*import* streamlit *as* st

*import* pandas *as* pd

*from* geopy.geocoders *import* Nominatim

*import* folium

*from* streamlit\_folium *import* folium\_static

*import* ssl

*import* certifi

*import* geopy.geocoders

*import* urllib.request

*# ---------- Styling & Page Config ----------*

COLORS = {

'baby\_blue': '#bfd7ed',

'blue\_grotto': '#60a3d9',

'royal\_blue': '#0074b7',

'navy\_blue': '#003b73',

'white': '#ffffff',

'error\_red': '#ff4b4b',

'success\_green': '#00c853',

'warning\_yellow': '#ffd700'

}

st.set\_page\_config(

*page\_title*="Build Feasibility Analyzer",

*page\_icon*="🏗️",

*layout*="wide"

)

*# Custom CSS with updated colors for better visibility*

st.markdown("""

<style>

.stApp {

background-color: #ffffff;

}

.st-emotion-cache-18ni7ap {

background-color: #bfd7ed;

}

.st-emotion-cache-16idsys p {

color: #000000 !important;

font-weight: 500;

}

.st-emotion-cache-10trblm {

color: #000000 !important;

font-weight: 600;

}

.st-emotion-cache-1gulkj5 {

background-color: #ffffff;

}

div[data-testid="stMetricValue"] {

color: #0074b7;

}

.st-emotion-cache-16idsys {

color: #000000;

}

.st-emotion-cache-1wbqy5l {

color: #000000;

}

.st-emotion-cache-1629p8f {

color: #000000;

}

.stButton > button {

background-color: #0074b7;

color: white;

}

.stButton > button:hover {

background-color: #003b73;

color: white;

}

div.stAlert > div {

color: #000000;

font-weight: 500;

}

h1 {

color: #003b73 !important;

font-weight: 800 !important;

background-color: #bfd7ed;

padding: 20px;

border-radius: 10px;

margin-bottom: 30px !important;

}

h2, h3, h4 {

color: #0074b7 !important;

font-weight: 600 !important;

}

</style>

""", *unsafe\_allow\_html*=True)

*# ---------- Session State ----------*

*if* 'proposed\_sites' not in st.session\_state:

st.session\_state.proposed\_sites = []

*# ---------- Utility Functions ----------*

def geocode\_address(*address*):

ctx = ssl.create\_default\_context(*cafile*=certifi.where())

geopy.geocoders.options.default\_ssl\_context = ctx

geopy.geocoders.options.default\_user\_agent = "my\_feasibility\_analyzer\_v2"

geolocator = Nominatim(

*user\_agent*="my\_feasibility\_analyzer\_v2",

*scheme*='http' *# Use HTTP instead of HTTPS*

)

*try*:

*# First try with city and state*

location = geolocator.geocode(

f"{address}, San Jose, California, USA",

*timeout*=10,

*exactly\_one*=True

)

*if* not location:

*# Try without state*

location = geolocator.geocode(

f"{address}, San Jose, USA",

*timeout*=10,

*exactly\_one*=True

)

*if* not location:

*# Try just the address*

location = geolocator.geocode(

address,

*timeout*=10,

*exactly\_one*=True

)

*if* location:

*return* {

'lat': location.latitude,

'lon': location.longitude,

'address': location.address

}

*else*:

st.error("📍 Could not find this address. Please try another one.")

*return* None

*except* Exception *as* e:

st.error(f"🚫 Error: {str(e)}")

*return* None

def evaluate\_feasibility(*lat*, *lon*):

flood\_risk = "High" *if* lon < -121.91 *else* "Low"

soil\_stability = "Unstable" *if* lat < 37.32 *else* "Stable"

slope = "Steep" *if* lat > 37.35 *else* "Moderate"

base\_cost\_sqft = 250

site\_prep\_multiplier = 1.4 *if* slope == "Steep" *else* 1.2 *if* slope == "Moderate" *else* 1.0

est\_cost = round(base\_cost\_sqft \* site\_prep\_multiplier, 2)

score = 1.0

*if* flood\_risk == "High":

score -= 0.3

*if* soil\_stability == "Unstable":

score -= 0.3

*if* slope == "Steep":

score -= 0.2

*elif* slope == "Moderate":

score -= 0.1

*return* {

"Flood Risk": flood\_risk,

"Soil Stability": soil\_stability,

"Terrain Slope": slope,

"Estimated Cost ($/sqft)": est\_cost,

"Feasibility Score": round(max(score, 0), 2)

}

def create\_map(*proposed\_sites*):

m = folium.Map(

*location*=[37.3382, -121.8863], *# San Jose center*

*zoom\_start*=12,

*tiles*='cartodbpositron',

*control\_scale*=True

)

*# Add a title to the map*

title\_html = '''

<div style="position: fixed;

top: 10px; left: 50px; width: 300px; height: 30px;

z-index:9999; font-size:16px; font-weight: bold;

background-color: white; border-radius: 5px;

padding: 5px;

box-shadow: 0 0 5px rgba(0,0,0,0.2);">

📍 Proposed Sites in San Jose

</div>

'''

m.get\_root().html.add\_child(folium.Element(title\_html))

*for* i, site *in* enumerate(proposed\_sites, 1):

*# Add marker for each site*

popup\_html = f"""

