Hackathon Project Phases Template

Project Title:

AI Personalized Email Generator

Team Name: Femini AI

Team Members:

- Shruthi Kodityala
- Srigadha Keerthi
- Sreeram Raveena
- Sreeja Polsani
- Shinde Rutuja

Phase-1: Brainstorming & Ideation

Objective:

To develop an AI-powered personalized email generator using advanced natural language processing and machine learning techniques, enabling users to efficiently create context-aware, personalized, and professionally structured emails with customizable tone, intent, and language support.

Key Points:

1. **Problem Statement:**

   - AI Personalized Email Generator is an innovative project designed to revolutionize email communication by automating the process of drafting personalized emails.

2. This system uses state-of-the-art language models to generate tailored email content based on user inputs, such as the recipient's name, event details, and special instructions.

3. **Proposed Solution:**

   Context-Aware Personalization & Dynamic Tone Adjustment

   - AI analyzes recipient details, previous interactions, and user preferences to craft highly personalized emails
   - Users can adjust the tone and emotion (formal, friendly, persuasive, etc.) using real-time AI suggestions.

   Bullet-to-Email Conversion & Smart Auto-Refinement

   - Converts key points or rough notes into a professionally structured email instantly.
   - AI suggests alternative drafts and refines grammar, clarity, and conciseness automatically.

4. **Target Users:**

   - Corporate Professionals & Business Executives – For client communication, meetings, and follow-ups.
   - Sales & Marketing Teams – For outreach, promotions, and personalized engagement.
   - Students & Job Seekers – For resumes, cover letters, and formal emails.

5. **Expected Outcome:**

   - The AI Personalized Email Generator will produce well-structured, context-aware emails with customizable tone, and can send to multiple receiptents saving users time and effort. It will seamlessly integrate with email platforms and CRM tools, ensuring efficiency and personalization in professional communication.

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
   - Framework: Streamlit

2. **API & credentials :**
   - Hugging Face API key
   - Gmail credentials
   - Environment variable .env

3. **Functional Requirements:**

   - Ability to generate personalized emails based on user input, recipient details, and context.
   - Provide tone adjustment options (formal, casual, persuasive, apologetic) for customization.
   - Offer real-time grammar and clarity suggestions to refine email content.
   - Convert bullet points or rough notes into well-structured, professional emails.
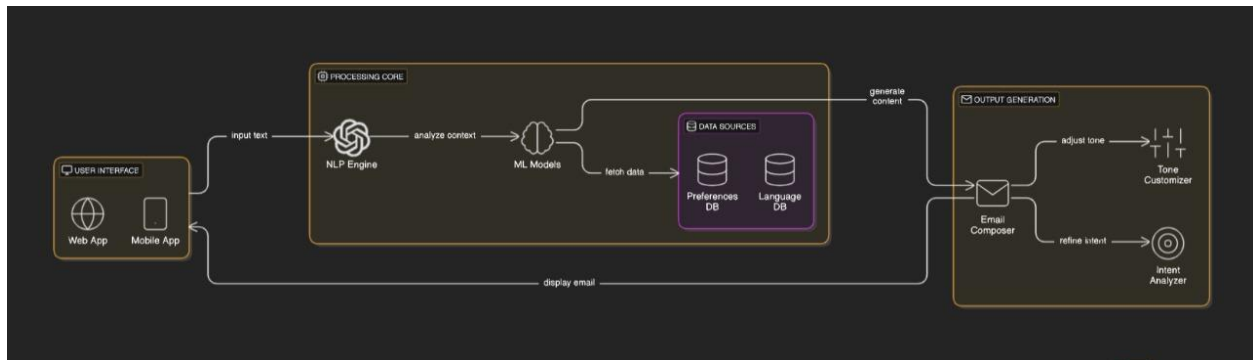
4. **Constraints & Challenges:**

   - Ensuring real-time AI-generated email responses with minimal latency.
   - Handling API rate limits and optimizing calls to AI models (e.g., GPT, Hugging Face).
   - Providing a smooth UI experience with Streamlit.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.

Key Points:

1. **System Architecture:**
   - User interface for inputting email details, generating, previewing, and sending emails.
   - Hugging Face API: Generates email content based on the user's input.
   - SMTP Email Sending: Sends personalized emails using Gmail SMTP.

2. **User Flow:**

   - Step 1: Input Details: Enter email prompt, tone, sender email, recipient emails, names, and subject.
   - Step 2: Generate Email: Click "Generate Email" to get AI-generated content from Hugging Face.
   - Step 3: The app processes the data and **displays results** in an easy-to-read format.
   - Step 4: Send Email: Validate inputs and send the email using Gmail SMTP.
   - Step 5: Test Email Login Verify Gmail login and connection.

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	2 hours (Day 1)	End of Day 1	Shurthi & Raveena	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	🟡 Medium	1 hours (Day 1)	End of Day 1	Rutuja	API response format finalized	Basic UI with input fields
Sprint 2	Email Personalization Module	🔴 High	2 hours (Day 2)	Mid-Day 2	Keerthi& Sreeja	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	Shruthi	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Raveena	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)** Test API responses, refine UI, & fix UI bugs.
- (🟢 **Low Priority)** Final **demo preparation & deployment**.

Phase-5: Project Development

Objective:

Implement core features of the AI Personalized Email Detector.

Key Points:

1. **Technology Stack Used:**

   - **Frontend:** Streamlit
   - **Backend:** Google Gemini Flash API
   - **Programming Language:** Python

2. **Development Process:**

   - Implement **API key authentication** and **Gemini API integration**.
   - Optimize **search queries for performance and relevance**.

3. **Challenges & Fixes:**

   - **Challenge:** Generating the body of email.
     Fix: Checking the API keys.
   - **Challenge:** Sending email to multiple recipient.
     Fix: checking the receiver's mail address and recipient name.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the AI Personalized Email Generator works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Generate a formal business email for meeting request	Relevant budget cars should be displayed.	✅ Passed	Sreeja
TC-002	Functional Testing	Generate an informal thank-you email	Seasonal tips should be provided.	✅ Passed	Rutuja
TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	⚠ Needs Optimization	Keerthi
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Raveena
TC-005	Deployment Testing	Host the app using Streamlit in VS Code	App should be accessible online.	🚀 Deployed	Shruthi

Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)** https://drive.google.com/file/d/1HvBaH4T6bUuB6-_i58E09bX7Ga_kgcQZ/view?usp=sharing
3. **GitHub/Code Repository Link** https://github.com/ShruthiKodityala-5n6/Team-Femini-AI/tree/main
4. **Presentation**