



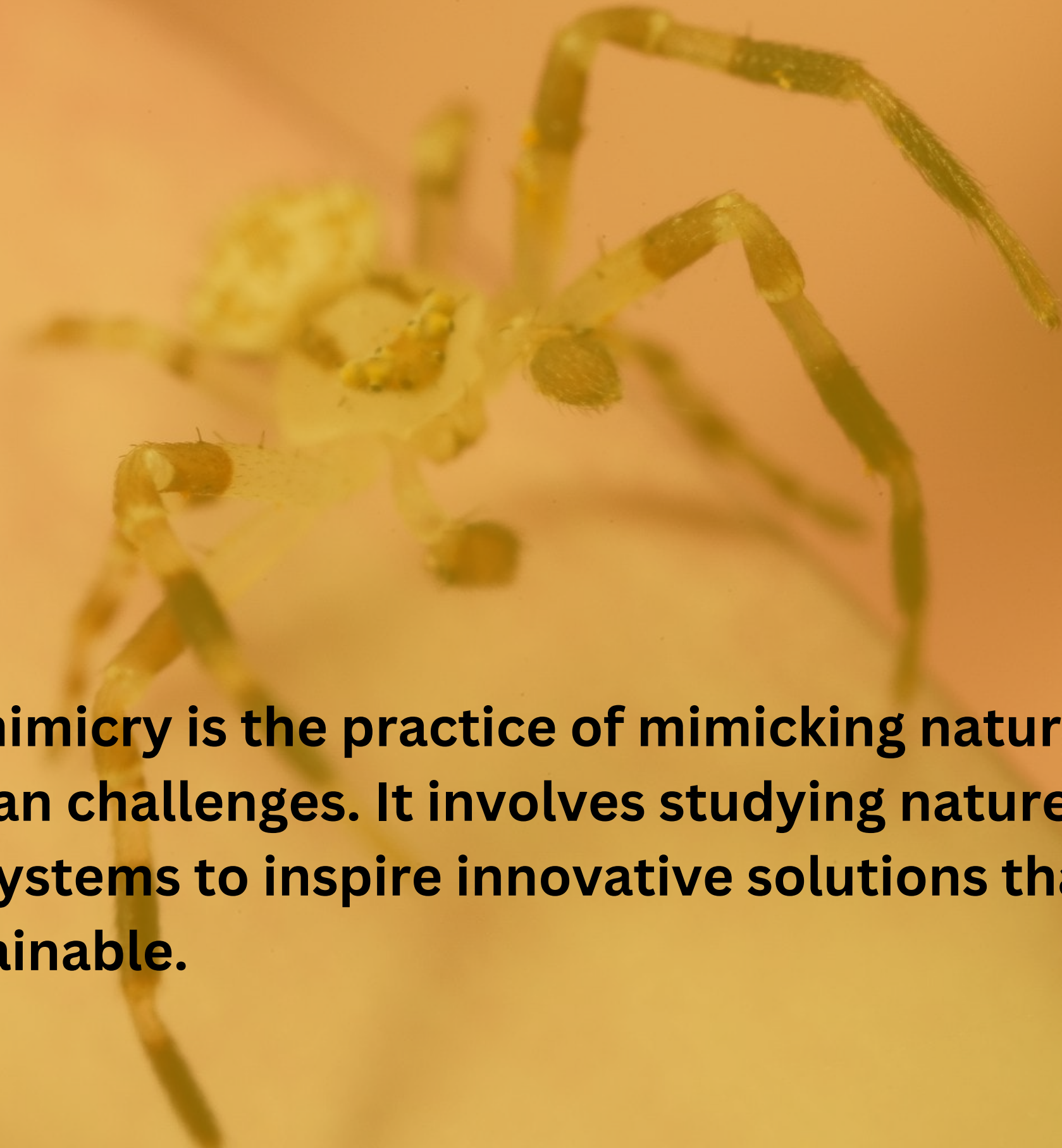
# **ConscienceOS**

**Harnessing Nature's Wisdom**

**Utilizing Biomimicry for Sustainable Solutions**

# **Nature is the most ancient design teacher**

**Biomimicry is the practice of mimicking nature's time-tested strategies to solve human challenges. It involves studying nature's forms, processes, and ecosystems to inspire innovative solutions that are both efficient and sustainable.**



# AskNature.org by Biomimicry Institute provides:

Biological Strategies

1700+

Biological Innovations

600+

The screenshot displays the AskNature.org website interface. At the top, there are navigation tabs for "BIOLOGICAL STRATEGIES 1764", "INNOVATIONS 306", and "RESOURCES 75". Below the tabs, a sidebar menu lists various categories under "Functions" and "Living Systems". The main content area features a grid of biological strategy cards, each with a title and a small image.

Category	Count
Break Down	85
Get, Store, or Distribute Resources	419
Maintain Community	213
Make	137
Modify	413
Move or Stay Put	306
Process Information	285
Protect From Physical Harm	813
Provide ecosystem services	1
Animals	1184
Archaeans	7
Bacteria	103
Ecosystems	35
Fungi	51
Plants	430
Protists	23

**Biological Strategies**

We've curated m innovation. Wha and mentor.

