You are an experienced Web Developer.

You need to generate the frontend (HTML & JavaScript) for a web page that will call a JSON web API to chat with the GPT AI model, i.e we are generating a clone of ChatGPT.

You need to use bootstrap CSS library.

You need to create one single HTML file that only contains HTML, Vanilla JavaScript and bootstrap classes.

The content of the page shall be included in a div with the container bootstrap class. The container needs to have rounded corners.

The background color of the page shall be light grey while the container shall be dark grey.

You need to create a header that has a logo (logo.png) and below it a title of the page, both are centered.

The page includes a form at the top to collect the System Prompt and the User Prompt and also has a Send & Clear buttons (Both the System Prompt & the User Prompt can be changed between calls).

Below the form there is a list of div’s to display the history of the User input and the returned output (content) from the JSON API putting the last at the top. The User Prompt starts with “Me: ” while the returned content starts with “Chat Bot: ”

When the submit button is clicked on the form a vanilla JavaScript code shall read the System Prompt (Prefilled to this default “You reply as concisely as possible.”) & the User Prompt inputs and call the following url “http://localhost:4000/openai/chatgpt\_clone” and pass the messages list (which is described bellow) in the body as json object.

The messages list has the following format.

1. It is a JSON array of Objects.
2. The first Object has a key of “role” with a value of “system” and a key of “content” with a value of the provided System Prompt in the form.
   1. An entry of role system only appears just once at the top of the messages list.
3. The following object in the messages list has a key of “role” with a value of “user” and a key of “content” with a value of the provided User Prompt in the form.

When the API is returned it has a json object with a key of “content” which shall update the display field

The application shall remember the User Prompts and the returned content from the previous API calls.

The messages list for the subsequent calls shall include

1. It is a JSON array of Objects.
2. The first Object has a key of “role” with a value of “system” and a key of “content” with a value of the provided System Prompt in the form.
   1. An entry of role system only appears just once at the top of the messages list.
   2. The user can modify the system role content between calls
3. The following object in the messages list has a key of “role” with a value of “user” and a key of “content” with a value of the previously provided User Prompt in the form.
4. The following object in the messages list has a key of “role” with a value of “assistant” and a key of “content” with a value of the previously returned content from the API.
5. The following object in the messages list has a key of “role” with a value of “user” and a key of “content” with a value of the current provided User Prompt in the form.
6. For each of these subsequent calls, the User Prompt & the returned content of all previous calls are added to the messages list.

An example messages list

messages = [

{ 'role': 'system', 'content': ‘You reply as concisely as possible.’ },

{ 'role': 'user', 'content': ‘What is the capital of France’ },

{ 'role': 'assistant', 'content': ‘Paris’ },

{ 'role': 'user', 'content': What is its population },

{ 'role': 'assistant', 'content': ’10 Million’ },

{ 'role': 'user', 'content': ‘What is its area?’ }

]

During the API call show a spinner and disable the fields on the form.

In the history, make the background of the user entry dark and the background of the assistant entry light.

In the history always add new items at the top.

when the user clicks on Enter inside the User Prompt field the form shall submit

Please include all the code in one file called chatgpt\_clone.html