Hackathon Project Phases Template for the **AutoSage App** project.

Hackathon Project Phases Template

Project Title:

AI Personalised Email Generator

Team Name:

PANDT

Team Members:

- Kumbham Dinesh
- Anshuman Gutti
- K.Prathyusha
- Nithya Sai
- Khetavath Sai Tanishka

Phase-1: Brainstorming & Ideation

Objective:

The objective of an AI-Personalized Gmail Generator is to automate and customize email creation based on user preferences, tone, and context.

Key Points:

1. **Problem Statement:**
   - In today's fast-paced digital world, professionals and businesses struggle with **email overload**, spending excessive time crafting, responding to, and managing emails. Generic or poorly written emails can lead to **miscommunication, lost opportunities, and reduced productivity**
2. **Proposed Solution:**
   - Instantly generates well-structured emails based on minimal input.
   - Improves response rates and engagement with optimized messaging.
   - Reduces manual email drafting efforts, allowing users to focus on high-value tasks.

3. **Target Users:**

   - **Business Professionals & Executives**.
   - Sales & Marketing Teams.
   - Customer Support & Service Teams.

4. **Expected Outcome:**

   - Time Savings & Efficiency Boost.
   - Enhanced Personalization & Engagement.
   - Increased Productivity & Workflow Optimization

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the GEN-eM app.

Key Points:

1. **Technical Requirements:**

   - Programming Language: **HTML,CSS and JAVA SCRIPT**
   - Backend: **JAVA SCRIPT**
   - Frontend: **HTML, CSS**

2. **Functional Requirements:**

   - Email Drafting & Personalization.
   - Smart Reply.
   - Grammar, Spelling & Clarity Enhancement.

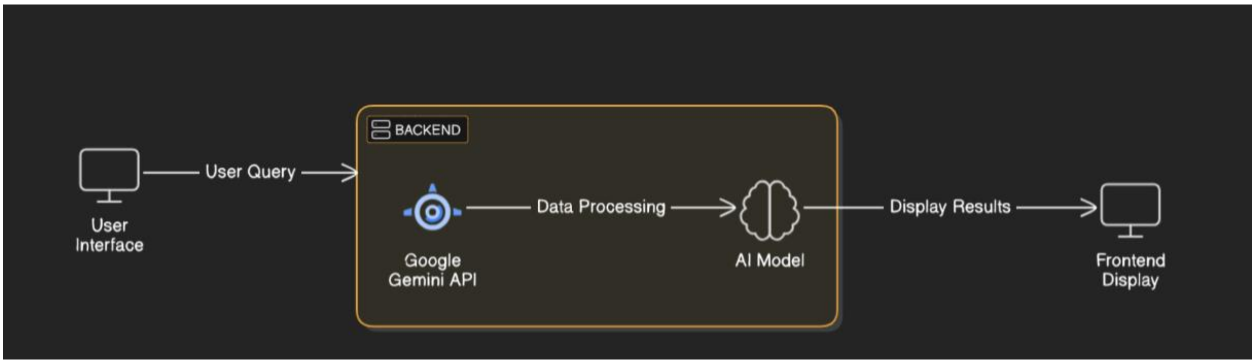
3. **Constraints & Challenges:**

   - **Real-Time Processing Speed** – AI must generate emails within **2-3 seconds** to ensure a seamless user experience, requiring optimized processing
   - **Ease of Use & Onboarding –** Users should be able to adopt the tool with minimal learning curve; complex interfaces could reduce adoption rates.
   - **Over-Reliance on AI** – Users might become overly dependent on AI-generated emails, leading to reduced personal engagement.

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 **HuggingFace and Brevo**.
   - AI model fetches and processes the data.
   - The frontend displays **Email**.

2. **User Flow:**

   - Step 1: User enters a query (e.g., Senders name,Senders Email,Receivers name,etc…).
   - Step 2: The backend **calls the Gemini Flash API** to retrieve given 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.

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	8 hours (Day 1)	End of Day 1	Kumbham Dinesh & Nithya Sai	Javascript,Hugging Face	API connection established & working
Sprint 1	Frontend UI Development	□ Medium	2 hours (Day 1)	End of Day 1	K Prathyusha	HTML,CSS	Basic UI with input fields

Sprint 2	Error Handling & Debugging	● High	1.5 hours (Day 2)	Mid-Day 2	Kumbham Dinesh	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	□ Medium	1.5 hours (Day 2)	Mid-Day 2	Nithya Sai	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	□ Low	1 hour (Day 2)	End of Day 2	Anshuman Gutti & Khetavath Sai Tanishka	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 **HuggingFace and Brevo**.
(□ 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) Test API responses, refine UI, & fix UI bugs.
(□ Low Priority) Final **demo preparation & deployment**.

Phase-5: Project Development

Objective:

Implement core features of the AutoSage App.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** HTML, CSS
    - **Backend:** Java Script
    - **Programming Language:** HTML,CSS,JAVA SCRIPT
2. **Development Process:**
    - Implement **API key authentication** and **Hugging Face.**

   - **Natural Language Processing (NLP) Model –** Train an AI model (e.g., GPT-based, fine-tuned BERT) for email writing.
   - **Browser Extension & Web App –** Develop a lightweight, **Gmail-embedded UI** for email generation.
3. **Challenges & Fixes:**

   - **Challenge:** Delayed API response times.
     Fix: Implement **caching** to store frequently queried results.

Phase-6: Functional & Performance Testing

Objective:😊

Ensure that the AutoSage App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Google Job application to khetavathsaitanishka24@gmail.com	Relavanet mail sent to khetavathsaitanishka24@gmail.com	☑ Passed	Khetavath Sai Tanishka
TC-002	Functional Testing	Leave Application for 2 days due to fever	Relavanet mail sent to anshuman.gutti@gmail.com	☑ Passed	Gutti Anshuman
TC-003	Bug Fixes & Improvements	Integrate buttons	Goes to the next page	☑ Fixed	Kumbham Dinesh & Nithya Sai
TC-004	Final Validation	Ensure UI is responsive across devices.	UI should work on desktop.	☑ Passed	Prathyusha

---

Final Submission

1. **Project Report Based on the templates**
2. **GitHub/Code Repository Link**

3. **Presentation**