

Research regarding User Input Processing:

The purpose of this research document is to provide detailed understanding of how HEPIK will process the users input and provide an accurate response. Below we have broken down the process in smaller chunks for simplification.

Parsing the Input:

The core idea of Input processing lies in identifying two pieces of critical and essential information from user's input: the location for which the user wants the information and the time of the year for that specific location. To accomplish this very task, we think that using of regular expressions (regex) will be most efficient. This system employs a combination of keyword searching.

Regular Expressions:

Regex is used to identify patterns in the user's text that match common way's locations and times are mentioned. For instance, a regex pattern can detect city names by looking for capitalization patterns or by matching against a precompiled list of known city names. Similarly, dates and times can be identified through formats like "DD/MM/YYYY", "tomorrow", or "next week".

Example Regex for Location:

```
([A-Z][a-z]+(?:\s[A-Z][a-z]+)*)
```

This pattern matches city names that start with a capital letter, allowing for multi-word city names.

Example Regex for Time:

```
((today|tomorrow|(\d{1,2})[/-](\d{1,2})[/-](\d{2,4}))\b)
```

This captures keywords like "today", "tomorrow", or date formats.

User Input Processing for Integration with Open-Meteo:

Adjusting the Parsing Logic:

Continue to identify the location using regex as you currently do.

Utilize our existing geocoding API to resolve the city name to its geographic coordinates. This step will provide the coordinates needed for querying the Open-Meteo API.

Validation and Error Handling:*Time Frame Validation:*

HEPIK ensure's that the user's input for the forecast time frame falls within the supported range of Open-Meteo, which provides forecasts up to 14 days.

Handling API Responses:

While Open-Meteo doesn't require an API key, HEPIK will handle potential errors in the API response gracefully. This includes network errors, unexpected response formats, or situations where forecast data cannot be retrieved. HEPIK will respond with user-friendly messages to guide the user accordingly.

Clarification and Fallback Mechanisms:

Implement clarification prompts for ambiguous inputs. For example, if the location can't be resolved or the time frame is not supported, HEPIK's prompt's the user to provide clearer input.

We will implement fallback responses for scenarios where forecast data cannot be retrieved. This could include suggesting alternative locations or time frames that are supported by Open-Meteo.

By incorporating these adjustments into HEPIK's user input processing, we will ensure that HEPIK seamlessly integrates with Open-Meteo, providing accurate weather forecasts to users while handling various input scenarios and potential errors gracefully.