Name : Naga Venkata Shalini Chitta 700 Number : 700735190

**About the Application:**

This application is for booking movie tickets. In this application, we used ASP.NET Core to create this project and this project contains Five models, Three controllers, and six views as per the application properties.

**CodeFlow:**

* Firstly, when we open the application in the web browser it takes us to the home page.
* When the user selects the input it’s the post accessor that helps with the display, post and get accessors are executed with the help of the controllers for displaying the content. In this project, the Home Controller is where we start.
* Then the Home Controller decides to display data according to the user input if the user selects the home, it shows the content on the home page which takes us to the Index View in the application and when the user selects About it takes us to the About View.
* Events: To execute this section we have an Events controller which presents the data. In this controller, we have an action method called Event List. In this controller, it redirected to the Event Service Model which presents the list of events and related categories of the events. When we click on the events it takes us to the details of the event, we selected with the help of the Details View present in the application.
* When we click on the following link it takes us to the Buy page which represents the data presented in the Cart Controller. In this Controller, we have two action methods: Buy and confirmation. When they select to buy the ticket, it redirects to the buy page which represents the data present in Buy Model and the following Buy View, and when they conform it takes them to the confirmation page which again represents the data present in the confirmation Model and confirmation view.

**Explanation of the Model, View, and Controller:**

* Category model has the Category class which creates a type of categories, and each category has two properties: CategoryID and CategoryName
* Events Model has the Events class which creates the type for events and each event has the event name, description of the event, the category of the event, and an image to display.
* EventsService Model has the following class which helps us to create a list of events with each event of the type of event class and also creates a list of categories with each category of the category class this model contains three methods that help us to get the data according to the properties of each class.
  + - * + GetEvents () which returns the events based on the incoming parameters category.
        + GetCategories () which returns the list of categories and the events.
        + GetAllEvents () which will return all the categories.
      * ListViewModel has listviewmodel which helps to get the list of categories and selected categories and it also has the constructor which helps us to create the instance of the events, categories and selected categories with no retruntype.
      * Buy Tickets model is the representing model for the buy view which is a form that helps the class. And this class has an overloaded constructor with two sub constructors: a default constructor and one with parameters for the event name and buying and ticket price. Default parameter less is used as the binding model in the following controller and one with the parameters is called from the cart controller ‘s buys action method. Also, this model contains two methods which are Calculate Discount and Calculate Amount Due which help to calculate the total amount and the discounted amount and also the amount due.
      * Home Controller has two action methods which are Index and About.
      * Events Controller has the class called event controller class and Event List as the action method which instantiates the event service class and it uses to get the events and even if there is a value or none and based on the id of the event. And we used the if-else loop to get the events based on the category id or the id which the user is asking for and we used foreach to call the date from all the events and categories.
      * Cart Controller has the following action methods Buy and confirmation and Buy action method gets the id of the event according to the user input and by using the event services we can get the objects representing the events and if all the validation are valid in the confirmation method it calculates the amount due and returns the model view or else it redirects to the buy view.
      * And all the view represents the information based on the model and controller. We have the following view based on the models in the cart folder we have Buy and Confirmation view, In the events folder we have the Details and EventList view which represents the data about the event present in the theater and give the details of each event with the data present in the model and Home folder has the Index and About view pages which are the default view pages of the application.