NewsInsights

Capstone Project Report

MID SEMESTER EVALUATION

Submitted by:

(102216134) Nandan Garg

(102216081) Sakaar Sen

(102216019) Ridham Uppal

(102216032) Palak Mahajan

(102216103) Manav Singhal

BE Fourth Year, CoSE

CPG No: 92

Under the Mentorship of

Dr. Rinkle Rani

Professor

Department of Computer Science and Engineering
Thapar Institute of Engineering and Technology, Patiala

August 2025

ABSTRACT

In today's digital era, news consumption is challenged by overwhelming content, political bias, and misinformation. Traditional news aggregators, while offering personalization, often fail to ensure credibility, balanced perspectives, or media transparency. NewsInsights is an AI-powered news aggregation platform that leverages NLP (NLP) and ML (ML) to deliver bias-aware, concise, and personalized news updates. The system integrates AI-driven summarization, sentiment analysis, keyword extraction, and political bias detection, enabling users to critically evaluate news rather than passively consume it.

Unlike conventional platforms such as Google News or Flipboard, IndiaInsights introduces real-time multi-source comparisons, bias scoring, and vector-based recommendation engines to break echo chambers and promote fact-driven consumption. The platform further employs custom ML models for dynamic bias analysis, going beyond pre-labeled datasets to ensure adaptability and transparency. By combining personalization with bias-aware insights, trend analysis, and keyword highlighting, IndiaInsights empowers users to make informed decisions while enhancing media literacy and democratic awareness.

Ultimately, this project addresses the critical challenges of misinformation, echo chambers, and biased reporting, building a trustworthy, user-centric, and AI-enhanced digital news ecosystem for the modern reader.

DECLARATION

We hereby declare that the design principles and working prototype model of the project entitled News Insights is an authentic record of our own work carried out in the Computer Science and Engineering Department, TIET, Patiala, under the guidance of Dr. Rinkle Rani during 6th semester (2025).

Date: 22 Aug 2025

Roll No.	Name	Signature
102216019	Ridham Uppal	*Ridham Uppal*
102216032	Palak Mahajan	*Palak Mahajan*
102216081	Sakaar Sen	*Sakaar Sen*
102216103	Manav Singhal	*Manav Singhal*
102216134	Nandan Garg	*Nandan Garg*

Counter Signed By:

Faculty Mentor: Rinkle 22/08/25

Dr. Rinkle Rani

Professor

DCSE,

TIET, Patiala

ACKNOWLEDGEMENT

We would like to express our thanks to our mentor Dr. Rinkle Rani. She has been of great help in our venture and an indispensable resource of technical knowledge.She is truly an amazing mentor to have.

We are also thankful to Dr. Neeraj Kumar, Head, Computer Science and Engineering Department, the entire faculty and staff of the Computer Science and Engineering Department, and also our friends who devoted their valuable time and helped us in all possible ways towards successful completion of this project. We thank all those who have contributed either directly or indirectly towards this project.

Lastly, we would also like to thank our families for their unyielding love and encouragement.

They always wanted the best for us and we admire their determination and sacrifice.

Date: 22 Aug 2025

Roll No.	**Name**	**Signature**
102216019	Ridham Uppal	*Ridham Uppal*
102216032	Palak Mahajan	*Palak Mahajan*
102216081	Sakaar Sen	*Sakaar Sen*
102216103	Manav Singhal	*Manav Singhal*
102216134	Nandan Garg	*Nandan Garg*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVATIONS

Abbrevation	Meaning	.
AI	Artificial Intelligence	
ML	Machine Learning	
NLP	Natural Language Processing	
LLM	Large Language Model	
BERT	Bidirectional Encoder Representations from Transformers	
RoBERTa	Robustly Optimized BERT Approach	
DeBERTa	Decoding-Enhanced BERT with Disentangled Attention	
RNN	Recurrent Neural Network	
LSTM	Long Short-Term Memory	
UI	User Interface	
API	Application Programming Interface	
DB	Database	
ER	Entity–Relationship	
RSS	Really Simple Syndication	
WBS	Work Breakdown Structure	
DRF	Django REST Framework	

INTRODUCTION

1.1 Project Overview

In today's digital age, news consumption is at an all-time high, yet the landscape is increasingly fragmented. With countless media outlets, each carrying their own political leanings and editorial agendas, readers often struggle to identify unbiased and trustworthy information. Social media further amplifies this problem by prioritizing engagement over accuracy, resulting in echo chambers and misinformation. Against this backdrop, there is a growing demand for platforms that not only consolidate diverse news sources but also present them in a fair, transparent, and accessible manner.

The motivation behind developing NewsInsights stems from the need to bridge this gap between overwhelming information availability and meaningful, unbiased understanding. Many readers today face "information overload" with lengthy articles, repetitive coverage, and sensationalized headlines that make it difficult to quickly grasp the essence of current events.

Another key motivation is personalization without bias reinforcement. While most platforms push content aligned with existing user preferences, this often narrows perspectives. NewsInsights instead aims to provide readers with personalized yet diverse content by giving them relevant updates while ensuring they are not trapped in one-sided narratives. Additionally, features like bookmarks and sharing enhance the reading experience, making it not only informative but also convenient and user-friendly.

NewsInsights is an AI-powered unbiased news aggregator designed to address the challenges of media bias, information overload, and fragmented content consumption. The platform provides readers with a consolidated view of current events by collecting articles from multiple sources, analyzing their political bias, summarizing the content, and recommending personalized yet balanced information. Unlike conventional aggregators that merely display articles, NewsInsights emphasizes fairness, transparency, and clarity, enabling users to engage with news critically rather than passively.

At its core, the system leverages NLP and ML models to classify news into left, right, or neutral political orientations. This classification ensures that users are aware of the potential biases in reporting. In addition, the platform generates concise AI-driven summaries, reducing the time required to comprehend complex articles and making news more accessible to a wider audience.

Another defining aspect of NewsInsights is its multi-layered analytical capability. At its core, the platform not only detects political bias and generates concise summaries, but also incorporates sentiment analysis to reflect public opinion drawn from social media discussions. This helps readers understand how issues are perceived beyond traditional reporting. Additionally, features like keyword highlighting support quick navigation through articles, giving users a clearer sense of the themes being discussed. By combining these layers, NewsInsights transforms raw information into structured insights, empowering readers to make informed judgments.

The system architecture follows a modular design, where the backend handles data ingestion, analysis, and processing, while the frontend ensures an interactive and seamless user experience. The platform's recommendation engine focuses on delivering a personalized reading experience by suggesting articles that match user interests while also prioritizing recent and unread content. This ensures that users stay updated with the latest developments without encountering repetitive information.

Key Objectives

1. To build an unbiased and transparent news aggregation platform.
2. To detect and classify political bias in articles as Left, Right, or Neutral.
3. To provide AI-generated summaries and for reducing information overload.
4. To highlight important keywords and entities for quick context.
5. To integrate sentiment analysis to improve insightful news.
6. To design a news recommendation system.

Core Features

1. News Aggregation: Collects articles from multiple trusted sources.
2. Political Bias Detection: Machine learning model that classifies news into Left, Right, or Neutral categories.
3. AI Summaries: NLP-driven concise summaries of long articles for faster understanding.
4. Keyword Highlighting: Automatic extraction and emphasis of key terms for quick navigation.
5. Sentiment Analysis: Identifies the tone (positive, negative, or neutral) within news articles.
6. Recommendation Engine: Suggests articles tailored to user interests and prioritizes recent unread content.
7. Bookmarks: Enables users to save and revisit important articles for later reading.

Technologies Used

1. Backend: Django & Django REST Framework (API services)
2. Task Processing: Celery + Redis for background processing
3. Database: PostgreSQL for structured data storage
4. Frontend: Next.js for user interaction and UI design
5. NLP/ML: Hugging Face transformers, custom ML models for bias classification, summarization, and sentiment analysis
6. Deployment: Docker for containerization.

1.2 Need Analysis

In today’s digital landscape, news consumption has become increasingly complex due to bias, misinformation, and information overload. With numerous sources presenting the same news with different perspectives, users struggle to distinguish between factual reporting and biased narratives. Traditional news platforms often fail to provide a balanced view, leaving audiences vulnerable to echo chambers and manipulated information.

Individual news websites typically present events from a single perspective and do not offer multi-source comparisons. While existing aggregators like Google News and Flipboard do bring together articles from different outlets, they still lack political bias detection or real-time AI-driven analysis. Some platforms, such as Ground.News, attempt to address media bias but often depend on pre-labeled datasets instead of dynamic analysis. NewsInsights bridges these gaps by leveraging ML to analyze and summarize news from multiple perspectives, ensuring users gain a balanced and fact-based understanding of current events.

The relevance of this project in the real world extends beyond individual users to academics, journalists, and policymakers, who require unbiased, well-structured, and data-driven news insights. By integrating bias-aware analysis, keyword highlighting, and trend detection, NewsInsights promotes media literacy and critical thinking, helping users navigate the complexities of modern journalism. As misinformation continues to influence public opinion and decision-making, the need for an intelligent, fact-driven, and AI-powered news aggregator like NewsInsights has never been more critical.

1.3 Research Gaps

1. Interpretability and Explainability (XAI) Beyond Classification:
    a. Explanation: The dominant paradigm in bias detection remains classification—assigning a discrete label (e.g., "left," "right," "neutral") to a piece of text. While model accuracy has improved, this approach provides little insight into the nature of the detected bias. A simple label is insufficient for the ultimate goals of fostering media literacy, training journalists, or enabling nuanced academic analysis. The PRISM framework's move towards interpretable embeddings is a promising but nascent direction.
    b. Identified Gap: There is a critical need for models that move beyond classification to generation of fine-grained, evidence-based explanations. A truly useful system should not only flag a sentence as biased but also perform extractive explanation by highlighting the specific words or phrases contributing to the bias, and abstractive explanation by identifying the rhetorical or framing device being employed (e.g., "This sentence uses loaded language to frame the subject negatively" or "This passage creates a false balance by giving undue weight to a fringe viewpoint").

Research in this area is essential for building systems that are not just detectors but also diagnostic and educational tools.

2. Cross-Lingual and Cross-Cultural Bias Detection:
    a. Explanation: The existing body of research is overwhelmingly concentrated on the English language and the specific socio-political context of the United States. This creates a significant gap in our understanding, aBs political bias is a global phenomenon that manifests in culturally and linguistically specific ways. The framing of issues, the choice of loaded terms, and the entire political spectrum can differ dramatically between a two-party presidential system and a multi-party parliamentary one.
    b. Identified Gap: A major research frontier lies in the development of robust multilingual and cross-cultural bias detection systems. This requires a multi-pronged effort:
    c. Creating new, culturally-aware annotation schemas and datasets for a diverse range of languages and political systems;
    d. Investigating the efficacy of cross-lingual transfer learning to see if models trained on high-resource languages like English can be adapted to low-resource languages;
    e. Performing comparative analyses to understand how the linguistic markers of bias differ across cultures[18].
3. Modeling Subjectivity and Annotator Disagreement:
    a. Explanation: A fundamental flaw in the standard supervised learning approach is the assumption of a single, objective "ground truth" label for bias. In reality, the perception of bias is highly subjective. Forcing annotators to agree on a single label for an ambiguous text either results in low inter-annotator agreement (making the data unreliable) or washes out the very nuance the model should be learning. The BiasLab dataset is a notable exception that explicitly captures annotator perception and disagreement.
    b. Identified Gap: There is a significant opportunity to develop models that embrace subjectivity rather than ignoring it. Instead of predicting a single class, future models could be designed to predict a distribution of perceived bias. For example, a model could be trained to predict that "70% of liberal readers would perceive this sentence as neutral, while 80% of conservative readers would perceive it as biased." This would require new datasets annotated by a diverse panel of raters with recorded demographic and ideological information, as well as novel probabilistic model architectures capable of learning from and representing human disagreement[19].
4. Holistic Multimodal Bias Detection:
    a. Explanation: News consumption in the digital age is an inherently multimodal experience. The overall message and its potential bias are conveyed not just through the text of an article but through the interplay of headlines, images, captions, infographics, and embedded videos. An unflattering photograph of a politician, the selective cropping of an image to alter its context, or the juxtaposition of a neutral headline with a highly emotional image are all powerful methods of conveying bias that text-only models are blind to.

    b. Identified Gap: A largely unexplored research area is the development of models that can perform holistic, multimodal bias detection. This requires integrating state-of-the-art computer vision and NLP techniques to analyze how textual and visual elements work together to construct a biased narrative. While initial datasets like NewsMediaBias-Plus provide a starting point , the architectural and theoretical frameworks for fusing and reasoning over this multimodal information are still in their infancy[20].

5. Robustness, Nuance, and Mitigating LLM-Specific Failures:
    a. Explanation: The increasing reliance on LLMs for bias detection introduces a new class of challenges related to their specific failure modes. These models are known to be highly sensitive to prompt phrasing and can exhibit sycophancy, where they uncritically adopt the biases present in the user's query.43 Furthermore, all current models, including LLMs, struggle with the most subtle and powerful forms of bias, particularly
    bias by omission, where the partiality is defined by what is strategically left unsaid. Detecting this requires a model to have deep world knowledge and the ability to reason about what constitutes a complete and balanced account of an event.
    b. Identified Gap: There is a pressing need for research focused on improving the robustness and reliability of LLMs for this sensitive task. This includes: (1) developing methods to quantify and mitigate the inherent political biases of the LLMs themselves 30; (2) designing adversarial training regimes and prompting strategies to make models less susceptible to manipulation and sycophancy; and (3) creating novel architectures and training objectives specifically designed to tackle non-lexical biases like omission, perhaps by incorporating knowledge graphs or cross-referencing multiple sources to identify missing information.

1.4 Problem Definition and Scope

The exponential rise in online news platforms has created an environment where readers are overwhelmed with diverse perspectives, hidden biases, and excessive redundancy of information. Traditional news aggregators mainly focus on presenting headlines from multiple sources but lack advanced mechanisms to detect bias, summarize content, or provide contextual clustering. This creates challenges in enabling readers to consume news in a balanced, concise, and meaningful way. Thus, there is a pressing need for an intelligent system that not only aggregates news but also enhances it through ML techniques.

The scope of this project is limited to the domain of text-based news aggregation and analysis. The system focuses on functionalities such as political bias detection, clustering of related articles, summarization, sentiment analysis, and recommendation of relevant content. The solution is designed to assist readers, researchers, and decision-makers in obtaining unbiased and concise news insights. However, the scope does not extend to real-time verification of authenticity of news, detection of deepfakes, or handling of multimedia formats such as images, audio, or video.

1.5 Assumptions and Constraints

Table 1: Assumptions and Constraints Table

S No.	Assumptions and Constraints
1.	It is assumed that the news data being aggregated originates from legitimate, trustworthy, and credible publishers. The system primarily collects content through scraping official websites or consuming RSS feeds. While the framework leverages the reputation of these sources to ensure reliability, it does not implement a dedicated fact-checking mechanism at the article level. Consequently, if misinformation exists at the source, such inaccuracies may be propagated within the aggregated system. This limits the platform's ability to guarantee absolute authenticity of the presented information.
2.	The framework assumes continuous accessibility to APIs, RSS feeds, and scraping endpoints from major news outlets to ensure multi-source data collection. These feeds are expected to provide comprehensive coverage of diverse domains such as politics, business, technology, sports, and entertainment. In scenarios where feeds become unavailable, are paywalled, or restrict automated access, the system's ability to compare across multiple sources and maintain balanced aggregation could be constrained. The system therefore operates under the assumption of long-term data availability from external publishers.
3.	The system assumes that the employed machine learning and natural language processing models for tasks such as content summarization and sentiment classification operate at an acceptable level of accuracy. While state-of-the-art pre-trained models are fine-tuned for these purposes, no AI system can achieve perfect precision or recall across all contexts, languages, and domains. Factors such as sarcasm, implicit bias, or culturally nuanced reporting may affect the accuracy of predictions, and occasional misclassifications must be acknowledged as a constraint of the system.
4.	It is assumed that user activity, such as article clicks, reading duration, reactions, or category selections, reliably represents genuine user preferences. The system interprets these behavioral signals as proxies for interests and incorporates them into the recommendation engine. However, user interactions may not always reflect true intent for example, a user might click on sensational content out of curiosity rather than preference, or avoid certain articles despite being interested due to lack of time. Such discrepancies may affect the personalization outcomes, limiting the accuracy of inferred interests.
5.	It is recognized that new users, lacking historical activity or preference data, may initially receive less accurate or generalized recommendations. The recommendation engine relies on accumulating sufficient interaction signals to progressively refine personalization. Similarly, newly published articles may lack adequate contextual embeddings or comparative clustering information during

	their initial inclusion in the database, leading to suboptimal categorization or recommendation. This “cold-start” limitation is intrinsic to recommendation systems and is mitigated gradually as user activity and article metadata grow over time.

