

# **InsightStream: Navigate the News Landscape**

By

*BHUMI PRASAD-AP23110010218*

*JEETESH KUMAR VERMA-AP23110010223*

*AYUSH KUMAR-AP23110010278*

*NAMAN UPADHYAY-AP23110010558*

SUBMITTED TO:

**Yatharth Shahrawat**

**Department of Computer Science and Engineering SRM University-AP**

**In partial fulfilment for the requirements of the project**

**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND  
ENGINEERING**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SRM UNIVERSITY-AP**

**NEERUKONDA**

**MANAGALAGIRI - 522503**

**ANDHRA PRADESH, INDIA**

**DECEMBER-2025**

## Introduction:

In an era where information flows faster than ever, staying updated with reliable and relevant news has become both essential and challenging. With hundreds of digital platforms publishing content every minute, users often struggle to filter meaningful stories from the noise. **NewsWave** is a full-stack news aggregation and discovery platform designed to simplify this experience by bringing curated, organized, and personalized news to users in one unified space.

NewsWave leverages modern web technologies to collect news from multiple sources, categorize them intelligently, and present them through a clean and user-friendly interface. The platform allows users to browse trending topics, search for specific categories, and engage with real-time updates. By implementing efficient backend data processing and a responsive frontend design, NewsWave ensures seamless navigation and an enhanced reading experience.

The primary objective of NewsWave is to empower users to “ride the wave of information” with clarity and convenience. Whether it is breaking news, technology updates, entertainment highlights, or global affairs, the platform helps users stay informed without feeling overwhelmed. Through its thoughtful architecture and interactive features, NewsWave aims to redefine how modern audiences explore and consume digital news.

## Scenario-Based Intro:

Imagine a college student named Arjun, who starts his day by checking news across different platforms—social media, news websites, and mobile apps. Each source shows different information, some cluttered with ads, some lacking credibility, and others overwhelming him with too many unrelated stories. By the time he finds reliable updates on topics he actually cares about—technology, world affairs, and science—he has already spent more time searching than reading.

Now consider Priya, a working professional. Her schedule barely allows her to scroll through multiple apps. She wants quick, accurate, and category-wise news summaries that fit into her busy routine. But with scattered sources and constant information overload, staying updated feels like a daily challenge.

These scenarios reflect a common problem faced by millions of users today:

- Too many news sources

- Lack of personalization

- No single platform that provides clean, organized, and relevant updates

**NewsWave** was created to address exactly this challenge.

NewsWave acts as a unified news ecosystem where users can “ride the wave of information” effortlessly. Instead of navigating multiple websites, users can open NewsWave and instantly access curated news categorized into topics like Politics, Technology, Sports, Entertainment, Health, and more. With a clean UI, smooth navigation, real-time fetching, and personalized recommendations, NewsWave transforms the traditional news-reading journey into a simpler, smarter, and more enjoyable experience.

By integrating robust backend APIs with a responsive and user-friendly frontend, NewsWave demonstrates how full-stack technology can improve everyday digital tasks and solve real-world problems. Ultimately, NewsWave is not just a platform—it is a modern approach to consuming news in a fast-moving world.

# Target Audience:

NewsWave is designed for a wide range of users who rely on digital news platforms to stay informed. The platform focuses on delivering a personalized, organized, and efficient news-reading experience that caters to the needs of the following audiences:

## 1. Students and Young Learners

Individuals who want quick access to current affairs, educational news, and trending topics without navigating multiple platforms. NewsWave helps them stay updated for exams, competitions, and general awareness.

## 2. Working Professionals

Users with tight schedules who need fast, credible, and category-based news updates. NewsWave offers a streamlined experience that saves time while providing essential insights.

## 3. Researchers and Analysts

People who require structured and topic-wise information to support research, case studies, or analysis. NewsWave's organized categories and real-time data make it easier to track trends and gather information.

## 4. News Enthusiasts

Regular readers who enjoy staying informed about global events, technology updates, sports highlights, entertainment, and more. NewsWave ensures a clean, ad-minimal reading experience suited for daily use.

