DAY – 21, 22

27, 28 August 2025

FRONTEND OVERVIEW (Chatbot UI)

The frontend of your chatbot is a **floating popup chatbot interface** that appears on the bottom-right corner of the website. It allows users to interact with the chatbot in **English or Punjabi** through a clean and simple user interface.

Frontend Technologies Used

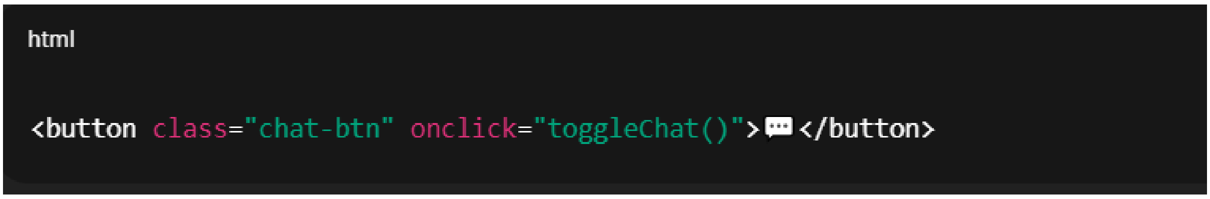
- **HTML** – Structure of the chatbot
- **CSS** – Styling and layout of the chatbot
- **JavaScript** – User interaction, API calls, and message handling
- **Flask Backend** – Used only to fetch responses via API

1. index.html – Chatbot Structure

This file defines the **layout and structure** of the chatbot.

Key Components

Floating Chat Button

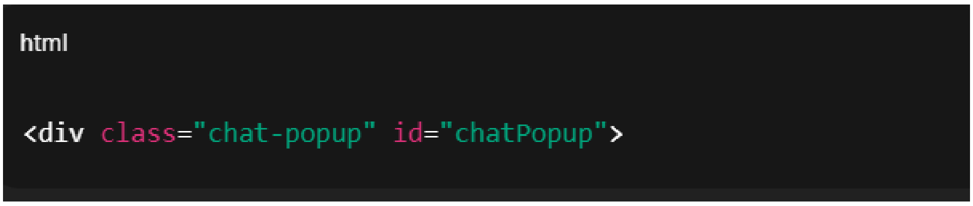


```html
<button class="chat-btn" onclick="toggleChat()">💬</button>
```

Displays a floating chat icon

- Opens and closes the chatbot popup

Chat Popup Window

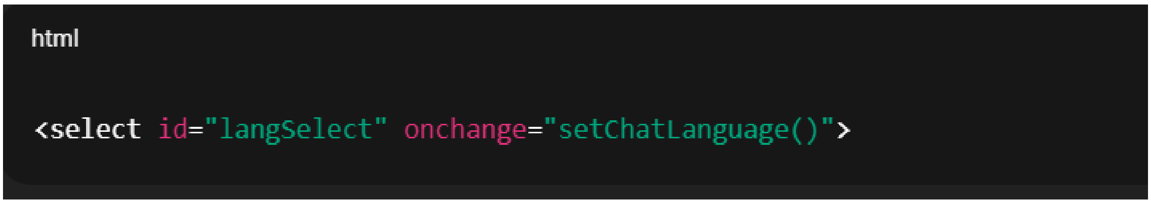


```html
<div class="chat-popup" id="chatPopup">
```

- Main chatbot container
- Initially hidden and shown using JavaScript

Header Section

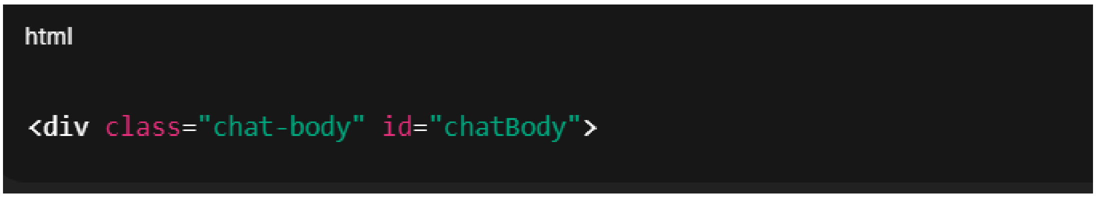
- Chat title
- Language selector (English / Punjabi)
- Close button



```html
<select id="langSelect" onchange="setChatLanguage()">
```

