```
import streamlit as st
import pandas as pd
import geopandas as gpd
from shapely.geometry import Point
from streamlit_folium import st_folium
import folium
from streamlit_geolocation import streamlit_geolocation
from geopy.geocoders import Nominatim

# ==== CONFIG ==== #
st.set_page_config(page_title="Bukit Bintang Explorer", layout="wide", page_icon="📍")
st.markdown("""
   <style>
   body {
      background-color: #f4f6fa;
      color: #2b2b2b;
   }
   .stApp {
      background-color: #f4f6fa;
   }
   .full-width-map .folium-map {
      width: 100% !important;
   }
   </style>
""", unsafe_allow_html=True)

# ==== SIDEBAR MENU ==== #
st.sidebar.title("🏙️ Menu")
page = st.sidebar.radio("Go to", ["🏙️ About Bukit Bintang", "🛰️ Nearest Finder"])

# ==== SECTION 1: ABOUT ==== #
if page == "🏙️ About Bukit Bintang":
   st.title("🏙️ About Bukit Bintang")

   col1, col2 = st.columns([1, 1])
   with col1:
      st.image("Bukit Bintang Front Page (1).jpg", use_container_width=True, caption="Bukit Bintang ✨")
   with col2:
      st.markdown("""
   ### 🌟 Welcome to Bukit Bintang – The Pulse of Kuala Lumpur! 🏙️

   Step into the dynamic energy of Bukit Bintang, where the city never sleeps and every corner bursts with life, colour, and excitement. Whether you're a first-time visitor or a KL local, this iconic district is your ultimate destination for shopping, food, nightlife, and unforgettable moments.

   #### ✨ Why Bukit Bintang is a Must-Visit:
```

🛍️ **Shop ‘til You Drop** – Explore world-class malls like Pavilion Kuala Lumpur, Fahrenheit88, Lot 10, and Berjaya Times Square, packed with international brands, local designers, and exclusive finds.

🍜 **A Culinary Wonderland** – From sizzling street food at Jalan Alor to chic rooftop restaurants, artisanal cafes, and global dining spots – there’s something delicious around every corner.

🎶 **Unmatched Nightlife** – As the sun sets, Bukit Bintang lights up with rooftop bars, live music venues, trendy nightclubs, and karaoke lounges that keep the party going till dawn.

🚇 **Seamless Connectivity** – Easily accessible by Monorail, MRT, LRT, and public buses – getting here is a breeze, no matter where you're coming from.

🎭 **A Place Where Cultures Collide** – Experience the perfect blend of modern luxury and authentic charm, where traditional markets sit alongside high-fashion boutiques and street performers liven up every night.

Bukit Bintang is not just a place – it’s a vibe. Come for the buzz, stay for the memories. ✨
""", unsafe_allow_html=True)

st.markdown("---")

# ==== SECTION 2: FIND NEAREST ==== #
elif page == "📍 Nearest Finder":
st.title("📍 Find Nearest Facility")

with st.container():
st.subheader("📌 Detect Your Location")
st.write("Click the button below to get your current location (if your browser allows it).")
location = streamlit_geolocation()

default_lat, default_lon = 3.1475, 101.7118

if location and location['latitude'] and location['longitude']:
default_lat = location['latitude']
default_lon = location['longitude']
st.success(f"📍 Auto-detected location: ({default_lat:.4f}, {default_lon:.4f})")
else:
st.info("⚠️ Using default location (Bukit Bintang)")

# Optional: search by name
geolocator = Nominatim(user_agent="streamlit-app")
place_name = st.text_input("🔍 Search by address or landmark (optional)")
if place_name:
result = geolocator.geocode(place_name)

```
            if result:
                default_lat, default_lon = result.latitude, result.longitude
                st.success(f"📍 Found: ({default_lat:.6f}, {default_lon:.6f})")
            else:
                st.error("❌ Location not found.")

        # Load facilities
        df = pd.read_csv(
            r"C:\Users\User\Documents\2025 Master UiTM\Sem 1\GES 716 Programming\INDIVIDUAL PROJECT\Facilities.csv",
            encoding='ISO-8859-1'
        )
        df.columns = df.columns.str.strip().str.lower()
        required_cols = ['longitude', 'latitude', 'name', 'type', 'contact', 'email', 'office hour']
        if not all(col in df.columns for col in required_cols):
            st.error("❌ CSV missing required columns.")
            st.stop()

        gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df['longitude'], df['latitude']), crs="EPSG:4326")

        # Facility filter
        st.markdown("### 🏥 Select Facility Type")
        facility_type = st.selectbox("Choose facility type:", sorted(gdf['type'].dropna().unique()))
        gdf_filtered = gdf[gdf['type'] == facility_type].copy()

        if gdf_filtered.empty:
            st.warning("No facilities found.")
            st.stop()

        # Map to select location
        st.markdown("### 🗺️ Pinpoint Your Location")
        pin_map = folium.Map(location=[default_lat, default_lon], zoom_start=14)
        pin_map.add_child(folium.LatLngPopup())
        folium.Marker(
            [default_lat, default_lon],
            tooltip="Current Location",
            icon=folium.Icon(color="blue")
        ).add_to(pin_map)

        # Show map full-width
        st.markdown('<div class="full-width-map">', unsafe_allow_html=True)
        map_data = st_folium(pin_map, height=600, use_container_width=True)
        st.markdown('</div>', unsafe_allow_html=True)

        if map_data and map_data.get("last_clicked"):
            lat = map_data["last_clicked"]["lat"]
```

```
            lon = map_data["last_clicked"]["lng"]
            st.success(f" 🖱️ Selected location: ({lat:.6f}, {lon:.6f})")
        else:
            lat, lon = default_lat, default_lon
            st.info(f" 📍 Using: ({lat:.6f}, {lon:.6f})")

        # Find nearest 3
        try:
            user_point = Point(lon, lat)
            user_proj = gpd.GeoSeries([user_point], crs="EPSG:4326").to_crs(epsg=3857)
            gdf_proj = gdf_filtered.to_crs(epsg=3857)
            gdf_proj["distance"] = gdf_proj.geometry.distance(user_proj[0])
            nearest_3 = gdf_proj.sort_values("distance").head(3)

            st.markdown("### 🖷 Nearest 3 Facilities")
            for _, row in nearest_3.iterrows():
                dist_km = row['distance'] / 1000
                contact = row['contact'] if pd.notna(row['contact']) else "N/A"
                email = row['email'] if pd.notna(row['email']) else "N/A"
                hours = row['office hour'] if pd.notna(row['office hour']) else "N/A"

                info_text = f"""
**{row['name']}** ({row['type']})
📍 {dist_km:.2f} km away
📞 **Phone:** {contact}
✉️ **Email:** {email}
⏰ **Hours:** {hours}
"""
                st.info(info_text)

        except Exception as e:
            st.error(f"Something went wrong: {e}")
```