

Application Task

In this task you would have to build a simple News Website

You should have 4 pages - Login, Main, Story and Profile views.

1. Login view - Initial view when the application loads. There should be a form with 2 fields:
 - a. Email - email input
 - b. API token - token that would be used for the [API request](https://newsapi.org/docs/authentication) (<https://newsapi.org/docs/authentication>)
 - c. Prompt the user how to get an API token from [here](https://newsapi.org/register) (<https://newsapi.org/register>)Once the user has entered both of them, he/she should be able to submit the login form, the data should be stored and later on the API token should be used for fetching news from the news api. There is no need for validation of those fields, but the user should not be able to go to any other view unless he has populated them and submitted their values.
2. Main view - The main view should be rendered once the user has logged in. It should display a list of top news headlines and a search input field.
The list should show brief info (ex. title, description, etc.) about each story.
When I enter an input and submit (by pressing 'Enter' or pressing a submit button), I should see the result of my search in the same list view.
I should be able to open an article in a Story view or go to Profile view.
3. Story view - When I click on an article(news story), I should see all the information that the story provides in a nice and persistent way. I should be able to go back to the Main view or Profile view.
4. Profile view - The profile view should show few fields:
 - a. Name - user's name
 - b. Email - user's email - pre populated by the user's input on login.
 - c. API token - token that would be used for the [API request](https://newsapi.org/docs/authentication) (<https://newsapi.org/docs/authentication>) - pre populated by the user's input on login.The user should be able to update the values of the fields and they should be stored in a way that their values would not be lost in case the user refreshes the page. Simulate API calls during the load of the Profile view and the updating of its fields.

5. Navigation bar - The navigation bar should be always visible, but the user should only be able to navigate through the pages once he's logged in. The user should be able to log out if he/she is logged in.

Resources and suggestions:

- News api: <https://newsapi.org> or any other free API that you might find suitable. In case you choose another API, explain in the Readme.md how one can get an Authentication token to test the application.
- It is recommended to use [react-router](https://www.npmjs.com/package/react-router-dom) (<https://www.npmjs.com/package/react-router-dom>) for navigating between pages but feel free to use your own approach.
- Make sure you handle errors and loading times with a friendly UI.
- The above points are just the basic scaffolding. Feel free to think out of the box and improve the implementation in any way you find suitable.
- If you have the time, it would be nice to see some unit tests as well.
- If you struggle with some part, try to do your best and commit anyway.

Requirements:

1. Give a time estimation for your work.
2. Using typescript is preferable but javascript(ECMAScrip) is ok as well.
3. Use the React framework.
4. You can create your custom styles, use css/scss/sass framework or use a React UI framework such as React-bootstrap, Material-UI or Antd. (Using a framework is not considered a minus.)
5. Use Redux library for state management.
6. Add instructions on how to download and run your application in Readme.md.
7. Post the finished version of your task in github.com.