CURD API USING JAVASCRIPT

```
<!DOCTYPE html>
<html>
<head>
    <h1 style="background-color:powderblue;"><center>CRUD API using JavaScript</center></h1>

    <style>
        table
        {
            width: 100%;
            font: 17px Calibri;
        }
        table, th, td
        {
            border: solid 1px #DDD;
            border-collapse: collapse;
            padding: 2px 3px;
            text-align: center;
        }

        input[type='button']
        {
            font: 15px Calibri;
            cursor: pointer;
            border: none;
            color: black;
        }

        input[type='text'], select
        {
            font: 17px Calibri;
            text-align: center;
            border: solid 1px #CCC;
            width: auto;
            padding: 2px 3px;
        }
    </style>
</head>
<body>
    <div id="container" style="width:700px;">
    </div>
</body>

<script>
    var crudApp = new function () {
        // AN ARRAY OF JSON OBJECTS WITH VALUES.
        this.myBooks = [
            { ID: '1', Book_Name: 'Artificial Intelligence', Category: 'Technology', Price: 150 },
            { ID: '2', Book_Name: 'Python', Category: 'Programming', Price: 200},
            { ID: '3', Book_Name: 'Infinite Powers', Category: 'Science', Price: 250 },
              { ID: '4', Book_Name: 'Database', Category:'Programming', Price:300}
        ]

        this.category = ['Business', 'Computers', 'Programming', 'Science','Technology'];
        this.col = [];

        this.createTable = function () {

            // EXTRACT VALUE FOR TABLE HEADER.
            for (var i = 0; i < this.myBooks.length; i++) {
                for (var key in this.myBooks[i]) {
                    if (this.col.indexOf(key) === -1) {
                        this.col.push(key);
                    }
                }
            }


            // CREATE A TABLE.
            var table = document.createElement('table');
            table.setAttribute('id', 'booksTable');     // SET TABLE ID.


            var tr = table.insertRow(-1);               // CREATE A ROW (FOR
HEADER).


            for (var h = 0; h < this.col.length; h++) {
                // ADD TABLE HEADER.
                var th = document.createElement('th');
                th.innerHTML = this.col[h].replace('_', ' ');
                tr.appendChild(th);
            }


            // ADD ROWS USING JSON DATA.
            for (var i = 0; i < this.myBooks.length; i++) {


                tr = table.insertRow(-1);           // CREATE A NEW ROW.


                for (var j = 0; j < this.col.length; j++) {
                    var tabCell = tr.insertCell(-1);
```

```
          tabCell.innerHTML = this.myBooks[i][this.col[j]];
        }


        // DYNAMICALLY CREATE AND ADD ELEMENTS TO TABLE
CELLS WITH EVENTS.


        this.td = document.createElement('td');


        // *** CANCEL OPTION.
        tr.appendChild(this.td);
        var lblCancel = document.createElement('label');
        lblCancel.innerHTML = '✖';
        lblCancel.setAttribute('onclick', 'crudApp.Cancel(this)');
        lblCancel.setAttribute('style', 'display:none;');
        lblCancel.setAttribute('title', 'Cancel');
        lblCancel.setAttribute('id', 'lbl' + i);
        this.td.appendChild(lblCancel);


        // *** SAVE.
        tr.appendChild(this.td);
        var btSave = document.createElement('input');


        btSave.setAttribute('type', 'button');     // SET ATTRIBUTES.
        btSave.setAttribute('value', 'Save');

        btSave.setAttribute('id', 'Save' + i);

        btSave.setAttribute('style', 'display:none;');

        btSave.setAttribute('onclick', 'crudApp.Save(this)');       // ADD THE BUTTON's 'onclick' EVENT.

        this.td.appendChild(btSave);


        // *** UPDATE.

        tr.appendChild(this.td);

        var btUpdate = document.createElement('input');


        btUpdate.setAttribute('type', 'button');    // SET ATTRIBUTES.

