3

LEARNING PATHS

Improve your solution

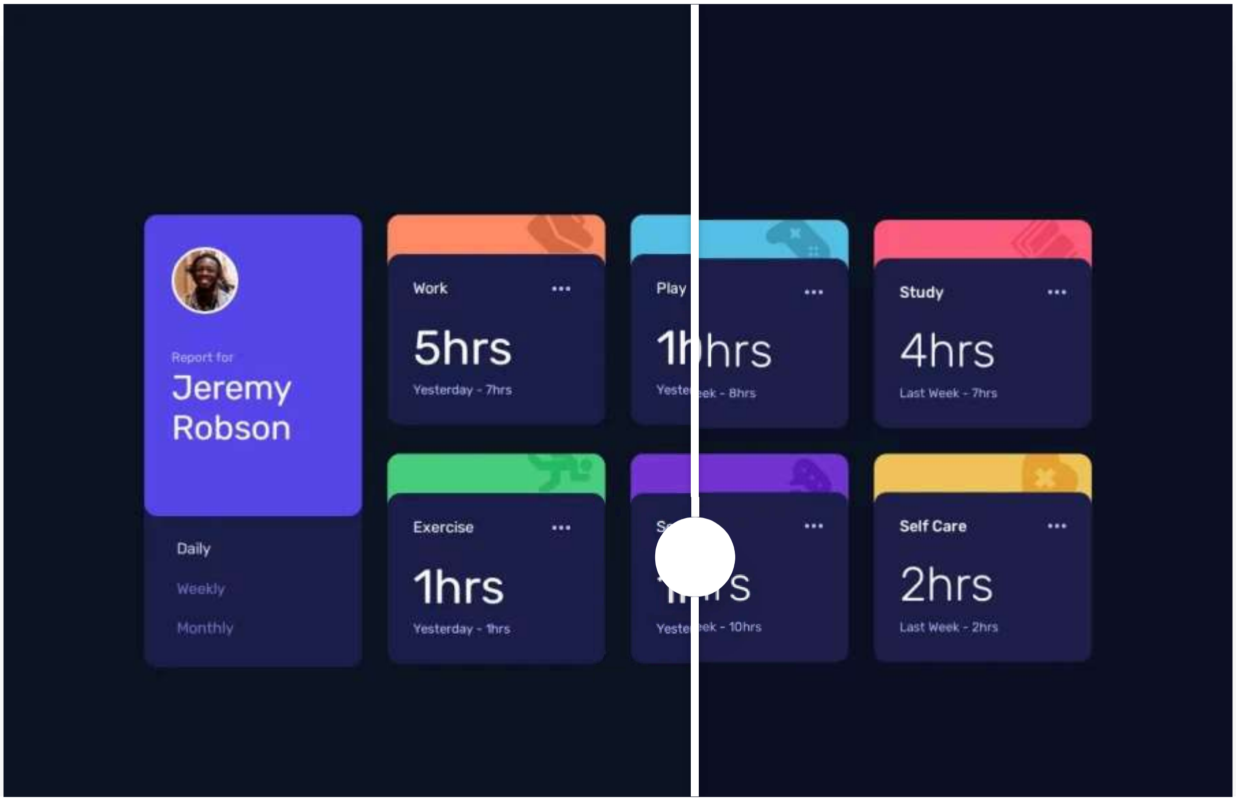
Take some time to find where you can improve your solution. You can use the design comparison slider and reports to help discover areas to work on. After updating your code and pushing the updates to your live site, generate a new screenshot or report to see the changes.

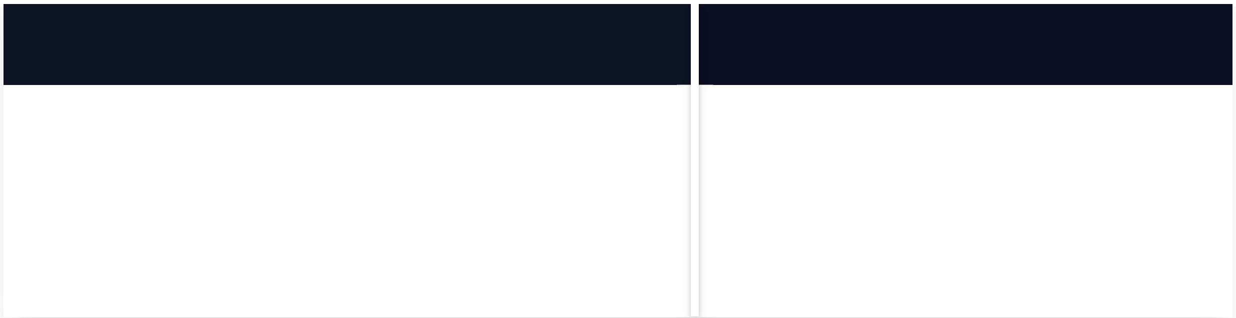
Improved your solution?

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 JavaScript report

AI Analysis Summary

You've demonstrated excellent modular design and accessibility awareness in your code! Your clear variable naming and separation of concerns make it easy to follow. To take it further, consider loading data dynamically from the JSON file instead of using hardcoded data.

Priority recommendations:

1. Consider fetching the data from the external `data.json` file using `fetch` API to make your app more flexible and maintainable.
2. Add error handling for the data fetch to improve robustness in case the JSON file is missing or malformed.
3. Use a mapping object or switch statement to reduce repetitive `if-else` blocks in `updateActiveTimeframe` and `updateLinkStyles` functions for cleaner code.

Issues 3

All (3) **Error (0)** **Warning (3)** **Info (0)**

WARNING **Maintainability** 2 occurrences

Read more

Remove variables that are declared but never used

```
const activityDetails = document.querySelectorAll(".activity__details");
```

script.js:10

LEARN HOW TO FIX THIS ISSUE

WARNING **Modern JavaScript** 1 occurrence

Read more

Use const for variables that are never reassigned

```
let data = [
```

script.js:12

LEARN HOW TO FIX THIS ISSUE

WARNING **Best Practice** 1 occurrence

Read more

Don't declare variables that hide outer variables with the same name

```
function addData(data, timeframe) {
```

script.js:118

LEARN HOW TO FIX THIS ISSUE

STRENGTHS 5

READABILITY & FORMATTING

CLEAR AND CONSISTENT NAMING OF VARIABLES LIKE DAILY , WEE
MONTHLY MAKES THE CODE EASY TO UNDERSTAND.

MAINTAINABILITY

FUNCTIONS LIKE UPDATETIMEFRAME , UPDATELINKSTYLES ,
UPDATEACTIVETIMEFRAME SEPARATE CONCERNS WELL, MAKING
MODULAR AND EASIER TO MAINTAIN.

ACCESSIBILITY

USE OF ARIA-PRESSED ATTRIBUTES ON TIMEFRAME BUTTONS IM
ACCESSIBILITY FOR SCREEN READER USERS.

MODERN JAVASCRIPT

USE OF TEMPLATE LITERALS FOR STRING INTERPOLATION ENH
READABILITY.

LOGIC & CORRECTNESS

THE CODE CORRECTLY UPDATES UI ELEMENTS BASED ON SELECTED T
ENSURING CONSISTENT 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
- 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

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
FAQs
Become a partner
Use cases

EXPLORE

Learning paths
Challenges
Solutions
Articles

COMMUNITY

Discord
Guidelines

FOR COMPANIES

Hire developers
Train developers

Terms Cookie Policy Privacy Policy License