1. Configuration & Constants

- **Voiceflow Project IDs**
    - voiceflowProjectID – The unique ID of the Voiceflow project.
    - voiceflowVersionID – Defaults to “production,” but could be changed to another environment.
- **Supabase**
    - proxyUrl – An endpoint on Supabase used as a proxy for requests.
    - supabaseAnonKey – The Supabase anonymous key for authorization.
- **Unique User Session**
    - uniqueId – A generated, random session ID to associate messages and transcripts with a particular user session.
- **Allowed Domains**
    - allowedDomains – An array of domain names that are authorized to use this script. The script checks the current site’s hostname against this list.

---

2. Device & Browser Detection

- **getDeviceInfo()** – Returns a string "iOS", "Android", or "Desktop" based on the navigator.userAgent.
- **getOperatingSystem()** – Returns a string representing the OS, such as "Windows", "macOS", "Linux", "Android", "iOS", or "Unknown OS".
- **getBrowser()** – Returns a string for the browser type, such as "Chrome", "Firefox", "Safari", "Edge", "Opera", or "Unknown Browser".

These functions help record metadata about the user environment in transcripts.

---

3. Domain Normalization & Authorization

- **normalizeDomain()** – Strips out protocols (http://, https://), “www.”, and extra paths or query parameters to compare just the root domain.
- **isDomainAllowed()** – Checks if the current hostname is allowed by comparing it to the array in allowedDomains. Supports matching subdomains or parent domains.

If the domain is **not** allowed, the script displays an error and stops execution to prevent the chatbot from running on unauthorized domains.

---

4. DOM Elements & Initialization

- The script references various HTML elements by document.getElementById(...), such as:
    - assistantMessageContainer – Where assistant messages are displayed.
    - inputContainer, input – The user input area.
    - submitButton – The “Send” button.
    - thinkingScreen – Overlay indicating the bot is “thinking.”
    - errorContainer – Container to display errors.
    - buttonContainer – Container for quick-reply buttons or choice buttons.

Upon DOMContentLoaded:

- **initializeChat()** is called to:
    1. Check if the domain is allowed.
    2. Create a transcript with Voiceflow if allowed.
    3. Send a “launch” action to start the conversation.

Event listeners are also set up on the input and the send button to handle user submission.

5. Transcript Management

1. **createTranscript()** – Makes a request to the proxy to create a new transcript in Voiceflow’s system.
    - Stores the newly created transcriptID and creatorID in localStorage.
2. **updateTranscriptName()** – Updates the transcript name after creation, using the stored transcriptID and creatorID.

These actions are optional but useful for naming or tracking the conversation in Voiceflow’s logs.

---

6. Main Chat Logic

- **generateUniqueId()** – Produces a unique ID for the session by combining a random string with the current timestamp.
- **initializeChat()** – (Described above) sets up domain checking, transcript creation, and triggers the initial “launch” action with interact().
- **toggleSendButton()** – Handles showing and hiding the “Send” button, depending on whether there is user input.

6.1 Sending & Handling User Messages

- **sendMessage(userInput)** – Main function to handle when a user submits text. It:
    1. Clears errors.
    2. Hides or resets UI elements (assistant container, input container).
    3. Shows the “thinking” overlay.
    4. Calls interact({ type: 'text', payload: userInput }) to send the message to Voiceflow.

6.2 Interacting with the Voiceflow Proxy

- **interact(action)** – Performs a fetch to proxyUrl with the action object. The response is handled as an **SSE (Server Sent Event)** stream with processStream().

---

7. Processing the Event Stream

- **processStream(stream)** – Reads and decodes the server-sent event stream line by line, identifying JSON objects.
  - Each chunk is processed by handleStreamEvent(event).

7.1 Event Handlers

1. **handleTextResponse(item)**
   - Clears the “thinking” screen, processes text, animates typed-out words, and re-displays the input container for the user’s next input.
2. **handleChoiceResponse(item)**
   - Creates interactive choice buttons (i.e., Quick Replies). When clicked, these buttons call the handleButtonClick() function to send the corresponding request to Voiceflow.
3. **handleEnd()**
   - Called when the conversation or flow ends. Hides the “thinking” overlay and hides the input container.
4. **handleErrorSignal(event)**
   - Displays an error message from the server and stops the conversation flow if there’s a major issue.

---

8. UI/Animation Helpers

- **clearInput()** – Resets the user input field.

- **showThinkingScreen() / hideThinkingScreen()** – Show or hide the “thinking” overlay (using GSAP animations).
- **dynamicallyFormatMessage(message)** – Converts markdown-like syntax (**bold**, *italic*) into HTML, and line breaks into <br>.
- **displayErrorMessage(message)** – A fallback function to show errors if something goes wrong while sending or receiving data.

---

9. GSAP Usage

GreenSock (GSAP) is used for smooth animations:

- Fading in/out the thinking screen.
- Typing effect on text.
- Hiding/showing message containers, input areas, and buttons with transitions.