News Aggregator Using NLP

Project Report

21AD1513- INNOVATION PRACTICES LAB

Submitted by

DEEPIKA R	**-**	**211422243053**
BENILA M	**-**	**211422243044**
DEEPIKA R K	**-**	**211422243054**

in partial fulfillment of the requirements for the award of degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123

ANNA UNIVERSITY: CHENNAI-600 025

October, 2024

BONAFIDE CERTIFICATE

Certified that this project report titled "**NEWS AGGREGATOR USING NLP**" is the bonafide work of **DEEPIKA R(211422243053), BENILA M(211422243044), DEEPIKA R K (211422243054)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

INTERNAL GUIDE
Mrs. S. Swathi M.E
Assistant Professor
Department of AI &DS

HEAD OF THE DEPARTMENT
Dr. S. MALATHI M.E. , Ph.D
Professor and Head,
Department of AI & DS.

Certified that the candidate was examined in the Viva-Voce Examination held on ………………………

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I also take this opportunity to thank all the Faculty and Non-Teaching Staff Members of Department of Artificial Intelligence and Data Science for their constant support. Finally I thank each and every one who helped me to complete this project. At the outset we would like to express our gratitude to our beloved respected Chairman, **Dr.Jeppiaar M.A.,Ph.D,** Our beloved correspondent and Secretary **Mr.P.Chinnadurai M.A., M.Phil., Ph.D.,** and our esteemed director for their support.

We would like to express thanks to our Principal, **Dr. K. Mani M.E., Ph.D.,** for having extended his guidance and cooperation.

We would also like to thank our Head of the Department, **Dr.S.Malathi M,E.,Ph.D.,** of Artificial Intelligence and Data Science for her encouragement.

Personally we thank **Mrs.S.Swathi M.E** , Assistant Professor, Department of Artificial Intelligence and Data Science for the persistent motivation and support for this project, who at all times was the mentor of germination of the project from a small idea.

We express our thanks to the project coordinator **Ms.K.CHARULATHA M.E.,** Assistant Professor in Department of Artificial Intelligence and Data Science for their Valuable suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, friends, and well-wishers who have helped us for the successful completion of our project.

We also take the opportunity to thank all faculty and non-teaching staff members in our department for their timely guidance in completing our project.

DEEPIKA R(211422243053)

BENILA M(211422243044)

DEEPIKA R K(211422243054)

ABSTRACT

The rise of digital media has led to an overwhelming influx of information, making it challenging for users to efficiently access and consume relevant news. To address this issue, our project introduces a comprehensive news aggregation platform, built on the MERN stack (MongoDB, Express.js, React, and Node.js). This platform collects news from multiple sources via the News API and applies extractive summarization using the TextRank algorithm to generate concise summaries. By leveraging Natural Language Processing (NLP), our system transforms long-form articles into brief, accessible summaries, enhancing user experience by delivering essential information without the need for extensive reading . The project integrates modern web development frameworks to ensure a scalable, responsive, and user-friendly interface. The backend architecture, powered by Node.js and Express, efficiently handles API requests and data management, while MongoDB stores articles and user preferences. The React frontend allows seamless navigation, personalization, and real-time updates, ensuring a cohesive experience. Experimental results show that our summarization approach maintains high relevance, and user feedback indicates a positive reception to the summarized content format. Our platform aims to improve information accessibility, catering to users who seek quick insights into the latest news. This project serves as a foundation for further enhancements, such as personalized content filtering, multilingual support, and advanced NLP-driven summarization techniques.

Keywords : News Aggregator, MERN Stack, News API, Extractive Summarization, Natural Language Processing (NLP)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	**MEANING**
NLP	NATURAL LANGUAGE PROCESSING
VPM	NODE PACKAGE MANAGER
JDK	JAVA DEVELOPMENT TOOL KIT
UI	USER INTERFACE

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

This project is a news aggregation platform designed to deliver summarized news from a variety of reliable sources. Built using the MERN stack, the application is structured to fetch news articles via the News API, process the content, and display concise summaries to users in real-time. The core of this platform is an extractive summarization model that identifies key information within articles using the Text Rank algorithm, allowing users to quickly understand the main points without reading the full text. The platform's architecture includes MongoDB for efficient data storage, Express.js and Node.js to handle backend processes and manage data flows, and React as the frontend framework to deliver a dynamic and interactive user interface. This combination of technologies supports a seamless user experience and allows the platform to scale effectively as more users and data sources are added. This project is not only a practical tool for everyday news consumers but also a contribution to the ongoing development of content aggregation and summarization technologies. By integrating advanced NLP techniques with robust web development frameworks, the platform serves as a basis for future enhancements such as content personalization, multilingual support, and the inclusion of more sophisticated NLP-driven summarization techniques.

1.2 PROBLEM DEFINITION

With the vast and growing amount of information generated daily across numerous digital news sources, readers often struggle to keep up with relevant, reliable content. Traditional news websites and aggregation platforms provide access to numerous articles but lack efficient methods for summarizing information, leading to information overload. Additionally, many platforms do not prioritize conciseness, forcing users to sift through lengthy articles to find key insights. As a result, users are faced with the challenge of managing excessive amounts of information while maintaining awareness of current events in a time-efficient manner.Our project addresses this issue by developing a news aggregator platform that integrates a summarization feature, aimed at condensing articles to provide essential information at a glance. Leveraging the TextRank algorithm for extractive summarization, the platform reduces the need for extensive reading by presenting only the most relevant portions of each article. By combining this summarization approach with a modern, scalable architecture built on the MERN stack, our solution aims to enhance accessibility and usability for individuals who require quick, digestible news updates.This project not only aims to reduce information overload but also strives to improve user satisfaction through an easy-to-navigate, responsive interface. The platform serves as an efficient solution for time-constrained readers, addressing the key challenges of information overabundance, content accessibility, and user engagement in digital news consumption.

1.3 PURPOSE AND OBJECTIVES

The purpose of the news aggregator project is to develop a comprehensive platform that efficiently collects, summarizes, and presents news articles from a diverse array of sources, addressing the challenges posed by information overload in today's fast-paced digital environment. With countless news articles published daily across various domains, users often struggle to find relevant, high-quality content that aligns with their interests. By consolidating information into a single platform, the news aggregator aims to streamline the news consumption process, allowing users to access essential updates without sifting through an overwhelming volume of information.

Key objectives of the news aggregator include the aggregation of content from multiple reputable sources, ensuring users receive diverse perspectives on current events. By leveraging APIs to connect with various news outlets, the platform provides users with a broad spectrum of articles covering different topics, such as politics, technology, health, and entertainment. This approach not only enriches the user experience but also fosters a more informed public by presenting a variety of viewpoints.

Another primary objective is the implementation of extractive summarization techniques, particularly utilizing the TextRank algorithm. This feature is designed to distill lengthy articles into concise summaries that capture the essence of the news without compromising on critical information. For busy individuals seeking quick insights, summarization enhances the overall user experience by allowing them to stay informed while saving time.

User experience is at the forefront of the platform's design philosophy. The news aggregator is committed to creating a user-friendly interface that facilitates seamless navigation and interaction. This includes features such as easy filtering and searching for articles based on categories or keywords, which ensures users can quickly find the content that resonates with their interests. Prioritizing usability and accessibility, the platform aims to increase user engagement and satisfaction, encouraging more frequent visits and interactions.

