Mini Project Report

**On**

**To-Do List Project**

**Submitted by**

**Aviral Singh(2115000244)**

**Nikhil Pandey(2115000669)**

**Shivansh Kumar(2115000961)**

**Anuj Gupta(2115000174)**

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**



**GLA University**

**Mathura- 281406, INDIA**

**2023**

**Department of computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**

**Declaration**

I hereby declare that the work which is being presented in the Mini Project **“To-Do List Website”** in partial fulfillment of the requirements for Mini Project viva voce, is an authentic record of my own work carried under the supervision of MR Suman Kumar Dasat GLA University, Mathura during 22nd Oct 2023 to 29th Nov 2023 .

**Contents**

|  |  |
| --- | --- |
| Abstract | ii |
| Certificate | iii |
| Acknowledgments | iv |
| 1. **Introduction** | **1** |
| 1. **Website Requirement Analysis** | **2** |
| 2.1 Define the problem ………………………………………………….. | 2 |
| 2.2 Define the modules and their functionalities ……………..………………………………… | 3 |
| 1. **Software Design** | **5** |
| 3.1 Data Flow Diagram, Use Case Diagram | 5 |
| 3.2 Database Design | 8 |
| 1. **Testing and Implementation** | **13** |
| **References/Bibliography** | **18** |
| 1. **Appendices** | **19** |

**ABSTRACT**

The To-Do List Website is an online platform designed to assist users in organizing and managing their tasks efficiently. The website provides a user-friendly interface that allows individuals to create, edit, prioritize, and track their to-do lists. Key features include:

* Task Management: Users can easily add, delete, and modify tasks, enabling dynamic adjustment of their to-do lists based on changing priorities.
* Prioritization and Categories: The website allows users to assign priorities to tasks and categorize them according to projects, deadlines, or custom labels, facilitating a structured organization of responsibilities.
* User Accounts: To enhance personalization, the website offers user account functionalities. Registered users can save their to-do lists, access them from multiple devices, and receive reminders.
* Reminders and Notifications: Integration of reminder features enables users to set deadlines for tasks and receive notifications via email or in-app alerts, ensuring timely completion of assignments.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to all those who have contributed to the successful completion of the To-Do List Project. This endeavor would not have been possible without the support, guidance, and encouragement of several individuals and resources.

First and foremost, I extend my heartfelt thanks to Mr. Suman Kumar Das, whose invaluable insights and expert guidance played a pivotal role in shaping the direction of this project. Their mentorship provided both inspiration and clarity throughout the development process.

I am also grateful to MDN Web Docs, which provided the necessary resources, infrastructure, and environment conducive to creative problem-solving. The collaborative atmosphere fostered within the organization greatly contributed to the project's success.

Special appreciation goes to Team Members for their dedicated efforts and collaboration. The synergy of our team significantly enhanced the quality and efficiency of the project, and I am thankful for their commitment to excellence.

Finally, I express my gratitude to my friends and family for their unwavering support and understanding during the demanding phases of this project. Their encouragement kept me motivated and focused, even during challenging times.

### INTODUCTION AND OBJECTIVE

In the fast-paced modern world, effective task management is crucial for individuals to navigate through their daily responsibilities efficiently. Recognizing the need for a comprehensive solution to organize and prioritize tasks, the To-Do List Project was conceived. This project aims to develop a user-friendly and feature-rich platform that empowers users to manage their tasks, deadlines, and projects with ease.

The proliferation of digital tools and the increasing complexity of daily schedules call for a centralized and intuitive solution. The To-Do List Project seeks to address this by providing a robust online platform accessible across various devices. Whether it's personal goals, work assignments, or collaborative projects, the To-Do List Project aims to simplify task management, enhance productivity, and contribute to a more organized and stress-free daily life.

Objectives :

The primary objectives of the To-Do List Project are as follows:

* User-Friendly Interface: Develop an intuitive and easy-to-navigate user interface that accommodates users with varying levels of technical proficiency, ensuring a seamless and enjoyable experience.
* Comprehensive Task Management: Implement a comprehensive set of features for task creation, modification, prioritization, categorization, and deletion, enabling users to tailor their to-do lists to their specific needs.
* Cross-Platform Accessibility: Ensure the platform is accessible across desktops, laptops, tablets, and smartphones, allowing users to manage their tasks anytime, anywhere.
* Reminder and Notification System: Integrate a robust reminder system that allows users to set deadlines for tasks and receive timely notifications via email or in-app alerts.
* Collaboration Features: Facilitate collaboration on projects by allowing users to share their to-do lists with team members, fostering teamwork and coordination.
* Security and Privacy: Prioritize user data security by implementing encryption measures to protect sensitive information, ensuring a secure environment for users to manage their tasks.
* Scalability: Design the platform to be scalable, accommodating future enhancements and a growing user base.

By achieving these objectives, the To-Do List Project aims to provide a valuable and accessible tool for individuals and teams seeking an efficient and organized approach to task management in their personal and professional lives.

**2.1 PROBLEM STATEMENT:**

In the contemporary landscape of hectic lifestyles and dynamic work environments, individuals often find themselves overwhelmed by the sheer volume and diversity of tasks they need to manage. The absence of an efficient and centralized system for organizing and prioritizing these tasks leads to increased stress, missed deadlines, and a decrease in overall productivity. Traditional paper-based to-do lists lack adaptability, while existing digital solutions may lack the essential features needed for comprehensive task management.

**SOLUTION:**

The comprehensive task management features, including the ability to edit, prioritize, and delete tasks, provide users with a flexible and efficient way to adapt their to-do lists based on changing circumstances. This adaptability contributes to more effective task management.

**SCOPE:**

* The primary focus is on providing users with a platform to create, edit, prioritize, and delete tasks. This includes the ability to organize tasks into categories, set due dates, and assign priorities.
* The To-Do List Website should be accessible across various devices and platforms, including desktops, laptops, tablets, and smartphones. This ensures users can manage their tasks seamlessly regardless of the device they are using.

**2.2 MODULES:**

* User Authentication:
* Dashboard:
* Task Management:
* Collaboration:
* Reminders and Notifications:
* Categories and Labels:
* User Profiles:
* Search and Filters:

**Task Management:**

**Task Creation :**

* Description: Users should be able to create tasks with clear and concise descriptions.
* Due Dates: Include the option to set due dates for tasks to help users prioritize and manage their time effectively.
* Categories/Labels: Allow users to categorize tasks by projects, goals, or custom labels to enhance organization.

**Task Organization :**

* Priority Levels: Implement a system for users to assign priority levels to tasks (e.g., high, medium, low).

