Product Requirements Document: Recruiter's Assistant Tool

1. Introduction

- **Purpose:** To automate and streamline common recruitment tasks using AI-driven insights and integrations with existing platforms (Claap, Workable, BrightHire). The tool aims to reduce recruiters' manual effort in research, job description generation, candidate analysis, and submission preparation, enabling them to focus on strategic activities.
- **Goals:**
  - Automate repetitive recruitment tasks.
  - Improve recruiter efficiency and productivity.
  - Provide AI-powered insights for better decision-making.
  - Integrate seamlessly with existing recruitment tools (Claap, Workable, BrightHire).
  - Standardize processes like job description creation and candidate profiling.

2. Target Audience

- **Primary Users:** Recruiters (access to all functionalities).
- **Secondary Users:**
  - Hiring Managers/Clients (receivers of candidate profiles/submission emails).
  - System Administrators (user management, integrations, maintenance).

3. Features (MVP Scope)

The MVP includes the following modules:

- **3.1 Project Preparation**
  - **Objective:** Equip recruiters with comprehensive information about the vacancy, client, market context, search strings, and keywords.
  - **Requirements:**
    - Upload/Input Job Description (client's version) and Intake Form data.
    - Integrate with Claap to view debrief meetings and analyze transcripts for key requirements. (Fallback: manual transcript upload if no native integration).
    - Automated company research (founding date, size, turnover, LinkedIn, website, news, reviews, competitors) using web search capabilities.
    - Provide salary comparisons for similar roles in the specific location.
    - Identify similar job postings locally.
    - AI-generated questions: Potential candidate questions and suggested recruiter screening questions.
    - AI-generated, customizable search strings: LinkedIn Boolean, Google X-ray for LinkedIn, Google X-ray for CVs/Resumes.

        - AI-generated keyword lists with synonyms and relevant translations.
        - Store all gathered data (company info, job profile, documents, search strings) linked to the job record.
        - UI: Step-by-step intake wizard, embedded/linked Claap content, dedicated output pages/tabs for research and AI-generated content.

- **3.2 Job Description Generator**
    - **Objective:** Generate standardized, complete job descriptions using data from Project Preparation.
    - **Requirements:**
        - AI generates a draft JD including role overview, responsibilities, requirements, compensation, benefits, location, and EEO disclaimer.
        - AI generates concise, relevant screening questions based on established guidelines.
        - Allow recruiters to edit and finalize the generated JD and questions.
        - Export JD in various formats (PDF, Word) or potentially post to job boards (future integration).
        - Store final JDs for reference/reuse.
        - UI: Initiate generation from Project Preparation data, review/edit draft, save final version.

- **3.3 CV Analyzer**
    - **Objective:** Rank candidates against role requirements, allowing interactive refinement of criteria via chat.
    - **Requirements:**
        - Input CVs via manual upload (PDF/DOCX) or Workable integration (from "applied" stage). (Save option for Workable URLs).
        - Interactive chat interface for recruiters to add/remove/modify requirements (e.g., skills, experience).
        - AI recalculates candidate rankings in real-time based on refined requirements.
        - AI ranks CVs using a weighted scoring system (automatically defined weights, editable by recruiter).
        - Display the top candidates with scores/match percentages and criteria breakdown.
        - Persist refined requirements for the specific job record.
        - Store candidate resumes, info, and match scores.
        - UI: Import/fetch CVs, chat interface for requirements, automatically updating ranked list, detailed view per candidate.

- **3.4 Profile Creation**
    - **Objective:** Generate concise, client-facing candidate summaries using CV, interview data (BrightHire), and web presence.
    - **Requirements:**
        - Integrate with BrightHire to access interview recordings/transcripts and summarize key points.
        - Extract key data: Location, availability, salary, LinkedIn link, languages, education, career highlights, reasons for

        interest/leaving, potential concerns.
        - Allow definition of up to 5 custom headings based on client priorities (AI suggests, recruiter edits).
        - AI generates summaries (max 5 bullet points) for each custom heading.
        - Allow recruiters to edit and finalize the generated profile.
        - Store consolidated profile data.
        - UI: Select candidate, generate profile pulling from sources, add/edit custom headings, save final version.

- **3.5 Submission to Client**
    - **Objective:** Create a succinct, professional email body for candidate submission.
    - **Requirements:**
        - Generate email template using data from the created candidate profile.
        - Highlight candidate strengths and suitability briefly.
        - Allow recruiters to edit the generated text.
        - Provide options to copy text to an external email client or send via integrated mail service (optional, requires mailbox linking and signatures).
        - Use placeholders for client name, job title, candidate name, etc..
        - UI: Initiate from candidate profile, review/modify auto-generated email, send or copy.

