CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

IV Trimester MCA

Specialization Project

Department of Computer Science

AI-powered News Summarizer and Bias Detector

By

Abhinandan Karanth(2447101)

Glory Reji (2447120)

Saumya Dwivedi (2447149)

Under the guidance of

Dr Vaidhehi V

TABLE OF CONTENTS

Acknowledgement

I would like to express my sincere gratitude to all those who have supported and guided me throughout the successful completion of this project report titled "AI-Powered News Summarizer and Bias Detector."

First and foremost, I am deeply thankful to our guide, Dr. Vaidhehi V, for their invaluable encouragement, insightful suggestions, and constant support throughout the course of this project. Their expertise and guidance have been instrumental in shaping the project from its initial idea to a structured plan.

I would also like to extend my appreciation to all the faculty members of the Department of Computer Science, Christ University, for providing a strong academic foundation and for fostering a research-oriented environment.

A special thanks to my project teammates Abhinandan Karanth and Saumya Dwivedi, whose dedication, teamwork, and technical contributions played a vital role in designing and planning the various components of this system.

Last but not least, I am grateful to my family and friends for their constant motivation, patience, and moral support during every stage of this work.

This project marks an important learning milestone, and I am proud to have undertaken it as part of my academic journey.

Abhinandan karanth K U
IV MCA-A
Christ University

ABSTRACT

This project presents the design and development plan for an AI-powered web application that enhances how users consume and evaluate news content by offering automatic summarization and political bias detection. With the growing volume of online articles and the rise of misinformation, there is an urgent need for tools that support quick comprehension and critical analysis of media. The proposed system, titled *BiasLens*, takes raw news content—either pasted as text or extracted from URLs—and applies advanced Natural Language Processing (NLP) techniques to generate concise summaries and classify the political or ideological bias of the content into categories such as left, center, or right. By streamlining the information overload and revealing hidden bias patterns, the system aims to encourage balanced media consumption.

The application architecture integrates pre-trained transformer models such as BART for abstractive summarization and RoBERTa for zero-shot bias classification, both powered by the Hugging Face ecosystem. A web-based interface allows users to interact with the system, displaying results through a visually rich UI that includes bias meters, sentiment indicators, and potential explanations for the bias label. The backend leverages Flask or Node.js for API management, while the frontend is developed using React or Jetpack Compose, ensuring a responsive and intuitive experience. Future enhancements include multi-source comparison, user feedback loops, and multilingual support, making BiasLens a promising tool for education, journalism, and media literacy research.

CHAPTER 1: INTRODUCTION

1.1 Background of the Project:

In today's digital era, people are bombarded with a massive amount of news from various sources. While this gives access to diverse perspectives, it also increases the chances of encountering misinformation and biased narratives. With the growing concern about media bias and fake news, it is essential to develop tools that help users quickly summarize news content and assess its neutrality. This project addresses the need for intelligent systems that can aid in media literacy by offering both summarization and bias detection in one platform.

1.2 Objectives:

- To develop an AI-powered application that summarizes news articles.
- To detect political/media bias in the article content.
- To provide sentiment analysis of the text.
- To create a user-friendly web interface for interaction.
- To visualize bias and tone using intuitive UI elements.

1.3 Purpose, Scope and Applicability:

Purpose:

- To reduce time spent on reading long articles.
- To help users become aware of bias in news sources.
- To encourage critical reading and balanced media consumption.

Scope:

- Assumes access to clean news content either as raw text or via URL scraping.
- Uses pre-trained NLP models for summarization and bias classification.
- Addresses bias detection within political or ideological contexts.
- Main functions include summarization, bias detection, tone analysis, and multi-source comparison.

Applicability:

- Directly useful for students, educators, journalists, and researchers.
- Indirectly contributes to digital literacy and responsible media usage in society.

1.4 Overview of the Report:

The report is divided into chapters detailing the system analysis, requirements, design, implementation, testing, and conclusion. Each chapter documents specific aspects of the project development lifecycle.

CHAPTER 2: SYSTEM ANALYSIS AND REQUIREMENTS

This chapter provides an analysis of the existing systems, outlines limitations, and explains the proposed system in detail.

2.1 Existing System:

Currently, several news aggregator platforms like Google News and Flipboard provide article previews or basic summaries using extractive techniques. Likewise, sentiment analysis tools are widely used in marketing and media monitoring platforms to gauge emotional tone. However, these systems typically operate independently, offering either summarization or sentiment detection — not both. Bias detection, when available, is usually based on static, source-level labels rather than real-time content analysis.

Few existing platforms offer a combined solution for summarization and political bias detection at the article level. Most do not provide explainability, meaning users cannot see what influenced the bias classification. Real-time multi-source comparison is also rare, limiting the user's ability to evaluate diverse perspectives on the same topic.

2.2 Limitations of the Existing System:

- Lack of explainability in bias detection.
- No combined solution for summarization, bias, and sentiment.
- No user interaction or customization.
- Limited scope in comparing articles across sources.

2.3 Proposed System:

This project proposes the development of an AI-powered platform that integrates news summarization, political bias detection, and sentiment analysis into a single, user-friendly application. Unlike existing tools, the system is designed to operate on article-level content in real time, using transformer-based models to extract key information and assess bias. It will allow users to input raw text or a news URL, and receive a clear, concise summary along with visual indicators showing the political leaning and emotional tone of the article.

The proposed system is modular in design and consists of four core components:

1. **Article Summarization** – Generates brief yet meaningful summaries using models like BART.

2. **Bias Detection** – Classifies the political orientation (e.g., left, center, right) using models like RoBERTa.

3. **Sentiment Analysis** – Identifies the tone of the article as positive, negative, or neutral.

4. **UI Visualization and User Interaction** – Presents results through a clean, intuitive interface with visual meters, tags, and explanations.

2.4 Benefits of the Proposed System:

- Reduces reading time.
- Encourages awareness of media bias.
- Provides sentiment context to the summary.
- Offers explainable results with confidence scores.

2.5 Features of the Proposed System:

- Paste or URL input for news articles.
- Summarization using transformer models.
- Bias detection with visual indicator (slider/meter).
- Sentiment tagging and explanation.
- Responsive frontend interface.

2.6 System Requirements Specification:

2.6.1 User Characteristics:

- General users: Read and interpret news.
- Journalists: Analyze framing and sentiment.
- Researchers: Study bias patterns.

2.6.2 Software and Hardware Requirements:

Software Requirements:

- OS: Windows 10 or Ubuntu 20+
- IDE: VS Code / PyCharm
- Backend: Python 3.10, Node.js
- Libraries: Flask, Hugging Face Transformers, React, Axios

Hardware Requirements:

- Processor: i5 or above
- RAM: Minimum 8GB
- Disk: Minimum 256GB SSD
- GPU (optional): For faster model inference

2.6.3 Constraints:

- Requires internet connectivity for API-based article scraping.
- Dependent on pre-trained models' limitations.
- Moderate latency in large articles.