**Task Details and Notes :**

* Notes Section: Provide a space for users to add detailed notes or additional information related to each task.

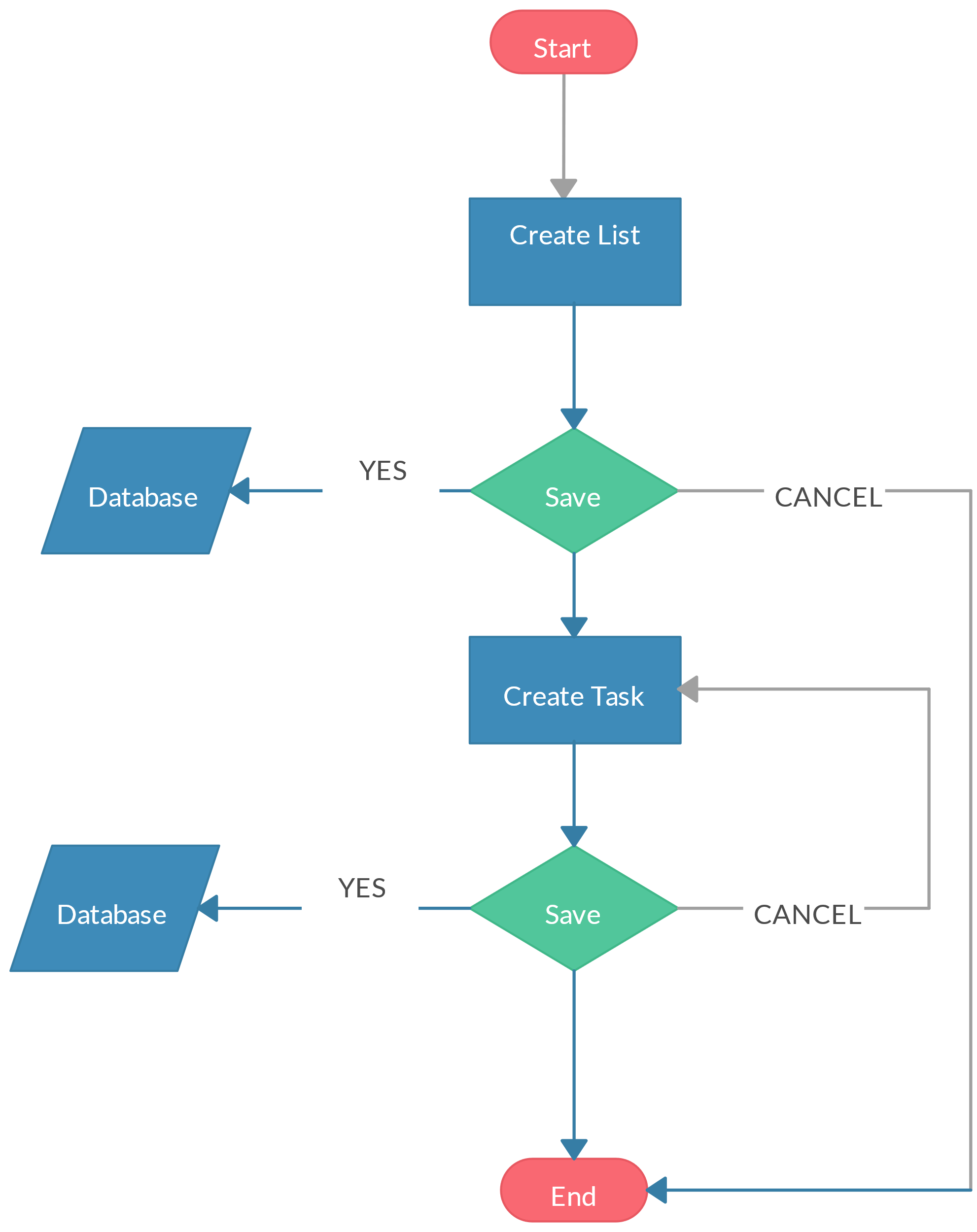
**Task Status and Progress :**

* Status Updates: Include status options (e.g., to-do, in progress, completed) to track the status of each task.

