# **Project 6: Ethical AI and Bias Detection System**

Assigned to: Trainee, Trainee,

**Background:**

AI systems in healthcare, such as disease prediction models, treatment recommendation engines, and medical language models, are revolutionizing patient care. However, these systems can sometimes violate ethical AI standards, leading to unfair outcomes, biased decisions, or compromised patient trust. Detecting and addressing flaws in these systems is critical to ensure they align with ethical principles and promote fairness, accountability, and transparency in healthcare AI.

**Problem Statement:**

Develop a Multi-Agent System (MAS) to detect biases and ethical flaws in healthcare AI systems, ensuring compliance with global ethical AI standards while providing actionable recommendations for improvement.

**Tasks:**

* Investigate the current challenges with using AI in healthcare domain and compile global ethical AI guidelines (e.g., WHO, IEEE, EU regulations) w.r.t health sector.
* Identify key principles relevant to healthcare AI, such as fairness, accountability, transparency, and inclusivity.
* Identify common flaws and biases in healthcare AI systems, such as Data bias, Algorithmic bias, outcome bias etc.
* Develop a Multi-Agent System (MAS) to analyze and evaluate existing AI models for flaws, including automated checks for non-compliance with ethical standards and detection of biases and their sources.
* Develop a detailed guideline to develop future in-house AI systems that follow ethical AI standards.
* Generate detailed reports on detected flaws and biases, explaining their implications for patient care.
* Provide actionable recommendations to improve compliance with ethical standards.
* Document system design, testing process, and evaluation results. Include guidelines for integrating ethical principles into future AI systems.
* Be Creative, think smart, come up with elegant solutions. Any solution that involves creativity, mathematical modelling, and is end-to-end production grade will always be valued.

**Deliverables:**

* Design a Multi-Agent System (MAS) with specialized agents responsible for:
  + Data Analysis Agent: Aggregating global ethical AI guidelines and identifying key principles for fairness, transparency, and accountability.
  + Bias Detection Agent: Analyzing AI systems for data, algorithmic, and outcome biases using real-time queries.
  + Compliance Assessment Agent: Scoring AI systems based on adherence to ethical guidelines.
  + Recommendation Agent: Suggesting actionable improvements for non-compliant systems.
  + Coordinator Agent: Orchestrating the workflow among other agents and ensuring timely communication of findings.
* Detailed report on flaws detected in existing systems, their sources, and their impact on patient care.
* Detailed guide book to follow while developing future in-house AI systems.
* Actionable insights for mitigating flaws and improving model compliance with ethical standards.
* Visualizations: Graphical representations of flaws, bias trends, and compliance scores. (where ever valid)
* Comprehensive documentation covering system development, evaluation, methodology, and future recommendations.

**Project Engineering Guidelines**

Here is a suggested research and development engineering cycle to follow:

* First do a thorough literature survey of the prior art related to the project and then shortlist the ones based on some justifiable rationale for this use case.
* Understand the models at depth including mathematics so you know what is happening inside the state-of-the-art method.
* Find suitable studies that relate to your project and analyze how they approached this problem.
* Build a software architecture diagram including class and sequence diagrams you were taught in OOP
* Build your pipeline in real-time by using big data technologies if needed
* Bonus would be awarded if you did model engineering with strong mathematics background instead of just doing API calls like programmers
* Pick or create your testing and performance evaluation metrics and then measure them to show the inner working and behavior of your models on real-world scenarios. Moreover, some notion of accuracy and effectiveness of such systems need to be established.
* Add comments in the code for documentation purpose, code with no comments will be penalized heavily.

