Project Documentation

**Prime news**

1. **Introduction:**

* **Project Title:** PRIME NEWS
  + **Team ID** : NM2025TMID38546
  + **Team Leader**: JANANI. S &janani2511jannu@gmail.com[Coder]
  + **Team Members & [Roles]**

1. JANANI[.M & jananimunikrishnan@gmail.com](mailto:.M%20%20&%20%20jananimunikrishnan@gmail.com)[Documenter]

2. JANAPRIYA.J & [janapriya6002@gmail.com](mailto:janapriya6002@gmail.com)[Documenter]

3. JENIFER.D & jeniferdhanraj13@gmail.com[Coder]

**2. Project Overview:**

**Purpose**:

* Project helps users quickly move between different sections/pages of your news website.
* Keeps the site organized, so readers can find content categories without confusion.

**Features:**

* **Personalization** – Delivers insights tailored to the user’s role, goals, or projects.
* **Clarity & Simplicity** – Provides short, clear messages (like news headlines) that highlight what’s important and what to do next.
* **Prioritization** – Surfaces the most critical insights first, so users don’t get lost in too much data
* **Integration** – Connects with analytics, SEO, marketing, or customer data sources to pull insights automatically.
* **Public Interest** – Focuses on topics that affect or attract people.
* **Balance** -All sides of an issue news are covered fairly

**3. Architecture:**

**Frontend:**

Frontend is the part of a website or app that users see and interact with directly. It includes design, layout, and interactivity built using HTML, CSS, and JavaScript.

* [**HTML**](https://www.google.com/search?sca_esv=01d7d1da4f3739f4&cs=0&sxsrf=AE3TifN-R0gD-ochhu5MTx7Tg4FwpTWMmg%3A1757584320383&q=HTML&sa=X&ved=2ahUKEwjL4qzot9CPAxXMQ2cHHbMbOwQQxccNegQIDhAB&mstk=AUtExfDuxLr5Bn5l-fW8Xs1HHgN_GJImbp-YBOMeXiV3tvXXOana1ZMa3s4bMttADDkbxYCCqwFrTDjCkmeXqck3HutlI9HukzYySi-xzHEnVOSovv5Vv9YOcV2i75coDOp5ccBxYNa9lcCSTiGVvd-JwoPTt2Kh-N1NRyEtkz7WI8aJ5lByxQd6YyvmmS0Gr02dZG-CRSNHksRrP-5I-gPapyH5ph8ZgVDhT73GgOKHbpTcV2dlOHeelCH8X-VPgfrrChHcMKQXkIbJ_ZjXcvvVhbIB&csui=3)**:** Hyper Text Markup Language structure the content and layout of the insight stream.
* [**CSS**](https://www.google.com/search?sca_esv=01d7d1da4f3739f4&cs=0&sxsrf=AE3TifN-R0gD-ochhu5MTx7Tg4FwpTWMmg%3A1757584320383&q=CSS&sa=X&ved=2ahUKEwjL4qzot9CPAxXMQ2cHHbMbOwQQxccNegQIExAB&mstk=AUtExfDuxLr5Bn5l-fW8Xs1HHgN_GJImbp-YBOMeXiV3tvXXOana1ZMa3s4bMttADDkbxYCCqwFrTDjCkmeXqck3HutlI9HukzYySi-xzHEnVOSovv5Vv9YOcV2i75coDOp5ccBxYNa9lcCSTiGVvd-JwoPTt2Kh-N1NRyEtkz7WI8aJ5lByxQd6YyvmmS0Gr02dZG-CRSNHksRrP-5I-gPapyH5ph8ZgVDhT73GgOKHbpTcV2dlOHeelCH8X-VPgfrrChHcMKQXkIbJ_ZjXcvvVhbIB&csui=3)**:** Cascading Style Sheets style the stream, making it visually appealing and ensuring responsiveness across different devices.
* [**JavaScript**](https://www.google.com/search?sca_esv=01d7d1da4f3739f4&cs=0&sxsrf=AE3TifN-R0gD-ochhu5MTx7Tg4FwpTWMmg%3A1757584320383&q=JavaScript&sa=X&ved=2ahUKEwjL4qzot9CPAxXMQ2cHHbMbOwQQxccNegQIFBAB&mstk=AUtExfDuxLr5Bn5l-fW8Xs1HHgN_GJImbp-YBOMeXiV3tvXXOana1ZMa3s4bMttADDkbxYCCqwFrTDjCkmeXqck3HutlI9HukzYySi-xzHEnVOSovv5Vv9YOcV2i75coDOp5ccBxYNa9lcCSTiGVvd-JwoPTt2Kh-N1NRyEtkz7WI8aJ5lByxQd6YyvmmS0Gr02dZG-CRSNHksRrP-5I-gPapyH5ph8ZgVDhT73GgOKHbpTcV2dlOHeelCH8X-VPgfrrChHcMKQXkIbJ_ZjXcvvVhbIB&csui=3)**:** [JavaScript](https://www.google.com/search?sca_esv=01d7d1da4f3739f4&cs=0&sxsrf=AE3TifN-R0gD-ochhu5MTx7Tg4FwpTWMmg%3A1757584320383&q=JavaScript&sa=X&ved=2ahUKEwjL4qzot9CPAxXMQ2cHHbMbOwQQxccNegQIFBAB&mstk=AUtExfDuxLr5Bn5l-fW8Xs1HHgN_GJImbp-YBOMeXiV3tvXXOana1ZMa3s4bMttADDkbxYCCqwFrTDjCkmeXqck3HutlI9HukzYySi-xzHEnVOSovv5Vv9YOcV2i75coDOp5ccBxYNa9lcCSTiGVvd-JwoPTt2Kh-N1NRyEtkz7WI8aJ5lByxQd6YyvmmS0Gr02dZG-CRSNHksRrP-5I-gPapyH5ph8ZgVDhT73GgOKHbpTcV2dlOHeelCH8X-VPgfrrChHcMKQXkIbJ_ZjXcvvVhbIB&csui=3) add interactivity, handle data updates, and manage the real-time nature of the stream.