- Changes chatbot language
- Updates text and sends language info to backend

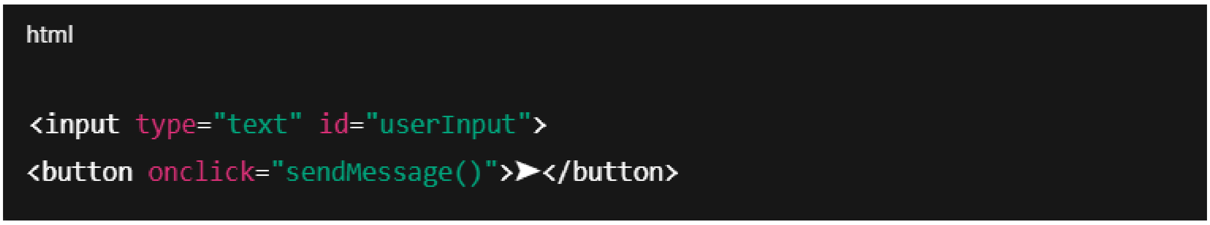
Chat Body



```html
<div class="chat-body" id="chatBody">
```

- Displays user messages and bot replies dynamically

Input Area



html

```html
<input type="text" id="userInput">
<button onclick="sendMessage()">➤</button>
```

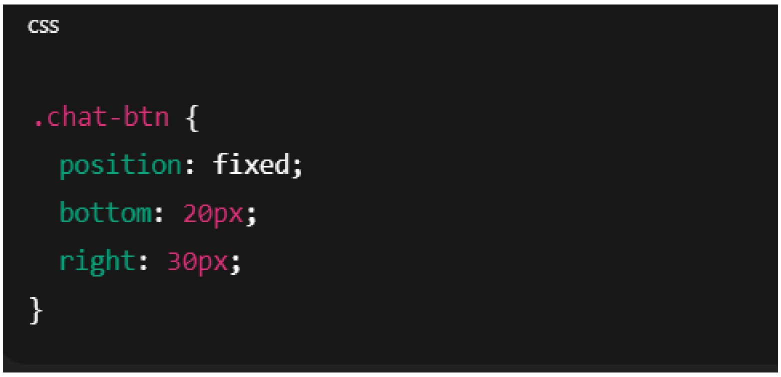
- User types message
- Button sends message to chatbot

2. style.css – UI Design & Styling

This file controls the **appearance** of the chatbot.

Important Styling Features

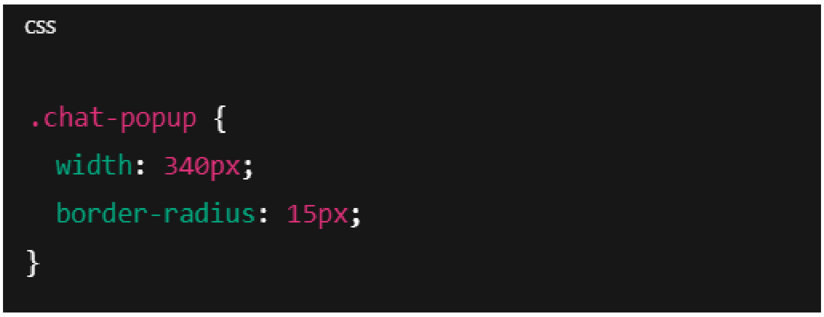
Floating Button Design



```css
.chat-btn {
  position: fixed;
  bottom: 20px;
  right: 30px;
}
```

