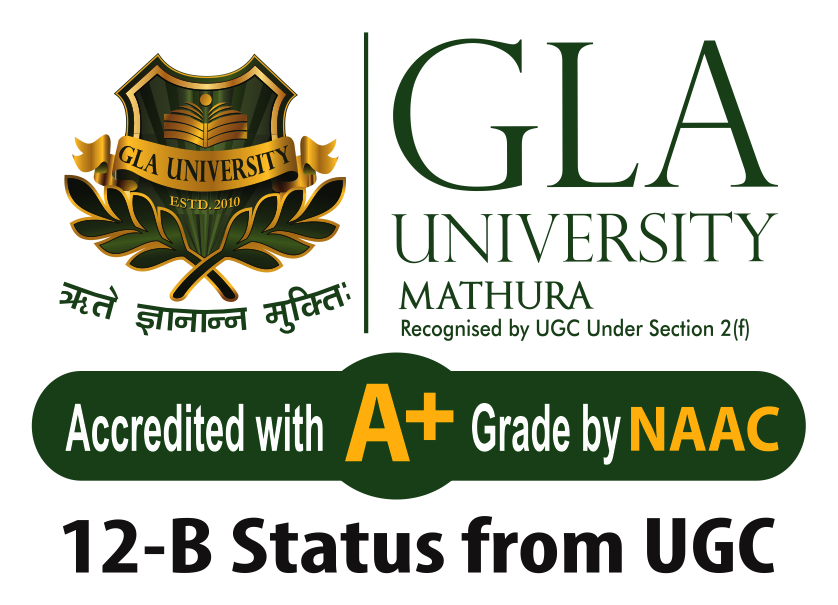
**GLA UNIVERSITY**

**Topic: MINI PROJECT ON “NEWS PORTAL”**

# **SUBMITTED BY: - SUBMITTED TO: -**

**Name: Devansh kumar Faculty Name: Ruchi Gupta**

**University Roll No: 2215000579 (Training Master)**

**Name: Lakshya Bhardwaj**

**University Roll No: 2215000980**

**Name: Navneet Singh**

**University Roll No: 2215001138**

**Name: Pradeep Chauhan**

**University Roll No: 2215001246**

**Name: Vijay Pratap Tomar**

**University Roll No: 2215001944**

Project Report: News Portal Web Application

Using HTML, CSS, JavaScript, Bootstrap, React, API, Backend Integration, and Infinite Scrolling

1. Introduction

In today’s digital age, the demand for real-time information has grown rapidly. People prefer digital news platforms over traditional newspapers due to accessibility, speed, and category-wise filtering. This project, News Portal Web Application, aims to create a responsive, scalable, and real-time news platform that fetches data from a third-party API and renders it on a beautifully designed user interface.

The project is a React-based Single Page Application (SPA) that uses modern front-end technologies along with backend support and deployment tools to ensure high availability, performance, and scalability. It provides news to users across different categories like business, entertainment, technology, health, science, and sports with features like infinite scrolling, category filtering, and a plan to convert this into an official GLA University News Portal accessible only to GLA students.

2. Objective

The core objectives of the News Portal project are:

#To build a dynamic, user-friendly, and responsive news website.

#To provide users with the latest news fetched from reliable public APIs.

#To integrate real-time rendering using JavaScript and React.

#To use infinite scrolling to load news without reloading or pagination.

#To prepare a foundation for a future GLA-exclusive news platform.

#To deploy both frontend and backend for global access and maintenance.

3. Technologies Used

Frontend Technologies:

a. HTML5

Used to structure the web pages semantically. Elements like <section>, <article>, <main> and <aside> were used to provide meaningful markup.

b. CSS3 and Bootstrap

Bootstrap provided responsive design capabilities with grid systems, cards, buttons, and utilities. Combined with custom CSS for personalized design.

c. JavaScript (ES6+)

JavaScript was used for:

\*Making API calls.

\*DOM manipulation.

\*Creating reusable logic for infinite scrolling and dynamic rendering.

d. React.js

React is the core framework of the portal. It provided:

Component-based development.

State and lifecycle handling using hooks (useState, useEffect, etc.).

Smooth navigation with react-router-dom.

e. Axios

Used for efficient API communication, better error handling, and more readable asynchronous code compared to fetch.

Backend Technologies:

a. Node.js & Express.js

Backend server built using Express.js to handle:

API key security (serving news data via server proxy).

Routing.

Future user authentication and content submission by students.

Middleware for error handling and logging.

b. MongoDB (Future Plan)

To store GLA-specific news and user submissions for persistent storage and future updates.

4. Features

✅ Live News Feed

Displays news from top sources and updates in real time.

✅ Category Filtering

Users can click on different tabs (e.g., Business, Sports, Tech) to view categorized content.

✅ Infinite Scrolling

News articles load automatically as the user scrolls down the page using the Intersection Observer API in React. No need for pagination or manual reload.

