

# Using React

**Dr. Atef Bader**

# Code Café React App

File Edit View History Bookmarks Tools Help

6. Prop Types | React Programming X

https://learning.oreilly.com/library/view/react-programming-the/9780137901777/ch06.xhtml 67%

O'REILLY Topics Start Learning Search 50,000+ courses, events, titles, and more

Featured

## Chapter 6. Prop Types

Now that your application is set up, you are ready to begin building Code Café.

When you worked on Ottergram, you built it using components – reusable JavaScript functions that return React elements. Components are key to building scalable React applications, so understanding how they work is crucial.

Building your own components is an important skill for React developers. In this chapter, you will reinforce what you learned from building Ottergram by creating and using components for Code Café.

Another important aspect of React development is using components that other developers have built – and building components that other developers will use. Knowing the exact props that a component needs can be difficult, especially when it needs many of them. And passing incorrect props to a component can lead to errors in the application.

To make components easier to work with, you can assign *prop types* to your component props. A prop type defines the data type, such as `string` or `number`, required for a given prop. (If you have experience with typed function parameters in TypeScript or other languages, prop types will likely feel familiar.)

Prop types serve as documentation to help future developers know how to use your component. They also give you some automatic checks for compatibility.

5. Linting 6. Prop Types 7. Styles

4h 31m remaining

# Code Café React App

File Edit View History Bookmarks Tools Help

6. Prop Types | React Programming

https://learning.oreilly.com/library/view/react-programming-the/9780137901379


O'REILLY Topics Start Learning Search 50,000+ courses, events, titles, and more

quired for a given prop. (If you have experience with typed function parameters in TypeScript or other languages, prop types will likely feel familiar.)

Prop types serve as documentation to help future developers know how to use your component. They also give you some automatic checks for compatibility between the type required for a component's prop and the actual prop being passed in, as you will see in a moment.

By the end of this chapter, Code Café will have three components: a **Header** component proudly proclaiming the name of the site, a **Home** component that renders when a user first navigates to your site, and a **Thumbnail** component you will use multiple times to display a list of items available for purchase (Figure 6.1, “Code Café at the end of this chapter”). And you will use the prop-types library to document all the props your components use.

Figure 6.1. Code Café at the end of this chapter



The screenshot shows the Code Café website. At the top is a header with the site's name and a coffee cup icon. Below the header is a grid of six food items, each with an icon and a label: Drip Coffee, Cookie, Croissant, Cupcake, French Press, and Iced Coffee. The website is displayed in a browser window with the URL https://learning.oreilly.com/library/view/react-programming-the/9780137901379. A red box highlights the text 'Code Café will have three components' and a red arrow points from this box to the text 'By the end of this chapter, Code Café will have three components: a Header component proudly proclaiming the name of the site, a Home component that renders when a user first navigates to your site, and a Thumbnail component you will use multiple times to display a list of items available for purchase (Figure 6.1, “Code Café at the end of this chapter”). And you will use the prop-types library to document all the props your components use.'

5. Linting 6. Prop Types 7. Styles

4h 31m remaining

# Code Café React App

The screenshot shows a web browser window with the address bar displaying `https://learning.oreilly.com/library/view/react-programming-the-9780137901`. The page content includes text about `imageId` and `items` arrays, and instructions for creating a `Thumbnail` component. A blue callout box with a red border contains the text: "Your app will display each item in a Thumbnail component". A red arrow points from this box to the `Thumbnail` component name in the text. At the bottom, a search bar contains the text "Your app will display" and has several checkboxes for search options: "Highlight All", "Match Case", "Match Diacritics", and "Whole Words". The search results show "1 of 1 match".

File Edit View History Bookmarks Tools Help

6. Prop Types | React Programming

`https://learning.oreilly.com/library/view/react-programming-the-9780137901`

`imageId` and the value is the corresponding image. Hyphenated object properties, such as the multi-word `imageId`s, are wrapped in single quotes. Importing this object into any file will give you access to all the images – without having to import each one individually.

The second variable, `items`, is an array of the items available for sale. It represents the information that you would normally store in a database, including details about each item such as the title, price, `imageId`, and `itemId`. Eventually, you will replace this variable with a call to an API. But for now, you will use this mock data so you can continue to focus on your React skills.

