INSIGHTSTREAM: NAVIGATE THE NEWS LANDSACPE

NAAN MUDHALVAN PROJECT REPORT

Bachelor of Computer Science

Submitted by

TEAM LEADER

G.KRISHNARAJ (222209503) **Krishnaraj24092005@gmail.com**

TEAM MEMBERS

B.BASKAR (222209494) **baskar18052005@gmail.com**

M.GOPALAKRISHNAN (222209497) **balabala13052005@gmail.com**

S. NITHISH KUMAR (222209507) **crazynithish2004@gmail .com**

DEPARTMENT OF COMPUTER SCIENCE

TAGORE COLLEGE OF ARTS AND SCIENCE

(Affiliated to the University of Madras)

MARCH – 2025

TABLE OF THE CONTENT

ABSTRACT

InsightStream:

In today's fast-paced digital world, staying informed can be overwhelming with an endless stream of news, misinformation, and biased reporting. InsightStream is designed to cut through the noise—an AI-driven, React-based news aggregator that delivers a **personalized, efficient, and reliable** news consumption experience. By leveraging advanced AI functionalities, InsightStream offers **tailored content recommendations, automated news summarization, sentiment analysis, and category-based filtering**, ensuring that users receive relevant and meaningful news updates.

Beyond intelligent curation, the platform provides a **seamless user experience**, allowing users to **search, bookmark, and receive real-time updates** effortlessly. With **multi-language support, an intuitive UI, and secure authentication**, InsightStream ensures accessibility and inclusivity for a diverse audience.

Built with a **robust tech stack**, InsightStream combines **React.js for the frontend, Node.js with Express.js for the backend, and MongoDB for efficient data management**. It integrates third-party services such as **NewsAPI for real-time news aggregation and OpenAI for AI-powered enhancements**, making it a cutting-edge solution in digital news consumption.

Looking ahead, InsightStream is set to introduce **social media integrations** for seamless content sharing, **advanced fake news detection** to enhance credibility, and **voice-based news summarization** for hands-free accessibility. By merging **state-of-the-art AI with an intuitive user interface**, InsightStream transforms the way users engage with news, making it smarter, faster, and more reliable.

SYNOPSIS

CHAPTER 1 SYNOPSIS

1. Introduction

The exponential rise in digital content has made it increasingly difficult for users to stay informed without feeling overwhelmed. Traditional news platforms often lack personalization, making it challenging for users to find content that aligns with their interests. **InsightStream** is a React-based news aggregator that leverages AI to provide personalized and real-time news updates, ensuring users have effortless access to relevant and credible information.

2. Objectives

The primary objectives of InsightStream are to develop a **dynamic and intuitive** news aggregator platform that **personalizes content recommendations using AI**, provides **real-time news updates from trusted sources**, and enhances accessibility with features like **multi-language support and dark mode**. Additionally, robust authentication measures will be implemented to ensure user data security.

3. Features

InsightStream offers several key features, including a **Personalized News Feed** that curates articles based on user preferences, **Category-Based Filtering** that allows users to explore news by topics such as technology, politics, and sports, and **Real-Time Updates** to ensure the latest news is always available. Users can also utilize **Search & Bookmarking** to easily find and save articles, and the platform extends accessibility through **Multi-Language Support**. AI-powered functionalities such as **Automated News Summarization** provide quick and concise news briefs, while **User Authentication & Security** features, including Firebase/Auth0, ensure a secure login experience. Additionally, **Trending Topics & Notifications** help users stay updated on global and regional news trends.

4. Technology Stack

InsightStream is built using a modern tech stack for efficient performance. The **frontend** is developed with **React.js, Redux, and Tailwind CSS**, while the **backend** is powered by **Node.js with Express.js**. The platform uses **MongoDB** for database management and integrates **NewsAPI, Google News API, and OpenAI API** for real-time news fetching and AI-driven enhancements. Authentication is secured using **Firebase/Auth0**.

5. Development Methodology

The development process follows a structured methodology, beginning with **Requirement Analysis** to understand user needs, followed by **UI/UX Design**, which involves wireframing and prototyping. The **Implementation** phase includes developing frontend and backend components, while **Testing** ensures quality through unit, integration, and usability tests. **Deployment** will be carried out using cloud services such as **AWS, Vercel, or Netlify**, and ongoing **Maintenance** will include continuous updates and feature enhancements.

6. AI Integration

AI plays a crucial role in InsightStream by enhancing the user experience. **News Summarization** extracts key points from articles, making them easier to consume. **Personalized Content Recommendations** adapt based on user behavior, ensuring that users receive the most relevant news. Additionally, **Sentiment Analysis** evaluates articles to determine their tone, categorizing them as **positive, negative, or neutral**.

7. Security & Privacy

To ensure user safety and data protection, InsightStream implements **Role-Based Access Control (RBAC)**, allowing different access levels for users. **Data Encryption** is used for secure storage of user preferences and bookmarks, while **Privacy Compliance** ensures adherence to **GDPR and other industry standards**.

8. Target Audience & Use Cases

InsightStream caters to a wide range of users, including **News Enthusiasts** who seek a tailored news experience, **Professionals** looking for quick industry-specific updates, and **Students & Researchers** who require access to reliable sources for their studies.

9. Conclusion & Future Enhancements

InsightStream aims to **redefine digital news consumption** by integrating AI, real-time updates, and personalization into a seamless user experience. Future developments will focus on **expanding social media sharing options**, implementing **advanced fake news detection**, and introducing **voice-based news summarization** to enhance accessibility. With these innovations, InsightStream will continue to be a go-to platform for informed and efficient news consumption.

1.1 Scope of the Project

Designed as a **web-based application**, InsightStream will serve a diverse audience, including **students, professionals, journalists, and general readers**. The project scope includes **Frontend Development** using **React.js**, **Backend Development** with **Node.js and Express.js**, and **AI Integration** for functionalities such as **summarization, recommendations, and sentiment analysis**. The platform will fetch news from trusted sources like **Google News API and NewsAPI** while ensuring **robust security measures** for authentication and data protection. Future enhancements will focus on **scalability**, **social media integration**, and **fake news detection**.

