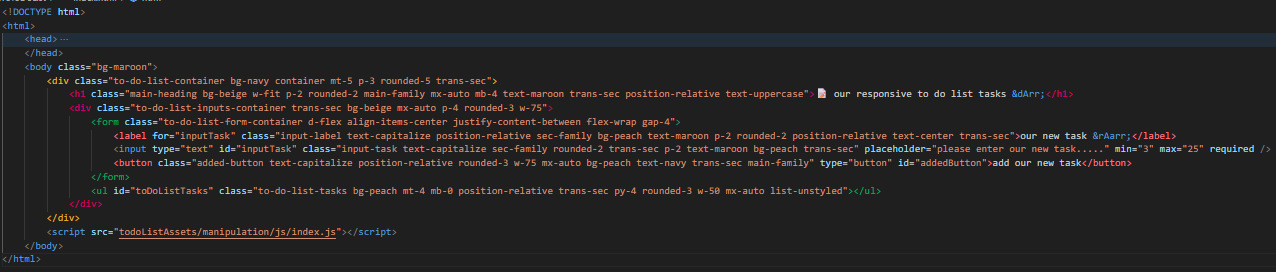
***\*\*The Responsive To Do List Tasks Project Documentation:***

***A)\*\*HTML Language:***

***\*\*It Is The Main Language To Build The Structure Of The Responsive To Do List Tasks Project By Using Different HTML Elements Like HTML Semantic Elements To Make The Responsive To Do List Tasks Project SEO Friendly , And It Is Very Useful In The Web Accessibility Of The Responsive To Do List Tasks Project.***

******

***B)\*\*CSS Language:***

***\*\*It Is The Main Language To Make The Styling , Or Decorations Of The Responsive To Do List Tasks Project By Using CSS Language , And CSS Library Like BootStrapCSS Library To Make The Media Queries (Responsive Design , And FelxBox) In The Responsive To Do List Tasks Project.***

***C)\*\*JS Language:***

***\*\*It Is The Main Language To Make The Manipulation , Or Logical Operations Of The Responsive To Do List Project , And There Are Many Features In The Responsive To Do List Tasks Project Like:***

***1)\*\*At The Frist Step , You Should Get The Added Button From The HTML Document File By Using: document.getElementById(“addedButton”).***

***2)\*\*At The Second Step , You Should Create The Main Function To Add Tasks In The To Do List That Contains All Tasks From The User Of The Responsive To Do List Tasks Project By Using function addedTasksIntoToDoListTasks(){}.***

***3)\*\*At The Third Step , You Should Get or Select The Input From The HTML Document File By Using: document.getElementById(“inputTask”).***

***4)\*\*At The Fourth Step , You Should Get The Text That The User Of The Responsive To Do List Task Writing In The Input Task To Added In The To Do List Of Different Tasks By Using: inputTask.value.trim(). , You Should Know That The Main Purpose Of Using Value Keyword , It Is Built-In Function In The Java Script Language , It Is Used To Get The Value That The User Of The Responsive To Do List Taks Project Entered In The Input Task, And The Main Purpose Of Using trim() Keyword , It Is Built-In Function In the Java Script Language Also , It Is Used To Remove Different White Spaces Before And After The Text That The User Enters In The Input Of The Responsive To Do List Tasks , And It Is Used To Prevent Different Users Of The Responsive To Do List Tasks To Enter White Spaces Only As An Input Task.***

***5)\*\*At The Fifth Step , You Should Make An If Condition To Check If The Input Task Text That The User Of The Responsive To Do List Tasks Entered Is An String , Or Empty String , If The Input Task Is An Empty String , You Should Return Alert For The User Of The Responsive To Do List Tasks Project That You Should Enter A String To Store In The To Do List Tasks. By Using: if(inputTaskText === “”) {return alert(“Please Enter A Valid Task”).}.***

***6)\*\*At The Sixth Step , You Should Get Or Select The Un Ordered List That Contains All Tasks That The User Of The Responsive To Do List Tasks Project Entered To Store In The To Do List Tasks Container By Using: document.getElementById(“toDoListTasks”).***