Your app will display each item in a `Thumbnail` component. Clicking a `Thumbnail` will route the user to a details page for the item. You will add the routing functionality later; for now, you will focus on displaying the `Thumbnail` components on the page.

Create the component file and corresponding CSS file in the `components` directory. Name the files `Thumbnail.js` and `Thumbnail.css`.

Add the styles for the component to `Thumbnail.css` first:

Example 6.5. Adding thumbnail styles (`Thumbnail.css`)

[Click here to view code image](#)

```
.thumbnail-component {
  display: flex;
  flex-direction: column;
  width: 100%;
  text-decoration: none;
  color: black;
}
```

Your app will display

☐ Highlight All ☐ Match Case ☐ Match Diacritics ☐ Whole Words 1 of 1 match

# Code Café React App

App.js uses Home component for the items

File Edit Selection View Go Run Terminal Help App.js - react-book - Visual Studio Code

EXPLORER

OPEN EDITORS

- JS App.js 06-prop-types/src
- JS Home.js 06-prop-types/src/components
- JS Thumbnail.js 06-prop-types/src/components
- index.html 07-styles/public
- # Thumbnail.css 07-styles/src/components

REACT-BOOK

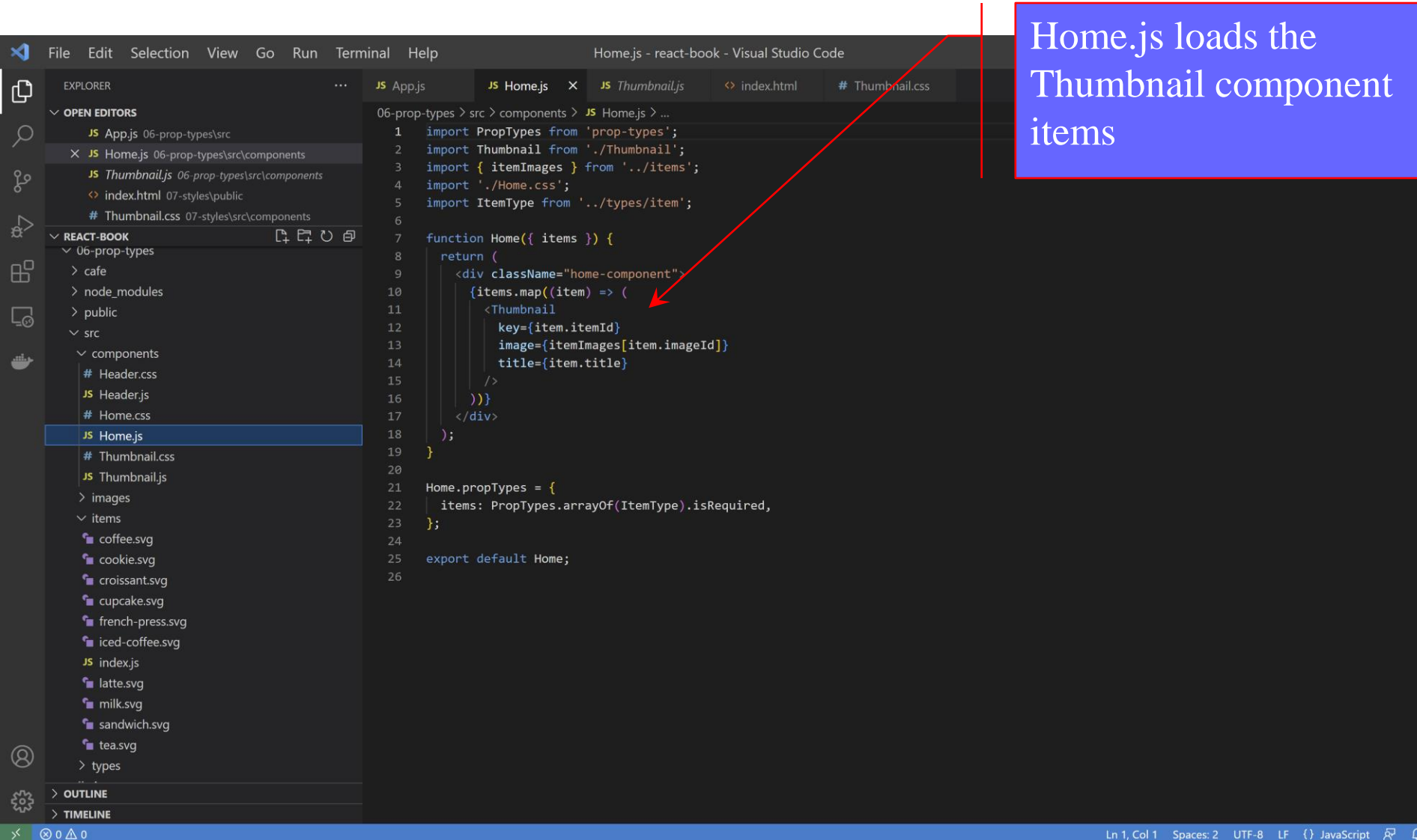
- coffee.svg
- cookie.svg
- croissant.svg
- cupcake.svg
- french-press.svg
- iced-coffee.svg
- index.js
- latte.svg
- milk.svg
- sandwich.svg
- tea.svg
- > types
- # App.css
- JS App.js**
- JS App.test.js
- # index.css
- JS index.js
- logo.svg
- JS reportWebVitals.js
- JS setupTests.js
- .gitignore
- { } package-lock.json
- { } package.json
- ① README.md
- > 07-styles

