

Large Scale Question

Answering at Tourism Data

High Level Design

Domain: NLP

Author : Debasish Mohanty

Date: 15.02.2024

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date issued | Version | Description | Author |
| Feb 15th , 2024 | 1.1 | First Draft | Debasish Mohanty |

CONTENTS

[Abstract 3](#_Toc110595993)

1. [Introduction 4](#_Toc110595994)

1.1 [What is High-Level Design Document? 4](#_Toc110595995)

1.2 [Scope 4](#_Toc110595996)

2. [General Description 5](#_Toc110595997)

2.1 [Definitions 5](#_Toc110595998)

2.2 [Product Description 5](#_Toc110595999)

2.3 [Problem Statement 5](#_Toc110596000)

2.4 [Proposed solution 6](#_Toc110596001)

2.5 [Tools used 6](#_Toc110596004)

2.7 Hardware Requirments .......................................................................................................................7

2.8 [Constraints](#_Toc110596006) 7

2.9 [Assumptions](#_Toc110596007) 7

3. [Design Details](#_Toc110596008) 7

3.1 [Process Flow](#_Toc110596009) 7

3.2 [Event Log](#_Toc110596010) 8

3.3 [Error Handling](#_Toc110596011) 9

4. [Performance](#_Toc110596012) 9

4.1 [Reusability](#_Toc110596013) 9

4.2 [Application Compatibility](#_Toc110596014) 9

4.3 [Resource Utilization](#_Toc110596015) 10

4.4 [Deployment .....................................................................................................................................](#_Toc110596015)10

5. [Dashboards](#_Toc110596016) 10

5.1 [KPIs (Key Performance Indicators)](#_Toc110596017) 11

6. [Conclusion](#_Toc110596018) 11

# Abstract

This document lays out the architectural details of "Large Scale Question Answering System for tourism data," a cloud-based platform designed to address tourism-related inquiries. At its core lies a pre-trained LLM (Gemini - Pro) equipped with extensive tourism knowledge. By harnessing this knowledge through strategic prompt engineering and context management, the system delivers informative and user-friendly responses in real-time. To ensure the information flows seamlessly, backend generators and frontend Server-Sent Events (SSE) work in tandem, minimizing user latency. Furthermore, the system transforms raw LLM output into visually appealing and well-structured responses through the implementation of HTML parsing on the frontend. ReactJS is employed for the user interface, facilitating intuitive navigation and effortless interaction with the platform. The backend, built on the robust FastAPI framework, seamlessly integrates the LLM and handles user input processing and response delivery. Finally, deployment on an AWS EC2 instance with Cross-Origin Resource Sharing (CORS) configuration guarantees smooth communication between the frontend and backend, ensuring a comprehensive and accessible solution for tourism information seekers.

# Introduction:

## 1.1 What is High-Level Design Document?

The goal of this HLD or a high-level design document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all of design aspects and define them in detail
* Describe all user interfaces being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and architecture of the project
* List and describe the non-functional attributes such as security, reliability, maintainability, portability, reusability, application compatibility. resource utilization, serviceability

## 1.2 Scope:

The HLD documentation presents the structure of the system, such as database architecture, application architecture (Layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

# General Description

## 2.1 Definitions:

|  |  |
| --- | --- |
| **Term** | **Description** |
| LLM | Large Language Model |
| SSE | Server Site Event |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 

## 2.2 Product Description:

"Large Scale Question Answering System" is the one-stop platform for insightful tourism guidance. This cloud-based system leverages the power of advanced large language models, trained on vast amounts of tourism data, to answer your questions effortlessly. Get real-time, personalized recommendations tailored to your interests.

