2026 Volume: 02 Issue: 05 www.ijrpa.com ISSN 2456-9995 Review Article

International Journal Research Publication Analysis

Page: 01-22

AGENTIC TRAVEL PLANNER

***Hariharan P. R.**

Bachelor of Technology In Artificial Intelligence and machine learning Sri shakthi institute of Engineering and Technology.

Article Received: 24 March 2026
Article Revised: 14 April 2026
Published on: 04 May 2026

***Corresponding Author: Hariharan P. R.**
Bachelor of Technology In Artificial intellegence and machine learning Sri shakthi institute of Engineering and Technology.
DOI: https://doi-doi.org/101555/ijrpa.9887

ABSTRACT

In the rapidly evolving landscape of modern travel, travelers often struggle to consolidate destination research, real-time weather conditions, local attractions, transportation options, and budget planning into a coherent and actionable itinerary. Existing travel platforms p rimarily offer fragmented, static information without intelligent orchestration, personalized recommendations, or dynamic cost awareness, resulting in time-consuming manual planning and suboptimal travel experiences. To address these challenges, the proposed Agentic Travel Planner introduces an intelligent Agentic AI-based travel planning framework that integrates Large Language Models (LLMs), a suite of real-world data tools. Unlike traditional travel applications, the system operates as an autonomous planning agent capable of interpreting natural-language travel requests, dynamically invoking specialized tools for weather lookup, attraction and restaurant discovery, transportation guidance, currency conversion, and expense estimation, and synthesizing the gathered information into structured, markdown-formatted travel itineraries. receive comprehensive trip plans, and observe the agent's step-by-step reasoning trace. By continuously reasoning over retrieved data and adapting responses to user-specified parameters such as duration, group size, and budget, the agentic system reduces planning overhead, improves itinerary quality, and enables scalable, personalized travel assistance—transforming conventional travel search into an intelligent, self-guided trip planning experience.

CHAPTER 1 INTRODUCTION

In the rapidly evolving digital era, travel has undergone a significant transformation with the widespread availability of online booking platforms, location-based services, and AI-powered recommendation engines. Travelers today have access to vast amounts of information spanning destinations, accommodations, local attractions, weather conditions, transportation options, and budgeting tools. However, effectively consolidating and utilizing these resources to produce a coherent, personalized travel itinerary remains a major challenge. Travelers often struggle to coordinate multiple aspects of trip planning simultaneously, balance budget constraints with desired experiences, and make informed decisions in the absence of a unified, intelligent planning system [1], [7], [15].

One of the most critical challenges in modern travel planning is the absence of a systematic and adaptive approach to itinerary generation. Unlike guided tour packages where schedules are predefined and curated by professionals, independent travelers rely on scattered resources such as review websites, weather apps, currency converters, and map services. This results in fragmented planning, where travelers gather information from multiple disconnected sources without a clear sequence, often leading to confusion, inefficiency, and suboptimal travel experiences [8], [15]. The difficulty increases when planning trips across unfamiliar destinations, as there is no centralized intelligent system to retrieve live data, estimate costs, or synthesize recommendations into actionable plans.

In conventional travel platforms, content delivery is largely static and does not adapt dynamically to a traveler's specific preferences, group size, budget, or duration. Most systems provide generic destination guides that lack personalization, real-time data integration, and end-to-end planning capabilities. As a result, travelers are unable to receive tailored itineraries, accurate cost breakdowns, or live weather-aware recommendations. These limitations lead to increased planning overhead, poor resource utilization, and diminished travel experiences [2], [7], [8].

Recent advancements in artificial intelligence have introduced new possibilities for enhancing digital travel assistance. Large Language Models (LLMs) have demonstrated the ability to generate coherent, human-like travel plans and contextual recommendations [3], [4]. Similarly, tool-augmented AI frameworks have been developed to improve response accuracy by integrating external real-world data sources such as weather APIs, location services, and search engines into the generation process [5], [6]. Despite these advancements, most existing systems function as passive question-answering tools and lack the capability to autonomously orchestrate multiple data sources, reason over retrieved information, and produce structured, end-to-end travel plans.

Furthermore, modern travel planning demands intelligent systems capable of handling dynamic user inputs, retrieving real-time information, performing multi-step reasoning, and synthesizing diverse data into a unified response. The increasing complexity of travel requirements—spanning weather forecasting, local attraction discovery, currency conversion, and expense estimation—requires systems that can plan, act, and adapt based on contextual information. However, current platforms fail to integrate these capabilities into a cohesive, autonomous, and conversational planning framework [11], [12].

To address these limitations, this research proposes the Agentic Travel Planner—an Agentic AI-based intelligent travel planning system designed to provide a unified, real-time, and personalized travel planning experience. The system introduces an autonomous planning agent built on LangGraph that interprets natural-language travel requests, dynamically invokes specialized tools for weather lookup, attraction and restaurant discovery, transportation guidance, currency conversion, and budget calculation, and synthesizes all retrieved information into a structured, markdown-formatted travel itinerary. By integrating LLMs with a multi-tool orchestration pipeline and live external APIs, the system enables seamless, end-to-end trip planning within a single conversational interface.

This report is organized as follows: Chapter 2 presents a comprehensive literature review of existing travel recommendation systems and AI-based planning technologies. Chapter 3 discusses the proposed system, including its architecture and functional components. Chapter 4 describes the system design, including architecture diagrams, workflow models, and tool integration structure. Chapter 5 details the implementation and technologies used. Chapter 6 presents the results and system evaluation. Finally, Chapter 7 concludes the study with key findings and future enhancements.

CHAPTER 2 LITERATURE REVIEW

This chapter presents a comprehensive review of existing studies related to intelligent travel recommendation systems, AI-driven itinerary generation, and agentic planning frameworks. It highlights the major challenges in modern digital travel planning, summarizes key research contributions, and identifies the limitations that still persist in the domain. Each subsection focuses on a specific problem area and the corresponding approaches proposed by researchers. The discussion aims to identify research gaps that led to the development of the proposed **Agentic Travel Planner system**.

2.1 INEFFICIENT TRAVEL ITINERARY GENERATION

Generating structured and coherent travel itineraries remains a significant challenge in existing travel platforms. Buhalis and Law [1] highlighted the evolution of travel technology and the growing demand for intelligent, personalized trip-planning solutions, yet noted that most platforms still rely on static templates. Lu et al. [2] proposed automated itinerary generation models using optimization algorithms, improving scheduling efficiency but failing to account for real-time constraints such as weather and local availability. Vansteenwegen et al. [8] formalized the Orienteering Problem as a basis for itinerary planning, demonstrating improvements in route optimization while lacking natural language accessibility and dynamic adaptation.

