provides a graphical seat map to assist users in making selections. The need of Movie Ticket Booking System is for the intuitive interferences, secure payment processing, and efficient database management.

A well-designed Movie Ticket Booking System not only simplifies the ticket purchasing process for customers but also provides valuable data and marketing opportunities for theater owners and operators.

1. **Literature Review**

**C++:**

* Introduction

C++, as we all know is an extension to C language and was developed by Bjarne Stroustrup at Bell Labs. C++ is an intermediate level language, as it comprises a confirmation of both high level and low level language features. C++ is a statically typed, free form, multi-paradigm, compiled general-purpose language.

It is an Object Oriented Programming language but is not purely Object Oriented. Its features like friend and virtual violate some of the very important OOP concepts such as data hiding, so this language can’t be called completely Object Oriented.

* Structure of code

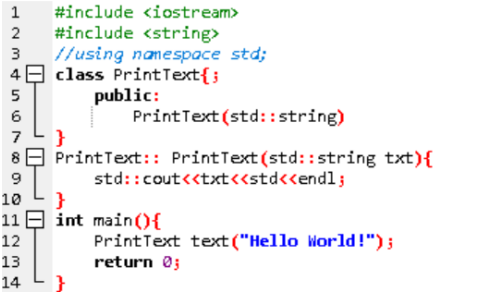
A C++ program is structured in a specific and particular manner. In C++, a program is divided into the following four sections:

1. Standard Libraries Section

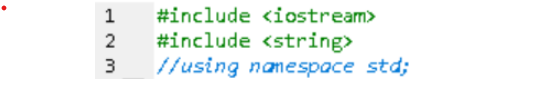
2. Class Definition Section

3. Main Function Section

For example, let us look at the implementation of the Hello World program:



1. **Standard Libraries Section**



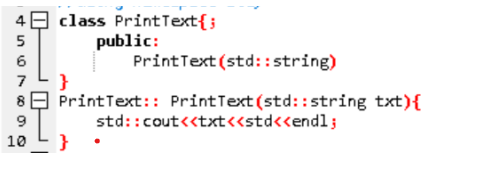
1. #**include** is a specific preprocessor command that effectively copies and pastes the entire text of the file, specified between the angle brackets, into the source code. This file is also called header file.

2. The file is merely for input output streams. This header file contains code for console input and output operations. This file is a part of std namespace.

3. The other header file is used for string related functions and is also a part of std namespace.

4. Namespace is a prefix that is applied to all the names in a certain set. For example, iostream file is defined in a set what we call std and it defines two names used in this program cout and endl.

5. If using namespace std; is used, the compiler understands that the names like cout and endl of the std namespace are being used.



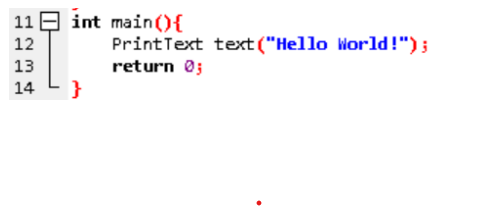
1. **Class definition Section**

The classes are used to map real world entities into programming. The classes are key building blocks of any C++ program. A C++ program may include several class definitions. This is the section where we define all of our classes. In above program PrintText is the class and text is its object.

C++ allows the programmer to define their own function. A user-defined function groups the code to perform a specific task and that group of code is given a name (identifier). When the function is invoked from any part of the program, it executes the codes defined in the body of the function.

In this program the function PrintText is a special member function called constructor of the class PrintText.

1. **Main Function Section**



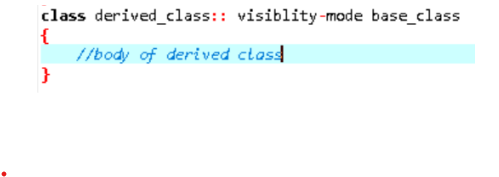
The main() function is a function called when any C++ program is run. The execution of all C++ programs begins with the main() function and also ends with it, regardless of where the function is actually located within the code.

* Features of C++:

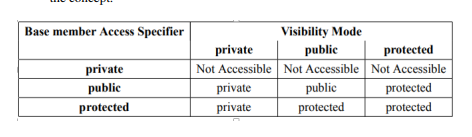
C++ is a general-purpose programming language that was developed as an enhancement of the C language to include object oriented paradigm. It is an imperative and a compiled language. Some of its main features are:

1. **Namespace**: A namespace is a declarative region that provides a scope to the identifiers (the names of types, functions, variables, etc) inside it. Namespaces are used to organize code into logical groups and to prevent name collisions that can occur especially when the code base includes multiple libraries.

2. **Inheritance**: Inheritance is a process in which one object acquires some (or all) of the properties and behaviors of its parent object automatically. In such way, one can reuse, extend or modify the attributes and behaviors which are defined in other classes, from existing classes. The class which inherits the members of another class is called derived class and the class whose members are inherited is called base class. The derived class is the specialized class for the base class. The syntax for derived class is:

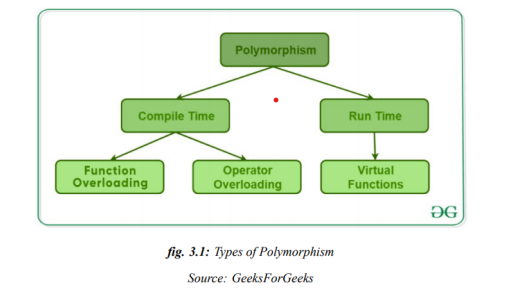


3. There are three visibility modes in which a derived class inherits from its base class. They are private, public and protected. Following table clarifies the concept:



4. **Polymorphism**: The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. Polymorphism is considered as one of the important features of Object Oriented Programming.

In C++ polymorphism is mainly divided into two types:



6. **Templates**: A template is a simple and yet very powerful tool in C++. The simple idea is to pass data type as a parameter so that we don’t need to write the same code for different data types. For example, a software company may need sort() for different data types. Rather than writing and maintaining the multiple codes, we can write one sort() and pass data type as a parameter.

7. C++ adds two new keywords to support templates: ‘template’ and ‘typename’. The second keyword can always be replaced by keyword ‘class’.

**WxWidgets:**

WxWidgets (formerly known as WxWindows) is a C++ library and framework for developing cross-platform graphical user interfaces (GUIs). It enables developers to create native-looking applications that can run on various operating systems, including Windows, mac OS, Linux, and more.

The WXwidgets lists these features:

* WxWidgets supports a wide range of platforms, including Windows, macOS, Linux/Unix, iOS, Android, and embedded systems.
* WxWidgets responds to various events like button clicks, mouse movements, and keyboard input.
* It offers a variety of common dialogs, such as file dialogs, font dialogs, color dialogs, message boxes, and more.
* WxWidgets includes advanced controls for more specialized tasks, including a rich text editor, HTML rendering, chart plotting, and OpenGL integration.
* You can extend the library with custom controls to meet specific project requirements.
* WxWidgets provides cross-platform file handling functions, simplifying tasks like reading and writing files, managing directories, and file dialog interactions.
* WxWidgets can be integrated with third-party tools like WxFormBuilder or WxCrafter to visually design and generate code for your application's interface.

# **Methodology**

The following methods were used for the completion of the project:

1. Planning and Gathering information:

We researched online and gathered various information about Movie Ticket Booking System and different libraries required for the project. The planning for the work division among the three of us was done.

1. System Model and Design:

A basic model of the game was proposed documenting the information about the project was carried out before the coding and implementation during the proposal submission. New testable features were added regularly and validated.

1. Software Development:

This project is based on C++ programming language and Object Oriented Programming concept using WxWidgets for graphical user interface which is a popular C++ library that provides a framework for creating cross-platform graphical user interfaces (GUIs) and applications. It allows developers to write code in C++ while creating native-looking GUIs for various operating systems

1. Testing and Implementation:

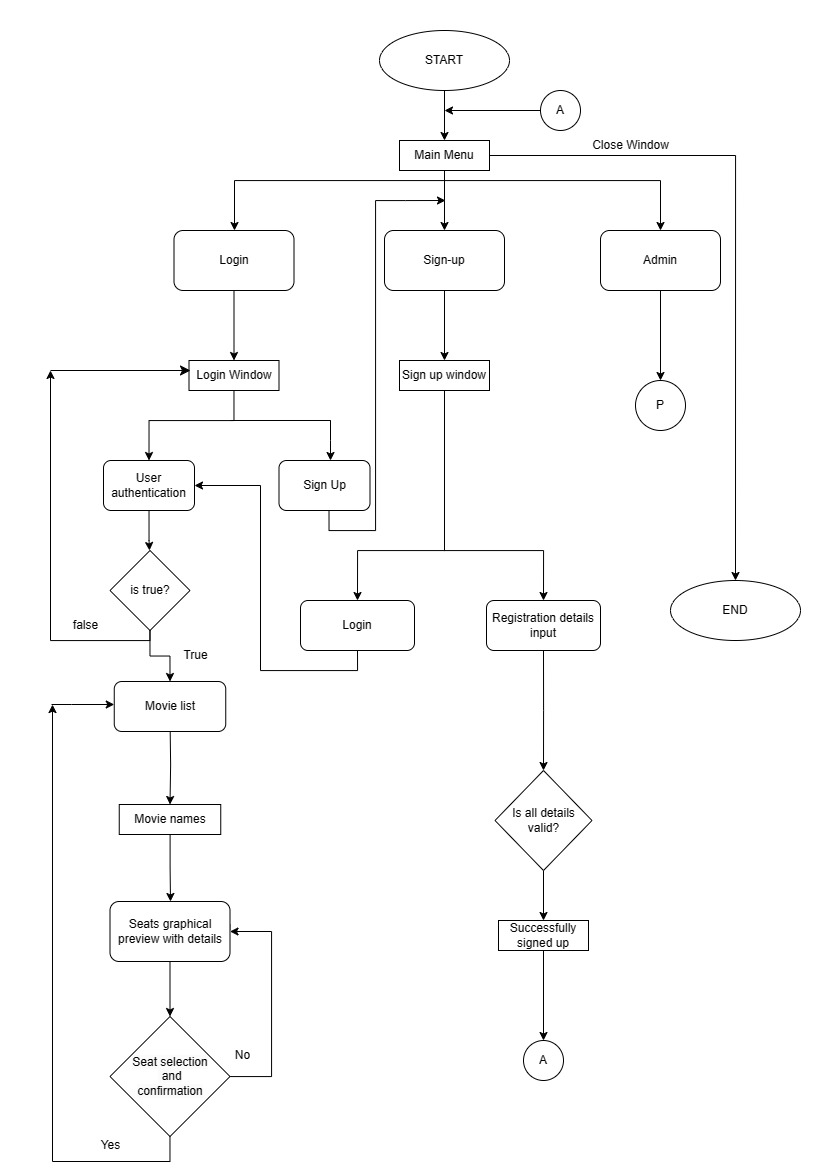
The program went through testing to measure usability, functionality and performance. We used git for version control and GitHub for sharing of code among our team members.