✅ Responsive Design

Optimized for mobile, tablet, and desktop screens using Bootstrap’s fluid layout and media queries.

✅ Search Functionality

Users can enter keywords and search for specific articles.

✅ Reusable Components

Modular React components like:

✅ Error & Loading States

Custom loading spinners and error messages in case of API failures.

5. Infinite Scrolling Implementation (React)

js

Copy

Edit

useEffect(() => {

const observer = new IntersectionObserver(handleScroll, { threshold: 1 });

if (lastElementRef.current) observer.observe(lastElementRef.current);

return () => observer.disconnect();

}, [articles]);

const handleScroll = (entries) => {

if (entries[0].isIntersecting && !loading && !endReached) {

fetchMoreNews(); // Loads next batch of articles

}

};

This increases UX fluidity by removing pagination and keeping the user engaged.

6. Folder Structure

pgsql

Copy

Edit

news-portal/

├── client/

│ ├── public/

│ └── src/

│ ├── components/

│ ├── App.js

│ ├── index.js

│ └── styles/

├── server/

│ ├── routes/

│ ├── controllers/

│ ├── server.js

│ └── .env

├── .gitignore

├── package.json

7. Backend Integration

#A Node.js + Express.js backend was set up to:

#Act as a middleman between the client and the News API.

#Secure API keys using .env.

#Provide custom endpoints like /api/news/:category.

\*\*In the future, this backend will:

#Connect to MongoDB.

#Allow GLA faculty or students to post campus news.

#Authenticate users using JWT or session-based auth.

8. Deployment

Frontend Deployment:

Vercel or Netlify for the React app.

Continuous Deployment via GitHub.

Backend Deployment:

Render, Railway, or Heroku used to host the Express backend.

Environment variables configured for production.

Auto redeployment on commit.

9. Challenges & Solutions

#Challenge and Solution

API key exposure Moved API key to backend

Load time & performance Added infinite scroll instead of pagination

Cross-origin errors Used proxy server in Express

State sync during scroll Debounced scroll listener, memoized state

Dynamic routing issues Used React Router’s dynamic paths

10. Future Plan: GLA Exclusive News Portal

We plan to evolve this application into “GLA News Portal”, which will be:

✅ For GLA Students Only:

Accessible only to authenticated users via GLA credentials.

Student-specific announcements, campus news, notices.

✅ Admin Panel:

Faculty and admin access to post verified content.

Admin moderation tools to approve student-submitted articles.

✅ Student Contributors:

Students can submit campus event news, placement updates, or academic articles.

Points-based contribution system for encouraging participation.

✅ Notifications:

Real-time push/email notifications for important campus updates.

✅ Mobile App Integration:

React Native version in the future for iOS and Android access.

**Project Data Flow Diagram (DFD):**

**Level 0 DFD:**

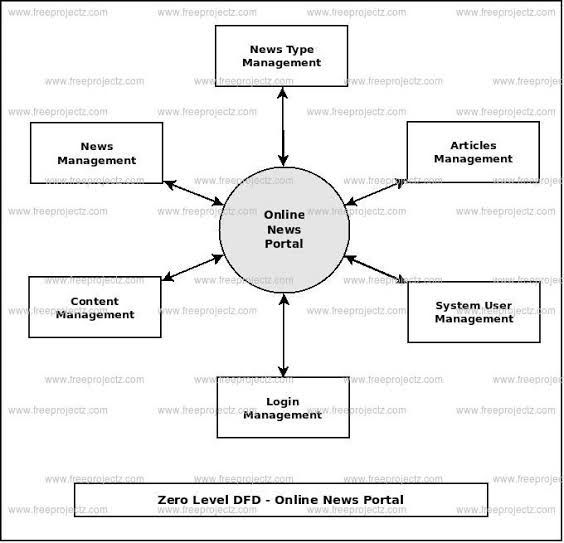
* User interacts with the News Portal to browse articles.
* The system fetches news data from an external API.
* The portal displays categorized news sections.

**Level 1 DFD:**

* User selects a news category (Technology, Sports, etc.).
* The system retrieves filtered articles from the API.
* The user reads the news and can search or filter results.

**Level 2 DFD (if applicable):**

* User logs in (optional feature) and bookmarks articles.
* Bookmarked articles are stored in local storage or a database.
* User can access and manage bookmarked news later.
* Users can share news articles on social media platforms.



11. Conclusion

The News Portal project is not just a news reading platform but a scalable, modular, and real-time digital solution designed with modern technologies. With its clean UI, infinite scrolling, and live API integration, it delivers an exceptional user experience. It lays the groundwork for a full-fledged internal communication platform – the GLA News Portal – that will bridge the gap between campus updates and students.

With future upgrades like login systems, moderation, MongoDB storage, and a mobile app version, the project holds tremendous potential to become a crucial part of GLA’s digital ecosystem.