### **Hackathon Project Phases Template** for the AI Personalized Email Generator

# **Hackathon Project Phases Template**

## **Project Title:**

### AI Personalized Email Generator

## **Team Name:**

Team Karthik

## **Team Members:**

* K. Karthikeya Sharma
* G. Hari Bala phanindra Kumar
* Md. Khader
* G. Jayanth

## **Phase-1: Brainstorming & Ideation**

### **Objective:**

AI-powered tool that helps businesses draft professional and engaging emails effortlessly using Generative AI, making the process more personalized and efficient.

### **Key Points:**

1. **Problem Statement:**
   * Businesses and professionals often spend a lot of time crafting similar emails repeatedly.
   * Finding the right balance between consistency and personalization can be challenging.
   * While existing tools offer predictive text, they often fall short when it comes to real-time customization.
2. **Proposed Solution:**
   * An AI-powered tool that generates personalized emails based on the user input.
   * Features dynamic tone adaption (formal, informal, persuasive, etc.).
   * AI suggestions for clarity, grammar, and message impact
3. **Target Users:**
   * Business professionals, customer support teams, and marketers.
   * Small businesses and startups looking for efficient communication solutions.
   * Non-native speakers needing assistance with professional email writing.
4. **Expected Outcome:**
   * A functional AI-powered email generator that enhances business communication and engagement.

## **Phase-2: Requirement Analysis**

### **Objective:**

Define the technical and functional requirements for the AI personalized Email Generator.

### **Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: **Google Gemini Flash API, Hugging Face, Google API, Python Flask**
   * Frontend: **React, Material UI, JavaScript (JS)**
   * Database: **Fire Base**
2. **Functional Requirements:**
   * User input fields for recipient details, email context, and tone selection, Humanizing the Response.
   * AI-generated email suggestions with grammar and clarity enhancements.
   * Based on the user input, the application redirects to the GMAIL and paste’s the text in Compose.
   * User-friendly interface for editing, regenerating and copying emails.
3. **Constraints & Challenges:**
   * Ensuring low latency for **AI** responses.
   * Handling **API rate limits** and optimizing API calls.
   * Providing a **seamless UI experience.**

## **Phase-3: Project Design**

### **Objective:**

Develop the system architecture and user flow for the AI personalized Email Generator.

### **Key Points:**

1. **System Architecture:**
   * **User Input Layer (Frontend):** Users enter recipient details, email context, and Tone preferences
   * **API Request Handler (Backend-Flask Server):** processes user inputs and interacts with the AI model.
   * **AI Email Generation Module ( SmolAgents & AI model Processing) :** AI generates personalized email content dynamically.
   * Output (Frontend Display & copying): Displays the generated email, allowing users to modify or regenerate it.
2. **User Flow:**

* Step 1: User enters email details (recipient, tone, content context, etc.).
* Step 2: Backend processes inputs and sends them to the AI model.
* Step 3: AI model generates a personalized email.
* Step 4: The email is displayed, with options for modification or regeneration.

**UI/UX Considerations:**

* Clean and user-friendly interface.
* Options for different tones and styles.
* Dark & light mode support.

## 

## **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 | K. karthik & G. Phanindra | Google API Key, Python, Flask setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | G. Jayanth & Md. Khader | API response format finalized | Basic UI with input fields |
| Sprint 2 | AI Email Generation Implementation | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | K. karthik & G. Phanindra | Backend setup completed | AI-generated emails displayed |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Entire Team | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Entire Team | 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 |

### 

## **Phase-5: Project Development**

### **Objective:**

**Objective:** Implement core features of the AI Personalized Email Generator.

### **Key Points:**

**Technology Stack Used:**

* **Frontend:** React, Material UI
* **Backend:** Google Gemini Flash API, Flask
* **Programming Language:** Python

**Development Process:**

* + Implement API key authentication and Gemini API integration.
  + Develop AI-based personalized email generation logic.
  + Optimize UI for better usability.

**Challenges & Fixes:**

* **Challenge:** Delayed API response times.
  + **Fix:** Implement caching to store frequently queried results.
* **Challenge:** API rate limits.
  + **Fix:** Optimize queries to fetch only necessary data.

## **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 | User enters email details | AI generates a personalized email | ✅ Passed | K. Karthik & G. Phanindra |
| TC-002 | Functional Testing | Modify generated email | UI allows seamless edits | ✅ Passed | G. Jayanth & Md. Khader |
| TC-003 | Performance Testing | API response time under 500ms | Fast AI-generated responses | ✅ Passed | K. Karthik & G. Phanindra |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses. | AI suggestions improved | ✅ Fixed | Entire Team |
| TC-005 | UI Validation | |  |  | | --- | --- | |  | Ensure responsiveness | | UI should work on desktop. | ✅ Passed | Entire Team |
| TC-006 | Deployment Testing | Opens GMAIL & Paste the context into Mail Composer | On click, Should re-direct to Gmail and paste the Generated Mail | ✅ Passed | G. Phanindra Kumar |

## **Final Submission**

* Project Report
* GitHub/Code Repository Link
* Final Presentation