# Assignment 2 - Streamlit

## Question 1:

Create a python programming code review and helper app

**Hierarchy of programs files were saved**

**Lecture4HW/**
- app_code_review1.py **# Main entry file**
- **pages/**
  - app_code_review2.py

**app_code_review1.py**

**CODE1:**

```
import streamlit as st
import google.generativeai as genai

# Configure Generative AI (Ensure API key is set securely)
#Give your correct GEMINI API KEY
genai.configure(api_key="AIzaSyBW0FGG-00000000000XXXXXXXXfkscT3lqIcJ0w1")

# Initialize model
model = genai.GenerativeModel("gemini-1.5-flash")

# Code Review Section
st.title("Code Review ")
st.subheader("Code Review")

code = st.text_area("Enter your programming code to rectify the error:")

if st.button("Code Review"):
    if code.strip():
        response = model.generate_content(f"""
        Rectify the following code:
        {code}

        1. **Code Review** - Provide an overall assessment.
        2. **Error Explanation** - Highlight mistakes and why they occur.
        3. **Fixing Issues** - Suggest changes and where to apply them.
        4. **Corrected Code** - Provide a fully corrected version.
        5. **Optimization Tips** - Suggest ways to improve efficiency.

        Format the response in an easy-to-understand manner with **bold key points**.
        """)
        st.write(response.text)
    else:
        st.warning("Please enter some code before submitting.")

st.page_link("pages/app_code_review2.py", label="Help Desk")
```


After clicking on Code review

**Key improvements:**

- **Secure API Key:** The API key is now retrieved from the `GOOGLE_GENERATIVEAI_API_KEY` environment variable. This is a much more secure method than storing it directly in the code or using `st.secrets` (which is also insecure if the secrets are not properly managed). Set this variable in your system's environment.
- **Error Handling:** A `try-except` block is added to catch potential errors during the API call and display a user-friendly error message.
- **Code Formatting:** The input code is now wrapped in a `python` block within the prompt to ensure the LLM correctly interprets it as Python code. The response is also requested to wrap the corrected code in a `python` block for better readability.
- **Clear Error Message:** If the API key is missing, a clear error message is displayed, and the app stops execution to prevent unexpected behavior.

**To run this:**

1. **Install necessary libraries:** `pip install streamlit google-generativeai`
2. **Set the environment variable:** In your terminal (before running the Streamlit app), set the `GOOGLE_GENERATIVEAI_API_KEY` environment variable to your actual API key:

```
export GOOGLE_GENERATIVEAI_API_KEY="YOUR_API_KEY"
```

3. **Run the Streamlit app:** `streamlit run your_file_name.py` (replace `your_file_name.py` with the actual filename).

Remember to replace `"YOUR_API_KEY"` with your actual Google Generative AI API key. Always prioritize security when handling API keys. Consider using more robust secrets management solutions for production environments.

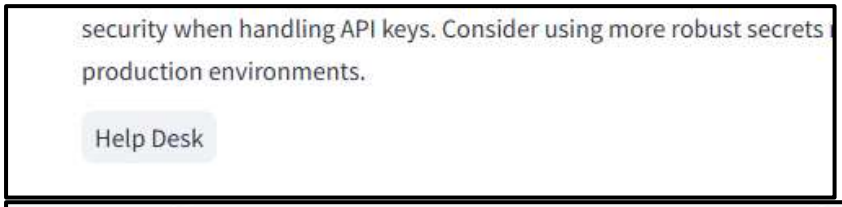
**app_code_review2.py**

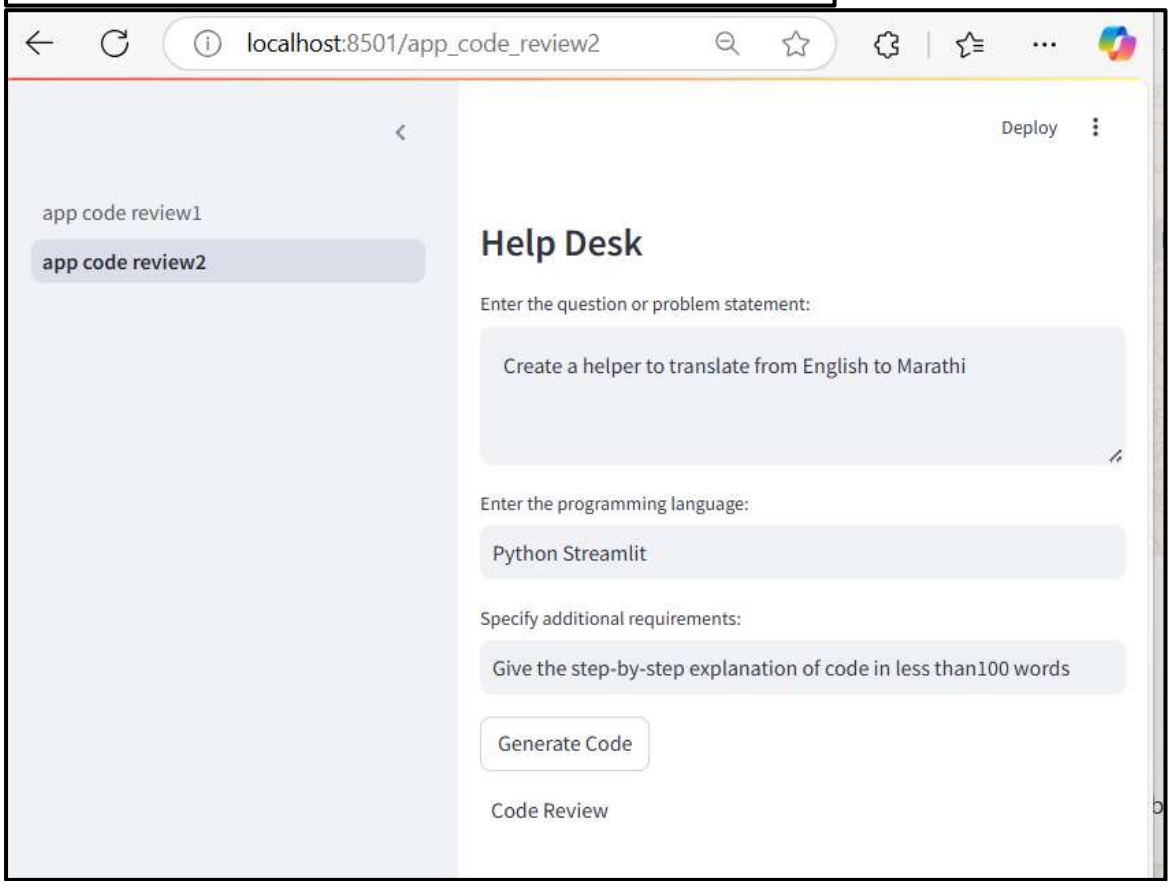
**CODE2:**

```
import streamlit as st
import google.generativeai as genai

# Configure Generative AI API KEY
genai.configure(api_key="AIzaSyBW0FGG-00000000000XXXXXXXXCfkscT3lqIcJ0w1")

# Initialize model
model = genai.GenerativeModel("gemini-1.5-flash")

# Help Desk Section
st.title("Help Desk")

st.subheader("Help Desk")

questions = st.text_area("Enter the question or problem statement:")
programming_language = st.text_input("Enter the programming language:")
requirements = st.text_input("Specify additional requirements:")
```

```
if st.button("Generate Code"):
    if questions.strip() and programming_language.strip():
        response = model.generate_content(f"""
        Generate a {programming_language} program based on the following details:

        **Problem Statement:** {questions}
        **Requirements:** {requirements}

        1. **Brief Explanation** - Describe what the code does.
        2. **Step-by-Step Implementation** - Guide on how to use it.
        3. **Optimized Code Solution** - Provide well-structured code.

        Ensure the output is clear and human-readable.
        """)
        st.write(response.text)
    else:
        st.warning("Please enter both the problem statement and programming language.")

st.page_link("pages/app_code_review2.py", label="Help Desk")
```

**Scroll below, Click on Help Desk button to navigate to Help Desk page**



Generate Code

**1. Brief Explanation:** This Streamlit app uses the Google Translate API to translate English text to Marathi. It takes English input from the user and displays the Marathi translation.