In addition to a streamlined interface, the aggregator aims to provide real-time updates to keep users informed about the latest news. By implementing mechanisms that continuously fetch and display new articles as they are published, the platform ensures that users can access timely information, particularly regarding rapidly evolving events. This real-time functionality is crucial for users who wish to stay abreast of current developments without needing to refresh or manually check for updates.

Personalization features are another critical aspect of the project. The news aggregator intends to allow users to customize their news feeds based on personal interests and preferences. By enabling users to select specific topics, sources, and categories, the platform can deliver tailored content that enhances user engagement. This customization not only improves the relevance of the news presented but also fosters a sense of ownership and connection with the platform.

To ensure the platform is robust and scalable, the project will focus on developing an architecture that can efficiently handle increased loads and a growing user base. This involves optimizing back-end processes for data storage and retrieval, as well as implementing effective caching mechanisms to improve response times. By ensuring that the platform can grow alongside its user base, the project aims to maintain high performance and user satisfaction even during peak usage times.

Finally, incorporating a feedback mechanism allows users to share their opinions regarding the quality of articles and summaries. This objective not only aids in the continuous improvement of the platform based on user input but also fosters a community among users who can engage with the content in a meaningful way. By actively involving users in the development process through feedback, the news aggregator can adapt to changing preferences and needs.

1.4 IMPORTANCE

Summarization is a critical feature in news aggregation, addressing the challenges posed by the sheer volume of news content produced daily. As readers are inundated with extensive articles from multiple sources, extracting essential information from long-form content becomes increasingly difficult. Summarization, specifically extractive summarization, provides an effective solution by identifying and presenting the most relevant portions of each article, allowing users to quickly comprehend the core message without having to read through the entire text. By implementing summarization algorithms like TextRank, our platform ensures that users gain quick access to high-value content, significantly improving reading efficiency. This not only saves time but also enhances user engagement, as readers are more likely to stay updated on current events when the information is presented in a digestible format. Summarization also reduces information overload by filtering out less critical details, enabling readers to focus on what matters most. In the fast-paced digital age, where time and attention are limited, summarization in news aggregation plays a pivotal role in transforming raw data into concise, accessible insights, ultimately enriching the user experience.

1.5 KEY FEATURES

The news aggregator platform offers several key features designed to enhance the user experience and streamline news consumption. First, it aggregates news from multiple reliable sources via the News API, providing users with diverse content coverage across various topics and perspectives. A standout feature of the platform is its real-time summarization, powered by the TextRank algorithm, which generates concise, extractive summaries of each article. This enables users to quickly grasp main ideas without the need to read lengthy texts, making news consumption more efficient. The platform's frontend, built with React, offers a responsive and interactive interface that adapts seamlessly to different devices, ensuring a smooth experience whether on mobile or desktop. The backend, supported by MongoDB, Express.js, and Node.js, manages data flow and facilitates real-time updates, ensuring that users always have access to the latest news. Moreover, the platform’s MERN stack architecture supports scalability and maintainability, making it well-suited for accommodating a growing user base and for implementing future expansions like personalized recommendations. Lastly, by offering summaries instead of full articles, the platform helps reduce information overload, allowing users to stay informed without feeling overwhelmed. These features together provide a robust, accessible, and engaging news consumption experience, tailored to meet the demands of modern users.

1.6 PROJECT WORKFLOW

The project workflow for the news aggregator platform is designed to facilitate the efficient collection, processing, and presentation of news articles from various sources. It begins with the Data Collection phase, where the platform connects to the News API to fetch articles based on user-defined parameters such as keywords, categories, and preferred sources. This data is then stored in a MongoDB database, ensuring easy retrieval for subsequent processes. In the Data Processing phase, the platform employs the TextRank algorithm for extractive summarization, which analyzes the content of each article to identify key sentences and generate concise summaries. This is crucial for providing users with quick insights into the articles without needing to read them in full.

The User Interface Rendering step involves the frontend, built with React, which fetches the summarized data from the backend and presents it in a user-friendly and visually appealing manner. Users can navigate through the aggregated news articles, view summaries, and access detailed content easily. To keep users engaged, the platform features a Real-Time Updates mechanism that continuously checks for new articles, ensuring that the content displayed is always current without requiring a manual refresh. Furthermore, the User Interaction capabilities allow users to search for specific topics, save articles for later, and share content on social media, all facilitated by efficient state management within the React framework. The backend, developed with Node.js and Express.js, manages Data Management, handling requests between the frontend and database while processing user actions. Finally, Performance Monitoring tools are integrated to track user engagement and system performance, providing insights for ongoing improvements. This comprehensive workflow enables the news aggregator platform to efficiently collect, summarize, and deliver news articles, thus providing users with a valuable and streamlined news consumption experience.

CHAPTER 2

LITERATURE REVIEW

2.1 NEWS AGGREGATOR AND EFFICIENT SUMMARIZATION SYSTEM

The paper discusses the development of a news aggregator designed to compile and summarize global news stories from various sources, addressing the challenges of information overload in today's digital landscape. As the volume of news increases, users often find it difficult to identify relevant content efficiently. The proposed aggregator aims to streamline this process by collecting articles based on specific keywords or phrases and providing concise summaries of the aggregated content using the Text Rank algorithm.

The introduction highlights the growing importance of news in society and the shifting dynamics between traditional newspapers and online media, particularly news aggregators. The paper emphasizes the convenience and time-saving benefits of using a news aggregator, which organizes content by subject and presents it in a user-friendly format.

Summarization is identified as a key requirement for the system, with a focus on extractive summarization methods that preserve the original meaning of the text. The paper outlines its structure, including a review of related work, a detailed description of the proposed system, experimental results demonstrating its effectiveness, and a conclusion discussing potential future developments.

AUTHOR : Alaa Mohamed, Marwan Ibrahim, Mayar Yasser

YEAR : 06 June 2020

2.2 NEWS AGGREGATOR: THE WORLD AT YOUR FINGER TIPS

This paper presents the development of an online news aggregator designed to collect and present news feeds and articles from diverse sources in one centralized location. A news aggregator serves as an online application or software platform that systematically gathers news stories and events from various websites, significantly improving user experience by reducing the time required to access news from multiple publications.

The primary objective of this aggregator is to provide users with a streamlined method of retrieving relevant news articles based on specific input keywords or key phrases. By aggregating content, the platform allows users to avoid the hassle of visiting multiple sites, consolidating their news consumption into a single, efficient source. This capability not only enhances convenience but also ensures that users can keep up with current events without the burden of information overload.

A key feature of the aggregator is its implementation of the TextRank algorithm for summarization. This algorithm has demonstrated promising results in generating coherent and concise summaries of the collected articles. By enhancing the aggregated content, the system produces summaries that are both understandable and efficient, catering to the needs of readers who require quick access to essential information.

The concept of news aggregation is rooted in content syndication, where content created by one or more news-gathering organizations is redistributed by a different entity. This model benefits users by providing access to a wide array of content from various sources without requiring them to navigate to each individual site. As a result, users can visit their preferred news aggregator and access all their favorite publications in one place, streamlining the news consumption process.

Most news aggregator websites do not create original content; instead, they fetch articles from various publications through RSS (Really Simple Syndication) feeds. This approach allows aggregators to present curated news content in a visually appealing manner, enhancing user engagement. Consequently, these platforms are often referred to as RSS feed readers, emphasizing their role in organizing and displaying content from other sources efficiently.

In summary, this paper articulates the need for an effective news aggregator that not only aggregates diverse news articles but also enhances user experience through efficient summarization techniques, thereby addressing the challenges posed by the increasing volume of information available online.

