<!DOCTYPE html>

<html lang="en" dir="ltr">

  <head>

    <meta charset="utf-8">

    <title>ChatBot</title>

    <link rel="stylesheet" href="style.css">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@48,400,0,0" />

    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Rounded:opsz,wght,FILL,GRAD@48,400,1,0" />

    <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">

    <script src="script.js" defer></script>

    <style>

        body {

            font-family: Arial, sans-serif;

            background: #E3F2FD;

            margin: 0;

            padding: 0;

        }

        .navbar {

            display: flex;

            justify-content: space-between;

            align-items: center;

            padding: 10px 20px;

            background-color: white;

            border-bottom: 1px solid #ccc;

            position: sticky;

            top: 0;

            z-index: 1000;

        }

        .navbar-logo img {

            width: 50px; /\* Reduced size \*/

            height: auto;

        }

        .navbar-links {

            display: flex;

            gap: 30px;

            position: absolute;

            left: 50%;

            transform: translateX(-50%);

        }

        .navbar-links a {

            text-decoration: none;

            color: black;

            font-size: 16px;

        }

        .navbar-links a:hover {

            color: green; /\* Change to green on hover \*/

        }

        .content {

            display: flex;

            justify-content: center;

            align-items: center;

            height: calc(100vh - 60px); /\* Adjust height to fill the remaining space \*/

            background: url('img2.jpg') no-repeat center center fixed; /\* Updated image source \*/

            background-size: cover; /\* Make the image cover the entire page \*/

        }

    </style>

  </head>

  <body>

    <nav>

      <div class="navbar">

        <div class="navbar-logo"><img src="fis-logo.png" alt="Logo"></div>

        <div class="navbar-links">

          <a href="/home">Home</a>

          <a href="/form">Raise a Ticket</a>

          <a href="/tickets">My Tickets</a>

        </div>

      </div>

    </nav>

    <div class="content"></div>

    <button class="chatbot-toggler">

      <span class="material-symbols-rounded">mode\_comment</span>

      <span class="material-symbols-outlined">close</span>

    </button>

    <div class="chatbot">

      <header>

        <h2>ChatBot</h2>

        <span class="close-btn material-symbols-outlined">close</span>

      </header>

      <ul class="chatbox">

        <li class="chat incoming">

          <span class="material-symbols-outlined">smart\_toy</span>

          <p>Hi there 👋<br>How can I help you today?</p>

        </li>

      </ul>

      <div class="chat-input">

        <textarea placeholder="Enter a message..." spellcheck="false" required></textarea>

        <span id="send-btn" class="material-symbols-rounded">send</span>

        <span id="voice-btn" class="material-symbols-rounded voice-icon">mic</span>

      </div>

    </div>

  </body>

</html>

/\* Google Fonts \*/

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: "Poppins", sans-serif;

}

body {

  background: #E3F2FD;

}

.navbar {

  display: flex;

  justify-content: space-between;

  align-items: center;

  padding: 10px 20px;

  background-color: white;

  border-bottom: 1px solid #ccc;

  position: sticky;

  top: 0;

  z-index: 1000;

}

.navbar-logo img {

  width: 50px;

  height: auto;

}

.navbar-links {

  display: flex;

  gap: 30px;

  position: absolute;

  left: 50%;

  transform: translateX(-50%);

}

.navbar-links a {

  text-decoration: none;

  color: black;

  font-size: 16px;

}

.navbar-links a:hover {

  color: green;

}

.content {

  display: flex;

  justify-content: center;

  align-items: center;

  height: calc(100vh - 60px);

  background: url('img2.jpg') no-repeat center center fixed;

  background-size: cover;

}

.chatbot-toggler {

  position: fixed;

  bottom: 30px;

  right: 35px;

  outline: none;

  border: none;

  height: 50px;

  width: 50px;

  display: flex;

  cursor: pointer;

  align-items: center;

  justify-content: center;

  border-radius: 50%;

  background: #4CAF50;

  transition: all 0.2s ease;

}

body.show-chatbot .chatbot-toggler {

  transform: rotate(90deg);

}

.chatbot-toggler span {

  color: #fff;

  position: absolute;

}

.chatbot-toggler span:last-child,

body.show-chatbot .chatbot-toggler span:first-child {

  opacity: 0;

}

body.show-chatbot .chatbot-toggler span:last-child {

  opacity: 1;

}

.chatbot {

  position: fixed;

  right: 35px;

  bottom: 90px;

  width: 400px;

  height: 500px;

  background: #fff;

  border-radius: 15px;

  overflow: hidden;

  opacity: 0;

  pointer-events: none;

  transform: scale(0.5);

  transform-origin: bottom right;

  box-shadow: 0 0 128px 0 rgba(0, 0, 0, 0.1),

              0 32px 64px -48px rgba(0, 0, 0, 0.5);

  transition: all 0.1s ease;

}

body.show-chatbot .chatbot {

  opacity: 1;

  pointer-events: auto;

  transform: scale(1);

}

.chatbot header {

  padding: 16px 0;

  text-align: center;

  color: #fff;

  background: #4CAF50;

  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

  position: relative;

}

.chatbot header span {

  position: absolute;

  right: 15px;

  top: 50%;

  transform: translateY(-50%);

  cursor: pointer;

}

header h2 {

  font-size: 1.4rem;

}

.chatbot .chatbox {

  overflow-y: auto;

  height: 410px;

  padding: 30px 20px 100px;

}

.chatbot :where(.chatbox, textarea)::-webkit-scrollbar {

  width: 6px;

}

.chatbot :where(.chatbox, textarea)::-webkit-scrollbar-track {

  background: #fff;

  border-radius: 25px;

