



## LEARNING PATHS

### 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.

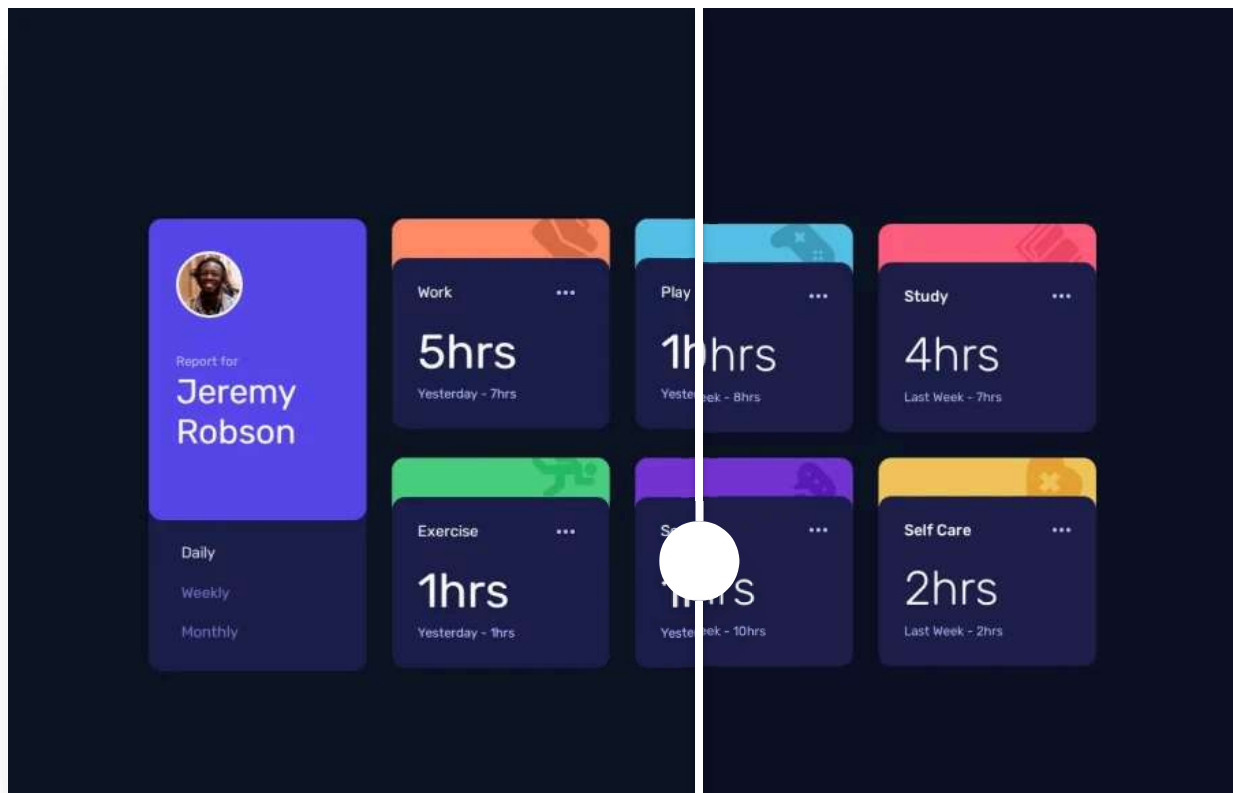
#### Improved your solution?

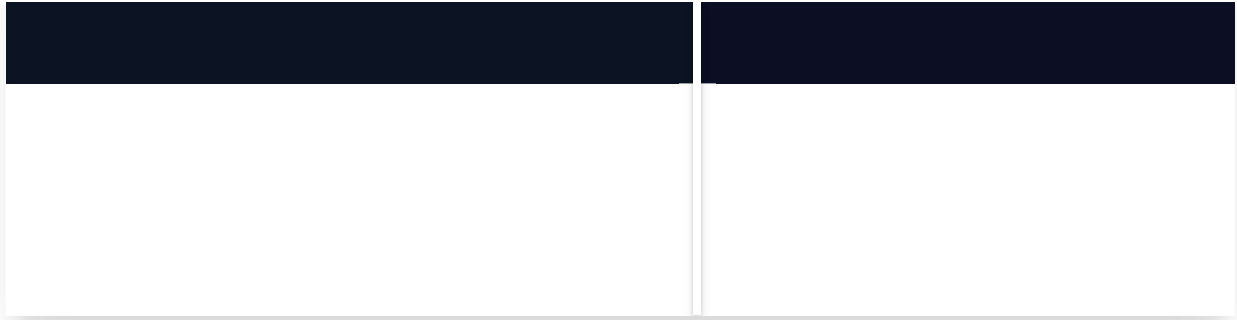
Generate a new screenshot to display your changes. You have unlimited screenshots.

