



Web Owl Multi-Agent RAG System

Intelligent Website Navigation & Information Retrieval

1. Project Introduction

The Problem We Solve

Information Overload

Users get lost in complex websites.

Poor Search Results

Generic search engines lack context.

Navigation Complexity

Multi-step processes are confusing.

Fragmented Information

Content scattered across multiple pages.

No Intelligent Guidance

Users need step-by-step navigation help.

"A student wants to apply for a Master's program but spends hours clicking through different pages, downloading PDFs, and still misses crucial requirements buried in documentation."



Introducing Web Owl



Intelligent Understanding

Comprehends website structure, content hierarchies, and navigation patterns across multiple pages.



Multi-Agent Intelligence

Four specialized AI agents work together, each with specific expertise and responsibilities, ensuring quality through multi-layer verification.



Smart Navigation Guidance

Provides step-by-step navigation paths, recommends optimal routes, and suggests related content based on user intent.



Project Scope & Vision

Target Applications:

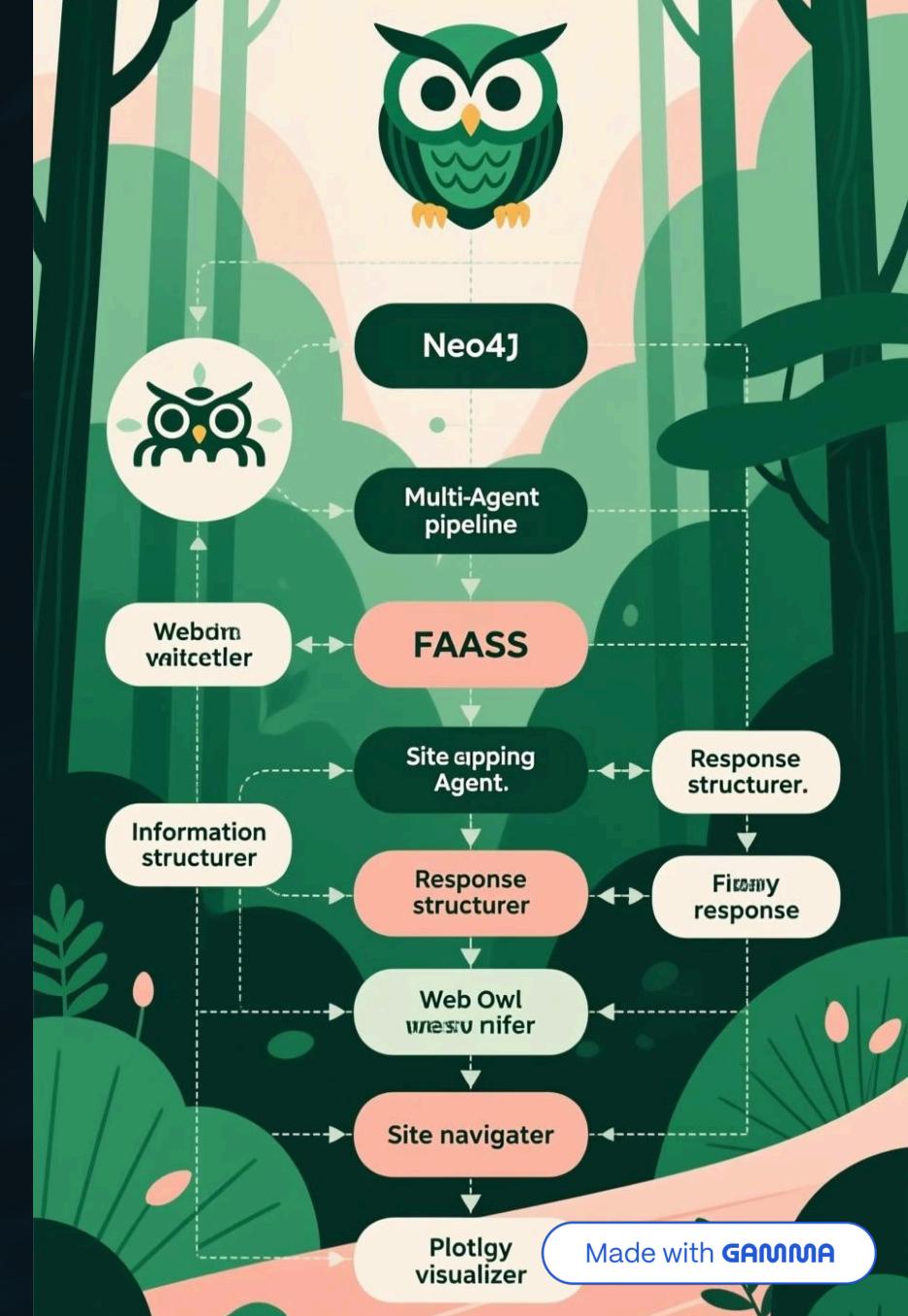
- Educational Institutions
- Corporate Websites
- Government Portals
- Healthcare Systems