AUTHOR : Liyaqat Fayaz, Iqbal Bashir, Mrs. Sahila

YEAR : December 2021

2.3 NEWS AGGREGATOR WEB APPLICATION USING DJANGO

The delay-tolerant networking routing problem, where messages are to be moved end-to-end across a connectivity graph that is time-varying but whose dynamics may be known in advance. The problem has the added constraints of finite buffers at each node and the general property that no contemporaneous end-to-end path may ever exist. This situation limits the applicability of traditional routing approaches that tend to treat outages as failures and seek to find an existing end-to-end path. We propose a framework for evaluating routing algorithms in such environments. We then develop several algorithms and use simulations to compare their performance with respect to the amount of knowledge they require about network topology. We find that, as expected, the algorithms using the least knowledge tend to perform poorly. We also find that with limited additional knowledge, far less than complete global knowledge, efficient algorithms can be constructed for routing in such environments. To the best of our knowledge this is the first such investigation of routing issues in DTNs.

AUTHOR : Mr Rakesh Kumar Rai ,Dr Isha Singh, Ankit Mudia, Karandeep Bisht

YEAR : 7 July 2021

2.4 ARTIFICIAL INTELLIGENCE BASED NEWS AGGREGATOR

The rapid expansion of information available on the Internet has made manual searching and monitoring of news, blogs, and articles a cumbersome and inefficient task. This paper proposes the development of a customized AI-based news aggregator that focuses on collecting and curating web-based information of strategic, defense, and geopolitical significance, specifically tailored to the needs of military leaders and analysts.

As geopolitical relations and military activities evolve, staying informed with relevant news is essential. Traditional manual monitoring methods are time-consuming and often ineffective. While RSS-based feed readers can gather news, they typically lack the ability to provide curated content based on specific organizational interests. The proposed AI-driven aggregator will address this gap by learning from user preferences and selecting news from both domestic and foreign sources.

The aggregator aims to create a centralized platform where users can access news articles from various sources in one place. Key features will include a user-friendly interface for filtering articles by category, source, and date, as well as real-time updates and customization options. By providing a comprehensive and personalized view of the news, the aggregator will enable users to stay informed about current events while saving time and effort.

The existing systems need improvement in user interaction and content curation. They should allow users to filter articles, search for specific topics, and save content for later. Additionally, the system must prioritize high-quality sources and filter out low-quality information based on user-defined preferences. This enhancement will ensure that users receive relevant and reliable news tailored to their strategic needs.

AUTHOR : Gokula Krishnan T, Prof. N. Sakthivel

YEAR : April 2023

2.5 EFFICIENT DAILY NEWS PLATFORM GENERATION USING NATURAL LANGUAGE PROCESSING

This paper explores the role of automated journalism in today's digital landscape, where online news significantly influences public perception and decision-making, including political elections. As automation becomes increasingly prevalent across various sectors, automated journalism emerges as a critical area of research. The study presents an AI-based information platform designed to automatically generate news articles by analyzing internet trends and mining relevant data from various sources.

Utilizing machine learning and Natural Language Processing (NLP) tools, particularly the NLTK module in Python, the platform aims to create content that resembles human writing both grammatically and linguistically. By processing trends from social media platforms, such as Twitter, the system can classify and categorize information effectively.

Online journalism, which encompasses text, audio, and video content, has gained prominence due to the widespread availability of the internet and advancements in mobile technology. Major media organizations now prioritize their online presence over traditional print formats. News aggregators play a pivotal role by consolidating content from various sources, thus saving users time.

The project emphasizes the importance of automated journalism as a means to alleviate the workload of human journalists, allowing them to focus on higher-level tasks. With the continued advancement of AI and NLP technologies, automated journalism represents a promising frontier for enhancing news generation and delivery.

AUTHOR Ambeth Kumar Visvam Devadoss , Vijay Rajasekar Thirulokachander , Ashok Kumar Visvam Devadoss
YEAR : 14 August 2018

2.6 NLP BASED MACHINE LEARNING APPROACHES FOR TEXT SUMMARIZATION

In an era characterized by an overwhelming amount of information, text summarization has become crucial for extracting relevant insights from extensive texts, such as news articles, blogs, and customer reviews. This review paper investigates various approaches to text summarization, focusing primarily on two key methods: Abstractive (ABS) and Extractive (EXT) summarization. It also explores query-based summarization techniques, which generate summaries based on user-defined topics.

The paper emphasizes structured and semantic-based approaches for summarizing text documents, supported by an analysis of diverse datasets including the CNN corpus and DUC2000. Through a comprehensive study of existing literature, the review outlines the methodologies employed in recent years and highlights their trends, achievements, and potential future developments in the field. Text summarization has gained significant attention across multiple domains, including science, medicine, and law. The techniques analyzed in this paper include machine learning (ML), neural networks (NN), reinforcement learning, sequence-to-sequence modeling, and fuzzy logic. It is noted that while single document summarization methods provide concise summaries for individual texts, multi-document summarization techniques aggregate information from various sources. Query-based models are also discussed, offering tailored summaries based on specific user queries.

In terms of practical implementation, several optimization techniques have been applied to enhance summarization accuracy. Python libraries such as scikit-learn, NLTK, spaCy, and fastai are utilized for natural language processing (NLP) tasks, facilitating the development of effective summarization systems. The paper includes a review of related work in the field, highlighting various methods employed by researchers. For instance, Massimo Mauro and colleagues focus on sentence extraction methods for EXT summarization, scoring sentences for relevance and clustering similar ones to identify the most informative content. Sarda A.T. et al. incorporate neural networks and Rhetorical Structure Theory (RST) to enhance EXT summarization by training models to select appropriate sentences based on learned features. Gabriel Silva's experiments with the CNN corpus leverage feature vectors and dimensionality reduction techniques to improve summarization outcomes. Lastly, Taeho Jo's approach employs the KNN algorithm to classify sentences for inclusion in summaries based on similarity scores. In summary, this paper provides a comprehensive overview of text summarization methodologies, presenting both theoretical frameworks and practical applications, while also examining the current landscape and future directions for research in this essential field.

