AMERICAN UNIVERSITY OF BEIRUT

Faculty of Arts and Sciences
Department of Computer Science
CMPS 278 – Web Programming & Design
Lab9/Asst 7, Spring 2022-2023
Due on Thursday May 4th at midnight

Exercise 1 (EF Core)

In this exercise you will be creating a hotel explorer website as a ASP.NET MVC web application as shown in the images below.

Part 1 (Frontend):

Start by modifying the navbar in the **_Layout.cshtml** file and give it 3 items (Home, Hotels, About Us).

Edit the Home view to display a grid of hotel photos as shown below (feel free to use a design of your preference as well).

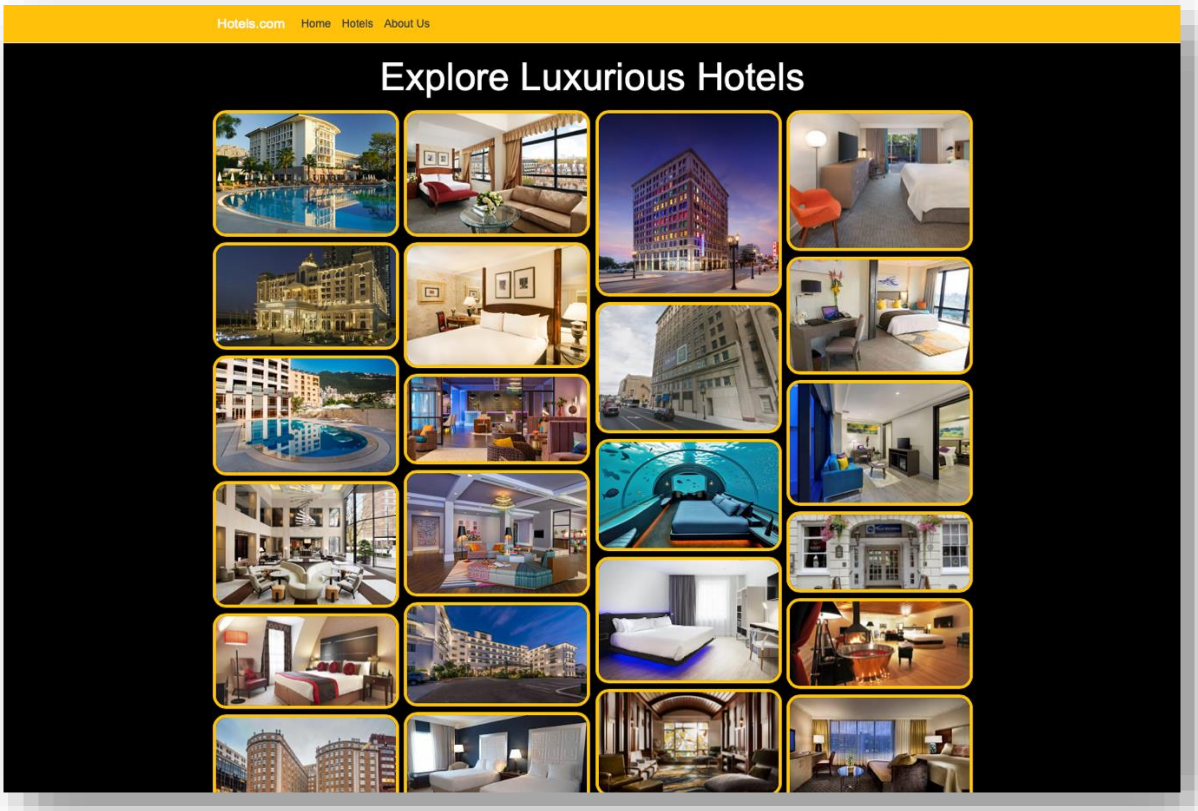


Image 1

(You may find the links of the images above in the Images.txt file on Moodle)

