Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="/css/style.css">

</head>

<body>

    <div class="main">

        <form id="form-addtarea">

            <label for="nombre">Nombre:</label>

            <input type="text" id="nombre" name="nombre" required>

            <hr>

            <label for="descripcion">Descripción:</label>

            <textarea id="descripcion" name="descripcion" rows="4" required></textarea>

            <input type="submit" value="Nueva Tarea">

        </form>

        <div class="tareas"></div> <!-- Contenedor para las tareas -->

    </div>

    <script type="module" src="./js/tareas.class.js"></script>

    <script type="module" src="./js/app.js"></script>

</body>

</html>

Style.css

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

.main {

    display: flex;

    flex-direction: column;

    align-items: center;

    min-height: 100vh; /\* Ajustar la altura mínima \*/

    padding: 20px; /\* Añadir padding para asegurar espacio alrededor \*/

}

#form-addtarea {

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    border: 1px solid #ccc;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    width: 100%; /\* Asegurarse de que ocupe todo el ancho disponible \*/

    max-width: 600px; /\* Limitar el ancho máximo \*/

    box-sizing: border-box; /\* Asegurarse de que el padding se incluya en el tamaño total \*/

}

#form-addtarea > \* {

    margin: 5px;

    width: calc(100% - 10px); /\* Asegurarse de que los elementos hijos no excedan el ancho del contenedor \*/

    box-sizing: border-box; /\* Asegurarse de que el padding y el margin se incluyan en el tamaño total \*/

}

#form-addtarea input[type="text"],

#form-addtarea textarea {

    padding: 8px;

    border: 1px solid #ccc;

    border-radius: 4px;

    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1);

}

#form-addtarea input[type="submit"] {

    padding: 10px 20px;

    border: none;

    border-radius: 4px;

    background-color: #007bff;

    color: #fff;

    cursor: pointer;

    transition: background-color 0.3s;

}

#form-addtarea input[type="submit"]:hover {

    background-color: #0056b3;

}

.tareas {

    width: 100%;

    margin-top: 20px;

    display: flex;

    overflow-y: hidden;

    overflow-x: scroll;

}

.tarjeta {

    background-color: #fff;

    border: 1px solid #ccc;

    border-radius: 8px;

    padding: 15px;

    margin-bottom: 10px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

    transition: transform 0.2s;

    margin: 1%;

}

.tarjeta:hover {

    transform: scale(1.02);

}

.terminada {

    text-decoration: line-through;

    filter: blur(1px);

    color: #888;

}

Tareas.class.js

export class Tarea {

    static autoin = 1;

    constructor(nombre, descripcion) {

        this.nombre = nombre;

        this.descripcion = descripcion;

        this.id = Tarea.autoin++;

        this.completada = false; // Inicialmente no completada

        this.eliminada = false; // Inicialmente no eliminada

    }

    crearElemento(tag, className, value = '', disabled = true) {

        let element = document.createElement(tag);

        if (className) element.classList.add(className);

        if (tag === 'input' && value) {

            element.type = 'text';

            element.value = value;

            element.disabled = disabled;

        } else if (tag === 'textarea' && value) {

            element.value = value;

            element.disabled = disabled;

        }

        return element;

    }

    mostrarTarea(arrayTareas) {

        if (this.eliminada) return; // No mostrar si está eliminada

        let divTarjeta = document.createElement('div');

        divTarjeta.classList.add('tarjeta');

        if (this.completada) divTarjeta.classList.add('terminada');

        let inputNombreTarea = this.crearElemento('input', 'inputNombre', this.nombre);

        let textareaDescripcionTarea = this.crearElemento('textarea', 'textareaDescripcion', this.descripcion);

        // Crear checkbox y etiqueta para marcar como completada

        let divCompletada = document.createElement('div');

        divCompletada.classList.add('divCompletada');

        let checkboxCompletada = document.createElement('input');

        checkboxCompletada.type = 'checkbox';

        checkboxCompletada.checked = this.completada;

        let etiquetaCompletada = document.createElement('span');

        etiquetaCompletada.textContent = this.completada ? 'Desmarcar' : 'Marcar como hecha';

        checkboxCompletada.addEventListener('change', () => {

            this.completada = checkboxCompletada.checked;

            etiquetaCompletada.textContent = this.completada ? 'Desmarcar' : 'Marcar como hecha';

            divTarjeta.classList.toggle('terminada', this.completada);

            inputNombreTarea.classList.toggle('terminada', this.completada);

            textareaDescripcionTarea.classList.toggle('terminada', this.completada);

        });

        divCompletada.append(checkboxCompletada, etiquetaCompletada);

        // Crear botón "Eliminar"

        let botonEliminar = document.createElement('button');

        botonEliminar.classList.add('botonEliminar');

        botonEliminar.textContent = 'Eliminar';

        botonEliminar.addEventListener('click', () => this.borrartarea(divTarjeta));

        // Crear botón "Modificar"

        let botonModificar = document.createElement('button');

        botonModificar.classList.add('botonModificar');

        botonModificar.textContent = 'Modificar';

        botonModificar.addEventListener('click', () => {

            const editando = botonModificar.textContent === 'Guardar';

            [inputNombreTarea, textareaDescripcionTarea].forEach(input => input.disabled = editando);

            if (editando) {

                this.nombre = inputNombreTarea.value;

                this.descripcion = textareaDescripcionTarea.value;

                alert('Tarea modificada');

            }

            botonModificar.textContent = editando ? 'Modificar' : 'Guardar';

        });

        divTarjeta.append(inputNombreTarea, textareaDescripcionTarea, divCompletada, botonEliminar, botonModificar);

        document.querySelector('.tareas').appendChild(divTarjeta);

    }

    borrartarea(divTarjeta) {

        this.eliminada = true;

        divTarjeta.style.display = 'none';

    }

}

App.js

import { Tarea } from "./tareas.class.js";

document.addEventListener('DOMContentLoaded', () => {

    const arrayTareas = [];

    document.getElementById('form-addtarea').onsubmit = function(event) {

        event.preventDefault(); // Prevenir la recarga de la página

        console.log('Formulario de nueva tarea enviado');

        // Obtener los valores de los campos de entrada

        let nombreTarea = document.getElementById('nombre').value;

        let descripcionTarea = document.getElementById('descripcion').value;

        // Crear una nueva instancia de Tarea con el nombre y la descripción

        let nuevaTarea = new Tarea(nombreTarea, descripcionTarea);

        // Añadir la nueva tarea al array de tareas

        arrayTareas.push(nuevaTarea);

        // Mostrar todas las tareas en la interfaz

        document.querySelector('.tareas').innerHTML = ''; // Limpiar tareas previas

        arrayTareas.forEach(element => {

            element.mostrarTarea(); // Mostrar cada tarea

        });

        // Vaciar los campos de entrada

        document.getElementById('nombre').value = '';

        document.getElementById('descripcion').value = '';

        // Mostrar el array de tareas en la consola

        console.log(arrayTareas);

    };

});