Part 3: The Development and testing

Changes

In terms of content on the webpages, my planning of which content to include was all implemented successfully. However, in implementing my content I had to change the positioning of some elements, for example on the home page as the browser size increases I introduce three separate columns. This is because as a viewer of the site you would expect more content to fill the screen rather than having to scroll down to see content that could otherwise be viewable without having to traverse down a page. For this reason I have also introduced column views on several other of my webpages so that content fills the screen easier.

Organisation

When I created my website I first began by creating a single 'master page' template that did not include any page specific content but rather all the content I would like to keep exactly the same for each webpage such as navigation bars and logo positioning. For this reason, I also created a default CSS template to style these elements. This allowed me to quickly reproduce webpages with varying content whilst keeping the continuity across webpages present. I did not use any CSS normalisation or reset since my website only needs to be compatible with the latest two versions of web browsers. However, if this were not the case I would use one of these options in order to make a websites style rules continuous across all browser types since different browsers have different default properties.

File structure

I have organised the file structure of my website so that it is easily maintainable and to reflect the way pages would be stored on a server. Hence, I have put all my main HTML files in one folder, and within that folder I have sub-folders for images, CSS and JavaScript.

Debugging tools

During development of my website I used the Mozilla Firefox browser console in order to debug any errors in my code. This was particularly useful when implementing the JavaScript for my webpages because I could not view what the effects of altering code were whereas with HTML and CSS I could.

Image optimisation

In order to increase image loading speeds on web browsers I have made sure that the size of each image is not too great. I have therefore used lossless compression techniques for each image included on the website resulting in no image being of a size greater than 200kb. I have used lossless compression since it reduces the size of the image without actually

reducing the quality of the image as viewed by the human eye since only certain wavelengths of light can actually be detected. As such lossless compression removes anything from the image that is completely indistinguishable by the human eye anyway.

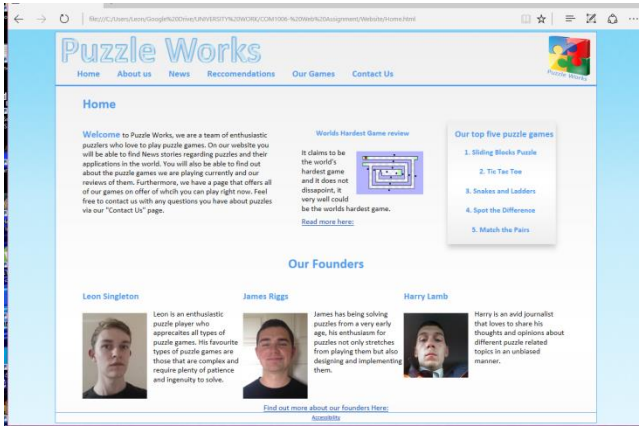
Security concerns

There are several security concerns associated with implementing a website that includes HTML forms. One of these is cross site scripting which means that a user could create arbitrary content and make it appear on my web pages along with capture data from any user form submissions. Another issue with forms is that there needs to be a validation method to prevent users from submitting spam messages and messages in incorrect formats. Furthermore, the form page should be an https page since it will contain sensitive information that is private to an individual. For my website though I do not need to be worried about cross site scripting since the case where I use a form is on the "about us" page, and when the content of the form is submitted it is passed through the email gateway FormMail.pl which parses he results of any form entry, thus providing protection from cross site scripts. I also do not need to worry about validation since there are features in html that make sure that a text entry field has been completed and that the email text box is in an email form, of which are included on my website.

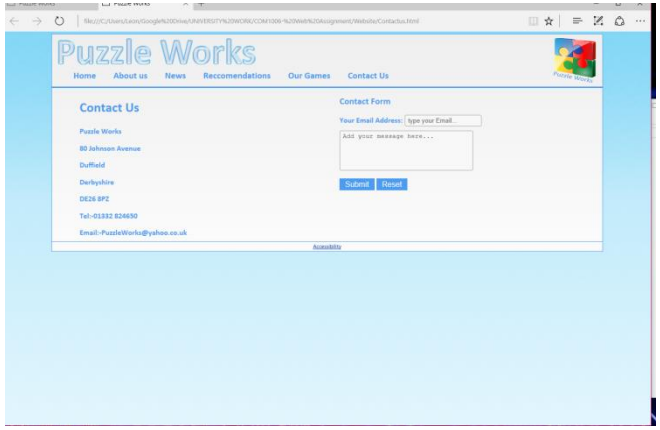
Testing

In order to carry out my testing I used three different browsers and three separate devices to test how my webpages would display on different browsers and whether they would appear the same across all devices and browsers tested. The devices I used were my own laptop and desktop and a university desktop machine. In order to test the mobile functionality of my website I used the responsive design mode of Mozilla Firefox which shows how the web page will be displayed as the browser resolution alters. I also tested all three spot the difference puzzles. When testing the puzzles I did not test them on a mobile device since they should be completely compatible as I have used click events which correspond to a user's touch input as well as mouse input. Although I have not included image evidence of all pages successfully resizing and altering their display of content, I can confirm that they worked as expected for each browser tested.

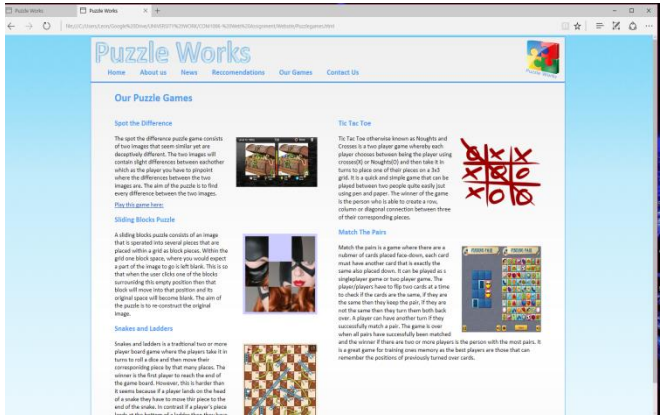
Microsoft Edge(15.6” Laptop)



