**Project 3**

**Description:**

Report All the prompts utilize the same format, where the list of ingredients, kitchen utensils, and kitchen appliances are stated, then the menu of dishes with their ingredients is listed, the prompt of what needs to be done is stated, and lastly, an example of a task tree is shown for Google Gemini to learn from. The prompt of what needs to be done differs for all the prompting approaches, and each will be evaluated in this report. The first prompt states the purpose of the first two lists, and simply asks “Utilize what is given, what do I need to do to make this dish?”. In addition, one-shot prompting is utilized where one example of a task tree is provided. This provides fairly decent results, where 17 out of the 30 dishes were generated correctly in the proper task tree format. The pro of this prompting approach is the simplicity and conciseness of the prompt, which allows Google Gemini to understand the task easily and therefore achieve the task without fail. Additionally, the task tree is detailed and provides every motion and input needed to make the dish. However, the con of this is that the generated output utilizes ingredients and appliances that are not listed in the prompt since the prompt does not explicitly state to use the ingredients in the recipe and the tools in the list. In addition, some of the generated output exceeds Google Gemini’s output token limit, and some files are incomplete. Lastly, with only one example of a task tree, there are occasional outputs that do not contain a task tree. The second prompt improves upon the first prompt by stating, “Utilizing the list of available ingredients, utensils, and appliances, following the recipe, what do I need to do to make this dish?”. In addition, this uses a few-shot prompting approach, where two examples of task trees are provided. This prompting approach achieves better results than the first prompting approach where 21 out of the 30 dishes were generated correctly. The pro of this prompting approach is that the prompting approach is still simple and helps Gemini perform the task without confusion. Furthermore, ingredients utilized are in the list of available ingredients and follow the recipe, while maintaining the high level of detail of task trees. In addition, task trees are generated consistently. While this prompting approach fixes the issue with the utilization of resources unavailable to the user, a con of this approach is that the generated output still exceeds the output limit, and therefore the task trees are incomplete. The third prompt improves upon the last two prompts by following the same strategy of the second prompt to explicitly state to use the ingredients and tools in the provided lists. In addition, this uses a few-shot prompting approach, where two examples of task trees are provided. Furthermore, it is mentioned to not use markdown, since some outputs result in markdown, which does not work with JSON files. This prompting approach adds “only focus on high-level actions without specific measurements”. This prompting approach achieves better results than the first two prompting approaches where 26 out of the 30 dishes were generated correctly. The pro of this prompting approach is that the results are consistent, and most of the generated output does not exceed the token limit. In addition, the ingredients utilized are in the list of available ingredients and follow the recipe. Moreover, the task trees are generated consistently with the few-shot approach. While this prompting approach fixes the issue of exceeding the output limit, the con of this approach is its complexity and removal of details in task trees. The prompt is complex by providing small micro rules for Google Gemini to follow, which may confuse the chatbot. Moreover, by focusing on high-level actions on task trees, the trees are less detailed, where actions are simplified, so the output generated is not practical for a robot to follow, this is once again done to meet the output token limit of Google Gemini.