

RESUME GENERATOR

5TEEN

K.Preetham Reddy

Harshith Rathod

M.Ram

V.DeepSitha

G.Karthika

Phase 1 - Brainstorming & Ideation Objective:

An AI-powered resume generator that automates the process of creating professional resumes tailored to user input.

KEY POINTS

- ◆ PROBLEM STATEMENT
- ◆ PROPOSED SOLUTION
- ◆ TARGET USERS
- ◆ EXPECTED OUTCOME

Problem Statement:

- Many job seekers struggle with resume formatting and content optimization.
- Existing solutions lack personalization and require manual effort.
- Users need a tool that generates resumes instantly, enhancing job application success.

Proposed Solution:

- AI-powered resume builder that tailors resumes based on user input.
- Automatic formatting and content structuring.
- Provides industry-specific customization and optimization suggestions.

Target Users:

- Job seekers needing professionally crafted resumes.
- Fresh graduates and students entering the job market.
- Professionals looking to switch careers or enhance their resumes.

Expected Outcome:

- A functional AI-driven resume generator.
- Well-structured, professional, and optimized resumes.
- Increased job application success rate for users.

Phase 2 - Requirements Analysis Objective

The technical and functional requirements for the Resume Generator project

- ◆ TECHNICAL REQUIREMENT
- ◆ FUNCTIONAL REQUIREMENTS
- ◆ CONSTRAINTS AND CHALLENGES

Technical Requirements:

- Programming Language: Python
- Backend: AI-based generative model
- Frontend: Streamlit/Web Framework
- Database: Optional for user history storage

Functional Requirements:

- Accepts user input for resume sections.
- Generates structured, formatted resumes.
- Provides optimization and customization options.

Constraints & Challenges:

- Ensuring AI-generated content aligns with industry standards.
- Handling user data securely and ensuring privacy.
- Optimizing AI processing time for instant resume generation.

Phase 3 - Project Design Objective

The architecture and user flow of the application.

- ◆ SYSTEM ARCHITECTURE
- ◆ USER FLOW
- ◆ UI/UX CONSIDERATIONS

System Architecture:

- User enters resume details via UI.
- AI model processes and formats the input.
- Resume is generated in a structured template.
- User can download or customize further.

User Flow:

- Step 1: User inputs personal details, job experience, and career goals.
- Step 2: AI processes input and applies formatting and optimization.
- Step 3: The system generates a professional resume ready for download.

UI/UX Considerations:

- Minimalist, user-friendly interface.
- Multiple pre-designed resume templates.
- Export functionality (PDF, DOCX).

Phase 4 - Project Planning (Agile methodologies)

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Assigned To	Expected Outcome
Sprint 1	Environment Setup & AI Integration	● High	6 hours (Day 1)	Team Member 1	AI model setup completed
Sprint 1	Frontend UI Development	● Medium	2 hours (Day 1)	Team Member 2	Basic UI with input fields
Sprint 2	Resume Generation & Formatting	● High	4 hours (Day 2)	Team Member 3	AI-generated resume templates functional
Sprint 2	Error Handling & Debugging	● High	2 hours (Day 2)	Team Member 4	Improved stability
Sprint 3	Testing & UI Enhancements	● Medium	2 hours (Day 2)	Team Member 5	Responsive UI & enhanced user experience
Sprint 3	Final Presentation & Deployment	● Low	1 hour (Day 2)	Entire Team	Demo-ready project

Phase 5 - Project Development Objective

Implement core features of the Resum Generator.

- ◆ TECHNOLOGY STACK USED
- ◆ DEVELOPMENT PROCESS
- ◆ CHALLENGES AND FIXES

Technology Stack Used:

- Frontend: Streamlit/Web Framework
- Backend: AI Generative Model
- Programming Language: Python

Development Process:

- Implement AI logic for resume generation.
- Optimize text formatting and content structuring.
- Ensure export functionality in multiple formats.

Challenges & Fixes:

- Challenge: Ensuring AI-generated content is relevant and well-structured.
• Fix: Implement industry-specific resume templates.
- Challenge: UI responsiveness and user customization.
• Fix: Provide multiple layout options and drag-and-drop features.

Phase 6 - Functional & Performance Testing

Ensure that the Resume Generator works as expected.

- ◆ FUNCTIONAL TESTING
- ◆ PERFORMANCE TESTING
- ◆ BUG FIXES AND IMPROVEMENTS

Functional Testing:

- Verify AI processes user input correctly.
- Ensure different templates generate properly.
- Validate export functionality (PDF, DOCX).

Performance Testing:

- Check AI response time for resume generation.
- Test system load with multiple users.

Bug Fixes & Improvements:

- Fix formatting errors in generated resumes.
- Enhance user experience with real-time content suggestions.



THANK YOU

