```
import requests
from rdflib import Graph, Namespace
from flask import Flask, request, jsonify

def fetch_user_profile(webid_url: str) -> dict:
    """Fetch and parse public RDF data from the user's WebID profile"""
    headers = {"Accept": "text/turtle"}
    response = requests.get(webid_url, headers=headers)

    if response.status_code != 200:
        raise Exception("Failed to fetch WebID profile")

    g = Graph()
    g.parse(data=response.text, format="turtle")

    FOAF = Namespace("http://xmlns.com/foaf/0.1/")

    # Try to find a subject with a foaf:name and foaf:interest
    subject = None
    for s in g.subjects(predicate=FOAF.name):
        subject = s
        break

    name = g.value(subject=subject, predicate=FOAF.name)
    interests = list(g.objects(subject=subject, predicate=FOAF.interest))

    return {
        "name": str(name) if name else "User",
        "interests": [str(i) for i in interests]
    }

def generate_recommendations(user_data: dict) -> str:
    interests = user_data.get("interests", [])

    if any("tech" in interest.lower() for interest in interests):
        return "Explore our new gadgets, AI tools, and smart devices!"
    elif any("fashion" in interest.lower() for interest in interests):
        return "Check out our new fashion line curated just for you!"
    else:
        return "Discover personalized offers and products tailored for your lifestyle!"

app = Flask(__name__)

@app.route('/personalize', methods=['GET'])
def personalize():
    webid = request.args.get('webid')
    if not webid:
        return jsonify({"error": "Missing WebID URL"}), 400

    try:
        user_data = fetch_user_profile(webid)
        recommendation = generate_recommendations(user_data)
        return jsonify({
```

```
            "user": user_data["name"],
            "recommendation": recommendation
        })
    except Exception as e:
        return jsonify({"error": str(e)}), 500

if __name__ == "__main__":
    app.run(debug=True)
```