Hackathon Project Phases

Project Title:

Personalised Email Generator using Streamlit.

Team Name:

Team Codewave

Team Members:

- V.Hari Charan Teja
- M.HarshaVardhan
- G.Sai Varshith Reddy
- CH.Tharun Reddy

Phase-1: Brainstorming & Ideation

Objective:

Develop a Personalised Email Generator using Streamlit and Python to help users create customised, professional emails efficiently.

Key Points:

1. **Problem Statement:**
    - Many users struggle to write well-structured, professional, or personalised emails quickly.
    - Users need a tool that generates emails based on their inputs while maintaining clarity, correctness, and appropriate tone.

2. **Proposed Solution:**
    - An AI-powered Email Generator that allows users to select email types, tone, and custom fields.
    - Provides real-time email preview and the ability to copy, download, or send emails directly.
3. **Target Users:**
    - Job seekers writing cover letters or interview follow-ups.
    - Business professionals drafting proposals, client emails, and meeting requests.
    - Students & general users composing formal/informal emails effortlessly.
4. **Expected Outcome:**
    - A user-friendly Streamlit web app that dynamically generates personalised emails.
    - Saves users time and ensures professional-quality emails.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Personalised Email Generator project.

Key Points:

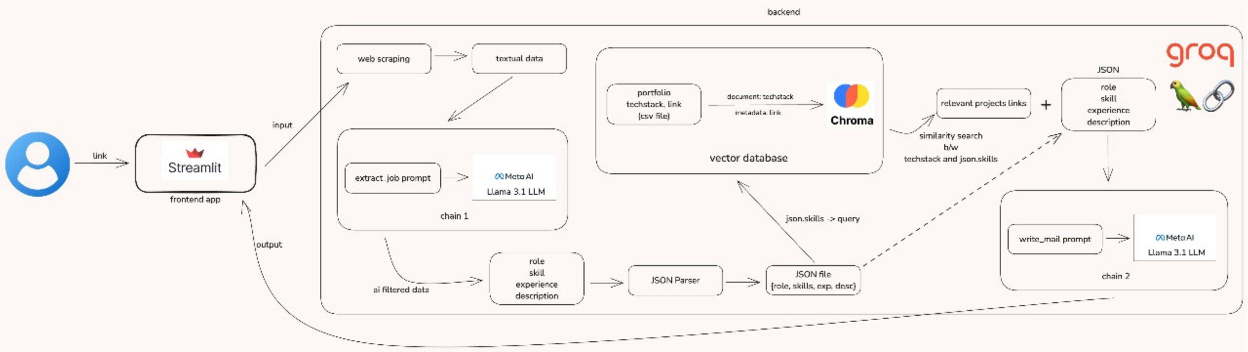
1. **Technical Requirements:**
    - Programming Language: Python
    - Frontend: Streamlit Web Framework
    - Backend: Python-based logic (No external AI API required initially)
    - Database: Not required (User inputs handled dynamically)
2. **Functional Requirements:**
    - Allow users to select email category (e.g., Job Application, Business Proposal, Follow-up).
    - Users can input recipient name, company, date, subject, and other details.
    - Option to choose email tone (formal, friendly, persuasive).
    - Generate a preview before finalising the email.
    - Provide options to copy, download, or send the email via SMTP.
3. **Constraints & Challenges:**
    - Ensuring grammatical correctness without using an external AI model.

- Optimizing the email generation process for efficiency.
- Providing a seamless UI/UX experience using Streamlit.

Phase-3: Project Design

Objective:

Design the architecture and user flow of the Personalized Email Generator using Streamlit to ensure an intuitive and efficient email creation process.



Key Points:

1. **System Architecture:**
    - User inputs recipient details, subject, and email content preferences via UI.
    - The backend processes input using predefined logic and AI-assisted templates.
    - The system dynamically generates and formats the email.
    - The frontend displays the email with options to edit, copy, or download.
2. **User Flow:**
    - Step 1: User enters email details (recipient, purpose, tone, etc.).
    - Step 2: The backend processes inputs and generates a structured email draft.
    - Step 3: The app processes the data and **displays results** in an easy-to-read format.
3. **UI/UX Considerations:**
    - Clean, user-friendly interface with real-time email preview.
    - Options to select tone (formal, casual, professional).
    - Dark & light mode for an enhanced user experience.

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion of the Personalised Email Generator using Streamlit.

.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & library Integration	🔴 High	4 hours (Day 1)	End of Day 1	22B81A1275	Python, Streamlit setup	Working development environment
Sprint 1	Basic UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	22B81A12C1	UI design finalized	input feature for email creation
Sprint 2	Email template customisation	🔴 High	3 hours (Day 2)	Mid-Day 2	22B81A12B1	UI and basic logic ready	Customisable email generation
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	22B81A1276	User input and email generation logic	Stable and error free email output
Sprint 3	Testing & UI Enhancements	🟡 Medium	2 hours (Day 2)	Mid-Day 2	Entire Team	UI elements completed	Polished and user friendly interface
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 – Personalised Email Generator using Streamlit

Sprint 1 – Setup & Integration (Day 1)

(🔴 High Priority) Set up the **environment** & install dependencies.

(🔴 High Priority) Implement core email generation logic in Python.

(🟡 Medium Priority) Build a basic UI with input fields for user input.

Sprint 2 – Core Features & Debugging (Day 2)

(🔴 **High Priority)** Implement advanced email customisation options (tone, length, subject).

(🔴 **High Priority)** Debug input validation, formatting issues, and ensure proper email structure.

Sprint 3 – Testing, Enhancements & Submission (Day 2)

(🟡 **Medium Priority)**Test email generation accuracy, refine UI, and fix UI-related issues.

(🟢 **Low Priority)** Final demo preparation & deployment of the application.

.

Phase-5: Project Development

Objective:

Implement core features of the Personalised Email Generator using Streamlit.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** Python-based email generation logic
    - **Programming Language:** Python
2. **Development Process:**
    - Implement user input handling for personalized email generation.
    - Develop customization options (tone, length, and subject line).
    - Optimize email content generation for coherence and relevance.
3. **Challenges & Fixes:**
    - **Challenge:**Generating diverse and natural-sounding email content.
    - **Fix:** Implement predefined templates and AI-driven content variations.
    - **Challenge:** Ensuring proper email formatting and structure.
    - **Fix:** Apply text formatting rules and preview functionality for users.

Phase-6: Functional & Performance Testing

Objective:

Ensure that Personalised Email Generator using Streamlit works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Generate a professional email for a job application.	A well-structured, formal email should be generated.	✅ Passed	Entire Team
TC-002	Functional Testing	Generate an informal email for a friend.	The email should have a casual and friendly tone.	✅ Passed	Entire Team
TC-003	Performance Testing	Email generation time under 500ms.	The email should be generated quickly.	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed grammar and sentence structuring errors.	Generated emails should have improved coherence.	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work seamlessly on mobile & desktop.	❌ Failed - UI broken on mobile	Tester 2
TC-006	Deployment Testing	Host the app using Streamlit Sharing.	The app should be accessible online.	🚀 Deployed	DevOps

Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**