PROJECT CODE

```python
import streamlit as st

from PIL import Image

import altair as alt

import pandas as pd

from streamlit_marquee import streamlit_marquee

from pydataset import data

import folium

from geopy.geocoders import Nominatim

from dbf import Table  # Import the correct class

from streamlit_folium import folium_static

import json

import os

import plotly.express as px

import plotly.graph_objects as go

import time

import requests

from streamlit_lottie import st_lottie


# Set page title and icon

st.set_page_config(

    page_title="GLIS",

    page_icon=" 🌍 ",

)


# Opening the image

image = Image.open(r"C:\Users\HARSHINI\Downloads\Screenshot_2023-10-30_134617-removebg-preview.png")

st.sidebar.image(image, use_column_width=True)
```

```python
# Create a sidebar for the menu

st.sidebar.title("Menu")


# Define the menu options

menu_options = ["Home", "Land Information", "Datasets", "Dashboard", "Chatbot"]


# Create a selectbox widget for navigation

selected_option = st.sidebar.selectbox("Select Page:", menu_options)


# Define the content for each menu option

if selected_option == "Home":

    # Create a container for title and login button

    st.markdown("<h1 style='text-align: center; color: black;'>Government Land Informative System -
(GLIS)</h1>", unsafe_allow_html=True)


    # Use forward slashes in the file path

    st.video(r"C:\Users\HARSHINI\Downloads\Untitled design.mp4")


    def load_lottieurl(url: str):

        r = requests.get(url)

        if r.status_code != 200:

            return None

        return r.json()


    lottie_url_hello = "https://lottie.host/d076e07c-91fb-4b28-b443-24d10c3c6f50/EnCn0jBhv5.json"

    lottie_hello = load_lottieurl(lottie_url_hello)


    st.markdown("<h1 style='text-align: center; color: black;'>Visualizing Land Resources</h1>",
unsafe_allow_html=True)
```

```python
# Create two columns

col1, col2 = st.columns([1, 1])


with col1:

    st.write("""

        The Government Land Information System (GLIS) is an invaluable resource that houses an
extensive repository of geospatial data pertaining to land resources, ownership, boundaries, land
utilization, and a wealth of other critical information. This comprehensive system serves as a
cornerstone in the realm of land management and governance, providing users with unparalleled
access to data that empowers decision-makers, researchers, and the public alike. Through the GLIS
Dashboard, we offer a user-friendly interface to harness the rich insights within the GLIS, enabling
you to explore, analyze, and visualize land-related information, thereby promoting informed
decision-making and greater transparency in land resource management. Welcome to a world of
geospatial knowledge at your fingertips.

        """)
    with col2:

        st_lottie(lottie_hello, key="hello", width=350, height=330)



elif selected_option == "Datasets":

    st.markdown("<h1 style='text-align: center; color: black;'>Datasets</h1>",
unsafe_allow_html=True)


    streamlit_marquee(**{

    # the marquee container background color

    'background': "white",

    # the marquee text size

    'font-size': '12px',

    # the marquee text color

    "color": "red",

    # the marquee text content

    'content': 'You are now on the Datasets page. You can find your required Datasets here.',

    # the marquee container width

    'width': '1000px',
```

```python
    # the marquee container line height
    'lineHeight': "35px",
    # the marquee duration
    'animationDuration': '20s',
})


    dataset_paths = {
        'Land Utilization': r"C:\Users\HARSHINI\Downloads\Land_Utilization.csv",
        'Classification of Land(Year-Wise)': r"C:\Users\HARSHINI\Downloads\Land_Utilization (1).csv",
        'Land Allocation': r"C:\Users\HARSHINI\Downloads\PATTERN_OF_LAND_UTILISATION_0.csv",
        'StateWise Report(2008-15)':
r"C:\Users\HARSHINI\Downloads\PATTERN_OF_LAND_UTILISATION_0.csv",
        'StateWise Report(2015-23)':
r"C:\Users\HARSHINI\Downloads\PATTERN_OF_LAND_UTILISATION_0.csv"
    }


    # Sidebar for the "Datasets" page
    st.sidebar.header('Select Dataset')
    selected_dataset = st.sidebar.selectbox('Choose a dataset', list(dataset_paths.keys()))


    # Read selected dataset
    @st.cache_data
    def load_data(path):
        return pd.read_csv(path)


    df = load_data(dataset_paths[selected_dataset])


    # Main content for the "Datasets" page
    st.subheader('Display Selected Dataset')
    st.write(f'You have selected: {selected_dataset}')
    st.write('Displaying the selected dataset:')
    st.write(df)
```

```python
# ...

elif selected_option == "Land Information":

    st.markdown("<h1 style='text-align: center; color: black;'>Land Parcel Information</h1>",
unsafe_allow_html=True)


    streamlit_marquee(**{

    # the marquee container background color

    'background': "white",

    # the marquee text size

    'font-size': '12px',

    # the marquee text color

    "color": "red",

    # the marquee text content

    'content': 'This page provides details about land parcels.',

    # the marquee container width

    'width': '1000px',

    # the marquee container line height

    'lineHeight': "35px",

    # the marquee duration

    'animationDuration': '15s',

})


    # Create a map of India with state boundaries

    m = folium.Map(location=[20.5937, 78.9629], zoom_start=5)  # Centered on India, you can adjust