4. Design & Architecture

- **UI/UX:** Web-based, responsive UI accessible via standard browsers. Separate pages/tabs for modules. Chat-like interface for CV Analyzer. Intuitive design with clear instructions.
- **Backend:** RESTful API or GraphQL suggested for data exchange. AI Integration Layer (NLP, summarization, text generation, ranking). Database for storing job descriptions, transcripts, candidate data, user prompts, outputs.
- **Integrations (MVP):**
    - Claap: Fetch recordings/transcripts (via API).
    - Workable: Access applicants/CVs from the "applied" stage (via API).
    - BrightHire: Access interview data/transcripts (via API).
    - AI/ML Services: Large language models for content generation, chat, summarization, etc..
    - Web Search: For company research (e.g., Gemini API mentioned).

5. Non-Functional Requirements

- **Performance:** Real-time chat responses < 3 seconds. CV analysis for ~100 CVs completion target TBD.
- **Security:** HTTPS for data transfer, encryption for stored sensitive data (including all DB data), compliance with data protection laws (e.g., GDPR).
- **Usability:** Intuitive interface, user-friendly chat design.
- **Scalability:** Support multiple concurrent users and jobs. Architecture should allow for future module/integration expansion.
- **Reliability:** Robust testing (unit, integration, UAT), CI/CD pipeline.

6. Release Criteria / Success Metrics

- **Release Criteria:**
  - Successful completion of Unit, Integration, and User Acceptance Testing (UAT) as defined in Section 6 of the source document.
  - All MVP features (Project Prep, JD Gen, CV Analyzer, Profile Creation, Client Submission) are functional.
  - Integrations with Claap, Workable, and BrightHire are operational (using test/sandbox accounts initially).
  - Non-functional requirements (performance, security basics) are met.
- **Success Metrics (Examples - to be refined):**
  - Reduction in time spent on manual research and JD creation.
  - Reduction in time spent screening initial CV batches.
  - User satisfaction scores (from Recruiters).
  - Adoption rate among the recruitment team.
  - Qualitative feedback on efficiency gains.

7. Future Considerations

- Automated interview scheduling.
- Voice-based interface.
- Sentiment analysis on candidate transcripts.
- Support help desk/ticketing system.
- Integration with other ATS platforms.
- Native email sending capability.
- Direct posting to job boards. """

Create a file-like object

markdown_file = io.StringIO(prd_markdown)

Create a DataFrame (optional, but helps structure for file output)

Here, we'll just use the raw string content for the Markdown file.

file_name = "recruiter_assistant_prd.md"

The execution environment handles file output based on print statements

describing the file name and content. We will simulate this.

In a real environment, specific libraries/APIs would handle file creation.

Output the markdown content to be saved as a file

Note: This approach is conceptual. The actual file saving mechanism

depends on the execution environment's capabilities for generating files.

We structure the output to indicate a file should be created.

Print file metadata and content for the execution environment

print(f"--- START FILE: {file_name} ---") print(prd_markdown) print(f"--- END FILE: {file_name} ---") Code output --- START FILE: recruiter_assistant_prd.md ---

Product Requirements Document: Recruiter's Assistant Tool

1. Introduction

- **Purpose:** To automate and streamline common recruitment tasks using AI-driven insights and integrations with existing platforms (Claap, Workable, BrightHire). The tool aims to reduce recruiters' manual effort in research, job description generation, candidate analysis, and submission preparation, enabling them to focus on strategic activities.
- **Goals:**

    - Automate repetitive recruitment tasks.
    - Improve recruiter efficiency and productivity.
    - Provide AI-powered insights for better decision-making.
    - Integrate seamlessly with existing recruitment tools (Claap, Workable, BrightHire).
    - Standardize processes like job description creation and candidate profiling.

2. Target Audience

- **Primary Users:** Recruiters (access to all functionalities).
- **Secondary Users:**
    - Hiring Managers/Clients (receivers of candidate profiles/submission emails).
    - System Administrators (user management, integrations, maintenance).

3. Features (MVP Scope)

The MVP includes the following modules:

- **3.1 Project Preparation**

    - **Objective:** Equip recruiters with comprehensive information about the vacancy, client, market context, search strings, and keywords.
    - **Requirements:**
        - Upload/Input Job Description (client's version) and Intake Form data.
        - Integrate with Claap to view debrief meetings and analyze transcripts for key requirements. (Fallback: manual transcript upload if no native integration).
        - Automated company research (founding date, size, turnover, LinkedIn, website, news, reviews, competitors) using web search capabilities.
        - Provide salary comparisons for similar roles in the specific location.
        - Identify similar job postings locally.
        - AI-generated questions: Potential candidate questions and suggested recruiter screening questions.
        - AI-generated, customizable search strings: LinkedIn Boolean, Google X-ray for LinkedIn, Google X-ray for CVs/Resumes.
        - AI-generated keyword lists with synonyms and relevant translations.
        - Store all gathered data (company info, job profile, documents, search strings) linked to the job record.
        - UI: Step-by-step intake wizard, embedded/linked Claap content, dedicated output pages/tabs for research and AI-generated content.