1.2 Background & Problem Statement

As traditional news consumption methods, such as newspapers and television broadcasts, are rapidly being replaced by online platforms, several challenges arise. These include **Information Overload**, where users face an overwhelming number of articles from multiple sources, and the **Spread of Misinformation & Fake News**, which makes it difficult to distinguish between credible and misleading content. Additionally, many platforms suffer from a **Lack of Personalization**, failing to tailor content based on user preferences. Another common issue is **Time Constraints**, where users struggle to read lengthy articles amid their busy schedules. InsightStream addresses these challenges by **implementing AI-driven content curation**, filtering unreliable news sources, and **summarizing articles** for quick consumption.

1.3 Objectives of the Project

The key objectives of InsightStream are to **develop a user-friendly and intuitive news aggregation platform**, implement **AI-based personalized recommendations**, and provide **real-time updates** to ensure users receive the latest news. The project also aims to offer **AI-powered summarization** for quick content consumption, **multi-language**

support to serve a global audience, and **enhanced security and authentication** to protect user data and preferences.

1.4 Significance of the Project

InsightStream brings significant benefits to digital news consumption. By reducing **Information Overload**, AI-curated news ensures that users receive only the most relevant content. The platform also **enhances accessibility** through features like **multi-language support** and **dark mode**, ensuring a more inclusive user experience. Additionally, InsightStream helps **combat misinformation** by filtering out unreliable sources and **saving users time** through AI-powered summaries, allowing them to grasp essential news quickly.

SYSTEM SPECIFICATION

Chapter 2: System Specification

The system specification for InsightStream outlines the essential **hardware, software, security, and network requirements** to ensure smooth development and deployment. As the platform scales, future enhancements may necessitate additional resources to support increasing user traffic and advanced AI capabilities.

2.1 Hardware Requirements

For development, the recommended hardware specifications include a **processor** of at least **Intel Core i5 (8th Gen or above) or AMD Ryzen 5**, a **minimum of 8GB RAM (16GB recommended for better performance)**, and an **SSD with at least 256GB storage (preferably 512GB)** to ensure faster data retrieval. A **dedicated GPU** is beneficial for faster rendering, particularly when handling AI-related tasks, and a **stable internet connection** is necessary for API calls and cloud-based operations. The development environment should include a **monitor with a minimum 1080p resolution** to support high-quality UI/UX design and testing.

For deployment, high-performance server infrastructure is required to manage user requests efficiently. The server should be equipped with a **high-end processor like Intel Xeon or AMD EPYC (or a cloud equivalent), a minimum of 16GB RAM (32GB recommended for high traffic loads), and an NVMe SSD with at least 500GB of storage** to facilitate high-speed read/write operations. A separate **database storage of at least 100GB is allocated for MongoDB** to ensure scalability and optimized data management. The **network bandwidth should be at least 100Mbps** to support seamless real-time updates, and cloud hosting solutions such as **AWS, Google Cloud, or Microsoft Azure** will be used to guarantee reliability, security, and flexible scaling as needed.

2.2 Software Requirements

The software stack of InsightStream is designed for efficiency, security, and seamless integration of AI-driven functionalities. The **frontend is developed using React.js (latest stable version), with Redux or Context API for state management and Tailwind CSS or Material UI for modern and responsive UI styling**. The application is compatible with multiple browsers, including **Chrome, Firefox, Edge, and Safari**, ensuring a consistent user experience.

The **backend is built using Node.js with Express.js** to facilitate a robust and scalable server environment. The **MongoDB NoSQL database** is utilized for storing and managing structured and unstructured data efficiently. To ensure real-time news updates,

NewsAPI and Google News API are integrated for fetching articles from reliable sources, while **OpenAI API** is employed for AI-driven features such as **automated news summarization, personalized recommendations, and sentiment analysis**. **Firebase Authentication or Auth0** is implemented to provide **secure login and user authentication**, and **JWT (JSON Web Token) authentication** is used to enhance data security and prevent unauthorized access.

To streamline the development workflow, InsightStream utilizes **Visual Studio Code or JetBrains WebStorm** as the preferred **code editor**, along with **GitHub, GitLab, or Bitbucket** for **version control and collaborative development**. API testing is performed using **Postman or Thunder Client**, and **Docker** is incorporated for **containerization and scalable microservices architecture**. The **CI/CD pipeline is managed through GitHub Actions, Jenkins, Netlify, or Vercel**, enabling **automated builds, testing, and deployment** to ensure rapid and error-free updates to the live platform.

2.3 Network Requirements

To ensure optimal performance and a secure connection between the client and the server, the **development environment requires a minimum internet speed of 10Mbps**, while the **production server should have at least 100Mbps bandwidth** to handle high traffic loads and API requests efficiently. Cloud hosting must maintain **secure API endpoints and strict database access controls** to prevent cyber threats and unauthorized data breaches. Additionally, an **SSL certificate is mandatory to enable HTTPS encryption**, ensuring secure data transmission and protecting users' sensitive information from potential cyber-attacks.

Overall, these system specifications define the **foundational infrastructure necessary for InsightStream** to operate efficiently while maintaining security, scalability, and a seamless user experience. As the platform evolves, additional optimizations in **hardware resources, software frameworks, and network configurations** will be implemented to accommodate future feature expansions, including **advanced AI models, social media integrations, and real-time multimedia content delivery**.

SYSTEM ANALYSIS

CHAPTER 3 SYSTEM ANALYSIS

System analysis is a crucial phase in the development of InsightStream, where we examine the functional and non-functional requirements, system feasibility, and architectural design. This ensures that the project meets its objectives effectively and efficiently while providing a seamless user experience.

3.1 Existing System

Traditional news aggregation platforms face the following issues:

- **Lack of Personalization:** Users receive generalized news instead of tailored content.
- **Information Overload:** Large amounts of news articles without proper filtering.
- **Misinformation & Fake News:** Difficulty in distinguishing reliable sources.
- **Inefficient User Experience:** Slow navigation, unstructured content, and lack of AI-driven features.

3.2 Proposed System

- **AI-Powered Personalization:** Tailored content recommendations based on user preferences.
- **Real-Time Updates:** Fetching the latest news from trusted sources.
- **Fake News Filtering:** AI-driven analysis to detect misinformation.
- **Summarization & Sentiment Analysis:** Quick insights into lengthy articles.
  Multi-Language Support: News accessible to a wider audience.

3.3 Feasibility Study

A feasibility study evaluates InsightStream's viability in terms of technical, operational, and financial aspects.

3.3.1 Technical Feasibility

- Utilizes modern technologies like **React.js, Node.js, MongoDB, and AI APIs**.
- Scalable architecture for high-performance news aggregation. Secure authentication using Firebase/Auth0.

