

# Agentic AI Coding Assistant for Repository

## Introduction:

We designed and deployed an advanced Agentic AI Coding Assistant to help developers understand, navigate, and transform large-scale organizational codebases. The assistant is built using a multi-agent architecture orchestrated by a central controller. It delivers deep repository reasoning, multi-file and multi-repo editing, document understanding, online information retrieval, and personalized long-term memory.

With direct GitHub integration and powerful retrieval-augmented reasoning, the system elevates developer productivity by automating code exploration, refactoring, debugging, and documentation tasks that traditionally consume hours of manual effort.

## Client Details:

**Name:** Confidential | **Industry:** Finance, Software | **Location:** USA

## Technologies:

- Python
- Streamlit
- FastAPI
- ChatGPT-5
- LangChain
- Qdrant Vector Database

# Agentic AI Coding Assistant for Repository

## Project Description:

### Challenges:

Organizations with large engineering teams often maintain **hundreds of repositories** spread across services, microservices, SDKs, tools, and automation pipelines. This introduces several recurring challenges:

- Locating where specific functions, classes, or patterns are defined
- Understanding how components interact across repositories
- Identifying call flows and dependencies
- Performing consistent, error-free multi-repo refactoring
- Manipulating very large files that exceed typical LLM token limits
- Handling vague developer queries that span code, documents, API references, and web knowledge
- Maintaining conversational and contextual memory over time
- Retrieving context from past work sessions or multi-day tasks

A typical coding assistant cannot operate at this scale. Developers needed a solution capable of **interacting seamlessly with GitHub**, analyzing code across multiple repositories, and applying consistent updates across deeply nested folder structures.

### The Solution: A Multi-Agent, GitHub-Integrated Coding Assistant

The system uses a **central Master Agent** that orchestrates interactions between specialized agents for code analysis, document reasoning, web research, and knowledge retrieval.

Every user query flows through this pipeline:

**User → Master Agent → Specialized Agent → VectorDB/GitHub/Web → LLM → Result**

This architecture ensures accurate intent routing, scalability, and modular processing.

# Agentic AI Coding Assistant for Repository

## Agents and Their Capabilities

### 1. Master Agent (Orchestrator)

Master Agent is the coordination center responsible for:

- Intent detection
- Routing queries to the correct agent
- Managing multi-step tasks
- Maintaining context across the conversation
- Ensuring smooth transitions between code, doc, web, and knowledge reasoning

### 2. Code Agent: Repository Intelligence with Direct GitHub Integration

The Code Agent is the system's most advanced component, designed for **enterprise-level, multi-repo environments**.

#### GitHub Integration

- Direct connection to the organization's GitHub
- On-demand retrieval of repositories, files, folder trees, commits, PRs, and issues
- Ability to process **dozens or even hundreds of repositories** as part of a single enterprise setup
- Every file becomes searchable, analyzable, and editable through vector-based retrieval
- Supports both public and private repos within the organization's GitHub organization

#### Internal Sub-Agents

- **Commits Agent** – analyzes commit histories and author-level changes
- **PR/Issues Agent** – interprets discussions, rationales, and issue descriptions
- **Source Code Agent** – understands and retrieves content from GitHub repositories

#### Core Capabilities

- Explain any class, function, method, or module and identify where it resides in the repo(s)
- Answer vague questions and locate the exact file(s) relevant to the query
- Debug, refactor, rewrite, or optimize a single file
- Modify variables, functions, and classes with precision
- Show every location where a function is invoked across multiple repos
- Maintain seamless follow-up context

# Agentic AI Coding Assistant for Repository

- Return fully updated files with download support

## Advanced Multi-Repo Capabilities (GitHub Enhanced)

### Large File Handling

- Processes files larger than LLM token limits through chunking and reconstruction.

### Folder-Level Batch Editing

- Updates all files inside a selected folder in any GitHub repository.

### Recursive Subfolder Updates

- Automatically traverses nested directories to apply changes.

### Multi-Repository Editing:

This is where the GitHub integration truly shines:

- Searches across all repositories in the organization
- Identifies every impacted file
- Applies consistent updates across multiple repos
- Returns complete updated versions for download

This transforms the assistant into a **cross-repository refactoring engine** — a capability not typically seen in AI coding assistants.

