

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>TASK MANAGER</title>
  <link rel="stylesheet" href="styles.css">
  <link href="https://fonts.googleapis.com/css2?family=Outfit:wght@400;700&display=swap" rel="stylesheet">
</head>
<body>
  <header>
    <h1>TASK MANAGER</h1>
  </header>
  <div class="container">
    <aside class="sidebar">
      <button class="statusbtn" id="all">ALL</button>
      <button class="statusbtn" id="pending">PENDING</button>
      <button class="statusbtn" id="completed">COMPLETED</button>
    </aside>
    <main>
      <form class="tform" action="#">
        <h2>ADD YOUR TASK HERE!</h2>
        <input type="text" id="title" name="title" placeholder="Title">
        <textarea name="desc" id="desc" placeholder="Title Description"></textarea>
        <input type="date" name="dateinput" id="dateinput">
        <button class="addtask">ADD TASK</button>
      </form>
    </main>
    <div class="tasks">
      <h2 style="margin-top: 30px;">TASKS</h2><br>
      <div class='list' style="gap: 39px;">
        <h3>TITLE</h3>
        <h3>DESCRIPTION</h3>
        <h3>DUE &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</h3>
        <h3>EDIT</h3>
        <h3>DELETE</h3>
        <h3>STATUS</h3>
      </div>
      <ul class="taskslist">
      </ul>
    </div>
  </div>
  <footer>
    <div class="nav-link">
    </div>
  </div>
```

<p>© Task Manager 2024.All rights reserved</p>

</footer>

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

</body>

</html>

styles.css:

```
*{
  margin: 0;
  padding: 0;
  font-family: "Outfit",sans-serif;
}
header{
  background-color: rgba(15, 167, 83, 0.7);
  text-align: center;
  color: white;
  padding: 2%;
}
.container{
  display: flex;
  flex-direction: row;
  height: 72.6vh;
}
.sidebar{
  display: flex;
  flex-direction: column;
  gap: 15px;
  padding: 30px;
  background-color: rgba(0,0,0,0.2);
}
.form{
  display: flex;
  flex-direction: column;
  width: 78%;
  margin: 30px 70px;
  gap: 30px;
}
main{
  width: 50%;
  background-color: #f4f4f4;
}
input{
  height: 30px;
  font-size: 18px;
  border: #38814d solid 2.5px;
  border-radius: 3px;
  padding: 5px;
}
textarea{
  height: 60px;
  font-size: 18px;
  border: #38814d solid 2.5px;
  border-radius: 3px;
  padding: 5px;
}
.addtask{
  margin: 0 auto;
  width: 30%;
  height: 35px;
  border: none;
  background-color: rgba(15, 167, 83, 0.7);
```

```
text-align: center;
cursor: pointer;
border-radius: 5px;
color: white;
transition: background-color 0.2s ease-in;
}
footer{
background-color: rgba(15, 167, 83, 0.7);
text-align: center;
text-align: center;
color: white;
align-items: center;
display: flex;
flex-direction: column;
gap: 20px;
padding: 1.3%;
top: 0;
}
.nav-link{
word-spacing: 50px;
align-items: center;
}
.nav-link a{
text-decoration: none;
font-size: 22px;
color: #ffffff;
}
.nav-link a:hover{
color: gray;
}
.statusbtn{
border: none;
border-radius: 5px;
padding: 9px;
background-color: #38814d;
cursor: pointer;
color: white;
transition: background-color 0.35s ease-out;
}
.statusbtn:hover{
background-color: #65d083;
}
.addtask:hover{
background-color:#38814d;
}
.tasks{
background-color: rgba(0,0,0,0.2);
width: 50%;
padding-left: 30px;
}
.taskslist{
overflow: auto;
padding-top: 15px;
max-height: 54vh;
}
.list{
display: flex;
flex-direction: row;
gap: 20px;
align-items: center;
}
.list .des{
```

