Playbook for Open-Source v2.2.0

# Contents

**Introduction ....................................................................................................................................... 4**

**Infosys Responsible AI Office .............................................................................................................. 4**

**Infosys Responsible AI toolkit ............................................................................................................. 5**

**FM - Moderation Layer ....................................................................................................................... 5**

Introduction ........................................................................................................................................... 5

Components of Input ............................................................................................................................. 5

Prompt and Temperature Parameters .................................................................................................... 5

Multi-Lingual Support .......................................................................................................................... 5

Emoji Moderation ................................................................................................................................ 6

Prompt Template ................................................................................................................................. 6

Components of Output ....................................................................................................................... 7

Template-based guardrails .................................................................................................................. 7

*Request Moderation* ............................................................................................................................ 7

*Response Moderation* ......................................................................................................................... 9

*Response Comparison* ....................................................................................................................... 10

Model-based guardrails: ................................................................................................................... 10

*Request Moderation* .......................................................................................................................... 10

*Response Moderation* ........................................................................................................................ 11

*Response Comparison* ........................................................................................................................13

**Endpoints and Functionalities .............................................................................................................14**

**Explainability ...................................................................................................................................... 16**

Introduction ........................................................................................................................................... 16

Explainability Features........................................................................................................................... 16

Global Interpretability ........................................................................................................................... 16

Local Interpretability ............................................................................................................................. 16

SHAP (SHapley Additive exPlanations) ...................................................................................................16

LIME (Local Interpretable Model-Agnostic Explanations) ..................................................................... 16

Anchor Tabular ...................................................................................................................................... 17

Self-reasoning techniques ..................................................................................................................... 17

Chain of Thought (CoT) .......................................................................................................................... 17

Thread-of-Thought (ThoT) ..................................................................................................................... 17

Graph-of-Thought (GoT) .........................................................................................................................17 Chain-of-Verification (CoV) .................................................................................................................... 17

Attention Visualization using Token Importance Charts ........................................................................ 18

**Endpoints and Functionalities ............................................................................................................. 18**

**Fairness & Bias ................................................................................................................................... 20**

Introduction ........................................................................................................................................... 20

Fairness & Bias Components ................................................................................................................. 20

Text: [Evaluate texts generated by LLM] ................................................................................................ 20

Image: [Evaluate images generated by LLM] ..................................................................................... 20

Fairness & Bias Evaluations for Traditional Models [Structured Data] .............................................. 20

**Endpoints and Functionalities ............................................................................................................. 21**

**Hallucination ...................................................................................................................................... 22**

Introduction ........................................................................................................................................... 22

Components of Hallucination ................................................................................................................ 22

Thread of Thoughts (ThoT) .................................................................................................................... 22

G-Eval Metrics ........................................................................................................................................ 22

Chain of Verification (CoVe) ................................................................................................................... 23

Chain of Thoughts (CoT) ........................................................................................................................ 23

Chain of Thoughts (CoT) ........................................................................................................................ 23

Logic of Thoughts (LoT) .............................................................................................................23

Hallucination Score ................................................................................................................................ 23

**Endpoints and Functionalities ............................................................................................................. 23**

**Privacy and Safety .............................................................................................................................. 24**

Introduction ........................................................................................................................................... 24

Components of Privacy and Safety ........................................................................................................ 24

PII Detection: ......................................................................................................................................... 24

Anonymization: ...................................................................................................................................... 25

Data Protection: ..................................................................................................................................... 25

Enhanced Privacy: .................................................................................................................................. 25

Confidentiality Preservation: ................................................................................................................. 25

**Endpoints and Functionalities ............................................................................................................ 25**

**Security .............................................................................................................................................. 26**

Introduction ........................................................................................................................................... 26

Features of a Security Module ............................................................................................................... 26

Component of Security .......................................................................................................................... 26 Bulk Vulnerability Assessment ............................................................................................................... 26

**Endpoints and Functionalities ............................................................................................................. 27**

**Responsible AI toolkit UI .................................................................................................................... 27**

Introduction ............................................................................................................................................ 27

Features of a UI ....................................................................................................................................... 28

**Endpoints and Functionalities ............................................................................................................. 29**

**Red Teaming....................................................................................................................................... 31**

Introduction.............................................................................................................................31

Types of Red Teaming..............................................................................................................31

Automated Red Teaming.........................................................................................................31

Manual Red Teaming...............................................................................................................31

PAIR Technique (Prompt Adversarial Iterative Refinement) ....................................................32

TAP Technique (Tree of Attacks with Pruning) ........................................................................32

Evaluator Models ....................................................................................................................32

Supported Models & API End Points ...................................................................................... 33

Report Management .............................................................................................................. 33

Endpoints & Functionalities ....................................................................................................34

**Contact** .........................................................................................................................................35

# Introduction

Responsible AI, also known as Ethical AI, is a fundamental concept in the field of artificial intelligence that emphasizes the importance of using AI in a manner that respects human well-being and ethical principles. It seeks to ensure that AI is developed and used thoughtfully and considerately, minimizing potential negative impacts on individuals and society. This involves designing and implementing AI technologies in a way that aligns with societal values, upholds human rights, and avoids unintended biases and discriminatory effects. Responsible AI principles serve as guidelines or frameworks to ensure the ethical and responsible use of AI.

To understand the importance of responsible AI principles, let's consider the example of facial recognition technology. While it has gained popularity, it has also raised concerns about privacy and potential misuse. If a company decides to implement a facial recognition system without proper safeguards or oversight, it could lead to unauthorized surveillance, profiling of individuals, privacy rights violations, potential discrimination, or false identification leading to wrongful arrests. By adhering to responsible AI principles, such as incorporating transparency, consent, and accountability measures, companies can mitigate these risks and ensure that facial recognition technology is deployed in a responsible and ethical manner.

Infosys Responsible AI Office

Infosys is working on a comprehensive framework, through its Responsible AI Office, to guide businesses in implementing Responsible AI. This framework is structured into three main components: Scan, Shield, and Steer.

**Scan:** This component focuses on monitoring and assessing compliance and risks. It includes tools like the Infosys Responsible AI Watchtower for monitoring external regulations, the Infosys Responsible AI Maturity Assessment and Audits for assessing compliance readiness, and the Infosys Responsible AI Telemetry for internal compliance monitoring.

**Shield:** The Shield component provides technical solutions to protect AI models and systems from various risks. It includes the Infosys Responsible AI Toolkit, which offers a range of solutions to safeguard AI systems, Infosys Gen AI Guardrails for moderating generative AI systems, and the Infosys Responsible AI Gateway to enforce responsible AI protocols throughout the AI lifecycle.

**Steer:** This part of the framework focuses on governance, legal consultation, and strategy formulation. It involves managing a dedicated Responsible AI practice, legal reviews of AI contracts with vendors, strategy development, standardized audits, and industry certifications.

# Infosys Responsible AI toolkit

The Infosys Responsible AI Toolkit, a technical offering within the Shield component of the 3S framework, provides solutions for detecting and addressing risks to ensure responsible

AI.

The Infosys Responsible AI toolkit features a suite of APIs, known as a FM-Moderation Layer, to safeguard AI models by verifying inputs and outputs for safety, security, privacy, and fairness. Additionally, These APIs assist in providing explanations for the outcomes of various traditional and generative AI models, ensuring transparency in their decisionmaking processes.

The moderation layer is a modular component that can be seamlessly integrated into clients’ AI systems. It enables integration with custom traditional and large language models, implementation of security measures, configuration of moderation checks, and ongoing monitoring of AI models to maintain ethical, legal, and compliance standards.

# FM - Moderation Layer

## Introduction

The Filtering and Moderation module (called as FM-Moderation layer) is a crucial part in content moderation systems that ensure the safety, compliance, and appropriateness of responses generated by AI models like GPT3.5, GPT4, Gemini, Llama and AWS Anthropic. It applies a set of checks and filters to both the input prompt and the generated response to keep control over the content generated by the language model. The module consists of multiple layers, including Request Moderation, Response Moderation, and Response Comparison, each serving a specific purpose in the moderation process.

