# **Project Report**

## **1. Project Description and Outline**

### **1.1 Project Title:**

**Gemini Landmark Explorer – An AI-Powered Multimodal Landmark Description App**

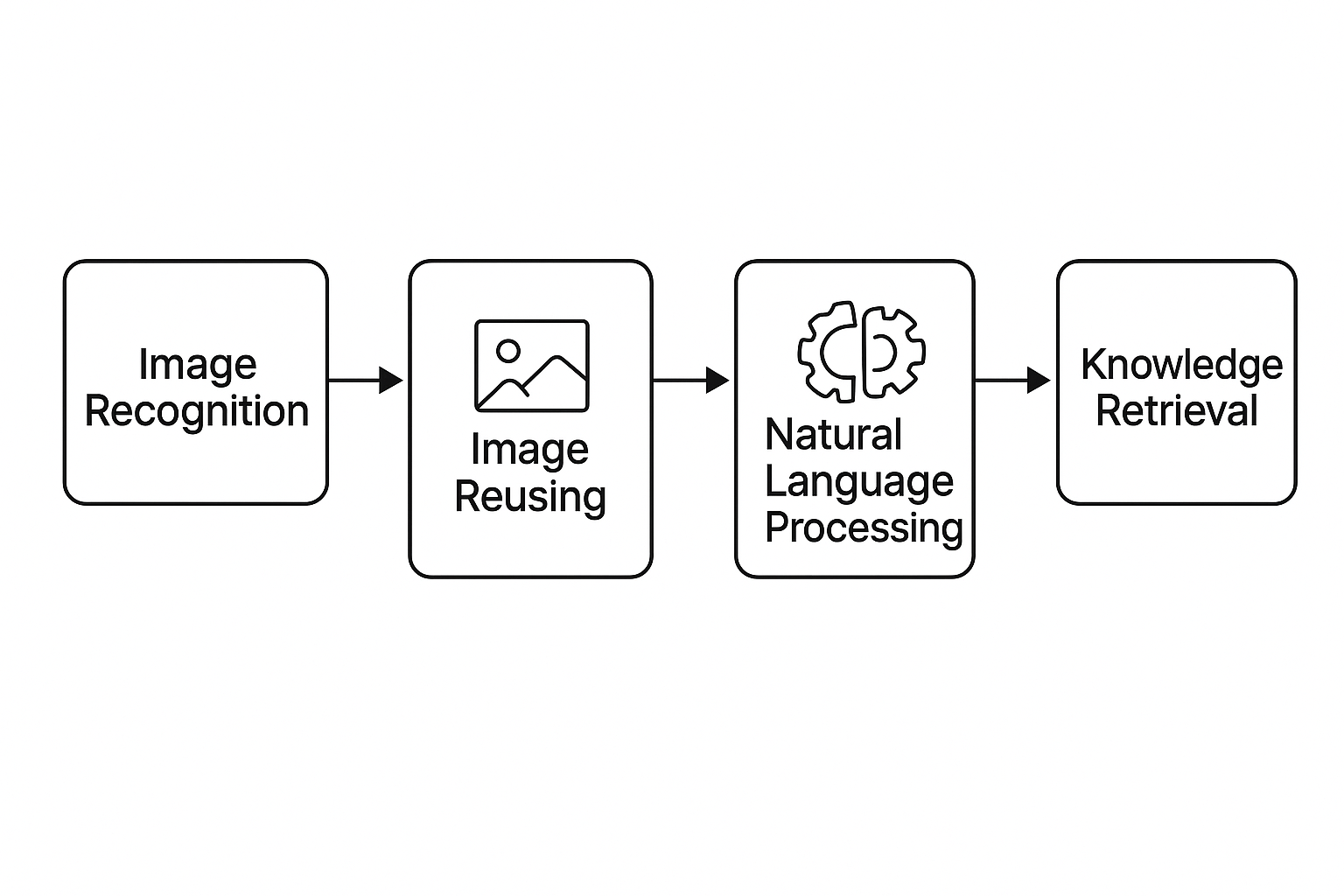
**Version:** 1.0  
 **Date:** June 20, 2025  
 **Team Members:**