2.2 PERSONALIZATION IN TRAVEL RECOMMENDATIONS

Personalized travel recommendations are essential for enhancing user satisfaction and trip quality. Ricci et al. [7] extensively reviewed recommender systems in tourism, demonstrating that collaborative filtering and content-based approaches can improve destination suggestions but are limited by cold-start problems and sparse user data. Zhang et al. [15] explored deep learning-based personalization for travel, achieving improved recommendation accuracy but requiring large-scale historical datasets. yet these approaches lack real-time adaptability to user-specified trip parameters such as budget, group size, and duration.

2.3 AI-BASED CONTENT GENERATION AND RETRIEVAL

Recent advancements in artificial intelligence have significantly improved automated content generation and data retrieval for travel applications. Brown et al. [3] demonstrated the capabilities of Large Language Models (LLMs) in generating coherent, contextually relevant text, enabling conversational travel assistance. OpenAI [4] extended these capabilities through instruction-tuned generative models that support complex multi-step reasoning and structured output generation. Schick et al. [5] introduced tool-augmented language models capable of invoking external APIs to retrieve real-time information, substantially improving response accuracy. Nakano et al. [6] further demonstrated web-augmented generation, enabling systems to access live data sources during inference.

2.4 REAL-TIME DATA INTEGRATION IN TRAVEL SYSTEMS

Modern travel planning demands real-time integration of dynamic data including weather forecasts, point-of-interest availability, currency exchange rates, and transportation schedules. Gavalas et al. [10] reviewed mobile tourism recommender systems and emphasized the importance of context-aware, real-time data in delivering relevant travel suggestions, while noting the difficulty of integrating heterogeneous data sources. Pan and Fesenmaier [11] explored online information search behavior in travel planning, revealing that travelers consult numerous independent services before finalizing decisions. Russell and Norvig [12] highlighted the importance of intelligent agents capable of perceiving dynamic environments and taking actions based on real-time observations.

2.5 AGENTIC AI AND AUTONOMOUS PLANNING FRAMEWORKS

The emergence of Agentic AI has introduced systems capable of autonomously reasoning, planning, and acting across multi-step tasks. Yao et al. [13] proposed ReAct, a framework enabling LLMs to interleave reasoning traces with tool-use actions, demonstrating significant improvements in task completion for complex, multi-step problems. Chase [14] introduced LangChain, a framework for building agent-based pipelines that chain LLM calls with external tool invocations, enabling modular and extensible AI applications. Despite these developments, the application of agentic frameworks specifically to end-to-end travel planning remains limited. Most existing travel systems do not leverage autonomous agent architectures to coordinate weather retrieval, attraction discovery, cost estimation, and itinerary synthesis within a unified, conversational workflow.

TABLE 2.1 SUMMARY OF EXISTING METHODS.

Paper & Author	Methodology	Limitations
Buhalis & Law [1]	Travel technology review	No real-time dynamic generation
Lu et al. [2]	Optimization-based itinerary generation	Lacks natural language interface
Brown et al. [3]	Large Language Models (LLMs)	No multi-tool orchestration
Schick et al. [5]	Tool-augmented language models	Single-tool, not multi-step
Ricci et al. [7]	Collaborative filtering recommenders	Cold-start and data sparsity issues
Vansteenwegen et al.[8]	Orienteering problem for routing	No real-time adaptability
Gavalas et al. [10]	Context-aware mobile tourism systems	Heterogeneous data integration issues
Russell & Norvig [12]	Intelligent agent frameworks	Not domain-specific to travel
Yao et al. [13]	ReAct agentic reasoning	Limited travel-domain

	framework	application

Although significant research has been conducted in travel recommendation systems, AI-based content generation, and agentic planning frameworks, most existing solutions suffer from limitations such as lack of real-time multi-API orchestration, insufficient end-to-end itinerary synthesis, and absence of unified autonomous decision-making pipelines for travel planning. To overcome these challenges, the proposed Agentic Travel Planner integrates LLMs, a LangGraph-based multi-tool agent, and live external APIs to provide structured, real-time, and personalized travel planning within a single conversational interface.

CHAPTER 3
PROPOSED SYSTEM

3.1 OVERVIEW

Planning a travel itinerary in modern digital environments requires real-time data integration, adaptive decision-making, and personalized recommendations. Travelers often struggle to consolidate weather information, local attractions, accommodation, transportation, and budgeting across multiple platforms. These systems fail to incorporate real-time contextual reasoning, autonomous decision-making, and conversational interaction, resulting in inefficient trip planning and poor user satisfaction. To address these challenges, the proposed **Agentic Travel Planner** is introduced to enable intelligent, automated AI-driven interface. The system aims to transform traditional travel planning into a dynamic, agent-orchestrated experience capable of generating complete, personalized itineraries from a single natural language request.

3.2 PROPOSED SYSTEM

The Agentic Travel Planner is an AI-based intelligent travel planning framework designed to generate structured day-by-day itineraries, provide destination-specific recommendations, and adapt dynamically based on user preferences and real-time data. It integrates Large Language Models (LLMs), a LangGraph-based multi-tool agent, and live external APIs to deliver personalized and context-aware travel plans through a interface.

The Agentic Travel Planner incorporates multiple specialized tool integrations including a weather retrieval tool (**OpenWeatherMap API**), an attractions discovery tool (**Google Places API**), a currency conversion tool, and a cost estimation tool. These tools are orchestrated by the **LangGraph agent**, ensuring that all retrieved data is current, relevant, and integrated seamlessly into the final itinerary.

A key component of the system is its **Agentic AI capability**, where the system operates as an autonomous planning agent capable of reasoning about user requirements, selecting appropriate tools, executing multi-step retrieval workflows, and synthesizing outputs into a structured travel plan. This enables the system to handle complex, multi-faceted travel queries within a single conversational session. Additionally, the platform includes an interactive UI interface that provides real-time itinerary generation, allowing users to request plans, view weather summaries, explore attractions, and receive cost estimates in a conversational format.