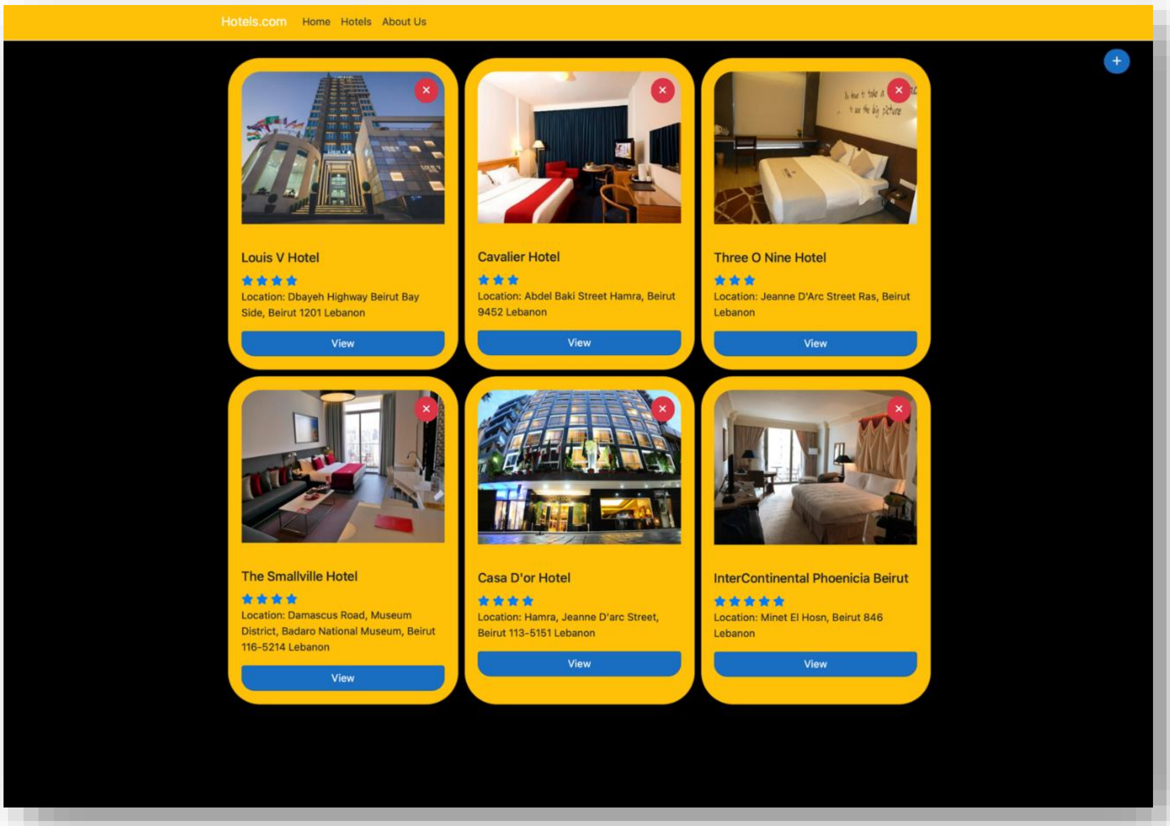
Create a new **“Hotel”** view which consists of 3 pages (these have to go in a folder called “Hotel”):

- index.cshtml (main page)
- createHotel.cshtml
- view.cshtml

Construct an **index.cshtml** page as shown in Image 2.
Start by creating some kind of a grid of cards showing each the image of a hotel (you may use any front-end technology for this) that displays every hotel element in a card view in Image 2.

- Each card consists of an image, name, location, rating, view button, and a delete button.
- A row must contain no more than 3 card elements.
- The *view* button must take the user to the **view.cshtml** page and present him/her with more info about the selected hotel as shown in Image 3.
- The *delete* button must delete the corresponding hotel from the database (after confirming this action of course).

In the top right of the page, you should have a button that allows the user to add a new hotel to the database. It should take the user to a new page (as shown in Image 4) that contains a form which allows the user to enroll a new hotel that he or she would like to add.



