**Week 3:**

**Name: VISWATHKUMAR N**

**Dept: B.Tech(IT)**

**Roll No: 717821F163**

**Mini Project 3:**

**Task Scheduler**

**Objective:** Develop a web-based task management application where users can add, delete, modify, and Page 17 of 19 save tasks to local storage. The application will incorporate asynchronous programming, DOM operations, and JSON serialization for storing and retrieving tasks.

**GitHub Link:**

**CODE:**

  <!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

  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"

  rel="stylesheet" integrity="sha384-

  4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9"

  crossorigin="anonymous">

  <script

  src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js

  " integrity="sha384-

  HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm"

  crossorigin="anonymous"></script>

  <style>

  h1{

  background-color: white;

  color:black;

  opacity: 0.7;

  align-items: center;

  }

  .card{

  width: 60%;

  height: 40%;

  left: 20%;

  opacity: 0.8;

  }

  body {

  background: linear-gradient(-45deg, #ff0b0b, #3ce789, #7f23d5, #2382d5);

  background-size: 400% 400%;

  animation: gradient 15s ease infinite;

  height: 100vh;

  }

  @keyframes gradient {

  0% {

  background-position: 0% 50%;

  }

  50% {

  background-position: 100% 50%;

  }

  100% {

  background-position: 0% 50%;

  }

  }

  .display{

  position: relative;

  align-items: center;

  justify-content: center;

  }

  </style>

  </head>

  <body>

  <center><h1>TASK SCHEDULER</h1></center>

  <div class="container">

  <div class="card">

  <form class="form-control" action="#">

  <div class="input-group mb-3">

  <span class="input-group-text" id="inputGroup-sizingdefault">Task:</span>

  <input type="text" id="task" class="form-control" aria-label="Sizing

  example input" aria-describedby="inputGroup-sizing-default">

  </div>

  <div class="input-group mb-3">

  <span class="input-group-text" id="inputGroup-sizingdefault">Time:</span>

  <input type="number" id="deadline" class="form-control" arialabel="Sizing example input" aria-describedby="inputGroup-sizing-default"

  placeholder="Minutes">

  </div>

  <div class="input-group mb-3">

  <label class="input-group-text" for="status">Status:</label>

  <select class="form-select" id="Status">

  <option selected>None</option>

  <option value="completed">completed</option>

  <option value="not completed">not completed</option>

  </select>

  </div>

  <div class="input-group mb-3">

  <span class="input-group-text" id="inputGroup-sizingdefault">Priority:</span>

  <input type="number" id="priority" class="form-control" arialabel="Sizing example input" aria-describedby="inputGroup-sizing-default">

  </div>

  <center>

  <button type="submit" id="add-task" class="btn btn-outline-success

  form-control" onclick="generate()">Create</button>

  </center>

  <!-- <center>

  <button type="submit" id="show" class="btn btn-primary form-control"

  onclick="gen()">show</button>

  </center> -->

  <br>

  <center>

  <div>

  <div class="input-group mb-3">

  <!-- <button type="button" class="btn btn-outline-dark" value="Display"

  onclick="generateTable()">Display Details:</button> -->

  <details>

  <Summary onclick="generateTable()"class="btn btn-outline-success formcontrol" >Display Tasks:</Summary>

  <div class="display">

  <center><h5>TASKS:</h5></center>

  <table id="mytable" class="table table-dark table-bordered">

  <tr>

  </tr>

  </table>

  </div>

  </details>

  </div>

  </div>

  </center>

  </form>

  </div>

  <br>

  <div class="card">

  <form class="form-control" action="#">

  <div class="input-group mb-3">

  <span class="input-group-text" id="delete">Task:</span>

  <input type="text" id="task-r" class="form-control" aria-label="Sizing

  example input" aria-describedby="delete">

  <button type="button" id="delete" class="btn btn-outline-dark"

  onclick="RemovebyTitle()">Delete</button>

  </div>

  <center>

  <button type="submit" id="delete" class="btn btn-outline-dark"

  onclick="RemoveFirstTask()">Delete-FIRST</button>

  <button type="submit" id="delete" class="btn btn-outline-dark" onclick="