3.3 ADDRESSING LIMITATIONS OF EXISTING SYSTEMS

The proposed **Agentic Travel Planner** is designed to overcome the limitations of existing digital travel platforms. Traditional systems rely on static content delivery, disconnected services, and manual aggregation of information, leading to inefficient and fragmented planning experiences. The proposed model introduces intelligence, automation, and real-time multi-API orchestration to enhance planning efficiency and user satisfaction.

3.3.1 FRAGMENTED TRAVEL INFORMATION

Existing platforms require users to visit multiple websites for weather, attractions, accommodation, and currency information. The proposed system consolidates all these data sources into a single agent-orchestrated workflow, delivering a unified travel plan from one natural language query.

3.3.2 LACK OF PERSONALIZATION

Traditional travel platforms offer generic recommendations without considering user-specific preferences such as trip duration, budget, travel style, or group size. The proposed system uses LLM-driven reasoning to tailor itineraries, daily schedules, and activity recommendations to each user's stated preferences.

3.3.3 FRAGMENTED LEARNING RESOURCES

Current learning approaches rely on scattered resources such as videos, blogs, and documentation, making it difficult to maintain consistency. The proposed system integrates multiple data sources into a unified platform using RAG, providing structured and context-aware content within a single learning environment.

3.3.4 ABSENCE OF REAL-TIME DATA INTEGRATION

Current platforms largely depend on pre-cached or static data that may be outdated. The proposed system performs live API calls during each planning session, ensuring that weather forecasts, attraction availability, and currency rates reflect current conditions.

3.4 PROCESS FLOW

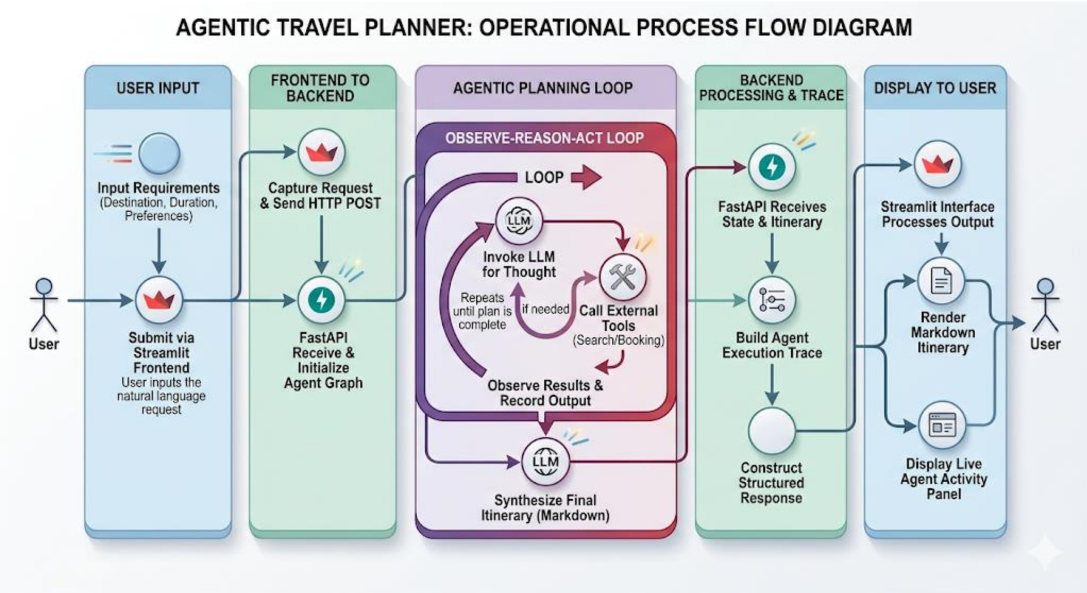

Figure 3.4 Process flow

The process flow of the Agentic Travel Planner begins with the user submitting a natural language travel request through the chat interface, specifying destination, travel dates, budget, and preferences. This input is received by the LangGraph agent, which performs intent parsing using the underlying LLM to identify required planning sub-tasks. The agent then autonomously invokes the appropriate tools in sequence: the weather tool retrieves live forecasts for the destination, the attractions tool fetches nearby points of interest and activities, the currency tool performs exchange rate conversion, and the cost estimation tool calculates an approximate budget breakdown. All retrieved data is passed back to the LLM, which synthesizes the information into a structured, day-by-day itinerary. The final plan is delivered through the formatted conversational response, allowing users to refine requests or ask follow-up questions.

3.5 USE CASE DIAGRAM

The System Use Case Diagram describes how different users interact with the Agentic Travel Planner to plan trips, retrieve travel information, and generate personalized itineraries using AI and live APIs. It involves two main actors — the Traveler/User and the AI Travel Agent — each performing specific roles such as submitting travel requests and autonomous multi-tool orchestration. The diagram highlights key functions including natural language query input, destination selection, weather retrieval, attractions discovery, currency conversion, cost estimation, itinerary generation, and conversational follow-up interactions. It also shows the interaction between the Chat Interface (user-facing layer) and the LangGraph Agent Engine (orchestration layer), which together enable efficient, real-time, and personalized travel planning. The interface allows users to submit queries and view structured itineraries, while the agent engine continuously invokes tools, synthesizes data, and refines responses to deliver complete and accurate travel plans.