the zoom level


    # You'll need the GeoJSON data of Indian states and the DBF file. Make sure to provide the correct
file paths.

    geojson_path = r"C:\Users\HARSHINI\Downloads\india-osm.geojson"  # Replace with the actual
file paths
```

```python
folium.GeoJson(
    geojson_path,
    name='geojson'
).add_to(m)



# Centering the map on the page
st.write('<h3 align="center">Map of Indian States</h3>', unsafe_allow_html=True)


# Add CSS styling to center the map
folium_static(m, width=1000, height=400)


# Add CSS to center the map
st.markdown(
    """
    <style>
    .map-container {
        display: flex;
        justify-content: center;
    }
    </style>
    """,
    unsafe_allow_html=True,
)



# Create two columns
col1, col2 = st.columns([1, 1])


# Add a paragraph to the first column
```

```python
col1.write("""

     Land parcel information refers to detailed data about specific pieces of land, providing essential
insights into various aspects of a particular property. This information typically includes crucial details
such as land boundaries, geographical coordinates, land size, topographical features, and legal
descriptions. Additionally, it encompasses information about land use classification, zoning
regulations, and any environmental constraints or considerations relevant to the parcel. Land parcel
information serves as a fundamental resource for land management, urban planning, real estate
development, and infrastructure projects, facilitating informed decision-making processes. It aids in
assessing the suitability of land for specific purposes, evaluating potential risks or constraints
associated with development, and ensuring compliance with land-use regulations and policies.

    """)


  # Add an image to the second column

  image = Image.open(r"C:\Users\HARSHINI\Downloads\home-ins.png")

  # Replace 'path_to_your_image.png' with your actual image file path

  col2.image(image,width=400)


  # Function to load GeoJSON data based on the selected state

  def load_geojson(state):

    geojson_path = os.path.join(r"C:\Users\HARSHINI\Downloads\geojson files", f'{state}.geojson')
# Replace 'path_to_geojson_folder' with your folder path

    with open(geojson_path, 'r', encoding='utf-8') as geojson_file:

        return json.load(geojson_file)


  # Function to load dataset based on the selected state

  def load_data(state):

    file_path = os.path.join(r"C:\Users\HARSHINI\Desktop\Datasets", f'{state}.csv')  # Replace
'path_to_csv_folder' with your folder path

    return pd.read_csv(file_path)


  # Dummy data for the dropdown

  states = ['Andhra Pradesh', 'Arunachal Pradesh', 'Assam', 'Bihar', 'Chhatisgarh', 'Goa', 'Gujarat',
'Haryana', 'Himachal Pradesh', 'Jharkhand', 'Karnataka', 'Kerala', 'Madhya Pradesh','Maharastra',
'Manipur', 'Meghalaya', 'Mizoram', 'Nagaland', 'Odissa', 'Punjab', 'Rajasthan', 'Sikkim', 'Tamil Nadu',
'Telangana', 'Tripura', 'Uttar Pradesh', 'UttarKhand', 'West Bengal']  # Add all 29 states
```

```python
# Main function
def main():
    st.title('Indian States Data Visualizer')

    # Dropdown for states
    state_selected = st.selectbox('Select a state:', states)

    # Load GeoJSON data and dataset based on the selected state
    geojson_data = load_geojson(state_selected)
    dataset = load_data(state_selected)

    # Create two columns
    col1, col2 = st.columns([1, 1])

    # Column 1 - Folium map
    with col1:
        st.subheader(f'Folium Map')
        m = folium.Map(location=[20.5937, 78.9629], zoom_start=5)  # Centered on India, adjust zoom level
        folium.GeoJson(geojson_data, name='geojson').add_to(m)
        folium_static(m,width=490, height=400)

    # Column 2 - Dataset
    with col2:
        st.subheader(f'Dataset for {state_selected}')
        st.write(dataset)

if __name__ == '__main__':
    main()
```

```python
# ... (the rest of your code)


# Move the chart code to the "Dashboard" page
# ...


elif selected_option == "Dashboard":
    st.markdown("<h1 style='text-align: center; color: black;'>Dashboard</h>",
unsafe_allow_html=True)


    streamlit_marquee(**{
    # the marquee container background color
    'background': "white",
    # the marquee text size
    'font-size': '12px',
    # the marquee text color
    "color": "red",
    # the marquee text content
    'content': 'This dashboard provides insights and visualizations related to the Government Land
Informative System (GLIS).',
    # the marquee container width
    'width': '1000px',
    # the marquee container line height
    'lineHeight': "35px",
    # the marquee duration
    'animationDuration': '20s',
})


# Move the chart code to the "Dashboard" page
    # Long text split into smaller sections
    dashboard_text = [
        "Here's more information about the dashboard...",
```

```python
    "You can include additional details and explanations in these sections.",
]


for text_section in dashboard_text:
    st.write(text_section)


# Create three columns
col1, col2 = st.columns(2)


# Interactive Chart 1: Land Utilization (Bar Chart)
with col1:
    st.subheader("Land Utilization Analysis")
    land_utilization_data = [10, 20, 30, 40, 50]
    land_utilization_labels = ["Residential", "Commercial", "Agricultural", "Industrial", "Other"]
    land_utilization_color = "skyblue"


    # Create an Altair chart with specified color