## Components of Input

Prompt and Temperature Parameters

The FM Moderation module requires two input parameters: Prompt and Temperature. The Prompt is the first text provided to the model to generate a response, while the Temperature parameter controls the creativity of the response. A lower temperature value produces more precise and deterministic outputs, while a higher value introduces more creative elements into the response.

Multi-Lingual Support

Multilingual support in FM moderation entails the ability to process prompts and content in various languages. When a prompt is provided in any language, the system automatically detects the language and translates it to English for further processing. This functionality allows users to select options such as Google, Azure or none, while ensuring seamless backend operations with language detection and translation mechanisms in place. This approach enables efficient and effective moderation across different languages, using automated processes to manage content consistently and accurately.

Emoji Moderation

In FM moderation, default emoji support allows users to include emojis to convey sentiments or expressions within sentences. Behind the scenes, the system detects these emojis and seamlessly integrates them into the moderation process. This functionality ensures that emojis are recognized and interpreted as part of the input, enabling the moderation system to effectively consider them when generating results or making decisions. This capability significantly enhances the system's ability to understand and respond to content having emojis, thereby improving the accuracy and relevance of moderation outcomes.

Prompt Template

### Global Priority(Safety Focus Template):

### This Defense Template enables LLM to put Safety over helpfulness in its response, safeguarding the model from Unsafe and harmful prompts.

### Self-Reminder (Ethics Based Template)

### This defense is a protective mechanism that helps LLM resist harmful prompts and generate responsible answers by reminding itself to adhere to ethical guidelines.

### Cove Complexity

Cove Complexity can be considered a synonymous term for The Chain of Verification. It is a rigorous evaluation method for LLMs that subjects it to generate responses to a series of increasingly complex logical questions. By tailoring the query difficulty to the selected level (simple, medium, or complex), this process effectively probes the LLM's ability to provide right, consistent, and logically sound information. Essentially, it acts as a quality control checkpoint, identifying potential weaknesses in the model's reasoning capabilities and factual knowledge.

LLM Model

We have provided support to various LLM models to generate response and perform template-based checks, these include: gpt-4o-mini, gpt-35-turbo, Llama3-70b, anthropic-claude-3-sonnet, gemini-2.5-pro and gemini-2.5-flash models.

### Components of Output

In FM moderation, two types of outputs are distinguished: Output from template-based guardrails and Output from model-based guardrails.

Template-based guardrails

Here we make use of dynamic and efficient prompt templates through Prompt Engineering that enhance the detection capability of the LLMs to detect and block adversarial attacks.

#### *Request Moderation*

In the Request Moderation layer, various checks are performed on the input prompt before generating a response. These checks include Prompt Injection, Jailbreak, Fairness and Bias, language critique coherence, language critique fluency, language critique grammar, language critique politeness, evaluator check, context relevance, context conciseness and context reranking. These checks ensure that the input prompt adheres to the defined guidelines and standards.

#### *Prompt Injection Check*

This check evaluates the presence of injected content in the input prompt. It calculates an injection confidence score based on the prompt and compares it against a dynamic injection threshold. The result writes down whether the check passes or fails.

#### Jailbreak Check

The Jailbreak check examines the input prompt for potential attempts to manipulate or bypass the moderation system. It calculates a jailbreak similarity score and compares it against a dynamic jailbreak threshold. The result says whether the check passes or fails.

#### Fairness and Bias

In the context of request moderation, fairness and bias analysis aims to show potential discriminatory patterns or biases in the system's responses. This analysis typically examines various aspects of the moderation process.

#### Language Critique Coherence

An evaluation of the logical connection and consistency between distinct parts of an explanation. It figures out how well the ideas are organized and linked together to form a cohesive narrative.

#### Language Critique Fluency

A measure of the smoothness and naturalness of the language used in an explanation. It assesses the flow of ideas and the overall readability of the text.

#### Language Critique Grammar

An assessment of the grammatical correctness and adherence to language conventions in an explanation. It evaluates the syntax, punctuation, and overall linguistic accuracy of the text.

#### Language Critique Politeness

A measure of the appropriateness and respectfulness of the language used in an explanation. It assesses the tone and choice of words to figure out if the explanation is polite and considerate of the audience.

#### *Infosys Advanced Jailbreak Check & Infosys Random Noise Check*

Infosys Advanced Jailbreak Check help to check and identify any jailbreak attacks made on the toolkit.

#### *Context Relevance*

Evaluates how relevant the retrieved context is to the specified question. Context relevance score measures if the retrieved context has enough information to answer the question being asked. This check is important since bad context reduces the chances of the model giving a relevant response to the question asked, as well as leading to hallucinations.

#### *Context Conciseness*

Evaluates the concise context cited from an original context for irrelevant information. Context conciseness refers to the quality of a reference context generated from retrieved context in terms of being clear, brief, and to the point. A concise context effectively conveys the necessary information without unnecessary elaboration or verbosity.

*Response Moderation*

In the Response Moderation layer, the generated response from the Request Moderation layer is further evaluated before being presented to the user. This evaluation includes checks for language critique coherence, language critique fluency, language critique grammar, language critique politeness.

##### Response Completeness

Check whether the response has answered all the aspects of the question specified. Response completeness score measures if the generated response has adequately answered all aspects to the question being asked. This check is important to ensure that the model is not generating incomplete responses.

##### Response Conciseness

Grades show how concise the generated response is or if it has any other irrelevant information for the question asked. Response conciseness measures whether the generated response holds any other information irrelevant to the question asked.

##### Response Validity

Check if the response generated is valid or not. A response is valid if it holds any information. In some cases, an LLM might not generate a response due to reasons like limited knowledge or the asked question not being clear. Response Validity score can be used to name these cases, where a model is not generating an informative response.

##### Response Completeness with respect to Context

Response completeness with respect to context in response moderation refers to the ability of a moderation system to accurately assess and address the content of a response based on its surrounding context. This involves understanding the nuances of the conversation, the intent of the user, and the potential impact of the response.

Model-based Guardrails

In Model-based Guardrails we use Pre-trained models trained on extensive billion-parameter datasets to effectively combat adversarial attacks. These models are equipped with sophisticated algorithms capable of discerning and mitigating malicious content by employing various metrics such as scores and thresholds. These metrics are meticulously set based on detection entities or through comparative analysis of text embeddings, ensuring robust detection and response mechanisms against adversarial threats. This approach enables our system to support ambitious standards of security and reliability, safeguarding against potential risks and ensuring the integrity of moderated content.

##### Request Moderation

In the Request Moderation layer, various checks are performed on the input prompt before generating a response. These checks include Prompt Injection, Jailbreak, Privacy, Profanity, Toxicity, and Restricted Topic checks. These checks ensure that the input prompt adheres to the defined guidelines and standards.

##### Prompt Injection Check

This check evaluates the presence of injected content in the input prompt. It calculates an injection confidence score based on the prompt and compares it against a dynamic injection threshold. The result writes down whether the check passes or fails.

##### Jailbreak Check

The Jailbreak check examines the input prompt for potential attempts to manipulate or bypass the moderation system. It calculates a jailbreak similarity score and compares it against a dynamic jailbreak threshold. The result writes down whether the check passes or fails.

##### Privacy Check

The Privacy check allows configuration of specific entities that should be recognized and protected within the prompt, such as Aadhar numbers, passport details, or PAN numbers. It names the recognized entities and compares them against the configured entities to block. The result writes down whether the check passes or fails.

##### Profanity Check

The Profanity check names profane words within the input prompt. It reports the profane words named and compares them against a customized threshold. If the number of occurrences of profane words is greater than the threshold, it falls to failed category.

##### Toxicity Check

The Toxicity check assesses the level of toxicity in the input prompt based on metrics such as toxicity, severe toxicity, obscenity, identity attack, insult, threat, and sexual explicitness. It compares the calculated toxicity scores against a predefined toxicity threshold and at last we can assess whether response passes or fails.

##### Restricted Topic Check

The Restricted Topic check ensures that certain predefined topics, such as explosives, terrorism, or political subjects, are not included in the input prompt. We can configure what topics need to be restricted.

