AXIO AI

AI Personal Assistant

Version 1.0

An intelligent, feature-rich AI-powered personal assistant with advanced chat capabilities, document analysis, data visualization, task management, and productivity tools.

by Perfionix AI

Python 3.8+ | **Flask 3.0.0** | AI **License**

December 2025

Table of Contents

1. Overview

Axio is a comprehensive AI-powered personal assistant developed by **Perfionix AI**. It combines the power of Large Language Models (LLMs) with modern web technologies to provide users with an intelligent, intuitive, and feature-rich productivity platform.

Axio goes beyond simple chatbots by offering specialized modules for document analysis (DocIQ), data visualization (VizIQ), task management, note-taking, and reminders - all wrapped in a beautiful, responsive dark-themed interface.

Key Highlights:

- Unified platform combining multiple productivity tools
- Advanced AI chat with web search integration
- Document Q&A using RAG (Retrieval Augmented Generation)
- Automatic data visualization and insights
- Complete privacy with local processing
- Modern, responsive dark-themed UI

2. Problem Statement

In today's digital world, professionals and individuals face several challenges:

Problem	Impact
Information Overload	Difficulty extracting insights from large documents and datasets
Fragmented Tools	Using multiple apps for chat, tasks, notes, and data analysis
Complex Data Analysis	Non-technical users struggle to visualize and understand data
Document Comprehension	Time-consuming to read and analyze lengthy documents
Productivity Gaps	Lack of unified platform for personal productivity management
Accessibility	Enterprise AI solutions are expensive and complex to deploy

3. Solution Overview

Axio provides a comprehensive solution to modern productivity challenges:

#	Solution	Description	Benefit
1	Unified Platform	Single app combining chat, docs, data, tasks	Eliminates multiple tools
2	AI Chat	LLM-powered conversational AI with web search	Instant answers & code help
3	DocIQ	RAG-based document Q&A system	Extract insights from any document
4	VizIQ	Upload data, get instant dashboards	No coding required for analytics
5	Task Management	Priority-based task tracking	Stay organized & productive
6	Local Processing	All data processed on your machine	Complete data privacy

Key Value Propositions

Value	Description
■ Unified Platform	All productivity tools in one place - no more app switching
■ AI Intelligence	Leveraging LLMs for smart, contextual responses
■ No-Code Analytics	Upload CSV/Excel, get instant KPIs, charts, and insights
■ Document Q&A	Chat with your documents using RAG technology
■ Beautiful UX	Modern, responsive, dark-themed interface
■ Privacy First	All data stays on your machine - complete privacy control
■ Real-time	Instant responses with streaming AI and live updates
■ Web Search	Get up-to-date information from the internet

4. Key Features

4.1 AI Chat Assistant

- Real-time conversational AI powered by LLM
- Syntax highlighting for 10+ programming languages
- Markdown rendering with code blocks
- Web search integration for real-time information
- Message editing and regeneration
- Typing indicators and streaming responses
- Context management (15 messages history)

4.2 DocIQ - Document Intelligence

- Upload and analyze PDF, DOCX, TXT, and MD files
- RAG (Retrieval Augmented Generation) for accurate answers
- Chat with your documents naturally
- Multi-document support
- Context-aware responses
- Advanced text chunking and retrieval

4.3 VizIQ - Data Visualization

- Upload CSV, Excel, JSON data files
- Automatic data preprocessing and cleaning
- AI-generated dashboard names and descriptions
- Auto-generated KPIs (Key Performance Indicators)
- Multiple chart types: Column, Doughnut, Comparison, Line
- AI-powered insights generation
- Interactive data preview table

