

Practica 1 Parcial 2

Alumno:

- CIAU DZUL JOSUE GEOVANI

Escuela: Universidad Tecnológica Metropolitana

Asignatura: Aplicaciones Web

Maestro: Ing. Diego May

3er Cuatrimestre

Grupo: 3D-DSM

Parcial II

15/06/2024

Introducción

En este proyecto se desarrolló un sistema de gestión de tareas pendientes utilizando tecnologías web como HTML, CSS y JavaScript, con la integración de Bootstrap para el diseño y jQuery para facilitar la manipulación del DOM y la interacción con los elementos de la página. El objetivo principal de este sistema es proporcionar una herramienta eficiente y fácil de usar para la administración de tareas, permitiendo a los usuarios agregar, visualizar, filtrar, buscar, completar y eliminar tareas.

El sistema se basa en un formulario que captura la información de nuevas tareas, incluyendo el nombre de la tarea, fecha de inicio, fecha de fin y responsable. Estas tareas se almacenan en el localStorage del navegador, lo que permite persistir los datos incluso después de cerrar y reabrir la página. Las tareas se muestran en una tabla que se actualiza dinámicamente y se pueden filtrar según su estado (todas, completadas o vencidas), además de permitir la búsqueda por nombre de tarea.

Para mejorar la experiencia del usuario, se implementaron alertas que proporcionan retroalimentación en diversas situaciones, como al agregar una tarea exitosamente, al intentar completar una tarea vencida o al ingresar fechas incorrectas. También se integraron componentes de datepicker para facilitar la selección de fechas en el formulario.

Este proyecto combina varias tecnologías web para ofrecer una solución completa y funcional para la gestión de tareas, destacando por su interfaz interactiva y su facilidad de uso.

Conclusión

En este proyecto se implementó un sistema de gestión de tareas pendientes utilizando HTML, CSS y JavaScript, con la integración de Bootstrap para el diseño y jQuery para facilitar la manipulación del DOM y la interacción con los elementos de la página.

Funcionalidades Principales:

Formulario para Agregar Tareas:

- Se captura la información de nuevas tareas a través de un formulario.
- Los campos incluyen el nombre de la tarea, fecha de inicio, fecha de fin y responsable.
- Al enviar el formulario, se previene el comportamiento predeterminado de envío mediante `event.preventDefault()`.
- Se valida que la fecha de inicio no sea posterior a la fecha de fin. Si esto ocurre, se muestra una alerta al usuario.

Almacenamiento de Tareas:

- Las tareas se almacenan en el localStorage del navegador en formato JSON.
- Cada tarea incluye información sobre su estado de completitud y vencimiento.

Visualización y Filtros de Tareas:

- Las tareas se muestran en una tabla que se actualiza dinámicamente.

- Se pueden filtrar las tareas por todas, completadas y vencidas.
- Además, se incluye una función de búsqueda por nombre de tarea.

Interacciones con las Tareas:

- Completar Tarea: Las tareas se pueden marcar como completadas, siempre y cuando la fecha de fin no haya pasado. Si la fecha de fin ya pasó, se muestra una alerta.
- Desmarcar Tarea: Las tareas completadas pueden ser desmarcadas.
- Eliminar Tarea: Las tareas pueden ser eliminadas, previa confirmación del usuario.

Alertas de Usuario:

- Se utilizan alertas para proporcionar retroalimentación al usuario, como cuando se agrega una tarea exitosamente, se intenta completar una tarea vencida o se ingresan fechas incorrectas.

Inicialización y Carga de Datos:

- Al cargar la página, se muestra la lista de tareas existentes.
- Se inicializan los componentes de datepicker para facilitar la selección de fechas en el formulario.

Código HTML y JavaScript:

- El código HTML define la estructura de la página, incluyendo el formulario para agregar tareas y la tabla para mostrar la lista de tareas. El JavaScript gestiona la lógica de la aplicación, manejando eventos, validaciones, almacenamiento en localStorage y actualización dinámica del DOM.

Este proyecto proporciona una solución completa para la gestión de tareas, permitiendo a los usuarios agregar, visualizar, filtrar, buscar, completar y eliminar tareas de manera eficiente. La combinación de tecnologías como HTML, CSS, JavaScript, Bootstrap y jQuery facilita la creación de una interfaz