```

max-width: 110px;
word-wrap: break-word;
overflow-wrap: break-word;
text-align: justify;
min-width: 110px;
}
.list .h44{
max-width: 70px;
word-wrap: break-word;
overflow-wrap: break-word;
font-size: 20px;
text-align: justify;
font-weight: 800;
min-width: 70px;
}
.actions{
padding: 3px;
width: 77px;
border-radius: 10px;
background: none;
cursor: pointer;
background-color: rgba(0,0,0,0.4);
}
.actions:hover{
background-color: rgba(0,0,0,0.2);
}
.statusbtn:focus-within{
background-color: #023b0b;
color: red;
}

```

script.js:

```

document.addEventListener('DOMContentLoaded', function () {
    let tasks = [];
    document.querySelector('.tform').addEventListener('submit', function (e) {
        e.preventDefault();
        const title = document.getElementById('title').value.trim();
        const desc = document.getElementById('desc').value.trim();
        const date = document.getElementById('dateinput').value;
        if (title === "" || desc === "" || date === "") {
            alert('Please fill out all fields');
            return;
        }
        const newTask = {
            id: Date.now(),
            title,
            desc,
            date,
            status: 'pending' // Default status is pending
        };
        tasks.push(newTask);
        // Send task to the server (mock AJAX)
        addTaskToServer(newTask);
        // Clear form
        document.querySelector('.tform').reset();
    });

```

```

// Display updated tasks
displayTasks('all');
});
// Filter buttons
document.querySelectorAll('.statusbtn').forEach(button => {
  button.addEventListener('click', function () {
    const filter = this.textContent.toLowerCase();
    displayTasks(filter);
  });
});
// Add task to server (simulated)
function addTaskToServer(task) {
  // Simulate AJAX request to add task to the server
  const xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function () {
    if (this.readyState === 4 && this.status === 200) {
      console.log('Task added:', task);
    }
  };
  xhttp.open('POST', '/add-task', true);
  xhttp.setRequestHeader('Content-Type', 'application/json');
  xhttp.send(JSON.stringify(task));
}
// Display tasks based on filter
function displayTasks(filter) {
  const taskList = document.querySelector('.tasklist');
  taskList.innerHTML = "";
  const filteredTasks = tasks.filter(task => {
    if (filter === 'all') return true;
    if (filter === 'pending') return task.status === 'pending';
    if (filter === 'completed') return task.status === 'completed';
  });
  filteredTasks.forEach(task => {
    const li = document.createElement('li');
    li.innerHTML = `
      <div class='list'>
        <div class='h44'>${task.title}</div>
        <div class='des'>${task.desc}</div>
        <p style='margin-left:10px;' >${task.date}</p>
        <button class='actions' onclick="editTask(${task.id})">Edit</button>
        <button class='actions' onclick="deleteTask(${task.id})">Delete</button>
        <button class='actions' style='margin-left:10px;' onclick="toggleTaskStatus(${task.id})">
          ${task.status === 'pending' ? 'Completed' : 'Pending'}
        </button>
      </div>
    `;
    li.classList.add(task.status);
    taskList.appendChild(li);
  });
}

```

```

// Edit Task
window.editTask = function (id) {
  const task = tasks.find(task => task.id === id);
  if (!task) return;

  // Pre-fill form with task data
  document.getElementById('title').value = task.title;
  document.getElementById('desc').value = task.desc;
  document.getElementById('dateinput').value = task.date;
  // Remove the task from the list temporarily
  tasks = tasks.filter(task => task.id !== id);
};

// Delete Task
window.deleteTask = function (id) {
  tasks = tasks.filter(task => task.id !== id);
  // Remove task from the server (mock AJAX)
  deleteTaskFromServer(id);
  displayTasks('all');
};

// Delete task from server (simulated)
function deleteTaskFromServer(id) {
  // Simulate AJAX request to delete task
  const xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function () {
    if (this.readyState === 4 && this.status === 200) {
      console.log('Task deleted:', id);
    }
  };
  xhttp.open('DELETE', `/delete-task/${id}`, true);
  xhttp.send();
}

// Toggle task status (mark as completed/pending)
window.toggleTaskStatus = function (id) {
  const task = tasks.find(task => task.id === id);
  if (task) {
    task.status = task.status === 'pending' ? 'completed' : 'pending';
    displayTasks('all');
  }
};
});

```

IMPLEMENTATION SCREENSHOTS:

