<!DOCTYPE html>

<html>

<head>

<title>State Dropdown</title>

<style>

select {

height: 35px;

padding: 5px 5px;

border: 1px solid #ccc;

font-size: 14px;

border-radius: 7px;

width: 169px;

margin-bottom: 15px;

}

select:focus {

border-color: #f86200; /\* Active color \*/

}

/\* Updated style for checkboxes \*/

.checkbox input[type="checkbox"] {

display: none;

}

.checkmark {

position: relative;

display: inline-block;

width: 20px;

height: 20px;

background-color: blue;

border-radius: 50%;

margin-right: 8px;

}

.checkbox input[type="checkbox"]:checked + .checkmark::after {

content: "";

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

width: 6px;

height: 6px;

background-color: white;

border-radius: 50%;

}

</style>

</head>

<body>

<h1>State Dropdown</h1>

<div class="dropdown">

<select class="stateDropdown">

<option value="" disabled selected>Select a state</option>

<option value="Alabama">Alabama</option>

<!-- Add all 50 states as options here -->

</select>

</div>

<div id="checkboxesContainer">

<div class="checkbox-container">

<label class="checkbox">

<input type="checkbox" name="gridOption" value="gridExist" onclick="handleCheckbox(this)">

<span class="checkmark"></span> Grid Exist

</label>

<label class="checkbox">

<input type="checkbox" name="gridOption" value="newGrid" onclick="handleCheckbox(this)">

<span class="checkmark"></span> New Grid

</label>

<label class="checkbox">

<input type="checkbox" name="gridOption" value="approveGrid" onclick="handleCheckbox(this)">

<span class="checkmark"></span> Approve Grid

</label>

</div>

</div>

<button class="add-on-btn" onclick="addOn()">Add On</button>

<script>

function handleCheckbox(checkbox) {

const checkboxes = document.getElementsByName("gridOption");

const currentStateDropdown = checkbox.closest(".container").querySelector(".stateDropdown");

const selectedState = currentStateDropdown.value;

// If "Grid Exist" is selected, disable other grid options

if (checkbox.value === "gridExist" && checkbox.checked) {

checkboxes.forEach((cb) => {

if (cb !== checkbox) {

cb.checked = false;

cb.disabled = true;

}

});

} else {

checkboxes.forEach((cb) => (cb.disabled = false));

}

// If "New Grid" is selected, enable "Grid Approval" checkbox

if (checkbox.value === "newGrid" && checkbox.checked) {

const gridApprovalCheckbox = checkbox.closest(".container").querySelector(

'input[value="approveGrid"]'

);

if (gridApprovalCheckbox) {

gridApprovalCheckbox.disabled = false;

}

}

}

function addOn() {

const currentStateDropdown = document.querySelector('.stateDropdown');

const currentState = currentStateDropdown.value;

if (currentState) {

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

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

newContainer.className = 'container';

const newDropdown = currentStateDropdown.cloneNode(true);

newDropdown.value = '';

newContainer.appendChild(newDropdown);

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

newCheckboxContainer.className = 'checkbox-container';

const checkboxLabels = ['Grid Exist', 'New Grid', 'Approve Grid'];

for (const label of checkboxLabels) {

const checkboxLabel = document.createElement('label');

checkboxLabel.className = 'checkbox';

checkboxLabel.innerHTML = `

<input type="checkbox" name="gridOption" value="${label.replace(' ', '').toLowerCase()}">

<span class="checkmark"></span> ${label}

`;

newCheckboxContainer.appendChild(checkboxLabel);

}

for (const checkbox of newCheckboxContainer.querySelectorAll('input[type="checkbox"]')) {

checkbox.addEventListener("click", () => handleCheckbox(checkbox));

}

newContainer.appendChild(newCheckboxContainer);

checkboxesContainer.appendChild(newContainer);

}

}

</script>

</body>

</html>