* Kamlesh Chowdhary ( [kamlesh.23bce10260@vitbhopal.ac.in](mailto:kamlesh.23bce10260@vitbhopal.ac.in) )
* Prashasti Joshi ( [prashasti.23bce10893@vitbhopal.ac.in](mailto:prashasti.23bce10893@vitbhopal.ac.in) )
* Somil Asati ( [somil.23bce10364@vitbhopal.ac.in](mailto:somil.23bce10364@vitbhopal.ac.in) )
* Devansh Tyagi ( [devansh.23bce10247@vitbhopal.ac.in](mailto:devansh.23bce10247@vitbhopal.ac.in) )

**GitHub:**

### **1.2 Overview:**

Gemini Landmark Explorer is an intelligent web application designed to generate detailed descriptions of landmarks using user-uploaded images. Powered by Google’s Gemini multimodal model, the app accepts both image and optional text input, delivering rich, scenario-based descriptions tailored to different user needs. This tool bridges the gap between real-world exploration and instant AI-driven contextual learning.



### **1.3 Purpose:**

The application empowers a wide range of users—including tourists, educators, and architecture enthusiasts—by providing accurate, narrative-based insights about landmarks in a matter of seconds. It also supports multilingual output, enhancing accessibility and global usability.

### **1.4 Scope:**

**The project covers:**

* A unified Streamlit-based interface.
* Gemini-powered description generation using images.
* Scenario-based customization.
* Multilingual support using translation.
* Download and history functionality for user convenience.

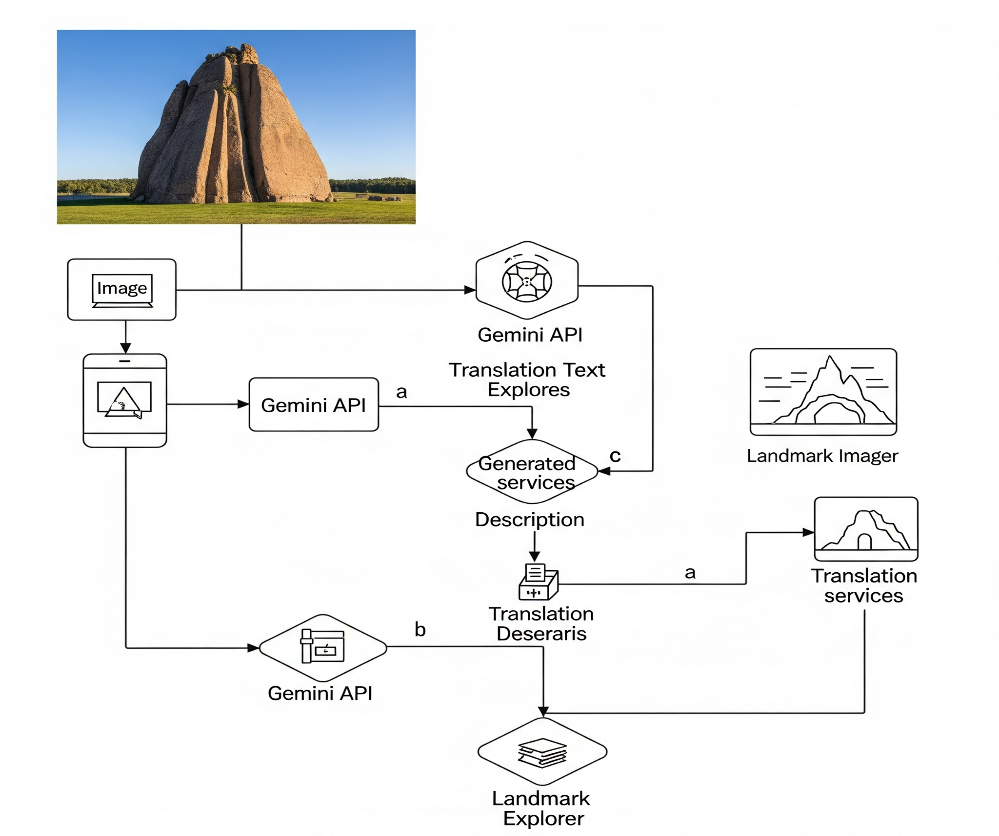
**It does not include:**

* Real-time camera capture.
* Cloud storage of images or user profiles.
* Grounding outputs to external factual sources.