*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 responsive design skills with mobile-first CSS Grid layouts and thoughtful accessibility focus styles. Your use of CSS variables and global box-sizing shows good maintainability practices. To take it further, consider using consistent naming for variables, avoid fixed widths for better flexibility, and optimize transitions for performance.

### Priority recommendations:

- 1 Use relative units (% , em , rem) or CSS functions like clamp() for widths and heights instead of fixed pixel/rem values to improve responsiveness.
- 2 Specify only the properties you want to animate in transitions (e.g., transition: color 0.3s; ) instead of all to enhance animation performance.
- 3 Adopt a consistent naming convention for CSS custom properties, such as all lowercase with hyphens (e.g., --blue , --light-orange ), to improve readability and maintainability.

## Issues 7

**All (7)**   **Error (0)**   **Warning (5)**   **Info (2)****WARNING**   **Accessibility**   2 occurrences[Read more](#)

Provide alternatives for users who prefer reduced motion to prevent motion sickness and other accessibility issues.

```
transition: all 0.3s;
```

*styles.css:221*

*LEARN HOW TO FIX THIS ISSUE*

**WARNING**   **Maintainability**   1 occurrence[Read more](#)

Consolidate duplicate selectors to maintain an organized and efficient stylesheet.

```
body {  
  font-family: "Rubik", sans-serif;  
  background: var(--Very-dark-blue);  
}
```

*styles.css:127*

*LEARN HOW TO FIX THIS ISSUE*

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

Using fixed widths like `width: 4rem` and fixed heights like `height: 2.375rem` can cause layout issues on different screen sizes. This reduces flexibility and may break responsiveness on smaller or larger devices.

```
.profile-card__avatar {  
  width: 4rem;  
  border: 0.1875rem solid var(--white);  
  border-radius: 100%;  
  background: #d8d8d8;  
}
```

*styles.css:54*

*LEARN HOW TO FIX THIS ISSUE***WARNING**    **Performance**    **AI Detected**    1 occurrence

Animating or transitioning the `all` property (e.g., `transition: all 0.3s;`) can cause unnecessary performance overhead because it triggers many layout and paint operations. This can make animations less smooth, especially on lower-end devices.

```
.profile-card__duration-option {  
  font-weight: 400;  
  font-size: 1.125rem;  
  line-height: 1.3125;  
  color: var(--Desaturated-blue);  
  transition: all 0.3s;  
  cursor: pointer;  
  width: fit-content;  
  border: transparent;  
  background: transparent;  
}
```

*styles.css:70**LEARN HOW TO FIX THIS ISSUE***WARNING**    **Maintainability**    **AI Detected**    1 occurrence

Using multiple color custom properties with inconsistent naming conventions (e.g., `--Blue` vs `--white`) can cause confusion and reduce maintainability. Consistent lowercase and hyphen-separated names improve readability and ease of use.

```
:root {  
  /* Primary */  
  --Blue: hsl(246, 80%, 60%);  
  --Light-orange: hsl(15, 100%, 70%);  
  --Soft-blue: hsl(195, 74%, 62%);  
  --Light-red: hsl(348, 100%, 68%);  
  --Lime-green: hsl(145, 58%, 55%);  
  --Violet: hsl(264, 64%, 52%);  
  --Soft-orange: hsl(43, 84%, 65%);  
  /* Neutral */  
  --Very-dark-blue: hsl(226, 43%, 10%);  
  --Dark-blue: hsl(235, 46%, 20%);  
  --Desaturated-blue: hsl(235, 45%, 61%);  
  --Pale-blue: hsl(236, 100%, 87%);  
  --white: hsl(0, 0%, 100%);  
}
```

*styles.css:1*

**LEARN HOW TO FIX THIS ISSUE****INFO**   **Best Practice**   15 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-inline: 1.5rem;
```

```
styles.css:132
```

**LEARN HOW TO FIX THIS ISSUE****INFO**   **Best Practice**   2 occurrences[Read more](#)

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

```
background: #d8d8d8;
```

```
styles.css:205
```

**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**Responsive Design**

Uses mobile-first media queries with clear grid-template-areas for responsive layouts, making the design adapt smoothly from mobile to desktop.

### Accessibility

Focus styles with `:focus-visible` improve keyboard navigation without distracting mouse users, enhancing accessibility.

### Layout & Styling

Employs CSS Grid effectively for complex layouts, including nested grids for components, which simplifies layout management.

### Maintainability

Uses CSS custom properties for colors, centralizing theme management and enabling easy updates.

### Best Practice

Sets `box-sizing: border-box` globally to simplify sizing and layout calculations.

*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

Managing forms with JavaScript

Newsletter sign-up form with success message

Working with data

Time tracking dashboard

Start solution

Submit solution

**Improve solution**

Peer review

Challenge completed

Refactoring your code

Tip calculator app

Password generator app

Frontend Quiz app



## Stay up to date

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

*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](#)