Article Draft MVP

A **Human in the Loop (HITL)** article drafting application that transforms interview transcripts and supporting sources into compelling, well-sourced articles using AI assistance. Designed as a **polished MVP**.

Problem Framing & Assumptions

Problem

Content creators spend hours manually extracting key points from interview transcripts, mapping content to sources, and verifying quotes. The process is time-consuming, error prone, and lacks systematic source verification.

Assumptions

- **Editor Control**: Human editors need full control over story direction and content approval.
- **Source Integrity**: Every claim must be traceable to its source with confidence scoring.
- **AI Assistance**: LLMs accelerate content analysis but require human oversight.
- **Flexible Input**: Support for multiple source types (PDFs, web articles, YouTube transcripts).
- **Export Ready**: Final output must be publication-ready with proper attribution

Architecture

Frontend (React + Vite)

```
src/
├── components/
│   ├── ProjectSetup.jsx        # Project creation & transcript/source
upload
│   ├── KeyPointsExtraction.jsx # AI-powered key point extraction
│   ├── StoryDirection.jsx      # Tone, angle, length configuration
│   ├── DraftGeneration.jsx     # AI article generation
│   ├── SourceMapping.jsx       # Source mapping & quote verification
│   ├── Export.jsx              # Markdown export with provenance
│   └── PromptEditor.jsx        # Editable AI prompts
├── services/
│   └── api.js                  # Backend API communication
└── App.jsx                     # Main application state management
```

Backend (Node.js + Express)

```
backend/
├── server.js                   # Express server with AI integration
├── package.json                # Dependencies (Gemini/OpenAI, file
processing)
└── .env                        # API keys and configuration
```

AI Integration

- **Primary**: Google Gemini (free tier)
- **Fallback**: OpenAI GPT - 4 (paid)
- **Local Fallback**: Pattern matching for offline or API - limited scenarios

Core Workflow

1. **Project Setup**
    - Create a project with name and transcript.
    - Upload or attach supporting sources (PDF, web, YouTube).
    - Extract text content from sources.
2. **Key Points Extraction**
    - AI analyzes transcript + sources to generate key points with confidence scores.
    - Editor can approve, edit, reorder, or add custom points.
    - Edited/custom points are visually highlighted.
3. **Story Direction**
    - Set tone (professional, casual, technical).
    - Choose angle (founder story, technical deep dive, etc.).
    - Configure length (short, medium, long) or add custom instructions.
4. **Draft Generation**
    - AI generates article using approved points and direction.
    - Supports outline or full draft generation.
    - Maintains source attribution.
5. **Source Mapping & Quote Verification**
    - Maps each paragraph to its most likely source.
    - Highlights and verifies quotes against source snippets.
    - Provides confidence scores for all mappings.
6. **Export**
    - Generates Markdown output.
    - Includes optional provenance JSON mapping paragraphs/quotes to sources.
    - Ready for publication or further editing.

Key-Point Approval, Source Mapping & Quote Checks

Key-Point Approval

- AI extracts key points from transcript + sources.
- Editor reviews, edits, adds custom points, and reorders them via drag and drop.
- Clear visual indicators distinguish AI generated, edited, and custom points.

Source Mapping

- Paragraph-level mapping to sources.
- Confidence scoring (0 - 100%) for mappings.
- Displays source snippets for verification.

Quote Verification

- Detects multiple quote formats (“…”, ‘…’)
- Exact and partial match detection.
- Provides surrounding context for verification.
- Confidence scoring for each match.

Trade-offs & Future Improvements

Current Trade-offs

- **AI Dependency**: Requires backend for full functionality, pattern matching fallback exists.
- **Source Processing**: Limited to text extraction, no images/tables.
- **Collaboration**: Single user only, no multi-editor support.
- **Version Control**: No article versioning or checkpoint tracking.

With More Time

- OCR for images and table extraction.
- Multi-editor collaboration with comment and approval workflow.
- Fine-tuned AI for domain specific quote accuracy.
- Caching, lazy loading, and performance improvements.
- Analytics for source usage and editor patterns.

