**Introduction ~50**

* Improve on progress report introduction.

**Illustration**

* Main architecture, considering adding more details to the figure, e.g. some reference number for prompts to be explained in later sections, reference for workflow of the system.
* DO NOT GIVE LOW LEVEL DETAILS (DJANGO)

**Background and Related Work ~200**

Prompt engineering is a technique of crafting inputs to elicit desired outputs from generative models. It helps tune the models to give more accurate outputs and improve the overall outputs quality for specific complex tasks, which therefore is widely used in the real of applications of natural language processing. Prompt engineering is not just redesign and reconstruct prompts, it also contains a wide range of skills and patterns to be able to guide the models to give better outputs. White (2023) and his colleagues came up with multiple prompt engineering techniques presented in pattern form that can be used to generate better prompts to successfully improve the outputs of LLM conversations, allowing the model to able to focus on different keywords in different circumstances and hence produces desired outputs. There are difference approaches regards to difference patterns. The Persona Pattern conveys an idea that it’s easier for the models to produce outputs from a certain point of view when telling them to act as persona X, and the Cognitive Verifier Pattern suggests that dividing the question into sub-questions can provide a better answer to the original question. GenPerfectTrip used several prompt engineering techniques to instruct the GPT model to give the optimal outputs during the trip plan generation process, which include the Persona Pattern and the Cognitive Verifier Pattern that mentioned above.

**Data and Data Processing ~300**

* Example of testing data
* DO NOT GIVE LOW LEVEL DESCRIPTION OF THE DATA USED (JSON)
* Explain how system process data to be used for hotel and pan generation.

**Architecture and Software ~ 700**

* Overall description of the architecture
* Detailed explanation of each prompt (how are they constructed – with examples, where are they used - refer to the architecture illustration)
* DO NOT FOCUS ON LOW LEVEL DETAILS

The JSON prompt constructor's purpose is to guide the GPT in converting user input into JSON format. It constructs the system prompt instructing GPT to extract information such as destination, trip duration, and price range, from the user input (if avaliable) and return a JSON. There are some specific rules for constructing the JSON in this system prompt. For instance, The specified price range must be in the format like ``100-200” and can be calculated by dividing the budget by the number of nights.

The data processor’s role is to utilize the information provided in JSON for web scraping. It parses the JSON, extracts relevant information, and uses it to form a URL that can access an available web page on BOOKING.com for search results for hotels. Upon sending a request to this URL, the data processor retrieves the HTML content, which contains the details about the listed hotels. It then parses HTML content to extract information such as name, location, price, rating, and distance to the location. Ultimately, it outputs a list of hotels information to the Preliminary Plan Prompt Constructor.

The preliminary plan prompt constructor creates a system prompt instructing the model in a two-step plan generation process. Firstly, it guides the model to select hotels from the provided list, shaping the accommodation aspect of the plan. Secondly, it directs the model to generate a detailed itinerary for each day of the trip. Subsequently, the GPT model returns the preliminary plan to the user.

The improved plan generator shares a similar structure with the preliminary plan generator but diverges in terms of logic and content flow.

The Improved Plan Generator shares a similar structure with the Preliminary Plan Generator but diverges in terms of logic and content flow. It primarily comprises three components: the Improved Plan Prompt Constructor, Data Processor, and Hotel Improvement Prompt Constructor.

The Improved Plan Prompt Constructor is responsible for generating a system prompt that guides the GPT model to produce either a JSON or an enhanced plan based on the user's preference to modify hotels or attractions. An excerpt from the prompt illustrates this functionality: "You are a trip planner tasked with refining the provided plan according to the user's specifications. Your changes should be limited to details based on the user's requirements. If the user wishes to enhance activities, do not alter hotel details; if the user wants to modify hotel information, output ONLY a JSON file." Utilizing the Persona pattern mentioned earlier, the model is instructed to assume the role of a trip planner, facilitating the enhancement of the plan from a trip planner's perspective. The prompt then directs the model to decide whether to generate a JSON or an improved plan by replacing the existing attractions with the user's specified points of interest. The preliminary plan is seamlessly integrated into the system prompt, allowing the model to make enhancements to the attractions before returning the refined plan to the user.

In the event that the GPT determines the user's intention to improve hotels, it considers the user's new requirements and generates a JSON file using the same rules as the JSON prompt constructor. In this scenario, the Data Processor executes a process akin to that in the Preliminary Plan Generator, producing a list of hotel information. This information is then delivered to the Hotel Improvement Prompt Constructor, which constructs a system prompt incorporating the provided hotel list, the preliminary plan, and instructions for the model to focus solely on changing the hotels.

.

**Baseline Model and Comparison ~ 300**

* Detail description of the accuracy criteria
* Show how we evaluate the results.

**Qualitative Result ~ 200**

* Example output
* Discuss how the output is evaluated and how well it matches to the criteria.

**Quantitative Result ~ 100**

* Overall accuracy of the system, and how well it did based on the criteria.
* Comparison to the GPT plugins

**Discussion and Learning ~100**

* Commet on the overall success and to what extent GPT is useful for
* How can this system be further developed

GenPerfectTrip demonstrated notable success in generating comprehensive and personalized trip plans for users, encompassing hotel selections and detailed trip itineraries, which perfectly align with their requirements and preferences such as the destination, trip duration, budget constraints, and specific attractions of interest. One notable aspect is the application’s ability to seamlessly combine both hotel selection and itinerary planning in a single solution, which set GenPerfectTrip apart, especially in comparison to the existing Expedia's plugin. Through the development of GenPerfectTrip, the most significant learning is the critical role of prompt engineering and the boundless possibilities that GPT offers. When starting a similar project in the future, the development team would build it on the top of this project and utilize prompt engineering and GPT to implement more detailed trip arrangements such as hotel booking, ticket purchasing, UBER requests, and restaurant reservations that come with the plan generated. With just a single tap, users can have all aspects of their trip meticulously prepared.

**Individual Contribution ~50**

* Was responsible for constructing the system prompt for generating the preliminary plan.
* Set up the Django framework
* Implemented back-end logic and API integrations.
* Implemented the dynamic web scraping function by utilizing the Beautiful Soup library, which included JSON parsing, URL construction, web page parsing, and data extraction.
* Designed and implemented the responsive front-end web interface.
* integrated JQuery to handle front-end side logic and dynamic content updates.