Web Technologies Coursework Part 2 Report:

Public GitHub Deployment:

https://lewisstevenmackenzie.github.io/WebTechCoursework/index.html

Differences from Initial Plan:

Heist or Hostage follows an unchanged story line developed in the part 1 submission. The code however is much more advanced than initially planned. It is important to note that the website runs flawlessly on all major browsers except "internet explorer" (edge is fine though).

The website consists of 4 unique pages. These are as follows:

Start Page: This page is the initial load screen. It allows the user to choose whether to start the game or not.

Game Page: This is the most advanced page. I had initially planned on the game taking place over multiple pages. However, as I became more familiar with JavaScript, I changed the concept to one that only required a single page. More on this later in the report.

Instruction Page: This will explain the different features of the game.

About Us: This page gives a quick synopsis of the project and why it has been created.

How does the Game work?

In the game.html page there are multiple tags, these correspond to image, buttons, and text elements on the page. Each of these have a unique Id that can be referenced. The buttons are used to navigate to the next step in the story. The game logic is stored in a dedicated file called game.js. This file can only be accessed by the game.html file. The JavaScript in this file will determine what information is displayed on the page in the tagged elements. There is also a second JavaScript file called index.js that interacts with the index.html file.

I decided to store the variables that were to be displayed on the game page in a list called scenarios. Each item would contain an image, text to be displayed, audio file and a children list. This would contain the information to be displayed on the buttons and where they would direct you upon interaction. To do this efficiently I created a Scenario class and a ScenarioChild Class and then filled in the data to the scenario list using the constructor methods. In total there are 25 unique pages in the storyline. The game reaches an end state when it encounters a scenario that contains an empty child list. A play again button will now be displayed giving the user the choice to return to the start (scenario 0). This design method allows the game to be easily scaled to any size in the future, creativity being the only limiting factor.

Once a player has chosen a move, the index of the associated scenario will be added to a stack called indexStack. This allows the user to return to previous pages using the 'back' button. It

will load the scenario associated with the last element in the stack. The element is then popped off the stack. This means the player can retrace their steps right back to the start of the game. The back button is not visible on the initial page.

To personalise the game, you will be prompted to enter your name at the start after clicking 'Play!'. The game will then use this value at different points in the storyline depending on the path you choose.

Audio:

I decided to implement a web API for TTS (Text to Speech) called Speech Synthesis Utterance. This allows the user to click on text within the game page and hear it read aloud by the computer. Not everyone would wish to have this enabled, for this reason I have included a toggle TTS button that is displayed in the toolbar at the top when playing the game.

The other audio feature in this game is activated when you hover your cursor over an image in the game. On some pages this will activate a sound effect. Like the TTS there is also an icon on the toolbar to mute this feature.

Assignment reflection:

At the beginning of this module, it was fair to say that I had no prior JavaScript, CSS or HTML experience. This made planning the Coursework part 1 submission rather challenging. I was not 100% certain what would be possible within this environment and within the timescale. As the weeks progressed, I became more familiar and comfortable with these programming languages. As a result, my web page design began to evolve from the original submission into something more user friendly and achievable. I was able to pull on the help of friends and colleagues to test my adventure game and provide useful feedback. I took this information on board and adapted the layout and design colours to provide a more pleasant user-experience.

If I were to continue this project in the future, there are a couple things I would like to improve upon to enhance the immersion. Each page of the game contains an image. Its purpose, to set the scene and enhance the detail provided by the text. Ideally, I would have designed each image personally. Unfortunately, I did not have enough time to design these images to a standard I would be happy with. Therefore, I needed to use images from the internet (referenced in the images folder). As this was a web tech module and not an art module, I hoped this would not affect the grade too severely as the concepts are all in place to allow the images to be changed in a couple clicks. Simply change the image name in the corresponding Scenario constructor. I would also like to implement more sound effects when hovering over the images. Similarly, to the images this can be done by changing the empty "" value in the audio section of the Scenario constructor to an audio file address.

I would also like to change how the user enters their name. Currently it is a prompt that pops out. However, I would change this to be a sign in page of its own. However, I was not too sure on the best practice to pass this variable between pages. A possible method would be to use URL parameters.

Overall, I am very proud of the achievements I made whilst carrying out this coursework. The web page has a solid foundation with the possibility to further develop if so desired. The Code is clear, functional, robust and the website looks good.