**HTML**

<body>

<div class="container">

<h1>SAVEETHA To-Do List</h1>

<div class="input-area">

<input type="text" id="todo-input" placeholder="Add a new task...">

<button id="add-btn" onclick="addTodo()">Add Task</button>

</div>

<ul id="todo-list"></ul>

<div class="counter">

<p>Total Tasks: <span id="task-count">0</span></p>

</div>

</div>

<!-- Font Awesome for icons -->

<script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous"></script>

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

</body>

**CSS**

/\* Import Google Fonts \*/

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap');

/\* General Styles \*/

body {

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

background: linear-gradient(to right, #FFDEE9, #B5FFFC); /\* Soft Gradient Background \*/

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

color: #333;

}

.container {

background-color: rgba(255, 255, 255, 0.85); /\* Semi-transparent white \*/

padding: 30px 20px;

border-radius: 15px;

box-shadow: 0 10px 25px rgba(0, 0, 0, 0.2);

width: 90%;

max-width: 500px;

text-align: center;

}

h1 {

font-size: 2.5em;

margin-bottom: 25px;

color: #1E90FF; /\* Vibrant Blue \*/

}

.input-area {

display: flex;

justify-content: space-between;

margin-bottom: 25px;

}

#todo-input {

width: 70%;

padding: 12px 15px;

font-size: 16px;

border-radius: 30px;

border: 2px solid #1E90FF;

outline: none;

transition: border-color 0.3s ease;

}

#todo-input:focus {

border-color: #FF69B4; /\* Bright Pink on focus \*/

}

#add-btn {

padding: 12px 20px;

font-size: 16px;

border-radius: 30px;

border: none;

background-color: #1E90FF; /\* Vibrant Blue \*/

color: white;

cursor: pointer;

transition: background-color 0.3s ease, transform 0.2s ease;

display: flex;

align-items: center;

justify-content: center;

}

#add-btn:hover {

background-color: #FF69B4; /\* Bright Pink on hover \*/

transform: scale(1.05);

}

#add-btn i {

margin-left: 8px;

}

#todo-list {

list-style-type: none;

padding: 0;

margin: 0;

text-align: left;

}

#todo-list li {

background-color: #FFF;

margin: 10px 0;

padding: 15px 20px;

border-radius: 30px;

display: flex;

justify-content: space-between;

align-items: center;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

transition: transform 0.2s ease, background-color 0.3s ease;

}

#todo-list li:hover {

transform: translateY(-3px);

background-color: #FFDEE9; /\* Light Pink \*/

}

#todo-list li.completed {

text-decoration: line-through;

opacity: 0.6;

}

#todo-list li button {

border: none;

background-color: #FF6347; /\* Tomato Red \*/

color: white;

padding: 8px 12px;

border-radius: 20px;

cursor: pointer;

transition: background-color 0.3s ease, transform 0.2s ease;

display: flex;

align-items: center;

justify-content: center;

}

#todo-list li button:hover {

background-color: #FF4500; /\* Orange Red on hover \*/

transform: scale(1.1);

}

.counter {

margin-top: 25px;

font-size: 1.3em;

color: #32CD32; /\* Fresh Green \*/

}

.counter span {

font-weight: bold;

color: #1E90FF; /\* Vibrant Blue \*/

}

/\* Responsive Design \*/

@media (max-width: 600px) {

.container {

padding: 25px 15px;

}

h1 {

font-size: 2em;

}

#todo-input {

width: 65%;

}

#add-btn {

padding: 10px 15px;

}

#todo-list li {

padding: 12px 15px;

}

.counter {

font-size: 1.1em;

}

}

**JS**

import java.util.ArrayList;

import java.util.Scanner;

class Task {

private String description;

private boolean isCompleted;

public Task(String description) {

this.description = description;

this.isCompleted = false;

}

public String getDescription() {

return description;

}

public boolean isCompleted() {

return isCompleted;

}

public void markAsCompleted() {

this.isCompleted = true;

}

@Override

public String toString() {

return (isCompleted ? "[X] " : "[ ] ") + description;

}

}

class ToDoList {

private ArrayList<Task> tasks;

public ToDoList() {

tasks = new ArrayList<>();

}

public void addTask(String description) {

tasks.add(new Task(description));

}

public void removeTask(int index) {

if (index >= 0 && index < tasks.size()) {

tasks.remove(index);

} else {

System.out.println("Invalid task index.");

}

}

public void markTaskAsCompleted(int index) {

if (index >= 0 && index < tasks.size()) {

tasks.get(index).markAsCompleted();

} else {

System.out.println("Invalid task index.");

}

}

public void displayTasks() {

if (tasks.isEmpty()) {

System.out.println("Your to-do list is empty.");

} else {

for (int i = 0; i < tasks.size(); i++) {

System.out.println(i + ". " + tasks.get(i));

}

}

}

}

public class ToDoListApp {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

ToDoList toDoList = new ToDoList();

boolean running = true;

while (running) {

System.out.println("\nTo-Do List:");

toDoList.displayTasks();

System.out.println("\nOptions: ");

System.out.println("1. Add a task");

System.out.println("2. Remove a task");

System.out.println("3. Mark a task as completed");

System.out.println("4. Exit");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

System.out.print("Enter task description: ");

String description = scanner.nextLine();

toDoList.addTask(description);

break;

case 2:

System.out.print("Enter task index to remove: ");

int removeIndex = scanner.nextInt();

toDoList.removeTask(removeIndex);

break;

case 3:

System.out.print("Enter task index to mark as completed: ");

int completeIndex = scanner.nextInt();

toDoList.markTaskAsCompleted(completeIndex);

break;

case 4:

running = false;

break;

default:

System.out.println("Invalid option. Try again.");

}

}

System.out.println("Goodbye!");

scanner.close();

}

}