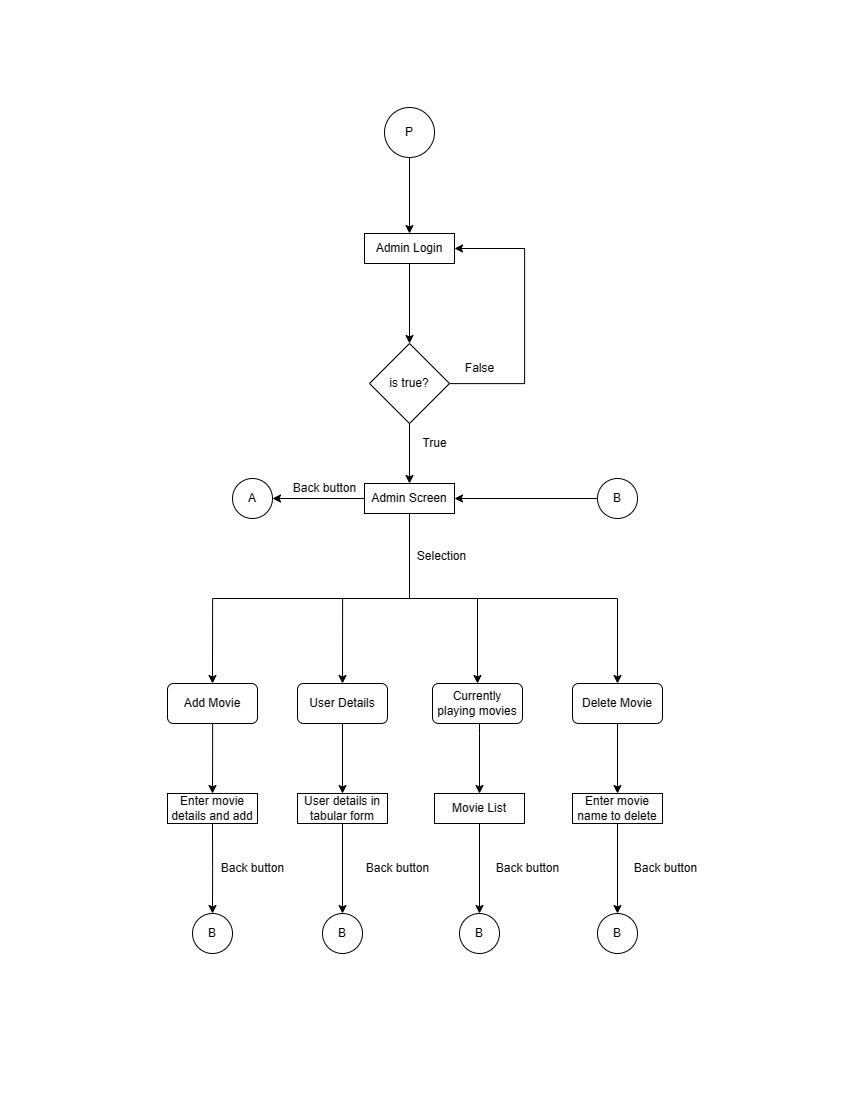
Adhering to the style of OOP, different classes were made with suitable access specifiers and data hiding was given a priority by making the members private as much as possible. The concept of code re-usability, data abstraction were implemented in the project. We have created an object for each of the elements of our game and then worked on their interaction with each other.

1. **Implementations and Results:**

The code for this project can be accessed through GitHub repository of the project: <https://github.com/pra7ik7/movie_ticket_reservation.git>