### Train Mode

- The system remembers user instructions, project patterns, naming conventions, and architectural rules.

## 3. Web Agent: Real-Time Information Retrieval

The Web Agent answers queries based on fresh online information.

### Capabilities

- Fetches library documentation, syntax updates, best practices
- Retrieves latest community knowledge, blog posts, or specifications
- Enhances LLM reasoning by supplementing missing knowledge

# Agentic AI Coding Assistant for Repository

## 4. Doc Agent: Document Understanding & Knowledge Ingestion

The Doc Agent handles all forms of user-uploaded files and organizational documents.

### Modes

1. **Train Mode**
  - Ingest documents into VectorDB
  - Makes them permanently searchable
2. **Ask Mode**
  - Answers questions based on document content
  - Summaries, explanations, and contextual insights

### Supported Files

- PDFs
- Text documents
- Code files
- Logs
- Markdown
- Configuration files

## 5. Learner Agent: Memory & Contextual Intelligence

Learner Agent is a long-term memory layer supporting organization-wide knowledge.

### Capabilities

- Understands content from Teams messages
- Processes information sent via email
- All the information will be provided by the user.

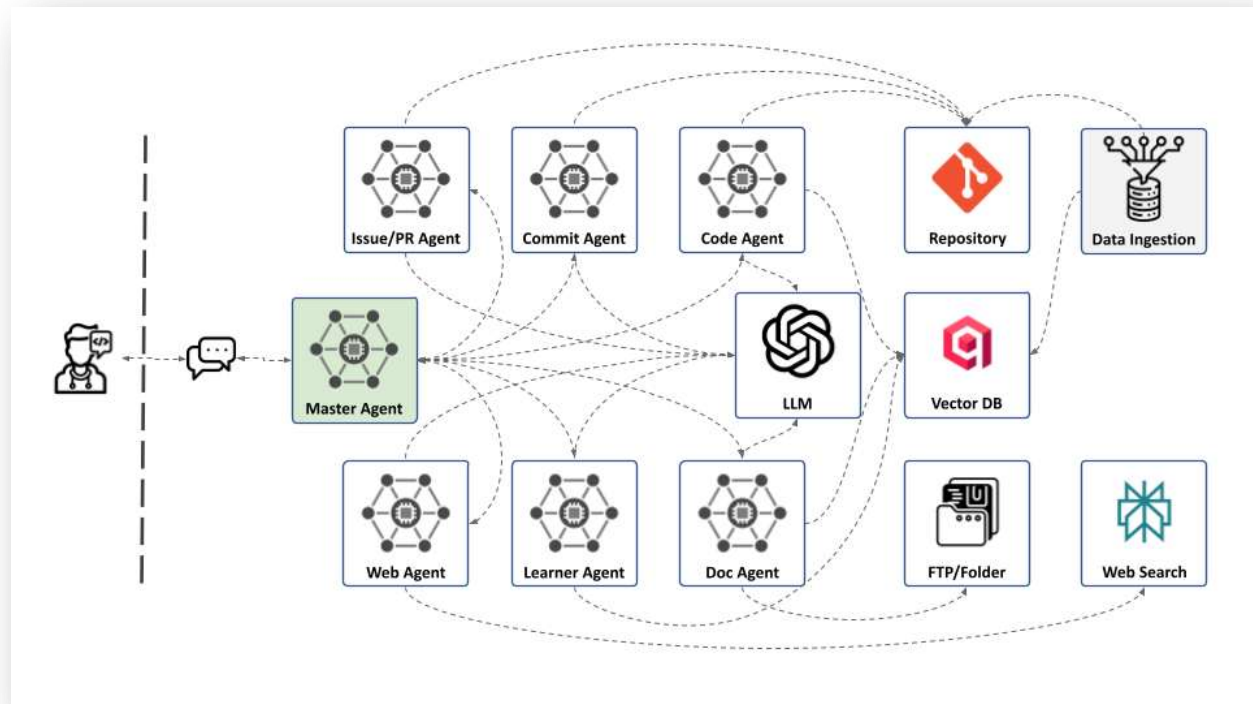
This gives the assistant continuity, improving accuracy and relevance over time.

