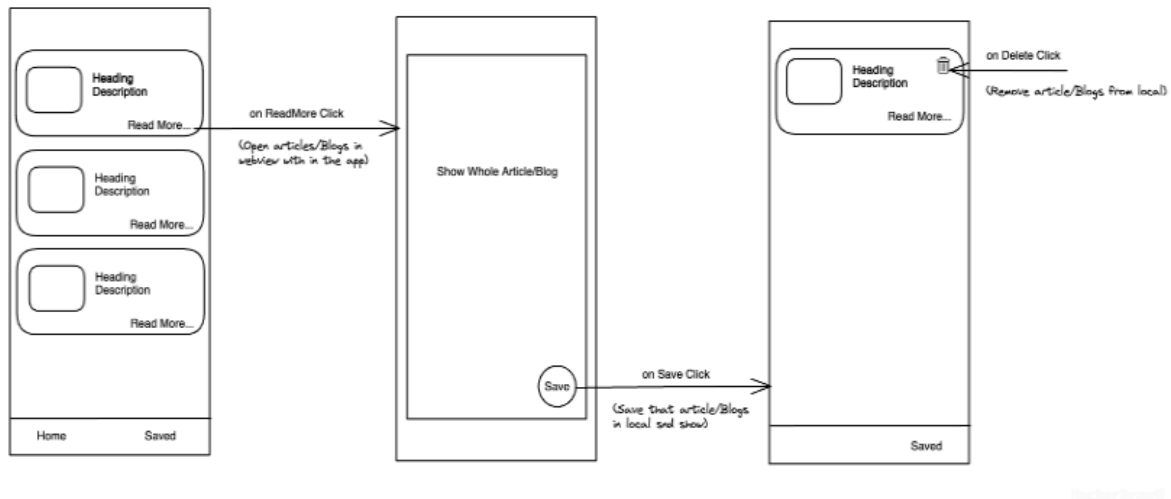Assignment Description

Create a fully functional iOS application that fetches data from the NewsAPI, displays it in a user-friendly manner, and allows users to save articles locally for offline access. The application should adhere to iOS best practices and be built entirely using SwiftUI.

Assignment Requirements

1. API Integration

•   Use NewsAPI to fetch the latest news articles.
•   Display the articles, including headings, descriptions, and images, in a scrollable List or LazyVStack.

2. Article Details

•   When a user clicks on a news item ("Read More"), open the full article in an in-app WebView (using WebKit in SwiftUI).
•   Provide a "Save" button in the WebView to store the article locally.

3. Local Database

•   Implement a local database using Core Data or Realm to save articles.
•   Provide a "Saved Articles" section where users can view their saved articles.
•   Allow users to delete saved articles.

4. UI/UX

•   Follow the UI outline provided or design a more engaging SwiftUI-based interface.

• 	Use SwiftUI's modern UI components (List, NavigationStack, AsyncImage, etc.) to enhance the user experience.

Bonus Points

• 	Make the app offline-first, ensuring it works seamlessly without an internet connection.
• 	Add extra features like:
• 	Dark Mode support.
• 	Article categorization (e.g., filtering by category like Business, Sports, Tech).
• 	Share button to share articles via iOS ShareSheet.
• 	Maintain a clean, modular codebase using best Swift practices.

Documentation and GitHub

• 	Upload the project to GitHub, maintaining proper version control.
• 	Include a detailed README.md file with:
• 	Project description.
• 	Steps to set up the project.
• 	Libraries used and their purpose.
• 	Explanation of the architecture and design choices.
• 	Use branches to track progress and demonstrate good project management practices.

Suggested Tech Stack

• 	SwiftUI for UI.
• 	Combine for reactive programming and API handling.
• 	URLSession or Alamofire for networking.
• 	Core Data or Swift Data for local storage.
• 	WebKit (WKWebView) for in-app article reading.