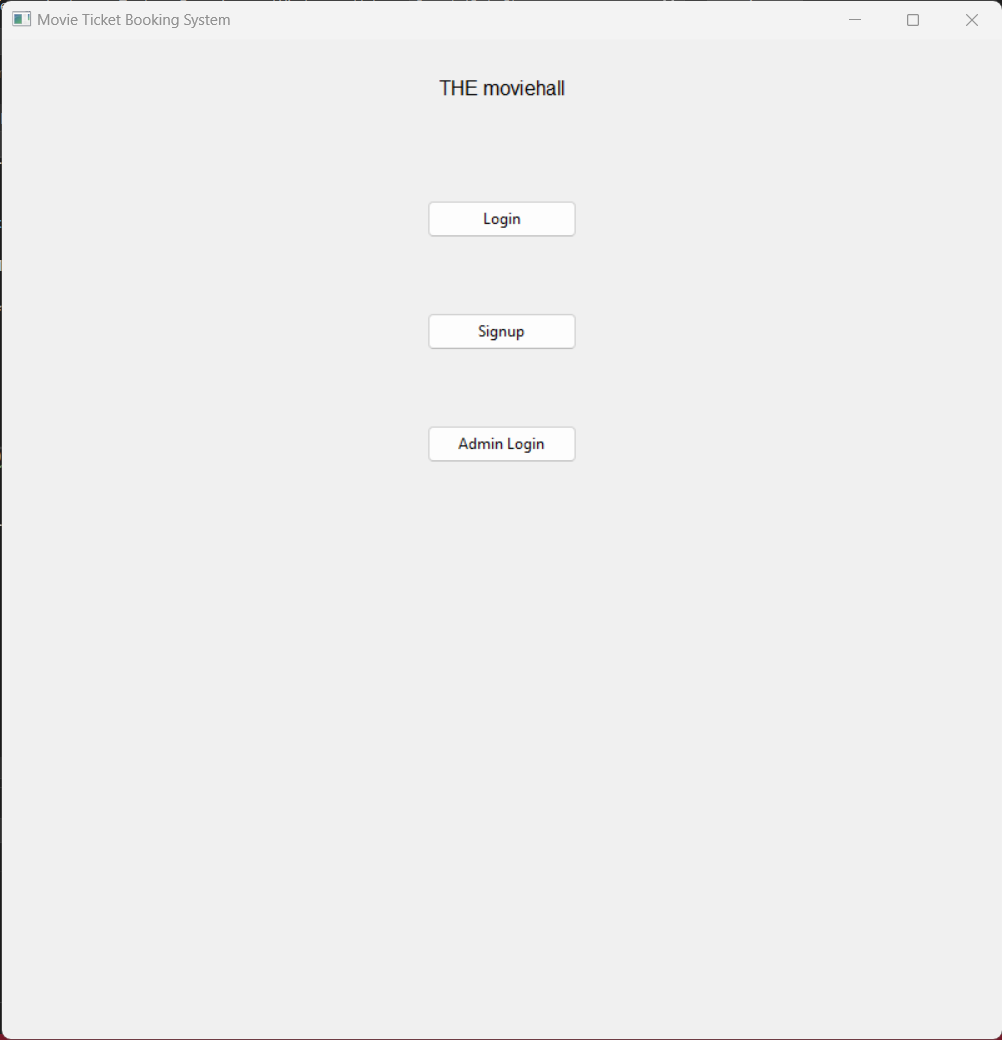
**Block Diagram**

****

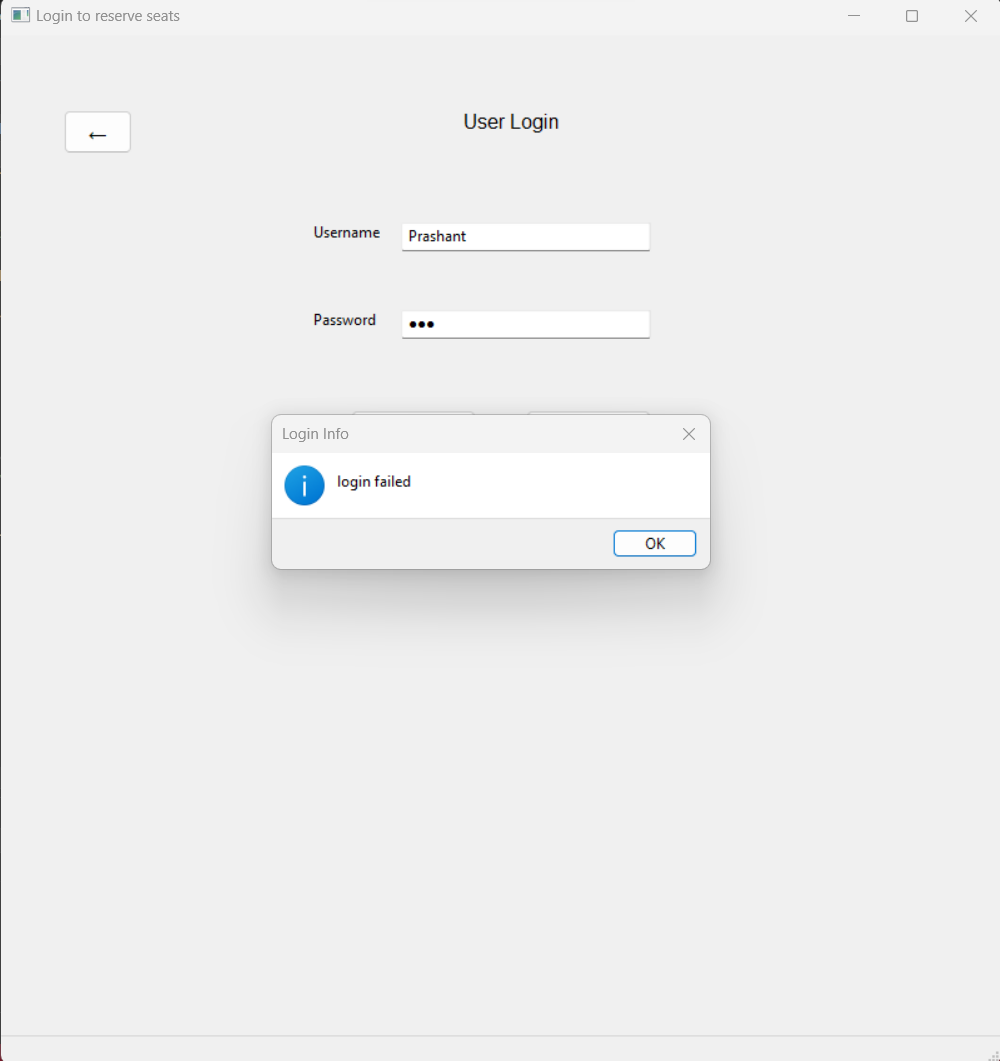


**Now, the results:**

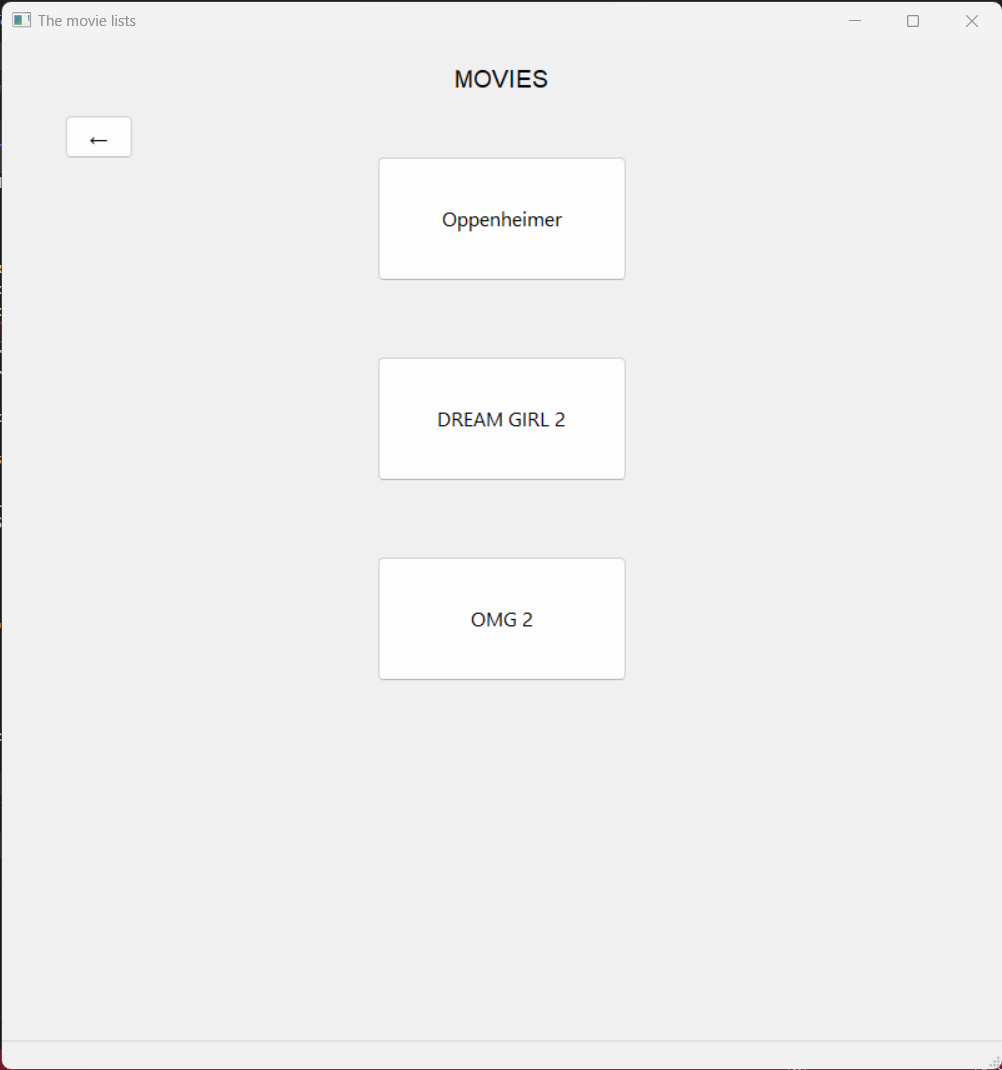
1. The opening UI looks as in screenshot below. It consists of Login, Signup and Admin Login button that takes you to the next window accordingly.



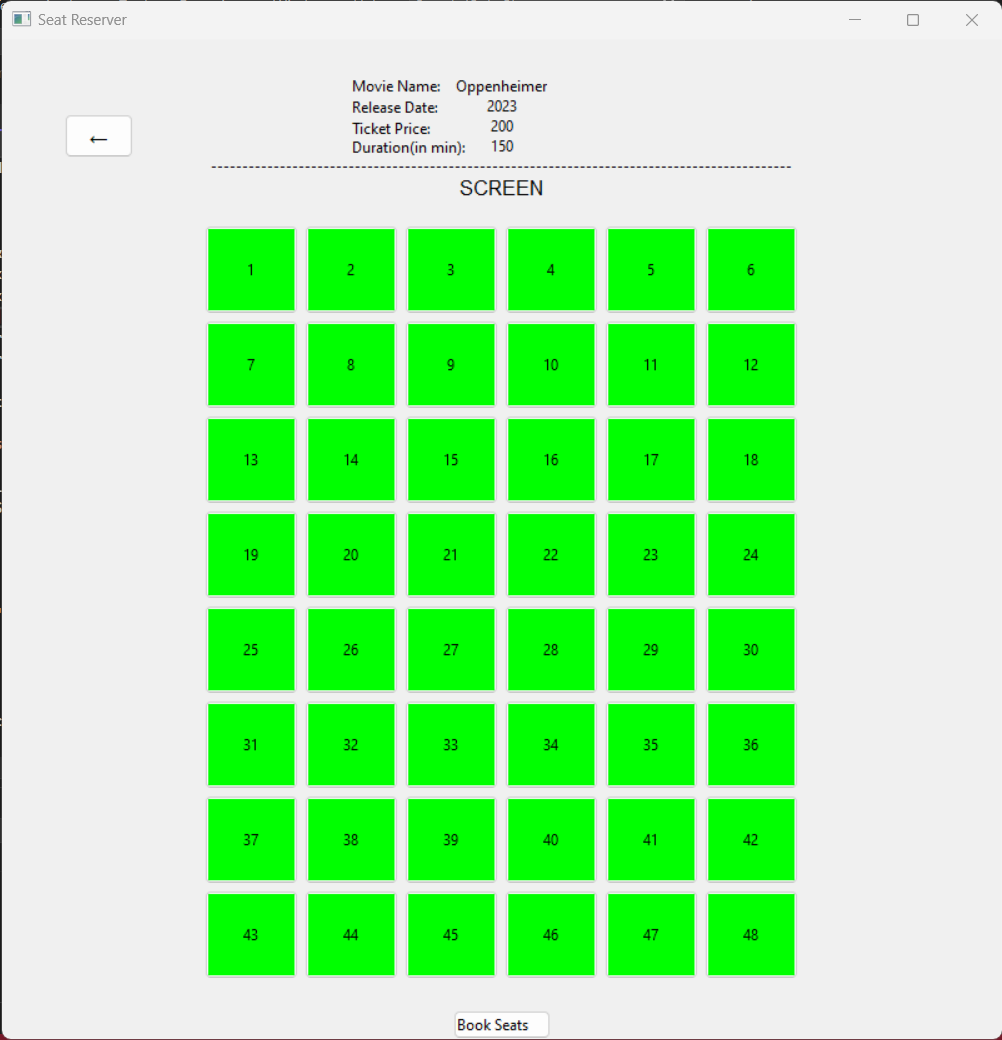
**1.1a** With the first option selection, we enter to the login frame, where we need to enter a valid username and password as per the details entered during one’s registration, or it results a “login failed” message display.



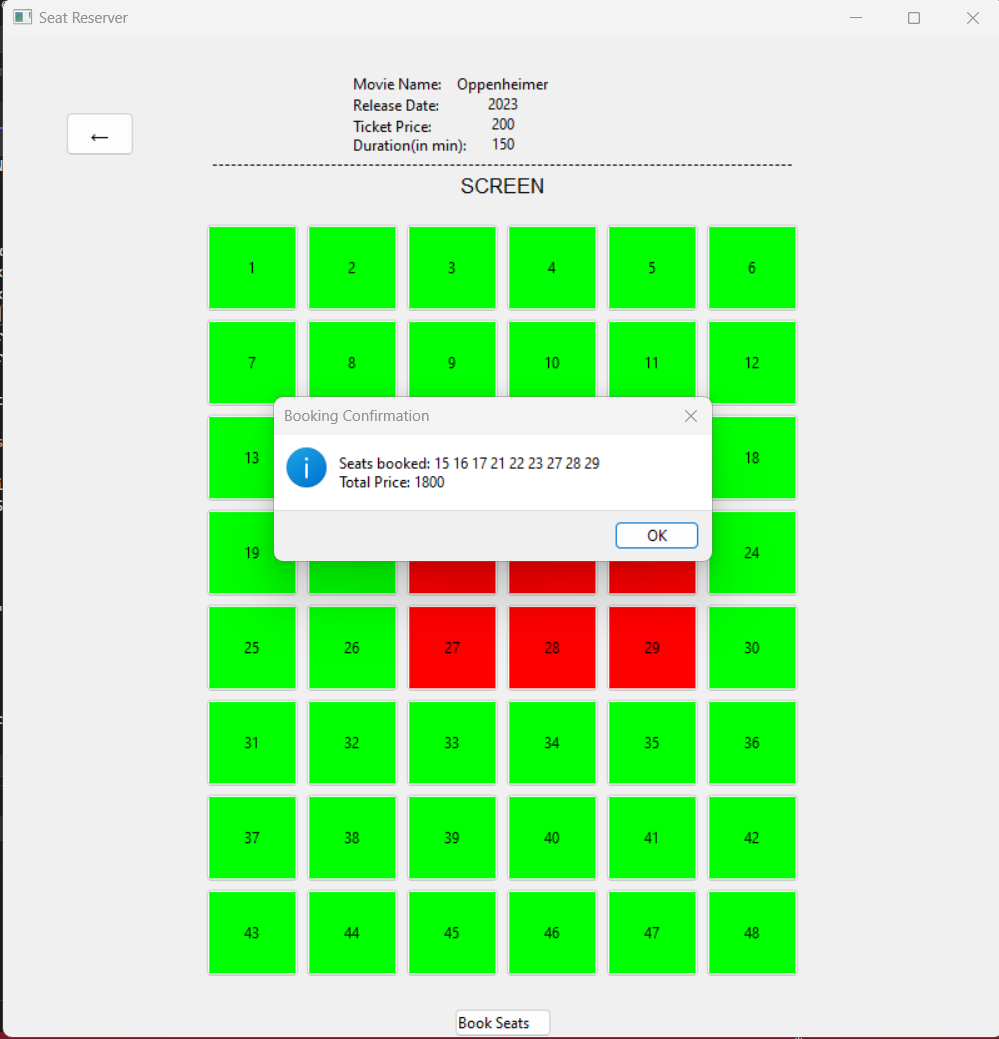
**1.1b** The successful entry for user results the Movies List window that contains the list of movies, which can be clicked in accordance of your choice to reserve the seat(s).