06-prop-types > src > JS App.js > ...

```
1 import './App.css';
2 import Header from './components/Header';
3 import Home from './components/Home';
4 import { items } from './items';
5
6 function App() {
7   return (
8     <div>
9       <Header />
10      <Home items={items} />
11    </div>
12  );
13 }
14
15 export default App;
16
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF { } JavaScript

# Code Café React App



Home.js loads the Thumbnail component items

```
06-prop-types > src > components > JS Home.js > ...
1  import PropTypes from 'prop-types';
2  import Thumbnail from '../Thumbnail';
3  import { itemImages } from '../items';
4  import './Home.css';
5  import ItemType from '../types/item';
6
7  function Home({ items }) {
8    return (
9      <div className="home-component">
10        {items.map((item) => (
11          <Thumbnail
12            key={item.itemId}
13            image={itemImages[item.imageId]}
14            title={item.title}
15          />
16        ))}
17      </div>
18    );
19  }
20
21  Home.propTypes = {
22    items: PropTypes.arrayOf(ItemType).isRequired,
23  };
24
25  export default Home;
26
```

# Code Café React App

index.js exports all items

```
06-prop-types > src > items > JS index.js > ...
1  import Croissant from './croissant.svg';
2  import Coffee from './coffee.svg';
3  import Cookie from './cookie.svg';
4  import Cupcake from './cupcake.svg';
5  import FrenchPress from './french-press.svg';
6  import IcedCoffee from './iced-coffee.svg';
7  import Latte from './latte.svg';
8  import Milk from './milk.svg';
9  import Sandwich from './sandwich.svg';
10 import Tea from './tea.svg';
11
12 export const itemImages = {
13   coffee: Coffee,
14   cookie: Cookie,
15   croissant: Croissant,
16   cupcake: Cupcake,
17   'french-press': FrenchPress,
18   'iced-coffee': IcedCoffee,
19   latte: Latte,
20   milk: Milk,
21   sandwich: Sandwich,
22   tea: Tea,
23 };
24
25 export const items = [
26   {
27     itemId: 'coffee',
28     imageId: 'coffee',
29     title: 'Drip Coffee',
30     price: 1,
31     description: '',
32     salePrice: 0,
33   },
34   {
35     itemId: 'cookie',
36     imageId: 'cookie',
37     title: 'Cookie',
38     price: 0.99,
39     description: 'May contain nuts.',
40     salePrice: 0.50,
```

# Code Café React App

npm start

Windows PowerShell

```
> code-café@0.1.0 start  
> react-scripts start
```

Browserslist: caniuse-lite is outdated. Please run:

```
npx update-browserslist-db@latest
```

Why you should do it regularly: <https://github.com/browserslist/update-db#readme>

(node:6572) [DEP\_WEBPACK\_DEV\_SERVER\_ON\_AFTER\_SETUP\_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use

(Use 'node --trace-deprecation ...' to show where the warning was created)

(node:6572) [DEP\_WEBPACK\_DEV\_SERVER\_ON\_BEFORE\_SETUP\_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.

Starting the development server...

Compiled successfully!

You can now view code-café in the browser.

Local: <http://localhost:3000>

On Your Network: <http://192.168.0.222:3000>

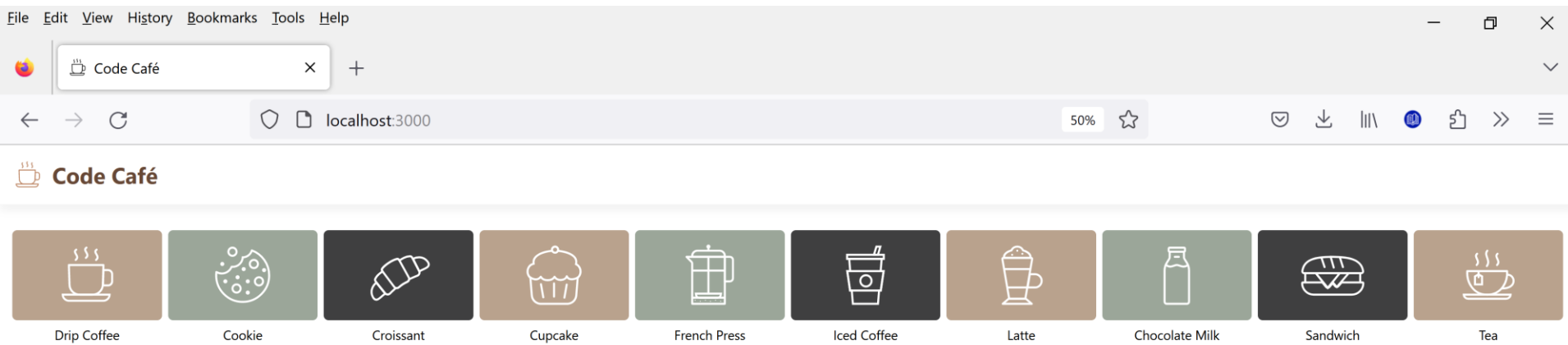
Note that the development build is not optimized.

To create a production build, use `npm run build`.

webpack compiled successfully



# Code Café React App



# Code Café React App

CSS for Code Café that will have a UI that looks good on both mobile devices and desktop browsers

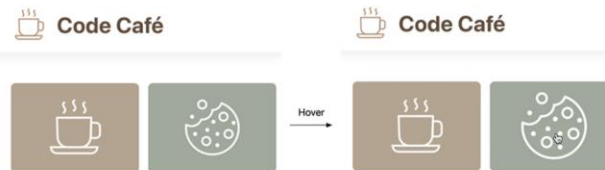
## Chapter 7. Styles

When it comes to UI development, form and function go hand in hand. A well-styled UI is important for a successful app.

You have already seen that React makes it easy to add styles by importing CSS and using the `className` keyword in components. In this chapter, you will add more style rules to enhance the design of your application.

By the end of the chapter, Code Café will have a UI that looks good on both mobile devices and desktop browsers and that has some polished features (**Figure 7.1. "Hover effect"**). This will set the stage for the order functionality you will begin adding in the next chapter.

Figure 7.1. Hover effect



# Code Café React App

File Edit View History Bookmarks Tools Help

7. Styles | React Programming: Tl X

