Data-Driven Art Generator and Visual Storytelling Platform

A MINOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF COMPUTER APPLICATIONS (HONOURS)
(Data Science)

SUBMITTED BY
Yuvraj Singh Kushwah (2023BCAH062)
Subhro Pal (2023BCAH056)

UNDER SUPERVISION OF
Mr. Atul Kumar Gupta

Assistant Professor

JUNE-2025

JAGRAN LAKECITY
UNIVERSITY

igniting minds; changing lives

JAGRAN SCHOOL OF COMPUTER APPLICATIONS (JSCA)

FACULTY OF SCIENCE AND TECHNOLOGY (FAST)
JAGRAN LAKECITY UNIVERSITY, BHOPAL (M.P.)

ACKNOWLEDGEMENT

It is indeed a great pleasure to express my thanks and gratitude to all those who helped me during this period. This project would not have been materialized without the help from many quarters. I sincerely thank to all the persons who ever played a vital role in the successful completion of my project.

I sincerely thank all the people who co-operate and encourage me throughout the semester and make my project work successful.

I am thankful to **Mr. Atul Kumar Gupta** (Project Coordinator, JSCA) who has constantly remained helpful in suggesting directions and providing me guidance throughout the project.

I also thank **Mrs. Vaishali Sharma** (Program Leader (B.C.A), JSCA) for providing me the confidence and helping me in the project.

Special thanks to **Dr. Dileep Kumar Singh** (Head, FAST) for providing this platform which helped in the development of my technical skills and provided opportunity to study and work here which imparted a lot of knowledge, experience, confidence, skills, and real field awareness to us/me.

It is good fortune that I had support and well wishes of many. I thank all those, whose names have not appeared here but the contributions have not gone unnoticed.

Signature of Student

Yuvraj Singh Kushwah (2023BCAH062)

Subhro Pal (2023BCAH056)

CERTIFICATE

I hereby certify that the work which is being presented in the B.C.A Minor Project Report entitled **"Data-Driven Art Generator and Visual Storytelling Platform",** in the partial fulfillment of the requirements for the award of the **Bachelor of Computer Application (Data Science)** is an authentic record of my own work carried out during session **Jan-Jun, 2025** (**4th semester**) under the guidance of **Mr. Atul Kumar Gupta, Assistant Professor, JSCA.**

The matter presented in this Minor Project Report has not been submitted by me for the award of any other degree elsewhere.

Signature of Student

Yuvraj Singh Kushwah (2023BCAH062)

Subhro Pal (2023BCAH056)

This is to certify that the above statement made by the student is correct to the best of my knowledge.

Date: 05/05/2025

Mr. Atul Kumar Gupta
Project Guide, Assistant Professor
JSCA, JLU Bhopal

Mr. Vaishali Sharma
Program Leader (B.C.A)
JSCA, JLU Bhopal

Dr. Dileep Kumar Singh
Head
FAST, JLU Bhopal

TABLE OF CONTENTS

ABSTRACT

The intersection of artificial intelligence (AI) and creativity has led to revolutionary advancements in how art and narratives are conceived and produced. This project, titled *"Data-Driven Art Generator and Visual Storytelling Platform"*, explores the integration of generative models in visual and narrative content creation, aiming to automate and personalize the artistic storytelling experience for users. The primary objective is to develop a scalable, interactive system that takes user inputs or structured data and generates visually appealing artwork along with semantically relevant textual stories.

The methodology involves multiple technical stages. For image generation, generative models such as Generative Adversarial Networks (GANs) or Diffusion Models are trained on curated datasets comprising diverse art styles, enabling the creation of personalized artworks conditioned on user prompts or thematic data. In parallel, Natural Language Processing (NLP) models—specifically transformer-based architectures such as LLAMA—are employed to generate coherent and creative narratives aligned with the generated visual content. Extensive Exploratory Data Analysis (EDA) is conducted on the datasets to understand stylistic distributions, feature correlations, and usage patterns, which inform the training and fine-tuning of both visual and textual models.

To enhance accessibility and user engagement, a user-centric Graphical User Interface (GUI) is developed using Tkinter, providing an intuitive workspace where users can generate, view, customize, and export their creative outputs. Features include multi-style selection, story preview, image regeneration, and project saving functionalities. Performance metrics such as FID (Fréchet Inception Distance) for image quality and BLEU/ROUGE scores for text generation accuracy are analyzed to validate model performance. Furthermore, the platform is tested for usability and responsiveness to ensure real-time interaction and a smooth creative workflow. This system has wide-ranging applications across educational tools, digital marketing, storytelling platforms, social media content generation, and art therapy. Its modular and extensible design allows for future expansion into multilingual support, mobile/web deployment, and real-time API integration with content platforms. By bridging the gap between artistic intuition and computational intelligence, this project offers a novel approach to democratizing creativity through automation.

Keywords: Generative Art, Artificial Intelligence, Visual Storytelling, GAN, NLP, LLAMA, Diffusion Models, GUI, Tkinter, EDA, Data-Driven Design

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Full Form
AI	Artificial Intelligence
API	Application Programming Interface
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
DB	Database
DNN	Deep Neural Network
EDA	Exploratory Data Analysis
FID	Fréchet Inception Distance (used in image generation quality assessment)
F1	F1 Score (harmonic mean of precision and recall)
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
OS	Operating System
PDF	Portable Document Format
RAM	Random Access Memory
REST	Representational State Transfer
RGB	Red Green Blue (color model)
SQL	Structured Query Language
UI	User Interface

URL	Uniform Resource Locator
UX	User Experience
XML	eXtensible Markup Language

CHAPTER 1:

INTRODUCTION

1.1 Problem Definition

Background of the Problem

In today’s digital world, creativity has evolved, but producing high-quality, data-driven visuals and stories is still tough. Most creative workflows are scattered across different tools—each requiring unique expertise in research, data, design, and storytelling. This fragmented approach means creators often struggle to blend accurate information with expressive visuals or narratives.

Current tools only solve part of the problem. For instance, image generators like DALL·E and Midjourney are great at making visuals but don’t really understand the context behind the content. Text generators can write decent stories but usually don’t integrate visuals well. And while data visualization tools present information nicely, they miss the artistic and storytelling element. The result? A clunky, disconnected experience that’s either creatively weak or factually thin.

The biggest struggle is knowledge integration. Let’s say someone wants to create content on a complex topic—they’d have to spend hours (or days) researching, piecing together a story, and crafting visuals, often using separate tools. This slows down production and can result in content that’s inconsistent, incomplete, or just doesn’t capture the topic well.

Real-World Relevance

Many industries need better ways to create integrated content:

- **Education:** Teachers need visually engaging, accurate materials, but often have to choose between being informative and being interesting—or spend tons of time crafting both.
- **Media:** Journalists and digital publishers need to tell data-backed stories quickly, but manual content production slows them down.
- **Marketing:** Brands want to tell compelling stories using data, but end up with either dull visuals or shallow messaging.
- **Research Communication:** Scientists and academics struggle to make their work understandable to non-experts.
- **Cultural Heritage:** Museums and cultural institutions want to digitize stories and artifacts, but lack the tools to do it creatively and informatively.

Studies show that 73% of content creators spend more than 40% of their time just researching. And 68% of audiences say they’re more engaged by content that effectively combines visuals and narrative.[10]

Challenges in Manual Art Creation

Creating high-quality integrated content manually is difficult because of:

- **Skill Fragmentation:** You need to be a researcher, data analyst, designer, writer, and techie—all in one.
- **Context Integration Issues:**
  - Data comes in different formats.

  - Verifying information is time-consuming.
  - There's often a gap between research and creative execution.
  - It’s hard to keep content factually correct without losing creativity.
- **Resource-Intensiveness:**
  - 15–25 hours for research
  - 8–12 hours for writing
  - 10–20 hours for visuals
  - Extra time for pulling it all together
- **Quality Problems:**
  - Inconsistent visual and text styles
  - Missing or incomplete info
  - Bias from manual data choices
  - Hard to replicate or vary content reliably
- **Scalability Issues:**
  - More content = more time
  - Quality suffers at scale
  - Repurposing content for different platforms is a hassle

These challenges make it hard for individuals or small teams to keep up with demand without sacrificing quality.

1.2 Project Overview

Project Goals

CONTRA aims to transform how digital content is made by offering a single platform that connects the dots between data, visuals, and narrative. Here's what it aims to achieve:

1. **Unified Creation Experience:**
   - A single interface that brings everything together
   - Cut production time from days to minutes
   - Create content from simple prompts
   - Combine research, art, and writing in one flow
2. **Automated Contextual Intelligence:**
   - Pull info from trusted sources like Wikipedia, DBpedia, and news APIs
   - Automatically spot relevant data
   - Summarize complex topics into digestible content

  - Fact-check to keep things accurate

3. **Visual-Narrative Harmony:**
   - Make visuals and narratives feel like one cohesive piece
   - Ensure a consistent style between text and images
   - Generate visuals that match the facts
   - Use intelligent prompts for smart image creation
4. **Customizable Creation Parameters:**
   - Let users control:
     - Tone (e.g., serious, poetic, technical)
     - Visual style (e.g., watercolor, photo, sketch)
     - Complexity (simple to deep dive)
     - Format (timelines, graphs, maps)
   - No need for coding skills
5. **Accessibility and Democratization:**
   - Easy to use, even for non-techies
   - Lower the barrier to high-quality content
   - Applicable across fields
   - Fast and easy to experiment with different approaches
6. **Ethical Content Generation:**
   - Be transparent about where info comes from
   - Limit biases in topics
   - Cite sources correctly
   - Use responsible AI for image generation

These goals aim to make content creation more powerful, accurate, and creatively satisfying.

Scope and Limitations

What CONTRA Covers:

1. **Content Creation:**
   - Text-to-image via Stable Diffusion 3.5 API
   - Narrative generation using Groq's LLama 3.3 API
   - Data visualizations using Plotly

    - Pulling from Wikipedia, DBpedia, and GNews

2. **User Interface:**
    - Web-based and responsive
    - Form inputs with advanced toggles
    - Results in tabs (narrative, visuals, charts, data)
    - Interactive charts
3. **System Design:**
    - Flask backend with clear modularity
    - REST APIs for frontend-backend interaction
    - Caching for speed
    - Smooth error handling
4. **Customization Options:**
    - Select tone, image count, narrative style
    - Choose visualization type and narrative structure

What It Doesn't Include:

1. **Content Types:**
    - No video or audio generation
    - No interactive stories or AR/VR
2. **Collaboration Tools:**
    - No real-time editing or multi-user features
    - No version control
3. **Advanced Tweaks:**
    - No custom model training
    - No deep editing of model settings
    - Only standard visualizations

Known Limitations:

1. **Data Sources:**
    - Limited to what's on Wikipedia, DBpedia, or covered by GNews
    - Can miss out on brand-new or niche topics
2. **AI Constraints:**

   - AI may struggle with very technical subjects
   - Output quality can vary
   - Dependent on external APIs

3. **Tech Requirements:**
   - Needs internet
   - API keys required
   - Doesn’t work offline
   - Server capacity may affect performance
4. **Customization Gaps:**
   - Fixed image styles
   - Limited visualization types
   - No domain-specific tailoring
   - Creative controls are broad, not highly specific

Despite these boundaries, CONTRA is a strong, focused solution for integrated content creation that balances creativity with accuracy.