2.6.4 Functional Requirements:

Input	Process	Output
News text or URL	Summarization	Article summary
News text	Bias Classification	Left, Center, Right
News text	Sentiment Analysis	Positive/Negative/Neutral

2.6.5 Non-Functional Requirements:

- Usability: Simple, intuitive UI.
- Reliability: Consistent results across articles.
- Performance: Near real-time processing.
- Maintainability: Modular codebase.
- Security: Basic input validation.

CHAPTER 3: SYSTEM DESIGN

3.1 System Architecture

The proposed system follows a multi-layered modular architecture that separates responsibilities across distinct tiers, improving scalability, maintainability, and clarity of the application. The architecture comprises components for data ingestion, NLP processing, scoring, storage, and user interaction, all orchestrated via a centralized API layer.

1. News Ingestion Layer

News content is collected from various sources such as RSS feeds, News APIs, Web scraping, and even social media. This ensures the system is capable of analyzing a diverse range of real-world data.

2. Data Processing Pipeline

This pipeline includes:

- Content Extraction and Preprocessing (removing HTML tags, formatting, etc.)
- Deduplication to avoid redundant article processing
- Language Detection to filter non-English content

A Source Verification module cross-references data with a credibility database and optional fact-check APIs to evaluate trustworthiness.

3. AI/ML Processing Layer

This is the core of the BiasLens system. It includes:

- Summarization Module using models like BART
- Bias Detection using classifiers such as RoBERTa or DistilBERT

- Sentiment Analysis and optional extensions like Entity Recognition and Topic Classification

These models process the article content and output structured insights.

4. Data Storage Layer

The system uses multiple storage types:

- Raw Articles DB to keep original article text
- Processed Data DB for summaries and metadata
- Vector DB (for embeddings) to support future search-based comparisons
- Cache Layer using Redis to improve real-time performance

5. API Layer

The API layer (built with Flask/Node.js) exposes endpoints to the frontend and external services. It handles:

- RESTful API calls
- WebSocket support for real-time updates
- Authentication and authorization for secure access

6. User Interface Layer

The frontend is developed using React or Jetpack Compose, offering:

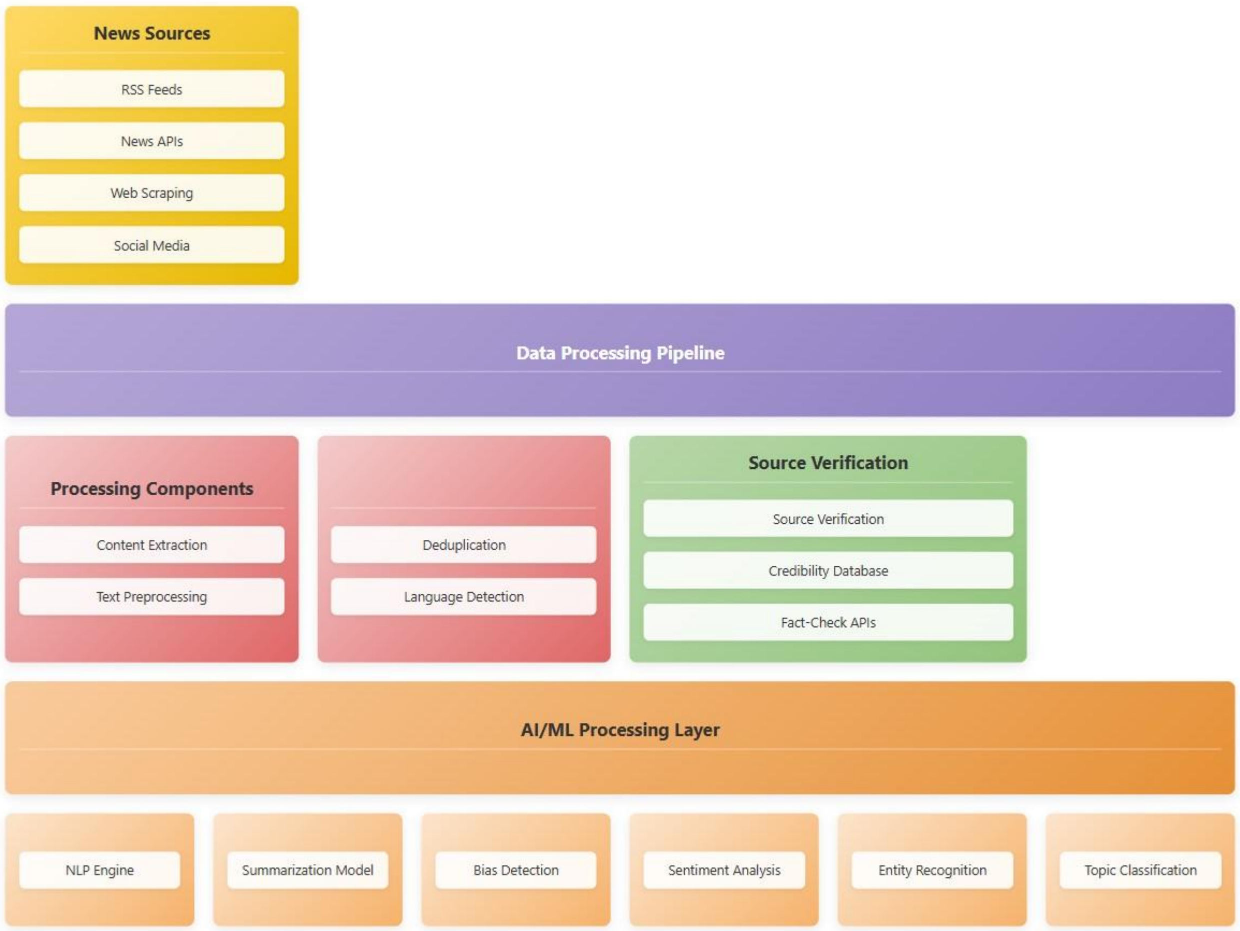
- A web dashboard for users to input URLs/text and view results
- Visual components like bias meters, sentiment tags, and explainable highlights
- An admin panel for monitoring usage and fine-tuning models