# **Project 7: Disease Outbreak Prediction and Management System**

Assigned to: Trainee, Trainee

**Background**:  
The emergence of pandemics poses significant challenges for global health, requiring proactive, coordinated responses. This project aims to integrate Knowledge Graphs (KGs), Multi-Agent Systems (MAS), and Classical AI (CAI) to build a system that predicts and manages disease outbreaks. The system will combine epidemiological data, disease progression patterns, and patient encounters to identify potential outbreaks early. Agents in the MAS will communicate in real time to raise alarms, coordinate interventions, and manage resources, while classical AI models provide predictive analytics to prevent escalation. This integrated system ensures that responses are data-driven, timely, and effective, thereby reducing societal and economic disruptions.

**Problem Statement:**Develop an integrated system using KGs, MAS, and CAI to predict disease outbreaks, identify regions at risk, and coordinate real-time interventions to proactively manage pandemics.

**Tasks:**

* Design a Multi-Agent System (MAS) for tasks such as monitoring rising disease trends, raising alarms, and coordinating interventions.
* Ensure agents can communicate effectively and collaborate on decision-making in real time.
* You may use Classical AI models or data driven techniques to predict regions at high risk of outbreaks, forecast timelines, and identify potential interventions.
* Integrate KG, MAS, and CAI into a unified system capable of handling real-time outbreak management scenarios.
* Integrate real-time communication between agents to exchange findings, raise alarms for critical flaws, and propose immediate interventions.
* Test the system’s ability to identify and respond to outbreaks using historical data and simulated scenarios.
* Validate the system’s accuracy, response time, and effectiveness in controlling the spread of diseases.
* Document the development process, system architecture, and evaluation results.
* Create visualizations to explain predictions, outbreak trends, and the coordination activities of agents.
* Be Creative, think smart, come up with elegant solutions. Any solution that involves creativity, mathematical modelling, and is end-to-end production grade will always be valued.

**Deliverables:**

* A Multi-Agent System for monitoring, coordinating, and managing disease trends across regions.
* Predictive models for forecasting outbreak regions, timelines, and severity.
* An integrated system that combines the KG, MAS, and AI models for proactive outbreak management.
* Performance evaluation reports, including outbreak prediction accuracy, response time, and intervention outcomes.
* Dashboards or visual tools demonstrating outbreak predictions, agent activities, and intervention results.
* Comprehensive documentation covering system design, development process, and future recommendations.

**Project Engineering Guidelines**

Here is a suggested research and development engineering cycle to follow:

* First do a thorough literature survey of the prior art related to the project and then shortlist the ones based on some justifiable rationale for this use case
* Understand the models at depth including mathematics so you know what is happening inside the state-of-the-art method
* Find suitable datasets or scientifically generate them for this use case
* Build a software architecture diagram including class and sequence diagrams you were taught in OOP
* Build your pipeline in real-time by using big data technologies if needed
* Bonus would be awarded if you did model engineering with strong mathematics background instead of just doing API calls like programmers
* Pick or create your testing and performance evaluation metrics and then measure them to show the inner working and behavior of your models on real-world scenarios. Moreover, some notion of accuracy and effectiveness of such systems need to be established.
* Add comments in the code for documentation purpose, code with no comments will be penalized heavily.

**Project 8: Knowledge Graph Embedding Based Information Retrieval**

Assigned to:

**Background:**

Traditional search engines rely heavily on keyword matching, which often fails to understand the context and semantic nuances of user queries. To bridge this gap, Knowledge Graphs (KGs) have emerged as a powerful tool for structuring and representing complex relationships between entities, enabling more intuitive and context-aware information retrieval. Coupled with embedding techniques, which transform textual and relational data into continuous vector spaces, there lies a promising avenue to revolutionize Information Retrieval (IR) systems.

**Problem Statement:**

Develop an advanced Information Retrieval (IR) system that leverages Knowledge Graphs and embedding techniques to ensure accurate retrieval of relevant nodes/relationships.

**Tasks:**

* Understand and analyze the clinical KG.
* Identify different ways to embed the graph nodes and relationships.
* Understand vector similarity search and how it’s applicable to this task.
* Generate vector embeddings used to retrieve relevant nodes/relationships etc
* The IR system should be dynamic enough to retrieve patients that specify a certain criterion.
* Be Creative, think smart, come up with elegant solutions. Any solution that involves creativity, mathematical modelling, and is end-to-end production grade will always be valued.

