**Project Title:**

PERSONALIZED EMAIL GENERATORS

**Team Name :**

AI MAILCRAFTS

**Team Members:**

G.Tejashwini

G.Rochisna

T.Chidwila

U.Nikhitha

**Phase-1: Brainstorming & Ideation**

**Objective:**

To develop an AI-powered **Personalized Email Generator** that dynamically crafts tailored emails based on user inputs, intent, and context. Leveraging **NLP and machine learning**, the system ensures emails are structured, engaging, and contextually relevant for professional, marketing, or personal communication.

* **Key Points:**

**1. Problem Statement:**

A **Personalized Email Generator** is an AI-powered application that automates the creation of **tailored, context-aware emails** based on user intent and preferences. By leveraging **NLP and machine learning**, it ensures emails are **efficient, engaging, and relevant**, saving time while enhancing communication across professional, marketing, and personal domains.

**2. Proposed Solution**

To develop an **AI-driven email generation system** that utilizes **NLP and machine learning** to create **customized, well-structured emails** based on user intent, tone, and context. The system will automate email composition, ensuring **efficiency, personalization, and consistency**, reducing manual effort while enhancing communication effectiveness.

**3. Target users:**

**Marketing & Sales Teams** – Generate personalized promotional emails, newsletters, and customer engagement content.

**Customer Support & Service Teams** – Craft quick, context-aware responses for customer inquiries.

**Students & Job Seekers** – Create professional emails for networking, job applications, and academic inquiries.

**4. Expected Outcome:**

Ensuring AI-driven adaptive learning remains accurate and personalized. Handling real-time updates and dynamic schedule adjustments efficiently. Optimizing performance within Streamlit for a seamless user experience

**Phase-2: Requirement Analysis**

**Objective:**

The objective of the **requirement analysis** for the **Personalized Email Generator** is to systematically identify, define, and document the functional and technical needs of the system. This process ensures that the AI-powered email generator meets user expectations by providing **context-aware, structured, and engaging email content**..

**1.Technical Requirements :**

* **Programming Language:** Python
* **Backend:** Pre-Trained language models from Hugging face like GPT-2 or BART
* **Frontend:** Stream lit Web Framework
* **Database:** MONGODB

**2.Functional Requirements:**

**Personalized Email Generation:** Create context-aware emails based on user input, intent, and recipient details.

**Tone & Style Adaptation:** Customize email tone (formal, casual, persuasive) to match the user’s communication needs.

**AI-Based Content Suggestions:** Provide subject lines, key points, and optimized email structures using NLP.

**Smart Editing & Refinement:** Allow users to modify AI-generated emails with grammar suggestions and content improvements.

**Template Library:** Offer predefined email templates for common use cases (job applications, client outreach, follow-ups).

**3.Constraints & Challenges:**

Ensuring AI-generated content is **coherent, relevant, and contextually appropriate**.

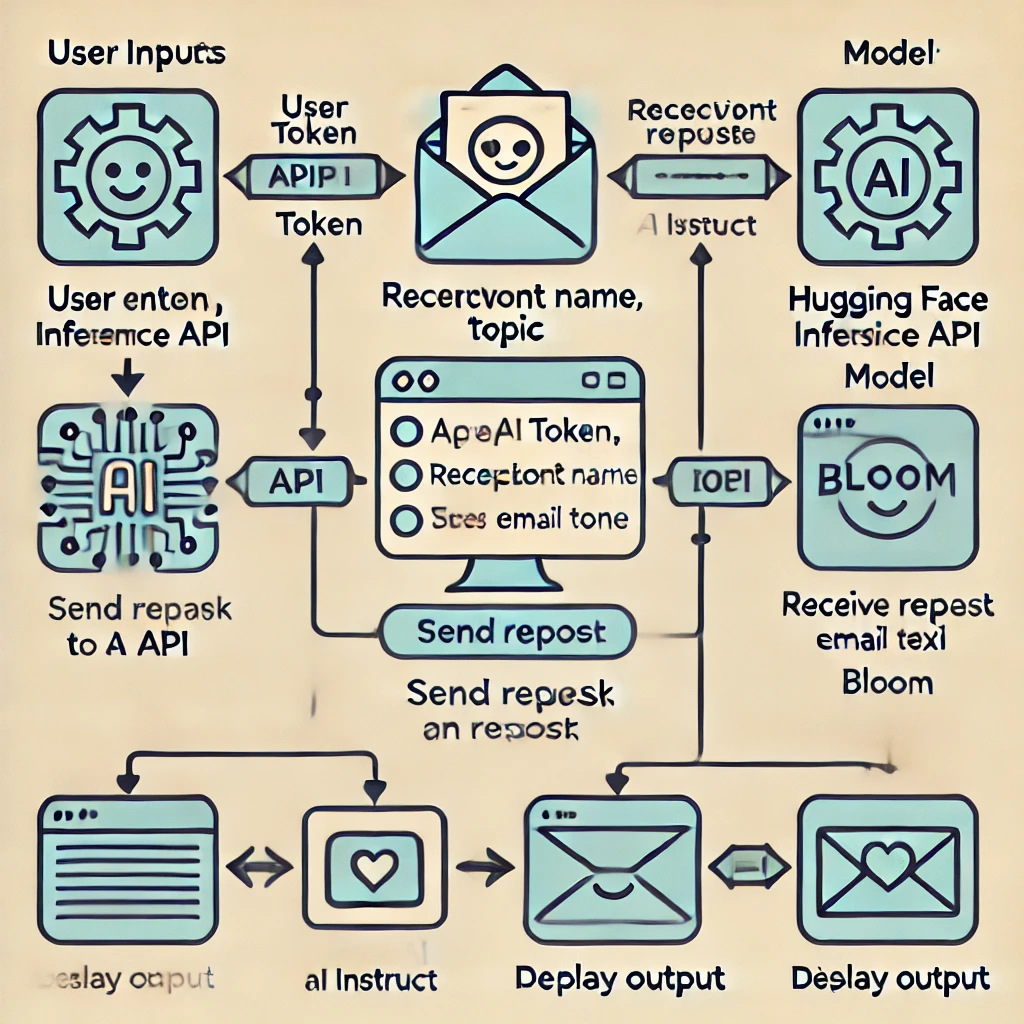
Handling **real-time personalization** based on user preferences and historical interactions.