3.3.2 Operational Feasibility

- User-friendly UI with intuitive navigation and customization.
- AI-enhanced recommendations to optimize news consumption. Efficient content filtering and news summarization.

3.3.3 Economic Feasibility

- **Cost-effective development** using open-source technologies.
- **Minimal operational costs** with cloud-based deployment.

- **Potential revenue models:** Advertisements, premium subscriptions, and AIpowered insights.

SYSTEM DESIGN

CHAPTER 4 SYSTEM DESIGN

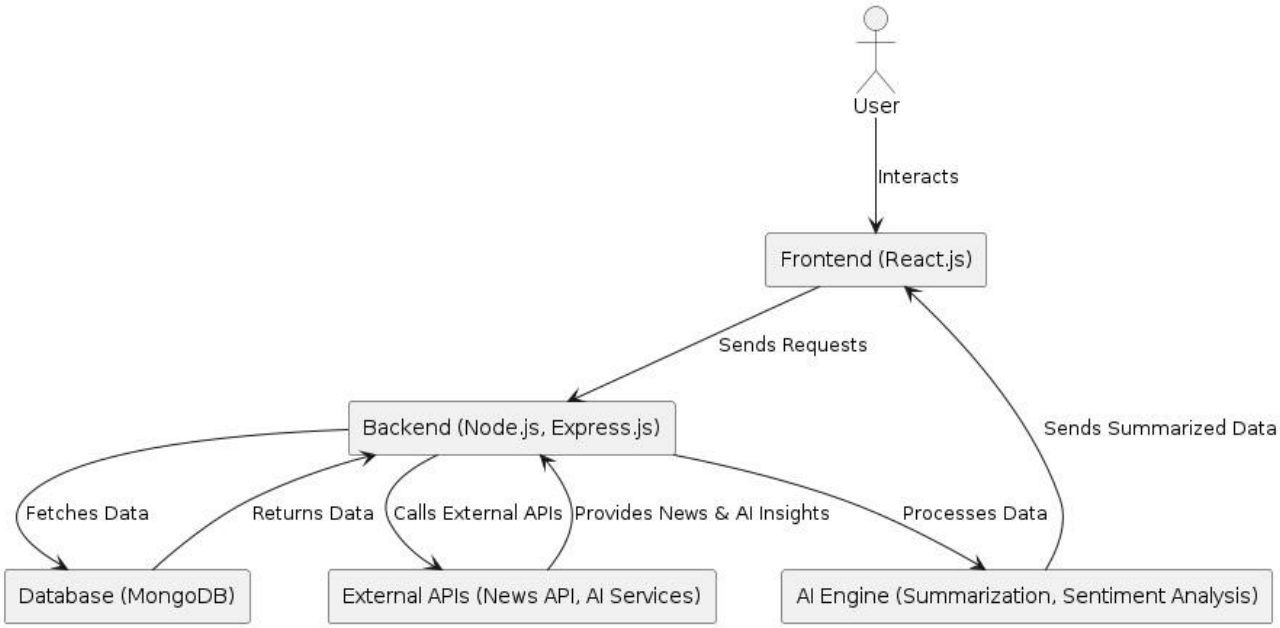
System design is a critical phase in the development of InsightStream, focusing on defining the system architecture, data flow, component interactions, and security aspects. This ensures that the platform is scalable, efficient, and user-friendly.

4.1 Architectural Design

InsightStream follows a **three-tier architecture**, which includes:

1. **Presentation Layer (Frontend)** – User interface built with React.js.
2. **Application Layer (Backend)** – Handles business logic using Node.js and Express.js.
3. **Data Layer (Database & External APIs)** – Manages data using MongoDB and fetches news from APIs like NewsAPI or Google News API.

4.1.1 System Architecture Diagram

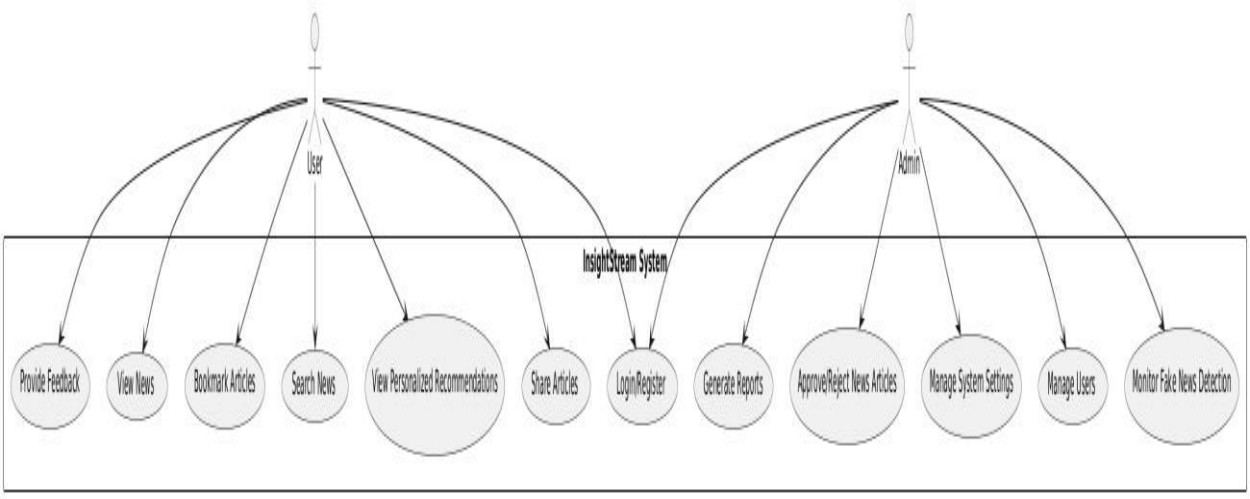


4.2 UML DIAGRAM

A **Use Case Diagram** represents the interaction between users and the system, defining user roles and their accessible functionalities. A **Class Diagram** shows the system structure with its classes, attributes, methods, and relationships, helping in object-oriented design. A **Sequence Diagram** illustrates the step-by-step flow of interactions between system components, detailing message exchanges over time. An **Activity Diagram** visualizes the system workflow, representing activities, decisions, and parallel processes to model business logic effectively.

