Project Title:

Gemini Landmark Description App Enhancing Tourist Experience with AI

Team Name:

Toffers

Team Members:

- J. Sri Charan Reddy
- Abhinav Putta
- Chikka Sai Siddarth
- Chintha Nithish
- Gurram Raghavendra

Phase-1: Brainstorming & Ideation

Objective:

Gemini Landmark Description App is a premium AI-based app that works to give users correct information regarding any major landmark in the world. Users can get a brief description of a landmark, such as historical significance, architectural features, cultural context, and interesting facts, by taking a photo of the landmark. The app also creates a realistic 3D model of the landmark so that users can see its structure in various angles, thus making it a complete experience.

Key Points:

1. **Problem Statement:**

    - Most tourists and history buffs have a hard time getting reliable, current information on landmarks when traveling to new destinations.
    - People tend to depend on search engines or guides, which might not be readily available, particularly in rural areas.
    - Traditional sources provide only static presentations without interactive experiences like 3D visualization and immersive content.

2. **Proposed Solution:**

    - Utilize Gemini Flash AI to analyze landmark images captured by users and provide instant descriptions.
    - **3D Model Generation:** The app will reconstruct a realistic, interactive 3D model of the landmark, allowing users to rotate, zoom, and explore the structure from different perspectives.
    - **Offline Capabilities:** The app will store landmark data for offline access, helping users in areas with limited connectivity.

3. **Target Users:**

    - **Travelers & Tourists:** People at historical or cultural destinations who wish to have immediate, accurate information on landmarks.
    - **History & Architecture Enthusiasts:** Individuals who want to see in-depth historical and architectural information, such as major events and structural aspects.

- **Students & Researchers:** Scholars seeking well-organized milestone descriptions for research and study.
- **Content Providers & Teachers:** Travel bloggers, vloggers, and instructors who require accurate landmark information to enhance their content and lessons.

4. **Expected Outcome:**

- A revolutionary AI-based landmark recognition application that enables the user to upload a photo or take a photo in order to get real-time and accurate descriptions of landmarks using Gemini Flash API.
- Rich historic, architectural, and cultural description to be easily read.
- Simulation 3D modeling of landmarks such that users are able to explore buildings from any direction.
- Offline availability of stored landmark information, facilitating usage in areas with poor connectivity.
- A scalable platform with possible future additions like AR/VR support, multilingual descriptions, and richer AI-facilitated interactions

---

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for Gemini Landmark Description App Enhancing Tourist Experience with AI

Key Points:

1. **Technical Requirements:**
   - **Frontend:** HTML, CSS, JavaScript for user interface.
   - **Backend:** Flask framework for handling API requests and responses.
   - **AI Integration:** Gemini API for landmark recognition and description generation.
2. **Functional Requirements:**
   - **Image Upload:** Users can upload images of landmarks for AI analysis.
   - **Landmark Recognition:** AI should accurately identify landmarks from images.

- **Description Generation:** Provide detailed landmark descriptions based on AI analysis.

3. **Constraints & Challenges:**
    - **Accuracy of AI Recognition:** Ensuring reliable landmark identification across diverse images**.**
    - **Latency Issues:** Managing API response time to deliver results efficiently.
    - **Internet Dependency:** Requires a stable internet connection for AI processing.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.

Key Points:

1. **System Architecture:**
    - The user enters a landmark name or uploads an image via the web interface.
    - The Flask backend processes the input and sends a query to the Google Gemini API to fetch the landmark description.
    - The API response is formatted and displayed in the frontend**.**

2. **User Flow:**
    - **Step 1:** The user provides a landmark name in a text field or uploads an image.
    - **Step 2:** Gemini API for the landmark description and Flask app processes the request and calls Google
    - **Step 3:** The app extracts and formats the description.

3. **UI/UX Considerations:**
    - Simple, intuitive design with clear input fields using HTML

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion.

Sprint 1 – Setup & Integration (Day 1)

- **(🔴 High Priority)** Set up the Flask environment and install dependencies.
- **(🔴 High Priority)** Integrate Google Gemini API for fetching landmark descriptions.
- **(🟡 Medium Priority)** Implement the basic UI with input fields for text and image upload.

Sprint 2 – Core Features, Debugging & Deployment (Day 2)

- **(🔴 High Priority)** Develop backend logic for processing text/image inputs and fetching API data.
- **(🔴 High Priority)** Implement text-to-speech (TTS) functionality for landmark descriptions.
- **(🔴 High Priority)** Debug API responses and ensure proper error handling.
- **(🟡 Medium Priority)** Optimize response formatting and UI presentation.
- **(🟡 Medium Priority)** Perform functional and performance testing.
- **(🟢 Low Priority)** Deploy the Flask app using a cloud platform (e.g., Streamlit Sharing, Heroku).
- **(🟢 Low Priority)** Prepare final documentation and demo presentation.

Phase-5: Project Development

Objective:

Implement core features of the Gemini Landmark Description App Enhancing Tourist Experience with AI

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** HTML, CSS (Bootstrap for styling)
    - **Backend:** Flask (Python)
    - **AI Integration:** Google Gemini API

2. **Development Process:**
    - Implement API key authentication and integrate Google Gemini API.
    - Develop logic to process user inputs (text/image) and fetch descriptions.

3. **Challenges & Fixes:**
    - **Challenge:** API response delays.
        - **Fix:** Implement caching for frequently queried landmarks**.**
    - **Challenge:** Handling API rate limits**.**
        - **Fix:** Optimize API calls to fetch only necessary data.
    - **Challenge:** UI responsiveness issues.
        - **Fix:** Use Bootstrap to ensure cross-device compatibility.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the Gemini Landmark Description App Enhancing Tourist Experience with AI works as expected.

Test case ID	Category	Test Scenario	Expected Output	Status
TC-001	Functional Testing	Query “Eiffel tower”	Accurate landmark description displayed	Passed
TC-002	Functional Testing	Upload an image of a landmark	Description retrieved	Passed
TC-003	Performance Testing	API response time under 500ms	API should return quickly	Needs optimization
TC-004	Bug fixes & Improvements	Handle incorrect API responses	Data accuracy should be improved	Fixed
TC-005	UI Testing	Ensure UI is responsive across devices	UI should work on desktop	Failed on MacOS
TC-006	Deployment Testing	Host the app using flask deployment	App should be accessible online	Deployed

---