Hackathon Project Phases Template

# Project Title:

**Generator SmartResume**: Customized Resumes for Every Opportunity

# Team Name:

Code Hunts

# Team Members:

* Nandini Tandra
* Charishma Lekkala
* Misha Emmadi
* Neha sree Maheshwaram

# Phase-1: Brainstorming & Ideation

## Objective:

## Develop an AI-powered Smart Resume Generator that enables users to create professional, customized resumes efficiently by leveraging AI for content suggestions, formatting, and optimization.

## Key Points:

## 1. Problem Statement:

## Many job seekers struggle with creating well-structured, ATS-friendly resumes that highlight their skills effectively.

## Users often find it challenging to tailor resumes for specific job roles and industries.

## Formatting resumes can be time-consuming, and users may not be aware of industry best practices.

## 2. Proposed Solution:

## An AI-powered Smart Resume Generator that assists users in crafting professional resumes with AI-driven content recommendations.

## Features include pre-built templates, role-based resume customization, grammar correction, and keyword optimization for ATS compatibility.

## Users can generate resumes in different formats (PDF, Word) and receive suggestions for improvements based on job descriptions.

## 3. Target Users:

## Job seekers needing tailored and professional resumes.

## Students & fresh graduates looking to create their first resume.

## Freelancers & gig workers wanting a skills-based resume for client proposals.

## 4. Expected Outcome:

## A functional AI-powered Smart Resume Generator that simplifies resume creation, enhances content, and ensures ATS compliance.

## A seamless, user-friendly interface that enables users to generate, edit, and download high-quality resumes effortlessly.

# Phase-2: Requirement Analysis

## Objective:

## Define the technical and functional requirements for the Smart Resume Generator application.

## Key Points:

## 1. Technical Requirements:

## Programming Language: Python

## Backend: AI-powered text processing (e.g., OpenAI API, Gemini Flash)

## Frontend: Streamlit Web Framework

## Database: Not required initially (local storage or API-based queries)

## 2. Functional Requirements:

## Allow users to input personal and professional details through an interactive UI.

## Provide AI-assisted resume content suggestions based on job roles and industries.

## Offer multiple pre-designed resume templates with customization options.

## Generate ATS-friendly resumes with keyword optimization.

## Support PDF and Word export for easy downloading and sharing.

## Enable real-time grammar correction and content enhancement.

## Allow users to tailor resumes based on job descriptions.

## 3. Constraints & Challenges:

## Ensuring real-time AI-powered resume suggestions with minimal latency.

## Handling formatting and layout consistency across different resume templates.

## Optimizing the Streamlit UI for a smooth and responsive user experience.

* Managing API rate limits for AI-powered text processing

# Phase-3: Project Design

## Objective:

## Develop the architecture and user flow of the application.

## System Architecture:

## User enters resume details via UI.

## Input data is processed using Google Gemini API.

## AI model generates and optimizes resume content.

## The frontend displays the smart resume in various customizable formats.

## User Flow:

## Step 1: User enters personal and professional details in the app.

## Step 2: The backend calls the Gemini Flash API to generate optimized resume content.

## Step 3: The app processes the data and displays the resume in an easy-to-read and downloadable format.

## UI/UX Considerations:

## ✔ Minimalist, user-friendly interface for easy navigation.

## ✔ Customizable resume templates for personalization.

## ✔ Dark & light mode for better user experience.

## ✔ Real-time resume preview before downloading

# 

# Phase-4: Project Planning (Agile Methodologies)

## Objective:

## 

## 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 AI-based resume content generation.

## (🔴 High Priority) Debug API issues & handle errors in user input processing.

## 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 Smart Resume Generator.

## Key Points:

## 1. Technology Stack Used:

## Frontend: Streamlit

## Backend: Google Gemini API

## Programming Language: Python

## 2. Development Process:

## Implement API key authentication and integrate Google Gemini API.

## Develop resume generation logic with customizable templates.

## Optimize user input processing for structured and efficient resume creation.

## 3. Challenges & Fixes:

## Challenge: Formatting inconsistencies in generated resumes.

## Fix: Standardize resume templates with predefined layouts.

## Challenge: API limitations affecting response speed.

* **Fix:** Implement local caching for previously generated resumes.

# Phase-6: Functional & Performance Testing

## Objective:

* + Ensure the **Smart Resume Generator** works as expected.

## 

## Key Points:

## 

# Final Submission

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