1.6 Approved Objectives

1. Efficient News Aggregation
   Develop a system that collects news articles from multiple sources in real-time, ensuring users receive the latest and most relevant updates.

2. AI-Powered Summarization, Sentiment Analysis & Keyword Extraction
   Utilize AI-powered summarization, sentiment analysis, and keyword extraction to generate concise summaries, highlight essential topics, and provide deeper insights for better content understanding.

3. Political Bias Detection.
   Design and integrate a custom machine learning model to analyze news articles and detect political bias and provide transparency in news reporting, enabling users to assess different viewpoints before forming opinions.

4. Personalized News Recommendations.
   Develop an AI-powered recommendation engine that suggests articles based on user preferences, reading history, and content relevance.

5. Content Filtering & Customization.
   Enable users to filter news content based on categories, sources, bias levels, or sentiment, ensuring a personalized experience.

1.7 Methodology

The methodology adopted for the NewsInsights project follows an iterative and incremental development model. This approach was chosen to effectively manage the complexities of integrating web development, data engineering, and machine learning components. The project lifecycle is broken down into five distinct phases, allowing for continuous refinement and integration.

1. System Design and Planning
   This initial phase focused on establishing the foundational blueprint for the project. It involved defining the system architecture, designing the database schema for PostgreSQL, and creating detailed API specifications. A comprehensive Work Breakdown Structure (WBS) was developed to outline all tasks and deliverables, ensuring a clear development roadmap.

2. Data Aggregation and Processing Pipeline
   The second phase concentrated on building the core data ingestion engine. This involved developing robust web scraping scripts to collect news articles from diverse sources, implementing data cleaning and deduplication logic, and creating a process to generate and store vector embeddings for all content using SentenceTransformer models.

3. AI Model Development and Integration
   This phase involved the research, development, and integration of the core AI modules. It included implementing transformer-based models from Hugging Face for summarization (BART) and sentiment analysis (RoBERTa), as well as training and deploying a custom machine learning classifier for political bias detection.

4. Full-Stack Application Development
   With the data and AI pipelines in place, this phase focused on building the user-facing application. The backend was developed using Django and Django REST Framework to create APIs for serving news, managing users, and delivering recommendations. Concurrently, the frontend was built as a single-page application using Next.js to create an interactive and responsive user interface.

5. System Integration, Testing, and Deployment
   The final phase involved integrating all components—the data pipeline, AI modules, backend, and frontend—into a cohesive system. A comprehensive testing strategy, including unit, integration, and performance testing, was executed to ensure the platform's stability and reliability before preparing for containerized deployment with Docker.

1.8 Project outcomes and Deliverables

The proposed system is expected to achieve the following outcomes:

1. Aggregated News Feed: A unified news repository created by scraping and storing articles from multiple news outlets, enabling centralized access to diverse sources.
2. Clustered and Summarized Content: Automatic grouping of semantically similar news articles into clusters, with concise AI-generated summaries to reduce redundancy.
3. Sentiment and Political Bias Analysis: Identification of article-level sentiment (positive, negative, neutral) and political orientation (left, right, neutral), providing users with contextual insights.
4. Personalized Recommendations: AI-driven recommendations based on user interaction history, embeddings, and semantic similarity, ensuring relevance and personalization.

The key deliverables of this project include:

1. Working Web Application:
   Frontend interface for browsing, searching, and interacting with news articles.
   Backend services for scraping, clustering, summarization, and recommendations.

2. AI/ML Models:
   Custom built model for political bias detection.
   Recommendation engine based on embeddings and cosine similarity.

3. Documentation:
   IEEE-style project report with methodology, results, and analysis.
   User manual for application usage.

1.9 Novelty of Work

1. Transparent and Explicit Bias Detection
   A core innovation of this project is the integration of a custom-trained machine learning model to explicitly classify the political leaning of news content. While other platforms may implicitly try to balance sources, NewsInsights provides direct, transparent labeling as part of the user experience. This feature is a direct response to the growing problems of media echo chambers and misinformation, aiming to empower users with the tools for critical media consumption.

2. Comprehensive Aggregation for a Broader Perspective
   This ensures a complete and multi-faceted understanding of world events, a feat impossible for any single news channel due to inherent limitations in editorial focus, resources, and perspective. By systematically collecting articles from a wide array of sources spanning different political leanings, and categories the system allows users to see beyond a single narrative. This approach not only builds a more nuanced picture of major events by contrasting different angles.

3. Smart Personalization
   The recommendation system goes beyond simply showing trending topics. It combines recent news with user preferences using semantic embeddings and interaction data, ensuring that recommendations are both timely and user-centric.

Requirement Analysis

2.1 Literature Survey

2.1.1 Theory Associated With Problem Area

Defining Bias is a deviation from standards of objective journalism that appears more often as subtle partisan slant than outright fabrication. It manifests as (a) content-level biases (omission, story selection, placement, source selection, labeling), (b) structural biases (gatekeeping, agenda-setting, framing), and (c) economic/psychological biases (corporate and demand-driven incentives; audience confirmation bias) [1].

1. Foundational communication theories:
    a. Agenda-Setting: Media influence what the public thinks about by elevating issue salience through selection and prominence [2].
    b. Framing: Media influence how the public thinks about issues by emphasizing selected aspects to define problems, diagnose causes, make moral judgments, and suggest remedies [3].

2. Operationalizing bias for computation:
    a. Transition from outlet-level analysis → lexical/sentence-level markers → model-level statistical properties.
    b. Two key linguistic forms from NPOV studies: framing bias (loaded/subjective language) and epistemological bias (presuppositions/entailments via grammar/verbs like "acknowledged" vs "claimed").
    c. Modern view recognizes bias in model representations (representational and allocational harms) [4].
    d. Recursive challenge: Using LLMs to annotate bias risks laundering their intrinsic leanings into "ground truth," creating feedback loops.

2.1.2 Existing Systems and Solutions

Phase 1 – Lexicon-based/statistical counts

Transparent but brittle; fail on context, negation, irony; struggle with evolving political language [5].

Phase 2 – Traditional ML (SVM/LogReg)

Better than lexicons but reliant on heavy feature engineering (n-grams, POS, syntax, sentiment)[6].

Phase 3 – Deep learning

1. RNN/LSTM: Capture order/short-range dependencies; data-hungry; limited long-range context.

2. Transformers (BERT family): Pretraining + self-attention yields contextual understanding; fine-tuning became standard; domain-adaptive pretraining (e.g., POLITICS on news corpora) boosts results[7]

Phase 4 – State of the art & LLMs:

1. Architectures beyond vanilla Transformers (graph-based for event/entity relations; cross-encoders for nuanced scoring).
2. LLMs (GPT, Llama, Gemini): Zero-/few-shot with advanced prompting (e.g., Chain-of-Thought).
3. Emerging emphasis on interpretability (e.g., PRISM embeddings, tools like BiasScanner) to answer not just "is it biased?" but "why and where?"

Table 2: Literature Survey

S. No.	Roll Number	Name	Paper Title	Tools/Technology	Findings	Citation
1	102216103	Manav Singhal	Target-Aware Contextual Political Bias Detection in News	BERT, RoBERTa, BASIL dataset, Data Augmentation (BANC, ABTA, EBTA)	Context is critical for sentence-level bias detection. Proposed target-aware data augmentation techniques significantly improve performance, achieving a state-of-the-art F1-score of 58.15 on the BASIL dataset. Highlights the difficulty of distinguishing informational vs. lexical bias.	Maab, I., Marrese-Taylor, E., & Matsuo, Y. (2023). In Proceedings of the 13th IJCNLP and the 3rd AACL (Volume 1: Long Papers).[9]
2			Political Leaning and Politicalness Classification of Texts	Transformer Models (BERT, RoBERTa, DeBERTa), POLITICS model, 12 leaning datasets, 18 politicalness datasets	Existing models suffer from poor generalization ('siloed solutions'). Introduces 'politicalness' as a prerequisite classification task. Creates a large, diverse dataset for this task. Finds that domain-specific pre-training (POLITICS model) is highly effective.	Volf, M., & Simko, J. (2025). arXiv:2507.13913[7].

3			Quantifying Generative Media Bias with a Corpus of Real-world and Generated News Articles	LLMs (GPT series, Llama 2, Mistral), RoBERTa, POLITICS model, AllSides dataset	Instruction-tuned LLMs exhibit consistent, measurable political bias (typically left-leaning). Proposes a metric for 'political bias shift' between human and AI text. Bias is topic-dependent and influenced by prompt design.	Trhlik, F., & Stenetorp, P. (2024). In Findings of the Association for Computational Linguistics: EMNLP 2024[10].
4			PRISM: A Framework for Producing Interpretable Political Bias Embeddings with Political-Aware Cross-Encoder	Cross-Encoder (DeBERTa-v3-large), LLMs (GPT-4o), NewsSpectrum & BigNews datasets	Proposes a shift from classification to producing interpretable embeddings. PRISM outperforms SOTA models in classification and diversified retrieval without fine-grained annotations. Embeddings are explicitly tied to controversial topics, enabling explainability.	Sun, Y., Huang, Q., Tung, A. K. H., & Yu, J. (2025). In Proceedings of the 63rd Annual Meeting of the ACL[8].
5			A Novel BERT-based Classifier to Detect Political Leaning of YouTube Videos based on their Titles	BERT, Word2Vec, GloVe, YouTube API, Custom dataset of 10M video titles	Adapts bias detection to a new domain (YouTube titles). Fine-tuned BERT model achieves 75% accuracy and 77% F1-score, outperforming alternatives. Demonstrates that even short, informal text like video titles contains strong political signals.	AlDahoul, N., Rahwan, T., & Zaki, Y. (2024). arXiv:2404.04261[11].
6			Navigating Nuance: In Quest for Political Truth	Llama-3 (70B), Chain-of-Thought (CoT) prompting, Media Bias Identification Benchmark (MBIB) dataset	Explores the effectiveness of advanced prompting (CoT) for LLMs in bias detection. Achieves performance comparable to a fully fine-tuned SOTA ConvBERT model, highlighting the power of in-context learning and reducing the need for extensive fine-tuning.	Sar, S., & Roy, D. (2024). In The 2024 ACM/IEEE Joint Conference on Digital Libraries (JCDL '24)[12].

2.1.3 Research Findings for Existing Literature

1. Performance & difficulty:
   Transformers dominate, yet sentence-level bias detection remains hard (e.g., mid-50s F1 on BASIL), highlighting persistent ambiguity and nuance[9].

2. Primacy of context:
   Strong gains come from incorporating target-aware and cross-document context (multiple articles on the same event), discourse structure, and event graphs; bias is rarely visible from isolated tokens[10].

3. Granularity shift:
   Movement from outlet/article labels to sentence/span-level detection enables actionable highlights of biased language, though it is more challenging[7].

4. Data bottleneck:
   High-quality, fine-grained annotations are scarce and subjective; progress relies on data-centric strategies: unifying many datasets (e.g., “politicalness” as a precursor task), domain pretraining, and data augmentation tailored to events/targets[11].

5. LLM observations:
   Instruction-tuned LLMs show measurable, topic-dependent leanings; prompt design impacts outputs, underscoring the need for robustness and auditing[12].

2.1.4 Problem Identified

1. Inherent subjectivity & low agreement:
   Bias judgments vary by reader ideology and background; omission bias is especially hard (reasoning about what’s not said)[13].

2. Poor generalization/domain dependence:
   Models trained on U.S. mainstream news falter on social media, YouTube titles, opinion blogs, or other countries’ media—yielding siloed solutions[14].

3. Data scarcity & noisy labels:
   Expert sentence-level labels are costly; weak supervision via publisher slant introduces noise and overfitting risks[15].

LLM-specific challenges

1. Inherent leanings from pretraining data.
2. Prompt sensitivity & sycophancy (alignment with user premise).
3. Opacity of decisions (black-box reasoning), motivating XAI approaches[16].

4. Anglophone/U.S.-centric focus: Limited multilingual/cross-cultural coverage reduces global applicability[14].

2.1.5 Survey of Tools and Technologies Used

Benchmark datasets

1. BASIL: Sentence/span-level annotations with targets and bias types; cornerstone for fine-grained evaluation.
2. BABE: Expert sentence-level labels; used in practical explainers (e.g., BiasScanner).
3. MBIB: Unified benchmark across political/cognitive/linguistic bias types.
4. Publisher-level corpora (AllSides, BigNews, NewsSpectrum): Scalable but weak/noisy labels; valuable for pretraining and trend learning.
5. Emerging sets: BiasLab (captures annotator disagreement and rationales); NewsMediaBias-Plus (multimodal: text + images); domain/platform-specific corpora (social media, podcasts, YouTube)[15].

Core architectures/models

1. Transformer encoders: BERT/RoBERTa/DeBERTa as baselines for fine-tuning.
2. Domain-specific models: e.g., POLITICS (RoBERTa continually pre-trained on millions of news articles).
3. LLMs: GPT/Llama/Gemini for zero-/few-shot with advanced prompting.
4. Specialized designs: Cross-encoders for interpretable multi-dimensional scores (PRISM); Graph Neural Networks for event/entity relation reasoning[16].

Ecosystem & frameworks

1. Hugging Face Transformers (models, datasets, training utilities).
2. PyTorch / TensorFlow as primary deep learning backends.

Key research gaps

1. Interpretability beyond labels: Extractive highlights + abstractive rhetorical explanations.
2. Cross-lingual/cross-cultural bias modeling and datasets.
3. Modeling subjectivity: Predicting distributions of perceived bias by audience segment.
4. Multimodal integration: Jointly analyzing headlines, body text, images, captions, and visuals.
5. LLM robustness: Bias auditing/mitigation, anti-sycophancy prompting/training, and strategies for detecting bias by omission (e.g., knowledge graphs, cross-source corroboration)[17].

2.1.6 How Our Work Builds on Existing Research

Our proposed work is firmly grounded in the current state-of-the-art for political bias detection, while introducing innovations in both contextual modeling and system architecture

We deliberately leverage proven techniques that have become the gold standard in computational linguistics:

1. Foundation in Transformer Models:
   For the core classification task, we rely on RoBERTa and DeBERTa, following the evidence that transformer-based models consistently outperform earlier deep learning methods such as LSTMs in capturing subtle linguistic markers of bias (Kulkarni et al., 2021; Chen et al., 2023). This ensures our system starts with a high-performing baseline aligned with best practices.

2. Leveraging Contextual Embeddings:
   To capture semantic similarity between news articles, we employ Sentence-BERT. Bias is inherently contextual, and shallow lexical features are insufficient. By embedding full article text (title + content), the system can reason about semantic framing choices rather than word overlap alone.

3. Data-Centric Strategies:
   We incorporate insights from prior research emphasizing the role of dataset unification, augmentation, and domain-adaptive pretraining. Our pipeline leverages these strategies to reduce dependence on limited annotated corpora and improve robustness.

