```
import spacy

nlp = spacy.load("en_core_web_sm")

def analyze_text(text):
    doc = nlp(text)
    # Example: Extract entities
    for entity in doc.ents:
        print(entity.text, entity.label_)

Integrate Reasoning into Web Application

Flask app structure:

from flask import Flask, request, jsonify
app = Flask(__name__)

@app.route('/reason', methods=['POST'])
def reason():
    data = request.json
    text = data.get('text')

```

```
        # Process text and perform reasoning
        result = analyze_text(text)

        return jsonify(result=result)

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

5. Create the Frontend

Use HTML, CSS, and JavaScript to build a user interface.

For dynamic content, consider using frameworks like React or Vue.js.

Example HTML form:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```
        <title>Reasoning Website</title>
    </head>
    <body>
        <form id="reasonForm">
            <textarea id="inputText" placeholder="Enter text
    here..."></textarea>
            <button type="submit">Analyze</button>
        </form>

        <div id="result"></div>

        <script>
            document.getElementById('reasonForm').onsubmit = async
    function(event) {
                event.preventDefault();

                const text = document.getElementById('inputText').value;
                const response = await fetch('/reason', {
                    method: 'POST',
```

```
                    headers: {'Content-Type': 'application/json'},
                    body: JSON.stringify({text})
                });

                const result = await response.json();

                document.getElementById('result').innerText = result.result;
            };
        </script>
    </body>
    </html>
```