***7)\*\*At The Seventh Step , You Should Create An Li HTML Element That Used To Store Different Task That The User Of The Responsive To Do List Tasks Project Entered To Store In The To Do List Tasks Container By Using: document.createElement(“li”).***

***8)\*\*At The Eight Step , You Should Add The Task Into The Inner HTML Of The To Do List Task Container By Using Span HTML Element To Add The Task Input Text By Using Backtalks (``) At The Frist To Add The Frist Span Of Adding The Text Of The Input Value , And At The Second You Should Use Template Literal (${}) To Add The Input Task Text In The Span HTML Element That The User Of The Responsive To Do List Tasks Project Entered By Using: <span>${inputTaskText}</span>.***

***9)\*\*At The Nine Step , You Should Get Class For The To Do List Task to Distinguish This Element From Another HTML Element In The HTML Document File To Make The Styling Or Decorations Of This HTML Element That Stored The Text Of The Input Task That The User Of The Responsive To Do List Tasks Entered Previous.***

***10)\*\*At The Ten Step , You Should Replace The Styling , Or Decoration Of The ToDoListTasksContainer That Contains Different Tasks That Entered From The User Of The Responsive To Do List Tasks By Using: ToDoListTasksContainer.classList.replace(“d-none” , “d-block”) To Appear The ToDoListTasksContaainer For The User Of The Responsive To Do List Tasks Project. , You Should Know That classList Keyword Is A Built-In Function In The Java Script Language , It Is Mainly Used To Add Different Classes From The CSS Language , Or BootStrapCSS Library For Different HTML Elements , And replace() Is A Built-In Function In The Java Script Language , It Is Mainly Used To Replace The Class From The CSS Language , Or BootStrapCSS Library With Another Class From The CSS Language , Or BootStrapCSS Library. , And There Is Another Method To Change The Styling , Or Decoration Of The ToDoListTasksContainer That Contains Different Tasks That Entered From The User Of The Responsive To Do List Tasks Project , At The Frist By Using remove() To Remove The Class From The ToDoListTasksContainer By Using: ToDoListTasksContainer.classList.remove(“d-none”) , The Main Purpose Of Using remove() Function: It Is Built-In Function In The Java Script Language , It Is Used To Remove The CSS Styling , Or CSS Decoration From The ToDoListTasksContainer That Added Class From The CSS Language , Or BootStrapCSS Library By Using ToDoListTasksContainer.classList.remove(“d-none”) , And Finally , You Should Add The New CSS Class From The CSS Language , Or BootStrapCSS , At The Second You Should Add New CSS Class From The CSS Language , Or BootStrapCSS Library By Using: ToDoListTasksContainer.classList.add(“d-block”) , And The Main Purpose Of Using add() Function: It Is Built-In Function In The Java Script Language , It Is Mainly Used To Add New Class From The CSS Language , Or BootStrapCSS Library For The ToDoListTasksContainer To Make New CSS Styling , Or CSS Decoration Of The ToDoListTasksContainer.***

***11)\*\*At The Eleven Step , You Should Add Different Tasks In The ToDoListTasksContainer That Entered From The User Of The Responsive To Do List Tasks Project , To Make This You Should Use appendChild(): It Is Built-In Function In The Java Script Language , It Is Mainly Used To Add Different Tasks That Entered From The User Of The Responsive To Do List Tasks Project.***

***12)\*\*At The Twelve Step , When Adding Different Tasks That Entered From The User Of The Responsive To Do List Tasks Project , Then I Needed To An Outer Spaces From The Top , And Bottom Of The Different List Input Task Container , and You Can Do This By Using style Keyword To Make The Outer Spaces (margin Block) From The Top , And Bottom Of Elements By Using: toDoListTask.style.marginBlock=“12px”.***