### Additional Feature: Date-Based Chat History Loading

- Users can filter and load chat sessions by date.
- This is ideal for multi-day refactors, documentation creation, or long-running debugging tasks.

# Agentic AI Coding Assistant for Repository

## Architecture Overview



# Agentic AI Coding Assistant for Repository

## Benefits & Outcomes

### 1. Enterprise-Grade Multi-Repo Code Intelligence

The assistant navigates and updates code across many organizational repos — a capability beyond traditional LLMs.

### 2. Faster Code Understanding

Developers can instantly explore codebases, view call flows, and locate relevant modules.

### 3. Reduced Manual Effort

The assistant automates redundant refactoring, documentation updates, and code exploration across large repositories.

### 4. Enhanced Developer Productivity

Complex multi-file tasks that previously took hours or days can now be performed in minutes.

### 5. Continuous Learning

Train Mode allows the assistant to evolve with the team's coding standards and architectural patterns.

### 6. Unified Knowledge Hub

Code, documents, chat history, Teams messages, emails, and web knowledge are integrated into one intelligent system.

### 7. Improved Accuracy and Consistency

Multi-repo updates ensure consistency across interconnected services.

# Agentic AI Coding Assistant for Repository

## Screenshots:



Fig-1: Home Page and chat interface

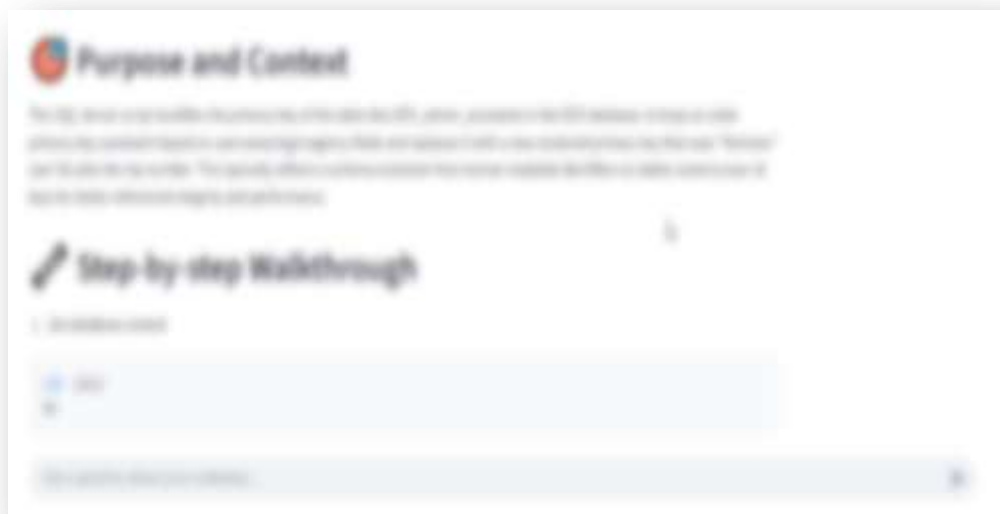


Fig-2: Example for explanation type query



# Agentic AI Coding Assistant for Repository



**Fig-3: Update type user query**

**Fig 4: Vague question**

# Agentic AI Coding Assistant for Repository



Fig-5: Folder type question

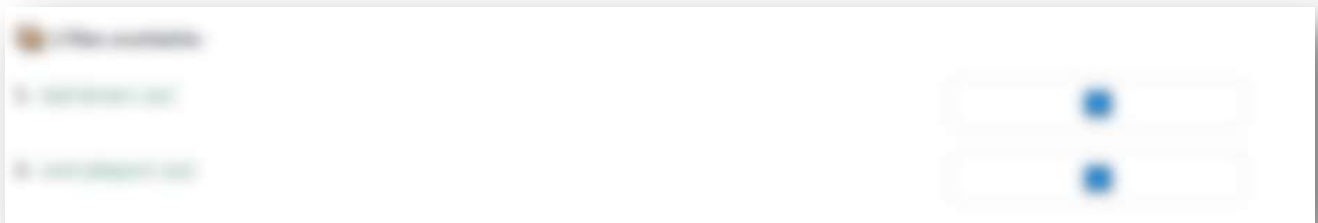


Fig-6: Download option for files