**2. Step-by-Step Implementation:**

1. Install necessary libraries: `pip install streamlit googletrans==4.0.0-rc1`
2. Run the provided code using `streamlit run app.py`.
3. Enter English text in the text box.
4. Click the "Translate" button.
5. The Marathi translation will appear below.

**3. Optimized Code Solution:**

```
import streamlit as st
from googletrans import Translator

st.title("English to Marathi Translator")

translator = Translator()

english_text = st.text_area("Enter English text here:")

if st.button("Translate"):
    try:
        translation = translator.translate(english_text, dest='mr')
        st.write("Marathi Translation:")
        st.success(translation.text)
    except Exception as e:
        st.error(f"An error occurred: {e}")
```

Code Review

**Question 2:**

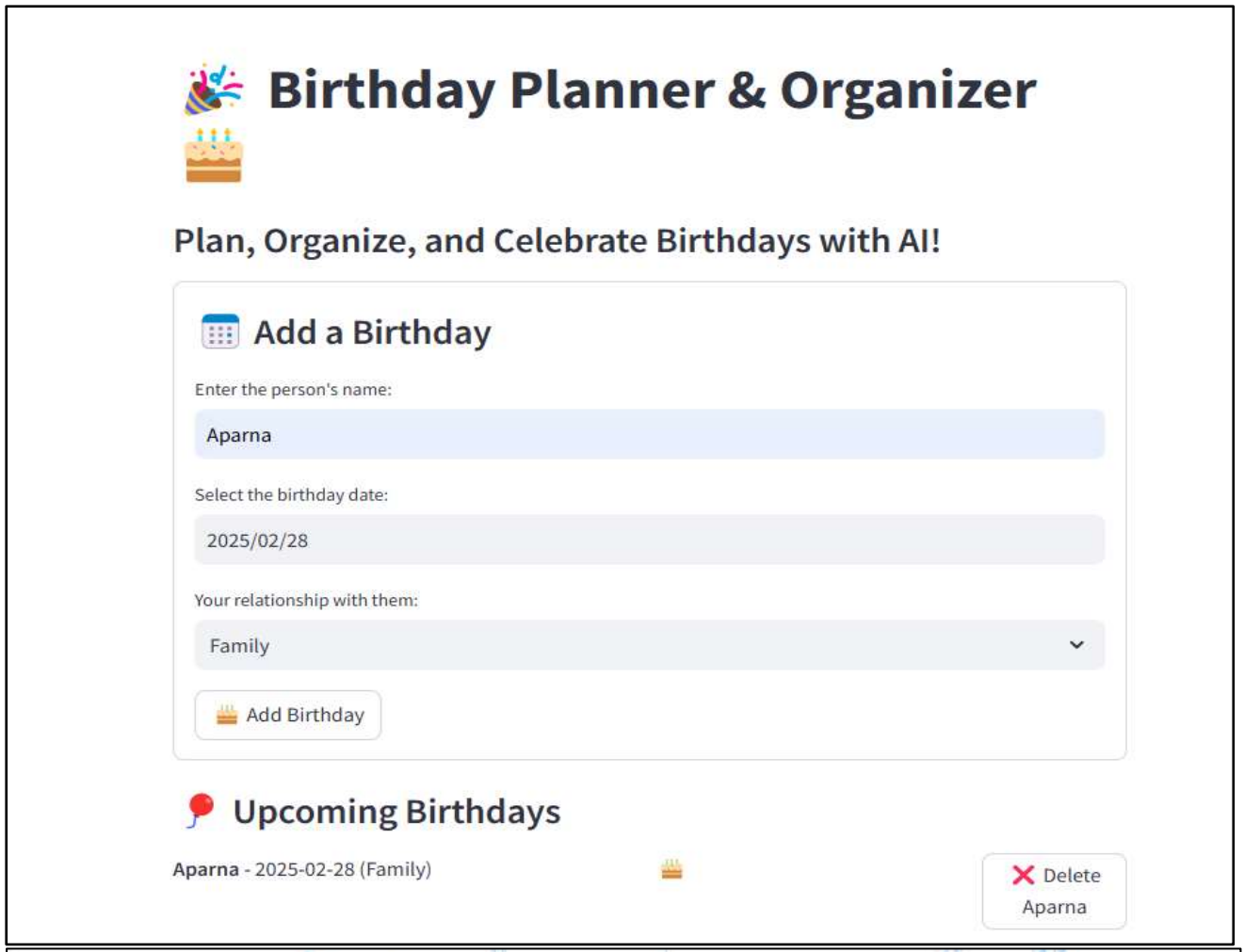
Create a birthday planner and organizer

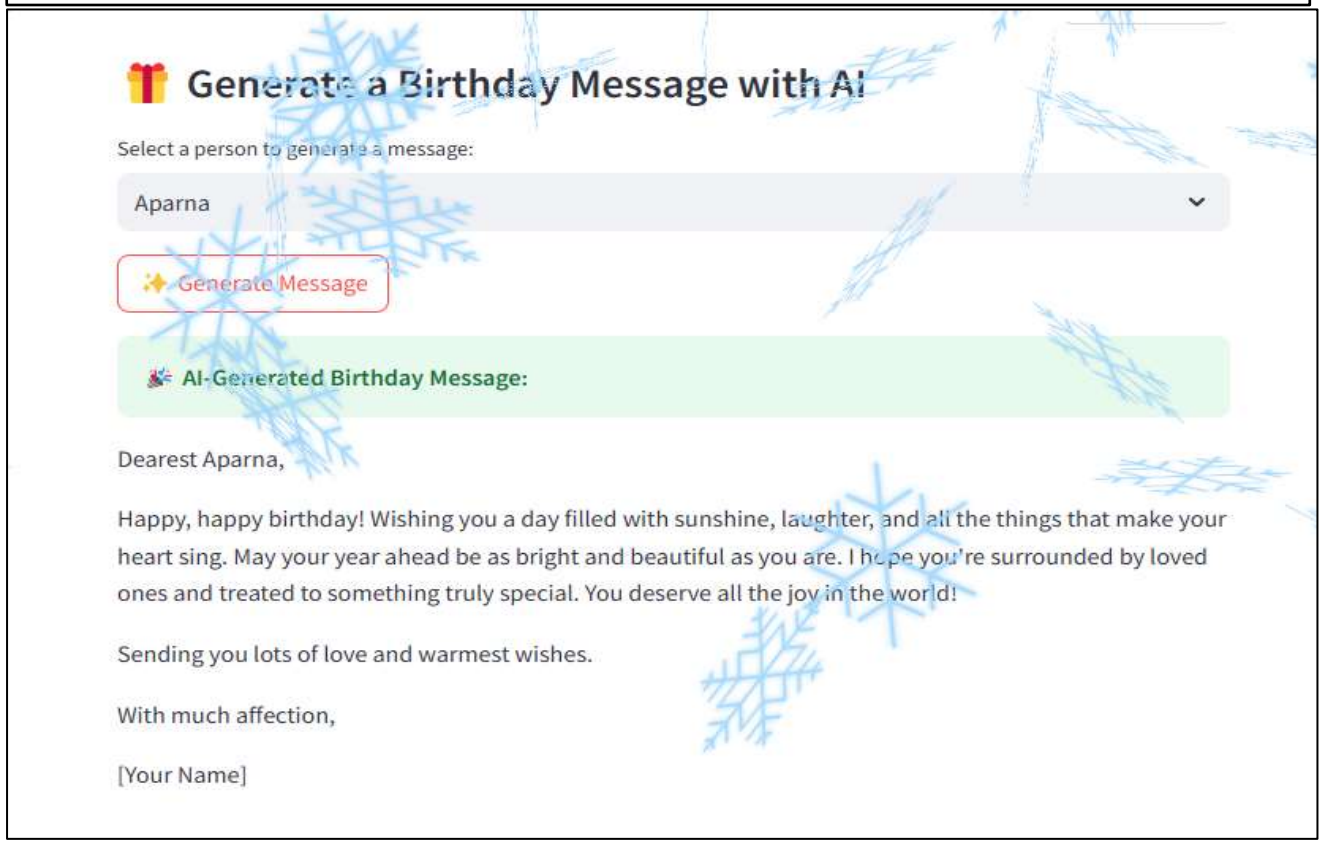
CODE:

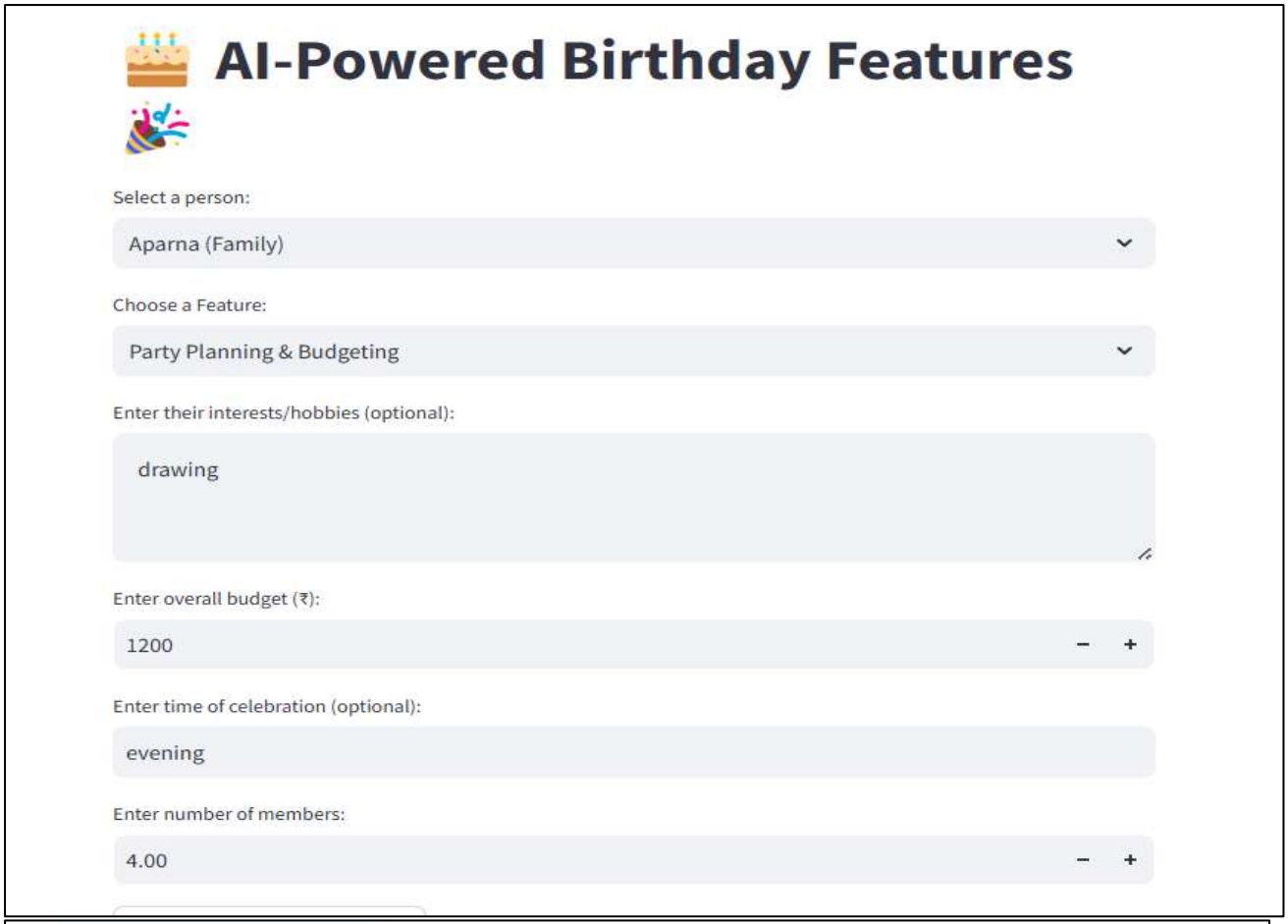
```
import streamlit as st
import google.generativeai as genai
import pandas as pd
from datetime import date, datetime

# Configure Generative AI (Replace with your API key securely)
#Give your correct GEMINI API KEY
genai.configure(api_key=" AIzaSyBW0FGG-00000000000XXXXXXXXfkscT3lqIcJ0w1")
model = genai.GenerativeModel("gemini-1.5-flash")

# 🎉 Show animations when opening the app
st.balloons()  # Celebrate 🎈
st.snow()  # Fun effect ❄️

# ======================================================================
# 🎂 App Title
st.title("🎉 Birthday Planner & Organizer 🎂")
st.subheader("Plan, Organize, and Celebrate Birthdays with AI!")

# Initialize birthday storage if not already set
if "birthdays" not in st.session_state:
    st.session_state.birthdays = []

# 🎈 Add a new birthday
with st.form("birthday_form"):
    st.subheader("🗓️ Add a Birthday")
    name = st.text_input("Enter the person's name:", key="name_input")
    birthday_date = st.date_input("Select the birthday date:")
    relationship = st.selectbox("Your relationship with them:", ["Friend", "Family", "Colleague",
"Other"])

    submit = st.form_submit_button("🎂 Add Birthday")

if submit and name and birthday_date:
    st.session_state.birthdays.append({"Name": name, "Date": birthday_date, "Relationship":
relationship})
    st.success(f"✅ Birthday added for {name} on {birthday_date}!")


# ======================================================================
# 🎈 Display upcoming birthdays with a delete button
st.subheader("🎈 Upcoming Birthdays")
today = datetime.today().date()

if st.session_state.birthdays:
    # Convert the list to a DataFrame
```