https://learning.oreilly.com/library/view/react-programming-the/9780137901777/ch0

(Leave the `.header-component p` style rule in place. Although it does not do anything at the moment, it will play a role later.)

## Responsive Design

The next style concept to consider is *responsive design*, a design approach meant to ensure that your site works across a variety of screen sizes and devices.

Ultimately, Code Café should function and look good on both desktops and mobile devices. **Figure 7.4. “Desktop design”** and **Figure 7.5. “Mobile design”** show the target designs:

Figure 7.4. Desktop design

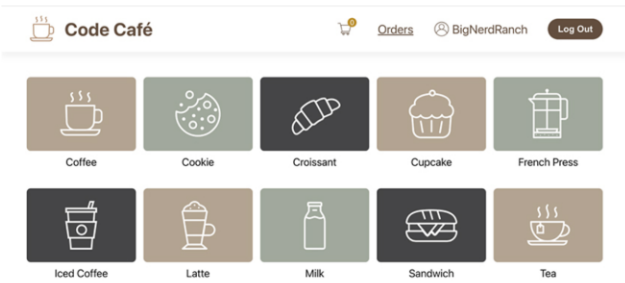
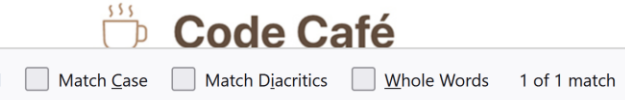


Figure 7.5. Mobile design

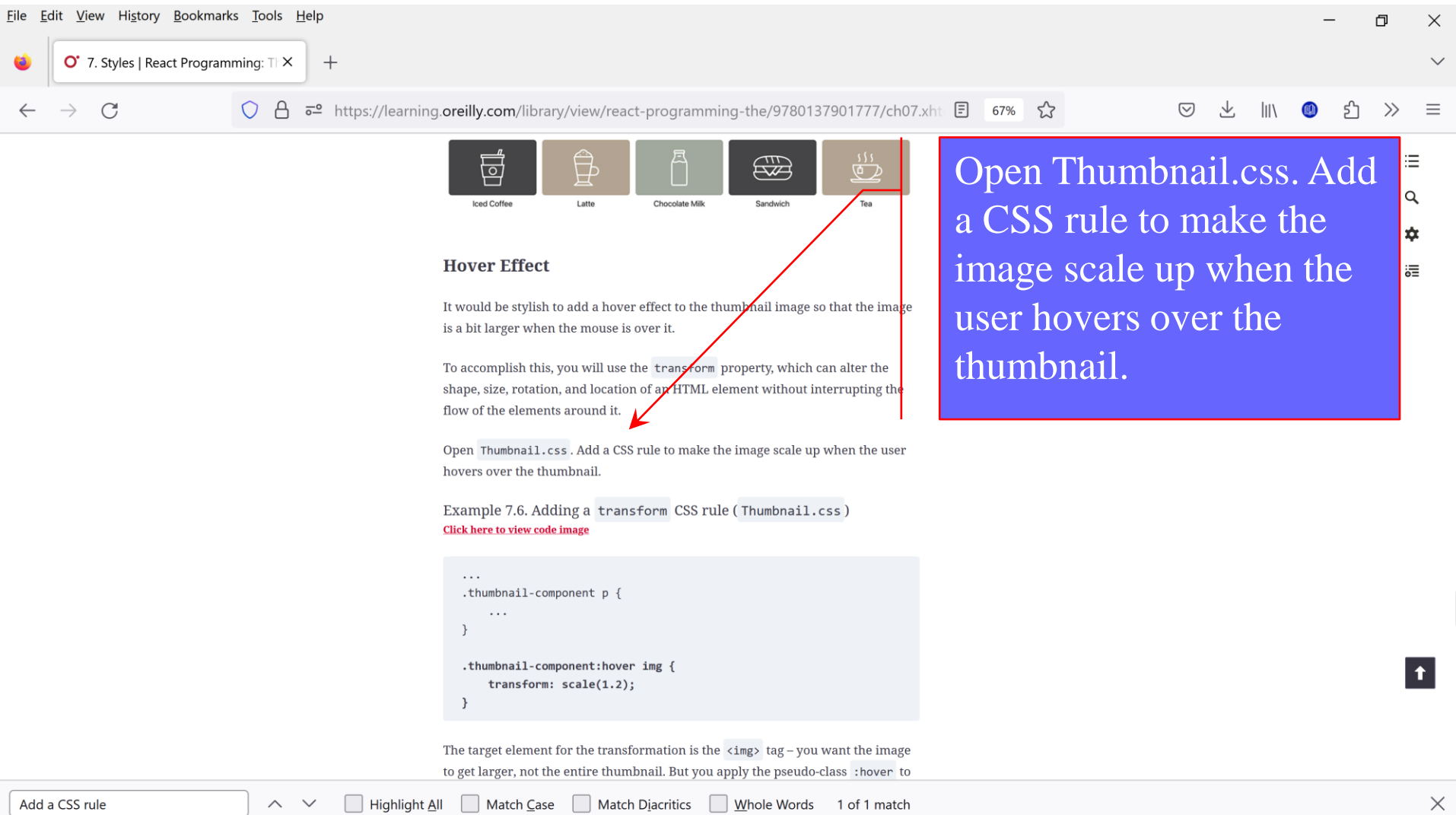


Your app will display

☐ Highlight All ☐ Match Case ☐ Match Diacritics ☐ Whole Words 1 of 1 match

Responsive design is a design approach meant to ensure that your site works across a variety of screen sizes and devices.

# Code Café React App



File Edit View History Bookmarks Tools Help

7. Styles | React Programming: TI X

https://learning.oreilly.com/library/view/react-programming-the/9780137901777/ch07.xhtml 67%

Iced Coffee Latte Chocolate Milk Sandwich Tea

### Hover Effect

It would be stylish to add a hover effect to the thumbnail image so that the image is a bit larger when the mouse is over it.

To accomplish this, you will use the `transform` property, which can alter the shape, size, rotation, and location of an HTML element without interrupting the flow of the elements around it.

Open `Thumbnail.css`. Add a CSS rule to make the image scale up when the user hovers over the thumbnail.

Example 7.6. Adding a `transform` CSS rule (`Thumbnail.css`)

[Click here to view code image](#)

