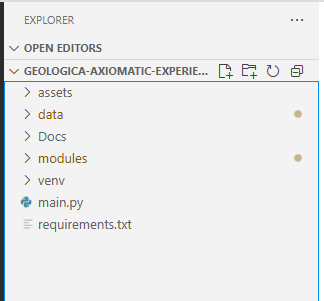
Directory Structure (VS Code Project: GeoLogica-Axiomatic-Experience)



Under assets

styles.css

body {

    font-family: 'Segoe UI', sans-serif;

}

h1 {

    color: #2E8B57;

}

data

definitions.json

{

  "postulates": [

    "A straight line segment can be drawn joining any two points.",

    "A straight line segment can be extended indefinitely in a straight line.",

    "Given any straight line segment, a circle can be drawn having the segment as radius and one endpoint as center.",

    "All right angles are equal to one another.",

    "If a straight line falling on two straight lines makes the interior angles on the same side less than two right angles, then the two lines, if extended indefinitely, meet on that side."

  ]

}

modules

axioms.py

import streamlit as st

import json

def display\_axioms():

    st.header("Euclid's Axioms")

    axioms = [

        "Things which are equal to the same thing are equal to one another.",

        "If equals are added to equals, the wholes are equal.",

        "If equals are subtracted from equals, the remainders are equal.",

        "Things which coincide with one another are equal to one another.",

        "The whole is greater than the part."

    ]

    for i, axiom in enumerate(axioms, 1):

        st.markdown(f"\*\*Axiom {i}:\*\* {axiom}")

logger.py

import pandas as pd

from datetime import datetime

LOG\_FILE = "data/proof\_log.csv"

def log\_attempt(user\_text, matched\_axioms):

    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    # Create a new entry as a DataFrame

    entry = pd.DataFrame([{

        "timestamp": timestamp,

        "proof": user\_text,

        "matched\_axioms": ", ".join(matched\_axioms)

    }])

    try:

        # Read existing log file

        df = pd.read\_csv(LOG\_FILE)

        # Concatenate new entry

        df = pd.concat([df, entry], ignore\_index=True)

    except FileNotFoundError:

        # If log file doesn't exist, start with the new entry

        df = entry

    # Save updated log

    df.to\_csv(LOG\_FILE, index=False)

proof\_validator.py

import spacy

# from modules import proof\_validator

# with st.spinner("Analyzing your proof using Euclid's logic..."):

#     proof\_text = st.text\_area("Enter your geometric proof:")

#     feedback = proof\_validator.validate\_proof(proof\_text)

#     st.markdown(feedback)

nlp = spacy.load("en\_core\_web\_sm")

AXIOM\_PATTERNS = {

    "Axiom 1": ["equal", "same thing"],

    "Axiom 2": ["add", "equals", "whole"],

    "Axiom 3": ["subtract", "equals", "remainder"],

    "Axiom 4": ["coincide", "equal"],

    "Axiom 5": ["whole", "greater", "part"]

}

def validate\_proof(proof\_text):

    if not proof\_text.strip():

        return "❌ No proof submitted. Please write your reasoning."

    doc = nlp(proof\_text.lower())

    matched\_axioms = []

    for axiom, keywords in AXIOM\_PATTERNS.items():

        if any(token.text in keywords for token in doc):

            matched\_axioms.append(axiom)

    if matched\_axioms:

        feedback = "✅ Your proof references the following axioms:\n"

        for axiom in matched\_axioms:

            feedback += f"- {axiom}\n"

        feedback += "\n🧠 Great! Try refining your logic or adding diagrams."

    else:

        feedback = "⚠️ No recognizable axioms found. Try rephrasing or reviewing Euclid's principles."

    return feedback

theorems.py

import streamlit as st

from difflib import SequenceMatcher

from modules.logger import log\_attempt

# Euclid's axioms

EUCLID\_AXIOMS = [

    "A straight line segment can be drawn joining any two points",

    "Any straight line segment can be extended indefinitely",

    "A circle can be drawn with any center and radius",

    "All right angles are equal",

    "If a line intersects two lines such that the sum of interior angles on the same side is less than two right angles, the lines meet on that side"

]

# Fuzzy matching threshold

FUZZY\_THRESHOLD = 0.6

def match\_axioms(user\_text):

    matched = []

    for axiom in EUCLID\_AXIOMS:

        ratio = SequenceMatcher(None, axiom.lower(), user\_text.lower()).ratio()

        if ratio >= FUZZY\_THRESHOLD:

            matched.append((axiom, round(ratio, 2)))

    return matched

def prove\_theorem():

    st.title("🔍 Euclidean Theorem Checker")

    st.markdown("Try proving this theorem using Euclid's axioms:")

    st.markdown("> \*\*Theorem:\*\* Two distinct lines cannot have more than one point in common.")

    user\_text = st.text\_area("Enter your proof attempt:", height=200)

    if st.button("Check Proof"):

        matched = match\_axioms(user\_text)

        log\_attempt(user\_text, [m[0] for m in matched])

        if matched:

            st.success(f"✅ Matched {len(matched)} axiom(s) with fuzzy logic:")

            for axiom, score in matched:

                st.markdown(f"- \*\*{axiom}\*\* \_(match score: {score})\_")

        else:

            st.warning("⚠️ No recognizable axioms found. Try rephrasing or reviewing Euclid's principles.")

            with st.expander("📜 View Euclid's Axioms"):

                for axiom in EUCLID\_AXIOMS:

                    st.markdown(f"- {axiom}")

            st.info("💡 Tip: Use geometric terms like 'line', 'circle', 'angle', and refer to known postulates.")

visuals.py

import streamlit as st

import plotly.graph\_objects as go

def show\_visuals():

    st.header("Visual Playground")

    st.markdown("Drag points and observe geometric relationships.")

    fig = go.Figure()

    fig.add\_trace(go.Scatter(x=[1, 4], y=[2, 6], mode='lines+markers', name='Line AB'))

    fig.update\_layout(width=600, height=400, title="Line through Two Points")

    st.plotly\_chart(fig)

inside Project Root

maian.py

import streamlit as st

from modules import axioms, theorems, visuals

st.set\_page\_config(page\_title="Euclid Unfolded", layout="wide")

st.title("📐 Euclid Unfolded: Geometry Reimagined")

st.markdown("Explore axioms, prove theorems, and visualize classical geometry interactively.")

tab1, tab2, tab3 = st.tabs(["Axiom Explorer", "Theorem Prover", "Visual Playground"])

with tab1:

    axioms.display\_axioms()

with tab2:

    theorems.prove\_theorem()

with tab3:

    visuals.show\_visuals()

requirements.py

spacy

streamlit

plotly

pandas

en\_core\_web\_sm @ https://github.com/explosion/spacy-models/releases/download/en\_core\_web\_sm-3.7.1/en\_core\_web\_sm-3.7.1.tar.gz