

Hackathon Project Phases Template for the **QwiX CV** project.

Hackathon Project Phases Template

Project Title:

QwiX CV : AI-Powered Resume Builder

Team Name:

QwikZen

Team Members:

- Dhadi Sai Praneeth Reddy
 - Kasireddy Manideep Reddy
 - Batchu Pravalika
-

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered resume builder that helps job seekers create professional, ATS-optimized resumes tailored to job descriptions, with intelligent suggestions and automated job matching.

Key Points:

1. Problem Statement:

- Many job seekers struggle to create resumes that pass Applicant Tracking Systems (ATS).
- Lack of guidance on resume content optimization and formatting.
- Difficulty in finding jobs that match their skills and experience.

2. Proposed Solution:

- AI-driven resume builder with content suggestions, keyword optimization, and ATS compatibility analysis.
- Real-time resume scoring and job matching based on resume content.
- Professionally designed templates with customization options.
- Automated job recommendations and direct application functionality.

3. Target Users:

- **Job seekers across various industries.**
- **Students and fresh graduates entering the job market.**
- **Professionals looking for career advancement.**

4. Expected Outcome:

- A fully functional AI-powered resume builder that enhances job applications with ATS-optimized resumes and job-matching capabilities.
-

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for QWIK CV.

Key Points:

1. Technical Requirements:

- Frontend: React, TypeScript, Tailwind CSS, shadcn/ui, Three.js.
- Backend Libraries: Tanstack Query, React Router, React Hook Form, date-fns.
- API Integrations: AI-powered content generation, Indeed API (for job search), ATS Scanner API (for resume analysis).
- PDF Generation: html2pdf.js for exporting resumes as professionally formatted PDFs.

2. Functional Requirements:

- AI-generated resume content suggestions.
- Real-time ATS compatibility scoring.
- Resume comparison tool to analyze job description relevance.
- Multi-step resume-building process.
- Job board integration for direct applications.

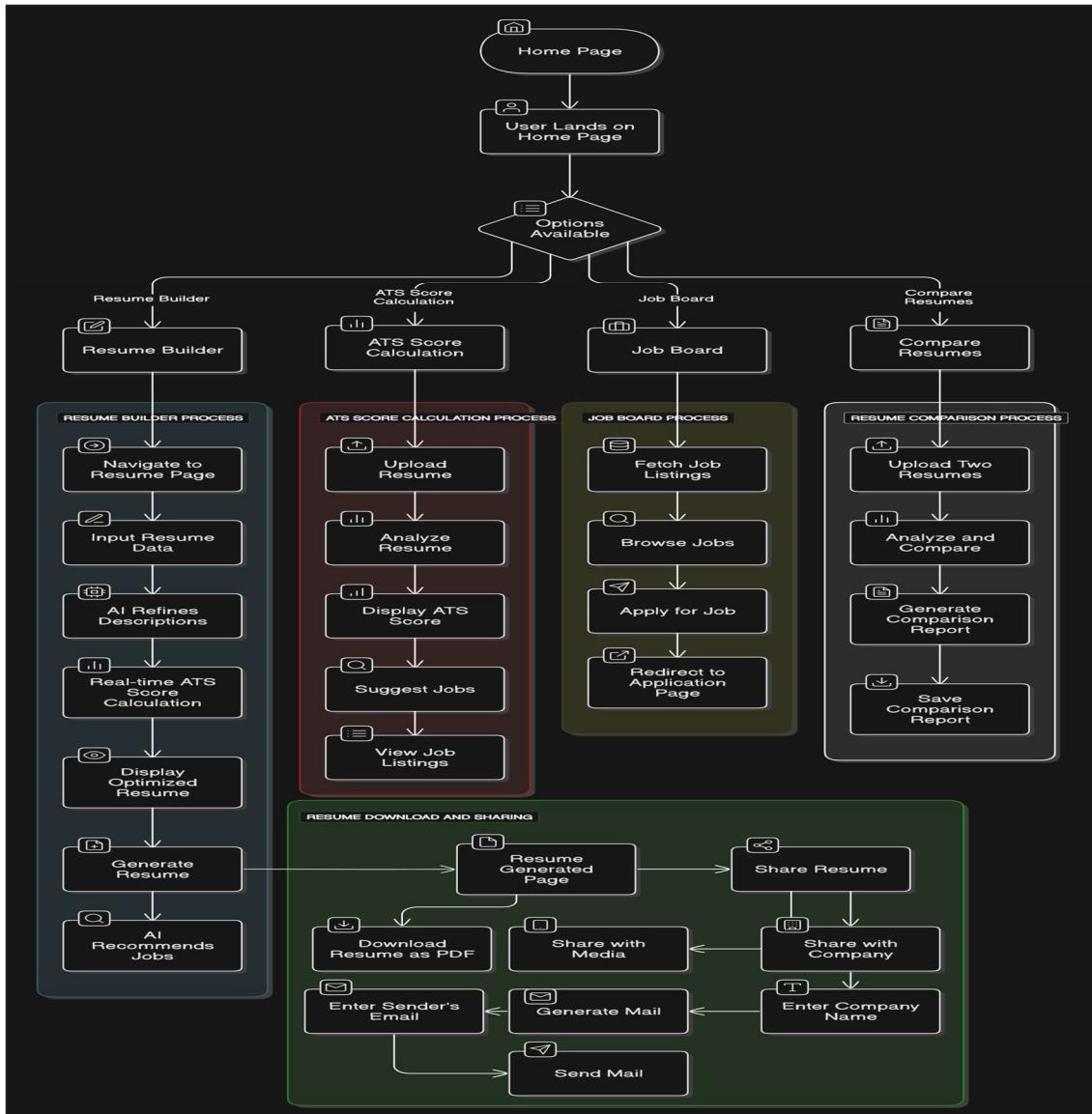
3. Constraints & Challenges:

- Ensuring accurate ATS scoring and compatibility analysis.
 - Optimizing AI-generated content for different industries.
 - Handling real-time job matching efficiently.
-

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the QwiX CV.



Key Points:

1. System Architecture:

- Users enter their details in a multi-step resume builder.
- AI generates content suggestions for each section.
- ATS Scanner analyzes resume strength and compatibility.
- Job board provides matching job recommendations.
- Users export or share resumes with potential employers.

2. User Flow:

- User logs in and starts building their resume.
- AI suggests improvements and provides keyword recommendations.
- ATS scanner evaluates resume against job descriptions.
- User selects a resume template and customizes formatting.
- Resume is exported as a PDF or shared with recruiters.
- Job board suggests relevant openings based on resume content.

3. UI/UX Considerations:

- Minimalist and intuitive interface for seamless navigation.
 - Mobile-first responsive design with Tailwind CSS.
 - Highly professional & ATS Friendly resume template.
-

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)	Mid-Day 1	Developer 1	AI API, Indeed API	API connection established
Sprint 1	UI Development	● Medium	4 hours (Day 1)	Mid-Day 1	Developer 2	API response format	Basic UI with input fields
Sprint 2	Resume Builder	● High	5 hours	Mid-Day 2	Developer 3	API, UI elements	Functional resume

	Core Features		(Day 2)			ready	builder
Sprint 2	ATS Scanner Integration	● High	4 hours (Day 2)	Mid-Day 2	Developer 4	Resume builder	ATS compatibility analysis
Sprint 2	Job Matching Feature	● Medium	3 hours (Day 2)	Mid-Day 2	Developer 5	Resume builder, APIs	Personalized job suggestions
Sprint 2	Testing & Debugging	● High	3 hours (Day 2)	Mid-Day 2	Tester	Completed features	Bug-free functionality
Sprint 2	Final UI/UX Enhancements	● Low	2 hours (Day 2)	Mid-Day 2	UI Designer	Testing feedback	Polished interface

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

- (● High Priority) Set up the environment & install dependencies.
- (● High Priority) Integrate AI API and resume analysis APIs.
- (● Medium Priority) Build a basic UI with input fields.

Sprint 2 – Core Features & Debugging (Day 2)

- (● High Priority) Implement resume builder functionalities.
 - (● High Priority) Integrate ATS Scanner and Job Matching.
 - (● High Priority) Debug API issues & handle errors in queries.
 - (● Medium Priority) Final UI/UX Enhancements.
 - (● Low Priority) Test API responses, refine UI, fix UI bugs & deployment.
-

Phase-5: Project Development

Objective:

Implement core features of the QwiX CV.

Key Points:

1. Technology Stack Used:

- **Frontend:** React, TypeScript, Tailwind CSS, shadcn/ui, Three.js.
- **Backend:** AI API integrations, Tanstack Query, React Hook Form.
- **Other Tools:** html2pdf.js (PDF export), Indeed API (job search), ATS Scanner API.

2. Development Process:

- I Implement API key authentication and AI API integration.
- Develop resume-building logic with intelligent content suggestions.
- Optimize search queries for performance and relevance.

3. Challenges & Fixes:

- **Challenge:** Ensuring AI-generated resume content is industry-relevant.
Fix: Train AI model with diverse job descriptions and industry-specific data.
 - **Challenge:** Optimizing ATS scanner for accuracy.
Fix: Use real job descriptions to test and refine the scoring algorithm.
-

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	AI generates resume content based on job role	Relevant resume content is suggested	Passed	TC-001
TC-002	Functional Testing	ATS Scanner evaluates resume compatibility	Resume receives a compatibility score	Passed	TC-002
TC-003	Performance Testing	AI response time under 500ms	AI suggestions appear quickly	Needs Optimization	TC-003
TC-004	Bug Fixes & Improvements	Fixed formatting inconsistencies in PDF export	Professional layout is maintained	Fixed	TC-004
TC-005	Final Validation	Ensure UI is responsive across devices	UI works on mobile & desktop	Fixed	TC-005
TC-006	Deployment Testing	Host QWIK CV for public access	Platform is live	Deployed	TC-006