}

.chatbot :where(.chatbox, textarea)::-webkit-scrollbar-thumb {

  background: #ccc;

  border-radius: 25px;

}

.chatbox .chat {

  display: flex;

  list-style: none;

}

.chatbox .outgoing {

  margin: 20px 0;

  justify-content: flex-end;

}

.chatbox .incoming span {

  width: 32px;

  height: 32px;

  color: #fff;

  cursor: default;

  text-align: center;

  line-height: 32px;

  align-self: flex-end;

  background: #4CAF50;

  border-radius: 4px;

  margin: 0 10px 7px 0;

}

.chatbox .chat p {

  white-space: pre-wrap;

  padding: 12px 16px;

  border-radius: 10px 10px 0 10px;

  max-width: 75%;

  color: #fff;

  font-size: 0.95rem;

  background: #4CAF50;

}

.chatbox .incoming p {

  border-radius: 10px 10px 10px 0;

  color: #000;

  background: #f2f2f2;

}

.chatbox .chat p.error {

  color: #721c24;

  background: #f8d7da;

}

.chatbot .chat-input {

  display: flex;

  gap: 5px;

  position: absolute;

  bottom: 0;

  width: 100%;

  background: #fff;

  padding: 3px 20px;

  border-top: 1px solid #ddd;

}

.chat-input textarea {

  height: 55px;

  width: 100%;

  border: none;

  outline: none;

  resize: none;

  max-height: 180px;

  padding: 15px 15px 15px 0;

  font-size: 0.95rem;

}

.chat-input span {

  align-self: flex-end;

  color: #4CAF50;

  cursor: pointer;

  height: 55px;

  display: flex;

  align-items: center;

  font-size: 1.35rem;

}

/\* Voice Icon Specific Styling \*/

.voice-icon {

  margin-left: 5px;

}

/\* Blinking effect \*/

.blinking {

  animation: blink-animation 1s infinite;

}

@keyframes blink-animation {

  0%, 100% { opacity: 1; }

  50% { opacity: 0.3; }

}

@media (max-width: 490px) {

  .chatbot-toggler {

    right: 20px;

    bottom: 20px;

  }

  .chatbot {

    right: 0;

    bottom: 0;

    height: 100%;

    border-radius: 0;

    width: 100%;

  }

  .chatbot .chatbox {

    height: 90%;

    padding: 25px 15px 100px;

  }

}

const chatbotToggler = document.querySelector(".chatbot-toggler");

const closeBtn = document.querySelector(".close-btn");

const chatbox = document.querySelector(".chatbox");

const chatInput = document.querySelector(".chat-input textarea");

const sendChatBtn = document.querySelector("#send-btn"); // updated: send button has id now

const voiceBtn = document.querySelector("#voice-btn"); // added: mic button by ID

let userMessage = null;

// const API\_KEY = "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"; // old api

const API\_KEY = "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"; // new api

const inputInitHeight = chatInput.scrollHeight;

// Voice assistant blinking logic

let isRecording = false;

voiceBtn.addEventListener("click", () => {

    isRecording = !isRecording;

    voiceBtn.classList.toggle("blinking", isRecording);

    if (isRecording) {

        console.log("Voice assistant activated 🎤");

        // Optional: Add actual speech recognition logic here

    } else {

        console.log("Voice assistant deactivated ⛔");

    }

});

const createChatLi = (message, className) => {

    const chatLi = document.createElement("li");

    chatLi.classList.add("chat", `${className}`);

    let chatContent = className === "outgoing"

        ? `<p></p>`

        : `<span class="material-symbols-outlined">smart\_toy</span><p></p>`;

    chatLi.innerHTML = chatContent;

    chatLi.querySelector("p").textContent = message;

    return chatLi;

};

const generateResponse = (chatElement) => {

    const API\_URL = "https://api.openai.com/v1/chat/completions";

    const messageElement = chatElement.querySelector("p");

    const requestOptions = {

        method: "POST",

        headers: {

            "Content-Type": "application/json",

            "Authorization": `Bearer ${API\_KEY}`

        },

        body: JSON.stringify({

            model: "gpt-3.5-turbo",

            messages: [{ role: "user", content: userMessage }]

        })

    };

    fetch(API\_URL, requestOptions).then(res => res.json()).then(data => {

        messageElement.textContent = data.choices[0].message.content.trim();

    }).catch(() => {

        messageElement.classList.add("error");

        messageElement.textContent = "Oops! Something went wrong. Please try again.";

    }).finally(() => chatbox.scrollTo(0, chatbox.scrollHeight));

};

const handleChat = () => {

    userMessage = chatInput.value.trim();

    if (!userMessage) return;

    chatInput.value = "";

    chatInput.style.height = `${inputInitHeight}px`;

    chatbox.appendChild(createChatLi(userMessage, "outgoing"));

    chatbox.scrollTo(0, chatbox.scrollHeight);

    setTimeout(() => {

        const incomingChatLi = createChatLi("Thinking, Please Wait...", "incoming");

        chatbox.appendChild(incomingChatLi);

        chatbox.scrollTo(0, chatbox.scrollHeight);

        generateResponse(incomingChatLi);

    }, 600);

};

chatInput.addEventListener("input", () => {

    chatInput.style.height = `${inputInitHeight}px`;

    chatInput.style.height = `${chatInput.scrollHeight}px`;

});

chatInput.addEventListener("keydown", (e) => {

    if (e.key === "Enter" && !e.shiftKey && window.innerWidth > 800) {

        e.preventDefault();

        handleChat();

    }

});

sendChatBtn.addEventListener("click", handleChat);

closeBtn.addEventListener("click", () => document.body.classList.remove("show-chatbot"));

chatbotToggler.addEventListener("click", () => document.body.classList.toggle("show-chatbot"));