```
import requests
from bs4 import BeautifulSoup
from sentence_transformers import SentenceTransformer
import pinecone

# Initialize the embedding model
embedding_model = SentenceTransformer('all-MiniLM-L6-v2')

# Initialize the Pinecone vector database
pinecone.init(api_key="<sk-proj-tskVYFwmETi2Y5LBUzYOEUG3aNRMDk7mPbFGBkhDzshAAakA9od-
1Il5A7mohQV4S8Lfxe3hjeT3BlbkFJ4Iynv04e4ddXEwZ0kQL7qlXF-Qnh-
9gvats22RlSPU0zLmjB5wPIW_J2b9F7eElpMB0ebVAesA>", environment="us-west1-gcp")
index = pinecone.Index("rag-pipeline-index")

# Function to scrape website content
def scrape_website(url):
    response = requests.get(url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        # Extract all text content from the website
        text = ' '.join([element.get_text() for element in soup.find_all(['p', 'h1', 'h2', 'h3', 'li'])])
        return text
    else:
        print(f"Failed to fetch {url}: {response.status_code}")
        return None

# Function to chunk text into smaller segments
def chunk_text(text, max_chunk_size=300):
    words = text.split()
    chunks = []
    current_chunk = []

    for word in words:
        current_chunk.append(word)
        if len(current_chunk) >= max_chunk_size:
            chunks.append(" ".join(current_chunk))
            current_chunk = []

    if current_chunk:
        chunks.append(" ".join(current_chunk))

    return chunks

# Function to embed and store chunks in Pinecone
def store_chunks_in_pinecone(chunks, metadata):
    for i, chunk in enumerate(chunks):
        embedding = embedding_model.encode(chunk).tolist()
        metadata_with_id = metadata.copy()
        metadata_with_id['chunk_id'] = f"{metadata['id']}_chunk_{i}"
        index.upsert([(metadata_with_id['chunk_id'], embedding, metadata_with_id)])

# Main pipeline function for website scraping
def process_website(url):
    # Scrape website content
    text_data = scrape_website(url)
    if text_data:
        # Chunk text into smaller pieces
        chunks = chunk_text(text_data)

        # Metadata for the website
        metadata = {
            "id": url,
            "source": "website",
            "url": url
        }

        # Store chunks in Pinecone
        store_chunks_in_pinecone(chunks, metadata)

# Example usage
if __name__ == "__main__":
    websites = [
        "https://www.uchicago.edu/",
        "https://www.washington.edu/",
        "https://www.stanford.edu/",
        "https://und.edu/"
    ]

    for website in websites:
        process_website(website)

    # Example query processing
    query = "What is the mission of the University of Chicago?"
    query_embedding = embedding_model.encode(query).tolist()

    # Perform similarity search in Pinecone
    results = index.query(query_embedding, top_k=5, include_metadata=True)

    for match in results["matches"]:
        print(f"Source: {match['metadata']['url']}")
print(f"Chunk ID: {match['metadata']['chunk_id']}\nContent: {match['metadata']}\n")
```