4. Alignment with Explainable AI Trends:
   By extending beyond label prediction, our design integrates event comparisons and contextual analysis, aligning with ongoing efforts in interpretable bias detection frameworks such as PRISM.

2.1.7 How Our Work Differs and Innovates

At the same time, our methodology introduces several technical and conceptual innovations:

1. Focus on the Indian Political Context:

   Most political bias datasets and models (e.g., BASIL, AllSides) are U.S.-centric, overlooking linguistic and rhetorical nuances of Indian politics. By targeting Indian news sources, even in English, our system addresses this cultural and regional gap. Markers of bias—choice of descriptors, references to parties, and framing of events—are India-specific, and our work is among the first to model them systematically.

2. Novel Event-Clustering Pipeline:

   Unlike prior work that classifies articles in isolation, we introduce a semantic event-clustering stage before classification. This is implemented through:
   a. Sentence-BERT embeddings to represent articles.
   b. Deduplication via text hashing to prevent repeated content from skewing clusters.

c. Temporal filtering (72-hour sliding window) to ensure articles grouped together are from the same event cycle.
d. Category-aware similarity boosts so coverage of the same type of event (e.g., elections, protests) clusters more tightly.
e. Dynamic centroid recomputation as new articles are added, keeping clusters semantically stable[10].
f. Database persistence (RawNewsFeed and Cluster models) to maintain embeddings and centroids across sessions for robustness.

This allows the model to detect framing bias and omission bias, which only emerge by comparing coverage across outlets reporting on the same event—something single-article models cannot achieve.

3. Hybrid Unsupervised + Supervised Design:

   Our architecture combines unsupervised clustering with supervised classification:
   a. Unsupervised Stage: Articles are organized into semantically coherent clusters of events using Sentence-BERT and temporal similarity checks.
   b. Supervised Stage: Once clustered, a fine-tuned classifier (RoBERTa/DeBERTa) labels the clusters and their member articles with bias categories.

   This hybrid setup makes the system:
   a. Robust to noisy, unlabeled news corpora (since clustering organizes chaos into structure).
   b. Scalable to large real-world news streams.
   c. Context-aware, because bias is determined relative to how multiple outlets frame the same event, not how one article reads in isolation.

2.2 Software Requirement Specification

2.2.1 Introduction

2.2.1.1 Purpose

The purpose of this project is to design and implement an AI-powered news aggregator system that collects, processes, and presents news articles from multiple sources in a structured, unbiased, and user-personalized manner. The system aims to overcome limitations of traditional aggregators by integrating natural language processing (NLP) techniques for bias detection, summarization, sentiment analysis, and clustering of similar stories. Ultimately, the platform empowers users to access diverse perspectives, make informed judgments, and discover relevant news efficiently.

2.2.1.2 Intended Audience and Reading Suggestions

Developers & Engineers: To understand functional modules (scraping, clustering, recommendation) and technical specifications.

Researchers & Academics: To analyze novelty in bias detection and multilingual handling.
End Users: To explore system capabilities like recommendations, summaries, and unbiased reporting.

2.2.1.3 Project Scope

The system aggregates news from at least 10 major news outlets through web scraping and APIs, stores raw data in a structured database, and processes it using clustering and embeddings. NLP models then perform summarization, sentiment analysis, and bias classification to present concise and balanced perspectives. A recommendation engine tailors news feeds based on user activity and preferences.
The scope includes:

1. Multi-source data ingestion.
2. News clustering & redundancy reduction.
3. AI-powered summaries & bias detection.
4. Sentiment classification for articles.
5. Personalized recommendations.
6. Secure and scalable web application for users.

2.2.2 Overall Description

2.2.2.1 Product Perspective

The system operates as a web-based platform where backend services (scraping, clustering, ML models) interact with a PostgreSQL database and vector embeddings for semantic similarity. The architecture follows a modular design.

2.2.2.2 Product Features

1. Aggregation of news from multiple sources.
2. Automatic clustering of similar articles to reduce redundancy.
3. AI-powered summarization for concise news delivery.
4. Political bias detection (Left, Right, Neutral).
5. Sentiment classification (Positive, Negative, Neutral).
6. Ability to add bookmarks.
7. Personalized recommendations based on user interaction.
8. Search and filter functionalities.

2.2.3 External Interface Requirements

2.2.3.1 User Interfaces

Web application (React-based frontend) with features like:

1. News cards with summaries, bias tags, and sentiment labels.
2. Filtering options by category, sentiment, or bias.
3. Recommendation section.
4. Bookmarks Section.

2.2.3.2 Hardware Interfaces

No direct hardware interaction apart from client devices (desktop/laptop/mobile).

2.2.3.3 Software Interfaces

Database: PostgreSQL with vector similarity search.
Frontend: Next.js, Axios
APIs: RESTful API endpoints built using Django and DRF for article retrieval, recommendations, and user activity logging.
Libraries: Hugging Face Transformers (summarization, embeddings), Celery + Redis (background tasks).

2.2.4 Other Non-functional Requirements

2.2.4.1 Performance Requirements

1. The system must support at least 1,000 concurrent users.
2. Article scraping latency should not exceed 15 minutes for new content.
3. Clustering, Summarization, Sentiment Analysis & Political Bias detection processing time should not exceed 15 minutes for new content.
4. The recommendation engine should respond within 500 ms for user queries.

2.2.4.2 Safety Requirements

1. Backup and recovery mechanisms for database failures.
2. Other components of the system should continue working if one component become unavailable.

2.2.4.3 Security Requirements

1. Role-based authentication and authorization.
2. HTTPS communication across all services.
3. Protection against scraping detection and API rate-limits by implementing throttling strategies.

2.3 Cost Analysis

Cost Analysis Assumptions

The following cost analysis provides a detailed estimate of the monthly operational expenses required to run the NewsInsights platform. The figures presented in the subsequent tables are based on a set of foundational assumptions regarding the hosting environment, usage scale, and third-party service models. It is important to consider these assumptions as they provide the context for the financial projections.

1. Hosting Environment:
   The costs are calculated based on a Virtual Private Server (VPS) or a comparable Infrastructure-as-a-Service (IaaS) cloud model from major providers (e.g., AWS, Google Cloud, DigitalOcean). This assumes we are renting virtualized hardware rather than purchasing physical servers.

2. Pricing and Currency:
   All financial figures are estimates presented in Indian Rupees (INR) and are based on current market rates as of Q3 2025. Actual costs may vary depending on the chosen provider and any future price fluctuations.

3. User Load:
   The infrastructure is sized to support an initial target of approximately 5,000 to 10,000 Daily Active Users (DAU). It is further assumed that an average active user reads 6-8 clustered stories per day. This translates to an estimated 40,000 to 60,000 article cluster views per day, culminating in approximately 1.2 to 1.8 million views per month. The costs for the database and application server are sized to handle this level of read traffic and the associated user event logging.

4. AI Model Deployment:
   The costs for AI models are based on a daily ingestion rate of approximately 1,500 to 2,500 new articles. This volume translates directly to the processing load on our AI services, requiring roughly 2,000 sentiment analysis and political bias checks per day. Furthermore, this ingestion rate is expected to create 300 to 500 new story clusters daily, each of which must be processed by the GPU-intensive summarization model. The high costs associated with the AI inference endpoints are a direct function of this daily content processing volume.

5. Scope of Cost:
   This analysis is strictly limited to recurring monthly operational costs. It excludes one-time setup fees, domain registration (which is annualized), software licensing (as the stack is primarily open-source), and any human resource costs related to development, management, or maintenance.

The operational cost for the NewsInsights platform's infrastructure is based on a modern, modular architecture designed for scalability and performance. As detailed in Table 3: Monthly Infrastructure Cost Breakdown, the total estimated monthly expense ranges from ₹19,600 to ₹27,150. This budget covers all essential components required to host the application, manage data, and serve a significant user base, ensuring a responsive and reliable user experience from launch.The primary Application Server, projected to cost between ₹5,000 and ₹7,000, is designated to run the Django backend and the Celery workers responsible for data processing. A significant portion of the budget is allocated to a more powerful Database Server, estimated at ₹10,000 to ₹13,000, which is necessary to handle the memory-intensive vector search operations required by PostgreSQL with the pgvector extension. This setup is complemented by a dedicated Redis Server (₹1,500 – ₹2,500) for efficient caching and task queuing, along with essential allocations for block storage and data transfer, ensuring a robust and responsive system.

The most significant cost driver within this budget is the Database Server, a deliberate investment reflecting its critical role in powering the platform's core AI recommendation feature. By allocating substantial resources to the database, we ensure that the computationally expensive similarity searches remain fast, providing a seamless and personalized user experience. This balanced allocation ensures that while the overall cost is managed, performance is not compromised, establishing a scalable foundation capable of handling the initial demands of data ingestion and user traffic.

Table 3: Monthly Infrastructure Cost Breakdown

Component	**Minimum Requirement**	**Estimated Monthly Cost (INR)**	**Justification**
Application Server	8 vCPU, 16 GB RAM	₹5,000 – ₹7,000	To run the Django application, DRF APIs, and Celery workers for scraping and processing. The custom political bias model would also run here.
Database Server	8 vCPU, 32 GB RAM	₹10,000 – ₹13,000	For PostgreSQL with the pgvector extension. Vector similarity searches are memory-intensive, requiring a powerful instance for good performance.
Redis Server	2 vCPU, 8 GB RAM	₹1,500 – ₹2,500	To act as the Celery message broker and for application-level caching.
Block Storage	~200 GB	₹1,500 – ₹2,000	For storing news articles, user data, system logs, and database backups.
Data Transfer	~2-3 TB/month	₹1,500 – ₹2,500	To account for data from web scraping and traffic from users accessing the platform.
Domain & SSL	N/A	₹100 – ₹150	Standard annual cost of a domain name and SSL certificate, broken down monthly.
Subtotal		**₹19,600 – ₹27,150**	

Beyond the core infrastructure, the platform's intelligence layer is powered by a suite of specialized AI and Machine Learning models, whose costs are outlined in the Table 4: AI/ML Models Cost Breakdown. A hybrid hosting strategy has been adopted to balance performance and expense. The lightweight models for embedding generation and the custom political bias detection are designed to be self-hosted, running directly on the main Application Server, thus their costs are absorbed within the infrastructure budget. In contrast, the more demanding services are deployed on dedicated inference endpoints. The Sentiment Analysis model operates on a modest CPU instance, costing an estimated ₹4,000 to ₹5,000 per month. The most significant operational expense is the Summarization service, which requires a dedicated GPU-powered instance to handle the facebook/bart-large-cnn model effectively, incurring a substantial cost of approximately ₹30,000 to ₹35,000 monthly.

Table 4: AI/ML Models Cost Breakdown

Model / Service	**Hosting Strategy**	**Estimated Monthly Cost (INR)**	**Justification**
Embeddings (all-MiniLM-L6-v2)	Self-hosted on App Server	Included in Infrastructure	This model is lightweight and can efficiently run on a CPU within a Celery task.
Political Bias Detection	Self-hosted on App Server	Included in Infrastructure	The custom-trained model is assumed to be optimized to run on the main application server's CPUs.
Sentiment Analysis (cardiffnlp/twitter-roberta-base-sentiment)	Hugging Face Inference Endpoint (CPU Instance)	₹4,000 – ₹5,000	This model runs well on a small, dedicated CPU instance for reliable performance without burdening the main server.
Summarization (facebook/bart-large-cnn)	Hugging Face Inference Endpoint (GPU Instance)	₹30,000 – ₹35,000	This is a large model that requires a GPU (like an NVIDIA T4) for acceptable inference speed. Running it 24/7 on a dedicated endpoint is a significant cost.
Subtotal (AI/ML)		**₹34,000 – ₹40,000**	

Conclusion

In conclusion, the total estimated monthly operational cost for the NewsInsights platform, combining both infrastructure and specialized AI services, falls within the range of ₹54,000 to ₹67,000. It is crucial to note that the AI/ML services, driven almost entirely by the GPU requirement for the summarization model, constitute the majority of this expenditure. This financial distribution underscores a strategic decision to invest heavily in the advanced, user-facing features that define the platform. While the infrastructure provides a robust and scalable foundation, it is the significant investment in AI that delivers the core value proposition of intelligent, concise, and insightful news analysis, justifying the cost as essential to the project's success.

2.4 Risk Analysis

1. Scalability Risks:
   The system may face performance degradation if user traffic exceeds expected capacity especially during peak news cycles or major events.

2. Dependency Risks
   Reliance on third-party APIs (Hugging Face Inference API, RSS feeds, external NLP services) may cause service disruption or unexpected costs if access is limited, changed, or revoked.

3. Data Quality Risks:
   Since the platform depends on external news sources, misinformation, biased reporting, or incomplete data at the source level can propagate into the system, reducing credibility.

4. Model Accuracy Risks:
   The political bias detection, sentiment analysis, summarization, and recommendation models may produce inaccurate or misleading outputs, affecting trust and user experience.

5. Security & Privacy Risks:
   Handling user interaction data (clicks, reading patterns,preferences) introduces risks of unauthorized access, data leakage, or misuse if strong security practices are not enforced.

6. Cost Overrun Risks:
   Usage-based pricing of inference APIs may lead to higher-than-expected operational costs if user adoption scales quickly, especially for sentiment and summarization tasks.

7. User Adoption Risks:
   Initial cold-start problems in recommendation systems for new users or limited engagement with bias-detection features may reduce the system's perceived value.

Methodology Adopted

3.1 Investigative Techniques

The detection of political bias in news articles is a complex, multi-layered challenge that combines linguistics, political science, communication theory, and computational modeling. To address this effectively, investigative techniques must be technically robust and conceptually aligned with the nuances of media bias. Our chosen methods—Transformer-based models, contextual embeddings, custom similarity-driven clustering, and a hybrid pipeline—are justified both by research trajectories in the field and by the unique requirements of analyzing bias in the Indian political context.

Justification for Transformer Models (RoBERTa, DeBERTa)

Traditional approaches such as lexicon-based systems and SVMs struggle to capture the subtlety of framing, omission, and epistemological biases. Transformer models like RoBERTa and DeBERTa excel because their self-attention mechanisms capture long-range dependencies and nuanced contextual signals.

1. Evidence from Literature: Prior studies have demonstrated that BERT-family models achieve state-of-the-art performance on bias detection datasets such as BASIL, particularly in sentence-level framing analysis.
2. Why We Use Them: RoBERTa and DeBERTa are pre-trained on massive, diverse corpora, ensuring strong generalization. DeBERTa's disentangled attention further refines positional bias handling, a subtle but relevant aspect of sentence framing in political reporting.

Contextual Embeddings via Sentence-BERT

Bias often emerges not from single words but from semantic framing at the sentence or article level. Sentence-BERT embeddings allow us to represent news articles in a dense semantic space that supports cross-outlet comparisons.

1. Why This Matters: Embeddings enable comparative framing analysis—for example, contrasting coverage of a protest described as "defending democracy" versus "causing disruption."
2. Justification: This approach aligns with communication theories such as Framing Theory and Agenda-Setting Theory, which emphasize that media bias is often expressed through issue framing rather than factual distortion.

Event Clustering via Custom Similarity-Based Pipeline

A major innovation in our work is clustering articles into real-world events before classification, using a custom pipeline rather than traditional clustering algorithms like DBSCAN or k-means.

1. Gap in Literature: Most prior research classifies articles independently, missing the comparative context bias detection requires.

2. Our Approach: Articles are clustered based on:
    a. Sentence-BERT embeddings, normalized for consistency.
    b. Temporal filtering (72-hour window) to ensure event coherence.
    c. Deduplication via content hashing to avoid repeated articles skewing clusters.
    d. Category boosting, which slightly increases similarity scores for articles of the same topic/category.
    e. Dynamic centroid updates as new articles join clusters.

   Justification: This pipeline ensures clusters reflect real-world event groupings, enabling direct comparisons across outlets and capturing both framing bias and omission bias, which are otherwise extremely difficult to detect.

