AI Response Synthesis Tool

A powerful web application that analyzes and synthesizes responses from multiple AI language models (LLMs) into a comprehensive, superior analysis. This tool leverages Claude 4.0 Sonnet to evaluate, compare, and combine insights from ChatGPT, Gemini, Perplexity, CoPilot, DeepSeek, and other AI systems.

🎯 Purpose

When you have the same question answered by multiple AI systems, this tool helps you:

- **Evaluate each response** across 6 quality dimensions with numerical scores
- **Identify convergence and divergence** points between different AI models
- **Synthesize insights** into a comprehensive analysis that's better than any individual response
- **Attribute insights** to specific AI models for actionable decision-making
- **Resolve conflicts** between contradicting recommendations using evidence-based reasoning

✨ Key Features

🏷️ LLM-Specific Labeling

- Choose the source AI for each response (ChatGPT, Claude, Gemini, CoPilot, Perplexity, DeepSeek)
- Output clearly identifies which AI provided which insights
- No more generic "Response A" labels - see exactly what each AI contributed

📊 Comprehensive Analysis Framework

- **Individual Response Evaluation**: 6-dimension scoring (Accuracy, Completeness, Clarity, Creativity, Relevance, Evidence Quality)
- **Comparative Analysis**: Convergence points, divergence points, unique contributions, gap analysis
- **Quality Assessment Matrix**: Visual scoring comparison across all responses
- **Synthesis Decision Log**: Transparent reasoning for how conflicts were resolved

🎨 Rich Text Output

- Beautifully formatted analysis with proper headers, bold text, and bullet points
- Easy-to-scan results with professional typography
- Copy-to-clipboard functionality preserves original markdown formatting

⚙️ Customizable Meta-Prompt

- External `meta-prompt.txt` file for easy modification
- Adjust evaluation criteria, output structure, and analysis techniques
- No code changes needed - just edit the text file

🛡️ Robust Architecture

- Local Node.js server bypasses browser CORS restrictions
- Secure API key storage (local only)
- Comprehensive error handling and validation
- Real-time debugging and logging

🚀 Quick Start

Prerequisites

- Node.js (Download from nodejs.org)
- Claude API key (Get from Anthropic Console)

Installation

1. **Clone or download the project files**
2. **Create project directory:**

   bash

   ```bash
   mkdir ai-synthesis-tool
   cd ai-synthesis-tool
   ```

3. **Install dependencies:**

   bash

   ```bash
   npm init -y
   npm install express cors
   ```

4. **Set up project structure:**

```
ai-synthesis-tool/
├── server.js           # Backend server
├── meta-prompt.txt     # Synthesis instructions
├── public/
│   └── index.html      # Web interface
├── package.json        # Node.js dependencies
└── node_modules/       # Installed packages
```

5. **Start the server:**

   bash

   ```bash
   node server.js
   ```

6. **Open the application:** Navigate to `http://localhost:3000` in your browser

📋 How to Use

Step 1: Prepare Your Data

- Take the same prompt/question to multiple AI systems
- Copy each AI's response for comparison
- Have your Claude API key ready

Step 2: Input Your Data

1. **Enter your original prompt** in the first text area
2. **Paste each AI response** in the response sections
3. **Select the source AI** from the dropdown for each response
4. **Enter your Claude API key** (saved locally for future use)

Step 3: Add More Responses (Optional)

- Click "Add Another Response" to include up to 5 different AI responses
- Each new response gets its own LLM selector dropdown
- Remove responses you don't need with the "Remove" button

Step 4: Synthesize

- Click "🔄 Synthesize Responses"
- Watch the loading animation while Claude analyzes your data
- Review the comprehensive synthesis report

Step 5: Use Your Results

- Read the formatted analysis with clear sections and scoring
- Copy the results to clipboard for use in documents
- Make informed decisions based on the synthesized insights

📖 Understanding the Output

Individual Response Evaluation

Each AI response receives scores (1-10) across six dimensions:

- **Factual Accuracy**: Correctness and evidence quality
- **Completeness**: Thoroughness and depth of coverage
- **Reasoning Clarity**: Logical flow and structure
- **Creativity/Originality**: Novel insights and approaches
- **Relevance**: Direct applicability to the prompt
- **Evidence Quality**: Supporting data and citations

Comparative Analysis

- **Convergence Points**: Where multiple AIs agree
- **Divergence Points**: Contradictions and conflicting recommendations
- **Unique Contributions**: Insights only provided by specific AIs
- **Gap Analysis**: Important aspects not covered by any response