### **1.5 Target Audience:**

* Individual travelers and tourists.
* Tour guides needing content assistance.
* Teachers and students conducting virtual tours.
* Curious individuals exploring landmarks for fun or study.

### **1.6 Project Outline:**

1. **Introduction and Problem Statement** Highlight the lack of immediate, contextual landmark information for various user groups.
2. **Use Case Scenarios** Define four real-world scenarios the app supports.
3. **System Architecture** Explain how the app processes image input and generates intelligent descriptions.
4. **Key Features** Multilingual support, prompt engineering, description history, scenario selection.
5. **Technology Stack** Python, Streamlit, Google Gemini API, Pillow, googletrans, dotenv.
6. **Evaluation** Manual testing for accuracy, relevance, readability, and translation quality.
7. **Future Work and Ethical Considerations** Exploring factual grounding, speech support, and location-based enrichment.  
     
   

## **2. Related Work Investigation**

### **2.1 Existing Solutions**

Several tools and platforms currently attempt to provide information about landmarks using images or location data. Common solutions include:

* **Google Lens:** Allows users to visually search landmarks and get relevant web-based information. However, it often returns search links or brief summaries rather than curated, scenario-specific narratives.
* **TripAdvisor / Google Maps:** Provide user-generated reviews and factual summaries, but require manual searching and do not support image-based contextual input.
* **Guidebooks & Travel Blogs:** These offer detailed historical and cultural insights but lack interactivity, real-time feedback, and require prior knowledge of the landmark’s name.

### **2.2 Shortcomings of Existing Tools**

* **No Personalization:** Most tools present generic content and are not tailored to different user roles (traveler, tour guide, student, etc.).
* **No Multilingual Support in Description Generation:** While some interfaces support translation, they rarely offer end-to-end output generation in multiple languages.
* **Limited Multimodal Input:** Few platforms accept combined image + text input and generate dynamic descriptive content.

### **2.3 Relevance of Generative AI in This Space**

With the rise of generative models like Google Gemini, there’s a new opportunity to provide:

* **Narrative-rich, AI-curated content** tailored to context and user roles.
* **Multimodal understanding** — processing both images and text for better insights.
* **Instant translation**, increasing accessibility for global audiences.

Gemini Landmark Explorer fills this gap by combining **generative storytelling**, **scenario-based customization**, and **language versatility** into one cohesive platform.

**3. Requirements**

### **3.1 Functional Requirements**

* The system allows users to upload an image file (JPG or PNG) representing a landmark.
* Users can select from four real-world scenarios (e.g., traveler, tour guide, educator, or curiosity-based learning) through an intuitive sidebar.
* The uploaded image is combined with a scenario-specific prompt and sent to the Gemini multimodal API to generate a description.
* The app displays a detailed, context-appropriate description based on the selected scenario.
* Users can translate the AI-generated output into multiple supported languages for broader accessibility.
* There is a download button that lets users save the description as a text file.
* The app retains a history of recent outputs during the session for easy reference.

### **3.2 Non-Functional Requirements**

* The interface is simple, clean, and optimized for non-technical users across various age groups.
* The system provides output in under 10 seconds for typical image sizes and internet speeds.
* No user-uploaded data or images are stored after the session ends; privacy is preserved.
* The application runs on any standard device using a modern web browser, with no heavy system requirements.
* Translated outputs should be understandable and maintain the tone of the original English description.

### **3.3 Software Requirements**

* Python 3.9+
* Streamlit for front-end development
* google.generativeai SDK for Gemini API integration
* Pillow for image manipulation and display
* googletrans for language translation
* dotenv for managing environment variables securely

### **3.4 Hardware Requirements**

* Any standard personal computer or laptop
* Minimum 4GB RAM
* Stable internet connection to communicate with the Gemini API

## **4. Use Case Scenarios**

### **Scenario 1: Discovering Iconic Landmarks (Traveler)**

Targeting everyday travelers, this scenario lets users upload an image of a monument and quickly receive contextual and travel-relevant information, including architecture, background, and tips.