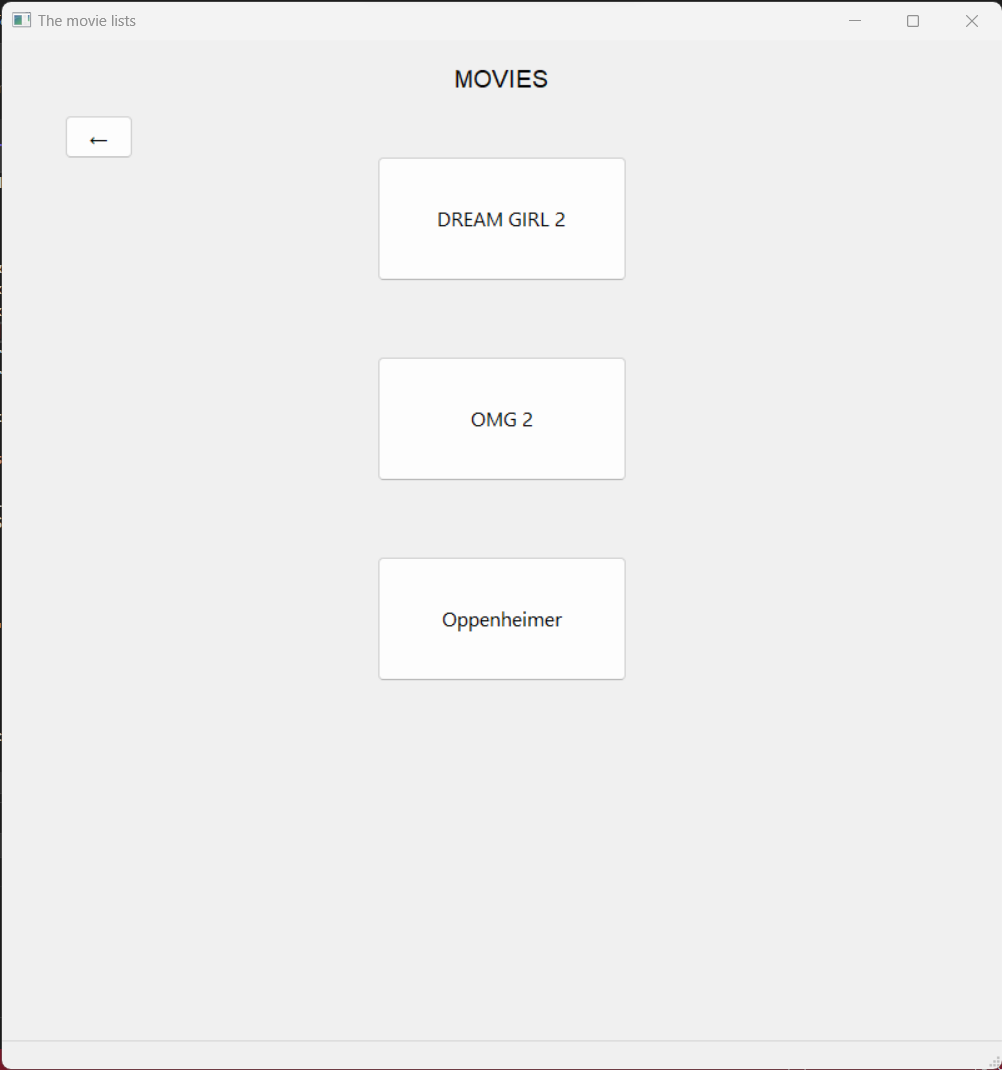
**1.1c** The following image shows the display of graphical view of seats implemented using WxWidgets. The sample movie “Oppenheimer” is newly added and none of the seats have been reserved yet.



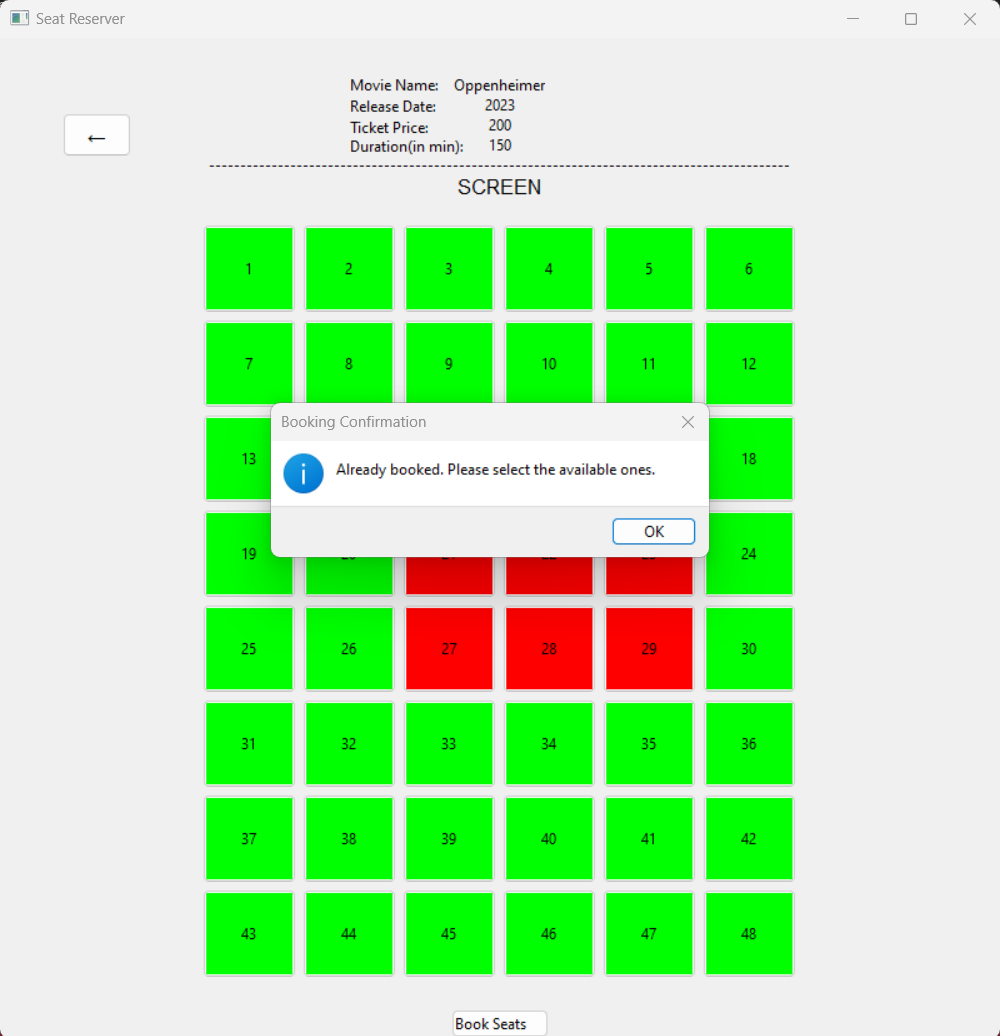
**1.1d** We can select the seats as our wish and click the “Book Seats” button present at the bottom of the window. The message box displays the seat numbers and total cost that would ultimately be charged to the users.



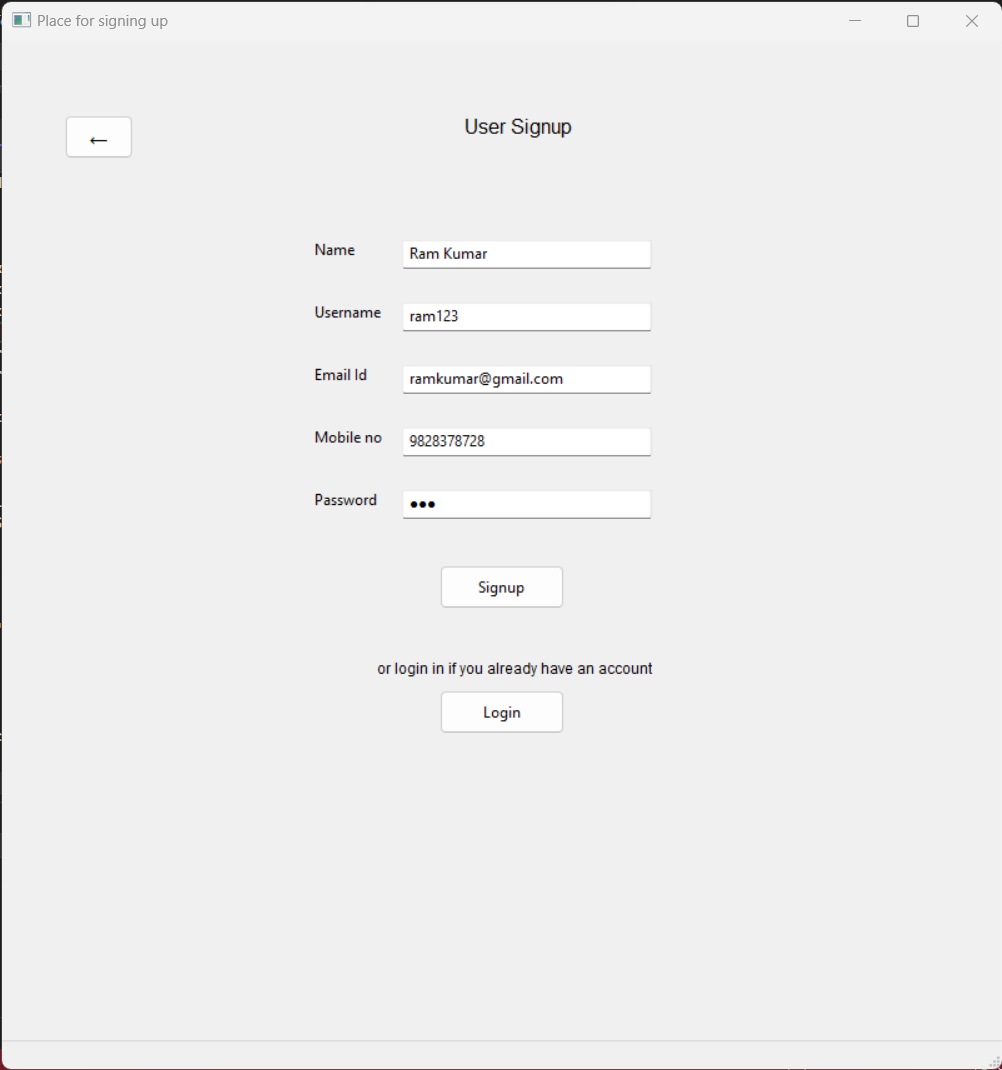
**1.1e** The shot below is the photo of redirected window just after clicking “Ok” in the above message box.



