# Project Requirements Document

## 1. Project Overview

* **Project Name: CV Management System**
* **Client Name: DR/Walid Osman**
* **Date: 1/3/2025**
* **Prepared By: Gamal Walid**
* **Version: 1.0**
* **Objective: CV management system that saves CVs and sorts them.**

## 2. Functional Requirements

* **User Roles & Permissions:** 
  + **HR uploads CVs and searches for any CV.**
* **Core Features:** 
  + **CV store**
  + **Search**
  + **Charts and dashboards**
* **User Interface & Experience:** 
  + **Screen for CVs**
  + **Screen for dashboards**
  + **Screen for CV details**
* **Data Management:** 
  + **Fire store will be used for data management.**
* **Integrations:** 
  + **Google Cloud**
  + **Firebase**
  + **Python**
  + **Fire store**

## 4. Technical Requirements

* **Platform: Desktop**
* **Technology Stack: Flutter, Python, Fire store , google cloud services**
* **Hosting & Deployment: Google Cloud , fire store**

## 5. User Stories

* **As an HR, I want to upload a CV so that it is stored in the system.**
* **As an HR, I want to search for a CV so that it is easier to find the best candidate.**
* **As an HR, I want to see the details of a CV so that I can read the information in the CV.**
* **As an HR, I want to see dashboards so that I can view the total CVs and related information.**
* **As an HR, I want to filter CVs so that I can refine searches and find CVs with applied filters more easily.**
* **As an HR, I want to categorize CVs based on skills so that I can efficiently shortlist candidates.**
* **As an HR, I want to filter CVs by age, skills, and education so that I can efficiently narrow down candidates.**
* **As an HR, I want to process multiple CV formats (PDF, Word, plain text) so that I can upload CVs seamlessly.**
* **As an HR, I want to automatically extract key information from CVs so that I can save time on manual entry.**
* **As an HR, I want a user-friendly search interface so that I can quickly locate relevant CVs.**
* **As an HR, I want to split my CVs into parts for the CVs, assigned CVs and archived CVs**

## 6. Acceptance Criteria

* **The first version of the application is accepted by the customer.**

## 7. Deliverables & Timeline

Phase 1: Planning and Requirements

* + **Define the scope: Clarify CV formats (PDF, Word, plain text) and required features (filtering by age, skills, education).**
  + **Data standards: Decide on a structured format (JSON, XML, relational database) and create a schema.**
  + **Tool selection: Choose technologies for NLP, database, UI, and backend.**

Phase 2: Backend and Core Functionality

* + **File reader: Handle multiple file formats using libraries like Apache Tika, PyPDF2, or python-docx.**
  + **Data extraction and parsing: Use regex for patterns (e.g., extracting dates for age calculation) and train NLP models for skills and experience identification.**
  + **Data storage: Set up fire store and develop a process for inserting extracted CV data.**
  + **Search algorithm: Optimize queries or use Elasticsearch for fast searches with advanced filters.**
  + **API development: Design RESTful for data retrieval, filtering, and updates.**

Phase 3: Frontend and User Interface

* + **UI design: Create a layout for CV uploads, management, and searching.**
  + **Implement UI: Build upload features, search filters.**
  + **Testing UI: Ensure cross-browser/device compatibility and usability validation.**

Phase 4: Quality Assurance and Testing

* + **Unit testing: Test individual components (file reader, NLP extraction, etc.).**
  + **Integration testing: Verify seamless functionality across components.**
  + **Performance testing: Test scalability with large datasets (50+ CVs).**
  + **Error handling: Address corrupted files, unsupported formats, and missing data.**

Phase 5: Deployment and Maintenance

* + **Deployment: Deploy backend and frontend on cloud services like google cloud**
  + **Monitoring: Implement logging and error tracking.**
  + **Documentation: Provide user guides and technical documentation.**

## 8. Assumptions & Constraints

* **The budget is constrained due to the use of resources without any income.**

## 9. Risks & Mitigation Strategies

* **Risk 1: The application may not function properly.** 
  + **Mitigation: Ensure the application runs without errors by having Python installed and properly configured.**
* **Risk 2: The application may not function properly.** 
  + **Mitigation: Ensure the device connected to network .**

## 10. Approval & Sign-off

* **Client Name & Signature: [Signature]**
* **Date: [Enter date]**