AUTHOR: Rahul , Surabhi Adhikari , Monika

YEAR : May/Jun 2015

CHAPTER 3

SYSTEM DESIGN

3.1 ARCHITECTURE DIAGRAM

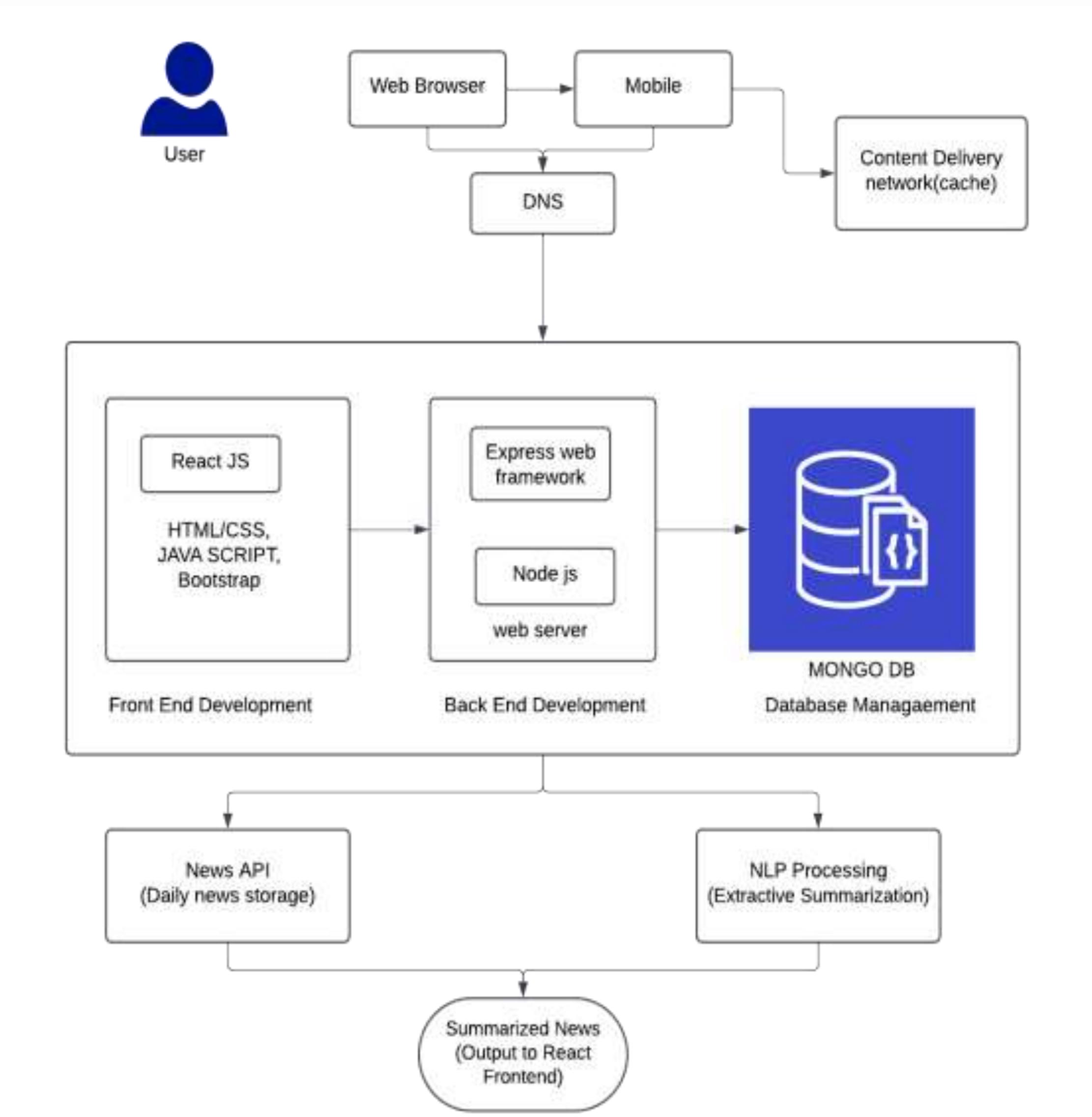

User
Web Browser
Mobile
DNS
Content Delivery network(cache)
React JS
HTML/CSS, JAVA SCRIPT, Bootstrap
Express web framework
Node js
web server
MONGO DB
Front End Development
Back End Development
Database Managaement
News API (Daily news storage)
NLP Processing (Extractive Summarization)
Summarized News (Output to React Frontend)

fig 3.1 : Architecture Diagram

The system is designed to efficiently deliver news summaries to users through a coordinated interaction among various components, each playing a vital role in the overall process. At the forefront of this system is the user, who initiates the interaction by accessing the application via a React frontend. This user interface is crucial as it not only facilitates user interactions but also serves as the visual representation of the summarized news content. The frontend is designed to be intuitive and user-friendly, allowing users to easily navigate through the application and request the latest news summaries.

Upon receiving a request from the user, the React frontend communicates with the backend, which is built using Node.js and Express.js. The backend acts as a powerful intermediary, responsible for handling the application's logic and data processing. It receives the requests from the frontend and processes them accordingly, ensuring that the right information is fetched and delivered to the user. The backend plays a critical role in managing communication between the frontend and the database, as well as coordinating the NLP processing.

The MongoDB database is another key component of the system, designed to store the news articles that are fetched from an external source, along with their processed summaries. This database not only provides a persistent storage solution but also ensures that previously fetched articles are available for future reference and retrieval. This capability is essential for maintaining an efficient workflow and enabling users to access summaries of past articles as needed.

To obtain the latest news articles, the backend makes requests to an external News API, which serves as the primary source of current news content. This API provides real-time access to a vast array of articles from various publishers, ensuring that users receive the most up-to-date information. Once the articles are fetched, they are temporarily stored in the MongoDB database, where they await processing.

The next critical step in the workflow involves the NLP processing component, which applies advanced Natural Language Processing techniques, specifically the Text Rank algorithm. This algorithm is designed to analyze the fetched articles and extract the most significant information by identifying the most important sentences. By doing so, the NLP component distills the content of the articles into concise summaries that capture the essence of the original texts. This process enhances the user experience by providing clear and relevant information without overwhelming users with excessive details.

After the NLP processing is completed, the backend takes the summarized news content and sends it back to the React frontend. This step is crucial, as it allows the processed information to be presented in a visually appealing manner that is easy for users to read and understand. The React frontend displays the summarized news to the user, effectively closing the loop of interaction.

In essence, the entire system functions as a seamless entity that efficiently fetches news articles from an external API, processes them using sophisticated NLP techniques to extract key information, and presents the summarized content to the user through an engaging interface. The coordinated efforts of the user, React frontend, Node.js and Express.js backend, MongoDB database, News API, and NLP processing component work in harmony to deliver a streamlined news summary experience. This comprehensive workflow not only improves the accessibility of news information but also enhances user satisfaction by delivering concise and relevant summaries tailored to their interests

3.2 FLOW CHART

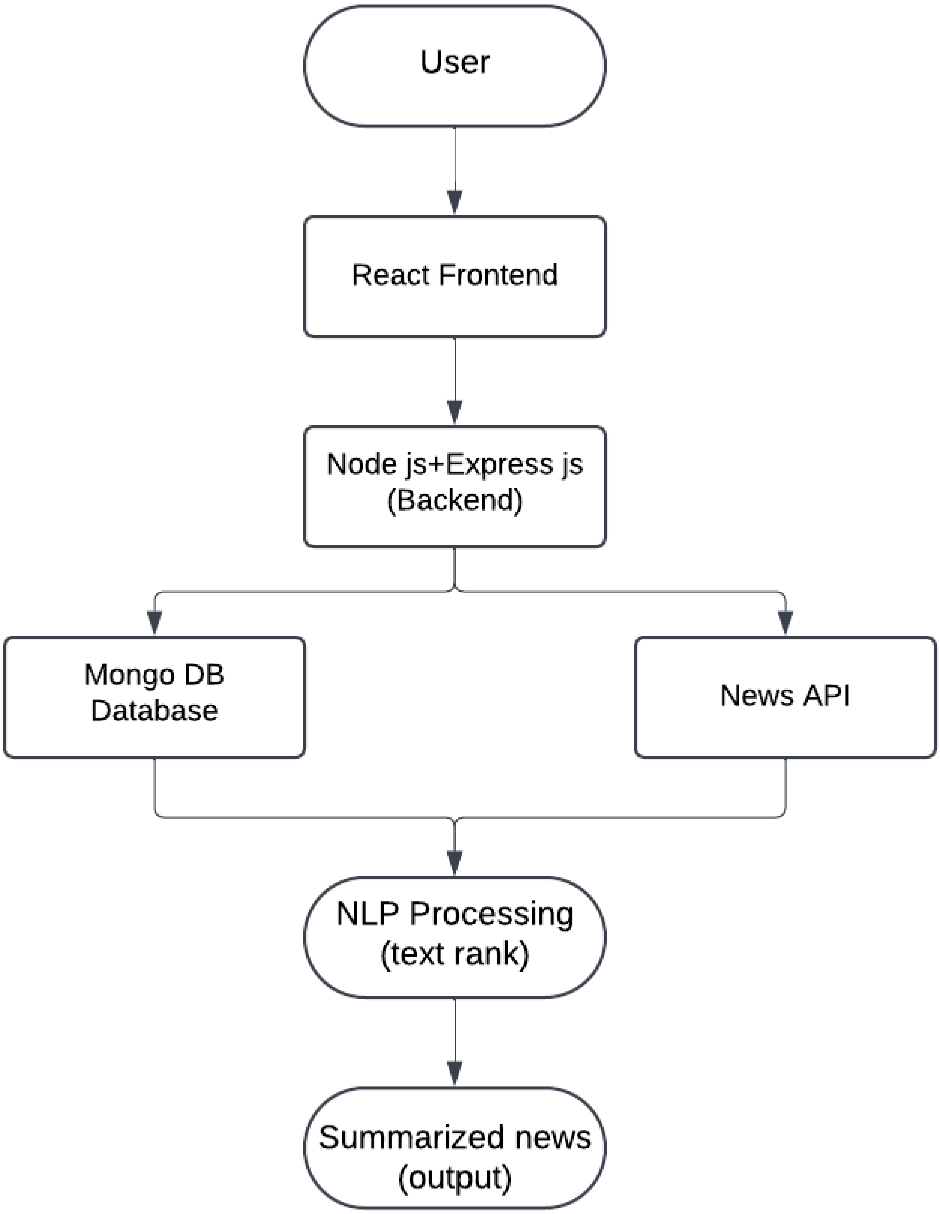


Fig 3.2 : Flow Diagram

The system is designed to efficiently deliver news summaries to users through a comprehensive flow of operations that involves multiple interconnected components. It begins with user interaction, where individuals engage with the application via a user-friendly React frontend. This interface allows users to input specific keywords or phrases for direct searches, select predefined categories such as "Technology," "Business," or "Entertainment," and even create personalized feeds based on their interests to receive tailored news recommendations. Once a user makes a request, the React frontend sends an HTTP request to the Node.js backend server, including essential parameters like the search query, selected categories, and user preferences stored in their profiles.

Upon receiving this request, the Node.js backend processes it by integrating with the News API to retrieve relevant news articles that match the user’s input. The backend filters the fetched data according to the user's preferences, such as language, region, or source. Subsequently, the retrieved articles are stored in a MongoDB database for future reference and analysis. Additionally, caching mechanisms may be implemented to store frequently accessed data, which helps reduce database load and enhances performance.

Next, the system employs advanced Natural Language Processing (NLP) techniques, specifically the Text Rank algorithm, to analyze the fetched articles. The text of each article undergoes preprocessing to eliminate stop words, punctuation, and other irrelevant elements, followed by the creation of a graph where nodes represent words or phrases, and edges depict their relationships. The Text Rank algorithm is then applied to this graph to identify the most important sentences, iteratively calculating the significance of each node based on its connections. The result is a concise summary generated from the top-ranked sentences.

The summarized news is then formatted into user-friendly formats, such as bullet points highlighting key information, concise paragraphs providing a condensed version of the article, or interactive summaries that may include clickable links to the original articles or related content. Following the generation of summaries, the backend sends an HTTP response to the frontend containing the summarized articles. This response includes article titles, the generated summaries, links to the original articles on the news source's website, and additional metadata like publication dates and authors.

Finally, the React frontend receives this response and renders the summarized news in a visually appealing and interactive manner, enhancing the user experience. Users can further engage with the summaries by clicking on links, sharing articles, or providing feedback. Through these coordinated steps and the integration of advanced NLP techniques, the system effectively delivers accurate, concise, and relevant news summaries to users, significantly enhancing their news consumption experience.

CHAPTER 4

PROJECT MODULES

5 MODULES

The project consists of Five modules. They are as follows,

1. News Collection and Aggregation
2. Summarization
3. User Interface (UI)
4. Data Management
5. Real-Time Update

4.1 NEWS COLLECTION AND AGGREGATION

The News Collection and Aggregation Module serves as the foundation of the news aggregator platform, tasked with gathering and organizing news articles from diverse, reliable sources via the News API. This module fetches articles across various categories—such as politics, technology, entertainment, and sports—ensuring a broad spectrum of news coverage. For each article, structured data is collected, including the title, source, author, publication date, and full text. This information is then processed and categorized to help users easily navigate through different news sections. The data is stored in MongoDB, which enables quick and efficient access, with a flexible schema to accommodate the diverse structures of news data from multiple sources. To maintain timely updates, the module incorporates a Node.js scheduling system, which automates data retrieval at regular intervals, ensuring users have access to the latest news without manual refreshing. By structuring data for integration with the summarization and UI modules, this component allows for seamless browsing, quick article summaries, and real-time news accessibility. With its efficient aggregation, categorization, and updating processes, the News Collection and Aggregation Module provides a consistent, up-to-date, and organized stream of news content that enhances the overall user experience on the platform.

4.2 SUMMARIZATION

The Summarization Module is a key component of the news aggregator platform, designed to deliver concise, extractive summaries of each article. This module employs the TextRank algorithm, an effective extractive summarization method that identifies and ranks key sentences within the text based on their importance. By focusing on the most relevant content, TextRank condenses lengthy articles into brief summaries that capture the main points, enabling users to quickly grasp essential information without reading through the entire text. This approach significantly improves reading efficiency and reduces time spent on news consumption. In addition to providing summaries, the Summarization Module integrates seamlessly with the data storage system, allowing both full articles and their summaries to be stored and retrieved from MongoDB. The module is designed to work alongside the User Interface (UI) module, displaying each article's summary directly in the user’s feed for easy browsing. This functionality not only enhances the user experience but also mitigates information overload, enabling users to stay informed without feeling overwhelmed by the volume of content. Through this efficient and user-focused design, the Summarization Module plays a central role in enhancing the accessibility and usability of the news aggregator platform.

4.3 USER INTERFACE (UI)

The User Interface (UI) Module is the frontend layer of the news aggregator platform, developed using React to create a responsive, interactive, and user-friendly experience. This module is designed to adapt seamlessly across devices, ensuring optimal usability on both desktop and mobile platforms. It provides users with an intuitive navigation system, allowing them to browse through various news categories, search for specific topics, and view summaries or full articles as needed. With a clean and organized layout, the UI module focuses on making news accessible and engaging, enabling users to quickly access and consume information without distractions. This module also integrates closely with other platform components, displaying real-time updates and dynamically retrieving article summaries from the Summarization Module and MongoDB database. By providing users with quick-access summaries on their newsfeed, it minimizes information overload while keeping readers informed. Additionally, React's component-based architecture ensures that the UI is both scalable and easy to maintain, allowing future feature expansions, such as personalized content recommendations, to be integrated with minimal disruption. Overall, the UI Module not only supports efficient news browsing but also enhances the overall user experience by presenting content in an accessible and visually appealing format.

