**URL –**

<https://norah723.github.io/MajorProjectGame/>

**Social & Ethical Issues Documentation**

With advancement of technology in the 21st century, software undoubtedly experienced incredible growth in its development. The increased implementation of software applications and solutions have provoked numerous social and ethical issues. These increased complications include privacy, confidentiality, ergonomics and inclusivity issues.

Privacy is the protection of an individual's personal information and details that allow others to identify them. Since businesses commonly require the user's personal information to be given to continue the process of carrying out their functions, a collection of personal data will be recorded. The confidentiality of such private information is categorised as the highest privacy concern~~s~~. To ensure privacy, private information will not be collected, disclosed or used without the individual's knowledge or permission. Therefore, it is essential to consider and acknowledge how personal details can be recorded and used only for intended tasks when implementing a software application. Unlawful leaking of such information would result in inevitable consequences to both the individual whose information is leaked and the business where the information is leaked from. Other than information that can indicate one's identity, data containing details about personal preference is also crucial. Leaking of any information that can be categorised in these two sections would be detrimental as anyone would be able to detect and identify the individual with leaked details.

Within my software application, this issue appears when the user is required to enter their name at the start of the game. However, considering the privacy of such personal information, the data inserted by the user will only be recorded when the webpage browser is not refreshed. Once the webpage is reloaded, the information stored (ie. the name of user) will be cleared. This type of setting aims to minimise the security and privacy issue of these personal details to a certain extent as the data will not be recorded permanently within the software application.

Ergonomics is defined as the study of interactions between human workers and their working environment. Ergonomics is an important aspect of software solutions that requires consideration as it is the connection between users and applications. One issue is the user interface, which is officially defined as the screen designs and the connections between screens that allow the user to communicate with a software solution. The most important aspect of user interfaces is the consistency in design and appearance to each user, as individuals may experience differences in vision.

For instance, in my software solution, colour is used in contrast between texts and background to ensure that all users can see the text~~s~~ easily. The chosen colours have been assessed through the Web Content Accessibility Guidelines (WCAG) 2.0. This Success Criterion determines the accessibility between texts and its backgrounds so that it can also be read by individuals with moderately low vision and does not use contrast-enhancing assistive technology. As colour deficiencies can affect luminance contrast to a certain extent, the contrast is calculated in a way that colour is not the main aspect. This way, individuals who possess a colour vision deficit can also read the texts easily. A contrast ratio of 3:1 is the minimum level recommended for the text and background of a webpage, and a 4.5:1 ratio is used for the user population with moderately low visual acuity, colour deficiencies and loss of sensitivity to contrast caused by aging. The 4.5:1 ratio is classified for level AA as it compensated for the loss in visual acuity for elders approximately at 80 of age with a vision of 20/40. For my major project, the contrast ratio of 7:1, which is level AAA, maximises the inclusivity of disabilities in terms of vision. It compensates for the vision loss equivalent to approximately 20/80 vision. Normally, individuals with more than 20/80 of vision use assistive technologies to support them. This contrast enhances the inclusion of users with different vision abilities, minimising the issue of vision disabilities in terms of user interface in ergonomics.

**Storyboard/Wireframe –**

Diagram, text, letter

Description automatically generatedDiagram

Description automatically generated

**Data Dictionary –**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Scope** | **Description** | **Examples** | **Validation/**  **Constraints** |
|  |  |  |  |  |  |
| myInput | String | Global | Used to identify each user | MajorProject.htm 22  MajorProject.js 9 - 15 |  |
| arrObjects | Array (String & Number) | Global | Used to store and generate random images for users to purchase and sell | MajorProject.js 53 - 59    Ie. arrObjects[0]; |  |
| var apikey | Variable (String) | Global | Used to identify the database | MajorProject.js 2 |  |
| var url | Variable (String) | Global | Used to identify the database | MajorProject.js 3 |  |
| var initialCurrency | Variable (Number) | Global | To set the initial amount of currency in the code | MajorProject.js 5 |  |
| var itemClicked | Variable (Number) | Global | To identify which image is clicked | MajorProject.js 6 |  |
| var rndNo | Variable (Number) | Global | To generate random images to be displayed | MajorProject.js 61 |  |
| var numberUsed | Variable (String) | Global | To identify the images that are being used | MajorProject.js 62 |  |
| var price | Variable (Number) | Global | To randomly generate a price of each object between their minimum and maximum set values | MajorProject.js 64 |  |
| var successfulCount | Variable (Number) | Local | To create a loop that only loops over 3 times | MajorProject.js 68 |  |
| var numberOf | Variable (Number) | Local | Determine the amount of stock of the item randomly generated | MajorProject.js 84 |  |
| var numberOf2 | Variable (Number) | Local | Identify the amount of stock available of the item randomly generated | MajorProject.js 85 |  |
| var displayHTML | Variable (String) | Local | To display the random generated images to the user | MajorProject.js 87 |  |
| var displayHTML1 | Variable (String) | Local | To display information about the item randomly generated to the user | MajorProject.js 89 |  |
| var displayHTML2 | Variable (String) | Local | To display information about the item randomly generated to the user | MajorProject.js 91 |  |
| var purchaseAmount | Variable (Number) | Local | To calculate the price for the purchase of an item randomly generated | MajorProject.js 140 |  |
| var numberOf3 | Variable (Number) | Local | Identify the amount of stock available of the item randomly generated | MajorProject.js 141  MajorProject.js 171 |  |
| var numberOf4 | Variable (Number) | Local | To display the random generated images to the user | MajorProject.js 142  MajorProject.js 172 |  |
| var displayHTML3 | Variable (String) | Local | To display information about the item randomly generated to the user | MajorProject.js 154  MajorProject.js 187 |  |
| var displayHTML4 | Variable (String) | Local | To display information about the item randomly generated to the user | MajorProject.js 156  MajorProject.js 188 |  |
| var afterCurrency | Variable (Number) | Local | To calculate the amount of money that is going to be added to the currency after selling | MajorProject.js 170 |  |
| var settings | Variable (Boolean, String, Any) | Local | Used to add information to the database | MajorProject.js 207 |  |
| Var tempItem | Variable (String & Number) | Local | Used to identify the information that is to be added to the database | MajorProject.js 232 |  |

**Context Diagram –**

Diagram

Description automatically generated

**Data Flow Diagram –**

Diagram

Description automatically generated

**IPO Chart –**

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSES** | **OUTPUTS** |
|  |  |  |
|  | Initialisation | Images of Objects Displayed + Price + Relevant Information About Stock |
| Letters (Alphabet) | Enter Text Input |  |
| Click (Enter Button) | Countdown (Timer Starts) | Name + Introduction & Context + Display Countdown |
| Click (Purchase Button) | Purchase (Update Currency, Stock Available & In Stock Amount) | Display The Updated Currency, Stock Available & In Stock Amount |
| Click (Sell Button) | Sell (Update Currency, Stock Available & In Stock Amount) | Display The Updated Currency, Stock Available & In Stock Amount |
| Click (Finish Button) | Redirect Page | Display Other html Page + Purchase List |
| Click (Reload Page On Top Left Corner) | Refresh Page | New Turn |