**Backend:**

Backend is the server-side part of an application that handles logic, databases, and APIs. It works behind the scenes to process requests and deliver data to the frontend.

* **Node.js** - scalable for real-time updates.

**Database:**

A database is an organized collection of data stored electronically for easy access and management. It helps in storing, retrieving, and updating information efficiently.

* **Mongo DB** - if content is highly flexible (JSON-based storage).

**4. Setup Instructions**

* **Prerequisites:**
* Node.js
* Mongo DB
* Git
* React.js
* Mongoose
* Visual Studio Code
  + Git hub
  + CSS
  + Java script

**• Installation Steps:**

# Clone the repository git clone

# Install client dependencies cd

client npm install

# Install server dependencies

../server npm install

**5. Folder Structure:**

project-root/

├── node\_modules/

├── public/

├── src/

│ ├── assets/

│ │ ├── images/

│ │ └── icons/

│ ├── components/

│ ├── pages/

│ ├── context/

│ ├── styles/

├── .gitignore

├── package-lock.json

├── package.json

└── README.md

**6.** **Running the Application :**

**• Frontend:**

Frontend is the part of a website or app that users see and interact with directly. It includes design, layout, and interactivity built using HTML, CSS, and JavaScript.

cd client npm start

**• Backend:**

Backend is the server-side part of an application that handles logic, databases, and APIs. It works behind the scenes to process requests and deliver data to the frontend.

cd server npm start

**• Access:**

Visit <http://localhost:3000>

**7. API Documentation**

**• User:**

* + - **/api/user/signup:**

const SignUp = ({ onClose }) => {

const [fullName, setFullName] = useState('');

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

alert(`Thank you for signing up, ${fullName}! Welcome to Prime News.`);

onClose();

};

<h2>Prime News - Sign Up</h2>

<input

type="text"

placeholder="Full Name"

value={fullName}

onChange={(e) => setFullName(e.target.value)}

<input

type="email"

placeholder="Email"

value={email}

<input

type="text"

placeholder="Full Name"

value={fullName}

onChange={(e) => setFullName(e.target.value)}

required

autofocus

/>

<input

type="email"

placeholder="Email"

value={email}

onChange={(e) => setEmail(e.target.value)}

required

/>

<input

type="password"

value ={password}

onChange ={(e) => setPassword(e.target.value)}

required

/>

export default SignUp;

* + - **/api/user/Signin:**

const SignIn = ({ onClose }) => {

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

const handleSubmit = (e) => {

return (

<div className="auth-container">

<form className="auth-form" onSubmit={handleSubmit}>

<h2>Prime News - Sign In</h2>

<input

type="email"

placeholder="Email"

value={email}

onChange={(e) => setEmail(e.target.value)}

required

autoFocus

/>

<input

type="password"

placeholder="Password"

value={password}

onChange={(e) => setPassword(e.target.value)}

required

/>

<button type="submit" className="submit-btn">Sign In</button>

<button type="button" onClick={onClose} className="link-btn">Back</button>

</form>

</div>

);

};

export default SignIn;

**• Projects:**

* **Navbar components**

<div className='navbar'>

<div className='navbar\_\_logo'>

<h2 onClick={()=> {setActive("Home"); navigate("/")}}>Prime News</h2>

</div>

<div className='navbar\_\_links'>

<p className={active === 'Home' ? 'active' : ''} onClick={()=> {setActive("Home"); navigate("/")}} >Home</p>

<p className={active === 'General' ? 'active' : ''} onClick={()=> {setActive("General"); navigate("/category/general")}}>General</p>

<p className={active === 'Technology' ? 'active' : ''} onClick={()=> {setActive("Technology"); navigate("/category/technology")}}>Technology</p>

<p className={active === 'Politics' ? 'active' : ''} onClick={()=> {setActive("Politics"); navigate('/category/politics')}}>Politics</p>

<p className={active === 'Health' ? 'active' : ''} onClick={()=> {setActive("Health"); navigate("/category/health")}}>Health</p>

<p className={active === 'Art & Culture' ? 'active' : ''} onClick={()=> {setActive("Art & Culture"); navigate("/category/art&culture")}}>Art & Culture</p>

</div>

<div className="navbar\_\_search">

<input className='search-input' type="text" placeholder="Search" onChange={(e)=> setSearch(e.target.value)} value={search} />

<IoSearch className='search-icon' onClick={()=> handleSearch()} />

</div>

</div>

)

}

export default NavbarComponent

**newsletter:**

<div className='newsletter-container'>

<h2>Subscribe to the newsletter</h2>

<h4>Get a weekly digest of our most important stories direct to your inbox.</h4>

<form className='newsletter-form'>

<input type='text' placeholder='Enter Your Email' onChange={(e)=> setEmail(e.target.value)} value={email} />

<button onClick={hangleSubmit} >Subscribe</button>

</form>

{showA ?

<div className='newsletter-success'>

<p>Thank you for subscribing!</p>

<p>You will receive a confirmation email shortly.</p>

</div>

:""}

<p>Place some disclaimer text here about how subscriber’s email, Privacy Policy and all that.</p>

</div>

)