## 2.3 Problem Statement :

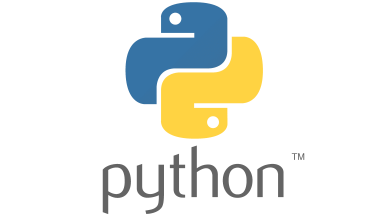
Navigating the vast sea of travel information can often feel like sailing a rudderless boat in a storm. Tourists yearning for a dream vacation frequently find themselves grappling with a frustrating reality: unearthing comprehensive, personalized, and current information proves remarkably challenging. Traditional search engines and travel websites, while seemingly abundant, tend to bombard users with overwhelming, generic results. Crucial context and details specific to individual needs are often lost in the deluge, leaving travelers feeling bewildered and overwhelmed. This information overload translates to lost hours spent sifting through irrelevant content, ultimately leading to disappointment and missed opportunities for truly enriching experiences. Imagine the frustration of a solo backpacker seeking hidden eco-adventures in Bali, only to be flooded with cookie-cutter resort packages. Or picture the anguish of a family hoping to discover unique cultural experiences in Rome, only to find generic recommendations for overcrowded tourist traps. It's precisely this struggle that fuels our mission to bridge the gap between travelers and the information they crave.

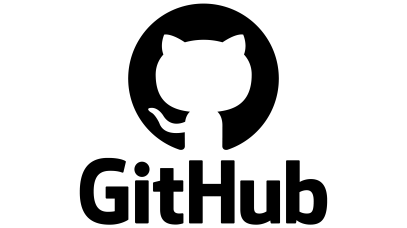
## 2.4 Proposed Solution :

Develop "Large Scale Question Answering System for Tourism Data," a cloud-based platform leveraging the power of pre-trained language models (LLMs) to provide instant, informative, and personalized answers to tourism-related inquiries.

## 2.5 Tools Used:

The process of developing the architecture involves using Python programming language and frameworks like vertexai ( gemini-pro ), langchain, are used, and FastAPI is utilized for backend API development. Additionally, other tools are used to deploy the model.

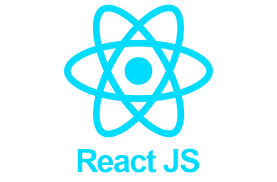












* For using the Gemini LLM vertexai is used
* Langchain is used to manage the context and prompt
* Uvicorn is used to run the server
* ReactJS and FastAPI were used for building the web application.
* GitHub was used as version control system
* VSCode and NeoVim were used as primary IDE.
* Amazon EC2 instance is used for model deployment.

## 2.6 Hardware Requirements :

* Windows Server, Linux, or any operating system that can run as a webserver, capable of delivering HTML5 content.
* Minimum 1.10 GHz processor or equivalent.
* Between 1-2 GB of free storage
* Minimum 512 MB of RAM
* 3 GB of hard-disk space

## 2.7 Constraints :

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

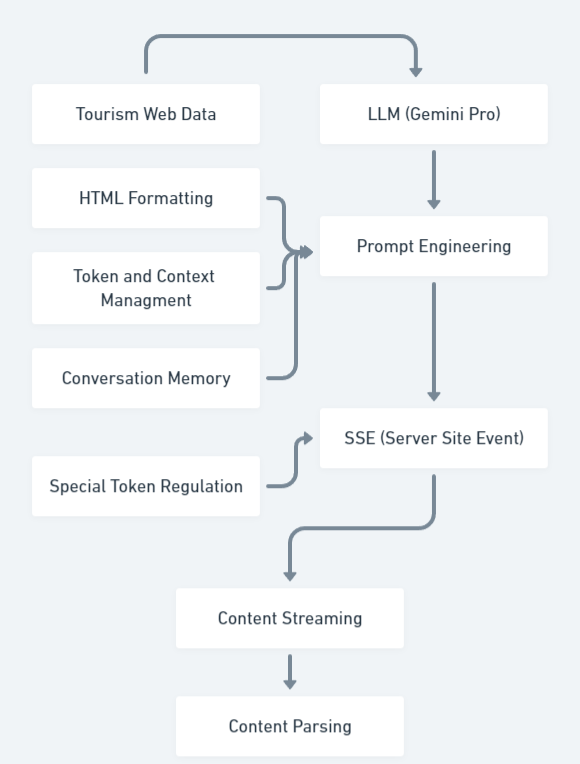
## 2.8 Assumptions :

The primary goal of this project is to apply the previously described use case to new datasets received via the user interface. It is assumed that all components of the project are able to function seamlessly as intended by the designer and that the training data for the model is as accurate as possible.