        btUpdate.setAttribute('value', 'Update');

        btUpdate.setAttribute('id', 'Edit' + i);

        btUpdate.setAttribute('style', 'background-color:#44CCEB;');

        btUpdate.setAttribute('onclick', 'crudApp.Update(this)');   // ADD THE BUTTON's 'onclick' EVENT.

        this.td.appendChild(btUpdate);


        // *** DELETE.

        this.td = document.createElement('th');

        tr.appendChild(this.td);

        var btDelete = document.createElement('input');
```

```
                btDelete.setAttribute('type', 'button');    // SET INPUT ATTRIBUTE.
                btDelete.setAttribute('value', 'Delete');
                btDelete.setAttribute('style', 'background-color:#ED5650;');
                btDelete.setAttribute('onclick', 'crudApp.Delete(this)');   // ADD THE BUTTON's 'onclick' EVENT.
                this.td.appendChild(btDelete);
            }



            // ADD A ROW AT THE END WITH BLANK TEXTBOXES AND A DROPDOWN LIST (FOR NEW ENTRY).

            tr = table.insertRow(-1);           // CREATE THE LAST ROW.

            for (var j = 0; j < this.col.length; j++) {
                var newCell = tr.insertCell(-1);
                if (j >= 1) {

                    if (j == 2) {   // WE'LL ADD A DROPDOWN LIST AT THE SECOND COLUMN (FOR Category).

                        var select = document.createElement('select');      // CREATE AND ADD A DROPDOWN LIST.
                        select.innerHTML = '<option value=""></option>';
                        for (k = 0; k < this.category.length; k++) {
                            select.innerHTML = select.innerHTML +
                                '<option value="' + this.category[k] + '">' + this.category[k] + '</option>';
                        }
                        newCell.appendChild(select);
                    }
                    else {
                        var tBox = document.createElement('input');          // CREATE AND ADD A TEXTBOX.
                        tBox.setAttribute('type', 'text');
                        tBox.setAttribute('value', '');
                        newCell.appendChild(tBox);
                    }
                }
            }

            this.td = document.createElement('td');
            tr.appendChild(this.td);

            var btNew = document.createElement('input');

            btNew.setAttribute('type', 'button');       // SET ATTRIBUTES.
```

```
            btNew.setAttribute('value', 'Create');
            btNew.setAttribute('id', 'New' + i);
            btNew.setAttribute('style', 'background-color:#207DD1;');
            btNew.setAttribute('onclick', 'crudApp.CreateNew(this)');       // ADD THE BUTTON's 'onclick' EVENT.
            this.td.appendChild(btNew);

            var div = document.getElementById('container');
            div.innerHTML = '';
            div.appendChild(table);    // ADD THE TABLE TO THE WEB PAGE.
        };

        // ****** OPERATIONS START.

        // CANCEL.
        this.Cancel = function (oButton) {

            // HIDE THIS BUTTON.
            oButton.setAttribute('style', 'display:none; float:none;');

            var activeRow = oButton.parentNode.parentNode.rowIndex;

            // HIDE THE SAVE BUTTON.
            var btSave = document.getElementById('Save' + (activeRow - 1));
            btSave.setAttribute('style', 'display:none;');

            // SHOW THE UPDATE BUTTON AGAIN.
            var btUpdate = document.getElementById('Edit' + (activeRow - 1));
            btUpdate.setAttribute('style', 'display:block; margin:0 auto; background-color:#44CCEB;');

            var tab = document.getElementById('booksTable').rows[activeRow];

            for (i = 0; i < this.col.length; i++) {
                var td = tab.getElementsByTagName("td")[i];
                td.innerHTML = this.myBooks[(activeRow - 1)][this.col[i]];
            }
        }


        // EDIT DATA.