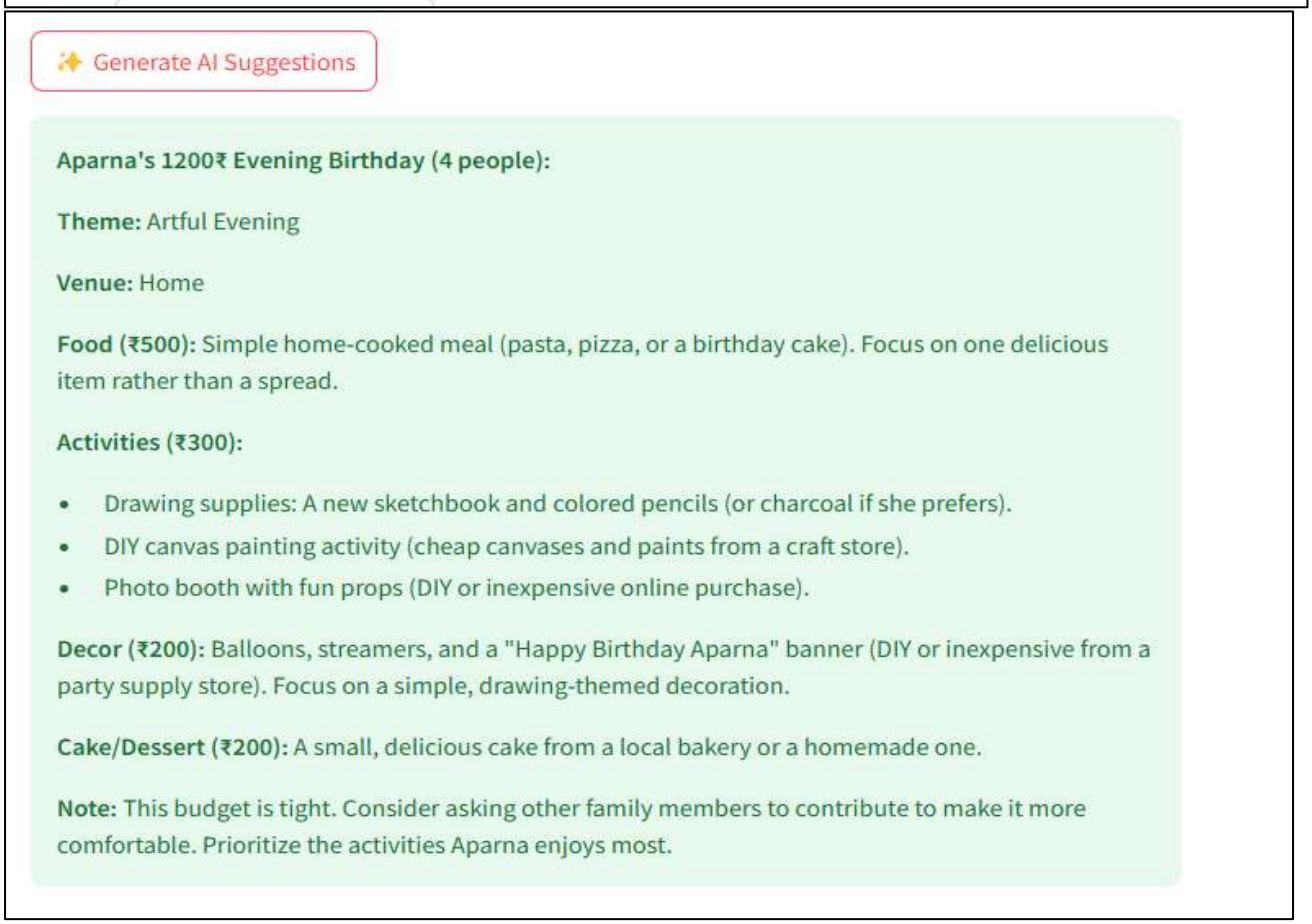
```
    birthdays_df = pd.DataFrame(st.session_state.birthdays)

    # Convert 'Date' column to datetime format
    birthdays_df["Date"] = pd.to_datetime(birthdays_df["Date"]).dt.date

    # Filter for birthdays occurring today or in the future
    upcoming = birthdays_df[birthdays_df["Date"] >= today].sort_values("Date")

    if not upcoming.empty:
        for index, row in upcoming.iterrows():
            col1, col2, col3 = st.columns([3, 2, 1])
            with col1:
                st.write(f"**{row['Name']}** - {row['Date']} ({row['Relationship']})")
            with col2:
                st.write(" 🎂 ")
            with col3:
                # Unique key for delete button
                if st.button(f" ❌ Delete {row['Name']}", key=f"del_{index}"):
                    # Remove the birthday entry
                    st.session_state.birthdays = [
                        b for b in st.session_state.birthdays if not (b["Name"] == row["Name"] and b["Date"] ==
row["Date"])
                    ]
                    st.rerun() # Ensure UI updates
    else:
        st.info("No upcoming birthdays! 🎉 ")

else:
    st.info("No birthdays added yet! 🎉 ")
# ================================================================
# 🎁 AI-Powered Birthday Message Generator
st.subheader(" 🎁 Generate a Birthday Message with AI")
if st.session_state.birthdays:
    bday_person = st.selectbox("Select a person to generate a message:", [b["Name"] for b in
st.session_state.birthdays])
    if st.button(" ✨ Generate Message"):
        try:
            prompt = f"Write a heartfelt birthday message for {bday_person}. Make it warm and
cheerful."
            response = model.generate_content(prompt)
            message = response.text if hasattr(response, "text") else str(response)
            st.success(" 🎉 **AI-Generated Birthday Message:**")
            st.write(message)
        except Exception as e:
            st.error(f" ⚠️ Error generating message: {e}")
else:
    st.warning("Add at least one birthday to generate a message!")

# ================================================================
# 🎂 AI-Powered Birthday Planner & Organizer
```

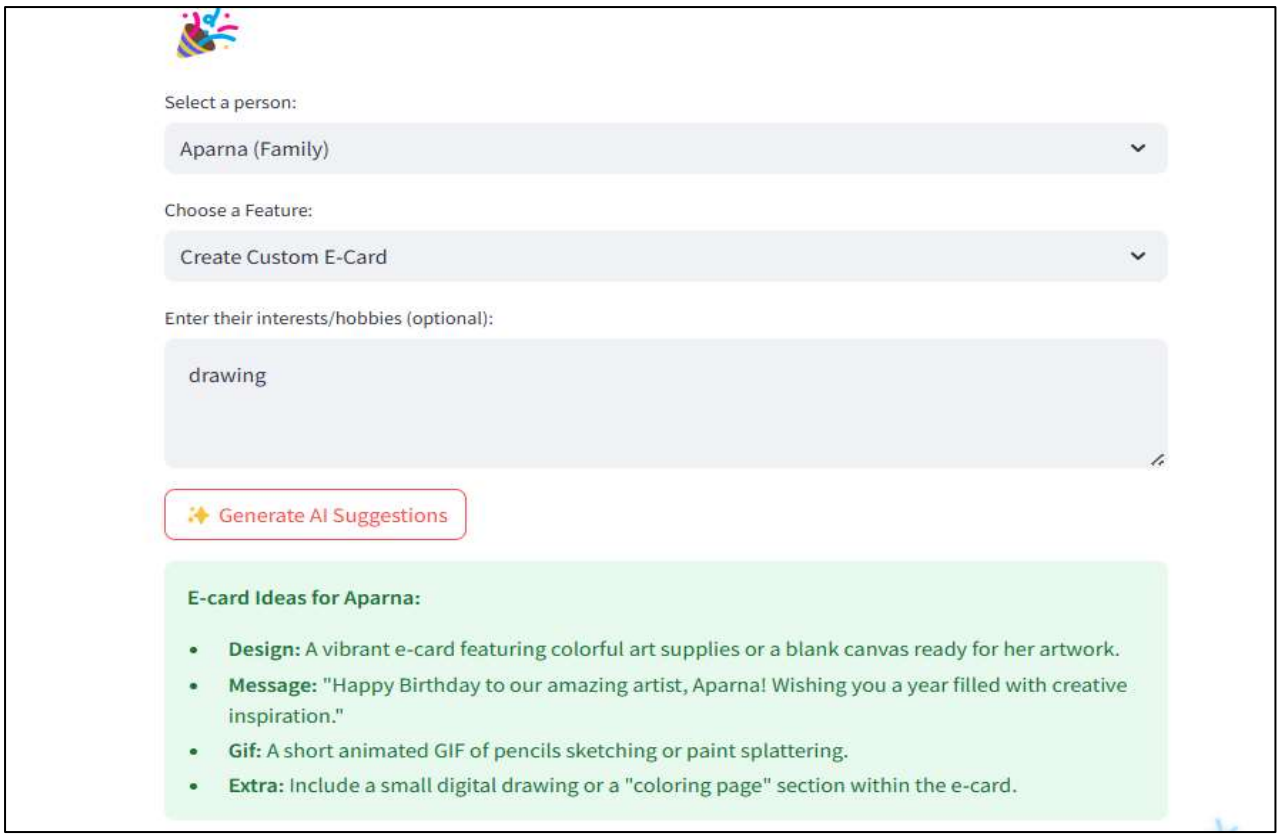
```
st.title("🎂 AI-Powered Birthday Features 🎉")

if st.session_state.birthdays:
    selected_person = st.selectbox("Select a person:", [f"{b['Name']} ({b['Relationship']})" for b in
st.session_state.birthdays])
    selected_name = selected_person.split(" (")[0]
    selected_relationship = next((b["Relationship"] for b in st.session_state.birthdays if b["Name"] ==
selected_name), "Other")

    feature = st.selectbox("Choose a Feature:", [
        "Generate Birthday Wishes",
        "Gift Suggestions",
        "Party Planning & Budgeting",
        "Create Custom E-Card"
    ])

    interests = st.text_area("Enter their interests/hobbies (optional):", key="planner_interests")

    # Only show budget input when planning a party
    if feature == "Party Planning & Budgeting":
        overall_budget = st.number_input("Enter overall budget (₹):", min_value=0, step=500,
key="budget_input")
        timing = st.text_input("Enter time of celebration (optional):", key="time_input")
        members = st.number_input("Enter number of members:")
    if st.button("✨ Generate AI Suggestions"):
        prompt = f"""
        You are an AI birthday assistant. Help with {feature}.
        - Name: {selected_name}
        - Relationship: {selected_relationship}
        - Interests: {interests if interests else 'General'}
        - {f"Overall budget: ₹{overall_budget}" if feature == "Party Planning & Budgeting" else ""}
        - {f"Suggested time: {timing}" if feature == "Party Planning & Budgeting" and timing else ""}
        - {f"Number of members : {members}" if feature == "Party Planning & Budgeting" and timing
else ""}
        Suggest relevant ideas concisely.
        """
        try:
            response = model.generate_content(prompt)
            st.success(response.text if hasattr(response, "text") else str(response))
        except Exception as e:
            st.error(f" ⚠️ Error generating response: {e}")
else:
    st.warning("Please add at least one birthday first!")
```