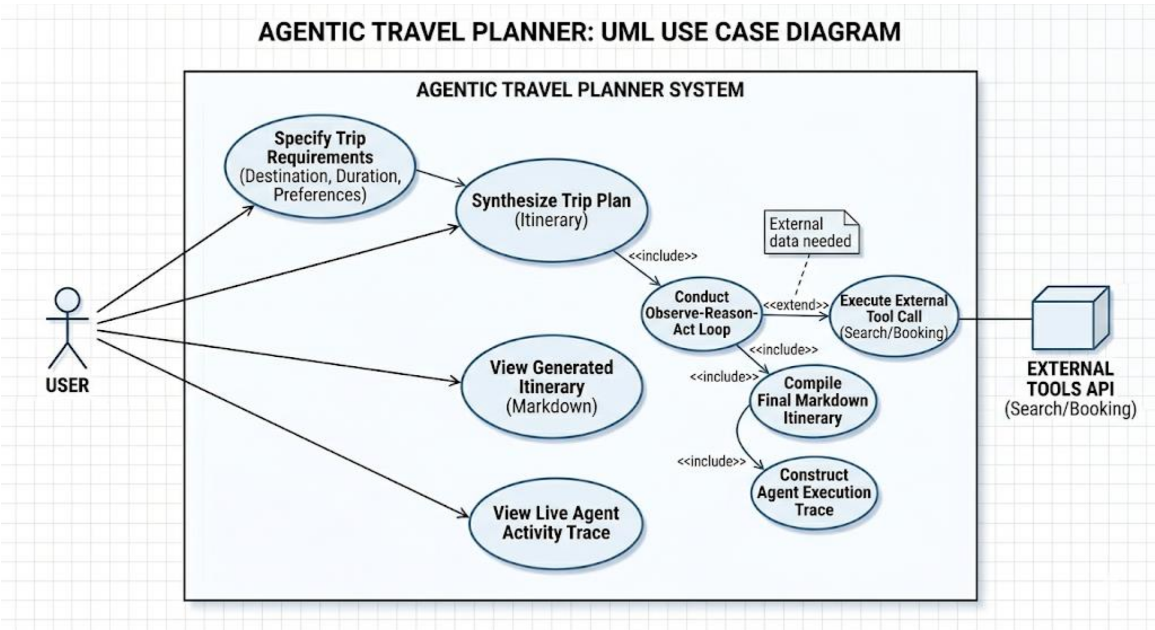

Figure 3.5 Use Case Diagram.

CHAPTER 4

SYSTEM DESIGN

This chapter presents a detailed analysis and design of the **Agentic Travel Planner – an Agentic AI-Based Intelligent Travel Planning System**. It focuses on transforming the system's conceptual framework into a structured and implementable design. The chapter includes the System Architecture, Sequence Diagram, and Functionalities of the System.

Together, these components ensure that the proposed system is efficient, scalable, and capable of addressing the limitations identified in existing travel planning platforms.

4.1 SYSTEM ARCHITECTURE

The Agentic Travel Planner system architecture integrates agentic intelligence, real-time multi-API data retrieval, autonomous tool orchestration, and user-centric visualization to ensure efficient and personalized travel planning experiences. It is organized into five major components — the Data Layer, the User Interface, the AI/ML Engine (Agentic Core), the Tool Execution Layer, and the Backend API Layer. Together, these modules create a closed-loop intelligent travel planning system that enhances personalization, improves itinerary quality, and enables adaptive decision-making.

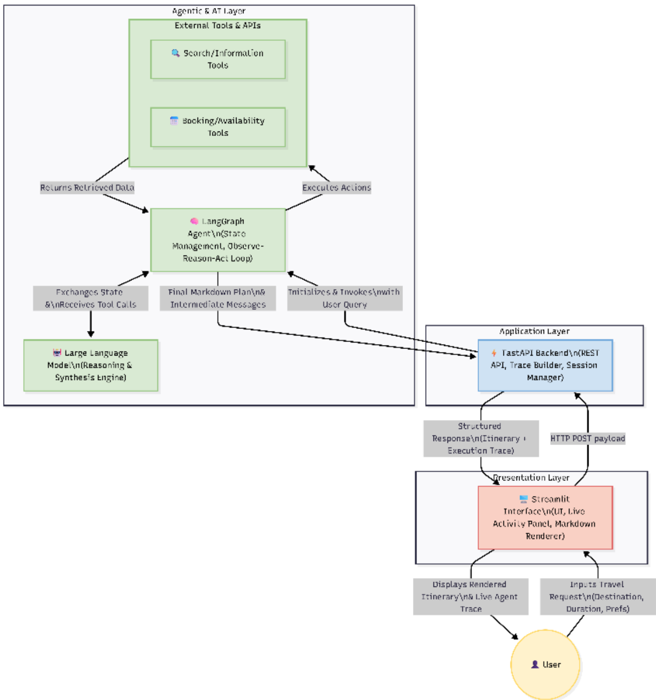

Figure 4.1 System Architecture.

4.1.1 Data Layer

The Data Layer forms the foundation of the Agentic Travel Planner system, supplying real-time inputs to the AI engine and tool modules. It aggregates data from external live sources, ensuring that planning decisions are based on accurate and current information. The inputs include:

- **Weather Data**: Live weather conditions and multi-day forecasts fetched from the OpenWeatherMap API

- **Place and Attraction Data**: Tourist attractions, restaurants, activities, and transportation options fetched from the Google Places API and Tavily Search API
- **Currency Data**: Live exchange rates fetched from the Exchange Rate API for cost conversion
- **Configuration Data**: LLM model settings and API configurations managed through YAML-based config files
- **User Query Data**: Natural language travel requests submitted through the Streamlit chat interface

This structured data layer enables real-time updates, personalized travel recommendations, and accurate cost calculations.

4.1.2 User Interface Layer

This module acts as the interactive layer between the user and the system. It is built using UI and provides a conversational, form-based interface for submitting travel requests and viewing generated itineraries.

- Users can input natural language queries such as destination, trip duration, and preferences
- The interface displays structured, Markdown-formatted day-by-day travel plans
- A live Agent Activity panel renders the step-by-step execution trace, showing which tools were invoked and what data was retrieved
- A progress indicator provides real-time feedback while the agent processes the request

This layer ensures seamless interaction, usability, and transparency of the AI planning process.

4.1.3 AI/ML Engine (Agentic Core)

The AI/ML Engine is the core intelligence of the system. It is implemented using LangGraph and operates as a ReAct (Reasoning and Acting) Agentic AI capable of understanding user intent, selecting appropriate tools, executing multi-step workflows, and synthesizing outputs into a comprehensive travel plan. The engine performs the following functions

- **Intent Understanding**: The LLM analyzes user input and extracts destination, duration, budget, and preferences
- **Agentic Planning**: The LangGraph StateGraph orchestrates the agent loop, deciding

which tools to call and in what order

- **Tool Binding**: The LLM is bound with all available tools (weather, places, currency, calculator) and selects them conditionally based on the planning context
- **Response Synthesis**: The LLM aggregates all tool outputs and generates a structured, Markdown-formatted itinerary
- **Multi-Provider LLM Support**: The system supports both Groq (LLaMA-based) and OpenAI models, configured through the ModelLoader module

This module ensures intelligent, personalized, and dynamic travel plan generation.