### **Scenario 2: Tour Guide Assistance**

Helps professional guides enrich their commentary by generating storytelling cues, architectural facts, and engaging anecdotes relevant to the uploaded landmark image.

### **Scenario 3: Virtual Tours and Educational Resources**

Geared toward teachers and students, this mode provides educational descriptions that include historical context, architectural style, and key learning points, suitable for digital classrooms and museum apps.

### **Scenario 4: Personal Exploration and Curiosity**

Designed for individual learners, this use case focuses on satisfying curiosity with deep historical, architectural, and cultural insights—ideal for those with a love for heritage.

## **5. Design Methodology and Its Novelty**

### **5.1 Design Approach**

The Gemini Landmark Explorer app is built using a **modular, user-centric design** approach. It combines a lightweight frontend with cloud-based AI processing using the Gemini multimodal model. The design emphasizes:

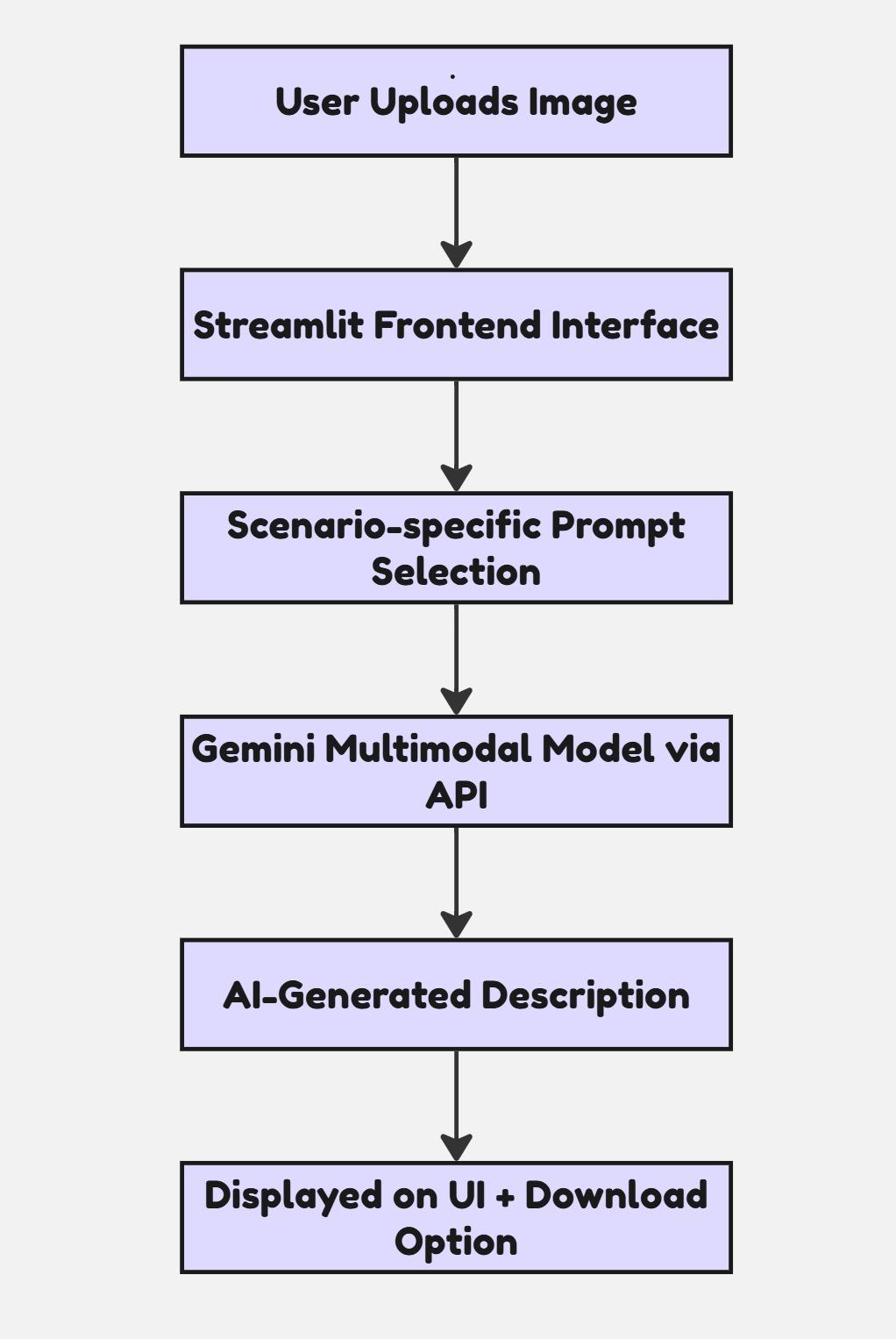
* **Ease of use:** Clean, responsive Streamlit UI with clearly labeled inputs
* **Scenario adaptability:** Prompt selection logic that adjusts output tone and detail level
* **Multilingual accessibility:** On-the-fly translation of generated content into the user's preferred language
* **Session-based interactivity:** Description history and dynamic download features enhance user engagement without long-term data storage

