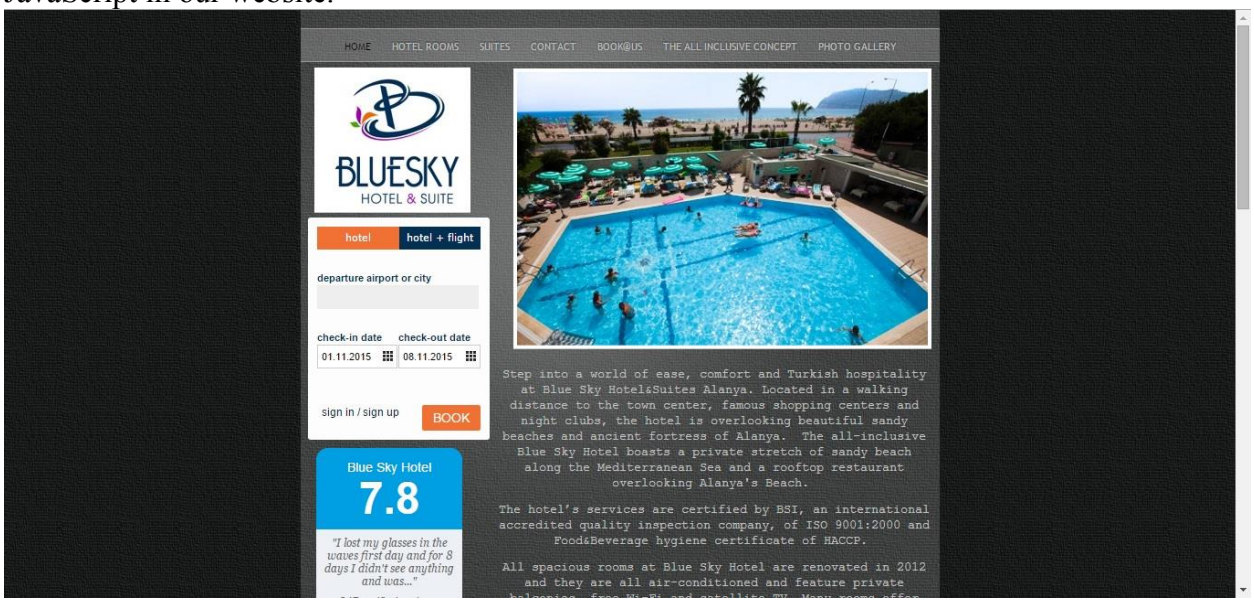
Pair DZ Hotel website's documentation

- The assumptions and justification of the major design and implementation decisions:

First of all, our website has a home page that shows the general description of the hotel, and gallery that shows pictures of the hotel. We have chosen that particular order because we assumed that people wanted to know general information regarding the hotel beforehand: how it looks like, what is the local area, what does the hotel provides, and so on. Facilities section shows the services that are provided, and we put it as a third option because we thought that customers wanted to know about the services after they looked at the gallery. After the Facilities, we wanted our customers to see available types of rooms with their prices, descriptions and services. The Comments section is supposed to help people see others' opinions and leave their own opinions. We put it after Gallery, Facilities, Rooms and Services sections because we do not want users to judge us without going through what the hotel is able to provide. Lastly, Contact us section, which is usually at the end of any menu, is the last task that customers would want to do in order to contact us if they wish to stay in our hotel.

- Discussion of the main features of similar sites:

Blue Sky Hotel (http://www.bluehotel.com/BLUE_SKY_HOTEL/HOME.html)
It is a good example of a nice layout. The webpage is clean and has a good style and not confusing to use. However, every button (e.g. HOME, HOTEL ROOMS) send user to a different .html document. Therefore, we tried to prevent that by using JavaScript in our website.

