

DAY – 35, 36

16, 17 September 2025

To build a chatbot popup like the one in the image, and ensure it's portable across any application (web, mobile, laptop), you should create a responsive web-based widget using HTML, CSS, and JavaScript, which can be embedded into any website or webview within a mobile app.

Full Source Code (Responsive + Portable)

You can embed this in any website, React app, mobile webview, or hybrid app (like Flutter, React Native WebView).

□ index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Chatbot Popup</title>

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

<style>

body {

margin: 0;

font-family: Arial, sans-serif;
```

```
}
```

```
/* Chat button */
```

```
.chat-button {
```

```
    position: fixed;
```

```
    bottom: 20px;
```

```
    right: 20px;
```

```
    background: linear-gradient(to right, #2E81FC, #0B5FFF);
```

```
    border-radius: 50%;
```

```
    width: 60px;
```

```
    height: 60px;
```

```
    border: none;
```

```
    cursor: pointer;
```

```
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);
```

```
}
```

```
.chat-button::after {
```

```
    content: "□";
```

```
    font-size: 24px;
```

```
    color: white;
```

```
}
```

```
/* Chat popup */
```

```
.chat-popup {  
    display: none;  
    position: fixed;  
    bottom: 90px;  
    right: 20px;  
    width: 300px;  
    max-height: 500px;  
    background: white;  
    border-radius: 10px;  
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);  
    overflow: hidden;  
    z-index: 1000;  
    flex-direction: column;  
}  
  
-----  
}
```

```
.chat-header {  
    background: linear-gradient(to right, #2E81FC, #0B5FFF);  
    color: white;  
    padding: 15px;  
    font-weight: bold;  
    display: flex;  
    align-items: center;  
}
```

```
.chat-header img {  
    border-radius: 50%;  
    margin-right: 10px;  
    width: 32px;  
    height: 32px;  
}
```

```
.chat-body {  
    padding: 15px;  
    font-size: 14px;  
}
```

```
.chat-body .message {  
    margin-bottom: 10px;  
}
```

```
.chat-options button {  
    display: block;  
    width: 100%;  
    margin: 5px 0;  
    padding: 10px;  
    background-color: #f1f1f1;
```

```
border: none;  
border-radius: 5px;  
cursor: pointer;  
}  
  
.chat-options button:hover {  
background-color: #e0e0e0;  
}  
  
.chat-footer {  
padding: 10px;  
font-size: 12px;  
text-align: center;  
color: #777;  
}  
</style>  
</head>  
<body>  
  
<button class="chat-button" onclick="toggleChat()"></button>  
  
<div class="chat-popup" id="chatPopup">  
  <div class="chat-header">
```

```



Chat with us!

</div>

<div class="chat-body">

<div class="message">Hi there! I'm Tidus ☺ a bot working for our team. How can I
help you today?</div>

<div class="chat-options">

<button onclick="botReply('Just playing around')">Just playing around</button>

<button onclick="botReply('I need support')">I need Support</button>

</div>

</div>

<div class="chat-footer">Powered by YourCompany</div>

</div>

<script>

function toggleChat() {

const popup = document.getElementById("chatPopup");

popup.style.display = (popup.style.display === "flex") ? "none" : "flex";

}

function botReply(choice) {

alert(`You selected: "${choice}" . This would trigger a chatbot response.`);

