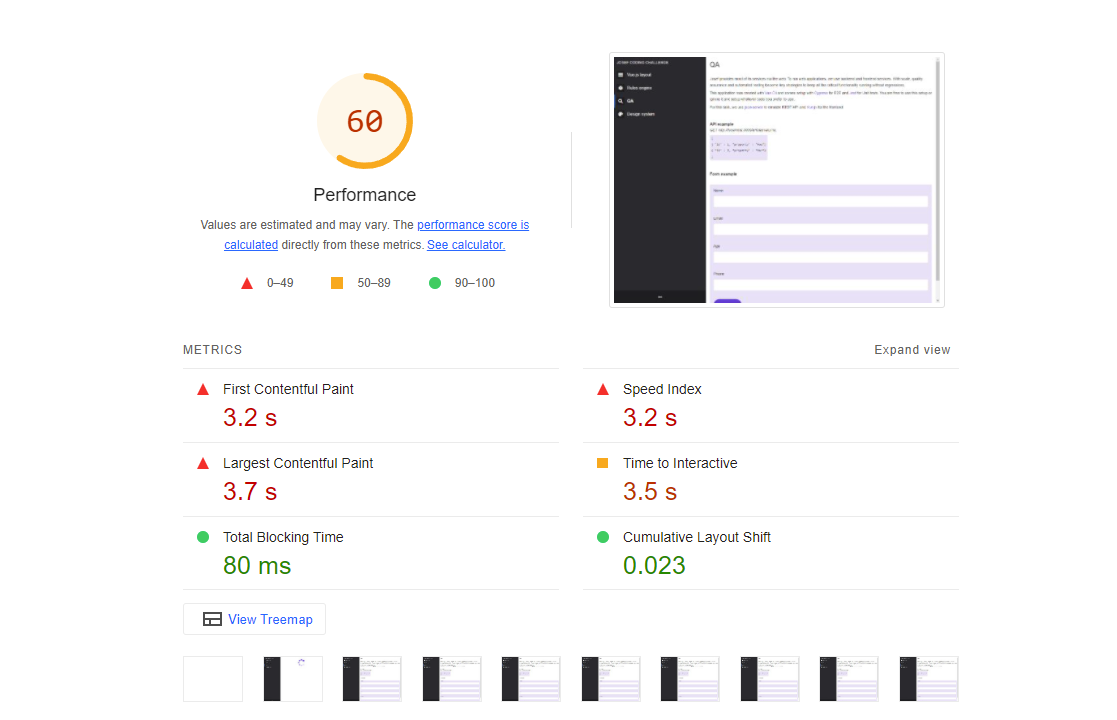
**Lighthouse Audit**

1. **Performance**



As far as performance is concerned, the audit generates a metric score of 60 out of 100. The score indicates that the site needs improvement. Some of the metrics that are in good shape are: Total Blocking Time (Sum of all time periods between FCP and Time to Interactive) and Cumulative Layout Shift (Cumulative Layout Shift measures the movement of visible elements within the viewport.) Time to Interactive needs improvement, while First Contentful Paint, Largest Contentful Paint, Speed Index are in poor shape. These metrics are related to how quickly the contents, first text or image, largest text or image are visible.

Here are some opportunities to improve:

* Enable Text Compression to minimize network bytes:

| URL | Transfer Size | Potential Savings |
| --- | --- | --- |
| [/js/chunk-vendors.js](http://localhost:9001/js/chunk-vendors.js)  (localhost) | 2,937.1 KiB | 2,197.3 KiB |
| [/js/app.js](http://localhost:9001/js/app.js)  (localhost) | 629.2 KiB | 542.8 KiB |
|  |  |  |
|  |  |  |

* Serve static assets with an efficient cache policy: A long cache lifetime can speed up repeat visits to your page.

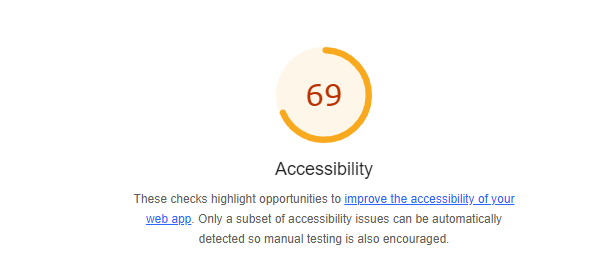
| URL | Cache TTL | Transfer Size |
| --- | --- | --- |
| [/js/chunk-vendors.js](http://localhost:9001/js/chunk-vendors.js)  (localhost) | None | 2,937 KiB |
| [/js/app.js](http://localhost:9001/js/app.js)  (localhost) | None | 629 KiB |
|  |  |  |