4.1.4 Tool Execution Layer

Tool is implemented as a LangChain tool and registered with the LangGraph ToolNode for autonomous invocation by the agent. The tools include:

- **Weather Tool**: Retrieves current weather and multi-day forecasts for the destination using OpenWeatherMap API
- **Place Search Tool**: Fetches tourist attractions, restaurants, activities, and transportation options using Google Places API with Tavily Search as a fallback
- **Currency Converter Tool**: Converts travel costs between currencies using live exchange rates
- **Expense Calculator Tool**: Computes hotel costs, daily budgets, and total trip expenses using arithmetic operations

This layer ensures that all travel data is sourced from live APIs and processed accurately before being passed to the LLM for synthesis.

4.1.5 Backend API Layer

The Backend API Layer acts as the bridge between the UI frontend and the LangGraph agent engine. It is implemented using FastAPI and exposes a REST endpoint that receives user queries and returns structured itinerary responses. Its primary functions include:

- Receiving travel queries from the Streamlit frontend via HTTP POST requests
- Instantiating the LangGraph agent graph and invoking it with the user query
- Extracting the final itinerary from the agent's message state
- Building and returning the agent execution trace for frontend visualization
- Saving the compiled agent graph as a PNG for inspection and debugging

This layer ensures clean separation between the UI and the agent logic, enabling scalable and maintainable system operation.

4.2 SEQUENCE DIAGRAM

The sequence diagram illustrates the interactions between different entities in the Agentic Travel Planner for itinerary generation. The process begins with the User submitting a natural language travel request through the Streamlit Interface, specifying destination, duration, and preferences. The Streamlit frontend forwards this request as an **HTTP POST** payload to the **FastAPI Backend,** which initializes the LangGraph Agent Graph and invokes it with theuserquery as the initial message state.

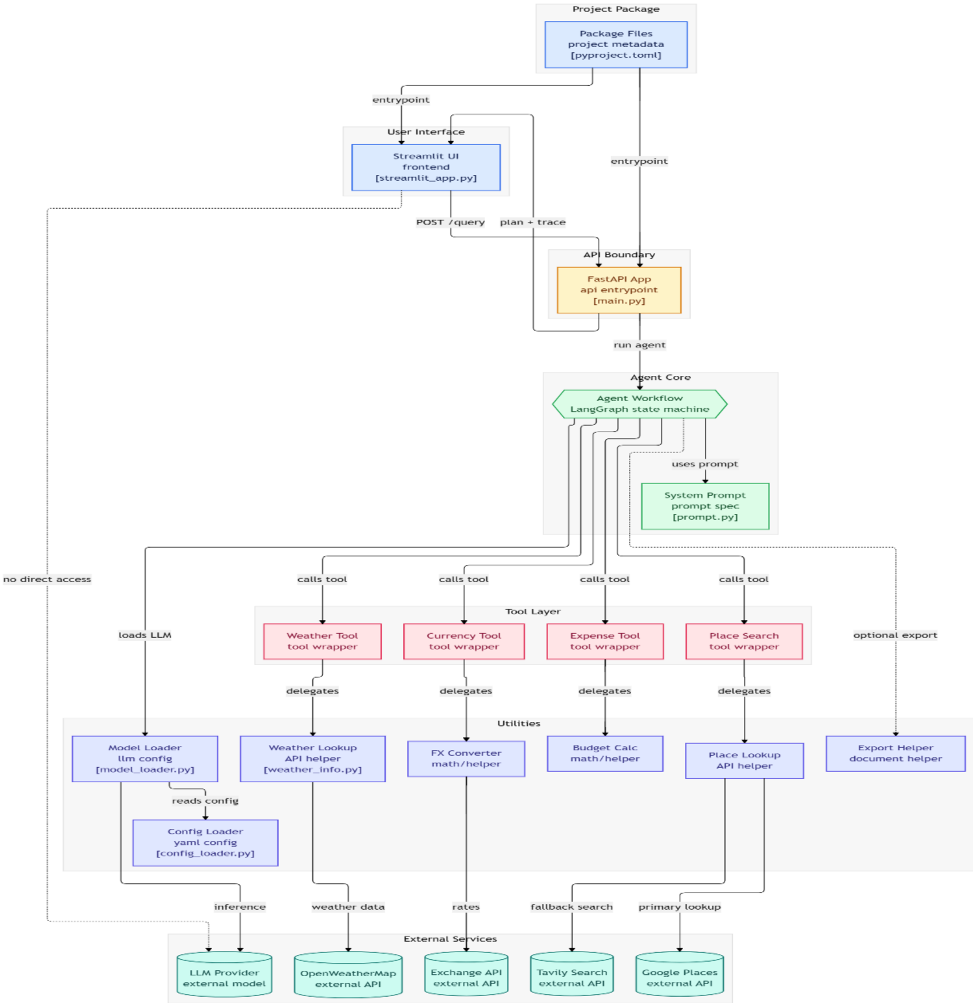

Figure 4.2 Sequence Diagram.

Each tool returns its result back to the agent, which appends the tool output to the message state and passes it back to the LLM for the next reasoning step. This observe-reason-act loop continues until the LLM determines that sufficient information has been collected. The LLM then synthesizes all retrieved data into a complete, Markdown-formatted day-by-day travel plan

The FastAPI backend extracts the final response from the last AI message in the agent's state and builds the agent execution trace by parsing all intermediate messages. The structured response, containing the itinerary and trace, is returned to the Streamlit frontend, which renders the travel plan and displays the live agent activity panel to the user.

4.3 FUNCTIONALITIES

The Agentic Travel Planner integrates multiple intelligent modules that together ensure efficient itinerary generation, real-time data retrieval, and accurate expense planning. Each module is designed to handle a specific aspect of the travel planning workflow, enhancing automation, adaptability, and user engagement.

4.3.1 Agent Orchestration Module

This module is the core of the system, implemented using LangGraph's StateGraph. It manages the full agentic reasoning loop — receiving user queries, invoking tools conditionally, processing tool results, and generating comprehensive travel plans. It supports multi-step reasoning where the agent iteratively calls tools until all required information is collected, then synthesizes a final response.

4.3.2 Weather Information Module

This module retrieves live weather data for the travel destination using the OpenWeatherMap API. It provides both current weather conditions (temperature and description) and multi-day forecasts, enabling the agent to include weather-aware recommendations in the itinerary.

4.3.3 Place Search Module

This module discovers and retrieves destination-specific travel information using the Google Places API as the primary source and the Tavily Search API as a fallback. It covers four categories: tourist attractions, restaurants, activities, and transportation options, providing the agent with rich, location-specific content for itinerary building.