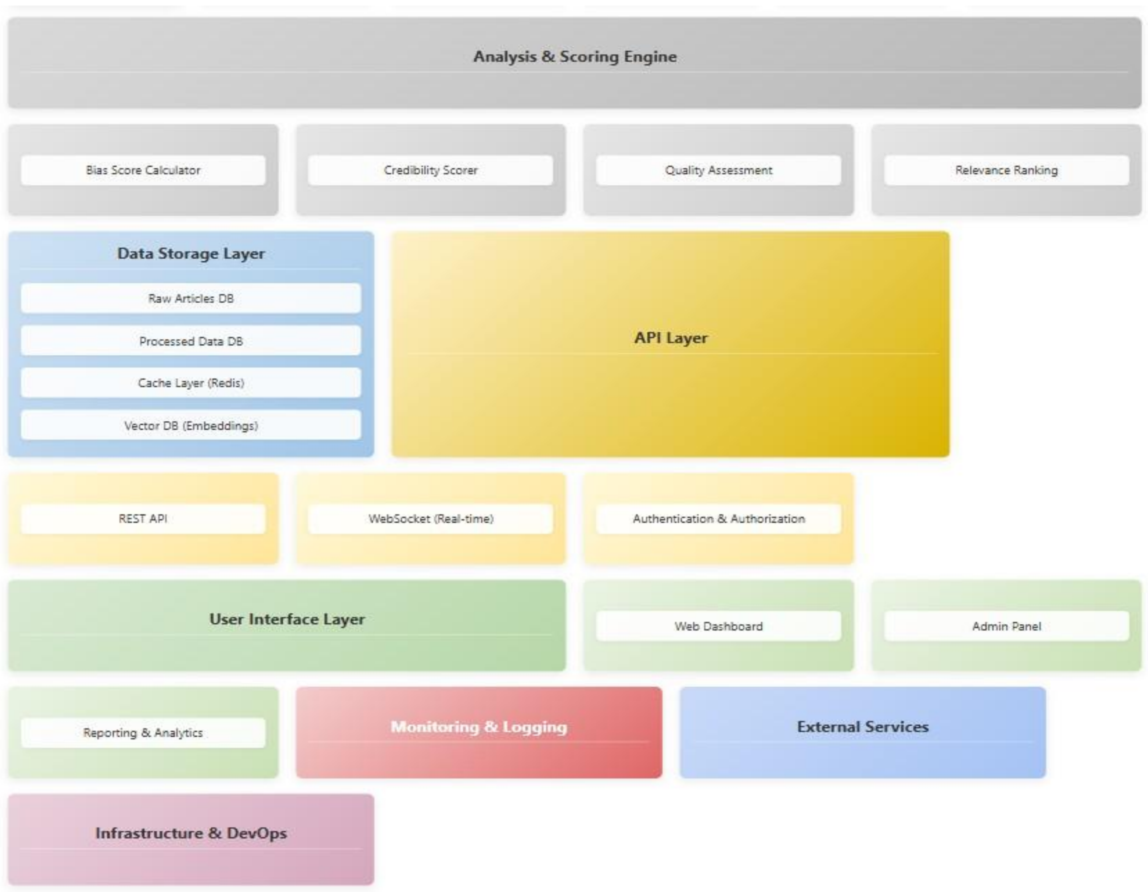
7. Monitoring & Logging / Analytics

Integrated modules track system performance, user activity, and model behavior, which is vital for continuous improvement and error tracing.

8. Infrastructure & DevOps

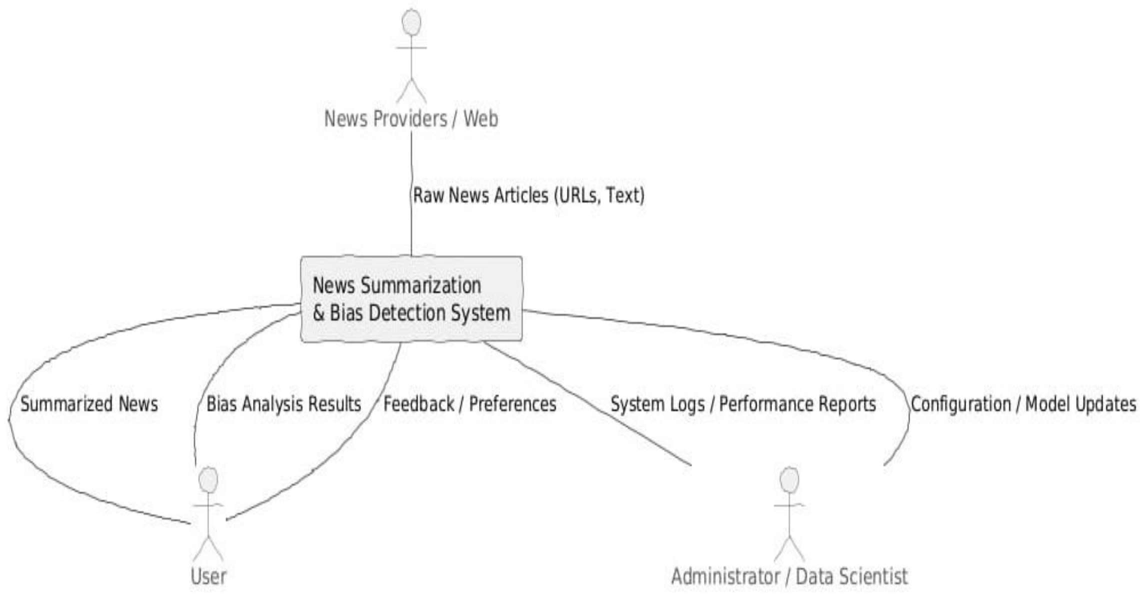
The system is deployable via containerized services (e.g., Docker, Kubernetes), supporting continuous integration/deployment, monitoring, and cloud scalability.



