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 app</title>

    <!-- Linkki ulkoiseen CSS-tiedostoon -->

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

  </head>

  <body>

    <div class="container">

      <!-- Sovelluksen otsikko -->

      <h1>📝 Todo-app</h1>

      <!-- Lomake uuden tehtävän lisäämiseen -->

      <form id="todoForm">

        <input type="text" id="todoInput" placeholder="Add a new task..." />

        <button type="submit">Add</button>

      </form>

      <!-- Virheilmoitus tyhjälle tai liian lyhyelle syötteelle -->

      <p id="error" class="error"></p>

      <!-- Laskuri auki oleville tehtäville -->

      <p id="counter" class="counter">Open tasks: 0</p>

      <!-- Lista päätehtävistä -->

      <ul id="todoList"></ul>

      <!-- Active / Done napit samalla rivillä -->

      <div id="filterButtons">

        <button id="showAllBtn" type="button">Show all tasks</button>

        <button id="showActiveBtn" type="button">Active</button>

        <button id="showDoneBtn" type="button">Done</button>

      </div>

      <!-- Yhteenvetolista -->

      <ul id="summaryList"></ul>

    </div>

    <script src="script.js"></script>

  </body>

</html>

STYLE.CSS

body {

  font-family: Arial, sans-serif;

  background-image: url('Background\_.jpg'); /\* Päätin laittaa taustakuvan jotta olisi paremman näköinen \*/

  background-size: cover;       /\* täyttää koko näytön \*/

  background-position: center;  /\* keskittää kuvan \*/

  background-repeat: no-repeat; /\* ei toista kuvaa \*/

  display: flex;

  justify-content: center;

  align-items: center;

  height: 100vh;

}

.container {

  background: rgba(0,0,0,0); /\* täysin läpinäkyvä \*/

  padding: 20px 30px;

  border-radius: 12px;

  border: 2px solid #000000;

  width: 350px;

}

#summaryList {

  margin-top: 15px;

  background-color: #eef3f8;

  padding: 10px;

  border-radius: 8px;

}

form {

  display: flex;

  gap: 10px;

}

input {

  flex: 1;

  padding: 8px;

  border: 1px solid #ccc;

  border-radius: 6px;

}

input.error-border {

  border: 2px solid red;

}

/\* Kaikki napit tumman pinkiksi \*/

button {

  background-color: #c2185b; /\* tumman pinkki \*/

  color: white;

  border: none;

  padding: 8px 12px;

  border-radius: 6px;

  cursor: pointer;

}

/\* Hover-efekti hieman tummempana pinkkinä \*/

button:hover {

  background-color: #880e4f;

}

ul {

  list-style: none;

  padding: 0;

  margin-top: 15px;

}

li {

  background: #f9f9f9;

  padding: 8px;

  margin-bottom: 8px;

  display: flex;

  justify-content: space-between;

  align-items: center;

  border-radius: 6px;

}

li.done {

  text-decoration: line-through;

  color: gray;

}

.error {

  color: red;

  font-size: 0.9em;

}

SCRIPT.JS

const form = document.getElementById('todoForm');

const input = document.getElementById('todoInput');

const list = document.getElementById('todoList');

const error = document.getElementById('error');

const summaryList = document.getElementById('summaryList');

const counter = document.getElementById('counter'); // Käytetään HTML:n laskuria

const showAllBtn = document.getElementById('showAllBtn');

const showActiveBtn = document.getElementById('showActiveBtn');

const showDoneBtn = document.getElementById('showDoneBtn');

let todos = JSON.parse(localStorage.getItem('todos')) || [];

let summaryVisible = false;

let filter = 'all'; // all, active, done

function saveTodos() {

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

}

function renderCounter() {

    const activeCount = todos.filter(todo => !todo.done).length;

    counter.textContent = `Open tasks: ${activeCount}`;

}

function renderTodos() {

    list.innerHTML = '';

    todos.forEach((todo, index) => {

        if ((filter === 'active' && todo.done) || (filter === 'done' && !todo.done)) return;

        const li = document.createElement('li');

        li.className = todo.done ? 'done' : '';

        const span = document.createElement('span');

        span.textContent = todo.text;

        li.appendChild(span);

        const buttons = document.createElement('div');

        const doneBtn = document.createElement('button');

        doneBtn.textContent = '✔';

        doneBtn.onclick = () => toggleDone(index);

        const delBtn = document.createElement('button');

        delBtn.textContent = '🗑';

        delBtn.onclick = () => deleteTodo(index);

        buttons.append(doneBtn, delBtn);

        li.appendChild(buttons);

        list.appendChild(li);

    });

    renderCounter();

}

function renderSummary() {

    summaryList.innerHTML = '';

    todos.forEach(todo => {

        if ((filter === 'active' && todo.done) || (filter === 'done' && !todo.done)) return;

        const li = document.createElement('li');

        li.textContent = todo.text + (todo.done ? ' (done)' : ' (pending)');

        summaryList.appendChild(li);

    });

}

function addTodo(text) {

    todos.push({ text, done: false });

    saveTodos();

    renderTodos();

    if (summaryVisible) renderSummary();

}

function toggleDone(index) {

    todos[index].done = !todos[index].done;

    saveTodos();

    renderTodos();

    if (summaryVisible) renderSummary();

}

function deleteTodo(index) {

    todos.splice(index, 1);

    saveTodos();

    renderTodos();

    if (summaryVisible) renderSummary();

}

form.addEventListener('submit', (e) => {

    e.preventDefault();

    const value = input.value.trim();

    if (value.length < 3) {

        error.textContent = 'Type at least 3 characters!';

        input.classList.add('error-border');

        return;

    }

    error.textContent = '';

    input.classList.remove('error-border');

    addTodo(value);

    input.value = '';

});

// Show all tasks ja summary

showAllBtn.addEventListener('click', () => {

    summaryVisible = !summaryVisible;

    if (summaryVisible) {

        renderSummary();

        summaryList.style.display = 'block';

        showAllBtn.textContent = 'Hide tasks';

    } else {

        summaryList.style.display = 'none';

        showAllBtn.textContent = 'Show all tasks';

    }

});

// Filter napit

showActiveBtn.addEventListener('click', () => {

    filter = 'active';

    renderTodos();

    if (summaryVisible) renderSummary();

});

showDoneBtn.addEventListener('click', () => {

    filter = 'done';

    renderTodos();

    if (summaryVisible) renderSummary();

});

// Alusta

renderTodos();

summaryList.style.display = 'none';