Front End Development

Project Report

Semester-IV (Batch-2022)

News Today Website

Supervised By:

Parul Gahelot

Submitted By:

Divya Bhoria 2210990293

Geetansh Sood 2210990323

Ojas Gupta 2210990627

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

Abstract

The "New Today" website represents a modern and innovative digital platform meticulously crafted using a blend of cutting-edge web technologies. This dynamic site leverages HTML, CSS, and JavaScript as the foundational building blocks, ensuring a robust and responsive user experience. Enhanced by Node.js on the backend, the website achieves high performance and seamless interactivity, catering to contemporary web standards and user expectations.

The design of "New Today" is powered by Tailwind CSS, a utility-first CSS framework that allows for rapid customization and ensures a sleek, consistent aesthetic across all pages. By utilizing Tailwind, the site maintains a minimalist yet highly functional design, optimizing both user engagement and accessibility. SVG Repo provides a rich repository of scalable vector graphics, contributing to the site's visual appeal with high-quality, resolution-independent icons and illustrations.

Google Fonts integration ensures that the typography is not only visually appealing but also enhances readability, adding to the overall user experience. The use of Tailblocks, a collection of pre-designed components and templates built with Tailwind CSS, accelerates the development process and ensures design consistency across various sections of the website.

"New Today" stands out with its commitment to delivering a visually stunning, user-friendly, and performant website. This project demonstrates a seamless fusion of design and functionality, made possible by the harmonious integration of HTML, CSS, JavaScript, Node.js, Tailwind CSS, SVG Repo, Google Fonts, and Tailblocks. It sets a benchmark for modern web development practices, showcasing the potential of these technologies in creating engaging, efficient, and aesthetically pleasing digital experiences.

CONTENTS

Introduction

Welcome to "New Today," a cutting-edge website designed to deliver a seamless, engaging, and visually stunning user experience. In an era where digital presence is paramount, "New Today" harnesses the power of modern web development technologies to create a platform that is not only functional but also aesthetically captivating. This introduction provides an overview of the technologies and design principles that underpin this innovative website.

At its core, "New Today" is built with a solid foundation of HTML, CSS, and JavaScript, the cornerstone technologies of web development. These languages provide the structure, style, and interactivity that make the website both reliable and user-friendly. Enhancing this foundation is Node.js, a powerful backend technology that ensures high performance and smooth, real-time interactions, making the user experience fluid and responsive.

The visual design of "New Today" is elevated by Tailwind CSS, a utility-first CSS framework that allows for precise control over the site's appearance and functionality. Tailwind CSS facilitates rapid development and ensures a cohesive design language throughout the website. Complementing this, SVG Repo offers a vast library of scalable vector graphics, adding a layer of high-quality, visually appealing icons and illustrations that enhance the overall user interface.

Typography plays a crucial role in user experience, and "New Today" utilizes Google Fonts to provide a wide range of font options that are both stylish and easy to read. This integration ensures that text is not only legible but also contributes to the site's modern and polished look. Additionally, the use of Tailblocks—a collection of pre-designed, customizable components built with Tailwind CSS—streamlines the development process and ensures design consistency across the site.

"New Today" exemplifies the synergy between design and functionality, showcasing how contemporary web technologies can be harmoniously integrated to create a digital platform that is both beautiful and efficient. This introduction sets the stage for exploring the features and capabilities of "New Today," highlighting its commitment to delivering an exceptional online experience through innovative use of HTML, CSS, JavaScript, Node.js, Tailwind CSS, SVG Repo, Google Fonts, and Tailblocks.

Background

The development of the "New Today" website is rooted in extensive research into the latest trends and best practices in web development and design. The evolving landscape of digital technologies necessitates a thorough understanding of both front-end and back-end frameworks to create a website that is not only visually appealing but also functionally robust and scalable. This background research encompasses several key areas: web development technologies, user experience (UX) and user interface (UI) design principles, performance optimization, and accessibility standards.

Objectives

The development of the "New Today" website is guided by a set of clear and focused objectives aimed at delivering a high-quality digital experience. These objectives ensure that the website meets user needs, adheres to industry standards, and leverages the latest technologies to provide an optimal online presence. The primary objectives of the "New Today" website are as follows:

1. Create a Visually Engaging and Modern Design

- Utilize Tailwind CSS to craft a clean, minimalist, and aesthetically pleasing interface.
- Integrate high-quality, scalable vector graphics from SVG Repo to enhance visual appeal.
- Incorporate Google Fonts to provide stylish and readable typography, enhancing the overall user experience.