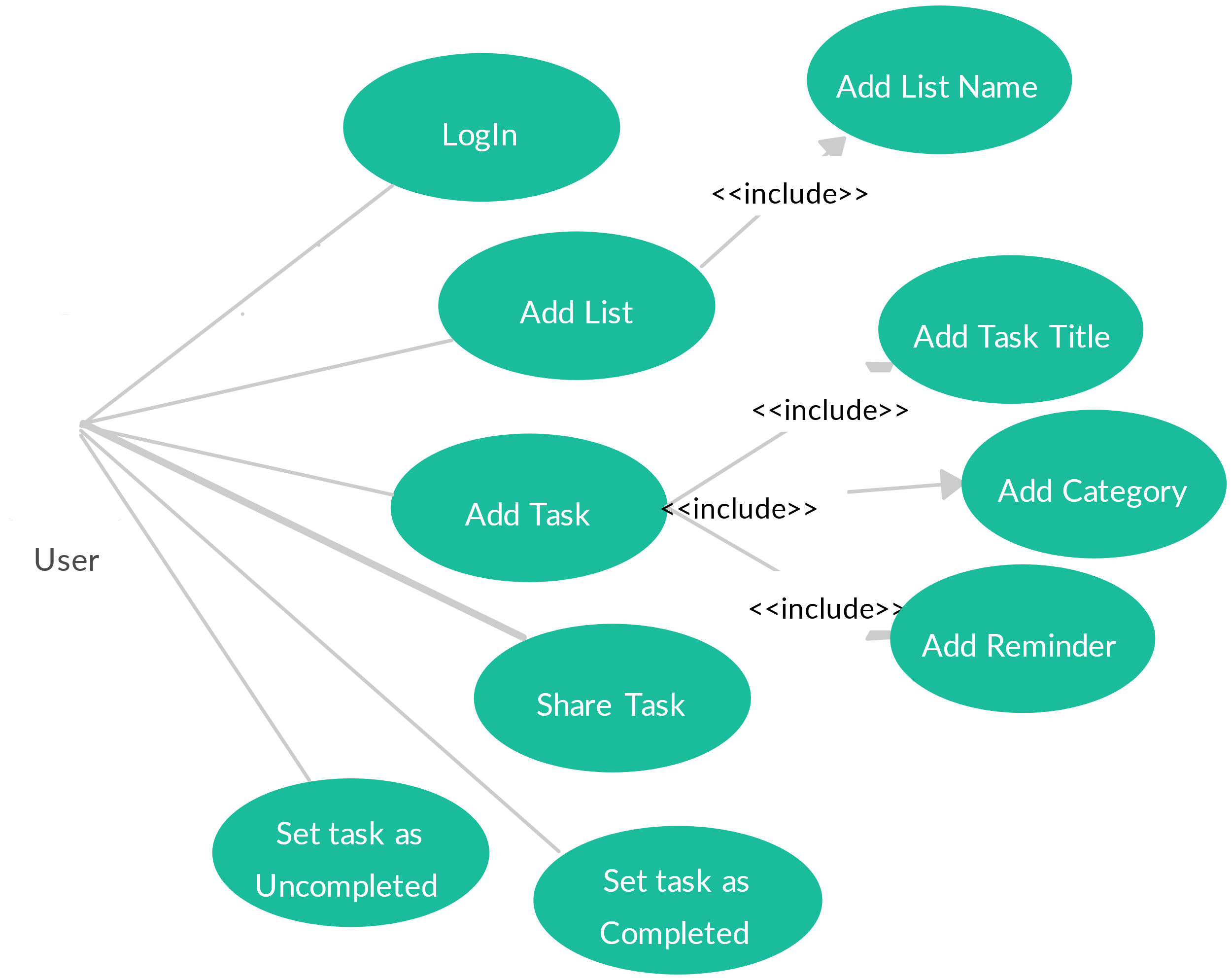
**Progress Tracking :**

* For longer-term projects, consider a progress-tracking feature to monitor overall project completion.

