Hackathon Project Phases

# Project Title:

**MailMate:** Personalized Email Generator

# Team Name:

*Status200*

# Team Members:

* Gade Ruthvik Reddy
* Thammi Suraj
* C. Pradeep

# Phase-1: Brainstorming & Ideation

## Objective:

* To create an AI system to automate the generation of the emails by taking the necessary information and providing help with the sending of the mail and removing the stress of writing the emails

## Key Points:

1. **Problem Statement:**

AI Personalized Email Generator is an innovative project designed to revolutionize email communication by automating the process of drafting emails. This system uses state-of-the-art language models to generate tailored email content based on user inputs, such as recipient’s name , event details and special instructions.

1. **Proposed Solution:**
   * Creating an AI system that uses the GenAI model to generate email based on the give inputs from the user and helps to create new templates or use to the predefined to write email removing the stress of templating. The AI also automates the process of sending the mail by SMPT-SSL by authenticating the user details.
2. **Target Users:**
   * The AI system targets Business professionals, Students, Researchers, Custom service representatives, Job seekers and Marketing teams. The AI system removes the stress from these target users who write many emails as a part of their work like Students who need to write many permissions letter can automate it using the AI system.
3. **Expected Outcome:** 
   * The expected outcome of this AI system is to provide the Business professionals, Students, Researchers, Custom service representatives, Job seekers and Marketing teams with overcome the hassle of writing humongous mails. To provide these target users with a way to save time from writing mails. Templates reducing the work of rewriting of the mail for user.

# Phase-2: Requirement Analysis

## Objective:

* + The requirement analysis aims to understand what stakeholders need from the AI system, define its essential features, set technical and performance standards, ensure it aligns with current email trends, and create a detailed plan to guide the development process. This helps ensure the generated email meets user demands effectively and efficiently.

## Key Points:

1. **Technical Requirements:**

* **Programming Language:**Python
* **Frontend:** Streamlit
* **Backend:** Python
* **GenAi:**Gemini-2.0-flash
* **SMTP-SSL**

1. **Functional Requirements:**

* Users enter a subject, and the AI generates an email body.
* Option to edit the generated content before sending.
* Authentication of the user mail using app password and sending email by SMTP-SSL
* Template to store a format to use a mail for multiple similar mails

1. **Constraints & Challenges:**

* AI generated content relevance.
* Ensuring SMPT security for email sending
* Storing of the templates

# Phase-3: Project Design

## Objective:

## Create the architecture and user flow

## Key Points:

1. **System Architecture Diagram:**
   * User → Frontend (Streamlit) → Backend (Python) → AI Model → SMTP Server → Recipient
2. **User Flow:** 
   * User saves his details → User inputs subject → AI generates email → User reviews/edit content → Can save template → Sends email via SMTP → Email delivered
   * User selects template → Makes changes → User reviews → Enters recipient details →

Sends email via SMTP → Email delivered

1. **UI/UX Considerations:**
   * Clean and simple email editor
   * Easy authentication process

# 

# Phase-4: Project Planning (Agile Methodologies)

## Objective:

* + Break down the tasks using Agile methodologies.

## Key Points:

1. **Sprint Planning:**

* Sprint 1: Setup backend API and AI integration
* Sprint 2: Develop frontend UI and authentication
* Sprint 3: Implement email sending and SMTP integration
* Sprint 4: Testing and bug fixes

1. **Tasks:**

* Backend (AI Integration, Email API)
* Frontend (Streamlit)
* SMTP & Authentication
* Testing & Deployment

1. **Timeline & Milestones:**

* Day 1: AI model integration
* Day 1: Email generation and UI development
* Day 1: SMTP authentication & email sending
* Day 2: Testing, bug fixes, and final deployment

# Phase-5: Project Development

## Objective:

* + Code the project and integrate components.

## Key Points:

1. **Technology Stack Used:**

* Python (Backend)
* Streamlit (Frontend)
* Gemini-2.0-flash API for AI email generation
* SMTP-SSL for sending the mail

1. **Development Process:**

* Step 1: Set up AI email generation with OpenAI API
* Step 2: Develop backend for handling user inputs and SMTP authentication
* Step 3: Create the frontend UI for email generation and editing
* Step 4: Implement email sending via SMTP
* Step 5: Testing and debugging

1. **Challenges & Fixes:**
   * **SMTP Security:** Use OAuth authentication instead of password storage.
   * **Email Relevance:** Fine-tune AI prompts to generate professional emails.

# Phase-6: Functional & Performance Testing

## Objective:

* + Ensure the project works as expected.

## Key Points:

1. **Test Cases Executed:** (List the scenarios tested)
   * AI generates proper email text based on various subjects.
   * Users can edit and modify the content before sending.
   * Email delivery is successful using SMTP.
   * Authentication is secure with no password leaks.
2. **Bug Fixes & Improvements:** (Mention fixes made)
   * Fixed formatting issues in AI-generated emails.
   * Improved security in SMTP authentication.
   * Optimized API response time for better performance.
3. **Final Validation:** (Does the project meet the initial requirements?)
   * AI email generation accuracy > 90%
   * Secure email delivery without SMTP authentication errors
   * No UI/UX usability issues
4. **Deployment:** (Hosting details or final demo link)
   * <https://mailmate7.streamlit.app>

**Github:** <https://github.com/suraj719/MailMate>

**Deployed link:** <https://mailmate7.streamlit.app>

**Demo video:** <https://youtu.be/dGSK_Apgkbk>