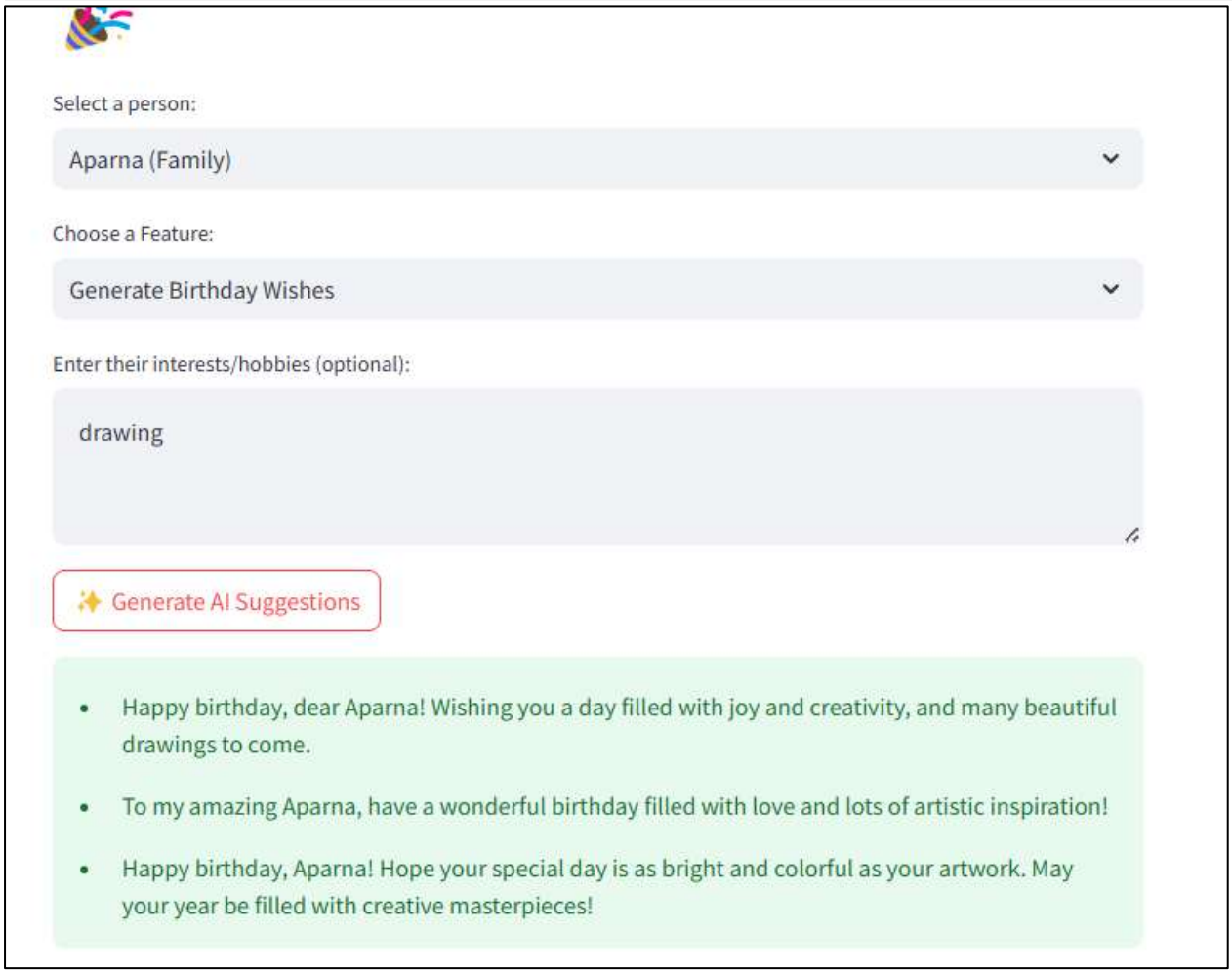

Birthday Planner & Organizer
Plan, Organize, and Celebrate Birthdays with AI!
Add a Birthday
Enter the person's name:
Aparna
Select the birthday date:
2025/02/28
Your relationship with them:
Family
Add Birthday
Upcoming Birthdays
Aparna - 2025-02-28 (Family)
Delete Aparna