**3.1 Data Flow Diagram**

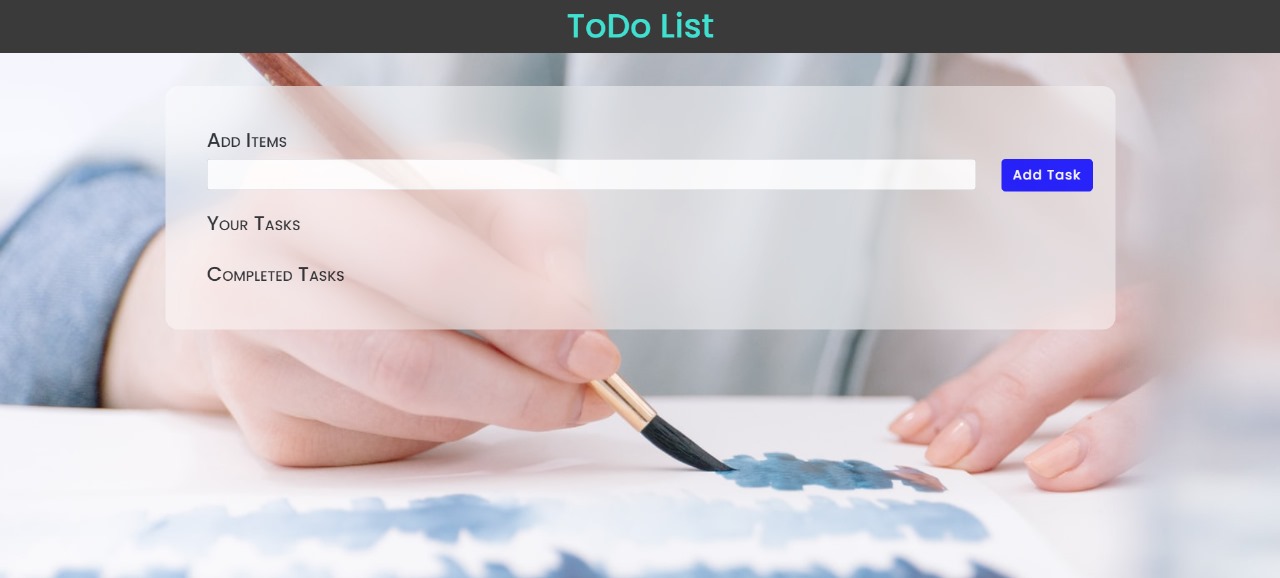
****

**Use Case Diagram**



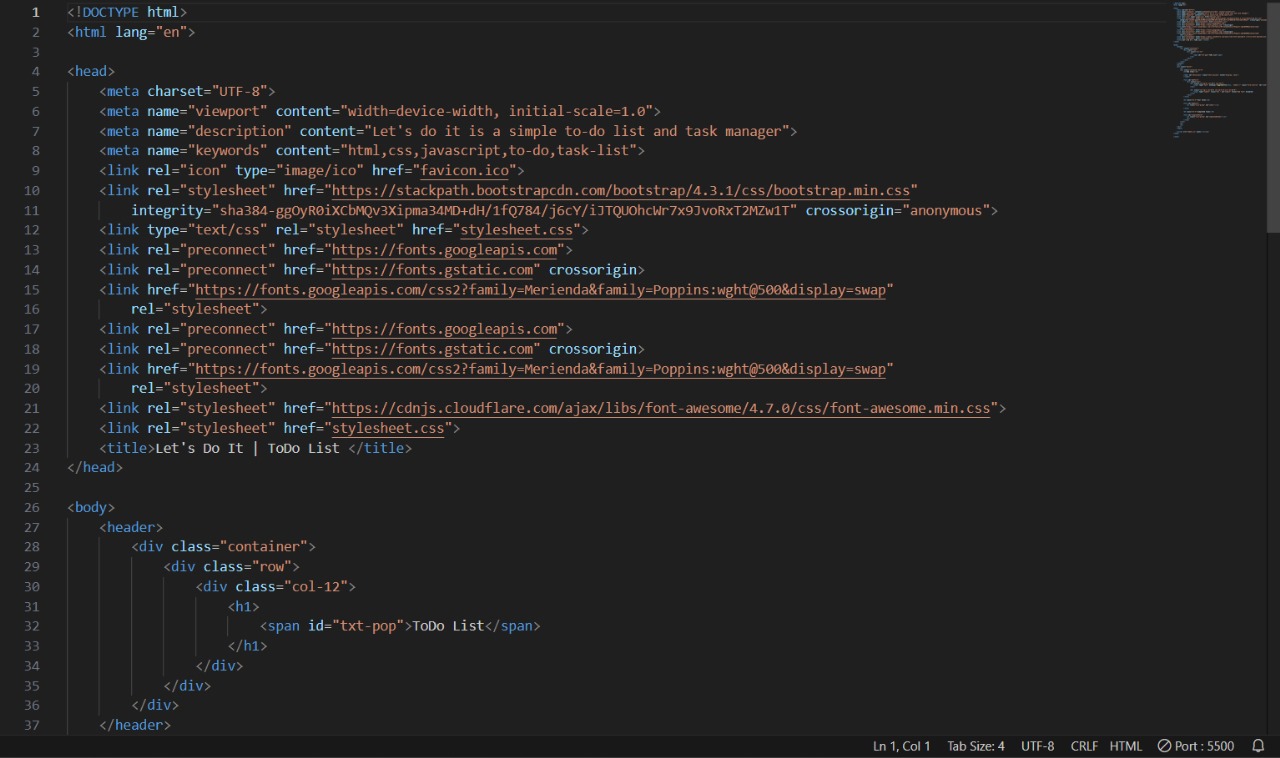
**TEST CASES:**

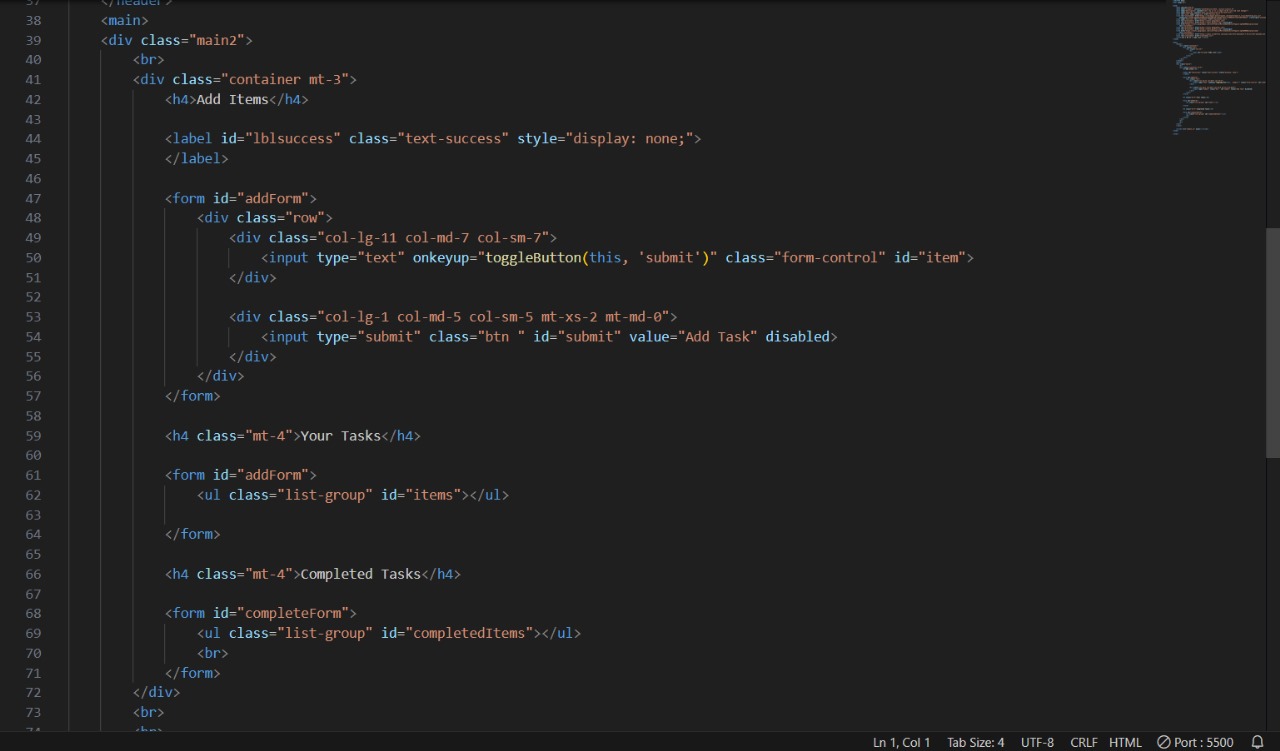
**VALID DATA DURING LOGIN:**

****

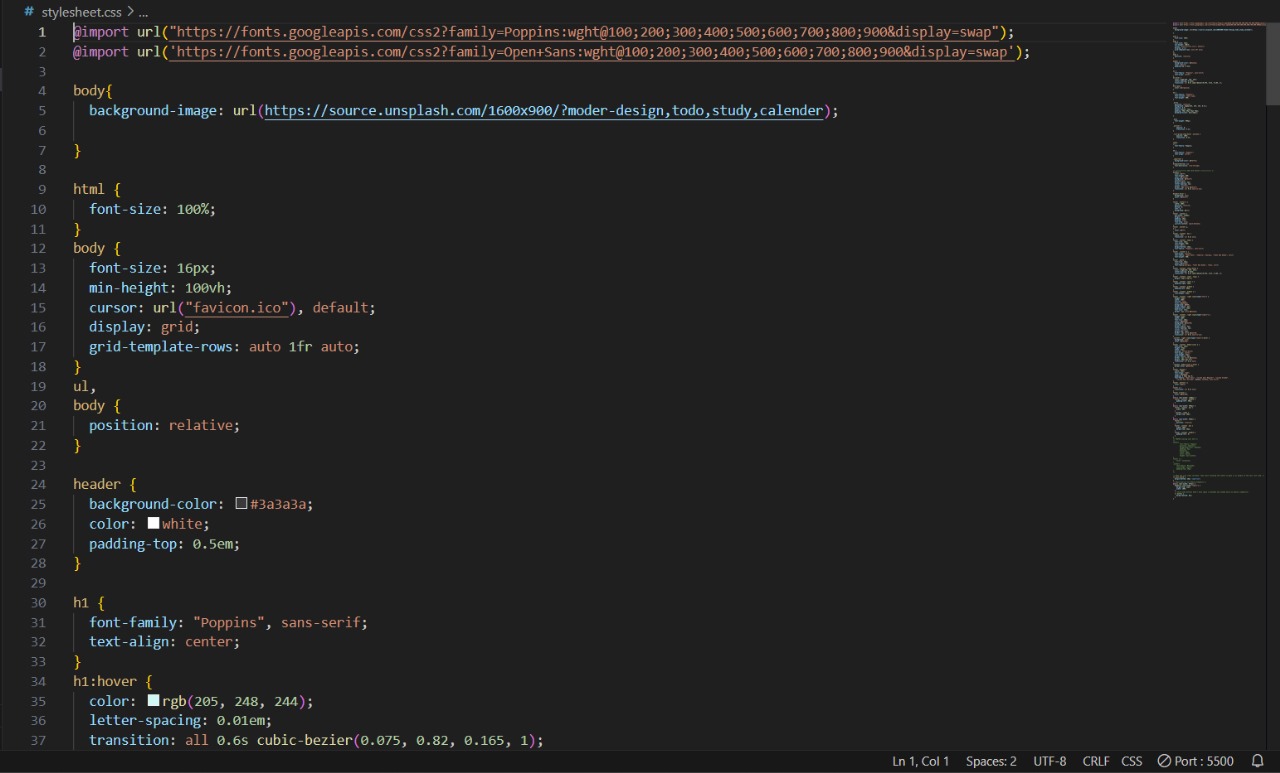
**Screen Shots –**

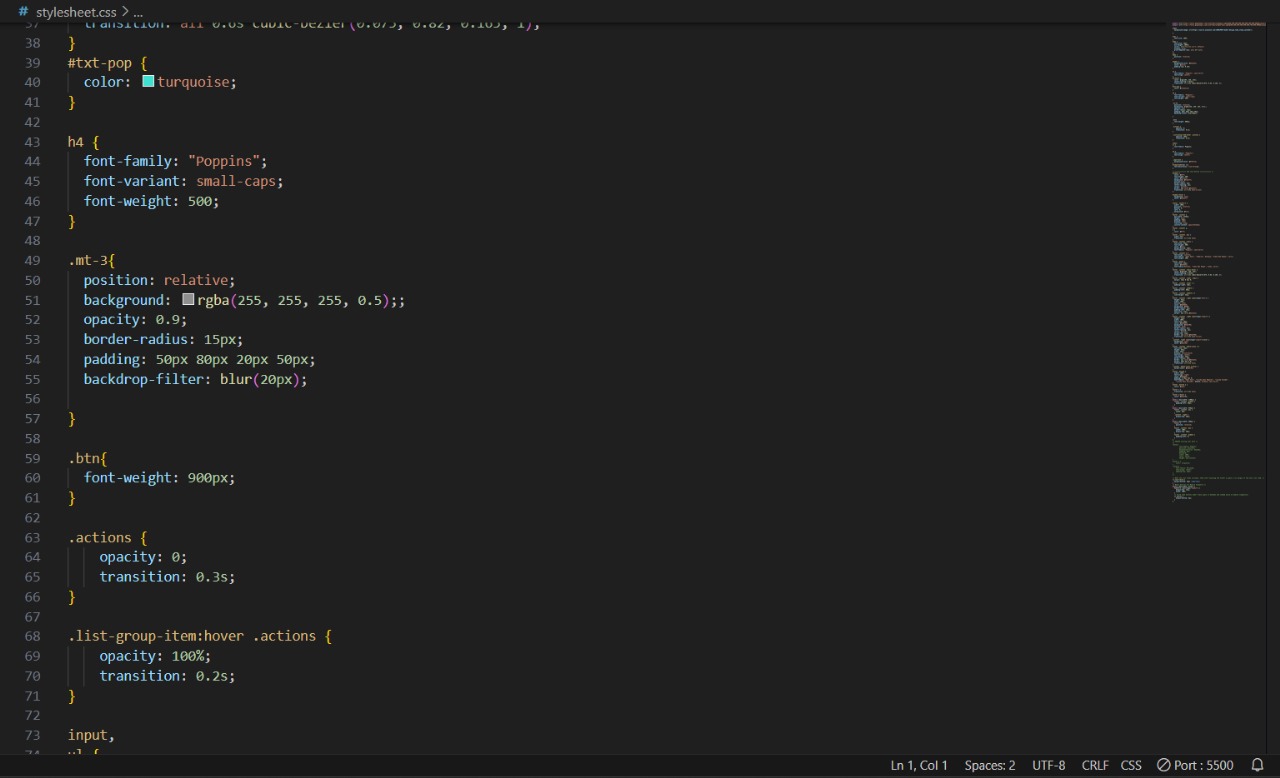
**HTML**

****

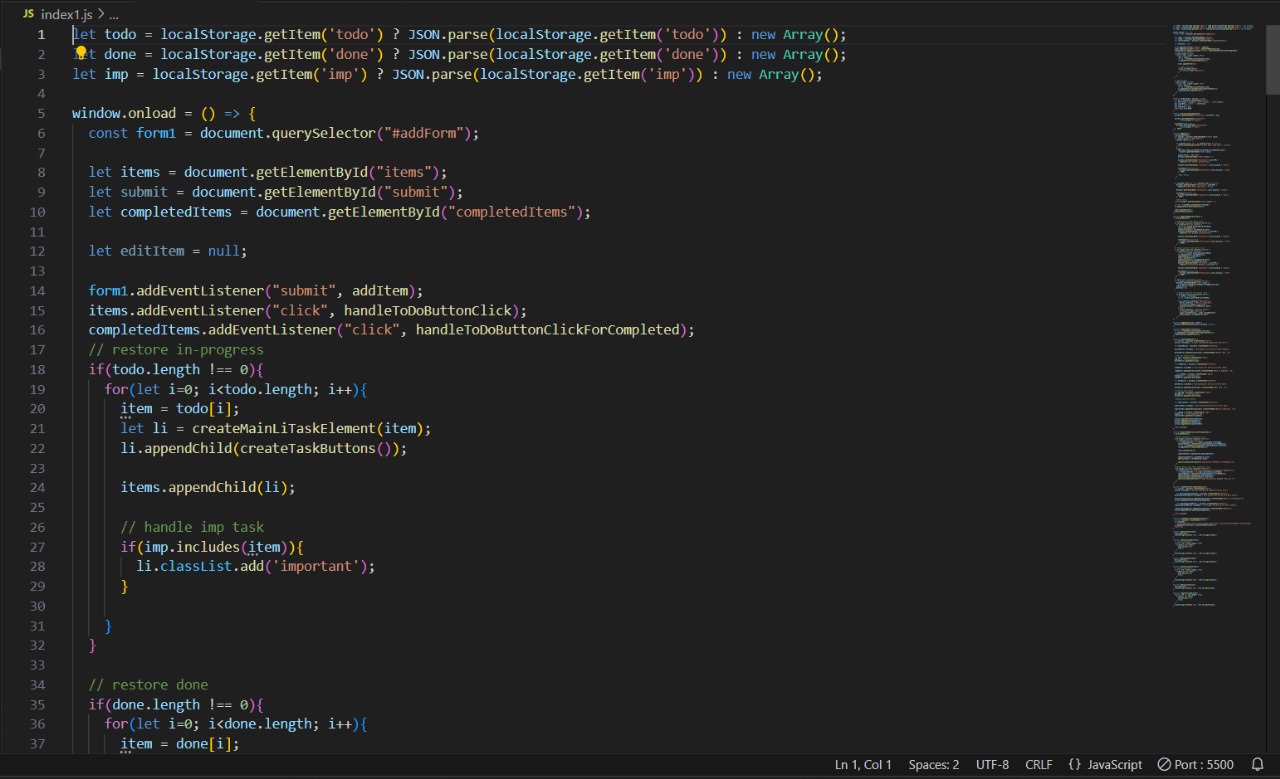
****

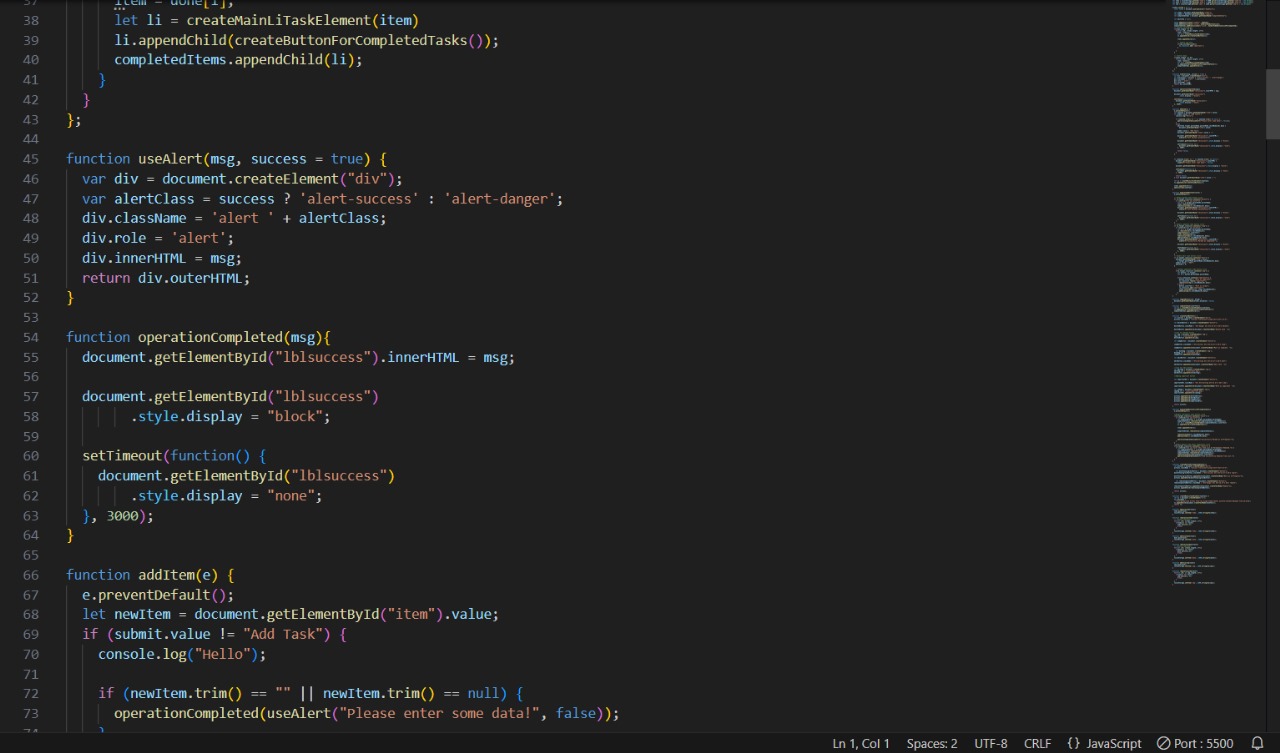
**CSS**

****

****

**JS**

****

****

**REFERENCES:**

* https://www.tutorialpoint.com/mysql/index.htm
* https://www.w3schools.com/java/java\_intro.asp

**CODE TEMPLATES:**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="Let's do it is a simple to-do list and task manager">

<meta name="keywords" content="html,css,javascript,to-do,task-list">

<link rel="icon" type="image/ico" href="favicon.ico">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Merienda&family=Poppins:wght@500&display=swap"

rel="stylesheet">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Merienda&family=Poppins:wght@500&display=swap"

rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="stylesheet.css">

<title>Let's Do It | ToDo List </title>

</head>

<body>

