we'll use the IBM Watson Java SDK to interact with Watson Assistant.

Here's a high-level overview of the steps:

1. Create an IBM Cloud account and provision an instance of Watson Assistant.
2. Obtain the credentials (API Key and URL) for your Watson Assistant instance.
3. Create a Java program to interact with Watson Assistant.
4. Develop a simple HTML webpage to provide a user interface for interacting with the chatbot.

Java Source Code:

```java
import com.ibm.cloud.sdk.core.security.IamAuthenticator;
import com.ibm.watson.assistant.v2.Assistant;
import com.ibm.watson.assistant.v2.model.*;

public class WatsonAssistantExample {

    public static void main(String[] args) {
        // Set up the Watson Assistant service with your API Key and URL
        String apiKey = "YOUR_API_KEY";
        String serviceUrl = "YOUR_SERVICE_URL";

        IamAuthenticator authenticator = new IamAuthenticator(apiKey);
        Assistant assistant = new Assistant("2020-04-01", authenticator);
```

```
assistant.setServiceUrl(serviceUrl);

// Define the assistant session
CreateSessionOptions sessionOptions = new CreateSessionOptions.Builder()
        .assistantId("YOUR_ASSISTANT_ID")
        .build();

SessionResponse session = assistant.createSession(sessionOptions).execute().getResult();
String sessionId = session.getSessionId();

// Interact with the assistant
MessageInput input = new MessageInput.Builder()
        .text("Hello, Watson!")
        .build();

MessageOptions messageOptions = new MessageOptions.Builder()
        .assistantId("YOUR_ASSISTANT_ID")
        .sessionId(sessionId)
        .input(input)
        .build();

MessageResponse response = assistant.message(messageOptions).execute().getResult();

// Print the assistant's response
System.out.println("Assistant: " + response.getOutput().getGeneric().get(0).getText());
```

```java
        // Close the session
        assistant.deleteSession(new DeleteSessionOptions.Builder()
                .assistantId("YOUR_ASSISTANT_ID")
                .sessionId(sessionId)
                .build()
        );
    }
}
```

HTML Webpage:

```html
<!DOCTYPE html>
<html>
<head>
    <title>IBM Watson Assistant</title>
</head>
<body>
    <h1>Chat with Watson Assistant</h1>
    <div id="chat"></div>
    <input type="text" id="userInput" placeholder="Type your message...">
    <button onclick="sendMessage()">Send</button>
    <script>
        function sendMessage() {
            var userInput = document.getElementById("userInput").value;
            document.getElementById("chat").innerHTML += "<p>You: " + userInput + "</p>";

            // Make a call to your Java program or server to send the user input to Watson Assistant

            // Replace this with your actual API request to Watson Assistant
            // You may use AJAX, fetch, or any other method to send the user's input and get a response
            // Display the response in the chat
            var response = "Watson: This is a sample response.";
            document.getElementById("chat").innerHTML += "<p>" + response + "</p>";

            document.getElementById("userInput").value = "";
        }
    </script>
</body>
</html>
```

This is a basic example, and you would need to replace the placeholder values with your actual Watson Assistant credentials and implement the API calls to Watson Assistant on your server. The HTML webpage provides a simple interface for users to interact with the chatbot.