ELEC 490 / 498 Proposal

AI for Fake News and Deepfake Detection

Submitted by Group 16, Group members:

Ivan Samardzic

Erhowvosere Otubu

Mihran Asadullah

Faculty supervisor: Ali Etemad

October 7th, 2025

Executive Summary

With the rise of misinformation in digital media, the spread of inaccurate health information has become a critical concern, capable of negatively influencing public healthcare decisions and undermining trust in professional guidance. This project aims to build an Artificial Intelligence (AI) based web system used for automatic detection of medical-related misinformation. Users are prompted to input text or upload article links, and in turn receive classification predictions alongside respective accuracy metrics to justify validity and model conclusions.

Development of the project is structured around five quantified milestones. The data pipeline focuses on manipulating healthcare datasets and fine-tuning BERT/RoBERTa transformer models (Oct 1–30, 2025). The backend Development handles requests and communicates with the AI pipeline for real-time model predictions in Node.js/Python (Nov 1–20, 2025). The frontend development delivers the React/Next.js based web interface (Nov 21–Dec 20, 2025). Testing and explainability ensure the system highlights keywords, scores confidence, and validates outputs (Jan 5–25, 2026). The final integration and optional extension focuses on system-wide testing and potential inclusion of a web extension, as well as image and video misinformation detection (Jan 26–Feb 28, 2026).

The cost of building this project consists of GPU access and website hosting, estimated at $30 CAD/month. The project emphasizes efficiency while relying on open-source tools to keep costs low. Ethical considerations are another core aspect of the design process, and uncertainty within predictions is always communicated to the end user.

This document provides an overview of the full-stack architecture, AI design, testing strategy, and resource allocation of the product. By combining Natural Language Processing (NLP) with an accessible user interface and thoughtful ethical considerations, this project aims to provide an efficient tool to navigate through healthcare information with confidence.

Table of Contents

List of Tables

List of Figures

Introduction

Background

The rise of digital media has transformed how people access their news and information, but has also amplified the spread of deception in news, especially in sectors where accuracy is vital. In healthcare specifically, falsified articles can lead to several harmful consequences, such as risky health choices, unsafe self-treatment, and a skewed perception of healthcare in the eyes of the public. During global healthcare crises such as the COVID-19 pandemic, where procedures and guidelines are far understudied, misinformation has the ability to propagate through digital media at rates exceeding the ability for manual validation.

While existing approaches for validating misinformation do exist, they are largely limited in scope and require manual reviewers or general-purpose text classifiers, which may not be optimized for medical content specifically. These existing validation methods generally are not able to process at a scale required to limit the spread of inaccurate claims. This directly highlights the need for an automated method of evaluating news credibility with real-time feedback, specific to a domain such as healthcare and human wellness.

Scope & Objectives

This project seeks to build a framework for misinformation detection in the healthcare industry using artificial intelligence. The final product aims to consist of a fully functioning website, where users are able to upload articles or paste text as input, the system then outputs classification results of “True”, “False”, and “Uncertain”. These classification labels will be accompanied by their respective confidence scores, as well as a contextual description to help users visualize the rationale behind model predictions. Additionally, by establishing a scalable backend framework, this will allow the potential future integration of additional data sources and model updates.

While the primary focus of the proposal is the development of a website-based detection system, a web-based browser extension and image/video detection can also be implemented in future iterations to further enhance accessibility and real-time classification feedback.

System Overview

The proposed system is composed of three main components, as seen in Table 1, below. The frontend, developed using React and JavaScript, will provide an intuitive user experience and allow for a seamless display of classification results based on given user input. The backend handles efficient communication and routing between the frontend and AI model and is built on either Node.js or Next.js. Finally, the AI pipeline aims to integrate a Natural Language Processing (NLP) model such as BERT/RoBERTa, trained on pre-existing medical datasets, in order to produce classification results for given article inputs.

