Path :"C:\Users\ttmcn\Desktop\Niamh"

Local machine system specs:  
CPU: AMD Ryzen 9 7900x 12-core, 24-thread 4701 Mhz  
GPU: Nvidia RTX 4060 Ti 16 GB  
RAM: 32 GB  
OS: Windows 11 Home  
Cuda toolkit 12.8  
Development platform: Visual Studios 2022  
LLM Model: Llama3.1 8B   
LLM Source and server: Ollama (local server)

C:\Users\ttmcn\Desktop\Niamh

│── main.py # Entry point for running Niamh

│── requirements.txt # List of required dependencies

│── .env # Environment variables (MongoDB, Redis config, API keys)

│── README.md # Project documentation

│

├── config\ # Configuration files

│ ├── settings.py # Centralized project settings

│ ├── logging\_config.py # Logging configuration

│

├── niamh\ # Core AI system

│ ├── \_\_init\_\_.py

│ ├── niamh.py # Main AI agent (Niamh) logic

│ ├── evolution.py # AI learning, RLHF, and model evolution logic

│ ├── verification.py # Verification for instructions, hallucinations, and responses

│ ├── monitoring.py # System resource monitoring (CPU, RAM, GPU usage)

│

├── agents\ # AI Agent modules

│ ├── \_\_init\_\_.py

│ ├── fact\_checker.py # Web-based fact validation agent

│ ├── instruction\_guard.py # Past instruction adherence agent

│ ├── code\_validator.py # Validates proposed code changes before approval

│ ├── ethics.py # Ensures ethical and safe AI behavior

│ ├── compute\_monitor.py # Tracks system resource usage

│ ├── database\_manager.py # Manages MongoDB and Redis interactions

│ ├── query\_handler.py # Handles information retrieval from memory

│

├── tools\ # AI tools and utilities

│ ├── \_\_init\_\_.py

│ ├── web\_search.py # Agent for trusted online searches

│ ├── data\_analysis.py # Utility for analyzing historical AI interactions

│ ├── security.py # Security and permission enforcement

│ ├── json\_handler.py # Manages AI-generated knowledge storage

│

├── memory\ # AI memory and persistence

│ ├── database.py # MongoDB + Redis connection layer

│ ├── redis\_cache.py # Redis caching and expiration rules

│ ├── migration.py # Manages migration of data from Redis to MongoDB

│

├── database\ # MongoDB storage for AI knowledge and history

│ ├── conversation\_logs.json # JSON-based conversation logs

│ ├── knowledge\_base.json # AI-stored verified knowledge

│ ├── evolution\_logs.json # Stored evolution records

│

├── tests\ # Unit tests for AI behavior and functions

│ ├── \_\_init\_\_.py

│ ├── test\_niamh.py # Unit tests for core AI logic

│ ├── test\_agents.py # Unit tests for AI agents

│ ├── test\_database.py # Unit tests for MongoDB and Redis

│

├── logs\ # Logs and debugging data

│ ├── niamh.log # Main AI log file

│ ├── agent\_logs.log # Logs interactions between agents

│ ├── errors.log # Logs errors for debugging

│

├── scripts\ # Utility scripts for admin and maintenance

│ ├── init\_db.py # Script to initialize MongoDB collections

│ ├── backup\_db.py # Script for backing up MongoDB data

│ ├── reset\_cache.py # Script to clear Redis cache

│

└── data\ # AI learning datasets and model files

├── training\_data\ # Training and fine-tuning datasets

├── embeddings\ # Word and context embeddings

├── model\_weights\ # Pre-trained model weight files

**Breakdown of Key Directories and Files**

**1️⃣ Core AI System (niamh/)**

* **niamh.py** → Primary AI logic, interacts with agents and manages responses.
* **evolution.py** → Handles self-improvement and RLHF learning.
* **verification.py** → Ensures AI responses comply with instructions.
* **monitoring.py** → Tracks system performance.

**2️⃣ AI Agents (agents/)**

Each agent is a specialized module:

* **Fact Checker** → Validates AI responses using trusted sources.
* **Instruction Guard** → Ensures AI follows previous instructions.
* **Code Validator** → Reviews AI-generated code before execution.
* **Ethics Agent** → Ensures safety and ethical considerations.
* **Compute Monitor** → Monitors system resources.
* **Database Manager** → Handles MongoDB & Redis interactions.
* **Query Handler** → Retrieves relevant past data for AI decisions.