4.4 DATA MANAGEMENT

The Data Management Module is responsible for handling the efficient storage, retrieval, and organization of news data within the platform, using MongoDB as the core database. It stores detailed records of each article, including titles, summaries, full text, publication dates, and source information. Designed to support fast access to both full articles and their summaries, this module enables smooth and reliable data flow between different components, such as the Summarization Module and the User Interface (UI) Module. MongoDB's schema flexibility ensures that the system can adapt to a wide range of data formats and accommodate varying types of content from multiple sources.In addition to storage, the Data Management Module is optimized for real-time operations, enabling rapid queries and updates to meet user demands for fresh news content. Node.js and Express handle the backend processes, managing database interactions to ensure high responsiveness and minimal latency. This module also maintains data consistency across the platform, supporting regular updates and integration with scheduling tasks for fetching new content. The Data Management Module thus forms the backbone of the platform's infrastructure, ensuring a seamless, scalable, and efficient experience for users as they access diverse, up-to-date news content.

4.5 REAL-TIME UPDATE

The Real-Time Update Module is an essential component of the news aggregator platform, designed to ensure that users always have access to the latest news articles. This module utilizes Node.js to implement a scheduling system that automatically fetches new content at regular intervals from the News API. By performing these automated data retrieval processes, the module keeps the platform's content fresh and relevant, eliminating the need for users to manually refresh the page to see the latest updates. The timely updating of articles enhances user engagement and encourages users to return frequently for new information. In addition to fetching updates, the Real-Time Update Module also plays a crucial role in maintaining the overall performance and responsiveness of the platform. It is responsible for efficiently managing incoming data streams, ensuring that new articles are seamlessly integrated into the existing database managed by the Data Management Module. This integration allows for the immediate availability of newly fetched articles and their corresponding summaries, providing users with an uninterrupted and dynamic news consumption experience. By prioritizing real-time access to content, this module significantly enhances the platform's usability and user satisfaction, making it an integral part of the news aggregator's functionality.

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 INTRODUCTION

This chapter involves the technology used, the hardware requirements and the software requirements for the project .

5.2 REQUIREMENTS

5.2.1 *Hardware Requirements*

Hard disk	:	500 GB and above
Ram	:	4GB and above
Processor	:	I-3 and above

5.2.2 *Software Requirements*

Operating System	:	Windows 7 and above
Java Development Kit (JDK)	:	Version 1.8
NetBeans	:	Version 8.1
Node.js	:	Latest LTS version
NPM	:	Included with Node.js
MongoDB	:	Latest stable version
React	:	Latest version (installed via NPM)
Express.js	:	Installed via NPM
TextRank Library	:	As needed for summarization
Version Control System	:	Git
Code Editor or IDE	:	Visual Studio Code or equivalent
Browser	:	Chrome

5.3 SOFTWARE DESCRIPTION

5.3.1 MongoDB

MongoDB is a NoSQL, document-oriented database designed to handle large volumes of structured, semi-structured, and unstructured data, making it ideal for applications like the news aggregator. Unlike traditional relational databases, MongoDB stores data in flexible, JSON-like documents, allowing it to adapt to the evolving data structure of real-world applications. This flexibility is particularly advantageous for a news aggregator, where data types can vary widely, from textual content and metadata to images and links. MongoDB's schema-less design enables quick and seamless modifications to the data structure, allowing the platform to evolve and scale without significant restructuring. Additionally, MongoDB supports powerful indexing and querying capabilities that facilitate rapid retrieval of news articles, summaries, and user preferences. This ensures that the platform can deliver relevant information promptly, even as the database grows. MongoDB’s ability to scale horizontally by distributing data across multiple servers further enhances its suitability, ensuring that as user demand increases, the platform remains responsive and capable of managing a high volume of data efficiently. MongoDB’s integration with other tools and frameworks in the MERN stack also makes it a natural choice, streamlining the development and deployment process.

5.3.2 Express.js

Express.js is a lightweight, fast, and highly flexible web application framework built on top of Node.js, designed to facilitate efficient server-side development. Known for its simplicity and powerful features, Express.js allows developers to manage requests and responses seamlessly, a critical requirement for the news aggregator platform. Through its routing capabilities, Express.js enables the platform to handle HTTP requests and direct them to appropriate endpoints, ensuring that users receive accurate data with minimal delay. Express also supports middleware functions, allowing the platform to implement features like user authentication, data validation, and error handling efficiently. This modular approach makes it easier to manage and organize the code, leading to more maintainable and scalable application architecture. Express's integration with the rest of the MERN stack, especially Node.js, enhances the overall development workflow, making it easier to create, test, and deploy API endpoints that connect the front-end React components with MongoDB's database layer. This synergy helps ensure smooth interactions between the client and server sides, offering a responsive and efficient user experience.

5.3.3 React

React is an open-source JavaScript library developed by Facebook, primarily used for building dynamic and interactive user interfaces for web applications. It's especially suitable for single-page applications, where quick responsiveness and minimal reloading are essential to provide a smooth user experience. In the news aggregator platform, React is used to create a rich and responsive user interface that allows users to easily navigate through articles, view summaries, and interact with real-time updates. React's component-based architecture is particularly beneficial, enabling developers to create reusable UI components such as article lists, category filters, and search bars, all of which can be efficiently managed and updated. The virtual DOM (Document Object Model) in React optimizes rendering performance by only re-rendering components that have changed, ensuring fast and seamless updates even as the user interacts with various sections of the platform. Additionally, React's state management capabilities allow for efficient handling of user preferences and personalized content display, contributing to a more tailored and engaging experience. The modularity of React also enhances maintainability, making it easier to expand the platform with new features or UI improvements over time.

5.3.4 Node.js

Node.js is a runtime environment built on Chrome's V8 JavaScript engine that enables the execution of JavaScript on the server side. As a non-blocking, event-driven framework, Node.js is particularly suited for I/O-heavy applications like the news aggregator, where handling multiple simultaneous requests efficiently is essential. With its asynchronous processing capabilities, Node.js can manage numerous connections without tying up resources, ensuring that the platform remains responsive even during peak usage. This characteristic is especially important in news aggregation, where real-time data fetching and processing are crucial for delivering up-to-date content to users. Node.js's event-driven model supports high concurrency, allowing the platform to handle tasks like fetching news articles from various APIs, summarizing content, and interacting with MongoDB databases simultaneously without delays. Node.js also includes a built-in package manager, NPM (Node Package Manager), which simplifies the installation and management of libraries and modules, facilitating the development process. Serving as the backbone of the entire platform, Node.js enables seamless integration with MongoDB for data storage, Express.js for server management, and React for front-end rendering, making it the cornerstone of the MERN stack and a powerful tool for building scalable, high-performance web applications.

CHAPTER 6
IMPLEMENTATION

6.1 OVERVIEW OF SYSTEM ARCHITECTURE

The architecture of the news aggregator platform is designed to facilitate efficient data retrieval, processing, and presentation. At its core, the system comprises several interconnected modules, each responsible for specific tasks.

The News Collection Module interacts directly with the News API to fetch articles based on user-defined parameters. This module is essential for ensuring a diverse range of content, as it retrieves news from various sources and categorizes them accordingly.

The Summarization Module utilizes the TextRank algorithm to analyze the fetched articles and produce concise summaries. This is particularly important in today's fast-paced information landscape, where users often seek quick insights rather than full-length articles.

Once the articles are summarized, they are stored in a Data Management Module, which employs MongoDB as its database solution. This allows for efficient storage, retrieval, and organization of articles, summaries, and metadata.

The User Interface Module is developed using React, providing a dynamic and responsive design that enhances user engagement. Users can easily navigate through categories, search for specific topics, and interact with the content.

Finally, the Real-Time Update Module ensures that users are presented with the latest news by regularly checking for new articles and refreshing the content displayed on the platform. Overall, this architecture promotes a seamless experience for users while maintaining robust back-end operations.

