

Algorithm for Pizza Order Chatbot

Description of the project

1. Initialization:

- Set the environment variable ``GOOGLE_API_KEY`` with the actual API key.
- Import necessary libraries (``os``, ``panel``, ``google.generativeai``).
- Initialize Panel extension.

2. API Key Check:

- Retrieve the API key from the environment variable ``GOOGLE_API_KEY``.
- Raise an error if the API key is not found.

3. GUI Setup:

- Create a text input widget ``inp`` for user input.
- Create a button ``button_conversation`` to initiate the chat.
- Initialize an empty list ``context`` to store conversation history.

4. Prompt Function:

- Define the function ``prompt(user_input)`` to generate the prompt for the AI model based on the user's input. The prompt includes instructions for the OrderBot and the current user question.

5. Collect Messages Function:

- Define the function ``collect_messages(event)``:
 - Retrieve user input from ``inp``.
 - Generate the complete prompt by calling ``prompt(user_input)``.
 - Initialize the AI model (``gemini-1.5-flash``) from ``google.generativeai``.
 - Generate a response from the AI model using ``model.generate_content(text)``.
 - Extract the response text from the model's output.
 - Append the user and assistant messages to the ``context`` list.
 - Clear the text input field ``inp``.
 - Create a conversation history display by formatting the ``context`` list.
 - Return the formatted conversation history as a Markdown pane.

6. Bind Button Event:

- Bind the button click event `button_conversation`` to the `collect_messages`` function using `pn.bind``.

7. Layout:

- Create the dashboard layout with the text input, button, and conversation display.

8. Serve Application:

- Serve the Panel application using `serve(dashboard, show=True)``.

Algorithm

1. Start.

2. Initialize environment variable `GOOGLE_API_KEY``.

3. Import necessary libraries:

- `os``, `panel as pn``, `google.generativeai as genai``.

4. Initialize Panel extension.

5. Retrieve API key from environment variable `GOOGLE_API_KEY``.

- If no API key is found, raise a `ValueError``.

6. Create GUI Elements:

- Text input widget `inp``.
- Button widget `button_conversation``.
- Initialize empty list `context``.

7. Define prompt function `prompt(user_input)``:

- Return formatted prompt string for AI model.

8. Define message collection function `collect_messages(event)``:

- Retrieve user input from `inp``.
- Generate complete prompt using `prompt(user_input)``.
- Initialize AI model `gemini-1.5-flash``.
- Generate response from AI model using `model.generate_content(text)``.
- Extract response text.
- Append user and assistant messages to `context``.

- Clear text input field ``inp`` .
 - Create and return formatted conversation history.
9. Bind button click event ``button_conversation`` to ``collect_messages`` .
 10. Create dashboard layout with text input, button, and conversation display.
 11. Serve the Panel application using ``serve(dashboard, show=True)`` .
 12. End.