Lab — JavaScript vs Vue.js: Task 1  
ToDo List in Vue.js

# CDN include (use this one)

<script src="https://unpkg.com/vue@2.7.8/dist/vue.js"></script>

# What changes with Vue?

* Declarative templates: HTML reflects your data automatically.
* Reactivity: when data changes, Vue updates the DOM for you.
* Two-way binding: v-model wires inputs ↔ data.
* v-for loops: render arrays succinctly.

# Complete example

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>ToDo List — Vue.js</title>

<script src="https://unpkg.com/vue@2.7.8/dist/vue.js"></script>

</head>

<body>

<div id="app">

<header>

<input v-model.trim="newTask" type="text" placeholder="New task..." @keyup.enter="addTask" />

<button @click="addTask">Add task</button>

</header>

<ul>

<li v-for="(t, i) in tasks" :key="i">{{ t }}</li>

</ul>

</div>

<script>

new Vue({

el: '#app',

data: {

newTask: '',

tasks: ['Task 1', 'Task 2', 'Task 3']

},

methods: {

addTask() {

if (!this.newTask) return;

this.tasks.push(this.newTask);

this.newTask = '';

}

}

});

</script>

</body>

</html>

# Assessment checklist

* Vue 2.7.8 loaded via CDN.
* Input uses v-model.trim and supports Enter key submit.
* Clicking Add pushes to tasks[] and the list updates automatically.