Hackathon Project Phases Template

Project Title:

ProVision AI - Smart Image Captioning

Team Name:

Annotators AI

Team Members:

- **S. Harika**
- **G. Harshitha**
- **R. Harshitha**
- **V. Hasini**

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered image captioning and translation tool using Gemini Flash to help users generate captions, descriptions, and stories from images in multiple languages with text-to speech capabilities

Key Points:

Problem Statement:

1. Many users struggle with understanding and describing images, especially in different languages.
2. Artists, content creators, and researchers often need automated captions and descriptions for images.
3. Language barriers prevent seamless communication, making it hard for non-native speakers to access AI-generated content.

Proposed Solutions:

1. An AI-powered web application that generates image-based captions, summaries, and descriptions using Gemini Flash.
2. The app translates generated text into multiple languages and supports text-to-speech (TTS) for spoken output.
3. Users can interact with the AI using custom prompts (e.g., "Tell a story about this image in Telugu").

Target Users:

1. Content creators & bloggers → Need automated captions for images.
2. Students & researchers → Require image-based insights and multilingual translations.
3. People with visual impairments → Can listen to AI-generated descriptions via TTS.

Expected Outcome:

A fully functional AI-driven image captioning and translation app.
Users can upload images and receive descriptive captions in their preferred language.
The app will provide text-to-speech support for accessibility and improved engagement.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the AI-powered image captioning and translation app.

Key Points:

Technical Requirements:

Programming Language: Python
Backend: Google Gemini Flash API
Frontend: Streamlit Web Framework
Database: SQLite (for user authentication and session management)

Functional Requirements:

Ability to generate captions and descriptions from uploaded images using Gemini Flash API.
Support for multilingual translations using Google Translate API.
Text-to-speech (TTS) support for audio output of generated responses.
Provide a user-friendly UI to upload images, enter prompts, and select output languages.
Implement user authentication (Login/Signup) to personalize experience.

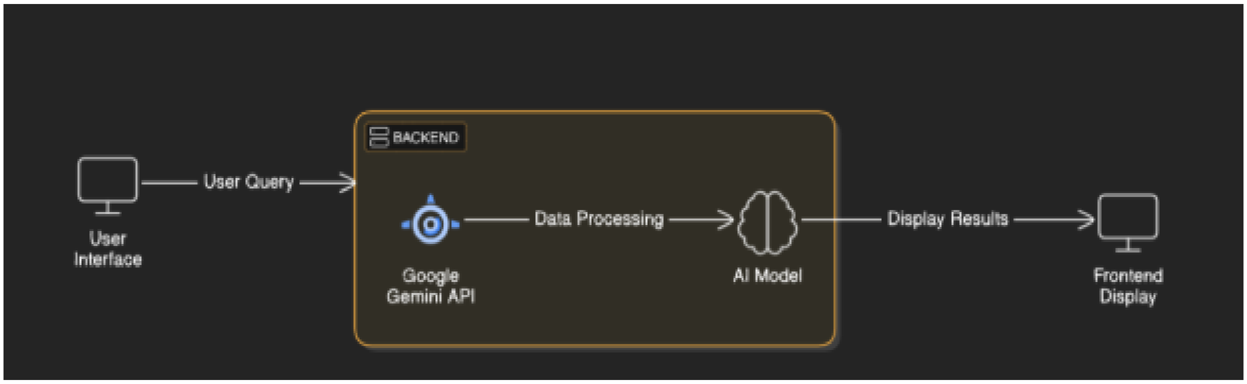
Constraints & Challenges:

Ensuring efficient API calls and managing potential rate limits.
Handling large image processing without slowing down performance.
Maintaining a smooth and intuitive UI using Streamlit.
Ensuring accurate translations across multiple languages.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the AI-powered image captioning and translation application.



Key Points:

System Architecture:

- The query and image are processed using Google Gemini Flash API.
- The AI model analyzes the image and generates a response.
- The response is translated into the selected language.
- If enabled, text-to-speech (TTS) generates an audio output.
- The frontend displays the image, AI-generated text, and audio output

User Flow:

Step 1: User uploads an image and enters a query (e.g., "Describe this image in Telugu").