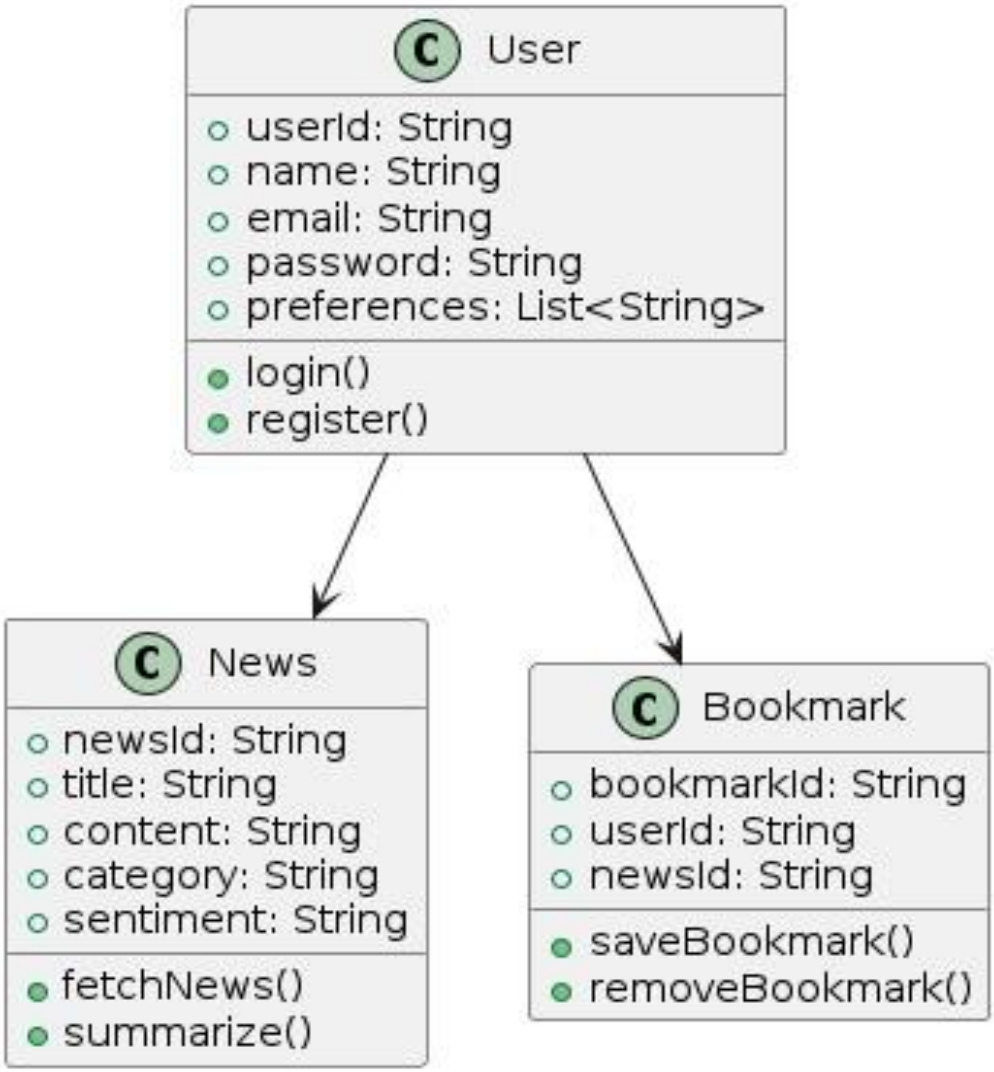
4.2.1. Use Case Diagram

A **Use Case Diagram** visually represents the interactions between users (actors) and the system. It highlights various functionalities offered by the system and how users interact with them. This diagram helps in understanding system requirements and user roles.



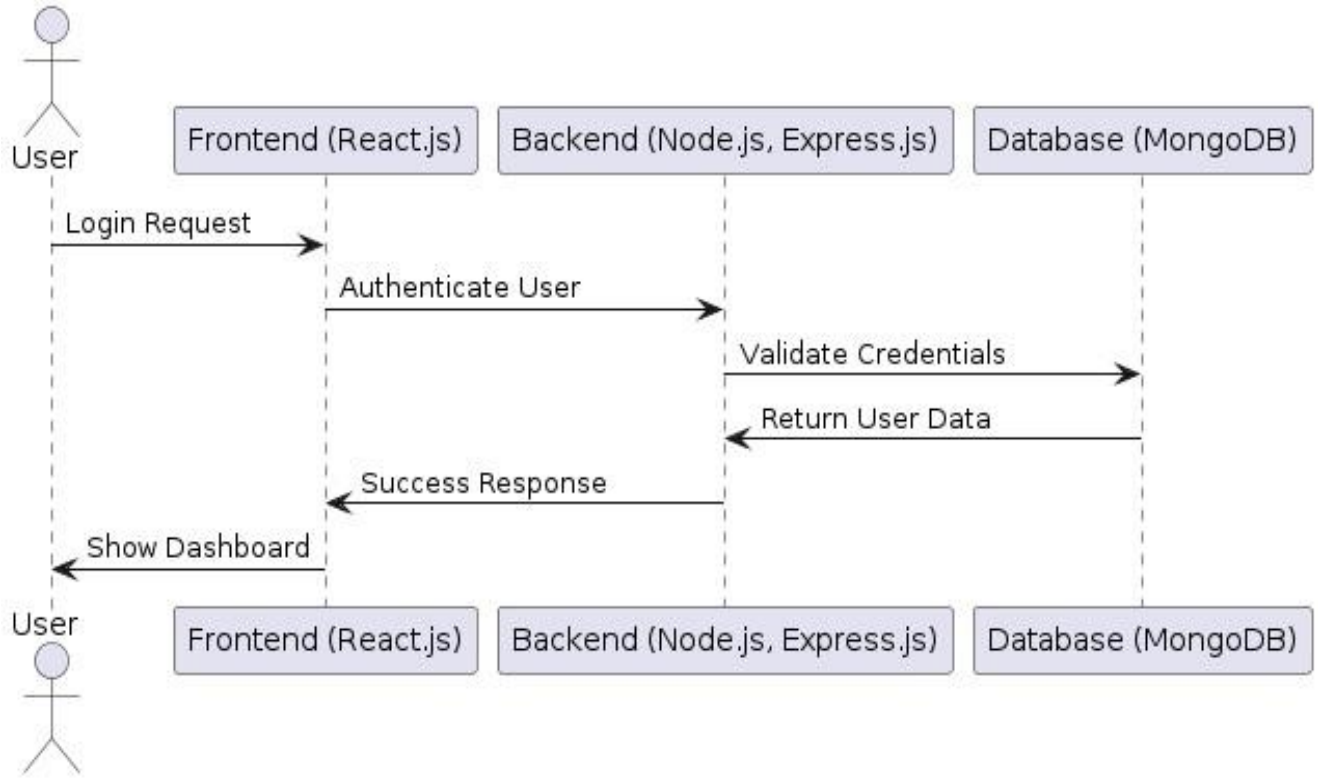
4.2.2. Class Diagram

A **Class Diagram** illustrates the structure of the system by showing its classes, attributes, methods, and relationships. It provides a blueprint for object-oriented design and helps developers understand data organization.



