**Main Requirements by components:**

Input handling:

* We need a user interface for recruiters to input criteria (skills, description, age range, gender).
* We need to handle file uploads for CVs, likely in various formats (PDF, DOCX, etc.).

CV analysis and matching:

* We need to implement text processing and analysis to extract relevant information from CVs.
* We need to develop a matching algorithm to compare CV content with recruiter criteria.
* We need to implement a ranking system for partial matches.

Database creation and management:

* We need to design a database schema to store extracted CV information.
* We need to implement a system to update this database as new CVs are processed.

User interface:

* We need to create a main page with input forms and buttons for various functions.
* We need to implement a results display with clickable links to original CVs.
* We need to create a separate view for the database contents.

Dashboard:

* We need to design and implement a dynamic dashboard with relevant metrics and visualizations.

**Key decisions to make:**

Application type:

* Will this be a web application, desktop application, or something else?

Technology stack:

* Frontend: What framework will we use for the UI (e.g., React, Vue, Angular)?
* Backend: What language/framework will we use (e.g., Python with Flask/Django, Node.js with Express)?
* Database: What type of database will we use (e.g., SQL, NoSQL)?

CV processing:

* What libraries will we use for text extraction from different file formats?
* How will we implement the text analysis and matching algorithms?

Deployment:

* How and where will the application be hosted?

Scalability:

* How will we ensure the application can handle a growing number of CVs and users?

Security:

* How will we handle data privacy and security, especially considering the sensitive nature of CV information?

Testing:

* What testing strategy will we implement to ensure the application's reliability?