1. Home Page



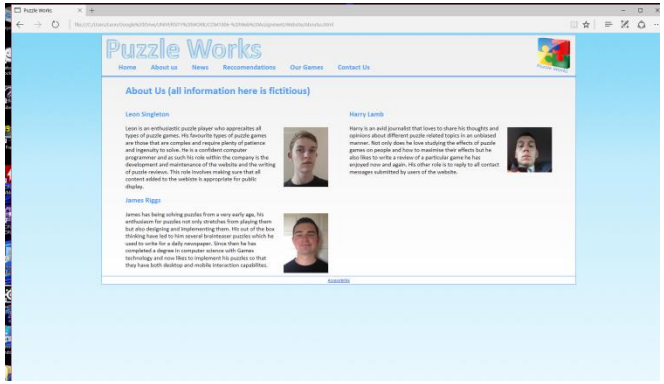
3. Contacts page



2. Our Puzzle Games

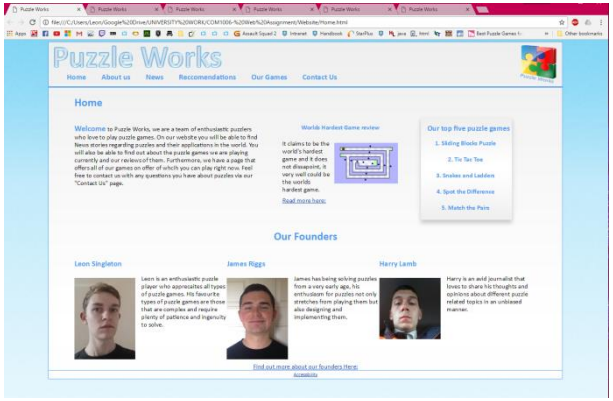
6. Recommendations Page

5. News page

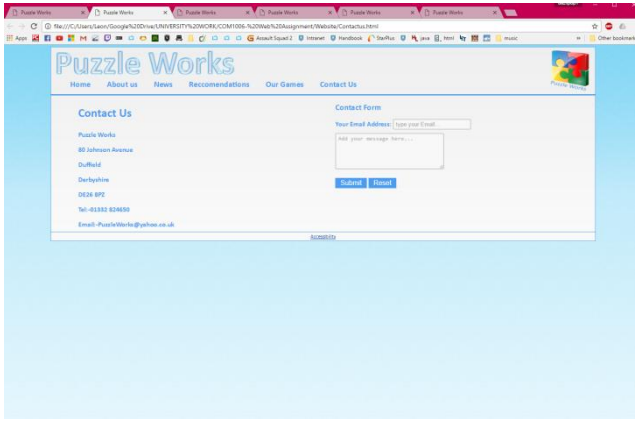


4. About Us Games

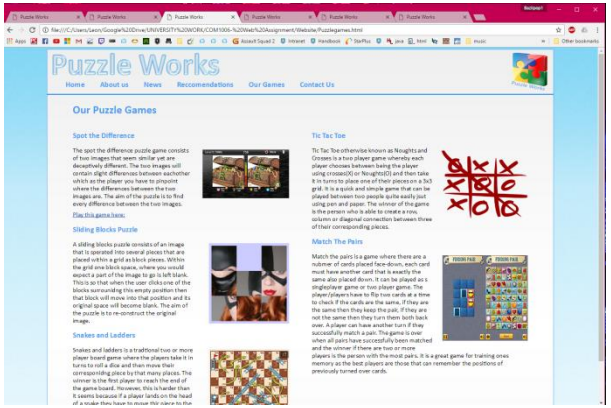
Google Chrome(23” desktop monitor)