**3️⃣ AI Tools (tools/)**

* **web\_search.py** → Queries trusted sources for real-time updates.
* **data\_analysis.py** → Analyzes AI learning history and improvements.
* **security.py** → Enforces AI security policies.
* **json\_handler.py** → Converts AI-generated knowledge into JSON format.

**4️⃣ AI Memory System (memory/)**

* **database.py** → Handles MongoDB and Redis caching.
* **redis\_cache.py** → Manages cache expiration and storage logic.
* **migration.py** → Moves data from Redis to MongoDB at scheduled times.

**5️⃣ Database Storage (database/)**

* **conversation\_logs.json** → Stores AI conversation history.
* **knowledge\_base.json** → Stores facts AI has learned.
* **evolution\_logs.json** → Tracks AI’s growth and improvements.

**6️⃣ Logging System (logs/)**

* **niamh.log** → Logs AI system events.
* **agent\_logs.log** → Tracks interactions between agents.
* **errors.log** → Stores error messages for debugging.

**7️⃣ Scripts for Maintenance (scripts/)**

* **init\_db.py** → Initializes MongoDB collections.
* **backup\_db.py** → Backups database data for security.
* **reset\_cache.py** → Clears old Redis cache.

**8️⃣ AI Data Storage (data/)**

* **training\_data/** → AI training datasets for improvement.
* **embeddings/** → Stores vector embeddings for context recall.
* **model\_weights/** → Pre-trained model weight files.

Revised 2/20/25 @ 1100

C:\Users\ttmcn\Desktop\Niamh\

│── main.py # Entry point to run Niamh & other agents

│── requirements.txt # List of dependencies

│── .env # Environment variables (MongoDB, Redis, API keys)

│

├── niamh\ # Niamh’s core behavior

│ ├── \_\_init\_\_.py

│ ├── niamh.py # Niamh's AI-specific logic (excludes LLM & database calls)

│

├── agents\ # AI agents

│ ├── \_\_init\_\_.py

│ ├── fact\_checker.py # Validates AI responses against trusted sources

│ ├── instruction\_guard.py # Ensures Niamh follows previous instructions

│

├── core\ # Global shared functions (LLM, database, tools)

│ ├── \_\_init\_\_.py

│ ├── llm\_handler.py # Handles all LLM calls (Llama 3.1 via Ollama)

│ ├── database.py # MongoDB & Redis memory system

│ ├── utils.py # Helper functions (logging, config handling)

│

├── logs\ # Log files

│ ├── niamh.log # Logs Niamh’s responses

│

└── tests\ # Testing scripts

├── test\_niamh.py

Update as of 2/20/25 at 1147

C:\Users\ttmcn\Desktop\Niamh\

│── main.py

│

├── niamh\

│ ├── \_\_init\_\_.py

│ ├── niamh.py # Now integrates with the Fact-Checker

│

├── core\

│ ├── \_\_init\_\_.py

│ ├── llm\_handler.py # Handles Llama 3.1 API calls (Ollama)

│ ├── database.py # MongoDB & Redis integration

│ ├── web\_scraper.py # (NEW) Scrapes internet for real-time facts

│

├── agents\

│ ├── \_\_init\_\_.py

│ ├── fact\_checker.py # (NEW) Fact-checks Niamh's responses

│

└── tests\

├── test\_fact\_checker.py

Update as of 2/20/25 at 1151

C:\Users\ttmcn\Desktop\Niamh\

│── main.py

│

├── niamh\

│ ├── \_\_init\_\_.py

│ ├── niamh.py # Niamh-specific logic (integrates Fact-Checker)

│

├── core\

│ ├── \_\_init\_\_.py

│ ├── llm\_handler.py # Handles Llama 3.1 API calls (Ollama)

│ ├── database.py # MongoDB & Redis integration

│ ├── tools.py # ✅ (NEW) Shared tools (web scraper, API fetch, etc.)

│

├── agents\

│ ├── \_\_init\_\_.py

│ ├── fact\_checker.py # Fact-checks Niamh's responses

│

└── tests\

├── test\_fact\_checker.py