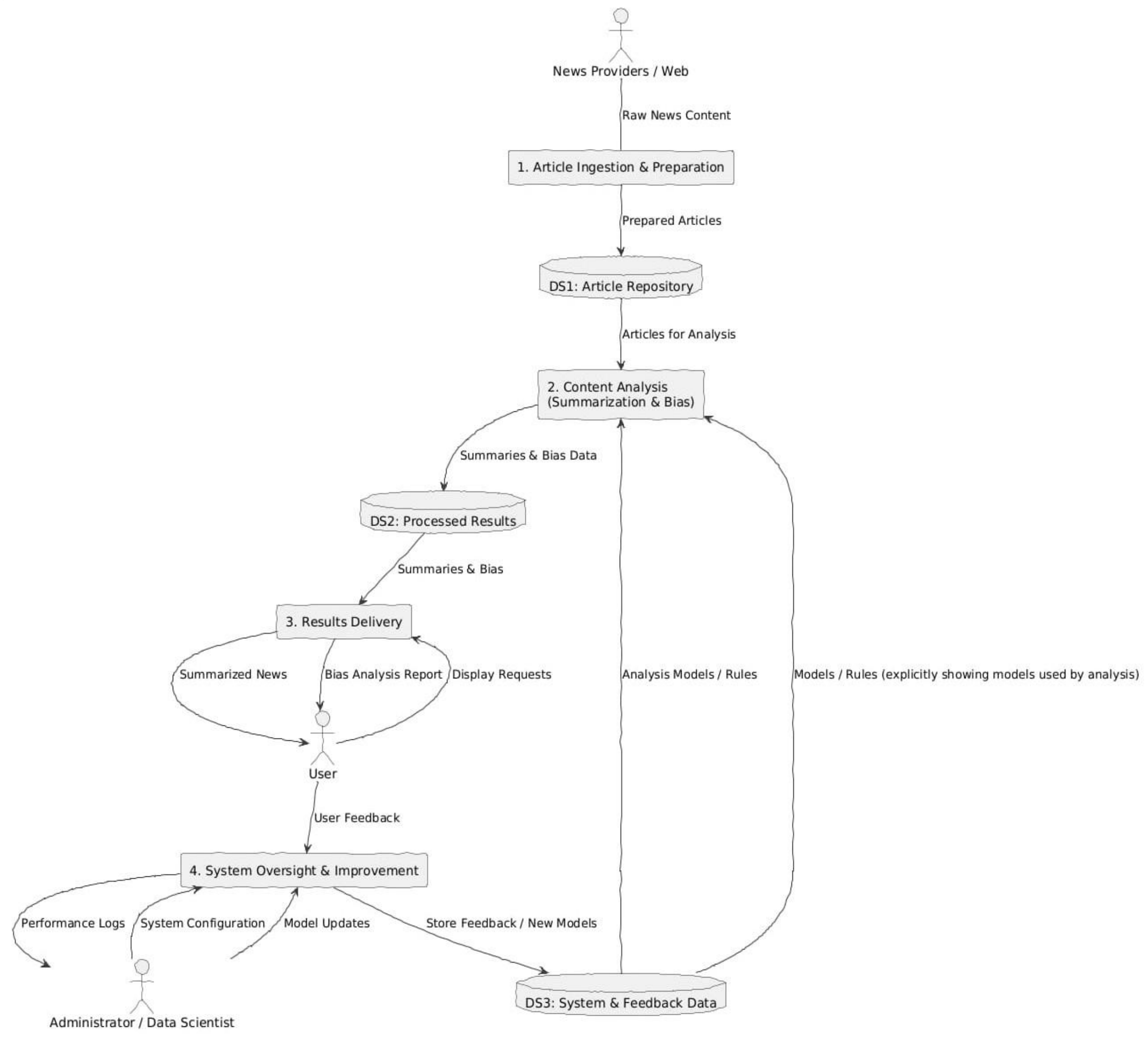


3.2 Data Flow Diagram

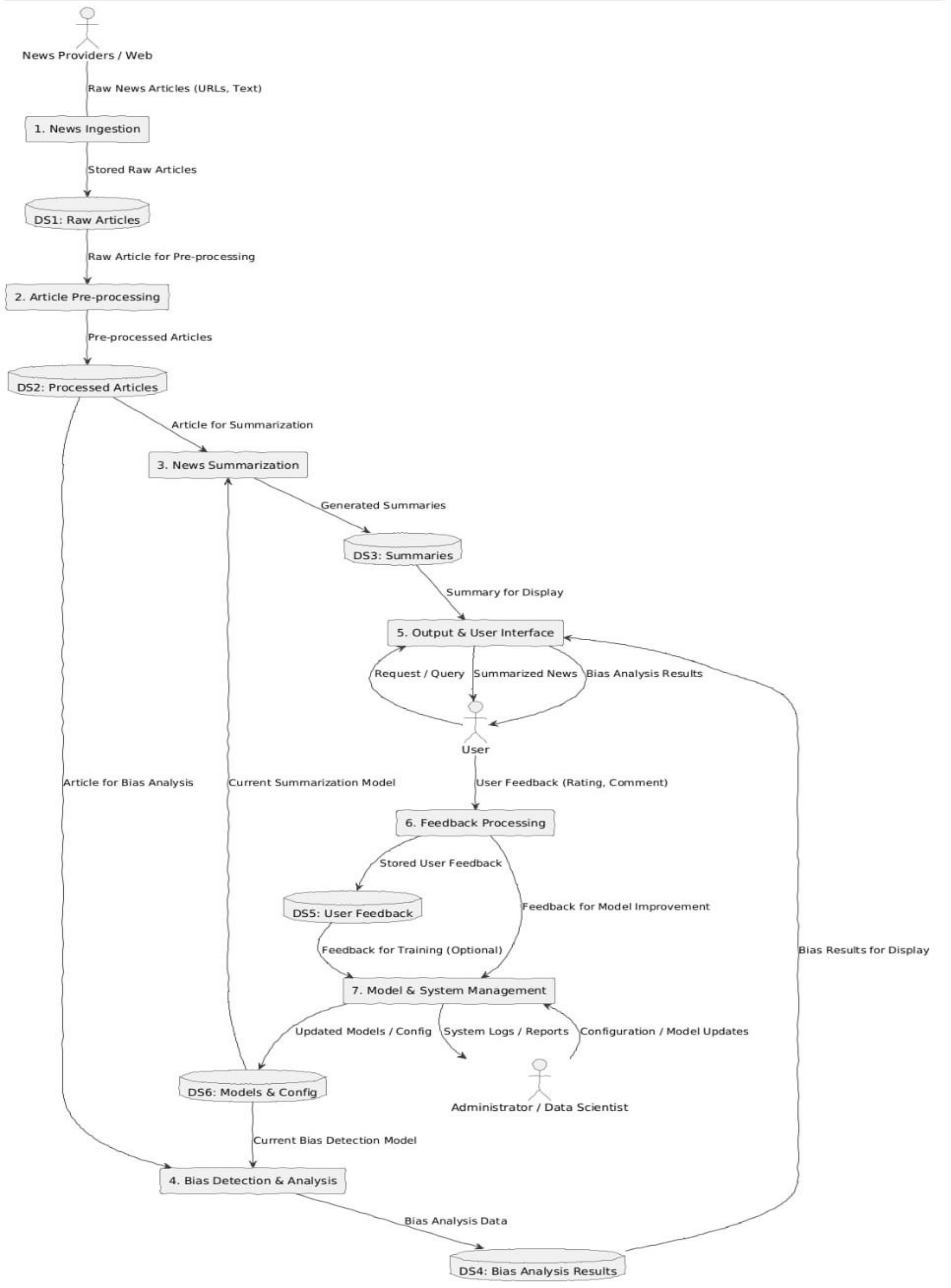
DFD Level 0


News Providers / Web
Raw News Articles (URLs, Text)
News Summarization & Bias Detection System
Summarized News
Bias Analysis Results
Feedback / Preferences
System Logs / Performance Reports
Configuration / Model Updates
User
Administrator / Data Scientist