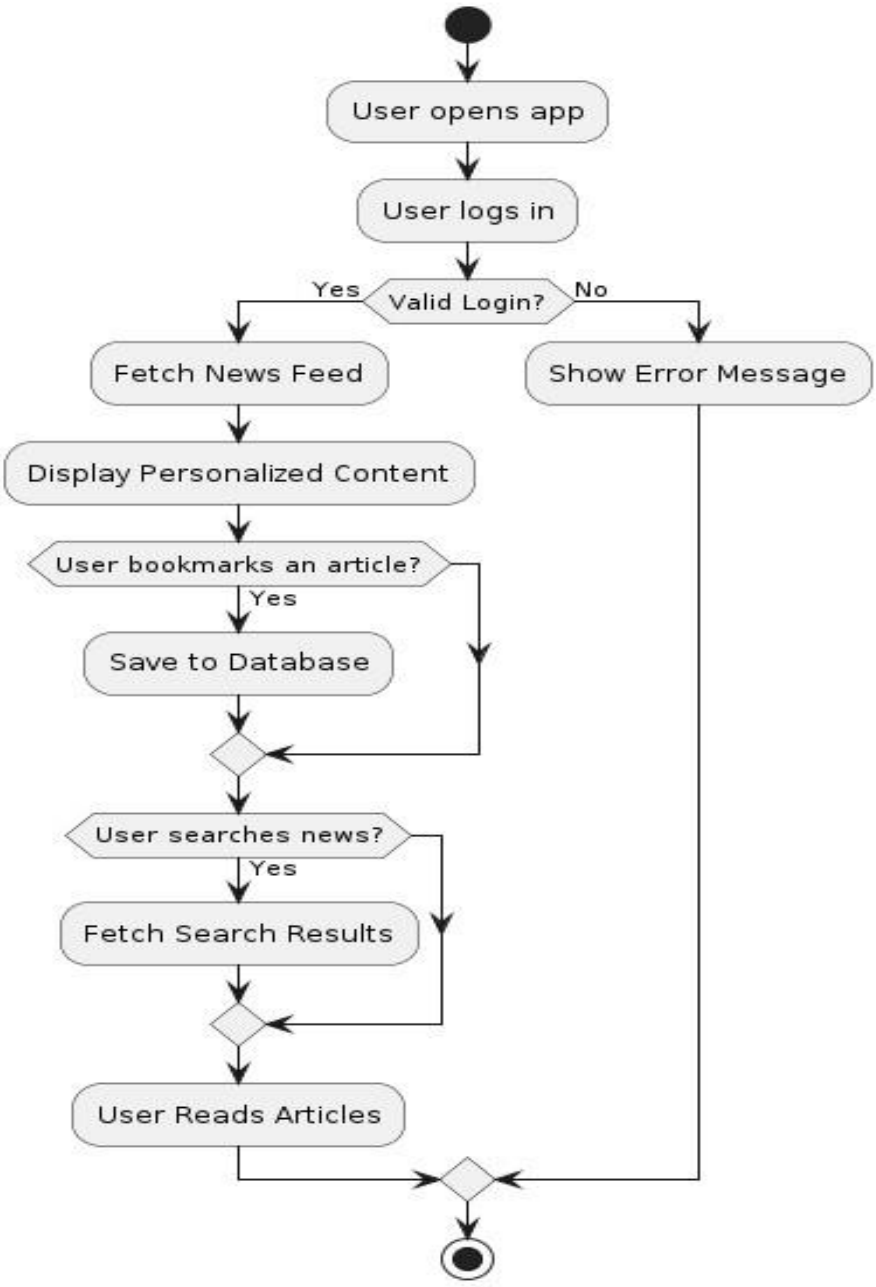
4.3.3. Sequence Diagram

A **Sequence Diagram** represents the flow of interactions between different components of the system in a step-by-step manner. It describes how processes operate in a sequential order and how messages are exchanged between system elements.



4.3.4. Activity Diagram

An **Activity Diagram** depicts the workflow of a system, showing different activities, decisions, and parallel processes. It is useful for modeling business logic and dynamic behaviors in the system.


User opens app
User logs in
Valid Login?
Yes
No
Fetch News Feed
Show Error Message
Display Personalized Content
User bookmarks an article?
Yes
Save to Database
User searches news?
Yes
Fetch Search Results
User Reads Articles

SYSTEM IMPLEMENTATION

CHAPTER 5 SYSTEM IMPLEMENTATION

System implementation refers to the process of integrating and deploying the developed system into a real-world environment. This phase ensures that the system functions as intended and meets user requirements.

1. Implementation Strategy

The system is implemented using a phased approach to minimize risks and ensure smooth deployment. The key strategies include:

- **Pilot Implementation:** Deploying the system to a small group of users for initial testing.
- **Parallel Implementation:** Running the new system alongside the existing system to compare performance.
- **Direct Implementation:** Replacing the old system with the new one immediately.
- **Phased Implementation:** Gradually introducing system components in different stages.

2. Technology Stack

- **Frontend:** React.js for a responsive and dynamic user interface.
- **Backend:** Node.js with Express.js for handling server-side logic.
- **Database:** MongoDB for efficient data storage and retrieval.
- **Authentication:** JWT-based user authentication for secure access.
- **Hosting & Deployment:** Deployed on cloud platforms like AWS or Firebase.

3. Integration and Testing

- **Unit Testing:** Testing individual components to ensure correctness.
- **Integration Testing:** Checking communication between different system modules.
- **User Acceptance Testing (UAT):** Ensuring the system meets user requirements.
- **Performance Testing:** Measuring system speed, scalability, and responsiveness.

