

### Step 1: Document Declaration

```
<!DOCTYPE html>
```

```
<html lang="en">
```

- Declares an **HTML5 document**.
  - `lang="en"` specifies the language as English for accessibility and SEO.
- 

### Step 2: Head Section

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Agri Chatbot</title>
```

```
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
```

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

```
</head>
```

1. `<meta charset="utf-8">` → supports all characters (e.g., emojis, regional languages).
  2. `<title>` → sets the browser tab title.
  3. `<link>` → links your **CSS file** for styling.
  4. `<script>` → loads **jQuery**, a library for easier DOM manipulation and AJAX.
- 

### Step 3: Body and Chat Container

```
<body class="bg">
```

```
<div class="chat-container">
```

- `<body>` contains all visible content.
  - `class="bg"` → background styling via CSS.
  - `<div class="chat-container">` → main wrapper for your chatbot.
- 

### Step 4: Header

```
<header class="chat-header">
```

```
<div><h2>🌿 Agri Chatbot</h2></div>
```

```
<div class="user-info">Welcome, <strong>{{ username }}</strong> — <a href="{{ url_for('logout') }}">Logout</a></div>
```

```
</header>
```

- Displays **chatbot title**.
  - Shows logged-in user's name ({{ username }} is Flask template variable).
  - Provides a **Logout link**.
- 

### Step 5: Chat History

```
<div id="chat-box" class="chat-box">
  {% for c in chats %}
    <div class="chat-row">
      <div class="user-msg">👤 {{ c.message }}</div>
      <div class="bot-msg">🤖 {{ c.response }}</div>
    </div>
  {% endfor %}
</div>
```

- <div id="chat-box"> → scrollable area for chat messages.
  - Loops through chats (Flask variable with previous messages).
  - Each message has:
    - User message → 👤 {{ c.message }}
    - Bot reply → 🤖 {{ c.response }}
- 

### Step 6: Chat Form

```
<form id="chat-form" class="chat-form">
  <select id="lang" name="lang">
    <option value="en">English</option>
    <option value="ta">தமிழ்</option>
    <option value="hi">हिन्दी</option>
    <option value="te">తెలుగు</option>
    <option value="ml">മലയാളം</option>
  </select>
  <input id="message" name="message" placeholder="Ask about crops, soil, pests..."
  autocomplete="off" required>
  <button type="submit">Send</button>
```

</form>

- Allows users to:
    - Choose **language** from dropdown.
    - Type a **message** (id="message").
    - Click **Send** to submit form.
  - autocomplete="off" → prevents browser suggestions in input.
- 

### Step 7: JavaScript / jQuery

```
$(function(){  
  const chatBox = $("#chat-box");  
  
  $("#chat-form").on("submit", function(e){  
    e.preventDefault();  
    const text = $("#message").val().trim();  
    const lang = $("#lang").val();  
    if(!text) return;  
  
    // Append user message  
    chatBox.append(`<div class="user-msg">👤 ${$('#<div>').text(text).html()}</div>`);  
    $("#message").val("");  
    chatBox.scrollTop(chatBox[0].scrollHeight);  
  
    // Send POST request to Flask  
    $.post("/chat", { message: text, lang: lang }, function(data){  
      chatBox.append(`<div class="bot-msg">🤖 ${$('#<div>').text(data.response).html()}</div>`);  
      chatBox.scrollTop(chatBox[0].scrollHeight);  
    }, "json").fail(function(){  
      chatBox.append(`<div class="bot-msg">🙏 Sorry, something went wrong.</div>`);  
      chatBox.scrollTop(chatBox[0].scrollHeight);  
    });  
  });  
});
```

});

### Step-by-step explanation:

1. `$(function(){ ... })` → waits until the page is fully loaded.
2. `const chatBox = $("#chat-box");` → stores the chat container.
3. `$("#chat-form").on("submit", function(e){ ... })` → intercepts form submission.
  - `e.preventDefault()` stops the page from reloading.
  - `const text = $("#message").val().trim();` → gets user input.
  - `const lang = $("#lang").val();` → gets selected language.
  - `if(!text) return;` → prevents empty messages.
4. Append **user message** to chat:
5. `chatBox.append('<div class="user-msg">👤 ${$("#message").val().trim().html()}</div>');`
  - Uses `text()` and `html()` to **escape HTML** (prevents injection).
6. Clear input field:
7. `$("#message").val("");`
8. Scroll chat to bottom:
9. `chatBox.scrollTop(chatBox[0].scrollHeight);`
10. Send **AJAX POST request** to Flask /chat route:
11. `$.post("/chat", { message: text, lang: lang }, function(data){`
12. `chatBox.append('<div class="bot-msg">🤖 ${data.response.html()}</div>');`
13. `chatBox.scrollTop(chatBox[0].scrollHeight);`
14. `}, "json")`
  - Server returns JSON: `{ response: "Bot reply" }`.
  - Appends bot reply to chat.
15. If AJAX fails → shows error message:
16. `.fail(function(){`
17. `chatBox.append('<div class="bot-msg">🤖 Sorry, something went wrong.</div>');`
18. `});`

---

### Step 8: Closing Tags

`</div>`

</body>

</html>

- Closes chat container, body, and HTML.

---

### Summary / Flow

1. User opens page → sees **chat history** and chat form.
2. User selects language, types a message, and clicks **Send**.
3. JavaScript intercepts the form → appends user message locally → sends AJAX POST to Flask.
4. Flask responds with bot reply → JavaScript appends reply to chat box.
5. Chat scrolls automatically to the bottom.
6. If server fails → shows **“Sorry, something went wrong”**.