**CLASS LanGaimaCyberFortSentinel:**

// Constructor to initialize all layers and their corresponding data

**FUNCTION \_\_init\_\_():**

*Initialize all layer data (Layer-1 to Layer-7)*

*Initialize the update flags for each layer*

// \*\*\*\*\*\* Layer-1: GenFusion Data Repository \*\*\*\*\*\*

**FUNCTION load\_genfusion\_data\_repository():**

*// Load data from public and private datasets*

*public\_data = load\_public\_datasets()*

*private\_data = load\_private\_datasets()*

*return public\_data + private\_data*

**FUNCTION load\_public\_datasets():**

*// Load public datasets*

*RETURN ["Public Dataset 1", "Public Dataset 2", "Public Dataset 3"]*

**FUNCTION load\_private\_datasets():**

*// Load private generated datasets*

*RETURN ["Private Dataset 1", "Private Dataset 2", "Private Dataset 3"]*

// \*\*\*\*\*\* Layer-2: SynerTrain Pipeline Augmentation \*\*\*\*\*\*

**FUNCTION synertrain\_pipeline\_function(data):**

*// Perform model training with public and private datasets*

*models = train\_ensemble\_models(data)*

*augmented\_results = augment\_results(models)*

*RETURN augmented\_results*

**FUNCTION train\_ensemble\_models(data):**

*// Train different ensemble models with the data*

*RETURN ["Ensemble Model 1", "Ensemble Model 2", "Ensemble Model 3"]*

**FUNCTION augment\_results(models):**

*// Perform augmentation tasks like tokenization*

*RETURN {*

*"tokenization": "Tokenization results",*

*"sentiment\_analysis": "Sentiment Analysis results",*

*"contextual\_features": "Contextual Features",*

*"content\_recognition": "Content Recognition results",*

*"feature\_integration": "Integrated features"*

*}*

**// \*\*\*\*\*\* Layer-3: LangAima IntraChain Framework \*\*\*\*\*\***

**FUNCTION langaima\_intrachain\_function(augmented\_results):**

*// Process the augmented results through the LangAima IntraChain Framework*

*external\_integration = integrate\_external\_components(augmented\_results)*

*RETURN "Processed Data from LangAima IntraChain"*

**FUNCTION integrate\_external\_components(augmented\_results):**

*// Simulate the integration of external systems and tools*

*RETURN "External Components Integrated"*

**//** \*\*\*\*\*\* **Layer-4: ChainTrust Content Validator** \*\*\*\*\*\*

**FUNCTION chaintrust\_validator\_function(processed\_data):**

*// Validate the content's authenticity and integrity*

*validated\_data = validate\_content(processed\_data)*

*RETURN validated\_data*

**FUNCTION validate\_content(processed\_data):**

*// Simulate content validation*

*RETURN "Validated Content"*

// \*\*\*\*\*\*Layer-5: LanGaima Swarm Intelligence Model \*\*\*\*\*\*

**FUNCTION swarm\_intelligence\_model\_function(validated\_data):**

*// Create decentralized communities for threat detection*

*communities = create\_communities(validated\_data)*

*RETURN "Threats Detected by Communities"*

**FUNCTION create\_communities(validated\_data):**

*// Simulate the creation of communities based on the data*

*RETURN "Communities Created"*

**//** \*\*\*\*\*\* **Layer-6: Classification Model** \*\*\*\*\*\*

**FUNCTION classification\_model\_function(swarm\_results):**

*// Classify events, activities, and actions using advanced AI models*

*classification\_results = classify\_events(swarm\_results)*

*RETURN classification\_results*

**FUNCTION classify\_events(swarm\_results):**

*// Perform classification of events, activities, and actions*

*RETURN {*

*"event\_classification": "Event Classified",*

*"activity\_classification": "Activity Classified",*

*"action\_classification": "Action Classified"*

*}*

// \*\*\*\*\*\* Layer-7: Fine-Tuning \*\*\*\*\*\*

**FUNCTION fine\_tuning\_model\_function(classification\_results):**

*// Fine-tune the model based on the classification results*

*fine\_tuned\_model = fine\_tune\_model(classification\_results)*

*RETURN fine\_tuned\_model*

**FUNCTION fine\_tune\_model(classification\_results):**

*// Simulate model fine-tuning*

*RETURN "Fine-Tuned Model Ready"*

// \*\*\*\*\*\* Main Process \*\*\*\*\*\*

**FUNCTION main\_process():**

*// Step 1: Data Curation (Layer-1)*

layer\_1\_data = load\_genfusion\_data\_repository()

*// Step 2: SynerTrain Pipeline Augmentation (Layer-2)*

augmented\_results = synertrain\_pipeline\_function(layer\_1\_data)

*// Step 3: LangAima IntraChain Framework (Layer-3)*

layer\_3\_data = langaima\_intrachain\_function(augmented\_results)

*// Step 4: Content Validation (Layer-4)*

validated\_data = chaintrust\_validator\_function(layer\_3\_data)

*// Step 5: Swarm Intelligence for Threat Detection (Layer-5)*

swarm\_results = swarm\_intelligence\_model\_function(validated\_data)

*// Step 6: Classification (Layer-6)*

classification\_results = classification\_model\_function(swarm\_results)

*// Step 7: Fine-Tuning the Model (Layer-7)*

fine\_tuned\_model = fine\_tuning\_model\_function(classification\_results)

*// Return the fine-tuned model*

RETURN fine\_tuned\_model

// \*\*\*\*\*\* Update Logic \*\*\*\*\*\*

FUNCTION update\_layers(update\_layer):

IF update\_layer == 3:

*// Update Layer-3 and Layer-2*

*update\_layer\_2()*

CONTINUE *process at Layer-3*

IF update\_layer == 5:

*// Update Layer-5, Layer-2, and go to Layer-6*

*update\_layer\_2()*

*update\_layer\_5()*

CONTINUE *process at Layer-6*

IF update\_layer == 6:

*// Update Layer-6, Layer-5, and Layer-2, then go to Layer-7*

*update\_layer\_2()*

*update\_layer\_5()*

*update\_layer\_6()*

CONTINUE *process at Layer-7*

FUNCTION update\_layer\_2():

*// Update the SynerTrain Pipeline (Layer-2) logic*

FUNCTION update\_layer\_5():

*// Update the Swarm Intelligence Model (Layer-5) logic*

FUNCTION update\_layer\_6():

*// Update the Classification Model (Layer-6) logic*