6.2 INTEGRATION OF EXTERNAL APIs

The integration of external APIs, particularly the News API, is pivotal for the functionality of the news aggregator platform. This section outlines the techniques and methods utilized for this integration.

API calls are structured as RESTful requests, allowing the application to fetch data asynchronously. This is achieved through JavaScript’s async/await functionality, which ensures that the application remains responsive while waiting for API responses. The system is designed to handle various endpoints of the News API, accommodating different categories and search queries as specified by the user.

Error handling mechanisms are implemented to manage potential issues such as network errors, API rate limits, and unexpected data formats. This includes providing users with informative messages when issues arise, thereby enhancing user experience.

To optimize performance, caching strategies are adopted. Frequently accessed articles may be stored temporarily to allow for quicker retrieval and reduced latency during peak usage. Additionally, the integration strategy ensures compliance with the API’s usage policies, preventing service disruptions.

6.3 SYSTEM WORKFLOW AND DATA MANAGEMENT

The workflow of the news aggregator platform is designed to ensure a seamless flow of data from collection to presentation. The system begins by initiating API calls through the News Collection Module, fetching articles based on user preferences.

Once the articles are retrieved, they pass to the Summarization Module, where the TextRank algorithm analyzes the content to extract key sentences, creating concise summaries. These summaries, along with their source articles, are then stored in the MongoDB database.

The Data Management Module plays a crucial role in organizing this data, allowing for efficient querying and retrieval. This module ensures that articles can be categorized by topics, sources, and publication dates, facilitating easy access for users.

The final stage of the workflow involves presenting this data through the User Interface Module, where users can interact with the content. The Real-Time Update Module regularly refreshes the data, ensuring that users have access to the most current news without needing to manually reload the page. This structured workflow enhances user engagement and satisfaction.

6.4 USER INTERACTION AND EXPERIENCE DESIGN

The user interface and experience design of the news aggregator platform are critical to its success. Developed with React, the interface emphasizes clarity and simplicity, allowing users to navigate effortlessly through various sections.

The design incorporates a clean layout, with articles presented in an organized manner, including headlines, summaries, and publication dates. Users can filter articles by categories such as technology, health, and sports, providing a tailored experience that meets their interests.

features allow users to save articles for later reading and share content on social media platforms directly from the interface. The design also incorporates user feedback mechanisms, enabling users to rate summaries and report any issues, contributing to continuous improvement based on real-world usage.

Accessibility considerations are integrated into the design, ensuring that users with disabilities can navigate the platform effectively. Overall, this focus on user experience ensures that the news aggregator not only delivers relevant content but also engages users in a meaningful way.

6.5 QUALITY ASSURANCE AND PERFORMANCE EVALUATION

Quality assurance is a critical aspect of the implementation process, ensuring that the news aggregator platform functions correctly and efficiently. A multi-tiered testing strategy is employed, encompassing various methodologies to validate both functionality and performance.

Unit testing is conducted to verify the operation of individual modules, ensuring that each component performs its intended task. Integration testing assesses how well these modules work together, identifying any issues that may arise from interactions between components.

User acceptance testing involves real users interacting with the platform, providing feedback on usability and functionality. This feedback is crucial for making necessary adjustments and enhancements to the platform before its official release.

Performance evaluation focuses on the system's responsiveness under different loads, using stress testing to determine how well the platform handles high traffic. This includes monitoring API response times and ensuring that the database can efficiently manage multiple concurrent requests.

By implementing these quality assurance measures, the project team aims to deliver a reliable, user-friendly news aggregator platform that meets high standards of performance and usability.

CHAPTER 7
CONCLUDING REMARKS

7.1 CONCLUSION

In conclusion, the news aggregator platform successfully addresses the growing need for efficient and effective news consumption in an era where information overload is prevalent. By leveraging modern technologies such as the MERN stack and implementing the TextRank algorithm for extractive summarization, the platform not only streamlines the collection and aggregation of news articles from diverse sources but also enhances user experience through an intuitive interface and real-time updates. The integration of various modules, including data management, user interaction, and API techniques, ensures a robust and scalable system capable of adapting to users' evolving needs.

The effectiveness of the summarization feature allows users to quickly grasp the essence of news articles, making the platform a valuable tool for those seeking to stay informed without dedicating extensive time to reading lengthy articles. Furthermore, the comprehensive project workflow ensures that every aspect of the application, from data collection to user engagement, operates seamlessly and efficiently. As the platform continues to evolve, there are ample opportunities for further enhancements, including the incorporation of machine learning techniques for personalized news recommendations and improved user engagement metrics. Overall, this project not only demonstrates the potential of technology in addressing contemporary challenges in information dissemination but also lays the groundwork for future innovations in the realm of news aggregation.

REFERENCES

[1] Mohamed, A., Ibrahim, M., Yasser, M., Ayman, M., Gamil, M., & Hassan, W. (2020). News Aggregator and Summarization System. methods, 11(6).

[2] Liyaqat Fayaz., , Iqbal Bashir., Mrs. Sahila (2021). News Aggregator: The World at Your Finger Tips

[3] Mr Rakesh Kumar Rai,Dr Isha Singh,Ankit Mudia, Karandeep Bisht (2021). News aggregator web application using Django

[4] Gokula Krishnan, Prof. N. Sakthivel (2023). ARTIFICIAL INTELLIGENCE-BASED NEWS AGGREGATOR

[5] Visvam Devadoss, A. K., Thirulokachander, V. R., & Visvam Devadoss, A. K. (2019). Efficient daily news platform generation using natural language processing. International journal of information technology, 11, 295-311.

[6] Adhikari, S. (2020, March). Nlp based machine learning approaches for text summarization. In 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC) (pp. 535-538). IEEE.

[7] Wang, J., Maguin, J., Orloff, G., & Groves, J. (2024). Daily Digest: A News Aggregation Site.

[8] Shukla, S. K., Dubey, S., Rastogi, T., & Srivastava, N. (2022). Application using MERN stack. International journal for modern trends in science and technology, 8(06), 102-105.

[9] Bawane, M., Gawande, I., Joshi, V., Nikam, R., & Bachwani, S. A. (2022). A Review on Technologies used in MERN stack. Int. J. Res. Appl. Sci. Eng. Technol, 10(1), 479-488.

[10] Awasthi, I., Gupta, K., Bhogal, P. S., Anand, S. S., & Soni, P. K. (2021, January). Natural language processing (NLP) based text summarization-a survey. In 2021 6th International Conference on Inventive Computation Technologies (ICICT) (pp. 1310-1317). IEEE.