4. Security Considerations

- **Data Encryption:** Using SSL/TLS for secure data transmission.
- **User Authentication:** Implementing multi-factor authentication (MFA) for enhanced security.
- **Access Control:** Role-based access control (RBAC) to restrict unauthorized actions.

5. Deployment Process

- **Code Review & Approval:** Ensuring code quality before deployment.
- **CI/CD Pipeline:** Automating deployment using GitHub Actions or Jenkins.

- **Server Setup:** Configuring production servers for hosting.
- **Monitoring & Maintenance:** Using logging tools like LogRocket or Datadog for continuous monitoring.

SYSTEM TESTING

CHAPTER 6 SYSTEM TESTING

System testing is a critical phase in software development that evaluates the entire system's functionality, performance, security, and usability. It ensures that the system meets the specified requirements before deployment.

1. Objectives of System Testing

- To verify that the system functions as expected.
- To identify and fix defects before production.
- To ensure system stability under various conditions.
- To validate system security and data integrity.

Types of System Testing

6.1 Functional Testing

Ensures that all features work as intended by executing test cases based on requirements.

- **Unit Testing:** Tests individual components for correctness.
- **Integration Testing:** Verifies interactions between integrated modules.
- **User Acceptance Testing (UAT):** Ensures the system meets user expectations.

6.2 Non-Functional Testing

Focuses on system performance, security, and usability.

- **Performance Testing:** Measures system speed, responsiveness, and scalability.
- **Load Testing:** Evaluates system behavior under expected and peak loads.
- **Security Testing:** Identifies vulnerabilities to prevent unauthorized access.
- **Usability Testing:** Assesses user-friendliness and overall experience.

1. **Testing Tools and Techniques**

- **Automation Testing:** Using Selenium, Junit, or Jest for automated test execution.
- **Manual Testing:** Human testers execute test cases for usability validation.
- **Penetration Testing:** Ethical hacking methods to find security loopholes. **Regression Testing:** Ensuring new updates do not introduce new bugs.

2. **Bug Tracking and Reporting**

Bugs and defects are tracked using tools like Jira, Bugzilla, or Trello. Each bug is classified based on severity (Critical, Major, Minor) and resolved accordingly.

DEVELOPMENT PROCESS

CHAPTER 7 DEVELOPMENT PROCESS

The development process outlines the structured approach used to design, build, and implement software solutions. It ensures efficiency, quality, and maintainability throughout the software lifecycle.

7.1 Software Development Life Cycle (SDLC) Phases

1. Requirement Analysis

- Understanding user needs and system requirements.
- Documenting functional and non-functional requirements.

2. Planning

- Defining project scope, timeline, and resources.
- Creating a roadmap for development.

3. Design

- Architectural and UI/UX design of the system.
- Creating wireframes, database schema, and system models.

4. Development

- Writing code based on design specifications.
- Implementing features, backend logic, and database integration.

5. Testing

- Conducting unit, integration, and system testing.
- Fixing bugs and ensuring system stability.

6. Deployment

- Releasing the application to a production environment.
- Configuring hosting and database servers.

7. Maintenance and Updates

- Monitoring system performance and security.
- Rolling out new features and fixes as needed.

8. Development Methodologies

- **Waterfall Model:** Sequential and structured development approach.
- **Agile Methodology:** Iterative and flexible approach for continuous improvements.
- **DevOps:** Integration of development and operations for faster releases.

7.2 Tools and Technologies

- **Version Control:** Git, GitHub, GitLab
- **Development Frameworks:** React.js, Node.js, Express.js
- **Database:** MongoDB, MySQL
- **Testing Tools:** Jest, Selenium
- **CI/CD:** Jenkins, GitHub Actions

FEATURES OVERVIEW

CHAPTER 8 FEATURES OVERVIEW

The features of the system define its functionality, usability, and efficiency. These features are designed to enhance user experience and provide seamless navigation within the platform.

Key Features

1. User Authentication and Authorization

- Secure login and registration.
- Role-based access control (Admin/User).
- Multi-factor authentication for added security.

2. Personalized News Feed

- AI-driven news recommendations based on user interests.
- Real-time updates with categorized news.
- Bookmarking and saving articles for later.

3. Advanced Search and Filtering

- Keyword-based search functionality.
- Filters for categories, sources, and date range.
- Sorting options based on relevance and popularity.

4. Content Management

- Admin dashboard for managing news sources.
- Ability to publish, edit, and remove news articles.
- Fake news detection and verification system.

5. Interactive User Experience

- User comments, likes, and sharing options.
- Dark mode and customizable themes.
- Multi-language support for global accessibility.

6. AI-Powered Insights and Analytics

- Sentiment analysis of news articles.
- Trending topics and news highlights.
- Graphical reports for user engagement.

7. Notifications and Alerts

- Personalized push notifications for breaking news.
- Daily and weekly news digest emails.
- Custom alert settings for preferred topics.

