Web App Development Project

|  |  |
| --- | --- |
| **Group Name** | |
| ABC | |
| **Student Names** | **GitHub Accounts** |
| Conor Hughes x17454386  Tony Calbby x16411702  Oscar O’Byrne x17363253  Aaron Hynes x17742385 | |
| **Link to your final Workspace in Cloud9 (just one)** | |
| https://ide.c9.io/studentusername/WAD\_project | |
| **Node.js modules used in the project (if applicable)** | |
| express, body-parser, etc. | |

Project Report

The report should be 6-8 pages including:

1. The main aim of the project of the project.

The Aim of our project is to create an application that demonstrates both XML data processing and client side scripting. We chose to create an online application listing houses. The website should contain crud functionality. The webpage must contain the function read, edit, Add and delete. In our application we created buttons on each page that demonstrated these functions. We also added scroll bars to the list of houses page. We also used different elements such as JavaScript etc.

**Home page** - this page display information about the website and about the company also. The page contains a button that display a background image for the page. This is an example of JavaScript.

**List of houses** – this page displays the houses that are for sale on the webpage. It features a scroll bar that goes from left to right. The page also has an edit button which edits the information inside of the JSON file. The page also has a drop down list which display information about the house such as type, price, description etc.

**Buy a home** – this page has that same layout as the list of houses page but the functionality inside the page is different. It contains a button called buy when this button is clicked it removes the details from the house. json file, it then also removes if from the buy a house page also. The page also contains a scroll bar and a drop down list.

**Sell a home** – this page contains a form which contains the information required by the webpage for you to list your house. These details are type, description, address, seller, contact number, contacts email. When the Add button is clicked it then adds the information to the house.json file and then displays the information on the list of houses and buy a home page.

**Make a comment** – this page contains a form where customers can provide feedback about the business and the website. When the submit button is clicked it added the information below the form and displays it there.

**Upload an image** – the page gives you the ability to upload an image to the websites image folder that can be used when creating a listing. This step is completed by click the choose file button and selecting an image and chose upload.

1. The strategy employed to meet the stated project goals.

In order develop the application we split the work up evenly. We used GitHub the push and pull certain pages inside the web page for team members to edit them etc. we created a WhatsApp group where we could communicate between us and delegate the tasks evenly. We discussed issues that we were having issues that required fixing inside the web-page.

1. The key characteristics of your project in terms of functionality.

**CRUD** – The finished app contains complete CRUD functionality. Data can be Created using the Sell Houses page, inputting into this append your data into the JSON file.

Data can be read from the House List page, which displays data from the JSON file back to the user when the app is running.  
  
Update can also be done on the house list page. The Delete functionality is done via the Buy Homes page.

**CSS –** CSS was used throughout to provide specialized and unique styling to every page. Bar some very minor exceptions, all the CSS was pulled from a single external .css file.

**JavaScript –** With JS being the basis of all of NodeJS and with our App.js tying our application together, I can safely say that JavaScript tied every element of this application together.

**Forms –** Excluding the homepage that was made for the application, every page on our application utilized the form functionality.

**jQuery** – At several points we added in jQuery effects to add flare to our finished application. Examples include a sliding div in the home page.

**HTML 5 form validation** – HTML was implemented into the form on the sell houses page and gave each field data exclusivity.

1. A list errors encountered and methods used to find solutions. Identify the solutions used.

Syntax Error: **comments is not defined**

This error occurred when the comments page tried to get information from the comments.json. The first step taken to see if the line of code in the app.js that allowed access was named correctly. \*\*\*INSERT IMAGE\*\*\* Seeing this was not the case that meant the error was in the line that rendered the comments page. \*\*\*INSERT IMAGE\*\*\* The comments page only was able to access information from the house.json instead of the comments.json. This was simply fixed by changing house to comments.

TypeError: path must be a string or Buffer

Images Error: **Images do not appear of houses**

This error would happen when the home list page and the buy home page were opened. The error was caused due to the application not knowing where to find the images. In the app.js a line of code was missing that allows the web app to know where to look for files to use. \*\*\*INSERT IMAGE\*\*\* Adding in the line allowed the images to be rendered in the web app.

1. Overview of the technologies used in your app.

**Express** –The framework we employed as part of our NodeJS application building. Its built in modules were essential and allowed us to tie this whole project together.

**NodeJS** – The ultimate framework builder of this application; which we used to build the vast majority of the front-end and back-end of our application.

**Body parser** – This extension was used when reading from and editing the entries in our JSON files as it connected the JSON data into something we could display when our application was active.

**JSON** – JSON was the foundation of our CRUD functionality. We employed two unique JSON files, a single JSON to hold the details of individual houses for sale and a second entirely used for the comments page of the app. Thus it is used in the Housing list, the buy houses page, the sell houses page and the comments page.

**jQuery** – We used jQuery effects on several pages in order to add fun effects. In the house list page for example, we have an effect where clicking on the show description paragraph causes a short description to slide down and become visible.  
  
We also included a fun feature on the homepage where clicking a button will set the background to a new image.

**JavaScript** *–* Employed in all stages of the app per say, as the NodeJS framework is founded on JS and the App.js file in our app is the metaphorical trunk of the application.

1. Describe in detail all testing implemented and results of this testing.

Testing was implemented by each of the team members on the team.

Each time new code was added to the website it was saved and ran. Testing was done a lot when add new elements to our page such as JavaScript and jQuery and especially when creating the layout for the page also. A lot of testing was done with the upload an image page. When the button was clicked it would create a new file instead of adding an image. This was due to a .ejs method in the app.js section which was causing this. A lot of testing was also done on the list of houses page as all the drop down lists would not work. We resolved the issue due to the code being misplaced.

1. Instructions for use.