Product Requirements and Specification Document

Project Name

DocuCluster - Hierarchical Document Clustering Tool

Description

DocuCluster is a web-based tool for hierarchical clustering of uploaded text documents. Users can upload documents, select clustering parameters, visualize dendrograms, and export cluster assignments. The tool is built with Flask and leverages Scikit-learn for clustering.

---

1. Goals & Objectives

Goal	Description
Document Clustering	Enable users to cluster uploaded text documents hierarchically
Visualization	Provide interactive dendrogram visualization
Parameter Selection	Allow selection of linkage methods and number of clusters
Export	Enable export of cluster assignments in CSV format
Accessibility & Usability	Simple, intuitive web interface for educational use
Open Source	Codebase to be open and well-documented

---

2. Core Features

Feature	Description
Document Upload	Upload multiple text files (e.g., .txt, .pdf)
Preprocessing	Text cleaning, tokenization, vectorization (TF-IDF)
Clustering	Hierarchical clustering using Scikit-learn (AgglomerativeClustering)
Linkage Selection	User selects linkage method: single, complete, average, ward
Dendrogram	Interactive dendrogram visualization (e.g., using D3.js or Plotly)
Cluster Assignment	Display and export document-cluster mapping as CSV
Basic Authentication	Optional: Simple login for user sessions

---

3. User Stories

As a…	I want to…	So that…
User	Upload a set of documents	I can analyze their similarity

User	Select clustering parameters	I can customize the clustering
User	Visualize the dendrogram	I can explore document relationships
User	Export cluster assignments	I can use results elsewhere
Developer	Access open-source code	I can contribute or adapt the tool

---

4. Functional Requirements

4.1 Document Upload

- Accept multiple files (.txt, .pdf)
- Max file size: 5MB per file
- Validate file types and sizes

4.2 Preprocessing

- Convert PDFs to text (if applicable)
- Clean and tokenize text
- Vectorize using TF-IDF (Scikit-learn)

4.3 Clustering

- Use AgglomerativeClustering (Scikit-learn)
- User selects linkage: `single`, `complete`, `average`, `ward`
- User sets number of clusters (optional; default: auto-determined)

4.4 Visualization

- Generate dendrogram from clustering results
- Interactive: zoom, select clusters
- Display document names/IDs on dendrogram

4.5 Export

- Export cluster assignments as CSV: `document,cluster_id`
- Download link available after clustering

4.6 Authentication (Optional)

- Simple username/password login (Flask-Login)
- Session management

---

5. Non-Functional Requirements

Requirement	Specification
Performance	Clustering for up to 100 documents < 10s
Usability	Clean, responsive UI (desktop & mobile)
Security	Input validation, secure file handling
Compatibility	Python 3.8+, modern browsers

Documentation	User guide and developer README
Licensing	MIT License

---

6. Technology Stack

Component	Technology
Backend	Python, Flask
ML/Clustering	Scikit-learn, Numpy, Pandas
Frontend	HTML, CSS, JS, Bootstrap
Visualization	D3.js or Plotly.js
File Handling	Flask, PyPDF2 (for PDFs)
Authentication	Flask-Login (optional)

---

7. API & Data Flow

7.1 High-Level Flow

```
graph TD
A[User Uploads Files] --> B[Preprocessing]
B --> C[Clustering]
C --> D[Dendrogram Visualization]
C --> E[Cluster Assignment Export]
```

7.2 Key Endpoints

Endpoint	Method	Description
`/upload`	POST	Upload documents
`/cluster`	POST	Run clustering, return results
`/dendrogram`	GET	Serve dendrogram visualization
`/export`	GET	Download cluster assignments (CSV)

---

8. UI/UX Requirements

- **Upload Page:** Drag-and-drop or file picker, file list, upload button
- **Parameter Selection:** Dropdowns for linkage, input for cluster count
- **Results Page:** Dendrogram visualization, cluster assignment table, export button
- **Navigation:** Simple, minimal header/footer

---

9. Acceptance Criteria

ID	Criteria
AC1	User can upload and preprocess at least 10 documents
AC2	User can select linkage method and cluster count
AC3	Dendrogram is generated and interactive
AC4	Cluster assignments are displayed and exportable as CSV
AC5	All core features work on Chrome, Firefox, Edge
AC6	Codebase is open-source and documented

---

10. Out of Scope

- Real-time collaboration
- Advanced NLP (topic modeling, embeddings)
- User management beyond basic authentication
- Large-scale document handling (>100 docs)

---

11. Milestones

Milestone	Target Date
Requirements Finalized	Week 1
Core Backend Complete	Week 2
Frontend & Visualization	Week 3
Export & Auth Features	Week 4
Testing & Documentation	Week 5
Release	Week 6

---

12. Appendix

- **References:** Scikit-learn documentation, Flask docs
- **License:** MIT

End of Document