}

export default NewsLetter

**8. User Interface** :

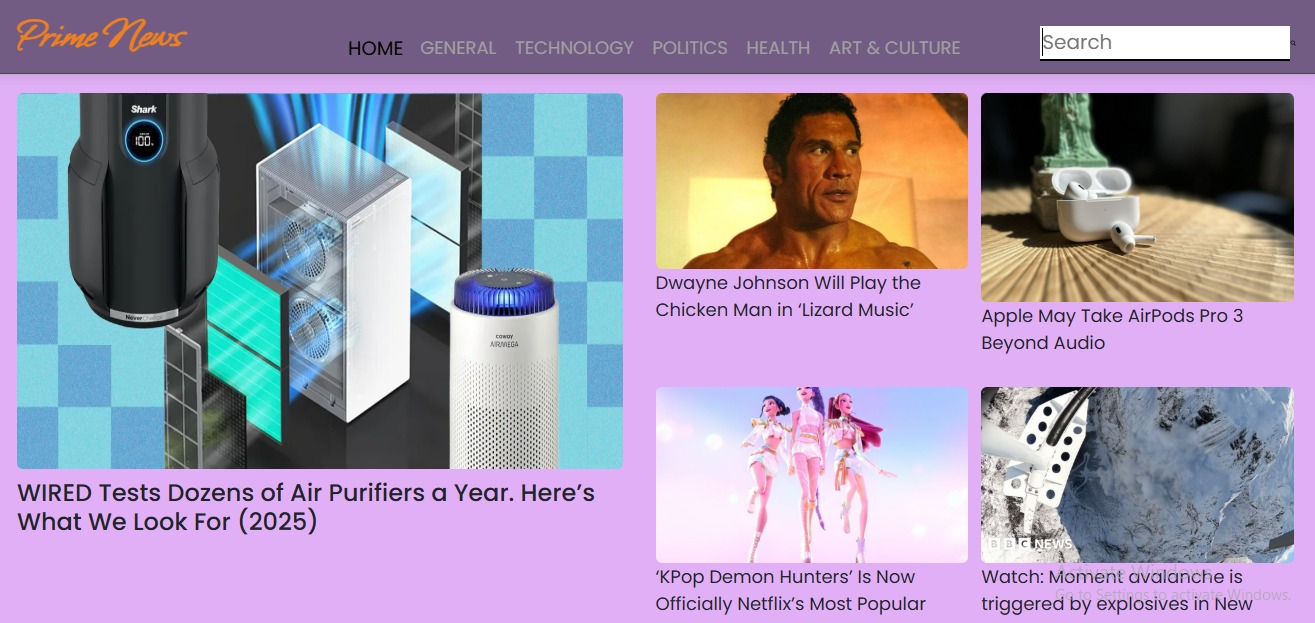
* **Signup page:**

****

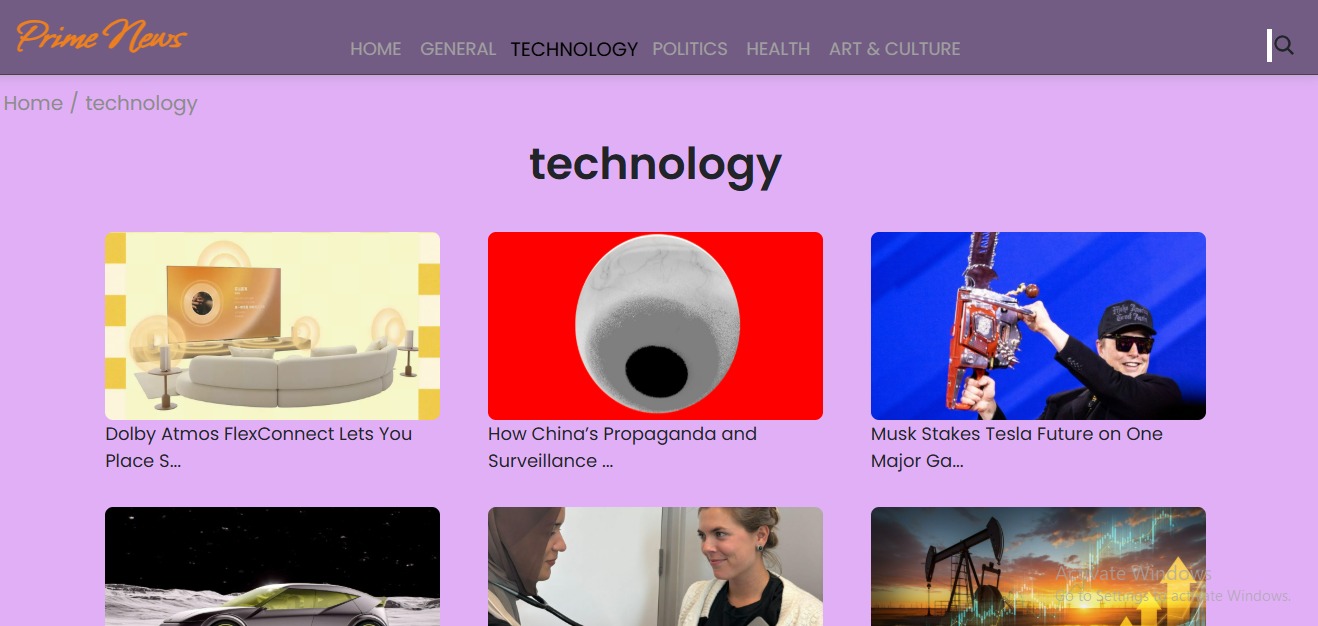
* **Signin page:**

****

* **Home page:**



* **Technology page:**



**9. demo video:**

**https://drive.google.com/file/d/1AbfpH5pAkl\_Nqqo2p8K02hDkZrqpIQtW/view?usp=drive\_link**

**10. Known Issues:**

* No Offline news stream

**11. Future Enhancements:**

* Push notifications → Send alerts for breaking news.
* Voice-enabled news reading (for accessibility).
* Dark/Light mode → Give readers a choice of themes for comfort.
* Newsletter subscription → Send daily/weekly top stories to emails.