4.4 Task Management System

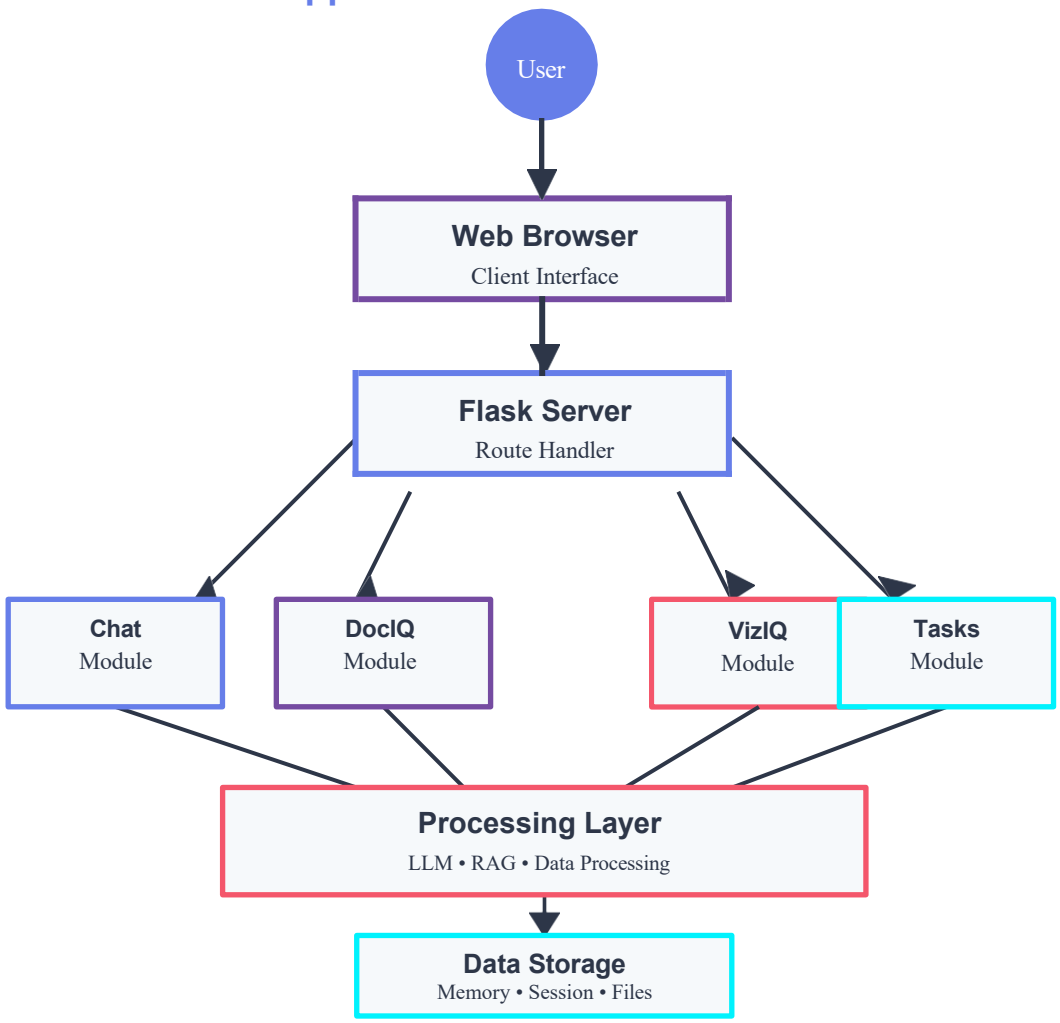
- Create, edit, and delete tasks
- Priority levels (High, Medium, Low)
- Status tracking (Pending, Completed)
- Filter and sort capabilities
- Visual progress indicators

5. System Architecture & Workflows

Axio follows a layered architecture pattern with clear separation of concerns. The system is organized into four distinct layers: Client, Application, Service, and Data layers.