    land_utilization_df = pd.DataFrame({"Land Type": land_utilization_labels, "Land Utilization":
land_utilization_data})
    chart1 = alt.Chart(land_utilization_df).mark_bar(color=land_utilization_color).encode(
        x=alt.X("Land Type:O", title="Land Type"),
        y=alt.Y("Land Utilization:Q", title="Land Utilization")
    )
    st.altair_chart(chart1, use_container_width=True)


# Interactive Chart 2: Property Ownership (Line Chart)
with col2:
    st.subheader("Property Ownership Analysis")
    property_ownership_data = [15, 25, 10, 30, 20]
    property_ownership_labels = ["Private", "Government", "Corporate", "Individual", "Other"]
    property_ownership_color = "red"
```

```python
    # Create a Line Chart with specified color
    property_ownership_df = pd.DataFrame({"Ownership Type": property_ownership_labels,
"Ownership Percentage": property_ownership_data})
    chart2 = alt.Chart(property_ownership_df).mark_line(color=property_ownership_color).encode(
        x=alt.X("Ownership Type:O", title="Ownership Type"),
        y=alt.Y("Ownership Percentage:Q", title="Ownership Percentage")
    )
    st.altair_chart(chart2, use_container_width=True)


  # Function to load dataset based on the selected state
  def load_data(state):
    file_path = os.path.join(r"C:\Users\HARSHINI\Desktop\Datasets", f'{state}.csv')  # Replace
'path_to_folder' with the actual folder path
    df = pd.read_csv(file_path)  # Load the CSV file
    return df


  # Dummy data for the dropdown
  states = ['Andhra Pradesh', 'Arunachal Pradesh', 'Assam', 'Bihar', 'Chhatisgarh', 'Goa', 'Gujarat',
'Haryana', 'Himachal Pradesh', 'Jharkhand', 'Karnataka', 'Kerala', 'Madhya Pradesh','Maharastra',
'Manipur', 'Meghalaya', 'Mizoram', 'Nagaland', 'Odissa', 'Punjab', 'Rajasthan', 'Sikkim', 'Tamil Nadu',
'Telangana', 'Tripura', 'Uttar Pradesh', 'UttarKhand', 'West Bengal']  # Add all 29 states


  # Main function
  def main():
    st.title('Indian States Data Visualizer')


    # Dropdown for states
    state_selected = st.selectbox('Select a state:', states)


    # Load dataset based on the selected state
    dataset = load_data(state_selected)
```

```python
# Create donut chart
fig1 = px.pie(dataset, values='Agriculture Land', names='Year', title='Agriculture Land by Year')


# Area chart
fig2 = px.area(dataset, x='Year', y='Infrastructures', color_discrete_sequence=['purple'],
title='Area Chart for Various Infrastructures')


# Scatter plot
fig3 = px.scatter(dataset, x='Year', y='Water Bodies', color_discrete_sequence=['red'],
title='Scatter Plot of Water Bodies over Years')


# Bar chart
fig4 = px.bar(dataset, x='Year', y='Industries', color='Industries', title='Bar Chart of Industries')


# Bubble chart
fig5 = px.scatter(dataset, x='Year', y='Waste Land', size='Industries', color='Agriculture Land',
color_discrete_sequence=px.colors.qualitative.Set1, title='Bubble Chart for Waste Land')


# Histogram
fig6 = px.histogram(dataset, x='Year', y='Forest Land', color='Agriculture Land', title='Histogram
of Forest Land')


# Split the page into rows and columns
c1, c2 = st.columns((1, 1))


# Row 1
with c1:
    st.plotly_chart(fig1, use_container_width=True)
with c2:
    st.plotly_chart(fig2, use_container_width=True)


# Row 2
```

```python
    with c1:
        st.plotly_chart(fig3, use_container_width=True)
    with c2:
        st.plotly_chart(fig4, use_container_width=True)


    # Row 3
    with c1:
        st.plotly_chart(fig5, use_container_width=True)
    with c2:
        st.plotly_chart(fig6, use_container_width=True)


    st.markdown("<h5 style='text-align: center; color: red;'>Visualizing Dynamic Land Records Over
Time</h5>", unsafe_allow_html=True)


    # Create figure
    fig = go.Figure()


    # Add initial data to the figure
    fig.add_trace(go.Scatter(x=[0], y=[0], mode='lines+markers'))


    # Create animation frames
    frames = []
    for i in range(len(dataset)):
        frames.append(go.Frame(data=[go.Scatter(x=dataset['Year'][:i+1], y=dataset['Agriculture
Land'][:i+1])],
                    traces=[0],
                    name=f'frame_{i}'))


    # Add animation
    fig.frames = frames
    fig.update_layout(updatemenus=[dict(type='buttons', showactive=False, buttons=[dict(label='*',
method='animate', args=[None, {'frame': {'duration': 500, 'redraw': True}, 'fromcurrent': True}])])])
```

```python
    st.plotly_chart(fig,use_container_width=True)


if __name__ == '__main__':
    main()




elif selected_option == "Chatbot":

    st.markdown("<h1 style='text-align: center; color: black;'>How can I help you!</h1>",
unsafe_allow_html=True)


    def load_lottieurl(url: str):

        r = requests.get(url)

        if r.status_code != 200:

            return None

        return r.json()


    # Lottie animation URL (replace this with your own Lottie animation URL)

    lottie_url_hi = "https://lottie.host/2101fe8c-c6f8-4c17-9680-3be18d175832/xDmBLNP8GA.json"

    lottie_hi = load_lottieurl(lottie_url_hi)


    # Center the Lottie animation using CSS

    st.markdown(

        """

        <style>

            body {

                margin: 0;

                padding: 0;

                display: flex;

                flex-direction: column;

                min-height: 100vh;
```

```css
}
main {

    flex: 1;

    display: flex;

    flex-direction: column;

    align-items: center; /* Center content horizontally */

