**Hackathon Project Phases Template**

**Project Title: Smart Resume Generator: Customized Resumes for Every Opportunity**

**Team Name:** CVBoost

**Team Members:**

● Name 1 : CH Shriya

● Name 2 : K Jeswitha

● Name 3 : M Indu

● Name 4 : K Lakshmi Prasanna

**Phase-1: Brainstorming & Ideation**

**Objective:** Identify the problem of resume creation and propose an AI-powered solution.

**Key Points:**

1. **Problem Statement:**

Creating a professional resume can be a challenging task, especially for individuals with limited experience in designing and structuring resumes. Many job seekers struggle with formatting, content, and compatibility, leading to lower chances of getting shortlisted.

1. **Proposed Solution:**

An AI-powered resume maker that helps users generate professional resumes by dynamically filling in key details using AI prompts. The platform will provide an intuitive UI where users can input their information, and the AI will generate a well-structured resume.

3. **Target Users:**

* Job seekers (freshers & experienced professionals)
* Students and graduates
* Freelancers and consultants
* Career changers

4. **Expected Outcome:**

* Users will get a polished, AI-generated resume with minimal effort.
* The system will support a good looking and simple resume templates.
* The platform will be user-friendly, customizable, and optimized.

**Phase-2: Requirement Analysis**

**Objective:**

● Define the technical and functional requirements of the project.

**Key Points:**

1. **Technical Requirements:**

* **Languages:** Python, HTML&CSS
* **Frameworks:** Flask & Python, SQLite (Backend), HTML&CSS(Frontend)
* **Tools & Libraries:** OpenAI API (for AI-based resume generation), SQLite (Database), CSS (Styling), Gemini API, FPDF, Flask-login, Flask-Bcrypt, Beautifulsoup.
* **Deployment:**

2.**Functional Requirements:**

* User authentication (login, registration)
* AI-powered resume content generation
* Resume customization (templates, styling options)
* Resume preview and download (PDF format)
* User dashboard to manage resumes

3. **Constraints & Challenges:**

* Ensuring AI-generated content is accurate and professional
* Keeping the UI simple yet feature-rich
* Handling multiple resume templates efficiently
* Managing user data securely

**Phase-3: Project Design**

**Objective:**

● Create the system architecture and define user interactions.

**Key Points:**

1. **System Architecture Diagram:**

User → Frontend (Python, HTML) → Backend (Flask API) → Database(SQLite)

Backend communicates with Gemini AI API for resume generation

1. **User Flow:**

User registers/logs in

Fills out resume details

AI generates resume content

User downloads the resume

1. **UI/UX Considerations:**

Minimalist, professional design

Easy-to-use form for input collection

Live preview of resume format

**Phase-4: Project Planning (Agile Methodologies) Objective:**

● Break down tasks using Agile methodologies to ensure smooth development.

**Key Points:**

1. **Sprint Planning:** 
   * + Sprint 1: User authentication & database setup
     + Sprint 2: Resume form & AI integration
     + Sprint 3: UI design & customization features
     + Sprint 4: Testing, debugging & deployment

1. **Task Allocation:**

Backend Development: Flask API, database integration

Frontend Development: Python, styling-CSS

AI Integration: OpenAI API implementation

Testing & Deployment: Debugging, hosting setup

1. **Timeline & Milestones:**

|  |  |
| --- | --- |
| **Task** | **Duration** |
| Authentication & Database Setup | 1day |
| Resume Input & AI Integration | 1day |
| Frontend UI & Customization | 1day |
| Testing & Deployment | 1 day |

1. **Phase-5: Project Development**

**Objective:**

● Develop the project, integrate AI components, and ensure functionality.

**Key Points:**

* 1. **Technology Stack Used:**

**Frontend:** Python, CSS- Styling

**Backend:** Flask, Gemini AI API

**Database:** SQLite

**UI:** Bootstrap5, Beautifulsoup

**Login Authentication:** Flask-login and Flask-Bcrypt

* 1. **Development Process:** 
     + Backend API setup with Flask
     + AI integration for resume content generation
     + Frontend UI design with live preview feature
     + Resume PDF generation & download feature
  2. **Challenges & Fixes:** 
     + **Challenge:** Formatting AI-generated resumes properly
       - **Fix:** Implementing a structured template system
     + **Challenge:** User session management
       - **Fix:** Using Flask-Login for authentication

**Phase-6: Functional & Performance Testing Objective:**

● Ensure that the project works as expected and performs optimally.

**Key Points:**

1. **Test Cases Executed:**

* User authentication (register/login/logout)
* AI resume generation accuracy
* Resume download functionality
* UI responsiveness across devices

2. **Bug Fixes & Improvements:**

* Improved AI-generated content quality
* Fixed UI inconsistencies on mobile devices
* Optimized database queries for better performance

**3. Final Validation:**

* Verified all core functionalities match project requirements
* Tested on different browsers and screen sizes

4. **Deployment (if applicable):** (Hosting details or final demo link)

**Final Submission**

**GitHub/Code Repository Link :** [**https://github.com/Neelanjana22/SmartResume.git**](https://github.com/Neelanjana22/SmartResume.git)