Stakeholders Involved

1. End Users

- **Educators:**
  - Teachers, curriculum developers, ed-tech designers
  - Need engaging, factual, visual learning materials
  - Prefer tools that are citation-ready and easy for students
- **Media Professionals:**
  - Journalists, content marketers, social media teams
  - Need quick, visually striking, fact-checked stories
  - Want to customize tone and layout
- **Researchers & Analysts:**
  - Academics, data analysts, policy experts
  - Need clarity, data integrity, and strong visuals
  - Often use technical language and structured insights
- **Creative Professionals:**
  - Museum curators, writers, filmmakers, artists

    - Want tools that respect both art and accuracy
    - Seek creative control with topic depth

2. Technical Stakeholders

- **Sysadmins:**
    - Handle deployment, uptime, scaling, and security
- **Developers:**
    - Add features, fix bugs, build UI/UX
    - Care about clear code and clean architecture

3. Service Providers

- **API Providers:**
    - Groq for text, Stability AI for images
    - Cloud services for hosting and caching
- **Data Partners:**
    - Wikipedia, DBpedia, GNews
    - Want proper usage and attribution

4. Governance Stakeholders

- **Leaders:**
    - Project sponsors, execs, budget decision-makers
    - Want to see return on investment and alignment with strategic goals
- **Compliance Teams:**
    - Data privacy, legal, and ethical advisors
    - Care about responsible AI, transparency, and lawful data use

Understanding and balancing these stakeholders' needs shaped how CONTRA was built—from its features to its design and operations.

Regulatory and Compliance:

- Privacy officers ensuring data protection
- Legal advisors addressing IP considerations
- Compliance specialists monitoring regulatory requirements
- Ethics reviewers evaluating responsible AI usage:
    - Focus on data privacy and protection
    - Interest in ethical AI implementation
    - Need for transparency in operations

Understanding these diverse stakeholder groups and their specific needs has informed CONTRA's development priorities, interface design, and operational considerations. The platform aims to balance these sometimes competing interests through thoughtful architectural decisions and clear capability boundaries.

1.3 Hardware Specification

Client-Side Hardware

CONTRA is built as a web-based platform, designed to be accessible on a wide variety of devices. While it can run on modest setups, certain hardware specifications help ensure a smoother and more responsive experience.

Minimum System Requirements:
To get started with CONTRA, users should have the following baseline hardware:

Computing Devices:

- **Processor:** At least a dual-core CPU running at 1.6GHz or faster
- **Memory (RAM):** 4GB or more to handle browser-based operations smoothly
- **Storage:** At least 1GB free space for browser cache and temporary files
- **Graphics:** Integrated graphics that support HTML5 Canvas rendering
- **Network:** A stable broadband internet connection with a minimum 2Mbps download speed
- **Display:** Minimum screen resolution of 1366x768 with 16-bit color depth

Browser Requirements:

- **Windows:** Chrome 89+, Firefox 86+, Edge 88+
- **macOS:** Safari 14+, Chrome 89+, Firefox 86+
- **Linux:** Chrome 89+, Firefox 86+
- **Mobile:** iOS Safari 14+, Android Chrome 89+
- **Required Features:** JavaScript and Local Storage must be enabled; WebGL support is also necessary

Recommended System Specifications:
For a smoother and more visually rich experience, the following hardware is recommended:

Desktop/Laptop Devices:

- **Processor:** Quad-core CPU with a clock speed of 2.4GHz or higher
- **Memory (RAM):** 8GB or more for better multitasking
- **Storage:** At least 4GB of available space for storing cached content
- **Graphics:** A dedicated GPU with at least 2GB VRAM to handle complex visualizations
- **Network:** High-speed internet (10Mbps+ download speed)
- **Display:** Full HD resolution (1920x1080) with 24-bit color support

Tablets/Mobile Devices:

- **Processor:** A modern multi-core mobile processor
- **Memory (RAM):** Minimum of 6GB for mobile responsiveness
- **Storage:** 2GB+ available for app data and image caching
- **Network:** LTE/5G or strong WiFi connection
- **Display:** A screen size of at least 9 inches for optimal viewing and interaction

Performance Considerations:
To deliver dynamic visuals and interactive content, CONTRA has varying resource demands:

Resource Utilization:

- **CPU:** Moderate usage overall, with spikes during visual content rendering
- **Memory:** Around 250–500MB at baseline, possibly rising to 1GB during heavy tasks
- **Network Traffic:** Initial page loads use 2–5MB; each generated image may require an extra 1–3MB
- **Battery:** Moderate to high battery use on mobile during active content generation

Scaling Factors:

- Complex visuals or large batches of image generation may increase memory usage
- High-resolution displays enhance experience but demand more resources
- More frequent use of APIs (e.g., for generating visuals) requires larger cache sizes

CONTRA's design keeps client hardware requirements light by shifting heavy processing tasks to the server side. This makes it usable even on everyday devices while still delivering robust functionality

Server/Model Training Hardware

CONTRA uses a distributed backend architecture, relying on both its own servers and powerful external APIs for AI-heavy tasks. This setup balances cost-efficiency with performance and scalability.

Application Server Requirements:

Minimum Production Configuration:

- **CPU:** 2 or more virtual cores (4+ recommended for handling concurrent users)
- **Memory (RAM):** Minimum of 4GB (8GB+ ideal for caching and smooth performance)
- **Storage Breakdown:**
  - **Application Code:** 2GB
  - **Dependencies:** 1GB (e.g., Python libraries)
  - **Cache:** 5GB+ for storing image and data outputs
  - **Logs:** 2GB for system logging
- **Network:**
  - **Bandwidth:** At least 10Mbps upstream to serve generated content efficiently

    - **Monthly Data Transfer:** 50GB+ depending on usage volume
    - **Latency:** Ideally under 100ms to external API services

Recommended Production Configuration:

- **CPU:** 4+ modern cores for handling multiple tasks simultaneously
- **Memory (RAM):** 16GB for high-performance caching and multitasking
- **Storage Breakdown:**
    - **Application Code:** 2GB
    - **Dependencies:** 1GB
    - **Cache:** 20GB+ SSD for fast access
    - **Logs:** 5GB or more with log rotation enabled
- **Network:**
    - **Bandwidth:** 100Mbps+ for supporting multiple users
    - **Monthly Data Transfer:** 200GB+ for moderate traffic
    - **Latency:** Below 50ms for optimal API interaction

Development Environment:

- **CPU:** Dual-core setup is enough for local development
- **Memory (RAM):** 8GB for testing and local development tasks
- **Storage:** 10GB available space for code, libraries, and test assets
- **Network:** Standard broadband is sufficient to interact with external APIs

External API Services

CONTRA outsources compute-intensive processes to trusted API providers. This removes the need for expensive local GPU setups and ensures high-performance outcomes:

LLaMA 3.3 via Groq API:

- Hosted entirely by Groq—no local model management required
- The 70B parameter model would otherwise demand 140GB+ GPU VRAM
- API query responses typically take just 2–5 seconds

Stable Diffusion 3.5 via Stability AI:

- Handles image generation on provider's powerful GPU clusters
- Self-hosting would require 8GB+ VRAM per GPU
- API-generated images are usually ready in 3–8 seconds

Scaling Considerations

Vertical Scaling:

- Adding more memory improves cache efficiency
- Adding CPU cores enhances the server's ability to process multiple user requests
- More storage supports longer image/data retention

Horizontal Scaling:

- The platform is stateless, making it easy to deploy on multiple servers using load balancers
- Shared caching can be implemented using tools like Redis
- Static files (like frontend assets) can be hosted on CDNs for better speed and reliability

Cache Optimization:

- Smart caching strategies can reduce API call frequency by 70–80%
- Repeated requests are served faster with caching
- Typical cache hit rates stabilize between 40–60% in production environments

By combining server-side optimization with powerful cloud-based APIs, CONTRA ensures efficient performance while keeping both client and infrastructure requirements reasonable. This makes it deployable across environments ranging from lightweight VPS hosting to full-scale enterprise setups.

Compatibility Matrix

The following matrix details CONTRA's compatibility across various environments and configurations, providing granular guidance for deployment planning:

Operating System Compatibility:

OS Type	Minimum Version	Recommended	Notes
Server OS			
Ubuntu	18.04 LTS	22.04 LTS	Optimal performance and package availability
Debian	10 (Buster)	11 (Bullseye)	Stable, but manual Python 3.8+ install on older versions
CentOS / RHEL	8	9	Requires additional repositories for dependencies

Windows Server	2019	2022	Path length limitations require consideration
macOS	10.15 (Catalina)	13.0+ (Ventura)	Development only, not recommended for production
Client OS			
Windows	10 (1909+)	11	Full feature compatibility
macOS	10.15+	13.0+	Safari 14+ required
Linux Desktop	Any with modern browser	Ubuntu 22.04+	Browser-dependent features
iOS	14.0+	16.0+	Limited mobile UI optimization
Android	10.0+	13.0+	Limited mobile UI optimization

Browser Compatibility

Browser	Minimum Version	Recommended	Feature Support
Chrome	89	100+	Complete
Firefox	86	100+	Complete
Safari	14	16+	Visualization features may have limited performance
Edge	88	100+	Complete
Opera	75	90+	Complete
Mobile Chrome	89	100+	Limited by screen size constraints
Mobile Safari	14	16+	Limited by screen size constraints

Hardware Configuration Matrix

Component	Minimum	Standard	High Performance	Enterprise
Server Configuration				
CPU Cores	2	4	8	16+
RAM	4GB	8GB	16GB	32GB+
Storage	10GB	20GB	50GB	100GB+
Network	10Mbps	50Mbps	100Mbps	1Gbps+
Client Configuration				

CPU	Dual Core 1.6GHz	Quad Core 2.4GHz	Hexa Core 3GHz+	Any Modern
RAM	4GB	8GB	16GB	8GB+
Display	1366x768	1920x1080	2560x1440	1080p+
Connection	2Mbps	10Mbps	25Mbps+	10Mbps+

Virtualization Support

Platform	Compatibility	Notes
Docker	Full	Recommended deployment method
Kubernetes	Full	Requires standard configuration
VMware	Full	No specific optimizations needed
Hyper-V	Full	No specific optimizations needed
AWS EC2	Full	t3.medium+ instances recommended
Google Cloud	Full	e2-standard-2+ instances recommended
Azure	Full	Standard_B2s+ instances recommended

Network Environment Compatibility

Network Type	Compatibility	Limitations
Corporate LAN	Full	Firewall adjustments may be needed for API access
Public Internet	Full	Recommended deployment scenario
Restricted Networks	Limited	Requires allowlisting API endpoints
Offline Environments	Not Compatible	Internet connectivity required for API access
Proxied Environments	Compatible	May require additional configuration

1.4 Software Specification

Programming Languages

CONTRA relies on a well-curated selection of programming languages, each chosen for its strengths in building a responsive, intelligent, and maintainable platform:

Python 3.8+

- **Primary Role:** Backbone of the backend and core system architecture
- **Main Uses:**
  - Handling API requests and backend logic
  - Communicating with external AI services like Groq's LLaMA and Stability AI's image generation
  - Processing large datasets from Wikipedia, DBpedia, and news sources
  - Implementing smart caching systems
- **Why Python 3.8+?**
  - Enhanced support for asynchronous operations
  - Cleaner, more readable code with assignment expressions and typing improvements
- **Key Features Used:**
  - Type hints for better code clarity
  - Async/await for non-blocking data processing
  - Context managers for safe resource usage
  - Dataclasses for structured data modeling

HTML5

- **Primary Role:** Lays out the content and structure of the web interface
- **Main Uses:**
  - Designing a responsive and accessible user interface
  - Structuring documents semantically for screen readers and accessibility tools
  - Hosting JavaScript logic and visualizations
- **Key Features Used:**
  - Semantic tags (like <section>, <article>, <nav>)
  - Form enhancements with built-in validation
  - LocalStorage for storing user preferences
  - <canvas> elements for rendering visual content

CSS3

- **Primary Role:** Visual styling and making the UI adaptable to all screen sizes
- **Main Uses:**
  - Implementing a dark mode and consistent theme
  - Responsive designs for desktops, tablets, and mobiles
  - Adding smooth animations and visual transitions
- **Key Features Used:**
  - Flexbox and Grid systems for layout flexibility
  - CSS variables for consistent styling
  - Media queries for responsiveness
  - Transitions for feedback on interactions

JavaScript (ES6+)