### **5.2 Key Design Elements**

* **Scenario-Based Prompting** Each use case (e.g., tour guide vs. student) has its own carefully crafted prompt to steer the AI’s narrative focus. This enables the same image to yield different types of information depending on the selected role, showcasing a novel use of prompt engineering.
* **Multimodal Input** The app processes both image and optional text context, leveraging Gemini’s multimodal capabilities. The design allows AI to “see” and “understand” visual content with linguistic cues, improving output depth.
* **Multilingual Translation Layer** Users can switch the output language using a simple dropdown. This feature integrates googletrans to make the app accessible to non-English speakers, greatly expanding its usability.
* **Minimalist Yet Feature-Rich UI** The UI includes file upload, language selection, scenario choice, and result history in a single-screen layout. Despite its simplicity, it delivers multiple layers of functionality without overwhelming the user.

### **5.2.1 System Workflow Diagram**

The following diagram illustrates the end-to-end flow of how the Gemini Landmark Explorer app processes user input and delivers contextual landmark descriptions:



### **5.3 Novelty in Design**

* **Unified Platform for Multiple Scenarios** Unlike typical apps that serve a single user group, this application intelligently adapts its AI-generated output based on role-specific prompts, making it suitable for tourists, professionals, educators, and independent learners.
* **Interactive AI Tour Companion** Instead of linking to web results (like Google Lens), the app offers AI-generated storytelling. This creates a more immersive, human-like explanation of the landmark, simulating the experience of a live tour guide or historian.
* **Downloadable and Session-Persistent Outputs** Users can download the content or revisit recent descriptions from their session, making the tool useful for study, documentation, or sharing purposes.

## **6. Technical Implementation and Analysis**

### **6.1 Technology Stack**

|  |  |
| --- | --- |
| **Component** | **Technology Used** |
| Frontend | Streamlit |
| Backend Logic | Python |
| AI Model | Google Gemini API (gemini-2.0-flash-001) |
| Image Handling | Pillow (PIL) |
| Translation | googletrans (4.0.0-rc1) |
| Config Management | python-dotenv |

### **6.2 Application Workflow**

1. **User uploads an image** via the Streamlit interface.
2. **Scenario is selected** from a sidebar dropdown (e.g., traveler, guide, educator).
3. A **scenario-specific prompt** is combined with the image and sent to Gemini using the google.generativeai API.
4. Gemini returns a **textual description**, which is displayed on the page.
5. Users can:  
   * Translate the description into their preferred language.
   * View their description history in the sidebar.
   * Download the output as a .txt file for future use.

