**To-Do List Application**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>To-Do List Application</title>

  <!-- Bootstrap CSS -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

  <!-- Bootstrap Icons -->

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">

  <!-- Custom CSS -->

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

</head>

<body data-bs-theme="light">

  <div class="container mt-4">

    <h1 class="mb-4 text-center">To-Do List</h1>

    <!-- Task Input -->

    <div class="input-group mb-3">

      <input type="text" id="taskInput" class="form-control" placeholder="What needs to be done?">

      <button id="addTask" class="btn btn-primary"><i class="bi bi-plus-lg"></i></button>

    </div>

    <!-- Extra Fields -->

    <div class="row g-2 mb-3">

      <div class="col-md-4">

        <label class="form-label">Category</label>

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

          <option value="Work">Work</option>

          <option value="Personal">Personal</option>

          <option value="Shopping">Shopping</option>

        </select>

      </div>

      <div class="col-md-4">

        <label class="form-label">Priority</label>

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

          <option value="">No Priority</option>

          <option value="low">Low</option>

          <option value="medium">Medium</option>

          <option value="high">High</option>

        </select>

      </div>

      <div class="col-md-4">

        <label class="form-label">Due Date</label>

        <input type="datetime-local" id="dueInput" class="form-control">

      </div>

    </div>

    <!-- Tags Input -->

    <input type="text" id="tagsInput" class="form-control mb-3" placeholder="Tags (comma separated, optional)">

    <input type="text" id="searchInput" class="form-control mb-3" placeholder="Search tasks...">

    <!-- Task List -->

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

    <!-- Filters -->

    <div class="filter-btns mt-3">

      <button id="filterAll" class="btn btn-outline-secondary active">All</button>

      <button id="filterActive" class="btn btn-outline-secondary">Active</button>

      <button id="filterCompleted" class="btn btn-outline-secondary">Completed</button>

    </div>

    <!-- Actions -->

    <div class="action-btns mt-3">

      <button id="clearCompleted" class="btn btn-outline-danger">Clear Completed</button>

      <button id="toggleTheme" class="btn btn-outline-secondary">Dark Mode</button>

    </div>

    <!-- Quick Stats -->

    <div class="row text-center mt-4" id="quickStats">

      <div class="col">

        <div class="p-3 bg-light border rounded"> <h5>2</h5> <p>Total Tasks</p></div>

      </div>

      <div class="col">

        <div class="p-3 bg-light border rounded"> <h5>1</h5> <p>Completed</p></div>

      </div>

      <div class="col">

        <div class="p-3 bg-light border rounded"> <h5>1</h5> <p>Active</p></div>

      </div>

      <div class="col">

        <div class="p-3 bg-light border rounded"> <h5>0</h5> <p>Overdue</p></div>

      </div>

    </div>

    <!-- Progress -->

    <div id="progressBar" class="mt-4">

      <div class="progress">

        <div id="progressBarInner" class="progress-bar" role="progressbar"

          style="width: 0%;" aria-valuenow="0" aria-valuemin="0" aria-valuemax="100"></div>

      </div>

    </div>

  </div>

  <!-- Scripts -->

  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

  <script src="script.js"></script>

</body>

</html>

**script.js**

$(document).ready(function () {

  let tasks = JSON.parse(localStorage.getItem("tasks")) || [];

  let currentFilter = "all";

  let searchQuery = "";

  // Save to localStorage

  function saveTasks() {

    localStorage.setItem("tasks", JSON.stringify(tasks));

  }

  // Render tasks

  function renderTasks() {

    $("#taskList").empty();

    let filteredTasks = tasks.filter((task) => {

      const matchesFilter =

        (currentFilter === "active" && !task.completed) ||

        (currentFilter === "completed" && task.completed) ||

        currentFilter === "all";

      const matchesSearch =

        task.text.toLowerCase().includes(searchQuery.toLowerCase()) ||

        (task.tags &&

          task.tags.some((tag) =>

            tag.toLowerCase().includes(searchQuery.toLowerCase())

          ));

      return matchesFilter && matchesSearch;

    });

    filteredTasks.forEach((task, index) => {

      const dueDate = task.due ? new Date(task.due) : null;

      const isOverdue =

        dueDate && dueDate < new Date() && !task.completed ? "overdue" : "";

      const priorityBadge = task.priority

        ? `<span class="badge bg-${

            task.priority === "low"

              ? "success"

              : task.priority === "medium"

              ? "warning"

              : "danger"

          }">${task.priority}</span>`

        : "";

      const tagsBadges = task.tags

        ? task.tags

            .map((tag) => `<span class="badge bg-secondary">${tag}</span>`)

            .join(" ")

        : "";

      const dueText = dueDate

        ? `Due: ${dueDate.toLocaleDateString()} ${dueDate.toLocaleTimeString([], {

            hour: "2-digit",

            minute: "2-digit",

          })}`

        : "";

      const taskHtml = `

        <li class="list-group-item d-flex justify-content-between align-items-center" data-index="${index}">