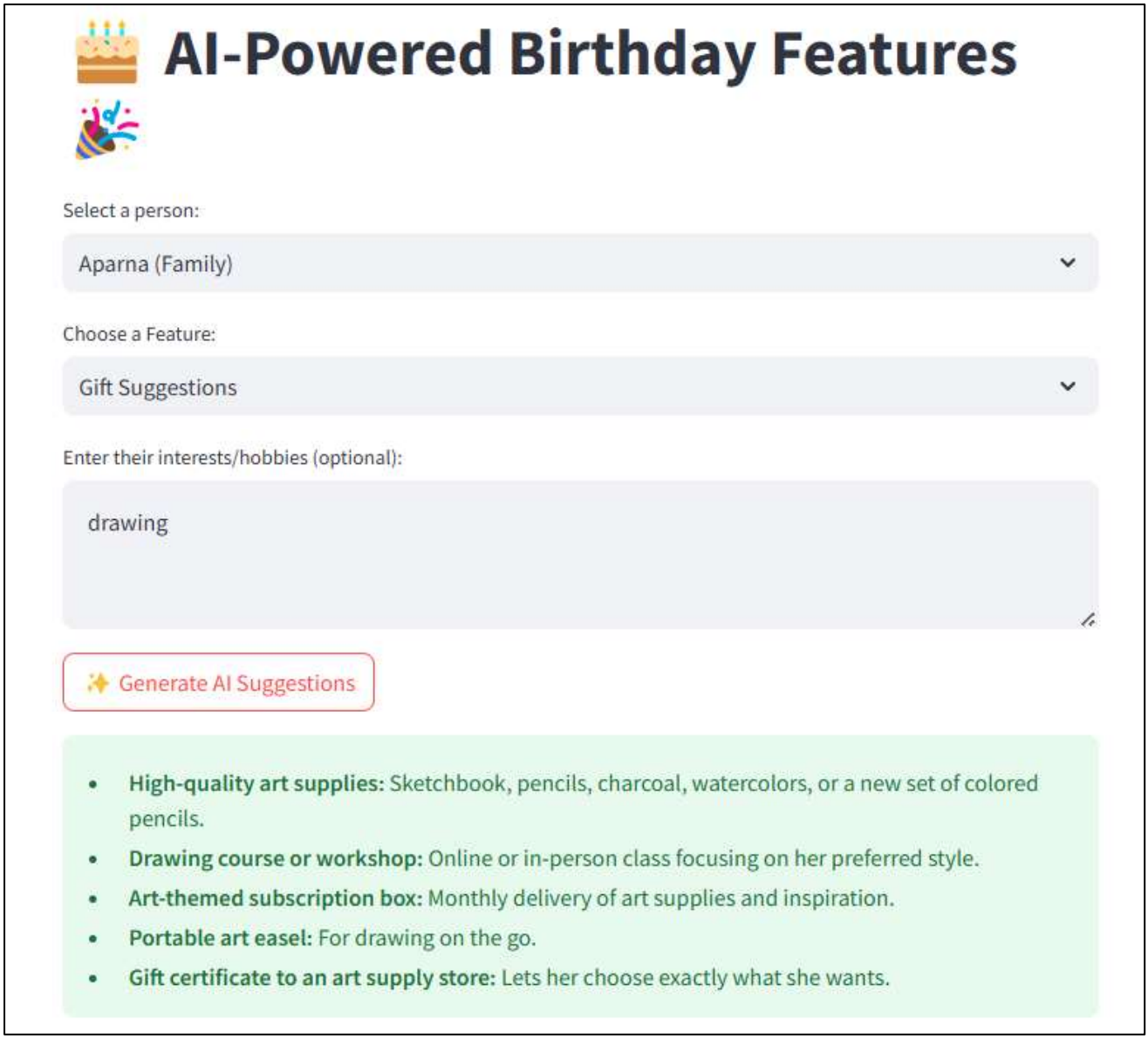

Generate a Birthday Message with AI
Select a person to generate a message:
Aparna
Generate Message
AI-Generated Birthday Message:
Dearest Aparna,
Happy, happy birthday! Wishing you a day filled with sunshine, laughter, and all the things that make your heart sing. May your year ahead be as bright and beautiful as you are. I hope you're surrounded by loved ones and treated to something truly special. You deserve all the joy in the world!
Sending you lots of love and warmest wishes.
With much affection,
[Your Name]


AI-Powered Birthday Features
Select a person:
Aparna (Family)
Choose a Feature:
Party Planning & Budgeting
Enter their interests/hobbies (optional):
drawing
Enter overall budget (₹):
1200
Enter time of celebration (optional):
evening
Enter number of members:
4.00


✨ Generate AI Suggestions
Aparna's 1200₹ Evening Birthday (4 people):
Theme: Artful Evening
Venue: Home
Food (₹500): Simple home-cooked meal (pasta, pizza, or a birthday cake). Focus on one delicious item rather than a spread.
Activities (₹300):
Drawing supplies: A new sketchbook and colored pencils (or charcoal if she prefers).
DIY canvas painting activity (cheap canvases and paints from a craft store).
Photo booth with fun props (DIY or inexpensive online purchase).
Decor (₹200): Balloons, streamers, and a "Happy Birthday Aparna" banner (DIY or inexpensive from a party supply store). Focus on a simple, drawing-themed decoration.
Cake/Dessert (₹200): A small, delicious cake from a local bakery or a homemade one.
Note: This budget is tight. Consider asking other family members to contribute to make it more comfortable. Prioritize the activities Aparna enjoys most.


Select a person:
Aparna (Family)
Choose a Feature:
Create Custom E-Card
Enter their interests/hobbies (optional):
drawing
Generate AI Suggestions
E-card Ideas for Aparna:
Design: A vibrant e-card featuring colorful art supplies or a blank canvas ready for her artwork.
Message: "Happy Birthday to our amazing artist, Aparna! Wishing you a year filled with creative inspiration."
Gif: A short animated GIF of pencils sketching or paint splattering.
Extra: Include a small digital drawing or a "coloring page" section within the e-card.


Select a person:
Aparna (Family)
Choose a Feature:
Generate Birthday Wishes
Enter their interests/hobbies (optional):
drawing
Generate AI Suggestions
Happy birthday, dear Aparna! Wishing you a day filled with joy and creativity, and many beautiful drawings to come.
To my amazing Aparna, have a wonderful birthday filled with love and lots of artistic inspiration!
Happy birthday, Aparna! Hope your special day is as bright and colorful as your artwork. May your year be filled with creative masterpieces!




Choose any of the feature and the output will be generated by AI.

- "Generate Birthday Wishes",
- "Gift Suggestions",
- "Party Planning & Budgeting",
- "Create Custom E-Card"