* Avoid enormous network payloads: Total size was 3,767 KiB

In regards to the passed performance audits, we can list the following:

* Eliminate render-blocking resources
* Properly size images
* Defer offscreen images
* Reduce unused CSS
* Reduce unused JavaScript
* Efficiently encode images
* Avoid multiple page redirects
* Use HTTP/2
* Minimize third-party usage
* Use passive listeners to improve performance scrolling
* Avoids document.write()
* All text remains visible during Webfont loads
* JavaScript execution time
* Avoids excessive DOM size
* Initial server response time was short

1. **Accessibility**



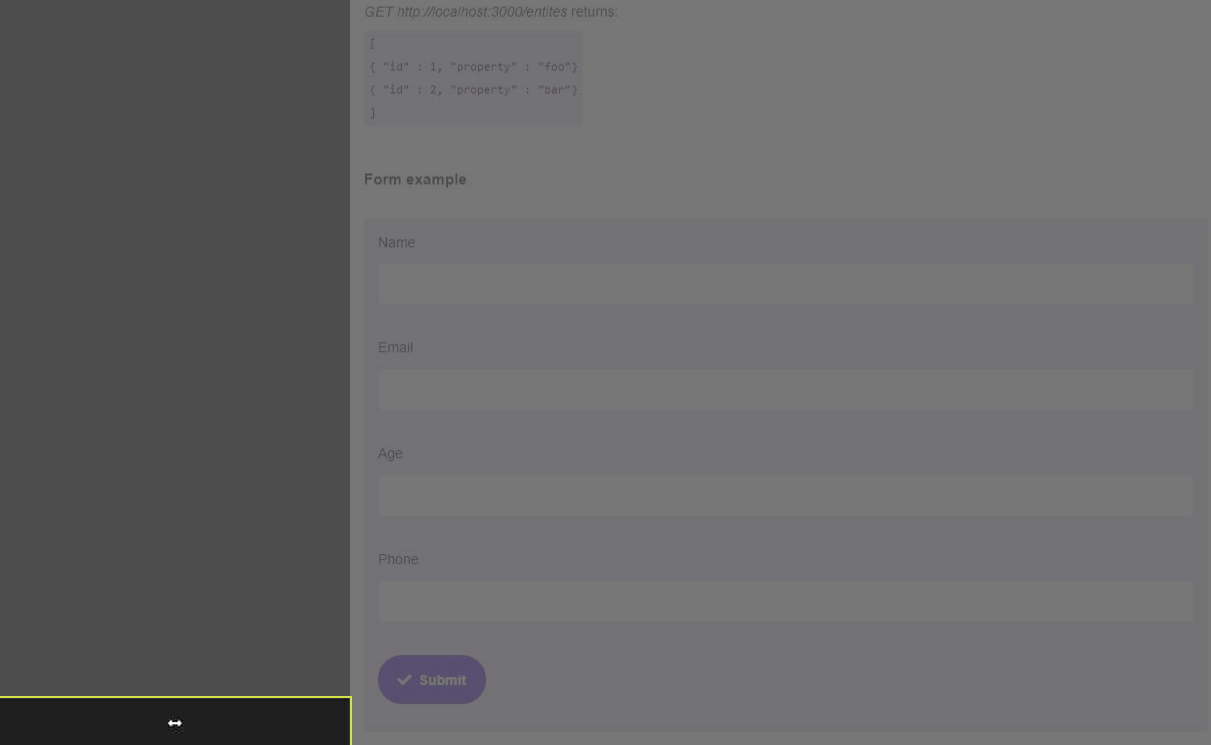
Accessibility metrics show a score of 69 out of 100, topping the performance metrics. However, there are still aspects that could be improved, especially for users with disabilities who rely on screen readers, like:

**NAMES AND LABELS**

* Buttons do not have an accessible name: When a button doesn't have an accessible name, screen readers announce it as a "button", making it unusable for users who rely on screen readers

div#app > div.sidebar-container > div.v-sidebar-menu > button.vsm--toggle-btn

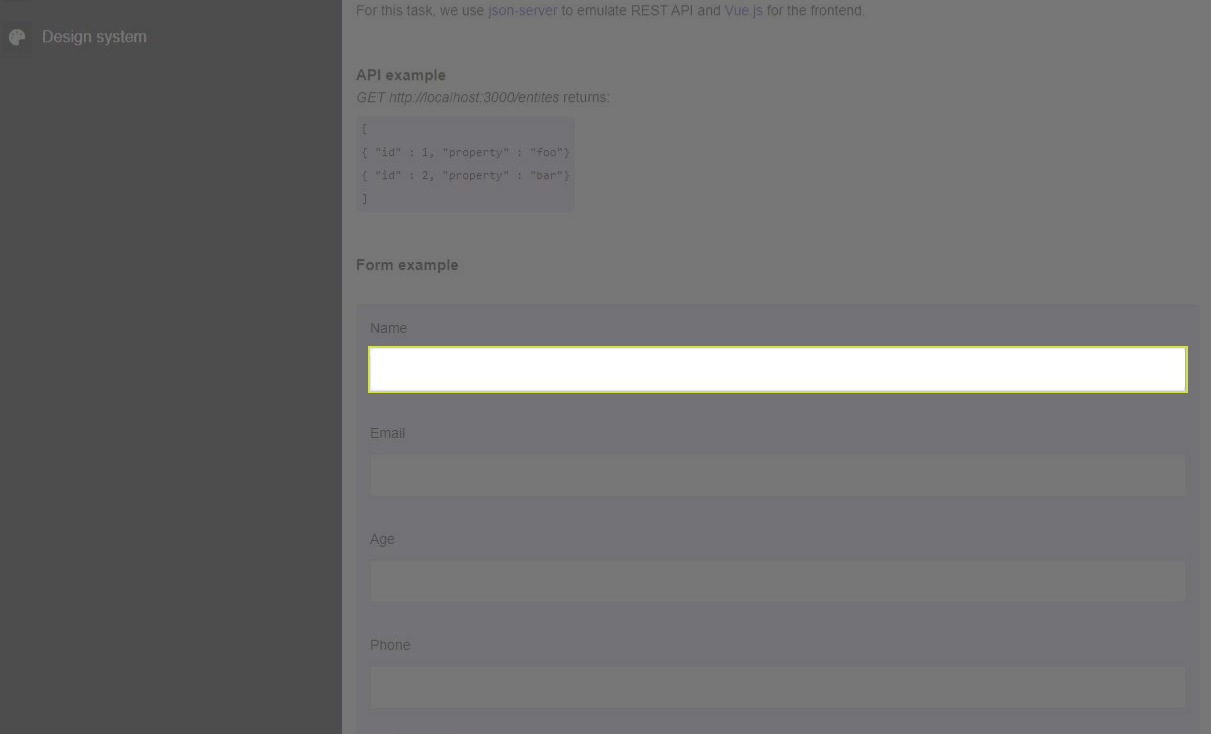
<button class="vsm--toggle-btn">



* Form elements do not have associated labels: Labels ensure that form controls are announced properly by assistive technologies, like screen readers.

form.user-data > div.input-field > div.input-field\_\_wrapper > input.input-field\_\_input

<input data-v-9f62a1b8="" name="name" autocomplete="on" required="required" type="text" class="input-field\_\_input">



**INTERNATIONALIZATION AND LOCALIZATION**

* <html> element does not have an [lang] attribute: If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly.

<html lang="">

**TABLES AND LISTS**

* Lists do not contain only <li> elements and script supporting elements (<script> and <template>): Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output.

[ { "id" : 1, "property" : "foo"} { "id" : 2, "property" : "bar"} ] - <ul data-v-7b79ecac="" class="entities">

* List items (<li>) are not contained within <ul> or <ol> parent elements: Screen readers require list items (`<li>`) to be contained within a parent `<ul>` or `<ol>` to be announced properly.

{ "id" : 1, "property" : "foo"} - <li data-v-7b79ecac="">

**NAVIGATION**

* Heading elements are not in a sequentially-descending order: Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies.

API example - <h3 data-v-7b79ecac="">

**PASSED ACCESSIBILITY AUDITS**

* [aria-\*] attributes match their roles
* [aria-hidden="true"] is not present on the document <body>
* [aria-\*] attributes have valid values
* [aria-\*] attributes are valid and not misspelled
* [user-scalable="no"] is not used in the <meta name="viewport"> element and the [maximum-scale] attribute is not less than 5.
* [aria-hidden="true"] elements do not contain focusable descendents
* Background and foreground colors have a sufficient contrast ratio
* Document has a <title> element
* <html> element has a valid value for its [lang] attribute
* Links have a discernible name