**InsightStream: Navigate the news language**

Team Members:

Koliparthi Hema lakshmi

Pooja .k

Abinaya .c

Sunandha .u.k

**Project Overview**

**Purpose:**

The primary purpose of the “Insight Stream” news app is to provide users with a fast, reliable, and personalized way to stay updated with the latest news from around the world

**Features:**

* News Aggregation
* Personalized News Feed
* Search & Filters
* User Authentication & Profile
* Bookmark & Read Later
* Dark Mode & UI Customization

**Architecture**

**Component Structure:**

**Real-time News Aggregation –** Successfullyfetches news from multiple sources via News API / RSS Feeds.

**Personalized User Experience –** Allows users to customize their news feed based on topics of interest**.**

**Search & Filtering –** Enables quick access to relevant news with category, date, and source filters**.**

**User Authentication –** Implements Google/Firebase Authentication for secure login & profile management.

**Bookmark & Read Later –** Provides users with a save feature for offline reading.

**Dark Mode & UI Customization –** Supports theme switching for better readability**.**

**Push Notifications –** Sends breaking news alerts based on user preferences.

**Smooth Performance & Deployment –** Hosted on Firebase/Render/Vercel, ensuring fast and scalable access.

**SETUP INSTRUCTIONS**

**Prerequisites:**

        Ensure you have the following installed:

Node.js (for backend) → Download

React Native CLI / Flutter (for frontend)

MongoDB / Firebase (for database)

News API Key → Get it here

Git (for version control)

**Test the App:**

Open the app on your device or emulator.

Sign up, browse news, search, bookmark, and test notifications.

**Folder Structure**

**Frontend (frontend/):**

Components/ – Reusable UI elements like buttons & cards.

Screens/ – Different screens/pages of the app.

Navigation/ – Manages screen-to-screen transitions.

Context/ – Global state (Redux/Context API).

Assets/ – Images, icons, fonts.

Styles/ – Global styles for UI design

**Running the Application**

**Prerequisites:**

Make sure you have installed:

1. Node.js & npm → Download
2. React Native CLI / Expo CLI (for React Native apps)

Npm install -g react-native-cli

Npm install -g expo-cli

3.MongoDB (if using local DB) → Download

4. Firebase CLI (if using Firebase Firestore)

Npm install -g firebase-tools

**Run the Frontend (React Native):**

1. Install Dependencies

Cd ../frontend

Npm install  # For React Native

1. Configure API Endpoint

In config.js or .env file (frontend):

API\_BASE\_URL=http://localhost:5000

FIREBASE\_API\_KEY=your\_firebase\_api\_key

1. Start the App

For Android Emulator

Npx react-native run-android

For iOS Simulator (Mac only)

Npx react-native run-ios

For Expo (Alternative Method)

Expo start

Scan the QR code in the Expo Go app (Android/iOS) to view the app on your phone.

**Component Documentation**

**1.Key Components:**

* RecipeCard.js – Displays a summary of a recipe

* RecipeDetails.js – Shows detailed recipe information

* IngredientList.js – Manages shopping lists

**2.Reusable Components:**

* Button.js – Customizable button component

* Modal.js – Reusable modal for alerts and pop-ups

**3.HomeScreen.js (Main News Feed):**

* Location: src/screens/HomeScreen.js

* Purpose: Fetches and displays latest news using NewsCard.js.

**State Management**

State Management in “Insight Stream”

In the frontend (React Native), we manage state using either:

**1.Context API (Simple & lightweight) –** Used to store news articles and avoid prop drilling.

**2.Redux Toolkit (For complex apps) –** Useful for managing user authentication, bookmarks, and large datasets efficiently.

**User Interface**

**1.Home Screen (News Feed):**

Displays a list of news articles using FlatList. Uses NewsCard.js for each article preview. Infinite Scroll for a seamless news feed experience.

**2.News Card (Reusable UI Component)**

Displays image, title, and description of each news article. Clicking on a news card navigates to the NewsDetailScreen.