##### Response Moderation

In the Response Moderation layer, the generated response from the Request Moderation layer is further evaluated before being presented to the user. This evaluation includes checks for Text Quality, Text Relevance, and Refusal. All the checks which are there in Request Moderation layer are same with response Moderation also except jailbreak check and Prompt injection check since the response from the LLM may not hold these attacks. Other than it has Privacy Check, Profanity Check, Toxicity Check, Restricted Topic Check.

##### Privacy Check

The Privacy check allows configuration of specific entities that should be recognized and protected within the prompt, such as Aadhar numbers, passport details, or PAN numbers. It finds the recognized entities and compares them against the configured entities to block. The result shows whether the check passes or fails.

##### Restricted Topic Check

The Restricted Topic check ensures that certain predefined topics, such as explosives, terrorism, or political subjects, are not included in the input prompt. We can configure what topics need to be restricted.

##### Toxicity Check

The Toxicity check assesses the level of toxicity in the input prompt based on metrics such as toxicity, severe toxicity, obscenity, identity attack, insult, threat, and sexual explicitness. It compares the calculated toxicity scores against a predefined toxicity threshold and at last we can assess whether response passes or fails.

##### Profanity Check

The Profanity check finds profane words within the input prompt. It reports the profane words shown and compares them against a customized threshold. If the number of occurrences of profane words is greater than the threshold, it falls to the failed category.

##### Text Quality

The Text Quality check assesses the readability and grade level of the generated response. The score stands for the grade level using this scale below:

|  |  |  |
| --- | --- | --- |
| **Score** | **School level** | **Remarks** |
| ≥ 90 | 5th Grade | It's quite easy to read. Easily understood by an average 11-year-old student. |
| 80 – 89 | 6th Grade | Easy to read. Conversational English for consumers. |
| 70 – 79 | 7th Grade | Fairly easy to read. |
| 60 – 69 | 9th Grade | Plain English. Easily understood by 13- to 15-year-old students. |
| 50 – 59 | 11th Grade | It is fairly difficult to read. |
| 30 – 49 | College Level | Difficult to read. |
| < 30 | College Graduate | Exceedingly difficult to read. Best understood by university graduates. |
| *(Default)* | 4th Grade or Below | Default grade, very easy to understand. |

##### Text Relevance Check

The Text Relevance check measures the relevance of the generated response with respect to the input prompt.

##### Refusal Check

The Refusal check finds cases where the language model refuses to provide a response due to content that violates norms or guidelines.

#### *Additional Optional Checks added for Request Moderation and Response Moderation under Model-based Guardrails*

##### Invisible Text Check

It is a package-based evaluation check designed to detect and remove non-printable, invisible Unicode characters from text inputs.

##### Gibberish Check

It is a model-based check designed to identify and filter out gibberish or nonsensical inputs in English language text.

##### Ban Code Check

It is a model-based check designed to detect code snippet in the prompt.

##### Sentiment Check

It is a package-based check with the primary objective to gauge the sentiment of a given prompt.

##### Response Comparison

The Response Comparison element compares the response generated from the FM Moderation module with a reference response directly from the LLM. This comparison helps ensure consistency and evaluate the effectiveness of the moderation process.

The FM Moderation module ensures that the generated responses are compliant, safe, and aligned with defined guidelines. It provides granular control over the content generated by AI language models, reducing risks associated with inappropriate or harmful outputs.

**Response with Infosys RAI guardrails**

It is the response from the LLM after the prompt as well as the response has passed through various checks at request and response level of the guardrails.

**Result with LLM**

It is the response directly from the LLM without passing through the guardrails.

## Endpoints and Functionalities

The following table lists the available Moderation Layer endpoints and their functionalities. Refer to the API documentation: [Moderation Layer Models](https://github.com/Infosys/Infosys-Responsible-AI-Toolkit/blob/main/responsible-ai-moderationLayer/Docs/MODERATION_LAYER_endpoints_instructions.pdf)

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /rai/v1/moderations | Decoupled guardrail provides checks for the prompt like privacy check, prompt injection check, jailbreak check, toxicity check, restricted topic, custom theme check. |
| /rai/v1/moderations/coupledmoderations | Coupled guardrail provides checks for input prompt, LLM interaction for generating response and checks for response. |
| /rai/v1/moderations/getTemplates/ | To retrieve and store all the custom templates. |
| /rai/v1/moderations/clearCache | To clear the cache of prompts and responses. |
| /rai/v1/moderations/evalLLM | Provides Template based guardrails to check prompts for prompt injection, jailbreak, language coherence etc. using LLM as evaluator. |
| /rai/v1/moderations/multimodel | Provides checks for multimodal prompts like prompt injection, jailbreak, etc. |
| /rai/v1/moderations/recommend | Recommends prompts for each check for the user. |
| /rai/v1/moderations/translate | To convert text in any language to English using Google Translate or Azure Translation. |
| /rai/v1/moderations/openai | Provides Open AI completion Response for a prompt. |
| /rai/v1/moderations/openaiCOT | Provides the Chain of Thought that LLM went through to provide the response to the prompt. |
| /rai/v1/moderations/healthcareopenaiCOT | Provides the chain of thoughts the LLM went through to provide response to the prompt, adding in example prompt response to tell the LLM which details to be included in the response and what format the response should be in. |
| /rai/v1/moderations/openaiTHOT | Provides Thread of Thought ie. how LLM breaks down the prompt to correctly understand it and generate response. |
| /rai/v1/moderations/ModerationTime | Provides the time taken by all Moderation Checks. |
| /rai/v1/moderations/setTelemetry | To Check Telemetry Status. |
| /rai/v1/moderations/ToxicityPopup | To check Toxicity of the prompt. |
| /rai/v1/moderations/ProfanityPopup | To Check if the prompt contains profane words. |
| /rai/v1/moderations/PrivacyPopup | To check whether the prompt contains any PII entities. |
| /rai/v1/moderations/COV | Provides the Chain Of Verifications - Questions the LLM asked itself and answered to reach the final response. |
| /rai/v1/moderations/OrgPolicy | To check if prompt is associated with any Restricted Topic. |
| /rai/v1/moderations/gEval | Provides Faithfullness Check and scores for how summary is related to the text. |
| /rai/v1/moderations/Hallucination\_Check | To check whether the prompt is related to the sources provided. |

# Explainability

## Introduction

Explainability in AI, including Machine Learning and Large Language Models, is about understanding the logic behind a model's decisions. By revealing the reasoning process, it builds trust and improves model performance. It helps us spot biases, fix mistakes, and align models with our goals. Ultimately, explainability makes AI more reliable and trustworthy. *Infosys Responsible AI* toolkit provides a range of explainability techniques to enhance transparency and understanding of AI model decisions Explainability Features

Explainability aims at understanding how AI models arrive at its predictions. It involves breaking down the complex decision-making process of the model into human-understandable terms.

Feature Importance is a critical component of Model Interpretability, used to understand the relative importance of different features (variables) an AI model prediction. By quantifying the contribution of each feature, it helps to explain the model's decision-making process.

AI models require explanations of key features at both the overall model level and the individual instance level, termed global and local interpretability. Brief descriptions of terms related to model interpretability are outlined below

Global Interpretability

Understanding the overall behavior and decision-making process of a model across its entire input space. It provides insights into the general patterns and trends that the model has learned from the data.

Local Interpretability

The ability to understand how a machine learning model arrives at its predictions for specific instances. Unlike global interpretability, which focuses on understanding the overall behavior of the model, local interpretability provides explanations tailored to individual predictions.

SHAP (SHapley Additive exPlanations)

a game-theoretic method for explaining the output of any machine learning model. It calculates the contribution of each feature to a prediction, providing a more comprehensive understanding of the model's decision-making process.

Purpose: Feature ranking, Model understanding, Bias detection, Debugging

Applicability: SHAP is a model-agnostic method and applied to wide range of machine learning models

(e.g., Decision trees, Random forests, Gradient boosting machines, Neural networks, Support vector machines)

LIME (Local Interpretable Model-Agnostic Explanations)

a technique used in machine learning to explain the predictions of complex models in a locally interpretable way. It works by approximating a complex model with a simpler, linear model around a specific prediction.