    padding-bottom: 80px; /* Adjust the padding to accommodate the fixed search bar */

}
.stLottie {

    display: flex;

    justify-content: center;

    align-items: center;

    height: 70vh;  /* Adjust the height as needed */

    max-width: 100%; /* Ensure the animation doesn't exceed the screen width */

}
.chat-container {

    display: flex;

    flex-direction: row;

    align-items: flex-start;

    padding: 10px;

    margin-bottom: 10px;

    margin-top: 10px;

}
.stTextInput {

    position: fixed;

    bottom: 20px; /* Adjust the distance from the bottom */

    width: 50%; /* Adjust the width of the search bar */

    margin-left: 120px;

    margin-right: 50%;

    background: linear-gradient(to right, #e66465, #9198e5);

    border-radius: 10px;  /* Set border-radius for a rectangular box */
```

```
            padding: 5px;  /* Adjust padding as needed */


        }
        .chatbot-response {
            background-color: #ccffcc;  /* Adjust the background color */
            border-radius: 10px;  /* Adjust the border radius */
            padding: 10px;
            max-width: 50%;  /* Adjust the maximum width */
            margin-left: 350px;  /* Push response to the right */
            margin-bottom: 10px;  /* Add margin to separate from the search bar */
        }
        .user-input {
            background-color: #e6f2ff;  /* Adjust the background color */
            border-radius: 10px;  /* Adjust the border radius */
            padding: 10px;
            max-width: 30%;
            margin-left: 100px;
            margin-bottom: 10px;  /* Add margin to separate from the search bar */
        }
    </style>
    """,
    unsafe_allow_html=True,
)


c1, c2 = st.columns((1, 2.2))
with c1:
    st.markdown("")


with c2:
    # Display the centered Lottie animation
    st_lottie(lottie_hi, key="hi", width=350, height=330, speed=1)
```

```
responses = {

    "greetings": "Hello! How can I help you with information about government lands?",

    "goodbye": "Goodbye! Feel free to return if you have more questions.",

    "info_about_government_lands": "Government lands include public spaces and regulated
areas,Government lands, owned and regulated by public entities, serve diverse purposes, including
public spaces, agriculture, conservation, and economic development. These areas play a vital role in
urban planning, environmental conservation, and the overall well-being of communities. If you have
specific questions, feel free to ask!",

    "lands_info": "Government lands serve various purposes, including public spaces, regulated
areas, and more.Government lands encompass diverse areas serving public purposes, ranging from
open green spaces and regulated zones to vital resources like agricultural lands and wildlife habitats.
Land information is crucial for effective management, guiding policies that balance economic
development, environmental conservation, and community well-being. If you have questions about
specific regulations, feel free to ask.",

    "land_regulations": "Land regulations are in place to ensure proper use and management of
government lands.Land regulations are rules and guidelines established by government authorities
to govern the use, development, and management of land. These regulations ensure responsible and
sustainable land practices, addressing aspects such as zoning, construction standards, and
environmental impact to maintain orderly and efficient land utilization. If you have questions about
specific regulations, feel free to ask.",

    "land_details": "Government lands are typically owned by the public or government entities.
Ownership details can vary.Land details encompass information about the characteristics, use, and
ownership of a particular piece of land, including its size, topography, land use regulations, and
ownership status. These details are essential for effective land management, development planning,
and ensuring compliance with land-use policies. Do you have a specific question?",

    "land_use": "Land use policies govern how government lands are utilized.Land use refers to the
way in which a specific area of land is utilized, encompassing activities such as residential,
commercial, industrial, agricultural, or recreational purposes. Effective land use planning ensures
sustainable development and the optimal allocation of resources to meet societal needs. If you need
information on land use, feel free to specify your question.",

    "agricultural_lands": "Agricultural lands are an essential part of government lands. They are
used for cultivation and farming activities.Agricultural lands are dedicated areas used for cultivating
crops, raising livestock, and conducting farming activities. They play a vital role in food production,
contributing to global sustenance and supporting the livelihoods of farming communities. Do you
have questions about agricultural lands?",

    "forest_info": "Forests are crucial for biodiversity and environmental balance.Forests are vital
ecosystems that provide habitat for diverse plant and animal species, contributing to biodiversity and
environmental balance. They play a crucial role in carbon sequestration, climate regulation, and offer
resources essential for human well-being, such as timber and medicinal plants. If you're interested in
forest information, feel free to ask.",
```