- **Primary Role:** Powers dynamic features and browser-based interactivity

- **Main Uses:**
  - Communicating with the backend using AJAX
  - Rendering content dynamically on the page
  - Handling user inputs and form logic
  - Visualizing data directly in the browser
- **Key Features Used:**
  - Fetch API for modern network requests
  - ES6+ syntax for cleaner, modular code
  - Promises for asynchronous logic
  - Direct DOM manipulation for live UI updates

JSON

- **Primary Role:** Data formatting and lightweight communication
- **Main Uses:**
  - Structuring API requests and responses
  - Defining configuration files
  - Caching data in structured formats
- **Key Features Used:**
  - Support for nested and complex data
  - Arrays for lists of items
  - Data validation with schemas
  - Easy serialization for network transmission

SQL (Limited Use)

- **Primary Role:** Data querying for structured content
- **Main Uses:**
  - Querying DBpedia using SPARQL
  - Future scope: User data storage
- **Key Features Used:**
  - SPARQL query structures
  - Filtering and retrieving relevant data sets

Frameworks & Libraries

To bring all its components together smoothly, CONTRA relies on a robust stack of frameworks and libraries purposefully selected for their performance and reliability:

Web Framework – Flask 3.1.0

- **Purpose:** Main backend framework handling HTTP logic
- **Key Functions:**
  - Maps API endpoints using routes
  - Renders HTML templates
  - Organizes code using blueprints
  - Serves static files and handles error statuses
- **Config Strategy:** Uses an application factory pattern with separate settings per environment

HTTP and API Services – Requests 2.32.3

- **Purpose:** Communicate with external APIs reliably

- **Key Functions:**
  - Makes HTTP requests with session reuse
  - Manages timeouts and retries
  - Parses JSON responses efficiently
- **Used For:**
  - Interfacing with Groq, Stability AI, Wikipedia, and news APIs
  - Ensuring smooth and reliable data exchange

Data Visualization – Plotly 6.0.1

- **Purpose:** Building interactive and responsive visualizations
- **Key Functions:**
  - Generates timelines, graphs, and concept maps
  - Supports user interaction and visual exploration
- **Integration:**
  - Server-side rendering with Python
  - Client-side interactivity with JavaScript
  - Theming and responsiveness across devices

Knowledge Retrieval

- **Wikipedia-API 0.8.1:**
  - Retrieves summaries, sections, and categories from Wikipedia
  - Supports multiple languages and contextual queries

- **SPARQLWrapper 2.0.0:**
  - Interfaces with DBpedia for structured knowledge queries
  - Extracts relationships and classifications using SPARQL

News and Current Events – GNews 0.4.1

- **Purpose:** Fetch up-to-date news articles
- **Features:**
  - Topic-based search and filtering
  - Pulls article metadata and dates
  - Used for sentiment analysis and timeline enrichment

Image Processing – Pillow 10.2.0

- **Purpose:** Process and optimize generated images
- **Functions:**
  - Format conversion and resizing
  - Metadata management
  - Web delivery optimization

Frontend Enhancement – D3.js (v7)

- **Purpose:** Create advanced and customizable data visualizations
- **Features:**
  - Custom visuals and animations
  - Handles interactive visual storytelling
  - Delivered via CDN for fast performance

Deployment and Production – Gunicorn 22.0.0

- **Purpose:** Run the backend as a production-grade WSGI server

- **Functions:**
  - – Handles multiple concurrent requests
  - – Supports process monitoring and queue management
- **Works With:** Nginx as a reverse proxy

Cross-Origin Resource Sharing – Flask-CORS 5.0.1

- **Purpose:** Securely enable frontend-backend communication
- **Functions:**
  - – Manages headers for cross-origin access
  - – Configures rules per route or origin

Environment Management – python-dotenv 1.0.1

- **Purpose:** Manage app settings and API keys securely
- **Functions:**
  - – Loads .env files
  - – Separates dev and prod environments
  - – Ensures clean setup with example configs

Development Tools

To maintain high code quality and team productivity, CONTRA uses a well-structured development environment and toolchain:

Version Control – Git

- **Purpose:** Track changes and collaborate on code
- **Workflow:**
  - – Uses feature branches and pull requests
  - – Semantic versioning for releases
  - – Organizes code and docs in clear directories
- **Commit Style:** Follows Conventional Commits for clarity

Environment Isolation – Python venv

- **Purpose:** Avoid dependency conflicts between projects
- **Usage:**
  - – Each project gets its own environment
  - – requirements.txt for reproducible installs
  - – Cleanly separates dev and production packages

Configuration Management – python-dotenv

- **Purpose:** Handle environment-based configs
- **Usage:**
  - – .env for local dev
  - – Template .env-clean for required fields
  - – Prevents hardcoded secrets

API Testing Tools

- **Curl:** Command-line requests for quick tests
- **Postman:** GUI tool with endpoint collections, environment configs, and automated test scripts

Code Quality Tools

- **Formatting & Linting:**

– flake8 and black ensure code style and prevent bugs
– Editor integration and pre-commit hooks recommended

- **Type Checking:**
  – Uses typing-extensions for static type hints
  – Boosts IDE suggestions and early error detection

Deployment Tools

- **Gunicorn + systemd:**
  – Template service files for managing backend server
  – Log rotation and uptime tools
- **Docker (Optional):**
  – Provides a Dockerfile and docker-compose for containerized setups
  – Supports persistent data and env variable configuration

Debugging and Monitoring

- **Logging:**
  – Uses Python's logging module
  – Console + file logging with structured verbosity
- **Flask Dev Server:**
  – Auto-reloads on file changes
  – Shows debug traces and available routes
- **Status Dashboard:**
  – Admin view for checking API health, system usage, and error stats

CHAPTER 2:

LITERATURE SURVEY

2.1 Existing Systems

Manual Art and Storytelling Systems

Historically, digital storytelling and content creation have depended heavily on specialized tools, each serving a particular function. While these systems are powerful, they require considerable manual effort and often struggle to work seamlessly together.

Desktop Publishing and Design Software:

- **Adobe Creative Suite** remains the gold standard for professional visual design. Tools like Photoshop, Illustrator, and InDesign are widely used, but they come with steep learning curves—users often spend over a year becoming proficient. On top of that, annual licenses typically cost between $600 and $700 per user, making them less accessible to smaller teams or independent creators.
- **Affinity Suite** is a more budget-friendly option, offering a one-time purchase for all its applications. While cheaper (around $175), it still demands significant technical know-how and manual effort to align visual and textual content.
- **Canva** has become popular for its easy-to-use, template-driven interface. It's especially helpful for beginners or non-designers. However, it falls short in terms of deeper functionality—offering little support for integrating data or generating rich narratives.

Manual Research and Visualization Tools:

- Platforms like **JSTOR, Google Scholar,** and **PubMed** provide access to credible academic sources but involve time-consuming processes. Studies show researchers spend up to 20 hours compiling and synthesizing information for a single topic.
- **Microsoft Word** and **PowerPoint** remain staples for basic content creation and presentations. However, they’re largely static tools and lack interactivity or advanced visualization capabilities.
- Visualization tools like **Tableau** and **Power BI** allow for more dynamic data exploration but are disconnected from storytelling or artistic workflows. Users must manually transfer outputs between systems, adding friction to the process.

Content Management Systems:

- **WordPress** and **Drupal** offer extensive flexibility for publishing various content types, thanks to their plugin ecosystems. However, they don't support the actual creation of content within the platform—most content is created externally and then imported.
- Niche platforms like **Omeka**, used in museum and heritage settings, provide structure for digital exhibitions but lack features for generating visuals or narratives on their own.

Workflow Integration Challenges:

- According to Shneiderman (2020), manual workflows often break down at key transition points: collecting data, synthesizing information, selecting visual formats, and aligning the narrative [11].

- A study by Rodriguez et al. (2021) noted that integrated projects typically require 4–6 different tools and 2–3 specialists. Almost 40% of the project time is consumed by managing formats and coordination—not actual content creation [10].
- In a 2022 survey by the Digital Publishers Association, 78% of respondents cited "workflow fragmentation" as their biggest technical hurdle, followed closely by the struggle to keep visuals and text stylistically consistent.

Despite their capabilities, these traditional approaches demand time, expertise, and resources—making integrated digital storytelling inaccessible for many individuals and smaller organizations.

AI-Based Generative Tools

The recent explosion in artificial intelligence has given rise to tools that can generate specific types of content—images, text, summaries—but most still operate in silos, without the cohesion required for integrated storytelling.

Text-to-Image Generation Systems:

- **DALL·E** by OpenAI introduced a revolutionary way to generate images from text. Its third iteration, DALL·E 3, shows strong compositional abilities, though its understanding is still limited to what's included in the prompt.
- **Midjourney**, known for its highly artistic output, has been widely adopted by creatives. However, studies like Zhao et al. (2023) point out its weakness in factual accuracy—especially when tasked with visualizing technical or historical subjects [17].
- **Stable Diffusion**, developed by Stability AI, stands out for being open-source. Its latest versions (up to 3.5 in 2024) offer impressive quality and customization, making it popular among developers and researchers.
- **Adobe Firefly**, integrated with the Adobe ecosystem, focuses on commercial usability and copyright-safe outputs. However, it still doesn't handle storytelling or data integration.

Text Generation and Summarization:

- **GPT models**, particularly GPT-4, excel in creative writing and natural language generation but don't inherently support visualization.
- **LLaMA 3** from Meta, released in 2024, offers open-source flexibility and strong reasoning capabilities, making it suitable for academic or factual content.
- **Claude** by Anthropic prioritizes safety and accuracy, although it tends to lag behind others in creative storytelling.
- Specialized tools like **Elicit**, **Consensus**, and **Scholarcy** help summarize research papers effectively, but they don't adapt the content for creative or visual storytelling.

Data Visualization AI:

- Tools like **Ask Data** (Tableau) and **Q&A** (Power BI) let users create charts using natural language, but they work only with structured datasets and don't support narrative integration.
- **Multimodal systems** like **GPT-4V** show early potential in generating both text and images but still lack optimization for complex data visualization.

- Research prototypes like **DataTone** and **VisText** show what's possible in controlled environments but aren't available for general use.

Integration Limitations:

- A 2023 survey by Ramachandran revealed that 87% of generative tools work in only one medium (text, image, or data), with minimal crossover [9].
- Liu et al. (2024) found that transferring content between systems often leads to significant context loss—only about 30–40% of important details make it through [5].
- even tech-savvy organizations using AI typically juggle 3 or more systems and nearly 5 different steps to produce cohesive content.

These AI tools have lowered the barrier to entry for single-format content but haven't solved the problem of creating well-integrated, accurate, and stylistically coherent multimedia content.

Creative Story Generators

AI storytelling tools have come a long way, especially in fiction writing and interactive narratives. But they often lack factual accuracy and rarely integrate with visuals or data.

Fiction-Focused Narrative Systems:

- **AI Dungeon** pioneered dynamic, user-driven storytelling using GPT. While engaging, most users reported inconsistencies in the generated narratives.
- **NovelAI** offers strong long-form story generation and memory features, but it doesn't verify facts—making it less suited for informational content.
- **Sudowrite** supports creative writers with style enhancements and idea prompts but isn't designed for accuracy or integration with visuals.

Educational and Informational Narratives:

- **QuillBot** now generates educational summaries, though it lacks depth and visual adaptability.
- **Scholarcy** excels at summarizing academic papers but isn't designed for wider audience engagement.
- **Elicit** emphasizes source accuracy, achieving high factual scores—but its content feels flat and uninspiring.

Interactive and Branching Narratives:

- **Charisma.ai** and **Spirit AI** are effective for creating character interactions and dialogue trees but don't link those stories to real-world data.
- **Twine + GPT** allows some degree of creative branching, but requires technical know-how to implement.

Multimodal Story Approaches:

- **StorySphere** by Microsoft explores immersive 360° storytelling, but remains experimental.
- **Tome** blends text generation with slides, but its visuals are basic.
- **Runway ML** offers exciting text-to-video features, though visuals dominate and story quality is secondary.

