**Inputs**: Users upload multiple documents.

**Processing**:

* Extract text from documents.
* Convert text into meaningful representations.
* Cluster documents based on similarity.
* Generate summaries for each cluster.

**Outputs**: Clustered documents and summary reports for each group.

**Web Interface**: User-friendly UI for document upload and retrieval.

**Backend**: Python with FastAPI (for handling requests efficiently).

**NLP & ML**:

* NLTK for text processing.
* Word Embeddings for vectorization.
* K-Means, Hierarchical Clustering, or DBSCAN for clustering.(which will provide great result).
* BERT or TextRank for summarization.

**Database**: PostgreSQL (for document storage and retrieval).

**Frontend**: React.js with Next.js (for a responsive UI).

Requirements Installation: -

* **python-docx** → For extracting text from .docx files (Microsoft Word).
* **pdfplumber** → For extracting text from .pdf files.
* **nltk** → For natural language processing tasks.

**Step 1: -** pip install python-docx pdfplumber nltk

Libraries :-

**os** → Helps in handling file paths.

**docx** → Reads .docx files.

**pdfplumber** → Reads .pdf files.

**nltk.tokenize.word\_tokenize** → Splits text into words.

**nltk.corpus.stopwords** → Contains common words to be removed (like "is", "the").

**nltk.stem.WordNetLemmatizer** → Converts words to their root form (e.g., "running" → "run").