4.3.4 Currency Conversion Module

This module fetches live exchange rates and converts travel costs between currencies using the Exchange Rate API. It allows the agent to present budget estimates in the user's preferred currency, particularly converting costs to Indian Rupees for domestic traveler convenience.

4.3.5 Expense Calculator Module

This module provides arithmetic computation capabilities for travel cost planning. It calculates total hotel costs based on nightly rates and trip duration, aggregates total trip expenses across multiple cost categories, and computes daily budget allocations, enabling the agent to produce detailed and accurate financial breakdowns.

CHAPTER 5

IMPLEMENTATION

5.1 OVERVIEW

This chapter outlines the implementation phase of the Agentic Travel Planner – an Agentic AI-Based Intelligent Travel Planning System. The system is built using a modern, modular, and scalable architecture that integrates artificial intelligence, large language models, and agent-based decision-making to automate and personalize travel itinerary generation. Each technology is carefully selected to ensure efficiency, real-time responsiveness, and accurate data retrieval from live external APIs. The implementation covers the user interface layer, backend API services, AI/ML agent modules, tool integrations, and deployment setup, ensuring seamless operation across all system components.

5.2 TECHNOLOGY STACK

5.2.1 Frontend

The frontend is developed using Streamlit, chosen for its rapid development capabilities and its ability to build interactive, data-driven web applications in pure Python without requiring a separate frontend framework. Streamlit provides a clean, conversational interface where users submit natural language travel queries and receive structured, Markdown-formatted itineraries. It is augmented with custom CSS styling for activity trace cards, animated pulse indicators, and a progress bar that provides real-time feedback during agent execution. The Streamlit interface also communicates with the FastAPI backend via HTTP POST requests using the Python Requests library, enabling seamless data exchange between the UI and the agentic backend.

5.2.2 Backend

The backend is implemented using FastAPI, chosen for its high performance, native support for asynchronous request handling, and automatic OpenAPI documentation generation. FastAPI serves as the bridge between the Streamlit frontend and the LangGraph agent engine. It exposes a /query POST endpoint that receives natural language travel requests, instantiates and invokes the LangGraph agent graph, extracts the final itinerary from the agent's message state, builds a structured execution trace for frontend rendering, and returns a JSON response containing both the itinerary and the agent activity log. The backend is served using Uvicorn, an ASGI server suitable for handling concurrent requests efficiently. CORS middleware is configured to allow cross-origin requests from the Streamlit frontend.

5.2.3 AI and ML Layer

The AI/ML layer forms the core intelligence of the system and is built using Large Language Models (LLMs), LangGraph, LangChain, and LangChain tool integrations. LLMs are accessed through two configurable providers — **Groq** (running the **Qwen3-32B model**) and **OpenAI** (running the **o4-mini model**) — managed through a **YAML-based configuration** file and a ModelLoader class that supports runtime provider switching. The LangGraph StateGraph is used to implement the ReAct (Reasoning and Acting) agent loop, where the LLM observes tool outputs, reasons about next steps, and decides which tools to invoke. LangChain's tool-binding mechanism binds all available tools to the LLM, and LangGraph's ToolNode handles autonomous tool execution within the agent graph. This layer enables multi-step intelligent planning, real-time data synthesis, and comprehensive itinerary generation in a single agent invocation.

5.2.4 External API Integrations

The system integrates four live external APIs to supply real-world travel data to the agent:

- **OpenWeatherMap API**: Provides current weather conditions and multi-day forecasts for any destination worldwide, accessed through the *WeatherForecastTool* utility class.
- **Google Places API**: Supplies tourist attractions, restaurants, activities, and transportation options for the requested destination, accessed through the *GooglePlaceSearchTool* utility class via the langchain-google-community integration.
- **Tavily Search API**: Acts as a fallback search engine for place discovery when Google Places API is unavailable or returns no results, accessed through the *TavilyPlaceSearchTool* utility class via the langchain-tavily integration.
- **Exchange Rate API**: Provides live currency exchange rates for accurate cost conversion, accessed through the *CurrencyConverter* utility class.

5.2.5 Configuration and Environment Management

System configuration is managed through a YAML-based configuration file (config/config.yaml) that stores LLM provider settings and model names. Sensitive credentials — including API keys for Groq, OpenAI, OpenWeatherMap, Google Places, Tavily, and Exchange Rate services — are managed through environment variables loaded via the **python-dotenv** library from a **.env file**, ensuring secrets are never hardcoded into the source. The project uses uv as its package manager with a **pyproject.toml** manifest defining all dependencies and requiring Python 3.12 or higher.

5.3 IMPLEMENTATION PROCESS

The implementation process of the Agentic Travel Planner follows a modular and layered approach, ensuring each component is developed and integrated systematically

Initially, the utility layer is established by implementing individual service classes for each externalAPI— *WeatherForecastTool, GooglePlaceSearchTool, TavilyPlaceSearchTool, CurrencyConverter, ExpenseCalculator*, and *ModelLoader* — each encapsulating API communication logic, error handling, and data formatting within the utils/ module.

Subsequently, the LangChain tool wrappers are built in the tools/ module. Each tool class (*WeatherInfoTool, PlaceSearchTool, CurrencyConverterTool, CalculatorTool*) wraps the utility functions using the @tool decorator, exposing them as callable LangChain tools that the LLM can invoke autonomously. Fallback logic is integrated into the *PlaceSearchTool* so that Tavily Search is automatically used when Google Places returns an error.

Next, the agentic workflow is constructed in the agent/ module using LangGraph's StateGraph. The GraphBuilder class loads the LLM, binds all tools, and compiles the agent graph with a START → agent → conditional tool routing → tool execution → agent feedback loop, terminating when the LLM produces a final response without further tool calls. A system prompt defines the agent's behavior, instructing it to always produce complete day-by-day itineraries with cost breakdowns, weather information, and both mainstream and offbeat location options.

The FastAPI backend is then developed in main.py, wiring together the GraphBuilder, query invocation, agent trace extraction, and REST endpoint exposure. An agent trace builder function parses the full message history to produce a structured execution log for the frontend.

