|  |  |  |
| --- | --- | --- |
| **Botany Downs Secondary College**  AS91896 & 91897 - Use advanced techniques and processes to develop a digital technologies outcome (Computer Program)- 12 credits   |  | | --- | | **Student Declaration:**  I Nikhil declare that this assessment is my own work, except where acknowledged appropriately (e.g., use of referencing). I declare that I composed the writing and/or translations in this assessment independently, using the tools and resources defined for use in this assessment. I am aware that any breach of this statement or identified academic misconduct will be followed up and may result in disciplinary action.  Signed: Nikhil  Date: 5/5/22 | |  | |

|  |  |
| --- | --- |
| **Date** | **Log** |
| **5/05/22** | **I started off my project by writing why I am making this program and who my target audience are. I also Set up my Trello and Gant chart** |
| **6/05/22** | **I listed down the program requirements for my program** |
| **10/05/22** | **I finished making a basic flowchart of what my program will be like.** |
| **13/05/22** | **I decomposed my components which made it easier for me to write code, I decomposed my program into 5 different components** |
| **17/05/22** | **I wrote down all the relevant implication in my project and described and explained them** |
| **23/05/22** | **I finished writing the code for my first component** |
| **25/05/22** | **I finished testing and trialling my first component** |
| **26/05/22** | **I started coding my second component** |
| **1/6/22** | **I finished the code for my second component** |
| **2/06/22** | **I finished testing and trialling my second component** |
| **3/06/22** | **I started writing code for my third component** |
| **7/06/22** | **I finished writing the code for my second component** |

|  |
| --- |
| **1. Getting started** |

I will be making a **unit converter**; the purpose of this program will be to convert different units for example feet to cm or millimetres to centimetres. The target users of this program can be anyone, it can help people measure when cooking, it can help builders who want to convert units to a bigger/smaller scale, and it can be helpful to anyone from time to time who would want to convert units for any reason. I have chosen to make this because the convertor on google doesn’t work properly, for example if we put in 180cm to feet and inches we get 5’9 but actually 180cm converts to 5’10.97 even though there are websites which convert correctly they are hard to find. There are probably many other errors on google so I thought I would make a convertor which converts correctly

**What are the program requirements?** (For example, name of the program you will write the code in, variables, functions, libraries etc)

**Software requirements:**

* **I will use functions, so I will not have to rewrite unnecessary code**
* **I will use a word document to document my progress and processes.**
* **I will use other tool such as GitHub and Trello to help me track my progress and remember the processes.**

**Program Requirements:**

* **I will need knowledge in python**
* **I will use tkinter to make a GUI program**
* **I will have to use variables**
* **I will have to use correct python conventions such as correct indentations**
* **I will use lists and dictionaries to hold information**
* **I will use correct syntax**

**What project management and version control tools will you be using?**

(link to Gantt chart and link to Trello)

**What relevant implications will you consider? and how do you plan to address them? (How you have addressed these can be in another table after you develop the outcome)**

|  |  |  |
| --- | --- | --- |
| **Relevant Implication** | **Describe** | **Explain** |
| **Functionality** | **This implication means that my program must work properly with minimal errors, it should function properly without any problems** | **I plan to address this implication by using testing tables to test my code and also by breaking down my code into components so I can refine each part of my code thoroughly** |
| **Usability** | **This implication means that the user needs to be able to use the program without any issues and they will need to be able to use the program with ease** | **I will address this issue by making the buttons bug which will be easy to click. And also I will not add to many colours and keep the colour scheme simple this will ensure that the program will be easy to navigate through.** |
| **Acessibility** | **This implication will require me to consider all people with disabilities. This would mean that the program must not have a lot of user requirements.** | **I plan to address this implication by not adding too many things for the user to do. I plan to address this by using dropdowns the user can click instead of typing what they want.** |
| **Aesthetics** | **This implication is about how my program will look, It will need to look good and will cause the users to have a better overall experience** | **I will address this implication my aiming for a clean and simplistic look on my program. If I add too many colours it might cause problems for users and make it hard for them to navigate but by using a simplistic colour scheme I will be able to make my program look good.** |
| **Sustainability and Futureproofing** | **This implication will require me to ensure that my program can be used in the future without any mistakes and if things change I will need to update the information.** | **I believe I will not need to futureproof this program this is because my program revolves around math concepts that have been used for a lot of years. The math concepts are very unlikely to change therefore my program will already be futureproof.** |
| **Intellectual Propertry** | **This program will require me not use anyone’s work or ideas that have been copyrighted.** | **My program will not use information which is not permitted to me. I will only use math formulas and nobody has a copyright claim over it.** |
|  |  |  |

**Initial flowchart of program:**

**Diagram

Description automatically generated**

**Decompose list of the components of your project:**

(This might be a photo of a mind-map, a typed list, etc)

Graphical user interface

Description automatically generated

Here is an initial look of what I want my program to look like, there will be three dropdowns, one to select type of conversion and 2 to select unit to conversion E.g., mm to m

One of the components will be the dropdown I will have to pick the type of conversion for example I will have to make a dropdown which asks the user if they want to do a weight conversion, distance conversion or liquid conversion etc. Another component will be another dropdown, the purpose of this dropdown is to let the user choose the type of unit they would like to convert. And there will be another dropdown which will let the user choose the type of unit they would like to convert to. Then there will be one component which will let the user type in a number which they want to convert. And finally the last component will be the submit button which will cause the program to convert the chosen units with the correct number.

|  |
| --- |
| **2. Developing for each component:**  These steps will be repeated for each component of your outcome. **Copy and paste the table below** as many times as you need (3 to 6 times would be suitable).  Reminder: you must keep evidence of your trialling and testing, either in a separate document or in this document, under each component table. Evidence could include notes of user feedback, before and after screenshots, testing tables with annotated screenshots and/or screencastify videos.  [Testing document](Testing%20doc.docx) |

|  |  |
| --- | --- |
| **Component 1** | |
| What do I want to achieve? | I want to create dropdown features which will show all the possible types of conversion. For example, when the dropdown is clicked it will show options such as distance, liquid, temperature and more allowing the user to choose one. |
| Do I plan to trial alternative design or techniques? What are they? | I plan to change the design of the dropdown by changing the size and testing how it will look.  The alternative design I can think of can be creating radio buttons instead of dropdown. |
| How did the trialling go? Which design or technique did I choose to use and why? |  |
| How did the testing go? Were there any issues in the code? What was the feedback from users? |  |
| |  |  | | --- | --- | | **Testing the component1** | | | **What was tested** |  | | **Expected** |  | | **Boundary** |  | | **Invalid** |  | | **Next Steps** |  | | |
| **Update of my project management tool at this point (screenshot)** | |
|  | |

|  |
| --- |
| **3. Finishing up** |

**Final test data:** *this is done when the entire program is completed*

|  |  |  |
| --- | --- | --- |
| **Sample Inputs** | **Expected Outputs** | ✔️ |
|  |  |  |
|  |  |  |

*Add extra rows as needed.*

*Attach screenshots or videos that show your program works with the sample inputs above.*

**Relevant implications:** If you haven’t already explained these as you worked on each component, explain them here, and state how they are addressed in your outcome.

**How did planning assist in the development of a high-quality outcome?**

**How did testing and trialling assist in the development of a high-quality outcome?**

|  |
| --- |
| **4. Final checklist: page 6 of the assessment booklet** |