Purpose: Local interpretability, Model agnosticism, Feature importance, Bias detection, Debugging

Applicability: LIME is a model-agnostic method and applied to wide range of machine learning models

(e.g., Decision trees, Random forests, Gradient boosting machines, Neural networks, Support vector machines)

Anchor Tabular

a technique used in explainability to identify a minimal set of conditions (anchors) that are sufficient to explain a prediction made by a machine learning model. These anchors are humanreadable rules that capture the essence of the model's decision-making process for a specific instance.

Purpose: Local interpretability, Simplicity, Feature importance, Bias detection, Debugging

Applicability: Anchor Tabular is primarily designed for tabular data, but it can also be applied to other types of data with some modifications. It can be used to explain the predictions of various machine learning models (e.g., Decision trees, Random forests, Gradient boosting machines, Neural networks, Support vector machines)

Self-reasoning techniques

a set of techniques that enable LLMs to generate more comprehensive and coherent explanations for their outputs. These frameworks often involve breaking down complex problems into smaller, more manageable steps and guiding the LLM through a reasoning process. This technique relies heavily on prompt engineering, with thoughtfully crafted prompts leading to informative outcomes.

Few models built on self-reasoning framework are:

Chain of Thought (CoT)

Step-by-step reasoning technique prompts the LLM to break down a complex task into smaller, more manageable steps and explain its reasoning for each step. This allows users to follow the LLM's thought process and understand how it arrived at its conclusion.

Thread-of-Thought (ThoT)

Generate a series of interconnected thoughts that form a coherent narrative. This helps to visualize the LLM's reasoning process and identify any inconsistencies or biases.

Graph-of-Thought (GoT) represents the LLM's reasoning process as a graph, where nodes represent intermediate thoughts and edges represent the connections between them. This visual representation can be helpful for understanding the LLM's decision-making process.

Chain-of-Verification (CoV) prompts the LLM to verify its responses against external knowledge sources. This helps to ensure the accuracy and reliability of the LLM's outputs.

Logic of Thought (LoT)

Logic-of-Thought (LoT) prompting which employs propositional logic to generate expanded logical information descriptions and utilizes them as an additional augmentation to original contexts, thereby ensuring information completeness and enhancing logical reasoning ability.

ReRead Reasoning (RE2)

RE2 shifts the focus to the input by processing questions twice, thereby enhancing the understanding process. Consequently, RE2 demonstrates strong generality and compatibility with most thought eliciting prompting methods, including CoT

Image Explanation

Image Explain offers detailed explanations for images generated by Large Language Models (LLMs). It provides an in-depth analysis of the image, highlighting key insights such as the presence of watermarks, detection of potential biases, and identification of the visual style used. In addition, the system evaluates the image using key metrics like creativity and certainty. When applicable, Image Explain also performs object detection, identifying and labeling objects within the image. It then provides an explanation based on the detected objects, This enhances the interpretability and trustworthiness of AI-generated visuals by combining both image-level and object-level understanding.

Attention Visualization using Token Importance Charts

In AI language models, the importance of tokens can significantly influence the generated responses. Understanding which tokens (words or phrases) are most impactful can be crucial for interpreting and trusting the model's decisions. This is particularly relevant in applications where precise and reliable outputs are essential, such as in healthcare, finance, and legal domains.

Endpoints and Functionalities



The following table lists the available Moderation Layer endpoints and their functionalities. Refer to the API documentations : [Traditional Models](https://github.com/Infosys/Infosys-Responsible-AI-Toolkit/blob/main/responsible-ai-moderationLayer/Docs/MODERATION_LAYER_endpoints_instructions.pdf) , [Gen AI Models](https://github.com/Infosys/Infosys-Responsible-AI-Toolkit/blob/main/responsible-ai-llm-explain/Docs/LLM_EXPLAIN_endpoints_instructions.pdf)

**Explainability for LLMs**

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /rai/v1/llm-explainability/sentimentanalysis | Sentiment Analysis based on input prompt and displays token importance charts |
| /rai/v1/llm-explainability/uncertainty | Generates uncertainty and coherence score for the given input |
| /rai/v1/llm-explainability/token-importance | Gives a table of important tokens and associated scores for the given input |
| /rai/v1/llm-explainability/got | Provides explainability based on Graph of Thoughts technique |
| /rai/v1/llm-explainability/serper\_response | Extracts facts from Google internet search based on given input text |
| /rai/v1/llm-explainability/cot | Provides explainability using Chain of Thoughts |
| /rai/v1/llm-explainability/thot | Provides explainability using Thread of Thoughts |
| /rai/v1/llm-explainability/COV | Provides explainability using Chain of Verification |
| /rai/v1/moderations/healthcareopenaiCOT | Get explainability using Chain of Thoughts in RAG scenario |
| /rai/v1/llm-explainability/reread\_reasoning | Logic of ReReading the prompt along with Thot to get better explanation |
| /rai/v1/llm-explainability/bulk\_processing | Allow user to upload a CSV or Excel file with multiple records to get the result for all the applicable Explainability techniques and respective tokenization cost |
| /rai/v1/image-explainability/analyze | This endpoint provides explanations for both uploaded and generated images, along with insights and evaluation metrics. It also offers explanations for queries related to image data |
| /rai/v1/image-explainability/objectdetection | Analyze object detection output in an image using the YOLO model. |
| ai/v1/image-explainability/objectdetection/yolo | This endpoint generates explanations for uploaded adversarial or non-adversarial image with bounding boxes |
| /rai/v1/llm/openai | To provide an explanation for the given input |
| /rai/v1/llm/image | To generate an image in base64 format for the given input Image |

**Explainability for traditional ML models**

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /rai/v1/explainability/methods/get | Returns list of explanation methods applicable for selected model and dataset |
| /v1/workbench/adddata,  /v1/workbench/data | To upload Data and retrieve dataid and name respectively |
| /v1/workbench/addmodel,  /v1/workbench/model | To upload Modle file and retrieve modelid and name respectively |
| /v1/workbench/addpreprocessor,  /v1/workbench/preprocessor | To upload encoder file and retrieve encoderid and name respectively |
| /v1/workbench/batchgeneration | Maps data, model and preprocessor to create a pipeline for explainability generation |
| /rai/v1/explainability/explanation/get | Get explanation for the given model and dataset |
| /rai/v1/explainability/report/generate | Generates a doc report for the given input of data, model, preprocessor for a specific batchid |
| /v1/report/downloadreport | To download the report generated against specific batchid |

# Fairness & Bias

## Introduction

The AI Fairness module is an essential component in the field of responsible AI and aims to address and mitigate biases in machine learning models, specifically it works with data sets. Biases can emerge from various factors like race, sex, religion, and other sensitive attributes present in the dataset. The goal of AI Fairness is to ensure that these biases do not influence the outcomes generated by the models, promoting fairness, and mitigating discriminatory effects.

The AI Fairness module helps identify and address biases in machine learning models, promoting fairness, and mitigating the impact of discriminatory outcomes. By incorporating fairness considerations, organizations can foster more equitable and unbiased decisionmaking processes.

### Fairness & Bias Components

Text: [Evaluate texts generated by LLM]

Given an un-structured text, a prompt template has been designed to check the text’s fairness and evaluate its bias indicator [High / Medium / Low / Neutral] using LLMs like GPT, Gemini, LLAMA, Mixtral, Claude etc. The prompt template also provides additional information like “Affected group“, indicating the group of people affected by the context of the sentence and also the type of Bias [Historical Bias, Confirmation Bias, etc.,] as well. We are working to extend this prompt template to generate the neutral versions of the given text .

We have added support for models provided by Azure, Google Cloud Platform (GCP) and Amazon Web Services (AWS).

Image: [Evaluate images generated by LLM]

“A picture can speak 1000 words. “

As good as this statement is, the context perceived about the picture just by looking at it can also differ from person to person. With this established, to see if a given picture / image is Fair or biased, we depend on the input prompt given to the LLM to generate this image. The input prompt given by the user sets the contextual expectation of the user and the picture / image generated can be validated with similar context. For the template-based approach, we are currently leveraging LLMs like GPT-4o, Gemini, Claude etc., for evaluation. We have plans to extend this to Gemini as well in the future.

We have added support for models provided by Azure, Google Cloud Platform (GCP) and Amazon Web Services (AWS).

