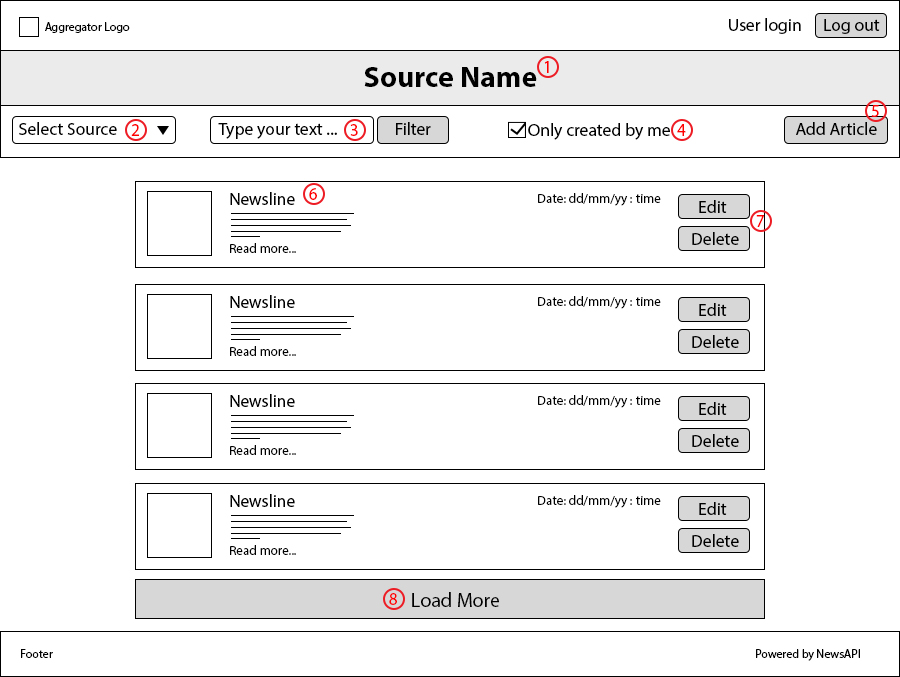
**Approximate Mockup for your application, description and Homeworks.**

The idea behind this, is that you are making sort of news aggregator (portal), based on a NewsAPI (the one form the first task). But also you can add and edit your own, local news. It can be something like ‘Local’ and ‘Worldwide’ news.

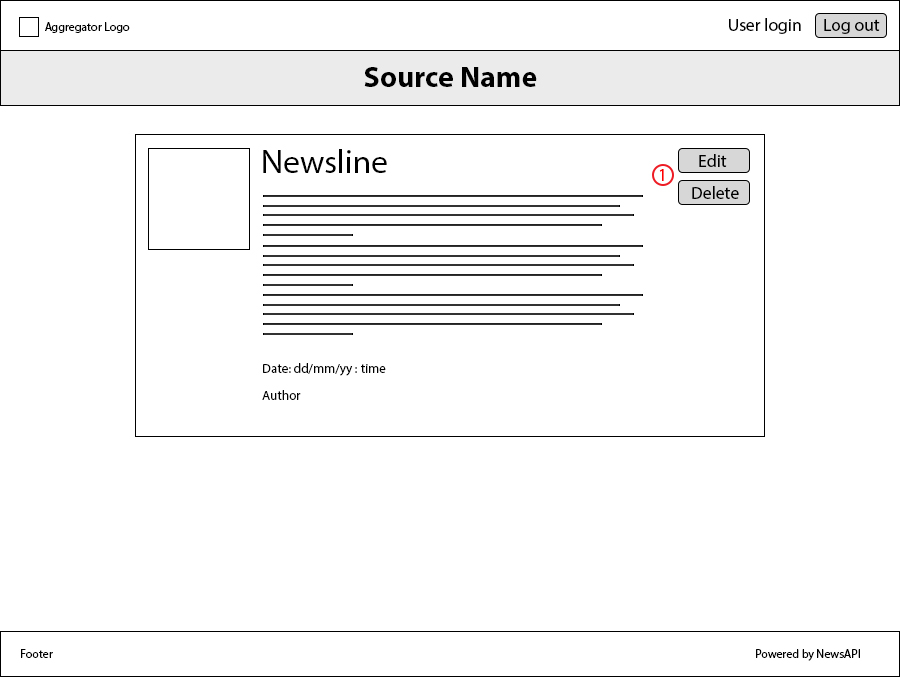
You can choose any design for the template, change a structure as you like, but try to stick to the following points:

1. Make a source name, e.g BBC as a heading, after you choose the one from the list (2). If the source is your own database, hide source name or display the name you choose for your portal.
2. List of sources from NewsAPI, ad in first task. After you choose one or another from dropdown, the news-articles list updates according to the source. Source Name also updates. It would be a plus if you will add an opton like ‘all sources’, when all news from all the sources will be shown without selecting any specific.
3. Filter: type a few words, and list of news should be filtered by keywords
4. Created by me – when this flag is checked, you see only those news, that was created by you. Dropdown is disabled in that case.
5. Add article – redirects you to the creation page (3rd image)
6. Newsline. This is shorten news-article. Contains an image and short description. The news created by you will have edit and delete controls, otherwise not. After a click on it – redirects to the appropriate page of an article, with full text. See 2nd picture.
7. Edit / Delete – shown only for ‘your’ articles. Do not display it for articles from NewsAPI
8. Load more. Initially you see only limited number of articles, e.g 5. After a click on this button it should load 5 more articles.



