Checklist:

- ☑ ~~Vision~~
- ☑ ~~Market analysis~~
- ☑ ~~Competitors analysis~~
- ☑ ~~Target audience~~
- ☑ ~~Stakeholders~~
- ☑ ~~Requirements~~
- ☑ ~~Quality assurance~~
- ☑ ~~Development timeline~~
- ☑ ~~Process model~~
- ☑ ~~Future growth and scaling potential~~

1. Vision

Ciwi is an intelligent CV enhancement tool that empowers job seekers by optimizing their resumes to match specific job postings. Using advanced AI, Ciwi analyzes users' CVs and delivers targeted recommendations, transforming applications to align perfectly with employer needs and significantly boosting interview success rates.

Source: gpt-4o, prompt: Write a short (2-3 sentences), amazing and unforgettable project description about CV enhancement tool designed to help job seekers optimize their resumes for specific job postings. Ciwi scans the user's CV and provides tailored suggestions for improvement, increasing chances of landing the job.

2. Market Analysis

Market Demand: With job seekers constantly looking for ways to improve their applications, Ciwi addresses a strong demand.

3. Competitors Analysis

Direct competitors:
- Jobscan offers similar optimization features.
Indirect competitors:
- Linkedin offers courses on optimising resumes.
- ChatGPT can be used for CV improvement.

4. Target Audience

The primary target includes job seekers at various career stages, especially those in competitive or specialized fields.

5. Stakeholders

Potential stakeholders might include hiring firms and educational institutions interested in improving candidates' job readiness. Engaging stakeholders for partnerships could expand Ciwi's reach and credibility.

6. Requirements

Functional Requirements: Ciwi needs an intuitive user interface. Additional functionality could include customizable CV templates, keyword suggestions, and a scoring system to assess alignment with job descriptions. Functional Requirements

1. Resume Parsing and Analysis: the system must parse uploaded resumes to extract key information like work experience, education, and skills. Ciwi must identify missing or underrepresented elements in the CV based on job descriptions.
2. Job Description Parsing: Ciwi must extract and analyze keywords, skills, and requirements from job descriptions.
3. Keyword Suggestions: Ciwi should provide tailored keyword suggestions to align the resume with the job description.
4. Customizable Templates: offer a range of editable CV templates for users to choose from, supporting industry-specific formats.
5. Scoring System: display a compatibility score indicating how well the current CV matches the job description.
6. Recommendations Dashboard: provide users with detailed suggestions for improvement, including language enhancements, formatting changes, and content alignment.
7. Feedback Mechanism: allow users to rate the suggestions for future improvements of the AI algorithm.
8. User Account Management: enable users to create, update, and manage their accounts, including storing multiple CV versions.

9. File Format Compatibility: support multiple file formats for uploads (e.g., DOCX, PDF).
10. Multilingual Support: provide functionality for CV optimization in multiple languages.
11. Progress Tracking: track changes made to CVs and display improvement trends over time.
12. Integration Capabilities: allow integration with job boards for seamless job application submissions.
13. Customer Support: Offer an in-app help feature or chat support for user queries.

Non-Functional Requirements: Ensuring data security and privacy is essential. The system should have scalability and high performance for handling multiple users simultaneously. It should be accessible to individuals with disabilities.

1. Performance: the system must process CV and job description uploads and generate recommendations within a few seconds.

2. Scalability: ensure the system can handle thousands of simultaneous users through a cloud-based infrastructure.

3. Data Security: implement end-to-end encryption for data transfers and secure storage of user data.

4. Privacy Compliance: comply with data privacy regulations such as GDPR and CCPA.

5. Usability: the interface should be intuitive and user-friendly, accommodating users with varying levels of technical proficiency.

6. Accessibility: the tool should adhere to accessibility standards (e.g., WCAG 2.1), making it usable for individuals with disabilities.

7. Availability: guarantee 99.99% uptime to ensure the service is consistently available to users.

8. Maintainability: ensure code quality through reviews and maintain clear documentation for easier updates and bug fixes.

9. Adaptability: the system should be designed to allow easy integration of new features as market demands evolve.

10. Localization Support: design the system to support different regional requirements, such as date formats, languages, and job market trends, usage of european data centers for users connecting from the EU.

11. AI Model Reliability: regularly retrain and update the AI models to maintain recommendation accuracy and relevance.

12. Feedback Incorporation: implement a mechanism to incorporate user feedback into the system’s improvement cycle.

13. Cost Efficiency: optimize cloud resources to minimize operational costs while maintaining performance standards.

7. Quality Assurance

User Testing: Conduct usability testing to ensure that the interface is intuitive.

Quality Control:
- Implement regular quality checks on the AI models to ensure relevance and accuracy of recommendations.
- Automated code reviews and version control to maintain high development standards.

8. Development timeline:

Development could be organized into phases, such as:

1. initial research.
2. prototype: early-stage prototypes should validate core functionalities, especially resume parsing and keyword matching. Testing with a limited user group can provide insights into feature adjustments (should include feedback mechanisms, allowing users to rate suggestions, which can guide improvements in the recommendation algorithms).
3. MVP (Minimum Viable Product).
4. launch.

Establishing clear milestones would help track progress and ensure timely delivery.

9. Process Model:

An Agile development model with iterative feedback loops may be best, as user feedback is crucial to refining Ciwi’s functionality.

10. Future Growth and Scaling Potential

Ciwi’s infrastructure should be designed to support scalability, especially if user numbers grow rapidly. Cloud-based architecture could support this need cost-effectively.
Over time, Ciwi could expand into additional markets by partnering with international job boards.