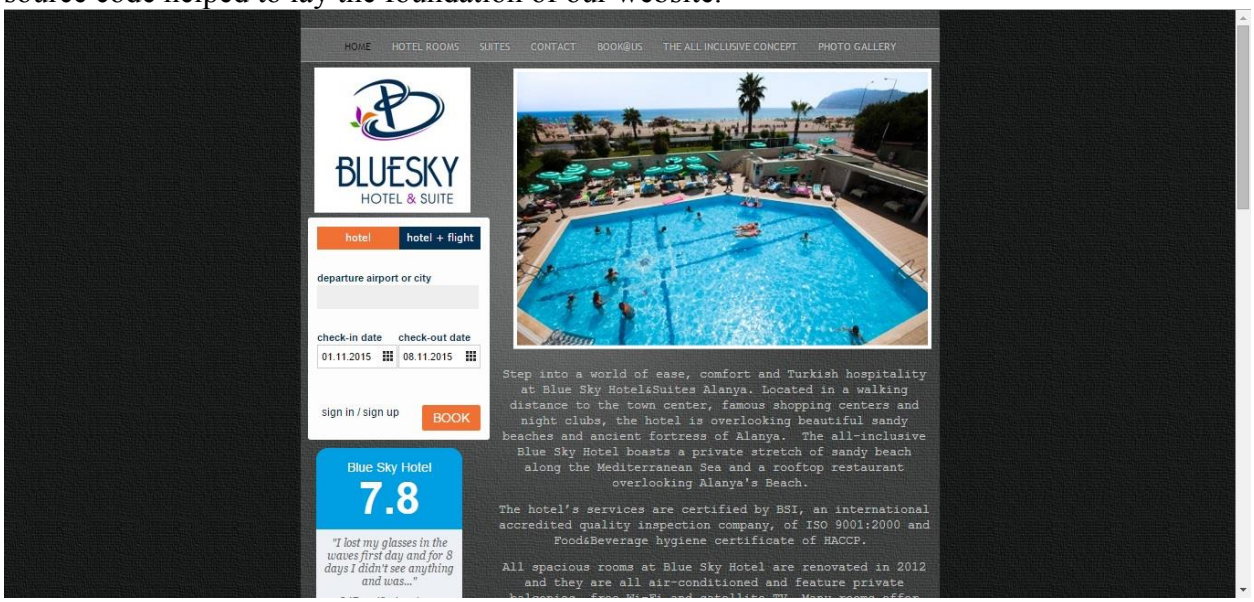


Fancybox (http://fancybox.net/)

We wanted our gallery to look stylish and with lots of research on the web how to make it possible, we decided to use an external plugin's JavaScript, called "Fancybox".

Lab Example website

A great example of a proper functional website without additional styles added. Raw source code helped to lay the foundation of our website.



Exmouth View page (http://www.exmouth-view.co.uk/)

A great example of how not to make a website. There is no layout and everything is cluttered. The pictures are very small and do not become enlarged on click. No descriptions about the rooms, services and local area. Contacts are very hard.

- Brief description of the structure of the site:

First of all the website will show the homepage where it has different pictures with different views of the local area so basically customers feel a bit tempted. The layout of the website has a header where the buttons located and a footer where it has our names and name of school and name of module, between them is the contents where it looks like inside a box so that means it is well organized not a random design. Each picture on the website is a thumbnail and uses an external JavaScript plugin called Lightbox to open larger images in a popup whenever you click on an image. Website's HTML only contains the main DIV, yet the content of it is changed by JavaScript (with the help of JQuery) through a set of buttons in the navigation menu that use JS and JQuery to switch between parts of the website. Home button shows the description of the hotel and the location. In our gallery we have different kind of pictures. Also, the pictures in the gallery sections are thumbnails, with a function to make them larger on click. The facilities section contains descriptions of different facilities that are provided in the hotel so customers can enjoy their time while staying at the hotel, also, each facility provided has its own thumbnail picture so users can recognize the place they are going to. Rooms and Services section shows each type of room available at the hotel, description and price of said room and the services that are included there. Comments section provides the user a way to give their initial impressions or feedback about hotel's services, website, etc. or to read other users' impressions. 'Contact us' shows a way of contacting us and if there are any problems with the user or website or any queries with facilities and the room booing questions we will try respond as soon as possible.

- Descriptions of tests for validity, browser compatibility, accessibility, etc.:

We executed all the system components to evaluate all its properties. This involved testing the extent to which the system's components:
1. Meets the requirements and specifications that were designed in the coursework sheet.
2. Responds correctly.
3. Execute the functions within an acceptable time.
4. It is usable and runs on different platforms and environments.

To do this, we tested the system on different levels. These are: the validation test, the compatibility test, usability test and performance test. In the system validation test, we demonstrated that the website fulfilled its intended use when deployed on appropriate environment. Here we studied: requirement specification, general design of the website, small details design and coding as well. We tested the website by resizing the window and everything worked perfectly on any size (mobile devices included). For the browser compatibility and with the endless combination of modern browsers, it is our responsibility as software developers to ensure the websites we build perform adequately according, we conducted comprehensive test for our website, thus we checked if it appropriately displays across various browsers, platforms, including: Safari on OS X operating system, Internet Explorer, Google Chrome and Microsoft Edge on Microsoft Windows operation system, as well as on other mobile devices (Android and iOS).