Winston Downes

2872112

COSC212 Assignment #2

For assignment 2, we were required to make the front end of a website for a fictitious hotel, which used a lot javascript as well for functionality. My Hotel was named ‘The Harbour Hotel’ and was located on the Otago Peninsula.

The most testing time was done with the calendar functionality. It was important that users not be allowed to enter bad input into the website, or else it would break the code. I wanted to make sure that one couldn’t enter dates before the present day, as that wouldn’t make sense in a booking scenario. In line with this, users are also not able to enter departure dates before the arrival date, even if both dates are in the future. It is also not possible to book a time span that overlaps with current bookings. An arrival and a departure are allowed on the same day, as this is often the case in hotels (they’ll generally have a certain time by which you must vacate the room). The overlap was calculated by taking the arrival of one person and measuring whether this date was before the departure of another’s departure; if the first person’s departure was scheduled to be after the arrival date of the second, it was then determined that an overlap would exist. Again, this is not including arrivals and departures on the same day. Customers have an option to view currently held bookings in the system, and then use that to help them determine their booking. In all of these cases, an alert is displayed to the user informing them of the situation, and all the hitherto entered date data for this current room will be reset. Although it may seem a little inconvenient, it was found to be safer for input to have the dates cleared in the event of an input error.

When customers chose dates without error, they were put into a cookie, a long with the type of room, and whether that room was standard or deluxe. Based on this info, a calculation was done that multiplied the amount for that room by the number of nights staying. The customers were then able to visit a page where they could see the total.

Concerning the map functionality, most of the locations on there were on the peninsula – all were fictitious apart from Dunedin City and Lanarch Castle. To make the markers disappear, I checked the opacity setting to see whether it was set to 0 or 1 - I the used the setOpacity() method to switch it to the other value.

I used three XML files for the project: One provided the booking information that staff and customers had access to; one lists the room types, and one provides information regarding the options they have for each room type, of which there are two (standard and deluxe). I used ajax to parse data either into functions or variables, depending on what I wanted to do with it. When I had several methods that needed to get to the data, I found it easier to parse it into a variable to which several functions had access. I had to access a lot of information from variables on other pages, and I had a lot of difficulty using ‘this’ to refer to the caller of the current item, so I often ended up passing the reference in itself as I called the function, which made for some long method calls: For example: <input type="button" value="Book Standard" onclick="addToBookings(this, $(this).parent().parent().find('span.price1')[0].textContent)" class="bookStan">.By doing this, I was able to work with variables that I needed when I was in another part of the code, where referencing them would have been an issue.

When customers go to the ‘Rooms’ page, the appropriate XML is loaded and they are able to see the room types. There are highlighted links that , when clicked, will display the two subtypes of room (standard and deluxe), with a little blurb. The other XML file provides this information.

I tried to implement a toggle function so that the customers could hide the ‘availability’ information for the rooms, but I had an issue where I had to click it twice in order to get it to work; after the one superfluous click, it word perfectly.

I took a little code from a couple of websites in order to help with the javascript – I have written the addresses of these sites in the code, so that this can be checked. The biggest bit borrowed was for a pop up window, which I wanted to use to display room options. I found trying to do it myself was a bit finicky and time consuming; it’s not a plugin.

The CSS validates with no warnings and all the HTML validates apart from one page, which has the same warning several times: I use a datepicker and it’s warning that not all browsers are capable of displaying them.