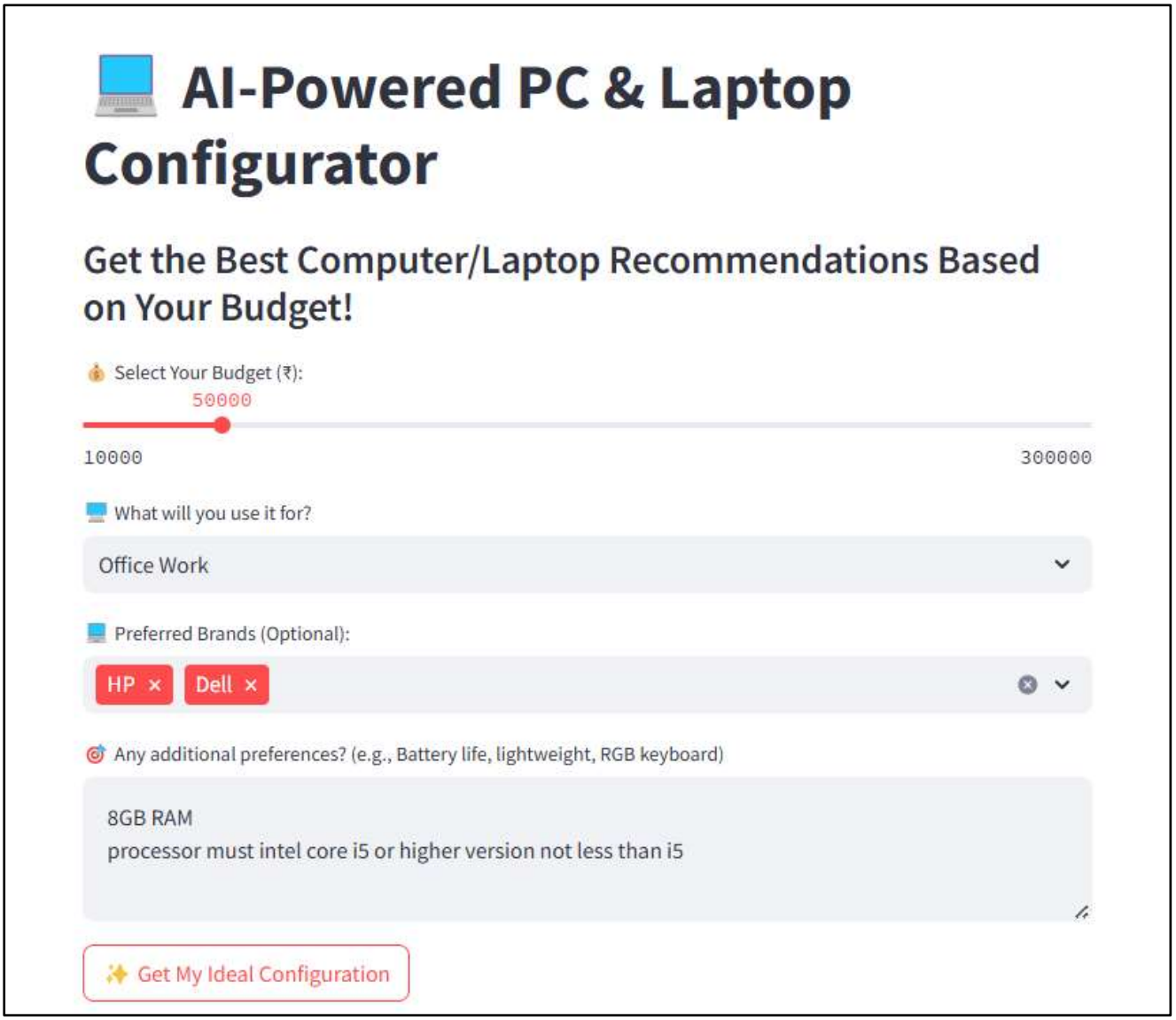
**Question 3:**

Create a helper to suggest computer or laptop configuration based on budget

**app.py**

**CODE:**

```
#Create a helper to suggest computer or laptop configuration based on budget

import streamlit as st
import google.generativeai as genai

# Configure Generative AI (Replace with your API key securely)
#Give your correct GEMINI API KEY
genai.configure(api_key=" AIzaSyBW0FGG-00000000000XXXXXXXXfkscT3lqIcJ0w1")
model = genai.GenerativeModel("gemini-1.5-flash")

# 🎉 App Title
st.title("💻 AI-Powered PC & Laptop Configurator")
st.subheader("Get the Best Computer/Laptop Recommendations Based on Your Budget!")

# User Inputs
budget = st.slider("💰 Select Your Budget (₹):", min_value=10000, max_value=300000,
step=5000, value=50000)
purpose = st.selectbox("🖥️ What will you use it for?", ["Gaming", "Office Work",
"Programming", "Video Editing", "3D Rendering", "Casual Use"])
brand_preference = st.multiselect("💻 Preferred Brands (Optional):", ["Dell", "HP", "Asus",
"Acer", "Apple", "Lenovo", "MSI", "Any"])
extra_requirements = st.text_area("🎯 Any additional preferences? (e.g., Battery life,
lightweight, RGB keyboard)")

# Generate Configuration Suggestion
if st.button("✨ Get My Ideal Configuration"):
    with st.spinner("🔍 Finding the best configuration for you..."):
        prompt = f"""
        Suggest the best laptop or PC configuration under ₹{budget} for {purpose} use.
        - Preferred brands: {', '.join(brand_preference) if brand_preference else "Any"}
        - Additional requirements: {extra_requirements if extra_requirements else "None"}
        Provide CPU, GPU, RAM, storage, display, battery life, and key features.
        """
        try:
            response = model.generate_content(prompt)
            config_suggestion = response.text if hasattr(response, "text") else str(response)
            st.success("💡 Recommended Configuration:")
            st.write(config_suggestion)
        except Exception as e:
            st.error(f"⚠️ Error generating configuration: {e}")

st.info("💡 Adjust your budget and preferences for the best results!")
```


AI-Powered PC & Laptop Configurator
Get the Best Computer/Laptop Recommendations Based on Your Budget!
Select Your Budget (₹):
50000
10000
300000
What will you use it for?
Office Work
Preferred Brands (Optional):
HP
Dell
Any additional preferences? (e.g., Battery life, lightweight, RGB keyboard)
8GB RAM
processor must intel core i5 or higher version not less than i5
Get My Ideal Configuration

💡 Recommended Configuration:

It's challenging to provide a precise configuration within a ₹50,000 budget because prices fluctuate and availability varies by region. However, I can suggest a *target* configuration and some potential models to look for within that budget from HP and Dell in India. You'll need to check current prices and availability at major online retailers.

**Target Configuration:**

- **CPU:** Intel Core i5 11th generation or higher (e.g., i5-1135G7, i5-1235U). Aim for a 12th gen if possible within your budget.
- **GPU:** Integrated Intel Iris Xe Graphics (sufficient for office work). Don't expect gaming capabilities.
- **RAM:** 8GB DDR4 (minimum). Consider upgrading to 16GB if possible, as this will future-proof your system.
- **Storage:** 512GB SSD (Solid State Drive) – crucial for fast boot times and application loading. Avoid HDDs (Hard Disk Drives) for office work.
- **Display:** 14-inch or 15.6-inch FHD (1920x1080) anti-glare display.
- **Battery Life:** Aim for at least 6-8 hours of battery life. This can vary greatly depending on usage.

**Possible Models to Look For (Check Current Prices & Availability):**

- **HP:** Look for HP 14s, HP 15s, or HP Envy models within your price range. Pay close attention to the specifications because they vary significantly within each series.
- **Dell:** Look for Dell Inspiron 14, Dell Inspiron 15, or Dell Vostro models. Similar to HP, there's a wide range of configurations, so check carefully.

**Important Considerations:**

- **Refurbished vs. New:** You might find slightly older models with better specs as refurbished options within your budget. Consider this only from reputable sellers with warranties.
- **Sales and Offers:** Keep an eye out for sales on major online retailers (Amazon, Flipkart) as prices can drop significantly during festive seasons or promotional periods.
- **Specifications:** Don't solely rely on the model name; always check the complete specifications (CPU, RAM, storage, etc.) before purchasing.

**Recommendation:**

Before making a purchase, I strongly advise you to:

1. **Visit major online retailers:** Check the websites of Amazon India and Flipkart to see what's currently available within your budget and compare specifications.
2. **Read reviews:** Check customer reviews on the specific models you're considering to get an idea of their performance and reliability.

By following these steps and focusing on the target configuration, you'll have a much better chance of finding a suitable laptop for office work within your ₹50,000 budget. Remember that the specific model you choose will depend on current availability and pricing.

💡 Adjust your budget and preferences for the best results!