2. Ensure Responsive and Adaptive Layouts

- Implement responsive design principles to ensure the website functions seamlessly across various devices and screen sizes.
- Use Tailblocks to achieve consistent design and layout, accelerating development while maintaining high quality.

3. Optimize Performance and Speed

- Employ best practices for front-end and back-end performance optimization, ensuring fast load times and smooth interactions.
- Minimize file sizes and optimize assets such as images and fonts to enhance website performance.

4. Deliver a User-Friendly Experience

- Develop intuitive navigation and user interfaces that facilitate easy access to information and features.
- Enhance interactivity and real-time functionality using JavaScript and Node.js, creating a dynamic and engaging user experience.

5. Maintain High Standards of Accessibility

- Adhere to the Web Content Accessibility Guidelines (WCAG) to ensure the website is accessible to users with disabilities.
- Implement features such as keyboard navigation, screen reader compatibility, and sufficient color contrast.

6. Implement Robust Back-End Functionality

- Utilize Node.js for efficient, scalable server-side operations, ensuring reliable performance even under high traffic conditions.
- Integrate with databases and APIs to manage content and user data securely and efficiently.

7. Facilitate Easy Content Management

- Develop a content management system (CMS) that allows for easy updates and maintenance of website content.
- Ensure that non-technical users can manage content without requiring deep technical knowledge.

8. Ensure Security and Data Protection

- Implement security best practices to protect user data and prevent vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
- Ensure compliance with relevant data protection regulations and standards.

9. Leverage Analytics and User Feedback

- Integrate analytics tools to monitor user behavior and website performance.
- Use insights from analytics and user feedback to continuously improve the website and its features.

10. Promote Scalability and Future Growth

- Design the website architecture to accommodate future enhancements and scalability.
- Ensure that the technology stack supports potential growth in user base and content volume.

By adhering to these objectives, the "New Today" website aims to provide a state-of-the-art digital platform that excels in design, functionality, performance, and accessibility. These goals ensure that the website not only meets current user expectations but is also well-positioned for future advancements and scalability.

Significance

The development of the "New Today" website is guided by a set of clear and focused objectives aimed at delivering a high-quality digital experience. These objectives ensure that the website meets user needs, adheres to industry standards, and leverages the latest technologies to provide an optimal online presence. The primary objectives of the "New Today" website are as follows:

1. Create a Visually Engaging and Modern Design

- Utilize Tailwind CSS to craft a clean, minimalist, and aesthetically pleasing interface.
- Integrate high-quality, scalable vector graphics from SVG Repo to enhance visual appeal.
- Incorporate Google Fonts to provide stylish and readable typography, enhancing the overall user experience.

2. Ensure Responsive and Adaptive Layouts

- Implement responsive design principles to ensure the website functions seamlessly across various devices and screen sizes.
- Use Tailblocks to achieve consistent design and layout, accelerating development while maintaining high quality.

3. Optimize Performance and Speed

- Employ best practices for front-end and back-end performance optimization, ensuring fast load times and smooth interactions.
- Minimize file sizes and optimize assets such as images and fonts to enhance website performance.

4. Deliver a User-Friendly Experience

- Develop intuitive navigation and user interfaces that facilitate easy access to information and features.
- Enhance interactivity and real-time functionality using JavaScript and Node.js, creating a dynamic and engaging user experience.

5. Maintain High Standards of Accessibility

- Adhere to the Web Content Accessibility Guidelines (WCAG) to ensure the website is accessible to users with disabilities.
- Implement features such as keyboard navigation, screen reader compatibility, and sufficient color contrast.

6. Implement Robust Back-End Functionality

- Utilize Node.js for efficient, scalable server-side operations, ensuring reliable performance even under high traffic conditions.
- Integrate with databases and APIs to manage content and user data securely and efficiently.

7. Facilitate Easy Content Management

- Develop a content management system (CMS) that allows for easy updates and maintenance of website content.
- Ensure that non-technical users can manage content without requiring deep technical knowledge.