<header>

<div class="container">

<div class="row">

<div class="col-12">

<h1>

<span id="txt-pop">ToDo List</span>

</h1>

</div>

</div>

</div>

</header>

<main>

<div class="main2">

<br>

<div class="container mt-3">

<h4>Add Items</h4>

<label id="lblsuccess" class="text-success" style="display: none;">

</label>

<form id="addForm">

<div class="row">

<div class="col-lg-11 col-md-7 col-sm-7">

<input type="text" onkeyup="toggleButton(this, 'submit')" class="form-control" id="item">

</div>

<div class="col-lg-1 col-md-5 col-sm-5 mt-xs-2 mt-md-0">

<input type="submit" class="btn " id="submit" value="Add Task" disabled>

</div>

</div>

</form>

<h4 class="mt-4">Your Tasks</h4>

<form id="addForm">

<ul class="list-group" id="items"></ul>

</form>

<h4 class="mt-4">Completed Tasks</h4>

<form id="completeForm">

<ul class="list-group" id="completedItems"></ul>

<br>

</form>

</div>

<br>

<br>

</div>

</main>

<script src="index1.js" async></script>

</body>

</html>

**CSS**

@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap");

@import url('https://fonts.googleapis.com/css2?family=Open+Sans:wght@100;200;300;400;500;600;700;800;900&display=swap');

body{

