

TripTrek: Intelligent Travel Planning using Palm's chat-baison-001

Project Description:

TripTrek is an AI-powered travel planning platform designed to revolutionize the way people plan and organize their trips. By leveraging advanced artificial intelligence algorithms, TripTrek offers users personalized travel itineraries tailored to their preferences, interests, and budget constraints. The platform combines machine learning models with rich travel data to provide users with comprehensive recommendations for accommodations, activities, dining options, transportation, and more. With TripTrek, travelers can say goodbye to the hassle of manually researching and organizing every aspect of their trip and instead enjoy a seamless and stress-free travel planning experience.

Scenario 1: Family Vacation Coordination

TripTrek helps families plan their vacations by taking user inputs such as destination and number of days to generate a detailed itinerary. It suggests family-friendly attractions like amusement parks, museums, and scenic spots, and provides recommendations for nearby restaurants and cafes that cater to diverse dietary needs. The output is a day-by-day itinerary that includes timings for visits to attractions, meal breaks at recommended food places, and suggested activities for relaxation and entertainment, ensuring a balanced and enjoyable trip for all family members.

Scenario 2: Business Travel Planning for Professionals

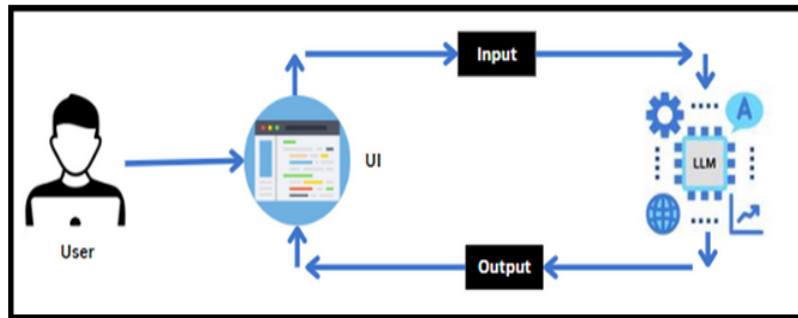
TripTrek streamlines business travel for professionals by taking user inputs like destination and number of days to create a comprehensive itinerary. It recommends key business venues such as conference centers and meeting locations, along with local attractions for downtime. Additionally, it provides suggestions for nearby restaurants and cafes suitable for business lunches and dinners. The output is a detailed day-by-day schedule that includes meeting times, locations, and meal breaks at recommended food places, helping professionals maximize their time and maintain productivity during their trip.

Scenario 3: Educational Trip for Students

TripTrek assists in planning educational trips for students by taking inputs like destination and number of days to produce a structured itinerary. It suggests educational and historical sites, museums, universities, and science centers that align with the trip's educational goals. Furthermore, it provides recommendations for student-friendly dining options, including affordable restaurants and food courts. The output is a day-by-day itinerary that includes timings for visits to educational sites, meal breaks at recommended food places, and leisure

activities, ensuring a balanced and engaging trip for students.

Technical Architecture



Key Features:

- Personalized Destination Recommendations:
 - Utilize user preferences, travel history, and interests to suggest ideal travel destinations.
 - Offer insights into the best times to visit, local attractions, and cultural highlights.
- Itinerary Creation and Management:
 - Generate customized travel itineraries based on user inputs and preferences.
 - Allow users to modify and manage their itineraries effortlessly.
- Accommodation and Transport Booking
 - Integrate with various booking platforms to provide options for flights, hotels, and car rentals.
 - Offer real-time availability and pricing information to make informed decisions.
- Activity and Experience Suggestions:
 - Recommend local activities, tours, and experiences tailored to user interests.
 - Provide booking options and user reviews for each suggestion.

Project Goals:

- Develop a user-friendly interface that enhances the travel planning experience.
- Achieve high accuracy in personalized recommendations using advanced AI algorithms.
- Ensure seamless integration with major travel service providers for comprehensive booking options.
- Provide exceptional customer support through real-time assistance features

Project Flow:

- Users input text into the UI of Inquisitive.
- The input text is then processed and analyzed by the PALM architecture, which is integrated into the backend.
- PALM autonomously generates questions based on the input text.
- The generated questions are sent back to the frontend for display on the UI.
- Users can view the dynamically generated questions and interact with them to gain deeper insights into the content.

