

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Chat with Dominic Toretto</title>
  <link
    href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.3/dist/tailwind.min.css"
    rel="stylesheet"
  />
</head>
<body class="bg-blue-100 flex flex-col min-h-screen">
  <div class="flex-grow container mx-auto py-6 px-4">
    <div class="h-full w-full max-w-2xl mx-auto">
      <div id="chatbox" class="flex flex-col items-start overflow-y-auto h-full p-1 rounded"></div>
    </div>
    <div class="sticky bottom-0 w-full">
      <div class="flex justify-center">
        <div class="px-2 w-full max-w-2xl">
          <div class="flex flex-col my-5">
            <input
              class="shadow flex-grow rounded p-2 mb-2 shadow appearance-none border rounded
w-full text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
              id="messageInput"
              type="text"
              placeholder="Type your message"
            />
            <div class="flex justify-between space-x-2">
              <button
                class="bg-red-500 hover:bg-red-400 text-white font-bold py-2 px-4 border-b-4
border-red-700 hover:border-red-500 rounded w-full sm:w-auto"
                id="clearButton"
              >
                Clear Chat
              </button>
              <button
                class="bg-blue-500 hover:bg-blue-400 text-white font-bold py-2 px-4 border-b-4
border-blue-700 hover:border-blue-500 rounded w-full sm:w-auto"
                id="sendButton"
              >
                Ask Dom!
              </button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
<script>
    const chatbox = document.getElementById("chatbox");
    const messageInput = document.getElementById("messageInput");
    const sendButton = document.getElementById("sendButton");
    const clearButton = document.getElementById("clearButton");
    const chatId = crypto.randomUUID();
    let websocket = null;

    let receiving = false;
    // Change the main chatbot prompt here
    const systemPrompt = "You are Dominic Toretto, a street racer and skilled mechanic known
for your loyalty to family and friends.";

    // A function to create a message element based on the text and alignment
    function createMessageElement(text, alignment) {
        const messageElement = document.createElement("div");

        messageElement.className = `inline-block my-2.5 p-2.5 rounded border ${
            alignment === "left" ? "self-start bg-white" : "self-end bg-blue-200"
        }`;
        messageElement.textContent = text;
        return messageElement;
    }

    // A function to connect to the WebSocket server and send/receive messages
    function connectWebSocket(message, initChat) {
        receiving = true;
        sendButton.textContent = "Cancel";
        const url = "wss://backend.buildpicoapps.com/api/chatbot/chat";
        websocket = new WebSocket(url);

        websocket.addEventListener("open", () => {
            websocket.send(
                JSON.stringify({
                    chatId: chatId,
                    appId: "something-shake",
                    systemPrompt: systemPrompt,
                    message: initChat ? "A very short welcome message from Dominic Toretto" : message, //
                    Change the welcome message here
                })
            );
        });
    }

```

```
    })  
  );  
});
```

```
const messageElement = createMessageElement("", "left");  
chatbox.appendChild(messageElement);
```

```
websocket.onmessage = (event) => {  
  messageElement.textContent += event.data;  
  chatbox.scrollTop = chatbox.scrollHeight;  
};
```

```
websocket.onclose = (event) => {  
  if (event.code === 1000) {  
    receiving = false;  
    sendButton.textContent = "Ask Dom!";  
  } else {  
    messageElement.textContent += "Error getting response from server. Refresh the page  
and try again.";  
    chatbox.scrollTop = chatbox.scrollHeight;  
    receiving = false;  
    sendButton.textContent = "Ask Dom!";  
  }  
};  
}
```

```
// A function to create a welcome message when the chat starts  
function createWelcomeMessage() {  
  connectWebSocket("", true);  
}
```

```
// Event listeners
```

```
sendButton.addEventListener("click", () => {  
  if (!receiving && messageInput.value.trim() !== "") {  
    const messageText = messageInput.value.trim();  
    messageInput.value = "";  
    const messageElement = createMessageElement(messageText, "right");  
    chatbox.appendChild(messageElement);  
    chatbox.scrollTop = chatbox.scrollHeight;  
  
    connectWebSocket(messageText, false);  
  } else if (receiving && websocket) {  
    websocket.close(1000);  
    receiving = false;
```

```
        sendButton.textContent = "Ask Dom!";
    }
});

messageInput.addEventListener("keydown", (event) => {
    if (
        event.key === "Enter" &&
        !receiving &&
        messageInput.value.trim() !== ""
    ) {
        event.preventDefault();
        sendButton.click();
    }
});

clearButton.addEventListener("click", () => {
    chatbox.innerHTML = "";
});

createWelcomeMessage();
</script>
</body>
</html>
```