background-image: url(https://source.unsplash.com/1600x900/?moder-design,todo,study,calender);

}

html {

font-size: 100%;

}

body {

font-size: 16px;

min-height: 100vh;

cursor: url("favicon.ico"), default;

display: grid;

grid-template-rows: auto 1fr auto;

}

ul,

body {

position: relative;

}

header {

background-color: #3a3a3a;

color: white;

padding-top: 0.5em;

}

h1 {

font-family: "Poppins", sans-serif;

text-align: center;

}

h1:hover {

color: rgb(205, 248, 244);

letter-spacing: 0.01em;

transition: all 0.6s cubic-bezier(0.075, 0.82, 0.165, 1);

}

#txt-pop {

color: turquoise;

}

h4 {

font-family: "Poppins";

font-variant: small-caps;

font-weight: 500;

}

.mt-3{

position: relative;

background: rgba(255, 255, 255, 0.5);;

opacity: 0.9;

border-radius: 15px;

padding: 50px 80px 20px 50px;

backdrop-filter: blur(20px);

}

.btn{

font-weight: 900px;

}

.actions {

opacity: 0;

transition: 0.3s;

}

.list-group-item:hover .actions {

opacity: 100%;

transition: 0.2s;

}

input,

ul {

font-family: Poppins;

}

h5 {

font-family: "Poppins";

text-align: center;

}

.important {

background-color: #fbf71e;

}

#completedItems li{

text-decoration: line-through;