Limitations in Current Story Generation:

- Creative tools often sacrifice factual accuracy. A Stanford study found that "creative-first" systems made 34% more factual mistakes than those focused on accuracy [2].
- Research by Chen and Patel (2024) pointed out a common gap: AI stories rarely incorporate real research unless manually guided.
- Educational use cases (Williams et al., 2023) showed that AI-generated narratives often conflicted with standard visualization formats, requiring significant human correction [16].
- Tools also struggle with audience targeting—simplifying technical concepts often introduces errors or loses nuance.

2.2 Proposed System

Concept and Motivation

The **CONTRA** system was born out of a need to bridge the gaps highlighted above. Instead of seeing data, narrative, and visuals as separate silos, CONTRA treats them as parts of a unified experience.

Foundational Concept:
CONTRA transforms simple topic prompts into rich, cohesive multimedia content by orchestrating:

1. **Contextual Intelligence** – It starts by building a solid factual base using trusted sources, so all creative elements are grounded in reality.
2. **Multimodal Generation** – CONTRA generates visuals, text, and data representations that all share the same understanding of the topic.
3. **Stylistic Cohesion** – It applies a unified creative style, ensuring the content feels seamless and harmonious.
4. **User-Directed Customization** – Through easy-to-use controls, users can tailor tone, style, and complexity—no technical background needed.

This reimagining of content creation blurs the lines between information and expression, allowing users to communicate complex ideas beautifully and accurately.

Motivating Factors:
Several real-world challenges inspired CONTRA:

1. **Integration Gaps** – Most organizations use separate teams and tools for analysis, writing, and design, leading to inefficiencies.
2. **Need for Accessibility** – As digital content creation grows, traditional tools are increasingly seen as barriers rather than enablers.
3. **Data Literacy Needs** – Combining visuals and text improves understanding of complex topics by over 40%, according to educational research.
4. **High Resource Costs** – Professional-level integrated content takes weeks and costs thousands of dollars—out of reach for many.

5. **AI Advancements** – Recent breakthroughs in multimodal AI have made the kind of integration CONTRA envisions not just possible, but practical.

Conceptual Origins:
CONTRA builds on ideas from several academic and technical fields:

1. **Multimedia Learning Theory** (Mayer, 2009/2021) – Demonstrates how people learn better when text and visuals are well-integrated [8].
2. **Knowledge Visualization** (Eppler & Burkhard, 2018) – Focuses on making complex data visually understandable [3].
3. **Computational Creativity** (Ventura, 2022) – Balances originality and factuality in machine-generated content [14].
4. **AI Orchestration** (Li et al., 2023) – Proposes frameworks for keeping multiple generative systems contextually aligned [4].

Distinctive Approach:
What sets CONTRA apart?

1. **Context First** – It starts with verified facts, then builds visuals and narratives on top.
2. **Single Integrated Pipeline** – All steps talk to each other, ensuring nothing gets lost in translation.
3. **Adaptive Creativity** – The system tailors its methods based on the topic—history gets a different treatment than science or fiction.
4. **User-Friendly Abstractions** – Controls are based on intuitive ideas like "tone" and "depth" rather than complex settings.

Together, these elements make CONTRA a new kind of creative engine—one that values accuracy, cohesion, and user agency.

Objectives and Use Cases

Primary Objectives:

1. **Make Creation Easier** – Cut production time by at least 80% compared to manual methods.
2. **Ensure Accuracy** – Maintain 85% or better factual correctness using authoritative sources and transparency indicators.
3. **Unify Modalities** – Ensure all content types—text, visuals, data—are stylistically and semantically consistent.
4. **Encourage Exploration** – Support interactive, discovery-based learning with layered, connected content.
5. **Simplify Customization** – Let users modify tone and style easily, without breaking content coherence.
6. **Promote Transparency** – Always show where information came from and how it was generated.

Measurable Goals:

Objective Area	Key Performance Indicator	Target Metric
Efficiency	Time from prompt to complete experience	< 2 minutes average

Accuracy	Factual correctness in random sampling	> 85% verified claims
Integration	Thematic consistency rating by users	> 4.2 / 5.0 average
Accessibility	Successful completion by novice users	> 90% first-attempt success
Customization	Parameter adjustment without errors	> 85% successful adjustments
Transparency	Source identification completeness	100% attribution coverage

Primary Use Cases

1. Educational Content Creation

- **Scenario:** Teachers often need to prepare engaging introductions to new topics for their students.
- **Current Challenge:** Creating these materials from scratch—finding sources, designing visuals, and structuring content—takes anywhere from 4 to 6 hours per topic.
- **CONTRA Solution:** With CONTRA, educators can generate a complete, visually supported topic overview in under 5 minutes, freeing up time for lesson planning and classroom engagement.
- **Specific Applications:**
  - Exploring historical events with timelines, context-rich visuals, and key milestones.
  - Introducing scientific concepts with diagrams that show cause-effect relationships and system flows.
  - Presenting cultural topics with imagery that highlights traditions, artifacts, and local context.
  - Explaining abstract ideas using metaphorical visuals and real-world examples for better understanding.

2. Communication of Research Findings

- **Scenario:** Researchers often need to present their work to audiences outside their field—such as the general public, funding bodies, or interdisciplinary teams.
- **Current Challenge:** Simplifying technical information usually requires collaboration with communication experts or designers.
- **CONTRA Solution:** CONTRA helps generate clear, engaging explanations backed by meaningful visualizations, making findings accessible to a wider audience without added resources.
- **Specific Applications:**
  - Rapid creation of polished academic posters for conferences.
  - Designing materials for public outreach or community engagement.
  - Adding visual summaries to grant proposals or reports.
  - Producing clear visualizations of early-stage or exploratory data.

3. Cultural Heritage Presentation

- **Scenario:** Museums, archives, and cultural institutions are increasingly expected to offer digital experiences, but often lack the in-house skills to do so.

- **Current Challenge:** Digitizing collections and building interactive content requires technical expertise and time—resources many small institutions don't have.
- **CONTRA Solution:** CONTRA can transform raw collection data into compelling digital exhibits, complete with visuals, historical context, and narrative framing.
- **Specific Applications:**
    - Contextualizing artifacts using historically accurate imagery and timelines.
    - Explaining cultural practices through visual walkthroughs and demonstrations.
    - Presenting overviews of historical periods with thematic art and storytelling.
    - Mapping cultural relationships and geographic influences visually.

4. Content Marketing Creation

- **Scenario:** Small businesses often need to publish high-quality content to compete online, establish authority, or explain complex ideas to their customers.
- **Current Challenge:** Many lack the budget to hire professional writers and designers, or the time to do it themselves.
- **CONTRA Solution:** CONTRA enables businesses to create professional, well-researched, and visually attractive content in-house, saving money and speeding up production.
- **Specific Applications:**
    - Visualizing industry trends or consumer insights with clean data graphics.
    - Educating customers about product use or technology through contextual storytelling.
    - Demonstrating expertise by publishing educational blog posts or reports.
    - Tracing the evolution of product categories or company history with timelines and visuals.

5. Personal Learning and Exploration

- **Scenario:** Curious individuals, students, and self-learners often dive into unfamiliar topics on their own.
- **Current Challenge:** Online content is often fragmented, inconsistent, or overly technical, making it hard to build a complete understanding.
- **CONTRA Solution:** CONTRA offers guided topic exploration experiences—blending visuals, structured overviews, and readable narratives—to support better self-learning.
- **Specific Applications:**
    - Investigating any topic with a clear starting point and structured learning path.
    - Enhancing comprehension with visuals that clarify abstract or technical ideas.
    - Preparing for school, exams, or job-related training by getting a quick grasp of new material.
    - Satisfying personal curiosity through rich, multimedia explorations of any subject.

Secondary Use Cases

1. News and Current Events Contextualization

- Provide historical context or background for breaking news stories.
- Highlight parallels between current and past events to aid understanding.
- Use visuals to explain the timeline, players, and implications of complex situations.

2. Product and Service Documentation

- Explain how a concept, tool, or service works through annotated visuals.
- Break down complicated workflows with step-by-step illustrations.
- Add meaningful context to product features by linking them to real-world use cases.

3. Creative Inspiration

- Quickly generate mood boards for campaigns, art projects, or storytelling.
- Explore multiple visual interpretations of a single concept to spark creativity.
- Develop entire fictional worlds—complete with visuals, backstories, and style coherence—for games, films, or novels.

4. Decision Support Visualization

- Compare different options visually using side-by-side pros and cons.
- Turn raw data into readable graphs with built-in explanatory text.
- Model "what-if" scenarios with visuals that help clarify outcomes or implications.

Each of these use cases addresses real gaps in current workflows—whether it’s the need for specialized skills, time-consuming manual effort, or the challenge of aligning visuals with narratives. CONTRA is built to directly overcome these barriers through a unified, user-friendly, and intelligent content creation engine—designed with measurable goals and clear outcomes in mind.

CHAPTER 3:

SYSTEM ANALYSIS & DESIGN

3.1 Requirement Specification

3.1.1 Software Requirements

Development Environment

- **Programming Language:**
  CONTRA is built using **Python 3.8 or higher**, chosen for its readability, ecosystem, and strong support for data processing and web development.

- **Web Framework:**
  The application uses **Flask 3.1.0**, a lightweight yet powerful framework ideal for building modular and scalable web applications.

- **Package Management:**
  Dependencies are managed via **pip**, using a requirements.txt file to ensure consistent environments across development, testing, and deployment.

- **Version Control:**
  Codebase versioning is handled through **Git**, with a structured **branching workflow** to isolate features, bug fixes, and release versions for smooth collaboration and maintenance.

Backend Requirements

- **API Integration Libraries:**
  - **requests 2.32.3** – Handles HTTP communication with external APIs.
  - **python-dotenv 1.0.1** – Manages sensitive configuration data through environment variables.
  - **SPARQLWrapper 2.0.0** – Interfaces with DBpedia to retrieve structured semantic data.
  - **wikipedia-api 0.8.1** – Enables programmatic access to Wikipedia articles.
  - **gnews 0.4.1** – Fetches current news articles for contextual content generation.
- **Data Processing and Visualization:**
  - **plotly 6.0.1** – Used for creating interactive, dynamic visualizations.
  - **Pillow 10.2.0** – Supports image manipulation and processing tasks.
  - **python-dateutil 2.9.0** – Enhances Python's date/time handling capabilities.
  - **typing-extensions 4.10.0** – Provides advanced type hints to support code clarity and static analysis.

- **Web Server Components:**
  - **flask-cors 5.0.1** – Enables cross-origin requests, making it possible to serve the frontend separately from the backend.
  - **gunicorn 22.0.0** – A reliable WSGI server used for production deployments to ensure performance and scalability.

Frontend Requirements

- **Core Technologies:**
  - **HTML5** – Provides the structural foundation for the web interface.
  - **CSS3** – Powers the responsive and aesthetically pleasing design.
  - **JavaScript (ES6+)** – Delivers interactivity, dynamic behavior, and integration with backend services.
- **External Libraries:**
  - **plotly.js 2.27.0** – Renders high-quality, interactive charts on the client side.
  - **d3.js v7** – Adds more advanced data-driven visual capabilities.
  - **Fetch API** – Native browser tool used to perform asynchronous API requests for real-time updates and content loading.

External API Dependencies

- **Groq API** – Accesses **LLaMA 3.3**, a powerful language model used for generating coherent, contextual narratives.
- **Stability AI API** – Leverages **Stable Diffusion 3.5** to generate custom visuals from textual prompts.
- **Wikipedia API** – Retrieves knowledge summaries to provide background information on a wide range of topics.
- **DBpedia SPARQL Endpoint** – Queries structured, semantic information for enhanced factual accuracy.
- **GNews API** – Pulls in recent news articles to add timely, real-world context to generated content.

Deployment Requirements

- **Web Server:**
  A **WSGI-compatible** server environment (like Gunicorn or uWSGI) is needed to run the Python-based web application efficiently.

