**Hackathon Project Phases Template** for the **AI personalised Email Generator** project.

# Hackathon Project Phases Template

**Project Title: AI personalized Email Generator**

**AI personalized Email Generator App Using Generative AI**

**Team Name: AI CHEERS**

**Team Members:**

* **ETTABOINA ALEKHYA**
* **DHARAVATH YAKUB**
* **KEMIDI SHIVANI**
* **KALAKOTI SRAVYA**
* **LINGALA PRANITHA VARMA**

## Phase-1: Brainstorming & Ideation

**Objective:**

Develop an AI-Personalized Email generator Is great idea for business, marketers or individuals seeking to enhance their communication.

**Key Points:**

1. **Problem Statement:**

* 1. AI-personalized email generator that can create personalized emails with a human-like touch.

○ Automate the email creation process to reduce the time spent on crafting emails.

1. **Proposed Solution:**

* 1. An AI-personalized email generator uses natural language processing and machine learning algorithms to create customized emails that are tailored reciepient name.

○ collecting recipient information , such as name, email address, and event details.

**Target Users:**

○ Email marketers who want to create a personalised email compaigns to engage with customers.

○ content creators who want to create personalized emails to promote their content and engage with their audience.

1. **Expected Outcome:**

* 1. A functional **AI PERSONALIZED EMAIL GENERATOR** that provides insights based on real-time data and user queries.

## Phase-2: Requirement Analysis

**Objective:**

Define the technical and functional requirements for the AI PERSONALIZED EMAIL GENERATOR.

**Key Points:**

1. **Technical Requirements:**

* 1. Programming Language: **Python**

○ Backend: **Google AI Studio**

○ Frontend: **Git**

○ Database: **Email Service Providers (ESPs)**

1. **Functional Requirements:**

* 1. Ability to **fetching email details** using Generator AI personalized email generator using Gen AI..

○ Display **ai generated** **email , reviews, and ESP integration** in an intuitive gemini.

○ Provide **speed,scalability and uptime** with the ensuring the delivered reliably.

○ Allow users to **generating eco-friendly emails** based on pre-requisites.

1. **Constraints & Challenges:**

* 1. Ensuring real-time email updates from **Gen AI**.

○ AI email generators require high-quality data to learn and improve.

○ AI **data collection** ,**data labelling** and **data security** are the main challenges of ai personalized email generator.

## Phase-3: Project Design

**Objective:**

Develop the architecture and user flow of the ai personalized email generator.

[Start]

|

[Data Collection]

|

[Preprocessing] ---> [Template Selection]

| |

[Personalization] ----> [Content Generation]

| |

[Quality Assurance] ----> [Email Scheduling & Sending]

| |

[Feedback & Optimization]

|

[End]

**Key Points:**

1. **System Architecture:**

* 1. User enters user-Interface is a web-based interface for users to recipient information, select templates, and customize email content.

○ Query is processed using **Google Ai Studio**.

○ ○ The frontend displays **AI processing , data ingestion, data storage and generation of email.**

1. **User Flow:**

* 1. Step 1: User registers the AI-powered personalized email generator service.

○ Step 2: user creates a profile list for the email campaign.

○ Step 3: The app processes the data and **email template selection** in an easy-to-read format.

1. **UI/UX Considerations:**

* 1. **clear, simple user-friendly**  navigation.

○  **prominent call to actions**  uses clear to prominent CTA’s to encourage users to take action.

○  **Email template preview** provides a preview of the email template to help user to visualize the final product.

## 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 | planning | 🔴 High | 6 hours  (Day 1) | End of Day  1 | Member 1 | Google AI studio,  Python, Streamlit setup | Plan of the personalized email generator |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 2 hours  (Day 1) | End of Day  1 | Member 2 | Visual studio code | Basic UI with input fields |
| Sprint 2 | Email generating | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Member 1& 2 | AI Gemini , UI elements ready | Search functionality with requirements |
| Sprint 2 | Error Handling &  Debugging | 🔴 High | 1.5 hours  (Day 2) | Mid-Day 2 | Member 1&4 | AI generated email , UI inputs | Improved personalized AI emails |
| Sprint 3 | Testing & UI  Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Member 2& 3 | Test-Driven Development | Codebase tested properly |
| 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 Generator.

**Key Points:**

1. **Technology Stack Used:**

* 1. **Frontend:** Python

○ **Backend:** Google AI Studio or Google Gemini

○ **Programming Language:** Python

1. **Development Process:**

* 1. **Planning and researching** the project scope ,goals and deliverable.

○ **Data collection** and preparation to collect the data of an ai personalized email generator.

○ **Developing** , **moduling,testing** and **monitoring(deploying)** the collected data to generate ai personalized email generator.

1. **Challenges & Fixes:**

* 1. **Challenge:** To generate AI personalized Email Generator .

**Fix:** Implement **Google AI Studio** to store frequently queried results.

○ **Challenge:**collecting the user data like recipient’s name,event address,event date and additional information.

**Fix:** Optimizeto fetch **only necessary data** of an user.

## 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 | Functional testing is a type of testing to making the error-free. | Verifies the AI-powered email generator can generate recipient’s data. | ✅ Passed | Tester 1 |
| TC-002 | Load  Testing | Simulate a large volume of users interacting with the AI email generator. | Response time for generating amd sending personalized emails was 2.5 seconds. | ✅ Passed | Tester 2 |
| TC-003 | Stress  Testing | Checks how it recovers from overload situations or fails gracefully. | System crash,error rate and Response time should be tested here. | ⚠ Needs Optimizing | Tester 3 |
| TC-004 | Scalability Testing | Test the system’s ability to handle growing of users ,emails, to data input without bottlenecks. | Increasing concurrent users ,increasing email volume and also data size increased. | ✅ Fixed | Develop er |
| TC-005 | Final Validation | Final testing is done without any debugs. | Verifying correctness,ensure quality and validate user experience. | ✅ validated | Tester 2 |
| TC-006 | Deployment  Testing | Deploying the project to Git or GitHub. | Email should be generated with recipient’s details. | 🚀 Deployed | DevOps |

## Final Submission

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