}

#submit {

color: #fff;

font-weight: 500;

color: #fafafa;

background: #1612ff;

outline:none;

border-radius: 5px;

letter-spacing: 1px;

cursor: pointer;

border: 2px solid #1612ff;

transition: all 0.3s ease-in-out;

}

#submit:hover {

background: none;

color: #1612ff;

}

footer .footer-1 {

width: 100%;

position: relative;

bottom: 0;

left: 0;

background: #111;

}

footer .content {

max-width: 1350px;

margin: auto;

padding: 20px;

display: flex;

flex-wrap: wrap;

justify-content: space-between;

}

footer .content p,

a {

color: #fff;

}

footer .content .box {

width: 33%;

transition: all 0.4s ease;

}

footer .content .topic {

font-size: 24px;

font-weight: 500;

color: #fff;

margin-bottom: 16px;

font-family: "Poppins", sans-serif;

}

footer .content p {

text-align: justify;

font-family: "Open Sans", "Cambria", Georgia, "Times New Roman", serif;

font-weight: 400;

}

footer .quote {

font-size: 24px;

color: #f6f5f3;

font-family:Georgia, 'Times New Roman', Times, serif;

}

footer .content .topic:hover {

color: rgb(205, 248, 244);

letter-spacing: 0.01em;

transition: all 0.6s cubic-bezier(0.075, 0.82, 0.165, 1);

}

footer .content .lower .topic {

margin: 24px 0 5px 0;

}

footer .content .lower i {

padding-right: 16px;

}

footer .content .middle {

padding-left: 80px;

}

footer .content .middle a {

line-height: 32px;

}

footer .content .right input[type="text"] {

height: 45px;

width: 100%;

outline: none;

color: #d9d9d9;

background: #000;

border-radius: 5px;

padding-left: 10px;

font-size: 17px;

border: 2px solid #222222;

}

footer .content .right input[type="submit"] {

height: 42px;

width: 100%;

font-size: 18px;

color: #d9d9d9;

background: #eb2f06;

outline: none;

border-radius: 5px;

letter-spacing: 1px;

cursor: pointer;

margin-top: 12px;

border: 2px solid #eb2f06;

transition: all 0.3s ease-in-out;

}

.content .right input[type="submit"]:hover {

background: none;

color: #eb2f06;

}

footer .content .media-icons a {

font-size: 16px;

height: 45px;

width: 45px;

display: inline-block;

text-align: center;

line-height: 43px;

border-radius: 5px;

border: 2px solid #222222;

margin: 30px 5px 0 0;

transition: all 0.3s ease;

}

.content .media-icons a:hover {

border-color: #eb2f06;

}

footer .bottom {

width: 100%;

text-align: right;

color: #d9d9d9;

padding: 0 40px 5px 0;

font-family: "Open Sans", "Lucida Sans Regular", "Lucida Grande",

"Lucida Sans Unicode", Geneva, Verdana, sans-serif;

}

footer .bottom a {

color: aqua;

}

footer a {

transition: all 0.3s ease;

}

footer a:hover {

color: #eb2f06;

}

@media (max-width: 1100px) {

footer .content .middle {

padding-left: 50px;

}

}

@media (max-width: 950px) {

footer .content .box {

width: 50%;

}

.content .right {

margin-top: 40px;

}

}

@media (max-width: 560px) {

footer {

position: relative;

}

footer .content .box {

width: 100%;

margin-top: 30px;

}

footer .content .middle {

padding-left: 0;

}

}

li:last-child {

margin-bottom: 20px !important;

}

@media (max-width: 577px) {

#addForm input[type="submit"] {

margin-top: 12px;

width: 100%;

}

li button {

margin-bottom: 8px;

}

}

**JS**

let todo = localStorage.getItem('todo') ? JSON.parse(localStorage.getItem('todo')) : new Array();

let done = localStorage.getItem('done') ? JSON.parse(localStorage.getItem('done')) : new Array();

let imp = localStorage.getItem('imp') ? JSON.parse(localStorage.getItem('imp')) : new Array();

window.onload = () => {

const form1 = document.querySelector("#addForm");

let items = document.getElementById("items");

let submit = document.getElementById("submit");

let completedItems = document.getElementById("completedItems");

let editItem = null;

form1.addEventListener("submit", addItem);

items.addEventListener("click", handleToDoButtonClick);

completedItems.addEventListener("click", handleToDoButtonClickForCompleted);

// restore in-progress

if(todo.length !== 0){

for(let i=0; i<todo.length; i++){

item = todo[i];

let li = createMainLiTaskElement(item);

li.appendChild(createTaskButtons());

items.appendChild(li);

// handle imp task

if(imp.includes(item)){

li.classList.add('important');

}

}

}

// restore done

if(done.length !== 0){

for(let i=0; i<done.length; i++){

item = done[i];

let li = createMainLiTaskElement(item)

li.appendChild(createButtonForCompletedTasks());

completedItems.appendChild(li);

}

}

};

function useAlert(msg, success = true) {

var div = document.createElement("div");

var alertClass = success ? 'alert-success' : 'alert-danger';

div.className = 'alert ' + alertClass;

div.role = 'alert';

div.innerHTML = msg;

return div.outerHTML;

}

function operationCompleted(msg){

document.getElementById("lblsuccess").innerHTML = msg;

document.getElementById("lblsuccess")

.style.display = "block";

setTimeout(function() {

document.getElementById("lblsuccess")

.style.display = "none";

}, 3000);

}

function addItem(e) {

e.preventDefault();

let newItem = document.getElementById("item").value;

if (submit.value != "Add Task") {

console.log("Hello");

if (newItem.trim() == "" || newItem.trim() == null) {

operationCompleted(useAlert("Please enter some data!", false));

}

else{

editItem.target.parentNode.parentNode.childNodes[0].data =

document.getElementById("item").value;

submit.value = "Add Task";

document.getElementById("item").value = "";

document.getElementById("lblsuccess").innerHTML =

useAlert("Text edited successfully");

document.getElementById("lblsuccess").style.display = "block";

setTimeout(function () {

document.getElementById("lblsuccess").style.display = "none";

}, 3000);

return false;

}

}

if (newItem.trim() == "" || newItem.trim() == null) {

document.getElementById("lblsuccess").innerHTML =

useAlert("Please enter some data!", false);

document.getElementById("lblsuccess").style.display = "block";

setTimeout(function () {

document.getElementById("lblsuccess").style.display = "none";

}, 3000);

return false;

} else document.getElementById("item").value = "";

let li = createMainLiTaskElement(newItem);

li.appendChild(createTaskButtons());

items.appendChild(li);

addlocaltodo(newItem);

}

function handleToDoButtonClick(e) {

e.preventDefault();