Louis V Hotel
★★★★
Location: Dbayeh Highway Beirut Bay Side, Beirut 1201 Lebanon

Cavalier Hotel
★★★
Location: Abdel Baki Street Hamra, Beirut 9452 Lebanon

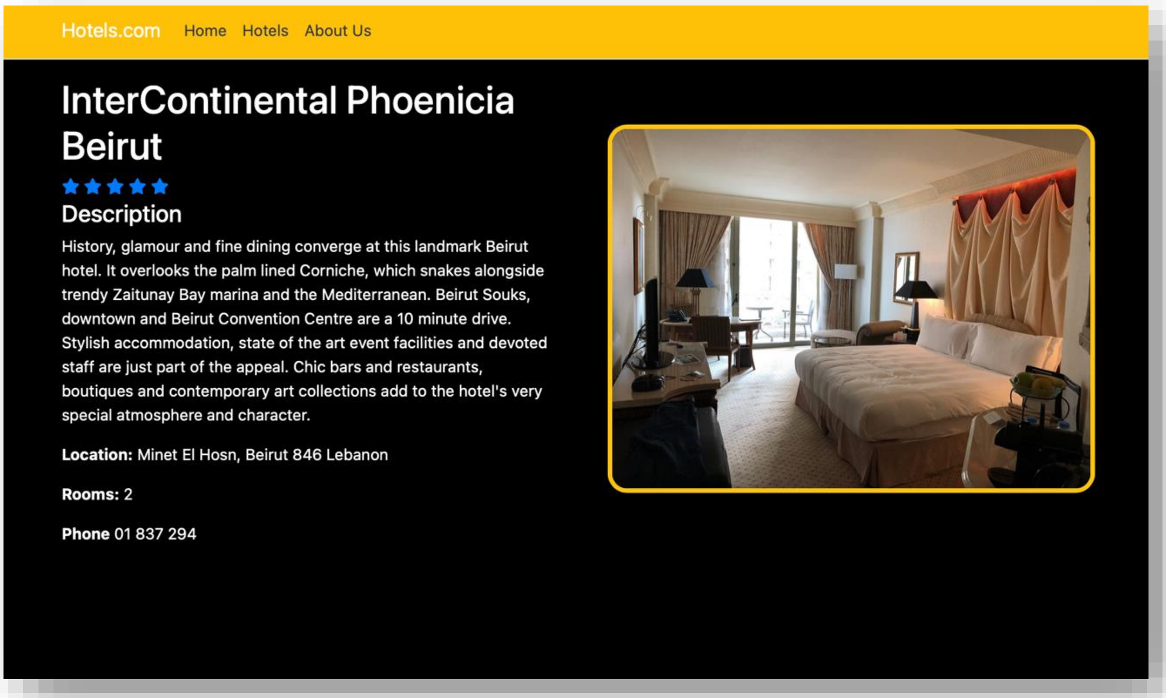
Three O Nine Hotel
★★★
Location: Jeanne D'Arc Street Ras, Beirut Lebanon

The Smallville Hotel
★★★★
Location: Damascus Road, Museum District, Badaro National Museum, Beirut 116-5214 Lebanon

Casa D'or Hotel
★★★★
Location: Hamra, Jeanne D'arc Street, Beirut 113-5151 Lebanon

InterContinental Phoenicia Beirut
★★★★★
Location: Minet El Hosn, Beirut 846 Lebanon

Image 2



InterContinental Phoenicia Beirut

★★★★★

Description

History, glamour and fine dining converge at this landmark Beirut hotel. It overlooks the palm lined Corniche, which snakes alongside trendy Zaitunay Bay marina and the Mediterranean. Beirut Souks, downtown and Beirut Convention Centre are a 10 minute drive. Stylish accommodation, state of the art event facilities and devoted staff are just part of the appeal. Chic bars and restaurants, boutiques and contemporary art collections add to the hotel's very special atmosphere and character.