All proposed frameworks and software tools are subject to change as the project advances. Adjustments may be made to accommodate technical challenges or improve existing integrated solutions.

Table 1: High-level overview outlining component functionality within the system.

Component	**Description**	**Purpose**	**Integration**
Frontend (React/Next.js)	User web interface	Display input form and output classification summary	Sends user input to backend model
Backend (Node.js/Next.js)	Receive, process, and send requests	Middle layer between frontend and AI pipeline	Receives data, invokes model, and returns responses
AI Pipeline (Python, PyTorch, BERT/RoBERTa)	Machine Learning Inference Module	Performs text preprocessing, inference, and result generation	Processes content and outputs classification with confidence score

Further information regarding system architecture and model design is explored in succeeding sections of the report.

Functional Description

The approach for this project is to build a web-based application designed to automatically identify and classify misinformation in healthcare-related topics. The system integrates front-end, back-end, and AI/ML modules that collectively process a user’s input, analyze textual data using transformer-based models, and return explainable results indicating the likelihood of misinformation. Each part of the system listed serves a role in data collection, processing, and result presentation.

The system's workflow found in Figure 1 includes the following functionalities:

1. User Input: The user's input is text from a website, article, social media post, etc.
2. Backend Processing: As the text input is received, the data is processed through the AI model pipeline where the text is cleaned, tokenized, and evaluated for any healthcare misinformation detection
3. Classification Output: The model processes the text and generates a classification label of either 'True', 'False', or 'Uncertain' and a confidence score
4. Clarity Layer: Text like the keywords or phrases are highlighted by the system to help generate a clear explanation of the results found by the model
5. User Display: The display encompasses the model's results, explanation, and confidence score in a clear, user-friendly format

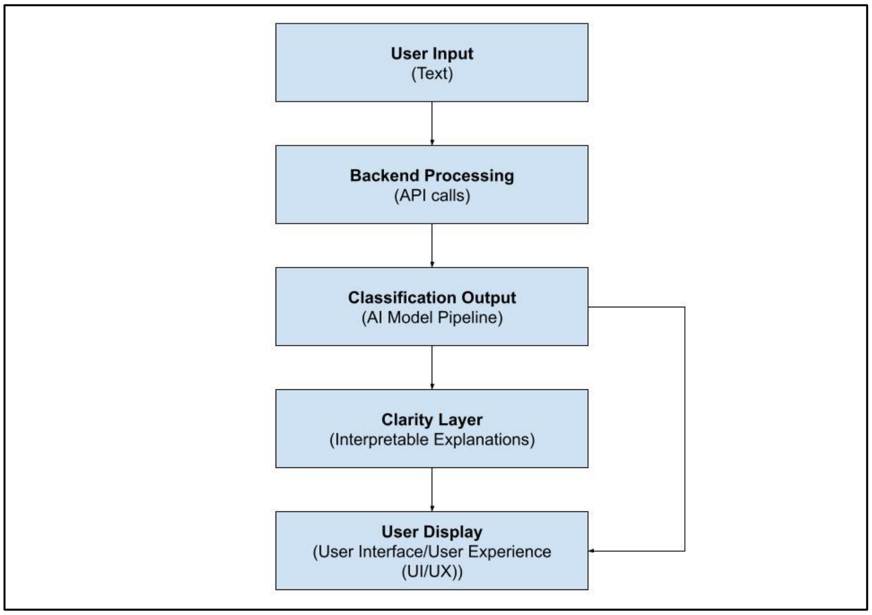

Figure 1: Block diagram outlining the system's architectural workflow.

