**Genesys Code Challenge**

A frontend application built with Angular to display the top and latest news from the Hacker News API.

**Features**

**Display Top News**: Fetches and displays the top 500 news stories.

**Display Latest News**: Fetches and displays the latest news stories.

**Pagination**: Implemented a "Load More" button to display articles in batches of 10 for better performance.

**Dynamic UI**: Visually distinguishes between different news types (e.g., stories, jobs) to improve readability.

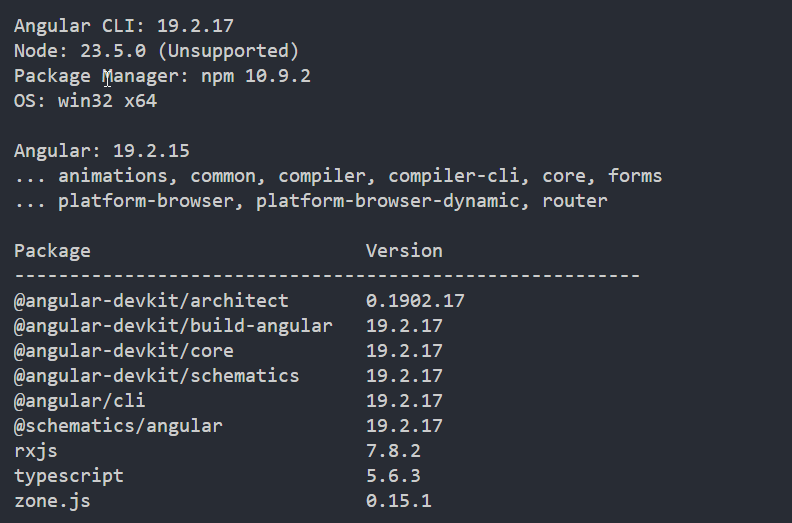
More details about the code challenge are available in the document - **Frontend Coding Challenge - Hacker News Updated 2024.pdf**

**Technologies Used**

* Frontend : Angular 19 , TypeScript , SCSS/CSS, FontAwsome
* Backend : Hacker News API (provided) (https://github.com/HackerNews/API )

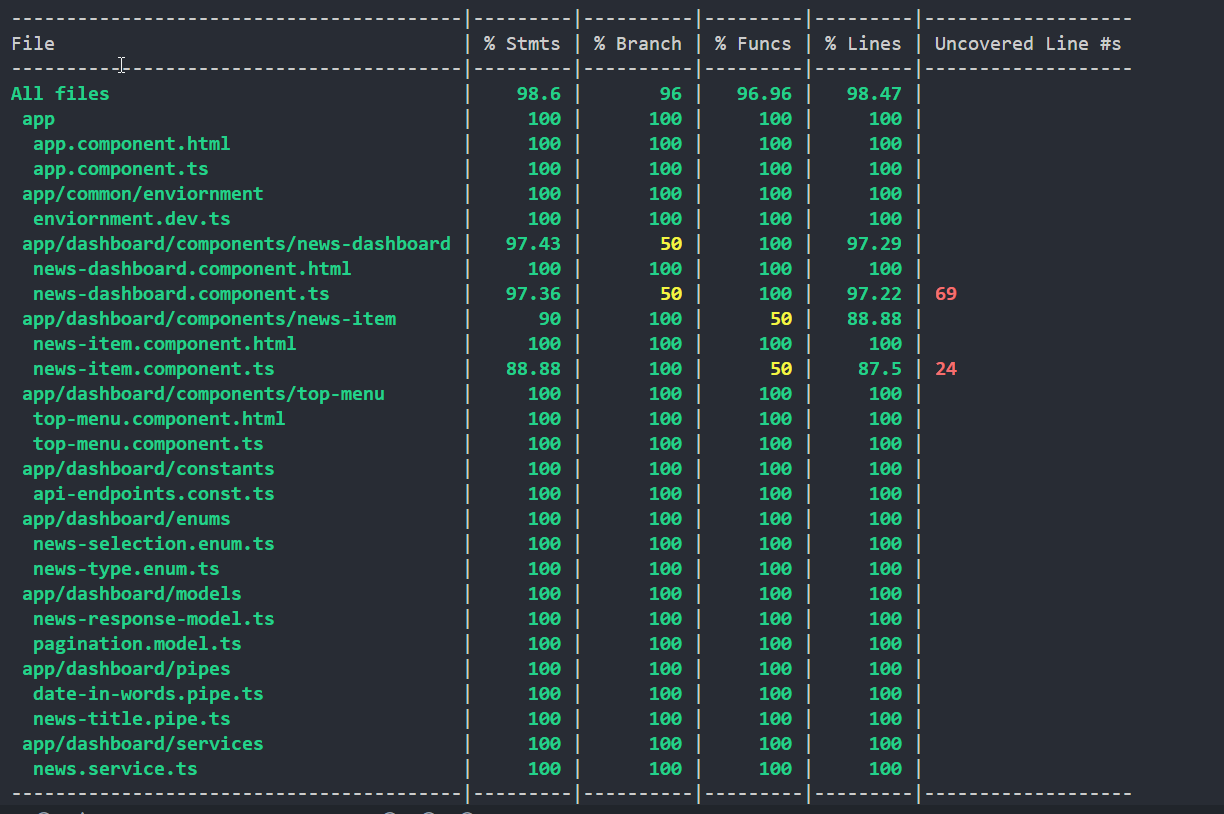
**Prerequisites**

Make sure the following tools, with specific versions or later, are installed on your machine.



**Installation and Setup**

* Clone the repository - git clone [makarandvichare2/genesysNewsTest](https://github.com/makarandvichare2/genesysNewsTest)
* Navigate to the project directory - cd your-repository
* Install the dependencies - npm install
* Run the application - npm start
* Run the unit tests - npm run test



* Run the cypress tests(e2e) - npm run e2e
* Run the storybook tests - npm run storybook

**Development assumptions**

* **Pagination** -
  + The provided API returns all top 500 news items in a single request. Client-side pagination was implemented to handle subsequent API calls for individual news details and improve application performance. The page size is set to 10 news items per page.
* **News Entity** - The “score” property from the API was assumed to correspond to the “points” property on the existing Hacker News UI.
* **Scope -** Only the "Top News" and "Latest News" features were implemented; other features were excluded from this challenge.
* **API Response** – The API is expected to return the top and new stories in sorted order, so no additional sorting logic is needed on the client side.
* **Local Memory** – Technologies like NgRx or local storage are not used since RxJS caching is sufficient for this purpose.

**Design and Architecture**

**Sequence diagram for the load more news feature -**

