

## Simulation Storyboard

### Use CSS to style a web page

1. In the VS Code tool, you are starting with the empty CSS file called **style.css** to style your web page. You need to think about the background and the style of the individual elements such as the height, width, margins, padding, borders, fonts, and colors. Select the **Next arrow** to continue.
2. First, you need to add general styles that apply to your entire page. You can start by creating a property that automatically adjusts elements for the size of your web page borders. Select the **Next arrow** to continue.
3. Under **style.css**, place your cursor next to **1**. This is the first line of code. Type one line of code in each field exactly as it displays in the Code pad. Then, press **Enter**.

Code pad code:

```
*,
*:before,
*:after{
padding: 0;
margin: 0;
box-sizing: border-box;
}
```

4. Next, you set the height to **100 maximum possible viewpoint height (vh)** to give your users a consistent browsing experience. You follow company branding by setting the background color to **#0F62FE**, which is the hexadecimal value for the company's blue. Select the **Next arrow** to continue.
5. Now, set the height and background for the page. Set an element style to adjust the position of the HTML element. For an element style, you don't need to use a period or pound symbol. Under the code you just typed, type one line of code in each field exactly as it displays in the Code pad. Then, press **Enter**.

Code pad code:

```
body{
height: 100vh;
background: #0F62FE;
}
```

6. You just set some **general styles**. Now you can style **individual elements**. Create an **id selector** for the header in the body element to set the text color as the company's blue

and the font as sans serif. Under the code you just typed, type the code from the Code pad. Then, press **Enter**.

Code pad code:

```
#head-text{  
margin-top: 50px;  
margin-left: 10px;  
font-family: sans-serif;  
color: #0F62FE;  
}
```

7. Your page is shaping up! You can see that CodeSwing is showing how your code will display in the Preview pane as you develop your page. Select the **Next arrow** to continue.
8. Now you can style the container that has the task list items. To make the container stand out from the background, create a white background and set the sizing. Use a **class selector**. Under the code you just typed, type the code from the Code pad. Then, press **Enter**.

Code pad code:

```
.container{  
width: 40%;  
top: 30%;  
left: 50%;  
background: white;  
border-radius: 1px;  
min-width: 450px;  
position: absolute;  
min-height: 100px;  
transform: translate(-50%,-50%);  
}
```

### Continue styling the web page with CSS

For the purpose of your project scenario, you add more individual style elements to design the user interface and improve the look of the elements on your web page. For instance, you can add positions, colors, fonts, and sizing elements to the section where users will add their new tasks in the task list.

Next, see what some of this code looks like in your **style.css** file.

Select **X** to close this window and continue.

9. Here's the code for the styles that set the position and colors of the interface where your users add tasks. Notice the **chained selectors** in the id selector for the parent task entry element. This lets you style specific children individually. Select the **Next arrow** to continue.
10. You've added styles for the section of your user interface that will display the tasks users add. Notice that the color of the button matches the brand color and is the same as the background of your web page. You can see how this displays in the Preview panel. Select the **Next arrow** to continue.
11. That section will look better if it scrolls rather than continues to grow after the users add a certain number of tasks. Next, you set the scroll behavior of the task section with a **chained class selector**. Select the **Next arrow** to continue.
12. Under the last code, type one line of code in each field exactly as it displays in the Code pad. Then, press **Enter**.

Code pad code:

```
.tasks.overflow{  
  overflow-y: auto;  
  max-height: 300px;  
}
```

13. In the next step, you style the scrollbar in your task list section. You can use a **pseudo element** for this purpose. It styles part of an element. You can set the color of the background, foreground, and the slider of the scrollbar. Select the **Next arrow** to continue.
14. Under the last code you typed, type one line of code in each field exactly as it displays in the Code pad. Then, press **Enter**.

Code pad code:

```
.tasks::-webkit-scrollbar {  
  width: 10px;  
}  
.tasks::-webkit-scrollbar-track {  
  background: #BAE6FF;  
  border-radius: 25px;  
}  
.tasks::-webkit-scrollbar-thumb {
```

```
background: #0D75EC;  
border-radius: 25px;  
}
```

15. Nice work! Take a moment to look at the Preview pane. The scrollbar in the tasks area isn't showing yet. It will show after the user adds tasks. The scrollbar will have brand colors, and its background color will be different from the foreground color. Select the **Next arrow** to continue.

### **Continue styling the web page with CSS**

For the purpose of your project scenario, you continue to build out styles for the rest of your user interface. For example, you create styles for the tasks that users can add, the button to delete a task, and the checkmark that displays next to a completed task. You use class selectors and id selectors.

Next, you can see what some of this code looks like in your **style.css** file.

Select **X** to close this window and continue.

16. The code you added creates the individual task items. It controls the size and layout of a task item, the design of the delete button, and the behavior of each checkbox. The Preview pane is empty because this code does not add tasks. Select the **Next arrow** to continue.

You have successfully created your CSS file in VS Code. You have styles now for the overall web page and specific elements. Your CSS code improves the appearance of your page.