Problem Statement: AI-Powered Personalised News Application with Bias Detection and Deepfake Detection

Group Members:

Siddhant Singh 23CS10085 Utsa Ghosh 23CS10090

ClearView: Your Window to Balanced, Unfiltered News (CVBUN)

Are you fed up with **one-sided news** and the endless doomscrolling that leaves you more confused than informed? Imagine a news feed that truly gets you—a platform that cuts through the clutter and shows you all sides of the story. That's where ClearView comes in. ClearView isn't just another news app; it's your friendly guide in a world where **biased reporting twists the truth and deepfake videos muddy the facts**. We gather articles from a whole bunch of sources and **use smart AI to spot political bias, flag clickbait, and even catch deepfake videos,** so you always know you're getting the full picture—not just what an algorithm thinks you should see.

And we know that reading news should be easy and enjoyable for everyone. That's why ClearView offers accessibility settings like dark mode and custom themes, plus offline reading so you can catch up anytime, anywhere. Add in features like fact-checking, personalized recommendations, and our unique three-column view that clearly separates left, right, and neutral perspectives, and you have a news platform designed to keep you informed and in control.

Welcome to ClearView—news the way it should be: honest, balanced, and made just for you.

Key Features:

*Note: We plan to build both a **web and mobile-based application** but depending on time constraints we may be restricted.*

1. ****Collecting and Displaying News Articles:**
    - We plan to use tools such as **web scraping**, **RSS feeds**, and **API calls** to get the **multimedia news content** for our application.
    - **Web Scraping:** We plan to use **BeautifulSoup or Scrapy** to extract news from websites that do not provide APIs.
    - **RSS Feeds:** We plan to use **feedparser** to extract structured news from RSS feeds.

    - **API Calls:** We plan to combine multiple APIs (**NewsAPI:** newsapi.org; **GNews API:** gnews.io; **Mediastack:** mediastack.com; **New York Times API**: developer.nytimes.com) to get diverse coverage and avoid hitting request limits. We plan to implement caching using Redis to minimize API calls and handle rate limits. Redis (REmote DIctionary Server) is an open-source, in-memory database and cache used for applications that require low latency, as it stores data in memory instead of on disk, making it faster and more reliable.
    - **Processing & Storage:** We plan to store data in a database (**MongoDB or PostgreSQL**), clean and deduplicate articles, and categorise them using **NLP-based topic modeling.**
2. ****Accessibility Settings:**
    - **Dark Mode & Custom Themes:** We plan to offer **UI customization** for a better reading experience through **dark mode** and **custom themes**.
    - **Offline Reading Mode:** We plan to allow users to save articles for **offline access.**
3. ****Personalized News Feed:**
    - **Preferential Filtering:** We plan to provide users with the opportunity to choose their preferences from a variety of categories like sports, politics, and entertainment. We also plan to offer additional choices within these categories to select or get recommended news about their favourite sports, sports teams, or particular political parties (i.e., add certain topics to their favourites).
    - **Content-Based and Collaborative Filtering:** We plan to use machine learning algorithms such as *content-based filtering and collaborative filtering* to recommend news articles based on *user preferences, ratings, comments, reading history, and engagement.*
    - **Usage Tracking:** We plan to provide users with a weekly report of their most read news genres and the respective time spent on them.
4. ****Bias Detection and Analysis:**
    - **Biased News:** We plan to build an **NLP model** to classify articles as **left-leaning**, **right-leaning**, or **politically neutral**, recognizing that in the current world most news sources have either shifted towards the left or the right with very few sharing a neutral view with the general public.
    - **Bias Scores and Percentages:** We plan to provide a bias score with a visualization that includes **colour coding** and **percentages** for various kinds of news (left, right, neutral) when comparing different sources reporting the same event.
    - **Multiperspective News Display:** We plan to provide a display that shows **three different columns of news articles** corresponding to right, left, and neutral perspectives. This feature combats the saturation of one-sided content due to recommendation systems.
    - **PS:** This part of our project is inspired by **Ground News(https://ground.news/)**.
5. ****Identification of Deepfake Videos (Computer Vision):**
    - We plan to implement ***computer vision models*** such as **XceptionNet** to analyse facial inconsistencies and unnatural expressions to **identify potential**

deepfakes. Once a deepfake is recognised, we plan to red flag it for easy identification by users.

6. **Fact-Checking Integration:**
    - **Checking Claims:** We plan to implement a system that automatically cross-checks claims in news articles against verified fact-checking databases like **PolitiFact**, **Snopes**, and **FactCheck.org**.
    - **LLM Integration:** We plan to consider using **LLM-based fact-checking tools** (e.g., integrating **APIs** like **ClaimReview** or the **Google Fact Check API**).
    - We plan to offer user alerts in the form of a fact-check badge or pop-up notification to indicate the authenticity of claims.
7. ****Community for Sharing Views and Interaction Space for Users:**
    - **Rating + Commenting:** We plan to allow users to **add comments and ratings** for every news article, as well as the **option to like and share** the article.
8. **Clickbait Detection:**
    - We plan to use **NLP-based classifiers (BERT, LSTM)** *to analyse headlines for sensational phrases, exaggerated claims, and emotional manipulation*. Once a news article is detected as clickbait, we plan to red flag it for easy identification by users.

Optional Features:

If time permits we would also try to implement some other features which are given below.

9. **Chatbot integration: AI Chat Assistant:**
    - **AI Chat Assistant:** We may also try to add a **chatbot** to summarize or answer questions about articles through chat primarily.
    - **Text-to-Speech (TTS):** This tool can used to **read news aloud** using AI-generated voices
10. **Real-Time Summarization and Sentiment Analysis**:
    - Generate **concise summaries** of articles using AI.
    - **Analyse sentiment** (positive, negative, neutral) to help users gauge the tone of an article.
11. **Multilingual Support with Real-Time Translation**:
    - Aggregate news from **different languages** and translate them in real-time.
    - Primarily we would be working with national or international news as the main focus but later add **regional news** to it also.
    - Allow users to **compare global perspectives** on the same topic.
12. **News Timeline and Trends:**
    - Show how a topic evolved over time using graphs and historical articles.

Expected Tech Stack Options:

- **Frontend (Web):** React.js / Next.js, Tailwind CSS / Material-UI

- **Frontend (Mobile):** Kotlin (Jetpack Compose), Swift (SwiftUI), Flutter / React Native
- **Backend:** FastAPI / Flask / Django
- **Database:** PostgreSQL, MongoDB
- **Caching & Rate Limiting:** Redis
- **News Scraping & Aggregation:** BeautifulSoup, Scrapy, Selenium, Feedparser, NewsAPI, GNews, Mediastack, NY Times API
- **AI for Bias Detection:** BERT / DistilBERT / GPT-based models, VADER, TextBlob
- **Recommendation System:** Scikit-learn, TensorFlow, PyTorch, Content-Based & Collaborative Filtering
- **Fact-Checking:** Google Fact Check API, PolitiFact API
- **Deepfake Detection:** XceptionNet, OpenCV, DeepFace
- **Text-to-Speech & Summarization:** Google TTS / Amazon Polly, Hugging Face Transformers, GPT models
- **Multilingual Support & Translation:** Google Translate API / DeepL API
- **News Trends & Visualization:** D3.js, Plotly, Matplotlib, Seaborn
- **Deployment & DevOps:** AWS / Google Cloud / Azure, Docker & Kubernetes, GitHub Actions / Jenkins, Prometheus / Grafana