// Handle delete todo button click

if (e.target.classList.contains("delete")) {

if (confirm("Are you Sure?")) {

let li = e.target.parentNode.parentNode;

items.removeChild(li);

removelocaltodo(li.childNodes[0].data);

document.getElementById("lblsuccess").innerHTML =

useAlert("Text deleted successfully");

document.getElementById("lblsuccess").style.display = "block";

setTimeout(function () {

document.getElementById("lblsuccess").style.display = "none";

}, 3000);

}

}

// Handle complete todo button click

if (e.target.classList.contains("comp")) {

if (confirm("Are you Sure?")) {

let li = e.target.parentNode.parentNode;

li.removeChild(li.childNodes[1]);

completeTask(li.innerText);

items.removeChild(li);

removelocaltodo(li.childNodes[0].data);

addlocaldone(li.childNodes[0].data)

document.getElementById("lblsuccess").innerHTML =

useAlert("Successfully Marked as Completed!");

document.getElementById("lblsuccess").style.display = "block";

setTimeout(function () {

document.getElementById("lblsuccess").style.display = "none";

}, 3000);

}

}

// Handle edit task button click

if (e.target.classList.contains("edit")) {

document.getElementById("item").value =

e.target.parentNode.parentNode.childNodes[0].data;

submit.value = "EDIT";

editItem = e;

}

// Handle important todo button click

if(e.target.classList.contains("imp")) {

let button = e.target;

let li = button.parentNode.parentNode;

if(li.classList.contains("important")) {

button.innerText = "Mark as important";

li.classList.remove("important");

removelocalimp(li.childNodes[0].data);

} else {

button.innerText = "Mark as normal";

li.classList.add("important");

items.insertBefore(li, items.childNodes[0]);

addlocalimp(li.childNodes[0].data);

}

}

}

function toggleButton(ref, btnID) {

document.getElementById(btnID).disabled = false;

}

function completeTask(innerText){

let li = createMainLiTaskElement(innerText)

li.appendChild(createButtonForCompletedTasks());

completedItems.appendChild(li);

}

function createTaskButtons() {

let actions = document.createElement("div");

actions.className = "actions flex-grow-0 align-self-start ml-2";

let deleteButton = document.createElement("button");

deleteButton.className = "btn-danger btn btn-sm mr-2 mb-2 delete";

deleteButton.appendChild(document.createTextNode("Delete Task "));

//icon for delete button

var img = document.createElement('img');

img.src = 'icons/delete.png';

deleteButton.appendChild(img);

let compButton = document.createElement("button");

compButton.className = "btn-success btn btn-sm mr-2 mb-2 comp";

compButton.appendChild(document.createTextNode("Mark as Complete "));

var doneImg = document.createElement('img');

doneImg.src = 'icons/done.png';

compButton.appendChild(doneImg);

let editButton = document.createElement("button");

editButton.className = "btn-warning btn btn-sm mr-2 mb-2 edit";

editButton.appendChild(document.createTextNode("Edit Task "));

//icon for edit button

var editImg = document.createElement('img');

editImg.src = 'icons/edit.png';

editButton.appendChild(editImg);

//Adding important button

let importantBtn = document.createElement("button");

importantBtn.className = "btn btn-warning btn-sm mr-2 mb-2 imp";

importantBtn.appendChild(document.createTextNode("Mark as important "));

var impImg = document.createElement('img');

impImg.src = 'icons/important.png';

importantBtn.appendChild(impImg);

actions.appendChild(deleteButton);

actions.appendChild(editButton);

actions.appendChild(compButton);

actions.appendChild(importantBtn);

return actions;

}

function handleToDoButtonClickForCompleted(e){

e.preventDefault();

//Handle inProgress task button click

if(e.target.classList.contains("inpro")) {

if (confirm("Are you Sure?")) {

let completedTaskLi = e.target.parentNode.parentNode;

completedTaskLi.removeChild(completedTaskLi.childNodes[1])

let li = createMainLiTaskElement(completedTaskLi.innerText)

li.appendChild(createTaskButtons());

items.appendChild(li);

completedItems.removeChild(completedTaskLi);

removelocaldone(li.childNodes[0].data);

addlocaltodo(li.childNodes[0].data);

operationCompleted(useAlert("Successfully Marked as In-Progress!"));

}

}

//Handle Remove Item After Completion click

if(e.target.classList.contains("remove")) {

if (confirm("Are you Sure? This Item will be Permanently Removed.")) {

let completedTaskLi = e.target.parentNode.parentNode;

completedTaskLi.removeChild(completedTaskLi.childNodes[1])

completedItems.removeChild(completedTaskLi);

removelocaldone(completedTaskLi.innerText);

operationCompleted(useAlert("Item Successfully Removed from List."))

}

}

}

function createButtonForCompletedTasks(){

let actions = document.createElement("div");

actions.className = "actions flex-grow-0 align-self-start ml-2";

let moveToInprogressButton = document.createElement("button");

moveToInprogressButton.className = "btn-success btn btn-sm mr-2 mb-2 inpro";

moveToInprogressButton.appendChild(document.createTextNode("Mark as In-Progress"));

actions.appendChild(moveToInprogressButton);

let removeCompletedButton = document.createElement("button");

removeCompletedButton.className = "btn-danger btn btn-sm mr-2 mb-2 remove";

removeCompletedButton.appendChild(document.createTextNode("Remove"));

actions.appendChild(removeCompletedButton);

return actions;

}

function createMainLiTaskElement(taskText) {

let li = document.createElement("li");

li.className =

"list-group-item d-flex flex-row align-items-center justify-content-between flex-sm-wrap";

li.appendChild(document.createTextNode(taskText));

return li;

}

function addlocaltodo(task){

todo.push(task);

localStorage.setItem('todo', JSON.stringify(todo));

}

function removelocaltodo(task){

// console.log(task);

for(let i=0; i<todo.length; i++){

if(todo[i] === task){

todo.splice(i,1);

break;

}

}

localStorage.setItem('todo', JSON.stringify(todo));

}

function addlocaldone(task){

done.push(task);

localStorage.setItem('done', JSON.stringify(done));

}

function removelocaldone(task){

// console.log(task);

for(let i=0; i<done.length; i++){

if(done[i] === task){

done.splice(i,1);

break;

}

}

localStorage.setItem('done', JSON.stringify(done));

}

function addlocalimp(task){

imp.push(task);

localStorage.setItem('imp', JSON.stringify(imp));

}

function removelocalimp(task){

for(let i=0; i< imp.length; i++){

if(imp[i] === task){

imp.splice(i, 1);

break;

}

}

localStorage.setItem('imp', JSON.stringify(imp));

}