# Design Details :

## 3.1 Process Flow :

We utilized various machine learning models to complete the task, and the model development workflow is illustrated below.

**Development Work-Flow:**

## 3.2 Event Log :

The system should log every event so that the user will know what process is running internally. Initial step-by-step description:

1. The system identifies at what level logging is required
2. The system should be able to log each and every system flow
3. Developer can choose logging method.
4. System should not hang even after so many loggings. Logging just because we can easily debug issues, so logging is mandatory to do.

The event logging for the machine learning development process is facilitated by the implementation of a logging library, which is utilized consistently throughout the codebase for each individual model. The event logs, encompassing valuable information regarding the execution and progress of the models, are securely stored within the code repository, ensuring their availability for future reference and analysis.

## 3.3 Error Handling :

Errors should be encountered, an explanation will be displayed as to what went wrong ? An error will be defined as anything that falls outside the normal intended usage.

## 4. Performance:

Designed with user experience in mind, "Large Scale Question Answering System for Tourism Data" prioritizes accessibility and ease of use. Forget clunky search bars and endless menus. Simply ask your question in plain language, and our system instantly retrieves the knowledge you need. No complex queries or specific syntax required – just natural conversation. The intuitive interface guides you effortlessly through the process, with clear prompts and visual cues ensuring a smooth interaction.

## 4.1 Reusability :

The code written and the components used should have the ability to be reused with no problems.

## 4.2 Application Compatibility :

The different components for this project will be using Python as an interface between them, each component will have its own task to perform, and it is the job of Python to ensure proper transfer of information.

## 4.3 Resource Utilization :

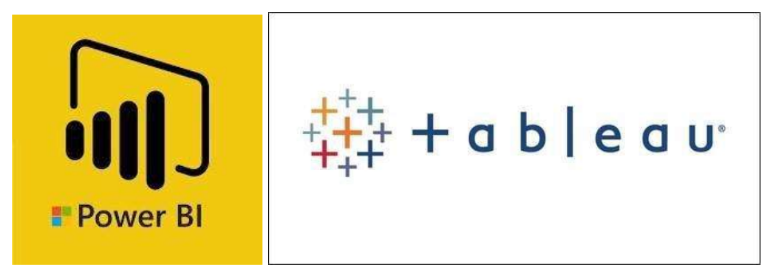
When any task is performed, it will likely use all the processing power available to it until finished.

## 4.4 Deployment :



## 5. Dashboard:

To ensure ongoing improvement and user satisfaction, "Large Scale Question Answering System for Tourism Data" will leverage a dedicated performance dashboard. This transparent window into the system's inner workings will showcase insights across various metrics, empowering us to refine the tool for optimal results.



Upon the initiation of chemical data acquisition for a user, the system will seamlessly integrate dynamic dashboards to present temporal charts showcasing the evolving progress across diverse chemical factors.

## 5.1 KPIs (Key Performance Indicators):

1. **User engagement**: Metrics like average session duration and number of questions asked will reveal how users interact with the system.
2. **Answer accuracy**: We'll track the percentage of questions answered correctly and user feedback to assess the quality of information provided.
3. **Response time**: Server-Sent Events technology ensures near-instant responses, but the dashboard will monitor and report on average response times, identifying areas for potential optimization.
4. **Topic coverage**: Analyzing the range of questions being asked will help us assess if the system's knowledge base effectively covers diverse tourism information needs.
5. **User demographics**: Understanding the user base will allow us to tailor the system and content to better serve different segments of travelers.

# Conclusion:

This project demonstrates the immense potential of Generative AI to revolutionize information access across diverse domains. By empowering users to retrieve relevant data in their preferred language proficiency, tailored to their specific needs, it transcends the limitations of traditional search tools. The intuitive, conversational interface underscores the versatility of Generative AI, fostering natural interaction and seamless knowledge acquisition. Further, the successful completion showcases a multifaceted skillset, encompassing full-stack web development, cloud deployment expertise, and the ability to harness cutting-edge AI models to achieve desired outcomes. This project stands as a testament to the transformative power of Generative AI and paves the way for future developments that empower individuals with personalized, accessible information across various fields.