Hybrid Unsupervised + Supervised Pipeline

We integrate unsupervised event clustering with supervised bias classification:
Why Hybrid? News scraped from the web is noisy and unlabeled. Clustering imposes structure, while supervised models (RoBERTa/DeBERTa) assign bias labels.
Justification: This two-stage process mirrors real-world conditions—chaotic news streams first organized into events, then labeled contextually. This increases robustness and ensures classification occurs in meaningful, event-aware groupings.

Suitability for the Indian Political Context
Existing datasets (BASIL, AllSides, MBIB) are US-centric. Our project specifically targets Indian political discourse, requiring techniques that can adapt to unique linguistic and cultural markers of bias.

Why Our Techniques Work Here:
1. Fine-tuned Transformers adapt to India-specific rhetorical patterns.
2. Event clustering allows analysis of how Indian outlets cover the same events (e.g., elections, protests, reforms).
3. Contextual embeddings capture cultural framings such as caste, religion, and region-specific terminology.

Conclusion
Our investigative techniques such as Transformer models, contextual embeddings, custom similarity-based clustering, and a hybrid pipeline all align with the state of the art while addressing gaps in context-awareness, generalization, and cultural applicability. This ensures our system is both academically rigorous and practically impactful for the Indian news landscape.

3.2 Proposed Solution

1. Data Aggregation

   The system begins by aggregating raw news content from multiple online sources. A custom web scraping pipeline continuously extracts articles from ten distinct news channels, ensuring diversity in coverage across categories such as India, World, Business, Technology, Entertainment, Sports, Science, and Health.

   Each article is stored in a structured database model with fields including title, content, publication timestamp, source, and category. Deduplication is enforced through source-based uniqueness constraints on URLs.

   This aggregation layer ensures a centralized repository of heterogeneous news data, serving as the input for subsequent clustering and analysis stages.

2. Clustering of News Articles

   To group semantically similar news articles, the system employs the SentenceTransformer model `all-MiniLM-L6-v2`. Each incoming article is preprocessed by combining its title and body content, removing excessive whitespace, and irrelevant characters. The resulting cleaned text is embedded into a 384-dimensional vector space.

   Deduplication is performed using hash-based checks on the article titles, preventing near-duplicate entries. For clustering, embeddings are normalized and compared with existing cluster centroids using cosine similarity. A similarity threshold of 0.75 is applied, with an additional category-based boost of 0.1 when articles share the same category label. To ensure temporal relevance, only clusters formed within the last 72 hours are considered for assignment.

   The clustering process follows two cases:

   a. If the similarity score with an existing cluster centroid exceeds the threshold, the article is added to that cluster. The cluster centroid is then updated as the normalized mean of its member embeddings.
   b. If no suitable cluster is found, a new cluster is created, and the article is designated as its first member.

   To maintain efficiency, cluster sizes are bounded between one and fifty articles. Embeddings are persisted in the database to ensure consistent retrieval and incremental updates.

3. Summarization and Sentiment Analysis

   The system integrates transformer-based models for higher-level news understanding. Summarization is performed using the BART-Large-CNN model hosted on Hugging Face (`facebook/bart-large-cnn`), while sentiment classification is handled by the RoBERTa model (`cardiffnlp/twitter-roberta-base-sentiment`). These models generate concise summaries, sentiment polarity, and confidence scores for each cluster, enhancing interpretability and user experience.

4. Political Bias Detection
   Political bias detection is the central pillar of the proposed system. While existing approaches often rely on classifying single articles in isolation, our design emphasizes event-level comparison across multiple outlets to capture subtler forms of framing and omission bias.
   a. Central pillar of the system → focuses on identifying political bias not only in single articles but through event-level comparison across multiple outlets.
   b. Nature of bias:
      i. Rarely overtly partisan.
      ii. Emerges through story selection, framing, and omission.
          Example: One outlet highlights economic benefits of a policy; another emphasizes public protests.
   c. Event clustering:
      i. Articles grouped into coherent event sets using Sentence-BERT embeddings
      ii. Temporal and categorical similarity applied for robust clustering.
      iii. Ensures all reports of the same incident are analyzed together.
   d. Bias classification::
      i. Within each cluster, transformer-based classifiers (RoBERTa/DeBERTa) are fine-tuned for bias detection.
      ii. Detects bias at both:
          I. Sentence level → loaded words, presuppositions, sentiment polarity.
          II. Article level → framing choices, omission of critical perspectives.
   e. Advantages of integration::
      i. Combines clustering + classification to detect framing and omission bias.
      ii. Captures differences in how outlets report the same event.
      iii. Overcomes limitations of single-document models.
   f. Hybrid design:
      i. Unsupervised stage: Organizes noisy, real-world news into structured clusters.
      ii. Supervised stage: Assigns bias labels to clusters and articles.
      iii. Ensures scalability, robustness, and adaptability to continuous news flows.
   g. Explainability & transparency:
      i. Outputs contextual evidence (highlighted biased sentences + cross-outlet comparisons)
      ii. Aligns with Explainable AI (XAI) principles.
      iii. Aims to make bias detection accurate, interpretable, and educational for users.

5. Personalized News Recommendation
   The system incorporates an AI-powered recommendation engine that leverages user interaction data and semantic similarity of news clusters. User interactions are tracked, which records events such as clicks, reads, likes, and bookmarks.

   a. User Profile Construction: A user profile embedding is computed by averaging the vector representations of the clusters with which the user has interacted. These embeddings are normalized to ensure robust cosine similarity comparisons.

   b. Recommendation Retrieval: To generate recommendations, the user profile embedding is compared against candidate cluster embeddings stored in the database. The system employs PostgreSQL with the `pgvector` extension, enabling efficient nearest-neighbor search using cosine distance. Clusters that are semantically closest to the user profile but not yet interacted with are selected as recommendations.

   c. Bookmarking and Personalization: Users can bookmark clusters for later access, with bookmarks also contributing to the recommendation process. This supports both short-term preferences (recent reads) and long-term personalization (saved content).The recommendation preferences (recent reads) and long-term personalization (saved content). The recommendation engine ensures that each user receives a personalized, semantically relevant news feed, while maintaining diversity by sourcing articles from multiple perspectives.

3.3 Work Breakdown Structure

This diagram represents the overall workflow of NewsInsights, structured as a hierarchical process. At the top level, it captures inputs (news articles), which then flow into functional components like preprocessing, clustering, and classification. The event clustering stage groups related articles together using Sentence-BERT embeddings, temporal filters, and similarity measures. After clustering, the bias detection stage applies transformer-based classifiers (RoBERTa/DeBERTa) to label clusters/articles with bias categories. Finally, the system generates outputs and insights, providing context-aware comparisons across outlets and interpretable explanations of bias patterns.

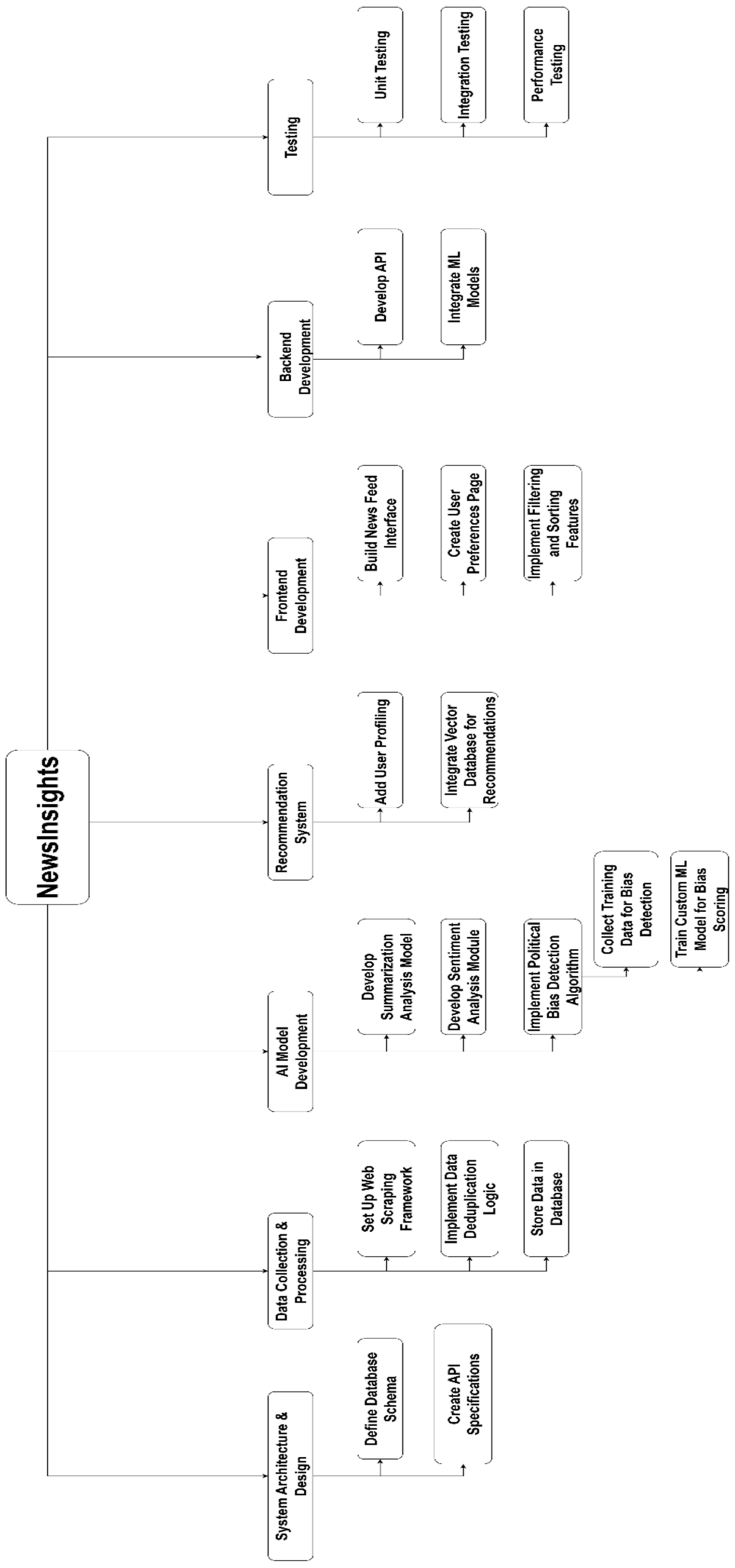

NewsInsights
System Architecture & Design
Define Database Schema
Create API Specifications
Data Collection & Processing
Set Up Web Scraping Framework
Implement Data Deduplication Logic
Store Data in Database
AI Model Development
Develop Summarization Analysis Model
Develop Sentiment Analysis Module
Implement Political Bias Detection Algorithm
Collect Training Data for Bias Detection
Train Custom ML Model for Bias Scoring
Recommendation System
Add User Profiling
Integrate Vector Database for Recommendations
Frontend Development
Build News Feed Interface
Create User Preferences Page
Implement Filtering and Sorting Features
Backend Development
Develop API
Integrate ML Models
Testing
Unit Testing
Integration Testing
Performance Testing

Figure 1: Work Breakdown Structure

The development of the NewsInsights platform is organized into seven primary work packages as shown in Figure 1. This structure provides a clear roadmap for development, covering everything from initial design to final testing and deployment.

1. System Architecture & Design
   This foundational phase establishes the technical blueprint. It involves designing the PostgreSQL database schema to structure all platform data and creating detailed API specifications to define the communication protocol between the frontend and backend.

2. Data Collection & Processing
   This work package focuses on building the data ingestion pipeline. It includes setting up parallel web scrapers to fetch news from multiple sources, implementing deduplication logic to ensure data quality, and storing processed articles and their vector embeddings.

3. AI Model Development
   This package covers the creation of the platform's core intelligence. The main deliverables are three functional AI modules: a transformer-based model for summarization, a sentiment analysis classifier, and a custom-trained model to detect political bias in articles.

4. Recommendation System
   This component is dedicated to personalizing the user experience. It involves tracking user interactions and leveraging this data with article embeddings to build a recommendation engine that suggests relevant content.

5. Frontend Development
   This package involves building the entire user-facing application with React. The work covers creating all necessary pages (homepage, authentication, bookmarks) and implementing key features like filtering and sorting.

6. Backend Development
   This work package focuses on creating the server-side logic using Django. The primary tasks are to build the REST APIs that handle user authentication, news feeds, and recommendations, and to integrate the AI models into the main application.

7. Testing
   This final package ensures platform reliability and performance. It employs a multi-layered strategy, including unit tests for individual functions, integration tests for module interactions, and performance tests to validate system speed and scalability.

3.4 Tools and Technology

1. Backend Development
    a. Python: The primary language for backend and AI development due to its extensive
    b. libraries.
    c. Django: A high-level Python framework for building the core backend application logic.
    d. Django REST Framework (DRF): A toolkit for rapidly developing RESTful APIs to serve data from the backend.

2. Frontend Development
    a. React: A JavaScript library used to build the dynamic and interactive single-page user interface.
    b. HTML5, CSS3, JavaScript: Fundamental web technologies for creating the client-side application.

3. Database
    a. PostgreSQL: A robust relational database for storing all application data, including articles and user profiles. pgvector is used to store high-dimensional vectors directly in PostgreSQL tables.

4. AI & ML
    a. Hugging Face Transformers: A library providing pre-trained NLP models for text summarization and sentiment analysis.
    b. Scikit-learn / PyTorch: Machine learning libraries for training the custom political bias detection model.

5. Asynchronous Task Processing
    a. Celery: A distributed task queue for running background processes like web scraping and AI inference.
    b. Redis: An in-memory data store used as a message broker for Celery and for application caching.

6. Infrastructure & Deployment
    a. Docker: A containerization platform to ensure consistent deployment and development environments.
    b. Git: A version control system for managing the codebase and facilitating team collaboration.

Design Specifications

4.1 System Architecture

The diagram in Figure 2: System Architecture illustrates the system architecture of NewsInsights, showing the flow of news data from collection to end-user delivery. News articles are first gathered from multiple sources (e.g., Indian Express, Hindustan Times, Zee News) through the data stream service. These are stored in the news database, then processed through the AI representation module, which creates embeddings for clustering and bias analysis. User-specific preferences are also stored separately in a user profile database, enabling personalization. Analytical services (bias detection, sentiment analysis, summarization) operate on the processed data and feed results into the backend API, which finally powers the website interface, providing users with context-rich and bias-aware news insights.