  RemoveLastTask()">Delete-LAST</button>

  </center>

  <br>

  <div class="input-group mb-3">

  <span class="input-group-text" id="inputGroup-sizingdefault">Task:</span>

  <input type="text" id="name" class="form-control" aria-label="Sizing

  example input" aria-describedby="inputGroup-sizing-default">

  </div>

  <div class="input-group mb-3">

  <label class="input-group-text" for="status">Status:</label>

  <select class="form-select" id="Status-op">

  <option value="completed">completed</option>

  <option value="not completed">not completed</option>

  </select>

  <button class="btn btn-outline-dark" type="button"

  onclick="toggle()">Set</button>

  </div>

  </form>

  </div>

  </div>

  <script>

    function Create(){

 return {

 name:"task",

 date:0,

 time:0,

 status:"none",

 priority:0,

 display:function(){

 return '<td>Task\_Name :'+this.name+'</td><td>Date:'+this.date+'</td><td>Time :'+this.time+'</td><td>Status:'+this.status+'</td><td>priority :'+this.priority+'</td>';

 },

 ToggleStatus:function(status)

 {

 this.status=status;

 }

 }

}

let task=Create();

let store=new Array();

task.name="Java Assignment";

task.due\_date=20;

task.time=30;

task.status="completed";

task.priority=2;

store.push(task);

function AddTask(task){

 store.push(task);

}

function displayAll()

{

 let temp = store.map(e=>e.display());

 return temp;

}

function displayTitles()

{

 let temp = store.map(e=>e.name);

 return temp;

}

function AddTaskDetails(name,date,time,status,priority){

 let temp = Create();

 temp.name=name;

 temp.date=date;

 temp.status=status;

 temp.time=time

 temp.priority=priority;

 AddTask(temp);

 prioritySort();

}

function removeTaskByTitle(title){

 let index=store.findIndex(x=>x.title==title);

 console.log(index);

 store.splice(index,1);

}

function RemoveFirstTask()

{

 if(store.shift()==undefined)

 {

 alert("The list is Empty");

 }

 else

 {

 store.shift();

 alert("The Element is removed at First Successfully");

 }

}

function RemoveLastTask(){

 if(store.pop()==undefined)

 {

 alert("The list is Empty");

 }

 else

 {

 store.shift();

 alert("The Element is removed at Last Successfully");

 }

}

function prioritySort()

{

 store=store.sort((a,b)=>parseInt(a.priority)-parseInt(b.priority));

 console.log(store);

}

function generate(){

const currentDate = new Date();

const name = document.getElementById("task").value;

const priority = document.getElementById("priority").value;

const time = document.getElementById("deadline").value;

const date=currentDate;

const status = document.getElementById("Status").value;

 if (date === ""||time <= 0) {

 alert("Please select an correct time in minute for the deadline.")

 return;

 }

 let min=time\*60\*1000;

 setTimeout(()=>{alert("Time is over for this work "+name+" !Work faster Boss!");removeTaskByTitle(name);},min);

 AddTaskDetails(name,date,time,status,priority);

}

function generateTable(){

 let title=displayAll();

 mytable.innerHTML="";

 for(var i=0;i<title.length;i++)

 {

 mytable.innerHTML += "<td>"+title[i]+"</td>";

 }

}

function RemovebyTitle()

{

 let name=document.getElementById("task-r").value;

 removeTaskByTitle(name);

}

function toggle() {

 let name = document.getElementById("name").value;

 let status = document.getElementById("Status-op").value;

 let value=store.map(x=>

 {if(x.name==name)

 {

 x.ToggleStatus(status);

 console.log(x);

 }

 });