Finally, the UI interface is developed in streamlit_app.py, providing the user-facing chat form, real-time progress indicators, Markdown itinerary rendering, and the agent activity trace panel. The complete application is run as two concurrent processes — the FastAPI backend on port 8000 and the UI frontend — communicating via HTTP, forming a fully integrated and deployable travel planning system

CHAPTER 6

RESULT

6.1

TRAVEL PLANNER UI

The Travel Planner UI acts as the primary interface for users to interact with the Agentic Travel Planner system. It presents a conversational, single-page layout that combines natural language input, real-time agent execution feedback, and a rich Markdown-formatted travel itinerary output. The page header clearly identifies the application as "Travel Planner Agentic Application," and a prominent prompt — *"How can I help you in planning a trip? Let me know where you want to visit"* — guides users to enter their travel request. The interface is rendered with a soft gradient background and card-based styling to deliver a modern, visually comfortable experience.

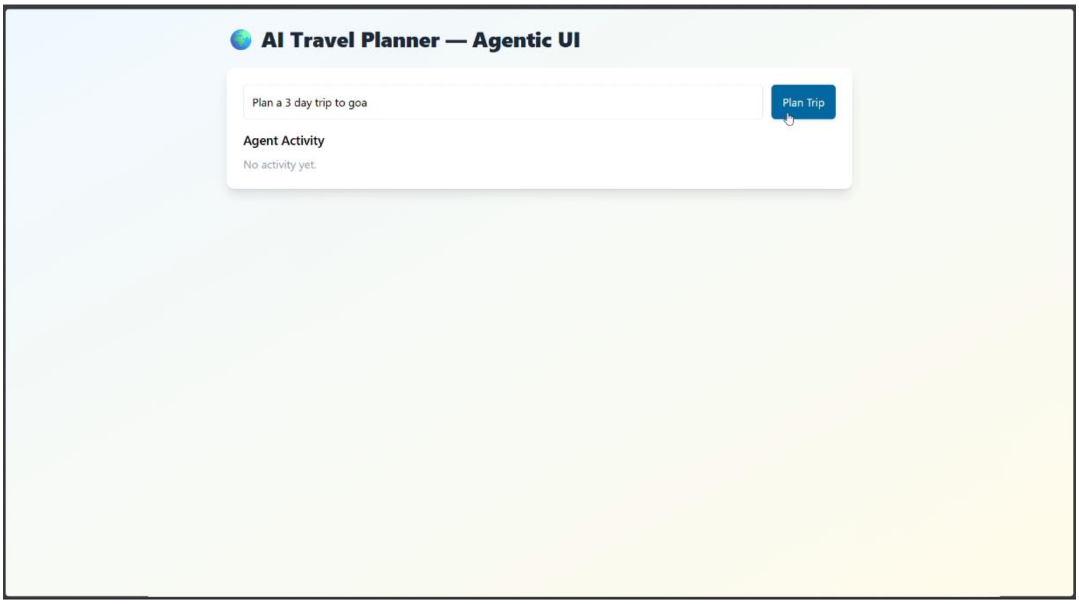

Figure 6.1 Travel Planner UI.

6.2 TRIP REQUEST INPUT FORM

The Trip Request Input Form enables users to describe their intended journey in natural language. A clean text input field accepts free-form queries such as *"Plan a 5-day trip to Goa"* or *"Suggest a budget honeymoon trip to Paris in July."* A Send button submits the query to the FastAPI backend, which instantiates the LangGraph ReAct agent and begins tool-assisted planning. The form auto-clears after each submission, keeping the interface uncluttered. This page serves as the entry point for the entire agentic workflow, converting user intent into a structured agent execution pipeline.

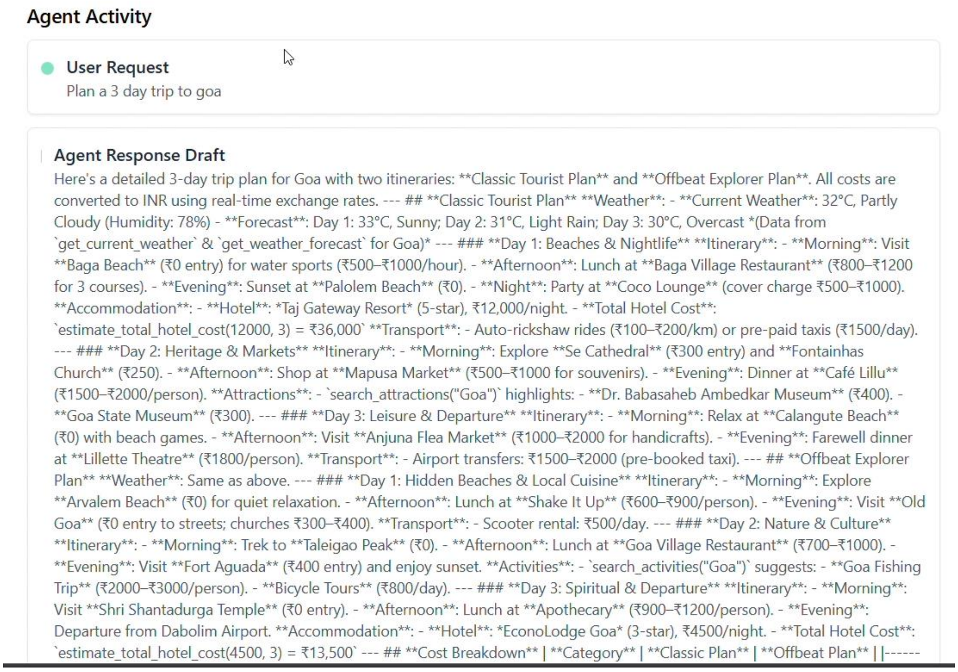

Figure 6.2 Trip Request Input Form.

6.3 GENERATED ITINERARY OUTPUT

The Generated Itinerary Output section displays the fully composed travel plan returned by the AI agent. The output is rendered as structured Markdown, organized into day-by-day sections covering activities, accommodation suggestions, dining recommendations, estimated costs with currency conversion, weather advisories, and transport options. The itinerary header includes the generation timestamp and agent attribution. A disclaimer at the bottom reminds users to verify time-sensitive information such as prices and operating hours before travel. This output page represents the culmination of the agentic pipeline — from user query to actionable, comprehensive travel journey.