Fairness & Bias Evaluations for Traditional Models [Structured Data]

Based on the selected sensitive / protected attribute for the given dataset, the positive / favorable outcome distribution is compared with the rest of the groups in the dataset and the metrics are calculated.

*i. Statistical Parity Difference:*

The Statistical parity difference metric calculates the difference in the ratio of favorable outcomes between privileged groups and un-privileged groups.

 *ii. Disparate Impact Ratio:*

The Disparate Impact Ratio metric calculates the ratio of favorable outcomes between privileged groups and un-privileged groups.

 *iii. Smooth Empirical Differential:*

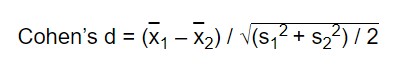
SED calculates the differential in the probability of favorable and unfavorable outcomes between intersecting groups divided by features. All intersecting groups are equal, so there are no unprivileged or privileged groups. The calculation produces a value between 0 and 1 that is the minimum ratio of Dirichlet smoothed probability for favorable and unfavorable outcomes between intersecting groups in the dataset.

1. *Four Fifths:*

This function computes the four fifths rule (ratio of success rates) between group\_unprivileged and group\_privileged. The minimum of the ratio taken both ways is returned. A value of 1 is desired. Values below 1 are unfair. The range (0.8,1) is considered acceptable.

1. *Cohen’s D:*

This function computes the Cohen D statistic (normalized statistical parity) between group\_unprivileged and group\_privileged.A value of 0 is desired. Negative values are unfair towards group\_unprivileged. Positive values are unfair towards group\_privileged. Reference values: 0.2 is considered a small effect size, 0.5 is considered medium, 0.8 is considered large.



#### *Mitigation of Bias - Reweighing*

To ensure fairness in AI, we can adjust the importance of different data points during training. Additionally, we can modify the decision-making process to ensure equal outcomes for different groups. By combining these techniques, we can create AI models that are both accurate and equitable. Following are some of the approaches to mitigate bias in AI models.

Reweighing is a preprocessing technique that Weights the examples in each (group, label) combination differently to ensure fairness before classification.

## Endpoints and Functionalities

The following table lists the available Fairness & Bias endpoints and their functionalities.

Refer to the API documentation : [Fairness & Bias Models](https://github.com/Infosys/Infosys-Responsible-AI-Toolkit/blob/main/responsible-ai-fairness/Docs/FAIRNESS_endpoints_instructions.pdf)

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /api/v1/fairness/Analyse | Detects bias in traditional ML models and datasets using statistical techniques like Statistical Parity Difference, Disparate Impact Ratio, Smooth Emperical Differential, Fourfifths and Cohen's D |
| /ap/v1/fairness/inprocessing/ exponentiated\_gradient\_reduction | Detects bias in traditional ML models using exponential gradient reduction method |
| /api/v1/fairness/individualMetrics | Analyze the pretrain data and post-train data [with model’s predictions] for group bias using metrics like Statistical parity |
| api/v1/fairness/pretrain/mitigation/getDataset | Generate metrics score for provided dataset |
| /api/v1/fairness/analysis/llm | Generates response of bias analysis from text prompt |
| /api/v1/fairness/analysis/image | Generates response of bias analysis from Image |

# Hallucination

## Introduction

Hallucinations in LLMs occur when the model's internal biases, learned patterns, or overgeneralizations lead to the creation of text that is not consistent with information. This can happen when the model struggles to distinguish between real and imagined information.

To address hallucinations in LLMs, it is essential to detect, quantify, and mitigate them. The Infosys Responsible AI toolkit offers features to achieve this. By employing techniques such as Chain of Thought, Thread of Thought, and Graph of Thought, potential hallucinations in diverse LLM outputs can be identified. Internet searches aid in verifying the accuracy of responses. G Eval metrics like adherence, faithfulness, and correctness assess the degree of hallucinations in both original and refined LLM outputs.

### Components of Hallucination

Thread of Thoughts (ThoT)

Thread of Thoughts (ThoT) is developed to address challenges in chaotic contexts, where LLMs struggle to sift through and prioritize relevant information amidst a plethora of data.

This is a novel prompting technique designed to enhance the reasoning capabilities of Large Language Models (LLMs) in handling chaotic contexts. The approach involves systematic segmentation, summarization, and analysis, which aligns with human cognitive patterns. It draws inspiration from human cognitive processes and aims to systematically segment and analyze extended contexts for better comprehension and accuracy.

G-Eval Metrics

The G-eval metrics are tools used to evaluate the performance and quality of texts generated by LLMs, taking into account the many specific nuances of these models. The GEval metrics leverage existing AI models and evaluate metrics like relevance, adherence, correctness, and faithfulness to assess the response generated by the LLM.

Chain of Verification (CoVe)

This measure is implemented directly to counteract hallucinations. As a reminder, hallucinations occur when the LLM responds incorrectly but in a logically coherent manner to a prompt.

Chain of Thoughts (CoT)

CoT refers to a structured approach in problem-solving, where complex tasks are broken down into a sequence of logical steps. It mirrors human reasoning by constructing a coherent argument from premises to a conclusion. In artificial intelligence, CoT prompting involves systematically reasoning through a problem rather than providing context-based responses. It enhances problem-solving capabilities.

CoT can be relevant to understanding hallucinations. CoT can help break down the phenomenon, considering factors like neural processes, sensory input, and cognitive biases. CoT ensures that AI-generated responses maintain coherence, avoiding hallucinatory or nonsensical outputs.

Chain of Thoughts (CoT)

CoT refers to a structured approach in problem-solving, where complex tasks are broken down into a sequence of logical steps. It mirrors human reasoning by constructing a coherent argument from premises to a conclusion. In artificial intelligence, CoT prompting involves systematically reasoning through a problem rather than providing context-based responses. It enhances problem-solving capabilities.

CoT can be relevant to understanding hallucinations. CoT can help break down the phenomenon, considering factors like neural processes, sensory input, and cognitive biases. CoT ensures that AI-generated responses maintain coherence, avoiding hallucinatory or nonsensical outputs.

Logic of Thoughts (LoT)

Logic-of-Thought (LoT) prompting employs propositional logic to generate expanded logical information from input context, and utilizes the generated logical information as an additional augmentation to the input prompts, thereby enhancing the capability of logical reasoning.

Hallucination Score

The hallucination score is determined through G- Eval metrics values and assessing similarity score across multiple categories, including input prompt to output prompt, input prompt to source, and output prompt to source. This score is a weighted average that considers the scores from these below categories.

## Endpoints and Functionalities

The following table lists the available Hallucination endpoints and their functionalities. Refer to the API documentation: [Hallucination Models](https://github.com/Infosys/Infosys-Responsible-AI-Toolkit/blob/main/responsible-ai-hallucination/Docs/HALLUCINATION_endpoints_Instructions.pdf)

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /rag/v1/FileUpload | It is used to upload document with pdf format and return vectorestoreid of vectorestore and blobname for the pdf. |
| /rag/v1/RetrievalKepler | This will take keys FileUpload, text, vectorestoreid as Input and will return a rag response along with the score. |
| /rag/v1/cov | It is used for getting the response with 5 more variants of questions generated by LLM to verify the answer and a refined final response |
| /rag/v1/cot | Provides the explanation steps and reasoning behind and from where (source referred which document) |
| /rag/v1/thot | Provides more descriptive and complex information that are spread over the file |
| /rag/v1/geval | Assess the alignment of the generated text with real-world information and can help detect hallucinations |
| /rag/v1/caching | This will take blobname which is generated while uploading file and return a array of length |
| /rag/v1/removeCache | It will take input as cache id and will remove the cached Vectorestore from the Cache |

# Privacy and Safety

## Introduction

In the era of growing concerns over data privacy and protection, the Privacy-Analyze and Privacy-Anonymize modules provide crucial functionalities to safeguard sensitive information within unstructured text data. These modules focus on identifying Personally Identifiable Information (PII) and anonymizing it to address privacy concerns effectively. By employing advanced algorithms and techniques, these modules enable users to protect personal data while retaining the contextual details of the text. AI technologies often collect and analyze large amounts of personal data, raising issues related to data privacy and security. To address these concerns, it is crucial to promote stringent data protection regulations and the adoption of secure data handling practices.