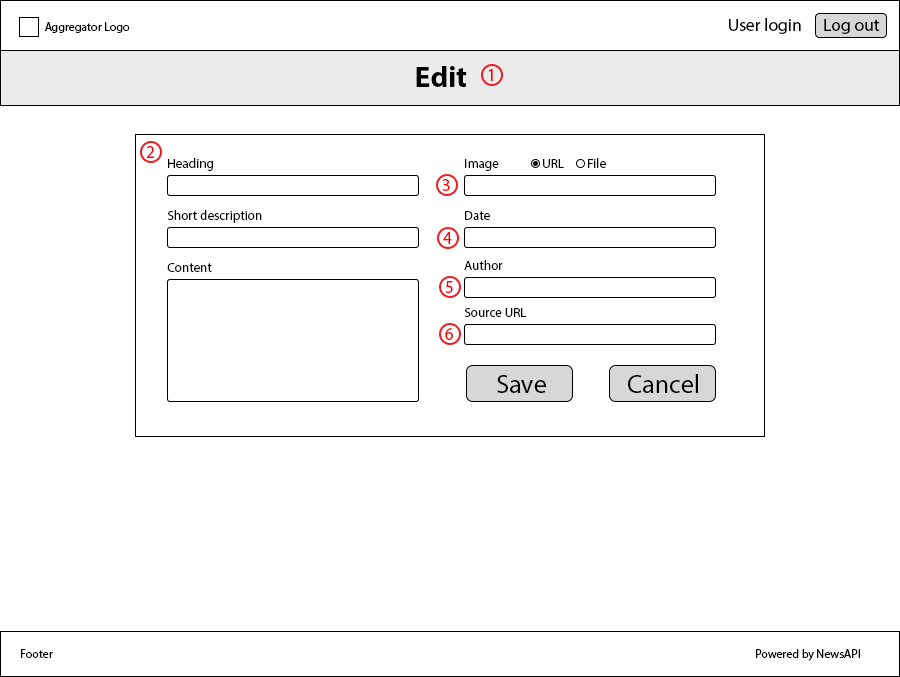
This is a specific page for news-article

1. Edit \ delete are shown only for articles created by you



This is Create/Edit page, which is working with your API, that you’ve done on the Node/Express topics.

1. If it’s for a new article creation – name is Create, for update of an existing - Edit. In ‘Edit’ case - all fileds should be prepoulated.
2. Heading and Content are required fields (because without them there are nothing to display)
3. If URL is chosen – it should take an url for the image, if it’s a file – use your backend endpoint for uploading file. Also? It's not on the mock-up, but **show a small preview-picture after it’s uploaded** one or another way.
4. Date is prepolulated whith current date when create OR set automatically based on the time when article is saved.
5. Author may be prepoluated by the one who is Logged in.
6. Here you can set the URL name, that will be used for routing. E.g Heading is ‘Upcoming Minsk Events’, the url will be ‘upcoming-minsk-events’. The field is also mandatory, otherwise you will have to generate url for your article.



Homework for Part1:

1. Create a project with Angular CLI, it should run on localhost by ng serve.
2. Create most of the components according to the mockup, make some template styling.
3. Add mock data, one for source list, one for the newslist (just and array of nock objects, representing news). Display it on page using directives \*ngFor, component interactions (@Input for passing data to a news-item, and @Output for passing event from the ‘Load More’ button.
4. Add routing. It should be possible to go to the specific article page, to ‘edit’ and to the ‘create’ pages.
5. Done all of the above: all components, all routes, all buttons. No need to bind to any real API –for example ‘Save’ button may just use console.log(‘Saved’).

Homework for Part2:

1. Re-write cross-component communication to services, where it is necessary.
2. Add default Angular pipes (e.g date pipe) where available. Add filtering by your custom @Pipe.
3. Get list of sources from News API, get list of news based on selected source. Add type that GET method will return (this.http.get<type>(‘url...’)), create interfaces for that. At least create an interface for Article. Make a request to get full data when view the whole Article.
4. Create a Form and it’s model (Reactive approach) for Create / Update page. Make required fileds obligatory. Without them form should not be valid, and the button should be disabled (<button [disabled]=”form.invalid”>Save</button> - something like that). Prepopulate form fileds if it’s Update mode.
5. Everything works, Form allows to create and update your ‘local’ news, you can delete them. Add any other features you consider useful for your app.
6. **Bonus**: Try to deploy FE on any platform (Heroku, GitHub-pages etc.), and ensure it works from other device/computer.