"wildlife_conservation": "Wildlife conservation efforts are essential for preserving biodiversity.Wildlife conservation involves efforts to protect and preserve the diverse species of flora and fauna, aiming to maintain their natural habitats and prevent the extinction of endangered animals. Conservation measures include habitat restoration, anti-poaching initiatives, and public awareness campaigns to promote sustainable coexistence with wildlife. Do you have questions about wildlife conservation on government lands?",

"water_resources": "Government lands often include water resources.Water resources encompass all available sources of water, including rivers, lakes, groundwater, and reservoirs. These vital natural reservoirs sustain ecosystems, support human needs, and play a crucial role in agriculture, industry, and overall environmental balance. If you need information about water resources management, feel free to ask.",

"urban_planning": "Urban planning plays a role in the development of government lands in urban areas.Urban planning involves the systematic organization and design of cities, focusing on land use, infrastructure, and spatial development to create sustainable, functional, and aesthetically pleasing urban environments that cater to the needs of residents and promote overall well-being. It encompasses strategic zoning, transportation networks, and public spaces to foster efficient and harmonious urban living. If you have questions about urban planning, feel free to ask.",

"economic_development": "Government lands contribute to economic development.Economic development refers to the sustained improvement in a region's economic well-being, encompassing increased production, employment opportunities, and the overall standard of living. It involves initiatives and policies aimed at fostering economic growth, diversification, and enhancement of infrastructure to create a thriving and resilient economy. If you're interested in economic aspects, feel free to specify your question.",

