Insight Stream - The News: Project Documentation

1. Introduction

Project Title: Insight Stream - The News

Project Type: React-based news aggregation and streaming frontend

Team Leader: HARISH T.D – Tdharishharish@gmail.com

Team Members: JOHN PAUL.A – johnpaul1112006@gmail.com

KARTHICK.D – Sivakarthikarthi446@gmail.com

KARTHICK RAJA.S – Karthickraja1625@gmail.com

Purpose: This project delivers a modern, responsive frontend that aggregates and displays news articles and live streams.

2. Project Overview

Purpose:
Insight Stream aims to provide users with a fast, accessible, and customizable way to follow news across categories. It aggregates headlines, full articles, and live news streams into a single React application.

Key Features:
- Home feed with top stories and category filters
- Article detail pages with related articles and tags
- Live streaming integration for breaking news or scheduled shows
- Search and saved articles/bookmarks
- Responsive layout for mobile and desktop
- Theming support (light/dark)
- Basic accessibility considerations (keyboard navigation, ARIA attributes)

3. Architecture

Component Structure:
The app follows a component-driven architecture. Major components include:
- App (root) — sets up routing and global providers
- NavBar — site navigation and category selectors
- HomeFeed — lists aggregated story cards
- StoryCard — reusable card for each article summary
- ArticlePage — full article view with metadata and related content
- StreamPlayer — component that wraps embedded live streams

- SearchResults — displays search output

State Management:
Use React Context for lightweight global state (user preferences, theme, bookmarks). For heavier data flows (caching feeds, optimistic updates), use a data-fetching library like React Query.

Routing:
Use react-router-dom. Routes example:
- / — HomeFeed
- /category/:name — Category filtered feed
- /article/:id — ArticlePage
- /search?q= — SearchResults
- /live — Live streams

4. Setup Instructions

Prerequisites:
- Node.js (v16+ recommended)
- npm or yarn

Installation:
1. Clone the repository: git clone <repo-url>
2. cd insight-stream-client
3. Install dependencies: npm install (or yarn)
4. Create a .env file and add required environment variables (REACT_APP_API_BASE, REACT_APP_STREAM_KEY, etc.)

Environment Variables (example):
- REACT_APP_API_BASE=https://api.example.com
- REACT_APP_STREAM_PROVIDER_URL=...

5. Folder Structure

Client (React app):

```
src/
├── components/     # Reusable UI components (StoryCard, Button, Inputs)
├── pages/          # Page-level components (HomeFeed, ArticlePage, LivePage)
├── hooks/          # Custom hooks (useFetchArticles, useBookmarks)
├── context/        # React Context providers (ThemeProvider, UserProvider)
├── services/       # API wrappers and streaming utilities
├── styles/         # Global styles, variables and themes
├── assets/         # Static assets (icons, fonts)
```

```
└── App.jsx
```

Utilities:
- helpers/ for small utility functions (date formatting, URL builders)
- constants/ for app constants and config

6. Running the Application

Development server:
1. cd insight-stream-client
2. npm start

Production build:
1. npm run build
2. Serve the build directory with a static server (e.g., serve -s build) or integrate with your hosting pipeline.

7. Component Documentation

Key Components:

App.jsx
- Purpose: App root, sets up Router and global providers
- Props: none

NavBar
- Purpose: Top navigation, category selection, search input
- Props: onSearch(query), activeCategory

HomeFeed
- Purpose: Fetch and render StoryCard list
- Props: category, page

StoryCard
- Purpose: Present article preview (title, snippet, source, timestamp)
- Props: story {id, title, snippet, source, publishedAt}

ArticlePage
- Purpose: Show full article content and related articles
- Props: articleId

StreamPlayer
- Purpose: Embed or wrap streaming provider SDKs/iframes

- Props: streamUrl, config

8. State Management

Global State:
- Theme (light/dark)
- User preferences (saved categories)
- Bookmarks

Local State:
- Component-level UI states (open modals, local form inputs)

Data Fetching:
Use React Query (or similar) to cache article feeds and simplify loading / error states. Keep server state in the query cache and local UI state in components.

9. User Interface

Design Approach:
- Responsive grid for lists and a single-column reading view on mobile
- Prioritize content readability: legible font sizes, adequate line-height

Accessibility:
- Use semantic HTML elements
- Include ARIA labels where needed
- Ensure keyboard navigation across interactive elements

10. Styling

CSS Frameworks/Libraries:
- Tailwind CSS recommended for utility-first styling and rapid iteration. Alternatively, use Styled-Components for component-scoped styles.

