

Project 2. Checklist

You're already familiar with a lot of these requirements. But hang on, there are some new ones, too! You have learned a lot since the last sprint, and our expectations have become higher as well. We are certain you can meet all of them though.

Project rejected without review

The project is considered unfinished if it has at least one of the mistakes listed below. In this case, you'll be asked to fix any issues and to resubmit your project for review. If this happens, it will not count as one of your four attempts.

General:

- Critical mistakes discovered during a previous review (or reviews) have not been fixed.
- There are questions addressed to the reviewers in the project.

Layout mistakes:

- One or several blocks are incomplete, or they're displayed incorrectly in Google Chrome or Firefox.
- Five or more elements are missing.
- One or more sections are missing.
- Blocks are not ordered according to the brief requirements.
- Alignment or CSS layout techniques (flexbox and positioning elements) are not used.
- Displays with a width of `1100px` or more have horizontal scrolling.
- A font is specified incorrectly.
- Any of the animations are missing.

Mistakes in the code structure:

- More than five sections of code have HTML nesting errors.
- More than five sections of code have incorrectly written CSS declarations.

- All styles are written in one CSS file. They are not split by blocks and are not stored in separate folders.

Project requirements

General

- Your project contains:
 - The `index.html` file.
 - The `favicon.ico` file stored in the root directory.
 - A `blocks` directory with BEM blocks stored in separate CSS files (Flat BEM).
 - An `images` directory with all the images, except for `favicon.ico`.
 - A `pages` directory with an `index.css` file stored inside it.
 - A `vendor` directory with a `normalize.css` file stored inside it.
 - A `README.md` file.
- There are no redundant directories and files.
- Meta tags in the head are set according to the brief.
- Fonts are set correctly:
 - The `Inter` and `Noto Serif` font families are used.
 - Fonts are taken from the Google Fonts and connected in the `<head>` via `<link>`.
 - Connected fonts have weights according to the brief.
 - Fallback fonts are connected as alternatives to each of the fonts.
 - There are no extra fonts connected.
 - Font sizes, weights, and colors for each element are the same as in the brief.
- `normalize.css` is imported into `index.css` before other CSS files.
- Matching layout and the brief:
 - The color settings for all links, section backgrounds, and fonts are implemented according to the brief.
 - The margin and padding values of the elements match the brief.

- Blocks, elements, and modifiers are named according to the brief requirements. Blocks are reused when the brief requires it.
- Sections have been ordered in accordance with the brief requirements.
- Images are displayed properly and appear the same as in the brief.
- All animations are made according to the brief requirements. Animations in the `pulsate` block are implemented using the `@keyframes` at-rule.
- The `README.md` file contains:
 1. The project's name.
 2. A description of the project and its functionality.
 3. A description of technologies and techniques used.
 4. Pictures, GIFs, or screenshots that detail the project features.
- There is a `div` element with the `page` class with specified styles and font from the brief.
- There are no typos in the HTML and CSS, and the code is valid. We recommend using the [W3C Markup Validator](#) to check your code for validity.

Semantics

- Content images are inserted through `` and they contain `alt` with a short description of the picture as its value corresponding to the language of the page.
- Navigation links work correctly and lead to the corresponding sections. There are `id` attributes assigned to the target sections and `href` attributes with correct values for the link elements.
- Headings are made using the appropriate tags (from `<h1>` to `<h6>`). Text blocks are made with `<p>` tags.
- The `<header>`, `<main>`, `<footer>`, `<section>`, `<nav>`, ``, and `` elements are present in the code and they are used for their intended purposes.
- `<iframe>` is used to add video content. `iframe` has no invalid, deprecated attributes.
- Elements are not wrapped in `<div>` tags unless required for alignment purposes.

Headings

- The page has only one first-level heading and the hierarchy of headings is consistent. For example, you use `<h1>` for the highest-level heading, `<h2>` for lower-level headings, and so on, without skipping any heading levels.

The BEM Methodology

- Stylesheets do not use tag selectors to set CSS rules.
- No more than two selectors are nested in any CSS rule.
- Stylesheets and images are located in separate folders and organized into blocks. Files are all organized within a Flat BEM file structure.
- Blocks are imported into the corresponding page file. CSS rules for elements and modifiers are declared in the corresponding CSS block file.
- Modifiers do not contain duplicate styles of the element or block being modified. They contain only the changing properties.

CSS

- Images don't overlap any text. Texts are made visible in front of images using the `z-index` property.
- If the margin or padding values are not specified for headings, paragraphs, or lists, they are set to zero by default.
- Absolute positioning is not applied to elements unless explicitly specified in the brief.
- All elements with absolute positioning have values specified for both horizontal and vertical positioning properties (e.g. using the `top` and `left` properties).
- For elements with absolute positioning, `top` and `bottom` declarations are not used simultaneously, and neither are `left` and `right`.
- Text elements don't have a fixed height, have a fixed width only if specified in the brief, and will expand if more text is added to them.
- `flex` or `grid` layout is used to arrange elements. Elements are arranged using the `display: flex;` or `display: grid;` declarations where applicable.
- Links have a hover effect defined by `:hover`.

Form

- There is a form tag with other form elements nested inside it.
- Each field has:
 - A `label` with a corresponding text.
 - A `type` attribute with an appropriate value.
 - A `required` attribute.
 - A `placeholder` attribute with a corresponding text.
- The “number of persons” field has a minimum and maximum value constraint.

If the project complies with these criteria, you can send it for review. But you'll only pass if all of the criteria are met.

It's very useful to check your projects with your own checklists. This way, you'll develop a habit of checking that your work complies with the brief requirements.