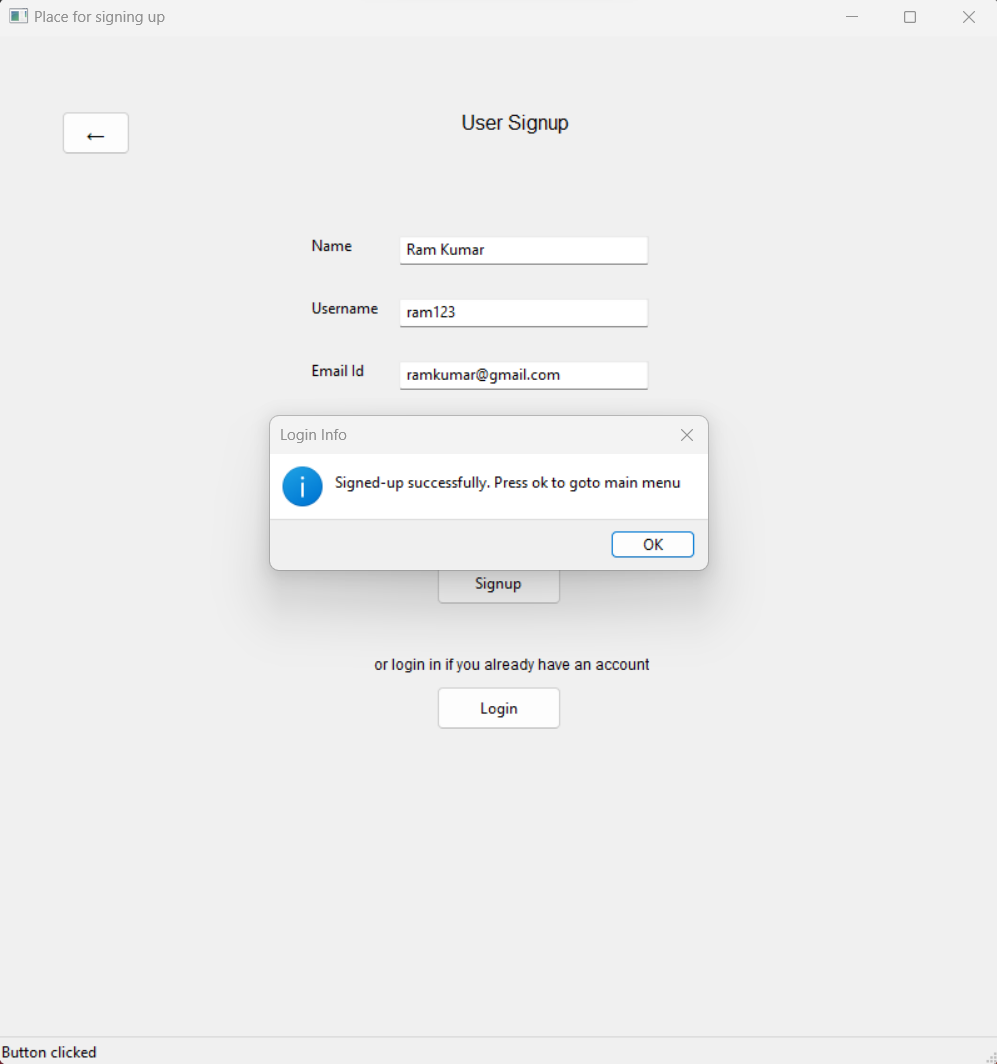
**1.1f** On again going to the movie “Oppenheimer”, we can see the seats being booked and on re-clicking these reserved seats, an “Already booked” message appears.



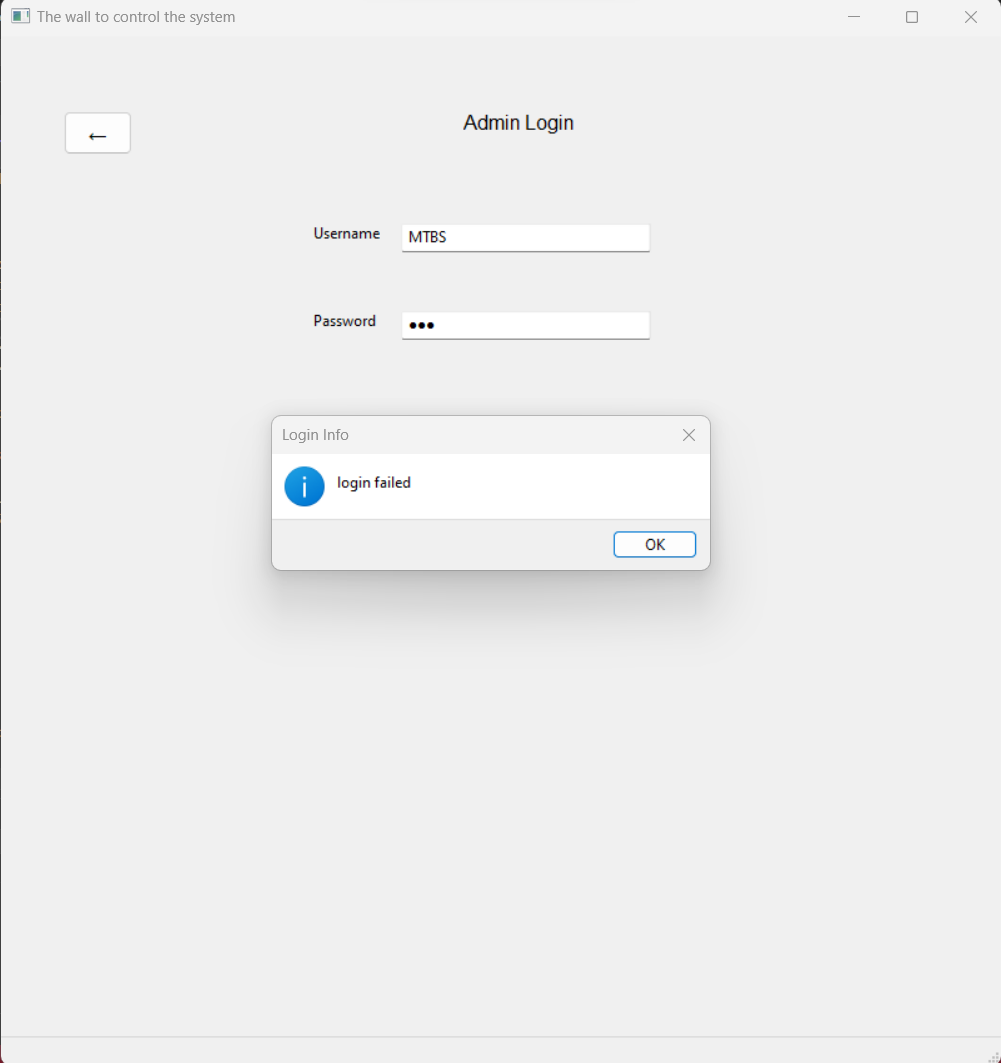
**1.2a** In case of new registration, user need to go to signup page either directly from main menu or from login page.



