2. You have three layout tasks. The first will be the large text on top of the banner image in the header. Use absolute positioning to place the `<div>` with the banner—title class on top of the banner image. The exact position is not important; just try to get it approximately in the center. Also style the rest of the banner title text.
3. The next layout task is the highlights section. You will use a float to move the image within the highlights—media element to the left of the text. The highlights—buttonlink within the highlights—text element will be floated to the right. You will also use a float to move the highlights—container elements to the left.
4. The final layout task will use absolute positioning to construct the mosaic of paintings. Remember that absolute positioning is relative to the last positioned ancestor. We recommend that you use relative positioning on the mosaic container. You can then use absolute positioning for each mosaic image.

Testing

1. View **chapter07-project01.html** in the browser. It should look similar to that shown in Figure 7.55. Note that you will need a wide browser window on a desktop machine.
2. Try resizing the browser window and make it smaller. The layout will be a mess! It illustrates one of the main problems with complicated positioning layouts: they don't scale well to smaller browser windows.

PROJECT 2: Book CRM

DIFFICULTY LEVEL: Intermediate

Overview

Use the flexbox layout mode and media queries to create a responsive layout.

HANDS-ON EXERCISES

LAB 7
Project 2

Instructions

1. Open **chapter07-project02.html** in the browser. You will be modifying the CSS only.
2. Modify **styles.css** and float the `<h1>` in the header to the left and the vertical line menu image to the right.
3. Right now each card fills the entire width of the available space. Change the width of the card class to 24%. By taking less than a quarter of the available space, we will eventually be able to fit four cards on a row.
4. Now we need to use the flexbox mode. You will need to add display:flex to the cards class.
5. Change the max-width property of the `<figure>` image to 100%.
6. Modify the card class by setting its flex property to 0 1 auto. Test in browser. Set the justify-content, align-items, and flex-wrap properties appropriately in order to achieve a layout similar to that shown in Figure 7.56.

FIGURE 7.56 Completed Project 2

7. Trying resizing the browser; notice how the flex containers continue to shrink in width in order to maintain the four columns. Why four columns? Remember back in step three we set the width to 24%, so the browser is trying to maintain that rule.
8. Add a media query for screens 480 px wide and less. In it, change the card width to 100% and test. Now on a small screen, each card will fill the entire width. Also reduce the height of the header as well as the margin and padding of its heading.
9. Add a media query for tablets between 481 px and 768 px. Change the card width so two cards are displayed on each row.
10. Add a 2 second transition on the opacity property when hovering over or off of the See More span with the *button* class This will create the illusion of the span fading into (or out of) visibility. Also, add a drop shadow and a saturation filter of about 130% when hovering over any of the card images.

Testing

1. View **chapter07-project02.html** in the browser. Be sure to test at different sizes to verify the media queries work as expected (see Figure 7.56).

PROJECT 3: Share Your Travel Photos

DIFFICULTY LEVEL: Intermediate

Overview

Use the Bootstrap CSS framework (included with the start files, but you may want to instead download the most recent version) as well as modify **chapter07-project03.css** and **chapter07-project03.html** so it looks similar to that shown in Figure 7.57.

HANDS-ON EXERCISES

LAB 7
Project 3

Instructions

1. Examine **chapter07-project03.html** in the browser. You will need to add a fair bit of HTML in accordance with the Bootstrap documentation. Since you can use the various Bootstrap classes, you will need to write very little CSS (the solution shown in Figure 7.57 has just over a dozen rules defined).
2. The first step will be defining the basic structure. Figure 7.57 shows that most of the content is contained within a main row (i.e., below the navbar and above the footer) that is composed of two columns (one 2 wide, the other 10 wide). The Bootstrap grid classes (e.g., col-md-10) are shown at the top of the figure. One of the columns has a nested row within it that contains the main photo image and the data on the photo.
3. The footer contains three columns. One of these contains another nested row.
4. Figure 7.57 identifies the other Bootstrap components that are used in this project. You will need to use the online Bootstrap documentation for more information on how to use these components.