CONCLUSION

CONCLUSION

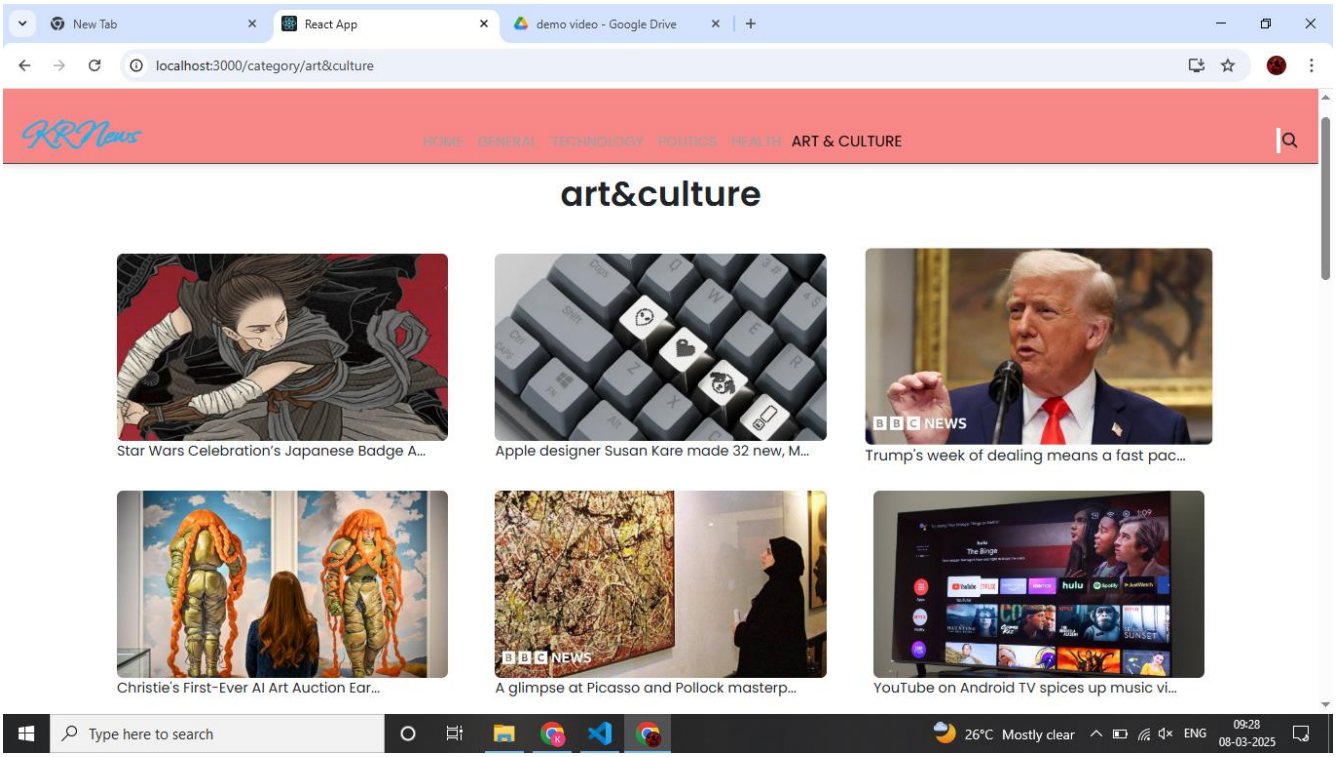
The development of InsightStream: Navigate the News Landscape has successfully created an intelligent, user-friendly, and secure news platform that enhances digital news consumption. By integrating modern web technologies such as React.js, Node.js, MongoDB, and a structured content management system, the platform ensures seamless performance, scalability, and an engaging user experience. Security features like multi-factor authentication, role-based access control, and fake news detection strengthen credibility and user trust. With real-time updates, interactive user engagement through comments and sharing, and an intuitive admin dashboard for content management, InsightStream provides a comprehensive and efficient way for users to stay informed.

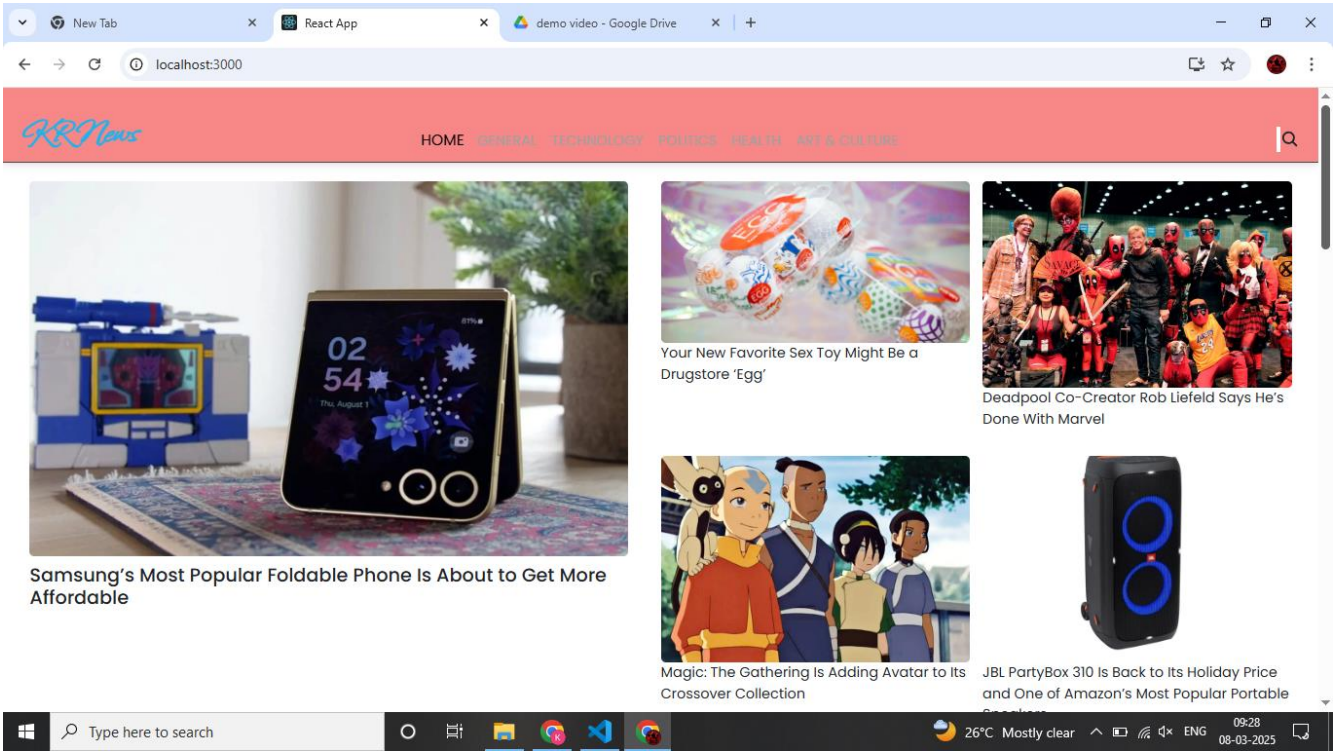
The platform's impact includes personalized content organization, advanced search and filtering capabilities, real-time notifications, and multi-device accessibility, making it a modern and innovative solution for news consumption. Future enhancements could focus on improving content categorization, expanding language