## 5. Casual Readers

Users who do not actively follow news but want important stories curated in an easy-to-read format. NewsWave provides simplified, relevant content that enhances convenience.

## 6. Mobile-First Users

People who prefer consuming news on the go. The responsive design of NewsWave ensures a smooth experience across smartphones, tablets, and desktops.

# Project Goals and Objectives:

## Overall Goal

The primary goal of **NewsWave** is to create a unified, intelligent, and user-friendly news aggregation platform that simplifies how people consume digital news by providing organized, personalized, and real-time updates in one place.

## Specific Objectives

### 1. Deliver a Centralized News Platform

To aggregate news from multiple reliable sources and present it in a single, easy-to-navigate interface, eliminating the need for users to switch between different apps or websites.

### 2. Provide Topic-Wise Categorization

To classify news articles into categories such as Technology, Sports, Politics, Business, Entertainment, Health, and more, offering users a clear and structured browsing experience.

### 3. Enhance User Personalization

To implement features such as preferences, bookmarks, trending topics, and recommended categories, allowing each user to customize their news feed.

### 4. Ensure Real-Time Updates

To fetch and display the latest news as it happens through API integration, ensuring timely and accurate information.

### 5. Create an Intuitive and Responsive UI/UX

To design a clean, minimal, and mobile-friendly interface that supports smooth navigation and provides a seamless reading experience across all devices.

## **6. Optimize Backend Performance**

To develop a scalable backend capable of efficiently handling requests, storing user interactions, and managing news data without delays.

## **7. Improve Reliability and Credibility**

To source news from trusted APIs and present it without misinformation, ensuring users receive authentic content.

## **8. Support User Engagement Features**

To include features such as bookmarking, saving articles, reading history, and topic-based exploration to increase user involvement.

## **9. Demonstrate Full-Stack Competency**

To showcase strong skills in frontend development, backend architecture, database design, API integration, and deployment through the development of a complete full-stack application.

## **10. Enable Future Scalability**

To design the system architecture in a way that allows future enhancements such as user authentication, AI-based recommendations, sentiment analysis, and push notifications.

# PRE-REQUISITES: -

## 1. Code Editor (VS Code Recommended)

A professional code editor is essential for writing, organizing, and managing the project source code.

- **Visual Studio Code** is recommended as it provides excellent support for JSX, modern JavaScript, React components, and JSON structures.
- Helpful extensions include:
  - Prettier
  - ESLint
  - React Snippets
  - Live Server

## 2. Web Browser (Chrome / Firefox)

A modern web browser is required to test, debug, and run the **NewsWave** React application.

- **Google Chrome** is preferred due to its advanced Developer Tools and compatibility with the React DevTools extension.
- Firefox or Microsoft Edge may also be used for UI testing and cross-browser compatibility checks.

## 3. Node.js and npm

Node.js is required to run the React environment, and npm (Node Package Manager) is used to install dependencies.

- Node.js provides the runtime needed for executing React scripts.
- npm installs the required packages such as:
  - React



- Fetch API
- JSON Server (optional)

**React projects cannot run without Node.js and npm.**

#### **4. React.js Framework Knowledge**

Since **NewsWave** is built using React.js, the developer should have a basic understanding of:

- Creating functional components using **JSX**
- Managing **props** and **state**
- React Hooks such as **useState** and **useEffect**
- Using **React Router** for navigation (if implemented)
- Event handling and dynamic rendering
- Integrating the Fetch API within React components

This knowledge ensures smooth development of features such as category pages, search results, dynamic news rendering, and UI updates.

#### **5. Fetch API Knowledge (External News Data)**

The NewsWave application retrieves data from external news APIs; therefore, the developer should understand:

- Making GET requests using the **fetch()** method
- Parsing JSON responses
- Displaying API data dynamically (headlines, descriptions, images, categories)
- Implementing search functionality
- Handling errors such as network issues or empty API responses

## 6. JSON Server (Optional Mini-Backend)

If the project includes features such as saved articles or bookmarked news, **JSON Server** can be used as a lightweight backend.