Risk Considerations

Several ethical and technical risks must be considered to ensure reliable and responsible use of the project. One significant risk is the model's bias. Training data from the model may contain underlying biases that may influence the model’s ability to be fair or accurate. To address this issue, the team will use a diverse collection of datasets and regularly monitor the model’s performance. Another risk to note involves the case of the model producing inaccurate results in its classification process. These types of errors can affect the user’s trust and potentially contribute more to the spread of misinformation, in which the main goal of the project is to prevent the spread of fake news. The team will develop a confidence scoring system, continuously validate information through debugging and testing, and allow users to review evidence behind each result returned. Additionally, there is a risk of overreliance on the model’s output. This is where the user may treat the model’s classifications as absolute truth without doing their due diligence and confirming the results on their own. To reduce this, the user interface will clearly communicate that the results are probabilistic and prompt the user to verify the results themselves. By addressing these risks, the project aims to ensure ethical and safety measures are considered in all areas of the project.

The environmental and ethical considerations of this project center on responsible AI development and sustainable deployment. Ethically, the system is designed to promote transparency, fairness, and accountability in healthcare information verification. Since misinformation can directly influence health decisions, the team intends to uphold engineering code of ethics in the project. This involves prioritizing public welfare above all else, aiming to reduce possible harm caused by false medical claims [1]. The design emphasizes clarity to prevent blind reliance on the model’s results and to encourage users to critically think and challenge the produced results when necessary. Recognizing that a lot of AI models may contain some sort of a bias, the team will critically assess training data to mitigate bias. From an environmental perspective, the project will use energy-efficient models to limit energy consumption during training and deployment. Overall, the system seeks to balance technology with engineering social responsibility and ethical duties to society.

Design & Production

Development Process

This system will be developed using a milestone-based structure to ensure efficient work across all core components. Development of this project begins with preparing the data and training the baseline BERT/RoBERTa model focusing on classifying healthcare related misinformation. The backend infrastructure will then be built to support real-time model inference. The frontend will be a browser-based website developed using modern web and UI frameworks.

The project is organized around five major milestones, as shown in the table below.

Table 2: Summary of major milestones, their core objectives, and the planned timeline.

Milestone	Description	Timeline
Data Pipeline and Baseline Model	Collect labeled misinformation datasets with emphasis on healthcare and fine-tune BERT/RoBERTa model to establish baseline performance.	Oct 1 – Oct 30, 2025
Backend API Development	Implement secure Node.js backend and RESTful Python interface to serve model predictions and support web application functionality.	Nov 1 – Nov 20, 2025
Frontend UI	Develop React-based web interface and lightweight browser extension for real-time text analysis.	Nov 21 – Dec 20, 2025
Testing and Explainability	Integrate testing procedures including keyword highlighting and confidence scoring for interpretability.	Jan 5 – Jan 25, 2026
Final Integration and Optional Multimodal Extension	System integration and user testing. Optionally integrating a browser extension into the full pipeline and, if time allows, implement image/video misinformation detection using pretrained deepfake models.	Jan 26 – Feb 28, 2026

Contingency Plan

Potential challenges that could arise when approaching this project include dataset quality, model performance and time constraints. If the data retrieved is insufficient or weak, supervised analysis and data formatting methods will be used to bring the data retrieved up to standard. Furthermore, should the RoBERTa model underperform, other similar models such as BERT or BioBERT will be evaluated. If time is limited, image and video detection will be treated as an optional extension without compromising the text-based system's core functionality.

Division of Labour

Project responsibilities are distributed as follows:

- Ivan: Frontend and UI development, user interface design, and API integration.
- Sere: Documentation, backend inference infrastructure, and system design architecture.
- Mihran: Data pipeline, model training and optimization and testing.

This distribution allows for parallel progress across all group members while also allowing each group member to maintain ownership of their tasks.

Testing & Evaluation

The testing for this project will follow a modular approach. The goal being to target each subsystem independently before system-level integration. The testing procedure is as follows.

Table 3: Testing strategy, success criteria, and timeline.