Getting Started

Prerequisites

- Node.js 16+
- npm or yarn
- Google Gemini API key (free) or OpenAI API key

Installation

```
Clone repository
git clone <repository-url>
cd article-draft-mvp

Install dependencies
npm install

Set up environment variables
cp backend/env.example backend/.env
Edit backend/.env with your API keys

Start development servers
npm run dev:full
```

Environment Variables

```
backend/.env
USE_GEMINI=true
GEMINI_API_KEY=your_gemini_key_here
OPENAI_API_KEY=your_openai_key_here
PORT=5000
```

Testing

Sample Inputs

- **Transcript**: Tech startup founder interview (3,000 words)
- **Sources**: Company website article, product brochure PDF, YouTube interview video

Test Scenarios

1. Complete workflow (transcript -> article -> export).
2. Verify all claims map to sources.
3. Confirm quotes match original text.
4. Handle invalid inputs gracefully.

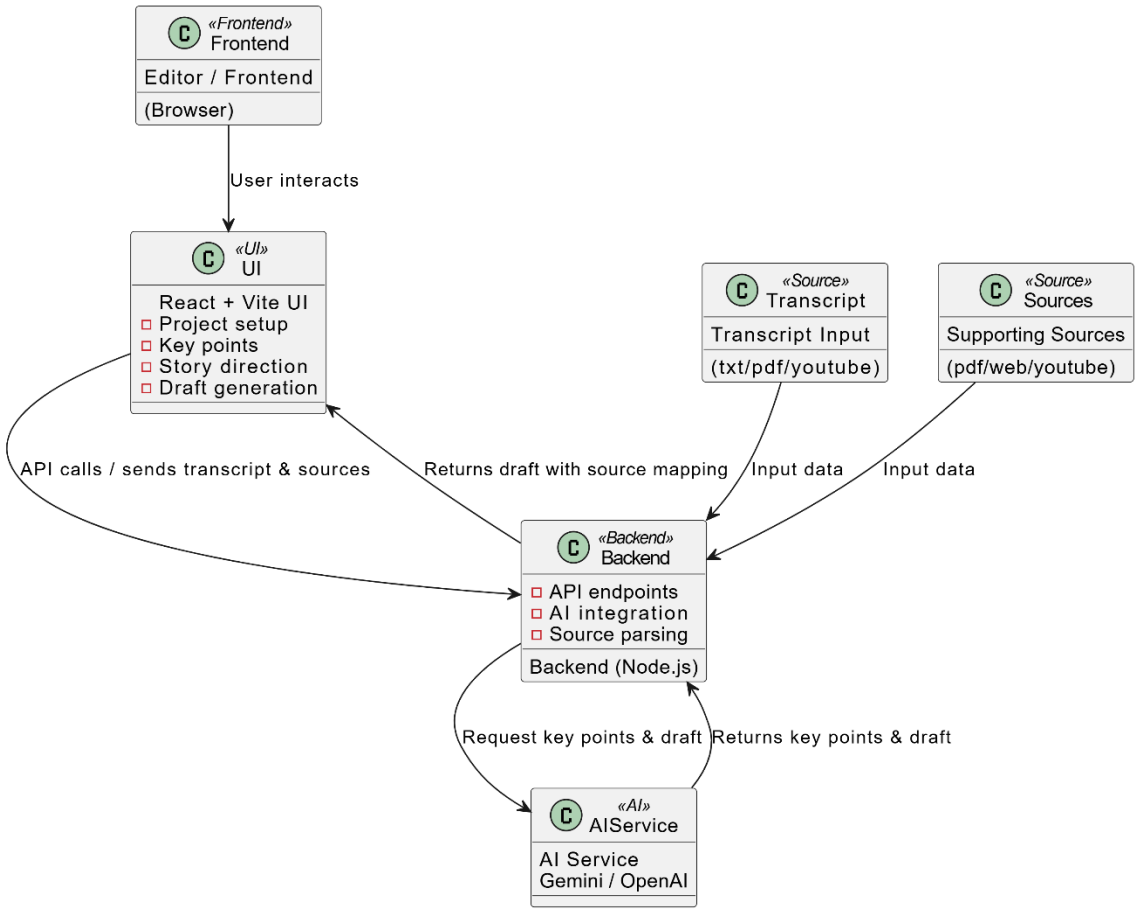
Performance

- Supports transcripts >=10k characters.
- Handles multiple PDFs, web articles, and YouTube sources.
- AI extraction: 2 - 5 seconds; full draft: 10 - 30 seconds.
- Optimized for browser memory usage.

Security & Privacy

- API keys stored in `.env`; never committed.
- Processing happens locally or via secure APIs.
- No persistent storage of user data.
- Uses **public sources only**.

Architecture Diagram



Application Video DEMO Link

https://drive.google.com/file/d/1IWpXa-2BRJ7dGrB6cXc1xSJtIY3k9ho7/view?usp=drive_link