**Question 4:**

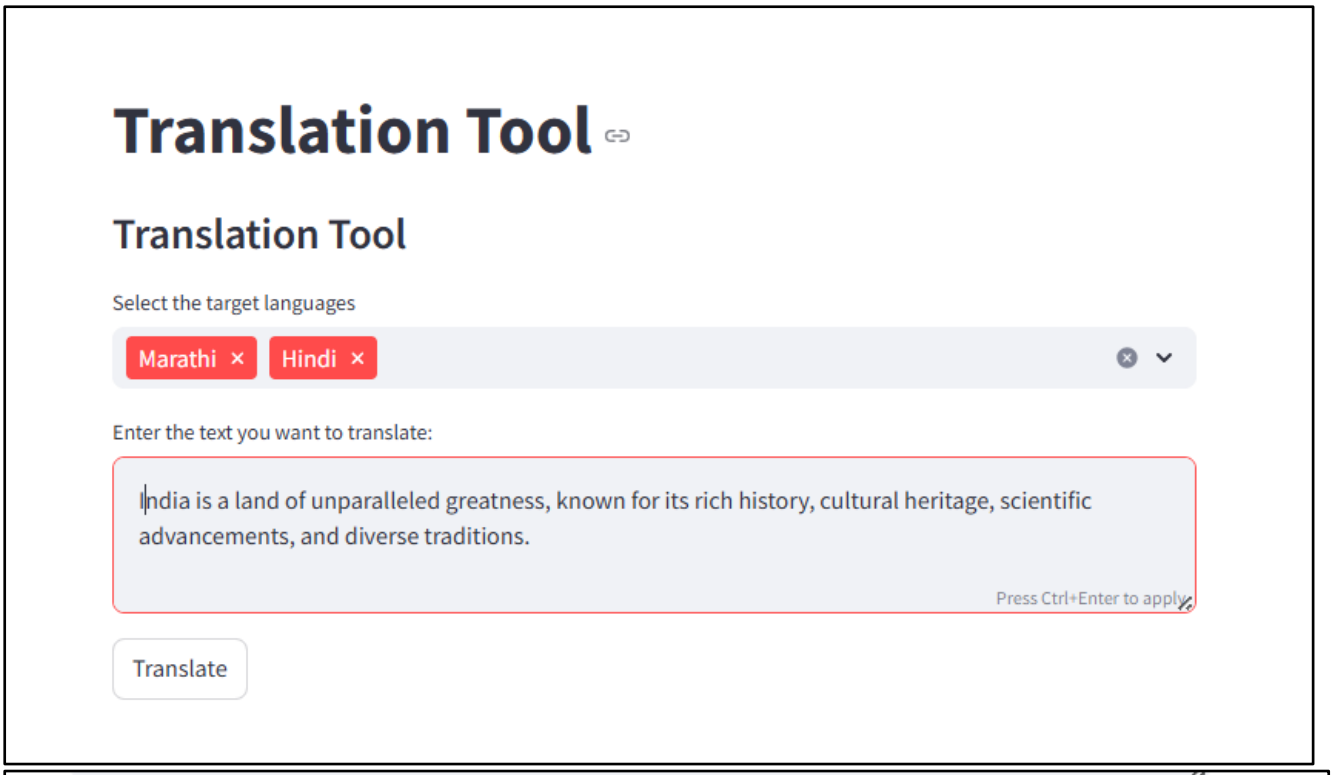
Create a helper to translate from English to Marathi

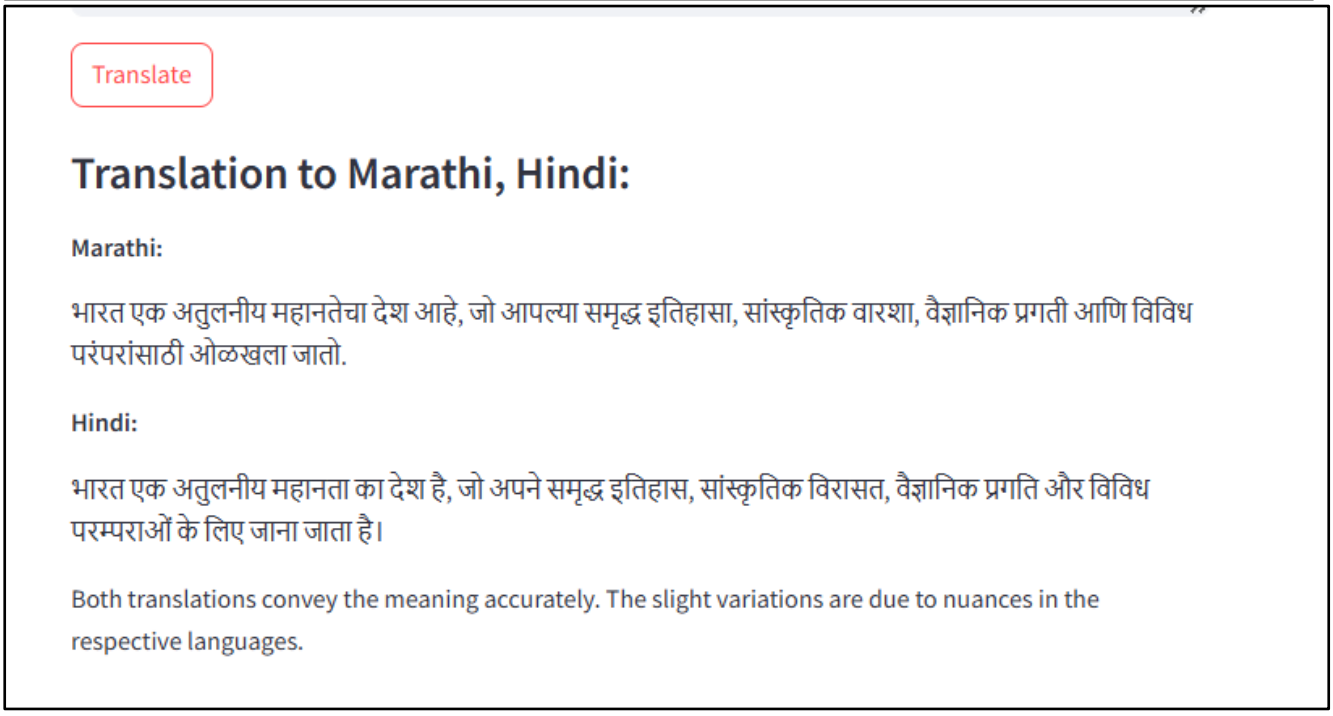
**app1.py**

CODE:

#Create a helper to translate from English to Marathi

```
import streamlit as st
import google.generativeai as genai

# Configure Generative AI (Ensure API key is set securely)
#Give your correct GEMINI API KEY
genai.configure(api_key=" AIzaSyBW0FGG-00000000000XXXXXXXXfkscT3lqIcJ0w1")

# Initialize the model
model = genai.GenerativeModel("gemini-1.5-flash")

# UI Components
st.title("Translation Tool")

# Multi-language selection
languages = st.multiselect("Select the target languages", ["Marathi", "Hindi"])
text = st.text_area("Enter the text you want to translate:")

if st.button("Translate"):
    if not text:
        st.warning("Please enter text to translate.")
    elif not languages:
        st.warning("Please select at least one language.")
    else:
        # Convert selected languages into a string format
        target_languages = ", ".join(languages)

        # Corrected prompt formatting
        prompt = f"Translate the following text from English to {target_languages}: \n\n{text}"

        try:
            response = model.generate_content(prompt)
            translation = response.text if hasattr(response, "text") else str(response)

            st.subheader(f"Translation to {target_languages}:")
            st.write(translation)

        except Exception as e:
            st.error(f"Error in translation: {e}")
```


Translation Tool
Translation Tool
Select the target languages
Marathi ×
Hindi ×
Enter the text you want to translate:
India is a land of unparalleled greatness, known for its rich history, cultural heritage, scientific advancements, and diverse traditions.
Press Ctrl+Enter to apply
Translate


Translate
Translation to Marathi, Hindi:
Marathi:
भारत एक अतुलनीय महानतेचा देश आहे, जो आपल्या समृद्ध इतिहासा, सांस्कृतिक वारशा, वैज्ञानिक प्रगती आणि विविध परंपरांसाठी ओळखला जातो.
Hindi:
भारत एक अतुलनीय महानता का देश है, जो अपने समृद्ध इतिहास, सांस्कृतिक विरासत, वैज्ञानिक प्रगति और विविध परम्पराओं के लिए जाना जाता है।
Both translations convey the meaning accurately. The slight variations are due to nuances in the respective languages.