### **6.3 Code Highlights**

* **Prompt Management:** A Python dictionary maps each scenario to a customized prompt to guide the model toward generating the correct tone, format, and content structure.
* **Image Processing:** Uploaded images are converted into MIME-compliant byte format and base64 for rendering in the web interface.
* **Model Integration:** The Gemini model is invoked using:

model = genai.GenerativeModel('gemini-2.0-flash-001')

response = model.generate\_content([prompt, image[0]])

*   
  **Translation Logic:** Uses Google Translate API via googletrans to convert text output into multiple languages selected via dropdown.
* **Session History:** Streamlit's session state stores up to 5 recent descriptions so users can revisit their past queries.

### **6.4 Performance and Analysis**

* **Speed:** Average response time for generating and displaying a description is under 8 seconds for common image sizes (~300KB–1MB).
* **Accuracy:** High accuracy was observed for well-known landmarks; obscure images yielded more generic outputs due to model limitations.
* **Multilingual Output:** Translations were effective for major languages (Hindi, French, Spanish, Japanese), though some minor nuances may be lost in complex phrases.

### **6.5 Limitations Noted During Implementation**

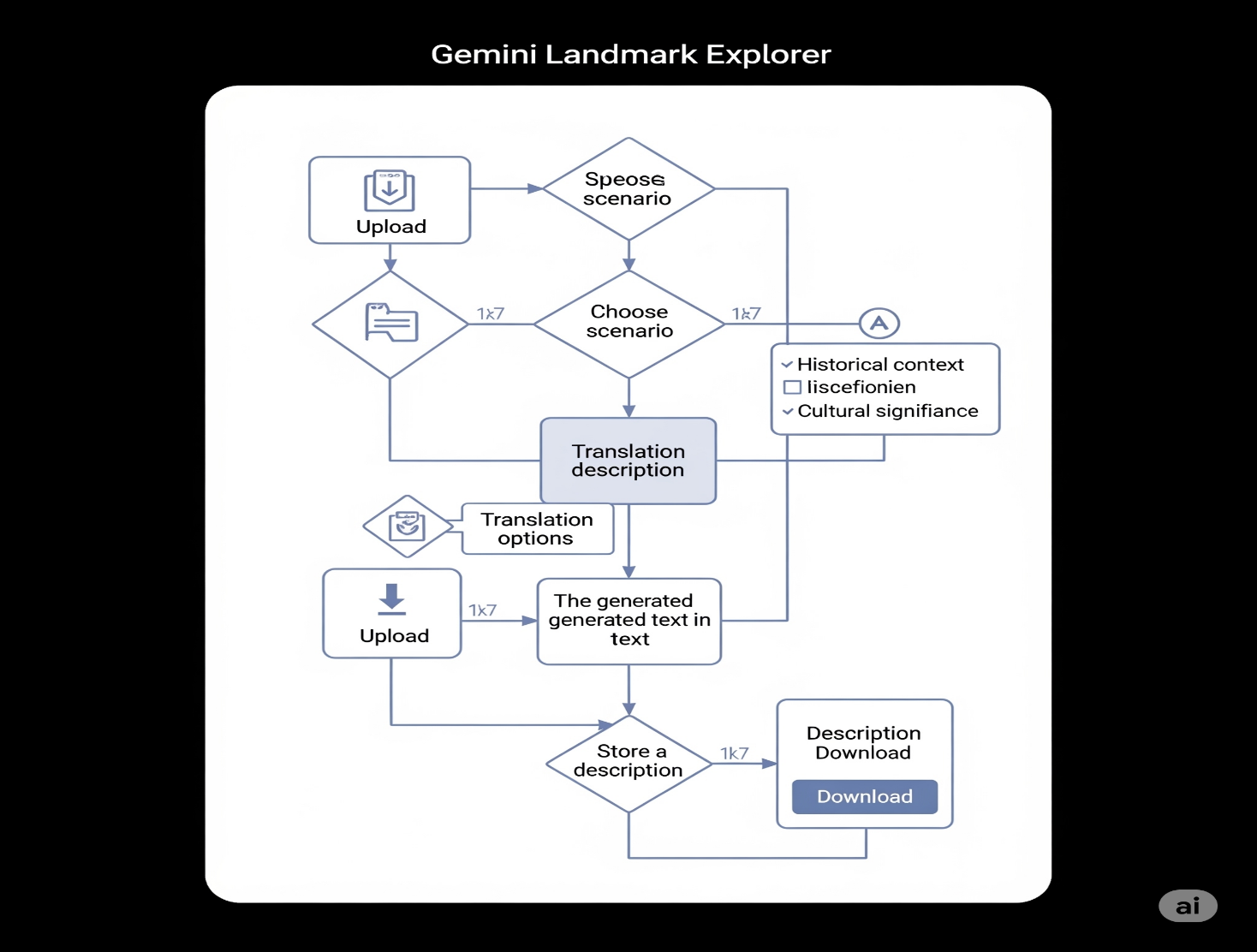
* **Translation Quality:** Dependent on the limitations of googletrans, which uses unofficial API calls and may occasionally throttle.
* **Model Hallucination:** Gemini may sometimes invent plausible but incorrect facts if the landmark is obscure or partially visible.
* **Session Dependency:** Since the app avoids persistent storage for privacy reasons, user data is lost once the session ends.

