Lesson 6 Exercises - Routing, Git, Ecommerce Project

6a. In the checkout page, split up the header into a separate component called `CheckoutHeader`.

- Since `CheckoutHeader` is not shared (it's only used on the checkout page), save it in the `pages` folder instead of `components` folder.
- Rename `checkout-header.css` to `CheckoutHeader.css`. Update the imports so each component imports the css file with the same name.

6b. When there are multiple components for a page, we usually group them together into a folder.

- Inside the `pages` folder, create a folder named `checkout`, and move all components and CSS related to the checkout page into this folder.
- Update all your imports (for example in `App.jsx`) and test the website.

6c. In `CheckoutHeader` replace all <a> elements with <Link> components.

6d. Save the changes from the previous exercises into git.

- For the rest of the exercises, save changes into git after each exercise.

6e. Rename `header.css` to `Header.css` so it matches the component's name. Check the git section to see if git detected the change.

- If git detected the change, save the change into git (using Commit).
- If git did not detect the change, this happens because on some operating systems (Windows and Mac), git is case-insensitive. That means `header.css` and `Header.css` are considered the same.
- To fix this, rename `header.css` to something like `header2.css`. Save these changes into git (using Commit). Then, rename `header2.css` to `Header.css` and git will now detect the change.

6f. React Router has another component called `<NavLink>` that is useful for navigation links (links at the top of a page, usually in a header).

- For an example, check apple.com and see the links at the top.
- In `Header.jsx`, update all `<Link>`s to `<NavLink>`s (props are the same).
- The special feature of `<NavLink>` is it knows which page is loaded. For example, if you're on the Orders page, it adds a class called `active` to the Orders link (`className="orders-link ... active"`).
- Inside `Header.css`, style the orders link so when it's active, "Orders" is underlined (Hint: `.orders-link.active` will style an element with the class `orders-link` and `active`. Use `text-decoration: underline;`)
- Open the website, go to the orders page, and check the Orders link in the header.

Solutions in description

6g. We'll use React to change the icon in the tabs (this is called a favicon)

- In index.html, look for `<link rel="icon" href="...">`. This sets the favicon. Copy this into `HomePage`, `CheckoutPage`, and `OrdersPage`.
- In your browser, open `supersimple.dev/images/home-favicon.png` and download the image into the `public` folder. Do the same for `supersimple.dev/images/cart-favicon.png` and `supersimple.dev/images/orders-favicon.png`.
- In `HomePage` set the href of the favicon to `home-favicon.png` (if we set href to a file name, Vite will look for the file in the `public` folder).
- Update `CheckoutPage` to use `cart-favicon.png` and `OrdersPage` to use `orders-favicon.png`
- In index.html, remove `<link rel="icon" href="...">`
- Open the website and switch pages. Notice the favicon changes.

Challenge Exercises

6h. We'll move the tracking page to React (open the website, go to the orders page, and click a "Track package" button).

- Move over the HTML (remember to reuse the `<Header>` component).
- Download `supersimple.dev/images/tracking-favicon.png` and use it as the favicon.
- Move over the CSS, rename it to match the Component, and import.
- Create a route for the tracking page in App.jsx (use `path="tracking"`).
- Open the website and test out the tracking page.

6i. Using the Search section of VSCode (`Ctrl + Shift + F` on Windows or `Command + Shift + F` on Mac), find any other `<a>` elements in the code and replace them with the `<Link>` component.

6j. Create a folder `src/assets/images`. Open `public/images`, move the 4 logos at the bottom and the `icons` folder to `src/assets/images` (we usually save logos and icons in `src/assets/images`, but not favicons).

- Using the Search section of VSCode, find where each logo and icon is used in the code, import the image, and insert it using `src={...}` instead of using a string for the `src` attribute.
- (We don't do this for product images or ratings because each product can have a different image and rating. This makes importing difficult).

6k. We'll add a 404 (Not Found) page. Create a page that displays the `<Header>` and message "Page not found" (style it however you want).

- Create a `<Route>` with `path="*"` (this matches any URL path), set the `element` to your 404 page. Add this route to the bottom of `<Routes>` (if the URL does not match any other route, it will display your 404 page).