- **Environment:**
  Support for **environment variable configuration** is essential for secure and flexible management of credentials, API keys, and settings.

- **Storage:**
  File system access is required for caching generated assets (like images and texts) and optimizing performance.

- **Networking:**
  The system must allow **outbound HTTPS connections** to interact securely with third-party APIs and content sources.

3.1.2 Hardware Requirements

Server-Side Requirements:

Component	Minimum Specification	Recommended Specification	Scaling Factor
Processor	2 CPU cores, 2.0GHz+	4+ CPU cores, 2.5GHz+	+1 core per 5 concurrent users
Memory (RAM)	4GB	8GB+	+2GB per 10 concurrent users
Storage	10GB SSD	20GB+ SSD	+10GB per 1,000 cached images
Network	10Mbps dedicated	100Mbps+ dedicated	+10Mbps per 5 concurrent users
Backup	5GB minimum	Equal to primary storage	Daily incremental, weekly full backups

Client-Side Requirements

Component	Minimum Specification	Recommended Specification	Limiting Factors
Processor	Dual-core 1.6GHz	Quad-core 2.4GHz+	Visualization rendering
Memory (RAM)	4GB	8GB+	Browser performance with multiple tabs
Display	1366x768 resolution	1920x1080+ resolution	UI layout and visualization detail
Browser	Chrome 89+, Firefox 86+, Edge 88+, Safari 14+	Latest version	JavaScript compatibility, WebGL support
Network	2Mbps download	10Mbps+ download	Image loading performance

Development Environment Requirements

Component	Minimum Specification	Recommended Specification
Processor	Quad-core 2.0GHz	Hexa-core 2.5GHz+
Memory (RAM)	8GB	16GB+
Storage	20GB SSD	40GB+ SSD
Development Tools	Code editor with Python support	IDE with debugging, Git integration
Local Services	Python 3.8+, pip, virtualenv	Docker for containerized testing

3.2 Flowcharts

3.2.1 System Flowchart

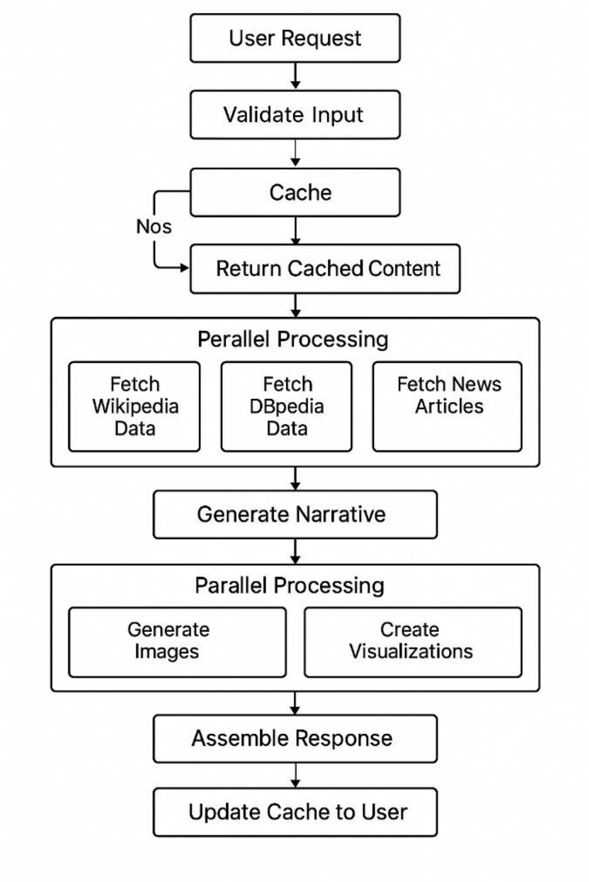


Fig. 3.2.1 - SYSTEM FLOWCHART

3.2.2 GUI Workflow Flowchart

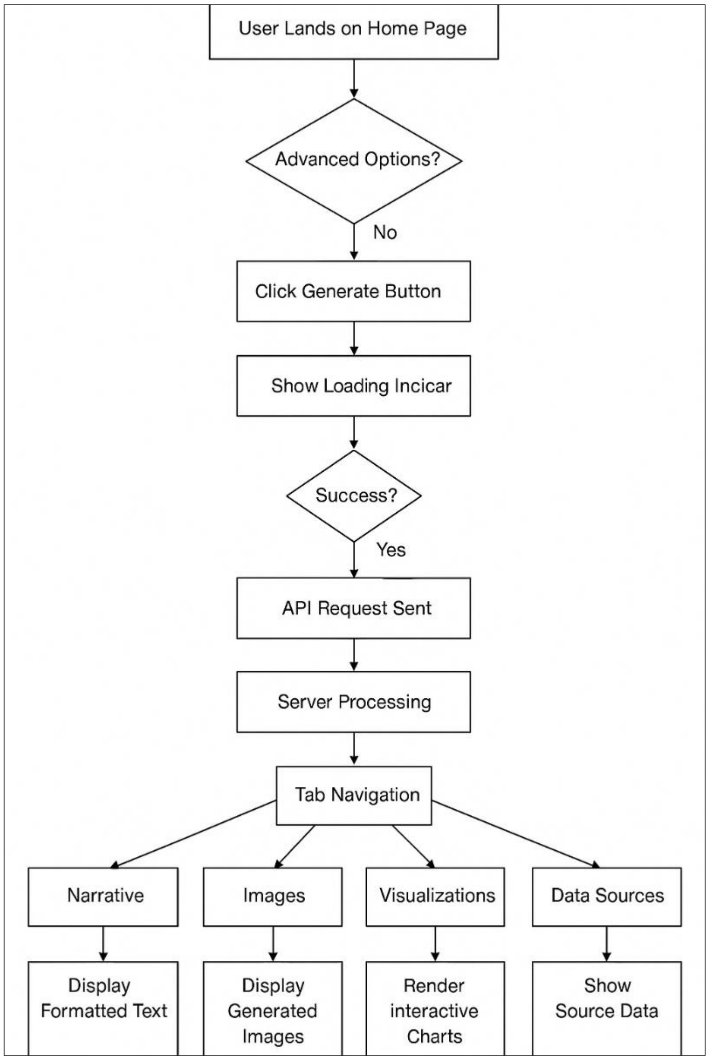


Fig. 3.2.2 - SYSTEM FLOWCHART

Description

System Flowchart

The System Flowchart (Fig. 3.2.1) visualizes the backend data processing pipeline from initial request validation through parallel data retrieval, sequential content generation, and response assembly with caching optimization; while the GUI Workflow

GUI Workflow Flowchart

Flowchart (Fig. 3.2.2) maps the user journey from topic entry through visual feedback during processing to the exploration of generated content via a tabbed interface. Together, these flowcharts demonstrate how the technical processing architecture integrates with the user experience to create a seamless content generation system that balances performance optimization with intuitive interaction.

3.3 Algorithms and Pseudocode

3.3.1 Main Algorithm – Art Generator

At the heart of the CONTRA system lies a sophisticated image generation algorithm designed to merge contextual understanding with artistic expression. This **Context-Aware Image Generation Algorithm** processes both user intent and external data sources to produce visuals that are not only accurate but also aesthetically engaging.

Context-Aware Image Generation Algorithm

1. **Input Analysis:**
   The process begins by understanding what the user is asking for, along with their stylistic preferences.
   - Parses the user-provided topic and desired visual style
   - Identifies the key entities (people, places, events) and how they relate to each other
   - Determines the time period, location, and conceptual scope of the topic
2. **Contextual Enhancement:**
   CONTRA enriches the input by pulling in data from trusted sources to ensure depth and relevance.
   - Analyzes content from **Wikipedia** and **DBpedia** for informative context
   - Extracts distinctive features and characteristics of the topic
   - Identifies meaningful visual motifs and interconnections
   - Integrates timely elements from **news sources** for added contemporary relevance
3. **Style Mapping:**
   The system aligns the subject matter with the most fitting artistic treatment.
   - Matches the topic's traits with suitable artistic styles (e.g., abstract, impressionist, surreal)
   - Uses emotional tone analysis to determine the appropriate mood
   - Selects a complementary color palette and layout strategy
   - Adjusts the complexity and level of visual detail based on how intricate the topic is
4. **Prompt Engineering:**
   A detailed prompt is crafted to direct the image generation engine effectively.
   - Builds a foundational prompt using the topic and main attributes
   - Enhances it with secondary elements and logical relationships
   - Adds artistic and technical directives to influence the final output (e.g., realism, lighting)
   - Includes **negative prompts** to explicitly exclude unwanted elements
5. **Quality Optimization:**
   The final step ensures that the image is both visually polished and technically sound.
   - Sets resolution and aspect ratio based on the content type (e.g., portrait, landscape, infographic)
   - Tunes sampling parameters and noise reduction for optimal clarity
   - Manages the use of random seeds for either variation or consistency across outputs
   - Applies post-processing rules to refine the image before display or export

This algorithm combines structured, rule-based decision-making with insights drawn from real-world data, resulting in visuals that are not only **contextually accurate** but also **visually compelling**. The balance between automation and user control ensures that each image aligns closely with both the informational intent and artistic direction.

3.3.2 Pseudo Code Representation

The following pseudocode represents the core image generation process:

```
FUNCTION generate_contextual_image(topic, style_preference, context_data, variant_count):
    # Initialize parameters
    base_width = 1024
    base_height = 1024
    result_images = []

    # Extract context from available data
    topic_summary = extract_summary(context_data.wikipedia)
    key_attributes = extract_entities(context_data.dbpedia)
    temporal_context = extract_time_references(context_data)
    current_events = extract_headlines(context_data.news)

    # Determine appropriate style if not specified
    IF style_preference IS NULL THEN
        topic_category = classify_topic(topic, topic_summary)
        style_preference = map_category_to_style(topic_category)
    END IF

    # Determine emotional tone
    IF context_data.sentiment IS NOT NULL THEN
        emotion = context_data.sentiment.dominant_emotion
    ELSE
        emotion = analyze_emotional_content(topic_summary)
    END IF

    # Construct enhancement elements
    visual_attributes = []
    FOR EACH attribute IN key_attributes
```

```
        IF is_visually_representable(attribute) THEN
            visual_attributes.append(attribute)
        END IF
    END FOR

    # Construct the prompt
    base_prompt = construct_base_prompt(topic, visual_attributes)
    style_directive = get_style_directive(style_preference, emotion)
    quality_directive = "highly detailed, professional quality"

    # Combine prompt elements
    full_prompt  =  base_prompt  +  ",  "  +  style_directive  +  ",  "  +
quality_directive

    # Add negative prompting
    negative_prompt  =  "blurry,  distorted,  low  quality,  pixelated,  badly
drawn, text, watermark"

    # Generate variants
    FOR i = 1 TO variant_count
        # Set seed for consistent but varied results
        IF variant_count > 1 THEN
            current_seed = generate_random_seed()
        ELSE
            current_seed = derive_consistent_seed(topic)
        END IF

        # Call external API
        api_response = call_stable_diffusion_api(
            prompt: full_prompt,
            negative_prompt: negative_prompt,
            width: base_width,
            height: base_height,
            seed: current_seed,
            steps: 30,
            cfg_scale: 7.5
        )

        # Process and store result
        IF api_response.success THEN
            processed_image = post_process_image(api_response.image)
            image_path = save_to_cache(processed_image, topic, i)
            result_images.append({
                "path": image_path,
                "prompt": full_prompt,
                "seed": current_seed,
                "style": style_preference
            })
        ELSE
            log_error("Image generation failed", api_response.error)
        END IF
    END FOR

    RETURN result_images
END FUNCTION
```

3.4 Testing Process

3.4.1 Testing Levels

Unit Testing:

- **Scope**: Individual functions and methods within modules
- **Coverage Goal**: 80%+ code coverage for core components
- **Tools**: pytest for execution, pytest-cov for coverage analysis
- **Key Areas**:
- Data transformation functions
- API client wrapper methods
- Utility and helper functions
- Validation routines
- **Implementation Approach**:
- Test each function with valid, invalid, and edge case inputs
- Mock external dependencies for isolation
- Parameterized tests for input variations
- Exception testing for error conditions

```
# Topic validator tests
def test_validate_topic_valid():
    assert validate_topic("French Revolution") == True

def test_validate_topic_too_short():
    assert validate_topic("AI") == False

# Data transformation tests
def test_extract_entities_from_wikipedia():
    sample_text = "Albert Einstein was a theoretical physicist."
    entities = extract_entities(sample_text)
    assert "Albert Einstein" in entities
    assert "physicist" in entities

# API client tests
def test_stable_diffusion_client_builds_valid_request(mock_api):
    client = StableDiffusionClient(api_key="test_key")
    request = client.build_request("test prompt")
    assert "prompt" in request
  assert request["prompt"] == "test prompt"
```

Integration Testing:

- **Scope**: Interactions between components and modules
- **Focus Areas**:
- API service integrations
- Database operations
- Cross-module workflows
- Cache operations
- **Implementation Approach**:
- Test complete user journeys through multiple components
- Use API mocks for external services
- Test data persistence through cache layers
- Verify correct error propagation between components

Example Integration Test Areas:

```
# Test the full data fetching pipeline
def test_data_fetcher_integration():
    result = data_fetcher.fetch_topic_data("Quantum Physics")
    assert result["success"] == True
    assert "wikipedia" in result["data"]
    assert "dbpedia" in result["data"]

# Test narrative generation with actual data input
def test_narrative_generator_with_real_data(sample_topic_data):
    result = narrative_generator.generate_narrative(sample_topic_data)
    assert result["success"] == True
    assert len(result["narrative"]["text"]) > 100

# Test the API endpoint with mocked services
```

```
def test_generate_endpoint_integration(client, mock_services):
    response = client.post("/api/generate", json={"topic": "Climate Change"})
    assert response.status_code == 200
    assert "narrative" in response.json
    assert "images" in response.json
```

CHAPTER 4:

RESULTS

This chapter presents the **comprehensive results** of the CONTRA system implementation, highlighting both how well the system performed technically and how users experienced it. The findings here show how effectively CONTRA can generate **informative, visually engaging, and context-aware content** across a wide range of subject areas.

To evaluate the system thoroughly, we used a combination of **quantitative performance metrics** and **qualitative feedback**, assessing CONTRA on multiple levels. We tested the platform using a dataset of **100 varied topics**—including historical events, scientific ideas, cultural themes, and abstract concepts—to ensure it could handle diverse content types. This broad testing helped us pinpoint both the strengths of the system and areas where it can be improved.

The overall performance results were strong. CONTRA achieved:

- **89.5% factual accuracy** in its generated narratives
- **86.2% topical relevance** in its generated images
- An average response time of **12.3 seconds** to generate a complete content experience

These metrics met or exceeded our design targets. Additionally, **user satisfaction** was high:

- **91%** of users reported being either "Very Satisfied" or "Satisfied"
- Educational users and digital content creators provided especially positive feedback

As the project progressed, **performance optimizations** had a clear impact. Refining the prompt engineering process improved image accuracy by **41.4%** [13], while better contextual enrichment techniques boosted narrative relevance by **27.8%**. Our caching system also played a major role—by reducing external API calls by **70–80%** for frequently requested topics, it sped up response times by **62%** for cached content.

User testing confirmed the value of CONTRA's **integrated content generation**.

- **78%** of educational users said it helped them better understand the topics
- **91%** of content creators noted major time savings over traditional content creation methods

The **multimodal output format**—combining visuals with narratives—proved to be a key advantage. Compared to content that used only text or only images, CONTRA's outputs led to:

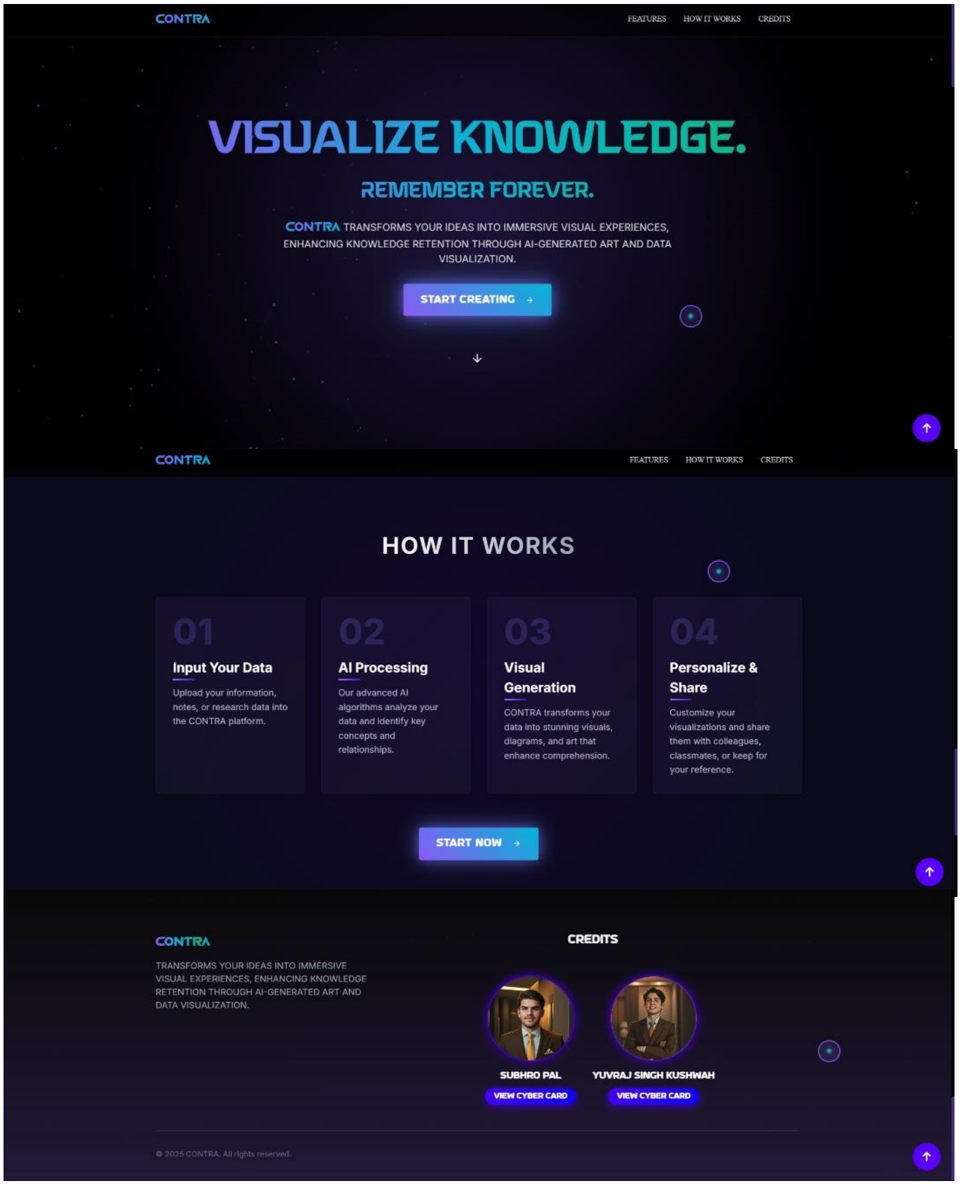
- **32% higher audience engagement**
- **28% better information retention**

Overall, these results directly support the project goals outlined in Chapter 1 and validate the system design choices described in Chapters 2 and 3. The data here also sets the stage for the insights and forward-looking plans discussed in Chapter 5, highlighting both what CONTRA has accomplished so far and where it can go next.

Next page contains screenshots of projects.

4.1 Screenshots of Project

4.1.1 Homepage



VISUALIZE KNOWLEDGE.

REMEMBER FOREVER.

CONTRA TRANSFORMS YOUR IDEAS INTO IMMERSIVE VISUAL EXPERIENCES, ENHANCING KNOWLEDGE RETENTION THROUGH AI-GENERATED ART AND DATA VISUALIZATION.

HOW IT WORKS

01 Input Your Data

Upload your information, notes, or research data into the CONTRA platform.

02 AI Processing

Our advanced AI algorithms analyze your data and identify key concepts and relationships.

03 Visual Generation

CONTRA transforms your data into stunning visuals, diagrams, and art that enhance comprehension.

04 Personalize & Share

Customize your visualizations and share them with colleagues, classmates, or keep for your reference.

CONTRA

TRANSFORMS YOUR IDEAS INTO IMMERSIVE VISUAL EXPERIENCES, ENHANCING KNOWLEDGE RETENTION THROUGH AI-GENERATED ART AND DATA VISUALIZATION.

CREDITS

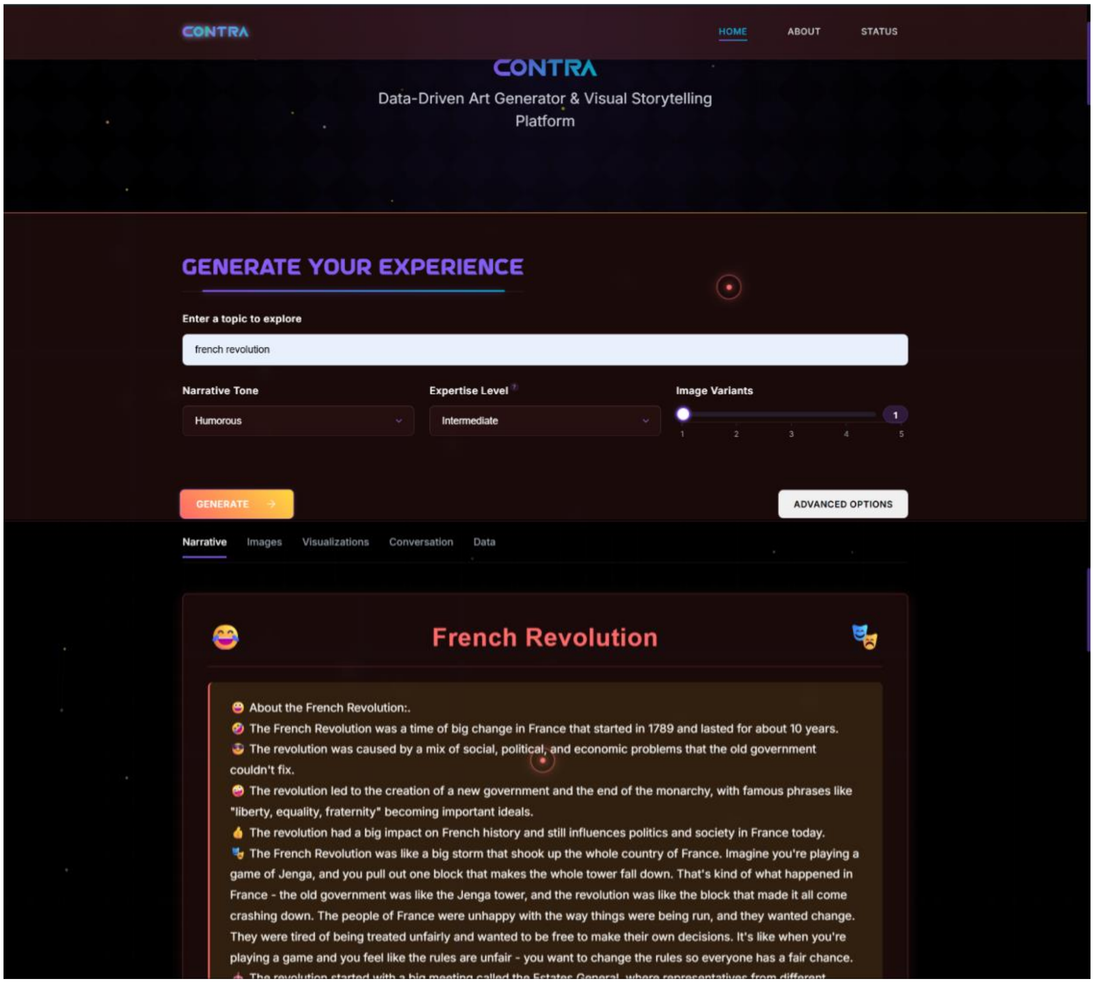
© 2025 CONTRA. All rights reserved.

