Improve your solution

Take some time to find where you can improve your solution. You can use the design comparison slider and reports to help discover areas to work on. After updating your code and pushing the updates to your live site, generate a new screenshot or report to see the changes.

Improved your solution?

Generate a new screenshot to display your changes. You have unlimited screenshots.

REGENERATE SCREENSHOT

SOLUTION　　　　　　　DESIGN

Accessibility report   **HTML report**   CSS report   **JavaScript report**

Reports are only visible to you

Improved your solution?

Generate a new report to resolve errors and warnings.

REGENERATE REPORTS

Learn more about our CSS report

AI Analysis Summary

You've demonstrated strong use of CSS variables and responsive design principles with mobile-first media queries. Your global box-sizing and font inheritance improve consistency and usability. To take it further, consider avoiding fixed pixel positioning and absolute layout inside flex containers to enhance responsiveness and layout stability.

Priority recommendations:

1   Replace fixed pixel widths and percentage positioning with flexible units like percentages, flexbox, or grid to improve responsiveness.

2   Avoid using absolute positioning for major layout elements inside flex containers to prevent layout collapse and overlapping content.

3   Use CSS variables consistently for all repeated colors and values to improve maintainability and ease future theme changes.

Issues   7

All (7)   Error (0)   Warning (4)   Info (3)

WARNING   **Maintainability**   1 occurrence

Read more

Consolidate duplicate selectors to maintain an organized and efficient stylesheet.

```css
body {
  font-family: "Manrope", sans-serif;
  display: flex;
}
```

styles.css:105

LEARN HOW TO FIX THIS ISSUE

WARNING   **Responsive Design**   AI Detected   1 occurrence

Uses fixed pixel widths and positioning (e.g., left: 75.2%) which can break layouts on different screen sizes. This causes poor responsiveness and content overflow on smaller or zoomed screens.

```css
.social {
  max-width: 15.5rem;
  padding-top: 0.75rem;
  padding-bottom: 0.75rem;
  border-radius: 0.625rem;
  left: 75.2%;
  bottom: 33%;
}
```

styles.css:110

LEARN HOW TO FIX THIS ISSUE

WARNING   **Layout & Styling**   AI Detected   1 occurrence

Uses absolute positioning for the .social element inside a flex container which can cause parent container collapse or unexpected overlap. This impacts layout stability and can confuse users if content overlaps or is hidden.

```css
.social {
  box-shadow: 0 0.625rem 0.625rem 0 rgba(201, 213, 225, 0.5);
  background-color: var(--Very-Dark-Grayish-Blue);
  display: flex;
  align-items: center;
  padding: 1rem 1rem 1rem 2rem;
  width: 100%;
  position: absolute;
  left: 0;
  bottom: 0;
  border-bottom-left-radius: 0.625rem;
  border-bottom-right-radius: 0.625rem;
  border-top-right-radius: 0;
  border-top-left-radius: 0;
}
```

styles.css:92

LEARN HOW TO FIX THIS ISSUE

WARNING   **Maintainability**   AI Detected   1 occurrence

Repeated color values are hard-coded instead of using CSS variables consistently. This reduces maintainability and increases risk of inconsistent colors across the site.

```css
body {
  font-family: "Manrope", sans-serif;
  display: flex;
  align-items: center;
  justify-content: center;
  text-align: left;
  background-color: var(--Light-Grayish-Blue);
  padding: 0 1.5rem;
}
```

styles.css:25

LEARN HOW TO FIX THIS ISSUE

INFO   **Best Practice**   17 occurrences

Read more

Use logical properties (e.g., inline-start instead of left) to support different reading directions and improve internationalization.

```css
margin-top: 0;
```

styles.css:356

LEARN HOW TO FIX THIS ISSUE

INFO   **Best Practice**   21 occurrences

Read more

Consider using CSS functions like calc() , min() , and clamp() to create more responsive and flexible layouts that adapt to different viewport sizes.

padding: 0 1.5rem;

styles.css:112

LEARN HOW TO FIX THIS ISSUE

INFO   **Best Practice**   3 occurrences

Read more

Use CSS custom properties (variables) to centralize values, improve consistency, and make site-wide changes easier to implement.

background: #ffffff;

styles.css:119

LEARN HOW TO FIX THIS ISSUE

AI-detected issues notice

Some issues shown above were detected by AI analysis and may require human review for accuracy. If you're unsure about any findings or need clarification, please reach out in the help channel on Discord for assistance.

Strengths 5

Maintainability

Uses CSS custom properties (variables) for colors which centralizes theme management and improves consistency.

Layout & Styling

Applies box-sizing: border-box globally which simplifies sizing and layout calculations.

Responsive Design

Includes a mobile-first media query to adjust layout and typography for larger screens.

Performance

Uses filter for icon hover effects which is GPU-accelerated and avoids layout thrashing.

Accessibility

Inherits fonts for form elements to maintain visual consistency and improve user experience.

CONTINUE

Your path progress

Welcome to the JS fundamentals path!

Setting up your project to use JavaScript

The Document Object Model

Handling user events

Article preview component

Start solution

Submit solution

Improve solution

Peer review

Challenge completed

Managing forms with JavaScript

Newsletter sign-up form with success message

Working with data

Time tracking dashboard

Refactoring your code

Tip calculator app

Password generator app

Frontend Quiz app

|< Frontend Mentor

Stay up to date

with new challenges, featured solutions, selected articles, and our latest news

| email@example.com | SUBSCRIBE |

FRONTEND MENTOR

Unlock Pro

Contact us

EXPLORE

Learning paths

Challenges

FAQs

Become a partner

Use cases

Solutions

Articles

COMMUNITY

Discord

Guidelines

FOR COMPANIES

Hire developers

Train developers

© Frontend Mentor 2019 - 2026

Terms   Cookie Policy   Privacy Policy   License