To accomplish this, we have to complete all the activities listed below,

- Initializing the PALM
 - Generate PALM API
 - Initialize the pre-trained model
- Interfacing with Pre-trained Model
 - Questions Generator
- Model Deployment
 - Deploy the application using Streamlit

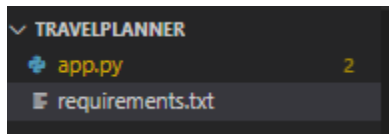
Prior Knowledge

You must have prior knowledge of the following topics to complete this project.

- LLM & PALM: <https://cloud.google.com/vertex-ai/docs/generative-ai/learn-resources>
- Streamlit: <https://www.datacamp.com/tutorial/streamlit>

Project Structure

Create the Project folder which contains files as shown below:



- app.py: It serves as the primary application file housing both the model and Streamlit UI code.
- requirements.txt: It enumerates the libraries necessary for installation to ensure proper functioning.

Milestone 1: Requirements Specification

Specifying the required libraries in the requirements.txt file ensures seamless setup and reproducibility of the project environment, making it easier for others to replicate the development environment.

Activity 1: Create A Requirements.Txt File To List The Required Libraries.

```
#libraries to be installed
streamlit==1.10.0
google-generativeai
```

- streamlit: Streamlit is a powerful framework for building interactive web applications with Python.

google-generativeai: Python client library for accessing the GenerativeAI API, facilitating interactions with pre-trained language models like chat-bison-001 .

Activity 2: Install The Required Libraries

```
(base) C:\Users\Navyaa\Desktop\SCIT\DSDA\SmartBridge\Projects\Week 7 Projects\TravelPlanner>pip install -r requirements.txt
```

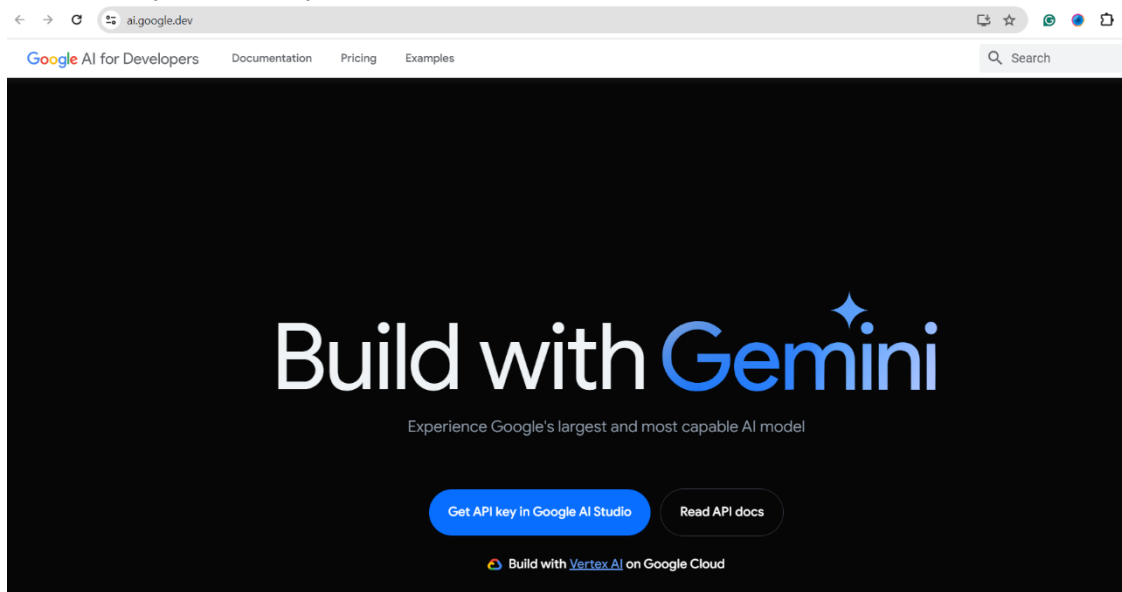
- Open the terminal.
- Run the command: pip install -r requirements.txt
- This command installs all the libraries listed in the requirements.txt file

