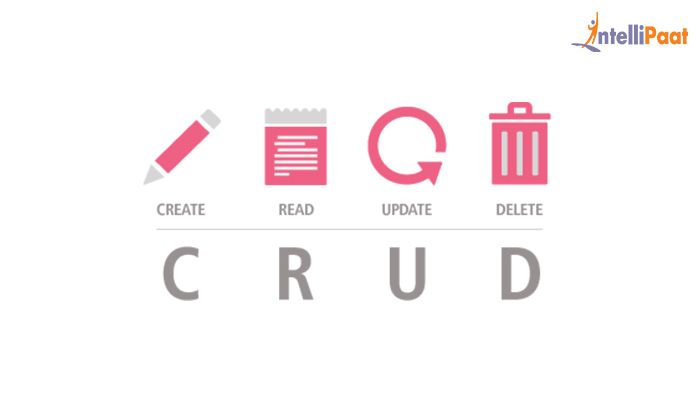
16/02/2022

|  |
| --- |
| [Tapez le sous-titre du document] | dydysaliou |



|  |
| --- |
| C R U D (créer, lire, mettre à jour, supprimer) |

Table des matières

[I. Introduction 3](#_Toc96273590)

[II. Code Html 3](#_Toc96273591)

[III. Code Css 4](#_Toc96273592)

[IV. Create, Read, Update, Delete 5](#_Toc96273593)

[V. Affichage à l’écran 7](#_Toc96273594)

[1. Quand il y’a aucun patient en attente 7](#_Toc96273595)

[2. Ajout de patients 7](#_Toc96273596)

[3. Modification des informations d’un patient 8](#_Toc96273597)

[4. Suppression 8](#_Toc96273598)

# Introduction

**CRUD** (create, read, update, delete) (créer, lire, mettre à jour, supprimer) est un acronyme pour les façons dont on peut agir sur des données à stocker. CRUD est un moyen mnémotechnique pour les quatre fonctions de base du stockage persistant. Il fait généralement référence aux opérations effectuées dans une base de données, mais peut également s'appliquer aux fonctions de niveau supérieur d'une application telles que les suppressions logicielles lorsque les données ne sont pas supprimées mais marquées comme supprimées via un état.

Nous allons mettre en place un page qui permettra d’ajouter un patient, modifier ses informations grâce à des codes HTML, CSS et JavaScript.

# Code Html

<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>CRUD</title>

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

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">

</head>

<body>

    <div class="title">

        <h1>C R U D (créer, lire, mettre à jour, supprimer)</h1>

        <p>Liste des patients:</p>

    </div>

<div class="form">

    <form action="javascript:void(0);" method="POST" onsubmit="app.Add()"> <br>

    <input type="text" id="add-patient" placeholder="Ajouter un patient">

    <!-- <input type="text" id="add-patient1" placeholder="Ajouter le prenom du patient"> -->

    <br> <br>

    <input type="submit" value="Ajouter" class="btn btn-primary">

    </form>

    <p>Vous avez : </p>

    <!-- pour compter le nombre de patients en attente -->

    <p id="counter"></p>

    <table>

    <tr class="to-do">

        <th><h3> Les patients en attente : </h3></th>

    </tr>

        <tbody id="tasks">

        </tbody>

    </table>

</div>

<div id="edit-box" role="aria-hidden">

    <form action="javascript:void(0);" method="POST" id="save-edit">

      <input type="text" id="edit-todo">

      <input type="submit" value="Enregitrer" class="btn btn-success"/> <a onclick="CloseInput()" aria-label="Close">&#10006;</a>

    </form>

  </div>

<script src="script.js"></script>

</body>

</html>

# Code CSS

\*{

  align-items: center;

}

body{

  background-color: green;

}

  .title{

    text-align: center;

    margin-top: 25px;

  }

  .form{

      margin: auto;

      max-width:400px;

      text-align: center

  }

  input[type='submit'], button, [aria-label]{

    cursor: pointer;

  }

  #edit-box{

    display: none;

    max-width: 80%;

    margin: auto;

    margin-top: 10px;

  }

  td{

      background-color: aliceblue;

      padding-right: 10px;

      text-align: left;

  }

  .to-do{

      background: transparent !important;

  }

# Create, Read, Update, Delete

var app = new function() {

  this.el = document.getElementById('tasks');

  this.tasks = [];

  this.FetchAll = function() {

    var data = '';

    if (this.tasks.length > 0) {

      for (i = 0; i < this.tasks.length; i++) {

        data += '<tr>';

        data += '<td>'+(i+1)+". " + " 🤒 " + this.tasks[i] + '</td>';

        //creation des boutons modifier et supprimer

        data += '<td><button onclick="app.Edit(' + i + ')"  class="btn btn-warning">Modifier</button></td>';

        data += '<td><button onclick="app.Delete(' + i + ')"  class="btn btn-danger">Supprimer</button></td>';

        data += '</tr>';

      }

    }

    this.Count(this.tasks.length);

    return this.el.innerHTML = data;

  };

  //Ajout de patient

  this.Add = function () {

    el = document.getElementById('add-patient');

    // Get the value

    var task = el.value;

    if (task) {

      // Add the new value

      this.tasks.push(task.trim());

      // Reset input value

      el.value = '';

      // Dislay the new list

      this.FetchAll();

    }

  };

  //modification des informations d'un patient

  this.Edit = function (item) {

    var el = document.getElementById('edit-todo');