Knowledge required:

- Starting JSON Server using:  
`npx json-server --watch db.json`
- Performing CRUD operations:
  - **GET** → Retrieve saved data
  - **POST** → Add new entries
  - **PUT/PATCH** → Update existing items
  - **DELETE** → Remove data
- Structuring JSON for local storage

JSON Server acts as a simple local database for development and testing.

## 7. Local Development Server (React Scripts)

React automatically provides a local development server using:

`npm start`

This eliminates the need for tools like Apache, PHP, or MySQL.

For backend simulation, developers may use:

- JSON Server (recommended)
- A custom Node.js API
- Static JSON files

## 8. Basic HTML, CSS, and JavaScript Skills

Although React handles most of the UI logic, developers should understand:

- JSX for structuring UI layouts
- CSS for styling, animations, or responsive design
- JavaScript ES6+ features such as:
  - Arrow functions
  - Promises
  - `async/await`
  - Array methods (`map`, `filter`, `reduce`)

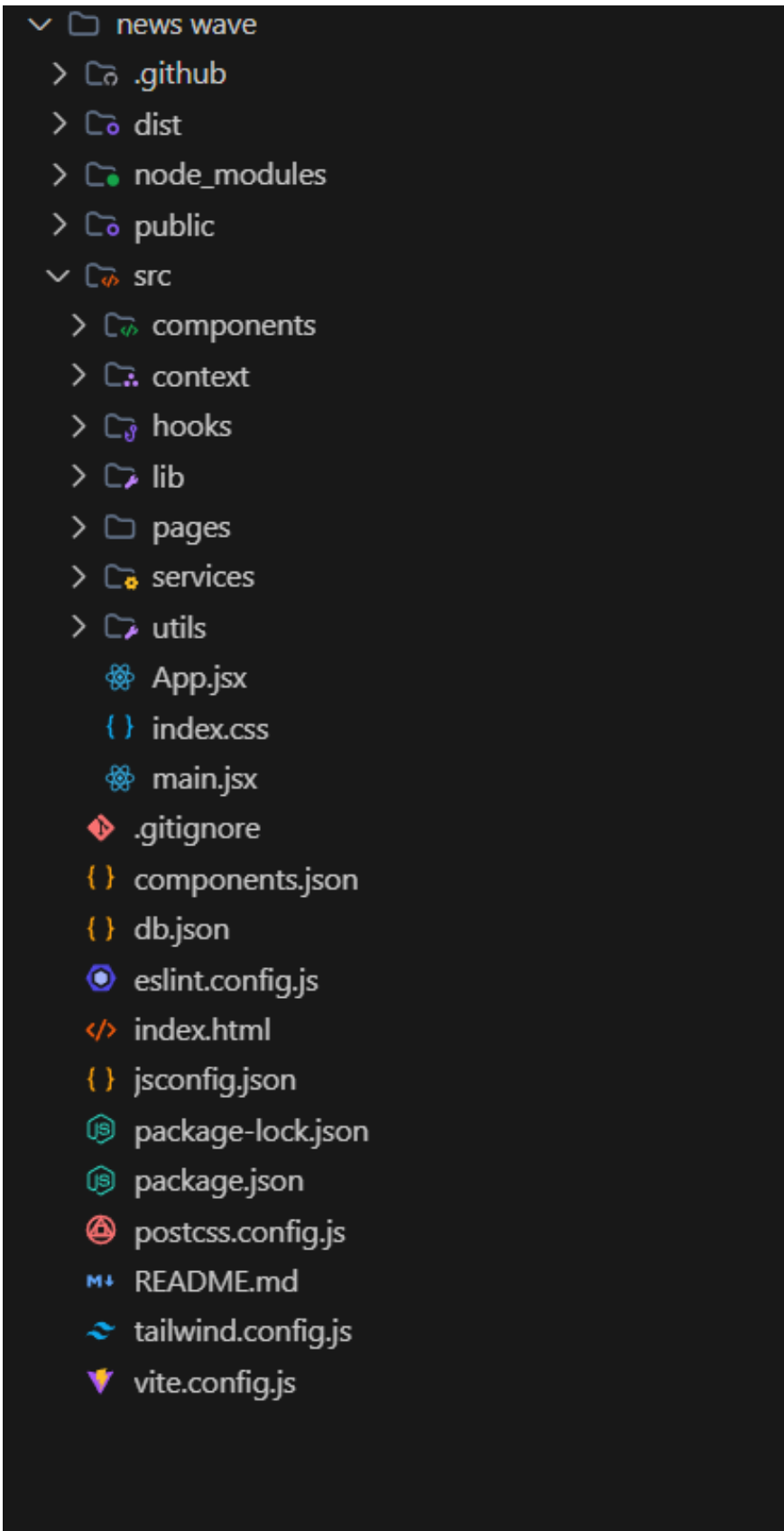
These skills are essential for implementing components like news lists, search bars, article pages, and custom user interfaces.