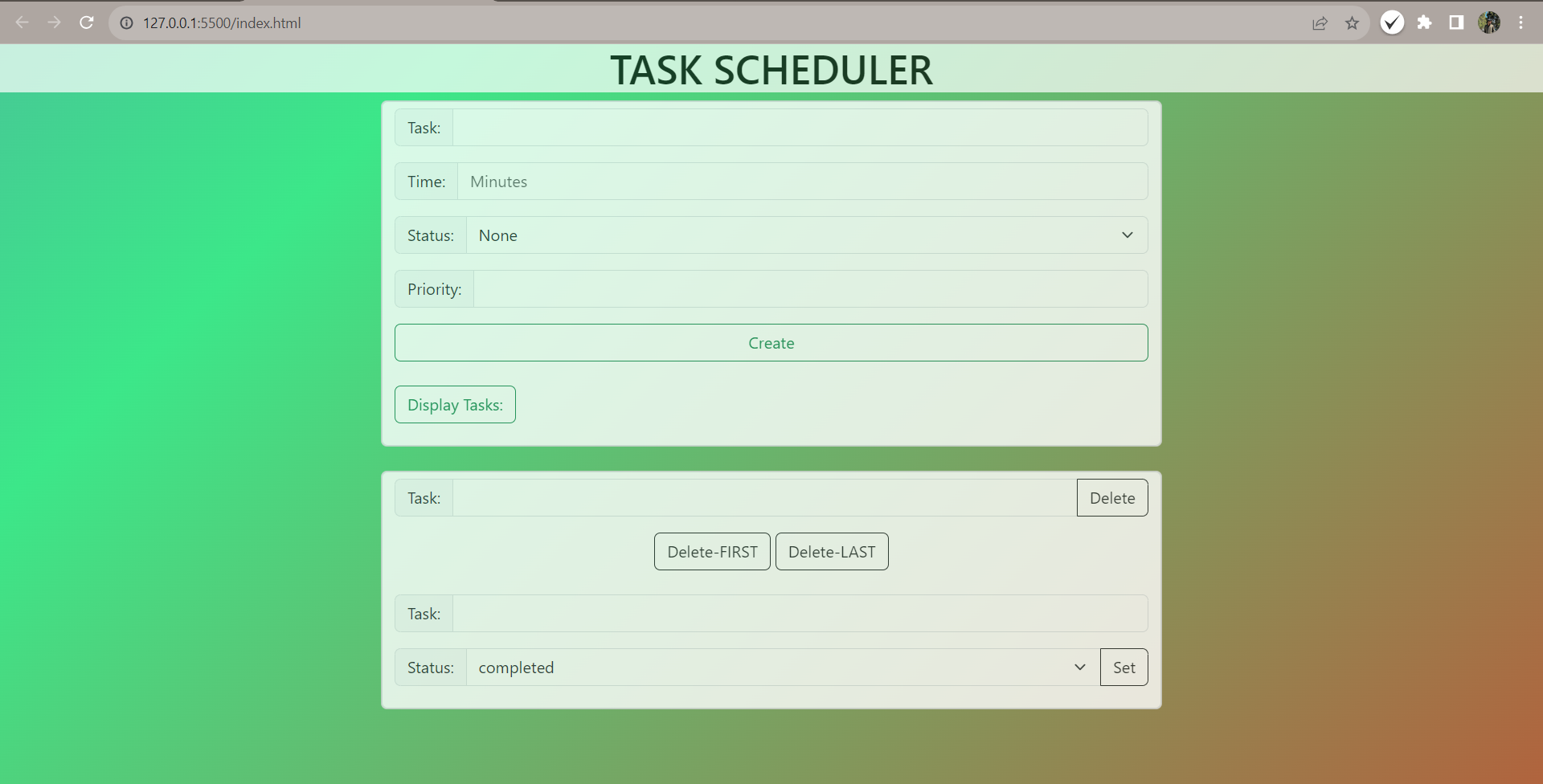
}

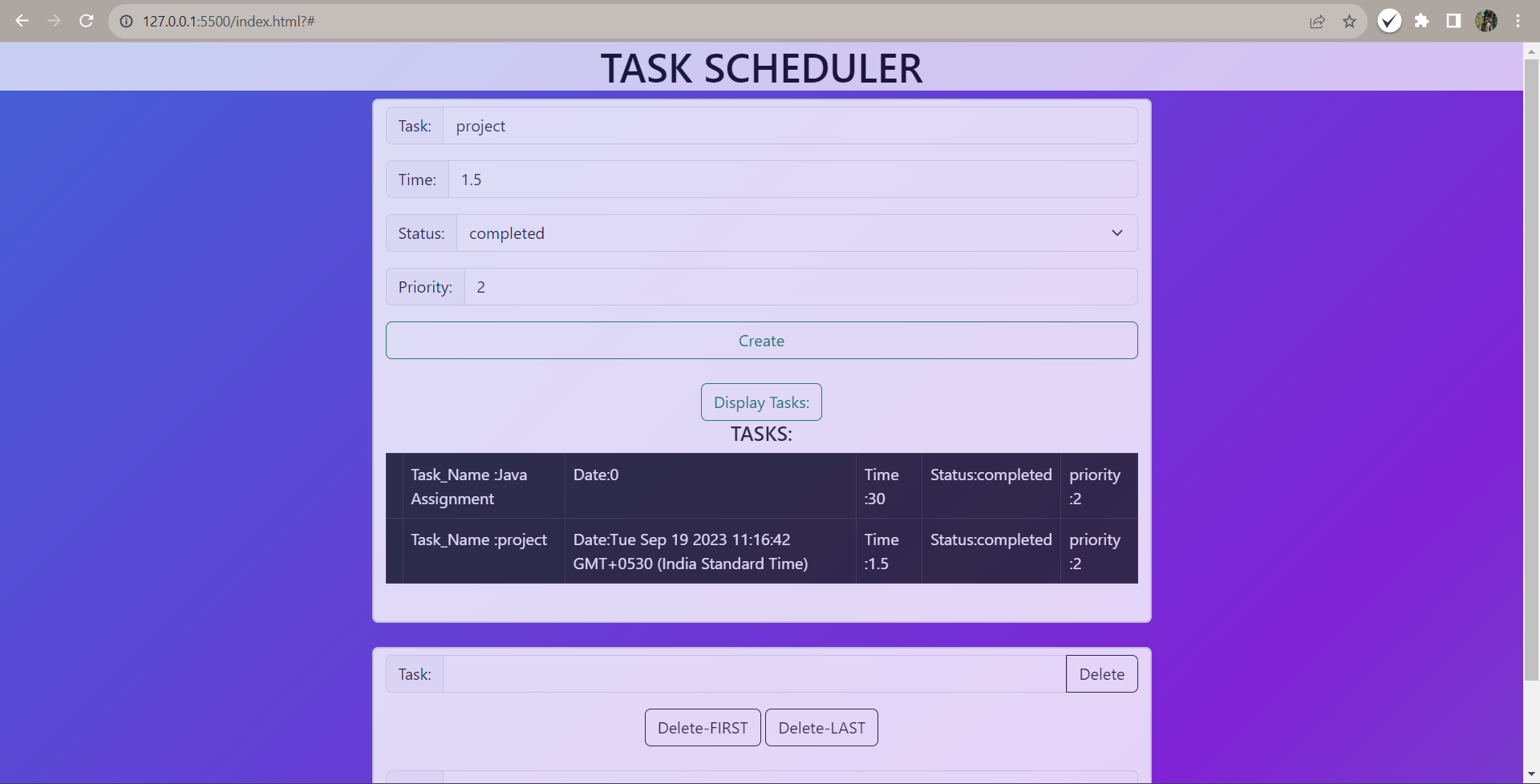
  </script>

  </body>

  </html>

**SCREENSHOTS:**

****

****