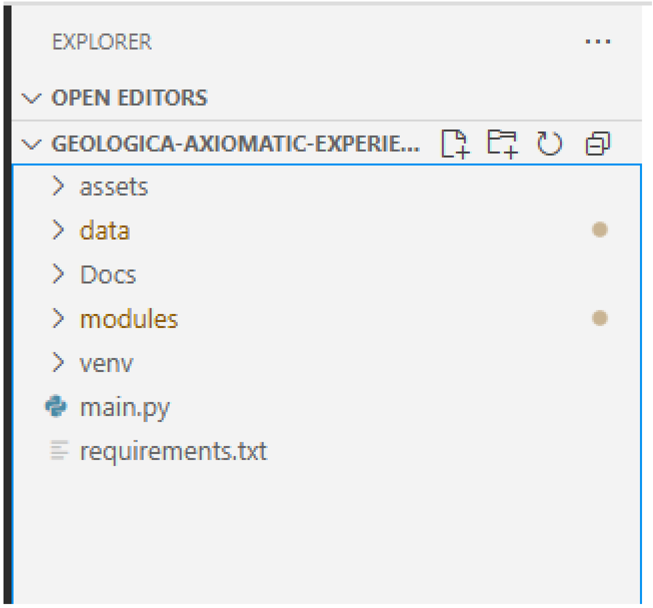
Directory Structure (VS Code Project: GeoLogica-Axiomatic-Experience)



```
> assets
> data
> Docs
> modules
> venv
main.py
requirements.txt
```

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
```