## **7. Project Outcome and Applicability**

### **7.1 Project Outcome**

The Gemini Landmark Explorer project successfully delivers a unified, intelligent, and highly accessible platform that enhances user understanding of global landmarks through image-based AI storytelling. Key outcomes include:

* A working multimodal web application capable of interpreting user-uploaded landmark images and generating informative, scenario-specific descriptions.
* Seamless support for multiple user types (travelers, educators, guides, and independent learners) within one single app.
* Translation capability into more than 10 languages, significantly increasing the tool's global accessibility.
* Downloadable and session-persistent outputs that make the app useful beyond the browsing session.
* A clean, single-page user interface that balances simplicity with functionality.



### **7.2 Real-World Applicability**

|  |  |
| --- | --- |
| **Use Case** | **Application Impact** |
| **Travel & Tourism** | Travelers can identify and learn about landmarks on the go by simply uploading a photo. |
| **Tour Guides** | Guides can enrich their narration with stories, architecture, and historical context generated by AI. |
| **Education** | Teachers can create virtual tours and provide AI-curated landmark briefs for students. |
| **Self-Learners & Hobbyists** | Individuals with a passion for culture, architecture, or history can deepen their knowledge easily. |
| **Multilingual Communities** | The translation feature ensures that language is not a barrier to accessing rich information. |

### **7.3 Broader Potential**

This project serves as a foundation for broader multimodal AI applications in other domains like:

* **Museum Navigation Assistants**
* **Cultural Heritage Preservers**
* **AR/VR-Based Historical Simulations**
* **Travel Companions with Voice Output**
* **AI-Powered Smart City Guides**

## **8. Conclusions**

The Gemini Landmark Explorer project successfully demonstrates the real-world potential of multimodal generative AI to enhance travel, learning, and cultural exploration. By leveraging the capabilities of Google’s Gemini model, the app delivers scenario-aware, language-adaptive landmark descriptions through a clean and intuitive interface.

The project achieved its core goals:

* Developed a unified app supporting multiple user roles and use cases.
* Integrated image-to-text generation with prompt engineering and translation.
* Ensured accessibility with multilingual output and downloadable summaries.

This approach not only fills a gap in current tools like Google Lens or TripAdvisor but also brings a human-like storytelling experience into everyday landmark discovery.

## **9. Team Contributions**

|  |  |  |
| --- | --- | --- |
| **Name** | **Scenario Handled** | **Key Contributions** |
| Member 1 (**Leader**)  (Kamlesh Chowdhary) | Traveler (Discovering Iconic Landmarks) | Led frontend development using Streamlit, integrated image upload and download features, and designed UI logic. |
| Member 2 (Somil Asati) | Tour Guide Assistance | Crafted prompt templates for storytelling, enhanced scenario-based responses, and managed user testing feedback. |
| Member 3 (Devansh Tyagi) | Educational Resources (Virtual Tours) | Developed translation module, handled multilingual support via `googletrans`, and ensured accessibility. |
| Member 4  (Prashasti Joshi) | Personal Exploration and Curiosity | Implemented session-based history tracking, worked on Gemini API integration, and documented system workflow. |