```python
import pandas as pd

# Load CSV
df = pd.read_csv('techcrunch.csv')

# Preprocess content (if needed)
df['Content'] = df['Content'].fillna('')  # Handle missing content if any
combined_texts = df['Content'].tolist()
```

```python
print(combined_texts)
```

```
['Comment Back in 2019, Synex Medical founder Ben Nashman spent the night detained by US customs. Nashman tried to expla
```

```python
from sentence_transformers import SentenceTransformer

# Load pre-trained transformer model
model = SentenceTransformer('paraphrase-MiniLM-L6-v2')

# Create embeddings for each article's content
embeddings = model.encode(combined_texts)
```

```
/Users/mahamadoumoudjahidmahamadounouroudini/Documents/NYU/fall24/projects_in_programming/startupFundingTracker/.venv/li
  from tqdm.autonotebook import tqdm, trange
/Users/mahamadoumoudjahidmahamadounouroudini/Documents/NYU/fall24/projects_in_programming/startupFundingTracker/.venv/li
  warnings.warn(
```

```python
import faiss
import numpy as np

# Create a FAISS index
d = embeddings.shape[1]  # Embedding dimension
index = faiss.IndexFlatL2(d)  # L2 distance for similarity

# Convert embeddings to a NumPy array and add them to the index
embeddings_np = np.array(embeddings).astype('float32')
index.add(embeddings_np)

# Check the number of vectors in the index
print(f"Total articles indexed: {index.ntotal}")
```

```
Total articles indexed: 105
```

```python
def search_articles(query, top_n=5):
    # Embed the query
    query_embedding = model.encode([query])

    # Search the FAISS index
    distances, indices = index.search(np.array(query_embedding).astype('float32'), top_n)

    # Return the results
    results = []
    for idx in indices[0]:
        result = {
            'Title': df.iloc[idx]['Title'],
            'Author': df.iloc[idx]['Author'],
            'URL': df.iloc[idx]['URL'],
            'Content': df.iloc[idx]['Content']
        }
        results.append(result)

    return results

# Example search
# query = input("how can I help you today?: ")
# results = search_articles(query)

# for res in results:
#     print(f"Title: {res['Title']}\nAuthor: {res['Author']}\nURL: {res['URL']}\nContent: {res['Content'][:200]}...\n")
```

```python
from openai import OpenAI
client = OpenAI()
import os
from dotenv import load_dotenv

# Load environment variables from .env file
load_dotenv()
```

```
# Set OpenAI API key
client.api_key = os.getenv("OPENAI_API_KEY")
systemPrompt = "you are the best assistant ever and you answers are full with enthousiasm"

# Function to enhance search results using GPT
def enhance_with_gpt(articles):
    enhanced_results = []

    for article in articles:
        # Format content for GPT
        prompt = f"Please summarize the following article details in a more conversational tone:\n\n"
        prompt += f"Title: {article['Title']}\n"
        prompt += f"Author: {article['Author']}\n"
        prompt += f"Content Snippet: {article['Content'][:500]}\n\n"
        prompt += "Make the summary user-friendly and informative."