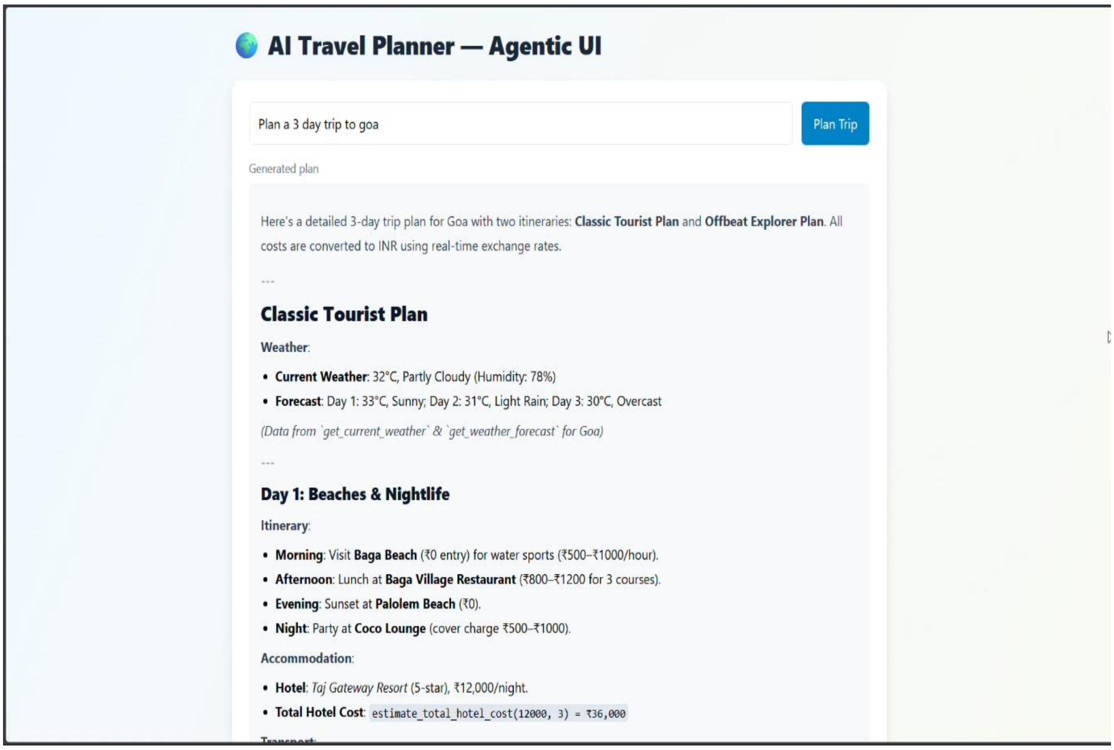

Figure 6.3 Generated Itinerary Output.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The Agentic Travel Planner successfully transforms the traditional travel planning process by introducing an intelligent, tool-augmented, and fully automated approach to trip curation. By integrating Large Language Models (LLMs), a ReAct-based agentic workflow built on LangGraph, and a suite of real-time external tools — including weather information, place search, currency conversion, and hotel discovery — the system enables end-to-end itinerary generation from a single natural language query. Its conversational UI and live agent execution trace provide users with complete visibility into the planning process, eliminating guesswork and improving trust in AI-generated recommendations. The FastAPI backend ensures seamless communication between the frontend interface and the agentic pipeline, while the modular graph-based architecture supports maintainability and extensibility. Overall, the system offers a scalable and intelligent digital solution that modernizes travel planning by combining real-time data, autonomous decision-making, and user-friendly interaction.

7.2 FUTURE ENHANCEMENT

In the future, the system can be enhanced by integrating advanced personalization models that learn from user travel history, preferences, and feedback to deliver increasingly tailored itineraries over time. Incorporating multi-modal support — such as map visualizations, embedded photo previews of destinations, and voice-based query input — can significantly improve user engagement and accessibility. Expanding the agent's toolset to include flight booking APIs, live transportation scheduling, restaurant reservation systems, and visa requirement checkers will make the planner a truly end-to-end travel assistant. Adding support for multi-destination and multi-traveler trip planning, along with collaborative itinerary editing for group travel, will broaden the platform's applicability. Additionally, integrating real-time budget tracking, cloud-based user profile persistence, and multilingual support can improve content diversity and global usability, making the Agentic Travel Planner a fully autonomous, intelligent, and comprehensive travel companion.

REFERENCES

1. Russell, S., & Norvig, P., *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2021.
2. Brown, T. B., et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
3. OpenAI, "GPT-4 Technical Report," 2023.
4. Yao, S., et al., "ReAct: Synergizing Reasoning and Acting in Language Models," *International Conference on Learning Representations (ICLR)*, 2023.
5. Wei, J., et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
6. Vaswani, A., et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
7. LangChain Documentation, "Building Applications with LLMs using LangChain," 2024. [Online]. Available: https://docs.langchain.com
8. LangGraph Documentation, "LangGraph: Stateful, Multi-Actor Applications with LLMs," 2024. [Online]. Available: https://langchain-ai.github.io/langgraph
9. FastAPI Documentation, "FastAPI – Modern, Fast Web Framework for Building APIs with Python," 2024. [Online]. Available: https://fastapi.tiangolo.com
10. OpenWeatherMap API, "Current Weather Data API Documentation," 2024. [Online]. Available: https://openweathermap.org/api

11. Google Places API, "Places API Documentation," Google Developers, 2024. [Online]. Available: https://developers.google.com/maps/documentation/places
12. Bommasani, R., et al., "On the Opportunities and Risks of Foundation Models," *arXiv preprint arXiv:2108.07258*, 2021.
13. OpenAI, "Function Calling and Tool Use in LLMs," 2024. [Online]. Available: https://platform.openai.com/docs/guides/function-calling
14. Mnih, V., et al., "Human-level Control through Deep Reinforcement Learning," *Nature*, vol. 518, pp. 529–533, 2015.
15. Significant Gravitas, "AutoGPT: An Autonomous GPT-4 Experiment," GitHub Repository, 2023. [Online]. Available: https://github.com/Significant-Gravitas/AutoGPT
16. Nakano, R., et al., "WebGPT: Browser-Assisted Question-Answering with Human Feedback," *arXiv preprint arXiv:2112.09332*, 2021.
17. ExchangeRate-API, "Currency Conversion API Documentation," 2024. [Online]. Available: https://www.exchangerate-api.com/docs
18. Streamlit Documentation, "Streamlit – A Faster Way to Build and Share Data Apps," 2024. [Online]. Available: https://docs.streamlit.io
19. Patil, S., et al., "Gorilla: Large Language Model Connected with Massive APIs," *arXiv preprint arXiv:2305.15334*, 2023.
20. Qin, Y., et al., "Tool Learning with Foundation Models," *arXiv preprint arXiv:2304.08354*, 2023.