**Cursor Session Strategy for Mingus Article Library Implementation**

**🎯 RECOMMENDED APPROACH: Separate Sessions**

**YES, create a new Cursor session for each step.** This pipeline is designed for **file-based dependencies**, not code-context dependencies.

**🔄 Why Separate Sessions Work Better**

**✅ Focused Context**

* Each session concentrates on **one specific task**
* Cursor generates more targeted, accurate code
* No context pollution from unrelated previous steps
* Cleaner prompt processing and response quality

**✅ File-Based Dependencies**

The steps connect through **output files**, not shared code:

Step 1 → raw\_urls\_complete.csv → Step 2

Step 2 → domain\_recommendations.json → Step 3

Step 3 → approved\_domains.txt → Step 4

Step 4 → scraped\_articles.csv → Step 5

Step 5 → classified\_articles.json → Step 6

**✅ Error Isolation**

* Problems in one step don't affect others
* Easy to debug and fix specific components
* Can re-run individual steps without starting over
* Maintain working components while fixing issues

**📋 SESSION WORKFLOW STRATEGY**

**SESSION 1: Email Extraction**

**Focus:** Extract URLs from .mac email **Prompt:** Step 1 prompt **Output Files:** raw\_urls\_complete.csv, domain\_analysis\_report.csv **Session Goal:** Get all URLs discovered and analyzed

**Close session after:** Files are generated and validated

**SESSION 2: Domain Intelligence**

**Focus:** Analyze domains and generate recommendations **Prompt:** Step 2 prompt **Input:** Load files from Session 1 **Output Files:** domain\_intelligence\_report.html, domain\_recommendations.json **Session Goal:** Smart domain categorization and recommendations

**Close session after:** Intelligence report generated

**SESSION 3: Domain Approval Interface**

**Focus:** Web interface for domain approval decisions **Prompt:** Step 3 prompt  
**Input:** Load files from Session 2 **Output Files:** approved\_domains.txt, domain\_decisions.json **Session Goal:** Visual approval interface working

**Close session after:** Approved domain list finalized

**SESSION 4: Article Scraping**

**Focus:** Scrape content from approved domains **Prompt:** Step 4 prompt **Input:** Load approved domains and original URLs **Output Files:** scraped\_articles.csv **Session Goal:** Full article content extracted

**Close session after:** Article scraping complete

**SESSION 5: AI Classification**

**Focus:** OpenAI Be-Do-Have classification **Prompt:** Step 5 prompt **Input:** Load scraped articles **Output Files:** classified\_articles.json **Session Goal:** All articles classified with AI

**Close session after:** Classification complete

**SESSION 6: Database Models**

**Focus:** SQLAlchemy models and migrations **Prompt:** Step 6 prompt **Input:** Reference classified article structure **Output Files:** Model files and migrations **Session Goal:** Database schema ready

**Close session after:** Models and migrations created

**SESSION 7: Backend APIs**

**Focus:** Flask API endpoints **Prompt:** Step 7 prompt **Input:** Reference database models **Output Files:** API route files **Session Goal:** REST API functionality complete

**Close session after:** All endpoints working

**SESSION 8: Frontend Components**

**Focus:** React UI components  
**Prompt:** Step 8 prompt **Input:** Reference API endpoints **Output Files:** React component files **Session Goal:** User interface complete

**Close session after:** UI components functional

**SESSION 9: Configuration Setup**

**Focus:** Environment and deployment config **Prompt:** Step 9 prompt **Input:** Reference all previous components **Output Files:** Config files and environment setup **Session Goal:** Deployment-ready configuration

**Close session after:** Configuration complete

**SESSION 10: Testing Suite**

**Focus:** Comprehensive testing **Prompt:** Step 10 prompt (abbreviated) **Input:** Reference all components **Output Files:** Test files and coverage reports **Session Goal:** Full testing infrastructure

**Close session after:** Testing suite complete

**SESSION 11: Data Import**

**Focus:** Import classified articles to database **Prompt:** Custom import script prompt **Input:** Load classified articles **Output:** Database populated with articles **Session Goal:** Production data ready

**Close session after:** Articles imported successfully

**SESSION 12: Analytics Dashboard**

**Focus:** Admin analytics and monitoring **Prompt:** Analytics dashboard prompt  
**Input:** Reference database and user interactions **Output Files:** Analytics components **Session Goal:** Admin monitoring complete

**📂 File Management Between Sessions**

**Create Data Directory Structure:**

mingus\_project/

├── data/

│ ├── step1\_email\_extraction/

│ ├── step2\_domain\_analysis/

│ ├── step3\_domain\_approval/

│ ├── step4\_article\_scraping/

│ └── step5\_ai\_classification/

├── scripts/

├── backend/

├── frontend/

└── config/

**Session Handoff Checklist:**

Before starting each new session:

✅ **Verify previous step outputs exist** ✅ **Note any file path changes needed**  
✅ **Prepare input requirements for current step** ✅ **Have previous session's output file locations ready**

**🎯 Session Management Best Practices**

**Start Each Session With:**

Context: This is Step X of 12 in the Mingus article library implementation.

Previous Step Output: I have these files ready:

- file1.csv from Step X-1

- file2.json from Step X-1

Current Goal: [Specific objective for this session]

Integration Notes: [How this connects to the overall system]

**End Each Session With:**

* ✅ Test the generated code works
* ✅ Verify output files are created correctly
* ✅ Note any modifications needed for next step
* ✅ Save file locations for next session

**⚡ Quick Session Strategy**

**For Maximum Efficiency:**

1. **Prepare each prompt in advance** - Copy the step prompt to a text file
2. **Have input files ready** - Know exactly what files you're loading
3. **One session = One working component** - Don't try to do multiple steps
4. **Test immediately** - Verify each step works before moving on
5. **Document file locations** - Keep track of outputs for next session

**🔧 Example Session Handoff**

**Ending Session 1:**

✅ COMPLETED: Email extraction from .mac account

📁 Generated Files:

- data/raw\_urls\_complete.csv (1,247 URLs)

- data/domain\_analysis\_report.csv (89 unique domains)

- logs/extraction\_log.txt

🔄 NEXT SESSION: Load these files into Step 2 domain intelligence analysis

**Starting Session 2:**

CONTEXT: Step 2 of Mingus article library - Domain Intelligence Analysis

INPUT FILES FROM STEP 1:

- data/raw\_urls\_complete.csv (1,247 URLs extracted)

- data/domain\_analysis\_report.csv (89 domains to analyze)

GOAL: Generate domain intelligence report and recommendations for approval interface

[Paste Step 2 prompt here]

This approach keeps each session **focused, efficient, and error-free** while building toward your complete article library system.