<div style='width: 200px'>

<h4>Site {i}</h4>

<b>Address:</b> {site.get('address', 'N/A')}<br>

<b>Coordinates:</b> ({site['lat']:.4f}, {site['lon']:.4f})<br>

</div>

"""

*# Main marker*

folium.CircleMarker(

*location*=[site['lat'], site['lon']],

*radius*=8,

*color*=COLORS['royal\_blue'],

*fill*=True,

*popup*=folium.Popup(popup\_html, *max\_width*=300),

*fill\_opacity*=0.7,

*weight*=2

).add\_to(m)

*# Range circle (1 mile radius)*

folium.Circle(

*location*=[site['lat'], site['lon']],

*radius*=1609, *# 1 mile in meters*

*color*=COLORS['baby\_blue'],

*fill*=True,

*popup*=f"1 mile radius around Site {i}",

*opacity*=0.2,

*fill\_opacity*=0.1

).add\_to(m)

*return* m

*# ---------- Main UI ----------*

st.markdown("<h1 style='text-align: center;'>🏗️ Build Feasibility Analyzer</h1>", *unsafe\_allow\_html*=True)

st.markdown("""

<div style='background-color: #f0f2f6; padding: 15px; border-radius: 5px; margin-bottom: 25px;'>

<h4 style='color: #003b73 !important; margin-top: 0;'>About this tool</h4>

<p style='color: #000000; font-size: 16px;'>

Evaluate whether proposed locations are feasible for constructing Emergency Interim Housing (EIH) units

based on geolocation analysis of flood risk, soil stability, and terrain slope.

</p>

</div>

""", *unsafe\_allow\_html*=True)

*# ---------- Sidebar ----------*

st.sidebar.markdown("""

<h2 style='color: #003b73; margin-bottom: 20px;'>🏗️ Add Proposed Site</h2>

""", *unsafe\_allow\_html*=True)

address = st.sidebar.text\_input(

"Enter site address (San Jose, CA):",

*placeholder*="e.g., 2011 Naglee Ave",

*help*="Enter a complete address in San Jose, CA"

)

col\_add, col\_clear = st.sidebar.columns(2)

*with* col\_add:

*if* st.button("➕ Add Site", *help*="Click to add the entered address"):

*if* address:

*with* st.spinner('Finding location...'):

coords = geocode\_address(address)

*if* coords:

st.session\_state.proposed\_sites.append(coords)

st.success(f"✅ Added site: {coords['address']}")

st.balloons()

*else*:

st.warning("Please enter an address first.")

*with* col\_clear:

*if* st.button("🧹 Clear All", *help*="Click to remove all sites"):

st.session\_state.proposed\_sites = []

st.success("All sites cleared.")

st.sidebar.markdown("---")

st.sidebar.markdown("""

<h4 style='color: #003b73;'>Example addresses to try:</h4>

<ul style='color: #000000;'>

<li>1661 Alum Rock Ave, San Jose</li>

<li>2011 Naglee Ave, San Jose</li>

<li>1 Washington Square, San Jose (SJSU Campus)</li>

</ul>

""", *unsafe\_allow\_html*=True)

*# ---------- Map Section ----------*

st.markdown("<h3 style='color: #003b73; margin-top: 30px;'>🗺️ Proposed Site Map</h3>", *unsafe\_allow\_html*=True)

*if* st.session\_state.proposed\_sites:

map\_obj = create\_map(st.session\_state.proposed\_sites)

folium\_static(map\_obj, *width*=1200)

*else*:

st.info("No sites added yet. Use the sidebar to enter an address.")

*# ---------- Feasibility Table ----------*

st.markdown("---")

st.markdown("<h3 style='color: #003b73; margin-top: 30px;'>📊 Feasibility Results</h3>", *unsafe\_allow\_html*=True)

*if* st.session\_state.proposed\_sites:

results = []

*for* i, site *in* enumerate(st.session\_state.proposed\_sites, 1):

row = evaluate\_feasibility(site['lat'], site['lon'])

row["Site #"] = f"Site {i}"

row["Address"] = site.get('address', 'N/A')

row["Latitude"] = round(site['lat'], 5)

row["Longitude"] = round(site['lon'], 5)

results.append(row)

df = pd.DataFrame(results)

*# Reorder columns*

columns\_order = [

"Site #", "Address", "Latitude", "Longitude",

"Flood Risk", "Soil Stability", "Terrain Slope",

"Estimated Cost ($/sqft)", "Feasibility Score"

]

df = df[columns\_order]

def color\_score(*val*):

*if* isinstance(val, (int, float)):

*if* val >= 0.8:

*return* f'background-color: {COLORS["success\_green"]}; color: white'

*elif* val >= 0.6:

*return* f'background-color: {COLORS["warning\_yellow"]}; color: black'

*else*:

*return* f'background-color: {COLORS["error\_red"]}; color: white'

*return* ''

*# Apply styling to the dataframe*

styled\_df = df.style\

.applymap(color\_score, *subset*=["Feasibility Score"])\

.format({

"Latitude": "{:.5f}",

"Longitude": "{:.5f}",

"Estimated Cost ($/sqft)": "${:.2f}",

"Feasibility Score": "{:.2f}"

})\

.set\_properties(\*\*{

'background-color': '#f0f2f6',

'color': 'black',

'border-color': '#ffffff'

})

st.dataframe(styled\_df, *use\_container\_width*=True)

*else*:

st.info("Add at least one site to view feasibility results.")