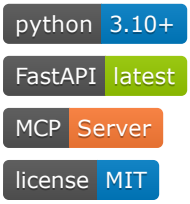
Study Tools MCP



An AI-powered study assistant built with Model Context Protocol (MCP) that generates quizzes, flashcards, summaries, and concept explanations from your study materials.

Features

- **Smart Summarization** - Generate concise summaries from study materials
- **Quiz Generation** - Create customizable quizzes with difficulty levels
- **Concept Explanation** - Get beginner/intermediate/advanced explanations
- **Flashcards** - Auto-generate flashcard decks from documents
- **Comparison Tool** - Compare and contrast multiple concepts
- **MCP Integration** - Works directly with Claude Desktop
- **Web UI** - Standalone chat interface with FastAPI backend

Tech Stack

- **Backend**: FastAPI + Python 3.10
- **MCP**: Model Context Protocol server/client
- **AI**: OpenAI API
- **Document Parsing**: PyPDF2, pdfplumber, python-docx
- **Frontend**: Vanilla JavaScript, HTML, CSS

Quick Start

Prerequisites

- Python 3.10+
- OpenAI API key

Installation

1. Clone the repository:

```
git clone <your-repo-url>
cd 4.study-tools-mcp
```

2. Install dependencies:

```
pip install -e .
```

3. Create `.env` file:

```
cp .env.example .env
# Edit .env and add your OPENAI_API_KEY
```

4. Add your study materials:

Place PDF or Markdown files in `data/notes/`:

```
data/notes/
├── Machine Learning.pdf
└── Your Notes.md
```

5. Run the application:

```
python app.py
```

6. Open browser:

```
http://localhost:8080
```

Docker Deployment

Build and Run

```
docker build -t study-tools-mcp .
docker run -p 8080:8080 --env-file .env -v ./data/notes:/app/data/notes study-tools-mcp
```

Project Structure

```
study-tools-mcp/
├── app.py                      # FastAPI web application
├── src/study_tools_mcp/
│   ├── server.py               # MCP server entry point
│   ├── config.py               # Configuration
```

```
│   ├── tools/                  # Quiz, flashcards, summarizer, explainer
│   ├── parsers/                # PDF and Markdown parsers
│   └── utils/                  # Logger
├── static/                     # Frontend assets
├── templates/                  # HTML templates
├── data/notes/                 # Your study materials
├── logs/                       # Application logs
└── pyproject.toml              # Dependencies
```

💻 Usage

Web UI

The web interface provides an interactive chat where you can ask the AI to:

Tool	Example Prompt
Summarize	`Summarize the topic: neural networks`
Quiz	`Create a 5-question quiz on "decision trees" at intermediate level`
Explain	`Explain the concept "gradient descent" at beginner level`
Compare	`Compare these concepts: SVM KNN`
Flashcards	`Create 10 flashcards for: ensemble methods`

Claude Desktop Integration

Add to `%APPDATA%\Claude\claude_desktop_config.json`:

```
{
  "mcpServers": {
    "study-tools-mcp": {
      "command": "uv",
      "args": ["--directory", "C:\\path\\to\\study-tools-mcp", "run", "study-
tools-mcp"]
    }
  }
}
```

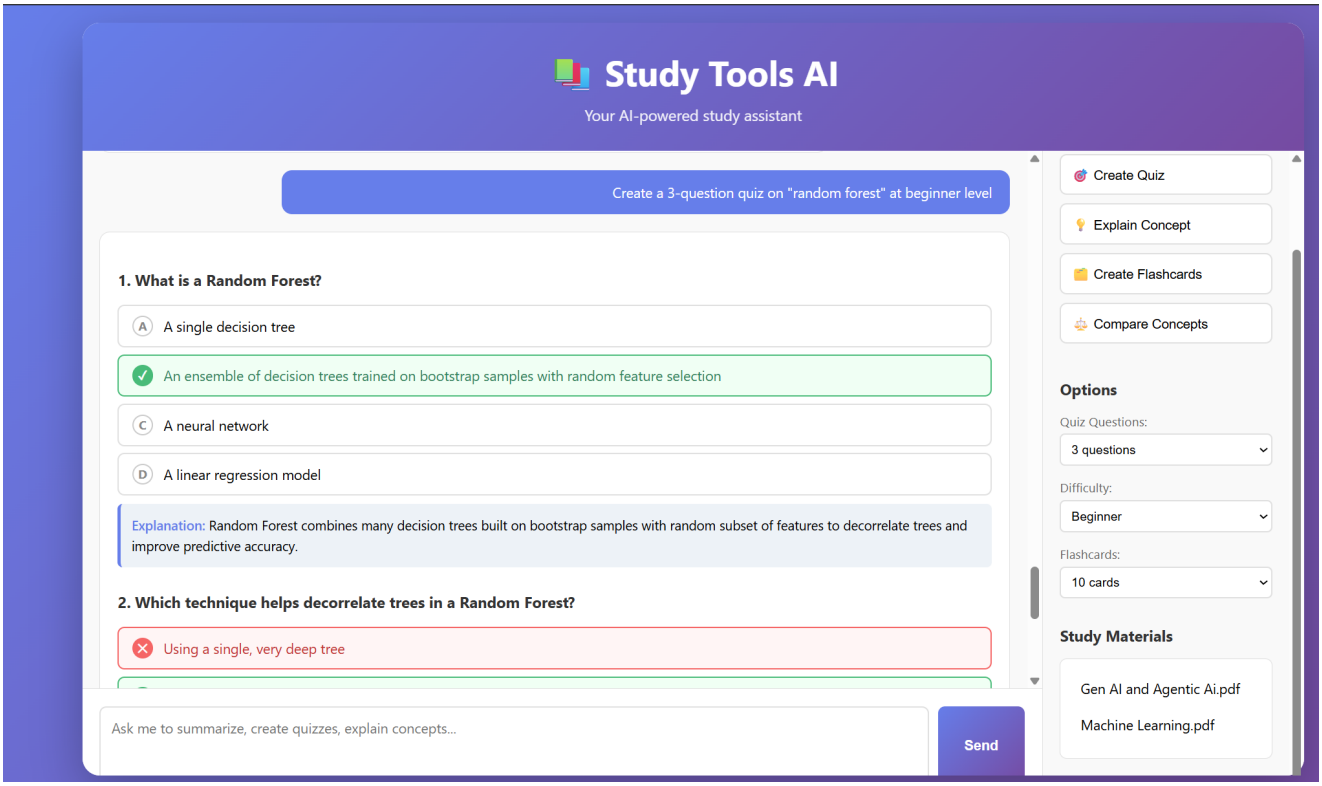
Restart Claude Desktop - the tools will be available automatically.