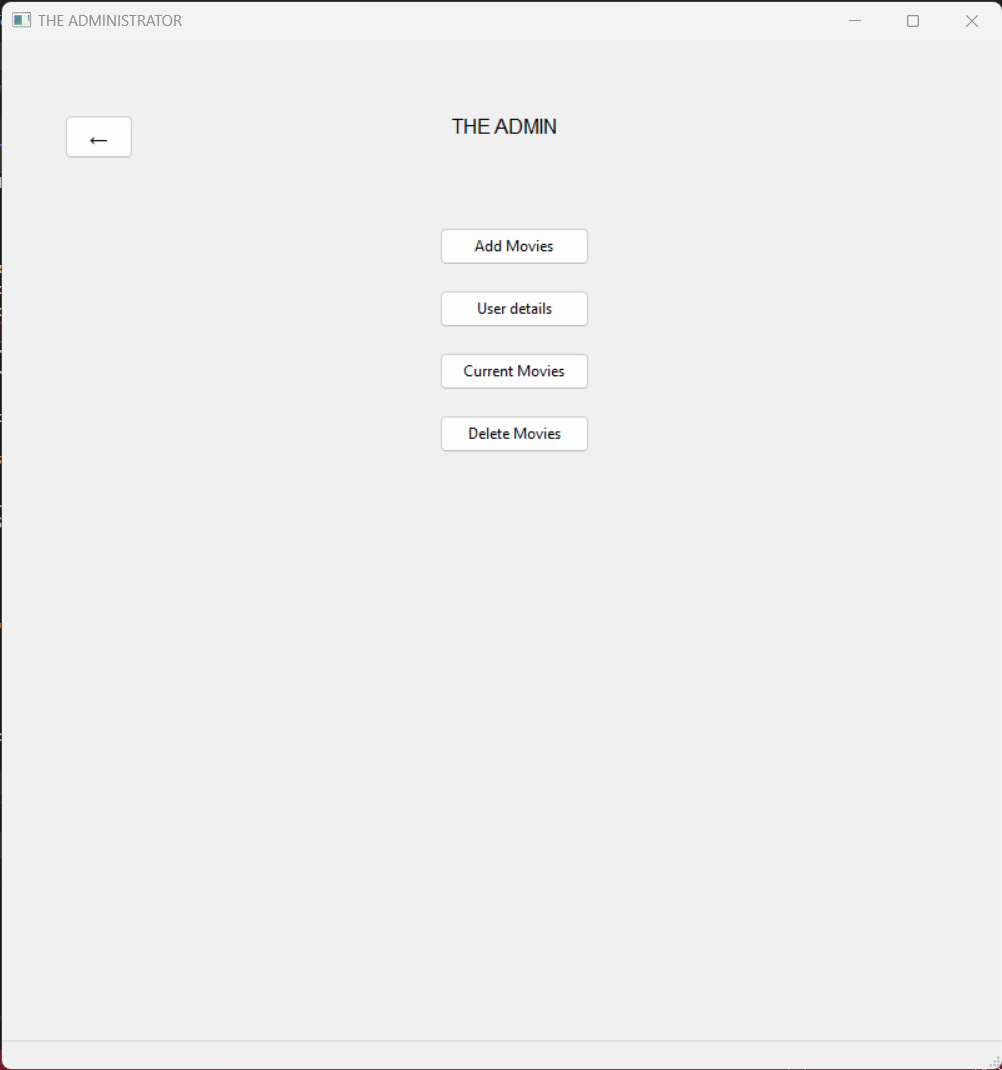
**1.2b** On entering every details, the sign up ends up successfully with a message displaying the “successfully signed up” message.



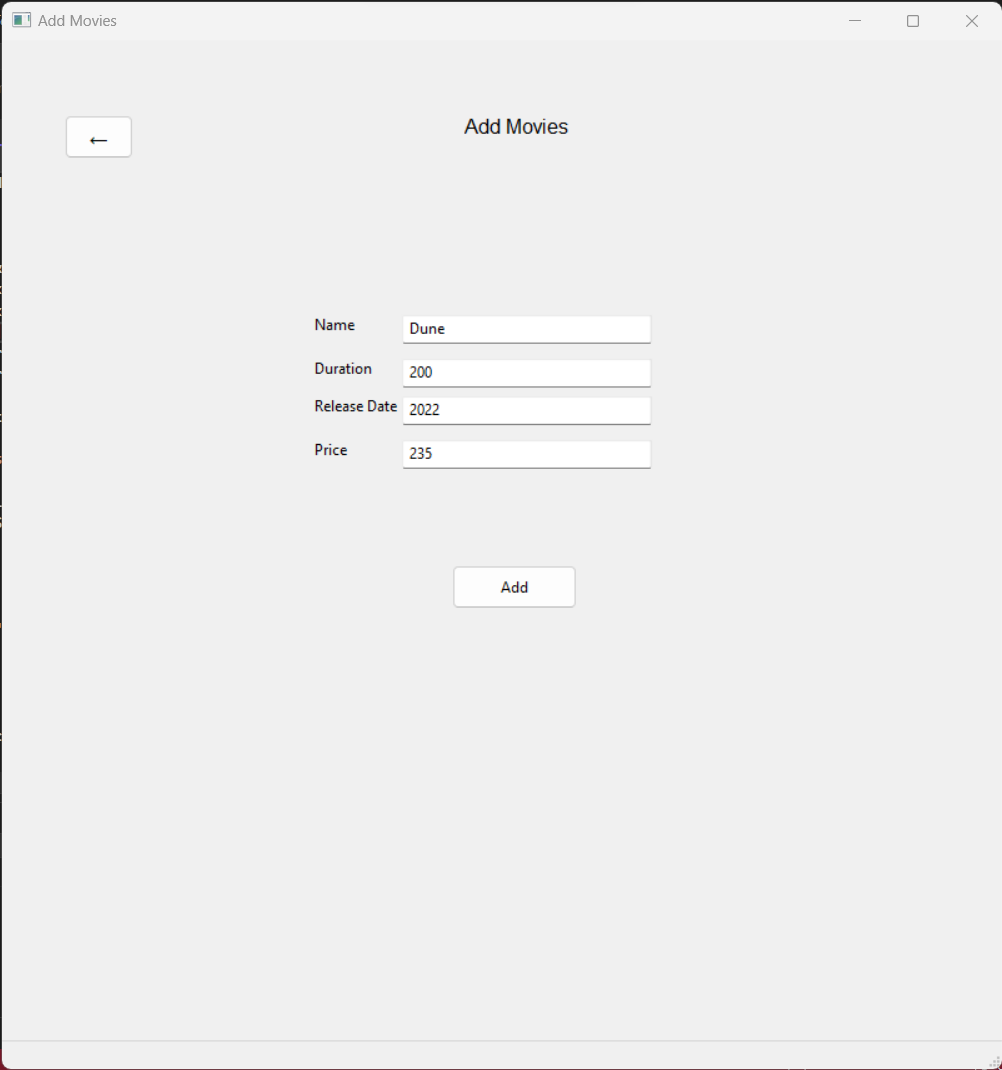
**1.3a** In case you select the Admin Login from the main menu, a window for entering the login details appears. In case of wrong password, the login fails.



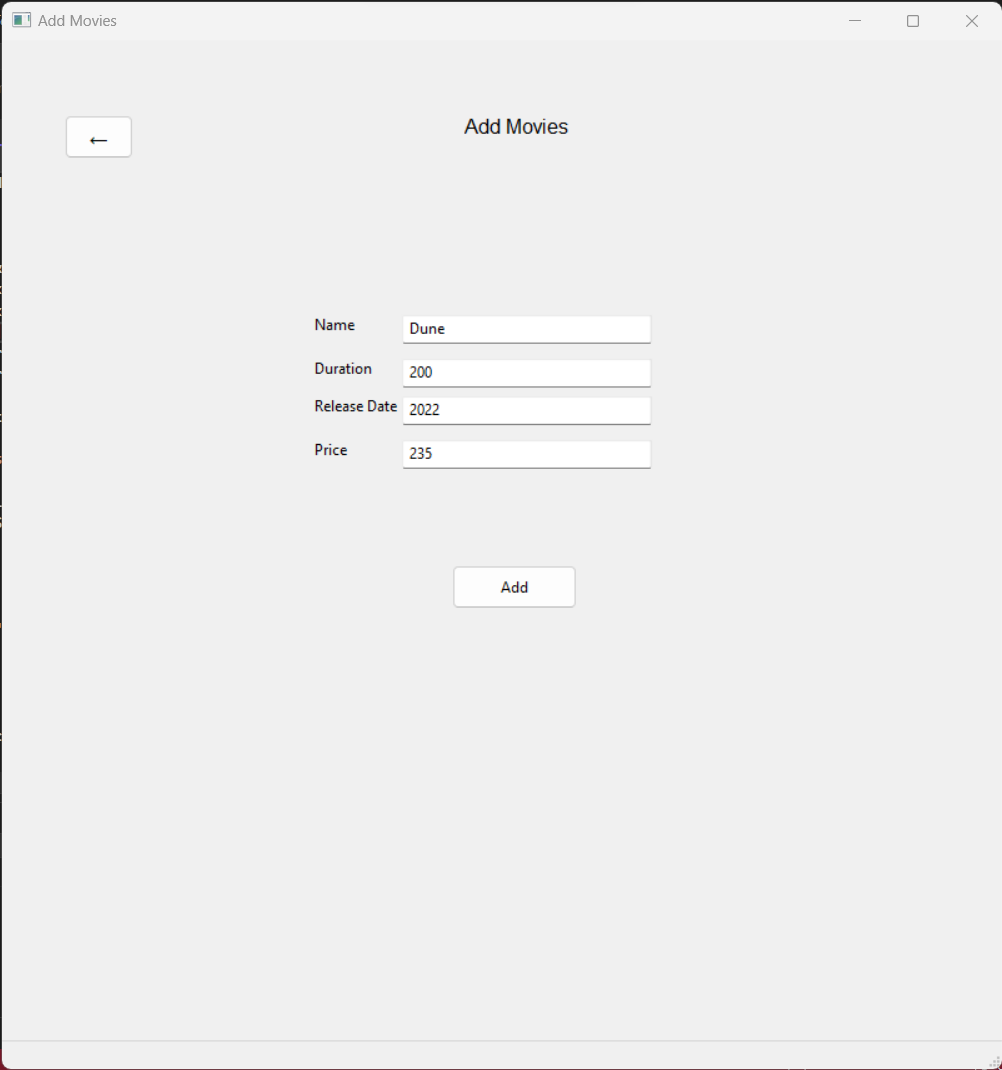
**1.3b** With the correct admin name and password input, you will be redirected to admin screen as shown below.