**Example UI Code:**

import { Route, Routes } from 'react-router-dom';

import './App.css';

import Home from './pages/Home';

import CategoryPage from './pages/CategoryPage';

import NewsPage from './pages/NewsPage';

import NavbarComponent from './components/NavbarComponent';

import Footer from './components/Footer';

function App() {

  return (

    <div className="App">

        <NavbarComponent />

        <Routes>

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

          <Route exact path="/category/:id" element={<CategoryPage/>} />

          <Route exact path="/news/:id" element={<NewsPage/>} />

        </Routes>

        <Footer />

    </div>

  );

}

export default App;

**STYLING**

* **Global Styles (Common Styles for Reusability)**

 @import 'bootstrap/dist/css/bootstrap.min.css';

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700;800;900&family=Whisper&display=swap');

body {

  margin: 0;

  font-family: 'Poppins', sans-serif;

  -webkit-font-smoothing: antialiased;

  -moz-osx-font-smoothing: grayscale;

  -ms-overflow-style: none;  /\* IE and Edge \*/

  scrollbar-width: none;

}

body::-webkit-scrollbar {

  display: none;

}

code {

  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

    monospace;

}

.spinners{

  display: flex;

  justify-content: center;

  align-items: center;

  gap: 10px;

}

**Home Screen Styling:**

     .Home{

    padding: 15vh 0 0 0;

    -ms-overflow-style: none;  /\* IE and Edge \*/

    scrollbar-width: none;

}

.Home::-webkit-scrollbar {

    display: none;

  }

.home-components{

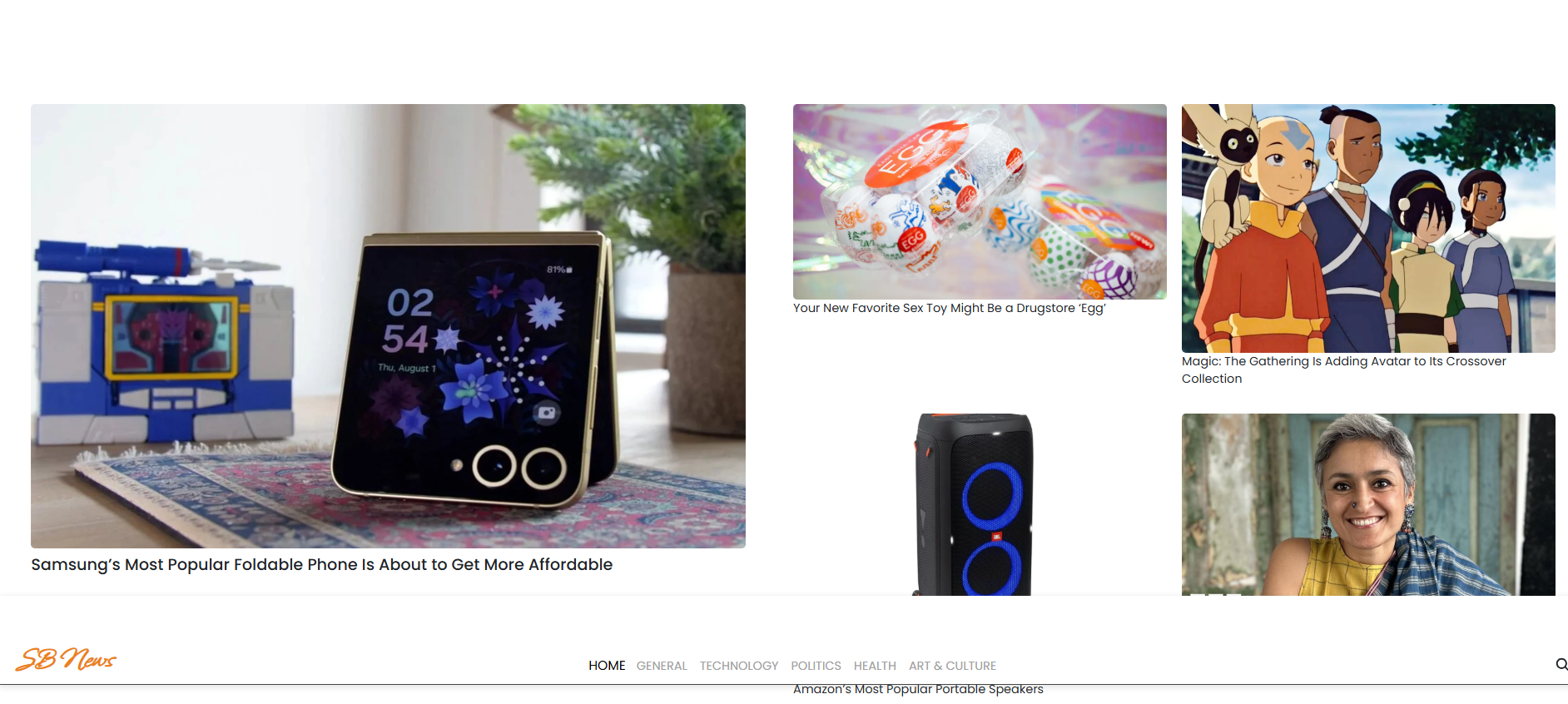
    display: grid;