- Keeps chatbot button fixed on screen

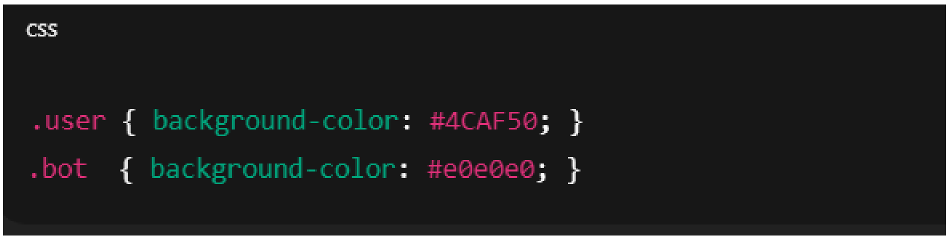
Chat Popup Styling



```css
.chat-popup {
  width: 340px;
  border-radius: 15px;
}
```

- Modern popup design
- Rounded corners and shadow

Message Bubbles



css

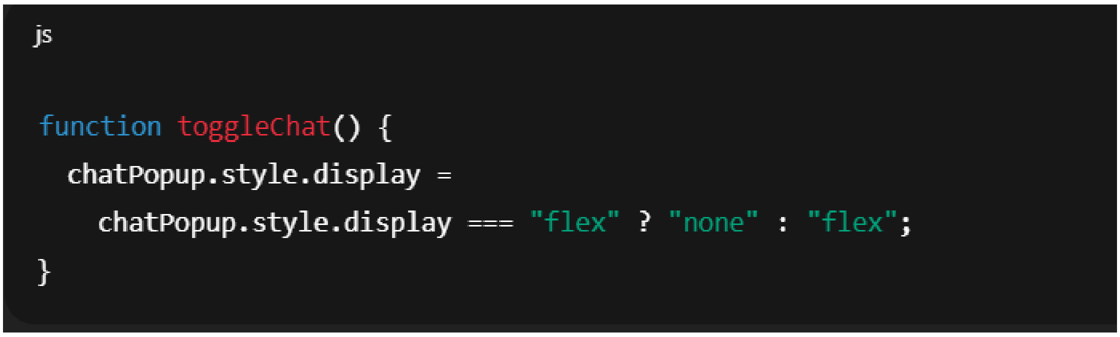
```css
.user { background-color: #4CAF50; }
.bot  { background-color: #e0e0e0; }
```

- User messages appear in green
- Bot messages appear in grey

3. script.js – Frontend Logic (Main File)

This file controls **how the chatbot behaves**.

A. Open / Close Chatbot

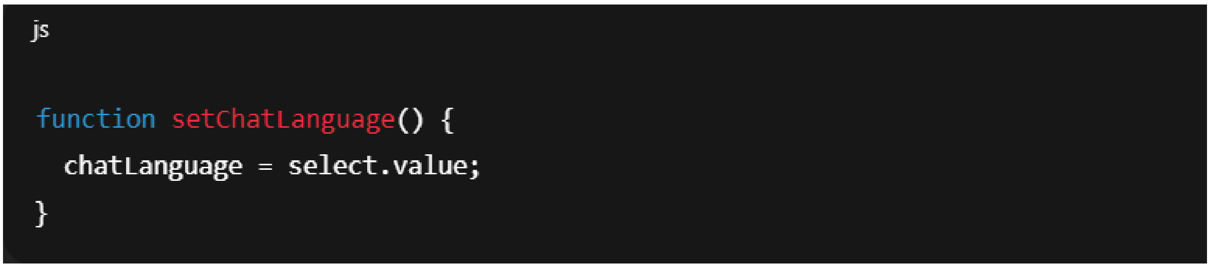


js

```js
function toggleChat() {
  chatPopup.style.display =
    chatPopup.style.display === "flex" ? "none" : "flex";
}
```

