



Frontend coding test

Elevio is a SaaS platform that makes it easy to create, deliver and improve help content to drive customer education.

In this test you will build a page that has a search UI that calls our REST API to find relevant articles and display the result to the user.

The world of frontend development is huge and this is a chance to show off where you shine, so don't feel you have to polish all aspects of this test.

Task

Build an interface that allows users to search for and display articles from Elevio

Functional requirements

- On enter or by pressing the submit button, the current text entered in the search input should be submitted to our REST API to fetch a list of matching articles
- To be submitted there must be a minimum of **three** characters in the search input
- The results of the search should be shown on the page; how this happens is up to you.
- On clicking of the search result, the user will be navigated to a page to display the article (don't create the page to display the article but put in a mechanism to forward the user to a new page)
- Also, we would like to track both the search and the clicking of the search result, put in place a mechanism to send events off to an API
- Submit the code and documentation via GitHub (don't squash your commits)

Non-functional requirements

- Use the Elevio REST API to fetch a list of matching articles
- Handle loading, error and success states
- Allow the user to clear the input by pressing some sort of clear button
- Deal with limiting the number of results displayed
- Ensure the code is testable and maintainable

Bonus points

Don't feel you have to cover all of these bases, just showcase your skills and interests.

If you would like to showcase you UI/UX skills, spend time doing that.

If you would like to show your engineering skills, spend time there.

- Accessibility
- Testing
- Functional programming
- Design and UX so the user feels the component is polished and they can easily understand how to interact with it. Feel free to show off your ability to encapsulate styling so it's easy to work this (BEM, Less, Sass, Tailwind, Styled-Components... etc)
- Typescript
- Sweet dev environment setup
- Usage of iconography and animations
- Deployment

Getting started

- Let us know that you accept the challenge so we can create a test account for you

- Create some dummy content: <https://app.elevio-staging.com/content/>
- **NOTE** Make sure you publish the articles or the endpoint will throw a 500 error
- Create an API key: <https://app.elevio-staging.com/apikeys>
- Refer to the REST API documentation: <https://api-docs.elevio.help/en/articles/73-rest-api-introduction>
 - NOTE: Instead of using api.elev.io as the API domain, please use api.elevio-staging.com. Everything else should stay the same.

If you have any questions, feel free to reach out to me at careers@elev.io