"tourism": "Government lands may have tourism significance.Tourism is the practice of traveling for pleasure, leisure, or business to destinations outside one's usual environment. It encompasses activities such as sightseeing, cultural exploration, and relaxation, contributing to economic growth and cultural exchange between regions. If you have questions about tourism on government lands, feel free to ask.",

"energy_production": "Energy production on government lands is a significant aspect.Energy production involves generating usable forms of energy, such as electricity or heat, through various methods such as fossil fuel combustion, nuclear reactions, or harnessing renewable sources like solar, wind, and hydropower. This process is crucial for powering homes, industries, and infrastructure, contributing to economic activities and societal functions. Do you have questions about energy production and government lands?",

"community_engagement": "Community engagement is crucial for the sustainable development of government lands. Do you have questions about community engagement?",

"environmental_impact": "Understanding the environmental impact of activities on government lands is important.Environmental impact refers to the effect of human activities on the environment, encompassing changes in ecosystems, biodiversity, and natural resources. Understanding and mitigating environmental impact are crucial for sustainable development and the preservation of the planet's health. If you have questions, feel free to ask.",

"green_spaces": "Green spaces on government lands contribute to a healthier environment.Green spaces refer to areas within urban or natural environments that are covered with

vegetation, such as parks, gardens, or open fields. These spaces provide essential recreational areas, promote biodiversity, and contribute to a healthier environment by enhancing air quality and reducing the urban heat island effect. If you're interested in green spaces, feel free to ask.",

"public_access": "Ensuring public access to government lands is essential. If you have questions about public access, feel free to ask.",

"disaster_management": "Government lands play a role in disaster management.Disaster management involves the planning, coordination, and implementation of measures to mitigate the impact of natural or man-made disasters. It aims to minimize loss of life, property damage, and disruption to communities through preparedness, response, and recovery strategies. If you have questions about disaster management on government lands, feel free to ask.",

"biodiversity_conservation": "Biodiversity conservation efforts are crucial for maintaining ecological balance.Biodiversity conservation involves the protection and sustainable management of the variety of life on Earth, encompassing diverse species, ecosystems, and genetic resources. It aims to maintain ecological balance, preserve natural habitats, and safeguard the interconnected web of life for the well-being of present and future generations. Do you have questions about biodiversity conservation on government lands?",