- Opens or closes the chatbot popup

B. Language Switching

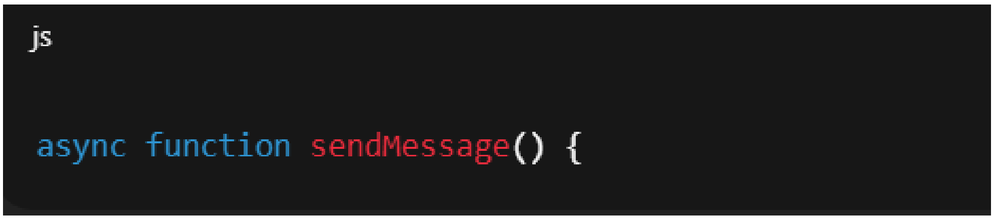


js

```js
function setChatLanguage() {
  chatLanguage = select.value;
}
```

- Changes:
    - Chat title
    - Input placeholder
    - Initial bot message
- Sends selected language to backend

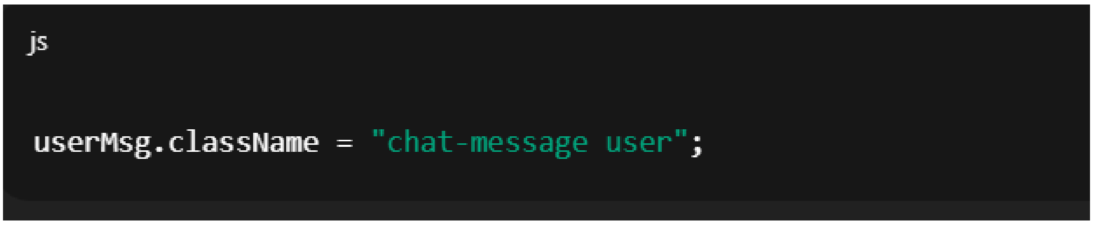
C. Sending a Message (Main Flow)



```js
async function sendMessage() {
```

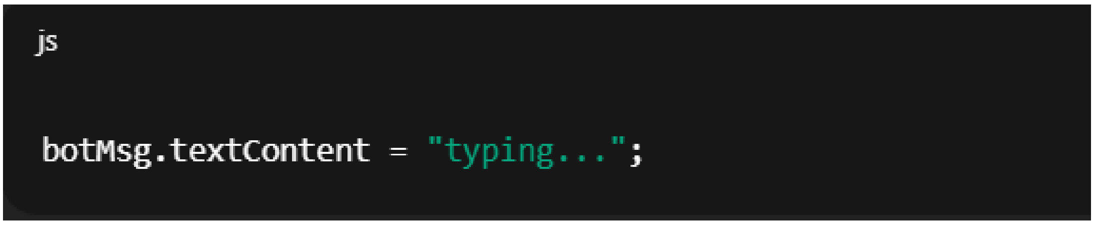
Step-by-step process:

1 User message is shown in chat



```js
userMsg.className = "chat-message user";
```

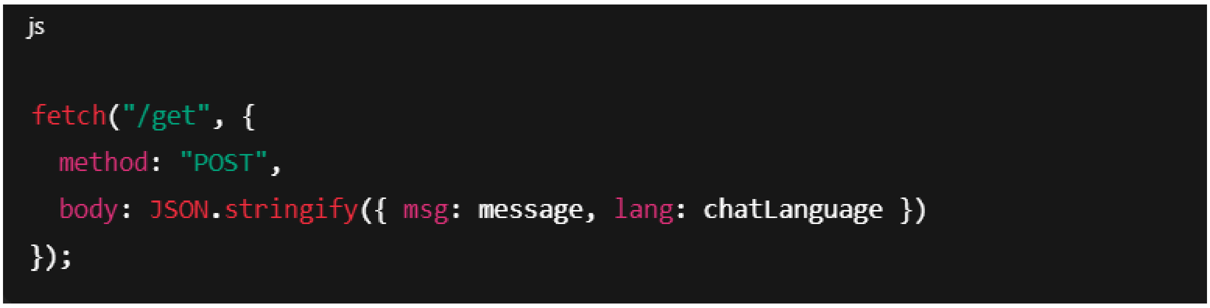
2 Typing animation is displayed



```js
botMsg.textContent = "typing...";
```