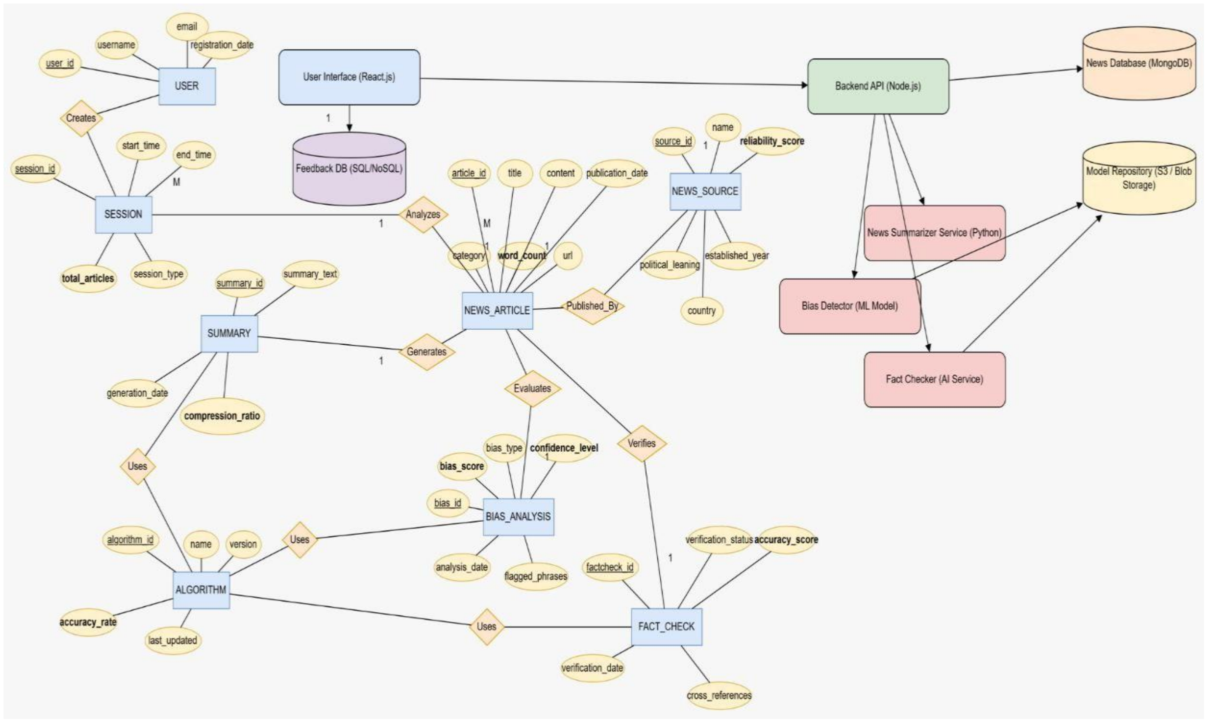
DFD Level 1


News Providers / Web
Raw News Content
1. Article Ingestion & Preparation
Prepared Articles
DS1: Article Repository
Articles for Analysis
2. Content Analysis (Summarization & Bias)
Summaries & Bias Data
DS2: Processed Results
Summaries & Bias
3. Results Delivery
Summarized News
Bias Analysis Report
Display Requests
Analysis Models / Rules
Models / Rules (explicitly showing models used by analysis)
User
User Feedback
4. System Oversight & Improvement
Performance Logs
System Configuration
Model Updates
Store Feedback / New Models
Administrator / Data Scientist
DS3: System & Feedback Data

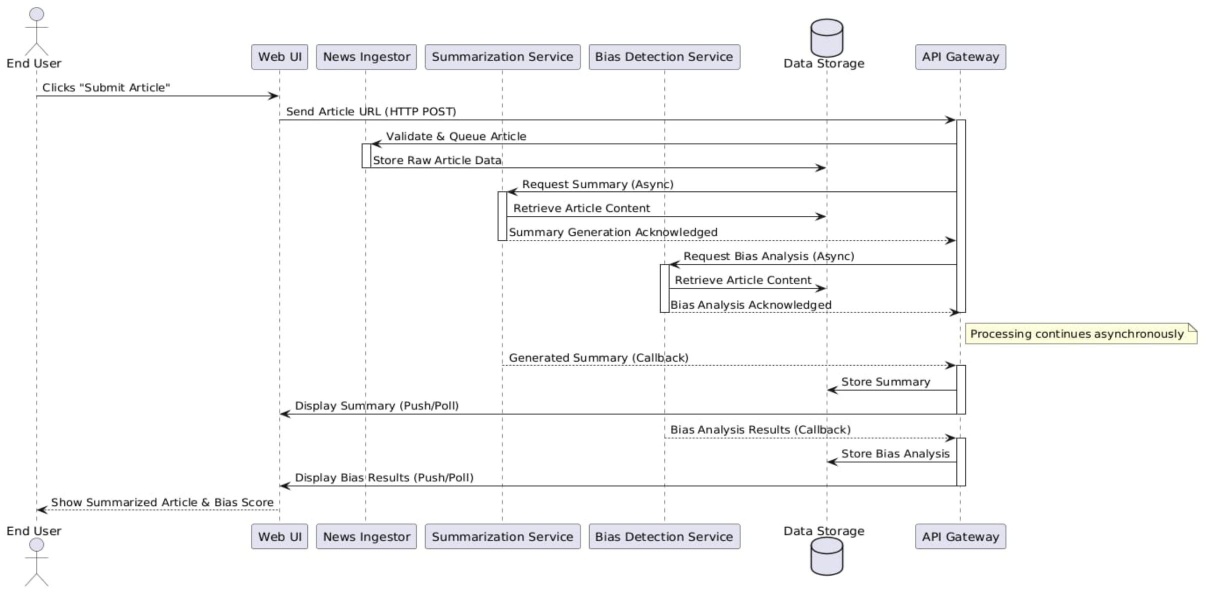
DFD Level 2


News Providers / Web
Raw News Articles (URLs, Text)
1. News Ingestion
Stored Raw Articles
DS1: Raw Articles
Raw Article for Pre-processing
2. Article Pre-processing
Pre-processed Articles
DS2: Processed Articles
Article for Summarization
3. News Summarization
Generated Summaries
DS3: Summaries
Summary for Display
5. Output & User Interface
Request / Query
Summarized News
Bias Analysis Results
User
Article for Bias Analysis
Current Summarization Model
User Feedback (Rating, Comment)
6. Feedback Processing
Stored User Feedback
DS5: User Feedback
Feedback for Model Improvement
Feedback for Training (Optional)
Bias Results for Display
7. Model & System Management
Updated Models / Config
System Logs / Reports
Configuration / Model Updates
DS6: Models & Config
Administrator / Data Scientist
Current Bias Detection Model
4. Bias Detection & Analysis
Bias Analysis Data
DS4: Bias Analysis Results