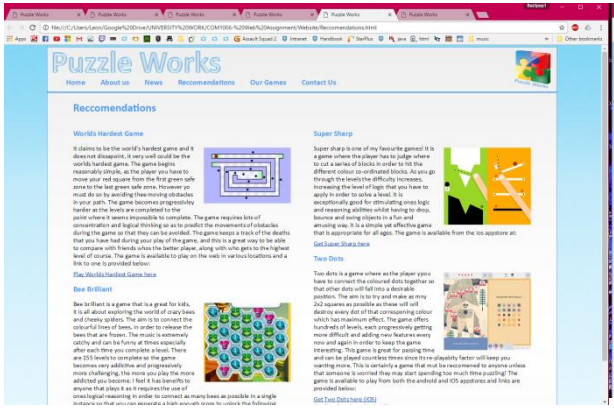
9. Home Page



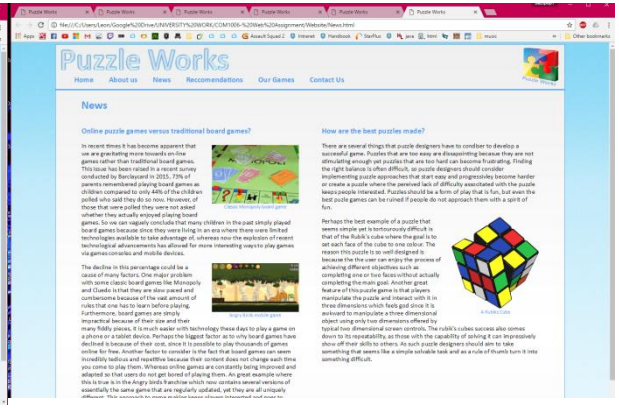
8. Contacts page



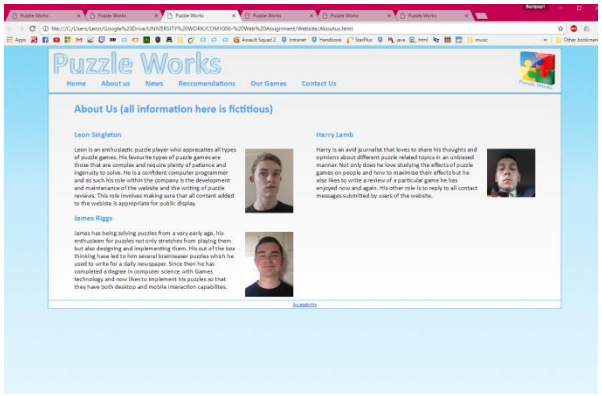
7. Our Puzzle Games



11. Recommendations Page

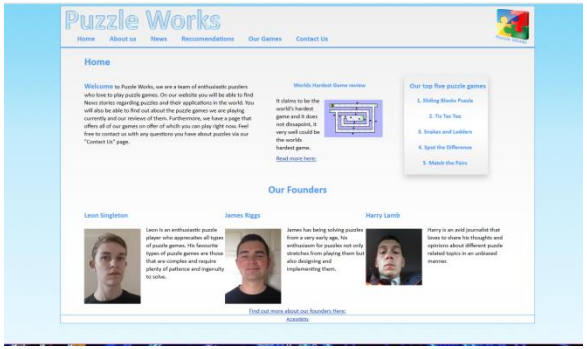


12. News page

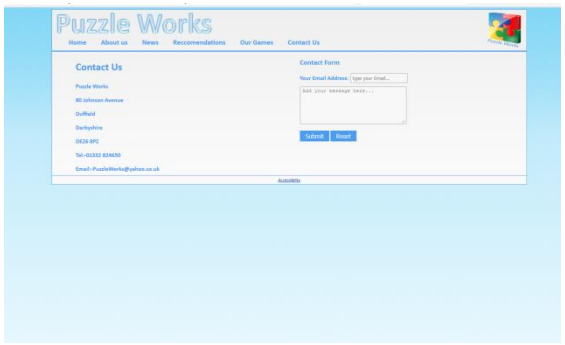


10. About Us Games

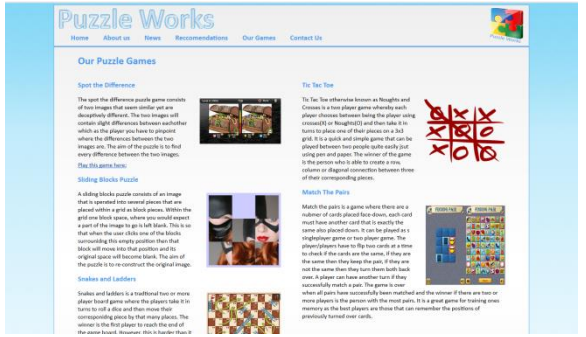
Mozilla Firefox (27” monitor display)



15. Home Page



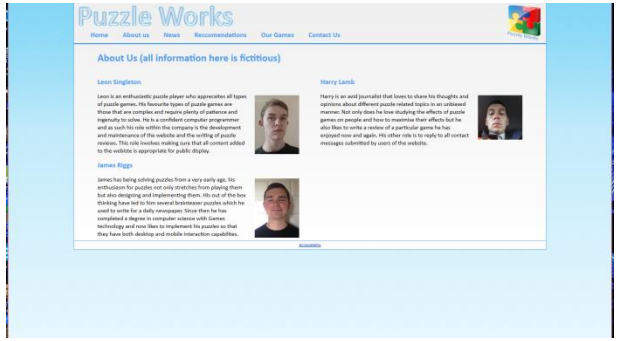
14. Contacts page



13. Our Puzzle Games

18. Recommendations Page

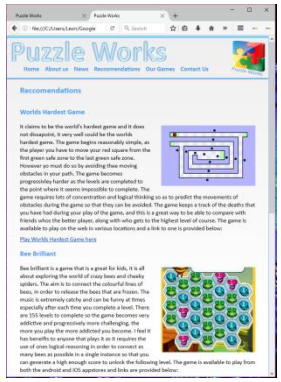
17. News page



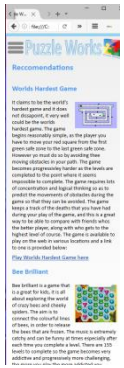
16. About Us Games

Mozilla Firefox (re-size test)