📡 API Endpoints

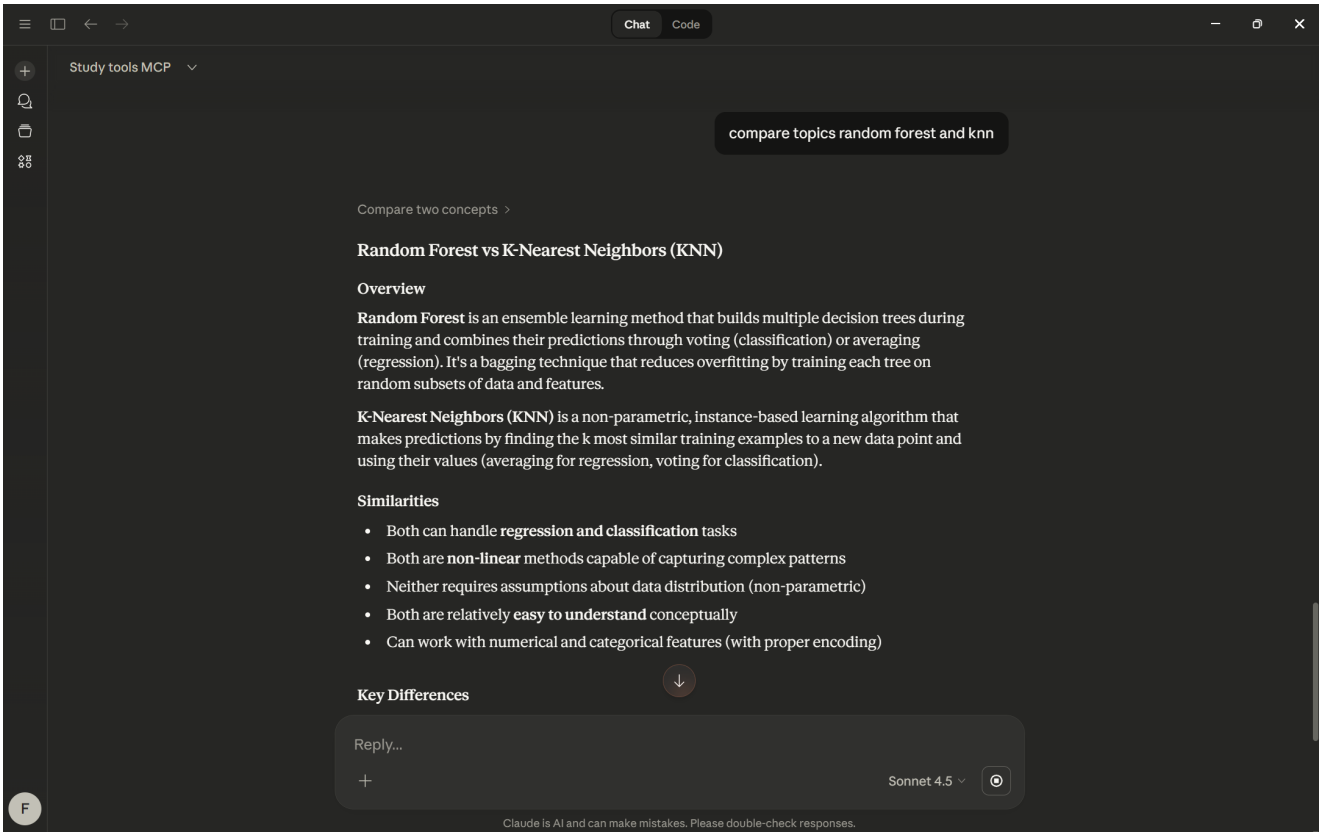
- `GET /` - Web UI
- `GET /health` - Health check
- `GET /api/files` - List available study materials
- `POST /api/chat` - Chat with streaming

- `POST /api/chat/clear` - Clear conversation history

Screenshots



Study Tool AI Interface with quiz generation



Study Tool AI Integration with Claude code desktop

License

MIT License