

Frontend test assessment

Frame 2

Share **Board name** with others

john@miro.com × invalid.email × mike@miro.com ×
alexander@miro.com × add more people...

Add email Get emails count

Please create a form based on the screenshot above.

Input field with emails (let's call it **emails-input**) should be implemented as a separate js-library, so it can be used in any other form or app independently.

An example of **emails-input.js** usage:

```
<div id="emails-input"></div>  
<script src="emails-input.js"></script>  
<script>  
  var inputContainerNode = document.querySelector('#emails-input');  
  var emailsInput = EmailsInput(inputContainerNode, ...);  
</script>
```

Assessment requirements:

- A link to a demo page with implemented form should be provided. You may host it on [GitHub Pages](#).
- A link to the public git repository with source code and built version should be provided as well.
- Git repository should contain a README that describes how to use that component. Examples + documentation.
- Demo page should work in IE 11 and the latest versions of Chrome, Safari, Firefox, Edge.

Functional requirements:

- EmailsEditor component can be possible to use in any other form or app independently.
- Email block should be created by pressing Enter, entering comma, or by losing focus on the input field. A Block can be deleted.
- Input width must depend on the parent container's width and height. If parent width changes, emails should be redistributed by rows.
- Other than that input should neither depend on the form or page styles, nor conflict with them.
- If input has too many emails, user should be able to scroll it.
- Pasted emails should be converted into blocks immediately. If multiple comma-separated emails are pasted (e.g., "ivan@mail.ru, max@mail.ru"), they should be converted into multiple blocks.
- **"Add email"** button adds a random email to the list.
- **"Get emails count"** button shows an alert with valid emails count.
- Do NOT implement editing of added emails.
- It should be possible to create several emails editors on the same page.
- **emails-input** should have no external dependencies like React, Lodash or any polyfills. Usage of TypeScript, Less, Webpack, etc. is for your consideration.
- Tests are not mandatory, but having them is a plus.
- Performance isn't a big concern, but there should be no major flaws, such as memory leaks or re-rendering all email blocks every time you add or remove a single email.

Design:

<https://www.figma.com/file/CWdAs3rN4d2gZpnoN7ZPvj/Share-test?node-id=0%3A1>

- Log into Figma to see details: font, sizes, colors, offsets, etc.
- Open Sans font can be found [at Google Fonts](#).