

*Thank you for applying for the position of **Front-end Developer** at Printful. We'd be happy to continue the selection process with you and look forward to receiving your take on our technical task below.*

---

## Instructions

1. Before you send us your task or any additional personal information, make sure you've reviewed our [recruitment process privacy policy](#).
  2. Please let us know when we can expect your task (applicants usually send us their tasks in **3 business days**).
  3. Submit all of the code with a README file where you concisely describe the architecture of your chosen solution. If you're creating a database, the tables need to be saved in a MySQL file or .json (MongoDB). Please submit your answers **in English!**
- 

## Task

You need to develop a simple system with a test, where a user can enter their name, choose a test, and see their results.

To develop the system, use Vue.js with ES6 (or newer JavaScript syntax). You can submit it in plain vanilla JS or React, but we highly prefer Vue.js. The design is not limited—you can use Bootstrap, Tailwind, etc., or feel free to be creative.


### **The test consists of 3 different views:**

1. Homepage view: the user inputs their name and chooses one of the tests.
2. Test question view: each question has at least two possible answers (there's no limit to the max number of answers). One of the answers is correct.
3. Results view: the user sees their results.

## Detailed view descriptions

### 1. Homepage view

The user enters their name and chooses one of the tests. If the user has forgotten to input their name or choose the test, they can't access the next page, and an error message appears.



The screenshot shows a web browser window with the title 'Mozilla'. The address bar is empty. The main content area displays the heading 'Technical task' in a large, bold, black font. Below the heading, there is a text input field with the placeholder text 'Enter your name'. Underneath the text field is a dropdown menu with the text 'Choose test' and a downward-pointing arrow. At the bottom of the form is a rectangular button labeled 'Start'.

### 2. Test questions view

The user takes the test. They choose one of the possible answers and go to the next question. The user can't move on to the next question unless they choose an answer. There's a dynamic progress bar that shows the user's progress. It changes depending on which questions the user is on.



### 3. Results view

This view needs to show the name user entered in the Homepage view, and the final results—how many questions there were, and how many of them the user answered correctly.



## Technical requirements

Front-end:

1. You need to use Vue.js with ES6 or newer JavaScript (you can also use vanilla JS or React, but we prefer Vue.js).
2. The design has to be responsive:
  - if the width of the screen is wide enough the answers have to appear in 2 columns (as in the example above)
  - if the width of the screen is not wide enough to show 2 columns (e.g., on the mobile phone), then all answers need to appear below each other)
3. When it comes to sizes, fonts, colors, spaces, etc., feel free to improvise. The visual examples in the previous steps are just schematic sketches of the layout. You need to pick the text color, fonts, background color, button colors, etc. You can use existing CSS frameworks like Bootstrap, Tailwind, etc.

Back-end:

For AJAX requests to the backend, you can use our custom quiz API for accessing quizzes, questions, and requesting correct answers.

You can find documentation (Postman) on how to use our quiz API requests [here](#).

NB: there are no back-end sessions, so you should store the user state in the front-end. You can also implement your back-end, but it isn't mandatory.

1. The tests, questions, and answers need to be stored in the back-end.
2. The data can be hard-coded in PHP files. The data can also be kept in a database, but it's not mandatory.
3. After submitting each answer, a request needs to be sent to the back-end, where it's determined whether the answer is correct or not.

## **What to submit**

Submit all of the code with a README file where you concisely describe the architecture of your chosen solution. If you're creating a database, the tables need to be saved in a MySQL file or .json (MongoDB).

You need to submit an assignment in a GIT repository of your choice (e. g., gitlab.com, github.com) instead of an archived folder (e. g., zip). Please send us a link to your GIT repository. Depending on the repository, you might also need to send us data to access your assignment.

*Thank you and good luck!*