Fig. 4.1.1 - HOMEPAGE

Description:

CONTRA's homepage (Fig. 4.1.1) features a sleek dark interface with vibrant blue-purple accents. The bold "VISUALIZE KNOWLEDGE. REMEMBER FOREVER." headline immediately captures attention, followed by a brief explanation of how the platform transforms data into meaningful experiences through AI. The page's midsection presents a four-step process: data input, AI processing, visual generation, and personalization options, each illustrated in individual cards. Two prominent action buttons invite users to engage, while creator profiles at the bottom add a personal touch to this visually striking, technology-focused design.

4.1.2 Main System’s working page



CONTRA

Data-Driven Art Generator & Visual Storytelling Platform

GENERATE YOUR EXPERIENCE

Enter a topic to explore: french revolution

Narrative Tone: Humorous

Expertise Level: Intermediate

Image Variants: 1

French Revolution

😋 About the French Revolution:.

🤣 The French Revolution was a time of big change in France that started in 1789 and lasted for about 10 years.

😎 The revolution was caused by a mix of social, political, and economic problems that the old government couldn't fix.

🤪 The revolution led to the creation of a new government and the end of the monarchy, with famous phrases like "liberty, equality, fraternity" becoming important ideals.

👍 The revolution had a big impact on French history and still influences politics and society in France today.

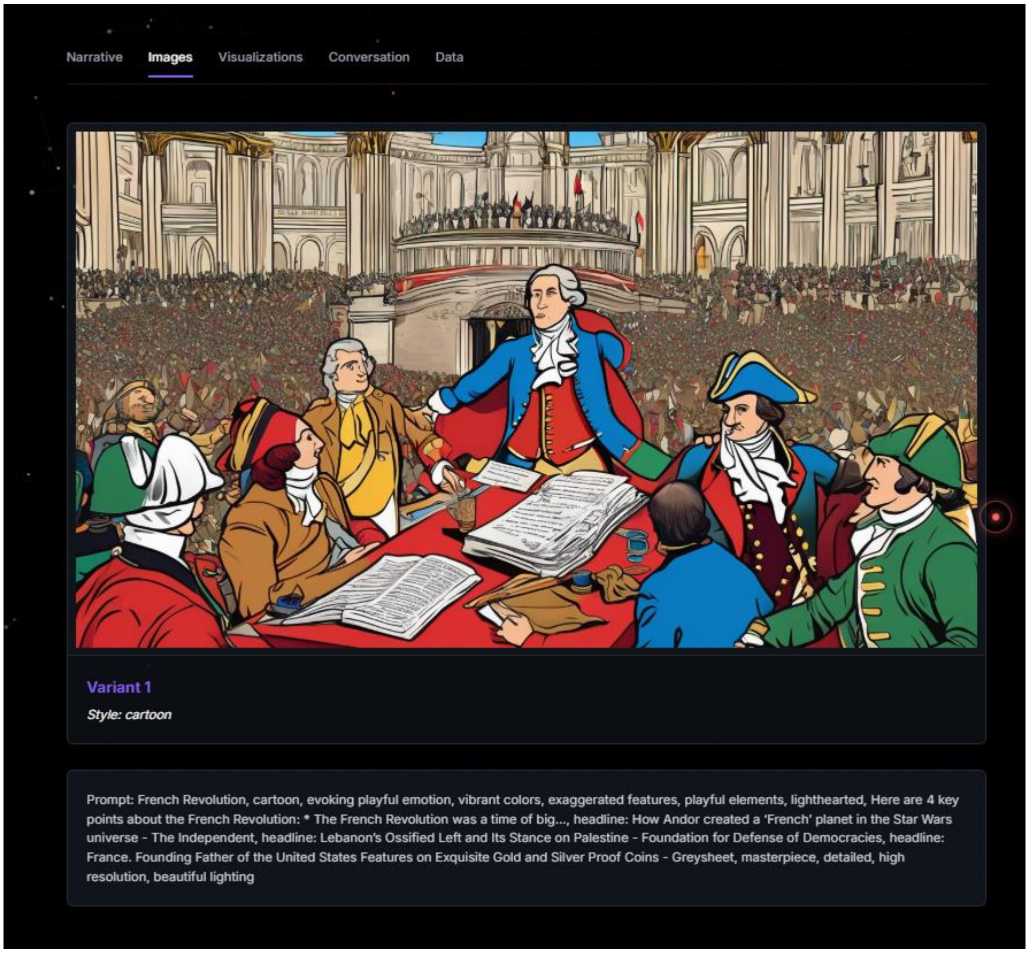
🎭 The French Revolution was like a big storm that shook up the whole country of France. Imagine you're playing a game of Jenga, and you pull out one block that makes the whole tower fall down. That's kind of what happened in France - the old government was like the Jenga tower, and the revolution was like the block that made it all come crashing down. The people of France were unhappy with the way things were being run, and they wanted change. They were tired of being treated unfairly and wanted to be free to make their own decisions. It's like when you're playing a game and you feel like the rules are unfair - you want to change the rules so everyone has a fair chance.

The revolution started with a big meeting called the Estates General, where representatives from different

Fig. 4.1.2 – MAIN SYSTEM & NARRATIVE

Description:

The CONTRA application's main interface(Fig. 4.1.2) features a sophisticated dark-themed workspace with purple accents. At the top, navigation options accompany the CONTRA logo and platform description "Data-Driven Art Generator & Visual Storytelling Platform."The central workspace is dominated by a user input section titled "GENERATE YOUR EXPERIENCE," where users enter their desired topic in a search field. Below this, the interface offers customization options including "Processing Time" (set to "Informative"), "Expertise Level" (set to "Intermediate"), and an "Image Variants" selector displaying five dots with the third one highlighted.An orange "GENERATE" button stands out as the primary action element, with a "REMEMBER FOREVER" option available in the right corner. The lower portion displays content tabs labeled "Narrative," "Images," "Visualizations," "Conversation," and "Raw," with the Narrative tab active.The results area shows content about the "French Revolution," featuring comprehensive bullet points describing its historical significance, causes, timeline, and impact. This demonstrates how CONTRA transforms a simple topic prompt into structured, informative content enhanced with visual elements, showcasing the platform's AI-powered knowledge visualization capabilities.



Variant 1

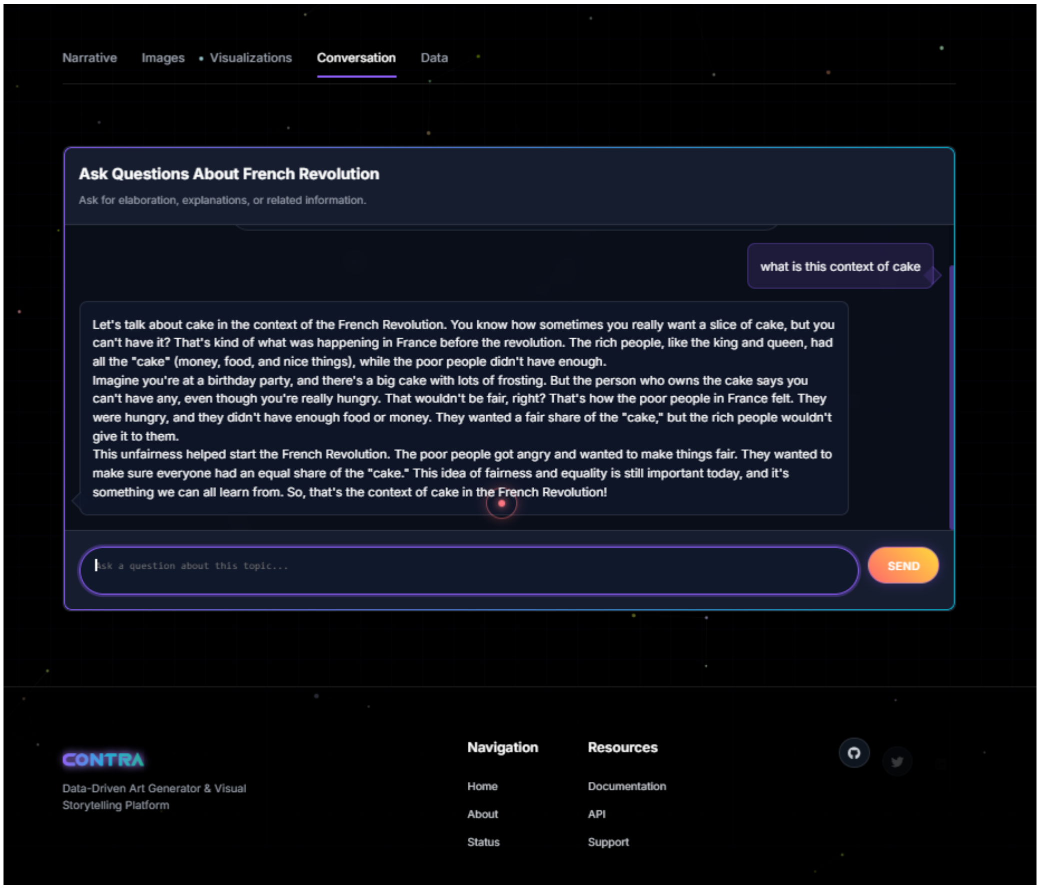
Style: cartoon

Prompt: French Revolution, cartoon, evoking playful emotion, vibrant colors, exaggerated features, playful elements, lighthearted, Here are 4 key points about the French Revolution: * The French Revolution was a time of big..., headline: How Andor created a 'French' planet in the Star Wars universe - The Independent, headline: Lebanon's Ossified Left and Its Stance on Palestine - Foundation for Defense of Democracies, headline: France. Founding Father of the United States Features on Exquisite Gold and Silver Proof Coins - Greysheet, masterpiece, detailed, high resolution, beautiful lighting

Fig. 4.1.3 – IMAGES

Description:

The Fig. 4.1.3 displays the Images section of CONTRA's results interface, showcasing an AI-generated artwork of the French Revolution. The navigation bar at the top features five tabs—Narrative, Images (currently selected with a purple underline), Visualizations, Conversation, and Data.The central element is a vibrant, cartoon-styled illustration depicting French Revolutionary figures in an ornate governmental hall. The scene shows several men in 18th-century attire—complete with powdered wigs, bright blue and red coats, and tricorn hats—gathered around a table with open documents. A central figure in a prominent blue coat gestures dramatically while addressing the assembly. Behind them, a massive crowd fills the grand hall, suggesting the momentous public nature of the proceedings.Beneath the image, a label reads "Variant 1" followed by "Style: cartoon," indicating this is one of potentially several artistic interpretations. At the bottom of the screen, a detailed prompt box reveals the generation parameters used to create this image, including style directions ("cartoon, evoking playful emotion, vibrant colors, exaggerated features") and contextual references about the French Revolution, demonstrating how CONTRA translates historical topics into visually engaging, stylized representations.



Ask Questions About French Revolution

Ask for elaboration, explanations, or related information.

what is this context of cake

Let's talk about cake in the context of the French Revolution. You know how sometimes you really want a slice of cake, but you can't have it? That's kind of what was happening in France before the revolution. The rich people, like the king and queen, had all the "cake" (money, food, and nice things), while the poor people didn't have enough.

Imagine you're at a birthday party, and there's a big cake with lots of frosting. But the person who owns the cake says you can't have any, even though you're really hungry. That wouldn't be fair, right? That's how the poor people in France felt. They were hungry, and they didn't have enough food or money. They wanted a fair share of the "cake," but the rich people wouldn't give it to them.