**Deliverables:**

**Graph Embedding Framework**

* Selection and justification of embedding algorithms (e.g., Node2Vec, TransE, GraphSAGE)
* Development of custom modules or scripts for embedding generation
* Conducting a comparative analysis of different embedding methods

**Vector Similarity Search Module**

* Implementation of similarity search algorithms (e.g., cosine similarity, Euclidean distance)
* Integration with embedding storage solutions
* Optimization for scalability and low-latency responses

**Vector Embeddings Dataset**

* Creation of preprocessed and cleaned embedding data
* Establishment of storage solutions ensuring fast access and retrieval
* Documentation detailing embedding generation processes and parameters

**End-to-End IR System**

* Design and provision of system architecture diagrams
* Integration of all modules into a cohesive pipeline
* Development of a user interface for interacting with the chatbot

**Comprehensive Documentation**

* Preparation of technical specifications and architecture overviews

**Performance Evaluation Report**

* Benchmarking all the various approaches tried and reporting their metrics
* Gathering of user feedback and formulation of iterative improvement plans

**Project Engineering Guidelines**

Here is a suggested research and development engineering cycle to follow:

* First do a thorough literature survey of the prior art related to the project and then shortlist the ones based on some justifiable rationale for this use case.
* Understand the models at depth including mathematics so you know what is happening inside the state-of-the-art method.
* Find suitable studies that relate to your project and analyze how they approached this problem.
* Build a software architecture diagram including class and sequence diagrams you were taught in OOP
* Build your pipeline in real-time by using big data technologies if needed
* Bonus would be awarded if you did model engineering with strong mathematics background instead of just doing API calls like programmers
* Pick or create your testing and performance evaluation metrics and then measure them to show the inner working and behavior of your models on real-world scenarios. Moreover, some notion of accuracy and effectiveness of such systems need to be established.
* Add comments in the code for documentation purpose, code with no comments will be penalized heavily.

**Project 9: Knowledge Graph Construction from Provider Notes**

Assigned to:

**Background:**

Knowledge graphs (KGs) are a powerful way to represent and connect information by linking entities and their relationships in a structured format, often built from sources like databases and forms. However, a vast amount of valuable knowledge lies in unstructured data, such as provider notes, which contain detailed descriptions of patient conditions, treatments, and care plans. Large Language Models (LLMs) have revolutionized the way we process and utilize unstructured text, enabling not just the extraction of entities but the direct construction of comprehensive knowledge graphs. By leveraging the advanced reasoning and contextual understanding of LLMs, we can transform unstructured provider notes into rich, interconnected graphs that represent both entities and their relationships.

**Problem Statement:**

Design a system to construct a patient-centric knowledge graph from the provider notes of patients that utilizes RAG and Large Language Models (LLMs) for criteria inference.

**Tasks:**

* Conduct a comprehensive literature review on Knowledge Graphs (KGs), Retrieval-Augmented Generation (RAG), and Large Language Models (LLMs).
* Analyze the existing approaches for constructing knowledge graphs from unstructured data, particularly in healthcare.
* Develop a framework to preprocess and clean provider notes for optimal use with LLMs.
* Design schema for better understandability of the graph
* Design and implement a RAG-based pipeline to retrieve relevant information from provider notes for criteria inference.
* Try out different embedding techniques for enhanced and accurate document retrieval
* Optimize the system for scalability and efficiency to handle large volumes of provider notes.

**Deliverables:**

* A functional pipeline leveraging RAG to retrieve relevant information from provider notes, complete with source code and implementation details.
* A comparative analysis of different embedding techniques for document retrieval
* A prototype of the knowledge graph construction process, including a sample knowledge graph derived from all notes of the patients.
* Documentation of the transformation process from unstructured text to a structured graph format.
* Comprehensive documentation of the system, including the architecture, implementation details, and evaluation results.
* Recommendations for future improvements and extensions of the system.

