Fiddle Engineering Intern Take Home Test

Build a **Tone Picker Text Tool** where users can adjust the tone of text (e.g., from formal to casual) using a picker interface. The tool should integrate with the Mistral AI API to change the tone and provide smooth interactions with undo/redo functionality..

**Requirements**

**Core Functionality**

Reference video: <https://www.youtube.com/shorts/pswPT-iFHEE>

* **Text Editor**: Create a text editor where users can input and edit text.
* **Tone Picker**: Implement a 2\*2 matrix that adjusts the tone of the text (e.g., formal to casual) by making API calls to Mistral AI.
* **Undo/Redo**: Implement undo/redo functionality to track and revert tone changes.
* **Responsive UI**: Ensure the UI responds smoothly, displaying appropriate loading states during API calls.

For the sake of this test, have an editable text field to the left of your interface and the picker to the right.

You do not have to include the presets (“Executive”, “Technical”, “Educational” and Basic) shown in the video. But the reset button and the 2\*2 matrix picker are required.

**Technical Requirements**

* **Stack**: React, the Mistral small model (it is free, [docs](https://docs.mistral.ai/api/)). The rest of the stack is up to you.
* **Backend**: Implement a lightweight backend to manage API security and request caching in a framework of your choice
* **Error Handling**: Gracefully handle API failures and edge cases (e.g., network errors, invalid API responses).
* Optional:
  + **Persistence**: Store text and revision history in local storage to preserve data across sessions.

**Design & UX**

* **Intuitive Interface**: Create a clean interface with smooth interactions.
* **Visual Feedback**: Provide clear feedback (e.g., loading indicators or tone change previews) while the API processes the request.
* **Error state handling:** Surface and show errors to users

**Evaluation Criteria**

* **Code Quality & Organization**: Is the code clean, well-organized, and easy to read?
* **State Management**: How well is state managed, especially for undo/redo functionality and tone adjustments?
* **API Integration**: How well does the application integrate with the Mistral AI API? Is the integration smooth and reliable?
* **UI/UX Design**: How polished is the UI? Does it offer a seamless and intuitive user experience?
* **Documentation**: Is the documentation clear and thorough, helping others to understand your choices and set up the project?
* **Error Handling**: How robust is the error handling? Does the app gracefully handle API failures and edge cases?

**Submission**Please submit a GitHub repository containing your complete source code, including:

* **README** with setup instructions: How you document is super important because this is a remote role. Please make a concise and clear readme.
* A link to the app deployed. You can use a free service like Vercel or Render for this
* A video recording of the final output
* A brief explanation of:
  + Technical architecture decisions you made. If you made any trade offs, document and explain them
  + Describe how state is managed, especially with regard to the undo/redo functionality.
  + Detail your approach to error handling and handling edge cases.

All of the above is required to evaluate your profile.