```
...  
.thumbnail-component p {  
  ...  
}  
  
.thumbnail-component:hover img {  
  transform: scale(1.2);  
}
```

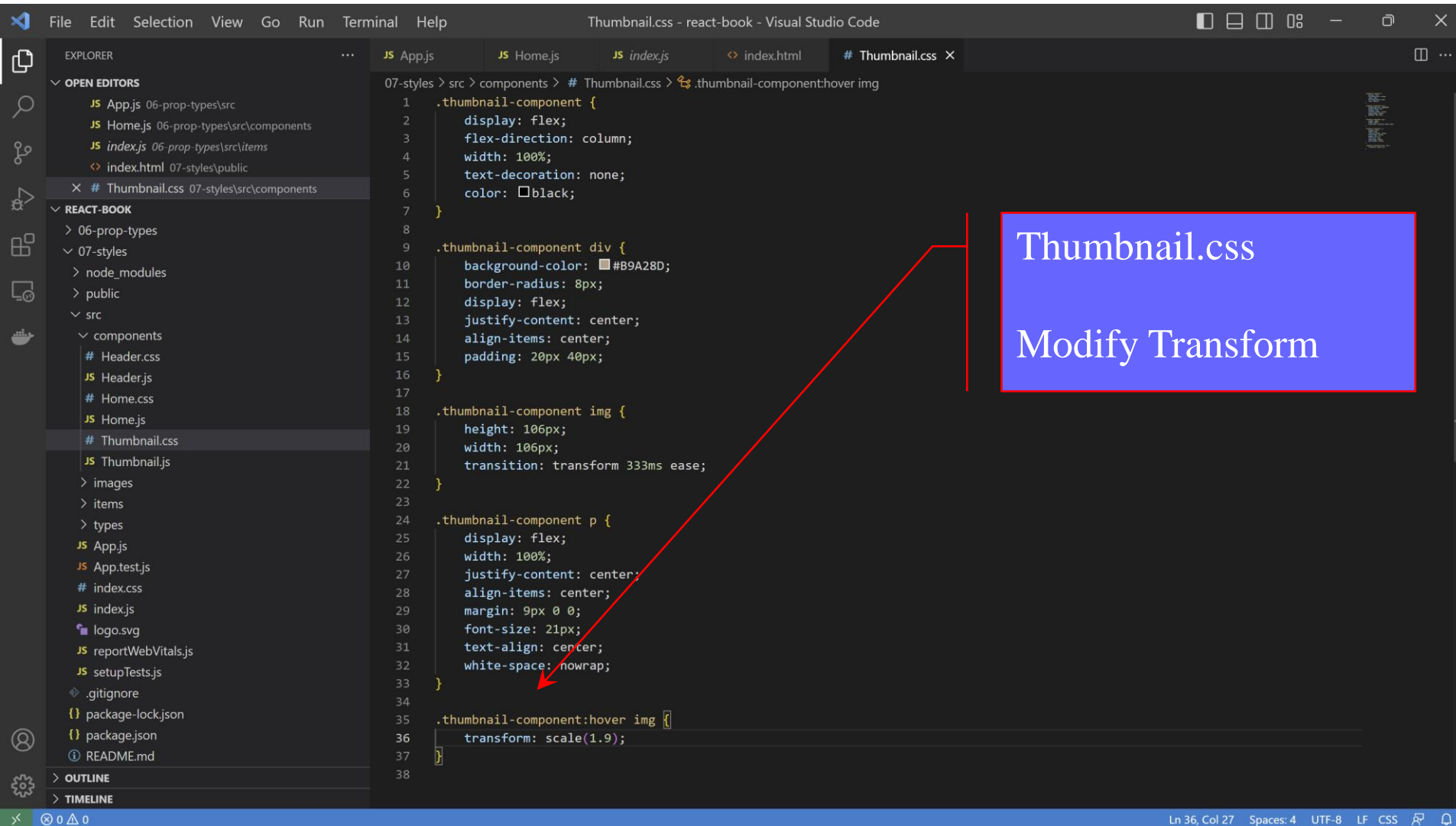
The target element for the transformation is the `<img>` tag – you want the image to get larger, not the entire thumbnail. But you apply the pseudo-class `:hover` to

Add a CSS rule

☐ Highlight All ☐ Match Case ☐ Match Diacritics ☐ Whole Words 1 of 1 match

Open Thumbnail.css. Add a CSS rule to make the image scale up when the user hovers over the thumbnail.

# Code Café React App



# Code Café React App

Windows PowerShell