Milestone 2: Initialization The Model

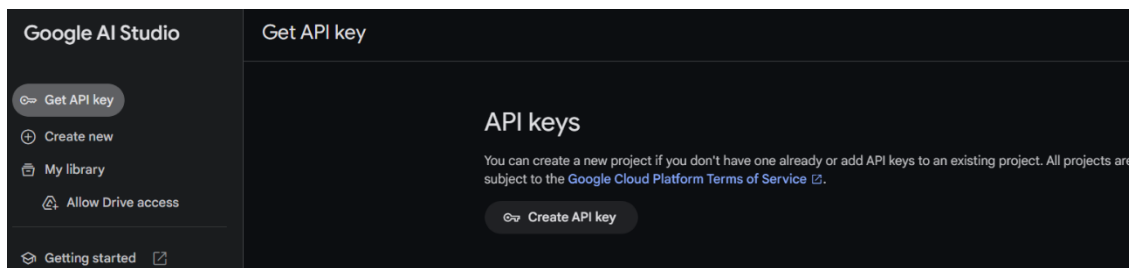
The Google API key is a secure access token provided by Google, enabling developers to authenticate and interact with various Google APIs. It acts as a form of identification, allowing users to access specific Google services and resources. This key plays a crucial role in authorizing and securing API requests, ensuring that only authorized users can access and utilize Google's services. For initializing the model we need to generate PALM API.

Activity 1:Generate PALM API Key

- Click on the link (<https://developers.generativeai.google/>).
- Then click on "Get API key in Google AI Studio".
- Click on "Get API key" from the right navigation menu.
- Now click on "Create API key". (Refer the below images)
- Copy the API key.



After signing in to your account, navigate to the 'Get an API Key' option. Clicking on this option will redirect you to another webpage as shown below.



Next, click on 'Create API Key' and choose the generative language client as the project. Then, select 'Create API key in existing project'.

Copy the newly generated API key as it is required for loading the pre-trained model.

Activity 2: Initialize The Pre-Trained Model

Import necessary files

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

- Streamlit, a popular Python library, is imported as st, enabling the creation of user interfaces directly within the Python script.
- Importing the palm module: This line imports the palm module from the google.generativeai package.

Configuration of the PALM API with the API key and initialize translator

```
# Configure the API with your API key
palm.configure(api_key="AIzaSyBv_2Br0SYxHDxuC-u7FM4fKCwqXXXXXXXXXXXX")
```

- Configuring the API key: The configure function is used to set up or configure the Google API with an API key. The provided API key, in this case, is " AIzaSyBv_2Br0SYxHDxuC-u7FM4fKCwqXXXXXXXXXXXX".
- The Translator class facilitates language translation capabilities within the application.

Define the model to be used

```
# Define the model to use
model_name = "models/chat-bison-001"
```

- The line model_name = "models/chat-bison-001" sets the variable model_name to the string "models/chat-bison-001", which identifies a specific model provided by Google's PaLM (Pathways Language Model) API.
- This variable is used in subsequent API calls to specify which model to use for

generating responses based on user prompts.

Milestone 3: Interfacing With Pre-Trained Model

In this milestone, we will build a prompt template to generate code based on user description and language.

Activity 1: Generate Itinerary

```
# Button to Generate Itinerary
if st.button("Generate Itinerary"):
    # Placeholder for the generated itinerary
    itinerary = ""
    food_places = ""

    # Generate Itinerary using the selected model
    try:
        with st.spinner("Generating Itinerary..."):
            # Generate text using the model
            prompt = f"Generate an itinerary for a {num_days}-day trip to {destination}. Include details about nearby food places."
            response = palm.generate_text(model=model_name, prompt=prompt)
            itinerary = response.result # Adjust this based on the actual response structure
    except Exception as e:
        # Display detailed error message
        st.error(f"Error generating itinerary: {e}")
        st.exception(e)
        st.warning("Please check your inputs and try again.")

    # Display the generated itinerary and food places
    if itinerary:
        st.success("Itinerary generated successfully!")
        st.text_area("Generated Itinerary:", value=itinerary, height=400)
    else:
        st.warning("No itinerary generated. Please try again with different inputs.")
```