Location: Minet El Hosn, Beirut 846 Lebanon

Rooms: 2

Phone 01 837 294

Image 3

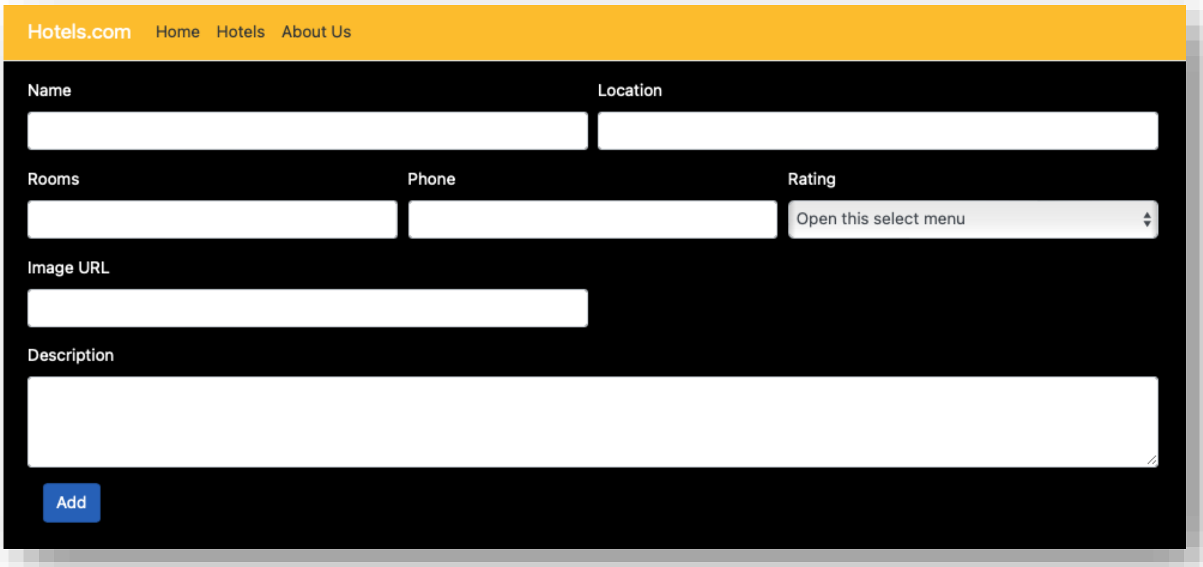
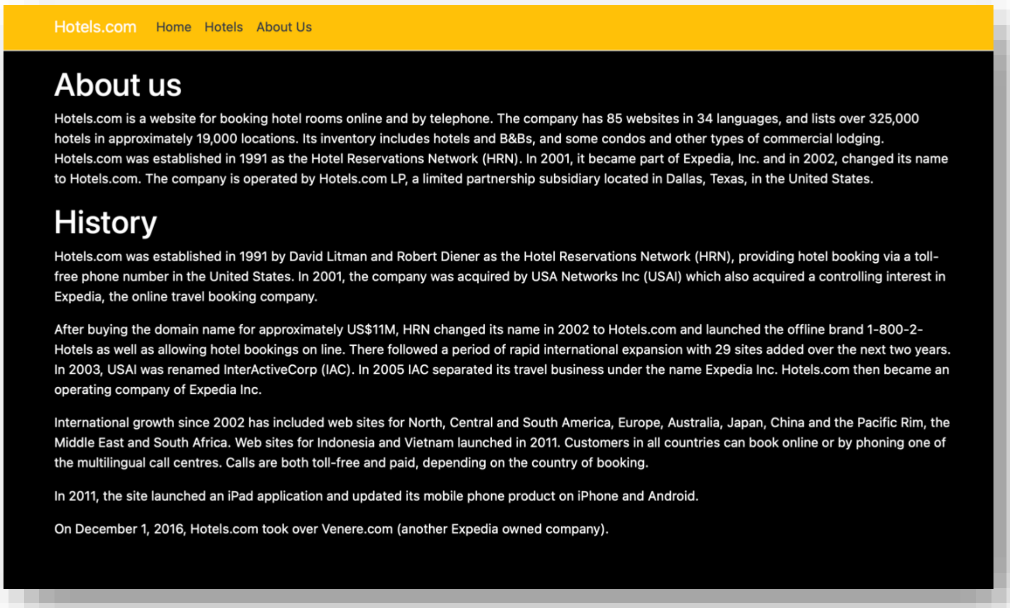


Image 4

Create a new view **aboutUs.cshtml** to display some more details about Hotels.com as a company (Image 5). You may find the text in the “text.txt” file on Moodle.



About us

Hotels.com is a website for booking hotel rooms online and by telephone. The company has 85 websites in 34 languages, and lists over 325,000 hotels in approximately 19,000 locations. Its inventory includes hotels and B&Bs, and some condos and other types of commercial lodging. Hotels.com was established in 1991 as the Hotel Reservations Network (HRN). In 2001, it became part of Expedia, Inc. and in 2002, changed its name to Hotels.com. The company is operated by Hotels.com LP, a limited partnership subsidiary located in Dallas, Texas, in the United States.