Step 2: The backend calls the Gemini Flash API, passing the image and query.

Step 3: The AI processes the image and generates a caption/description.

Step 4: The response is translated based on the selected language.

Step 5: The final output is displayed on the UI, with an audio option if needed.

UI/UX Considerations:

- Minimalist, user-friendly interface for seamless navigation
- Dropdown selection for output language choices.
- Dark & Light mode for better accessibility.
- Responsive layout for mobile and desktop use

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks into sprints for efficient and structured completion.

Sprint	Task	Priority	Duration	Deadline	Assigned to	Dependencies	Expected Outcome
Sprint1	Environement set up and api integration	High	6 hours (Day1)	End of Day1	Member 2	API key,python, streamlit setup	API connection established & working
Sprint1	Frontend UI development	Medium	2 hours(Day 1)	End of Day1	Member 1	API response for matfinalized	Basic UI with input fields
Sprint 2	AI Image Analysis & Captioning	High	3 hours(Day 2)	Mid-Day 2	Member 3&4	API response UI elements ready	Image processing and AI generated caption
Sprint 2	Translation and TTS integration	High	1.5 hours(Day 2)	Mid-day 2	Member 1&4	API logs,UI inputs	Improved API Stability
Sprint 3	Testing and UI Enhancements	Medium	1.5 hours(Day 2)	Mid-Day 2	Member 3	API response,UI Layout Completed	Responsive UI,better user experience
Sprint 3	Final Presentation and Deployment	Low	1 hour(Day 2)	End of Day 2	Entire Team	Working Prototype	Demo ready project

Sprint Planning with Priorities

- **Sprint 1 – Setup & Integration (Day 1)**
- **High Priority**

Set up the environment (Python, Streamlit, required libraries).

Integrate Google Gemini Flash API for image processing and text generation.

Medium Priority

Develop a basic UI with an image uploader and text input field.

- **Sprint 2 – Core Features & Debugging (Day 2)**
- **High Priority**

Implement AI-based image analysis and captioning using Gemini API.

Integrate translation feature for multilingual output.

Enable text-to-speech (TTS) functionality for audio responses.

Debug API response issues and handle potential errors.

- **Sprint 3 – Testing, Enhancements & Deployment (Day 2)**
- **Medium Priority**

Test API responses for accuracy and ensure smooth user interaction.

Enhance UI/UX by adding dark/light mode, better layout, and responsiveness.

- **Low Priority**

Final demo preparation and deployment of the app.

Phase-5: Project Development

Objective:

Implement the core features of the AI Image Captioning & Translation App.

Key Points

- **Technology Stack Used:**
  - Frontend: Streamlit
  - Backend: Google Gemini Flash API
  - Programming Language: Python
- **Development Process:**
  - Implement API key authentication and integrate Gemini Flash API.
  - Develop AI-powered image analysis to generate captions.
  - Enable multilingual translation of generated captions.
  - Add text-to-speech (TTS) support for spoken output.
  - Optimize search and API calls to improve performance.

- **Challenges & Fixes:**
  - **Challenge:** Delayed API response times Fix
  - **Fix:**Implement caching to store frequently queried results.
  - **Challenge:**Limited API calls per minute
  - **Fix:**Optimize queries to fetch only necessary data.

Phase-6: Functional & Performance Testing

Objective:

Ensure the AI-powered image captioning & translation app functions as expected.

Test cases:

Test case ID	**Category**	**Test Scenario**	**Expected Outcome**	**Status**	**Tester**
TC-001	Functional Testing	Upload an image of a car with text query: ”Generate Caption”	AI Should generate an accurate caption	Passed	Member 3
TC-002	Functional Testing	Request translation of generated caption to telugu	The caption on should be translated correctly	Passed	Member 1
TC=003	Performance Testing	API response time under 500ms	API Should return results quickly	Needs optimization	Member 2
TC=004	Bug fixes & improvement	API caption outputs	Adapts should be more accurate	Fixed	Member 4
TC-005	UI Responsiveness	Ensure UI works on mobile and desktop	UI Should be fully responsive	Failed UI broken on mobile	Member 3
TC-006	Deployment testing	Host the app using streamlit sharing	App should be accessible online	Deployed	Devops

Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**