# Advice report(Timeline)

## Introduction

This technical advice report focuses on the selection of frameworks and programming languages for building an AI-driven investigative journalism workflow. Specifically, we will explore the usage of JavaScript (JS) and various frameworks that enable the integration of object detection AI and text analysis AI, such as CHAGPT, into a system for efficient data analysis.

## Selection of programming languages

JavaScript (JS) is a versatile programming language widely used for web development. Its ability to run on both the client-side and server-side makes it a suitable choice for building AI-driven applications in investigative journalism. With JavaScript, developers can leverage existing libraries and frameworks that facilitate the integration of AI models and APIs into the workflow.

## Frameworks for AI Integration

**Node.js**

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It enables server-side development and provides a rich ecosystem of libraries and modules. Investigative journalists can utilize Node.js to build robust backend systems that handle data analysis, integration with AI models, and API interactions. It allows seamless integration with popular AI libraries and frameworks such as TensorFlow.js for object detection and Natural Language Processing (NLP) libraries for text analysis.

**Express.js**

Express.js is a minimalistic and flexible web application framework for Node.js. It simplifies the creation of APIs and facilitates the communication between frontend and backend components. Investigative journalists can leverage Express.js to develop a RESTful API that interacts with AI models for object detection and text analysis. The API endpoints can receive data for analysis and return the results in a structured format, enabling seamless integration with frontend interfaces.

**React.js**

React.js is a popular JavaScript library for building user interfaces. It provides a component-based architecture that enables the creation of dynamic and interactive frontend interfaces. Investigative journalists can utilize React.js to build a user-friendly interface for data visualization, timeline representation, and interaction with AI analysis results. React.js seamlessly integrates with other libraries and frameworks, making it ideal for creating a responsive and intuitive UI.

# Integration of AI Models

**TensorFlow.js:**

TensorFlow.js is a JavaScript library that allows running trained AI models directly in the browser or on Node.js. Investigative journalists can utilize TensorFlow.js for object detection, leveraging pre-trained models or training their own models on relevant datasets. TensorFlow.js provides APIs for performing inference on images and videos, extracting key details such as objects, scenes, and sentiment.

**CHAGPT API:**

CHAGPT, an AI language model, can be integrated into the investigative journalism workflow using its API. The CHAGPT API allows developers to interact with the language model for text analysis, sentiment analysis, and generating key takeaways from articles or social media posts. Investigative journalists can leverage the CHAGPT API to analyze text data, extract relevant information, and provide insights for their investigations.

**Language Embedded Radiance Fields (Lerf):**

The Lerf platform, as described in the research, employs natural language processing (NLP) to assist investigative journalists in analyzing vast amounts of text data. This aligns with the report's recommendation of using AI text analysis tools like CHAGPT to extract key details and patterns from leaked documents, financial records, and government databases. Lerf's ability to identify named entities, detect trends, and cross-reference information with other sources enhances efficiency and supports reporting endeavors, as highlighted in the report.

# Data Management and Analysis

To efficiently manage and analyze large datasets, investigative journalists can leverage database systems like MongoDB or PostgreSQL. These databases provide powerful querying capabilities, indexing, and scalability. By integrating with Node.js and Express.js, data can be stored and retrieved from the database, enabling efficient data analysis and retrieval for AI processing.

# Conclusion

In conclusion, JavaScript (JS) and its associated frameworks provide a robust foundation for building an AI-driven investigative journalism workflow. Node.js, Express.js, and React.js enable seamless integration of AI models and APIs, allowing for efficient data analysis, object detection, and text analysis. TensorFlow.js and the CHAGPT API serve as powerful tools for AI-related tasks. By leveraging these frameworks and programming languages, investigative journalists can enhance their workflow, analyze large datasets, and uncover crucial insights more effectively.