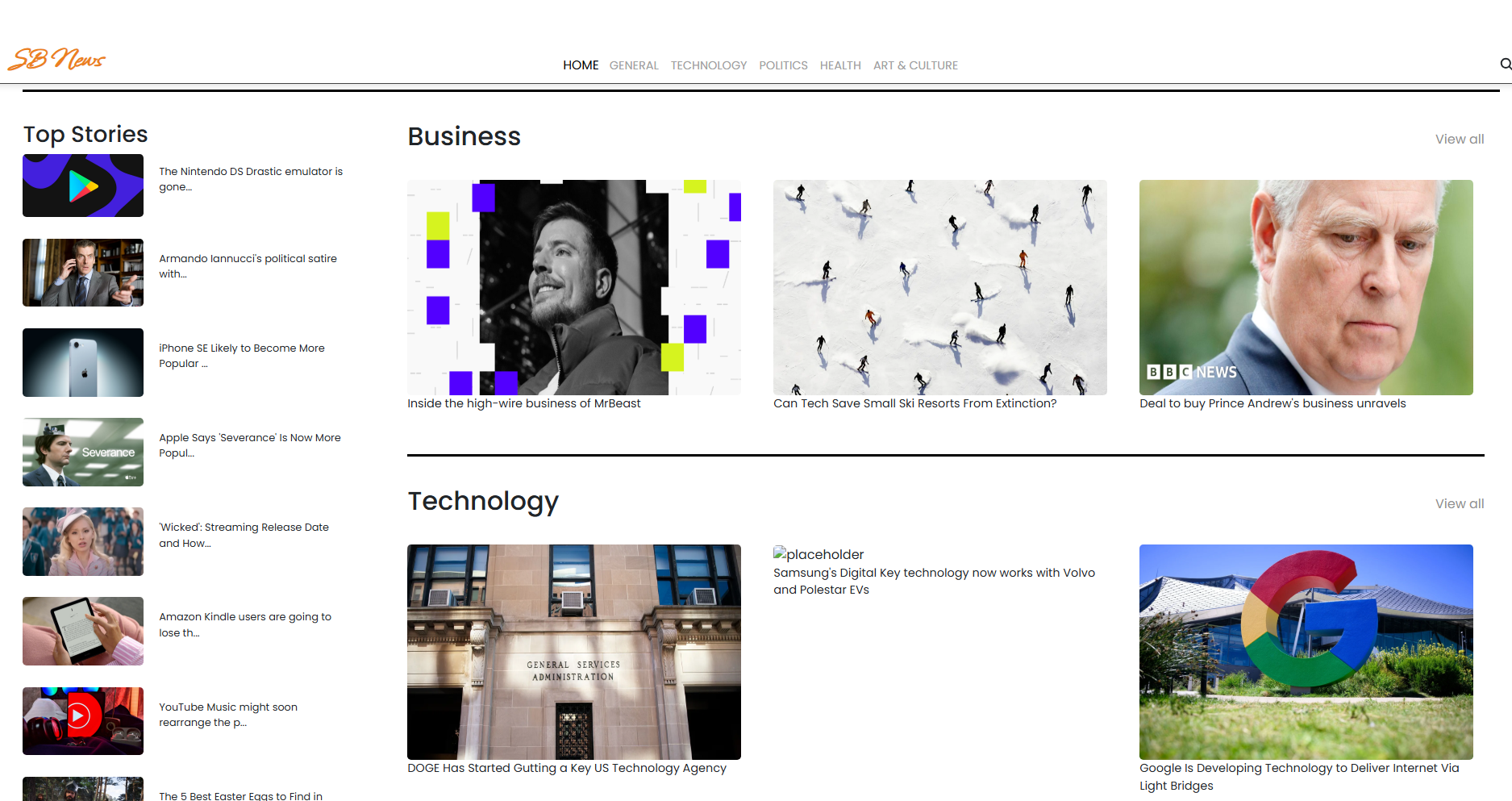
    grid-template-columns: 23% 70%;

    gap: 4%;

}

**SCREENSHOTS**





**KNOWN ISSUES**

* **Slow API Response –** The app may take time to fetch news if the API is slow.

* **Dark Mode Not Fully Implemented** – Some UI components don’t switch themes correctly

**FUTURE ENHANCEMENTS**

* **Offline Mode** – Allow users to read saved news without internet using local storage.
* **Category-Based News** – Add filters for sports, politics, technology, entertainment, etc.