📌 Final Indexed List of Files in JC1-Inference

This is the **most accurate and structured index** of all files in your **JC1-Inference** project.

📂 1. Application Code (app/)

The **core application directory** responsible for API endpoints, model loading, utilities, and core logic.

📂 1.1 API Endpoints (app/api/)

- [`chat.py`] → Handles chat-based LLM API.
- [`vision.py`] → Processes image-based inputs (multimodal support).
- [`speech.py`] → Converts speech-to-text (ASR model integration).
- [`tools.py`] → External tool integrations (search, plugins, RAG functions).
- [`memory.py`] → Context memory retrieval (longer conversations).
- [`__init__.py`] → Package initializer for the API module.

📂 1.2 Model Inference (app/models/)

- [`loader.py`] → Loads models using DeepSpeed & vLLM (GPU acceleration).
- [`tokenizer.py`] → Tokenization utilities (BPE, WordPiece, SentencePiece).
- [`inference.py`] → Manages inference execution (batched and optimized).
- [`__init__.py`] → Package initializer for the model module.

📂 1.3 Utility Functions (`app/utils/`)

- [`config.py`] → Stores app configurations (paths, API keys, settings).
- [`logger.py`] → Implements logging for debugging and monitoring.
- [`security.py`] → Handles security features (encryption, access control).
- [`memory.py`] → Manages memory with a vector database (retrieval-augmented generation).
- [`__init__.py`] → Package initializer for the utilities module.

📂 1.4 Core Processing (`app/core/`)

- [`retriever.py`] → Implements retrieval-augmented generation (RAG).
- [`cache.py`] → Key-value cache and session handling.
- [`executor.py`] → Executes Python code dynamically for tool use.
- [`__init__.py`] → Package initializer for the core module.

📂 1.5 Main Application File

- [`main.py`] → FastAPI entry point (exposes the REST API).

📂 2. Configuration Files (`configs/`)

- [`settings.yaml`] → Defines settings (model paths, API keys).
- [`logging.yaml`] → Logging configurations.
- [`security.yaml`] → Access control policies.

📂 3. Deployment (`deployment/`)

📂 3.1 Docker Setup (`deployment/docker/`)

- [`Dockerfile`] → Defines containerized build.
- [`docker-compose.yaml`] → Sets up multi-container architecture.

📂 3.2 Kubernetes Setup (`deployment/k8s/`)

- [`deployment.yaml`] → Kubernetes deployment configuration.
- [`service.yaml`] → Kubernetes service for exposing the API.

📂 3.3 Inference Server Setup (`deployment/inference-server/`)

- [`triton-config.yaml`] → Configures NVIDIA Triton Inference Server.

📂 3.4 Miscellaneous

- [`setup.sh`] → Shell script to set up the environment.

📂 4. Testing Suite (`tests/`)

- [`unit/`] → Unit tests.
- [`integration/`] → Integration tests.
- [`performance/`] → Performance benchmarking.

📂 5. Data Storage (`data/`)

- [`embeddings/`] → Stores vectorized memory embeddings.
- [`cache/`] → Temporary session caches.

📂 6. Automation Scripts (`scripts/`)

- [`train_model.py`] → Fine-tunes the model using RLHF.
- [`inference_benchmark.py`] → Measures inference speed & performance.

📂 7. Documentation (`docs/`)

- [`api-docs.md`] → API documentation.

- [`setup-guide.md`] → Installation guide.
- [`architecture.md`] → System architecture overview.

📂 8. Root Files

- [`requirements.txt`] → Python dependencies.
- [`README.md`] → Project overview.