Frontend Mentor – Time Tracking Dashboard

Overview

This project is a solution to the **Time Tracking Dashboard** challenge from Frontend Mentor. The goal of the challenge is to build a responsive dashboard that updates time tracking data based on selected timeframes (Daily, Weekly, Monthly).

Through this project, I focused on:

- Writing clean and semantic HTML
- Improving CSS layout skills (Flexbox & Grid)
- Using JavaScript to handle UI interactions and dynamic data updates
- Understanding common frontend issues and best practices

This project helped me move from *following tutorials* to *thinking and debugging on my own*.

Built with

- **HTML5** – Semantic structure
- **CSS3** – Flexbox, Grid, custom properties (CSS variables)
- **JavaScript (ES6+)** – Event handling, modules, DOM manipulation

---

What I learned

1️⃣ Avoiding fragile CSS hacks for alignment

Problem: I initially aligned an image to the right using this:

```css
.tck-category-img img {
  margin-left: auto;
}
```

While this worked, it depended on the image's context and could easily break if the layout changed.

Better approach:

```css
.tck-category-img {
  display: flex;
  justify-content: flex-end;
}
```

Why this is better:

- Uses layout rules instead of element-level hacks
- More predictable and responsive
- Easier to maintain and understand later

2️⃣ Practical understanding of Event Delegation

Instead of adding click listeners to each tab individually, I used **event delegation** by attaching one listener to the parent element.

Key concepts I understood clearly:

- `event.target` → The actual element that was clicked
- `event.currentTarget` → The element the event listener is attached to

This approach:

- Reduces the number of event listeners
- Works even if new elements are added dynamically
- Makes the code cleaner and more scalable

3️⃣ Understanding `event.target.dataset`

I learned that:

```
event.target.dataset
```

- Returns an **object** containing all `data-*` attributes of the clicked element
- Helps identify which tab (daily / weekly / monthly) is active

Example:

```
<li data-active-tab="daily">Daily</li>
```

```
event.target.dataset.activeTab // "daily"
```

This made it easy to switch UI states and update data based on user interaction.

4️⃣ Fixing `Cannot use import statement outside a module`

Error:

```
Uncaught SyntaxError: Cannot use import statement outside a module
```

Cause: The browser did not know that my JavaScript file was using ES modules.

Solution:

```
<script type="module" src="index.js"></script>
```

What I learned:

- ES module syntax (`import/export`) only works when the script type is set to `module`
- Browsers treat module files differently (strict mode by default)

5️⃣ Preventing image resizing on screen changes

Issue: Images were resizing unexpectedly when the screen size changed.

Fix & best practices:

- Use fixed dimensions or `max-width`
- Control scaling with `object-fit`

Example:

```
img {
  max-width: 100%;
  height: auto;
}
```

Lesson: Responsive images need explicit constraints to avoid layout shifts.

Links

- Live Site: https://apsbundela.github.io/fem-time-tracking/