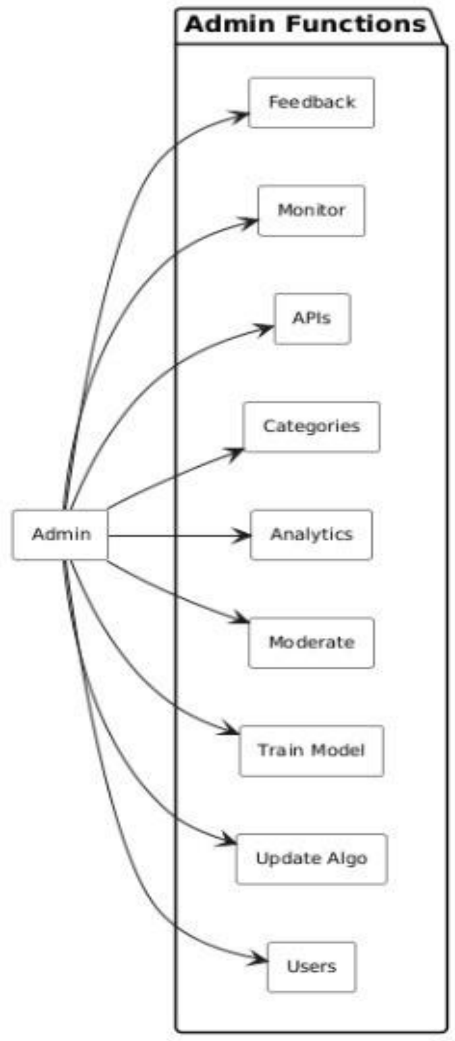
3.3 ER Diagram

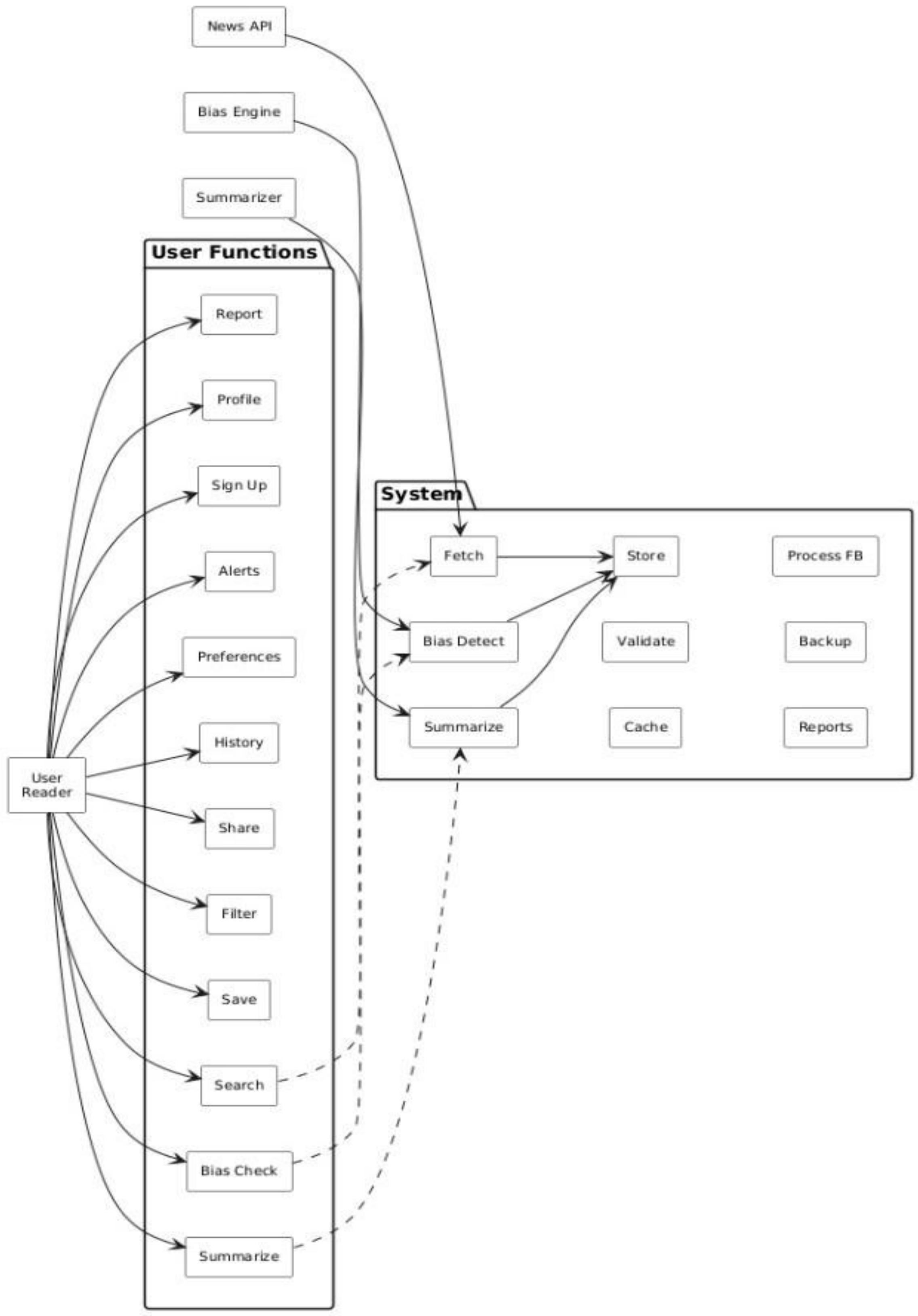


3.4 Sequence Diagram


End User
Web UI
News Ingestor
Summarization Service
Bias Detection Service
Data Storage
API Gateway
Clicks "Submit Article"
Send Article URL (HTTP POST)
Validate & Queue Article
Store Raw Article Data
Request Summary (Async)
Retrieve Article Content
Summary Generation Acknowledged
Request Bias Analysis (Async)
Retrieve Article Content
Bias Analysis Acknowledged
Processing continues asynchronously
Generated Summary (Callback)
Store Summary
Display Summary (Push/Poll)
Bias Analysis Results (Callback)
Store Bias Analysis
Display Bias Results (Push/Poll)
Show Summarized Article & Bias Score

3.5 Use Case Diagram




News API
Bias Engine
Summarizer
User Functions
Report
Profile
Sign Up
Alerts
Preferences
History
Share
Filter
Save
Search
Bias Check
Summarize
User Reader
System
Fetch
Store
Process FB
Bias Detect
Validate
Backup
Summarize
Cache
Reports

3.6 Dataset and Model

3.6.1 Dataset Used: Newsroom Dataset

The Newsroom dataset contains over 1.3 million article-summary pairs from 38 reputable news publishers. It supports multiple summarization strategies (extractive, mixed, abstractive), making it highly valuable for training and evaluating summarization models.

3.6.2 Model Used: BART (facebook/bart-large-cnn)

The BART model, specifically the `facebook/bart-large-cnn` version, is used for abstractive summarization. It is a transformer-based sequence-to-sequence model that combines the benefits of BERT (encoder) and GPT (decoder), making it suitable for generating high-quality human-like summaries.

3.6.3 Implementation Strategy

- The model is loaded from the Hugging Face Transformers library.
- The input article (either pasted or scraped from a URL) is tokenized and passed through the model.
- The generated summary is displayed on the frontend using a result card or collapsible UI block.
- This model is used in inference mode without fine-tuning to reduce compute requirements.

3.7. Bias Detection Module

3.7.1 Dataset Used: AllSides Political Bias Dataset

The AllSides dataset contains thousands of news articles labeled with political orientation: Left, Center, or Right. These labels are assigned by expert annotators, making the dataset highly reliable for bias classification tasks.

3.7.2 Model Used: RoBERTa-base (Fine-tuned)

We use RoBERTa-base, a robust transformer-based language model well-suited for classification tasks. It is fine-tuned on the AllSides dataset to detect ideological bias in article text.

3.7.3 Implementation Strategy

- RoBERTa is fine-tuned using the labeled AllSides dataset.
- The model is deployed via a Flask or FastAPI backend.
- On receiving an article, the model returns a predicted bias label and a confidence score.
- The result is visually rendered as a Bias Meter (Left ↔ Center ↔ Right) on the UI.