## 9. System Requirements

To run the **NewsWave Application** efficiently, the system must meet the following specifications:

- Minimum **4GB RAM** (8GB recommended)
- Dual-core processor
- At least **5GB** free disk space
- Stable internet connection for installing npm packages and retrieving news from APIs

## Project structure:



```

  news wave
  ├── .github
  ├── dist
  ├── node_modules
  ├── public
  └── src
      ├── components
      ├── context
      ├── hooks
      ├── lib
      ├── pages
      ├── services
      ├── utils
      ├── App.jsx
      ├── index.css
      ├── main.jsx
      ├── .gitignore
      ├── components.json
      ├── db.json
      ├── eslint.config.js
      ├── index.html
      ├── jsconfig.json
      ├── package-lock.json
      ├── package.json
      ├── postcss.config.js
      ├── README.md
      ├── tailwind.config.js
      └── vite.config.js

```

A screenshot of a file explorer showing the project structure of 'news wave'. The structure is as follows:

- news wave
  - .github
  - dist
  - node\_modules
  - public
  - src
    - components
    - context
    - hooks
    - lib
    - pages
    - services
    - utils
  - App.jsx
  - index.css
  - main.jsx
  - .gitignore
  - components.json
  - db.json
  - eslint.config.js
  - index.html
  - jsconfig.json
  - package-lock.json
  - package.json
  - postcss.config.js
  - README.md
  - tailwind.config.js
  - vite.config.js

The project contains:

- **Frontend (React.js)**

Contains all UI elements and screens.

- **Components:** Navbar, Footer, Recipe Cards, Search Bar, Voice Search, Timer, Checklist, Theme Toggle, etc.
- **Pages:** Home, About, Community, MyCookBook, UploadRecipe, EditRecipe, Signup, Login.
- **Contexts:** Global states like language and theme.
- **Styles:** Global CSS and theme files.
- **Utils:** API handlers, localStorage utilities.

- **Backend (JSON Server)**

- **db.json** stores user-created and community recipes.
- JSON Server handles CRUD operations for recipes without needing MySQL or PHP.

- **Fetch API (TheMealDB)**

- Uses **Fetch API** to load thousands of recipes from TheMealDB.
- Supports search by dish, ingredient, or category.

- **Assets (Media Files)**

- Includes images, icons, and graphics used in UI components.

- **Configuration Files**

- **package.json:** Project dependencies and scripts.
- **vite.config.js:** Vite build configuration.
- **db.json:** Local database.
- **.gitignore:** Ignored files.
- **index.html:** Root file for React.

- **server.js:** JSON Server setup.

## PROJECT FLOW: -

### Project demo:

Before starting to work on this project, let's see the demo.

Demolink:

Use the code in:

### App.js component

```
import { Toaster } from "@components/ui/toaster";
import { Toaster as Sonner } from "@components/ui/sonner";
import { TooltipProvider } from "@components/ui/tooltip";
import { QueryClient, QueryClientProvider } from "@tanstack/react-query";
import { ThemeProvider } from "next-themes";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Layout from "../components/Layout";
import Home from "../pages/Home";
import CategoryPage from "../pages/CategoryPage";
import ArticlePage from "../pages/ArticlePage";
import Favorites from "../pages/Favorites";
import Settings from "../pages/Settings";
import Admin from "../pages/Admin";
import NotFound from "../pages/NotFound";

const queryClient = new QueryClient({
  defaultOptions: {
    queries: {
      refetchOnWindowFocus: false,
      retry: 1,
    },
  },
});

const App = () => (
  <QueryClientProvider client={queryClient}>
    <ThemeProvider attribute="class" defaultTheme="light" enableSystem={false}>
      <TooltipProvider>
        <Toaster />
        <Sonner position="top-right" />
        <BrowserRouter>
          <Routes>
            <Route element={<Layout />} />
            <Route path="/" element={<Home />} />
            <Route path="/category/:slug" element={<CategoryPage />} />
            <Route path="/article/:id" element={<ArticlePage />} />
            <Route path="/favorites" element={<Favorites />} />
          </Routes>
        </BrowserRouter>
      </ThemeProvider>
    </QueryClientProvider>
  )
);
```

```

        <Route path="/settings" element={<Settings />} />
        <Route path="/admin" element={<Admin />} />
        <Route path="*" element={<NotFound />} />
      </Route>
    </Routes>
  </BrowserRouter>
</TooltipProvider>
</ThemeProvider>
</QueryClientProvider>
);

export default App;

```

## Code Description:

### 1. Importing Core Libraries and Components

The code begins by importing essential modules:

UI Components

- Toaster & Sonner → Used for showing toast notifications (alerts, success messages, errors).
- TooltipProvider → Enables tooltip UI across the app.

State Management (React Query)

- QueryClient, QueryClientProvider → Manage API caching, fetching, and background updates.

Theme Management

- ThemeProvider → Controls Light/Dark theme support.

Routing (react-router-dom)

- BrowserRouter, Routes, Route → Handle navigation and routing between pages.

Custom Pages

These imports point to different pages in your app:

- Home
- CategoryPage

- ArticlePage
- Favorites
- Settings
- Admin
- NotFound
- Layout

Each represents a screen within the NewsWave application.

## 2. Creating the QueryClient

```
const queryClient = new QueryClient({
  defaultOptions: {
    queries: {
      refetchOnWindowFocus: false,
      retry: 1,
    },
  },
});
```

This block configures React Query, which manages API calls:

- `refetchOnWindowFocus:` false  
→ Prevents refetching data every time the user switches tabs.
- `retry:` 1  
→ If an API request fails, it will retry once.

This improves performance and reduces unnecessary API calls.

## 3. App Component Structure

```
const App = () => (
```



The entire application is wrapped inside a series of providers, each adding global functionality.

#### **4. Providers Used in the App**

##### a) QueryClientProvider

Wraps the app to enable React Query caching and API state management.

##### b) ThemeProvider

Enables:

- Light mode by default
- System theme disabled (enableSystem={false})

##### c) TooltipProvider

Enables tooltips across all pages.

##### d) Toaster / Sonner

Provides toast notifications for success/error messages.

#### **5. Routing Setup (BrowserRouter & Routes)**

The `<BrowserRouter>` defines client-side routing, and `<Routes>` contains all navigable paths.

Layout Route

`<Route element={<Layout />}>`

The Layout component wraps all pages, typically containing:

- Navbar
- Footer
- Sidebar
- Shared UI elements

#### **6. Defined Routes**

Inside the Layout:

Homepage

```
<Route path="/" element={<Home />} />
```

Categories Page

```
<Route path="/category/:slug" element={<CategoryPage />} />
```

Uses a dynamic URL parameter (slug) such as:

- /category/sports
- /category/technology

Article Page

```
<Route path="/article/:id" element={<ArticlePage />} />
```

Shows details of a single news article based on its id.

Favorites Page

```
<Route path="/favorites" element={<Favorites />} />
```

Settings

```
<Route path="/settings" element={<Settings />} />
```

Admin Page

```
<Route path="/admin" element={<Admin />} />
```

404 Not Found Page

```
<Route path="*" element={<NotFound />} />
```

Handles invalid URLs and displays a "Page Not Found" message.

## 7. Exporting the App

```
export default App;
```

This makes the App component available to the main entry file (usually main.jsx), where it is rendered into the DOM.