        # Use client.chat.completions
        response = client.chat.completions.create(
            model="gpt-4o-mini",
            messages=[
                {"role": "system", "content": systemPrompt},
                {"role": "user", "content": prompt}
            ],
            max_tokens=200,
            temperature=0.7
        )

        # Extract the response content

        enhanced_content = response.choices[0].message.content

        # Append enhanced content to the result
        enhanced_results.append({
            'Title': article['Title'],
            'Author': article['Author'],
            'URL': article['URL'],
            'Enhanced Content': enhanced_content
        })

    return enhanced_results


# Example enhanced search
query = input("how can i help you today")
faiss_results = search_articles(query)

# Enhance with GPT
enhanced_results = enhance_with_gpt(faiss_results)

# Display the results
for res in enhanced_results:
    print(f"Title: {res['Title']}\nAuthor: {res['Author']}\nURL: {res['URL']}\nEnhanced Summary: {res['Enhanced Content']}\n"
```

```
Title: Fal.ai, which hosts media-generating AI models, raises $23M from a16z and others
Author: Kyle Wiggers
URL: https://techcrunch.com/2024/09/18/fal-ai-which-hosts-media-generating-ai-models-raises-23m-from-a16z-and-others/
Enhanced Summary: Hey there! So, here's the scoop on Fal.ai, a super cool platform that focuses on creating AI-generated

The funding came in two parts: $14 million from a Series A round led by Kindred Ventures and an additional $9 million fr

Title: Virtuous, a fundraising CRM for nonprofits, raises $100M from Susquehanna Growth Equity
Author: Marina Temkin
URL: https://techcrunch.com/2024/09/19/virtuous-a-fundraising-crm-for-nonprofits-raises-100m-from-susquehanna-growth-equ
Enhanced Summary: Hey there! So, there's some exciting news in the nonprofit world! Virtuous, a customer relationship ma

The article, written by Marina Temkin, touches on a particularly relatable point. It mentions that many nonprofits, like

However, Gabe Cooper, the founder and CEO of Virtuous, points out a big issue: a lot of these organizations struggle wit

Title: Health insurance startup Alan reaches $4.5B valuation with new $193M funding round
Author: Romain Dillet
URL: https://techcrunch.com/2024/09/20/health-insurance-startup-alan-reaches-45-billion-valuation-with-new-funding-round
Enhanced Summary: Hey there! Exciting news in the world of health insurance! Alan, the fantastic French startup that's b

The big news is that they've teamed up with Belfius, one of Belgium's major banks. This partnership is pretty cool becau

It looks like Alan is on a thrilling ride toward growth and innovation in health insurance! 🚀

Title: Craig Newmark pledges $100M to fight hacking by foreign governments
Author: Connie Loizos
URL: https://techcrunch.com/2024/09/18/craig-newmark-pledges-100m-to-fight-hacking-by-foreign-governments/
Enhanced Summary: Hey there! So, here's some exciting news: Craig Newmark, the founder of Craigslist, is stepping up to

In an interview with The Wall Street Journal, Newmark shared that he plans to split the funds right down the middle. Hal
```

```
This isn’t Newmark’s first rodeo in philanthropy—since he started in 2015, he’s already donated or pledged over $400 mil

Title: AI notetaker Fathom raises $17M
Author: Frederic Lardinois
URL: https://techcrunch.com/2024/09/19/ai-notetaker-fathom-raises-17m/
Enhanced Summary: Hey there! Exciting news in the world of AI notetakers! Fathom, a company that's been around since 202
```

```
# Using pinecone
```

```
import os
from dotenv import load_dotenv
import openai
from sentence_transformers import SentenceTransformer
from pinecone import Pinecone, ServerlessSpec


# Load environment variables from .env file
load_dotenv()

# Initialize OpenAI API (assuming you need GPT for the conversational tone)
openai.api_key = os.getenv("OPENAI_API_KEY")

# Initialize Pinecone
pinecone_api_key = os.getenv('PINECONE_API_KEY')
pc = Pinecone(api_key=pinecone_api_key, environment="us-east-1")  # Change environment based on your region

# Create or connect to a Pinecone index
index_name = "article-index"
index = pc.Index(index_name)

# Initialize the model for generating embeddings
model = SentenceTransformer('all-MiniLM-L6-v2')  # You can use other models as well

# Function to add articles to Pinecone
def add_articles_to_pinecone(articles):
    for idx, article in enumerate(articles):
        # Create the embedding for the article content
        embedding = model.encode(article['Content'])

        # Upload the vector to Pinecone
        index.upsert(vectors=[(str(idx), embedding)])

# Function to search articles using Pinecone
def search_articles(query):
    # Create the query vector
    query_embedding = model.encode(query)

    # Query Pinecone for similar articles
    result = index.query(queries=[query_embedding], top_k=5)

    # Retrieve matched article IDs
    matched_articles = result['matches']

    return matched_articles
# Example workflow



# Add articles to Pinecone
add_articles_to_pinecone(articles)

# Example search query
query = "non-invasive glucose testing"

# Search in Pinecone
pinecone_results = search_articles(query)


# Print the enhanced results
for res in enhanced_results:
    print(f"Title: {res['Title']}\nAuthor: {res['Author']}\nURL: {res['URL']}\nEnhanced Content: {res['Enhanced Content']}\r
```