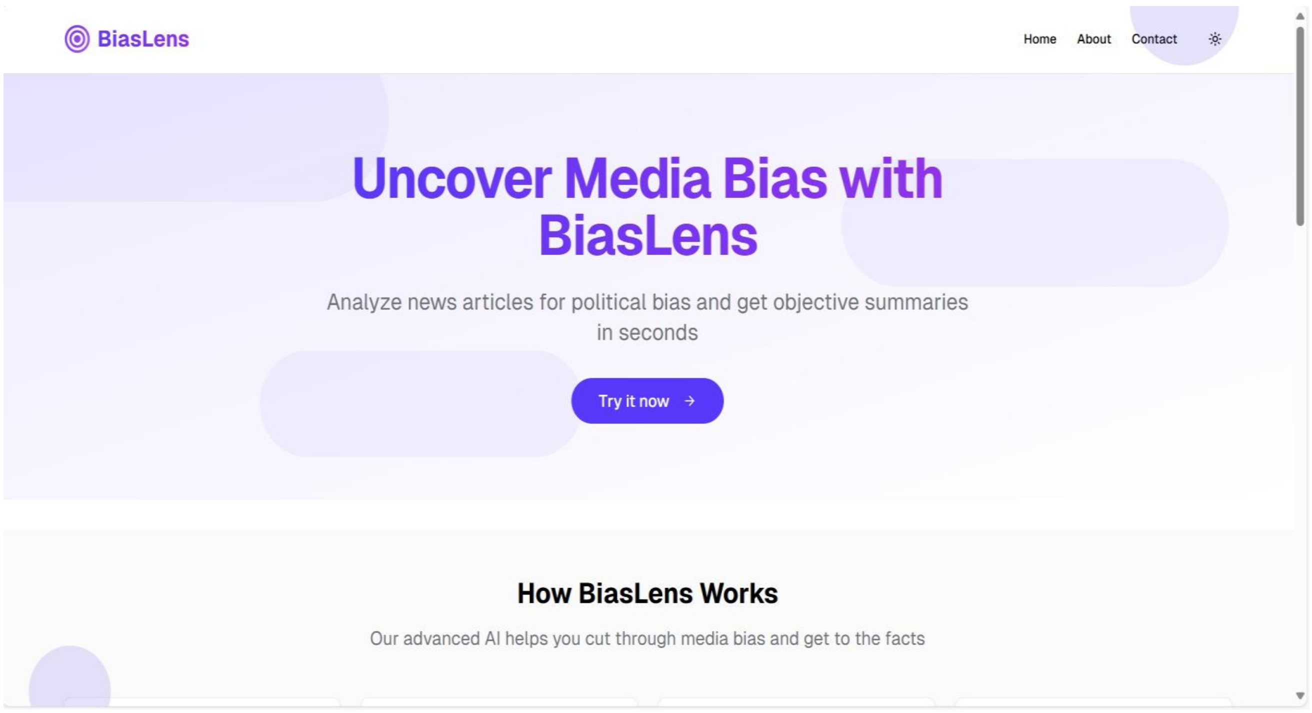
3.8 System Integration and Output Flow

Once both models are integrated into the backend service:

1. The article input is processed and passed to the summarizer.
2. The same article is sent to the bias classifier.
3. Results from both modules are returned as a structured JSON response.
4. The frontend interface renders:
    - The generated summary
    - The political bias label with a visual meter
    - Optional sentiment analysis

This modular pipeline ensures efficiency, clarity, and real-world usability while maintaining scalability for future features like multi-source comparison and multilingual support.

