Intelligent Document Processing (MERN)

Short title: MERN — Intelligent Document Processing Pipeline

Audience: Developers (MERN), data scientists (spaCy, Hugging Face), DevOps, QA, product managers.

---

1. Project Summary (Abstract)

This project builds an end-to-end intelligent document processing pipeline that extracts, understands, and analyzes information from PDF files. It integrates OCR and layout-aware extraction to retrieve both raw text and tabular data. Extracted content is cleaned, tokenized, and structured to preserve meaning and formatting. NLP and Transformer-based models (spaCy + Hugging Face) detect semantics, keywords, and entities. A ChatGPT-powered query system allows context-aware user queries with human-like answers. Automatic report generation (tables, charts) and export to PDF/Word are included. Extra features: voice responses, anomaly detection, visual summaries. Designed for researchers, analysts, and organizations that process large volumes of unstructured documents.

---

2. Technology Stack (MERN + ML)

- **Frontend:** React, Redux or Zustand, Tailwind CSS, Chart.js
- **Backend:** Node.js + Express.js (REST or GraphQL)
- **Database:** MongoDB (Atlas or self-hosted)
- **File storage:** AWS S3 / MinIO
- **Queue:** Redis + BullMQ or RabbitMQ
- **OCR & layout:** layoutparser + Tesseract
- **Table extraction:** Camelot, Tabula
- **NLP:** spaCy, Hugging Face Transformers
- **Semantic search:** Pinecone / Milvus / Weaviate
- **Chat:** ChatGPT API or local LLMs
- **Auth:** JWT + OAuth2
- **DevOps:** Docker, Kubernetes, GitHub Actions

---

3. Key Features & Flow

1. **Upload & Ingest:** Users upload PDFs; stored in object storage with metadata in MongoDB.
2. **Preprocessing & OCR:** Workers extract text or perform OCR if scanned.
3. **Table Extraction:** Structured tables are extracted and stored as JSON or CSV.
4. **Cleaning & Normalization:** Removes noise and maintains structure.
5. **NLP Processing:** Detects entities, summarizes, and extracts keywords.
6. **Embeddings & Semantic Indexing:** Embeddings stored in a vector database for intelligent retrieval.
7. **Chat & Query System:** Context-based question answering with precise results.
8. **Reporting & Exports:** Auto-generated reports, charts, and exports to PDF/Word.
9. **Extras:** Voice support, anomaly detection, and visual dashboards.

4. Data Models (MongoDB Collections)

Collections: Users, Documents, Pages, Extractions, and Vectors — store metadata, extracted text, structured data, and embeddings.

---

5. API Design (REST Examples)

- **Authentication:** Register, login, token refresh.
- **Document Upload:** Upload PDFs, check processing status.
- **Query System:** Contextual Q&A using semantic search.
- **Report Exports:** Generate and download results in PDF/Word formats.

---

6. Frontend (React) — Key Screens

- Dashboard: Recent documents and stats.
- Upload: Drag-and-drop PDF upload.
- Document Viewer: Interactive view with extracted data.
- Table Viewer: View and export structured tables.
- Chat Interface: Contextual Q&A.
- Reports Page: Generated summaries and visualizations.
- Admin Panel: User and access management.

---

7. Security & Privacy

- HTTPS, JWT, and secure S3 buckets.
- Role-based access control.
- Data encryption and compliance with privacy policies.
- Audit logging for document access and processing.

---

8. Performance & Scaling

- Scalable vector database and worker-based architecture.
- Batching for embeddings and caching for frequent queries.
- GPU-enabled inference for transformers.

---

9. Deployment & DevOps

- Dockerized microservices for frontend, backend, and workers.
- Kubernetes for scalable deployment.
- CI/CD pipelines with GitHub Actions.
- Centralized secrets and config management.

10. Testing Strategy

- Unit, integration, and end-to-end testing.
- Validation of OCR and NLP outputs.
- Automated regression testing for document extraction quality.

---

11. Roadmap & Prioritization

MVP: PDF upload, OCR, text extraction, NER, and basic Q&A.

v1: Vector DB integration, report generation, and user management.

v2: Voice responses, anomaly detection, and multi-language support.

---

12. Deliverables Checklist

- React frontend with viewer and chat
- Express backend API
- OCR and NLP workers
- MongoDB schema setup
- Vector DB integration
- Reporting and export system
- Docker and CI/CD configuration
- Comprehensive testing

This document presents a clean, end-to-end blueprint for developing an Intelligent Document Processing system using the MERN stack integrated with advanced NLP and OCR models.