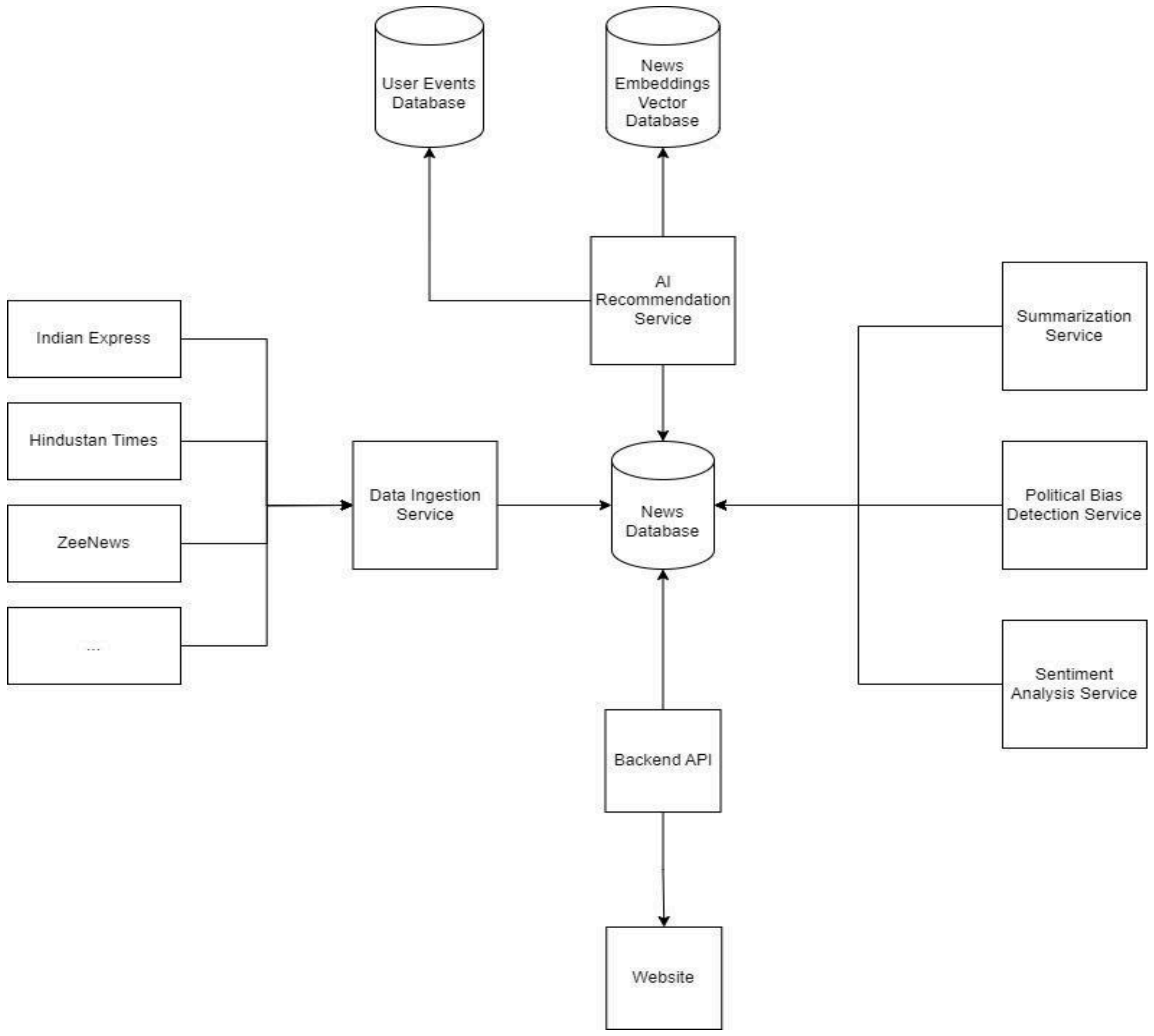


Figure 2: System Architecture

The diagram in Figure 2: System Architecture, illustrates the data flow and interactions between the various components of the NewsInsights platform. It can be broadly divided into several key services and data stores.

1. Data Ingestion Service
   The Data Ingestion Service is the frontline of the NewsInsights platform, responsible not only for collection but also for the initial processing and organization of all incoming news. It employs a fleet of web scrapers to autonomously gather articles from diverse sources. Crucially, as data flows in, this service performs deduplication to identify and discard redundant articles, ensuring data integrity. Immediately following this, it orchestrates the semantic clustering by converting articles into vector embeddings and grouping them based on conceptual similarity. This dual process of cleaning and organizing ensures that all downstream services receive a reliable, well-structured, and non-redundant stream of information, forming the foundation for all subsequent analysis and recommendation tasks.

2. News Database
   At the core of the architecture lies the News Database, which functions as the central repository and the single source of truth for all article-related information. This database is designed to store not only the raw text and metadata collected by the ingestion service but also the enriched data generated by the AI analysis modules. It holds the generated summaries, sentiment scores, and political bias classifications, linking them directly to their corresponding articles. Its robust design ensures that data is efficiently stored and can be quickly retrieved by the Backend API to serve content to the user or by other services for further processing.

3. AI Analysis Services
   The AI Analysis Services represent the intelligence layer of the platform, where raw news content is transformed into insightful, machine-readable data. This suite of services works in parallel to process articles from the News Database, with each module performing a specialized task. The Summarization Service uses advanced models to distill long articles into concise summaries, the Political Bias Detection Service employs a custom-trained classifier to categorize content, and the Sentiment Analysis Service evaluates the tonal leaning of the text. The outputs from these services provide the valuable, multi-faceted analytical layers that are a core feature of the NewsInsights experience.

4. News Embeddings Vector Database
   The News Embeddings Vector Database is a specialized, high-performance data store designed specifically to handle the vector representations of news articles. After an article is processed, its textual content is converted into a numerical vector that captures its semantic meaning. This database is optimized for extremely fast similarity searches, allowing the system to find conceptually related articles in milliseconds. This capability is the technical backbone for both the news clustering feature and the semantically aware recommendation engine.

5. AI Recommendation Service
   The AI Recommendation Service is the engine of personalization, dedicated to creating a unique and relevant news feed for every user. It operates by synergizing two key data sources: the semantic embeddings from the Vector Database and the interaction logs from the User Events Database. By creating a dynamic profile of a user's interests based on their reading history, the service can intelligently query the vector database to find new, unread articles that are semantically similar to what the user has previously engaged with. This ensures that recommendations are not just topically related but are genuinely aligned with the user's nuanced interests.

6. User Events Database
   The User Events Database is a critical component for personalization, acting as the system's memory of all user interactions. Every action a user takes on the website—from clicking on an article and liking it to creating a bookmark—is captured and stored as an event in this database. This continuous stream of interaction data provides the raw material needed by the AI Recommendation Service to build and refine user profiles. By understanding user behavior, the platform can adapt its suggestions over time, leading to an increasingly personalized and engaging experience.

7. Backend API
   The Backend API is responsible for managing the flow of data and communication between the frontend and all the backend services. It exposes a set of well-defined endpoints that the website can call to perform actions like authentication, fetching news articles, retrieving user-specific recommendations, or submitting user interactions. The API handles all the business logic, authenticates users, and orchestrates the necessary calls to the various databases and AI services, ensuring that data is securely and efficiently delivered to the end-user.

8. Website
   The Website is the user-facing application and the primary touchpoint for all interactions with the NewsInsights platform. Built as a modern, responsive web application, it provides an intuitive interface for users to browse news, view the AI-generated analysis, and discover new content through personalized recommendations. It communicates exclusively with the Backend API to request data and send user activity, effectively separating the user interface from the complex backend processing. Its design is focused on delivering a clean, informative, and seamless user experience across all devices.

4.2 Design Level Diagrams

Data Scraping Activity Diagram

The diagram in Figure 3: Data Scraping Activity Diagram, represents the data ingestion pipeline for NewsInsights. The process begins with an automatic trigger that runs every 10 minutes, fetching content from news feeds. If the HTTP request succeeds, the system proceeds to parse the feed/page and normalize essential fields such as title, summary, URL, timestamp, and source. Before insertion, it checks for duplicates in RawNewsFeed; if found, the item is skipped, otherwise, it is inserted into the database. In case of a failed HTTP request, the system logs an error and applies a backoff mechanism to retry gracefully. This ensures the database remains consistent, up-to-date, and free from redundant entries.

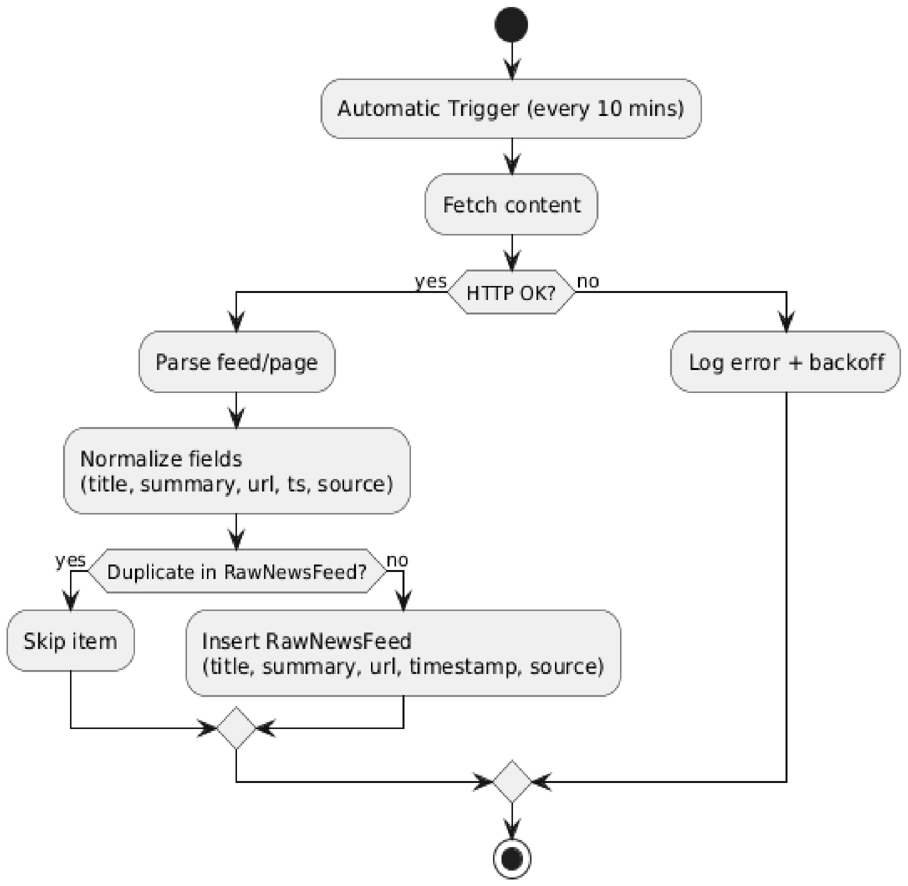
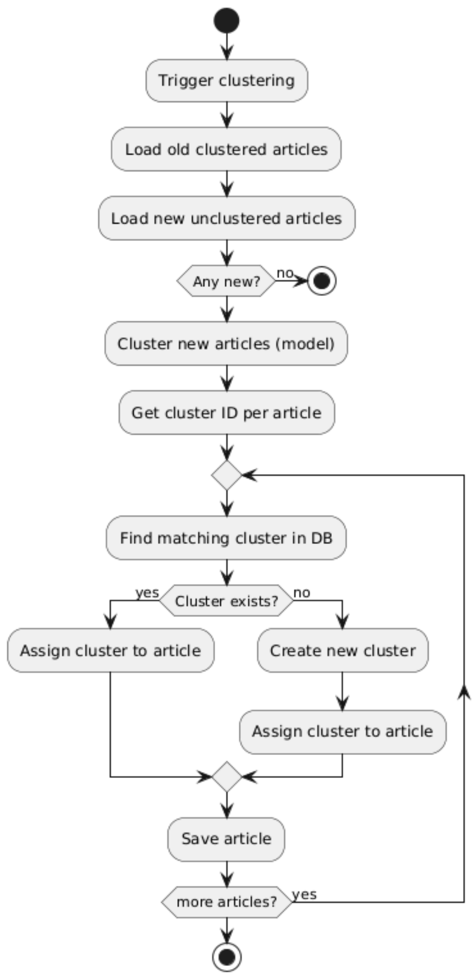

Automatic Trigger (every 10 mins)
Fetch content
yes
HTTP OK?
no
Parse feed/page
Log error + backoff
Normalize fields
(title, summary, url, ts, source)
yes
Duplicate in RawNewsFeed?
no
Skip item
Insert RawNewsFeed
(title, summary, url, timestamp, source)

Figure 3: Data Scraping Activity Diagram

Clustering Activity Diagram

The diagram in Figure 4: Clustering Activity Diagram, depicts the event clustering pipeline in NewsInsights. The process begins with a clustering trigger, which loads previously clustered articles alongside new, unclustered ones. If new articles are present, the system applies the clustering model to assign a cluster ID to each article. It then checks the database for matching clusters, if a suitable cluster exists, the article is added; if not, a new cluster is created and the article is assigned accordingly. Each processed article is saved, and the pipeline iterates until all new articles are clustered. This ensures that coverage of the same event (from multiple outlets) is grouped together, enabling meaningful cross-source comparisons and bias detection.



Trigger clustering
Load old clustered articles
Load new unclustered articles
Any new?
no
Cluster new articles (model)
Get cluster ID per article
Find matching cluster in DB
Cluster exists?
yes
no
Assign cluster to article
Create new cluster
Assign cluster to article
Save article
more articles?
yes

Figure 4: Clustering Activity Diagram

Recommendation System Activity Diagram

The diagram in Figure 5: Recommendation System Activity Diagram outlines the workflow of the personalized news recommendation engine. The process initiates upon an authenticated request, where it first parses parameters defining the response limit and the days for content recency. It then queries the user's historical interaction events to compile a comprehensive set of news cluster IDs with which they have previously engaged. This initial phase includes a critical branching condition to handle the cold start problem: if a user has no prior interaction data, a profile cannot be generated, and the system immediately returns an empty recommendation set, terminating the workflow. For users with a history, this collection of interacted cluster IDs serves as the foundational data for constructing a personalized interest model.

The core of the recommendation logic lies in generating a user profile vector, a high-dimensional numerical representation of the user's latent interests. To achieve this, the system retrieves the pre-computed vector embeddings for all news clusters in the user's interaction history. It then computes the mean of these embedding vectors to create a single, averaged vector. This vector is subsequently subjected to L2 normalization to ensure its magnitude does not skew similarity calculations. The resulting normalized vector acts as a sophisticated digital fingerprint of the user's content preferences, effectively modeling their position within the multi-dimensional semantic space of the news corpus.

With the user profile vector computed, the final phase involves candidate retrieval, ranking, and filtering. The vector is used to perform a similarity search against the entire corpus of news cluster embeddings, explicitly excluding any clusters the user has already seen to ensure novelty in the recommendations. The ranking is executed using cosine similarity, which identifies the candidate clusters that are most semantically aligned with the user's profile. This ranked list undergoes a crucial post-processing step: a timeliness filter is applied to discard any cluster that does not contain articles published after a pre-calculated cutoff date. Finally, the validated, relevant, and timely clusters are serialized into the required JSON format and returned to the user as their personalized feed.