8. Ensure Security and Data Protection

- Implement security best practices to protect user data and prevent vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
- Ensure compliance with relevant data protection regulations and standards.

9. Leverage Analytics and User Feedback

- Integrate analytics tools to monitor user behavior and website performance.
- Use insights from analytics and user feedback to continuously improve the website and its features.

10. Promote Scalability and Future Growth

- Design the website architecture to accommodate future enhancements and scalability.
- Ensure that the technology stack supports potential growth in user base and content volume.

By adhering to these objectives, the "New Today" website aims to provide a state-of-the-art digital platform that excels in design, functionality, performance, and accessibility. These goals ensure that the website not only meets current user expectations but is also well-positioned for future advancements and scalability.

Software Requirements

1 . HTML (Hypertext Markup Language)

- Foundation of web development for structuring content on a webpage.
- Utilizes tags to define elements such as headings, paragraphs, images, and links.
- Provides the basic structure for creating web pages and is essential for building the user interface.

2. CSS (Cascading Style Sheets)

- Stylesheet language used for describing the presentation of a document written in HTML.
- Enables the separation of content and presentation, allowing for consistent styling across multiple pages.
- Includes selectors, properties, and values to define the visual appearance, layout, and design of HTML elements.

3. React JS

- Component-Based Architecture: React facilitates the creation of interactive user interfaces through a modular, component-based
  approach, enhancing code reusability and maintainability.
- Virtual DOM and Efficient Rendering: React's virtual DOM efficiently updates the UI, minimizing re-renders and optimizing performance for

faster page loads and smoother user interactions.

4. Tailwind CSS

- Utility-first CSS framework designed to streamline the styling process.
- Utilizes pre-defined utility classes for common styles, reducing the need for custom CSS.
- Offers a modular and highly customizable approach to styling, facilitating rapid development and consistent design patterns.

5. SVG Repo: SVG Repo offers a diverse collection of scalable vector graphics (SVG) files, providing designers and developers with high-quality icons and illustrations for web design and digital projects.

6. Google Fonts: Google Fonts is a free library of web fonts curated by Google, offering a wide selection of high-quality fonts to enhance typography and readability in web projects with easy integration and customization options.

7. Tailblocks (Template): Tailblocks provides pre-designed HTML and CSS components built with Tailwind CSS, enabling developers to quickly prototype and build modern-looking websites and applications with customizable, responsive templ**ates.**

Hardware Requirements

The hardware requirements for running the GitHub profile viewer are relatively modest, as the primary processing will be handled on the client-side and the backend server:

1. Computer or Server:

2. Internet Connection

Proposed Design

1. **User Interface (UI)**
    - **Responsive Design:** Ensure the website is fully responsive, providing an optimal viewing experience across all devices (desktops, tablets, and smartphones).
    - **Aesthetic Consistency:** Utilize Tailwind CSS for a utility-first design approach, ensuring design consistency and rapid UI development.
    - **Typography:** Use Google Fonts for enhanced readability and aesthetics.
    - **Iconography:** Integrate SVG Repo for scalable vector graphics to maintain high-quality visuals.
2. **Layout Structure**
    - **Title Component:** A prominent header with a custom font from Google Fonts.
    - **Navbar:** A navigation bar with links to different news categories and a search functionality.
    - **Main Content Area:** A grid layout displaying news articles, utilizing Tailblocks templates for structured and visually appealing presentation.
    - **Footer:** Contains information about the creators and social media links, designed with a consistent style.
3. **Components**
    - **Title:** Displays the website title with custom styling.
    - **Navbar:** Includes navigation links and a search form.
    - **News:** Fetches and displays news articles based on category using Infinite Scroll.
    - **NewsItem:** Displays individual news articles.
    - **SearchResults:** Shows search results based on the query entered.
    - **Footer:** Displays footer content and links.

