

Hackathon Project Phases Template

Project Title:

AI Personalized Email Generator

Team Name:

Team MARS

Team Members:

- Gudikandula Ramya
 - Pentam Madhulika
 - Bangaru Sindhuja
 - Kanneboina Maheshwari
 - Parupelly Sai Aishwarya
-

Phase-1: Brainstorming & Ideation

Objective:

The main objective of an AI Personalized Email Generator is to automatically create customized, relevant, and engaging email content tailored to individual recipients. It enhances engagement, saves time, improves customer experience, and allows scalable, efficient email marketing.

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. 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

2. Proposed Solution:

- The AI Personalized Email Generator uses machine learning to analyze customer data and create tailored email content. It automates personalization at scale, boosting engagement and marketing efficiency.
- The app offers Automated Personalization and Improved Engagement

3. Target Users:

- **E-commerce businesses** for tailored product recommendations.
- **Marketers** looking to improve email campaigns
- **Sales teams** aiming to enhance lead conversion.
- **Customer relationship managers** focused on engagement.
- **Content creators** sending personalized newsletters.
- **Small and medium-sized businesses** seeking efficient email marketing.

4. Expected Outcome:

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

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 **Server-Side Frameworks and Languages**
- Frontend: **HTML**
- Database: **Not required initially (API-based queries)**

2. Functional Requirements:

- Ability to **Personalized Content Generation**.
- Ability to **Automated Email Sending**.
- Provide **real-time vehicle maintenance tips** based on seasons.
- Allow **Customer Feedback and Interaction**.

3. Constraints & Challenges:

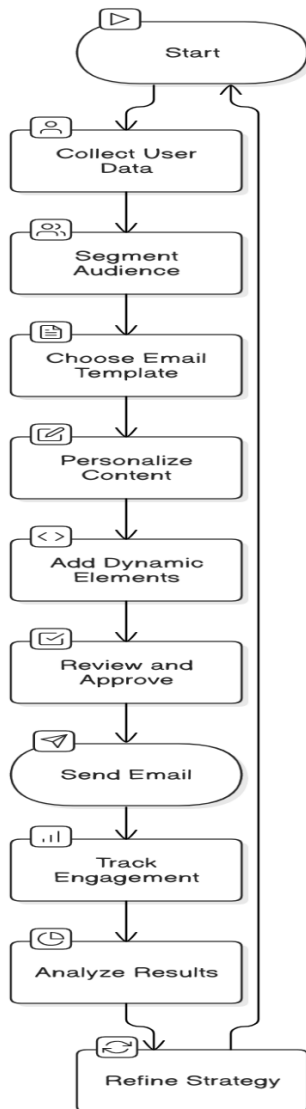
- Ensuring **Quality and Accuracy of Personalization**:
- Handling **Data Privacy and Security**
- Providing a **smooth Email Deliverability**.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.

Personalized Email Generator



Key Points:

1. System Architecture:

- 0 User Registration & Onboarding

- **Email Template Creation**
- **AI-Powered Email Personalization**
- **Preview email and Test email sending**
- **Email Tracking and Analytics.**

2. User Flow:

- Step 1: User enters a query (e.g., "Mail to manager for leave").
- Step 2: The backend **calls OPEN API** to retrieve query.
- Step 3: The app processes the data and **displays results** in an easy-to-read format.

3. UI/UX Considerations:

- **Simple and Intuitive Interface**
- **Personalization Options**
- **AI Recommendations and Suggestions**
- **Interactive Feedback & Iteration**
- **Security and Privacy**

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	Set up databases and APIs	🔴 High	6 hours (Day 1)	End of Day 1	Ramya	Flask (flask), Python, API Documentation	supports data storage, retrieval, and secure communication
Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	Madhulika	API response format finalized	Basic UI with input fields
Sprint 2	Implement authentication & authorization	🔴 High	3 hours (Day 2)	Mid-Day 2	Aishwarya	API response, flask-security	Secure authentication and API protection

Sprint 2	Error Handling & Debugging	🔍 High	1.5 hours (Day 2)	Mid-Day 2	Maheshwari	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🔍 Medium	1.5 hours (Day 2)	Mid-Day 2	Sindhuja	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 Required Libraries.
- (🔍 High Priority) Set Up **OpenAI API Key** and Create a **Python File**.
- (🔍 Medium Priority) Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

- (🔍 High Priority) Run the Application
- (🔍 High Priority) Test the Application and 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 Personalized Email generator.

Key Points:

1. Technology Stack Used:

- **Frontend:** HTML
- **Backend:** Flask (Python)

- **Programming Language:** Python
2. **Development Process:**

- Develop the **user interface** based on the frameworks.
- Implement a text editor for users to create and **customize email templates**.
- Develop interactive dashboards
- **Python** to handle email campaign requests, user authentication, and tracking.
- Implement the data layer for storing customer profiles, email campaigns, and performance data.

3. **Challenges & Fixes:**

- **Challenge:** Ensuring compliance with data protection regulations.
Fix: Implement **data encryption** both in transit and at rest to protect user data.
- Challenge:** AI models must be able to generate high-quality, personalized email content
Fix: Use feedback loops where user interaction 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	Query "Email to HR for job"	Relevant mail writings should be displayed.	✓ Passed	Tester 1
TC-002	Functional Testing	Query "Motorcycle maintenance tips for winter"	Seasonal tips should be provided.	✓ Passed	Tester2
TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✓ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	✗ Failed - UI broken on mobile	Tester 4
TC-006	Deployment Testing	Host the app using Streamlit Sharing	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**