    // Affichage des valeurs

    el.value = this.tasks[item];

    // champ d'affichage

    document.getElementById('edit-box').style.display = 'block';

    self = this;

    document.getElementById('save-edit').onsubmit = function() {

      // obtenir la valeur à enregistrer

      var task = el.value;

      if (task) {

        // edition de la valeur

        self.tasks.splice(item, 1, task.trim());

        // affichage de la nouvelle liste

        self.FetchAll();

        //Masquer le champ

        CloseInput();

      }

    }

  };

  //suppression d'un patient

  this.Delete = function (item) {

    // suppression de sa ligne

    this.tasks.splice(item, 1);

    // Afficher la nouvelle liste apres suppression

    this.FetchAll();

  };

  // if (onclick==this.Delete){

  //   confirm("Confirmez-vous la suppression des informations de ce patients ?") ;}

  ///affichage du nombre de patients restants

  this.Count = function(data) {

    var el   = document.getElementById('counter');

    var name = 'patients';

//s'il y'a un patient sur la liste

    if (data) {

        if(data ==1){

            name = "patient(s)"

        }

        // s'il y'a plus de 1 an patient,on incremente le nombre

      el.innerHTML = "🤒" +data + ' ' + name ;

    }

    //sinon s'il n'y a plus de patients ,on affiche pas de patient restant

    else {

      el.innerHTML = 'Pas de  ' + "🤒 " + name + " en attente ";

    }

  };

}

app.FetchAll();

function CloseInput() {

  document.getElementById('edit-box').style.display = 'none';

}

# Affichage à l’écran

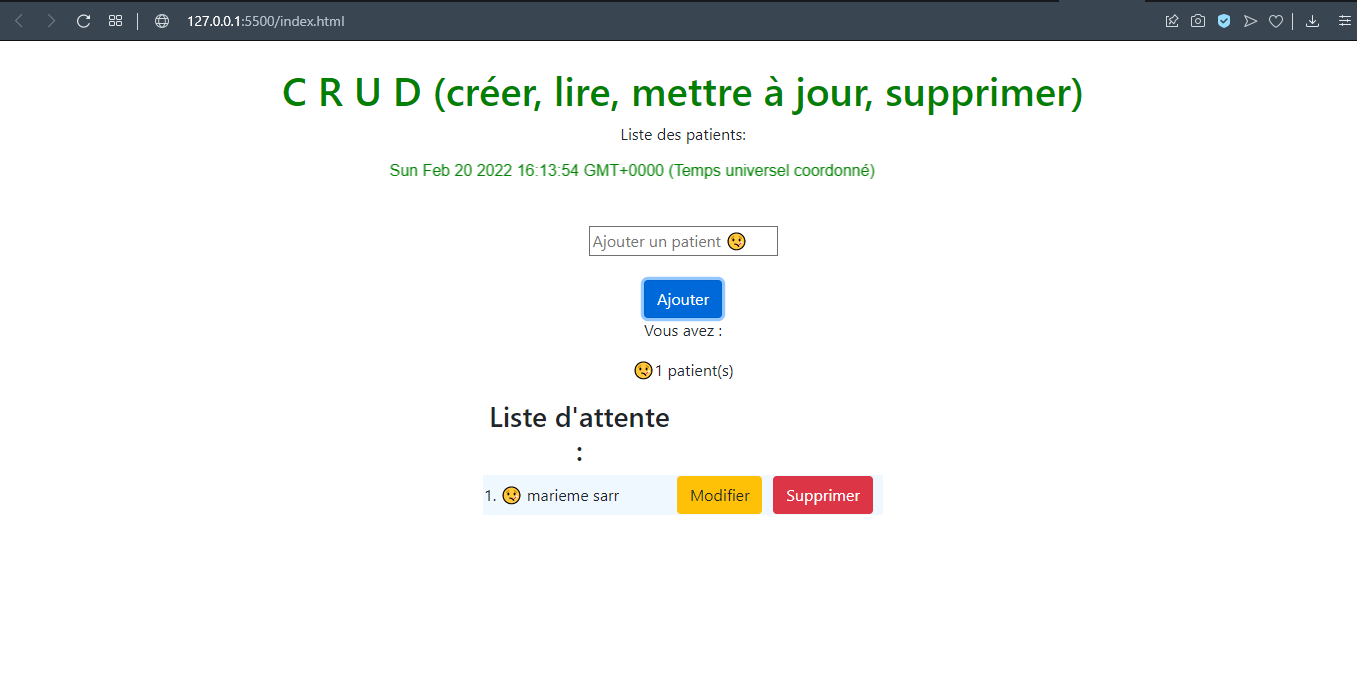
## Quand il y’a aucun patient en attente

**Lorsque la liste des patients est vide, il sera affiché *pas de patients en attente* et la liste d’attente sera vide.**



## Ajout de patients

**Pour ajouter un patient, saisissez son nom dans le champ et cliquez sur le bouton ajouter. Et voila le nouveau patient figure sur la liste. Et le nombre de patients en attente vous sera notifié.**



## Modification des informations d’un patient

**Pour modifier les informations d’un patient, rien de difficile ; Cliquez juste su le bouton *modifier* et insérez la nouvelle valeur, et appuyez sur *enregistrer*. Si vous ne voulez plus modifier, appuyez sur la *croix*.**



## ****Suppression****

**Tout comme la modification, pour supprimer il suffit juste de cliquer sur le bouton supprimer. Et la nouvelle liste sera mise à jour et affichée.**