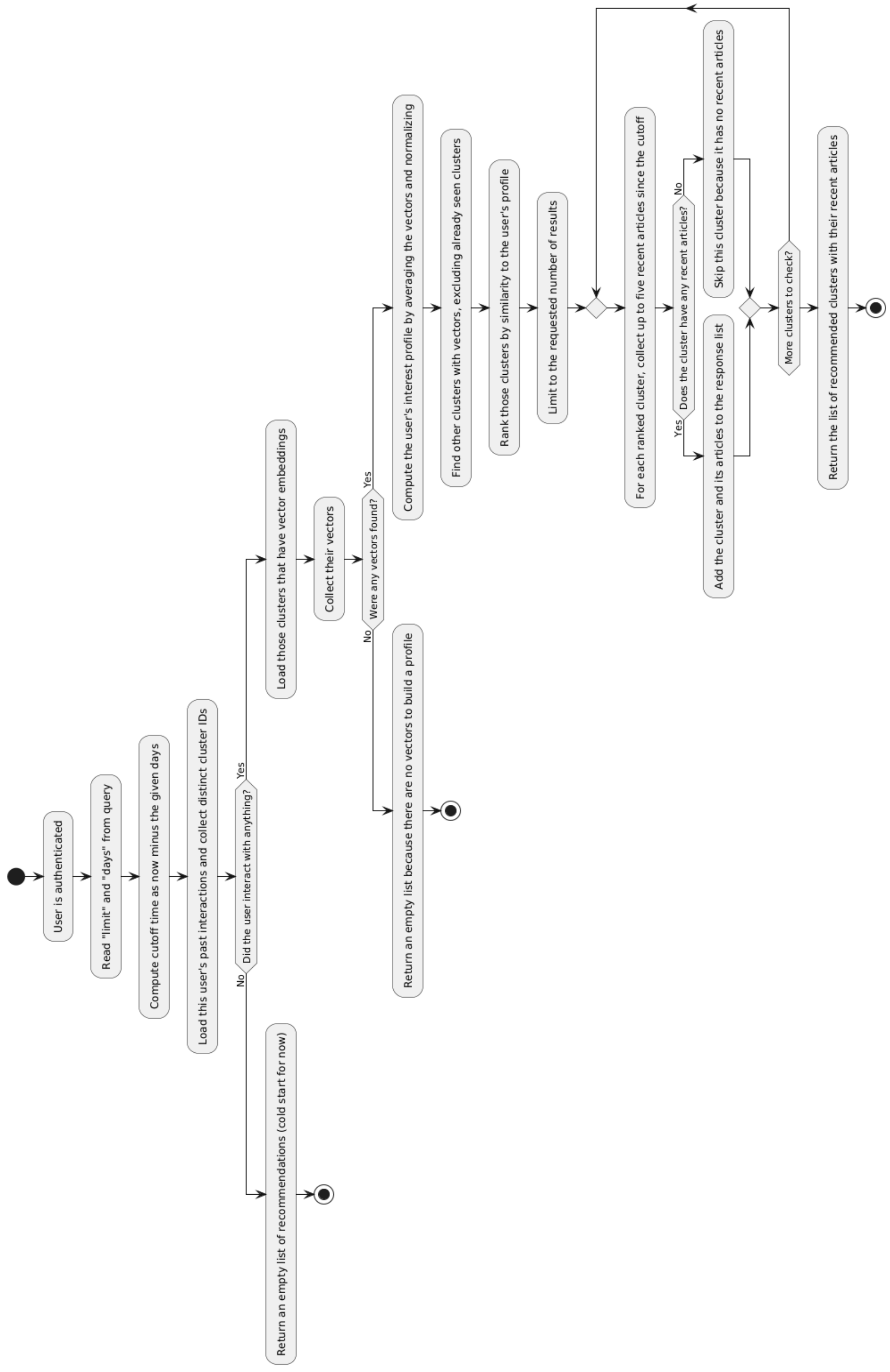

User is authenticated
Read "limit" and "days" from query
Compute cutoff time as now minus the given days
Load this user's past interactions and collect distinct cluster IDs
Did the user interact with anything?
No
Yes
Return an empty list of recommendations (cold start for now)
Load those clusters that have vector embeddings
Collect their vectors
Were any vectors found?
No
Yes
Return an empty list because there are no vectors to build a profile
Compute the user's interest profile by averaging the vectors and normalizing
Find other clusters with vectors, excluding already seen clusters
Rank those clusters by similarity to the user's profile
Limit to the requested number of results
For each ranked cluster, collect up to five recent articles since the cutoff
Does the cluster have any recent articles?
Yes
No
Add the cluster and its articles to the response list
Skip this cluster because it has no recent articles
More clusters to check?
Return the list of recommended clusters with their recent articles

Figure 5: Recommendation System Activity Diagram

ER Diagram

The diagram in Figure 6: ER Diagram illustrates the Entity–Relationship (ER) model for the NewsInsights system. At its core, the schema connects users, events, and clustered news data.

1. The User entity stores account information (username, password, email) and is linked to multiple bookmarks and user events.
2. Bookmark allows users to save clusters for later, recording the cluster ID and creation timestamp.
3. UserEvent captures interactions (e.g., clicks, views) tied to a user and a specific cluster, with event type and timestamp for analytics.
4. RawNewsFeed stores ingested articles with metadata (title, content, source, category, sentiment, embedding vectors) and links each article to its corresponding cluster.
5. Cluster represents groups of related articles, storing summary, sentiment, total views, and embedding vectors for semantic similarity.

Together, this ER model ensures that articles are systematically organized into clusters, while also enabling personalization through user-specific events and bookmarks.

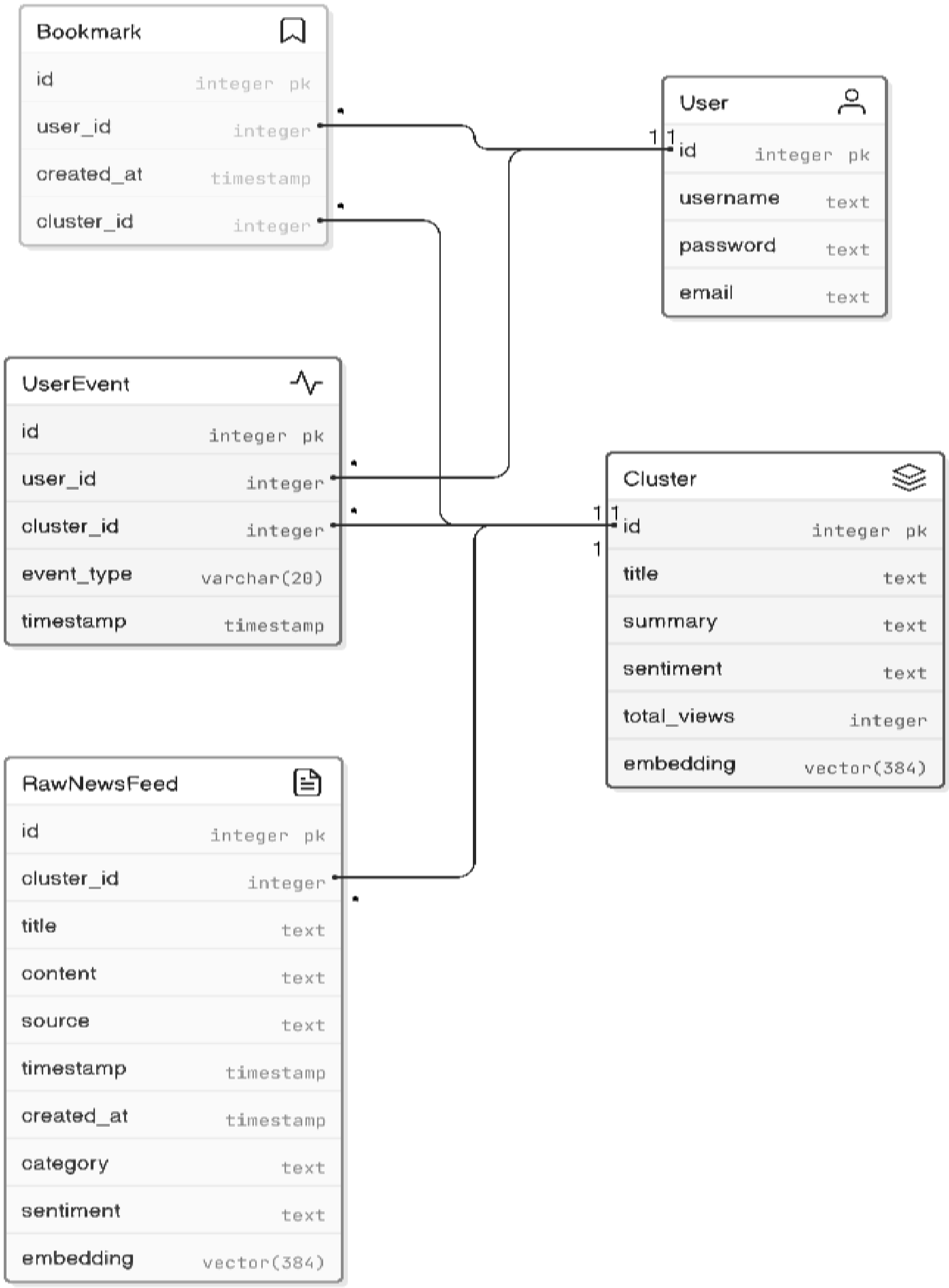


Figure 6: ER Diagram

System State Diagram

The diagram in Figure 7: System State Diagram provides a comprehensive overview of the NewsInsights platform's operational flows, segmented into four primary domains: Account Management, System Processing, Content Exploration, and Personalization. The Account Management section outlines the user authentication lifecycle, beginning with an unregistered user who can register an account. Once registered, the user enters a "Start Not Authenticated" state, from which they can log in. A successful authentication initiates an active session, which can be refreshed to maintain login status or terminated via logout. This section also includes auxiliary flows for checking username/email availability and resetting a forgotten password, encapsulating all user identity and access control processes.

The System Processing block details the platform's automated, continuous content pipeline, which operates independently of user sessions. This lifecycle begins with the ingestion phase, where articles are scraped from various online news sources. These articles are then aggregated, and AI-driven processes generate concise summaries and analyze sentiment. Following this, related articles are grouped into coherent Clusters. In the final stages, both the individual articles and the newly formed clusters are converted into numerical vector embeddings, and the content is filtered to ensure only recent and relevant information is presented to the user, making it ready for discovery and recommendation.

The Content Exploration and Bookmarking sections describe the primary ways an authenticated user interacts with the platform's content. The core loop involves a user browsing news clusters and selecting one to view its detailed summary and source articles. Parallel to this, the bookmarking feature allows a user to check the bookmark status of any cluster and perform actions such as adding it to their saved list, viewing their collection of saved bookmarks, or removing an item. This flow represents the direct, manual engagement a user has with the aggregated news content, forming the basis for their activity profile.

Finally, the Personalization domain illustrates the platform's intelligent features, which create a tailored user experience. Every time a user views a cluster's details, their engagement is tracked. This tracked data serves as a direct input for the recommendation engine, which processes the user's interaction history to generate and present personalized news recommendations. This creates a critical feedback loop where a user's browsing and reading habits actively refine the content they are shown in the future, transforming the platform from a static aggregator into a dynamic, responsive news discovery tool.

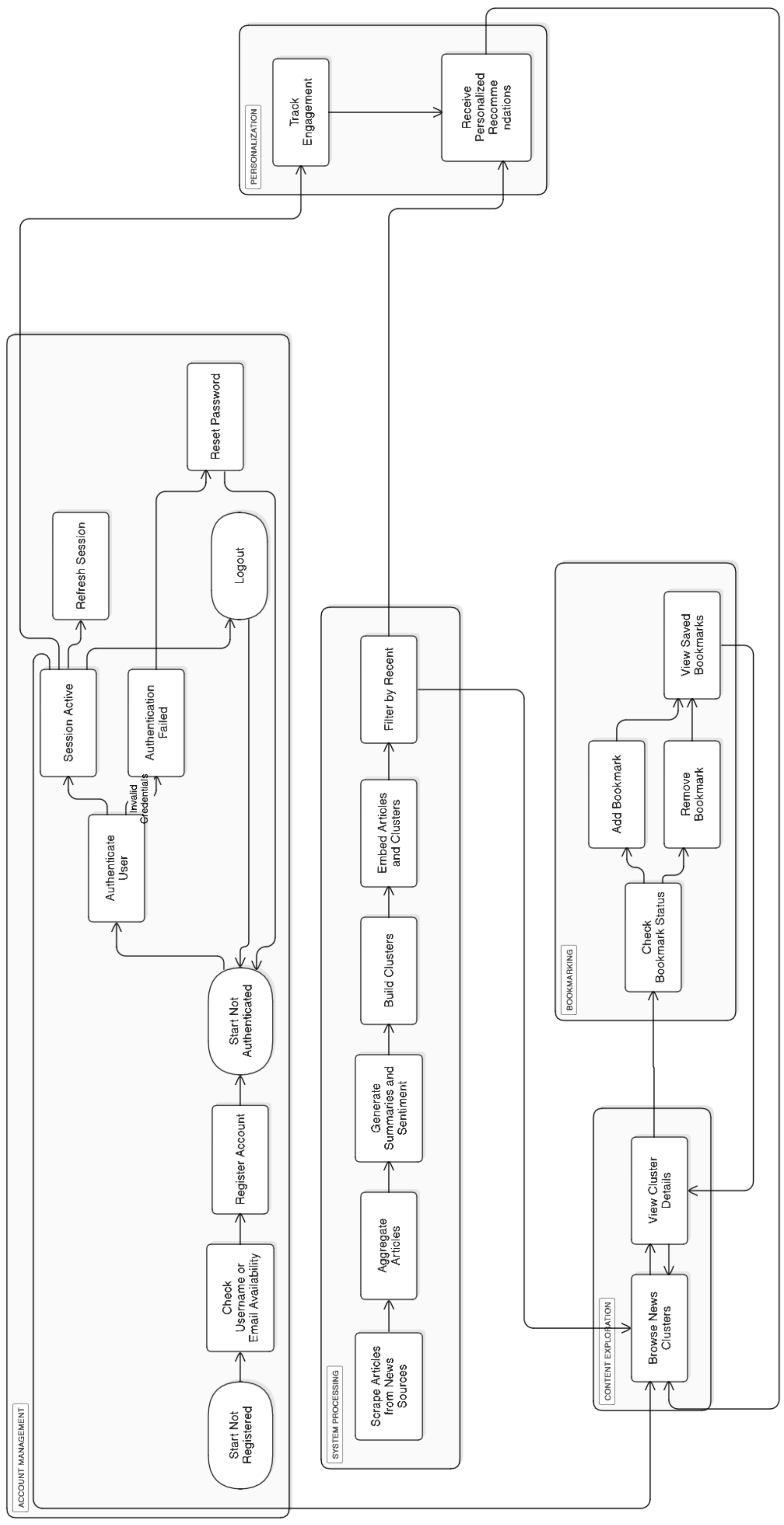


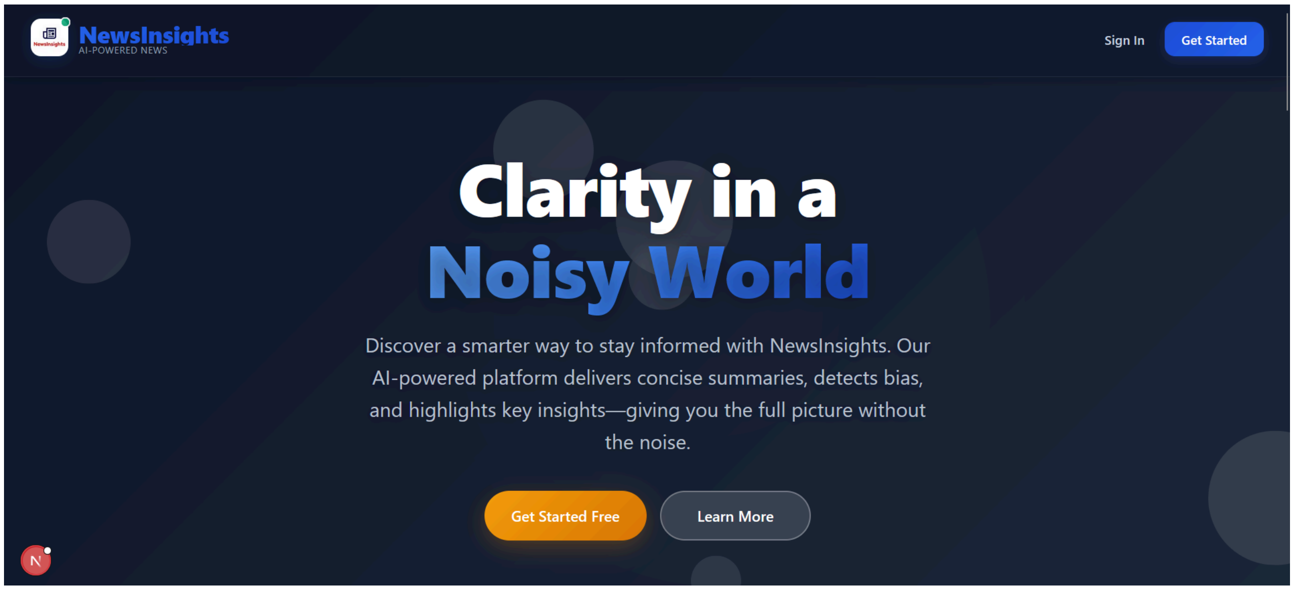
Figure 7: System State Diagram

4.4 Snapshots of Working Prototype

Home Page

This is the main home page for the NewsInsights platform, serving as the primary entry point for new and returning visitors. It's designed to quickly convey the core value of the service and guide users toward signing up. Scrolling down the page reveals additional sections that build on this initial pitch:

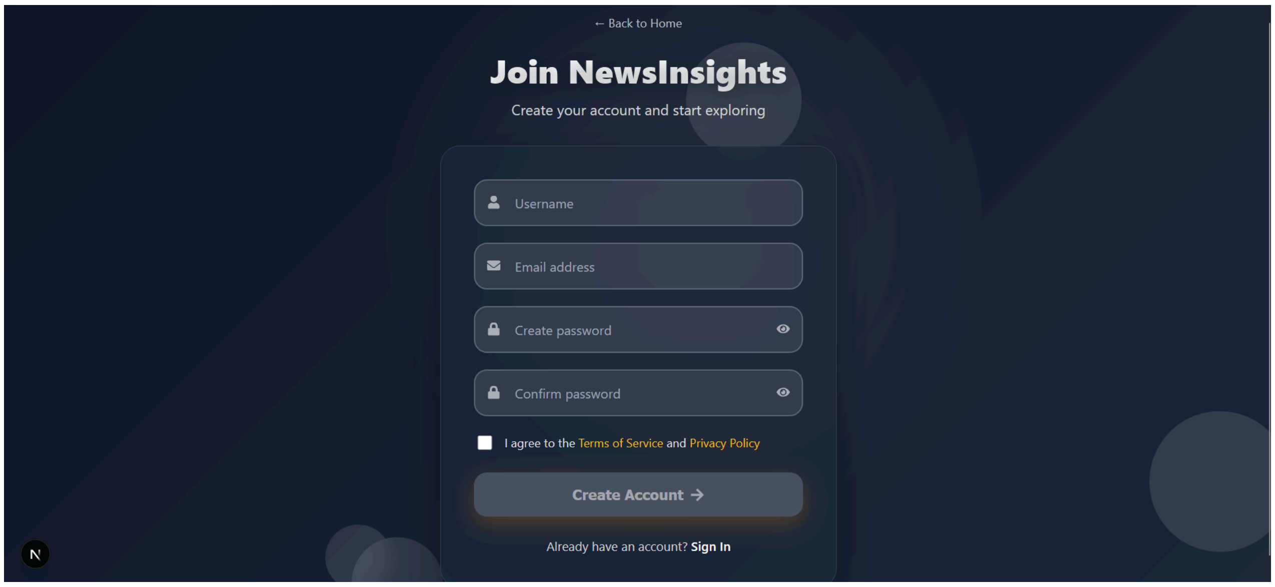
1. News Sources: A section dedicated to showcasing the logos and names of the various news outlets the platform aggregates data from. This builds credibility and transparency.
2. Features Offered: An in-depth breakdown of the platform's key features, likely with visuals and descriptions for functionalities like AI summarization, clustering, sentiment analysis, and the personalized For You feed.
3. Contact Us Form: At the bottom of the page, a form allows users to send inquiries, feedback, or support requests directly to the team.



Discover a smarter way to stay informed with NewsInsights. Our AI-powered platform delivers concise summaries, detects bias, and highlights key insights—giving you the full picture without the noise.

Register Page

1. The form requires the user to provide the following information:
2. Username: A unique identifier for the user's account. The system performs a check during registration to ensure that the chosen username is not already in use by another member, guaranteeing each user has a distinct identity on the platform.
3. Email address: Used for account verification, communication, and password recovery.
4. Create password & Confirm password: The user must enter a password twice to ensure accuracy and prevent typing errors. The eye icon allows the user to toggle the password's visibility to check their input.



← Back to Home
Join NewsInsights
Create your account and start exploring
Username
Email address
Create password
Confirm password
I agree to the Terms of Service and Privacy Policy
Create Account →
Already have an account? Sign In

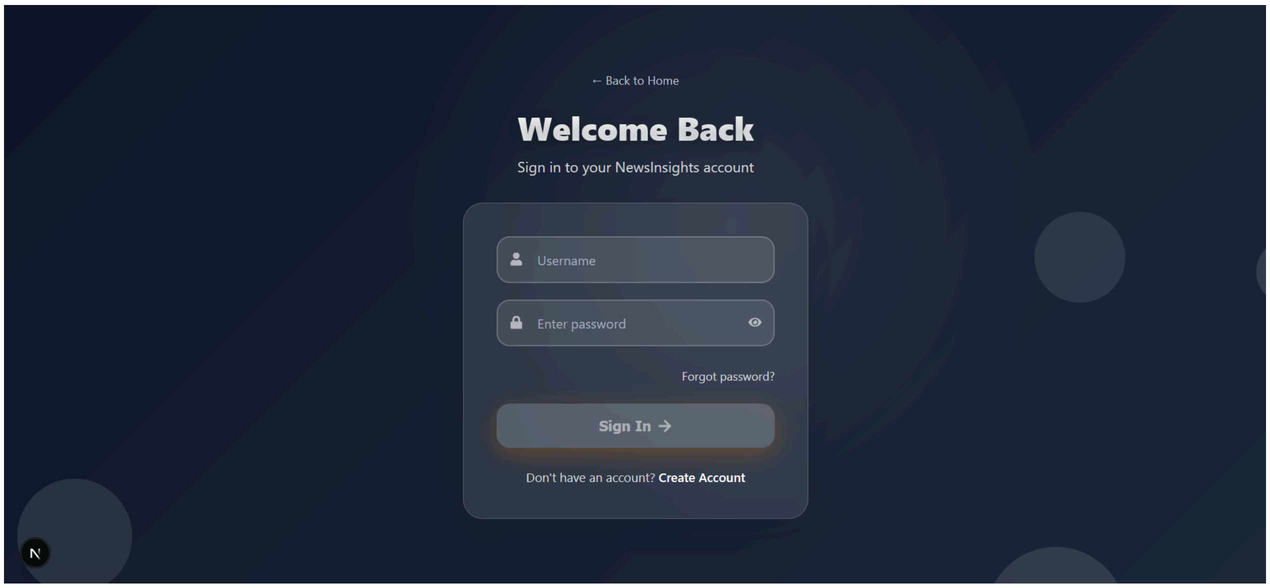
Login Page

The login form consists of:

1. Username: The field where the user enters their unique, registered username.
2. Enter password: The field for the user's password. It includes an eye icon that allows the user to toggle the password's visibility to ensure it is typed correctly.

Below the input fields, there are several options:

1. Forgot password?: A crucial feature for account recovery. If a user has forgotten their password, they can click this link. This action initiates a process where a password reset link is sent to the user's registered email address.
2. Sign In button: The primary action button that the user clicks after entering their credentials to log into the platform.
3. Create Account link: For individuals who do not yet have an account, this link directs them to the registration page.



← Back to Home
Welcome Back
Sign in to your NewsInsights account
Username
Enter password
Forgot password?
Sign In →
Don't have an account? Create Account

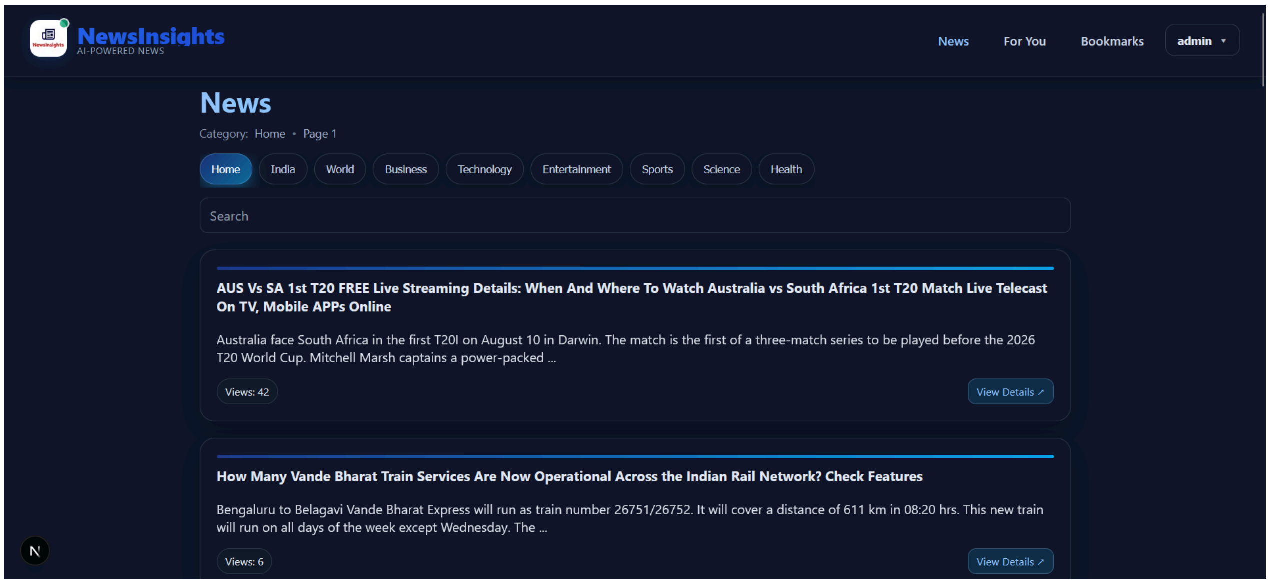
News Page

This page presents the latest news after it has been processed by the platform's AI engine. The articles shown are the result of a two-step process:

1. Aggregation: Content is automatically collected from numerous online news sources.
2. Clustering: Related articles covering the same event are grouped together into a single, coherent story.

Key features visible on the page include:

1. Main Navigation: At the top right, users can navigate between the general News feed, a personalized For You section, their saved Bookmarks, and an Admin panel.
2. Category Filters: A dedicated bar allows users to filter the news by various categories such as Home, India, World, Business, Technology, and Sports, enabling them to focus on topics of interest.
3. Search Functionality: A search bar is prominently placed to help users find specific news stories quickly.
4. Article Cards: The news is presented in a card-based layout. Each card represents a clustered news story and provides:
    a. A clear Title.
    b. A short summary snippet of the content.
    c. An engagement metric, like the number of Views.
    d. A View Details button, which would likely navigate the user to a page with the full summary and links to the original source articles.



News

Category: Home • Page 1

AUS Vs SA 1st T20 FREE Live Streaming Details: When And Where To Watch Australia vs South Africa 1st T20 Match Live Telecast On TV, Mobile APPs Online

Australia face South Africa in the first T20I on August 10 in Darwin. The match is the first of a three-match series to be played before the 2026 T20 World Cup. Mitchell Marsh captains a power-packed ...

Views: 42

How Many Vande Bharat Train Services Are Now Operational Across the Indian Rail Network? Check Features

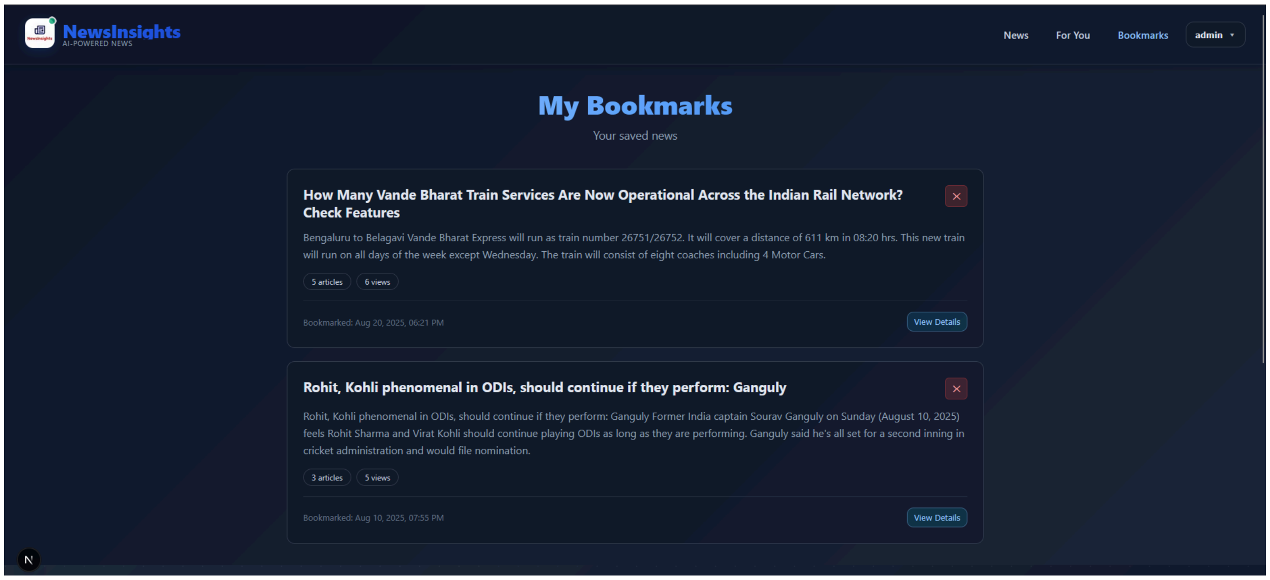
Bengaluru to Belagavi Vande Bharat Express will run as train number 26751/26752. It will cover a distance of 611 km in 08:20 hrs. This new train will run on all days of the week except Wednesday. The ...

Views: 6

Bookmarks Page

The key purpose of this page is to provide users with quick access to their curated content. Features shown on the page include:

1. Page Title: A clear heading "My Bookmarks" with the sub-heading "Your saved news" immediately informs the user of the page's function.
2. Saved Article Cards: Each bookmarked item is presented as a card, consistent with the main news feed's design. Each card contains detailed information:
    a. The Title and a summary of the news story.
    b. Metadata about the cluster, such as the number of articles it contains and its total views.
    c. A timestamp indicating exactly when the article was bookmarked.
    d. A View Details button to navigate to the full story.
3. Remove Bookmark: A distinct 'x' icon is located in the top-right corner of each card, allowing the user to easily remove the article from their saved list.



My Bookmarks

Your saved news

How Many Vande Bharat Train Services Are Now Operational Across the Indian Rail Network? Check Features

Bengaluru to Belagavi Vande Bharat Express will run as train number 26751/26752. It will cover a distance of 611 km in 08:20 hrs. This new train will run on all days of the week except Wednesday. The train will consist of eight coaches including 4 Motor Cars.

5 articles 6 views

Bookmarked: Aug 20, 2025, 06:21 PM

View Details

Rohit, Kohli phenomenal in ODIs, should continue if they perform: Ganguly

Rohit, Kohli phenomenal in ODIs, should continue if they perform: Ganguly Former India captain Sourav Ganguly on Sunday (August 10, 2025) feels Rohit Sharma and Virat Kohli should continue playing ODIs as long as they are performing. Ganguly said he's all set for a second inning in cricket administration and would file nomination.