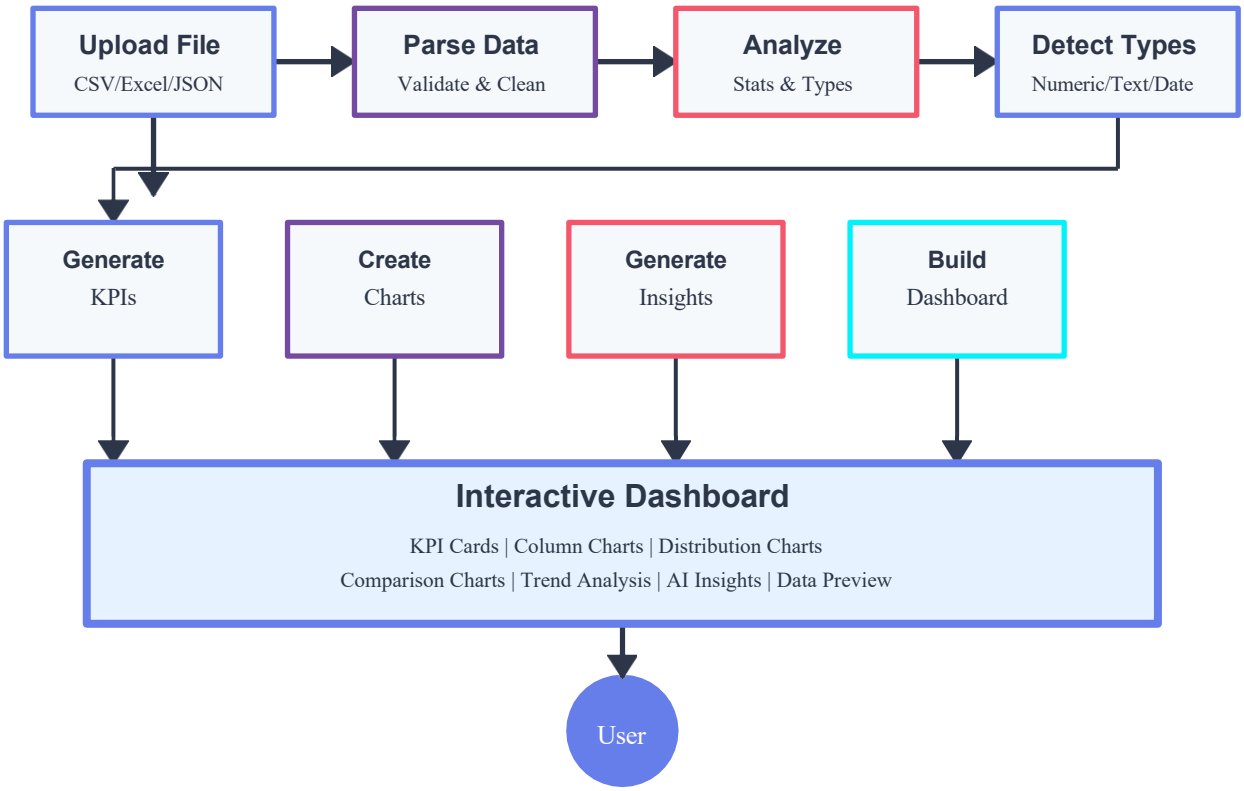
Layer	Components	Responsibility
Client	HTML, CSS, JavaScript, Chart.js	User interface and interaction
Application	Flask Routes (Chat, DocIQ, VizIQ, Tasks)	Request handling and routing
Service	LLM API, Doc Processor, Data Processor, RAG	Business logic and processing
Data	In-Memory, Session Storage, File Storage	Data persistence

Application Architecture Flow



The application follows a request-response cycle: User interacts with the browser interface, requests are routed through Flask to appropriate modules, processed using service layer components, stored/retrieved from the data layer, and results are returned to the user interface.

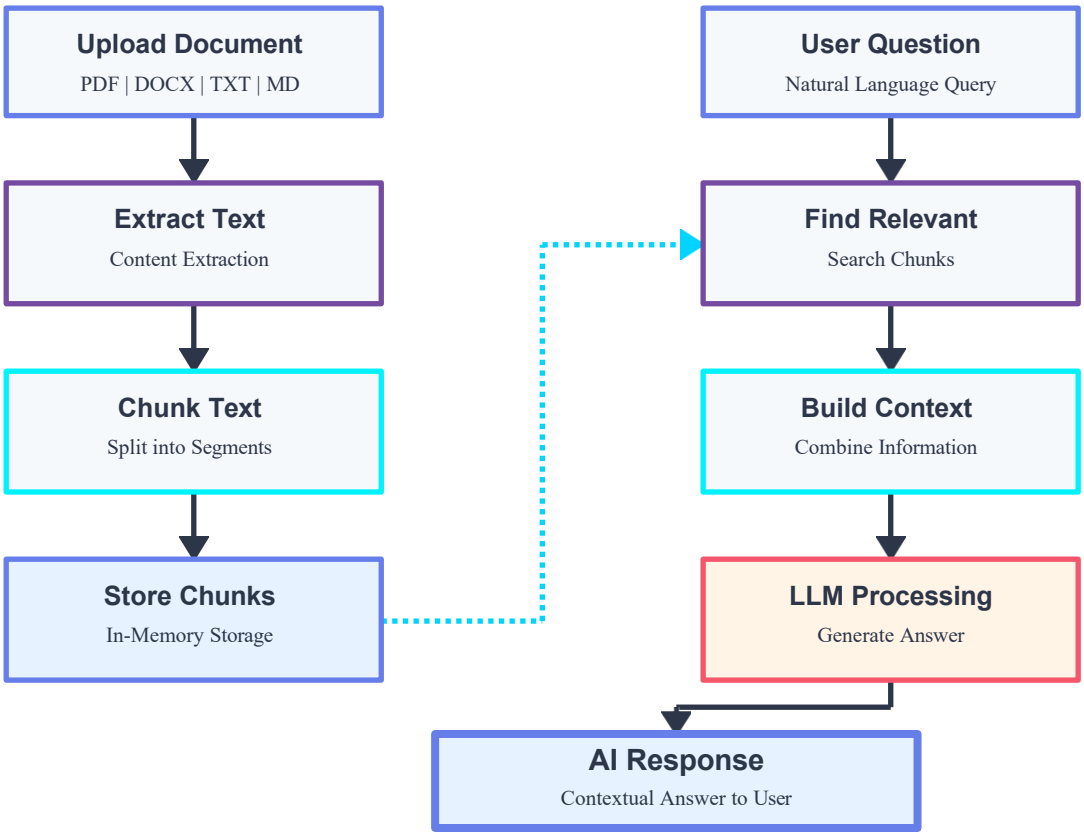
5.1 VizIQ Data Processing Workflow



VizIQ processes data through a multi-stage pipeline:

(1) Upload - User uploads CSV/Excel/JSON file;
(2) Parse - Data is validated and cleaned;
(3) Analyze - Statistical analysis and column type detection;
(4) Generate - KPIs, charts, and insights are automatically created;
(5) Render - Interactive dashboard is displayed to the user with all visualizations and data preview.

5.2 DocIQ RAG Processing Workflow



DocIQ implements RAG (Retrieval Augmented Generation) architecture:

Upload Path - Document is uploaded, text is extracted, split into chunks, and stored.

Query Path - User asks a question, relevant chunks are retrieved from storage, context is built, and LLM generates an accurate, contextual answer based on the document content.

6. Technology Stack

6.1 Backend Technologies

Technology	Purpose	Version
Python	Core programming language	3.8+
Flask	Web framework	3.0.0
Werkzeug	WSGI utilities	3.0.1
Requests	HTTP client	2.31.0
python-dotenv	Environment management	1.0.0

6.2 Document Processing

Technology	Purpose	Version
PyPDF2	PDF parsing	3.0.1
python-docx	DOCX parsing	1.1.0
BeautifulSoup4	HTML/XML parsing	4.12.2

6.3 Data Processing & Frontend

Data Processing: openpyxl (Excel), CSV (built-in), JSON (built-in)
Frontend: HTML5, CSS3 (custom dark theme), JavaScript (ES6+), Chart.js, Highlight.js, Marked.js
External Services: Ollama (local LLM), Google Search (web integration), ElevenLabs (TTS - optional)

7. Installation Guide

7.1 Prerequisites

Before installing Axio, ensure you have: Python 3.8+, pip, Ollama (or compatible LLM API), and a modern web browser.