Synthesis Decision Log

- **Information Source Attribution**: What came from which AI
- **Conflict Resolution**: How contradictions were resolved
- **Enhancement Decisions**: Gaps filled and improvements made
- **Exclusion Rationale**: What was left out and why

🔧 Customization

Modifying the Meta-Prompt

Edit `meta-prompt.txt` to customize:

- **Evaluation criteria** (add new scoring dimensions)

- **Output structure** (change sections and format)
- **Analysis techniques** (different synthesis approaches)
- **Instructions** for specific use cases

The file uses placeholders that are automatically replaced:

- `{{ORIGINAL_PROMPT}}` - Your original question
- `{{RESPONSES}}` - All AI responses with their labels

Changing the Port

If port 3000 is in use, modify `server.js`:

```javascript
const PORT = 3001; // Change to any available port
```

Adding New LLM Options

Update the dropdown options in `public/index.html`:

```html
<option value="NewAI">NewAI</option>
```

🛠️ Troubleshooting

Common Issues

Port 3000 already in use:

```bash
# Find and kill the process
kill -9 $(lsof -ti:3000)
# Or change the port in server.js
```

Meta-prompt file not found:

- Ensure `meta-prompt.txt` exists in the project root
- Check file permissions and spelling
- Server will exit with clear error message if missing

API key issues:

- Verify your key starts with `sk-ant-`
- Check your Anthropic account has sufficient credits
- Use the "Test API Key" button to verify

CORS errors:

- Make sure you're accessing `http://localhost:3000` (not opening HTML directly)
- Restart the Node.js server if needed

Debug Mode

The server provides detailed logging:

- LLM labels received and processed
- Response section preview
- Meta-prompt snippet verification
- API call status and errors

🔒 Security & Privacy

- **API key stored locally** in browser localStorage only
- **No data logging** - server acts as a simple proxy
- **Direct Claude API calls** - your data goes only to Anthropic
- **Local processing** - all synthesis happens on your machine

🚀 Advanced Usage

Batch Analysis

- Save multiple synthesis reports for comparison
- Track AI model performance over time
- Build a knowledge base of AI capabilities

Team Collaboration

- Share meta-prompt configurations across team members
- Standardize evaluation criteria for consistent analysis
- Export results for presentation and decision-making

Research Applications

- Compare AI model capabilities across different domains
- Analyze evolution of AI responses over time
- Identify strengths and weaknesses of different models

📊 Use Cases

Business Strategy

- **Market analysis**: Compare AI insights on market trends
- **Product decisions**: Synthesize recommendations from multiple AI advisors
- **Risk assessment**: Comprehensive evaluation of business risks

Research & Analysis

- **Literature reviews**: Combine AI summaries of research papers
- **Technical decisions**: Evaluate different technological approaches
- **Competitive analysis**: Multi-perspective market intelligence

Creative Projects

- **Content strategy**: Blend creative ideas from different AI models
- **Problem solving**: Comprehensive solution synthesis
- **Innovation**: Identify unique insights across AI perspectives

🎯 Tips for Best Results

Prompt Engineering

- Use **clear, specific questions** for better AI responses
- **Provide context** about your goals and constraints
- **Ask for evidence** and reasoning in your original prompts

Response Selection

- Choose **diverse AI models** for varied perspectives
- Include both **general** (ChatGPT) and **specialized** (Perplexity) models
- Ensure responses **address the same core question**

Meta-Prompt Optimization

- Customize **evaluation criteria** for your domain
- Adjust **scoring weights** based on what matters most
- Add **domain-specific** analysis techniques

📈 Version History

Current Version

- LLM-specific response labeling
- Rich text formatting with markdown parsing
- External meta-prompt configuration
- Comprehensive error handling and validation
- Professional UI with responsive design

Planned Enhancements

- Export to PDF/Word formats
- Batch processing capabilities
- Historical analysis tracking
- Custom AI model integrations
- Advanced visualization options

🤝 Contributing

This tool is designed to be easily extensible:

- **Add new LLM options** by updating dropdown lists
- **Enhance meta-prompt** techniques in the external file
- **Improve UI/UX** by modifying the HTML and CSS
- **Add new features** to the Node.js backend

📄 License

🆘 Support

For issues and questions:

1. Check the troubleshooting section above
2. Verify your project structure matches the documentation
3. Review server console logs for error details
4. Ensure all dependencies are properly installed

Built with ❤️ to harness the collective intelligence of multiple AI systems for better decision-making.