Hackathon Project Phases Template

Project Title:

Essay and assignment feedback tool using BERT and T5 models.

Team Name:

Text Crafters

Team Members:

- G Arpan Varma
- Marpaka Lokesh
- Lalaji Praneeth
- Gaddam Manideep
- C Bramarambika

Phase-1: Brainstorming & Ideation

Objective:

Leveraging BERT for understanding and analysing text and T5.

Key Points:

1. **Problem Statement:**
   - The traditional ways of professional writing and academic writing faces problems such as time consuming, inconsistent & subjective, limited accessibility, surface-

level corrections, lack of personalized learning. There is need for AI powered grammar correction system which is easy to understand and implemented

.

2. **Proposed Solution:**

   - An AI-powered application using **BERT and T5**.
   - The app offers **grammatical corrections tips** and **sentence formation** based on user preferences.

3. **Target Users:**

   - **Students**
   - **Teachers**
   - **Employees**

4. **Expected Outcome:**
   The AI will generate appropriate sentences that fit the narrative and also help dramatic expression with maximum accuracy.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Essay and Assignment feedback Model.

Key Points:

1. **Technical Requirements:**

   - Programming Language: **Python**
   - Backend: **Google Gemini API**
   - Frontend: **Gradio Web Framework**
   - Database: **Not required initially (API-based queries)**

2. **Functional Requirements:**

   - Ability to **fetch** using BERT AND T5 AI.
   - Display **specifications, reviews, and comparisons** in an intuitive UI.
   - Provide **real-time grammar correction tips** based on context.
   - Allow users to **search eco-friendly vehicles** based on emissions and incentives.

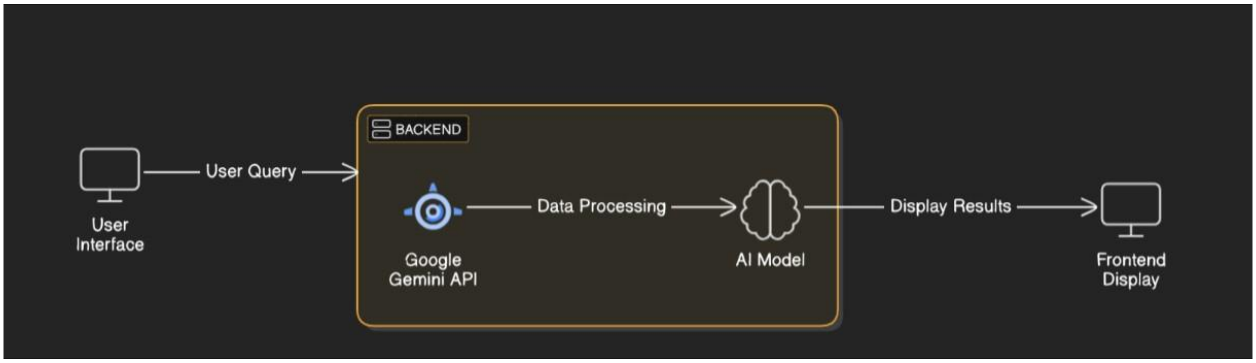
3. **Constraints & Challenges:**

- Ensuring real-time updates from **BERT AND T5**.
- Handling **API rate limits** and optimizing API calls.
- Providing a **smooth UI experience** with Gradio.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
    - User enters vehicle-related query via UI.
    - Query is processed using **Google Gemini API**
    - AI model fetches and processes the data.
    - The frontend displays **vehicle details, reviews, and comparisons**.
2. **User Flow:**
    - Step 1: User enters a query (e.g., "how do I form a sentence to apply for a leave").
    - Step 2: The backend **calls the Google Gemini API** to retrieve vehicle data.
    - Step 3: The app processes the data and **displays results** in an easy-to-read format.
3. **UI/UX Considerations:**
    - **Minimalist, user-friendly interface** for seamless navigation.
    - **Suggest Grammar Corrections along with sentence formations required by the user.**

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & API Integration	⍰ High	6 hours (Day 1)	End of Day 1	Member 1	Google API Key, Python, Gradio setup	API connection established & working
Sprint 1	Frontend UI Development	⍰ Medium	2 hours (Day 1)	End of Day 1	Member 2	API response format finalized	Basic UI with input fields
Sprint 2	Vehicle Search & Comparison	⍰ High	3 hours (Day 2)	Mid-Day 2	Member 1& 2	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	⍰ High	1.5 hours (Day 2)	Mid-Day 2	Member 1&4	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	⍰ Medium	1.5 hours (Day 2)	Mid-Day 2	Member 2& 3	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & 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** & install dependencies.
(⍰ High Priority) Integrate **Google Gemini API**.
(⍰ Medium Priority) Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

(□ High Priority) Implement **search & comparison functionalities**. **(□ High Priority)** Debug API issues & handle **errors in queries**.

Sprint 3 – Testing, Enhancements & Submission (Day 2)

(□ Medium Priority) Google Gemini API I responses, refine UI, & fix UI bugs. **(□ Low Priority)** Final **demo preparation & deployment**.

Phase-5: Project Development

Objective:

Developing and Leveraging BERT for understanding and analysing text and T5.

Key Points:

1. **Technology Stack Used:**
   - **Frontend:** gradio
   - **Backend: Google Gemini API**
   - **Programming Language:** Python
2. **Development Process:**
   - Implement **API key authentication** and **Gemini API integration**.
   - Develop **grammar and sentence correction logic**.
   - Optimize **search queries for performance and relevance**.
3. **Challenges & Fixes:**
   - **Challenge:** Delayed API response times.
     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 that the Essay and Feedback Model works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "Best budget cars under ₹10 lakh"	Relevant budget cars should be displayed.	⍰ Passed	Tester 1
TC-002	Functional Testing	Query "Motorcycle maintenance tips for winter"	Seasonal tips should be provided.	⍰ Passed	Tester 2
TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	⍰ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	⍰ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	⍰ Failed - UI broken on mobile	Tester 2
TC-006	Deployment Testing	Host the app using Gradio 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**