# Assignment 2 – Report Template

|  |  |  |  |
| --- | --- | --- | --- |
| Introduction | | | |
| **Name** | Jiang Zong Zhe | | |
|  | | | | |
| Prior Experience with AI Tools | | | | |
| *If you used any Gen AI in this assignment, please list them in the table below to indicate if you had prior experience using them.*   |  |  | | --- | --- | | *Gen AI* | *Prior Experience* | | *NIL* | *NIL* | | | | | |
|  |  |  |  |

|  |
| --- |
| **F&B Recommendation maximum 350 words** |
| CORRECTNESS  While implementing the search function by location, I ensured that the canteens displayed together are the closest to both locations instead of just one location for higher accuracy. I also ensured that the user input for keywords is case insensitive and accommodates for extra spaces within the input.  THOROUGHNESS  While most keywords were 1 word long, there was a keyword that spans 2 words that needs to be accounted for, as splitting user input based on spaces would incorrectly cause the 2-word keyword to split into 2 sub keywords.  When implementing the search\_by\_price function, distinguishing between inputting 1 store when the max\_price variable is too small, and inputting exactly 1 store that is within the max\_price variable was also crucial.  USER-FRIENDLINESS  I avoided throwing errors as much as possible and replaced them with readable error statements that is easy for users to understand.  EFFICIENCY & REUSABILITY  While implementing the ability to accept “[INPUT] OR [INPUT]”, I had to check that the stall matches the keyword, and whether the stall instance is already within the list. While converting the list to a set to get rid of duplicates would be twice as fast, the stall instance is a list, which is unhashable. A possible way to circumvent this is to convert the stall instance into a tuple, which is hashable.  While implementing the search\_by\_price function, I sorted the filtered stores by price in ascending order, and broke the loop when the price of the stores started becoming higher than the max time to minimise number of iterations through the loop, thereby improving efficiency.  By decomposing one large search function into multiple smaller functions with specialised uses, I could identify certain functions that can also be applied across different search functions, improving reusability. Proper naming of variables and functions, as well as the documentation of the code is also crucial for other programmers to understand the code for reusing in the future.  There were no uses of Gen AI while doing this assignment. |
| Reflection on Use of Gen AIs |
| **maximum 100 words** |
| I did not rely on the use of Gen AI as I was mainly concerned with the documentation and proper implementation of certain Python function, which could be accomplished with a search engine or on public forums, such as Stack Overflow. However, in the future, Gen AI could also be used as a way to brainstorm on efficient ways to execute code or come up with possible flowcharts and pseudocode. |