Test Type	Description	Success Criteria	Timeline
Model Testing	Evaluate the fine-tuned BERT/RoBERTa model using validation and testing splits to measure detection effectiveness.	Achieve > 80 % accuracy score on test data.	Oct – Nov 2025
Backend Testing	Verify API functionality including correct responses, latency, and	All endpoints respond correctly with acceptable latency (< 3 seconds) and error handling.	Nov – Dec 2025

	error handling under various conditions.		
Frontend Testing	Conduct functional tests to ensure correct data flow, result display, and cross-browser compatibility.	Interface works reliably on major browsers (Chrome, Firefox, Edge) with correct outputs.	Dec 2025 – Jan 2026
Integration Testing	Perform end-to-end user testing to assess the full system's performance and usability.	System operates seamlessly with low latency and delivers clear results.	Jan – Feb 2026

Resource Requirements

Resource Scheduling

The project will use primarily open-source tools and university provided resources. Additionally, utilizing GPU-enabled environments such as Google Colab or Jupyter Notebook will support model training significantly. Key software tools include PyTorch, HuggingFace Transformers, Node.js, React and Chrome/Firefox extension APIs.

Resource needs for each major milestone/objectives are summarized in Table 4, below.

Table 4: Project milestones alongside their individual key resources.

Milestone	**Key Resources**
Data & Baseline Model	Datasets, GPU access, PyTorch/Transformers
Backend API	Node.js server, trained model weights
Frontend & Extension	React framework, extension SDKs
Testing	Testing frameworks, user feedback sessions
Integration	All of the above

Cost Analysis

Project estimation costs are summarized in Table 5. Most resources used are either open-source or institutionally provided with some minor costs incurring with training and server hosting. As a result, a variable total cost of approximately $30/month is expected to be incurred. Over an estimated 6-month timeline, this equates to a $180 total cost. A detailed breakdown is shown below.

Table 5: Estimated project costs and respective use-cases.

Item	Estimated Cost (CAD)	Notes
GPU access (Colab Pro / lab)	$0–20/month	Free Google provided online GPUs can be used
Data storage & hosting	$0-10/month	Online server hosting/local server hosting
Browser extension SDKs	$0	Open source

The provided design, testing strategy and resource plan outline a clear and realistic pathway for the development of this project. By using this modular development approach, the project can be delivered effectively within the set timeline and budget. This structured approach to this project ensures each component is well supported and integrated towards achieving the project's overall objectives.

Conclusion

In conclusion, the project aims to be a full-stack, web-based application designed to automatically identify and classify healthcare-related misinformation through the use of transformer-based NLP models. This proposal outlined the motivation for developing an accessible and ethical solution to help mitigate the spread of medical misinformation. The project's objectives emphasize delivering a functional website and possible browser extension that will allow users to input text or article links and receive credibility results based off the model's classification along with confidence scores and highlighted explanations to back up the results.

The report described the system’s architecture, consisting of an interactive web interface built with React/Next.js, a secure backend implemented in Node.js, and an AI pipeline developed using Python and fine-tuned BERT/RoBERTa models for healthcare misinformation detection. The functional description detailed how data flows starting from the user’s input through the backend and then a computed result from model’s classification that is visualized in a user-friendly way to ensure interpretability. The design and production plan follows key milestones to help ensure steady progress amongst the team in data processing, model training, backend integration, and interface design. Testing and evaluation procedures were defined to validate the model’s accuracy and overall system reliability. Additionally, resource planning demonstrated that the project is both technically feasible within the time constraints and financially sustainable using open-source tools and third-party resources.

In short, this project embodies the past four years of engineering values taught through classes, team projects, and internship experience. The values of responsibility, transparency, and innovation that engineers must possess. By combining efficient software design with AI infusion, the proposed system will contribute to improving digital literacy in the healthcare sector, reducing misinformation and fostering public trust. Through this project, the team aims to build not only a technical solution need in today’s times, but also a meaningful tool that empowers individuals to navigate online healthcare information with greater confidence and safety.

References

[1] Professional Engineers Ontario, "Code of Ethics," [Online]. Available: https://www.peo.on.ca/licence-holders/code-ethics.