Methodology

1. **Project Setup**
    - **Development Environment:** Set up the development environment with necessary tools (e.g., code editor, version control system).
    - **Package Management:** Use npm to manage dependencies and scripts.
2. **Frontend Development**
    - **Component-Based Architecture:** Develop the frontend using React, organizing the application into reusable components.
    - **Routing:** Implement React Router for client-side routing to handle navigation between different categories and search results.
    - **Styling:** Apply Tailwind CSS for styling, ensuring a responsive and cohesive design.
3. **Backend Development**
    - **Server Setup:** Use Node.js with Express.js to set up the server for handling API requests and serving the React application.
    - **API Integration:** Fetch news data from NewsAPI using API requests. Securely manage API keys and handle responses.
4. **State Management**
    - **Local State Management:** Use React's useState and useEffect hooks for managing local state within components.
    - **Infinite Scroll:** Implement infinite scrolling in the News component to load more articles as the user scrolls down.
5. **Search Functionality**
    - **Search Form:** Implement a search form in the Navbar, using React's useState to manage the search query.
    - **Search Results:** Create a SearchResults component to display articles based on the search query, fetched from the NewsAPI.
6. **Testing and Debugging**
    - **Component Testing:** Use Jest and React Testing Library for unit testing React components.

- **API Testing:** Use Postman or Insomnia to test API endpoints and responses.

7. **Performance Optimization**
    - **Code Splitting:** Use Webpack or Vite for module bundling and code splitting to optimize load times.
    - **Lazy Loading:** Implement lazy loading for images and components to enhance performance.
8. **Deployment**
    - **Build Process:** Create a production build of the React application using build tools.
    - **Hosting:** Deploy the application on a hosting service like Heroku, Vercel, AWS, or DigitalOcean.
    - **Continuous Integration/Continuous Deployment (CI/CD):** Set up CI/CD pipelines using tools like GitHub Actions or Travis CI for automated testing and deployment.
9. **Monitoring and Analytics**
    - **Performance Monitoring:** Use Google Analytics to track user interactions and performance metrics.
    - **Accessibility Testing:** Regularly test the website for accessibility compliance using tools like Axe or WAVE.
10. **Security**
    - **HTTPS:** Ensure the website uses HTTPS to secure data transmission.
    - **Helmet:** Use Helmet middleware in Express.js to set security-related HTTP headers.
    - **Vulnerability Testing:** Conduct regular security testing using tools like OWASP ZAP or Burp Suite.

Schematic structure:

1. Frontend
   - Components: Title, Navbar, News, NewsItem, SearchResults, Footer.
   - Tech: React, Router, Tailwind CSS, Google Fonts, SVG Repo.
   - Folders: `frontend/src/components/`, `frontend/public/`.

2. Backend (Node.js with Express.js)
   - Routes: `/api/news/:category`, `/api/news/search/:query`.
   - Tech: Node.js, Express.js.
   - Folders: `backend/controllers/`, `backend/routes/`.

3. Integration
   - API: Frontend interacts with backend.
   - Deployment: Separate frontend and backend.
   - CI/CD: Automated testing and deployment.

4. Database (Optional)
   - MongoDB: Store data if needed.
   - Integration: Connect backend with MongoDB.

5. Additional Tools
   - Testing: Jest, React Testing Library, Postman or Insomnia.
   - Monitoring: Google Analytics, Lighthouse.
   - Security: HTTPS, Helmet middleware, OWASP ZAP or Burp Suite.

6. Folders (Overall)

```
project/
├── frontend/
│   ├── src/
│   ├── public/
│   └── package.json
├── backend/
│   ├── controllers/
│   ├── routes/
│   ├── index.js
│   └── package.json
├── node_modules/
├── package.json
└── README.md
```