3.9 User Interface Design

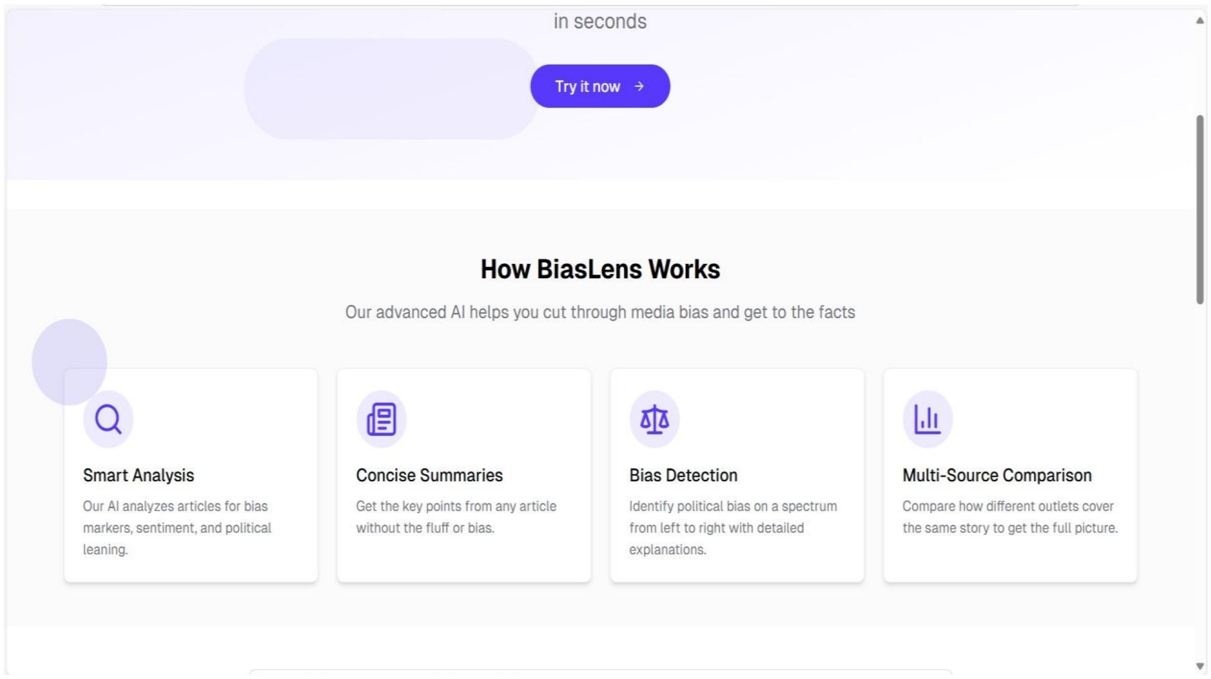


Uncover Media Bias with BiasLens

Analyze news articles for political bias and get objective summaries in seconds

How BiasLens Works

Our advanced AI helps you cut through media bias and get to the facts



How BiasLens Works

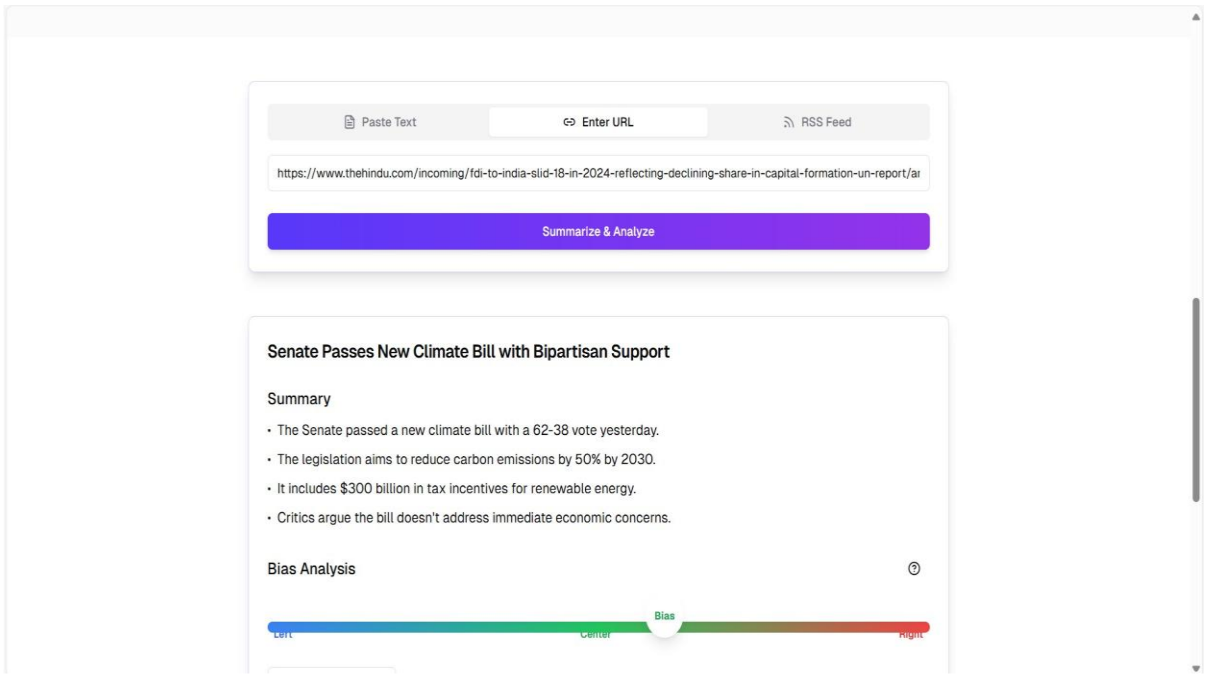
Our advanced AI helps you cut through media bias and get to the facts

Smart Analysis
Our AI analyzes articles for bias markers, sentiment, and political leaning.

Concise Summaries
Get the key points from any article without the fluff or bias.

Bias Detection
Identify political bias on a spectrum from left to right with detailed explanations.

Multi-Source Comparison
Compare how different outlets cover the same story to get the full picture.



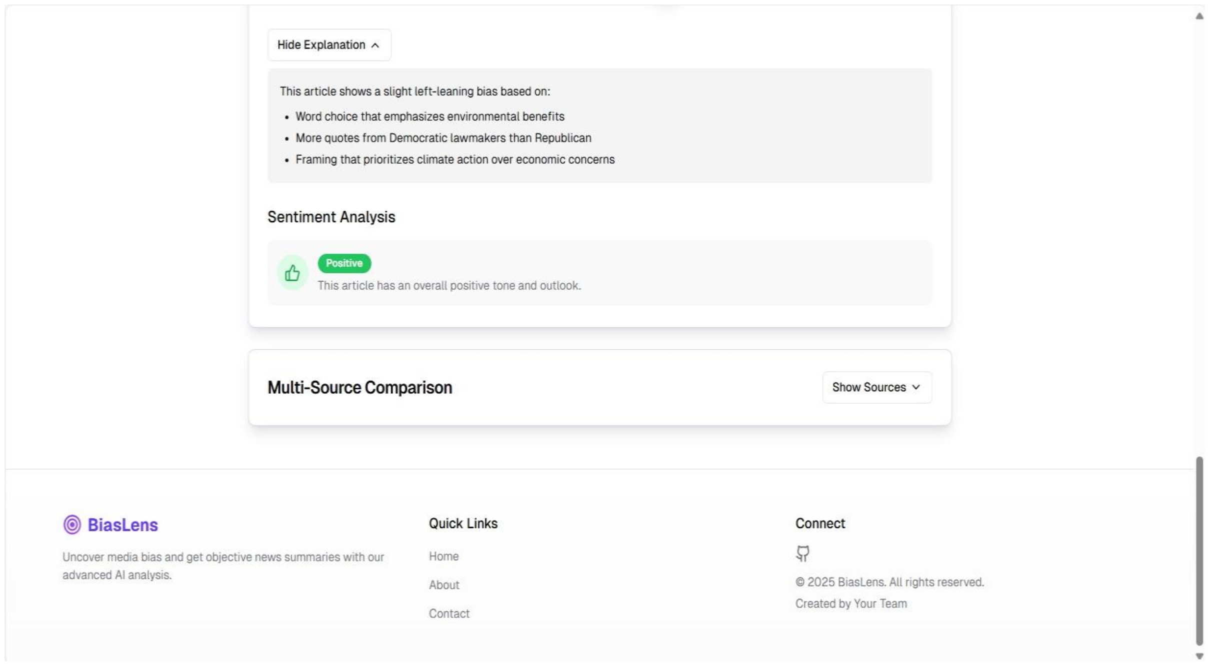
https://www.thehindu.com/incoming/fdi-to-india-slid-18-in-2024-reflecting-declining-share-in-capital-formation-un-report/ar

Senate Passes New Climate Bill with Bipartisan Support

Summary

- The Senate passed a new climate bill with a 62-38 vote yesterday.
- The legislation aims to reduce carbon emissions by 50% by 2030.
- It includes $300 billion in tax incentives for renewable energy.
- Critics argue the bill doesn't address immediate economic concerns.

Bias Analysis



Hide Explanation

This article shows a slight left-leaning bias based on:

- Word choice that emphasizes environmental benefits
- More quotes from Democratic lawmakers than Republican
- Framing that prioritizes climate action over economic concerns

Sentiment Analysis

Positive

This article has an overall positive tone and outlook.

Multi-Source Comparison

Show Sources

BiasLens

Uncover media bias and get objective news summaries with our advanced AI analysis.

Quick Links

Home

About

Contact

Connect

© 2025 BiasLens. All rights reserved.

Created by Your Team

CHAPTER 4: CONCLUSION

This project report outlines the design and development plan for an AI-powered system aimed at simplifying news consumption through automatic summarization and political bias detection. The system is intended to help users make informed decisions by providing concise, meaningful summaries of articles along with an assessment of their ideological stance.

The initial phase involved requirement analysis, architectural design, module breakdown, and interface planning. Detailed documentation of the system's objectives, scope, architecture, data flow, and proposed modules has been completed. We also identified the core tools and models to be used, such as BART for summarization and RoBERTa for bias classification, and have planned for the integration of these models through a modular backend.

Future Scope

In the upcoming stages of the project, we will focus on implementing the backend logic, training or fine-tuning the models (as needed), and building the frontend interface to visualize results. After deployment of the basic system, future enhancements may include:

- Adding multilingual support
- Improving explainability of model outputs
- Enabling real-time article comparisons from multiple sources
- Expanding to a browser extension or mobile version.