```
npx update-browserslist-db@latest
Why you should do it regularly: https://github.com/browserslist/update-db#readme
(node:6736) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use 'onAfterSetup' instead. (Use 'node --trace-deprecation ...' to show where the warning was created)
(node:6736) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use 'onBeforeSetup' instead.
Starting the development server...
Compiled successfully!
```

You can now view code-cafe in the browser.

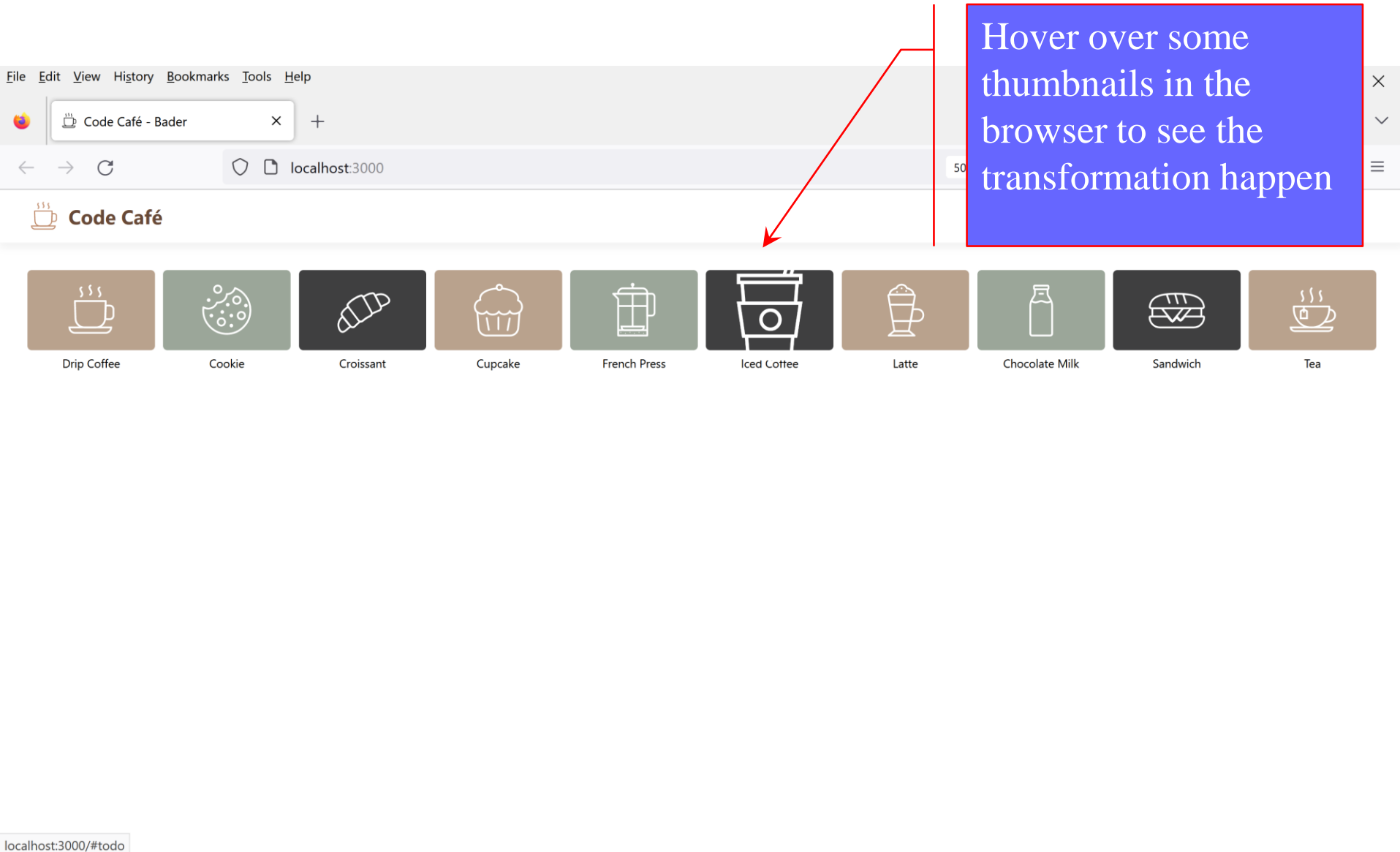
```
Local:      http://localhost:3000
On Your Network: http://192.168.0.222:3000
```

Note that the development build is not optimized.  
To create a production build, use `npm run build`.

webpack compiled **successfully**

npm start

# Code Café React App



# References

- Rosson, M. B., John M. 2001. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction* (1st Edition). San Francisco, CA: Morgan Kaufmann Publishers. [ISBN: 978-1558607125]
  - Tidwell, J., Brewer, C., Valencia, A. 2020. *Designing Interfaces* (3rd Edition). Sebastopol, CA: O'Reilly Media. [ISBN: 978-1492051961]
  - Klingman, L. 2023. *React Programming: The Big Nerd Ranch Guide*. Indianapolis, IN: Pearson Technology Group. [ISBN: 978-0137901692]
  - De Voil, N. 2016. *User Experience Foundations*. Swindon, UK: BCS Learning & Development. [ISBN: 978-1780173498]