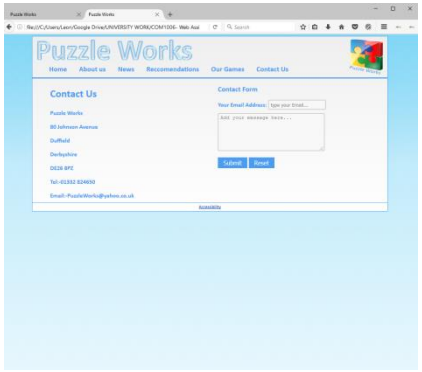
24. Recommendations Page (desktop)



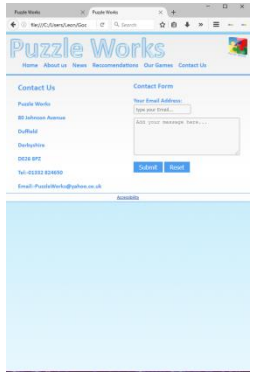
23. Recommendations Page (narrow)



22. Recommendations Page (mobile)



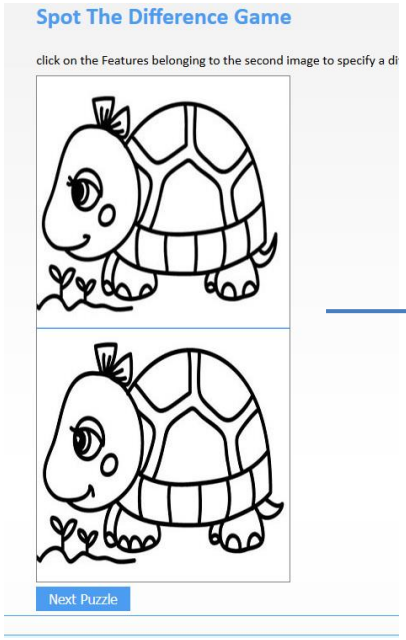
21. Contacts page (desktop)



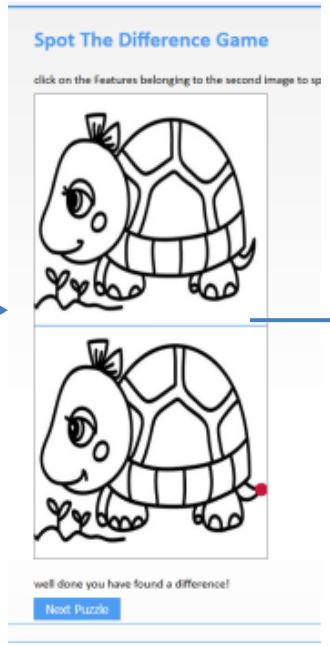
20. Contacts page (narrow)

19. Contacts page (mobile)

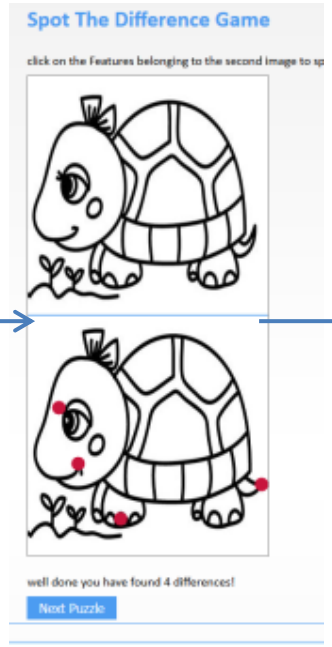
Mozilla Firefox (Puzzle game test)



29. No differences

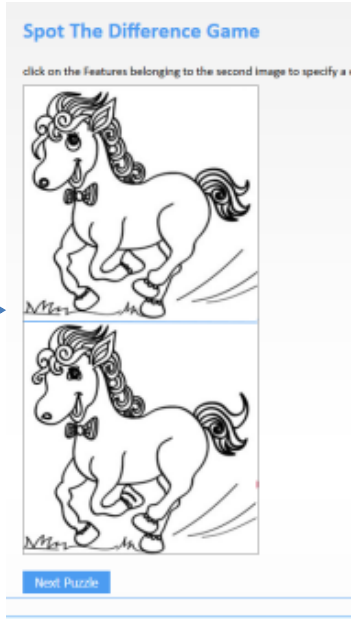


28. one differences



27. four differences

26. All differences



25. Next Puzzle