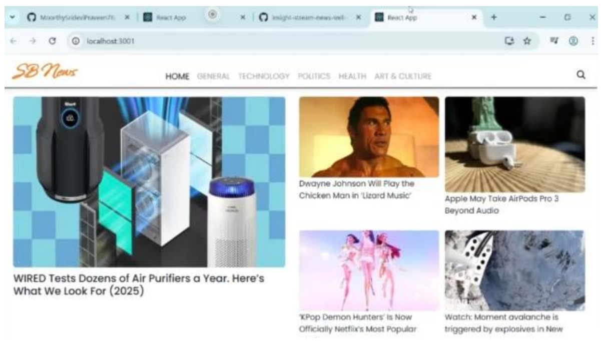
Theming:
- Provide CSS variables or a theme provider to toggle light/dark modes.

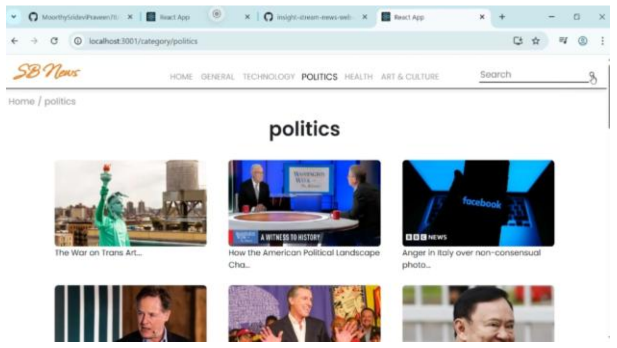
11. Testing

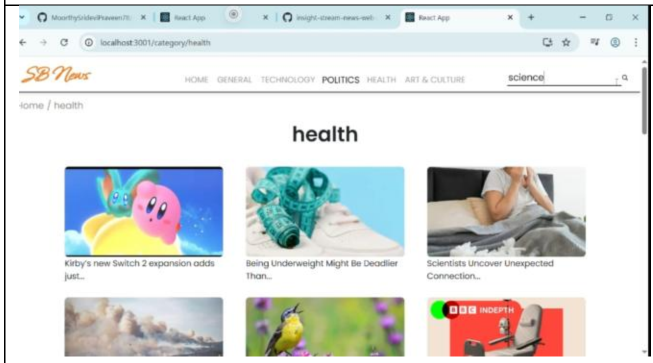
Testing Strategy:
- Unit tests: Jest + React Testing Library for components
- Integration tests: React Testing Library for page flows
- End-to-end: Cypress (or Playwright) for critical user journeys (view article, search, bookmark)

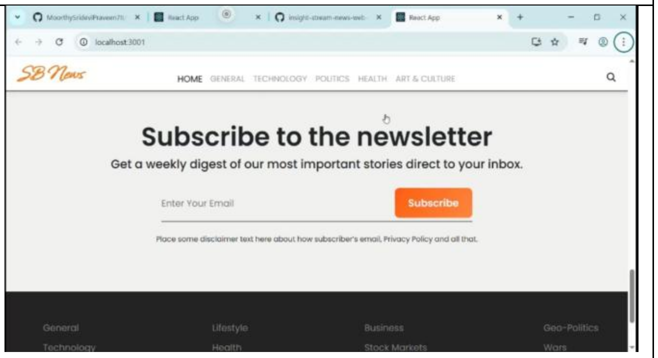
Code Coverage:
- Use coverage reports from Jest and enforce thresholds in CI (e.g., 80% lines, 75% functions)

12. Screen shorts


localhost:3001
SB News
HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE
WIRED Tests Dozens of Air Purifiers a Year. Here's What We Look For (2025)
Dwayne Johnson Will Play the Chicken Man in 'Lizard Music'
Apple May Take AirPods Pro 3 Beyond Audio
'KPop Demon Hunters' Is Now Officially Netflix's Most Popular
Watch: Moment avalanche is triggered by explosives in New


localhost:3001/category/politics
SB News
HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE
Search
Home / politics
politics
The War on Trans Art...
How the American Political Landscape Cha...
Anger in Italy over non-consensual photo...


localhost:3001/category/health
SB News
HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE
science
Home / health
health
Kirby's new Switch 2 expansion adds just...
Being Underweight Might Be Deadlier Than...
Scientists Uncover Unexpected Connection...


localhost:3001
SB News
HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE
Subscribe to the newsletter
Get a weekly digest of our most important stories direct to your inbox.
Enter Your Email
Subscribe
Place some disclaimer text here about how subscriber's email, Privacy Policy and all that.
General
Technology
Lifestyle
Health
Business
Stock Markets
Geo-Politics
Wars

12. Known Issues

- Streaming provider may have CORS restrictions in some environments.
- Some older browsers may require polyfills for modern JS features.
- Rate-limiting from third-party news APIs could affect feed freshness.

13. Future Enhancements

- Personalization: ML-driven recommended articles
- Offline reading: service worker + local cache
- Multi-lingual support
- Improved analytics and A/B testing for feed ranking

Appendix: Useful Commands & Links

- Start dev server: npm start
- Build: npm run build
- Run tests: npm test