## Components of Privacy and Safety

PII Detection: The Privacy-Analyze module scans the input text to identify PII such as Aadhar numbers, PAN numbers, and confidential company data. It provides a comprehensive summary of the detected PII entities, along with corresponding metric scores. Higher scores indicate a higher likelihood of PII presence, empowering users to assess the sensitivity of the identified information.

Anonymization: The Privacy-Anonymize module replaces the identified PII entities with generic placeholders while preserving the overall structure and context of the text. Personal information such as names, locations, appointment dates, passport numbers, Aadhaar numbers, and PAN numbers are anonymized to protect individual privacy. This ensures that the text remains meaningful while safeguarding sensitive data.

Data Protection: The Privacy-Analyze module assists organizations and individuals in identifying and understanding potential privacy risks within unstructured text data. It helps to ensure compliance with privacy regulations and mitigate the risk of unauthorized data exposure.

Enhanced Privacy: The Privacy-Anonymize module enables the secure sharing and analysis of text data containing PII. It is particularly useful in scenarios where data needs to be anonymized for research, analysis, or collaboration purposes while maintaining privacy standards.

Confidentiality Preservation: Both modules aid in preserving the confidentiality of sensitive information in unstructured text data. This is crucial in industries such as healthcare, finance, and legal, where privacy and data protection are paramount.

# About Safety

AI models can sometimes generate harmful content, such as profanity, toxic language, explicit images, or sexually suggestive text. To ensure safe and responsible AI, it's crucial to implement measures that filter and prevent the generation of such content. This involves using advanced techniques to detect and mitigate harmful outputs, protecting users from exposure to inappropriate material, and maintaining a positive and inclusive online environment.

### **Components of Safety**

**Profanity Detection:**  
 The Safety module analyzes text input to detect profane, offensive, or toxic language. It provides detailed reports with toxicity scores, helping users identify and assess harmful content. This empowers organizations to moderate and maintain respectful communication.

**Content Censorship:**  
 The module can automatically mask or censor identified profane or inappropriate words in text outputs, ensuring that harmful language is effectively obscured while keeping the overall message intact.

**NSFW Image and Video Detection:**  
 The Safety module inspects images and videos to detect Not Safe For Work (NSFW) content, including pornography, sexually suggestive material, nudity, and related categories. It identifies specific parts of nudity within media files to help organizations filter and moderate visual content.

**Malicious URL Detection:**  
 By scanning URLs in input data, the Safety module detects potentially malicious or harmful links, helping to prevent phishing, malware distribution, and other cyber threats.

**CSV File Detection:**  
 The module supports analysis of CSV file contents for unsafe, toxic, or malicious material and removing or flagging such content. This facilitates large-scale data safety management efficiently.

**Enhanced User Protection:**  
 Together, these capabilities protect users from exposure to harmful, offensive, or inappropriate material across various content types, promoting a safer and more inclusive online environment.

## Endpoints and Functionalities

The following table lists the available Privacy & Safety endpoints and their functionalities. Refer to the API documentation: [Privacy Models,](https://github.com/Infosys/Infosys-Responsible-AI-Toolkit/blob/main/responsible-ai-privacy/Docs/PRIVACY_endpoints_instructions.pdf) [Safety Models](https://github.com/Infosys/Infosys-Responsible-AI-Toolkit/blob/main/responsible-ai-safety/Docs/SAFETY_endpoints_instructions.pdf)

**Privacy**

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /rai/v1/privacy/text/analyze | Checks if the input text contains any PII entities or not. |
| /rai/v1/privacy/text/anonymize | Anonymizes all PII entities in the input text. |
| /rai/v1/privacy/text/encrpyt | Encrypts all PII entities in the input text |
| /rai/v1/privacy/text/decrpyt | <to be updated> |
| /rai/v1/privacy/image/analyze | Analyzes the uploaded image for any PII entities |
| /rai/v1/privacy/image/anonymize | Anonymizes the PII entities present in the uploaded image |
| /rai/v1/privacy/image/hashify | Hashes the PII text present in an Image which is mapped to account. Gives back the Json which contains the hash value with a key that is mentioned on the bounding boxes in the response image. |
| /rai/v1/privacy/dicom/anonymize | Identifies and anonymizes the PII entities present in dicom images |
| /rai/v1/privacy/code/anonymize | Identifies and anonymizes the PII entities present in the code entered as text |
| /rai/v1/privacy/codefile/anonymize | Identifies and anonymizes the PII entities present in the code file that we uploaded as input |
| /rai/v1/privacy/DifferentialPrivacy/file,  /rai/v1/privacy/DifferentialPrivacy/anonymize | Using the first API, we can upload the file we want to check for differential privacy and using the second API we can add suppression, noise etc. to the file values. |
| /rai/v1/privacy-files/video/anonymize | Detect and anonymize PII entities in video frames by drawing the bounding boxes |
| /rai/v1/privacy-files/PDF/anonymize | Detect and anonymize PII entities in pdf pages in text or image format. |
| /rai/v1/privacy-files/PPT/anonymize | Detect and anonymize PII entities in ppt slides in text, image or tabular format. |
| /rai/v1/privacy-files/DOCX/anonymize | Detect and anonymize PII entities in docx in text, image format. |
| /rai/v1/privacy-files/CSV/anonymize | Detect and anonymize PII entities in CSV. |
| /rai/v1/privacy-files/json/anonymize | Detect and anonymize PII entities in Json. |
| /rai/v1/privacy-files/anonymize | Detect and anonymize PII entities from any filetype. |
| /rai/v1/privacy-files/loadRecognizer | Dynamically registers **custom entity recognizers** |
| /rai/v1/privacy-files/getRecognizer | It detects all recognizers |

**Safety**

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /api/v1/safety/profanity/addProfaneWords | Profane words can be added to the existing profane words list. |
| /api/v1/safety/profanity/analyze | check if the text contains any profane words or not and we can get the toxicity score for the same |
| /api/v1/safety/profanity/censor | Censors any profane words identified in the text |
| /api/v1/safety/profanity/imageanalyze | Checks if the image falls under any of the following labels – drawings, hentai, porn, neutral or sexy |
| /api/v1/safety/profanity/imageGenerate | Generate images based on the prompt and can check under which label(drawings, hentai, porn, neutral or sexy) they would fall |
| /api/v1/safety/profanity/videosafety | Check under which labels the uploaded video belongs to – drawings, hentai, neutral, porn or sexy and can mask those profane objects in the video |
| /api/v1/safety/profanity/nudanalyze | Checks if the uploaded image or video contains any nudity and can blur the same. |
| /api/v1/safety/profanity/nudvideosafety | Detects the specific parts of nudity in the given video. |
| /api/v1/safety/profanity/maliciousUrl | Detects the malicious url and returns passed or unmoderated |
| /api/v1/safety/profanity/csvSafety | Analyze the csv file content and remove the malicious content from it. |

# Security

## Introduction

The introduction of a security module in RAI is a crucial step to safeguard the integrity and reliability of the platform. By incorporating robust security measures, RAI aims to protect user data, prevent unauthorized access, and ensure the platform's resilience against cyber threats. This module likely encompasses various security protocols, encryption techniques, authentication mechanisms, and risk management strategies to create a secure environment for users and their transactions.

Security module is a software component safeguarding digital systems by implementing robust protection mechanisms. Key features include authentication, authorization, encryption, intrusion detection, and access control to prevent unauthorized access, data breaches, and system failures.

## Features of a Security Module

A security module typically incorporates the following features:

* Authentication: Verifies user identity to grant access.
* Authorization: Defines permitted actions based on user roles and privileges.
* Encryption: Protects data confidentiality through secure encoding.
* Intrusion Detection: Monitors for suspicious activities and threats.
* Access Control: Restricts system access to authorized individuals.
* Audit Logging**:** Records system activities for security analysis.
* Data Integrity: Ensures data accuracy and consistency.
* Non-repudiation: Prevents denial of actions or transactions.