**Project Engineering Guidelines**

Here is a suggested research and development engineering cycle to follow:

* First do a thorough literature survey of the prior art related to the project and then shortlist the ones based on some justifiable rationale for this use case.
* Understand the models at depth including mathematics so you know what is happening inside the state-of-the-art method.
* Find suitable studies that relate to your project and analyze how they approached this problem.
* Build a software architecture diagram including class and sequence diagrams you were taught in OOP
* Build your pipeline in real-time by using big data technologies if needed
* Bonus would be awarded if you did model engineering with strong mathematics background instead of just doing API calls like programmers
* Pick or create your testing and performance evaluation metrics and then measure them to show the inner working and behavior of your models on real-world scenarios. Moreover, some notion of accuracy and effectiveness of such systems need to be established.
* Add comments in the code for documentation purpose, code with no comments will be penalized heavily.

**Project 10: Phenotypes Extraction from Radiology/Pathology Reports and Ingestion into Knowledge Graph**

Assigned to:

**Background:**

Phenotype extraction from radiology and pathology reports offers a valuable opportunity to enrich clinical knowledge graphs (KGs) with detailed, structured insights derived from unstructured data. These reports often contain critical information about patient conditions, biomarkers, and disease phenotypes, which are essential for research and decision-making. Large Language Models (LLMs) have transformed the way we process such unstructured text by enabling the extraction of complex clinical phenotypes with high accuracy and contextual understanding. This project aims to extract phenotypes from radiology and pathology reports and design a schema that seamlessly integrates this information into an existing knowledge graph. By leveraging LLMs for phenotype extraction and carefully crafting an ingestion schema, this project will enhance the knowledge graph’s ability to support advanced clinical queries and analytics.

**Problem Statement:**

Develop a system to extract clinical phenotypes from radiology and pathology reports using Large Language Models (LLMs) and design a schema for integrating the extracted information into an existing knowledge graph.

**Tasks:**

* Review existing methods for phenotype extraction from clinical text.
* Analyze radiology and pathology reports to identify key phenotype-related information.
* Preprocess data to clean and standardize unstructured text for input to LLMs.
* Utilize Large Language Models (LLMs) to extract phenotypes, biomarkers, and related entities from radiology and pathology reports.
* Design a schema for the knowledge graph to accommodate extracted phenotypes and link them to existing entities.
* Map extracted phenotypes to the designed schema and integrate them into the existing knowledge graph
* Optimize the system for scalability and efficiency to handle large volumes of provider notes.

**Deliverables:**

* A functional pipeline utilizing LLMs for extracting phenotypes from clinical text, complete with source code and implementation details.
* A detailed schema designed for the ingestion of phenotypes and their relationships into the existing knowledge graph.
* A prototype of the updated knowledge graph containing the extracted phenotypes and relationships.
* Comprehensive documentation of the system, including the architecture, implementation details, and evaluation results.
* Recommendations for future improvements and extensions of the system.

**Project Engineering Guidelines**

Here is a suggested research and development engineering cycle to follow:

* First do a thorough literature survey of the prior art related to the project and then shortlist the ones based on some justifiable rationale for this use case.
* Understand the models at depth including mathematics so you know what is happening inside the state-of-the-art method.
* Find suitable studies that relate to your project and analyze how they approached this problem.
* Build a software architecture diagram including class and sequence diagrams you were taught in OOP
* Build your pipeline in real-time by using big data technologies if needed
* Bonus would be awarded if you did model engineering with strong mathematics background instead of just doing API calls like programmers
* Pick or create your testing and performance evaluation metrics and then measure them to show the inner working and behavior of your models on real-world scenarios. Moreover, some notion of accuracy and effectiveness of such systems need to be established.
* Add comments in the code for documentation purpose, code with no comments will be penalized heavily.