- Gives real-time chatbot feel

3 Message is sent to backend

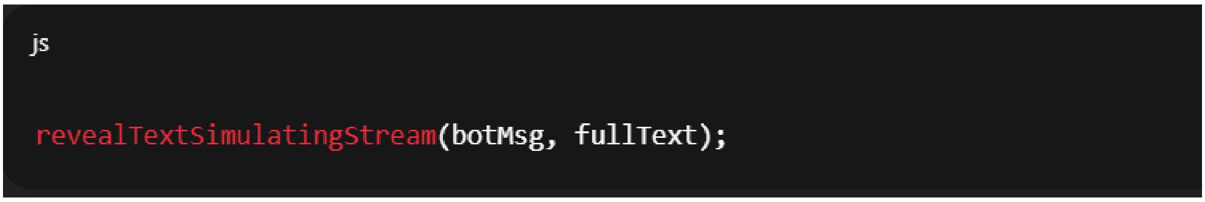


js

```js
fetch("/get", {
  method: "POST",
  body: JSON.stringify({ msg: message, lang: chatLanguage })
});
```

- Sends user message + selected language

4 Bot response is displayed with typing effect

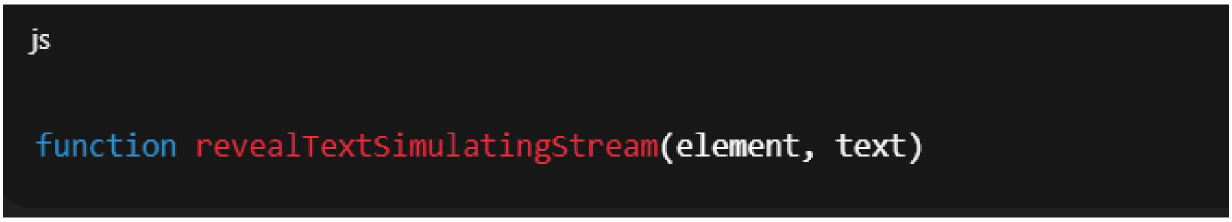


js

```js
revealTextSimulatingStream(botMsg, fullText);
```

- Text appears character-by-character

D. Typing / Streaming Effect



js

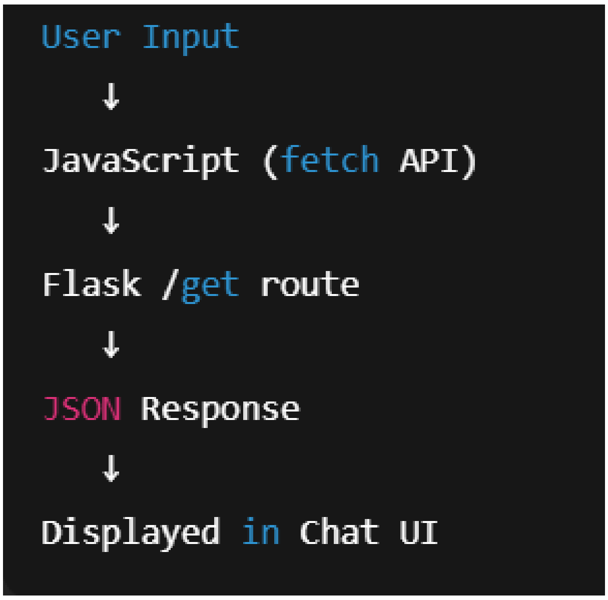
```js
function revealTextSimulatingStream(element, text)
```

- Displays bot response gradually
- Improves user experience

Frontend → Backend Communication

The frontend does **not** handle logic or data processing.

Flow:



```
User Input
   ↓
JavaScript (fetch API)
   ↓
Flask /get route
   ↓
JSON Response
   ↓
Displayed in Chat UI
```

FRONTEND SUMMARY

The frontend of the chatbot is implemented as a floating popup interface using HTML, CSS, and JavaScript. HTML defines the chatbot structure, CSS provides a responsive and modern design, and JavaScript manages user interactions, language switching, API communication, and real-time typing effects. The frontend communicates with the Flask backend through the Fetch API and dynamically displays chatbot responses in a user-friendly conversational format.