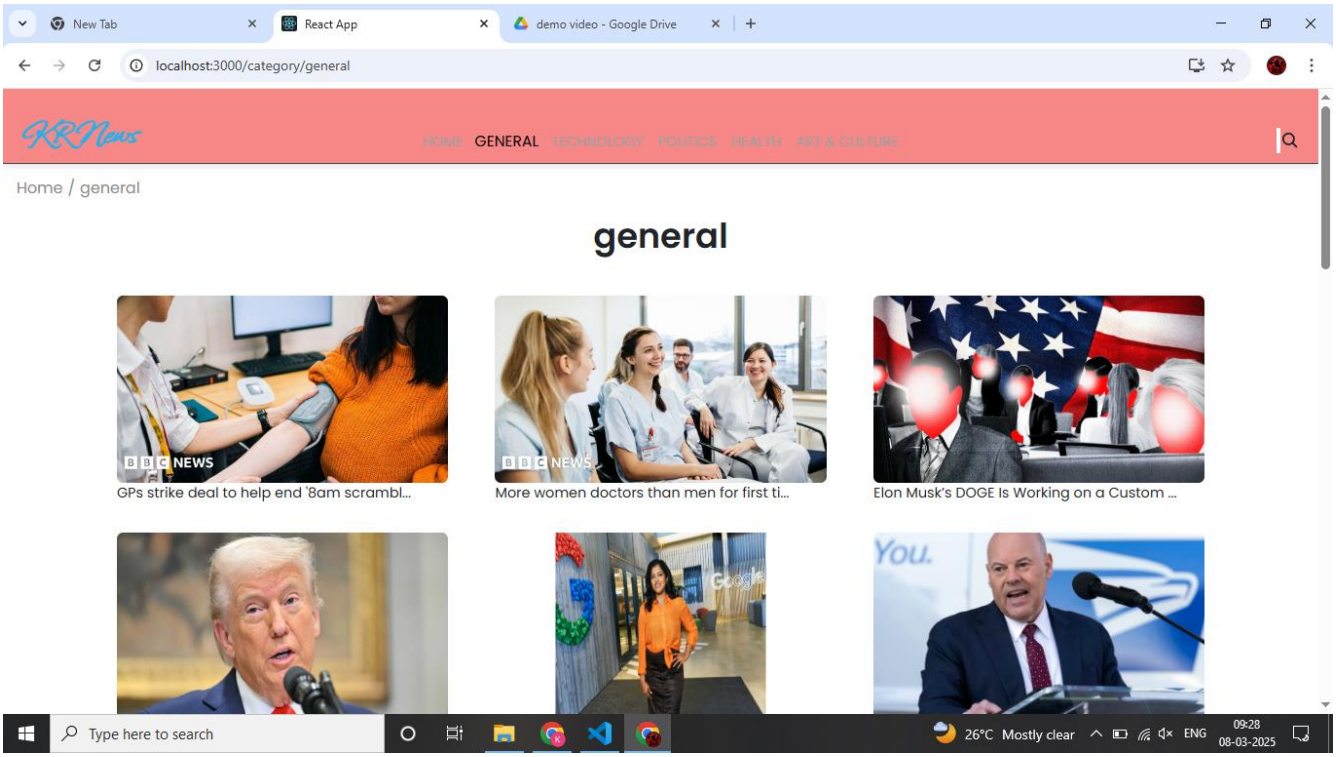
support, refining the user interface, and strengthening verification mechanisms to further improve the accuracy and reliability of news delivery. Additionally, integrating offline reading capabilities and improved performance optimizations would make the platform more versatile and user-friendly. By continuously evolving and adapting to user needs, InsightStream can remain a pioneering force in digital journalism, providing a seamless, reliable, and engaging news experience for users worldwide.

APPENDIX

SCREEN LAYOUTS:


New Tab
React App
demo video - Google Drive
localhost:3000/category/art&culture
KRNews
HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE
art&culture
Star Wars Celebration's Japanese Badge A...
Apple designer Susan Kare made 32 new, M...
BBC NEWS
Trump's week of dealing means a fast pac...
Christie's First-Ever AI Art Auction Ear...
BBC NEWS
A glimpse at Picasso and Pollock masterp...
YouTube on Android TV spices up music vi...
Type here to search
26°C Mostly clear
ENG
09:28
08-03-2025


New Tab
React App
demo video - Google Drive
localhost:3000
KRNews
HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE
Samsung's Most Popular Foldable Phone Is About to Get More Affordable
Your New Favorite Sex Toy Might Be a Drugstore 'Egg'
Deadpool Co-Creator Rob Liefeld Says He's Done With Marvel
Magic: The Gathering Is Adding Avatar to Its Crossover Collection
JBL PartyBox 310 Is Back to Its Holiday Price and One of Amazon's Most Popular Portable
Type here to search
26°C Mostly clear
ENG
09:28
08-03-2025



SOURCE CODE:

```
import React, { useEffect, useState } from 'react'
import '../styles/CategoryPage.css'
import axios from 'axios';
import { useNavigate, useParams } from 'react-router-
dom';
import { Pagination, Spinner } from 'react-bootstrap';

const CategoryPage = () => {

  const navigate = useNavigate();

  const [categoryNews, setCategoryNews] = useState([]);
```

```
  const {id} = useParams();
    useEffect(() => {
      if (id){
        fetchNews(id)
      }
      }, [window.location.pathname])

      const fetchNews = async (id) => {
        try {
          const response = await
axios.get(`https://newsapi.org/v2/everything?q=${id}&api
Key=e42d1a238b774f7d93287a4115c31017`);
          setCategoryNews(response.data.articles);
        } catch (error) {
          console.error(error);
        }
      }

      const [activePage, setActivePage] = useState(1);
      let items = [];
      for (let number = 1; number <= (categoryNews.length /
21); number++) {
        items.push(
```

```
        <Pagination.Item key={number} active={number
=== activePage} onClick={()=> setActivePage(number)}
>
          {number}
        </Pagination.Item>,
      );
    }

  return (
    <div className='category-page'>

      <div className="category-header">
        <div>
          <p onClick={()=> navigate("/")} >Home</p>
          <p>/</p>
          <p>{id}</p>
        </div>
        <h3>{id}</h3>
      </div>

      {categoryNews && categoryNews.length > 0 ?

        <div className="category-articles">

          {categoryNews.map((article, index) => {
```

```
          return index >= (activePage * 21) - 21 && index <
activePage * 21 &&
          (

          <div className="category-article"  key={index}
onClick={()=> window.open(article.url, '_blank')}>
                <img src={article.urlToImage} alt="article " />
                <p>{article.title.slice(0,40)}...</p>

          </div>

         )})}


       </div>
      :
      <div className="spinners">
        <Spinner animation="grow" size="sm" />
        <Spinner animation="grow" size="sm" />
        <Spinner animation="grow" size="sm" />
      </div>}

      <div className="pagination-container">
        <p>Page: </p>
        <Pagination>
          {items}
```

```
        </Pagination>
      </div>
    </div>
  )
}

export default CategoryPage
```