

Hackathon Project Phases Template for the **Smart Resume Generator** project.

Hackathon Project Phases Template

Project Title:

Smart Resume Generator: Customized Resumes for Every Opportunity

Team Name:

NEURAL NOMADS

Team Members:

- Kavali Vinay
 - Nyalakanti Manideep
 - Battula Ram Gopal
-

Phase-1: Brainstorming & Ideation

Objective:

Create an **AI-powered Smart Resume Generator** that helps users build personalized, compelling resumes. The tool will analyze job descriptions and individual experiences to highlight relevant skills and achievements.

Key Points:

1. Problem Statement:

- Many job seekers struggle to create tailored resumes that stand out to employers.
- Users need guidance on how to optimize their resumes for different job applications, including ATS compatibility.

2. Proposed Solution:

- An AI-powered **Smart Resume Generator** that offers personalized resume building by analyzing job descriptions and user experiences.
- The app provides tips on industry-specific keywords, formatting, and content customization.

3. Target Users:

- **Job seekers** looking to create effective resumes for specific positions.
- **Individuals wanting** to improve their resumes using industry insights and ATS advice.
- **Recent graduates** and professionals re-entering the job market.

4. Expected Outcome:

A user-friendly **AI application** that helps individuals generate **high-quality, tailored resumes** that increase their chances of **securing interviews**.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the **Smart Resume Generator**

Key Points:

1. Technical Requirements:

- Programming Language: **Python**
- Backend: **Google Deep AI**
- Frontend: **HTML,CSS,JS**
- Database: **Not required initially (API-based queries)**

2. Functional Requirements:

- Ability to fetch **resume templates** and relevant job description data using an AI integration.
- Display customizable **resume formats, relevant skills**, in a user-friendly interface.
- Provide **real-time tips** on improving resume content based on current job market trends
- Allow users to explore eco-conscious career paths and industries focused on sustainability

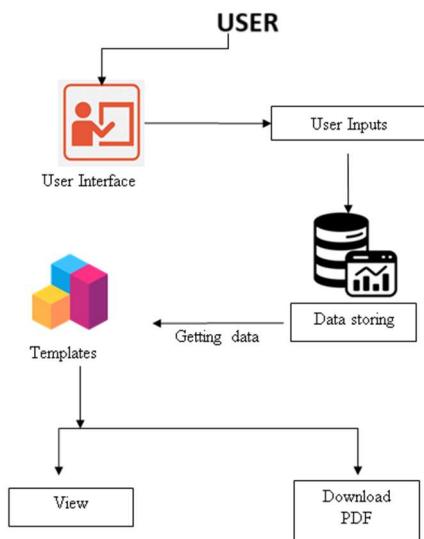
3. Constraints & Challenges:

- Ensuring real-time updates from **Gemini API**.
 - Handling **limitations on API** calls when fetching job descriptions,
 - Providing a seamless particularly with **larger datasets** or complex results
-

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. System Architecture:

- User inputs a resume-related query via the UI.
- Query is processed using the Smart Resume **Generator's backend algorithms**.
- The AI model analyzes the **query and extracts** relevant resume content.

- The frontend displays tailored resume options, formatting suggestions, and tips for improvement.

2. User Flow:

- Step 1: User enters a query (e.g., "Resume template for marketing manager").
- Step 2: The backend processes **the input and retrieves suitable resume templates** and content suggestions.
- Step 3: The app organizes the data and **presents results in a clean, editable format** for user refinement.

3. UI/UX Considerations:

- **Intuitive, user-friendly interface** designed for effortless navigation.
 - **Filters for experience level, job role, and industry to tailor content recommendations.**
 - **Default light mode** for better user experience.
-

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 | Environment Setup & API Integration | ● High | 6 hours (Day 1) | End of Day 1 | Vinay | Google API Key, JS | API connection established & working |
| Sprint 1 | Frontend UI Development | ● Medium | 2 hours (Day 1) | End of Day 1 | Manideep | Basic UI with input fields | API response format finalized |
| Sprint 2 | Resume Generation Logic Implementation | ● High | 3 hours (Day 2) | Mid-Day 2 | Ram Gopal | API response, UI elements ready | API response, UI elements ready |
| Sprint 2 | Error Handling & Debugging | ● High | 1.5 hours (Day 2) | Mid-Day 2 | Ramgopal | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | ● Medium | 1.5 hours (Day 2) | Mid-Day 2 | Vinay | API response, UI layout completed | API response, UI layout completed |
| 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 **Smart Resume Generator**

Key Points:

1. **Technology Stack Used:**
 - **Frontend:** HTML,CSS,JS
 - **Backend:** Google Gemini
 - **Programming Language:** Python
2. **Development Process:**
 - Implement **API key authentication** for user data security
 - Develop **logic for generating personalised resume content**.
 - **Optimize content retrieval and formatting suggestions based on user inputs**

3. Challenges & Fixes:

- **Challenge:** Inconsistent formatted outputs for different job roles.
Fix: Create **standardized** templates and guidelines to ensure uniformity across resumes.
 - **Challenge:** User difficulty in navigating the resume customization options
Fix: Optimize queries to fetch **only necessary data**.
-

Phase-6: Functional & Performance Testing

Objective:

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

| Test Case ID | Category | Test Scenario | Expected Outcome | Status | Tester |
|--------------|--------------------------|---|---|--|-----------|
| TC-001 | Functional Testing | Query " Resume template for software developer" | Suitable resume templates should be displayed | <input checked="" type="checkbox"/> Passed | Tester 1 |
| TC-002 | Functional Testing | Query " Tips for writing a standout resume" | Practical tips should be provided. | <input checked="" type="checkbox"/> Passed | Tester 2 |
| TC-003 | Performance Testing | API response time under 300ms | API should return results quickly. | ⚠ Needs Optimization | Tester 3 |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses. | Resume formatting should be accurate. | <input checked="" type="checkbox"/> Fixed | Developer |
| TC-005 | Final Validation | Ensure UI is user-friendly and intuitive | UI should be easy to navigate on all devices. | <input checked="" type="checkbox"/> Failed - UI Issues | Tester 2 |
| TC-006 | Deployment Testing | Host the app using a Cloud service | 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**