Optimizing **performance and response time** within Streamlit for a seamless user experience.

**Phase-3: Project Design**

**Objective:**

**Develop the architecture and user flow of the application**



**Phase-4:ProjectPlanning(Agile Development)**

**Objective:**

Break down the tasks using Agile methodologies to ensure effective sprint planning, task allocation, and timelines

**Key Points:**

**Sprint Planning**

Divide the project into **small, manageable sprints** for iterative development.

Prioritize **core features** like email generation, tone adaptation, and UI/UX improvements.

**Task Allocation**

**Frontend Team:** Build the **Stream lit UI** for user interaction.

**Backend Team:** Develop **Flask/Fast API APIs** for AI-powered email generation.

**AI/ML Team:** Fine-tune **NLP models** for personalized email output.

**Database Team:** Implement **MongoDB** for storing user preferences.

**Testing Team:** Validate system performance, accuracy, and scalability.

**Timelines and Mile Stones**

On **Day 1 (Morning)**, the team sets up the **development environment** and integrates the AI model. By **Day 1 (Afternoon)**, the core **email generation functionality** is developed and connected to the frontend. In **Day 1 (Evening)**, **tone customization and user input handling** are implemented.

On **Day 2 (Morning)**, performance is **optimized, UI is refined, and bug fixes** are applied. By **Day 2 (Afternoon)**, the final **testing and validation** are completed, ensuring the system is fully functional and ready for use.

**Phase 5 : Project Development**

**OBJECTIVE**

Develop and integrate the AI-powered **Personalized Email Generator** ensuring seamless functionality, accuracy, and efficiency

**Technology Stack:**

**Language:** Python

**Frontend:** Stream lit

**Backend:** Flask/Fast API

**AI Model:** GPT-3, BART, T5 (Hugging Face)

**Database:** MongoDB

**APIs:** OpenAI API, SMTP

**Development Process**:

**Set up environment** and dependencies.

Implement **AI-powered email generation**.

Build **Stream lit frontend** & Flask/Fast API backend.

Integrate **database for user preferences**.

Enable **email customization & tone adaptation**.

Test, optimize, and **deploy** the system .

**Challenges & Fixes:**

**Personalization Issues →** Used user-specific data.

**Slow Response Time →** Optimized API calls & models.

**UI/UX Complexity →** Improved design & responsiveness.

**Grammar & Tone Issues →** Integrated grammar-checking API.

**Phase6: Functional & Performance Testing**

**Objective:**

Develop an AI-powered **Personalized Email Generator** leveraging **NLP and machine learning** to create **context-aware, structured, and engaging emails** for various communication needs.

**Test Cases Executed:**

**Input Customization:** Validate email generation based on user input (tone, purpose, recipient).

**AI Model Accuracy:** Ensure relevance, coherence, and grammatical correctness of generated emails.

**Batch Generation:** Test the system's ability to create multiple emails efficiently.

Performance: Measure response time, scalability, and system stability under load

**Bug Fixes & Improvements:**

**Bug Fixes:**

Resolved incorrect tone adaptation and personalization issues.

Fixed API response delays and email formatting inconsistencies.

**Improvements:**

Enhanced **UI/UX** for intuitive customization.

Optimized AI-generated email structures for clarity and engagement.

Improved response time and system scalability

.

**Final Validation:**

**Meets Requirements:** Generates high-quality, personalized emails tailored to user inputs.

**Performance:** Fast, scalable, and efficient under varying loads.

**Deployment:**

**Hosting:** Deployed on **AWS EC2** (for scalability) and **Heroku**.