Hall Budget Problem

Mr. Gilbert is an event organizer. He wants to book a hall during the Amusing Carnival for his event. There are many halls available for the event, but he is unaware of the cost per day for a hall. He wants to find the list of halls that are available within his budget, but the online application to book the hall did not  have this feature. So his request, Mr. Jones the owner of a hall to develop an application which list the halls that are available for the given budget.  
  
Jones seeks your help to create a web page to find lists of halls within his budget from the list of halls available.

Consider the class **Hall.cs** with the following attributes

|  |  |
| --- | --- |
| **Data Type** | **Attributes** |
| int | \_id |
| string | \_hallName |
| string | \_ownerName |
| int | \_costPerDay |
| string | \_mobile |
| string | \_address |

The methods for **getters**, **setters** and **constructors** are given in the template code.

Consider a class **HallDAO**.**cs** with below methods to handle all database related operations  

|  |  |  |
| --- | --- | --- |
| **S.No** | **Method Name** | **Description** |
| 1. | public List<Hall> getHall(int price) | This method will get the hall budget as input and returns the list of halls within the budget. |

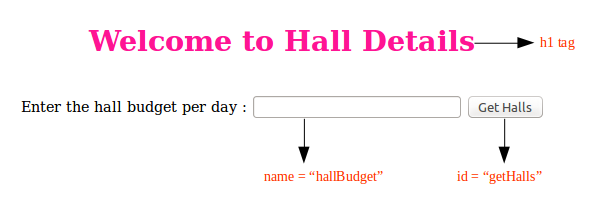
**Page Flow :**  
1. Create**Home Controller**with **Index**action which gets the hall budget as input.  
2. By clicking **Get Halls** button the request must be redirected to **Search**action.  
3. **Search action** gets the hall list within the budget from the user and displays the hall details in a table format.  
4.  If there is no hall available within the given budget, then display **"No halls found within the budget!!!"**

**Use EF for DataAccess**

Insert some sample records data in the table hall as per given in the Screenshot 3:  
  
**Table properties :**  
CREATE TABLE hall(  
    id int PRIMARY KEY IDENTITY(1,1),  
    hall\_name varchar(255) not null,  
    owner\_name varchar(255) not null,  
    cost\_per\_day int not null,  
    mobile varchar(255) null,  
    address varchar(255) not null  
);

**Note: Make sure the UI constraints like element names, id's, UI design, element order, alignment and field labels should be followed as specified in the mock-up screen.**

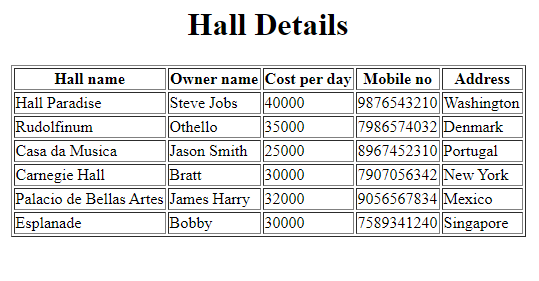
**Screenshot 1 (Index.cshtml) :**

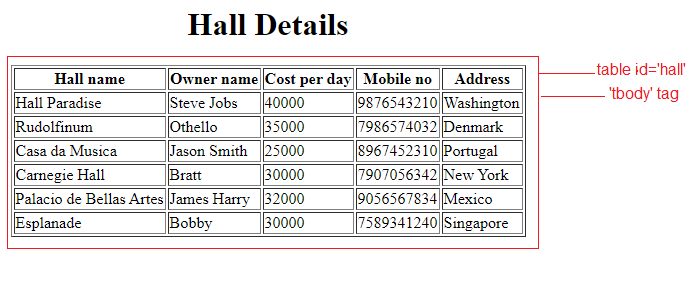
****

**Screenshot 2 (Index.cshtml) :**

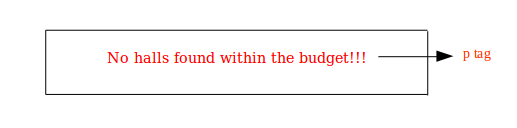


**Screenshot 3 (Search.cshtml) :**

****

**Screenshot 4 (Search.cshtml) :**  
****  
  
**Screenshot 5 (Index.cshtml) :**  
****

**Screenshot 6 (Search.cshtml) :**

****