Local storage is a web browser feature that lets web applications store key-value pairs persistently within a user's browser. This allows web apps to save data during one session, then retrieve it in a later page session.

In this TODO application, you'll learn how to handle form inputs, manage local storage, perform CRUD (Create, Read, Update, Delete) operations on tasks, implement event listeners, and toggle UI elements.

**Step 6**

Now, you will work on opening and closing the form modal.

In earlier projects, you learned how to add and remove classes from an element with el.classList.add() and el.classList.remove(). Another method to use with the classList property is the toggle method.

The toggle method will add the class if it is not present on the element, and remove the class if it is present on the element.

Example Code

element.classList.toggle("class-to-toggle");

Add an event listener to the openTaskFormBtn element and pass in a "click" event for the first argument and an anonymous callback function for the second argument.

Inside the callback function, use the classList.toggle() method to toggle the "hidden" class on the taskForm element.

Now you can click on the "Add new Task" button and see the form modal.

A *modal* is an element that prevents all interaction with elements outside it until the modal has been dismissed.

The HTML dialog element has a showModal() method that can be used to display a modal dialog box on a web page.

Example Code

dialogElement.showModal();

Add an event listener to the closeTaskFormBtn variable and pass in a click event for the first argument and a callback function for the second argument.

For the callback function, call the showModal() method on the confirmCloseDialog element. This will display a modal with the Discard and Cancel buttons.

**Step 8**

If the user clicks the Cancel button, you want to cancel the process and close the modal so the user can continue editing. The HTML dialog element has a close() method that can be used to close a modal dialog box on a web page.

Example Code

dialogElement.close();

Add an event listener to the cancelBtn element and pass in a click event for the first argument and a callback function for the second argument.

For the callback function, call the close() method on the confirmCloseDialog element.

# Step 10

Now that you've worked on opening and closing the modal, it's time to get the values from the input fields, save them into the taskData array, and display them on the page.

To start, add a submit event listener to your taskForm element and pass in e as the parameter of your arrow function. Inside the curly braces, use the preventDefault() method to stop the browser from refreshing the page after submitting the form.