<!DOCTYPE html>

<html>

<head>

<title>State Dropdown</title>

<link rel="stylesheet" type="text/css" href="StatesRadioDropdown.css">

<link rel="stylesheet" type="text/css" href="//fonts.googleapis.com/css?family=Open+Sans" />

</head>

<body>

<h1>State Dropdown</h1>

<div class="dropdown">

    <select class="stateDropdown">

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

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

    <option value="AK">Alaska</option>

    <option value="AZ">Arizona</option>

    <option value="AR">Arkansas</option>

    <option value="CA">California</option>

    <option value="CO">Colorado</option>

    <option value="CT">Connecticut</option>

    <option value="DE">Delaware</option>

    <option value="DC">District of Columbia</option>

    <option value="FL">Florida</option>

    <option value="GA">Georgia</option>

    <option value="HI">Hawaii</option>

    <option value="ID">Idaho</option>

    <option value="IL">Illinois</option>

    <option value="IN">Indiana</option>

    <option value="IA">Iowa</option>

    <option value="KS">Kansas</option>

    <option value="KY">Kentucky</option>

    <option value="LA">Louisiana</option>

    <option value="ME">Maine</option>

    <option value="MD">Maryland</option>

    <option value="MA">Massachusetts</option>

    <option value="MI">Michigan</option>

    <option value="MN">Minnesota</option>

    <option value="MS">Mississippi</option>

    <option value="MO">Missouri</option>

    <option value="MT">Montana</option>

    <option value="NE">Nebraska</option>

    <option value="NV">Nevada</option>

    <option value="NH">New Hampshire</option>

    <option value="NJ">New Jersey</option>

    <option value="NM">New Mexico</option>

    <option value="NY">New York</option>

    <option value="NC">North Carolina</option>

    <option value="ND">North Dakota</option>

    <option value="OH">Ohio</option>

    <option value="OK">Oklahoma</option>

    <option value="OR">Oregon</option>

    <option value="PA">Pennsylvania</option>

    <option value="RI">Rhode Island</option>

    <option value="SC">South Carolina</option>

    <option value="SD">South Dakota</option>

    <option value="TN">Tennessee</option>

    <option value="TX">Texas</option>

    <option value="UT">Utah</option>

    <option value="VT">Vermont</option>

    <option value="VA">Virginia</option>

    <option value="WA">Washington</option>

    <option value="WV">West Virginia</option>

    <option value="WI">Wisconsin</option>

    <option value="WY">Wyoming</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>

    select {

      height: 35px;

      padding: 5px 5px;

      border: 1px solid #ccc;

      font-size: 14px;

      border-radius: 7px;

      width: 169px;

      margin-bottom: 15px;

      margin-top: 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;

      border-radius: 50%;

      margin-right: 8px;

      background-color: #e0f2ff;

      border: 2px solid #b3d8ff;

      <span class="checkmark" style="

        margin-bottom: 0px;

        margin-top: -5;

        margin-top: -20;

        top: 5px;

        right: -5;

        left: 5px;

        right: 0px;

    "></span>

    }

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

      content: "";

      position: absolute;

      top: 50%;

      left: 50%;

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

      width: 6px;

      height: 6px;

      background-color: #0066ff;

      border-radius: 50%;

    }

    button {

  padding: 10px 20px;

  background-color: #004A94;

  color: white;

  border: none;

  border-radius: 5px;

  font-size: 16px;

  cursor: pointer;

  font-family: 'Open Sans', sans-serif;

  margin-top: 15px;

}

<!DOCTYPE html>

<html>

<head>

  <link rel="stylesheet" type="text/css" href="//fonts.googleapis.com/css?family=Open+Sans" />

  <link rel="stylesheet" type="text/css" href="StatesDropDownDatabase.css">

</head>

<body>

  <div id="FeaturesHeader">

    <h1></h1>

  </div>

  <div id="FeaturesContainer">

    <table id="coveragelimits">

      <tr>

        <th>State</th>

        <th>Grid Exist</th>

        <th>New Grid</th>

        <th>Approve Grid</th>

      </tr>

    </table>

  </div>

  <script>

    // Fetch the JSON data (Replace this with your server-side API endpoint)

  const jsonData = {

  "states": [

    {

      "stateName": "Alabama",

      "data": {

        "Grid Exist": true,

        "New Grid": false,

        "Approve Grid": false

      }

    },

    {

      "stateName": "Alaska",

      "data": {

        "Grid Exist": false,

        "New Grid": true,

        "Approve Grid": false

      }

    },

    {

      "stateName": "Arizona",

      "data": {

        "Grid Exist": false,

        "New Grid": true,

        "Approve Grid": true

      }

    },

    {

      "stateName": "Arkansas",

      "data": {

        "Grid Exist": false,

        "New Grid": true,

        "Approve Grid": true

      }

    },

    {

      "stateName": "California",

      "data": {

        "Grid Exist": true,

        "New Grid": false,

        "Approve Grid": false

      }

    },

    {

      "stateName": "Florida",

      "data": {

        "Grid Exist": false,

        "New Grid": true,

        "Approve Grid": false

      }

        },

        // Add more states with their respective data here

      ]

    };

    // Function to create table rows from the JSON data

    function createTableRow(stateName, gridExist, newGrid, approveGrid) {

      const tableRow = document.createElement("tr");

      tableRow.innerHTML = `

        <td>${stateName}</td>

        <td><label class="checkbox">

          <input type="checkbox" ${gridExist ? "checked" : ""} disabled />

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

        </label></td>

        <td><label class="checkbox">

          <input type="checkbox" ${newGrid ? "checked" : ""} disabled />

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

        </label></td>

        <td><label class="checkbox">

          <input type="checkbox" ${approveGrid ? "checked" : ""} disabled />

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

        </label></td>

      `;

      return tableRow;

    }

    // Function to populate the table with data from JSON

    function populateTable() {

      const tableBody = document.getElementById("coveragelimits");

      jsonData.states.forEach((state) => {

        const { stateName, data } = state;

        const { "Grid Exist": gridExist, "New Grid": newGrid, "Approve Grid": approveGrid } = data;

        const tableRow = createTableRow(stateName, gridExist, newGrid, approveGrid);

        tableBody.appendChild(tableRow);

      });

    }

    // Call the function to populate the table with data

    populateTable();

  </script>

</body>

</html>

#coveragelimits {

  font-family: 'Open Sans', sans-serif;

  border-collapse: collapse;

  width: auto;

  margin: auto;

}

#coveragelimits td, #coveragelimits th {

  border: 1px solid #ddd;

  padding: 12px;

}

#coveragelimits tr:nth-child(even) {

  background-color: #FFFFFF;

}

#coveragelimits th {

  padding-top: 10px;

  padding-bottom: 1px;

  text-align: center;

  background-color: #004A94;

  color: white;

  padding-right: 50px;

}

/\* Updated style for checkboxes \*/

.checkbox input[type="checkbox"] {

  display: none;

}

.checkmark {

  position: relative;

  display: inline-block;

  width: 20px;

  height: 20px;

  border-radius: 50%;

  margin-right: 8px;

  background-color: #e0f2ff;

  border: 2px solid #b3d8ff;

  <span class="checkmark" style="

    margin-bottom: 0px;

    margin-top: -5;

    margin-top: -20;

    top: 5px;

    left: 40px;

"></span>

}

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

  content: "";

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

  width: 6px;

  height: 6px;

  background-color: #0066ff;

  border-radius: 50%;

}