        this.Update = function (oButton) {
            var activeRow = oButton.parentNode.parentNode.rowIndex;
            var tab = document.getElementById('booksTable').rows[activeRow];

            // SHOW A DROPDOWN LIST WITH A LIST OF CATEGORIES.
        for (i = 1; i < 4; i++) {
            if (i == 2) {
                var td = tab.getElementsByTagName("td")[i];
                var ele = document.createElement('select');      // DROPDOWN LIST.
                ele.innerHTML = '<option value="' + td.innerText + '">' +
td.innerText + '</option>';
            for (k = 0; k < this.category.length; k++) {
                ele.innerHTML = ele.innerHTML +
                    '<option value="' + this.category[k] + '">' + this.category[k] +
'</option>';
            }
            td.innerText = '';
            td.appendChild(ele);
        }
        else {
            var td = tab.getElementsByTagName("td")[i];
            var ele = document.createElement('input');      // TEXTBOX.
            ele.setAttribute('type', 'text');
            ele.setAttribute('value', td.innerText);
            td.innerText = '';
            td.appendChild(ele);
        }
    }
```

```
        var lblCancel = document.getElementById('lbl' + (activeRow - 1));
        lblCancel.setAttribute('style', 'cursor:pointer; display:block; width:20px; float:left; position: absolute;');

        var btSave = document.getElementById('Save' + (activeRow - 1));
        btSave.setAttribute('style', 'display:block; margin-left:30px; float:left; background-color:#2DBF64;');

        // HIDE THIS BUTTON.
        oButton.setAttribute('style', 'display:none;');
    };


    // DELETE DATA.
    this.Delete = function (oButton) {
        var activeRow = oButton.parentNode.parentNode.rowIndex;
        this.myBooks.splice((activeRow - 1), 1);    // DELETE THE ACTIVE ROW.
        this.createTable();                         // REFRESH THE TABLE.
    };

    // SAVE DATA.
    this.Save = function (oButton) {
        var activeRow = oButton.parentNode.parentNode.rowIndex;
        var tab = document.getElementById('booksTable').rows[activeRow];

        // UPDATE myBooks ARRAY WITH VALUES.
        for (i = 1; i < this.col.length; i++) {
            var td = tab.getElementsByTagName("td")[i];
            if (td.childNodes[0].getAttribute('type') == 'text' || td.childNodes[0].tagName == 'SELECT') {  // CHECK IF ELEMENT IS A TEXTBOX OR SELECT.
                this.myBooks[(activeRow - 1)][this.col[i]] = td.childNodes[0].value;      // SAVE THE VALUE.
            }
        }
        this.createTable();     // REFRESH THE TABLE.
    }

    // CREATE NEW.
    this.CreateNew = function (oButton) {
        var activeRow = oButton.parentNode.parentNode.rowIndex;
        var tab = document.getElementById('booksTable').rows[activeRow];
        var obj = {};

        // ADD NEW VALUE TO myBooks ARRAY.
        for (i = 1; i < this.col.length; i++) {
            var td = tab.getElementsByTagName("td")[i];
            if (td.childNodes[0].getAttribute('type') == 'text' || td.childNodes[0].tagName == 'SELECT') {      // CHECK IF ELEMENT IS A TEXTBOX OR SELECT.
                var txtVal = td.childNodes[0].value;
                if (txtVal != '') {
                    obj[this.col[i]] = txtVal.trim();
                }
                else {
                    obj = '';
                    alert('all fields are compulsory');
                    break;
                }
            }
        }
        obj[this.col[0]] = this.myBooks.length + 1;     // NEW ID.

        if (Object.keys(obj).length > 0) {      // CHECK IF OBJECT IS NOT EMPTY.
            this.myBooks.push(obj);             // PUSH (ADD) DATA TO THE JSON ARRAY.
            this.createTable();                 // REFRESH THE TABLE.
        }
    }
```

```
        // ****** OPERATIONS END.
    }

    crudApp.createTable();
</script>
</html>
```