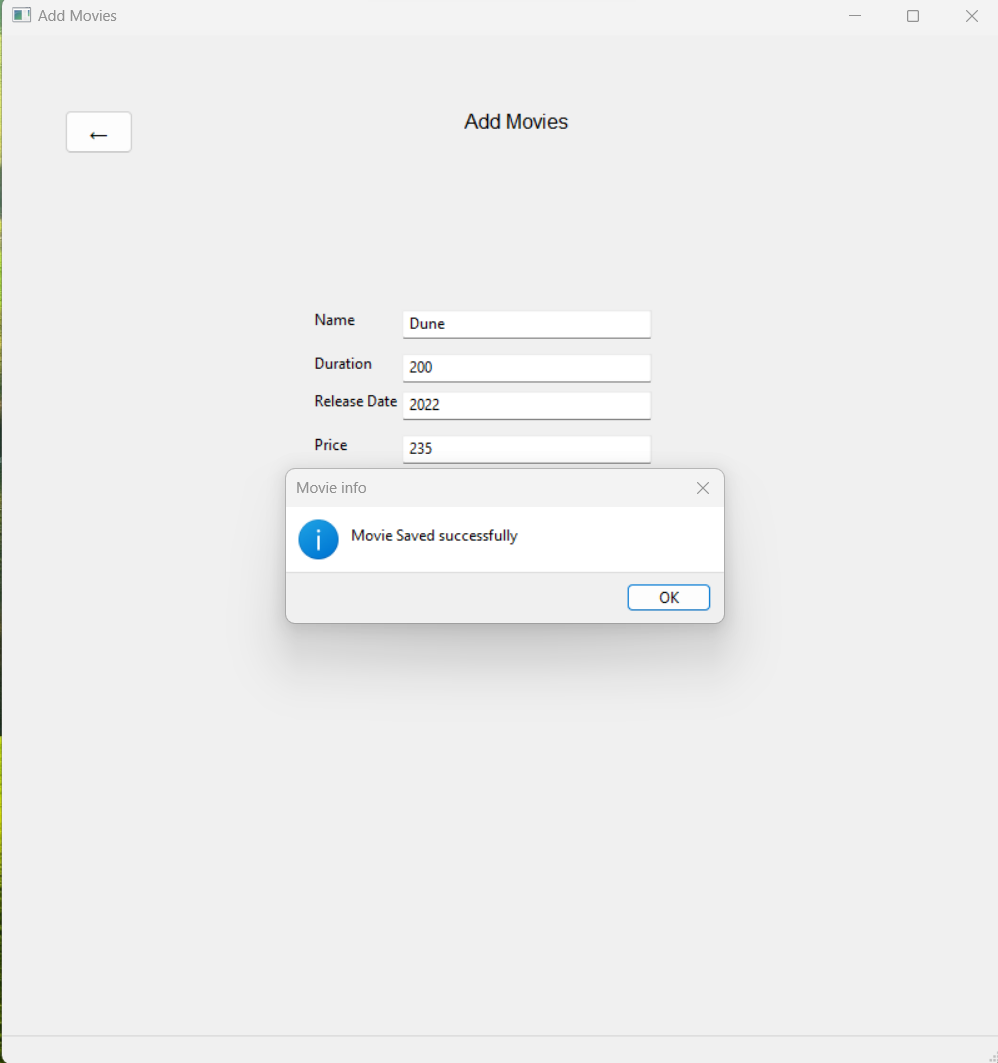
**1.3c** By selecting the first option present over the admin screen, you can add the movie as follow. The snapshot shows the addition of movie named “Dune” along with some of its other details.



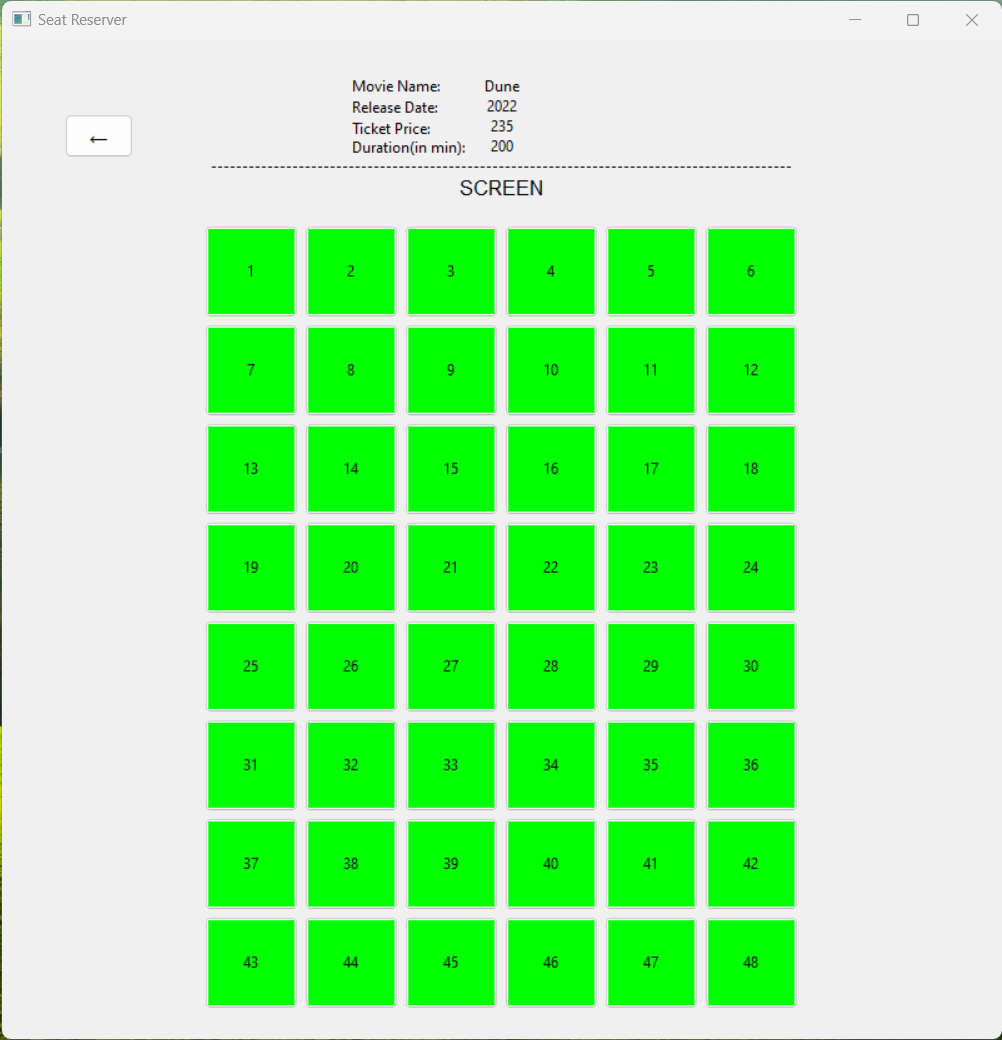
**1.3d** The addition process of a new movie is shown below.



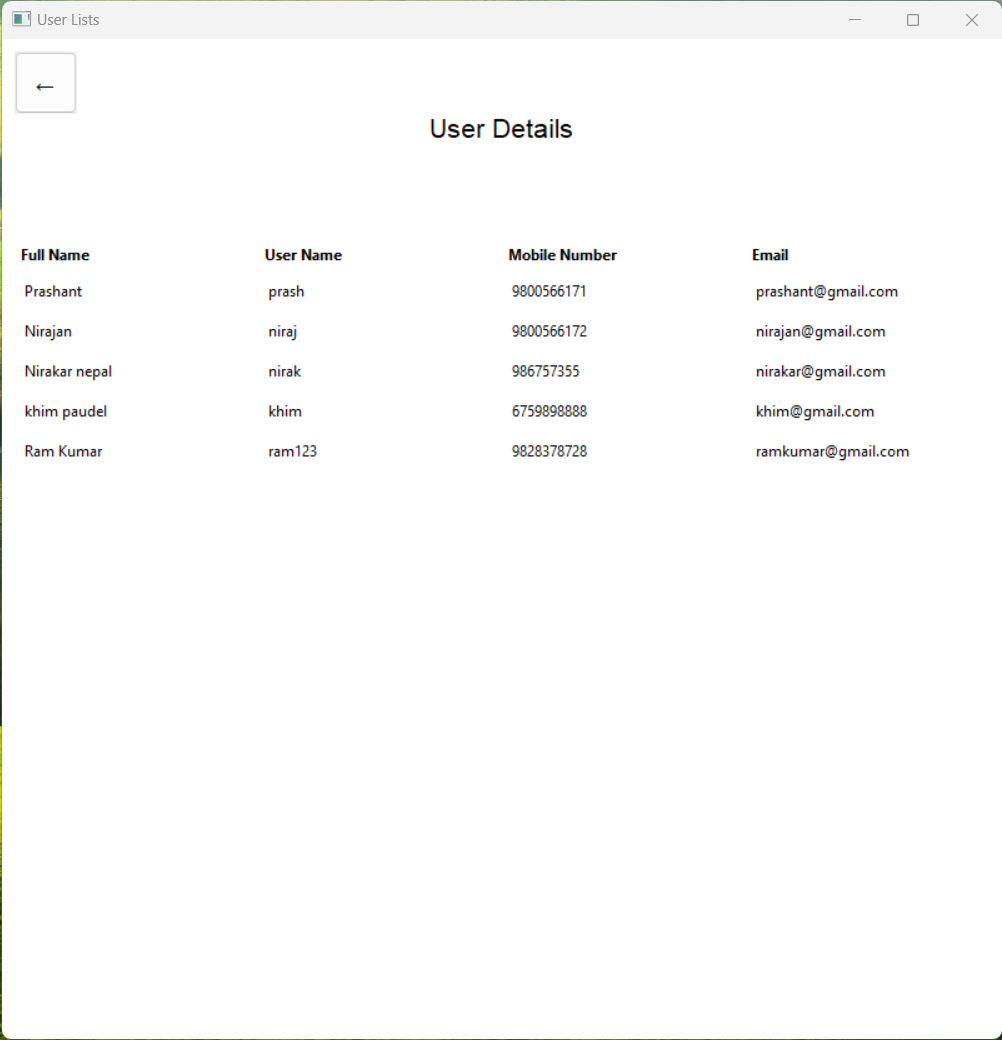
**1.3e** To confirm the addition, “currently playing movies” button is clicked in the admin screen where we can see the movie “Dune” in the bottom.