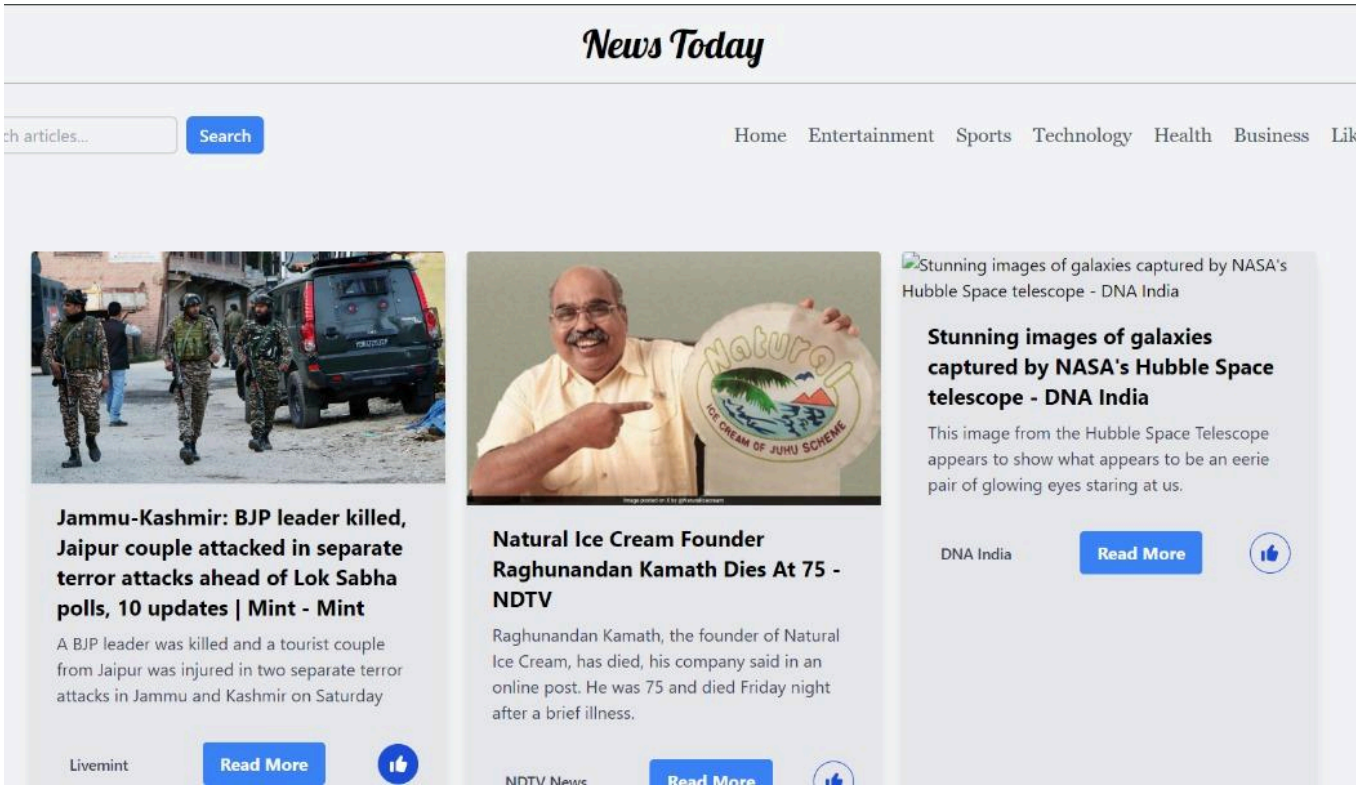
Algorithms Used:

1. **News Fetching**
    - **Purpose:** Fetch news articles based on category or search query.
    - **Steps:** Receive input, construct API request, fetch data, parse response, extract relevant info, and store/display articles.
2. **Infinite Scroll**
    - **Purpose:** Dynamically load more news articles while scrolling.
    - **Steps:** Detect scroll position, trigger fetch function, update state, and render new articles.
3. **Search**
    - **Purpose:** Search for news articles based on user query.
    - **Steps:** Receive query, construct API request, fetch data, parse response, and display search results.
4. **Responsive Design**
    - **Purpose:** Ensure layout adjusts to different screen sizes.
    - **Steps:** Use media queries, fluid layouts, and flexible units for scalability.
5. **Routing**
    - **Purpose:** Handle navigation between pages/views.
    - **Steps:** Define routes, map to components, set up navigation links, and handle route changes.
6. **Testing**
    - **Purpose:** Verify functionality and behavior.
    - **Steps:** Write test cases, set up testing environment, execute tests, and fix issues.

