

Complete the following:

1. Using your editor, open the sb_payment_txt.html and sb_validate_txt.css files from the html07 => case2 folder. Enter your name and the date in the comment section of each file, and save them as sb_payment.html and sb_validate.css respectively.
2. Return to the sb_payment.html file in your editor. Add links to the sb_forms.css and sb_validate.css style sheet files to the document head. Add a script element to load the sb_script.js file.
3. The page will contain a form in which customers can log into their account. Directly after the sb_logo.png image, insert a form element with the ID login. Have the form use the <http://www.example.com/sb/login> via the post method.
4. Within the login form, insert the following fields and controls:
 - a. A text input box with the ID userbox for the username field. Add the placeholder text username.
 - b. A text input box with the ID pwdbox for the password field. Add the placeholder text password.
5. Next, insert a payment form. Directly below the Payment Form h1 header, insert a form element that employs action at <http://www.example.com/sb/payment> via the post method. Assign the web form the id payment.
6. Insert a field set with the id billing to the payment form. Add the legend billing information (required) to the field set.
7. Within the billing field set, add the labels and input boxes specified in Figure 7-64. Note that none of the input boxes contain placeholder text. You do not need to enclose the label or input elements within div elements. Create a data list with the ID stateList containing the two-letter abbreviations of all the states. (You can use the list of abbreviations in the sb_state.txt file.) Apply the data list to the stateBox input text box.
8. Create a data list with the ID stateList containing the two-letter abbreviations of all the states. (You can use the list of abbreviations in the sb_state.txt file.) Apply the data list to the stateBox input text box.
9. The text of the zip field should follow the `^\d{5}(-\d{4})?$/` regular expression pattern. The text of the phone field should follow the `^\d{10}$|^(\\(\d{3}\\)\\s*)?\\d{3}\\[\\s-]\\d{4}$` pattern. Note that both regular expression patterns can be found in the sb_regex.txt file.
10. Set the default value of the country field to United states.
11. Create a field set with the ID creditcard, which you use to insert credit card fields. Add the legend credit card (required).
12. Within the creditCard field set, insert another field set containing four label elements, with each label element belonging to the cardLabel class. Within each of the four label elements, insert an option button from the cCard field with the value amex, discover, master, and visa. Make cCard a required field. Follow each option button with an image element containing the image of its corresponding credit card image using the sb_amex.png, sb_discover.png, sb_master.png, and sb_visa.png files.
13. After the creditCard field set, insert a label containing the text credit card Number associated with the cardBox input control. Create an input text box with the ID cardbox for the cardNumber field. Make the field cardNumber required and have the field value follow the regular expression pattern for credit card numbers (using the regular expression in the sb_regex.txt file).

14. Create the expiration Date label associated with the monthList control.
15. Add a selection list with the ID monthList for the cardMonth field. Make the cardMonth field required. Populate the selection list with the option text “--Month--”, “January (01)”, “February (02)” and so forth up to “December (12)”. For the “--Month--”, set the field value to an empty text string. For the month options, insert the month value from 1 up to 12.
16. Add a selection list with the ID yearList for the cardYear field. Make the cardYear field required. Populate the selection list with the option text “--Year--”, “2017”, “2018”, “2019”, “2020”, and “2021”. For the “--Year--”, set the field value to an empty text string and, for the year options, insert the 4-digit year value.
17. Create a label with the text csc associated with the cscBox. Add a text input box with the ID cscbox for the required csc field. Have the csc field value follow the regular expression pattern `^\d{3}$`. Set the maximum length of the field value to 3 characters and display the placeholder text nnn.
18. After the creditCard field set, insert a button element of the submit type. Within the button, insert the `sb_button.png` inline image with the alternate text next. (Note: The text for the button is part of the button image, so the value attribute is not needed.)
19. Save your changes to the file and then go to `sb_validate.css` file in your editor to design the validation styles for the web form.
20. Within the Validation Styles section, insert the following style rules to perform inline validation:
 - a. For every input element that is not a radio type and that has the focus, change the background color to `rgb(255, 218, 165)`. (Hint: Use the attribute selector `input:not([type='radio'])` to select input elements that are not radio types.)
 - b. For every input element that is not a radio type and that has the focus with a valid value, change the background color to `rgb(215, 255, 215)` and display the image file `sb_valid.png` with no tiling in the right edge of the input box.
 - c. For every input element that is not a radio type and that has the focus with an invalid value, change the background color to `rgb(255, 245, 215)` and display the image file `sb_invalid.png` with no tiling in the right edge of the input box.
21. Save your changes to the style sheet and then open `sb_payment.html` in your browser.
22. Verify that you cannot submit the form without all required fields entered in the proper format.
23. Verify the validation checks for the credit card number by confirming that the form rejects the following credit card number `6012123456789019` (which does not follow a valid card number pattern). Further verify that the form accepts the following credit card number `6011123456789019` (which follows a valid card number pattern).

Figure 7-64

Label	Data Field	Control ID	Data Type	Required
First Name	fName	firstBox	text	yes
Last Name	lName	lastBox	text	yes
Street Address	street	streetBox	text	yes
Street Address (2)	street2	streetBox2	text	no
City	city	cityBox	text	yes
State	state	stateBox	text	yes
ZIP/Postal Code	zip	zipBox	text	no
Country	country	countryBox	text	yes
Phone	phone	phoneBox	tel	yes