Paper Summarizer - Design Document

Project Summary

The Paper Summarizer is a web-based tool that transforms lengthy, complex documents into clear, concise summaries by eliminating unnecessary fluff and extracting key points. This product matters because it saves time for researchers, students, and professionals while making information more accessible to people with attention deficits, dyslexia, and other learning differences who struggle to process dense text on their first read.

Problem Statement

Many people struggle to condense lengthy papers and documents into essential information, either because they lack the skill to identify key points, include too much unnecessary detail in their writing, or have difficulty processing dense text due to learning differences or time constraints.

Use Case

Students, researchers, professionals, and individuals with neurological disabilities or attention deficits will use this product to quickly understand the core concepts of lengthy documents without reading through excessive detail. Users will upload or paste their document into the interface, and the tool will automatically generate a simplified summary that highlights main ideas, making complex material immediately comprehensible and saving valuable time for other tasks.

Goals and Objectives

1. **Reduce reading time by 70%**: Enable users to understand the essential content of any document in a fraction of the time it would take to read the full text.
2. **Improve accessibility**: Make information comprehensible for people with dyslexia, ADHD, and other learning differences by presenting content in clear, straightforward language without jargon or complexity.

Key Features and Functions

1. **Intelligent Text Summarization**: Automatically analyzes uploaded documents (PDF, Word, or plain text) and extracts the most important sentences and concepts while removing redundant information and fluff.
2. **Adjustable Summary Length**: Users can select summary depth (brief, moderate, or detailed) based on their needs, with the tool dynamically adjusting output length from 10% to 50% of the original document.
3. **Highlight Key Concepts**: Automatically identifies and emphasizes critical terms, main arguments, and conclusions using visual formatting to help users quickly locate the most important information.

Tech Stack and Tools

- **Frontend**: React.js for user interface, Tailwind CSS for styling
- **Backend**: Node.js with Express.js framework
- **AI/ML**: Anthropic Claude API for natural language processing and summarization
- **File Processing**: pdf-parse for PDF extraction, mammoth for Word documents, multer for file uploads
- **Database**: PostgreSQL for storing user history and preferences
- **Hosting**: Vercel (frontend), Railway (backend and database)
- **Authentication**: Auth0 for user management (optional for saved summaries)
- **Version Control**: GitHub

Algorithm

```
START
1. Receive document input (file upload or text paste)
2. Extract raw text from document format
3. Preprocess text:
   - Remove headers/footers/page numbers
   - Identify paragraph boundaries
   - Detect section headings
4. Send text to Claude API with summarization prompt:
   - Specify desired summary length
   - Request key concept extraction
   - Emphasize clarity and simplicity
5. Receive AI-generated summary
6. Post-process summary:
   - Format for readability
   - Highlight key terms
   - Add section breaks if applicable
7. Display summary to user with options:
   - Adjust summary length
   - Download as new document
   - Copy to clipboard
```

```
8. (Optional) Save to user history
END
```

Flowchart

```
┌─────────────────┐
│  User Opens App │
└────────┬────────┘
         │
         ▼
┌─────────────────────┐
│ Upload/Paste Document│
└────────┬────────────┘
         │
         ▼
┌─────────────────────┐
│ Extract Text Content │
└────────┬────────────┘
         │
         ▼
┌─────────────────────┐
│  Select Summary     │
│  Length Preference  │
└────────┬────────────┘
         │
         ▼
┌─────────────────────┐
│ Process Text with   │
│   Claude API        │
└────────┬────────────┘
         │
         ▼
┌─────────────────────┐
│ Generate Summary &  │
│ Highlight Key Points│
└────────┬────────────┘
         │
         ▼
┌─────────────────────┐
│  Display Summary    │
│   to User           │
└────────┬────────────┘
         │
```

```
       ▼
   ┌─────┴─────┐
   │     │
   ▼     ▼
┌──────────┐ ┌────────────┐
│Download│ │Copy/Edit │
│Summary │ │ Summary  │
└──────────┘ └────────────┘
```

Timeline

Month 1: Planning & Setup

- Milestone: Complete design document, set up development environment, create basic React frontend structure, establish Claude API connection

Month 2: Core Development

- Milestone: Implement file upload functionality, integrate PDF and Word document parsing, build Claude API summarization pipeline

Month 3: Feature Enhancement

- Milestone: Add adjustable summary length controls, implement key concept highlighting, create user-friendly interface design

Month 4: Testing & Launch

- Milestone: Conduct user testing with target demographics (students, professionals, people with learning differences), fix bugs, deploy to production, create documentation

Risk Mitigation

Risk: API costs could become prohibitive with high user volume, as Claude API charges per token processed.

Mitigation Plan: Implement a tiered usage model with free users limited to 5 summaries per day (approximately 50,000 tokens), caching common documents to avoid reprocessing, and displaying estimated cost before processing large documents. Monitor API usage through a dashboard and set up automatic alerts when daily spending exceeds $50. Additionally, explore alternative summarization models or hybrid approaches (traditional NLP for initial reduction, AI for final polish) to reduce dependency on expensive API calls.

Evaluation Criteria

1. **Summary Quality Score**: At least 85% of users rate summaries as "accurate and useful" in capturing the original document's key points through post-use surveys.
2. **Time Savings**: Users report saving an average of 15+ minutes per document compared to reading the full text, measured through in-app time tracking and user feedback.
3. **User Retention**: 60% of users return to use the tool at least three times within their first month, indicating the product provides genuine value and solves their problem effectively.

Future Considerations

Maintenance Need: Regular updates to the Claude API integration will be necessary as Anthropic releases new models and deprecates older versions, requiring quarterly reviews of API documentation and testing to ensure continued functionality and optimal performance.

Future Functionality: Add collaborative features that allow multiple users to annotate and discuss summaries together, enabling study groups and research teams to collectively analyze documents. This could include shared workspaces, comment threads on specific summary sections, and the ability to compare different summary lengths side-by-side to facilitate deeper understanding and group learning.