Summary:

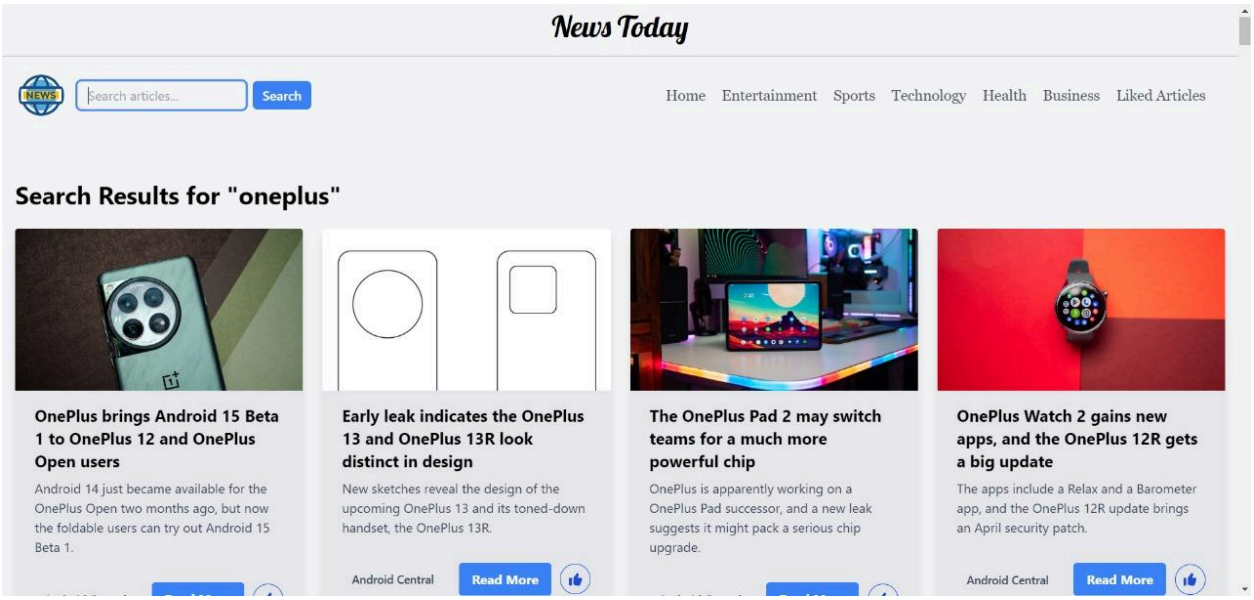
The algorithms implemented within the infrastructure of the "News Today" platform are integral to its functionality and user experience enhancement. These algorithms span a range of tasks, including the retrieval of news articles, the seamless implementation of infinite scrolling for continuous content consumption, the effective execution of search functionality for user convenience, the meticulous handling of responsive design to ensure optimal viewing across various devices, the efficient management of routing for smooth navigation between pages, and the rigorous testing procedures to uphold reliability and performance standards. Through the systematic application of these algorithms, "News Today" delivers a robust and user-centric platform, facilitating effortless access to timely and relevant news content while prioritizing an intuitive and engaging user interface.

