Practical 15 JavaScript Local Storage

* **Objective:**  
  Enhance the to-do list application to save tasks using JavaScript Local  
  Storage.
* **Activities:**
* Store  
  tasks in local storage so they persist after page reloads.
* Implement  
  functionality to load tasks from local storage on page load.
* Add a clear all tasks button that also clears  
  local storage.
* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <link rel="stylesheet" type="text/css" href="pr15.css">
* <title>Todo List</title>
* </head>
* <body>
* <div class="container">
* <h1>Todo</h1>
* <form class="todo-form">
* <input type="text" class="todo-input" placeholder="Add a Todo...">
* <button type="submit" class="add-button">Add</button>
* </form>
* <ul class="todo-items">
* <!-- dummy item -->
* <li class="item" data-key="1594003133171">
* <input class="checkbox" type="checkbox">
* Go to Gym
* <button class="delete-button">X</button>
* </li>
* </ul>
* </div>
* <script type="text/javascript" src="pr15.js"></script>
* </body>
* </html>
* \* {
* padding: 0;
* margin: 0;
* }
* body {
* width: 100vw;
* min-height: 100vh;
* display: flex;
* justify-content: center;
* background: linear-gradient(#F00000, #DC281E);
* font-family: sans-serif;
* }
* button:hover {
* cursor: pointer;
* background-color: #73E831;
* }
* ul {
* list-style-type: none; /\* get rid of bullet points on side of list items \*/
* }
* /\* common style ends \*/
* /\* container \*/
* .container {
* min-width: 700px;
* display: flex;
* flex-direction: column;
* align-items: center;
* padding: 20px;
* }
* h1 {
* color: #fff;
* font-size: 3rem;
* }
* /\* todo-form \*/
* .todo-form {
* margin: 40px 0px;
* }
* .todo-input {
* width: 250px;
* border: none;
* outline: none;
* border-radius: 5px;
* padding: 10px;
* margin-right: 10px;
* font-size: 1rem;
* }
* .add-button {
* background-color: #0000ff;
* color: #fff;
* border: none;
* outline: none;
* border-radius: 5px;
* padding: 7px;
* font-size: 1.2rem;
* }
* /\* todo-form style ends \*/
* /\* todo-items \*/
* .todo-items {
* min-width: 350px;
* }
* /\* each li with class="item" \*/
* .item {
* background-color: #fff;
* padding: 10px;
* font-size: 1.1rem;
* }
* .item:first-child {
* border-top-left-radius: 7px;
* border-top-right-radius: 7px;
* }
* .item:last-child {
* border-bottom-left-radius: 7px;
* border-bottom-right-radius: 7px;
* }
* /\* item style end \*/
* .checkbox {
* margin-right: 10px;
* }
* .delete-button {
* float: right;
* background-color: #dc143c;
* border: none;
* outline: none;
* border-radius: 7px;
* padding: 2px 5px;
* margin-left: 10px;
* font-size: 1.1rem;
* font-weight: 550;
* }
* /\* applied when the todo item is checked \*/
* .checked {
* text-decoration: line-through;
* }
* /\* todo-items style ends \*/
* /\* container style ends \*/
* const todoForm = document.querySelector('.todo-form');
* const todoInput = document.querySelector('.todo-input');
* const todoItemsList = document.querySelector('.todo-items');
* let todos = [];
* todoForm.addEventListener('submit', function(event) {
* event.preventDefault();
* addTodo(todoInput.value);
* });
* function addTodo(item) {
* if (item !== '') {
* const todo = {
* id: Date.now(),
* name: item,
* completed: false
* };
* todos.push(todo);
* addToLocalStorage(todos);
* todoInput.value = '';
* }
* }
* function renderTodos(todos) {
* todoItemsList.innerHTML = '';
* todos.forEach(function(item) {
* const checked = item.completed ? 'checked' : null;
* const li = document.createElement('li');
* li.setAttribute('class', 'item');
* li.setAttribute('data-key', item.id);
* if (item.completed === true) {
* li.classList.add('checked');
* }
* li.innerHTML = `
* <input type="checkbox" class="checkbox" ${checked}>
* ${item.name}
* <button class="delete-button">X</button>
* `;
* todoItemsList.append(li);
* });
* }
* function addToLocalStorage(todos) {
* localStorage.setItem('todos', JSON.stringify(todos));
* renderTodos(todos);
* }
* function getFromLocalStorage() {
* const reference = localStorage.getItem('todos');
* if (reference) {
* todos = JSON.parse(reference);
* renderTodos(todos);
* }
* }
* function toggle(id) {
* todos.forEach(function(item) {
* if (item.id == id) {
* item.completed = !item.completed;
* }
* });
* addToLocalStorage(todos);
* }
* function deleteTodo(id) {
* todos = todos.filter(function(item) {
* return item.id != id;
* });
* addToLocalStorage(todos);
* }
* getFromLocalStorage();
* todoItemsList.addEventListener('click', function(event) {
* if (event.target.type === 'checkbox') {
* toggle(event.target.parentElement.getAttribute('data-key'));
* }
* if (event.target.classList.contains('delete-button')) {
* deleteTodo(event.target.parentElement.getAttribute('data-key'));
* }
* });

