Hackathon Project Phases Template for the AI personalized email generator.

Hackathon Project Phases Template

Project Title:

AI personalized email generator

Team Name:

Code Quill

Team Members:

- B.Harichandra prasad
- K.Dileep Raju
- K.Aditya Sai
- A.Abhilash Reddy

Phase-1: Brainstorming & Ideation

Objective:

Create an AI-powered email generator that personalizes email content for users based on their input and preferences. The application will adapt tone, style, and content according to the recipient, purpose, and preferred email structure.

Key Points:

1. **Problem Statement:**
    - Individuals often spend a lot of time drafting emails, especially for repetitive communication.
    - Personalization of emails is time-consuming and can lack consistency.

2. **Proposed Solution:**

- Develop an AI-powered tool that personalizes emails based on user input (e.g., recipient, tone, subject, purpose).
- Uses NLP (Natural Language Processing) to generate personalized emails that align with the user's communication style and recipient's needs.

3. **Target Users:**

   - Professionals looking for quick email drafting.
   - Sales teams needing personalized email outreach at scale.

4. **Expected Outcome:**

   - A working AI-powered email generation tool.
   - A customizable experience for the tone, style, and structure of the emails.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the AI-Personalized Email Generator.

Key Points:

1. **Technical Requirements:**
   - Backend: Node, express
   - Frontend: React and Tailwind CSS
   - Database: postgresql
   - API: Integration with OpenAI's GPT-3/4 for content generation.
2. **Functional Requirements:**

   - Allow users to input recipient, purpose, tone, and subject.
   - Generate email content using GPT-3/4.
   - Provide preview and edit features for generated emails.

3. **Constraints & Challenges:**

   - Ensuring the accuracy of personalization based on the user input.
   - Handling large-scale requests and ensuring API limits are managed efficiently.
   - Ensuring security and privacy of user data, especially sensitive email content.

Phase-3: Project Design

Objective:

Design the system architecture and user interface flow for the email generator tool.

Key Points:

1. **System Architecture:**
   - User inputs data through the frontend UI (e.g., recipient details, tone, subject).
   - Frontend sends input data to the backend API
   - The backend communicates with GPT-3/4 API to generate email content.
   - The response is sent back to the frontend, where users can view, edit, and send the email.
2. **User Flow:**
   - **Step 1:** User enters recipient, tone, subject, and other preferences.
   - **Step 2:** Frontend sends the input to the backend.
   - **Step 3:** Backend sends request to GPT-3/4 API for content generation.
   - **Step 4:** GPT-3/4 API generates email and sends it back to the backend.
   - **Step 5:** Backend processes and sends the generated email content to the frontend.
   - **Step 6:** User can preview, edit, and send the email.
3. **UI/UX Considerations:**
   - A clean, minimalistic UI with clear input fields for customization.
   - Responsive design to ensure compatibility across devices.
   - Option to save drafted emails for later use or reuse.

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down the development tasks and plan the sprints for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & API Integration	◍ High	6 hours (Day 1)	End of Day 1	abhilash	Google API Key,	API integrated, ready for use
Sprint 1	Frontend UI Development	◍ Medium	4 hours (Day 1)	End of Day 1	aditya	API response format finalized	Basic UI with input fields

Sprint 2	Email Generation Logic	● High	3 hours (Day 2)	Mid-Day 2	hari	Frontend ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	● High	1.5 hours (Day 2)	Mid-Day 2	Aditya and hari	Email logic ready	Improved API stability
Sprint 3	Testing & UI Enhancements	● Medium	1.5 hours (Day 2)	Mid-Day 2	dileep	Email logic integrated	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 development environment and install dependencies.
- ○ Initialize the React project with TypeScript and Tailwind CSS.

Sprint 2 – Core Features & Debugging (Day 2)

● **High Priority**

- ○ Implement user input forms and filters on the frontend.
- ○ Develop backend logic to process user input and fetch recipes from the Gemini API.
- ○ Ensure proper data flow between frontend and backend.
- ○ Debug any issues related to API integration and data handling.

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

● **Medium Priority**

- ○ Conduct functional and performance testing.
- ○ Refine UI components for better user experience.
- ○ Implement additional features such as saving favourite recipes.
- ○ Prepare final documentation and deployment.

Phase-5: Project Development

Objective:

Implement core features of the AI-Personalized Email Generator.

Key Points:

1. **Technology Stack Used:**

    - **Frontend:** React with TypeScript and Tailwind CSS.
    - **Backend:** Node.js ,OpenAI API for GPT-3/4.
    - **API:** OpenAI GPT-3/4 for email generation.

2. **Development Process:**

    - Set up the React project and integrate Tailwind CSS.
    - Develop the backend to handle API requests.
    - Integrate GPT-3/4 to generate email content dynamically.
    - Ensure seamless communication between frontend and backend.

3. **Challenges & Fixes:**

    - **Challenge:** Managing user data input and generating personalized responses.
      Fix: Utilize state management libraries such as Redux or Context API.
    - **Challenge:** Handling real-time generation with GPT-3/4.

1. **Fix:** Optimize API calls with proper error handling and asynchronous processing.

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	Generate email with recipient, tone, and subject	Email should be personalized as per input.	☑ Passed	aditya
TC-002	Performance Testing	API response time under 500ms	Email should be generated quickly	⚠ Needs Optimization	hari

TC-003	Bug Fixes & Improvements	Ensure email structure and tone are appropriate	Email structure and tone should match the input.	☑ Fixed	dileep
TC-004	Final Validation	Test UI responsiveness on mobile and desktop.	Data accuracy should be improved.	☑ Fixed	abhilash
TC-005	Deployment Testing	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	🚀 Deployed	hari

Final Submission

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