7.2 Installation Steps

Step 1: Clone Repository

```
git clone https://github.com/perfionix-ai/axio.git cd axio
```

Step 2: Create Virtual Environment

```
python -m venv venv # Windows: venv\Scripts\activate # Linux/Mac: source venv/bin/activate
```

Step 3: Install Dependencies

```
pip install -r requirements.txt
```

Step 4: Configure Environment

```
cp .env.example .env # Edit .env: # GPT_SERVER_URL=http://localhost:11434/api/chat # GPT_MODEL=llama3.2
```

Step 5: Start Ollama

```
ollama serve ollama pull llama3.2
```

Step 6: Run Application

```
python app.py
```

Step 7: Access Axio

Open your browser and navigate to: **http://localhost:5000**

8. Module Details

8.1 AI Chat Module

Key Functions: chat(), format_response(), web search integration, streaming response
Features: Multi-turn conversations with 15-message context, code syntax highlighting, web search for real-time data, message editing and regeneration, markdown rendering

8.2 DocIQ Module

Key Functions: dociq_upload(), dociq_chat(), extract_text_from_pdf(), extract_text_from_docx(), chunk_text(), find_relevant_chunks()
Supported Formats: PDF, DOCX, TXT, MD
RAG Process: Document upload → Text extraction → Chunking → Storage → Query → Retrieval → Context building → LLM generation → Contextual answer

8.3 VizIQ Module

Key Functions: viziq_upload(), parse_csv_data(), parse_excel_data(), detect_column_types(), calculate_statistics(), generate_kpis(), generate_chart_configs(), generate_insights()
Chart Types: Column (categorical data), Distribution (doughnut), Comparison (metrics), Trend (time series)
Pipeline: Input → Parse → Analyze → Detect Types → Generate KPIs → Create Charts → Generate Insights → Render Dashboard

9. API Endpoints Reference

Endpoint	Method	Description
/api/chat	POST	Send chat message and receive AI response
/api/chat/clear	POST	Clear chat history
/api/dociq/upload	POST	Upload document for analysis
/api/dociq/chat	POST	Query uploaded documents
/api/dociq/clear	POST	Clear all documents
/api/viziq/upload	POST	Upload data file for visualization
/api/viziq/clear	POST	Clear uploaded data
/api/tasks	GET	Get all tasks
/api/tasks	POST	Create new task
/api/tasks/<id>	PUT	Update existing task
/api/tasks/<id>	DELETE	Delete task
/api/notes	GET	Get all notes
/api/notes	POST	Create new note
/api/notes/<id>	DELETE	Delete note
/api/reminders	GET	Get all reminders
/api/reminders	POST	Create new reminder
/api/reminders/<id>	DELETE	Delete reminder

10. Future Roadmap

The Axio development team has an exciting roadmap planned for future releases:

Version 2.1 (Q1 2025)

• Database integration (SQLite/PostgreSQL) | User authentication | Export dashboards as PDF/PNG | Enhanced security | Performance optimizations

Version 2.2 (Q2 2025)

• Voice input (speech-to-text) | Multi-language support | Calendar integration | Email integration | Advanced filtering and search

Version 3.0 (Q3-Q4 2025)

• Mobile PWA | Real-time collaboration | Plugin architecture | Custom AI model fine-tuning | Advanced analytics | Team features

11. Contributing Guidelines

We welcome contributions! Whether it's bug fixes, new features, documentation improvements, or suggestions, your input is valuable.

How to Contribute

1. **Fork the Repository** → Fork Axio to your GitHub account
2. **Create Feature Branch** → git checkout -b feature/AmazingFeature
3. **Make Changes** → Implement your feature or bug fix
4. **Commit Changes** → git commit -m 'Add some AmazingFeature'
5. **Push to Branch** → git push origin feature/AmazingFeature
6. **Open Pull Request** → Submit detailed PR with description

Code Style Guidelines

Follow PEP 8 | Use meaningful names | Add comments for complex logic | Write unit tests | Update documentation | Ensure proper formatting

12. Contact & License

Perfionix AI

Website: perfionix.ai | **Email:** support@perfionix.ai | **GitHub:** github.com/perfionix-ai | **Docs:** docs.perfionix.ai

License

Built with ❤■ by Perfionix AI

Empowering productivity through intelligent automation

Version 2.0.0 | Last Updated: December 2025 | Status: Active Development