[11] Boorugu, R., & Ramesh, G. (2020, July). A survey on NLP based text summarization for summarizing product reviews. In 2020 Second International

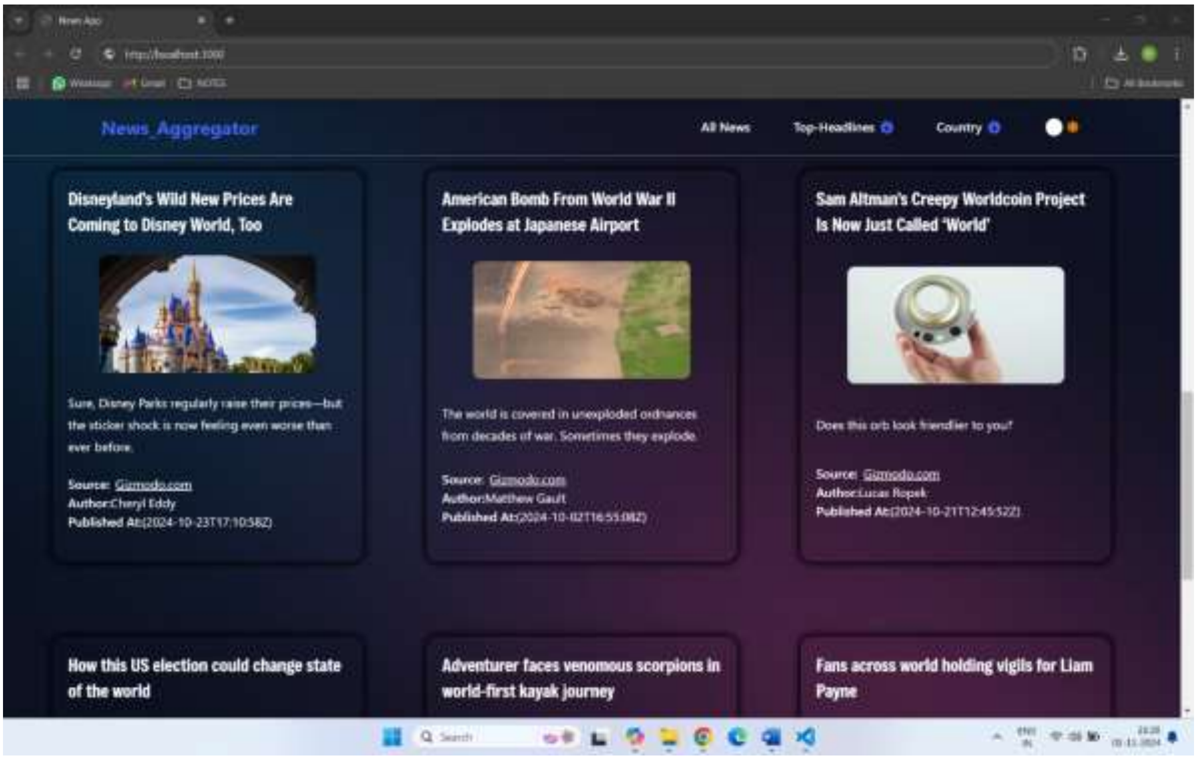
Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 352-356). IEEE.

[12] Batra, P., Chaudhary, S., Bhatt, K., Varshney, S., & Verma, S. (2020, August). A review: Abstractive text summarization techniques using NLP. In 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM) (pp. 23-28). IEEE.

[13] Jugran, S., Kumar, A., Tyagi, B. S., & Anand, V. (2021, March). Extractive automatic text summarization using SpaCy in Python & NLP. In 2021 International conference on advance computing and innovative technologies in engineering (ICACITE) (pp. 582-585). IEEE

APPENDICES

SAMPLE SCREENSHOT



Disneyland's Wild New Prices Are Coming to Disney World, Too

Sure, Disney Parks regularly raise their prices—but the sticker shock is now feeling even worse than ever before.

Source: Gizmodo.com
Author: Cheryl Eddy
Published At:(2024-10-23T17:10:58Z)

American Bomb From World War II Explodes at Japanese Airport

The world is covered in unexploded ordnances from decades of war. Sometimes they explode.

Source: Gizmodo.com
Author: Matthew Gault
Published At:(2024-10-02T16:55:08Z)

Sam Altman's Creepy Worldcoin Project Is Now Just Called 'World'

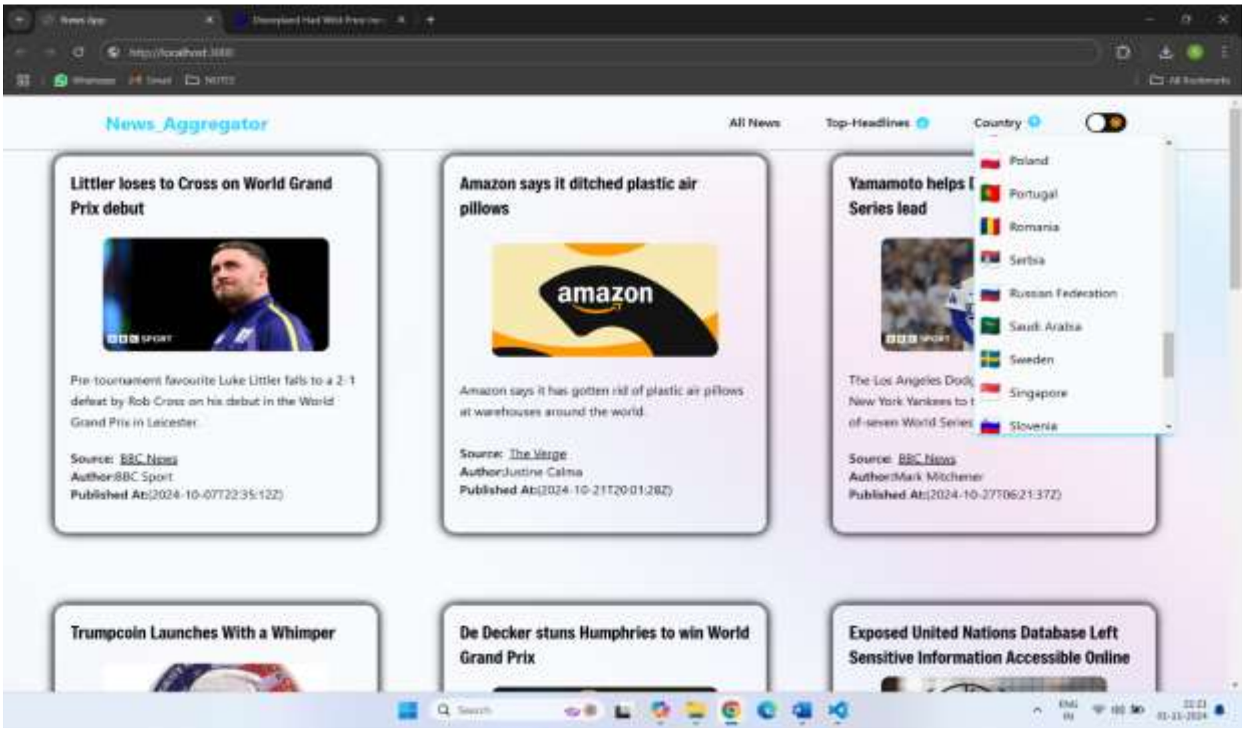
Does this orb look friendlier to you?

Source: Gizmodo.com
Author: Lucas Ropek
Published At:(2024-10-21T12:45:52Z)

How this US election could change state of the world

Adventurer faces venomous scorpions in world-first kayak journey

Fans across world holding vigils for Liam Payne



Littler loses to Cross on World Grand Prix debut

Pre-tournament favourite Luke Littler falls to a 2-1 defeat by Rob Cross on his debut in the World Grand Prix in Leicester.

Source: BBC News
Author:BBC Sport
Published At:(2024-10-07T22:35:12Z)

Amazon says it ditched plastic air pillows

Amazon says it has gotten rid of plastic air pillows at warehouses around the world.

Source: The Verge
Author: Justine Calma
Published At:(2024-10-21T20:01:28Z)

Yamamoto helps [illegible] Series lead

The Los Angeles Dod[illegible] New York Yankees to [illegible] of-seven World Series [illegible]

Source: BBC News
Author: Mark Mitchener
Published At:(2024-10-27T06:21:37Z)

Trumpcoin Launches With a Whimper

De Decker stuns Humphries to win World Grand Prix

Exposed United Nations Database Left Sensitive Information Accessible Online