          <div>

            <input type="checkbox" class="form-check-input me-2 task-checkbox" ${

              task.completed ? "checked" : ""

            }>

            <span class="task-text ${

              task.completed ? "completed" : ""

            }">${task.text}</span>

            <div class="small text-muted">

              ${priorityBadge} <span class="${isOverdue}">${dueText}</span> ${tagsBadges}

            </div>

          </div>

          <div>

            <button class="btn btn-sm btn-outline-primary edit-task"><i class="bi bi-pencil"></i></button>

            <button class="btn btn-sm btn-outline-danger delete-task"><i class="bi bi-trash"></i></button>

          </div>

        </li>

      `;

      $("#taskList").append(taskHtml);

    });

    updateQuickStats();

    updateProgressBar();

  }

  // Add task

  $("#addTask").click(function () {

    const taskText = $("#taskInput").val().trim();

    const category = $("#categoryInput").val();

    const priority = $("#priorityInput").val();

    const due = $("#dueInput").val();

    const tags = $("#tagsInput")

      .val()

      .trim()

      .split(",")

      .map((t) => t.trim())

      .filter((t) => t);

    if (taskText) {

      tasks.push({

        text: taskText,

        category: category || "General",

        completed: false,

        priority: priority || null,

        due: due || null,

        tags: tags.length ? tags : null,

      });

      $("#taskInput").val("");

      $("#priorityInput").val("");

      $("#dueInput").val("");

      $("#tagsInput").val("");

      saveTasks();

      renderTasks();

    }

  });

  // Mark complete

  $(document).on("change", ".task-checkbox", function () {

    const index = $(this).closest("li").data("index");

    tasks[index].completed = $(this).is(":checked");

    saveTasks();

    renderTasks();

  });

  // Delete

  $(document).on("click", ".delete-task", function () {

    const index = $(this).closest("li").data("index");

    tasks.splice(index, 1);

    saveTasks();

    renderTasks();

  });

  // Edit task

  $(document).on("click", ".edit-task", function () {

    const index = $(this).closest("li").data("index");

    const newText = prompt("Edit Task:", tasks[index].text);

    if (newText !== null && newText.trim() !== "") {

      tasks[index].text = newText.trim();

      saveTasks();

      renderTasks();

    }

  });

  // Filters

  $("#filterAll").click(() => {

    currentFilter = "all";

    $(".filter-btns button").removeClass("active");

    $("#filterAll").addClass("active");

    renderTasks();

  });

  $("#filterActive").click(() => {

    currentFilter = "active";

    $(".filter-btns button").removeClass("active");

    $("#filterActive").addClass("active");

    renderTasks();

  });

  $("#filterCompleted").click(() => {

    currentFilter = "completed";

    $(".filter-btns button").removeClass("active");

    $("#filterCompleted").addClass("active");

    renderTasks();

  });

  // Search

  $("#searchInput").on("keyup", function () {

    searchQuery = $(this).val();

    renderTasks();

  });

  // Clear completed

  $("#clearCompleted").click(() => {

    tasks = tasks.filter((t) => !t.completed);

    saveTasks();

    renderTasks();

  });

  // Theme toggle

  $("#toggleTheme").click(function () {

    const theme = $("body").attr("data-bs-theme") === "light" ? "dark" : "light";

    $("body").attr("data-bs-theme", theme);

    $(this).text(theme === "light" ? "Dark Mode" : "Light Mode");

  });

  // Update Quick Stats

  function updateQuickStats() {

    const total = tasks.length;

    const completed = tasks.filter((t) => t.completed).length;

    const active = total - completed;

    const overdue = tasks.filter(

      (t) => t.due && new Date(t.due) < new Date() && !t.completed

    ).length;

    const stats = [total, completed, active, overdue];

    $("#quickStats .col").each(function (i) {

      $(this).find("h5").text(stats[i]);

    });

  }

  // Progress bar

  function updateProgressBar() {

    const total = tasks.length;

    const completed = tasks.filter((t) => t.completed).length;

    const percentage = total > 0 ? (completed / total) \* 100 : 0;

    $("#progressBarInner").css("width", `${percentage}%`);

  }

  // Initial render

  renderTasks();

});

**style.css**

body {

  background: #f8f9fa;

  font-family: 'Poppins', sans-serif;

  color: #2c3e50;

}

h1 {

  font-weight: 600;

}

.input-group input {

  border-radius: 10px 0 0 10px;

}

.input-group button {

  border-radius: 0 10px 10px 0;

}

.list-group-item {

  display: flex;

  justify-content: space-between;

  align-items: center;

}

.task-text.completed {

  text-decoration: line-through;

  color: gray;

}

.filter-btns button.active {

  background: #0d6efd;

  color: white;

}

#quickStats .col div {

  background: #e9f2ff;

  border: 1px solid #cfe2ff;

  border-radius: 10px;

}