- When the button is pressed, it initializes placeholders for the itinerary and nearby food places. It then constructs a prompt incorporating the user-provided destination and number of days.
- The prompt is sent to the AI model (specified by model_name) to generate the itinerary. During this process, a spinner is displayed to indicate that the system is working.
- If the generation is successful, the resulting itinerary is displayed in a text area.
- If an error occurs, detailed error messages are shown, and users are prompted to check their inputs and try again.
- This approach ensures a user-friendly experience, guiding users through generating and viewing their custom travel plans.

Milestone 4: Model Deployment

In this milestone, we are deploying the created model using streamlit. Model deployment using Streamlit involves creating a user-friendly web interface for deploying machine learning models, enabling users to interact with the model through a browser. Streamlit provides easy-to-use tools for developing and deploying data-driven applications, allowing for seamless integration of machine learning models into web-based applications.

Activity 1: Give The Project Title, Description And Describe The Scenarios

```
# Introduction text
st.markdown("""
# TripTrek: Intelligent Travel Planning with AI

TripTrek is an AI-powered travel planning platform designed to revolutionize the way people plan and organize their trips. By leveraging advanced ar

## Scenario 1: Family Vacation Coordination
TripTrek helps families plan their vacations by taking user inputs such as destination and number of days to generate a detailed itinerary. It sugge

## Scenario 2: Business Travel Planning for Professionals
TripTrek streamlines business travel for professionals by taking user inputs like destination and number of days to create a comprehensive itinerary

## Scenario 3: Educational Trip for Students
TripTrek assists in planning educational trips for students by taking inputs like destination and number of days to produce a structured itinerary.
""")

# Streamlit App Title
st.title("AI Travel Planner Itinerary")
```

- The provided code introduces TripTrek, an AI-driven travel planning platform aimed at revolutionizing trip organization. It leverages advanced AI algorithms to offer personalized itineraries
- Three scenarios illustrate TripTrek's capabilities: Family Vacation Coordination, Business Travel Planning for Professionals, and Educational Trips for Students, showcasing its ability to cater to diverse travel needs with tailored itineraries and recommendations.

Activity 2: Take User Inputs

```
# Streamlit App Title
st.title("AI Travel Planner Itinerary")

# User Input for Travel Destination
destination = st.text_input("Enter your travel destination:")

# User Input for Number of Days
num_days = st.number_input("Enter the number of days for your trip:", min_value=1, max_value=30, step=1)
```

- In this section of the Streamlit application, users are prompted to input their travel destination and the number of days for their trip.
- The `st.text_input` function is used to capture the destination as a text input from the user, allowing them to specify the location they plan to visit.
- The `st.number_input` function is then employed to obtain the number of days for the trip, with a minimum value of 1 day and a maximum of 30 days, allowing users to select the duration of their stay using a convenient stepper interface.
- These inputs are essential for generating a personalized travel itinerary, as they provide the necessary parameters for the AI model to tailor recommendations for activities, accommodations, dining options, and other travel-related details specific to the user's chosen

destination and trip length.

Activity 3: Run The Web Application

- Open the anaconda prompt from the start menu
- Navigate to the folder where your Python script is.
- Now type “streamlit run app.py” command
- Navigate to the localhost where you can view your web page

```
(base) C:\Users\Navyaa\Desktop\SCIT\DSDA\SmartBridge\Projects\Week 7 Projects\CodeGenerator>streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.12:8501
```

Now, the application will open in the web browser.





Input 1:

AI Travel Planner Itinerary

Enter your Travel Destination:

Chennai

Which best describes the purpose of the trip ?

Family

Enter the number of days for the trip:

1

Generate Itinerary

Itinerary generated successfully!

Output 1:

Generated Itinerary:

****Day 1 Itinerary for a Family Trip to Chennai****

****Morning:****

- * 9:00 AM - Arrive at Chennai Airport
- * 10:00 AM - Take a taxi to your hotel
- * 11:00 AM - Check in to your hotel and leave your luggage
- * 11:30 AM - Visit the Marina Beach
- * 1:00 PM - Have lunch at a nearby restaurant (such as Saravana Bhavan or Hotel Ashok Leyland)
- * 2:00 PM - Visit the Kapaleeshwar Temple
- * 3:00 PM - Take a break at a nearby cafe (such as Theobroma or Starbucks)
- * 4:00 PM - Visit the Chennai Museum
- * 6:00 PM - Have dinner at a nearby restaurant (such as Bukhara or The Park)
- * 7:00 PM - Go back to your hotel and relax

****Evening:****

- * 9:00 PM - Watch a cultural show at the Kalakshetra Foundation
- * 11:00 PM - Go back to your hotel and sleep

****Nearby Food Places:****

* Saravana Bhavan: This restaurant serves a variety of South Indian dishes, including dosas, idlis, and vada. It is located near the Marina Beach.

Input 2:

AI Travel Planner Itinerary

Enter your Travel Destination:

Chennai

Which best describes the purpose of the trip ?

Business

Enter the number of days for the trip:

1

Generate Itinerary

Itinerary generated successfully!

Generated Itinerary:

Output 2:

Generate Itinerary

Itinerary generated successfully!

Generated Itinerary:

****Day 1:****

- * 9:00 AM - Arrive at Chennai International Airport
- * 10:00 AM - Check in to hotel (The Park Chennai)
- * 11:00 AM - Visit the Marina Beach
- * 12:00 PM - Lunch at Saravana Bhavan (South Indian cuisine)
- * 1:00 PM - Visit the Kapaleeshwarar Temple
- * 2:00 PM - Relax at your hotel
- * 5:00 PM - Meet with your business partners at the Taj Coromandel Hotel
- * 7:00 PM - Dinner at The Chennai Club (North Indian cuisine)

****Nearby food places:****

- * Saravana Bhavan: This South Indian restaurant is a popular spot for both locals and tourists. The menu features a wide variety of dishes, including dosas, idlis, vadas, and more.
- * The Chennai Club: This fine-dining restaurant offers a modern take on North Indian cuisine. The menu features dishes like lamb biryani, tandoori chicken, and butter chicken.
- * The Park Chennai: This hotel's restaurant offers a variety of international dishes, including Italian, Chinese, and Japanese cuisine.

****Additional tips:****

- * If you're looking for a quick and affordable meal, there are plenty of street food vendors around the

Input 3:

AI Travel Planner Itinerary

Enter your Travel Destination:

Chennai

Which best describes the purpose of the trip ?

Educational

Enter the number of days for the trip:

1

Generate Itinerary

Itinerary generated successfully!

Output 3:

Itinerary generated successfully:

Generated Itinerary:

****Day 1 Itinerary for Chennai****

****9:00 AM**** - Arrive at Chennai Airport and take the metro to your hotel. Check in and leave your luggage.

****9:30 AM**** - Visit the Madras High Court, a UNESCO World Heritage Site. Learn about the history of the court and its architecture.

****11:00 AM**** - Have lunch at a nearby restaurant, such as Saravana Bhavan or Annalakshmi.

****12:00 PM**** - Visit the Government Museum, one of the largest museums in India. See exhibits on art, history, and culture.

****2:00 PM**** - Take a break and relax at the Marina Beach. Watch the sunset over the Bay of Bengal.

****4:00 PM**** - Visit the Kapaleeshwarar Temple, a Hindu temple dedicated to Shiva. Learn about the history and significance of the temple.

****6:00 PM**** - Have dinner at a nearby restaurant, such as Woodlands or Buhari.

****8:00 PM**** - Go for a walk around the city center and explore the nightlife.

****Food Places:****

* Saravana Bhavan: South Indian vegetarian restaurant.

* Annalakshmi: North Indian vegetarian restaurant.

* Woodlands: Chinese restaurant.

* Buhari: Continental restaurant.