"First RAG system that doesn't just find information—it teaches users how to navigate and discover content intelligently."

2. System Architecture & Innovation

🏗️ High-Level Architecture





Core Innovation: Multi-Modal Knowledge Representation

Traditional RAG

✗ Text chunks only

✗ Flat document storage

✗ Simple similarity search

✗ Single-agent processing

✗ Information only

Web Owl RAG

✓ Multi-modal content (HTML, PDF, images, tables)

✓ Rich graph relationships

✓ Hybrid semantic + graph search

✓ Multi-agent intelligence

✓ Information + Navigation guidance



Data Flow & Processing Pipeline

01

Content Ingestion

Website URL → Intelligent Crawler →
Multi-format Extraction → Content
Classification → Relationship
Mapping → Knowledge Graph

02

Knowledge Preparation

Graph Database → Text Chunking →
Vector Embeddings → FAISS Index →
Site Structure Analysis →
Navigation Maps

03

Query Processing

User Query → Retrieval → Multi-
Agent Processing → Response
Generation

3. Technical Deep Dive

Intelligent Web Crawler

Advanced Content Extraction

```
def extract_content(url, soup):
    content = {} # Multi-format support
    content["text"] = clean_text_extraction(soup)
    content["images"] = collect_images(soup)
    content["tables"] = extract_structured_tables(soup)
    content["pdfs"] = process_pdf_documents(soup)
    return content
```

Key Features:

- Smart Content Classification
- Relationship Mapping
- PDF Intelligence
- Depth-First Navigation



Neo4j Knowledge Graph Design

Sophisticated Schema

```
// Node Types
(:Page {url, title, text_len})
(:Asset {url, type, filename})
(:Chunk {id, modality, text})

// Rich Relationships
(Page)-[:LINKS_TO]->(Page) // Navigation flow
(Page)-[:CONTAINS]->(Asset) // Asset ownership
(Page)-[:HAS_CHUNK]->(Chunk) // Content chunks
(Asset)-[:HAS_CHUNK]->(Chunk) // Asset content
```

Advanced Capabilities:

- Bidirectional Relationships
- Content Versioning
- Semantic Tagging



Hybrid Search Engine

Four Intelligent Search Modes

1. Semantic Search

```
def semantic_search(self, query: str, top_k: int = 5):  
    query_embedding =  
        self.embedder.encode([query])  
    scores, indices =  
        self.faiss_index.search(query_embedding, top_k)  
    return self.enrich_results(indices, scores)
```

2. Graph Walk Search

```
def graph_walk_search(self, query: str, max_depth:  
    int = 2):  
    return context_aware_results
```

3. Hybrid Search

```
def hybrid_search(self, query: str,  
    semantic_weight: float = 0.7, graph_weight: float =  
    0.3):  
    return best_of_both_worlds
```

4. Multimodal Search

- Cross-format content discovery
- Related asset identification
- Comprehensive context building