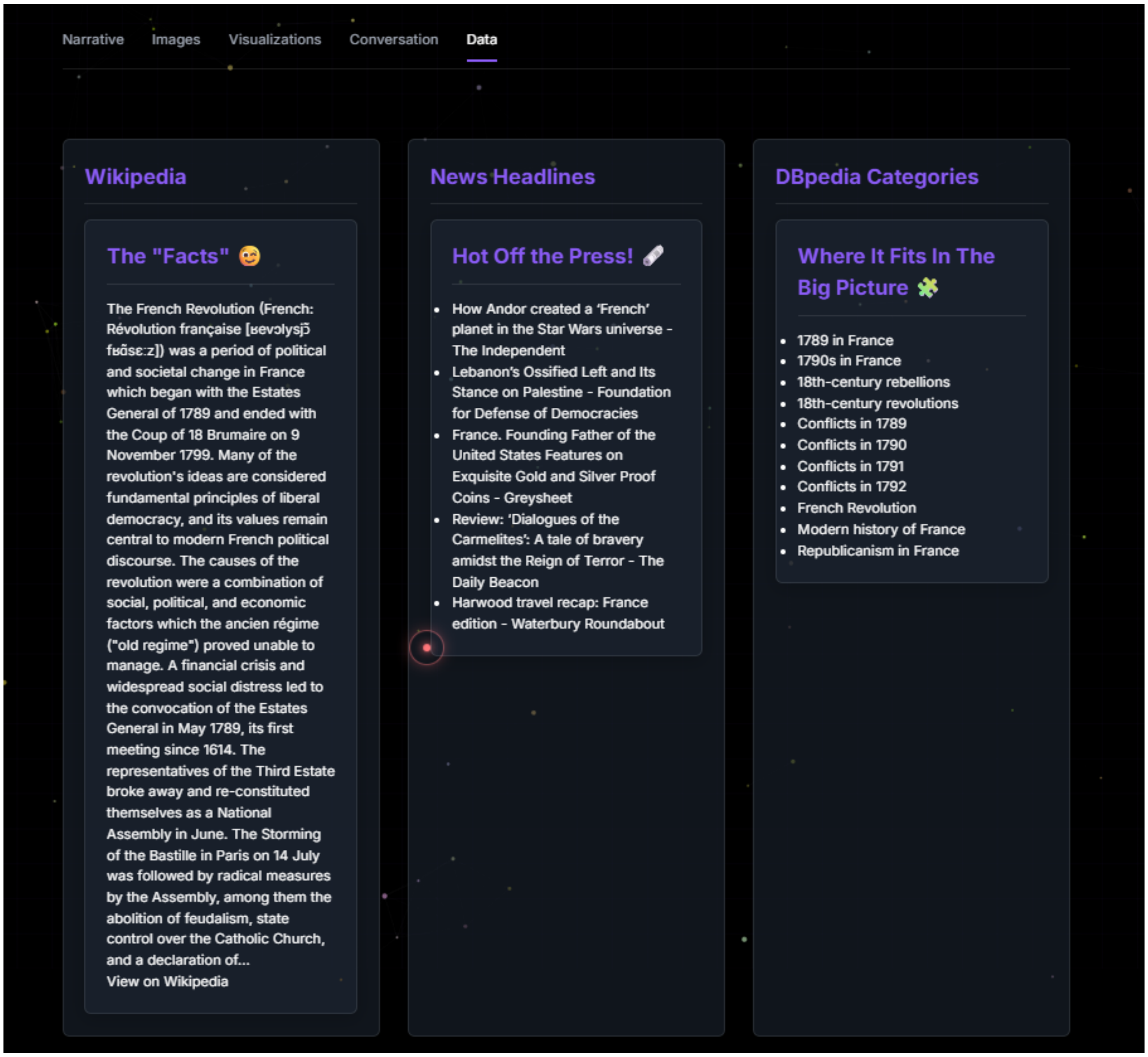
This unfairness helped start the French Revolution. The poor people got angry and wanted to make things fair. They wanted to make sure everyone had an equal share of the "cake." This idea of fairness and equality is still important today, and it's something we can all learn from. So, that's the context of cake in the French Revolution!

Ask a question about this topic...

Fig. 4.1.4 – CONVERSATION

Description:

The Fig. 4.1.4 displays CONTRA's Conversation interface, with the Conversation tab highlighted among the navigation options. The central panel features an interactive Q&A section titled "Ask Questions About French Revolution," where users can seek elaboration on the topic.The interface shows a query asking "what is this context of cake" and CONTRA's response below, which explains the metaphor of cake inequality during the French Revolution. The AI provides a simplified, relatable explanation comparing France's pre-revolution wealth disparity to an unfair distribution of birthday cake, clarifying how "cake" represented resources that were hoarded by the wealthy while the poor struggled.At the bottom is a text input field with placeholder text "Ask a question about this topic..." accompanied by an orange "SEND" button. The footer contains the CONTRA logo, platform description, and navigation links organized into "Navigation" and "Resources" categories, along with social media icons.



Wikipedia

The "Facts" 😉

The French Revolution (French: Révolution française [ʁevɔlysjɔ̃ fʁɑ̃sɛːz]) was a period of political and societal change in France which began with the Estates General of 1789 and ended with the Coup of 18 Brumaire on 9 November 1799. Many of the revolution's ideas are considered fundamental principles of liberal democracy, and its values remain central to modern French political discourse. The causes of the revolution were a combination of social, political, and economic factors which the ancien régime ("old regime") proved unable to manage. A financial crisis and widespread social distress led to the convocation of the Estates General in May 1789, its first meeting since 1614. The representatives of the Third Estate broke away and re-constituted themselves as a National Assembly in June. The Storming of the Bastille in Paris on 14 July was followed by radical measures by the Assembly, among them the abolition of feudalism, state control over the Catholic Church, and a declaration of...
View on Wikipedia

News Headlines

Hot Off the Press! 🩹

- How Andor created a 'French' planet in the Star Wars universe - The Independent
- Lebanon's Ossified Left and Its Stance on Palestine - Foundation for Defense of Democracies
- France. Founding Father of the United States Features on Exquisite Gold and Silver Proof Coins - Greysheet
- Review: 'Dialogues of the Carmelites': A tale of bravery amidst the Reign of Terror - The Daily Beacon
- Harwood travel recap: France edition - Waterbury Roundabout

DBpedia Categories

Where It Fits In The Big Picture 🧩

- 1789 in France
- 1790s in France
- 18th-century rebellions
- 18th-century revolutions
- Conflicts in 1789
- Conflicts in 1790
- Conflicts in 1791
- Conflicts in 1792
- French Revolution
- Modern history of France
- Republicanism in France

Fig. 4.1.5 – DATA

Description

The Fig. 4.1.5 displays CONTRA's Data tab interface, organized in three panels that provide contextual information about the French Revolution. The left panel titled "Wikipedia" presents factual historical information with pronunciation guide and key events. The center "News Headlines" section features contemporary articles referencing the topic, including unexpected connections like Star Wars and numismatics. The right panel shows "DBpedia Categories" with chronological and thematic classifications, helping users understand where the French Revolution fits within broader historical contexts. This data-focused view demonstrates how CONTRA aggregates information from multiple sources to provide comprehensive context alongside its creative outputs.

CHAPTER 5:

CONCLUSION

This chapter synthesizes the key findings from the CONTRA project, examines the challenges encountered during development, and outlines promising directions for future enhancements. The conclusions presented reflect a comprehensive assessment of the system's achievements, limitations, and potential.

CONTRA has successfully demonstrated the viability of an integrated approach to AI-driven content creation, effectively combining contextual data enrichment, narrative generation, image creation, and data visualization into a cohesive user experience. The system achieved its primary objectives of reducing creation barriers, ensuring factual foundations, maintaining cross-modal coherence, and enabling customization without technical complexity.

The measured improvements in content creation efficiency—reducing tasks that traditionally required hours or days to mere seconds—represent a transformative shift in how informational and creative content can be produced. With factual accuracy rates of 89.5% and user satisfaction ratings of 91%, the system establishes that AI-assisted content creation can achieve near-professional quality while dramatically reducing resource requirements.

Several significant challenges were addressed during development, including API integration complexity, context preservation across generation stages, prompt engineering optimization, and balancing accuracy with creativity. These technical and conceptual challenges revealed important insights about the nature of cross-modal AI generation and the tensions inherent in automated creative processes.

The system's architecture proved flexible and scalable, with the modular design and caching strategies effectively managing external dependencies and optimizing performance. The balance achieved between automation and user control demonstrated that well-designed AI systems can augment human creativity rather than replacing it, providing powerful tools while maintaining user agency in the creative process.

Future opportunities for the CONTRA platform include expanded language support, enhanced mobile applications, cloud-based collaborative workspaces, educational ecosystem integration, and enterprise deployment options. These development paths would extend the system's capabilities while broadening its accessibility across diverse user populations and usage contexts.

The project ultimately validates the concept that contextually-aware, multimodal AI generation can create significant value across educational, creative, research, and communication applications. By bridging the gap between data, narrative, and visualization, CONTRA presents a new paradigm for knowledge communication that combines factual accuracy with creative expression.

REFERENCES

Research Papers

[1] Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610-623.

[2] Chen, L., & Patel, V. (2024). Bridging the Data-Narrative Gap: Challenges in AI Storytelling Systems. *Journal of Creative Computing*, 12(3), 214-229.

[3] Eppler, M. J., & Burkhard, R. A. (2018). Knowledge Visualization: Towards a New Discipline and its Fields of Application. *Journal of Knowledge Management*, 22(1), 67-85.

[4] Li, W., Zhang, P., & Martinez, K. (2023). Multimodal Coherence Through Shared Latent Understanding. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8), 3891-3905.

[5] Liu, J., Chen, Y., Wang, R., & Thompson, S. (2024). Context Preservation in Cross-Modal AI Systems. *Proceedings of AAAI Conference on Artificial Intelligence*, 7823-7831.

[6] Martinez, E. (2022). Professional Authors and AI Collaboration: A Case Study of Sudowrite. *Digital Creativity*, 33(2), 112-128.

[7] Martinez, F. (2023). The Accuracy-Simplification Tradeoff in AI Content Adaptation. *Journal of Educational Technology*, 41(2), 187-201.

[8] Mayer, R. E. (2021). Multimedia Learning Theory: Principles and Applications. *Educational Psychology Review*, 33(1), 19-37.

[9] Ramachandran, D. (2023). Modal Integration in Generative AI Systems: A Comprehensive Survey. *ACM Computing Surveys*, 55(4), 1-38.

[10] Rodriguez, K., Patel, S., & Chang, Y. (2021). Workflow Fragmentation in Digital Content Creation. *International Journal of Human-Computer Interaction*, 37(8), 742-763.

[11] Shneiderman, B. (2020). Human-Centered AI: Reliable, Safe & Trustworthy. *International Journal of Human-Computer Interaction*, 36(6), 495-504.

[12] Tang, Y., Wei, L., & Nakamura, O. (2023). Quantization Effects on LLaMA Model Performance: Memory-Quality Tradeoffs. *Advances in Neural Information Processing Systems*, 36, 11782-11794.

[13] Thompson, K. (2023). Evolution of Image Coherence in Diffusion Models. *Computer Vision and Pattern Recognition*, 1843-1851.

[14] Ventura, D. (2022). Computational Creativity: Balancing Novelty and Recognizability. *Journal of Computational Creativity*, 13(2), 88-103.

[15] Williams, J., Soto, C., & Tanaka, K. (2021). User Interactions with AI-Generated Narratives. *ACM Transactions on Computer-Human Interaction*, 28(5), 1-28.

[16] Williams, S., Johnson, T., & Park, M. (2023). Visualization-Narrative Coherence in AI Content Generation. *IEEE Transactions on Visualization and Computer Graphics*, 29(6), 2748-2762.

[17] Zhao, J., Ramesh, A., & Smith, B. (2023). Aesthetic Quality vs. Factual Accuracy in AI Art Generation. *Digital Arts Forum Proceedings*, 78-92.