**Date : 03/04/2024**

**Name : V.S.Pradip**

**Email Id :** [**pradipvengal2001@gmail.com**](mailto:pradipvengal2001@gmail.com)

**Title :**

To-Do Application Development

**Description :**

Our first objective is to create a To-Do Application development. In this application , It

will be help us to add the events , edit the events and delete the events . It will store the

events in the form of array with the help of local storage. Once the event has been added, it

will be stored in this array and when the event has been deleted, it would be removed from the

array. This task includes the proficiency knowledge of HTML , CSS , JavaScript and Local

storage concepts. The application uses local storage to store the to-do list items. This allows the

data to persist even when the user reloads or revisits the page. We also displayed the current

day and date near to the event bar.

**Steps :**

**Step 1** : Create an HTML web page named index.html to create a layout and use some tags such as heading tags to create headings.

**Step 2** : Provide some global attributes to the division tags and heading tags , such as class attribute to specify the property to the HTML elements.

**Step 3** : Create a CSS file named index.css to provide the design the layout. Use class selectors and element selectors to select the HTML elements and are used to provide the properties to the HTML elements.

**Step 4** : The CSS file (index.css) can be linked by the link tag with src as a local attribute to provide the file name with rel as a local attribute to provide stylesheet as a value inside the head tag of the HTML page.

**Step 5** : Create a JavaScript named index.js to provide the functionality to the web page such as adding the events , edit the events and deleting the events.

**Step 6** : Create an EventListeners for the functions like **activateSaveListeners** , **activateDeleteListeners** , **activateEditListeners** and **activateCancelListeners** to listen for user interactions like clicks. These listeners trigger specific functions to perform actions like adding, editing, deleting, or canceling tasks.

**Challenges and Solutions :**

Several challenges may arise when developing a To-Do Application page. Here are some potential challenges and how to address them:

1. Data Persistence: Challenge: Storing tasks locally so they persist even when the page is reloaded or closed.
   * Solution: Utilize browser local storage (as demonstrated in the provided JavaScript code) to store task data. This allows tasks to be retrieved and displayed even after refreshing the page.
2. Responsive Design: Challenge: Ensuring the application works well on various screen sizes and devices.
   * Solution: Implement responsive design techniques using CSS media queries to adjust the layout and styling based on the device's screen size. Test the application on different devices and browsers to ensure compatibility.

1. User Experience (UX): Challenge: Providing a user-friendly interface with intuitive interactions.

* Solution: Design the interface to be clean and easy to understand. Add visual cues such as hover effects, transitions, and animations to enhance the user experience. Use descriptive button labels and icons for clear actions (e.g., "Add Task", "Delete", "Edit").

1. Editing Tasks: Challenge: Enabling users to edit existing tasks efficiently.

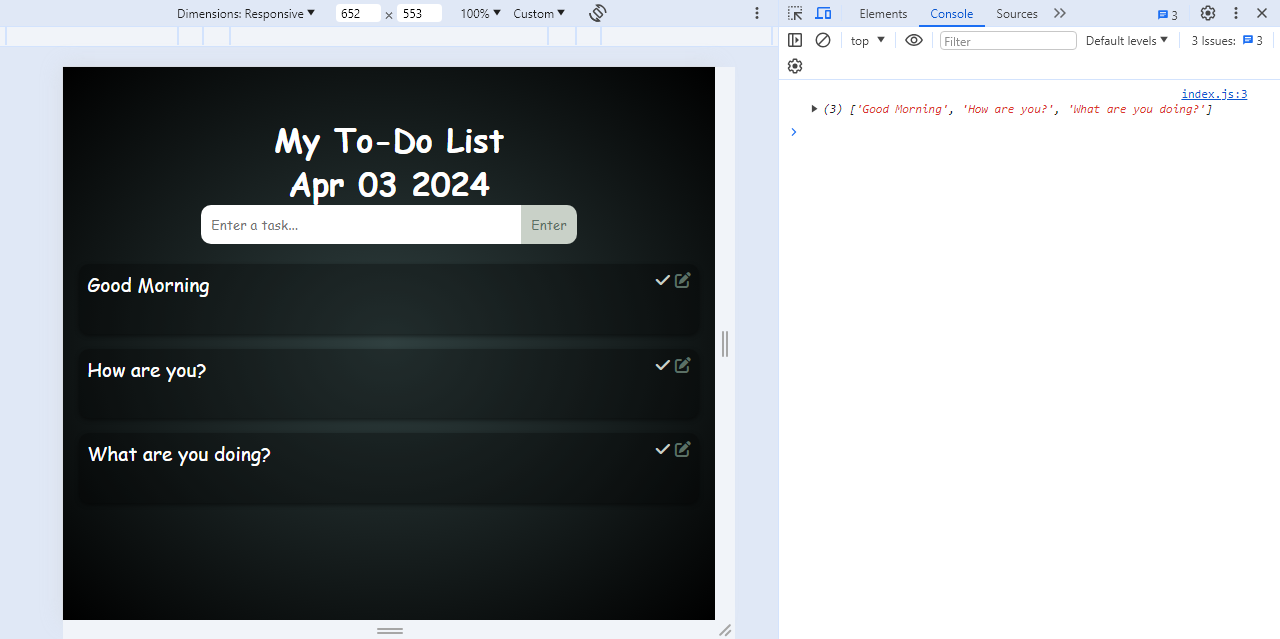
* Solution: Implement an edit feature where users can click on a task to enable editing. Display a text input or textarea with the task content pre-filled, allowing users to make changes. Provide save and cancel buttons for confirming or discarding edits.

1. Accessibility: Challenge: Ensuring the application is accessible to users with disabilities.

* Solution: Use semantic HTML elements and provide appropriate attributes to make the application usable with screen readers and assistive technologies. Ensure keyboard navigation and focus management are implemented correctly. Test the application with accessibility tools and follow best practices for accessibility.

1. Performance: Challenge: Optimizing the application for performance, especially when dealing with large datasets.

* Solution: Implement efficient algorithms for rendering tasks and handling user interactions. Minimize DOM manipulation and use event delegation where possible to improve performance. Consider implementing pagination or lazy loading for large task lists to avoid rendering all tasks at once.

**Output:**