Summary of What This Code Does

Feature	Description
Routing	Creates all navigation paths (Home, Category, Article, Favorites, Settings, Admin).
Layout	Wraps all pages inside a consistent layout structure.
React Query	Manages API calls and caching.
Theme Support	Enables light mode and theme switching.
Notifications	Adds toast messages through Toaster/Sonner.
Global Providers	Wraps the entire app with necessary utilities.

This file forms the core structure of the NewsWave frontend application.

## Project Execution:

After completing the code, run your NewsWave JSON server using “npm run server” and start the frontend using “npm run dev”. These commands will launch your JSON Server API and your Vite React application.

```
PS C:\NewsWave> npm run server

> vite_react_shadcn_ts@0.0.0 server
> json-server --watch db.json --port 3001

--watch/-w can be omitted, JSON Server 1+ watches for file changes by default
JSON Server started on PORT :3001
Press CTRL-C to stop
Watching db.json...
```

```
PS C:\NewsWave> npm run dev

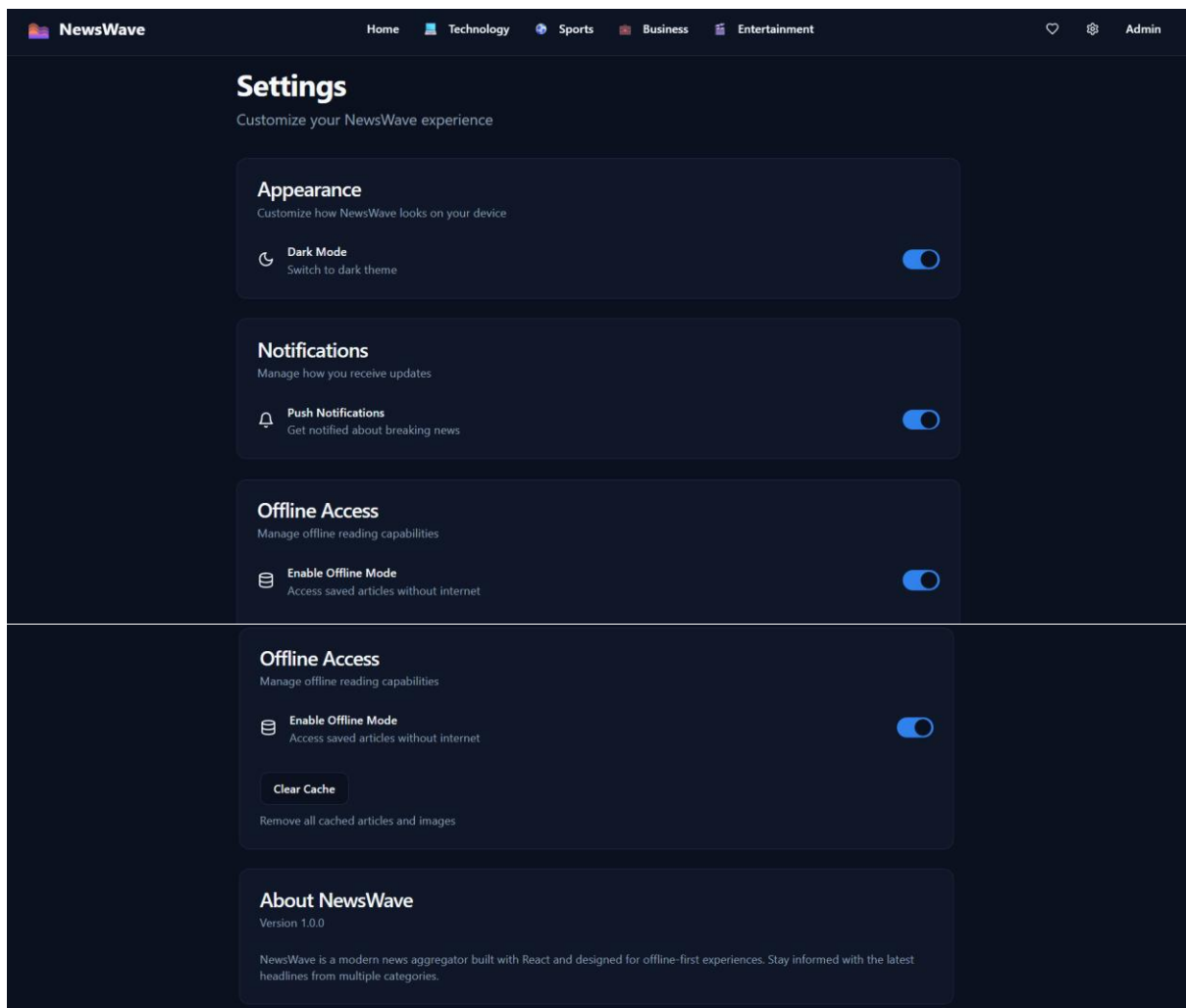
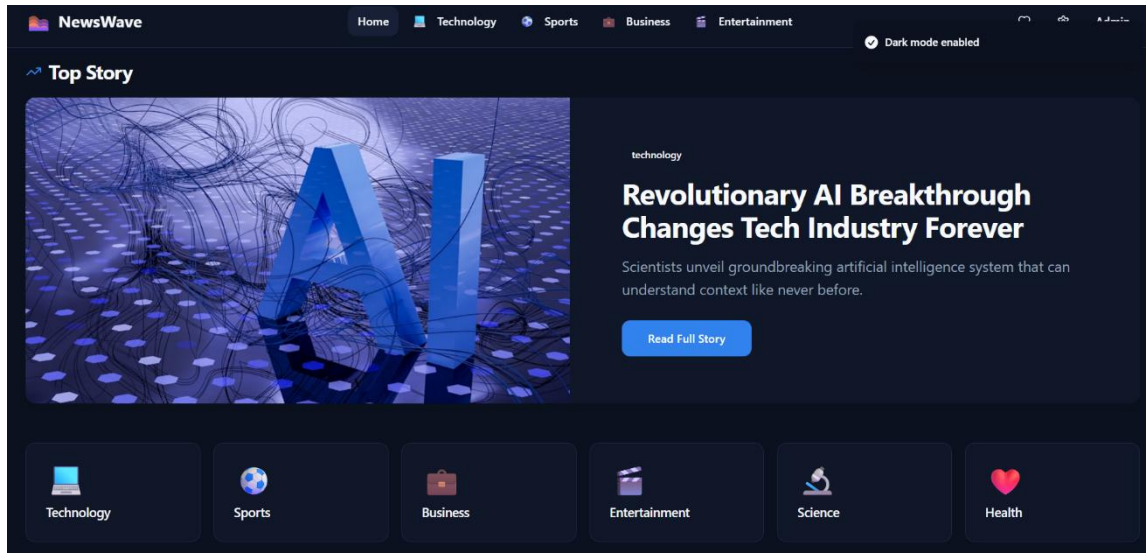
> vite_react_shadcn_ts@0.0.0 dev
> vite

Re-optimizing dependencies because vite config has changed

VITE v5.4.19  ready in 597 ms

→ Local:   http://localhost:8080/
→ Network: http://10.2.1.88:8080/
→ press h + enter to show help
```

# Output:



NewsWave

HomeTechnologySportsBusinessEntertainment


Admin

Entertainment

Latest entertainment news and updates


Search entertainment articles...

entertainment



**Award-Winning Film Captivates Global Audiences**  
New cinematic masterpiece receives critical acclaim and breaks box office records.  
Entertainment Weekly almost 2 years ago

entertainment



**Streaming Service Announces Highly Anticipated Series**  
Major platform reveals exciting new content lineup for subscribers.  
Entertainment Insider almost 2 years ago

NewsWave

HomeTechnologySportsBusinessEntertainment

Admin

Saved Articles


Your favorite articles are saved and available offline

Search saved articles...

ExportImport


2 saved articles

business



**Global Markets Reach All-Time Highs Amid Recovery**  
Stock markets worldwide celebrate strong economic indicators and investor confidence.  
Business Daily almost 2 years ago

entertainment



**Award-Winning Film Captivates Global Audiences**  
New cinematic masterpiece receives critical acclaim and breaks box office records.  
Entertainment Weekly almost 2 years ago