***13)\*\*At The Thirteen Step , After Adding Task In The ToDoListTasksContainer From The User Of The Responsive To Do List Tasks Project , In This Point After Adding New Task , Then You Should Clear The Input Text Value That Entered In The Input To Add New Task To Add In The ToDoListTasksContainer By Using: inputTask.value = “”. To Clear The Input Field That Entered Input Task Text Value Here. , And This Line Is The Finally Of The Added New Task Into The To Do List Tasks Container.***

***14)\*\*At The Fourteen Step , You Should Add Event To The Main Button That Selected , Or Got From The HTML Document File To Add Different Task Input Text Value That Entered From The User Of The Responsive To Do List Tasks To Store In The To Do List Tasks Container , Event Like Click , And Then You Should Get This Event A Reference Of The Function That You Want To Invoke , Or Firing , Or Calling Without Calling , Invoking , Or Calling When Rendering , Or Archiving The Responsive To Do List Tasks Project The Function In This Case. By Using: mainButton.addEventListener(“click” , addedNewTaskIntoToDoListTaks).***

***15)\*\*At The Fifteen Step , You Should Add Event To The Document , Event Like Keydown On The Keyboard Of The Computer Device , If You Want To Add Task That Entered From The User Of The Responsive To Do List Tasks That Stored In The To Do List Tasks Container , In This Case By Using Case Keyword Event By Using: document.addEventListener(“keydown” , function(event) {}). , Keydown Event Refers To The Keydown On Keyboard On The Computer Device Like Any Keys On The Keyboard , And The Event Handler In This Case Refers To The Anemones Function That Takes An Event As Parameter For The Anemones Function , Event (EventInfo) , Or (E) Refers To The Event Information , It Is Object Event That Contains All Details Of Event That Done Like Click , Keydown Like Enter , Escape , Insert , Del , Or ETC , Or ETC , And Then Check If The Event On The Keyboard When Keydown On Enter , Insert To Add Task Input Text Value That Entered From The User Of The Responsive To Do List Tasks Project To Add In The To Do List Tasks Container , Then You Should Check If The Event.key == “Enter” , Or Event.Key == “Insert” Then Added New Task Into To Do List Tasks Container By Using: document.addEventListener(“keydown” , function(event){if((event.key == “Enter”) || (event.key == “Insert”)){addedNewTaskIntoToDoListTasks()}}).***

***16)\*\*At The Sixteen Step , You Should Make A Function To Delete Any Task From Different Tasks That Stored In The To Do List Tasks Container , That Different Tasks Entered From The User Of The Responsive To Do List Tasks , And In This Function , You Should Get Element Parameter For The deletedAnyTaskFromToDoListTasks(element){}. , This Function For Deleting Any Task From The To Do List Tasks Container That Entered From The User Of The Responsive To Do List Tasks Project , It Takes Only One Parameter As Element , Element Refers To The HTML Element That Triggered (Executing , Or Running The Function) The Function That Deleted A Task From The To Do List Tasks , In This Case Is <Span> That Stored The Task.***

***17)\*\*At The Seventeen Step , You Should Select ,Or Get The Input Task Container , In This Case Like <LI> By Using Element.parentElement. , In This Case The Parent Element Refers To The Container Like <DIV> , Or <LI> That Takes The Task Input Text Value , And The Trash Icon To Delete Task From The To Do List Tasks Container , And On Clicked In The Trash Icon , Is Used To Delete The Full Container Of The Task Input Text Value , And Trash Icon In <LI> Tag.***

***18)\*\*At The Eighteen Step , If You Want To Delete The Task From The To Do List Container That Entered From The User Of The Responsive To Do List Tasks , You Should Use remove() Function , It Is Built-In Function In The Java Script , It Is Used To Delete Or Remove Any Element From The Document. By Use: inputTask.remove().***

***19)\*\*At The Nineteen Step: , You Should Check If The Length Of Different Tasks In The To Do List Tasks Container , If The Length Of Tasks Equal Zero Then You Should Change The Displaying Of The To Do List Tasks Container From Display:None To Display:Block.***

******