Component of Security Bulk Vulnerability Assessment

It is a comprehensive model assessment report that involves simulating a variety of potential attacks on the model to identify vulnerabilities. This process allows us to understand how the model might be compromised under different scenarios. In addition to analyzing these threats, we are also proposing a defense strategy designed to protect user models from the predicted attacks. This dual approach ensures that we not only identify weaknesses but also provide actionable solutions to enhance the security of the models.

## Endpoints and Functionalities

The following table lists the available Security endpoints and their functionalities. Refer to the API documentation: [Security Models](https://github.com/Infosys/Infosys-Responsible-AI-Toolkit/blob/main/responsible-ai-safety/Docs/SAFETY_endpoints_instructions.pdf)

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /v1/workbench/adddata | Use for Uploading data |
| /v1/workbench/data | To retrieve the details of the uploaded data |
| /v1/workbench/addmodel | Use for Model Upload |
| /v1/workbench/model | To retrieve the details of the uploaded model |
| /rai/v1/security\_workbench/attack | To check which attacks is applicable, user have to provide classifier and data type to this api endpoint |
| /v1/workbench/batchgeneration | Use for Batchgeneration |
| /rai/v1/security\_workbench/runallattacks | Use for creating the report and storing it in zip format in the database. |
| /v1/report/downloadreport | Use for downloading the generated report |

# Responsible AI toolkit UI

## Introduction

Responsible AI toolkit UI designed with user experience at the forefront, offering an intuitive and organized interface that brings various functionalities to your fingertips. The interface is structured around multiple tabs, each serving a distinct purpose to accommodate different types of data inputs and outputs. Whether you're working with text, images, videos, audio, file or code prompts, each tab dynamically interfaces with the backend APIs to process your requests and return the relevant results.

This multi-tab setup allows users to seamlessly switch between various tasks without cluttering the workspace, providing a clean, efficient environment. It ensures that no matter the format or nature of the input, the outputs are delivered in a structured and easy-to-understand way. Whether you're analyzing visual content, interacting with multimedia, or working with code-based requests, each section of the interface is purpose-built to support and display results in a way that enhances productivity and reduces complexity.

By abstracting complex backend processes into distinct views, our platform empowers users to effortlessly navigate and interact with a variety of functionalities, making it the perfect tool for both technical and non-technical users alike

## Features of a UI

A Infosys Responsible AI toolkit UI module typically incorporates the following features:

* Admin: Admin is the supporting module, which is used for configuring the modules like Privacy, Safety, ML etc. User can create recognizer, custom templates, configure Thresholds and map it to created account and portfolio.
* Frontend: The shell application in a micro frontend architecture is like the main framework or container. It acts as the backbone of the app, managing the loading and display of various independent MFEs. The shell provides a consistent and unified look and feel across the entire app, while each MFE is responsible for its own part of the UI. The shell ensures everything works together smoothly, letting users interact with different MFEs seamlessly and efficiently.
* Shell Application: The shell application in a micro frontend architecture is like the main framework or container. It acts as the backbone of the app, managing the loading and display of various independent MFEs. The shell provides a consistent and unified look and feel across the entire app, while each MFE is responsible for its own part of the UI. The shell ensures everything works together smoothly, letting users interact with different MFEs seamlessly and efficiently.
* Backend: This module is focused on registration and authentication handles user account management, including user registration, login, password validation, and session management.
* File Storage: This module provides a simplified, high-level interface for interacting with Azure Blob Storage. It encapsulates the complexities of the Azure Storage SDK, offering convenient functions for common operations such as uploading, downloading, listing, and deleting blobs. This wrapper aims to make working with Azure Blob Storage more accessible and efficient by handling authentication, container management, and error handling internally, allowing users to focus on the core logic of their applications. It abstracts away the need to directly interact with the Azure Storage SDK's client objects, providing a more streamlined and Pythonic experience.
* Telemetry**:** A python backend module defining the various tenets structure for ingestion of the API's data into Elasticsearch indexes. It provided customizable input validation and insertion of data coming from tenets into elasticsearch, which can be further displayed using kibana.
* Workbench: Workbench is for processing unstructured text data and generating risk management reports based on questionnaire responses. It serves as a hub for analyzing text data and assessing risks.

## Endpoints and Functionalities

The following table lists the available Responsible AI toolkit UI endpoints and their functionalities.

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /rai/v1/telemetry/privacytelemetryapi | Receive the telemetry data from privacy API's and push to elastic search |
| /rai/v1/telemetry/moderationtelemetryapi | Receive the telemetry data from Decoupled Moderation API's and push to elastic search |
| /rai/v1/telemetry/coupledmoderationtelemetryapi | Receive the telemetry data from coupled Moderation API's and push to elastic search |
| /rai/v1/telemetry/admintelemetryapi | Receive the telemetry data from admin API's and push to elastic search |
| /rai/v1/telemetry/errorloggingtelemetryapi | Receive the error telemetry data from all tenets and push to elastic search |
| /rai/v1/telemetry/accmastertelemetryapi | Receive the telemetry data from admin API's and push to elastic search |
| /rai/v1/telemetry/usermanagementtelemetryapi | Receive the user login telemetry data from rai backend and push to elastic search |
| /rai/v1/telemetry/registertelemetryapi | Receive the user registration telemetry data from rai backend and push to elastic search |
| /rai/v1/telemetry/profanitytelemetryapi | Receive the telemetry data from rai safety and push to elastic search |
| /rai/v1/telemetry/explainabilitytelemetryapi | Receive the telemetry data from explainability api and push to elastic search |
| /rai/v1/telemetry/evalllmtelemetryapi | Receive the telemetry data from evallm api of moderation layer and push to elastic search |
| /api/v1/rai/admin/getRecognizer | It is used to get all the Recognizers |
| /api/v1/rai/admin/getAccount | It is used to get all the accounts |
| /api/v1/rai/admin/DataRecogGrp | It creates a new recognizer |
| /api/v1/rai/admin/DataRecogGrplist | It will show all the recognizer list |
| /api/v1/rai/admin/DataRecogGrpEntites | It will give all the patterns/data |
| /api/v1/rai/admin/DataEntitesUpdate | It updates the data entity name |
| /api/v1/rai/admin/SafetyUpdate | It updates Safety |
| /api/v1/rai/admin/DataGrpUpdate | It will update the name of recognizers |
| /api/v1/rai/admin/DataEntityAdd | Add New Entity to the recognizer |
| /api/v1/rai/admin/DataRecogGrpDelete | It is used to delete recognizer |
| /api/v1/rai/admin/DataRecogEntityDelete | It deletes single entity from recognizer |
| /api/v1/rai/admin/PrivacyParameter | Map privacy recognizer to acc to portfolio |
| /api/v1/rai/admin/SafetyParameter | Map safety configuration to acc to portfolio |
| /api/v1/rai/admin/AccMasterEntry | Create/Fetch account & portfolio name |
| /api/v1/rai/admin/AccMasterList | It gives all the Account & Portfolio Name |