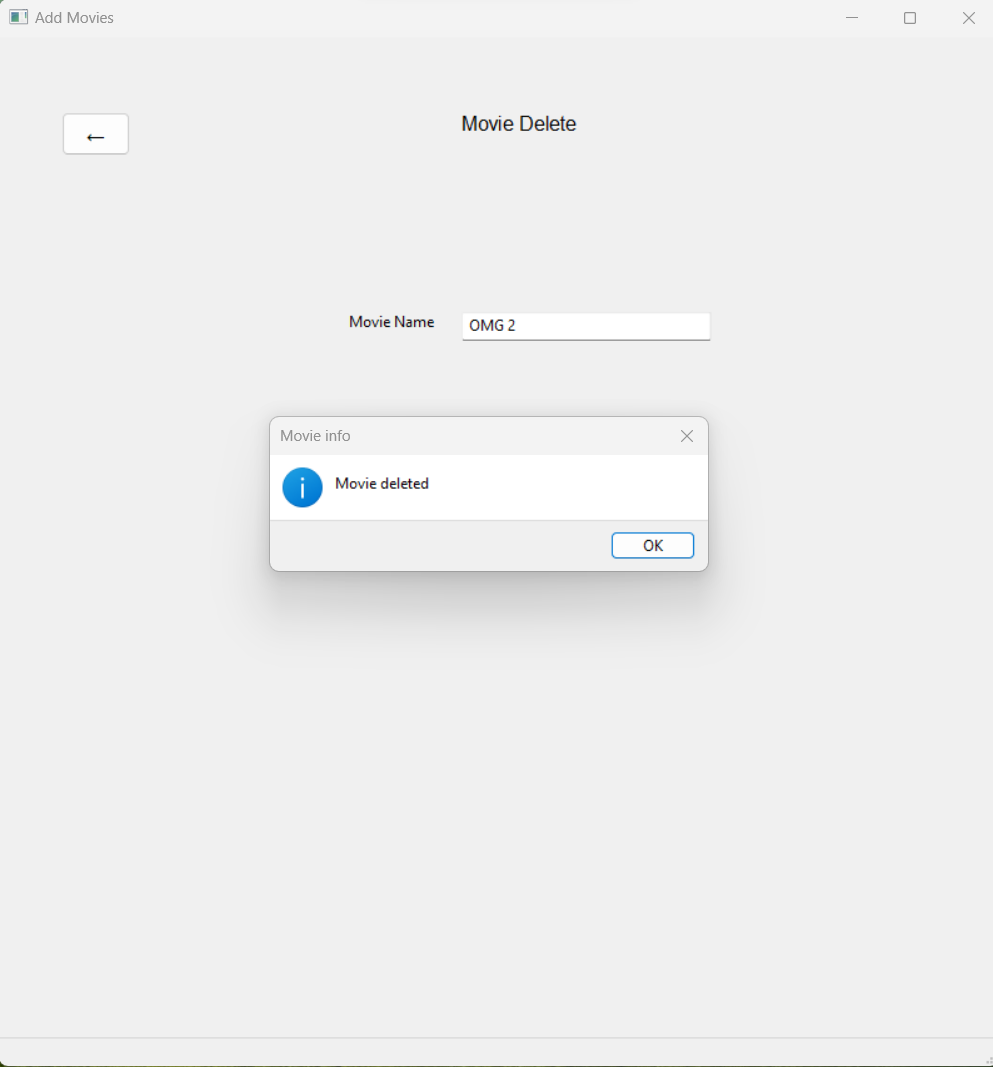
**1.3f** The newly formed movie “Dune” has all seats made available as none of the user has reserved any. This display of seats in admin section is only for display of the availability of the seats unlike in the “users” window where one can book seats as per their wish.



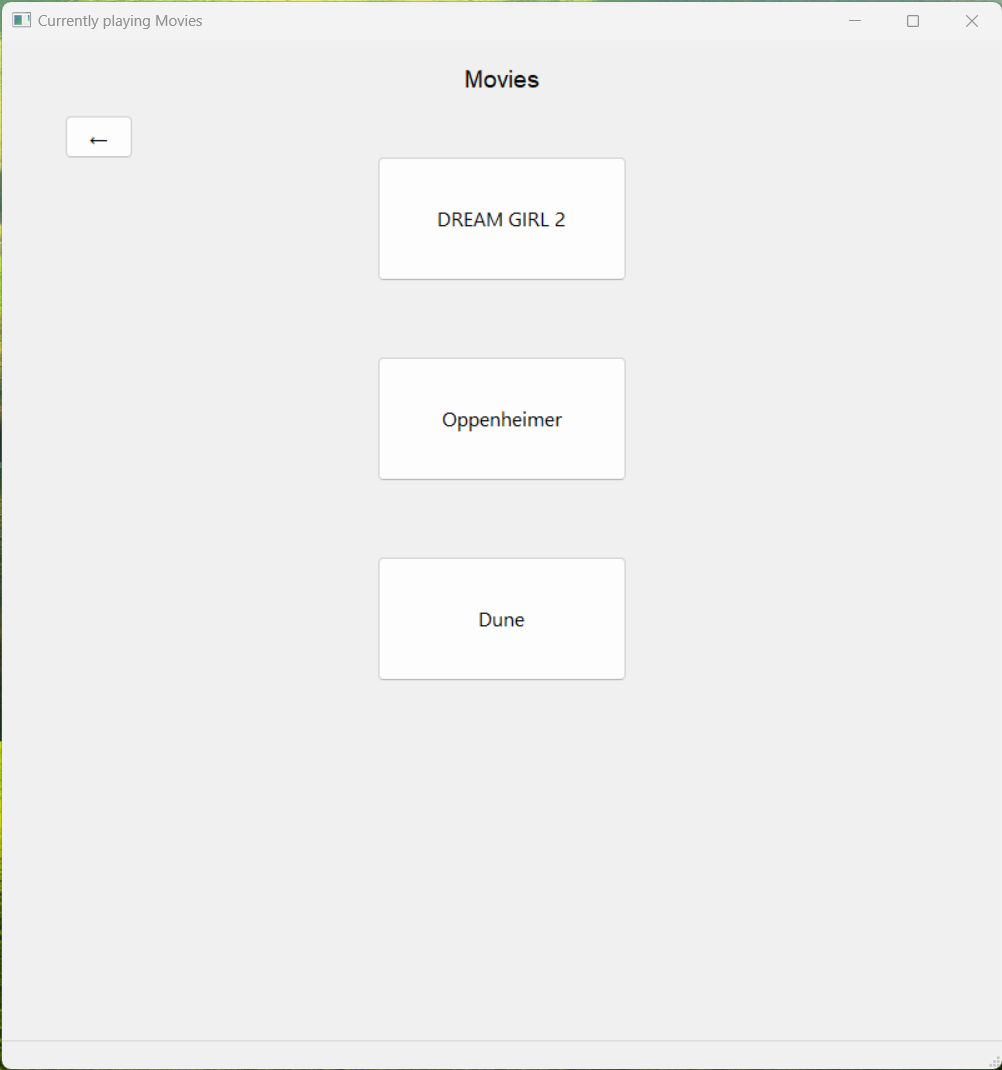
**1.3g** The “User details” button takes the admin to the page that shows the user details that have signed up and their details are previewed as follows.



**1.3h** To delete a movie, admin can simply click “Delete Movie” button. The window of movie removal takes the movie name as input. As shown, “OMG 2" has been entered to be removed.



**1.3i** After deleting the movie above, we can see the movie “OMG 2" has officially been deleted from the theatre.



**7. Problem faced and solution**

Developing a movie ticket booking system in C++ can be a complex project, and there are several challenges that you might encounter during the development process. Here are some common problems we faced and tips on how to address them:

* Data Management:

Managing data such as movie listings, show times, and seat availability can be complex. We used data structures like arrays, vectors, or linked lists to organize and store data efficiently.

* Error Handling:

Managing errors gracefully and providing informative error messages to users.

So we implemented error-checking mechanisms and use exception handling to catch and handle errors.

* Security:

Ensuring the security of user data and transactions. So we implemented security best practices, such as input validation, data encryption, and access control to protect user information.

* Scalability:

Handling a large number of users and transactions as your system grows.

We designed our system to be scalable from the beginning. For this we use load balancing and distributed database systems.

* Integration with the Backend:

Integrating the WxWidgets GUI with the backend logic of your movie ticket booking system. We define clear interfaces and communication channels between your GUI and backend components.

**8. Limitation and future enhancements**

A movie ticket booking system implemented in C++ can be a valuable tool for cinema management and customers. However, like any software system, it has limitations and potential areas for future enhancements. Here are some limitations and ideas for future improvements:

Payment Processing:

* Secure payment gateway integration for ticket purchase.
* Support for various payment methods like credit/debit cards, digital wallets, or cash-on-delivery.

Promotions and Discounts:

* Apply discounts, promo codes, or special offers to ticket purchases.
* Implement loyalty programs to reward frequent users.

Notification System:

* Send email or SMS confirmations of bookings.
* Reminders for upcoming show times.

Data Analytics:

* Collect and analyze user data to gain insights into customer preferences and behavior.
* Use data to optimize movie scheduling and marketing efforts.

Real-Time Updates:

* Implement real-time updates for seat availability, show times, and promotions using technologies like Web Socket or server-sent events.

**9. Conclusion**

In this project, we successfully designed and implemented a comprehensive movie ticket booking system using C++ and the WxWidgets framework to create an intuitive graphical user interface (GUI). The primary goal of this project was to develop a practical and user-friendly application that streamlines the process of booking movie tickets, enhances the customer experience, and provides efficient management tools for cinema administrators.

We learned how powerful the object oriented paradigm is, with the help of key concepts like inheritance, polymorphism, encapsulation and abstraction. We were able to create interfaces that made the coding experience much cleaner.

We also learned a great deal about collaboration and team work from the project. We also learned various industry tool like *Visual Studio 2022* for the graphical user interfaces, *LATEX* for typesetting documentation, *Draw.io* for making flowcharts.

The project taught us a great deal about the development cycle, including planning, analysis, development, testing and debugging. We learned the power of Object

Orientated Programming paradigm in making efficient software.

As we conclude this project report, we reflect on the valuable knowledge and experience gained throughout the development process. This movie ticket booking system represents a tangible application of our programming and GUI design skills

**10. References**

**Book References:**

* “The Secrets of Object oriented Programming”, Daya Sagar Baral, Diwakar Baral
* “The C++ Programming Language”, Bjarne Stroustrup
* “C++ How To Program” , Paul Deitel, Harvey Deitel
* “Object Oriented Programming with C++” , Robert Lafore

**Web References:**

* WxWidgets tutorials from : <https://www.youtube.com/watch?v=ONYW3hBbk-8&list=PLFk1_lkqT8MbVOcwEppCPfjGOGhLvcf9G>
* Different code lines with its debugging, and theoretical references from: [ChatGPT (openai.com)](https://chat.openai.com/)
* [GeeksforGeeks | A computer science portal for geeks](https://www.geeksforgeeks.org/)