- Young Leaves Channel Water**  
Fish-pole bamboo
- Vessels Resist Bubble Formation**
- Foods Reduce or Enhance Fertility**  
Woolly spider monkey
- Chemical Signals Help the Living Avoid the Dead**  
Woodlice

# ConscienceOS

Making Biomimicry Accessible to all  
designers with the power of AI and  
Semantic Search

**User:**

**AI:**



**How do I ....?**

**How does nature?**

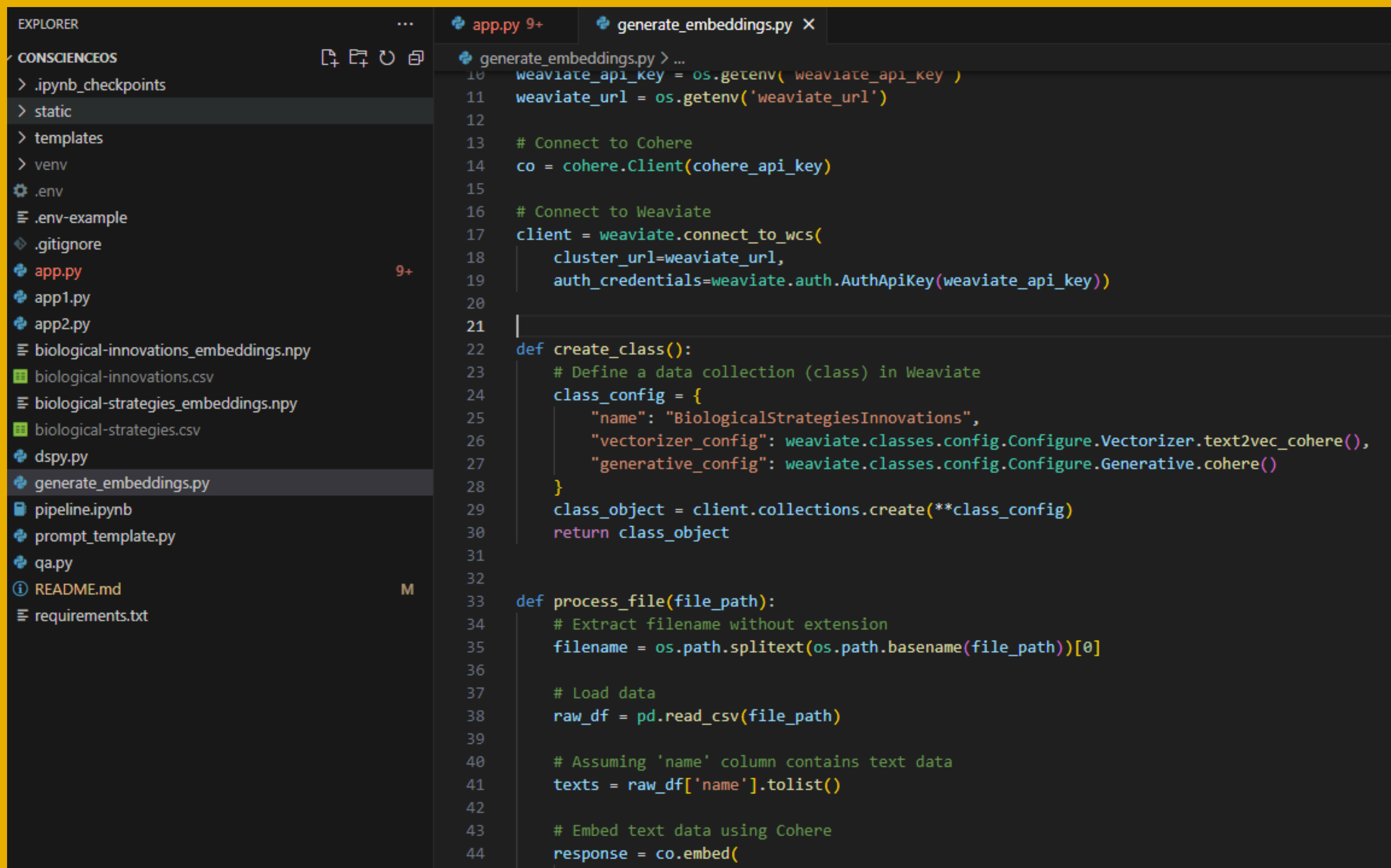
# Hybrid Search



**Weaviate**



# Step 1: Generate Embeddings



```
EXPLORER
CONSCIENCEOS
  > .ipynb_checkpoints
  > static
  > templates
  > venv
  .env
  .env-example
  .gitignore
  app.py
  app1.py
  app2.py
  biological-innovations_embeddings.npy
  biological-innovations.csv
  biological-strategies_embeddings.npy
  biological-strategies.csv
  dsp.py
  generate_embeddings.py
  pipeline.ipynb
  prompt_template.py
  qa.py
  README.md
  requirements.txt

generate_embeddings.py 9+
generate_embeddings.py > ...
10 weaviate_api_key = os.getenv( weaviate_api_key )
11 weaviate_url = os.getenv('weaviate_url')
12
13 # Connect to Cohere
14 co = cohere.Client(cohere_api_key)
15
16 # Connect to Weaviate
17 client = weaviate.connect_to_wcs(
18     cluster_url=weaviate_url,
19     auth_credentials=weaviate.auth.AuthApiKey(weaviate_api_key))
20
21
22 def create_class():
23     # Define a data collection (class) in Weaviate
24     class_config = {
25         "name": "BiologicalStrategiesInnovations",
26         "vectorizer_config": weaviate.classes.config.Configure.Vectorizer.text2vec_cohere(),
27         "generative_config": weaviate.classes.config.Configure.Generative.cohere()
28     }
29     class_object = client.collections.create(**class_config)
30     return class_object
31
32
33 def process_file(file_path):
34     # Extract filename without extension
35     filename = os.path.splitext(os.path.basename(file_path))[0]
36
37     # Load data
38     raw_df = pd.read_csv(file_path)
39
40     # Assuming 'name' column contains text data
41     texts = raw_df['name'].tolist()
42
43     # Embed text data using Cohere
44     response = co.embed(
```