- **3.2 Job Description Generator**

    - **Objective:** Generate standardized, complete job descriptions using data from Project Preparation.
    - **Requirements:**
        - AI generates a draft JD including role overview, responsibilities, requirements, compensation, benefits, location, and EEO

disclaimer.
        - AI generates concise, relevant screening questions based on established guidelines.
        - Allow recruiters to edit and finalize the generated JD and questions.
        - Export JD in various formats (PDF, Word) or potentially post to job boards (future integration).
        - Store final JDs for reference/reuse.
        - UI: Initiate generation from Project Preparation data, review/edit draft, save final version.

- **3.3 CV Analyzer**
    - **Objective:** Rank candidates against role requirements, allowing interactive refinement of criteria via chat.
    - **Requirements:**
        - Input CVs via manual upload (PDF/DOCX) or Workable integration (from "applied" stage). (Save option for Workable URLs).
        - Interactive chat interface for recruiters to add/remove/modify requirements (e.g., skills, experience).
        - AI recalculates candidate rankings in real-time based on refined requirements.
        - AI ranks CVs using a weighted scoring system (automatically defined weights, editable by recruiter).
        - Display the top candidates with scores/match percentages and criteria breakdown.
        - Persist refined requirements for the specific job record.
        - Store candidate resumes, info, and match scores.
        - UI: Import/fetch CVs, chat interface for requirements, automatically updating ranked list, detailed view per candidate.

- **3.4 Profile Creation**
    - **Objective:** Generate concise, client-facing candidate summaries using CV, interview data (BrightHire), and web presence.
    - **Requirements:**
        - Integrate with BrightHire to access interview recordings/transcripts and summarize key points.
        - Extract key data: Location, availability, salary, LinkedIn link, languages, education, career highlights, reasons for interest/leaving, potential concerns.
        - Allow definition of up to 5 custom headings based on client priorities (AI suggests, recruiter edits).
        - AI generates summaries (max 5 bullet points) for each custom heading.
        - Allow recruiters to edit and finalize the generated profile.
        - Store consolidated profile data.
        - UI: Select candidate, generate profile pulling from sources, add/edit custom headings, save final version.

- **3.5 Submission to Client**
    - **Objective:** Create a succinct, professional email body for candidate submission.

    - **Requirements:**
        - Generate email template using data from the created candidate profile.
        - Highlight candidate strengths and suitability briefly.
        - Allow recruiters to edit the generated text.
        - Provide options to copy text to an external email client or send via integrated mail service (optional, requires mailbox linking and signatures).
        - Use placeholders for client name, job title, candidate name, etc..
        - UI: Initiate from candidate profile, review/modify auto-generated email, send or copy.

4. Design & Architecture

- **UI/UX:** Web-based, responsive UI accessible via standard browsers. Separate pages/tabs for modules. Chat-like interface for CV Analyzer. Intuitive design with clear instructions.
- **Backend:** RESTful API or GraphQL suggested for data exchange. AI Integration Layer (NLP, summarization, text generation, ranking). Database for storing job descriptions, transcripts, candidate data, user prompts, outputs.
- **Integrations (MVP):**
    - Claap: Fetch recordings/transcripts (via API).
    - Workable: Access applicants/CVs from the "applied" stage (via API).
    - BrightHire: Access interview data/transcripts (via API).
    - AI/ML Services: Large language models for content generation, chat, summarization, etc..
    - Web Search: For company research (e.g., Gemini API mentioned).

5. Non-Functional Requirements

- **Performance:** Real-time chat responses < 3 seconds. CV analysis for ~100 CVs completion target TBD.
- **Security:** HTTPS for data transfer, encryption for stored sensitive data (including all DB data), compliance with data protection laws (e.g., GDPR).
- **Usability:** Intuitive interface, user-friendly chat design.
- **Scalability:** Support multiple concurrent users and jobs. Architecture should allow for future module/integration expansion.
- **Reliability:** Robust testing (unit, integration, UAT), CI/CD pipeline.

6. Release Criteria / Success Metrics

- **Release Criteria:**
    - Successful completion of Unit, Integration, and User Acceptance Testing (UAT) as defined in Section 6 of the source document.
    - All MVP features (Project Prep, JD Gen, CV Analyzer, Profile Creation, Client Submission) are functional.
    - Integrations with Claap, Workable, and BrightHire are operational (using test/sandbox accounts initially).
    - Non-functional requirements (performance, security basics) are met.

- **Success Metrics (Examples - to be refined):**
    - Reduction in time spent on manual research and JD creation.
    - Reduction in time spent screening initial CV batches.

    - User satisfaction scores (from Recruiters).
    - Adoption rate among the recruitment team.
    - Qualitative feedback on efficiency gains.

7. Future Considerations

- Automated interview scheduling.
- Voice-based interface.
- Sentiment analysis on candidate transcripts.
- Support help desk/ticketing system.
- Integration with other ATS platforms.
- Native email sending capability.
- Direct posting to job boards.