"agroforestry": "Agroforestry practices on government lands contribute to sustainable land use.Bio-agroforestry is an agricultural practice that integrates trees, crops, and livestock in a sustainable and symbiotic system. It promotes biodiversity, enhances soil fertility, and provides multiple benefits for both the environment and agricultural productivity. If you're interested in agroforestry, feel free to ask.",

"land_surveying": "Land surveying is important for effective land managementLand surveying is the process of measuring and mapping the Earth's surface to determine the precise locations, distances, and boundaries of land, providing essential information for property delineation, development, and legal documentation. Surveyors use specialized instruments to collect accurate data, ensuring proper land management and construction planning. If you have questions about land surveying on government lands, feel free to ask.",

"pollution_control": "Pollution control measures are essential for maintaining the health of government lands.Pollution control involves implementing measures and regulations to minimize or eliminate the release of harmful pollutants into the environment, safeguarding ecosystems and human health. It encompasses practices such as emission reductions, waste management, and sustainable technologies to mitigate the impact of pollution. Do you have questions about pollution control?",

"waste_management": "Efficient waste management on government lands is crucial.Waste management involves the collection, transportation, and proper disposal or recycling of waste materials to minimize environmental impact and promote public health. It aims to efficiently handle and process waste to mitigate pollution and optimize resource utilization. If you have questions about waste management practices, feel free to ask.",

"renewable_energy": "Exploring renewable energy sources on government lands is important for sustainability.Renewable energy management involves overseeing the sustainable production, distribution, and utilization of energy derived from renewable sources, such as solar, wind, and hydropower. It aims to optimize the efficient use of clean energy technologies to reduce

environmental impact and promote a shift towards more sustainable and eco-friendly power systems. If you're interested in renewable energy, feel free to ask.",

"indigenous_land_rights": "Respecting indigenous land rights is important for ethical land management.Indigenous land rights pertain to the legal and cultural recognition of ancestral lands owned and occupied by indigenous communities. These rights acknowledge the unique connection indigenous peoples have to their traditional territories and aim to protect their heritage, self-determination, and sustainable land use. If you have questions about indigenous land rights, feel free to ask.",

"land_degradation": "Preventing land degradation on government lands is a key concern.Land degradation refers to the deterioration of the quality and productivity of land, often caused by human activities such as deforestation, overgrazing, and improper agricultural practices. It results in the loss of fertile soil, biodiversity, and ecosystem services, leading to environmental challenges and reduced land functionality. If you have questions about land degradation, feel free to ask.",

"land_acquisition": "Understanding the process of land acquisition for government projects is important.Land acquisition is the process by which government or private entities acquire parcels of land for specific purposes such as infrastructure development, public projects, or conservation efforts. This involves legal and administrative procedures to transfer ownership from private individuals or entities to the acquiring entity. If you have questions about land acquisition, feel free to ask.",

"flood_management": "Managing floods on government lands is crucial for minimizing damage.Flood management involves the implementation of strategies and measures to minimize the impact of flooding, including the construction of flood barriers, early warning systems, and sustainable land-use planning to reduce vulnerability to flood risks. Effective flood management aims to protect communities, infrastructure, and natural environments from the adverse effects of floods. If you have questions about flood management, feel free to ask.",

"rural_development": "Government lands play a role in rural development.Rural development involves initiatives and policies aimed at improving the economic, social, and infrastructural conditions in rural areas. It focuses on enhancing livelihoods, promoting agriculture, and addressing the unique challenges faced by rural communities to foster overall growth and well-being. If you're interested in rural development aspects, feel free to ask.",

"property_tax": "Property tax on government lands is a source of revenue.Property tax is a local government levy on the assessed value of real estate properties, serving as a primary source of revenue for funding public services and infrastructure. Property owners are required to pay this tax annually based on the value of their properties. If you have questions about property tax, feel free to ask.",

"land_records": "Maintaining accurate land records is essential for effective land management.Land records are official documents that provide comprehensive information about a specific piece of land, including details on ownership, boundaries, and any encumbrances. These records are crucial for legal and administrative purposes, facilitating transparent land transactions and effective land management. Do you have questions about land records?",