3 articles 5 views

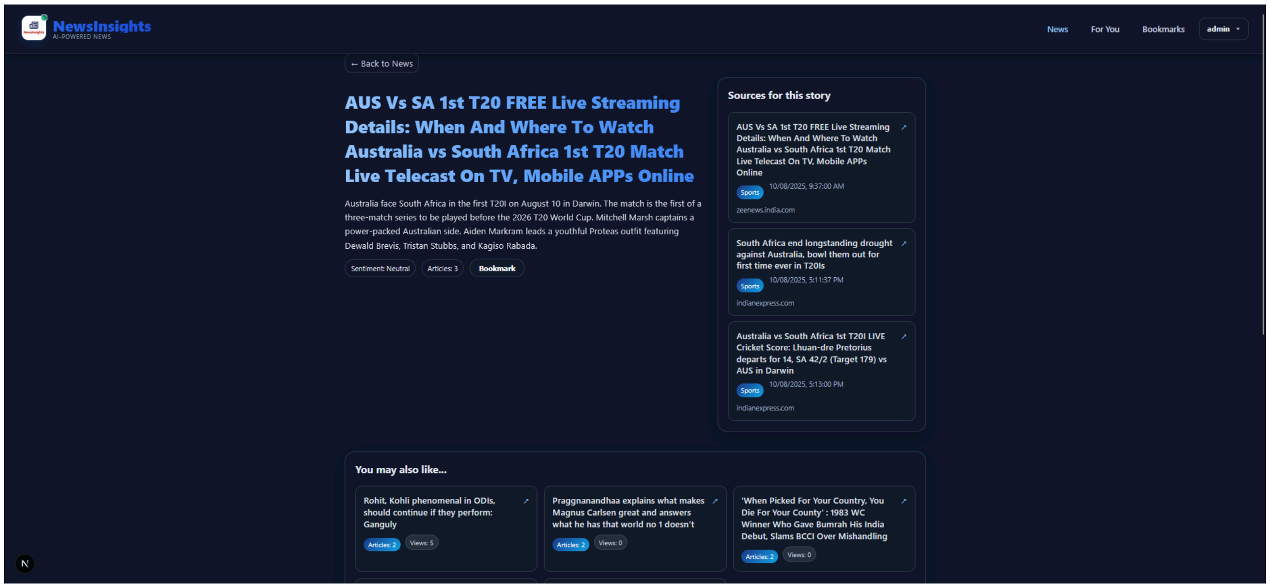
Bookmarked: Aug 10, 2025, 07:55 PM

View Details

Detailed News

The key components of this page are:

1. Main Story View (Left):
    a. Title: The prominent headline of the clustered news story.
    b. AI-Generated Summary: A detailed summary is displayed, which is automatically created to consolidate information from all the source articles.
    c. Insights and Actions: Below the summary, several AI-driven data points will be shown, such as the overall Sentiment of the story (e.g., "Neutral") and Political Bias. There is also a Bookmark button for the user to save the story.
2. Sources Panel (Right):
    a. The "Sources for this story" section provides full transparency by listing all the original articles used to create the summary.
    b. Each entry shows the source's headline and domain name (e.g., zeenews.india.com, indianexpress.com). Clicking on any of these links redirects the user to the original article on the source's website.
3. Recommendation Section (Bottom):
    a. The "You may also like..." section is designed to keep the user engaged.
    b. This feature suggests other news stories that are both recent and belong to the same category as the article currently being viewed, providing relevant content for further reading.



AUS Vs SA 1st T20 FREE Live Streaming Details: When And Where To Watch Australia vs South Africa 1st T20 Match Live Telecast On TV, Mobile APPs Online

Australia face South Africa in the first T20I on August 10 in Darwin. The match is the first of a three-match series to be played before the 2026 T20 World Cup. Mitchell Marsh captains a power-packed Australian side. Aiden Markram leads a youthful Proteas outfit featuring Dewald Brevis, Tristan Stubbs, and Kagiso Rabada.

Sentiment: Neutral | Articles: 3 | Bookmark

Sources for this story

AUS Vs SA 1st T20 FREE Live Streaming Details: When And Where To Watch Australia vs South Africa 1st T20 Match Live Telecast On TV, Mobile APPs Online
Sports 10/08/2025, 9:37:00 AM
zeenews.india.com

South Africa end longstanding drought against Australia, bowl them out for first time ever in T20Is
Sports 10/08/2025, 5:11:37 PM
indianexpress.com

Australia vs South Africa 1st T20I LIVE Cricket Score: Lhuan-dre Pretorius departs for 14, SA 42/2 (Target 179) vs AUS in Darwin
Sports 10/08/2025, 5:13:00 PM
indianexpress.com

You may also like...

Rohit, Kohli phenomenal in ODIs, should continue if they perform: Ganguly
Articles: 2 Views: 5

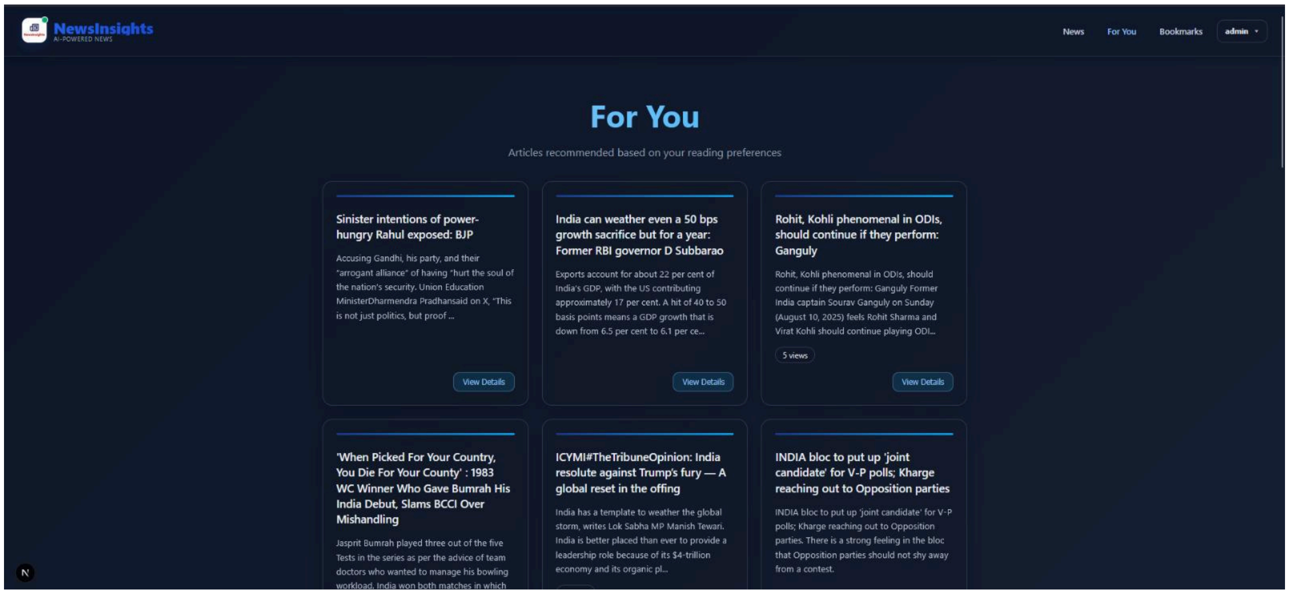
Praggnanandhaa explains what makes Magnus Carlsen great and answers what he has that world no 1 doesn’t
Articles: 2 Views: 0

'When Picked For Your Country, You Die For Your County' : 1983 WC Winner Who Gave Bumrah His India Debut, Slams BCCI Over Mishandling
Articles: 2 Views: 0

For You Page

Key aspects of this page include:

1. Personalized Curation: Unlike the general "News" feed which shows the latest aggregated articles for everyone, the content on this page is uniquely tailored to the individual user.
2. Content Discovery: The recommendation engine helps users discover new articles that align with their interests but which they might not have found by browsing general categories.
3. Standard Layout: The recommended articles are presented in a clean grid layout using the same card-based design seen elsewhere on the site. Each card provides a title, a short summary, and a "View Details" button for a consistent user experience.



For You

Articles recommended based on your reading preferences

Sinister intentions of power-hungry Rahul exposed: BJP

Accusing Gandhi, his party, and their "arrogant alliance" of having "hurt the soul of the nation's security. Union Education MinisterDharmendra Pradhansaid on X, "This is not just politics, but proof ...

View Details

India can weather even a 50 bps growth sacrifice but for a year: Former RBI governor D Subbarao

Exports account for about 22 per cent of India's GDP, with the US contributing approximately 17 per cent. A hit of 40 to 50 basis points means a GDP growth that is down from 6.5 per cent to 6.1 per ce...

View Details

Rohit, Kohli phenomenal in ODIs, should continue if they perform: Ganguly

Rohit, Kohli phenomenal in ODIs, should continue if they perform: Ganguly Former India captain Sourav Ganguly on Sunday (August 10, 2025) feels Rohit Sharma and Virat Kohli should continue playing ODI...

5 views

View Details

'When Picked For Your Country, You Die For Your County' : 1983 WC Winner Who Gave Bumrah His India Debut, Slams BCCI Over Mishandling

Jasprit Bumrah played three out of the five Tests in the series as per the advice of team doctors who wanted to manage his bowling workload. India won both matches in which

ICYMI#TheTribuneOpinion: India resolute against Trump's fury — A global reset in the offing

India has a template to weather the global storm, writes Lok Sabha MP Manish Tewari. India is better placed than ever to provide a leadership role because of its $4-trillion economy and its organic pl...

INDIA bloc to put up 'joint candidate' for V-P polls; Kharge reaching out to Opposition parties

INDIA bloc to put up 'joint candidate' for V-P polls; Kharge reaching out to Opposition parties. There is a strong feeling in the bloc that Opposition parties should not shy away from a contest.

Conclusions and Future Scope

5.1 Work Accomplished

1. Efficient News Aggregation (Completed)
   A robust aggregation pipeline has been implemented to collect news articles from multiple sources in real-time. This ensures that users are continuously updated with the latest and most relevant information, addressing the issue of fragmented access to news

2. AI-Powered Summarization, Sentiment Analysis & Keyword Extraction (Partially Completed)
   AI-driven summarization and sentiment analysis have been integrated, providing concise article overviews and insights into the emotional tone of content. However, the keyword extraction module still needs to be developed to highlight essential topics and improve discoverability.

3. Political Bias Detection (Pending)
   A custom machine learning model for political bias detection has been conceptualized, aimed at classifying articles into Left, Right, or Neutral categories. This feature is intended to provide transparency in news reporting and help users assess different viewpoints before forming opinions. However, the model has not yet been developed or integrated into the system.

4. Personalized News Recommendations (Completed, scope for improvement)
   An AI-powered recommendation engine has been built to suggest articles based on user preferences, reading history, and semantic similarity. The module works effectively but can be enhanced by assigning weights to different user interaction events for more nuanced personalization.

5. Content Filtering & Customization (Partially Completed)
   Users are currently able to filter news content based on categories, improving navigation and personalization. Future iterations will extend filtering to include sources, sentiment, and bias levels, offering more flexibility in customizing the news experience.

5.2 Conclusions

The development of NewsInsights demonstrates the potential of integrating artificial intelligence with real-time news aggregation to deliver a more transparent, personalized, and insightful news consumption experience. The system successfully implements efficient aggregation, AI-powered summarization, and a recommendation engine that adapts to user preferences, thereby addressing the challenges of information overload and biased reporting.

While modules such as political bias detection are still under progress, the current implementation already showcases the platform's value in enhancing user engagement, improving accessibility of information, and fostering informed decision-making. The project highlights how combining automation, natural language processing, and personalization techniques can transform the way users interact with digital news platforms.

5.3 Environmental, Economic and Social Benefits

1. Environmental Benefits: By promoting digital-first consumption of news through aggregation, the system reduces reliance on physical newspapers, thereby lowering paper usage and the associated carbon footprint of print and distribution.
2. Economic Benefits: The platform supports efficient access to diverse news sources in a single interface, reducing the cost and time of information gathering for users. Additionally, the underlying AI modules—such as recommendation and summarization—demonstrate potential for scalable deployment in media and content industries, offering opportunities for monetization and cost-effective operations.
3. Social Benefits: NewsInsights enhances informed decision-making by addressing political bias, highlighting key insights, and providing personalized recommendations. This contributes to combating misinformation, fostering transparency in media consumption, and empowering users to access balanced perspectives. Ultimately, it encourages democratization of information, making quality news accessible to a wider audience.

5.4 Future Work Plan

The current implementation of the NewsInsights system provides a strong baseline for news aggregation and intelligent content delivery. However, additional work is planned to extend its functionality, scalability, and usability:

1. Completion of NLP Features – Implementation of keyword extraction and refinement of summarization and sentiment analysis modules to improve accuracy and searchability.
2. Political Bias Detection – Development and integration of a robust bias detection model capable of classifying articles into Left, Right, and Neutral categories, supported by diverse training datasets.
3. Advanced Personalization – Enhancement of the recommendation engine through weighted user interactions (e.g., clicks, likes, bookmarks, shares), combined with trending content for balanced personalization.
4. Expanded Content Filtering – Addition of filters based on political bias, sentiment, and source credibility, including a verified news mode to prioritize fact-checked sources.
5. Scalability and Platform Expansion – Optimization of backend systems and making the frontend responsive for mobile applications.

REFERENCES

[1]: Media bias - Wikipedia, accessed August 21, 2025, https://en.wikipedia.org/wiki/Media_bias

[2]: Agenda-Setting Theory - Mass Communication Theory, accessed August 21, 2025, https://masscommtheory.com/theory-overviews/agenda-setting-theory/

[3]: Framing Theory - Oxford Research Encyclopedia of Communication, accessed August 21, 2025, https://oxfordre.com/communication/

[4]: Detecting and mitigating bias in NLP - Brookings Institution, accessed August 21, 2025, https://www.brookings.edu/articles/detecting-and-mitigating-bias-in-natural-language-processing/

[5]: Types of Media Bias - McGill-Toolen, accessed August 21, 2025, https://www.mcgill-toolen.org/ourpages/users/tenhunm/ap_us_gov/Chapter%2012%20The%20Media/Types%20of%20media%20bias.pdf

[6]: Political News Bias Detection using ML - Earlham College, accessed August 21, 2025, https://portfolios.cs.earlham.edu/wp-content/uploads/2018/12/senior-thesis-political.pdf

[7]: Volf, M., & Simko, J. (2025). Political Leaning and Politicalness Classification of Texts - arXiv, accessed August 21, 2025, https://arxiv.org/abs/2507.13913

[8]: Sun, Y., Huang, Q., Tung, A. K. H., & Yu, J. (2025). PRISM: A Framework for Producing Interpretable Political Bias Embeddings - ACL 2025, accessed August 21, 2025, https://aclanthology.org/2025.acl-long.1344.pdf

[9]: Maab, I., Marrese-Taylor, E., & Matsuo, Y. (2023). Target-Aware Contextual Political Bias Detection in News - IJCNLP, accessed August 21, 2025, https://aclanthology.org/2023.ijcnlp-main.50/

[10]: Trhlik, F., & Stenetorp, P. (2024). Quantifying Generative Media Bias with a Corpus of Real-world and Generated News Articles - EMNLP, accessed August 21, 2025, https://aclanthology.org/2024.findings-emnlp.255.pdf

[11]: Sar, S., & Roy, D. (2024). Navigating Nuance: In Quest for Political Truth - JCDL, accessed August 21, 2025, https://arxiv.org/abs/2501.00782

[12]: Unpacking Political Bias in LLMs: A Cross-Model Comparison on U.S. Politics - arXiv, accessed August 21, 2025, https://arxiv.org/abs/2412.16746

[13]: Detecting and Mitigating Bias in NLP Models - Brookings Institution, accessed August 21, 2025, https://www.brookings.edu/articles/detecting-and-mitigating-bias-in-natural-language-processing/

[14]: Bias in Text Analysis for International Relations Research - Oxford Academic, accessed August 21, 2025, https://academic.oup.com/isagsq/article/2/3/ksac021/6584769

[15]: BASIL, BABE, MBIB, AllSides, BigNews, NewsSpectrum, BiasLab, and NewsMediaBias-Plus datasets - accessed August 21, 2025, https://aclanthology.org/

[16]: Hugging Face Transformers, PyTorch, and TensorFlow Frameworks - accessed August 21, 2025, https://huggingface.co/transformers/

[17]: NewsInsights Project Report (2025). Event Clustering & Hybrid Pipeline - TIET, accessed August 21, 2025, https://tiet.ac.in/

[18]:Bias in Text Analysis for International Relations Research - Oxford Academic, accessed August 21, 2025, **https://academic.oup.com/isagsq/article/2/3/ksac021/6584769**

[19]: arxiv.org, accessed August 21, 2025, **https://arxiv.org/html/2501.00782v1**

[20]:Detecting Bias in the Media, accessed August 21, 2025, **https://www.edu.gov.mb.ca/k12/cur/socstud/foundation_gr9/blms/9-1-3g.pdf**