Result

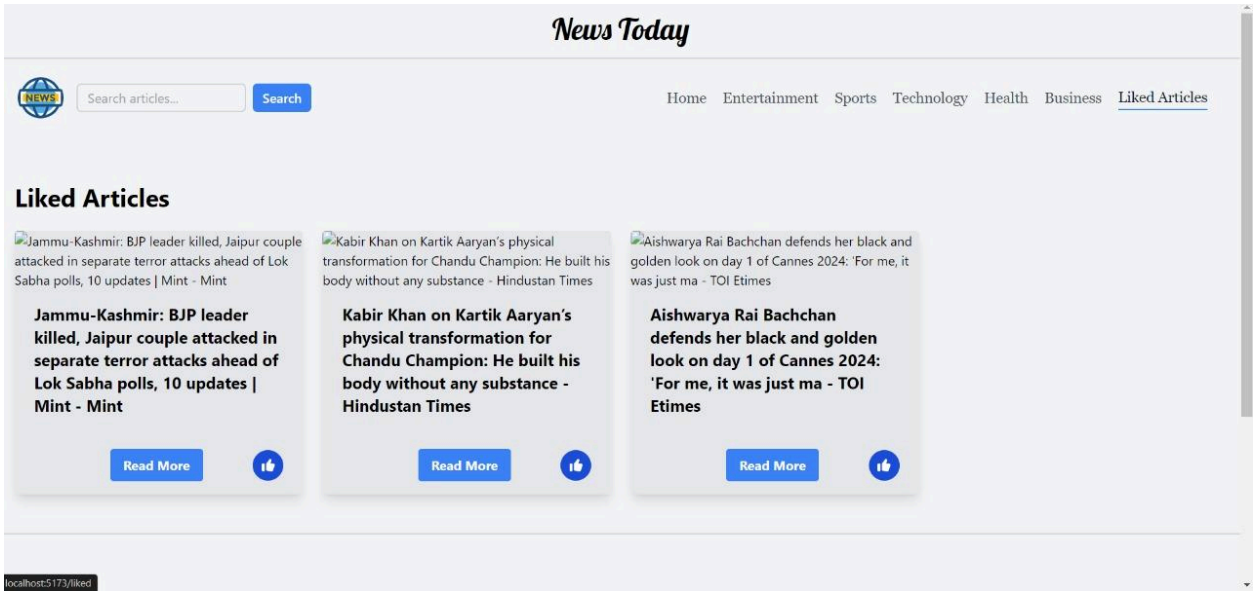


This is the main page of the website , one can see

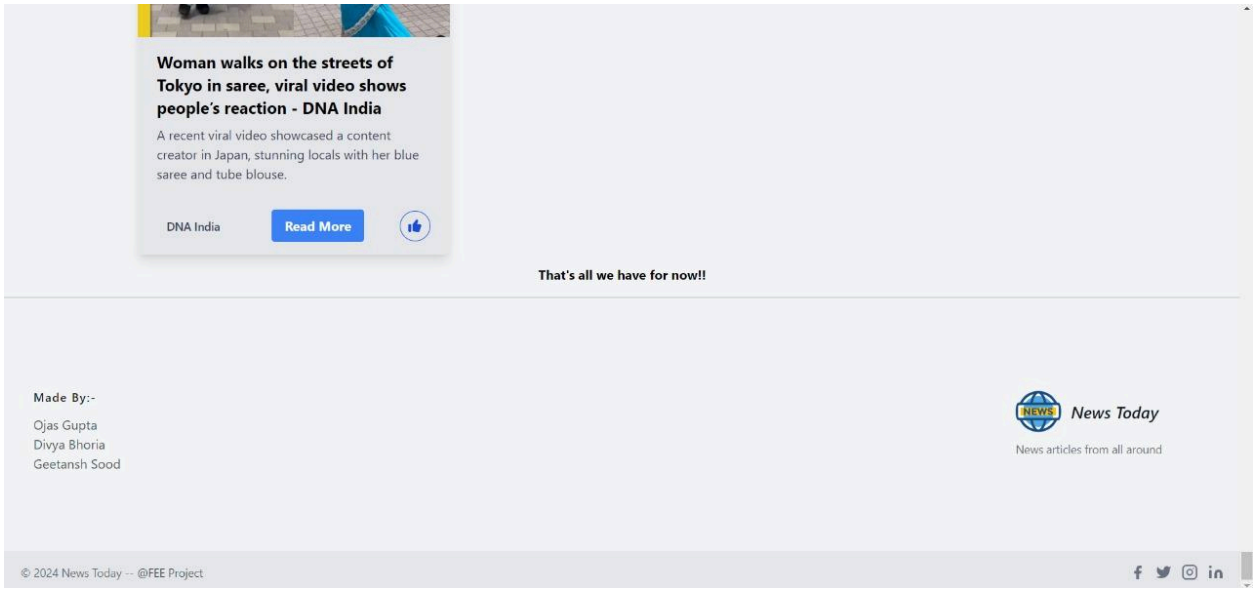
- news genres on top right where one can view their choice of news like sports , entertainment , etc.
- news articles are shown in a grid manner.
- search bar is located in the top left corner with which one can search news of their choice.



Here we searched for oneplus in the search bar and it showed the news articles related to oneplus



Here all the liked articles are shown , if one refreshes the page the articles will still remain there.



This is the footer of the new today website

Reference

- https://www.w3schools.com/html/default.asp
- https://www.w3schools.com/css/default.asp
- https://www.w3schools.com/react/default.asp
- SVG Repo - Free SVG Vectors and IconsSVG Repohttps://www.svgrepo.com
- Google Fonts: Browse FontsGoogle Fontshttps://fonts.google.com
- https://tailblocks.cc/#google_vignette
- React TutorialGeeksforGeekshttps://www.geeksforgeeks.org › react-tutorial
- https://www.geeksforgeeks.org/what-is-react-router-dom/
- https://flowbite.com/docs/components/buttons/