|  |  |
| --- | --- |
| /api/v1/rai/admin/AccDataList | Pass accMasterID to get recognizer mapped with that account |
| /api/v1/rai/admin/AccSafetyListAccountWise | Pass accMasterID to get SafetyConfiguration with that account(Safety) |
| /api/v1/rai/admin/PrivacyDataList | Pass Portfolio & Account Name(Privacy) |
| /api/v1/rai/admin/SafetyDataList | Pass Portfolio & Account Name(Safety) |
| /api/v1/rai/admin/AccMasterDelete | It deletes Account Master |
| /api/v1/rai/admin/AccDataDelete | Delete Recognizer From Account |
| /api/v1/rai/admin/LoadRecognizers | Default Recognizer will be loaded, from env file |
| /api/v1/rai/admin/AccEntityAdd | Add New Recognizer to account & Portfolio |
| /api/v1/rai/admin/PrivacyEncrypt | It will change hashify parameter to true/false |
| /api/v1/rai/admin/ThresholdUpdate | Update Threshold |
| /api/v1/rai/admin/ConfigApi | Getting Api endpoint from API Post |
| /api/v1/rai/admin/ApiPost | It creates API EndPoints |
| /api/v1/rai/admin/ApiUpdate | It updates API Endpoint |
| /api/v1/rai/admin/ApiDelete | It is udes to Delete API |
| /api/v1/rai/admin/UpdateOpenAI | It is used to update OpenAI |
| /api/v1/rai/admin/UpdateReminder | It is used to update Remainder |
| /api/v1/rai/admin/UpdateGoalPriority | It is used to updateGoal Priority |
| /api/v1/rai/admin/getOpenAI | It is used to get OpenAI |
| /api/v1/rai/admin/userRole | It is used to post UserRole |
| /api/v1/rai/admin/getRole | It is used in getting the user role |
| /api/v1/rai/admin/FMConfigEntry | FM Config Entry |
| /api/v1/rai/admin/FMConfigEntryList | Get FM List |
| /api/v1/rai/admin/FmGrpDataList | Pass Account masterid |
| /api/v1/rai/admin/FmGrpDataUpdate | Pass Account masterid(Update) |
| /api/v1/rai/admin/getAttributes | Data List of FM |
| /api/v1/rai/admin/FmConfigDelete | FM Config Delete |
| /api/v1/rai/admin/ModerationCheckLists | Data List of FM |
| /api/v1/rai/admin/RestrictedTopicsLists | Restricted Topic List |
| /api/v1/rai/admin/OutputModerationCheckLists | Moderation List |
| /api/v1/rai/admin/uploadFile | RAG-UploadFile |
| /api/v1/rai/admin/getFiles | RAG-getFiles |
| /api/v1/rai/admin/setCache | RAG-setCache |
| /api/v1/rai/admin/getEmbedings | RAG-getEmbedings |
| /api/v1/rai/admin/clearEmbedings | RAG-clearEmbedings |
| /api/v1/rai/admin/deleteFile | RAG-deleteFile |
| /api/v1/rai/admin/getAWSCreds | It gets AWSCredential |
| /api/v1/rai/admin/addAWSCreds | It adds AWS Credential |
| /api/v1/rai/admin/updateAWSCreds | It update AWS Credential |
| /api/v1/rai/admin/createCustomeTemplate | Create Custom Template |
| /api/v1/rai/admin/getCustomeTemplate/{userId} | Provide the userID and gets all the respective details of custome template |
| /api/v1/rai/admin/getTemplate | Get template - The mode of private/master can be view |
| /api/v1/rai/admin/updateCustomeTemplate | Updates Custome Template to change template data |
| /api/v1/rai/admin/deleteCustomeTemplate | To Delete TemplateID |
| /api/v1/rai/admin/deleteSubTemplate | To Delete Sub TemplateID |
| /api/v1/rai/admin/AccTemplateMap | It is used for acount template mapping |
| /api/v1/rai/admin/getModMaps | It is used to Get Mod Maps userID |
| /api/v1/rai/admin/getModMap | It is used to Get Mod Maps Category |
| /api/v1/rai/admin/getModConfig | It is used to Get Mod Sub Category |
| /api/v1/rai/admin/getTempMap | It is used to get TemplateMap from accMasterId |
| /api/v1/rai/admin/deleteTempMap | It is used to Delete TemplateMap |
| /api/v1/rai/admin/addTempMap | It updates TempMap |
| /api/v1/rai/admin/removeTempMap | Remove TempMap from request,response or comparision |
| /api/v1/azureBlob/addFile | Add File - The functionality is used to save a file in CosmosDB under a specified container. |
| /api/v1/azureBlob/getBlob | Get File - The functionality is used to retrieve a file from CosmosDB with in a specified container. |
| /api/v1/azureBlob/delete\_blob | Delete File - The functionality is used to delete a file from a specified container. |
| /api/v1/azureBlob/updateFile | Update File - The functionality is used to update a file in a container by replacing the existing file with a new version of the file. |
| /api/v1/azureBlob/addContainer | Add Container - The functionality is used to add a new blob storage container. |
| /api/v1/azureBlob/List | List Containers - The functionality is used to list all storage containers, retrieving the names of all containers within a storage account. |

Red Teaming

# Introduction

Red Teaming is a proactive security measure designed to identify and mitigate vulnerabilities in AI systems, particularly Large Language Models (LLMs). It involves simulating adversarial attacks to evaluate the robustness, safety, and ethical compliance of these models. By mimicking real-world threats, Red Teaming ensures that AI models are resilient and responsibly deployed. Infosys Responsible AI Toolkit incorporates advanced red teaming methodologies to enhance the security and reliability of AI systems.

# Types of Red Teaming

## **Automated Red Teaming**

Automated Red Teaming utilizes scripts and tools to generate adversarial prompts and evaluate AI models at scale. This method is efficient and allows for extensive testing across diverse scenarios. It supports batch processing and integrates with APIs for streamlined execution. Automated techniques are particularly useful for large-scale evaluations and continuous monitoring of model vulnerabilities.

## **Manual Red Teaming**

Manual Red Teaming involves human experts crafting unique and complex adversarial scenarios to test AI models. This approach leverages creativity and domain expertise to uncover unexpected weaknesses. While it provides deep insights into model behavior, it is time-consuming and requires significant effort. Manual red teaming is ideal for targeted assessments and high-stakes applications where precision is critical.

# PAIR Technique (Prompt Adversarial Iterative Refinement)

The PAIR technique is designed to evaluate and improve the robustness of language models against adversarial prompts. It follows a structured process involving initialization, attack generation, response collection, evaluation, refinement, and iteration. Adversarial prompts are crafted to bypass safety mechanisms and elicit forbidden behavior. Evaluator models such as GCGJudge and GPTJudge assess the responses for safeguard violations and truthfulness. The iterative refinement process continues until the prompts achieve the desired jailbreak score, indicating a successful attack.

# TAP Technique (Tree of Attacks with Pruning)

TAP is an automated, query-efficient method for generating jailbreak prompts using tree-of-thought reasoning. It operates in a black-box setting, relying solely on input-output queries to the target model. The process includes initialization, attack generation, pruning of irrelevant prompts, response evaluation, and iterative refinement. Evaluator models score the responses, and low-performing prompts are pruned to optimize the attack strategy. TAP is effective for exploring large search spaces and identifying vulnerabilities without direct access to model internals.

# Evaluator Models

Red Teaming employs evaluator models to assess the safety and ethical compliance of AI responses. GCGJudge evaluates responses for safeguard violations using predefined keywords and assigns scores based on the presence of forbidden phrases. GPTJudge assesses responses for truthfulness, clarity, and adherence to ethical standards, using a scoring system from 1 (safe and ethical) to 10 (severely violating guidelines). These models ensure that AI outputs are scrutinized rigorously and help refine adversarial prompts effectively.

# Supported Models and API Endpoints

The Red Teaming toolkit supports various AI models including GPT-3, GPT-4, Gemini-Pro, Gemini-Flash. It also allows integration with endpoint-based models via APIs. Key endpoints include /v1/redteaming/pair for PAIR attacks and /v1/redteaming/tap for TAP attacks. Batch processing endpoints enable large-scale evaluations and report generation. These APIs facilitate seamless interaction with the red teaming service and support comprehensive vulnerability assessments.

# Report Management

The toolkit includes robust report management features that compile attack results into detailed PDF reports. Reports include executive summaries, category-wise analysis, success metrics, and individual test findings. These reports are stored in a database for future reference and can be retrieved using specific report IDs. This functionality ensures transparency and traceability in red teaming evaluations.

# Endpoints and Functionalities

|  |  |
| --- | --- |
| **Endpoint** | **Description** |
| /v1/redteaming/models | Returns information about available attack and judge models. |
| /v1/redteaming/pair | Executes a single PAIR attack with specified parameters. |
| /v1/redteaming/pair/batch | Processes multiple PAIR attack scenarios from an Excel file. |
| /v1/redteaming/tap | Executes a single TAP attack with specified parameters. |
| /v1/redteaming/tap/batch | Processes multiple TAP attack scenarios from an Excel file. |
| /v1/redteaming/report | Retrieves generated vulnerability assessment reports in PDF format. |

This section summarizes the typical flow for utilizing the Red Teaming API endpoints.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

## Contact

Infosysraitoolkit@infosys.com

© 2024-2025 Infosys Limited, Bengaluru, India.

All Rights Reserved. Infosys believes the information in this document is exact as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.