}

```

```
</script>
```

```
</body>
```

```
</html>
```

How to Use It Anywhere

Platform Method

Web Embed HTML in your website or app.

Mobile Web Responsive design works natively.

React Wrap code inside a component using useState for visibility toggle.

Flutter Use WebView widget to embed this chatbot UI.

React Native Use WebView to load this chatbot code from a hosted source.

WordPress Add this code to a Custom HTML block or plugin.

Optional: Enhance with Chat API

Connect to backend using WebSockets, Firebase, or API.

Use NLP or LLM (like OpenAI GPT) to power conversations.

Add language support (e.g. Punjabi + English).

Use speech-to-text (Web Speech API) if needed.

Add Backend Support (e.g. Flask)

Here's a complete setup:

PART 1: Frontend + Server Integration

We'll now update the previous **HTML code** so that it sends messages to your **Flask backend server** via **AJAX (or fetch)**.

Updated index.html (with Live Server Chat Support)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Live Chatbot</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
  body { margin: 0; font-family: Arial, sans-serif; }

.chat-button {
  position: fixed;
  bottom: 20px;
  right: 20px;
  background: linear-gradient(to right, #2E81FC, #0B5FFF);
  border-radius: 50%;
  width: 60px;
```

```
height: 60px;  
border: none;  
cursor: pointer;  
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);  
}
```

```
.chat-button::after {  
content: "□";  
font-size: 24px;  
color: white;  
}
```

```
.chat-popup {  
display: none;  
position: fixed;  
bottom: 90px;  
right: 20px;  
width: 300px;  
max-height: 500px;  
background: white;  
border-radius: 10px;  
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);  
overflow: hidden;  
z-index: 1000;  
flex-direction: column;  
}
```

```
.chat-header {  
background: linear-gradient(to right, #2E81FC, #0B5FFF);  
color: white;  
padding: 15px;  
font-weight: bold;  
display: flex;
```

```
align-items: center;  
}  
  
.chat-header img {  
    border-radius: 50%;  
    margin-right: 10px;  
    width: 32px;  
    height: 32px;  
}  
  
.chat-body {  
    padding: 15px;  
    font-size: 14px;  
    height: 200px;  
    overflow-y: auto;  
}  
  
.chat-input {  
    display: flex;  
    border-top: 1px solid #eee;  
}  
  
.chat-input input {  
    flex: 1;  
    border: none;  
    padding: 10px;  
    font-size: 14px;  
    outline: none;  
}  
  
.chat-input button {  
    background: #2E81FC;  
    color: white;
```

```

border: none;
padding: 10px;
cursor: pointer;
}

.chat-message {
margin-bottom: 10px;
}

.chat-footer {
padding: 10px;
font-size: 12px;
text-align: center;
color: #777;
}

</style>
</head>
<body>

<button class="chat-button" onclick="toggleChat()"></button>

<div class="chat-popup" id="chatPopup">
<div class="chat-header">

Chat with us!
</div>
<div class="chat-body" id="chatBody">
<div class="chat-message"><strong>Bot:</strong> Hello! How can I assist you
today?</div>
</div>
<div class="chat-input">
<input type="text" id="userInput" placeholder="Type your message..." />
<button onclick="sendMessage()">Send</button>

```

```

</div>
<div class="chat-footer">Powered by YourServer</div>
</div>

<script>
function toggleChat() {
    const popup = document.getElementById("chatPopup");
    popup.style.display = (popup.style.display === "flex") ? "none" : "flex";
    popup.style.flexDirection = "column";
}

async function sendMessage() {
    const userInput = document.getElementById("userInput").value;
    if (!userInput) return;

    const chatBody = document.getElementById("chatBody");
    chatBody.innerHTML += `<div class="chat-message"><strong>You:</strong>
${userInput}</div>`;
    document.getElementById("userInput").value = "";

    const response = await fetch("http://127.0.0.1:5000/chat", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ message: userInput })
    });

    const data = await response.json();
    chatBody.innerHTML += `<div class="chat-message"><strong>Bot:</strong>
${data.reply}</div>`;
    chatBody.scrollTop = chatBody.scrollHeight;
}
</script>

```

```
</body>
</html>
```

PART 2: Backend Flask Server

```
# app.py (Flask server)
from flask import Flask, request, jsonify
from flask_cors import CORS

app = Flask(__name__)
CORS(app) # Allow cross-origin requests

@app.route('/chat', methods=['POST'])
def chat():
    user_message = request.json.get('message')

    # Simple bot logic (can integrate NLP/LLM here)
    if "help" in user_message.lower():
        bot_reply = "Sure! Let me assist you. What do you need help with?"
    elif "hello" in user_message.lower():
        bot_reply = "Hello there! How can I help you today?"
    else:
        bot_reply = "I'm just a simple bot. Ask me about support or help!"

    return jsonify({"reply": bot_reply})

if __name__ == '__main__':
    app.run(debug=True)
```

How to Run

1. **Save HTML code** as index.html and open it in any browser.
2. **Install Flask:**

3. pip install flask flask-cors

4. **Run Flask server:**

5. python app.py

Now You Have:

- A real **chatbot UI (popup style)**
- Live connection to your **server**
- Works on **web & mobile apps**