History

Hotels.com was established in 1991 by David Litman and Robert Diener as the Hotel Reservations Network (HRN), providing hotel booking via a toll-free phone number in the United States. In 2001, the company was acquired by USA Networks Inc (USAI) which also acquired a controlling interest in Expedia, the online travel booking company.

After buying the domain name for approximately US$11M, HRN changed its name in 2002 to Hotels.com and launched the offline brand 1-800-2-Hotels as well as allowing hotel bookings on line. There followed a period of rapid international expansion with 29 sites added over the next two years. In 2003, USAI was renamed InterActiveCorp (IAC). In 2005 IAC separated its travel business under the name Expedia Inc. Hotels.com then became an operating company of Expedia Inc.

International growth since 2002 has included web sites for North, Central and South America, Europe, Australia, Japan, China and the Pacific Rim, the Middle East and South Africa. Web sites for Indonesia and Vietnam launched in 2011. Customers in all countries can book online or by phoning one of the multilingual call centres. Calls are both toll-free and paid, depending on the country of booking.

In 2011, the site launched an iPad application and updated its mobile phone product on iPhone and Android.

On December 1, 2016, Hotels.com took over Venere.com (another Expedia owned company).

Image 5

Part 2 (Backend):

A. Let's start by setting up the database.
- Import the following NuGet package to your project:
  - Microsoft.EntityFrameworkCore
- In your project's directory, create a new folder "Data".
- Inside the Data folder create a new file "AppDbContext.cs"
- In AppDbContext.cs, pass "DbContextOptions<HotelContext> options" as parameter to the constructor to base(options) "… : base(options)"
- For each model, add public DbSet<ModelName> ModelName {get; set;}
- In Program.cs add the Db service "builder.Services.AddDbContext< AppDbContext >(options => options.UseSqlite(builder.Configuration["ConnectionStrings:LocalConnectionString"]));".
- Migrate your entities to the database.
- Notice the Migrations directory that was just added to your solution explorer. You should be using "SQL Server Express LocalDB" as a database in this exercise.

B. Add a new Model object "Hotel" which contains the following:

- int id
- string name
- int rooms
- string description
- string location

- string phone
- int rating
- string imgURL

C. Add a new controller called **HotelController** that controls the actions of the Hotel view.

As a reminder, the Hotel view consists of:

- index.cshtml (Image 2)
- createHotel.cshtml (Image 4)
- view.cshtml (Image 3)

Use the necessary console commands to migrate your changes to the database.

The HotelController should read the data (Hotels) from the database and store them in a list which is then returned when index.cshtml loads.

The HotelController should:

- Read the data (Hotels) from the database and store them in a list which is then returned when index.cshtml loads.
- Include an Add action that adds a hotel element to the database and redirects the user back to the index.cshtml page.
- Handle the view action by searching the database for a hotel with the same id as the selected hotel and redirecting the user to the view.cshtml page to display the hotel with all of its details.
- Include a *delete* action which simply removes the current hotel and redirects the user back to the index.cshtml page.

Exercise 2 (Standard SQL)

In the previous exercise you implemented your database and performed CRUD operations on it using Microsoft's Entity Framework Core (EF Core).

In this exercise, you will be doing the exact same thing you did in exercise 1, but this time without using the Entity Framework Core. We're also going to change the backend database and use mySQL instead. Don't replace Exercise 1. Keep it and create a copy of it and do the necessary modifications to it. At the end, you need to submit both exercises.

Your schema should include one table (Hotels) using the same fields as before:

- id → int
- name → varchar
- rooms → int
- description → varchar
- location → varchar
- phone → varchar
- rating → int
- imgURL → varchar

Once you get your mySQL database connection done, you should be able to execute standard SQL queries against it. Your pages should function the exact same way they did in Exercise 1.

Again, please make sure you don't overwrite your previous implementation as you are required to submit both implementations (Exercise 1 & Exercise 2).

Enjoy your work ☺

Congrats for completing the final assignment ☺