"coastal_management": "Coastal management is important for government lands near coastlines.Coastal management involves strategic planning and implementation of measures to safeguard coastal areas from erosion, flooding, and environmental degradation while promoting

sustainable development and preserving biodiversity. It aims to balance the needs of communities, industry, and ecosystems along coastlines. If you have questions about coastal management, feel free to ask.",

"archaeological_sites": "Government lands may contain archaeological sites. If you're interested in archaeological information, feel free to ask.",

"land_zoning": "Land zoning is an important aspect of land use planning. If you have questions about land zoning on government lands, feel free to ask.",

"farming_communities": "Farming communities on government lands contribute to agricultural activities. If you have questions about farming communities, feel free to ask.",

"land_rights": "Understanding land rights and ownership is crucial for effective land management. Do you have questions about land rights?",

"recreational_spaces": "Providing recreational spaces on government lands is important for the community. If you have questions about recreational spaces, feel free to ask.",

```
    }


    # User input for chat

    user_input = st.text_input("🤖 Message Chatbot:", "")


    # Process user input and generate response

    if user_input:

        # Process the user input

        user_input_lower = user_input.lower()


        # Check for greetings

        if any(greeting in user_input_lower for greeting in ["hello", "hi", "hey", "greetings", "howdy"]):

            st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>", unsafe_allow_html=True)

            st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot: {responses['greetings']}</div>", unsafe_allow_html=True)


        # Check for goodbye

        elif any(goodbye in user_input_lower for goodbye in ["bye", "goodbye", "see you", "farewell"]):

            st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>", unsafe_allow_html=True)
```

```python
        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot: {responses['goodbye']}</div>", unsafe_allow_html=True)


    # Check for information about government lands
    elif any(keyword in user_input_lower for keyword in ["government lands", "land regulations", "land ownership"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>", unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot: {responses['info_about_government_lands']}</div>", unsafe_allow_html=True)


    # Check for land ownership specifically
    elif any(keyword in user_input_lower for keyword in ["land details"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>", unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot: {responses['land_details']}</div>", unsafe_allow_html=True)


    # Continue with more elif statements for additional responses...

    elif any(keyword in user_input_lower for keyword in ["land use"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>", unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot: {responses['land_use']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["agricultural lands"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>", unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot: {responses['agricultural_lands']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["forest info"]):
```

```python
        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['forest_info']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["wildlife conservation"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['wildlife_conservation']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["water resources"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['water_resources']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["urban planning"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['urban_planning']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["economic development"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['economic_development']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["tourism"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['tourism']}</div>", unsafe_allow_html=True)
```

```python
    elif any(keyword in user_input_lower for keyword in ["energy production"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['energy_production']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["community engagement"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['community_engagement']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["environmental impact"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['environmental_impact']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["green spaces"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['green_spaces']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["public access"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['public_access']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["disaster management"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)
```

```python
        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['disaster_management']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["biodiversity conservation"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['biodiversity_conservation']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["agroforestry"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['agroforestry']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["land surveying"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['land_surveying']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["pollution control"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['pollution_control']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["waste management"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['waste_management']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["renewable energy"]):
```

```python
        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['renewable_energy']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["indigenous land rights"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['indigenous_land_rights']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["land degradation"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['land_degradation']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["land acquisition"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['land_acquisition']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["flood management"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['flood_management']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["rural development"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)
```

```python
        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['rural_development']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["property tax"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['property_tax']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["land records"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['land_records']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["coastal management"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['coastal_management']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["archaeological sites"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['archaeological_sites']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["land zoning"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['land_zoning']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["farming communities"]):
```

```python
        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['farming_communities']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["land rights"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['land_rights']}</div>", unsafe_allow_html=True)


    elif any(keyword in user_input_lower for keyword in ["recreational spaces"]):

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown(f"<div class='chat-container chatbot-response'>🤖 Chatbot:
{responses['recreational_spaces']}</div>", unsafe_allow_html=True)


    else:

        st.markdown(f"<div class='chat-container user-input'>🤖 User: {user_input}</div>",
unsafe_allow_html=True)

        st.markdown("<div class='chat-container chatbot-response'>🤖 Chatbot: I'm sorry, I didn't
understand that. Could you please rephrase or ask another question?</div>",
unsafe_allow_html=True)
```