**Book-Your-Movie-Project-Edyoda-**

A python project which guides the user to book tickets in a cinema.

**Let's talk about the program here!**

I have made the entire class Cinema which contains all the methods required to complete the basic functionalities of this program! We ask the user to input row and column values as the program initializes! Let's see how each method corresponds to the options in the interactive menu!

**1st option - Show the Seats**

If the choice entered is 1, we call the object method *ShowSeats()*.

Now this method checks if any seats have been booked previously or not. If they have we will mention that seats have been booked previously and call protected *seat\_matrix()* method. This *seat\_matrix()* method prints the cinema matrix and checks if seats have been booked previously, if they have, it displays a **B** instead of a **S**.

If no seats have been booked previously, then the *ShowSeats()* method will ask for the required number of rows and column for the matrix and call *seat\_matrix()* to display the vacant theatre!

After the seat matrix is printed, the function exits and *obj.menu()* is called and the user returns to the main menu.

**2nd option - Buying the tickets and storing details in a dict.**

If the choice entered is 2, we call the *Tickets()* method which guides the user to book tickets.

We ask them how many tickets would they like to book and save the input in a list format. For eg, if they enter 5, we save it as [1,2,3,4,5] in the global list nr\_of\_tickets.

Next, we call the protected *book\_tickets()* method which takes the nr\_of\_tickets list and loops through it. We ask the user for their choice of row and column number, first we check if the row and column number exist in our cinema, and then we check if their entered choice of row and column are not previously booked,if they have been booked previously, we call the function again after printing an exception, this time only passing the ticket no (as a list) for which the exception occured. We display the ticket price and ask the user to confirm if they would like to purchase!

After successfully booking the tickets, we need to take the user's details. We call a *get\_details()* method in our *book\_tickets()* method. *get\_details()* method loops through ["Name", "Gender", "Age", "Phone Number"] and takes an input from the user, saving the list elements as keys and user inputs as values. Then we add the row number and column number along with the ticket price (called by the method *ticket\_price()* ) to the details dictionary. We add each dictionary (each ticket booked) into a list. And we print out all the details for the user once.

We then call the *seat\_matrix()* function once more and show the user their booked seats!

**3rd option - Statistics**

If the user's choice is 3, we call the *Statistics()* method which displays the required statistics.

We use *ticket\_price()* to get the ticket price and then we print the rest of the statistics. Total income is calculated by the method *total\_income()*

**4th option - Show booked ticket user info**

If the user's choice is 4, we call the *User\_info()* method which first checks if any details dict has been created and appended to the details\_list, if not we print that no tickets have been booked as of now. If tickets have been booked, we ask the user to enter the row and column number they are looking for. After they enter, we check to see if the row and column number exist in our list of dictionaries and if we have a match then we return the details of that particulat dictionary. If there is no match, the loop exits and prints that these tickets have not been booked!