  - Nielsen, J. 1994. *Usability Engineering*. Cambridge, MA: Morgan Kaufmann. [ISBN: 978-0125184069]
  - Ritter, M., Winterbottom, C. 2017. *UX for the Web*. Birmingham, UK: Packt Publishing. [ISBN: 978-1787128477]
  - Nielsen, J., Loranger, H. 2006. *Prioritizing Web Usability*. Berkeley, CA: New Riders. [ISBN: 978-0321350312]
  - Yablonski, J. 2020. *Laws of UX*. Sebastopol, CA: O'Reilly Media. [ISBN: 978-1492055310]
  - MacDonald, D. 2019. *Practical UI Patterns for Design Systems: Fast-Track Interaction Design for a Seamless User Experience*. New York, NY: Springer Science. [ISBN: 978-1484249376]
  - Timms, S. 2016. *Mastering JavaScript Design Patterns (2<sup>nd</sup> Edition)*. Birmingham, UK: Packt Publishing. [ISBN: 978-1785882166]
  - Larsen, J. 2016. *React Hooks in Action*. Shelter Island, NY: Manning Publications. [ISBN: 978-1617297632]
  - Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., Diakopoulos, N. 2016. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (6<sup>th</sup> Edition)*. Essex, England: Pearson Education Publications. [ISBN: 978-1292153919]
  - Flanagan, D. 2020. *JavaScript: The Definitive Guide (7<sup>th</sup> Edition)*. Sebastopol, CA: O'Reilly Media. [ISBN: 978-1491952023]
  - Leventhal, Li., Julie Barnes, J. 2007. *Usability Engineering: Process, Products & Examples* (1st Edition). Essex, England: Pearson Education Publications. [ISBN: 978-0131570085]
-